Quantum Approach to The Integer Wavelet Transform and Its Application to Quantum Lossless Compression

M.Sc Freddy Alejandro Chaurra Gutiérrez, Ph.D. Claudia Feregrino Uribe, Ph.D. Julio César Pérez Sansalvador, Ph.D. Gustavo Rodríguez Gómez

Technical Report No. CCC-22-001
August, 2021

Luis Enrique Error 1,
Santa María Tonantzintla,
72840, Puebla, México

Abstract

With the rapid technological progress, the need for high processing and storage capacities has increased dramatically. Therefore, it is a necessity to discover new ways to manipulate and transform information. One potential solution is quantum information processing, which significantly reduce the amount of stored data, the number of operations, and the complexity of classical tools such as the wavelet transform (WT). WT is a primary tool in many areas, such as encryption, signal coding, watermarking, compression, de-noising, and information retrieval. Its classical relevance drives its progress at the quantum level, leading to improvements in computation efficiency for the one-, two-, and three-dimensional quantum wavelets transform. However, conventional, real-valued WT is not suitable for lossless applications and is computationally complex. The Integer-to-Integer WT (IWT) is another kind of transform that maps integer to integer, which uses the lifting scheme to perform the signal decomposition analysis. This scheme reduces the computational cost, allows practical lossless applications over real-valued WT, and generates new wavelet families. So far, there is no definition of the QWT for the integer version (Q-IWT), which could be valuable in quantum information processing. Therefore, we propose a quantum approach for the one-dimensional integer wavelet transform for Haar, Daubechies, and CDF kernels, including quantum algorithms for signal decomposition and lossless compression. We will evaluate the proposed transform and the compression application using complexity and mathematical analysis, performance, flexibility, signal recovery, entropy, and noise addition metrics. Additionally, we will use IBM's simulation environment as a means of analysis and verification.

Keywords: Quantum Computing, Quantum Information Processing, Wavelet Transform, Integer-to-Integer, Lossless Compression.

Contents

1 Introduction

Quantum computing enables the development and experimental realization of quantum algorithms in many areas to solve complex computational problems [1–4]. For example, the algorithms of Shor and Grover run on a quantum computer with an exponential and quadratic speedup, respectively [2, 4, 5]. The ideas behind these algorithms allow the construction of new techniques and solutions to classical problems in areas such as cryptography, watermarking, information processing, optimization, and machine learning [2, 6–9]. However, there is a limited number of these quantum solutions. In principle, this is due to the difficulty in the transition between the classical and quantum world. In addition to the relatively new emergence of the field [2, 4, 10–12]. Therefore, the development of solutions that take advantage of the properties of quantum mechanics is a new challenge to be explored and solved [2,4,5].

With the rapid technological progress, the need for high processing and storage capacities has increased dramatically. Therefore, it is a necessity to discover new ways to manipulate and transform information [5, 6, 11]. One potential solution is quantum information processing, which significantly reduce the amount of stored data, the number of operations, and the complexity of classical operations [5, 6]. Until now, quantum algorithms such as feature extraction, quantum representation, quantum transformations, and quantum operations have demonstrated quantum-theoretical power in contrast to classical counterparts [5, 6, 10, 11]. However, the quantum paradigm also introduces new constraints and challenges that need to be studied such as the unitarity of quantum operators, the limitation of nonlinearity, the quantum cost of implementation, information representation and manipulation [2,4].

Wavelet Transform (WT) is a primary tool in the information processing fields such as encryption, signal coding, feature extraction, compression, information retrieval, de-noising, and watermarking [5, 6, 10, 12]. Its classical relevance drives its progress at the quantum level, leading to improvements in computational efficiency. Preliminary work presented a quantum version of one-dimensional WT for two main kernels, the Haar (H) and Daubechies (D4) bases [13–15]. There is a multi-level approach for 1D-HQWT and 1D-D4QWT using a generalized tensor product [11], multi-level two-dimension QWT based on the packet for Haar and Daubechies bases [10, 11], quantum circuits of the multi-level and single-level QWT [11, 12], and a new multi-level version of 2D-QWT involving entanglement and superposition [11]. However, research on Quantum Wavelet Transform (QWT) is still under progress and improvement.

Among the conventional wavelet transform, also called real-valued wavelet transform, there is another attractive version of this transform called the integer wavelet transform (IWT), which maps integer signals to integer decomposition values [4]. This transform is especially useful in applications where minimal perturbations of the primary data are not acceptable, as it allows for true lossless processing [16–18]. In addition, the IWT requires less storage capacity and reduces the computational cost concerning the conventional WT [16–19]. So far, to our knowledge, there is neither a quantum construction for the integer version of the wavelet transform nor a definition of the QWT for kernels other than the Haar and Daubechies bases.

Therefore, due to the necessity of high processing and storage capacities of current information, the requirements for efficient compression schemes, the demand for efficient solutions in different areas, and the progress in quantum computing, this research focuses on the quantum computing field. We propose to design the one-dimensional Quantum Integer Wavelet Transform (Q-IWT) approach, which could serve as a basis for new proposals in the quantum area. This quantum transform should provide an improvement on computational complexity over its integer classical counterpart. We will present a quantum definition for some primary wavelet kernels through permutation matrices, mainly the Haar, the Daubechie-4 and the CDF family. Also, we will develop quantum algorithms for signal decomposition and lossless compression using the quantum integer wavelet transform. Finally, we will simulate the quantum approaches on the software development kit Qiskit by IBM.

2 Background

In this section, we present the general ideas about quantum computing, wavelet analysis, and compression. First, we define general concepts about quantum computing, including quantum representation and manipulation, quantum operator's factorization, quantum algorithms, and applications. These elements will allow us to know the formalism used in this research.

Also, we describe the classical wavelet analysis, which includes the wavelet transform, the decomposition process, wavelet classification, and wavelet implementation. We focus on a classical description of the wavelet transform as it will allow us to understand some of the necessary elements for its quantum construction.

Finally, we introduce the data compression process and the application of lossless compression. These concepts will give us the general ideas of the compression process and the importance of identifying and modeling the structure present in the data.

2.1 Information Representation

Quantum computers use quantum phenomena such as photon polarization, levels of atomic energy, or nuclear spin to represent information and employ quantum systems to do computation. A primary feature of quantum computation is that this follows a probabilistic nature. If we try to see the results, we can only obtain a particular outcome with a certain probability [2].

The basic unit of information in a quantum computer are qubits like the classical bits. We can describe a qubit by a vector in a superposition of states, where they overlap, unlike bits that can only have one value at a time. Due to this superposition nature, a qubit can represent more information than classical bits. It is known as quantum parallelism [2, 20]. The general computational states are represented in a vectorial form as follows:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (1)$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2}$$

Thus, a unit vector of states superposition describes a qubit as a linear combination.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \sum_{k=0}^{1} \delta_k |k\rangle \tag{3}$$

The parameters α and β are complex numbers called amplitudes and hold the constraint $|\alpha|^2 + |\beta|^2 = 1$. The $|0\rangle$, and the $|1\rangle$ states form an orthonormal basis for the vector space [3, 20]. However, the power of quantum computing becomes apparent if we consider a superposition of multiples qubits (quantum registers). We obtain quantum registers through the tensor product [20]. For example, two classical bits give four possible states $(00, 01, 10, 11)$. Similarly, the tensor product between two qubits generates the four possible superposed states $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$ [3,20]. The four possible qubit states are expressed as the tensor product as follows:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \tag{4}$$

$$|\psi\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle \tag{5}$$

Therefore, n qubits can simultaneously express 2^n states. It implies that if we have many quantum bits, the computation speed can increase exponentially [3]. Nevertheless, we cannot examine a qubit to determine its quantum state. Instead, when we read (measure) a qubit, we only get one of the basis states with a certain probability, 0 with probability $|\alpha|^2$ or 1 with probability $|\beta|^2$. A qubit can exist in an overlapping state until observation. Therefore, to compute and take advantage of the superposition state, qubits states must be manipulated and transformed in ways that lead to a desirable measurement outcome [3, 20].

2.1.1 Quantum Formats

Quantum formats allow to encode and efficiently manipulate information as an extension of the primary state representation. These formats enable the storage of different kinds of information (image, audio, and video) using amplitudes, phases, and basis quantum states to perform various processing tasks and operations such as compression, filtering, denoising, retrieval, and general transformations [6,21–23].

The basic idea to store information consist of two parts that capture information about amplitude and time or spatial position about every component that makes up the signal. A general representation for a quantum signal in basis states is described by

$$|S\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} |C_k\rangle \otimes |P_k\rangle \tag{6}$$

where $|C_k\rangle$ represent the amplitude information of the quantum signal and $|P_k\rangle$ indicates the corresponding position information [6]. This representation is similar to the Novel Enhaced Quantum Representation (NEQR) format [24].

Another representation stores the signal information in the amplitude coefficients and position in a quantum state given by

$$|S\rangle = \frac{1}{2^n} \sum_{k=0}^{2^n-1} (\alpha_k|0\rangle + \beta_k|1\rangle) \otimes |P_k\rangle \tag{7}$$

where α_k and β_k are the amplitude coefficients and $|P_k\rangle$ the corresponding signal component position [25]. The most widely used image representation is the Flexible Representation of Quantum Images (FRQI), which captures the color and position information into amplitudes of a quantum state. As well as image representation, there are quantum movie and digital audio formats inspired by image representation. However, research in quantum movies and audio representation is still relatively rare [6,26–28].

The importance of quantum formats lies in decreasing the cost of information storage and the advantages that these formats provide for different applications. For example, we require $2^n \times 2^n \times 8$ bits to store an $2^n \times 2^n$ gray image represented by 8 bits. FQRI requires only $2n+1$ qubits are needed, and quantum audio representation only needs n qubits to represent a signal of size 2^n [5,29].

2.2 Quantum Information Manipulation

Quantum operators serve as mechanisms to transform the quantum states of the system. The most common way to implement this is through quantum gates to manipulate the quantum information by unitary transformations in matrix form [2, 3, 20]. Therefore, a quantum gate, U, applied to arbitrary states is defined by

$$|b\rangle = U\,|a\rangle \tag{8}$$

where U is the quantum operator in matrix form, and $|b\rangle$ and $|a\rangle$ are vectors.

Since unitary transformations are reversible, quantum computation is reversible by nature. Reversible quantum computation does not affect the universality of quantum circuits, and there are a set of quantum universal gates [3, 20]. Three important single-qubit gates are quantum NOT gate (X), Identity (I) and Hadamard transform (H) given by

$$X : \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} ; X\,|0\rangle \to |1\rangle\,, X\,|1\rangle \to |0\rangle \tag{9}$$

$$I : \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ; I\,|0\rangle \to |0\rangle\,, I\,|1\rangle \to |1\rangle \tag{10}$$

$$H : \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} ; H\,|0\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), H\,|1\rangle \to \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{11}$$

The Hadamard gate is one of the primary quantum gates since it allows an equiprobable superposition of states [3, 20]. Another useful gate is the two-qubit controlled-not (CNOT), which enables the creation of an entangled state [20]. The CNOT applies X gate on a target qubit if the control qubit is $|1\rangle$. It is described as:

$$CNOT(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \to (|00\rangle + |01\rangle + |11\rangle + |10\rangle) \tag{12}$$

where the first qubit is the control and the second the target qubit.

The three-qubit Toffoli gat is a generalization of the CNOT-gate, where the target qubit changes its state if the two control qubits are $|1\rangle$. In addition to the gates described above, there are many other useful gates such as the Pauli gates (Y, Z, X, I), Phase Shift gates (R_θ), $\pi/8 - gate$ (T) and Swap gate. These gates can perform various computations except cloning an unknown qubit state (no-cloning theorem) [2, 3].

2.3 Quantum Operators Design

One of the main constraints in quantum computing is that operations must be unitary. However, classical computations generally do not use unitary operators, and operators that are easy and inexpensive classically are not always easy to implement in the quantum world. Therefore, general techniques for defining and computing unitary operators or transformations are needed [10, 11, 13].

A solution to develop an efficient algorithm to define unitary operators is to factor a general operator into a small number of unitary operators [13, 14]. For example, we can describe every 2×2 operator M as a linear combination of four basic operators.

$$M = \alpha I + \beta X + \gamma Y + \delta Z$$

where α, β, γ and δ are complex constants. I, X, Y, and Z are Identity, Negation, Y and Z operators, respectively. In addition, polar and singular value decomposition are ways of factorizing general linear operators up into products of unitary operators [30, 31]. Nevertheless, finding an efficient factorization can be challenging because it could produce exponential terms in the factorization [4, 13, 14].

