Quantum State Preparation using Adaptive Circuits

MSc Thesis *(Afstudeerscriptie)*

written by

Daan In de Braekt
(born September 16, 1999 in Amsterdam)

under the supervision of **Juan C. Boschero**, **Niels M.P. Neumann** and **prof. dr. Ronald de Wolf**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:	**Members of the Thesis Committee:**
April 10, 2025	dr. Benno van den Berg (chair)
	dr. John van de Wetering
	dr. Māris Ozols
	Niels M.P. Neumann MSc (supervisor)
	prof. dr. Ronald de Wolf (supervisor)

Institute for Logic, Language and Computation

Contents

Abstract

Quantum state preparation (QSP) plays a pivotal role in quantum computing, enabling quantum algorithms to process classical data efficiently. Determining the complexity of QSP is crucial for understanding whether quantum algorithms can offer exponential speed-ups over their classical counterparts. Recent advances in adaptive quantum circuits – circuits that are augmented with mid-circuit measurements and classical feedback – suggest potential advantages in reducing circuit depth, which is especially relevant given the high error rates of current quantum hardware. Previous work has demonstrated that adaptive circuits can have an exponentially higher success probability over non-adaptive counterparts in preparing specific quantum states, such as the GHZ-state and the W-state. However, it remains an open question whether these advantages extend to *general QSP* algorithms, and, in particular, to *sparse QSP*, where only a small subset of amplitudes are non-zero.

This thesis systematically examines the trade-offs between adaptive and non-adaptive circuits for both general and sparse QSP. We analyse two QSP approaches: (1) UCG-based QSP, which utilizes uniformly controlled gates, and (2) unary-based QSP, which encodes data in a unary representation before transformation. By leveraging known speed-ups for adaptive implementations of important multi-qubit gates, such as the Fanout-gate and the OR-gate, we assess the conditions under which adaptive circuits provide an exponential advantage. Our results indicate that adaptive circuits significantly improve success probabilities for both general and sparse QSP, particularly in regimes where qubit idling errors dominate.

Using empirical error rates from current quantum processors, we derive lower bounds on the number of qubits required to observe an adaptive advantage. While non-adaptive circuits assuming all-to-all connectivity generally perform well for preparing quantum states with less than 1000 qubits, adaptive circuits demonstrate substantial improvements when compared to non-adaptive circuits constrained to a 2D grid topology. These findings suggest that adaptivity could help in enabling efficient quantum state preparation, particularly for near-term quantum devices with limited connectivity.

Chapter 1

Introduction

1.1 General introduction

Quantum computing promises to revolutionize problem solving in ways classical computers cannot. Richard Feynman's seminal lecture *Simulating physics with computers* [Fey82] laid the foundations for quantum computing as a research field, leading to a surge in algorithms promising speed-ups over classical methods. The epitome of this is undoubtedly Shor's algorithm [Sho97], which can find the prime factors of an integer much faster than a classical computer can, and which could break current cryptographic schemes once large quantum computers are realised. Other notable examples include: Grover's algorithm [Gro96], algorithms for Hamiltonian simulation [Ber+15; LC19; Chi+18], the HHL algorithm [HHL09], and other quantum linear algebra algorithms [LMR14; Reb+18; Ker+18; KL21].

To process classical data – as is needed in many linear algebra tasks – the first step is usually to encode this data into a quantum state. This task of producing a specific quantum state given some classical description of the state is often called *quantum state preparation*, or *QSP*. A computationally inefficient preparation step may offset the promised speed-ups of the algorithm. In fact, for many quantum linear algebra algorithms, it has been shown that promised exponential speed-ups only exist *because* of the assumption of efficient QSP [Tan23].

Partially due to this mitigating result, understanding the complexity of QSP is a subject of ongoing study. Early algorithms were described in [Zal98; GR02; KM04]. These essentially used the same basic idea, based on sequences of uniformly controlled gates. Later works improved upon these algorithms [Mot+04; YZ23; ZLY22; Sun+23]. Since a quantum state of n qubits can encode a maximum of 2^n distinct data points, any algorithm for preparing a general n-qubit quantum state unvoidably incurs a scaling that is exponential in n, which either manifests itself in the number of qubits it uses, or in the time it takes to run the algorithm.

On today's quantum computers, this time is still limited. Since the reported error rates for current hardware are above the rates required by the Threshold Theorem [NC10, Section 10.6.4] to enable error correction and fault-tolerant quantum computing, researches will have to deal with errors that accumulate during the computation of a quantum circuit. In this context, it is natural to reduce the time needed to execute all the gates in a quantum circuit, or, in other words, to look for circuits that are *low-depth* or *shallow*.

On the topic of finding shallow quantum circuits, much advantage has recently been booked by the use of *adaptive circuits* [Buh+23; PSC24; Fos+23; PSC21; Bäu+24; Smi+24; Smi+23]. Adaptive circuits are regular quantum circuits that are augmented with the power of a classical controller. This classical controller can obtain the outcome of mid-circuit measurement, perform some classical computation on it, and, based on the outcome of this computation, control future quantum gates. In recent years, adaptive circuits have been shown to prepare structured, long-range entangled quantum states using circuits of constant depth[1], wheras the best-known non-adaptive methods require at least logarithmic depth (in the number of qubits).

[1] An asterisk should be attached, since 'depth' is defined differently for adaptive circuits. Formally, the circuits consist of a constantly many alternations of constant-depth quantum circuits and logarithmic depth classical circuits. See Section 3.3 for details.

Examples of these include the GHZ-state, the W-state [Buh+23], some Dicke-states [Buh+23], and specific classes of matrix product states [Smi+23; Smi+24].

Due to their asymptotically reduced depth, adaptive circuits usually incur fewer errors on idling qubits, primarily because these qubits are not idle for a very long time. At the same time, the circuits usually have a more elaborate structure, and require *more* quantum gates and *more* qubits than non-adaptive approaches. These, and more prominently the intermediate classical computations, may also form a potential source of error.

Intuitively, assuming quantum hardware supports adaptive circuits, if the error rates for applying quantum gates are relatively low and the probability of an error occurring while a qubit is idling is relatively high, we would expect the adaptive circuit to perform better than the non-adaptive circuit. Conversely, if gate errors are relatively high, and the probability of an error occuring while a qubit is idling is relatively low, then the non-adaptive circuit may be preferable.

To weigh this trade-off, Neumann [Neu25] compared the success probability of adaptive and non-adaptive circuits for the GHZ-state and W-state, in terms of the success probabilities of the individual components that make up each circuit. To determine these probabilities, he defined a worst-case error model describing the behavior of the quantum circuit in case operations are performed incorrectly or in case qubits decohere.

For preparing the n-qubit GHZ-state, he found that the adaptive circuit can have an exponentially higher success probability than a standard approach using all-to-all connectivity. A condition for this is that the probability of a CNOT-gate introducing an error is at most the probability that an error is introduced while a qubit is idling during $\Omega(\log n)$ CNOT-gates. Compared to a standard approach using a 1D grid qubit connectivity, he found that this advantage exists already if the error rate of a CNOT-gate is at most the probability of an error introduced while a qubit is idling during $\Omega(n)$ CNOT-gates. Similarly, for the W-state, he found that the adaptive approach had an exponentially greater success probability than a standard approach using a 1D grid connectivity, if the error rate of a CNOT-gate is at most the probability of an error introduced while a qubit is idling during $\Omega(n \log(n) \log\log(n))$ CNOT-gates. This was expected based on the respective circuit sizes.

An open question stemming from this research, is whether these adaptive advantages extend to *general* quantum state preparation schemes, i.e. to algorithms that prepare an arbitrary quantum state based on some classical description of it.

1.2 This thesis

The present thesis aims to answer this question. Specifically, our main research question will be the following:

'under which conditions can we expect an advantage in using adaptive circuits over non-adaptive circuits for quantum state preparation?'

To tackle this question, we adopt the same error model as [Neu25], and determine the success probability of two types of QSP algorithms. The first type, which we will call *UCG-based*, originates from the earlier results in quantum state preparation and use uniformly controlled gates (UCGs) to sequentially prepare a quantum state. We will look at two implementations of these UCGs: a sequential and a parallelized one. The second type, which we will call *unary-based*, prepares the required state in *unary encoding* [Joh+20], and then transform it into the required (binary-encoded) quantum state.

Both types of algorithms make considerable use of gates that see a theoretic speed-up in the adaptive setting, such as the Fanout-gate and the OR-gate. Our analysis will therefore start by comparing the adaptive and non-adaptive implementations of these gates. Specifically, we will compare the adaptive circuit to a non-adaptive approach using an all-to-all qubit connectivity, and a standard approach using 2D grid qubit connectivity.

These results will then be used in the analyses of the QSP algorithms. Additionally, we will discuss a version of QSP, called *sparse QSP*, which concerns the preparation of quantum states that have only d non-zero amplitudes, where d is ideally smaller than the maximum sparsity N.

Just as in [Neu25], we observe an exponentially higher success probability for the adaptive approach compared to the non-adaptive approach, *conditioned* on the success probability of a CNOT-gate being at least the probability that a qubit that is idling during $\gamma(n)$ CNOT-gates does not introduce an error, where $\gamma(n)$ is a function depending on the circuit and on the number of qubits the circuit acts on. Using reported gate error and idling error probabilities for current quantum hardware, we will make an estimate of γ for these devices. This allows us to get an intuition when we can expect this adaptive advantage to materialize: if the observed γ exceeds the γ of one of the QPUs, we will say there is a 'possible adaptive advantage'.

Table 1.1 summarizes our results. Our results indicate that adaptive circuits offer an advantage primarily when qubit connectivity is limited to a 2D grid, with increasing advantage as the number of qubits grows. Especially for preparing dense states, the advantage seems very clear. Compared to non-adaptive circuits that assume all-to-all connectivity, we see no adaptive advantage for preparing quantum states of less than 1000 qubits, with the exception of the UCG-based QSP method using the parallelized UCG implementation.

possible adaptive advantage over...	**all-to-all connectivity**					**2D grid connectivity**				
	$d = 10$	$d = n$	$d = n^2$	$d = n^3$	$d = 2^n$ (dense)	$d = 10$	$d = n$	$d = n^2$	$d = n^3$	$d = 2^n$ (dense)
UCG-based QSP (sequential)	-	-	-	-	-	$n \gtrsim 700$	$n \gtrsim 700$	$n \gtrsim 600$	$n \gtrsim 90$	$n \gtrsim 700$
UCG-based QSP (parallelized)					$n \gtrsim 600$					$n \gtrsim 10$
unary-based QSP					-					$n \gtrsim 17$

Table 1.1: Summary of our results. The cells contain lower bounds on the number of qubits n needed to see a possible advantage for an adaptive circuit over a non-adaptive circuit, when performing (sparse) quantum state preparation. The adaptive circuits we considered are constrained to a grid topology, and are compared against non-adaptive circuits with either all-to-all connectivity or 2D grid connectivity. A '-' indicates that we expect no adaptive advantage for preparing quantum states of ≤ 1000 qubits.

1.3 Outline

After this introduction, we start in Chapter 2 with a review of some preliminary knowledge on quantum computing and computational complexity. We will then discuss implementations of multi-qubit gates that play a role in later QSP algorithms, such as the aforementioned Fanout-gate and OR-gate, the AND-gate, the Permutation-gate. We will also shortly introduce the concept of uniformly controlled gates. We will see that the efficiency of any of these multi-qubit gate implementations strongly depends on the assumed qubit connectivity.

In Chapter 3, we introduce the concept of adaptive quantum circuits through their relation with quantum teleportation. We then describe a formalization of these, the LAQCC model [Buh+23], and discuss how it relates to other such formalizations. We then show that every Clifford circuit is accessible in a specific instance of this model. Next, we review the main constructions providing a speed-up in this setting, namely the GHZ-state preparation scheme, and the Fanout-gate. The chapter is concluded by discussing how the LAQCC class instance relates to other classical and quantum circuit complexity classes.

In Chapter 4, we formally define quantum state preparation and discuss the circuits for both UCG-based and unary-based algorithms, and the variants catered to sparse states. For both variants, we obtain an asymptotic expression for the circuit depth and circuit width of these circuits, both in the adaptive and non-adaptive case.

In Chapter 5, we compute the success probability of the (sparse) QSP algorithms, using the error model introduced in [Neu25]. We obtain expressions for the conditions under which the adaptive circuit performs better than its non-adaptive counterparts assuming either all-to-all qubit connectivity or a 2D grid topology.

In Chapter 6, we conclude and sketch directions for future research.

Chapter 2

Preliminaries

This chapter functions as a brief introduction to the material discussed in this thesis. In Section 2.1, we fix some notation and definitions. In Section 2.2, 2.3, and 2.4, we will introduce the field of quantum computing, and define its most common model of computation: the quantum circuit model. Section 2.5 defines terms describing circuit complexity – quantum *and* classical. Some multi-qubit gates, such as the Fanout-gate, OR-gate, and the class of uniformly controlled gates, play a pivotal role in this thesis; they are discussed in Section 2.7.

2.1 Fixing some notation and definitions

In this thesis, we typically start counting from 0. Given any natural number n, we use $[n]$ to denote the set $\{0, \ldots, n-1\}$. We use $\log(\cdot)$ to denote the base-2 logarithm $\log_2(\cdot)$, and $\oplus$ to denote addition modulo 2. Capital letters such as N usually refer to powers of 2, where the exponent is denoted by the respective lower case n. Given two n-bit strings x, y, we define x_i to be the i-th element of x, where $i \in [n]$, and let $x \oplus y$ denote the string z of length n such that $z_i = x_i \oplus y_i$. For a string x, set $\bar{x} := x \oplus 1^n$, where 1^n is the n-bit string consisting only of ones. We use $|x|$ to denote the Hamming weight of x. The bit-wise inner product between x and y is defined as $x \cdot y := \sum_{i \in [n]} x_i y_i \mod 2$. We use $e_i^{(n)}$ to denote an n-bit string in which all bits are zero except for the i-th bit (a 'one-hot' encoding). For any predicate p, the *indicator function* $\mathbb{1}_p$ is a variable that takes on value 1 if the predicate is true, and 0 otherwise. For example, $\mathbb{1}_{x>y}$ is 1 iff $x > y$.

If $z = a + bi$ is a complex number, then $z* := a - bi$ denotes its *complex conjugate*. The *norm* or *amplitude* of a complex number, denoted $|z|$, is defined as $|z| := \sqrt{zz*}$. Any complex number z can be written as $z = re^{i\theta}$, where $r = |z|$ and $\theta \in [0, 2\pi)$ is the *phase* of z, denoted $\arg(z)$. We will also write $\exp(x) := e^{ix}$. The norm of an n-dimensional vector v is defined as $\|v\| := \sqrt{|v_0|^2 + |v_1|^2 + \cdots + |x_{n-1}|^2}$. The inner product of two n-dimensional complex vectors x and y is defined as $\langle x, y \rangle := \sum_{i \in [n]} x_i^* y_i$. Given a matrix M, we use $\mathsf{M}^\dagger$ to denote its conjugate transpose.

A classical circuit is a finite, directed, acyclic graph consisting of AND, OR and NOT-gates. The single-bit NOT-gate is defined as $\mathrm{NOT}(x_i) = \bar{x}_i$. They are defined as follows:

$$\mathrm{NOT}(x) = \bar{x} \qquad \mathrm{OR}^{(n)}(x) = \mathbb{1}_{|x|>0} \qquad \mathrm{AND}^{(n)}(x) = \mathbb{1}_{|x|=n}. \tag{2.1}$$

Another multi-qubit gate is the Parity-gate PA, defined as $\mathrm{PA}^{(n)}(x) = \mathbb{1}_{x \cdot 1^n = 1}$. A circuit has n input nodes, containing the n input bits. The internal nodes consist of gates, and there are some designated output nodes, that eventually take on some value. The *fan-in* of a gate is defined as the number of input bits to that gate.

Lastly, a permutation on 2^n elements is a bijective function $\sigma : \{0,1\}^n \to \{0,1\}^n$. The *size* of σ, denoted $|\sigma|$, is defined as $|\sigma| := |\{x \in \{0,1\}^n \mid \sigma(x) \neq x\}|$. The set of all permutation on m elements is denoted S_m.

2.2 Quantum computing

We will now give a concise overview of quantum computing. For a more thorough introduction, we refer to the book by Nielsen and Chuang [NC10] or the lecture notes by de Wolf [Wol23], on which the following sections are based.

Consider some physical system that can be in N different, mutually exclusive classical states. For example, a transistor manipulates two states: 0 and 1. We will call these N states $|0\rangle, |1\rangle, \ldots, |N-1\rangle$. Roughly, by a 'classical' state we mean a state in which the system can be found when we observe it. Although intuitively this might seem true for any state in any system, we do not presume this property for quantum states. Rather, a *(pure) quantum state* (often just called *state*) $|\psi\rangle$ is a *superposition* of classical states, written

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_{N-1} |N-1\rangle . \tag{2.2}$$

Here, α_i is a complex number that is called the *amplitude* of $|i\rangle$ in $|\psi\rangle$. Mathematically, the states $|0\rangle, \ldots, |N-1\rangle$ form an orthonormal basis of an N-dimensional Hilbert space, i.e., an N-dimensional vector space equipped with an inner product. A quantum state $|\psi\rangle$ is a unit vector in this space which can be written as a N-dimensional column vector of its amplitudes:

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}, \tag{2.3}$$

Such a vector $|\psi\rangle$ is called a *ket*. Its conjugate transpose its called a *bra*, and corresponds to the row vector

$$\langle\psi| = \begin{bmatrix} \alpha_0^* & \alpha_1^* & \ldots & \alpha_{N-1}^* \end{bmatrix} . \tag{2.4}$$

Thus $\langle\psi| = |\psi\rangle^\dagger$. This notation is called *Dirac notation*, after Paul Dirac, and is the standard notation used in quantum mechanics. Among other things, it allows for a convenient representation of the inner product as $\langle\psi| \cdot |\varphi\rangle = \langle\psi|\varphi\rangle$ – a *braket* –, and the outer product as $|\psi\rangle \langle\varphi|$.

In classical computing, the smallest system used is a 2-level system, called a *bit*. The analogous notion for quantum computing is aptly named *qubit* (or *quantum bit*). Using the notation above, we can write a qubit as a 2-dimensional vector ψ with

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}, \tag{2.5}$$

with $\alpha_0, \alpha_1 \in \mathbb{C}$ and $|\alpha_0|^2 + |\alpha_1|^2 = 1$. In Equation (2.3) and (2.5), we have explicitly picked a basis in which the vector is represented, namely

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{2.6}$$

This basis is called the *computational basis*. Upon observing or *measuring* a quantum system, the quantum state *collapses* to one of its classical states. In the case of measuring a qubit in the computational basis, it will collapse to either $|0\rangle$ or $|1\rangle$, with probability $|\alpha_0|^2$ and $|\alpha_1|^2$ respectively. This concept, known as *Born's rule*, is one of the fundamental postulates of quantum mechanics.

Commonly, quantum algorithms possess computations involving many qubits. We can create quantum states with more qubits by considering their *tensor product*, which for two states $|\psi\rangle$ and $|\varphi\rangle$ is denoted $|\psi\rangle \otimes |\varphi\rangle$.[1] For the n-fold tensor product $|\psi\rangle \otimes \cdots \otimes |\psi\rangle$ we use the short-hand $|\psi\rangle^{\otimes n}$. In vector form, the tensor of two qubits can be written as

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \otimes \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix} \tag{2.7}$$

[1] Alternatively, we will write $|\psi\rangle |\varphi\rangle$, $|\psi, \varphi\rangle$ or $|\psi\varphi\rangle$, where the tensor product is understood.

Name	Number of qubits	$\|\text{ket}\rangle$
GHZ-state	$n \in \mathbb{N},\ n \geq 2$	$\|\text{GHZ}^{(n)}\rangle = \frac{1}{\sqrt{2}}(\|0\rangle^{\otimes n} + \|1\rangle^{\otimes n})$
W-state	$n \in \mathbb{N}$	$\|\text{W}^{(n)}\rangle = \frac{1}{\sqrt{n}} \sum_{i\in[n]} \|e_i\rangle$
Dicke-state	$n, k \in \mathbb{N},\ k \leq n$	$\|\text{D}_k^{(n)}\rangle = \frac{1}{\sqrt{\binom{n}{k}}} \sum_{\substack{x\in\{0,1\}^n \\ \|x\|=k}} \|x\rangle$

Table 2.1: Some well-known quantum states.

For a single qubit, there are two computational basis states. For multiple qubits, the computational basis states are all possible combinations of single-qubit basis states. That means that for two qubits, there are four basis states, and for general n, there are 2^n. Depending on the context, we will either represent these using their binary expansion (e.g. $|00101\rangle$) or as a decimal integer (e.g. $|5\rangle$). Two n-qubit basis states satisfy $\langle x|y\rangle = \mathbb{1}_{x=y}$, a fact that follows from the properties of the tensor product and the fact that $|0\rangle$ and $|1\rangle$ are orthonormal.

A central phenomenon in quantum mechanics is *entanglement*, which occurs when groups of particles exhibit behaviour, such that the quantum state of individual particles cannot be described independently of the state of the other particles in the group. Mathematically, an n-qubit entangled state $|\psi\rangle$ is a multi-qubit state that cannot be written as a tensor product of smaller states, i.e., $|\psi\rangle \neq |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$. The simplest such states are the *Bell-states*:

$$|\Psi_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Psi_{01}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{2.8}$$

$$|\Psi_{10}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad |\Psi_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{2.9}$$

For $a, b \in \{0, 1\}$, we have $|\Psi_{ab}\rangle = \mathsf{X}^a\mathsf{Z}^b\,|\Psi_{00}\rangle$. The Bell-states form an orthonormal basis of the space of all 2-qubit states, the *Bell basis*. Measuring in this basis gives outcome $ab \in \{0, 1\}^2$ and lets the state collapse to $|\Psi_{ab}\rangle$, or equivalently, projects the two qubits on

$$\sum_{i\in\{0,1\}} \langle ii| \left(\mathsf{Z}^a\mathsf{X}^b \otimes \mathsf{I}\right). \tag{2.10}$$

Table 2.1 lists some other well-known and -used entangled quantum states.

2.3 Quantum operations

In quantum mechanics, the Schrödinger equation implies that only linear operations can be applied to quantum states. Since quantum states are nothing more (or less) than complex unit vectors, matrices can be used to operate on them. To make sure that the outcome of an operation on an n-qubit quantum state is still a valid n-qubit quantum state, these matrices must be *square* and *norm-preserving*, i.e. map unit vectors to unit vectors of the same dimension. It is easy to show that *unitary* matrices, for which $\mathsf{U}\mathsf{U}^\dagger = \mathsf{I}$, form the class of matrices that satisfy these constraints. The restriction of quantum operations to unitaries also means that some operations cannot be performed on a quantum computer. For example, by the No-Cloning Theorem, there is no quantum operation that copies an arbitrary qubit, i.e., that maps $|\psi\rangle\,|0\rangle \to |\psi\rangle\,|\psi\rangle$. Unitaries are often called *gates*, in analogy to classical logic gates like AND, OR, and NOT. The set of all n-qubit gates – that is, the set of all $2^n \times 2^n$-dimensional unitaries – is denoted $\mathcal{U}(n)$. The *arity* of a gate is the number of qubits it acts upon, which we may emphasize using a superscript when necessary: $\mathsf{U}^{(n)}$. Two gates $\mathsf{U}^{(n)}$ and $\mathsf{V}^{(m)}$ can be composed using the tensor product $\mathsf{U} \otimes \mathsf{V}$ to become a gate with arity $n + m$.

We will now list some elementary quantum gates used in quantum computing. They are used on their own, and also serve as building blocks for some of the more elaborate gates later on in this thesis. Perhaps the most important unitary is the Hadamard-gate

$$\mathsf{H} := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{2.11}$$

which can be used to create uniform superpositions:

$$\mathsf{H}\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) := \left|+\right\rangle, \qquad \mathsf{H}\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) := \left|-\right\rangle. \tag{2.12}$$

For n qubits, applying $\mathsf{H}^{\otimes n}$ to $\left|0\right\rangle^{\otimes n}$ yields

$$\mathsf{H}^{\otimes n}\left|0\right\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} \left|x\right\rangle = \frac{1}{\sqrt{N}} \sum_{i\in[N]} \left|i\right\rangle, \tag{2.13}$$

which is the uniform superposition over all n-bit strings. On computational basis state $\left|i\right\rangle$, it acts as follows:

$$\mathsf{H}^{\otimes n}\left|i\right\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{i\cdot j} \left|j\right\rangle, \tag{2.14}$$

where $i \cdot j$ is the bit-wise inner product between i and j (see Section 2.1). The Identity-gate I

$$\mathsf{I} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.15}$$

leaves the state unaltered. Other often-used gates include the Pauli matrices

$$\mathsf{X} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \mathsf{Y} := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \mathsf{Z} := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.16}$$

On the (computational basis) states $\left|0\right\rangle$ and $\left|1\right\rangle$, X acts as a quantum version of a NOT-gate, mapping $\left|0\right\rangle$ to $\left|1\right\rangle$ and vice-versa, while Z adds a phase of -1 to $\left|1\right\rangle$. The gate Y, which can be written as $\mathsf{Y} = i\mathsf{XZ}$, combines the two gates while adding a global phase of i. Together, the Pauli matrices generate a group P – the *Pauli group* – that spans the space of all one-qubit unitaries. For one qubit, it consists of all Pauli matrices together with the factors ± 1 and $\pm i$:

$$P := \{\pm\mathsf{I}, \pm i\mathsf{I}, \pm\mathsf{X}, \pm i\mathsf{X}, \pm\mathsf{Y}, \pm i\mathsf{Y}, \pm\mathsf{Z}, \pm i\mathsf{Z}\}. \tag{2.17}$$

The n-qubit Pauli group P_n is the set of all n-fold tensor products of matrices from P, with a global phase of ± 1 or $\pm i$.

The non-identity Pauli gates have a rotational version, defined for any angle $\theta \in [0, 2\pi)$:

$$\mathsf{R}_\mathsf{X}(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad \mathsf{R}_\mathsf{Y}(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \quad \mathsf{R}_\mathsf{Z}(\theta) := \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{2.18}$$

Note that the Pauli gates correspond (up to a global phase) to their rotational counterparts if $\theta = \pi$. Other rotation gates that go by their own names are the gates $\mathsf{S} := \mathsf{R}_\mathsf{Z}(\pi/4)$, $\mathsf{T} := \mathsf{R}_\mathsf{Z}(\pi/8)$, and

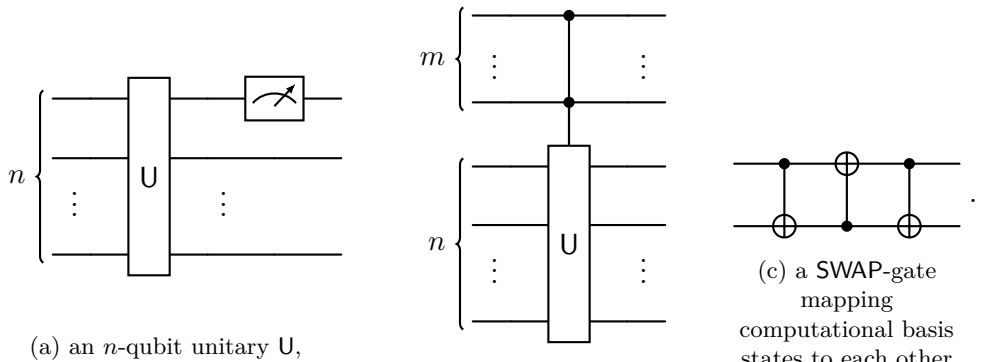
$$\sqrt{\mathsf{X}} := \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \propto \mathsf{R}_\mathsf{X}(\pi/2). \tag{2.19}$$

As for two-qubit unitaries, the most common one is the controlled X-gate (or CNOT-gate). It is defined as

$$\mathsf{CNOT} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.20}$$

which acts on two qubits by flipping the second qubit if and only if the first qubit is $\left|1\right\rangle$. Generally, an n-qubit unitary U that is controlled by m qubits defines the $(m+n)$-qubit controlled unitary CU:

$$\mathsf{CU} = \begin{bmatrix} \mathsf{I}^{\otimes m} & 0 \\ 0 & \mathsf{U} \end{bmatrix}, \tag{2.21}$$



(a) an n-qubit unitary U, followed by a measurement of the first qubit.

(b) an n-qubit unitary U controlled by m other qubits.

(c) a SWAP-gate mapping computational basis states to each other (e.g. $|a,b\rangle \to |b,a\rangle$).

Figure 2.1: Some gate visualizations in the quantum circuit model.

where we note that $\mathsf{I}^{\otimes m}$ is the Identity-gate on m qubits. The set of gates $C := \{\mathsf{CNOT}, \mathsf{H}, \mathsf{S}\}$ generate the n-qubit *Clifford group* C_n. This is a well-known and -studied group, for two main reasons. Firstly, the class of n-qubit circuits that are constructed using only elements of this group correspond precisely to the set of quantum operations that *normalize* the n-qubit Pauli group. That is, we have $\mathsf{U} \in C_n$ iff for every $\mathsf{P} \in P_n$ we have $\mathsf{UPU}^\dagger \in P_n$. For example, the commutation relations of the CNOT-gate and the Pauli gates are given in Table 2.2. The second reason is that any computation using Clifford gates can be efficiently simulated on a classical computer, a result known in literature as the Gottesman-Knill Theorem [Got98].

P	$\mathsf{P}' = \mathsf{CNOT} \cdot \mathsf{P} \cdot \mathsf{CNOT}$
$\mathsf{X} \otimes \mathsf{I}$	$\mathsf{X} \otimes \mathsf{X}$
$\mathsf{I} \otimes \mathsf{X}$	$\mathsf{I} \otimes \mathsf{X}$
$\mathsf{Z} \otimes \mathsf{I}$	$\mathsf{Z} \otimes \mathsf{I}$
$\mathsf{I} \otimes \mathsf{Z}$	$\mathsf{Z} \otimes \mathsf{Z}$

Table 2.2: Commutation relations of the CNOT-gate and Pauli gates.

2.4 Quantum circuits

The discussion in Section 2.3 naturally leads to a model of quantum computation. Suppose we are given $n+m$ qubits such that the first n qubits (*the working register*) are in the state $|\psi\rangle$ (the *input state*), and the last m qubits (the *ancilla register*) are in the state $|0\rangle^{\otimes m}$. Then we can apply any number of quantum gates to (some of) the qubits, with any amount of measurements of (some of) the qubits in between. This results in an n-qubit *output state* $|\psi'\rangle$, which can either be left as is or acted upon by measuring a subset of its qubits and returning the result. Operations can be applied to the ancilla register too, however, by the end of the circuit they need either to be measured or to be restored to the all-zeroes state. These configurations of quantum operations and measurements are called *quantum circuits*, and thus this model of computation is called the *quantum circuit model*.

Figure 2.1 shows visual examples of some simple quantum circuits. If the unitary that is controlled is an X-gate, we use the symbol $\oplus$ instead of X. For example, a circuit implementing the SWAP-gate (which interchanges two qubits) is pictured in Figure 2.1c. For clarity of presentation, we will sometimes label some of the qubits, or group them together into *registers*. As an example, we can group the m control qubits in Figure 2.1b into register $\mathtt{C}$, and the n target qubits into register $\mathtt{T}$; this is denoted $|0\rangle_{\mathtt{C}} |0\rangle_{\mathtt{T}}$ We denote a unitary controlled by register $\mathtt{A}$ and applied to another register $\mathtt{B}$ as $\mathsf{C}_{\mathtt{A}}\mathsf{U}_{\to \mathtt{B}}$, whenever we want to explicitly specify this.

In the classical circuit model, the AND and NOT-gate serve as a *universal gate set*: a set of gates from which

any classical circuit can be constructed. This immediately begs the question: can we find a universal gate set in the quantum circuit model, that is, a set of unitaries from which every unitary can be implemented?

Since there are uncountably many unitary operators, a simple counting argument shows that defining such universal gate set requires uncountably many unitary operators. The simplest example of this is the set of all single-qubit gates together with the CNOT-gate [NC10]. However, since we can not expect that whoever is building a quantum computer will be able to implement every single-qubit gate to infinite precision, it may be a bit unreasonable to assume access to such a gate set. As such, our second-best option is to find a (finite) set of gates that *approximate* all unitaries well enough. Fortunately, given a finite set of gates such as $\{\mathsf{CNOT}, \mathsf{H}, \mathsf{S}, \mathsf{T}\}$ (Clifford+T), or another finite universal gate set, we can approximate other quantum circuits arbitrarily well with only little overhead. In particular, a quantum circuit consisting of k gates of constant arity can be approximated to error ε using a quantum circuit with $\mathcal{O}(k \log_c(k/\varepsilon))$ gates of such gate set. This result is known as the Solovay-Kitaev Theorem [NC10].

2.5 Circuit complexity classes

For our purposes, we will consider the set of all single-qubit unitary operations, together with the CNOT-gate and measurements in the computational basis as our elementary operations, unless specified otherwise. Relative to this set of operations, we will now define some complexity notions concerning decision problems and their quantum circuits.

Definition 1. *Let Q be a quantum circuit. We define*

- *the depth of Q, denoted depth(Q), as the number of layers in a circuit, where each layer consists of gates that act on disjoint sets of qubits.*
- *the width of Q as the total number of qubits involved in the circuit.*
- *the size of Q, denoted size(Q), as size(Q) = depth(Q) · width(Q).*

If quantum computers have the ability to perform gates on disjoint sets of qubits simultaneously – which we will assume –, circuit depth can be used as a proxy for the computation time of running the quantum circuit. The following notation will be useful when expressing complexities.

Definition 2 (*Bachmann-Landau* (or *'big-Oh'*) notation)**.** *Let $f, g : \mathbb{N} \to \mathbb{R}$ be non-decreasing. We say that $g(n)$ is in the set*

- *$\mathcal{O}(f(n))$ if there exists a $c \in \mathbb{R}_{\geq 0}$ and $n_0 \in \mathbb{N}$ such that $0 \leq g(n) \leq c \cdot f(n)$ for all $n \geq n_0$;*
- *$o(f(n))$, if for all $c \in \mathbb{R}_{\geq 0}$ there exists $n_0 \in \mathbb{N}$ such that $0 \leq g(n) < c \cdot f(n)$ for all $n \geq n_0$;*
- *$\Omega(f(n))$, if there exists a $c \in \mathbb{R}_{\geq 0}$ and $n_0 \in \mathbb{N}$ such that $g(n) \geq c \cdot f(n)$ for all $n \geq n_0$;*
- *$\omega(f(n))$, if for all $c \in \mathbb{R}_{\geq 0}$ there exists $n_0 \in \mathbb{N}$ such that $g(n) > c \cdot f(n)$ for all $n \geq n_0$; and*
- *$\Theta(f(n))$ is both in $\mathcal{O}(f(n))$ and $\Omega(f(n))$.*

A circuit family $\{C_n\}_{n\in\mathbb{N}}$ (where C_n accepts inputs of size n) has *polynomial size* if its size can be bounded by a polynomial, i.e. size(C_n) = $\mathcal{O}(n^c)$ for some constant c. The circuit class $\mathsf{QPoly}(n)$ consists of all polynomial-size (in n) quantum circuit families that use one- and two-qubit gates.

We now define some classical and quantum complexity classes that will appear in later chapters. Usually, these are defined as classes of decision problems solvable by some kind of circuit. In the following definitions, having access to *bounded-fan-in* OR-gates means that we can implement an $\mathrm{OR}^{(n)}$-gate for n up to some fixed constant c. Having *unbounded-fan-in* OR-gates means that $\mathrm{OR}^{(n)}$ can be implemented for any n.

Definition 3 (Complexity classes)**.** *A decision problem is in the class*

- NC^i *if it is solvable by a circuit family of polynomial size,* $\mathcal{O}((\log n)^i)$ *depth, and consisting of bounded-fan-in OR-gates, and NOT-gates;*
- AC^i *if it is solvable by a circuit family of polynomial size,* $\mathcal{O}((\log n)^i)$ *depth, and consisting of unbounded-fan-in OR-gates, and NOT-gates;*
- TC^i *if it is solvable by a circuit family of polynomial size,* $\mathcal{O}((\log n)^i)$ *depth, and consisting of unbounded-fan-in OR-gates, NOT-gates, and Threshold*$_i$*-gates*[2]*.*
- QNC^i *if it is solvable by a quantum circuit that is in* $\mathsf{QPoly}(n)$*, and has* $\mathcal{O}((\log n)^i)$ *depth.*

Lemma 1. $\mathsf{NC}^1 \subseteq \mathsf{QNC}^1$.

Proof. For fixed input size n, let C be the Boolean circuit of depth $\mathcal{O}(\log n)$, containing OR- and NOT-gates of bounded fan-in, that decides on a decision problem in NC^1. Using ancilla qubits, we can replace all gates with their quantum equivalents: after substituting the OR-gate for AND- and NOT-gates, we replace the AND-gates with Toffoli-gates and the NOT-gates with X-gates, giving us a quantum circuit. The outputs of the AND-gates are stored in fresh ancilla qubits, and can be used as input qubits further in the circuit. In total, there are polynomially many quantum gates and ancilla qubits, since there were polynomially many gates in the classical circuit. The depth of the new quantum circuit remains logarithmic, hence we can conclude that the decision problem is in QNC^1. □

The logarithmic-depth circuits in the class NC^1 also allows us to implement a Parity-gate.

Lemma 2. *[AB09, Example 6.26] For any* n*,* $PA^{(n)} \in \mathsf{NC}^1$.

In the remainder of this thesis, we will slightly abuse notation and say that a circuit is an X-circuit if that circuit corresponds to a decision problem in the class X.

2.6 Qubit topology

On most current quantum hardware, gates cannot be applied between arbitrary pairs of qubits. Rather, qubits are arranged in a specific topology that describes how they can interact with each other. For example, qubits can be arranged in a 2D grid pattern, where each qubit can only interact with its nearest (vertical or horizontal) neighbor. See Figure 2.2. Arbitrary qubit connectivity can be naturally modelled using a

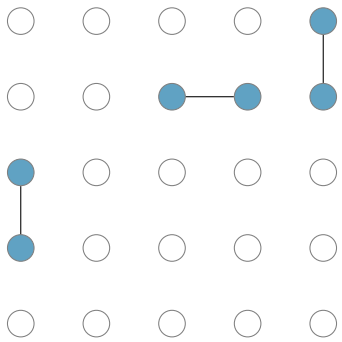


Figure 2.2: Qubits arranged in a 2D grid. Two blue-colored, connected qubits indicate that a gate is applied on those qubits.

connected *constraint graph* $G = (V, E)$, where the vertices V represent the qubits and the edges E specify the pairs of qubits on which two-qubit gates can act. If $E = V \times V$, then all qubits are connected to each other. We refer to this as *all-to-all* qubit connectivity.

