Quantum Approach to the Integer Wavelet Transform and its Application to Quantum Lossless Compression

by

M.Sc. Freddy Alejandro Chaurra Gutiérrez

A Dissertation

submitted to the Program in Computer Science,

Computer Science Department

in Partial fulfillment of the requirements for the degree of

DOCTOR IN COMPUTER SCIENCE

at the

Instituto Nacional de Astrofísica, Óptica y Electrónica

August, 2024

Tonantzintla, Puebla, México

Advisors by:

Dr. Claudia Feregrino Uribe,

Dr. Julio César Pérez Sansalvador,

Dr. Gustavo Rodríguez Gómez,

Computer Science Coordination

Acknowledgements

Special thanks to my family for always supporting me.

Thanks to my advisors for their guidance.

This research was supported by the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT) with the CVU 937653.

Contents

List of Figures

List of Tables

Abstract

Quantum computing uses superposition and parallelism properties to address different computational problems, decreasing time and storage requirements. Thus, quantum transforms have demonstrated their capabilities for developing powerful algorithms and solving complex problems. Quantum wavelet transforms play a fundamental role in information processing applications such as data hiding and cryptography, reducing computational complexities but limiting their capabilities due to the small set of quantum transforms, constraints of quantum representation formats, and challenges of quantum computing. Therefore, we propose a new class of one-dimensional quantum wavelet transforms based on the lifting scheme. We develop the quantum integer version of the Haar, CDF(2,2), and Daubechies-4 (DB4) wavelets. We design quantum circuits, avoiding nonlinearities and giving polynomial complexities. Also, we present the unitary and algorithmic definitions of the transformations. We define the new Quantum Block Representation by Basis States (QBRBS), facilitating signal manipulation. Additionally, we propose a hybrid quantum-classical lossless compression scheme based on wavelet decomposition and fixed-length coding, decreasing time complexity. Finally, we perform a set of analyses, including wavelet simulations, quantum complexities, comparative descriptions of wavelet transforms, features and limitations of quantum representation formats, and compression properties, showing the feasibility and applicability of the proposed quantum wavelet transforms.

Resumen

La computación cuántica utiliza las propiedades de superposición y paralelismo para abordar distintos problemas computacionales, disminuyendo los requisitos de tiempo y almacenamiento. Así, las transformadas cuánticas han demostrado sus capacidades para desarrollar potentes algoritmos y resolver problemas complejos. Las transformadas cuánticas wavelet juegan un papel fundamental en aplicaciones de procesamiento de información como el ocultación de información y la criptografía, reduciendo las complejidades computacionales pero limitando sus capacidades debido al pequeño conjunto de transformadas cuánticas, las restricciones de los formatos de representación cuántica y los retos de la computación cuántica. Por lo tanto, proponemos una nueva clase de transformadas wavelet cuánticas unidimensionales basadas en el esquema lifting. Desarrollamos la versión cuántica entera de las wavelets Haar, CDF(2,2) y Daubechies-4 (DB4). Diseñamos los circuitos cuánticos, evitando las no linealidades y proporcionando complejidades polinómicas. También presentamos las definiciones unitarias y algorítmicas de las transformaciones. Definimos una nueva Representación Cuántica de Bloques por Estados Base (QBRBS), facilitando la manipulación de señales. Además, proponemos un esquema de compresión sin pérdidas híbrido cuántico-clásico basado en la descomposición wavelet y la codificación de longitud fija, disminuyendo la complejidad en tiempo. Por último, realizamos una serie de análisis, que incluyen simulaciones wavelets, complejidades cuánticas, descripciones comparativas de las transformadas wavelet, características y limitaciones de los formatos de representación cuántica, y propiedades de compresión, demostrando la viabilidad y aplicabilidad de las transformadas wavelet cuánticas propuestas.

1

Introduction

Quantum Computing (QC) enables the storage and manipulation of information using quantum superposition, entanglement, and interference properties, leading to reduced computational costs and memory requirements compared to classical computing [1, 2]. This efficiency has been demonstrated in various application areas such as cryptography, information hiding, information processing, machine learning, and optimization [3–5]

Furthermore, research in quantum information storage and quantum transforms has demonstrated the potential of QC for solving complex problems. Quantum representation formats, for instance, enable an exponential decrease in information storage in the quantum realm, making use of the entanglement and superposition of quantum states [6–8]. In addition, the Quantum Fourier Transform (QFT) serves as the basis for robust quantum algorithms like Integer Prime Factoring (Shor's algorithm) and Phase Estimation [9]. As a result, the Quantum Cosine (QCT) and Wavelet Transforms (QWTs) have been developed, increasing the impact of the quantum computing field [10–12].

Specifically, QWTs are essential tools in compression, watermarking, cryptography, coding, signal analysis, filtering, and denoising processes [11, 13]. However, it is worth noting that only quantum versions of the Haar and Daubechies-4 (DB4) wavelet

transforms have been developed, which focus on the real-valued transformation, thereby reducing computational complexity compared to classical versions [10, 14, 15]. Nevertheless, they still need to be improved in comparison to the extensive range of classical wavelet transforms, which incorporate integer-to-integer transforms and various wavelet bases.

Integer transforms facilitate an integer-to-integer mapping without loss of information between input values and decomposition elements, and are fundamental in applications in which recovery of the initial data is critical, such as signal analysis, banking, military operations, medical diagnostics, and data hiding [16, 17]. Therefore, the development of quantum versions of integer wavelet transforms holds potential value for quantum information processing, lossless applications, and the expansion of available quantum transforms.

On the other hand, the development of quantum integer wavelet transforms (QIWTs) involves various challenges due to the quantum constraints and features [1, 18–20]. Consequently, classical operators must be defined in terms of quantum gates to ensure QIWTs are unitary and have polynomial quantum complexity [20, 21]. Furthermore, certain operations may or may not be applicable, depending on the representation format used [11]. The probabilistic nature of quantum information poses a hindrance to the measurement process, making it highly complex to observe stored data [1, 20]. Thus, the development of this class of quantum transformations requires a clever solution.

This research aims to design quantum integer versions of the Haar, CDF(2,2), and Daubechies-4 (DB4) wavelet transform based on the lifting scheme. We present a comprehensive quantum description for each wavelet, including quantum circuits and complexity analysis, unitary definitions, algorithmic descriptions, and simulation experiments. The computational complexity of all three wavelet transforms is significantly reduced in comparison to their classical counterparts, achieving polynomial quantum complexity. The unitary and algorithmic definitions enhance the applicability and usability of the proposed transforms. Also, we introduce the Quantum Block Representation

by Basis States (QBRBS), a novel quantum representation format that facilitates the decomposition process and enhances the applicability of the proposed quantum transforms. Furthermore, we propose a hybrid quantum-classical lossless compression scheme based on wavelet decomposition and fixed-length coding, decreasing time complexity. Finally, we perform a set of analyses, including wavelet simulations, quantum complexities, comparative descriptions of wavelet transforms, features and limitations of quantum representation formats, and compression properties, showing the feasibility and applicability of the proposed quantum wavelet transforms.

1.1 Problem Statement

Quantum computing uses quantum superposition, entanglement, and interference properties to solve problems in different areas, decreasing computational times. However, to our knowledge, there are no quantum versions of integer-to-integer transforms, which could be valuable for quantum research and the development of lossless quantum applications. Therefore, this research addresses the problem of developing a quantum version of the one-dimensional integer-to-integer wavelet transform for the Haar, CDF(2,2), and Daubechies-4 (DB4) kernels with an application to quantum lossless compression. Thus, we consider the following specific issues:

- Find and select a quantum one-dimensional signal representation to apply the quantum integer wavelet transform.
- Manipulate the quantum states to achieve successful decomposition results for the proposed approach.
- Construct and factorize the unitary operators used in the proposed transform.
- Design quantum algorithms to signal decomposition and lossless compression.
- Improve the performance of the integer wavelet transforms based on the development of a quantum lifting scheme.

1.1.1 Problem Definition

Given a classical description of the one-dimensional wavelet transform, how do we develop a one-dimensional quantum integer wavelet transform based on the lifting scheme? (To quantum signal decomposition and quantum lossless compression).

Classical Wavelet Transform (WT)

We define the classical WT as

$$W_m^{(j)} S^T \rightarrow (A_1^0,\ D_1^0,\ D_1^1,\ D_2^1, \ldots,\ D_{i-1}^j,\ D_i^j\ ) = (A,\ D) \tag{1.1}$$

where $S = (s_1,\ s_2,\ \ldots,\ s_m)$ is the input signal, $W_m^{(j)}$ is the matrix form of the WT for a signal of length $m = 2^n$ at the j-th decomposition level with $j \in [0, l-1]$, being l the maximum decomposition level. $A = A_1^0$ is the approximation coefficient at the zero level, and D_i^j is the i-th detail coefficient at the j-th level, that is,

$$D = \sum_{j=0}^{l-1} \sum_{i=1}^{2^j} D_i^j \tag{1.2}$$

where l is the maximum decomposition level.

Quantum WT

Based on (1.1) and (1.2), we define a quantum representation for the integer wavelet transform by

$$U_{W_m}^j |S\rangle \rightarrow \left|A^0\right\rangle \otimes \sum_{j=0}^{l-1} \left|D^j\right\rangle = |A, D\rangle \tag{1.3}$$

where $|S\rangle$ is the signal vector encoded in a quantum format, $\otimes$ the tensor product, $U_{W_m}^j$ the matrix operator for the quantum WT for a signal of length $m = 2^n$ with l-decomposition levels, $|A^0\rangle = |A\rangle$ the approximation coefficient at the zero level, and $|D\rangle = |D^j\rangle$ the detail coefficient at j-th level, given by

$$\left|D^j\right\rangle = \sum_{i=1}^{2^j} \left|d_i^j\right\rangle \tag{1.4}$$

where $|d_i^j\rangle$ is the i-th detail coefficient at j-th level.

Factorization

We require to find and implement the QWT operator, which we will reduce to the problem of factoring the U_{W_m} operator. Our approach is to factor the classical operator for this transform into products, and sums of smaller unitary operators. We will consider the permutation matrices and some other unitary matrices as the basis of the development. The key is to exploit the specific structure of each unitary operator to find an efficient representation to implement it.

Given the U_{W_m} operator for the QWT, we will select subsets of unitary operators that efficiently perform the QWT for Haar, CDF(2,2), and DB4 kernels.

$$U_{W_m} = (U_0 \circ U_1 \circ \ldots \circ U_{n-1}) \tag{1.5}$$

Where U_i are unitary operators, and $(\circ)$ can be any of the following operators: the tensor product, $(\otimes)$, the direct sum, $(\oplus)$, and/or the dot product, $(U_i \cdot U_j)$.

1.2 Research Questions

The main questions that guide this research are:

1. Which unitary operators allow us to extend the one-dimensional integer wavelet transform to the quantum domain?

2. How can one-dimensional signals be represented, using the existing quantum format techniques, to improve the signal decomposition results of the proposed quantum integer wavelet transform?

3. How can a lossless compression algorithm be designed using the proposed quantum transform?

1.3 Hypothesis

Based on unitary operator factorization through permutation matrices' it is possible to develop a quantum approach to the one-dimensional integer wavelet transform for Haar, Daubechies-4, and CDF(2,2) kernels.

A quantum representation based on the existing quantum formats, using basis states to store information, improves the signal decomposition of the proposed quantum transform compared to the classical counterpart.

A neighborhood and redundancy relationship among signal elements allows us to design a lossless compression algorithm based on the proposed quantum transform.

1.4 General Objective

Propose a quantum approach for the one-dimensional integer wavelet transform for Haar, Daubechies-4, and CDF(2,2) kernels and design algorithms for quantum signal decomposition and quantum lossless compression.

1.4.1 Specific Objectives

In order to accomplish the general objective, the following specific objectives must be completed:

1. To identify the factorization matrices that characterize the unitary operators for the one-dimensional quantum integer wavelet transform.
2. To analyze and select a quantum format to represent one-dimensional signals that allows and improves signal decomposition.

3. To develop a quantum algorithm for one-dimensional signal decomposition using the proposed quantum transform.

4. To develop a quantum lossless compression algorithm based on the proposed quantum integer wavelet transform.

1.5 Research Scope

This work is limited by the following conditions:

- This research is concerned to the one-dimensional quantum integer wavelet transform.
- Quantum lossless compression is considered the main application.
- Simulation environments are used as a means of verification.
- Noisy environments are out of the scope of this research.

1.6 Contributions

The main contributions in the area of computer sciences from this doctoral research are the following:

1. A quantum approach for a subset of wavelet kernels (Haar, CDF(2,2), and Daubechies-4).

2. Three quantum integer wavelet transforms with an improvement in computational cost over the classical counterpart.

3. Quantum algorithms to one-dimensional signal decomposition using the quantum integer wavelet transforms.

4. A quantum lossless compression algorithm for one-dimensional signals based on the proposed transform.

1.7 Publications

- Chaurra-Gutierrez, F. A., Feregrino-Uribe, C., Perez-Sansalvador, J. C., & Rodriguez - Gomez, G. (2023). QIST: One-dimensional quantum integer wavelet S-transform. Information Sciences, 622, 999-1013. (**Published**).
- Chaurra-Gutierrez, F. A., Perez-Sansalvador, J. C., Rodriguez-Gomez, G., & Feregrino - Uribe, C. (2022, September). Qbrbs: Quantum block representation by basis states. In 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) (pp. 125-132). IEEE. (**Published**).
- Chaurra-Gutierrez, F. A., Rodriguez-Gomez, G., Feregrino-Uribe, C., & Perez - Sansalvador, J. C. (202x). One-dimensional Quantum Integer CDF(2,2) Wavelet Transform. Information Sciences, xxx, xxx-xxx. (**Under review**).
- Chaurra-Gutierrez, F. A., Perez-Sansalvador, J. C., Feregrino-Uribe, C., & Rodriguez - Gomez, G. (202x). Quantum Operations on Superposed States. -Journal-, xxx. (**Submitted**).
- Chaurra-Gutierrez, F. A., Feregrino-Uribe, C., Perez-Sansalvador, J. C., & Rodriguez - Gomez, G. (2024). Quantum Mean Filter: Features and Issues. In 2024 IEEE International Conference on Quantum Computing and Engineering (QCE) (pp. xxx-xxx). IEEE. (**Under review**).

1.8 Thesis Outline

Chapter 2 introduces the background about quantum computing, and wavelet transforms. Chapter 3 presents the related work. Chapter 4 describes the proposed quan-

tum solution for the integer wavelet transforms and lossless compression application. Chapter 5 shows the simulation results. Finally, Chapter 6 draws conclusions and future work.

2

Background

This chapter provides definitions of quantum computing, wavelet transforms, and compression. First, we outline the fundamental concepts of quantum computing, such as representation, manipulation, and algorithms in the quantum domain. Then, we examine the classical wavelet transform, emphasizing its classes and implementations. Finally, we introduce general data compression, including its lossless model.

2.1 Quantum Computing

Quantum computing leverages properties of quantum mechanics that are unavailable in classical computing, including superposition, entanglement, and interference, to solve different problems in innovative ways [20, 22]. As a result, the development of quantum computing has centered around three main areas: encoding data in the quantum domain, manipulating quantum information, creating quantum algorithms, and measuring to extract expected results [20, 22].

2.1.1 Quantum Information Representation

Quantum computers use quantum systems such as atomic energy levels, nuclear spin, or photon polarization to store and process information. The main characteristic of quantum computing is its probabilistic nature, where we can only obtain a particular result with a certain probability [9, 20, 22].

The *qubit* (quantum bit) is the basic information unit in quantum computing, similar to the classical bits. However, qubits can be simultaneously in different states or entangled with other qubits, giving new properties and storing more information than classical bits [9, 20]. Qubits could be described as single elements into the computational basis

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

Also, it could describe a **superposition** of different states as

$$|\psi\rangle = \sum_{i=0}^{N-1} \alpha_i |a_i\rangle, \ N = 2^n \tag{2.2}$$

where $|a_i\rangle$ are the states, and α_i are the coefficients associated to the probability of observing a given state, $P(|a_i\rangle) = |\alpha_i|^2$. The qubit register $|\psi\rangle$ is in all the $|a_i\rangle$ states simultaneously [9, 20, 22]. Additionally, qubits could be **entangled**, defining a strong correlation between different elements that share information. For example, a Bell state describing two entangled qubits is

$$|\Phi^+\rangle = |q_1 q_0\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right) \tag{2.3}$$

where measuring the qubit $|q_1\rangle$ or $|q_0\rangle$ gives information about the other state, that is, if measurement yields $|q_0\rangle = |0\rangle$, we are sure that $|q_1\rangle = |0\rangle$. This property helps to develop quantum representation formats, algorithms, and communication protocols with improved computational time [20, 22, 23].

On the other hand, by combining the properties of superposition and entanglement, we could define **quantum representation formats** to encode and manipulate

large amounts of information efficiently. Thus, quantum formats allow storing image, audio, or video information using amplitudes, phases, and basis states to perform different processing operations [24, 25]. The idea is to capture information about signal values and position coordinates in a quantum register. For example, Quantum Basis States (QBS) representations are described by

$$|S\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |X_i\rangle |f_i\rangle \tag{2.4}$$

where $|X_i\rangle$ is the n-dimensional coordinate position, $|f_i\rangle$ are the signal values, and N is the signal length [25]. These formats require $n + q$ qubits. Figure 2.1 shows a one- and two-dimensional signals in QBS, where two-dimensional sum is split to describe rows and columns.

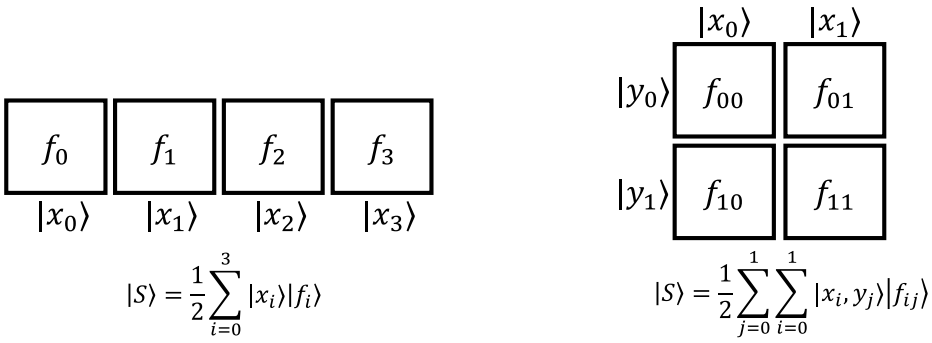

Figure 2.1: QBS representation of one and two-dimensional signals, respectively.

Finally, quantum formats exponentially decrease storage requirements compared to classical methods. However, some operations are easy to implement, and others have a high quantum complexity depending on the quantum representation model [11].

2.1.2 Quantum Information Manipulation

Quantum operators (*gates*) enable the manipulation of information into the quantum domain through unitary transformations, guaranteeing the linearity and reversibility of the processes. Thus, complex networks of quantum circuits are the basis of quantum

algorithms solving different problems, where each proposed quantum operation must have a polynomial quantum complexity [20, 22, 26].

A useful **quantum set** includes single-qubit operators such as the NOT (X), V and $V^\dagger$, Hadamard (H), and two-qubit Conditional-NOT (CNOT) gate, which performs a NOT operation on qubit two if the first qubit is in state $|1\rangle$.

$$X : \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} ; X|0\rangle \rightarrow |1\rangle\,,\ X|1\rangle \rightarrow |0\rangle \tag{2.5}$$

$$\begin{aligned} V : \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix} &; V|0\rangle \rightarrow \frac{1+i}{2}(|0\rangle - i|1\rangle),\ V|1\rangle \rightarrow \frac{1+i}{2}(-i|0\rangle + |1\rangle) \\ V^\dagger : \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} &; V^\dagger|0\rangle \rightarrow \frac{1-i}{2}(|0\rangle + i|1\rangle),\ V^\dagger|1\rangle \rightarrow \frac{1-i}{2}(i|0\rangle + |1\rangle) \end{aligned} \tag{2.6}$$

$$H : \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} ; H|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ H|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{2.7}$$

$$\begin{aligned} CNOT : \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} &; CNOT|00\rangle \rightarrow |00\rangle\,,\ CNOT|01\rangle \rightarrow |01\rangle \\ & CNOT|10\rangle \rightarrow |11\rangle\,,\ CNOT|11\rangle \rightarrow |10\rangle \end{aligned} \tag{2.8}$$

Additionally, quantum algorithms require **complexity analysis** to ensure the feasibility and applicability of the solutions. Therefore, different metrics can be used, such as [20, 22, 26]:

- **Gate set:** base assembly with the class of gates used in the circuit.
- **Gate cost:** number of single -or two-qubit quantum gates.
- **Depth:** number of layers containing parallel gates.
- **Qubit cost:** number of qubits to implement the quantum circuit.

- **Ancillary qubits:** additional qubits (auxiliary) required at the beginning of the circuit to obtain the expected result.
- **Garbage qubits**: any output qubit that does not store useful values.

The above metrics allow characterizing quantum algorithms, providing a quantum complexity analysis. However, further analysis elements must be considered depending on actual implementations or noise considerations [26, 27].

2.1.3 Quantum Measurement

Quantum measurement is an active and irreversible process to obtain the expected results at the end of quantum algorithms, where the output values are stored into classical bits. However, due to the probabilistic nature of the quantum domain, measuring or observing a quantum element destroys any kind of superposition or entanglement property [20, 22, 28, 29]. For example, perform a measurement, M, on a superposed state, $|a\rangle = \alpha_0|a_0\rangle + \alpha_1|a_1\rangle$, yields state $|a_0\rangle$ with a probability $|\alpha_0|^2$ or $|a_1\rangle$ with probability $|\alpha_1|^2$ on the computational basis, so that at the end we can only get one of the values in superposition. Therefore, at least $N = 2^n$ measurements are needed to observe all the values of a superposed or entangled state.

2.2 Wavelet Transform

The Wavelet transform is a processing tool utilized for signal analysis. It decomposes input values into approximation (low-frequency) and detail (high-frequency) components to fully describe all the information content of the input signal in a new representation. This approach enables the analysis of various signal features, including trends, discontinuities, variations, breakpoints, correlations, and frequency responses [30–32].

