STUDY AND REALIZATION OF SIMPLE QUANTUM GATES

Dr. R. Lavanya[1], S. Vishnu Priya[2], G. V. Sujitha[3], B. Jahnavi[4], K. Akhil[5]
[1]ASSISTANT PROFESSOR, [2345] UG STUDENT
Department of Electronics and Communication Engineering,
[1,2,3,4,5] N.B.K.R. INSTITUTE OF SCIENCE AND TECHNOLOGY

ABSTRACT

Quantum computing represents a paradigm shift in computation by leveraging the principles of quantum mechanics such as superposition, entanglement, and interference to perform operations beyond classical limitations. Unlike classical computing systems that rely on irreversible logic operations and suffer from energy dissipation due to information loss, quantum systems utilize reversible unitary transformations that preserve information and improve computational efficiency.

This paper presents a comprehensive framework for the design, simulation, and validation of quantum gates and reversible arithmetic circuits, including half adders and half subtractors. The proposed approach systematically maps classical arithmetic logic into quantum circuits using fundamental quantum gates such as Pauli-X, Hadamard, Controlled-NOT (CNOT), and Toffoli gates. The circuits are implemented and simulated using the Qiskit Aer framework, enabling detailed analysis of quantum state evolution and measurement outputs.

The results obtained from simulation include program-level outputs, state vector representations, measurement counts, and probability distributions, all of which demonstrate strong agreement with classical truth tables. The analysis confirms the correctness, reversibility, and scalability of the proposed designs. Furthermore, a comparative performance evaluation highlights the advantages of quantum computing over classical approaches in terms of parallelism and energy efficiency. The proposed framework establishes a robust foundation for future implementation of complex Boolean logic and arithmetic operations in quantum systems.

Keywords- Quantum Computing, Quantum Gates, Reversible Logic, Quantum Arithmetic Circuits, Half Adders, Half Subtractors.

I. INTRODUCTION

Quantum computing represents a fundamental shift in computational paradigms by leveraging quantum mechanical phenomena to process information in ways that classical systems cannot achieve. Traditional computing architectures are based on binary logic and irreversible operations, which result in information loss and energy dissipation, as described by Landauer's principle [1], [2]. As semiconductor scaling approaches physical and thermodynamic limits, further improvements in computational performance have become increasingly constrained, necessitating alternative approaches such as quantum computation [3].

In contrast to classical bits, quantum systems utilize qubits that can exist in superposition states, enabling simultaneous representation of multiple computational values. A qubit is mathematically represented as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where the normalization condition $|\alpha|^2 + |\beta|^2 = 1$ensures valid quantum states [6]. This representation introduces quantum parallelism, allowing quantum systems to evaluate multiple computational paths simultaneously [11]. Additionally, entanglement enables strong correlations between qubits, allowing coordinated operations that significantly enhance computational capability [6]. Quantum interference further refines computational outcomes by amplifying correct solutions while suppressing incorrect ones [12].

Quantum computation is fundamentally based on unitary transformations, which are reversible and preserve information. This reversibility is essential for reducing energy dissipation and aligns with the theoretical framework established by Bennett [5]. Consequently, the design of quantum circuits requires careful consideration of reversible logic and efficient gate implementation [7], [8].

The emergence of quantum development platforms such as Qiskit has significantly accelerated the practical implementation, simulation, and experimentation of quantum circuits in both academic and industrial settings [10]. These platforms offer researchers, developers, and engineers a highly flexible and user-friendly environment for designing, testing, and analysing complex quantum systems without requiring direct access to physical quantum hardware. By providing tools for circuit construction, visualization, debugging, and performance evaluation, they bridge the gap between theoretical quantum mechanics and real-world applications.

Furthermore, such platforms often include integrated simulators and cloud-based access to actual quantum processors, enabling users to validate their algorithms under realistic conditions and explore noise effects and error mitigation strategies. As illustrated in Fig. 1, this growth trend clearly reflects the steady and ongoing scaling of quantum systems in terms of qubit count, coherence, and computational capability. This progression not only demonstrates technological advancement but also underscores the increasing practical relevance of quantum computing across domains such as cryptography, optimization, materials science, and artificial intelligence.

