Universitat Autònoma de Barcelona
Qiskit Hackathon

BSC Challenge: The Last Circuit Cut

Q-BUDDIES
Gabriel De Araujo
Oriol De Miguel
Gokarna Dumre
Ho June Kim
Jorge Plazas

Date: April 2023

First step

The cut had to be made in order to completely split the circuit in two parts without separating or breaking a CNOT gate. The qubits had to be rearranged into the position of the figure 1. The red line marks the spot where we cut the circuit into two parts and both parts of the original circuit can be seen at figures 2a and 2b.

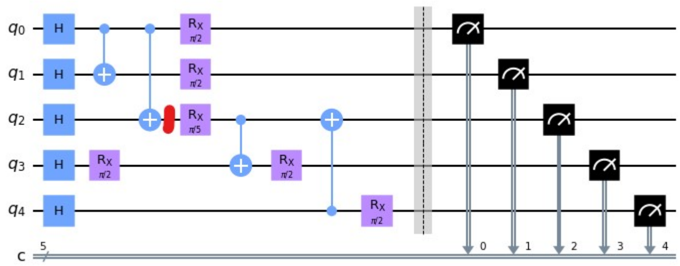

Fig. 1: Diagram of the circuit.

(a) First part of the circuit.

(b) Second part of the circuit.

Fig. 2: Slices of the circuit after the wire is cut.

Second step

For each cut, there are four expected values computed for the first part of the circuit (for the identity and the Pauli matrices) and the second part of the circuit is simulated eight times, twice for each of the previous matrices. The reason why it is computed twice is because each two dimensional matrices has two eigenvectors. We can observe that, with each cut, the number of combinations scale quickly. More exactly, the number of measurements scale exponentially with the number of cuts.

When the initial circuit was cut, we observed that Z and I were written using the same basis, therefore, its eigenstates are the same and only have to be computed once. The number of states that need to be prepared can also be reduced using the fact that the eigenstates of the Pauli matrices can be written as a linear combination of the Z basis. Therefore, knowing how the circuits transforms each vector of this basis, we can calculate the expected values for Y and Z. Therefore, we are able to write the expected value as a bra-Pauli tensor-ket product. Since both bra and ket can be written as a linear combination on 0 and 1 ket, only the action of the circuit onto those vectors will be needed to find all the values. The Pauli tensor for the X matrix acts only as an "inversor" of the coefficients of each vector, since it swaps the 0 by the 1 and viceversa. That is the reason why the function reverse qargs is called in the code. From there, only the inner product has to be computed having in mind all the normalization constants. The equation 1 is an example of how this method can be applied to measure the expected value of $\sigma_x \otimes \sigma_x \otimes \sigma_x$ to $|+00\rangle$ where M is the operator that represents the second part of the circuit and $|M_0\rangle$ and $|M_1\rangle$ are the resulting vectors of applying $|000\rangle$ and $|100\rangle$ to the second part of the circuit, respectively.

$$\langle \sigma_x \otimes \sigma_x \otimes \sigma_x \rangle_{M|+00\rangle} = \langle +00| M \sigma_x \otimes \sigma_x \otimes \sigma_x M |+00\rangle = \frac{1}{2}(\langle 000| + \langle 100|) M \sigma_x \otimes \sigma_x \otimes \sigma_x M (|000\rangle + |100\rangle)$$

$$= \frac{1}{2}(\langle M_0| + \langle M_1|) \sigma_x \otimes \sigma_x \otimes \sigma_x (|M_0\rangle + |M_1\rangle)$$

$$= \frac{1}{2}(\langle \sigma_x \otimes \sigma_x \otimes \sigma_x \rangle_{|M_0\rangle} + \langle \sigma_x \otimes \sigma_x \otimes \sigma_x \rangle_{|M_1\rangle} + \langle M_0|\sigma_x \otimes \sigma_x \otimes \sigma_x|M_1\rangle + \langle M_1|\sigma_x \otimes \sigma_x \otimes \sigma_x|M_0\rangle) \tag{1}$$