2.4 Quantum Algorithms and Applications

Quantum computers outperform classical ones because of their intrinsic characteristics, such as reversibility, superposition, and entanglement. For example, quantum circuits can simulate classical logical circuits. Also, they can perform any classical deterministic computation and non-deterministic computation [3, 30]. Furthermore, quantum parallelism allows an exponential increase in storage capacity and a decrease in time complexity [3, 20]. However, to take advantage of quantum characteristics, we need to perform sequences of transformations and measurements to extract information through quantum algorithms [2,3]. There are best-known ways to develop quantum algorithms [3,32–34], they are:

- Approaches based on Quantum Fourier Transform (QFT),
- Approaches based on the Grove's search algorithm (GO), and
- Algorithms for simulate or solve problems in quantum physics.

The number of known quantum algorithmic paradigms is smaller than that of classical paradigms. It is due to the constraint of unitarity of the operators and the limitations of measurement. Nevertheless, the search for new design paradigms is is constant [32–34].

Some of the most important known quantum algorithms are The Shor algorithms for factoring and discrete logarithm based upon the QFT [35], The Hidden subgroup algorithm [36], Grover's search algorithm [37], Bernsteins-Vazirani algorithm [38], linear systems solver [39], quantum algorithms for random walk [40], and adiabatic quantum algorithm [41]. Thus, quantum mechanics provides some solutions to complex classical problems opening a new world of unimaginable algorithms in classical computations [2,3,20,34].

2.4.1 Quantum Complexity

The choice of a specific algorithm to solve a problem depends on the available time and memory resources. The complexity is related to the rate at which a resource grows as a function of the problem size. Therefore, we have two primary measurements in quantum computing [10,30]:

- **Quantum cost** is the total number of basic operations which simulate the circuit.
- **Time complexity** is the total number of time steps executed serially.

2.4.2 Applications of Quantum Computing

There are several representative applications of quantum computing, such as quantum codes [2], quantum key distribution [2], quantum teleportation [2, 3], quantum watermarking [5], quantum information processing [5, 6], and quantum communication [2, 5].

The challenge of constructing universal quantum computing mainly comes from technical issues, such as quantum state initialization, manipulation of multiple qubits, external insulation, and holding low temperatures [6]. These problems impose new technological-scientific constraints and challenges in developing quantum tools. A trend in recent years has been the study of quantum error correction algorithms and the construction of noisy quantum computers known as noisy intermediate-scale quantum (NISQ) computers composed of noisy qubits to perform imperfect operations. The goal is to leverage the limited resources to perform classically complex tasks in some practical application areas [1, 6].

2.5 Wavelet Analysis

Wavelet analysis is a set of tools and techniques for signal analysis and decomposition like Fourier analysis. One of the main advantages of using wavelets is that the time information is not lost and allows to perform local analysis, unlike Fourier analysis. Also, wavelet analysis identify trends, breakdown points, discontinuities, and self-similarity [42, 43].

Wavelet analysis is a windowing analysis technique with variable-sized regions using different basis waveforms to decompose a signal into shifted and scaled versions of the original wavelet. It is described by the continuous wavelet transform:

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\Psi(scale, position, time)dt \tag{13}$$

The result of the wavelet transform are wavelet coefficients C, which can represent the original signal through scaled and shifted basis wavelets [42]. These coefficients represent the degree of correlation of the wavelet, Ψ, with a section of the signal, f, in the following way [42]. Figure 1 shows the decomposition process.

a. Take a wavelet and compare it to a section of the original signal.

b. Calculate the coefficient C.

c. Shift the wavelet and calculate the coefficient again until the whole signal is covered.

d. Scale the wavelet and repeat the steps above for all scales.

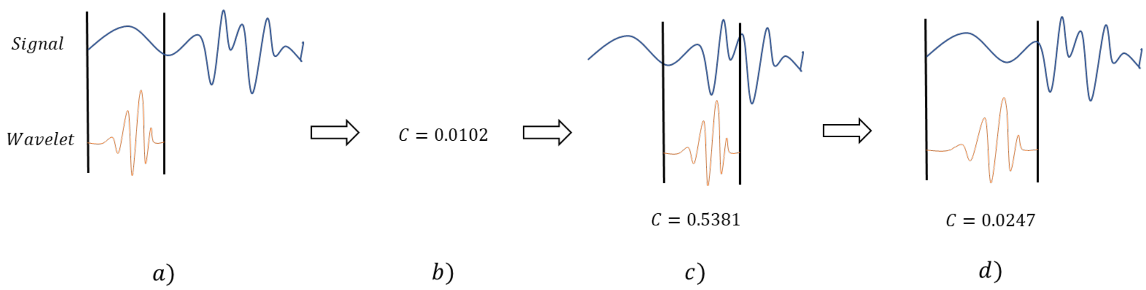

Figure 1: Wavelet decomposition process [42].

A high scale parameter corresponds to the stretched wavelets and a low scale to the compressed wavelet. The stretched wavelet enables to compare a long section of the signal, associated with slowly changing or low frequency. A compressed wavelet is related to details or high-frequency characteristics. Low frequencies are the approximation coefficients, and high frequencies the details coefficients [42, 43].

2.5.1 Wavelet Families

The main wavelet shapes (kernels) include Haar, Daubechies family, Biorthogonal, Morlet, and Mexican Hat. We select these kernels according to the original signal shape or the application area. For example, short wavelets are usually more effective for detecting signal discontinuities. Smooth wavelets are suitable for detecting singularities and revealing information hidden in the noise, and the packet analysis does a better job of removing noise [42, 43]. Figure 2 shows some wavelet kernels.

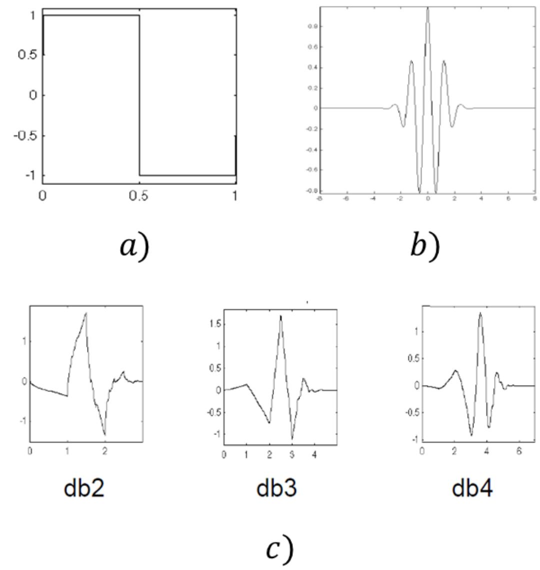

Figure 2: Wavelet families. a) Haar waveform, b) Morlet waveform, and c) Daubechies families waveform [42].

2.5.2 Multi-level Decomposition

The signal decomposition is an iterative process where we broke the original signal into many lower-resolution components. It is called the wavelet decomposition tree [42]. Figure 3 present the decomposition signal tree.

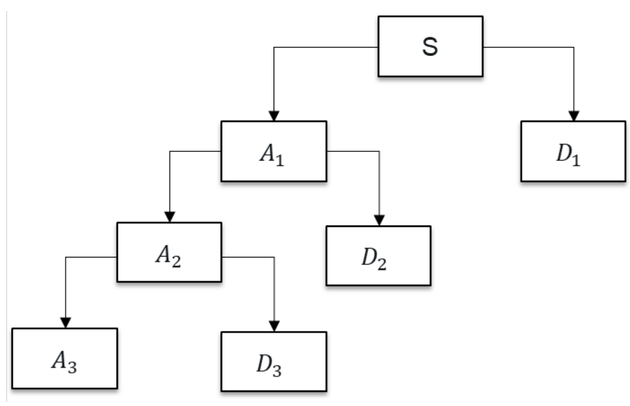

Figure 3: Decompostion signal tree [42].

where the signal, S, is decomposed into principal approximation and detail coefficients, A_1 and D_1, respectively. Then, it brokes the new approximation coefficient into two new elements of approximation and detail, and so on. This process is like a signal filtering process. The number of decomposition levels is selected based on the nature of the signal or a suitable criterion [42].

2.6 Wavelet Transform Classification

We can represent wavelet transform in a continuous or discrete domain, where the main differences are the scales and shifts at which they operate. Continuous wavelet transform operates at all scales and offsets of the wavelet kernel, where we perform a smooth wavelet analysis over the whole domain. But calculating wavelet coefficients at each scale and shift requires an enormous amount of work. However, when we perform any information processing on a computer, we use discrete signals. Therefore, it is necessary to choose only a subset of elements of analysis in the decomposition. We can perform this analysis by using the discrete wavelet transform (DWT) [42,43].

Conventional wavelet transforms produce real-valued coefficients in the transform domain, which is limited by the finite precision of the computer. Also, applications using the real-valued transforms require auxiliary data to store fractional parts for retrieval processes. In contrast, it is possible to construct integer-to-integer wavelet transforms, which convert the real-domain of the coefficients to the integer domain. Unlike conventional wavelet transformations, the integer version is invertible in infinite precision [44]. Figure 4 shows the general classification of the wavelet transform.

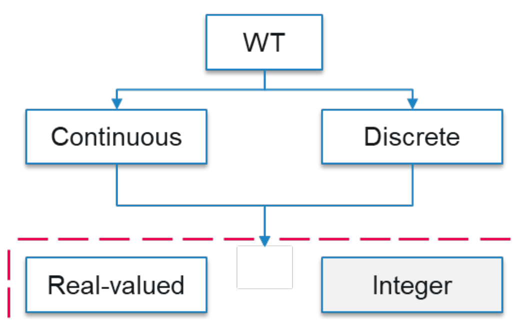

Figure 4: Wavelet transform classification

2.7 Wavelet Transform Implementation

An effective way to implement the wavelet transform is to use a filtering scheme known as a two-channel subband coder or pyramid algorithm. This scheme yields a practical filtering algorithm to obtain a fast wavelet transform to signal decomposition by convolving the signal with a filter. The filter is related to the wavelet shape and is designed based on quadrature mirror decomposition filters. Figure 5 shows the filtering process, which is described by lowpass and highpass filters [42, 45].

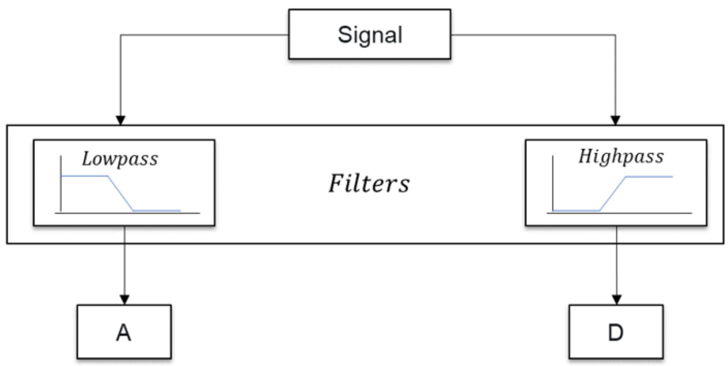

Figure 5: Wavelet filter bank decomposition

Another way to implement the wavelet transform is based on the lifting scheme, which uses a filter bank structure to compute the decomposition values (approximation and details coefficients) through finite sequences of simple filtering steps in the spatial domain. This decomposition corresponds to a factorization of the subband filters into elementary matrices. The basic idea is to split a signal into two disjoint sets, even, and odd samples. Then, we generate the decomposition components by a prediction and update operations. The goal of these operations is to exploit some characteristics and preserve the internal structure of the signal [4, 46–48]. Figure 6 shows a general prediction and update scheme, where A and D are the approximation and detail coefficients of the signal, respectively [4, 46, 48].

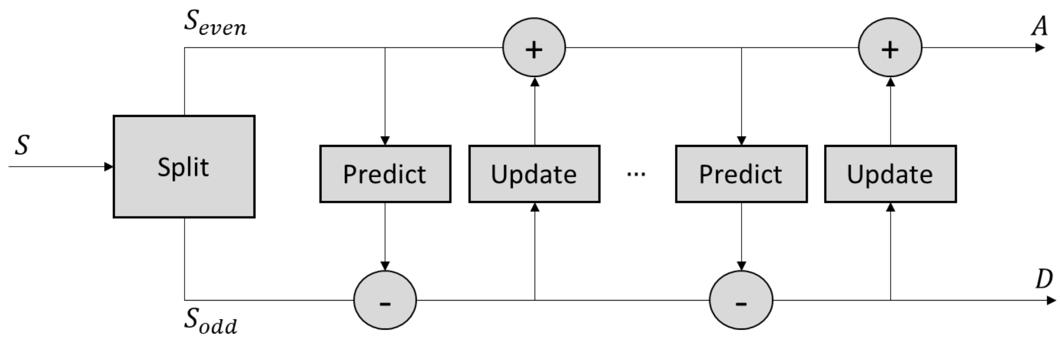

Figure 6: Lifting scheme of prediction and update steps.

