Quantum Computing Notes
Prof. Hamzeh Roumani
EECS, York University, Toronto, Ont., Canada

QUANTUM ERROR CORRECTION

1. The Problem

When abstract computing models (such as the circuits that underlie quantum algorithms) are implemented physically , things do not unfold as neatly as the precise mathematical formalism that we have been using. For example, stray magnetic fields or microwave pulses, spontaneous decays, and imperfections in gates and measurements can all introduce errors by changing the states of qubits without our knowledge or control. For example, the state can undergo a *bit flip* (as in $|0\rangle \rightarrow |1\rangle$) or a *phase flip* (as in $|0\rangle + |1\rangle \rightarrow |0\rangle - |1\rangle$). Any such change renders our computation incorrect.

Given an *error model* that describes the various ways in which the state can change, and gives the probability of each, we need a solution that detects the presence of errors (the *syndrome*) and corrects the state as needed.

2. A Classical Solution

The classical error model is simple: With probability *p*, the error occurs, and the bit does flip. And with probability *1−p*, the error does not occur, and the bit stays as is. Moreover, *p* is < ½ and is independent of the state of the bit, i.e. the probability of 0→1 is the same as 1→0. The model further assumes that errors affect bits independently, i.e. no correlated errors.

With this model in place, the classical solution calls for replicating each bit used in the circuit. Specifically in the *three-bit-code*, we protect a bit by making two copies of it, i.e. 0→000 and 1→111. The copying process is known as *encoding*, and it turns the original *logical* bit to three *physical bits* known as the *codeword*. If a codeword is subjected to an error that causes one of its three physical bits to flip, the result can be corrected by taking a *majority vote*. For example, if 000 became 010, the majority is 0, and we correct back to 000. The same applies to all other codewords under a single bit flip. Moreover, the error syndrome can be determined by testing the equality of all three bits. The process of detecting errors and correcting then (if needed) is known as *decoding*.

The above 3-bit-code fails to correct errors if more than one physical bit within a codeword can flip. For example, it cannot correct 011 because it could have originated from 111 via one bitflip or from 000 via two bitflips or from 100 via three bitflips.

Let us quantify the success of this code by computing the probability of having un-correctable errors with and without using it:

- If we don't use this code (i.e. do not replicate the bit) then the probability of un-correctable errors is p (per the error model).
- If we use this code, the probability of un-correctable errors is equal to the probability of two or more bitflips, which is $p^2(1-p)$ for two flips (it happens in 3 ways) and p^3 for three, for a total of $3p^2(1-p)+p^3$. This is less than p (since $p<½$ in the model).

Hence, the 3-bit-code reduces the probability of un-correctable errors.

- Show that if $p=10\%$ then the 3-bit-code reduces the irrecoverable error rate to about 3%.
- Compute the irrecoverable error probability if $p=50\%$.
- Prove that the 3-bit-code always reduces the irrecoverable error rate as long as $p < ½$.
- Argue that a 2-bit repetition code doesn't work.
- Argue that the 3-bit-code, albeit effective classically, cannot be applied to qubits.
  Hint: Its encoding relies on copying (cloning), and its decoding relies on taking a vote (measuring).

3. A Quantum Solution for Qubit Flip Errors

We will adopt the same error model for qubits as the classical one, i.e. we entertain only *qubit flip* errors (|0> ←→ |1>) with probability p and assume $p < ½$ and state-independent, and that errors affect qubits independently, i.e. no correlated errors.

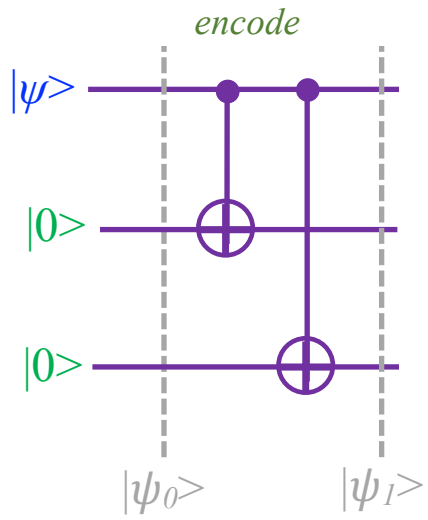
Although the classical 3-bit-code violates no-cloning and causes state collapse, we can adopt its key idea of creating redundancy by enlarging the space and adapt its encoding/decoding to the rules of quantum mechanics. Given a qubit:

$$|\psi> = \alpha|0> + \beta|1>$$

we embed its two-dimensional Hilbert space in a bigger, eight-dimensional space by adding two ancilla qubits initialised to zero.

