Probabilistic implementation of universal quantum processors

Mark Hillery,[1] Vladimír Bužek,[2,3] and Mário Ziman[2]
[1]Department of Physics and Astronomy, Hunter College of CUNY, 695 Park Avenue, New York, New York 10021
[2]Research Center for Quantum Information, Slovak Academy of Sciences, Dúbravská cesta 9, 842 28 Bratislava, Slovakia
[3]Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

(Received 31 May 2001; revised manuscript received 4 September 2001; published 3 January 2002)

We present a probabilistic quantum processor for qudits on a single qudit of dimension N. The processor itself is represented by a fixed array of gates. The input of the processor consists of two registers. In the program register the set of instructions (program) is encoded. This program is applied to the data register. The processor can perform any operation on a single qudit of dimension N with a certain probability. For a general unitary operation, the probability is $1/N^2$, but for more restricted sets of operators the probability can be higher. In fact, this probability can be independent of the dimension of the qudit Hilbert space of the qudit under some conditions.

DOI: 10.1103/PhysRevA.65.022301 PACS number(s): 03.67.−a, 03.67.Lx

I. INTRODUCTION

Schematically we can represent a classical computer as a device with a processor, which is a fixed piece of hardware, that performs operations on a *data* register according to a program encoded initially in the *program* register. The action of the processor is fully determined by the program. The processor is universal if we can realize any operation on the data by entering the appropriate program into the program register.

In this paper we shall examine a quantum version of this picture. Specifically, in close analogy with recent papers by Nielsen and Chuang [1] and Vidal and Cirac [2], we will study how a quantum program initially put into a program register can cause a particular operation to be applied to a data register initially prepared in an unknown state. We shall first consider the case in which the data consists of a single qubit, and the program of two qubits. We shall then examine higher-dimensional systems.

Nielsen and Chuang [1] originally formulated the problem in terms of a programmable array of quantum gates, which can be described as a fixed unitary operator, P_{dp}, that acts on both the program and the data. The initial state, $|\Xi_U\rangle_p$, of the program register stores information about the one-qubit unitary transformation U that is going to be performed on a single-qubit data register initially prepared in a state $|\psi\rangle_d$. The total dynamics of the programmable quantum gate array is then given by

$$P_{dp}[|\psi\rangle_d \otimes |\Xi_U\rangle_p] = (U|\psi\rangle_d) \otimes |\tilde{\Xi}_U\rangle_p, \tag{1}$$

where only pure data states were considered. The program register at the output of the gate is in the state $|\tilde{\Xi}_U\rangle_p$, which was shown to be independent of the input data state $|\psi\rangle_d$.

Nielsen and Chuang proved that any two inequivalent operations U and V require orthogonal program states, i.e., $\langle\Xi_U|\Xi_V\rangle = 0$. Thus, in order to perfectly implement a set of inequivalent operations, $\{U_j | j \in J\}$, the state space for the program register must contain the orthonormal set of program states, $\{|\Xi_{U_j}\rangle | j \in J\}$. This means that the dimension of the program register must be at least as great as the number of unitary operators that we want to perform. Since the set of unitary operations is infinite, the result of Nielsen and Chuang implies that no universal gate array can be constructed using finite resources, that is, with a finite-dimensional program register. They did show, however, that if the gate array is probabilistic, a universal gate array is possible. A probabilistic array is one that requires a measurement to be made at the output of the program register, and the output of the data register is only accepted if a particular result, or set of results, is obtained. This will happen with a probability, which is less than 1.

Because a finite gate array can only implement a finite set of unitary operations with certainty, we are forced to consider probabilistic gate arrays if we want to be able to exactly perform any one of an infinite number of unitary operations. For example, we might want to have a gate array, let us call it G, that will perform any SU(2) operation on an input qubit, but exactly which one we do want to perform will depend on the results of previous computations, which we do not know when the array is constructed. Because our processor is a quantum one, when faced with deciding which operation we want G to perform, we will have two choices. We can measure the quantum state produced by the previous stage of the calculation, and then use the result of the measurement to set some external parameters in G, with the parameter setting determining that operation G will perform. The other possibility is for G to be a programmable probabilistic gate array, and to have the previous computation produce a program state that determines, which operation G will perform. However, the measurement strategy suffers from a serious problem; if our overall system is finite, the measurement can only have a finite number of outcomes, which means that it can implement only a finite number of operations. Therefore, if we want to be able to condition computations on the results of previous ones, probabilistic gate arrays need to be considered. Because we know when these arrays succeed and when they fail, we only continue the calculation when they succeed. If they fail, we repeat the previous part of the calculation to produce another program state and input qubit, and try again. This type of the probabilistic approach is widely applied in the field of quantum information processing. To mention just a few examples: probabilistic quantum teleportation [3], probabilistic state discrimination [4], probabilistic quantum cloning [5], and probabilistic quantum memories [6], etc. Obviously, one of the main tasks in the probabilistic quantum information processing is to achieve the maximum possible probability of the success.