2.8 Compression

Data compression consists of representing information in a compact form. We create this compact representation by characterizing and taking advantage of the structure that exists in the data. We need to compress data due to the increase in the transmission of information and the number of elements needed to represent this information [49, 50].

Generally, we use the statistical structure to provide compression, but it is not the only type of structure in data. There are many other kinds of structures in the data of different forms that can be exploited for compression. Furthermore, we can also take the perceptual limitations of the receiver to discard irrelevant information [49, 50].

Depending on the reconstruction requirements, we can divide data compression schemes into two classes: lossless compression, in which the original and reconstructed data are identical, and lossy compression, which generally provides much higher compression but with some loss of information [49].

2.8.1 Lossless Compression

Lossless compression implies no loss of information and that we can recover the original data exactly from the compressed data. We use this scheme for applications that do not allow any difference between the original and reconstructed data, such as medical, space, and banking applications. For example, a lossy medical image can

9	11	11	11	14	13	15	17	16	17

Figure 7: Sequence of numbers to store, x_n.

lead to artifacts that could seriously confuse the specialist, so be careful when using a compression scheme that generates a reconstruction different from the original data [49].

We can evaluate a compression algorithm in several ways by measuring the complexity, the memory needed to implement it, how fast the algorithm is, the amount of compression, and how closely the reconstruction resembles the original data [49].

Although reconstruction requirements may dictate whether a compression scheme should be lossy or lossless, the compression scheme we use will depend on different factors. Some of the most important factors are the characteristics of the data to be compressed. A compression technique that works well for compressing text may not work as well for compressing images. Each application presents a different set of challenges [49].

We can divide data compression into two phases. The first is called modeling, in which we try to extract information about any redundancy or pattern in the data. Then, we describe the structure of the data in the form of a model. The second phase is called encoding, in which we transform the original data into a new representation. The difference between the data and the model is called the residual [49]. For example, if we were to transmit or store the binary representation of the data in Figure 7, we would need to use 5 bits per sample. However, by exploiting the structure of the data, we can represent the sequence using fever bits. We see that the data appears to fall in a straight line [49]. A model for the data could be given by

$$x_n^* = n + 8, \; n = 1, 2, ..., 10 \tag{14}$$

To take advantage of this structure, examine the difference between the data and the model. The difference or residual is given by the sequence

$$e_n = x_n^* - x_n = (0, 1, 0, -1, 1, -1, 0, 1, -1, -1) \tag{15}$$

where x_n is the original data [49]. This sequence consists of only three numbers

$\{-1, 0, 1\}$. If we assign a code of 00 to -1, 01 to 0, and 10 to 1, we need to use 2 bits to represent each element of the residual sequence. Therefore, we obtain a compression when transmitting or storing the model parameters and the residual components. The encoding can be exact if the required compression must be lossless or approximate if it can be lossy [49, 50].

The type of structure or redundancy that existed in the previous data follows a simple law. Once we recognize this law, we can use the structure to predict the value of each element in the sequence and then encode the residue. The structure of this type is only one of many types [49, 50].

Finally, there will be situations where it is easier to take advantage of the structure if we decompose the data into a series of components. We can then study each component separately and use an appropriate model for that component. One way to perform this decomposition is to use different transforms, such as the Fourier and Wavelet transforms, which allow us to split the signal into some components. The more ways we have to view the information, the more successful we will be in developing compression schemes that take full advantage of the data structure [49, 50].

3 Related work

This section presents the main related work to this dissertation proposal. We present the research and applications of quantum wavelet transforms.

3.1 Quantum Wavelet Transforms

Research on Quantum Wavelet Transform (QWT) is still under progress and improvement. Preliminary work presentes a quantum version of wavelet transform for the Haar (H) and Daubechies four-order (D4) kernels. Hoyer [13] develops quantum networks for computing these two wavelets based on a generalized Kronecker product with a complexity of $O(n)$. Fijany [14] considers permutation matrices to derive efficient quantum circuits for the quantum Haar and Daubechies transform using packet algorithm (PAA) and pyramid algorithm (PYA). The proposed representation leads to an $O(n^2)$ complexity by using $O(n^2)$ gates. Besides, this paper shows that Hoyer's solution is incomplete and does not allow to obtain a correct complexity analysis. Klappenecker [15] implements periodized quantum wavelet packet transform (QWPT) and found that WPT is less expensive on a quantum computer. It requires $O(log^2(n))$ operations in contrast to classical WPT that needs $O(nlog(n))$ operations. Gosal and Lawton [51] construct quantum algorithms for Haar wavelet transforms and show its application to multi-scale decomposition of a dynamical system. Regarding multidimensional versions, Li et al. [10] propose multi-level and multi-dimensional (1D, 2D, 3D) quantum wavelet packet transforms (QWPT) based on the periodization extension, generalized tensor product, and permutation matrices, and their inverse transforms for the first time. The time complexities of the multi-level QWPT is $O(n)$, and the classical fast WPTs need 2^n basic operations on n elements. Li et al. [12] present the iteration equations of the general and inverse QWT using a generalized tensor product. The implementation shows an $O(n^3)$ complexity. Li et al. [11] construct the multi-level 2D-QWT's using the generalized tensor product and permutation matrices, offering exponential speedup over their classical counterparts. Also, they performed a quantum image compression application. So far, as far as we know, we have not found a definition of the QWT for different kernels nor a construction for the integer version of the wavelet transform.

Table 1 shows the main features of previous research and our proposal such as dimensionality, kernel implementation (Haar or Daubechies), primary construction tools or operations (tensor product, permutation matrices), time complexity, and whether they present the inverse version of the transform or not, simulations and direct applications.

3.2 Quantum Wavelet Transforms Applications

In addition, the development of the quantum wavelet transform has allowed some applications in different areas. Some researchers designed quantum watermarking schemes based on quantum wavelet transform and image formats (FQRI and NEQR), where the decomposition coefficients are extracted by executing QWT on a quantum image. These schemes outperform some classical watermarking proposals and performed simulations on Matlab [52–54]. Also, Li et al. [55] developed a lossy compression based on quantum wavelet transform and quantum Fourier transform with different formats. Wang et al. [56] proposed a quantum encryption scheme for quantum images based on QWT with FRQI and chaotic maps. Chakraborty et al. [21] designed image denoising based on QWT and FRQI with better visual quality over some classical methods.

Author	Dimension	Type	Kernels	Decomposition level	Basic operations	Time complexity	Inverse	Simula-tions	Applica-tion
Hoyer [13] 1997	1D	Real-valued	H, D4	—	Kronecker product, permutation matrices	$O(n)$ - Incorrect	No	No	No
Fijany [14] 1998	1D	Real-valued	H, D4	—	Permutation matrices	$O(n^2)$	No	No	No
Klappenecker [15] 1999	1D	Real-valued	—	—	Spliting matrices	$O(log^2(n))$	No	No	No
Gosal [51] 2001	1D	Real-valued	H	—	Permutation matrices	—	No	No	Dynamical analysis
Li et al. [10] 2018	1D, 2D, 3D	Real-valued	H, D4	Multi-level	Tensor product, permutation matrices	$O(n^3)$	Yes	Yes (Matlab)	No
Li et al. [12] 2019	1D	Real-valued	H, D4	Multi-level	Tensor product	$O(n^3)$	Yes	Yes (Matlab)	No
Li et al. [11] 2021	2D	Real-valued	H, D4	Multi-level	Tensor product, permutation matrices	$O(n^3)$	Yes	Yes (Matlab)	Image compre-ssion
Proposal	**1D**	**Integer**	**H, D4, CDF**	**Multi-level**	**—**	**Time Speedup**	**Yes**	**Yes (Qiskit)**	**Lossless compre-ssion**

Table 1: Related work on the Quantum Wavelet Transform.

4 Research Proposal

4.1 Motivation and Justification

The Wavelet transform is an essential classical tool in signal processing, numerical analysis, and information hiding. This transform enables analyzing stationary and non-stationary signals. It provides a way for multiresolution analysis and local signal features extraction. The Wavelet transform, unlike other transforms such as the Fourier transform, preserves the signal frequency and time information. These characteristics make the Wavelet transform useful for many applications such as compression, segmentation, enhancement, pattern recognition, denoising, feature extraction, watermarking, and cryptography [4, 16, 44, 57]. However, conventional (real-value) wavelet transforms are usually lossy and computationally complex. For example, in compression applications, 60% of the time is consumed by the Wavelet transform [19]. Also, the limited computer precision prevents its use in practical lossless applications, where the recovery and preservation of the original data are crucial [16, 17, 44].

Wavelet transforms are widely used in different fields due to their unique characteristics and advantages over traditional (Fourier basis) transforms. Their characteristics suggest that the development of the quantum formalism for the wavelet transform algorithm could play a significant role in the areas of quantum computing and quantum information processing. So far, research on the quantum wavelet transform (QWT) is still preliminary, but there are significant advances [10–14, 31].

QWT has helped to develop more sophisticated and efficient quantum algorithms for different applications and has allowed the construction of complex operations. Also, it has shown an exponential speedup compared to classical transforms and deals with some issues in quantum information processing as signal decomposition, compression, watermarking, and cryptography [4, 10–14].

Research on quantum wavelet transforms focuses on the definition of the real-value transforms for two representative wavelet kernels, the Haar and Daubechies bases. Different researchers propose and implement one-, two- and three-dimensional quantum wavelets at the gate-level circuit. These approaches generate the decomposition coefficients according to a packet or pyramid algorithm, and they are based on the factorization of unitary operators by permutation matrices [10–14].

Despite the advantages of the QWT over its classical counterpart, this transform is difficult to exploit because its use is limited to a subset of quantum representation formats. The internal fractional representation of the data is not considered in the quantum decomposition process, while these issues are rarely discussed. The number of applications and analysis capabilities are small, since there are only two implementations of the wavelet kernels. Also, like the classical wavelet transform, the real-valued QWT is not suitable for lossless applications such as compression and data hiding [4, 12].

The main challenge in developing a quantum formalism of the wavelet transform is to define a set of unitary operators to implement it. This is a consequence of many classical operators not being originally described in terms of unitary operators. Thus, the transition between the classical and the quantum world is not straightforward [6, 13, 14, 27–29]. Also, to apply QWTs to quantum information to obtain the decomposition result, the information signal must be encoded in a quantum representation format that allows the transform to be used efficiently and correctly. This issue is rarely addressed [6, 27].

In practice, finite precision is employed to represent the decomposition coefficients. However, this representation requires rounding operations to truncate the floating-point results, which is inherently inaccurate, and the invertibility of the transform depends on using exact arithmetic. If the transform is not invertible, then information will be lost [16, 18, 58, 59]. Therefore, to allow lossless reconstruction of the original data, it is necessary to construct an invertible transform in finite-precision, that is, an integer-to-integer transform (ITI), which is based on the lifting scheme [4, 17, 19, 44, 47, 57].

ITI transforms are especially useful in medical, military, banking, and space sciences applications, where minimal perturbations of the primary data are not acceptable [16, 18, 44]. Also, ITI transforms require less storage capacity to store the coefficients and reduce the computational cost by half over conventional transforms [16–19]. Nevertheless, ITI transforms have some limitations related to low compression rates and degradation in lossy applications [18]. They only allow efficient lossless construction for a few kernel wavelets. Finally, the factorization of the operators of the lifting scheme has an impact on the performance of the transform and makes the transition from the standard real-valued wavelet to ITI transforms not straightforward [4, 16–19, 44, 57, 58].

To the best of our knowledge, there is no definition of the QWT for different kernels, nor a quantum construction for the ITI version of the wavelet transform. Quantum ITI wavelet transforms (Q-IWTs) could help to build efficient lossless applications, such as data compression and data hiding. It could decrease the computational cost over the classical counterpart and enhance the performance in various applications.

Therefore, given the implications and limitations of classical and quantum wavelet transforms, this research focuses on developing a quantum approach to the one-dimensional integer wavelet transform. This development will be based on the factorization of unitary operators for a subset of wavelet kernels, mainly the Haar, Daubechies-4, and CDF bases. The transform will be performed according to the lifting scheme. We will analyze and select a suitable quantum signal representation to improve the signal decomposition and manipulate the internal fractional data in the process. Finally, we aim to design two algorithms, one for quantum signal decomposition and the other for quantum lossless compression, both based on the proposed transform.

4.2 Problem Statement

Quantum computing research focuses on the development of quantum tools and algorithms to solve problems in different areas. However, it lacks knowledge on how to develop quantum versions of integer-to-integer transforms. Thus, to extend the knowledge and provide a broader set of quantum tools, it is necessary to investigate the definition of quantum integer transforms, such as the quantum integer wavelet transform. Therefore, this research addresses the problem of developing a quantum version of the one-dimensional integer wavelet transform for different kernels and design quantum algorithms for one-dimensional signal decomposition and lossless compression. 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 algorithm to signal decomposition and lossless compression.
- Improve the performance of the integer wavelet transforms based on the development of a quantum lifting scheme.

