Chapter

# Quantum Fourier Operators and Their Application

*Eric Sakk*

## Abstract

The application of the quantum Fourier transform (QFT) within the field of quantum computation has been manifold. Shor's algorithm, phase estimation and computing discrete logarithms are but a few classic examples of its use. These initial blueprints for quantum algorithms have sparked a cascade of tantalizing solutions to problems considered to be intractable on a classical computer. Therefore, two main threads of research have unfolded. First, novel applications and algorithms involving the QFT are continually being developed. Second, improvements in the algorithmic complexity of the QFT are also a sought after commodity. In this work, we review the structure of the QFT and its implementation. In order to put these concepts in their proper perspective, we provide a brief overview of quantum computation. Finally, we provide a permutation structure for putting the QFT within the context of universal computation.

**Keywords:** quantum Fourier transform, quantum computation, quantum circuit, entanglement, unitary operators, permutation operators

## 1. Introduction

The quantum Fourier transform (QFT) has been applied in a number of different contexts within the field of quantum computation [1–3]. As this operator is central to so many quantum algorithms, a major thrust of current research is directed toward its efficient implementation [4–9]. The QFT calculation is, to a degree, based upon the discrete Fourier transform (DFT) where, given a discrete sequence

$$x = \{x_0, x_2, \cdots, x_{N-1}\} \tag{1}$$

of length $N$, the DFT of $x$ can be computed as

$$DFT\{x\} = Fx \tag{2}$$

with DFT matrix elements

$$F_{jk} = \frac{1}{\sqrt{N}} e^{i\frac{2\pi}{N}jk} \quad j,k = 0,1,\cdots,N-1 \tag{3}$$

Since the DFT matrix is $N \times N$, the computational complexity of computing $DFT\{x\}$ is $\mathcal{O}(N^2)$. If the input sequence length of the input sequence $x$ can be written as $N = 2^n$ (i.e. a power of two for some positive integer, $n$), there exist fast

Fourier transform (FFT) implementations that can compute $DFT\{x\}$ with $\mathcal{O}(N \log N)$ complexity. While there are other FFT implementations that do not require $N = 2^n$, the 'radix-2' implementation will be the starting point as it is relevant when introducing quantum computational bases. Before elevating the DFT to its quantum description, in Section 2 we will take a brief tour of quantum computation in order to provide some necessary context. We will then, in Section 3, develop the QFT operator and discuss its quantum implementation. Finally, in Section 4, we will discuss the QFT in the context of universal computation and its formulation in terms of permutation matrices.

## 2. Quantum computation

A starting point for quantum computation begins with choosing a qubit representation for the computational basis [3]

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{4}$$

This qubit basis forms a complete orthonormal set so that any single qubit quantum mechanical state can be written as the linear superposition

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{5}$$

where the coefficients $\alpha$ and $\beta$ are complex scalars. If $\langle\psi|$ represents the Hermitian conjugate of $|\psi\rangle$, according to quantum mechanics, the inner product

$$\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1 \tag{6}$$

is normalized so that $\psi$ represents a probability density function. This implies that, at any given instance in its time evolution, a quantum system can simultaneously be in the logical states $|0\rangle$ and $|1\rangle$ with their associated probabilities $|\alpha|^2$ and $|\beta|^2$. This is in stark contrast to classical digital computation whose operations must always exclusively evaluate to a value of either 0 or 1. Quantum computation allows an algorithm to simultaneously visit *both* logical states $|0\rangle$ and $|1\rangle$ of a single qubit. If $n$ qubits (i.e. multiple qubits) are applied, then a quantum system, in principal, has the potential to simultaneously visit $2^n$ logical states (again, with their associated probabilities). This exponential computational capacity is the source of quantum parallelism. However, there is a catch. Only when some observable is measured can we ascertain the current logical state of the system. Hence, quantum computers require large samples of measurements in order to build up the statistics necessary to determine the outcome of any given algorithm.

