THEORY OF COMPUTING, Volume 1 (2005), pp. 81–103
http://theoryofcomputing.org

Quantum Fan-out is Powerful

Peter Høyer* Robert Špalek†

Received: October 26, 2004; published: August 3, 2005.

Abstract: We demonstrate that the unbounded fan-out gate is very powerful. Constant-depth polynomial-size quantum circuits with bounded fan-in and unbounded fan-out over a fixed basis (denoted by $\mathrm{QNC}^0_\mathrm{f}$) can approximate with polynomially small error the following gates: parity, mod[q], And, Or, majority, threshold[t], exact[t], and Counting. Classically, we need logarithmic depth even if we can use unbounded fan-in gates. If we allow arbitrary one-qubit gates instead of a fixed basis, then these circuits can also be made exact in log-star depth. Sorting, arithmetic operations, phase estimation, and the quantum Fourier transform with arbitrary moduli can also be approximated in constant depth.

ACM Classification: F.2.1, F.2.2

AMS Classification: 68Q15, 81P68

Key words and phrases: quantum computing, quantum circuits, fan-out, quantum Fourier transform, constant depth circuits, threshold circuits

1 Introduction

In this paper, we study the power of shallow quantum circuits. Long quantum computations encounter various problems with decoherence, hence we want to speed them up as much as possible. We can exploit the following two types of parallelism:

1. Gates on different qubits can be applied at the same time.

*Supported by Canada's NSERC and the Canadian Institute for Advanced Research (CIAR).

†Work conducted in part while at Vrije Universiteit, Amsterdam. Partially supported by EU fifth framework project QAIP, IST-1999-11234 and RESQ, IST-2001-37559.

DOI: 10.4086/toc.2005.v001a005

2. *Commuting* gates can be applied to the *same* qubits at the same time.

The first approach is just the classical parallel computation. The second approach only makes sense when the gates applied on the same qubits commute, i.e. $AB = BA$, otherwise the outcome would be ambiguous. Being able to do this is a strong assumption, however there are models of quantum computers, in which it is physically feasible: ion-trap computers [4] and bulk-spin resonance (NMR) [9]. The basic idea is that if two quantum gates commute, so do their Hamiltonians and therefore we can apply their joint operation by performing both evolutions at the same time. This type of research started after the Mølmer–Sørensen paper [15]. Recently, a Hamiltonian implementing the fan-out gate (which is crucial for all our simulations) has been proposed by Fenner [8].

In our paper, we investigate how much the power of quantum computation would increase if we allow such commuting gates. The computation in the stronger model must be efficient, therefore we do not require the ability to perform *any* set of commuting gates. This is in accordance with standard quantum computation, where we also allow only some gates. We choose a representative, the so-called *unbounded fan-out gate*, which is a sequence of controlled-not gates sharing one control qubit. We call it fan-out, because if all target qubits are zero, then the gate copies the *classical* source bit into n copies. We show that fan-out is in some sense universal for all sets of commuting gates. In particular, the joint operation of any set of commuting gates (that can be easily diagonalised) can be simulated by a constant-depth quantum circuit using just one-qubit and fan-out gates. To achieve this, we generalise the parallelisation method of [17, 10] and adapt it to the constant-depth setting.

We state our results in terms of circuit complexity classes. Classically, the main classes computed by constant-depth, polynomial-size circuits are:

NC^0 with Not and bounded fan-in gates: And, Or,
AC^0 with Not and unbounded fan-in gates: And, Or,
TC^0 with Not and unbounded fan-in gates: And, Or, threshold[t] for all t,
$\mathrm{AC}^0[q]$ with Not and unbounded fan-in gates: And, Or, mod[q],
$\mathrm{ACC}^0 = \bigcup_q \mathrm{AC}^0[q]$.

The zero in the exponent means constant depth, in general NC^k means $(\log^k n)$-depth circuits. Several separations between these classes are known. Razborov [18] proved that TC^0 is strictly more powerful than ACC^0. Using algebraic methods, Smolensky [21] proved that $\mathrm{AC}^0[q] \neq \mathrm{AC}^0[q']$, where q, q' are powers of distinct primes. In other words, threshold gates cannot be simulated by constant-depth circuits with unbounded fan-in Or gates, and mod[q] gates do not simulate each other.

The main quantum circuit classes corresponding to the classical classes are QNC^0, QAC^0, QTC^0, and QACC^0. We use subscript 'f' to indicate circuits where we allow the fan-out gate (e.g. $\mathrm{QNC}^0_\mathrm{f}$). Classically, fan-out (copying the result of one gate into inputs of other gates) is taken for granted. Surprisingly, in contrast to the classical case, some of the quantum circuit classes are the same. Moore [16] proved that parity is equivalent to fan-out, i.e. $\mathrm{QAC}^0_\mathrm{f} = \mathrm{QAC}^0[2]$. Green et al. [10] proved that allowing mod[q] gates with different moduli always leads to the same quantum classes, i.e. $\mathrm{QACC}^0 = \mathrm{QAC}^0[q]$ for every integer $q \geq 2$.

In this paper, we extend these results and show that even exact[t] gates (which output 1 if the input is of Hamming weight t, and 0 otherwise) can be approximated with polynomially small error by fan-out and single qubit gates in constant depth. Our simulations have polynomially small error. Since exact[t] gates can simulate And, Or, threshold[t], and mod[q] gates, we conclude that the bounded-error versions of the following classes are equal: $\mathrm{B\text{-}QNC}^0_{\mathrm{f}} = \mathrm{B\text{-}QAC}^0_{\mathrm{f}} = \mathrm{B\text{-}QTC}^0_{\mathrm{f}}$. The exact[t] gate can be approximated in constant depth thanks to the parallelisation method. However, the simulation is not so straightforward as for mod[q] in [10] and it works only with high probability.

We then introduce a so-called Or-reduction that converts n input bits x into $\log n$ output bits y and preserves the Or function, i.e. x is nonzero if and only y is. We show how to implement it exactly in constant depth and use it to achieve exact computation of Or and exact[t] in log-star depth. (Circuits of log-star depth are defined in Section 5.) We also apply the Or-reduction to decrease the size of most of our circuits.

Our results concerning the threshold[t] gate have several interesting implications. Siu et al. [20] proved that sorting and integer arithmetic (addition and multiplication of n integers, and division with remainder) are computable by constant-depth threshold circuits. It follows that all of them can be approximated in $\mathrm{B\text{-}QNC}^0_{\mathrm{f}}$.

The last contribution of our paper concerns the quantum Fourier Transform (QFT). Cleve and Watrous [5] published an elegant log-depth quantum circuit that approximates the QFT. By optimising their methods to use the fan-out gate, we can approximate the QFT in constant depth with polynomially small error. First, we develop a circuit for the QFT with respect to a power-of-2 modulus, and then, using a technique of [11], we show that the QFT with respect to arbitrary moduli can be approximated too. Hence the QFT is in $\mathrm{B\text{-}QNC}^0_{\mathrm{f}}$. The QFT has many applications, one of which is the phase estimation of an unknown quantum state.

Shor's original algorithm for factoring [19] uses the QFT and the modular exponentiation. Cleve and Watrous [5] have shown that it can be adapted to use modular multiplication of n integers. Since we prove that both the QFT and arithmetic operations are in $\mathrm{B\text{-}QNC}^0_{\mathrm{f}}$, polynomial-time bounded-error algorithms with oracle $\mathrm{B\text{-}QNC}^0_{\mathrm{f}}$ can factorise numbers and compute discrete logarithms. We can make the following conclusions: First, if $\mathrm{B\text{-}QNC}^0_{\mathrm{f}}$ can be simulated by a BPP machine, then factoring can be done in polynomial time by bounded-error Turing machines. Second, since it unlikely that $\mathrm{BQP} = \mathrm{B\text{-}QNC}^0_{\mathrm{f}}$, factoring and discrete logarithms are likely not the hardest things quantum computers can do.

2 Quantum circuits with unbounded fan-out

Quantum circuits resemble classical reversible circuits. A quantum circuit is a sequence of quantum gates ordered into *layers*. The gates are consecutively applied in accordance with the order of the layers. Gates in one layer can be applied in parallel. The size of a gate is the number of affected qubits. The *depth* of a circuit is the number of layers and the *size* is the total size of all its gates. A circuit can solve problems of a fixed input size, so we define *families* of circuits containing one circuit for every input size. We consider only *uniform* families, whose description can be generated by a log-space Turing machine.

A *quantum gate* is a unitary operator applied to some subset of qubits. We usually use gates from a fixed *universal basis* (Hadamard gate, rotation by an irrational multiple of π, and the controlled-not gate) that can approximate any quantum gate with good precision [1]. The qubits are divided into 2 groups: *Input/output* qubits contain the description of the input at the beginning and they are measured in the computational basis at the end. *Ancilla qubits* are initialised to $|0\rangle$ at the beginning and the circuits

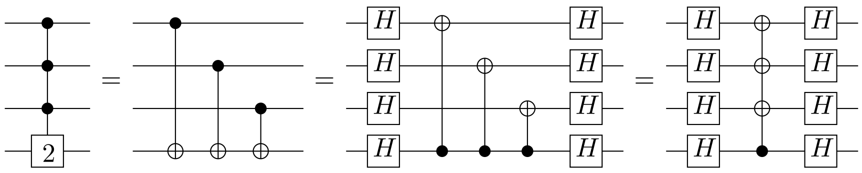

Figure 1: Equivalence of parity and fan-out

usually clean them at the end, so that the output qubits are in a pure state and the ancillas may be reused.

Since unitary evolution is reversible, every operation can be undone. Running the computation backward is called *uncomputation* and is often used for cleaning ancilla qubits.

