Qcamp: Understanding The Error Correction Function

Group 4

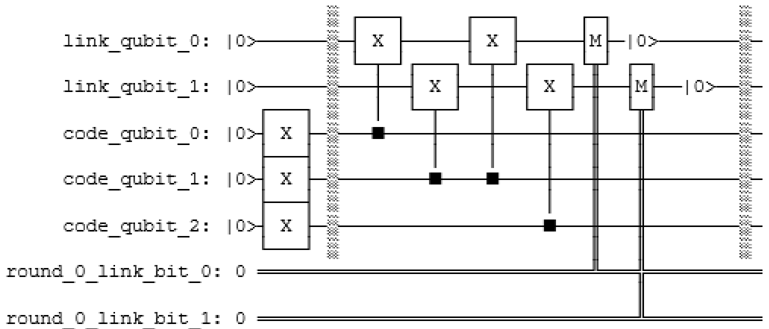
December 2019

1 Introduction

Quantum circuits in error correction codes are prone to noise which are caused by either a bit flip after applying a given gate or by measuring the link qubits.

Decoding the code qubits is a crucial part of the error correction process.Therefore the following decoding algorithm can be used to test if there is any form of noise in the circuit, where it is and how it affects the logical readout.The code can still be improved further.

Description of the decoding process
Suppose we have the following logical 1 qubit operations.



In error correction we have the code qubits, link qubits and rounds (which in this case imply the classical bits where the output of the syndrome measurements are stored).

Generally, measurements are usually done on qubits to be able to obtain their value at the given point but this makes the given qubits collapse into a classical state and this will limit us from going on with the original operations. There is need to tell if errors have occurred while still carrying out any algorithm.this is where error detection and correction comes in.

In error correction codes, we have the code qubits which represent the repetition codes and the link qubits which are less code qubit by one.i.e link qubits= n, code qubits =n+1.this enables the error correction process to detect errors using the syndrome measurement which is applied in the link qubits.this allows us to detect points where the error has occured which type of error it is and how to correct the error. The possible errors are bit flip and measurement error.

Bit flip errors are noted if the susequent measurements differ for a given qubit while measurement error is noted when a given error is not noted in subsequent measurements.

The number of syndrome measurements are usually represented by 'T' and the number of measurements done are not restricted. Link qubits are usually reset to 0 if the result at the end of a given measurement are not all 0 so that more measurements can be carried out.

Suppose we have the following results after a given number of measurements:
Example 1:

1 0 001 100 100

1. 1 0 represent the logical readouts
2. 001 100 100 represent the various measurements

The 1 0 in 1 above represents the logical readout which in a perfect scenario should be either 1 1 or 0 0.
Every bit in 2 above represent the difference between a pair of subsequent qubits eg.

1. From 001 we are able to tell that $d_1 = d_2 = d_3 \neq d_4$
2. From 100 we are able to tell that $d_1 \neq d_2 = d_3 \neq d_4$
3. From 100 we are able to tell that $d_1 = d_2 = d_3 \neq d_4$
   (a) In 2 that $d_1 \neq d_2 = d_3 \neq d_4$ d_1 flipped while d_4 remains unequal to the rest since the error was carried forward.
   (b) In 3 d_1 flips again showing us that the bit never flipped to begin with and hence we take it as a measurement error.
   (c) d_4 remains constant since there is no change between 2 and 3 above

Example 2:

0 0 000 001 001

1. 0 0 represent the logical readouts
2. 000 001 001 represent the various measurements

The 0 0 in 1 above represents the logical readout which would give a logical bit of 0.
Every bit in 2 above represent the difference between a pair of subsequent qubits eg.

1. From 000 we are able to tell that $d_1 = d_2 = d_3 = d_4$
2. From 001 we are able to tell that $d_1 = d_2 = d_3 \neq d_4$
3. From 001 we are able to tell that $d_1 = d_2 = d_3 = d_4$
   (a) In 1 no error occurred
   (b) In 2 a bit flipped on d_4
   (c) In 3 d_4 flipped again showing us that is was probably a measurement error.

A better understanding of the logical process behind the error detection and correction through the minimum weight perfect matching method is required to further break down and understand the decoding process hence improve error correction.