



Paired quantum Fourier transform with $\log_2 N$ Hadamard gates

Artyom M. Grigoryan¹ · Sos S. Agaian²

Received: 9 January 2019 / Accepted: 9 May 2019 / Published online: 25 May 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The quantum Fourier transform (QFT) is perhaps the furthestmost central building block in creation quantum algorithms. In this work, we present a new approach to compute the standard quantum Fourier transform of the length $N = 2^r$, $r > 1$, which also is called the r -qubit discrete Fourier transform. The presented algorithm is based on the paired transform developed by authors. It is shown that the signal-flow graphs of the paired algorithms could be used for calculating the quantum Fourier and Hadamard transform with the minimum number of stages. The calculation of all components of the transforms is performed by the Hadamard gates and matrices of rotations and all simple NOT gates. The new presentation allows for implementing the QFT (a) by using only the r Hadamard gates and (b) organizing parallel computation in r stages. Also, the circuits for the length- 2^r fast Hadamard transform are described. Several mathematical illustrative examples of the order the $N = 4, 8$, and 16 cases are illustrated. Finally, the QFT for inputs being two, three and four qubits is described in detail.

Keywords Quantum Fourier transform · Quantum computing · Quantum Hadamard transform · Paired transform · Fast Fourier transform

1 Introduction

Quantum Fourier transform (QFT) is analogues to the discrete Fourier transform (DFT) and is one of the essential operations in quantum computing. The QFT is a linear transformation on quantum bits. The QFT is a basic part of many quantum algorithms, including the Shor's algorithm for factoring, Deutsch and Simon algorithms, the dis-

✉ Artyom M. Grigoryan
amgrigoryan@utsa.edu

¹ Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, USA

² Computer Science Department, The College of Staten Island New York, New York City, USA

crete logarithm, the quantum phase estimation algorithm for estimating the eigenvalues of a unitary operator, algorithms for the hidden subgroup problem and others [1–3]. Additionally, the quantum QFT plays an essential role on the exponential speed-up over the best classical algorithms for integer factorization. The QFT also is considered in color image encryption, processing and representation of quantum images [4–9].

The r -qubit QFT can be completed resourcefully on a quantum computer, with

- a certain decomposition into a product of sparse unitary and diagonal matrices.
- a quantum circuit consisting of only $O(r^2)$ Hadamard gates and controlled phase shift gates, where r is the number of input qubits.

The critical issue of the quantum information processing algorithm is to create algorithms that are far superior in asymptotic running time to any “classical” counterparts known to date which solve the same problems. Shor presented the QFT computation “mixed-radix” method [1]. Coppersmith showed that in the $N = 2^r$ case, there exist quantum circuits performing the QFT with $O(r^2)$ gates [10]. This complexity is small when comparing with $O(r2^r)$ for the Cooley–Tukey fast Fourier transform (FFT) [11]. These circuits are based on a recursive description of the QFT that is analogous to the description of the DFT exploited by the fast transforms. Different FFT algorithms have their advantages and can lead to different effective circuits for the QFT. Practically, the QFT offers a different way to make arithmetic operations on a quantum computer. The QFT algorithm represents a significant perfection in the complexity of the commonly used FFT algorithm, from $O(N\log_2 N)$ to $O(\log_2^2 N)$ [10, 12]. The QFT can be implemented approximately by eliminating the rotation gates with smaller a certain threshold value angle. Recently, several approximate QFT computation algorithms have been developed [10, 13–19]. For example, Coppersmith presented an approximate QFT computation, with complexity $O(N\log_2 N)$, where $1 \leq m \leq \log_2 N$ is the parameter that defines a degree of approximation, usually taken as $O(\log_2 \log_2 N)$. Lately, an implementation of the QFT by using multiple-valued quantum gates (by replacing the Walsh–Hadamard gate with the so-called Chrestenson quantum gates) was proposed [20].

In this paper, we build quantum circuits for the r -qubit QFT from the N -point fast Fourier transform (FFT), when $N = 2^r$. We describe and analyze signal-flow graphs of the paired FFT and circuits that can be implemented for calculating the discrete Fourier transform in a quantum computer. The method of paired transform is used [21–23]. This technique may provide insight into the development of new quantum algorithms. The quantum DFT for inputs being two, three and four qubits is described in detail. The resulting unitary matrices can be interpreted as quantum logic gates. The circuits for calculating the fast discrete Hadamard transform (DHT) are also considered. In paired representation, the DFT is reduced to the DHT, when omitting all twiddle factors [24, 25]. Therefore, for instance, the circuit for the 16-point DFT can easily be simplified for the 16-point DHT. The rest of the paper is organized in the following way. In Sect. 2, a brief background information related to the FFT and QFT is given. Section 3 introduces the concept of the discrete paired transform and its fast algorithm by the Hadamard gates. Examples with the 2-qubit DFT and DHT are described. Sections 4 and 5 present the 8- and 16-point paired FFTs and the corresponding circuits for the 3- and 4-qubit DFTs. The conclusion is given in Sect. 6.

2 Background

The concept of the discrete Fourier transform is used widely in different areas of signal processing and communication systems. Many fast algorithms were developed, which are called the fast Fourier transforms (FFT), and one of them is the split radix N -point FFT, when $N = 2^r$, $r > 1$; it is the well-known decimation-in-frequency FFT (Cooley–Tukey 1965). This algorithm splits the transform by two length- $N/2$ transforms, and such process is continued for the $N/2$ -point DFTs and so on. One of the first FFTs used for quantum circuits was the decimation-in-time algorithm, wherein $N/2$ butterflies over two $N/2$ DFTs define the values of the N -point DFT. We also mention the paired FFT, or the FFT by the paired transform (Grigoryan 1986) that splits recursively the transform by the $N/2$, $N/4$, $N/8$, ..., 4, 2, 1-point DFTs. This splitting requires $m(N) = N/2(r - 3) + 2$ operations of complex multiplication [21–24]. The concept of the DFT was also applied in quaternion and octonion arithmetic, fast algorithms were developed and applied for color image processing [26]. The DFT became important in quantum computing and used in many different algorithms [17, 27–31].

When comparing the quantum DFT and the classical FFT, it should be noted the following:

1. The concept of the N -point DFT was moved formally from the complex arithmetic to quantum computing. It is not a transform on N qubits; it is a transform of amplitudes of the N -dimensional state of r qubits. The classical DFT has a physical meaning; it is the spectral characteristic of the signal. As a system, this transform presents a beautiful and real process of rotation of data around circles in the complex plane. In general, the DFT exists with rotation around ellipses; we call such transforms the elliptic DFTs [24, 32]. The Fourier transform was originated from the concept of the Fourier series of discrete periodic signals. Does the concept of periodic discrete qubit signals exist in quantum computing? The geometric interpretation of the DFT in quantum computing is unclear. It is possible that the definition of the Fourier transform on qubits will be found and clearly justified in future.
2. All fast algorithms of the DFT were developed on real and complex inputs, as well as for quaternions and octonions [26, 33–35]. In matrix representation, any FFT algorithm presents a decomposition of the DFT matrix by simple and sparse unitary matrices plus a few diagonal matrices that contains a small number of twiddle factors. The first known circuit of the quantum r -qubit DFT, which is shown in Fig. 1 for the 3-qubit DFT [2], is not exception.
3. The analysis of the paired FFT shows that, with optimal organization and parallel computing, the splitting of the FFT in the classical computer can be performed for $2\log_2(N)$ steps, including the reordering of the output. In quantum computing, the DFT can be accomplished by about $(\log_2 N)^2$ elements, or steps. For instance, the 3-qubit DFT, or the 8-point inverse DFT that is shown in Fig. 1 uses seven elements.
4. The FFT has been used in classical computers for many decades and applied in many areas of engineering, and the QFT exists only in the theory. To this day,

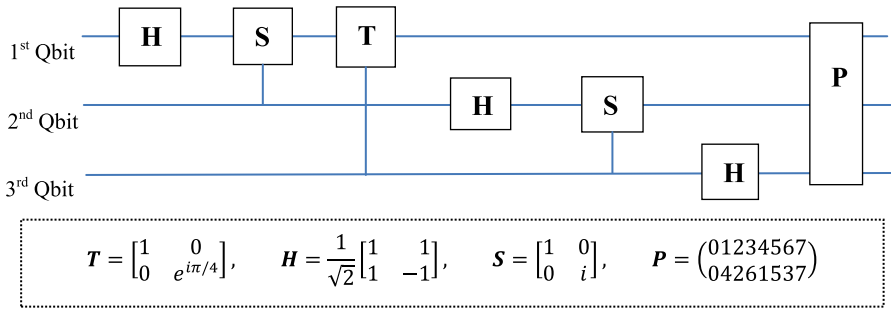


Fig. 1 The known circuit for the 3-qubit DFT

the concept of the QFT is used as a theoretical tool for solving different tasks in quantum computing. There are no real practical examples of accomplishment of the QFT in the quantum computer.