Vidal and Cirac [2] have recently presented a probabilistic programmable quantum gate array with a finite-program register, which can realize a one-parameter family of operations, where the parameter is continuous, with arbitrarily high probability. The higher the probability of success, the greater the dimensionality of the program register, but the number of transformations that can be realized is infinite. They have also considered *approximate* programmable quantum gate arrays, which perform an operation E_U very similar to the desired U, that is $F(E_U,U)\geqslant 1-\epsilon$ for some transformation fidelity F.

Another aspect of the encoding of quantum operations in the states of program registers has been discussed by Huelga and co-workers [7]. In this paper the implementation of an arbitrary unitary operation U upon a distant quantum system has been considered. This so-called teleportation of unitary operations has been formally represented as a completely positive, linear, trace preserving map on the set of density operators of the program and data registers

$$\mathcal{T}[|\xi\rangle_{ab}\otimes|\Xi_U\rangle_p\otimes|\psi\rangle_d]=|\tilde{\xi}_U\rangle_{ap}\otimes(U|\psi\rangle_d). \tag{2}$$

Here $|\xi\rangle_{ab}$ represents a specific entangled state that is shared by two parties, Alice and Bob, who want to teleport the unitary operation U from Alice to Bob. Huelga *et al.* [7] have investigated protocols that achieve the teleportation of U using local operations, classical communication, and shared entanglement.

In the present paper we will address the problem of implementing an operation U, encoded in the state of a program register $|\Xi_U\rangle_p$, on the data state $|\psi\rangle_d$. The gate arrays we present are probabilistic; the program register must be measured at the end of the procedure. In Sec. II we present a simple example of how to apply an arbitrary operation to a single qubit initially prepared in a state $|\psi\rangle$. The gate array consists of four controlled-NOT (C-NOT) gates, and can implement four programs perfectly. These programs cause the one of the operations $\mathbb{1}$, σ_x, $-i\sigma_y$, or σ_z to be performed on the data qubit. Here $\mathbb{1}$ is the identity and σ_j, where $j=x,y,z$ is a Pauli matrix. By choosing programs that are linear combinations of the four basic ones, it is possible to probabilistically perform any linear operation on the data qubit. In Sec. III we generalize the idea to an arbitrary dimensional quantum system, a qudit.

II. OPERATIONS ON QUBITS

We wish to construct a device that will do the following. The input consists of a qubit, $|\psi\rangle_d$, and a second state, $|\Xi_U\rangle_p$, which may be a multiqubit state, that acts as a program. The output of the device will be a state $U|\psi\rangle_d$, where U is an operation that is specified by $|\Xi_U\rangle_p$. In order to make this a little less abstract, we first consider an example. Let $|\phi\rangle$ and $|\phi_\perp\rangle$ be two orthogonal qubit states, and suppose that we want to perform the operation

$$A_z=|\phi_\perp\rangle\langle\phi_\perp|-|\phi\rangle\langle\phi|=\mathbb{1}-2|\phi\rangle\langle\phi|, \tag{3}$$

on $|\psi\rangle_d$. The action of this operator is analogous to that of σ_z in the basis $\{|0\rangle,|1\rangle\}$, except that it acts in the basis $\{|\phi_\perp\rangle,|\phi\rangle\}$. That is, σ_z does nothing to $|0\rangle$ and multiplies $|1\rangle$ by -1, while A_z does nothing to $|\phi_\perp\rangle$ and multiplies $|\phi\rangle$ by -1. Can we find a network and a program vector to implement this operation on $|\psi\rangle_d$?

We can, in fact, do this by using the network for a quantum information distributor (QID) as introduced in Ref. [8] (this is a modification of the quantum cloning transformation [9,10]). In this network the program register is represented by a two-qubit state $|\Xi_A\rangle_p$. Before we present the network for the programmable gate array, we shall introduce notation for its components. A controlled-NOT gate D_{jk} acting on qubits j and k performs the transformation,

$$D_{jk}|m\rangle_j|n\rangle_k=|m\rangle_j|m\oplus n\rangle_k, \tag{4}$$

where j is the control bit, k is the target bit, and m and n are either 0 or 1. The addition is modulo 2. The QID network consists of four controlled-NOT gates, and acts on three qubits (a single data qubit denoted by a subscript 1 and two program qubits denoted by subscripts 2 and 3, respectively). Its action is given by the operator $P_{123}=D_{31}D_{21}D_{13}D_{12}$. As our first task, we shall determine how this network acts on input states where qubit 1 is in the state $|\psi\rangle$, and qubits 2 and 3 are in Bell basis states. The Bell basis states are defined by

