Quantum Computation with Untunable Couplings

Xingxiang Zhou,[1] Zheng-Wei Zhou,[2] Guang-Can Guo,[2] and Marc J. Feldman[1,*]

[1]*Superconducting Digital Electronics Lab, Electrical and Computer Engineering Department, University of Rochester, Rochester, New York 14627*

[2]*Key Laboratory of Quantum Information, University of Science and Technology of China, Chinese Academy of Sciences, Hefei, Anhui, China, 230026*

(Received 1 April 2002; published 21 October 2002)

Most quantum computer realizations require the ability to apply local fields and tune the couplings between qubits, in order to realize single bit and two bit gates which are necessary for universal quantum computation. We present a scheme to remove the necessity of switching the couplings between qubits for two bit gates, which are more costly in many cases. Our strategy is to compute with encoded qubits in and out of carefully designed interaction free subspaces analogous to decoherence free subspaces. We give two examples to show how universal quantum computation is realized in our scheme with local manipulations to physical qubits only, for both diagonal and off diagonal interactions.

DOI: 10.1103/PhysRevLett.89.197903 PACS numbers: 03.67.Lx

Quantum computation is generally formulated in terms of a collection of qubits subject to a sequence of single and two bit operations [1]. This implies that the effective local fields applied to individual qubits, and the couplings between the qubits, are variable functions subject to external control. In many cases, two bit operations, whose implementation depends on certain interactions between qubits, are more difficult than single bit gates. They can require more sophisticated manipulations, and therefore may take a longer time and cause stronger decoherence. This usually results from the requirement to vary (in the simplest case just switch on and off) the couplings between qubits, which is not always possible or easy to realize. One such example is quantum computing with Josephson junction devices, both charge and flux type [2–6]. In this case, the coupling between qubits is most naturally realized with a hard wired capacitor or inductor, whose value is fixed by the fabrication and cannot be tuned during the computation. The superconducting quantum computing community has been working hard to devise variable coupling schemes [5,7–9], but it is generally agreed that none of these proposed switches is completely satisfactory [9]. Most of them [5,7] require external controls, and thus are likely to be major decoherence sources. Others were designed to avoid such external controls, but may suffer other problems, for instance, the number of qubits that can be incorporated into the system can be limited [8,9], which is at odds with the supposed scalability of a solid state quantum computer.

An always on and untunable coupling causes certain problems for quantum computation, depending on the particular form of the interaction. If the interaction Hamiltonian is diagonal in the computational basis, each qubit state will gain additional phases depending on the states of the qubits to which it is coupled, even in the idle mode. It is then necessary to keep track of these phases, or suppress them by repeated refocusing pulses like those used in NMR, which requires high precisions and complicates the operation [8,9]. In the case of off diagonal interactions, the interactions will moreover cause the states of the qubits to propagate, which results in errors. It is then necessary to devise methods to avoid these problems. Even if the couplings can be tuned, a scheme which allows one to compute without switching the couplings is very useful, because it simplifies the operation drastically and is likely to help reduce decoherence. This simplified computing scheme, in which the necessity of switching the couplings between qubits is removed, is the goal of this work, and we attack this problem by computing in carefully designed subspaces analogous to decoherence free subspaces.