$$|\psi_0> = (\alpha|0> + \beta|1>) \otimes |0> \otimes |0>$$

The classical code calls for copying $|\psi>$ into the two ancilla, but this is impossible here. (In fact, we can't even access the values of α and β.) What we can do, however, is entangle these three qubits together via CNOT gates. This would be the closest (legal) thing to copying a state from one qubit to another.



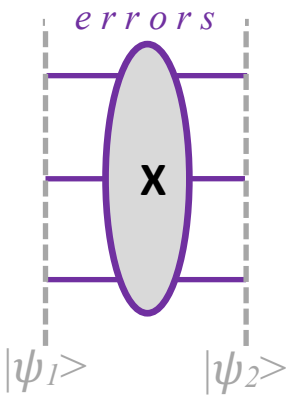
Based on this, we find:

$$|\psi_1> = \alpha|000> + \beta|111>$$

Note that this is not the same as copying:

$$|\psi_1> \neq (\alpha|0> + \beta|1>) \otimes (\alpha|0> + \beta|1>) \otimes (\alpha|0> + \beta|1>)$$

The error E will flip (i.e., apply an X gate to) at most one of the three qubits:



And depending on E (whether it occurred or not, and which qubit it flipped if it did, we have four possibilities:

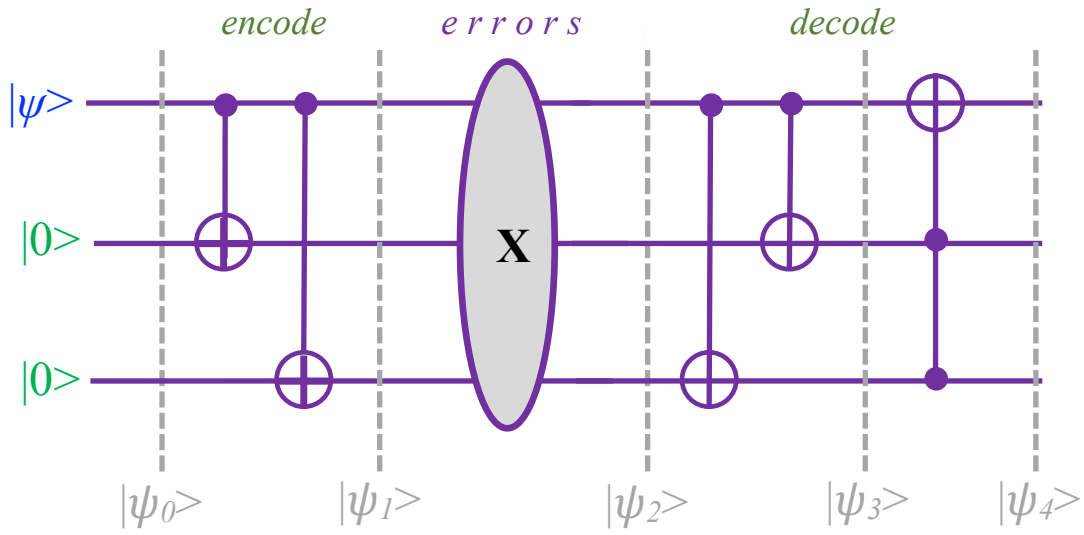
$$E = I \otimes I \otimes I \Rightarrow |\psi_{2a}> = \alpha|000> + \beta|111>, \text{or}$$

$$E = I \otimes I \otimes X \Rightarrow |\psi_{2b}> = \alpha|001> + \beta|110>, \text{or}$$

$$E = I \otimes X \otimes I \Rightarrow |\psi_{2c}> = \alpha|010> + \beta|101>, \text{or}$$

$$E = X \otimes I \otimes I \Rightarrow |\psi_{2d}> = \alpha|100> + \beta|011>$$

To detect the error, we cannot measure (to find the majority) as this would collapse the state (lose all information about of α and β). But we can compute the parity (xor) of the 1st and 3rd qubits, and that of the 1st and 2nd. These would be both zero if, and only if, no error occurred. This gives us the error syndrome *without* measuring. Moreover, the AND of the parities can correct the error, if present, in the 1st qubit via an xor (a Toffoli gate).