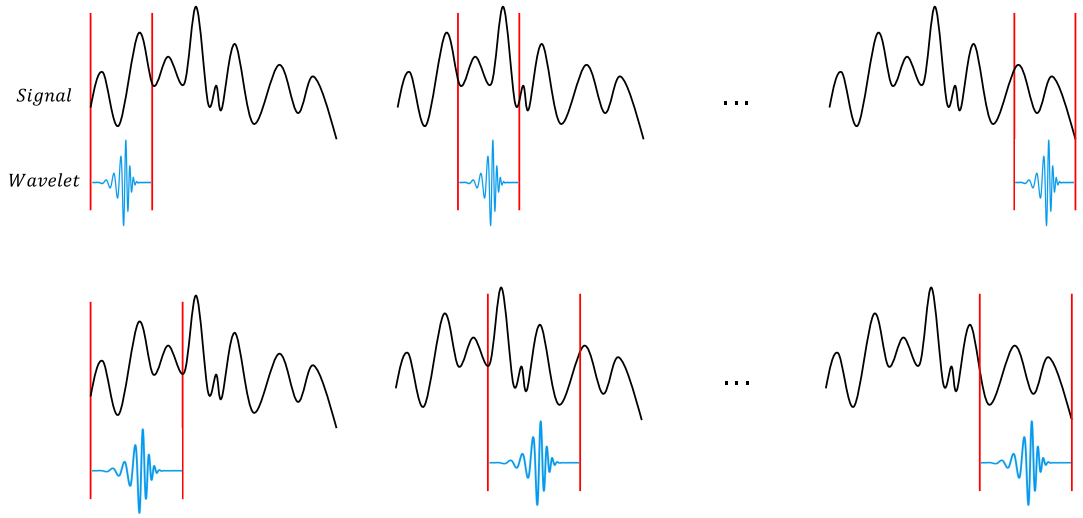

Figure 2.2: General wavelet decomposition process. The wavelet kernel is shifted and scaled to cover the entire signal.

Wavelet analysis employs a sliding window technique of varying sizes to decompose signals with different base waveform (*kernels*). The process involves selecting a wavelet function within a limited domain and indicating similarities between various parts of the input signal using scaled and shifted wavelet kernels that generate the approximation and detail coefficients [30–32]. Figure 2.2 outlines the overall process, wherein compressed wavelets represent the high-frequency features (details) of the signal, and stretched shapes correspond to the low-frequency components (approximation).

The above described process can achieve up to n levels of decomposition on a 2^n signal. This is done by taking the obtained approximation coefficients and performing the wavelet transform successively. The choice of an appropriate wavelet kernel is determined by the application and the problem's characteristics. In general, wavelets are effective in preserving time-frequency information, performing local analysis, and identifying variations at different scales [32].

The wavelet transform can be classified as either continuous, where it considers all scales and shifts of the domain, or discrete, using only subsets of the domain elements for analysis [30–32]. The wavelet coefficients can also be in a real or integer-valued

domain. However, continuous transforms are limited by the finite precision of the computer, which can cause information loss when the input data are real values. In contrast, the integer domain transform is reversible finite precision, which is useful in multimedia applications where information retrieval is crucial [16, 17, 33].

2.2.1 Wavelet Lifting Definition

The wavelet transform can be implemented using different approaches, either with time-based or frequency-based analysis employing lifting steps or filter banks, respectively. However, both methods describe the same process, and time-based lifting has an equivalent representation in frequency [16, 17, 33].

The lifting method takes advantage of the characteristics of the data values, involving a series of prediction and update steps for the decomposition process. Each step performs distinct operations to preserve particular features of the input signal, such as the mean value or the first moment [16, 17, 33]. Therefore, below are the descriptions of three primary wavelets.

1. **Haar**: This transformation splits an input signal, S, into non-overlapping elements of odd and even samples, s_{2i+1} and s_{2i}. It applies prediction and update operators, assuming a strong correlation between adjacent values while preserving the signal's mean value [16, 17, 33]. Therefore, the detail coefficients at the j-th decomposition level, D_i^j, are given by the prediction operator $P(\cdot)$:

$$P(S) = s_{2i} \tag{2.9}$$

$$D_i^j = s_{2i+1} - P(S) = s_{2i+1} - s_{2i} \tag{2.10}$$

where $j \in [0, l-1]$, and l the maximum decomposition level. Then, the approximation coefficients, A_i^j, are calculated by the update $W(\cdot)$:

$$W(D) = \frac{D_i^j}{2} \tag{2.11}$$

$$A_i^j = \lfloor s_{2i} + W(D) \rfloor = \left\lfloor \frac{s_{2i} + s_{2i+1}}{2} \right\rfloor \tag{2.12}$$

The floor function guarantees the integer and lossless transformation. Figure 2.3 shows the one-level prediction and update scheme (lifting scheme) over the signal. Additionally, l-decomposition levels can be achieved by operating again over the resulting approximation coefficients.

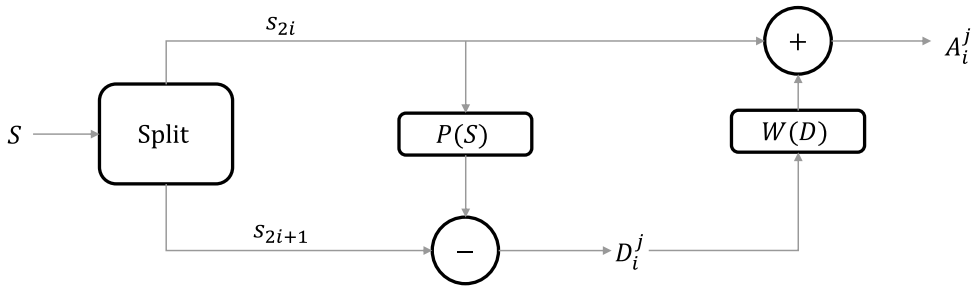

Figure 2.3: Classical lifting Haar scheme.

2. **CDF (2,2)**: The Cohen–Daubechies–Feauveau (CDF) wavelet assumes a piecewise linear input signal to define the lifting operators, preserving the first moment of the data values. Furthermore, the transform is biorthogonal, meaning it is symmetric and has a linear phase [16, 17, 33]. Thus, the detail element, D_i^j, is predicted using $P(\cdot)$:

$$P(S) = \frac{1}{2}\left[s_{2i} + s_{2i+2}\right] \tag{2.13}$$

$$D_i^j = s_{2i+1} - \lfloor P(S) \rfloor \tag{2.14}$$

and the approximation, A_i^j, by means of $W(\cdot)$:

$$W(D) = \frac{1}{4}\left[D_{i-1}^j + D_i^j\right] \tag{2.15}$$

$$A_i^j = s_{2i} + \lfloor W(D) \rfloor \tag{2.16}$$

Figure 2.4 presents the lifting scheme for the CDF (2,2) transform.

3. **DB4**: The Daubechies-4 (DB4) uses multiple prediction and update steps in the lifting scheme to decompose signals. This transformation is based on classical Euclidean factorization, and does not require any signal assumptions. It is an orthogonal transformation that possesses favorable frequency properties [16, 17, 33]. The

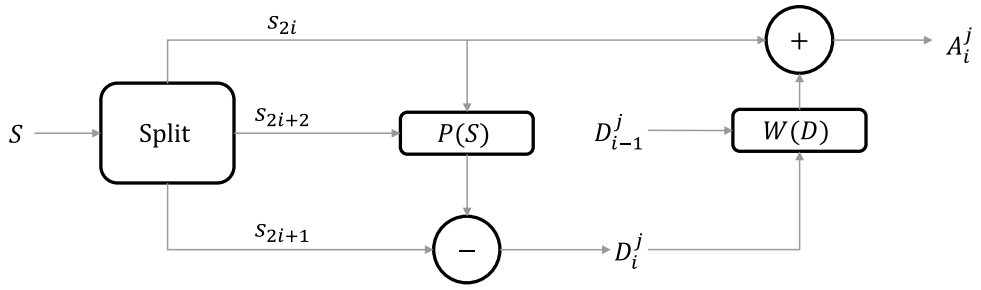

Figure 2.4: Classical lifting CDF scheme.

detail, D_i^j, and approximation coefficient, A_i^j, are described by the operators $P(\cdot)$ and $W(\cdot)$:

$$P(S) = \sqrt{3}s_{2i} \tag{2.17}$$

$$W(D') = \sqrt{3}D'_i + (\sqrt{3}-2)D'_{i-1} \tag{2.18}$$

as

$$D'_i = s_{2i+1} - \lfloor P(S) \rfloor \tag{2.19}$$

$$A_i^j = s_{2i} + \left\lfloor \frac{1}{4} W(D') \right\rfloor \tag{2.20}$$

$$D_i^j = D'_i + A_{i+1}^j \tag{2.21}$$

where D'_i is an auxiliary element, and A_{i+1}^j the approximation coefficient at next iteration. Fig 2.5 depicts the DB4 lifting scheme.

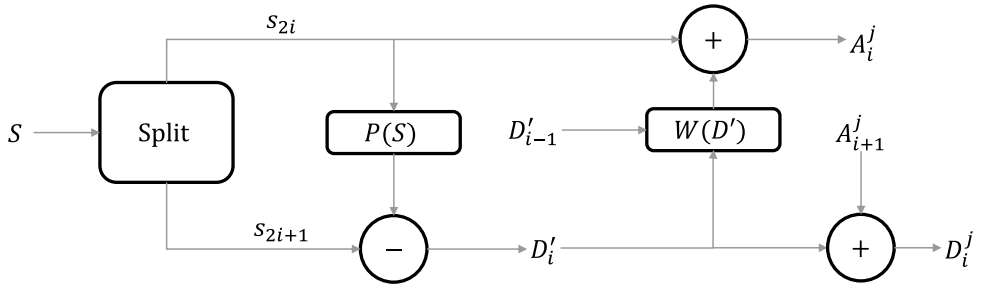

Figure 2.5: Classical lifting DB4 scheme.

2.3 Compression

Data compression enables the representation of information in a condensed format, utilizing the structure of the signals, correlations between values, and redundancies in data. The aim is to reduce the number of bits required for storing and transmitting information [16, 34].

Typically, statistical structures such as repetitive patterns or probabilities of data occurrence are utilized for compression. However, applying domain transformations such as Fourier or Wavelet techniques can aid in identifying and modifying distinct signal components, thus improving data compression. Moreover, human perceptual limitations can be exploited to achieve high compression levels [16, 34].

Finally, based on the reconstruction requirements, compression could be divided into **Lossless Compression**, which preserve the original information and enables recovering all the initial data, and **Lossy Compression**, where the original data is disturbed, losing information but providing high compression ratios [16, 34].

2.3.1 Lossless Compression

This type of compression involves no information loss and allows recovery of all original values from the compressed data, which is useful for applications where information retrieval is critical, such as medical imaging, military tasks and banking transactions [34]. In addition, lossless schemes are evaluated in terms of time and implementation complexity, compression ratio, and information recovery [16, 34].

Data compression could be described in two stages: first, the structure of the input signal is modeled by extracting information about patterns, redundancies or correlation between values. Then, the original data is transformed into a new representation by assigning codes or using different transforms [16, 34]. For example, an input signal, $S_i =$

$\{8, 10, 10, 10, 13, 12, 14, 16, 15, 16\}$, requires five bits per sample to be transmitted or stored. However, the signal could be described with fewer bits by exploiting its internal structure as follows:

- Construct a model of the data as

$$S_i^1 = \{i + 8 \mid i \in [0, 9]\} \tag{2.22}$$

- Calculate the difference between the model and the input signal

$$\begin{aligned} e_i = S_i^1 - S_i =& \{(8-8), (9-10), (10-10), (11-10), (12-13), \\ & (13-12), (14-14), (15-16), (16-15), (17-16)\} \\ =& \{0, -1, 0, 1, -1, 1, 0, -1, 1, 1\} \end{aligned} \tag{2.23}$$

- Assign a code to the residual values, e_i,

$$\begin{aligned} -1 &\rightarrow 00 \\ 0 &\rightarrow 01 \\ 1 &\rightarrow 10 \end{aligned} \tag{2.24}$$

  where only two bits per sample are required.
- Transmit (or store) the model S_i^1 and the residual values, e_i, obtaining a compressed representation.

The previous description shows a basic compression scheme, where the structure of the data is used and the original signal, S_i, can be recovered from the model, S_i^1, and the residual values, e_i, resulting in a lossless process [16, 34].

Finally, using tools such as Fourier or Wavelet transforms helps to exploit the structure of the signals by decomposing the data into different components. Each component can then be analyzed and manipulated separately, providing new views of the information. Therefore, more robust and ad hoc schemes are developed [16, 34].

2.3.2 Quantum Compression

In classical compression, the goal is to compress information data for transmission (increasing channel capacity) and/or storage (reducing the number of bits) by simplifying or condensing the information of a given input signal [35–37]. Thus, quantum compression involves reducing the number of qubits used for storage, limiting the quantum resources (number of quantum gates) to carry the information into the quantum domain, and minimizing the number of measurements at the end of a process. Lossless and lossy techniques, similar to classical methods, are also employed in quantum compression [38–40, 40–42].

On the other hand, there are challenges and issues related to achieving compression in the quantum domain due to the properties and physical limitations of the quantum world [20, 39, 42, 43]. For example, the variable length (non-orthogonality) of codewords in some compression schemes, such as Huffman and Arithmetic coding, cannot be used in the same sense, because quantum non-orthogonality involves non-differentiability of states, that is, given two elements X and Y with codewords $X = |1\rangle$ and $Y = |0\rangle + |1\rangle$, there is no measurement operator that can perfectly distinguish between both states [9, 43]. Also, if we want to recover all the information stored in the quantum domain, we need many copies of the input signal and perform several (exponential) measurements on the states to get all the values, thus obscuring the possible benefits of quantum compression [1, 28, 29]. Additionally, the early stage of quantum noise and error correction techniques has limitations on lossless compression applications [27].

2.4 Summary

This chapter has introduced some notions about quantum computing, including the representation, manipulation, and measurement of quantum information. The qubit (quantum bit) is the basic information unit, which can be in superposed or entangled states,

allowing quantum representation formats to be defined. Quantum operators (gates) are the means to manipulate quantum information through unitary transformations, and quantum measurement allows the recovery of some of the possible computational values. Additionally, we have given the ideas behind wavelet transforms, defining the lifting forms of Haar, CDF(2,2), and Daubechies-4 (DB4) wavelets. Finally, we have presented the basic concepts of classical compression (lossless compression) and the goals of a general quantum compression scheme.

3

Literature Review

This chapter describes work related to this research, including quantum wavelet development, applications of quantum wavelet transforms, and different quantum compression schemes.

3.1 Quantum Wavelet Transforms

Research on quantum wavelet transforms (QWTs) has included Haar and DB4 kernels for single and multilevel decomposition. Additionally, quantum versions for multidimensional (1D, 2D, 3D) and packet analyses have been developed. Hoyer [44] proposed quantum networks for one-dimensional Haar and DB4 using the generalized Kronecker product with time complexity $O(n)$. Fijany [45] used permutation matrices to devise quantum circuits for the Haar and DB4 kernels utilizing the 1D packet and pyramid algorithm, resulting in a complexity of $O(n^2)$. Klappenecker [46] demonstrated a decrease in time complexity compared to the classical method by executing a 1D periodized quantum wavelet packet with $O(\log^2(n))$ operations. Gosal and Lawton [47] formulated quantum algorithms for the multilevel Haar transform. Nevertheless, the current complexity of

the aforementioned quantum definitions is $O(n^3)$, and differs from the complexities presented in the respective papers, as demonstrated by Li et al. [14]. Furthermore, Li et al. [48] presented iteration equations for direct and inverse QWTs using generalized tensor product, achieving a complexity of $O(n^3)$ utilizing the generalized tensor product and permutation matrices. Li et al. [14] constructed multilevel 2D quantum transforms with a time complexity of $O(n^3)$. Li et al. [15] proposed a quantum wavelet packet using generalized tensor products, permutation matrices, and periodization extension for multidimensional (1D, 2D, 3D) and multilevel systems, resulting in a quantum complexity of $O(n^3)$. Li et al. [10] also developed multilevel 3D quantum Haar and DB4 wavelet transforms with a complexity of $O(n^3)$. To the best of our knowledge, there are no quantum definitions available for other wavelet bases or integer versions of wavelet transforms. Table 3.1 shows the characteristics of the quantum wavelets in the literature and the proposed transforms, including the transform domain, implementation method, signal dimension, transform class, and complexity for a signal of size $N = 2^{d \times n}$, where d denotes the signal dimension.

Table 3.1: QWTs development for the literature review and the proposed solutions.

	Literature [10, 14, 15, 44–48] (1997-2023)		**Proposed**		
Wavelet	Haar	DB4	Haar	CDF	DB4
Domain	Real-Valued	Real-Valued	Integer	Integer	Integer
Method	Filter	Filter	Lifting	Lifting	Lifting
Dimension	1D, 2D, 3D	1D, 2D, 3D	1D	1D	1D
Class	Orthogonal	Orthogonal	Orthogonal	Bi-orthogonal	Orthogonal
Complexity	$O(n^3)$	$O(n^3)$	$\boldsymbol{O(qn)}$	$\boldsymbol{O(qn)}$	$\boldsymbol{O(q^2n)}$

3.2 Quantum Wavelet and Compression Applications

Applications of quantum wavelet include watermarking, encryption, compression, and denoising. Song et al. [49] developed a dynamic watermarking scheme based on quantum DB4 wavelet transform, controlling the embedding strength by a dynamic vector. Heidari et al. [50] proposed a watermarking based on quantum wavelet transform, including a scrambling step to enhance the security. Hu et al. [51] designed a quantum image watermarking based on Haar wavelet, using the diagonal coefficients to insert the watermark information in the frequency domain. Yu et al. [52] presented an adaptive LSB quantum image watermarking using Haar wavelet transform in the frequency domain. Wang et al. [53] implemented a quantum image encryption based on quantum DB4 wavelet and diffusion method using chaotic maps. Wang et al. [54] developed an adaptive quantum image encryption scheme by encrypting the low frequency information into the quantum DB4 wavelet coefficients. Li et al. [55] presented a lossy compression scheme based on thresholding and quantum Haar transform using NASS quantum representation. Chakraborty et al. [13] proposed an image denoising scheme based on quantum DB4 wavelet transform and thresholding technique including hard and soft thresholds. Figure 3.2 presents the different applications using real-valued quantum Haar and DB4 wavelet transforms.

Table 3.2: QWTs applications

Application	Watermarking [49–52]	Encryption [53, 54]	Compression [55]	Denoising [13]
Wavelet	DB4, Haar	DB4	Haar	DB4

On the other hand, research on quantum compression involves different methods to decrease the number of qubits, gates and measurements. Haque et. al [56] presented a new quantum representation to reduce the number of qubits and quantum gates. In addition, a lossy method based on Block Truncation Coding (BTC) is described. Ma et al. [36] used Quantum Haar Transform (QHT) and Quantum Fibonacci Transform

(FibT) with the Generalized Quantum Image Representation (GQIR), reaching compression by a measurement matrix. Haque et al. [35] decreased the number of image operations to prepare a quantum image, combining the classical Discrete Cosine Transform (DCT) with Quantum Storage (QS). Zhou et. al [57] used the Discrete Quantum Wavelet Transform (DQWT) for encrytion and compression. Pang et. al [38] developed a lossy quantum compression scheme based on quantum DCT and Grover Algorithm (GA) with a quantization method. Li et. al [41] presented a compression model based on Quantum Cosine Transform (QCT) and Zig-Zag Encoding (ZZE). Jiang et. al [40] designed a hybrid compression scheme using classical DCT and storing the coefficients into the quantum domain. Jiang et. al [37] developed a lossy compression to reduce the preparation complexity to store information into the quantum domain by classical Differential Pulse Code Modulation (DPCM) and Quantum Point Cloud (QPC). Li et. al [58] implemented a lossless compression based on Run-Length-Encoding (RLE), decreasing qubit complexity. Rogers et. al [39] defined a lossless compression using Indeterminate Length States (ILS) to define new codewords for each signal element. Table 3.3 shows the characteristics of compression models in the literature and the proposed scheme based on Fixed-Length Coding (FLC) and Quantum Wavelt Transform (QWT), including scheme type, compression method, and domain area.

3.3 Summary

In this chapter, we have presented the related work on the definition of quantum versions of the wavelet Haar and Daubechies-4 (DB4) transforms, including a comparative analysis of the characteristics of the literature and the proposed versions. The proposed quantum transforms include the CDF(2,2) wavelet, which is the first quantum Bio-orthogonal wavelet transforms. Additionally, the integer quantum wavelets decrease time complexity compared to real-valued transforms. Also, we have described applications of quantum wavelet to watermarking, encryption, compression, and denoising, being information hiding the main application. Furthermore, we have discussed different quantum com-

Table 3.3: Quantum compression schemes.

Reference	Scheme	Method	Domain
Proposed	Lossless	FLC, QWT	Hybrid
[56] (2023)	Lossy	BTC	Quantum
[36] (2023)	Lossless	QHT, FibT, GQIR	Quantum
[35] (2023)	Lossy	DCT, QS	Hybrid
[57] (2020)	Lossy	DQWT	Quantum
[38] (2019)	Lossy	QDCT, GA	Quantum
[41] (2018)	Lossless	QCT, ZZE	Quantum
[40] (2018)	Lossy	DCT	Hybrid
[37] (2017)	Lossy	DPCM	Hybrid
[58] (2013)	Lossless	RLE	Quantum
[39] (2011)	Lossless	ILS	Quantum

pression models, showing the scheme type, the compression method, and the domain area compared to the proposed scheme, where we use the proposed quantum transforms with a Fixed-Length Coding to develop a hybrid compression scheme.

4

Proposed Solution

This chapter presents quantum wavelet definitions of the Haar, CDF, and DB4 transforms, along with circuit designs for addition, subtraction, multiplication, halving, and rounding operations. It also includes a quantum storage definition for representing one- or two-dimensional signals as block components, as well as unitary descriptions and algorithmic formulations of the proposed wavelet transforms. Finally, it discusses their applications in lossless compression by developing a hybrid scheme based on quantum wavelet transforms and Fixed-Length Coding.

