Programmable Networks for Quantum Algorithms

Norbert Schuch[1,2] and Jens Siewert[1]
[1]Institut für Theoretische Physik, Universität Regensburg, D-93040 Regensburg, Germany
[2]Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Strasse 1, D-85748 Garching, Germany
(Received 26 February 2003; published 7 July 2003)

The implementation of a quantum computer requires the realization of a large number of N-qubit unitary operations which represent the possible oracles or which are part of the quantum algorithm. Until now there have been no standard ways to uniformly generate whole classes of N-qubit gates. We develop a method to generate arbitrary controlled phase-shift operations with a *single* network of one-qubit and two-qubit operations. This kind of network can be adapted to various physical implementations of quantum computing and is suitable to realize the Deutsch-Jozsa algorithm as well as Grover's search algorithm.

DOI: 10.1103/PhysRevLett.91.027902 PACS numbers: 03.67.Lx, 03.67.Mn, 07.05.Bx

The experimental implementation of complex N-qubit operations (where $N \geq 3$) and the realization of complete quantum algorithms are major challenges in the field of quantum computation. Any progress in this direction proves the practical feasibility of quantum computation. Further, it provides a tool for systematic study of the physical und technological requirements for quantum computers such as parametric constraints of a given implementation, decoherence times, robustness of many-qubit entanglement, measurement efficiency, etc.

The attempts of practical implementation focus essentially on Shor's algorithm [1], Grover's database search [2], and the Deutsch-Jozsa algorithm [3,4]. Qubit-based experimental realizations of quantum algorithms have been achieved with liquid-state NMR techniques [5–9] and with trapped ions [10]. In order to practically implement (or rather simulate) a quantum computer one needs to complete the following steps: prepare the initial state, apply an N-bit unitary operation which encodes the properties of a certain function f (the so-called *oracle*), perform another sequence of operations (the quantum algorithm that extracts the properties of f), and finally measure the qubit register which contains the desired information about f.

The universality of quantum computation implies that it is possible to generate arbitrary N-qubit gates by using sequences of one-qubit and two-qubit operations only [11–13]. Barenco *et al.*, and later Cleve *et al.*, have developed methods to design networks for N-qubit controlled operations [14,15]. While these methods, in principle, are sufficient to generate any N-qubit gate required for the known quantum algorithms, it is not obvious how to apply them for a systematic and efficient *practical* realization of the operations.

A feature common to all existing implementations so far is that the sequences of one-qubit and two-qubit operations to realize the algorithm depend on the specific physical system and the number of qubits. More importantly, they even depend on the particular choice among the possible functions f. For example, all $N = 3$ implementations of the Deutsch-Jozsa algorithm [7,8,16] use some classification of the $(2^N)!/[(2^{N-1})!]^2$ balanced functions and give prescriptions how to realize the functions in each class. While for $N = 3$ there are only 70 balanced functions, this approach appears hard to extend even to $N = 4$. This situation is not satisfactory. Clearly, scalability is a requirement not only for quantum hardware [17], but also for quantum software.

In practical realizations of quantum information processing, one has to cope also with other problems: first, it is often not possible to directly perform two-qubit operations between arbitrary pairs of bits. Second, the controlled-NOT (CNOT) gate is not the genuine two-qubit gate for many proposed qubit systems. Different interaction Hamiltonians provide different types of two-qubit gates in a "natural way," i.e., gates which can be achieved with a single two-qubit operation [18–20].

In order to overcome these difficulties, we have worked out a systematic approach to design quantum networks which are suitable to perform arbitrary controlled phase-shift operations

$$U_{\vec{\theta}}: |\mathbf{x}\rangle \mapsto e^{-i\theta_{\mathbf{x}}}|\mathbf{x}\rangle \tag{1}$$

on N qubits (here $|\mathbf{x}\rangle = |x_1, \ldots, x_N\rangle$, $x_j \in \{0, 1\}$ denotes an element of the N-qubit computational basis). The most important feature of these networks is that the parameters $\theta_{\mathbf{x}}$ of the N-bit operations are determined *exclusively* by the rotation angles of single-qubit operations. Therefore, these networks may be regarded as programmable (in a technical sense): one can imagine to load network instructions into the processor with a punch tape where the instructions are encoded by the $2^N - 1$ rotation angles. This scheme resembles the structure of information processing in conventional computer processors and may be considered as a step towards a von Neumann architecture for quantum computers.

