SIGACT News Complexity Theory Column 55

Lane A. Hemaspaandra
Dept. of Computer Science, University of Rochester
Rochester, NY 14627, USA

Introduction to Complexity Theory Column 55

Warmest thanks to Debajyoti, Fred, and Steve for this issue's column on small-depth quantum circuits. Please stay tuned to future columns to hear from Salil Vadhan on the complexity of zero knowledge and from Arnaud Durand, (the late) Clemens Lautemann, and Malika More on "A Simple Proof of Polylog Counting Ability of First-Order Logic." And wishing everyone a theorem-filled summer.

Guest Column: Small Depth Quantum Circuits[1]
Debajyoti Bera,[2] *Frederic Green,*[3] *Steven Homer*[2]

Abstract

Small depth quantum circuits have proved to be unexpectedly powerful in comparison to their classical counterparts. We survey some of the recent work on this and present some open problems.

1 Introduction and Motivation

Quantum circuits are the most natural and general formulation of quantum computation. They are general in the sense that they are a universal model; any quantum computation can be efficiently simulated by a quantum circuit [27, 19]. They are natural in that a quantum computation (or quantum algorithm) can best be understood as a collection of qubits being acted on by quantum operators represented as (defined by) a tensor product of quantum gates. Although the quantum circuit model is quite different than the classical one, it has nevertheless proven to be quite fruitful to look to classical circuit models for insight. Classical circuits of small depth[4] (e.g., polylog as in the class NC) have been proposed as realistic models of parallel computation. Furthermore, *very* small (i.e., constant) depth classical circuits present us with computational models for which we can actually prove interesting lower bounds. What happens when we extend these concepts to quantum circuits?

The answer is a bit surprising, and leads fairly quickly to interesting variants of the fundamental problems of quantum computing. For example, consider the class AC^0 of constant-depth,

---

		Form Approved OMB No. 0704-0188
	Report Documentation Page	

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **SIGACT News Complexity Theory Column 55**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Rochester,Dept. of Computer Science,Rochester,NY,14627**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

polynomial-size circuits consisting of NOT together with unbounded fan-in and fan-out AND and OR gates. Some interesting functions can be computed in this class (addition of n-bit numbers for example), and some cannot (e.g., parity [11]). It would be interesting to see what an analogous quantum complexity class would look like. But if we try to translate the class AC^0 into the quantum setting, we are faced with an immediate problem. The unbounded-fanin "quantum AND gate," the generalized Toffoli gate (which also encompasses negation and hence OR as well) does not allow for fanout in any way we may take for granted. The reason is the "no-cloning theorem," which says that it is in general not possible to make a copy of a quantum state. One can, however, make copies of classical bits, which suggests the idea of introducing a unitary operator to implement fanout of classical bits. (A similar operation is really implicit in the AC^0 model; we solder multiple outgoing wires to an AND gate, obtaining copies of the output. While it is not entirely realistic to have an unbounded number of wires fanning out of the gate even classically, this is a useful abstraction out of which the definition of AC^0 arises.)

We therefore seem to have two candidates for quantum analogs of AC^0. One, which just includes generalized Toffoli gates and single-qubit gates, is called QAC^0. The other, which includes fanout gates, is called $\mathsf{QAC}^0_{\mathrm{wf}}$. The latter appears to be the more realistic version, since it is straightforward to see that it includes AC^0. However, we will see in this article that $\mathsf{QAC}^0_{\mathrm{wf}}$ is much more powerful than its classical counterpart. There nevertheless remains much that we do not understand. For example, the classical version of QAC^0 (in which AND and OR gates have at most constant fanout) is provably weaker than AC^0. In stark contrast, it is unknown to this day if QAC^0 is the same as $\mathsf{QAC}^0_{\mathrm{wf}}$.

This gives an interesting theoretical framework in which quantum analogs of classical circuit models are provably more powerful. But does it have bearing on reality? It might. Realizable quantum computations will have very limited duration, due to short coherence times, which suggests that highly parallelized quantum circuits (such as those we will see can be obtained using fanout) are desirable. Fanout gates as well as Toffoli gates might actually be feasible to build via ion trap [3] or bulk NMR techniques [12]. It may therefore be of more than mere theoretical interest to explore their power, both intrinsic and relative to each other.

2 Circuit Elements and Classes

In this brief survey, we will be unable to provide a review of quantum computation. For a quick introduction, we recommend Fenner [7] or Fortnow and Rogers [10]; for an in-depth treatment, see Nielson and Chuang [17].

To make the treatment as self-contained as possible, we introduce the following notation. Let $\mathcal{H}$ denote the 2-dimensional Hilbert space spanned by the computational basis states $|0\rangle, |1\rangle$. Let $\mathcal{H}_1, \ldots, \mathcal{H}_n$ be n copies of $\mathcal{H}$. By $\mathcal{B}_n$ we denote the 2^n-dimensional Hilbert space $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$ spanned by the usual set of computational basis states of the form $|x_1, \ldots, x_n\rangle$, where each $x_i \in \{0, 1\}$. A "state over a set of n bits" is a state in $\mathcal{B}_n$. Let $\mathcal{U}_n$ denote the set of unitary matrices that act on states in $\mathcal{B}_n$ ($\mathcal{U}_n$ is just a convenient notation for the group $U(2^n)$). A quantum gate G corresponds to an element of $\mathcal{U}_n$, which is also denoted G. Thus, for example, a *single-qubit gate* is an element of $\mathcal{U}_1$, acting on states in $\mathcal{B}_1$.

