International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT23903106

Implementation of Combinational and Sequential Logic Circuits using Quantum Computing

Neha M Harapanhalli[1], Kavipriya M[1], Ishwari Jigajinni[1], Dr. Keerti Kulkarni[2]
[1]ECE Department,BNM Institute of Technology, Bangalore, India
[2]Associate Professor, ECE Department, BNM Institute of Technology, Bangalore, India

ARTICLEINFO

Article History:

Accepted: 01 June 2023
Published: 10 June 2023

Publication Issue
Volume 9, Issue 3
May-June-2023

Page Number
430-439

ABSTRACT

Quantum computing is an area of computer science that uses the principles of quantum theory. Quantum computing is a booming field and could lead significant advances in various fields, from chemistry to material science to nuclear physics and machine learning. Quantum computing uses quantum bits as a fundamental building block of computations. The power of quantum bits is their ability to scale exponentially, which makes complex computations faster and efficient. The objective of this paper is to design Flip Flops and combinational logic circuits using quantum computing. Flip Flops and combinational circuits assist in complex computations with reduced delay, and high efficiency when designed using quantum computing. The flip flops and adders can henceforth be used in several applications.

Keywords—Quantum computing, Qubits, Speed, Gate count.

I. INTRODUCTION

Quantum Computing is an area of study focused on the development of computer-based technologies centred around the principles of quantum theory. In 1999, first quantum computer was developed out of superconductors by D-Wave Systems a Canadian organization. Richard Feynman was the first to foresee the need for quantum computers to accurately recreate quantum physics. Calculations based on quantum mechanics are performed using a quantum computer. Atoms, photons, ions, or electrons, along with their accompanying control mechanisms, can physically materialize quantum computers at the atomic level, acting as the computer's processor and memory. Quantum computers with the ability to process and store several states concurrently have the potential to be millions of times more powerful than even the most powerful supercomputers in use today. Additionally, compared to their conventional counterparts, quantum computers promise safe transmission, extremely fast speed, and the capacity to store vast amounts of data. Several scholars are turning to this topic because of its potential to change the modern computing industry. The limitations of traditional CMOS technology for high density and high-performance applications may be solved by quantum computing. The binary bits 0 and 1 used in computing today are decoded by computational processors and are then processed by logic gates based on switching transistors.

Quantum computing (QC) is believed to change the future of computers depending on the underlying circuitry, from medicine to computer science. Due to the large number of activities and research in the quantum field, there is a need for memory organization which will play a major role in the above activities. Memory layout is not dependent on anything other than sequential circuits. Basic sequential circuits such as Flip - Flops and registers play a major role in memory processing. It is easy to set up a sequential circuit on a classical computer, but it is the opposite of quantum computing as the reference is to so-called "qubits" rather than a bit. Due to the lack of quantum hardware, Quantum simulation (QS) is still the most common way to use quantum circuits. This paper uses IBM hardware simulation tools and Qiskit tools for simulation. The paper demonstrates the working of Flip Flops on actual quantum system over the classical flip flops. Our main objective is to utilize the booming technology of Quantum computing to simulate the D Flip Flop, JK Flip Flop and SR Flip Flop and 8 bit Ripple Carry Adder on the Qiskit simulation tool and later implement the same on the IBM open-source hardware simulation tool. Upon simulation on both these simulators, we will be able to analyse and exhibit that Quantum based simulations offer lesser delays, reduced runtime, lesser gate counts etc. There can be errors when the simulation is implemented on real time hardware. These errors can henceforth be corrected and brought to a stage where a desirable output is generated.

II. REVERSIBLE GATES

Reversible gates are those that do not lose information and are reversible in nature. The inputs and outputs of a reversible logic gate can be retrieved from each other. In classical gates, the input information cannot be traced back. It loses the information during computation in the form of heat dissipation. The reversible logic operations can't erase information and dissipate zero heat. The circuit operates in a backward operation, allows reproducing the inputs from the outputs and consumes zero power. Of the many optimizations in reversible logic gates, optimization in delay, quantum cost and garbage outputs are some important parameters. Each reversible gate has a cost associated with it called the quantum cost.

