

## Efficient one-way quantum computations for quantum error correction

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2009 J. Phys. A: Math. Theor. 42 295301

(<http://iopscience.iop.org/1751-8121/42/29/295301>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 171.66.16.155

The article was downloaded on 03/06/2010 at 07:59

Please note that [terms and conditions apply](#).

# Efficient one-way quantum computations for quantum error correction

Wei Huang<sup>1</sup> and Zhaohui Wei<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

<sup>2</sup> State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China

E-mail: [weihuang@eecs.umich.edu](mailto:weihuang@eecs.umich.edu)

Received 4 August 2008, in final form 7 May 2009

Published 30 June 2009

Online at [stacks.iop.org/JPhysA/42/295301](http://stacks.iop.org/JPhysA/42/295301)

## Abstract

We show how to explicitly construct an  $O(nd)$  size and constant quantum depth circuit which encodes any given  $n$ -qubit stabilizer code with  $d$  generators. Our construction is derived using the graphic description for stabilizer codes and the one-way quantum computation model. Our result demonstrates how to use cluster states as scalable resources for many multi-qubit entangled states and how to use the one-way quantum computation model to improve the design of quantum algorithms.

PACS numbers: 03.67.-a, 03.67.Lx, 03.67.Pp

## 1. Introduction

The *one-way quantum computation* (1WQC) model [1–3], due to its simplicity, universality and parallelism, is widely considered as a very promising scheme for the experimental development of a quantum computer [4–10].

The 1WQC model starts with a highly entangled cluster state and performs quantum computing simply by a sequence of adaptive single-qubit measurements and post-measurement local corrections. Thus the whole computation is separated into four parts: (1) preparing cluster states, (2) performing single-qubit measurements, (3) classically processing measurement outcomes and (4) performing post-measurement local unitary corrections.

Such a simple model has been proved to be universal for quantum computation since any quantum circuit can be efficiently simulated on it. Moreover, by translating normal quantum circuits into 1WQC-compatible circuits, it is possible to reduce circuit depth and increase parallelism, which is critical to overcome the quantum decoherence problem [11, 12].

In this paper, we reproduce the previous encoding procedure for quantum error correction [13] under the one-way quantum computation model. Using only  $O(nd)$  single-bit operations

and a small number of two-bit measurements, we encode any given  $n$ -qubit stabilizer code with  $d$  generators. Furthermore, we will show that the depth of our construction is constant. Our construction is derived using the graphic description for stabilizer codes.

This paper is organized as follows: in section 2, we review the connections between graph states and stabilizer codes. In section 3, we produce an  $O(nd)$  size and constant-depth 1WQC-compatible circuit for the encoding and decoding procedure of arbitrary graph codes.

## 2. Preliminaries

In this section, let us recall some basic notions concerning this paper. More details can be reviewed in [14, 15].

### 2.1. Stabilizer codes, graph codes and graph states

The *Pauli group*  $\mathcal{P}_n$  on  $n$  qubits is defined to consist of  $n$ -fold tensor products of the Pauli matrices  $\{I, X, Y, Z\}$

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

with multiplicative factors  $\{\pm 1, \pm i\}$ .

The *Clifford group*  $\mathcal{C}_n$  is defined as the normalizer of the Pauli group  $\mathcal{P}_n$ ,

$$\mathcal{C}_n = \{U \in SU(2^n) \mid \forall P \in \mathcal{P}_n, UPU^\dagger \in \mathcal{P}_n\}.$$

The *local Clifford group*  $\mathcal{LC}_n = \mathcal{C}_1^{\otimes n}$  is a subgroup of  $\mathcal{C}_n$  which only consists of the tensor products of local unitary operations. The *Clifford group*  $\mathcal{C}_n$  can be generated, up to a global phase factor, by the Hadamard gate  $H$ , the phase gate  $S$  and the CNOT gate, while the local Clifford group  $\mathcal{LC}_n$  can be generated by  $H$  and  $S$  only. There are only 24 elements in  $\mathcal{LC}_1$ , up to a global phase.