### 2.1 Unitary operators

The time evolution operator $U$ associated with a quantum system must be unitary meaning that

$$U^\dagger U = I \tag{7}$$

where $U^\dagger$ is the conjugate transpose of $U$. A major implication of this requirement is that the forward time system evolution must (at least mathematically) be

reversible. This requirement, in turn, constrains computations that are implemented by quantum operators to be ***reversible***. Therefore, logical operations such as AND, OR, and XOR (exclusive-or) would not be a quantum mechanical possibility unless some additional input information were to be preserved. This is because, in the absence of information about the input, measuring the output of these operations is not enough to ascertain the values of the inputs. Hence, these boolean processes, by themselves, are not reversible. However, there is a theory of reversible computation that can augment these logical operations so that input information is recoverable. Furthermore, much thought has gone into phrasing reversible computation in the context of unitary operators. Given the discussion so far, it is appropriate to give a short list of standard single qubit operators:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, R_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \tag{8}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{9}$$

The reader can check that these are all unitary. As a simple example of how to apply such operators, consider the action of $X$ on the basis vector $|0\rangle$

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \tag{10}$$

where $|0\rangle$ and $|1\rangle$ are 'swapped', indicating a form of logical inversion. $H$ is a Hadamard transform (i.e. a DFT for a sequence of length N=2). $X$, $Y$ and $Z$ are Pauli matrices. $R_\phi$ is a generalization of $Z = R_\pi$ and $I = R_0$. While these are single quhit operators, the next sections discuss how they can be extended to the multiple qubit case. Amazingly, this set of quantum operators can be applied to devise some very powerful quantum algorithms (e.g. QFT computation) [3, 10].

## 2.2 Tensor product (Kronecker product)

The Kronecker product of an $m \times n$ matrix $A$ with a $p \times q$ matrix $B$ is defined to be

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}. \tag{11}$$

Furthermore, assuming the dimensions are compatible for matrix multiplication, the following identity often proves useful

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{12}$$

for matrices $A, B, C, D$.

The computational basis can be extended to any number of qubits using the tensor product. For example, if two qubits are required for the computational space, using Eq. (2), the basis becomes

$$
\begin{aligned}
|0\rangle \equiv |00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
|1\rangle \equiv |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\
|2\rangle \equiv |10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \\
|3\rangle \equiv |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}
\end{aligned}
\tag{13}
$$

To generalize this example for $n$ qubits, the set of computational basis vectors can, for the sake of brevity, be labeled in base 10 as

$$\{|0\rangle, |1\rangle, |2\rangle, \cdots, |2^n - 1\rangle\}. \tag{14}$$

On the other hand, in order to highlight the qubit values, this basis can equivalently be expressed in base 2 as

$$|k_1 k_2 \cdots k_n\rangle = |k_1\rangle \otimes |k_2\rangle \otimes |\cdots \otimes |k_n\rangle \tag{15}$$

where $k_i \in \{0, 1\}$ for $i = 1, \cdots, n$. In other words, $\{k_1, k_2, \cdots, k_n\}$ represents the binary expansion

$$k = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_{n-1} 2^1 + k_n 2^0 = \sum_{t=1}^{n} k_t 2^{n-t} \tag{16}$$

for the $k^{th}$ basis vector $|k\rangle \equiv |k_1 k_2 \cdots k_n\rangle$. We have chosen this bit index ordering as it will prove convenient for the QFT formulation in the next section. An equally acceptable (and, quite typical) bit index convention for an $n$ qubit system could, for example, be $|q\rangle \equiv |q_{n-1} q_{n-2} \cdots q_1 q_0\rangle$.

Eq. (15) tells us that the $n$ qubit basis is derived from the tensor product of single qubits. This is important to keep in mind in order to avoid confusion when using the symbol $|0\rangle$. For example, when using $n = 1$ qubit, $|0\rangle$ in decimal is equivalent to $|0\rangle$ in binary; however,, when using $n = 3$ qubits, $|0\rangle$ in decimal is equivalent to $|000\rangle$ in binary. Hence, the number of qubits $n$ is the anchor for the relationship between Eq. (14) and Eq. (15). Assuming $n$ qubits, there are $2^n$ basis vectors that can be used to construct a quantum state. Hence, all $2^n$ basis vectors will simultaneously evolve with their associated probabilities; again, this is the source of quantum parallelism.

