

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
 Department of Physics, EECS, and Department of Applied Math
 MIT 6.443J / 8.371J / 18.409 / MAS.865
 Quantum Information Science
 March 1, 2006

Problem Set #3
 (due in class, 16-Mar-06)

Lecture Topics (3/2, 3/7, 3/9, 3/14): fault tolerant QC; the threshold theorem; models of QC

Recommended Reading: Nielsen and Chuang, Section 10.6

Problems:

P1: (Threshold estimates for Bacon-Shor code) The fault tolerance threshold is a property of a code, its quantum error correction procedure, and circuits for performing computations on the encoded data. As these procedures and circuits become more complex (say, due to increasing code complexity), the number of ways they can fail grows, and thus the threshold becomes worse. It is thus tempting to focus on the simplest possible codes, such as the 7-qubit Steane code, for implementing fault tolerant QC. However, other codes may have properties which significantly simplify the necessary complexity of the QEC procedures and basic encoded operations. Here, we explore one such code, a version of Shor’s 9-qubit code modified by David Bacon.

The Bacon-Shor codes are defined by their stabilizer generators,

$$g_1 = \text{XXXXXXIII} \tag{1}$$

$$g_2 = \text{IIIXXXXXX} \tag{2}$$

$$g_3 = \text{ZZIZZIZZI} \tag{3}$$

$$g_4 = \text{IZZIZZIZZ}, \tag{4}$$

the normalizer operators for four “gauge” qubits,

$$Z_1 = \text{ZZIIIIIII} \quad X_1 = \text{XIIIIIXII} \tag{5}$$

$$Z_2 = \text{IZZIIIIIII} \quad X_2 = \text{IIXIIIIIX} \tag{6}$$

$$Z_3 = \text{IIIIZZIIII} \quad X_3 = \text{IIIXIIXII} \tag{7}$$

$$Z_4 = \text{IIIIZZIII} \quad X_4 = \text{IIIIIXIIX}, \tag{8}$$

and the normalizer operators for the one encoded qubit,

$$Z_5 = \text{ZZZZZZZZZ} \quad X_5 = \text{XXXXXXXXX}. \tag{9}$$

(a) Assuming that errors on the gauge qubits can be neglected, give the set of one and two-qubit

errors which can be corrected (or ignored). Note that the “gauge” qubits play a role similar to the DFS explored in problem set #2.

- (b) Draw quantum circuits for fault-tolerantly performing logical X , Z , H , and S gates on an encoded qubit, and a logical CNOT gate on two encoded qubits. You can leave the quantum error correction procedure as a black box for this part. Be sure, however, that your constructions do not allow single failures to propagate to become unacceptably many failures in any single code block.
- (c) The straightforward method for performing quantum error correction on this code is to measure the stabilizer generators, decode the error syndrome, and then perform the appropriate recovery procedure. Assume you can obtain n -qubit $|00 \cdots 0\rangle + |11 \cdots 1\rangle$ “cat state” ancilla qubit states, with any n . Draw a circuit which uses such ancilla to fault-tolerantly perform quantum error correction.
- (d) Estimate the number of ways components can fail, and cause the error correction procedure to perform incorrectly. Assume the cat-states might each have a single qubit error, and don’t worry about getting the exact number. A bound on the fault tolerance threshold is given by the inverse of this count, although a more realistic threshold should take into account the failure probability of encoded gates, and not just error correction.
- (e) Can you come up with a simpler quantum error correction procedure, and thus obtain a better threshold? [Think about this, but don’t spend too much time on the problem if you have research to do.]

P2: (Quantum Fredkin gate via teleportation) A fault-tolerant construction for the $\pi/8$ gate was shown in class. This construction used a certain ancilla state $|\chi\rangle$, together with Bell basis measurement (CNOT, Hadamard, and projective single qubit measurement) and single qubit Pauli operations. A similar construction can also be used to perform a fault-tolerant quantum Fredkin gate, which also completes a universal gate set for quantum computation.

- (a) The quantum Fredkin gate is defined as the following three qubit unitary transform:

$$U_F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

where the first qubit is considered the control, and the second two (the LSB and next-to-LSB) are the targets. If we label the states as $|abc\rangle$, note how the effect of this gate is to swap b and c when $a = 1$. Show that the gate set $\{H, \text{CNOT}, \text{and } U_F\}$ is universal for quantum computation, that is, any unitary transform can be arbitrarily well approximated by a combination of these gates.

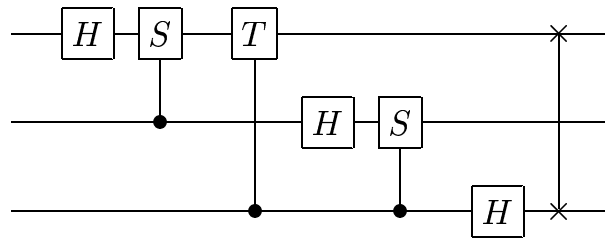
- (b) Compute the commutators of U_F and the Pauli gates, that is, give $U_g = U_F g U_F^\dagger$ for $g \in \{IIX, IXI, IXX, XII, XIX, XXI, XXX\}$ and similarly with Z replacing X. Express each U_g as a product of known Clifford group gates (recall that the Clifford group is generated by H , S , and CNOT, and is not universal for quantum computation because of the Gottesman-Knill theorem).

(c) Let $|\chi\rangle$ be the six qubit ancilla state

$$|\chi\rangle = \sum_{x=0}^7 |x\rangle \otimes U_F|x\rangle; \quad (11)$$

this state is obtained by applying a Fredkin gate to the halves of three EPR pairs. Use this state as the ancilla input to three teleportation circuits, and apply the results of the previous part of this problem to give the eight classically controlled operations which allow an arbitrary state $|\psi\rangle$ to be transformed through this ancilla to give $U_F|\psi\rangle$, up to an irrelevant global phase.

P3: (Cluster model implementation of quantum Fourier transform) The cluster model of quantum computation, described in class, uses a fixed, initial state of many qubits, together with a sequence of classically controlled measurements, to implement arbitrary quantum circuits. Here, we explore the cluster model by implementing a specific three qubit circuit, the quantum Fourier transform, which has this circuit representation:



S and T are the phase and $\pi/8$ gates. The right-most gate is a swap, needed to correct the ordering of the bits in the result. As a matrix the quantum Fourier transform in this instance may be written out explicitly, using $\omega = e^{2\pi i/8} = \sqrt{i}$, as

$$\frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^1 & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^1 & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega^1 \end{bmatrix}. \quad (12)$$

- One way to construct a cluster model implementation is to replace each gate in the circuit with ones which have known cluster model implementations, such as single qubit gates and the CNOT. Using this approach, estimate the number of qubits needed in the initial cluster state.
- Draw the cluster state, labeling the qubits, and describe a sequence of measurements and classical control steps which implement the QFT. Hint: there is a simple way to obtain controlled- $R_z(\theta)$ operations using single qubit operations and CNOT gates.
- Comment on the efficiency of your implementation. Do you think a smaller cluster model implementation is possible?