A. FEYMAN GATE

Feynman gate (FG) or Controlled-NOT gate (CNOT) is a 2 inputs 2 outputs reversible gate having the mapping (A, B) to (P=A, Q=A ⊕ B) where A, B are the inputs and P, Q are the outputs, respectively. Since it is a 2x2 gate, it has a quantum cost of 1. The figure below shows the circuit and quantum representation of a CNOT gate.

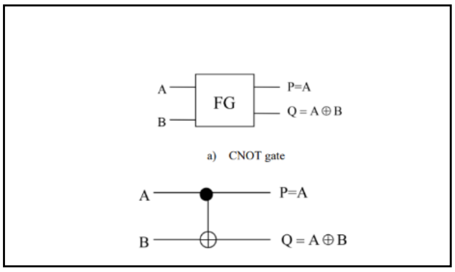


Fig 1: CNOT gate and its Quantum representation

Input (q1 q0)	Output (q1 q0)
00	00
01	11
10	11
11	01

Table 1: Truth table of CNOT gate

B. TOFFOLI GATE

Toffoli Gate (TG) is a 3x3 two-through reversible gate as shown in Fig. 2(a). Two-through means two of its outputs are the same as inputs with the mapping (A, B, C) to (P=A, Q=B, R=A ·B⊕ C), where A, B, C are inputs and P, Q, R are outputs, respectively. Toffoli gate is one of the most popular reversible gates and has a quantum cost of 5 as shown below. The quantum cost of Toffoli gate is 5 as it needs 2 Controlled-V gates, 1 Controlled-V + gate and 2 CNOT gates to implement it.

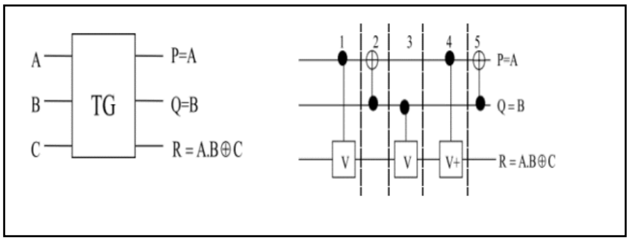


Fig 2: Toffoli Gate and its quantum representation

INPUT			OUTPUT		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 2 : Truth table of Toffoli gate

C. FREDKIN GATE

Fredkin gate is a (3x3) conservative reversible gate, having the mapping (A, B, C) to (P=A, Q=A'B+AC, R=AB+A'C), where A, B, C are the inputs and P, Q, R are the outputs, respectively. It is called a 3x3 gate because it has three inputs and three outputs. The figure below shows the Fredkin gate and its quantum implementation with a quantum cost of 5. Each dotted rectangle in the second image is equivalent to a 2x2 Feynman gate and so the quantum cost of each dotted rectangles is 1. Hence Fredkin gate cost consists of 2 dotted rectangles, 1 Controlled-V gate and 2 CNOT gates resulting in its quantum cost as 5.

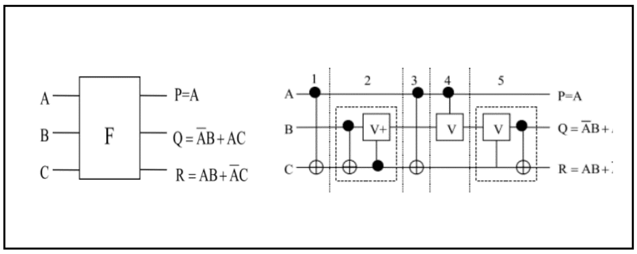


Fig 3 : Fredkin Gate and its quantum representation

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Table 3: Truth table of Fredkin Gate

III. QUANTUM BASICS

A quantum gate (or quantum logic gate) is a fundamental quantum circuit that uses a few qubits in quantum computing, particularly the quantum circuit model of computation. Similar to how classical logic gates are the building blocks of traditional digital circuits, they are the building blocks of quantum circuits. Before discussing quantum gates, let's first grasp a few fundamental ideas.

A.QUANTUM BITS

A digital computer uses bits, which can be either 0 or 1, to store and process information. Physically, anything that has two unique configurations one denoted by "0" and the other by "1" can be referred to as a bit. In contemporary computing, bits are represented by the presence or absence of an electrical signal, which respectively encodes "0" or "1". Any bit created from a quantum system, such as an electron or photon, is known as a quantum bit. A quantum bit must have two different states, one representing "0," and the other representing "1," much like classical bits.

