Quantum Algorithm based on Quantum Fourier Transform for Register-by-Constant Addition

Filipe Chagas Ferraz
Federal University of Mato Grosso (UFMT)
email: filipe.ferraz@sou.ufmt.br

July, 2022

Abstract

Since Shor's [8] proposition of the method for factoring products of prime numbers using quantum computing, there has been a quest to implement efficient quantum arithmetic algorithms. These algorithms are capable of applying arithmetic operations simultaneously on large sets of values using quantum parallelism. Draper [9] proposed an addition algorithm based on the quantum Fourier transform whose operands are two quantum registers, which I refer to as register-by-register addition. However, for cases where there is the need to be added a constant value to a target register, Draper's algorithm is more complex than necessary in terms of number of operations (*depth*) and number of qubits used. In this paper, I present a more efficient addition algorithm than Draper's for cases where there needs to be added just a constant to a target register.

1 Introduction

Quantum computing is an emerging area of technology that addresses techniques for solving computational problems using principles of quantum mechanics. Although there has been a surge of interest and industry activity in quantum computing [14], it is not recent that this area has started to be explored. In early 80s, Benioff introduced the quantum version of Turing machines [1, 2]. Also in early 80s, the possibility of using quantum machines to simulate quantum systems was addressed by Feynman [3, 12]. In 1985, Deutsch presented a universal quantum computer model capable of simulating finite physical systems and performing fast probabilistic tasks using quantum parallelism [4]. In the early 90s, Deutsch showed a category of problems that could be solved by quantum computers faster than classical ones [5]. Other notable pioneer contributions to the field were made by Coppersmith [11], Shor [8], Vedral et al. [7] and Draper [9]. Coppersmith [11] proposed the quantum Fourier transform (QFT), an implementation of the discrete Fourier transform for quantum computers, and Shor [8] proposed a method to factor the product of two prime numbers in polynomial time using quantum computing. The proposition of Shor's method initiated the search for efficient quantum algorithms for performing arithmetic operations such as addition, subtraction, multiplication, and exponentiation with natural numbers stored in quantum registers. The paper by Vedral et al. [7] is the first one to propose quantum algorithms for computing the arithmetic operations aforementioned with natural numbers stored in quantum registers using ripple-carry schemes. Later, Draper [9] proposed a quantum algorithm for summing integers stored in two quantum registers based on the quantum Fourier transform. Draper's [9] algorithm is a register-by-register adder, that is, both operands of the addition are quantum registers. When it is necessary to add just a constant to the natural values of a target quantum register – and by "constant" I mean an operand

that does not have multiple superposed values – the algorithms of Vedral et al. [7] and Draper [9] are more complex than necessary in terms of the amount of operations (*depth*) and the use of qubits. For this reason, I present in this paper a register-by-constant addition algorithm based on the quantum Fourier transform.

This paper is organized as follows:

- Section 2 briefly adresses the register-by-register addition algorithm proposed by Draper [9] and explains why this algorithm is more complex than necessary in cases where the intention is to add just a constant to the target register.

- Section 3 presents the QFT-based register-by-constant addition algorithm: an addition algorithm based on the quantum Fourier transform (QFT) specific for cases in which there needs to add just a constant to the target register, and less complex than the algorithm proposed by Draper [9].

- Section 4 presents a mathematical proof of correctness of the register-by-constant addition algorithm presented here.

- Section 5 shows an alternative representation of the Draper adder using the register-by-constant addition operator in the Fourier basis.

- Finally, section 6 makes some final considerations and concludes the article.

2 Draper's register-by-register addition algorithm

Draper [9] proposed a register-by-register addition algorithm based on the quantum Fourier transform. Draper's addition algorithm performs as in the following mapping:

$$|a, b\rangle \mapsto |a, b + a \pmod{2^N}\rangle \qquad (1)$$

In the equation 1, a and b are natural numbers. $|a\rangle$ and $|b\rangle$ are states each composed by N qubits, i.e., $|a\rangle = |a_N, ..., a_2, a_1\rangle$ and $|b\rangle = |b_N, ..., b_2, b_1\rangle$

such that $a_j \in \{0, 1\}$ and $b_j \in \{0, 1\}$ for any $j \in \{1, ..., N\}$.