*Stabilizer group*  $\mathcal{S}_n$  is an Abelian subgroup of *Pauli group*  $\mathcal{P}_n$  without  $-I$ . Any  $\mathcal{S}_n$  is a stabilizer for a non-trivial vector space, which can be defined as the codespace of a *stabilizer code*. The code words of the Stabilizer code form the +1-eigenspace of the all the operations in  $\mathcal{S}_n$ .

An  $n$ -qubit *Stabilizer code* with  $d$  generators can encode an  $(n - d)$ -qubit state into an  $n$ -qubit state. The stabilizer group  $\mathcal{S}_n$  corresponding to the stabilizer code can be generated by  $d$  independent elements  $g_1, \dots, g_d$ . Other elements in  $\mathcal{S}_n$  can be represented as products of  $g_1, \dots, g_d$ . Thus we can use  $g_1, \dots, g_d$  to describe a stabilizer code. We will use the *binary framework* of stabilizer formalism to represent elements in the stabilizer group efficiently.

Define a homomorphic map from  $\mathcal{P}_1$  to  $\mathbb{Z}_2^2$  as the following:

$$I \rightarrow 00, \quad X \rightarrow 10, \quad Y \rightarrow 11, \quad Z \rightarrow 01.$$

After mapping, an element of Pauli group  $P = P_1 \otimes P_1 \otimes \dots \otimes P_N$  can be described by a binary vector  $[\hat{x}|\hat{z}]$ , where  $\hat{x}$  is the vector consisting of the first bits of  $P_1, \dots, P_n$  while  $\hat{z}$  is the vector consisting of the second bits. Therefore an  $n$ -qubit stabilizer code  $C$  with  $d$  generators  $g_1, \dots, g_d$  can be described by a  $2n \times d$  *generator matrix*:  $[\hat{X}|\hat{Z}]$  where both  $\hat{X}$  and  $\hat{Z}$  are  $n \times d$  binary matrices.

An  $n$ -qubit *stabilizer state*  $|\psi\rangle$  is an  $n$ -qubit stabilizer code with exactly  $n$  generators. In this case, the dimension of the code space is one.  $|\psi\rangle$  is the only vector stabilized by  $n$  generators, up to a global phase. The stabilizer of  $|\psi\rangle$  can be described by a  $2n \times n$  generator matrix. A *graph state*  $|G\rangle$  is a stabilizer state with graphical generator matrix  $[\hat{X}|\hat{Z}] = [I|G]$ , where  $I$  is the identity matrix and  $G$  is the adjacency matrix of the underlying graph of the graph

state. Suppose that an  $n$ -qubit *graph code* with  $d$  generators is generated by the generator matrix  $[\widehat{X}_1|\widehat{Z}_1] = [I, R|A + RC^T, C]$ , then it is closely related to the graph state  $|G\rangle$  stabilized by the generator matrix  $[\widehat{X}_2|\widehat{Z}_2] = \begin{bmatrix} I & 0 & A & C \\ 0 & I & C^T & 0 \end{bmatrix}$ . The adjacency matrix of the underlying graph is  $G = \begin{pmatrix} A & C \\ C^T & 0 \end{pmatrix}$ .

An important result [16–18] is that any stabilizer state can be transformed into a graph state with generator matrix  $[\widehat{X}|\widehat{Z}] = [I|G]$  by a local unitary operation  $U \in \mathcal{L}\mathcal{C}_n$ . Similarly, any  $n$  qubits stabilizer code with  $d$  generators can be transformed into a graph code with generator matrix  $[\widehat{X}_1|\widehat{Z}_1] = [I, R|A + RC^T, C]$ .

### 2.2. Relation between graph states and graph codes

In this subsection, we give an example to demonstrate how to generate graph codes based on graph state. According to the relationship between stabilizer codes and graph codes, the basic idea can be generalized to find the relationship between stabilizer codes and graph states. Suppose we have a graph state  $|G\rangle$  with generators  $\{g_1, g_2, g_3, g_4, g_5, g_6\}$  which is stabilized by the following generator matrix:  $[\widehat{X}_2|\widehat{Z}_2] = \begin{bmatrix} I & 0 & A & C \\ 0 & I & C^T & 0 \end{bmatrix}$

$$\left( \begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right),$$

and a six-qubit graph code with four generators  $\{g'_1, g'_2, g'_3, g'_4\}$  which is stabilized by the following generator matrix:  $[\widehat{X}_1|\widehat{Z}_1] = [I, R|A + RC^T, C]$

$$\left( \begin{array}{cccccc|cccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array} \right),$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad R = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