Let us first explain our approach intuitively. One of the strategies to fight against decoherence is to compute in the so-called "decoherence free subspace" (DFS), in which the state of the system (a logical bit consisting of several physical qubits) is unaffected by the environment even though they are always coupled [10]. DFS exists in the case of "collective decoherence," i.e., when the involved qubits couple to the same mode of the environment. Now imagine that we replace the environment with another collection of qubits. Obviously, we expect analogous subspaces of the two coupled collections of qubits to exist, as long as certain conditions similar to those for DFS are satisfied. To be more specific, states that are annihilated by the interaction Hamiltonian will not evolve because of the coupling. If we stay in these subspaces, we can then prevent the (encoded) qubits from affecting each other and operate on the individual logical bits as if they were not coupled to other bits. On the other hand, when we do want the (encoded) bits to interact for two bit operations, we simply drive them out of these subspaces. Therefore, we can effectively turn off and turn on the interaction between the encoded bits by staying in and getting out of these subspaces. We see that even though the subspaces we discussed rely on the same algebraic properties with DFS, we are using them for a different purpose, and we do not intend to stay in them throughout the computation, as opposed to DFS. Another important difference is that in the case of DFS, it is assumed that the environment and its coupling strength with the qubits are uncontrollable. On the contrary, in our case we can certainly manipulate both parties involved and put them in or out of their related subspaces. We can choose the way and the strength with which the qubits are coupled, either in fabrication or during initialization of the system, though the couplings cannot be tuned throughout the computation. Because of these distinctions, we call this new concept the "interaction free subspace" (IFS).

In the following we discuss in detail how our scheme can be used for both diagonal and off diagonal interactions. A general model for a quantum computer is described by the following Hamiltonian:

$$H = -\sum_i \vec{f}_i \cdot \vec{\sigma}_i + \sum_{i<j} \sum_{\alpha,\beta} J_{ij}^{\alpha\beta} \sigma_i^\alpha \sigma_j^\beta, \tag{1}$$

where $\vec{f}_i$ is the effective local field applied to individual qubits, σ^α, σ^β are the Pauli matrices, and J is the coupling strength. As we discussed before, we are interested in situations in which local operations are easy and fast to implement, while two bit operations are hard and slow. We assume that local resources are "free", i.e., strong local pulse fields $\vec{f}_i$ can be applied and single bit gates are instantaneous. We then only count the time when the interaction is on (i.e., when the encoded bits sit out of the IFS). We assume that the values of J_{ij}, i.e., the coupling strength between qubits can be chosen in the fabrication or initialization of the system, but cannot be tuned during the computation. (The dependence on α, β is determined by the nature of the physical interaction.) When the interaction is on, we occasionally need to apply local gates to the individual qubits involved, and we assume that the field used for these local gates is so strong that the local operation is not distorted by the interactions.

We first consider the case of diagonal interactions, i.e., when the interaction between two (physical) qubits takes the form $J_{12}^z \sigma_1^z \cdot \sigma_2^z$ (the Ising interaction). In this case, two physical qubits per logical bit can fulfill our needs. As shown in Fig. 1 a possible architecture of the quantum computer in this case is a one dimensional array consisting of encoded qubits, which are two physical qubits (a, b in the figure) coupled with strength J_0. All physical qubits in neighboring logical bits are coupled with the same strength J_1, which is not necessarily different from J_0. Our codes for the IFS are simply $|0\rangle = |\uparrow_a \downarrow_b\rangle$ and $|1\rangle = |\downarrow_a \uparrow_b\rangle$. Indeed, for two neighboring encoded bits, the interaction Hamiltonian is $H_{\rm int} = J_1(\sigma_{1a}^z + \sigma_{1b}^z) \cdot (\sigma_{2a}^z + \sigma_{2b}^z)$, which annihilates these two states. In addition, these two states are degenerate under the self Hamiltonian $J_0 \sigma_a^z \cdot \sigma_b^z$. Therefore, if we store information in these states, no evolution whatsoever is present. There is thus no need to keep track of any phases; a logical bit in the IFS is automatically in the idle mode. Actually, in order to avoid interactions between the encoded qubits, it suffices to keep half of the qubits (all odd or even numbered ones) in the IFS. However, as will be seen it is inevitable to get out of the IFS even when single bit operations are performed on the encoded qubits, since the only resources we have are single (physical) bit rotations. We thus keep all the logical bits in the IFS during the idle mode.

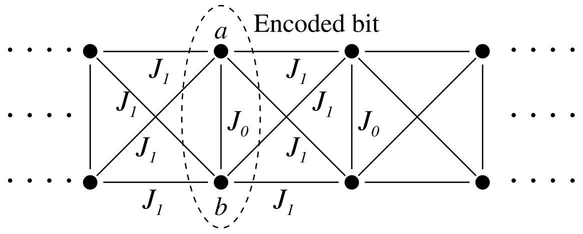

FIG. 1. Architecture of the quantum computer for diagonal interactions. Each dot is a physical qubit and the lines represent couplings between qubits. Two qubits (a, b) connected by a vertical line is an encoded qubit.