We now exhibit the main quantum gates that we will consider. The single-qubit *Hadamard gate*

is defined by,

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Let $f : \{0,1\}^n \longrightarrow \{0,1\}$ be a Boolean function of n inputs. Many of our gates take the form,

$$G|x_1, ..., x_n, b\rangle = |x_1, ..., x_n, b \oplus f(x_1, \ldots, x_n)\rangle$$

This is just an easy form of simulating any classical function in a reversible manner. If $f(x_1, \ldots, x_n) = \wedge_{i=1}^n x_i$, G is called a *generalized Toffoli gate*, or (in this paper) simply a *Toffoli* gate, and is written as T. If $f(x_1, \ldots, x_n) = \bigoplus_{i=1}^n x_i$, G is a *parity* gate, written P. Generalizing this, define the classical Boolean function $\mathrm{Mod}_q : \{0,1\}^n \longrightarrow \{0,1\}$ so that $\mathrm{Mod}_q(x_1, \ldots, x_n) = 1$ iff $\sum_{i=1}^n x_i \not\equiv 0 \pmod{q}$. If $f = \mathrm{Mod}_q$, we call G a MOD_q gate. Next, for any t, define the boolean function $s(x_1, ..., x_n) = 1$ iff $\sum_{i=1}^n x_i \geq t$. If $f = s$, we call G a *threshold gate* and write it as S. Finally, the *fanout* gate F is defined by,

$$F|x_1, ..., x_n, b\rangle \quad = \quad |b \oplus x_1, ..., b \oplus x_n, b\rangle.$$

It is known that the T-gate for $n = 1$ (known as controlled-not, or "CNOT") together with single-qubit gates (in particular, the Hadamard, phase, and $\pi/8$ gates) are a universal set of gates in that any unitary operator can be approximated to an arbitrary degree of precision with them.

A quantum circuit is constructed out of layers. Each layer L is a tensor product of a certain fixed set of gates. A circuit is simply a (matrix) product of layers $L_1 L_2 \cdots L_d$. (Observe that the "last" layer L_d is actually the one that is applied directly to the inputs, and L_1 is the output layer.) The number of layers d is called the *depth* of C. A circuit C over n qubits is then a unitary operator in $\mathcal{U}_n$. Clearly, C computes a unitary operator U *exactly* if for all computational basis states, $C|x_1, ..., x_n\rangle = U|x_1, ..., x_n\rangle$. This is in general too restrictive, however. One must allow for the presence of "work bits," called *ancillæ*, that make extra space available in which to do a computation. In that case, in order to exactly compute the operator U we extend the Hilbert space in which C acts to the 2^{n+m}-dimensional space $\mathcal{B}_{n+m}$ spanned by computational basis states $|x_1, ..., x_n, a_1, ..., a_m\rangle$, where again $x_i, a_i \in \{0,1\}$, the a_i serving as ancillæ. Then we say that C *cleanly computes* U if, for any $x_1, ..., x_n$ and $y_1, ..., y_n$,

$$\langle y_1, ..., y_n, 0, ..., 0|C|x_1, ..., x_n, 0, ..., 0\rangle = \langle y_1, ..., y_n, 0, ..., 0|(U \otimes I)|x_1, ..., x_n, 0, ..., 0\rangle,$$

where I is the identity in the subspace that acts on the ancillæ, and the number of 0's in each state above is m. That is, C does a clean computation if the ancillæ begin and end all as 0's. We assume all of our circuits perform clean computations. This is a reasonable constraint, since only then is it easy to compose the circuits.

All circuits should be understood to be elements of an infinite *family* of circuits $\{C_n | n \geq 0\}$, where C_n is a quantum circuit for n input qubits and each circuit family contains a fixed finite set of gates.

In this paper we deal with various quantum circuit classes which are defined in analogy with the classical circuit classes, e.g., similar to NC^k, QNC^k is defined as $\log^k n$ depth, polynomial size circuits containing only single qubit and CNOT gates. We list some of the quantum circuit classes below:

Definition 2.1 Quantum circuit classes

    Quantum analogues of NC^k

        QNC^k: *consisting of single qubit and CNOT gates (Toffoli gates with $n = 1$).*

        $\mathsf{QNC}^k_{\mathrm{wf}}$: QNC^k *+ fanout gates.*

    Quantum analogues of AC^k

        QAC^k: *consisting of single qubit and Toffoli gates.*

        $\mathsf{QAC}^k_{\mathrm{wf}}$: *consisting of single qubit, Toffoli and fan-out gates.*

    Quantum analogues of ACC^k

        QACC^k: $\mathsf{QACC}^k[q]$ *is* QAC^k *+* MOD_q *gates.* $\mathsf{QACC}^k = \cup_q \mathsf{QACC}^k[q]$.

        QACC: QACC *is defined as* QACC^0 *and* $\mathsf{QACC}[q]$ *is defined as* $\mathsf{QACC}^0[q]$.

    Quantum analogues of TC^k

        QTC^k: QAC^k *+ arbitrary fanin threshold gates.*

        $\mathsf{QTC}^k_{\mathrm{wf}}$: $\mathsf{QAC}^k_{\mathrm{wf}}$ *+ arbitrary fanin threshold gates.*

It should be emphasized that these are classes of unitary operators. There are various ways in which they can be used to define classes of sets, but we will not explore that here.

This is an unacceptably large array of complexity class definitions (perhaps even gratuitous). Surprisingly and happily, however, most of them are either the same or are very close, unlike the corresponding classical classes.

3 Upper Bounds

The first hint that something different is going on in quantum circuits is in the intimate relationship between fanout and parity. There is no obvious *a priori* relation between these operators, and indeed we wouldn't expect there to be any on the basis of our experience with classical circuits. But as was observed by Moore [15], F is conjugate to P via an $(n + 1)$-fold tensor product of Hadamards applied to all the bits:

$$F = H^{\otimes(n+1)} P H^{\otimes(n+1)} \tag{1}$$

This is a consequence of the well-known fact that a CNOT gate, conjugated with Hadamards, flips the input and target bits (see Figure 1).
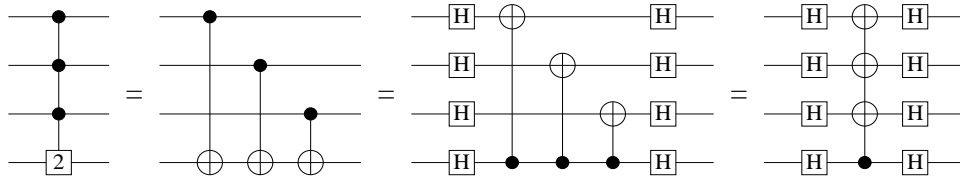

Figure 1: The parity and fanout gates are conjugates of each other by a layer of Hadamard gates.