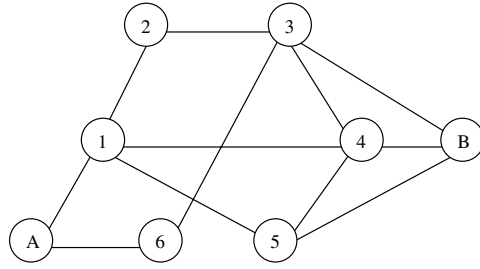
It is not hard to see the following equations relating the two sets of generators:

$$g'_1 = g_1 \cdot g_6, \quad g'_2 = g_2, \quad g'_3 = g_3 \cdot g_5, \quad g'_4 = g_4 \cdot g_5.$$

Based on the above relation between the generators of the graph code and the graph state, we can obtain the graphic representation of the graph code as shown in figure 1.

Let  $G'$  denote the graph which includes  $G$  plus input nodes  $A$  and  $B$ . Let  $|G'\rangle$  denote the graph state corresponding to  $G'$ . Suppose the code words of the graph code are  $\{|00_L\rangle, |01_L\rangle, |10_L\rangle, |11_L\rangle\}$ , then we have

$$|G'\rangle = |0\rangle_A|0\rangle_B|00_L\rangle + |0\rangle_A|1\rangle_B|01_L\rangle + |1\rangle_A|0\rangle_B|10_L\rangle + |1\rangle_A|1\rangle_B|11_L\rangle$$



**Figure 1.** Graphic representation of the graph code in subsection 2.2. Vertices A and B are input nodes.

and

$$\begin{aligned} |00_L\rangle &= |G\rangle \\ |01_L\rangle &= Z_3 Z_4 Z_5 |G\rangle \\ |10_L\rangle &= Z_1 Z_6 |G\rangle \\ |11_L\rangle &= Z_1 Z_3 Z_4 Z_5 Z_6 |G\rangle \end{aligned}$$

where  $Z_i$  denotes local unitary  $Z$  on qubit  $i$ .

Fault-tolerant  $X$  and  $Z$  operations on the first and second qubits of the encoded state are  $(X_L)_1 = Z_1 Z_6$ ,  $(X_L)_2 = Z_3 Z_4 Z_6$ ,  $(Z_L)_1 = g_1 = X_1 Z_2 Z_5$  and  $(Z_L)_2 = g_3 = X_3 Z_2 Z_4 Z_6$ . More details about graph states will be explained in the following section.

**Remark.** If we can construct the uniform encoded state  $\sum_{x \in \{0,1\}^k} |x\rangle |x_L\rangle$ , then we can encode any given unknown  $k$ -qubit state  $|\psi\rangle$  in a stabilizer code by quantum teleportation [15]. In the following section, we focus on using cluster states to generate any graph state including uniform encoded states for graph codes.

### 3. Construction of 1WQC-compatible circuit encoding graph codes

#### 3.1. Preparation of graph states by definition

According to the section above, for the purpose of encoding, we have to prepare the graph state we need. In this subsection, we focus on this topic. Firstly, let us recall the definition of graph states. Actually, this definition itself is a method of creating graph states.

Let  $G = (V, E)$  be a graph with  $n = |V|$  vertices and  $m = |E|$  edges, then graph state  $|G\rangle$  corresponding to the graph  $G$  is the following superposition over all basis states,

$$|G\rangle = \prod_{\substack{i < j \\ (i,j) \in E}} Z_{ij} |+\rangle^{\otimes n} = \sum_{x \in \{0,1\}^n} (-1)^{q(x)} |x\rangle.$$

Here  $Z_{ij}$  denotes the controlled phase gate between qubits  $i$  and  $j$

$$Z_{ij} |+\rangle_i |+\rangle_j = |0\rangle_i |+\rangle_j + |1\rangle_i |-\rangle_j.$$

