CSE 599d - Quantum Computing
Fault-Tolerant Quantum Computation and the Threshold Theorem

Dave Bacon
Department of Computer Science & Engineering, University of Washington

In the last few lectures, we discussed quantum error correcting codes and defined a set of codes, the stabilizer codes, which had a very nice structure for allowing us to understand their properties. In this lecture we will see at the last the outlines of how to put these results to use in what is called the threshold theorem for fault-tolerant quantum computation.

I. PROBLEMS, PROBLEMS, EVERYWHERE

The theory of quantum error correcting codes was designed to deal with a setting where we can encode our quantum information perfectly, then subject the qubits in our codes to noise, and then we can perfectly decode the quantum information. This model is, in some ways, fine for talking about the transmission of quantum information but is clearly not adequate for building a quantum computer. Let's list the problems that we worry about when trying to build a robust quantum computer.

Imperfect evolution In the real world, when we try to implement some transform U on our quantum information, we will not be able to do this perfect. We may implement a different unitary V, or we may even implement some superoperator not equal to the simple unitary evolution. Thus we cannot assume that *any* of the gates we use implemented perfectly. This will have consequence for us not just in implementing transform on our encoded quantum information, but also on other tasks like performing recovery operations during our quantum error correcting routine.

Imperfect preparation We need to be able to prepare initial states into some fixed states. As a minimal requirement, we might require the preparation of some fixed $|0\rangle$ states. However, in the real world, such preparation routines may fail to produce the desired state.

Imperfect measurements When we read our our quantum states, our measurement apparatus may fail and give us the wrong result.

Doing nothing is hard When our quantum information is not even being acted upon, it is subject to coupling with the environment. Thus, even implementing quantum wires which do nothing to our quantum information can be hard.

These problems, which go above and beyond what we could achieve using just the ideas of quantum error correction we have discussed previously. In fact, one might think at first site that these challenges cannot be overcome. However, one of the remarkable discoveries of the mid-nineties was that these problems could be overcome. This is the theory of fault-tolerant quantum computation.

There are now numerous methods to deal with the problems we detail above. We will discuss one of these approaches which achieves fault-tolerance which involves the use of concatenated quantum error correcting codes.

II. A SAMPLE OF FAULT-TOLERANCE

Suppose that we are working with a qubit encoded into the Steane seven qubit stabilizer code. Now we previously asked the question of how to perform a gate on stabilizer codes. Of course, we also said that this question was, in many ways, trivial: just define the appropriate gate on the encoded quantum information. But when we come to fault-tolerance, we being to see that life is not so easy. Consider, for example, our circuit to implement the Hadamard gate on the encoded quantum information. We saw that this could be implemented by the 7 qubit parallel gate $H^{\otimes 7}$. Now when we implement this gate, a good assumption is that we implement each H separately via single qubit rotations. Then, if, say we over or under rotate to achieve each of these individual gates, then to first order in this rotation angle, this will look like we have achieved the correct $H^{\otimes 7}$ plus a single qubit error. But a single qubit error is what our code is designed to correct. This isn't bad, since we assume our quantum error correcting procedure will be able to fix this (don't worry about this procedure itself not being fault-tolerant yet.) But suppose that if we had decided to implement that Hadamard gate a single qubit at a time, we had actually coupled two of the qubits together to produce on of the $H^{\otimes 2}$ gates. This isn't so hard to do: just note that $H \otimes H = X \otimes X \exp\left(-i\frac{\pi}{4} Y \otimes Y\right)$. But now if we over-rotate this coupling, we will produce, to first order in the rotation angle an operation which is a two-qubit error. This is bad! Thus we see that how we implement gates is important if we are going to deal with imprecise gates.

This leads to the following notion of fault-tolerant gates. Suppose that we are working with quantum data encoded into a quantum error correcting code. Periodic application of the error correcting procedure will keep the probability of failure due to the accumulation of too many errors small. But if the gates we enact on this information can, by failing causes errors which cannot be corrected in by the code, then we are in trouble. Procedures which do not suffer from this problem are called fault-tolerant procedures. Further we need similar constructions not just for the gates we enact on our quantum computation, but also for preparation, measurement, and the quantum error correcting procedure itself.

