Quantum arithmetic with the Quantum Fourier Transform

Article *in* Quantum Information Processing · April 2017

DOI: 10.1007/s11128-017-1603-1 · Source: arXiv

2 authors:

Lidia Ruiz Pérez
Universidad de Valladolid
14 PUBLICATIONS **129** CITATIONS
SEE PROFILE

Juan Carlos Garcia-Escartin
Universidad de Valladolid
51 PUBLICATIONS **824** CITATIONS
SEE PROFILE

QUANTUM ARITHMETIC WITH THE QUANTUM FOURIER TRANSFORM

LIDIA RUIZ-PEREZ[a]

Dpto. de Teoría de la Señal y Comunicaciones e Ing. Telemática. Universidad de Valladolid. ETSI de Telecomunicación. Campus Miguel Delibes. Paseo Belén n 15. 47011, Valladolid. Spain.

JUAN CARLOS GARCIA-ESCARTIN[b]

Dpto. de Teoría de la Señal y Comunicaciones e Ing. Telemática. Universidad de Valladolid. ETSI de Telecomunicación. Campus Miguel Delibes. Paseo Belén n 15. 47011, Valladolid. Spain.

The Quantum Fourier Transform offers an interesting way to perform arithmetic operations on a quantum computer. We review existing Quantum Fourier Transform adders and multipliers and propose some modifications that extend their capabilities. Among the new circuits, we propose a quantum method to compute the weighted average of a series of inputs in the transform domain.

Keywords: Quantum Fourier Transform, quantum adder, quantum multiplier, quantum weighted average

1 Introduction. Quantum arithmetic

The discovery of Shor's algorithm for efficient quantum factoring [1] awakened an interest on the quantum implementation of the modular arithmetic operations that are the building blocks of the quantum factorization circuit. Since then there have been many proposals on how to build the required quantum modular adders, multipliers and exponentiators using a set of elementary quantum gates.

The first suggested circuits were reversible versions of known classical circuits [2, 3]. Many subsequent proposals have been improvements and alterations of different reversible generalization of the adders and multipliers of classical digital logic [4, 5, 6, 7].

There are also solutions with more of a "quantum flavour" such as teleportation-based operations [8], repeat-until-success circuits [9] or implementations that restrict to experimentally achievable quantum operations like the nearest-neighbour interaction [10].

A particularly elegant quantum alternative is the Quantum Fourier Transform, QFT, adder of Draper [11] and its generalizations in a variety of QFT adders and multipliers [12, 13, 14].

In this paper, we study those systems and propose a new QFT-based circuit to compute weighted sums. In Sections 2, 3, 6 and 7, we review the basics of arithmetic operations in

[a]E-mail: lidiaruiz14@gmail.com
[b]E-mail: juagar@tel.uva.es

the transformed domain, describing the basic QFT adders and multipliers and their implementation with elementary gates. We also comment some minor adjustments that allow the circuits to perform non-modular operations and work with signed integers.

Our main result is a modified QFT adder scheme that can compute weighted sums. Section 4 presents the particular case of the arithmetic mean. We then give a description of a weighted adder with constant weights in Section 5 and, in Section 8, a qubit implementation of a programmable weighted adder where the numbers and the weights are the inputs of the circuit. We conclude the paper with a discussion of the applications of this circuit in Section 9.

2 The Quantum Fourier Transform and distributed phase encoding

The Quantum Fourier Transform, QFT, provides an alternative way to perform arithmetic operations on a quantum computer. We consider a d-dimensional system with states $|x\rangle$ from the computational basis $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$. In this basis, we define the *QFT* operation as

$$QFT\,|x\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{i\frac{2\pi xk}{d}} |k\rangle\,. \tag{1}$$

We can use the QFT to encode a number in the relative phases of the states in a superposition that is uniform in amplitude and contains all the states in the computational basis.

Imagine we want to work with natural numbers from 0 to $d-1$. One possible encoding is mapping number x into state $|x\rangle$. With the QFT we can take the information into the phases $e^{i\frac{2\pi xk}{d}} = \omega^{xk}$ that appear together with each state $|k\rangle$ of the superposition. The QFT can be interpreted as a change of basis. We call $|\phi(x)\rangle$ to the state encoding x in this new transformed basis.

