Quantum Full Adder Circuit

1

1. Problem Statement

We need to design a quantum full adder circuit that takes three inputs, a, b, and cin, represented as combinations of quantum states, and produces two outputs, sum and cout, also represented as combinations of quantum states.

2. Circuit Diagram

In this case, we have to implement two oracles. The first is the sum oracle, and the second is the carry oracle. We define our sum oracle by applying CNOT gates to each set of three input qubits, with the control qubits being the inputs and the sum qubit as the target. Similarly, we define the carry oracle by applying CCNOT gates to all input qubits, with any two qubits acting as control and the carry-out qubit as the target. Below is the circuit diagram provided.

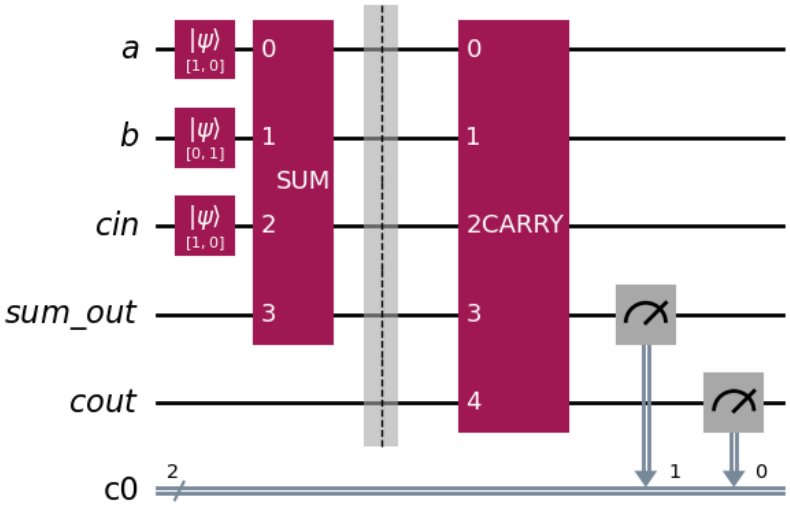


Figura 1. Circuit for Quantum Full Adder

3. Methods

"We have defined two oracles in this circuit: one is the sum oracle, and the other is the carry oracle. The sum oracle flips the state of the sum qubit if any one of the a, b, or c_{in} states is initialized as $|1\rangle$. The carry oracle flips the state of the c_{out} qubit if two states from a, b, and c_{in} are initialized as $|1\rangle$.

We simulated this circuit using the 'qasm-simulator' and obtained the results shown in the next section.

Similarly, we also ran this on an actual quantum computer backend, 'ibm-brisbane', and obtained the results.

4. Results and Discussions

4.1. Simulation Results

Below are the simulation results illustrated in the figure. The first statevector represents the sum, while the second statevector represents the carry bit. We simulated this circuit for '1024' shots and obtained the results.

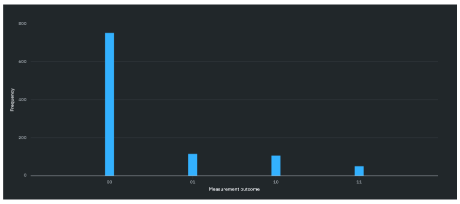
```
Initial states: a=0, b=0, cin=0
{'00': 1024}

Initial states: a=0, b=0, cin=1
{'10': 1024}

Initial states: a=0, b=1, cin=0
{'10': 1024}

Initial states: a=0, b=1, cin=1
{'01': 1024}

Initial states: a=1, b=0, cin=0
{'10': 1024}

Initial states: a=1, b=0, cin=1
{'01': 1024}

Initial states: a=1, b=1, cin=0
{'01': 1024}

Initial states: a=1, b=1, cin=1
{'11': 1024}
```

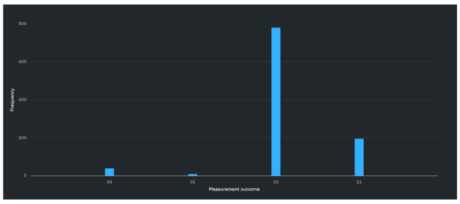
Figura 2. Simulation Results of Full Adder

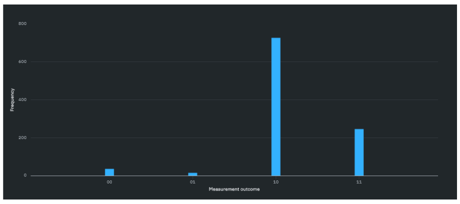
4.2. Quantum Hardware Results

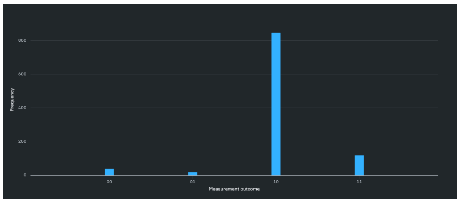
The results given below were obtained by running the program on actual quantum hardware. We obtained outcomes in the form of a probability distribution, represented as ab, where 'a' represents the sum bit and 'b' represents the carry bit. It can be observed that there are additional amplitudes associated with other combinations, and the amplitude of one combination is notably greater. This variation is attributed to the probabilistic nature of quantum computing.

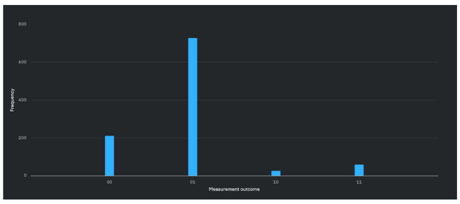
As seen from these results, the nature of results are probabilistic so as the size of the input increases, the number of operations required for computation also increases, potentially leading to exponential growth in computational complexity. Additionally, the probabilistic nature of quantum computing introduces variations in outcomes, influencing the reliability and accuracy of results. Resource-wise, the implementation demands a significant quantum hardware infrastructure capable of handling the computational workload and managing the inherent noise and errors associated with quantum operations. Consequently, scaling the adder algorithmically necessitates a balance between computational efficiency and resource availability, emphasizing the need for advancements in both algorithm design and quantum hardware development to achieve scalable and reliable quantum adder implementations.

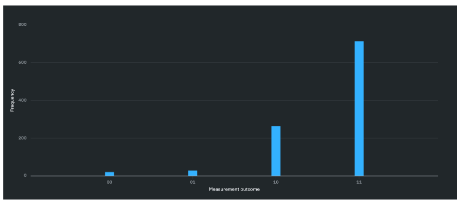

800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome


800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome


800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome


800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome


800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome


800
600
400
200
0
Frequency
00
01
10
11
Measurement outcome

5. Conclusion

In conclusion, this full adder project showcases the practical implementation of quantum computing principles to perform arithmetic operations. Through the utilization of actual quantum hardware, we obtained results in the form of a probability distribution, demonstrating the potential of quantum computers in handling complex computations. However, challenges such as algorithmic scalability and resource requirements highlight areas for further research and development.