In general, to apply a gate to an arbitrary pair of qubits under some constraint constraint, a simple method is to insert SWAP-gates to bring the qubits together, apply the gates, and swapping them back. The *depth*

[2] A Threshold$_k$-gate outputs one iff the sum of the inputs is at least k

overhead that this induces can be defined, for some circuit C, as

$$\mathrm{doh}(G, C) := \min_{C'} \frac{\mathrm{depth}(C')}{\mathrm{depth}(C)}, \tag{2.22}$$

where the minimum is over all C' that are equivalent to C and that respect the constraint graph G. The depth overhead of a constraint graph G is defined as the maximum of the depth overhead over all possible circuits C, that is, it characterizes the maximum extra depth factor that a certain constraint graph induces. By a very recent result, this depth overhead is fully characterized by the *routing number* of the constraint graph.

Theorem 1 (Theorem 3 and 4 from [YZ24])**.** *Let G be any constraint graph. Then it holds that*

$$doh(G) = \max_{C} doh(G, C) = \Theta(rt(G)). \tag{2.23}$$

This routing number $\mathrm{rt}(G)$ of a graph G is a well-studied complexity measure, and is defined by the following game[ACG94; YZ24]. Given a connected graph G with n vertices, a pebble is placed at each vertex $v \in V$. The pebbles are then moved in rounds. In each round, a matching $M_i \subseteq E$ is selected, and the pebbles are swapped between u and v for every $(u, v) \in M_i$. Given some permutation σ of the vertices, the routing number of σ, denoted $\mathrm{rt}(G, \sigma)$, is the *minimum* number of rounds needed to move all pebbles according to the permutation σ, that is, from initial position v to $\sigma(v)$. For a graph G, the routing number $\mathrm{rt}(G)$ is defined as the maximum routing numbers over all permutations of its vertices:

$$\mathrm{rt}(G) = \max_{\sigma \in S_n} \mathrm{rt}(G, \sigma). \tag{2.24}$$

2.7 Multi-qubit gate implementations

In this section, we show the quantum circuits that implement some commonly used multi-qubit gates and their complexity.

2.7.1 Controlled gates

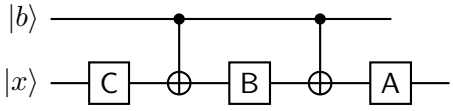
Firstly, using single-qubit and CNOT-gates, the following lemma shows that we can make any single-qubit unitary controlled.

Lemma 3. *Let* U *be an arbitrary single-qubit unitary. We can implement* CU*, that acts as*

$$\mathsf{CU}\,|b\rangle\,|x\rangle \to \begin{cases} |b\rangle\,|x\rangle & \text{if } b = 0 \\ |b\rangle\,(\mathsf{U}\,|x\rangle) & \text{if } b = 1 \end{cases}, \tag{2.25}$$

using only single-qubit and CNOT*-gates.*

Proof. In [NC10, Corollary 4.2] it is shown that for every single-qubit unitary U there exist one-qubit gates A, B and C such that ABC = I and AXBXC = U. Then it is easy to see that the circuit



implements Operation (2.25). □

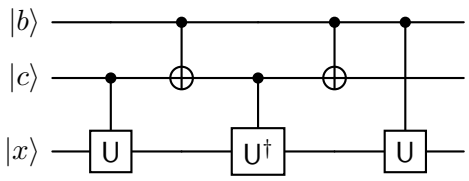
For single-qubit unitaries controlled by two qubits, we have another result.

Lemma 4. *Let* U *be an arbitrary one-qubit unitary. We can implement* $\mathsf{CC}(\mathsf{U}^2)$*, that acts as*

$$\mathsf{CCU}^2 |b\rangle |c\rangle |x\rangle \to \begin{cases} |b\rangle |c\rangle |x\rangle & \text{if } b \wedge c = 0 \\ |b\rangle |c\rangle (\mathsf{U}^2 |x\rangle) & \text{if } b \wedge c = 1 \end{cases}, \tag{2.26}$$

using only single-qubit and CNOT*-gates.*

Proof. It can be easily verified that the following circuit corresponds to the desired transformation:



By the previous lemma, the controlled U and $\mathsf{U}^\dagger$-gates can be implemented using only single-qubit and CNOT-gates, and so the result follows. □

2.7.2 Fanout-gate

In this thesis, a large focus will be on the implementation of multi-qubit gates using low-depth (or *shallow*) circuits. To construct such circuits, it is often important to copy ('fan-out') one of the inputs into multiple copies. In classical circuits, we mostly assume that we get this operation 'for free', by simply splitting the input wires as many times as we like. In quantum circuits, this is difficult, since copying arbitrary quantum states is forbidden by the No-cloning Theorem. However, we can copy *classical* information to multiple quantum states. The n-qubit quantum Fanout-gate (denoted $\mathsf{FO}^{(n)}$) is the operation that does this, i.e. it acts as

$$\mathsf{FO}^{(n)} |b\rangle |x_1\rangle \dots |x_{n-1}\rangle := |b\rangle |x_1 \oplus b\rangle \dots |x_{n-1} \oplus b\rangle . \tag{2.27}$$

This seemingly benign operation turns out to be pretty powerful. For starters, it allows us to implement the Parity-gate $\mathsf{PA}^{(n)}$, which acts as

$$\mathsf{PA}^{(n)} |x_0 \dots x_{n-1}\rangle |b\rangle := |x_0 \dots x_{n-1}\rangle \left|b \oplus \bigoplus_{i=0}^{n-1} x_i\right\rangle , \tag{2.28}$$

by noting that

$$\mathsf{PA}^{(n)} = \mathsf{H}^{\otimes n} \mathsf{FO}^{(n)} \mathsf{H}^{\otimes n}. \tag{2.29}$$

This duality has no classical counterpart [GM24; Ajt83]. Furthermore, Figure 2.3 shows how the $|\mathrm{GHZ}^{(5)}\rangle$-state can be prepared using two Fanout-gates. The Fanout-gate entangles many qubits at once, allowing

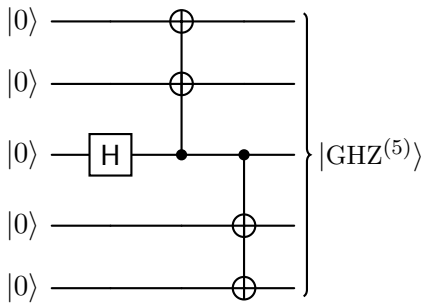


Figure 2.3: Circuit to prepare the state $|\mathrm{GHZ}^{(5)}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes 5} + |1\rangle^{\otimes 5})$, using two $\mathsf{FO}^{(3)}$-gates.

multiple operations to be performed on each of them separately at the same time. The first such parallelization techniques were constructed in [MN98a; MN98b; Moo99]. They also defined circuit classes in which Fanout-gates is assumed to be an elementary operation, i.e., where Fanout-gates are counted as a single gate,

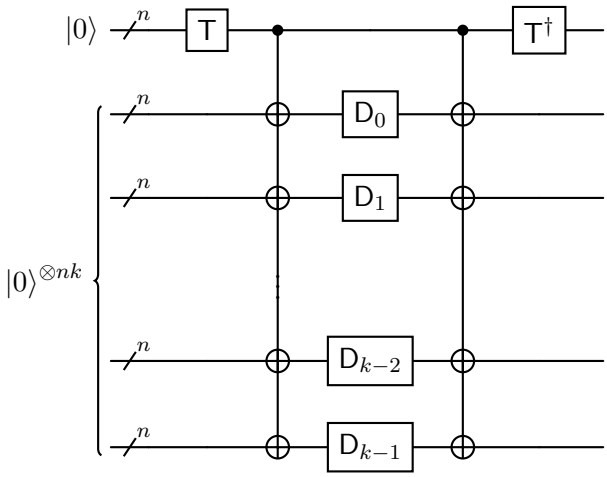


Figure 2.4: This circuit shows the parallelization construction using Fanout-gates. The diagonal unitaries can be made controlled (not shown here; see [Hv05] for the full circuit).

just like a single-qubit gate or a CNOT-gate. The class of decision problems solvable by polynomial-size, constant-depth quantum circuits with Fanout-gates is called QNC^0_f.

We will now review one of the parallelization constructions.

Lemma 5 (Theorem 1.3.19 from [HJ17])**.** *For every set of pairwise commuting unitary gates, there exists an orthonormal basis in which all these gates are diagonal.*

Theorem 2 (Proposition 3 from [MN98a])**.** *Let $\{\mathsf{U}_i\}_{i\in[k]}$ be a pairwise commuting set of gates on n qubits, and let T be a unitary that mutually diagonalizes all U_i. Let $\mathtt{a_i}$ be the register of the qubit controlling U_i. Then there exists a quantum circuit, using Fanout-gates, that computes $\mathsf{U} = \prod_{i\in[k]} \mathsf{C}_{\mathtt{a_i}}\mathsf{U}_i$. It has depth $\max_{i\in[k]} depth(\mathsf{U}_i) + 4 \cdot depth(\mathsf{T}) + 2$, size $\sum_{i\in[k]} size(\mathsf{U}_i) + (2k+2) \cdot size(\mathsf{T}) + 2k$, and uses $(k-1)n$ ancilla qubits.*

In particular, if all commuting gates can be implemented in constant depth, then the circuit is in QNC^0_f.

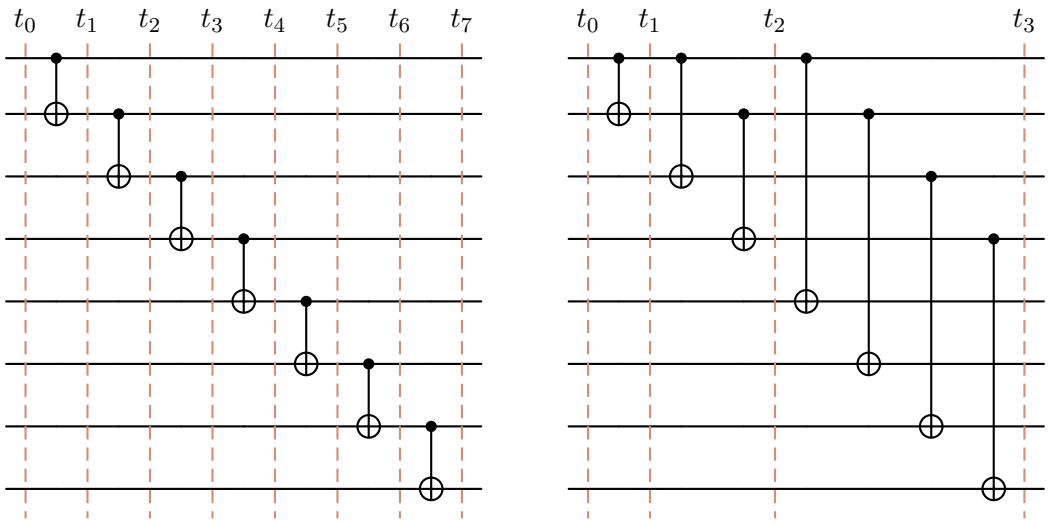
Proof. Consider a quantum circuit that applies all $\mathsf{C}_{\mathtt{a_i}}\mathsf{U}_i$ sequentially. Let $\mathsf{U}_i := \mathsf{T}^\dagger\mathsf{D}_i\mathsf{T}$ for every i, where $\{\mathsf{D}_i\}_{i\in[k]}$ are the simultaneously diagonalized matrices guaranteed to exist by Lemma 5, and the columns of T are the orthonormal basis vectors of this diagonalizing basis. Then since $\mathsf{T}\mathsf{T}^\dagger = \mathsf{I}$, we have that

$$\prod_{i\in[k]} \mathsf{C}_{\mathtt{a_i}}\mathsf{U}_i = \mathsf{T}\left(\prod_{i\in[k]} \mathsf{C}_{\mathtt{a_i}}\mathsf{D}_i\right)\mathsf{T}^\dagger. \tag{2.30}$$

Since the D-matrices are diagonal, they merely impose some phase shifts on the qubits. Since multiple phase shifts on entangled qubits multiply, all D_i can be applied in parallel. To do this, we first use a total of n $\mathsf{FO}^{(k)}$-gates to create k entangled copies of the n target qubits; we then apply gate D_i to the i-th entangled copy, and then use another n $\mathsf{FO}^{(k)}$-gates to uncompute the copies. See the circuit in Figure 2.4. □

Implementations of Fanout-gates

Figure 2.5a shows a simple sequential implementation of a Fanout-gate of arity 8, using a ladder of CNOT-gates. For arbitrary n, this yields a circuit of depth $\mathcal{O}(n)$ and size $\mathcal{O}(n^2)$. If the underlying qubit topology is a 1D grid, this circuit is optimal. When assuming all-to-all connectivity, we can reduce this depth by parallelizing some of the CNOT-gates. For arbitrary n, this circuit consists of $\lceil\log(n)\rceil$ layers. The i-th layer (for $0 \leq i \leq \lceil\log n\rceil - 2$) consists of 2^i parallel CNOT-gates, while the last layer consists of $n - 2^{\lceil\log n\rceil - 1}$ gates. In total, this gives precisely $n-1$, the same as for the CNOT-ladder implementation. When the qubits adhere to a k-dimensional grid topology, Rosenbaum [Ros13] showed that the Fanout-gate can be implemented in depth $\Theta(\sqrt[k]{n})$, using $\mathcal{O}(n)$ gates and $\mathcal{O}(n)$ ancilla qubits. He also showed that this depth is



(a) A sequential implementation using a CNOT-ladder. Layers are separated by red dashed lines.

(b) The log-depth implementation. Layers are separated by red dashed lines. Gates within a single layer can be applied in parallel since the qubits they operate on do not overlap.

Figure 2.5: Two implementations of the $\mathsf{FO}^{(8)}$-gate.

asymptotically optimal when constrained to said topology. If we assume all-to-all connectivity, the log-depth implementation from Figure 2.5b is optimal, as the next lemma shows.

Lemma 6. *Any quantum circuit that implements the Fanout-gate must have a depth $d \geq \log(n)$. In particular, no QNC^0 circuit can compute the Fanout-gate.*

Proof. Consider any quantum circuit C of depth d that acts on n qubits and uses m ancilla qubits. Define a directed graph $G = (V, E)$ with $d+1$ layers of nodes $V_0, \ldots, V_d$, each corresponding to one time step of the circuit, separated by d layers $\mathsf{U}_0, \ldots, \mathsf{U}_{d-1}$ of quantum gates. Every layer consists of $n+m$ nodes. Two nodes in adjacent layers V_i and V_{i+1} have an edge $(v_i, v_{i+1}) \in E$ between them if the corresponding qubits are involved in the same gate. For any node in $v \in V_0$, we define its *light cone* as the set of nodes it can affect, i.e., the set of nodes in $v' \in V_d$ such that there is a path from v to v'. The light cone of a qubit in the circuit is the light cone of the corresponding node. Considering a gate set consisting only of single-qubit and two-qubit gates, it is easy to see that the light cone of any qubit is at most 2^d.

After applying a $\mathsf{FO}^{(n)}$-gate, the $n-1$ target qubits are all fully entangled with the control qubit. In particular, the control qubit must affect all n qubits in the circuit (including itself), meaning the light cone must be at least n. It follows that $2^d \geq n$, so $d \geq \log(n)$. □

2.7.3 OR-gate

We can use Fanout-gates to construct other useful gates, such as the $\mathsf{OR}^{(n)}$-gate, which acts as the logical OR of the input qubits:

$$\mathsf{OR}^{(n)} |x_1, \ldots, x_{n-1}\rangle |b\rangle := |x_1, \ldots, x_{n-1}\rangle |b \oplus \bigvee_{i=0}^{n-1} x_i\rangle . \tag{2.31}$$

Høyer and Špalek first showed how to reduce the problem of computing OR on n qubits to computing it on $\log(n)$ qubits. Later, Takahashi and Tani [TT12] discovered a circuit for computing OR that is 'almost' in QNC^0_f – not fully, since it has exponential size! Combined with the OR-reduction, however, the circuit size becomes polynomial in n.

Lemma 7 ('OR-reduction', Lemma 5.1 from [Hv05]). *The* $\mathsf{OR}^{(n+1)}$*-gate can be reduced to* $\mathsf{OR}^{(m+1)}$*, where* $m = \lceil \log(n+1) \rceil$*, using* $2n\lceil \log(n+1) \rceil$ *ancilla qubits and* $2n + 2\lceil \log(n+1) \rceil$ *Fanout-gates of arity at most* n*. That is, there is a family of quantum circuits in* QNC^0_f *that acts as*

$$|x\rangle |0\rangle^{\otimes m} \to |x\rangle |\psi_x\rangle \tag{2.32}$$

for every $x \in \{0,1\}^n$*, where* $|\psi_x\rangle$ *is an* m*-qubit quantum state such that* $\langle 0^m | \psi_x \rangle = 1 - OR(x)$*.*

Proof. For $\theta \in [-1,1]$, we define the state $|\mu_\theta^{|x|}\rangle := \frac{1}{2}(1+\exp(i\pi\theta\,|x|))\,|0\rangle + \frac{1}{2}(1-\exp(i\pi\theta\,|x|))\,|1\rangle$. To prepare this state, we apply a Hadamard-gate to the qubit in state $|0\rangle$, followed by a $\mathsf{FO}^{(n)}$-gate copying this qubit onto $n-1$ ancilla qubits. This yields the state

$$|x\rangle \frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}}. \tag{2.33}$$

Next, we apply a $\mathsf{R_Z}(\theta x_i)$-gate, controlled by $|x_i\rangle$, to the i-th qubit of $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$, for all $i \in [n]$, resulting in the state

$$|x\rangle \frac{|0\rangle^{\otimes n} + \exp\left(i\pi\theta \sum_{i\in[n]} x_i\right) |1\rangle^{\otimes n}}{\sqrt{2}} = |x\rangle \frac{|0\rangle^{\otimes n} + \exp(i\pi\theta\,|x|)\,|1\rangle^{\otimes n}}{\sqrt{2}}. \tag{2.34}$$

Finally, uncomputing the first step leads to $|x\rangle\,|\mu_\theta^{|x|}\rangle$, as we desired. The OR-reduction works by computing the states $|\psi_k\rangle = |\mu_{\theta_k}^{|x|}\rangle$, with $\theta_k = 1/2^k$ in parallel for all $k \in [m]$. This requires copying and un-copying the input $|x\rangle$ a total of $m-1$ times, using Fanout-gates of arity m. We are left to show that the resulting state $|\psi\rangle = |\psi_0\rangle\,|\psi_1\rangle \ldots |\psi_{m-1}\rangle$ behaves precisely as $|\psi_x\rangle$. Indeed, if $|x| = 0$, then $|\psi_k\rangle = |0\rangle$ for each $k \in [m]$ (see Equation (2.34)). If $|x| > 0$, then we have $\langle 1|\psi_k\rangle = 1$ with certainty for at least one $k \in [m]$. Indeed, considering the unique prime factorization of $|x|$ as $|x| = 2^a(2b+1)$, for integers $a \in [p]$ and $b \geq 0$, it follows that

$$|\psi_a\rangle = \frac{1}{2}(1+\exp(i\pi(2b+1)))\,|0\rangle + \frac{1}{2}(1-\exp(i\pi(2b+1)))\,|1\rangle = |1\rangle\,. \tag{2.35}$$

This proves the correctness of the OR-reduction. The reduction uses a total of $n(m-1) + m(n-1)$ ancilla qubits, $2n$ Fanout-gates of arity m, and $2m$ Fanout-gates of arity n. □

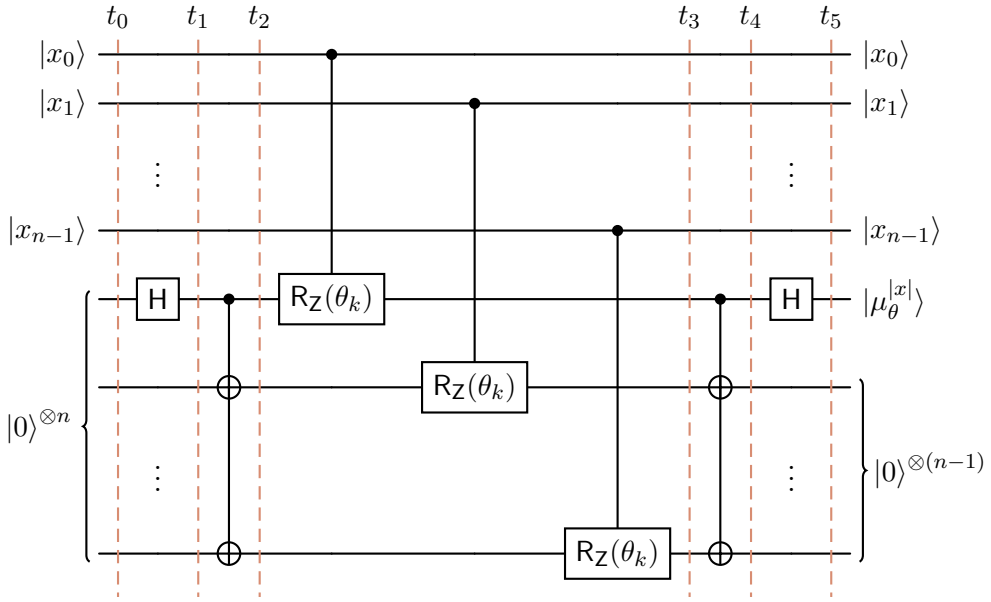
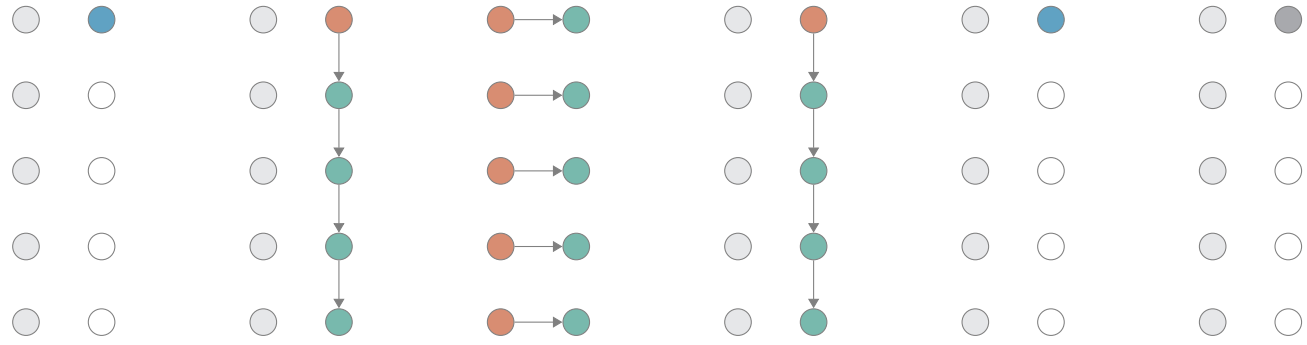


Figure 2.6: Circuit for preparing the state $|\mu_\theta^{|x|}\rangle$. Note that the indicated slices here do not necessarily represent a single time step.



(a) t_0, H-gate. (b) t_1, $\mathsf{FO}^{(5)}$-gate. (c) t_2, $\mathsf{R_Z}$-gates. (d) t_3, $\mathsf{FO}^{(5)}$-gate. (e) t_4, H-gate. (f) t_5.

Figure 2.7: The circuit in Figure 2.6, applied to qubits arranged in a 2D grid. The qubits colored gray are the input qubits (in this case, five of them). The dark gray qubit in t_5 is in the output state $|\mu_\theta^{|x|}\rangle$. A Hadamard-gate is denoted by a blue-colored qubit, while red and green qubits denote the control and target qubit respectively of a two-qubit controlled gate.

A nice property of the OR-reduction (and the OR-gate discussed shortly) is that it easily adheres to 2D grid topology. Figure 2.7 exemplifies this by showing how the circuit for preparing the state $|\mu_{\theta_k}\rangle$ can be performed on a 2D grid, with no depth overhead.

The exponential-size circuit for the OR-gate uses the Fourier expansion of the OR-function, in particular, it uses the following result:

Lemma 8 (Theorem 1 from [TT12])**.** *For any $x \in \{0,1\}^n$, $OR(x) = \frac{1}{2^n-1}\sum_{a\in\{0,1\}^n\setminus\{0^n\}} PA_a(x)$. Here, $PA_a(x) = \oplus_{i=0}^{n-1} a_i x_i$ is the parity of x, weighted by the non-zero binary vector a.*

The exponential-sized circuit for the OR-gate consists of the following steps:

1. Do the following at the same time:
   (a) Copy the input state $|x\rangle$ and apply a $\mathsf{PA}_a^{(n)}$-gate in parallel for every non-zero n-bit string a (i.e. 2^n-1 times);
   (b) Prepare a (2^n-1)-qubit GHZ state;
2. Apply controlled $\mathsf{R_Z}$-gates in parallel, controlled by the qubits holding the results of the weighted Parity-gates and targeting the qubits of the GHZ-state. By Lemma 8, this prepares the state
$$\begin{aligned}&\frac{|0\rangle^{\otimes(2^n-1)} + \exp\left(i\pi\frac{1}{2^n-1}\sum_{a\in\{0,1\}^n\setminus\{0^n\}}\mathrm{PA}_a^{(n)}(x)\right)|1\rangle^{\otimes(2^n-1)}}{\sqrt{2}}\\ =&\frac{|0\rangle^{\otimes(2^n-1)} + \exp\left(i\pi\mathrm{OR}^{(n)}(x)\right)|1\rangle^{\otimes(2^n-1)}}{\sqrt{2}}.\end{aligned} \tag{2.36}$$
3. Again apply a $\mathsf{FO}^{(2^n-1)}$ and a Hadamard-gate to the first of these qubits, such that in the end it holds the result $|\mathrm{OR}^{(n)}(x)\rangle$.
4. Finally, uncompute the results of the weighted Parity-gates and the copies of the input state $|x\rangle$.

Note that the size of the circuit is dominated by the first step, and is thus $\mathcal{O}(n2^n)$. The depth of the circuit is constant, modulo the implementation of the $\mathsf{FO}^{(2^n-1)}$-gate. Combining this circuit with the OR-reduction of Lemma 7 gives a circuit implementing the $\mathsf{OR}^{(n)}$-gate with $\mathcal{O}(n\log(n))$ ancilla qubits, and – assuming all-to-all connectivity – a circuit depth of $\mathcal{O}(\log(n))$.

2.7.4 AND-gate

We now shift our attention to another important gate, namely the AND-gate (or *Toffoli*-gate), which acts as

$$\mathsf{AND}^{(n)} |x_0, \dots, x_{n-1}\rangle |b\rangle := |x_0, \dots, x_{n-1}\rangle |b \oplus \bigwedge_{i=0}^{n-1} x_i\rangle . \tag{2.37}$$

Figure 2.8 shows the standard implementation of an $\mathsf{AND}^{(3)}$-gate. To construct the $\mathsf{AND}^{(n)}$-gate for $n = 4$

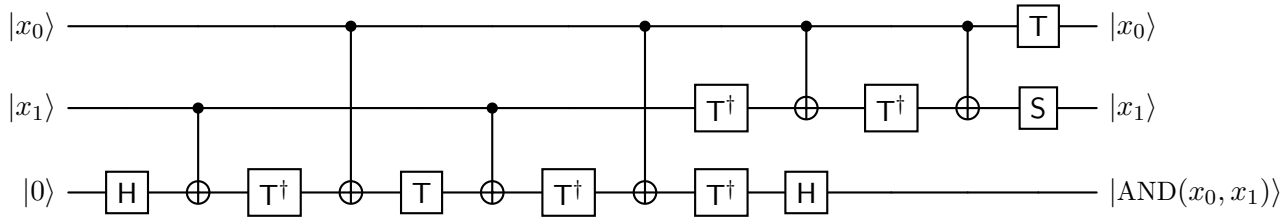
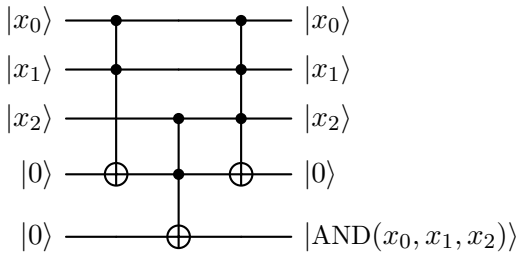
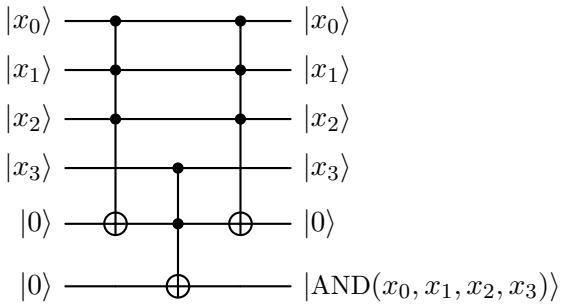


Figure 2.8: Standard implementation of an $\mathsf{AND}^{(3)}$-gate (Toffoli-gate).

and $n = 5$, observe the circuits in Figure 2.9. For general n, this yields a circuit of depth $\mathcal{O}(n)$. A simple



(a) An $\mathsf{AND}^{(4)}$-gate (using one ancilla).



(b) An $\mathsf{AND}^{(5)}$-gate (using two ancillas, one of which 'hidden' in the implementation of the $\mathsf{AND}^{(4)}$-gate).

Figure 2.9: An $\mathsf{AND}^{(4)}$-gate and $\mathsf{AND}^{(5)}$-gate. A gate with m controls and a single target drawn represents an $\mathsf{AND}^{(m+1)}$-gate.

way to implement the $\mathsf{AND}^{(n)}$-gate in depth $\mathcal{O}(\log(n))$ is to use the $\mathsf{OR}^{(n)}$-gate implementation from the previous section, together with the observation that $\mathsf{AND}^{(n)} = \mathsf{X}^{\otimes n}\mathsf{OR}^{(n)}$. Just like the OR-gate, this uses $\mathcal{O}(n \log(n))$ ancilla qubits. A very recent paper by Nie et al. [NZS24] shows that this logarithmic depth can be attained even when few ancilla qubits are available.

Lemma 9 (Theorem 1 and 6 from [NZS24], logarithmic-depth AND)**.** *There exists a quantum circuit that implements the* $\mathsf{AND}^{(n)}$*-gate, which has a circuit depth* $\mathcal{O}(\log n)$*, and uses one ancilla qubit. If no ancilla qubits are available, the depth increases to* $\mathcal{O}(\log^2 n)$*.*

An implementation of the $\mathsf{AND}^{(n)}$-gate automatically gives an implementation for the $\mathsf{EQ}_k^{(n)}$-gate, which acts as

$$\mathsf{EQ}_k^{(n)} |x_0, \dots, x_{n-1}\rangle |b\rangle := |x_0, \dots, x_{n-1}\rangle |b \oplus \mathbb{1}_{x=k}\rangle , \tag{2.38}$$

by noting that $\mathsf{EQ}_k^{(n)} = \mathsf{X}_{\Delta\bar{k}}^{(n-1)}\mathsf{AND}^{(n)}\mathsf{X}_{\Delta\bar{k}}^{(n-1)}$.

2.7.5 Permutation-gate

Another gate we will use is the Permutation-gate $\mathsf{PERM}_\sigma^{(n)}$, which permutes n-qubit computational basis states according to a permutation σ:

$$\mathsf{PERM}_\sigma^{(n,d)} |x\rangle := |\sigma(x)\rangle . \tag{2.39}$$

For a permutation σ on n-bit strings with $|\sigma| = d$, let $\{x_k\}_{k\in[d]}$ be the non-identity elements in the domain. The circuit implementing $\mathsf{PERM}_\sigma^{(n,d)}$ starts by applying Fanout-gates to make d copies of the input. Then a EQ_{x_i}-gate is applied to the i-th copy as well as to the i-th fresh ancilla qubit. Then, we uncopy the input register. The result of these first steps is a one-hot encoding of the domain of σ, i.e. the k-th ancilla qubit is in the state $|1\rangle$ if the input state is $|x_k\rangle$:

$$|x_k\rangle\,|e_k\rangle\,. \tag{2.40}$$

To perform the permutation, we need to apply a $\mathsf{X}^{(n)}_{\Delta x_k \oplus \sigma(x_k)}$-gate controlled by the k-th ancilla qubit, to transform the state $|x_k\rangle$ to $|\sigma(x_k)\rangle$. To do this, we can invoke Theorem 2, by noting that all $\mathsf{X}^{(n)}_{\Delta x_k \oplus \sigma(x_k)}$-gates are mutually diagonalized by the $\mathsf{Z}^{\otimes n}$-gate. After permuting, the one-hot encoding needs to be uncomputed. This can be done in the same way as the computing step, though this time using $\mathsf{EQ}_{\sigma(x_k)}$-gates.

2.7.6 Uniformly controlled gates

The last class of multi-qubit gates we will consider in this chapter are the so called *uniformly controlled gates*. For any function $f : \{0,1\}^{n-1} \to \mathcal{U}(2)$ from the set of $(n-1)$-bit strings to the set of single-qubit unitaries, the f-uniformly controlled gate of arity n (f-$\mathsf{UCG}^{(n)}$) is defined as

$$\begin{aligned} f\text{-}\mathsf{UCG}^{(n)} &:= \sum_{x\in\{0,1\}^{n-1}} |x\rangle\langle x| \otimes f(x) \\ &= \sum_{x\in\{0,1\}^{n-1}} |x\rangle\langle x| \otimes \mathsf{U}_x, \end{aligned} \tag{2.41}$$

where in the second line we have equated $\mathsf{U}_x := f(x)$. In other words, a uniformly controlled gate applies a different single-qubit gate to the target qubit for every distinct n-qubit computational basis state of the control qubits. A simple way to visualize an f-$\mathsf{UCG}^{(n)}$-gate is shown in Figure 2.10, where each gate U_x is applied sequentially controlled on the state $|x\rangle$.

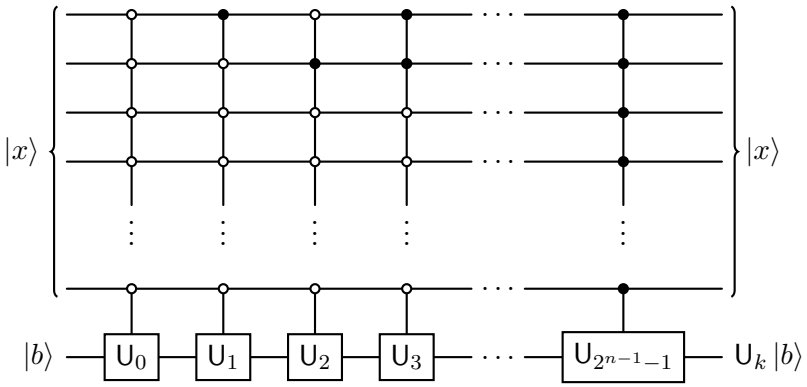


Figure 2.10: A sequential implementation of the f-$\mathsf{UCG}^{(n)}$-gate. Here, the filled control dots are activated when the qubit is in state $|1\rangle$, while unfilled dots are activated when the qubit is in state $|0\rangle$.

We may write UCG instead of f-UCG when this does not cause confusion. Uniformly controlled gates have been studied extensively in the context of quantum memory circuits [All+23], recommendation systems [KP16] and linear system solving [HHL09]. In Chapter 4, we will use these gates as a subroutine for quantum state preparation, and explore multiple implementations of them.

Chapter 3

Adaptive quantum circuits

Considering the surprising parallelizing power that the quantum Fanout-gate provides, researchers have been on a quest to find scalable, constant-depth implementations for this gate. In [BKP09], it is shown that the *one-way quantum computation model*, which works by generating an entangled multi-qubit state, followed by a sequence of *adaptive* measurements, is equivalent to the standard quantum circuit model augmented with unbounded Fanout-gates. 'Adaptive' in this context means that the choice for a particular operation can depend in some way on the outcome of previous measurements. The proof in [BKP09] makes use of an important connection between Clifford circuits and measurement-based quantum computing. In particular, it has been known that Clifford circuits can be parallelized – to constant depth – by allowing intermediate measurements, classical processing of the measurement outcomes and quantum gates that depend on the outcome of this processing [GC99]. Given that the Fanout-gate is a Clifford circuit, this provides a pathway towards perhaps more efficient and more easily implementable quantum computations.

In Section 3.1 and 3.2, we will shortly review quantum teleportation and show how Clifford circuits can be implemented in constant-depth when adaptive gates are allowed. In Section 3.3, we define the *Local Alternating Quantum Classical Computations* (LAQCC) circuit class, that formalizes these 'adaptive circuits', and adds constraints on the allowed qubit topology. The LAQCC model first appeared in [Buh+23], where it was used to construct shallow circuits for some entangled and structured states. In Section 3.4, we will present some of these subroutines and state preparation schemes, which are of later use in the general quantum state preparation algorithms. We end the chapter in Section 3.5 by discussing the relation between LAQCC and other complexity classes, and between it and other, similar adaptive quantum computation schemes.

3.1 Quantum teleportation

Quantum teleportation is a technique for transferring quantum information between two parties that share an entangled quantum state and can communicate classically. First discussed in [Ben+93], it has been a quintessential tool in many quantum information processing protocols. Since its advent, it has been experimentally tested plenty of times, for example through sending information across the Danube river and between two Canary Islands [Urs+04; Ma+12].

Formally, the input to the standard teleportation scheme is a single qubit (say, in register $\mathtt{a}$) in some state $|\psi\rangle_{\mathtt{A}} = \alpha|0\rangle + \beta|1\rangle$, and two qubits (say, in register $\mathtt{b}$ and $\mathtt{c}$) that together are in a Bell-state $|\Psi_{00}\rangle_{\mathtt{bc}} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Now, by performing a Bell measurement on registers $\mathtt{a}$ and $\mathtt{b}$, we get (using the projection from Equation (2.10)):

$$\begin{aligned}\sum_{i,j\in\{0,1\}} \left(\langle ii| \left(\mathsf{Z}^a\mathsf{X}^b \otimes \mathsf{I}\right)|\psi\rangle_{\mathtt{a}} |jj\rangle_{\mathtt{bc}}\right) &= \sum_{i,j} \langle i| \mathsf{Z}^m\mathsf{X}^{m'} |\psi\rangle |j\rangle_{\mathtt{c}} \\ &= \sum_i |i\rangle \langle i| \mathsf{Z}^a\mathsf{X}^b |\psi\rangle_{\mathtt{c}} \\ &= \mathsf{Z}^m\mathsf{X}^{m'} |\psi\rangle_{\mathtt{c}} .\end{aligned} \tag{3.1}$$

That means that after this measurement, register c now holds the state that originally was in register a, modulo some Pauli errors. If we know the measurement outcomes and are allowed to apply quantum gates that depend on these, we can correct these errors and succesfully transport $|\psi\rangle$ from register a to register c. This is called teleportation, since we can – in theory – take registers b and b and move them to different locations far away from each other, after creation of the Bell-state. Performing a measurement on one of the two entangled qubits together with the qubit in register a, and subsequently sending the measurement outcomes over some classical communication channel, *moves* the data of register a to the other qubit in the entangled pair.