III. CONCATENATION TO THE THRESHOLD THEOREM

Before discussing how to perform fault-tolerant operations, it is nice to see where all of our discussion is heading. Suppose that we have constructed a fault-tolerant set of operations (gates, preparation, measurement, error correction) for a quantum error correcting code. In such a construction, if the failure probability of individual components is p (we will assume failure probabilities, a more detailed analysis using superoperators should really be carried out, but for our purposes of getting the basic outline of fault-tolerance, the simpler model will be fine.) If the procedure is fault-tolerant, then for any of these procedures, the probability of the encoded operations failing is at most cp^2 for some constant c. If $pleq\frac{1}{c}$, then we see that we will have decreased the probability of failing below p. Is it possible to boost this probability even smaller with a reasonable overhead of resources?

A way to do this is via the concatenation of codes. Actually we've already saw the concatenation of two codes: the Shor code is a concatenation of the three qubit bit flip code with the three qubit phase flip code. Suppose we have two codes, C_1 and C_2 which each encode a single qubit of quantum information are are of size n_1 and n_2. Then we can construct a new code by taking n_2 sets of n_1 qubits. Each block of n_1 qubits can be used to construct one encoded qubit for the code C_1. Then each of these n_2 encoded qubits can be further used in a code C_2. This results in a new code make up of $n_1 n_2$ qubits. If C_1 corrects d_1 errors and C_2 corrects d_2 errors, then the new code corrects at least $d_1 + d_2$ errors. The procedure of taking two codes and combining them into a new code in this manner is known as concatenating coding.

Returning now to our question about using a fault-tolerant procedure to go beyond reducing the error from p to cp^2, we see that one possible way to do this is to use concatenation. We can take our original code and concatenated with itself. In the process of doing this, every operation on the second level of encoding will be enacted by a fault-tolerant construction from the first level of encoding, and this second level of encoding will also be done in a fault-tolerant manner. This second level of encoding will send the probability of failing from cp^2 to $c(cp^2)^2$. If we concatenate a times, then we see that the probability of a fault-tolerant component failing will be $c^{-1}(cp)^{2^a}$. If our code uses n qubits to encode 1 qubit of information, then the size of our code is n^a. Notice that the probability of failing falls doubly-exponentially in a whereas the size of the circuit grows exponentially in a. Suppose that we wish to implement one of our fault-tolerant circuit elements to accuracy ϵ. If $p < \frac{1}{c}$, then we can achieve any accuracy by this concatenation. We need $\epsilon = c^{-1}(cp)^{2^a}$. How big will our circuit need to be to achieve this accuracy? Suppose that the maximum number of operations involved in our fault-tolerant procedures is d (a fixed constant.) Then the size of our circuit after k concatenations is d^a. Solving the accuracy for a yields

$$a = \log_2 \log_{cp}(c\epsilon) \tag{1}$$

Thus the size of the circuit will be

$$d^a = d^{\log_2 \log_{cp}(c\epsilon)} = d^{\frac{\log_d(\log_{cp}(c\epsilon))}{\log_d 2}} = [\log_{cp}(c\epsilon)]^{\frac{1}{\log_d 2}} = \left[\frac{\log_2(c\epsilon)}{\log_2(cp)}\right]^{\frac{1}{\log_d 2}} = \left[\frac{\log_2\left(\frac{1}{c\epsilon}\right)}{\log_2(\left(\frac{1}{cp}\right))}\right]^{\log_2 d} \tag{2}$$

or

$$d_a = O\left(\text{poly}\left(\log\left(\frac{1}{c\epsilon}\right)\right)\right) \tag{3}$$

Thus if we have a quantum circuit with t gates, then to implement this circuit to accuracy ϵ, we need to implement each gate to accuracy $\frac{\epsilon}{t}$. Using concatenation this can be done with a circuit of size

$$O\left(\text{poly}\left(\log\left(\frac{t}{c\epsilon}\right)\right)\right) \tag{4}$$

The above result is at the heart of the threshold for fault-tolerant quantum computation. We see that if our probability of error is less that $\frac{1}{c}$, then we can perform robust quantum computation with a blowup in our circuit size which is only poly logarithmic in the accuracy with which we wish to perform our quantum computation. Of course we haven't demonstrated that fault-tolerant constructions which take p to cp^2 are possible, but this is our next task. And given those constructions, we have essentially shown the threshold theorem for fault-tolerant quantum computation (in a far from rigorous manner, however. There are lots of details which we need to pay attention to in the full proof. We will touch on some of these as we go through the fault-tolerant constructions.)