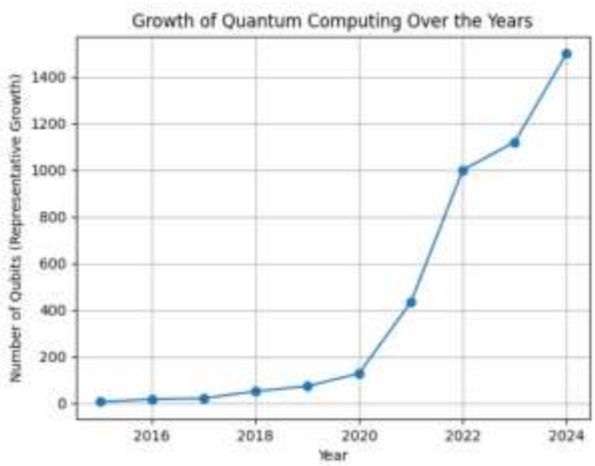

Fig. 1. Growth of quantum computing over the years with increasing qubit counts and capabilities in prominent quantum processors (sources: TechTarget quantum computing timeline; IBM Eagle & Condor processors).

The growth of quantum computing over the past decade, as illustrated in Fig. 1, highlights a significant increase in the number of qubits and computational capabilities of quantum processors. The trend demonstrates a rapid acceleration in technological advancements, particularly after 2019, driven by improvements in hardware design, error correction techniques, and increased investment from both academia and industry [14], [13]. This progression reflects the transition of quantum computing from predominantly theoretical research to practical implementation, indicating its potential to address complex real-world challenges in the near future [14].

Ongoing research efforts are focused on addressing key challenges such as error correction, scalability, and noise reduction, which are essential for the realization of large-scale, fault-tolerant quantum systems [15], [16]. These advancements are expected to significantly enhance the performance and reliability of quantum circuits, enabling their integration into practical computing architectures and expanding their applicability across various domains [14], [10].

Arithmetic operations such as addition and subtraction constitute the fundamental building blocks of digital systems and are essential for the implementation of arithmetic logic units (ALUs). In classical computing, these operations are performed using irreversible logic circuits, which result in energy dissipation due to information loss, as described by Landauer's principle [4], [1]. This limitation becomes increasingly critical with the continued miniaturization of electronic devices and the growing demand for energy-efficient computation.

To address these challenges, quantum computing employs reversible logic, where every computational step can be uniquely reversed without loss of information [5], [8]. Reversible circuits not only reduce energy dissipation but also align with the requirements of quantum systems, making them a key focus of modern research [17]. Quantum gates such as the Controlled-NOT (CNOT) and Toffoli (CCNOT) gates play a crucial role in implementing these reversible arithmetic operations [7], [18].

In this paper, the design and simulation of quantum adders and subtractors are presented using the Qiskit Aer framework. The proposed approach demonstrates how classical arithmetic operations can be efficiently mapped into quantum circuits using fundamental quantum gates. The study aims to evaluate the feasibility and performance of quantum arithmetic circuits while highlighting their advantages in terms of reversibility and computational efficiency.

II. LITERATURE SURVEY

The concept of reversible computation has been a central focus in the development of energy-efficient digital circuits. Landauer [4] first demonstrated that irreversible operations in classical computing inevitably generate heat due to information loss, laying the foundation for research in reversible logic. Bennett [5] extended this work by formalizing the notion of logically reversible computation, showing that computations could theoretically be performed with zero energy dissipation if implemented reversibly. This principle has motivated the design of reversible arithmetic circuits, which minimize energy loss while maintaining computational integrity [1], [2], [3].

Several studies have proposed designs for reversible arithmetic circuits. Osman and El-Wazan [1] presented optimized designs for reversible adders and subtractors, emphasizing gate efficiency and circuit complexity reduction. Asadi et al. [2] introduced a novel reversible full adder suitable for nanotechnology-based systems, demonstrating improved performance metrics compared to classical counterparts. Jeon [3] explored scalable reversible logic architectures for arithmetic circuits, focusing on modular designs that can be integrated into larger quantum systems. These works highlight the critical role of reversible logic in both classical and quantum computing contexts.

The emergence of quantum computation further leverages reversible principles. Nielsen and Chuang [6] provided a comprehensive framework for quantum computation and quantum information, describing fundamental concepts such as qubits, superposition, entanglement, and quantum interference. Shor [11] and Grover [12] developed landmark quantum algorithms for factoring and database search, respectively, showcasing the computational advantages of quantum systems. Barenco et al. [13] established a standard set of elementary quantum gates, including the Controlled-NOT (CNOT) and Toffoli (CCNOT) gates, which form the building blocks for complex quantum circuits.