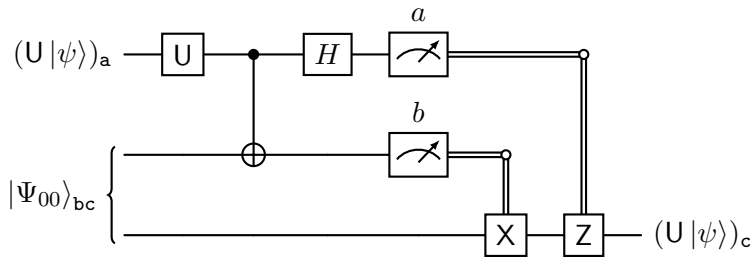


Figure 3.1: The quantum state teleportation protocol for transferring the state $\mathsf{U}|\psi\rangle$. The thick wires represent classical communication.

Note that there are no requirements on the amplitudes of the state $|\psi\rangle_\mathtt{a}$, and in particular we could have applied a single-qubit gate U before the teleportation protocol, thereby teleporting the state $\mathsf{U}|\psi\rangle_\mathtt{A}$; see Figure 3.1.

Suppose that after two of these teleportation protocols, we have ended up with two qubits in some registers c and c′, to which we want to apply a CNOT-gate. Then we can choose to apply the CNOT-gate after the Pauli corrections of these previous protocols. However, as Gottesman and Chuang [GC99] noticed, the commutation relations of the CNOT-gate with the Pauli-gates (see Table 2.2) also allows us to move the CNOT-gate to the beginning of the circuit, before the error corrections:

$$\mathsf{CNOT}(\mathsf{Z}^a\mathsf{X}^b \otimes \mathsf{Z}^{a'}\mathsf{X}^{b'})\,|\psi\rangle_\mathtt{a}\,|\psi\rangle_{\mathtt{a}'} = (\mathsf{Z}^a\mathsf{X}^{b+b'} \otimes \mathsf{Z}^{a+a'}\mathsf{X}^{b'})\mathsf{CNOT}\,|\psi\rangle_{\mathtt{aa}'}\,. \tag{3.2}$$

See also Figure 3.2. Instead of waiting to apply the CNOT-gate until after the corrections of the preceding gates, we can apply it directly, and *then* perform the Pauli corrections. Equation (3.2) shows that the Pauli corrections that need to be applied on c now also depend on the measurement outcomes in the protocol for c′, and vice-versa.

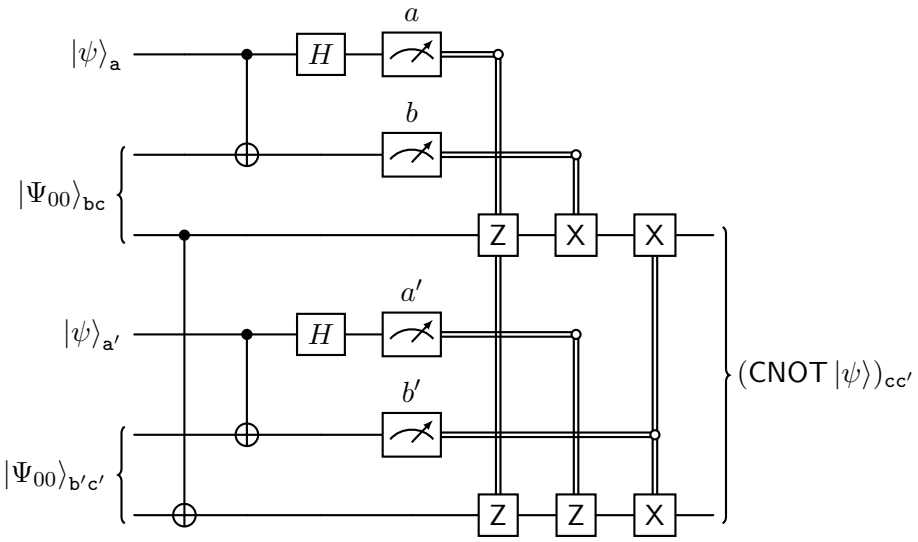


Figure 3.2: Implementing a CNOT-gate using the teleportation protocol.

3.2 Teleporting Clifford circuits

In principle, the protocols above allow us to perform any number of quantum gates on disjoint qubits in constant depth using adaptive measurements. How does this extend to circuits containing multiple layers?

Suppose we want to apply a polynomial-size n-qubit quantum circuit consisting of layers $\mathsf{U}_0, \dots, \mathsf{U}_{d-1}$ using the teleportation protocols from the last section. In principle, this can be done by applying all gates from U_0 using the teleportation protocols, then use the resulting registers (registers c and c' in Figure 3.2) as input for the protocols for U_1, and so forth. Importantly, however, we can not perform the measurements for the different layers at the same time; in general, the Pauli errors caused by the teleportation scheme for U_0 must be corrected before we can start applying U_1. As a result, using this teleportation-based implementation does not result in lower-depth circuits compared to the generic circuit model.

If we instead choose *not* to perform the Pauli corrections after every layer, then we can decrease the depth. In fact, since the different circuit layers do not interact with each other except for these corrections, we can perform all teleportation protocols in parallel, achieving a circuit of *constant* depth. The catch is of course, that the output register of the last circuit layer will no longer contain the state $\mathsf{U}_{d-1} \dots \mathsf{U}_0 |\psi\rangle$, but will instead be interspersed with accumulated Pauli errors:

$$|\psi'\rangle = \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{d-1,i}} \mathsf{X}_{\to i}^{b_{d-1,i}}\right) \mathsf{U}_{d-1} \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{d-2,i}} \mathsf{X}_{\to i}^{b_{d-2,i}}\right) \cdots \mathsf{U}_1 \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{0,i}} \mathsf{X}_{\to i}^{b_{0,i}}\right) \mathsf{U}_0 |\psi\rangle\,, \tag{3.3}$$

where $a_{k,i}$ and $b_{k,i}$ for $0 \le i \le n-1$ and $0 \le k \le d-1$ denote either a measurement outcome or, in the case of a CNOT-gate, the sum of two measurement outcomes. For general quantum circuits, it is unclear how to retrieve the desired output state $\mathsf{U}_{d-1} \dots \mathsf{U}_0 |\psi\rangle$ from $|\psi'\rangle$. If the circuit is *Clifford*, however, we can use the fact that all gates commute with the Pauli group to move all errors to the end of the circuit. For example, for just two layers we get

$$\left(\prod_{i=0}^{n-1} \mathsf{X}_{\to i}^{a_{1,i}} \mathsf{Z}_{\to i}^{b_{1,i}}\right) \mathsf{U}_1 \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{0,i}} \mathsf{X}_{\to i}^{b_{0,i}}\right) \mathsf{U}_0 |\psi\rangle = \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{1,i}} \mathsf{X}_{\to i}^{b_{1,i}}\right) \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a'_{0,i}} \mathsf{X}_{\to i}^{b'_{0,i}}\right) \mathsf{U}_1 \mathsf{U}_0 |\psi\rangle \tag{3.4}$$

$$= \left(\prod_{i=0}^{n-1} \mathsf{Z}_{\to i}^{a_{1,i}} \mathsf{Z}_{\to i}^{a'_{0,i} \oplus 1} \mathsf{X}_{\to i}^{b_{1,i}} \mathsf{X}_{\to i}^{b'_{0,i}}\right) \mathsf{U}_1 \mathsf{U}_0 |\psi\rangle \tag{3.5}$$

$$= \left(\prod_{i=0}^{n-1} (\mathsf{Z}^{a_{1,i} \oplus a'_{0,i} \oplus 1})_{\to i} (\mathsf{X}^{b_{1,i} \oplus b'_{0,i}})_{\to i}\right) \mathsf{U}_1 \mathsf{U}_0 |\psi\rangle\,, \tag{3.6}$$

where the new coefficients a' and b' are sums of elements of a and b, and we can write $\oplus$ instead of $+$ since $\mathsf{X}^2 = \mathsf{Z}^2 = \mathsf{I}$. By iterating the procedure above – commuting and then combining the coefficients – the output state of the full circuit can be written as

$$|\psi'\rangle = \left(\prod_{i=0}^{n-1} (\mathsf{Z}^{\hat{a}_i} \mathsf{X}^{\hat{b}_i})_{\to i}\right) |\psi\rangle\,, \tag{3.7}$$

where the elements of n-bit strings $\hat{a}$ and $\hat{b}$ result from calculating the parity of some of the measurement outcomes. Using an NC^1-circuit, we can determine these elements. This circuit consists of a layer of classical Fanout-gates – one for each measurement outcome – followed by a layer of classical Parity-gates based on the gates applied. The outcome of this classical circuit can be used to apply the actual Pauli corrections, obtaining the state $|\psi'\rangle$ in constant depth.[1]

3.3 The LAQCC model

We will proceed to formally define the LAQCC model, and discuss some gates and tools that are accessible within this model. The definition for this model stems from [Buh+23], on which much of this section is also

[1] Determining which Parity-gates to apply is not in NC^1, but it does follow directly from the quantum circuit. A logspace (L) precomputation gives this circuit [Buh+23].

based.

Definition 4 (Local Alternating Quantum-Classical Computations)**.** $\mathsf{LAQCC}(\mathcal{Q},\mathcal{C})^d$ *is the class of circuits such that:*

- *every quantum layer implements a quantum circuit* $Q \in \mathcal{Q}$*, constrained to a grid topology;*
- *every classical layer implements a classical circuit* $C \in \mathcal{C}$*;*
- *there are* d *alternating layers of quantum and classical circuits;*
- *after every quantum layer a subset of the qubits is measured;*
- *the classical circuit receives the measurement outcomes of previous quantum layers as its input; and*
- *the output of the classical circuits control quantum operations in future layers.*

We set $\mathsf{LAQCC} := \mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1)^{\mathcal{O}(1)}$, that is, a constant number of alternating layers of constant-depth quantum circuits and classical NC^1-circuits.

Each layer in an LAQCC circuit consists of three steps: a quantum circuit (that may depend on the outcome of the previous classical circuit), a measurement, and a classical circuit. Compared to other adaptive circuit models in the literature, the CCNTC model of Pham and Svore [PS13] was the first protocol to formalize these three steps in the layout of the LAQCC model. It already included qubit topology constraints, specifically a two-dimensional grid. This is generalized in the LAQCC model to also include grids of higher dimensions. Furthermore, extra constraints are added to the intermediate classical computations in the definition of LAQCC, allowing to upper bound its power.

The LAQCC model seems very similar to an instantiation of Jozsa's Class, defined in [CM20]. Specifically, LAQCC is equivalent to the $(n, \mathcal{O}(1), \mathcal{O}(\log(n)), \mathcal{O}(1), \mathcal{O}(\mathrm{poly}(N)))$-jozsa-quantum circuit class, with the exception of the grid constraints.

3.4 Gates, states, and subroutines in LAQCC

The teleportation gadgets from Section 3.2 can be used to implement any Clifford circuit in constant depth using an adaptive circuit. To adhere to the grid constraint imposed by the LAQCC model, we use the fact that any Clifford unitary can be mapped to a linear-depth circuit that adheres to a *1D grid* constraint [MR18; BM21].

A general form of such circuits are so-called *Clifford-grid* circuits.

Definition 5. *Let* $\{\mathsf{U}_{ij}\}_{i\in[d],j\in[n/2]}$ *be Clifford unitaries. A Clifford-grid circuit of depth* d *is a circuit of the form*

$$C = \prod_{i\in[d]} \bigotimes_{j\in[n/2]} \mathsf{U}_{ij}, \tag{3.8}$$

where U_{ij} *acts on qubits* $2j$ *and* $2j+1$ *if* i *is even, and* $2j+1$ *and* $2j+2$ *if* i *is odd.*

Theorem 3. *Any Clifford-grid circuit* C *of depth* $\mathcal{O}(n)$ *has an* LAQCC *implementation that uses* $\mathcal{O}(n^2)$ *ancilla qubits.*

Theorem 3 gives rise to plethora of useful routines accessible in LAQCC. For starters, any SWAP-circuit that is needed to perform a 2-qubit gate between non-adjacent qubits is a Clifford-grid circuit and thus in LAQCC, which in a sense removes the locality constraint imposed by the grid topology for applying a two-qubit gate on non-adjacent qubits. In particular, it can be seen from the teleportation protocols that implementing a single-qubit gate uses two ancilla qubits, while a CNOT-gate uses four. A gate of particular interest is – unsurprisingly – the Fanout-gate, which is also available by Theorem 3. However, there is a slightly more efficient circuit, which we will show in this section. This uses an LAQCC preparation scheme of the GHZ-state.

3.4.1 GHZ-state

There is a simple LAQCC-circuit to prepare a GHZ-state, based on *parity measurements*. It is shown in Figure 3.3 for $n = 3$.

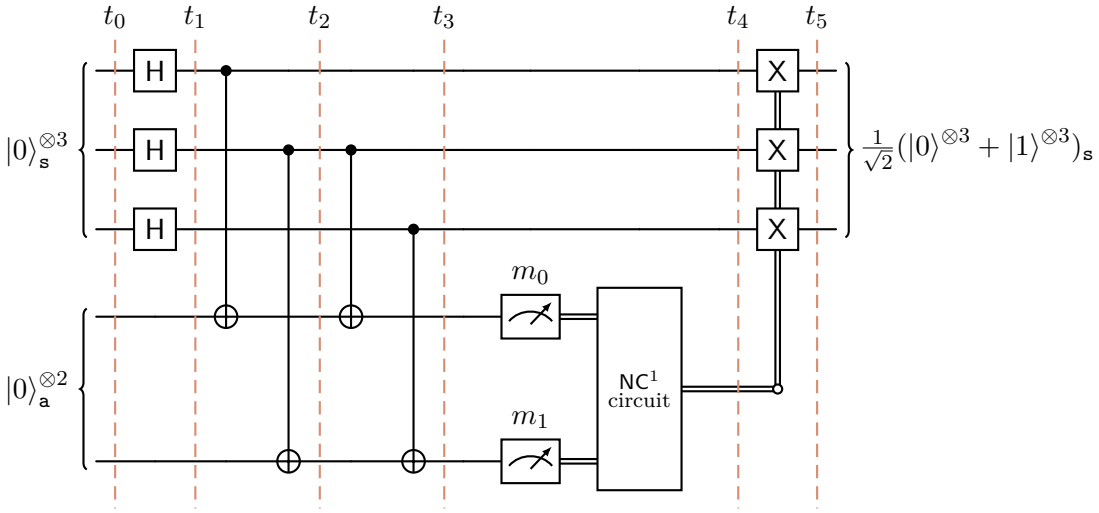


Figure 3.3: An LAQCC-circuit to prepare the 3-qubit GHZ-state. Again, the thick wires indicate classical information and processing.

At time step t_2, the qubits are in the state

$$\frac{1}{\sqrt{8}} \sum_{x \in \{0,1\}^3} |x_0 x_1 x_2\rangle_{\mathtt{s}} \left(|x_0 \oplus x_1\rangle |x_1 \oplus x_2\rangle\right)_{\mathtt{a}}, \tag{3.9}$$

with $\mathtt{s}$ being the state register (which will end up in the state $|\mathrm{GHZ}^{(3)}\rangle$) and $\mathtt{a}$ the ancilla register. Measuring register $\mathtt{a}$ yields two measurement results (m_0 and m_1), and register $\mathtt{s}$ collapses to

$$\frac{1}{\sqrt{2}} \sum_{\substack{x \in \{0,1\}^3 \\ x_0 \oplus x_1 = a \\ x_1 \oplus x_2 = b}} |x_0 x_1 x_2\rangle_{\mathtt{s}} = \frac{1}{\sqrt{2}} \sum_{x \in \{0,1\}} (|x\rangle |x \oplus m_0\rangle |x \oplus m_0 \oplus m_1\rangle)_{\mathtt{s}}. \tag{3.10}$$

By flipping the qubits according to m_0 and m_1, we can prepare the GHZ-state. This procedure can be generalized to prepare the n-qubit GHZ-state: given measurement outcomes $m_0, \ldots, m_{n-2}$ an NC^1-circuit can determine the parities $\hat{m}_j := \bigoplus_{i \in [j]} m_i$ for $0 \leq j \leq n-2$ and flip the j-th qubit of register $\mathtt{s}$ if $\hat{m}_j = 1$. This procedure uses $n-1$ ancilla qubits.

Lemma 10. *For any $n \geq 3$, the $\mathrm{GHZ}^{(n)}$-state can be prepared using an* LAQCC*-circuit with $n-1$ ancilla qubits.*

If the parity corrections are skipped, the n qubits in register $\mathtt{s}$ are of the form $\frac{1}{\sqrt{2}}(|x\rangle + |\bar{x}\rangle)$ for some $x \in \{0,1\}^n$ that is determined by the measurement outcomes. The resulting circuit is in QNC^0, since there is no adaptivity in play. These so-called *poor man's cat states* are used in [Wat+19] to show separations between the class QNC^0 and the classical circuit class AC^0.

3.4.2 Fanout-gate

Figure 3.4 shows the LAQCC implementation of the $\mathsf{FO}^{(4)}$-gate when given an 4-qubit GHZ-state as input. Combined with the state preparation protocol discussed in the previous section, we obtain a LAQCC-circuit for a general $\mathsf{FO}^{(n)}$-gate that uses $2n-1$ ancilla qubits.

The circuit starts with entangling the control qubit (in register $\mathtt{c}$) with the first qubit of the GHZ-state (in ancilla register $\mathtt{a}$), measuring the first qubit, and then appling an X-gate to all other qubits in the GHZ-state

if the measurement outcome is 1. The total quantum state is then

$$|x\rangle_{\mathtt{c}}\,|x\rangle_{\mathtt{a}}^{\otimes n-1}\,|b_0\dots b_{n-2}\rangle_{\mathtt{t}}\,. \tag{3.11}$$

Hereafter, we can entangle the $n-1$ target qubits in one layer via CNOT-gates controlled by the qubits in register a and targeting the qubits in target register t. Applying a Hadamard-gate to all qubits in register a and then measuring it induces a phase-flip on the control qubit if the parity of all measurement outcomes is 1. After correcting this, we end up with the implemented Fanout-gate.

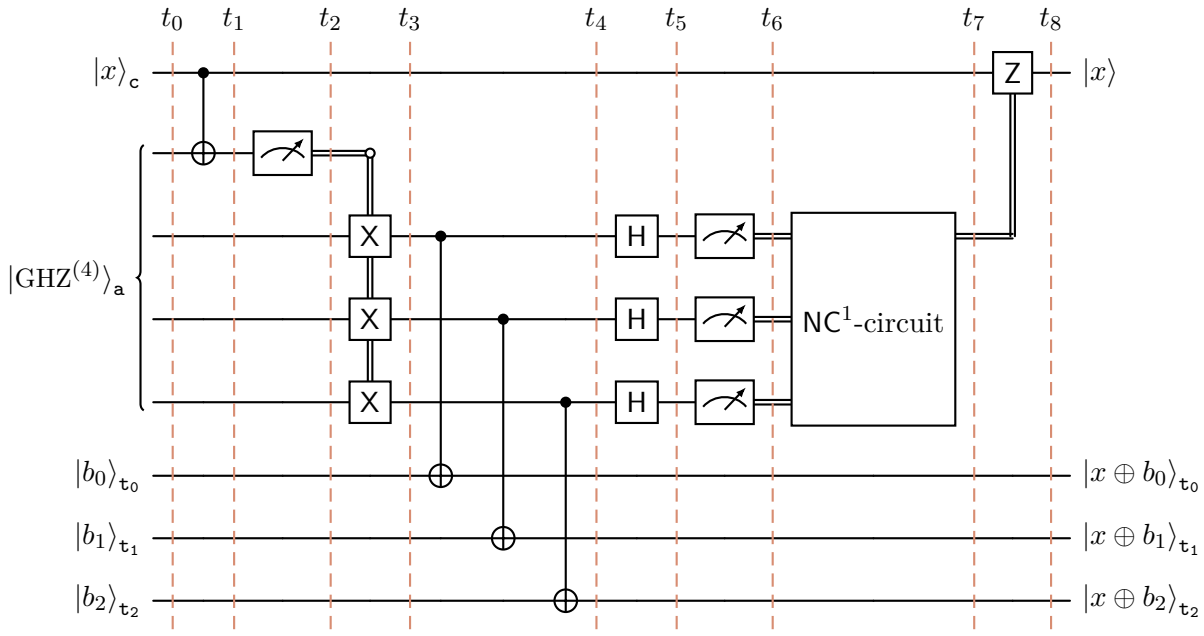


Figure 3.4: LAQCC-circuit implementing the $\mathsf{FO}^{(4)}$-gate when given an $|\mathrm{GHZ}^{(n)}\rangle$-state as input.

Following the implementations in Section 2.7 using Fanout-gates, we immediately obtain a LAQCC implementation for the gates $\mathsf{OR}^{(n)}$, $\mathsf{AND}^{(n)}$ and $\mathsf{EQ}_{\mathsf{k}}{}^{(n)}$. Asymptotically, these use $\mathcal{O}(n\log n)$ ancilla qubits, which is the same as the non-adaptive circuits. However, the constant hiding underneath these asymptotics are higher for the LAQCC-circuit.

3.4.3 Other states preparable in LAQCC

In [Buh+23], Buhrman et al. use combinations of Fanout-gates and EQ-gates to prepare the W-state in LAQCC, using $\mathcal{O}(n\log(n)\log\log(n))$ ancilla qubits. A similar method prepares the Dicke-state $|\mathrm{D}_k^{(n)}\rangle$ in LAQCC as long as $k=\mathcal{O}(\sqrt{n})$. For larger k, a $\mathsf{LAQCC}(\mathsf{QNC}^0,\mathsf{NC}^1)^{\mathcal{O}(\log n)}$-circuit is presented, which uses mappings between two different number representation methods, and needs $\mathcal{O}(n^2\log(n))$ ancilla qubits. More recently [Yu+24], procedures have been shown that prepare the Dicke-state using an adaptive circuit of polylogarithmic depth that uses less ancillas (only polylogarithmically many).

Another recent paper by Smith et al. [Smi+24] encompassed a method for preparing certain *translationally-invariant matrix product states* (TI MPS), which are superpositions of qudits with amplitudes that can be expressed by a set of matrices. In particular, such an n-qudit state $|\psi\rangle$ can be written as

$$|\psi\rangle = \sum_{m_0,\dots,m_{n-1}\in[d]} \mathrm{Tr}\left[\mathsf{A}_{m_0}\mathsf{A}_{m_1}\dots\mathsf{A}_{m_{n-1}}\mathsf{B}\right]|m_0\dots m_{n-1}\rangle\,, \tag{3.12}$$

where $\{\mathsf{A}_{m_k}\}_{k\in[d]}$ are $D\times D$-dimensional matrices and the matrix B specifies its *boundary conditions*. The number D is called the *bond dimension* of a matrix product state. In [Smi+24], it is shown that TI MPS can be prepared in LAQCC if $D=\mathcal{O}(1)$ and a suitable correction mechanism – in the form of measurement basis and adaptive correction gates – can be found. In most cases, there is yet no clear algorithmic way known

to find these. An exception to this is the class of states correctable via measurement in the (generalized) D-dimensional Bell basis, generated by applying the following gates to $|\mathrm{GHZ}^{(D)}\rangle$:

$$\mathsf{X}_D = \sum_{k\in[D]} |k+1\rangle\langle k|, \qquad \mathsf{Z}_D = \sum_{k\in[D]} \exp\left(i2\pi\frac{k}{D}\right)|k\rangle\langle k|, \tag{3.13}$$

as shown in [ZGS24]. As an example: the AKLT states, which are n-qutrit (so $d = 3$) states with a bond dimension of two, described by the matrices

$$A^0 = \sqrt{\frac{2}{3}}\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}, \qquad A^1 = \sqrt{\frac{2}{3}}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \qquad A^2 = -\sqrt{\frac{2}{3}}\begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}, \tag{3.14}$$

and any choice of boundary conditions, can be prepared using this method [Smi+23].

In other recent work, this time by Raveh and Nepomechie [RN24], new MPS representations of the (n, k)-Dicke-state have been found. However, they are not translationally invariant and no suitable correction mechanism for applying the method from [Smi+24] for these representations are known. In general, finding a 'nice' MPS representation for these states seems hard: even for the n-qubit W-state, it is conjectured that any TI MPS representation with periodic boundary conditions (i.e., with $\mathsf{B} = \mathsf{I}$) will have a bond dimension of at least $\mathcal{O}(n^{1/3})$ [Per+07]. A weaker version of this hypothesis was proved in [MS18], namely that for all $\delta > 0$, the bond dimension for a TI MPS representation the W-state is lower bounded by $\mathcal{O}(n^{\frac{1}{3+\delta}})$. The best known minimum bond dimension for a TI MPS representation known with periodic boundary conditions is $\lfloor\frac{n}{2}\rfloor + 1$, shown in [KSB23].

Even if suitable correction mechanisms are found for these smaller representations, a *constant-depth* preparation scheme for *all* Dicke-states remains elusive, both in the adaptive and non-adaptive case.

3.5 Complexity of LAQCC

We end this chapter by briefly discussing how LAQCC relates to other circuit complexity classes.

Any LAQCC-circuit can be written as a composition of unitary layers, measurements and classical computation layers. Specifically, for some constant d, unitaries $\{\mathsf{U}_k\}_{k\in[d]}$ (each of them in QNC^0), measurements $\{M_k\}_{k\in[d]}$ and classical circuits $\{C_k\}_{k\in[d]}$ (each of them in NC^1), a LAQCC-circuit L can be written as

$$L = M_{d-1}\mathsf{U}_{d-1}C_{d-1}\dots M_0\mathsf{U}_0C_0. \tag{3.15}$$

Using this, we get the following lemma.

Lemma 11. *For any* LAQCC*-circuit L there exists a* QNC^1*-circuit Q that outputs the same state as L and does not use intermediate measurements.*

Proof. Consider the decomposition of L as in Equation (3.15). By the deferred measurement principle, we can postpone the measurement layers until the end of the circuit using CNOT-gates to fresh ancilla qubits. These quantum wires can be used in place of the classical output wires. By Lemma 1 and since $\oplus_{\mathrm{f}} \subset \mathsf{NC}^1$, a QNC^1 circuit V_k can take on the role of the classical circuits C_k. Since there are only constantly many layers of quantum and classical circuits, the resulting circuit L', which can be written as

$$\mathsf{L}' = \mathsf{U}_{d-1}\mathsf{V}_{d-1}\dots\mathsf{U}_0\mathsf{V}_0, \tag{3.16}$$

is in QNC^1. □

Furthermore, since Fanout-gates are constructible in LAQCC, we immediately have the following result.

Lemma 12. $\mathsf{QNC}^0_f \subseteq \mathsf{LAQCC}$.

in [GS19], it is conjectured that

$$\mathsf{NC}^1 \not\subseteq \mathsf{QNC}^0_\mathrm{f}. \tag{3.17}$$

This is an extension of the widely assumed conjecture that $\mathsf{NC}^1 \not\subseteq \mathsf{TC}^0$, since it was shown in [Hv05; TT12] that Majority-gates, and therefore TC^0-circuits, can be implemented in $\mathsf{QNC}^0_\mathrm{f}$.[2] Obviously, $\mathsf{NC}^1 \subseteq \mathsf{LAQCC}$, and so if the conjectured separation from [GS19] indeed holds, then this would mean that the inclusion in Lemma 12 is strict, i.e. that $\mathsf{QNC}^0_\mathrm{f} \subsetneq \mathsf{LAQCC}$.

In [Buh+23], it is shown that all *Instantaneous Quantum Polynomial-time* IQP-circuits, which are circuits consisting of 'temporally unstructured' (i.e., commuting) quantum gates [SB09], have an LAQCC implementation.

Definition 6 (Definition 1 from [NM14])**.** *An* IQP*-circuit on n qubits is a polynomial-size quantum circuit with the following structure:*

- *the input state is $|+\rangle^{\otimes n}$;*
- *each gate in the circuit is diagonal in the computational basis; and*
- *the output is the result of a measurement in the Hadamard basis on a specified set of qubits in the output register.*

Lemma 13 (Lemma 3.16 from [Buh+23])**.** *Every* IQP*-circuit has an* LAQCC *implementation.*

Proof. Note that all steps comprising an IQP-circuit are accessible in the LAQCC class. Firstly, the input state can be created by a single layer of Hadamard-gates on all qubits. Since in this basis, all gates in the computational basis commute, we can apply the parallelization lemma (see Lemma 2) to achieve a constant-depth LAQCC-circuit. Finally, applying another layer of Hadamard-gates and measuring in the computational basis gives the same result as an original IQP-circuit. □

Earlier in 2010, Bremner, Jozsa and Shepherd [BJS11] had shown that efficient weak classical simulation[3] of all IQP-circuits would imply a collapse of the polynomial hierarchy to the third level, which is not believed to be true. By Lemma 13, the same statement holds in LAQCC.

[2] This proof uses the OR-gate implementation from Section 2.7.3 to implement quantum Threshold-gates.

[3] A circuit family is *weakly simulatable* if, given the description of the circuit family, its output distribution can be sampled classically in polynomial time, up to small multiplicative error.

Chapter 4

Quantum State Preparation

In this chapter, we will dive into the problem of Quantum State Preparation (QSP). For starters, we will define the problem of QSP formally, and fix some notation.

Definition 7. *Quantum State Preparation (QSP) is the problem defined by the following input and output:*

- ***Input**: a classical vector $\psi = (\psi_0, \ldots, \psi_{N-1}) \in \mathbb{C}^N$ with $\sum_{i=0}^{N-1} |\psi_i|^2 = 1$ and $N = 2^n$ for some $n \in \mathbb{N}$.*
- ***Output**: a unitary U_ψ such that $\mathsf{U}_\psi |0\rangle = |\psi\rangle$, where $|\psi\rangle$ is the n-qubit state defined as*

$$|\psi\rangle = \sum_{i=0}^{N-1} \psi_i |i\rangle . \tag{4.1}$$

The restriction of N to powers of two can be 'bypassed' by padding ψ with zeroes at the end. This definition of QSP is also known as *general QSP*, since no structural features on the quantum state to be prepared are assumed.

We call a vector ψ *d-sparse* if ψ has exactly d non-zero coefficients. We define the set

$$\Psi := \{(\psi_i^*, q_i) | \psi_i^* \in \mathbb{C}, 0 \leq q_i \leq N-1, 0 \leq i \leq d-1\} \tag{4.2}$$

consisting of pairs (ψ_i^*, q_i) where ψ_i^* is the value of the i-th non-zero entry of ψ, and q_i its location. Then

$$|\psi\rangle = \sum_{i \in [N]} \psi_i |i\rangle = \sum_{i \in [d]} \psi_i^* |q_i\rangle . \tag{4.3}$$

For the general QSP procedures, we will see that there will always be a comparable procedure with circuit complexity depending on the sparsity d of the input vector, rather than the maximum sparsity N. If d is sufficiently small (for example, polynomial in n), this can speed up the preparation process considerably.

The QSP algorithms discussed in this chapter can be divided into two categories. The first category, which we will call *UCG-based QSP*, goes back to the idea by Grover and Rudolph [GR02] to use uniformly controlled gates (see Section 2.7.6) for preparing quantum states. This idea is described in Theorem 6, which we will prove in Section 4.2.

Theorem 6. *QSP can be solved by n UCGs of growing sizes*

$$\mathsf{U}_\psi = \mathsf{UCG}^{(n)} \mathsf{UCG}^{(n-1)} \cdots \mathsf{UCG}^{(2)} \mathsf{UCG}^{(1)} . \tag{4.4}$$

In the second category, which we will call *unary-based QSP*, quantum states are prepared by first preparing an N-qubit *unary encoding* of the entries of ψ:

$$|\psi_{unary}\rangle := \sum_{i=0}^{N-1} \psi_i |e_i\rangle , \tag{4.5}$$

	method	based on	depth all-to-all	depth LAQCC	width all-to-all	width LAQCC
UCG-based	**sequential**	[GR02],[RLR23], [Mot+04]	$\mathcal{O}(Nn\log(n))$ $\mathcal{O}(d\log\log(d)+\log(n))$	$\mathcal{O}(N)$ $\log(d)$	$\mathcal{O}(n)$ $\mathcal{O}(dn)$	$\mathcal{O}(n\log(n))$ $\mathcal{O}(dn\log(n))$
	parallelized	[All+23]	$\mathcal{O}(n^2)$ $\mathcal{O}(\log^2(d)+\log(n))$	$\mathcal{O}(n)$ $\mathcal{O}(\log(d))$	$\mathcal{O}(Nn)$ $\mathcal{O}(dn)$	$\mathcal{O}(Nn\log(n))$ $\mathcal{O}(dn\log(n))$
	Yuan et al.	[Sun+23],[YZ23], [Ros23]	$\mathcal{O}(n+\frac{N}{n+m})$ -	- -	$n+m$ -	- -
unary-based	**data loader**	[Joh+20; Buh+23]	$\mathcal{O}(n)$ $\mathcal{O}(\log(dn))$	$\mathcal{O}(N)$ $\mathcal{O}(d)$	$\mathcal{O}(Nn)$ $\mathcal{O}(dn)$	$\mathcal{O}(Nn\log(n))$ $\mathcal{O}(dn\log(n))$
	Luo and Li	[LL24]	- $\mathcal{O}(\log(dn))$	- -	- $\mathcal{O}(\frac{dn}{\log(d)})$	- -

Table 4.1: Some QSP algorithms along with their complexity in terms of circuit depth and width, both for non-adaptive, all-to-all connectivity and for the LAQCC setting. The algorithms in **bold** are discussed in this chapter, and used in the analyses of Chapter 5. The complexities are given both for preparing a fully dense n-qubit state and for a d-sparse n-qubit state. For the method from Yuan et al. (resp. Luo and Li), only the preparation of dense (resp. sparse) quantum states using non-adaptive circuits are considered (hence, the '-' in some of the cells). The circuit by Yuan et al. gives a QSP circuit for any specified amount of ancilla qubits m.

and then transforming this into the state $|\psi\rangle$ required in Definition 7. The best known algorithm for preparing $|\psi_{unary}\rangle$ is by Johri et al. [Joh+20].

Recent studies [LL24; Sun+23; ZLY22; YZ23] have explored QSP algorithms that achieve optimal circuit depth, both in the general and in the sparse setting. The following two results give a lower bound on the depth needed to prepare arbitrary (d-sparse) quantum states.

Theorem 4 (Theorem 3 from [Sun+23])**.** *Given m ancilla qubits, the minimum circuit depth of preparing an arbitrary quantum state is $\Omega(\max(n, \frac{N}{m+n}))$, even if the circuit uses arbitrary one- and two-qubit gates.*

Theorem 5 (Lemma 3 from [ZLY22])**.** *Given an arbitrary amount of ancilla qubits, the minimum circuit depth of preparing all arbitrary d-sparse quantum state is $\Omega(\log(dn))$.*

Theorem 5 also bounds the power of constant-depth LAQCC-circuits, in the following sense.

Corollary 1. *When $d \in \Omega(n^{\log(n)})$, not every d-sparse quantum state can be prepared using an $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1)^{\mathcal{O}(1)}$-circuit.*

Proof. By Theorem 5, the minimum circuit depth of preparing all arbitrary d-sparse quantum states is $\Omega(\log(dn))$. Let $d \in \Omega(n^{\log(n)})$, say $d \geq c \cdot n^{\log(n)}$ for some constant c. Let ψ be a d-sparse vector for which this circuit depth is achieved. Then the minimum circuit depth of preparing $|\psi\rangle$ is at least $c' \cdot \log(dn) \geq c' \log\left(cn^{\log(n)}n\right) = c'\log(c) + c'(\log(n)+1)\log(n) \in \Omega(\log^2(n))$. In particular, $|\psi\rangle$ can not be prepared by a QNC^1 circuit. Since $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1)^{\mathcal{O}(1)} \subseteq \mathsf{QNC}^1$ (by Theorem 11), we conclude that not every d-sparse quantum state can be prepared using a $\mathsf{LAQCC}(\mathsf{QNC}^0, \mathsf{NC}^1)^{\mathcal{O}(1)}$-circuit. □

In recent papers [Sun+23; All+23; LL24], QSP algorithms have been proposed that achieve these lower bounds. We summarize some of them in Table 4.1, along with their complexities in terms of circuit depth and width. Of course, besides these general or sparse QSP algorithms, there is a plethora of procedures catering towards more structured states, often using much simpler and more efficient circuits. We have already seen examples of these, in the form of GHZ-state and the W-state preparation protocols.

We end this introduction to QSP by listing the content of the rest of the chapter. In Section 4.1, we sketch the classical pre-processing steps that lie at the start of all discussed QSP algorithms. In Section 4.2, we discuss UCG-based QSP, and describe two UCG implementations. In Section 4.3, we discuss unary-based QSP.

4.1 Classical pre-processing using a binary tree

Since we are given the vector ψ as a classical input, all QSP algorithms start with some classical pre-processing of ψ. A simple way of doing this is by representing its entries in a binary tree. One such tree is given in Example 1.

For an N-dimensional vector ψ, this concerns a tree of depth $\log(N)+1 = n+1$, with each node (k,j) in the tree ($k \in [n]$, $j \in [2^k]$) consisting of two numbers: a real number $t_{k,j}$ and an angle $\varphi_{k,j} \in [0, 2\pi)$. The leaf nodes are defined by the amplitude and phase of the entries of ψ:

$$t_{n,j} = |\psi_j|\,, \qquad \varphi_{n,j} = \arg(\psi_j). \tag{4.6}$$

For internal nodes, the values at parent nodes are defined by the values of its children:

$$t_{k,j} = \sqrt{(t_{k+1,2j})^2 + (t_{k+1,2j+1})^2}, \qquad \varphi_{k,j} = \varphi_{k+1,2j+1} - \varphi_{k+1,2j}. \tag{4.7}$$

The number $t_{k,j}$ contains the total amplitude in the state $|\psi\rangle$ of all n-qubit computational basis states of which the first k qubits are in the state $|j\rangle$. In other words: upon measuring the first k qubits, we expect to measure $|j\rangle$ with probability $(t_{k,j})^2$. For every internal node (k,j), the angle $\theta_{k,j}$ quantifies the relative contribution of the two child nodes to the total amplitude $t_{k,j}$.

$$\theta_{k,j} = \arccos\left(\frac{t_{k+1,2j}}{t_{k,j}}\right). \tag{4.8}$$

Using this definition, we indeed see that the cosine and sine,

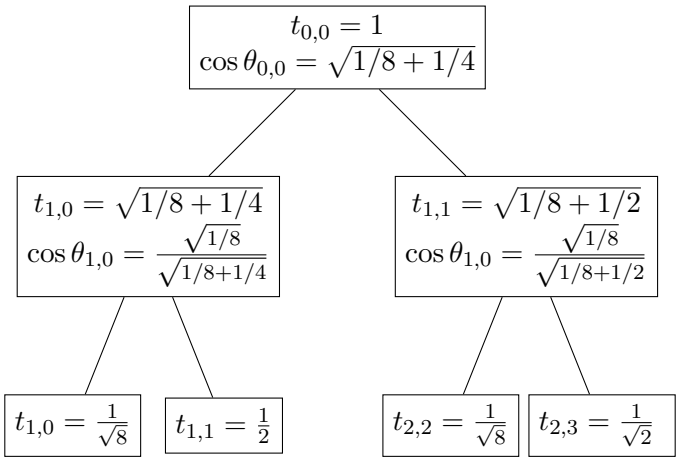
$$\begin{aligned}
\cos(\theta_{k,j}) &= \frac{t_{k+1,2j}}{t_{k,j}} = \frac{t_{k+1,2j}}{\sqrt{(t_{k+1,2j})^2 + (t_{k+1,2j+1})^2}} \\
\sin(\theta_{k,j}) &= \sqrt{1-\cos^2(\theta_{k,j})} = \sqrt{1 - \frac{(t_{k+1,2j})^2}{(t_{k+1,2j})^2 + (t_{k+1,2j+1})^2}} = \frac{t_{k+1,2j+1}}{\sqrt{(t_{k+1,2j})^2 + (t_{k+1,2j+1})^2}} = \frac{t_{k+1,2j+1}}{t_{k,j}}
\end{aligned} \tag{4.9}$$

denote the relative contribution to the total parent amplitude of the left and right child respectively.

