High-threshold universal quantum computation on the surface code

Austin G. Fowler,[1] Ashley M. Stephens,[1] and Peter Groszkowski[2]

[1]*Centre for Quantum Computer Technology, University of Melbourne, Victoria, Australia*

[2]*Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario, Canada*

We present a comprehensive and self-contained simplified review of the quantum computing scheme of Raussendorf *et al.* [Phys. Rev. Lett. **98**, 190504 (2007); N. J. Phys. **9**, 199 (2007)], which features a two-dimensional nearest-neighbor coupled lattice of qubits, a threshold error rate approaching 1%, natural asymmetric and adjustable strength error correction, and low overhead arbitrarily long-range logical gates. These features make it one of the best and most practical quantum computing schemes devised to date. We restrict the discussion to direct manipulation of the surface code using the stabilizer formalism, both of which we also briefly review, to make the scheme accessible to a broad audience.

I. INTRODUCTION

Classical computers manipulate bits that can be exclusively 0 or 1. Quantum computers manipulate quantum bits (qubits) that can be placed in arbitrary superpositions $\alpha|0\rangle + \beta|1\rangle$ and entangled with one another $(|00\rangle + |11\rangle)/\sqrt{2}$. This additional flexibility provides both additional computing power and additional challenges when attempting to correct the now quantum errors in the computer. An extremely efficient scheme for quantum error correction and fault-tolerant quantum computation is required to correct these errors without making unphysical demands on the underlying hardware and without introducing excessive time overhead and thus wasting a significant amount of the potential performance increase.

This paper is a simplified review of the quantum computing scheme of [1,2]. The scheme requires a two-dimensional (2D) square lattice of nearest-neighbor coupled qubits with initialization, readout, memory, and quantum gates all operating with error rates less than approximately 1%—the least challenging set of physical requirements devised to date. Furthermore, despite the modest physical requirements, logical qubits (qubits of data distributed over many physical qubits and protected by error correction) can be interacted over arbitrarily large distances with time overhead only growing logarithmically in their separation. This is remarkable as most nearest-neighbor quantum computing schemes are associated with a time overhead that grows linearly with logical qubit separation. Finally, in most, if not all, physical quantum computer technologies, bit flips $|0\rangle \leftrightarrow |1\rangle$ are less likely than phase flips $|1\rangle \leftrightarrow -|1\rangle$ opening the door for asymmetric error correction schemes that make use of fewer physical qubits to preserve a given amount of quantum data with a given confidence level. Practically, it is also helpful if additional error correction resources can be dynamically allocated to critical data during the quantum computation. The scheme we review permits both asymmetric and dynamic error correction in a natural manner.

A number of technologies are well suited to implementing surface code quantum computing. Proposals exist for 2D arrays of qubits making use of superconductors [3,4] and semiconductor nanophotonics [5]. An equivalent measurement based version of the scheme calling for a three-dimensional cluster state [1,2,6] could be implemented using photonic modules [7,8] or ion traps [9].

The discussion is organized as follows. In Sec. II we briefly review the stabilizer formalism of quantum computing [10]. Section III briefly reviews the surface code [11], which forms the error correction substrate of everything that follows. Logical qubits are introduced into the surface code in Sec. IV, along with their initialization, measurement, and basic logical operations. Section V describes logical controlled-NOT (CNOT) in detail. Section VI completes the universal set of logical gates with a discussion of state injection, state distillation, and appropriate quantum circuits making use of the distilled states. An efficient implementation of logical Hadamard inspired by [12] that avoids the extensive machinery of Sec. VI is described in Sec. VII. Section VIII then describes simulations used to estimate the threshold error rates of physical qubit initialization, measurement, memory, and two-qubit gates. Looking further into the future, Sec. IX discusses distributed quantum computing to make it clear that impractically large 2D square lattices of qubits are not required to tackle problems of interesting size. Section X summarizes the discussion and points to further reading.

II. QUANTUM STATES AND STABILIZERS

A quantum state can be specified in a number of equivalent ways. Two of the most common are to choose a basis and to express the state as a state vector such as $(|00\rangle + |11\rangle)/\sqrt{2}$. In this review, it will be much more convenient to express this state as the unique simultaneous +1 eigenvector of the commuting operators $X \otimes X$ and $Z \otimes Z$. Such operators are called stabilizers [10]. This entire review is based on the manipulation of stabilizers.

Any set of n mutually commuting and independent operators over n qubits has a unique simultaneous +1 eigenstate. We will restrict our attention to stabilizers that are a tensor product of the identity operator I and the Pauli matrices X, Y, Z (with $Y = XZ$ real). A set of such stabilizers cannot be used to specify an arbitrary quantum state, though a sufficiently broad range of states can be specified for most of our pur-

poses. See Sec. VI A for a simple extension to the stabilizer formalism permitting arbitrary states to be specified.

Consider a set of stabilizers M specifying state $|\psi\rangle$. Suppose we wish to apply an operator U to state $|\psi\rangle$. If we consider $U|\psi\rangle = UMU^\dagger U|\psi\rangle$, we can see that the new set of stabilizers will be $UMU^\dagger$. To give an explicit example,

$$M = Z_1Z_2, \ Z_2Z_3, \ Z_3Z_4, \ X_1X_2X_3X_4, \tag{1}$$

$$U = X_2, \tag{2}$$

$$UMU^\dagger = -Z_1Z_2, \ -Z_2Z_3, \ Z_3Z_4, \ X_1X_2X_3X_4. \tag{3}$$

In addition to unitary manipulation, we will frequently discuss measurement of a given operator, for example, X, Z, or some more complicated operator involving a larger tensor product. A very simple example is a single qubit in an unknown state with stabilizer I and the subsequent measurement of the Z operator. We will write the stabilizer of a qubit after such a measurement as $\pm Z$. Note that the probabilities of the two possible measurement outcomes, the $+1$ and -1 eigenstates of Z, are typically not recorded in the stabilizer formalism, just their possibility. Note also that given any operator there is always a nonzero probability of obtaining at least one of the two eigenstates.

Care needs be taken when measuring if other nontrivial stabilizers are present. There are three cases to consider. If the operator to be measured can be expressed as a product of stabilizers, no change is made to the stabilizers as we already have an eigenstate of the operator. For example, if we have two qubits and stabilizers Z_1 and $-Z_2$, measuring the Z_1Z_2 operator will always give the -1 eigenstate.

If the operator to be measured cannot be expressed as a product of stabilizers and commutes with each stabilizer, the operator is added to the list of stabilizers with a sign that depends on whether we have projected into the $+1$ or -1 eigenstate of the operator. For example, if we have three qubits and stabilizers Z_1Z_2 and Z_2Z_3, measuring Z_2 will yield one of the ± 1 eigenstates, meaning we will introduce the new stabilizer $\pm Z_2$. Note again that the probability of the two outcomes is neither recorded nor known.

Finally, if the operator cannot be expressed as a product of stabilizers and anticommutes with one or more stabilizers, the second and subsequent anticommuting stabilizers are multiplied by the first anticommuting stabilizer to form commuting stabilizers and the first anticommuting stabilizer is replaced with the operator being measured, again with sign depending on which state we have projected into. For example, if we have four qubits and stabilizers Z_1Z_2, Z_2Z_3, and Z_3Z_4, to measure X_3 we first multiply Z_3Z_4 by Z_2Z_3 and then replace Z_2Z_3 with $\pm X_3$ to give the new set of stabilizers Z_1Z_2, $\pm X_3$, and Z_2Z_4. In this instance we know that the probability of the two outcomes is equal as, given any state $|\psi\rangle$ stabilized by some operator S and any operator to measure M such that $MS = -SM$, we have

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left[\frac{1}{\sqrt{2}}(1+M)|\psi\rangle + \frac{1}{\sqrt{2}}(1-M)|\psi\rangle\right], \tag{4}$$

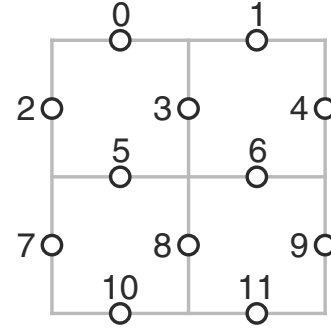meaning we have an equal superposition of the ± 1 eigenstates of M.

FIG. 1. Basic layout of surface code data qubits, each represented by a circle. A data qubit is located at the center of each edge of a square lattice. The square lattice is a guide for the eye only; it does not represent interactions.

Many other examples of measurements falling into each of these three categories will be discussed in subsequent sections.

III. SURFACE CODE

The surface code was first presented in [11]. A small surface showing the basic layout of qubits, a square grid with qubits on each edge, is shown in Fig. 1. The stabilizers of this surface are

$$X_0X_2, \quad X_0X_1X_3, \quad X_1X_4, \quad X_2X_5X_7, \quad X_3X_5X_6X_8,$$

$$X_4X_6X_9, \quad X_7X_{10}, \quad X_8X_{10}X_{11}, \quad X_9X_{11},$$

$$Z_0Z_2Z_3Z_5, \quad Z_1Z_3Z_4Z_6, \quad Z_5Z_7Z_8Z_{10}, \quad Z_6Z_8Z_9Z_{11}. \tag{5}$$

These correspond to a tensor product of Z around each face and X around each vertex. Note that X_9X_{11} can be expressed as a product of the other X stabilizers. This leaves 12 independent stabilizers on 12 qubits implying a unique state. Given a w by h face surface, in general there will be $2wh + w + h$ qubits and independent stabilizers.

