Quantum Computing based Implementation of Full Adder

Mohammed Arifuddin Sohel
Professor ECED
Muffakham Jah college of engineering and technology
Hyderabad
arif.sohel@mjcollege.ac.in

Naailah Zia
Electrical and Computer Department
University of Illinois
Chicago,USA
naailahzia.nz@gmail.com

Mohd Akef Ali
ECED
Muffakham Jah College of Engineering and Technology
Hyderabad
akefali25@gmail.com

Nida Zia
ECED
Muffakham Jah College of Engineering and Technology
Hyderabad
nidazia04@gmail.com

***Abstract*—Quantum computing is the most advanced form of computing that is evolving today. Classical computers work on binary bits - 0 or 1, but quantum computers work on quantum bits or qubits which can be in the state 0 and 1 simultaneously and exhibit properties like superposition and entanglement. This paper proposes the implementation of a full adder with the concepts of quantum computing using quantum gates on IBM composer. IBM composer is the choice of tool for this paper, and the resultant speed of operation is calculated to be 6.064ms for 1024 shots (1 shot is 100 repetitions). By designing a full adder in this paper, it is contemplated that more complicated digital circuitry will be implemented on quantum computers in near future, thus paving the way for bringing high speed quantum computing in our daily lives to solve critical problems that are unresolvable by conventional computing.**

Keywords— IBM composer, Quantum Computer, Qubit, Speed of Operation, Superposition.

I. INTRODUCTION

With humongous research and increasing speculations of Quantum computers replacing conventional computers, Quantum computing has taken a huge leap over the years. As cited in [1] Quantum computing and AI have already met in machine learning. Which means there are innumerable applications and future scope in this field. The research [2] talks about the freedom of qubits due to superposition which can be linked due to entanglement. As a consequence of which with enough qubits, a billion-year operation on classical computers can take days or hours on a quantum device. The research [3] mentions that companies like IBM, Microsoft, Google are hard at work to take this theory into practice which inspires us to believe that we are nearing quantum supremacy. The main purpose of this paper is to simulate full adder using a 5-bit quantum simulation interface provided by IBM. For readers who are not acquainted with quantum computing, a brief introduction is given. Along with that, a guide on IBM composer is given. It is hoped that this paper will serve as a pre-requisite for researchers who are looking forward to go deeper and design real time computers using quantum computing. At the end of this paper, a proposed circuit of a full adder using quantum gates is shown. This circuit is optimised to reduce garbage values.

II. BACKGROUND

There are numerous papers on full adders using quantum computing. The most relevant paper to this paper is [5] which uses Islam gate and generates some garbage values. The paper [6] does the same using a proposed khan gate and still generates some garbage values. In this paper the proposed implementation of full adder circuit using quantum gates helps generate the classical full adder circuit truth table, without obtaining any garbage values which makes the following simulation much more efficient. Section I gives a brief introduction about quantum computing, section II gives us an idea of an existing research about full adder and how the research done now optimizes it. Section III is a brief introduction about IBM composer and the gates used in it. Section IV explains how the program implements a full adder. Section V exhibits the circuit generated and the results obtained.

III. IBM COMPOSER

For the purpose of this research, we chose to use IBM's Quantum experience. IBM Q experience is an online simulator made available for research enthusiasts in an intuitive interface with guides to understand it and get started. In IBM composer (as is standard), the measurements are performed in the computational basis. After it is measured, we know a qubit loses its quantum properties and becomes a classical bit. The measurement either takes the value 0 if the qubit is in state $|0\rangle$(pronounced as KET 0) , and value 1 if the qubit is in state $|1\rangle$(pronounced as KET 1). Each qubit will have a corresponding quantum register to store quantum information. Since measuring a qubit results in a classical bit, for each qubit there is also a classical register in which the classical information is stored. So a 5 qubit processor has 5 quantum registers and 5 classical registers. We also know that the probability that a superposed qubit will fall to 0 or 1 depends on the modulus square of their respective coefficients. The IBM composer, by default runs an experiment 100 times, each run of experiment is called a shot. If we were to measure a superimposed qubit, the results will show how many times out of 100 the qubit fell to 0 and how many times it fell to 1. The number of shots is also customizable. In this simulation the number of shots is taken as 1024 to increase the accuracy. For ease of understanding and readability, the results are always represented in form of histograms. For the purpose of this research we used the 5-qubit processor on IBM Q experience. This processor is housed in the cryoperm shield of the IBM quantum computer .It is from this initial $|0>$, that we begin manipulating the qubits, thus giving us control over their generally random behaviour. All operations and behaviour of qubits and quantum computers from here on will be with regards to the construction and principles of IBM quantum computer and processor. An IBM composer has a number of gates like X gate, Hadamard gate, CNOT gate and Toffoli gate. Each gate serves its own purpose in the composer. The conditional gate used is

978-1-7281-9744-9/20/$31.00 ©2020 IEEE

Controlled-NOT. The CNOT gate's action on classical basis states is to flip (apply a NOT or X gate to) the target qubit only if the control qubit is |1>; otherwise it does nothing. A Toffoli gate has two inputs and one output, and outputs 1 if and only if both inputs are 1. This reversible AND gate is called the Toffoli gate, shown in fig2.