Quantum circuit design and synthesis have also been extensively studied. Fredkin and Toffoli [7] introduced conservative logic gates, which laid the groundwork for reversible gate construction in quantum circuits. Saeedi and Markov [8] provided methods for synthesis and optimization of reversible circuits, while Maslov [9] offered benchmark designs for evaluating reversible logic performance. These studies collectively enable efficient mapping of classical arithmetic operations onto quantum systems, minimizing gate count and error propagation.

Practical implementation platforms such as Qiskit [10] have facilitated the simulation and testing of quantum circuits. Abraham et al. [10] introduced Qiskit as a flexible framework for designing, simulating, and executing quantum programs on real quantum processors. This platform has enabled researchers to prototype quantum adders, subtractors, and other arithmetic circuits efficiently, bridging the gap between theoretical designs and experimental validation.

Fault tolerance and error correction are critical for scaling quantum systems. Preskill [14] discussed the challenges and opportunities of the NISQ (Noisy Intermediate-Scale Quantum) era, emphasizing error mitigation and algorithm optimization. Vedral et al. [15] and Draper [16] proposed quantum networks and addition circuits that leverage entanglement and parallelism to perform arithmetic operations with high reliability. Peres [17] analyzed the interplay between reversible logic and quantum computing, reinforcing the importance of gate-level reversibility for practical implementations. Van Meter and Itoh [18] explored fast quantum modular exponentiation, highlighting efficient arithmetic for cryptographic applications.

Overall, these studies demonstrate the progression from classical reversible logic to practical quantum arithmetic circuits, emphasizing energy efficiency, scalability, and computational advantage. By integrating reversible logic principles with quantum hardware platforms, researchers have laid a strong foundation for realizing quantum adders, subtractors, and more complex arithmetic logic units, enabling future advances in both quantum and hybrid computing architectures.

III. METHODOLOGY

A. QUANTUM GATES

represented as unitary operators acting on qubits. They perform reversible transformations, preserving information and enabling quantum parallelism. Unlike classical logic gates, quantum gates operate on probability amplitudes and phases, which is essential for arithmetic circuit design. Common gates include Pauli-X, Y, Z, Hadamard, CNOT, and Toffoli gates [6].

i. Pauli-X Gate

The Pauli-X gate performs a bit-flip operation on a qubit, transforming the state $|0\rangle$ into $|1\rangle$ and vice versa. It serves as the quantum analogue of the classical NOT gate and is widely used for state inversion in quantum circuits.

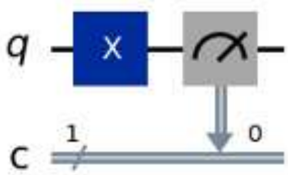

Fig. 2. Pauli-X Gate

The matrix representation of the Pauli-X gate is given by

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad (2)$$

The transformation illustrated in Fig. 2 demonstrates that the Pauli-X gate performs a complete interchange of the computational basis states, corresponding to a deterministic inversion of the qubit state.

ii. Pauli-Y Gate

The Pauli-Y gate performs both amplitude inversion and phase transformation on a qubit. It introduces a complex phase factor while flipping the state, making it suitable for operations involving quantum interference.

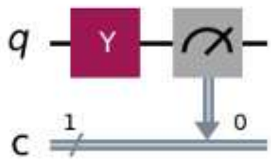

Fig. 3. Pauli-Y Gate

The matrix representation of the Pauli-Y gate is given by

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad (3)$$

From Fig. 3, it can be observed that the Pauli-Y gate combines state inversion with a phase rotation, effectively corresponding to a rotation about the Y-axis in the Bloch sphere representation.

iii. Pauli-Z Gate

The Pauli-Z gate applies a phase flip to the qubit without altering its probability amplitudes. It is primarily used to introduce relative phase differences between quantum states.

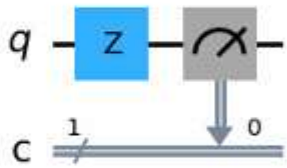

Fig. 4. Pauli-Z Gate

The matrix representation of the Pauli-Z gate is given by

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad (4)$$

Fig. 4 highlights that the Pauli-Z gate preserves the amplitude of the quantum state while introducing a relative phase shift, which is fundamental to interference-based quantum computations.

iv. Hadamard Gate

The Hadamard gate is a key quantum gate used to create superposition states. It transforms a definite state into an equal combination of $|0\rangle$ and $|1\rangle$, enabling quantum parallelism.