An interesting question which we can ask right now, however, is how the threshold theorem applies to building a quantum computer. On the one hand it gives us the optimism that building a robust quantum computer is not impossible. On the other hand, the requirements for quantum codes are fairly tough to achieve in practice. Further, one often gets the impression that from reading about the concatenated threshold theorem that indefinite concatenation is necessary. However, for a fixed accuracy, only a certain number of levels of concatenation will be necessary. Since the accuracy we wish to achieve is related to the algorithm we wish to enact, it will probably be true that quantum computers we build will be pushed to the limits of the smallest circuits to achieve the desired quantum algorithm with the desired accuracy. For example, recent work demonstrates that for the purposes of breaking RSA only two or three levels of concatenation are really needed. Further there is another possibility, and this is to use codes which are not concatenated but which can correct bigger errors. Which actual code and procedures will be used in future quantum computers is a subject area of great current research.

IV. WHAT TO ASSUME

In any discussion of the threshold theorem for fault-tolerant quantum computation, one of the most important points is what is assumed in the order to produce the threshold. It is useful to list the assumptions which are put into the model. We will not be able to discuss the threshold theorem under the most restrictive of all assumptions which can be made, but it is important to know what current methods can and cannot deal with. Of course it is a matter of physics and of engineering to determine which assumptions actually matter.

Noise Model A particular model (or models) of noise is assumed for all proofs and heuristic methods for the threshold theorem. These models must be motivated by the physics of our devices. Noise here includes all of the possible sorts of errant processes we described above, i.e. preparation, measurement, gate, and decoherence. An assumption which is most often made about decoherence is that it follows an independent error model: each qubit is effected by errors which are not correlated with errors on other qubits nor are the errors which occur on the qubit correlated in time. This may seem like quite a large assumption, and there is plenty of debate as to how realistic this model of noise is. However, in its defense, the independent error model corresponds to a physical model of decoherence which seems rather reasonable for many implementations of quantum computers. That being said, even the first threshold theorems for quantum computing also allowed for some deviation beyond the independent error model. Recently there has been a surge of methods for dealing with more general error models. Of course the nice thing is that the noise model relevant to a particular physical implementation of a quantum computer is an experimental question. It seems prudent, given the threshold theorem, to first work to deal with the errors that we know exist. If when we correct these, a new form of error comes to light, then we can work on fixing this type of error. Indeed it seems unlikely in many implementations that we can even measure the different noise model right now, even if they exist, since most of the implementations we know are dominated by decoherence methods related to an independent error model (the main exception to this is probably in many solid state implementations, where a lot less is known about the decoherence mechanism at work in these systems. On the other hand, for atomic systems like ions and neutral atoms, we have a pretty good idea that the main methods of decoherence at this point are well described by the independent error model.)

Parallelism In our fault-tolerant methods we will need to be able to apply many quantum gates simultaneously to our physical qubits. In fact, one can show that without such parallelism, fault-tolerant quantum computation is not possible.

Classical Computation Do we allow robust classical computation to act alongside our quantum computation? If so, this significantly reduces the complexity of proving a threshold theorem. A critical concern here from the experimental side is when dealing with quantum systems where gates act extremely fast. For example in many solid state implementations of qubits, the gate speeds are faster than modern classical computer clock speeds. Thus control of these systems must be performed by different methods than simple classical computer control, which significantly changes the difficulty in building fault-tolerant quantum circuits.

Spatial Locality In a real implementation of a fault-tolerant quantum circuit, the physical systems implementing these ideas must be laid out on a spatially local two (or perhaps three) dimensional connected geometry. In many early threshold calculations this locality was not considered in the threshold. However now there is considerable interest in understand how the threshold is effected by this assumption. This leads, in particular, to the very interesting study of quantum micro-architectures of fault-tolerant quantum computation.

Flying Qubits In some implementations of quantum computers, we may be able to have fixed qubits as well as flying qubits. The former of these qubits stay fixed in space, while the latter, usually photons, can be used to rapidly transmit quantum information. Such hybrid methods for quantum computing should not suffer as much of a penalty for spatial locality that we discussed above.