5. Simplest gate on one qubit is called the Hadamard gate which is shown in Fig. 1 as the **H** gate.

3 Fast discrete paired transform

In this section, we briefly describe the concept of the paired representation of the signal and the splitting of the DFT. The discrete paired transform (DPT) is the frequency–time representation of the signal that corresponds to a special splitting of both discrete Fourier and Hadamard transforms. The Hadamard matrix is considered in the recursive form of Sylvester’s construction as

$$A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{1}$$

$$A_4 = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} A_2 & -A_2 \\ A_2 & A_2 \end{bmatrix},$$

$$A_N = \begin{bmatrix} A_{N/2} & -A_{N/2} \\ A_{N/2} & A_{N/2} \end{bmatrix}, \quad N = 2^r, \quad r > 2. \tag{2}$$

The basic 2×2 matrix A_2 is considered without the normalized coefficient $1/\sqrt{2}$. The determinant of this matrix is 2, and the inverse matrix is

$$A_2^{-1} = \frac{1}{2} A_2' = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

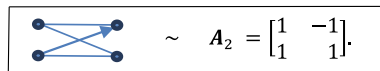
The normalized matrix $1/\sqrt{2}A_2$ is unitary, which is important to mention since in quantum computing the operations on one qubit are described by only unitary transforms. The matrix A_2 is similar to the Walsh–Hadamard matrix

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{3}$$

with different order of rows. The matrix A_2 can be described by the NOT gate X and gate H ,

$$A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = XH. \tag{4}$$

Thus, the matrix A_2 is the same Hadamard gate that is used in quantum computing (or the matrix H in Fig. 1), but with different order of output. In the graphs, this matrix will be shown by the following butterfly:



The normalized coefficient is omitted because the N -point DFT is defined in digital signal processing as

$$F_p = \sum_{n=0}^{N-1} f_n W_N^{np} = \sum_{n=0}^{N-1} f_n e^{-i2\pi np/N}, \quad p = 0, 1, 2, \dots, (N - 1). \tag{5}$$

The inverse transform is calculated by

$$f_n = \frac{1}{N} \sum_{p=0}^{N-1} F_p W_N^{-np} = \frac{1}{N} \sum_{p=0}^{N-1} F_p e^{i2\pi np/N}, \quad n = 0, 1, 2, \dots, (N - 1). \tag{6}$$

It should be noted that in many publications of quantum computing [2, 28], the discrete Fourier transform is defined in a different way, namely with clock-wise rotation of input data and normalized coefficient,

$$\check{F}_p = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{i2\pi np/N}, \quad p = 0, 1, 2, \dots, (N - 1). \tag{7}$$

The paired FFT splits the transform of order $N = 2^r, r > 1$, by the set of $(r + 1)$ DFTs of lengths $N/2, N/4, N/8, \dots, 2, 1, 1$ [24, 36].

Example 1 The signal-flow graph of the 4-point DFT is shown in Fig. 2.

Fig. 2 Signal-flow graph of the paired 4-point FFT

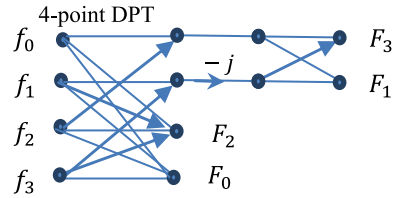
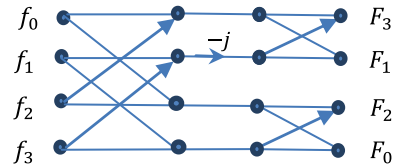


Fig. 3 Signal-flow graph of the 4-point DFT



The transform is split by the 4- and 2-point discrete paired transforms (DPT) with the matrices [24]

$$\chi'_4 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \chi'_2 = A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

as follows:

$$\begin{bmatrix} F_3 \\ F_1 \\ F_2 \\ F_0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}.$$

The graph of the 4-point DFT given in Fig. 2 can be redrawn, as shown in Fig. 3. Therefore, the transform can be written as

$$\begin{bmatrix} F_3 \\ F_1 \\ F_2 \\ F_0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}.$$

Considering the matrix of permutation of outputs

$$S_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

the matrix of the 4-point DFT can be presented as

$$F_4 = S_4 \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Direct calculations show that

$$F_4 = S_4 \begin{bmatrix} 1 & i & -1 & -i \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}.$$

Let W_2 be the matrix of rotation

$$W_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and let $A_{0,2;1,3}$ and $A_{0,1;2,3}$ be, respectively, the matrices

$$A_{0,2;1,3} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad A_{0,1;2,3} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Each of these matrices is described by two Hadamard gates,

$$A_{0,2;1,3} = A_{0,2} A_{1,3} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$A_{0,1;2,3} = A_{0,1} A_{2,3} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Here and hereinafter, we denote by $A_{k,l}$ the identity matrix with two additional coefficients equal 1 and -1 at positions (k, l) and (l, k) , respectively, when $k \neq l$. This matrix describes the Hadamard gate with inputs being the k th and l th components of data.

Thus, the matrix of the 4-point DFT can be decomposed as

$$F_4 = S_4 A_{0,1} A_{2,3} W_2 A_{1,3} A_{0,2}, \tag{8}$$

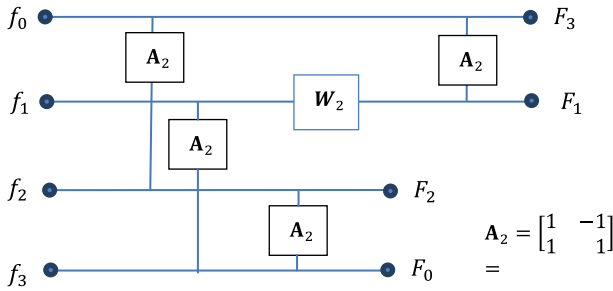


Fig. 4 Circuit for the 4-point DFT

or

$$F_4 = S_4 A_{0,1} W_2 \underbrace{A_{2,3} A_{0,2} A_{1,3}} \tag{9}$$

Here, the product of three matrices is the matrix of the 4-point paired transform, χ'_4 . Figure 4 shows the circuit of the 4-point DFT with four Hadamard gates and one trivial rotation matrix. Each Hadamard gates requires one addition and one subtraction.

A two-qubit state is described by four complex or real numbers of four-dimensional vector $\check{f} = (f_{00}, f_{01}, f_{10}, f_{11})$,

$$|\check{f}\rangle = f_{00}|00\rangle + f_{01}|01\rangle + f_{10}|10\rangle + f_{11}|11\rangle,$$

with the condition that $|f_{00}|^2 + |f_{01}|^2 + |f_{10}|^2 + |f_{11}|^2 = 1$. These numbers are called the amplitudes of the 2-qubit state. The two-qubit DFT state

$$F|\check{f}\rangle = F_{00}|00\rangle + F_{01}|01\rangle + F_{10}|10\rangle + F_{11}|11\rangle$$

is calculated by

$$\begin{aligned} F_{00} &= f_{00} + f_{01} + f_{10} + f_{11}, \\ F_{01} &= f_{00} - if_{01} - f_{10} + if_{11}, \\ F_{10} &= f_{00} - f_{01} + f_{10} - f_{11}, \\ F_{11} &= f_{00} + if_{01} - f_{10} - if_{11}. \end{aligned}$$

The circuit for calculating the 4-point DFT, or 2-qubit DFT by the paired transforms is shown in Fig. 5.

In quantum computing, the states $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$ are numbered simply by 0, 1, 2, and 3, as these numbers presented in binary form. The 2-qubit state $|\check{f}\rangle$ can be written as the vector $\check{f} = (f_0, f_1, f_2, f_3)$, and the same for the Fourier transform, $F|\check{f}\rangle = (F_0, F_1, F_2, F_3)$. Therefore, there is no difference between the circuits in Figs. 4 and 5, which are shown for the complex data and amplitudes of qubits, respectively.

