Quantum Hamming Code and Quantum Reed-Solomon

Lorenzo Cavuoti — lorenzo.cavuoti@gmail.com
Davide Incalza — davide_incalza@aol.com
Debora Ramacciotti — d.ramacciotti@outlook.com
Francesco Sacco — francesco215@live.it
Zhiyong Zhang — zyzhang@stanford.edu

June 4, 2021
Qiskit Hackaton Europe 2021

Abstract

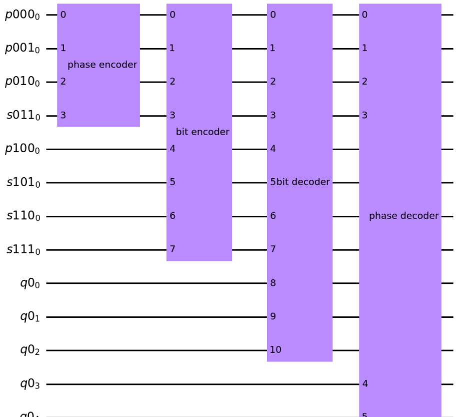
Unlike bits in classical computing, qubits are subject to large errors due to interactions with the physical environment, as well as the intrinsic decoherence effects of the superposition of the quantum states of a qubit. Quantum error correction protocols is central to the realisation of the power of quantum computing and quantum communication. The goal of this project is to develop and implement quantum versions of the Hamming Code, one of the first efficient classical error correcting code, and of the Reed-Solomon code, nowadays the most used ECC for classical computers, as well as to simulate and explore the error correction capacities of the code on quantum computers with error and noise profiles obtained from real quantum computing devices.

I. Quantum Hamming Code Implementation

Authors: Lorenzo Cavuoti, Francesco Sacco

The classical Hamming code is an error-correction algorithm with distance $k = 3$ that encodes $2^N - N$ bits with N parity bits to detect and correct for bit flips. The quantum version of the code is more complex since it also needs to account for phase flips of the encoded qubits.Adding a set of Hadamard gates at the end of the encoder and at the beginning of the decoder extends the bit-flip code, (the code that corrects the bit flip errors present only in classical computers), to also correct for phase-flip errors. If no error occurs the gates do nothing since $HH = I$. In the case of a phase flip, since $HZH = X$, the error becomes a bit flip that can be corrected with the Hamming code for bit flips. The Hamming codes that corrects bit-flip and phase flips form a complete basis to correct for any errors that can be described with unitary transformations/matrix es.

The bit flips and phase flips errors can be represented by the X and Z gates respectively. Since Y=-iXZ, and any single qubit error can be represented as linear combinations of the operators X, Y, Z, thus any error acting on single qubits can be corrected with the algorithm, assuming that errors act independently on each qubit.The same procedure described here can be generalized to any classical error correcting code for extensions to quantum error corrections.



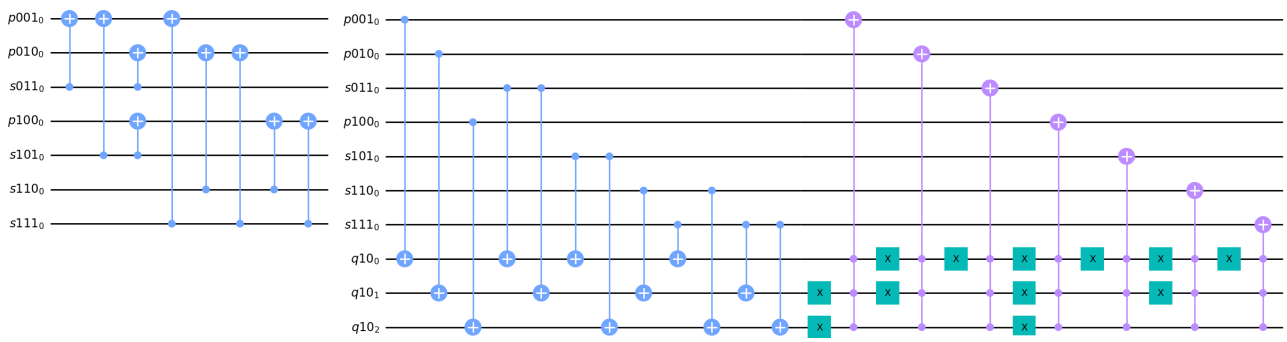


Figure 1: *Bit flip Hamming encoder on the left, Bit flip Hamming decoder on the right*

II. Quantum Reed-Solomon Implementation

Authors: Davide Incalza, Debora Ramacciotti

The goal is to implement a generic Reed-Solomon code, starting from a classic one over the finite fields of characteristic two.