2.1 Definition of quantum gates

Quantum circuits cannot use a naive quantum fan-out gate mapping every quantum superposition

$$|\phi\rangle|0\rangle\ldots|0\rangle \to |\phi\rangle\ldots|\phi\rangle$$

due to the no-cloning theorem [23]. Such a gate is not linear, let alone unitary. Instead, our fan-out gate copies only classical bits and the effect on superpositions is determined by linearity. It acts as a controlled-not-. . . -not gate, i.e. it is an unbounded sequence of controlled-not gates sharing one control qubit. Parity is a natural counterpart of fan-out. It is an unbounded sequence of controlled-not gates sharing one target qubit.

Definition 2.1. The fan-out gate maps $|y_1\rangle\ldots|y_n\rangle|x\rangle \to |y_1\oplus x\rangle\ldots|y_n\oplus x\rangle|x\rangle$, where $x\oplus y=(x+y)$ mod 2. The parity gate maps $|x_1\rangle\ldots|x_n\rangle|y\rangle \to |x_1\rangle\ldots|x_n\rangle|y\oplus(x_1\oplus\cdots\oplus x_n)\rangle$.

Example 2.2. As used in [16], parity and fan-out can simulate each other in constant depth. The Hadamard gate is $H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$ and it holds that $H^2=I$. If a controlled-not gate is preceded and succeeded by Hadamard gates on both qubits, it just turns around. Since parity is a sequence of controlled-not gates, we can turn around all of them in parallel. The circuit is shown in Figure 1.

In this paper, we investigate the circuit complexity of, among others, these gates:

Definition 2.3. Let $x=x_1\ldots x_n$ and let $|x|$ denote the Hamming weight of x. The following $(n+1)$-qubit gates map $|x\rangle|y\rangle \to |x\rangle|y\oplus g(x)\rangle$, where $g(x)=1$ iff

$$\begin{array}{rlrlrl} |x|>0: & \text{Or}, & |x|=n: & \text{And (Toffoli)}, & |x|\geq\frac{n}{2}: & \text{majority},\\ |x| \mod q=0: & \text{mod[q]}, & |x|\geq t: & \text{threshold[t]}, & |x|=t: & \text{exact[t]}, \end{array}$$

A counting gate is any gate that maps $|x\rangle|0^m\rangle \to |x\rangle|\,|x|\,\rangle$ for $m=\lceil\log(n+1)\rceil$.

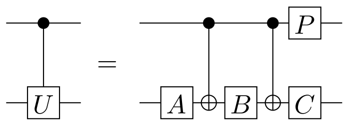

Figure 2: Implementing an arbitrary controlled one qubit gate

2.2 Quantum circuit classes

Definition 2.4. $\mathrm{QNC_f}(d(n))$ contains operators computed exactly (i.e. without error) by uniform families of quantum circuits with fan-out of depth $\mathrm{O}(d(n))$, polynomial size, and over a fixed basis. $\mathrm{QNC_f^k} = \mathrm{QNC_f}(\log^k n)$. R-$\mathrm{QNC_f^k}$ contains operators approximated with one-sided, and B-$\mathrm{QNC_f^k}$ with two-sided, polynomially small error.

Remark 2.5. The circuits below are over a fixed universal basis, unless explicitly mentioned otherwise. Some of our circuits need arbitrary one-qubit gates to be exact. For simplicity, we sometimes include several fixed-size gates (e.g. the binary Or gate and controlled one-qubit gates) in our set of basis gates. This inclusion does not influence the asymptotic depth of our circuits, since every s-qubit quantum gate can be decomposed into a sequence of one-qubit and controlled-not gates of length $\mathrm{O}(s^3 4^s)$ [2].

For every one-qubit gate U, there exist one-qubit gates A, B, C and a rotation $P = R_z(\alpha)$ such that the controlled gate U is computed by the constant-depth circuit shown in Figure 2 [2, Lemma 5.1]. If a qubit controls more one-qubit gates, then we can still use this method in constant depth. We just replace the controlled-not gate by the fan-out gate and the rotations P are multiplied.

3 Parallelisation method

In this section, we describe a general parallelisation method for achieving very shallow circuits. We then apply it to the rotation by Hamming weight and the rotation by value, and show how to compute them in constant depth.

3.1 General method

The unbounded fan-out gate is universal for commuting gates in the following sense: Using fan-out, gates can be applied to the same qubits at the same time whenever (1) they commute, (2) we know the basis in which they all are diagonal, and (3) we can efficiently change into the basis. The method reduces the depth, but may in general require the use of ancilla qubits.

Lemma 3.1. [13, Theorem 1.3.19] *For every set of pairwise commuting unitary gates, there exists an orthogonal basis in which all the gates are diagonal.*

Theorem 3.2. [17, 10] *Let $\{U_i\}_{i=1}^n$ be pairwise commuting gates on k qubits. Gate U_i is controlled by qubit $|x_i\rangle$. Let T be a gate changing the basis according to Lemma 3.1. There exists a quantum circuit with fan-out computing $U = \prod_{i=1}^n U_i^{x_i}$ having depth $\max_{i=1}^n \mathrm{depth}(U_i) + 4 \cdot \mathrm{depth}(T) + 2$, size $\sum_{i=1}^n \mathrm{size}(U_i) + (2n+2) \cdot \mathrm{size}(T) + 2n$, and using $(n-1)k$ ancillas.*

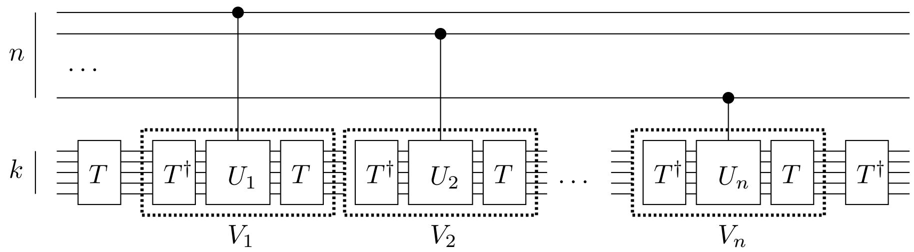

Figure 3: A serial circuit with interpolated basis changes

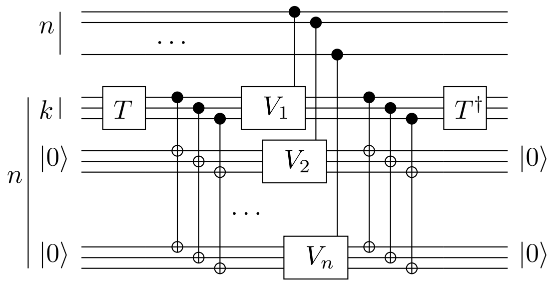

Figure 4: A parallelised circuit performing $U = T^\dagger(\prod_{i=1}^n V_i^{x_i})T = \prod_{i=1}^n U_i^{x_i}$

Proof. Consider a circuit that applies all U_i sequentially. Put $TT^\dagger = I$ between U_i and U_{i+1}. The circuit is shown in Figure 3. Take $V_i = T^\dagger U_i T$ as new gates. They are diagonal in the computational basis, hence they just impose some phase shifts. Multiple phase shifts on entangled states multiply, so can be applied in parallel. We use fan-out gates twice: first to create n entangled copies of target qubits and then to destroy the entanglement. The final circuit with the desired parameters is shown in Figure 4. ☐

Example 3.3. As used in [16], it is simple to prove that mod[q] $\in \mathrm{QNC}_\mathrm{f}^0$. Each input qubit controls one increment modulo q on a counter initialised to 0. At the end, we obtain $|x| \mod q$. The modular increments commute and thus can be parallelised. Since q is fixed, changing the basis and the increment can both be done in constant depth.

3.2 Rotation by Hamming weight and value

In this paper, we often use a *rotation by Hamming weight* $R_z(\varphi|x|)$ and a *rotation by value* $R_z(\varphi x)$, where $R_z(\alpha)$ is one-qubit rotation around the z-axis by angle α: $R_z(\alpha) = |0\rangle\langle 0| + e^{i\alpha}|1\rangle\langle 1|$. They can both be computed in constant depth.

Lemma 3.4. *For every angle φ, there exist constant-depth, linear-size quantum circuits with fan-out computing $R_z(\varphi|x|)$ and $R_z(\varphi x)$ on input $x = x_{n-1}\dots x_1 x_0$.*

Proof. The left circuit in Figure 5 shows how to compute the rotation by Hamming weight. Each input qubit controls $R_z(\varphi)$ on the target qubit, hence the total angle is $\varphi|x|$. These controlled rotations are

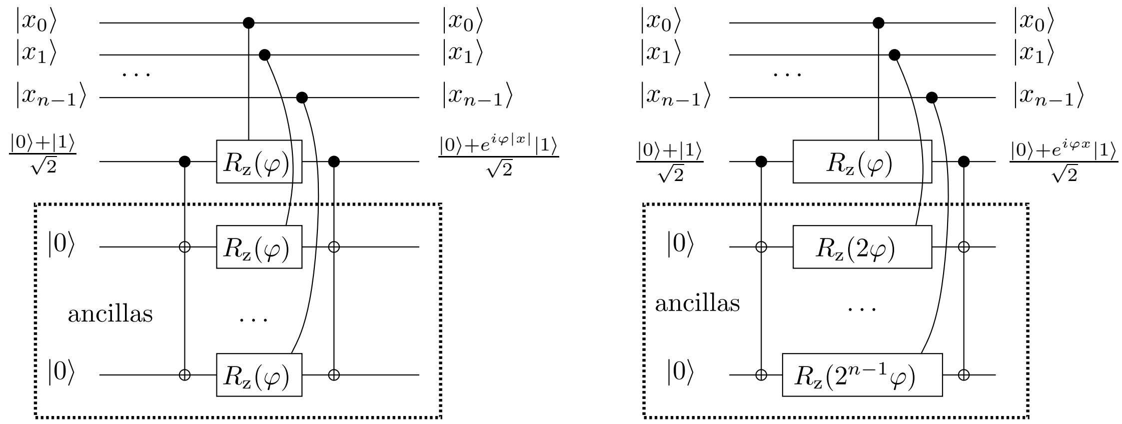

Figure 5: Rotation by Hamming weight and value

parallelised using the parallelisation method. The right circuit shows the rotation by value. It is similar to the rotation by Hamming weight, only the input qubit $|x_j\rangle$ controls $R_z\left(\varphi 2^j\right)$, hence the total angle is $\varphi \sum_{j=0}^{n-1} 2^j x_j = \varphi x$. □

Remark 3.5. The construction uses rotations $R_z(\varphi)$ for arbitrary $\varphi \in \mathbb{R}$. However, we are only allowed to use a fixed set of one-qubit gates. It is easy to see that every rotation can be approximated with polynomially small error by $R_z(\theta q) = (R_z(\theta))^q$, where $\sin\theta = \frac{3}{5}$ and q is a polynomially large integer [1]. These q rotations commute, so can be applied in parallel and the depth is preserved. The approximation can be kept down to polynomially small error while increasing the size of the circuit only polynomially.