4.2.1 Problem Definition

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

Classical WT

We can define the classical WT as

$$W_m^{(l)} X^T \to (A_1^0,\ D_1^0,\ D_1^1,\ D_2^1, \ldots,\ D_{k-1}^i,\ D_k^i\ ) = (A,\ D) \tag{16}$$

where $X = (x_1,\ x_2,\ \ldots,\ x_m)$, $W_m^{(l)}$ is the matrix form of the WT for a signal of length $m = 2^n$ with decomposition level (l), $A = A_1^0$ is the approximation coefficient at the zero level, and D_k^i is the k-th detail coefficient at the level (i), that is,

$$D = \sum_{i=0}^{l-1} \sum_{k=1}^{2^i} D_k^i \tag{17}$$

where l is the maximum decomposition level.

Quantum WT

Based on (16) and (17), we define a quantum representation for the integer wavelet transform by

$$U_{W_m}^l \left|X\right\rangle \to \left|A^0\right\rangle \otimes \sum_{i=0}^{l-1} \left|D^i\right\rangle = \left|A, D\right\rangle \tag{18}$$

where $|X\rangle$ is the signal vector encoded in a quantum format, $U_{W_m}^l$ is the unitary operator for the quantum WT for a signal of length $m = 2^n$ with decomposition level (l), $|A^0\rangle = |A\rangle$ is the approximation coefficient at the zero level, and $|D\rangle = |D^i\rangle$ is the detail coefficient at the level (i) given by,

$$\left|D^i\right\rangle = \sum_{k=1}^{2^i} \left|d_k^i\right\rangle \tag{19}$$

where $|d_k^i\rangle$ is the k-th detail coefficient at the level (i).

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 such that efficiently perform the QWT for Haar, Daubechies-4 and CDF kernels.

$$U_{W_m} = (U_0 \circ U_1 \circ \ldots \circ U_{n-1}) \tag{20}$$

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

4.3 Research Questions

The main questions that guide this research are:

1. Which unitary operators allow 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?

4.4 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 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 to design a lossless compression algorithm based on the proposed quantum transform.

4.5 General Objective

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

4.5.1 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 improve signal decomposition.
3. To develop a quantum algorithm to 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.

4.6 Methodology

We propose the following methodology to achieve the objectives and answer the research questions.

1. ***Select quantum elements and operators.*** It is the first stage to define a quantum approach of the integer wavelet transform. We need to consider and describe the following components.

   - Define a quantum version of the classical lifting scheme, it is, to represent the involved operations, operators, and processes using quantum formalism.
   - Find an efficient and suitable factorization for the operators involved in the quantum integer wavelet transform.
   - Identify the factorization matrices that characterize some quantum wavelet kernels, mainly the Haar, Daubechies-4, and CDF bases.

2. ***Select a quantum representation format.*** This stage allows us to find an appropriate quantum signal representation to apply the quantum integer wavelet transform. In this stage, we analyze the following elements.

   - Compare a set of the existing quantum representation formats for a one-dimensional signal.
   - Select a suitable quantum format to represent a one-dimensional signal to improve the signal decomposition results and compression.

3. ***Design a quantum algorithm for signal decomposition.*** To develop this algorithm, we adress the following issues.

   - Encode the signal information into a selected quantum format to allows perform the specific operations.
   - Apply the quantum integer wavelet transform to decompose the signal and extract the approximation and detail coefficients.
   - Store the decomposition components using the selected quantum format to signal analysis.

4. ***Design a quantum lossless compression algorithm.*** To design this quantum algorithm, we need to address the following issues.

   - Determine an acceptable decomposition level of the signal using the quantum integer wavelet transform.

- Modify some decomposition coefficients to generate similarities between components.
- Select the representative coefficients of decomposition to eliminate redundancy.
- Choose an efficient quantum coding method to store the resulting data.

5. ***Evaluate and analyze.*** To evaluate the results and performance of this proposal, we use the next set of experiments and metrics.

    - Mathematical development to guarantee the quantum definition of the integer wavelet transform.
    - Time complexity and quantum complexity analysis of the proposed transform.
    - Use a quantum simulation environment to analyze the performance of the quantum integer wavelet transform for different kernels.
    - To select a quantum format, we will compare different quantum representations across some metrics such as the decrease in the number of qubits to store a signal, the method of information storage, the flexibility and ease of applying the quantum integer wavelet transform, the performance of signal recovery and noise addition, and the difficulty of implementation.
    - To evaluate the lossless compression algorithm, we will use different parameters such as compression rate, maximum average compression, signal distortion, time complexity, signal recovery, first order entropy, and noise addition.

4.7 Scope and Limitations

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 as the main application.
- A Simulation environment are used as a means of verification.
- Noise environments are out of the scope of this research.

4.8 Expected Contributions

The main expected 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, Daubechies-4, and CDF).
2. A quantum integer wavelet transform with an improvement in computational cost over the classical counterpart.
3. A quantum algorithm to one-dimensional signal decomposition using quantum integer wavelet transform.
4. A quantum lossless compression algorithm for one-dimensional signals based on the proposed transform.

4.9 Work Plan

Figure 8 shows the Gantt chart for the activities schedule to carry out during this research project.

4.9.1 Publication Plan

1. Quantum S transform (First integer wavelet transform).
2. Survey of quantum computing (implementation, algorithms, applications).
3. Quantum algorithm for the integer wavelet decomposition.
4. Application of the quantum integer wavelet transform (quantum lossless compression).

4.9.2 Target Journals and Conferences

Journals: Nature, Information Sciences, IEEE Transactions on Cybernetics, Quantum Information Processing, SPIE.

Conferences: Quantum Information Processing (QIP), International Conference on Quantum Communication, Measurement and Computing.

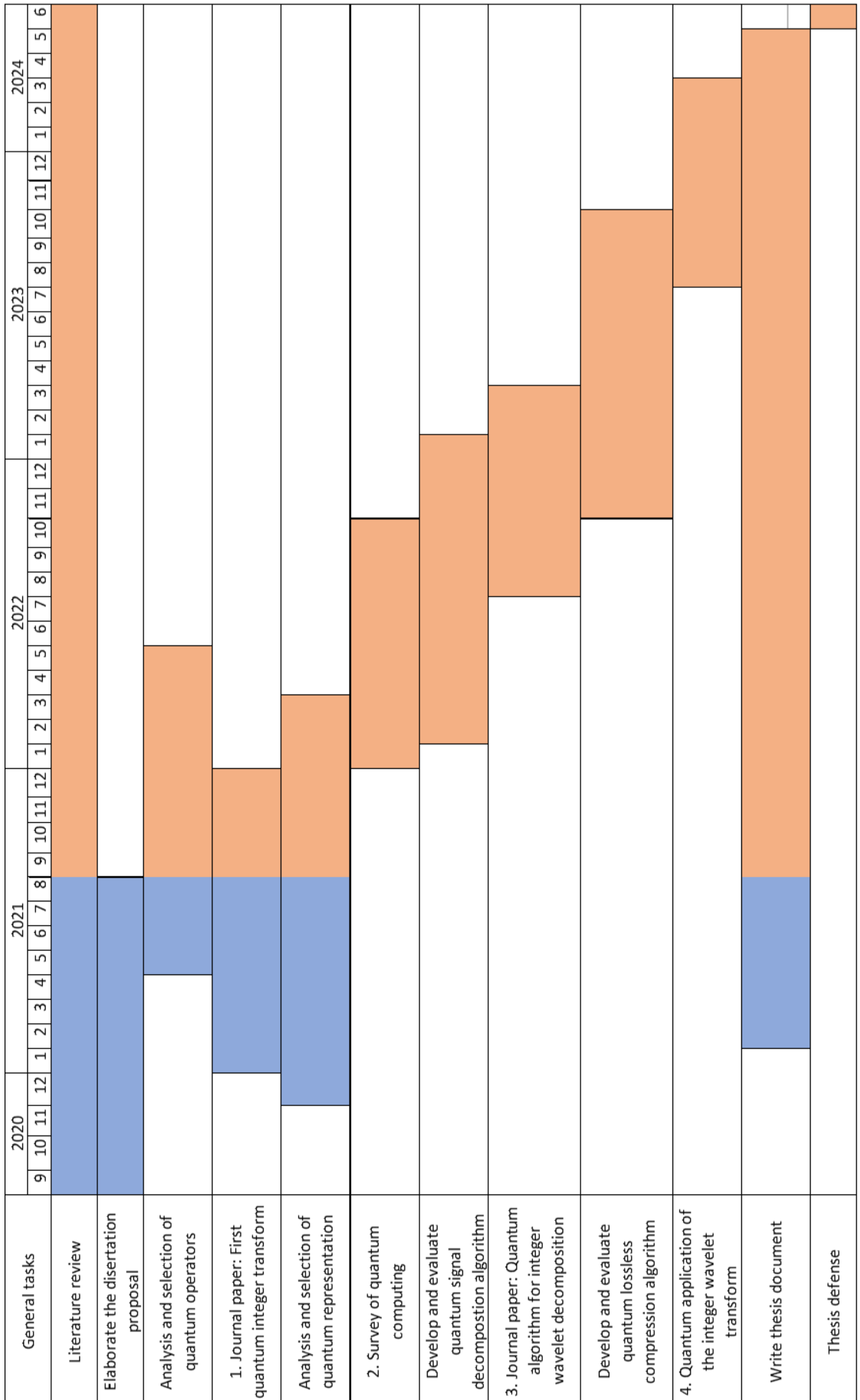

Figure 8: Activities schedule, in which the blue and orange segments represent finished and unfinished activities, respectively.

5 Preliminary Results

Preliminary results focused on gaining knowledge about wavelet transforms and quantum simulation environments. We developed and implemented the first quantum model of a classical integer-to-integer wavelet transform called S transform. Also, we proposed a quantum representation format, and developed a quantum rouding operator. In each experiment, we use the Quantum Experience simulation environment by IBM [60].

- **Quantum S transform.** The development of this wavelet transform-based integer transform approach allowed us to analyze the quantum design procedure, study the circuit-gate implementation and propose our first quantum algorithm. In the following sections, we provide a classical description of the S transform and our quantum construction.

5.1 Classical S-Transform

The S transform was the first approach to an integer-to-integer wavelet transform. It allows signal decomposition through the Haar kernel. The idea is to divide the signal into two no overlapping subsets, odd and even samples. Then, we generate the decomposition values by performing some operations over the signal components called prediction and update steps. The goal of these steps is to exploit some characteristics and preserve the internal structure of the signal. This transform supposes a correlation between close samples to predict the next value and the Haar kernel allows us to keep the mean value of the signal information [61, 62]. Figure 9 shows the prediction and update scheme.

The equations describing the steps in Figure 9 are given by

$$\begin{aligned} D &= S_{odd} - P(S_{even}) \\ A &= S_{even} + U(D) \end{aligned} \tag{21}$$

where $P(.)$ and $U(.)$ are the Prediction and Update operators, respectively [61, 62]. Then, we achieve the expected properties using the following Prediction and Update

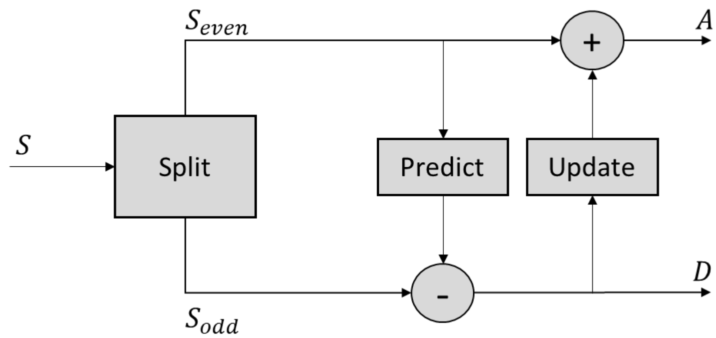

Figure 9: Prediction and update scheme, where A and D are the approximation and detail coefficients respectively.

operators [61, 62].

$$\begin{aligned} P(S_{even}) &= S(2n) \\ U(D) &= \frac{D}{2} \end{aligned} \tag{22}$$

Then,

$$\begin{aligned} D &= S(2n+1) - S(2n) \\ A &= \left\lfloor S(2n) + \frac{D}{2} \right\rfloor = \left\lfloor \frac{S(2n) + S(2n+1)}{2} \right\rfloor \end{aligned} \tag{23}$$

The floor function is to guarantee the integer-to-integer version of the transformation [61, 62].

5.2 Quantum S-Transform

Based on the above ideas, we define a suitable way to develop a quantum S-transform approach. Therefore, we identify the following components to represent this transformation.

- **Encode the information.** We analyze quantum formats for representing signal information. In this case, we focus on the basis state representation given by NEQR.

- **Perform the operations.** We describe the quantum version of the operators involved in the classical S-transform and exploit the quantum features to perform the decomposition.