Analogously to the notation $C = [N, K, d]$ for a classical error-correcting code encoding K information symbols using N bits with minimum distance d, a quantum code encoding K qubits using N qubits, such that any error of weight less than $d/2$ can be corrected, is denoted with $Q = [N, K, d]$.

The classical Reed-Solomon code over the field $\mathbb{F}_{2^k}$, by definition, has parameters: $N = 2^k - 1$, $K = N - d + 1$ and $d > N/2 + 1$. From this one, its quantum counterpart can be constructed as $Q = [kN, k(N - 2K), d > K]$. Notice that all the parameters can be derived only from k. The trivial example is the one sending three qubit, i.e. the code $Q = [21, 3, 5]$.

The encoder and the circuit to compute the syndrome are shown in the following images, and the theory behind them is well explained in [1].

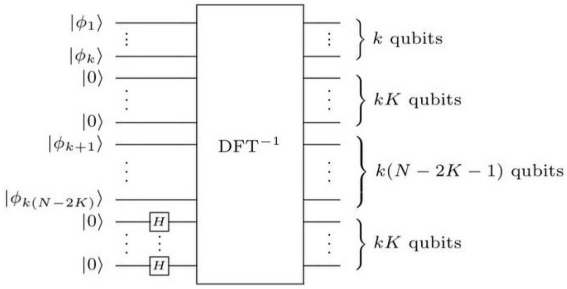


Figure 3: *RS Encoder*

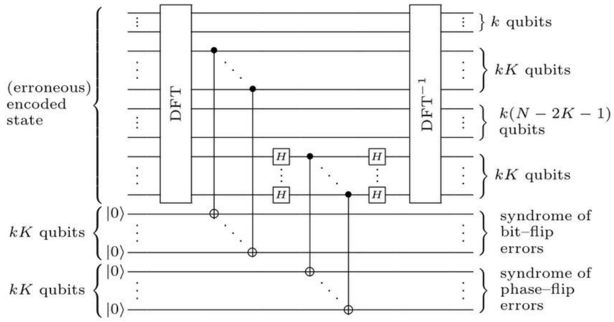


Figure 4: *RS Syndrome computation circuit*

The encoder takes the message Φ of length $k(N - 2k)$ and transforms it in one of length kN thanks to the Quantum Fourier Transformation, implemented with *qiskit.circuit.library.QFT* [2].

After that the noise acts on the system, the second circuit takes the erroneous message and computes the syndrome on the last $2kK$ qubits. Measuring the two syndromes allow us to use already known classical algorithms to compute the position of the errors: in particular, we adapted the Berlekamp-Massey algorithm and the Chien search to our purposes, using the existent unireedsolomon library [3].

The Berlekamp-Massey algorithm and the Chien search were merged into one function that returns the string with the positions of the errors. Calling this function (*error_string*) two times, results in having two strings: one with the position of the bit-flips and the other one that locates the phase-flips.

Finally, bit-flip errors and phase flips are corrected, respectively, with X gates and with Z gates inside Hadamard transforms. Applying another QFT to the first kN qubits returns (hopefully) the original message.