We show that this kind of network can be extended recursively to an arbitrary number of qubits assuming only nearest-neighbor coupling. As the method utilizes only one type of two-qubit gate, it can be, in principle, adapted to a large class of hardware implementations. As an example we discuss the implementation of the network with up to four Josephson charge qubits.

The elementary one-qubit gate in our networks is the phase shift (z rotation)

$$-[\phi]_Z = \begin{pmatrix} e^{-i\phi/2} & \\ & e^{i\phi/2} \end{pmatrix} .$$

Note that $-[\phi]_Z$ does not mix the states $\{|0\rangle, |1\rangle\}$ of the computational basis, but generates a phase shift whose sign depends on the qubit state. The fundamental two-qubit gate is the CNOT

$$\text{CNOT} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix} .$$

To motivate the discussion of the general method below, let us consider the simple two-bit network in Fig. 1. The easiest way to understand this network is to analyze its action on the states of the computational basis $|\mathbf{x}\rangle = |x_1, x_2\rangle$. We see that the network leaves the basis states unchanged, it merely generates a phase factor that depends on the index $\mathbf{x}$ of the basis state (see the caption of Fig. 1). Therefore, the action on a superposition of basis states is that the modulus of the amplitude for each component remains the same while the relative phases change in a well-defined way.

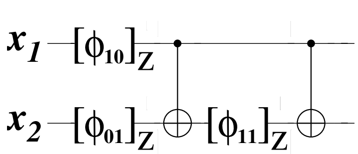


FIG. 1. Phase-shifting network for two qubits. The first two z rotations change only the phase of the input state $|x_1, x_2\rangle$: the gate $[\phi_{10}]_Z$ on the x_1 line shifts the phase of the input state by $-\phi_{10}/2$ if $x_1 = 0$ or by $+\phi_{10}/2$ if $x_1 = 1$. The gate $-[\phi_{01}]_Z$ acts correspondingly on x_2. The first CNOT gate generates the state $|x_1, x_1 \oplus x_2\rangle$ without further changing the phase. The line x_2 contains now the result of $x_1 \oplus x_2$ (where $\oplus$ is the addition modulo 2). Subsequent application of $-[\phi_{11}]_Z$ on this line results in a *conditional* shift of the phase by $-\phi_{11}/2$ if $x_1 \oplus x_2 = 0$ or by $\phi_{11}/2$ if $x_1 \oplus x_2 = 1$. The second CNOT restores the state $|x_1, x_2\rangle$ so that the input state is left unchanged apart from a phase shift that depends on the value of x_1, x_2, and $x_1 \oplus x_2$.

The total phase shift $\theta_\mathbf{x}$ due to the action of the network $|\mathbf{x}\rangle \mapsto e^{-i\theta_\mathbf{x}}|\mathbf{x}\rangle$ can be written more formally as

$$\theta_\mathbf{x} = (-1)^0 \frac{\phi_{00}}{2} + (-1)^{1\cdot x_1 \oplus 0\cdot x_2} \frac{\phi_{10}}{2} + (-1)^{0\cdot x_1 \oplus 1\cdot x_2} \frac{\phi_{01}}{2} + (-1)^{1\cdot x_1 \oplus 1\cdot x_2} \frac{\phi_{11}}{2}. \tag{2}$$

Here we have added a (physically irrelevant) global phase $\phi_{00}/2$ which applies uniformly to all basis states.

By introducing the inner product modulo two of the binary strings $\mathbf{x} = (x_1, \ldots, x_N)$ and $\mathbf{y} = (y_1, \ldots, y_N)$ ($x_j, y_j \in \{0, 1\}$) as $\mathbf{x}\cdot\mathbf{y} := x_1 \cdot y_1 \oplus \ldots \oplus x_N \cdot y_N$ we see that the phase shift $\theta_\mathbf{x}$ according to Eq. (2) is equal to

$$\theta_\mathbf{x} = \frac{1}{2}\sum_{\mathbf{y}=0}^{2^2-1} (-1)^{\mathbf{x}\cdot\mathbf{y}} \phi_\mathbf{y}. \tag{3}$$

That is, the phase shifts $\vec{\theta} = (\theta_{00}, \ldots, \theta_{11})$ are related to the one-qubit rotation angles $\vec{\phi} = (\phi_{00}, \ldots, \phi_{11})$ by an (unnormalized) two-qubit Hadamard transformation $\mathcal{H}$ [the matrix of this transformation has the entries $\mathcal{H}_{\mathbf{x},\mathbf{y}} = (-1)^{\mathbf{x}\cdot\mathbf{y}}$]. Each term in the sum of Eq. (3) represents a conditional phase shift by an angle $\phi_\mathbf{y}$ where the binary representation of the index $\mathbf{y}$ indicates the digits x_j that are part of the corresponding XOR control condition.