Now we discuss how universal computation can be realized on these encoded qubits, using local operations to the physical qubits only. Suppose we are operating on a particular encoded bit (meaning applying single bit gates on its qubits a and b). Since its neighboring bits are in IFS, we can work on the particular bit as if it were decoupled from the rest of the system. First, we notice that single bit gates can be decomposed into arbitrary rotations around the z axis and the Hadamard gate

$$\mathrm{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2}$$

To induce a rotation around the z axis, all we have to do is to break the degeneracy between $|0\rangle$ and $|1\rangle$, which can be done by applying a local field in the z direction to bit a (or b) for a certain amount of time. A Hadamard gate on the bases $|0\rangle$, $|1\rangle$ is more complex, but as can be easily verified it can be realized with the gate sequence CNOT$(a, b) \cdot \mathrm{H}_a \cdot$ CNOT(a, b), where CNOT(a, b) is a CNOT gate between a, b with a as the control bit, and H_a is a Hadamard on bit a. H_a is readily realizable, while CNOT(a, b) can be done by sandwiching a CPHASE gate between a and b with H_a. The CPHASE gate is easy to implement with the Ising interaction and local phase gates: CPHASE $= e^{-i\sigma_a^z \sigma_b^z \pi/4} e^{i\sigma_a^z \pi/4} e^{i\sigma_b^z \pi/4}$. Altogether, a Hadamard on the code states requires nine local gates and two interaction periods, which takes a time $\pi/2J_0$ (local gates are assumed instantaneous). We note that it is necessary to get out of the code space $|\uparrow_a \downarrow_b\rangle$, $|\downarrow_a \uparrow_b\rangle$ in order to realize the Hadamard. This is unavoidable in the current model, since the only allowed resources are local unitaries. As a result, single bit gates on the encoded qubits cannot be performed simultaneously to neighboring bits. But a "half parallel" operation mode is still allowed, in which all odd or even numbered logical bits are operated on at the same time. This restriction can be removed by exploring more complex encoding schemes (see the example for off diagonal interactions below), but more resources (more physical qubits per logical bit) are required.

We still need to show how two bit gates can be realized. For this purpose, we need to drive the involved (neighboring) bits out of the IFS, let them interact for some time, then drive them back. The first step is to apply a local gate to flip the state of b, which changes the two code states to $|\uparrow_a\uparrow_b\rangle$ and $|\downarrow_a\downarrow_b\rangle$. Note these two states are eigenstates of $\Sigma^z/2 = (\sigma_a^z + \sigma_b^z)/2$ with eigenvalues ± 1. It is then straightforward that a CPHASE gate between 1, 2 (on $|\uparrow_a\uparrow_b\rangle$ and $|\downarrow_a\downarrow_b\rangle$) can be realized with the sequence $e^{-i(\Sigma_1^z/2)(\Sigma_2^z/2)\pi/4}e^{i(\Sigma_1^z/2)\pi/4}e^{i(\Sigma_2^z/2)\pi/4}$, or $e^{-i(\sigma_{1a}^z+\sigma_{1b}^z)(\sigma_{2a}^z+\sigma_{2b}^z)\pi/16}e^{i\sigma_{1a}^z\pi/8}e^{i\sigma_{1b}^z\pi/8}e^{i\sigma_{2a}^z\pi/8}e^{i\sigma_{2b}^z\pi/8}$. Once this is done, we simply flip the state of b again to drive the logical bits back into the IFS. This procedure realizes a CPHASE gate between two encoded bits and puts them back in the IFS at the end. A total number of eight local operations and one interaction is necessary, and the time required is $\pi/16J_1$. This time is actually shorter than that needed in an ordinary Ising model with switchable couplings, $\pi/4J_1$.

