To the Graduate Council:
I am submitting herewith a dissertation written by Bhaskar Gaur entitled "Distributed and Approximate Quantum Circuits for Noise Resilience and Security." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Himanshu Thapliyal, Major Professor

We have read this dissertation
and recommend its acceptance:

Himanshu Thapliyal

Jiangbiao He

Rebekah Herrman

Travis S. Humble

Accepted for the Council:

Marieke Van Puymbroeck

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Distributed and Approximate Quantum Circuits for Noise Resilience and Security

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Bhaskar Gaur

August 2025

Dedicated to my beloved family, who taught me that education is the one gift that can never be taken away.

Acknowledgements

I want to sincerely thank my advisor, Dr. Himanshu Thapliyal, for believing in me and providing invaluable support. His guidance and encouragement have been crucial to my journey, shaping my research style, academic discipline, and integrity. The discussions and brainstorming sessions with him were not only intellectually inspiring but also key to my growth and development in academia.

I am grateful to my committee members, Dr. Jiangbiao He and Dr. Rebekah Herrman, for their constructive feedback and thoughtful discussions, which have significantly contributed to the development of this work. Special thanks to Dr. Travis S. Humble, who is both a committee member and collaborator, for sharing insights, resources, and extensive knowledge of quantum computing from ORNL.

To my family, whose steadfast faith in me served as the foundation for this achievement. I am thankful to my mother for nurturing my love of education and inspiring me to look beyond the immediate horizon. My Ph.D. would not have been possible without the support and care of my wife, Manu. Her constant companionship at home and in the department, while she was pursuing her Ph.D., gave me strength and stability to handle the demanding schedule of doctoral studies. Her feedback provided invaluable insights that strengthened both my analysis and writing.

To my fellow labmates at the Cy-QuBIT Lab, I will remember the countless discussions about research and the lighthearted moments we shared at UTK, as well as our volunteer work for GLSVLSI 2023 and ISVLSI 2024. I dedicate this thesis to all those who have helped and supported me in this journey.

Abstract

Quantum computing promises to solve complex computational problems that classical computers struggle to solve efficiently. Quantum arithmetic circuits serve as essential building blocks among the quantum circuits necessary to implement quantum algorithms like quantum approximate optimization, Harrow–Hassidim–Lloyd (HHL), and quantum cryptanalysis. However, the Noisy Intermediate Scale Quantum (NISQ) era quantum computers that are available today face significant challenges like noise, decoherence, and low quantum gate fidelity. They have limited qubits and connectivity, restricting the complexity and scalability of quantum circuits. Attackers can utilize these issues to introduce critical vulnerabilities, such as crosstalk, stuck at 0/1 attacks, and errors in state preparation and quantum gates to deduce quantum states or potentially inject faults.

This work focuses on resolving these challenges for quantum arithmetic circuits using alternate computing paradigms, such as distributed quantum computing and approximate computing, to create scalable quantum arithmetic circuits suitable for both NISQ and Fault Tolerant Quantum (FTQ) machines. This work presents distributed quantum arithmetic based on the Residue Number System (RNS), which distributes quantum addition or multiplication across multiple quantum modulo circuits, allowing for parallel execution over multiple jobs or quantum computers. RNS-based Distributed Quantum Addition (DQA) and multiplication offer multiple advantages like lower overall depth and scalability beyond the existing qubit capacity. To enable RNS based distributed quantum arithmetic, this work proposes: i) a

quantum carry-lookahead modulo (2^n - 1) adder that operates with logarithmic complexity, in contrast to existing solutions that have linear complexity; ii) quantum modulo (2^n + 1) adders; and iii) a quantum modulo (2^n + 1) multiplier. We also find DQA superior than non-distributed addition against noise and crosstalk attacks on ion-trap qubits. We then present five Approximate Quantum Adders (AQA) possessing constant depth, that demonstrate superior noise resilience compared to linear depth-based quantum full adders. Three of these AQA also demonstrate attack resilience against state preparation, gate and crosstalk attacks, although the exact adders fare better against the stuck at 0/1 attacks. In conclusion, this work lays the foundation for utilizing alternative computing paradigms to develop higher resilience against noise and attacks in quantum circuits.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Quantum computing is an emerging field that offers significant advantages over classical computing due to quantum algorithms like Shor's algorithm, quantum phase estimation, amplitude amplification, variational quantum algorithms, quantum approximate optimization algorithms, the Harrow–Hassidim–Lloyd (HHL) algorithm, and quantum convolutional neural networks [1, 2, 3, 4]. Quantum circuits form the foundation for constructing these algorithms, making them valuable for applications such as quantum cryptography, image processing, and various scientific computations, including variational quantum methods for nonlinear problems, simulation of many-body systems, solving linear systems of equations using the HHL algorithm, singular value thresholding, and quantum machine learning [1, 5, 4, 6, 7].

Quantum arithmetic circuits are critical components within these quantum circuits, enabling the implementation of quantum algorithms. Researchers have developed specialized libraries of quantum arithmetic circuits for quantum programming languages like Qiskit, Qualtran and the Microsoft Quantum Development Kit, which facilitate the design of resource-efficient quantum algorithm circuits [8, 9, 10]. Quantum adders play an essential role among quantum arithmetic circuits by forming building blocks for quantum subtraction, multiplication, and square root operation [11, 12, 13]. Another category of circuits made possible by quantum adders is quantum

modulo arithmetic circuits. These specialized circuits are designed specifically to support modulo operations. Modulo operations ensure that the results remain within the range of (0, modulus-1), thus maintaining closure under operations like addition, subtraction, and multiplication. Within these circuits, the quantum modulo adder is particularly useful, as it facilitates the construction of other quantum modulo arithmetic circuits, including those for multiplication and exponentiation [14, 15, 16]. Quantum arithmetic and quantum modulo arithmetic circuits are useful in applications including quantum image processing, quantum cryptanalysis, HHL algorithm, quantum convolution neural networks, and also as benchmark circuits in error-aware compilation [13, 17, 18, 19, 20, 21, 22].

1.1 Motivation

With the emergence of quantum computing, various designs for quantum full adders and quantum modulo adders have been proposed [23, 24, 16, 25, 26]. However, the quantum hardware available today, operating in the Noisy Intermediate Scale Quantum (NISQ) era, suffers from noise sources such as decoherence, crosstalk, State Preparation And Measurement (SPAM), and thermal relaxation, making them unreliable [27, 28, 29]. Frequent calibration is required to maintain gate fidelity, but limited or non-availability of error correction leads to accumulation of errors due to prolonged operations for high depth quantum addition circuits [30, 31]. As a result, we focus on addressing several key challenges faced by quantum addition in NISQ quantum computers, including:

1. **Quantum Resource Consumption:** The resources consumed by quantum circuits are typically measured in terms of quantum gate count, circuit depth, the number of qubits, and the count of ancillae. The most resource-intensive operations are two-qubit quantum gates, such as CNOT and Toffoli gates. Depth is defined as the number of layers of gates in the circuit that can be

executed sequentially [32]. Quantum circuits often require ancillary qubits (or ancillae) to provide constant inputs, which can later be utilized to hold intermediate computation steps or the final output. Designs that utilize fewer qubits and ancillae are generally preferred in NISQ and Fault-Tolerant Quantum Computing (FTQC), as they allow for the construction of wider circuits capable of handling larger data sizes. Similarly, circuits with fewer gates and lower depth indicate lower algorithmic complexity, leading to faster execution and lower noise and decoherence issues [33].

2. **Noise-Resilience:** The primary source of error in NISQ-era quantum computers is noise, which is dependent on multiple factors [31]. Quantum gates accumulate errors due to noise, increasing with higher depth of the quantum circuits. Idling qubits face damping and are affected by operations on neighboring qubits [34]. NISQ qubits also face decoherence issues when the quantum circuit execution time exceeds a certain threshold, resulting in quantum state decay and loss of entanglement [35]. Hence, the focus is on achieving high-fidelity operations by developing quantum circuits with shallow circuit depths and fewer qubits and gates. As shown in Figure 1.1, existing quantum adders mainly use the ripple carry mechanism, where the final carry depends on a long carry path, increasing both circuit depth and gate count, which results in a proportional rise in noise [23, 25]. This design leads to qubit idling, causing decoherence leading to loss of information. [33].

3. **Security and Privacy:** Quantum computers promise considerable advantages over classical computing, but their susceptibility to various attacks, such as fault injection, privacy, and scheduling, poses significant security challenges [36, 37, 38]. Since quantum computers are usually accessed remotely, scheduler attacks can misreport error rates, resulting in the allotment of inferior hardware or qubits with worse noise characteristics [39]. Furthermore, the quantum circuits from the user pass to the hardware via compilation and optimization,

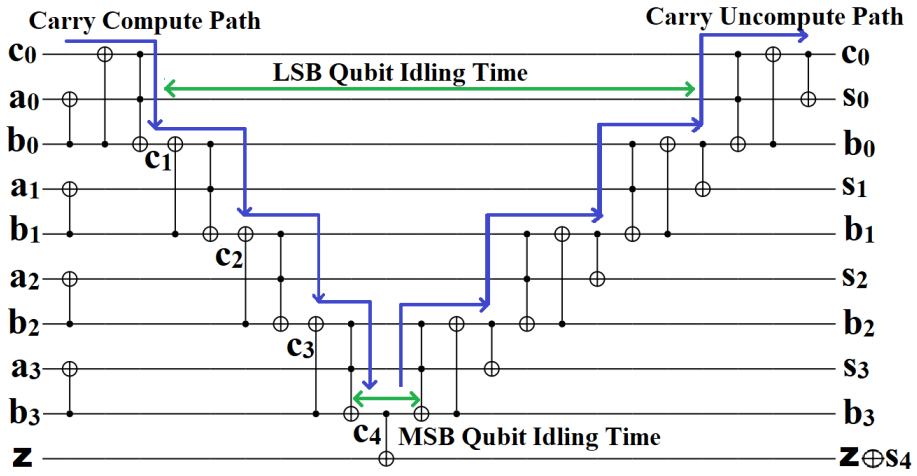


Figure 1.1: Exact quantum adder by Cuccaro et al [23]. The inputs are in range a_0 to a_3 and b_0 to b_3 while S_0 to S_4 represent Sum bits. The Carry generation path extends from c_0 to c_4, creating higher depth and longer carrypath, increasing the noise susceptibility of carryout. LSB qubits spend much more time idling than MSB, making lower qubits of Sum vulnerable to noise [33].

significantly expanding the attack surface and leaving the circuit vulnerable to cloning, counterfeiting, and reverse engineering [40]. Susceptibility to quantum noise further complicates the situation, with common noise sources such as crosstalk causing fault injection (denial of service) in a multi-user scenario [41, 42, 43].

1.2 Overview of Solution Strategies

In this section we provide a high-level overview of the predominant approaches that we have explored in addressing the challenges discussed in Section 1.1. The solution strategies are:

- **Distributed Computing:** Distributed computing involves dividing a computational task across multiple interconnected processing units (nodes), which collaborate toward the common goal of solving the task [44]. Modern cluster based high-performance computing and cloud computing are some examples of distributed computing that achieve difficult to solve problems like weather forecasting, material science simulations, and medical drug discovery [45, 46, 47]. Distributed computing systems are scalable, optimally utilize resources, and tolerate faults better [48]. In quantum computing, Distributed Quantum Computing (DQC) extends this paradigm to quantum systems, where parts of a quantum circuit or application are executed on multiple Quantum Processing Units (QPUs) [49]. DQC aims to overcome hardware limitations like the number of qubits or coherence time by parallelizing quantum computations, thereby enabling larger and more complex quantum circuits to be executed across separate quantum devices. To implement DQC in quantum arithmetic circuits, we utilize an encoding scheme called Residue Number System (RNS). RNS encodes integers as residues relative to a set of co-prime moduli. Arithmetic operations in RNS, such as addition and multiplication, are performed in parallel

for each residue without carry propagation[17]. In this work, we investigate the ways in which distributed quantum computing can:

1. Enable parallel execution of independent quantum modulo arithmetic circuits, optimizing execution time and available computational resources.
2. Reduce circuit depth and gates per circuit to improve noise resilience, mitigating the effects of noise and decoherence critical in NISQ-era quantum computers.
3. Overcome qubit limitations by splitting quantum arithmetic circuits that require more qubits than available on a single device to be executed across multiple QPUs or jobs on a single QPU.
4. Improve the security of quantum arithmetic circuits by leveraging the non-dependence of RNS based quantum modulo arithmetic circuits on each other.

Potential applications for RNS based DQC include quantum cryptography, quantum cryptanalysis, quantum signal, and image Processing.

- **Approximate Computing:** Approximate computing is a computational paradigm that allows a trade-off between accuracy and computational complexity, leveraging the error resilience of certain applications. It intentionally permits predetermined inaccuracies in computation to achieve significant gains in energy efficiency and performance [50]. In classical computing, approximate circuits are found particularly relevant for applications where precise results are not critical, such as signal/image processing and machine learning [51, 52]. Similar to classical computing, quantum computing requires resource conservation, but it emphasizes noise resilience rather than energy efficiency, as in classical circuits. Current quantum arithmetic circuits have complex architectures, significant depth, and high computational complexity. In the Noisy Intermediate Scale Quantum (NISQ) era, quantum computers are limited

by noise and the short coherence times of qubits, making exact quantum arithmetic circuits impractical. Due to these reasons, we investigate in this work if approximate computing can:

1. Reduce circuit depth and gate count, which are key factors influencing noise and output fidelity in quantum circuits.
2. Reduce the complexity of quantum circuits, increasing their scalability.
3. Achieve high fidelity output even on NISQ devices, making quantum computing more viable for real-world applications on current quantum hardware.
4. Increase resilience against vulnerabilities introduced by attacks

1.3 Contributions

To address the challenges faced by the quantum arithmetic circuits, we propose the following:

- **Quantum Carry-Lookahead Modulo (2^n - 1) Addition (QCLMA) Circuit:** We have designed QCLMA to have an O(log n) depth compared to the O(n) depth of existing works. The proposed QCLMA offers superior noise resilience due to its improved carry path and reduced depth, demonstrated by running experiments on a 27-qubit quantum computer IBM Cairo [53].
- **Quantum Modulo (2^n + 1) Addition (QMA) Circuits:** We have developed QMA circuits for modulo (2^n + 1) addition, divided into two categories based on their output type. The first category features quantum adders that perform (a + b + 1) modulo (2^n + 1) addition using the regular number system. For these adders, we introduce a systematic optimization methodology that utilizes design enhancements and zero reset operations to minimize the depth and count of ancilla qubits in QMA, thereby reducing errors

on the IBM Washington, a 127-qubit quantum computer. The second type uses the diminished-1 number system for (a + b) modulo (2^n + 1) addition, offering greater usability due to the efficient handling of zero inputs [54, 55].

- **Residue Number System (RNS) based Distributed Quantum Arithmetic:** This work focuses on achieving scalability for quantum addition through a distributed quantum computing approach. We utilize the Residue Number System (RNS) to distribute quantum addition across multiple circuits that can be run in parallel across multiple quantum computers or jobs without any carry dependencies. We chose a set of three relatively prime moduli: (2^n-1, 2^n, and 2^n+1), that can approximate the range of a quantum full adder for input size '3n' while having lower overall depth. The work demonstrates higher noise resilience of RNS-based distributed quantum addition over its non-distributed counterpart by conducting simulations based on Quantinuum's H1 20-qubit ion trap-based quantum computer [55]. We have also estimated the quantum resource usage for RNS based distributed quantum multiplication and have established up to 46.018% lower Toffoli depth, and reduction in T gates of 34.483% to 86.25% for 6 to 16 qubit sized output. We also proposed a design of Quantum Diminished-1 Modulo (2^n+1) Multiplier, an essential component needed for RNS based distributed quantum multiplication [56].

- **Crosstalk Attack Resilience of Distributed Quantum Addition:** Crosstalk is an inherent challenge in quantum computing due to unwanted interactions between qubits. We investigated the effect of crosstalk noise as a fault injection attack vector on a cloud-based quantum computer in a multi-user setting. We first customized the current crosstalk attacks to make them more effective for ion-trap qubits, and then studied if RNS based distributed quantum addition performs better than existing non-distributed quantum addition [57].

- **Approximate Quantum Addition Circuits:** We have designed five approximate quantum adders (three proposed adders with carryout and two

proposed adders without carryout) designed using approximate computing having constant O(1) depth and reduced depth and quantum gate cost compared to existing works. The superior noise resilience of the proposed quantum approximate adders is demonstrated by conducting simulations based on IBM Qiskit noise models for thermal, depolarizing, phase, amplitude, and bitflip noise sources [33]. We also find the three proposed approximate quantum adders with carryout resilient against crosstalk, state preparation and gate based quantum attacks. However, we also find them worse than exact adders againt stuck at 0/1 attacks.

1.4 Outline of Thesis

Chapter 2 provides background on the essential fundamentals of quantum computing, quantum gates and circuits, and quantum arithmetic circuits. Chapter 3 proposes Quantum Carry-Lookahead Modulo (2^n - 1) Addition (QCLMA) circuit having logarithmic depth complexity compared to the linear complexity of earlier works. Chapter 4 proposes novel optimized designs of Quantum Modulo (2^n + 1) Addition (QMA) Circuits. Chapter 5 discusses achieving enhanced noise resilience and scalability using a Residue Number System (RNS) based distributed quantum addition and multiplication. Chapter 6 introduces the impact of security vulnerabilities such as crosstalk attacks and evaluates the crosstalk attack resilience of distributed quantum addition. Chapter 7 covers the use of approximate computing in reducing depth and increasing the noise and attack resilience of approximate quantum adders. Chapter 8 briefly summarizes the overall ideas discussed here and lays down the future research possibilities.

Chapter 2

Background

2.1 Quantum Computing Fundamentals

2.1.1 Qubit

Qubit is the fundamental unit of quantum information, comparable to the classical bit but with unique quantum properties. Like the classical bit, it has two mutually exclusive states, state 0 and state 1, but the qubit obeys the laws of quantum mechanics. In linear algebra, a qubit is represented using two-dimensional vectors, which together are called the computational basis. State 0 is also known as the ground state as it is the lowest energy state of a quantum mechanical system, and state 1 is called an excited state, possessing a higher energy. Hence, we need a quantum mechanical system with atleast two energy levels to implement the two states [58, 59]. For instance, the energy levels of an electron can also be used to implement a qubit. Using bra-ket notation, the qubit states are represented as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.1}$$

2.1.2 Quantum States

A quantum state is formally defined as a vector within a Hilbert space, which is a complete vector space equipped with an inner product. This mathematical framework allows for the representation of quantum states as high-dimensional vectors, similar to vectors in Euclidean space, but extended to accommodate quantum mechanics. The inner product allows for the definition of angles and lengths, enabling the measurement of quantum probabilities. A single qubits state can be described in two-dimensional Hilbert space.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.2}$$

where $|\alpha|^2 + |\beta|^2 = 1$

The inner product $\langle\phi|\psi\rangle$ is defined as:

$$\langle\phi|\psi\rangle = \sum_i \phi_i^* \psi_i \tag{2.3}$$

Here, ϕ_i^* is the complex conjugate of the component of $|\phi\rangle$ in the i^{th} basis direction, and ψ_i is the corresponding component of $|\psi\rangle$. We will see in the Section 2.1.3 that the inner product is crucial for qubit properties because it:

- Measures the overlap between quantum states.
- Establishes orthogonality and independence among states.
- Forms the basis for determining the probabilities of measurement results.

According to the convention, the quantum state labeled ψ is represented using "ket" notation as $|\psi\rangle$, while the corresponding dual vector is denoted in "bra" notation as $\langle\psi|$. The inner product of these two vectors is expressed as $\langle\psi|\psi\rangle$ and is normalized to one.

$$\langle\psi|\psi\rangle = i\sum_{i}|\psi_i|^2 = 1 \tag{2.4}$$

This ensures that the total probability of finding the system in one of the basis states is 1. Also, two states $|\phi\rangle$ and $|\psi\rangle$ are orthogonal or independent if $\langle\phi|\psi\rangle = 0$.

2.1.3 Qubit Properties

The properties that a qubit possesses in a quantum mechanical system are:

- **Superposition**: A qubit can be in the state 0 as well as 1, as quantum mechanics allows the qubit to be in any arbitrary combination of states. We describe this combination as superposition, using probability amplitudes, denoted as α and β. This ability to be in multiple states at once allows quantum computers to process a vast number of possibilities simultaneously.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.5}$$

Bloch Sphere, a geometric representation of a qubit's state that provides insight into superposition and other quantum properties, is shown in Figure 2.1. It maps a qubit's state to a point on a unit sphere, with $|0\rangle$ and $|1\rangle$ at the poles. Any qubit can be described using spherical coordinates on the bloch sphere:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle\,, \quad 0 \le \theta \le \pi, \quad 0 \le \phi < 2\pi \tag{2.6}$$

The coordinates (θ, ϕ) correspond to the angles of the state vector, enabling visualization of superposition and phase differences. The surface of the bloch sphere is always a unit distance away from the center, denoting the overall probability of the qubit $|\psi\rangle$. This is another way of denoting that qubit always satisfies:

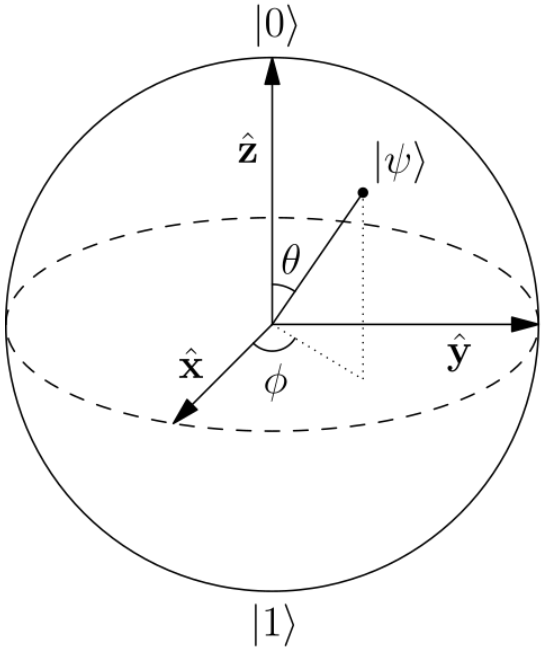


Figure 2.1: Block sphere: A qubit can take any position on the surface of the sphere. The basis states $|0\rangle$ and $|1\rangle$ are aligned with the z-axis.

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.7}$$

- **Entanglement**: Qubits can be entangled, meaning the state of one qubit depends on the state of another, no matter the distance between them. Entanglement occurs when the state of a quantum system cannot be factored into product states. For example, if a two-qubit quantum state $|\psi\rangle$ cannot be expressed as $|\psi\rangle \otimes |\phi\rangle$ for any valid selections of $|\psi\rangle$ and $|\phi\rangle$, then $|\psi\rangle$ is considered to be entangled. A distinct outcome of entanglement is that one can gather information about the entangled qubits without needing to evaluate all the qubits of a quantum system, which is essential for quantum communication and computation. Once two qubits are entangled, measuring one of the qubits allows us to instantly ascertain the values of both qubits, irrespective of the separation between them. Quantum entanglement plays a significant role in quantum computing and the implementation of multi-qubit quantum gates (explained later). When paired with superposition, quantum circuits can simultaneously produce all possible computational outcomes. The measurement process will reduce this continuum to the most likely solution. Additionally, entanglement enables the creation of unique quantum circuits, such as the Quantum Fourier Transform (QFT), which offers a resource-efficient alternative to the discrete fourier transform and is a key component in quantum algorithms [60].

- **Measurement**: Quantum measurement is the process of extracting information about a quantum system by interacting with it in a controlled manner. Unlike classical measurement, quantum measurement is probabilistic and fundamentally alters the state of the system due to wavefunction collapse [61]. A quantum system's state $|\psi\rangle$ is measured with respect to an orthonormal basis

$|b_1\rangle, |b_2\rangle, ..$ of a Hilbert space. Before measurement, Equation 2.8 denotes that the system is in a superposition:

$$|\psi\rangle = \sum_i c_i |b_i\rangle, \quad \text{where } c_i \text{ are complex coefficients (amplitudes)} \tag{2.8}$$

Measurement collapses the state into one of the basis states $|b_i\rangle$, chosen probabilistically based on the amplitude $|c_i\rangle$. The probability of measuring the system in the state $|b_i\rangle$ is given by:

$$P(b_i) = |\langle b_i|\psi\rangle|^2 = |c_i|^2 \tag{2.9}$$

Equation 2.9 is also known as **Born's Rule** [62]. After measurement, the system's state becomes $|\psi'\rangle = |b_i\rangle$. Measurement is non-reversible due to wavefunction collapse. Post-measurement, information about the original superposition is lost, altering the quantum system fundamentally.

2.1.4 Quantum Theorems

The quantum theorems are fundamental results in quantum mechanics, emphasizing the uniqueness and preservation of quantum information. Some of the important quantum theorems are:

- **No-Cloning Theorem**: The no-cloning theorem states that it is not possible to create an exact duplicate of an arbitrary unknown quantum state [63]. This is due to the linearity of quantum mechanics, as attempting to copy a quantum state in superposition would require a process violating the unitary evolution of quantum systems. One significant consequence of the theorem is the limitation it imposes on quantum error correction methods by making classical error correction where backup copies of data are used to address errors, which is

impossible in quantum systems. As a result, quantum error correction relies on less intuitive, specialized techniques to mitigate errors during computation [64]. In quantum cryptography, the no-cloning theorem provides a safety mechanism by not allowing the creation of exact copies of transmitted quantum keys, hence protecting against eavesdropping . However, approximate cloning is not prohibited, creating a nuanced challenge against probabilistic attempts to estimate approximate copies of a quantum state [65].

- **No-Deleting Theorem**: The no-deleting theorem is a time-reversed dual of the no-cloning theorem which states that, given two copies of an arbitrary quantum state, deleting one of the copies is not possible. It imposes another limitation on quantum information due to the linearity of quantum mechanics. Together with the no-cloning theorem, it highlights the conservation of quantum information [66, 67].

2.2 Quantum Gates

Quantum gates are quantum operations that act upon an input quantum state and perform modifications to produce an output quantum state [58]. They are the basic building blocks in quantum computing, comparable to boolean logic gates in classical computing. The evolution of an isolated quantum system over time t is formally described by a unitary transformation that takes an initial state $|\psi(0)\rangle$ to a final state $|\psi(t)\rangle = U(|\psi(0)\rangle)$. This demonstrates that a quantum logic gate acting on an isolated quantum computer will transform that state unitarily until an observation is made. Therefore, quantum logic gates are mathematically represented by unitary matrices, ensuring that their actions are always logically reversible [59]. A matrix, U, is unitary if and only if its inverse equals its conjugate transpose, i.e., $U^{-1} = U^{\dagger}$. Table 2.1 discusses the properties of quantum gates:

Table 2.1: Properties of Quantum Gates [59].

Property	**Description**
U^{-1} and $U^{\dagger}$ are unitary	Reverse operation for a quantum gate always exists.
$U^{\dagger}U = UU^{\dagger} = I$	Quantum gate is logically reversible.
$\lvert \det(U) \rvert = 1$	Quantum gate is normalized.
$\det(U) = \pm 1$ or $\pm i$	Elements of a quantum gate's unitary matrix can be complex numbers.
For a fixed column or row i: $\sum_{i=1}^{2^n} \lvert U_{ij} \rvert^2 = 1$	If input is a properly normalized quantum state, output is also a properly normalized quantum state.

2.2.1 Types of Quantum Gates

Tables 2.2 discusses the different types of single qubit quantum gates, along with their 2x2 unitary matrices. A qubit's state is represented as a superposition of the basis states $|0\rangle$ and $|1\rangle$, and single qubit gates operate on this state through rotations, flips, and phase shifts. Among these, the **rotation gates** (R_X, R_Y, R_Z) provide a continuous parameterization to rotate the qubit's state vector around the X, Y, and Z axes of the bloch sphere by an arbitrary angle θ. These gates are versatile and can be combined to construct other gates, such as the pauli gates (X, Y, Z), by setting $\theta = \pi$ radians. The pauli-x gate which flips the magnitude of quantum state by 180-degree, from $|0\rangle$ to $|1\rangle$ or vice-versa, is also known as NOT gate. The Phase (S) and T gates introduce specific phase shifts that are vital for quantum algorithms, and can be created using $R_Z(\theta)$ gate by tuning the rotation angle. Among the single qubit gates, Hadamard gate plays a crucial role in quantum computing. As shown in Equation 2.10, the Hadamard gate (H) creates an equal superposition of basis states, with or without a phase difference. When used in multi-qubit systems, the Hadamard gate can be combined with controlled operations (like CNOT) to produce entangled states.

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.10}$$

Table 2.3 shows the quantum gates that operate on two or more qubits. Multi-qubit gates are essential in quantum computing as they enable interactions between qubits by facilitating entanglement, state swapping, and other complex operations. In the controlled gates, one or more qubits acts as a control for an operation. The most widely used controlled gate is CNOT gate that performs NOT operation only when the control qubit is $|1\rangle$, enabling XOR like operation. Toffoli or CCNOT gate is also similar, which performs NOT only when both control qubits are $|1\rangle$, executing AND/NAND like operations. Apart from CNOT, other multi qubit quantum gates can be constructed using universal quantum gates.

Table 2.2: One Qubit Quantum Gates

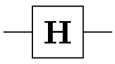
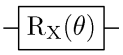
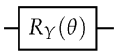
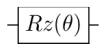
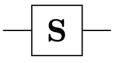
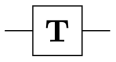
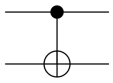
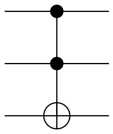
Operator	Gate	Description	Unitary
Hadamard		Creates a superposition state by equally distributing the probability amplitude.	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
$\mathbf{R_x}(\theta)$		Rotation-X gate rotates a qubit by angle θ around x-axis.	$\begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right)\\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix}$
$\mathbf{R_y}(\theta)$		Rotation-Y gate rotates a qubit by angle θ around y-axis.	$\begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix}$
$\mathbf{R_z}(\theta)$		Rotation-Z gate rotates a qubit by angle θ around z-axis.	$\begin{bmatrix}e^{-i\frac{\theta}{2}} & 0\\ 0 & e^{i\frac{\theta}{2}}\end{bmatrix}$
S		S or phase gate introduces phase change of 90°around z-axis.	$\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$
T		T gate denotes a $\pi/4$ phase shift around z-axis. Also, $S = T^2$.	$\begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix}$
NOT		NOT or pauli-x gate flips the state of qubit, and is equivalent to $R_x(\pi)$.	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$

Table 2.3: Multi-Qubit Quantum Gates

Operator	Gate	Description	Unitary
CNOT		Controlled X gate enables XOR operation by flipping amplitude on target qubit if control is $\|1\rangle$.	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
CZ		Controlled Z gate applies a phase flip only when control qubit is $\|1\rangle$.	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
Swap		Swaps the state of two qubits using three alternate CNOT gates.	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli		Also known as the Controlled-Controlled NOT (CCNOT). Flips the target qubit only if both control qubits are $\|1\rangle$.	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Universal Quantum Gates is a set of gates that can perform any possible operation on quantum computers, by expressing the unitary of the operation using a finite sequence of gates from the set. Universal quantum gate sets typically include few single qubit gates with a two qubit gate like CNOT or CZ. Clifford+T gate set is one such gate set, which includes CNOT, H, S, and T gates, and is efficiently simulated by classical computers [68]. Quantum computing vendors include CNOT along with rotation gates as a custom set for their machines [69, 70].