$q(x)$  is a quadratic function related to the graph  $G$

$$q(x) = \sum_{\substack{i < j \\ (i,j) \in E}} x_i x_j.$$

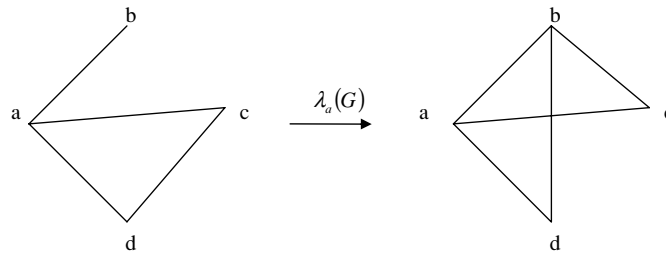


Figure 2. An example of a local complement operation on a graph.

We can verify that  $|G\rangle$  is the stabilizer state with the graphical generator matrix  $[\widehat{X}|\widehat{Z}] = [I|G]$ . Thus we have the following procedure of preparing the graph state  $|G\rangle$  by its natural definition:

- the qubit at each vertex  $v \in V$  has the initial state  $|0\rangle$ ;
- apply the Hadamard gate on each qubit, so each qubit is now in the state  $|+\rangle$ ;
- apply the controlled phase gate  $Z_{ij}$  to each edge  $(i, j) \in E$ .

Actually, cluster states and graph states are used so widely in quantum information processing that their preparation becomes an important issue. Many efforts have been made on this problem. On one hand, it has been shown that cluster states can be grown using a ‘divide-and-conquer’ approach [19–23]. In this approach, bigger cluster states are created by iteratively connecting smaller clusters together.

On the other hand, another scheme for the preparation of cluster states is based on the optical lattice of ultracold atoms [24, 25]. In this proposal, the cluster state can be prepared in one step using a natural nearest-neighbor interaction. Though this is a theoretical proposal at the present time because of the difficulties in experiments, it may be a promising and efficient method of preparing cluster states in the future. In this situation, it seems necessary to propose a general method for preparing arbitrary graph states from 2D cluster states. In the following, we will give such a procedure. Firstly, let us recall some properties about graph states in the following subsection.

### 3.2. Graphical rules of single-qubit Pauli measurements

We start by introducing the concept of *local complement* operations on the graphs and describing some graphical rules of the operations on the graph states.

Letting  $G = (V, E)$  be a graph and  $a \in V$ , the local complement operation  $\lambda_a$  on vertex  $a$  is obtained by complementing the subgraph of  $G$  induced by the neighborhood of  $a$  and leaving the rest of the graph unchanged. Figure 2 depicts an example for the local complement operation.

The local complement operations are closely related to the *graphical rules* of some quantum operations on the graph states.

Let  $[X]_a, [Y]_a$  and  $[Z]_a$  denote single-qubit *Pauli measurements*  $X, Y, Z$  on qubit  $a$ , respectively. After each Pauli measurement, a graph state  $|G\rangle$  will transform into another graph state  $|\widetilde{G}\rangle$ , up to a local Clifford unitary depending on the measurement outcome [26]. The graphical rules for Pauli measurements are the following:

- $[Z]_a$ : deletes vertex  $a$  and related edges from  $G$ ,  $\widetilde{G} = G - a$ .
- $[Y]_a$ : first applies local complement on vertex  $a$ , then deletes vertex  $a$ ,  $\widetilde{G} = \lambda_a(G) - a$ .

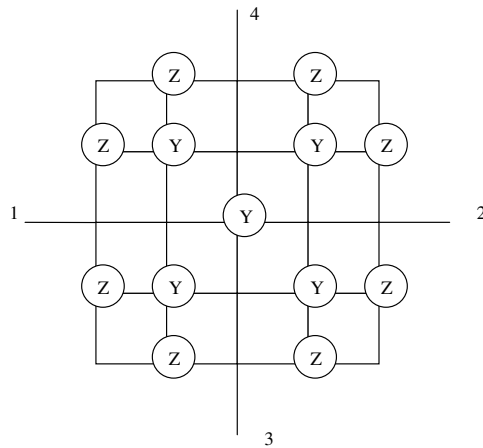


Figure 3. Two chains cross each other without sharing qubits.

