

Implementation of 2 Qubit Quantum Arithmetic Logic Unit on Qiskit

M.P. Oviya Nandini¹ and Sangam Kumar²

¹⁻²PES University, Bangalore, India

Email: sct.mpo.0320@gmail.com, hgsangam@pes.edu

Abstract— This paper describes the design and development of a 2-bit Quantum Arithmetic Logic Unit (ALU) with Qiskit. The ALU executes arithmetic operations—addition, two's complement subtraction, multiplication—and logical operations such as AND, OR, and XOR, controlled by a special unit. The design optimizes qubit utilization for compatibility with noisy intermediate-scale quantum (NISQ) devices. The operations were manually tested through simulations to ensure that the quantum circuits functioned as intended. The presented work illustrates the applicability of classical ALU concepts to quantum computing architectures and lays the ground for scalable quantum processors.

Index Terms— Quantum Computing, Quantum ALU, Quantum Logic Operations, Quantum Arithmetic Operations, Quantum Circuit Design.

I. INTRODUCTION

Quantum computing presents a radically new paradigm for computation based on the principles of superposition, entanglement and quantum parallelism. In contrast to classical systems, quantum computers can potentially carry out certain computations exponentially faster and with important benefits in areas like cryptography, optimization and sophisticated simulations.[17] The Arithmetic Logic Unit (ALU) is one of the primary building block in classical and quantum designs, tasked with executing arithmetic and logical operations. This paper presents the design and implementation of a 2-bit Quantum ALU that can execute addition, subtraction based on two's complement, multiplication and simple logical operations like AND, OR and XOR. The circuits have been designed based on quantum gates and have been manually verified by simulation for accurate working. This research also focuses on reducing qubit overhead, which is very important for efficient quantum circuit design.

II. FOUNDATIONAL CONCEPTS AND TERMINOLOGY

The quantum ALU circuits were designed and simulated using Qiskit, a widely-used open-source quantum computing framework. Manual verification of each operation was carried out through quantum circuit simulations, ensuring that the expected outcomes matched the results without relying on automated verification tools. This manual approach validates the correctness of the design while providing insights into the circuit behavior at each stage.

A. Maintaining Quantum Circuit Constraints

All operations were ensured to be reversible, a fundamental requirement for any quantum computation. Additional ancillary qubits were introduced wherever necessary to preserve unitarity without sacrificing circuit efficiency. Gate optimization techniques were carefully applied to reduce circuit depth and minimize the number of multi-qubit operations, thereby enhancing the reliability of simulations. Every circuit was tested to confirm that no information was lost during computation, ensuring the physical realizability of the designed ALU components. This disciplined approach ensured that the constructed quantum circuits were suitable not only for simulation but also for potential deployment on real quantum hardware.

B. Classical and Quantum Registers

Information is processed and stored in quantum computing in both classical and stored quantum computing in both classical and quantum registers.

- Classical Register (CR): A classical register stores measurement outcomes from quantum bits(qubits). These registers behave like traditional memory registers in classical computing. They can hold values such as 0 and 1, and are used to store the results of quantum measurements.
- Quantum Register (QR): A quantum register consists of multiple qubits that can exist in a superposition of states.[18] Quantum registers are manipulated using quantum gates, allowing for parallel computation through quantum entanglement and interference.[3]

C. How Qiskit Backend Server Works

In Qiskit, a backend is the execution target—either a quantum device or simulator—that runs quantum circuits. When a user submits a quantum circuit through Qiskit, the circuit is first transpiled into a set of low-level instructions compatible with the target backend. These instructions are then sent to the backend server, which queues the job, runs the circuit, and returns the measurement results. IBM Quantum’s cloud backends use job IDs for tracking and users can monitor job status and retrieve results once execution is complete. This system allows users to interface with quantum hardware remotely and reliably.