Not shown in Fig. 1 are additional qubits on each face and vertex that enable one to check the sign of the associated stabilizer. These additional syndrome qubits make the lattice a simple nearest-neighbor connected square lattice. Discussion of the syndrome qubits and the quantum circuits used to extract the signs of the stabilizers will be deferred until Sec. VIII.

If no errors of any kind occur, the surface remains in the simultaneous $+1$ eigenstate of every stabilizer. When discussing errors, we will restrict our attention to bit flips and phase flips. Very general noise can be tolerated with just the ability to correct these two types of errors [13]. Figure 2 shows the effect of single bit flips and phase flips on the surface code—the adjacent stabilizers become negative. If we could reliably detect when a stabilizer becomes negative, this clearly would be sufficient to pinpoint and then correct these single errors.

Two additional complications need to be accounted for. First, it is possible for long chains of errors to occur. Second, it is possible for the reported eigenvalue of a given stabilizer
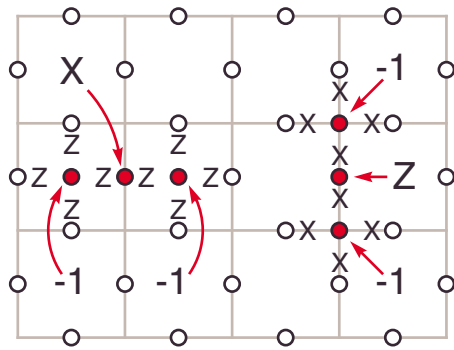
FIG. 2. (Color online) Effect of single bit flips (X) and phase flips (Z) on the surface code—adjacent face and vertex stabilizers are made negative, respectively.

FIG. 4. (Color online) (a) Locations in space and time, indicated by filled dots, where and when the reported syndrome is different from that in the previous time step. Note that this is not a three-dimensional physical structure, just a three-dimensional classical data structure. (b) Optimal matching highly likely to lead to a significant reduction of the number of errors if bit-flips are applied to the spacelike edges.

to be wrong. Both of these situations are illustrated in Fig. 3. To cope with these complications, we keep track of every time the reported eigenvalue of each stabilizer changes. Without loss of generality, let us focus solely on Z stabilizers, which detect bit flips, as both types of errors are treated independently.

Figure 4(a) gives an example of appropriate Z stabilizer information. In practice, correction is delayed as long as possible and pairs of flipped syndromes are then connected by paths in space and time or "matched" such that the total number of edges used is minimal, as shown in Fig. 4(b). Polynomial time minimum weight matching algorithms exist [14], hence this can be done efficiently. Recently, logarithmic time parallel algorithms have been developed [15,16]. Note that X errors can be matched to smooth boundaries and Z errors to rough boundaries of the surface. A smooth boundary is a boundary with four term Z stabilizers and three term X stabilizers as shown in Fig. 5. A rough boundary is a boundary with four term X stabilizers and three term Z stabilizers, also shown in Fig. 5. Given a minimum weight matching, bit flips are applied to the spacelike edges to correct the errors with high probability. Further discussion of the details of error correction will be delayed until Sec. VIII.

Initialization of the surface code substrate is not completely trivial. If every qubit is prepared in the $|0\rangle$ state, we automatically have the $+1$ eigenstate of every Z stabilizer, but when we measure the X stabilizers the eigenstates will be

randomly positive and negative. For simplicity, we choose to treat the random negative eigenvalues as errors and correct them.

IV. LOGICAL QUBITS

Armed with the surface code described in Sec. III, we can now discuss logical qubits. The simplest logical qubit consists of a single face where we stop measuring the associated Z stabilizer. This introduces one new degree of freedom into the surface. We can manipulate this degree of freedom using any chain of X operators connecting this face or "smooth defect" to a smooth boundary and any chain of Z operators encircling the smooth defect as shown in Fig. 6. We choose to call any such X chain X_L and any Z ring Z_L. This implies
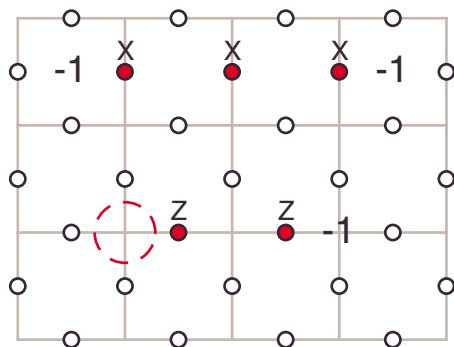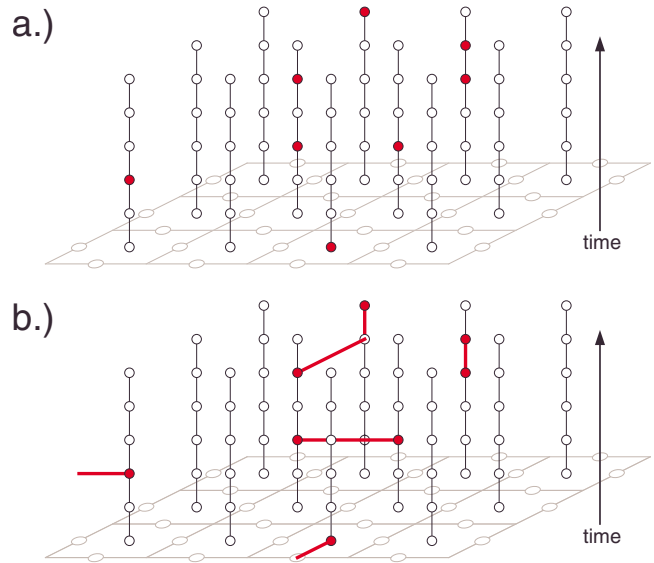

FIG. 3. (Color online) Surface code suffering from multiple errors (indicated by filled dots) and an incorrect syndrome measurement (indicated by dashed circle).
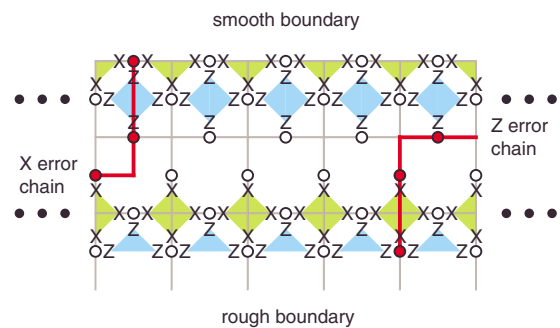

FIG. 5. (Color online) Examples of smooth and rough boundaries including a chain of X errors ending in a smooth boundary without changing the sign of any stabilizers, and a chain of Z errors ending in a rough boundary, also without leaving any evidence of its presence. X and Z stabilizers are represented by shaded shapes.
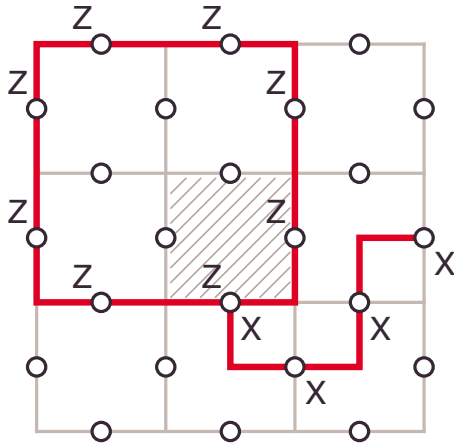
FIG. 6. (Color online) Surface code with one additional degree of freedom introduced by not enforcing the stabilizer associated with one face (shaded). This face or a region of such faces is called a smooth defect. The degree of freedom can be phase flipped by any ring of Z operators encircling the defect and bit flipped by any chain of X operators connecting the defect to a smooth boundary.

that, by definition, our logical qubit is initialized to $|0_L\rangle$ as the surface is initially in the simultaneous +1 eigenstate of every Z stabilizer and therefore also in the +1 eigenstate of Z_L.

Larger smooth defects can be created using X measurements as shown in Fig. 7. Note that arbitrarily large defects still only introduce one degree of freedom. The given example shows the removal of four qubits and five stabilizers—four Z stabilizers and one X stabilizer. After the X measurements, a number of new three term X stabilizers are created with not necessarily positive sign. As in the case of surface initialization, we will treat any negative eigenvalues as syndrome changes which will then be matched and corrected with chains of Z operators. The qubits inside the defect, which have been projected into a product state, play no further role in the computation unless the defect moves.

In practice, it is inconvenient to use a logical qubit with a logical operator that connects to a potentially distant boundary. This situation can be avoided by using a pair of defects to represent a single logical qubit as shown in Fig. 8. A chain
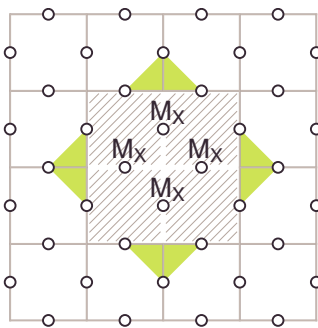

FIG. 7. (Color online) Surface code with one degree of freedom introduced via the measurement M_X of four qubits in the X basis and removal of five stabilizers. Note that four new three term X stabilizers are created with not necessarily positive sign (indicated by shaded triangles).
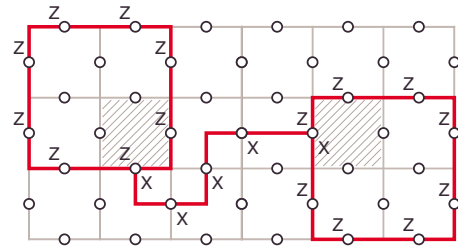

FIG. 8. (Color online) Smooth qubit comprised of two smooth defects. Z_L corresponds to any ring of Z operators around either defect. X_L corresponds to any chain of X operators connecting the two defects.

of X operators connecting the two defects is then used as the X_L operator. The Z_L operator is any ring of Z operators around either defect—these two classes of Z_L operators are equivalent as they have the same commutation relations.