A quantum bit, in contrast to a conventional bit, is also capable of superposition states, incompatible measurements, and even entanglement with other quantum bits. They have the features of superposition, interference and entanglement which make them fundamentally different and much more powerful than classical bits.

B. BITS V/S QUBITS

A single qubit can be represented by a linear combination of f |0> and |1> as:

$$\Psi = \alpha\, |0> + \beta\, |1> \quad (1)$$

Where, α and β are probability amplitudes and can in general both be complex numbers with

$$\alpha 2 + \beta 2 = 1 \quad (2)$$

The special states 0 and 1 are known as computational basis states, and form an orthonormal basis for this vector space, being,

$$|0> = (0, 1) \text{ and } |1> = (1, 0). \quad (3)$$

Bloch sphere representation of qubit:

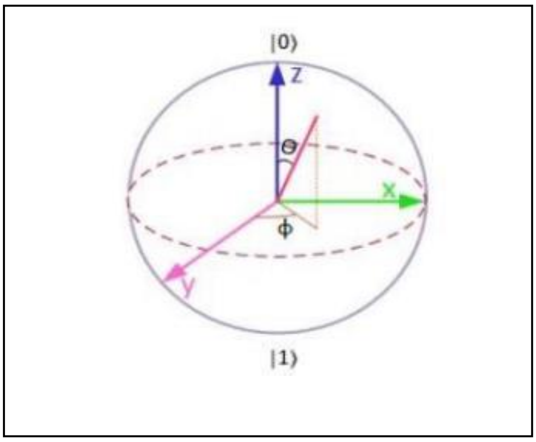


Fig 1.3 Bloch sphere representation of qubit

C. FEATURES OF QUANTUM COMPUTING

SUPERPOSITION

Superposition is the ability of a quantum system to exist in multiple states simultaneously. The possibility of parallel computation is given real significance by this powerful property of quantum computers. A qubit can concurrently represent two states. Thus, n qubits can represent n states simultaneously. As a result, by establishing a superposition of 2n states as an input and processing it in one step, the problem that a conventional computer requires to solve in 2n steps can be solved. Quantum computers have an advantage over conventional computers, which can only do one thing at a time, in terms of power, efficiency, and speed due to their capacity to exist in several states simultaneously. Since quantum computers can conduct multiple calculations at once, they can answer problems far more quickly than conventional machines.

ENTANGLEMENT

When two systems are in the quantum entangled state, no matter how far apart they are, learning something about one system immediately reveals something about the other. In quantum computers, if the state of an entangled qubit is changed, the associated qubit's state will also be changed instantly. Entanglement therefore increases the processing efficiency of quantum computers. Since processing one qubit reveals information about numerous qubits, doubling the number of qubits will not necessarily result in doubling the number of processes.

INTERFERENCE

Quantum states can be manipulated using interference, which can also be used to amplify signals pointing in the proper direction while cancelling those pointing in the wrong direction. There are two categories of quantum interference: constructive interference and destructive interference. A wave that is twice as tall is created when two in-phase waves, or waves that peak at the same time, interact positively. On the other hand, out-of-phase waves have opposing peaks and interfere destructively, creating an entirely flat wave. In quantum computing, interference is utilised to change probability amplitudes. In other words, there is a probability that every scenario will play out. There are just two possible results when one qubit is utilised for computation: 0 and 1. When two qubits are employed, the number of potential outcomes doubles, and the potential outcomes are 00, 01, 10, and 11. The number of outcomes doubles to 000, 001, 010, 011, 100, 101, 110, and 111 when three qubits are employed. One of the characteristics that helps quantum computers execute calculations that were previously thought to be beyond the capabilities of even the most powerful supercomputers is this exponential growth.

QUANTUM LOGIC GATES

Quantum logic gates are fundamental building blocks of quantum circuits, the equivalent of classical logic gates in classical digital circuits. Quantum logic gates come in various types, each designed to perform specific quantum operations on qubits. Quantum gates operate on qubits(quantum bits). The key properties of quantum computing are superposition, interference, and entanglement. Quantum logic gates, in contrast to many classical logic gates, are reversible. Unitary matrices are used to depict quantum logic gates. Similar to how popular classical logic gates function on one or two bits, the most common quantum gates operate on spaces of one or two qubits. As a result, 2x2 or 4x4 unitary matrices can be used to identify quantum gates as matrices.