- **Store the decomposition values.** The values are stored using the initial representation to generate the new components.

5.2.1 Encode the Information

We use a representation model based on basis states given by NEQR image format in contrast to quantum real-valued wavelet transforms that use a storage format over the amplitude coefficients of the quantum states as FRQI [25, 27]. The NEQR enables us to store the signal information in the following quantum superposition representation [24, 27].

$$|S\rangle = \frac{1}{2^n} \sum_{j=0}^{2^n-1} |f(x_j)\rangle |x_j\rangle \tag{24}$$

where $|f(x_j)\rangle$ is the binary representation of the signal components and $|x_j\rangle$ the corresponding position.

However, to apply the S-transform using this representation, we need to extract and store on new states the necessary components. For example, if we want to apply the transform, we need to take the values $|f(x_o)\rangle$ and $|f(x_1)\rangle$, store them in new states, and then perform operations on the new basis-states. Then, we move on to the next pair of elements, $|f(x_2)\rangle$ and $|f(x_3)\rangle$, and so on. This set of additional steps increase the number of operators and manipulations required to achieve the expected results, which increases the complexity of the quantum circuit [27, 63–66]. Therefore, to overcome these issues, we propose a quantum representation based on NEQR.

5.2.2 New Quantum Representation

The proposed format encodes the information of the signal on a superposition of basis state in the following way. Given a signal of 2^n elements, we store the information of two adjacent elements in the same position as,

$$|S\rangle = |f(x_0)\, x_0 f(x_1)\rangle + |f(x_2)\, x_1 f(x_3)\rangle + \ldots + |f(x_{m-1})\, x_n f(x_m)\rangle, \quad m = 2^n - 1$$
$$|S\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{i=0}^{2^{n-1}-1} |f(x_{2i})\, x_i f(x_{2i+1})\rangle \tag{25}$$

where $|f(x_{2i})\rangle$ and $|f(x_{2i+1})\rangle$ are the binary representation of the elements at the even and odd position repectively, and x_i is the new storage position of the signal components.

This new representation allows us to operate in a simple way on two adjacent components, given by the odd and even elements of the signal. Thus, we reduce the number of additional operators and manipulations required to extract the components by the conventional NEQR format [27, 63–66].

5.2.3 Quantum Operations

Given the format to encode the information, we describe a subset of quantum operations to achieve the decomposition results in the transform. This subset consists of quantum addition, subtraction, halving operation, and rounding function.

- **Addition.** We perform the addition of two quantum registers $|a\rangle$, and $|b\rangle$, without loss of information and in a reversible way as:

$$U_{add}|a, b, 0\rangle \rightarrow |a, b, a+b\rangle \tag{26}$$

  where $|a\rangle$ and $|b\rangle$ are encoded on n qubits and the register $|a+b\rangle$ on $n+1$ qubits to store the carry [67]. U_{add} is the addition operator. Figure 10 illustrates the general representation, and the circuit-gate scheme for two qubits. Figure 11 shows a circuit-gate implementation on Qiskit for $n = 1$, $|a\rangle = |a_0\rangle$, $|b\rangle = |b_0\rangle$, and $|a+b\rangle = |carry,\ add\rangle$. Also, it presents the probability distribution results when $|a\rangle = |1\rangle$ and $|b\rangle = |1\rangle$.

- **Subtraction.** The quantum reversible subtractor of two registers, $|a\rangle$ and $|b\rangle$, is given by:

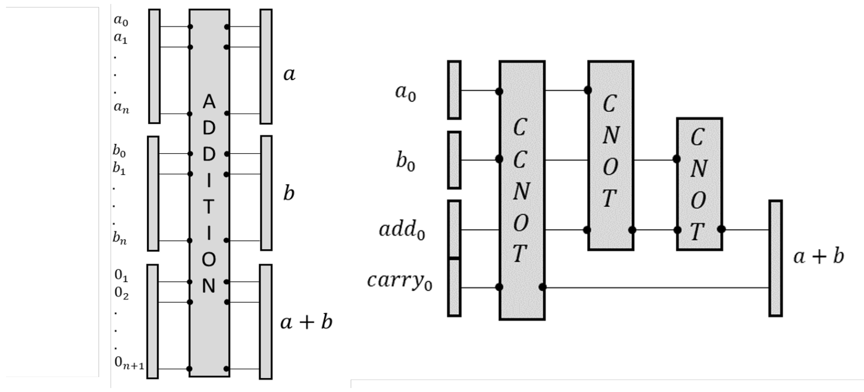

Figure 10: Simplified general representation and circuit-gate implementation of the addition operation. CCNOT is the controled-controled-NOT gate, and CNOT the controled-NOT gate.

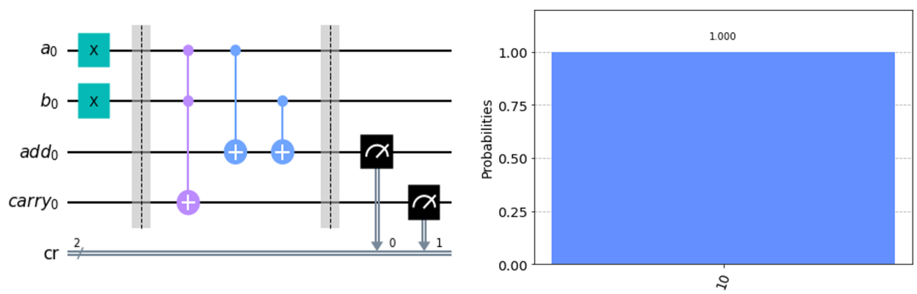

Figure 11: Circuit-gate implementation on Qiskit and probability distribution of performing operation $|a + b\rangle$.

$$U_{sub}\,|a, b\rangle \rightarrow |\, a - b,\ b\rangle \tag{27}$$

where $|a\rangle$ and $|b\rangle$ are encoded on n qubits and the register $|a - b\rangle$ on $n + 1$ qubits to store the borrow bit [67, 68]. U_{sub} is the subtraction operator. If we want to implement the quantum circuit-gate to this operation, we need to use the U_c, U_s, and $U_c^\dagger$ operators [67, 68]. Figure 12 depicts the general representation, and a circuit-gate scheme for four qubits. Figure 13 presents a circuit-gate implementation on Qiskit for $n = 2$, $|a\rangle = |a_1 a_0\rangle, |b\rangle = |b_1 b_0\rangle$, and $|a - b\rangle = |borrow,\ a_1, a_0\rangle$. Also, it presents the probability distribution results when $|a\rangle = |11\rangle$ and $|b\rangle = |01\rangle$.

- **Halving operation.** The halving operation halves the register value by one

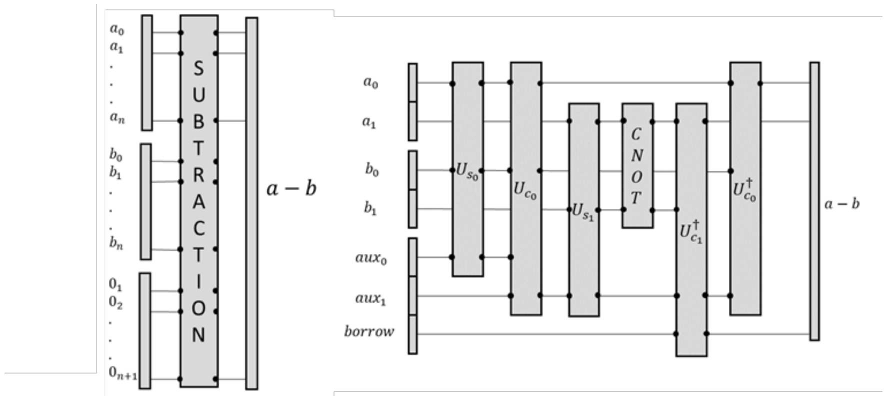

Figure 12: Simplified representation and circuit-gate scheme of the subtraction operation. CNOT is the controlled-NOT gate.

cycle shift downwards given by:

$$U_H|a_n \dots a_1 a_0\rangle \to |a_0, a_n \dots a_1\rangle \tag{28}$$

This operation does not require the use of additional qubits [69]. U_H is the halving operator. Figure 6 shows the blocks scheme and the circuit-gate representation for a four-element register. Figure 7 depicts the Qiskit implementation for $n = 4$, and $|a\rangle = |a_3 a_2 a_1 a_0\rangle$, the probability distribution results for $|a\rangle = |1010\rangle$ is also shown.

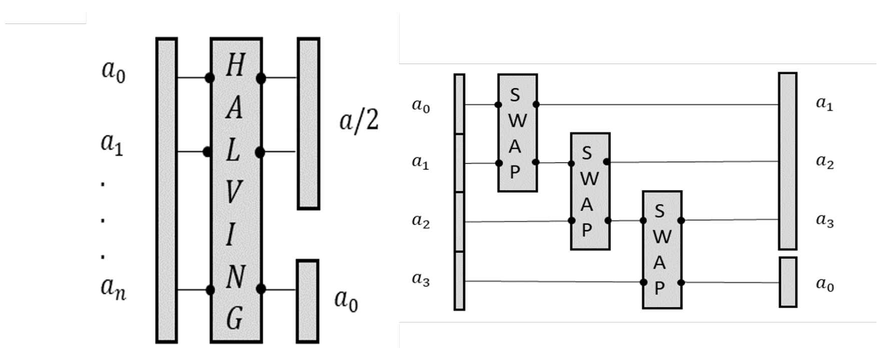

Figure 14: Blocks scheme and circuit-gate representation of halving operation. SWAP is the swap quantum gate.

- **Rounding operation.** The S-transform uses a rounding operation to give an integer-to-integer map between the signal elements and the decomposition results. This operation could be problematic in the quantum domain due to its

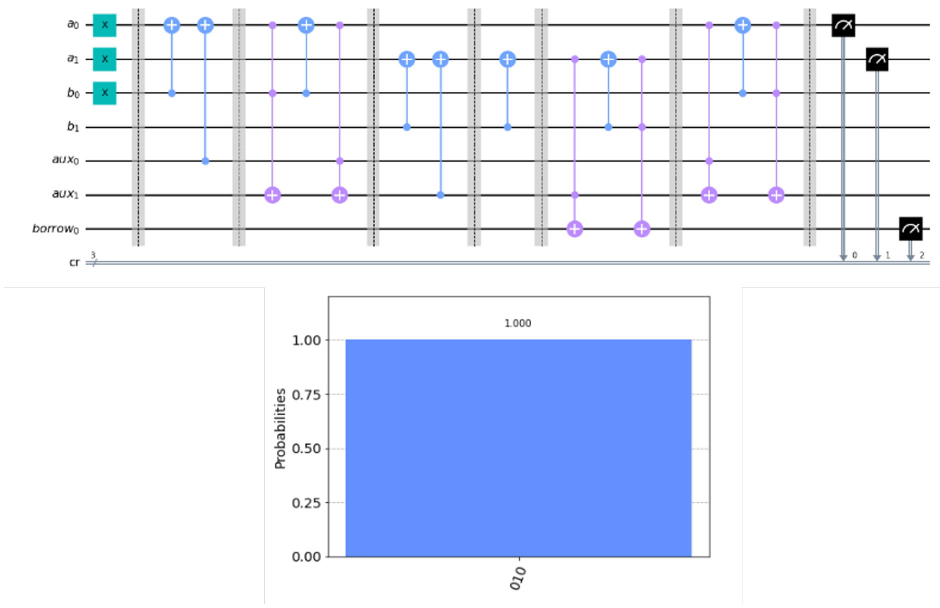

Figure 13: Circuit-gate implementation on Qiskit and probability distribution of the result, $|a - b\rangle$.

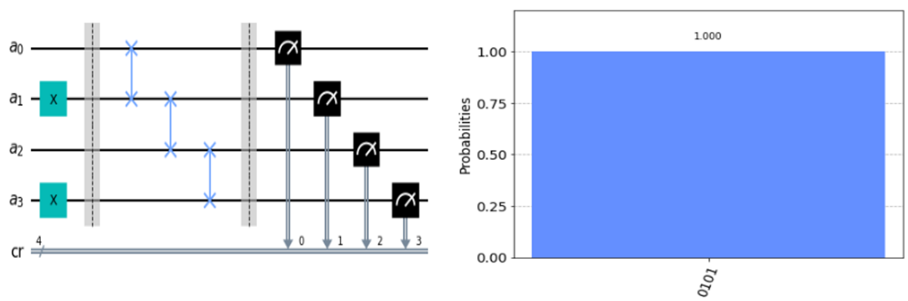

Figure 15: Qiskit implementation and probability results for the halving operation.

nonlinearity. However, we can split the rouding operation in the S-transform as follows. Given a integer value a,

- If a is even $\rightarrow a_{even} = 2m_1$.
  Then,

$$\left\lfloor \frac{a_{even}}{2} \right\rfloor = \left\lfloor \frac{2m_1}{2} \right\rfloor = \lfloor m_1 \rfloor = m_1 \tag{29}$$

- If a is odd $\rightarrow a_{odd} = 2m_2 + 1$.