4 Constant-depth approximate circuits

4.1 Or gate

It is easy to see that the rotation by Hamming weight of a string y of length m with angle $\varphi = \frac{2\pi}{m}$ can be used to distinguish the zero string $y = 0^m$ from strings with approximately $\frac{m}{2}$ ones. We, however, want to distinguish the zero string from *all* nonzero strings. It turns out that if we compute $m = \mathrm{O}(n \log n)$ rotations by Hamming weight of the input x with angles distributed evenly around the circle, we obtain a string y that is either zero (for $x = 0^n$), or has expected Hamming weight $\frac{m}{2}$ (for $x \neq 0^n$). By combining these two results, we can approximate the Or gate and, with a minor modification, also the exact[t] gate in constant depth.

Let $w \in \mathbb{N}_0$ and let φ be an angle. Define a notation for the following one-qubit state:

$$|\mu_\varphi^w\rangle = (H \cdot R_z(\varphi w) \cdot H)\,|0\rangle = \frac{1+e^{i\varphi w}}{2}|0\rangle + \frac{1-e^{i\varphi w}}{2}|1\rangle. \tag{4.1}$$

By Lemma 3.4, $|\mu_\varphi^{|x|}\rangle$ can be computed in constant depth and linear size.

Theorem 4.1. *Or* $\in$ R-QNC$^0_\mathrm{f}$. *In particular, Or can be approximated with one-sided error* $\frac{1}{n}$ *in constant depth and size* $\mathrm{O}(n^2 \log n)$.

Proof. Let n denote the size of the input x. Let $m = a \cdot n$, where a will be chosen later. For all $k \in \{0, 1, \ldots, m-1\}$, compute in parallel $|y_k\rangle = |\mu_{\varphi_k}^{|x|}\rangle$ for angle $\varphi_k = \frac{2\pi}{m} k$. If $|y_k\rangle$ is measured in the computational basis, the expected value of the outcome $Y_k \in \{0,1\}$ is

$$E[Y_k] = \left| \frac{1 - e^{i\varphi_k |x|}}{2} \right|^2 = \left| e^{-i\varphi_k |x|} \right| \cdot \frac{\left| e^{i\varphi_k |x|} + e^{-i\varphi_k |x|} - 2 \right|}{4} = \frac{1 - \cos(\varphi_k |x|)}{2} \quad .$$

If all these m qubits $|y\rangle$ are measured, the expected Hamming weight of all Y's is

$$E[|Y|] = E\left[\sum_{k=0}^{m-1} Y_k \right] = \frac{m}{2} - \frac{1}{2} \sum_{k=0}^{m-1} \cos\left(\frac{2\pi k}{m} |x| \right) = \begin{cases} 0 & \text{if } |x| = 0, \\ \frac{m}{2} & \text{if } |x| \neq 0. \end{cases}$$

The qubits $|y\rangle$ are actually not measured, but their Hamming weight $|y|$ controls another rotation on a new ancilla qubit $|z\rangle$. So compute $|z\rangle = |\mu_{2\pi/m}^{|y|}\rangle$. Let Z be the outcome after $|z\rangle$ is measured. If $|y| = 0$, then $Z = 0$ with certainty. If $\left| |y| - \frac{m}{2} \right| \leq \frac{m}{\sqrt{n}}$, then

$$P[Z = 0] = \left| \frac{1 + e^{i \frac{2\pi}{m} |y|}}{2} \right|^2 = \frac{1 + \cos\left(\frac{2\pi}{m} |y| \right)}{2} \leq \frac{1 - \cos \frac{2\pi}{\sqrt{n}}}{2} = \mathrm{O}\left(\frac{1}{n} \right) \quad .$$

Assume that $|x| \neq 0$. We want to upper-bound the probability of the bad event that $|Y|$ is not close to $\frac{m}{2}$. Since $0 \leq Y_k \leq 1$, we can use Hoeffding's Lemma 4.2 below and obtain $P\left[\left| |Y| - \frac{m}{2} \right| \geq \varepsilon m \right] \leq \frac{1}{2^{\varepsilon^2 m}}$. Fix $a = \log n$ and $\varepsilon = \frac{1}{\sqrt{n}}$. Now, $P\left[\left| |y| - \frac{m}{2} \right| \geq \frac{m}{\sqrt{n}} \right] \leq \frac{1}{2^{m/n}} = \frac{1}{2^a} = \frac{1}{n}$. The probability that we observe the incorrect result $Z = 0$ is at most the sum of the probabilities of the two bad events, i.e. $\mathrm{O}\left(\frac{1}{n}\right)$. Hence

$$P[Z = 0] = \begin{cases} 1 & \text{if } |x| = 0, \\ \mathrm{O}\left(\frac{1}{n}\right) & \text{if } |x| \neq 0. \end{cases}$$

The circuit has constant depth and size $\mathrm{O}(mn) = \mathrm{O}(n^2 \log n)$. It is outlined in Figure 6. The figure is slightly simplified: unimportant qubits and uncomputation of ancillas are omitted. $\square$

Lemma 4.2 (Hoeffding [12]). *If* $Y_1, \ldots, Y_m$ *are independent random variables bounded by* $a_k \leq Y_k \leq b_k$, *then, for all* $\varepsilon > 0$,

$$P\left[|S - E[S]| \geq \varepsilon m \right] \leq 2 \exp \frac{-2m^2 \varepsilon^2}{\sum_{k=1}^m (b_k - a_k)^2} \quad , \quad \textit{where } S = \textstyle\sum_{i=k}^m Y_k.$$

Remark 4.3. Since the outcome z of the circuit in Figure 6 is a classical bit, we can save it in an ancilla qubit by applying a controlled-not gate and clean $|y\rangle$ by uncomputation. It remains to prove that the intermediate qubits $|y\rangle$ need not be measured, in order to be able to uncompute them. We show above that the output qubit is a good approximation of the logical Or, provided $|y\rangle$ is immediately measured.

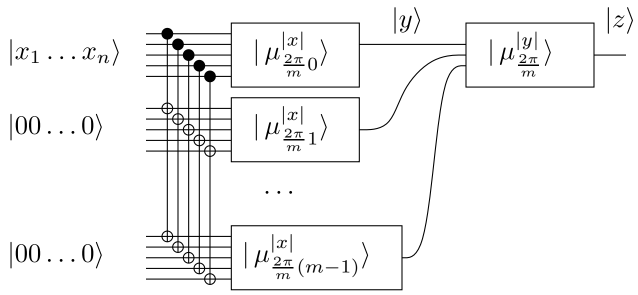

Figure 6: Constant depth circuit approximating Or

By the principle of deferred measurement, we can use controlled quantum operations and measure $|y\rangle$ at the end. However, the output bit is close to a classical bit (the distance depends on the error of the computation), thus it is only slightly entangled with $|y\rangle$, and hence it does not matter whether $|y\rangle$ is measured.

Definition 4.4. Let $\log^{(k)} x$ denote the k-times *iterated logarithm* $\log\log\dots\log x$. The *log-star function*, $\log^* x$, is the maximum number of iterations k such that $\log^{(k)} x$ exists and is real.[1]

Remark 4.5. If we require error $\frac{1}{n^c}$, we create c copies and compute the exact Or of them by a binary tree of Or gates. The tree has depth $\log c = \mathrm{O}(1)$. In Section 6.1, we show how to approximate Or in constant depth and size $\mathrm{O}(n \log^{(k)} n)$ for any constant k. In Section 6.2, we show how to compute Or *exactly* in log-star depth and linear size.

4.2 Exact[t] and threshold[t] gates

Theorem 4.6. *exact[t]* $\in$ R-QNC$_\mathrm{f}^0$.

Proof. We slightly modify the circuit for Or. As outlined in Figure 7, by adding the rotation $R_z(-\varphi t)$ to the rotation by Hamming weight in the first layer, we obtain $|\mu_\varphi^{|x|-t}\rangle$ instead of $|\mu_\varphi^{|x|}\rangle$. The second layer stays the same. If the output qubit $|z\rangle$ is measured, then

$$P[Z=0] = \begin{cases} 1 & \text{if } |x| = t, \\ \mathrm{O}\left(\frac{1}{n}\right) & \text{if } |x| \neq t. \end{cases}$$

We obtain an approximation of the exact[t] gate with one-sided polynomially small error. ☐

Remark 4.7. Other gates are computed from the exact[t] gate by standard methods. For example, threshold[t] can be computed as the parity of exact[t], exact[$t+1$], ..., exact[n]. The depth stays constant and the size is just n-times bigger, i.e. $\mathrm{O}\big(n^3 \log n\big)$, hence threshold[t] $\in$ B-QNC$_\mathrm{f}^0$. In Section 6.3, we show how to approximate exact[t], threshold[t], and counting in constant depth and size $\mathrm{O}(n \log n)$.