PAULI’S X GATE

The Pauli-X gate acts on a single qubit. It is the quantum equivalent of a NOT gate (with respect to the standard basis x |0⟩, |1⟩ which privileges the Z-direction). It equates to a rotation of the Bloch Sphere around the X-axis by π radians. It maps |0⟩ to |1⟩ and |1⟩ to |0⟩. Due to this nature, it is sometimes called bit-flip. It is represented by the Pauli matrix.

$$x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

TOFFOLI GATE

The Toffoli gate, also called CCNOT gate, is a 3-bit gate, which is universal for classical computation. If the first two bits are in the state |1⟩, it applies a Pauli-X on the third bit, else it does nothing. It is an example of a controlled gate. The matrix representation of a Toffoli gate is given by:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

HADAMARD GATE

The Hadamard gate creates a superposition of two states, which is a key component in quantum algorithms like Shor's algorithm for factoring large numbers. The Hadamard gate acts on a single qubit. It maps the basis state |0⟩ to and |1⟩ to and represents a rotation of about the axis π. Equivalently, it is the combination of two rotations, π/2 about the Y-axis followed by π about the X-axis. It is represented by the Hadamard matrix: (11) Since HH* =I where I is the identity matrix, H is indeed a unitary matrix. (i.e.) In mathematics, a complex square matrix U is unitary if its conjugate transpose U* is also its inverse – that is,

$$U^*U = UU^* = I$$

IV. METHODOLOGY

The project aims to implement various flip-flops and adder using quantum computing.

Sequential logic circuits called flip-flops are used to store and process data. A 1-bit memory component having SET and RESET inputs is called an SR flip-flop. The device is reset to output 0 by the RESET input and is set to output 1 by the SET input. The D flip-flop is a clocked flip-flop that keeps its input in the same state. In order to avoid invalid output scenarios, the JK flip-flop is a gated SR flip-flop with an additional clock input. There are four different input combinations that might be used with it: "logic 1," "logic 0," "no change," and "toggle."

Eight complete adders are joined in cascade to form the 8-bit ripple carry adder. The carry output from each full adder acts as the input carry for the full adder's following step. Two 8-bit binary sequences are added with the help of this adder.

The Qiskit library, a simulation tool for quantum computation (IBM Quantum Experience), is used to develop and simulate the circuits. Reversible quantum gates from the library, including the Toffoli, Fredkin, Pauli X, and Hadamard gates, are used to optimise and verify the operation of the simulated circuits. These gates aid in increasing the effectiveness of quantum circuits.

PARAMETERS

Parametric analysis of circuits involves analyzing the behavior of an electronic circuit as the values of its parameters are varied. Parameters such as speed, memory, gate count, precision, can have a significant impact on the behavior of a circuit, and by understanding how changes in these parameters affect the circuit's performance, designers can optimize circuit performance and ensure that the circuit operates within its desired range. In this project we insist on the significance of speed and gate count as two parameters which makes quantum computing better than classical computing.

SPEED

Speed is a key parameter that distinguishes classical and quantum computers. Quantum computers have the potential to be significantly faster than classical computers for certain types of problems. In classical computing, the speed of a computer is largely determined by its clock speed, which is measured in gigahertz (GHz) and represents the number of clock cycles per second. Classical computers perform computations using classical bits, which can have a value of either 0 or 1 and perform operations on these bits using classical logic gates. In contrast, quantum computers use qubits, which can exist in a superposition of both 0 and 1 states simultaneously. This allows quantum computers to perform certain computations using quantum parallelism, which can potentially provide exponential speedup for certain types of problems. Overall, speed is a key parameter that distinguishes classical and quantum computers, and quantum computers have the potential to provide significant speedup for certain types of problems. However, the exact speedup will depend on the specific problem being solved and the architecture and performance of the quantum computer.