It is immediate from this that $\mathsf{QAC}^0_{\mathrm{wf}} = \mathsf{QACC}_{\mathrm{wf}}[2] = \mathsf{QACC}[2]$. Contrast this with the famous classical result of Furst, Saxe and Sipser [11] that parity is not in AC^0 (and hence $\mathsf{ACC}[2] \neq \mathsf{AC}^0$).

It is also immediate that for all k, $\mathsf{QAC}^k_{\mathrm{wf}} = \mathsf{QACC}^k_{\mathrm{wf}}[2] = \mathsf{QACC}^k[2]$. Since it is possible to fanout n copies in $\log n$ depth using CNOT (via divide and conquer), we also have $\mathsf{QAC}^k \subseteq \mathsf{QAC}^k_{\mathrm{wf}} \subseteq \mathsf{QAC}^{k+1}$ (subsequent results in this survey lead to similar relationships for the $k > 0$ classes).

The $\mathsf{QACC}[q]$ classes for $q \neq 2$ present a more subtle problem. Recall the Razborov/Smolensky Theorem [21, 24] that says that for any relatively prime q, p, $\mathsf{ACC}[q] \neq \mathsf{ACC}[p]$, and hence $\mathsf{ACC}[q] \not\subseteq \mathsf{AC}^0$. One would think, in accordance with this, that the $\mathsf{QACC}[q]$ classes are all incomparable with $\mathsf{QACC}[2]$. In fact the opposite turns out to be true: $\mathsf{QACC}[q] = \mathsf{QACC}[2]$ for all q.
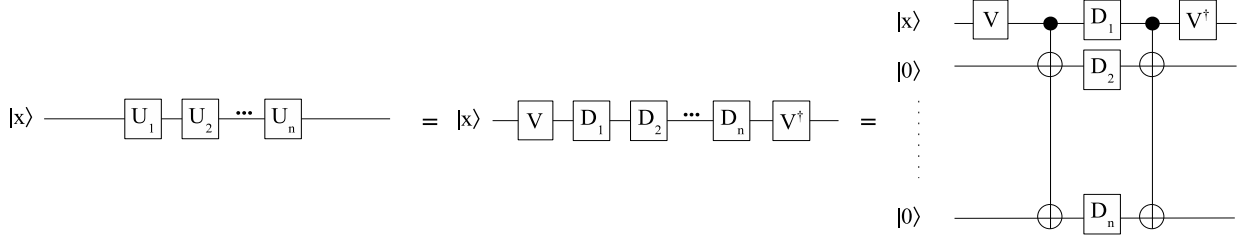

Figure 2: If a group of gates $U_1, \ldots, U_n$ are simultaneously diagonalizable ($U_i = V D_i V^\dagger$) or are the same, then they can be applied in parallel using fanout gates. This holds even if the gates are controlled by some control qubit.

The proof that $\mathsf{QACC}[q] \subseteq \mathsf{QACC}[2]$ uses the most important technique to date in this area: parallelization, which was first observed by Moore and Nilsson [16]. A series of commuting (for example, identical) unitary operations can be implemented in a constant number of layers by diagonalizing the operators and using fanout to apply the diagonal operators in parallel. The process is sketched in Figure 2. In this case, the operations we want to parallelize are those that increment a register mod q:

$$M_q|x\rangle = |(x + 1) \bmod q\rangle,$$

where x denotes n bits. Applying n controlled M_q gates in series (each M_q controlled by one of the x_i's) leaves the sum of the bits mod q in a register. Once this is done, applying a Toffoli gate to the register yields $\mathrm{Mod}_q(x_1, \ldots, x_n)$. Thus, via parallelization, we can do MOD_q in constant depth using fanout.

The other direction, $\mathsf{QACC}[2] \subseteq \mathsf{QACC}[q]$, essentially amounts to showing that there's nothing really special about "2" in the conjugacy relation between fanout and parity. This relationship can be generalized to an analogous one between "fanout of digits base q" and "sum of the inputs mod q." This requires a Fourier transform which works on "quantum *digit*" (rather than qu*bit*) registers, which is analogous to the Hadamard gate. By a result of Barenco et al. [2], such a constant-dimensional unitary transformation can be realized in constant depth with just CNOT and single-qubit gates. With considerable added circuitry to represent the digits as bits, these techniques result in the following:

Theorem 3.1 *[13] For all q, $\mathsf{QAC}^0_{\mathrm{wf}} = \mathsf{QACC}[q] = \mathsf{QACC}$.*

The biggest surprise is yet to come. Consider the class QNC^0. Like its classical counterpart, it is not very useful; as we explain later on, no output can depend on all the inputs! What if we add fanout, as in the class $\mathsf{QNC}^0_{\mathrm{wf}}$? Høyer and Špalek showed that with high accuracy, one can compute *threshold* functions with only fanout and single-qubit gates, in constant depth!

Theorem 3.2 *[14] Let $\{F_n\}$ be a family of operators in $\mathsf{QNC}^0_{\mathrm{wf}}$, $\mathsf{QAC}^0_{\mathrm{wf}}$, or $\mathsf{QTC}^0_{\mathrm{wf}}$. Then there is a family of operators $\{G_n\}$ in either of the other classes that approximates $\{F_n\}$ with two-sided polynomially small error.*

In a strong sense, therefore, the classes $\mathsf{QNC}^0_{\mathrm{wf}}$, $\mathsf{QAC}^0_{\mathrm{wf}}$, $\mathsf{QTC}^0_{\mathrm{wf}}$ (to say nothing of QACC and all its kindred) are equivalent in computational power. The theorem essentially says that fanout and single-qubit operations form a universal set of quantum gates.