We can equally define an Inverse Quantum Fourier Transform, *IQFT*, operator

$$IQFT\,|x\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{-i\frac{2\pi xk}{d}} |k\rangle\,. \tag{2}$$

With the direct and the inverse Fourier transforms we can move back and forth between the computational basis and the phase representation.

3 QFT adders

Previously proposed QFT arithmetic circuits [11, 12, 13, 14] take advantage of this phase encoding. Before describing addition, we need to define a controlled phase gate. We start from the well-known controlled Pauli Z gate, CZ, that for two input qubits $|x\rangle$ and $|y\rangle$ gives

$$CZ\,|x\rangle\,|y\rangle = e^{i\pi xy}\,|x\rangle\,|y\rangle\,. \tag{3}$$

We can generalize the gate for d-dimensional systems (qudits) so that

$$CZ\,|x\rangle\,|y\rangle = e^{i\frac{2\pi xy}{d}}\,|x\rangle\,|y\rangle\,. \tag{4}$$

When $d = 2$ we recover the qubit gate.

We can also define a modified version of the controlled phase shift gate

$$CZ^F \left|x\right\rangle \left|y\right\rangle = e^{i\frac{2\pi xy}{Fd}} \left|x\right\rangle \left|y\right\rangle \tag{5}$$

that introduces a factor F in the divisor which will be useful later.

These ingredients are enough to give a modulo d adder. We can add two numbers originally in the computational basis by taking one of them into phase encoding and then applying a controlled phase shift. The adder comes from the sequence of operations

$$IQFT_2 \cdot CZ \cdot QFT_2 \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|x+y \mod d\right\rangle . \tag{6}$$

Here and in the following equations, when we apply a quantum gate on only a subset of all the possible input states, we introduce subindices to show on which states the gates are acting. The first operation

$$\left|x\right\rangle \left|y\right\rangle \xrightarrow{\mathrm{QFT}_2} \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{i\frac{2\pi yk}{d}} \left|x\right\rangle \left|k\right\rangle \tag{7}$$

encodes number y into the phase basis. The phase gate introduces a phase shift that is equivalent to a modulo d addition in that basis, so that

$$\frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{i\frac{2\pi yk}{d}} \left|x\right\rangle \left|k\right\rangle \xrightarrow{\mathrm{CZ}} \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{i\frac{2\pi yk}{d}} e^{i\frac{2\pi xk}{d}} \left|x\right\rangle \left|k\right\rangle . \tag{8}$$

Finally, the inverse QFT takes the result back into the computational basis with

$$\frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} e^{i\frac{2\pi (x+y)k}{d}} \left|x\right\rangle \left|k\right\rangle \xrightarrow{\mathrm{IQFT}_2} \frac{1}{d} \sum_{k,l=0}^{d-1} e^{i\frac{2\pi (x+y)k}{d}} e^{-i\frac{2\pi kl}{d}} \left|x\right\rangle \left|l\right\rangle = \left|x\right\rangle \left|x+y \mod d\right\rangle . \tag{9}$$

The adder can be extended to any number of inputs. Imagine we have N integers x_1, x_2, …, x_N encoded into the state $\left|x_1\right\rangle \left|x_2\right\rangle \ldots \left|x_N\right\rangle$. Then we can repeat the sum in Equation 6 with the operation

$$IQFT_N \cdot CZ_{1,N} \cdots CZ_{N-2,N} \cdot CZ_{N-1,N} \cdot QFT_N \left|x_1\right\rangle \left|x_2\right\rangle \ldots \left|x_{N-1}\right\rangle \left|x_N\right\rangle \tag{10}$$

that produces an output state

$$\left|x_1\right\rangle \left|x_2\right\rangle \ldots \left|x_{N-1}\right\rangle \left|x_1+x_2+\ldots+x_N \mod d\right\rangle . \tag{11}$$

Each controlled phase shift adds an integer in the phase encoding. This operation uses the minimum possible number of qudits, but it can also be interesting to preserve all the input states and store their sum in an ancillary qudit. In that case, we can apply the procedure to an initial state $\left|x_1\right\rangle \left|x_2\right\rangle \ldots \left|x_N\right\rangle \left|0\right\rangle$. The result is the sum of the N integers plus 0, which gives the same result as in the compact version.