2.3 Quantum Circuits

Quantum circuit is a paradigm that helps understand and optimize quantum operations and algorithms. A quantum circuit consists of the initialization of qubits with state preparation, followed by a sequence of quantum gates, and finally, output observation in the form of measurements [58]. A quantum circuit helps visually represent a compute-on-memory type of system since the computation and initial input are stored on the qubits. Unlike classical circuits, the horizontal rails are not physical connections but rather qubits themselves. As demonstrated in Figure 2.2 The left corner represents the input states, and the right corner of the horizontal lines represents the output states. Time (and order of execution of quantum gates) progresses from left to right.

The overall unitary of a quantum circuit can be computed by conducting operations on unitary matrices of the underlying quantum gates. As shown in Figure 2.3(a), the overall effect of gates applied sequentially is determined by taking their dot product. Similarly Figure 2.3(b) demonstrates that quantum gates acting on separate sets of qubits can be executed simultaneously or in any order without affecting the overall operation.

Like quantum gates, quantum circuits are also inherently reversible due to their unitary nature. Reversible computing is a computation model where the process can be reversed without loss of information, contrasting with classical computing, which

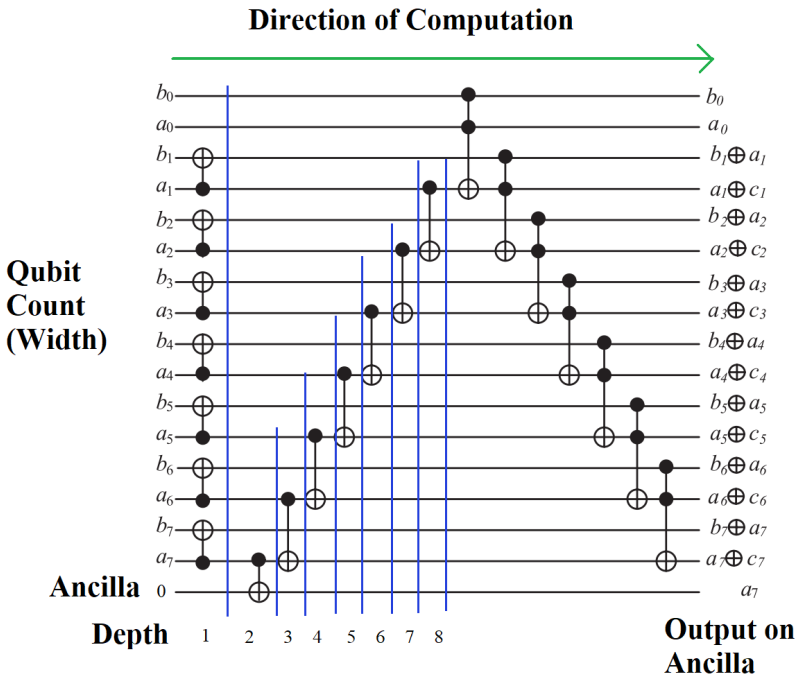


Figure 2.2: Understanding a quantum circuit diagram.

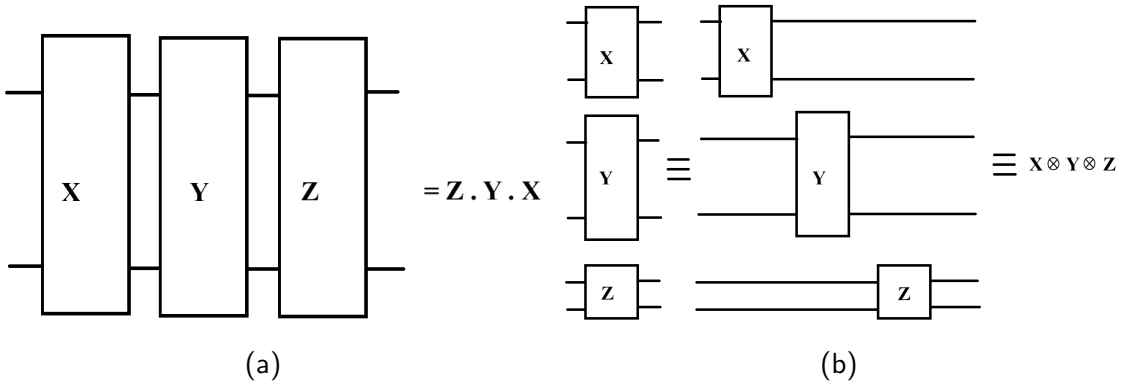


Figure 2.3: Rules to compute the unitary matrix of a quantum circuit when underlying quantum gates are in: (a) Series (b) Parallel [59].

irreversibly loses information via entropy. Due to reversibility, there exists a one-to-one mapping between inputs and outputs, and all inputs and intermediate states can be traced back to a quantum circuit without information loss. Quantum circuits also share other properties shown in Section 2.2, such as normalized output with quantum gates. Quantum circuits can help achieve functions of higher complexity, such as quantum arithmetic circuits like quantum adders and multipliers, and quantum algorithms such as quantum fourier transform.

2.3.1 Quantum Circuit Cost

Quantum circuits use spare qubits known as ancillae qubits that act as temporary storage to hold intermediate computational results during multi-step quantum operations. This is particularly useful when computations need to be performed in stages to build up to the final result. The undesirable states produced by quantum circuits are called garbage outputs. In order to free qubits from garbage outputs, ancillae qubits are sometimes used to restore circuits to their initial states. In order to compare quantum circuits, several metrics are utilized to measure their resource consumption, such as:

- **Qubit count (Width)**: The total number of qubits, including ancillae, required by the circuit.
- **Gate count (Size)**: The total number of quantum gates utilized by the quantum circuit.
- **Depth**: The number of gate layers in the circuit that can be executed sequentially.
- **Ancillae qubits**: The number of ancillae qubits used by the circuit.
- **Garbage outputs**: Outputs that do not contribute to the final output and must be removed through reverse computation.

Circuits are considered efficient if their resource requirements grow linearly or in a polynomial manner with the number of qubits, while exponential growth indicates inefficiency. As CNOT and Toffoli gates are more resource intensive than single-qubit gates, their depth and gate counts are measured separately for better comparison. In many quantum circuits, ancillae provide a way to optimize the overall resource cost by conducting a trade-off between the circuit depth, gate count, and circuit width (qubit count), improving performance in noisy quantum systems.

2.3.2 Dynamic Circuit Features

Dynamic Circuits are a class of quantum circuits in which feedback and adaptive operations occur during computation. These operations allow real-time decision-making within the circuit based on intermediate measurements of qubits. Unlike static quantum circuits that have a fixed sequence of operations that are determined before execution, dynamic circuits use features such as mid-circuit measurement and zero reset, leading to flexible and efficient quantum algorithms [71].

Mid-circuit measurement refers to the measuring of a qubit's state during the execution of a quantum circuit rather than at the end of the computation. Zero reset operation resets a qubit to the $|0\rangle$ state, allowing the same qubit to be reused for further computation within the same circuit without undergoing reversibility to the initial state. This feature is especially valuable for optimizing qubit usage in quantum devices with limited qubit availability [72]. While dynamic circuits enable more sophisticated and efficient algorithms by allowing feedback loops, they are more complex to design and simulate.

2.4 Quantum Noise Sources

Quantum noise represents the various disturbances that affect quantum systems, leading to errors in computation or loss of information. It arises from various

environmental, operational, and hardware imperfections in quantum machines, disrupting the fragile quantum states and leading to errors in quantum information processing. Table 2.4 outlines the types of quantum noise, their causes and impacts.

The mitigation strategies for the noise sources also vary widely like their origins. A common technique employed by vendors is to frequently reset qubits, especially for noise sources that are uncorrelated and often have hardware origins, like thermal, depolarizing, amplitude and phase damping [73]. For gate noise that is accumulated due to prolonged operations in high depth quantum circuits, error correction techniques are better suited [74]. Development of better qubit initialization and detection mechanisms can help in State Preparation And Measurement (SPAM) and readout noise.

However, these methods cannot be used by quantum circuit designers. Design techniques such as lowering of circuit depth, use of architectures that help trade-off between circuit width and depth, saving quantum resources such as gates, and effective use of placement strategies are some of the methods which we will discuss in this work. Reduction of depth and quantum gates helps reduce all noise sources except SPAM and readout noise as they are independent of quantum circuit being executed [75]

2.5 Quantum Computing Hardware

Quantum computing hardware refers to the physical systems used to implement and manipulate qubits and leverage quantum mechanical phenomena such as superposition and entanglement to perform computations. The DiVincenzo Criteria defines the fundamental requirements for building a functional quantum computer, such as the ability to initialize qubits reliably into a known and consistent state, typically $|0\rangle$, and long coherence times to maintain quantum states [76]. A universal set of quantum gates and high-fidelity qubit measurement methods are the other requirements. In this section, two major types of quantum computers are described.

Table 2.4: Quantum noise sources

Noise Type	**Cause**	**Impact**
Thermal	Interaction with thermal environment.	Qubits relax to lower energy states (T1 decay) leading to decoherence [58].
Depolarizing	Random noise that equalizes all qubit states.	Random Pauli operations on all 3 axes causing collapse to mixed states [28].
Amplitude Damping	Spontaneous emission of energy.	Qubits transition to state $\|0\rangle$ [27].
Phase Damping	Environmental interactions.	Random rotation along z-axis, leading to loss of information without energy loss [78].
Bitflip Noise	When error crosses threshold.	Magnitude flip from $\|0\rangle$ to $\|1\rangle$ or vice versa [79].
SPAM	Calibration and readout mechanisms.	Produces incorrect initializations and readouts [29].
Readout Noise	Detector inaccuracies.	Unreliable results independent of circuit depth [29].
Crosstalk	Unintended coupling among nearby qubits or control signals.	Fault injection due to activity on nearby qubits [42].
Gate Noise	Calibration errors in control pulses.	Error accumulation with an increase in circuit depth [58].

2.5.1 Superconducting Qubits

Superconducting qubits are created using solid-state superconducting circuits consisting of Josephson junctions, capacitors, and inductors. They are designed to mimic the quantum behavior of microscopic two-level systems and are often referred to as artificial atoms. By leveraging the non-linearity of Josephson junctions, superconducting qubits exhibit anharmonic energy levels, which are essential for isolating two energy levels for quantum computation. These qubits operate by encoding quantum information in specific energy states of a superconducting circuit. They rely on manipulating the states of qubits using microwave pulses and external magnetic flux [77].

The single qubit gates are implemented using microwave pulses resonant with the qubit's energy level spacing. The phase and amplitude of the pulses control rotations in the Bloch sphere's X, Y, and Z axes. Multi-qubit gates on the other hand are achieved by tuning the coupling and frequencies of interacting qubits. They typically rely on capacitive or inductive coupling between qubits. Measurement of superconducting qubits is primarily performed via dispersive coupling to a microwave resonator. The qubit state modifies the resonator's frequency, and the measurement of the transmitted or reflected signal reveals the qubit state [80]. The main advantages of superconducting qubits are:

- Design Flexibility: Superconducting qubits allow for customization of parameters like energy levels and coupling strength by adjusting circuit elements.
- Scalability: They are fabricated using established semiconductor techniques, enabling the integration of multiple qubits on a chip.
- Fast Gate Operations: The qubit operations occur on nanosecond timescales, enabling high-speed computation and suitability for quantum machine learning applications.

The main disadvantage of superconducting qubits is that they are prone to decoherence, typically lasting only tens of microseconds. They also require operation at mK temperatures, making complex and expensive cryogenic systems necessary. This also increases their susceptibility to noise from environmental conditions such as external magnetic flux, charged particles from stray radiation, and temperature variations. Improvement of decoherence time in superconducting qubits is an active research area.

2.5.2 Ion-Trap Qubits

Ion-trap qubits are quantum bits realized using individual atomic ions trapped and controlled in electromagnetic potentials whose internal electronic states represent the computational basis states $|0\rangle$ and $|1\rangle$. The elements favored for creating ions are those found in groups II A and II B of the periodic table, such as Be, Mg, Ca, Sr, Ba, Cd, and Yb [81]. Ions are kept in vacuum traps, usually among the two types, Paul and Penning traps. The Paul trap uses rapidly oscillating electric field, and the Penning trap uses perpendicular electrostatic and magnetostatic fields to hold the ions in place. The single qubit gates are implemented using microwave pulses or laser beams to drive the transition between the energy levels and coherent rotations. Multi-qubit gates are implemented by coupling the ions in the same trap using their shared states. The process of measurement or state detection relies on fluorescence, in which a laser is tuned to a cycling transition of one state (for e.g. $|1\rangle$), which scatters photons, making the ion appear bright. The other state (here $|0\rangle$) does not scatter light, appearing dark, allowing for high-fidelity readout [82]. The main advantages of ion-trap qubits are:

- Long Coherence Times: Ion-trap qubits can maintain coherence for an extended period.

- High Fidelity: Both single and two-qubit gate operations have fidelities exceeding 99.9%, due to the isolation provided by ultra-high vacuum environments and the precise control of electromagnetic fields.
- Identical Qubits: All ions of a given species are fundamentally identical, avoiding fabrication variability.
- Connectivity: A significant advantage over superconducting qubits is the ability to have any-to-any connection among ion-trap qubits.

However, ion-trap qubits suffer from two main disadvantages compared to superconducting qubits. First, their slower operating speed can limit their usability across the spectrum of quantum computing applications. Secondly, they face scaling challenges as increasing the number of ions while maintaining high fidelity is complicated within the same vacuum chamber or tube. Also, these systems require ultra-high vacuum in addition to cryogenic environments for optimal performance.

2.6 Quantum Addition Circuits

Quantum adder is a quantum circuit designed to add inputs using reversible logic gates, ensuring that the operations are reversible and align with the principles of quantum computation. A quantum adder typically takes two n qubit inputs, a and b, and at the end of the computation, one of the inputs is transformed to (a + b) of n+1 qubit size, while the other remains unchanged. Most quantum addition circuits can be described using CNOT and Toffoli gates. Equations 2.11 and 2.12 define the formulas for expressing sum (s_i) and carry (c_i) in quantum adder, where c_{input} and c_n are the input and output carry respectively.

$$s_i = \begin{cases} a_i \oplus b_i \oplus c_i & \text{if } 0 \leq i \leq n-1, \\ c_n & \text{if } i = n. \end{cases} \tag{2.11}$$

$$c_i = \begin{cases} c_{input} & \text{if } i = 0, \\ a_{i-1}b_{i-1} \oplus b_{i-1}c_{i-1} \oplus c_{i-1}a_{i-1} & \text{if } 1 \leq i \leq n. \end{cases} \tag{2.12}$$

The primary constraints while designing a quantum addition circuit are:

1. Minimization of resource consumption like qubits, gate count, and gate depth.
2. Use of auxiliary inputs like input carry and ancillae.
3. Output carry calculation technique.
4. Impact of noise sources such as decoherence, SPAM, phase, and amplitude damping.

Considering the above factors, this section discusses three different types of quantum addition circuits that use diverse architectures.

2.6.1 Quantum Ripple Carry Adder

The quantum adder proposed by Cuccaro et al. is a linear depth adder using a ripple-carry approach [23]. Figure 2.4 shows that the design propagates the carry sequentially through the Least Significant Bit (LSB) to the Most Significant Bit (MSB) and later erases the carry in reverse order from MSB to LSB. This design is superior to previous quantum ripple carry adders as it uses only one ancillae qubit in order to save the carryout while being simpler in implementation. However, it also suffers from high latency due to linear carry propagation, which peaks at LSB, making it vulnerable to noise-related errors [33].

2.6.2 Quantum Adder without Input Carry

This adder proposed by Thapliyal et al. also uses ripple carry mechanism but it eliminates the need for an initial input carry qubit [25]. Figure 2.5 shows it's double-V structure that reduces gate count and qubit idling between gates and results in better

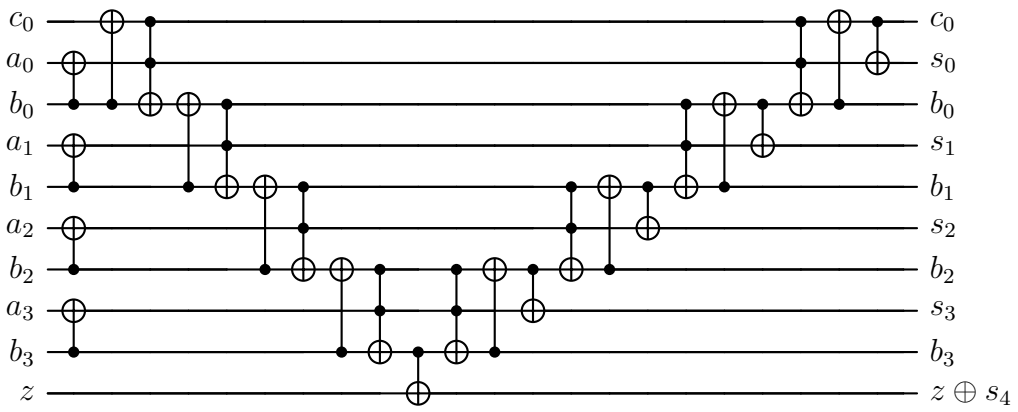


Figure 2.4: Quantum ripple carry adder in 4-qubit configuration. The carry computation is performed from LSB to MSB. The final carryout is stored in ancillae, and the carry is uncomputed from MSB to LSB [23].

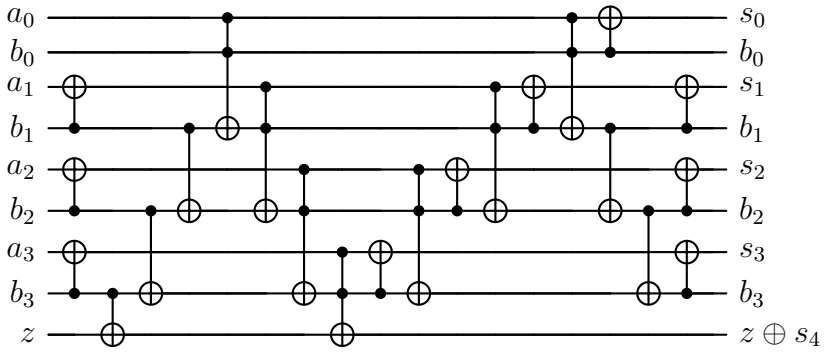


Figure 2.5: Quantum adder without input carry in 4-qubit configuration. The carry computation is performed using ripple carry mechanism but using a double-V structure that saves resources and causes lesser qubit idling [25].

noise-resilience compared to the quantum ripple carry adder discussed in Section 2.6.1 [33]. However, it has a greater design complexity as well.

2.6.3 Quantum Carry-Lookahead Adder

The quantum adder proposed by Draper et al. is a logarithmic depth adder that computes carry bits in a tree-like structure for rapid propagation, inspired by classical carry-lookahead adders [24]. It has a significantly reduced depth O(log n) achieved by parallel computation of carry bits reducing the need for sequential operations. There are two versions, out-of-place which generates sum on ancillae, and in-place which generates sum on one of the inputs. While it can execute faster than previously discussed quantum adders, its main disadvantage is the use of additional ancillae qubits to store intermediate results increasing the overall qubit count.

2.6.4 Resource Comparison

The Table 2.5 shows the comparison of quantum resource usage for the three quantum adders discussed in this Section. The no input carry based quantum adder has a slightly reduced number of qubits, Toffoli gates, and depth, but it has a higher CNOT count compared to the quantum ripple carry adder. The carry-lookahead variants offer lower depth but come with increased qubit and Toffoli gate counts.

2.7 Error Metrics

This section discusses the metrics employed to assess both the magnitude and frequency of errors caused when quantum circuits are executed in non-ideal circumstances such as noise based simulations or NISQ quantum computers. The primary metric, known as Error Distance (ED), is defined as the absolute difference between the exact output (S_{exact}) and the actual output (S_{approx}).

Table 2.5: Resource Comparison of Quantum Adders.
($\log_2$ denoted as log and w(n) is number of 1's in binary representation of n.)

Adder	**Qubits**	**Depth**		**Count**	
		CNOT	**Toffoli**	**CNOT**	**Toffoli**
Ripple Carry (Cuccaro et al.) [23]	$2n+2$	$3n+2$	$2n$	$4n+1$	$2n$
No Input Carry (Thapliyal et al.) [25]	$2n+1$	$3n-2$	$2n-1$	$5n-5$	$2n-1$
Carry-Lookahead (Draper et al.) In-Place [24]	$4n-1-w(n)-\lfloor \log n \rfloor$	4	$\lfloor \log n \rfloor + \lfloor \log(n-1) \rfloor + \lfloor \log \frac{n}{3} \rfloor + \lfloor \log \frac{n-1}{3} \rfloor + 10$	$4n-5$	$10n-3w(n)-3w(n-1) - 3\lfloor \log n \rfloor - 3\lfloor \log(n-1) \rfloor - 7$
Carry-Lookahead (Draper et al.) Out-of-Place [24]	$4n+1-w(n)-\lfloor \log n \rfloor$	3	$\lfloor \log n \rfloor + \lfloor \log \frac{n}{3} \rfloor + 4$	$3n-1$	$5n - 3w(n) - 3\lfloor \log n \rfloor - 1$

In Equation 2.14, we derive the Mean Error Distance (MED) by averaging the ED over the total number of input combinations (N). Using Equation 2.15, we normalize the MED by the maximum output across all inputs (S_{max}) to lessen the impact of outliers and variations in error distance distribution. NMED is especially valuable as it considers both the scale and distribution of the error, facilitating meaningful comparisons across various input scenarios [83].

$$ED = |S_{exact} - S_{approx}| \tag{2.13}$$

$$MED = \frac{\sum ED}{N} \tag{2.14}$$

$$NMED = \frac{MED}{S_{max}} \tag{2.15}$$

Another metric for assessing the accuracy of quantum circuits is the Error Rate (ER), which indicates the ratio of incorrect outputs to total input combinations. This is calculated as shown in Equation 2.16, where the number of incorrect outputs (N_{error}) is divided by the total number of input combinations (N). Quantum arithmetic circuits with lower error rates produce errors in fewer input combinations, making them ideal for applications like sensor data collection and image/audio filtering, where maintaining overall quality is crucial while ignoring outliers.

$$ER = \frac{N_{error}}{N} \tag{2.16}$$

Chapter 3

Logarithmic Depth Quantum Carry-Lookahead Modulo (2^n - 1) Adder

Quantum modulo arithmetic circuits are a specific type of quantum arithmetic circuits that execute arithmetic operations by computing the remainder after division by a specified modulus. The outcome of a modulo operation is permanently confined to a limited range (from 0 to modulus - 1), which guarantees closure for addition, subtraction, and multiplication. Among the different types of quantum modulo operations, the quantum modulo adder stands out as the most fundamental and adaptable quantum circuit, serving as a foundation for developing quantum modulo arithmetic circuits for tasks such as comparison, subtraction, and multiplication. Additionally, quantum modulo adders play important roles in areas like Quantum Image Processing (QIP), quantum cryptography, and quantum convolutional neural networks by improving computational efficiency and operating within a finite numerical range [1, 84, 85, 86, 3].

Vedral et al. created one of the first known general modulo N adders [26]. This adder produces a modulo sum by adding the two inputs and subtracting N through

a straightforward mechanism. The ultimate output is determined by whether the sum exceeds or falls short of N. However, this design consists of five stages, which increase both the quantum gate count and depth, rendering it impractical for the NISQ era. Therefore, future research efforts are directed toward developing resource-efficient modulo adders that can be utilized for important moduli with significant practical applications in specific areas. For instance, the quantum modulo 2^n adder is employed in quantum image scrambling and quantum image encryption [86]. Most quantum modulo 2^n adders can be easily derived from existing quantum full adder designs through carryout truncation [23, 87]. In contrast, the quantum modulo (2^n - 1) adder lowers computational costs in public cryptographic systems that function on moduli that can be expressed in Galois Field GF(2^n - 1) [3, 84, 85]. Nonetheless, unlike the designs for quantum modulo 2^n adders, creating a quantum modulo (2^n - 1) adder is not a simple task, with the design by Kim et al. currently being the most efficient alternative [88].

In contrast to Vedral et al., Kim et al. proposed design utilizes only two stages. However, it continues to use Quantum Ripple Carry Addition (QRCA) to create those two stages, thereby inheriting the limitations associated with QRCA [26, 88, 23]. The QRCA logic features a lengthy carry path, which contributes to deeper carryout and increased susceptibility to noise. Furthermore, the Least Significant Bits (LSB) of the output are produced last, despite being the initial inputs, resulting in longer idle times and a greater likelihood of being affected by amplitude damping and depolarizing noise sources [27].

To mitigate these challenges, we recommend the implementation of a quantum carry-lookahead modulo (2^n - 1) adder (QCLMA), which has a depth of O(log n), unlike Kim et al.'s quantum modulo (2^n - 1) adder that has a depth of O(n). To demonstrate the enhanced noise fidelity of our proposed quantum modulo (2^n - 1) adder in comparison to the version by Kim et al. [88], we conducted experiments on a 27-qubit quantum computer named IBM Cairo using IBM Qiskit [89]. Our findings indicate that the reduced depth of QCLMA, as opposed to Kim et al.'s version,

leads to decreased idle time available for the qubits, which may enhance the circuit's noise fidelity. In addition, QCLMA employs a tree structure for generating carryout, which is less prone to error accumulation than Kim et al.'s quantum modulo (2^n - 1) adder, characterized by a single elongated carry path passing through all qubits. Consequently, the enhanced noise fidelity of the proposed QCLMA renders it a more appropriate option for NISQ-era quantum computers that are affected by higher noise levels and erroneous outcomes. In the Fault Tolerant Quantum (FTQ) era, where quantum computers incorporate error correction, the proposed QCLMA would still hold significance, as it would perform more efficiently due to its lower depth.

This Chapter is organized with Section 3.1 providing background on modulo (2^n - 1) addition. Section 3.2 discusses the existing designs for quantum modulo adders. Section 3.3 presents the proposed Quantum Carry-Lookahead Modulo (2^n - 1) Adder (QCLMA). Section 3.4 compares the performance of the proposed QCLMA with Kim et al.'s version[88] and reasons behind the superior performance of the proposed quantum modulo (2^n - 1) adder. Finally, section 3.5 concludes this work.

3.1 Modulo (2^n - 1) addition

Modulo addition is a mathematical operation that involves adding two numbers and then taking the remainder when divided by a fixed positive integer, known as the modulus. The modulo addition function takes two inputs, a and b, satisfying $0 \leq a,b < N$, and outputs (a + b) mod N, where N is the modulus. Modulo addition is applicable in several fields, particularly in areas where periodicity, cyclic patterns, or constraints within a fixed range are involved. Some of the key fields in classical computing where modulo addition plays a vital role include digital signal processing and number theory, particularly in the study of congruences to solving problems related to divisibility and prime numbers. In the domain of security, modular addition is utilized in cryptography, hash functions, error detection, and correction techniques,

including Cyclic Redundancy Checks (CRC), to ensure the integrity of transmitted data.

$$Sum = \begin{cases} a+b, & \text{if } 0 \leq (a+b) < 2^n - 1 \\ 0, & \text{if } (a+b) = 2^n - 1 \\ (a+b) mod (2^n - 1) & \text{if } (a+b) > 2^n - 1 \end{cases} \tag{3.1}$$

$$\text{where } \ 0 \leq a, b, Sum < 2^n - 1$$

General modulo functions often involve division and remainder operations, which can be computationally expensive for frequent or large-scale calculations. Hence, customized functions for specific moduli, such as powers of 2, can use optimized binary operations to improve performance. One such moduli is (2^n - 1), which are also called as mersenne numbers. Equation 3.1 shows the modulo (2^n - 1) addition as a particular instance of modulo addition with the modulus N being (2^n - 1).

Mersenne numbers that are prime use a prime number 'n' to create them. Mersenne primes are some of the largest known prime numbers. Similarly, the largest known composite numbers, both with and without known factors, are also mersenne composite (or non-prime) numbers. Due to their structure and primality, mersenne numbers are used in elliptical curve cryptography and random number generation, making modulo (2^n - 1) addition worth optimizing. Utilizing modulus (2^n - 1) alongside other moduli like 2^n and (2^n + 1) in a Residue Number System (RNS) can also enhance computational efficiency [90]. Ultimately, the modulo (2^n - 1) adder can facilitate additional operations within the modulo and RNS algebra, such as subtraction and multiplication [15].

3.2 Existing Designs for Quantum Modulo Adders

The study of quantum modulo adders originated with the work of Vedral et al., who introduced a general quantum modulo adder capable of producing an output of

(a+b) mod N, where a and b are inputs that satisfy $0 \leq a,b \leq N$ [26]. Their approach utilizes the result from a quantum full adder and adjusts it by subtracting N based on whether the sum a + b exceeds or falls short of N. However, despite the versatility to accommodate any integer modulus N, their design involves five stages based on quantum addition, rendering it impractical for NISQ quantum computers and applications.

Further research concentrated on optimizing quantum modulo adders for specific applications. For example, Roetteler et al. have proposed a version of the quantum modulo adder intended for reversible elliptic curve operations in quantum cryptography, which utilizes a fixed integer modulus [18]. However, like Vedral et al. their design also faces the drawback of high depth since it incorporates four stages. Contrarily, Markov and Saeedi do not create a modulo adder; instead, they introduce a reversible circuit for conditional modular addition with a constant, as their emphasis is on optimizing modular multiplication for Shor's algorithm in quantum number factoring [15].

Research on quantum modulo adders has also progressed alongside the creation of quantum full adders. Cuccaro et al. and Takahashi et al. have adapted their designs for quantum full adders to develop quantum modulo 2^n adders by utilizing the straightforward method of truncating the carryout [23, 87]. This approach is effective because it delivers an n-bit output from n-bit input addition, with the truncated carryout denoting the most significant bit (MSB) representing the 2^n position.

Draper et al. introduce a novel architecture for quantum carry-lookahead adders that operates with O(log n) depth and expand their design to include both modulo 2^n and modulo $(2^n - 1)$ adders [24]. However, their study lacks adequate details regarding the conversion process and its constraints. Furthermore, their design employs one's complement arithmetic, which denotes zero using both an all-zero and an all-one representation, resulting in the issue of zero's double representation that remains unaddressed in their research.

$$Sum = \begin{cases} 0, & \text{if (a+b)=}2^{\text{n}} - 1 \quad (3.2a) \\ (a + b + c_{\text{n}}) mod (2^{\text{n}} - 1), & \text{else} \quad (3.2b) \end{cases}$$

where $0 \leq a, b, Sum <2^{\text{n}}$ - 1

Kim et al. develop a quantum modulo adder for the Galois Field GF(2^{n} - 1) that produces the result of (a+b) mod (2^{n} - 1), given the inputs meet the condition: 0 $\leq$ a,b <(2^{n} - 1) [88]. Their method operates in two phases. Initially, they apply Equation 3.2b with Carry generation logic to compute the carryout c_{n} and provide this as an input Carry to a Carry truncated full adder for the addition of a and b. In Figure 3.1, it is illustrated that their optimized circuit employs two ripple carry addition units: the first unit for Carry generation and the second for Carry truncated addition. However, if the Sum equals the modulus, the set zero logic activates to reset the output to zero. Figure 3.1 depicts the set zero logic utilizing two CNOT and two NOT gates positioned outside the addition units. Additionally, the set zero logic addresses the issue of the double representation of zero.