Effectively, the above means we are choosing to represent an arbitrary logical state by $\alpha|0_L\rangle|0_L\rangle+\beta|1_L\rangle|1_L\rangle$ as defined in the opening paragraph of this section. For the remainder of the review we shall redefine $|0_L\rangle$ and $|1_L\rangle$ such that an arbitrary logical state of a double defect logical qubit can be expressed as simply $\alpha|0_L\rangle+\beta|1_L\rangle$. Note that double smooth defect logical qubits are also initialized to $|0_L\rangle$ by default.

Double smooth defect logical qubits can also be initialized in the $|+_L\rangle$ state by first preparing a region of $|+\rangle$ as shown in Fig. 9. Such a region is automatically in the +1 eigenstate of X_L operators and X stabilizers not intersecting the boundary. X stabilizers on the boundary will have random sign. Smooth defects are then created by measuring all Z stabilizers outside the desired defect locations. The signs of the Z stabilizers will be random and we will again treat negative stabilizers as syndrome changes, match them, and correct them with chains of X operators. We will henceforth refer to a double smooth defect logical qubit as simply a smooth qubit.

Rough qubits are also possible to create via Z measurements as shown in Fig. 10. In this case the Z_L operator is any chain of Z operators linking the two defects and X_L any ring of X operators around either defect. Rough qubits are initialized to the +1 eigenstate of X_L, $|+_L\rangle$, by default, although $|0_L\rangle$ can be prepared starting with a region of qubits in the $|0\rangle$ state.
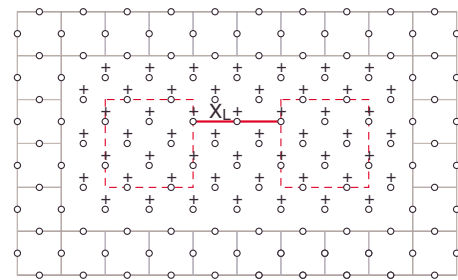

FIG. 9. (Color online) Initializing a smooth qubit in the $|+_L\rangle$ state. After preparing a region of qubits each in the $|+\rangle$ state, every X stabilizer on the boundary of the region and every Z stabilizer outside the dashed regions is measured. Negative eigenvalues are treated as errors and corrected.
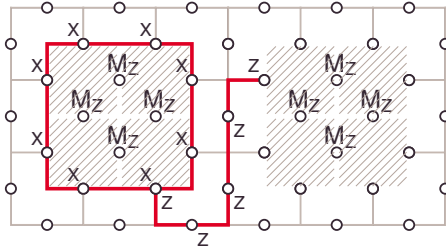
FIG. 10. (Color online) Initializing a rough qubit in the $|+_L\rangle$ state via Z basis measurements M_Z and ignoring stabilizers (shaded). X_L is any ring of X operators around either defect. Z_L is any chain of Z operators linking the two defects.

Logical measurement is similar to initialization. To measure a smooth qubit in the Z_L basis, a region of qubits encircling either or both defects is measured in the Z basis. In the absence of errors every path encircling either defect will have the same parity of Z measurements. If errors are present, they can be detected and corrected using the standard error correction procedure as directly measuring qubits in the Z basis is also an acceptable way to gain information about the eigenvalues of the Z stabilizers—even parity of Z measurements around a face corresponding to a positive eigenvalue and odd parity corresponding to a negative eigenvalue. Note that, as shown in Fig. 11, it is possible for every face to have even parity, meaning no errors, and every path around either defect to have odd parity, meaning a readout result of $|1_L\rangle$.

A smooth qubit can be measured in the X_L basis by measuring a region including both defects in the X basis. In this instance the parity of all chains of X measurements connecting the two defects will be the same in the absence of errors. Similarly, rough qubits can be easily measured in either logical basis.

V. LOGICAL CNOT

So far, we have discussed two types of logical qubits, smooth and rough, schemes to initialize and measure them in the Z_L and X_L bases and Z_L and X_L operations. The only two logical qubit gate in this scheme is the logical CNOT gate. To understand how logical CNOT works, we first need to under-
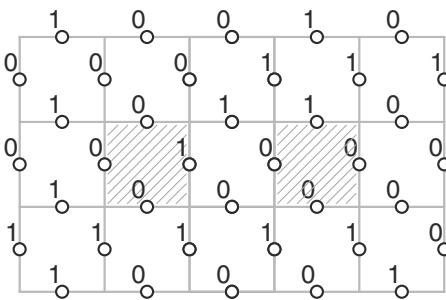

FIG. 11. Example of measurement of a smooth qubit in the Z_L basis in the absence of errors. Note that the measurements around every face have even parity whereas the parity of any path of measurements encircling either defect is odd. The figure thus corresponds to the measurement result $|1_L\rangle$.
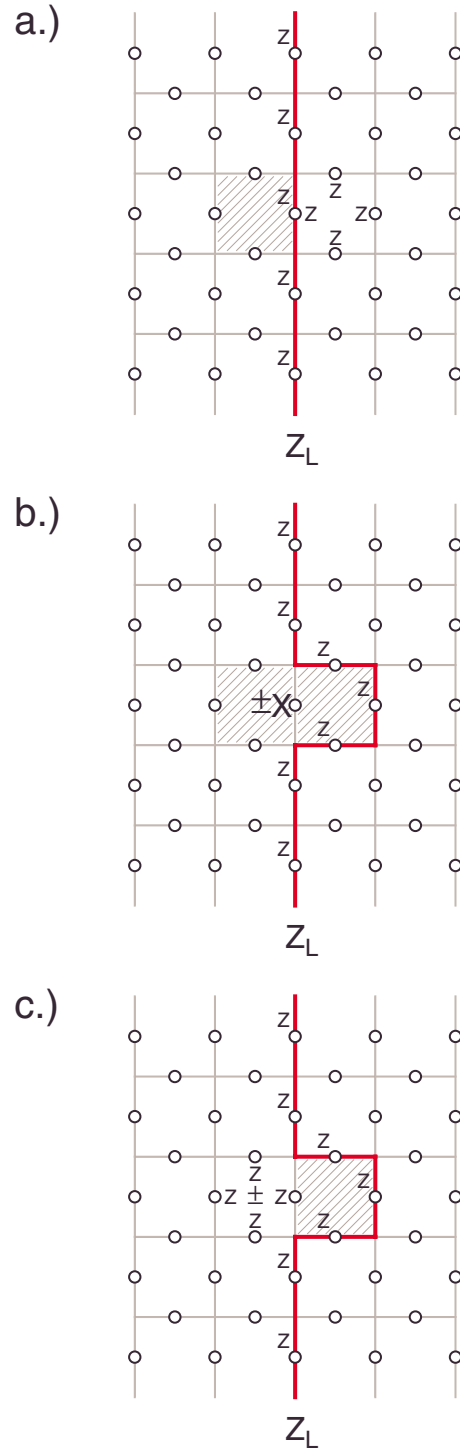

FIG. 12. (Color online) (a) Smooth defect and surface in the +1 eigenstate of Z_L. (b) After measuring the center qubit in the X basis, the shape of the Z_L operator is deformed. (c) Measuring and possibly correcting the indicated Z stabilizer using a bit flip on the center qubit completes the movement of the defect.

stand in detail the effect of moving a smooth defect.