### 2.3 Quantum circuits

One particularly useful application of Eq. (12) arises when building up $n$ qubit quantum circuits (i.e. schematic depictions of quantum operations on qubits). For instance, assume a two qubit system $|q_1q_0\rangle$ where two unitary operators $H$ and $Z$ act on single qubits as

$$H|q_1\rangle, Z|q_0\rangle \tag{17}$$

and the result is desired to be combined as

$$H|q_1\rangle \otimes Z|q_0\rangle. \tag{18}$$

Eq. (12) tells us that this action is equivalent to

$$(H \otimes Z)\big(|q_1\rangle \otimes |q_0\rangle\big). \tag{19}$$

However, by construction, $|q_1\rangle \otimes |q_0\rangle = |q_1q_0\rangle$. Therefore,

$$H|q_1\rangle \otimes Z|q_0\rangle = (H \otimes Z)|q_1q_0\rangle \tag{20}$$

making it straightforward to develop multiple qubit quantum systems from unitary operators. The schematic representation of $(H \otimes Z)|q_1q_0\rangle$ is show in **Figure 1**.

With the groundwork laid for multiple qubits, it becomes possible to introduce more unitary operators that facilitate reversible computation. For example, the controlled NOT (CNOT) function can be phrased as a two qubit reversible XOR operator

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{21}$$

where $c$ represents the control bit, $t$ represents the target XOR function and $|q_1q_0\rangle = |ct\rangle$. This operator is a permutation matrix that is consistent with **Table 1** in that it swaps the $|11\rangle$ and $|10\rangle$ qubits. The XOR operation, by itself, can act as an irreversible controlled NOT operation. For the sake of quantum computation, the CNOT operator is unitary and a reversible XOR function is achieved because the control bit $|q_1\rangle$ is preserved from input to output.

There exist powerful tools for the simulation of quantum operations (referred to as '*quantum gates*') and for the rendering of multiple qubit quantum circuits [11].

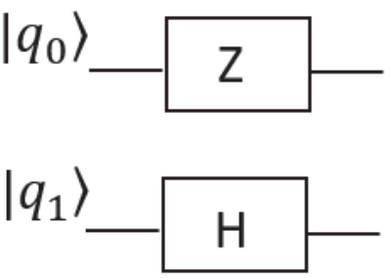


**Figure 1.**
*Two qubit quantum circuit for* $(H \otimes Z)|q_1q_0\rangle$.

| $c_{in}$ | $t_{in}$ | $c_{out}$ | $t_{out}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |

**Table 1.**
*Controlled NOT.*

**Figure 2** shows a schematic representation of the CNOT circuit corresponding to **Table 1**. In this circuit, the control bit is used to swap the target $|q_0\rangle$ (using an $X$ gate) if $|q_1\rangle = |1\rangle$.

For the sake of this work, we point out that an equally valid interpretation of the quantum CNOT function can be realized if the roles of the control and target are interchanged where $|q_1 q_0\rangle = |tc\rangle$ (see **Table 2**). In this case the CNOT operator becomes

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{22}$$

which is a permutation matrix that swaps the $|11\rangle$ and $|01\rangle$ qubits and corresponds to the circuit in **Figure 3**.

We shall have more to say about this implementation in the following sections. For now, with this brief overview of quantum computation, we can now introduce the quantum Fourier transform.

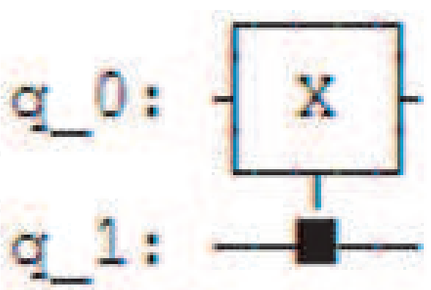


**Figure 2.**
*Two qubit CNOT quantum circuit swap of $|11\rangle$ and $|10\rangle$ using Qiskit [11].*

| $t_{in}$ | $c_{in}$ | $t_{out}$ | $c_{out}$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Table 2.**
*Controlled NOT where $|q_1 q_0\rangle = |tc\rangle$.*