4.1 Quantum Circuit Design

Given the classical descriptions of the lifting wavelet transforms (Chapter 2), we design quantum circuits for the addition, subtraction, swap, halving, rounding, cyclic shift, and multiplication operations involved in the quantum versions of the Haar, DB4, and CDF transforms. In addition, we develop two new operations, $U_{\sqrt{3}}$ and $U_{\sqrt{3}-2}$, for the DB4 transform. All the previous operations ensure the linearity and reversibility of the decomposition process. Furthermore, each operation has polynomial quantum complexity.

4.1.1 Quantum Addition

Quantum full addition operation of two binary numbers, $|a\rangle = |a_{q-1}, \ldots, a_1, a_0\rangle$ and $|b\rangle = |b_{q-1}, \ldots, b_1, b_0\rangle$, is given by

$$U_a|0\ldots0\rangle|b\rangle|a\rangle = |carry, a_{q-1} + b_{q-1}, \ldots, a_1 + b_1, a_0 + b_0\rangle|b\rangle|a\rangle \tag{4.1}$$

where U_a is the adder operator, q is the bit precision, and $|0\ldots0\rangle$ is an auxiliary register to store the addition results and the carry bit [59]. Fig 4.1 presents the quantum circuit for adding two binary numbers.

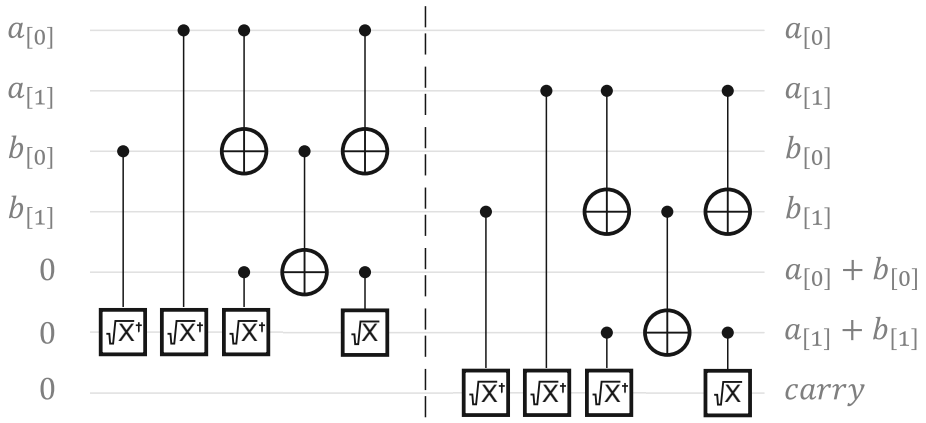

Figure 4.1: Quantum addition circuit for two binary numbers, $|a + b\rangle = |a_1a_0 + b_1b_0\rangle$.

4.1.2 Quantum Conditional Addition

A quantum conditional adder performs an addition operation on two binary registers $|a\rangle$ and $|b\rangle$ only if a control qubit, $|c\rangle$, is in the state $|1\rangle$, defined as

$$CU_a|00\rangle|b\rangle|a\rangle|c\rangle = \begin{cases} |00\rangle|b\rangle|a\rangle|0\rangle & , \ |c\rangle = |0\rangle \\ |0, carry\rangle|a_{q-1} + b_{q-1}, \ldots, a_0 + b_0\rangle|a\rangle|1\rangle & , \ |c\rangle = |1\rangle \end{cases} \tag{4.2}$$

where CU_a is the conditional operator, $|c\rangle$ is the control bit, the addition results are stored in $|b\rangle$, and the carry bit in the auxiliary register [60]. Figure 4.2 illustrates the quantum circuit for adding two binary numbers with bit precision $q = 3$.

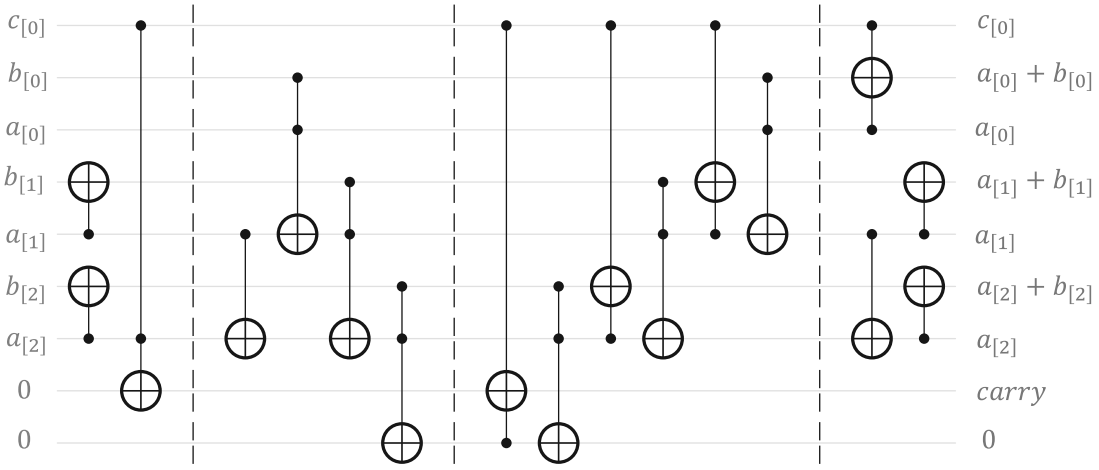

Figure 4.2: Quantum conditional addition circuit of two binary numbers, $CU_a|ab\rangle = |a_2a_1a_0 + b_2b_1b_0\rangle$.

4.1.3 Quantum Subtraction

Quantum full subtraction is described by

$$U_s|0\ldots0\rangle|b\rangle|a\rangle = |borrow, \ldots, garbage\rangle|b_{q-1} - a_{q-1}, \ldots, b_1 - a_1, b_0 - a_0\rangle|a\rangle \quad (4.3)$$

where U_s is the subtractor operator, $|b\rangle$ stores the binary subtraction, and the auxiliary register contains garbage data and the borrow bit [61]. Fig 4.3 shows the quantum circuit, where the part B must be cascaded to subtract more than two bits ($q > 2$).

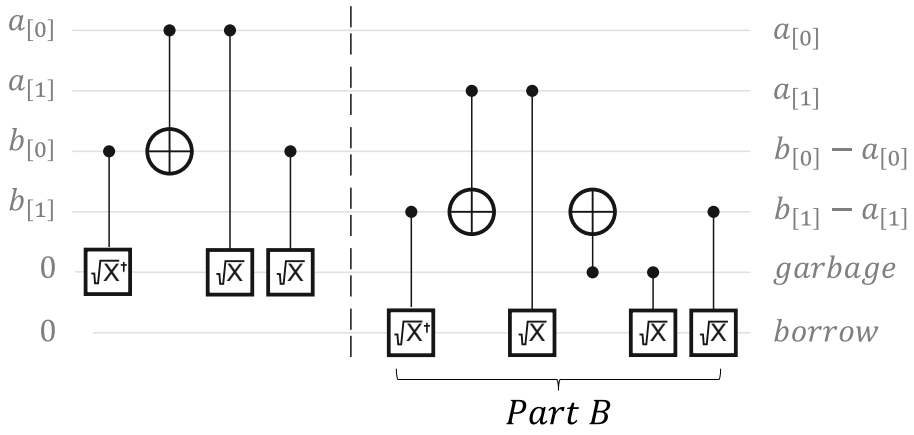

Figure 4.3: Quantum subtraction of two binary numbers, $U_s|ba\rangle = |b_1b_0 - a_1a_0\rangle$.

4.1.4 Quantum Swap

The quantum swap operation moves the information value from one qubit register to another as

$$U_{swap}|a_{i+1}\rangle|a_i\rangle = |a_i\rangle|a_{i+1}\rangle \tag{4.4}$$

where U_{swap} is the swap operator. In addition, it can move the bit values in a register by applying the U_{swap} operator multiple times [20]. Figure 4.4 illustrates the quantum circuit, where both implementations are equivalent [20].

4.1.5 Quantum Halving

Quantum halving operation on a register $|a\rangle = |a_{q-1}, \ldots, a_1, a_0\rangle$ is defined by

$$U_h|0\rangle|a\rangle = |0\rangle|a_{q-1}, \ldots, a_2, a_1\rangle|a_0\rangle = |a/2\rangle|a_0\rangle \tag{4.5}$$

where the qubit $|a_0\rangle$ is discarded, and $|0\rangle|a_{q-1}, \ldots, a_2, a_1\rangle$ is the halved register. Fig 4.5 depicts the quantum halving circuit by identity gates.

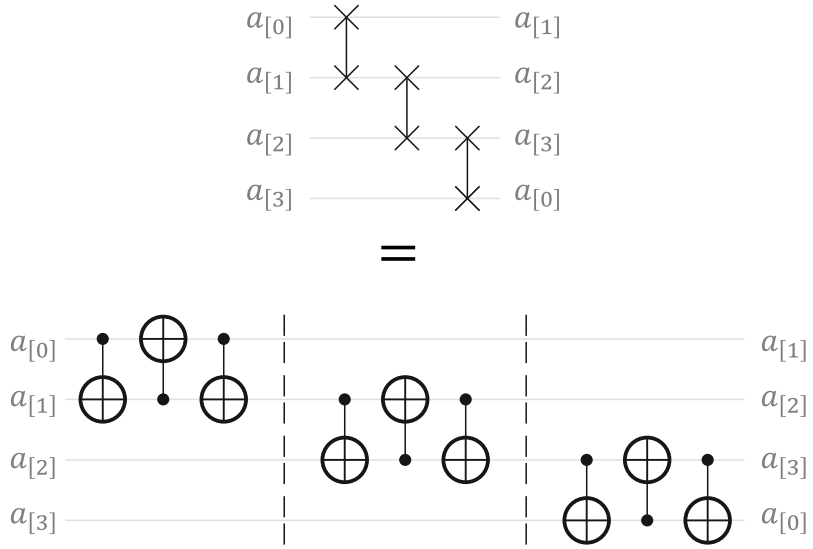

Figure 4.4: Quantum swap circuit, $U_{swap}|a_3a_2a_1a_0\rangle = |a_0a_3a_2a_1\rangle$.

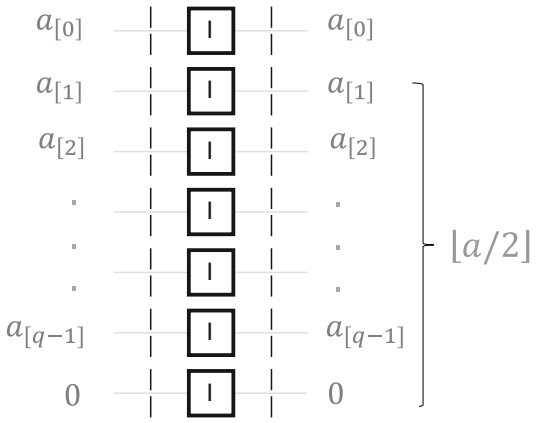

Figure 4.5: Quantum halving circuit, $U_h|a_{q-1}, \ldots, a_1, a_0\rangle = |0, a_{q-1}, \ldots, a_1\rangle$.

4.1.6 Quantum Rounding

Quantum rounding uses the halving properties of integers to avoid the nonlinearities in the operation [18]. Given $a = 2m$, with $m \in \mathbb{Z}$, the rounding operation is expressed as

$$\left\lfloor \frac{a}{2} \right\rfloor = \left\lfloor \frac{2m}{2} \right\rfloor = \lfloor m \rfloor = m \tag{4.6}$$

On the other hand, when $a = 2m + 1$,

$$\left\lfloor \frac{2m+1}{2} \right\rfloor = \left\lfloor \frac{2m}{2} + \frac{1}{2} \right\rfloor = \lfloor m \rfloor = m \tag{4.7}$$

where the additional term, $1/2$, does not contribute to the floor function. Therefore, the rounding operation can be modified by subtracting 1 before the halving operation when $a = 2m + 1$, giving the same result:

$$\left\lfloor \frac{a-1}{2} \right\rfloor = \left\lfloor \frac{2m}{2} \right\rfloor = m \tag{4.8}$$

Thus, a quantum rounding operation is defined based on (4.6) and (4.8), giving

$$U_r|a\rangle = |\lfloor a \rfloor\rangle = |\lfloor \frac{a}{2} \rfloor\rangle \leftrightarrow a \in \mathbb{Z} \tag{4.9}$$

where this operation is implemented by the halving operation in (4.5).

4.1.7 Quantum Cyclic Shift

Quantum cyclic shift cyclically increases (or decreases) the binary representation of a quantum register by adding (or subtracting) a bit value to the register, given by

$$U_{cs+}|a_{q-1}, \ldots, a_1, a_0\rangle = |(a_{q-1}, \ldots, a_1, a_0 + 1) \mod 2^q\rangle \tag{4.10}$$

$$U_{cs-}|a_{q-1}, \ldots, a_0, a_1\rangle = |(a_{q-1}, \ldots, a_0, a_1 - 1) \mod 2^q\rangle \tag{4.11}$$

where q is the bit precision, U_{cs+} and U_{cs-} are the shift operators to the right and left, respectively [62]. For example, applying U_{cs+} on $|a\rangle = |001\rangle$ gives $U_{cs+}|a\rangle = |010\rangle$. Figure 4.6 illustrates the quantum circuits for each operation.

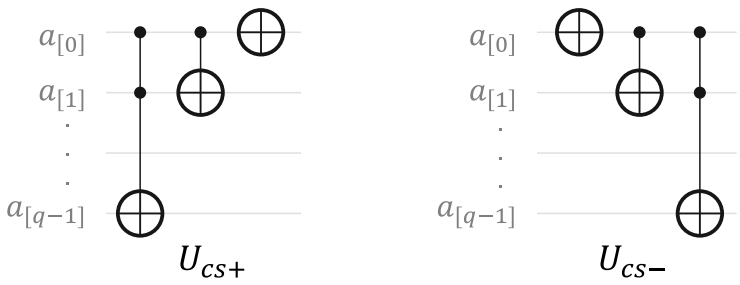

Figure 4.6: Quantum cyclic shift circuits.

4.1.8 Quantum Multiplication

Quantum multiplication returns the product of two binary numbers as

$$U_m|b\rangle|0\ldots0\rangle|a\rangle = |b\rangle|a \times b\rangle|a\rangle \tag{4.12}$$

where U_m is the quantum operator, the product is stored in the $|0\ldots0\rangle$ register [63]. This operation uses conditional addition and swap operation to obtain the product results. Figure 4.7 shows the block quantum circuit, where the $|a\rangle$ register acts as a control bit for the conditional adder, CU_a, which needs $(q+1)$ auxiliary qubits, and $(q-1)$ for the swap operations.

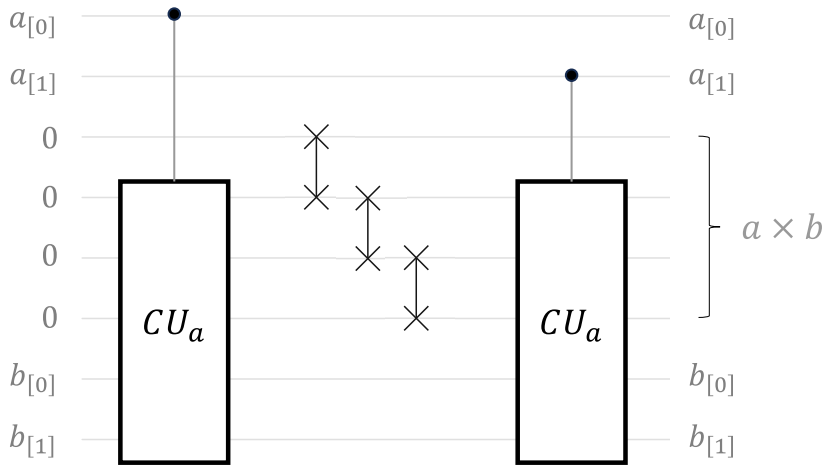

Figure 4.7: Quantum multiplication circuit of two binary registers, $U_m|ba\rangle = |a \times b\rangle$.

4.1.9 Quantum Special Operators

Quantum $U_{\sqrt{3}}$ and $U_{\sqrt{3}-2}$ are special operators included in the DB4 transform, which perform the product of $\sqrt{3}$ and $\sqrt{3}-2$ in (2.17) and (2.18). These constant values are approximated by

$$\begin{aligned}(\sqrt{3}) &\approx 17 \times (0.1)\\ (\sqrt{3}-2) &\approx -3 \times (0.1)\end{aligned} \tag{4.13}$$

Thus,

$$\begin{aligned}U_{\sqrt{3}}|a\rangle &= |(\sqrt{3})a\rangle = |17a \times 0.1\rangle\\ U_{\sqrt{3}-2}|a\rangle &= |(\sqrt{3}-2)a\rangle = |-3a \times 0.1\rangle\end{aligned} \tag{4.14}$$

where the operators $U_{\sqrt{3}}$ and $U_{\sqrt{3}-2}$ are defined by quantum addition, U_a, and multiplication, U_m.

4.2 Quantum Representation

The first step in developing quantum solutions is to define the appropriate representation model to store and then manipulate the information data. Thus, based on the QBS model (Chapter 2), we define a new quantum representation format called Quantum Block Representation by Basis States (QBRBS) to store the signal information in block elements of arbitrary size, facilitating signal manipulation and avoiding the generation of garbage information due to some operations [26, 64]. The QBRBS format is defined as

$$|S\rangle = \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} \left[\bigotimes_{j=0}^{m-1} |s_{v_j(i)+w_j}\rangle \right] |X_i\rangle \tag{4.15}$$

where k is the number of blocks, m is the number of elements per block, $|s_{(\cdot)}\rangle$ are the signal values, $|X_i\rangle$ is the coordinate position, and the coefficients $\{v_j, w_j\}$ determine the elements for each block. This format requires $(mq + \lceil \log k \rceil)$ qubits to store a 2^n signal

with q-bit precision [18, 64]. For example, we can store the odd/even components of a signal by

$$\begin{aligned} |S\rangle &= \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} \left[\bigotimes_{j=0}^{1} |s_{v_j(i)+w_j}\rangle \right] |X_i\rangle \\ &= \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} |s_{v_0(i)+w_0}\rangle |s_{v_1(i)+w_1}\rangle |X_i\rangle \\ &= \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} |s_{2i+1}\rangle |s_{2i}\rangle |X_i\rangle \end{aligned} \tag{4.16}$$

where $m = 2$ because only two elements are stored per block. Also, $v_0 = v_1 = 2$, $w_0 = 1$ and $w_1 = 0$ are employed to obtain the odd and even samples of the input signal, $|S\rangle$, stored at the same position, $|X_i\rangle$. On the other hand, we could store three-or more-different signals, $|S^1\rangle$, $|S^2\rangle$ and $|S^3\rangle$, sharing the same position coordinate as

$$|S\rangle = \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} |S_i^3\rangle |S_i^2\rangle |S_i^1\rangle |X_i\rangle \tag{4.17}$$

where the QBRBS format allows for simultaneous operations on all the signals. This is achieved by storing different elements in the same position coordinate, rather than using separate representations for each component. Such an approach eliminate the need for additional operations, such as quantum comparators to locate equal positions, and conditional operators to handle the corresponding elements [64, 65].

4.3 Quantum Wavelet Definitions

Based on the operations in (Sec. 4.1) and the classical descriptions in (Chapter 2), we present the unitary definitions of the quantum Haar, CDF, and DB4 wavelet transforms for $j \in [0, l-1]$ decomposition levels, with l the maximum level, as well as the general block circuits.

4.3.1 Quantum Haar

Quantum Haar transform uses odd/even elements, $|s_{2i}\rangle$ and $|s_{2i+1}\rangle$, of a input signal, $|S\rangle$, to get the approximation, $|A_i^j\rangle$, and detail, $|D_i^j\rangle$, coefficients. This transform includes quantum addition, U_a, subtraction, U_s, and rounding, U_r, operators. Therefore, the quantum Haar is defined by

$$U_{r(2)}U_{a(123)}|s_{2i+1}, \phi_0, s_{2i}\rangle \rightarrow |s_{2i+1}, A_i^j, s_{2i}\rangle \tag{4.18}$$

$$U_{s(12)}|s_{2i+1}, s_{2i}\rangle \rightarrow |D_i^j, s_{2i}\rangle \tag{4.19}$$

where the numerical subscript in the operators indicates the qubit they operate on, with the qubit position counted from left to right. Figure 4.8 shows the general circuit for one decomposition level, using ϕ_0 as an auxiliary register $|0 \ldots 0\rangle$. Furthermore, l-decomposition levels can be achieved by operating again over the resulting approximation coefficients (see Algorithm 1).

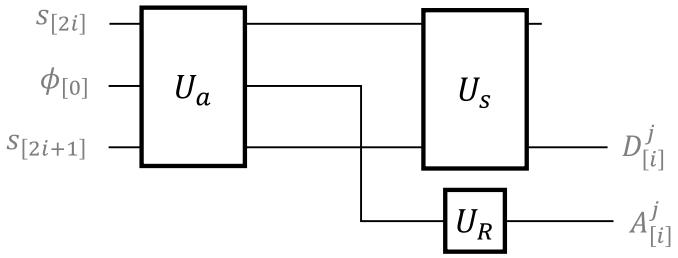

Figure 4.8: Quantum block circuit of the Haar transform.