[1] The log-star of the estimated number of atoms in the universe is 5. Consequently, for the computational problems we consider in this paper, the log-star is in practice at most 5.

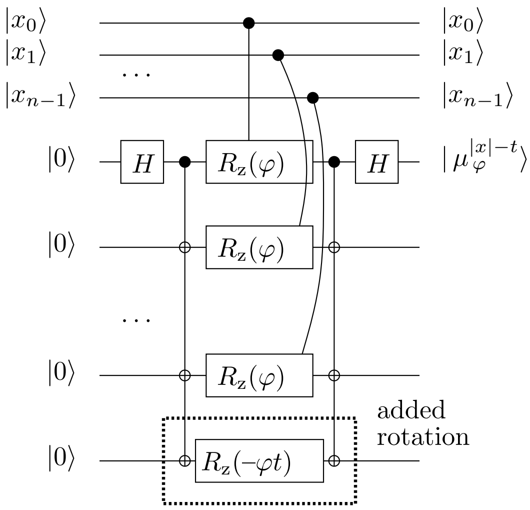

Figure 7: Rotation by Hamming weight with added rotation

4.3 Arithmetic operations

Using threshold gates, one can do arithmetic operations in constant depth. The following circuits take as part of the input an ancilla register in state $|0\rangle$ and output the result of the computation in that register.

Theorem 4.8. *The following functions are in* B-QNC$^0_\mathrm{f}$*: addition and multiplication of n integers, division of integers with remainder, and sorting of n integers.*

Proof. By [20], these functions are computed by constant-depth,[2] polynomial-size threshold circuits. A threshold circuit is built of weighted threshold gates. It is simple to prove that the weighted threshold gate (with polynomially large integer weights) also is in B-QNC$^0_\mathrm{f}$. One only needs to rotate the phase of the quantum state in Lemma 3.4 by integer multiples of the basic angle. □

In the following section, we require a reversible version of modular addition.

Definition 4.9. Let q be an n-bit integer and $x_1,\dots,x_m \in \mathbb{Z}_q$. The *reversible addition* gate maps add^m : $|q\rangle|x_1\rangle\dots|x_m\rangle \to |q\rangle|x_1\rangle\dots|x_{m-1}\rangle|y\rangle$, where $y = (\sum_{i=1}^m x_i) \mod q$.

Lemma 4.10. $\mathrm{add}^m \in$ B-QNC$^0_\mathrm{f}$.

Proof. By Theorem 4.8, $y = (\sum_{i=1}^m x_i) \mod q$ can be approximated in constant depth and polynomial size. The result is, however, stored into ancilla qubits. Hence we have to erase x_m, which we may achieve by first negating the contents in y by $|y\rangle \to |-y\rangle$, computing the sum $w = y + \sum_{i=1}^{m-1} x_i$ in a fresh ancilla, do a bitwise control-not of w into x_m, uncompute w, and finally re-negate y. We then swap the ancillas $|y\rangle$ with the erased qubits in $|x_m\rangle$. □

[2] The depths are really small, from 2 to 5.

4.4 Quantum Fourier transform

The QFT is a very powerful tool used in several quantum algorithms, e.g. factoring of integers and computing the discrete logarithm [19].

Definition 4.11. The quantum Fourier transform with respect to modulus q performs the Fourier transform on the quantum amplitudes of the state, i.e. it maps

$$F_q : |x\rangle \rightarrow |\psi_x\rangle = \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \omega^{xy} |y\rangle, \text{ where } \omega = e^{2\pi i/q}, \tag{4.2}$$

for $x \in \{0, 1, \ldots, q-1\}$ and it behaves arbitrarily on the other states.

4.4.1 QFT with a power-of-2 modulus

Let $q = 2^n$. Coppersmith has shown in [6] how to compute the QFT in quadratic depth, quadratic size, and without ancillas. The depth has further been improved to linear [folklore]. Cleve and Watrous have shown in [5] that the QFT can be approximated with error ε in depth $\mathrm{O}\big(\log n + \log\log \frac{1}{\varepsilon}\big)$ and size $\mathrm{O}\big(n \log \frac{n}{\varepsilon}\big)$. They also show that if only gates acting on a constant number of qubits are allowed (in particular, the fan-out gate is not allowed), logarithmic depth is necessary. We show that the approximate circuit for the QFT from [5] can be compressed to constant depth, if we allow the fan-out gate.

Theorem 4.12. *QFT* $\in$ B-QNC$^0_\mathrm{f}$.

Proof. The operator $F_{2^n} : |x\rangle \rightarrow |\psi_x\rangle$ can be computed by composing:

1. Fourier state construction (QFS): $|x\rangle|0\rangle \ldots |0\rangle \rightarrow |x\rangle|\psi_x\rangle|0\rangle \ldots |0\rangle$
2. Copying Fourier state (COPY): $|x\rangle|\psi_x\rangle|0\rangle \ldots |0\rangle \rightarrow |x\rangle|\psi_x\rangle \ldots |\psi_x\rangle$
3. Uncomputing phase estimation (QFP): $|\psi_x\rangle \ldots |\psi_x\rangle|x\rangle \rightarrow |\psi_x\rangle \ldots |\psi_x\rangle|0\rangle$
4. Uncomputing COPY: $|\psi_x\rangle \ldots |\psi_x\rangle|0\rangle \rightarrow |\psi_x\rangle|0\rangle \ldots |0\rangle$

The following lemmas show that each of these individual operators is in B-QNC$^0_\mathrm{f}$. □

Lemma 4.13. *QFS* $\in$ QNC$^0_\mathrm{f}$.

Proof. QFS maps $|x\rangle|0\rangle \rightarrow |x\rangle|\psi_x\rangle$. Define a shortcut $|\rho_r\rangle = \frac{|0\rangle + e^{2\pi i r}|1\rangle}{\sqrt{2}}$. It is simple to prove that $|\psi_x\rangle = |\rho_{x/2^1}\rangle|\rho_{x/2^2}\rangle \ldots |\rho_{x/2^n}\rangle$.

$$\begin{aligned} |\psi_x\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \omega^{xy}|y\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} \bigotimes_{k=1}^{n} \omega^{x2^{n-k}y_{n-k}} |y_{n-k}\rangle \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{k=1}^{n} \sum_{b=0}^{1} (\omega^{2^{n-k}x})^b |b\rangle = \bigotimes_{k=1}^{n} \frac{|0\rangle + e^{2\pi i x/2^k}|1\rangle}{\sqrt{2}} = \bigotimes_{k=1}^{n} |\rho_{x/2^k}\rangle. \end{aligned}$$

The n qubits $|\rho_{x/2^k}\rangle$ can be computed from x in parallel as follows: $|\rho_{x/2^k}\rangle = R_z\left(\frac{2\pi}{2^k}x\right)\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ is computed by the rotation by value (Lemma 3.4) in constant depth and linear size. □

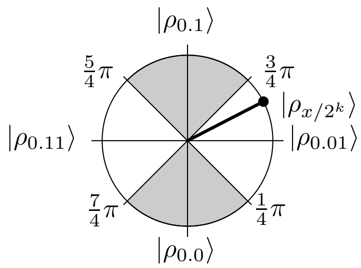

Figure 8: Measurement of $|\rho_{x/2^k}\rangle$ in a random basis

Lemma 4.14. *COPY* $\in$ B-QNC$^0_\mathrm{f}$.

Proof. COPY maps $|\psi_x\rangle|0\rangle\ldots|0\rangle \to |\psi_x\rangle\ldots|\psi_x\rangle$. Take the reversible addition gate modulo 2^n mapping $(\mathrm{add}^2_{2^n})|y\rangle|x\rangle = |y\rangle|(x+y) \mod 2^n\rangle$. It is simple to prove that $\mathrm{add}^{-1}|\psi_y\rangle|\psi_x\rangle = |\psi_{x+y}\rangle|\psi_x\rangle$.

$$\begin{aligned} |\psi_y\rangle|\psi_x\rangle &= \frac{1}{2^n}\sum_{l,k=0}^{2^n-1}\omega^{ly+kx}|l\rangle|k\rangle \to_{\mathrm{add}^{-1}} \frac{1}{2^n}\sum_{l,k}\omega^{ly+kx}|l\rangle|k-l\rangle \\ &= \frac{1}{2^n}\sum_{l,m}\omega^{ly+(m+l)x}|l\rangle|m\rangle = \frac{1}{2^n}\sum_{l,m}\omega^{l(x+y)+mx}|l\rangle|m\rangle = |\psi_{x+y}\rangle|\psi_x\rangle. \end{aligned}$$

Hence $\mathrm{add}^{-1}|\psi_0\rangle|\psi_x\rangle = |\psi_x\rangle|\psi_x\rangle$. The state $|\psi_0\rangle = H^{\otimes n}|0^n\rangle$ is easy to prepare in constant depth. Furthermore, $(\mathrm{add}^m_{2^n})^{-1}|\psi_0\rangle\ldots|\psi_0\rangle|\psi_x\rangle = |\psi_x\rangle\ldots|\psi_x\rangle|\psi_x\rangle$, because the addition of $m-1$ numbers into one register is equivalent to $m-1$ consecutive additions of one number. Each such a reversible addition copies $|\psi_x\rangle$ into 1 register. Note that the $\mathrm{add}^m_{2^n}$ gate performs all these additions in parallel. By Lemma 4.10, the reversible addition gate is in B-QNC$^0_\mathrm{f}$. □

Lemma 4.15. *QFP* $\in$ B-QNC$^0_\mathrm{f}$.