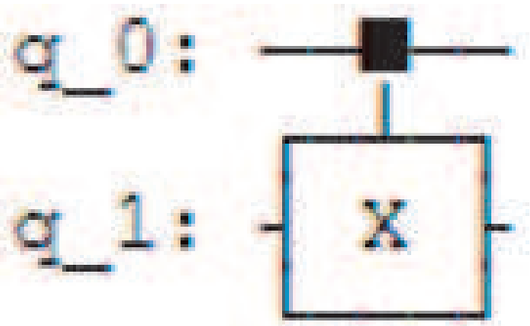


**Figure 3.**
*Two qubit CNOT quantum circuit swap of* $|11\rangle$ *and* $|01\rangle$.

## 3. The quantum Fourier transform

It should be clear that the DFT matrix in Eq. (2) is unitary where

$$F^{\dagger}F = I \tag{23}$$

and $F^{\dagger}$ is the Hermitian conjugate of $F$. Because of this unitarity, the potential for using the DFT within the context of quantum computation naturally follows. However, such an application requires a decomposition involving tensor products of unitary operations typically applied in quantum computation. As with the FFT, the choice of the decomposition dictates the algorithmic complexity. There is much introductory literature available regarding the QFT [3, 12–14]. Given a specific quantum algorithm where the QFT is applied, current research endeavors reside in attempts to improve the computational complexity [4, 7, 9, 15, 16].

The QFT matrix is defined as

$$Q = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}jk} |k\rangle\langle j|. \tag{24}$$

For example, with $N = 2^n$ and n = 1, we recover the Hadamard matrix

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{25}$$

or, for n = 2,

$$Q = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & 1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}. \tag{26}$$

As expected, this operator is unitary where, with

$$Q^{\dagger} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} e^{-i\frac{2\pi}{N}jk} |j\rangle\langle k|, \tag{27}$$

it should be clear that

$$\begin{aligned}
QQ^{\dagger} &= \left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}jk}|k\rangle\langle j|\right)\left(\frac{1}{\sqrt{N}}\sum_{j'=0}^{N-1}\sum_{k'=0}^{N-1} e^{-i\frac{2\pi}{N}j'k'}|j'\rangle\langle k'|\right) \\
&= \frac{1}{N}\sum_{k'=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j'=0}^{N-1}\sum_{j=0}^{N-1} e^{i\frac{2\pi}{N}\left(jk-j'k'\right)}|k\rangle\langle j|j'\rangle\langle k'| \\
&= \frac{1}{N}\sum_{k'=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j'=0}^{N-1}\sum_{j=0}^{N-1} e^{i\frac{2\pi}{N}\left(jk-j'k'\right)}\delta_{j'j}|k\rangle\langle k'| \\
&= \frac{1}{N}\sum_{k'=0}^{N-1}\sum_{k=0}^{N-1}\sum_{j=0}^{N-1} e^{i\frac{2\pi}{N}j\left(k-k'\right)}|k\rangle\langle k'| \\
&= \frac{1}{N}\sum_{k'=0}^{N-1}\sum_{k=0}^{N-1}(N\delta_{k'k})|k\rangle\langle k'| \\
&= \sum_{k=0}^{N-1}|k\rangle\langle k| \\
&= I.
\end{aligned} \tag{28}$$

In general, given a state vector

$$|\psi\rangle = \sum_{j=0}^{N-1} a_j|j\rangle \tag{29}$$

the QFT operates on $|\psi\rangle$ to form

$$\begin{aligned}
|\Psi\rangle = QFT\{|\psi\rangle\} &= QFT\left\{\sum_{j=0}^{N-1} a_j|j\rangle\right\} \\
&= \sum_{j=0}^{N-1} a_j QFT\{|j\rangle\} \\
&= \sum_{j=0}^{N-1} a_j Q|j\rangle.
\end{aligned} \tag{30}$$

Given this result, let us consider the QFT of a single $n$ qubit basis vector $|j\rangle$ where $N = 2^n$. First, observe that while

$$\begin{aligned}
QFT\{|j\rangle\} = Q|j\rangle &= \frac{1}{2^{n/2}}\sum_{j'=0}^{N-1}\sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}j'k}|k\rangle\langle j'|j\rangle \\
&= \frac{1}{2^{n/2}}\sum_{j'=0}^{N-1}\sum_{k=0}^{N-1} e^{i\frac{2\pi}{N}j'k}|k\rangle\delta_{j'j} \\
&= \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1} e^{i\frac{2\pi}{2^n}jk}|k\rangle,
\end{aligned} \tag{31}$$