The capacity for parallel processing of this algorithm is seen when the operand registers have multiple superposed values. For a pair of registers initialized as $|\psi_1\rangle = \sum_{j=0}^{2^N-1} \alpha_j |j\rangle$ and $|\psi_2\rangle = \sum_{k=0}^{2^N-1} \beta_k |k\rangle$, the Draper's adder (represented by D) operates as shown in the following equation:

$$D(|\psi_1\rangle \otimes |\psi_2\rangle) =$$

$$D\left[\left(\sum_{j=0}^{2^N-1} \alpha_j |j\rangle\right) \otimes \left(\sum_{k=0}^{2^N-1} \beta_k |k\rangle\right)\right] =$$

$$D\left(\sum_{j=0}^{2^N-1} \sum_{k=0}^{2^N-1} \alpha_j \beta_k |j, k\rangle\right) = \qquad (2)$$

$$\sum_{j=0}^{2^N-1} \sum_{k=0}^{2^N-1} \alpha_j \beta_k D|j, k\rangle =$$

$$\sum_{j=0}^{2^N-1} \sum_{k=0}^{2^N-1} \alpha_j \beta_k |j, j+k \pmod{2^N}\rangle$$

Equation 2 shows that when Draper's adder is applied over a pair of registers initialized with multiple superposed values, the result in the second register is a superposition of all possible modular sums $j + k \pmod{2^N}$, each with probability amplitude $\alpha_j \beta_k$.
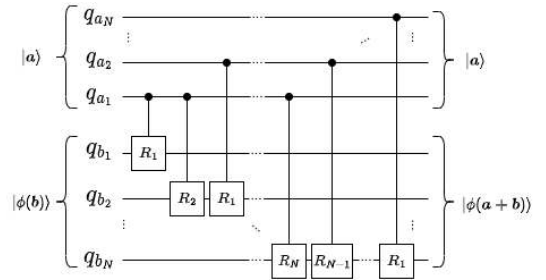

Figure 1: Quantum circuit of the Draper's addition algorithm

The structure of the Draper's adder consists of a sequence of $\frac{N(N+1)}{2}$ controlled R_l operations

between a QFT and an inverse QFT, where N is the number of qubits per operand and R_l is a unitary single-qubit operator defined as:

$$R_l = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^l}} \end{bmatrix} \qquad (3)$$

When there needs to add a constant to a target register of N qubits, i.e., to transform the state $\sum_{k=0}^{2^N-1} w_k |k\rangle$ of the second register into $\sum_{k=0}^{2^N-1} w_k |k + c \pmod{2^N}\rangle$ for $c \in \mathbb{Z}$, Draper's adder requires at least $\lceil \log_2 c \rceil$ additional qubits for storing the value c. In section 3, I present a QFT-based adder with fewer operations and that does not require additional qubits to add a constant to a target register.

3 Proposed register-by-constant addition algorithm

The proposed register-by-constant addition algorithm operates on a single target quantum register, eliminating the need for a second register for storing the constant value. The operation performed by this algorithm is defined for N qubits as the mapping $|a\rangle \mapsto |a + c \pmod{2^N}\rangle$, where a is the natural value stored in the target register before the operation, and c is the constant added to the target register after the operation. A circuit-like representation of this algorithm is presented in figure 2, and a pseudocode implementation is presented in algorithm 1.
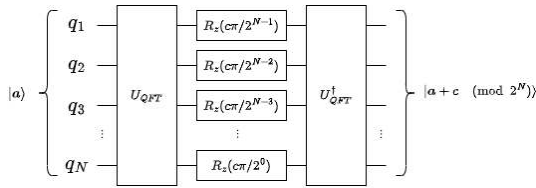

Figure 2: Quantum circuit of the proposed register-by-constant addition algorithm

Figure 2 shows the quantum register-by-constant addition circuit applied over a N-qubits register $(q_1, q_2, ..., q_N)$ initialized as $|a\rangle$. In a situation where the register is not initialized with multiple superposed values, $|a\rangle$ can be defined as $|a\rangle = |a_N, ..., a_2, a_1\rangle$ such that $a = \sum_{k=1}^{N} 2^{k-1} a_k$ and $a_k \in \{0, 1\}$. Each qubit q_k stores the respective a_k (note that the register in the circuit in figure 2 is big-endian).