4.3.2 Quantum CDF (2,2)

Quantum CDF decompose an input signal, $|S\rangle$, into approximation, $|A_i^j\rangle$ and detail, $|D_i^j\rangle$ coefficients by using the $|s_{2i}\rangle$, $|s_{2i+1}\rangle$ and $|s_{2i+2}\rangle$ signal elements. However, this transform also requires the term $|D_{i-1}^j\rangle$ in (2.15), but computing $|D_{i-1}^j\rangle$ needs the $|s_{2i-1}\rangle$ and $|s_{2i-2}\rangle$ values, which are not defined when $i = 0$. Therefore, we use zero padding to

obtain the values at that index[1]. Finally, the quantum CDF is described by

$$U_{s(13)}U_{r(3)}U_{a(234)}|s_{2i+1}, s_{2i+2}, \phi_0, s_{2i}\rangle \rightarrow |D_i^j, s_{2i+2}, a_0^r, s_{2i}\rangle \tag{4.20}$$

$$U_{s(13)}U_{r(3)}U_{a(234)}|s_{2i-1}, s_{2i-2}, \phi_1, s_{2i}\rangle \rightarrow |D_{i-1}^j, s_{2i-2}, a_1^r, s_{2i}\rangle \tag{4.21}$$

$$U_{a(245)}U_{r(2)}^2U_{a(123)}\ |D_{i-1}^j, \phi_2, D_i^j, \phi_3, s_{2i}\rangle \rightarrow |D_{i-1}^j, a_2^{2r}, D_i^j, A_i^j, s_{2i}\rangle \tag{4.22}$$

where ϕ_i are auxiliary registers $|0\ldots0\rangle$, and a_i^r are the results after rounding operations (Algorithm 2). Figure 4.9 depicts the general quantum circuit for one decomposition level.

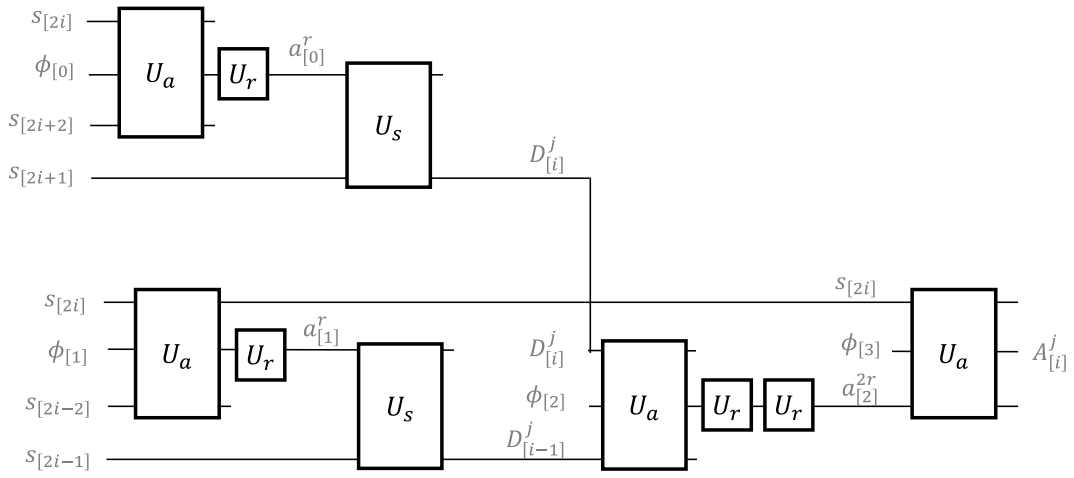

Figure 4.9: Quantum block circuit of the CDF(2,2) wavelet transform.

4.3.3 Quantum DB4

Quantum DB4 uses signal elements $|s_{2i}\rangle$, $|s_{2i+1}\rangle$, $|s_{2i-2}\rangle$, and $|s_{2i-1}\rangle$ to obtain the decomposition coefficients, $|A_i^j\rangle$ and $|D_i^j\rangle$. This transformation requires two supplementary values, $\sqrt{3}$ and $(\sqrt{3}-2)$, which are approximated by the operators $U_{\sqrt{3}}$ and $U_{\sqrt{3}-2}$. Therefore, we rewrite (2.17) and (2.18) as

$$P(S) = 17s_{2i} \times (0.1) \tag{4.23}$$

[1]Other methods exists, but they are beyond the scope of this research [16, 34].

$$W(D') = (17D'_i - 3D'_{i-1}) \times (0.1) \tag{4.24}$$

Thus, quantum DB4 is given by

$$U_{s(12)}U_{r(2)}U_{\sqrt{3}(23)}|s_{2i+1}, \phi_0, s_{2i}\rangle \rightarrow |D'_i, a^r_0, s_{2i}\rangle \tag{4.25}$$

$$U_{s(12)}U_{r(2)}U_{\sqrt{3}(23)}|s_{2i-1}, \phi_1, s_{2i-2}\rangle \rightarrow |D'_{i-1}, a^r_1, S_{2i-2}\rangle \tag{4.26}$$

$$U^2_{r(3)}U_{a(234)}U_{\sqrt{3}(45)}U_{\sqrt{3}-2(12)}|D'_{i-1}, \phi_4, \phi_3, \phi_2, D'_i\rangle \rightarrow |D'_{i-1}, \phi'_4, a^{2r}_2, \phi'_2, D'_i\rangle \tag{4.27}$$

$$U_{a(123)}|a^{2r}_2, \phi_5, s_{2i}\rangle \rightarrow |a^{2r}_2, A^j_i, s_{2i}\rangle \tag{4.28}$$

$$U_{a(123)}|D'_i, \phi_6, A^j_{i+1}\rangle \rightarrow |D'_i, D^j_i, A^j_{i+1}\rangle \tag{4.29}$$

where $|\phi'_2\rangle$ and $|\phi'_4\rangle$ are the values after the application of the operators $U_{\sqrt{3}}$ and $U_{\sqrt{3}-2}$. In addition, $|D'_i\rangle$ and $|D'_{i-1}\rangle$ are auxiliary registers, and $|A^j_{i+1}\rangle$ is the approximation coefficient at next step (Algorithm 3). Figure 4.10 illustrates the general circuit of the quantum DB4 for one decomposition level.

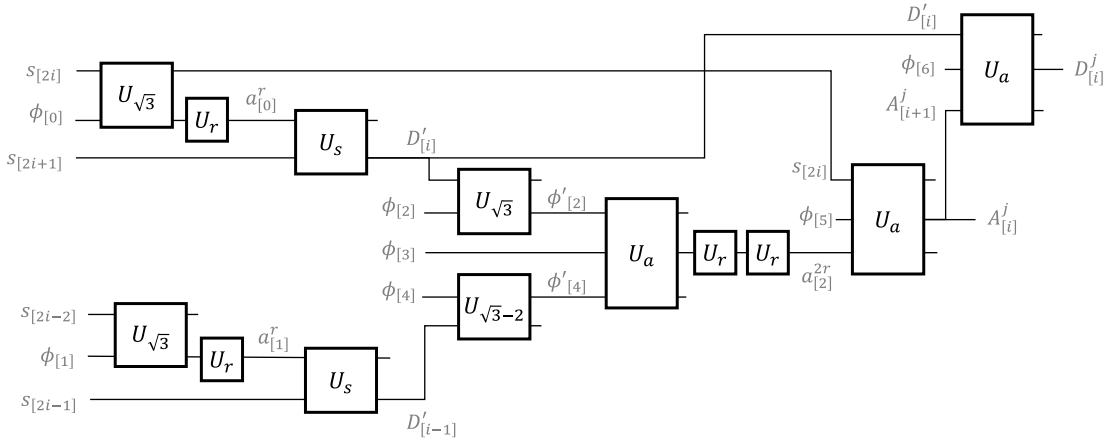

Figure 4.10: Quantum block circuit of the DB4 wavelet transform.

4.4 Quantum Decomposition Algorithms

The first step to obtain the decomposition coefficients $|A^l_i\rangle$ and $|D^l_i\rangle$ is to store pairs of elements of an input signal S in the quantum domain using the proposed QBRBS format.

However, if we want to reach l-decomposition levels, we need to split and store the input signal in the following way:

- Get the input signal S

$$S = \{s_0, s_1, \ldots, s_{N-1}\},\ N = 2^n$$

- Split S into $N/2$ subsets of odd and even elements

$$\Big\{\{s_{2i}, s_{2i+1}\}\Big\}_{i\in[0,N/2-1]} = \Big\{\{s_0, s_1\}, \{s_2, s_3\}, \ldots, \{s_{N-2}, s_{N-1}\}\Big\}$$

- Generate 2^{l-1} new subsets from the previous set

$$\begin{aligned}
S^1 =& \Big\{\{s_0, s_1\}, \{s_{2^l}, s_{2^l+1}\}, \{s_{2(2^l)}, s_{2(2^l)+1}\}, \{s_{3(2^l)}, s_{3(2^l)+1}\}, \ldots \\
& \ldots, \{s_{N-2^l}, s_{N-2^l+1}\}\Big\} \\
=& \Big\{S_0^1, S_1^1, \ldots, S_{N/2^l-1}^1\Big\} \\
S^2 =& \Big\{\{s_2, s_3\}, \{s_{2^l+2}, s_{2^l+3}\}, \{s_{2(2^l)+2}, s_{2(2^l)+3}\}, \{s_{3(2^l)+2}, s_{3(2^l)+3}\}, \ldots \\
& \ldots, \{s_{N-2^l+2}, s_{N-2^l+3}\}\Big\} \\
=& \Big\{S_0^2, S_1^2, \ldots, S_{N/2^l-1}^2\Big\} \\
S^3 =& \Big\{\{s_4, s_5\}, \{s_{2^l+4}, s_{2^l+5}\}, \{s_{2(2^l)+4}, s_{2(2^l)+5}\}, \{s_{3(2^l)+4}, s_{3(2^l)+5}\}, \ldots \\
& \ldots, \{s_{N-2^l+4}, s_{N-2^l+5}\}\Big\} \\
=& \Big\{S_0^3, S_1^3, \ldots, s_{N/2^l-1}^3\Big\} \\
& \vdots \\
s^{2^{l-1}} =& \Big\{\{s_{2^l-2}, s_{2^l-1}\}, \ldots, \{s_{3(2^l)+2^l-2}, s_{3(2^l)+2^l-1}\}, \ldots, \{s_{N-2}, s_{N-1}\}\Big\} \\
=& \Big\{S_0^{2^l-1}, S_1^{2^l-1}, \ldots, S_{N/2^l-1}^{2^l-1}\Big\}
\end{aligned}$$

- Store each subset S^j into the QBRBS format

$$\begin{aligned}|F\rangle &= \frac{1}{\sqrt{N/2^l}} \sum_{i=0}^{N/2^l-1} \left[\bigotimes_{j=1}^{2^{l-1}} |S_i^j\rangle\right] |X_i\rangle \\ &= \frac{1}{\sqrt{N/2^l}} \Big[|S_0^1\rangle|S_0^2\rangle \dots |S_0^{2^{l-1}}\rangle|X_0\rangle + |S_1^1\rangle|S_1^2\rangle \dots |S_1^{2^{l-1}}\rangle|X_1\rangle + \dots \\ &\qquad + |S_{N/2^l-1}^1\rangle|S_{N/2^l-1}^2\rangle \dots |S_{N/2^l-1}^{2^{l-1}}\rangle|X_{N/2^l-1}\rangle\Big]\end{aligned} \tag{4.30}$$

Previous description outlines how to store the signal components to achieve l-levels of decomposition using the proposed quantum transforms, where the operations are performed to pairs of elements from left to right. For example, given a signal, S, of size $N = 16$ with two decomposition levels, $l = \{1, 0\}$, and U_{WT} a "wavelet operator" acting on pairs of elements, we have

$$\begin{aligned}|F\rangle &= \frac{1}{\sqrt{4}} \sum_{i=0}^{3} \left[\bigotimes_{j=1}^{2} |S_i^j\rangle\right] |X_i\rangle \\ &= \frac{1}{2}\Big[|S_0^1\rangle|S_0^2\rangle|X_0\rangle + |S_1^1\rangle|S_1^2\rangle|X_1\rangle + |S_2^1\rangle|S_2^2\rangle|X_2\rangle + |S_3^1\rangle|S_3^2\rangle|X_3\rangle\Big]\end{aligned}$$

Then, replace the values for each S_i^j to get the odd and even components of the signal

$$\begin{aligned}|F\rangle =& \frac{1}{2}\Big[|s_0 s_1\rangle|s_2 s_3\rangle|X_0\rangle + |s_4 s_5\rangle|s_6 s_7\rangle|X_1\rangle + \\ &+ |s_8 s_9\rangle|s_{10} s_{11}\rangle|X_2\rangle + |s_{12} s_{13}\rangle|s_{14} s_{15}\rangle|X_3\rangle\Big]\end{aligned}$$

Now apply the operator U_{WT} to each pair of registers from left to right to reach a first decomposition level, $|F^1\rangle$

$$\begin{aligned}|F^1\rangle = U_{WT}|F\rangle =& \frac{1}{2}\Big[U_{WT}|s_0 s_1\rangle U_{WT}|s_2 s_3\rangle|X_0\rangle + U_{WT}|s_4 s_5\rangle U_{WT}|s_6 s_7\rangle|X_1\rangle + \\ &+ U_{WT}|s_8 s_9\rangle U_{WT}|s_{10} s_{11}\rangle|X_2\rangle + |s_{12} s_{13}\rangle U_{WT}|s_{14} s_{15}\rangle|X_3\rangle\Big] \\ =& \frac{1}{2}\Big[|C_0^1\rangle|C_1^1\rangle|X_0\rangle + |C_2^1\rangle|C_3^1\rangle|X_1\rangle + |C_4^1\rangle|C_5^1\rangle|X_2\rangle + |C_6^1\rangle|C_7^1\rangle|X_3\rangle\Big]\end{aligned}$$

where $|C_j^l\rangle = U_{WT}|s_{2j} s_{2j+1}\rangle$ are the decomposition coefficients of the first level $l = 1$. Finally, perform the U_{WT} operator again on the new pairs of elements to obtain the second decomposition level, $|F^0\rangle$

$$|F^0\rangle = U_{WT}|F^1\rangle = \frac{1}{2}\Big[U_{WT}|C_0^1 C_1^1\rangle|X_0\rangle + U_{WT}|C_2^1 C_3^1\rangle|X_1\rangle +$$

$$+ U_{WT}|C_4^1 C_5^1\rangle|X_2\rangle + U_{WT}|C_6^1 C_7^1\rangle|X_3\rangle\Big]$$
$$= \frac{1}{2}\Big[|C_0^0\rangle|X_0\rangle + |C_1^0\rangle|X_1\rangle + |C_2^0\rangle|X_2\rangle + |C_3^0\rangle|X_3\rangle\Big]$$

where $|C_k^l\rangle = U_{WT}|C_{2k}^1 C_{2k+1}^1\rangle$ are the coefficients at the second level $l = 0$. Figure 4.11 illustrates the whole process, where we can only operate on pairs of elements in the same qubit position, $|X_i\rangle$.

Finally, the quantum decomposition algorithms for each wavelet transform (Haar, CDF, DB4) are presented below, where we need to repeat the process on the approximation coefficients, $|A_i^j\rangle$, to get l-decomposition levels, with $l \leq n$.

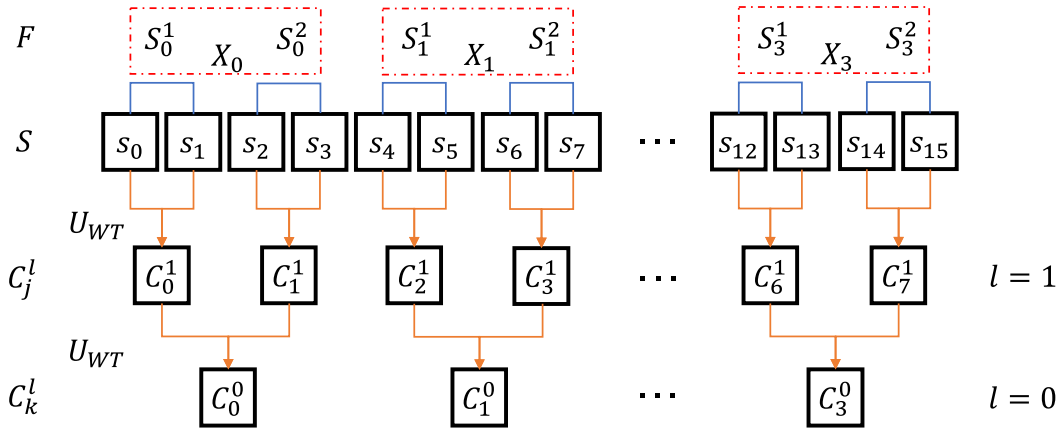

Figure 4.11: Quantum general decomposition process using the QBRBS format. The elements C_k^l do not share the same coordinate position, therefore, we cannot reach further levels of decomposition.

Algorithm 1 : Quantum Haar Transform - Approximation Coefficients

▷ ***Input***: One-dimensional signal $|S\rangle$ of 2^n.

▷ ***Output***: Decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$. ▷ With $j \leq l$

procedure

Step 1: State initialization

$|0 \ldots 0\rangle = |s_{2i+1}, s_{2i}\rangle$

$|0 \ldots 0\rangle = |\phi_0\rangle$ ▷ Auxiliary qubits

Step 2: Generate a qubit register

$|s_{2i+1}, \phi_0, s_{2i}\rangle$

▷ **Step 2.1:** Apply the addition and rounding operator $U_{r(2)}U_{a(123)}$

These operations produce the state,

$|s_{2i+1}, A_i^j, s_{2i}\rangle$

Step 3: Generate a second qubit register

$|s_{2i+1}, s_{2i}\rangle$

▷ **Step 3.1:** Apply the subtraction operator $U_{s(12)}$

This gives,

$|D_i^j, s_{2i}\rangle$

Step 4: Extract the decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$

▷ Repeat on $|A_i^j\rangle$ to get l-decomposition levels.

end procedure

Algorithm 2 : Quantum CDF Transform - Decomposition Process

▷ ***Input***: One-dimensional signal $|S\rangle$ of $2^n + 1$.

▷ ***Output***: Decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$. ▷ With $j \leq l$

procedure

Step 1: State initialization

$|0 \dots 0\rangle = |s_{2i-1}, s_{2i-2}, s_{2i+1}, s_{2i+2}, s_{2i}\rangle$

$|0 \dots 0\rangle = |\phi_3, \phi_2, \phi_1, \phi_0\rangle$ ▷ Auxiliary qubits

Step 2: Select the first block of elements

$|s_{2i+1}, s_{2i+2}, \phi_0, s_{2i}\rangle$

▷ **Step 2.1:** Apply the addition operator $U_{a(234)}$

▷ **Step 2.2:** Apply the rounding operator on qubit three $U_{r(3)}$

▷ **Step 2.3:** Apply the subtraction operator $U_{s(13)}$

These operations produce the state,

$|D_i^j, s_{2i+2}, a_0^r, s_{2i}\rangle$ ▷ a_i^r are the addition results after rounding

Step 3: Select the next block of elements

$|s_{2i-1}, s_{2i-2}, \phi_1, s_{2i}\rangle$

▷ **Step 3.1:** Apply addition operator $U_{a(234)}$

▷ **Step 3.2:** Apply rounding operator on qubit three $U_{r(3)}$

▷ **Step 3.3:** Apply the subtraction operator $U_{s(13)}$

This gives,

$|D_{i-1}^j, s_{2i-2}, a_1^r, s_{2i}\rangle$

Step 4: Select the last block of qubits

$|D_{i-1}^j, \phi_2, D_i^j, \phi_3, s_{2i}\rangle$

▷ **Step 4.1:** Apply the addition operator $U_{a(123)}$

▷ **Step 4.2:** Apply the rounding operator two-times in qubit four $U_{r(2)}$

▷ **Quantum CDF Transform - Last Part**

▷ **Step 4.3:** Apply the subtraction operator $U_{a(245)}$

This produces the final state,

$|D_{i-1}^{j}, a_2^{2R}, D_i^j, A_i^j, s_{2i}\rangle$

Step 5: Extract the decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$

▷ Repeat on $|A_i^j\rangle$ to get l-decomposition levels.

end procedure

Algorithm 3 : Quantum DB4 Transform - Decomposition Process

▷ $\boldsymbol{Input}$: One-dimensional signal $|S\rangle$ of 2^n.

▷ $\boldsymbol{Output}$: Decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$. ▷ With $j \leq l$

procedure

Step 1: State initialization

$|0 \ldots 0\rangle = |s_{2i-1}, s_{2i-2}, s_{2i+1}, s_{2i}\rangle$

$|0 \ldots 0\rangle = |\phi_6, \phi_5, \phi_4, \phi_3, \phi_2, \phi_1, \phi_0\rangle$ ▷ Auxiliary qubits

Step 2: Select the first block of elements

$|s_{2i+1}, \phi_0, s_{2i}\rangle$

▷ **Step 2.1:** Apply the operator $U_{\sqrt{3}(23)}$

▷ **Step 2.2:** Apply the rounding operator on qubit two $U_{r(2)}$

▷ **Step 2.3:** Apply the subtraction operator $U_{s(12)}$

These operations produce the state,

$|D_i', a_0^r, s_{2i}\rangle$ ▷ a_i^r are the addition results after rounding

Step 3: Select the second block of elements

$|s_{2i-1}, \phi_1, s_{2i-2}\rangle$

▷ **Step 3.1:** Apply the operator $U_{\sqrt{3}(23)}$

▷ **Quantum DB4 Transform - Last Part**

▷ **Step 3.2:** Apply the rounding operator on qubit two $U_{r(2)}$

▷ **Step 3.3:** Apply the subtraction operator $U_{s(12)}$

This gives,

$|D'_{i-1}, a_1^r, s_{2i-2}\rangle$

Step 4: Select the third block of qubits

$|D'_{i-1}, \phi_4, \phi_3, \phi_2, D'_i\rangle$

▷ **Step 4.1:** Apply the operator $U_{\sqrt{3}-2(12)}$

▷ **Step 4.2:** Apply the operator $U_{\sqrt{3}(45)}$

▷ **Step 4.3:** Apply the addition operator $U_{a(234)}$

▷ **Step 4.4:** Apply the rounding operator two times on qubit three $U^2_{r(3)}$

This produces the state,

$|D'_{i-1}, \phi'_4 a_2^{2r}, \phi'_2, D'_i\rangle$

Step 5: Select the next block of qubits

$|a_2^{2r}, \phi_5, s_{2i}\rangle$

▷ **Step 5.1:** Apply the addition operator $U_{a(123)}$

This produces,

$|a_2^{2r}, A_i^j, s_{2i}\rangle$

Step 6: Select the last block of qubits

$|D'_i, \phi_6, A_{i+1}^j\rangle$

▷ **Step 6.1:** Apply the addition operator $U_{a(123)}$

This produces,

$|D'_i, D_i^j, A_{i+1}^j\rangle$

Step 7: Extract the decomposition coefficients $|A_i^j\rangle$ and $|D_i^j\rangle$

▷ Repeat on $|A_i^j\rangle$ to get l-decomposition levels.

end procedure

4.5 Quantum Compression

Quantum compression involves reducing the number of qubits, quantum gates, and measurements on a quantum process. Thus, based on the features of the proposed representation (QBRBS), we achieve quantum compression in two different ways:

1. The QBRBS uses superposition and entanglement to reduce the number of elements needed to store information in the quantum domain, providing an exponential improvement over classical methods (Sec. 4.2). Consequently, the proposed format requires $(q+n)$ qubits for storing a 2^n signal, while classical representation employs $(q \times 2^n)$ bits, with q-bit precision.