Proof sketch: The proof of Theorem 3.2 is centered around the parallelization method. To give the clearest exposition possible, we find it advantageous to follow (initially) an earlier technique of Špalek [25]. Following exactly his line of reasoning, we sketch how to simulate an *exact* threshold gate, in which the boolean function f in the definition is 1 iff $\sum_{i=1}^{n} x_i = t$; using this, threshold gates can be easily constructed. The main roadblock in generalizing Theorem 3.1 to Theorem 3.2 is that the proof of the former relies heavily on the fixed dimensionality of the mod q increment operator which, by Barenco et al., can be simulated in constant depth. In order to implement an exact gate, we need to compute the sum of the inputs not mod q, but mod n, where n is the number of inputs (technically, mod $n + 1$). Denoting $\log n$ as k, the required k-bit increment operator M is defined as,

$$M|x\rangle = |(x+1) \bmod 2^k\rangle,$$

where x denotes a k-bit number. M grows exponentially in k so we can no longer rely on Barenco et al. The way around this problem starts with a simple but quite interesting observation, namely that M is diagonal in the Fourier basis. This is the basis of the Hilbert space that is obtained when we perform the quantum Fourier transform,

$$Q|x\rangle = \frac{1}{2^{k/2}} \sum_{y=0}^{2^k - 1} \omega^{xy}|y\rangle,$$

where $\omega = e^{2\pi i/2^k}$. Specifically, a straightforward computation shows that $M = Q^\dagger D Q$, where $D = \mathrm{diag}(1, \omega, \omega^2, \omega^3, \ldots, \omega^{2^k - 1})$. Although D is a "big" operator, it nevertheless can be written as a tensor product of single-qubit operators $|b\rangle \mapsto \omega^b|b\rangle$ where $b \in \{0, 1\}$. Thus if x is a k-bit string, $D|x\rangle = \omega^{\sum_{i=1}^{k} x_i}|x\rangle = \otimes_{i=1}^{k}(\omega^{x_i}|x_i\rangle)$.
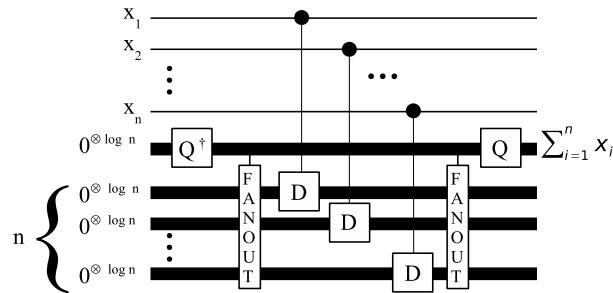

Figure 3: Compute $\sum_{i=1}^{n} x_i$ in constant depth by using Q, fanout gates and controlled-D gates. The n D-gates are parallelized here to reduce the depth.

For computing the sum of the inputs, we really need the M (and hence D) operator to be *controlled* by all the respective inputs. In fact, using known properties of $\mathcal{U}_1$, the *controlled D*

operator can be implemented in constant depth with single-qubit and CNOT gates. The next step is to use this to compute $\sum_{i=1}^{n} x_i$, and indeed this can be done with the aid of Q as shown in figure 3. In the working register of $\log n$ bits, the output is *exactly* the sum of the x_i's. The problem is, we don't know if Q can be implemented in constant depth (we will see that it can, but we must avoid circular reasoning). Špalek's inspired fix to this was simply to replace the Q and $Q^\dagger$ with layers of Hadamards, and applying a D^{-t} operator to another set of fanned out inputs. The output is no longer the sum of the x_i's, but it is enough for our purposes: The bits in the output are all 0 *if and only if* $\sum_{i=1}^{n} x_i - t = 0$. With the aid of a Toffoli gate, we can now determine if $\sum_{i=1}^{n} x_i = t$, and it is done in constant depth.

Thus far we can already see a quite interesting result. *If* the definition of $\mathsf{QAC}^0_{\mathrm{wf}}$ allowed an unbounded number of single-qubit gates in a circuit family (in this case, the operations $|b\rangle \mapsto \omega^b |b\rangle$, which depend on n), we could conclude from the above argument that $\mathsf{QAC}^0_{\mathrm{wf}} = \mathsf{QTC}^0_{\mathrm{wf}}$ *exactly*. However, the definition stipulates that only a finite number of such gates are allowed in any circuit family. Høyer and Špalek [14] showed how to overcome this problem. Any single-qubit gate can be approximated to arbitrary accuracy by a fixed number of gates, in constant depth. The cost is a polynomially larger circuit and polynomially small $(1/n^c)$ two-sided error. This is enough to establish the part of Theorem 3.2 regarding the equivalence of $\mathsf{QAC}^0_{\mathrm{wf}}$ and $\mathsf{QTC}^0_{\mathrm{wf}}$.

By further applications of single-qubit gates (or their approximations) and fanouts, Høyer and Špalek were able to reduce the computation of the final Toffoli gate to a measurement of a quantum register, whose outcome agrees with OR with small error. Dispensing with the Toffoli gate in this way gives the theorem as stated and concludes our proof sketch.

There are a number of immediate corollaries that follow from Theorem 3.2 when combined with some results on classical threshold circuits [23]. For example, iterated multiplication, division, and sorting of n integers can be done with polynomial-size TC^0-type circuits. Hence these operations can also be approximated in $\mathsf{QNC}^0_{\mathrm{wf}}$.

3.1 Quantum Fourier Transform

We saw the role of the quantum Fourier transform (QFT) in the construction above. The QFT is one of the most widely used unitary transformations in quantum circuits. It is one of the key components of many quantum algorithms, like Shor's [22] quantum algorithm for factoring. An efficient implementation of the QFT will improve a wide variety of quantum circuits and algorithms. Before we discuss low depth circuits for the QFT, it is interesting to compare it to its classical counterpart, the *discrete Fourier Transform* (DFT). The m-dimensional DFT maps $(a_0, \ldots, a_{m-1}) \in \mathcal{C}^m$ to $(b_0, \ldots, b_{m-1}) \in \mathcal{C}^m$ where,

$$b_x = \sum_{y=0}^{m-1} e^{(2\pi i/m)xy} a_y$$

The *fast Fourier transform* algorithm can compute the DFT in $O(m \log m)$ operations.

