Quantum Circuit Optimization Based on Entanglement and Hardware Design

Kevin J. Joven

***Abstract*—In this work that I developed during the QOSF Mentorship program cohort 7 - QOSF, we take into consideration the entanglement of the circuit to modify the connections of the qubits and improve the performance of the circuit. We implemented an entanglement matrix on the quantum full adder (QFA) circuit for two qubits to see how much the qubits are entangled with others. Based on that amount of entanglement, modify the architecture to improve performance.**
The second work is to design a honey architecture lattice to implement a quantum chip using Qiskit Metal. The implementation takes the size of the lattice from the x and y directions and packs the number of qubits allowed in the architecture. All the design is based on transmon-like-qubits non-tunable, and each of them has its own control line.

Keywords: Entanglement, Quantum Hardware

Quantum circuit optimization, a crucial aspect of quantum hardware design, seeks to allocate qubits with minimal errors in one and two-qubit gates, enhancing quantum computation performance.

I. Circuit Entanglement Optimization

Entanglement is one of the fundamental properties of quantum mechanics and is useful in quantum computing. You can calculate the entanglement of a quantum circuit of two qubits using the concurrence equation 1.

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

$$C = 2|ad - bc| \tag{1}$$

For example, the Bell circuit is a circuit that generates entanglement states (C=1). You can generalize this concurrence for more extensive systems. In this project, we calculated the entanglement given a circuit with the equation 1 for all the qubits pairs. Figures 1 and 2 show two different circuits and the entanglement matrix representation. In this work, we will work with figure 2 Quantum Full Adder (QFA) for hardware optimization.

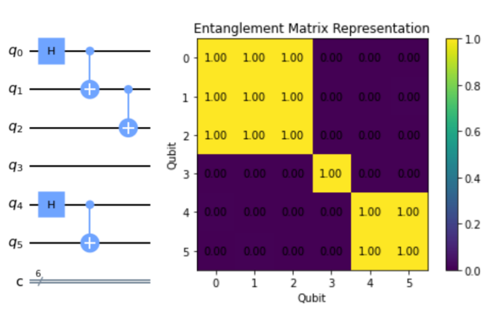

Fig. 1. Quantum circuit and entanglement matrix from a GHZ and Bell circuit to show the entanglement properties.

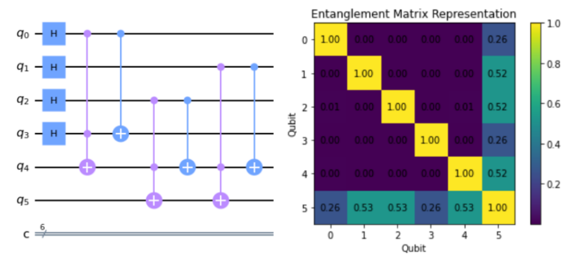

Fig. 2. Quantum Full Adder (QFA) for two qubits circuit and entanglement matrix. The qubit 5 is more entangled with the qubits compare to the others.

In figure 2 can be seen that the qubit that has more entanglement with the others is the fifth qubit, the qubit one, two, and four are more significant compared with qubits zero and three. Using the *Qiskit* framework with the noise library, we can add noise to the quantum circuit and compare the theoretical results of the quantum full adder of two qubits from figure 2. Figure 3 shows the theoretical result from the Quantum Full Adder (QFA) for two qubits with 100000 shots where just the last three qubits are measured, which refers to the C_0, C_1, and C_2.

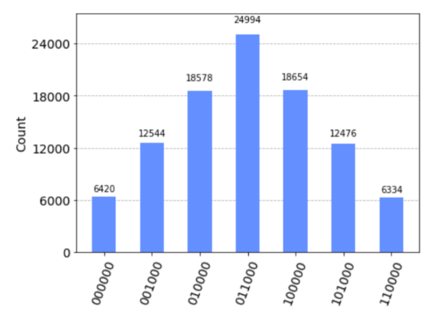

Fig. 3. Probability distribution for the superposition of the QFA for two qubits.

We will compare the error when the noise and architecture constraints are involved and how this is mitigated considering the connectivity of the qubits based on the entanglement matrix. The two architectures considered for this circuit and the architecture based on the entanglement matrix are shown in Figure 10.

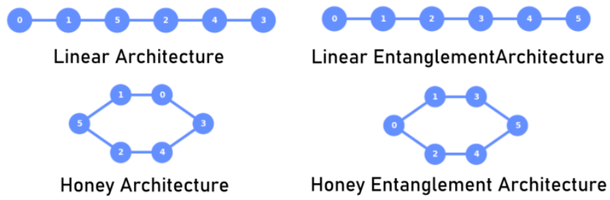

Fig. 4. Linear architecture and honey architecture to evaluate (left). Modified linear architecture and honey architecture based on entanglement (right).