2. The proposed representation format eliminates the requirement for multiple copies of the initial signal to reach l-decomposition levels. This is achieved by splitting and storing the input signal as dictated by the QBRBS format in (4.30). This guarantees an iterative decomposition process without the need for repeated measurements at the end of each level.

On the other hand, from a classical perspective of compression schemes, we achieve lossless compression by combining classical and quantum features to obtain a hybrid approach. Thus, the proposed method uses fixed-length coding to decrease the number of qubits required for signal storage or transmission. The process is outlined below.

1. **Classical procedure**

   - Get an input signal, $S = \{s_i\}_{i=0,1,2,\ldots,N-1}$.
   - Perform the classical Integer Wavelet Transform (IWT) to obtain the decomposition coefficients, $A^l = \{A^l_i\}_{i=0,1,2,\ldots}$ and $D^l = \{D^l_i\}_{i=0,1,2,\ldots}$, up to l-levels.

- Identify the different values of the coefficient elements, that is, obtain the subsets:

$$V = \{A_i^l \mid A_i^l \neq A_j^l\}_{i=0,1,2,\ldots}$$
$$W = \{D_i^l \mid D_i^l \neq D_j^l\}_{i=0,1,2,\ldots}$$

- Generate two new sets with the codewords for each element in V and W.

$$T_{cv} = \{C(V_i) \mid V_i \in V,\ C(V_i) = i\}_{i=0,1,2,\ldots}$$
$$T_{cw} = \{C(W_i) \mid W_i \in W,\ C(W_i) = i\}_{i=0,1,2,\ldots}$$

  where $C(\cdot)$ defines the fixed-length codeword for each element of V and W.

2. **Quantum procedure**

- Store the input signal into the quantum domain by the QBRBS format (4.15).
- Apply some of the proposed QIWT to achieve a lossless signal transformation, giving the $|A^l\rangle$ and $|D^l\rangle$ coefficients.
- Define two quantum operators based on T_{cx} and T_{cy}.

$$U_{qcv} = \sum_{i=0}^{|V_i|-1} |V_i\rangle\langle V_i| \otimes |C(V_i)\rangle\langle 0\ldots 0|$$
$$U_{qcw} = \sum_{i=0}^{|W_i|-1} |W_i\rangle\langle W_i| \otimes |C(W_i)\rangle\langle 0\ldots 0|$$

  where these operators acts over superposed states, and the register $|0\ldots 0\rangle$ stores the codewords.
- Perform the U_{qcv} and U_{qcw} operators on $|A^l\rangle$ and $|D^l\rangle$, respectively. Thus, all the elements $|A_i^l\rangle = |A_j^l\rangle$ and $|D_i^l\rangle = |D_j^l\rangle$ are modified by an unique application of the quantum operators.
- Obtain the compressed coefficients $|A^l\rangle_c$ and $|D^l\rangle_c$. Figure 4.12 illustrates the general hybrid compression scheme.

The above description shows how to compress an input signal using a hybrid approach, where we take advantage of quantum parallelism to assign a codeword to all

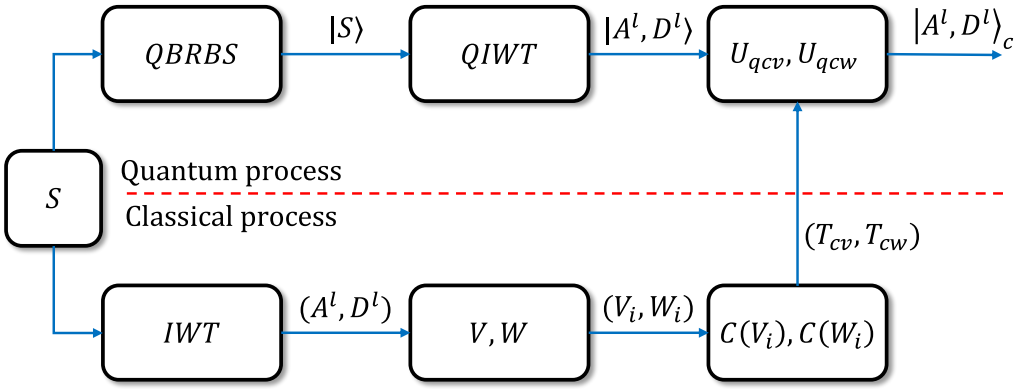

Figure 4.12: General hybrid compression scheme.

identical elements simultaneously. For example, giving a input signal, S, of size $N = 8$ with decomposition level $l = 1$, we have

1. **Classical procedure**

   - Input signal

   $$S = \{1, 3, 8, 1, 2, 3, 1, 3\}$$

   - Apply the integer transform, IWT, and get the decomposition coefficients

   $$A^1 = \{0, 2, 4, 0\}$$

   $$D^1 = \{1, 2, 0, 1\}$$

   - Get the subsets V and W

   $$V = \{0, 2, 4\}$$

   $$W = \{1, 2, 0\}$$

   - Generate T_{cv} and T_{cw}

   $$\begin{aligned} T_{cv} &= \{C(V_0), C(V_1), C(V_2)\} \\ &= \{C(0), C(2), C(4)\} = \{0, 1, 2\} \\ T_{cw} &= \{C(W_0), C(W_1), C(W_2)\} \end{aligned}$$

$$= \{C(1), C(2), C(0)\} = \{0, 1, 2\}$$

2. **Quantum procedure**

- Store S using the QBRBS format

$$|S\rangle = |s_{2i}\rangle|s_{2i+1}\rangle|X_i\rangle = \frac{1}{2}\Big[|1\rangle|3\rangle|X_0\rangle + |8\rangle|1\rangle|X_1\rangle + |2\rangle|3\rangle|X_2\rangle + |1\rangle|7\rangle|X_3\rangle\Big]$$

  where the elements are stored in odd/even pairs.

- Apply the one-level QWT on $|S\rangle$ and get the coefficients

$$\begin{aligned}
|A^1\rangle &= |0\rangle|X_0\rangle + |2\rangle|X_1\rangle + |4\rangle|X_2\rangle + |0\rangle|X_3\rangle \\
&= |A_i^1\rangle|X_i\rangle \\
|D^1\rangle &= |1\rangle|X_0\rangle + |2\rangle|X_1\rangle + |0\rangle|X_2\rangle + |1\rangle|X_3\rangle \\
&= |D_i^1\rangle|X_i\rangle
\end{aligned}$$

- Initialize an auxiliary register $|0 \ldots 0\rangle$ and distribute the tensor product with the previous coefficients

$$\begin{aligned}
|A^1\rangle|0 \ldots 0\rangle &= |0\rangle|0 \ldots 0\rangle|X_0\rangle + |2\rangle|0 \ldots 0\rangle|X_1\rangle \\
&\quad + |4\rangle|0 \ldots 0\rangle|X_2\rangle + |0\rangle|0 \ldots 0\rangle|X_3\rangle \\
|D^1\rangle|0 \ldots 0\rangle &= |1\rangle|0 \ldots 0\rangle|X_0\rangle + |2\rangle|0 \ldots 0\rangle|X_1\rangle \\
&\quad + |0\rangle|0 \ldots 0\rangle|X_2\rangle + |1\rangle|0 \ldots 0\rangle|X_3\rangle
\end{aligned}$$

- Define the quantum codeword operators

$$\begin{aligned}
U_{qcv} &= |V_0\rangle\langle V_0| \otimes |C(V_0)\rangle\langle 0 \ldots 0| + |V_1\rangle\langle V_1| \otimes |C(V_1)\rangle\langle 0 \ldots 0| \\
&\quad + |V_2\rangle\langle V_2| \otimes |C(V_2)\rangle\langle 0 \ldots 0| \\
&= |0\rangle\langle 0| \otimes |0\rangle\langle 0 \ldots 0| + |2\rangle\langle 2| \otimes |1\rangle\langle 0 \ldots 0| \\
&\quad + |4\rangle\langle 4| \otimes |2\rangle\langle 0 \ldots 0| \\
U_{qcw} &= |W_0\rangle\langle W_0| \otimes |C(W_0)\rangle\langle 0 \ldots 0| + |W_1\rangle\langle W_1| \otimes |C(W_1)\rangle\langle 0 \ldots 0|
\end{aligned}$$

$$
\begin{aligned}
&+ |W_2\rangle\langle W_2| \otimes |C(W_2)\rangle\langle 0\ldots 0| \\
&= |1\rangle\langle 1| \otimes |0\rangle\langle 0\ldots 0| + |2\rangle\langle 2| \otimes |1\rangle\langle 0\ldots 0| \\
&+ |0\rangle\langle 0| \otimes |2\rangle\langle 0\ldots 0|
\end{aligned}
$$

- Perform the previous operators on $|A^l\rangle|0\ldots 0\rangle$ and $|D^l\rangle|0\ldots 0\rangle$

$$
\begin{aligned}
U_{qcv}|A^1\rangle|0\ldots 0\rangle &= U_{qcv}(|0\rangle|0\ldots 0\rangle)|X_0\rangle + U_{qcv}(|2\rangle|0\ldots 0\rangle)|X_1\rangle \\
&+ U_{qcv}(|4\rangle|0\ldots 0\rangle)|X_2\rangle + U_{qcv}(|0\rangle|0\ldots 0\rangle)|X_3\rangle \\
&= (|0\rangle\langle 0|0\rangle \otimes |0\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_0\rangle \\
&+ (|2\rangle\langle 2|2\rangle \otimes |1\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_1\rangle \\
&+ (|4\rangle\langle 4|4\rangle \otimes |2\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_2\rangle \\
&+ (|0\rangle\langle 0|0\rangle \otimes |0\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_3\rangle \\
&= |0\rangle|0\rangle|X_0\rangle + |2\rangle|1\rangle|X_1\rangle \\
&+ |4\rangle|2\rangle|X_2\rangle + |0\rangle|0\rangle|X_3\rangle \\
U_{qcw}|D^1\rangle|0\ldots 0\rangle &= U_{qcw}(|1\rangle|0\ldots 0\rangle)|X_0\rangle + U_{qcw}(|2\rangle|0\ldots 0\rangle)|X_1\rangle \\
&+ U_{qcw}(|0\rangle|0\ldots 0\rangle)|X_2\rangle + U_{qcw}(|1\rangle|0\ldots 0\rangle)|X_3\rangle \\
&= (|1\rangle\langle 1|1\rangle \otimes |0\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_0\rangle \\
&+ (|2\rangle\langle 2|2\rangle \otimes |1\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_1\rangle \\
&+ (|0\rangle\langle 0|0\rangle \otimes |2\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_2\rangle \\
&+ (|1\rangle\langle 1|1\rangle \otimes |0\rangle\langle 0\ldots 0|0\ldots 0\rangle)|X_3\rangle \\
&= |1\rangle|0\rangle|X_0\rangle + |2\rangle|1\rangle|X_1\rangle \\
&+ |0\rangle|2\rangle|X_2\rangle + |1\rangle|0\rangle|X_3\rangle
\end{aligned}
$$

  where U_{qcv} and U_{qcw} act on element values and not on the position register $|X_i\rangle$. The red values are the new signal representation by the codewords and the blue ones are the original values.

- Get the compressed coefficients $|A^l\rangle_c$ and $|D^l\rangle_c$, which are stored in the aux-

iliary register

$$|A^1\rangle_c = |0\rangle|X_0\rangle + |1\rangle|X_1\rangle + |2\rangle|X_2\rangle + |0\rangle|X_3\rangle$$

$$|D^1\rangle_c = |0\rangle|X_0\rangle + |1\rangle|X_1\rangle + |2\rangle|X_2\rangle + |0\rangle|X_3\rangle$$

where the original $|A^1\rangle$ requires three qubits and $|A^1\rangle_c$ uses two qubits to store the element values. However, $|D^1\rangle$ and $|D^1\rangle_c$ uses the same number of qubits.

The above example shows the steps for the proposed hybrid compression scheme, where we decrease the number of qubits to represent the coefficient values of the decomposed signal. Figure 4.13 depicts the implementation circuit for the quantum codeword process, assuming the sets of classical codewords T_{cv} and T_{cw} are given.

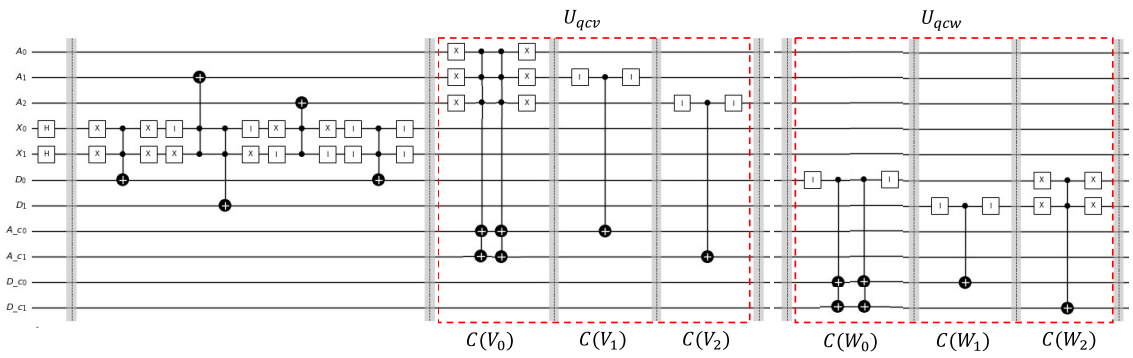

Figure 4.13: Quantum codeword circuit. Dashed boxes define the quantum codeword operators, where each sub-block describes the code assignment process.

4.6 Summary

In this chapter, we have defined the quantum circuits for the addition, subtraction, swapping, halving, rounding, cyclic shift, multiplication, and special operations on the quantum versions of the Haar, CDF, and Daubechies-4 (DB4) wavelet transform based on the

lifting scheme. We have generalized the traditional Quantum Basis States (QBS) representation to the Quantum Block Representation by Basis States (QBRBS) to facilitate signal manipulation and avoid garbage information. Then, we have presented the unitary and algorithmic definitions of the Haar, CDF, and DB4 wavelet transforms. Finally, we detailed the quantum compression model using a hybrid Fixed-Length Coding (FLC) scheme.

5

Experimental and Analysis Results

This chapter presents the analysis and experimental results of the quantum wavelet decomposition process, including simulations in the IBM Qiskit quantum toolkit and complexity evaluations. The comparative analysis of the quantum and classical versions of the wavelet transform is also described, as well as the comparative analysis of the proposed quantum representation with the QBS representation format. Finally, quantum compression results and evaluation metrics are provided.

We use a Intel(R) Core(TM) i5-9300H CPU 2.40GHz, 16RAM with GTX 1650. Python 3.9 with IBM Qiskit toolkit 1.1. The input signal for the experiments are random one-dimensional integer signals.

5.1 Quantum Wavelet Simulations

The proposed one-dimensional quantum wavelet transforms operate in integer space, that is, the transformation is integer-to-integer mapping. Thus, given a one-dimensional

integer signal of $N = 2^6$ for the Haar and DB4 transforms, and $N = 2^6 + 1$ for the CDF(2,2) wavelet, we apply the quantum and classical transforms for three decomposition levels; resulting in one approximation and three detail coefficients (Figures 5.1, 5.2 and 5.3).

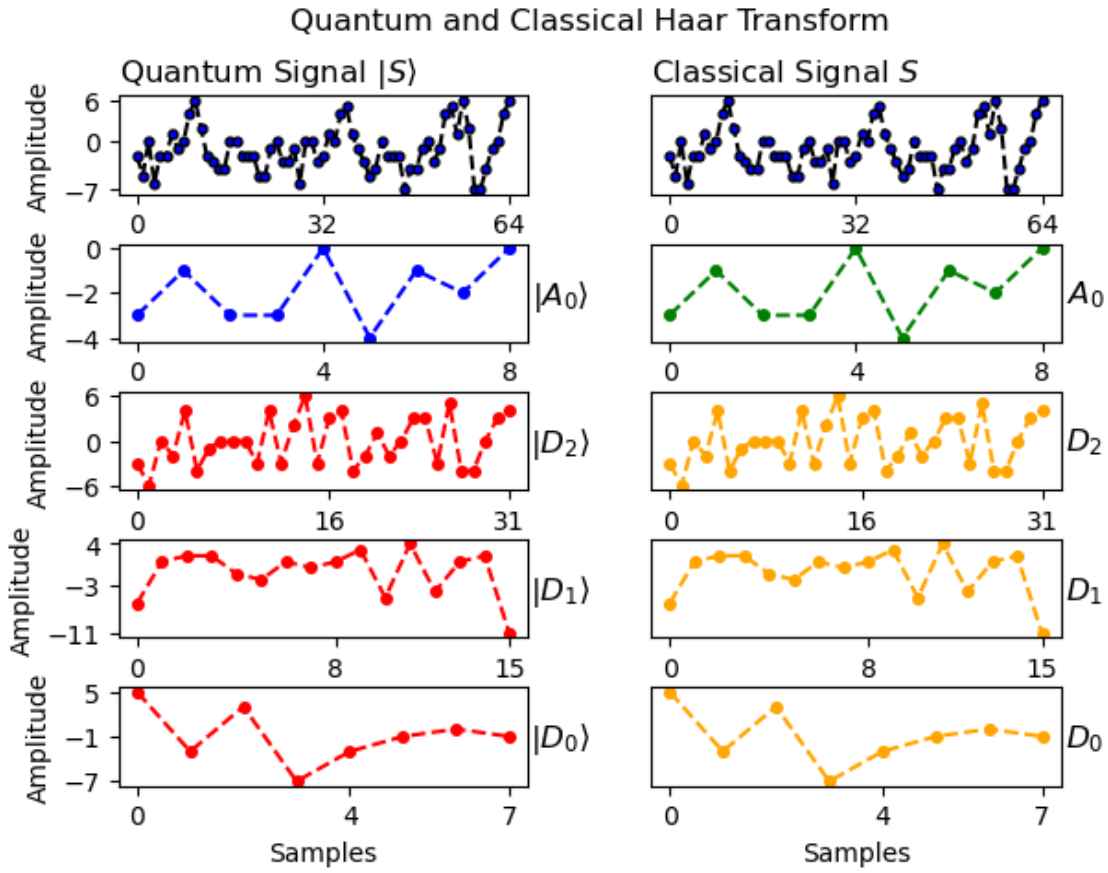

Figure 5.1: Quantum and classical decomposition by the Haar transform on a one-dimensional integer signal. The dot points are the integer values of the signals.

Figures 5.1, 5.2 and 5.3 depict the decomposition results of the quantum and classical transforms, where the error between the quantum and classical coefficients for each wavelet pair is zero, verifying the applicability and correctness of the quantum transforms.

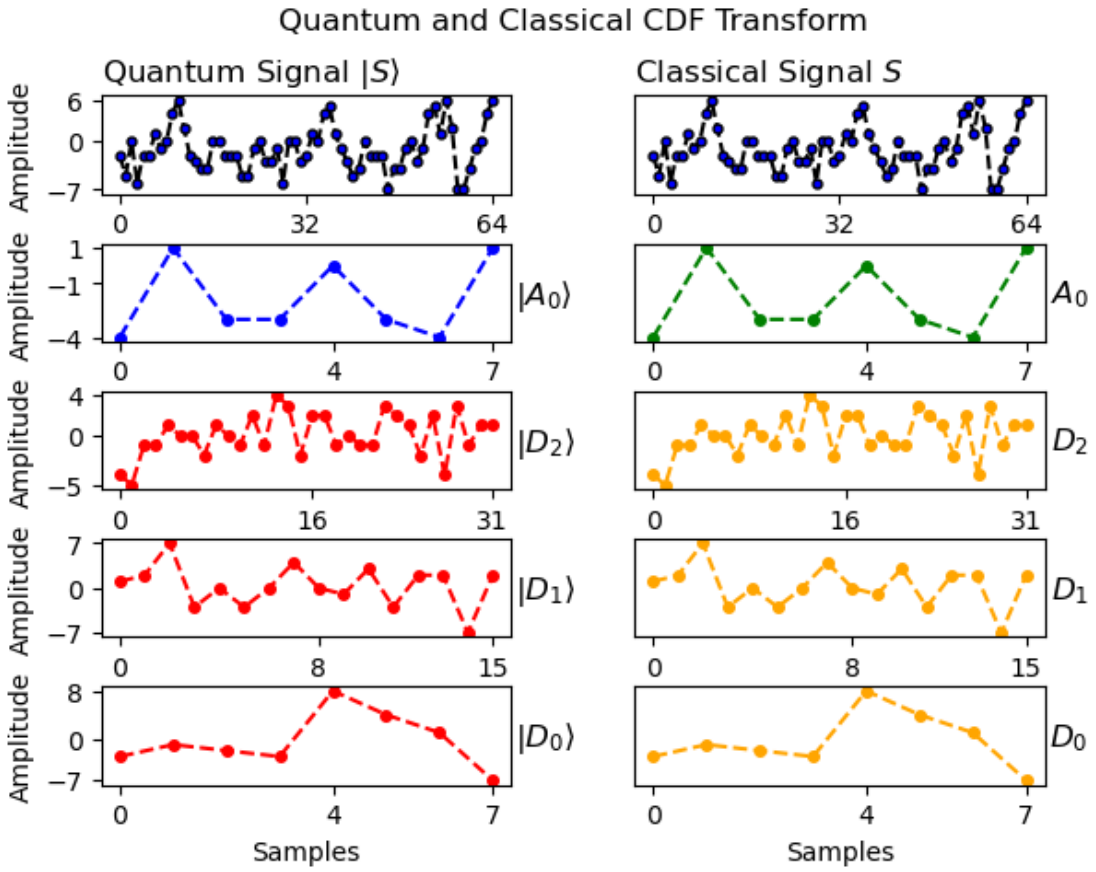

Figure 5.2: Quantum and classical decomposition by the CDF(2,2) transform on a one-dimensional integer signal. The dot points are the integer values of the signals.