Then,

$$\left\lfloor \frac{a_{odd}}{2} \right\rfloor = \left\lfloor \frac{2m_2+1}{2} \right\rfloor = \left\lfloor \frac{2m_2}{2} + \frac{1}{2} \right\rfloor = \left\lfloor m_2 + \frac{1}{2} \right\rfloor \\ = \lfloor m_2 \rfloor = m_2 \tag{30}$$

where m_1 and m_2 are integer values. From (29) and (30) we observe that the total contribution of the rounding operation in the S-transform is given by the integer even component. Thus, we can subtract 1 from the odd case, a_{odd}, without disturbing the result of the rounding operation.

$$\left\lfloor \frac{a_{odd}-1}{2} \right\rfloor = \left\lfloor m_2 + \frac{1}{2} - \frac{1}{2} \right\rfloor = \lfloor m_2 \rfloor = m_2 \tag{31}$$

The operations in (29) and (31) guarantee that we always perform the rounding operation on integer values. Therefore, since the rounding operation is linear over the integers, we can define the following quantum operator,

$$U_R \left|a\right\rangle \rightarrow \left|\lfloor a \rfloor\right\rangle \leftrightarrow a \in \mathbb{Z} \tag{32}$$

where U_R is the rounding operator. Figure 16 shows the circuit-gate representation of this process. The COM block allows us to select between the odd and even results, HAL and SUB are the previous halving and subtraction operators, respectively. The Qiskit implementation is given later in the full design.

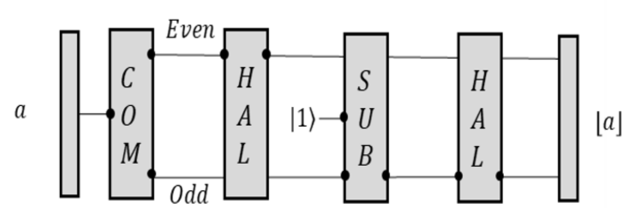

Figure 16: Circuit-gate representation of the quantum rounding operation.

5.3 Proposed Design

Firstly, we prepare and transform a signal of 2^n elements based on the proposed representation model 5.2.2, as equation (25). It provides an easy way to split the signal into blocks of two adjacent elements, odd and even components, which reduces the number of additional operations and extra qubits. Figure 17 shows an example of a quantum signal with $n = 2$ and its representation in the proposed model.

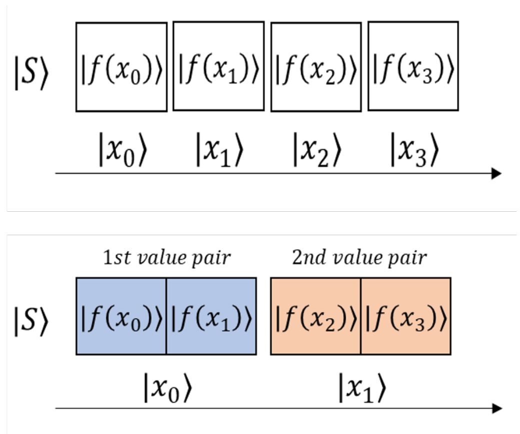

Figure 17: The top figure shows the signal components, and the bottom figure shows the quantum representation in the proposed model, where the signal elements are grouped in pairs.

Next, we perform different operations on each block according to the above set of operators to obtain the decomposition results given by the quantum S-transform. First, we use the adder operator to get the Sum of the components in each block, algorithm 1 on line 6. Second, we identify the odd and even results of the above operation. If the Sum is odd, we subtract the value of 1 from this result, $U_{Sub(1)}$. Otherwise, we do not change the value, algorithm 1, line 12. Then, we perform the halving operation through the U_H operator, algorithm 1 on line 20. Finally, we use the rounding operator, U_R, to obtain the first decomposition elements $|A\rangle$, algorithm 1 on line 24. The previous process is described in Algorithm 1, where the notation is simplified to illustrate the general procedure.

The Algorithm 1 allows us to compute the approximation components, $|A\rangle$, in the quantum S-transform. Now, we find the detail components, $|D\rangle$, using the subtractor operator, U_{Sub}, on the signal $|S\rangle$, see Algorithm 2.

Algorithm 1 : Quantum S Transform - Approximation Coefficients

▷ $|\boldsymbol{S}\rangle$: Signal
▷ $|\boldsymbol{x_i}\rangle$: Element position
▷ $|\boldsymbol{f(x_i)}\rangle$: Signal value
▷ $\boldsymbol{m}$: integer value

▷ **Addition operation** $(|S\rangle)$
for $i = 0$ and $n/2$ **do**
  $U_{add}\,|S\rangle = U_{add}|\,f\,(x_{2i})\,x_i f\,(x_{2i+1})\rangle \rightarrow |f\,(x_{2i}) + f\,(x_{2i+1})\rangle = |Sum\rangle$
end for
▷ **end Addition**

▷ **Odd and Even Elements**$(|Sum\rangle)$
if $Sum = 2m + 1$ **then**
  $U_{sub(1)}|Sum\rangle \rightarrow |Sum - 1\rangle = |Sum\rangle$
else
  $|Sum\rangle = |Sum\rangle$
end if
▷ **end Odd and Even**

▷ **Halving Operation**$(|Sum\rangle)$
$U_H|Sum\rangle \rightarrow \left|\frac{Sum}{2}\right\rangle$
▷ **end Halving**

▷ **Rounding Operation**$(|\frac{Sum}{2}\rangle)$
$U_R|\frac{Sum}{2}\rangle \rightarrow |\lfloor\frac{Sum}{2}\rfloor\rangle = |A\rangle$
▷ **end Rouding**

Algorithm 2 : Quantum S Transform - Detail Coefficients

▷ $|\boldsymbol{S}\rangle$: Signal
▷ $|\boldsymbol{x_i}\rangle$: Element position
▷ $|\boldsymbol{f(x_i)}\rangle$: Signal value

▷ **Subtraction Operation** $(|S\rangle)$
for $i = 0$ and $n/2$ **do**
  $U_{sub}\,|S\rangle = U_{sub}|\,f\,(x_{2i})\,x_i f\,(x_{2i+1})\rangle \rightarrow |f\,(x_{2i+1}) - f\,(x_{2i})\rangle = |D\rangle$
end for
▷ **end Subtraction**

5.4 Circuit-Gate Realization

We present the integrated quantum circuit realization of the S-transform based on the circuit-gate modules of each quantum operation, including addition, subtraction, halving, and rounding. Figure 18 illustrates the complete circuit design of the quantum S-transform.

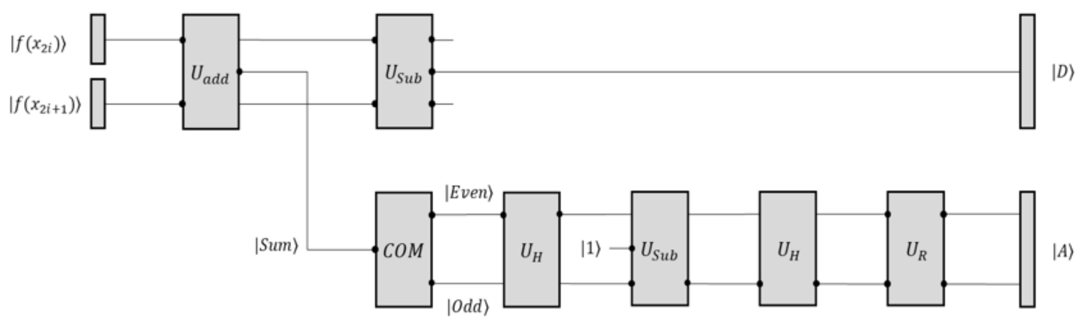

Figure 18: Complete circuit-gate design of the quantum S-transform.

The complete circuit represents only one level of decomposition, where the decomposition components are the detail and approximation coefficients, $|D\rangle$ and $|A\rangle$, respectively. If we want to reach more decomposition levels, we take the approximation coefficients and apply the transform on them.

5.5 Simulation Experiments

We performed simulation experiments of the proposed transformation using IBM's Qiskit simulation environment on a classical computer. We perform the one-level quantum S-transform on a signal of 2^2 elements as a case study to show the implementation and feasibility of the gate-circuit development.

1. **Quantum representation.** First, we encode the signal information in the proposed quantum representation format from defition 5.2.2. The signal components are $|f(x_0)\rangle = |101\rangle$, $|f(x_1)\rangle = |111\rangle$, $|f(x_2)\rangle = |111\rangle$, and $|f(x_3)\rangle = |111\rangle$. Figure 19 gives the signal values in the quantum representation. Figure 20 shows the quantum circuit representation and the probability results.

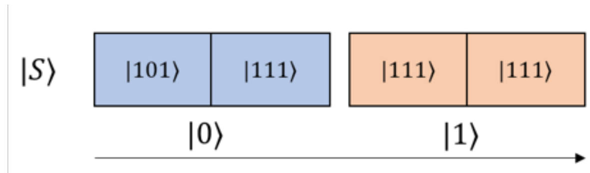

Figure 19: Quantum representation of the signal.

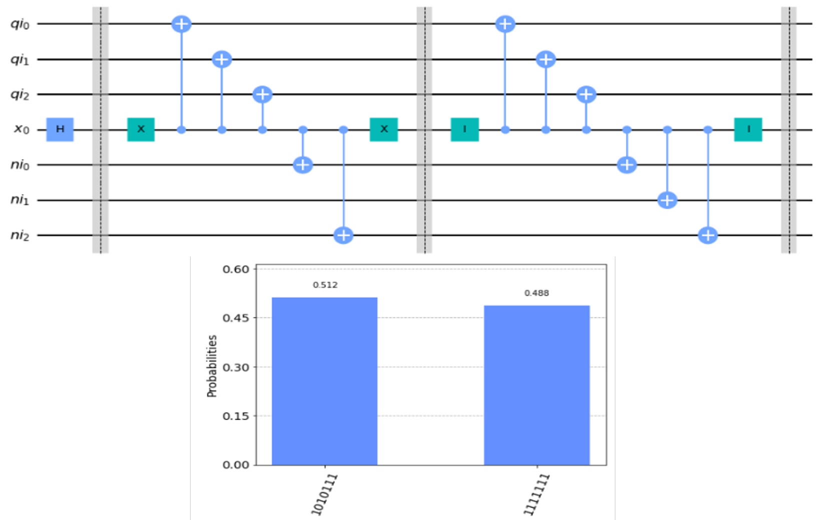

Figure 20: Circuit-gate implementation of the proposed quantum representation, and probability distribution results.

The quantum register $|q_{i2}q_{i1}q_{i0}\rangle\,|x_0\rangle\,|n_{i2}n_{i1}n_{i0}\rangle$ describes the signal components, $|101, 0, 111\rangle$ and $|111, 1, 111\rangle$, where the middle value is the respective position. Figure 20 gives the Qiskit implementation, and the probability distribution shows the superposed values in the quantum register and the probability of obtaining each of these values after a measurement.

2. **Quantum decomposition: Approximation coefficients.**We obtain the approximation coefficients, $|A\rangle$, of the signal using the addition, halving, and rounding operators. Therefore, we get a state in a superposition of approximation values as follows:

$$|A\rangle = \alpha\,|0110\rangle + \beta|0111\rangle$$

where α and β are the probability amplitudes. Figure 21 gives the circuit-gate implementation on Qiskit and the probability distribution of values.

Figure 21 shows a block implementation of the quantum decomposition, where each block contains the circuit-gate representation of the operators. We per-

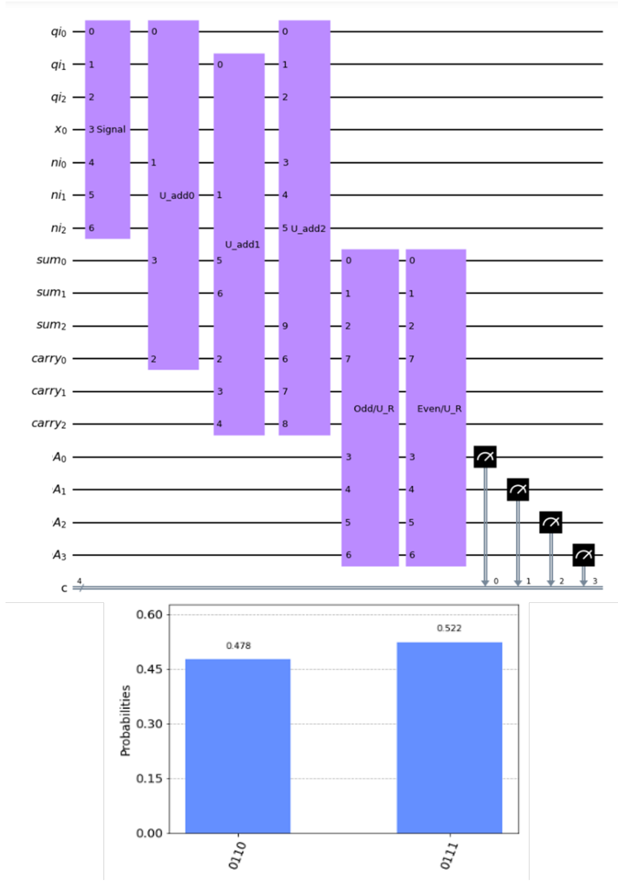

Figure 21: Qiskit implementation of the quantum decomposition to get the approximation coefficients, and below the possible values for these coefficients.

form the addition operation using the U_{add} block circuit and the $even/odd$ selection, halving, subtraction, and rounding operations using the Odd/U_R and $Even/U_R$ blocks. The probability figure gives the possible values of the approximation's coefficients, $|A\rangle$.