Let us perform the parity and correction in the four possibilities identified earlier:

$$|\psi_{2a}> = \alpha|000> + \beta|111>$$
$$|\psi_3> = \alpha|000> + \beta|100>$$
$$|\psi_4> = \alpha|000> + \beta|100> = (\alpha|0> + \beta|1>) \otimes |00>$$

or:

$$|\psi_{2b}> = \alpha|001> + \beta|110>$$
$$|\psi_3> = \alpha|001> + \beta|101>$$
$$|\psi_4> = \alpha|001> + \beta|101> = (\alpha|0> + \beta|1>) \otimes |01>$$

or:

$$||\psi_{2c}> = \alpha|010> + \beta|101>$$
$$|\psi_3> = \alpha|010> + \beta|110>$$
$$|\psi_4> = \alpha|010> + \beta|110> = (\alpha|0> + \beta|1>) \otimes |10>$$

or:

$$|\psi_{2d}> = \alpha|100> + \beta|011>$$
$$|\psi_3> = \alpha|111> + \beta|011>$$
$$|\psi_4> = \alpha|011> + \beta|111> = (\alpha|0> + \beta|1>) \otimes |11>$$

In all cases, the top qubit (the least significant in the ket) is *restored* and *de-entangled* from the syndrome in the two ancilla.

- Redo the 3-qubit-flip code in matrix form, i.e. instead of kets and gate operators, work with column vectors and matrices. For encoding, derive $|\psi_1\rangle$ from $|\psi_0\rangle$. For the error stage, you can derive $|\psi_2\rangle$ from $|\psi_1\rangle$ by assuming the error will flip the top qubit and applying the 8x8 matrix XII (X and two identity matrices). For decoding, derive $|\psi_4\rangle$ from $|\psi_2\rangle$.
- Performing a measurement amid a computation is typically risky as it may distort what we are computing. Is it OK to measure the two ancilla qubits after restoring the top qubit?
- Show that the 3-qubit-flip code can correct not only X on a single qubit but also any unitary superposition such as aI + bX where a,b∈C.
  Hint: measure the syndrome to collapse it.

4. A Quantum Solution for Phase Flip Errors

We adopt an error model in which only *phase flip* errors are allowed. We assume the phase of a qubit (the sign of one of its amplitudes) can flip (+ ←→ −) with probability *p*, and that $p < ½$ and sign-independent. We will retain the assumption of independence, i.e. no correlated errors.

Phase flip has no effect on standard basis states (the state $|0\rangle$ and the state $|1\rangle$) because a sign change in them is merely a global phase change. But given a superposition:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

with $\alpha, \beta \neq 0$, it will be transformed to:

$$|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$$

Hence, a phase flip amounts to applying the Z gate to the state whereas a qubit flip amounts to applying X as we saw in the previous section. But Z and X are connected by conjugation via H:

$$HZH = 1/\sqrt{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}1/\sqrt{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} = X$$

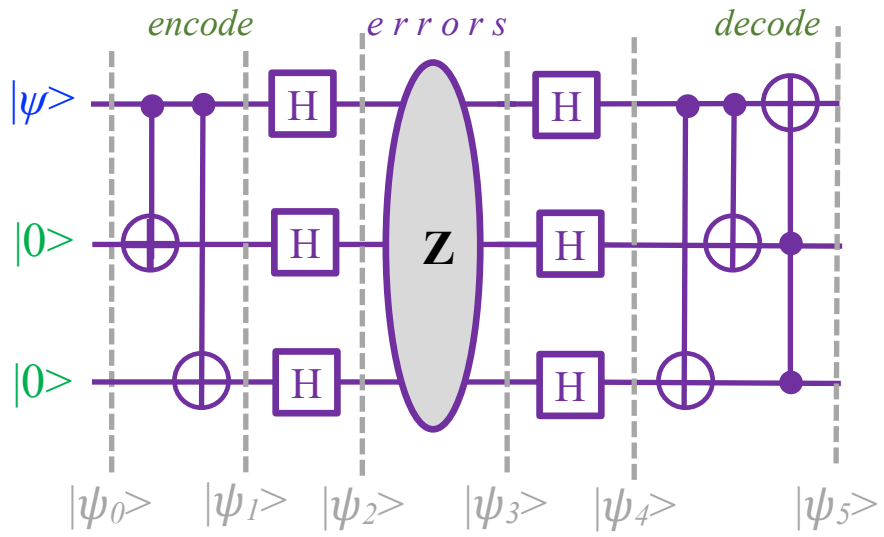
Hence, instead of coming up with a new algorithm for correcting phase flips, we can "transform the error" from a phase flip to a qubit flip by "sandwiching" between Hadamard gates! Another way to think about this is to note that the similarity transformation HZH that connects Z with X makes Z "looks like X" when viewed from a basis rotated by H. From this perspective, we would transform the representation of our state from the 0/1 basis to the +/−:

The phase flip in 0/1: $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle - \beta|1\rangle$

looks like: $(\alpha+\beta)\,|+\rangle + (\alpha-\beta)\,|-\rangle \rightarrow (\alpha-\beta)\,|+\rangle + (\alpha+\beta)\,|-\rangle$

which is a qubit flip: $|+\rangle \rightarrow |-\rangle$ in +/−.

This observation leads us to the following circuit:



As before, we can see that:

$$|\psi_0> = (\alpha|0> + \beta|1>) \otimes |0> \otimes |0>$$

$$|\psi_1> = \alpha|000> + \beta|111>$$

$$|\psi_2> = \alpha|+++> + \beta|---|$$

$$E = I \otimes I \otimes I \Rightarrow |\psi_{3a}> = \alpha|+++> + \beta|---|, \text{or}$$
$$E = I \otimes I \otimes Z \Rightarrow |\psi_{3b}> = \alpha|++-> + \beta|--+|, \text{or}$$
$$E = I \otimes Z \otimes I \Rightarrow |\psi_{3c}> = \alpha|+-+> + \beta|-+-|, \text{or}$$
$$E = Z \otimes I \otimes I \Rightarrow |\psi_{3d}> = \alpha|-++> + \beta|+--|$$

$|\psi_4>$ is the same as $|\psi_3>$ with +/- replaced with 1/0. Hence, the syndrome is captured as in the qubit flip case and the top qubit is corrected accordingly. $|\psi_5>$ will thus have the top qubit restored and de-entangled from the syndrome in the two ancilla. In fact, we can restore all qubits to their original $|\psi_0>$ state by adding two more CNOT gates to set the ancilla qubits.

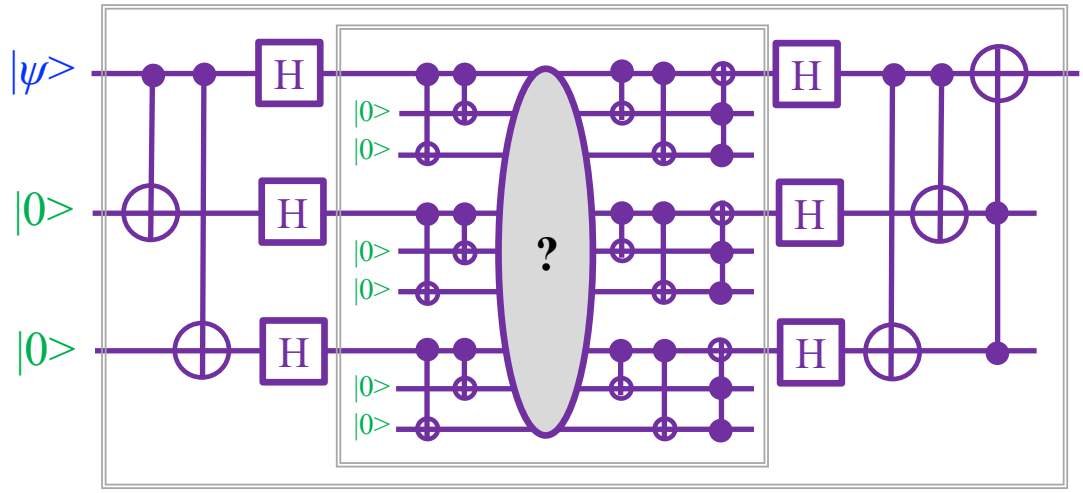
- Show that the 3-qubit-phase flip code can correct not only Z on a single qubit but also any unitary superposition such as aI + bZ, a, b∈C.
  Hint: measure the syndrome to collapse it.
- Show that the 3-qubit code for phase flips can correct not only 180° phase shifts (i.e. Z) but also any phase shift by an angle $\boldsymbol{\theta}$; i.e. E = [[1, 0] , [0, $e^{i\boldsymbol{\theta}}$]].
  Hint: rewrite the error matrix as a linear combination of I and Z.