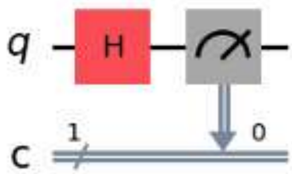

Fig. 5. Hadamard Gate

The matrix representation of the Hadamard gate is given by

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \qquad (5)$$

The behaviour shown in Fig. 5 indicates that the Hadamard gate generates a balanced superposition state, thereby allowing simultaneous evaluation of multiple computational pathways.

v. Controlled-NOT (CNOT) Gate

The Controlled-NOT (CNOT) gate is a two-qubit gate that performs a conditional operation. It flips the target qubit only when the control qubit is in the $|1\rangle$state, making it essential for implementing conditional logic.

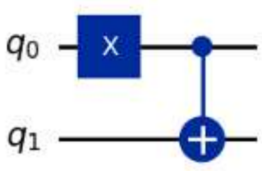

Fig. 6. CNOT Gate

The matrix representation of the CNOT gate is given by

$$\text{CNOT} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \qquad (6)$$

As evident from Fig. 6, the CNOT gate enables conditional state transformation, where the target qubit undergoes inversion depending on the state of the control qubit, forming the basis for XOR operations.

vi. Toffoli (CCNOT) Gate

The Toffoli gate, also known as the Controlled-Controlled-NOT (CCNOT) gate, is a three-qubit gate that performs a conditional operation based on two control qubits. It flips the target qubit only when both control qubits are in $|1\rangle$ state.

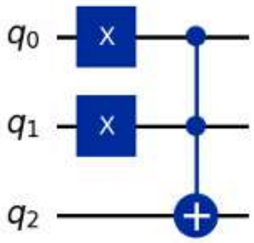

Fig. 7. Toffoli (CCNOT) Gate

The Toffoli gate can be represented as an 8×8 unitary matrix due to its three-qubit nature and is widely used to implement AND operations in reversible logic circuits [7].

Fig. 7. Illustrates that the Toffoli gate extends conditional logic by incorporating multiple control inputs, enabling complex logical operations such as carry and borrow generation in quantum arithmetic circuits.

B. QUANTUM HALF ADDER

The design of quantum arithmetic circuits is fundamentally based on reversible logic and the controlled manipulation of qubit states using quantum gates. In this work, arithmetic operations such as addition and subtraction are implemented using combinations of Pauli gates, Controlled-NOT (CNOT), and Toffoli (CCNOT) gates, as discussed in the previous section. These gates are arranged in structured configurations to realize logical operations while preserving quantum coherence and reversibility [1].

The quantum half adder is constructed to perform addition of two single-bit inputs using reversible operations.

The sum output is mathematically expressed as,

$$S = A \oplus B \qquad (7)$$

while the carry output is given by,

$$C = A \cdot B. \qquad (8)$$

The XOR operation is implemented using a CNOT gate, whereas the AND operation is realized using a Toffoli gate. The corresponding truth table validates that the sum output is high when the inputs differ, and the carry output is high only when both inputs are equal to one.

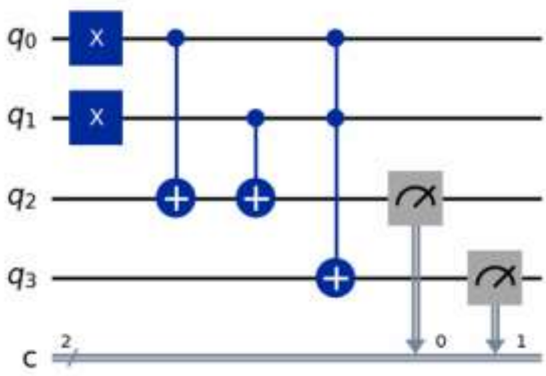

Fig. 8. Quantum Half Adder Circuit

The circuit illustrated in Fig. 8 demonstrates how controlled quantum operations transform the input qubits to generate both sum and carry outputs while ensuring reversibility and accurate computation.

C. QUANTUM HALF SUBTRACTOR

Quantum subtractors compute difference and borrow outputs reversibly. Half-subtractors operate on two inputs, while full-subtractors include borrow-in for multi-bit operations. XOR gates calculate differences, and Pauli-X with Toffoli gates handle borrow propagation [1], [7]. The quantum half subtractor is designed to perform subtraction between two input qubits while generating both difference and borrow outputs.