Consider Fig. 12(a). This shows a smooth defect and two stabilizers—a single face Z stabilizer and a Z_L stabilizer. If we now measure the center qubit in the X basis as shown in Fig. 12(b), we will be left with the center qubit in the $\pm X$ eigenstate and a stabilizer equal to the product of the face
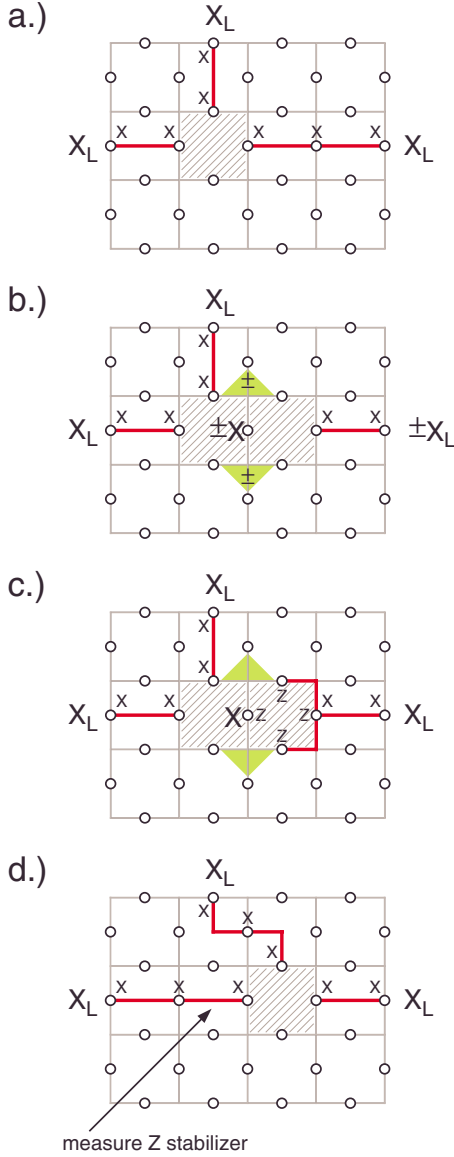
FIG. 13. (Color online) (a) Smooth defect and surface in the +1 eigenstate of X_L. (b) After measuring the center qubit in the X basis, it is possible that three term X stabilizers and X_L stabilizers with negative sign are created (potential locations indicated by shaded triangles). (c) All signs can be corrected by applying the appropriate single-qubit Z operators and chains of Z operators. (d) Measuring and possibly correcting the indicated Z stabilizer using a bit flip on the center qubit completes the movement of the defect.
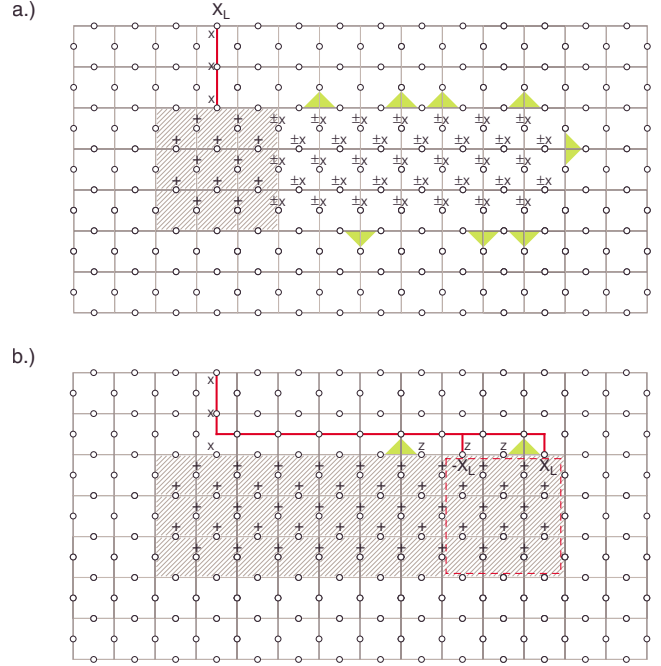

FIG. 14. (Color online) (a) Movement of a large smooth defect via many measurements in the X basis. Many pairs of three term X stabilizers with negative sign are likely to be created (indicated by shaded triangles). (b) After several rounds of error correction, it becomes exponentially unlikely that three term X stabilizers with negative sign remain that were generated in the measurement round. New chains of errors on the boundary can occur, but these will be corrected during normal error correction after the size of the defect is reduced to complete the movement.

and the path. We have effectively deformed the shape of the Z_L stabilizer without changing its sign. The movement of the defect can be completed by measuring the Z stabilizer indicated in Fig. 12(c) and possibly correcting the sign of the result by applying an X operator to the center qubit. By repeating this process we can see that moving a smooth defect deforms the shape of Z_L stabilizers passing nearby.

Consider Fig. 13(a). This shows a smooth defect and three X_L stabilizers. Measuring the center qubit in the X basis as before, we see in Fig. 13(b) that this has potential side effects, with a negative eigenvalue indicating the creation of three term negative X stabilizers and X_L stabilizers of

changed sign. As shown in Fig. 13(c), the measured qubit or qubits in the case of a larger defect are individually phase flipped to ensure they are all in the +1 eigenstate. Pairs of three term negative X stabilizers are corrected with chains of phase flips along the boundary of the defect. Note that this also corrects any X_L stabilizers of changed sign. Figure 13(d) shows the effect of completing the movement of the defect by measuring the appropriate Z stabilizer. With the signs of the X_L stabilizers appropriately corrected, all X_L stabilizers attached to the defect remain attached to the defect with unchanged sign. By repeating this process we can see that moving a smooth defect drags around X_L stabilizers attached to it.

At first glance, the procedure described in the previous paragraph does not appear to be fault tolerant as it seems to rely on perfect measurement and correction of single qubits. Indeed, the procedure is not fault tolerant unless a larger defect is used as shown in Fig. 14(a). After measuring a region of qubits in the X basis and using the individual measurements to give the sign of the X stabilizers across the entire measured region by taking their local parity, we use phase flips to reset them to the +1 eigenstate as best as we are able. We do not assume that we achieve this perfectly. Resetting helps simplify the later incorporation of these qubits into the surface code.

The three term negative X stabilizers on the boundary left over after resetting are again treated as syndrome changes and corrected using the procedure outlined in Sec. III and

described in more detail in Sec. VIII with the exception that if the procedure suggests connecting to syndrome changes on the boundary, either new or old, the direction of the chain of operators correcting these changes is chosen such that the minimum number of sites on the old (reliable) boundary are changed. By doing this, every round error correction makes it exponentially less likely that the three term negative X stabilizers created during the measurement step in a fixed unit length of boundary still remain. This implies that the number of rounds of error correction required to achieve a fixed probability of no errors remaining from the measurement step only grows logarithmically with the length of boundary. Note, however, that during the correction procedure new errors can occur. The primary desirable feature of these new errors is that they are unlikely to form very long chains.

Figure 14(b) shows a potential challenge when it is time to shrink the size of the defect and complete its movement. It is possible for a pair of errors to remain with one error on the boundary of the region of defect about to be healed and the other error on the boundary of the region of defect to remain. As shown, without correction, this would result in X_L stabilizers with sign dependent on where they attach to the moved defect—a situation that is not allowed. However, by measuring all appropriate X stabilizers outside and on the boundary of the final position of the defect (indicated by a dashed line) before measuring all of the Z stabilizers outside the final position, the presence of these two errors is preserved and subsequent correction by a chain of Z operators ensures that the sign of all deformed X_L stabilizers is the same.

To summarize the smooth defect movement procedure, a region of qubits is measured in the X basis, corrected as best as possible so that each measured qubit is in the +1 eigenstate, which takes constant time. Several rounds of error correction are then applied until it is sufficiently likely that only new errors occurring after the initial measurements are now present on the boundary of the new region, which takes a time that grows logarithmically in the length of the boundary, then measurement of all X stabilizers outside and on the boundary of the desired final position of the defect, measurement of all Z stabilizers outside the final defect position, and finally error correction until it is sufficiently likely that only errors occurring after the Z stabilizers were measured remain, which takes a time that grows logarithmically in the area being corrected. This movement procedure deforms nearby Z_L stabilizers and drags around X_L stabilizers attached to the defect and takes a total time that grows only logarithmically in the distance the defect is moved.

Now that we have a thorough understanding of the effect of moving a smooth defect, we can return to the question of how to build a logical CNOT. Any gate can be completely specified by stating its action on computational basis states and can be specified up to global phase by stating its action on a basis of stabilizers. Specifically, if we have a system of two qubits and denote the CNOT between them with the first qubit as the control as Λ_{12}, by simple matrix multiplication we can show that the following relationships hold:

$$\Lambda_{12}(I \otimes X)\Lambda_{12}^{\dagger} = I \otimes X, \qquad (6)$$
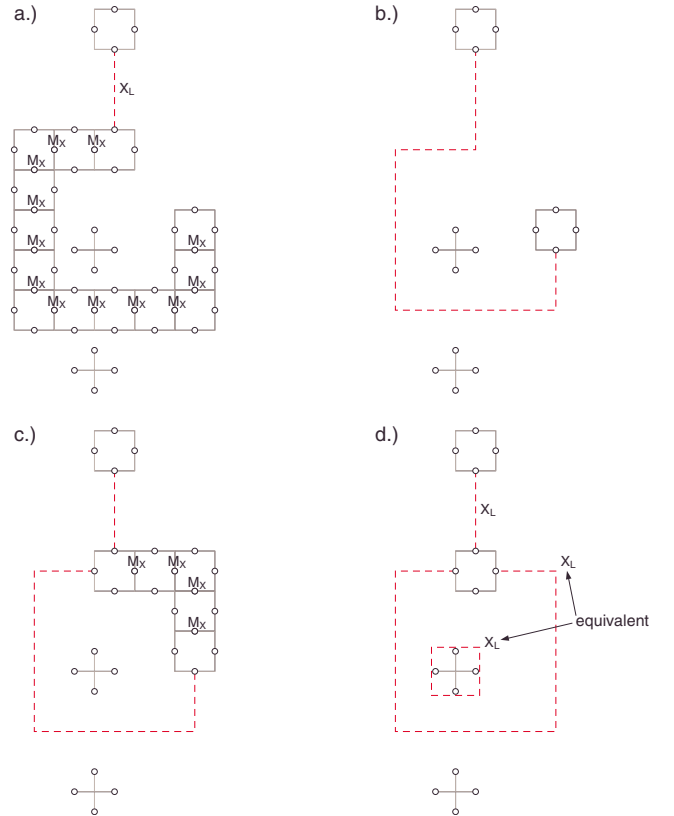

FIG. 15. (Color online) (a) Surface containing a smooth qubit in the +1 eigenstate of X_L and a rough qubit. The lower smooth defect has been braided around the upper rough defect using X measurements. Note that is not possible to complete the braiding in one step as a ring of X measurements corresponds to measurement of the rough qubit in the X_L basis. (b) Via correction of many Z stabilizers, the X_L operator is dragged around the upper rough defect. (c) Additional X measurements extend the defect back to its original position. (d) Further correction of Z stabilizers returns the defects to their original positions but the surface is now in the +1 eigenstate of both the smooth and rough X_L operators.

$$\Lambda_{12}(X \otimes I)\Lambda_{12}^{\dagger} = X \otimes X, \qquad (7)$$

$$\Lambda_{12}(I \otimes Z)\Lambda_{12}^{\dagger} = Z \otimes Z, \qquad (8)$$

$$\Lambda_{12}(Z \otimes I)\Lambda_{12}^{\dagger} = Z \otimes I. \qquad (9)$$

These relationships can be combined to determine the action of CNOT on an arbitrary two-qubit stabilizer. To show that we have a logical CNOT, it is sufficient to show that we can transform logical stabilizers in the above manner. Figures 15 and 16 show that the logical versions of Eqs. (7) and (8) hold if we use a smooth qubit as the control and a rough qubit as the target and braid one of the smooth defects around one of the rough defects. It is not important in which direction the braiding is done, only that the defect returns to its initial position. It is also not important which smooth defect is moved nor which rough defect it is braided around. It is similarly straightforward to show that Eqs. (6) and (9) hold.