The m-dimensional *quantum Fourier Transform* can be seen as a unitary operation performing the DFT on the amplitudes of a $\log m$-qubit state, mapping $\sum_{x=0}^{m-1} \alpha_x |x\rangle$ to $\sum_{x=0}^{m-1} \beta_x |x\rangle$ where,

$$\beta_x = \frac{1}{\sqrt{m}} \sum_{y=0}^{m-1} e^{(2\pi i/m)xy} \alpha_y$$

The rest of this section assumes $m = 2^n$ and uses $\omega = e^{2\pi i/2^n}$. Coppersmith [6] showed how to

compute the QFT in $O(n^2)$ size and depth.[5] The bounds have been reduced further to sub-quadratic size and linear depth.

One approach to computing the QFT in constant depth can be obtained by inspecting the state of each qubit in the transformed state.

$$|\Psi_x\rangle = Q|x\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \omega^{xy}|y\rangle = \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} \bigotimes_{k=1}^{n} \left(\omega^{x2^{n-k}y_{n-k}}|y_{n-k}\rangle \right) = \frac{1}{2^{n/2}} \bigotimes_{k=1}^{n} \left(\sum_{b=0}^{1} \omega^{2^{n-k}xb}|b\rangle \right) \tag{2}$$

The "rotations" $\frac{1}{\sqrt{2}} \sum_{b=0}^{1} \omega^{2^{n-k}xb}|b\rangle = \frac{|0\rangle + e^{2\pi i x/2^k}|1\rangle}{\sqrt{2}}$, controlled by x, can easily be implemented in parallel using fanout gates as described before. This suffices to obtain the circuit for $|x\rangle|0\rangle \mapsto |x\rangle|\Psi_x\rangle$. This is followed by the relatively harder step $|x\rangle|\Psi_x\rangle \mapsto |0\rangle|\Psi_x\rangle$ to complete the QFT transformation $|x\rangle|0\rangle \mapsto |x\rangle|\Psi_x\rangle \mapsto |\Psi_x\rangle|0\rangle$. Cleve and Watrous [4] showed how to compute the second step (with high accuracy) using log-depth circuits with just CNOT gates. Høyer and Špalek further adapted Cleve and Watrous' technique to show that QFT can be approximated with small error probability in $\mathsf{QNC}^0_{\mathrm{wf}}$.

The transformation $|x\rangle|\Psi_x\rangle \mapsto |0\rangle|\Psi_x\rangle$ is computed by first estimating $|\Psi_x\rangle|0\rangle \mapsto |\Psi_x\rangle|\tilde{x}\rangle$ using *quantum Fourier phase estimation* and then XORing $|\tilde{x}\rangle$ to $|x\rangle$. If the estimate is accurate enough, then $\tilde{x} = x$, so we end up with the desired state.

The quantum Fourier phase estimation requires several copies of $|\Psi_x\rangle$, which can be obtained by using the reversible addition gate: $|x_1\rangle \dots |x_t\rangle \mapsto |x_1\rangle \dots |x_{t-1}\rangle|\sum x_i \bmod 2^n\rangle$. Addition modulo 2^n can be computed using threshold gates, which as shown before can be approximated in $\mathsf{QNC}^0_{\mathrm{wf}}$. Using the inverse of the reversible addition gate, we can approximate in constant depth $|\Psi_x\rangle|0\rangle \dots |0\rangle \mapsto |\Psi_x\rangle|\Psi_x\rangle \dots |\Psi_x\rangle$. Denoting $\frac{|0\rangle + e^{2\pi i x/2^k}|1\rangle}{\sqrt{2}}$ by $|\rho_{x/2^k}\rangle$, this gives us multiple copies of $|\rho_{x/2^k}\rangle$ for each $k = 1 \cdots n$.

The quantum Fourier phase estimation part of the circuit measures the copies of $|\rho_{x/2^k}\rangle$ to approximately determine all bits of x. Measurement of $|\rho_{x/2^k}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_{k-1}\dots x_0)})$ in the basis $\left\{ \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \cdot \frac{1}{4}}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \cdot \frac{3}{4}}|1\rangle) \right\}$ and $\left\{ \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \cdot 0}|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i \cdot \frac{2}{4}}|1\rangle) \right\}$ allows us to estimate x_k. Using $O(\log n/\varepsilon)$ copies of $|\rho_{x/2^k}\rangle$ gives us an estimate $\tilde{x}$ which is ε close to x. After computing $|x \oplus \tilde{x}\rangle|\Psi_x\rangle|\tilde{x}\rangle$, the ancillæ $|\tilde{x}\rangle$ are returned to their initial states to perform a clean computation. All the measurements and the estimation can be done in parallel and the measurements can be deferred till the end. That gives us a constant depth approximation for computing the QFT.

Theorem 3.3 *[14]* *The QFT transformation can be approximated with polynomially small error by a quantum circuit of constant depth using CNOT and a fixed family of single-qubit gates.*

4 Lower Bounds

In the last section we saw the surprising power provided by allowing fanout gates in constant depth quantum circuits. We now turn to the problem of whether, in the presence of single qubit

---

[5]Though this looks like an exponential improvement over the classical case, note that unlike the DFT, the QFT does not explicitly compute $(\beta_0, \dots)$. Intuitively, the difference between DFT versus QFT is analogous to computing a probability distribution versus sampling from the distribution [4].

gates, these gates are strictly more powerful than (unbounded) Toffoli gates and are necessary for simulating stronger classes.