Proof. QFP maps $|\psi_x\rangle\ldots|\psi_x\rangle|0\rangle \to |\psi_x\rangle\ldots|\psi_x\rangle|x\rangle$. By Cleve and Watrous [5, Section 3.3], we can compute x with probability at least $1-\varepsilon$ from $\mathrm{O}\big(\log\frac{n}{\varepsilon}\big)$ copies of $|\psi_x\rangle$ in depth $\mathrm{O}\big(\log n + \log\log\frac{1}{\varepsilon}\big)$ and size $\mathrm{O}\big(n\log\frac{n}{\varepsilon}\big)$. Use $\varepsilon = \frac{1}{\mathrm{poly}(n)}$. It is simple to convert their circuit into constant depth, provided we have fan-out. The details are sketched below.

The input consists of $m = \mathrm{O}\big(\log\frac{n}{\varepsilon}\big)$ copies of $|\psi_x\rangle = |\rho_{x/2^1}\rangle|\rho_{x/2^2}\rangle\ldots|\rho_{x/2^n}\rangle$. Measure each $|\rho_{x/2^k}\rangle$ $\frac{m}{2}$ times in the basis $\{|\rho_{0.01}\rangle,|\rho_{0.11}\rangle\}$ and $\frac{m}{2}$ times in the Hadamard basis $\{|\rho_{0.00}\rangle,|\rho_{0.10}\rangle\}$. The state $|\rho_{x/2^k}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_{k-1}\ldots x_1x_0)})$ lies on the middle circle of the Bloch sphere; it is shown in Figure 8. If $|\rho_{x/2^k}\rangle$ is in the white region, then the measurement in the first basis tells whether $x_{k-1}=0$ or 1 with probability at least $\frac{3}{4}$. If $|\rho_{x/2^k}\rangle$ is in the shaded region, then the measurement in the Hadamard basis tells whether $x_{k-1}=x_{k-2}$ or $\neg x_{k-2}$ (denoted by P, N) with probability at least $\frac{3}{4}$.

For each k, perform the majority vote and obtain the correct answer $z_k \in \{0,1,\mathrm{P},\mathrm{N}\}$ with error probability at most $\frac{1}{2^m} = \frac{\varepsilon}{n}$. The probability of having any error is at most n times bigger, i.e. at most ε. Compute $x_{n-1}\ldots x_1x_0$ from $z_{n-1}\ldots z_1z_0$ in constant depth. The bit x_k is computed as follows:

1. If $z_k z_{k-1} \dots z_{l+1} \in \{\mathrm{P},\mathrm{N}\}$ and $z_l \in \{0,1\}$, compute the parity of the number of N's and add it to z_l (assuming $z_{-1} = 0$), otherwise return 0.

2. Check and compute all prefixes l in parallel and take the logical Or of the results.

All the gates used (fan-out, parity, And, Or, majority) are in B-QNC_f^0. □

4.4.2 QFT with an arbitrary modulus

Let $q \neq 2^n$. Cleve and Watrous have shown in [5] that the QFT can be approximated with error ε in depth $\mathrm{O}\big((\log\log q)(\log\log\frac{1}{\varepsilon})\big)$ and size $\mathrm{poly}(\log q + \log\frac{1}{\varepsilon})$. We show that their circuit can also be compressed into constant depth, if we use the fan-out gate. The relation between quantum Fourier transforms with different moduli was described in [11].

Remark 4.16. We actually implement a slightly more general operation, when q is not a fixed constant, but an n-bit *input* number. This generalised QFT maps $|q\rangle|x\rangle \to |q\rangle|\psi_x\rangle$. The register $|q\rangle$ is implicitly included in all operations. We will henceforth omit it and the generalised operations are denoted simply by QFT_q, QFS_q, COPY_q^m, and QFP_q.

Theorem 4.17. *$QFT_q \in$ B-QNC$_\text{f}^0$.*

Proof. Let $|\mathrm{dummy}_{q,x}\rangle$ denote an unspecified quantum state depending on two parameters q,x. The operator $F'_q : |x\rangle \to |\psi_x\rangle|\mathrm{dummy}_{q,0}\rangle$ can be computed by composing:

$$\begin{array}{lll}
1.\ \ \mathrm{QFS}_q: & |x\rangle & \to |x\rangle|\psi_x\rangle|\mathrm{dummy}_{q,x}\rangle \\
2.\ \ \mathrm{COPY}_q^{m+1}: & & \to |x\rangle|\psi_x\rangle|\mathrm{dummy}_{q,x}\rangle\big(|\psi_x\rangle|\mathrm{dummy}_{q,0}\rangle\big)^{\otimes m} \\
3.\ \ \text{Uncomputing } \mathrm{QFS}_q: & & \to |x\rangle\big(|\psi_x\rangle|\mathrm{dummy}_{q,0}\rangle\big)^{\otimes m} \\
4.\ \ \text{Uncomputing } \mathrm{QFP}_q: & & \to \big(|\psi_x\rangle|\mathrm{dummy}_{q,0}\rangle\big)^{\otimes m} \\
5.\ \ \text{Uncomputing } \mathrm{COPY}_q^m: & & \to |\psi_x\rangle|\mathrm{dummy}_{q,0}\rangle,
\end{array}$$

where empty registers are omitted for clarity. The state $|\mathrm{dummy}_{q,0}\rangle$ is not entangled with $|x\rangle$ and hence it can be traced out. We obtain the quantum Fourier transform F_q. The following lemmas show that each of these individual operators is in B-QNC_f^0. □

Lemma 4.18. *$QFS_q \in$ B-QNC$_\text{f}^0$.*

Proof. QFS_q maps $|x\rangle|0\rangle \to |x\rangle|\psi_x\rangle|\mathrm{dummy}_{q,x}\rangle$ for some "garbage" state $|\mathrm{dummy}_{q,x}\rangle$. We will show that QFS_q is well approximated by a QFS with a power-of-2 modulus of the magnitude q^3. Let $n = \lceil \log q \rceil$. Take $N = 3n$ and extend x by leading zeroes into N bits. Using Lemma 4.13, perform QFS_{2^N} and obtain the state $|x\rangle\frac{1}{\sqrt{2^N}}\sum_{y=0}^{2^N-1} e^{\frac{2\pi i}{2^N}xy}|y\rangle$.

Set $u = \lfloor 2^N/q \rfloor$ and apply integer division by u to the second register, i.e. map $|y\rangle \to |y_1\rangle|y_2\rangle$, where $y_1 = \lfloor y/u \rfloor \in \{0,1,\dots,q\}$ and $y_2 = y \mod u$. This can be done reversibly in constant depth by a few applications of Theorem 4.8 using the method from Lemma 4.10. The quantum state can be written as

$$\frac{1}{\sqrt{2^N}}\sum_{y=0}^{2^N-1} e^{\frac{2\pi i}{2^N}xy}|y_1\rangle|y_2\rangle = \frac{1}{\sqrt{2^N}}\sum_{y_1=0}^{q-1}\sum_{y_2=0}^{u-1} e^{\frac{2\pi i}{2^N}x(y_1u+y_2)}|y_1\rangle|y_2\rangle + |w\rangle \ ,$$

where $|w\rangle = \frac{1}{\sqrt{2^N}} \sum_{z=0}^{v-1} e^{\frac{2\pi i}{2^N} x(qu+z)} |q\rangle|z\rangle$ and $v = 2^N \mod u = 2^N - qu = 2^N \mod q < q$. The sum has been rearranged using $y = y_1 u + y_2$. Now, $\| \, |w\rangle \, \| = \sqrt{\frac{v}{2^N}} = \mathrm{O}(2^{-n})$ is exponentially small and so it can be neglected. Decompose the quantum state into the tensor product

$$\frac{1}{\sqrt{q}} \sum_{y_1=0}^{q-1} e^{\frac{2\pi i}{q} (\frac{q}{2^N} u) x y_1} |y_1\rangle \otimes \sqrt{\frac{q}{2^N}} \sum_{y_2=0}^{u-1} e^{\frac{2\pi i}{2^N} x y_2} |y_2\rangle \quad .$$

Now, u is exponentially close to $\frac{2^N}{q}$, because $\frac{q}{2^N} u = \frac{2^N - v}{2^N} = 1 - \mathrm{O}(2^{-2n})$. Since $\frac{xy_1}{q} = \mathrm{O}(2^n)$, the replacement of $\frac{qu}{2^N}$ by 1 in the exponent causes only exponentially small error $\mathrm{O}(2^{-n})$. Hence the quantum state is exponentially close to

$$\frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} e^{\frac{2\pi i}{q} xy} |y\rangle \otimes \frac{1}{\sqrt{u}} \sum_{z=0}^{u-1} e^{\frac{2\pi i}{2^N} xz} |z\rangle = |\psi_x\rangle |\mathrm{dummy}_{q,x}\rangle \quad .$$

The "garbage" state $|\mathrm{dummy}_{q,x}\rangle$ arises as a byproduct of the higher precision $3n$-bit arithmetic. We clean it up later by uncomputing QFS_q after copying $|\psi_x\rangle$; see the proof of Theorem 4.17. It actually gets replaced by $|\mathrm{dummy}_{q,0}\rangle = \frac{1}{\sqrt{u}} \sum_{z=0}^{u-1} |z\rangle$, which does not depend on x and it thus causes no harm. We have approximated QFS_q in constant depth. □

Lemma 4.19. *$COPY_q^m$* $\in$ B-QNC$^0_\mathrm{f}$.

Proof. COPY_q^m maps $|\psi_x\rangle|0\rangle \dots |0\rangle \to |\psi_x\rangle (|\psi_x\rangle |\mathrm{dummy}_{q,0}\rangle)^{\otimes(m-1)}$. The proof is similar to the proof of Lemma 4.14. First, prepare $m-1$ states $|\psi_0\rangle|\mathrm{dummy}_{q,0}\rangle$ by applying QFS_q to $|0\rangle|0\rangle$ (Lemma 4.18). Second, use the inverse of the reversible addition modulo q to map $(\mathrm{add}_q^m)^{-1} : |\psi_0\rangle \dots |\psi_0\rangle|\psi_x\rangle \to |\psi_x\rangle \dots |\psi_x\rangle|\psi_x\rangle$ (Lemma 4.10). □

Lemma 4.20. *QFP_q* $\in$ B-QNC$^0_\mathrm{f}$.