given Eqs. (15) and (16), it will be more helpful to express this relation as

$$\begin{aligned} QFT\{|j\rangle\} &= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} e^{i2\pi j\left(\sum_{t=1}^{n} k_t 2^{-t}\right)} |k_1 k_2 \cdots k_n\rangle. \\ &= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} e^{i2\pi j\left(k_1 2^{-1} + k_2 2^{-2} + \cdots + k_{n-1} 2^{-(n-1)} + k_n 2^{-n}\right)} |k_1 k_2 \cdots k_n\rangle. \\ &= \frac{1}{2^{n/2}} \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} e^{i2\pi j\left(k_1 2^{-1} + k_2 2^{-2} + \cdots + k_{n-1} 2^{-(n-1)} + k_n 2^{-n}\right)} |k_1\rangle \otimes |k_2\rangle \otimes |\cdots \otimes |k_n\rangle. \\ &= \frac{1}{2^{n/2}} \bigotimes_{v=1}^{n} \sum_{k_v=0}^{1} e^{i2\pi j(k_v 2^{-v})} |k_v\rangle. \end{aligned} \tag{32}$$

This leads to the result that

$$Q|j\rangle = \frac{1}{2^{n/2}} \bigotimes_{v=1}^{n} (|0\rangle + e^{i2\pi j 2^{-v}} |1\rangle). \tag{33}$$

### 3.1 QFT qubit representation

To forge a path toward efficient implementation, it is important to recognize how Eq. (33) can be decomposed into a set of operators relevant to quantum computation (see Section 2.1). First, consider the $n = 1$ single qubit case,

$$Q|j\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i\frac{2\pi j}{2}} |1\rangle\right). \tag{34}$$

Then, for each qubit state $|j\rangle = |0\rangle, |1\rangle$, it follows that

$$\begin{aligned} Q|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ Q|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{aligned} \tag{35}$$

as expected since $Q = H$ for the single qubit case. Hence, it should be nn surprise that the $v = 1$ contribution to Eq. (10) should be a Hadamard gate.

To handle the phase factors in the other contributions to the tensor product (where $v \geq 2$), the keen eye will recognize that the terms $e^{i2\pi j 2^{-v}}$ could lead to a unitary quantum mechanical operator. Before leveraging this observation in a QFT algorithm, it will be helpful to consider the qubit representation $|j\rangle = |j_1 j_2 \cdots j_n\rangle$. As the index $v$ ranges from 1 to $n$, the index $j$ in the term $e^{i2\pi j 2^{-v}}$ experiences successive divisions by 2 (i.e. successive right shifts of its binary representation by one bit):

$$\begin{aligned} v = 1: &\quad j2^{-1} &\Rightarrow &\quad j_1 j_2 \cdots j_{n-1} \cdot j_n \\ v = 2: &\quad j2^{-2} &\Rightarrow &\quad j_1 j_2 \cdots j_{n-2} \cdot j_{n-1} j_n \\ &\quad \vdots & & \\ v = n-1: &\quad j2^{-(n-1)} &\Rightarrow &\quad j_1 \cdot j_2 \cdots j_{n-1} j_n \\ v = n: &\quad j2^{-n} &\Rightarrow &\quad 0 . j_1 j_2 \cdots j_{n-1} j_n \end{aligned} \tag{36}$$

Since these values appear in the phase factor, the integer parts will only result in integer multiples of $2\pi$ and can therefore be discarded. Eq. (33) can then be expressed as

$$QFT\{|j\rangle\} = \frac{1}{2^{n/2}}[\left(|0\rangle + e^{i2\pi j_n 2^{-1}}|1\rangle\right) \otimes \left(|0\rangle + e^{i2\pi\left(\ j_{n-1}2^{-1}+j_n 2^{-2}\right)}|1\rangle\right) \otimes \cdots \\ \cdots \otimes \left(|0\rangle + e^{i2\pi\left(\ j_1 2^{-1}+j_2 2^{-2}+\cdots+j_n 2^{-n}\right)}|1\rangle\right). \tag{37}$$

It is often this version of the QFT that is used as a starting point for quantum circuit implementation when $N = 2^n$ [3].