We seek to fill a gap in our understanding of the relative power of Toffoli and fanout gates in quantum circuits. Roughly speaking, we know that given fanout, we can do *fanin*, where by fanin, we mean the quantum gates, for example the Toffoli gate, in which one output qubit of the gate depends on the values of (unboundedly) many inputs qubits. But can fanout do more? Are generalized Toffoli gates and fanout gates equivalent in power, up to polynomial size and constant depth; i.e., are QAC^0 and $\mathsf{QAC}^0_{\mathrm{wf}}$ equal? We believe they are not, and thus that fanout gates are strictly more powerful.

In answering this question we find it necessary to grapple with another likely limitation of real quantum computers. It is evident that they will be limited not only in their run-time duration but also by the number of qubits used in the computation, due to the difficulty in controlling the interactions of multiple qubits. It will be necessary to identify computations which use as few ancillæ as possible. Thus we consider here the number of ancillæ used by a circuit as an additional computational resource, and investigate cases where this resource is limited.

The main result of this section is that one cannot compute parity (and hence fanout) with QAC^0 circuits using a constant number of ancillæ. This is the first hard evidence that QAC^0 and $\mathsf{QAC}^0_{\mathrm{wf}}$ may be different, and that fanout may be necessary for all the upper bound results mentioned in Section 3 (it certainly is if we limit our computations to only constantly many ancillæ). The issue of the necessity of ancillæ in quantum computations is a murky one. It is generally accepted that a limited number (polynomially many relative to the number of inputs) are allowed. This seems reasonable as it allows polynomial extra space in which to carry out a computation. However, it is possible to approximate any unitary operator with a small set of universal gates without ancillæ (although one needs circuits of exponential depth and size in order to do so [17]). Furthermore, to our knowledge, no systematic investigation into the absolute necessity of ancillæ for efficient quantum computation has been done.

Our main lower bound theorem states:

Theorem 4.1 *[8] Let C be a circuit of depth d consisting of single-qubit gates and Toffoli gates, and using 0 ancillæ. Then, if $d \in o(log(n))$, C cannot compute the fanout operation.*

The theorem can be generalized to circuits with a limited number of ancillæ. Namely, in the case of a non-constant number a of ancillæ and n input qubits, we have a tradeoff between a and the required depth, that results in a non-constant lower bound for fanout when $a = n^{1-o(1)}$.

It is not hard to see, via a divide and conquer technique using CNOT gates, that one can compute parity in depth $2 \log n + 1$. We conjecture that this is optimal no matter what a is, and regardless of allowed size of Toffoli gates. However, the best lower bound on depth we can obtain is $1.44 \log n - 1$ for 0 ancillæ. With a more careful analysis, we find that circuit depth lower bound of at least $1.44 \log m - 1$ is required for any function with the property that, for any input string x, there is a set of m bits such that flipping any one of them changes $f(x)$ (The integer m is known as the **sensitivity** of the function f [18]). So more succinctly, any function of sensitivity m must have at least $1.44 \log m - 1$ depth.

The proof of the lower bound is quite long and can be found in [8]. We limit ourselves here to a brief discussion of the central ideas of the proof and its most interesting aspects.

We first consider the case of QNC^0 circuits when the quantum circuit contains only 1 and 2 qubit gates. The intuition behind the proof for this case of the main theorem seems quite obvious.

Namely, if such a circuit has depth d, then any output qubit of the circuit can depend on at most 2^d input qubits. This fact is obvious for classical circuits, by a simple connectivity argument. One might think it equally easy in the quantum case, but the picture of a quantum computation as a "classical" circuit can be deceiving. It is therefore important to verify carefully the intuitive fact that a quantum circuit must connect all the qubits on which its output depends to the qubit we will measure for the output. Furthermore, the technique we use here underscores the difficulties for the more general lower bound theorem for circuits including large Toffoli gates.
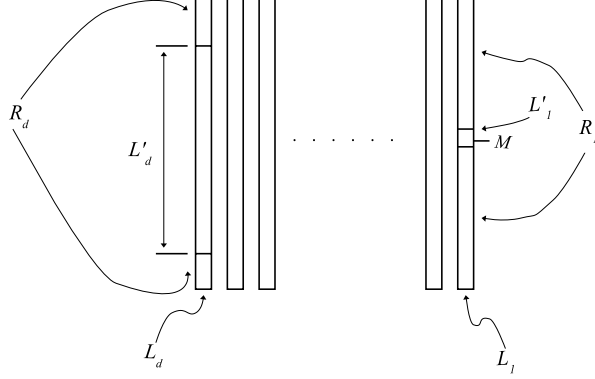

Figure 4: Decomposition of the layers of the QNC^0 circuit C.

Let $C = L_1 \cdots L_d$ consist entirely of arbitrary two-qubit gates and single-qubit gates. (The extension to arbitrary, but fixed-size gates is straightforward.) Further suppose that M is an observable on a single qubit in the last layer. Let L_1' denote the gate whose output M is measuring (Figure 4). L_1' could be a two-qubit or a single-qubit gate. In either case, $L_1 = L_1' \otimes R_1$, where R_1 is the tensor product of all the other gates in that layer, if any. If we only had this one layer, the result of the measurement M is determined by the expectation value of the operator $(L_1' \otimes R_1)^\dagger M (L_1' \otimes R_1)$. Since M commutes with R_1, the R_1's cancel, and the only gate involved is L_1. We proceed to include more layers, that is, decomposing layer i by writing $L_i = L_i' \otimes R_i$, where L_i' is a transformation that acts on some subset of the bits involving M, and R_i acts on the rest. A similar thing happens with these other layers. M remains sandwiched between some of the operators L_i', but the R_i's cancel and by induction the width of the layers L_i' that remain is at most 2^d.

From this result on connectivity of quantum circuits, we immediately obtain,

Theorem 4.2 *[8] Let C be a QNC^0 circuit on n inputs and depth d with any number of ancillæ that cleanly computes parity exactly. Then $d \geq \log n$. If C computes fanout in the same way, then $d \geq \log n - 2$.*

The theorem actually applies not only to parity but to any function whose output depends on all of its inputs. The proof technique can also be used to establish that constant depth circuit families with gates of bounded arity are not capable of simulating Toffoli gates.