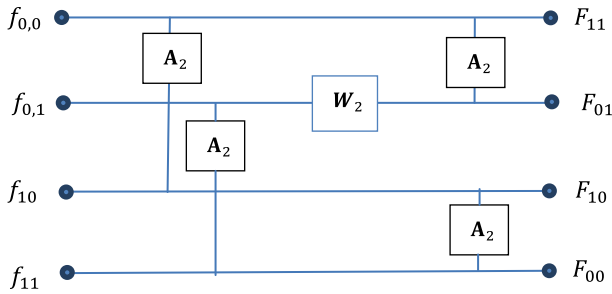


Fig. 5 Circuit for the 2-qubit DFT with three stages

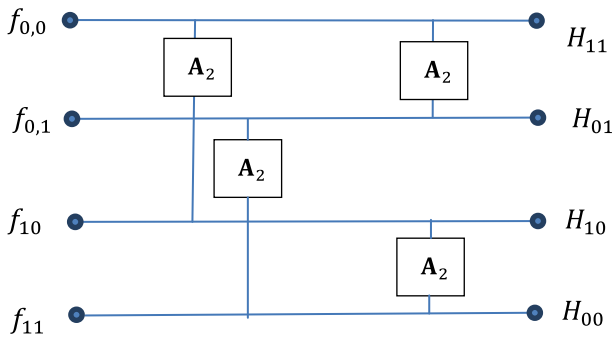


Fig. 6 Circuit for the 2-qubit FHT with two stages

Example 2 We consider the discrete Hadamard transform [37]. When omitting the diagonal matrix in this decomposition in Eq. 8, the transform becomes the 4-point discrete Hadamard transform (DHT), i.e.,

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The circuit for the 4-point fast Hadamard transform (FHT), or the 2-qubit FHT, is shown in Fig. 6.

When considering the 4-point DFT with the normalized coefficient,

$$F_p = \frac{1}{2} \sum_{n=0}^3 f_n W_4^{np} = \frac{1}{2} \sum_{n=0}^3 f_n e^{-i2\pi np/4}, \quad p = 0, 1, 2, 3,$$

the above signal-flow graphs and circuits can be used with the normalized matrix, i.e., in the circuit we need make changes $A_2 \rightarrow A_2/\sqrt{2}$.

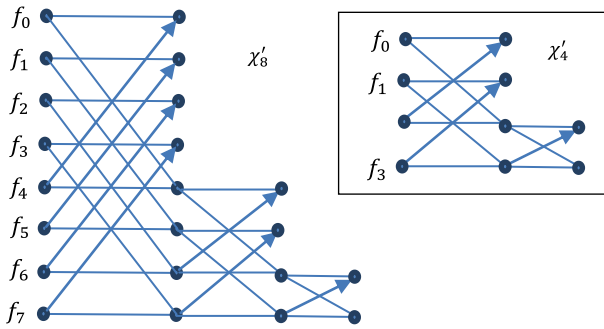


Fig. 7 Signal-flow graphs of the 4- and 8-point DPTs

4 Circuits for length-8 Fourier and Hadamard transforms

In general case of $N = 2^r$, $r > 1$, the discrete paired transform requires only $2(N - 1)$ operations of addition and subtraction [38]. The transform has a matrix with only 0 and ± 1 coefficients. For the $N = 8$ case, the matrix equals

$$[\chi'_8] = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{10}$$

The rows of this matrix, or basic functions of the 8-point DPT, are not normalized. The normalization of the matrix is performed by multiplication with the diagonal matrix $D = \text{diag}(1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/\sqrt{2}, 1/2, 1/2, 1/\sqrt{8}, 1/\sqrt{8})$. The matrix $D[\chi'_8]$ has determinant 1 and is unitary, i.e., its inverse matrix equals the transpose matrix.

For $N = 4$ and 8, the signal-flow graphs of fast DPT are shown in Fig. 7. One can see that the transforms can be calculated by 7 and 3 butterflies.

In the general case, the DPT can be calculated by $(N - 1)$ butterflies or Hadamard matrices A_2 . The matrix of the high-order discrete paired transform can be defined in recursive form

$$[\chi'_{16}] = \begin{bmatrix} I_8 & -I_8 \\ [\chi'_8] & [\chi'_8] \end{bmatrix}, \quad [\chi'_{32}] = \begin{bmatrix} I_{16} & -I_{16} \\ [\chi'_{16}] & [\chi'_{16}] \end{bmatrix}, \dots, \tag{11}$$

where I_M denotes the identity matrix $M \times M$.

Example 3 The signal-flow graph of the 8-point DFT by the paired transforms is shown in Fig. 8.

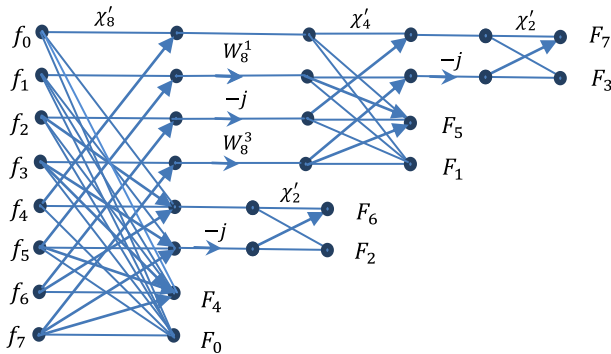


Fig. 8 Signal-flow graph of the paired 8-point FFT

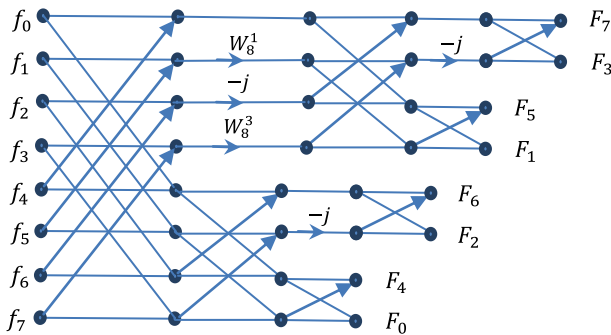


Fig. 9 Signal-flow graph of the 8-point DFT

Here, the twiddle factors are $W^1 = W_8^1 = e^{-i2\pi/8} = 0.7071(1 - i)$, and $W^3 = W_8^3 = e^{-i2\pi/8} = 0.7071(-1 - i)$. According to this signal-flow graph, the matrix of the 8-point DFT can be written as [24, 36]

$$F_8 = S_8 \left[\underbrace{\begin{bmatrix} \chi'_2 & & & \\ & 1 & & \\ & & \text{diag} \left\{ \begin{matrix} 1 \\ -i \\ 1 \\ 1 \end{matrix} \right\} \chi'_4 \\ & & & \chi'_2 \\ & & & & 1 \\ & & & & & 1 \end{bmatrix}} \right] \text{diag} \left\{ \begin{matrix} 1 \\ W_8^1 \\ -i \\ W_8^3 \\ 1 \\ -i \\ 1 \\ 1 \end{matrix} \right\} \chi'_8, \quad (12)$$

where the matrix $S_8: (7, 3, 5, 1, 6, 2, 4, 0) \rightarrow (0, 1, 2, 3, 4, 5, 6, 7)$ corresponds to the permutation of outputs. Using the full graphs of the 8- and 4-point DFTs, we can draw the signal-flow graph of the 8-point DFT, as shown in Fig. 9.

By separating the multiplications by twiddle coefficients, we obtain the signal-flow graph that is shown in Fig. 10. Three stages in this graph correspond to the calculation of the different lengths of butterflies.