3.3 Proposed Design

In the prior section, we examined the design of the quantum modulo (2^{n} - 1) adder by Kim et al. However, since both components of Kim et al.'s quantum modulo (2^{n} - 1) adder rely on Quantum Ripple Carry Addition (QRCA) logic, it encounters challenges similar to those found in Vedral et al.'s work. For instance, the depth increases linearly with the size of the input qubits, resulting in a longer Carry propagation chain, which in turn diminishes noise fidelity. We tackle these challenges in our proposed Quantum Carry-Lookahead Modulo (2^{n} - 1) Adder (QCLMA) by introducing a logarithmic depth with a complexity of O(log n). Additionally, we implement a tree-based Carry propagation method that is more robust against noise than the chain-based Carry propagation employed in QRCA.

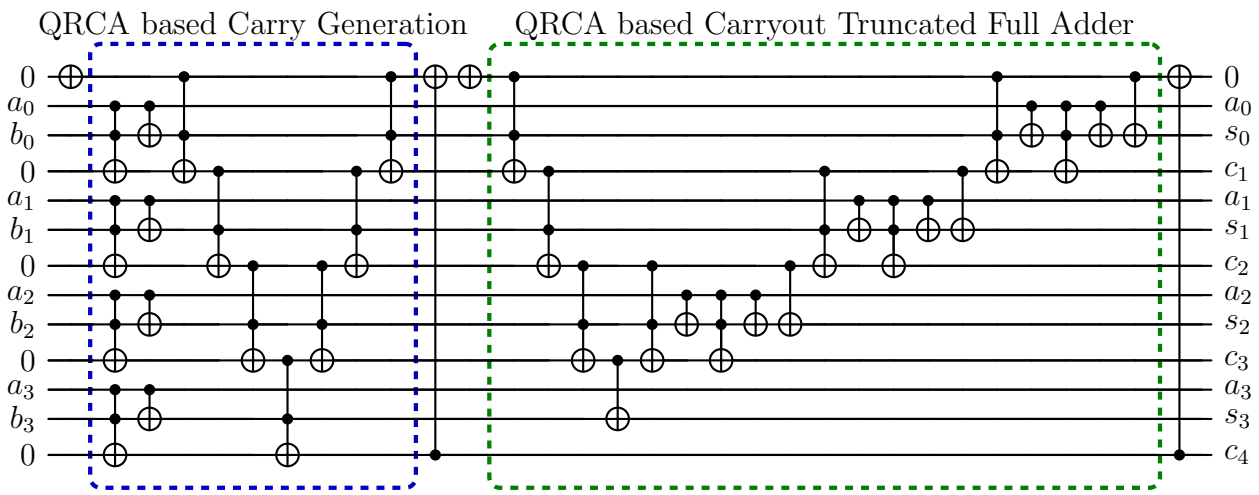


Figure 3.1: Quantum ripple-carry adder (QRCA) based quantum modulo (2^n - 1) adder by Kim et al. in 4-qubit configuration [88]. The inputs are (a_0:a_3) and (b_0:b_3) while (S_0:S_3) represent Sum qubits and (c_1:c_4) represent the Carry qubits. The QRCA mechanism causes higher depth and longer paths in both Carry and the Sum generation, increasing the noise susceptibility. The LSB qubits are more noise prone as they spend significantly longer time idling than the MSB qubits [27].

To create the proposed Quantum Carry-Lookahead Modulo (2^n - 1) Adder (QCLMA), we choose selectively between two categories of logic: in-place and out-of-place. The out-of-place logic allows both inputs to remain unchanged while generating the Sum on ancillae. As shown in Figure 3.2, we employ it as the initial stage of our quantum modulo (2^n - 1) adder since it can forward the inputs for the second stage. We enhance the out-of-place logic to produce only the Carry while also passing the inputs to the next stage. In addition, we prepare the first ancilla in the state $|1\rangle$ to remove the unnecessary NOT gate.

The initial stage produces carryout c_n for the subsequent stage, which it utilizes as input Carry for the carryout truncated full adder, as described in Equation 3.2b. In this case, we employ an in-place quantum carry-lookahead adder that generates Sum directly from input B, thereby conserving ancillae, as illustrated in Figure 3.2. We leverage the set zero logic mechanism from previous implementation by Kim et al. Table 3.1 presents the quantum resource utilization and depth for both types of adders. We observe a rise in the number of qubits utilized in the proposed QCLMA due to the additional qubits used as ancillae. But the depth complexity is O(log n), compared to O(n) for the quantum modulo (2^n - 1) adder developed by Kim et al.

For a 4-qubit configuration, the proposed QCLMA has 38% lower CNOT depth compared to Kim et al.'s modulo (2^n - 1) adder. Here CNOT depth is defined as a measure of CNOT gate layers that can be executed in parallel in the quantum circuit. Figures 3.1 and 3.2 illustrates that the CNOT depth in the proposed QCLMA and Kim et al.'s version are eight and thirteen, respectively. Similarly, the proposed QCLMA has 23% lower Toffoli depth compared to Kim et al.'s modulo (2^n - 1) adder, as the Toffoli depth in proposed QCLMA and Kim et al.'s version are thirteen and seventeen, respectively. We define Toffoli depth as a measure of Toffoli gate layers that can be executed in parallel in the quantum circuit. In the section 3.4, we present a detailed analysis of the implications of depth reduction on noise resilience.

For a configuration of 4-qubit output, the suggested QCLMA exhibits a CNOT depth that is 38% lower when compared to Kim et al.'s modulo (2^n - 1) adder. In

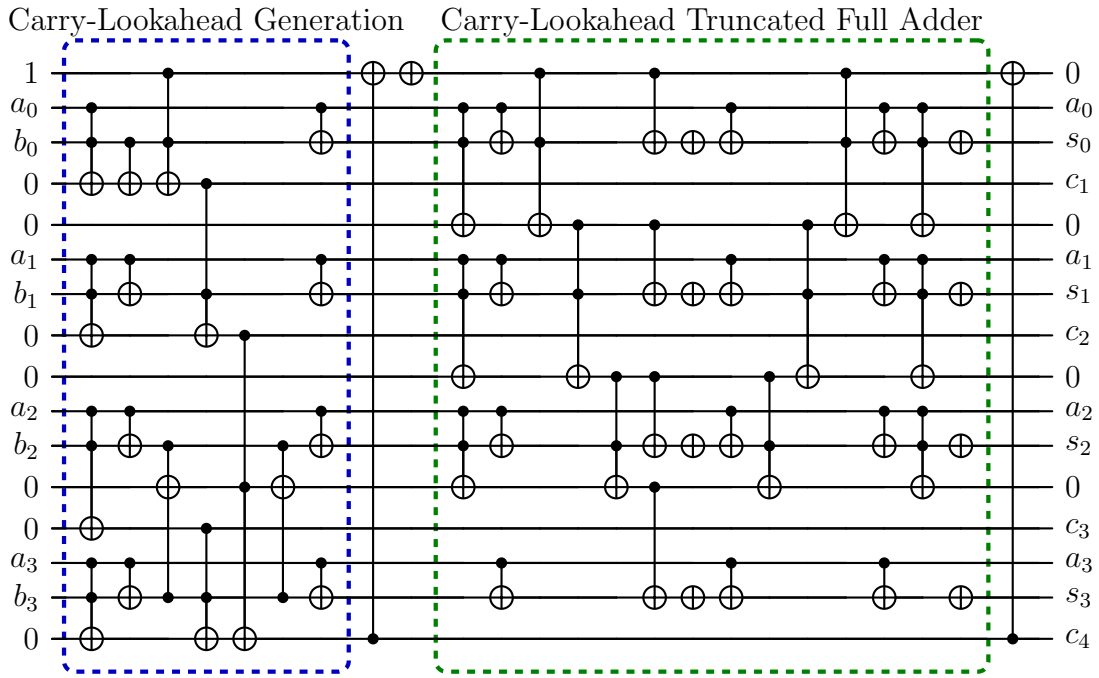


Figure 3.2: Proposed quantum carry-lookahead modulo adder (QCLMA) in 4-qubit configuration. First, the Carry generation logic takes the inputs (a_0:a_3) and (b_0:b_3) to generate Carry qubits (c_0:c_4). The out-of-place carry-lookahead generation logic makes sure that the inputs are passed intact for next stage. The in-place carry-lookahead truncated full adder generates the Sum (S_0:S_3) in place of input b. Both the Carry and Sum are generated using a tree-based structure which has a shorter O(log n) depth. Lesser variation in idling time among the qubits makes the proposed QCLMA noise resilient.

Table 3.1: Quantum Resources for Proposed QCLMA and Kim et al.'s QRCA based Modulo (2^n - 1) Adders. ($\log_2$ denoted as log and w(n) is number of 1's in binary representation of n.)

Adder	**Modules**	**Depth**		**Gate Count**			**Qubits**
		CNOT	**Toffoli**	**CNOT**	**Toffoli**	**NOT**	
Kim et al.	Carry Generator	1	2n	n	3n - 1	0	
	Truncated Adder	3n - 2	3n - 3	3n - 2	3n - 3	0	
	Set Zero Logic	2	0	2	0	2	
	Total	3n + 1	5n - 3	4n	3n -4	2	3n+1
Proposed QCLMA	Carry Generator	2	$\lfloor log(n) \rfloor$+3	2n	4n-3w(n)-2$\lfloor log(n) \rfloor$+1	0	
	Truncated Adder	4	$\lfloor log(n) \rfloor$+2$\lfloor log(n-1) \rfloor$+2	4n	4n-4	2n	
	Set Zero Logic	2	0	2	0	1	
	Total	8	3$\lfloor log(n) \rfloor$+2$\lfloor log(n-1) \rfloor$+5	6n+2	8n-3w(n)-2$\lfloor log(n) \rfloor$-3	2n+1	3n+2+2$\lfloor log(n) \rfloor$

this context, CNOT depth is characterized as a metric of the layers of CNOT gates that can be executed concurrently within the quantum circuit. Figures 3.1 and 3.2 demonstrate that the CNOT depth for the proposed QCLMA is eight, while Kim et al.'s approach has a depth of thirteen. Likewise, the proposed QCLMA shows a 23% reduction in Toffoli depth relative to Kim et al.'s modulo (2^n - 1) adder, with the Toffoli depths for the proposed QCLMA and Kim et al.'s design being thirteen and seventeen, respectively. We define Toffoli depth as a measure of the layers of Toffoli gates that can be simultaneously executed within the quantum circuit. In section 3.4, we provide a comprehensive analysis of how the reduction in depth affects noise resilience.

3.4 Noise-Resilience Analysis

To evaluate the noise resilience of the proposed quantum modulo (2^n - 1) adder in comparison to Kim et al.'s quantum modulo (2^n - 1) adder, we conduct experiments on a 27-qubit IBM quantum computer (IBM Cairo) following a specific methodology. We construct 4-qubit versions of both circuits, as illustrated in Figure 3.1 and Figure 3.2. With four bits, the modulus is (2^4 - 1) = 15, and the range of input and output values is $0 \leq a, b, Sum < 15$. For the proposed quantum adder, we maintain the value of input A as constant while varying the value of B from 0 to 14, resulting in the creation of fifteen circuits that are combined into a single job before being submitted to IBM Cairo. We group together fifteen such jobs for each value of A within the range of 0 to 14 in order to operate the proposed quantum adder for every one of the 255 input combinations. We apply the same approach to generate and organize the jobs for Kim et al.'s quantum adder as well. Each job is executed with 1024 shots. The results from IBM Cairo are obtained as shot frequency across the sixteen possible outputs, ranging from 0000 to 1111.

Due to the noisy outputs produced by NISQ-era machines like IBM Cairo, the correct result may not rank among the top three outputs and is highly dependent

on the specific input combinations. For instance, when the input combination that creates the least delay is A = B = 0, the correct result corresponds to the output with the highest shot count for both quantum modulo (2^n - 1) adders. In contrast, for the input combination that leads to the maximum delay, A = B = 14, the appropriate output ranks fourteenth for Kim et al.'s modulo adder and sixth for the proposed modulo adder. Since the position of the correct output among all possible outputs does not indicate its divergence from the leading output, we require an additional criterion to enable a meaningful comparison of the adders.

$$\text{QSFR} = \frac{\text{Frequency of Correct Output}}{\text{Frequency of Top Output}} \tag{3.3}$$

Therefore, we create a metric known as the Quantum State Fidelity Ratio (QSFR) to assess the degree of closeness of the correct output with the top output. As illustrated in Equation 3.3, QSFR represents the ratio of the frequency of the correct output to that of the top output. QSFR enables us to quantify the fidelity and the level of enhancement required to elevate the correct answer to the top position. We compute the average QSFR for each value of input A combined with B over the range (0, 14) and illustrate the results in Figure 3.3.

Each mark on the x-axis denotes the average QSFR calculated for a fixed input A, while input B varies between 0 and 14. The overall mean of these average QSFR values helps assess the design's improvement requirement across all 255 input combinations, yielding a result of 0.6891 for the proposed QCLMA in contrast to 0.4681 for Kim et al.'s QRCA based on modulo (2^n - 1) adder. The notable 47.21% increase in average QSFR underscores the enhanced noise fidelity and greater improvement potential of the proposed QCLMA.

To better understand the several factors that contribute to the enhanced performance of the proposed Quantum Carry-Lookahead Modulo Adder (QCLMA), we analyze an input combination of A = 10 and B = 2. In this case, both modulo (2^n - 1) adders produce the output S = 12, which is the eleventh output out of

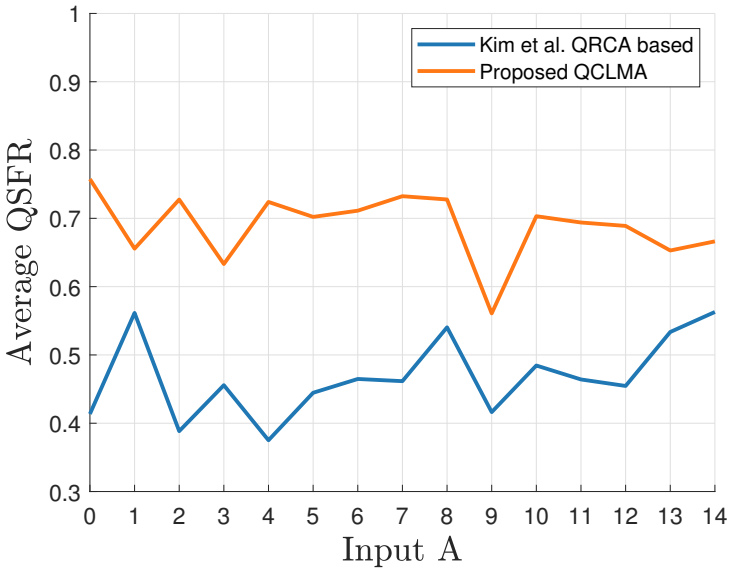


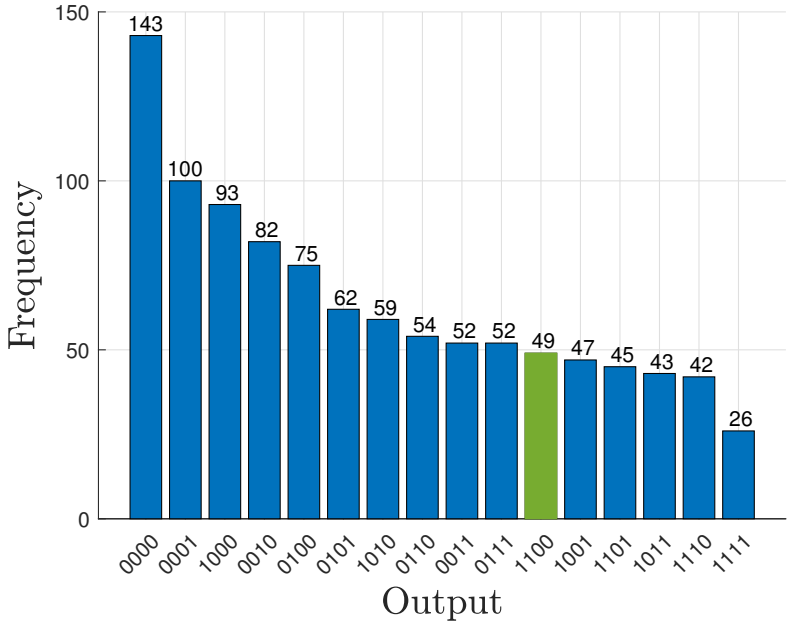
Figure 3.3: Comparison of Average QSFR of Proposed QCLMA and Kim et al.'s QRCA based modulo (2^n - 1) adder, obtained keeping constant A and varying B. Proposed adder has 47.21% higher average QSFR establishing superior noise fidelity.

sixteen possible outcomes. Figure 3.4 illustrates the differences in QSFR between the two scenarios, even though the frequency of the correct output is nearly the same. For example, Figure 3.4(b) indicates that the proposed QCLMA achieves the correct output with a frequency of 55 and a QSFR of 0.55, whereas Kim et al.'s QRCA based modulo (2^n - 1) adder, depicted in Figure 3.4(a), has a frequency of 49 and a QSFR of 0.343.

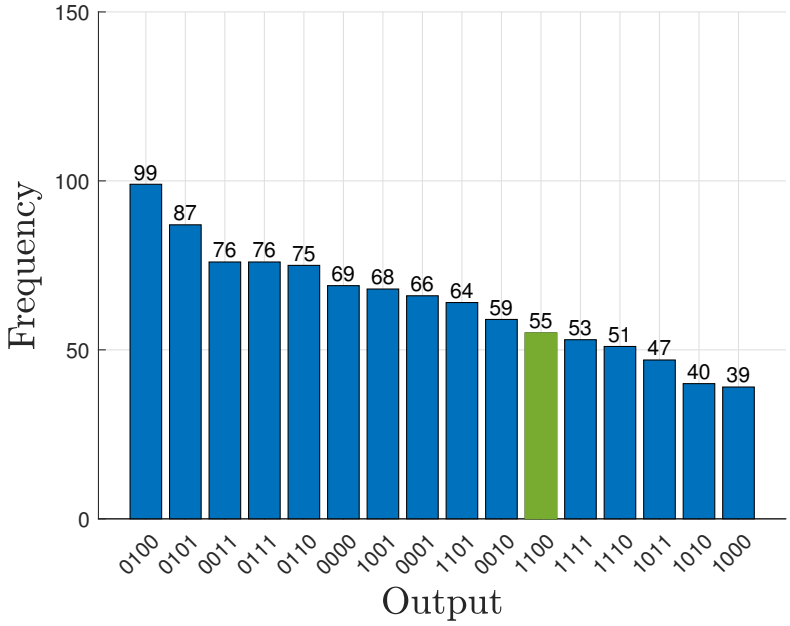
Figure 3.4(a) illustrates the output frequency profile of the QRCA-based modulo (2^n - 1) adder by Kim et al., showcasing a nearly exponential drop from the highest output. This indicates that unless it is the top output, the correct result will exhibit an exponentially increasing gap from the top output. The cause of this exponential decline can be traced back to the QRCA architecture, which introduces significant variation in depth and qubit idle time. In contrast, Figure 3.4(b) reveals that the proposed QCLMA experiences a linear decrease in frequency under identical experimental conditions, indicating greater resilience to noise-related disruptions. The tree structure of QCLMA effectively creates a self-normalization effect, minimizing variation between outputs and enhancing robustness against noise.

3.5 Conclusion

In this study, we have introduced a quantum carry-lookahead modulo (2^n - 1) adder (QCLMA) that primarily reduces the depth to O(log n) as opposed to the O(n) depth seen in previous works. We illustrate the significance of O(log n) depth in enhancing noise resilience through experiments conducted on a 27-qubit quantum computer known as IBM Cairo. Consequently, the proposed QCLMA shows an improvement of 47.21% in noise fidelity, utilizing the Quantum State Fidelity Ratio (QSFR), which quantifies how closely the accurate output aligns with the optimal output. We also identify the source of this performance by analyzing the output frequency profile of our quantum modulo (2^n - 1) adder against existing QRCA based modulo (2^n - 1) adder by Kim et al. under the same input and output conditions.



(a) Kim et al.'s QRCA based Modulo (2^n - 1) adder



(b) Proposed Quantum Carry-Lookahead Modulo Adder (QCLMA)

Figure 3.4: The above graphs illustrate the output of the modulo adders generated from IBM Cairo, arranged in descending order of frequency. Identical inputs A = 10 and B = 2 is supplied, with both adders yielding S = 12 or 1100 as the output at the same 11^{th} position among 16 possible outputs. The output profile of our proposed QLCMA adder exhibits a slower descent rate than the version proposed by Kim et al., reducing the relative gap with the top output and improving its chances of advancement.

Our findings demonstrate that the decreased depth and the distributed Carry path enhance noise fidelity, making the QCLMA appropriate for quantum computers in the Noisy Intermediate-Scale Quantum (NISQ) era. The proposed QCLMA could play a key role in developing circuits for quantum modulo subtraction and multiplication, as well as in applications like quantum cryptography and quantum image processing. We conclude that QCLMA may significantly improve the noise resilience of quantum modulo arithmetic circuits.

Chapter 4

Novel Optimized Designs of Quantum Modulo (2^n+1) Adders

Quantum modulo arithmetic circuits are specialized circuits designed to address modulo operations particularly. As discussed in Chapter 3, these operations guarantee that the outcome remains within the range of (0, modulus-1), thereby preserving closure under operations like addition, subtraction, and multiplication. Among these circuits, the quantum modulo adder is particularly valuable, as it allows for the creation of other quantum modulo arithmetic circuits, including those for multiplication and exponentiation. With the rise of quantum computing, various designs for quantum full adders and quantum modulo adders have been introduced [23, 24, 16, 25]. In spite of that, to our knowledge, there is no current design aimed explicitly at and optimized for a quantum modulo (2^n + 1) adder. The development of a quantum modulo (2^n + 1) adder has the potential to enhance existing quantum modulo arithmetic circuits, enhancing their applicability in pseudo-random number generation, Residue Number Systems (RNS), and quantum cryptanalysis [91, 17, 18, 19]. However, devising an efficient algorithm for (a + b) modulo (2^n + 1) addition presents a significant challenge. To the best of our knowledge, no quantum

modulo (2^n+1) adder design currently exists. In this Chapter, we introduce two types of quantum adders that are specifically designed for modulo (2^n+1) addition.

The first category, known as the Quantum Modulo (2^n+1) Adder (QMA), conducts (a + b + 1) modulo (2^n+1) addition within the standard number system, where a and b serve as inputs. This method offers the closest alternative to the initial problem statement. The second category, referred to as the Quantum Diminished-1 Modulo (2^n+1) Adder (QDMA), executes (a + b) modulo (2^n+1) in a diminished-1 number system. This method delivers the precise output but necessitates that the problem be converted into the diminished-1 number system. Depending on the requirements of the application and the appropriateness of the number system, the user has the option to choose between QMA and QDMA designs. Both categories of adders are capable of providing the standard and modulo (2^n+1) sums simultaneously, thereby enhancing their utility.

Furthermore, in this Chapter, we explore systematic optimization using static and dynamic circuit based approaches and propose four designs. Initially, we improve QMA1 to decrease the quantum gate count in QMA2, benefiting both Noisy Intermediate Scale Quantum (NISQ) devices and Fault Tolerant Quantum (FTQ) systems while preserving reversibility. To achieve a further reduction in qubit usage, we introduce two dynamic quantum modulo (2^n + 1) adders that incorporate non-reversible zero reset operations. The proposed adder QMA3 employs zero resets to decrease the number of ancilla qubits needed. In the design of QMA4, we apply zero resets two times to mitigate errors that arise from state preparation. We conduct experiments by implementing the four proposed designs of QMA on the IBM Washington quantum computer, which is based on a 127-qubit Eagle R1 architecture. Our experiments demonstrate a cumulative error reduction of 28.8%, validating the efficacy of our systematic optimization approach.

This Chapter is organized as follows. Section 4.1 provides background on modulo (2^n + 1) addition operation. Section 4.2 presents the proposed designs of Quantum

Modulo (2^n+1) Adder (QMA) and their resource comparison followed by noise-resilience analysis. Section 4.3 proposes Quantum Diminished-1 Modulo (2^n+1) Adder (QDMA), and compares it against QMA. The Section 3.5 at last concludes this Chapter.

4.1 Modulo ($2^n + 1$) addition

The modulo operation of addition with respect to ($2^n + 1$) involves two inputs, a and b, producing a Modulo Sum within the range $0 \leq a, b, \text{Modulo Sum} < 2^n + 1$. As illustrated in Equation 4.1, if the Modulo Sum reaches or surpasses ($2^n + 1$), it results in $(a + b) \mod (2^n + 1)$, identical to the remainder derived from division by the modulus.

$$ModuloSum = \begin{cases} a + b, & \text{if } (a + b) < 2^n + 1 \\ 0, & \text{if } (a + b) = 2^n + 1 \\ (a + b) mod (2^n + 1) & \text{if } (a + b) > 2^n + 1 \end{cases} \tag{4.1}$$

where $0 \leq a, b, ModuloSum < 2^n + 1$

This moduli can manage Fermat's numbers, which exist in the form ($2^{2^n} + 1$), where n are non-negative integers. Fermat numbers grow extremely rapidly, making calculations large and computationally intensive. This property also makes factorization of large fermat composite numbers difficult, enhancing their suitability in cryptography. Fermat numbers that are primes are useful in the constructibility of regular polygons. All fermat numbers are mutually co-prime, making them significant in number theory and Residue Number System (RNS). Overall, fermat numbers play an essential role in problems involving modular arithmetic and advanced number theory. Hence, implementing a quantum modulo ($2^n + 1$) adder can help apply quantum cryptanalysis and assist in other modulo ($2^n + 1$) functions like subtraction,

exponentiation, and multiplication [18, 19]. Leveraging the modulus ($2^n + 1$) can enhance computational efficiency when combined with other moduli such as 2^n and (2^n - 1) within a residue number system (RNS) [17]. However, developing an efficient algorithm for modulo ($2^n + 1$) addition is a significant challenge. We explore alternative strategies where:

- By optimizing the quantum (a + b + 1) modulo ($2^n + 1$) adder, which closely reflects the original problem statement if one of the input numbers is reduced by one.
- By developing quantum (a + b) modulo ($2^n + 1$) adder that uses diminished-1 number system.

4.2 Proposed Designs of Quantum Modulo ($2^n + 1$) Adder (QMA)

To perform modulo ($2^n + 1$) addition, we employ Equation 4.2 to design the quantum modulo ($2^n + 1$) adder known as QMA1. The two inputs, A = a_n a_{n-1} ... a_0 and B = b_n b_{n-1} ... b_0, consist of (n+1) qubits each, while the resulting Sum is represented as an (n+2) qubit integer, Sum = A + B = S_{n+1} S_n ... S_0. Using Equation 4.2, we achieve an optimized method for modulo ($2^n + 1$) addition that incorporates various components of the Sum.

$$\begin{aligned} &(a+b+1) mod (2^n+1) = (a+b) mod\, 2^n + S_{n+1}2^n + \overline{(S_{n+1} \vee S_n)} \\ &\text{where} \;\; 0 \leq a, b < 2^n + 1 \end{aligned} \tag{4.2}$$

We propose four Quantum Modulo ($2^n + 1$) Adders (QMA) and optimize them through different techniques. We first propose QMA1 and use only quantum gate reduction based optimization to propose QMA2. Since these designs, QMA1 and QMA2, utilize only static quantum gates, both circuits satisfy reversibility. Later, we

use zero resets, a non-reversible feature based on dynamic circuits to propose QMA3 and QMA4.

4.2.1 QMA1: Proposed Static Quantum Modulo ($2^n + 1$) Adder

To implement Equation 4.2 in quantum circuits, we utilize Algorithm 1. This algorithm employs a quantum full adder to produce an (n+2)-bit sum, represented as $S_{n+1}S_n \ldots S_0$. We create an intermediate (n+1)-bit sum, denoted as $S' = S_{n+1}S_{n-1} \ldots S_0$, by replacing the component S_n with S_{n+1}. The sum S' corresponds to the addition of $(a+b)$ modulo 2^n and $S_{n+1} \cdot 2^n$. To construct the NOR component $\overline{(S_{n+1} \vee S_n)}$ in the proposed quantum modulo (2^n+1) adder, we use a quantum NOR gate designed with a Toffoli gate and four NOT gates placed between the two full adder stages of QMA1, as shown in Figure 4.1.

The NOR component created is inserted into the least significant bit (LSB) of an (n+1)-qubit input that is initialized to $|0\rangle$. This (n+1)-qubit input is subsequently added to the Sum′ by using another quantum full adder, as illustrated in Figure 4.1. Because the output, known as the Modulo Sum M, has an upper limit defined by the modulus, it is constrained to (n+1) bits, represented as $M = M_n \ldots M_0$. We employ the carryless quantum full adder developed by Thapliyal et al. to reduce the number of quantum gates used. Its double-V architecture improves noise tolerance by decreasing the gap between qubit operations, which leads to less idle time for the qubits [25, 33].

Algorithm 1 : Calculation of $(a+b+1) mod (2^n+1)$

Require: $n \geq 1$
Ensure: $0 \leq a, b < 2^n + 1$
1. $Sum = a + b = S_{n+1}S_n \ldots S_0$
2. Omit S_N from Sum to generate: $Sum' = S_{n+1}S_{n-1} \ldots S_0$
3. $ModuloSum = Sum' + \overline{(S_{n+1} \vee S_n)}$

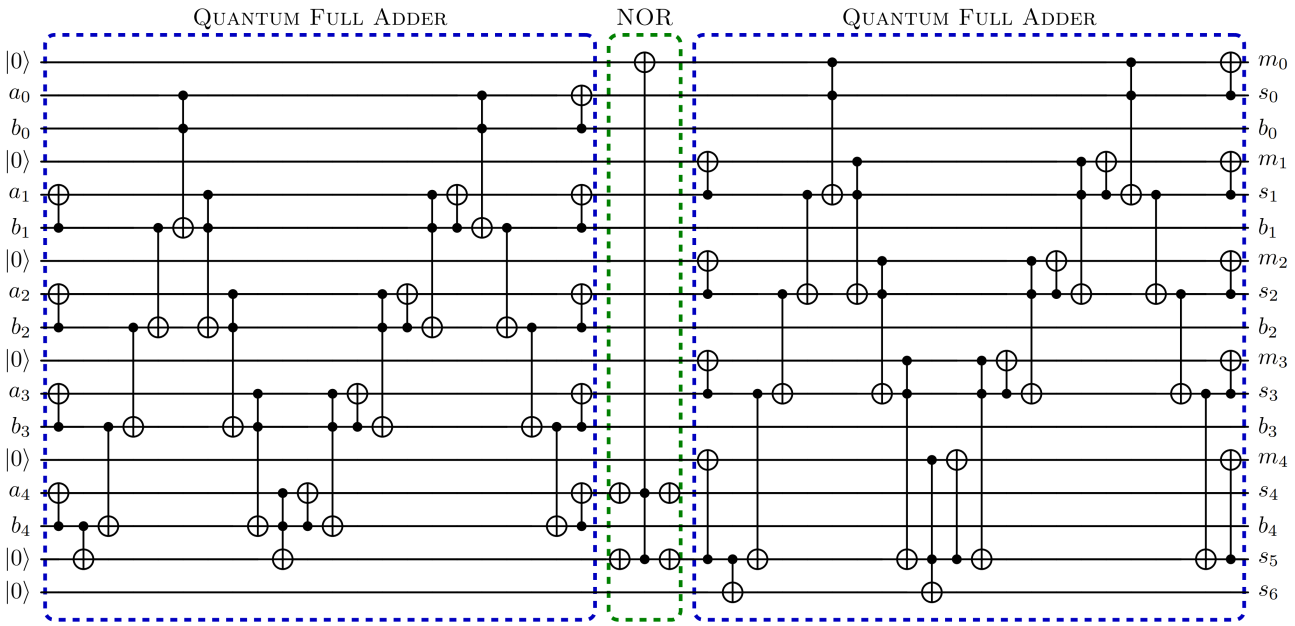


Figure 4.1: Proposed quantum modulo ($2^n + 1$) adder QMA1 for n=4 in a 5-qubit configuration. The QMA1 takes the inputs (a_0:a_4) and (b_0:b_4) to generate Modulo Sum (m_0:m_4). The Sum (s_0:s_6) and input (b_0:b_4) are retained. QMA1 is the basic interpretation of Algorithm 1, using quantum full adder to calculate (a + b) modulo 2^n as intermediate Sum, followed by NOR operation to compute $\overline{(S_{n+1} \vee S_n)}$. Another quantum full adder adds these with S_{n+1} 2^n.