We now turn to the separation of circuits with (unbounded) Toffoli gate and fanout gates. To see how to proceed, it is useful to briefly consider classical circuits with similar constraints. Suppose we have a classical circuit with NOT gates and unbounded fan-in AND and OR gates, but that we do *not* allow any fanout. Once inputs (or outputs of other gates) are used in either an AND or

an OR gate, they cannot be used again. It is obvious that if such a circuit has constant depth, it cannot compute such functions as parity. The AND and OR gates can be killed off by specifying their values on a small set of inputs, resulting in a constant function, while parity depends on all the inputs.

In the quantum case, it appears again that the only thing to do is to attempt to "kill off" the large Toffoli gates. However, the quantum case is much more subtle since we must face the fact that intermediate states are a superposition of computational basis states, and furthermore that the Toffoli gates, in combination with the single-qubit gates, may cause entanglement.
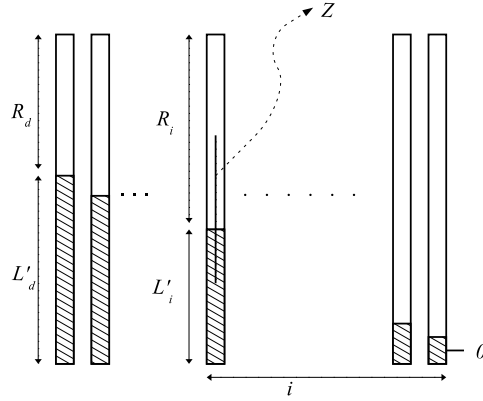

Figure 5: The sets R_i and L'_i for each layer i. A Z gate involving bits in both sets is shown.

Assume we have a circuit C composed of d levels $L_1, L_2 \cdots L_d$. The circuit C transforms the state $|\Psi\rangle$ to $L_1 \cdots L_d |\Psi\rangle$. We assume without loss of generality that each layer L_i is a tensor product of Z-gates[6] and single-qubit gates. Further assume without loss of generality that a specific bit (say, the nth bit) of C serves as the output or target bit (which eventually is supposed to agree with the output bit of a parity gate).

Our method is to work backward through he levels of C starting from the target qubit (refer to Figure 5). We fix the target bit to 0. We prove that there is a small set of bits in layer i that are involved in a quantum state $|\Psi_i\rangle$ that serves (when used as input at that layer) to keep the target bit at its fixed 0 value. Bits that are *not* involved in the construction of the state in the previous layer but that are inputs to Z-gates are set to 0 (to kill the Z-gate; this set of bits is denoted R_i in Figure 5). Bits that *are* involved (denoted L'_i in Figure 5) are allowed to propagate backwards through the layer, which can result in entanglement. The number of bits that are "committed to 0" (the size of L'_i) at any layer at most doubles as we work backwards through the layers. Thus we find that the number of bits involved in $|\Psi_i\rangle$ at the ith level is at most 2^i. For the full circuit of depth d, there thus exists a state $|\Psi_d\rangle$ involving only 2^d bits that, when applied to the circuit, will force the output to be 0. Hence if there are more than 2^d input bits to the circuit, the target bit is insensitive to some inputs and the circuit cannot compute parity exactly.

The above sketch works as well for circuits with limited numbers of ancillæ. The idea is simply to fix the ancillæ just as we fixed the target bit, and construct a state that guarantees they are 0 by working backwards. However, fixing many ancillæ (e.g., n of them) could force us to use all

---

[6]Such gates flip the sign of $|x_1, ..., x_n\rangle$ iff $\wedge_{i=1}^n x_i = 1$. They are equivalent to Toffoli gates conjugated with Hadamards at the target. They are more convenient to use here as they do not have a preferred target bit.

the inputs, and the proof breaks down. Thus, for a small number of ancillæ, we obtain a tradeoff between the depth of the circuit and the number of ancillæ it may have. Furthermore, a more careful analysis shows that bits are committed to 0 in approximately alternate layers, which leads to the factor of the golden ratio ϕ in the theorem.

Theorem 4.3 *[8] Let C be a depth d circuit of n inputs, consisting of single-qubit gates and Z-gates, and using a ancillæ, with $n > (a+1)\phi^{d+1}$. If $d < \frac{\log n}{\log \phi} - 1 \cong 1.44 \log n - 1$, then C cannot compute P, the parity gate with $n-1$ inputs and one target.*

The circuits described above required the ancillæ to be initialized to 0. Also all the circuits were doing *clean computation*, i.e., the ancillæ were 0 at the end. Clean computation ensures that circuits can be easily composed and ancillæ reused. There could be circuits which require the ancillæ to be initialized to a specific state. A clean circuit would return the ancillæ to the beginning state at the end.

There are some cases where we might want the circuit to work with any value in the ancillæ. Initializing the ancillæ to a specific state might be difficult. A standard technique used for clean computation is to copy the result to the output qubit and then apply the reverse computation to return the ancillæ to their initial state. For some circuits this technique does not work. Parker and Plenio [20] show that arbitrary initial states can be used in the quantum part of Shor's factoring circuit. Fang et al. [8] define robust computation if the circuit works with any initial state of the ancillæ but returns the ancillæ to the initial state in the end. This puts a stronger constraint on the circuit. It can be shown that any circuit needs $\log n$ depth to *robustly* compute parity using only single-qubit gates and Toffoli gate, regardless of the number of ancillæ used.

5 Open Problems

1. Is QAC^0 properly contained in $\mathsf{QAC}^0_{\mathrm{wf}}$? Stated in more detail, is it not the case that a fanout gate can be computed in constant depth with polynomially many generalized Toffoli gates, single-qubit gates, and polynomially many ancillæ?

2. A similar question applies to QTC^0 versus $\mathsf{QTC}^0_{\mathrm{wf}}$. It is unknown if threshold and fanout gates are equally powerful.