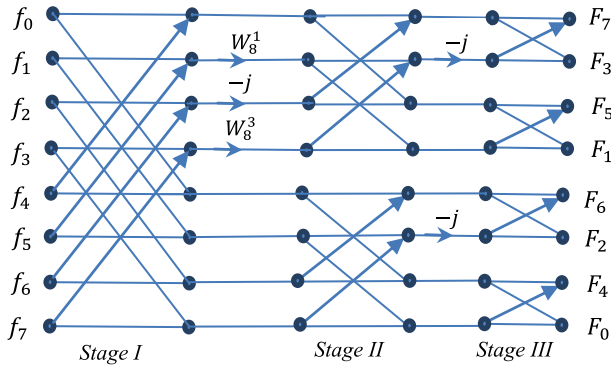


Fig. 10 Signal-flow graph of the 8-point DFT with three groups of butterflies

The matrix of the 8-point DFT can be written as

$$F_8 = S_8 A_{III} W_{II} A_{II} W_I A_I. \tag{13}$$

The matrices in this decomposition are described as follows. The first stage of calculation can be written in matrix form as

$$A_I = A_2 \otimes I_4 = \begin{bmatrix} I_4 & -I_4 \\ I_4 & I_4 \end{bmatrix}.$$

Here, the operation \otimes denotes the Kronecker product of matrices from the right. The matrix A_I describes the operation with four butterflies, or Hadamard gates,

$$A_I = A_{0,4;1,5;2,6;3,7} = A_{0,4} A_{1,5} A_{2,6} A_{3,7}. \tag{14}$$

On the second stage, the calculations are described by the matrix

$$A_{II} = I_2 \otimes (A_2 \otimes I_2) = \begin{bmatrix} I_2 & -I_2 & & \\ I_2 & I_2 & & \\ & & I_2 & -I_2 \\ & & I_2 & I_2 \end{bmatrix}.$$

This is another operation with four butterflies,

$$A_{II} = A_{0,2;1,3;4,6;5,7} = A_{0,2} A_{1,3} A_{4,6} A_{5,7}. \tag{15}$$

On the last stage, we consider the matrix

$$A_{III} = I_4 \otimes A_2 = \begin{bmatrix} A_2 & & & \\ & A_2 & & \\ & & A_2 & \\ & & & A_2 \end{bmatrix}$$

that describes the performance of four butterflies

$$A_{III} = A_{0,1;2,3;4,5;6,7} = A_{0,1}A_{2,3}A_{4,5}A_{6,7}. \tag{16}$$

Also, we introduce the diagonal matrices that describe the multiplications by twiddle factors W^1 , $W^2 = -i$, and W^3 ,

$$W_8^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & W_8^1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \text{diag} \left\{ \begin{matrix} 1 \\ W_8^1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} \right\},$$

$$W_8^2 = \text{diag}\{1, 1, -i, 1, 1, 1, 1, 1\}, \text{ and}$$

$$W_8^3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & W_8^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \text{diag} \left\{ \begin{matrix} 1 \\ 1 \\ 1 \\ W_8^3 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} \right\},$$

These rotation matrices describe three multiplications by the twiddle factors between Stages I and II, and we denote by W_I the product

$$W_I = W_8^1 W_8^2 W_8^3.$$

For two trivial multiplications by $-i$ after Stage II, we introduce the following diagonal matrix:

$$W_{II} = I_2 \otimes W_4^1 = I_2 \otimes \text{diag}\{1, -i, 1, 1\}.$$

All above matrices W are matrices of rotation. The last matrix we need is the matrix of the permutation

$$S_8 = \begin{pmatrix} 01234567 \\ 73516240 \end{pmatrix} = (07)(13)(25)(46).$$

The result of the above reasoning is shown in Fig. 11. It is the circuit for calculating the 8-point complex FFT that can be used in quantum computing with 12 Hadamard

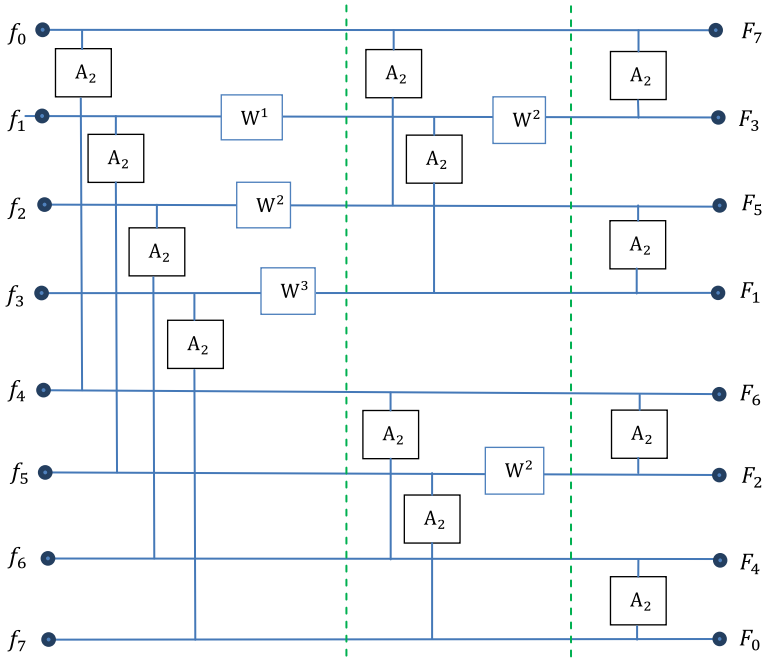


Fig. 11 Circuit for the quantum 8-point complex FFT

gates and five matrices of rotations (three of them are trivial). The 3-qubit state is described as

$$|\check{f}\rangle = f_0|0\rangle + f_1|1\rangle + f_2|2\rangle + \dots + f_7|7\rangle,$$

with condition $|f_0|^2 + |f_1|^2 + \dots + |f_7|^2 = 1$. We consider the new 3-qubit state

$$F|\check{f}\rangle = F_0|0\rangle + F_1|1\rangle + F_2|2\rangle + \dots + F_7|7\rangle.$$

The components of this state, which is called *the 3-qubit DFT*, are calculated by

$$F_p = \sum_{n=0}^7 f_n W_8^{np} = \sum_{n=0}^7 f_n e^{-i2\pi np/8}, \quad p = 0, 1, \dots, 7. \tag{17}$$

One can notice from this circuit, as well as from Eq. 13, that the 8-point FFT includes six stages of parallel computing $f_n \rightarrow S_8 A_{III} W_{II} A_{II} W_I A_I(f_n)$:

- 1 stage. Perform four Hadamard gates in A_I ,
- 2 stage. Perform three multiplications by rotation matrices of W_I ,
- 3 stage. Perform four Hadamard gates in A_{II} ,
- 4 stage. Perform two multiplications by rotation matrices of W_{II} ,
- 5 stage. Perform four Hadamard gates in A_{III} ,

6 stage. Reorder the outputs in accordance with the matrix S_8 .

Thus, there is only six steps and $6 = 2\log_2 N = 2 \times 3$.

Comments The circuits for the quantum DFT are usually drawn with lines that denote the qubits, as shown in Fig. 1. The circuit in Fig. 11 is given in the traditional form with the input and output components, f_n and F_p , respectively. Let us find out if the circuits in these two figures really differ. The matrix A_I with four Hadamard gates is

$$A_I = A_{0,4;1,5;2,6;3,7} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

and it presents the block H , i.e., the Hadamard gate but on the 3rd qubit, not the 1st qubit in the circuit in Fig. 1. Only block H should be changed by the equivalent A_2 . Similarly, the matrix A_{II} with other four Hadamard gates

$$A_{II} = A_{0,2;1,3;4,6;5,7} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{19}$$

is similar to the second block H on the 2nd qubit, as in Fig. 1. Also, in the decomposition of the Fourier matrix in Eq. 13, the matrix A_{III} with four Hadamard gates describes the block A_2 applied on the 1st qubit, and not on the last qubit, as shown in Fig. 1. The same correspondence has place for the rotation matrices in the figures. The diagonal matrix with the twiddle factors in the paired FFT is

$$\text{diag} \begin{Bmatrix} 1 \\ W_8^1 \\ -i \\ W_8^3 \\ 1 \\ -i \\ 1 \\ 1 \end{Bmatrix} = \text{diag} \begin{Bmatrix} 1 \\ W_8^1 \\ -i \\ W_8^3 \end{Bmatrix} \oplus \text{diag} \begin{Bmatrix} 1 \\ -i \end{Bmatrix} \oplus \text{diag} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}.$$

Input f_n and qubits			
n	Qbit 3	Qbit 2	Qbit 1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

→

Output F_p and qubits			
p	X-Qbit 1	X-Qbit 2	X-Qbit 3
7	1	1	1
3	0	1	1
5	1	0	1
1	0	0	1
6	1	1	0
2	0	1	0
4	1	0	0
0	0	0	0

Fig. 12 The order of amplitudes of the input and output in the paired 8-point FFT

