Optimal Quantum Circuits for Nearest-Neighbor Architectures

David J. Rosenbaum

University of Washington
Department of Computer Science & Engineering
djr@cs.washington.edu

Abstract

We show that the depth of quantum circuits in the realistic architecture where a classical controller determines which local interactions to apply on the kD grid $\mathbb{Z}^k$ where $k \geq 2$ is the same (up to a constant factor) as in the standard model where arbitrary interactions are allowed. This allows minimum-depth circuits (up to a constant factor) for the nearest-neighbor architecture to be obtained from minimum-depth circuits in the standard abstract model. Our work therefore justifies the standard assumption that interactions can be performed between arbitrary pairs of qubits. In particular, our results imply that Shor's algorithm, controlled operations and fanouts can be implemented in constant depth, polynomial size and polynomial width in this architecture.

We also present optimal non-adaptive quantum circuits for controlled operations and fanouts on a kD grid. These circuits have depth $\Theta(\sqrt[k]{n})$, size $\Theta(n)$ and width $\Theta(n)$. Our lower bound also applies to a more general class of operations.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases 2D, Nearest Neighbor, Quantum Architecture, Quantum Complexity, Quantum Computation

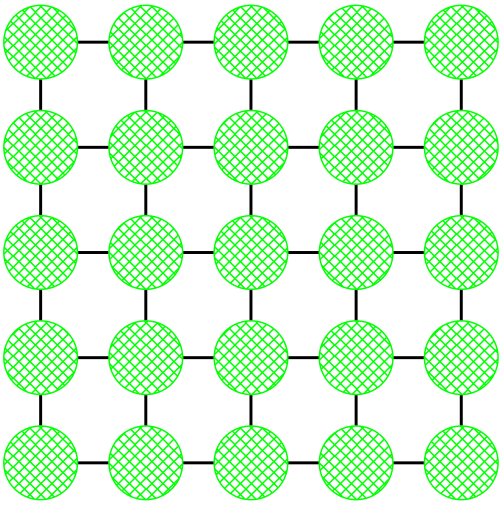
Digital Object Identifier 10.4230/LIPIcs.TQC.2013.294

1 Introduction

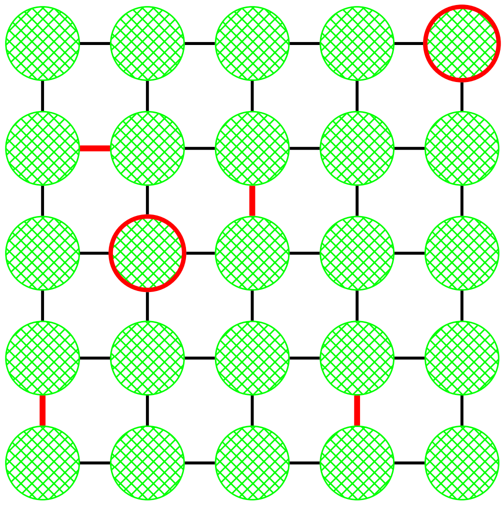
Quantum algorithms are typically formulated at an abstract level and allow arbitrary one- and two-qubit interactions. However, in physical implementations of quantum computers, typically only local interactions between neighboring qubits are possible. This motivates the kD *nearest-neighbor two-qubit concurrent* (kD NTC) architecture [19] (cf. [5]) in which the qubits are arranged on the kD grid $\mathbb{Z}^k$; this is shown in Figure 1a for the case where $k = 2$. Operations may involve one or two qubits with the restriction that two-qubit operations may only be performed along an edge in the grid. Multiple operations may be performed concurrently as long as they are on disjoint sets of qubits; an example is shown in Figure 1b.

The idea of using a classical controller to determine which operations to apply at each step is implicit in the pre- and post-processing stages of Shor's algorithm [16] and is often assumed for fault-tolerant quantum computation. Since the classical controller can take intermediate measurement outcomes into account, this model includes the class of adaptive quantum circuits as a special case. It is potentially even more powerful since the classical controller can perform randomized polynomial-time computations to determine which operations to apply as well as perform pre- and post-processing. Since quantum operations are far more expensive than classical operations, we are primarily concerned with the depth of the quantum circuit and do not count the operations performed by the classical controller as long as they take polynomial time.

8th Conference on Theory of Quantum Computation, Communication and Cryptography.
Editors: Simone Severini and Fernando Brandao; pp. 294–307
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) Interactions in the 2D NTC architecture: the grid lines indicate the two-qubit interactions which can be performed



(b) An example of concurrent interactions in the 2D NTC architecture: the components connected by the thick red edges indicate concurrent interactions and the thick red circles indicate single-qubit interactions

Figure 1 The 2D NTC architecture.

In this work, we study both the *classical-controller* kD *NTC* (kD CCNTC) architecture — a classical controller model where interactions are restricted to a kD grid — as well as the *non-adaptive* kD *NTC*[1] (NANTC) architecture where no classical controller is used and the operations applied cannot depend on intermediate measurement outcomes. The CCNTC model ignores the cost of offline computations performed by the classical controller and assumes that there are no classical locality restrictions. This is realistic since the clock rate for a classical computer is much faster than for a quantum computer. Because quantum computers are already forced to be parallel devices in order to perform operations fault tolerantly [1], the total runtime of a quantum circuit is proportional to the depth of the corresponding quantum circuit. The restriction that interactions are between neighbors on a kD grid comes from the underlying physical device: in most technologies, only qubits that are spatially close can interact.

We first show how to simulate the standard *classical controller abstract concurrent* (CCAC) architecture in kD CCNTC with constant factor overhead in the depth. We accomplish this using a 2D CCNTC teleportation scheme that allows arbitrary interactions on disjoint sets of qubits to be performed in constant depth.

▶ **Theorem 1.1.** *Suppose that C is a CCAC quantum circuit with depth d, size s and width n. Then C can be simulated in $O(d)$ depth, $O(sn)$ size and n^2 width in* 2D *CCNTC.*

This result justifies the standard assumption that non-local interactions can be performed efficiently. Simulating each of the d timesteps from the CCAC circuit in 2D CCNTC requires an $O(n)$ time classical computation; this can be reduced to $O(\log n)$ time if the classical controller is a parallel device or if it includes a simple classical circuit. Since the clock

[1] The original NTC architecture described by Van Meter and Itoh [19] is in fact NANTC; however, we prefer NANTC to avoid confusion with CCNTC where a classical controller is used.

speeds of classical devices are currently much faster than those of quantum devices, this overhead is not likely to be significant.