So as you can see there are lots of different assumptions to play around with for implementing a fault-tolerant quantum computer. Which of these will matter the most depends a great deal on what the physics experiments of the next few years tell us. I personally expect to see a quantum error correcting circuit which shows an improvement in decoherence times within the next three years (most likely in ion traps.)

V. FAULT-TOLERANT METHODS

Okay so now onto the methods for fault-tolerance. Suppose have some quantum data encoded into blocks of quantum error correcting codes. Then, for our purposes, a method will be considered fault-tolerant if, given a single fault of one of the components of this routine, the procedure will cause at most one error in each of the encoded blocks produced by the component.

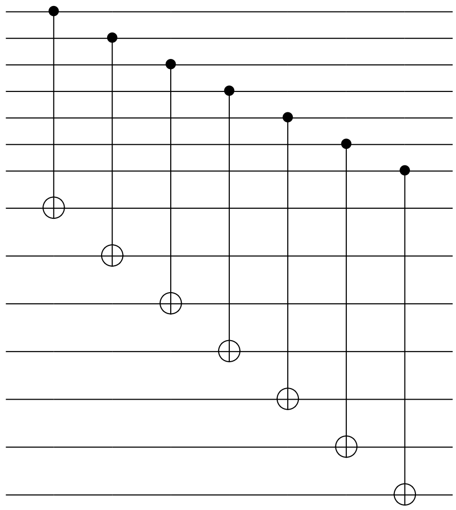
A. Fault-Tolerant Gates

We have already seen, for the Steane code, that implementing $H^{\otimes 7}$ on each individual qubit will results in a procedure which is fault-tolerant. A single failure of on of the H gates leads to a single error on our encoded quantum states. Similarly if a single qubit error occurred before the $H^{\otimes 7}$ gate, is like a single qubit error occurring after the gate. Of course the identity of this error may change: for example a single qubit X error before the $H^{\otimes 7}$ gate is like a single Z error occuring after the $H^{\otimes 7}$ gate. Thus we see that our implementation of the Hadamard gate is fault-tolerant.

What can we learn from this implementation of the Hadamard? Well notice that this gate is transversal: each qubit is acted upon by a single independent operator. It is easy to see that any such gate will be fault-tolerant. In fact, that this is true means that we often take the definition of fault-tolerant to mean transversal (investigation of fault-tolerant but not transversal gates is a highly neglected area of fault-tolerant quantum computation.)

What other gates can we implement on the Steane code in this manner? Well certainly the encoded Pauli operators, like $\bar{X} = X^{\otimes 7}$ can be implemented in this manner. What about a gate like the S gate we used in generating the Clifford group? Well one can check that $\bar{Z}S^{\otimes 7}$ implements the S gate on the information encoded into the Steane code. Again these implementations are transversal and so are fault-tolerant.

Now what else might we be able to implement? What about operations between two encoded qubits. One obvious gate we would like to implement is the controlled-NOT. This gate would be nice, since we know that H, S and C_X generate the Clifford group. Thus if we could implement an encoded C_X gate in a fault-tolerant manner, then we would have demonstrated how to perform all Clifford group elements in a fault-tolerant manner? So how do we implement fault-tolerantly a C_X on our encoded quantum information? Well luckily this is not hard! In fact all we need to do is to implement seven controlled-NOTs from the control encoded qubit to the target encoded qubit as

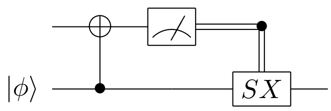


(5)

Of course in reality we would like to perform this operation in as parallel as manner as possible. Now why is this implementation of an encoded controlled-NOT fault tolerant? Well suppose that one of the controlled-NOTs fails when we are trying to implement it. Since this controlled-NOT only couples one qubit from the controlled encoded block and one qubit from the target encoded block, this will lead to at most a single error in either block. Further, suppose there was a single qubit error on a qubit in the first encoded qubit. Now if this was, say an X error on the control wire, then this is equivalent to an X error after the controlled-NOT on the control and the target wire. That the error has propagated this way is not bad for us, since we have a single error in one block which has changed into a single error on both blocks. Propagation of errors like this is one of the main concerns in implementing fault-tolerant constructions. But for transversal operations like the one we have performed for our encoded controlled-NOT, this isn't a problem.