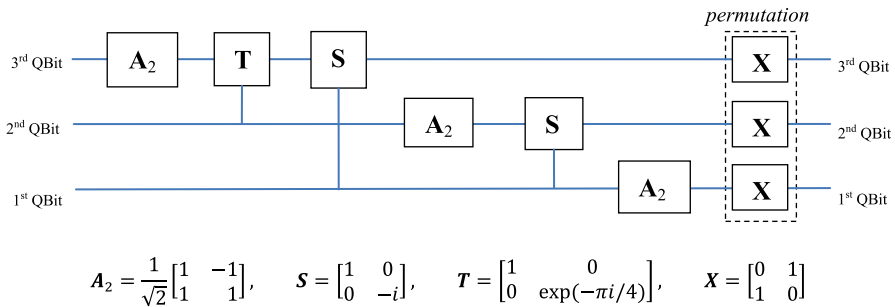


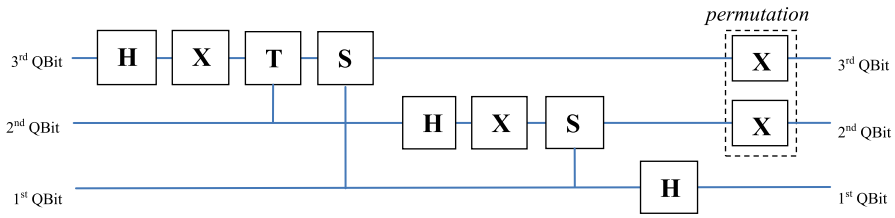
Fig. 13 The circuit for the 3-qubit direct DFT by the paired splitting

Here, the first 4×4 diagonal sub-matrix describes the rotations (similar to blocks S and T in Fig. 1) over the 3rd qubit;

$$\text{diag} \left\{ \begin{matrix} 1 \\ W_8^1 \\ -i \\ W_8^3 \end{matrix} \right\} = T_1 \otimes T_2 \triangleq \text{diag} \left\{ \begin{matrix} 1 \\ -i \end{matrix} \right\} \otimes \text{diag} \left\{ \begin{matrix} 1 \\ W_8^1 \end{matrix} \right\}.$$

The difference between these diagonal matrices, T_1 and T_2 , and matrices S and T in Fig. 1 is in the fact that $-i$ is substituted by i , since the circuit in Fig. 1 is for the inverse DFT not the direct DFT. Now, we consider the order of qubits in the input and output of the circuit for the 3-qubit DFT when splitting by the paired transform. The orders of amplitudes and qubits are shown in Fig. 12.

Here, to change the order of amplitudes in qubits, the quantum NOT gate can be used; $X = [01; 10]$. Fig. 13 shows the circuit of the paired algorithm for the 3-qubit DFT.



$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(-\pi i/4) \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Fig. 14 The circuit for the 3-qubit direct DFT by the paired splitting

When comparing the circuits in Figs. 1 and 13, the following can be noted:

1. The qubits in inputs are given in reversal order;
2. The gates A_2 are used in circuit of Fig. 13, instead of gates H in circuit of Fig. 1;
3. The same NOT gate is used for each qubit on the last stage of the circuit in Fig. 13. In the circuit of Fig. 1, the permutation P of amplitudes of the 8-dimensional vector of the 3-qubit state is used. The matrix of this permutation equals

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Since the A_2 gate is the composition of two gates, $A_2 = XH$, we can change the last gate A_2 with the gate X by one H gate or draw the above circuit with Walsh–Hadamard gates, as shown in Fig. 14. It is not difficult to notice, that the last operations on the 1st output qubit can be written as

$$XA_2 = XXH = H.$$

Example 4 When splitting by the paired transform, the matrix of the 8-point discrete Hadamard transform (DHT) is represented as [24, 39]

$$A_8 = S_8 \begin{bmatrix} \begin{bmatrix} \chi'_2 & & & \\ & 1 & & \\ & & \chi'_4 & \\ & & & 1 \end{bmatrix} & & & \\ & & & \chi'_2 & & & & \\ & & & & 1 & & & \\ & & & & & & & 1 \end{bmatrix} \chi'_8, \quad (\chi'_2 = A_2). \quad (20)$$

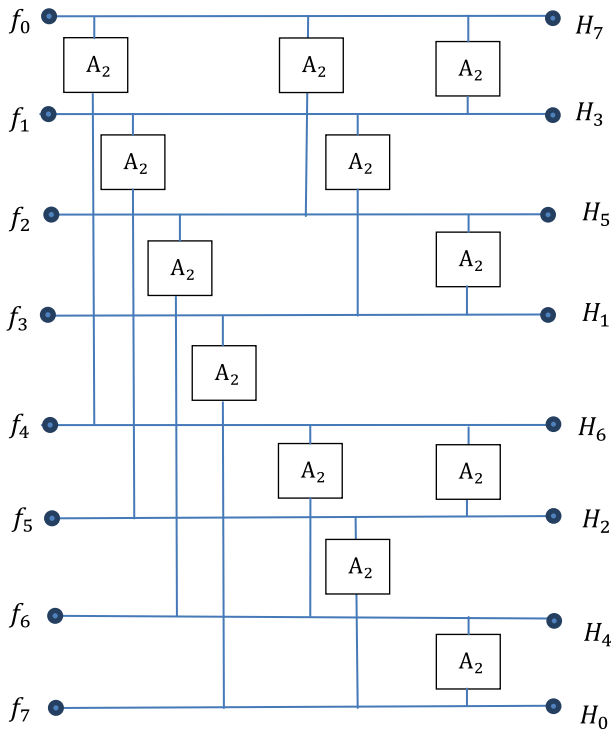


Fig. 15 The circuit for calculating the 8-point DHT with three stages

Here, the matrix of the transformation is considered to be equal

$$A_8 = \begin{bmatrix} A_4 & -A_4 \\ A_4 & A_4 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The circuit in Fig. 11 with omitted all matrices of rotation is shown in Fig. 15. It is the circuit for calculating the 8-point discrete Hadamard transform.

Thus, there are only 4 stages in computing the 8-point DHT, $f_n \rightarrow S_8 A_{III} A_{II} A_I (f_n)$:

- 1 stage. Perform four Hadamard gates in A_I ,
- 2 stage. Perform four Hadamard gates in A_{II} ,
- 3 stage. Perform four Hadamard gates in A_{III} ,
- 4 stage. Reorder the outputs in accordance with matrix S_8 .

The circuit for this 3-qubit DHT is given in Fig. 16.

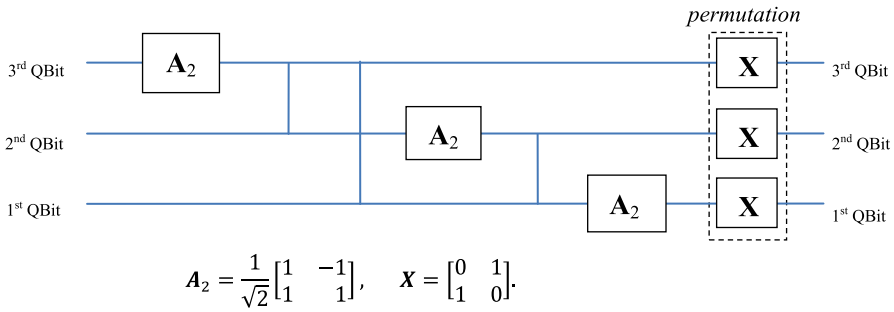


Fig. 16 The circuit for the 3-qubit direct DHT by the paired splitting

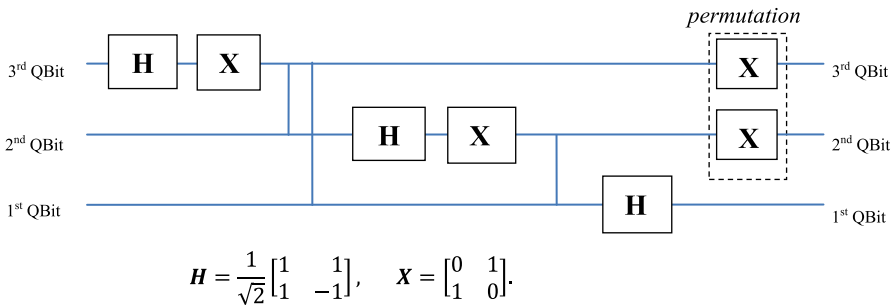


Fig. 17 The circuit for the 3-qubit direct DHT by the paired splitting with Walsh–Hadamard gates

The same circuit but with Walsh–Hadamard gates H instead of A_2 is shown in Fig. 17. Thus, we have two circuits for the same discrete Hadamard transform, one is written for calculation in the classical computer, and another for calculation in the quantum computer.