▶ **Corollary 1.2.** *Let $\mathcal{E}$ be a quantum operation on n qubits. Let d_1 and d_2 be the minimum depths*[2] *required to implement $\mathcal{E}$ with error at most ϵ using $\mathsf{poly}(n)$ size and $\mathsf{poly}(n)$ width in the CCAC and kD CCNTC models respectively where $k \geq 2$. Then $d_1 = \Theta(d_2)$.*

It is possible to implement Shor's algorithm [16] in constant depth in CCAC [3] which implies that it can also be implemented in constant depth in 2D CCNTC.

▶ **Corollary 1.3.** *Shor's algorithm can be implemented in constant depth, polynomial size and polynomial width in* 2D *CCNTC.*

Since controlled-U operations and fanouts can also be performed in constant depth and polynomial width in CCAC [8, 3, 17], we also have the following corollary.

▶ **Corollary 1.4.** *Controlled-U operations with n controls and fanouts with n targets can be implemented in constant depth, $\mathsf{poly}(n)$ size and $\mathsf{poly}(n)$ width in* 2D *CCNTC.*

Our main technical result allows any subset of qubits to be reordered in constant depth. Theorem 1.1 follows from this as a corollary.

▶ **Theorem 1.5.** *Suppose we have an $n \times n$ grid where all qubits except those in the first column are in the state $|0\rangle$. Let $T \subseteq \{0, \ldots, n-1\}$ and let $\pi : T \to \{0, \ldots, n-1\}$ be an injection such that for all $j \in T$ with $\pi(j) = 0$, $\{k \in T^c \mid k < j\} = \emptyset$. Set $m = |\{j \in T \mid \pi(j) \neq 0\}|$. Then we can move each qubit at $(0, j)$ to $(\pi(j), 0)$ for all $j \in T$ in $O(1)$ depth, $O(mn)$ size and $(m+1)n \leq n^2$ width in* 2D *CCNTC.*

Upper bounds for the depth of quantum circuits when converting between various architectures with no classical controller were previously studied by Cheung, Maslov and Severini [4]. Their results imply that CCAC can be simulated in kD CCNTC with $O(\sqrt[k]{n})$ factor depth overhead, $O(n)$ size overhead and no width overhead. In contrast to our results, their techniques are based on applying swap gates to move the interacting qubits next to each other and do not perform any measurements.

Implementations of Shor's algorithm in kD CCNTC with various super-constant depths were previously known for $k = 1$ and $k = 2$. Fowler, Devitt and Hollenberg [7] showed a 1D CCNTC circuit for Shor's algorithm which requires $O(n^3)$ depth, $O(n^4)$ size and $O(n)$ width where n is the number of bits in the integer which is being factored. Maslov [10] showed that any stabilizer circuit can be implemented in linear depth in 1D CCNTC from which the result of Fowler, Devitt and Hollenberg [7] can be recovered. Kutin [9] gave a more efficient 1D CCNTC circuit which uses $O(n^2)$ depth, $O(n^3)$ size and $O(n)$ width. For 2D CCNTC, Pham and Svore [12] showed an implementation of Shor's algorithm in polylogarithmic depth, polynomial size and polynomial width.

It was also previously known that controlled-U operations and fanouts can be implemented in constant depth, polynomial size and polynomial width in CCAC. This line of work was started by Moore [11] who showed that parity and fanout are equivalent and posed the question of whether fanout has constant-depth circuits. Høyer and Špalek [8] proved that if fanout has constant-depth circuits then controlled-U operations can also be implemented

[2] Here, we assume that there is a minimum depth required to implement $\mathcal{E}$ in CCAC when the size and width are $\mathsf{poly}(n)$.

in constant depth with inverse polynomial error. Browne, Kashefi and Predrix [3] showed that one-way quantum computation is equivalent to unitary quantum circuits with fanout. A consequence of this is that constant depth adaptive circuits for fanout can be used to implement controlled-U operations in constant depth in CCAC. Takahashi and Tani [17] reduced the size of this circuit by a polynomial and made it exact.

In many technologies, measurements are much more costly than unitary operations. For this reason, we also consider the non-adaptive kD NANTC model. Here, there is no classical controller and the operations applied depend only on the size of the input and not on intermediate measurement outcomes. Our result in this model is a characterization of the complexity of controlled-U operations and fanouts.

▶ **Theorem 1.6.** *The depth required for controlled-U operations with n controls and fanouts with n targets in kD NANTC is $\Theta(\sqrt[k]{n})$. Moreover, this depth can be achieved with size $\Theta(n)$ and width $\Theta(n)$.*

If the clock speeds of the quantum computer and its classical controller are comparable, then operations implemented using Theorem 1.6 are significantly faster than those implemented using Corollary 1.4. For this reason, Theorem 1.6 may become a better option as quantum computing technology matures.

The layout of our paper is as follows. In Section 2, we discuss definitions used in the rest of the paper and define the models of computation precisely. In Section 3, we review quantum teleportation and describe teleportation chains. In Section 4, we describe our 2D teleportation scheme and show that it allows arbitrary interactions to be implemented in constant depth in 2D CCNTC. In Section 5, we show an algorithm that implements controlled-U operations and fanouts for kD NANTC in depth $O(\sqrt[k]{n})$. In Section 6, we describe how our techniques can be applied to obtain kD NANTC quantum circuits for fanout with depth $O(\sqrt[k]{n})$. In Section 7, we prove a matching lower bound for a class of operations that includes controlled-U operations and fanouts.

2 Definitions

The one- and two-qubit operations that can be performed by the hardware are called the *basic operations*. We assume that the basic operations are a *universal gate set* so that any one- or two-qubit unitary can be constructed from the basic operations. We also assume that the basic operations include measurement in the computational basis.

It is useful to distinguish between physical and logical timesteps. During each *physical timestep*, we can perform any set of disjoint basic operations. During a *logical timestep*, we allow any set of disjoint t-qubit operations to be performed. In this work, we take $t = O(k)$ and assume k is constant.

▶ **Definition 2.1** (NANTC). *In the kD NANTC model, computation is performed by applying a sequence of sets of basic operations $S_1, \ldots, S_d$ to the kD grid of qubits. We require that the operations in the set S_i are disjoint and are either single-qubit operations or two-qubit operations between neighbors in the kD grid. The sequence of sets of operations must be randomized polynomial-time computable from the size n of the input.*

In the models where a classical controller is present, the classical controller is invoked after each physical timestep to determine which operations to apply at the next step.