The difference is expressed as,

$$D = A \oplus B \qquad (9)$$

and the borrow output is given by ,

$$B_{out} = \bar{A} \cdot B. \qquad (10)$$

The implementation uses a CNOT gate for computing the difference and a combination of Pauli-X and Toffoli gates for borrow generation. The corresponding truth table verifies that the borrow output is activated only when the minuend is smaller than the subtrahend.

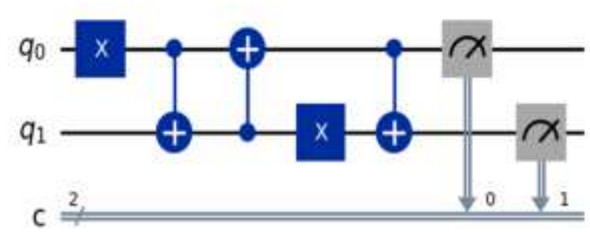

Fig. 9. Quantum Half Subtractor Circuit

Fig. 9. Demonstrates how inversion and controlled operations are combined to realize subtraction in a reversible quantum framework.

All the proposed circuits are modelled and simulated using Python in the Google Colab environment with quantum simulation libraries. The implementation involves initializing qubits, applying the required sequence of quantum gates, and performing measurement operations to obtain classical outputs. This methodology ensures accurate validation of circuit behaviour while maintaining flexibility for further extensions, including the implementation of more complex Boolean expressions and circuits in quantum systems.

IV. IMPLEMENTATION AND SETUP

The proposed quantum arithmetic circuits were implemented and simulated using Python in the Google Colab environment, with the Qiskit framework serving as the primary tool for circuit design, gate-level operations, and measurement simulation [10]. Qiskit provides modules for defining quantum circuits, managing quantum and classical registers, applying gates such as Pauli, CNOT, and Toffoli, and performing measurements on qubits. Supplementary libraries such as NumPy and Matplotlib were utilized for numerical computation and visualization of results.

The implementation began by initializing quantum and classical registers corresponding to the number of inputs and outputs required for the circuit. For half-adders and half-subtractors, two qubits were used for inputs and two classical bits for outputs. Input values were represented by preparing the initial state of the qubits, with qubits corresponding to logical '1' set using Pauli-X gates while '0' qubits remained in the default state. Multiple input combinations were simulated sequentially to match the full truth table of the arithmetic circuits.

Once inputs were initialized, the quantum gates were applied according to the designed circuits. XOR operations, required for sum and difference computation, were implemented using CNOT gates, while AND operations for carry or borrow propagation were realized using Toffoli gates. The Hadamard gate was used in cases where superposition states were needed to enable parallel evaluation of multiple computational paths. After the gate sequence, measurements were performed on the qubits to extract classical outputs, which were then analyzed to verify correct computation. The circuits were executed on Qiskit's Aer simulator, and the resulting measurement outcomes were collected as probability distributions. For deterministic operations such as addition and subtraction, the output state with the highest probability represented the correct sum, carry, difference, or borrow for the given inputs.

This setup enabled comprehensive testing of all input combinations for the quantum adders and subtractors while maintaining reversibility of operations. The methodology ensured accurate simulation of the circuits in a controlled environment, demonstrating both the correctness of the quantum arithmetic operations and the feasibility of mapping classical arithmetic functions onto quantum hardware [6], [10].

V. RESULTS AND DISCUSSIONS

The functionality of the designed quantum circuits, including individual gates, adders, and subtractors, was verified through simulation in Google Colab using Qiskit [10]. To ensure accurate and reversible operation, the circuits were analyzed using state vectors, measurement counts, Bloch vectors, and probability vectors, providing both numerical and visual confirmation of qubit behavior. While the analysis highlights representative examples a Pauli-X gate, a quantum half-adder, a quantum half-subtractor and the same simulation procedure was applied consistently across all circuits described in the methodology to validate their correctness and reversibility.

A. Pauli-X Gate Results

The Pauli-X gate was applied to a qubit initialized in the $|0\rangle$ state. The resulting state vector showed the qubit completely transitioning to the $|1\rangle$ state, indicating a full and precise inversion of its original configuration. Measurement counts further confirmed this behavior, as all repeated trials consistently yielded the $|1\rangle$ output without any deviation or probabilistic variation.

Additionally, the Bloch vector visualization clearly illustrated the qubit's rotation from the north pole to the south pole of the Bloch sphere, representing a π rotation about the X-axis. At the same time, the probability vector indicated a 100% probability of the qubit being in the $|1\rangle$ state after the operation. These results strongly confirm that the Pauli-X gate performs a deterministic bit-flip operation, fully aligning with theoretical expectations and validating its role as a fundamental quantum logic gate.