GATE COUNT

Gate count is a parameter used to evaluate the complexity and efficiency of a digital circuit. It refers to the total number of logic gates used in the circuit, where each gate performs a specific Boolean operation on input signals and produces an output signal. Gate count is an important parameter for evaluating the performance and efficiency of a digital circuit, as circuits with fewer gates tend to be more efficient and easier to implement. Reducing the gate count can also help to minimize power consumption, improve reliability, and reduce the overall cost of the circuit. In quantum computing, gate count is an important parameter used to evaluate the complexity and efficiency of quantum circuits. In quantum circuits, quantum gates are used to perform operations on qubits, which are the fundamental building blocks of quantum computation. Quantum gates can be used to implement a variety of quantum operations, such as the Hadamard gate, the Pauli gates, the phase gate, the CNOT gate, and others. These gates can be combined in various ways to create more complex circuits and perform different types of quantum computations. In addition, the gate count of a quantum circuit can be highly dependent on the specific quantum algorithm being implemented and the architecture of the quantum computer.

Since quantum gates can be quite complex and difficult to implement, reducing the gate count in a quantum circuit is often a major goal of quantum algorithm designers. Reducing the gate count can lead to faster computations and can help mitigate the effects of noise and decoherence in the quantum hardware. Speed, on the other hand, refers to the rate at which quantum computations can be performed in a quantum computer. In quantum computing, speed is often measured in terms of the number of qubits and the depth of the circuit, which is the number of layers of gates in the circuit. Quantum computers with more qubits and deeper circuits can perform more complex computations and can potentially offer faster quantum speedup for certain types of problems. Speed, gate count in conjunction with other parameters are used to evaluate the performance of a quantum system to get a better picture of performance in quantum systems.

V. IMPLEMENTATIONS

QUANTUM COMPUTATION

QISKIT: Qiskit is a free and open-source quantum software development kit (SDK). IBM developed a computing platform that includes a set of programming tools as well as quantum algorithm simulation. It is intended for both scholars and students as well as developers interested in quantum computing. Qiskit includes several quantum simulators, a set of quantum algorithms, and cloud access to IBM's quantum hardware are among the resources available. It is written in Python and has several tools for quantum circuit design, simulation, and optimisation.

The Qiskit SDK has a set of high-level APIs that allow developers to build quantum circuits using a familiar programming paradigm.

It also includes a set of low-level APIs that allow developers to have fine-grained control over the quantum hardware, allowing them to optimize circuit performance and reduce error rates. In addition, the SDK offers tools for quantum error correction, quantum chemistry simulations, and quantum physics simulations machine learning at the quantum level

QUANTUM COMPUTER: Lima and Quito, two South American cities, have emerged as centres for quantum computing research. A research group at the National University of Engineering in Lima is actively developing quantum computing techniques and software. They have made major contributions to the open-source software development kit Qiskit, including the invention of new algorithms, the enhancement of Qiskit Terra, and the validation of the software on their own hardware. Similarly, the Ecuadorian Centre for High Performance Computing (CEDIA) in Quito focuses on quantum computing research and has developed the Aer Simulator, a quantum computing simulator built on top of Qiskit. CEDIA has also helped to build Qiskit Terra and has worked with other organisations to promote quantum computing. Lima and Quito have both made significant contributions to the advancement of the cause.

The Qiskit tool is installed on a personal laptop to create combinational and sequential logic circuits using quantum computing by following the installation guide provided in the Qiskit documentation. An IBM Quantum Experience account is created to enable access to and execution of quantum programmes on IBM's quantum computers. A quantum circuit is built with Qiskit by organising a sequence of quantum gates that perform specified actions on qubits. A suitable backend, such as a quantum device or emulator, is chosen to test the code before running it on a real quantum computer. Once the backend has been chosen, the quantum programme can be run using Qiskit's execute function. This permits the use of quantum computing technologies to implement and simulate combinational and sequential logic circuits.

The reversible gates such as CNOT and CCNOT along with quantum gates such as Pauli’s X gate and Hadamard gates were used in designing the Flipflops. The combination of 3 CNOT gates was sufficient for the operation of one SR Flipflop. The D Flipflop and JK Flipflop additionally uses Pauli X gate and Hadamard gate respectively. The usage of Hadamard gate brings in superposition character of qubit. The 8-bit ripple carry adder uses CCNOT.