5 Sixteen-point Fourier and Hadamard transforms

The circuit for calculating the DFT of high order can be described in a way similar to the above considered cases $N = 4$ and 8 . The paired transform splits recursively the DFT into a minimum number of short transforms. It is difficult to expect a recursive calculation in a quantum computer. Therefore, all Hadamard gates in the paired transforms should be grouped by stages, as in the above case $N = 8$ (in Fig. 10). Figure 18 shows the splitting of the 16-point DFT by the paired transforms, namely by one 16- and 8-point DFTs, two 4-point DFTs, and four 2-point DFT which are the butterflies 2×2 [24, 36]. There are only 10 non-trivial twiddle coefficients; seven are trivial multiplications by $-i$. Thus, only 17 rotation matrices are used in the circuit for the 16-point FFT. The number of Hadamard gates (2×2 butterflies) for each 2^k -point DFT equals $(2^k - 1)$. Therefore, the total number of Hadamard gates in the N -point FFT equals $N/2 \log_2 N$, or 32 for the 16-point FFT.

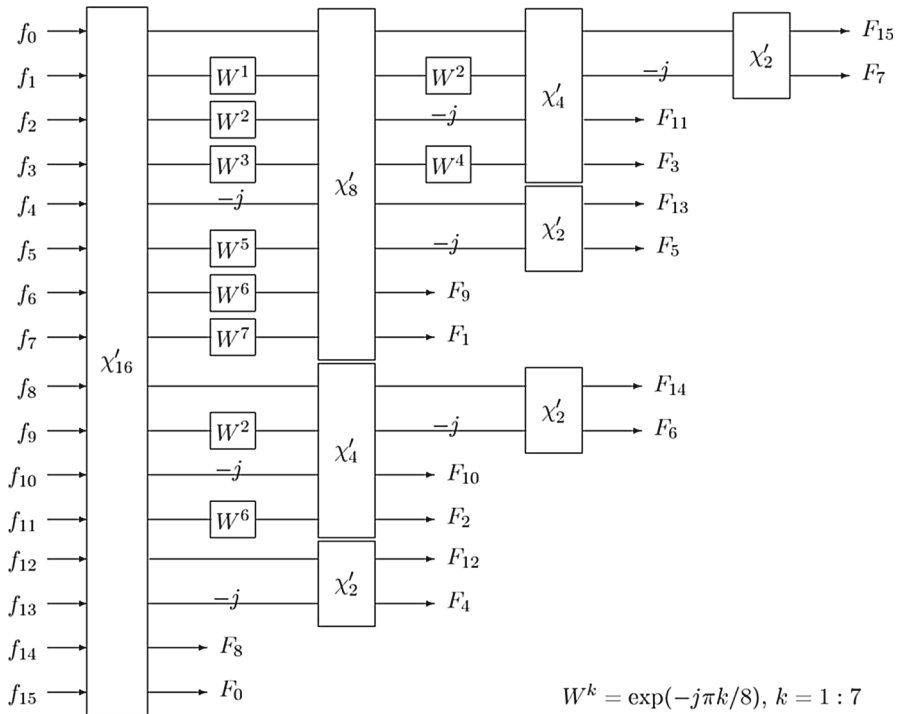


Fig. 18 Block-scheme of calculation of the 16-point paired FFT

When opening this block-diagram in detail, by calculating the paired transforms by the Hadamard gates, we obtain the circuit that is shown in Fig. 19. It is not difficult to see that the calculation of the 16-point FFT includes only 8 stages of parallel computing, including the last operation of reordering the output.

This is the circuit for the 16-point DFT for the classical computer. The same circuit for calculating the 4-qubit DFT in the quantum computer is given in Fig. 20. The performance of 32 operators A_2 is equivalent to 4 gates A_2 in the quantum circuit. As in the case of 3-qubit DFT, the permutations of outputs are performed inside each qubit by the NOT gate. It is considered that the diagonal matrix with the twiddle factors in the paired 16-point FFT is

$$\text{diag} \begin{Bmatrix} 1 \\ W_{16}^1 \\ W_{16}^2 \\ W_{16}^3 \\ -i \\ W_{16}^5 \\ W_{16}^6 \\ W_{16}^7 \end{Bmatrix} \oplus \text{diag} \begin{Bmatrix} 1 \\ W_8^1 \\ -i \\ W_8^3 \end{Bmatrix} \oplus \text{diag} \begin{Bmatrix} 1 \\ -i \end{Bmatrix} \oplus \text{diag} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}.$$

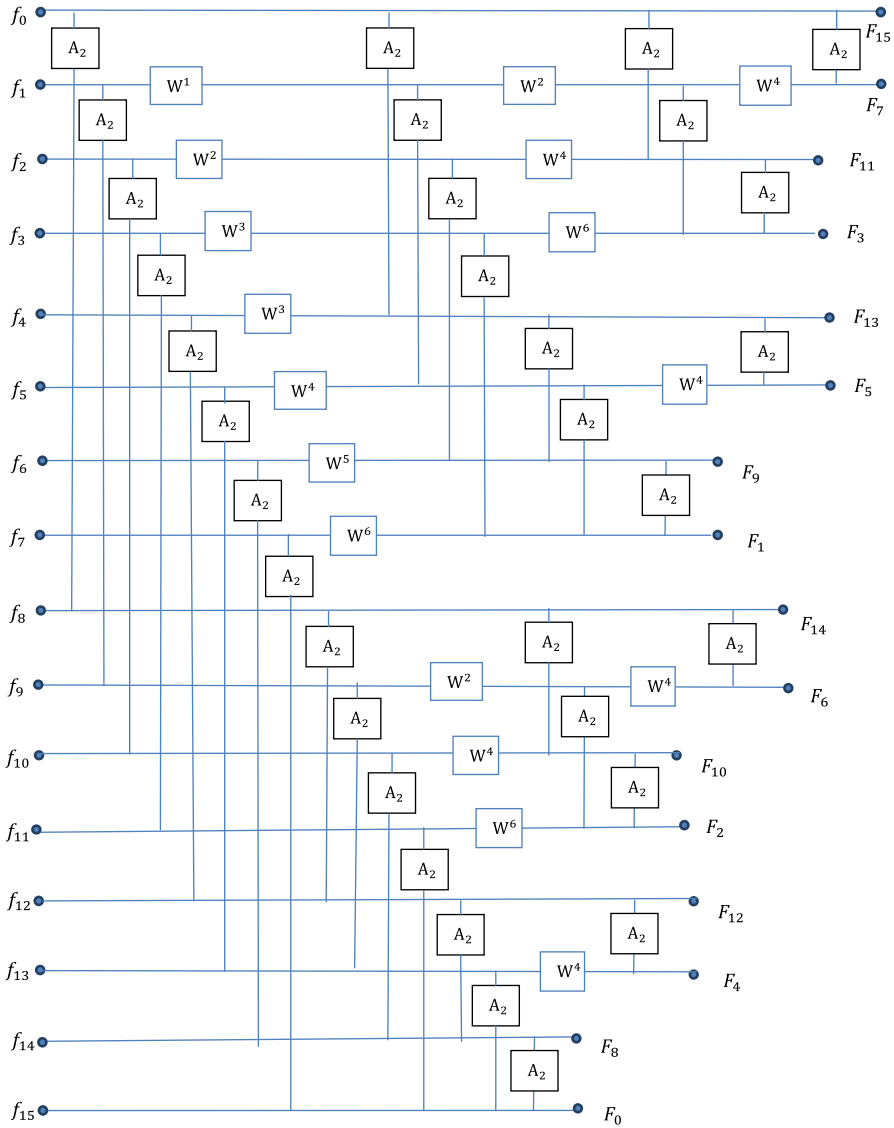


Fig. 19 Circuit for computing the 16-point FFT

Here, the first 8×8 diagonal sub-matrix can be written as

$$\text{diag} \begin{Bmatrix} 1 \\ -i \end{Bmatrix} \otimes \text{diag} \begin{Bmatrix} 1 \\ W_{16}^1 \\ W_{16}^2 \\ W_{16}^3 \end{Bmatrix} = T_1 \otimes T_2 \otimes T_3 \triangleq \text{diag} \begin{Bmatrix} 1 \\ -i \end{Bmatrix} \\ \otimes \text{diag} \begin{Bmatrix} 1 \\ W_8^1 \end{Bmatrix} \otimes \text{diag} \begin{Bmatrix} 1 \\ W_{16}^1 \end{Bmatrix} .$$