$$|\Phi_+\rangle=\frac{1}{\sqrt{2}}(|01\rangle+|10\rangle)\equiv|\Xi_{01}\rangle,$$

$$|\Phi_-\rangle=\frac{1}{\sqrt{2}}(|01\rangle-|10\rangle)\equiv|\Xi_{11}\rangle,$$

$$|\Psi_+\rangle=\frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)\equiv|\Xi_{00}\rangle,$$

$$|\Psi_-\rangle=\frac{1}{\sqrt{2}}(|00\rangle-|11\rangle)\equiv|\Xi_{10}\rangle. \tag{5}$$

We find that

$$P_{123}|\psi\rangle_1|\Phi_+\rangle_{23}=(\sigma_x|\psi\rangle_1)|\Phi_+\rangle,$$

$$P_{123}|\psi\rangle_1|\Phi_-\rangle_{23}=(-i\sigma_y|\psi\rangle_1)|\Phi_-\rangle,$$

$$P_{123}|\psi\rangle_1|\Psi_+\rangle_{23}=|\psi\rangle_1|\Psi_+\rangle,$$

$$P_{123}|\psi\rangle_1|\Psi_-\rangle_{23}=(\sigma_z|\psi\rangle_1)|\Psi_-\rangle. \tag{6}$$

Any operation on qubits can be expanded in terms of Pauli matrixes and the identity. The above equations mean that the Bell basis vectors are ‘‘programs’’ for a complete set of operations. In order to see how to make use of this, let us expand our proposed operation in terms of this complete set. Expressing $|\phi\rangle$ as $|\phi\rangle=\mu|0\rangle+\nu|1\rangle$, we have that

$$\begin{aligned}A_z=\mathbb{1}-2|\phi\rangle\langle\phi|&=\begin{pmatrix}|\nu|^2-|\mu|^2 & -2\mu\nu^* \\ -2\mu^*\nu & |\mu|^2-|\nu|^2\end{pmatrix},\\ &=-(\mu\nu^*+\mu^*\nu)\sigma_x+(\mu\nu^*-\mu^*\nu)(-i\sigma_y)\\ &\quad+(|\nu|^2-|\mu|^2)\sigma_z.\end{aligned} \tag{7}$$

We can now apply the operation A to $|\psi\rangle$ by sending in the ‘‘program’’ vector

$$\begin{aligned}|\Xi_A\rangle_{23}=&-(\mu\nu^*+\mu^*\nu)|\Phi_+\rangle_{23}+(\mu\nu^*-\mu^*\nu)|\Phi_-\rangle_{23}\\ &+(|\nu|^2-|\mu|^2)|\Psi_-\rangle_{23},\end{aligned} \tag{8}$$

and measuring the program outputs in order to determine if they are in the state $(|\Phi_+\rangle+|\Phi_-\rangle+|\Psi_-\rangle)/\sqrt{3}$. If they are, our operation has been accomplished. Note that the measurement is independent of the vector $|\phi\rangle$ so that no knowledge of this vector is necessary to make the measurement and to determine whether the procedure has been successful. As we see, the probability of success is 1/3 for the implementation of the operation A_z that is parameterized in general by two continuous parameters (i.e., the state $|\phi\rangle$).

Let us examine the program vector more carefully. If we define the unitary operation, U_{init}, by

$$\begin{aligned}U_{init}|00\rangle=-|10\rangle,\quad U_{init}|10\rangle=-|11\rangle,\\ U_{init}|11\rangle=|01\rangle,\quad U_{init}|01\rangle=|00\rangle,\end{aligned} \tag{9}$$

we have that

$$|\Xi_A\rangle_{12}=U_{init}\frac{1}{\sqrt{2}}(|\phi\rangle|\phi_\perp\rangle+|\phi_\perp\rangle|\phi\rangle). \tag{10}$$

Finally, we can summarize our procedure. The steps are (1) start with the state $1/\sqrt{2}(|\phi\rangle|\phi_\perp\rangle+|\phi_\perp\rangle|\phi\rangle)$; (2) apply U_{init}; (3) send the resulting state into the control ports (inputs 2 and 3) and $|\psi\rangle$ into port 1; (4) measure $(|\Phi_+\rangle+|\Phi_-\rangle+|\Psi_-\rangle)/\sqrt{3}$ at the output of the control ports; (5) if the result is yes, then the output of port 1 is $(\mathbb{1}-2|\phi\rangle\langle\phi|)|\psi\rangle$.

