December 24, 2022

Quantum Error Correction

Aayush Verma

Scheme for reducing decoherence in quantum computer memory
Peter W. Shor
Phys. Rev. A 52, R2493(R)
Published 1 October 1995

The idea of a quantum circuit and a quantum computer is a noble one. But the issue with such quantum computations is that it has decoherence. Decoherence results in the qubits interacting with the environment and collapsing. We can overcome this error by using quantum error correction techniques, in particular, using $9k$ qubits to store k qubits in a decoherence-free way. First, we start with encoding these k qubits in 9 qubits by

$$|0\rangle \to \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle), \tag{1}$$

$$|1\rangle \to \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle). \tag{2}$$

Moreover, we can read these qubits using auxiliary qubits by entangling them. We measure these nine qubits in their respective Bell states and count the majority of three triplets; in this way, we can know if the encoded qubit was 0 or 1. Now, the decoherence works something like follows. Let us take first qubit and assume that it de-coheres. The encoded qubit 0 and 1 will form basis for this first qubit

$$|e_0\rangle |0\rangle \to |a_0\rangle |0\rangle + |a_1\rangle |1\rangle \tag{3}$$

$$|e_0\rangle |1\rangle \to |a_2\rangle |0\rangle + |a_3\rangle |1\rangle \tag{4}$$

where a_n are states are associated with the environment. If we take just 0 encoded qubit superposition, the initial superposition results through decoherence in

$$(1/\sqrt{2})\left[(|a_0\rangle |0\rangle + |a_1\rangle |1\rangle) |00\rangle + (|a_2\rangle |0\rangle + |a_3\rangle |1\rangle) |11\rangle\right] \tag{5}$$

and in bell-basis, this is

$$
\begin{aligned}
&\frac{1}{2\sqrt{2}}\left(|a_0\rangle + |a_e\rangle\right)\left(|000\rangle + |111\rangle\right) + \\
&\frac{1}{2\sqrt{2}}\left(|a_0\rangle - |a_3\rangle\right)\left(|000\rangle - |111\rangle\right) + \\
&\frac{1}{2\sqrt{2}}\left(|a_1\rangle + |a_2\rangle\right)\left(|100\rangle + |011\rangle\right) + \\
&\frac{1}{2\sqrt{2}}\left(|a_1\rangle - |a_2\rangle\right)\left(|100\rangle - |011\rangle\right)
\end{aligned}
\tag{6}
$$

Similarly, we can decohere the first qubit in encode qubit 1 for the $|000\rangle - |111\rangle$. Since we know the original state of encoded qubits and now the decohered states of the first qubit, we can create an auxiliary qubit to measure the error and correct them. The superposition in the Bell state basis will remain to exist.

The way how to catch the error is given by checking the triples through a unitary computation in a quantum computer. The measure of this error is given by checking the auxiliary qubits. The computation first compares the triples in the Bell basis, and if the triples are the same (before and after the decoherence), then there was simply no decoherence in the qubit. However, if one finds the triples are not the same, then there is decoherence for which the quantum computer must restore it to the originally encoded state. See, for example, that in the second line of the decohered Bell state, the vectors have the wrong sign. Also, in the third and fourth line, the first qubit is wrong

If more than one qubit decoheres, this encoding and correction does not work. But given by Shor, the probability of two qubits decoheres is $36p^2$ and of $9k$ can be decoded to give the original state is $(1 - 36p^2)^k$. This scheme is suitable for a decoherence with a probability less than $1/36$.

The problems with this scheme are as follows. Since we increase the qubits from k to $9k$, there are a large number of qubits. The unitary computation of a computer is not precisely defined. Thus, the unitary computation of detecting the error implies a bit error itself. With every check of error, we inject a small error. For this reason, we can not store the qubits for a long time.

R. W. Hamming,
Error detecting and error correcting codes,
The Bell system technical
Journal 29 no. 2, (1950) 147–160.

A very simple error-free coding would be repetition. For instance, if Alice wants to send a message 1, she would simply send 111, a repetition of 3, if Bob does not get 111 but, for example, 101, then an error has occurred. However, such simple code does not work for many problems. Hamming code was introduced for single-bit error correction. Hamming codes work as parity-check codes.

We write numbers in binary form (base 2), 1 is 1, 2 is 10, 3 is 11, 4 is 100, 5 is 101 and so on. 'Parity bits' are those bits that are powers of 2 and contain only one 1 bit in the binary form, for example, $1, 2, 4, 8, 16$. Bits other than parity bits are called 'data bits'. There is two parity - even and odd. Suppose Alice wants to send 1101001; there are four 1-bit which is an even number. So Alice would attach, in this case, 0 at the end to make it 8-bits with parity making it to 11010010[1]. Then Alice computes the modular arithmetic (in mod 2), which, in this case, is 0. Then Alice transmits the 8-bits to Bob and Bob, after receiving it check the modular arithmetic (in mod 2) of 8 bits which is also 0. Bob confirms the parity of decoding to be even. So there was no error in the transmission. Transmission with odd parity is just the same, the reader should verify it.

The choice of parity is just a choice, but the sender and receiver must know the chosen parity. Hamming code uses the parity-check concept in it. One might realize that parity-check has a critical failure in determining the error in even parity as if Alice sends 1010 and Bob receives 1100, then Bob is not aware of the error. The bits that were replaced are called corrupted bits. In Hamming code, one has a generator matrix G (for encoding) and a parity-check matrix H. A message by Alice is first encoded with the generator matrix, then a parity bit is added in the way we did, then it is transferred to Bob, and Bob does the parity-check. Another, a more intuitive and simple, way to see Hamming code is through the Venn diagram

[1]One can say that it is encoding. After it reaches to Bob, he would decode it