Algorithm 1: Implementation in pseudocode of the register-by-constant addition algorithm

---

Data: $N \in \mathbb{N} - \{0\}$ ▷ Number of target qubits

Data: $c \in \mathbb{Z}$ ▷ Constant

Data: $q_1, q_2, ..., q_N$ ▷ Target qubits

$U_{\text{QFT}}(\text{target} = [q_t]_{t=1}^N)$ ▷ Applies a QFT over all target qubits;

for $t \in [1, ..., N]$ **do**

   $R_z(\theta = \frac{c\pi}{2^{N-t}}, \text{target} = q_t)$ ▷ Applies a $R_z(\theta)$ gate over q_t ;

end

$U_{\text{QFT}}^\dagger(\text{target} = [q_t]_{t=1}^N)$ ▷ Applies a IQFT over all target qubits;

The register-by-constant addition algorithm can be represented as the following unitary linear operator:

$$U_+(c) = U_{\text{QFT}}^\dagger \times U_{\phi(+)}(c) \times U_{\text{QFT}} \qquad (4)$$

The operators U_{QFT} and $U_{\text{QFT}}^\dagger$ are, respectively: the quantum Fourier transform and its inverse; and the operator $U_{\phi(+)}(c)$ is the register-by-constant adder for a constant c in Fourier basis. The operator $U_{\phi(+)}(c)$ is defined for N qubits as:

$$U_{\phi(+)}(c) = \bigotimes_{\tau=1}^{N} R_z \left(\frac{c\pi}{2^{N-t}} \right)$$
$$= R_z \left(\frac{c\pi}{2^{N-N}} \right) \otimes ... \otimes R_z \left(\frac{c\pi}{2^{N-1}} \right) \qquad (5)$$

Where $t = N - (\tau - 1)$ and $R_z(\theta)$ is a single-qubit **z**-rotation operator, defined as:

$$R_z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \qquad (6)$$

The application of $U_+(c)$ over a register initialized with a state $|\psi\rangle = \sum_{k=0}^{2^N-1} w_k|k\rangle$ is defined as follows:

$$U_+(c)|\psi\rangle = U_+(c)\left(\sum_{k=0}^{2^N-1} w_k|k\rangle\right)$$

$$= \sum_{k=0}^{2^N-1} w_k U_+(c)|k\rangle \qquad (7)$$

$$= \sum_{k=0}^{2^N-1} w_k|k + c \pmod{2^N}\rangle$$

The number of operations required by the register-by-constant addition algorithm for N qubits is $T(N) = T_{\text{QFT}}(N) + N + T_{\text{IQFT}}(N)$, where $T_{\text{QFT}}(N)$ is the number of operations required by QFT and $T_{\text{IQFT}}(N)$ is the number of operations required by IQFT. Assuming $T_{\text{QFT}}(N) = T_{\text{IQFT}}(N) = \frac{N(N+1)}{2}$ [6], we have $T(N) = N(N+1) + N = N^2 + 2N$, and therefore the register-by-constant addition algorithm is $\mathcal{O}(N^2)$.

The Draper's register-by-register addition algorithm with two operand registers of N qubits needs $\frac{N(N+1)}{2}$ controlled operations between the QFT and IQFT, whereas the presented register-by-constant addition algorithm with a target register of N qubits needs only N single-qubit operations between the QFT and IQFT.

4 Correctness proof of the register-by-constant addition algorithm

The correctness of the register-by-constant addition algorithm can be demonstrated by analyzing it in matrix form. First, the state of N qubits $|a\rangle$ for $a \in \{x \in \mathbb{N} : 0 \leq x \leq 2^N - 1\}$ is expanded in matrix form:

$$|a\rangle = [\lambda_j] \in \mathbb{C}^{2^N \times 1}, \quad \lambda_j = \begin{cases} 1, & j = a \\ 0, & j \neq a \end{cases} \qquad (8)$$

In equation 8, the expression $[\lambda_j] \in \mathbb{C}^{2^N \times 1}$ corresponds to a matrix of 2^N rows and 1 column. The variable j corresponds to the row index of the respective λ_j entry of the matrix, such that $0 \leq j \leq 2^N - 1$.