So we have seen how to implement in a fault-tolerant manner a set of gates which generates the Clifford group gates on our encoded quantum data. How do we complete this set and produce a universal set of fault-tolerant quantum gates? There are a number of ways to achieve this. One way is to construct a fault-tolerant Tofolli gate. Another way is to construct a fault-tolerant $\frac{\pi}{8}$ gate. In order to achieve this latter construction, we actually demonstrate how preparation of a particular ancilla state can be used to produce the appropriate gate.

In particular, suppose that we have been given the single qubit state $|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$. Then, using operations from the Clifford group and measurement in the computational basis (whose fault-tolerant construction we will consider in a bit) we can construct the following circuit:



(6)

What is the effect of this circuit? Suppose that $\alpha|0\rangle + \beta|1\rangle$ is fed into the first qubit. Then before the controlled not, the state is

$$(\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle). \tag{7}$$

After the controlled-NOT this becomes

$$\frac{1}{\sqrt{2}}\left[(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle + e^{\frac{i\pi}{4}}(\beta|0\rangle + \alpha|1\rangle) \otimes |1\rangle\right] = \frac{1}{\sqrt{2}}\left[|0\rangle \otimes (\alpha|0\rangle + e^{\frac{i\pi}{4}}\beta|1\rangle) + |1\rangle \otimes (\beta|0\rangle + e^{\frac{i\pi}{4}}\alpha|1\rangle)\right] \tag{8}$$

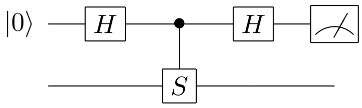
Thus we see that if we now measure the first qubit in the computational basis and get outcome $|0\rangle$ the second qubit will be T times in the input $\alpha|0\rangle + \beta|1\rangle$, where $T = |0\rangle\langle 0| + e^{\frac{i\pi}{4}}|1\rangle\langle 1|$. If however, we obtain outcome $|1\rangle$ then the second qubit will be $(\beta|0\rangle + e^{\frac{i\pi}{4}}\alpha|1\rangle)$. However in this case, if we apply X and then S, then we obtain T times in the input $\alpha|0\rangle + \beta|1\rangle$. Thus we see that the above construction is able to produce the $\frac{\pi}{8}$ gate T using only Clifford group elements and computational basis measurements assuming that we can prepare the state $|\phi\rangle$.

The gates T along with Clifford group gates, as we saw a long time ago in this course, are universal. Thus if we can show how to perform fault-tolerant measurement as well as fault-tolerant preparation of the $|\phi\rangle$ state, we will obtain a fault-tolerant universal set of quantum gates.

B. Fault-Tolerant Measurement

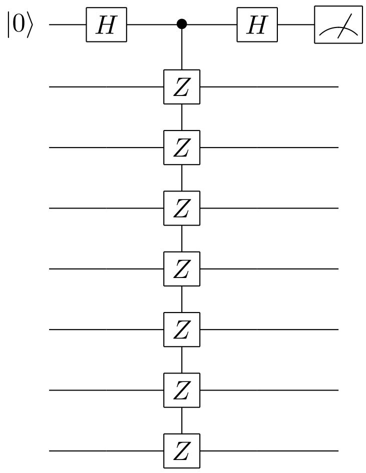
For fault-tolerant measurements, we require not just that a single component failing only spreads to one encoded block, but also that the probability of our measurement giving the wrong outcome is $O(p^2)$. This latter requirement comes from the fact that the measurement outcome becomes classical data which we may use to control future quantum operations.

So how do we perform fault-tolerant measurements? Recall that when we discussed measuring stabilizer operators we said that the circuit