▶ **Definition 2.2** (CCAC). *Let M be a randomized polynomial-time machine that takes the input x and the measurement outcomes from the first i physical timesteps and outputs a* *set $M_1, \ldots, M_\ell$ of disjoint basic operations to be applied to the qubits at the $i+1^{\text{th}}$ physical timestep. If no more physical timesteps are to be performed, then M outputs the special symbol $\boxdot$. Computation in the CCAC model is performed at physical timestep i by using M to compute the set of operations to apply and then applying them to the qubits.*

The CCNTC model is similar except that it also requires that two-qubit operations are only performed between neighbors on the kD grid.

▶ **Definition 2.3** (CCNTC). *Let M be a randomized polynomial-time machine that takes the input x and the measurement outcomes from the first i physical timesteps and outputs a set $M_1, \ldots, M_\ell$ of disjoint basic operations to be applied to the kD grid of qubits at the $i+1^{\text{th}}$ physical timestep. We require that each M_i is either a single-qubit operation or a two-qubit operation between neighbors in the kD grid. If no more physical timesteps are to be performed, then M outputs the special symbol $\boxdot$. Computation in the CCNTC model is performed at physical timestep i by using M to compute the set of operations to apply and then applying them to the kD grid of qubits.*

In this paper, the machine M from Definitions 2.2 and 2.3 will be deterministic except for the pre- and post-processing stages of Shor's algorithm.

For NANTC, a *quantum circuit* is the sequence of basic operations $M_1, \ldots, M_\ell$ be applied to the kD grid of qubits. For the CCAC and CCNTC models, a *quantum circuit* is described by the machine M from Definitions 2.2 and 2.3. We now define three standard measures of cost in these models.

▶ **Definition 2.4.** *The depth of a quantum circuit is*
(a) *d for NANTC where $S_1, \ldots, S_d$ is the sequence of operations from Definition 2.1 for an input of size n*
(b) $\max_{x\in\{0,1\}^n} \max_r d_{x,r}$ *for CCAC and CCNTC where $d_{x,r}$ is the number of physical timesteps it takes for the machine M from Definitions 2.2 and 2.3 to output $\boxdot$ when the input is x and the random seed is r. The first max is taken is over all possible inputs x of length n and the second is over all possible random seeds r.*

We note that the depth only changes by a constant factor if we use logical timesteps instead of physical timesteps in the above definition. This is due to our assumption that any operation performed in a logical timestep acts on at most $O(k) = O(1)$ qubits.

▶ **Definition 2.5.** *The size of a quantum circuit is*
(a) $\sum_i |S_i|$ *for NANTC where $S_1, \ldots, S_d$ is the sequence of operations from Definition 2.1 for an input of size n*
(b) $\max_{x\in\{0,1\}^n} \max_r s_{x,r}$ *for CCAC and CCNTC where $S_{x,r}$ is the total number of operations applied when the input is x and the random seed is r. The first max is taken over all possible inputs x of length n and the second is over all possible random seeds r.*

In the next definition, we assume that the qubits are indexed by $\mathbb{N}$ for CCAC.

▶ **Definition 2.6.** *The width of a quantum circuit is*
(a) *the total number of qubits acted on by operations in the sets S_i for NANTC where $S_1, \ldots, S_d$ is the sequence of operations from Definition 2.1 for an input of size n*
(b) $\max_{x\in\{0,1\}^n} |A_x|$ *for CCAC where A_x is the smallest subset of $\mathbb{N}$ such that every qubit acted on is contained in A_x for input x and all random seeds r*
(c) $\max_{x\in\{0,1\}^n} |A_x|$ *for CCNTC where A_x is the smallest hypercube in $\mathbb{Z}^k$ such that every qubit acted on is contained in A_x for input x and all random seeds r*

Typically, the depth is the most important metric to optimize since it is proportional to the amount of time required to execute the quantum operations. The width is also important since the number of qubits is currently quite limited but the size is largely irrelevant. Moreover, if parallelism is properly exploited then we expect the size to be roughly the depth times the width.

3 Quantum teleportation

In this section we review quantum teleportation [2]. As we shall see, teleportation is a useful primitive that allows non-local interactions to be performed in a constant-depth circuit in kD CCNTC. Let us denote the states of the Bell basis by $|\Phi_0\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$, $|\Phi_1\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}}$, $|\Phi_2\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}$ and $|\Phi_3\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}}$. Up to global phase, these can be written as $|\Phi_\ell\rangle^{AB} = \sigma_\ell^B |\Phi_0\rangle^{AB}$. Recall that in the quantum teleportation setting, Alice has a state $|\psi\rangle^S = \alpha|0\rangle^S + \beta|1\rangle^S$ that she wishes to send to Bob. The two parties are not allowed to send quantum states to each other but each have one qubit of a Bell state $\sigma_\ell^B|\Phi_0\rangle$ and can communicate classically.

To perform quantum teleportation, Alice performs a Bell measurement on the SA registers. If the measurement outcome is $|\Phi_k\rangle$, then a simple calculation shows that the resulting state is $|\Phi_k\rangle^{SA} \otimes \sigma_\ell\sigma_k|\psi\rangle^B$. Alice then sends the classical measurement outcome k to Bob; by applying the appropriate Pauli operation to his register B, Bob causes to overall state to become $|\Phi_k\rangle^{SA} \otimes |\psi\rangle^B$. Observe that Alice's state $|\psi\rangle$ has been recovered in Bob's register.

Let us now consider how quantum teleportation chains can be used in the 1D CCNTC model to perform non-local operations in constant depth. Suppose that we have a qubit in the state $|\psi\rangle^S$ along with m Bell states $|\Phi_{\ell_j}\rangle^{A_jB_j}$. These are arranged on a line so that the overall state is $|\psi\rangle^S \bigotimes_{j=1}^m |\Phi_{\ell_j}\rangle^{A_jB_j}$. Our goal is to move qubit S to B_m. One way to do this is to first teleport S to B_1 by performing a Bell measurement on SA_1. We then store the measurement outcome k_1 but do not apply the correcting Pauli operation; at this point, the state of B_1 is $\sigma_{\ell_1}\sigma_{k_1}|\psi\rangle$. Continuing this process, we obtain the state $\bigotimes_{j=1}^m |\Phi_{k_j}\rangle \prod_{j=m}^1 (\sigma_{\ell_j}\sigma_{k_j}) |\psi\rangle^{B_m}$. Since $\prod_{j=m}^1 (\sigma_{\ell_j}\sigma_{k_j})$ is just a Pauli operation, we obtain the state $\bigotimes_{j=1}^m |\Phi_{k_j}\rangle |\psi\rangle^{B_m}$ in a single quantum operation. The crucial point here is that all of the Bell measurements are performed on disjoint pairs of qubits so they can all be done in parallel as in one-way quantum computation [14, 13] and [18]. Thus, we can perform a non-local interaction of arbitrary distance in constant depth. It is important to note that this is not possible without a classical controller since otherwise there is no way to compute the correcting Pauli operation.