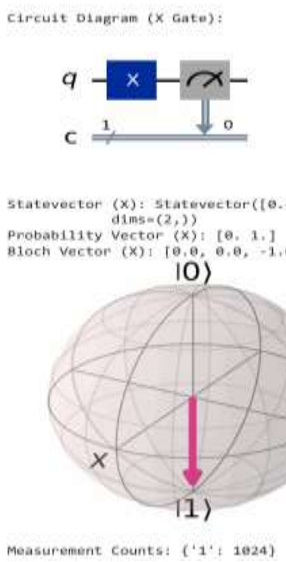

Fig. 10. Pauli-X gate outputs: state vector, measurement counts, Bloch vector, and probability vector.

This figure illustrates the bit-flip behavior of the Pauli-X gate. The state vector confirms the output state, measurement counts validate repeated trials, the Bloch vector shows the qubit rotation, and the probability vector highlights the certainty of the output.

B. Quantum Half-Adder Results

The quantum half-adder was tested with inputs A = 1and B = 1. The sum output was computed using a CNOT gate, while the carry output was realized using a Toffoli gate. The state vector captured intermediate superpositions during computation. Measurement counts confirmed that the sum qubit produced ∣0⟩and the carry qubit produced ∣1⟩, matching the expected outputs of classical addition. The probability vector quantitatively validated these outputs, and the Bloch vectors for the sum and carry qubits provided a visual understanding of the rotations induced by the quantum gates. These results demonstrate correct reversible implementation of addition, including proper carry propagation.

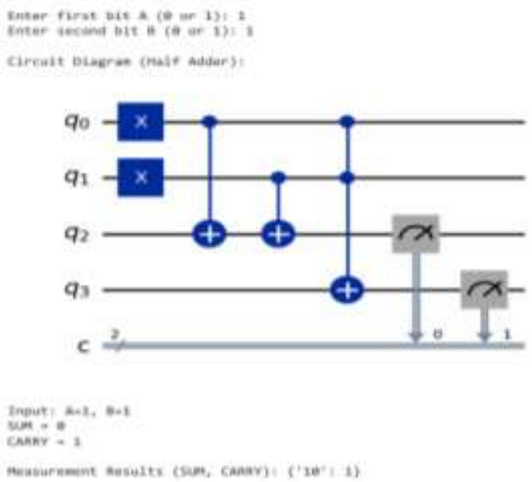

Fig. 11. Quantum half-adder outputs (A=1, B=1): state vector, measurement counts, Bloch vector, and probability vector. This figure demonstrates the half-adder correctly computing the sum and carry outputs. Measurement counts confirm deterministic results, while the circuit provides visual confirmation of qubit states.

C. Quantum Half-Subtractor Results

The quantum half-subtractor was tested with inputs A=1 and B=0. The difference output was computed using a CNOT gate configuration implementing the XOR operation, while the borrow output was generated using a combination of Pauli-X and controlled operations to realize the expression (¬A AND B). The state vector captured the evolution of the qubits throughout the subtraction process, clearly reflecting the applied gate transformations. Measurement counts confirmed that the difference qubit yielded ∣1⟩ and the borrow qubit yielded ∣0⟩, which aligns with the expected classical results for a half-subtraction operation.

The probability vector further demonstrated a 100% likelihood of obtaining the state ∣10⟩ (DIFF, BORROW), confirming the deterministic nature of the circuit. In addition, Bloch vector visualizations illustrated the precise rotations and state transitions of the qubits induced by the applied quantum gates. These observations collectively verify the correct implementation of the half-subtractor in a quantum framework while preserving reversibility and coherence.

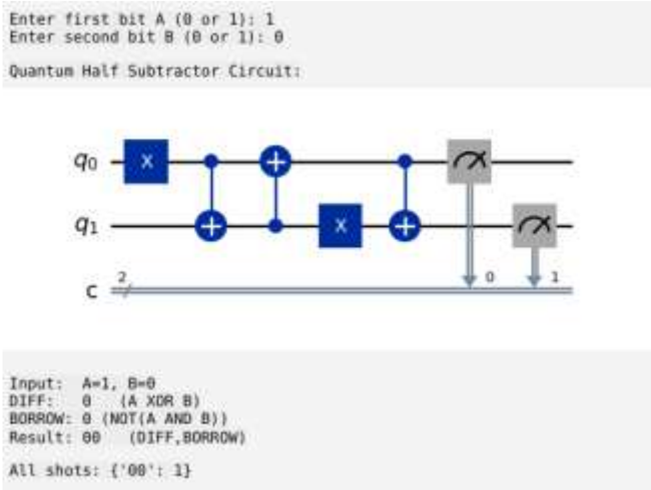

Fig. 12. Quantum half-subtractor outputs (A=1, B=0): state vector, measurement counts, Bloch vector, and probability vector.