We do not yet have what we truly need—a CNOT between logical qubits of the same type. Define M_X, M_Z to take the
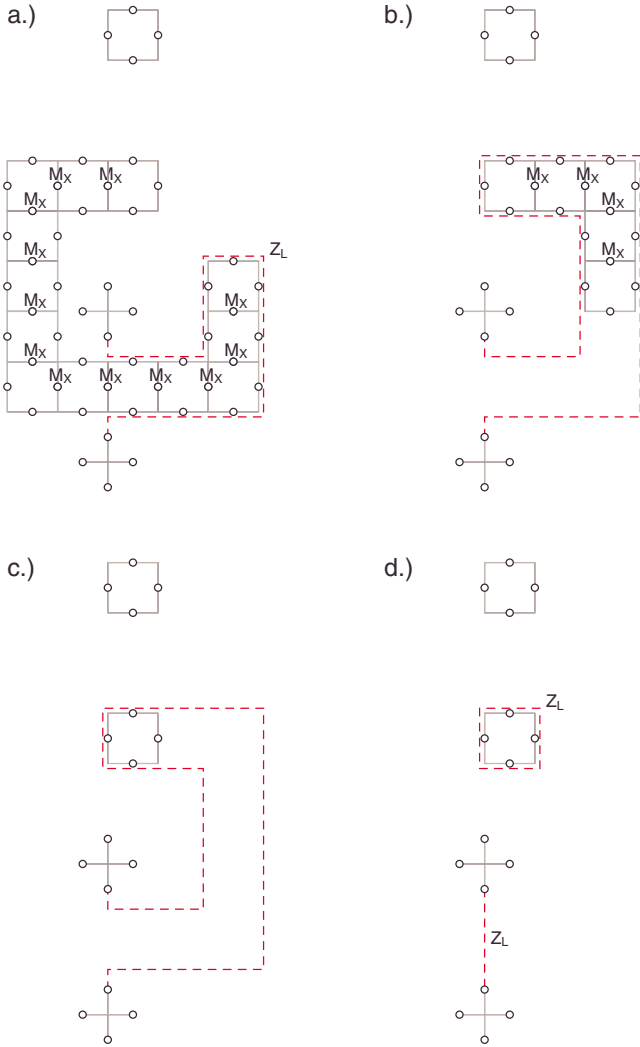
FIG. 16. (Color online) (a) Surface containing a smooth defect and a rough defect in the +1 eigenstate of Z_L. The lower smooth defect has been braided around the upper rough defect using X measurements, deforming the shape of the rough Z_L operator. (b) By first correcting many Z stabilizers and then performing further X measurements, the smooth defect can be extended back to its original position. (c) A final round of Z stabilizer correction returns the defects to their original configuration but with the state of the surface changed. (d) The Z_L operator shown in part (c) is equivalent to the tensor product of smooth and rough Z_L.

value 0 when the +1 eigenstate is measured and 1 when the −1 eigenstate is measured. Consider Fig. 17(a). This is built entirely out of logical circuit elements described above and is equivalent to Z^{M_X} on the target qubit followed by CNOT followed by X^{M_Z} on the target qubit. This is in turn equivalent to CNOT followed by $(Z \otimes Z)^{M_X}$ followed by X^{M_Z} on the target qubit. We will adopt the policy of applying corrective logical operations based on the measurement results immediately after such a CNOT to simplify the discussion of more complicated circuits. Figure 17(a) can also be represented as a braiding of defects of different types in two dimensions of space and one dimension of time as shown in Fig. 17(b) and simplified in Fig. 17(c). Note that since logical CNOT is built out of defect movement, which takes a time that grows loga-
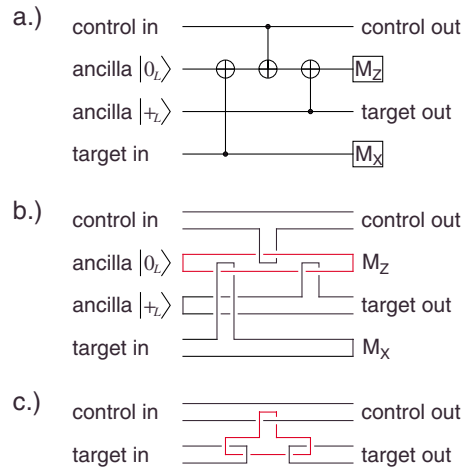

FIG. 17. (Color online) Smooth qubits are represented by black lines and rough qubits by lighter lines. (a) Circuit equivalent to Z^{M_X} on the target qubit followed by CNOT between the control and target qubit followed by X^{M_Z} on the target qubit. (b) Schematic representing the initialization, braiding, and measurement of defects in a surface code to implement (a). Time runs from left to right, and the surface code should be imagined oriented vertically and into and out of the page. (c) Simplified schematic equivalent to (b).

rithmically in the distance moved, and defect measurement, which takes constant time, the total time required to execute logical CNOT grows only logarithmically in the separation of the logical qubits.

VI. STATE INJECTION AND NON-CLIFFORD GATES

The set of gates discussed so far is not universal. To complete the universal set, we will first describe how it is possible to non-fault-tolerantly prepare arbitrary logical states and then discuss state distillation [17,18] and non-Clifford gates based on these distilled states.

A. State injection

Consider Fig. 18. We will focus on the numbered qubits and the four stabilizers, $X_1 X_2 X_3 X_5$, $X_5 X_7 X_8 X_9$, $Z_2 Z_4 Z_5 Z_7$, $Z_3 Z_5 Z_6 Z_8$, centered on qubit 5. The discussion of this section applies to a surface of arbitrary size—we shall see that none of the necessary manipulations affects stabilizers further away. We shall only explicitly work through the creation of an arbitrary rough qubit—the procedure for creating an arbi-
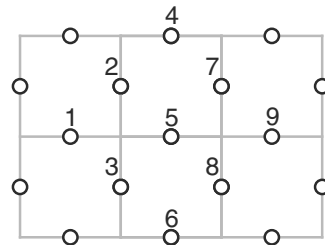

FIG. 18. Surface code fragment and numbered qubits used to assist the visualization of the discussion of Sec. VI A.

trary smooth qubit can be obtained by exchanging the roles of X and Z.

To create an arbitrary rough qubit, begin by measuring qubit 5 in the X basis. This gives a state stabilized by

$$(-1)^{M_X} \begin{array}{|ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & X & & & & \\ & & & & X & & X & X & X \\ & & & & X & & & & \\ & Z & Z & Z & & & Z & Z & Z \end{array} \qquad (10)$$

If the -1 eigenstate is obtained, apply either $Z_2Z_4Z_5Z_7$ or $Z_3Z_5Z_6Z_8$ to create the $+1$ eigenstate,

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & & & & & \\ & & & & & & X & X & X \\ & & & & X & & & & \\ & Z & Z & Z & & & Z & Z & Z \end{array} \qquad (11)$$

Next, Hadamard transform (for pedagogical clarity) and then unitarily rotate qubit 5 to the desired state,

$$\alpha \left(\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & & & & & \\ & & & & & & X & X & X \\ & & & & Z & & & & \\ & Z & Z & Z & & & Z & Z & Z \end{array} \right)$$

$$+ \beta \left(\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & & & & & \\ & & & & & & X & X & X \\ & & & & -Z & & & & \\ & Z & Z & Z & & & Z & Z & Z \end{array} \right). \qquad (12)$$

Measure either $Z_2Z_4Z_5Z_7$ or $Z_3Z_5Z_6Z_8$,

$$\alpha \left(\begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline & X & X & X & & & & X & X & X \\ & & & & & Z & & & & \\ (-1)^{M_Z} & & Z & & Z & Z & & Z & & \\ (-1)^{M_Z} & & & Z & & Z & Z & & Z & \end{array} \right)$$

$$+ \beta \left(\begin{array}{c|ccccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline & X & X & X & & & & X & X & X \\ & & & & & -Z & & & & \\ (-1)^{M_Z} & & Z & & Z & Z & & Z & & \\ (-1)^{M_Z} & & & Z & & Z & Z & & Z & \end{array} \right). \qquad (13)$$

If the -1 eigenstate of $Z_2Z_4Z_5Z_7$ and $Z_3Z_5Z_6Z_8$ is obtained, apply X_5 and then either $X_1X_2X_3X_5$ or $X_5X_7X_8X_9$ to give the desired logical state,
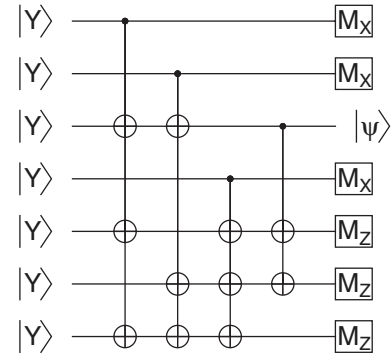

a.)
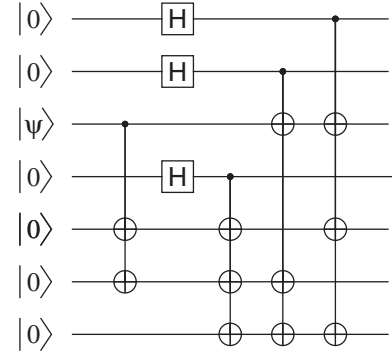
b.)