4 Depth complexity in kD CCNTC

In this section, we show that an arbitrary set of CCAC interactions corresponding to basic operations can be performed in constant depth in 2D CCNTC. We assume that there are n qubits on which the interactions are to be performed and store these in the first column of a 2D $n \times n$ CCNTC grid. Since we must handle interactions between qubits that are not neighbors, we may as well assume that the original n qubits are stored in the first column. The remaining columns are used as ancillas to implement teleportation chains. We teleport each of the n qubits horizontally to the right so that interacting pairs are in adjacent columns. Since these teleportations are on disjoint sets of qubits, they can be performed in parallel as in [14, 13, 18]. A second set of vertical teleportation chains is then used to move all the qubits down to the first row. At this point, the interacting qubits are neighbors so

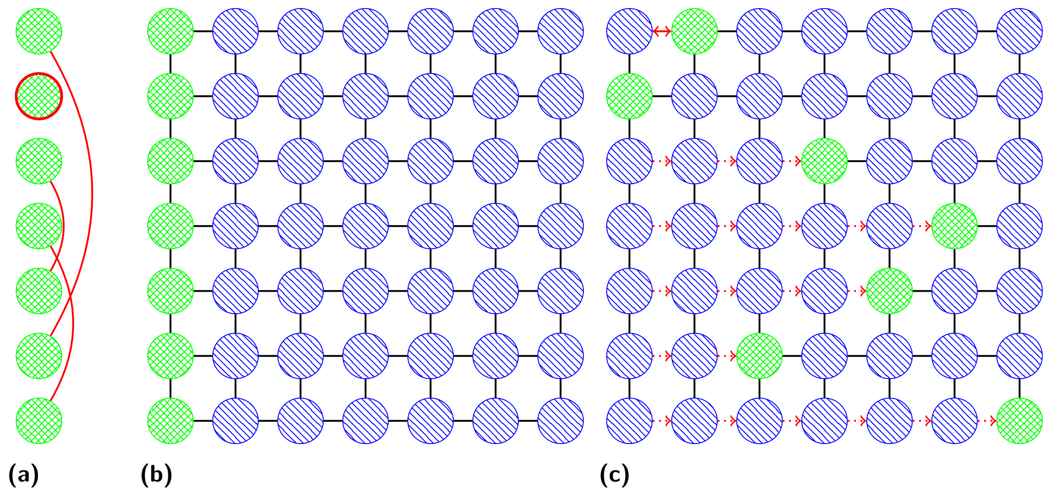


Figure 2 Performing an arbitrary set of interactions in 2D CCNTC. The qubits crosshatched green are the data qubits and the qubits shaded with diagonal downward blue lines are ancilla qubits.

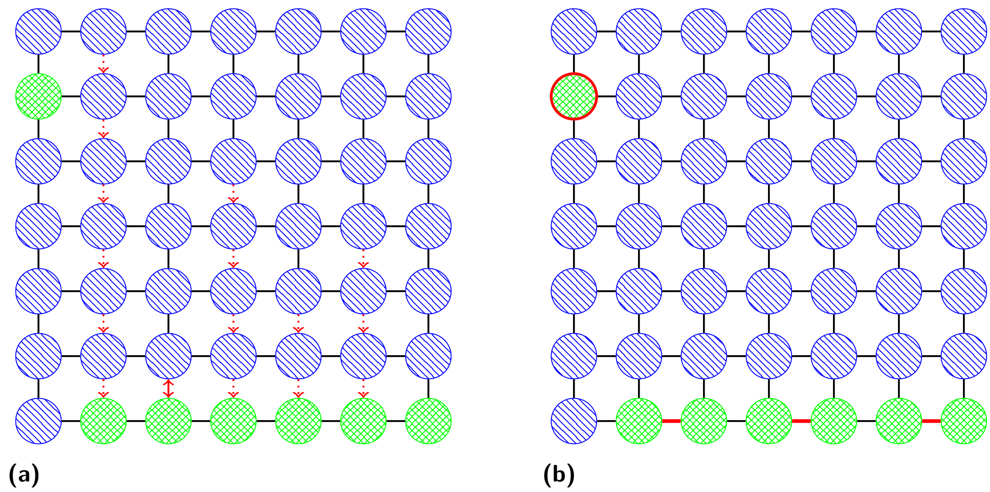


Figure 3 Performing an arbitrary set of interactions in 2D CCNTC.

the interactions may be implemented directly. We then perform the reverse teleportations to move the qubits back to their original positions.

4.1 An example of arbitrary interactions in 2D CCNTC

We show an example in Figure 2. The desired interactions are shown in Figure 2a. The layout of the data qubits in the 2D grid is shown in Figure 2b; the ancilla qubits are used to implement the teleportation chains and are initially set to $|0\rangle$. We start by horizontally teleporting the qubits that interact to adjacent columns in Figure 2c where the teleportation chains are denoted by the dotted red arrows. The red double arrow indicates a swap operation; this is just a less expensive way of achieving the same result when the qubits are neighbors. The next step is to vertically teleport the data qubits down to the first row as shown in Figure 3a. Finally, all interacting qubits are now neighbors so we perform the desired interactions in Figure 3b. The final reverse teleportations are not shown but can be obtained by reversing the arrows in Figures 2c and 3a.

4.2 An algorithm for performing arbitrary interactions in 2D CCNTC

It is easy to generalize the approach of Figure 2 to show that the qubits in the first column can be reordered arbitrarily in constant depth. The pseudocode is the obvious generalization of Figure 2 (see the full version of our paper [15]).

▶ **Theorem 1.5.** *Suppose we have an $n \times n$ grid where all qubits except those in the first column are in the state $|0\rangle$. Let $T \subseteq \{0, \ldots, n-1\}$ and let $\pi : T \to \{0, \ldots, n-1\}$ be an injection such that for all $j \in T$ with $\pi(j) = 0$, $\{k \in T^c \mid k < j\} = \emptyset$. Set $m = |\{j \in T \mid \pi(j) \neq 0\}|$. Then we can move each qubit at $(0, j)$ to $(\pi(j), 0)$ for all $j \in T$ in $O(1)$ depth, $O(mn)$ size and $(m + 1)n \leq n^2$ width in* 2D *CCNTC.*

We note that the teleportation chains in our scheme require an $O(n)$ time classical computation to determine the correcting Pauli matrix (see Section 3). Since this computation simply involves multiplying $O(n)$ Pauli matrices, it can be done more efficiently in $O(\log n)$ time by arranging the multiplications in a binary tree. The $O(\log n)$ runtime requires either that the classical controller is a parallel device or that it includes a special classical circuit for computing the correcting Pauli operation. Since classical operations are much faster than quantum operations on current devices, this overhead is unlikely to be a problem.