There are a few additional modifications worth noticing. First, if we want to perform arithmetic additions instead of modulo d addition, we can always encode the data in a system of a larger dimension d' where modulo d' addition and regular addition are the same for our range of values. For instance, for two integers x and y between 0 and $d-1$, the sum will

always stay between 0 and $2d-2$ and a system of dimension $d'=2d-1$ will suffice. We can take an input state $|x\rangle_d |y\rangle_d |0\rangle_{2d-1}$ with systems of dimensions d, d and $2d-1$ respectively and use the QFT for $d'=2d-1$ and CZ operations that can also be defined for inputs of a different size as

$$CZ\,|x\rangle_d\,|y\rangle_{2d-1} = e^{i\frac{2\pi xy}{2d-1}}\,|x\rangle_d\,|y\rangle_{2d-1}\,. \tag{12}$$

In Section 6 we describe these gates for the most common implementation that uses two dimensional systems (qubits).

Similarly, if we have N numbers, we need a system with dimension $d'=Nd-N+1$.

Finally, we can include signed addition for numbers up to $d/2$ if we encode a number x into phases $e^{i\frac{2\pi x}{d}}$ below π, which correspond to a phase $e^{i\frac{2\pi xk}{d}}$ accompanying each state $|k\rangle$, and $-x$ is encoded into states $|d-x\rangle$ in the computational basis.

4 Computing the mean with the QFT

A simple extension to the quantum adder can compute the arithmetic mean of a set of integers. We consider again N integers x_1, x_2, ..., x_N encoded into a state $|x_1\rangle\,|x_2\rangle\ldots|x_N\rangle$ and an ancillary $|0\rangle$ qudit. If we replace the CZ gates in Equation (10) by CZ^N gates, we have the evolution

$$\begin{aligned} IQFT_{N+1}\left(\prod_{m=1}^{N} CZ^N_{m,N+1}\right) QFT_{N+1}\,|x_1\rangle\,|x_2\rangle\ldots|x_N\rangle\,|0\rangle \\ = |x_1\rangle\,|x_2\rangle\ldots|x_N\rangle\left|\frac{1}{N}\sum_{m=1}^{N} x_m \mod d\right\rangle, \end{aligned} \tag{13}$$

which produces the desired average.

Notice that, in this case, the arithmetic mean is always equivalent to the modular addition. The mean of numbers from 0 to $d-1$ is always between 0 and $d-1$ and we can keep all the states in a space of dimension d.

5 Weighted sums and multiplication by a constant

We can also readily compute any weighted sum

$$\sum_{m=1}^{N} a_m x_m \tag{14}$$

if we start with a state

$$|x_1\rangle\,|x_2\rangle\ldots|x_N\rangle\,|0\rangle \tag{15}$$

and apply the gate sequence

$$IQFT_{N+1}\left(\prod_{m=1}^{N} CZ^{\frac{1}{a_m}}_{m,N+1}\right) QFT_{N+1}. \tag{16}$$

The resulting state

$$|x_1\rangle\,|x_2\rangle\ldots|x_N\rangle\,|a_1x_1+a_2x_2+\ldots+a_Nx_N \mod d\rangle \tag{17}$$

returns the modulo d weighted sum. As in regular addition, we might need to choose a different dimension for our ancillary qudit and operations if we want to obtain the pure weighted sum instead, or if we want to include signs.

A particular case happens when all the a_m are positive and $\sum_m a_m = 1$, like in the example of the arithmetic mean where $a_m = \frac{1}{N}$ for all m. Then the result is guaranteed to be between 0 and $d-1$ and the modulo d sum and the total weighted sum are always the same.

Multiplication by a constant can be seen as a particular case of weighted sum. We can multiply two numbers x and b, with b constant and x any integer from 0 to $d-1$, using the binary decomposition of b. If b has n bits, we can write the product bx as the sum

$$(b_1 2^{n-1} \cdot b_2 2^{n-2} \cdots b_{n-1} 2^1 \cdot b_n 2^0)x = \sum_{m=1}^{n} b_m 2^{n-m} x, \tag{18}$$

which is a weighted sum with coefficients $a_m = b_m 2^{n-m}$ and where all the x_m are equal. Section 7 describes a variation of this method that gives a QFT multiplier.

6 QFT Adder. Qubit implementation

We first describe a variant on Draper's QFT Adder [11] to compute arithmetic additions and its implementation with elementary gates. We consider a system composed of a collection of two-level systems (qubits).