- $[X]_a$ : chooses any of  $a$ 's neighbors  $b$ , applies local complement on vertex  $b$ , then applies local complement on  $a$ , deletes vertex  $a$  and applies local complement on  $b$  again,  $\tilde{G} = \lambda_b(\lambda_a \circ \lambda_b(G) - a)$ .

In summary, two graph operations, local complement and vertex deletion, can be achieved by single-qubit Pauli measurements and local Clifford operations. Based on the above simple graphic rules, two more graph operations, crossing and contraction, can be implemented as follows:

**Lemma 3.1.** (Crossing) *Two chains crossing each other without sharing any qubit can be simulated in the cluster state of the 2D lattice.*

**Proof.** In this paper, all the operations are based on cluster states. That is to say, we do not move any qubit during the whole process. As a consequence, we will find that sometimes we need to construct two chains such that they cross each other without sharing any qubit (these kinds of operations are important in section 3). This can be achieved as follows (see figure 3).

In the first step, we perform all  $Z$  measurements and correct the related byproduct local unitaries. Then, we do all  $Y$  measurements except the central one and correct them. Finally, we perform the  $Y$  measurement on the central qubit and correct it. In this way, all the qubits denoted by circles in figure 3 will be deleted by measurements in the end. The result is a chain from 1 to 2 and another chain from 3 to 4. The two chains cross each other without sharing any qubit.  $\square$

**Remark.** One can use the rewrite rules of the measurement calculus [27], to reduce the running time by postponing the local corrections till the end of 1WQC. However it is unnecessary here since the running time of simulating chain crossing is already constant.

**Lemma 3.2.** (Contraction[28]) *Let graph  $G(L-v-a-b-R)$  consist of subgraphs  $L$  and  $R$  and three vertices  $v, a$  and  $b$ . Vertex  $a$  has two edges  $(a,v)$  and  $(a,b)$ . Vertices  $v$  and  $b$  have edges connected to the vertices in the subgraphs  $L$  and  $R$ , respectively. After applying  $X$  measurements on vertices  $a$  and  $b$ , the graph state corresponding to graph  $G(L-v-a-b-R)$  will change to the graph state corresponding to the graph  $\tilde{G}(L-v-R)$ .*

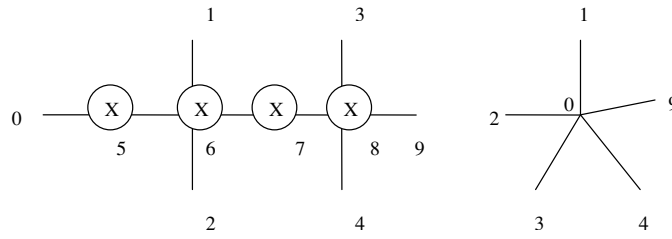


Figure 4. Contraction through a chain.

**Proof.** The lemma can be verified by applying the basic graphic rule about  $[x]_a$ . The contractions through a chain can be done simultaneously. For example in figure 4, if  $X$  measurements are applied on qubits 5–8 at the same time, the graph will contract to the vertex 0, whether or not a  $Z$  operation on qubit 0 is needed for local correction depends on the sum of measurement outcomes of qubits 6 and 8. Local  $Z$  corrections on qubits 1 and 2 depend on the measurement outcome of the qubit 5. Local  $Z$  corrections on qubits 3, 4 and 9 depend on the sum of measurement outcomes of qubits 5 and 7.

To understand why the contractions through a chain can be done simultaneously and how local operations can be postponed to the end of the computing, we have to go through some complicated calculations step-by-step carefully. A detailed proof of the lemma is included in the appendix.  $\square$

### 3.3. Generating arbitrary graph states from the cluster states of the 2D lattice

In this subsection, we discuss how to generate arbitrary graph states from the cluster states of the 2D lattice. We depict this result as the following theorem.

**Theorem 3.1.** Any graph state with the underlying graph  $G$  can be generated from an  $O(n) \times O(n)$  cluster state by local measurements and local unitaries in constant time.

