Homework 4

Due: October 27 at 11:59 PM. Submit on Canvas.

Problem 1 (Optimizing syndrome extraction in the surface code): In Lecture 14 we discussed an overly simplified syndrome extraction circuit for the rough surface code. To reach the best possible threshold, it is usually desirable to have the smallest depth syndrome extraction circuit. Therefore, one needs to parallelize the X-check and Z-check syndrome extraction as much as possible.

Find such a circuit for the rough surface code with the smallest possible depth.[1] Your circuit for syndrome extraction should involve only CNOT gates between data and ancilla qubits, as we discussed in Lecture 14. Each data qubit must be involved in at most one CNOT gate per circuit round. You should also avoid hook errors and bad CNOT scheduling, as discussed in Lecture 14.

[1] *Hint:* You may find the last two figures in Section 14 of the typed notes very useful for thinking about this problem.

Problem 2 (Logical gates via physical SWAP gates): In Lecture 15 we discussed the value of transversal gates for fault-tolerant quantum computation. A very related strategy is to implement fault-tolerant logical gates by physically swapping qubits, which may have a very low error rate in trapped ion or neutral atom hardware. The SWAP gate simply exchanges two qubits, exactly what it sounds like:

$$\mathrm{SWAP}_{1,2} \begin{pmatrix} X_1 \\ X_2 \\ Z_1 \\ Z_2 \end{pmatrix} \mathrm{SWAP}_{1,2}^\dagger = \begin{pmatrix} X_2 \\ X_1 \\ Z_2 \\ Z_1 \end{pmatrix}. \tag{1}$$

10 **A:** Explain why a necessary and "sufficient" condition for a CSS code to have a logical gate implemented by qubit permutation is the following. Let Σ be the $n \times n$ matrix that describes the permutation of qubits: namely, $\Sigma_{ij} = 1$ if and only if j is sent to i under the permutation. Then we require that there exist invertible $\mathbb{F}_2$-valued matrices M_X and M_Z such that

$$H_X = M_X H_X \Sigma, \tag{2a}$$
$$H_Z = M_Z H_Z \Sigma. \tag{2b}$$

10 **B:** It remains to find an example of a code that obeys (2), where the resulting transformation on on logical operators does not act as the identity. Here is one such example:

$$H_X = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3a}$$

$$H_Z = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3b}$$

Verify that this is a [[10,2,3]] code, and find G_X and G_Z.[2]

15 **C:** Consider the permutation that swaps qubit $2k-1$ with qubit $2k$ for $k = 1, 2, 3, 4, 5$. Show that this obeys (2). Further show that this physical SWAP implements a non-trivial logical Clifford gate, and find what it is in the presentation of G_X and G_Z you found in **B**.

[2] Of course the answer is not unique, although $\mathrm{im}(G_X^\mathsf{T})$ and $\mathrm{im}(G_Z^\mathsf{T})$ are.

Problem 3 (Universal "transversal" computation): Consider a concatenated CSS code where the 7-qubit Steane code is the outer code and the 15-qubit Reed-Muller code is the inner code. We will show how to implement logical H and T fault-tolerantly and "transversally" within this single code, in a sense that we will make concrete below. Before we do this, it is useful to organize the 105 qubits of the code into an array (α, i), where $\alpha = 1, \ldots, 7$ denotes the outer index of the Steane code, and $i = 1, \ldots, 15$ denotes the inner index of the Reed-Muller code.

10 **A:** What are k and d for the concatenated code, which has $n = 105$ qubits?[3]

10 **B:** Let H_{15} denote a (non-transversal) logical Hadamard gate on the Reed-Muller code. Explain why the 105-qubit code has a "transversal" logical Hadamard

$$H_{\mathrm{L}} = \bigotimes_{\alpha=1}^{7} (H_{15})_{\alpha}. \tag{4}$$

Further explain why any one fault in this process is a correctable error.

10 **C:** Using the Steane code stabilizers from Lecture 15, explain why the gate

$$T_{\mathrm{L}7} := \mathrm{CNOT}_{5\to6}\mathrm{CNOT}_{6\to7}T_7\mathrm{CNOT}_{6\to7}\mathrm{CNOT}_{5\to6} \tag{5}$$

implements a logical T gate on the 7-qubit Steane code (as it was presented in Lecture 15).

10 **D:** Use the method above to show how to implement a logical T (or $T^\dagger$) gate on the 105-qubit code by a sequence of "transversal" gates. Be sure to explain what partition of the 105 qubits is being used when defining transversal here. Explain why you can correct for any single fault that occurs during this process.

5 **E:** Conclude that if we perform stabilizer error-correction on the 105-qubit code after applying each logical T or H gate found above, if the probability p of any single-qubit error anywhere in the process is sufficiently small, then the probability of logical errors in *either* gate is $\mathrm{O}(p^2)$. You do not need to optimize the prefactor; the point is that this procedure has successfully suppressed the logical error rate below the physical one for small enough p.

5 **F:** Identify the "loophole" in the Eastin-Knill Theorem we are taking advantage of here, and give a physical picture for why we can implement (arbitrarily good approximations to) any logical gate using this code without violating the intuition from the proof sketch of this theorem in Lecture 16.

[3] Please do not write out H_X, H_Z, G_X or G_Z explicitly for the concatenated code – a conceptual understanding of where they come from should suffice for this entire problem!