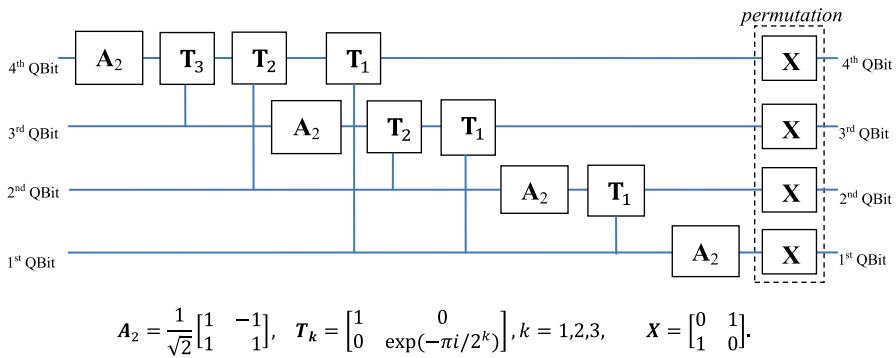


Fig. 20 The circuit for the 4-qubit direct DFT by the paired splitting

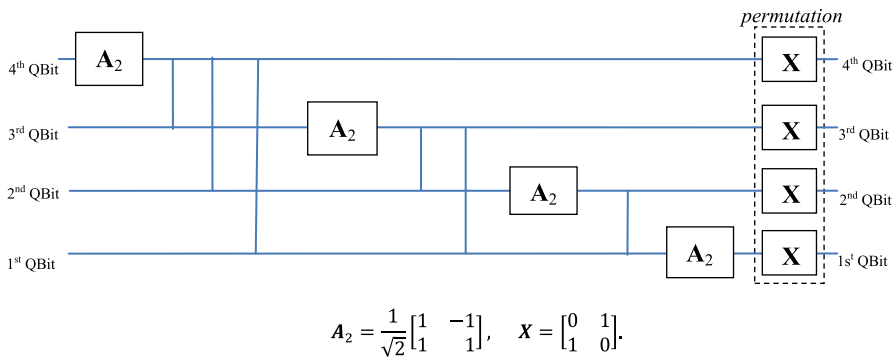


Fig. 21 The circuit for the 4-qubit direct DHT by the paired splitting

As it was done for the $N = 8$ case, the circuit in Fig. 20 can be reduced to the circuit for calculating the 16-point Hadamard, by removing all multiplications by the twiddle factors. For that, the same circuit can also be used, when setting all rotations matrices T_k to the 2×2 identity matrix. Figure 21 shows this circuit for calculating the 4-qubit DHT.

The equivalent circuit in the classical computer for calculating the 16-point fast Hadamard Transform (FHT) is shown Fig. 22. For high length $N = 2^r, r > 4$, the circuits for computing the N -point FFT and FHT by paired can be described in a similar way.

In the general case when $N = 2^r, r \geq 1$, a r -qubit state $|\check{f}\rangle$ as the N -dimensional vector of amplitudes is described as

$$|\check{f}\rangle = f_0|0\rangle + f_1|1\rangle + f_2|2\rangle + f_3|3\rangle + \dots + f_{2^{r-1}}|2^{r-1}\rangle, \tag{21}$$

with condition that $|f_0|^2 + |f_1|^2 + |f_2|^2 + \dots + |f_{2^{r-1}}|^2 = 1$. The components of the new state, which is called *the r-qubit DFT*,

$$F|\check{f}\rangle = F_0|0\rangle + F_1|1\rangle + F_2|2\rangle + F_3|3\rangle + \dots + F_{2^{r-1}}|2^{r-1}\rangle, \tag{22}$$

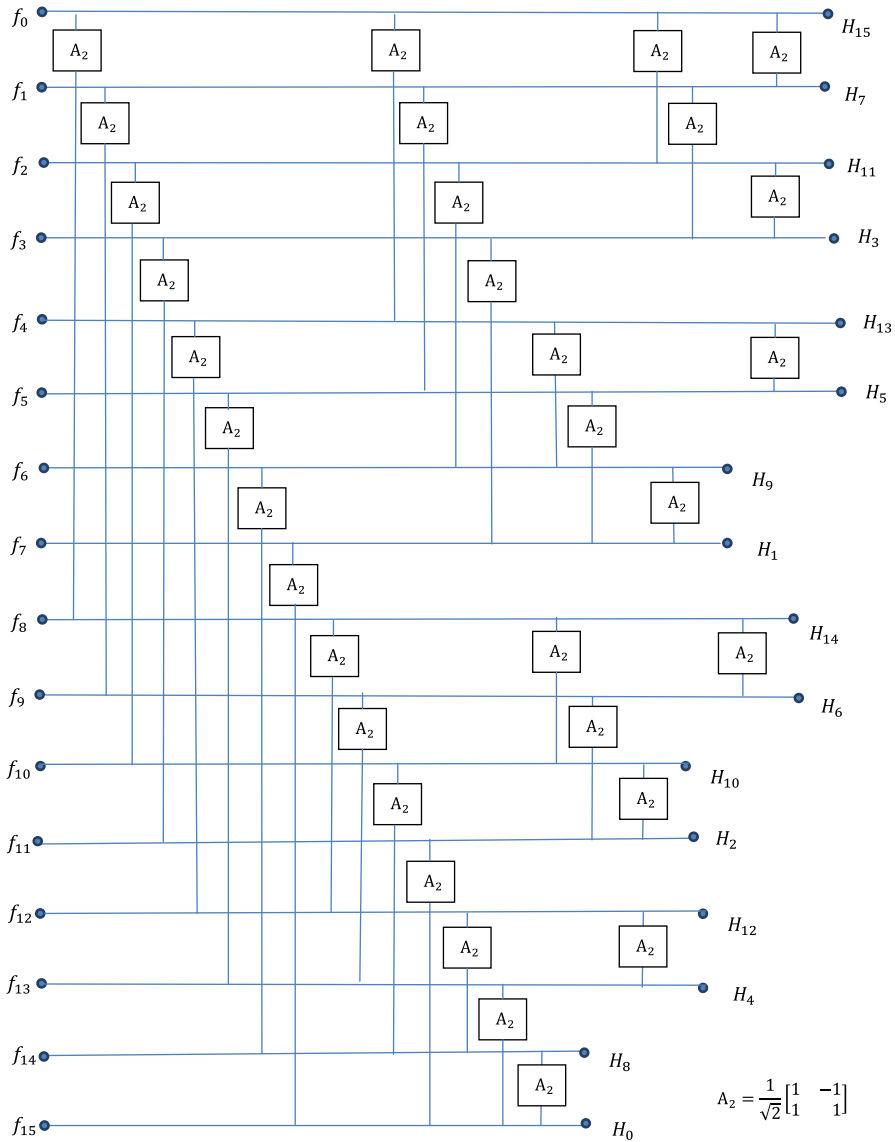


Fig. 22 Circuit for computing the quantum 16-point FHT

are defined as

$$F_p = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n W_N^{np} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f_n e^{-i2\pi np/N}, \quad p = 0, 1, \dots, (N - 1). \quad (23)$$

As stated in [16], the QFT performs the task different from the FFT that computes a new complex vector. The QFT changes the state of r qubits. If the result of measurement

$|\check{f}\rangle$ is $|n\rangle$ with probability $|f_n|^2$, the same value in the new state of the quantum DFT may occur, or can be measured with probability $|F_n|^2$. In Eq. 23, the normalized coefficient $1/\sqrt{N}$ is needed, in order to have the same condition $|F_0|^2 + |F_1|^2 + \dots + |F_{2^r-1}|^2 = 1$ for the new state $F|\check{f}\rangle$. Therefore, in all above circuits for calculating the r -qubit DFT and DHT, the matrix A_2 should be considered with the normalized coefficient $1/\sqrt{2}$.

5.1 Brief comparison with quantum circuits

The above analysis of the method was described in detail and compared with the first quantum circuit for the QFT [2], which is given in Fig. 1 for three qubits. The main difference is in using the gate A_2 instead of the Walsh–Hadamard gate H , and also in a simple permutation of the output qubits, by using only NOT X gates. The paired transform splits the DFT by a set of short DFTs which are processed separately. In contrast to such a splitting, the above QFT is based on the frequency-in-time algorithm and uses a number of butterflies over the short DFTs, which requires the sequence of SWAP gates.