FIG. 19. (a) Encoding circuit for the seven-qubit Steane code. (b) Distillation circuit for the $|Y\rangle = |0\rangle + i|1\rangle$ state.

$$\alpha \left(\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & & & X & X & X \\ & & & & Z & & & & \\ & Z & & Z & Z & & Z & & \\ & & Z & & Z & Z & & Z & \end{array} \right)$$

$$+ \beta \left(\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline X & X & X & & & & X & X & X \\ & & & & -Z & & & & \\ & Z & & Z & Z & & Z & & \\ & & Z & & Z & Z & & Z & \end{array} \right). \qquad (14)$$

After creating an arbitrary logical qubit using the procedure above, the two halves of the logical qubit would be both moved apart and made larger as quickly as possible to make the logical qubit fault tolerant.

B. State distillation

For our purposes, we are interested in the injection of two particular states $|Y\rangle = |0\rangle + i|1\rangle$ and $|A\rangle = |0\rangle + e^{i\pi/4}|1\rangle$. These states have very special properties. Consider Fig. 19(a). This takes an arbitrary state $|\psi\rangle$ and six ancilla qubits initialized to $|0\rangle$ and creates a logical qubit protected by the seven-qubit

TABLE I. Possible measurement patterns after running the distillation circuit of Fig. 19(b) with perfect $|Y\rangle$ states and no gate errors.

$Pr(M)$	M_X	M_X	M_X	M_Z	M_Z	M_Z	$|\Psi\rangle$
0.125	0	0	0	0	0	0	$Z|Y\rangle$
0.125	0	0	1	1	1	1	$Z|Y\rangle$
0.125	0	1	0	1	0	1	$|Y\rangle$
0.125	0	1	1	0	1	0	$|Y\rangle$
0.125	1	0	0	0	1	1	$|Y\rangle$
0.125	1	0	1	1	0	0	$|Y\rangle$
0.125	1	1	0	1	1	0	$Z|Y\rangle$
0.125	1	1	1	0	0	1	$Z|Y\rangle$

Steane code [19]. A peculiar property of this encoding circuit is that if it is run backward with seven states approximately equal to $|Y\rangle$ as in Fig. 19(b), the output $|\psi\rangle$ will be closer to $|Y\rangle$. This is state distillation [17,18]. Repeated multiple times, arbitrarily high fidelity $|Y\rangle$ states can be obtained exponentially quickly. Specifically, if the input states each have a probability p of X, Y, or Z error, the output state will have probability $7p^3$ of error [2].

Some technicalities exist surrounding the actual values of the measurements indicated in Fig. 19(b). If perfect $|Y\rangle$ states are input, Table I summarizes the possible measurement patterns, their probabilities, and the output state. It can be seen that in some cases a corrective Z operator needs to be applied to the output. If less than perfect $|Y\rangle$ states are input, other measurement patterns have nonzero probability. If a measurement pattern not listed in Table I is obtained, the distilled state is discarded. Note that for high fidelity input states the probability of obtaining an unacceptable measurement pattern is asymptotically zero; thus few distilled states need to be discarded.

A very similar distillation circuit exists for the $|A\rangle$ state. Figure 20(a) shows the encoding circuit for the 15-qubit Reed-Muller code [20]. Running this backward yields the distillation circuit shown in Fig. 20(b). As before, given per-

fect input states, only certain measurement patterns are possible and all measurement patterns result in the desired $|A\rangle$ state, though this time up to an X, Y, or Z operator. Convergence is similarly rapid with error probability p input states yielding an error probability $35p^3$ output state [17].

Note that both Figs. 19(b) and 20(b) are made of operations that can be performed easily and efficiently using the surface code. In particular, the single control multiple target CNOT can be implemented in the same amount of time as a single CNOT. The input $|Y\rangle$ and $|A\rangle$ states would be created factory style, with any errors detected early in the non-fault-tolerant process of their creation resulting in a restart of the creation process. Logical ancilla states that are likely to be sufficiently good would then be recursively fed into logical surface code versions of Figs. 19(b) and 20(b) until sufficiently high fidelity ancilla states are obtained.

C. Non-Clifford gates

Given states of the form $(|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$, rotations $R_Z(\theta)$ and $R_X(\theta)$ can be performed using the circuits shown in Figs. 21(a) and 21(b), respectively. Note that both of these circuits are probabilistic and actually perform rotations $XR_Z(-\theta)$ and $ZR_X(-\theta)$ if the measurement indicates a negative eigenstate. If we wish to apply $R_Z(\pi/2)$ or $R_X(\pi/2)$ and discover we have actually applied rotations $XR_Z(-\pi/2)$ or $ZR_X(-\pi/2)$, the correct gate can be achieved simply by a subsequent application of Z and X. If attempting $R_Z(\pi/4)$ and we discover we have applied $XR_Z(-\pi/4)$, an ancilla state $|0\rangle + i|1\rangle$ needs to be ready for an attempt to apply $R_Z(\pi/2)X$. If we again measure a negative eigenstate, subsequent application of ZX gives the desired rotation.
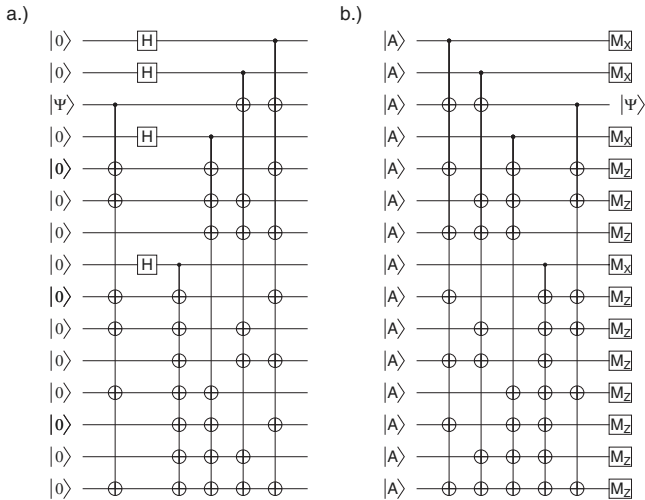

FIG. 20. (a) Encoding circuit for the 15-qubit Reed-Muller code. (b) Distillation circuit for the $|A\rangle = |0\rangle + e^{i\pi/4}|1\rangle$ state.
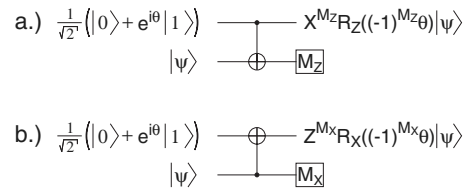

FIG. 21. (a) Circuit performing the single-qubit unitary $X^{M_Z}R_Z[(-1)^{M_Z}\theta]$ given an appropriate ancilla state. (b) Circuit performing the single-qubit unitary $Z^{M_X}R_X[(-1)^{M_X}\theta]$ given an appropriate ancilla state.
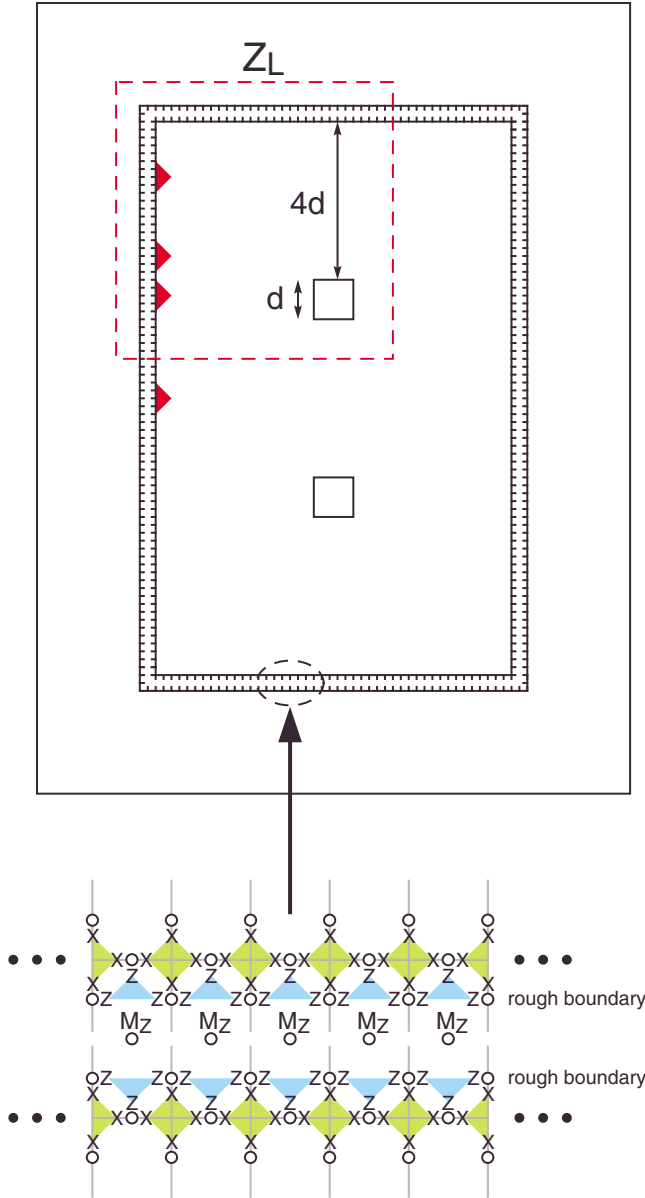
FIG. 23. Circuit showing how an additional syndrome qubit (top line of each figure) is used to measure (a) Z stabilizers and (b) X stabilizers.