**Proof.** Given a graph  $G = (V, E)$  with  $n$  vertices  $v_1, \dots, v_n$  and  $m$  edges  $E_1, \dots, E_m$ , we need to perform some crossings and contractions on a cluster state of the 2D lattice to generate a graph state  $|G\rangle$ .

We need several auxiliary qubits. A  $5 \times 5$  lattice is required for implementing crossings whereas contractions between any two vertices with degree greater than 2 require degree 2 auxiliary qubits. However, we can introduce those auxiliary qubits by increasing the length and width of the 2D lattice only by a constant value. Thus we ignore those auxiliary qubits and only consider an  $n \times n$  lattice. Besides, to avoid unnecessary complication and make our proof clear, we use the example in figure 5 to illustrate our basic idea. The proof can be generalized to any case directly and easily, thus this simplification is reasonable.

The whole procedure can be divided into three steps. In the first step, we perform  $Z$  measurements on the qubit located at the intersection of the  $i$ th row and the  $j$ th column where  $i < j$  and  $i, j \in \{1, 2, \dots, n\}$ . In the second step, we perform a crossing operation for any qubit on the location  $(i, j)$  which satisfies  $i > j$  and  $(v_i, v_j) \notin E$ . In the final step, we contract simultaneously through the columns 1, 2,  $\dots$ ,  $n$  to the locations  $(1, 1), (2, 2), \dots, (n, n)$ . Then we obtain the target graph state.  $\square$

In fact, since we only need to generate the specific graph states related to the graph codes, by carefully rearranging the protocol in theorem 3.1, it is not difficult to reduce the size of



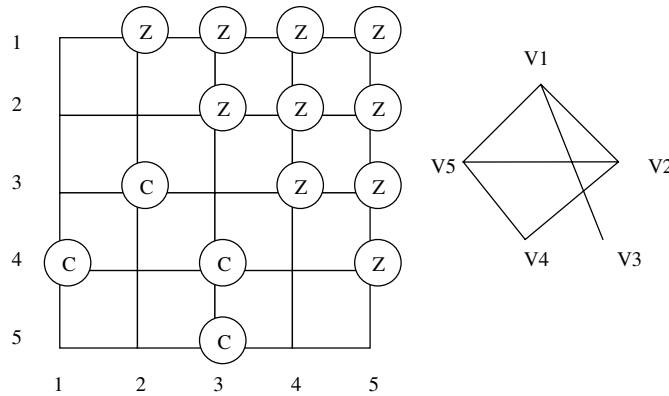


Figure 5. Generating arbitrary graph states from the cluster states of the 2D lattice.

the cluster state needed by our encoding method, to  $O(n) \times O(d)$ . For convenience, we introduced the above one.

### 3.4. Construction of 1WQC encoding graph codes

Now we can turn to the main result of this paper, how to encode stabilizer codes for quantum error correction. By combining these operations mentioned above together, we can realize the encoding algorithm as follows.

Suppose we need to encode an unknown quantum state using an  $n$ -qubit stabilizer code with  $d$  generators. Firstly, we determine the equivalent graph code and the graphic representation of the graph code. Then we build a proper cluster state of the 2D lattice and based on it generate the graph state corresponding to the graphic representation. Finally, we teleport the unknown quantum state onto the graph state by measuring the unknown quantum state and the input nodes of the graph state [15]. Then the remaining part of the graph state is the target encoding state we want, up to some local unitary operations.

## 4. Discussion

According to the previous section, the whole computation which generates the graph state related to any  $n$ -qubit graph code with  $d$  generators can be conducted on an  $O(n) \times O(d)$  lattice. Therefore the total number of quantum operations of the 1WQC is bounded by  $O(nd)$ , which is the length of the description of the generating matrix of the stabilizer code. Therefore, both the size and the depth of our 1WQC are most likely optimal in the general case.

Note that our construction has a constant running time. Since qubit coherent time is limited, improving the temporal overhead of the encoding procedure will be helpful for its physical implementation.

Furthermore, it should be pointed out that in the procedure of preparing graph state based on cluster states, most operations we need are single-qubit operation (except when teleportating an unknown state, where a small amount of two-qubit measurements are involved). Usually, in experiments the fidelity of one-qubit operations is very high. Thus, ignoring errors introduced by one-qubit operations, our encoding procedure will be reasonable as long as the quality of cluster states we use as foundation is good enough.

