Quantum Computation 101 for Physicists

Class exercise 7

Today we will learn how to construct the **5-qubit error correcting code**. Recall our analysis from class: Assume we have an error correcting code with codewords of length n qubits, and we want to be able to correct one error in one qubit, which can be of the form of the Pauli matrices X, Y, Z or any of its linear combinations. The number of possible errors is $3n+1$, 3 possible errors for each qubit, plus the possibility of no errors. We will thus need a Hilbert space of size $2(3n+1)$, since we want each codeword to represent $|0\rangle$ or $|1\rangle$. Then we require $2^n \leq 2(3n+1)$, and the smallest number that obeys this is 5 qubits.

We will follow the following process for designing an error correcting code:

1. Calculate the number of possible different errors that can occure.
2. From (1), calculate the number of syndrome operators you will need for identifying the mistake.
3. Find such syndrome operators M_i, which square to the identity matrix, and (preferably) commute with each other and are independent of each other. These requirements actually define a stabilizer code.
4. Define the logical states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ as the $n-$qubit codewords that represent $|0\rangle, |1\rangle$, and are eigenstates of the operators above. For clarity, choose these states such that For all i, $M_i|\bar{0}\rangle = |\bar{0}\rangle, M_i|\bar{1}\rangle = |\bar{1}\rangle$. Make sure these states are orthogonal and normalized.
5. Define the error correcting circuit using ancilla qubits.
6. Define a way to generate the states above and the logaical 1-qubit and 2-qubit gates.

So, let's begin!

1. **Number of possible errors:** $3 \cdot 5 + 1 = 16$.
2. **Number ν of syndrome operators:** The operators should square to the identity matrix, so each operator has 2 possible eigenvalues, 1 and -1, so $2^\nu = 16 \Rightarrow \nu = 4$.
3. **Find the operators:** Our requirement is that they would be independent, commute with each other and square to the identity. Our best bet would be to construct them from Pauli matrices:

$$M_0 = Z_1 X_2 X_3 Z_4$$
$$M_1 = Z_2 X_3 X_4 Z_0$$
$$M_2 = Z_3 X_4 X_0 Z_1$$
$$M_3 = Z_4 X_0 X_1 Z_2.$$

We are lucky enough to find such operators that are products of 1 qubit gates, which makes their implementation easier. They all commute with each other - you can test yourself and see that while each pair of M operators has exactly two pairs of anticommuting operators.

Now, let's see what happens if we have some general state $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$ and an X error occures on the qubit j, and we then measure the operator M_i:

$$\begin{aligned}
&(\alpha\langle\bar{0}| + \beta\langle\bar{1}|)X_j M_i X_j(\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle) = \\
&\pm (\alpha\langle\bar{0}| + \beta\langle\bar{1}|)X_j^2 M_i(\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle) = \\
&\pm (\alpha\langle\bar{0}| + \beta\langle\bar{1}|)(\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle) = \pm 1,
\end{aligned}$$

where the sign comes from the commutation / anti-cmmutation of M_i with X_j. Now, if we indeed have independent operators, we will have 16 different combinations of $[M_i, X_j]$, one for each possible error. So we need to check the $4 \cdot 16$ combinations of syndrome operators and Pauli matrices. Luckily, Mermin did this step for us:

	$\mathbf{X}_0\mathbf{Y}_0\mathbf{Z}_0$	$\mathbf{X}_1\mathbf{Y}_1\mathbf{Z}_1$	$\mathbf{X}_2\mathbf{Y}_2\mathbf{Z}_2$	$\mathbf{X}_3\mathbf{Y}_3\mathbf{Z}_3$	$\mathbf{X}_4\mathbf{Y}_4\mathbf{Z}_4$	$\mathbf{1}$
$\mathbf{M}_0 = \mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4$	$+++$	$--+$	$+--$	$+--$	$--+$	$+$
$\mathbf{M}_1 = \mathbf{Z}_2\mathbf{X}_3\mathbf{X}_4\mathbf{Z}_0$	$--+$	$+++$	$--+$	$+--$	$+--$	$+$
$\mathbf{M}_2 = \mathbf{Z}_3\mathbf{X}_4\mathbf{X}_0\mathbf{Z}_1$	$+--$	$--+$	$+++$	$--+$	$+--$	$+$
$\mathbf{M}_3 = \mathbf{Z}_4\mathbf{X}_0\mathbf{X}_1\mathbf{Z}_2$	$+--$	$+--$	$--+$	$+++$	$--+$	$+$

and we see that we have a different combination for each possible 1-qubit error.