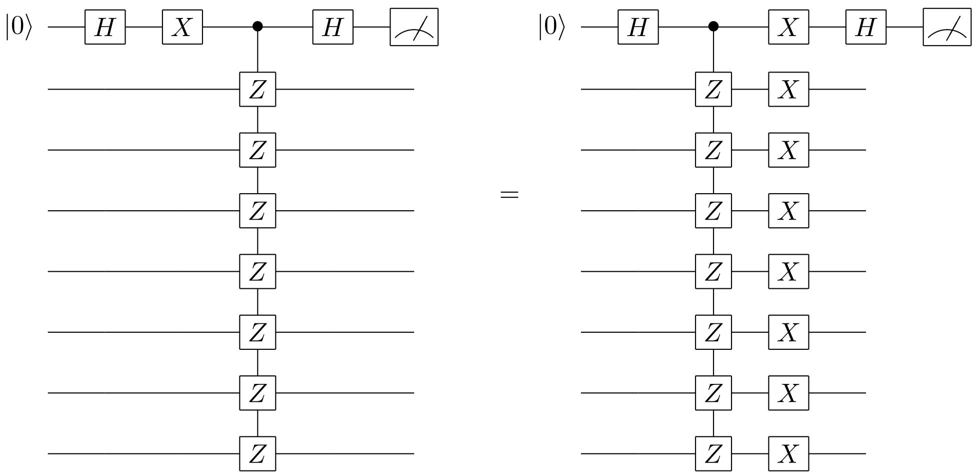
(9)

could be used to make a projective measurement onto the $+1$ or -1 eigenvalues of the operator S. Suppose we tried to do this to realize a gate which was implemented transversally in our quantum error correcting code. For example suppose we are measuring the $\bar{Z}$ operator on the Steane code. Then the circuit would be



(10)

But now notice that a single bit flip error before the control on this circuit will produce 7 errors on our encoded qubits!

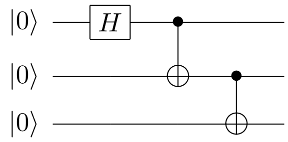


(11)

This is bad. Very bad! It violates our fault-tolerant criteria rather profoundly.

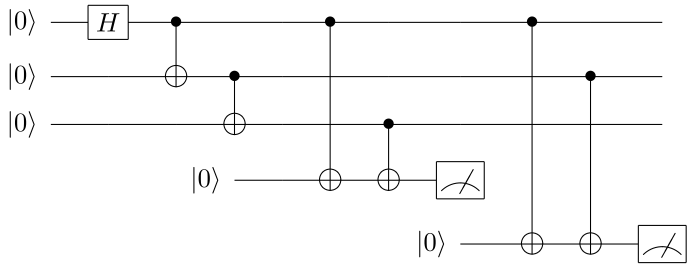
So how do we perform fault-tolerant measurement. One method is to do as follows. Suppose that we wish to measure a operator which is the tensor product of k operators which square to identity. What we do is we create a k qubit cat state: $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes k} + |1\rangle^{\otimes k})$ and then use this cat state to kick back the phase of each individual Pauli measurement (like our circuit above) and then finally make a measurement to distinguish whether we have the cat state $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes k}+|1\rangle^{\otimes k})$ or $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes k}-|1\rangle^{\otimes k})$. Now there are two issues to worry about. One is the propagation of errors when a component fails in this procedure (or an error occurs on a quantum wire.) The other is that the measurement result may not be correct. To deal with this latter procedure, we repeat the whole measurement procedure three times and take a majority vote. Thus errors which effect our measurement outcome can be changed from p to $O(p^2)$.

The other issue, the propagation of errors is more interesting. Let's examine the preparation of the cat state. A circuit for preparing a three qubit cat state, for example, is



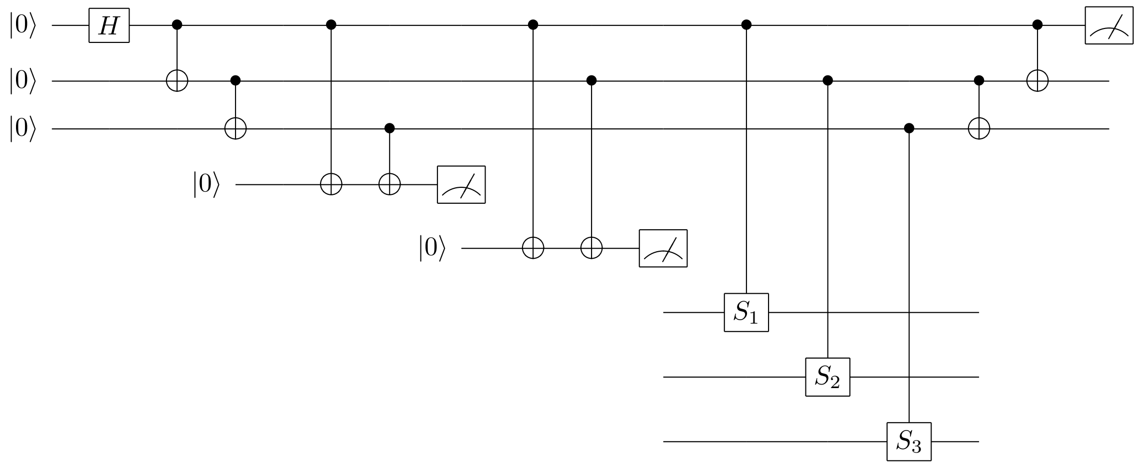
(12)