Similar to the known method of approximation of the QFT with a bounded error, which is performed by removing a few rotation gates from the circuit (Coppersmith 1994), the proposed circuit could be used in a similar way to approximate the transform. We also mention four quantum circuits for the QFT that are described in [28]. One of these quantum circuits is the quantum circuit for the QFT [2] (which we used above for comparison with the paired QFT), but completed with the bit-reversal circuit. Other three complete circuits are similar to the first one; all four circuits are based on the same iteration formula (Eq. 20 in [28]) of the FFT but with different implementations, wherein the permutation between components of qubits is performed in the beginning or between the intermediate stages. In our opinion, such permutations do not make the circuits more effective than the first circuit, and moreover, the proposed circuit.

6 Conclusion

In this paper, the circuits for computing the length $N = 2^r$ fast Fourier and Hadamard transforms with splitting by the paired transform were described and analyzed in detail for the $N = 4, 8,$ and 16 cases. It was shown that the signal-flow graphs of the paired algorithms can be used for calculating the quantum Fourier and Hadamard transforms with minimum number of stages. The calculation of all components of the transforms is performed by the Hadamard gates and matrices of rotations and all simple NOT gates for permutation of output qubits. The author hopes that this paper will be useful for the advance of new-paired transform-based resourceful quantum algorithms.

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**(5), 1484–1509 (1997)

2. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information, 2nd edn. Cambridge University Press, Cambridge (2001)
3. Young R.C.D., Birch P.M., Chatwin C.R.: A simplification of the Shor quantum factorization algorithm employing a quantum Hadamard transform. In: Proceedings of SPIE 10649, Pattern Recognition and Tracking XXIX, 1064903, p. 11. Orlando, Florida, USA (2018)
4. Gong, L.H., He, X.T., Tan, R.C., Zhou, Z.H.: Single channel quantum color image encryption algorithm based on HSI model and quantum Fourier transform". Int. J. Theor. Phys. **57**, 59–73 (2018)
5. Yan, F., Iliyasa, A.M., Le, P.Q.: Quantum image processing: a review of advances in its security technologies. Int. J. Quantum Inf. **15**(3), 1730001 (2017)
6. Yan, F., Iliyasa, A.M., Venegas-Andraca, S.E.: A survey of quantum image representations. Quantum Inf. Process. **15**(1), 1–35 (2016)
7. Sang, J., Wang, S., Li, Q.: A novel quantum representation of color digital images. Quantum Inf. Process. **16**(2), 1–14 (2017)
8. Zhang, W.W., Gao, F., Liu, B., et al.: A watermark strategy for quantum images based on quantum Fourier transform. Quantum Inf. Process. **12**(2), 793–803 (2013)
9. Yang, Y.G., Jia, X., Xu, P., et al.: Analysis and improvement of the watermark strategy for quantum images based on quantum Fourier transform. Quantum Inf. Process. **12**(8), 2765–2769 (2013)
10. Coppersmith D.: An approximate Fourier transform useful in quantum factoring. Technical, Report RC19642, IBM (1994)
11. Chan, I.C., Ho, K.L.: Split vector-radix fast Fourier transform. IEEE Trans. Signal Process. **40**(8), 2029–2040 (1992)
12. Cheung D.: Using generalized quantum Fourier transforms in quantum phase estimation algorithms, Thesis. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.572.9698&rep=rep1&type=pdf>
13. Marquezinoa, F.L., Portugalá, R., Sasse, F.D.: Obtaining the quantum Fourier transform from the classical FFT with QR decomposition. J. Comput. Appl. Math. **235**(1), 74–81 (2014)
14. Barencó, A., Ekert, A., Suominen, K.-A., Törmä, P.: Approximate quantum Fourier transform and decoherence. Phys. Rev. A **54**, 139–146 (1996)
15. Yoran, N.N., Short, A.: Efficient classical simulation of the approximate quantum Fourier transform. Phys. Rev. A **76**, 042321 (2007)
16. Cleve R., Watrous J.: Fast parallel circuits for the quantum Fourier transform. In: Proceedings of IEEE 41st Annual Symposium on Foundations of Computer Science, pp. 526–536, Redondo Beach, CA, USA (2000)
17. Karafyllidis, I.G.: Visualization of the quantum Fourier transform using a quantum computer simulator. Quantum Inf. Process. **2**(4), 271–288 (2003)
18. Muthukrishnan, A., Stroud Jr., C.: Quantum fast fourier transform using multilevel atoms. J. Mod. Optics **49**, 2115–2127 (2002)
19. Heo, J., Kang, M.S., Hong, C.H., et al.: Discrete quantum Fourier transform using weak cross-Kerr nonlinearity and displacement operator and photon-number-resolving measurement under the decoherence effect. Quantum Inf. Process. **15**(12), 4955–4971 (2016)
20. Zilic, Z., Radecka, K.: Scaling and better approximating quantum fourier transform by higher radices. IEEE Trans. Comput. **56**(2), 202–207 (2007)
21. Grigoryan, A.M.: New algorithms for calculating discrete Fourier transforms. USSR Comput. Math. Math. Phys. **26**(5), 84–88 (1986)
22. Grigoryan, A.M.: An algorithm of computation of the one-dimensional discrete Fourier transform. Izvestiya VUZov SSSR, Radioelectronica **31**(5), 47–52 (1988)
23. Grigoryan, A.M.: 2-D and 1-D multi-paired transforms: frequency-time type wavelets. IEEE Trans. Signal Process. **49**(2), 344–353 (2001)
24. Grigoryan, A.M., Grigoryan, M.M.: Brief Notes in Advanced DSP: Fourier Analysis with MATLAB. CRC Press Taylor and Francis Group, Boca Raton (2009)
25. Grigoryan, A.M., Agaian, S.S.: Split manageable efficient algorithm for Fourier and Hadamard transforms. IEEE Trans. Signal Process. **48**(1), 172–183 (2000)
26. Grigoryan, A.M., Agaian, S.S.: Practical Quaternion and Octonion Imaging with MATLAB. SPIE Press, Bellingham (2009)
27. Browne, D.E.: Efficient classical simulation of the semi-classical quantum Fourier transform. New J. Phys. **9**, 146 (2007)
28. Li, H.S., Fan, P., Xia, H., Song, S., He, X.: The quantum Fourier transform based on quantum vision representation. Quantum Inf. Process. **17**, 333 (2018)

29. Agaian S.S., Klappenecker A.: Quantum computing and a unified approach to fast unitary transforms. In: Proceedings of SPIE 4667, Image Processing: Algorithms and Systems, p. 11 (2002)
30. Perez, L.R., Garcia-Escartin, J.C.: Quantum arithmetic with the quantum Fourier transform. *Quantum Inf. Process.* **16**, 14 (2017)
31. Maynard, C.E., Pius, E.: A quantum multiply-accumulator. *Quantum Inf. Process.* **13**(5), 1127–1138 (2014)
32. Grigoryan, A.M.: Two classes of elliptic discrete Fourier transforms: properties and examples. *J. Math. Imaging Vis.* **0235**(39), 210–229 (2011)
33. Grigoryan, A.M., Agaian, S.S.: Tensor transform-based quaternion Fourier transform algorithm. *Inf. Sci.* **320**, 62–74 (2015). <https://doi.org/10.1016/j.ins.2015.05.018>
34. Grigoryan A.M., S.S. Agaian S.S.: 2-D Octonion discrete Fourier transform: fast algorithms. In: Proceedings of IS&T International Symposium, Electronic Imaging: Algorithms and Systems XV, Burlingame, CA (2017)
35. Grigoryan A.M., Agaian S.S.: 2-D left-side quaternion discrete Fourier transform fast algorithms. In: Proceedings of IS&T International Symposium, 2016 Electronic Imaging: Algorithms and Systems XIV, San Francisco, California (2016)
36. Grigoryan, A.M., Agaian, S.S.: *Multidimensional Discrete Unitary Transforms: Representation, Partitioning, and Algorithms*. Marcel Dekker, New York (2003)
37. Agaian, S.S.: *Hadamard Matrices and Their Applications*, Lecture Notes in Mathematics, vol. 1168. Springer, New York (1985)
38. Agaian, S.S., Sarukhanyan, H.G., Egiazarian, K.O., Astola, J.: *Hadamard Transforms*. SPIE Press, Bellingham (2011)
39. Grigoryan, A.M.: An algorithm of computation of the one-dimensional discrete Hadamard transform. *Izvestiya VUZov SSSR Radioelectron. USSR Kiev* **34**(8), 100–103 (1991)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.