The operator U_{QFT}, which is the quantum discrete Fourier transform, is expanded in matrix form as:

$$U_{\text{QFT}} = \frac{1}{\sqrt{2^N}}[\omega^{jk}] \in \mathbb{C}^{2^N \times 2^N} \qquad (9)$$

In the equation 9, the expression $\frac{1}{\sqrt{2^N}}[\omega^{jk}] \in \mathbb{C}^{2^N \times 2^N}$ corresponds to a matrix of 2^N rows and 2^N columns of which entries are $\frac{1}{\sqrt{2^N}}\omega^{jk}$, where j is the row index of the entry and k is the column index of the entry. The symbol ω is a constant defined as $\omega = e^{\frac{2\pi i}{2^N}}$ [6, 10].

The operation $U_{\text{QFT}}|a\rangle$ results in a column matrix equivalent to the column of index a of the matrix U_{QFT}. Describing this operation algebraically, we have:

$$U_{\text{QFT}}|a\rangle = \frac{1}{\sqrt{2^N}}[\omega^{ja}] \in \mathbb{C}^{2^N \times 1} \qquad (10)$$

In equation 10, the expression $\frac{1}{\sqrt{2^N}}[\omega^{ja}] \in \mathbb{C}^{2^N \times 1}$ corresponds to a column matrix with 2^N rows whose entries are $\frac{1}{\sqrt{2^N}}\omega^{ja}$, where j is the row index of the entry. This expression is equivalent to $\frac{1}{\sqrt{2^N}}[..., \omega^{ja}, ...]^{\mathsf{T}}$.

The application of $U_{\text{QFT}}^\dagger$ over $U_{\text{QFT}}|a\rangle$ results in $|a\rangle$ itself, which leads us to the following observation:

$$U_{\text{QFT}}^\dagger\left(\frac{1}{\sqrt{2^N}}[..., \omega^{ja}, ...]^{\mathsf{T}}\right) = |a\rangle,$$
$$0 \leq a \leq 2^N - 1 \qquad (11)$$

The equation 11 is constrained to $0 \leq a \leq 2^N - 1$. However, for any $a \in \mathbb{N}$, the expression $U_{\text{QFT}}^\dagger\left(\frac{1}{\sqrt{2^N}}[..., \omega^{ja}, ...]^{\mathsf{T}}\right)$ results in $|a \pmod{2^N}\rangle$ and this is due to the periodicity of $a \mapsto \omega^{ja}$. This statement is demonstrated in section 4.1.

Based on equation 11, it can be deduced that:

$$U_{\text{QFT}}^{\dagger}\left(\frac{1}{\sqrt{2^N}}[...,\omega^{j(a+c)},...]^{\mathsf{T}}\right) =$$

$$U_{\text{QFT}}^{\dagger}\left(\frac{1}{\sqrt{2^N}}[...,\omega^{ja}\omega^{jc},...]^{\mathsf{T}}\right) = \quad (12)$$

$$|a+c\rangle,$$

$$0 \le a+c \le 2^N - 1$$

In general, for any $|x\rangle$ such that $0 \le x \le 2^N - 1$, it is considered that $U_{\text{QFT}}|x\rangle = |\phi(x)\rangle$ and $U_{\text{QFT}}^{\dagger}|\phi(x)\rangle = |x\rangle$, where $|\phi(x)\rangle$ is a notation for the equivalent of $|x\rangle$ in Fourier basis.