FIG. 22. (Color online) A smooth qubit isolated from a larger piece of surface code using a ring of Z measurements so that logical Hadamard can be applied by local transversal Hadamard gates. Shaded triangles in the upper part of the figure represent three term Z stabilizers of negative sign. See text for details.

VII. LOGICAL HADAMARD

The Hadamard gate is called for in many quantum algorithms. In principle we could simply use the relation

$$H \equiv R_Z(\pi/4) R_X(\pi/4) R_Z(\pi/4) \qquad (15)$$

and the constructions of Sec. VI. There is, however, a much more efficient way [12].

Consider Fig. 22. This shows a smooth qubit cut out of a larger lattice using Z measurements. Note that the three term Z stabilizers thus created would need to be corrected as they would have random sign. Without correction, the indicated Z_L stabilizer would have random sign after the measure-
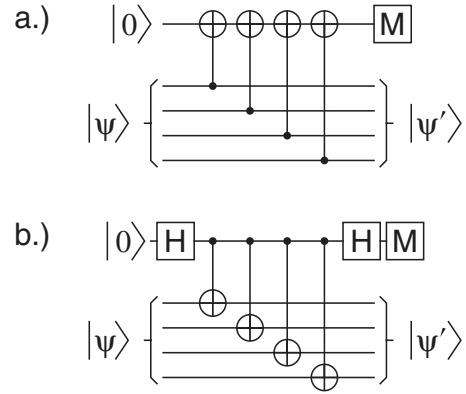
ments. Note that the ring of Z measurements provides no information about the state of the smooth qubit—such a ring is equivalent to the logical identity operator.

The logical Hadamard gate can now be performed transversely. Every face Z stabilizer becomes a vertex X stabilizer. The rough boundary becomes a smooth boundary. The smooth qubit becomes a rough qubit. Stabilizers Z_L and X_L are interchanged. This last point is precisely the action of logical Hadamard.

The interchanging of faces and vertices does create a slight problem—faces and vertices are no longer where they should be. Before connecting the logical qubit to the rest of the lattice, it would need to be moved diagonally in any direction a half lattice spacing. This could be achieved via physical swap gates. After realignment, the complete surface of stabilizers would be measured and corrected once more.

We are still not quite done—our logical qubit has been converted from smooth to rough. By preparing a smooth ancilla qubit in the $|+_L\rangle$ state and performing a simple smooth-rough CNOT followed by measurement of the rough qubit in the Z_L basis and application of X_L if the -1 eigenstate is obtained, we can convert the rough qubit back into a smooth qubit, completing the process. While not completely trivial, this complete process is vastly simpler than the necessary ancilla state preparation and distillation associated with Eq. (15).

VIII. THRESHOLD ERROR RATE

In our simulations we look at a planar square lattice with two smooth and two rough boundaries. This type of a lattice lets us encode one logical qubit. Our general calculation strategy involves preparing the system in the simultaneous $+1$ eigenvalue of all Z and X stabilizers and observing how long it takes for the encoded logical state to change as a result of randomly generating errors.

We have not yet discussed how X and Z stabilizers are actually measured. Figure 23 shows that a fifth syndrome qubit is required to detect whether the state of the surface $|\psi\rangle$ is in the ± 1 eigenstate of a Z or X stabilizer [21]. If the surface $|\psi\rangle$ is in neither eigenstate, the circuit projects the
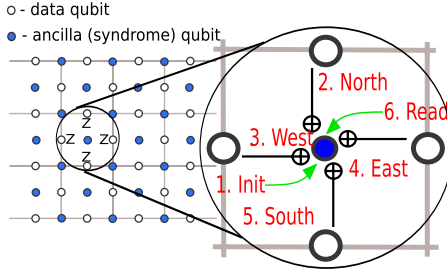
FIG. 24. (Color online) Syndrome measurement typically involves six gates: syndrome initialization, CNOT with the four surrounding data qubits (fewer on boundaries), and finally syndrome readout.

surface into a state $|\psi'\rangle$ that is one of the ± 1 eigenstates. It takes six steps to perform such a measurement. The syndromes are initialized, a CNOT operation is applied between each syndrome and the qubit to the north, west, east, and south, and finally every syndrome qubit is itself measured, as schematically shown in Fig. 24. This order of the CNOT gates has been chosen to ensure that adjacent syndrome circuits sharing a pair of data qubits are strictly ordered—one syndrome circuit touches both data qubits before the other syndrome circuit. CNOT gate orders without this property result in entangled syndrome qubits that provide no useful information for error correction. The placement of a syndrome qubit on each vertex and in the center of each face plus the CNOTs required during the circuit imply that we need a 2D nearest-neighbor coupled lattice of qubits.

The threshold error rate is derived from four error rates in our simulations—initialization error p_i, readout error p_r, memory error p_m, and the error associated with a two-qubit gate p_g. Note that we combine any single-qubit gates with neighboring two-qubit gates and thus do not have a separate single-qubit error rate. All four of these error rates are set to the same value p and all operations are assumed to take the same amount of time to permit our threshold error rate to be compared with others in the literature [2,20,22].

By initialization, we mean initialization to the state $|0\rangle$. An initialization error is therefore accidental preparation of state $|1\rangle$ with probability p. By readout, we mean readout in the Z basis. A readout error is a classical error—the qubit is projected into the ± 1 eigenstate of Z, but with probability p the eigenstate reported by the measurement device is incorrect. A memory error is the application of X, Y, or Z, each with probability $p/3$, to an idle qubit. A two-qubit gate error is the application of one of the 15 nontrivial tensor products of I, X, Y, and Z, each with probability $p/15$, after perfect application of the two-qubit gate.

As was briefly outlined in Sec. III, after each syndrome is read, its value is checked against a result from the previous iteration, and if the values differ, the syndrome change location (in time and space) is recorded. Next, a matching of all the syndrome changes collected up to this point [an example is shown in Fig. 4(a)] is used to guess where errors occurred. Since shorter error chains are more likely than longer ones, we use a minimum weight matching algorithm to do this [14]. Before the matching algorithm can find a minimum weight solution, we convert all the syndrome change results

into a graph, with locations of the syndrome changes representing the nodes and edges between these nodes having a weight which depends on the distance between them. The edge weight is measured in faces along the spatial dimensions and syndrome extraction cycles along the time dimension.

We once again stress that some error chains may begin at the boundary and end somewhere inside the lattice (see Fig. 5). In such cases, we can only observe the syndrome change on the interior of the lattice. To account for this (meaning enable the matching algorithm to guess that the error chain started on a boundary), for every interior node, we always create a closest boundary node. The edges between different boundary nodes are set to be of weight zero. One way to prepare our graph would be to include an edge between every pair of nodes (since in principle we do not know where actual errors occurred), but in practice this is not necessary. Only edges that connect nodes which are not further from each other than the sum of the weights between each node and their closest boundary nodes are included, since nodes that are further apart will always be matched with their respective closest boundaries in preference to each other.

We further optimize graph creation by noting that matches which are temporally far behind the current time step are unlikely to be modified by recent syndrome changes and therefore can be "remembered" from previous iterations. These techniques let us minimize the size of the graph that is passed to the matching algorithm which, despite scaling polynomially in the number of edges and nodes, can often still take substantial computing time. An example of a successful minimum weight perfect match is shown in Fig. 4(b).

As outlined at the beginning of this section, in order to know if the simulation should continue or not, we need to determine whether the lattice suffered a logical error (and hence the encoded state has changed). A logical error corresponds to a chain of errors that starts on one boundary and ends on the opposite one. In order to detect if a logical error has occurred, we repeat the readout cycle with all the error sources set to zero (i.e., set $p_i=p_r=p_m=p_g=0$, in other words have a "perfect readout"). This allows us to be certain that any logical $Z(X)$ error can be recognized by solely checking if the parity of $Z(X)$ operators crossing a vertical (horizontal) line of qubits is odd. A simple example of this is shown in Fig. 25. If no logical error is detected, we revert the simulation state to what it was just before the perfect readout cycle was executed and continue on.