As an example, consider the two qubit case where $n = 2$ and $|j\rangle = |j_1 j_2\rangle$, then

$$\begin{aligned} Q|j\rangle &= Q|j_1 j_2\rangle \\ &= \frac{1}{2}\left(|0\rangle + e^{i2\pi j_2 2^{-1}}|1\rangle\right) \otimes \left(|0\rangle + e^{i2\pi\left(\ j_1 2^{-1}+j_2 2^{-2}\right)}|1\rangle\right) \end{aligned} \tag{38}$$

If we let $|j_1 j_2\rangle = |01\rangle$, then

$$\begin{aligned} Q|j\rangle &= Q|01\rangle \\ &= \frac{1}{2}\left(|0\rangle + e^{i2\pi(1)2^{-1}}|1\rangle\right) \otimes \left(|0\rangle + e^{i2\pi\left((0)2^{-1}+(1)2^{-2}\right)}|1\rangle\right) \\ &= \frac{1}{2}(|0\rangle - |1\rangle) \otimes (|0\rangle + i|1\rangle) \\ &= \frac{1}{2}(|00\rangle + i|01\rangle - |10\rangle - i|11\rangle) \end{aligned} \tag{39}$$

which corresponds to the column $|01\rangle$ entries in Eq. (26). If not already obvious, it should be emphasized that the tensor product is ***not commutative*** and that consistent qubit ordering is instrumental to the success of this calculation.

### 3.2 Quantum implementation

Based upon Eq. (37), it is sensible to introduce an iterable version of the $R$ operator introduced in Section 2.1:

$$R_v = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i2\pi}{2^v}} \end{bmatrix}. \tag{40}$$

Furthermore, because each qubit contribution contains phase terms involving the binary expansion of $j$, one approach to addressing these interactions is to introduce a controlled version of $R_v$:

$$CR_v = \begin{bmatrix} I & 0 \\ 0 & R_v \end{bmatrix}. \tag{41}$$

This operator can be used to induce the correct phase factor as follows. Assume $tc\rangle$ is the target/control structure for single qubits $j_r j_s$ were $s > r$ in the binary representation of $|j\rangle$. Then, the following holds true

$$\begin{aligned} CR_v|j_r 0\rangle &= |j_r 0\rangle \\ CR_v|j_r 1\rangle &= e^{\frac{i2\pi}{2^v} j_r}|j_r 1\rangle \end{aligned} \tag{42}$$

Hence, the control bit determines when to introduce the phase factor involving the target bit.

The goal of this section is to introduce enough nomenclature in order to put the next section of this work in context. The reader is encouraged to visit the provided references in order to fill in the details of a generalized quantum circuit that can implement an $n$ qubit QFT. For now, we provide an $n = 2$ qubit example to illustrate an algorithm for performing the QFT. Whatever principled series of operations is chosen, the goal of the quantum algorithm (and, hence, the associated quantum circuit) is to reproduce Eq. (11). Starting with $|j\rangle = |j_1 j_2\rangle$,

a. Apply $H$ to $|j_1\rangle$ so that

$$\begin{aligned} |j_1\rangle \otimes |j_2\rangle &\rightarrow H|j_1\rangle \otimes |j_2\rangle \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j_1 2^{-1}}|1\rangle\right) \otimes |j_2\rangle \end{aligned} \tag{43}$$

b. Apply $CR_2$ to target qubit $j_1$ controlled by $j_2$. This yields

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j_1 2^{-1}}|1\rangle\right) \otimes |j_2\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\, j_2 2^{-2} + j_1 2^{-1}\right)}|1\rangle\right) \otimes |j_2\rangle \tag{44}$$

c. Apply $H$ to $|j_2\rangle$

$$\begin{aligned} \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\, j_2 2^{-2} + j_1 2^{-1}\right)}|1\rangle\right) \otimes |j_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi\left(\, j_2 2^{-2} + j_1 2^{-1}\right)}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i2\pi j_2 2^{-1}}|1\rangle\right) \\ &= \frac{1}{2}\left(|0\rangle + e^{\frac{i2\pi j}{2^2}}|1\rangle\right) \otimes \left(|0\rangle + e^{\frac{i2\pi j}{2^1}}|1\rangle\right) \end{aligned} \tag{45}$$