Example 1 (Binary tree for a simple vector)**.** *Below we draw the binary tree for the vector*

$$\psi = \begin{bmatrix} \frac{1}{\sqrt{8}} & \frac{1}{2} & \frac{1}{\sqrt{8}} & \frac{1}{\sqrt{2}}. \end{bmatrix}^T \tag{4.10}$$

If we measure the first l qubits (where $l \in \{1,2\}$ in this case) of the corresponding state $|\psi\rangle = \frac{1}{\sqrt{8}}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{\sqrt{8}}|10\rangle + \frac{1}{\sqrt{2}}|11\rangle$, the probability of measuring $|j\rangle$ is given by $(t_{l,j})^2$.



For each layer in the tree, the values in the tree can be calculated in parallel. Thus, this entire pre-processing step can be executed using a classical circuit of depth $\mathcal{O}(n)$.

4.2 UCG-based QSP

Using the notation from the tree constructed in the pre-processing circuit, we are ready to prove Theorem 6. This theorem is stated implicitly in [GR02].

Theorem 6. *QSP can be solved by n UCGs of growing sizes*

$$\mathsf{U}_\psi = \mathsf{UCG}^{(n)}\mathsf{UCG}^{(n-1)}\cdots\mathsf{UCG}^{(2)}\mathsf{UCG}^{(1)}. \tag{4.11}$$

Proof. The circuit consists of n UCGs $\mathsf{U}_0,\ldots,\mathsf{U}_{n-1}$. Each unitary U_k involves the first $k+1$ qubits and is controlled by the first k qubits, see Figure 4.1. U_0 is not controlled and consists of two single-qubit rotations on the first qubit:

$$\mathsf{U}_0 = \left(\mathsf{R}_\mathsf{Z}(\varphi_{0,0})\mathsf{R}_\mathsf{Y}(2\theta_{0,0})\right)_{\to 0}. \tag{4.12}$$

These bring the n qubits to the state

$$\begin{aligned}\left(\mathsf{R}_\mathsf{Z}(\varphi_{0,0})\mathsf{R}_\mathsf{Y}(2\theta_{0,0})\right)_{\to 0}|0\rangle^{\otimes n} &= \mathsf{R}_\mathsf{Z}(\varphi_{1,1}-\varphi_{1,0})_{\to 0}\left(t_{1,0}|0\rangle + t_{1,1}|1\rangle\right)|0\rangle^{\otimes n-1}\\ &= \left(e^{i\varphi_{1,0}}t_{1,0}|0\rangle + e^{i\varphi_{1,1}}t_{1,1}|1\rangle\right)|0\rangle^{\otimes n-1}.\end{aligned} \tag{4.13}$$

For $1\le k\le n-1$, the UCGs are defined similarly, though now controlled on the previous k qubits as

$$\mathsf{U}_k = \sum_{j=0}^{2^k-1}|j\rangle\langle j|_{\to 0\ldots k-1}\otimes\left(\mathsf{R}_\mathsf{Z}(\varphi_{k,j})\mathsf{R}_\mathsf{Y}(2\theta_{k,j})\right)_{\to k}. \tag{4.14}$$

We claim that, after applying U_{n-1}, the constructed state is precisely $|\psi\rangle$. To see this, let

$$|\psi_k\rangle = \left(\sum_{j=0}^{2^k-1}e^{i\varphi_{k,j}}t_{k,j}|j\rangle\right)|0\rangle^{n-k}, \tag{4.15}$$

and assume that after the unitary U_{k-1}, the qubits are in the state $|\psi_k\rangle$. Note that this certainly holds after U_0, since the state in Equation (4.13) is equal to $|\psi_1\rangle$. Applying U_k then gives

$$\begin{aligned}\mathsf{U}_k\left(\sum_{j\in[2^k]}e^{i\varphi_{k,j}}t_{k,j}|j\rangle\right)|0\rangle^{n-k} &= \left(\sum_{j\in[2^k]}|j\rangle\langle j|_{\to 0\ldots k-1}\otimes\left(\mathsf{R}_\mathsf{Z}(\varphi_{k,j})\mathsf{R}_\mathsf{Y}(2\theta_{k,j})\right)_{\to k}\right)\left(\sum_{j=0}^{2^k-1}e^{i\varphi_{k,j}}t_{k,j}|j\rangle\right)|0\rangle^{\otimes n-k}\\ &= \sum_{j\in[2^k]}e^{i\varphi_{k,j}}t_{k,j}|j\rangle\left(e^{i\varphi_{k+1,2j}}\frac{t_{k+1,2j}}{t_{k,j}}|0\rangle + e^{i\varphi_{k+1,2j+1}}\frac{t_{k+1,2j+1}}{t_{k,j}}|1\rangle\right)|0\rangle^{\otimes n-k-1}\\ &= \left(\sum_{j\in[2^k]}e^{i\varphi_{k+1,2j+1}}t_{k+1,2j}|2j\rangle + e^{i(2\varphi_{k+1,2j+1}-\varphi_{k+1,2j})}t_{k+1,2j+1}|2j+1\rangle\right)|0\rangle^{\otimes n-k-1}\\ &= \left(\sum_{j\in[2^k]}e^{i\varphi_{k+1,2j}}t_{k+1,2j}|2j\rangle + e^{i\varphi_{k+1,2j+1}}t_{k+1,2j+1}|2j+1\rangle\right)|0\rangle^{\otimes n-k-1}\\ &= \sum_{j\in[2^k]}e^{i\varphi_{k+1,j}}t_{k+1,j}|j\rangle|0\rangle^{\otimes n-k-1}.\\ &= |\psi_{k+1}\rangle.\end{aligned} \tag{4.16}$$

By induction, this means that after U_{n-1}, the qubits are in the state

$$|\psi_n\rangle = \sum_{j\in[2^n]}e^{i\varphi_{n,j}}t_{n,j}|j\rangle = \sum_{j\in[2^n]}\psi_j|j\rangle = |\psi\rangle. \tag{4.17}$$

□

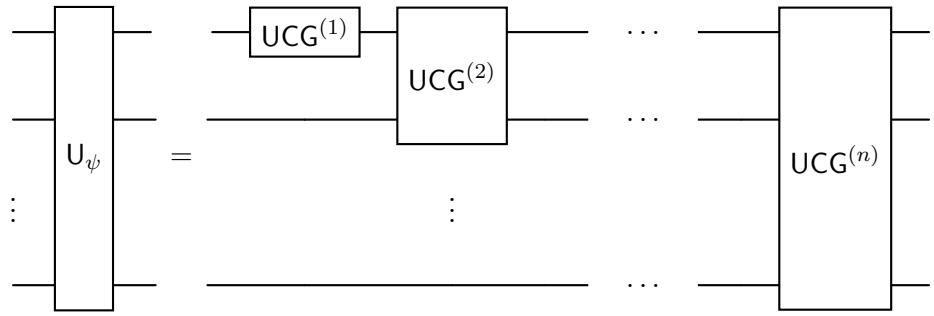


Figure 4.1: UCG-based QSP, the correctness of which is proved in Theorem 6.

To prepare a d-sparse quantum state using uniformly controlled gates, Ramaciotti et al. [RLR23] presented a permutation-based solution. For this, let σ_ψ be the permutation defined as $\sigma_\psi(i) := q_i$, where q_i are the locations of the non-zero entries of ψ (see Equation (4.2)). Then, the following procedure prepares $|\psi\rangle$:

(i) First, a dense state $|\psi_{dense}\rangle$ is prepared, with amplitudes given by the entries of ψ:

$$|\psi_{dense}\rangle = \sum_{i\in[d]} \psi_i^* |i\rangle , \tag{4.18}$$

using Theorem 6 on only $\lceil \log(d) \rceil$ qubits.

(ii) Then, a $\mathsf{PERM}_\sigma^{(n)}$-gate on the $\lceil \log(d) \rceil$ 'dense' qubits and $n - \lceil \log(d) \rceil$ fresh qubits permutes the computatoinal basis states, preparing $|\psi\rangle$:

$$\sum_{i\in[d]} \psi_i' |i\rangle |0\rangle^{\otimes n - \lceil \log(d) \rceil} \to \sum_{i\in[d]} \psi_i' |q_i\rangle . \tag{4.19}$$

The rest of this section instantiates Theorem 6 by discussing two circuits implementing a UCG: a *sequential* and a *parallelized* circuit.

4.2.1 Sequential UCG

Perhaps the most obvious way to implement a UCG is by quite literally following the circuit in Figure 2.10. To control a gate by an arbitrary $(n-1)$-qubit computational basis state $|k\rangle$, an $\mathsf{EQ}_k^{(n)}$-gate can be applied to the $n-1$ input qubits and an ancilla qubit. This ancilla qubit then controls the gate U_k, which is the unitary applied to the target register and controlled by $|k\rangle$. See Figure 4.2.

Dividing the qubits into an $(n-1)$-qubit input register $\mathtt{i}$, target register $\mathtt{t}$ and ancilla register $\mathtt{a}$, the full sequence of gates shown in Figure 4.2 is given by

$$\mathsf{UCG}^{(n)} = \mathsf{X}_{\to \mathtt{a}} \prod_{k\in[2^{n-1}]} \mathsf{C}_{\mathtt{a}}(\mathsf{U}_k)_{\to \mathtt{t}} (\mathsf{EQ}_k^{(n)})_{\to \mathtt{ia}}, \tag{4.20}$$

where by $\mathtt{ia}$ we mean the combined registers $\mathtt{i}$ and $\mathtt{a}$. The final X-gate restores the ancilla register to $|0\rangle$.

Complexity

Given all-to-all qubit connectivity, the circuit depth of an $\mathsf{EQ}_k^{(n)}$-gate is $\mathcal{O}(\log(n))$, and so the circuit depth of one instance of Equation (4.20) is $\mathcal{O}(N \log(n))$. For the full QSP protocol, we use the circuit in Figure 4.1 and Theorem 6 to get

$$\text{depth(QSP, seq. } \mathsf{UCG}\text{, all-to-all)} = \sum_{k=1}^{n} \mathcal{O}(2^k \log(k)) \in \mathcal{O}(N \log(n)). \tag{4.21}$$

Meanwhile, the best obtainable circuit width is $\mathcal{O}(n)$.

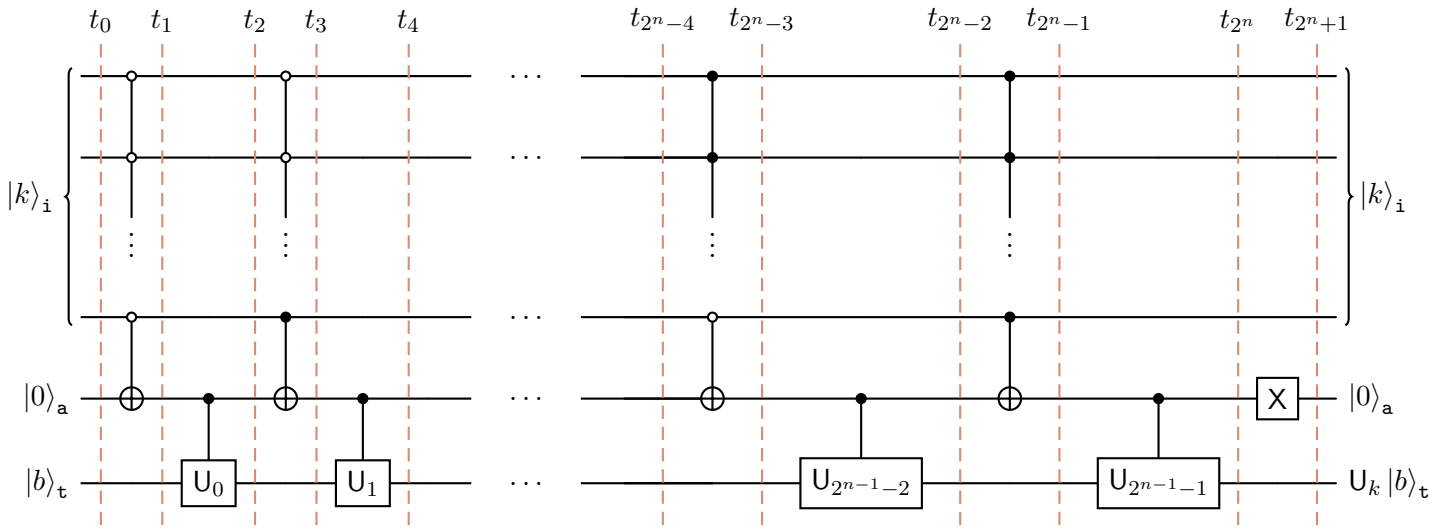


Figure 4.2: A sequential implementation of a UCG, using EQ-gates onto an ancilla qubit and controlling the gates on it being in the $|1\rangle$-state.

In the LAQCC setting, all gates used in Equation (4.20) can be performed in constant depth, giving a total circuit depth of $\mathcal{O}(N)$ both for a single UCG as for the full QSP protocol. Since the $\mathsf{EQ}^{(n)}$-gate implementation uses $\mathcal{O}(\log(n))$ extra ancilla qubits per input qubit, the total circuit width in the LAQCC setting becomes $\mathcal{O}(n\log(n))$.

As for preparing a d-sparse quantum state in this manner, we use the fact that the circuit depth of the Permutation-gate is $\mathcal{O}(\log(n))$ with all-to-all connectivity, and $\mathcal{O}(1)$ in the LAQCC setting. Similarly, the circuit widths of this gate are $\mathcal{O}(dn)$ and $\mathcal{O}(dn\log(n))$ respectively. Together, this gives circuit depths of $\mathcal{O}(d\log\log(d)+\log(n))$ and $\mathcal{O}(d)+\mathcal{O}(1)\in\mathcal{O}(d)$ respectively, and circuit widths of $\mathcal{O}(\log(d)+dn)\in\mathcal{O}(dn)$ and $\mathcal{O}(\log(d)\log\log(d)+dn\log(n))\in\mathcal{O}(dn\log(n))$ respectively.

4.2.2 Parallelized UCG

Using extra ancilla qubits, we can lower the depth of the UCG implementation. This so-called parallelized method starts off by obtaining a one-hot encoding of all input strings in the domain of f. If it is defined for all $n-1$-bit strings, this can be done by the following gates:

$$\begin{aligned}|k\rangle_{\mathtt{i}}\,|0\rangle_{\mathtt{a}}^{\otimes(2^{n-1}-1)(n-1)}\,|0\rangle_{\mathtt{e}}^{\otimes 2^{n-1}}\,|b\rangle_{\mathtt{t}} &\to |k\rangle_{\mathtt{ia}}^{\otimes 2^{n-1}}\,|0\rangle_{\mathtt{e}}^{\otimes 2^{n-1}}\,|b\rangle_{\mathtt{t}}\\ &\to |k\rangle_{\mathtt{ia}}^{\otimes 2^{n-1}}\,|e_k\rangle_{\mathtt{e}}\,|b\rangle_{\mathtt{t}}\\ &\to |k\rangle_{\mathtt{i}}\,|0\rangle_{\mathtt{a}}^{\otimes(2^{n-1}-1)(n-1)}\,|e_k\rangle_{\mathtt{e}}\,|b\rangle_{\mathtt{t}}\,.\end{aligned} \tag{4.22}$$

Here, the first and third step copy and uncopy the input register $\mathtt{k}$ using n $\mathsf{FO}^{(N-1)}$-gates, and the second step applies $\mathsf{EQ}_k^{(n)}$-gates in parallel to the k-th copy of the input register and the k-th qubit in the encoding register $\mathtt{e}$. To implement the rest of the circuit, we decompose all unitaries in the image of f using its so-called Z-*decomposition*.

Theorem 7 (Theorem 4.1 from [NC10])**.** *Let $f:\{0,1\}^n\to\mathcal{U}(2)$ be a function onto single-qubit gates. Then there are functions $\alpha,\beta,\gamma,\delta:\{0,1\}^n\to[-1,1]$ such that*

$$f(x)=e^{i\pi\alpha(x)}\mathsf{R_Z}(\beta(x))\mathsf{HR_Z}(\gamma(x))\mathsf{HR_Z}(\delta(x)). \tag{4.23}$$

We say that the tuple $(\alpha,\beta,\gamma,\delta)$ is the Z*-decomposition of f.*

Using the Z-decomposition of f and the one-hot encoding created in Equation (4.22), a simple sequential way of implementing the UCG would be to apply the gates

$$\prod_{x\in[2^{n-1}]}\mathsf{C}_{\mathtt{e}_x}(\mathsf{U}_x)_{\to\mathtt{t}}=\prod_{x\in[2^{n-1}]}\mathsf{C}_{\mathtt{e}_x}\left(e^{i\pi\alpha(x)}\mathsf{R_Z}(\beta(x))\mathsf{HR_Z}(\gamma(x))\mathsf{HR_Z}(\delta(x))\right)_{\to\mathtt{t}}. \tag{4.24}$$

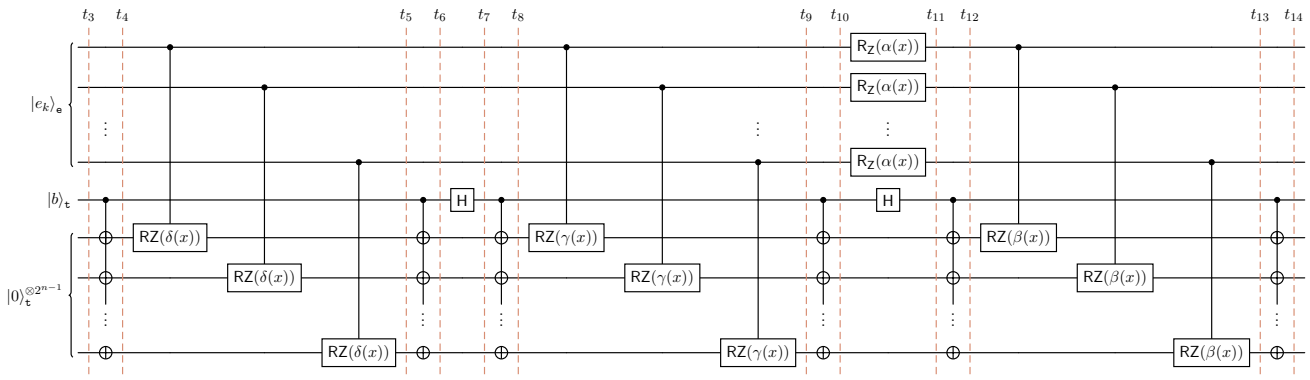


Figure 4.3: The main part of the parallelized UCG implementation. The input registers and the (un)computation of the one-hot encoding are not shown. Note that time step t_0 up to t_2 are not shown (they are part of the encoding step).

This just applies the gate U_x controlled by the qubit in the register $\mathsf{e_x}$ (that is, the x-th qubit of register e). Now, since e contains a one-hot encoding of the 2^{n-1} possible input states of the UCG, there is always at most one qubit in the register e that is set to $|1\rangle$ at this point in the circuit. Therefore, only one of the control wires will actually fire during this procedure. That means we can rearrange the gates in the Z-decomposition, and group together the Z-operators as follows:

$$\prod_{x\in[2^{n-1}]} \mathsf{C}_{\mathsf{e_x}}\mathsf{U}_{x\to\mathtt{t}} = \left(\prod_{x\in[2^{n-1}]} \mathsf{R_Z}(\alpha(x))_{\to\mathsf{e_x}}\right)\left(\prod_{x\in[2^{n-1}]} \mathsf{C}_{\mathsf{e_x}}\mathsf{R_Z}(\beta(x))_{\to\mathsf{t_x}}\right)\mathsf{H}_{\to\mathtt{t}}\left(\prod_{x\in[2^{n-1}]} \mathsf{C}_{\mathsf{e_x}}\mathsf{R_Z}(\gamma(x))_{\to\mathsf{t_x}}\right)$$
$$\mathsf{H}_{\to\mathtt{t}}\left(\prod_{x\in[2^{n-1}]} \mathsf{C}_{\mathsf{e_x}}\mathsf{R_Z}(\delta(x))_{\to\mathsf{t_x}}\right). \tag{4.25}$$

Since the $\mathsf{R_Z}$-gates are all diagonal (in the computational basis), we can parallelize the sequences of $\mathsf{R_Z}$-gates in Equation (4.25) using Fanout-gates, in a similar way to the circuit from Theorem 2. Figure 4.3 shows the result of this parallelization step. Note that the phase-gates parameterized by the function α are now applied to the register e in order to reduce the circuit depth. If this circuit is combined with the computation and uncomputation of the one-hot encoding, this gives an implementation of the full UCG.

Complexity

Given all-to-all qubit connectivity, the circuit depth of the full parallelized UCG circuit is

$$\begin{aligned}\text{depth}(\text{parallelized } \mathsf{UCG}^{(n)}, \text{all-to-all}) &= 2\cdot\mathcal{O}(\log(N)) + \mathcal{O}(\log(n)) && \text{(encoding step)}\\ &\quad + 6\cdot\mathcal{O}(\log(N)) && (t_3,t_5,t_7,t_9,t_{11},t_{13})\\ &\quad + 5\cdot\mathcal{O}(1) && (t_4,t_6,t_8,t_{10},t_{12})\\ &\quad \in \mathcal{O}(n), && (4.26)\end{aligned}$$

where the time steps t_i mentioned correspond to the time steps of the circuit in Figure 4.3, and the gates that are used in this time step. The circuit width is dominated by the encoding step, since this uses 2^{n-1} copies of an $(n-1)$-qubit input state, necessitating a circuit width of $\mathcal{O}(nN)$. For the full QSP protocol, we invoke the circuit in Figure 4.1 to get a circuit depth of $\mathcal{O}(n^2)$, and a circuit width of again $\mathcal{O}(nN)$.

In the LAQCC setting, all gates used in the UCG can be performed in constant depth, giving a circuit of constant depth performing the full UCG. By Theorem 6, this yields a QSP circuit of depth $\mathcal{O}(n)$. The total number of qubits used is again dominated by the encoding step, and since the $\mathsf{EQ}^{(n)}$-gates use an extra $\mathcal{O}(\log(n))$ ancilla qubits per input qubit, the total circuit width in the LAQCC setting becomes $\mathcal{O}(nN\log(n))$.

As for preparing a d-sparse quantum state in this manner, we note that the circuit depth of the Permutation-gate is $\mathcal{O}(\log(n))$ with all-to-all connectivity, and $\mathcal{O}(1)$ in the LAQCC setting. Furthermore, the circuit widths

for this gate are $\mathcal{O}(dn)$ and $\mathcal{O}(dn\log(n))$ respectively. Together, this gives circuit depths of

$$\text{depth(sparse QSP, parall. UCG, all-to-all)} \in \mathcal{O}(\log^2(d) + \log(n)), \tag{4.27}$$

and similarly

$$\text{depth(sparse QSP, parall. UCG, LAQCC)} = \mathcal{O}(\log(d)) + \mathcal{O}(1) \in \mathcal{O}(\log(d)), \tag{4.28}$$

and circuit widths of $\mathcal{O}(d\log(d) + dn) \in \mathcal{O}(dn)$ and $\mathcal{O}(d\log(d)\log\log(d) + dn\log(n)) \in \mathcal{O}(dn\log(n))$ respectively.

4.3 Unary-based QSP

4.3.1 Data loader

In this section, we will describe the *data loader* circuit from Johri et al. [Joh+20]. Given an N-dimensional vector ψ, this procedure prepares the N-qubit state

$$|\psi_{unary}\rangle := \sum_{i\in[N]} \psi_i |e_i\rangle . \tag{4.29}$$

A central role in this circuit is played by the so-called *Reconfigurable BeamSplitter*-gate, RBS, which for any angle θ is defined as

$$\mathsf{RBS}(\theta) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.30}$$

and acts as a rotation by an angle θ on the subspace spanned by the 2-qubit states $|10\rangle$ and $|01\rangle$. Figure 4.4 shows the decomposition of the RBS-gate into single-qubit and CNOT-gates. Assuming an all-to-all qubit

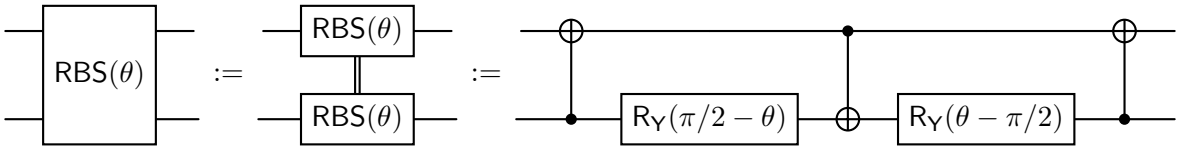


Figure 4.4: Circuit to implement an $\mathsf{RBS}(\theta)$-gate. In the second image, wires help indicate which gates are connected when applying the RBS-gate to qubits that are not adjacent.

topology, the data loader can be implemented using the circuit that is depicted in Figure 4.5 for $N = 8$. It has a circuit depth of $\mathcal{O}(\log(N)) = \mathcal{O}(n)$.

Note that for any node (k, j) in the binary tree induced by the vector ψ (see Section 4.1), we have

$$\mathsf{RBS}(\theta_{k,j})|10\rangle = \frac{1}{t_{k,j}} \left(t_{k+1,2j}|10\rangle + t_{k+1,2j+1}|01\rangle\right). \tag{4.31}$$

After the last layer of RBS-gates in Figure 4.5, the d qubits are in the state

$$\sum_{i\in[d]} t_{n,i}|e_i\rangle = \sum_{i\in[d]} |\psi_i|\,|e_i\rangle . \tag{4.32}$$

Finally, the last layer of $\mathsf{R_Z}$-gates adds the right phases to each amplitude, giving the state $|\psi_{unary}\rangle$.

If instead of an all-to-all topology, we have access only to a grid topology (for example, in the LAQCC model), we need to slightly change the circuit and take into account a significantly larger circuit depth, namely $\mathcal{O}(N)$. See Figure 4.6.

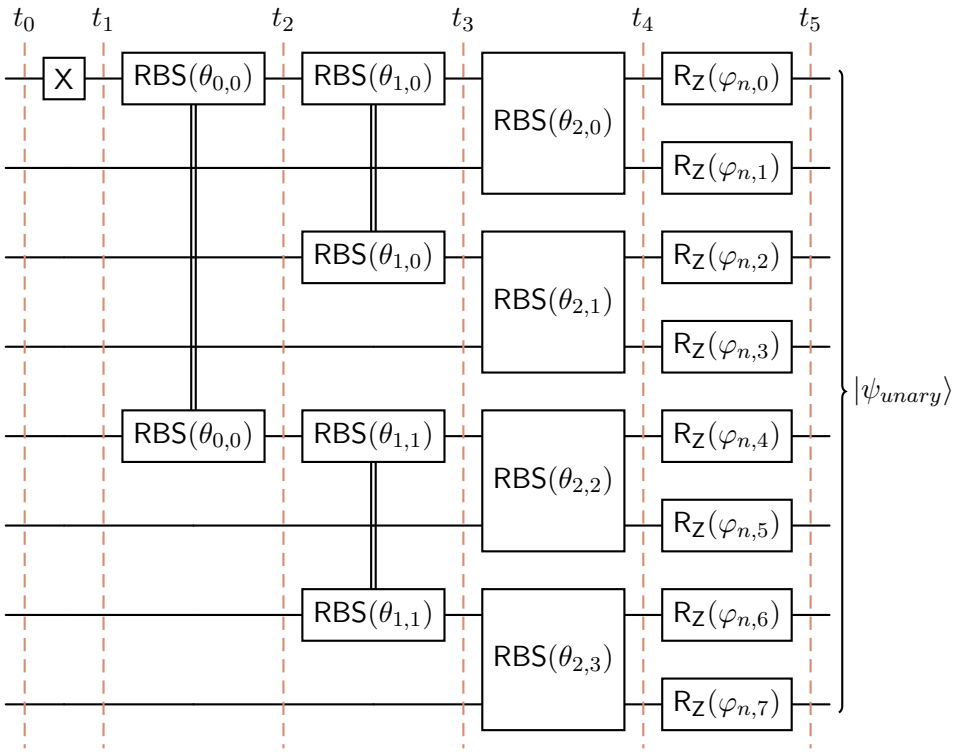


Figure 4.5: The data loader procedure preparing $|\psi_{unary}\rangle$ for $d = 8$, assuming all-to-all qubit connectivity.

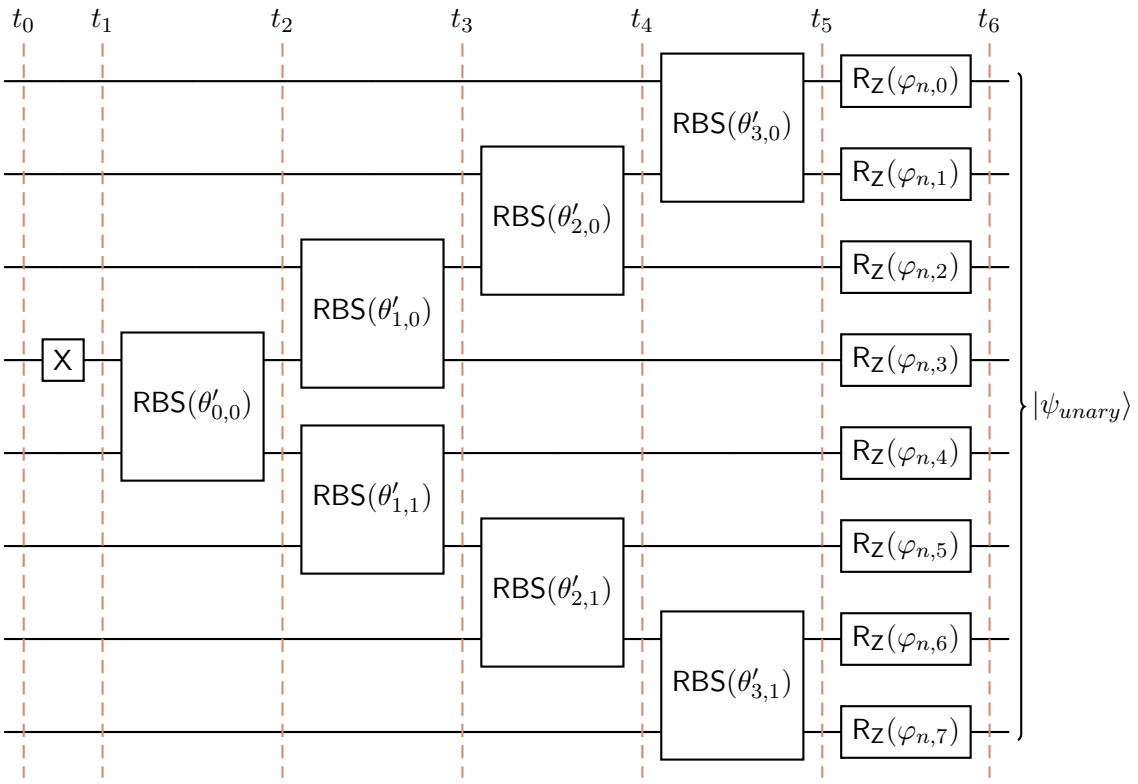


Figure 4.6: The data loader procedure preparing $|\psi_{unary}\rangle$ for $d = 8$, assuming 1D grid qubit connectivity.

To implement this circuit, we slightly adapt the parameters of the RBS-gates to be in line with the structure of the adapted circuit. In particular, we let T' be a tree of depth $N/2+1$, where each non-root layer consists of 2 nodes.

The value at the root node is defined as $t'_{0,0} = 1$; the values for the other $t'_{k,j}$ are defined as follows:

$$t'_{N/2,0} := |\psi_0|\,, \qquad t'_{N/2,1} := |\psi_{N-1}| \tag{4.33}$$

$$t'_{k,0} := \sqrt{(t'_{k+1,0})^2 + \left|\psi_{N/2-k}\right|^2}, \qquad t'_{k,1} := \sqrt{(t'_{k+1,1})^2 + \left|\psi_{N/2+k}\right|^2} \qquad (1 \le k \le N/2-1) \tag{4.34}$$

The corresponding angles $\theta'_{k,j}$, two for each layer in the tree, are defined similar to the regular θ:

$$\theta'_{0,0} := \theta_{0,0} = \arccos(t_{1,0}) \tag{4.35}$$

$$\theta'_{k,0} = \arcsin\left(\frac{t'_{k+1,j}}{t'_{k,j}}\right), \qquad \theta'_{k,1} = \arccos\left(\frac{t'_{k+1,j}}{t'_{k,j}}\right) \tag{4.36}$$

4.3.2 Converting to a binary encoding

The next lemma shows how to transform the state $|\psi_{unary}\rangle$ into the state $|\psi\rangle$.

Lemma 14 (Adapted from Lemma 4.3 and 4.4 from [Buh+23])**.** *There exists a procedure that performs the transformation*

$$|\psi_{unary}\rangle = \sum_{i\in[N]} \psi_i |e_i\rangle |0\rangle^{\otimes n} \to \sum_{i\in[N]} |0\rangle^{\otimes N} \psi_i |q_i\rangle = |\psi\rangle\,. \tag{4.37}$$

Proof. We organize the qubits into an N-qubit unary register $\mathtt{u}$ and an n-qubit binary register $\mathtt{b}$. The circuit implementing Equation (4.37) consists of two parts. The first part implements the transformation

$$\sum_{i\in[N]} \psi_i |e_i\rangle_{\mathtt{u}} |0\rangle_{\mathtt{b}}^{\otimes n} \to \sum_{i\in[N]} |e_i\rangle_{\mathtt{u}} \psi_i |q_i\rangle_{\mathtt{b}}\,. \tag{4.38}$$

In particular, we would like to transform the binary register from $|0\rangle$ into $|q_i\rangle$, conditioned on the i-th qubit in the unary register being in the $|1\rangle$-state. The Fanout-gates needed to implement these gates all pairwise commute, and they are mutually diagonalized by the n-qubit Hadamard-gate. We can thus implement the Transformation (4.38) by applying Theorem 2, with $\mathsf{T} = \mathsf{H}^{\otimes n}$ and D being a single layer with Z-gates on the qubits corresponding to ones in the binary representation of q_i.

The second part of the transformation uncomputes the unary register, i.e., implements the transformation

$$\sum_{i\in[N]} |e_i\rangle \psi_i |q_i\rangle \to \sum_{i\in[N]} |0\rangle \psi_i |q_i\rangle = |\psi\rangle \tag{4.39}$$

It consists of the following operations:

$$\begin{aligned} \sum_{i\in[N]} \psi_i |e_i\rangle |q_i\rangle |0\rangle^{\otimes(N-1)n} &\to \sum_{i\in[N]} \psi_i |e_i\rangle |q_i\rangle^{\otimes N} \\ &\to \sum_{i\in[N]} \psi_i |0\rangle^{\otimes N} |q_i\rangle_{\mathtt{ba}}^{\otimes N} \\ &\to \sum_{i\in[N]} \psi_i |0\rangle^{\otimes N} |q_i\rangle |0\rangle^{\otimes(N-1)n} = |\psi\rangle \end{aligned} \tag{4.40}$$

The first step uses Fanout-gates to copy the binary register $N-1$ times using an ancilla register $\mathtt{a}$. The second step applies $\mathsf{EQ}_{q_i}^{(n+1)}$-gates on the i-th qubit in the unary register and the i-th copy of the binary register. The last step uncopies the binary register, leaving us with the state $|\psi\rangle$, as desired. □

Complexity

Given all-to-all connectivity, the circuit depth of the data loader procedure is $\mathcal{O}(n)$, while the depth of implementing Lemma 14 is dominated by that of the Fanout-gates and EQ-gates, giving a circuit depth of $\mathcal{O}(\log(N)+\log(n)) = \mathcal{O}(n)$. As such, the total circuit depth for the unary-based QSP protocol is $\mathcal{O}(n)$. The circuit also requires a circuit width of $\mathcal{O}(Nn)$. When constrained to a grid topology (for example, in the LAQCC setting), we see a total circuit depth of $\mathcal{O}(N)$, following the data loader procedure shown in Figure 4.6. The circuit width in this case is $\mathcal{O}(Nn\log(n))$.

Both the data loader procedure and Lemma 14 can be easily transformed into a sparse version. Given all-to-all connectivity, the data loader procedure can be done in $\mathcal{O}(\log(d))$ depth, while Lemma 14 requires a circuit depth of $\mathcal{O}(\log(n)) + \mathcal{O}(\log(d)) \in \mathcal{O}(\log(dn))$. This brings the total circuit depth to $\mathcal{O}(\log(dn))$, and the total circuit width to $\mathcal{O}(dn)$. Note that this circuit depth is optimal by Theorem 5. In the LAQCC setting, due to the grid topology, we obtain a possibly *worse* circuit depth of $\mathcal{O}(d)$, and a circuit width of $\mathcal{O}(dn\log(n))$.

Chapter 5

Assessing adaptive advantage in QSP protocols

The main reason for considering shallow-depth quantum circuits is to minimize the number of idling qubits, thereby reducing the likelihood of decoherence during computation. As we have seen in the previous chapters, LAQCC circuits can reduce the depth of circuits significantly by performing part of the computation on a classical computer. This off-loading usually comes at the cost of extra quantum gates and extra ancilla qubits.

An example of this is the implementation of the Fanout-gate. The standard approach with all-to-all connectivity (see Figure 2.5b) has a circuit depth of $\mathcal{O}(\log(n))$, while the LAQCC approach (see Figure 3.4) has constant depth, making for a total circuit size of $\mathcal{O}(n\log(n))$ and $\mathcal{O}(n)$ respectively. At the same time, the circuit using all-to-all connectivity contains $n-1$ CNOT-gates, while the LAQCC circuit uses $3n-2$ CNOT-gates, as well as single-qubit gates, measurements, and extra classical computations. In other words, there exists a trade-off between having a smaller circuit size and using less qubits and quantum gates. A natural question to ask is then: how should we weigh this trade-off?