5. A Quantum Solution for ALL Single-Qubit Errors

We found that the 3-qubit code can correct any superposition involving at most one X. We also found a modified version of it that can correct any superposition involving at most one Z. But if the error involves both X and Z (on the same or on different qubits) then neither version of this code can correct such an error. The *Shor code* builds on the 3-qubit code by combining its X and Z capabilities. Specifically, it calls for encoding the original qubit as in the phase-flip Z case using 2 ancilla qubits, i.e.

$$|\psi> \rightarrow (\alpha|0> + \beta|1>) \otimes |0> \otimes |0> \rightarrow |+++> + \beta|---\rangle$$

We then encode each of the three qubits in the above codeword as in the bit-flip X case using 2 ancilla qubits. These leaves us with 9 qubits in total. Each 3 will then undergo bit-flip correction, followed by a Hadamard on the top qubit, followed by phase flip correction. The final top qubit will be de-entangled and the same as the original top qubit, as shown in the diagram below.



Note that the error region above could involve no errors at all; an X on one of the nine qubits; a Z on one of the nine qubits; or both X and Z on one of the nine qubits. The last case is the same as having a Y because iY = XZ. Hence, this code can correct any I, X, Y, Z error on one qubit.

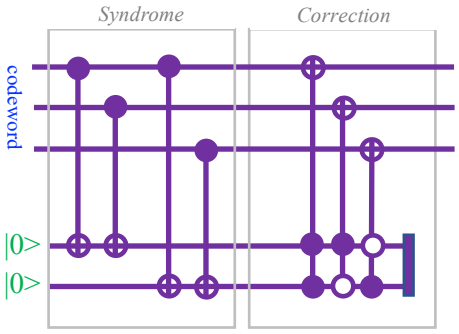
Note also that if the error involves a linear combination of I, X, Y, Z then it too can be corrected through a measurement that collapses the syndrome and accordingly restores the top qubit.

Finally, since the set I, X, Y, Z is universal (any single-qubit matrix can be written as a linear combination of them), this code can correct *any* single qubit error.

The above observations show that quantum error correction (QEC) is *possible* despite the seemingly unsurmountable challenges: the inability to copy and the inability to measure. Moreover, QEC is *manageable* despite the continuous nature of quantum errors (arbitrary superpositions involving complex numbers rather than all-or-nothing classical bit flips) thanks to the *Error Discretization phenomenon* (correcting the discrete I, X, Y, Z corrects any continuous error).

We saw that quantum codes can correct errors that occur between the encoding and decoding stage. Our decoding circuit for the Shor code successfully corrected the top physical qubit using the syndrome in the other two. This means we need to reencode our logical qubit before it proceeds to later stages, which leaves the qubit unprotected between decoding and reencoding.

To guard against that, we modify our decoding stage, so it computes the syndrome in two fresh ancilla qubits (rather than in the 2 ancilla qubits of the codeword). And based on the computed syndrome, we correct *all three* qubits in the codeword, not just the top one. We illustrate this in the diagram below:



The Toffoli gates are controlled by the bottom two ancilla qubits by ANDing them with or w/o negation (a white or a filled circle). By combining correction with re-encoding in one step, we avoid exposing our logical qubit to errors in between the two.

- Use Qiskit to set up a circuit for the 3-qubit-flip code and run it to verify that this code does indeed correct an error in any one of the three qubits (select it randomly).
- Same as above but for the error linear combination: E = (3 IXI + 4 IXI)/5
- Use Qiskit to set up a circuit for the 3-qubit-phase code and run it to verify that this code does indeed correct an error in any one of the three qubits (select it randomly).
- Use Qiskit to set up a circuit for the 3-qubit Shor code. To trace the progress, insert the line:
  `circuit.save_statevector(label='v1')`
  after each stage in the algorithm (use different labels v1, v2, … for each). This will insert a barrier in the circuit diagram and capture the state at that point. You can then output the progress of the state using result.data(0)['v1'].