Based on the statement presented in equation 12, it comes to the conclusion that, in order to create the mapping $|\phi(a)\rangle \mapsto |\phi(a+c)\rangle$, it is necessary to multiply each entry $\frac{1}{\sqrt{2^N}}\omega^{ja}$ of $|\phi(a)\rangle$ by a factor ω^{jc}. This operation can be described as the matrix product $U_{\phi(+)}(c)|\phi(a)\rangle$, since the operator $U_{\phi(+)}(c)$ is expanded in matrix form as shown in the following equation:

$$U_{\phi(+)}(c) = [\lambda_{(j,k)}] \in \mathbb{C}^{2^N \times 2^N},$$

$$\lambda_{(j,k)} = \begin{cases} \omega^{jc}, & j = k \\ 0, & j \ne k \end{cases} \quad (13)$$

The equation 13 defines $U_{\phi(+)}(c)$ as a diagonal matrix whose main diagonal entries are ω^{jc}, where j and k are, respectively, the row index and the column index of the entry. This definition can also be written as $U_{\phi(+)}(c) = \text{diag}[1, \omega^c, \omega^{2c}..., \omega^{(2^N-1)c}]$ or simply $U_{\phi(+)}(c) = \text{diag}[..., \omega^{jc}, ...]$. The equivalence between the definitions of $U_{\phi(+)}(c)$ in equations 5 and 13 is demonstrated by induction in section 4.2.

Based on the facts presented here, it is possible to conclude that:

$$|a+c\rangle =$$

$$U_{\text{QFT}}^{\dagger}|\phi(a+c)\rangle =$$

$$U_{\text{QFT}}^{\dagger} \times U_{\phi(+)}(c)|\phi(a)\rangle =$$

$$U_{\text{QFT}}^{\dagger} \times U_{\phi(+)}(c) \times U_{\text{QFT}}|a\rangle = \quad (14)$$

$$U_+(c)|a\rangle,$$

$$0 \le a+c \le 2^N - 1$$

4.1 Modularity

A function $f(\alpha) = e^{i\alpha} = \cos\alpha + i\sin\alpha$ has a periodicity of 2π, which means that $e^{i\alpha} = e^{i(\alpha \pmod{2\pi})} = e^{i(\alpha+2\pi n)}$ for any $n \in \mathbb{Z}$. Replacing α with $\theta\frac{2\pi}{2^N}$, we have $e^{\theta\frac{2\pi i}{2^N}} = e^{i(\theta\frac{2\pi}{2^N}+2\pi n)} = e^{\theta\frac{2\pi i}{2^N}}e^{2\pi ni}$ for any $n \in \mathbb{Z}$.

We have $e^{\theta\frac{2\pi i}{2^N}} = \omega^{\theta}$ and $e^{2\pi ni} = e^{\left(\frac{2\pi i}{2^N}\right)n2^N} = \omega^{n2^N}$, and by assuming that $e^{\theta\frac{2\pi i}{2^N}} = e^{\theta\frac{2\pi i}{2^N}}e^{2\pi ni}$ for any $n \in \mathbb{Z}$, we have $\omega^{\theta} = \omega^{\theta}\omega^{n2^N} = \omega^{\theta+n2^N}$, therefore $\omega^{\theta} = \omega^{\theta \pmod{2^N}}$. Based on this, we obtain the following statement from the equation 11:

$$U_{\text{QFT}}^{\dagger}|\phi(x)\rangle =$$

$$U_{\text{QFT}}^{\dagger}\left(\frac{1}{\sqrt{2^N}}[...,\omega^{jx},...]^{\mathsf{T}}\right) =$$

$$U_{\text{QFT}}^{\dagger}\left(\frac{1}{\sqrt{2^N}}[...,\omega^{j(x \pmod{2^N})},...]^{\mathsf{T}}\right) = \quad (15)$$

$$|x \pmod{2^N}\rangle$$

Equation 15 implies that for any $0 \le a \le 2^N$ and $c \in \mathbb{N}$, we have:

$$U_+(c)|a\rangle = |a+c \pmod{2^N}\rangle \quad (16)$$

4.2 Demonstration of the equivalence between the definitions 5 and 13

The proposition to be demonstrated is:

$$U_{\phi(+)}(c) = \bigotimes_{\tau=1}^{N} R_z\left(\frac{c\pi}{2^{N-t}}\right) \quad (17)$$

$$= \text{diag}[..., \omega^{jc}, ...]$$

Where $t = N - (\tau - 1)$, $\text{diag}[..., \omega^{jc}, ...]$ is a diagonal matrix whose entries are ω^{jc} and j is the row index of the entry.