From this, it follows that any set of one- and two-qubit operations on disjoint sets of qubits can be performed in constant depth in 2D. This implies that any CCAC circuit can be simulated with constant factor depth overhead in 2D CCNTC.

▶ **Theorem 1.1.** *Suppose that C is a CCAC quantum circuit with depth d, size s and width n. Then C can be simulated in $O(d)$ depth, $O(sn)$ size and n^2 width in* 2D *CCNTC.*

The rest of our results for kD CCNTC follow from Theorem 1.1. Let $\mathcal{D}_n$ denote the set of all $n \times n$ density matrices. A general quantum operation is represented as a completely positive trace preserving (CPTP) map $\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n$. Obviously, any circuit in the 2D CCNTC model can also be applied when arbitrary interactions are allowed. The following corollary is immediate.

▶ **Corollary 1.2** (continuing from p. 296). *Let $\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n$ be a CPTP map and let $\epsilon \geq 0$. Let d_1 and d_2 be the minimum depths required to implement $\mathcal{E}$ with error at most ϵ in the CCAC and kD CCNTC models respectively where $k \geq 2$. Then $d_1 = \Theta(d_2)$.*

It is known that Shor's algorithm can be implemented in constant depth, polynomial size and polynomial width in CCAC [3] from which we obtain another corollary.

▶ **Corollary 1.3.** *Shor's algorithm can be implemented in constant depth, polynomial size and polynomial width in* 2D *CCNTC.*

Because controlled-U operations and fanouts with unbounded numbers of control qubits or targets can be performed in constant depth, polynomial size and polynomial width in CCAC [8, 3, 17], we have the following result.

▶ **Corollary 1.4.** *Controlled-U operations with n controls and fanouts with n targets can be implemented in constant depth,* poly(n) *size and* poly(n) *width in* 2D *CCNTC.*

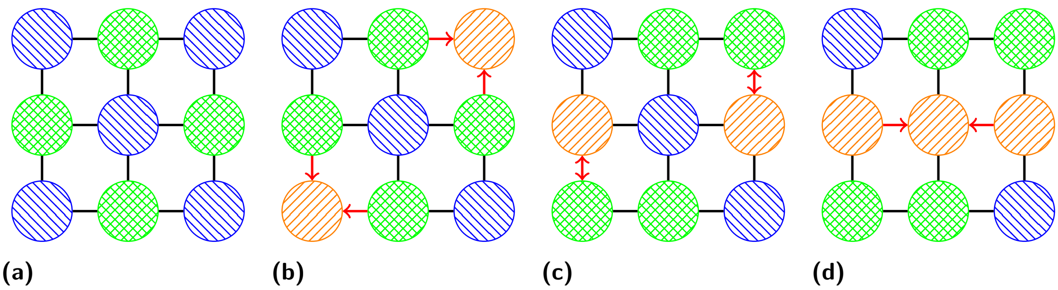


Figure 4 A controlled operation on a 3×3 grid. The qubits crosshatched green are the data qubits, the qubits shaded with diagonal upward orange lines are ancilla qubits which store intermediate data and the qubits shaded with diagonal downward blue lines are ancilla qubits which are currently unused.

5 Controlled operations in kD NANTC

In this section, we show how to control a single-qubit U operation by n controls using $O(\sqrt[k]{n})$ operations in kD NANTC. We start with an $m \times m$ grid; for reasons that will become clear later, we require that m is odd. The control qubits are placed such that they are not at adjacent grid points; the central 3×3 square has no controls except when $m = 3$. This is illustrated in Figures 4a and 5a for the cases where $m = 3$ and $m = 5$. Let c be the center of the grid which corresponds to the target qubit. The circuit works by considering each square ring in the grid with center c (i.e., a set of points in the grid that all have the same distance to the center under the ℓ_∞ norm). We start with the outermost such ring and propagate its control values into the next ring. At each such step, some of the control values are combined so that all the values can fit into the smaller ring. This continues until we reach a 3×3 ring at which point we apply a special sequence of operations to finish applying the controlled operation to the central qubit. Each stage can be implemented in constant depth so the overall depth is $O(\sqrt{n})$.

5.1 The base case: the 3×3 grid

We now describe how this circuit works in greater detail. First, consider the case where $m = 3$. The grid starts as shown in Figure 4a; note that we do not force the central 3×3 square to be devoid of controls in this case since this is the entire grid. All ancilla qubits start in the state $|0\rangle$. We start by setting the lower left and upper right corner ancilla qubits to the ANDs of their neighboring controls as shown in Figure 4b. Both of these operations are disjoint, so this can be done in one logical timestep. The next step is to swap these two corner qubits with the vertical middle qubits so they can interact with the central target qubit; this is done in Figure 4c. Finally, we apply a U operation to the target qubit and control by the two middle qubits in Figure 4d.

At this point, the target qubit has the desired value; however, there are two other ancilla qubits in Figure 4d that must have their values uncomputed. This is done by applying the operations of Figures 4b–c in reverse order.

5.2 An example of the general case: the 5×5 grid

We now consider an example of the general case where $m = 5$ as shown in Figure 5a. The first step is to propagate the values of the outer ring inwards; since the inner ring is 3×3, there are no controls in the inner ring so this can be done as shown in Figure 5b. We then rotate the inner ring as in Figure 5c. At this point, the remaining operations to perform are the same as in the 3×3 case and are shown in Figures 5d–f. At this point the target qubit has the desired value so we uncompute the intermediate ancillas by applying the operations of Figures 5b–e in reverse order.

The same idea applies to an $m \times m$ grid except that when the inner rings have controls (i.e. for $m \geq 7$), the controls from the outer ring must be combined with those in the inner ring at the same time they are propagated inwards. See the full version of our paper [15] for examples of the 7×7 and 9×9 cases.