D. IBM Aer Simulator

The IBM Aer Simulator was essential in the development and verification of the quantum ALU. Being a high-speed quantum circuit simulator within Qiskit, Aer enabled us to debug and test quantum logic gates and algorithms on classical hardware with very close approximation of real quantum behavior. Aer facilitates:

- Verifying circuits efficiently without the need for access to noisy quantum hardware.
- Benchmarking performance against gate depth, measurement results and fidelity.
- Scaling behavior exploration through the simulation of extended versions of the 2-bit ALU.
- Integration with GPU resources (if enabled) for accelerated simulation of larger circuits. Aer functions effectively as a platform for development and testing, securing the quantum ALU’s correctness of functionality as well as readiness for deployment on real world quantum hardware or incorporation into hybrid systems. It bridges current access limitations in quantum computing while enabling practical experimentation and iterative design refinement.

E. Jobs in Qiskit

In Qiskit, a job refers to the execution request submitted to a quantum backend, which could be either a real quantum processor or a simulator. Once a quantum circuit is constructed and transpiled, it is packaged into job and sent to the backend for processing. Jobs are handled asynchronously allowing the user to monitor their status, retrieve results, or cancel execution if necessary. This job-based structure provides flexibility and efficiency especially when working with cloud-based quantum devices where queue times and execution durations may vary. Each job is assigned a unique identifier, making it easy to track and manage. The job system in Qiskit is especially important when working with real quantum hardware as it manages queuing, execution time and result collection in a structured and scalable way. It also supports batch execution of multiple circuits, enabling users to explore quantum algorithm performance efficiently across different parameters and inputs.

F. Why Execution Time Cannot be Accurately Measured

- Simulated Execution: Circuits run on simulators like Aer Simulator are classically executed. Quantum gates are treated as instantaneous matrix operations, ignoring real-world time delays.
- Backend Overheads: Real quantum hardware introduces additional time due to job queuing, circuit compilation, qubit mapping, and calibration drifts, making the timing highly variable.

- Idealized Gate Representation: In Qiskit, gates are defined as unitary matrices with no association to physical gate durations, imperfections, or system-specific constraints.
- Dynamic Hardware Conditions: Real quantum devices exhibit noise, thermal variations, and de-coherence all of which impact timing in unpredictable ways

III. DESIGN AND DEVELOPMENT APPROACH

The design of the Quantum ALU was undertaken systematically to ensure functional correctness, efficiency and minimal qubit overhead. Initial planning involved selecting the operations to be supported namely addition, subtraction using two's complement, multiplication and logical operations such as AND, OR and XOR.

Each operation was translated into a quantum circuit using fundamental quantum gates such as CNOT, Toffoli and ancillary qubits where necessary. Special emphasis was placed on maintaining the reversibility of operations to align with quantum computing requirements. Separate quantum circuits were constructed for each operation and extensive manual simulations were carried out to verify the correctness of results.

The circuits were implemented using Qiskit and all gates were arranged carefully to ensure minimum de-coherence and qubit wastage.

A. Quantum Computing Metrics and Concepts

- Quantum Circuit Depth: A dimensionless quantity that quantifies the depth of layers of quantum gates in a quantum circuit. [12] It measures the sequential complexity of the quantum algorithm.
- Quantum Gates: Quantum gates act on qubits and are represented by unitary matrices.[14] Quantum gates themselves are dimensionless but the time used for gate operations can be specified in nanoseconds (ns) or microseconds (s).
- Entanglement Measures: Entanglement is dimension less, but certain measures such as Entanglement of Formation or Concurrence may be employed for quantifying it.
- Measurement Probabilities: The chance of measuring a qubit in state $P(|0\rangle)$ or $P(|1\rangle)$, unit less and given as $P(|0\rangle)$ or $P(|1\rangle)$.
- Error Rate: The chance of errors in quantum operations, usually given in percent or as a probability.
- Quantum State Fidelity: A dimensionless value that quantifies two quantum states are.