VI. CLASSICAL COMPUTATION:

The cadence tool allows the designing of universal gates such as NAND gates at the transistor level. To design and simulate circuits, the Cadence tool is installed on a Linux workstation. After that, the Cadence tool is launched, and a new design library is created. A new schematic is constructed using the proper schematic template, and design rules are chosen to guide the circuit design. Components are added to the schematic canvas by dragging and dropping them from the library and connecting them using wires or buses. The schematic is updated with pins representing input and output connections. The design is double-checked for accuracy and saved. A circuit symbol is formed in a new file, and a test bench file is created for simulation purposes. The

ADEL window is opened, and analysis parameters like transient and DC are entered.

For plotting and simulation, the desired designs are chosen. The simulation results are meticulously examined to ensure that they fulfill the required standards. The generated graphs are used to calculate the rise time, fall time, delay, and overall simulation duration. This procedure aids in assessing the performance of the planned circuits.

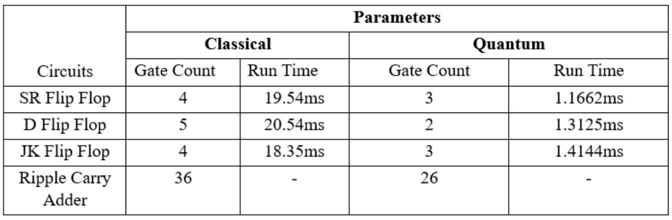
Designed NAND gates become the building blocks of Flipflops such as SR Flipflop, JK Flipflop, D Flipflop and Ripple adder. The circuit was simulated and the parameter was tabulated.

Results obtained with both classical and quantum computing are compared, and analyzed.

VII. RESULTS

On implementation of sequential and logic circuits through quantum computing and on comparing the same with classical computing we obtain the results for the two parameters, gate count and Run time as follows.

Table 4: Comparative Analysis



	Parameters			
	Classical		Quantum	
Circuits	Gate Count	Run Time	Gate Count	Run Time
SR Flip Flop	4	19.54ms	3	1.1662ms
D Flip Flop	5	20.54ms	2	1.3125ms
JK Flip Flop	4	18.35ms	3	1.4144ms
Ripple Carry Adder	36	-	26	-

The above results prove that the run time and gate count obtained during implementation using quantum computing is less and the circuit simulation with the same is more efficient in working.

VIII. CONCLUSION

Traditional computing has by far been one of the most used methods of simulating combinational and sequential circuits. These circuits find uses in various applications such as arithmetic, Boolean operations, storing data etc. The delay offered by each of these circuits although low, is still big when we realized that Quantum computing can reduce the delay further and increase the efficiency to enormous amounts. Therefore, our project aims at bringing about the difference in the delays offered by each of the circuits. To do this, the circuits such as the D Flip Flop, SR Flip Flop and the 8-bit ripple carry adder are initially simulated conventionally using a tool such as Cadence simulation tool. n. For quantum-based simulation, tools such as the Qiskit simulation tool and the IBM open-source software for quantum simulation are available. Our project utilizes the Qiskit simulation tool. The gates used here are different compared to the conventional gates and the method adopted also varies. Quantum computing offers the potential for solving problems that are intractable for classical computers, even if they require more gates or have longer run times. Moreover, the field of quantum computing is rapidly evolving, and future advancements may address the current limitations and significantly improve quantum computation performance. In summary, we can say that ongoing research and technological advancements may lead to the development of more efficient algorithms, improved hardware, and ultimately a paradigm shift where quantum computation becomes the preferred approach for certain circuit design tasks.

IX. FUTURE SCOPE

Quantum computing has the potential to revolutionize various fields, including the design and implementation of combinational and sequential circuits. While quantum computers are still in the early stages of development, they offer unique capabilities that could lead to significant advancements in circuit design and optimization. Here are some aspects of the future scope of quantum computing for combinational and sequential circuits:

1. Quantum Algorithms: Quantum computing can potentially provide more efficient algorithms for solving combinatorial optimization problems, which are at the heart of circuit design and optimization. Quantum algorithms like the Quantum Approximate Optimization Algorithm (QAOA) and the Quantum Annealing

Algorithm (QAA) show promise in finding optimized solutions for complex circuit designs.

2. Quantum-inspired Optimization: Even without fully realized quantum computers, quantum-inspired optimization techniques, such as quantum annealing or variational quantum algorithms, can be applied to circuit design problems. These methods leverage quantum concepts to enhance classical optimization algorithms, providing improved solutions and efficiency.