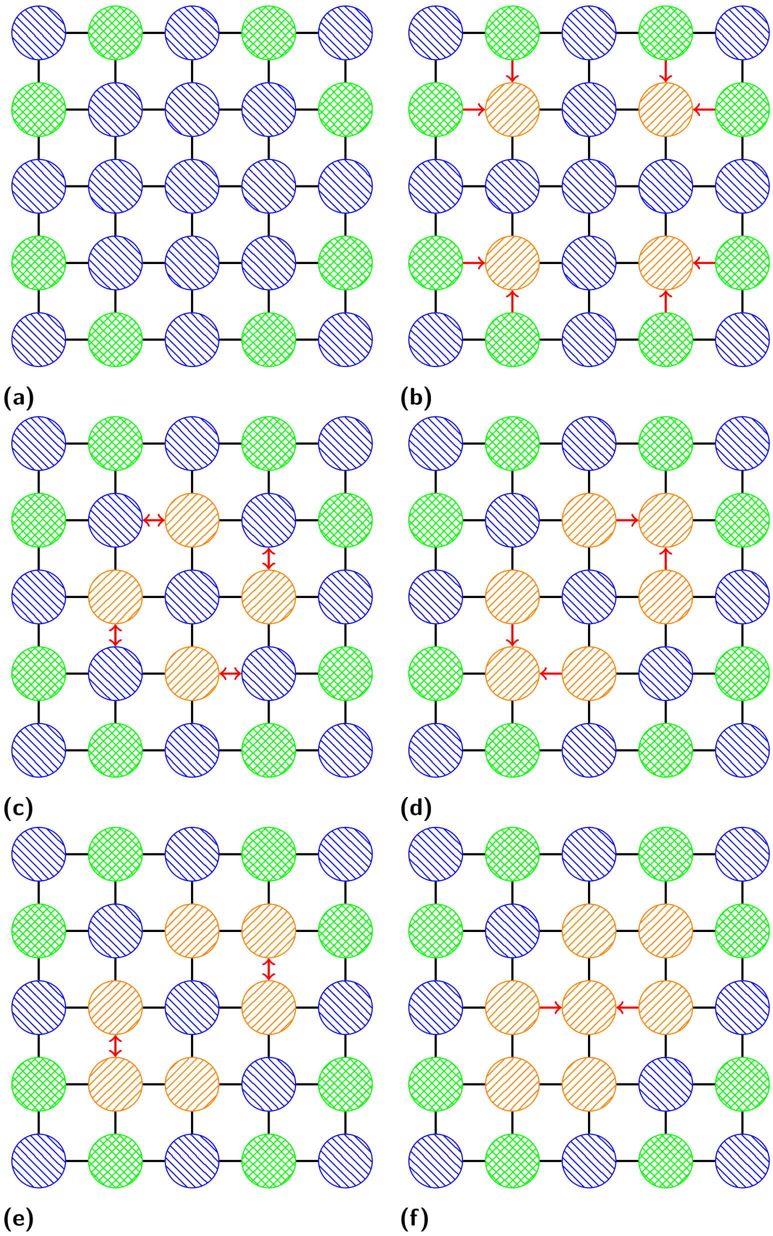


Figure 5 A controlled operation on a 5×5 grid. See Figure 4 for the meaning of the colors and shading used.

5.3 An algorithm for controlled-U operations in $O(\sqrt{n})$ depth in 2D NANTC

We now present the algorithm used in Figures 4 – 5 for the general $m \times m$ grid. Consider an odd $m > 3$. We denote the coordinates of the qubits on this grid by (x, y) where $0 \le x, y < m$. Let G be the set $\{0, \ldots, m-1\}^2$ of all points on the grid and let $c = ((m-1)/2, (m-1)/2)$ be the central point. As discussed previously, the geometry induced by the ℓ_∞ norm is useful for reasoning about this grid. From now on, all distances in this subsection are understood to be with respect to the ℓ_∞ norm.

We will say that the k^{th} *ring* is the set of points that have distance $(m-1)/2 - k$ to c so the zeroth ring is outermost; we denote by $R_k = (r_0^k, \ldots, r_{\ell_k}^k)$ the points of the k^{th} ring where r_0^k is the bottom left corner and the rest of the points are in clockwise order.

The ring R_k contains $4\left(\frac{m-1}{2} - k\right)$ controls so the entire grid has $n = 4\sum_{3<m-2k\le m}\left(\frac{m-1}{2} - k\right) = (1/2)(m^2 - 9/2)$ controls for $m > 3$. In the case where $m = 3$, there are 4 controls. Thus, it is indeed the case that the depth is $O(\sqrt{n})$.

Writing out the explicit pseudocode is straightforward (see the full version of our paper [15]).

From this, we obtain the following theorem.

▶ **Theorem 5.1.** *Controlled-U operations with n controls have depth $O(\sqrt{n})$, size $O(n)$ and width $O(n)$ in* 2D *NANTC.*

5.4 Generalization to kD NANTC

In this section, we discuss how the circuit can be generalized to k dimensions. The algorithm works in the same way except the ring R_k is replaced by the grid points on the surface of the hypercube formed by the points at ℓ_∞ distance $(m-1)/2 - k$ from the center c of the grid. We proceed as before and propagate the controls on R_k into R_{k+1} until we obtain a grid of width 3. Since the number of controls on a kD grid of length m is $O(m^k)$, we obtain a circuit of depth $O(\sqrt[k]{n})$ for implementing a controlled-U operation with n controls. The constant depends on k, but we assumed that k is constant in Section 2. From this, we obtain the following result.

▶ **Theorem 5.2.** *Controlled-U operations with n controls have depth $O(\sqrt[k]{n})$, size $O(n)$ and width $O(n)$ in k*D *NANTC.*

6 Fanout operations

In this section, we describe quantum circuits for fanout. In this case, we have a single control qubit and our goal is to XOR it into each of the target qubits. The construction of fanout circuits is adapted from that of controlled operations; the circuits are the same except that the qubit that was the target becomes the control qubit and qubits that were the controls become the targets. Let n be the number of targets. In the case of the circuit of Section 5, we simply apply all operations in reverse order and replace each Toffoli gate $y \leftarrow y \oplus x_1 \wedge \ldots \wedge x_n$ with a fanout operation $x_j \leftarrow x_j \oplus y$ for all $1 \le j \le n$. This yields a kD NANTC fanout circuit of depth $O(\sqrt[k]{n})$. We have shown the following.

▶ **Theorem 6.1.** *fanouts to n targets have depth $O(\sqrt[k]{n})$, size $O(n)$ and width $O(n)$ in k*D *NANTC.*

7 Optimality

In this section, we prove that the depth, size and width of the circuits of Theorem 5.1 (and its kD generalization) are optimal for NANTC. A similar lower bound for addition is discussed in [6]. These lower bounds hold regardless of where the controls and target qubits are located on the kD grid. They also hold for a more general class of operations that contains the controlled-U operations and fanouts.