Then, following this preparation we will perform a verification procedure to (roughly) check whether we have prepared a cat state



(13)

If the outcomes of these verification procedures measure a parity which is not even, the whole cat state is thrown out and the procedure is started again. Following the cat state verification, we then perform the controlled operations to perform the measurement and undo the cat state preparation and measure:



(14)

Now why is this procedure fault-tolerant? Well first note that Z errors from the ancilla qubits do not propagate to errors on the encoded data (where the S_is act.) Z errors on the ancilla blocks can cause our measurement outcome to be incorrect, however, but we deal with this by repeating this measurement procedure as described above. What about X and Y errors? Well notice that if the controlled-NOTs involved in the verification fail, then this will only possibly produce one X error which can propagate to the encoded quantum data. Further these X errors can only propagate to the extra qubits we add beyond the cat state qubits. None of these lead to more than one error in our encoded quantum data. Similar arguments hold for all locations of the X errors (and note that X errors which occur in the cat state prepration and could be multiple X errors, are taken care of by the cat state verification procedure.)

Now the above procedure can be used to measure observables which have eigenvalues ± 1 and are implemented transversally.

C. Fault-Tolerant Error Correction and Preparation

We've seen above how to perform fault-tolerant measurements. This allows us, fairly trivially to perform fault-tolerant error correction: we simply use the fault-tolerant measurement routines to measure the stabilizer generators and then, given this diagnosis, perform the appropriate restoration on the encoded quantum state. This restoration is certainly fault-tolerant if done one qubit at a time in implementing the recovery. Further, fault-tolerant preparation is also possible. We do this as described previously by measuring the stabilizer generators, and the logical $\bar{Z}_i$ operators and then applying the appropriate recovery and possible $\bar{X}_i$ operations.

Thus in this section we've seen the main ingredients in perform fault-tolerant constructions for the threshold theorem. Our discussion has been only cursory, but hopefully you get the idea of how these methods work.

VI. A NEW PHASE OF MATTER?

With the threshold theorem for quantum computation, there is, at least philosophically, no valid model of computers based on quantum theory. In this way, fault-tolerant quantum computation lies at the very heart of what it means for something to be a quantum computer.

Further one can argue that the threshold for fault-tolerant quantum computation is really an indication that there is a strange new phase of matter, a quantum computer, which can robustly store and manipulate quantum information. Indeed you will probably be not surprised to know that the threshold theorem is intimately related arguments which occur when a physical system undergoes a change of phase. The discovery of a new phase of matter is always a time of great excitement in physics. Right now, then, we have the situation where theory predicts this new phase, and experiment is pressing hard to move the into this new phase. Realization of this new state will be as exciting as the realization of superconductivity, superfluidity, Bose-Einstein condenstation, or the quantum Hall effect.

Of course, the real (billion dollar) question is whether it is possible to build a robust quantum computer. Suppose, for example, that there existed a physical system which had nearly extremely long decoherence times and which we control to an extremely high precision. Then there would be no need for quantum error correction in practice and we wouldn't be having this discussion. However, as far as we know, such miraculous quantum systems do not exists. Actually it is questionable whether, as basic building blocks, even such classical systems exist: when we build a transistor out of a single molecule, what is the fidelity of this gate? Right now it is certainly not one hundred percent and we might wonder what happens to classical computers when we build computers from noisy classical components. But back to quantum computers, the real question is how far can we drive down our noise processes using the ideas of fault-tolerant quantum computation. If we can drive these low enough, then we will be able to outperform classical computers (on tasks like factoring, assuming there is no efficient classical algorithm for factoring.) The quest to do this is one of the great technological and theoretical challenges of the twentieth century.