Thus we can summarize that the two-qubit network acts on states of the computational basis by shifting their phase. The total phase shift is obtained by subsequently applying all possible conditional phase shifts which can be derived by combining the digits x_j of the input state. The conditional phase shifts are realized by applying a single-qubit rotation to the qubit that contains the result of the corresponding control condition.

It is evident that this scheme can be generalized to an arbitrary number N of qubits: Provided that we are able to construct an N-qubit network which, on application to any state $|\mathbf{x}\rangle$ of the N-qubit computational basis, **[N1]** allows one to generate—following *classical* logics—all possible control conditions $\mathbf{x}\cdot\mathbf{y}$ from the digits of $\mathbf{x}$ (where $\mathbf{y}$ is a binary string of length N), **[N2]** applies z rotations to each condition $\mathbf{x}\cdot\mathbf{y}$, we can implement any generalized phase shift operator $U_{\vec{\theta}} := |\mathbf{x}\rangle \mapsto e^{-i\theta_\mathbf{x}}|\mathbf{x}\rangle$ by using one and the same network, merely by adjusting the angles $\vec{\phi} = (\phi_0, \ldots, \phi_{2^N-1})$ of one-qubit z rotations. By virtue of $\mathcal{H}^2 = 2^N \mathbb{1}$, the angles $\vec{\phi}$ are related to the desired phases $\vec{\theta} = (\theta_0, \ldots, \theta_{2^N-1})$ simply by an N-bit Hadamard transformation

$$\vec{\phi} = \frac{1}{2^{N-1}} \mathcal{H}\vec{\theta}. \tag{4}$$

The network can be viewed as a "black box" with $2^N - 1$ "knobs" whose settings determine the gate $U_{\vec{\theta}}$ realized by the black box. By choosing $\vec{\phi}$, any phase configuration $\vec{\theta}$ can be programmed into the network.

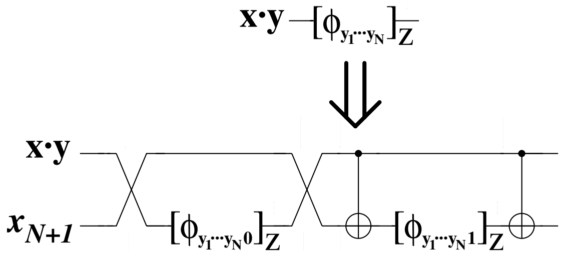
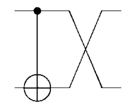


FIG. 2. Recursive extension of the network. By a SWAP with the $N + 1$st qubit the phase shift $\phi_{\mathbf{y}_1\ldots\mathbf{y}_N} \rightarrow \phi_{\mathbf{y}_1\ldots\mathbf{y}_N 0}$ (controlled by the original condition) appears now on the $N + 1$st qubit. A second SWAP and a subsequent CNOT from the Nth to the $N +$ 1st qubit give the original condition XORed with x_{N+1}—this condition controls the second z rotation. We mention the identity .

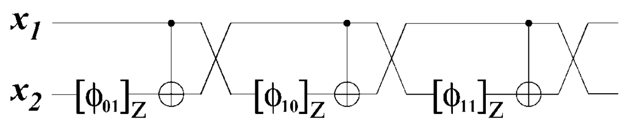
The remaining task is to show that it is indeed possible to construct such programmable networks for an arbitrary number of qubits. Since for many physical implementations coupling between arbitrary qubits is not available or falls off with increasing distance or qubit number, we will assume only nearest-neighbor coupling between the qubits [21]. Further, we will show that efficient network design is possible even if CNOT is not the natural two-qubit gate for a given implementation [18–20].

For the recursive method it is convenient that all controlled phase shifts appear on the last (i.e., the Nth) qubit. Then the network for $N + 1$ qubits can be obtained from the network for N bits simply by (i) adding the $N + 1$st line with a z rotation (i.e., a phase shift controlled by x_{N+1}), and (ii) by replacing each conditional z rotation on the Nth qubit in the N-bit network by two rotations of the $N + 1$st qubit, as shown in Fig. 2. By following these rules, we get z rotations for all possible XOR conditions of the $N + 1$ qubits. The conditions for the rotations characterized by the indices $\mathbf{y} = (y_1, \ldots, y_{N+1})$ appear in their natural order if read as a binary code.