3. Circuit Simulation and Verification: Quantum computers can be used to simulate and verify the behavior of complex circuits, including combinational and sequential designs. By harnessing the inherent quantum properties, such as superposition and entanglement, quantum simulators can potentially handle larger circuits more efficiently than classical simulators.

4. Quantum Gate Implementation: Quantum computing enables the design and implementation of specialized quantum gates that can be used in the construction of combinational and sequential circuits. These gates leverage quantum phenomena to perform computations that are not possible with classical gates, potentially leading to improved circuit functionality and performance.

X. REFERENCES

[1]. Bhat, H.A., Khanday, F.A., Kaushik, B.K., Bashir, F. and Shah, K.A., 2022. “Quantum computing: fundamentals, implementations and applications.” IEEE Open Journal of Nanotechnology, 3, pp.61-77.

[2]. Rahul, R., Hegde, N.N., Vigneshwaren, S., Sirajudeen, M. and Jayalatsumi, U., “QUANTUM BASED D FLIP–FLOP AND SR FLIP–FLOP SIMULATION.”

[3]. Thapliyal, Himanshu, M. B. Srinivas, and Mark Zwolinski. "A beginning in the reversible logic synthesis of sequential circuits." (2005).

[4]. Cuccaro, Steven A., Thomas G. Draper, Samuel A. Kutin, and David Petrie Moulton. "A new quantum ripple-carry addition circuit." arXiv preprint quantph/0410184 (2004).

[5]. Thapliyal, H. and Ranganathan, N., 2010, January. “Design of reversible latches optimized for quantum cost, delay and garbage outputs.” In 2010 23rd international conference on VLSI design (pp. 235-240). IEEE..

[6]. Sebastiano, F., Homulle, H., Patra, B., Incandela, R., van Dijk, J., Song, L., Babaie, M., Vladimirescu, A. and Charbon, E., 2017, June. “Cryo-CMOS electronic control for scalable quantum computing.” In Proceedings of the 54th Annual Design Automation Conference 2017 (pp. 1-6).

[7]. Sayem, Abu Sadat Md, and Masashi Ueda. "Optimization of reversible sequential circuits." arXiv preprint arXiv:1006.4570 (2010).

[8]. Thapliyal, Himanshu, and Nagarajan Ranganathan. "Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs." ACM Journal on Emerging Technologies in Computing Systems (JETC) 6.4 (2010): 1-31.

[9]. National Academies of Sciences, Engineering, and Medicine. "Quantum computing: progress and prospects." (2019).

[10]. Taha, Saleem Mohammed Ridha. “Reversible logic synthesis methodologies with application to quantum computing.” Vol. 37. Switzerland: Springer, 2016.

[11]. Bhattacharya, Soham, and Anindya Sen. "A Review on Reversible Computing and it’s applications on combinational circuits." International Journal 9.6 (2021).

[12]. Lee, Jae-Seung, Yongwook Chung, Jaehyun Kim, and Soonchil Lee. "A practical method of constructing quantum combinational logic circuits." arXiv preprint quant-ph/9911053 (1999).

[13]. Manjunath, K. M., PS, A. L. H., Pagi, A., & Ulaganathan, J. (2015, December). “Analysis of various full-adder circuits in cadence.” In 2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT) (pp. 90-97). IEEE.

[14].Martin, Antonio J. Lopez. "Cadence design environment." New Mexico State University, Tutorial paper (2002): 35.

[15].Chanda, Manash, et al. "Design of sequential circuits using single-clocked Energy efficient adiabatic Logic for ultra-low power application." 18th International Symposium on VLSI Design and Test. IEEE, 2014.

[16].Kalyani, P., P. Satish Kumar, and P. Chandra Sekhar. "Design of subthreshold adiabatic logic based combinational and sequential circuits." 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI). IEEE, 2017.

Cite this article as :

Neha M Harapanhalli, Kavipriya M, Ishwari Jigajinni, Dr. Keerti Kulkarni, "Implementation of Combinational and Sequential Logic Circuits using Quantum Computing", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 3, pp.430-439, May-June-2023. Available at doi : https://doi.org/10.32628/CSEIT23903106

Journal URL : https://ijsrcseit.com/CSEIT23903106