3. **Quantum decomposition: Details coefficients.** We obtain the details coefficients, $|D\rangle$, using the subtractor operator between the adjacent components of the signal. Therefore, we get a state in a superposition of details values as follows:

$$|D\rangle = \alpha\,|0010\rangle + \beta|0000\rangle$$

where α and β are the probability amplitudes. Figure 22 gives the circuit-gate implementation on Qiskit and the probability distribution of values.

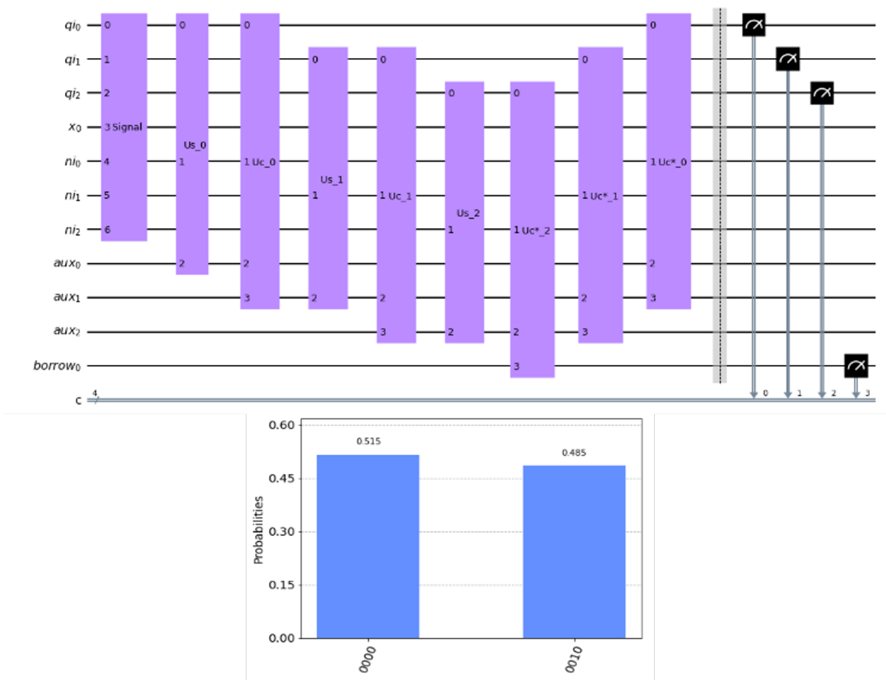

Figure 22: Qiskit implementation of the quantum decomposition to get the detail coefficients, and below the possible values for these coefficients.

Figure 22 shows the block implementation of the operators using the subtraction operation between the signal components. The probability figure gives the possible values of the detail's coefficients, $|D\rangle = |borrow_0, q_{i2}, q_{i1}, q_{i0}\rangle$.

5.6 Results Discussion

Preliminary results allowed us to get experience in quantum algorithm development, including their mathematical definition, the design of the operations involved, and their implementation. In addition, we observed the limitations and constraints given by the quantum simulator, such as available resources, topology graphs constraints, and access to a real-quantum computer. These considerations allowed us to know the maximum quantum simulation capacity of a classical computer, $30 - qubits$. We also identified the structural interconnection and interaction constraints between qubits in the IBM quantum computer. Finally, we accessed to the real IBM quantum computer of $5 - qubit$ and $15 - qubit$, where we implemented and observed the results of different quantum algorithms. However, we could not implement the S-transform quantum approach on the **real-quantum** computer because we required more than $15 - qubits$, and the access to the $15 - qubit$ computer is currently restricted.

We analyzed quantum representation formats, mainly quantum representation over basis states and amplitude coefficients such as NEQR and FRQI, respectively. We studied the properties and limitations of these formats as enabled operations, information extraction, and quantum complexity. Finally, we selected a basis state representation as NEQR, but with a proposed modification. This modification allowed us to directly perform the operations involved in the quantum S-transform, reduce the number of additional quantum operations and extra qubits required by traditional NEQR, and kept the quantum complexity in $O(n)$.

Also, we proposed the first integer-to-integer quantum transformation approach, called the quantum S-transform, which uses the proposed basis states representation, the Haar wavelet, and the quantum addition, subtraction, halving, and rounding operators. We gave the quantum description, quantum algorithm, and circuit-gate implementation on Qiskit of the proposed S-transform. This development allowed us to study the first version of the lifting scheme used in the integer wavelet transform, know some of the representation models useful for the final proposal, and construct different quantum elements.

Finally, the quantum S-transform approach represents our first journal paper in the field of quantum computing. So far, we have written 60% of the content of the article and plan to submit it during the remainder of this year. We will consider the journals in the section 4.9.2 as publication targets.

6 Final Remarks

In this preliminary research, we presented the description of the PhD proposal, which is in the field of quantum computing. We proposed a quantum approach to the one-dimensional integer wavelet transform based on lifting scheme and quantum algorithms for signal decomposition and lossless compression as the main application.

The key ideas are to define a quantum format for signal representation, analyzing and selecting a suitable representation using existing formats. Then, develop a quantum lifting scheme by defining the quantum version of the classical operators involved. Also, construct the unitary operators for the Haar, Daubechies-4 and CDF kernels. This development is based on the factorization of the unitary operators into smaller unitary matrices of efficient implementation. Finally, we will design the circuits for the multi-level quantum wavelet kernels and use Qiskit simulation enviroment by IBM as means of performance verification.

Preliminary results enabled significant advances in this research. First, we defined a general quantum formalization for the integer wavelet transform (sec. 4.2.1). In addition, we proposed a quantum representation format based on the NEQR (sec. 5.2.2). This new representation allowed us to efficiently manipulate the signal elements involved in the quantum S-transform, reducing the extra qubits and additional operations used in conventional NEQR. Also, we kept the computational complexity to $O(n)$. Furthermore, we defined a quantum version of the rounding operation involved in the classical S-transform, that is, we defined a unitary form of the rounding operation avoiding nonlinearities (sec. 5.2.3).

Finally, we developed the first approach to the quantum integer wavelet transform, called the S-transform (sec. 5.3). We gave the quantum description, quantum algorithms, and circuit-gate implementation on Qiskit. This approach represents our first journal paper in the field of quantum computing. So far, we have written 60% of the content of the article and plan to submit it during the remainder of this year.

References

[1] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, *et al.*, "Noisy intermediate-scale quantum (nisq) algorithms," *arXiv preprint arXiv:2101.08448*, 2021.

[2] S. Akama, *Elements of quantum computing*. Springer, 2015.

[3] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[4] E. Aboufadel and S. Schlicker, "Wavelets, introduction," 2003.

[5] N. Abura'ed, F. S. Khan, and H. Bhaskar, "Advances in the quantum theoretical approach to image processing applications," *ACM Computing Surveys (CSUR)*, vol. 49, no. 4, pp. 1–49, 2017.

[6] Y. Ruan, X. Xue, and Y. Shen, "Quantum image processing: Opportunities and challenges," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[7] W. C. Hughes, R.J, "Quantum computing: final frontier?," *IEEE Intelligent Systems and their Applications*, 2000.

[8] K. M. Svore and M. Troyer, "The quantum future of computation," *Computer*, vol. 49, no. 9, pp. 21–30, 2016.

[9] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum internet: Networking challenges in distributed quantum computing," *IEEE Network*, vol. 34, no. 1, pp. 137–143, 2020.

[10] H.-S. Li, P. Fan, H.-y. Xia, S. Song, and X. He, "The multi-level and multi-dimensional quantum wavelet packet transforms," *Scientific reports*, vol. 8, no. 1, pp. 1–23, 2018.

[11] H.-S. Li, P. Fan, H. Peng, S. Song, and G.-L. Long, "Multilevel 2-d quantum wavelet transforms," *IEEE Transactions on Cybernetics*, 2021.

[12] H.-S. Li, P. Fan, H.-y. Xia, and S. Song, "Quantum multi-level wavelet transforms," *Information Sciences*, vol. 504, pp. 113–135, 2019.

[13] P. Hoyer, “Efficient quantum transforms,” *arXiv preprint quant-ph/9702028*, 1997.

[14] A. Fijany and C. P. Williams, “Quantum wavelet transforms: Fast algorithms and complete circuits,” in *NASA international conference on quantum computing and quantum communications*, pp. 10–33, Springer, 1998.

[15] A. Klappenecker, “Wavelets and wavelet packets on quantum computers,” in *Wavelet Applications in Signal and Image Processing VII*, vol. 3813, pp. 703–713, International Society for Optics and Photonics, 1999.

[16] H. Chao, P. Fisher, and Z. Hua, “An approach to integer wavelet transformations for lossless image compression,” *Advances in Computational Mathematics*, vol. 202, pp. 13–38, 1998.

[17] M. Martina, G. Masera, G. Piccinini, and M. Zamboni, “A vlsi architecture for iwt (integer wavelet transform),” vol. 3, pp. 1174–1177 vol.3, 2000.

[18] M. D. Adams and F. Kossentni, “Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis,” *IEEE Transactions on image Processing*, vol. 9, no. 6, pp. 1010–1024, 2000.

[19] C. Lin, B. Zhang, and Y. F. Zheng, “Packed integer wavelet transform constructed by lifting scheme,” *IEEE transactions on circuits and systems for video technology*, vol. 10, no. 8, pp. 1496–1501, 2000.

[20] A. Gepp and P. Stocks, “A review of procedures to evolve quantum algorithms,” *Genetic programming and evolvable machines*, vol. 10, no. 2, pp. 181–228, 2009.

[21] S. Chakraborty, S. H. Shaikh, A. Chakrabarti, and R. Ghosh, “An image denoising technique using quantum wavelet transform,” *International Journal of Theoretical Physics*, vol. 59, no. 11, pp. 3348–3371, 2020.

[22] F. Yan, A. M. Iliyasu, and S. E. Venegas-Andraca, “A survey of quantum image representations,” *Quantum Information Processing*, vol. 15, no. 1, pp. 1–35, 2016.

[23] J. Su, X. Guo, C. Liu, and L. Li, “A new trend of quantum image representations,” *IEEE Access*, vol. 8, pp. 214520–214537, 2020.

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

[25] P. Q. Le, F. Dong, and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," *Quantum Information Processing*, vol. 10, no. 1, pp. 63–84, 2011.

[26] F. Yan, A. M. Iliyasu, and P. Q. Le, "Quantum image processing: a review of advances in its security technologies," *International Journal of Quantum Information*, vol. 15, no. 03, p. 1730001, 2017.

[27] H.-S. Li, S. Song, P. Fan, H. Peng, H.-y. Xia, and Y. Liang, "Quantum vision representations and multi-dimensional quantum transforms," *Information Sciences*, vol. 502, pp. 42–58, 2019.

[28] P. Li, B. Wang, H. Xiao, and X. Liu, "Quantum representation and basic operations of digital signals," *International Journal of Theoretical Physics*, vol. 57, no. 10, pp. 3242–3270, 2018.

[29] H.-S. Li, P. Fan, H.-Y. Xia, H. Peng, and S. Song, "Quantum implementation circuits of quantum signal representation and type conversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 341–354, 2018.

[30] R. K. Bera, *The Amazing World of Quantum Computing*. Springer, 2020.

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

[32] P. W. Shor, "Progress in quantum algorithms," *Quantum information processing*, vol. 3, no. 1, pp. 5–13, 2004.

[33] P. W. Shor, "Why haven't more quantum algorithms been found?," *Journal of the ACM (JACM)*, vol. 50, no. 1, pp. 87–90, 2003.

[34] A. Adedoyin, J. Ambrosiano, P. Anisimov, A. Bärtschi, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, *et al.*, "Quantum algorithm implementations for beginners," *arXiv preprint arXiv:1804.03719*, 2018.