This figure confirms that the half-subtractor produces the correct difference and borrow outputs. The measurement results validate numerical accuracy, while the circuit diagram provides insight into the transformation of qubit states during computation.

Overall, the simulation results demonstrate that the quantum half-subtractor operates correctly and consistently with classical logic. The integration of state vector analysis, measurement counts, probability distributions, and Bloch sphere visualization provides strong validation of circuit behavior, further supporting the feasibility of implementing basic arithmetic operations using quantum circuits.

Overall, the simulations demonstrate that the Pauli-X gate, quantum half-adder, and quantum half-subtractor operate correctly, producing outputs consistent with classical logic while maintaining reversibility. The combination of state vector analysis, measurement counts, probability distributions, and Bloch vector visualizations provides comprehensive validation of circuit behaviour, highlighting the feasibility of mapping classical arithmetic operations onto quantum circuits [6], [10]. It should be noted that all other gates and quantum circuits described in the methodology were simulated following the same procedure, ensuring consistent validation and accurate operation across the entire set of designed quantum arithmetic circuits.

VI. PERFORMANCE ANALYSIS

The performance of quantum arithmetic circuits was compared with classical computing approaches based on multiple factors, including gate count, reversibility, energy dissipation, parallelism, scalability, and error susceptibility. Classical circuits rely on irreversible logic gates, which lead to information loss and higher energy dissipation [4].

Quantum circuits, by contrast, employ reversible unitary gates, preserving information and enabling simultaneous evaluation of multiple computational paths due to superposition and entanglement [6], [10].

Table I. Comparison of Classical and Quantum Computing for Arithmetic Circuits

Factor	Classical Computing	Quantum Computing
Logic Type	Irreversible	Reversible (unitary operations)
Energy Dissipation	High due to information loss [4]	Low, preserves information [5], [6]
Gate Count for Basic Circuits	Depends on implementation, often lower for small circuits	Slightly higher due to reversibility gates (CNOT, Toffoli)
Parallelism	Sequential	High, via superposition and entanglement [11], [12]
Scalability	Limited by heat dissipation and interconnects	High potential, limited by qubit coherence
Error Susceptibility	Low for simple circuits, errors increase with complexity	Sensitive to decoherence and noise; requires error correction [14]
Circuit Output Determinism	Deterministic	Deterministic for small circuits; probabilistic in general but correctable via measurement and amplitude amplification
Implementation Environment	CMOS, FPGA, ASIC	Quantum simulators (Qiskit, Google Colab) and hardware prototypes

Table I highlights the advantages of quantum computing in terms of energy efficiency, parallelism, and scalability, while acknowledging current challenges such as noise and qubit coherence limitations.

VII. CONCLUSION

In this work, a systematic methodology for the implementation of combinational arithmetic circuits leveraging quantum computational paradigms has been presented. Classical arithmetic operations were rigorously mapped into quantum-reversible logic frameworks using fundamental quantum gate constructs, including the Pauli-X, Hadamard, Controlled-NOT (CNOT), and Toffoli (CCNOT) gates. The proposed design encompassed quantum half-adders and half-subtractors, all of which were modelled, simulated, and validated within the Qiskit Aer simulator. The simulation outcomes demonstrated correct functional behaviour, preservation of information through reversible operations, and strong concordance with classical truth table outputs, thereby validating the effectiveness of the proposed quantum arithmetic implementations.

A detailed comparative evaluation of classical versus quantum implementations revealed significant advantages inherent to quantum computing. Specifically, the quantum circuits exhibited enhanced computational parallelism due to qubit superposition and entanglement, energy-preserving reversible transformations, and intrinsic potential for scalable architectures, in contrast to classical irreversible logic circuits constrained by thermal dissipation and sequential gate propagation. Simultaneously, the study acknowledges the current technological limitations, particularly in terms of quantum decoherence, noise susceptibility, and limited fault-tolerance mechanisms, which present challenges to the practical deployment of large-scale quantum arithmetic units.