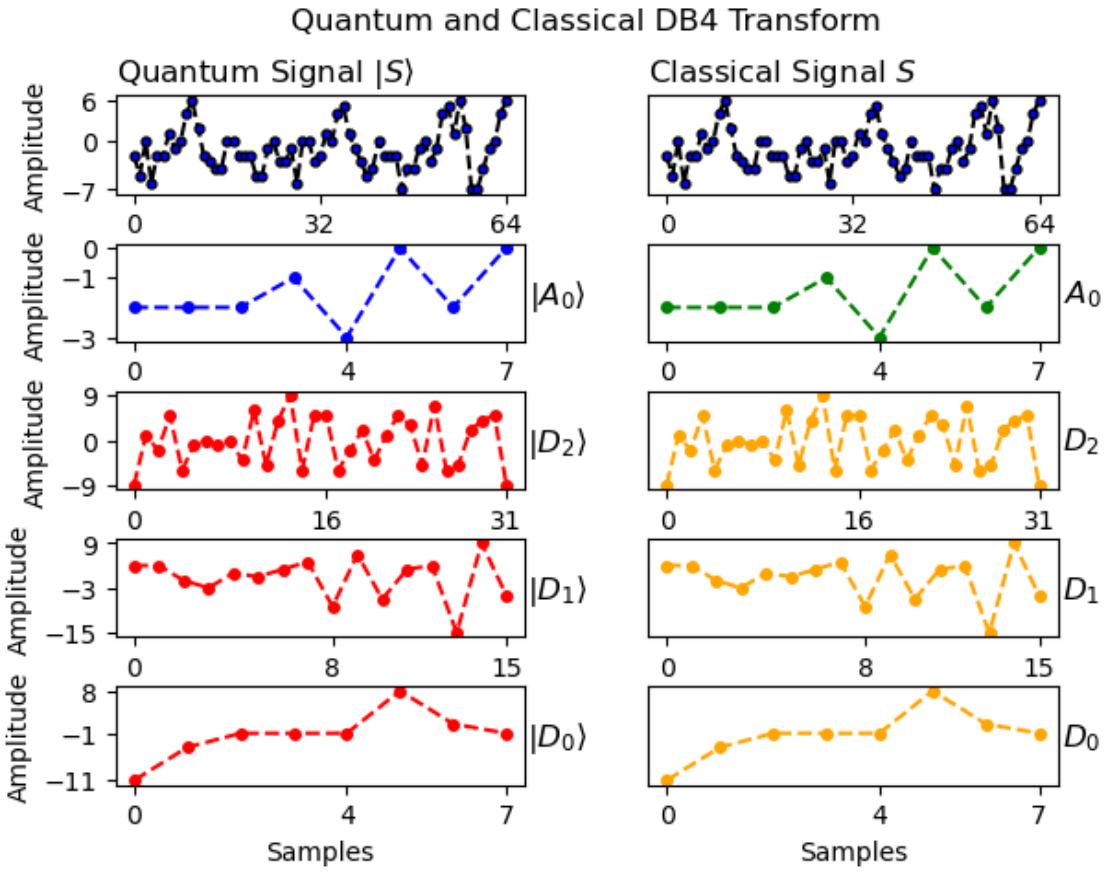

Figure 5.3: Quantum and classical decomposition by the DB4 transform on a one-dimensional integer signal. The dot points are the integer values of the signals.

5.2 Complexity Analysis

Based on the quantum operators in the Haar, CDF(2,2), and DB4 wavelet transforms, we perform a quantum complexity analysis based on the number of quantum gates (QG), auxiliary (A) and garbage (G) qubits for each defined operation. Table 5.1 shows the quantum complexity for the quantum operations, given a 2^n signal with q-bit precision. Thus, the complexities of the Haar, CDF and DB4 transform for the maximum decomposition level ($l_{max} = n$) are $O(qn)$, $O(qn)$ and $O(q^2n)$, respectively.

Table 5.1: Complexity analysis of the quantum operations.

Operation	**QG**	**A**	**G**
Add	$7q$	$q+1$	0
Sub	$6q-2$	q	$q-1$
Rounding	$q+1$	1	0
Product	$9q^2-7q+1$	$2q+1$	1

5.3 Comparative Analysis: Quantum Wavelet Transforms

We present a comparative analysis of the quantum and classical Haar, CDF(2,2), and DB4 wavelet transforms. Table 5.2 shows the transform type, decomposition scheme and domain, wavelet class, and complexity, where quantum integer versions decrease the computational complexity.

Table 5.2: Comparative analysis of quantum and classical Haar, CDF(2,2) and DB4 transforms, where 2^n is the signal length and q the bit precision.

	Quantum Ours			**Quantum Literature**	**Classical Literature**		
Wavelet	**Haar (Ours)**[18]	**CDF(2,2) (Ours)**[66]	**DB4 (Ours)**	**Haar, DB4** [10, 14, 15] [44–48]	**Haar** [67]	**CDF(2,2)** [17]	**DB4** [17]
Type	Integer	Integer	Integer	Real-valued	Integer	Integer	Integer
Scheme	Lifting	Lifting	Lifting	Filter	Lifting	Lifting	Lifting
Domain	Spatial	Spatial	Spatial	Spatial	Spatial	Spatial	Spatial
Class	Orthogonal	Biorthogonal	Orthogonal	Orthogonal	Orthogonal	Biorthogonal	Orthogonal
Complexity	$\boldsymbol{O(qn)}$	$\boldsymbol{O(qn)}$	$\boldsymbol{O(q^2n)}$	$O(n^3)$	$O(n2^n)$	$O(n2^n)$	$O(n2^n)$

Table 5.2 depicts the quantum and classical wavelet transforms, where the integer type uses the lifting scheme for decomposition and the real-valued type is based on the traditional filter bank process. The decomposition analysis is performed in the spatial signal domain. Also, the proposed transforms belong to the orthogonal and bi-orthogonal class, with the CDF(2,2) being the first bi-orthogonal quantum wavelet transform. Finally, the developed quantum wavelets significantly decrease the computational complexity compared to the classical versions, providing an exponential time speed-up.

5.4 Comparative Analysis: Quantum Representations

We perform a comparative analysis between the traditional QBS representation and the proposed QBRBS format. We analyze the features of the quantum formats, including the number of qubits, quantum storage, gate complexity, the generation of garbage information, and compatibility with iterative processes.

5.4.1 Quantum Qubit Complexity

Quantum qubit complexity measures the numbers of qubits to store an input signal in the quantum domain. Table 5.3 presents the qubit complexity to store a one- or two-dimensional signal with the QBS and the QBRBS format, where q is the bit precision and $N = 2^n$ the signal length. In addition, the number of elements per block for the QBRBS is $m = 1$ and the number of blocks $k = 2^n/m$.

Table 5.3 shows the qubit complexity for the QBS and QBRBS representation format. Both models have the same qubit complexity for storing one- and two-dimensional signals since the QBRBS model generalizes the traditional QBS format and behaves similarly for simple signals when no element blocks are provided.

Table 5.3: Qubit Complexity for a 2^n signal with $m = 1$ and $k = 2^n/m$.

Format	Qubit Complexity
QBS	$n + q$
QBRBS	$mq + \lceil \log k \rceil$

5.4.2 Quantum Storage, Gates, and Garbage Information

The proposed Quantum Integer Wavelet transforms (QIWTs) require the information of neighboring elements from the input signal to obtain the decomposition values. Therefore, we describe the process for storing and operating on adjacent elements using the QBS and the QBRBS format. Additionally, we highlight the issues of dealing with superposed states in generating garbage information. Finally, we show the shortcomings of iterative processes in both representation formats.

The QBS format allows for the storage of individual input signals without sharing information with other elements, as they are in superposition. Therefore, when operating with neighboring values, it is necessary to initialize the input signal multiple times, equal to the number of required neighbors. For instance, to access the right-neighboring elements in a one-dimensional signal of size N, we need to initialize the input signal twice, $|S\rangle_0$ and $|S\rangle_1$, and then apply a cyclic shift operator, U_{cs}, to the position values $|x_j\rangle$ of $|S\rangle_1$, moving the elements to the same position as their neighbors as follows:

- Initialize twice the same input signal

$$
\begin{aligned}
|S\rangle_0 &= |x_i\rangle|s_i\rangle \\
&= |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle \\
|S\rangle_1 &= |x_j\rangle|s_j\rangle \\
&= |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle
\end{aligned}
$$

- Apply the cyclic operator, $U_{cs(j-1)}$, on the position coordinate register of $|S\rangle_1$

$$\begin{aligned} U_{cs(j-1)}|S\rangle_1 &= (U_{cs}|x_j\rangle)|s_j\rangle \\ &= |x_{(j-1) \mod N}\rangle|s_j\rangle_{j=0}^{N-1} \\ &= |x_0\rangle|s_1\rangle + |x_1\rangle|s_2\rangle + |x_2\rangle|s_3\rangle + |x_3\rangle|s_0\rangle \end{aligned}$$

  where $U_{cs(j-1)}$ moves the position coordinate $|x_j\rangle$ to $|x_{(j-1) \mod N-1}\rangle$.
- Get the input signal, $|S\rangle_0$, and the signal, $|S\rangle_1$, with the right-neighboring elements of $|S\rangle_0$

$$\begin{aligned} |S\rangle_0 &= |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle \\ |S\rangle_1 &= |x_0\rangle|s_1\rangle + |x_1\rangle|s_2\rangle + |x_2\rangle|s_3\rangle + |x_3\rangle|s_0\rangle \end{aligned} \tag{5.1}$$

  where elements with the same position coordinate share the neighboring information of the signal.

Now, if we want to apply an operation on neighboring values, we must follow these steps:

- Add two auxiliary registers to the tensor product of $|S\rangle_1|S\rangle_0$

$$|d\rangle|c\rangle|S\rangle_1|S\rangle_0 = |0\ldots0\rangle|0\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle$$

  where $k = (j-1) \mod (N-1)$.
- Perform a comparator operation, U_c, on register $|c\rangle|S\rangle_1|S\rangle_0$ to identify when the position coordinates of both signals are equal, and then change the auxiliary qubit to $|c\rangle = |1\rangle$

$$\begin{aligned} |d\rangle(U_c|c\rangle|S\rangle_1|S\rangle_0) &= (|0\ldots0\rangle|1\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k=i} \\ &\quad + (|0\ldots0\rangle|0\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k\neq i} \\ &= |0\ldots0\rangle|1\rangle|x_0\rangle|s_1\rangle|x_0\rangle|s_0\rangle \end{aligned}$$

$$
\begin{aligned}
&+ |0\ldots0\rangle|1\rangle|x_1\rangle|s_2\rangle|x_1\rangle|s_1\rangle \\
&+ |0\ldots0\rangle|1\rangle|x_2\rangle|s_3\rangle|x_2\rangle|s_2\rangle \\
&+ |0\ldots0\rangle|1\rangle|x_3\rangle|s_0\rangle|x_3\rangle|s_3\rangle \\
&+ (|0\ldots0\rangle|0\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k\neq i}
\end{aligned}
$$

where only four registers share the same position coordinates with the neighboring information.

- Apply a conditional operator, CU, on $|s_j\rangle|s_i\rangle$ when $|c\rangle = |1\rangle$, and store the result on $|d\rangle$

$$
\begin{aligned}
CU|d\rangle|c\rangle|S\rangle_1|S\rangle_0 &= CU(|0\ldots0\rangle|1\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k=i} \\
&+ (|0\ldots0\rangle|0\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k\neq i}
\end{aligned}
$$

$$
\begin{aligned}
&= |r_0\rangle|1\rangle|x_0\rangle|s_1\rangle|x_0\rangle|s_0\rangle \\
&+ |r_1\rangle|1\rangle|x_1\rangle|s_2\rangle|x_1\rangle|s_1\rangle \\
&+ |r_2\rangle|1\rangle|x_2\rangle|s_3\rangle|x_2\rangle|s_2\rangle \\
&+ |r_3\rangle|1\rangle|x_3\rangle|s_0\rangle|x_3\rangle|s_3\rangle \\
&+ (|0\ldots0\rangle|0\rangle|x_k\rangle|s_j\rangle|x_i\rangle|s_i\rangle)_{k\neq i}
\end{aligned}
$$

where $|r_i\rangle$ is the result after the CU operation.

Finally, we extract the results after the conditional operation, CU, as follows:

- Apply an measurement operator, M, on register $|d\rangle$

$$M|d\rangle \in \{|r_0\rangle, |r_1\rangle, |r_2\rangle, |r_3\rangle, |0\ldots0\rangle\}$$

where each $|r_i\rangle$ has a probability of $P(r_i) = 1/16$, and the register $|0\ldots0\rangle$ a probability of $P(0\ldots0) = 12/16$. However, only measure the register $|d\rangle$ does

not give information about the position coordinate of the results. Therefore, we also need to measure registers $|x_k\rangle|x_i\rangle$:

$$M|d\rangle|x_k\rangle|x_i\rangle \in \{|r_0\rangle|x_0\rangle|x_0\rangle, |r_1\rangle|x_1\rangle|x_1\rangle, |r_2\rangle|x_2\rangle|x_2\rangle, \\ |r_3\rangle|x_3\rangle|x_3\rangle, |0\dots0\rangle|x_k\rangle|x_i\rangle_{k\neq i}\}$$

where the number of measurable registers is increased, but the probabilities remain the same as before.

The previous description shows how to access and operate on neighboring values of an input signal $|S\rangle_0$, where elements at the same position coordinate hold the neighboring information of the signal. However, we need as many copies of the input signal as neighbors are required in the process, increasing storage requirements. Moreover, the measurement complexity increases because of additional (garbage) states, $|0\dots0\rangle$, that do not provide any useful information about the process. These garbage data are generated by the tensor product of superposed states, which describes a Cartesian product of all possible states. Additionally, neighboring processing with QBS requires cyclic shifts, comparators, and conditional operations to obtain the results, increasing quantum gate complexity. Figure 5.4 illustrates the general process using QBS representation, where $|S_k\rangle_{k\neq0} = |x_j s_j\rangle$ are the copied signals, $U_{cs(j\pm h)}$ the cyclic operator, U_c the comparator, and CU the conditional operator.

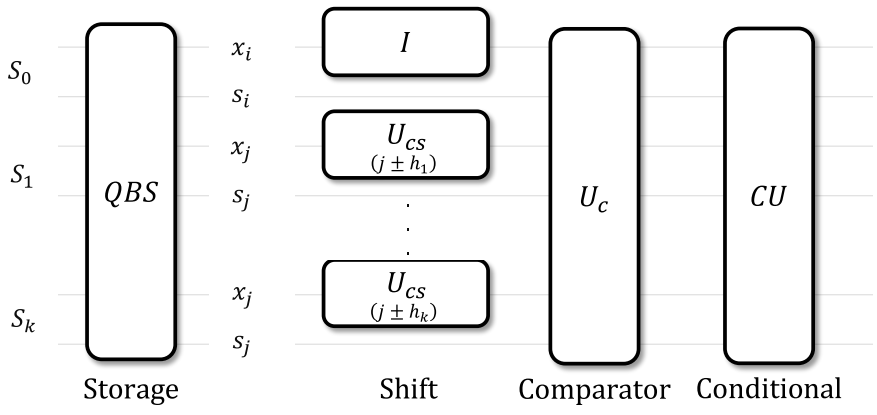

Figure 5.4: General storage and manipulation process by QBS representation.

On the other hand, the QBRBS format enables the storage of groups of signal values to share information between different elements in the same position coordinate. This eliminates the need to initialize the input signal multiple times to access neighboring values. For example, to process the right-neighboring elements of a one-dimensional signal, we must store each neighboring value at the same position coordinate $|x_i\rangle$, facilitating manipulation as follows:

- Get the input signal

$$\begin{aligned}|S\rangle_0 &= |x_i\rangle|s_i\rangle \\ &= |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle\end{aligned}$$

- Store in the QBRBS with the neighboring information

$$\begin{aligned}|S\rangle &= |x_i\rangle|s_i\rangle|s_{(j+1) \mod N}\rangle_{j=i=0}^{N-1} \\ &= |x_0\rangle|s_0\rangle|s_1\rangle + |x_1\rangle|s_1\rangle|s_2\rangle + |x_2\rangle|s_2\rangle|s_3\rangle + |x_3\rangle|s_3\rangle|e_0\rangle\end{aligned}$$

  where $|e_0\rangle = 0$ is a zero padding element, which solves the issue of non-adjacent values for the boundary element $|s_3\rangle$.

- Add an auxiliary register

$$\begin{aligned}|d\rangle|S\rangle = |0\ldots0\rangle|x_0\rangle|s_0\rangle|s_1\rangle + |0\ldots0\rangle|x_1\rangle|s_1\rangle|s_2\rangle \\ + |0\ldots0\rangle|x_2\rangle|s_2\rangle|s_3\rangle + |0\ldots0\rangle|x_3\rangle|s_3\rangle|e_0\rangle\end{aligned}$$

- Apply a non-conditional operator, U, on adjacent values and store in register $|d\rangle$

$$\begin{aligned}U|d\rangle|S\rangle &= U(|0\ldots0\rangle|x_0\rangle|s_0\rangle|s_1\rangle) + U(|0\ldots0\rangle|x_1\rangle|s_1\rangle|s_2\rangle) \\ &\quad + U(|0\ldots0\rangle|x_2\rangle|s_2\rangle|s_3\rangle) + U(|0\ldots0\rangle|x_3\rangle|s_3\rangle|e_0\rangle) \\ &= |r_0\rangle|x_0\rangle|s_0\rangle|s_1\rangle + |r_1\rangle|x_1\rangle|s_1\rangle|s_2\rangle \\ &\quad + |r_2\rangle|x_2\rangle|s_2\rangle|s_3\rangle + |r_3\rangle|x_3\rangle|s_3\rangle|e_0\rangle\end{aligned}$$

  where $|r_i\rangle$ store the result after the U operation.

- Perform a measurement operator, M, to get the results

$$M|d\rangle|x_i\rangle \in \{|r_0\rangle|x_0\rangle, |r_1\rangle|x_1\rangle, |r_2\rangle|x_2\rangle, |r_3\rangle|x_3\rangle\}$$

where the measurement process only gives the expected results with probability $P(r_i) = 1/4$, without additional (garbage) states.

The above process describes how to store and manipulate neighboring information using the QBRBS format, where adjacent values share the same position coordinate. This format decreases storage requirements because it is unnecessary to initialize multiple copies of the input signal, as in the QBS model. Also, QBRBS simplifies signal manipulation by avoiding comparators and conditional operators, which decreases the quantum gate complexity. Additionally, processing information in QBRBS representation does not generate additional (garbage) states after operations on adjacent elements, unlike the QBS format, which creates $N \times (N-1)$ garbage data. However, storing neighboring information in the QBRBS model requires an additional classical pre-processing step that defines the elements to be stored, i.e., it is necessary to know the neighbors of each signal component before quantum storage. Figure 5.5 shows the general process using QBRBS format, where the "Classc." block defines the classical pre-processing to get the neighboring values, each $|s_{(j\pm h_j)}\rangle$ is the corresponding neighboring element in some direction $(j \pm h_j)$, and U any non-conditional quantum operator.

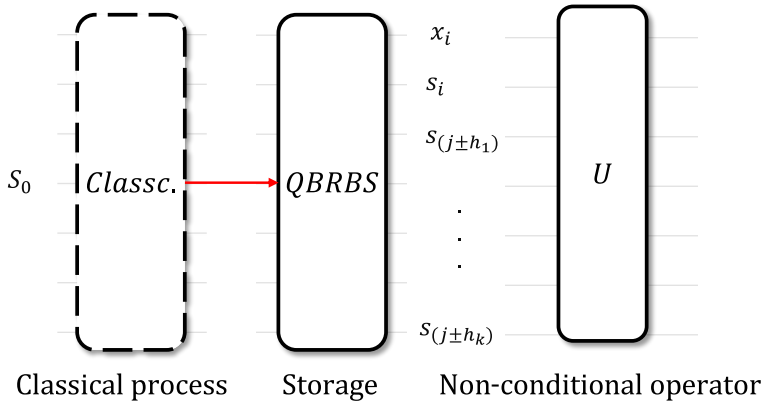

Figure 5.5: General storage and manipulation process by QBRBS representation.

5.4.3 Quantum Iterative Processes

Iterative processes define the repeatability of algorithmic steps and operations based on previous results, allowing outputs to be used as new inputs. This feature is relevant to the development of quantum transforms because each iteration describes a level of decomposition or transformation. Therefore, the proposed quantum integer wavelet transforms are expected to reach any decomposition level. The steps and issues of iterative processes for QBS representation are shown below.

1. Store an input signal in a superposed state using the QBS format, $|S\rangle_0$.
2. Initialize the input signal several times with the neighboring information, $|S\rangle_i$.
3. Apply the cyclic shift operation to get the neighbors, $|S'\rangle_i = U_{cs}|S\rangle_i$.
4. Perform a conditional operation between the input and neighboring signals, $CU|S'\rangle_i|S\rangle_0$.
5. Obtain the results into a superposition of states, $|R\rangle_0 = |r_i\rangle$.
6. Use the results as the new input signal, $|S\rangle_0 = |R\rangle_0$.
7. Initialize the new input signal multiple times, $|R\rangle_i$.
8. Repeat steps from 3 to 8.

The previous description outlines the general process for operating on neighboring elements in QBS representation. However, initializing the new input signal $|S\rangle_0 = |R\rangle_0$ multiples times (step 7) has a limitation: the register $|R\rangle_0$ is in a superposition of different states, where the values of the quantum state are unknown. Therefore, it is only possible to initialize the signal several times by a high-complex measurement process, increasing the quantum and storage complexity.