Comparing this result with either Eq. (33) or Eq. (37), it is clear that this algorithm, derived using quantum reversible operators, recovers the QFT from Eq. (38) with one slight difference: the bit ordering is reversed. Given $n$ qubits, it is possible to apply $n/2$ swaps using, for example, tensor products involving an $X$ operator (see Section 2.1) in order to reverse the bit order. Such bit reversal permutations are reminiscent of the radix-2 FFT algorithm. If one generalizes this algorithm to $n$ qubits, it can be shown that the algorithmic complexity is $\mathcal{O}(n^2)$. With $N = 2^n$, this is a considerable improvement over $N \log N = n2^n$ for the radix-2 FFT. However, algorithmic improvements and variations have been developed that can further reduce QFT complexity to $\mathcal{O}(n \log n)$ [9, 15].

## 4. QFT permutations

Universal computation, by its very nature, must involve some set of permutation operators [17–20]. As with other universal gates applied in quantum computation, in this section, we show that the QFT can generate operators that have the properties of a permutation. Consider a successive application of the QFT such as $Q^2 = QQ$ and let us analyze the matrix elements of such an operation:

$$
\begin{aligned}
[QQ]_{j,k} &= \frac{1}{N}\sum_{m=0}^{N-1}\left(e^{i\frac{2\pi}{N}jm}|j\rangle\langle m|\right)\left(e^{i\frac{2\pi}{N}mk}|m\rangle\langle k|\right) \\
&= \frac{1}{N}\sum_{m=0}^{N-1}e^{i\frac{2\pi}{N}m(\ j+k)}\langle m|m\rangle|j\rangle\langle k| \\
&= \begin{cases} 0 & j+k\neq 0 \mod N \\ 1 & j+k=0 \mod N \end{cases} \\
&\equiv \left[P_{Q^2}\right]_{j,k}.
\end{aligned}
\tag{46}
$$

For an $n$ qubit system $|q_{n-1}\cdots q_1 q_0\rangle$, it should be clear that $P_{Q^2}$ is a permutation operator that leaves the position of $|q_0\rangle$ unchanged and inverts the order of the remaining qubits to form $|q_1\cdots q_{n-1}q_0\rangle$. For example, the CNOT operator in Eq. (22) is equal to $P_{Q^2}$ for $n=2$

$$
CNOT = Q^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = P_{Q^2} \tag{47}
$$

having properties similar to that of a Sylvester shift matrix (i.e. a generalization of a Pauli matrix). It is sensible that a CNOT operation followed by a CNOT operation should result in the identity operation and, hence, that $P_{Q^2}P_{Q^2} = Q^4 = I$ (i.e. a double inversion recovers the original qubit sequence). These results can be generalized for any $n$. For example, with $n=3$, Eq. (46) becomes

$$
P_{Q^2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{48}
$$

which, after the appropriate sequence of swaps, can be transformed into a Toffoli (CCNOT) gate. Hence, $P_{Q^2}$ can be thought of as a generalization of swap permutation operators and the QFT can be phrased as its square root. For example, it is common to define a two qubit swap operator as

$$
S_w = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{49}
$$

along with its square root

$$\sqrt{S_w} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1+i) & 0 \\ 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1+i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{50}$$

In a similar manner, Eq. (46) leads us to the following

**Theorem 1** *Given the $N \times N$ inversion permutation matrix defined as*

$$\left[P_{Q^2}\right]_{j,k} = \begin{cases} 0 & j+k \neq 0 \mod N \\ 1 & j+k = 0 \mod N \end{cases}, \tag{51}$$

it follows that

$$Q = \sqrt{P_{Q^2}} \tag{52}$$

where $Q$ is a QFT matrix.

In addition, given that $Q^4 = I$ we have the following

**Corollary 1** Any algorithm that iteratively applies the QFT can result in only one of the following outcomes

a. $Q^k = \sqrt{P_{Q^2}}$ if $k = 1 \mod 4$.

b. $Q^k = P_{Q^2}$ if $k = 2 \mod 4$.

c. $Q^k = Q^{-1}$ if $k = 3 \mod 4$.

d. $Q^k = I$ if $k = 0 \mod 4$.

