Seminar Quantum Computation

Fault-Tolerant Quantum Computation

Bartosz Blimke

Cottbus, 29.01.2004

1. What is fault-tolerant computation ?

The quantum computing network is called fault tolerant if it can recover from some errors during operations. Of course, there are no perfect networks, and it can happen that all gates in network malfunction. Then we have to change a little definition of fault-tolerant network:
Computation network we will call fault-tolerant if probability that network malfunctions is at most ε^2 where ε is a probability that one error on a gate or qubit occurs.

2. Why error-correcting codes are not sufficient guarantee for successful error-correction ?

There are 5 main reasons why error-correcting codes are not sufficient for providing quantum computation on “noisy quantum computers”.

- Error-correcting codes only help, but they are not enought for fault-tolerant computation.
- Error-correcting codes need additional resources and they can slow overall computation highly.
- Quantum error-correcting networks are by themselves not trivial networks and errors can occur during their performance.
- It’s not sufficient to encode transmitted information. It’s important that we could transmit quantum information reliabile for a long time and through a long distance.
- We need to process quantum information for a sufficient long time.

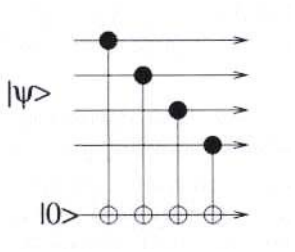
3. Some solutions for fault tolerant quantum networks.

3.1. Error propagation

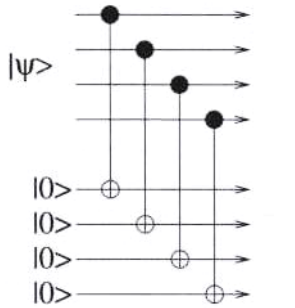
First problem that whe have to learn to fight is error-propagation due to entanglement. If we have two qubit gate then the error on one of the qubits could provide to error on the second qubit.

For example XOR gate that is very important for error-correcting networks could provide to such a problem. If there is an error on control qubit this could provide to error on target qubit. Under some circumstances it is

possible that target qubit becomes to be a target qubit and can propagate an error. On the picture below we can see that one phase error qubit of ancilla (in this case ancilla is $|0\rangle$) could propagate an error to several qubits of code.



Next picture presents a much better circuit where one error qubit of ancilla can propagate only to one qubit of code.



3.2. Fault-tolerant ancilla.

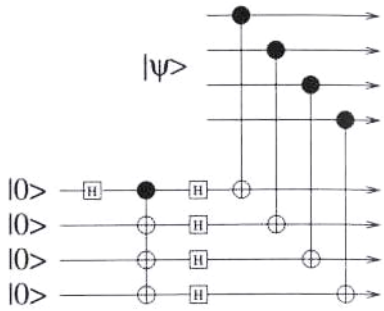
The solution for circuit above is still “risky”. The problem is that the code qubits keep being entangled with ancilla qubits and this can provide that code qubits will be destroyed during measurement of ancilla.
The circuit like above is very popular especially during syndrome computation for error-correcting codes. This what we need is a way to copy information form the code qubits to ancilla qubits without destructive effects on the state being “copied”.

One way have been presetnetd by Shor (1995). Instead of using $|0\rangle$ qubits for initial state for ancilla he used:

$$|\phi\rangle = \frac{1}{\sqrt{8}} \sum_{i \in \{0,1\}^4} (1 - parity(i)) |i\rangle$$

superposition of all even-parity 4 qubit codewords.

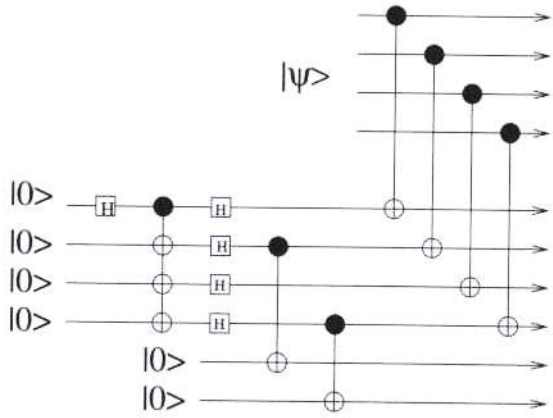
This state of ancilla can be created using circuit presented below.



Now if we will measure parity of ancilla according to basis H_{16} , then its parity is corresponding syndrome. But what is importand, after XOR operations ancilla state is not entangled with encoded qubit and measurement of ancilla cannot destroy it.

3.3. Verification of ancilla state

Another problem is that initial ancilla state can be erroneous, and the sign error which occurs during preparation of ancilla can propagate to encoded information.
The idea is to use two additional ancilla qubits and to make XOR operation with two randomly choosen qubits of ancilla. Then is the results of both operations don't agree then new ancilla should be constructed. Circuit using such a method is presented below.



3.4. Testing of syndrome measurements.

We know that there are circuits for measuring syndrome but that doesn't guarantee that the result will be correct.
To have correct syndrome with high probablility, we can repeat measurements until the same syndrome will obtained "k" times in a row, where k is some reliability parameter.

3.5. Fault-tolerant software quantum gates.

If we want to have fault-tolerant networks we would have to transmit encoded information, then decode it, apply quantum gate and then after gate encode information again. This method has unfortunately lot of drawbacks.