On the other hand, using the QBRBS format allows iterative processes by storing explicit neighboring information. The key idea is to perform a classical pre-processing

step to obtain the neighboring information for each iteration and store adjacent elements in the same position coordinate. For instance, to perform one and two iterations with a right-adjacent element in a one-dimensional signal, we have:

- Input registers for one iteration

$$|S\rangle_0 = |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle$$
$$|d_0\rangle = |0\ldots0\rangle$$

  where $|S\rangle_0$ the input signal, and $|d_0\rangle$ an auxiliary register.
- Store the right-adjacent elements in the QBRBS format

$$|S\rangle = |d_0\rangle\left(|x_0\rangle|s_0s_1\rangle + |x_1\rangle|s_1s_2\rangle + |x_2\rangle|s_2s_3\rangle + |x_3\rangle|s_3e_0\rangle\right)$$

  where $|e_0\rangle$ is the zero padding condition, $|e_0\rangle = |0\rangle$.
- Perform an operation, U, on neighboring elements $|s_is_{i+1}\rangle$ and store in $|d_0\rangle$

$$U|S\rangle = |r_0\rangle|x_0\rangle|s_0s_1\rangle + |r_1\rangle|x_1\rangle|s_1s_2\rangle + |r_2\rangle|x_2\rangle|s_2s_3\rangle + |r_3\rangle|x_3\rangle|s_3e_0\rangle$$

  where $|r_i\rangle$ is the result for one iteration level.

Now, to achieve two iteration levels, we need:

- Input register for two iterations

$$|S\rangle_0 = |x_0\rangle|s_0\rangle + |x_1\rangle|s_1\rangle + |x_2\rangle|s_2\rangle + |x_3\rangle|s_3\rangle$$
$$|d_0\rangle = |0\ldots0\rangle$$
$$|d_1\rangle = |0\ldots0\rangle$$
$$|d_2\rangle = |0\ldots0\rangle$$

- Store the right-neighboring elements in QBRBS for two iterations

$$|S\rangle = |d_2d_1d_0\rangle(|x_0\rangle|s_0s_1s_2\rangle + |x_1\rangle|s_1s_2s_3\rangle + |x_2\rangle|s_2s_3e_0\rangle + |x_3\rangle|s_3e_0e_1\rangle)$$

  where the boundary condition is $|e_i\rangle = |0\rangle$.

- Apply an U operator to the leftmost elements $|s_i s_{i+1}\rangle$ and store the result in $|d_0\rangle$

$$\begin{aligned}|S'\rangle = U|S\rangle &= |d_2 d_1\rangle U(|d_0\rangle|x_0\rangle|s_0 s_1\rangle)|s_2\rangle + |d_2 d_1\rangle U(|d_0\rangle|x_1\rangle|s_1 s_2\rangle)|s_3\rangle \\ &+ |d_2 d_1\rangle U(|d_0\rangle|x_2\rangle|s_2 s_3\rangle)|e_0\rangle + |d_2 d_1\rangle U(|d_0\rangle|x_3\rangle|s_3 e_0\rangle)|e_1\rangle \\ &= |d_2 d_1\rangle|r_0\rangle|x_0\rangle|s_0 s_1 s_2\rangle + |d_2 d_1\rangle|r_1\rangle|x_1\rangle|s_1 s_2 s_3\rangle \\ &+ |d_2 d_1\rangle|r_2\rangle|x_2\rangle|s_2 s_3 e_0\rangle + |d_2 d_1\rangle|r_3\rangle|x_3\rangle|s_3 e_0 e_1\rangle\end{aligned}$$

  where $|r_i\rangle$ is the result after U operation.

- Apply U on the rightmost elements $|s_j s_{j+1}\rangle$ and store in $|d_1\rangle$

$$\begin{aligned}|S''\rangle = U|S'\rangle &= |d_2\rangle|r_0\rangle|s_0\rangle U(|d_1\rangle|x_0\rangle|s_1 s_2\rangle) + |d_2\rangle|r_1\rangle|s_1\rangle U(|d_1\rangle|x_1\rangle|s_2 s_3\rangle) \\ &+ |d_2\rangle|r_2\rangle|s_2\rangle U(|d_1\rangle|x_2\rangle|s_3 e_0\rangle) + |d_2\rangle|r_3\rangle|s_3\rangle U(|d_1\rangle|x_3\rangle|e_0 e_1\rangle) \\ &= |d_2\rangle|r_0\rangle|s_0\rangle|r_1\rangle|x_0\rangle|s_1 s_2\rangle + |d_2\rangle|r_1\rangle|s_1\rangle|r_2\rangle|x_1\rangle|s_2 s_3\rangle \\ &+ |d_2\rangle|r_2\rangle|s_2\rangle|r_3\rangle|x_2\rangle|s_3 e_0\rangle + |d_2\rangle|r_3\rangle|s_3\rangle|e_2\rangle|x_3\rangle|e_0 e_1\rangle\end{aligned}$$

- Perform the U operation on the results $|r_i r_{i+1}\rangle$ and store in $|d_2\rangle$

$$\begin{aligned}U|S''\rangle &= U(|d_2\rangle|r_0\rangle|r_1\rangle)|x_0\rangle|s_0 s_1 s_2\rangle + U(|d_2\rangle|r_1\rangle|r_2\rangle)|x_1\rangle|s_1 s_2 s_3\rangle \\ &+ U(|d_2\rangle|r_2\rangle|r_3\rangle)|x_2\rangle|s_2 s_3 e_0\rangle + U(|d_2\rangle|r_3\rangle|e_2\rangle)|x_3\rangle|s_3 e_0 e_1\rangle \\ &= |r'_0\rangle|r_0\rangle|r_1\rangle|x_0\rangle|s_0 s_1 s_2\rangle + |r'_1\rangle|r_1\rangle|r_2\rangle|x_1\rangle|s_1 s_2 s_3\rangle \\ &+ |r'_2\rangle|r_2\rangle|r_3\rangle|x_2\rangle|s_2 s_3 e_0\rangle + |r'_3\rangle|r_3\rangle|e_2\rangle|x_3\rangle|s_3 e_0 e_1\rangle\end{aligned}$$

  where $|r'_i\rangle$ is the results for the second iteration level.

The previous examples describe the process for obtaining one and two iteration levels using the QBRBS representation. For two levels, not only the current adjacent value but also the neighbors for the first iteration are stored at the same position coordinate. Figure 5.6 and 5.7 depicts the storage and manipulation process with a right-neighbor value for one and two iteration levels, respectively. Adjacent elements are stored at the same position coordinate, $|x_i\rangle$, preserving the neighborhood relationships. Dashed rectangles holds the elements at the same position, and solid lines define the values selected for each step.

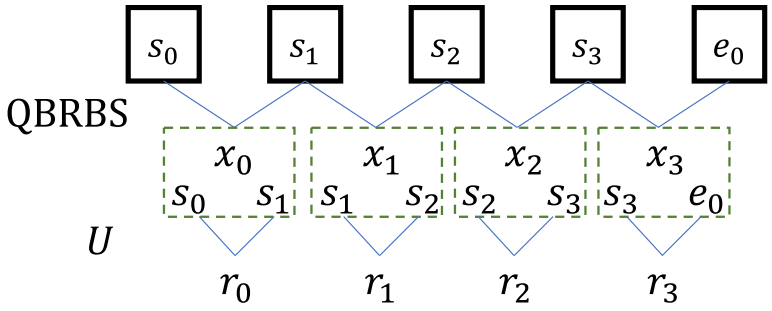

Figure 5.6: Quantum storage and manipulation for one iteration level. Elements $|s_i s_{i+1}\rangle$ share the same position coordinate $|x_i\rangle$, and the U operator acts on pairs of values in the same position.

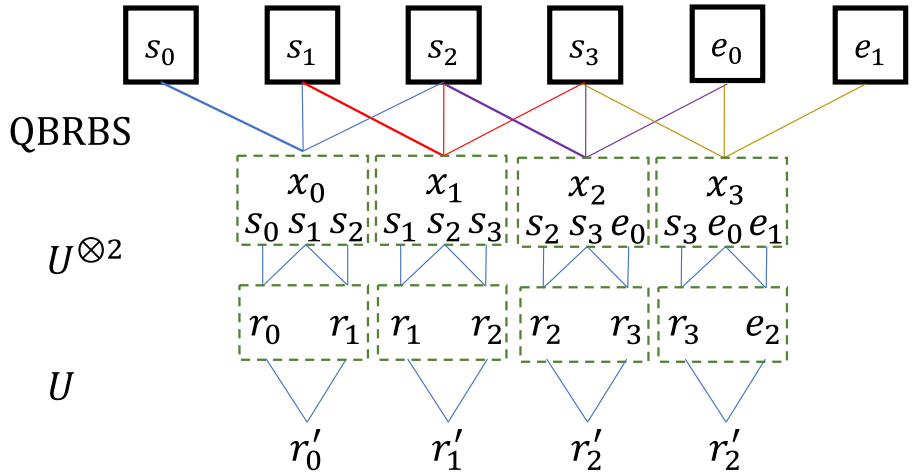

Figure 5.7: Quantum storage and manipulation for two iteration levels. Elements $|s_i s_{i+1} s_{i+2}\rangle$ share the same position coordinate $|x_i\rangle$, and the U operator acts on pairs of values in the same position.

Finally, Table 5.4 presents the comparative characteristics of the QBS and QBRBS formats, including qubit complexity (Q), type of operations (O), garbage information (G), iteration levels (I) and additional steps (A). The signal length is $N = 2^n$, and k is the number of neighboring values for each signal element. The operations in QBRBS do not use any conditional operators, decreasing quantum complexity. On the other hand, the QBS model generates $(N^2 - N)$ garbage information, increasing measurement complexity. However, QBRBS generates no garbage information, keeping measurement complexity low. Furthermore, the QBRBS format allows for multiple iteration steps, making it more applicable than the QBS format, which is limited to a single process. The proposed QBRBS format also requires a classical pre-processing step (Classc.) to access and store neighboring information. It is important to note that all neighborhood information must be known before quantum storage in the representation format. Thus, neighboring values can be stored at the same position coordinate, facilitating signal manipulation.

Table 5.4: Comparative analysis between the QBS and QBRBS formats, where $N = 2^n$.

Representation Format	**Comparative Elements**				
	Q	**O**	**G**	**I**	**A**
QBS	$kn + kq$	Condt.	$N^2 - N$	1	None
QBRBS	$n + kq$	NCondt.	0	n	Classc.

5.5 Quantum Lossless Compression: Analysis and Results

Compressing quantum information, like classical data compression methods, is a very complex task due to the constraints and limitations of quantum computing. Entangled and superposed states cannot be manipulated independently without increasing process complexity. Furthermore, the probabilistic nature of quantum states limits information

recovery due to measurement constraints. Therefore, common classical lossless methods, such as bit precision reduction and coding assignment, cannot be directly applied to quantum data. The issues in processing quantum information from the previous classical methods are detailed below.

- **Bit Precision Reduction**: this model takes advantage of the separability and independence of classical bits to change the bit precision of individual signal elements, it is, given an input signal, $S = \{s_i\}$, modify the bit precision, q, for each value $\{s_i\}$, such that

$$\begin{aligned} S &= \{s_0, s_1, s_2, \ldots, s_{N-1}\};\ N = 2^n \\ &= \{\{0,1\}^q, \{0,1\}^q, \{0,1\}^q, \ldots, \{0,1\}^q\} \end{aligned}$$

  where each signal value is given by $\{s_i\} \in \{0,1\}^q$. Then, individual bit precision can be modified as

$$S_M = \{\{0,1\}^{q-b_0}, \{0,1\}^{q-b_1}, \{0,1\}^{q-b_2}, \ldots, \{0,1\}^{q-b_{N-1}}\}$$

  where $b_i \in \mathbb{Z}^+$ defines the change in the precision value of each element. However, reducing the number of bits in quantum data for an individual element in superposition is an intricate process, as quantum superposition is described by a unique qubit register rather than multiple free bits. For instance, giving a superposed register

$$|x\rangle|a_2a_1a_0\rangle = |x_0\rangle|001\rangle + |x_1\rangle|011\rangle + |x_2\rangle|101\rangle + |x_3\rangle|110\rangle$$

  where elements $|001\rangle$ and $|011\rangle$ could be described by reduced states $|001\rangle = |1\rangle$ and $|011\rangle = |11\rangle$ with bit precision $(q-2)$ and $(q-1)$, respectively. Nevertheless, as all elements are defined by the same quantum register, $|a_2a_1a_0\rangle$, removing a bit is equivalent to discarding some qubit $|a_i\rangle$. Therefore, if $|a_0\rangle$ is removed, then

$$\begin{aligned} |x\rangle|a_2a_1\cancel{a_0}\rangle &= |x_0\rangle|00\cancel{1}\rangle + |x_1\rangle|01\cancel{1}\rangle + |x_2\rangle|10\cancel{1}\rangle + |x_3\rangle|11\cancel{0}\rangle \\ &= |x_0\rangle|00\rangle + |x_1\rangle|01\rangle + |x_2\rangle|10\rangle + |x_3\rangle|11\rangle = |x\rangle|a_2a_1\rangle \end{aligned}$$

where the bit reduction process modifies all elements in superposition, resulting in a signal that differs from the initial signal.

- **Coding Assignment**: this method involves creating new codewords for the elements of an input signal. The codes can be either fixed-length, based on the number of different elements, or variable-length, based on occurrence frequency. For instance, an input signal, S, can be represented as

$$S = \{12, 20, 12, 25, 20, 33, 20, 44, 16, 37, 44, 20, 33, 44, 59\}$$

where $(6 \times 15 = 90)$ bits are required to store the signal.

Fixed-length coding

$$\begin{aligned} C_f =& \{12 \rightarrow 000, 16 \rightarrow 001, 20 \rightarrow 010, 25 \rightarrow 011, \\ & 33 \rightarrow 100, 37 \rightarrow 101, 44 \rightarrow 110, 59 \rightarrow 111\} \\ S_c =& \{000, 010, 000, 011, 010, 100, 010, 110, 001, 101, 110, 010, 100, 110, 111\} \end{aligned}$$

where C_f contains the codewords, and S_c is the coded signal with 45 bits. The fixed coding scheme assigns a new bit representation to each signal element, decreasing storage requirements.

Variable-length coding

$$\begin{aligned} C_v =& \{12 \rightarrow 010, 16 \rightarrow 1010, 20 \rightarrow 11, 25 \rightarrow 1011, \\ & 33 \rightarrow 011, 37 \rightarrow 1000, 44 \rightarrow 00, 59 \rightarrow 1001\} \\ S_c =& \{010, 11, 010, 1011, 11, 011, 11, 00, 1010, 1000, 00, 11, 011, 00, 1001\} \end{aligned}$$

where C_v contains the codewords, and S_c is the coded signal with 42 bits. The variable coding model changes the bit representation based on the occurrence frequency of each element, assigning shorter codes to more likely values and longer codes to less likely ones. Generally, variable-length coding significantly reduces storage requirements compared to fixed-length coding.

On the other hand, quantum code assignment faces several problems that limit its applicability. For example, generating the sets of codewords, C_f and C_v, requires knowledge of the values of the signal, which in some cases can only be accessed

	Wavelet Compression			Input signal Compression	MSE
	Haar	**CDF**	**DB4**		
Coefficients	1.75	1.68	2.02	1.40	0

by a measurement process, increasing quantum complexity. Additionally, variable coding exploits the non-orthogonality of codewords to improve compression results, but the quantum non-orthogonality of elements implies a non-perfect discrimination of states, restricting information retrieval.

5.5.1 Compression Results

The proposed quantum compression method uses a hybrid approach that combines classical and quantum processing in a fixed-length coding scheme. The classical component generates codewords for the input signal, and the quantum component assigns each codeword, reducing computational time while maintaining the compression characteristics of the fixed-length scheme. Figures 5.8, 5.9 and 5.10 display a one-dimensional input signal and its decomposition coefficients for each proposed transform on the left side. The lossless compression results for the input signal and coefficients are shown on the right side, including the compression ratio, $CR = \frac{\text{input bits}}{\text{output bits}}$, and Mean-Square Error (MSE) between the original and the recovered signal. The input signal is a random sequence between $[-20, 20]$ of size $N = 64$.

Figures 5.8, 5.9, and 5.10 depict the compression results for a one-dimensional signal with three decomposition coefficients. The compression ratio for the coefficients is lower than the compressed signal because the wavelet transforms reduces the dynamic range of the signal at each decomposition level, resulting in improved compression. Also, the MSE values reflect the lossless features of the compression scheme. Additionally, Table 5.5 compares the classical and quantum codeword assignment by the number of

queries and operations required to map the codes. The input signal of size $N = 2^n$ has a value range, v_r, between 0 and 255, giving a codeword table of $L_T = 256$ different codes.

Table 5.5 presents a comparative analysis of classical and quantum codeword assignment. The classical process uses the same number of queries and operations as the signal size N. In contrast, the quantum process requires as many queries and uses as many operations as the length of the codeword table L_T, being independent of the signal size N. Consequently, quantum coding reduces computational time, query processes, and number of assignment operations, enhancing the compression process using a hybrid fixed-length coding. Finally, due to the inherent characteristics and limitations of quantum elements, the maximum compression ratio is determined by the maximum codeword length.

Table 5.5: Comparative analysis between classical and quantum codeword assignment.

Domain	**Queries**	**Operations**
Classical	N	N
Quantum	L_T	L_T

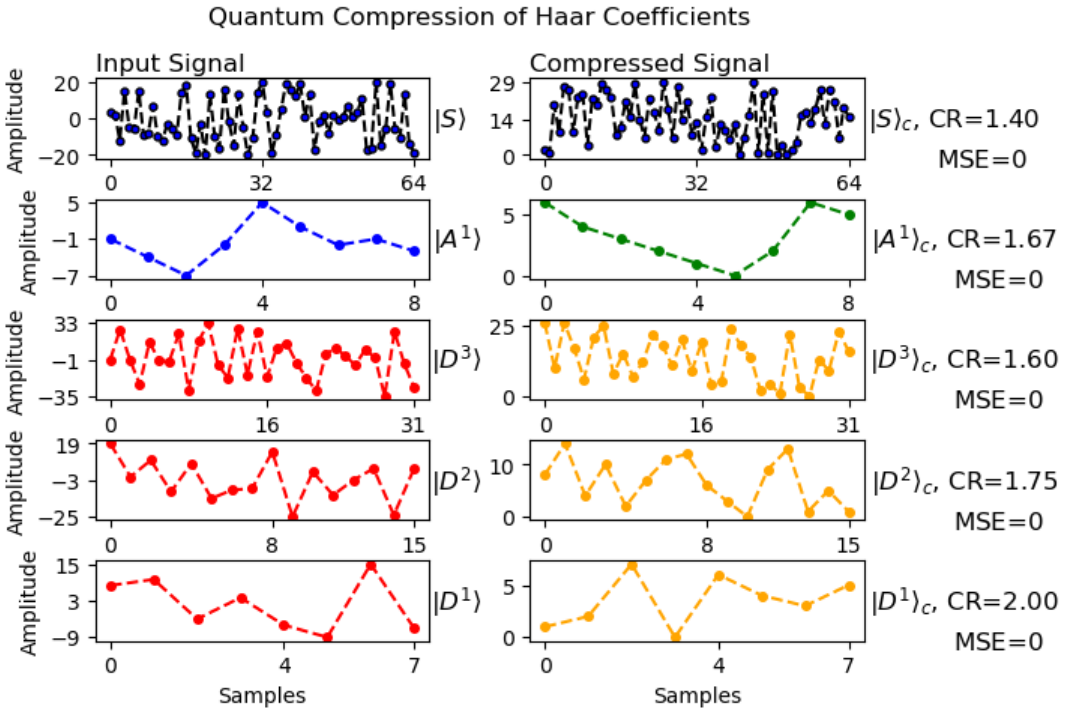

Figure 5.8: Quantum Compression results for the input signal and Haar wavelet coefficients.

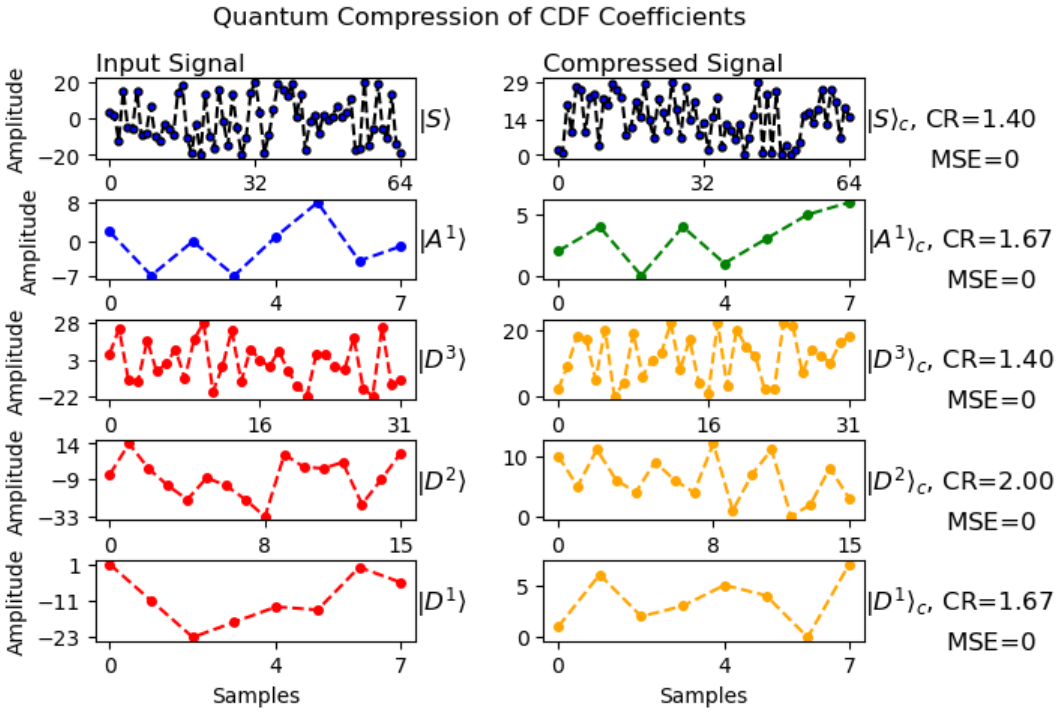

Figure 5.9: Quantum Compression results for the input signal and CDF wavelet coefficients.