Before proceeding to a more general consideration of this network, let us make an observation. Suppose that we carry out the same procedure, but instead of starting with the program vector $(|\phi\rangle|\phi_\perp\rangle+|\phi_\perp\rangle|\phi\rangle)/\sqrt{2}$, we start instead with the program vector $(|\phi\rangle|\phi\rangle-|\phi_\perp\rangle|\phi_\perp\rangle)/\sqrt{2}$. At the end of the procedure the output of the data register is $A_x|\psi\rangle$, where

$$A_x=|\phi\rangle\langle\phi_\perp|+|\phi_\perp\rangle\langle\phi|. \tag{11}$$

The operation A_x interchanges $|\phi\rangle$ and $|\phi_\perp\rangle$. Its action is analogous to that of σ_x, which interchanges the vectors $|0\rangle$ and $|1\rangle$. The probability of success for this procedure is also 1/3.

We now need to determine whether there is a program for any operator that could act on $|\psi\rangle$. The operator need not be unitary; it could be a result of coupling $|\psi\rangle$ to an ancilla, evolving the coupled system (a unitary process), and then measuring the ancilla. Therefore, if A is now any linear operator acting on a two-dimensional quantum system, the transformations in which we are interested are given by

$$|\psi\rangle\rightarrow\frac{1}{\|A\psi\|}A|\psi\rangle. \tag{12}$$

Let us denote the operators, which can be implemented by Bell state programs, by $S_{00}=\mathbb{1}$, $S_{01}=\sigma_x$, $S_{10}=\sigma_z$, and $S_{11}=-i\sigma_y$. Any 2×2 matrix can be expanded in terms of these operators, so that we have

$$A=\sum_{j,k=0}^{1}\tilde{a}_{jk}S_{jk}. \tag{13}$$

We now define $a_{jk}=\tilde{a}_{jk}/\sqrt{\eta}$, where

$$\eta=\sum_{j,k=0}^{1}|\tilde{a}_{jk}|^2, \tag{14}$$

so that

$$1=\sum_{j,k=0}^{1}|a_{jk}|^2. \tag{15}$$

Now let us go back to our network and consider the program vector given by

$$|\Xi_A\rangle=\sum_{j,k=0}^{1}a_{jk}|\Xi_{jk}\rangle, \tag{16}$$

and at the output of the program register we shall measure the projection operator corresponding to the vector $(1/2)\Sigma_{j,k=0}^{1}|\Xi_{jk}\rangle$. If the measurement is successful, the state of the data register is, up to normalization, given by

$$|\psi\rangle\rightarrow\left(\sum_{j,k=0}^{1}a_{jk}S_{jk}\right)|\psi\rangle. \tag{17}$$

After this state is normalized, it is just $(1/\|A\psi\|)A|\psi\rangle$. This means that for any transformation of the type given in Eq. (12), we can find a program for our network that will carry it out.

III. GENERALIZATION TO QUDITS

In order to extend the network presented in the preceding section to higher dimensions, we must first introduce a generalization of the two-qubit C-NOT gate [8] (see also Ref. [11]). As we noted previously, it is possible to express the action of a C-NOT gate as a two-qubit operator of the form

$$D_{ab}=\sum_{k,m=0}^{1}|k\rangle_a\langle k|\otimes|m\oplus k\rangle_b\langle m|. \tag{18}$$

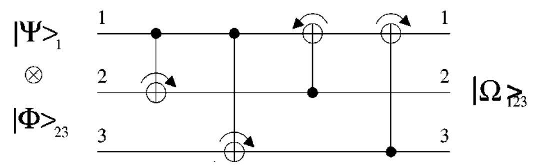

FIG. 1. A logic network for the universal quantum processor as given by the unitary transformation (24). The action of the controlled shift operator D_{jk} is represented as follows. The control qudit is represented by • while the target qudit is represented by $\oplus$ with the right arrow. The action of the operator $D_{jk}^{\dagger}$ is represented by left arrow.