TOF|a,b,c⟩=|a,b,(a AND b) XOR c⟩

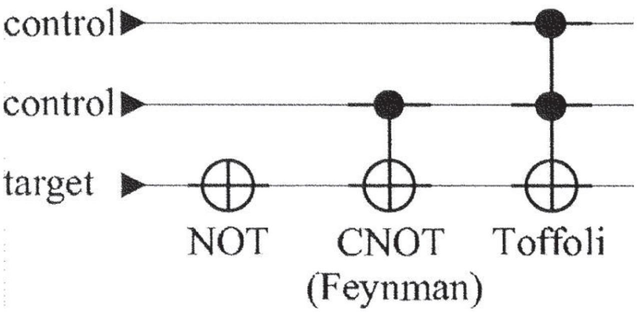


Fig. 1. Representation of Toffoli gate

IV. IMPLEMENTATION OF FULL ADDER

OPENQASM CODE:

```
include "qelib1.inc";
qreg q[5];
creg c[5];
h q[0];
h q[1];
h q[2];
ccx q[1],q[2],q[3];
cx q[2],q[4];
cx q[1],q[4];
cx q[0],q[4];
cx q[1],q[2];
ccx q[0],q[2],q[3];
cx q[1],q[2];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];
measure q[3] -> c[3];
measure q[4] -> c[4];
```

1. The code is begun by including the standard library and declaring the number of quantum and classical registers required. The number of inputs and outputs = number of qubits required. Therefore only 5 out of available 5 qubits are required.
2. The truth table can then be written as shown in the Table I. The second row in the table shows which qubit behaves as input or output.

TABLE I. FULL ADDER TRUTH TABLE FOR FIVE QUBITS

OUTPUTS		INPUTS		
q[4]	q[3]	q[2]	q[1]	q[0]
SUM	COUT	B	A	CIN
0	0	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	1	1	0
1	0	0	0	1
0	1	1	0	1
0	1	0	1	1
1	1	1	1	1

3. After declaring qreg and creg, then next step in writing the code is to create 8 input combinations, that can be done by taking 3 qubits, q[2],q[1],q[0] in superposition. Note: here that all qubits that have not been superimposed (q[3] and q[4]) are initially in |0> state. It is finally time to manipulate the actual outputs Sum (q[3]) and Cout (q[4]). This is done by using Toffoli and CNOT gates. 'ccx q[1],q[2],q[3];' means that when both q[1] and q[2] are |1> flip q[3].
4. When it is written 'cx q[2],q[4];', it means that when q[2] is |1>, flip q[4].
5. When it is written 'cx q[1],q[4];', it means that when q[1] is |1>, flip q[4].

The truth table is represented as shown in the table II.

TABLE II. FULL ADDER TRUTH TABLE FOR FIVE QUBITS : MANIPULATING O/P OF PT. 4,5

OUTPUTS		INPUTS		
q[4]	q[3]	q[2]	q[1]	q[0]
SUM	COUT	B	A	CIN
0	0	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	1	1	0
0	0	0	0	1
1	0	1	0	1
1	0	0	1	1
0	1	1	1	1

6. When it is written 'cx q[0],q[4];', it means that when q[0] is |1>, flip q[4].
7. When it is written 'cx q[1],q[2];', it means that when q[1] is |1>, flip q[2].

The resultant truth table is shown in table III.

TABLE III. FULL ADDER TRUTH TABLE FOR FIVE QUBITS : MANIPULATING O/P OF PT. 6,7

OUTPUTS		INPUTS		
q[4]	q[3]	q[2]	q[1]	q[0]
SUM	COUT	B	A	CIN
0	0	0	0	0
1	0	1	0	0
1	0	1	1	0
0	1	0	1	0
1	0	0	0	1
0	0	1	0	1
0	0	1	1	1
1	1	0	1	1

8. When it is written ‘ccx q[0], q[2],q[3];’, it means that when both q[0] and q[2] are |1> flip q[3].
9. The proper output is obtained by using CNOT. When it is written ‘cx q[1],q[2];’, it means that when q[1] is |1>, flip q[2]. The truth table becomes as shown in table IV

TABLE IV. FULL ADDER TRUTH TABLE FOR FIVE QUBITS : MANIPULATING O/P OF PT. 8,9

OUTPUTS		INPUTS		
q[4]	q[3]	q[2]	q[1]	q[0]
SUM	COUT	B	A	CIN
0	0	0	0	0
1	0	1	0	0
1	0	0	1	0
0	1	1	1	0
1	0	0	0	1
0	1	1	0	1
0	1	0	1	1
1	1	1	1	1

10. All the qubits are measured to obtain the results in simulation.

V. RESULT

As seen above, in order to obtain the final desired full adder truth table, multiple intermediate steps are involved which utilize the qubits to store the intermediary values. These values are of no relevance at the end. Therefore, it is important that we reset the values of these qubits using a CNOT gate (when q [1] is |1>, flip q [2]). By doing so using an IBM composer we eliminate the garbage values.