Proof. QFP_q maps $|\psi_x\rangle \dots |\psi_x\rangle|0\rangle \to |\psi_x\rangle \dots |\psi_x\rangle|x\rangle$. We use an idea similar to the proof of Lemma 4.18. Let $n = \lceil \log q \rceil$ and $N = 3n$. Extend $|\psi_x\rangle$ by leading zeroes to N bits and apply $F_{2^N}^\dagger$ to them (Theorem 4.12). We obtain many copies of the state

$$F_{2^N}^\dagger (|0\rangle|\psi_x\rangle) = \frac{1}{\sqrt{2^N q}} \sum_{z=0}^{2^N-1} \left(\sum_{y=0}^{q-1} e^{\frac{-2\pi i}{2^N} zy + \frac{2\pi i}{q} xy} \right) |z\rangle \quad .$$

The exponent can be rewritten to $2\pi i(\frac{x}{q} - \frac{z}{2^N}) \cdot y$. Intuitively, if $|z - 2^N \frac{x}{q}| \le \frac{2^N}{8q}$, then $|\frac{x}{q} - \frac{z}{2^N}| \le \frac{1}{8q}$, the absolute value of the angle in the exponent is at most $\frac{\pi}{4}$ for every $y \in \{0, 1, \dots, q-1\}$, and the amplitudes sum up constructively. If z is not close to $2^N \frac{x}{q}$, then the amplitudes interfere destructively. The quantum state has most of its amplitude on the good z's. So we compute reversibly by division with remainder an estimate $x' = \lfloor \frac{zq}{2^N} + \frac{1}{2} \rfloor$. A detailed analyzis shows that $P[x' = x] \ge \frac{1}{2} + \delta$ for some constant δ [5, 11]. Here we do not present the details, because our goal is the compression of the circuit from [5] into constant depth.

We transform all $m = \mathrm{O}\big(\log \frac{n}{\varepsilon}\big)$ input quantum states $|\psi_x\rangle$ into m independent estimates $|x'\rangle$. We then estimate all bits of x one-by-one from these m estimates by majority gates. Each bit of x is wrong with probability at most $2^{-m} = 2^{-\log \frac{n}{\varepsilon}} = \frac{\varepsilon}{n}$. The probability of having an error among the n bits of x is thus at most ε. Finally, save the estimation of x in the target register and uncompute the divisions and the quantum Fourier transforms. With probability at least $1-\varepsilon$, the mapping QFP_q has been performed. Use $\varepsilon = \frac{1}{\mathrm{poly}\, n}$. □

4.5 Quantum phase estimation

The method of computing QFT_{2^n} can be also used for phase estimation.

Theorem 4.21. *Given a gate $S_x : |y\rangle|\phi\rangle \to |y\rangle R_z\left(\frac{2\pi x}{2^n} y\right)|\phi\rangle$ for basis states $|y\rangle$, where $x \in \mathbb{Z}_{2^n}$ is unknown, we can determine x with probability at least $1-\varepsilon$ in constant depth, size $\mathrm{O}\big(n \log \frac{n}{\varepsilon}\big)$, and using the S_x gate $\mathrm{O}\big(n \log \frac{n}{\varepsilon}\big)$ times.*

Proof. Obtain an estimate of x by applying the QFP to $\mathrm{O}\big(\log \frac{n}{\varepsilon}\big)$ copies of the quantum state $|\psi_x\rangle = |\rho_{x/2^1}\rangle|\rho_{x/2^2}\rangle \ldots |\rho_{x/2^n}\rangle$. Each $|\rho_{x/2^k}\rangle$ can be computed by one application of S_x to $|2^{n-k}\rangle \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, because $|\rho_{x/2^k}\rangle = R_z\left(\frac{2\pi x}{2^k}\right)\frac{|0\rangle+|1\rangle}{\sqrt{2}} = R_z\left(\frac{2\pi x}{2^n} 2^{n-k}\right)\frac{|0\rangle+|1\rangle}{\sqrt{2}}$. □

5 Exact circuits of small depth

In the previous section, we have shown how to *approximate* the exact[t] gate in constant depth. In this section, we show how to *compute it exactly* in log-star depth. The circuits in this section use arbitrary one-qubit gates instead of a fixed basis, otherwise they would not be exact.

Lemma 5.1. *The function Or on n qubits can be reduced* exactly *to Or on $m = \lceil \log(n+1) \rceil$ qubits in constant depth and size* $\mathrm{O}(n \log n)$.

Proof. We use a technique similar to the proof of Theorem 4.1. Recall the quantum state $|\mu_\varphi^w\rangle$ defined by Equation (4.1) on page 87. For $k \in \{1, 2, \ldots, m\}$, compute in parallel $|y_k\rangle = |\mu_{\varphi_k}^{|x|}\rangle$ for angle $\varphi_k = \frac{2\pi}{2^k}$. Let $|y\rangle = |y_1 y_2 \ldots y_m\rangle$.

- If $|x| = 0$, then $\langle y|0^m\rangle = 1$, because $|y_k\rangle = |0\rangle$ for each k.
- If $|x| \neq 0$, then $\langle y|0^m\rangle = 0$, because at least one qubit y_k is one with certainty. Take the unique decomposition of $|x|$ into a product of a power of 2 and an odd number: $|x| = 2^a(2b+1)$ for $a, b \in \mathbb{N}_0$. Then

$$\langle 1|y_{a+1}\rangle = \frac{1 - e^{i\varphi_{a+1}|x|}}{2} = \frac{1 - e^{i\frac{2\pi}{2^{a+1}}2^a(2b+1)}}{2} = \frac{1 - e^{i\pi(2b+1)}}{2} = \frac{1 - e^{i\pi}}{2} = 1 \ .$$

It follows that x is non-zero if and only if y is. Hence the original problem is exactly reduced to a problem of logarithmic size. □

Theorem 5.2. *exact[t]* $\in \mathrm{QAC}_{\mathrm{f}}^0$.

Proof. Using the methods from Theorem 4.6 and Lemma 5.1, exact[t] can also be reduced to Or of logarithmic size. The reduction has constant depth and size $\mathrm{O}(n \log n)$. Hence exact[t] is $\mathrm{QNC}_{\mathrm{f}}^0$-reducible to Or, or simply exact[t] $\in \mathrm{QAC}_{\mathrm{f}}^0$, because $\mathrm{QAC}_{\mathrm{f}}^0$ includes both $\mathrm{QNC}_{\mathrm{f}}^0$ and the Or gate. □

Theorem 5.3. *exact[t]* $\in \mathrm{QNC}_{\mathrm{f}}(\log^* n)$*, i.e. exact[t] can be computed* exactly *in log-star depth and size* $\mathrm{O}(n \log n)$.

Proof. Apply the reduction used in Lemma 5.1 in total $(\log^* n)$-times, until the input size is at most 2. Compute and save the outcome, and clean ancillas by uncomputation. The circuit size is $\mathrm{O}(n \log n)$. □

6 Circuits of small size

In this section, we decrease the size of some circuits. We allow the use of arbitrary one-qubit gates instead of a fixed basis.

6.1 Constant depth approximation of Or

In this section, we apply the reduction from Lemma 5.1 repeatedly to shrink the circuit for Or. We first reduce the size of the circuit to $\mathrm{O}(n \log n)$. We then develop a recurrent method that reduces the size even further. Let us define a useful notation.

Definition 6.1. Let $x = x_1 x_2 \dots x_n$. By *Or-reduction* $n \to m$ with error ε we mean a quantum circuit mapping $|x\rangle|0^m\rangle \to |x\rangle|\varphi\rangle$ such that, if $|x| = 0$, then $|\varphi\rangle = |0^m\rangle$ and, if $|x| \neq 0$, then $\langle 0^m|\varphi\rangle \leq \varepsilon$.

The Or-reduction preserves the logical Or of qubits, i.e. $|x| = 0$ iff $|\varphi| = 0$ with high probability. Theorem 4.1 provides an Or-reduction $n \to 1$ with error $\frac{1}{n}$, constant depth, and size $n^2 \log n$. Lemma 5.1 provides an Or-reduction $n \to \log n$ with error 0, constant depth, and size $n \log n$.

Lemma 6.2. *There is an Or-reduction* $n \to 1$ *with error* $\frac{1}{n}$*, constant depth, and size* $n \log n$*.*

Proof. Divide the input into $\frac{\sqrt{n}}{\log n}$ blocks of size $\sqrt{n} \log n$. First, reduce each block by Lemma 5.1 to $\frac{1}{2} \log n + \log \log n = \mathrm{O}(\log n)$ qubits in constant depth and size $\sqrt{n} \log^2 n$. In total, we obtain $\sqrt{n}$ new qubits in size $n \log n$. Second, compute the logical Or by Theorem 4.1 in constant depth, size $\sqrt{n}^2 \log \sqrt{n} = \mathrm{O}(n \log n)$, and error $\frac{1}{\sqrt{n}}$. To amplify the error to $\frac{1}{n}$, repeat the computation twice and return 1 if any of them returns 1 (the error is one-sided). The circuit size is doubled. □

The circuit size can be reduced to $\mathrm{O}(n \log^{(d)} n)$ for any constant number d of iterations of the logarithm. The trick is to divide input qubits into small blocks and perform the reduction step on each of them. The number of variables is reduced by a small factor and we can thus afford to apply a circuit of a slightly bigger size. It we repeat this reduction step d times, we obtain the desired circuit.