For completeness let us discuss briefly how the system can be initialized in the IFS and how the states of the encoded bits can be measured. If we apply a strong global field in the z direction to all the physical qubits, at low temperatures all bits will line up with the field. Then starting with the leftmost qubit, we can drive all the bits into IFS by simply flipping the state of one of a, b. To read out the state of the encoded bits, a measurement on its a or b suffices.

With the procedures discussed above for diagonal interactions, we can turn on and off the interactions between the logical qubits without a physical switch. This is readily applicable in superconducting quantum computation. Here, local fields can be easily applied simply by changing the biases of the superconducting qubits or applying ac fields. Typical times for single bit gates range from hundreds of ps to tens of ns [2,5,8], depending on the type of the qubits and the choice of the parameters. Coherent control of single superconducting qubits has been experimentally realized [11,12]. With the fast operation speed [11] and long decoherence time [12] experimentally demonstrated, large scale superconducting quantum computers can be constructed with the aid of IFS. The uniform couplings required (in Fig. 1, couplings between physical qubits in neighboring logical bits should all be J_1) are relatively easy to realize in superconducting designs, as mutual inductances can be calculated and fabricated very precisely. Were it necessary to compensate for fabrication imperfections, simple schemes for minor adjustment of the coupling are readily accomplished. This calibration step can be very slow, so the leads used for calibration can be heavily filtered to keep out the noise. For more detailed discussion on the application of our scheme in superconducting quantum computing, see Ref. [13].

We now turn to the case of off diagonal interactions. We will focus on the isotropic and anisotropic exchange interactions, $H_{\text{int}} = J_{xy}(\sigma_1^x\sigma_2^x + \sigma_1^y\sigma_2^y) + J_z\sigma_1^z\sigma_2^z$, since these are most frequently encountered in quantum computing proposals. If both dissipation and decoherence are present, decoherence free subspace requires 4 bits per encoded bit [10]. However as we discussed earlier we have more control in constructing IFS, and it is possible to use less resources. In our case 3 bits per logical bit is enough. A few designs are possible, but let us consider the architecture shown in Fig. 2. Here, each logical bit contains three physical qubits. We have represented them with stars and dots, not because they are physically distinct, but because they play different roles. The stars are the information carrying qubits, while the dots are "isolators" in the singlet state $(|\uparrow_{i_1}\downarrow_{i_2}\rangle - |\downarrow_{i_1}\uparrow_{i_2}\rangle)/\sqrt{2}$. To see the origin of the name, we note that all the stars are coupled to their neighboring dots with the same strength J_1 (meaning the same J_{xy}, J_z). Hence the interaction Hamiltonian between the information carrier and its isolator is $2J_{xy}(\sigma_q^+\Sigma_i^- + \sigma_q^-\Sigma_i^+) + J_z\sigma_q^z\Sigma_i^z$, where $\sigma^\pm = (\sigma_x \pm i\sigma_y)/2$ and $\Sigma_i^{\pm,z} = \sigma_{i_1}^{\pm,z} + \sigma_{i_2}^{\pm,z}$. Since the singlet state is annihilated by the operators $\Sigma_i^{\pm,z}$, we see that if all the dots are in the singlet state, the stars will be isolated from each other and no phase exchange and state propagation will happen, hence the name isolators. The IFS is spanned by $|\uparrow_q\rangle(|\uparrow_{i_1}\downarrow_{i_2}\rangle - |\downarrow_{i_1}\uparrow_{i_2}\rangle)/\sqrt{2}$ and $|\downarrow_q\rangle(|\uparrow_{i_1}\downarrow_{i_2}\rangle - |\downarrow_{i_1}\uparrow_{i_2}\rangle)/\sqrt{2}$. To prepare the isolators in the singlet state, we turn on all couplings between the dots (the vertical dashed lines) while keep all couplings between the stars and the dots (the solid lines) off during the initialization process. This is necessary to stop the propagation of the states in the qubit array. Here we are assuming that switching of the coupling is possible but hard and slow. Since initialization and computation are subject to different restrictions (initialization needs done only once, and it does not need to be done quickly), the global switches are used to initialize the system. At low temperatures, the isolators will then relax to the singlet which