6. But Why do Errors Occur?

The first reason is easy to understand: imprecision in the gates. For example, the X gate rotates the qubit by a 180° (in the Bloch sphere) around the x-axis. It is typically implemented by turning on a control signal (microwave, magnetic field, etc.) long enough for the qubit to rotate by that much. But if the control duration were a bit longer or shorter than necessary, the rotation

angle θ would not be exactly 180° and the final state would not be as expected. Applying such an imprecise X to a qubit in state |0>, would yield:

$$\cos(\theta/2)\,|0> - i\sin(\theta/2)|1>$$

rather than |1>. Note that the transformation from |0> to the above is still unitary; it is just not what we expect. Errors due to imprecision in the gates are known as *coherent* because the state of the qubit, albeit incorrect, is pure—not entangled with the environment.

The second reason is more subtle: the qubit would change its state in between gates without any apparent reason. Moreover, the change is not unitary which doesn't make sense since all physical systems should evolve unitarily no matter what. What is happening here is that the isolation of our qubits (from the outside world) was breached. For example, a photon (a lump of energy) may have been emitted by the environment surrounding the computer, and that photon may have entered the computer and gotten absorbed by our qubit, thus forcing it to change its state. Hence, the "error" is in our assumption that our qubits are a closed system. Had we considered our qubits-plus-the-environment as one system, we would have found it evolving unitarily as expected.

To get a feel for the second reason, consider a single qubit $|\psi>$ and an environment $|E>$. When there is no interaction between them, the state of the "qubit-plus-environment" system would be a de-entangled product state: $|E>\otimes|\psi>$. But when they interact, the state would no longer be separable. Specifically, if the environment was in state $|E_1>$ and the qubit in state |0> e before the interaction, then they will become entangled thereafter:

$$|E_1> \otimes |0> \rightarrow c_1|E_1 0> + c_2|E_2 1>$$

where $|E_{1,2}>$ are environment states, and $c_{1,2}$ are amplitudes (complex numbers). You can think of the 2nd term as capturing the event in which the environment emits a photon and transitions to state $|E_2>$, and our qubit absorbs that photon and switches to state |1>. The first term is for the event in which no such photon is emitted. Similarly, if our qubit started in |1>:

$$|E_1> \otimes |1> \rightarrow\ c_1|E_1 1> + c_2|E_2 0>$$

Think of this as no interaction at all or a photon emitted by our qubit and absorbed by the environment. Admittedly, the two amplitudes here need not be the same as the ones for when the photon is transferred to the qubit, but we will keep things simple. Finally, if our qubit started in the superposition:

$$|\psi> = \alpha|0> + \beta|1>$$

then, by linearity:

$$|E_1> \otimes |\psi> \rightarrow c_1|E_1> (\alpha|0> + \beta|1>) + c_2|E_2> (\alpha|1> + \beta|0>)$$

Or in terms of operators acting on our qubit state:

$$|E_1> \otimes |\psi> \rightarrow [\ c_1|E_1> \mathrm{I} + c_2|E_2> X\ ]\ |\psi>$$

Hence, this simple photon exchange leads to our qubit being entangled with the environment. If we look only at our qubit and "average away" the environment states, we see that our qubit lives in a mixed (not pure) state, and it evolves non-unitarily. The introduced errors in this case are thus *incoherent* and the qubit is said to have undergone *decoherence* or *dephasing*.

This behaviour "explains" errors involving qubit flips. In more realistic models, in which the environment has many more states, the qubit will end up in a mixed state involving a linear combination I, X, Y, and Z errors. Note that if $c_1|E_1\rangle = \lambda c_2|E_2\rangle$ (where λ is any constant) then the above becomes a product state, so the qubit de-entangles from the environment and evolves unitarily. This models the imprecise rotation control type of error that we covered earlier.