In [Neu25], Neumann compared the theoretical success probabilities of state preparation schemes for the GHZ-state and the W-state in the LAQCC setting, and compared it to established state preparation approaches assuming all-to-all qubit connectivity or linear nearest-neighbor connectivity. He found that, relative to a worst-case error model and conditioned on some assumptions, the LAQCC protocols can have an exponentially higher success probability these standard state preparation approaches.

In this chapter, we will expand on this line of research by exploring the theoretical success probability of the QSP protocols discussed in Chapter 4, and determining the conditions under which the LAQCC protocols outperform the non-adaptive protocols. In Section 5.1 we discuss the error model from [Neu25], show how it applies to the success probability of some simple gates, and how it can be used to compare gate implementations. In Sections 5.2 and 5.3, we calculate the success probabilities for the FO-gate and the OR-gate, and compare the LAQCC approach to an all-to-all, 1D grid and 2D grid approach. In Section 5.4, we calculate the success probabilities of the AND-, EQ-, and PERM-gates. In Section 5.5 and 5.5.4, we calculate the success probabilities for the UCG-based and for the unary-based QSP protocols, and determine under which conditions the LAQCC approach performs best.

5.1 Analysis

5.1.1 Error model

We proceed to describe the error model from [Neu25]. It is an error model in which every error corresponds to an independent uniformly random gate being applied to the qubit(s). An error can happen both while a qubit is idling, while a gate is applied to the qubit, and when measuring a qubit. Since every error corresponds to an independent uniformly random gate, the probability that two errors cancel each other

is 0. Naturally, if we know the probability that an error occurs during some event is p, then a success probability for the same event is given by $1-p$. We define the *success probability* of a quantum circuit as the probability that no error occurs during any part of the circuit, that is, the probability that there are no gate errors, no measurement errors, and no idling errors.

We determine this success probability by multiplying the success probabilities of all elementary operations used in the circuit. The elementary operations are one-qubit gates, CNOT-gates, measurements and intermediate computations (used in the LAQCC circuits). We also determine the idling terms for qubits to which no operation is applied during one of the layers of the circuit.

Notation. *We use the following abbreviations to denote probabilities of elementary operations:*

- p_s *: the probability that a single-qubit gate succeeds*
- p_d *: the probability that a* CNOT*-gate succeeds*
- p_m *: the probability that a measurement succeeds (i.e. returns the correct value)*
- p_c *: the probability that an intermediate classical computation succeeds (i.e. returns the correct value)*
- p_{i_s} *: the probability that a qubit remains coherent, while idling during a single-qubit gate.*
- p_{i_d} *: the probability that a qubit remains coherent, while idling during a two-qubit gate.*
- p_{i_m} *: the probability that a qubit remains coherent, while idling during a measurement.*
- p_{i_c} *: the probability that a qubit remains coherent, while idling during an intermediate classical computation.*

We denote the success probability of a circuit C by $P(C)$, and the success probability of a qubit staying coherent while idling during the same circuit by $P_i(C)$. If necessary, the qubit topology or type of implementation used is added for clarity. For example, $P(\mathsf{FO}^{(n)}, \mathsf{LAQCC})$ refers to the LAQCC *implementation of the* $\mathsf{FO}^{(n)}$*-gate.*

To express the total number of gates, measurements, and idling terms in some circuit C, we will use the notation $|C|_x$, where $x \in \{s, d, m, i_s, i_d, i_m, i_c\}$ refers to the type of operations or idling terms. Similarly, $|i, C|_x$ refers to the number of idling terms for a qubit that is idling during circuit C.

In the following, we will always assume that $p_c = 1$, i.e., that intermediate classical computations will always return the same answer given the same measurement outcomes.

Using this error model and corresponding notation, we can start to obtain success probabilities of some simple gates. For example, a controlled single-qubit gate CU, that is implemented via the circuit in Lemma 3, has a success probability of

$$P(\mathsf{CU}) = (p_s)^3 (p_d)^2 (p_{i_s})^3. \tag{5.1}$$

In the $|C|_x$-notation, we have $|\mathsf{CU}|_s = 3$, $|\mathsf{CU}|_d = 2$, and $|\mathsf{CU}|_{i_s} = 3$. At the same time, the success probability of a single qubit staying coherent while idling during this circuit is

$$P_i(\mathsf{CU}) = (p_{i_d})^2 (p_{i_s})^3. \tag{5.2}$$

In the remainder of this chapter, we will determine success probabilities for state preparation schemes, that will depend on the input size n and at most seven probability variables. To get a first-order estimate of the difference in success probability between the different settings, we adopt the assumptions of [Neu25] on the magnitude of said variables.

Firstly, in current quantum hardware, the gate times and error rates for one-qubit gates are significantly lower than for two-qubit gates and measurements. Thus, we will assume that the errors occurring during single-qubit gates negligibly influence the total success probability. In practice, this means that we assume that $p_s = p_{i_s} = 1$.

Secondly, we assume that $p_d \approx p_m$ and $p_{i_d} \approx p_{i_m} \approx p_{i_c}$, which is supported by the error rates of these operations observed in current quantum hardware, for example by IBM [Ibm].

When comparing success probabilities, we use approximate equality ($\approx$) and inequality ($\gtrsim$ and $\lesssim$) signs to indicate that these assumptions on the success probability are applied.

5.1.2 Example comparison

We will now describe how we use this error model and the added assumptions to derive an expression comparing LAQCC circuits with non-adaptive counterparts, by means of an example. Suppose we want to determine whether

$$P(\mathsf{U}, \mathsf{LAQCC}) \geq P(\mathsf{U}, \text{non-adaptive}), \tag{5.3}$$

for some gate U with both a LAQCC implementation and a non-adaptive implementation. Suppose that

$$\begin{aligned} P(\mathsf{U}, \mathsf{LAQCC}) &= (p_s)^{a_0}(p_d)^{a_1}(p_m)^{a_2}(p_{i_s})^{a_3}(p_{i_d})^{a_4}(p_{i_c})^{a_5}(p_{i_m})^{a_6} \\ P(\mathsf{U}, \text{non-adaptive}) &= (p_s)^{b_0}(p_d)^{b_1}(p_m)^{b_2}(p_{i_s})^{b_3}(p_{i_d})^{b_4}(p_{i_c})^{b_5}(p_{i_m})^{b_6}. \end{aligned} \tag{5.4}$$

Applying the assumptions from the previous section reduces this inequality to

$$(p_d)^{a_1+a_2}(p_{i_d})^{a_4+a_5+a_6} \gtrsim (p_d)^{b_1+b_2}(p_{i_d})^{b_4+b_5+b_6}. \tag{5.5}$$

Note that there are only two success probability terms left. After bundling similar terms, we have

$$(p_d)^{a_1+a_2-(b_1+b_2)} \gtrsim (p_{i_d})^{b_4+b_5+b_6-(a_4+a_5+a_6)}, \tag{5.6}$$

If the circuit on the left-hand side of Equation (5.3) is an LAQCC-circuit, then it will usually have at least as many two-qubit gates and measurements as the non-adaptive circuit on the right-hand side. In that case, $a_1 + a_2 - (b_1 + b_2) \geq 0$, and we get that Equation (5.5) holds precisely when

$$(p_d) \gtrsim (p_{i_d})^{\gamma_1/\gamma_2}. \tag{5.7}$$

where

$$\gamma_1 := b_4 + b_5 + b_6 - (a_4 + a_5 + a_6), \qquad \gamma_2 := a_1 + a_2 - (b_1 + b_2).$$

Let $\gamma := \gamma_1/\gamma_2$. We conclude that

$$P(\mathsf{U}, \mathsf{LAQCC}) \gtrsim P(\mathsf{U}, \text{non-adaptive}) \iff (p_d) \gtrsim (p_{i_d})^{\gamma}. \tag{5.8}$$

In the $|C|_x$-notation, γ can be written as

$$\gamma = \frac{\gamma_1}{\gamma_2} = \frac{|\mathsf{U}, \text{non-adaptive}|_{i_d + i_m + i_c} - |\mathsf{U}, \mathsf{LAQCC}|_{i_d + i_m + i_c}}{|\mathsf{U}, \mathsf{LAQCC}|_{d+m} - |\mathsf{U}, \text{non-adaptive}|_{d+m}}. \tag{5.9}$$

Note that γ_1 expresses the difference in idling terms between the two circuits, while γ_2 expresses the difference in gates. Dividing the two thus expresses how much larger the difference in idling terms is compared to the difference in gates. Intuitively, as this fraction γ increases, then a comparison between a LAQCC circuit and a non-adaptive circuit should increasingly work in favor of the LAQCC circuit.

Well, this is precisely what Equation (5.8) implies! Namely, it states that the success probability of the LAQCC circuit is greater than that of the non-adaptive circuit if the probability that a two-qubit gate succeeds (i.e., p_d) is higher than the probability that a qubit that idles during γ two-qubit gates stays coherent (i.e., $(p_{i_d})^{\gamma}$). As γ increases, $(p_{i_d})^{\gamma}$ becomes smaller, hence the condition on p_d becomes *less* strict, and we can be *more* confident that the LAQCC prevails over its non-adaptive counterpart. In this sense, γ is a measure of *adaptive advantage*.

When considering gates that may affect a variable number of qubits n, the resulting γ is a function that depends on n. In case of sparse QSP, it may also depend on the sparsity d of the vector we are trying to prepare. In this case, we may call γ the *adaptive advantage function*.

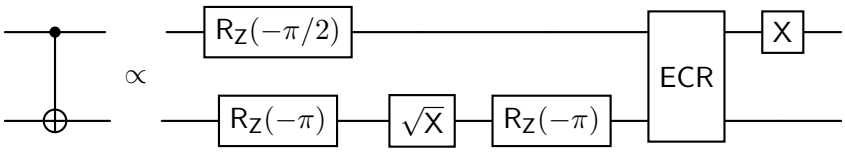


Figure 5.1: Converting a CNOT-gate (up to a global phase of $\frac{\pi}{2}$) into an ECR-gate surrounded by single-qubit rotations.

Note that, if Equation (5.7) holds, then in fact, the difference in success probability increases *exponentially* in n. To see this, we can rewrite Equation (5.5) to get

$$P(\mathsf{U}, \mathsf{LAQCC}) \gtrsim (p_d)^{\gamma_1(n)} (p_{i_d})^{-\gamma_2(n)} P(\mathsf{U}, \text{non-adaptive}). \tag{5.10}$$

For any $\varepsilon > 0$, filling in $p_d = (1+\varepsilon)(p_{i_d})^{\gamma(n)}$ into Equation (5.10) gives

$$\begin{aligned} P(\mathsf{U}, \mathsf{LAQCC}) &\gtrsim ((1+\varepsilon)(p_{i_d})^{\gamma(n)})^{\gamma_2(n)} (p_{i_d})^{-\gamma_1(n)} P(\mathsf{U}, \text{non-adaptive}) \\ &= ((1+\varepsilon)(p_{i_d})^{\frac{\gamma_1(n)}{\gamma_2(n)}})^{\gamma_2(n)} (p_{i_d})^{-\gamma_1(n)} P(\mathsf{U}, \text{non-adaptive}) \\ &= (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{U}, \text{non-adaptive}). \end{aligned} \tag{5.11}$$

So, indeed, if Equation (5.7) holds, then the LAQCC circuit performs exponentially better than its non-adaptive counterpart.

Note that if $0 \leq \gamma \leq 1$, then the difference in idling time is equal to or smaller than the difference in gate and measurement operations. In order for Equation (5.7) to hold then, it would be necessary that $p_d \geq p_{i_d}$, i.e. that letting qubits idle during a two-qubit gate leads to more errors than manipulating it using the same gate. In this case, adaptive advantage seems unlikely. Furthermore, $\gamma < 0$ corresponds to the condition that $p_d > 1$, which is, of course, impossible.

5.1.3 Estimating γ for current QPUs

To get a taste for which order of magnitude γ may have, we estimate this value for some current quantum hardware. For starters, Table 5.1 lists some QPUs from IBM, the success probability of single-qubit gates, the double-qubit gate that is native to the specific hardware, and double-qubit gate and decoherence times.

QPU	Processor type	p_s	$p_{d,native}$	T_1	T_2	gate time
IBM Brisbane	Eagle r3	$1 - 2.608 \cdot 10^{-4}$	$1 - 8.166 \cdot 10^{-3}$	231.16 µs	149.51 µs	660 ns
IBM Fez	Heron r2	$1 - 2.270 \cdot 10^{-4}$	$1 - 3.792 \cdot 10^{-3}$	145.91 µs	80.31 µs	68 ns
IBM Strasbourg	Eagle r3	$1 - 2.241 \cdot 10^{-4}$	$1 - 8.886 \cdot 10^{-3}$	303.18 µs	159.47 µs	660 ns
IBM Torino	Heron r1	$1 - 3.189 \cdot 10^{-4}$	$1 - 3.954 \cdot 10^{-3}$	189.9 µs	140.82 µs	84 ns

Table 5.1: Median success probability data for some current QPUs, obtained from the IBM Quantum Platform [Ibm] (on February 19, 2025, 17:25). For the Eagle-type processors, the native two-qubit gate is the ECR-gate, while for the Heron-type processors, it is the CZ-gate. We assume a single-qubit gate time of 30 ns. The two-qubit gate times correspond to their native gates.

The Eagle-type processors (used in IBM Brisbane and IBM Strasbourg) have the ECR-gate as their native double-qubit gate. It relates to a CNOT-gate by the circuit in Figure 5.1. Both Eagle processors can apply R_Z-gates with zero error [Abu24], and thus we estimate the success probability of a CNOT-gate using this circuit as

$$p_{d,Eagle} = (p_{d,native})(p_s)^2 (p_{i_s})^3. \tag{5.12}$$

The Heron-type processors (used in IBM Fez and IBM Torino) have the CZ-gate as their native double-qubit gate. It relates to a CNOT-gate by conjugating the target qubit with Hadamard-gates. This gives an estimate of the success probability of a CNOT-gate of

$$p_{d,Heron} = (p_{d,native})(p_s)^2 (p_{i_s})^2. \tag{5.13}$$

To estimate the idling terms, we use the decoherence times T_1 and T_2, combined with the gate times of a CNOT-gate. T_1 and T_2 are both decay constants describing how long a quantum state will stay intact. Specifically, T_1 refers to the expected time it takes for a qubit that is initially in the $|1\rangle$-state to evolve into an equal classical probabilistic mixture of states $|1\rangle$ and $|0\rangle$, while T_2 refers to the time it takes for a qubit that is initially in the $|+\rangle$-state to evolve into an equal classical probabilistic mixture of states $|+\rangle$ and $|-\rangle$. The probability that a qubit will stay in state $|1\rangle$ (resp. $|+\rangle$) after some time t is given by the formula e^{-t/T_1} (resp. e^{-t/T_2}). Since T_2 is usually lower than T_1, we will use T_2 to get a lower bound on the idling success probabilities.

Finally, we estimate γ by the fraction $\gamma = \log(p_d) \,/\, \log(p_{i_d})$, corresponding to $p_d = (p_{i_d})^\gamma$. Table 5.2 lists the results of these estimations. In the following analyses, we will obtain asymptotic expressions for γ

QPU	$p_d = p_{d,\mathsf{CNOT}}$	p_{i_d}	γ
IBM Brisbane	$1 - 9.280 \cdot 10^{-3}$	$1 - 4.804 \cdot 10^{-3}$	1.936
IBM Fez	$1 - 4.988 \cdot 10^{-3}$	$1 - 1.965 \cdot 10^{-3}$	2.542
IBM Strasbourg	$1 - 9.889 \cdot 10^{-3}$	$1 - 4.505 \cdot 10^{-3}$	2.201
IBM Torino	$1 - 5.013 \cdot 10^{-3}$	$1 - 1.235 \cdot 10^{-3}$	4.067

Table 5.2: Estimated values for γ for some QPUs.

when comparing gate implementations across topologies. These give an idea for how quickly the adaptive advantage grows as the number of qubits increases. Additionally, plotting the same function – including the reference values in Table 5.2 – allows us to inspect when an adaptive advantage is likely.

5.2 Fanout-gate

In this section, we will discuss the success probability of four implementations of the Fanout-gate:

(i) a non-adaptive protocol assuming all-to-all connectivity;

(ii) a non-adaptive protocol assuming 1D grid connectivity;

(iii) a (simplified) non-adaptive protocol with 2D grid connectivity; and

(iv) a LAQCC protocol.

After obtaining the success probabilities, we proceed by giving estimates on when the LAQCC protocol (exponentially) outperforms the standard approaches, by rewriting the obtained expressions for the success probability using the simplifications discussed in Section 5.1, and finding appropriate values for $\gamma(n)$.

5.2.1 All-to-all approach

Having all-to-all qubit connectivity allows us to implement the Fanout-gate as in Figure 2.5b. As per the discussion in Section 2.7.2, the circuit consists of $\lceil \log(n) \rceil$ layers. The i-th layer, where $0 \leq i \leq \lceil \log(n) \rceil - 2$, consists of 2^i parallel CNOT-gates, and the last layer consists of $n - 2^{\lceil \log(n) \rceil - 1}$ gates. That means that in the first $\lceil \log(n) \rceil - 1$ layers, there are $n - 2^{i+1}$ qubits that should stay coherent while idling during the execution of the CNOT-gates, while in the last layer there are $2^{\lceil \log(n) \rceil} - n$ qubits idling. As such, the total number qubits idling during these CNOT-gates is

$$\begin{aligned} |\mathsf{FO}^{(n)}, \text{all-to-all}|_{i_d} &= \left(\sum_{i=0}^{\lceil \log(n) \rceil - 2} n - 2 \cdot 2^i \right) + \left(2^{\lceil \log(n) \rceil} - n \right) \\ &= \left(n(\lceil \log(n) \rceil - 1) - (2^{\lceil \log(n) \rceil} - 2) \right) + (2^{\lceil \log(n) \rceil} - n) \\ &= n \lceil \log(n) \rceil - 2n + 2. \end{aligned} \tag{5.14}$$

Furthermore, the number of two-qubit gates is precisely

$$|\mathsf{FO}^{(n)}, \text{all-to-all}|_{\mathrm{d}} = n - 1. \tag{5.15}$$

Since there are no single-qubit gates and measurements, the success probability of this Fanout-gate implementation is

$$P(\mathsf{FO}^{(n)}, \text{all-to-all}) = (p_d)^{n-1}(p_{i_d})^{n\lceil\log(n)\rceil-2n+2} \tag{5.16}$$

At the same time, a qubit that must stay idle during this Fanout-gate has an idling term count of

$$|i, \mathsf{FO}^{(n)}, \text{2D grid}|_{i_d} = \lceil\log(n)\rceil, \tag{5.17}$$

and so a success probability of staying coherent of

$$P_i(\mathsf{FO}^{(n)}, \text{all-to-all}) = (p_{i_d})^{\lceil\log(n)\rceil} \tag{5.18}$$

5.2.2 1D grid approach

When assuming that qubits are connected via a 1D grid, we can implement a Fanout-gate as in Figure 2.5a. The circuit consists of $n-1$ layers, and the gate and idling term count are simply

$$|\mathsf{FO}^{(n)}, \text{1D grid}|_d = n-1 \tag{5.19}$$

and

$$|\mathsf{FO}^{(n)}, \text{1D grid}|_{i_d} = (n-1)(n-2) = n^2 - 3n + 2. \tag{5.20}$$

This gives a success probability of

$$P(\mathsf{FO}^{(n)}, \text{1D grid}) = (p_d)^{n-1}(p_{i_d})^{n^2-3n+2} \tag{5.21}$$

and a success probability for idling of

$$P_i(\mathsf{FO}^{(n)}, \text{1D grid}) = (p_{i_d})^{n-1} \tag{5.22}$$

5.2.3 2D grid approach

In [Ros13], Rosenbaum constructed circuits for multi-qubit Fanout and Toffoli-gates under a kD grid topology, where $k \geq 1$. He also proved that these have asymptotically optimal depth, namely, $\Theta(\sqrt[k]{n})$ for a Fanout-gate of arity n.

For $k = 2$, his Fanout-gate circuit works by assigning one qubit as a 'center' qubit (namely, the control qubit of the Fanout-gate), and then fanning out that qubit to the ring of qubits surrounding it. The qubits in this ring can then be used as control qubits for the Fanout-gates to apply to the next ring, and so on, until the number of desired target qubits is reached. For a gate of arity n, this protocol requires a grid of size $m \times m$ where $m = \mathcal{O}(\sqrt{n})$, and similar circuit depth, since the fanning out from one ring to the next can be done in constant depth.

Inspired by this approach, we will analyse a simpler version of this circuit, that fans out the control qubit directly to the closest grid of qubits surrounding it, without leaving open spaces with ancillas.[1] See Figure 5.2 for an example of a $\mathsf{FO}^{(25)}$-gate, applied to 5×5 qubits organized in a 2D grid.

A Fanout-gate of arity n implemented in this manner requires a grid size of at least $m \times m$, where $m \geq \sqrt{n}$. It consists of $(m-1)/2$ 'steps'. The first step of the circuit consists of 8 CNOT-gates, which fan out the control qubit in the center to all qubits of the ring directly surrounding it. These can be partially parallelized, giving a gate count of

$$|\mathsf{FO}^{(n)}, \text{2D grid, step 0}|_d = 8 \tag{5.23}$$

and an idling term count of

$$|\mathsf{FO}^{(n)}, \text{2D grid, step 0}|_{i_d} = (n-2) + (n-4) + (n-6) + (n-4) = 4n - 16. \tag{5.24}$$

[1] These extra spaces were necessary in Rosenbaum's implementation of the AND-gate, but not in the implementation of the Fanout-gate.

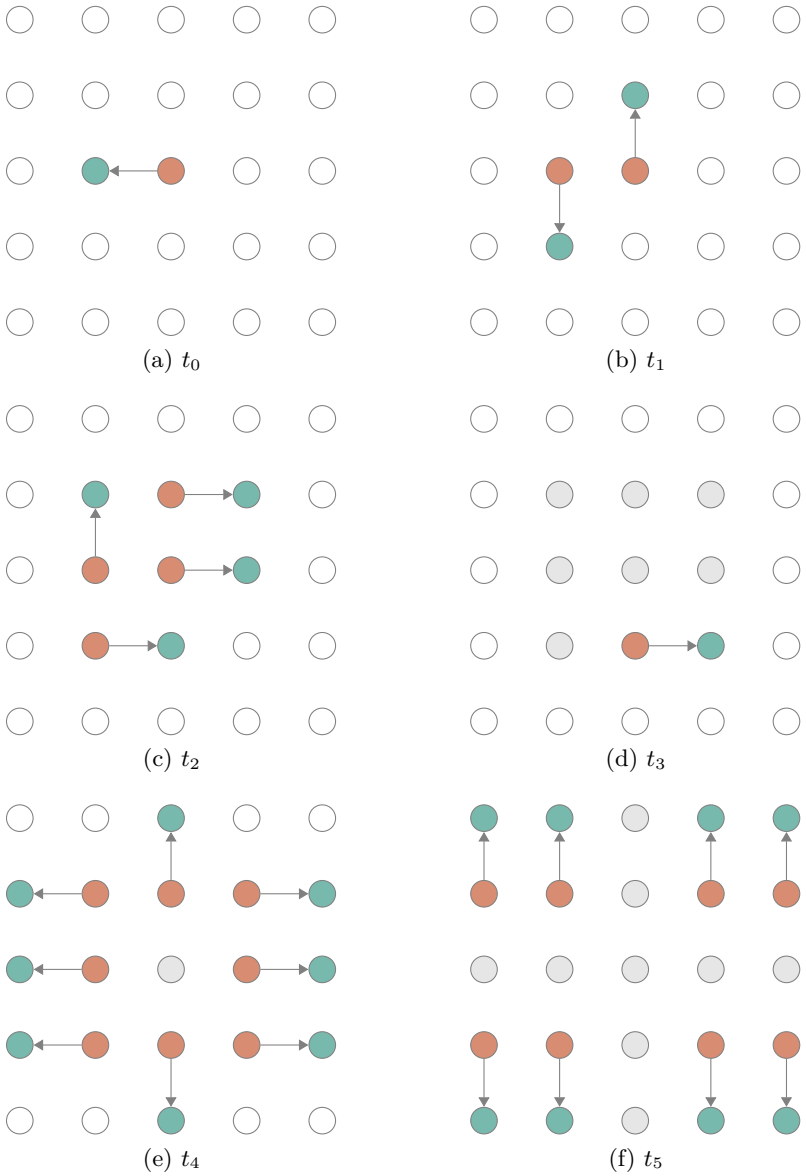


Figure 5.2: An implementation of an $\mathsf{FO}^{(n)}$-gate where $n = 25$, constrained to a 2D grid qubit topology. The arrows indicate CNOT-gates; the qubits colored red and green represent control and target qubits respectively, while a light gray color indicates the qubit has already been targeted in previous layers. Figure (a)-(d) visualize 'step 0' in the circuit, consisting 8 CNOT-gates to fan out the initial control qubit (in the middle of the grid) to the qubits in the ring immediately surrounding it – see Equation (5.23) and (5.24). Figure (e)-(f) consist of 'step 1' – see Equations (5.25) and (5.26). The circuit is based on [Ros13].

The newly targeted qubits are used as control qubits in the next step. In this step, the qubits are fanned out to the ring surrounding the current control ring, by first applying a CNOT-gate to all control qubits and one of their direct neighbors in the next ring, and then using another 8 CNOT-gates to reach the remaining qubits in the corners.[2]

The k-th step uses the ring of size $(1+2k)\times(1+2k)$ as its control qubits and fans out to the next ring, which is of size $(1+2(k+1))\times(1+2(k+1)) = (3+2k)\times(3+2k)$. During step $k \geq 1$, there are $4(1+2k)-4 = 8k$ control qubits, and so the gate count becomes

$$|\mathsf{FO}^{(n)}, \text{2D grid, step } k|_d = 8k + 8, \tag{5.25}$$

while the idling term count becomes

$$|\mathsf{FO}^{(n)}, \text{2D grid, step } k|_{i_d} = (n - 8k) + (n - 2 \cdot 8) = 2n + 8k - 16. \tag{5.26}$$

This procedure is continued until the number of desired targets is reached. Given k steps of this circuit, a Fanout-gate with arity at most $(1+2k)^2$ can be constructed.

If n is not a perfectly square integer, not all qubits in the outermost ring have to be targeted. Rather, the number of qubits that still need to be targeted during the last iteration is $n-(m-2)^2$. During these last CNOT-gates, $n-2(n-(m-2)^2)$ qubits are idling. The total gate count for the circuit becomes

$$|\mathsf{FO}^{(n)}, \text{2D grid}|_{\mathrm{d}} = n - 1, \tag{5.27}$$

and the idling term count becomes

$$\begin{aligned}
|\mathsf{FO}^{(n)}, \text{2D grid}|_{i_d} &= |\mathsf{FO}^{(n)}, \text{2D grid, step } 0|_{i_d} + \sum_{k=1}^{\frac{m-1}{2}-1} |\mathsf{FO}^{(n)}, \text{2D grid, step } k|_{i_d} \\
&= 4n - 16 + \sum_{k=1}^{\frac{m-1}{2}-1} (2n + 8k - 16) \\
&= 4n - 16 + \left(\frac{m-1}{2} - 1\right)(2n-16) + 4\left(\frac{m-1}{2} - 1\right)\left(\frac{m-1}{2}\right) \\
&= m^2 + mn - 12m + n + 11 \\
&\geq n\sqrt{n} + 2n - 12\sqrt{n} + 11,
\end{aligned} \tag{5.28}$$

where in the third line we used the identity

$$\sum_{i=1}^{n} k = \frac{n(n+1)}{2} \tag{5.29}$$

and in the fifth line we used $m \geq \sqrt{n}$. Equation (5.27) and (5.28) yield a success probability for the 2D grid Fanout-gate:

$$P(\mathsf{FO}^{(n)}, \text{2D grid}) \leq (p_d)^{n-1}(p_{i_d})^{n\sqrt{n}+2n-12\sqrt{n}+11}. \tag{5.30}$$

At the same time, a qubit that must stay idle during this Fanout-gate has an idling term count of

$$|i, \mathsf{FO}^{(n)}, \text{2D grid}|_{i_d} = 4 + 2\left(\frac{m-1}{2} - 1\right) = 4 + (m-1) - 2 \geq \sqrt{n} + 1, \tag{5.31}$$

and so a success probability of staying coherent of

$$P_i(\mathsf{FO}^{(n)}, \text{2D grid}) \leq (p_{i_d})^{\sqrt{n}+1}. \tag{5.32}$$

[2] In practice, this circuit can be optimized further by letting loose of the requirement for a grid, and leaving the target qubits in a 'diamond' shape, by skipping the second layer of each step that targets the corner qubits.

Asymptotically, the depth of this Fanout-gate implementation is optimal under a 2D grid topology constraint, as proven by [Ros13]. However, the final topology of the qubits is suboptimal, in fact, the target qubits in the inner grids are not directly accessible to qubits that were not part of the Fanout-gate. Subsequent procedures that use these gates, such as the OR-reduction or the parallelized UCG, assume that, after this gate implementation, the output qubits are placed on a *line* starting from the control qubits.

To achieve this with a 2D grid topology, a SWAP-circuit can put the qubits in the right place. An alternative would be to relax the 2D grid topology constraint to a *quasi-2D* grid topology, which is a three-dimensional grid in which the third dimension is of constant length (independent of n). The qubits in the third dimension can then directly interact with all qubits resulting from the Fanout-gate, and move them to the appropriate places.

For the purposes of our analysis, we use the success probability in Equation (5.30), and treat it as an *upper bound* on the success probability of a Fanout-gate under a 2D grid topology that places the target qubits on a line that starts from the control qubit. This way, when we later compare this version of the Fanout-gate to the LAQCC approach, we are actually comparing it to a 'steelmanned' version of the 2D grid topology. If the LAQCC approach comes out best in the approach, this gives more confidence to this result.

5.2.4 LAQCC approach

For the LAQCC implementation of the Fanout-gate, we use the circuit discussed in Section 3.4.2, which is shown in Figure 3.4 for $n = 4$. To implement a Fanout-gate for general n, this method uses a total of $3n-1$ qubits, of which $2n-1$ are needed for the preparation of the GHZ-state. After the parity measurements, at most $\lceil \frac{n-1}{2} \rceil$ qubits must be corrected. For these calculations, we assume that this maximum is always reached. Since $p_s \leq p_{i_s}$, this yields a lower bound on the success probability of the circuit. This also means that the other $\lfloor \frac{n-1}{2} \rfloor$ qubits that form the GHZ-state are assumed to be idling during these corrections. How this affects the single-qubit gate and idling terms can be seen in Table A.1.

The total two-qubit gate count for this implementation is

$$|\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_d = 3n - 2, \tag{5.33}$$

and the number of measurements is

$$|\mathsf{FO}^{(m)}, \mathsf{LAQCC}|_d = 2n - 1. \tag{5.34}$$

The number of qubits idling during two-qubit gates, during the measurements and during the intermediate classical computations are

$$|\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_{i_d} = 2n + 1, \qquad |\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_{i_m} = |\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_{i_c} = 3n - 1 \tag{5.35}$$

We obtain a success probability of

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \geq (p_s)^{2n+\lceil (n-1)/2 \rceil} (p_d)^{3n-2} (p_m)^{2n-1} (p_{i_s})^{5n+\lfloor (n-1)/2 \rfloor - 2} (p_{i_d})^{2n+1} (p_{i_m})^{3n-1} (p_{i_c})^{3n-1} \tag{5.36}$$

and a success probability of a qubit staying coherent while idling during this procedure of

$$P_i(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) = (p_{i_s})^4 (p_{i_d})^3 (p_{i_m})^2 (p_{i_c})^2. \tag{5.37}$$

5.2.5 Comparison of Fanout-gate implementations

In this subsection we compare the success probabilities for the different implementations of the Fanout gate, and see under which circumstances the LAQCC protocol outperforms the non-adaptive protocols.

For these analyses, we apply the reductions specified in Section 5.1. In Table 5.3, we have summed the two-qubit gate and measurement count (corresponding to the assumption $p_d \approx p_m$), and combined the idling term count of qubits idling during a two-qubit gate, measurement or classical intermediate computation (corresponding to the assumption $p_{i_d} \approx p_{i_m} \approx p_{i_m}$).

The results from this section are summarized in Theorem 8.

	$d+m$	$i_d+i_m+i_c$
All-to-all	$n-1$	$n\lceil\log(n)\rceil-2n+2$
1D grid	$n-1$	n^2-3n+2
2D grid	$n-1$	$n\sqrt{n}+2n-12\sqrt{n}+11$
LAQCC	$5n-3$	$8n-1$

Table 5.3: Gate and idling term counts for different $\mathsf{FO}^{(n)}$-gate implementations, after applying the simplifying assumptions from Section 5.1.

Theorem 8. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n\lceil\log(n)\rceil - 10n + 3}{4n-2} \in \mathcal{O}(\log(n)), \tag{5.38}$$

then

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n\sqrt{n} - 5n - 16\sqrt{n} + 15}{4n-2} \in \mathcal{O}(\sqrt{n}), \tag{5.39}$$

then

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{2D grid}).$$

(iii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n^2 - 11n + 3}{4n-2} \in \mathcal{O}(n), \tag{5.40}$$

then

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{1D grid}).$$

When looking at Table 5.3, wee see that the LAQCC approach has asymptotically the same scaling when it comes to the gate terms, and better scaling when it comes to the idling terms, compared to the non-adaptive approaches. However, the constants associated with the highest-order terms are a bit higher for the LAQCC approach, owing to the more elaborate structure of the used circuit in terms of gates and number of qubits. As an illustration, the number of idling terms in the LAQCC approach falls below that of the all-to-all approach when

$$8n - 1 < n\lceil\log(n)\rceil - 2n + 2. \tag{5.41}$$

This happens when

$$n(10 - \lceil\log(n)\rceil) < 1. \tag{5.42}$$

Since $n > 0$, this happens when $\lceil\log(n)\rceil > 10$, that is, when $n > 2^{10} = 1024$. Even then, this slight advantage in idling terms comes at the cost of significantly more gate and measurement terms, meaning it might take many more qubits before an adaptive advantage can be expected. In the remainder of this section, we will see when this advantage occurs, and derive the results from Theorem 8.

Part (i): all-to-all approach versus LAQCC approach

Comparing the success probability of the all-to-all approach with the LAQCC approach and applying the assumptions on the success probabilities, we see that

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{FO}^{(n)}, \text{all-to-all}) \tag{5.43}$$

precisely when

$$(p_d)^{5n-3}(p_{i_d})^{8n-1} \gtrsim (p_d)^{n-1}(p_{i_d})^{n\lceil\log(n)\rceil-2n+2}. \tag{5.44}$$

Rearranging Equation (5.44), we get that Equation (5.43) holds when

$$(p_d)^{\gamma_2(n)} \gtrsim (p_{i_d})^{\gamma_1(n)}, \tag{5.45}$$

where

$$\gamma_2(n) = 4n-2 \in \mathcal{O}(n), \qquad \gamma_1(n) = n\lceil\log(n)\rceil - 10n + 3 \in \mathcal{O}(n\log(n)) \tag{5.46}$$

From Equation (5.46), we see that

$$\gamma(n) = \frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n\lceil\log(n)\rceil - 10n + 3}{4n-2} \in \mathcal{O}(\log(n)) \tag{5.47}$$

By the same method as in the example in Section 5.1, we can rewrite Equation (5.44) as

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (p_d)^{\gamma_2(n)}(p_{i_d})^{-\gamma_1(n)} P(\mathsf{FO}^{(n)}, \text{all-to-all}). \tag{5.48}$$

For any $\varepsilon > 0$, setting $(p_d) = (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$ yields

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{all-to-all}), \tag{5.49}$$

proving part (i) of Theorem 8.

Figure 5.3 shows this ratio plotted for different values of n. We see that, as expected from Equation (5.42), adaptive advantage becomes possible when around 2^{10} qubits are involved. This adaptive advantage grows logarithmically in n (as Theorem 8 states).

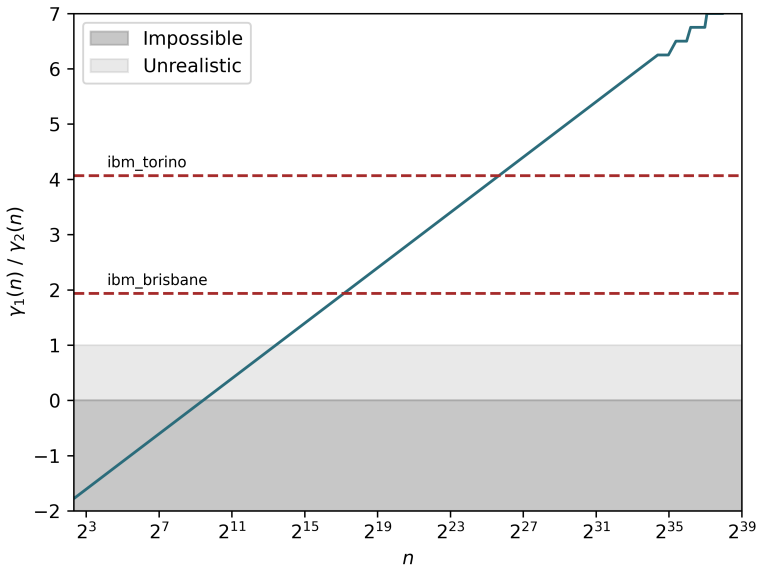


Figure 5.3: Comparison of the LAQCC and the non-adaptive all-to-all implementation of the $\mathsf{FO}^{(n)}$-gate in terms of the function γ as defined in Equation (5.47). Note that the x-axis has logarithmic scale. For any n, the LAQCC approach outperforms the all-to-all approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

Part (ii): 2D grid approach versus LAQCC approach

Comparing the 2D grid approach with the LAQCC approach and applying the assumptions on the success probabilities, we see that

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{FO}^{(n)}, \text{2D grid}) \tag{5.50}$$

precisely when

$$(p_d)^{5n-3}(p_{i_d})^{8n-1} \gtrsim (p_d)^{n-1}(p_{i_d})^{n\sqrt{n}+2n-12\sqrt{n}+11}. \tag{5.51}$$

Similar to the previous section, we can rearrange this expression, and get that Equation (5.57) holds when

$$(p_d)^{\gamma_2(n)} \gtrsim (p_{i_d})^{\gamma_1(n)}, \tag{5.52}$$

where

$$\gamma_1(n) = n\sqrt{n} - 6n - 12\sqrt{n} + 12 \in \mathcal{O}(n\sqrt{n}), \qquad \gamma_2(n) = 4n - 2 \in \mathcal{O}(n). \tag{5.53}$$

From Equation (5.53) we see that

$$\gamma(n) = \frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n\sqrt{n} - 6n - 12\sqrt{n} + 12}{4n - 2} \in \mathcal{O}(\sqrt{n}). \tag{5.54}$$

Figure 5.4 shows this ratio plotted for different values of n. Compared to the all-to-all implementation, the advantage for the 2D grid implementation already occurs for much smaller arities.