4.2.2 QMA2: Quantum Gate Optimized

In this section, we propose a new quantum modulo $(2^n + 1)$ adder, referred to as QMA2, aimed at enhancing QMA1 by substituting the quantum full adder with a quantum half adder during the second stage of QMA1. This optimization is feasible because the size of the input $\overline{(S_{n+1} \vee S_n)}$ that is added to Sum$'$ is limited to just one bit, resulting in a Carry during addition that acts as a secondary input for the remaining bits of Sum$'$. Using this improvement, QMA2 notably decreases the number of quantum gates and removes the requirement for an additional qubit for Most Significant Bit (MSB) in Sum to maintain reversibility, as shown in Figure 4.2.

4.2.3 QMA3: Zero Reset based Error Reduction

To reduce QMA2's qubit usage, we incorporate a dynamic circuit feature known as zero reset. Specifically, we focus on the qubits that hold the reverse computed input $|b\rangle$, which become unnecessary after they pass through the full adder that computes the intermediate (n+1)-bit Sum$'$ in QMA2. As shown in Figure 4.3 denoting QMA3, by resetting these qubits to $|0\rangle$, we can repurpose them for half addition along with the NOR component $\overline{(S_{n+1} \vee S_n)}$. The integration of zero resets decreases the qubit count by (n-1) and offers two key benefits, albeit at the expense of reversibility. Firstly, the qubits that hold input $|b\rangle$, situated between the control and target qubits of the Toffoli gates in the half adder, are reset to zero, allowing them to serve as target qubits. This reduction of input $|b\rangle$ mitigates design constraints during the physical mapping of qubits on a quantum computer. Secondly, the ancillary qubits initialized to zero, which are later used in QMA2 by the half adder, are vulnerable to decoherence errors when idle [33]. Nonetheless, QMA3 addresses this by resetting the qubits to $|0\rangle$ closer to the half adder, leading to more reliable $|0\rangle$ states. Both of these improvements significantly reduce qubit usage while preserving the functionality of our quantum modulo $(2^n + 1)$ QMA3 adder.

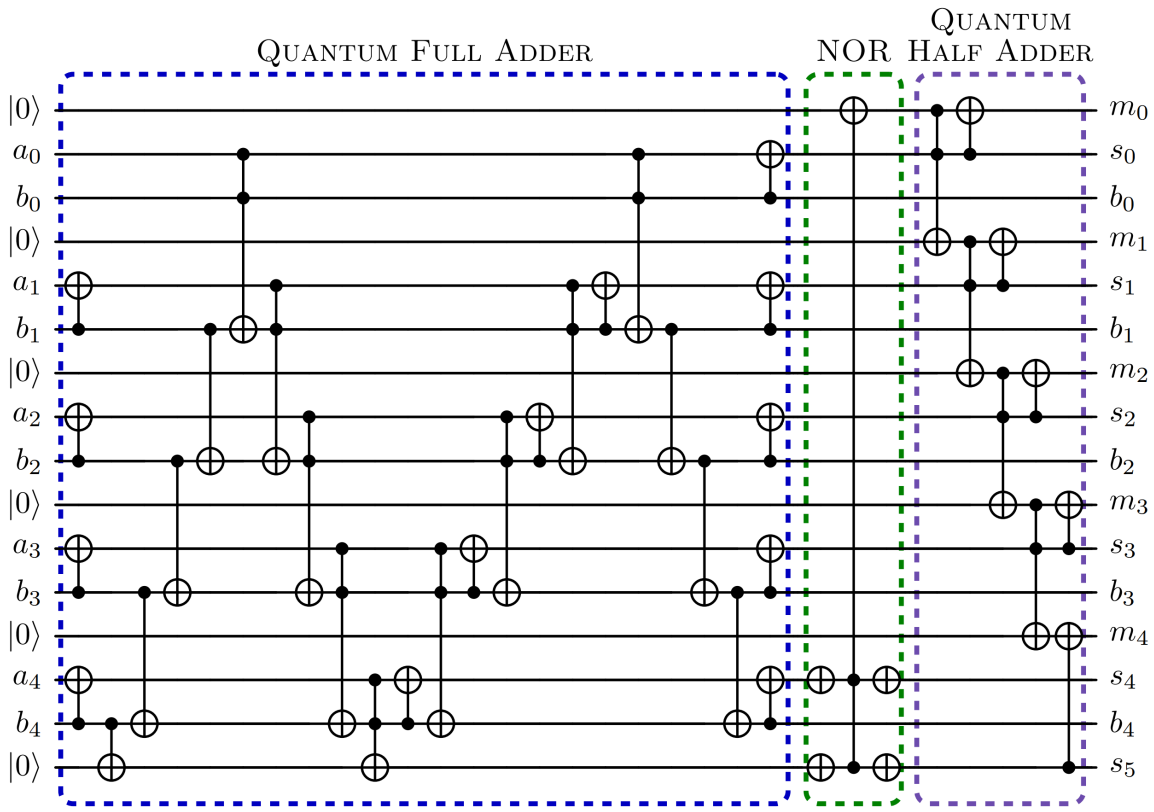


Figure 4.2: Optimized quantum modulo $(2^n + 1)$ adder QMA2 for n=4 in a 5-qubit configuration. The inputs (a_0:a_4) and (b_0:b_4) are used to generate Modulo Sum (m_0:m_4) and Sum (s_0:s_5). Unlike QMA1, QMA2 uses a quantum half adder instead of a full adder to reduce gate count.

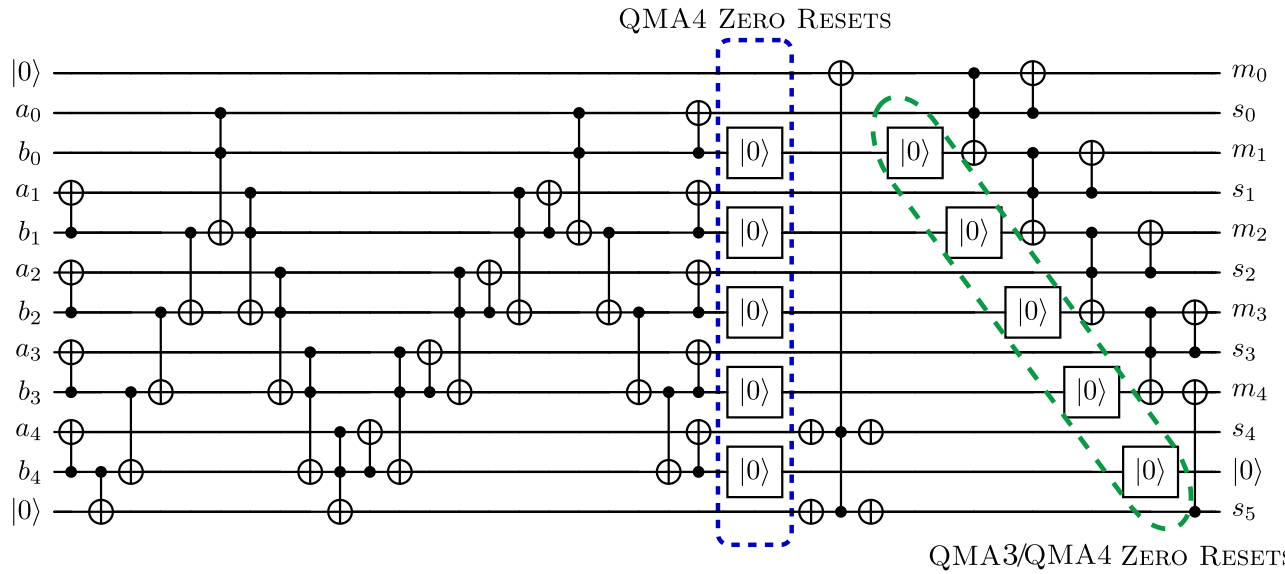


Figure 4.3: Proposed quantum modulo $(2^n + 1)$ adders QMA3 and QMA4 for n=4 in a 5-qubit configuration. Zero reset helps in reusing a qubit by resetting it to $|0\rangle$. The zero resets in the box (a) are in both QMA3 and QMA4, while zero resets in the box (b) are only in QMA4. QMA3 uses the zero resets in the box (a) to help reuse input (b_0:b_4) to generate Modulo Sum (m_0:m_4). Sum (s_0:s_5) is generated in parallel as before. QMA4 utilizes zero resets in box (b), in addition to zero resets present in box (a) for QMA3, to reduce quantum state preparation errors.

4.2.4 QMA4: Reducing Quantum State Preparation Errors

The quantum half adder in QMA3 employs a ripple carry mechanism that remains vulnerable to errors originating from imprecise $|0\rangle$ states. These errors can propagate from the Least Significant Bit (LSB) to the Most Significant Bit (MSB), which could jeopardize the entire computation. Previous approaches to correct quantum state preparation errors in order to attain purer states have required significant quantum resources in the form of extra quantum gates, measurements, and ancillae qubits. However, we adopt a simpler method that reduces design changes by incorporating multiple reset operations to realize a purer $|0\rangle$ state and mitigate state preparation errors.

$$\rho = (1-\delta)^k|0\rangle\langle 0| + \delta^k|1\rangle\langle 1| \tag{4.3}$$

By utilizing 'k' consecutive zero resets, each with a bitflip error probability of δ, we can attain the optimal mixed state density matrix (ρ), as demonstrated in Equation 4.3. The output state can vary from pure to completely mixed for the range of 0 $\leq \delta < \frac{1}{2}$, depending on δ. In QMA3, we add more zero resets to effectively mimic the behavior outlined in Equation 4.3 for k=2. These additional resets are depicted in the box (b) of Figure 4.3, complementing those shown in box (a) of the same figure for QMA3. This revised design is referred to as the proposed quantum modulo (2^n + 1) adder QMA4. Similar to QMA3, QMA4 is also non-reversible, but it achieves a more precise result for the MSB. By decreasing the probability of errors caused by inaccurate $|0\rangle$ states within the half adder, QMA4 exhibits enhanced error resilience compared to QMA3, as detailed in the results section.

4.2.5 Resource Usage and Noise-Resilience Comparison

We conduct experiments using IBM Washington, a 127-qubit quantum computer based on Eagle R1 architecture, on the proposed quantum modulo (2^n + 1) adders

for n=4 in a 5-qubit configuration. We create individual circuits initialized with all possible input combinations of $|0\rangle$ and $|1\rangle$ basis states in IBM Qiskit for each proposed adder from QMA1 through QMA4 [69]. To maintain uniformity and ensure that the circuits encounter equivalent noise and environmental conditions, we execute them consecutively on the quantum computer with a frequency of thousand shots each, using Qiskit's default qubit placement strategy. Upon execution, we identify the output with the highest frequency and compare it against the accurate or ideal Modulo Sum, calculated as (a + b) modulo $(2^n + 1)$. We then compute the Normalized Mean Error Distance (NMED), as explained in Section 2.7, a crucial metric whose decline helps us track the error reduction in the proposed adders.

Table 4.1 presents a detailed comparison of the quantum resource usage and depth of the four proposed quantum modulo $(2^n + 1)$ adders. For n = 4, QMA2 exhibits a significant reduction of 37.5% in the CNOT count compared to QMA1, with values of 25 and 40, respectively. Additionally, the CNOT depth is reduced by 46.15% in QMA2, dropping from 26 in QMA1 to 14. Similarly, the Toffoli count and depth show a reduction of 26.3% from 19 in QMA1 to 14 in QMA2. Hence, QMA2 represents a significant improvement over QMA1 regarding quantum resource conservation and depth reduction. QMA3, on the other hand, reduces the qubit requirement by 25% from 16 to 12 compared to QMA2, while the CNOT and Toffoli usage remains the same. QMA3 also uses 5 zero resets, which double to 10 in QMA4.

To account for the reduction in both circuit depth and width, we introduce a **Figure of Merit (FOM)**: defined as the product of circuit width (also known as qubit count) and its Toffoli depth. Here, we use the Toffoli gate's depth due to its higher potential for introducing error than the CNOT gate. Lower FOM is better, indicating the usage of lower quantum resources. We plot the FOM and NMED for the proposed adders in Figure 4.4. Table 4.1 shows the drop in NMED with respect to QMA1 to help understand the gradual reduction for the proposed adders.

Most of the reduction in NMED, roughly 18.63%, is attributed to the about 30% reduction in the FOM as we transition from QMA1 to QMA2, highlighting the

Table 4.1: Comparison of Proposed Quantum Modulo $2^n + 1$ Adders (QMA) for Quantum Resource Usage and NMED Reduction.

Adder	Qubits	Zero Resets	Depth		Gate Count		NMED Drop(%)
			CNOT	Toffoli	CNOT	Toffoli	
QMA1	$3n+5$	0	$6n+2$	$4n+3$	$10n$	$4n+3$	0.0
QMA2	$3n+4$	0	$3n+2$	$3n+2$	$6n+1$	$3n+2$	18.63
QMA3	$2n+4$	$n+1$	$3n+2$	$3n+2$	$6n+1$	$3n+2$	21.16
QMA4	$2n+4$	$2n+2$	$3n+2$	$3n+2$	$6n+1$	$3n+2$	28.8

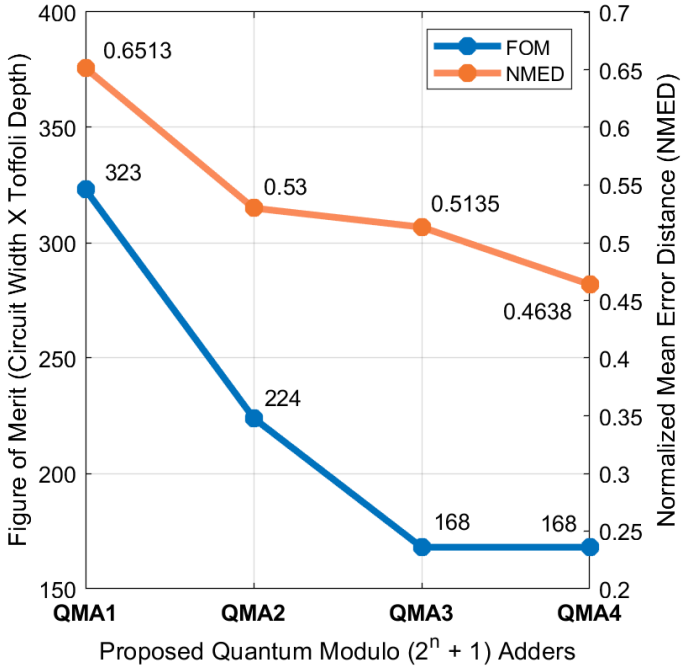


Figure 4.4: The above graph compares the proposed quantum modulo (2^n + 1) adders, QMA1 to QMA4, for n=4 in a 5-qubit configuration, with NMED representing error and Figure Of Merit (FOM) representing resource usage. FOM = (Circuit Width x Toffoli Depth). For both NMED and FOM, lower is better. From QMA1 to QMA3, the NMED falls along with the reduction in FOM, showing the impact of lower quantum resource usage. However, the 7.64% reduction in NMED from QMA3 to QMA4, without any drop in FOM, reflects reduction in quantum state preparation errors due to additional resets.

pronounced impact of the decrease in the Toffoli gate count on error reduction. As we further reduce the FOM from QMA2 to QMA3 by 17.34%, we observe only a modest decline of 2.53% in NMED for QMA3. This implies that while QMA3 (12 qubits) saves 25% qubits compared to QMA2 (16 qubits), its effect on error resilience is less significant than gate depth optimization. We calculate all reductions in NMED and FOM relative to QMA1, the first proposed design. Significantly, we witness another 7.64% drop in NMED over existing reductions when the count of zero resets doubles from 5 in QMA3 to 10 in QMA4, showing that the extra zero resets successfully increase the error resilience of QMA4. Initialization of purer $|0\rangle$ states by additional zero resets in the quantum half adder of QMA4 mitigates the propagation of error to MSB, even though both QMA3 and QMA4 have the same gate and qubit count.

4.3 Proposed Quantum Diminished-1 Modulo $(2^n + 1)$ Adder

In order to achieve modulo $(2^n + 1)$ addition, we utilize Algorithm 2 to construct the Quantum Diminished-1 Modulo $(2^n + 1)$ Adder (QDMA). The two inputs, A = $a_n\ a_{n-1}\ ...\ a_0$ and B = $b_n\ b_{n-1}\ ...\ b_0$, are of size (n+1) qubits, while their Sum is an (n+2) qubit integer Sum = A + B = $S_{n+1}\ S_n\ ...\ S_0$. Algorithm 2 achieves optimized modulo $(2^n + 1)$ addition using different components of the Sum.

Algorithm 2 : Calculation of $(A + B) mod (2^n + 1)$

Input: Inputs A, B are in diminished-1 format. $n \geq 1$
Ensure: $0 \leq A, B < 2^n + 1$

1. Add the inputs except their most significant bits (MSB).
$$Sum = A_{n-1}A_{n-2}...A_0 + B_{n-1}B_{n-2}...B_0 = Carry_n S_{n-1}...S_0$$
2. Using the Carry from Step 2, compute:
$$\overline{A_n}.\overline{B_n}.\overline{Carry_n}$$
3. $ModuloSum = (A_n.B_n, S_{n-1}...S_0) + \overline{A_n}.\overline{B_n}.\overline{Carry_n}$

In order to better handle the complexity introduced for handling 0 in modulo $(2^n + 1)$ arithmetic, we utilize diminished-1 representation. In this approach, we subtract one from the numbers in the range $(1, 2^n)$, while 0 is represented by 2^n. The Most Significant Bit (MSB) of the resulting diminished-1 number is true only when representing 0, helping achieve a more straightforward handling of 0.

The Algorithm 2 reflects the advantage of using the diminished-1 approach. The addition in Step 1 saves resources by having to add only (n-1) qubits, as it handles the $(1, 2^n)$. Step 2 calculates the partial product $\overline{A_n}.\overline{B_n}.\overline{Carry_n}$ which is True only if both input's MSB and Step1's Carry are zero. To create the three input quantum NAND gate, we utilize two Toffoli gates and six NOT gates, one each for the inputs and later for the reversibility. The resulting partial product is then placed at the Least Significant Bit (LSB) of the QDMA, initialized with $|0\rangle$. Step 3 utilizes a quantum half adder to ultimately add the previous step's partial product. The term $A_n.B_n$ in Step 3 helps achieve the addition when both inputs are zero. Figure 4.5 shows the proposed QDMA for modulo 9, achieved when used in n=3 configuration.

As the output or Modulo Sum M cannot exceed the modulus, its size is limited to (n+1)-bits and is represented as $M = M_n \ldots M_0$. Like in previous Section 4.2, we use a quantum full adder called TPL13 without input carry to save quantum resources, and also because its double inverted structure reduces the gap between qubit operations, preventing coherence time issues and reducing noise [25, 33]. In Table 4.2 we compare the quantum resources used in QDMA with QMA2, the closest variant among the four different Quantum Modulo $(2^n + 1)$ Adders. QDMA utilizes an extra qubit, and two extra Toffoli gates while using one CNOT gate lesser compared to QMA2, with no change in depth.

4.4 Conclusion

In this Chapter, we propose two different categories of quantum modulo $(2^n + 1)$ adders, a significant advancement in quantum modular arithmetic. The first one

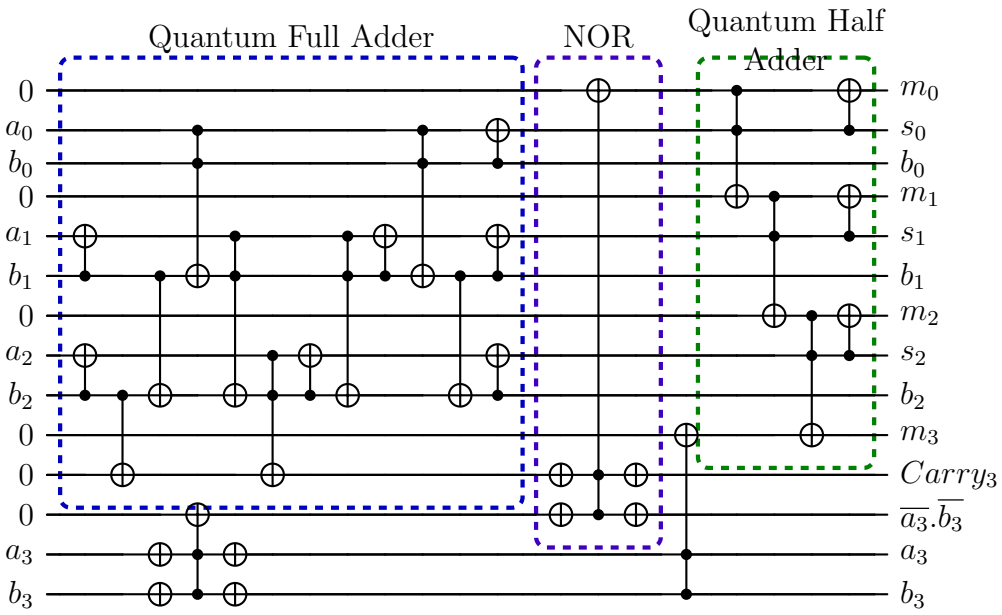


Figure 4.5: Proposed Quantum Modulo ($2^n + 1$) Adder for n = 3 configuration yielding Quantum Modulo 9 Adder. The inputs are A (A_0:A_3) and B (B_0:B_3), while the output is Modulo Sum M (M_0:M_3). Input B (B_0:B_3) and MSB of A (A_3) are passed unchanged. The Sum (S_0:S_2:$Carry_3$) represents the sum of (n-1) qubits of inputs A and B. As shown in Algorithm 2, first the Sum is calculated, with the $Carry_3$ used to compute $\overline{A_3}.\overline{B_3}.\overline{Carry_3}$. Finally, quantum half adder stage calculates the Modulo Sum M after adding (A_3.B_3, S_2 ... S_0) + $\overline{A_3}.\overline{B_3}.\overline{Carry_3}$

Table 4.2: Quantum Resource Usage Comparison of Proposed QDMA and QMA.

Adder	**Qubits**	**Depth (CNOT)**	**Depth (Toffoli)**	**Gate Count (CNOT)**	**Gate Count (Toffoli)**
QDMA	3n+5	3n+2	3n+2	6n	3n+4
QMA2	3n+4	3n+2	3n+2	6n+1	3n+2

provides (a + b + 1) mod($2^n + 1$) addition within the standard number system, called Quantum Modulo ($2^n + 1$) Adder (QMA). We further optimize QMA to propose four novel designs, two each in static and dynamic quantum circuits categories. The QMA1 and QMA2 are static quantum circuits containing design optimizations for quantum gate reduction, and hence are fully reversible. QMA3 and QMA4 are dynamic quantum circuits with non-reversible zero reset operations, that save (n+1) ancillae lines and achieve lower error due to purer $|0\rangle$ initialized states respectively. We demonstrate the error resilience of all four proposed quantum modulo (2^n + 1) adders by experiments on the IBM Washington quantum computer, a 127-qubit system based on Eagle R1 architecture. We demonstrate a 28.8% decrease in error with systematic design optimization of quantum modulo ($2^n + 1$) adders from QMA1 to QMA4.

In second category we propose (a + b) mod(2^n + 1) addition but in the diminished-1 number system, called Quantum Diminished-1 Modulo ($2^n + 1$) Adder (QDMA). The proposed QDMA design utilizes static quantum gates only to support reversibility, and uses marginally different quantum resources compared to QMA2, the closest adder among QMA based designs. We expect our work to lay the foundation for quantum circuits for subtraction, multiplication, and exponentiation within the modulo (2^n + 1) arithmetic. We conclude that as newer applications of quantum computing involving modulo arithmetic emerge, our proposed quantum modulo (2^n + 1) adders will find applications in diverse fields, including residue number systems based applications, quantum signal/image processing, and quantum cryptography.

Chapter 5

Residue Number System (RNS) based Distributed Quantum Computing

Quantum algorithms utilize quantum arithmetic circuits for different purposes, such as scientific computation, efficient data operations, and algebraic functions. Quantum arithmetic operations such as addition, subtraction, multiplication, and exponentiation find usage in quantum applications such as quantum image processing and quantum cryptanalysis [13, 17, 18, 19]. Among the quantum arithmetic circuits, optimizing quantum adder offers the highest incentive as it is a fundamental building block for other operations. Noise accumulation with increasing depth of quantum circuits results in lower output probability, limiting the scaling potential. Thus, a reduction in circuit depth and gate usage for quantum adders is a strong motivation. Recent progress in NISQ quantum computing, especially with ion-trap quantum computers, has resulted in improved gate fidelity. However, while ion-trap quantum computers are more dependable, they have a lower number of qubits compared to superconducting systems, leading to constraints on the size of quantum circuits.

Therefore, improvement in the quantum adder's scalability and applicability on NISQ-era quantum machines requires tackling the twin challenges of improving both circuit depth and width (qubit count) simultaneously.

In this Chapter, we seek to solve these issues for quantum addition and multiplication circuits using a hybrid distributed quantum computing approach. Distributed Quantum Computing (DQC) is a newly emerging approach that builds upon various individual quantum systems to significantly boost computing capabilities by making it possible to scale up quantum applications [92, 93]. We utilize classical computing to spread quantum addition and multiplication over several quantum circuits by employing Residue Number Systems (RNS). RNS expresses numbers as residues or remainders obtained after division by a set of relatively prime moduli. This representation is distinct as long as the numbers remain within the product range of the moduli. Each residue can be added or multiplied separately without any carry dependency on other residues in the RNS collection. This enables the operation to be distributed across multiple quantum computers or tasks.

Our method of employing the Residue Number System (RNS) for quantum addition and multiplication divides the quantum addition task into several circuits, each requiring fewer qubits and having reduced depth, which can be executed independently in a Distributed Quantum Computing (DQC) framework. For distributed quantum addition, we show that our approach results in a higher output probability, due to increased noise resilience in Noisy Intermediate-Scale Quantum (NISQ) computing, by performing noise based simulations in Quantinuum's H1 ion-trap based quantum computer on PyTKET [94]. In order to maintain a fair comparison, we compare the distributed and non-distributed quantum addition circuits using ripple carry based architecture. For non-distributed quantum addition we use TPL13 quantum full adder circuit, that utilizes ripple carry architecture but uses fewer qubits and quantum resources that other adders with same architecture.

This Chapter is organized as follows: Section 5.1 discusses the background for RNS and DQC in depth. Section 5.2 introduces the algorithm and circuits for

proposed RNS based distributed quantum addition, and Section 5.3 shows their resource comparison and performance under noise simulations. Section 5.4 proposes RNS based distributed quantum multiplication and resource consumption. Section 5.5 concludes the Chapter with critical observations.

5.1 Background

5.1.1 Residue Number System (RNS)

The Residue Number System (RNS) is a numerical framework in which integers are expressed as their residues taken with respect to a moduli set. The moduli set consists of mutually prime integers that provide a set of unique residues or remainders from the division. RNS enables rapid arithmetic operations, for addition, subtraction, and multiplication, as their residues can be processed in parallel and independently without carry dependencies among them. RNS can be expanded for larger numerical ranges by selecting sufficiently large relatively prime moduli. Typical moduli used in RNS include small prime numbers and powers of two. In this work, the chosen modulo set is (2^n-1, 2^n, 2^n+1), which can represent conventional numbers almost up to 2^{3n}. The main advantage of this set is that it can scale up with higher values of 'n' while its constituent moduli remain prime relative to each other. Equation 5.1 defines the range of the RNS as the product of the moduli set, where M is the number of moduli, and the moduli set is R_0, R_1, .., R_M. The efficiency of the RNS set is defined as the ratio of the range to K, the largest number intended to be represented, as demonstrated in Equation 5.2. K can assist in comparing different RNS sets.

$$Range = \prod_{1}^{M} R \tag{5.1}$$

$$Efficiency = \frac{Range}{K} \quad \text{where } Range \leq K \tag{5.2}$$

In classical computing, RNS is utilized in numerous applications, including digital image processing, digital signal processing, cryptography, neural networks, and fault-tolerant computing systems [17]. In quantum computing, RNS has the potential to enable larger and faster arithmetic operations, addressing challenges such as qubit count and coherence time. RNS has improved resource estimates for quantum factorization and its use in factoring 2048-bit RSA integers [95, 96]. RNS has the potential to make quantum computers more reliable by use of redundant residues in error detection or correction [97].

5.1.2 Distributed Quantum Computing

Distributed Quantum Computing (DQC) is an emerging paradigm of performing quantum computations across multiple interconnected quantum computing devices or nodes, also known as Quantum Processing Units (QPU). The goal is to leverage the collective power of multiple QPU's to solve complex computational problems that cannot be efficiently solved by a single QPU. DQC aims to overcome the limitations of individual quantum devices, such as limited qubit counts and high error rates, and to enable the scaling of quantum computations to larger problem sizes [93].

The pursuit of DQC involves various approaches, primarily categorized into hardware and software solutions. Hardware solutions, like quantum teleportation enabled using gate or qubit teleportation, necessitate efficient quantum networking schemes for the transmission of quantum states across QPUs, and are a vibrant area of research. On the other hand, software approaches, such as circuit cutting, distribute parts of circuits among QPUs without the need for a fully realized quantum network. However, circuit cutting faces significant challenges, including the exponential increase in cost with the entanglement between parts, the management of different noise profiles, and the reconstruction of output [92].