The decoding procedure can be done in a similar way to the encoding procedure, if one can implement the quantum teleportation on an encoded state. For the error detecting and

fault tolerant computation on the states encoded in the stabilizer codes, one can apply methods in [14].

## 5. Conclusion

In this paper, we have shown how to use one-way quantum computation to implement an encoding and decoding procedure for quantum error correction. We have constructed an  $O(nd)$  size and constant-depth 1WQC-compatible circuit which encodes any given  $n$ -qubit stabilizer code with  $d$  generators. The result demonstrates that the cluster states can be used as the scalable resources for many multi-qubit entangled states and the one-way quantum computation model can help to design better quantum algorithms than the traditional quantum circuit model.

## Acknowledgments

W Huang thanks Y Y Shi and Y-J Han, Z Wei thanks L M Duan and Y-J Han for discussions about graph states. W Huang would also like to thank labmates D R Vandenberg and R Duan for interesting discussions about local complement operations. This work was supported in part by The China Scholarship Council, the NSF awards (0431476), the ARDA under ARO contracts and the A P Sloan Fellowship.

## Appendix

**Proof of Lemma 3.2.** At the beginning, the initial state  $|\psi\rangle$  is the graph state  $|G\rangle$

$$\begin{aligned} |\psi\rangle &= |G\rangle = (|0\rangle + Z_v Z_b |1\rangle)_a (|0\rangle + Z_R |1\rangle)_b (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle \\ &= ((I + Z_v Z_b)|+\rangle + (I - Z_v Z_b)|-\rangle)_a (|0\rangle + Z_R |1\rangle)_b (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle. \end{aligned}$$

Apply  $X$  measurements on qubit  $a$ , let  $x \in \{0, 1\}$  be the measurement result, the remaining state of other qubits is

$$\begin{aligned} |\psi\rangle &= (I + (-1)^x Z_v Z_b) (|0\rangle + Z_R |1\rangle)_b (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle \\ &= ((I + (-1)^x Z_v)|0\rangle + (I - (-1)^x Z_v) Z_R |1\rangle)_b (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle \\ &= ((I + (-1)^x Z_v + Z_R - (-1)^x Z_v Z_R)|+\rangle + (I + (-1)^x Z_v - Z_R + (-1)^x Z_v Z_R)|-\rangle)_b \\ &\quad \times (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle. \end{aligned}$$

Then apply  $X$  measurements on qubit  $b$ , let  $y \in \{0, 1\}$  be the measurement result, let  $Z^0 = I$  and  $Z^1 = Z$ , we have

$$\begin{aligned} |\psi\rangle &= (I + (-1)^x Z_v + (-1)^y Z_R - (-1)^{x+y} Z_v Z_R) (|0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle \\ &= ((1 + (-1)^x)I + (-1)^y (1 - (-1)^x) Z_R) |0\rangle \\ &\quad + ((1 - (-1)^x)I + (-1)^y (1 + (-1)^x) Z_R) Z_L |1\rangle |G_R\rangle |G_L\rangle \\ &= \begin{cases} (|0\rangle + (-1)^y Z_R Z_L |1\rangle)_v |G_R\rangle |G_L\rangle & \text{if } x = 0, \\ ((-1)^y Z_R |0\rangle + Z_L |1\rangle)_v |G_R\rangle |G_L\rangle & \text{if } x = 1. \end{cases} \\ &= Z_v^y Z_R^x (|0\rangle + Z_R Z_L |1\rangle)_v |G_R\rangle |G_L\rangle \\ &= Z_v^y Z_R^x |\tilde{G}\rangle. \end{aligned}$$