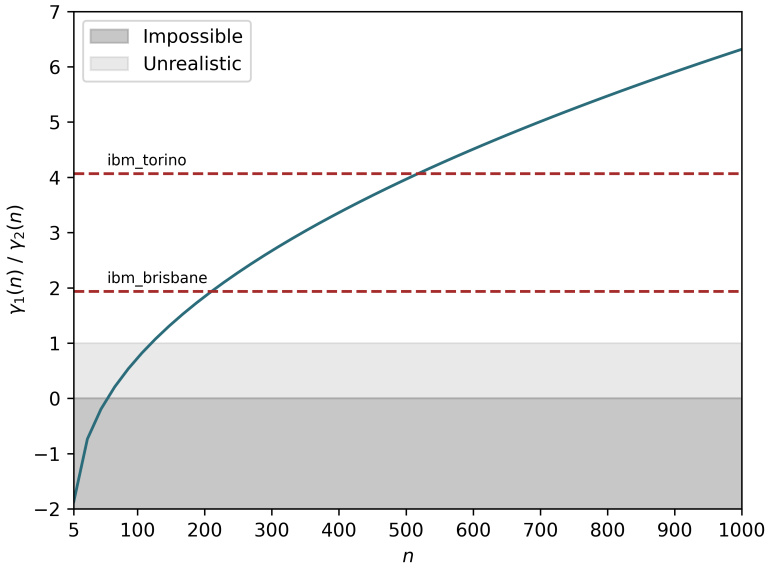


Figure 5.4: Comparison of the LAQCC and the non-adaptive 2D grid implementation of the $\mathsf{FO}^{(n)}$-gate in terms of the function γ, as defined in Equation (5.54). For any n, the LAQCC approach outperforms the 2D grid approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

Rewriting Equation (5.51), we get

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (p_d)^{\gamma_2(n)}(p_{i_d})^{-\gamma_1(n)} P(\mathsf{FO}^{(n)}, \text{2D grid}), \tag{5.55}$$

and thus, for any $\varepsilon > 0$ filling in $(p_d) = (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$ gives

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{2D grid}). \tag{5.56}$$

This proves part (ii) of Theorem 8.

Part (iii): 1D grid approach versus LAQCC approach

Comparing the 2D grid approach with the LAQCC approach and applying the assumptions on the success probabilities, we see that

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{FO}^{(n)}, \text{2D grid}) \tag{5.57}$$

precisely when

$$(p_d)^{5n-3}(p_{i_d})^{8n-1} \gtrsim (p_d)^{n-1}(p_{i_d})^{n^2-3n+2}. \tag{5.58}$$

Similar to the previous section, we can rearrange this expression, and get that Equation (5.58) holds when

$$(p_d)^{\gamma_2(n)} \gtrsim (p_{i_d})^{\gamma_1(n)}, \tag{5.59}$$

where

$$\gamma_1(n) = n^2 - 11n + 3 \in \mathcal{O}(n^2), \qquad \gamma_2(n) = 4n - 2 \in \mathcal{O}(n), \tag{5.60}$$

and so

$$\gamma(n) := \frac{\gamma_1(n)}{\gamma_2(n)} = \frac{n^2 - 11n + 3}{4n - 2} \in \mathcal{O}(n). \tag{5.61}$$

Figure 5.5 shows this ratio plotted for different values of n. Compared to the 1D grid approach, we see that adaptive advantage indeed increases rapidly. Rewriting Equation (5.58), we get

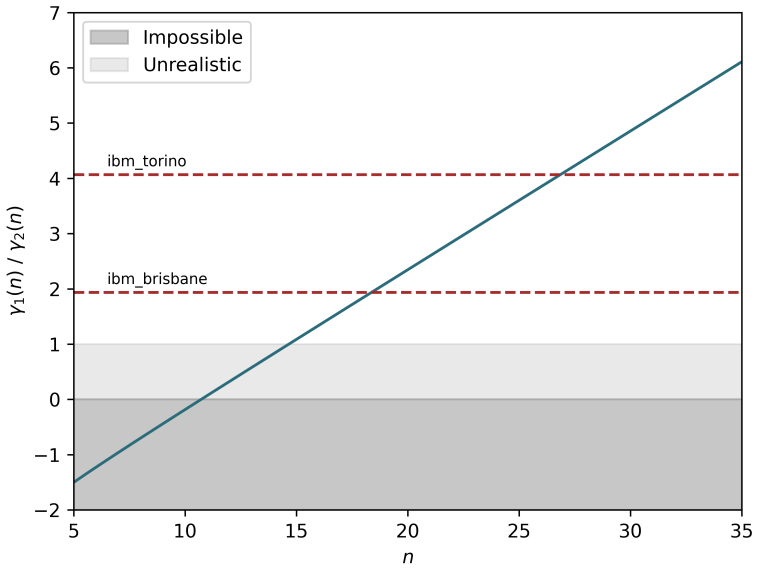


Figure 5.5: Comparison of the LAQCC and the non-adaptive 1D grid implementation of the $\mathsf{FO}^{(n)}$-gate in terms of the function γ, as defined in Equation (5.61). For any n, the LAQCC approach outperforms the 2D grid approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (p_d)^{\gamma_2(n)}(p_{i_d})^{-\gamma_1(n)} P(\mathsf{FO}^{(n)}, \text{1D grid}), \tag{5.62}$$

and thus, for any $\varepsilon > 0$ filling in $(p_d) = (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$ gives

$$P(\mathsf{FO}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{FO}^{(n)}, \text{1D grid}). \tag{5.63}$$

This proves part (iii) of Theorem 8.

5.3 OR-gate

We proceed by determining the success probability of the OR-reduction protocol from Lemma 7 and the exponential-size OR-gate from Section 2.7.3. Together, these provide a polynomial-size implementation of the OR-gate. Then, we will compare the OR-gate implementations across different topologies.

Since both the OR-reduction and the OR-gate protocols are well-suited for a 2D-grid topology – as can be seen from their circuits discussed in Section 2.7.3 – no extra gates are required for correctly executing the protocol when constrained to such topology.[3] As such, the differences in success probability between different topologies are fully determined by the differences between the implementations of the underlying operations. which are Fanout-gates, GHZ-states and Parity-gates.

In particular, we will discuss the success probability of the OR-gate in three settings:

(i) the non-adaptive setting assuming all-to-all connectivity;

(ii) the non-adaptive setting assuming 2D grid connectivity; and

(iii) a LAQCC protocol.

Counting the gates and idling terms for all of these implementations yield expressions that are quite involved. The full expressions can be obtained from [Bra25].

The 2D grid connectivity approach uses the simplified 2D grid implementation of the Fanout-gate discussed in Section 5.2.3. The 1D grid implementation is not separately analyzed here, since the analysis for the Fanout-gate showed that its implementation is clearly inferior compared to the LAQCC approach, even for small n. Results for this approach can be obtained in [Bra25].

5.3.1 OR-reduction

We give the success probability of the circuit reducing an n-qubit input to an OR-gate to an m-qubit input, where $m = \lceil \log(n+1) \rceil$. The OR-reduction works by computing 'reduced' states $|\psi_k\rangle = |\mu_{\theta_k}^{|x|}\rangle$, with $\theta_k = 1/2^k$, in parallel for all $k \in [m]$. The circuit for one such reduced state can be seen in Figure 2.6.

Preparing these reduced states in parallel requires copying and uncopying the register $|x\rangle$ $m-1$ times, using n Fanout-gates of arity m. These Fanout-gates can be applied in parallel with the m Fanout-gates of arity $n-1$ that are shown in Figure 2.6. This means that full success probability of the OR-reduction can be expressed as

$$P(\mathsf{OR}^{(n)}\text{-reduction}) = (P(\mathsf{FO}^{(n)})(p_s))^{2m}(p_{i_s})^{nm-2m}P(\mathsf{FO}^{(m)})^{2n}P(\mathsf{CU})^{nm}. \tag{5.64}$$

In the LAQCC setting, we can make a small reduction when it comes to single-qubit idling terms, namely by incorporating the $2m$ (p_s) terms (corresponding to the Hadamard-gates in the circuit in Figure 2.6) into the circuit of the Fanout-gates:

$$P(\mathsf{OR}^{(n)}\text{-reduction}, \mathsf{LAQCC}) = \left(\frac{P(\mathsf{FO}^{(n)}, \mathsf{LAQCC})(p_s)}{(p_{i_s})}\right)^{2m} P(\mathsf{FO}^{(m)}, \mathsf{LAQCC})^{2n}P(\mathsf{CU})^{nm}. \tag{5.65}$$

5.3.2 OR-gate

We continue our discussion with the success probability of the exponential-size circuit for the $\mathsf{OR}^{(n)}$-gate from Section 2.7.3. For clarity, we repeat the steps:

(i) Do the following:

  (a) Copy the input state $|x\rangle$ and apply a $\mathsf{PA}_a^{(n)}$-gate in parallel for every non-zero $(n-1)$-bit string a (i.e. $2^{n-1}-1$ times);

  (b) Prepare a $2^{n-1}-1$-qubit GHZ-state;

[3] Recall that in procedures such as the OR-reduction, we assume that the Fanout-gate implementation under a 2D grid constraint places all target qubits on a line. See Section 5.2.3.

(ii) Apply controlled-$\mathsf{R_Z}(\pi/2^{n-1})$-gates in parallel, controlled by the qubits that hold the results of the $\mathsf{PA}_a^{(n)}$-gates and targetting the qubits of the GHZ-state;

(iii) Apply an $\mathsf{FO}^{(2^{n-1}-1)}$-gate to the GHZ-state, followed by a Hadamard-gate, which prepares a qubit holding the OR of the input;

(iv) Finally, uncompute the results of the weighted Parity-gates and the copies of the input state $|x\rangle$.

We refer to Section 2.7.3 for correctness. The weighted Parity-gates are simply Parity-gates that target precisely the qubits placed at the non-zero entries of a. For a lower bound on the success probabilities of these gates, we will assume that they target all $n-1$ input qubits, giving a total of $2(2^{n-1}-1)$ Parity-gates of arity n. By Equation (2.29), the success probability of a Parity-gate is

$$P(\mathsf{PA}^{(n)}) = P(\mathsf{FO}^{(n)})(p_s)^{2n}. \tag{5.66}$$

The success probability of the $\mathsf{OR}^{(n)}$-gate combined with the OR-reduction can now be expressed as

$$\begin{aligned} P(\mathsf{OR}^{(n)}) &= P(\mathsf{OR}^{(n-1)}\text{-reduction})^2 P(\mathsf{FO}^{(2^m-1)})^{2m} P(\mathsf{PA}^{(m+1)})^{2(2^m-1)} P(\mathrm{GHZ}^{(2^m-1)}) P(\mathsf{FO}^{(2^m-1)}) \\ &\quad \cdot P(\mathsf{CU})^{2^m-1} P_i(\mathsf{CU})^{m(2^m-1)}(p_s) \\ &= P(\mathsf{OR}^{(n-1)}\text{-reduction})^2 P(\mathsf{FO}^{(2^m-1)})^{2m+1}(p_s)^{4(m+1)(2^m-1)+1} P(\mathsf{FO}^{(m+1)})^{2(2^m-1)} P(\mathrm{GHZ}^{(2^m-1)}) \\ &\quad \cdot P(\mathsf{CU})^{2^m-1} P_i(\mathsf{CU})^{m(2^m-1)}. \end{aligned} \tag{5.67}$$

Note that for an $\mathsf{OR}^{(n)}$-gate, the input to the OR-reduction is of arity $n-1$, hence $m = \lceil \log(n) \rceil$. The success probability for preparing the GHZ-state is induced by the circuit in Figure 2.3 for the non-adaptive topologies:

$$P(\mathrm{GHZ}^{(n)}, \text{non-adaptive}) = (p_s)(p_{i_s})^{n-1} P(\mathsf{FO}^{(\lfloor n/2 \rfloor)}) P(\mathsf{FO}^{(\lceil n/2 \rceil)}). \tag{5.68}$$

In the LAQCC setting, its success probability follows from the circuit in Figure 3.3:

$$P(\mathrm{GHZ}^{(n)}, \mathsf{LAQCC}) = (p_s)^{n+\lfloor n/2 \rfloor}(p_d)^{2n-2}(p_m)^{n-1}(p_{i_s})^{n+\lceil n/2 \rceil-1}(p_{i_d})^2(p_{i_m})^n(p_{i_c})^n. \tag{5.69}$$

Note that the LAQCC circuit was already partially utilized in determining the success probability of the Fanout-gates.

5.3.3 Comparison of OR-gate implementations

In this section, we compare the success probabilities of implementing an OR-gate in an LAQCC setting, and see under which circumstances the LAQCC protocol outperforms the non-adaptive protocols. The results are summarized in Theorem 9.

Theorem 9. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim ()(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\log(n)), \tag{5.70}$$

then

$$P(\mathsf{OR}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{OR}^{(n)}, \text{all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\sqrt{n}), \tag{5.71}$$

then

$$P(\mathsf{OR}^{(n)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\mathsf{OR}^{(n)}, \text{2D grid}).$$

Part (i): all-to-all approach versus LAQCC approach

We apply the success probability assumptions of Section 5.1, fill in the appropriate expressions for the Fanout-gates, GHZ-state preparation and controlled U-gates. From the full gate counts (see [Bra25]), we can infer that

$$|\mathsf{OR}^{(n)}, \text{all-to-all}|_d \approx 18n\log(n) + n - 24\log(n) - 3 \in \mathcal{O}(n\log(n)), \tag{5.72}$$

and

$$\begin{aligned}|\mathsf{OR}^{(n)}, \text{all-to-all}|_{i_d} \approx{}& (4n-4)\log(n)\log(\log(n)) + (4n-1)\log(n)\log(n-1) - 16n\log(n) + 4n \\ &- (2n-36)\log(n)\log(\log(n)+1) + (2n-2)\log(n)^2 + (2n-2)\log(\log(n)+1) - 1 \\ &\in \mathcal{O}(n\log^2(n)).\end{aligned} \tag{5.73}$$

For the LAQCC approach, the gate counts are

$$|\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_{d+m} \approx 36n\log(n) + 4n - 42\log(n) - 5 \in \mathcal{O}(n\log(n)), \tag{5.74}$$

and there is an idling qubit count of

$$|\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_{i_d + i_m + i_c} \approx 98n\log(n) + 16n - 60\log(n) - 16 \in \mathcal{O}(n\log(n)). \tag{5.75}$$

Comparing the success probabilities of the all-to-all topology and the LAQCC approach for the OR-gate implementation, we see that

$$P(\mathsf{OR}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{OR}^{(n)}, \text{All-to-all}) \tag{5.76}$$

holds when

$$(p_d)^{\gamma_2(n)} \gtrsim (p_{i_d})^{\gamma_1(n)}, \tag{5.77}$$

where

$$\begin{aligned}\gamma_1(n) &= |\mathsf{OR}^{(n)}, \text{all-to-all}|_{i_d} - |\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_{i_d + i_m + i_c} \\ \gamma_2(n) &= |\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_d - |\mathsf{OR}^{(n)}, \text{all-to-all}|_d.\end{aligned} \tag{5.78}$$

By inspecting the obtained gate and idling term counts (Equations (5.72)-(5.75)), we see $\gamma_1(n)/\gamma_2(n) \in \mathcal{O}(\log(n))$ and part (i) of Theorem 9 follows.

Figure 5.6 shows $\gamma_1(n)/\gamma_2(n)$ plotted for different values of n.

Part (ii): 2D grid approach versus LAQCC approach

We apply the success probability assumptions of Section 5.1, fill in the appropriate expressions for the Fanout-gates, GHZ-state preparation and Controlled-U-gates. From the full gate counts (again, see [Bra25]), we can infer that

$$|\mathsf{OR}^{(n)}, \text{2D grid}|_d = |\mathsf{OR}^{(n)}, \text{all-to-all}|_d \approx 18n\log(n) + n - 24\log(n) - 3 \in \mathcal{O}(n\log(n)). \tag{5.79}$$

and

$$\begin{aligned}|\mathsf{OR}^{(n)}, \text{2D grid}|_{i_d} \approx{}& (6n\log(n) + (1+\sqrt{2})n - 102\log(n) - \left(\frac{33}{\sqrt{2}} + 17\right)\sqrt{n-1} + n(4\log(n) - 64)\sqrt{\log(n)} \\ &+ (2n\log(n) - 30n - 2\log(n) + 30)\sqrt{\log(n)+1} + 38n\log(n) + 97n \\ &- (4\log(n) - 64)\sqrt{\log(n)} + 48\log(n) - 58 \in \mathcal{O}(n\sqrt{n}\log(n)).\end{aligned} \tag{5.80}$$

Comparing the success probabilities of the 2D grid topology and the LAQCC approach for the OR-gate implementation, we see that

$$P(\mathsf{OR}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{OR}^{(n)}, \text{2D grid}) \tag{5.81}$$

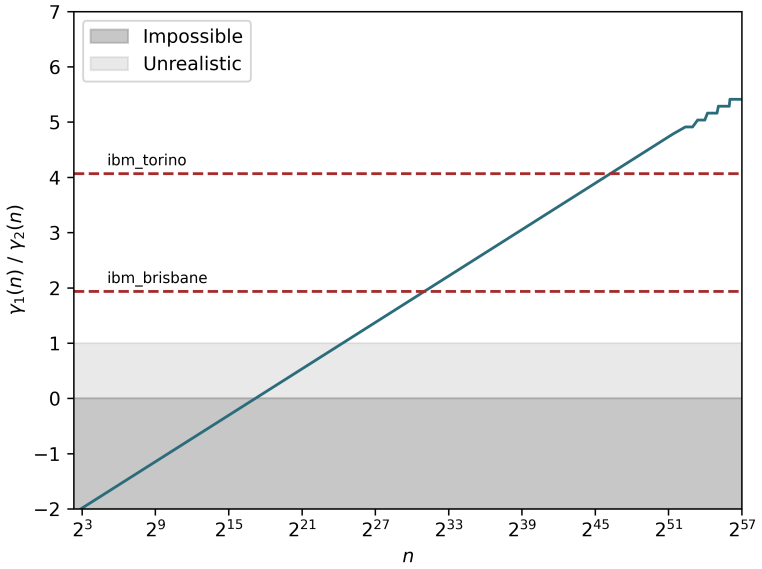


Figure 5.6: Comparison of the LAQCC and the non-adaptive all-to-all implementation of the $\mathsf{OR}^{(n)}$-gate in terms of the function $\gamma := \gamma_1/\gamma_2$, with γ_1 and γ_2 as defined in Equation (5.78). For any n, the LAQCC approach outperforms the all-to-all approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

holds when

$$(p_d)^{\gamma_2(n)} \gtrsim (p_{i_d})^{\gamma_1(n)}, \tag{5.82}$$

where

$$\begin{aligned} \gamma_1(n) &= |\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_d - |\mathsf{OR}^{(n)}, \text{2D grid}|_d \in \mathcal{O}(n\sqrt{n}\log(n)) \\ \gamma_2(n) &= |\mathsf{OR}^{(n)}, \text{2D grid}|_{i_d} |\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_{i_d+i_m+i_c} \in \mathcal{O}(n\log(n)). \end{aligned} \tag{5.83}$$

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{|\mathsf{OR}^{(n)}, \text{2D grid}|_{i_d} - |\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\mathsf{OR}^{(n)}, \mathsf{LAQCC}|_d - |\mathsf{OR}^{(n)}, \text{2D grid}|_d}. \tag{5.84}$$

By inspecting the gate and idling term counts, we see that $\gamma_1(n)/\gamma_2(n) \in \mathcal{O}(\sqrt{n})$ and part (ii) of Theorem 9 follows.

Figure 5.7 shows $\gamma_1(n)/\gamma_2(n)$ plotted for different values of n.

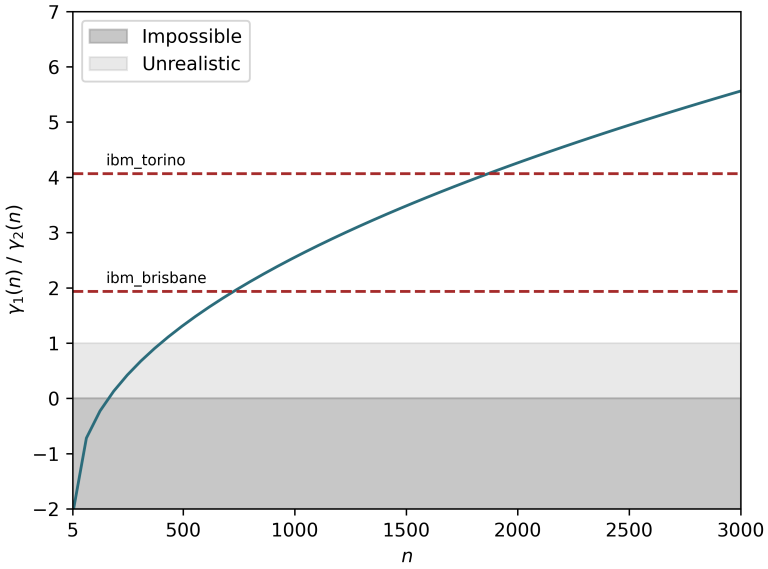


Figure 5.7: Comparison of the LAQCC and the non-adaptive 2D grid implementation of the $\mathsf{OR}^{(n)}$-gate in terms of the $\gamma := \gamma_1/\gamma_2$, with γ_1 and γ_2 as defined in Equation (5.78). For any n, the LAQCC approach outperforms the 2D grid approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

5.4 OR-based gates

5.4.1 AND-gates

Table 5.4 lists the gate and idling term counts for $\mathsf{AND}^{(n)}$ where $3 \leq n \leq 5$. The numbers for $n = 3$ are based on Figure 2.8, while the numbers for $n = 4$ and $n = 5$ are based on the circuits in Figure 2.9.

gate	idling?	s	d	i_s	i_d
$\mathsf{AND}^{(3)}$	no	10	6	12	5
$\mathsf{AND}^{(3)}$	yes	0	0	7	6
$\mathsf{AND}^{(4)}$	no	30	18	78	51
$\mathsf{AND}^{(4)}$	yes	0	0	28	24
$\mathsf{AND}^{(5)}$	no	70	42	238	167
$\mathsf{AND}^{(5)}$	yes	0	0	63	51

Table 5.4: Gate and idling term count for the $\mathsf{AND}^{(n)}$-gate where $3 \leq n \leq 5$.

For larger n, we can use the implementation using OR-gates discussed in Section 2.7.3. This yields success probabilities of

$$P(\mathsf{AND}^{(n)}) = (p_s)^n P(\mathsf{OR}^{(n)}) \tag{5.85}$$

and

$$P_i(\mathsf{AND}^{(n)}) = (p_{i_s}) P_i(\mathsf{OR}^{(n)}). \tag{5.86}$$

In Appendix B, we compare this implementation using OR-gates with a new recursive circuit by Nie et al. which implements the AND-gate in logarithmic depth [NZS24] without ancilla qubits. This circuit assumes all-to-all qubit connectivity. For large n, we find that the OR-gate implementation uses less gates and has less idling terms than the recursive implementation. Therefore, we will use the OR-gate implementation in subsequent results using the AND-gate.

5.4.2 EQ-gate

An EQ_k-gate can be implemented via an OR-gate surrounded by X-gates on the input qubits which are ones in the binary representation of k, and an X-gate on the output qubit. This gives success probabilities of

$$P(\mathsf{EQ}_k^{(n)}) = ((p_s)^{2|k|}(p_{i_s})^{2n-1-2|k|})^2(p_s)P(\mathsf{OR}^{(n)}). \tag{5.87}$$

and

$$P_i(\mathsf{EQ}_k^{(n)}) = P_i(\mathsf{OR}^{(n)})(p_{i_s})^2. \tag{5.88}$$

5.4.3 PERM-gate

In this section, we bound the success probability of a $\mathsf{PERM}^{(n,d)}$-gate, which permutes computational basis states according to a permutation $\sigma : \{0,1\}^n \to \{0,1\}^n$ with $d := |\sigma|$. We follow the circuit described in Section 2.7.5. The circuit starts by applying d $\mathsf{FO}^{(n)}$-gates and parallel $\mathsf{EQ}^{(n+1)}$-gates to get a one-hot encoding of the input domain. The success probability of this first part is given by

$$P(\mathsf{FO}^{(d)})^{2n}P(\mathsf{EQ}^{(n+1)})^d. \tag{5.89}$$

We then apply an n-qubit Hadamard-gate to change the input state into the Hadamard basis, fan out the input state, and apply, in parallel, controlled Z-gates with multiple targets. This effectively permutes the state $|i\rangle$ to the state $|\sigma(i)\rangle$ – albeit in the Hadamard basis. Another Fanout-gate followed by an n-qubit Hadamard-gate puts this state back into the computational basis. The success probability of this middle part of the operation is given by

$$(p_s)^{2n}(p_{i_s})^{2(n(d-1)+d)}P(\mathsf{FO}^{(d)})^{2n}\left(\prod_{i\in[d]} P(\mathsf{CZ}_{q_i}^{(n)})\right). \tag{5.90}$$

We can implement a controlled Z-gate with multiple targets using a Fanout-gate conjugated by Hadamard-gates, targeting the qubits corresponding to ones in the binary representation of q_i. For a lower bound on its success probability of these gates, we assume that we need a Fanout-gate of arity n for each q_i, giving

$$P(\mathsf{CZ}_{q_i}^{(n)}) = (p_s)^{2|k|}(p_{i_s})^{2(n-|k|)}P(\mathsf{FO}^{(|k|)}) \geq P(\mathsf{FO}^{(n)})(p_s)^{2n}. \tag{5.91}$$

Equation (5.90) then reduces to

$$(p_s)^{2n(d+1)}(p_{i_s})^{2(n(d-1)+d)}P(\mathsf{FO}^{(d)})^{2n}P(\mathsf{FO}^{(n)})^d. \tag{5.92}$$

Finally, we apply another round of Fanout-gates and parallel $\mathsf{EQ}^{(n+1)}$-gates to uncompute the one-hot register. In total, we get

$$P(\mathsf{PERM}^{(n,d)}) = (p_s)^{2n}(p_{i_s})^{2(n(d-1)+d)}P(\mathsf{FO}^{(d)})^{4n}P(\mathsf{EQ}^{(n+1)})^{2d}P(\mathsf{FO}^{(n)})^d. \tag{5.93}$$

The following theorem is shown in Appendix C.1.

Theorem 10. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\log(n)), \tag{5.94}$$

then

$$P(\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)}P(\mathsf{PERM}^{(n,d)}, \text{all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}\left(\sqrt{n} + \frac{\sqrt{d}}{\log(n)}\right), \tag{5.95}$$

then

$$P(\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)}P(\mathsf{PERM}^{(n,d)}, \text{2D grid}).$$

5.5 UCG-based QSP

In this section, we assess the adaptive advantage of the UCG-based QSP algorithms introduced in Section 4.2. After bounding the success probability of both the sequential and parallelized UCG implementation, we will compare the success probability of *general* UCG-based QSP in the LAQCC setting to the non-adaptive setting (with all-to-all connectivity or 2D grid connectivity). We then determine the success probability of the *sparse* UCG-based QSP algorithms, and compare the LAQCC approach to non-adaptive approach for different sparsities $d \in \text{poly}(n)$.

5.5.1 Sequential UCG

We revisit the procedure discussed in Section 4.2.1. The implementation of an n-qubit UCG requires 2^{n-1} layers consisting of an $\mathsf{EQ}^{(n)}$-gate and a controlled single-qubit gate, and finishes with an X-gate to reset the ancilla register. The $\mathsf{EQ}^{(n)}$-gates consist of $\mathsf{OR}^{(n)}$-gates surrounded by X-gates on the appropriate input qubits, and preceded by an X-gate on the output qubit. The first layer of the circuit contains an $\mathsf{EQ}_0^{(n)}$-gate, in which case the $\mathsf{OR}^{(n)}$-gate is preceded by a total of n X-gates.

For bounding the success probability of the circuit, we save some idling terms by incorporating the second layer of X-gates in the implementation of the EQ-gates into the first layer of the next EQ-gate. We can cycle through all $(n-1)$-qubit basis states using a Gray code, which means that we can start with the basis state corresponding to 0, and then use only one X-gate to map one state to the next. We can also incorporate the controlled single-qubit gates into this layer, since they do not interact with the X-gates. This yields a success probability of

$$\begin{aligned} P(\mathsf{UCG}^{(n)}, \text{sequential}) &= (p_s) \prod_{k\in[2^{n-1}]} \left(P(\mathsf{EQ}_k^{(n)}) P_i(\mathsf{EQ}_k^{(n)}) P(\mathsf{CU}) P_i(\mathsf{CU})^{(n-1)} \right) \\ &\leq (p_s)^{n+1} \prod_{k\in[2^{n-1}]} \left((p_s)^1 (p_{i_s})^{n-2} P(\mathsf{OR}^{(n)}) P_i(\mathsf{OR}^{(n)}) P(\mathsf{CU}) \right). \end{aligned} \tag{5.96}$$

At the same time, the success probability of a qubit staying coherent while idling during this circuit is

$$P_i(\mathsf{UCG}^{(n)}, \text{sequential}) = (p_{i_s})^{2^{n-1}+1} P_i(\mathsf{OR}^{(n)})^{2^{n-1}} P_i(\mathsf{CU})^{2^{n-1}} \tag{5.97}$$

5.5.2 Parallelized UCG

We revisit the procedure discussed in Section 4.2.2. The implementation of an n-qubit UCG requires the circuit in Figure 4.3, as well as computing and uncomputing the one-hot encoding shown in Equation (4.22). The (un)computation step requires $2n$ $\mathsf{FO}^{(2^{n-1})}$-gates, as well as 2^{n-1} $\mathsf{EQ}^{(n)}$-gates. Including idling terms, this yields a success probability of

$$P(\mathsf{FO}^{(2^{n-1})})^{2n} P(\mathsf{EQ}^{(n)})^{2^{n-1}} P_i(\mathsf{FO}^{(2^{n-1})})^2 P_i(\mathsf{EQ}^{(n)}). \tag{5.98}$$

The circuit in Figure 4.3 uses 6 $\mathsf{FO}^{(2^{n-1}+1)}$-gates, $3 \cdot 2^{n-1}$ controlled single-qubit gates, and $2^{n-1}+2$ uncontrolled single-qubit gates. Including idling terms, this yields a success probability of

$$P(\mathsf{FO}^{(2^{n-1}+1)})^6 P_i(\mathsf{FO}^{(2^{n-1}+1)})^{6\cdot(n+2^{n-1})} P(\mathsf{CU})^{3\cdot 2^{n-1}} P_i(\mathsf{CU})^{3n} (p_s)^{2^{n-1}+2} (p_{i_s})^{2n+3\cdot 2^{n-1}}. \tag{5.99}$$

We can incorporate the first and last Fanout-gates in the implementation of the Z-decomposition into the one-hot encoding procedure, which saves some idling terms.

In total, we get a success probability of

$$\begin{aligned} P(\mathsf{UCG}^{(n)}, \text{parallelized}) = {} & P(\mathsf{FO}^{(2^{n-1})})^{4n} P(\mathsf{EQ}^{(n)})^{2^n} P_i(\mathsf{FO}^{(2^{n-1})})^2 P_i(\mathsf{EQ}^{(n)})^2 \\ & P(\mathsf{FO}^{(2^{n-1}+1)})^6 P_i(\mathsf{FO}^{(2^{n-1}+1)})^{6(n+2^{n-1})} P(\mathsf{CU})^{3\cdot 2^{n-1}} P_i(\mathsf{CU})^{3n} (p_s)^{2^{n-1}+2} (p_{i_s})^{2n+3\cdot 2^{n-1}} \end{aligned} \tag{5.100}$$

and a success probability of staying coherent while idling during this circuit of

$$P_i(\mathsf{UCG}^{(n)}, \text{parallelized}) = P_i(\mathsf{FO}^{(2^{n-1})})^2 P_i(\mathsf{EQ}^{(n)})^2 P_i(\mathsf{FO}^{(2^{n-1}+1)})^6 P_i(\mathsf{CU})^3 (p_{i_s})^2. \tag{5.101}$$

5.5.3 Adaptive advantage for general UCG-based QSP

By Theorem 6, the success probability of preparing an n-qubit quantum state using UCGs can be expressed as

$$P(\text{QSP}^{(n)}, \text{UCG-based}) = \prod_{i=1}^{n} P(\mathsf{UCG}^{(i)}) P_i(\mathsf{UCG}^{(i)})^{n-i}. \tag{5.102}$$

This expression obviously depends on whether we use the sequential or the parallel UCG implementation. In both cases, we can infer functions γ that give a condition for when the success probability of the LAQCC approach is higher than one of the non-adaptive approaches. Interestingly, these functions turn out to be asymptotically equal. The precise asymptotics are summarized in Theorem 11, which is proved in Appendix C.

Theorem 11. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\gamma(n) = \frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\log(n)), \tag{5.103}$$

then

$$P(\text{QSP}^{(n)}, \text{UCG-based, LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{QSP}^{(n)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\gamma(n) = \frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\sqrt{n}), \tag{5.104}$$

then

$$P(\text{QSP}^{(n)}, \text{UCG-based, LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{QSP}^{(n)}, \text{UCG-based, 2D grid}).$$

In practice, however, we *do* see significant differences between the actual adaptive advantage functions of the two UCG implementations. Figure 5.8 shows these functions plotted for the sequential UCG. As can be expected from the structure of its circuit, the lines are very similar to those of the $\mathsf{OR}^{(n)}$-gate (cf. Figures 5.6 and 5.7). Assuming all-to-all connectivity, we do not expect an adaptive advantage over the non-adaptive protocol for preparing quantum states up to ≤ 1000 qubits. Assuming 2D grid connectivity, we see a slight (and increasing) adaptive advantage for preparing quantum states of ≥ 700 qubits using this method.

For the parallelized UCG, the results are shown in Figure 5.9. In this setting, the adaptive advantage is much clearer. Assuming all-to-all connectivity, we see an increasing adaptive advantage for preparing quantum states of ≥ 600 qubits using this method. For 2D grid connectivity, the results are very clear: even for preparing quantum states of 20 qubits, we see an adaptive advantage already if $p_d \approx (p_{i_d})^{12}$. These results too can be explained from the structure of its circuit: for preparing an n-qubit state, the parallelized UCG circuit uses Fanout-gates of arity 2^n. The growth of the functions in Figure 5.9 resemble the results obtained for Fanout-gates of similar arities (cf. Figures 5.3 and 5.4).

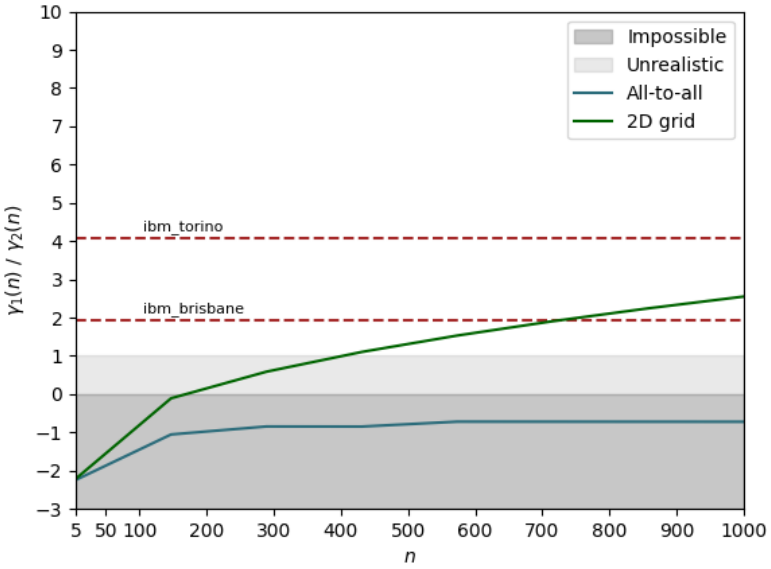


Figure 5.8: Comparison of the LAQCC approach of general UCG-based QSP using the *sequential* UCG implementation, to non-adaptive approaches for the same protocol assuming all-to-all and 2D grid qubit connectivity. This comparison is based on the functions γ_1 and γ_2 as defined in Equations (C.21) and (C.22) (in this case, assuming a sequential UCG). For any n, the LAQCC approach outperforms the non-adaptive approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

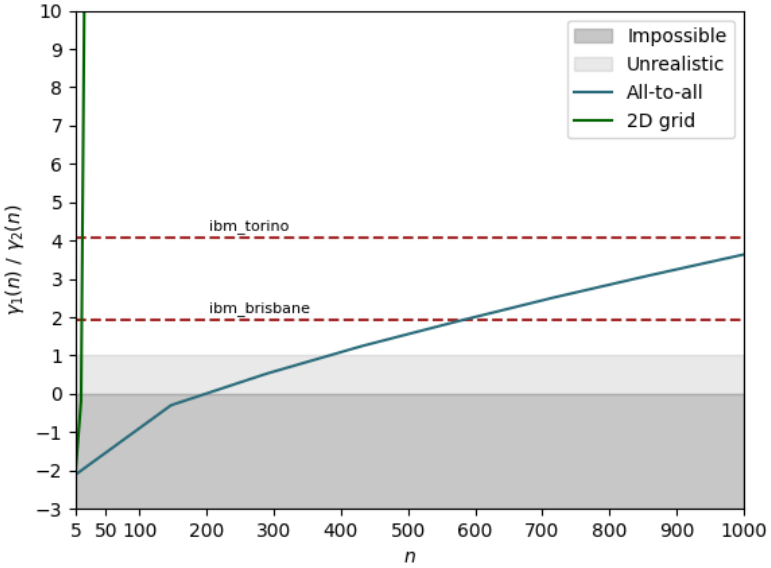


Figure 5.9: Comparison of the LAQCC approach of general UCG-based QSP using the *parallelized* UCG implementation, to non-adaptive approaches for the same protocol assuming all-to-all and 2D grid qubit connectivity. This comparison is based on the functions γ_1 and γ_2 as defined in Equations (C.21) and (C.22) (in this case, assuming a parallelized UCG). For any n, the LAQCC approach outperforms the non-adaptive approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

5.5.4 Adaptive advantage for sparse UCG-based QSP

The permutation-based procedure discussed in Section 4.2 gives the following success probability for preparing a d-sparse n-qubit state:

$$P(\text{QSP}^{(n,d)}, \text{UCG-based}) = P(\text{QSP}^{(\lceil \log(d) \rceil)}, \text{UCG-based}) P(\mathsf{PERM}_{\sigma_\psi}^{(n)}). \tag{5.105}$$

Asymptotically, the functions γ quantifying the conditions for adaptive advantage are still the same for both UCG implementations. This is to be expected, since we saw in the previous section that these functions are asymptotically the same in the dense setting. Moreover, the Permutation-gate that is added in the sparse setting does not use any UCG.