Figure 5.1 shows our approach of using a Residue Number System (RNS) based Distributed Quantum Computing (DQC), that converts quantum addition into

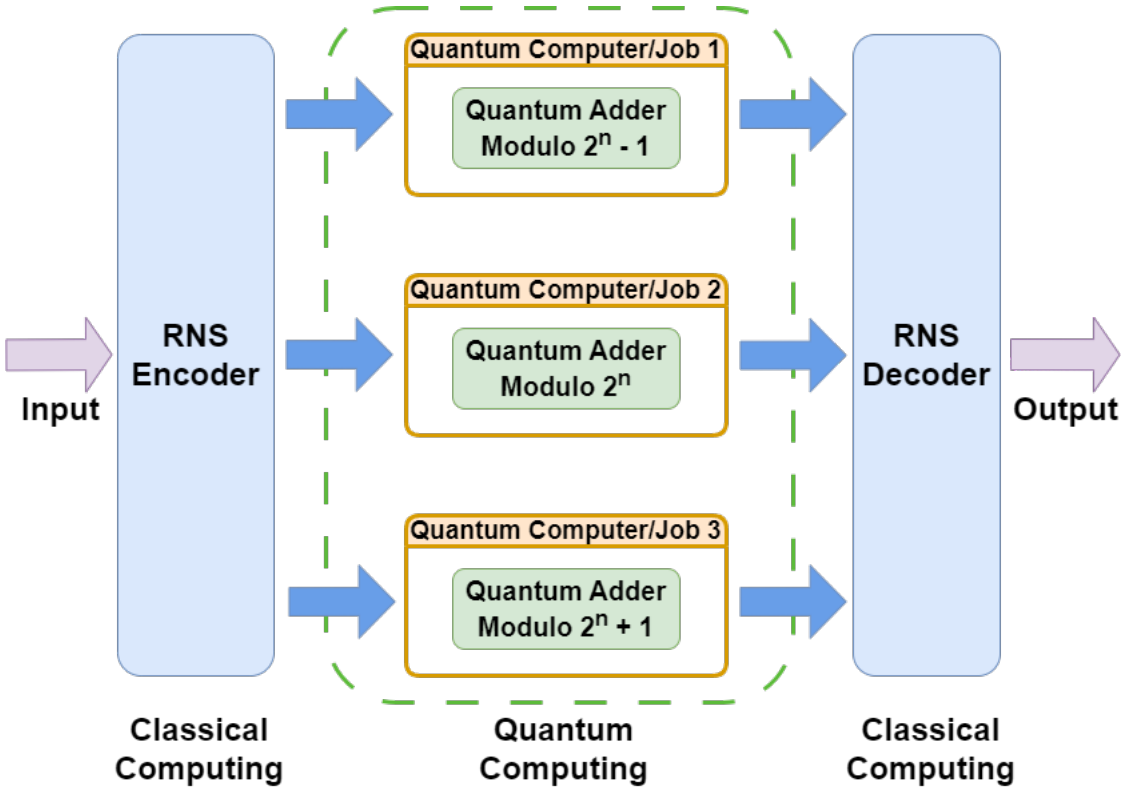


Figure 5.1: Residue Number System (RNS) based Distributed Quantum Addition (DQA) involves moduli set selection and Quantum Modulo Adders (QMA) generation during encoding, that can be executed in parallel without carry dependency across multiple quantum computers or jobs [55].

independent quantum modulo addition circuits across multiple QPUs or multiple jobs on a single QPU. We do not need complex networking infrastructure, like in hardware-based DQC approaches. Also, unlike circuit cutting, there is no dependency among different quantum modulo addition circuits of initializing the input of one circuit based on the output of another circuit.

5.2 Proposed RNS based Distributed Quantum Addition

In this Section, we propose our scheme that generates Residue Number System (RNS) based quantum modulo adders given input constraints such as the range of numbers to be represented (0, K-1) and the desired minimum efficiency of the RNS set (E).

5.2.1 Algorithm

As shown in Algorithm 3, we start with a default moduli count (C) three, as it corresponds to our starting RNS moduli set of (2^n-1, 2^n, 2^n+1). The advantage of this set is the scaling it provides in representing the range of numbers up to 2^{3n}. For instance, n = 2 yields a moduli set (3, 4, 5) that can represent a range up to (3x4x5) 60, close to (2^{3x2}) 64. Since the last four numbers cannot be represented, the efficiency of this RNS moduli set is 93.75%. However, the algorithm can be pushed to limits by choosing E = 1.0, leading to the choice of a moduli set that yields higher ranges.

For the range of numbers that do not correspond to 2^{3n}, we select other moduli created using one of the moduli from 2^n-1, 2^n or 2^n+1. For instance, when representing (2^8) 256, we choose the set (5, 8, 9) as it can provide us with the desired range using relatively prime moduli although derived using different values of 'n.' Moduli (2^2+1) 5, (2^3) 8, and (2^3+1) 9 can represent a range up to (5x8x9) 360, that is much higher than the needed 256. Although we could have chosen moduli 7 instead of 9, as the set

Algorithm 3 : Proposed RNS based Distributed Quantum Addition

Inputs: 1. Number K to be represented (K $\geq$ 50).
2. Moduli count C (default value 3).
3. RNS efficiency E (default value 0.9).
4. Quantum modulo adder set: (2^n-1, 2^n, 2^n+1)

Output: RNS set (R_1, R_2, R_3) to be created.

1. If K $\approx 2^{3h}$, where h $\geq$ 2. Select R where:
$$R = (2^h\text{-}1,\ 2^h,\ 2^h\text{+}1)$$
2. Return R if $R_1 * R_2 * R_3 \geq E * K$. Else go to Step 3.
3. Select lowest moduli from the moduli set such that:
$$\prod_1^C R \geq E * K$$
4. If cannot satisfy Step 1, increment C and repeat Step 1.
5. Determine the Toffoli depth for quantum adders for each moduli in set R.
6. For moduli with maximum Toffoli depth, check alternate moduli with lower Toffoli depth that meets constraint E.
7. Return R once cannot optimize any further in Step 3.

(5x8x7) would have fulfilled all requirements, we avoided it as the depth of quantum modulo 7 adder is higher than quantum modulo 9 adder by three Toffoli gates.

5.2.2 Constituent Modulo Adders

The three different types of quantum modulo adders used in this work are (2^n-1, 2^n or 2^n+1). While quantum modulo (2^n-1) and modulo (2^n+1) have been discussed in previous Chapters, we discuss the quantum modulo 2^n adders in this Chapter. Figure 5.2 shows designs for the quantum modulo 2^n for n = [2, 3], that are utilized in this Chapter. A quantum full adder with maximum output 'x' can be converted into modulo '$\frac{x}{2}$' by removing the gates responsible for generating carryout, resulting in equal number of qubits in output as input. We create quantum modulo 8 and higher sized modulo 2^n adders using this technique, while the quantum modulo 4 adder has a very simple and optimized design.

5.3 Resource Comparison and Noise Analysis of RNS based Distributed Quantum Addition

We use an emulator from Quantinuum to run simulations based on their 20-qubit H1 quantum computer using the PyTKET platform [70, 98]. These simulations involve the TPL13 based quantum full adder for outputs ranging from 6 bits to 10 bits, as well as quantum modulo adders with moduli ranging from 2 to 9 specifically for the sets (2^n-1, 2^n, and 2^n+1). The circuits for these designs are initialized in the $|0\rangle$ and $|1\rangle$ basis states. We perform the simulations of quantum modulo adders with a frequency of one hundred shots, while the TPL13 based quantum full adders are tested with two hundred shots. To determine the output probability, we calculate the ratio of the frequency of correct outputs to the total frequency and average it across all input combinations.

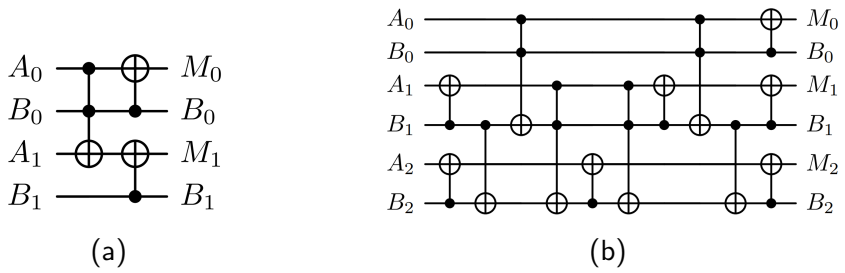


Figure 5.2: Quantum Modulo 2^n adders used in this work: (a) Mod 4 Adder. (b) Mod 8 Adder. The inputs are A and B. The output M is Modulo Sum, while input B is passed unchanged.

5.3.1 Quantum Modulo Adders

Table 5.1 presents the output probability and the utilization of quantum resources for quantum modulo adders. These include moduli (2^n-1, 2^n, 2^n+1), which facilitate RNS based quantum addition in this study. The main finding is the direct relationship between output probability and Toffoli depth. The output probability steadily decreases as the Toffoli depth rises from modulo 2 to modulo 9. However, this trend occurs at varying rates among the adders of the three different types.

Quantum adders for modulo 2^n exhibit the highest output probability among all moduli due to their optimal configuration. In the case of a modulo 2 quantum adder, just one CNOT gate is required. As shown in Figure 5.2, implementing a modulo 4 quantum adder necessitates only a single Toffoli gate alongside two CNOT gates. In contrast, the modulo 8 quantum adder employs the TPL13 quantum full adder by omitting the gates and qubits tied to carry generation. Our optimization for converting to modulo 2^n lowers both the Toffoli count and depth from five to three, marking a 40% reduction. Likewise, the number of CNOT gates also decrease by 40%, from ten to six, and the CNOT depth is reduced from seven to four, reflecting a decrease of 42.85%.

The Quantum Diminished-1 Modulo (2^n + 1) Adders (QDMA) represent the next best cohort based on output probability in Table 5.1. They require fewer Toffoli and CNOT gates in comparison to Quantum Modulo (2^n - 1) Adders (QMA), though they have a greater qubit count, resulting in reduced depth and enhanced output probability. This can be observed from the data for quantum modulo 3 adders shown in Table 5.1, where the QDMA design displays a 33% reduction in Toffoli depth and a 71.4% decrease in CNOT depth when compared with the QMA design, leading to a 5.7% increase in output probability. For n=3, the QDMA based modulo 9 adder exhibits a 25% reduction in Toffoli depth and a 30% decrease in CNOT depth compared to the QMA based modulo 7 adder while still achieving a 3.27% higher output probability.

Table 5.1: Comparison of Quantum Moduli Adders based on Quantum Resource Usage and Output Probability.

Modulo Adder		**Qubit**	**Depth**		**Count**		**Output**
Mod	**Type**		**Toffoli**	**CNOT**	**Toffoli**	**CNOT**	**Prob.**
2	2^n	2	0	1	0	1	0.995
3	2^n-1	7	6	7	8	8	0.912
3	2^n+1	8	4	2	5	2	0.964
4	2^n	4	1	1	1	2	0.985
5	2^n+1	11	6	5	8	7	0.931
7	2^n-1	10	12	10	14	12	0.865
8	2^n	6	3	4	3	6	0.966
9	2^n+1	14	9	7	11	13	0.893

5.3.2 Noise-Resilience

Tables 5.2 and 5.3 illustrate a comparison between quantum addition performed using TPL13 based non-distributed quantum full addition and the RNS based distributed quantum addition, based on noise-based simulations of the Quantinuum H1 quantum computer. For each size of the TPL13 adder, the corresponding RNS set (with a default efficiency setting of 90%) is presented. In Table 5.3, RNS efficiency represents the percentage of the range that the RNS moduli set covers. For RNS, the maximum qubits or depth refers to the highest qubit count or depth among the quantum modulo adder circuits within the RNS set. Likewise, the output probability of the RNS set indicates the lowest output probability found across the quantum modulo adder circuits that make up the RNS set. Finally, we provide the percentage increase in output probability that the RNS set achieves compared to the respective TPL13-based quantum full adder.

Table 5.2 depicts a definitive trend indicating an increase in output probability gain favoring distributed quantum addition (utilizing RNS) over non-distributed quantum addition (based on TPL13). When comparing the TPL13 adder with a 5-qubit input size, the corresponding RNS set (3, 4, 5) achieves a reduction of three gates in Toffoli depth and reduces CNOT depth to less than half, resulting in an 11.36% higher output probability. However, given that the effective range of the RNS set is 60 in contrast to 64 for TPL13, the efficiency of RNS stands at 93.75%. As we scale the adder output size from 6 to 10 qubits, the output probability drops from 0.836 to 0.371. However, for the corresponding RNS sets, the output probability declines from 0.931 to 0.865. The lower rate of decline for the RNS set originates from the distributed nature of parallel execution. For instance, in RNS sets (4, 5, 9) and (5, 8, 9) for output sizes of 7 and 8 qubit respectively, the only change is the use of modulo 8 adder instead of modulo 4 adder. It is evident from Table 5.1 that quantum modulo 8 adder utilizes greater resources than quantum modulo 4 adder. However, since the quantum modulo 8 adder consumes fewer resources than the other

Table 5.2: Output probability and resource usage for TPL13 based non-distributed quantum addition for various adder sizes.

Adder Size	**Qubit Count**	**Depth**		**Output Prob.**
		Toffoli	**CNOT**	
6	11	9	13	0.836
7	13	11	16	0.702
8	15	13	19	0.595
9	17	15	22	0.481
10	19	17	25	0.371
11	21	19	28	N.A.

Table 5.3: Output probability and resource usage for RNS based distributed quantum addition for various adder sizes.

Adder Size	**RNS Set**	**Eff. (%)**	**Max. Qubits**	**Max. Depth**		**Output Prob.**	**Gain w.r.t TPL13(%)**
				Toffoli	**CNOT**		
6	(3, 4, 5)	93.75	11	6	5	0.931	11.36
7	(4, 5, 9)	100	14	9	7	0.893	27.21
8	(5, 8, 9)	100	14	9	7	0.893	50.08
9	(7, 8, 9)	98.44	14	12	10	0.862	79.21
10	(4, 5, 7, 9)	100	14	12	10	0.865	133.15
11	(5, 7, 8, 9)	100	14	12	10	0.865	N.A.

two moduli in the RNS set, there is no change in the maximum qubit count, depth, or output probability. Hence, as shown in Figure 5.3, the advantage of RNS to run independent quantum jobs helps DQC to limit the loss of output probability and achieve greater noise resilience.

As implied in the final entries of Tables 5.3 and 5.2, we can achieve an addition similar to that of a quantum full adder with an output size of 11 or (2^{11}) 2048 by utilizing the RNS set (5, 7, 8, 9). However, since it requires 21 qubits, exceeding Quantinuum H1's capacity, it highlights the capability of RNS-based distributed quantum addition to perform calculations beyond the limits of NISQ-era quantum computers. To accomplish even greater ranges, we can strategically choose different moduli from the set (2^n-1, 2^n, 2^n+1) for various values of 'n', as illustrated in Table 5.1, helping manage quantum resource utilization efficiently.

5.4 Proposed RNS based Distributed Quantum Multiplication

As shown in Section 5.2, RNS has demonstrated the ability to perform distributed quantum addition [55]. Since RNS can also achieve closure for multiplication, we examine the impact of RNS-based distributed quantum multiplication on Toffoli depth and T gate count. Our approach combines classical computing with quantum computing to distribute multiplication across multiple quantum circuits using Residue Number Systems (RNS). We choose a set of three moduli (2^n-1, 2^n, and 2^n+1), that allows us to create scalable RNS sets with optimized quantum addition implementations that require lower Toffoli depth and T gate count. Although quantum modulo multipliers exist for the moduli 2^n-1 and 2^n, there was no optimized implementation for the quantum modulo multiplier 2^n+1. Hence, we propose a Quantum Diminished-1 Modulo (2^n+1) Multiplier (QDMM) [56].

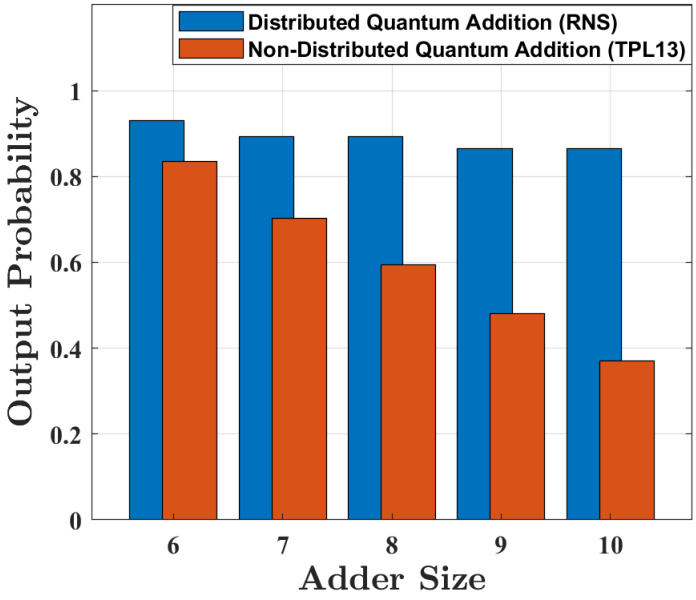


Figure 5.3: Comparison of Output Probability between TPL13 based non-distributed quantum full addition and RNS based distributed quantum addition for output sizes 6 to 10 qubit.

5.4.1 Proposed Quantum Diminished-1 Modulo $(2^n + 1)$ Multiplier

We utilize Algorithm 4 to construct the Quantum Diminished-1 Modulo $(2^n + 1)$ Multiplier (QDMM). The two inputs, $X = x_n\ x_{n-1}\ ...\ x_0$ and $Y = y_n\ y_{n-1}\ ...\ y_0$, are (n+1) qubit integers, and their modulo product, $P = X * Y = P_{n+1}\ P_n\ ...\ P_0$, is also an (n+1) qubit integer. Algorithm 4 uses a diminished-1 representation to simplify multiplication by a single qubit. To convert a standard integer to diminished-1, we subtract one in the range $(1, 2^n)$, with zero represented as 2^n. The algorithm optimizes modulo $(2^n + 1)$ multiplication using carry-save addition of partial products. It complements and anti-clockwise circular shifts intermediate quantities that exceed the $(n-1)^{th}$ qubit from left to right. Figure 5.4 shows the preprocessing and rearrangement of partial products for the (n-1) qubits of inputs X and Y.

These partial products are utilized in Algorithm 4 by passing through the Quantum 3:2 Compressor shown in Figure 5.5, which uses one ancilla and provides intermediate Sum and Carry. The inputs are passed through without modification, and their qubits can be recovered at this stage if needed by reversible uncomputation using Toffoli gates. Figure 5.6 illustrates the proposed QDMM for the n=4 configuration, resulting in quantum modulo $(2^4 + 1 = 17)$ multiplication. The MSB of each intermediate Carry is inverted and cyclically left-shifted. Ultimately, the final Sum and Carry pair is computed using the Quantum Carry Propagate Adder (QCPA), while the output's Carry is added back into the Sum using half QCPA, yielding the first (n-1) qubits of Product P. To determine if the product is zero, we simply need to calculate the logical OR of the n^{th} qubit of both inputs. In such a case, we employ Toffoli gates to reset the remaining qubits of Product P to $|0\rangle$.

5.4.2 RNS based Distributed Quantum Multiplier

This Section outlines the quantum resource usage of quantum modulo multipliers used to build Residue Number System (RNS) based distributed quantum multipliers.

Algorithm 4 : Calculation of $(X * Y) mod(2^n + 1)$

Input: Inputs X, Y are in diminished-1 format. $n \geq 1$

Ensure: $0 \leq X, Y < 2^n + 1$

1. Generate partial products: $X_{\mathrm{i}} Y_{\mathrm{j}} \, \forall i \in [0, 2], j \in [0, n-1]$
2. Complement 'i' most significant partial products and perform a left circular shift for them.
3. Use Quantum 3:2 Compressor to generate the Sum and Carry.
4. For $(i = 3, i{+}{+}, i \leq n)$:

A. Generate partial products: $X_{\mathrm{i}} Y_{\mathrm{j}} \, \forall \, j \in [0, n-1]$

B. Complement 'i' most significant partial products and perform a left circular shift for them.

C. Complement the MSB of Carry of previous stage and perform a left circular shift.

D. Add the Sum and Carry generated in (i-1)$^{\mathrm{th}}$ iteration along with 'i' partial products generated in Step A using Quantum 3:2 Compressor.

5. Complement the MSB of Carry and perform a left circular shift.
6. Add the final Sum and Carry using a carry propagate adder.
7. Complement the resulting Carry and add it back to the sum using a half carry propagate adder to generate $P_{\mathrm{i}} \, \forall \, i \in [0, n-1]$
8. $P_{\mathrm{n}} = X_{\mathrm{n}} \vee Y_{\mathrm{n}}$
9. If $P_{\mathrm{n}} = 1, P_{\mathrm{i}} = 0 \, \forall \, i \in [0, n-1]$

	Y_3	Y_2	Y_1	Y_0
x	X_3	X_2	X_1	X_0
	X_0Y_3	X_0Y_2	X_0Y_1	X_0Y_0
	X_1Y_2	X_1Y_1	X_1Y_0	$(X_1Y_3)'$
	X_2Y_1	X_2Y_0	$(X_2Y_3)'$	$(X_2Y_2)'$
	X_3Y_2	$(X_3Y_3)'$	$(X_3Y_2)'$	$(X_3Y_1)'$
	X_3	X_2	X_1	X_0
	Y_3	Y_2	Y_1	Y_0

Figure 5.4: Partial product arrangement for proposed Quantum Diminished-1 Modulo (2^{n}+1) Multiplier (QDMM).

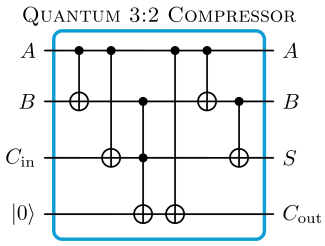


Figure 5.5: Quantum 3:2 compressor that conducts a single bit full addition for inputs A, B, C_{in}, and outputs Sum and C_{out} [56].

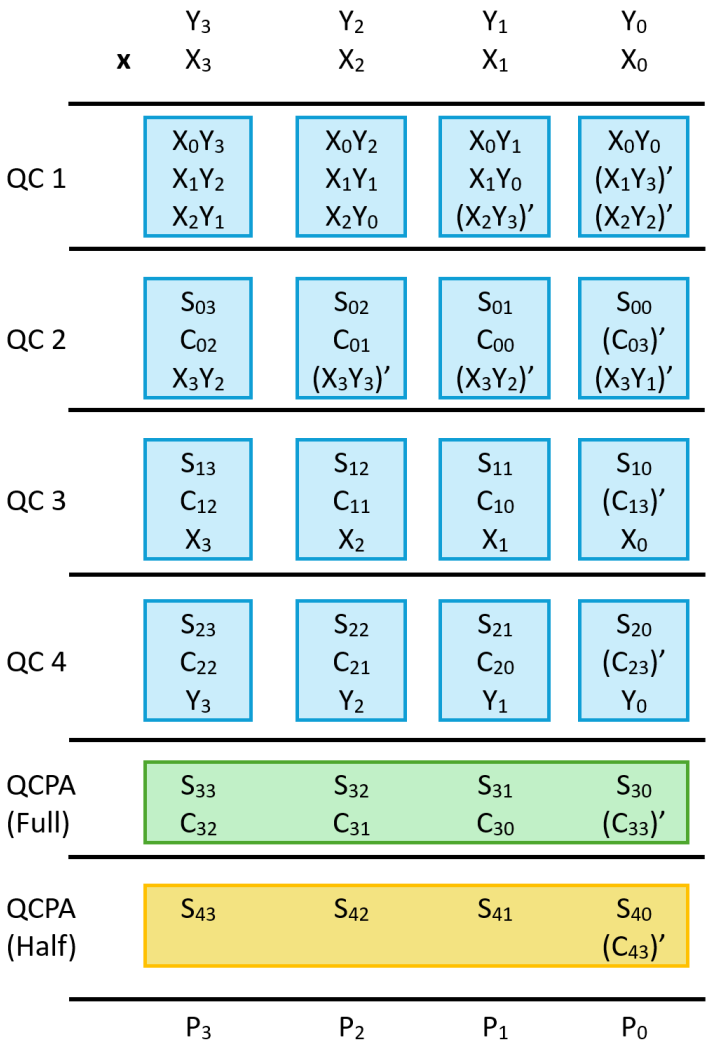


Figure 5.6: Architecture of proposed Quantum Diminished-1 Modulo (2^n+1) Multiplier (QDMM) for (n-1) qubits [56].

Table 5.4 breaks down the qubit count, Toffoli and CNOT gate count, and depth of different quantum multiplier designs. From Table 5.4, the T gate count can be calculated as 7 times the Toffoli count. Section 5.4.1 outlined our proposed algorithm and construction method for the Quantum Diminished-1 Modulo (2^n+1) Multiplier (QDMM).

For quantum modulo (2^n-1) and (2^n) multipliers, we relied on the designs presented by Cho et al [99]. These quantum modulo multipliers use O(log n) depth Quantum Carry Look Ahead (QCLA) adders, first introduced by Draper et al [24]. However, the work by Cho et al. didn't fully account for O(log n) depth, only providing basic complexity based quantities. This required us to recalculate their resource usage estimates.

For comparison with existing quantum multipliers, we selected the T gate count-optimized Quantum Multiplier by Muñoz-Coreas et al. [13]. This circuit uses a quantum conditional adder that requires very few ancilla and produces no garbage. However, its quantum adder is ripple carry-based, which is inherently less efficient than QCLA-based designs. Therefore, we provide estimates for a QCLA version of the Quantum Multiplier by Muñoz-Coreas et al. As shown in Table 5.4, its Toffoli depth complexity is O(n log n), compared to O(n^2) for the original ripple carry version.

5.4.3 Cost Analysis

Table 5.5 shows the quantum resource usage for the quantum modulo (2^n-1, 2^n, 2^n+1) multipliers, which are used in this work. The primary observation is that modulo (2^n) is the most optimal of the three designs. The proposed modulo (2^n + 1) design is better than modulo (2^n - 1) in terms of Toffoli count and depth but uses more CNOT gates and qubits. This is because it utilizes Quantum 3:2 Compressor as a building block while quantum modulo (2^n - 1) multiplier uses QCLA modulo (2^n - 1) adder which is garbage less and uses fewer CNOT gates. Table 5.6 shows the equivalent RNS sets for quantum multiplier of size 'n' whose output is of size '2n' and has a maximum

Table 5.4: Quantum Multiplier Resource Usage Comparison.

Quantum Multiplier	**Qubit**	**Toffoli**		**CNOT**	
		Count	**Depth**	**Count**	**Depth**
Proposed Modulo (2^n+1)	$2n^2+4n+2$	$2n^2+6n-1$	$2n^2+6n-2$	$10n^2+16n-6$	$10n^2+13n-1$
Modulo (2^n) [99]	$6n-2-w(n-1)-\lfloor\log(n-1)\rfloor$	$11n^2-22n+12-6(n-1)(w(n-1)+\lfloor\log(n-1)\rfloor)$	$n^2+12n-12+(n-1)(3\lfloor\log(n-1)\rfloor+\lfloor\log((n-1)/3)\rfloor)$	$4n^2-9n+5$	$4n-4$
Modulo (2^n-1) [99]	$6n-2$	$12n^2-22n+11$	$2n^2+7n-8+(n-1)(3\lfloor\log(n-1)\rfloor+\lfloor\log((n-1)/3)\rfloor)$	$4n^2-4n$	$4n-4$
Muñoz-Coreas et al (Original) [13]	$4n+1$	$3n^2-2$	$3n^2-2$	$5n^2-11n+6$	$3n^2-5n+2$
Muñoz-Coreas et al (QCLA version) [13]	$6n-w(n)-\lfloor\log n\rfloor-1$	$11n^2-15n+5-3(n-1)(w(n)+w(n-1)+\lfloor\log n\rfloor+\lfloor\log(n-1)\rfloor)$	$(n-1)(\lfloor\log n\rfloor+\lfloor\log(n-1)\rfloor+\lfloor\log n/3\rfloor+\lfloor\log((n-1)/3)\rfloor)+9n-8$	$4n^2-9n+5$	$4n-4$

Log is base 2, and w(n) represents number of 1's in binary expansion of n.

Table 5.5: Quantum Modulo Multiplier Cost Estimates.

n	Type	Mod	Qubit	Toffoli		CNOT	
				Count	Depth	Count	Depth
2	$2^n - 1$	3	10	15	12	8	4
2	2^n	4	9	6	14	3	4
2	$2^n + 1$	5	18	19	18	66	65
3	$2^n - 1$	7	16	53	35	24	8
3	2^n	8	14	21	37	14	8
3	$2^n + 1$	9	32	35	34	132	128
4	$2^n - 1$	15	22	115	61	48	12
4	2^n	16	19	46	61	33	12
4	$2^n + 1$	17	50	55	54	218	211

Table 5.6: Non-Distributed vs Distributed Quantum Multiplication Resource Consumption Comparison.

Input Size (n)	Max. Range	Out Size (2n)	Non-Distributed Quantum Multiplication (Munoz-Coreas et al (QCLA ver.)					Distributed Quantum Multiplication (RNS)						
			Qubit	Toffoli		CNOT		RNS Set	Range	Qubit	Toffoli		CNOT	
				Count	Depth	Count	Depth				Count	Depth	Count	Depth
3	49	6	14	29	21	14	8	(3, 4, 5)	59	18	19	18	66	65
4	225	8	20	67	37	33	12	(5, 8, 9)	359	32	35	37	132	128
5	961	10	25	121	53	60	16	(4, 5, 7, 9)	1260	32	53	35	132	128
6	3969	12	31	191	71	95	20	(5, 7, 9, 16)	5040	32	53	61	132	128
7	16129	14	36	277	91	138	24	(7, 9, 16, 17)	17136	50	55	61	218	211
8	65025	16	43	400	113	189	28	(5, 7, 9, 16, 17)	85680	50	55	61	218	211

range of $(2^n - 1)x(2^n - 1)$. The non-distributed quantum multiplication is represented by quantum multiplier of Munoz-Coreas et al (QCLA version) [13]. For RNS based distributed quantum multiplication, we display the maximum quantities among the constituent quantum modulo multipliers across the various quantum resources. This helps in making an objective comparison across the multiple quantum circuits that will be executed in parallel. Table 5.7 provides the improvement (or decrease) in Toffoli count, depth and T gate count. While Toffoli depth decreases upto 46.018%, the Toffoli count reduces from 34.483% to 86.25%, from 6 qubit to 16 qubit output size respectively. Since T gate usage is seven times in a fault tolerant implementation of Toffoli gate, its decline follows the same trend as that of Toffoli gate count [100].

5.5 Conclusion

In this work, we demonstrate the potential of achieving higher noise-resilience using Residue Number System (RNS) based distributed quantum addition by showing an improvement in output probability ranging from 11.36% to 133.15% when compared to non-distributed quantum addition for output sizes of six to ten qubits on the Quantinuum H1 noise simulator. Each parallel constituent circuit of RNS based distributed quantum addition utilizes fewer qubits and has a lower depth than non-distributed quantum addition. This results in non-distributed quantum addition hitting the 20-qubit count ceiling for an 11-qubit output size for Quantinuum H1 quantum computer. However, the distributed quantum addition is able to circumvent the circuit size limitation by using four moduli in parallel, each of which uses fewer qubits than the H1 machine's limit. This illustrates the feasibility of performing quantum addition for larger sizes that are not achievable by the Quantinuum H1 quantum computer using existing quantum adders.

Also, we conclude that the RNS based distributed quantum multiplication offers a promising solution to the scalability challenges of quantum arithmetic operations in Fault Tolerant Quantum (FTQ) computers. This is accomplished by achieving

Table 5.7: Comparative Analysis of Distributed Quantum Multiplication (RNS) with respect to Non-Distributed Quantum Multiplication.

Output Size (2n)	**Toffoli Count**		**Toffoli Depth**		**T Gate Count**	
	Impr.	**Impr. %**	**Impr.**	**Impr. %**	**Impr.**	**Impr. %**
6	10	34.483	3	14.286	70	34.483
8	32	47.761	0	0	224	47.761
10	68	56.198	18	33.962	476	56.198
12	138	72.251	10	14.085	966	72.251
14	222	80.144	30	32.967	1554	80.144
16	345	86.25	52	46.018	2415	86.25

substantial reductions in Toffoli depth, up to 46.018%, and decreasing T gate counts by 34.483% to 86.25% compared to existing non-distributed quantum multipliers for output sizes between 6 and 16 qubits. These improvements become more pronounced as input sizes increase, emphasizing the scalability advantages of the RNS approach. By distributing quantum modulo multiplication circuits across multiple quantum computers or jobs with lower resource requirements, this method addresses the fundamental limitations imposed by high Toffoli depth and T gate usage that have constrained the applicability of quantum multipliers in larger-scale quantum algorithms.