In principle one can also introduce an operator $D_{ab}^{\dagger}$ defined as

$$D_{ab}^{\dagger}=\sum_{k,m=0}^{1}|k\rangle_a\langle k|\otimes|m\ominus k\rangle_b\langle m|. \tag{19}$$

In the case of qubits these two operators are equal, but this will not be the case when we generalize the operator to Hilbert spaces whose dimension is larger than 2 [8,11]. In particular, we can generalize the operator D for dimension $N>2$ by defining

$$D_{ab}=\sum_{k,m=0}^{N-1}|k\rangle_a\langle k|\otimes|(m+k)\mathrm{mod}\,N\rangle_b\langle m|, \tag{20}$$

which implies that

$$D_{ab}^{\dagger}=\sum_{k,m=0}^{N-1}|k\rangle_a\langle k|\otimes|(m-k)\mathrm{mod}\,N\rangle_b\langle m|. \tag{21}$$

From this definition it follows that the operator D_{ab} acts on the basis vectors as

$$D_{ab}|k\rangle|m\rangle=|k\rangle|(k+m)\mathrm{mod}\,N\rangle, \tag{22}$$

which means that this operator has the same action as the conditional adder and can be performed with the help of the simple quantum network discussed in [12]. Now we see that for $N>2$ the two operators D and $D^{\dagger}$ do differ; they describe conditional shifts in opposite directions. Therefore, the generalizations of the C-NOT operator to higher dimensions are just *conditional shifts*.

In analogy with the quantum computational network discussed in the preceding section, we assume the network for the probabilistic universal quantum processor to be

$$P_{123}=D_{31}D_{21}^{\dagger}D_{13}D_{12}. \tag{23}$$

The data register consists of system 1 and the program register of systems 2 and 3. The state $|\Xi_U\rangle_{23}$ acts as the "software" for which the operation to be implemented on the qudit data state $|\Psi\rangle_1$. The output state of the three-qudit system, after the four controlled shifts are applied, reads

$$|\Omega\rangle_{123}=D_{31}D_{21}^{\dagger}D_{13}D_{12}|\Psi\rangle_1|\Xi_U\rangle_{23}. \tag{24}$$

A graphical representation of the logical network (24) with the conditional shift gates D_{ab} is shown in Fig. 1.

The sequence of four operators acting on the basis vectors $|n\rangle_1|m\rangle_2|k\rangle_3$ gives

$$\begin{aligned}D_{31}D_{21}^{\dagger}D_{13}D_{12}|n\rangle_1|m\rangle_2|k\rangle_3\\ &=|(n-m+k)\mathrm{mod}\,N\rangle_1|(m+n)\mathrm{mod}\,N\rangle_2\\ &\quad\times|(k+n)\mathrm{mod}\,N\rangle_3.\end{aligned} \tag{25}$$

We now turn to the fundamental program states. A basis consisting of maximally entangled two-particle states (the analogue of the Bell basis for spin-$\frac{1}{2}$ particles) is given by [13]

$$|\Xi_{mn}\rangle=\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\exp\left(i\frac{2\pi}{N}mk\right)|k\rangle|(k-n)\mathrm{mod}\,N\rangle, \tag{26}$$

where $m,n=0,\ldots,N-1$. If $|\Xi_{mn}\rangle_p$ is the initial state of the program register, and $|\Psi\rangle=\Sigma_j\alpha_j|j\rangle_d$ (here, as usual, $\Sigma_j|\alpha_j|^2=1$) is the initial state of the data register, it then follows that

$$\begin{aligned}P_{123}|\Psi\rangle_1|\Xi_{mn}\rangle_{23}\\ &=\sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\left(\frac{2\pi ikm}{N}\right)P_{123}|j\rangle|k\rangle|k-n\rangle\\ &=\sum_{jk}\frac{\alpha_j}{\sqrt{N}}\exp\left(\frac{2\pi ikm}{N}\right)|j-n\rangle|k+j\rangle|k+j-n\rangle\\ &=\sum_{jk}\alpha_j\exp\left(\frac{-2\pi ijm}{N}\right)|j-n\rangle|\Xi_{mn}\rangle\\ &=(U^{(mn)}|\Psi\rangle)|\Xi_{mn}\rangle,\end{aligned} \tag{27}$$

where we have introduced the notation

$$U^{(mn)}=\sum_{s=0}^{N-1}\exp\left(\frac{-2i\pi sm}{N}\right)|s-n\rangle\langle s|. \tag{28}$$

This result is similar to the one we found in the case of a single qubit. We shall now examine, which transformations we can perform on the state in the data register by using a program consisting of a linear combination of the vectors $|\Xi_{mn}\rangle$ followed by the action of the processor P_{123} and a subsequent measurement of the program register.