The precise asymptotics are summarized in Theorem 12, which is proved in Appendix C.

Theorem 12. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\gamma(n,d) = \frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}\left(\frac{\log^2(d)\log^2(\log(d))}{n\log(n)} + \log(n) \right), \tag{5.106}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{UCG-based}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\gamma(n,d) = \frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}\left(\frac{\log^2(d)\sqrt{\log(d)}\log\log(d)}{n\log(n)} + \frac{\sqrt{d}}{\log(n)} + \sqrt{n} \right) \tag{5.107}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}).$$

Figure 5.10 shows the functions γ for all-to-all connectivity and different values of d, when implementing UCGs via the sequential method. For $d \in \text{poly}(n)$, the gate and idling counts are dominated by the Permutation-gate, which uses Fanout-gates of arity d and EQ-gates of arity n. When $n \leq 1000$ and $d \in \text{poly}(n)$, we do not expect an adaptive advantage over the non-adaptive protocols for these gates (cf. Figure 5.3 and 5.6). The lines for this sparse UCG-based QSP are thus as can be expected, and we do not see an adaptive advantage for preparing sparse quantum states using this method.

Figure 5.11 shows the functions γ for the same implementation and sparsities, though now assuming 2D grid connectivity. In this setting, we see that the function *does* differ significantly based on the chosen sparsity, which is line with the asymptotic growth of the function γ shown in Thoerem 12. For $d = 10$ and $d = n$, this is $\mathcal{O}\left(\frac{n\sqrt{n}}{\log\log(n)}\right)$, while for $d = n^2$ it is $\mathcal{O}\left(\frac{n^2}{\log^2(n)\log\log(n)}\right)$, and for $d = n^3$ it is $\mathcal{O}\left(\frac{n^2\sqrt{n}}{\log^2(n)\log\log(n)}\right)$. For $d = n^3$, we see an adaptive advantage for preparing quantum states already for ≥ 50 qubits. For $d = n^2$, we see this for ≥ 500 qubits, while for $d \in \{10, n\}$, we see a slight (and increasing) adaptive advantage from about ≥ 700 qubits.

Since γ is dominated by the complexity of the Permutation-gate, we do not see substantial differences between the sequential and parallelized implementation of the UCG. Therefore, we omit the results of the parallelized version.

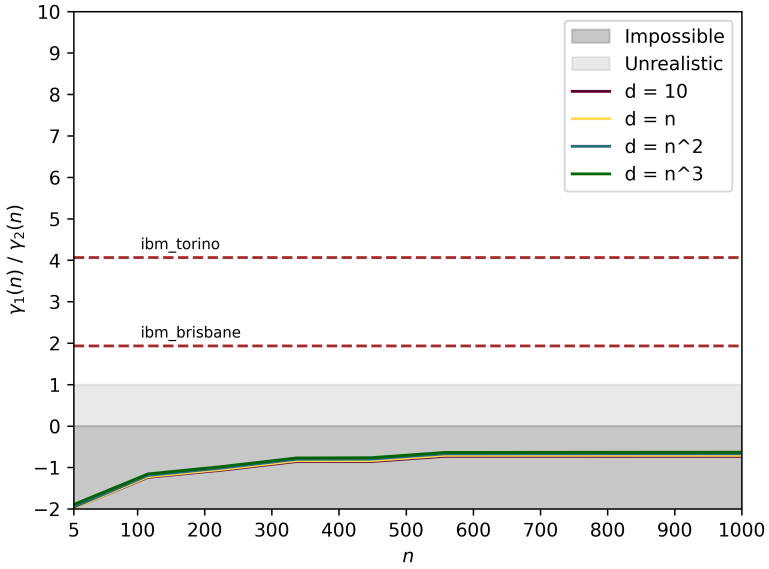


Figure 5.10: Comparison of the LAQCC approach of sparse UCG-based QSP using the sequential UCG implementation, to the non-adaptive approach for the same protocol assuming *all-to-all qubit connectivity*. This comparison is based on the functions γ_1 and γ_2 as defined in Equation (C.26), and is plotted for $d \in \{10, n, n^2, n^3\}$. By Theorem 12, $\gamma(n,d) \in \mathcal{O}(\log(n)\log\log(n))$ for all inspected values of d. For any n, the LAQCC approach outperforms the non-adaptive approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

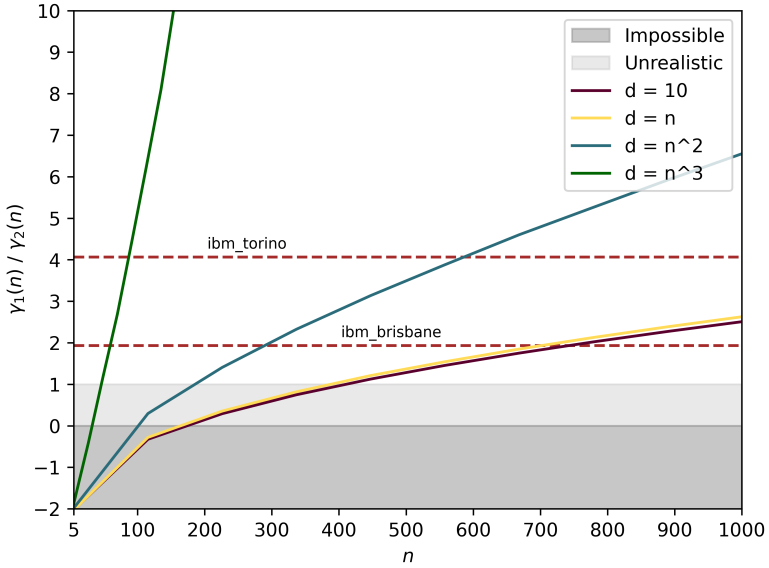


Figure 5.11: Comparison of the LAQCC approach of sparse UCG-based QSP using the sequential UCG implementation, to the non-adaptive approach for the same protocol assuming *2D grid qubit connectivity*. This comparison is based on the functions γ_1 and γ_2 as defined in Equation (C.27), and is plotted for $d \in \{10, n, n^2, n^3\}$. The asymptotic growth of functions pictured can be obtained from Theorem 12 part (ii). For any n, the LAQCC approach outperforms the non-adaptive approach if $(p_d) \geq (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$. The smallest and greatest value for γ in Table 5.2 are plotted for reference.

5.6 Unary-based QSP

In this section, we assess the adaptive advantage of the unary-based QSP algorithms introduced in Section 4.3. After bounding the success probability of the data loader procedure and the unary-to-binary transform, we compare the success probability of both general and sparse unary-based QSP, and assess adaptive advantage.

5.6.1 Data loading

We revisit the data loader procedure discussed in Section 4.3.1. Firstly, by the circuit in Figure 4.4, we observe that the success probability of the RBS-gate and the probability of a qubit staying coherent while idling during this gate are

$$P(\mathsf{RBS}) = (p_s)^2(p_d)^3(p_{i_s})^2 \tag{5.108}$$

and

$$P_i(\mathsf{RBS}) = (p_{i_s})^2(p_{i_d})^3, \tag{5.109}$$

respectively. Assuming all-to-all qubit connectivity, the data loader requires a circuit consisting of $\lceil \log(d) \rceil + 2$ layers to encode a total of d entries. See Figure 4.5 for an example when $d = 8$. The first layer contains a single X-gate. This is followed by $\lceil \log(d) \rceil$ layers that encode the amplitudes of the entries using RBS-gates. These are structured in a similar way to the Fanout-gate implementation with all-to-all connectivity. The last layer consists of d single-qubit phase-gates. The total success probability of this circuit is

$$P(\text{data loading}^{(d)}, \text{all-to-all}) = P(\mathsf{RBS})^{d-1} P_i(\mathsf{RBS})^{d\lceil \log(d) \rceil - 2d + 2} (p_s)^{d+1} (p_{i_s})^{d-1}. \tag{5.110}$$

For determining the value of the exponent, we refer to the calculations made for the Fanout-gate in Section 5.2.1.

Assuming 2D grid connectivity, the data loader requires a circuit consisting of $\lceil \frac{d}{2} \rceil + 2$ layers to encode d entries. See Figure 4.6 for an example when $d = 8$. This circuit is used in the non-adaptive setting assuming 2D grid connectivity, as well as in the LAQCC setting. We obtain a success probability of

$$P(\text{data loading}^{(d)}, \text{2D grid}) = P(\text{data loading}^{(d)}, \mathsf{LAQCC}) = P(\mathsf{RBS})^{d-1} P_i(\mathsf{RBS})^{d\lceil \frac{d}{2} \rceil - 2(d-1)} (p_s)^{d+1} (p_{i_s})^{d-1}. \tag{5.111}$$

5.6.2 Unary-to-binary transform

The success probability of the transformation from unary to binary encoding (Lemma 14) is made up of the success probabilities of the transformations described in Equations (4.38) and (4.39). In fact, we observe that the circuit implementing these transformation is the same circuit implementing a Permutation-gate, except for the first step computing the one-hot encoding, which is already done by the data loader circuit. In particular, applying Lemma 14 to prepare a binary encoding of a d-sparse n-qubit state corresponds to applying a $\mathsf{PERM}_\sigma^{(n)}$-gate where $|\sigma| = d$ (except for the computation of the one-hot encoding). Using Equation (5.89) and (5.92), we get

$$P(\text{unary-to-binary}^{(n,d)}) = (p_s)^{2n} (p_{i_s})^{2(n(d-1)+d)} P(\mathsf{FO}^{(d)})^{2n} P(\mathsf{EQ}^{(n+1)})^d P(\mathsf{FO}^{(n)})^d \tag{5.112}$$

5.6.3 Adaptive advantage for general unary-based QSP

The success probability of general unary-based QSP is given by simply combining the data loader procedure and the unary-to-binary transformation

$$P(\text{QSP}^{(n)}, \text{unary-based}) = P(\text{data loading}^{(2^n)}) P(\text{unary-to-binary}^{(n,2^n)}). \tag{5.113}$$

Based on the two transformations that make up the unary-based QSP approach, we do not expect an adaptive advantage when assuming all-to-all connectivity. This is due to the data loading procedure, which incurs exponentially more idling terms for the LAQCC approach than for the all-to-all approach (cf. Equations

(5.110) and (5.111)). For implementing the unary-to-binary transform, we expect an adaptive advantage only for very large n, based on the analyses for the gates comprising this procedure, and based on the results of the sparse UCG-based QSP protocol, which uses the (very similar) Permutation-gate (cf. Figure 5.10). Even for really large n, the adaptive advantage for the LAQCC protocol in the implementation of the unary-to-binary transform grows too slowly to offset the rapidly growing disadvantage it has in the data loader procedure. Figure 5.12 shows that this expectation indeed holds.

In comparison with the 2D grid approach, the situation looks very different. Since both protocols use the same variant of the data loader procedure, the adaptive advantage depends solely on the implementation of the unary-to-binary transform. For the (again, very similar) Permutation-gate, we already saw significant adaptive advantage over the 2D grid protocol for implementation permutations of size n^4 (see Figure 5.11). In this dense unary-based QSP setting, the unary-to-binary transform resembles the implementation of a permutation of size 2^n. Thus, we expect the adaptive advantage to be even more spectacular. Figure 5.13 shows the function γ plotted for small values of n, which shows that this expectation indeed holds.

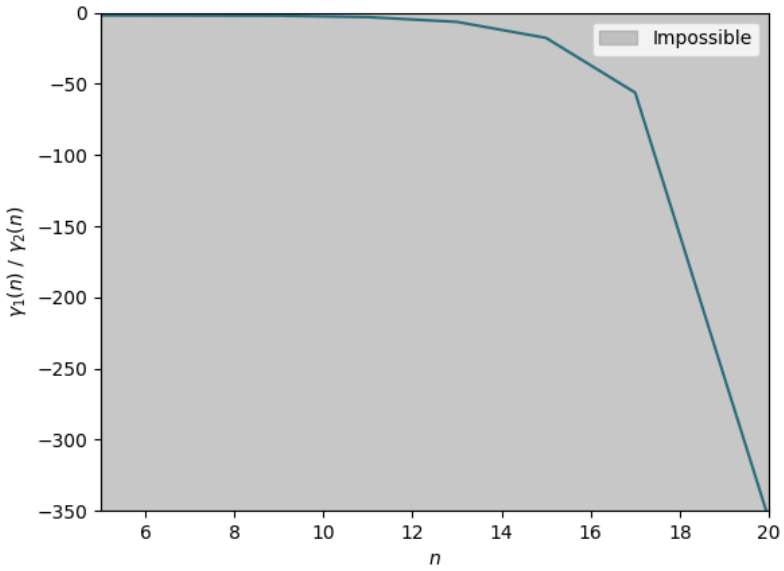


Figure 5.12: Comparison of the LAQCC approach of dense, unary-based QSP, to the non-adaptive approach for the same protocol assuming *all-to-all qubit connectivity*, in terms of the function γ associated with this protocol. Clearly, no adaptive advantage is to be expected.

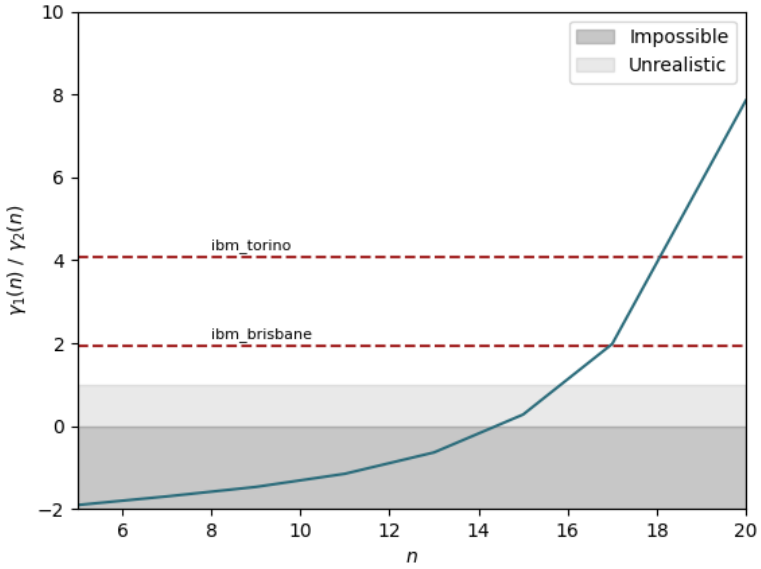


Figure 5.13: Comparison of the LAQCC approach of dense, unary-based QSP, to the non-adaptive approach for the same protocol assuming *2D grid qubit connectivity*, in terms of the function γ associated with this protocol. Adaptive advantage is expected even for small values of n.

5.6.4 Adaptive advantage for sparse unary-based QSP

The success probability of sparse unary-based is similar to the general case, except for swapping 2^n for some sparsity d:

$$P(\text{sparse QSP}^{(n,d)}, \text{unary-based}) = P(\text{data loading}^{(d)})P(\text{unary-to-binary}^{(n,d)}). \tag{5.114}$$

The asymptotics of the adaptive advantage functions are summarized in Theorem 13, which is shown in Appendix C.4

Theorem 13. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $d \in \mathcal{O}(n\log^2(n))$, and $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\log(n)), \tag{5.115}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(d + n\sqrt{n}\log(n) + n\sqrt{d}), \tag{5.116}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}).$$

Figure 5.14 shows the functions γ for all-to-all connectivity, different values of d, when implementing sparse unary-based QSP. As expected by Theorem 13, we see that for $d \in \mathcal{O}(1)$ and $d \in \mathcal{O}(n)$, the adaptive advantage increases slightly as n increases. However, comparable to results for sparse UCG-based QSP, it grows too slowly to get an actual adaptive advantage for preparing quantum states of ≤ 1000 qubits.

Figure 5.15 shows the functions γ for 2D grid connectivity and the same values of d. Similar to the results of sparse UCG-based QSP, the main gate and idling term counts are dominated by the unary-to-binary transform. Since this procedure is very similar to the Permutation-gate used in sparse UCG-based QSP, the resulting functions are very similar to Figure 5.11.

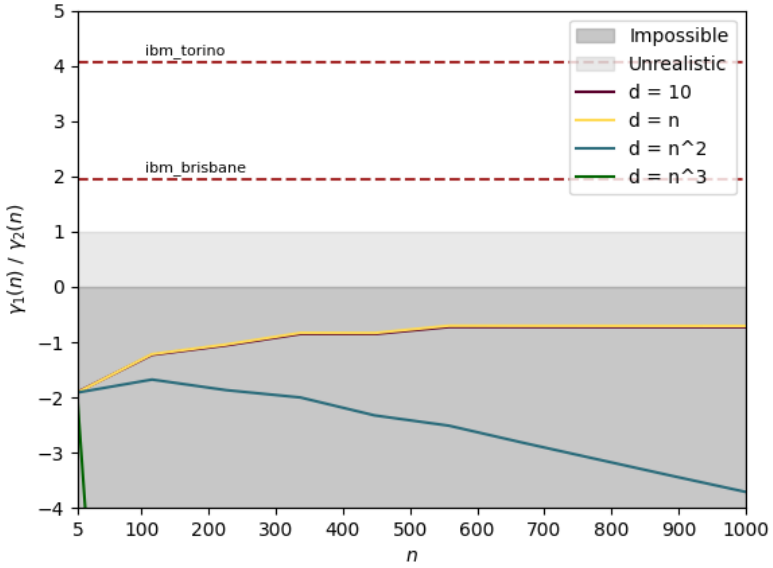


Figure 5.14: Comparison of the LAQCC approach of sparse, unary-based QSP, to the non-adaptive approach for the same protocol assuming *all-to-all qubit connectivity*, in terms of the function γ as defined in Equation (C.33), for $d \in \{10, n, n^2, n^3\}$.

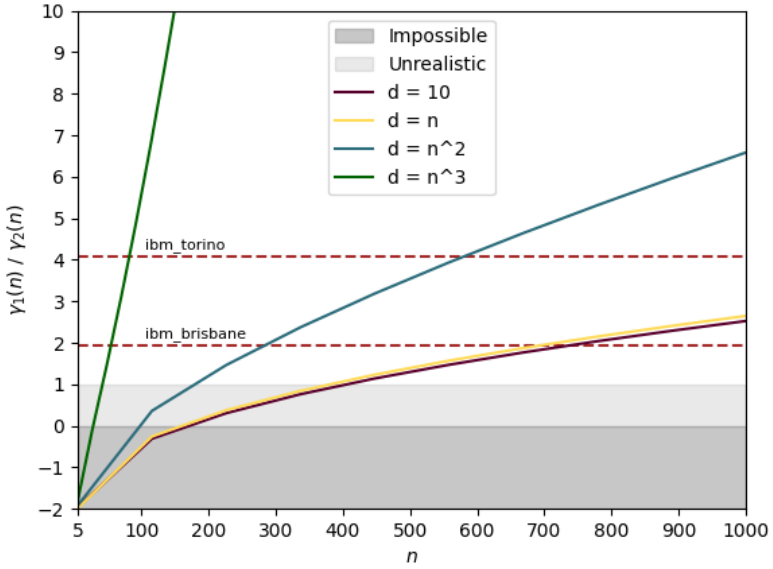


Figure 5.15: Comparison of the LAQCC approach of sparse, unary-based QSP, to the non-adaptive approach for the same protocol assuming *2D grid qubit connectivity*, in terms of the function γ as defined in Equation (C.34), for $d \in \{10, n, n^2, n^3\}$.

Chapter 6

Conclusion and future research

6.1 Conclusion

In this thesis, we have analysed the success probability of multiple approaches to quantum state preparation and investigated whether adaptive circuits can offer an advantage over non-adaptive approaches. Given the importance of efficient quantum state preparation for quantum algorithms, our results contribute to understanding when adaptive strategies may yield meaningful improvements.

We adopted an error model where each error corresponds to a uniformly random gate being applied to the qubits. Under this model, we introduced the adaptive advantage function γ to quantify when adaptive circuits outperform non-adaptive circuits. Specifically, we compared adaptive and non-adaptive QSP approaches for preparing quantum states up to 1000 qubits under different qubit connectivity assumptions, using both theoretical bounds and empirical error rate estimates based on quantum processing units from IBM.

For *general* UCG-based QSP, we found that the adaptive approach performed exponentially better than a non-adaptive approach under certain conditions. Specifically, adaptive circuits performed better than non-adaptive circuits with all-to-all qubit connectivity if $\gamma(n) \in \Omega(\log(n))$, and better than a non-adaptive approach with 2D grid connectivity if $\gamma(n) \in \Omega(\sqrt{n})$ (see Theorem 11). This can be interpreted as follows: for preparing an n-qubit quantum state, there is an adaptive advantage if the success probability of a two-qubit gate is at least the success probability of a qubit staying idle during $\gamma(n)$ two-qubit gates. The obtained resutls were the same both for a sequential and a parallelized implementation. In practice, we saw a possible adaptive advantage for preparing quantum states over the 2D grid approach, and for the parallelized UCG implementation, an advantage was seen even for the all-to-all approach.

For *sparse* UCG-based QSP, we saw similar trends, but with different growth factors: we found that the adaptive approach performed exponentially better than a non-adaptive all-to-all approach if $\gamma(n,d) \in \Omega\left(\frac{\log^2(d)\log^2(\log(d))}{n\log(n)} + \log(n)\right)$, and better than a non-adaptive 2D grid approach if $\gamma(n,d) \in \Omega\left(\frac{\log^2(d)\sqrt{\log(d)}\log\log(d)}{n\log(n)} + \frac{\sqrt{d}}{\log(n)} + \sqrt{n}\right)$ (see Theorem 12). In practice, we saw an adaptive advantage for preparing d-sparse quantum states (with $d \in \{10, n, n^2, n^3\}$) over the 2D grid approach, but no clear advantage over all-to-all connectivity. Table 1.1 lists the more precise estimations.

Next, we performed similar analyses for unary-based QSP. Here, the results were more nuanced. For *general* unary-based QSP, the adaptive approach does not outperform the non-adaptive all-to-all approach; it does exponentially outperform the 2D grid approach, already for preparing states consisting of few qubits. For *sparse* unary-based QSP, we found that, for $d \in \mathcal{O}(n\log^2(n))$, the adaptive approach performed exponentially better than a non-adaptive all-to-all approach if $\gamma(n,d) \in \Omega(\log(n))$. It performed exponentially better (with no restriction on d) than a 2D grid approach if $\gamma(n,d) \in \Omega(d + n\sqrt{n}\log(n) + n\sqrt{d})$ (see Theorem 13). Again, see Table 1.1 for more precise estimations when adaptive advantage could materialize.

Let us now come back to our original research question:

'under which conditions can we expect an advantage in using adaptive circuits over non-adaptive circuits for quantum state preparation?'

Our results indicate that grid-constrained adaptive quantum circuits can significantly improve the success probability of QSP algorithms, particularly in settings where qubit connectivity is constrained (e.g., in a 2D grid). The advantage is most pronounced when idling errors are relatively large compared to gate errors, as adaptive circuits reduce the time qubits spend idle. However, these benefits come with trade-offs: adaptive circuits introduce classical processing overhead, and require real-time feedback based on intermediate measurements. These may limit scalability. Furthermore, while adaptive circuits show clear advantage over 2D grid connnectivity, the benefits over all-to-all connectivity are less pronounced. Ultimately, whether adaptive circuits offer a *practical* advantage over non-adaptive circuits, will depend on the relative weight between error rates, qubit connectivity, and the cost of implementing adaptivity on a given quantum system.

6.2 Future research

A multitude of possible research directions exist. The following is a (non-exhaustive) list of them.

6.2.1 More realistic error model

An interesting follow-up to this research is to perform the same analysis using a more realistic error model.

For starters, we can lessen some of the assumptions of the current model, and include reported single-qubit gate and measurement errors. It is well-known that some single-qubit gates are much easier to implement than others, the T-gate being a prime example of the latter category. Instead of assuming that all single-qubit gates have the same success probability, we can take these differences into account. Similarly, we see that two-qubit gate error rates and measurement errors can also differ somewhat across devices. Taking this too into account may also give a more realistic view of possible adaptive advantage. At the same time, we expect that it will be harder to give closed-form asymptotic growth rates for adaptive advantage functions, since they will rely on more parameters.

Secondly, we may consider error models in which errors do not affect qubits independently. Rather, we can take into account correlations between errors on different qubits. In his paper called *The Argument against Quantum Computers* [Kal19], Kalai argues that entangled qubits are subject to positively correlated noise. Therefore, an error model with independent errors may be too simple to predict the behavior of complicated states and evolutions. In its most general form, an error model Λ is completely described by some Kraus operators $\{\mathsf{A}_i\}$, which act as

$$\Lambda(|\psi\rangle\langle\psi|) = \sum_i \mathsf{A}_i |\psi\rangle\langle\psi| \mathsf{A}_i^\dagger. \tag{6.1}$$

Well-known examples of these are *depolarizing channels*, *dephasing channels*, and *Pauli channels* [Wil16].

6.2.2 Advantage in actual hardware

In this thesis, we assessed adaptive advantage by comparing obtained ratios between gate errors and idling errors to the ratio between median gate and idling errors on some current quantum hardware. In and of itself, this does not mean that that these advantages are actually attainable on these devices. The number of available qubits and the variance in error between different qubits constitute some of the barriers. Taking more of these hardware considerations into account is imperative for research into adaptive speed-ups.

Relatedly, it would be interesting to see whether predicted adaptive advantages persist when we constrain ourselves to qubit topologies used in current quantum hardware, like IBMs *brick-wall* or Rigetti's *cycle-grid* [Mac+24]. It has been shown in [YZ24] that compilation of quantum circuits under both of these topologies induces only a constant factor of depth overhead over circuits assuming 2D grid connectivity. The precise adaptive advantage will depend on the magnitude of this constant.

6.2.3 Different QSP algorithms

For the purposes of this analysis, we have chosen (variants of) QSP algorithms that make extensive use of gates for which an adaptive depth reduction has been observed. Recent methods (such as the ones mentioned in Table 4.1) achieve similar asymptotic scaling of circuit depth and width using careful analyses of Gray codes and CNOT-ladders, and can sometimes be applied to any pre-specified number of ancilla qubits.

Note that we have only considered QSP algorithms that output the target state $|\psi\rangle$ exactly. Unavoidably, any such algorithm incurs a scaling that is exponential in the number of qubits, which either manifests itself in the width or the depth of the circuit implementing it. It has been suggested that this scaling issue can be addressed by employing variational methods [Nak+22] or quantum adversarial networks [ZLW19] with trained quantum circuits of low depth and width. In turn, this brought about the introduction of algorithms for *approximate QSP* with no exponential scaling. For example, in [ZD23] it is shown that one can approximate a quantum state up to error $\varepsilon > 0$ using only one ancilla qubit and a quantum circuit of depth $\mathcal{O}(1/\sqrt{\varepsilon})$ and size $\mathcal{O}(n + 1/\sqrt{\varepsilon})$.

6.2.4 Characterize states preparable in constant adaptive depth

Theorem 1 showed that for $d \in \Omega(n^{\log(n)})$, there are d-sparse n-qubit quantum states that are not preparable exactly using an LAQCC-circuit. An interesting direction for further research is to fully characterize the states that are contained within this class. For example, we can define an (d, d')-sparse vector as a d-sparse vector, of which d' are distinct (that is, there are d' distinct non-zero values, that can each occur multiple times, and together occur d times). We can then say a quantum state is (d, d')-sparse if it corresponds to a (d, d')-sparse input vector. By employing the data loader procedure from Section 4.3.1, and using the obtained unary encoding to control the W-state preparation scheme of [Buh+23], it is easy to show that any $(\text{poly}(n), \mathcal{O}(1))$-sparse quantum state is exactly preparable using an LAQCC-circuit.

Additionally, it would be interesting to explore how this characterization changes if we allow an LAQCC-circuit to prepare a state that *approximates* the target state with inner product at least $1 - \varepsilon$, rather than requiring exact preparation.

6.2.5 The power of adaptiveness

More generally, much is still unknown about the precise power of the LAQCC class. In Section 3.5, we discussed that a conjectured separation from [GS19] implies that $\mathsf{QNC}^0_\mathrm{f} \subsetneq \mathsf{LAQCC}$. Proving or disproving this separation will provide insights into the improvements in computational power provided by adding adaptivity.

One way to tackle this is by finding more use cases for the added intermediate measurements and classical computations. As far as we know, these have been mainly used to do two things: (1) determining and correcting measurement errors, and (2) imposing an ordering on a qubit register[1]. Finding more applications may help us better grasp what these models have to offer. Insights from the previous section may also contribute; distinguishing the capabilities of $\mathsf{QNC}^0_\mathrm{f}$ or LAQCC in preparing states could shed a light on the separation of the two circuit classes.

A related and pressing question is whether we can improve the adaptive circuits we already know. If we can find implementations that are more efficient, for example in terms of number of gates or ancilla use, this may lower the barrier for adaptive advantage.

[1] This is used, e.g., in the 'ordering' step of the constant-depth Dicke-state preparation scheme in [Buh+23].

Bibliography

[Abu24] Muhammad AbuGhanem. *IBM Quantum Computers: Evolution, Performance, and Future Directions*. September 2024. DOI: 10.48550/arXiv.2410.00916. arXiv: 2410.00916 [quant-ph].

[Ajt83] M. Ajtai. "Σ_1^1-Formulae on Finite Structures". In: *Annals of Pure and Applied Logic* 24.1 (July 1983), pp. 1–48. ISSN: 0168-0072. DOI: 10.1016/0168-0072(83)90038-6.

[All+23] Jonathan Allcock, Jinge Bao, João F. Doriguello, Alessandro Luongo and Miklos Santha. *Constant-Depth Circuits for Uniformly Controlled Gates and Boolean Functions with Application to Quantum Memory Circuits*. December 2023. arXiv: 2308.08539.

[ACG94] Noga Alon, F. R. K. Chung and R. L. Graham. "Routing Permutations on Graphs via Matchings". In: *SIAM Journal on Discrete Mathematics* 7.3 (May 1994), pp. 513–530. ISSN: 0895-4801. DOI: 10.1137/S0895480192236628. URL: https://epubs.siam.org/doi/10.1137/S0895480192236628 (visited on 18/03/2025).

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge; New York: Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.

[Bäu+24] Elisa Bäumer, Vinay Tripathi, Alireza Seif, Daniel Lidar and Derek S. Wang. *Quantum Fourier Transform Using Dynamic Circuits*. March 2024. DOI: 10.48550/arXiv.2403.09514. arXiv: 2403.09514 [quant-ph].

[Ben+93] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres and William K. Wootters. "Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels". In: *Physical Review Letters* 70.13 (March 1993), pp. 1895–1899. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.70.1895.

[Ber+15] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari and Rolando D. Somma. "Simulating Hamiltonian Dynamics with a Truncated Taylor Series". In: *Physical Review Letters* 114.9 (March 2015), p. 090502. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.114.090502. arXiv: 1412.4687 [quant-ph].

[Bra25] Daan In de Braekt. *adaptiveadvantage_qsp*. GitHub repository. 2025. URL: https://github.com/dnndbrkt/adaptiveadvantage_qsp.git.

[BM21] Sergey Bravyi and Dmitri Maslov. "Hadamard-Free Circuits Expose the Structure of the Clifford Group". In: *IEEE Transactions on Information Theory* 67.7 (July 2021), pp. 4546–4563. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2021.3081415. arXiv: 2003.09412 [quant-ph].

[BJS11] Michael J. Bremner, Richard Jozsa and Dan J. Shepherd. "Classical Simulation of Commuting Quantum Computations Implies Collapse of the Polynomial Hierarchy". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2126 (February 2011), pp. 459–472. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2010.0301. arXiv: 1005.1407 [quant-ph].

[BKP09] Dan E. Browne, Elham Kashefi and Simon Perdrix. *Computational Depth Complexity of Measurement-Based Quantum Computation*. September 2009. DOI: 10.48550/arXiv.0909.4673. arXiv: 0909.4673 [quant-ph].

[Buh+23] Harry Buhrman, Marten Folkertsma, Bruno Loff and Niels M. P. Neumann. *State Preparation by Shallow Circuits Using Feed Forward*. 2023. DOI: 10.48550/ARXIV.2307.14840.

[Chi+18] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross and Yuan Su. "Toward the First Quantum Simulation with Quantum Speedup". In: *Proceedings of the National Academy of Sciences* 115.38 (September 2018), pp. 9456–9461. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1801723115. arXiv: 1711.10980 [quant-ph].

[CM20] Matthew Coudron and Sanketh Menda. "Computations with Greater Quantum Depth Are Strictly More Powerful (Relative to an Oracle)". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. June 2020, pp. 889–901. DOI: 10.1145/3357713.3384269. arXiv: 1909.10503 [quant-ph].

[Fey82] Richard P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179.

[Fos+23] Michael Foss-Feig, Arkin Tikku, Tsung-Cheng Lu, Karl Mayer, Mohsin Iqbal, Thomas M. Gatterman, Justin A. Gerber, Kevin Gilmore, Dan Gresh, Aaron Hankin, Nathan Hewitt, Chandler V. Horst, Mitchell Matheny, Tanner Mengle, Brian Neyenhuis, Henrik Dreyer, David Hayes, Timothy H. Hsieh and Isaac H. Kim. *Experimental Demonstration of the Advantage of Adaptive Quantum Circuits*. February 2023. DOI: 10.48550/arXiv.2302.03029. arXiv: 2302.03029 [quant-ph].

[Got98] Daniel Gottesman. *The Heisenberg Representation of Quantum Computers*. July 1998. DOI: 10.48550/arXiv.quant-ph/9807006. arXiv: quant-ph/9807006.

[GC99] Daniel Gottesman and Isaac L. Chuang. "Demonstrating the Viability of Universal Quantum Computation Using Teleportation and Single-Qubit Operations". In: *Nature* 402.6760 (November 1999), pp. 390–393. ISSN: 1476-4687. DOI: 10.1038/46503.

[GM24] Daniel Grier and Jackson Morris. *Quantum Threshold Is Powerful*. November 2024. arXiv: 2411.04953 [quant-ph].

[GS19] Daniel Grier and Luke Schaeffer. *Interactive Shallow Clifford Circuits: Quantum Advantage against* NC^1 *and Beyond*. November 2019. arXiv: 1911.02555 [quant-ph].

[Gro96] Lov K. Grover. *A Fast Quantum Mechanical Algorithm for Database Search*. November 1996. DOI: 10.48550/arXiv.quant-ph/9605043. arXiv: quant-ph/9605043.

[GR02] Lov K. Grover and Terry Rudolph. *Creating Superpositions That Correspond to Efficiently Integrable Probability Distributions*. August 2002. DOI: 10.48550/arXiv.quant-ph/0208112. arXiv: quant-ph/0208112.

[HHL09] Aram W. Harrow, Avinatan Hassidim and Seth Lloyd. "Quantum Algorithm for Solving Linear Systems of Equations". In: *Physical Review Letters* 103.15 (October 2009), p. 150502. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.103.150502. arXiv: 0811.3171 [quant-ph].

[HJ17] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Second edition, corrected reprint. New York, NY: Cambridge University Press, 2017. ISBN: 978-0-521-83940-2 978-0-521-54823-6.

[Hv05] Peter Høyer and Robert Špalek. "Quantum Fan-out Is Powerful". In: *Theory of Computing* 1.1 (2005), pp. 81–103. ISSN: 1557-2862. DOI: 10.4086/toc.2005.v001a005.

[Ibm] *IBM Quantum*. URL: https://quantum.ibm.com/services/resources (visited on 14/01/2025).

[Joh+20] Sonika Johri, Shantanu Debnath, Avinash Mocherla, Alexandros Singh, Anupam Prakash, Jungsang Kim and Iordanis Kerenidis. *Nearest Centroid Classification on a Trapped Ion Quantum Computer*. December 2020. arXiv: 2012.04145 [quant-ph].

[Kal19] Gil Kalai. *The Argument against Quantum Computers*. August 2019. DOI: 10.48550/arXiv.1908.02499. arXiv: 1908.02499 [quant-ph].

[KM04] Phillip Kaye and Michele Mosca. *Quantum Networks for Generating Arbitrary Quantum States*. July 2004. DOI: 10.48550/arXiv.quant-ph/0407102. arXiv: quant-ph/0407102.

[KL21] Iordanis Kerenidis and Jonas Landman. "Quantum Spectral Clustering". In: *Physical Review A* 103.4 (April 2021), p. 042415. DOI: 10.1103/PhysRevA.103.042415.

[Ker+18] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo and Anupam Prakash. *Q-Means: A Quantum Algorithm for Unsupervised Machine Learning*. December 2018. DOI: 10.48550/arXiv.1812.03584. arXiv: 1812.03584 [quant-ph].

[KP16] Iordanis Kerenidis and Anupam Prakash. *Quantum Recommendation Systems*. September 2016. arXiv: 1603.08675 [quant-ph].

[KSB23] Petr Klimov, Richik Sengupta and Jacob Biamonte. *On Translation-Invariant Matrix Product States and Advances in MPS Representations of the W-State*. August 2023. arXiv: 2306.16456 [quant-ph].

[LMR14] Seth Lloyd, Masoud Mohseni and Patrick Rebentrost. "Quantum Principal Component Analysis". In: *Nature Physics* 10.9 (September 2014), pp. 631–633. ISSN: 1745-2481. DOI: 10.1038/nphys3029.

[LC19] Guang Hao Low and Isaac L. Chuang. "Hamiltonian Simulation by Qubitization". In: *Quantum* 3 (July 2019), p. 163. ISSN: 2521-327X. DOI: 10.22331/q-2019-07-12-163. arXiv: 1610.06546 [quant-ph].

[LL24] Jingquan Luo and Lvzhou Li. *Circuit Complexity of Sparse Quantum State Preparation*. June 2024. arXiv: 2406.16142.

[Ma+12] Xiao-song Ma, Thomas Herbst, Thomas Scheidl, Daqing Wang, Sebastian Kropatschek, William Naylor, Alexandra Mech, Bernhard Wittmann, Johannes Kofler, Elena Anisimova, Vadim Makarov, Thomas Jennewein, Rupert Ursin and Anton Zeilinger. "Quantum Teleportation Using Active Feed-Forward between Two Canary Islands". In: *Nature* 489.7415 (September 2012), pp. 269–273. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature11472. arXiv: 1205.3909 [quant-ph].

[Mac+24] Filip B. Maciejewski, Stuart Hadfield, Benjamin Hall, Mark Hodson, Maxime Dupont, Bram Evert, James Sud, M. Sohaib Alam, Zhihui Wang, Stephen Jeffrey, Bhuvanesh Sundar, P. Aaron Lott, Shon Grabbe, Eleanor G. Rieffel, Matthew J. Reagor and Davide Venturelli. "Design and Execution of Quantum Circuits Using Tens of Superconducting Qubits and Thousands of Gates for Dense Ising Optimization Problems". In: arXiv:2308.12423 (September 2024). DOI: 10.48550/arXiv.2308.12423. arXiv: 2308.12423 [quant-ph].