These results indicate a deeper connection between universal computation, permutations and the QFT. Furthermore, decomposing the QFT calculation into a product of permutations indicates a potential for reducing the computational complexity of QFT implementations.

## 5. Conclusions

In this work, we have revisited the quantum Fourier transform which is central to many algorithms applied in the field of quantum computation. As a natural extension of the discrete Fourier transform, the QFT can be implemented using efficient tensor products of quantum operators. Part of the thrust of current research deals with reducing the QFT computational complexity. With this goal in mind, we have phrased the QFT as a permutation operator. Future research will be directed toward quantum circuit implementation using QFT permutation operators within the context of universal computation.

## Acknowledgements

This research is funded by a grant from the National Science Foundation NSF #1560214.

## Author details

Eric Sakk
Morgan State University, Baltimore, MD, USA

*Address all correspondence to: eric.sakk@morgan.edu

IntechOpen

## References

[1] Shor, PW.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput., 1997; 26: 1484–1509.

[2] Josza, R.: Quantum Algorithms and the Fourier Transform. Proc. R. Soc. Lond. A, 1998; 454:323–337.

[3] Nielsen, MA., Chuang, IL.: Quantum Computation and Quantum Information. Cambridge University Press. 2011.

[4] Barenco, A., Ekert, A., Suominen, KA., Torma, P. : Approximate quantum Fourier transform and decoherence. Phys. Rev. A, 1996; 54.

[5] Fowler, A., Hollenberg, LCL. : Scalability of Shor's algorithm with a limited set of rotation gate. Phys. Rev. A, 2004; 70.

[6] Pavlidis,A., Gizopoulos, D.: Fast Quantum Modular Exponentiation Architecture for Shor's Factorization Algorithm. Quantum Information and Computation, 2014; 14.

[7] Prokopenya,AN.: Approximate Quantum Fourier Transform and Quantum Algorithm for Phase Estimation. International Workshop on Computer Algebra in Scientific Computing, 2015; 391–405.

[8] Ruiz-Perez, L., Garcia-Escartin, JC.: Quantum arithmetic with the quantum Fourier transform. Quantum Inf. Process., 2017; 16.

[9] Nam, Y., Su, Y., Maslov, D.: Approximate quantum Fourier transform with O(n log(n)) T gates. NPJ Quantum Information, 2020; 6(26).

[10] Barenco,A., Bennett,CH., Cleve, R., DiVincenzo,DP., Margolus, N., Shor,P., Sleator,T., Smolin,J.A., Weinfurter, H. : Elementary gates for quantum computation. Phys. Rev. A, 1995; 52.

[11] Open-Source Quantum Development. https://qiskit.org/ [Accessed: 1 September 2020]

[12] Quantum Fourier Transform. https://qiskit.org/textbook/ch-algorithms/quantum-fourier-transform.html [Accessed: 1 September 2020]

[13] QC - Quantum Computing Series. https://medium.com/@jonathan_hui/qc-quantum-computing-series-10ddd7977abd [Accessed: 1 September 2020]

[14] Camps, D., Van Beeumen, R., Yang, C.: Quantum Fourier Transform Revisited.Numerical Linear Algebra with Applications. 2020.

[15] Hales,L.,Hallgren,S.: An Improved Quantum Fourier Transform Algorithm and Applications. Proceedings 41st Annual Symposium on Foundations of Computer Science, 12-14 Nov. 2000, Redondo Beach, CA, USA.

[16] Wang, SP., Sakk, E.: Quantum Algorithms: Overviews, Foundations, and Speedup. ICCSP 2021, Zhuhai, China; January 8-10, 2021.

[17] DiVincenzo,DP.: Two-bit gates are universal for quantum computation. Phys. Rev. A, 1995. 51:1015–1022.

[18] Planat,M., Ul Haq,R.: The Magic of Universal Quantum Computing with Permutations. Advances in Mathematical Physics, 2020.

[19] de Almeida,AAA., Dueck,GW., daSilva,ACR.: CNOT Gate Optimizations via Qubit Permutations. Journal of Low Power Electronics, 2019; 15:182–192.

[20] Ouyangab,Y.,Shen,Y.,Chen,L.: Faster quantum computation with permutations and resonant couplings. Linear Algebra and its Applications, 2020; 592:270–286.