The expression $R_z\left(\frac{c\pi}{2^{N-t}}\right)$ can be expanded as:

$$R_z\left(\frac{c\pi}{2^{N-t}}\right) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{c\pi}{2^{N-t}}} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{\left(\frac{2i\pi}{2^N}\right)\frac{c}{2^{1-t}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \omega^{\frac{c}{2^{1-t}}} \end{bmatrix} \quad (18)$$

To show that the proposition 17 is true, a mathematical induction is employed. We begin by expanding $U_{\phi(+)}(c)$ for $N = 1$:

$$\bigotimes_{\tau=1}^{N} R_z \left(\frac{c\pi}{2^{N-t}} \right) \bigg|_{N=1} = \begin{bmatrix} 1 & 0 \\ 0 & \omega^{\frac{c}{2^{1-1}}} \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & \omega^c \end{bmatrix} \tag{19}$$

The equation 19 satisfies the proposition 17.

Expanding $U_{\phi(+)}(c)$ for $N = 2$, we have:

$$\bigotimes_{\tau=1}^{N} R_z \left(\frac{c\pi}{2^{N-t}} \right) \bigg|_{N=2} =$$
$$R_z \left(\frac{c\pi}{2^{N-2}} \right) \otimes R_z \left(\frac{c\pi}{2^{N-1}} \right) =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & \omega^{\frac{c}{2^{1-2}}} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \omega^{\frac{c}{2^{1-1}}} \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & \omega^{2c} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & \omega^{c} \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \omega^c & 0 & 0 \\ 0 & 0 & \omega^{2c} & 0 \\ 0 & 0 & 0 & \omega^c \omega^{2c} \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \omega^c & 0 & 0 \\ 0 & 0 & \omega^{2c} & 0 \\ 0 & 0 & 0 & \omega^{3c} \end{bmatrix} \tag{20}$$

The equation 20 also satisfies the proposition 17.

To proof that the proposition 17 is true for any number of qubits, an inductive step is conducted on the equation 21.

$$\bigotimes_{\tau=1}^{M} R_z \left(\frac{c\pi}{2^{M-t}} \right) \bigg|_{M=N+1} =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & \omega^{\frac{c}{2^{1-(N+1)}}} \end{bmatrix} \otimes \mathrm{diag}[..., \omega^{j_n c}, ...] =$$
$$\begin{bmatrix} 1 & 0 \\ 0 & \omega^{c2^N} \end{bmatrix} \otimes \mathrm{diag}[..., \omega^{j_n c}, ...] = \tag{21}$$
$$\mathrm{diag}[..., \omega^{j_n c}, ..., \omega^{c2^N} \omega^{j_n c}, ...] =$$
$$\mathrm{diag}[..., \omega^{j_n c}, ..., \omega^{(j_n + 2^N)c}, ...] =$$
$$\mathrm{diag}[..., \omega^{j_m c}, ...]$$

In equation 21, it is assumed that the proposition 17 is true for N qubits and it is verified that, under this assumption, it is also true for $M = N + 1$ qubits. The variable j_n corresponds to the row index of a matrix of N rows and columns, and the variable j_m corresponds to the row index of a matrix of M rows and columns.

The equation 17 is therefore true for any number of qubits.

5 Rewriting the register-by-register adder using the $U_{\phi(+)}(c)$ operator

It is possible to rewrite the Draper's register-by-register adder using the $U_{\phi(+)}(c)$ operator defined in sections 3 and 4. This alternative representation is easier to understand than the original one.

As already mentioned in section 2, the register-by-register adder executes the mapping $|a, b\rangle \mapsto |a, b + a \pmod{2^N}\rangle$, where $|a\rangle = |a_N, ..., a_1\rangle$ and $a_j \in \{0, 1\}$.

As in Draper's adder, this alternative register-by-register adder consists of a series of controlled gates applied on the qubits of the second register between QFT and IQFT, i.e, in the Fourier basis. Since it is possible to define $b + a$ as $b + \sum_{j=1}^{N} a_j 2^{j-1}$, the state of the second register in the Fourier basis can be changed from $|\phi(b)\rangle$ to $|\phi(b+a)\rangle$ by applying a $U_{\phi(+)}(2^{j-1})$ controlled by a_j on this register for each $j \in \{1, ..., N\}$.