To check the result i.e. the histograms, the code is written in the IBM editor and sent to their central server. After the code is simulated and the result is generated, the histograms are displayed. We will know that the code written and the circuit generated is working, by comparing the result with the truth table in Table IV. The following Fig 2 is the circuit generated after writing the code. As goes with the Full Adder.

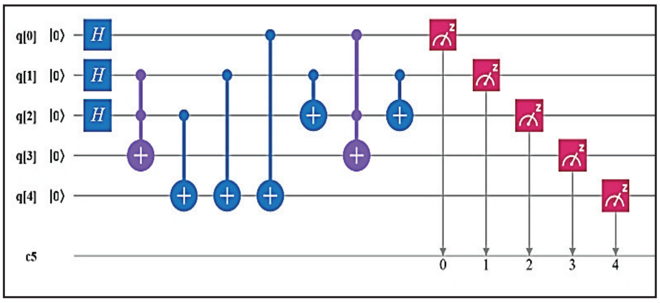
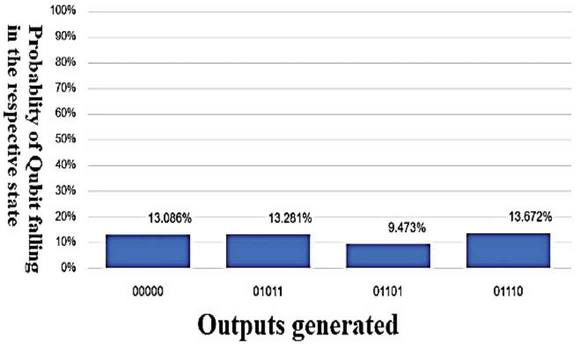
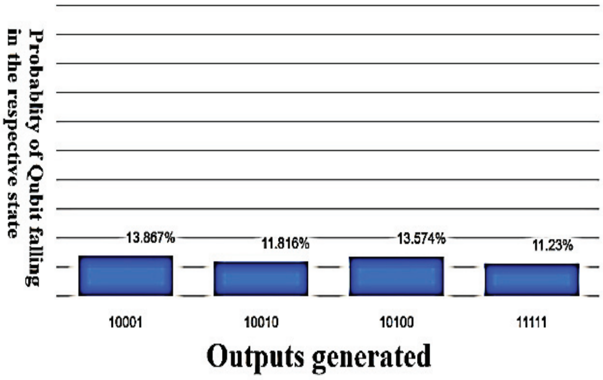


Fig. 2. Proposed Circuit for full adder

The histogram result is given below. It is to be read in the order specified in Table IV. We can observe that the results obtained match the table precisely. The outputs of the histogram (the bottom value) are to be read from right to left with each bar depicting A, B, CIN, COUT, SUM. The values on top however, show the probability of the qubit to fall in that state out of 100%.



Histogram. 1.



Histogram. 1. Continued

Fig. 3. Histogram result

Also, the time taken to simulate and obtain the is 6.064ms, which is a feat in the area of quantum computing. The number of shots can be manipulated as per the requirement and accuracy. In our case the number of shots is taken as 1024. In the histogram we can also see the relative probability of each output in %. Thus, the simulation results, the code and the circuit have been verified.

VI. CONCLUSION

In this paper, the proposed circuit of a Full Adder is introduced. In comparision with the circuit in [4]Quantum Realization Full Adder-Subtractor Circuit Design Using Islam gate and [5]Synthesis of Quantum Circuit for full adder using khan Gate, the garbage outputs have been eliminated. A comparison of the number of garbage values obtained is shown in TABLE V proving that the circuit now simulated works efficiently and briskly using the IBM interface.

TABLE V. COMPARISON WITH CITED LITERATURE

FULL ADDER	NO. OF GARBAGE LINES
USING ISLAM GATE	8
USING KHAN GATE	5
PROPOSED CIRCUIT	0

VII. FUTURE SCOPE

This project is more about understanding quantum computing and leads into the implementation of full adder. There is a lot of scope to further the main aim of the project with other combinational and sequential logic circuits. This would however require a more extensive study into other more elaborate tools than IBM Q experience and even other programming languages. As the complexity of computation increases, so will the method to make them more efficient. Advancement in this field can solve unresolved problems like synthesis of a drug for a deadly disease or space travel. These computers can also serve as a huge feat in AI and ML fields.

REFERENCES

[1] Mingsheng Ying , "Quantum computation, quantum theory and AI", Volume 174, Issue 2.

[2] Joseph Fernandez , "Classical computing vs quantum computing", July 19, 2018.

[3] Jacob Ridley,"Quantum computing :the key players leading us into the quantum age".

[4] Madhumita Mazumder and Indranil Guha Roy, "Quantum Realization Full Adder-Subtractor Circuit Design Using Islam gate", International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 5 Issue 6, June 2017

[5] Madhumita Mazumder, "Synthesis of Quantum Circuit for FULL ADDER Using KHAN Gate", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 6, Issue 6, June 2017.

[6] James Taylor Lewis – Quantum Computation: An Introduction Thesis, April 5, 1999