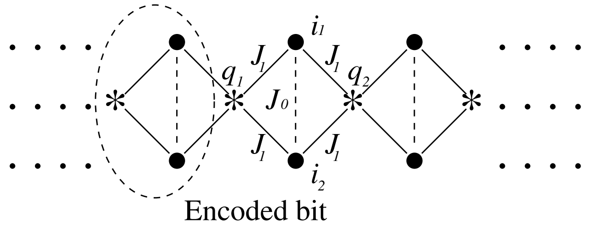

FIG. 2. Architecture of the quantum computer for off diagonal interactions. Each logical bit consists of three physical bits, the information carrying qubit (the star) and an "isolator" (the dots connected with a dashed line).

is the lowest energy eigenstate. Once the initialization is done, we can then start the computation by turning off all couplings represented by dashed lines and turning on those represented by solid lines. According to our assumption, these couplings will remain untuned throughout the computation.

We see that single bit gates are trivial; we simply operate on the stars directly as if they were not coupled to anything else. Fully parallel operations are possible, thanks to the use of isolators. Two bit gates are more complicated. A two bit operation between the stars q_1 and q_2 in Fig. 2 must involve the isolator i_1, i_2, which separates them, because there is no direct connection between q_1 and q_2. One idea is to swap the states of q_1 and i_1, perform a control gate between i_1 and q_2, and swap the states of q_1 and i_1 back. These steps can be done, if we can simulate dynamics generated by $\sigma^x_{q_1}\sigma^x_{i_1} + \sigma^y_{q_1}\sigma^y_{i_1}$ (and similarly for i_1 and q_2), which can be used to generate swap and CNOT gates [9]. This is possible with our available resources, namely, local unitaries and the interaction Hamiltonian $H_{qi} = J_{xy}(\Sigma^x_q\Sigma^x_i + \Sigma^y_q\Sigma^y_i) + J_z\Sigma^z_q\Sigma^z_i$ [14]. Here we discuss a method based on selective coupling. Note that for a small time $t = \pi/(32NJ_{xy})$, where N is some large integer, we have $e^{-iH_{qi}t}\sigma^x_{q_1}\sigma^x_{q_2}e^{-iH_{qi}t}\sigma^x_{q_1}\sigma^x_{q_2} = e^{-i2J_{xy}t\Sigma^x_q\Sigma^x_i} + o(t)$. This can be further used to generate operators we want: $e^{-i2J_{xy}t\Sigma^x_q\Sigma^x_i}\sigma^z_{q_2}e^{-i2J_{xy}t\Sigma^x_q\Sigma^x_i}\sigma^z_{q_2} = e^{-i4J_{xy}t\sigma^x_{q_1}\Sigma^x_i}$, $e^{-i4J_{xy}t\sigma^x_{q_1}\Sigma^x_i}\sigma^z_{i_2}e^{-i4J_{xy}t\sigma^x_{q_1}\Sigma^x_i}\sigma^z_{i_2} = e^{-i8J_{xy}t\sigma^x_{q_1}\sigma^x_{i_1}}$. Similarly we can synthesize $e^{-i8J_{xy}t\sigma^y_{q_1}\sigma^y_{i_1}}$. Repeating this procedure N times, we then get the transformation $e^{-i(\sigma^x_{q_1}\sigma^x_{i_1}+\sigma^y_{q_1}\sigma^y_{i_1})\pi/4}$, which is a swap gate between q_1 and i_1 (and multiplication by $-i$ when their states are different) [9]. In order to reduce the error, the number of repetition N can be quite large [15], therefore many local gates ($44N$ in total) are needed. The interaction time needed is $\pi/2J_{xy}$. The CNOT (or CPHASE) gate between i_1 and q_2 after this swap operation and the operation to swap back the states of i_1 and q_1 can be done by following the same procedure. The interaction times required are $\pi/2J_{xy}$ each. Therefore the total interaction time is $3\pi/2J_{xy}$, in comparison with $\pi/4J_{xy}$ which is needed in a switchable XY model [16]. This prescription verifies the possibility of universal quantum computation in our current example. Finding physical systems to which our scheme discussed above can apply, and a set of manipulations that allow us to minimize the complexity of the operation, is of further interest to us.