During every run, we note how many syndrome extraction cycles it took for a logical error to be observed. The simulation is repeated many times for different lattice sizes and values of the physical error rate p. All of this data is then used to calculate the average number of steps until a logical error occurs for a given lattice size and p. The graph of Fig. 26 shows the obtained results. In it we see a log-log plot of the average time until failure versus the physical error rate p for lattice sizes ranging from 4 to 20 faces across. We observe a crossing at approximately $p \approx 6.0 \times 10^{-3}$, which is our numerical threshold. If the physical error rate is below this threshold value, the average number of readout cycles until failure can be increased arbitrarily by increasing the distance of the code (lattice size).
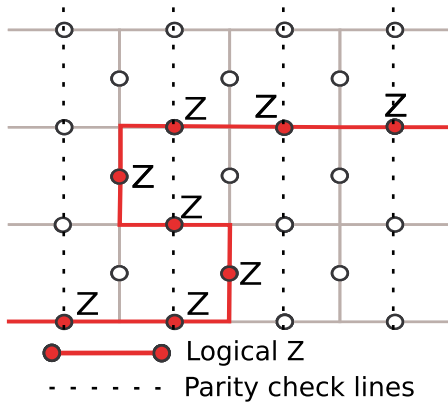
FIG. 25. (Color online) Logical $Z(X)$ error detection involves checking if the parity of $Z(X)$ operators along any of the vertical (horizontal) lines of qubits is odd. Above, we show this in an example of a logical Z error.

IX. DISTRIBUTED COMPUTING

In this section, we show that distributed quantum computing can be performed in a natural manner. For our purposes, a distributed quantum computer will consist of a number of separate rectangular lattices of qubits each capable of holding at least two logical qubits. Computing shall proceed by first moving logical qubits that need to interact onto a common plate before attempting the logical interaction. The movement of logical qubits from one plate to another is the only additional capability we need to discuss.

Consider Fig. 28(a). This shows a plate containing a rough qubit and an empty plate. Note that rough defects do not need to be kept very well separated from smooth boundaries as no error chain can link a rough defect with a smooth boundary. Figure 27 shows the minimum permissible separation from long straight boundaries and corners. Rough defects do, however, still need to be kept well separated from each other.

To move the rough qubit from one plate to the other, it must be possible to perform remote gates between either two complete edges or a smaller section of two edges if the plates
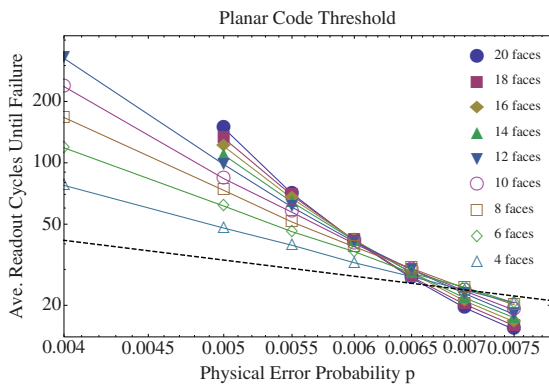

FIG. 26. (Color online) A plot of average time until failure versus the physical error rate p. A threshold is observed at $p \approx 6.0 \times 10^{-3}$ where the curves of different lattice sizes cross. The case where no error correction is used (single qubit) is represented by a dashed line.
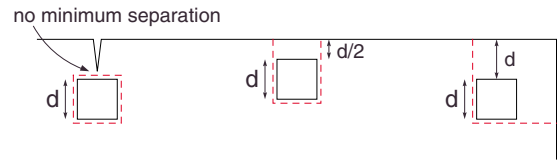

FIG. 27. (Color online) Potential X error chains (dashed lines) around a rough defect and consequent minimum separation from long straight smooth boundaries and two types of corners such that X error chains beginning and ending on the smooth boundary are not more likely than an X error ring around the rough defect.

are large relative to the size of a logical qubit. Generally speaking, implementing remote gates would be expected to involve entanglement distribution and purification [23]. We will not discuss the details here besides mentioning that this leads to significant qubit and gate overhead implying remote gates should be kept to a minimum.

Consider Fig. 28(b). This shows the pairs of qubits, including syndrome qubits, that need to be remotely interacted to enable a single round of the error correction to proceed seamlessly across the two plates. Note that one column of qubits on the empty plate has been omitted as though it is idle, but note that this figure does not include the qubits required for entanglement purification and it is unlikely that there would be idle qubits on the boundary in practice. In general, the joined plates will be in a random eigenstate of
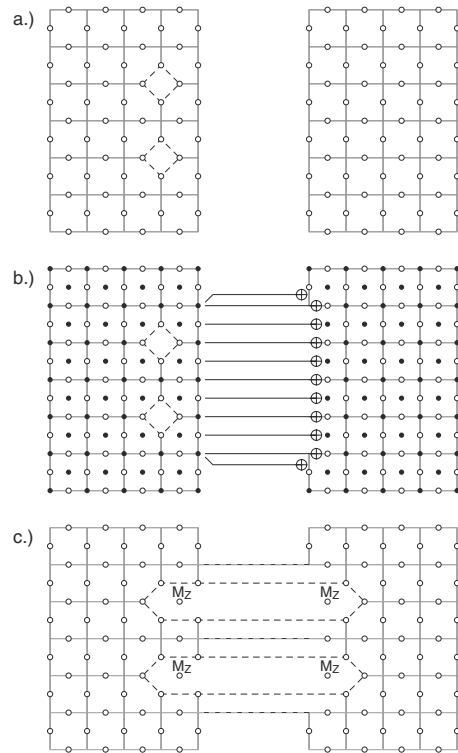

FIG. 28. (a) A rough qubit ready to be sent to a separate piece of surface code. (b) Remote gates are used to join the two surfaces together. (c) A sequence of measurements is used to move the rough qubit. After the necessary correction associated with completing the movement, the long-range gates can be discontinued to separate the two pieces of surface code once more.

both the X and Z stabilizers straddling both plates. We shall treat these random values as errors and correct them.

After correction of the join, the rough qubit can be moved over to the other plate via Z measurements as shown in Fig. 28(c). First the border Z stabilizers of this extended defect would need to be corrected as discussed in Sec. V, then, when shrinking the size of the defects to move the logical qubit, the unneeded regions of X stabilizers measured and corrected once more. Both of these correction procedures take a number of time steps that only grow logarithmically with the size of the computation and the length or area being corrected. After the necessary correction has been completed, error correction can continue on each plate individually without any further long-range interactions.

The most common reason to move a logical qubit from one plate to another would be to perform a remote CNOT. This would be achieved by creating a rough qubit on the control plate, braiding it around the control qubit, sending the rough qubit to the target plate, and completing the necessary braiding and measurement operations entirely on the target plate.

X. CONCLUSION AND FURTHER READING

We have presented a simplified yet comprehensive review of the 2D version of the quantum computation scheme originally presented in [1,2]. We started with a description of the surface code, as well as the stabilizer formalism which is used throughout this paper. We discussed in detail logical state initialization, logical CNOT, and non-Clifford group gates, which make use of state distillation. We calculated a numerical threshold for the surface code and obtained a value of $p \approx 6.0 \times 10^{-3}$ which is commensurate with other calculations in the literature [2,20,22].

[1] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).

[2] R. Raussendorf, J. Harrington, and K. Goyal, New J. Phys. **9**, 199 (2007).

[3] F. Helmer, M. Mariantoni, A. G. Fowler, J. von Delft, E. Solano, and F. Marquardt, EPL **85**, 50007 (2009).

[4] D. DiVincenzo, e-print arXiv:0905.4839.

[5] R. Van Meter, T. D. Ladd, A. G. Fowler, and Y. Yamamoto, Int. J. Quantum Inf. (to be published); e-print arXiv:0906.2686.

[6] A. G. Fowler and K. Goyal, Quantum Inf. Comput. **9**, 721 (2009).

[7] S. J. Devitt, A. D. Greentree, R. Ionicioiu, J. L. O'Brien, W. J. Munro, and L. C. L. Hollenberg, Phys. Rev. A **76**, 052312 (2007).

[8] S. J. Devitt, A. G. Fowler, A. M. Stephens, A. D. Greentree, L. C. L. Hollenberg, W. J. Munro, and K. Nemoto, New. J. Phys. **11**, 083032 (2009).

[9] R. Stock and D. F. V. James, Phys. Rev. Lett. **102**, 170501 (2009).

[10] D. Gottesman, Ph.D. thesis, Caltech, 1997.

[11] S. B. Bravyi and A. Y. Kitaev, e-print arXiv:quant-ph/9811052.

[12] H. Bombin and M. A. Martin-Delgado, J. Phys. A: Math. Theor. **42**, 095302 (2009).

[13] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[14] W. Cook and A. Rohe, INFORMS J. Comput. **11**, 138 (1999).

[15] S. J. Devitt, A. Fowler, T. Tilma, W. J. Munro, and K. Nemoto, e-print arXiv:0906.0415.

[16] G. Duclos-Cianci and D. Poulin, e-print arXiv:0911.0581.

[17] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).

[18] B. W. Reichardt, Quantum Inf. Process. **4**, 251 (2005).

[19] A. M. Steane, Proc. R. Soc. London, Ser. A **425**, 2551 (1996).

[20] R. Raussendorf, J. Harrington, and K. Goyal, Ann. Phys. **321**, 2242 (2006).

[21] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).

[22] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, e-print arXiv:0905.0531.

[23] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Phys. Rev. Lett. **76**, 722 (1996).