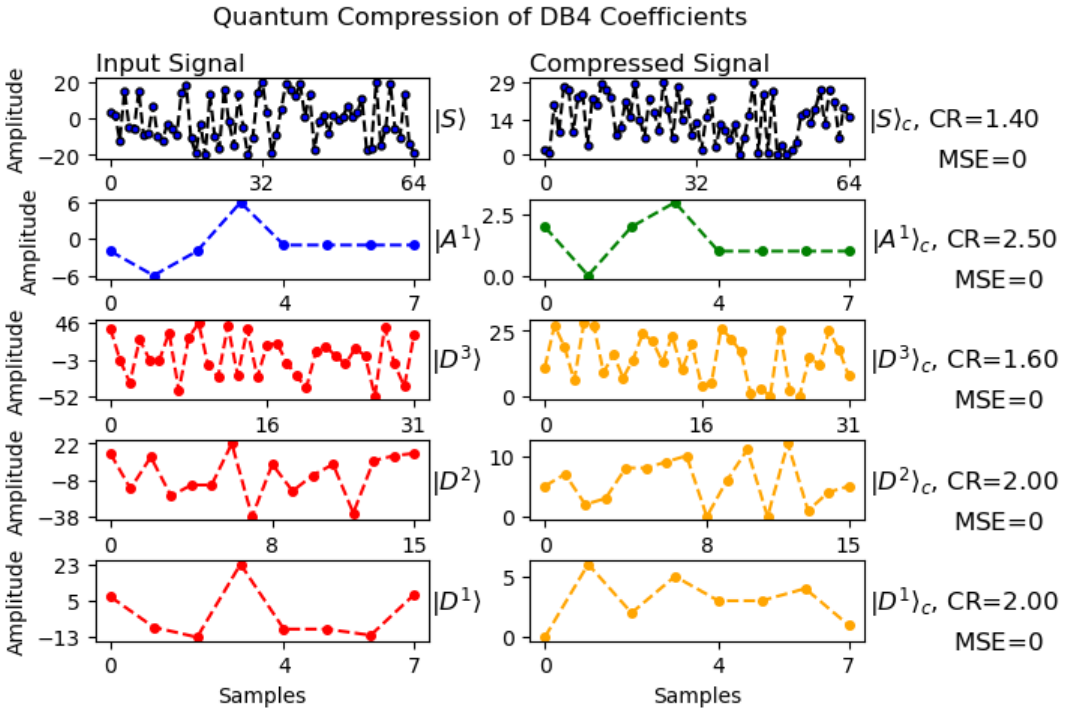

Figure 5.10: Quantum Compression results for the input signal and DB4 wavelet coefficients.

5.6 Summary

In this chapter, we have presented experimental and analytical results of the proposed quantum wavelet transforms and quantum lossless compression scheme. First, we have shown wavelet simulations of the quantum and classical Haar, CDF and DB4 transforms, The quantum versions of these transforms reproduce the classical decomposition behaviour, verifying the applicability and correctness of the quantum transforms. Subsequently, the complexity analysis demonstrated that each transform exhibited polynomial complexity. Additionally, a comparative analysis of the classical and quantum wavelets was conducted, wherein the proposed transforms significantly decrease computational complexity compared to their classical counterparts.

On the other hand, we have analyzed the features of the QBS and the proposed QBRBS quantum format. Consequently, the two representation formats exhibit the same qubit complexity when no element blocks are provided. Nevertheless, the QBRBS reduces the storage requirements when operated on neighboring values, eliminates garbage information, and simplifies signal manipulation by avoiding complex operations. Additionally, the QBRBS format allows for iterative processes with an additional classical pre-processing step.

Finally, we have presented the results of the hybrid lossless compression scheme, where the compression of the decomposition coefficients improves the compression rate compared to the compressed input signal. Furthermore, the quantum codeword assignment reduces the computational time, query processes, and number of assignments compared to the classical procedure, enhancing the compression process.

6

Conclusions and Future Work

6.1 Conclusions

In this research, we presented the quantum definitions of three classes of integer-to-integer wavelet transforms based on the lifting scheme. These include the two orthogonal Haar and Daubechies-4 (DB4) wavelets, and the bi-orthogonal CDF(2,2) transform. Also, we defined the new Quantum Block Representation by Basis States (QBRBS) to facilitate signal manipulation. Furthermore, we developed a hybrid quantum-classical lossless compression scheme based on wavelet decomposition and fixed-length coding. Additionally, we performed a series of analyses, including wavelet simulations, quantum complexities, comparative descriptions of wavelet transforms, features and limitations of quantum representation formats, and compression properties.

We designed the quantum circuits for the one-dimensional versions of the integer wavelet transforms, including addition, subtraction, halving, rounding, shifting, and multiplication operations, allowing for polynomial quantum complexities $O(qn)$, $O(qn)$, and $O(q^2n)$ for the Haar, CDF, and DB4, respectively. The nonlinearities of rounding operations on the lifting schemes were avoided by exploiting the features of the transforms,

where the halving operation implements the quantum linear rounding. We provided the unitary and algorithmic descriptions of the transformations, facilitating both mathematical and computational manipulation. Thus, we fulfill objectives one and three, where we need to define the unitary operators and develop the algorithmic descriptions of the proposed integer wavelet transforms.

Also, we presented the results of the wavelet decomposition simulation in an integer one-dimensional signal, applying the quantum and classical Haar, CDF, and DB4 transforms at three decomposition levels. The error between the coefficients was zero, demonstrating the practical feasibility and applicability of the proposed quantum transforms. Furthermore, we conducted a comparative analysis between the quantum and classical versions of the wavelet transforms, including the type of transformation, the decomposition scheme and domain, the class of wavelet, and their complexities. The analysis showed that the proposed quantum integer transforms significantly decrease the computational complexity compared to their real-valued quantum and integer classical counterparts.

We developed a new Quantum Block Representation by Basis states (QRBBS), which uses basis states to store signal information in superposed and entangled quantum registers. This format allows the storage of signal elements in block components of any size by concatenating different or neighboring elements in the same position coordinate, facilitating signal transformation. The proposed QBRBS has a qubit complexity ($mq+\lceil \log k \rceil$), decreasing storage requirements compared to classical storage. Moreover, a comparative analysis between the traditional Quantum Basis States (QBS) format and QBRBS demonstrates the advantages of the proposed format by facilitating neighboring manipulation, decreasing qubit and gate complexities, eliminating garbage data, and allowing for iterative processes by including a pre-processing classical step. Therefore, we met objective two by developing the QBRBS format to improve signal decomposition.

We derived a lossless compression method employing a hybrid quantum-classical scheme based on wavelet decomposition and fixed-length coding. The compression ra-

tio for the wavelet coefficients is lower than the compressed input signal because the wavelet transforms reduces the dynamic range at each decomposition level, giving an improved compression. However, the fixed-length quantum scheme produces the same compression characteristics as a classical one since the compression model is the same. Nevertheless, the comparative analysis shows that the proposed method significantly decreases the number of queries and operations, being independent of the signal size, and reducing the computational time. Accordingly, we achieve objective four by developing a lossless quantum compression algorithm using fixed-length coding with the proposed quantum transforms.

Finally, we have fulfilled the general objective with the proposed quantum approaches for the one-dimensional integer Haar, CDF, and DB4 wavelet transform based on the lifting scheme, and the design of the algorithmic descriptions for the quantum signal decomposition and the hybrid quantum-classical lossless compression scheme. Additionally, the hypotheses have been validated by designing the unitary operators for developing the proposed transforms, defining a representation format using basis states, and constructing a lossless compression scheme based on the proposed transforms. Moreover, the proposed quantum wavelet transforms expand the quantum information processing toolkit and its application areas, enabling lossless quantum schemes such as compression, watermarking, steganography, cryptography, and filtering processes. Thus, we proved the applicability and feasibility of the developed quantum integer transforms.

6.2 Future Work

Quantum information processing requires different processing tools to manipulate and extract information from input signals in different application areas. Thus, developing new classes of wavelet transforms would expand the set of processing tools, facilitate feature extraction, and allow the exploration of new application areas.

Specifically, the Daubechies and CDF wavelet transform variants are widely used

in information processing areas, such as compression and information hiding. Therefore, the quantum definition of such transformations plays an essential role in advancing the field of quantum computing.

Additionally, the proposed transforms could serve as a basis for other application areas, such as cryptography, watermarking, steganography, and machine learning, by allowing the decomposition and extraction of features from different types of signals.

Finally, this research only considers one-dimensional versions of the Haar, CDF, and DB4 wavelet transforms. However, they could be applied to higher-dimensional signals by decomposing the dimensional space into their basis vectors. Nevertheless, 2D and 3D quantum integer wavelet transforms need to be explored to extend the quantum toolkit and application areas.

Bibliography

[1] Yue Ruan, Xiling Xue, and Yuanxia Shen. Quantum image processing: opportunities and challenges. *Mathematical Problems in Engineering*, 2021:1–8, 2021.

[2] Franklin De Lima Marquezino, Renato Portugal, and Carlile Lavor. *A primer on quantum computing*. Springer, 2019.

[3] Zhaobin Wang, Minzhe Xu, and Yaonan Zhang. Review of quantum image processing. *Archives of Computational Methods in Engineering*, 29(2):737–761, 2022.

[4] Ivan B Djordjevic. *Quantum information processing, quantum computing, and quantum error correction: an engineering approach*. Academic Press, 2021.

[5] Fei Yan, Abdullah M Iliyasu, and Phuc Q Le. Quantum image processing: a review of advances in its security technologies. *International Journal of Quantum Information*, 15(03):1730001, 2017.

[6] Jie Su, Xuchao Guo, Chengqi Liu, and Lin Li. A new trend of quantum image representations. *IEEE Access*, 8:214520–214537, 2020.

[7] Phuc Q Le, Fangyan Dong, and Kaoru Hirota. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Information Processing*, 10:63–84, 2011.

[8] Marina Lisnichenko and Stanislav Protasov. Quantum image representation: A review. *Quantum Machine Intelligence*, 5(1):2, 2023.

[9] Ronald De Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019.

[10] Haisheng Li, Guiqiong Li, and Haiying Xia. Three-dimensional quantum wavelet transforms. *Frontiers of Computer Science*, 17(5):175905, 2023.

[11] Hai-Sheng Li, Shuxiang Song, Ping Fan, Huiling Peng, Hai-ying Xia, and Yan Liang. Quantum vision representations and multi-dimensional quantum transforms. *Information Sciences*, 502:42–58, 2019.

[12] Andreas Klappenecker and Martin Rotteler. Discrete cosine transforms on quantum computers. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, pages 464–468. IEEE, 2001.

[13] Sanjay Chakraborty, Soharab Hossain Shaikh, Amlan Chakrabarti, and Ranjan Ghosh. An image denoising technique using quantum wavelet transform. *International Journal of Theoretical Physics*, 59:3348–3371, 2020.

[14] Hai-Sheng Li, Ping Fan, Huiling Peng, Shuxiang Song, and Gui-Lu Long. Multilevel 2-d quantum wavelet transforms. *IEEE Transactions on Cybernetics*, 52(8):8467–8480, 2021.

[15] Hai-Sheng Li, Ping Fan, Hai-ying Xia, Shuxiang Song, and Xiangjian He. The multilevel and multi-dimensional quantum wavelet packet transforms. *Scientific reports*, 8(1):13884, 2018.

[16] Khalid Sayood. *Introduction to data compression*. Morgan Kaufmann, 2017.

[17] Arne Jensen and Anders la Cour-Harbo. *Ripples in mathematics: the discrete wavelet transform*. Springer Science & Business Media, 2001.

[18] Freddy Alejandro Chaurra-Gutierrez, Claudia Feregrino-Uribe, Julio Cesar Perez-Sansalvador, and Gustavo Rodriguez-Gomez. Qist: One-dimensional quantum integer wavelet s-transform. *Information Sciences*, 622:999–1013, 2023.

[19] Hai-Sheng Li, Ping Fan, Hai-Ying Xia, Huiling Peng, and Shuxiang Song. Quantum implementation circuits of quantum signal representation and type conversion. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(1):341–354, 2018.

[20] Colin P Williams. *Explorations in quantum computing*. Springer Science & Business Media, 2010.

[21] Panchi Li, Bing Wang, Hong Xiao, and Xiande Liu. Quantum representation and basic operations of digital signals. *International Journal of Theoretical Physics*, 57:3242–3270, 2018.

[22] Venkateswaran Kasirajan. *Fundamentals of quantum computing*. Springer, 2021.

[23] Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018.

[24] Fei Yan, Abdullah M Iliyasu, and Salvador E Venegas-Andraca. A survey of quantum image representations. *Quantum Information Processing*, 15:1–35, 2016.

[25] Yi Zhang, Kai Lu, Yinghui Gao, and Mo Wang. Neqr: a novel enhanced quantum representation of digital images. *Quantum information processing*, 12(8):2833–2860, 2013.

[26] Edgard Muñoz-Coreas and Himanshu Thapliyal. Everything you always wanted to know about quantum circuits, August 2022.

[27] Joschka Roffe. Quantum error correction: an introductory guide. *Contemporary Physics*, 60(3):226–245, July 2019.

[28] Mario Mastriani. Quantum image processing? *Quantum Information Processing*, 16(1):27, 2017.

[29] Mario Mastriani. Quantum image processing: the pros and cons of the techniques for the internal representation of the image. a reply to: A comment on "quantum image processing?". *Quantum Information Processing*, 19(5):156, 2020.

[30] Rodrigo Capobianco Guido. Wavelets behind the scenes: Practical aspects, insights, and perspectives. *Physics Reports*, 985:1–23, 2022.

[31] Dengsheng Zhang and Dengsheng Zhang. Wavelet transform. *Fundamentals of image data mining: Analysis, Features, Classification and Retrieval*, pages 35–44, 2019.

[32] Michel Misiti, Yves Misiti, Georges Oppenheim, and Jean-Michel Poggi. Wavelet toolbox. *The MathWorks Inc., Natick, MA*, 15:21, 1996.

[33] Anders la Cour-Harbo and Arne Jensen. Wavelets and the lifting scheme. 2007.

[34] David Bull and Fan Zhang. *Intelligent image and video compression: communicating pictures.* Academic Press, 2021.

[35] Md Ershadul Haque, Manoranjan Paul, Anwaar Ulhaq, and Tanmoy Debnath. Advanced quantum image representation and compression using a dct-efrqi approach. *Scientific Reports*, 13(1):4129, 2023.

[36] Yan Ma and Nan-Run Zhou. Quantum color image compression and encryption algorithm based on fibonacci transform. *Quantum Information Processing*, 22(1):39, 2023.

[37] Nan Jiang, Hao Hu, Yijie Dang, and Wenyin Zhang. Quantum point cloud and its compression. *International Journal of Theoretical Physics*, 56:3147–3163, 2017.

[38] Chao-Yang Pang, Ri-Gui Zhou, Ben-Qiong Hu, WenWen Hu, and Ahmed El-Rafei. Signal and image compression using quantum discrete cosine transform. *Information Sciences*, 473:121–141, 2019.

[39] Caroline Rogers and Rajagopal Nagarajan. Lossless quantum data compression and quantum kolmogorov complexity. *International Journal of Quantum Information*, 9(2), 2011.

[40] Nan Jiang, Xiaowei Lu, Hao Hu, Yijie Dang, and Yongquan Cai. A novel quantum image compression method based on jpeg. *International Journal of Theoretical Physics*, 57:611–636, 2018.

[41] Xiao-Zhen Li, Wei-Wei Chen, and Yun-Qian Wang. Quantum image compression-encryption scheme based on quantum discrete cosine transform. *International Journal of Theoretical Physics*, 57:2904–2919, 2018.

[42] Isaac L Chuang and Dharmendra S Modha. Reversible arithmetic coding for quantum data compression. *IEEE Transactions on Information Theory*, 46(3):1104–1116, 2000.

[43] Kim Bostroem and Timo Felbinger. Lossless quantum data compression and variable-length coding. *Physical Review A*, 65(3):032313, 2002.

[44] Peter Hoyer. Efficient quantum transforms. *arXiv preprint quant-ph/9702028*, 1997.

[45] Amir Fijany and Colin P Williams. Quantum wavelet transforms: Fast algorithms and complete circuits. In *Quantum Computing and Quantum Communications: First NASA International Conference, QCQC'98 Palm Springs, California, USA February 17–20, 1998 Selected Papers*, pages 10–33. Springer, 1999.

[46] Andreas Klappenecker. Wavelets and wavelet packets on quantum computers. In *Wavelet Applications in Signal and Image Processing VII*, volume 3813, pages 703–713. SPIE, 1999.

[47] Darwin Gosal and Wayne Lawton. Quantum haar wavelet transforms and their applications, 2001.

[48] Hai-Sheng Li, Ping Fan, Hai-ying Xia, and Shuxiang Song. Quantum multi-level wavelet transforms. *Information Sciences*, 504:113–135, 2019.

[49] Xian-Hua Song, Shen Wang, Shuai Liu, Ahmed A Abd El-Latif, and Xia-Mu Niu. A dynamic watermarking scheme for quantum images using quantum wavelet transform. *Quantum information processing*, 12:3689–3706, 2013.

[50] Shahrokh Heidari, Mosayeb Naseri, Reza Gheibi, Masoud Baghfalaki, Mohammad Rasoul Pourarian, and Ahmed Farouk. A new quantum watermarking based on quantum wavelet transforms. *Communications in theoretical Physics*, 67(6):732, 2017.

[51] Wen-Wen Hu, Ri-Gui Zhou, Ahmed El-Rafei, and She-Xiang Jiang. Quantum image watermarking algorithm based on haar wavelet transform. *IEEE Access*, 7:121303–121320, 2019.

[52] Yiming Yu, Jie Gao, Xiaoyi Mu, and Shumei Wang. Adaptive lsb quantum image watermarking algorithm based on haar wavelet transforms. *Quantum Information Processing*, 22(5):180, 2023.

[53] Shen Wang, Xianhua Song, and Xiamu Niu. A novel encryption algorithm for quantum images based on quantum wavelet transform and diffusion. In *Intelligent Data analysis and its Applications, Volume II: Proceeding of the First Euro-China Conference on Intelligent Data Analysis and Applications, June 13-15, 2014, Shenzhen, China*, pages 243–250. Springer, 2014.

[54] Jian Wang, Yacong Geng, and Jiqiang Liu. Adaptive quantum image encryption method based on wavelet transform. *arXiv preprint arXiv:1901.07762*, 2019.

[55] Hai-Sheng Li, Qingxin Zhu, Ming-Cui Li, Hou Ian, et al. Multidimensional color image storage, retrieval, and compression based on quantum amplitudes and phases. *Information Sciences*, 273:212–232, 2014.

[56] Md Ershadul Haque, Manoranjan Paul, Faranak Tohidi, Anwar Ulhaq, and Tonmoy Debnath. Enhancing image representation and compression: An innovative nz-nqer framework with block truncation quantum coding. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 304–311. IEEE, 2023.

[57] Nan-Run Zhou, Lang-Xin Huang, Li-Hua Gong, and Qing-Wei Zeng. Novel quantum image compression and encryption algorithm based on dqwt and 3d hyperchaotic henon map. *Quantum Information Processing*, 19(9):284, 2020.

[58] Hai-Sheng Li, Zhu Qingxin, Song Lan, Chen-Yi Shen, Rigui Zhou, and Jia Mo. Image storage, retrieval, compression and segmentation in a quantum system. *Quantum information processing*, 12(6):2269–2290, 2013.

[59] William NN Hung, Xiaoyu Song, Guowu Yang, Jin Yang, and Marek Perkowski. Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. *IEEE transactions on Computer-Aided Design of integrated circuits and Systems*, 25(9):1652–1663, 2006.

[60] Himanshu Thapliyal and Edgard Muñoz-Coreas. Design of quantum computing circuits. *IT Professional*, 21(6):22–26, 2019.

[61] Himanshu Thapliyal. Mapping of subtractor and adder-subtractor circuits on reversible quantum gates. In *Transactions on Computational Science XXVII*, pages 10–34. Springer, 2016.

[62] RiGui Zhou, WenWen Hu, GaoFeng Luo, XingAo Liu, and Ping Fan. Quantum realization of the nearest neighbor value interpolation method for ineqr. *Quantum Information Processing*, 17:1–37, 2018.

[63] Edgard Muñoz-Coreas and Himanshu Thapliyal. Quantum circuit design of a t-count optimized integer multiplier. *IEEE Transactions on Computers*, 68(5):729–739, 2018.

[64] Freddy Alejandro Chaurra-Gutierrez, Julio Cesar Perez-Sansalvador, Gustavo Rodriguez-Gomez, and Claudia Feregrino-Uribe. Qbrbs: Quantum block representation by basis states. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 125–132. IEEE, 2022.

[65] Freddy Alejandro Chaurra-Gutierrez, Claudia Feregrino-Uribe, Julio Cesar Perez-Sansalvador, and Gustavo Rodriguez-Gomez. Quantum mean filter: Features and issues. In *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages xxx–xxx. IEEE, 2024.

[66] Gustavo Rodriguez-Gomez, Claudia Feregrino-Uribe, Julio Cesar Perez Sansalvador, et al. One-dimensional quantum integer cdf (2, 2) wavelet transform. *Gustavo and Feregrino-Uribe, Claudia and Perez Sansalvador, Julio Cesar, One-dimensional Quantum Integer CDF (2, 2) Wavelet Transform (March 13, 2023)*, 2023.

[67] Robert A Brown, M Louis Lauzon, and Richard Frayne. A general description of linear time-frequency transforms and formulation of a fast, invertible transform that samples the continuous s-transform spectrum nonredundantly. *IEEE Transactions on Signal Processing*, 58(1):281–290, 2009.