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

[36] R. Jozsa, "Quantum factoring, discrete logarithms, and the hidden subgroup problem," *Computing in science & engineering*, vol. 3, no. 2, pp. 34–43, 2001.

[37] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.

[38] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

[39] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.

[40] J. Kempe, "Quantum random walks: an introductory overview," *Contemporary Physics*, vol. 44, no. 4, pp. 307–327, 2003.

[41] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem," *Science*, vol. 292, no. 5516, pp. 472–475, 2001.

[42] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, "Wavelet toolbox," *The MathWorks Inc., Natick, MA*, vol. 15, p. 21, 1996.

[43] P. S. Addison, *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance.* CRC press, 2017.

[44] A. Shaik and V. Thanikaiselvan, "Comparative analysis of integer wavelet transforms in reversible data hiding using threshold based histogram modification," *Journal of King Saud University - Computer and Information Sciences*, 2018.

[45] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," in *Fundamental Papers in Wavelet Theory*, pp. 494–513, Princeton University Press, 2009.

[46] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier analysis and applications*, vol. 4, no. 3, pp. 247–269, 1998.

[47] R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, "Lifting construction of non-linear wavelet transforms," in *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No. 98TH8380)*, pp. 49–52, IEEE, 1998.

[48] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Applied and computational harmonic analysis*, vol. 5, no. 3, pp. 332–369, 1998.

[49] K. Sayood, *Introduction to data compression.* Morgan Kaufmann, 2017.

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

[51] D. Gosal and W. Lawton, "Quantum haar wavelet transforms and their applications," 2001.

[52] W.-W. Hu, R.-G. Zhou, A. El-Rafei, and S.-X. Jiang, "Quantum image watermarking algorithm based on haar wavelet transform," *IEEE Access*, vol. 7, pp. 121303–121320, 2019.

[53] S. Heidari, M. Naseri, R. Gheibi, M. Baghfalaki, M. R. Pourarian, and A. Farouk, "A new quantum watermarking based on quantum wavelet transforms," *Communications in theoretical Physics*, vol. 67, no. 6, p. 732, 2017.

[54] X.-H. Song, S. Wang, S. Liu, A. A. Abd El-Latif, and X.-M. Niu, "A dynamic watermarking scheme for quantum images using quantum wavelet transform," *Quantum information processing*, vol. 12, no. 12, pp. 3689–3706, 2013.

[55] H.-S. Li, Q. Zhu, M.-C. Li, H. Ian, *et al.*, "Multidimensional color image storage, retrieval, and compression based on quantum amplitudes and phases," *Information Sciences*, vol. 273, pp. 212–232, 2014.

[56] S. Wang, X. Song, and X. Niu, "A novel encryption algorithm for quantum images based on quantum wavelet transform and diffusion," in *Intelligent Data analysis and its Applications, Volume II*, pp. 243–250, Springer, 2014.

[57] N. E. Miner, "An introduction to wavelet theory and analysis," 1998.

[58] A. Said and W. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1303–1310, 1996.

[59] M. Grangetto, E. Magli, M. Martina, and G. Olmo, "Optimization and implementation of the integer wavelet transform for image coding," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 596–604, 2002.

[60] M. S. ANIS, H. Abraham, AduOffei, R. Agarwal, G. Agliardi, M. Aharoni, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum, E. Arbel, A. Asfaw, A. Athalye, A. Avkhadiev, C. Azaustre, A. Banerjee, S. Banerjee, W. Bang, A. Bansal, P. Barkoutsos, A. Barnawal, G. Barron, G. S. Barron, L. Bello, Y. Ben-Haim, D. Bevenius, D. Bhatnagar, A. Bhobe, P. Bianchini, L. S. Bishop, C. Blank, S. Bolos, S. Bopardikar, S. Bosch, S. Brandhofer, Brandon, S. Bravyi, N. Bronn, Bryce-Fuller, D. Bucher, A. Burov, F. Cabrera, P. Calpin, L. Capelluto, J. Carballo, G. Carrascal, A. Carriker, I. Carvalho, A. Chen, C.-F. Chen, E. Chen, J. C. Chen, R. Chen, F. Chevallier, R. Cholarajan, J. M. Chow, S. Churchill, C. Claus, C. Clauss, C. Clothier, R. Cocking, R. Cocuzzo, J. Connor, F. Correa, A. J. Cross, A. W. Cross, S. Cross, J. Cruz-Benito, C. Culver, A. D. Córcoles-Gonzales, N. D, S. Dague, T. E. Dandachi, A. N. Dangwal, J. Daniel, M. Daniels, M. Dartiailh, A. R. Davila, F. Debouni, A. Dekusar, A. Deshmukh, M. Deshpande, D. Ding, J. Doi, E. M. Dow, E. Drechsler, E. Dumitrescu, K. Dumon, I. Duran, K. EL-Safty, E. Eastman, G. Eberle, A. Ebrahimi, P. Eendebak, D. Egger, A. Espiricueta, M. Everitt, D. Facoetti, Farida, P. M. Fernández, S. Ferracin, D. Ferrari, A. H. Ferrera, R. Fouilland, A. Frisch, A. Fuhrer, B. Fuller, M. GEORGE, J. Gacon, B. G. Gago, C. Gambella, J. M. Gambetta, A. Gammanpila, L. Garcia, T. Garg, S. Garion, T. Gates, L. Gil, A. Gilliam, A. Giridharan, J. Gomez-Mosquera, Gonzalo, S. de la Puente González, J. Gorzinski, I. Gould, D. Greenberg, D. Grinko, W. Guan, J. A. Gunnels, N. Gupta, J. M. Günther, M. Haglund, I. Haide, I. Hamamura, O. C. Hamido, F. Harkins, A. Hasan, V. Havlicek, J. Hellmers, Ł. Herok, S. Hillmich, H. Horii, C. Howington, S. Hu, W. Hu, J. Huang, R. Huisman, H. Imai, T. Imamichi, K. Ishizaki, Ishwor, R. Iten, T. Itoko, A. Javadi, A. Javadi-Abhari, W. Javed, M. Jivrajani, K. Johns, S. Johnstun, Jonathan-Shoemaker, JosDenmark, JoshDumo, J. Judge, T. Kach-

mann, A. Kale, N. Kanazawa, J. Kane, Kang-Bae, A. Kapila, A. Karazeev, P. Kassebaum, J. Kelso, S. Kelso, V. Khanderao, S. King, Y. Kobayashi, A. Kovyrshin, R. Krishnakumar, V. Krishnan, K. Krsulich, P. Kumkar, G. Kus, R. LaRose, E. Lacal, R. Lambert, J. Lapeyre, J. Latone, S. Lawrence, C. Lee, G. Li, J. Lishman, D. Liu, P. Liu, Y. Maeng, S. Maheshkar, K. Majmudar, A. Malyshev, M. E. Mandouh, J. Manela, Manjula, J. Marecek, M. Marques, K. Marwaha, D. Maslov, P. Maszota, D. Mathews, A. Matsuo, F. Mazhandu, D. McClure, M. McElaney, C. McGarry, D. McKay, D. McPherson, S. Meesala, D. Meirom, C. Mendell, T. Metcalfe, M. Mevissen, A. Meyer, A. Mezzacapo, R. Midha, Z. Minev, A. Mitchell, N. Moll, A. Montanez, G. Monteiro, M. D. Mooring, R. Morales, N. Moran, D. Morcuende, S. Mostafa, M. Motta, R. Moyard, P. Murali, J. Müggenburg, D. Nadlinger, K. Nakanishi, G. Nannicini, P. Nation, E. Navarro, Y. Naveh, S. W. Neagle, P. Neuweiler, A. Ngoueya, J. Nicander, Nick-Singstock, P. Niroula, H. Norlen, NuoWenLei, L. J. O'Riordan, O. Ogunbayo, P. Ollitrault, T. Onodera, R. Otaolea, S. Oud, D. Padilha, H. Paik, S. Pal, Y. Pang, A. Panigrahi, V. R. Pascuzzi, S. Perriello, E. Peterson, A. Phan, F. Piro, M. Pistoia, C. Piveteau, J. Plewa, P. Pocreau, A. Pozas-Kerstjens, R. Pracht, M. Prokop, V. Prutyanov, S. Puri, D. Puzzuoli, J. Pérez, Quintiii, R. I. Rahman, A. Raja, R. Rajeev, N. Ramagiri, A. Rao, R. Raymond, O. Reardon-Smith, R. M.-C. Redondo, M. Reuter, J. Rice, M. Riedemann, D. Risinger, M. L. Rocca, D. M. Rodríguez, RohithKarur, B. Rosand, M. Rossmannek, M. Ryu, T. SAPV, A. Saha, A. Ash-Saki, M. Sandberg, H. Sandesara, R. Sapra, H. Sargsyan, A. Sarkar, N. Sathaye, B. Schmitt, C. Schnabel, Z. Schoenfeld, T. L. Scholten, E. Schoute, M. Schulterbrandt, J. Schwarm, J. Seaward, Sergi, I. F. Sertage, K. Setia, F. Shah, N. Shammah, R. Sharma, Y. Shi, J. Shoemaker, A. Silva, A. Simonetto, D. Singh, P. Singh, P. Singkanipa, Y. Siraichi, Siri, J. Sistos, I. Sitdikov, S. Sivarajah, M. B. Sletfjerding, J. A. Smolin, M. Soeken, I. O. Sokolov, I. Sokolov, SooluThomas, Starfish, D. Steenken, M. Stypulkoski, A. Suau, S. Sun, K. J. Sung, M. Suwama, O. Słowik, H. Takahashi, T. Takawale, I. Tavernelli, C. Taylor, P. Taylour, S. Thomas, M. Tillet, M. Tod, M. Tomasik, E. de la Torre, J. L. S. Toural, K. Trabing, M. Treinish, D. Trenev, TrishaPe, F. Truger, G. Tsilimigkounakis, D. Tulsi, W. Turner, Y. Vaknin, C. R. Valcarce, F. Varchon, A. Vartak, A. C. Vazquez, P. Vijaywargiya, V. Villar, B. Vishnu, D. Vogt-Lee, C. Vuillot, J. Weaver, J. Weidenfeller, R. Wieczorek, J. A. Wildstrom, J. Wilson, E. Winston, WinterSoldier, J. J. Woehr, S. Woerner, R. Woo, C. J. Wood, R. Wood, S. Wood, J. Wootton, M. Wright, B. Yang, D. Yeralin, R. Yonekura, D. Yonge-Mallo, R. Young, J. Yu, L. Yu, C. Zachow, L. Zdanski, H. Zhang, C. Zoufal, aeddins ibm, alexzhang13, b63, bartek bartlomiej, bcamorrison, brandhsn, catornow, charmerDark, deeplokhande, dekel.meirom, dime10, ehchen, fanizzamarco, fs1132429, gadial, galeinston, georgezhou20, georgios ts, gruu, hhorii, hykavitha, itoko, jliu45, jscott2, klinvill, krutik2966, ma5x, michelle4654, msuwama, ntgiwsvp, ordmoj, sagar pahwa, pritamsinha2304, ryancocuzzo, saswati qiskit, septembrr, sethmerkel, shaashwat, sternparky, strickroman, tigerjack, tsura crisaldo, welien, willhbang, yang.luh, and M. Čepulkovskis, "Qiskit: An open-source framework for quantum computing," 2021.

[61] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on image processing*, vol. 5, no. 9, pp. 1303–1310, 1996.

[62] C. Mulcahy, "Plotting and scheming with wavelets," *Mathematics Magazine*, vol. 69, no. 5, pp. 323–343, 1996.

[63] G. Luo, R.-G. Zhou, J. Luo, W. Hu, Y. Zhou, and H. Ian, "Adaptive lsb quantum watermarking method using tri-way pixel value differencing," *Quantum Information Processing*, vol. 18, no. 2, pp. 1–20, 2019.

[64] R. Zhou, W. Hu, G. Luo, X. Liu, and P. Fan, "Quantum realization of the nearest neighbor value interpolation method for ineqr," *Quantum Information Processing*, vol. 17, no. 7, pp. 1–37, 2018.

[65] N. Jiang, N. Zhao, and L. Wang, "Lsb based quantum image steganography algorithm," *International Journal of Theoretical Physics*, vol. 55, no. 1, pp. 107–123, 2016.

[66] R.-G. Zhou, W. Hu, P. Fan, and H. Ian, "Quantum realization of the bilinear interpolation method for neqr," *Scientific reports*, vol. 7, no. 1, pp. 1–17, 2017.

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

[68] W. Scherer, *Mathematics of Quantum Computing*. Springer, 2019.

[69] J. Sang, S. Wang, and X. Niu, "Quantum realization of the nearest-neighbor interpolation method for frqi and neqr," *Quantum Information Processing*, vol. 15, no. 1, pp. 37–64, 2016.