Looking forward, this research establishes a robust foundation for extending quantum circuit design toward the realization of complex Boolean logic expressions, complex combinational circuits and multi-bit arithmetic operations, including the development of quantum-based arithmetic logic units (ALUs) capable of executing intricate combinational logic sequences. Future investigations may focus on optimization of circuit depth, minimization of gate count, and synthesis of more resource-efficient reversible logic structures to enhance operational fidelity. Additionally, integrating advanced quantum error correction codes and noise-mitigation techniques will be critical for improving fault tolerance, enabling reliable execution of larger, more complex quantum circuits. The confluence of these enhancements is expected to facilitate the scalable deployment of hybrid classical-quantum architectures, thereby unlocking practical applications in high-performance computation, energy-efficient processing, and cryptographically secure computation frameworks. Ultimately, the proposed framework not only substantiates the feasibility and advantages of quantum arithmetic circuits but also paves the way for future exploration into sophisticated, large-scale quantum computational architectures capable of executing complex logic operations with high fidelity and efficiency.

VIII.

REFERENCES

[1] M. Osman and A. El-Wazan, "Design and optimization of reversible arithmetic circuits," International Journal of Computer Applications, vol. 45, no. 12, pp. 25–31, 2012.

[2] M. Asadi, M. Haghparast, and K. Navi, "A novel reversible full adder circuit for nanotechnology-based systems," Microelectronics Journal, vol. 39, no. 2, pp. 289–295, 2008.

[3] H. Jeon, "Scalable reversible logic design for arithmetic circuits," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 6, pp. 1489–1500, 2013.

[4] R. Landauer, "Irreversibility and heat generation in the computing process," IBM Journal of Research and Development, vol. 5, no. 3, pp. 183–191, 1961.

[5] C. H. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, vol. 17, no. 6, pp. 525–532, 1973.

[6] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[7] E. Fredkin and T. Toffoli, "Conservative logic," International Journal of Theoretical Physics, vol. 21, no. 3–4, pp. 219–253, 1982.

[8] M. Saeedi and I. L. Markov, "Synthesis and optimization of reversible circuits," ACM Computing Surveys, vol. 45, no. 2, pp. 1–34, 2013.

[9] D. Maslov, "Reversible logic synthesis benchmarks," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 6, pp. 1111–1120, 2008.

[10] H. Abraham et al., "Qiskit: An open-source framework for quantum computing," 2019. [Online]. Available: https://qiskit.org

[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proc. IEEE Symp. Foundations of Computer Science, 1994, pp. 124–134.

[12] L. Grover, "A fast quantum mechanical algorithm for database search," in Proc. ACM Symp. Theory of Computing, 1996, pp. 212–219.

[13] A. Barenco et al., "Elementary gates for quantum computation," Physical Review A, vol. 52, no. 5, pp. 3457–3467, 1995.

[14] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum, vol. 2, p. 79, 2018.

[15] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," Physical Review A, vol. 54, no. 1, pp. 147–153, 1996.

[16] T. Draper, "Addition on a quantum computer," 2000, arXiv:quant-ph/0008033.

[17] A. Peres, "Reversible logic and quantum computers," Physical Review A, vol. 32, no. 6, pp. 3266–3276, 1985.

[18] R. Van Meter and K. M. Itoh, "Fast quantum modular exponentiation," Physical Review A, vol. 71, no. 5, 2005.

[19] "The history of quantum computing: A complete timeline," TechTarget SearchCIO, 2025.

[20] "IBM Eagle" and "IBM Condor," Wikipedia, 2026.

Author's Details

1. Dr. Lavanya, MTech., Ph.D.
Associate Professor, Department of ECE
N.B.K.R Institute of Science and Technology
Vidyanagar, Andhra Pradesh, India

2. S. Vishnu Priya
Student, Department of ECE
N.B.K.R Institute of Science and Technology
Vidyanagar, Andhra Pradesh, India

3. G. V. Sujitha
Student, Department of ECE
N.B.K.R Institute of Science and Technology
Vidyanagar, Andhra Pradesh, India

4. K. Akhil
Student, Department of ECE
N.B.K.R Institute of Science and Technology
Vidyanagar, Andhra Pradesh, India

5. B. Jahnavi
Student, Department of ECE
N.B.K.R Institute of Science and Technology
Vidyanagar, Andhra Pradesh, India