Theorem 6.3. *There exist constants c_1, c_2 such that for every $d \in \mathbb{N}$, there is an Or-reduction $n \to 1$ with error $\frac{1}{n}$, depth $c_1 d$, and size $c_2 dn \log^{(d)} n$.*

Proof. By induction on d: we have already verified the case $d = 1$ in Lemma 6.2. For the induction step: Divide n input qubits into $n/\log^{(d-1)} n$ blocks of $\log^{(d-1)} n$ qubits. Using Lemma 5.1, reduce each block to $\log^{(d)} n$ qubits in constant depth and size $c_2 \log^{(d-1)} n \cdot \log^{(d)} n$. Total size is $c_2 n \log^{(d)} n$. We obtain $\frac{n}{\log^{(d-1)} n} \log^{(d)} n = o(n)$ new qubits. Using the induction hypothesis, compute their logical Or in depth $c_1(d-1)$ and size $c_2(d-1)\left(\frac{n}{\log^{(d-1)} n} \log^{(d)} n\right) \cdot \log^{(d-1)} o(n) \le c_2(d-1) n \log^{(d)} n$. Together, it takes depth $c_1 d$ and size $c_2 dn \log^{(d)} n$.

The only approximate step is the application of Lemma 6.2 for $d = 1$. It is applied on $\frac{n}{\log n} \log^{(d)} n$ variables, hence the error is $\mathrm{O}(\log n/n)$. It can be amplified to $\frac{1}{n}$ by running the computation twice. ☐

6.2 Log-star depth computation of Or

Our best constant-depth circuit for Or is described by Theorem 6.3. It is approximate and it has slightly super-linear size. In this section, we show that we can achieve an *exact* circuit of *linear* size if we relax the restriction of constant depth. We consider d in Theorem 6.3 a slowly growing function of n instead of a constant. Now we can use an Or-reduction better than Lemma 6.2. Theorem 5.3 provides an Or-reduction $n \to 1$ with error 0, log-star depth, and size $n \log n$.

Lemma 6.4. *There exist constants c_1, c_2 such that for every $d \in \mathbb{N}$, there is an Or-reduction $n \to 1$ with error 0, depth $c_1 d + \log^* n$, and size $c_2 dn \log^{(d)} n$.*

Proof. The same as of Theorem 6.3, but use the Or-reduction from Theorem 5.3 instead of Lemma 6.2 in the last layer (for $d = 1$). The size stays roughly the same, the circuit becomes exact, and the depth is increased by an additional term of $\log^* n$. ☐

Theorem 6.5. *There is an Or-reduction $n \to 1$ with error 0, log-star depth, and linear size.*

Proof. Divide the input into $\frac{n}{\log^* n}$ blocks of size $\log^* n$. Compute the logical Or of each block by a balanced binary tree of depth $\log(\log^* n) < \log^* n$ and in *linear* size. Using Lemma 6.4 with $d = \log^* n$, compute the logical Or of $\frac{n}{\log^* n}$ new qubits in log-star depth and size $\mathrm{O}\left(\log^* n \cdot \frac{n}{\log^* n} \cdot \log^{(\log^* n)} n\right) = \mathrm{O}(n)$. ☐

6.3 Approximation of counting and threshold[t]

In this section, we use the QFT for the parallelisation of increments. This allows us to approximate the Hamming weight of the input in smaller size $\mathrm{O}(n \log n)$.

Definition 6.6. The increment gate maps $\mathrm{Incr}_n : |x\rangle \to |(x+1) \mod 2^n\rangle$.

Lemma 6.7. *The increment gate is diagonal in the Fourier basis and its diagonal form is in* QNC^0.

Proof. Let $\omega = e^{2\pi i/2^n}$ and let $|x\rangle$ be any computational basis state. It is simple to prove the following two equations:

1. $\mathrm{Incr}_n = F_{2^n}^\dagger D_n F_{2^n}$ for diagonal $D_n = \sum_{y=0}^{2^n-1} \omega^y |y\rangle\langle y|$.

$$F^\dagger DF|x\rangle = F^\dagger D \frac{\sum_{y=0}^{2^n-1} \omega^{xy}|y\rangle}{\sqrt{2^n}} = F^\dagger \frac{\sum_{y=0}^{2^n-1} \omega^{(x+1)y}|y\rangle}{\sqrt{2^n}} = |(x+1) \mod 2^n\rangle \ .$$

2. $D = R_z(\pi) \otimes R_z(\pi/2) \otimes \cdots \otimes R_z\left(\pi/2^{n-1}\right)$.

$$\begin{aligned} D|x\rangle &= \omega^x|x\rangle = \bigotimes_{k=1}^{n} \omega^{2^{n-k}x_{n-k}}|x_{n-k}\rangle = \bigotimes_{k=1}^{n} (e^{2\pi i/2^k})^{x_{n-k}}|x_{n-k}\rangle = \\ &= \bigotimes_{k=1}^{n} R_z\left(2\pi/2^k\right)|x_{n-k}\rangle = \left(R_z(\pi) \otimes \cdots \otimes R_z\left(\pi/2^{n-1}\right)\right)|x\rangle. \end{aligned}$$

We conclude that $\mathrm{Incr} = F^\dagger DF$, and that D is a tensor product of one-qubit operators. □

Remark 6.8. The addition of a fixed integer b is as hard as the increment. By Lemma 6.7, $\mathrm{Incr}^b = F^\dagger D^b F$ and $(R_z(\varphi))^b = R_z(\varphi b)$, hence the diagonal version of the addition of b is also in QNC^0.

Theorem 6.9. *Counting can be approximated in constant depth and size* $\mathrm{O}(n \log n)$.

Proof. Compute the Hamming weight of the input. Each input qubit controls one increment on an m-qubit counter initialised to 0, where $m = \lceil \log(n+1) \rceil$. The increments Incr_m are parallelised (Theorem 3.2 and Lemma 6.7), so we apply the quantum Fourier transform F_{2^m} twice (Theorem 4.12) and the n constant-depth controlled D_m gates in parallel. The size is $\mathrm{O}(\mathrm{poly}(m) + nm) = \mathrm{O}(n \log n)$. □

Remark 6.10. threshold[t] is equal to the most significant qubit of the counter if we align it to a power of 2 by adding a fixed integer $2^m - t$. exact[t] can be computed by comparing the counter with t.

7 Concluding remarks

7.1 Comparison with randomised circuits

Let us compare our results for quantum circuits with similar results for classical randomised circuits. We consider randomised circuits with bounded fan-in of Or and And gates, and unbounded fan-out and parity (similar to the quantum model). Classical lower bounds are folklore and we attach the proofs for the convenience of the reader in Appendix A.

Gate	Randomised	Quantum
Or and threshold[t] exactly	$\Theta(\log n)$	$\mathrm{O}(\log^* n)$
mod[q] exactly	$\Theta(\log n)$	$\Theta(1)$
Or with error $\frac{1}{n}$	$\Theta(\log\log n)$	$\Theta(1)$
threshold[t] with error $\frac{1}{n}$	$\Omega(\log\log n)$	$\Theta(1)$

7.2 Relations of quantum circuit classes

We have shown that $\text{B-QNC}^0_\text{f} = \text{B-QAC}^0_\text{f} = \text{B-QACC}^0 = \text{B-QTC}^0_\text{f}$ (Theorem 4.6). If we allow arbitrary one-qubit gates, then also $\text{QTC}^0_\text{f} = \text{QAC}^0_\text{f} \subseteq \text{QNC}_\text{f}(\log^* n)$ (Theorem 5.2 and Theorem 5.3). Several open problems of [10] have thus been solved. Only little is known about classes that do not include the fan-out gate. For example, we do not know whether $\text{TC}^0 \subseteq \text{QTC}^0$, we only know that $\text{TC}^0 \subseteq \text{QTC}^0_\text{f}$. It is simple to prove that parity is in TC^0. Take the logical Or of exact[1], exact[3], exact[5], ..., and compute exact[k] from threshold[k] and threshold[$k+1$]. However, this method needs fan-out to copy the input bits and hence it is not in QTC^0.

Fang et al. proved [7] a lower bound for fan-out. In particular, they showed that logarithmic depth is needed to approximate parity using only a constant number of ancillas. Unfortunately, their method breaks down with more than a linear number of ancillas and it cannot be extended to other unbounded fan-in gates such as majority or threshold[t].

7.3 Upper bounds for B-QNC^0_f

Shor's original factoring algorithm [19] uses modular exponentiation and the quantum Fourier transform modulo 2^n followed by a polynomial-time deterministic algorithm. The modular exponentiation a^x can be replaced by multiplication of some subset of numbers $a, a^2, a^4, \dots, a^{2^{n-1}}$ [5]. The n numbers a^{2^k} can be quickly precomputed classically.

Since both multiplication of n numbers (Theorem 4.8) and the QFT (Theorem 4.12) are in B-QNC^0_f, there is a polynomial-time bounded-error classical algorithm with oracle B-QNC^0_f factoring numbers, i.e. factoring $\in \text{RP}[\text{B-QNC}^0_\text{f}]$. If $\text{B-QNC}^0_\text{f} \subseteq \text{BPP}$,[3] then factoring $\in \text{RP}[\text{BPP}] \subseteq \text{BPP}[\text{BPP}] = \text{BPP}$. Discrete logarithms can be computed in a similar way using modular exponentiation and the quantum Fourier transform modulo general q [19]. Since $\text{QFT}_q \in \text{B-QNC}^0_\text{f}$ (Theorem 4.17), we conclude that also discrete-log $\in \text{RP}[\text{B-QNC}^0_\text{f}]$.

7.4 Open problems

We propose the following open problems on computational aspects of multi-qubit gates:

i. Is there a constant-depth exact circuit for Or?

ii. Is there a constant-depth linear-size circuit for Or?

iii. Are there exact circuits with a fixed basis?

iv. Can we simulate unbounded fan-out in constant depth using unbounded fan-in gates, e.g. threshold[t] or exact[t]?

[3] In this context, B-QNC^0_f denotes the set of languages decided with bounded error by constant-depth quantum circuits with fan-out.