III. Study with the Noise Model

Authors: Zhiyong Zhang

To test the two error-correcting codes, we ran simulations by adding X, Y or Z gates to random qubits between the encoder and decoder. The tests were made on pure states as well as entangled ones, and, when the errors were not too many, the retrieved message was corrected.
Then, a noise model was added to the simulator in order to test the behaviour of the codes on a real quantum device, in particular the `ibm_16_melbourne`, a quantum computer given by the IBMQ provider.

i. Quantum Hamming Code

Without the noise model the expected result of the Hamming code with one phase and bit flip after sending the message $s = (|101\rangle + |010\rangle)/\sqrt{2}$ is "101" and "010" with equal probability. If the code works correctly, this have to happen also if we apply up to one bit-flip and one phase-flip to the encoded qubits, as confirmed by our simulations. When a realistic noise model is added a percentage of measurements different from "010" and "101" is observed, as shown in the histogram below.

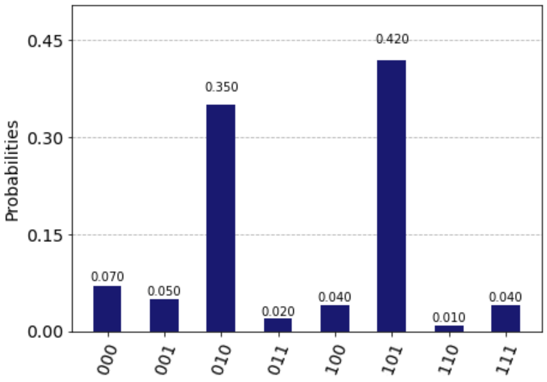


Figure 5: *Results of Hamming with noise*

ii. Quantum Reed-Solomon

For the noiseless Reed-Solomon code, the expected result after sending the string "110" for a three message qubits, is to retrieve the same string at the output, with probability one. When a realistic noise model is added, a percentage of measurement results other than 110 is observed, as shown in the histogram below.

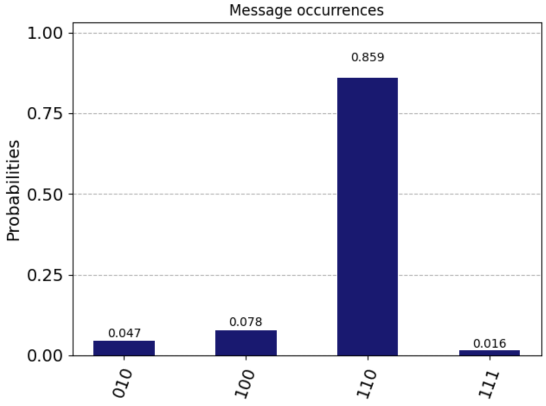


Figure 6: *Results of Reed-Solomon with noise*

Notice that, with error correction, the measurement results are expected to be almost identical to the ideal results obtained with no noises. In the actual results, we observe that the probability of "110" in the error-corrected result is 0.859, still significantly less than the ideal result of 1, due to the fact that additional errors are introduced in the circuit during the phases of error detection and correction. Although the circuit exhibits noise resistance, this indicates the difficulty in operating error correction algorithms in real devices due to the decoherence.

IV. Future Impact

Authors: The whole team

- The quantum version of the Hamming code can be useful when the channel is not too noisy, since it can correct up to one error acting on one single qubit. However, as the order of the code increases the Hamming code becomes more and more efficient since it only needs $2N - 1$ parity qubits to encode $2^N - 2N + 1$ qubits.

- The Reed-Solomon code is one of the most efficient error-correcting solutions for classical computers and we expect its quantum counterpart can be proven useful in the future.

- The Reed-Solomon procedure for encoding and for computing the syndrome, based on spectral techniques, is very general. In fact, it can also be applied to any cyclic code.

- Since the RS code is completely general, depending only on k, this procedure could be applied in concatenated coding.

V. Bibliography

[1] M. Grassl, W. Geiselmann, and T. Beth, Quantum Reed–Solomon Codes, In: M. Fossorier, H. Imai, S. Lin, A. Poli (eds) Applied Algebra, Algebraic Algorithms and Error-Correcting Codes. AAECC 1999. Lecture Notes in Computer Science, vol 1719. Springer, Berlin, Heidelberg.

[2] Qiskit QFT: `https://qiskit.org/documentation/stubs/qiskit.circuit.library.QFT.html`

[3] Unireedsolomon library: `https://github.com/lrq3000/unireedsolomon`