In summary, we have devised a scheme for universal and scalable quantum computation without the need to tune the couplings between qubits. This relies on the idea of computing with logical bits consisting of several physical qubits, which can be put in and driven out of the IFS. We gave two examples of how universal quantum computation can be done with our scheme. We emphasize that our strategy can be adopted in forms and contexts other than the examples discussed. With potential applications in superconducting quantum computation and a few other cases, our scheme is likely to help simplify the design and ease the operation of quantum computers.

Before we conclude, we should mention that the "complimentary" problem to ours, in which single bit operations are hard and desired to be avoided, has been discussed [17]. Our scheme is much in the same spirit in the sense of using encoded qubits for computation, but it is for a different purpose and it has a closer analogy to decoherence free subspaces.

X. Z. thanks L. Tian for helpful discussions. The work of X. Z. and M. J. F. was supported in part by AFOSR Grant No. F49620-01-1-0457 and funded under the DoD DURINT program and by the ARDA. Z-W. Z. and G-C. G. were supported by the National Fundamental Research Program (2001CB309300), National Natural Science Foundation of China, and the Innovation funds from the Chinese Academy of Sciences.

*Email address: feldman@ece.rochester.edu

[1] A. Barenco *et al.*, Phys. Rev. A **52**, 3457 (1995).

[2] A. Shnirman, G. Schön, and Z. Hermon, Phys. Rev. Lett. **79**, 2371 (1997).

[3] D.V. Averin, Solid State Commun. **105**, 659 (1998).

[4] M. F. Bocko, A. M. Herr, and M. J. Feldman, IEEE Trans. Appl. Supercond. **7**, 3638 (1997).

[5] J. E. Mooij *et al.*, Science **285**, 1036 (1999).

[6] D.V. Averin, J. R. Friedman, and J. E. Lukens, Phys. Rev. B **62**, 11 802 (2000).

[7] J. Clarke *et al.* (unpublished); T.V. Filippov *et al.* (unpublished); J. E. Mooij (unpublished).

[8] Y. Makhlin, G. Schön, and A. Shnirman, Nature (London) **398**, 305 (1999).

[9] Y. Makhlin, G. Schön, and A. Shnirman, Rev. Mod. Phys. **73**, 357 (2001).

[10] L-M. Duan and G-C. Guo, Phys. Rev. Lett. **79**, 1953 (1997); P. Zanardi and M. Rasetti, Phys. Rev. Lett. **79**, 3306 (1997); D. A. Lidar, I. L. Chuang, and K. B. Whaley, Phys. Rev. Lett. **81**, 2594 (1998).

[11] Y. Nakamura, C. D. Chen, and J. S. Tsai, Nature (London) **398**, 786 (1999).

[12] D. Vion *et al.*, Science **296**, 886 (2002); Y. Yu *et al.*, Science **296**, 889 (2002).

[13] M. J. Feldman and X. Zhou, in "Quantum Computing and Quantum Bits in Mesoscopic Systems" (Kluwer Academic Plenum Publishers, Dordrecht, to be published).

[14] S. Lloyd, Phys. Rev. Lett. **75**, 346 (1995).

[15] J. L. Dodd *et al.*, Phys. Rev. A **65**, 040301(R) (2002).

[16] A. Imamoglu *et al.*, Phys. Rev. Lett. **83**, 4204 (1999).

[17] D. P. Divincenzo *et al.*, Nature (London) **408**, 339 (2000); D. A. Lidar and L-A. Wu, Phys. Rev. Lett. **88**, 017905 (2002).