3. Can QTC^0 or $\mathsf{QTC}^0_{\mathrm{wf}}$ be simulated *exactly* by QACC circuits, using a fixed number of single-qubit gates?

4. We have seen three different classes of constant depth quantum circuits namely, QNC^0, QAC^0 and $\mathsf{QAC}^0_{\mathrm{wf}}$. ($\mathsf{QNC}^0$ is provably different from QAC^0 and $\mathsf{QAC}^0_{\mathrm{wf}}$, and the results of Section 4 give evidence that QAC^0 and $\mathsf{QAC}^0_{\mathrm{wf}}$ are different as well.) The last class, $\mathsf{QAC}^0_{\mathrm{wf}}$, is quite general and its computational power does not change, even if we add mod gates and threshold gates. Can every class of constant depth quantum circuits composed of single qubit gates and some finite sets of other gates be simulated by one of the classes QNC^0, QAC^0 or $\mathsf{QAC}^0_{\mathrm{wf}}$?

5. We sketched how a threshold gate can be approximated by fanout and single-qubit gates in constant depth with polynomially small error. Can the error be made *exponentially* small preserving constant depth?

6. The ability to compute either parity or fanout in constant depth (provided we have Hadamard and CNOT gates) is equivalent to the ability to create a "cat state," of the form $\frac{1}{\sqrt{2}}(|00\ldots0\rangle + |11\ldots1\rangle)$ (with n 1's and 0's), in constant depth [8, 15]. Assuming the answer to the question posed in (1) is affirmative, what insight does this give us into the nature of entanglement? In particular, does a useful entanglement measure emerge (in terms of the size and depth requirements for producing a state)? Alternatively, can existing entanglement measures (see, for example, [5]) be used to answer the question posed in (1)?

7. A significant roadblock to lower bounds in this area is the presence of ancillæ. The existing results give trade-offs between circuit size and the number of ancillæ. What techniques can be developed to obtain stronger trade-offs of this type either in this context or in more general settings?

8. There are further interesting questions concerning the power of ancillæ in quantum computation which have never been fully explored. For example, does there exist a function which needs more than linearly many ancillæ when computed by a constant depth quantum circuit family? More specifically, is computing parity using Toffoli and single qubit gates possible using linearly many ancillae? Many other variants of this question arise easily and have no ready answer.

6   Acknowledgments

References

[1] L. Adleman, J. DeMarrais, and M. Huang. "Quantum computability". *SIAM Journal on Computing*, 26:1524–1540, 1997.

[2] A. Barenco, C. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, and H. Weifurter. "Elementary gates for quantum computation". *Phys. Rev. A*, 52:3457–3467, 1995.

[3] J.I. Cirac and P. Zoller. "Quantum computers with cold trapped ions". *Phys. Rev. Lett.*, 74:4091–4094, 1995.

[4] R. Cleve and J. Watrous. "Fast parallel circuits for the quantum Fourier transform". *Proc. of 41st IEEE FOCS*, pages 526–536, 2000.

[5] V. Coffman, J. Kundu, and W. K. Wootters. "Distributed Entanglement". *Phys. Rev. A*, 61, 052306, 2000.

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

[7] S. Fenner. "A physics-free introduction to the quantum computation model", Computational Complexity Column. *Bulletin of EATCS*, 79:69–85, 2003.

[8] M. Fang, S. Fenner, F. Green, S. Homer, and Y. Zhang. "Quantum lower bounds for fanout". *Quantum Information and Computation*, 6(1):046–057, 2006.

[9] S. Fenner, F. Green, S. Homer, and R. Pruim. "Quantum NP is hard for PH". *Proceedings of 6th Italian Conference on theoretical Computer Science*, World Scientific, Singapore, pages 241–252, 1998.

[10] L. Fortnow and J. Rogers. "Complexity Limitations on Quantum Computation". *Proceedings of 13th IEEE Conference on Computational Complexity*, pages 202–209, 1998.

[11] M. Furst, J.B. Saxe, and M. Sipser. "Parity, circuits, and the polynomial-time hierarchy". *Math. Syst. Theory*, 17:13–27, 1984.

[12] N. Gershenfeld and I. Chuang. "Bulk spin resonance quantum computation". *Science*, 275:350–356, 1997.

[13] 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.

[14] P. Høyer and R. Špalek. "Quantum fan-out is powerful". *Theory of Computing*, 1:81–103, 2005.

[15] Cristopher Moore. "Quantum Circuits: Fanout, Parity, and Counting". In *Los Alamos Preprint archives* quant-ph/9903046, 1999.

[16] Cristopher Moore and Martin Nilsson. "Parallel Quantum Computation and Quantum Codes". In *Los Alamos Preprint archives* quant-ph/9808027, 1998.

[17] M. A. Nielsen and I. L. Chuang. "Quantum computation and quantum information". *Cambridge University Press*, 2000.

[18] N. Nisan. "CREW PRAMs and decision trees". *SIAM J. Computing*, 20:999–1007, 1991.

[19] H. Nishimura and M. Ozawa. "Computational complexity of uniform quantum circuit families and quantum Turing machines". *Theoretical Computer Science*, 1-2(276):147–181, 2002.

[20] S. Parker and M. B. Plenio. "Efficient factorization with a single pure qubit and logN Mixed qubits". *Physics Review Letters*, 85(14):3049–3052, Oct 2000.

[21] 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.

[22] P. W. Shor. "Polynomial-time algorithms for prime number factorization and discrete logarithms on a quantum computer". *SIAM J. Comp.*, 26:1484–1509, 1997.

[23] 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.

[24] R. Smolensky. "Algebraic methods in the theory of lower bounds for Boolean circuit complexity". *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 77-82, 1987.

[25] R. Špalek. "Quantum Circuits with Unbounded Fan-out". Master's Thesis, Faculty of Sciences, Vrije Universiteit, Amsterdam, 2002.

[26] H. Vollmer. "Introduction to Circuit Complexity". Springer-Verlag, 1999.

[27] A. C.-C. Yao. "Quantum circuit complexity". In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 352–361, 1993.