Since each qubit is acted on by a constant number of operations in Theorem 5.1, the size of the circuit is $O(n)$. This is clearly optimal since any circuit that implements a controlled operation must act on each of the controls.

▶ **Theorem 7.1.** *Any NANTC quantum circuit that implements a non-trivial controlled-U operation with n controls has size $\Omega(n)$.*

The *trace norm* of a density matrix ρ (denoted $\|\rho\|_{\mathrm{tr}}$) is equal to $(1/2)\,\mathrm{tr}\,|\rho|$ (the $(1/2)$ factor ensures that $\|\rho - \sigma\|_1$ is the probability of distinguishing ρ and σ with the best possible measurement). Consider a general quantum operation $\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n$ represented as a CPTP map. We will use an operator version of the trace norm defined by $\|\mathcal{E}\|_{\mathrm{tr}} = \sup_{\rho \in \mathcal{D}} \|\mathcal{E}(\rho)\|_1$; if $\mathcal{E}_1$ and $\mathcal{E}_2$ are two CPTP maps then $\|\mathcal{E}_1 - \mathcal{E}_2\|_{\mathrm{tr}}$ is the probability of distinguishing between them on the worst possible input. Thus, it is a measure of how much these operations differ. We will also make use of the partial trace. If x is a qubit, then we will denote the partial trace over all qubits except x by $\mathrm{tr}_{\neg x} = \mathrm{tr}_{\mathbb{Z}^k \setminus \{x\}}$.

Controlled-U operations are special case of a more general class of operations.

▶ **Definition 7.2.** *Let $\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n$ be a CPTP map. We say that $\mathcal{E}$ is ϵ-input sensitive if there exists a qubit y such that for $\Omega(n)$ qubits x, there exists a CPTP map $\mathcal{F} : \mathcal{D}_n \to \mathcal{D}_n$ acting only on x such that $\|\mathrm{tr}_{\neg y}(\mathcal{E}\mathcal{F} - \mathcal{E})\|_{\mathrm{tr}} \geq \epsilon$.*

Intuitively, an ϵ-input sensitive operation is a generalization of a Toffoli gate where modifying some input qubit x yields a different value on the output with probability ϵ. Similarly, we can define ϵ-output sensitive operations which are generalizations of fanout.

▶ **Definition 7.3.** *Let $\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n$ be a CPTP map. We say that $\mathcal{E}$ is ϵ-output sensitive if there exists a qubit x such that for $\Omega(n)$ qubits y, there exists a CPTP map $\mathcal{F} : \mathcal{D}_n \to \mathcal{D}_n$ acting only on x such that $\|\mathrm{tr}_{\neg y}(\mathcal{E}\mathcal{F} - \mathcal{E})\|_{\mathrm{tr}} \geq \epsilon$.*

We say that $\mathcal{E}$ is ϵ-*sensitive* if it is ϵ-input or ϵ-output sensitive. A family $\{\mathcal{E} : \mathcal{D}_n \to \mathcal{D}_n\}$ of CPTP maps is ϵ-sensitive if every $\mathcal{E}_n$ is ϵ-sensitive. Our lower bounds will apply to all families of ϵ-sensitive operations. All proofs will be for the case of ϵ-input sensitive operations but the argument of ϵ-output sensitive operations is all but identical.

▶ **Theorem 7.4.** *Let $\{\mathcal{E}_n : \mathcal{D}_n \to \mathcal{D}_n\}$ be a family of ϵ-sensitive operations. Then any family of kD NANTC circuits $\{C_n\}$ such that $\|\mathcal{E}_n - C_n\|_{\mathrm{tr}} < \epsilon/2$ for all n has size $\Omega(n)$.*

Proof. Suppose that C_n has size $o(n)$. Assume $\mathcal{E}_n$ is ϵ-input sensitive and choose a qubit y as in definition Definition 7.2 (the case where it is ϵ-output sensitive is very similar). There are $\Omega(n)$ qubits x such that there exists a CPTP map $\mathcal{F} : \mathcal{D}_n \to \mathcal{D}_n$ acting only on x such that $\|\mathrm{tr}_{\neg y}(\mathcal{E}_n\mathcal{F} - \mathcal{E}_n)\|_{\mathrm{tr}} \geq \epsilon$. For large n, there is such an x which is not acted on by C_n. Then $\mathrm{tr}_{\neg y} C_n\mathcal{F} = \mathrm{tr}_{\neg y} C_n$.Now $\|\mathrm{tr}_{\neg y}(C_n - \mathcal{E}_n)\|_{\mathrm{tr}} \geq \big|\|\mathrm{tr}_{\neg y}(C_n\mathcal{F} - \mathcal{E}_n\mathcal{F})\|_{\mathrm{tr}} - \|\mathrm{tr}_{\neg y}(\mathcal{E}_n\mathcal{F} - \mathcal{E}_n)\|_{\mathrm{tr}}\big|$ $> \epsilon/2$ which is a contradiction. ◀

We call a controlled-U operation *non-trivial* if $U \neq I$. It is easy to prove the following.

▶ **Lemma 7.5.** *Non-trivial controlled-U operations and fanouts are* 1*-sensitive.*

We now obtain a corollary of Theorem 7.4 of which Theorem 7.1 is a special case.

▶ **Corollary 7.6.** *Let* $\{\mathcal{E}_n : \mathcal{D}_n \to \mathcal{D}_n\}$ *denote a family of controlled-U operations or fanouts. Any family of* kD *NANTC circuits* $\{C_n\}$ *such that* $\|C_n - \mathcal{E}_n\|_{\mathrm{tr}} < 1/2$ *has size* $\Omega(n)$*.*

This shows that the circuits of Theorem 5.1 (and its kD generalization) have optimal size. Next, we will show that ϵ-sensitive kD NTC circuits have depth $\Omega(\sqrt[k]{n})$. For this we require the following easy lemma.

▶ **Lemma 7.7.** *For any subset* $S \subseteq \mathbb{Z}^k$ *and any* $x \in \mathbb{Z}^k$*, there exists a subset* $T \subseteq S$ *of size* $\Omega(|S|)$ *such that for all* $y \in T$*,* $\|x - y\|_1 = \Omega(\sqrt[k]{|S|})$*.*

We are now ready to prove our depth lower bound.