- Let us generalize the qubit-environment interaction model above by using 4 states: $E_1 \ldots E_4$ for the environment and 4 different amplitudes:

  $|E\rangle|0\rangle \rightarrow c_1|E_1\rangle|0\rangle + c_2|E_2\rangle|1\rangle$, $|E\rangle|1\rangle \rightarrow c_3|E_3\rangle|1\rangle + c_4|E_4\rangle|1\rangle$

  Show that when the qubit starts in $|\psi\rangle = a|0\rangle + b|1\rangle$ then

  $|E\rangle|\psi\rangle \rightarrow \frac{1}{2}[\, A.I + B.Z + C.X - iD.Y \,]\,|\psi\rangle$, where

  $A = c_1|E_1\rangle + c_3|E_3\rangle$, $B = c_1|E_1\rangle - c_3|E_3\rangle$
  $C = c_2|E_2\rangle + c_4|E_4\rangle$, $D = c_2|E_2\rangle - c_4|E_4\rangle$

7. Fault Tolerant Computing

Our strategy so far has been to start by encoding the input to the circuit by turning each logical qubit into a codeword made up of a number of physical qubit (3 in the Shor's code). The codeword is then decoded before each gate and re-encoded after. This guards against (i.e. corrects) errors that occur in between gates *but not ones that occur within gates or within the encoding and decoding stages.* For example, if an error occurred in a single ancilla qubit that computes the syndrome of the Shor code, then all three "corrected" qubits in the codeword would be in error. In such a case, error correction introduces more errors than it corrects: it got an error on a single qubit error (which is recoverable) and produced simultaneous errors in 3 qubits (which is irrecoverable)! Hence, in addition to error correcting codes, we need techniques to prevent the proliferation of errors, and thus make *fault-tolerant* [FT] computing possible.

The most commonly used techniques involve:

- Keeping the qubits encoded at all times, and
- Limiting the irrecoverable error probability per gate to $O(p^2)$.

The first technique adds redundancy within gates. All gates would have to be modified to take encoded input and produce encoded output. With such gates, our circuit would consist of one encoding stage for the input; error correction in between gates; and one decoding stage for the output. The second technique imposes a numerical constraint on the design of the FT gates. If a gate G produces an irrecoverable error with probability p, then its fault-tolerant version G_{FT} must have an irrecoverable error probability of at most cp^2, where c is some constant.

FT gates reduce the error probability if $cp^2 < p$, which happens if $p < 1/c$. This is why 1/c is called the *threshold* error rate. A circuit involving n FT gates would thus fail with probability ncp^2. This seems to imply that quantum computing is not practical, because n is typically very large, but a powerful additional technique exists: *concatenation*. If making the circuit fault-tolerant reduced its failure rate from p to cp^2, then why not do so recursively? This means each of the n qubits in a codeword will be encoded into n qubits, thus turning the original logical qubit into n^2 physical qubits. The new circuit (after one such iteration) would reduce the failure probability from cp^2 to $c(cp^2)^2 = c^3p^4 = (1/c) \times (cp)^4$. If we concatenate k times, the failure rate becomes:

$$(1/c) \times (cp)^{2^k}$$

This amazing result, known as the *threshold theorem*, proves, not only that quantum computing is feasible, but that it can be done to arbitrary accuracy. Moreover while the number of physical qubits increases exponentially with k, the failure rate decreases double exponentially with it.

Remarks

- The physical implementation of qubits is characterized by the so-called T1 and T2 times.
- T1 is known as the relaxation time. A qubit in state |1> in an environment tends to drop (relax) to|0> after some time, and T1 is a measure of that.
- T2 is known as the dephasing time. It is a measure of the time it takes the phase of a qubit to stay intact (before an interaction with the environment alters it).
- Note that T1 (and T2) is not a transition duration but rather a transition constant c. In fact, the probability that a qubit would stay in its state after a time t is given by formula $e^{-t/c}$.
- Both times T1 and T2 are together called *decoherence times*.
- Staen Code –CSS –Stabilizers –Topological codes
- The Shor code uses 9 physical qubits per logical qubit. The most commonly used code is the Steane code, and it uses 7 physical qubits and has a 10^{-5} threshold.
- Surface (aka topological) codes have a 10^{-2} threshold. Using them to factor a 2000-bit number requires about 10^8 qubits.
- Today's quantum computers have less than 100 qubits! *(Compare 10^2 to 10^8.)*
- The number of transistors in a classical CPU was a few thousands in 1970. It is about a thousand billion today! *(Compare 10^3 to 10^{12}.)*