Figure 3 shows a quantum circuit of the register-by-register adder using controlled $U_{\phi(+)}(c)$ gates.

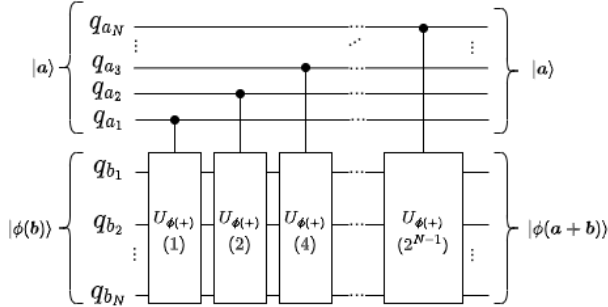Figure 3: Register-by-register adder using the $U_{\phi(+)}(c)$ operator.

6 Conclusion

This paper presented an arithmetic addition algorithm based on the quantum Fourier transform specific for cases where there needs to be added just a constant to a target register, which is called here register-by-constant adder. This adder uses fewer qubits than those proposed in [7, 9], as it requires neither an additional register to store the constant nor ancillary carry qubits. Furthermore, the register-by-constant adder needs fewer operations between the QFT and IQFT than Draper's adder.

This paper also showed an alternative representation of the Draper's adder using the operator $U_{\phi(+)}(c)$, which is easier to understand than the original one.

For futher works, it might be interesting to benchmark between the register-by-constant adder and other addition algorithms on real quantum hardware to compare the noise of the outputs of these algorithms. Moreover, alternative representations using $U_{\phi(+)}(c)$ of the QFT-based multiplier covered in [13] can be made, and more QFT-based algorithms can be designed using this operator.

References

[1] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (May 1980), pp. 563–591. DOI: 10.1007/bf01011339. URL: https://doi.org/10.1007%2Fbf01011339.

[2] Paul Benioff. "Quantum mechanical hamiltonian models of turing machines". In: *Journal of Statistical Physics* 29.3 (Nov. 1982), pp. 515–546. DOI: 10.1007/bf01342185. URL: https://doi.org/10.1007%2Fbf01342185.

[3] Richard P. Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6-7 (June 1982), pp. 467–488. DOI: 10.1007/bf02650179. URL: https://doi.org/10.1007%2Fbf02650179.

[4] David Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 400 (1985), pp. 97–117.

[5] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), pp. 553–558. DOI: 10.1098/rspa.1992.0167. URL: https://doi.org/10.1098/rspa.1992.0167.

[6] Adriano Barenco et al. "Approximate quantum Fourier transform and decoherence". In: *Physical Review A* 54.1 (July 1996), pp. 139–146. DOI: 10.1103/physreva.54.139. URL: https://doi.org/10.1103%2Fphysreva.54.139.

[7] Vlatko Vedral, Adriano Barenco, and Artur Ekert. "Quantum networks for elementary arithmetic operations". In: *Phys. Rev. A* 54 (1 July 1996), pp. 147–153. DOI: 10.1103/

PhysRevA.54.147. URL: https://link.aps.org/doi/10.1103/PhysRevA.54.147.

[8]   Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 1095-7111. DOI: 10.1137/s0097539795293172. URL: http://dx.doi.org/10.1137/S0097539795293172.

[9]   Thomas G. Draper. *Addition on a Quantum Computer*. 2000. arXiv: quant-ph/0008033 [quant-ph].

[10]   Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[11]   D. Coppersmith. *An approximate Fourier transform useful in quantum factoring*. 2002. eprint: arXiv:quant-ph/0201067.

[12]   T. D. Ladd et al. "Quantum computers". In: *Nature* 464.7285 (Mar. 2010), pp. 45–53. DOI: 10.1038/nature08812. URL: https://doi.org/10.1038%2Fnature08812.

[13]   Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. "Quantum arithmetic with the quantum Fourier transform". In: *Quantum Information Processing* 16.6 (Apr. 2017). DOI: 10.1007/s11128-017-1603-1.

[14]   John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), p. 79. DOI: 10.22331/q-2018-08-06-79. URL: https://doi.org/10.22331%2Fq-2018-08-06-79.