To illustrate this, we explain the method for up to three qubits with an XY interaction as, e.g., for Josephson charge qubits coupled by SQUID loops [16,22]. The construction is analogous for other two-bit operations used as the basic element. The XY interaction leads to a particularly compact design of the circuit. This is because the natural gate for this interaction is again a classical gate, namely, the product of CNOT and SWAP denoted by CNS [20]

CNS ≡  .

The "one-qubit network" is just a single z rotation x_1 $[\phi_1]_Z$. For two qubits we modify the network in Fig. 1 such that all z rotations appear on the second line



.

From here, the derivation of the three-qubit network is straightforward. The result is shown in Fig. 3.

We emphasize that the central issue is the concept of these networks expressed in the requirements [N1], [N2]. Since [N1] refers only to classical logics, optimized circuits can be found, e.g., by an exhaustive search (on a classical computer). An example is the four-qubit network in Fig. 4.

By realizing such a network with a physical system (and supplementing it with one-qubit operations such as, e.g., the Hadamard gate), both the Deutsch-Jozsa algorithm and Grover's search algorithm can be implemented. The refined version of the Deutsch-Jozsa algorithm [15,23] requires the gate $U_f := |\mathbf{x}\rangle \mapsto e^{-i\pi f(\mathbf{x})}|\mathbf{x}\rangle$ where f is a constant or balanced Boolean function. Analogously, for Grover's algorithm the gate U_g has to be implemented with a Boolean function g: $g(\mathbf{x}^*) = 1$ for one particular $\mathbf{x}^*$ and $g(\mathbf{x} \neq \mathbf{x}^*) = 0$. For the generalized Grover algorithm [24], several items may be marked. Hence the required phase shifts $\theta_{\mathbf{x}}$ are given by $\pi f(\mathbf{x})$ or $\pi g(\mathbf{x})$, respectively. The corresponding one-bit rotation angles $\vec{\phi}$ are readily obtained by Hadamard-transforming $\vec{\theta}$ according to Eq. (4).

We mention that there are other interesting applications for these programmable networks. For example, by Hadamard-transforming the Nth qubit a network is obtained which is capable of generating a generalized controlled-NOT operation on the Nth qubit with an arbitrary control condition composed from the other $N - 1$ qubits. Special cases are the Toffoli gate (with an arbitrary number of control bits [14]) and the CARRY operation

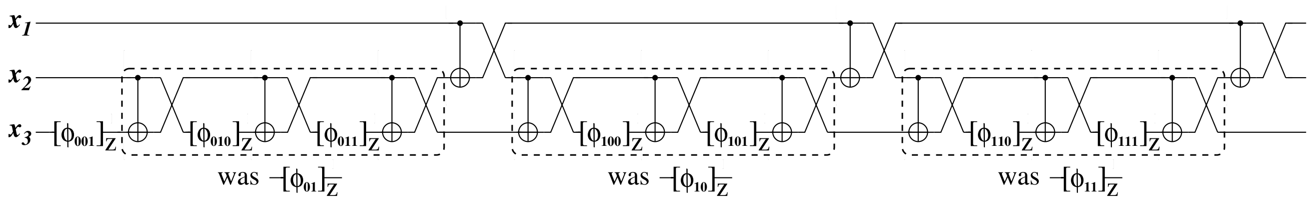


FIG. 3. The three-bit network for a system with nearest-neighbor XY coupling. The dashed lines indicate the parts of the two-bit network which have been replaced according to rule (ii).

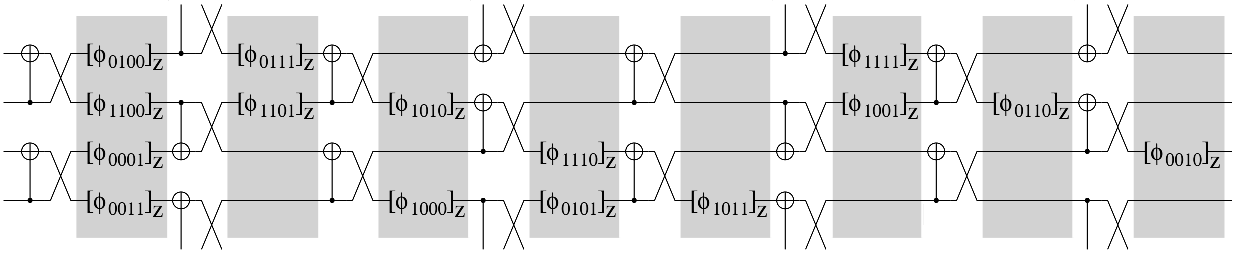


FIG. 4. Optimized four-bit network for qubits with nearest-neighbor XY coupling. With this network (plus the Hadamard gate for each qubit), both the Deutsch-Jozsa algorithm and database search can be realized, e.g., with Josephson charge qubits. We have assumed qubits coupled in a chain with periodic boundary conditions. A particularly interesting feature of this network is that nearest-neighbor coupling allows for parallel execution of operations. Note that the z rotations and the two-qubit operations appear in well-separated blocks. The optimization criterion here was to minimize the number of two-qubit blocks.