▶ **Theorem 7.8.** *Let* $\{\mathcal{E}_n : \mathcal{D}_n \to \mathcal{D}_n\}$ *be a family of* ϵ*-sensitive operations. Then any family of* kD *NANTC circuits* $\{C_n\}$ *such that* $\|\mathcal{E}_n - C_n\|_{\mathrm{tr}} < \epsilon/2$ *for all* n *has depth* $\Omega(\sqrt[k]{n})$*.*

Proof. Suppose $\{C_n\}$ has depth $t = o(\sqrt[k]{n})$. Assume that $\mathcal{E}_n$ is ϵ-input sensitive (the case where it is ϵ-output sensitive is very similar) and choose a qubit y as in Definition 7.2. There is a set S of $\Omega(n)$ qubits such that for each $x \in S$, there exists a CPTP map $\mathcal{F} : \mathcal{D}_n \to \mathcal{D}_n$ acting only on x with $\|\mathrm{tr}_{\neg y}(\mathcal{E}_n\mathcal{F} - \mathcal{E}_n)\|_{\mathrm{tr}} \geq \epsilon$. Let $c > 0$ be the hidden constant in the expression $\Omega(\sqrt[k]{|S|})$ from Lemma 7.7. For sufficiently large n, the depth of C_n is strictly less than $c\sqrt[k]{n}$. Let G_i be the set of disjoint one- and two-qubit operations that are performed at timestep $1 \leq i \leq t$ in C_n. For an operation $M \in G_i$, let us say that M is *active* if

(a) M acts non-trivially on y or

(b) there is an operation $M' \in G_j$ with $i < j \leq t$ such that M' is active and M and M' act non-trivially on a common qubit

Let us say that a qubit x *influences* y if there exists an active operation $M \in G_i$ that acts non-trivially on x. Suppose x influences y after t timesteps. Because all operations act on pairs of adjacent qubits, the ℓ_1 distance between x and y is at most t. By Lemma 7.7, there exists a subset T of S of size $\Omega(n)$ such that $\|x - y\|_1 \geq c\sqrt[k]{n}$ for all $x \in T$. Because $t < c\sqrt[k]{n}$, x does not influence y for $x \in T$. Let us fix some $x \in T$. Choosing a $\mathcal{F}$ acting only on x as in Definition 7.2, we have $\|\mathrm{tr}_{\neg y}(C_n - \mathcal{E}_n)\|_{\mathrm{tr}} \geq \left|\|\mathrm{tr}_{\neg y}(C_n\mathcal{F} - \mathcal{E}_n\mathcal{F})\|_{\mathrm{tr}} - \|\mathrm{tr}_{\neg y}(\mathcal{E}_n\mathcal{F} - \mathcal{E}_n)\|_{\mathrm{tr}}\right| > \epsilon/2$ which is a contradiction. ◀

By Lemma 7.5, we obtain the following corollary.

▶ **Corollary 7.9.** *Let* $\{\mathcal{E}_n : \mathcal{D}_n \to \mathcal{D}_n\}$ *denote a family of controlled-U operations or fanouts. Any family of* kD *NANTC circuits* $\{C_n\}$ *such that* $\|C_n - \mathcal{E}_n\|_{\mathrm{tr}} < 1/2$ *has depth* $\Omega(\sqrt[k]{n})$*.*

From Theorems 5.2 and 6.1 and Corollaries 7.6 and 7.9, we conclude that the circuits of Theorem 5.1 and its kD generalization are optimal in their depth, size and width.

▶ **Theorem 1.6.** *The depth required for controlled-U operations with* n *controls and fanouts with* n *targets in* kD *NANTC is* $\Theta(\sqrt[k]{n})$*. Moreover, this depth can be achieved with size* $\Theta(n)$ *and width* $\Theta(n)$*.*

Acknowledgments. I thank Paul Beame and Aram Harrow for useful discussions and feedback and the anonymous reviewers for helpful comments. Aram Harrow suggested the use of teleportation chains as a primitive. Paul Pham suggested applying the technique of Theorem 5.1 to fanouts. I was funded by the DoD AFOSR through an NDSEG fellowship. Partial support was provided by IARPA under the ORAQL project.

References

1 D. Aharonov, M. Ben-Or, R. Impagliazzo, and N Nisan. Limitations of Noisy Reversible Computation. *arXiv e-prints*, 1996.

2 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. *Physical Review Letters*, 70, 1993.

3 Dan E. Browne, Elham Kashefi, and Simon Perdrix. Computational depth complexity of measurement-based quantum computation. In *In Proceedings of the Fifth Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2010.

4 Donny Cheung, Dmitri Maslov, and Simone Severini. Translation techniques between quantum circuit architectures. In *Workshop on Quantum Information Processing*, 2007.

5 B.-S. Choi and R. Van Meter. An $\Theta(\sqrt{n})$-depth quantum adder on a 2D NTC quantum computer architecture. *arXiv:1008.5093*, 2010.

6 Byung-Soo Choi and Rodney Van Meter. On the effect of quantum interaction distance on quantum addition circuits. *ACM Journal on Emerging Technologies in Computing Systems*, 7:11:1–11:17, 2011.

7 A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg. Implementation of Shor's Algorithm on a Linear Nearest Neighbour Qubit Array. *arXiv e-prints*, 2004.

8 Peter Høyer and Robert Špalek. Quantum fan-out is powerful. *Theory of Computing*, 1:81–103, 2005.

9 S. A. Kutin. Shor's algorithm on a nearest-neighbor machine. *arXiv e-prints*, 2006.

10 D. Maslov. Linear depth stabilizer and quantum Fourier transformation circuits with no auxiliary qubits in finite-neighbor quantum architectures. *arXiv e-prints*, 2007.

11 C. Moore. Quantum Circuits: Fanout, Parity, and Counting. *arXiv e-prints*, 1999.

12 P. Pham and K. M. Svore. A 2D Nearest-Neighbor Quantum Architecture for Factoring. *arXiv e-prints*, 2012.

13 R. Raussendorf, D. E. Browne, and H. J. Briegel. The one-way quantum computer–a non-network model of quantum computation. *arXiv e-prints*, 2002.

14 Robert Raussendorf and Hans J Briegel. A one-way quantum computer. *Physical Review Letters*, 86:5188–5191, 2001.

15 D. Rosenbaum. Optimal Quantum Circuits for Nearest-Neighbor Architectures. *arXiv:1205:0036*, 2012.

16 Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Annual Symposium on Foundations of Computer Science*, 1994.

17 Y. Takahashi and S. Tani. Constant-Depth Exact Quantum Circuits for the OR and Threshold Functions. *arXiv e-prints*, 2011.

18 B. M. Terhal and D. P. DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games. *arXiv e-prints*, 2002.

19 Rodney Van Meter and Kohei M. Itoh. Fast quantum modular exponentiation. *Phys. Rev. A*, 71:052320, 2005.