4. **Logical states:** We want to get states that have an eigenvalue 1 for all M_i. One way to do it is to project onto the subspace of eigenvalues 1 of each operator: $\frac{1}{2}(1 + M_i)$ will annihilate states with eigenvalue -1 and leave states with eigenvalue 1 intact. We thus get:

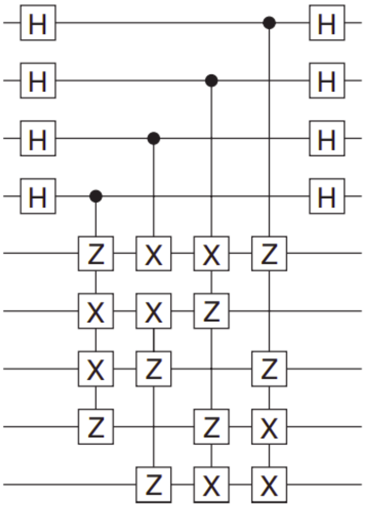
$$|\bar{0}\rangle = \frac{1}{4}(1 + M_0)(1 + M_1)(1 + M_2)(1 + M_3)|00000\rangle,$$
$$|\bar{1}\rangle = \frac{1}{4}(1 + M_0)(1 + M_1)(1 + M_2)(1 + M_3)|11111\rangle.$$

The states are orthogonal:

$$\langle\bar{0}|\bar{1}\rangle = \frac{1}{16}\langle 00000|2^4 \prod_i (1 + M_i)^2|11111\rangle =$$
$$\langle 00000|\prod_i (1 + M_i)|11111\rangle = 0,$$

since all terms in the product will flip at most 4 qubits. This can be better understood by the fact that we started with two orthogonal states, $|00000\rangle, |11111\rangle$, and only projected out parts of them.

5. **Error correcting circuit:** We saw above that we can measure the M_i operators in order to identify the error. However, we don't want to measure anything directly on our working qubits, or rather, we cannot do it without harming our state (Important note: Measuring $\langle M_i\rangle$ will not harm the state in principle, since both $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are of eigenvalue 1. However, we can only measure things in the computational basis on our qubits, one qubit at a time, and this will force us to measure something more specific than M_i). We therefor add ancilla qubits, one for each syndrome operator. Like we saw in class, we create the following circuit:



After applying the first $H^{\otimes 4}$ on the ancilla qubits, the state of the ancilla register is $\frac{1}{4}\sum_{x=0}^{15}|x\rangle$. Assume our working register is in state $E|\psi\rangle$, where E is the error operator. We define

$S_{i,E} = 0$ if M_i and E commute, and $S_{i,E} = 1$ if they anticommute. Applying the 4 syndrome operators, we entangle the ancilla and working qubits to get:

$$\frac{1}{4}\sum_{x=0}^{15}\prod_{i=0}^{3} M_i^{x_i} E|\psi\rangle|x\rangle = \frac{1}{4}\sum_{x=0}^{15}\prod_{i=0}^{3}(-1)^{x_i S_{i,E}} E|\psi\rangle|x\rangle,$$

where x_i is the ith bit of x. Applying $H^{\otimes 4}$ again will give us

$$\frac{1}{16}\sum_{y=0}^{15}\sum_{x=0}^{15}\prod_{i=0}^{3}(-1)^{x_i(S_{i,E}+y_i)} E|\psi\rangle|y\rangle = \frac{1}{16}\sum_{y=0}^{15}\sum_{x=0}^{15}(-1)^{x\cdot(S_E+y)} E|\psi\rangle|y\rangle.$$

The state of the ancilla register with a non-zero coefficient will be

$$y = |S_{3,E}S_{2,E}S_{1,E}S_{0,E}\rangle.$$

Now that we know E, i.e. which error occured, we can apply it again and cancel it.

Note tht if E was not a Pauli matrix but some superposition of Pauli matrices, that by measuring the ancilla qubits the state would collapse into one single type of error, which we can fix. We did not collapse the state itself - just the type of error that occured!

6. **Generation of $|\bar{0}\rangle, |\bar{1}\rangle$:** Naively, we could generate $|00000\rangle$ and apply $1 + M_i$ on it, However, while M_i is a product of 1-qubit gates and is thus easy to implement, $1 + M_i$ is not, and in fact, it is not even a unitary, it is a projector. We want to exploit the fact that the M_is are easy to implement. The idea we will use is to start with some state that is easy to generate, like $|00000\rangle$, and apply the error correcting method above on it in order to get some $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$.

   Say we want to extract $|\bar{0}\rangle$ out of this. If it was just a single qubit in the state $\alpha|0\rangle + \beta|1\rangle$, we would have just measured Z. If we got $|0\rangle$, we are done, and otherwise, we apply X. Same goes for extracting $|\bar{1}\rangle$. So what we need is to find $\bar{Z}, \bar{X}$, the logical 1-qubit gates, and we are done.

   We are lucky! $\bar{Z} = Z_0Z_1Z_2Z_3Z_4$ and $\bar{X} = X_0X_1X_2X_3X_4$ will perform the task we need: since both operators above commute with all M_i, we see that

$$\bar{Z}|\bar{x}\rangle = \bar{Z}\prod_{i=0}^{3}(1+M_i)|xxxxx\rangle = \prod_{i=0}^{3}(1+M_i)\bar{Z}|xxxxx\rangle = (\langle x|Z|x\rangle)^5\prod_{i=0}^{3}(1+M_i)|xxxxx\rangle = (\langle x|Z|x\rangle)|\bar{x}\rangle$$

   for $x = 0, 1$, and similarly for X.

   Unfortunately, for the 5-qubit code, more complex gates like Hadamard or CNOT will not be so easy to implement. This is the major advantage of the 7-qubit code over the 5-qubit code. Moshe showed you in class that for the 7-qubit code, the logical Hadamard gate is implemented in the same way as $\bar{X}$ and $\bar{Z}$, and at home you will do the same for $CNOT$. Hopefully it will give you some intuition regarding why it is not so easy for the 5-qubit code.