The operators $U^{(mn)}$ satisfy the orthogonality relation

$$\mathrm{Tr}[(U^{(m'n')})^{\dagger}U^{(mn)}]=N\delta_{mm'}\delta_{nn'}. \tag{29}$$

The space of linear operators $\mathcal{T}(\mathcal{H})$ defined on some Hilbert space $\mathcal{H}$ with the scalar product given by Eq. (29) we know as *Hilbert-Schmidt space*. Thus the unitary operators $U^{(mn)}$ form an orthogonal basis in it and any operator $A\in\mathcal{T}(\mathcal{H})$ can be expressed in terms of them:

$$A=\sum_{m,n=0}^{N-1} q_{mn}U^{(mn)}. \tag{30}$$

The orthogonality relation allows us to find the expansion coefficients in terms of the operators

$$q_{mn}=\frac{1}{N}\mathrm{Tr}[(U^{(mn)})^{\dagger}A]. \tag{31}$$

Equations (29) and (30) imply that

$$\sum_{m,n=0}^{N-1} |q_{mn}|^2=\frac{1}{N}\mathrm{Tr}(A^{\dagger}A). \tag{32}$$

Therefore, the program vector that implements the operator A is given by

$$|v_A\rangle_{23}=\left[\frac{N}{\mathrm{Tr}(A^{\dagger}A)}\right]^{1/2}\sum_{m,n=0}^{N-1} q_{mn}|\Xi_{mn}\rangle_{23}. \tag{33}$$

Application of the processor to the input state $|\Psi\rangle_1|v_A\rangle_{23}$ yields the output state

$$|\Omega\rangle_{123}=\sum_{mn} q_{mn}U^{(mn)}|\Psi\rangle_1\otimes|\Xi_{mn}\rangle_{23}. \tag{34}$$

To obtain the final result we perform a projective measurement of the program register onto vector $|M\rangle_{23}$

$$|M\rangle=\frac{1}{N}\sum_{m,n=0}^{N-1}|\Xi_{mn}\rangle. \tag{35}$$

If the outcome of the measurement is positive, then we get the required transformation A acting on an unknown, arbitrary input state $|\Psi\rangle_1$.

Let us consider an example. Suppose we choose for A the unitary operator $\mathbb{1}-2|\phi\rangle\langle\phi|$, where the normalized state $|\phi\rangle$ can be expressed as

$$|\phi\rangle=\sum_{k=0}^{N-1}\beta_k|k\rangle. \tag{36}$$

The expansion coefficients for this operation are given by

$$q_{mn}=\delta_{m0}\delta_{n0}-\frac{2}{N}\sum_{k=0}^{N-1}e^{2\pi ikm/N}\beta_k^*\beta_{k-n}, \tag{37}$$

and the program vector for this operation is

$$|\Phi\rangle_{23}=|\Xi_{00}\rangle_{23}-\frac{2}{\sqrt{N}}\sum_{k,n=0}^{N-1}\beta_{-k}^*\beta_{-(k+n)}|k\rangle_2|k-n\rangle_3. \tag{38}$$

The program vector can be obtained from a state more closely related to $|\phi\rangle$ if we introduce a new unitary operator and a ‘‘complex conjugate’’ vector. Define the operator W by

$$W|k\rangle=|-k\rangle, \tag{39}$$

and the vector $|\phi^*\rangle$ by

$$|\phi^*\rangle=\sum_{k=0}^{N-1}\beta_k^*|k\rangle. \tag{40}$$

We then have that

$$|\Phi\rangle_{23}=(W_2\otimes\mathbb{1}_3)(D_{23}^{\dagger})^2\left(|\Xi_{00}\rangle_{23}-\frac{2}{\sqrt{N}}|\phi^*\rangle_2|\phi\rangle_3\right). \tag{41}$$

A network that performs the operation $(W_2\otimes\mathbb{1}_3)(D_{23}^{\dagger})^2$ could be added to the input of the program register so that the simpler state that appears on the right-hand side of Eq. (41) could be used as the program. At the output of the processor we have to perform the projective measurement discussed in the preceding paragraph, and the probability of achieving the desired result is the same as the probability of successfully implementing the transformation, A. In this case the probability is $1/N^2$.

IV. SUCCESS PROBABILITY

The probability, p, of successfully applying the operator A to the state $|\Psi\rangle_1$ in our example is rather small. This is because the operator we chose was a linear combination of all of the operators $U^{(mn)}$. This means that if the data register consists of l qubits, i.e., $N=2^l$, then the probability of a successful implementation of a general transformation A decreases exponentially with the size of the data register. However, if we were to choose an operator, or set of operators, that was a linear combination of only a few of the $U^{(mn)}$, then the success probability can be significantly improved. This would entail making a different measurement at the output of the program register. Instead of making a projective measurement onto the vector $|M\rangle$, one would instead make a measurement onto the vector

$$|M'\rangle=\frac{1}{\mathcal{N}^{1/2}}\sum_{m,n:q_{mn}\neq 0}|\Xi_{mn}\rangle, \tag{42}$$

where $\mathcal{N}$ is the total number of nonzero coefficients q_{mn}, in the decomposition in Eq. (30). If the operation being implemented is unitary, then, the probability of implementing it is

$$p=\frac{1}{\mathcal{N}}. \tag{43}$$

There are, in fact, large classes of operations that can be expressed in terms of a small number of operators $U^{(mn)}$ [14]. For these operators, the probability of success can be relatively large and, in principle, independent of the size of the Hilbert space of the data register.