that appears in the network for the quantum adder used, e.g., in Shor's factoring algorithm [25].

Thus, the new type of network presented here offers efficient practical solutions for a wide range of computational tasks; this is comprehensible for modest qubit numbers (on the order of 10) just by comparing with existing solutions. On the other hand, the limit of large N requires more careful investigation. Although the parallel execution of operations (as shown in Fig. 4) can reduce, in principle, the complexity to $O(2^N/N)$ it is not clear whether further substantial improvement is possible. There is exponential scaling also for another resource. According to Eq. (4) the accuracy required for the one-qubit z rotations will scale $\sim 2^{-N}$. It remains a subject for future work whether this limit can be softened, e.g., for certain subclasses of gates $U_{\vec{\theta}}$.

The authors would like to thank R. Fazio, E. Kashefi, Yu. Makhlin, K. Richter, and V. Vedral for stimulating discussions and comments.

[1] P.W. Shor, in *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, edited by S. Goldwasser (IEEE Computer Society, Los Alamitos, CA, 1994), p. 124.
[2] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[3] D. Deutsch, Proc. R. Soc. London A **400**, 97 (1985).
[4] D. Deutsch and R. Jozsa, Proc. R. Soc. London A **439**, 553 (1992).
[5] J. A. Jones, M. Mosca, and R. H. Hansen, Nature (London) **393**, 344 (1998).
[6] L. M. Vandersypen, M. Steffen, M. H. Sherwood, C. S. Yannoni, G. Breyta, and I. L. Chuang, Appl. Phys. Lett. **76**, 646 (2000).
[7] D. Collins, K.W. Kim, W. C. Holton, H. Sierzputowska-Gracz, and E. O. Stejskal, Phys. Rev. A **62**, 022304 (2000).
[8] J. Kim, J.-S. Lee, S. Lee, and C. Cheong, Phys. Rev. A **62**, 022312 (2000).
[9] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Nature (London) **414**, 883 (2001).
[10] S. Gulde, M. Riebe, G. P. T. Lancaster, C. Becher, J. Eschner, H. Häffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt, Nature (London) **421**, 48 (2003).
[11] D. Deutsch, A. Barenco, and A. Ekert, Proc. R. Soc. London A **449**, 669 (1995).
[12] S. Lloyd, Phys. Rev. Lett. **75**, 346 (1995).
[13] M. J. Bremner, C. M. Dawson, J. L. Dodd, A. Gilchrist, A.W. Harrow, D. Mortimer, M. A. Nielsen, and T. J. Osborne, Phys. Rev. Lett. **89**, 247902 (2002).
[14] A. Barenco, C. Bennett, R. Cleve, D. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).
[15] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. London A **454**, 339 (1998).
[16] J. Siewert and R. Fazio, Phys. Rev. Lett. **87**, 257905 (2001).
[17] D. P. DiVincenzo, Fortschr. Phys. **48**, 771 (2000).
[18] Y. Makhlin, quant-ph/0002045.
[19] G. Vidal, K. Hammerer, and J. I. Cirac, Phys. Rev. Lett. **88**, 237902 (2002).
[20] N. Schuch and J. Siewert, Phys. Rev. A **67**, 032301 (2003).
[21] If we assume that a hardware setup is available which allows one to perform the CNOT operation between arbitrary qubits, the networks can be constructed efficiently by ordering the $\phi_{\mathbf{y}}$'s such that the $\mathbf{y}$'s form a Gray sequence (see Ref. [14]).
[22] P. Echternach, C. P. Williams, S. C. Dultz, P. Delsing, S. Braunstein, and J. P. Dowling, Quantum Inf. Comput. **1**, 143 (2001).
[23] D. Collins, K. Kim, and W. Holton, Phys. Rev. A **58**, R1633 (1998).
[24] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Fortschr. Phys. **46**, 493 (1998).
[25] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).