What we really need is a universal set of gates that could operate on encoded informations and all operations inside gates would be proceeded in “protected” subspace. Thats mean that we need fault-tolerant gates.

It has been show by Shor(1996) thah there exists set of fault-tolerant circuits for rotation, XOR and the Toffoli gate, which can operate on encoded informations for a special type of quantum error-correcting codes.

I will show for example how to realize Hadamard transformation and XOR for special set of codes called CSS codes with additional assumption that $C_1 = C_2^{\perp}$.

If the function $E(x)$ represents the act of encoding the qubit, the action of a Hadamard on an encoding qubit must follow the transformations:

$$E(|0\rangle) \to \frac{1}{\sqrt{2}}\left(E(|0\rangle) + E(|1\rangle)\right)$$
$$E(|1\rangle) \to \frac{1}{\sqrt{2}}\left(E(|0\rangle) - E(|1\rangle)\right)$$

A Hadamard transformation of each individual qubit, $H^{\otimes k}$, applied to $E(|0\rangle)$ will give precisely the correct encoded transformation.

$$H\frac{1}{\sqrt{|C_2|}}\sum_{x\in C_2}|x\rangle = \frac{1}{2^{\frac{k}{2}}\sqrt{|C_2|}}\sum_{y,x\in C_2}(-1)^{x*y}|x\rangle$$
$$= \frac{1}{\sqrt{|C_2^{\perp}|}}\sum_{y\in C_1}|y\rangle$$
$$= E\left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)$$

Last line follows becouse $E(|0\rangle)$ is composed from all codewords in C_2 and $E(|1\rangle)$ form everything in C_1 but not in C_2 by definition of CSS code.

Proof that XOR operation is fault-tolerant can be shown as follows:

$$U_{CNOT}^{\otimes k}\frac{1}{|C_2|}\sum_{x\in C_2}|x+v_a\rangle\otimes\sum_{y\in C_2}|y+v_b\rangle = \frac{1}{|C_2|}\sum_{x\in C_2}|x\rangle\otimes\sum_{y\in C_2}|x+y+v_a+v_b\rangle$$
$$= \frac{1}{|C_2|}\sum_{x\in C_2}|x\rangle\otimes\sum_{y\in C_2}|y+v_a+v_b\rangle$$

becouse of $x\in C_2$ and $y\in C_2$ then $x+y\in C_2$.

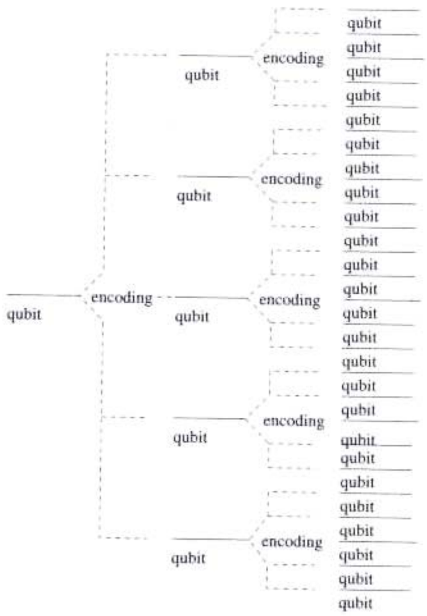
3.6. **Fault-tolerant hardware quantum gates.**

Another approach to fault-tolerant gates have been shown by Kitaev(1997). He showed that there exists set of fault-tolerant quantum computing gates and their fault-tolerance comes by their phisical nature.
Unfortunately it doesn't seem that it will be possible to realize these ideas in the near future.

3.7. **Concatenated coding**.

Error-correcting codes and fault-tolerant quantum gates are still not enough for long reliable quantum computing. Problem is that these techniques need additional often noisy gates and additional computational overhead and if error rate is not very small they can make more error than they fix.
Additionaly number of gates growth rapidly with the number of errors they can correct. We need a codes that allow us to increase quantity of errors that can be correct with the very small increase of time needed to perform these corrections.This will allow us to make long reliable quantum computations.

The idea for this was presented by Knill and Laflamme (1996). We have to encode qubits recursively up to a certain level of recursion and to perform often recovery operations.
For example data is first encoded using $[n,k,d]$ correction code, and in the next step every qubit of new codewords is encoded using $[n_1,1,d_1]$ code. The resulting qubits are coded again and so on to the chosen level "h" of hierarchy. This will give us resulting code $[nn_1n_2...n_h,k,dd_1d_2...d_h]c$. In special case encoding can be the same. Here we have example for n = 5, k = 1, h = 2 code.



Now it's easy to see why this can help. If ε is probablity that error will occure to one qubit with encoding [n,1,d] then the probablity of recovery error failure is ε^2 (according to our assumption about what is "fault-tolerance"). However if we have h levels of hierarchy the probability is ε^{2^h} (with the number of qubits n^h)

References:

[1] Gruska, Jozef: Quantum Computing.
McGraw-Hill Publishing Company, May 1999

[2] Lecture 23: Fault-Tolerant Quantum Computation
Scribed by: Jonathan Hodges
Department of Nuclear Engineering, MIT
December 4, 2003