Let a, b that are integers from 0 to $2^n - 1$ be the numbers to add. Let $a_1 a_2 \ldots a_n$ and $b_1 b_2 \ldots b_n$ be the binary representations of a and b, where $a = a_1 2^{n-1} + a_2 2^{n-2} + \ldots + a_n 2^0$ and $b = b_1 2^{n-1} + b_2 2^{n-2} + \ldots + b_n 2^0$. Then $|a\rangle = |a_1\rangle \otimes |a_2\rangle \otimes \ldots \otimes |a_n\rangle$ and $|b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \ldots \otimes |b_n\rangle$.

Draper's circuit first computes the quantum Fourier transform of a, evolving $|a\rangle$ into $|\phi(a)\rangle$:

$$|\phi(a)\rangle = QFT\,|a\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i\frac{2\pi a k}{N}} |k\rangle\,, \tag{19}$$

where $N = 2^n$. Then the circuit computes the addition, using the n qubits that represent the number b to take $|\phi(a)\rangle$ into $|\phi(a+b)\rangle$. To perform the addition, the circuit decomposes the CZ gates presented in Section 3 into conditional rotation phase gates of the form:

$$R_l = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^l}} \end{bmatrix}. \tag{20}$$

These gates are controlled by the n qubits that represent the number b. The combined effect of all the gates is to introduce a total phase $e^{\frac{2\pi i b k}{N}}$ for each state $|k\rangle$, so that the register containing the number b keeps the same value while the register containing the QFT of the number a now stores $|\phi\,(a+b)\rangle$.

We can extend the scheme to perform arithmetic additions by encoding a into a larger register. We represent the number a using $n+1$ qubits so $|a\rangle = |0\rangle\,|a_1\rangle\,|a_2\rangle \ldots |a_n\rangle$ and compute the QFT of $|a\rangle$:

$$QFT\,|a\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{k=0}^{2^{n+1}-1} e^{i\frac{2\pi a k}{2^{n+1}}} |k\rangle\,, \tag{21}$$

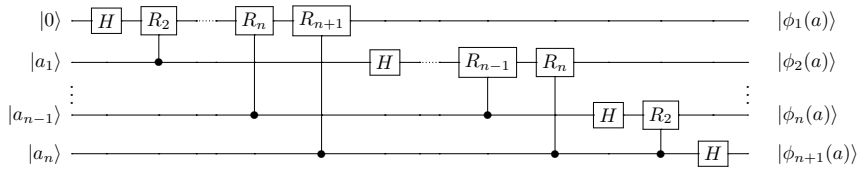

Fig. 1. QFT of the state $|0\rangle|a\rangle$. The circuit takes an n-bit number a that is encoded in the computational basis into a modulo 2^{n+1} phase encoding.

using the QFT circuit shown in Figure 1, where the states $|\phi_j(a)\rangle$ represent the jth qubit of the phase state $|\phi(a)\rangle$ encoding a. For simplicity, in the figure we have omitted the sequence of SWAP gates needed to invert the order of the output qubits.

Once we have $|\phi(a)\rangle$, we add the number b using controlled phase rotation gates as in Draper's scheme. We add a and b by applying the conditional phase rotation

$$e^{2\pi i \frac{(a_j+b_j)2^{n-j}k_s 2^{n-s}}{2^{n+1}}} = e^{2\pi i \frac{(a_j+b_j)k_s}{2^{j+s-n}}} \tag{22}$$

that depends on the jth qubits of the representation of the numbers to be added and is applied on the sth qubit in the transformed register containing superpositions of states $|k\rangle = |k_1\rangle \otimes \cdots \otimes |k_{n+1}\rangle$. The gate is controlled by the jth qubit of $|b\rangle$ and only produces a change if $b_j = 1$. We have to choose the conditional phase rotation gates $R_l = R_{j+s-n}$ when $j + s - n > 0$. Note that if $j + s - n \leq 0$, we are applying the phase $e^{2\pi 2^{n-j-s}i} = 1$, so the state remains unaltered. The resulting circuit is shown in figure 2.