Downstream applications such as quantum cryptanalysis and quantum image and signal processing tasks stand to gain from the speed and parallelism provided by Residue Number Systems (RNS). We conclude that RNS possess significant potential to help implement Distributed Quantum Computing (DQC) in the current NISQ era of quantum computers by improving noise resilience. Further, they can contribute to Fault Tolerant Quantum Computing (FTQC) computers by enabling the independent execution of smaller depth quantum arithmetic circuits across different machines or jobs.

Chapter 6

Crosstalk Attack Resilience of Distributed Quantum Addition

Quantum computers promise considerable advantages over classical computing, but their susceptibility to various attacks, such as fault injection, privacy, and scheduling, poses significant security challenges. As quantum computers scale, the rise of multi-user and cloud-based quantum platforms can give rise to new security challenges. Attacks within shared execution environments are becoming increasingly feasible due to the crosstalk noise that, in combination with a quantum computer's hardware specifications, can be exploited in the form of a crosstalk attack.

Crosstalk noise is an inherent challenge in quantum computing and is a significant source of errors on quantum hardware of multiple types, like superconducting and ion-traps. Crosstalk attacks are particularly relevant in shared quantum computing environments, like cloud-based services, where multiple users execute circuits simultaneously. The method and mechanism of crosstalk attack depends upon the attack vector along with the quantum computing platform used. Different qubits types, such as superconducting and ion-trap, would need different strategies for the attack as well as the defense against crosstalk attacks. Hence, understanding and studying crosstalk noise is necessary to mitigate crosstalk attacks [36, 37, 38].

6.1 Quantum Crosstalk

Crosstalk refers to unintended interactions between qubits due to their physical proximity or quantum operations using shared hardware resources. It acts like a shared coupling that spreads unintended signals to adjacent qubits, causing a noise like behavior resulting in errors. Since crosstalk often manifests during two-qubit gates, it reduces gate fidelity and disrupts quantum circuit performance. Crosstalk noise can cause unintentional interactions between qubits that are not meant to be entangled, leading to false correlations. This compromises the purity and utility of intended entangled states. For entangled qubits, noise on one qubit due to crosstalk can propagate through the system, disrupting the entire quantum circuit and introducing errors that are difficult to correct [41, 42].

In superconducting qubits, crosstalk occurs due to coupling between qubits placed in close vicinity in the physical layout or through sharing of readout resonators. Unlike superconducting quantum computers with fixed physical layouts for qubits, ion-trap quantum computers operate by using lasers as optical tweezers to place ions (that act like qubits) next to each other in traps according to gate operation needs and other constraints. In ion-trap quantum computers, crosstalk arises when laser pulses for one qubit inadvertently affect ions that also happen to be present in the trap. While superconducting quantum computers have been demonstrated to be susceptible to crosstalk in multiple works, ion-trap quantum computers enjoy much lower crosstalk (in the order of -6) [41, 82, 101].

6.2 Quantum Crosstalk Attacks

Since crosstalk based attacks are side-channel attack, the chances of its success increase with concurrent gate operations. A shared execution environment, such as those offered by cloud-based quantum services, can allow multiple quantum circuits to run simultaneously by allowing different users to operate on distinct sets of qubits

within the same physical device. There are two major ways crosstalk can be used to mount attacks on multi-tenant quantum computers:

1. **Fault Injection**: Figure 6.1 shows how crosstalk enables adversaries to use two-qubit gates in their own circuits to inject faults into quantum circuits executed by other users in the form of interference. If crosstalk exceeds the threshold, or if the victim qubits are in close vicinity of the aggressors, this can lead to Denial-of-Service attacks or force circuits to produce incorrect results. Attacks using this mechanism have been demonstrated experimentally for superconducting NISQ-era IBM quantum computers [41].

2. **Qubit Sensing**: Crosstalk provides attacker with a side-channel to infer the quantum state of victim qubits. For example, in frequency-multiplexed superconducting qubits, attacker can use correlated readout errors to predict victim qubit states with high accuracy [102].

In single-user systems, all qubits are under the control of one entity, eliminating the potential for adversarial interference. Crosstalk noise may still exist, but it would only affect the user's own circuit rather than facilitating a deliberate attack. This situation may be handled using noise-aware quantum compilation by adjusting the mapping of the circuit qubits to physical qubits based on crosstalk.

6.3 Existing Mitigation Strategies

Defensive techniques against crosstalk attacks include physical qubit placement based strategies such as circuit separation, qubit allocation optimization, and introducing spectator qubits for monitoring. As shown in Table 6.1, these measures show varying levels of effectiveness in superconducting qubits, reducing crosstalk induced errors but not entirely eliminating the threat due to the inherent limitations of physical architecture. In ion-trap qubits, the effect of qubit placement based strategies is

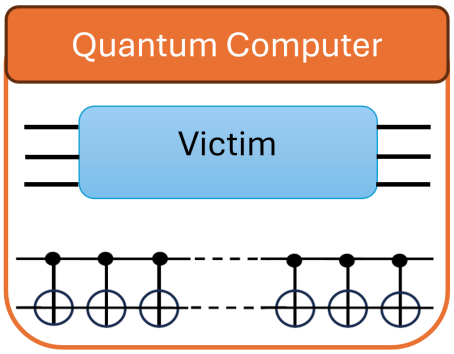


Figure 6.1: Crosstalk attack using CNOT chain in a multi-tenant quantum computing environment with multiple circuits running concurrently. Constant switching activity causes leakage of quantum state of attack vector in victim's circuit due to spurious entanglement.

Table 6.1: Summary of Crosstalk Attack Mitigation Strategies

Mitigation Strategy	Superconducting Qubits	Ion-trap Qubits
Circuit Separation	Effective: Physical separation of qubits reduces crosstalk [41, 42].	Not Effective: Physical placement of ions cannot be controlled [42].
Qubit Allocation Optimization	Effective: Reinforcement learning minimizes qubit overlap in high-crosstalk regions [41, 37].	Not Effective: Logical mapping can reduce overlap but physical placement is unknown [37].
Spectator Qubits	Effective: Extra qubits monitor and detect crosstalk early [41].	Not Effective: Introducing spectator qubits may increase noise in shared traps [103].
Crosstalk Aware Noise Models	Effective: Idle tomography and predictive modeling identify crosstalk behaviors [42, 37].	Partially Effective: Noise models can help understand crosstalk but limited by ion placement among traps[42, 37].
Hardware Level Improvements	Effective: Improved isolation, refined gate pulses, and better couplings reduce crosstalk [103].	Partially Effective: Laser shaping and targeted control offer modest improvements [42, 82].
Dynamical Decoupling (DD)	Effective: DD sequences suppress crosstalk and dephasing noise, improving performance [104].	Partially Effective: Requires precise synchronization due to tightly coupled nature of ion traps and is an active research area [105].

minimal. Other strategies include improvement in noise-resilience based strategies, that seek to improve noise models, the underlying hardware and use of noise suppressing techniques such as Dynamical Decoupling (DD). These techniques have demonstrated effectiveness for superconducting qubits, but are only partially effective in ion-trap quantum computers, as evident in Table 6.1. Hence, in following sections, we study the attack resilience of distributed quantum computing against crosstalk attacks, as existing mitigation techniques against crosstalk attacks for ion-trap qubits are either not effective or only partially so.

6.4 Proposed Crosstalk Attacks

As shown in Figure 6.2(a), existing crosstalk attack consists of a continuous array of CNOT gates, with the control qubit and a target qubit, initialized $|1\rangle$ and $|x\rangle$, respectively. Equation 6.1 shows target qubit switching between states $|x\rangle$ and $|\overline{x}\rangle$ generating crosstalk noise. Disadvantages of this attack are: 1) Only target qubit switches, making it more suitable for superconducting quantum computers having physical connectivity, to focus the attack on a particular victim qubit. 2) Possibility of easy detection by spectator qubit [41]. In ion-trap qubits that offer any-to-any connectivity, concentrating crosstalk only on single qubit can be a lesser effective strategy.

$$|\psi_{\text{Existing Attack}}\rangle = |1\rangle \otimes |(\sim)^n x\rangle \,, \text{where} \;\; n \geq 1 \tag{6.1}$$

In this section, we propose three novel crosstalk attacks effective in targeting ion-trap quantum computers with any-to-any connectivity.

6.4.1 Alternate CNOT (Proposed Attack 1)

Unlike existing attack having switching activity limited to one qubit, the proposed Alternate CNOT attack, referred as Attack 1, uses CNOT gates alternatively on both qubits for control/input. Figure 6.2(b) displays how NOT gates separate the CNOT

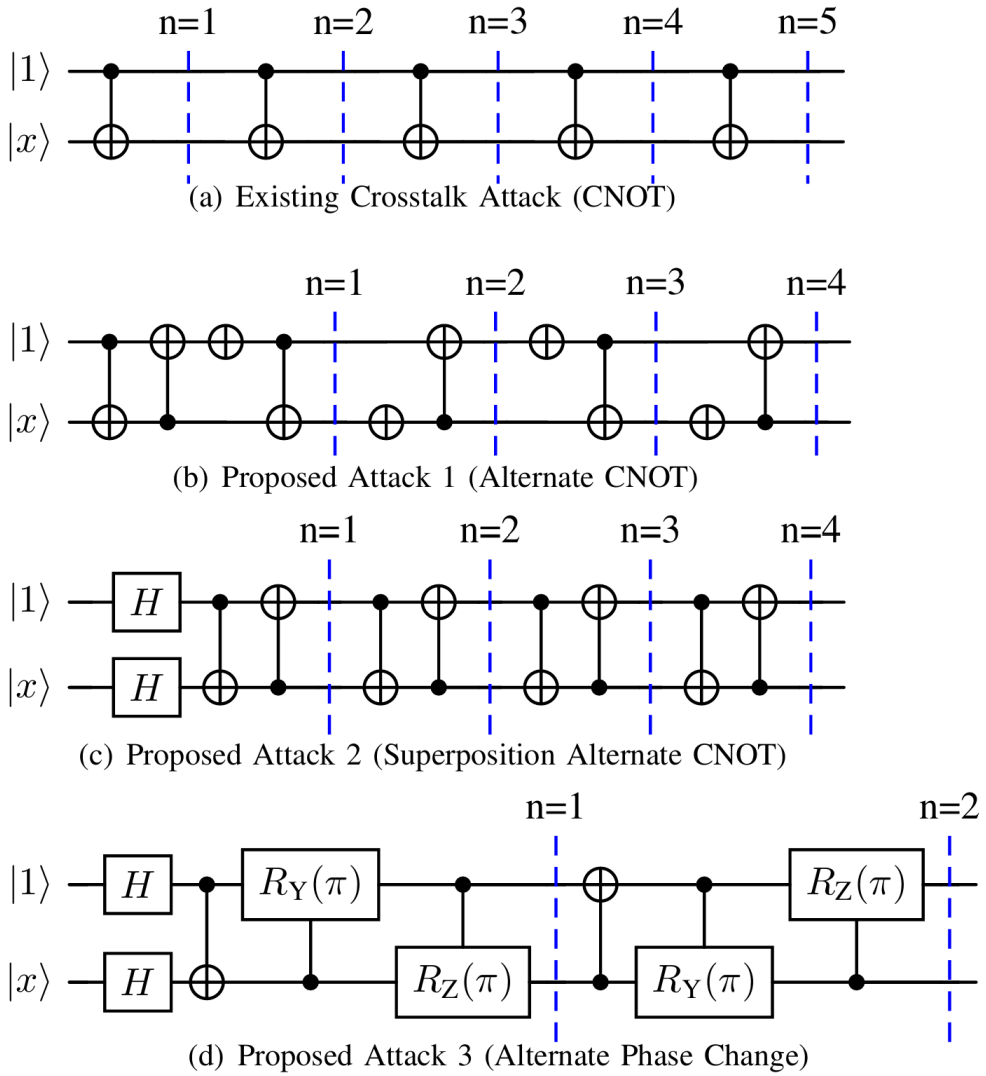


Figure 6.2: Quantum circuits for crosstalk attacks for input $|x\rangle$ with layer count n≥1 expandable with Victim's depth [57].

gates helping convert intermittent output $|0\rangle$ into $|1\rangle$, generating input for the next CNOT gate. As shown in Equation 6.2, Attack 1 leads to switching activity on both qubits, increasing the chances of crosstalk in ion-trap quantum computers as now any one of the attacking qubits can cause crosstalk. Here, $|x\rangle$ is input state and n is the layer count.

$$|\psi_{\text{Attack1}}\rangle = |(\sim)^{(n-1)}1\rangle \otimes |(\sim)^{(n-1)}x\rangle \text{ , where } \; n \geq 1 \tag{6.2}$$

6.4.2 Superposition Alternate CNOT (Proposed Attack 2)

While previously discussed attacks only utilized basis states, Superposition Alternate CNOT (SAC) attack or Attack 2, uses Hadamard gates on both qubits to introduce superposition. As evident from Figure 6.2(c), Attack 2 does not need NOT gates like Attack 1. Equation 6.3 shows the four mixed states that alternate among the phases 0 and π, introducing switching activity in form of phase differences across the four states.

$$|\psi_{\text{Attack2}}\rangle = \frac{1}{2}|00\rangle + \frac{(-1)^{(n+1+\lfloor\frac{n+2}{3}\rfloor)}}{2}|01\rangle + \frac{(-1)^{(n+\lfloor\frac{n+1}{3}\rfloor)}}{2}|10\rangle + \frac{(-1)^{(n-1+\lfloor\frac{n}{3}\rfloor)}}{2}|11\rangle \text{ , for } \; n \geq 1 \tag{6.3}$$

6.4.3 Alternate Phase Change (Proposed Attack 3)

Our final proposed Alternate Phase Change attack, referred to as Attack 3, utilizes a continuous series of flipped CNOT, Controlled-Y (CY), and Controlled-Z (CZ) gates shown in Figure 6.2(d) . Attack 3 causes the four mixed states to alternate between phases 0, $\pi/2$, π, and $3\pi/2$, leading to entanglement entropy alternating between 4.805e-16 and 2.73e-15, ultimately introducing more noise in the system [106]. Qubits in ion-trap quantum computers exist in form of ions driven in a shared trap for gate operations [82]. Higher entropy of attack qubits, if transferred to victim qubits due to crosstalk or leakage of control signal, can increase the effectiveness of the crosstalk

attack [101]. Equations 6.4 and 6.5 represent the four states for odd (1, 3, 5..) and even integers (2, 4, 6..) respectively.

$$|\psi_{\text{Attack3}}\rangle = \frac{|00\rangle}{2} + \frac{(-i)^{k+1}}{2}|01\rangle - \frac{i}{2}(-1)^k|10\rangle + \frac{(-i)^k}{2}|11\rangle$$

where k represents odd integer series (2n+1), $n \geq 0$ (6.4)

$$|\psi_{\text{Attack3}}\rangle = \frac{|00\rangle}{2} + \frac{(-1)^{j+1}}{2}|01\rangle - \frac{(i)^j}{2}|10\rangle - \frac{(-i)^j}{2}|11\rangle$$

where j represents even integer series (2n), $n \geq 1$ (6.5)

6.5 Crosstalk Attack Evaluation

In this section, we conduct crosstalk-based attacks to demonstrate the attack resilience of Distributed Quantum Addition (DQA) over Non Distributed Quantum Addition (NDQA) by utilizing emulator for Quantinuum H1, a 20-qubit ion-trap quantum computer. For NDQA, we conduct simulations using Quantum Full Adder (QFA) proposed by Thapliyal et al. due to its lower gate count and higher noise-resilience over other quantum adders [33, 25]. We perform simulations in 6-qubit to 9-qubit output range. For DQA, we conduct simulations for Quantum Modulo Adders (QMA) ranging from modulo 3 to modulo 9 output. These QMA can then be utilized to create Residue Number System (RNS) based distributed quantum addition for moduli set (2^n-1, 2^n, 2^n+1) [88, 55]. Since these circuits are initialized in $|0\rangle$ and $|1\rangle$ basis states, and provide non-mixed basis state as output, the output probability becomes the most definitive metric. Output probability refers to the probability of measuring a particular outcome in a quantum system after a measurement. We calculate the average of three input cases for each adder to determine the output probability. The first case is when both inputs A and B are zero. The second case is

when both A and B are equal to the maximum allowed input, which is (2^{n-1}-1) for a QFA with n-bit output and (k-1) for modulo k QMA. Finally, the third case is when A equals zero, and B receives the maximum allowed input. This strategy ensures fairness while evaluating adders of different types, such as QMA with QFA.

6.5.1 Attack Model

Attacker shares the ion-trap machine in a multi-tenant scenario with following attack models:

1. **Black Box Model:** The attacker has no access to details of the victim's quantum circuit and ends up dispatching CNOT chains of multiple widths and maximum length allowed as per the quantum volume of the machine.

2. **Grey Box Model:** The attacker has access to the victim circuit's width and depth, and uses that to customize the attack vector having depth slightly higher than the victim's circuit and a limited width that can help co-locate with victim. For instance, a 6-qubit output QFA (victim) utilizes eleven qubits, leaving space for an 8-qubit attack vector comprising of four CNOT chains. Also, measurements are performed only on victim and not on attacker to maintain focus on crosstalk generated error.

6.5.2 Effectiveness of Crosstalk Attacks on NDQA

Table 6.2 shows the resource usage comparison while Table 6.3 shows the output probability for TPL13-based Non Distributed Quantum Addition for output size ranging from 6 to 9 qubits. While it is evident that with increasing adder size from 6-qubit to 9-qubit output, the CNOT and Toffoli depth increases, and the output probability without attack reduces from 0.833 to 0.5, a drop of about 40%. The same trend is also evident for all attacks. For existing attack, the drop is 49.19%, while for attack 1, this decline is 45.86%.

Table 6.2: Resource Usage Comparison of NDQA with DQA.

Adder Size	Non Distributed Quantum Addition (NDQA)			Distributed Quantum Addition (DQA)			
	Qubit Count	Depth		RNS Set	Max. Qubits	Max. Depth	
		Toffoli	CNOT			Toffoli	CNOT
6	11	9	13	(3, 4, 5)	11	6	5
7	13	11	16	(4, 5, 9)	14	9	7
8	15	13	19	(5, 8, 9)	14	9	7
9	17	15	22	(7, 8, 9)	14	12	10

Table 6.3: Output Probability Comparison of Non-Distributed Quantum Addition (NDQA) with Distributed Quantum Addition (DQA).

Adder Size	Base Case (No Attack)			Existing Attack (CNOT)			Proposed Attack 1 (Alternate CNOT)			Proposed Attack 2 (SAC)			Proposed Attack 3 (APC)		
	NDQA	DQA	Impr. w.r.t NDQA (%)	NDQA	DQA	Impr. w.r.t NDQA (%)	NDQA	DQA	Impr. w.r.t NDQA (%)	NDQA	DQA	Impr. w.r.t NDQA (%)	NDQA	DQA	Impr. w.r.t NDQA (%)
6	0.833	0.94	12.8	0.74	0.92	24.3	0.713	0.927	30	0.673	0.927	37.7	0.653	0.911	39.5
7	0.653	0.911	39.5	0.62	0.878	41.6	0.567	0.822	45	0.473	0.811	71.5	0.44	0.8	81.8
8	0.616	0.911	47.9	0.46	0.878	90.9	0.382	0.822	115.2	0.361	0.811	124.7	0.356	0.8	124.7
9	0.5	0.893	78.6	0.376	0.878	133.5	0.386	0.822	113	0.362	0.811	124	0.345	0.8	131.9

For attacks 2 and 3, the decrease in output probability is 46.21% and 47.17% respectively. This trend shows the effectiveness of crosstalk attacks as the depth of victim circuit increases. Figure 6.3 shows the effectiveness of crosstalk attacks in form of reduction in output probability when the attacks are performed against NDQA on a per adder basis from 6-qubit to 9-qubit output size. For example, a drop of 10% in output probability would represent a 10% increase in attack effectiveness.

For the same adder size, Figure 6.3 shows the incremental effectiveness of attacks for each adder size. For instance, 7-qubit adder shows effectiveness of 5.054% due to existing attack. However, Attack 1 demonstrates effectiveness of 13.17%, an incremental increase of 8.116% over existing attack. Similarly Attack 2 and Attack 3 have 27.57% and 32.62% higher effectiveness respectively demonstrating incremental effectiveness of each attack. However, the same is not evident for adder 9-qubit adder, where the attack effectiveness remains range bound between 22.8% and 31%. The reason seems to be the higher width of 9-qubit adder at seventeen qubits, leaving only two qubits for a single attack vector. Whereas, the 7-qubit adder utilizes thirteen qubits leaving six qubits for three attack vectors, leading to more variation. However, higher depth of 9-qubit NDQA provides opportunity for higher effectiveness regardless of attack type.

6.5.3 Attack Resilient RNS based Distributed Quantum Addition

Residue Number System (RNS) encodes integers by calculating residues using a set of co-prime moduli. To this end, we use the Quantum Modulo Adders (QMA) from (2^n-1, 2^n, 2^n+1) to select candidates for constructing the RNS moduli set. Table 6.4 displays the resource usage, depth, and output probability of QMA from modulo 3 to modulo 9 output. The trend of decreasing output probability with increasing circuit depth is also observed in QMA, regardless of attack type. Modulo 2^n adders are least affected by attacks as they are resource efficient compared to other moduli. The

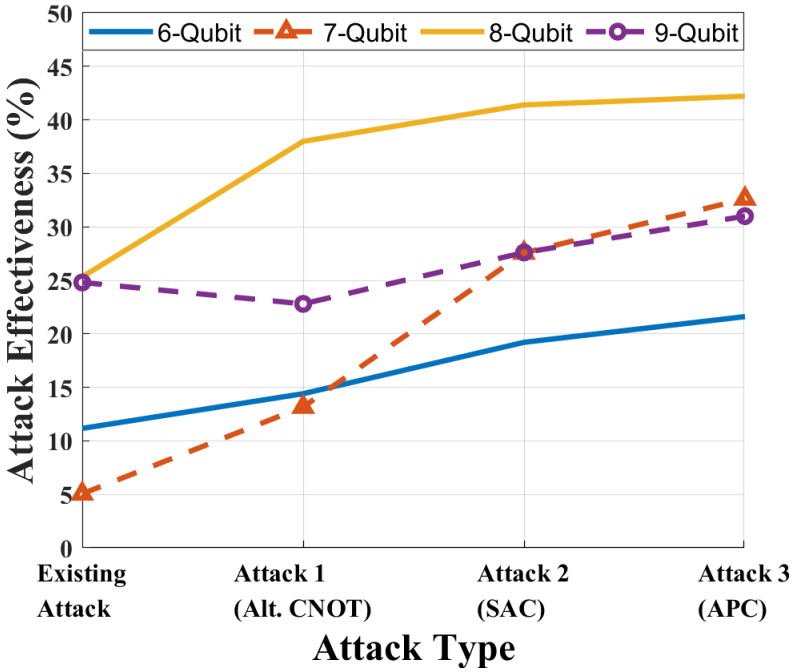


Figure 6.3: Attack effectiveness (%), calculated using output probability reduction of existing and proposed attacks over Non Distributed Quantum Addition (NDQA) for 6 to 9-qubit size [57].

Table 6.4: Comparison of Quantum Moduli Adders (QMA) based on Quantum Resource Usage and Output Probability.

Modulo Adder		Qubit	Depth		Count		Base Case	Existing	Attack 1	Attack 2	Attack 3
Mod	Type		Toffoli	CNOT	Toffoli	CNOT	(No Attack)	Attack	(Alt. CNOT)	(SAC)	(APC)
3	2^n-1	7	6	7	8	8	0.967	0.94	0.94	0.927	0.933
3	2^n+1	8	4	2	5	2	0.978	0.956	0.944	0.956	0.956
4	2^n	4	1	1	1	2	0.989	0.989	0.967	0.978	0.978
5	2^n+1	11	6	5	8	7	0.94	0.92	0.927	0.927	0.911
7	2^n-1	10	12	10	14	12	0.893	0.887	0.873	0.867	0.84
8	2^n	6	3	4	3	6	0.978	0.955	0.944	0.944	0.922
9	2^n+1	14	9	7	11	13	0.911	0.878	0.822	0.811	0.8

moduli adder with the highest Toffoli depth, modulo 7 of type (2^n-1) with a depth of twelve Toffoli gates, also has the lowest output probability of 0.893. Modulo 9 of type (2^n+1) with a depth of nine Toffoli follows with an output probability of 0.911. However, when simulated under attacks, the performance of modulo 7 is better than modulo 9. Modulo 7 output probability declines maximum up to 0.84 or 5.9% due to attack 4, while the output probability of modulo 9 drops by 12.2% from 0.911 to 0.8 due to attack 4.

Modulo 9 adder requires a 4-qubit output compared to modulo 7 adder's 3-qubit output, needing an extra measurement that increases susceptibility to attacks. The Table 6.3 demonstrates attack resilience in form of improvement in output probability of Residue Number System (RNS) based Distributed Quantum Addition (DQA) over Non Distributed Quantum Addition (NDQA) for sizes 6-qubit to 9-qubit output. As evident from Figure 6.4, the attack resilience of DQA increases as the adder size increases for all attacks. While for 6-qubit addition, the improvement from existing attack to attack 3 ranges from 24.3% to 39.5% respectively. For 8-qubit addition, improvement against same attacks ranges from 90.9% to 124.7%. This effectively demonstrates that as the size of quantum addition increases, the effect of attacks saturate in DQA as it is more depth efficient with fewer measurements per circuit. Also, for 9-qubit addition we notice a range bound improvement in attack resilience between 113% to 133.5%, as the effect of attacks saturate when victim occupies seventeen qubits on the commercial 20-qubit ion-trap quantum computer.

6.6 Conclusion

In this work, we determine that crosstalk attacks are a severe problem for ion-trap quantum computers. We discover that the use of superposition and phase changes can distribute switching activity more effectively across the CNOT chain, resulting in more effective crosstalk attacks. We propose three crosstalk attacks that are proven to reduce the output probability of Non Distributed Quantum

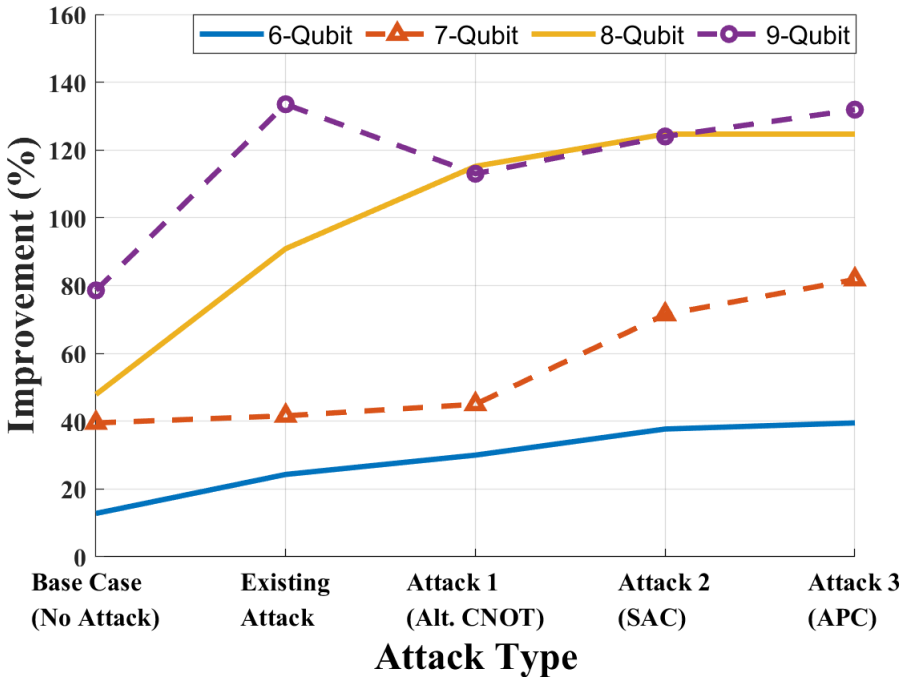


Figure 6.4: Improvement (%) Comparison of Output Probability between Non Distributed Quantum Addition (NDQA) and Distributed Quantum Addition (DQA) for output sizes 6 to 9 qubits without attack (base case), and ranging over existing and proposed crosstalk attacks [57].

Addition (NDQA) up to 42.2% using noise based simulations of a commercial 20-qubit ion-trap quantum computer. Finally, we show the attack resilience of Residue Number System (RNS) based Distributed Quantum Addition (DQA) over NDQA by showing improvement in output probability in the range of 24.3% and 133.5% when performing crosstalk attacks in a multi-tenant setting. Our findings not only expose security vulnerability in quantum full addition but also illustrate the potential of distributed quantum computing as a countermeasure against these attacks. Our work significantly enhances the understanding of crosstalk attacks and their mitigation, paving the way for more secure quantum computing systems.

Chapter 7

Noise-Resilient and Reduced Depth Approximate Quantum Adders

The quantum hardware available in the Noisy Intermediate Scale Quantum (NISQ) era is inadequate for extended operations due to the accumulation of errors [27]. Consequently, the focus is on managing noise and reducing errors by creating quantum circuits with reduced circuit depth and enhanced noise resilience. However, current quantum adders utilize a ripple carry approach, where the final carry relies on a lengthy carry path, thereby increasing the circuit depth and gate count, which in turn leads to a proportional increase in noise [23, 25]. In this Chapter, we delve into approximate computing, an innovative computing model that yields imprecise outcomes by easing the requirement for complete accuracy or deterministic operations. Hence, it is suitable for applications that can tolerate errors, such as data mining, multimedia, and image processing, that typically do not necessitate fully precise operations, as imprecise or suboptimal results are acceptable.

To the best of our knowledge, there are currently no available designs for uncontrolled (normal) quantum addition that apply approximate computing suitable

for the NISQ environment. Prior research has developed controlled quantum adders to develop approximate quantum multipliers, but they suffer from higher resource overhead and employ an uncomputation gate that necessitates measurement operations, restricting its use in the NISQ domain due to a higher susceptibility to noise from SPAM errors [14]. Thus, we introduce five innovative designs for quantum approximate adders that offer enhanced noise resilience and decreased circuit depth. This improvement is realized by eliminating the carry propagation dependency during the addition process, utilizing the concept of approximate computing. Because there is no carry dependency, each addition of the input qubits can occur simultaneously, leading to a reduction in the number of logic gates and depth, thereby enhancing noise resilience in comparison to precise quantum adder circuit designs.

Exact quantum adders developed by Cuccaro et al. [23], known as CQA1 (which includes output carry) and CQA0 (which does not include output carry), as well as TPL13 [25], an enhanced quantum ripple carry adder featuring output carry by Thapliyal et al., are well known existing designs of quantum ripple carry adders. Therefore, we assess our proposed approximate adders in comparison to these designs for comparison with earlier research. We demonstrate the enhanced noise resilience of our proposed quantum approximate adders through simulations using IBM Qiskit noise models. Additionally, we evaluate the error introduced by approximate computing in the proposed adders using error metrics like Normalized Mean Error Distance (NMED) and Error Rate (ER).

This Chapter is organized as follows: Section 7.1 presents the designs of proposed approximate adders. Section 7.2 performs the analysis of errors introduced due to approximation. Section 7.3 introduces noise models required to conduct noise-based simulations for comparison, while Section 7.4 compares the noise performance of proposed adders. Section 7.5 analyzes the performance of approximate quantum adders with carryout against quantum attacks. Section 7.6 concludes this work and most important contributions.