A Lower bounds on classical circuits

Using the polynomial method [3], we prove several lower bounds on the depths of deterministic circuits. We consider circuits with fan-in of Or and And gates at most 2, and unbounded fan-out and parity, the same as in the quantum model.

Basically, the value of each bit computed by a circuit can be computed by a multi-linear polynomial (over the field $\mathbb{Z}_2$) in the input bits. We are interested in the degree of such a polynomial; by proving a lower bound on the degree, we also lower-bound the depth of the circuit. It is simple to prove that the polynomial computing a Boolean function is unique.

Each input bit $x_k \in \{0,1\}$ is computed by the polynomial x_k of degree 1. The Not gate computes the polynomial $1-p(x)$, where $p(x)$ is the polynomial computing its argument, and the degree is unchanged. The And gate computes the polynomial $p_1(x) \cdot p_2(x)$ and the two degrees are summed. The parity gate computes the polynomial $(p_1(x)+\cdots+p_k(x)) \mod 2$ of degree equal to the maximum degree among the arguments.

Lemma A.1. *The output of a circuit of depth d has degree at most 2^d.*

Proof. By induction: by adding a new layer, we can at most double the degree when using the And gate. □

And of n bits is computed by a (unique) polynomial $x_1x_2\ldots x_n$ of degree n. Hence every circuit computing And has depth at least $\log n$. It is simple to prove by contradiction that also Or, threshold[t], and exact[t] have full degree n. Smolensky has proved a much stronger result [21], which implies that also the degree of mod[q] for $q>2$ is n.

Randomised circuits have access to random bits and may produce the result with a small error. Some functions are computed in smaller depth in this model.

Lemma A.2. *Or can be computed with one-sided error $\frac{1}{2}$ by a randomised circuit of depth 2. The error can be decreased to $\frac{1}{n}$ in additional depth $\log\log n$.*

Proof. Take n random bits and output the parity $x_1r_1 \oplus x_2r_2 \oplus \cdots \oplus x_nr_n$. If $|x|=0$, then the circuit always outputs 0. If $|x|>0$, then the probability that the parity is odd is equal to $\frac{1}{2}$. If we perform the computation $(\log n)$-times using independent random bits, we decrease the probability of error to $(\frac{1}{2})^{\log n} = \frac{1}{n}$. This can be done in additional depth $\log\log n$ by a balanced binary tree of Or gates. □

By Yao's principle [24], if we have a randomised circuit with error less than 2^{-n}, then there exists an assignment of random bits such that the result is always correct. That is there exists a deterministic circuit of the same shape. Hence also randomised circuits computing the logical Or with exponentially small error have depth at least $\log n$.

Lemma A.3. *Every circuit computing Or with error $\frac{1}{n}$ has depth at least $\log\log n$.*

Proof. Assume the converse: there exists a circuit of depth $d < \log\log n$ with error $\frac{1}{n}$. By computing the logical Or independently $\frac{n}{\log n}$-times, we can reduce the error to $(\frac{1}{n})^{\frac{n}{\log n}} = 2^{-n}$. This can be done in additional depth $\log\frac{n}{\log n} = \log n - \log\log n$. The total depth of this circuit is $\log n - \log\log n + d < \log n$. However, by Yao's principle, the depth has to be at least $\log n$. □

Acknowledgements

We thank Harry Buhrman, Hartmut Klauck, and Hein Röhrig at CWI in Amsterdam, and Fred Green at Clark University in Worcester for plenty of helpful discussions, and Ronald de Wolf at CWI for help with writing the paper. We thank Klaus Mölmer and Brian King for discussions on physical implementations of multi-qubit gates. We are grateful to Schloss Dagstuhl, Germany, for providing an excellent environment, where part of this work was carried out. We thank the anonymous reviewers for their valuable comments.

References

[1] * L. M. ADLEMAN, J. DEMARRAIS, AND M. A. HUANG: Quantum computability. *SIAM Journal on Computing*, 26(5):1524–1540, 1997. [SICOMP:29363]. 2, 3.5

[2] * A. BARENCO, C. BENNETT, R. CLEVE, D. P. DIVINCENZO, N. MARGOLUS, P. SHOR, T. SLEATOR, J. A. SMOLIN, AND H. WEINFURTER: Elementary gates for quantum computation. *Physical Review A*, 52:3457–3467, 1995. [PRA:10.1103/PhysRevA.52.3457, arXiv:quant-ph/9503016]. 2.5

[3] * R. BEIGEL: The polynomial method in circuit complexity. In *Proc. of 8th IEEE Structure in Complexity Theory Conf.*, pp. 82–95, 1993. [1993.336538]. A

[4] * J. I. CIRAC AND P. ZOLLER: Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091–4094, 1995. [PRL:10.1103/PhysRevLett.74.4091]. 1

[5] * R. CLEVE AND J. WATROUS: Fast parallel circuits for the quantum Fourier transform. In *Proc. of 41st IEEE FOCS*, pp. 526–536, 2000. [FOCS:10.1109/SFCS.2000.892140]. 1, 4.4.1, 4.4.1, 4.4.2, 4.4.2, 7.3

[6] * D. COPPERSMITH: An approximate Fourier transform useful in quantum factoring. IBM technical report RC19642, quant-ph/0201067, 1994. [arXiv:quant-ph/0201067]. 4.4.1

[7] * M. FANG, S. FENNER, F. GREEN, S. HOMER, AND Y. ZHANG: Quantum lower bounds for fanout. 2003. [arXiv:quant-ph/0312208]. 7.2

[8] * S. A. FENNER: Implementing the fanout gate by a Hamiltonian. 2003. [arXiv:quant-ph/0309163]. 1

[9] * N. GERSHENFELD AND I. CHUANG: Bulk spin resonance quantum computation. *Science*, 275:350–356, 1997. [doi:10.1126/science.275.5298.350]. 1

[10] * F. GREEN, S. HOMER, C. MOORE, AND C. POLLETT: Counting, fanout, and the complexity of quantum ACC. *Quantum Information and Computation*, 2(1):35–65, 2002. [arXiv:quant-ph/0106017]. 1, 3.2, 7.2

[11] * L. HALES AND S. HALLGREN: Quantum Fourier sampling simplified. In *Proc. of 31st ACM STOC*, pp. 330–338, 1999. [STOC:301250.301336]. 1, 4.4.2, 4.4.2

[12] * W. HOEFFDING: Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963. 4.2

[13] * R. A. HORN AND C. R. JOHNSON: *Matrix Analysis*. Cambridge University Press, 1985. 3.1

[14] * P. HØYER AND R. ŠPALEK: Quantum circuits with unbounded fan-out. In *Proc. of 20th STACS*, pp. 234–246, 2003. LNCS 2607. [STACS:80j4ju67n25kcf06].

[15] * K. MØLMER AND A. SØRENSEN: Multiparticle entanglement of hot trapped ions. *Phys. Rev. Lett.*, 82:1835–1838, 1999. [PRL:10.1103/PhysRevLett.82.1835]. 1

[16] * C. MOORE: Quantum circuits: Fanout, parity, and counting. 1999. [arXiv:quant-ph/9903046]. 1, 2.2, 3.3

[17] * C. MOORE AND M. NILSSON: Parallel quantum computation and quantum codes. *SIAM Journal on Computing*, 31(3):799–815, 2002. [SICOMP:35505, arXiv:quant-ph/9808027]. 1, 3.2

[18] * A. A. RAZBOROV: Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$. *Math. Notes Acad. Sci. USSR*, 41(4):333–338, 1987. 1

[19] * P. W. SHOR: Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. of 35th IEEE FOCS*, pp. 124–134, 1994. [FOCS:10.1109/SFCS.1994.365700]. 1, 4.4, 7.3

[20] * K.-Y. SIU, J. BRUCK, T. KAILATH, AND T. HOFMEISTER: Depth efficient neural networks for division and related problems. *IEEE Transactions on Information Theory*, 39(3):946–956, 1993. [TIT:10.1109/18.256501]. 1, 4.3

[21] * R. SMOLENSKY: Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. of 19th ACM STOC*, pp. 77–82, 1987. [STOC:28395.28404]. 1, A

[22] * R. ŠPALEK: Quantum circuits with unbounded fan-out. Master's thesis, Faculty of Sciences, Vrije Universiteit, Amsterdam, 2002. Shorter version and improved results in quant-ph/0208043. [arXiv:quant-ph/0208043].

[23] * W. K. WOOTTERS AND W. H. ZUREK: A single quantum cannot be cloned. *Nature*, 299:802–803, 1982. [doi:10.1038/299802a0]. 2.1

[24] * A. C-C. YAO: Probabilistic computations: Toward a unified measure of complexity. In *Proc. of 18th IEEE FOCS*, pp. 222–227, 1977. A

AUTHORS

Peter Høyer
professor
Department of Computer Science, University of Calgary
Alberta, Canada
hoyer@cpsc.ucalgary.ca
http://www.cpsc.ucalgary.ca/~hoyer/

Robert Špalek
graduate student
Centrum voor Wiskunde en Informatica
Amsterdam, The Netherlands
sr@cwi.nl
http://www.ucw.cz/~robert/

ABOUT THE AUTHORS

PETER HØYER received his Ph.D. in Computer Science from the University of Southern Denmark, Odense Campus, under the supervision of Joan Boyar and Gilles Brassard. His research interests include algorithmics, data structures, and quantum information. He is currently working on having the algorithms of the present article to be included as DLL's in the final release of Vista, formerly known as Longhorn.

ROBERT ŠPALEK received his Masters Degrees in Computer Science from Charles University, Prague and Vrije Universiteit, Amsterdam. He is currently a graduate student at CWI, advised by Harry Buhrman. His research interests include quantum computing, computational complexity, algorithms, data structures, and search engines. He also enjoys climbing, salsa, photography, travelling to distant countries, and playing guitar.