B. Hardware Resource Comparison Between Classical and Quantum Alus

TABLE I: NO OF BITS

	Classical	Quantum
Adder (CSA)	3 bits	7 qubits
Subtractor	4 bits	10 qubits
Multiplier	6 bits	15 qubits
AND Gate	2 bits each	6 qubits
OR Gate	2 bits each	6 qubits
XOR Gate	2 bits each	4 qubits

TABLE II: NO OF REGISTERS NEEDED FOR COMPUTATION

	Classical	Quantum
Adder	3 registers	7 QR + 3 CR
Subtractor	4 registers	10 QR + 3 CR
Multiplier	4 registers	15 QR + 15 CR
AND Gate	2 registers	6 QR + 2 CR
OR Gate	2 registers	6 QR + 2 CR
XOR Gate	2 registers	4 QR + 2 CR

TABLE III

	Classical	Quantum
Adder	4 gates	7 gates
Subtractor	4 gates	13 gates
Multiplier	8 gates	4 gates
AND Gate	1 gate	5 gates
OR Gate	1 gate	5 gates
XOR Gate	1 gate	3 gates

TABLE 4. TOTAL BITS FOR ALU IMPLEMENTATION

- Classical: 16 (intermediate) + 24 = 40 bits
- Quantum: 45 to 48 qubits (if not multiplexed)

TABLE V: CIRCUIT DEPTH

	Classical	Quantum
Adder	2 stages	a = 00, b = 00 – 6 stages a = xx, b = xx – 7 stages
Subtractor	2 stages	a = 00, b = 00 – 9 stages a = xx, b = xx – 10 stages
Multiplier	2 stages	a = 00, b = 00 – 10 stages a = xx, b = xx – 11 stages
Logical Operations	-	a = 00, b = 00 – 2 stages a = xx, b = xx – 3 stages

IV. CONCLUSIONS

This project explored and validated a quantum 2-bit Arithmetic Logic Unit (ALU) capable of executing basic arithmetic and logical operations using quantum gates, superposition, and entanglement. Implemented using Qiskit, the design supported operations such as addition, subtraction (via 2's complement), multiplication, and logic functions like AND, OR, XOR, and NOT, all within a reversible computation model—a key distinction of quantum systems over classical ones. Beyond standalone quantum processing, the study also examined the potential integration of this quantum ALU into a hybrid GPU–quantum architecture. While currently limited by hardware constraints such as qubit de-coherence and error rates, the quantum ALU offers clear advantages in parallelizable and reversible computations. When paired with the high-throughput, memory-intensive capabilities of modern GPUs, this quantum module could function as a specialized co-processor for tasks that benefit from quantum speedup and entangled state processing. Such a hybrid architecture envisions a system where classical ALU cores, GPU cores, and quantum logic units collaborate—each handling components of a computation best suited to their strengths. The proposed ALU could serve as a fundamental building block in this model.

FUTURE WORK

Enhancing the ALU by extending it beyond the current 2-qubit setup to 4 or more qubits would allow the execution of more complex arithmetic and logic operations, bringing the design closer to solving real-world problems and aligning it with practical quantum computing needs. Given the sensitivity of quantum systems to noise and de-coherence, incorporating quantum error correction (QEC) techniques—such as Shor codes— will be essential for achieving fault tolerance and improved reliability.

Testing the ALU on real quantum hardware, such as IBM Q systems, can provide valuable insights into execution performance, gate fidelity, and the effects of noise. Exploring hybrid quantum–classical integration could open new avenues for efficient computing by combining the strengths of both paradigms.

Finally, developing a specialized control unit for the quantum ALU may enable smarter decision-making and facilitate seamless integration into larger and more complex digital systems, such as processors or embedded platforms.