7.1 Design of Proposed Approximate Quantum Adders

In this section, we propose five approximate quantum adders to reduce the number of quantum gates and the depth to increase noise resilience. Existing ripple carry based exact quantum adders have a long carry path which introduces two issues, first it increases the noise susceptibility of carryout as it undergoes the long chain, and it also causes Least Significant Bits (LSB) qubits to have a higher idle time causing decoherence issues. We address these two root causes of noise in the design of existing quantum exact adders by utilizing approximate computing. First, we approximate the Sum output of the addition of two qubits using only its corresponding inputs. Our approximate adders operate in parallel, as there is no carry propagation during the addition operation. We do not generate intermediate carries and instead focus on creating the final carryout using only the MSB input qubits. These design techniques help generate all qubits of the Sum in parallel, avoiding the issue of idling qubits. In addition, the noise accumulated on carryout is lesser as there is no longer a carry path. Applying these design principles reduces the depth of proposed adders and increases their noise resilience.

Table 7.1 showcases the Sum and Carry logical equations for the exact quantum adders and the proposed quantum approximate adders. The proposed adders AQA1 and AQA2 are without Carry, whereas AQA3, AQA4, and AQA5 are with carryout. Table 7.1 also lists the CNOT and Toffoli gate count with their respective depth. The O(1) or constant depth complexity of proposed adders is superior to the O(n) or linear complexity of exact adders.

7.1.1 Proposed Adders Without Carryout

We first design adders without carryout for quantum applications that do not need an output carry.

Table 7.1: Design Characteristics of Exact and Proposed Approximate Adders

Adder Type	**Name**	**Sum**	**Carry**	**Qubits**	**Depth**		**Count**	
					CNOT	**Toffoli**	**CNOT**	**Toffoli**
Quantum Exact Adders	CQA0[23]	$A \oplus B \oplus C_{in}$	N.A.	$2n+1$	$3n+1$	$2n$	$4n$	$2n$
	CQA1[23]	$A \oplus B \oplus C_{in}$	$A \cdot B \oplus B \cdot C_{in} \oplus C_{in} \cdot A$	$2n+2$	$3n+2$	$2n$	$4n+1$	$2n$
	TPL13[25]	$A \oplus B \oplus C_i$	$A \cdot B \oplus B \cdot C_i \oplus C_i \cdot A$	$2n+1$	$3n-2$	$2n-1$	$5n-5$	$2n-1$
Proposed Adders Without Carryout	AQA1	A	N.A.	$2n$	0	0	0	0
	AQA2	$A \oplus B$	N.A.	$2n$	1	0	n	0
Proposed Adders With Carryout	AQA3	A	B_{n-1}	$2n$	0	0	0	0
	AQA4	$A \oplus B$	B_{n-1}	$2n$	1	0	n	0
	AQA5	$A \oplus B$	$A_{n-1} \cdot B_{n-1}$	$2n+1$	1	1	n	1

We only approximate the Sum for these adders, keeping the designs simple and with low depth.

1. **Approximate Quantum Adder 1 (AQA1):** The easiest way to approximate Sum is by using one of the inputs. In our first design AQA1, we map the input A to Sum and pass the input B unaltered, as shown in Equation 7.1. As evident from Figure 7.1(a), the adder is zero depth, and utilizes no quantum gates.

$$s_i = a_i \quad \text{if } 0 \leq i \leq n-1 \tag{7.1}$$

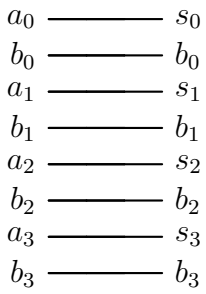
2. **Approximate Quantum Adder 2 (AQA2):** To create a better approximation than AQA1, we use logical-XOR of the two inputs as depicted in Equation 7.2. This design is also without carryout, has single depth, and has n CNOT gates for n-qubit input, evident from Figure 7.1(b).

$$s_i = a_i \oplus b_i \quad \text{if } 0 \leq i \leq n-1 \tag{7.2}$$

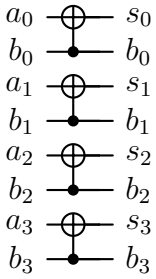
7.1.2 Proposed Adders With Carryout

In several applications, the output carry generated from the addition of two inputs is required. Therefore, to design the approximate adder with carryout we explore design strategies to generate it with low depth and minimal gates encountered in the carry path. We focus on generating the output carry using a constant complexity process, so that it can satisfy the above mentioned constraints and does not faces the same issues faced by a linear complexity output carry. In the process, we also improve their error metrics.

1. **Approximate Quantum Adder 3 (AQA3):** Our first design with an output carry AQA3, is derived from AQA1. The carryout is created from Input B's



(a) AQA1: Pass based
Zero Depth



(b) AQA2: CNOT based
Single Depth

Figure 7.1: Proposed quantum approximate adders without carryout in 4-qubit configuration. a) AQA1 needs no quantum gates to pass input A to Sum. b) Single CNOT gate layer in AQA2 is of depth O(1).

MSB as shown in Equation 7.3. Figure 7.2(a) illustrates that it has zero depth and zero gate count.

$$c_n = b_{n-1} \qquad s_i = \begin{cases} a_i, & \text{if } 0 \leq i \leq n-1 \\ c_n, & \text{if } i = n \end{cases} \tag{7.3}$$

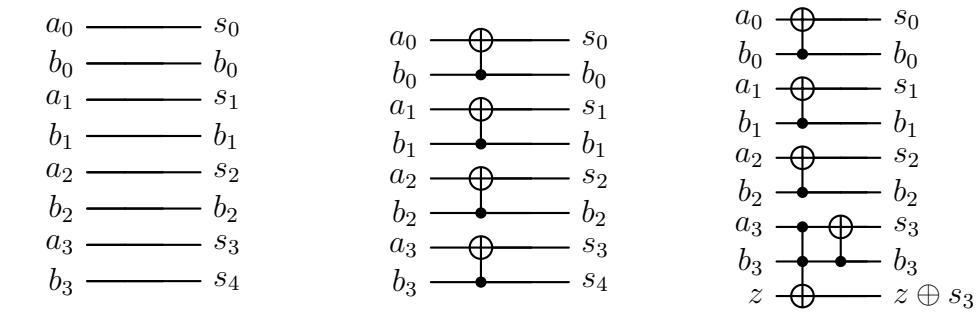
2. **Approximate Quantum Adder 4 (AQA4):** Like in the case of AQA3, we also design AQA4 using AQA2 by reassigning the last qubit of the input B as carryout, as shown in Equation 7.4. As observable from Figure 7.2(b), this approach helps ensure the depth remains constant with n CNOT gates for n inputs.

$$c_n = b_{n-1} \qquad s_i = \begin{cases} a_i \oplus b_i, & \text{if } 0 \leq i \leq n-1 \\ c_n, & \text{if } i = n \end{cases} \tag{7.4}$$

3. **Approximate Quantum Adder 5 (AQA5):** We next propose approximate adder AQA5, which generates carryout as $a_{n-1} \cdot b_{n-1}$ to provide a better approximation of the carryout, where a_{n-1} and b_{n-1} are the input MSB's. We use a Toffoli gate for this purpose as shown in Figure 7.2(c). AQA5 provides output carry on a separate qubit for quantum applications that require cascading a larger quantum circuit after the quantum adder. AQA5 shields the cascaded circuit from input noise sources and provides a high fidelity carryout while passing an input for downstream utilization.

$$c_n = a_{n-1} \cdot b_{n-1} \qquad s_i = \begin{cases} a_i \oplus b_i, & \text{if } 0 \leq i \leq n-1 \\ c_n, & \text{if } i = n \end{cases} \tag{7.5}$$

The proposed approximate adders reduce quantum gates to achieve zero or constant depth. Unlike exact adders, they generate parallel output on all Sum qubits, generating carryout using a shorter mechanism. Both these factors help



(a) AQA3: Pass based b_{n-1} carry

(b) AQA4: CNOT based b_{n-1} carry

(c) AQA5: Toffoli based $a_{n-1} \cdot b_{n-1}$ carry

Figure 7.2: Proposed Quantum Approximate Adders With Carryout in 4-qubit configuration.

improve the error profile and noise fidelity of proposed approximate adders, which we explore in Sections 7.2 and 7.4.

7.2 Error Analysis

The proposed adders are evaluated on the basis of degree and characteristics of error introduced by the approximation. We are also interested in understanding how the adders behave with scaling of input bit size. To achieve this, we use performance metrics such as Normalized Mean Error Distance (NMED) and Error Rate (ER). NMED measures the deviation of an approximate result from the exact result normalized to provide a relative measure that is independent of the absolute magnitude of the values being compared. ER provides the ratio of incorrect output within all outputs.

Figure 7.3 shows the error metrics for proposed adders without carryout. Both AQA1 and AQA2 demonstrate saturation in NMED and ER as the adder size increases. However, AQA2 has a lower NMED compared to AQA1, giving it an advantage in lower adder size, which diminishes as the size approaches 5-qubit. Similar trends are also witnessed in Figure 7.4, where AQA5 shows the relative advantage in NMED as well as ER. NMED for AQA3 merges with AQA5 after the 6-qubit size. AQA4, on the other hand, shows a relative advantage in terms of ER compared to AQA3. Another important observation is that the convergence point for proposed approximate adders with carry is 0.13, which is 56.67% lower than 0.3, at which proposed approximate adders without carryout stabilize. The lower NMED for higher sized adders demonstrates the importance of carryout.

7.3 Noise Models

Noise is the primary hindrance to the usage and scaling of quantum computing applications[58]. Section 2.4 discusses the causes and impact of the significant sources

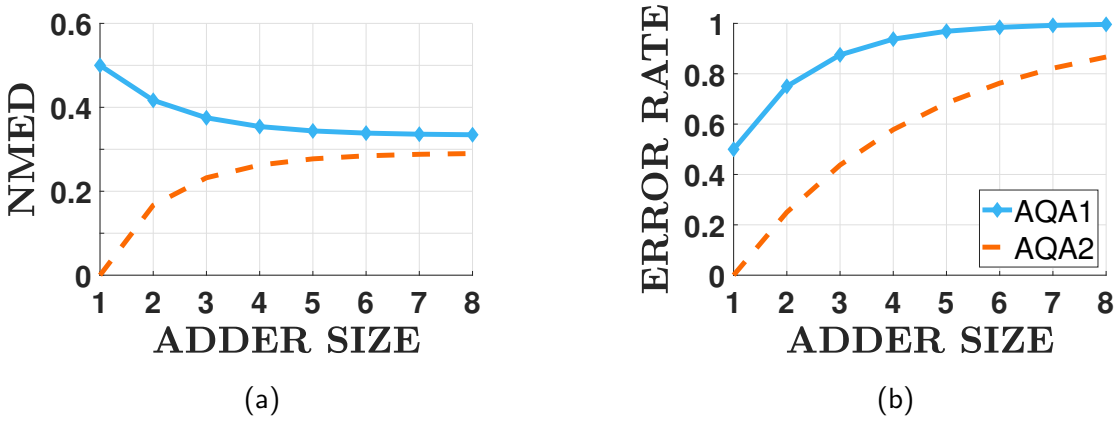


Figure 7.3: Error Metrics of Proposed Adders without Carryout for: (a) Normalized Mean Error Deviation. (b) Error Rate.

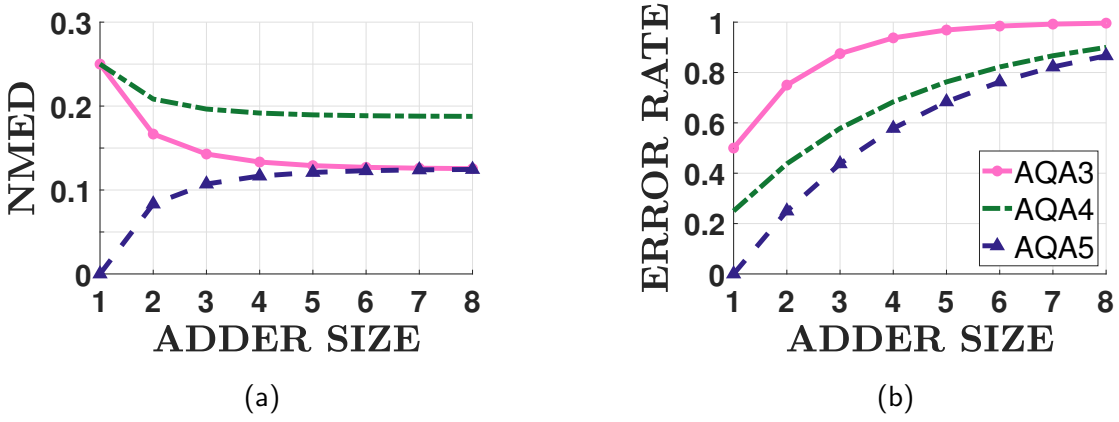


Figure 7.4: Error Metrics of Proposed Adders with Carryout for: (a) Normalized Mean Error Deviation (b) Error Rate.

of noise in quantum computers. In Section 2.5, we saw that while the majority of noise sources exist in all types of quantum computers, their quantity and impact vary and often depend upon the type of hardware and underlying technology involved. In this Chapter, we utilize the noise modeling capability of IBM quantum computers based on superconducting qubits. To achieve this, we utilize IBM Qiskit [9] commands to inject seven different types of noise into our simulations that are useful and can adequately represent the uncertainty faced by quantum addition circuits. Table 7.2 presents a compilation of the noise models employed in this Chapter.

7.4 Noise Resilience Analysis

To evaluate the noise resilience of proposed approximate quantum adders, we utilize the probability of obtaining the intended output to measure noise fidelity. To arrive at the output probability, we simulate the adders in a 4-qubit configuration using IBM Qiskit for all 256 input configurations. Then, we divide the number of accurate results by the simulation count (256 in this case). By modifying only the noise model in qiskit simulations and leaving the rest unchanged, we perform the same process for each adder using the different noise models. SPAM and readout noise depend on the measurement and input states making them proportional to the qubit count, impacting the exact and approximate adders in the same manner in our study. Due to this, we present the results of five noise models in Tables 7.3 and 7.4 for comparison. In the rest of this work, the average output probability is interchangeably referred to as accuracy (in percentage). To simplify the process of analyzing noise, we split the adders into two categories depending on the presence (or absence) of output carry.

To assess the noise resilience of the proposed approximate quantum adders, we measure noise fidelity by determining the probability of achieving the intended output. We calculate the output probability by simulating the adders in a 4-qubit configuration with IBM Qiskit across all 256 input combinations. Next, we compute the ratio of correct results to the total number of simulations (which is 256 in this

Table 7.2: Description of Noise Models

Model	Description
Thermal	Thermal relaxation error involves two time constants, with T1 (thermal relaxation time) $\leq$ T2 (dephasing time). Here, we assume T1 = 50μs and T2 = 70μs, as detailed in [9].
Depolarizing	The probability represents the chance that a completely mixed state $I/2$ replaces the state of a qubit. We set the error probability to 0.005 for single-qubit gates and 0.01 for two-qubit gates [58].
Amplitude Damping	This noise, caused by energy dissipation during operations, is assigned a probability of 0.01 [58].
Phase Damping	We utilize 0.01 probability for phase damping, which only affects the z-axis and corrupts phase and not amplitude of qubit.
SPAM	State Preparation And Measurement noise causes unreliable state initialization and measurement. In state preparation, we use the probability of resetting to 1 (0.04) twice that of resetting to 0 (0.02), while the probability of erroneous measurement is 0.1 [107].
Bitflip	We model this noise source as flipping the qubit between 0 and 1, with an error probability of 0.01 [9].
Readout	Readout noise affect the output measurement by probability P(x|y), which denotes the probability of measuring x while y is the correct output. In this work, P(1|0) = 0.05, P(0|1) = 0.1. [9]

Table 7.3: Output Probability of Proposed 4-Qubit Adders without Carryout from Noise Simulations

Noise Type	CQA0[23]	AQA1	Impr. % w.r.t. CQA0	AQA2	Impr. % w.r.t. CQA0
Thermal	0.712	0.951	33.57	0.935	31.32
Depolarizing	0.589	0.995	68.93	0.97	64.69
Phase	0.923	1.0	8.34	1.0	8.34
Amplitude	0.776	0.98	26.29	0.961	23.84
SPAM	0.657	0.656	-	0.656	-
Bitflip	0.307	0.98	219.22	0.924	200.98
Readout	0.733	0.732	-	0.73	-

Table 7.4: Output Probability of Proposed 4-Qubit Adders with Carryout from Noise Simulations

Noise Type	CQA1[23]	TPL13[25]	AQA3	AQA4	AQA5
Thermal	0.663	0.6956	0.953	0.926	0.904
Depolarizing	0.576	0.6292	0.994	0.968	0.917
Phase	0.924	0.9408	1.0	1.0	0.99
Amplitude	0.772	0.8139	0.975	0.961	0.94
SPAM	0.591	0.5901	0.591	0.59	0.591
Bitflip	0.207	0.263	0.975	0.915	0.814
Readout	0.678	0.6782	0.678	0.677	0.686

instance). By solely adjusting the noise model in the Qiskit simulations while keeping everything else constant, we repeat the same procedure for each adder using the various noise models. In the remainder of this section, the average output probability will be referred to interchangeably as accuracy (expressed as a percentage). To facilitate the analysis of noise, we categorize the adders into two types based on whether they produce an output carry or not:

7.4.1 Approximate Quantum Adders Without Carryout

Table 7.3 shows results for the proposed adders without carryout in 4-qubit configuration simulated with IBM Qiskit based noise models. While comparing with exact adder without carry CQA0[23], all noise models except SPAM and readout show improvement in output probability. SPAM and readout noise are influenced by the measurement and input states, which makes them proportional to the number of qubits. This affects both the exact and approximate adders in a similar way in our analysis and reaffirms the dependability of the noise based simulations.

Highest improvement of 200% is shown for bitflip noise by both AQA1 and AQA2. For thermal noise, AQA1 shows an improvement of 33.57%, while AQA2 demonstrates 31.32%. For depolarizing noise, AQA1 shows 68.93% improvement while AQA2 shows 4% lower than it. AQA1 shows about 3% higher accuracy improvement than AQA2 for amplitude damping, which shows 23.84%. Lowest accuracy improvement is shown for phase damping noise, for which AQA1 and AQA2 realize equal accuracy improvement of 8.24%. It is evident from these results that the noise resilience of AQA2 is only marginally affected by the presence of an extra CNOT gate depth layer.

7.4.2 Approximate Quantum Adders With Carryout

For the quantum adders with carryout mentioned in Table 7.4, we use CQA1[23] as the reference point to evaluate the proposed adders alongside the more optimized

TPL13[25]. As in the case of proposed adders without carryout, only readout and SPAM noise sources affect all adders under consideration with neglidgible difference. Due to this, we present the improvement comparison for other five noise models in Tables 7.5. The simulations show improvement with similar trends as in case of proposed adders without carryout.

Bitflip noise shows highest improvement percentage in the range of 371% to 209.51%. Depolarizing noise is the second highest gain in the range of 72.57% to 45.74%. This is closely followed by lead in accuracy improvement for proposed adders for thermal noise in the range of 43.74% to about 30%. Like in the case of proposed adders without carryout, amplitude and phase damping noise simulations take the last two places in improvement comparison. AQA3 leads the proposed adders in noise fidelity improvement as it is zero depth, compared to AQA4 with single depth. AQA5 features a Toffoli gate which lowers error deviation but introduces a higher noise impact.

Thermal and bitflip noise have a significant impact on exact adders due to their greater number of gates, which raises the chances of output errors. Depolarizing noise induces random Pauli operations along the three axes [58]. This type of error must accumulate before it influences the output, so it affects exact adders that have greater depth and carry paths where error can gather. Conversely, phase damping noise causes a rotation around the z-axis [58], but it has a negligible effect on adders since our input resides on the x-axis.

Amplitude damping noise correlates with the idle duration of the qubit [27], causing the most severe effects on exact adders using the ripple carry method. Since the least significant bit (LSB) in cuccaro exact adders experiences the longest wait for reverse computation, its output probability decreases in the amplitude damping noise model. In contrast, the suggested approximate quantum adders remain unaffected, as all qubits are processed simultaneously. This is due to the constant O(1) depth of the approximate quantum adders.

Table 7.5: Improvement Comparison of Proposed 4-Qubit Adders with Carryout with Existing Works

Adder	% Impr. w.r.t	Thermal(%)	Depolarizing(%)	Phase(%)	Amplitude(%)	Bitflip(%)
AQA3	**CQA1**[23]	43.74	72.57	8.23	26.3	371.01
	TPL13[25]	37	57.98	6.29	19.79	270.72
AQA4	**CQA1**[23]	39.67	68.06	8.23	24.48	342.03
	TPL13[25]	33.12	53.85	6.29	18.07	247.91
AQA5	**CQA1**[23]	36.35	59.2	7.14	21.76	293.24
	TPL13[25]	29.96	45.74	5.23	15.49	209.51

7.5 Security Analysis

Having established that approximate quantum adders exhibit noise resilience based on quantum simulations, we now proceed to investigate whether this resilience enhances their ability to mitigate known security vulnerabilities within quantum circuits. Of particular interest is a comparative analysis of various approximate adders with exact adders to understand how their performance and approximation quality are affected under attacks. We only compare approximate quantum adders with carryout to conserve valuable execution resources and directly compare them with existing exact adders. We use the Quantinuum emulator for the 20-qubit H1 quantum computer, which closely mimics the actual machine with high accuracy. Table 7.6 presents the attacks whose effects we aim to simulate. We now outline the methodology, purpose, and results comparison for each attack.

7.5.1 Stuck At 0/1

The Stuck At 0/1 attack highlights a core vulnerability in quantum computers where a target qubit becomes permanently fixed at either the $|0\rangle$ or $|1\rangle$ state, caused by faulty qubits or supporting hardware. We simulate this attack through a zero reset operation on the targeted qubit [38]. This attack shows behavior that depends on architecture, where targeting the lowest qubit affects only the least significant bit (LSB) in approximate adders, but causes errors to propagate to the most significant bit (MSB) in ripple carry-based exact adders. Therefore, we perform an attack on the middle qubit to remain neutral to the underlying architecture. Our implementation uses the Qiskit-based simulator instead of hardware platforms like the Quantinuum H1, focusing on error deviation rather than output probability, since the output completely changes in this attack scenario.

Experimental analysis shows in Figure 7.5 that exact quantum adders perform better and are more resistant to stuck at 0/1 attacks than approximate ones. Among approximate adders, architectural design greatly influences their attack resistance.

Table 7.6: Quantum attack types and metrics

Quantum Attack	Modus Operandi	Metric
Stuck At 0/1	Faulty qubit or hardware	NMED
State Preparation Attack	Pulse, Hardware, Circuit manipulation	Output prob.
Crosstalk Attack	Multi-tenant execution	Output prob.
Gate Attack	Pulse, Circuit manipulation	Output prob.

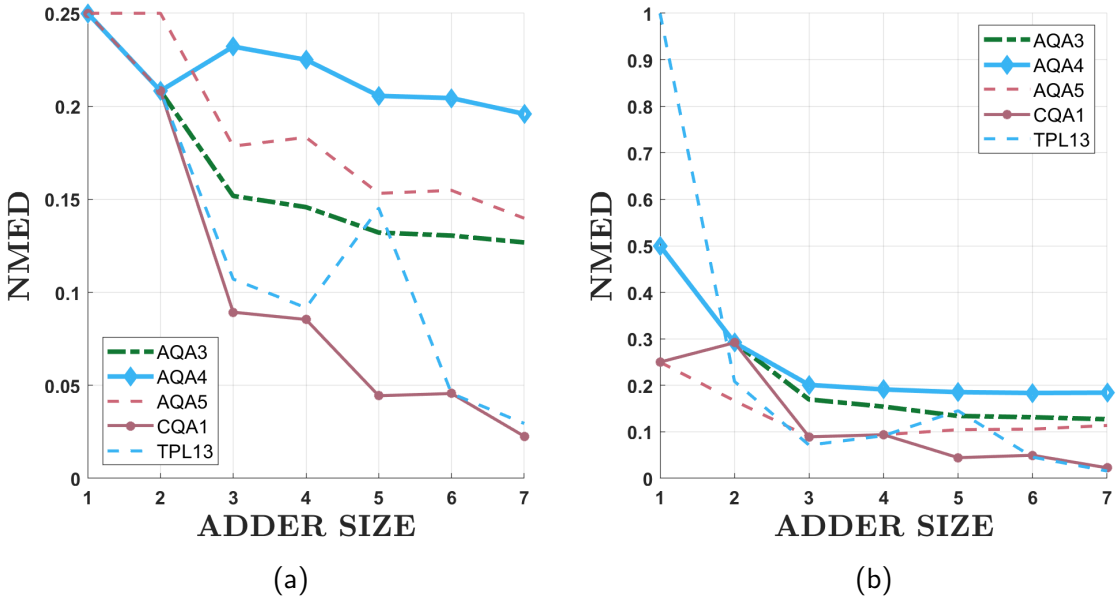


Figure 7.5: Normalized Mean Error Deviation (NMED) of proposed approximate quantum adders with carryout for: (a) Stuck at 0 attack. (b) Stuck at 1 attack.

Pass-based AQA3 adders perform better against stuck at 0 attacks, while Toffoli-based AQA5 adders show improved resistance to stuck at 1 attacks. Initially, stuck at 1 attack causes higher error values than stuck at 0 attack, but over time, both tend to similar performance levels.

7.5.2 Baseline Noise Simulation

The baseline noise analysis was carried out using the Quantinuum H1 system emulator, a 20-qubit quantum machine simulator, to evaluate the inherent noise resilience of different quantum adder architectures without external attack vectors. Testing was done across input sizes from 1-qubit to 9-qubit configurations, with output probability calculated as the ratio of shots producing the correct output to total shots taken. Our findings in Section 7.4 that approximate quantum adders are more noise-resilient than their exact counterparts are confirmed here as well using noise model based on actual quantum computers. Figure 7.6 (a) shows that exact quantum adders perform the worst under actual noise conditions, while approximate quantum adders consistently maintain output probabilities above 0.95 and exhibit only a slow decline in performance as circuit complexity increases.

7.5.3 State Preparation Attack

The State Preparation Attack targets vulnerabilities in the preparation of quantum states used as input by the quantum circuits. We conduct this attack by inserting Y-axis rotation gates on the qubits allotted for the second input using small rotation angles ($\frac{\pi}{8}$ radians or 22.5 degrees), which introduce consistent variability into the quantum state preparation process. This attack is especially effective because Ry rotations generate more variability compared to Rx and Rz rotations, and can be applied through various methods such as manipulation of control hardware, microwave pulses, or the quantum circuit itself. It can potentially remain undetectable to users since no visible changes are made to the circuit structure. Figure 7.6 (b) shows

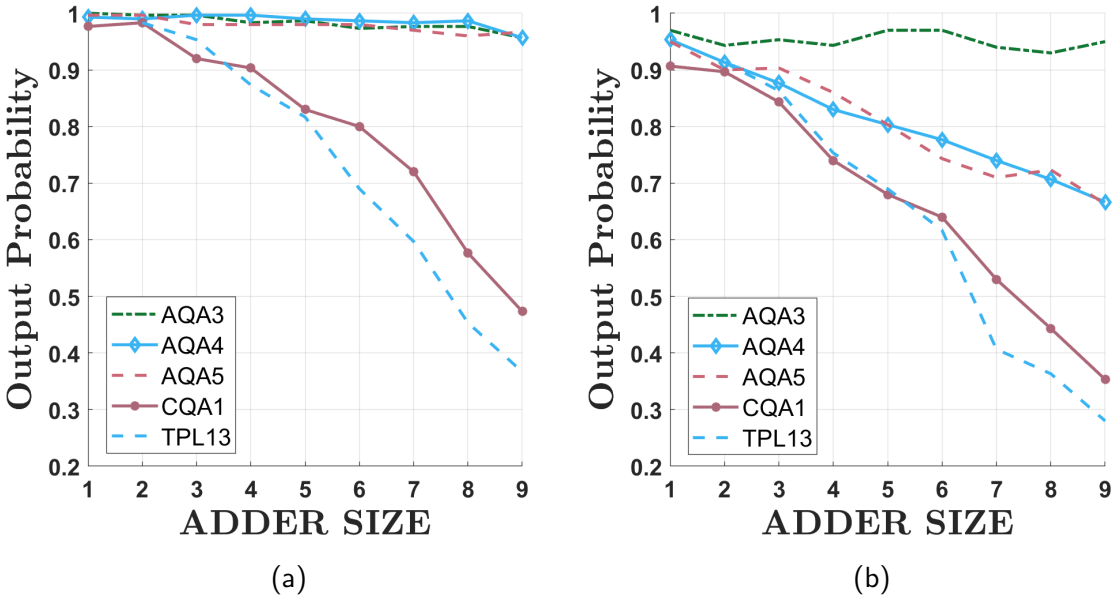


Figure 7.6: Output probability of proposed approximate quantum adders with carryout for: (a) Noise based simulation (no attack). (b) State preparation attack.

that pass-based AQA3 adders are more resilient to this attack, effectively managing the propagation of variation. Notably, other approximate quantum adders perform worse than AQA3, and exact adders cannot cope with the attack, with the impact worsening as the circuit size increases.

7.5.4 Gate Attack

To simulate the gate attack, we add an Rx gate with a 0.25 degree rotation to the target qubit of the CNOT and Toffoli gates used in the quantum adders. This mimics a small deviation that can be introduced through manipulation of the quantum circuit or the resulting pulses during quantum compilation [108] . Figure 7.7 shows that the results closely resemble those of a state preparation attack in Section 7.5.3, where the pass based AQA3 has virtually no impact, as it contains no gates. AQA4 and AQA5 have a single layer of gates, resulting in some degradation of output probability, albeit at a lower rate. As before, exact adders experience steep declines, but unlike state preparation attacks, TPL13 performs slightly better than CQA1 due to lower depth and gate count.

7.5.5 Crosstalk Attack

The crosstalk attack evaluation uses the maximum impact attack vector designed for ion-trap qubits as proposed by Gaur et al., where the attack vector involves a chain of controlled rotation gates strategically placed on the remaining qubits left unused after the quantum adder circuit allocation [57]. This attack leverages the natural physical coupling between neighboring qubits in ion-trap architectures to cause unintended interactions that undermine computational integrity. The detailed explanations of crosstalk noise sources and attack origins are discussed thoroughly in Chapter 6. Figure 7.8 shows that Exact adders experience the most notable performance reduction under crosstalk attacks, while approximate adders keep output probabilities above 0.95 for circuits with up to 8-qubit outputs. The experimental

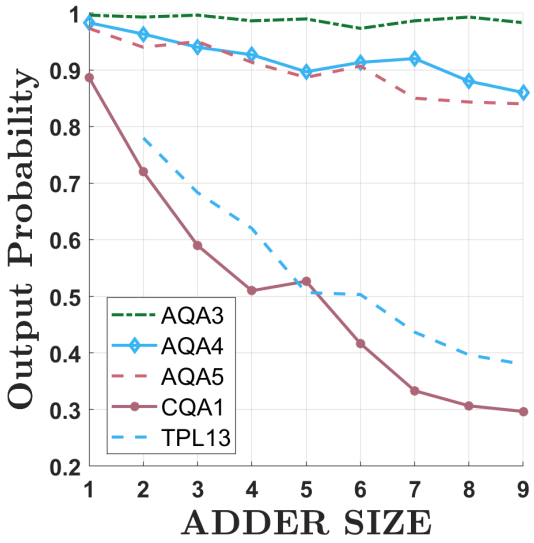


Figure 7.7: Output probability of proposed approximate quantum adders with carryout for gate attack.