[MR18] Dmitri Maslov and Martin Roetteler. "Shorter Stabilizer Circuits via Bruhat Decomposition and Quantum Circuit Transformations". In: *IEEE Transactions on Information Theory* 64.7 (July 2018), pp. 4729–4738. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2018.2825602. arXiv: 1705.09176 [quant-ph].

[MS18] Mateusz Michałek and Yaroslav Shitov. *Quantum Version of Wielandt's Inequality Revisited*. September 2018. DOI: 10.48550/arXiv.1809.04387. arXiv: 1809.04387 [math].

[Moo99] Cristopher Moore. *Quantum Circuits: Fanout, Parity, and Counting*. March 1999. arXiv: quant-ph/9903046.

[MN98a] Cristopher Moore and Martin Nilsson. *Parallel Quantum Computation and Quantum Codes*. August 1998. DOI: 10.48550/arXiv.quant-ph/9808027. arXiv: quant-ph/9808027.

[MN98b] Cristopher Moore and Martin Nilsson. *Some Notes on Parallel Quantum Computation*. April 1998. DOI: 10.48550/arXiv.quant-ph/9804034. arXiv: quant-ph/9804034.

[Mot+04] Mikko Mottonen, Juha J. Vartiainen, Ville Bergholm and Martti M. Salomaa. *Transformation of Quantum States Using Uniformly Controlled Rotations*. July 2004. DOI: 10.48550/arXiv.quant-ph/0407010. arXiv: quant-ph/0407010.

[Nak+22] Kouhei Nakaji, Shumpei Uno, Yohichi Suzuki, Rudy Raymond, Tamiya Onodera, Tomoki Tanaka, Hiroyuki Tezuka, Naoki Mitsuda and Naoki Yamamoto. "Approximate Amplitude Encoding in Shallow Parameterized Quantum Circuits and Its Application to Financial Market Indicator". In: *Physical Review Research* 4.2 (May 2022), p. 023136. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.4.023136. arXiv: 2103.13211 [quant-ph].

[NM14] Yoshifumi Nakata and Mio Murao. "Diagonal Quantum Circuits: Their Computational Power and Applications". In: *The European Physical Journal Plus* 129.7 (July 2014), p. 152. ISSN: 2190-5444. DOI: 10.1140/epjp/i2014-14152-9. arXiv: 1405.6552 [quant-ph].

[Neu25] Niels M. P. Neumann. "Adaptive Quantum Computers: Decoding and State Preparation". Unpublished PhD Thesis. 2025.

[NZS24] Junhong Nie, Wei Zi and Xiaoming Sun. *Quantum Circuit for Multi-Qubit Toffoli Gate with Optimal Resource*. February 2024. DOI: 10.48550/arXiv.2402.05053. arXiv: 2402.05053 [quant-ph].

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. December 2010. DOI: 10.1017/CBO9780511976667.

[Per+07] David Perez-Garcia, Frank Verstraete, Michael M. Wolf and J. Ignacio Cirac. *Matrix Product State Representations*. May 2007. DOI: 10.48550/arXiv.quant-ph/0608197. arXiv: quant-ph/0608197.

[PS13] Paul Pham and Krysta M. Svore. *A 2D Nearest-Neighbor Quantum Architecture for Factoring in Polylogarithmic Depth*. April 2013. DOI: 10.48550/arXiv.1207.6655. arXiv: 1207.6655 [quant-ph].

[PSC21] Lorenzo Piroli, Georgios Styliaris and J. Ignacio Cirac. "Quantum Circuits Assisted by Local Operations and Classical Communication: Transformations and Phases of Matter". In: *Physical Review Letters* 127.22 (November 2021), p. 220503. DOI: 10.1103/PhysRevLett.127.220503.

[PSC24] Lorenzo Piroli, Georgios Styliaris and J. Ignacio Cirac. "Approximating Many-Body Quantum States with Quantum Circuits and Measurements". In: *Physical Review Letters* 133.23 (December 2024), p. 230401. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.133.230401. arXiv: 2403.07604 [quant-ph].

[RLR23] Debora Ramacciotti, Andreea-Iulia Lefterovici and Antonio F. Rotundo. *A Simple Quantum Algorithm to Efficiently Prepare Sparse States*. October 2023. arXiv: 2310.19309 [quant-ph].

[RN24] David Raveh and Rafael I. Nepomechie. *Dicke States as Matrix Product States*. August 2024. arXiv: 2408.04729 [quant-ph].

[Reb+18] Patrick Rebentrost, Adrian Steffens, Iman Marvian and Seth Lloyd. "Quantum Singular-Value Decomposition of Nonsparse Low-Rank Matrices". In: *Physical Review A* 97.1 (January 2018), p. 012327. DOI: 10.1103/PhysRevA.97.012327.

[Ros13] David J. Rosenbaum. "Optimal Quantum Circuits for Nearest-Neighbor Architectures". In: *LIPIcs, Volume 22, TQC 2013* 22 (2013), pp. 294–307. ISSN: 1868-8969. DOI: 10.4230/LIPICS.TQC.2013.294.

[Ros23] Gregory Rosenthal. *Query and Depth Upper Bounds for Quantum Unitaries via Grover Search*. July 2023. arXiv: 2111.07992 [quant-ph].

[SB09] Dan Shepherd and Michael J. Bremner. "Instantaneous Quantum Computation". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 465.2105 (May 2009), pp. 1413–1439. ISSN: 1364-5021, 1471-2946. DOI: 10.1098/rspa.2008.0443. arXiv: 0809.0847 [quant-ph].

[Sho97] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (October 1997), pp. 1484–1509. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/S0097539795293172. arXiv: quant-ph/9508027.

[Smi+23] Kevin C. Smith, Eleanor Crane, Nathan Wiebe and S. M. Girvin. *Deterministic Constant-Depth Preparation of the AKLT State on a Quantum Processor Using Fusion Measurements*. April 2023. arXiv: 2210.17548 [cond-mat, physics:quant-ph].

[Smi+24] Kevin C. Smith, Abid Khan, Bryan K. Clark, S. M. Girvin and Tzu-Chieh Wei. *Constant-Depth Preparation of Matrix Product States with Adaptive Quantum Circuits*. August 2024. arXiv: 2404.16083 [quant-ph].

[Sun+23] Xiaoming Sun, Guojing Tian, Shuai Yang, Pei Yuan and Shengyu Zhang. *Asymptotically Optimal Circuit Depth for Quantum State Preparation and General Unitary Synthesis*. February 2023. arXiv: 2108.06150 [quant-ph].

[TT12] Yasuhiro Takahashi and Seiichiro Tani. *Collapse of the Hierarchy of Constant-Depth Exact Quantum Circuits*. September 2012. DOI: 10.48550/arXiv.1112.6063. arXiv: 1112.6063 [quant-ph].

[Tan23] Ewin Tang. "Quantum Machine Learning Without Any Quantum". PhD thesis. 2023.

[Urs+04] Rupert Ursin, Thomas Jennewein, Markus Aspelmeyer, Rainer Kaltenbaek, Michael Lindenthal, Philip Walther and Anton Zeilinger. "Quantum Teleportation across the Danube". In: *Nature* 430.7002 (August 2004), pp. 849–849. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/430849a.

[Wat+19] Adam Bene Watts, Robin Kothari, Luke Schaeffer and Avishay Tal. *Exponential Separation between Shallow Quantum Circuits and Unbounded Fan-in Shallow Classical Circuits*. June 2019. DOI: 10.48550/arXiv.1906.08890. arXiv: 1906.08890.

[Wil16] Mark M. Wilde. *From Classical to Quantum Shannon Theory*. November 2016. DOI: 10.1017/9781316809976.001. arXiv: 1106.1445 [quant-ph].

[Wol23] Ronald de Wolf. *Quantum Computing: Lecture Notes*. January 2023. DOI: 10.48550/arXiv.1907.09415. arXiv: 1907.09415 [quant-ph].

[Yu+24] Jeffery Yu, Sean R. Muleady, Yu-Xin Wang, Nathan Schine, Alexey V. Gorshkov and Andrew M. Childs. *Efficient Preparation of Dicke States*. November 2024. DOI: 10.48550/arXiv.2411.03428. arXiv: 2411.03428 [quant-ph].

[YZ23] Pei Yuan and Shengyu Zhang. "Optimal (Controlled) Quantum State Preparation and Improved Unitary Synthesis by Quantum Circuits with Any Number of Ancillary Qubits". In: *Quantum* 7 (March 2023), p. 956. ISSN: 2521-327X. DOI: 10.22331/q-2023-03-20-956. arXiv: 2202.11302 [quant-ph].

[YZ24] Pei Yuan and Shengyu Zhang. *Full Characterization of the Depth Overhead for Quantum Circuit Compilation with Arbitrary Qubit Connectivity Constraint*. February 2024. DOI: 10.48550/arXiv.2402.02403. arXiv: 2402.02403 [quant-ph].

[Zal98] Christof Zalka. "Simulating Quantum Systems on a Quantum Computer". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (January 1998), pp. 313–322. DOI: 10.1098/rspa.1998.0162.

[ZLY22] Xiao-Ming Zhang, Tongyang Li and Xiao Yuan. "Quantum State Preparation with Optimal Circuit Depth: Implementations and Applications". In: *Physical Review Letters* 129.23 (November 2022), p. 230504. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.129.230504. arXiv: 2201.11495 [quant-ph].

[ZGS24] Yifan Zhang, Sarang Gopalakrishnan and Georgios Styliaris. "Characterizing MPS and PEPS Preparable via Measurement and Feedback". In: *PRX Quantum* 5.4 (October 2024), p. 040304. ISSN: 2691-3399. DOI: 10.1103/PRXQuantum.5.040304. arXiv: 2405.09615 [quant-ph].

[ZLW19] Christa Zoufal, Aurélien Lucchi and Stefan Woerner. "Quantum Generative Adversarial Networks for Learning and Loading Random Distributions". In: *npj Quantum Information* 5.1 (November 2019), pp. 1–9. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0223-2.

[ZD23] Julien Zylberman and Fabrice Debbasch. *Efficient Quantum State Preparation with Walsh Series*. November 2023. DOI: 10.48550/arXiv.2307.08384. arXiv: 2307.08384 [quant-ph].

Appendix A

Gate and idling term count table for Fanout-gate

	s	d	m
All-to-all	0	$n-1$	0
All-to-all *(idle)*	0	0	0
1D grid	0	$n-1$	0
1D grid *(idle)*	0	0	0
2D grid	0	$n-1$	0
2D grid *(idle)*	0	0	0
LAQCC	$2n + \lceil (n-1)/2 \rceil$	$3n-2$	$2n-1$
LAQCC *(idle)*	0	0	0

	i_s	i_d	i_m	i_c
All-to-all	0	$n\lceil \log(n) \rceil - 2n + 2$	0	0.
All-to-all *(idle)*	0	$\lceil \log(n) \rceil$	0	0
1D grid	0	$n^2 - 3n + 2$	0	0
1D grid *(idle)*	0	$n-1$	0	0
2D grid	0	$n\sqrt{n} + 2n - 12\sqrt{n} + 11$	0	0
2D grid *(idle)*	0	$\sqrt{n} + 1$	0	0
LAQCC	$5n + \lfloor (n-1)/2 \rfloor - 2$	$2n+1$	$3n-1$	$3n-1$
LAQCC *(idle)*	4	3	2	2

Table A.1: Gate and idling term count for FO-gate implementations.

Appendix B

Comparing AND-gate implementations

Recently, Nie et al. [NZS24] published a recursive, logarithmic-depth circuit for implementing the $\mathsf{AND}^{(n)}$-gate. In this section, we will compare the gate and idling term counts (see Section 5.1 for an explanation) of this recursive circuit, to the gate and idling term counts of implementing the $\mathsf{AND}^{(n)}$-gate using an $\mathsf{OR}^{(n)}$-gate. Since the circuit in [NZS24] assumes all-to-all qubit connectivity, we will compare it to the OR-gate implementation assuming the same connectivity.

We will start by determining the success probability of this recursive circuit, which is shown in Figure B.1. The circuit works by recursively cutting down the input size n of the AND-gates in halves, until the resulting AND-gates have a small enough arity ($n \leq 5$) such that they can be implemented directly. The circuits shown in red on the left-hand side of the circuit, between step t_2 and t_3, recursively implements the left-hand side of the circuit (up to step t_4). At step t_4, the output qubit stores the AND of the input qubits. The circuits shown in red on the right-hand side recursively implements the right-hand side, which uncomputes the (borrowed) ancilla qubits and restores the input qubits to their input state.

The success probability of the left half of the circuit can be written as

$$\begin{aligned} P(\mathsf{AND}^{(n)}, \text{recursive, left}) = P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)})^{n-4} (p_s)^4 (p_{i_s})^{n-4} P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)})^{\frac{n}{2^j}-2} \\ (P(\mathsf{AND}^{(n/2+1)}, \text{recursive, left}) P_i(\mathsf{AND}^{(n/2+1)}, \text{recursive, left}))^2, \end{aligned} \tag{B.1}$$

where possible floor and ceiling functions are omitted for ease of analysis. After k recursive steps, the arity of the AND-gate in the recursive term is $n/2^{k+1}+1$. To determine when a base case ($3 \leq n \leq 5$) is reached, we solve

$$\frac{n}{2^{k+1}} + 1 = 5, \tag{B.2}$$

which holds when $k = \log(n) - 3$. Thus, setting $k := \lfloor \log(n) \rfloor - 2$ ensures that $3 \leq \frac{n}{2^{k+1}} + 1 \leq 5$. Filling this into Equation (B.1) yields

$$\begin{aligned} P(\mathsf{AND}^{(n)}, \text{recursive, left}) &= P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)})^{n-4} (p_s)^4 (p_{i_s})^{n-3} P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)})^{n-3} \\ &\quad (P(\mathsf{AND}^{(\frac{n}{2}+1)}, \text{recursive, left}) P_i(\mathsf{AND}^{(\frac{n}{2}+1)}, \text{recursive, left}))^2 \\ &= \prod_{j \in [k]} \left(P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)})^{\frac{n}{2^j}-3} (p_s)^4 (p_{i_s})^{\frac{n}{2^j}-2} P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)})^{\frac{n}{2^j}-2} \right)^{2^j} \\ &\quad \cdot (P(\mathsf{AND}^{(3)}) P_i(\mathsf{AND}^{(3)}))^{b_3} (P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)}))^{b_4} (P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)}))^{b_5} \\ &= \prod_{j \in [k]} \left(\left(P(\mathsf{AND}^{(5)}) (p_s)^4 P(\mathsf{AND}^{(4)}) \right)^{2^j} P_i(\mathsf{AND}^{(5)})^{n-3\cdot 2^j} (p_{i_s})^{n-3\cdot 2^j} P_i(\mathsf{AND}^{(4)})^{n-2\cdot 2^j} \right) \\ &\quad \cdot (P(\mathsf{AND}^{(3)}) P_i(\mathsf{AND}^{(3)}))^{b_3} (P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)}))^{b_4} (P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)}))^{b_5} \\ &= \left((p_s)^4 P(\mathsf{AND}^{(4)}) P(\mathsf{AND}^{(5)}) \right)^{2^k-1} P_i(\mathsf{AND}^{(5)})^{nk-3(2^k-1)} (P_i(\mathsf{AND}^{(4)})(p_{i_s}))^{nk-2(2^k-1)} \\ &\quad \cdot (P(\mathsf{AND}^{(3)}) P_i(\mathsf{AND}^{(3)}))^{b_3} (P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)}))^{b_4} (P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)}))^{b_5}. \end{aligned} \tag{B.3}$$

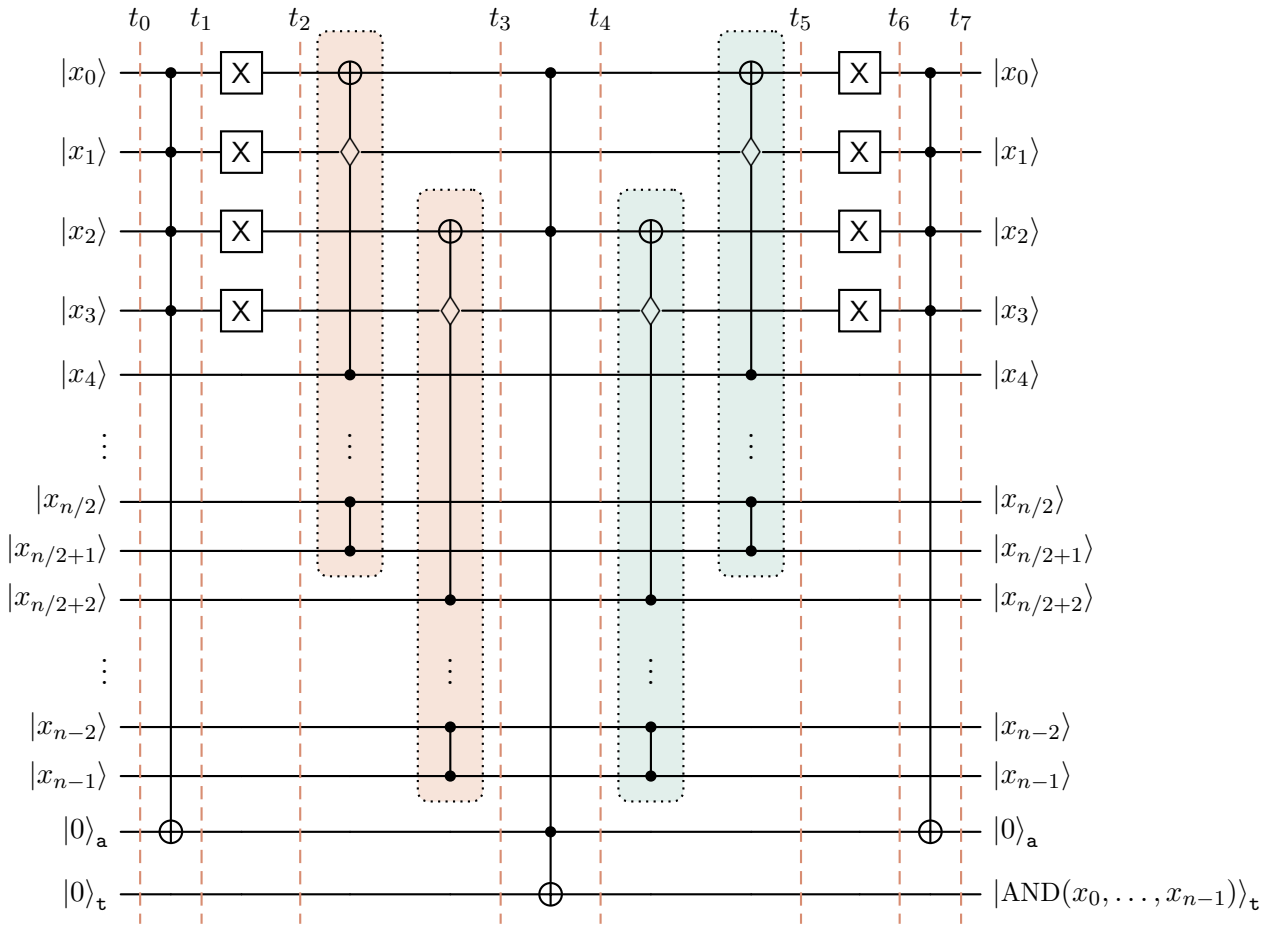


Figure B.1: Recursive implementation of an $\mathsf{AND}^{(n)}$-gate using one ancilla qubit, following the circuit in Figure 3 in [NZS24]. Boxed gates indicate AND-gates that are applied recursively. In particular, the red-colored boxes indicate that the AND-gate is implemented up to step t_4 (before the green boxes), while the green-colored boxes indicate the AND-gate is implemented from t_4 onwards. A '$\diamondsuit$' indicates that the qubit is used as an ancilla qubit in this recursive gate. We refer to [NZS24] for correctness.

Here, b_3, b_4, b_5 are integers such that $b_3 + b_4 + b_5 = 2^k$. For the right half of the circuit, the analysis is the same, except for the $P(\mathsf{AND}^{(4)})$ and $P_i(\mathsf{AND}^{(4)})$ terms that are only part of the left half:

$$P(\mathsf{AND}^{(n)}, \text{recursive, right}) = \left((p_s)^4 P(\mathsf{AND}^{(5)})\right)^{2^k-1} P_i(\mathsf{AND}^{(5)})^{nk-2(2^k-1)} (p_{i_s})^{nk-2(2^k-1)} \\ (P(\mathsf{AND}^{(3)}) P_i(\mathsf{AND}^{(3)}))^{b_3} (P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)}))^{b_4} (P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)}))^{b_5}. \tag{B.4}$$

Together, this gives a success probability of

$$\begin{aligned} P(\mathsf{AND}^{(n)}, \text{recursive}) &= P(\mathsf{AND}^{(n)}, \text{recursive, left}) P(\mathsf{AND}^{(n)}, \text{recursive, right}) \\ &= \left((p_s)^8 P(\mathsf{AND}^{(4)}) P(\mathsf{AND}^{(5)})^2\right)^{2^k-1} \left(P_i(\mathsf{AND}^{(5)})^2 (p_{i_s})^2\right)^{nk-3(2^k-1)} \\ &\quad \cdot P_i(\mathsf{AND}^{(4)})^{nk-2(2^k-1)} (P(\mathsf{AND}^{(3)}) P_i(\mathsf{AND}^{(3)}))^{2b_3} \\ &\quad (P(\mathsf{AND}^{(4)}) P_i(\mathsf{AND}^{(4)}))^{2b_4} (P(\mathsf{AND}^{(5)}) P_i(\mathsf{AND}^{(5)}))^{2b_5}. \end{aligned} \tag{B.5}$$

For each n, the OR-gate implementation uses more gates than the recursive implementation. Therefore, it makes sense to compare the two implementations in a similar to the comparison of non-adaptive approaches. In that sense, for $\varepsilon > 0$, if $(p_d) \gtrsim (1+\varepsilon)(p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$ we have

$$P(\mathsf{AND}^{(n)}, \mathsf{OR}) \gtrsim (1+\varepsilon) P(\mathsf{AND}^{(n)}, \text{recursive}). \tag{B.6}$$

Figure B.2 shows $\gamma_1(n)/\gamma_2(n)$ plotted for $n \leq 1000$. A clear advantage for one of the two methods is not

observed. In line with the other topologies, we will use the OR-gate implementation in subsequent analyses.

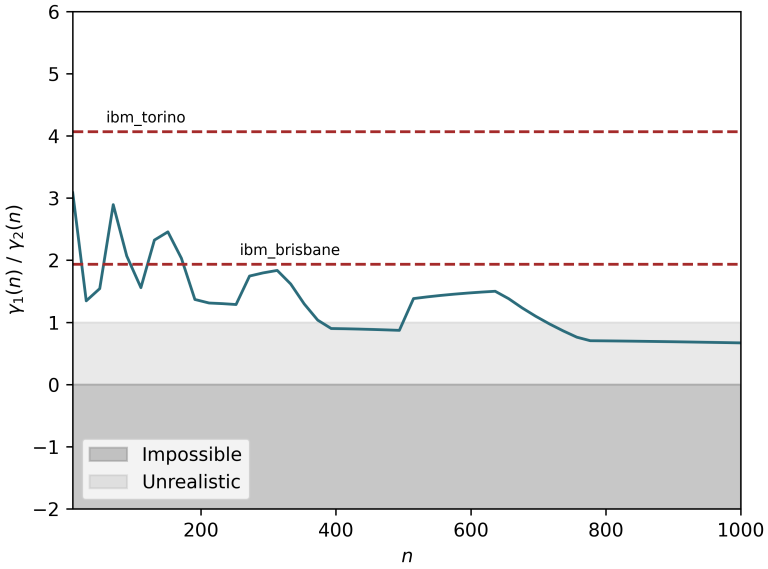


Figure B.2: Comparison of the OR-gate implementation and recursive implementation of the $\mathsf{AND}^{(n)}$-gate, both assuming a non-adaptive all-to-all topology. For $n \leq 1000$, no clear advantage for one of the two methods is observed.

Appendix C

Some deferred proofs from Chapter 5

C.1 Theorem 10

Theorem 10. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\log(n)), \tag{C.1}$$

then

$$P(\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\mathsf{PERM}^{(n,d)}, \text{all-to-all}).$$

(ii) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\sqrt{n} + \frac{\sqrt{d}}{\log(n)}), \tag{C.2}$$

then

$$P(\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\mathsf{PERM}^{(n,d)}, \text{2D grid}).$$

Asymptotically, the gate counts for the LAQCC approach are

$$\begin{aligned} |\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}|_{d+m} &= 4n \cdot \mathcal{O}(|\mathsf{FO}^{(d)}, \mathsf{LAQCC}|_{d+m}) \\ &\quad + 2d \cdot \mathcal{O}(|\mathsf{EQ}^{(n+1)}, \mathsf{LAQCC}|_{d+m}) \\ &\quad + d \cdot \mathcal{O}(|\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_{d+m}) \\ &= 4n \cdot \mathcal{O}(d) + 2d \cdot \mathcal{O}(n\log(n)) + d \cdot \mathcal{O}(n) \\ &\in \mathcal{O}(dn\log(n)) \end{aligned} \tag{C.3}$$

and the idling counts are

$$\begin{aligned} |\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}|_{i_d+i_m+i_c} &= 4n \cdot \mathcal{O}(|\mathsf{FO}^{(d)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}) \\ &\quad + 2d \cdot \mathcal{O}(|\mathsf{EQ}^{(n+1)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}) \\ &\quad + d \cdot \mathcal{O}(|\mathsf{FO}^{(n)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}) \\ &= 4n \cdot \mathcal{O}(d) + 2d \cdot \mathcal{O}(n\log(n)) + d \cdot \mathcal{O}(n) \\ &\in \mathcal{O}(dn\log(n)). \end{aligned} \tag{C.4}$$

Similarly, for an all-to-all topology, we have

$$|\mathsf{PERM}^{(n,d)}, \text{all-to-all}|_{d+m} \in \mathcal{O}(dn\log(n)) \tag{C.5}$$

and

$$|\mathsf{PERM}^{(n,d)}, \text{all-to-all}|_{i_d+i_m+i_c} \in \mathcal{O}(dn\log^2(n)). \tag{C.6}$$

For a 2D grid topology, we have

$$|\mathsf{PERM}^{(n,d)}, \text{2D grid}|_{d+m} \in \mathcal{O}(dn\log(n)) \tag{C.7}$$

and

$$|\mathsf{PERM}^{(n,d)}, \text{2D grid}|_{i_d+i_m+i_c} \in \mathcal{O}(dn\sqrt{n}\log(n) + dn\sqrt{d}). \tag{C.8}$$

Comparing the success probabilities of the all-to-all topology and the LAQCC approach for the PERM-gate implementation, we see that

$$P(\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{PERM}^{(n,d)}, \text{All-to-all}) \tag{C.9}$$

holds when

$$(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}, \tag{C.10}$$

where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} = \frac{|\mathsf{PERM}^{(n,d)}, \text{all-to-all}|_{i_d} - |\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\mathsf{PERM}^{(n,d)}, \mathsf{LAQCC}|_{d+m} - |\mathsf{PERM}^{(n,d)}, \text{all-to-all}|_d}. \tag{C.11}$$

From the terms in Equation (C.11), we see that $\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\log(n))$ and the first claim of Theorem 10 follows.

Comparing the success probabilities of the 2D grid topology and the LAQCC approach for the PERM-gate implementation, we see that

$$P(\mathsf{PERM}^{(n)}, \mathsf{LAQCC}) \gtrsim P(\mathsf{PERM}^{(n)}, \text{2D grid}) \tag{C.12}$$

holds when

$$(p_d) \gtrsim (p_{i_d})^{\gamma(n,d)}, \tag{C.13}$$

where

$$\gamma(n,d) = \frac{|\mathsf{PERM}^{(n)}, \text{2D grid}|_{i_d} - |\mathsf{PERM}^{(n)}, \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\mathsf{PERM}^{(n)}, \mathsf{LAQCC}|_{d+m} - |\mathsf{PERM}^{(n)}, \text{2D grid}|_d}. \tag{C.14}$$

From the terms in Equation (C.11), we see that $\gamma(n,d) \in \mathcal{O}(\sqrt{n} + \frac{\sqrt{d}}{\log(n)})$ and the second claim of Theorem 10 follows.

C.2 Theorem 11

Theorem 11. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\log(n)), \tag{C.15}$$

then

$$P(\text{QSP}^{(n)}, \text{UCG-based}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{QSP}^{(n)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(\sqrt{n}), \tag{C.16}$$

then

$$P(\text{QSP}^{(n)}, \text{UCG-based}, \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{QSP}^{(n)}, \text{UCG-based, 2D grid}).$$

From adding the gate counts of both UCG implementations, we can infer that they are asymptotically the same across all considered topologies:

$$|\mathsf{UCG}^{(n)}, \text{parallelized}|_{d+m} = |\mathsf{UCG}^{(n)}, \text{sequential}|_{d+m} \in \mathcal{O}(2^n n \log(n)) \tag{C.17}$$

This equality stems from the fact that the gate term count is dominated by the gate counts of the 2^n $\mathsf{EQ}^{(n)}$-gates present in both implementations. For the same reason, the idling terms of the two UCG implementation do not differ either:

$$\begin{aligned}
|\mathsf{UCG}^{(n)}, \text{parallelized, } \mathsf{LAQCC}|_{i_d+i_m+i_c} &= |\mathsf{UCG}^{(n)}, \text{sequential, } \mathsf{LAQCC}|_{i_d+i_m+i_c} \in \mathcal{O}(2^n n \log(n)) \\
|\mathsf{UCG}^{(n)}, \text{parallelized, all-to-all}|_{i_d+i_m+i_c} &= |\mathsf{UCG}^{(n)}, \text{sequential, all-to-all}|_{i_d+i_m+i_c} \in \mathcal{O}(2^n n \log^2(n)) \\
|\mathsf{UCG}^{(n)}, \text{parallelized, 2D grid}|_{i_d+i_m+i_c} &= |\mathsf{UCG}^{(n)}, \text{sequential, 2D grid}|_{i_d+i_m+i_c} \in \mathcal{O}(2^n n \sqrt{n} \log(n))
\end{aligned} \tag{C.18}$$

For the full UCG-based QSP protocol, we thus get[1]

$$|\mathrm{QSP}^{(n)}, \text{UCG-based}|_{d+m} = |\mathrm{QSP}^{(n)}, \text{UCG-based}|_{d+m} \in \mathcal{O}(2^n n^2 \log(n)) \tag{C.19}$$

and

$$\begin{aligned}
|\mathrm{QSP}^{(n)}, \text{UCG-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c} &\in \mathcal{O}(2^n n^2 \log(n)) \\
|\mathrm{QSP}^{(n)}, \text{UCG-based, all-to-all}|_{i_d+i_m+i_c} &\in \mathcal{O}(2^n n^2 \log^2(n)) \\
|\mathrm{QSP}^{(n)}, \text{UCG-based, 2D grid}|_{i_d+i_m+i_c} &\in \mathcal{O}(2^n n^2 \sqrt{n} \log(n))
\end{aligned} \tag{C.20}$$

Part (i) of Theorem 11 now follows since

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{|\mathrm{QSP}^{(n)}, \text{UCG-based, all-to-all}|_{i_d+i_m+i_c} - |\mathrm{QSP}^{(n)}, \text{UCG-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\mathrm{QSP}^{(n)}, \text{UCG-based, } \mathsf{LAQCC}|_{d+m} - |\mathrm{QSP}^{(n)}, \text{UCG-based, all-to-all}|_{d+m}} \in \mathcal{O}(\log(n)), \tag{C.21}$$

and part (ii) of Theorem 11 follows since

$$\frac{\gamma_1(n)}{\gamma_2(n)} = \frac{|\mathrm{QSP}^{(n)}, \text{UCG-based, 2D grid}|_{i_d+i_m+i_c} - |\mathrm{QSP}^{(n)}, \text{UCG-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\mathrm{QSP}^{(n)}, \text{UCG-based, } \mathsf{LAQCC}|_{d+m} - |\mathrm{QSP}^{(n)}, \text{UCG-based, 2D grid}|_{d+m}} \in \mathcal{O}(\sqrt{n}). \tag{C.22}$$

C.3 Theorem 12

Theorem 12. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}\left(\frac{\log^2(d)\log^2(\log(d))}{n\log(n)} + \log(n)\right), \tag{C.23}$$

then

$$P(\text{sparse } \mathrm{QSP}^{(n,d)}, \text{UCG-based, } \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse } \mathrm{QSP}^{(n,d)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}\left(\frac{\log^2(d)\sqrt{\log(d)}\log\log(d)}{n\log(n)} + \frac{\sqrt{d}}{\log(n)} + \sqrt{n}\right) \tag{C.24}$$

then

$$P(\text{sparse } \mathrm{QSP}^{(n,d)}, \text{UCG-based, 2D grid}) \gtrsim (1+\varepsilon)^{\gamma_2(n,d)} P(\text{sparse } \mathrm{QSP}^{(n,d)}, \text{UCG-based, 2D grid}).$$

[1] To determine the asymptotic growth of the idling term counts for the full UCG-based QSP protocol, we actually also need to infer the idling term count of a qubit staying idle during one UCG implementation. Due to the difference in circuit depth between the two implementations, we *do* see a difference here. However, this does not influence the total asymptotic upper bound. We omit the calculation.

By Equation (5.105), the gate and idling term counts for the sparse UCG-based QSP protocol can be determined as follows:

$$|\text{sparse QSP}^{(n,d)}, \text{UCG-based}| = |\text{sparse QSP}^{(d)}, \text{UCG-based}| + |\mathsf{PERM}^{(n,d)}|. \tag{C.25}$$

Theorem 12 now follows from the results from the Section C.1 and C.2 on the gate and idling term counts for the general UCG-based QSP protocol and the Permutation-gate. In particular, part (i) follows since

$$\begin{aligned}\frac{\gamma_1(n,d)}{\gamma_2(n,d)} &= \frac{|\text{sparse QSP}^{(n,d)}, \text{UCG-based, all-to-all}|_{i_d+i_m+i_c} - |\text{sparse QSP}^{(n,d)}, \text{UCG-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\text{sparse QSP}^{(n,d)}, \text{UCG-based, } \mathsf{LAQCC}|_{d+m} - |\text{sparse QSP}^{(n,d)}, \text{UCG-based, all-to-all}|_{d+m}} \\ &\in \mathcal{O}\left(\frac{d\log^2(d)\log^2(\log(d)) + dn\log^2(n)}{dn\log(n)}\right) = \mathcal{O}\left(\frac{\log^2(d)\log^2(\log(d))}{n\log(n)} + \log(n)\right),\end{aligned} \tag{C.26}$$

and part (ii) since

$$\begin{aligned}\frac{\gamma_1(n,d)}{\gamma_2(n,d)} &= \frac{|\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}|_{i_d+i_m+i_c} - |\text{sparse QSP}^{(n,d)}, \text{UCG-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c}}{|\text{sparse QSP}^{(n,d)}, \text{UCG-based, } \mathsf{LAQCC}|_{d+m} - |\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}|_{d+m}} \\ &\in \mathcal{O}(\frac{d\log^2(d)\sqrt{\log(d)}\log\log(d) + dn\sqrt{d} + dn\sqrt{n}\log(n)}{dn\log(n)}) \\ &= \mathcal{O}\left(\frac{\log^2(d)\sqrt{\log(d)}\log\log(d)}{n\log(n)} + \frac{\sqrt{d}}{\log(n)} + \sqrt{n}\right).\end{aligned} \tag{C.27}$$

C.4 Theorem 13

Theorem 14. *For any constant $\varepsilon > 0$, the following hold:*

(i) if $d \in \mathcal{O}(n\log^2(n))$, and $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n,d)/\gamma_2(n,d)}$, where

$$\frac{\gamma_1(n,d)}{\gamma_2(n,d)} \in \mathcal{O}(\log(n)), \tag{C.28}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{unary-based, } \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, all-to-all}).$$

(ii) if $(p_d) \gtrsim (p_{i_d})^{\gamma_1(n)/\gamma_2(n)}$, where

$$\frac{\gamma_1(n)}{\gamma_2(n)} \in \mathcal{O}(d + n\sqrt{n}\log(n) + n\sqrt{d}), \tag{C.29}$$

then

$$P(\text{sparse QSP}^{(n,d)}, \text{unary-based, } \mathsf{LAQCC}) \gtrsim (1+\varepsilon)^{\gamma_2(n)} P(\text{sparse QSP}^{(n,d)}, \text{UCG-based, 2D grid}).$$

Using the idling term counts of the data loading procedure (Equation (5.110) and (5.111)) and of Lemma 14 (Equation (4.37)), we see that

$$|\text{sparse QSP}^{(n,d)}, \text{unary-based, } \mathsf{LAQCC}|_{i_d+i_m+i_c} \in \mathcal{O}(d^2) + \mathcal{O}(dn\log(n)), \tag{C.30}$$

$$|\text{sparse QSP}^{(n,d)}, \text{unary-based, all-to-all}|_{i_d+i_m+i_c} \in \mathcal{O}(d\log(d)) + \mathcal{O}(dn\log^2(n)) \tag{C.31}$$

$$|\text{sparse QSP}^{(n,d)}, \text{unary-based, 2D grid}|_{i_d+i_m+i_c} \in \mathcal{O}(d^2) + \mathcal{O}(dn\sqrt{n}\log(n) + dn\sqrt{d}). \tag{C.32}$$

Comparing Equation (C.30) and (C.31), we see that, asymptotically, the idling term count of the LAQCC approach grows faster asymptotically when $d \in \omega(n \log^2(n))$. For $d \in \mathcal{O}(n \log^2(n))$, we get

$$\begin{aligned}\frac{\gamma_1(n,d)}{\gamma_2(n,d)} &= \frac{|\text{sparse QSP}^{(n,d)}, \text{unary-based, all-to-all}|_{i_d+i_m+i_c} - |\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}|_{i_d+i_m+i_c}}{|\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}|_{d+m} - |\text{sparse QSP}^{(n,d)}, \text{unary-based, all-to-all}|_{d+m}} \\ &\in \mathcal{O}\left(\frac{dn\log^2(n)}{dn\log(n)}\right),\end{aligned} \tag{C.33}$$

proving part (i) of Theorem 13. Similarly, for 2D grid connectivity we get

$$\begin{aligned}\frac{\gamma_1(n,d)}{\gamma_2(n,d)} &= \frac{|\text{sparse QSP}^{(n,d)}, \text{unary-based, 2D grid}|_{i_d+i_m+i_c} - |\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}|_{i_d+i_m+i_c}}{|\text{sparse QSP}^{(n,d)}, \text{unary-based, LAQCC}|_{d+m} - |\text{sparse QSP}^{(n,d)}, \text{unary-based, 2D grid}|_{d+m}} \\ &\in \mathcal{O}\left(\log(n)\right),\end{aligned} \tag{C.34}$$

proving part (ii) of Theorem 13.