The result compared with the theoretical results from Figure 3 are shown in Table I. All simulations are executed with 10000 shots.

Architecture	**Error sum(prob_i-prob_j)**
Full connectivity with noise	2.37 %
Linear connectivity with noise	9.93 %
Honey connectivity with noise	7.72 %
Linear connectivity with noise and entanglement awareness	4.80 %
Honey connectivity with noise and entanglement awareness	2.87 %

The result from Table I shows that if you modify the connectivity of your circuit based on the entanglement of the quantum circuit, the circuit's performance will increase. Also, there is better architecture for the qubit design. In this work, we will work under the honey architecture over the linear or other implementations.

II. Honey Architecture Physical Design Using Qiskit Metal

A good architecture for superconducting circuit design in quantum computation is the honey architecture. This architecture can offer high connectivity compared to other 2D designs. In this section, the main idea is to design the honey architecture and execute the physical implementation to hardware that builds the quantum chip using Qiskit Metal.

We will focus on the transmon 4-ports design. The main idea is to build a lattice, such as adapting the port design into the honey architecture. This is shown in the figure 5. The code is generalized to any number of qubits as is shown in figure 6.

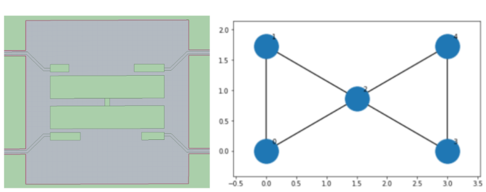

Fig. 5. Transmon like qubit and graph representation.

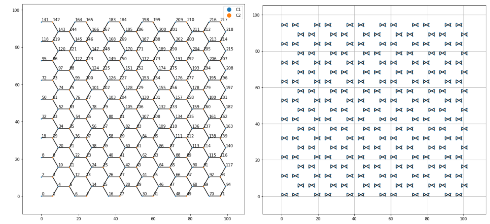

Fig. 6. Lattice implementation and transmon adaption.

Via code, all the information from the lattice parameters and the nearest transmon and ports are passed to Qiskit Metal to build the entire design as is shown in figure 7, 8, 9 for the case of 11 transmons.

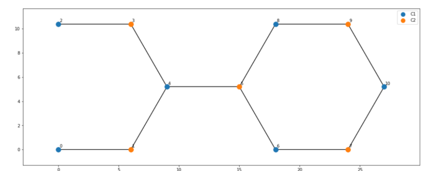

Fig. 7. Lattice design for the connection from qubit to qubit.

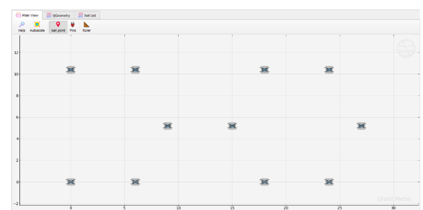

Fig. 8. Implementation in Qiskit Metal.

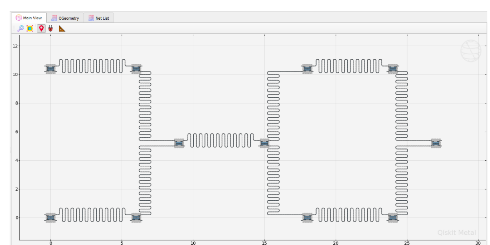

Fig. 9. Connectivity implementation in Qiskit Metal.

After all this implementations this can be design using $ANSYS$ for a real implementation as is shown in figure 12.

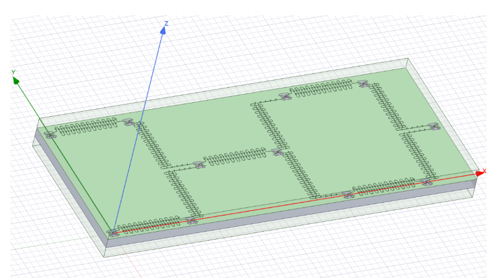

Fig. 10. Implementation in ANSYS.

III. Future Work

There are improvements for both problems. For the entanglement matrix and the QFA, there will be valuable to compare the other connectivities (720 configurations) to compare those to an architecture based on entanglement.

For the second problem, the design is still not practical due to the lack of control lines and measure lines. Also, the transmon-like qubit will be more efficient if they are tunable, which will be the next problem to tackle.