□

More generally, we consider the effect of graph contraction on the generalized graph state  $Z_L^{u_0} Z_v^{v_0} Z_R^{w_0} Z_a^{x_0} Z_b^{y_0} |G\rangle$ , where  $\{u_0, v_0, w_0, x_0, y_0\} \in \{0, 1\}$ . In this case, we first apply some

local  $Z$  operations on graph state  $|G\rangle$ , then apply  $X$  measurements on qubits  $a$  and  $b$ , denoted as  $[X]_a$  and  $[X]_b$ . Suppose the measurement results are  $x$  and  $y$ , respectively. Applying the  $Z$  operation before the  $X$  measurement  $[X]$  on a qubit does nothing but flip the measurement outcome, therefore

$$\begin{aligned} |\psi\rangle &= [X]_a [X]_b Z_L^{u_0} Z_v^{v_0} Z_R^{w_0} Z_a^{x_0} Z_b^{y_0} |G\rangle \\ &= Z_L^{u_0} Z_v^{v_0} Z_R^{w_0} [X]_a Z_a^{x_0} [X]_b Z_b^{y_0} |G\rangle \\ &= Z_L^{u_0} Z_v^{v_0+y_0+y} Z_R^{w_0+x_0+x} |\tilde{G}\rangle \end{aligned}$$

local  $Z$  operations on  $a$  and  $b$  pass to  $R$  and  $v$  respectively after graph contraction.

## References

- [1] Raussendorf R and Briegel H-J 2001 *Phys. Rev. Lett.* **86** 5188
- [2] Raussendorf R and Briegel H-J 2002 *Quantum Inf. Comput.* **2** 443
- [3] Raussendorf R, Browne D E and Briegel H-J 2003 *Phys. Rev. A* **68** 022312
- [4] Nielsen M A 2004 arXiv:quant-ph/0402005
- [5] Nielsen M A and Dawson C M 2004 arXiv:quant-ph/0405134
- [6] Browne D E and Rudolph T 2004 arXiv:quant-ph/0405157
- [7] Clark S R, Alves C M and Jaksch D 2005 *New J. Phys.* **7** 124
- [8] Barrett S D and Kok P 2004 arXiv:quant-ph/0408040
- [9] Walther P, Resch K J, Rudolph T, Schenck E, Weinfurter H, Vedral V, Aspelmeyer M and Zeilinger A 2005 *Nature* **434** 169
- [10] Benjamin S, Eisert J and Stace T 2005 *New J. Phys.* **7** 194
- [11] Jozsa R 2005 arXiv:quant-ph/0508124v2
- [12] Broadbent A and Kashefi E 2007 arXiv:0704.1736 [quant-ph]
- [13] Cleve R and Gottesman D 1997 *Phys. Rev. A* **56** 76
- [14] Gottesman D 1997 arXiv:quant-ph/9705052
- [15] Nielsen M and Chuang I 2000 *Quantum Information and Computation* (Cambridge: Cambridge University Press)
- [16] Cleve R and Gottesman D 2002 *Quantum Inf. Comput.* **2** 4
- [17] Grassl M, Klappenecker A and Roetteler M 2001 *Proc. IEEE ISIT*
- [18] Nest M V, Dehaene J and Moor B D 2004 arXiv:quant-ph/0308151v2
- [19] Nielsen M A 2004 *Phys. Rev. Lett.* **93** 040503
- [20] Yoran N and Reznik B 2003 *Phys. Rev. Lett.* **91** 037903
- [21] Browne D E and Rudolph T 2005 *Phys. Rev. Lett.* **95** 010501
- [22] Barrett S D and Kok P 2005 *Phys. Rev. Lett.* **71** 060310
- [23] Duan L-M and Raussendorf R 2005 *Phys. Rev. Lett.* **95** 080503
- [24] Jaksch D, Briegel H-J, Cirac J I, Gardiner C W and Zoller P 1999 *Phys. Rev. Lett.* **82** 1975
- [25] Mandel O, Greiner M, Widera A, Rom T, Hansch T W and Bloch I 2003 *Nature* **425** 937
- [26] Hein M, Dur W, Eisert J, Raussendorf R and Nest M V 2006 arXiv:quant-ph/0602096
- [27] Dan V, Kashefi E and Panangaden P 2007 *J. ACM* **54**
- [28] Hoyer P, Mhalla M and Perdrix S 2006 *Proc. 17th Int. Symp. on Algorithms and Computation (ISAAC) (Lecture Notes in Computer Science vol 4288)* (Berlin: Springer) pp 638–49