Example 1. Let us consider the one-parameter set of unitary transformations U_φ

$$U_\varphi=(\cos\varphi)\mathbb{1}+i(\sin\varphi)\left[\frac{1+i}{2}U^{(01)}+\frac{1-i}{2}U^{(03)}\right], \tag{44}$$

where the unitaries $U^{(mn)}$ are given by Eq. (28). These unitaries for $N=4$ can be explicitly written as

$$U^{(01)}=\sum_{s=0}^{3}(-i)^{s}P_{s}, \quad U^{(03)}=\sum_{s=0}^{3}(i)^{s}P_{s}, \tag{45}$$

where $P_s=|s\rangle\langle s|$. From here we find the expression for the operator (44) in the form

$$U_{\varphi}=(\cos\varphi)\mathbb{1}+i(\sin\varphi)[P_0+P_1-P_2-P_3]. \tag{46}$$

We note that if we rewrite the parameters s as binary numbers, $s=j_1 2+j_0$, where j_k is either 0 or 1, and express the states $|s\rangle$ as tensor products of qubits, i.e., $|s\rangle=|j_1\rangle\otimes|j_0\rangle$, we find that the operator in brackets on the right-hand side of Eq. (44) can be expressed as

$$\left[\frac{1+i}{2}U^{(01)}+\frac{1-i}{2}U^{(03)}\right]=\sigma_3\otimes\mathbb{1}. \tag{47}$$

From Eq. (46) it is clear that U_φ has eigenvalues of magnitude 1, which implies that U_φ is unitary. It can be realized by the universal quantum processor (23) with a probability of successful implementation equal to 1/3. This example illustrates that it is possible to realize large classes of unitary operations with a probability that is greater than the reciprocal of the dimension of the program register.

This example can be easily generalized. Consider a one-parameter set of unitary operators acting on a Hilbert space consisting of l qubits, which is given by

$$U_{\varphi}=(\cos\varphi)\mathbb{1}^{\otimes l}+i(\sin\varphi)\sigma_3\otimes\mathbb{1}^{\otimes(l-1)}. \tag{48}$$

The operator $\sigma_3\otimes\mathbb{1}^{\otimes(l-1)}$ is diagonal and, therefore, only the diagonal unitaries from our set $U^{(mn)}$, i.e., $U^{(m0)}$, appear in its expansion, Eq. (30). Moreover the coefficients q_{m0} in the expansion are nonvanishing only for odd m. It follows that

$$U_{\varphi}=(\cos\varphi)\mathbb{1}^{\otimes l}+i(\sin\varphi)\sum_{\text{odd } m}q_{m0}U^{(m0)}, \tag{49}$$

and the probability of a successful implementation of this unitary transformation is $p=2/(2^l+2)$.

Example 2. For some sets of operators it is possible to do even better than we were able to do in the preceding example. Consider the one-parameter set of unitary operators given by

$$U_{\vartheta}=(\cos\vartheta)\mathbb{1}+i(\sin\vartheta)U^{(0,N/2)}, \tag{50}$$

where N is assumed to be even. That this operator is unitary follows from the fact that $U^{(0,N/2)}$ is self-adjoint. A program vector that would implement this operator is

$$|\Phi\rangle_{23}=\cos\vartheta|\Xi_{00}\rangle_{23}+i\sin\vartheta|\Xi_{0,N/2}\rangle_{23}, \tag{51}$$

and at the output of the program register we make a projective measurement corresponding to the vector

$$|M\rangle_{23}=\frac{1}{\sqrt{2}}(|\Xi_{00}\rangle_{23}+|\Xi_{0,N/2}\rangle_{23}. \tag{52}$$

The probability for successfully achieving the desired result, i.e., the vector $U_\vartheta|\Psi\rangle_1$ in the data register, is 1/2 irrespective of the value N, i.e., the number of qubits.

V. CONCLUSION

We have presented here a programmable quantum processor that exactly implements a set of operators that form a basis for the space of operators on qudits. This processor has a particularly simple representation in terms of elementary quantum gates. It is, however, by no means unique. It is possible, in principle, to build a processor that exactly implements any set of unitary operators that form a basis for the set of operators on qudits of dimension N, and uses any orthonormal set of N^2 vectors as programs. Explicitly, if the set of operators is $\{V_n|n=1,\ldots,N^2\}$ and the program vectors are $\{|y_n\rangle|n=1,\ldots,N^2\}$, the processor transformation is given by

$$P_{dp}=\sum_{n=1}^{N^2}V_n^{(d)}\otimes|y_n\rangle_{p\,p}\langle y_n|, \tag{53}$$

where the superscript (d) on the operator V_n indicates that it acts on the data register.