REFERENCES

- [1] Taylor and Francis Group. Available: <https://www.taylorfrancis.com/books/9781032642079>.
- [2] ICISC 2023. Available: <https://journal-home.s3.ap-northeast-2.amazonaws.com/site/icisc2023/ICISC+2023>
- [3] “Emerging Research in Electronics, Computer Science and Technology” ,Proceedings of International Conference, ICERECT2012.Available: <https://link.springer.com/book/10.1007/978-81-322-1157-0>.
- [4] European IT Certification Institute (EITCI), “European IT Certification Curriculum Self-Learning Preparatory Materials,” 2023. [Online]. Available: <https://eitca.org/wp-content/uploads/materials/eitcaexammateite-qi-qif.pdf>
- [5] “Quantum systems.” [Online]. Available: <https://fastercapital.com/keyword/quantum-system.html>
- [6] “A novel quantum-enhanced model cascading approach based on support vector machine in blood-brain barrier permeability prediction,” *Science Direct*, Apr. 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S2352492825008530?via>
- [7] “Spin Architectures for Investigating Quantum Phenomena: NMR and NV Centers in Diamond,” IISER Pune. [Online]. Available: <http://dr.iiserpune.ac.in:8080/xmlui/bitstream/handle/123456789/3830/ThesisVSArevised.pdf?sequence=1>
- [8] “Stabilization analysis for nonlinear interconnected system with memory-based coupled sampled-data control via quantum-inspired genetic algorithm,” *Science Direct*, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0020025524007710?via>
- [9] “Streaming execution for a quantum processing sys tem,” *Justia Patents*, 2019. [Online]. Available: <https://patents.justia.com/patent/11194573>
- [10] “Quantum neural networks for multimodal sentiment, emotion, and sarcasm analysis,” *Alexandria Engineering Journal*, *Science Direct*, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016825003229?via>
- [11] “Quantum Computing in Nuclear Physics,” Lund University, 2022. [Online]. Available: <https://lup.lub.lu.se/luur/download?fileId=9084880&func=download&FilerecordId=9084876>
- [12] “A Portable Quantum Programming Framework for Quantum Accelerators,” TU Delft, 2022. [Online]. Available: <https://repository.tudelft.nl/record/uuid:a7cd75cd-7e50-4c2a-bc51-a65445299d04>
- [13] “Quantum Computation,” *World Scientific Connect*. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/97898127947960004>
- [14] “Don’t care values in induction,” *Artificial Intelligence in Medicine*, *Science Direct*. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0933365796003570?via>
- [15] FPGA Compiler II /FPGA Express VHDL Reference Manual*, Synopsys, 1999. [Online]. Available: <http://ebook.pldworld.com/Semiconductors/ALTERA/Digital>
- [16] “Model of a programmable quantum processing unit based on a quantum transistor effect,”*AIP Conference Proceedings*,2018.[Online].Available: [https://pubs.aip.org/aip/acp/article/1936/1/020014/739345/Model of-a-programmable-quantum-processing-unit](https://pubs.aip.org/aip/acp/article/1936/1/020014/739345/Model-of-a-programmable-quantum-processing-unit)
- [17] “LAB PROGRAMS.docx,” *Course Hero*. [Online]. Available: <https://www.coursehero.com/file/137614552/8LAB-PROGRAMSdocx/>
- [18] “Constructing and Benchmarking Noise Models for Quantum Computing,” University of Ham burg, 2024. [Online].Available: <https://ediss.sub.uni-hamburg.de/bitstream/ediss/11440/1/dissertation.pdf>
- [19] “Digital Communication: Principles and System Modelling (Signals and Communication Technology),” *EPDF*. [Online]. Available: <https://epdf.pub/digital-communication-principles-and-system-modelling-signals-and-communication-.html>
- [20] “Digital Electronics Part I: Combinational and Sequential Logic.” [Online]. Available: <https://mafiadoc.com/digital-electronics-part-i-combinational-and-sequential-logic59c00ef51723dd7110dd5033.htm>