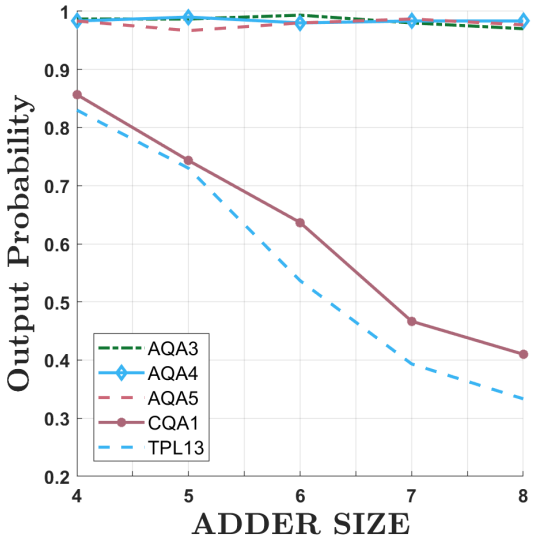


Figure 7.8: Output probability of proposed approximate quantum adders with carryout for crosstalk attack.

results align with theoretical expectations, demonstrating that approximate quantum adders are more resilient due to their constant-depth design and parallel qubit operations, which minimize unintended interactions with attacking qubits. This highlights their practical advantages in Noisy Intermediate Scale Quantum (NISQ) machines, where unintended qubit coupling cannot be fully eliminated.

7.6 Conclusion

In this Chapter, we propose approximate quantum adders, two without carryout and three with carryout, which have constant depth complexity compared to the linear complexity of exact quantum adders. We have optimized the design for parallel operations, which avoids carry propagation among the qubits during operation. The proposed adders utilize fewer resources in the form of lesser CNOT gates and a single or no Toffoli gate. We have shown that the proposed adders have better fidelity than the existing works by performing quantum simulations on the IBM Qiskit platform, conducted using seven different noise models. Among the seven noise models, SPAM and readout noise show insignificant differences in the impact on the proposed as well as existing adders. They depend only on a number of measurements and input states initialized, making them proportional to the qubit count and not the gate count. For the rest of the five noise models, the proposed approximate quantum adders outperform the exact adders by a vast margin in terms of improvement in output probability.

We analyze the error characteristics of the proposed approximate quantum adders using metrics such as Normalized Mean Error Distance (NMED) and error rate, demonstrating their scalability. Approximate quantum adders demonstrate superior resilience against attack vectors, such as state preparation, gate, and crosstalk, compared to exact adders, maintaining output probabilities above 0.95, whereas exact adders suffer significant performance degradation. However, for stuck at 0/1 faults, the exact adders show better resilience, creating a trade-off among

the security aspects. The proposed designs can prove particularly useful in error-tolerant applications like quantum image processing. Our analysis concludes that the proposed approximate quantum adders are highly beneficial in the NISQ era, with our comprehensive error and noise evaluation enabling the selection of an appropriate adder tailored to specific quantum applications.

Chapter 8

Summary

Quantum arithmetic circuits are critical for enabling quantum algorithms and applications, including quantum cryptography, quantum cryptanalysis, scientific computing, and quantum image processing. However, the currently available Noisy Intermediate Scale Quantum (NISQ) era machines face challenges such as noise, decoherence, limited qubits, and error accumulation, constraining the scalability and reliability of quantum arithmetic circuits. Therefore, these circuits must address the limitations of NISQ machines, including low quantum gate fidelity, and susceptibility to noise sources.

Another concern about the NISQ quantum computers is their inherent vulnerabilities that can introduce security concerns, as adversaries can exploit phenomena like crosstalk to compromise circuit integrity. Crosstalk-induced unintended qubit coupling allows attackers to disturb target qubits and analyze neighboring ones, potentially injecting faults or deducing quantum states.

Attackers can also introduce vulnerabilities like stuck at 0/1 attacks, and variability in state preparation and quantum gates, by manipulating the underlying hardware or quantum circuits during compilation. These challenges highlight the need for resilient and scalable designs that mitigate noise and improve reliability, even in the face of such attacks.

To address these challenges, the proposed work introduces innovative designs for quantum arithmetic circuits, utilizing paradigms like distributed quantum computing and approximate computing. Distributed quantum arithmetic, based on the Residue Number System (RNS), offers a scalable solution to quantum arithmetic circuits by distributing operations across multiple parallel circuits.

This methodology eliminates carry dependencies, reduces circuit depth, and improves noise resilience. Experimental validation of RNS based distributed quantum addition using ion-trap based quantum computer confirms its robustness against noise as well as crosstalk attacks, making it suitable for both NISQ and Fault Tolerant Quantum (FTQ) quantum computers.

Approximate quantum addition circuits with constant-depth designs significantly outperform traditional linear-depth adders, demonstrating superior noise resilience under NISQ conditions. The proposed designs in this work leverage techniques like newer architectures and zero resets based on dynamic circuits to reduce quantum resource consumption while improving noise resilience. The broader trend of out performance by approximate quantum adders continues against quantum attacks such as state preparation, gate and crosstalk attacks.

The work also introduces a quantum carry-lookahead modulo (2^n - 1) adder with logarithmic complexity and designs for modulo (2^n + 1) addition. Modulo arithmetic circuits, a subclass of quantum arithmetic, are useful for supporting complex tasks like quantum cryptanalysis and RNS based quantum arithmetic. The quantum modulo arithmetic circuits proposed in this research can enable multiple applications and higher order quantum arithmetic operations.

Overall, this research advances quantum arithmetic by developing scalable, efficient, and noise-resilient circuits. It bridges the gap between current NISQ-era limitations and future FTQ systems, ensuring robust implementations for quantum computing's diverse applications. In summary, by addressing both computational and security challenges, this work lays a foundation for practical, high-performing quantum arithmetic circuits.

8.1 Future Work

Below are the objectives to conduct future research to extend the agenda of distributed and approximate quantum circuits:

1. **Approximate quantum multiplication:** This work explores the potential of approximate quantum addition circuits in achieving constant depth and complexity. These proposed designs can be extended to approximate quantum multiplication circuits to reduce their complexity from $O(n^2)$ to $O(n)$ or even lower. On the other hand, newer approximate quantum addition circuits can also be designed so as to achieve lower quantum resource usage in approximate multiplication.

2. **Attack resilience of RNS based Distributed Quantum Addition (DQA):** The work shows the attack resilience of DQA against crosstalk attacks. It can be extended to other quantum attack sources used in evaluating approximate quantum adders, like stuck at 0/1 attacks, state preparation, and gate attacks. This can help understand the trends in quantum attack susceptibility along with the scaling of quantum addition. Additionally, research can explore whether properties of quantum gates, such as entanglement, superposition, noise, etc., can be used as a defense against the attacks mentioned above.

3. **Optimization of Quantum Diminished-1 Modulo (2^n+1) Multiplier (QDMM):** While this work demonstrates the advantage of Distributed Quantum Multiplication (DQM), the proposed QDMM is not optimized to its full potential. QDMM's qubit count has quadratic complexity, and the CNOT usage is more than twice that of other components of RNS-based DQM, such as modulo (2^n) and (2^n-1), although this saves Toffoli count significantly. The source of this disparity is the quantum 3:2 compressor-based architecture that requires an ancilla and five CNOT gates per block. Future research can explore

reducing QDMM's resource usage by using more suitable sizes of quantum compressors and optimizing the overall design.

4. **Noise and attack resilience of approximate and RNS based Distributed Quantum Multiplication (DQM):** This work studies the impact of various quantum attacks on distributed quantum addition and approximate quantum addition. Both strategies decrease effective depth and complexity, resulting in resilience to quantum noise and attack sources. This behavior can potentially extend to quantum multiplication circuits created using distributed quantum computing and approximation based strategies, making the study of their noise and attack resilience a significantly more valuable task. Also, the generated data can be studied to differentiate between random noise caused by the circuit from an attacker error, lending greater confidence in post processing.

5. **Exploration of optimality of quantum arithmetic circuits:** This work explores various aspects of quantum circuit design, including circuit depth, gate count, qubit usage, and sources of quantum noise, all of which influence the circuit's noise levels and resistance to attacks. However, achieving a comprehensive optimization of quantum arithmetic circuits, considering factors such as error correction, qubit connectivity, and coherence time, remains a distant goal. Inspiration can be drawn from optimal classical circuit design to guide the development of more effective quantum circuits.

Bibliography

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*, pp. 124–134, Ieee, 1994. 1, 35

[2] B. Duan, J. Yuan, C.-H. Yu, J. Huang, and C.-Y. Hsieh, "A survey on hhl algorithm: From theory to application in quantum machine learning," *Physics Letters A*, vol. 384, no. 24, p. 126595, 2020. 1

[3] T. Hur *et al.*, "Quantum convolutional neural network for classical data classification," *Quantum Machine Intelligence*, vol. 4, no. 1, p. 3, 2022. 1, 35, 36

[4] Y. Lee, J. Joo, and S. Lee, "Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience," *Scientific reports*, vol. 9, no. 1, p. 4778, 2019. 1

[5] H.-Y. Xia, H. Zhang, S.-X. Song, H. Li, Y.-J. Zhou, and X. Chen, "Design and simulation of quantum image binarization using quantum comparator," *Modern Physics Letters A*, vol. 35, no. 09, p. 2050049, 2020. 1

[6] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, "Variational quantum algorithms for nonlinear problems," *Physical Review A*, vol. 101, no. 1, p. 010301, 2020. 1

[7] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature Computational Science*, vol. 2, no. 9, pp. 567–576, 2022. 1

[8] Microsoft Quantum Libraries, "version 0.27.253010," 2023. quantum computing library. 1

[9] IBM Quantum, "Qiskit noise models.," dec 2022. https://qiskit.org/documentation/tutorials/simulators/3_building_noise_models.html and https://qiskit.org/documentation/stubs/qiskit_aer.noise.ReadoutError.html. 1, 117, 118

[10] M. P. Harrigan, T. Khattar, C. Yuan, A. Peduri, N. Yosri, F. D. Malone, R. Babbush, and N. C. Rubin, "Expressing and analyzing quantum algorithms with qualtran," 2024. 1

[11] E. Muñoz-Coreas and H. Thapliyal, "T-count and qubit optimized quantum circuit design of the non-restoring square root algorithm," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 14, no. 3, pp. 1–15, 2018. 1

[12] H. Thapliyal, "Mapping of subtractor and adder-subtractor circuits on reversible quantum gates," in *Transactions on Computational Science XXVII*, pp. 10–34, Springer, 2016. 1

[13] E. Muñoz-Coreas and H. Thapliyal, "Quantum circuit design of a t-count optimized integer multiplier," *IEEE Transactions on Computers*, vol. 68, no. 5, pp. 729–739, 2018. 1, 2, 67, 85, 86, 88

[14] S. Sajadimanesh, J. P. L. Faye, and E. Atoofian, "Practical approximate quantum multipliers for nisq devices," in *Proceedings of the 19th ACM International Conference on Computing Frontiers*, pp. 121–130, 2022. 2, 108

[15] I. L. Markov and M. Saeedi, "Constant-optimized quantum circuits for modular multiplication and exponentiation," *arXiv preprint arXiv:1202.6614*, 2012. 2, 38, 39

[16] R. Van Meter and K. M. Itoh, "Fast quantum modular exponentiation," *Phys. Rev. A*, vol. 71, p. 052320, May 2005. 2, 51

[17] P. A. Mohan, *Residue Number Systems*. Springer, 2016. 2, 6, 51, 54, 67, 70

[18] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum resource estimates for computing elliptic curve discrete logarithms," in *Advances in Cryptology–ASIACRYPT 2017*, pp. 241–270, Springer, 2017. 2, 39, 51, 54, 67

[19] D. S. C. Putranto, R. W. Wardhani, H. T. Larasati, and H. Kim, "Another concrete quantum cryptanalysis of binary elliptic curves," *Cryptology ePrint Archive*, 2022. 2, 51, 54, 67

[20] S. Nishio, Y. Pan, T. Satoh, H. Amano, and R. V. Meter, "Extracting success from ibm's 20-qubit machines using error-aware compilation," *J. Emerg. Technol. Comput. Syst.*, vol. 16, may 2020. 2

[21] H.-S. Li, P. Fan, H. Xia, H. Peng, and G.-L. Long, "Efficient quantum arithmetic operation circuits for quantum image processing," *Science China Physics, Mechanics & Astronomy*, vol. 63, pp. 1–13, 2020. 2

[22] A. Dubey, A. Ahmad, M. A. Pasha, R. Cammarota, and A. Aysu, "Modulonet: Neural networks meet modular arithmetic for efficient hardware masking," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 506–556, 2022. 2

[23] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, "A new quantum ripple-carry addition circuit," *arXiv preprint quant-ph/0410184*, 2004. xiii, 2, 3, 4, 30, 31, 33, 36, 39, 51, 107, 108, 110, 119, 120, 122

[24] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, "A logarithmic-depth quantum carry-lookahead adder," *arXiv preprint quant-ph/0406142*, 2004. 2, 32, 33, 39, 51, 85

[25] H. Thapliyal and N. Ranganathan, "Design of efficient reversible logic-based binary and bcd adder circuits," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 9, no. 3, pp. 1–31, 2013. xiii, 2, 3, 30, 31, 33, 51, 55, 64, 99, 107, 108, 110, 119, 121, 122

[26] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Physical Review A*, vol. 54, no. 1, p. 147, 1996. 2, 35, 36, 39

[27] A. Jayashankar, M. D. H. Long, H. K. Ng, and P. Mandayam, "Achieving fault tolerance against amplitude-damping noise," *Physical Review Research*, vol. 4, no. 2, p. 023034, 2022. xiv, 2, 26, 36, 41, 107, 121

[28] M. Urbanek, B. Nachman, V. R. Pascuzzi, A. He, C. W. Bauer, and W. A. de Jong, "Mitigating depolarizing noise on quantum computers with noise-estimation circuits," *Physical review letters*, vol. 127, no. 27, p. 270502, 2021. 2, 26

[29] R. Laflamme, J. Lin, and T. Mor, "Algorithmic cooling for resolving state preparation and measurement errors in quantum computing," *Physical Review A*, vol. 106, no. 1, p. 012439, 2022. 2, 26

[30] S. Dasgupta and T. S. Humble, "Impact of unreliable devices on stability of quantum computations," 2024. 2

[31] N. Cao, J. Lin, D. Kribs, Y.-T. Poon, B. Zeng, and R. Laflamme, "Nisq: Error correction, mitigation, and noise simulation," *arXiv preprint arXiv:2111.02345*, 2021. 2, 3

[32] E. Muñoz-Coreas and H. Thapliyal, "Everything you always wanted to know about quantum circuits," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–17, 1999. 3

[33] B. Gaur, T. Humble, and H. Thapliyal, "Noise-resilient and reduced depth approximate adders for nisq quantum computing," in *Proceedings of the Great Lakes Symposium on VLSI 2023*, pp. 427–431, 2023. xiii, 3, 4, 9, 30, 32, 55, 57, 64, 99

[34] P. Das, S. Tannu, S. Dangwal, and M. Qureshi, "Adapt: Mitigating idling errors in qubits via adaptive dynamical decoupling," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 950–962, 2021. 3

[35] M. Schlosshauer, "Quantum decoherence," *Physics Reports*, vol. 831, pp. 1–57, 2019. 3

[36] A. A. Saki, M. Alam, K. Phalak, A. Suresh, R. O. Topaloglu, and S. Ghosh, "A survey and tutorial on security and resilience of quantum computing," in *2021 IEEE European Test Symposium (ETS)*, pp. 1–10, IEEE, 2021. 3, 91

[37] F. Chen, L. Jiang, H. Müller, P. Richerme, C. Chu, Z. Fu, and M. Yang, "Nisq quantum computing: A security-centric tutorial and survey [feature]," *IEEE Circuits and Systems Magazine*, vol. 24, no. 1, pp. 14–32, 2024. 3, 91, 95

[38] C. Xu, F. Erata, and J. Szefer, "Classification of quantum computer fault injection attacks," *arXiv preprint arXiv:2309.05478*, 2023. 3, 91, 123

[39] N. Acharya, V. Saravanan, and S. M. Saeed, "An attack on quantum circuits based on the error rates of nisq systems and a countermeasure," in *Silicon Valley Cybersecurity Conference* (Y. Park, D. Jadav, and T. Austin, eds.), (Cham), pp. 109–114, Springer International Publishing, 2021. 3

[40] S. Das and S. Ghosh, "Trojan taxonomy in quantum computing," in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 644–649, IEEE, 2024. 5

[41] B. Harper, B. Tonekaboni, B. Goldozian, M. Sevior, and M. Usman, "Crosstalk attacks and defence in a shared quantum computing environment," *arXiv preprint arXiv:2402.02753*, 2024. 5, 92, 93, 95, 96

[42] A. Ash-Saki, M. Alam, and S. Ghosh, "Analysis of crosstalk in nisq devices and security implications in multi-programming regime," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 25–30, 2020. 5, 26, 92, 95

[43] S. Kundu and S. Ghosh, "Adversarial poisoning attack on quantum machine learning models," *arXiv preprint arXiv:2411.14412*, 2024. 5

[44] A. D. Kshemkalyani and M. Singhal, *Distributed computing: principles, algorithms, and systems*. Cambridge University Press, 2011. 5

[45] M. Fathi, M. Haghi Kashani, S. M. Jameii, and E. Mahdipour, "Big data analytics in weather forecasting: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 1247–1275, 2022. 5

[46] O. Spjuth, J. Frid, and A. Hellander, "The machine learning life cycle and the cloud: implications for drug discovery," *Expert opinion on drug discovery*, vol. 16, no. 9, pp. 1071–1079, 2021. 5

[47] E. O. Pyzer-Knapp, J. W. Pitera, P. W. Staar, S. Takeda, T. Laino, D. P. Sanders, J. Sexton, J. R. Smith, and A. Curioni, "Accelerating materials discovery using artificial intelligence, high performance computing and robotics," *npj Computational Materials*, vol. 8, no. 1, p. 84, 2022. 5

[48] Y. Zhang, K. Lu, Y. Gao, and M. Wang, "Neqr: a novel enhanced quantum representation of digital images," *Quantum information processing*, vol. 12, no. 8, pp. 2833–2860, 2013. 5

[49] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: a survey," *Computer Networks*, vol. 254, p. 110672, 2024. 5

[50] W. Liu and F. Lombardi, *Approximate Computing*. Springer, 2022. 6

[51] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: Imprecise adders for low-power approximate computing," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 409–414, IEEE, 2011. 6

[52] M. Masadeh, O. Hasan, and S. Tahar, "Machine-learning-based self-tunable design of approximate computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 4, pp. 800–813, 2021. 6

[53] B. Gaur, E. Muñoz-Coreas, and H. Thapliyal, "A logarithmic depth quantum carry-lookahead modulo (2n - 1) adder," in *Proceedings of the 2023 on Great Lakes Symposium on VLSI*, June 2023. 7

[54] B. Gaur and H. Thapliyal, "Novel optimized designs of modulo 2n+1 adder for quantum computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 32, no. 9, pp. 1759–1763, 2024. 8

[55] B. Gaur, T. S. Humble, and H. Thapliyal, "Residue number system (rns) based distributed quantum addition," in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 595–600, 2024. xvi, 8, 71, 80, 99

[56] B. Gaur and H. Thapliyal, "Residue number system (rns) based distributed quantum multiplication," 2025. xvi, 8, 80, 84

[57] B. Gaur and H. Thapliyal, "Crosstalk attack resilient rns quantum addition," *arXiv preprint arXiv:2410.23217*, 2024. xvii, 8, 97, 103, 105, 127

[58] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. American Association of Physics Teachers, 2002. 10, 16, 21, 26, 115, 118, 121

[59] C. P. Williams, *Explorations in quantum computing*. Springer Science & Business Media, 2010. xi, xiii, 10, 16, 17, 22

[60] Y. S. Weinstein, M. Pravia, E. Fortunato, S. Lloyd, and D. G. Cory, "Implementation of the quantum fourier transform," *Physical review letters*, vol. 86, no. 9, p. 1889, 2001. 14

[61] O. Astafiev, Y. A. Pashkin, T. Yamamoto, Y. Nakamura, and J. Tsai, "Single-shot measurement of the josephson charge qubit," *Physical Review B*, vol. 69, no. 18, p. 180507, 2004. 14

[62] S. Saunders, "Derivation of the born rule from operational assumptions," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046, pp. 1771–1788, 2004. 15

[63] W. K. Wootters and W. H. Zurek, "The no-cloning theorem," *Physics Today*, vol. 62, no. 2, pp. 76–77, 2009. 15

[64] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996. 16

[65] V. Bužek and M. Hillery, "Quantum copying: Beyond the no-cloning theorem," *Physical Review A*, vol. 54, no. 3, p. 1844, 1996. 16

[66] A. Kumar Pati and S. L. Braunstein, "Impossibility of deleting an unknown quantum state," *Nature*, vol. 404, no. 6774, pp. 164–165, 2000. 16

[67] A. K. Pati and S. L. Braunstein, "Quantum no-deleting principle and some of its implications," *arXiv preprint quant-ph/0007121*, 2000. 16

[68] S. Bravyi and A. Kitaev, "Universal quantum computation with ideal clifford gates and noisy ancillas," *Physical Review A—Atomic, Molecular, and Optical Physics*, vol. 71, no. 2, p. 022316, 2005. 21

[69] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024. 21, 61

[70] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. Campora, C. Carron, J. Chambers, J. W. Chan, Y. H. Chen, A. Chernoguzov, E. Chertkov, J. Colina, J. P. Curtis, R. Daniel, M. DeCross, D. Deen, C. Delaney, J. M. Dreiling, C. T. Ertsgaard, J. Esposito, B. Estey, M. Fabrikant, C. Figgatt, C. Foltz, M. Foss-Feig, D. Francois, J. P. Gaebler, T. M. Gatterman, C. N. Gilbreth, J. Giles, E. Glynn, A. Hall, A. M. Hankin, A. Hansen, D. Hayes, B. Higashi, I. M. Hoffman, B. Horning, J. J. Hout, R. Jacobs, J. Johansen, L. Jones, J. Karcz, T. Klein, P. Lauria, P. Lee, D. Liefer, S. T. Lu, D. Lucchetti, C. Lytle, A. Malm, M. Matheny, B. Mathewson, K. Mayer, D. B. Miller, M. Mills, B. Neyenhuis, L. Nugent, S. Olson, J. Parks, G. N. Price, Z. Price, M. Pugh, A. Ransford, A. P. Reed, C. Roman, M. Rowe, C. Ryan-Anderson, S. Sanders, J. Sedlacek, P. Shevchuk, P. Siegfried, T. Skripka, B. Spaun, R. T. Sprenkle, R. P. Stutz, M. Swallows, R. I. Tobey, A. Tran, T. Tran, E. Vogt, C. Volin, J. Walker, A. M. Zolot, and J. M. Pino, "A race-track trapped-ion quantum processor," *Phys. Rev. X*, vol. 13, p. 041052, Dec 2023. 21, 74

[71] B. Johnson, "Bringing the full power of dynamic circuits to qiskit runtime.," nov 2022. 24

[72] F. Hua, Y. Jin, Y. Chen, S. Vittal, K. Krsulich, L. S. Bishop, J. Lapeyre, A. Javadi-Abhari, and E. Z. Zhang, "Exploiting qubit reuse through mid-circuit

measurement and reset," *arXiv preprint arXiv:2211.01925*, 2022. 24

[73] G. P. Gehér, M. Jastrzebski, E. T. Campbell, and O. Crawford, "To reset, or not to reset–that is the question," *arXiv preprint arXiv:2408.00758*, 2024. 25

[74] D. A. Lidar and T. A. Brun, *Quantum error correction*. Cambridge university press, 2013. 25

[75] J. Kim, S. Seo, J. Yun, and J. Bae, "Static quantum errors and purification," *arXiv preprint arXiv:2405.06291*, 2024. 25

[76] D. P. DiVincenzo, "Quantum computation," *Science*, vol. 270, no. 5234, pp. 255–261, 1995. 25

[77] H.-L. Huang, D. Wu, D. Fan, and X. Zhu, "Superconducting quantum computing: a review," *Science China Information Sciences*, vol. 63, pp. 1–32, 2020. 27

[78] M.-Z. Piao and X. Ji, "Quantum decoherence under phase damping in non-inertial frames," *Journal of Modern Optics*, vol. 59, no. 1, pp. 21–25, 2012. 26

[79] N. I. Ishak, S. V. Muniandy, and W. Y. Chong, "Entropy analysis of the discrete-time quantum walk under bit-flip noise channel," *Physica A: Statistical Mechanics and its Applications*, vol. 584, p. 126371, 2021. 26

[80] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Applied physics reviews*, vol. 6, no. 2, 2019. 27

[81] F. Bernardini, A. Chakraborty, and C. R. Ordóñez, "Quantum computing with trapped ions: a beginner's guide," *European Journal of Physics*, vol. 45, no. 1, p. 013001, 2023. 28

[82] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," *Applied Physics Reviews*, vol. 6, no. 2, 2019. 28, 92, 95, 98

[83] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on computers*, vol. 62, no. 9, pp. 1760–1771, 2012. 34

[84] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *ACNS*, vol. 5, pp. 164–175, Springer, 2005. 35, 36

[85] A. Petzoldt, "Efficient key generation for rainbow," in *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, France*, pp. 92–107, Springer, 2020. 35, 36

[86] N. Jiang, W.-Y. Wu, and L. Wang, "The quantum realization of arnold and fibonacci image scrambling," *Quantum information processing*, vol. 13, no. 5, pp. 1223–1236, 2014. 35, 36

[87] Y. Takahashi and N. Kunihiro, "A linear-size quantum circuit for addition with no ancillary qubits," *Quantum Information & Computation*, vol. 5, no. 6, pp. 440–448, 2005. 36, 39

[88] A. Kim, S.-M. Cho, C.-B. Seo, S. Lee, and S.-H. Seo, "Quantum modular adder over gf (2 n- 1) without saving the final carry," *Applied Sciences*, vol. 11, no. 7, p. 2949, 2021. xiv, 36, 37, 40, 41, 99

[89] IBM Quantum, "version 0.39.0," 2022. computer software. 36

[90] L.-S. Didier and L. Jaulmes, "Fast modulo 2 n- 1 and 2 n; 1 adder using carry-chain on fpga," in *2013 Asilomar Conference on Signals, Systems and Computers*, pp. 1155–1159, IEEE, 2013. 38

[91] A. Ceschini, A. Rosato, and M. Panella, "Modular quantum circuits for secure communication," *IET Quantum Communication*, vol. 4, no. 4, pp. 208–217, 2023. 51

[92] W. Tang and M. Martonosi, "Distributed quantum computing via integrating quantum and classical computing," *Computer*, vol. 57, no. 4, pp. 131–136, 2024. 68, 70

[93] D. Barral, F. J. Cardama, G. Díaz, D. Faílde, I. F. Llovo, M. M. Juane, J. Vázquez-Pérez, J. Villasuso, C. Piñeiro, N. Costas, *et al.*, "Review of distributed quantum computing. from single qpu to high performance quantum computing," *arXiv preprint arXiv:2404.01265*, 2024. 68, 70

[94] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, "t—ket⟩: a retargetable compiler for nisq devices," *Quantum Science and Technology*, vol. 6, p. 014003, nov 2020. 68

[95] C. Chevignard, P.-A. Fouque, and A. Schrottenloher, "Reducing the number of qubits in quantum factoring," *Cryptology ePrint Archive*, 2024. 70

[96] C. Gidney, "How to factor 2048 bit rsa integers with less than a million noisy qubits," *arXiv preprint arXiv:2505.15917*, 2025. 70

[97] I. A. Kalmykov, V. P. Pashintsev, K. T. Tyncherov, A. A. Olenev, and N. K. Chistousov, "Error-correction coding using polynomial residue number system," *Applied Sciences*, vol. 12, no. 7, p. 3365, 2022. 70

[98] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan, "t—ket⟩: a retargetable compiler for nisq devices," *Quantum Science and Technology*, vol. 6, p. 014003, nov 2020. 74

[99] S.-M. Cho, A. Kim, D. Choi, B.-S. Choi, and S.-H. Seo, "Quantum modular multiplication," *Ieee Access*, vol. 8, pp. 213244–213252, 2020. 85, 86

[100] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, "An algorithm for the t-count," *arXiv preprint arXiv:1308.4134*, 2013. 88

[101] A. A. Saki, R. O. Topaloglu, and S. Ghosh, "Shuttle-exploiting attacks and their defenses in trapped-ion quantum computers," *IEEE Access*, vol. 10, pp. 2686–2699, 2021. 92, 99

[102] A. A. Saki and S. Ghosh, "Qubit sensing: A new attack model for multi-programming quantum computing," *arXiv preprint arXiv:2104.05899*, 2021. 93

[103] S. Maurya, C. N. Mude, B. Lienhard, and S. Tannu, "Understanding side-channel vulnerabilities in superconducting qubit readout architectures," *arXiv preprint arXiv:2405.08962*, 2024. 95

[104] D. Mehra and A. Kalev, "Defending crosstalk-mediated quantum attacks using dynamical decoupling," *arXiv preprint arXiv:2409.14598*, 2024. 95

[105] C. Fang, Y. Wang, S. Huang, K. R. Brown, and J. Kim, "Crosstalk suppression in individually addressed two-qubit gates in a trapped-ion quantum computer," *Physical Review Letters*, vol. 129, no. 24, p. 240504, 2022. 95

[106] C. Adami and N. J. Cerf, "von neumann capacity of noisy quantum channels," *Physical Review A*, vol. 56, no. 5, p. 3470, 1997. 98

[107] P. Nation, "Improving state prep errors on ibm quantum systems," nov 2021. 118

[108] C. Xu and J. Szefer, "Security attacks abusing pulse-level quantum circuits," in *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 83–83, IEEE Computer Society, 2024. 127

Vita

Bhaskar Gaur was born and raised in India, where he attended M.J.P. University in Bareilly, UP, earning a Bachelor of Technology degree in Electronics and Communication. His passion for semiconductors and computational logic design inspired him to pursue a Master's degree in VLSI Design at V.I.T. University in Vellore, TN. During his master's degree, he interned at Intel Corporation in Bangalore, KA, and after graduation, he joined the company as a CAD Engineer. He later worked at Qualcomm Corporation in Bangalore, KA, before moving to the USA for doctoral studies. During this period, he received Qualcomm's "Qualstar" award twice for outstanding delivery in top-level STA and was also recognized at Intel for developing the Timing Power co-optimization tool.

After over seven years in the semiconductor industry, he joined the University of Tennessee, Knoxville (UTK), as a graduate student in computer science. He conducted research at the Cy-QuBIT Lab under the guidance of Dr. Himanshu Thapliyal, focusing on the noise resilience and security of quantum circuits. He analyzed quantum arithmetic circuits through both simulation and execution on actual quantum computers. He explored approximate and distributed quantum computing to improve noise and attack tolerance. He has published several papers and presented at both national and international conferences. He received the Best Poster/Late Breaking Work Award at IEEE GLSVLSI 2023 and the Best Poster Award at QCUF 2023. Additionally, he was honored with the Volunteer of Distinction Award for Academic Achievement at UTK in 2025.