As an example, consider a data register consisting of l qubits. We could use the processor discussed in Sec. III to perform operations on states in this register, but we can also do something else; we can use l single-qubit processors, one for each qubit of the data register. Specifically, our unitary basis for the set operations on the data register would be

$$U_{JK}=U_{j_1k_1,\ldots,j_lk_l}=\otimes_{m=1}^{l}S_{j_mk_m}, \tag{54}$$

where $J=(j_1,\ldots,j_l)$ and $K=(k_1,\ldots,k_l)$ are sequences of zeros and ones, and the operators $S_{j_mk_m}$ are defined immediately after Eq. (12). The program register would consist of l pairs of qubits, $2l$ qubits in all, with each pair controlling the operation on one of the qubits in the data register. Each of the operators in our basis can be implemented perfectly by a program consisting of the tensor product state, $\Pi_{m=0}^{l}|\Xi_{j_mk_m}^{(m)}\rangle$, where $|\Xi_{j_mk_m}^{(m)}\rangle$ is a two-qubit state that implements the operation $S_{j_mk_m}$ on the mth qubit of the data register.

We are then faced with the problem of which processor to use. This very much depends on the set of operations we want to apply to the data. How to choose the processor so that a given set of operations can be implemented with the greatest probability, for a fixed size of the program register is an open problem. An additional issue is simplicity. One would like the processor itself and the program states it uses to be as simple as possible. The simplicity of the processor is related to the number of quantum gates it takes to construct it. We would maintain that the processors we have presented here are simple, though whether there are simpler ones we do not know. Judging the simplicity of the program states is somewhat more difficult, but they should be related in a relatively straightforward way to the operation that they encode. In many cases these states will have been produced by a previous part of a quantum algorithm, and complicated program states will mean more complexity for the algorithm that produces them. The program states proposed by Vidal and Cirac and the ones proposed by us in Sec. II are, in our opinion, simple.

A final open problem that we shall mention, is finding a systematic way of increasing the probability of successfully carrying out a set of operations by increasing the dimensionality of the space of program vectors. Vidal and Cirac showed how to do this in a particular case, but more general constructions would be desirable [2]. Doing so would give one a method of designing programs for a quantum computer.

ACKNOWLEDGMENTS

This work was supported in part by the European Union projects EQUIP and QUEST, and by the National Science Foundation under Grant No. PHY-9970507.

[1] M.A. Nielsen and I.L. Chuang, Phys. Rev. Lett. **79**, 321 (1997).

[2] G. Vidal and J.I. Cirac, e-print quant-ph/0012067; see also G. Vidal, L. Masanes, and J.I. Cirac, e-print quant-ph/0102037.

[3] D. Bouwmeester, J.W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger, Nature (London) **390**, 575 (1997); S. Bose, P.L. Knight, M.B. Plenio, and V. Vedral, Phys. Rev. Lett. **83**, 5158 (1999); Pramana **56**, 383 (2001); W.L. Li, C.F. Li, and G.-C. Guo, Phys. Rev. A **61**, 034301 (2000).

[4] A. Chefles, Contemp. Phys. **41**, 401 (2000).

[5] L.-M. Duan and G.-C. Guo, Phys. Rev. Lett. **80**, 4999 (1998); P. Deuar, and W.J. Munro, Phys. Rev. A **61**, 062304 (2000).

[6] C.A. Trugenberger, Phys. Rev. Lett. **87**, 067901 (2001).

[7] S.F. Huelga, J.A. Vaccaro, A. Chefles, and M.B. Plenio, Phys. Rev. A **63**, 042303 (2001).

[8] S. Braunstein, V. Bužek, and M. Hillery, Phys. Rev. A **63**, 052313 (2001).

[9] V. Bužek and M. Hillery, Phys. Rev. A **54**, 1844 (1996).

[10] V. Bužek, S. Braunstein, M. Hillery, and D. Bruss, Phys. Rev. A **56**, 3446 (1997).

[11] G. Alber, A. Delgado, N. Gisin, and I. Jex, e-print quant-ph/0008022.

[12] V. Vedral, A. Barenco, and A. Ekert, Phys. Rev. A **54**, 147 (1996).

[13] D.I. Fivel, Phys. Rev. Lett. **74**, 835 (1995).

[14] We note that an arbitrary sum of unitary operators $U^{(mn)}$ is not necessarily a unitary operator.