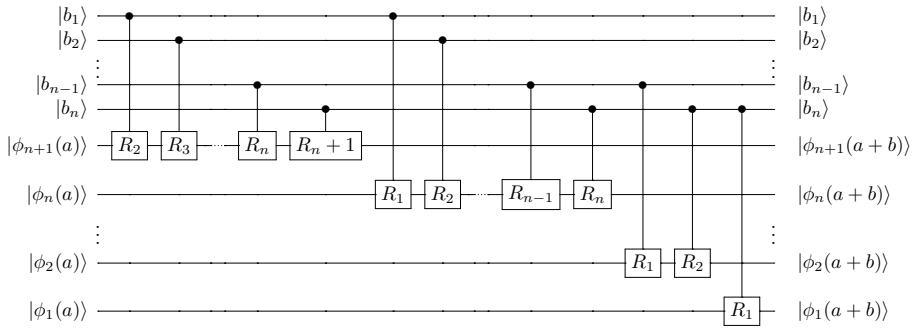

Fig. 2. Arithmetic sum in the transform domain. The circuit uses one of the integers, b, to introduce controlled phase shifts into the phase-encoded second integer, a.

As a result, the register containing the QFT of a now stores $|\phi(a+b)\rangle$. By adding an ancillary qubit to encode a we still perform a modular addition, but we avoid overflow and leave space to recover the arithmetic addition of a and b.

7 QFT Multiplier

We can design a quantum circuit to multiply two n-bit numbers by performing n consecutive controlled QFT additions. The result will be a $2n$-qubit register encoding the number $a \cdot b$. The circuit is shown in Figure 3. The first adder block, labelled as $2^0\Sigma$, takes as input the

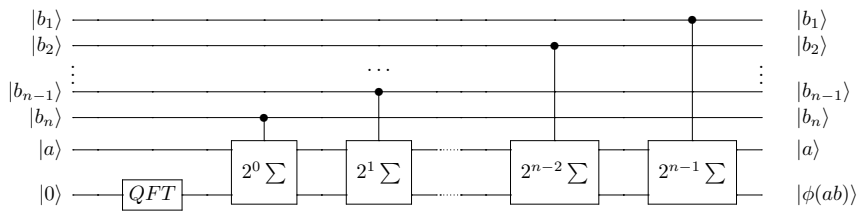

Fig. 3. QFT multiplier as a sequence of weighted QFT adders.

n qubits representing a number a, and $2n$ qubits representing the number 0. The block first computes the quantum Fourier transform of number 0, *i.e.*, $|\phi(0)\rangle$, and then applies conditional phase rotation gates to evolve the state into $|\phi(0+a)\rangle$. The block is controlled by the least significant qubit of $|b\rangle$ so it produces the output state

$$|a\rangle \left|\phi(0+b_n 2^0 a)\right\rangle. \tag{23}$$

We apply a second QFT adder controlled by b_{n-1}. Now the phase addition is scaled by a factor 2^1 so that the output state will be

$$\left|\phi(0+b_n 2^0 a + b_{n-1} 2^1 a)\right\rangle. \tag{24}$$

We proceed in a similar fashion with the remaining blocks. When the last QFT adder is applied, we get the output state

$$\left|\phi(0+b_n 2^0 a + b_{n-1} 2^1 a + \ldots + b_2 2^{n-2} a + b_1 2^{n-1} a)\right\rangle = |\phi(0+ab)\rangle = |\phi(ab)\rangle. \tag{25}$$

The key to compute the product $a \cdot b$ is to select the proper conditional phase rotation gates to implement each QFT adder block. After computing the QFT of 0, we obtain the output state

$$QFT\,|0\rangle = \frac{1}{\sqrt{2^{2n}}} \sum_{k=0}^{2^{2n}-1} e^{i\frac{2\pi 0 k}{2^{2n}}} |k\rangle = |\phi(0)\rangle, \tag{26}$$

where $k = k_1 2^{2n-1} + k_2 2^{2n-2} + \ldots + k_{2n} 2^0 = \sum_{s=1}^{2n} k_s 2^{2n-s}$. In order to take $|\phi(0)\rangle$ to $\left|\phi(0+b_j 2^{n-j} a)\right\rangle$, we need to use a number of phase rotation gates controlled by b_j and by each a_i, chosen so that they apply a phase rotation

$$e^{i\frac{2\pi(a_i 2^{n-1} b_j 2^{n-j}) k_s 2^{2n-s}}{2^{2n}}} = e^{i\frac{2\pi a_i b_j k_s}{2^{i+j+s-2n}}}. \tag{27}$$

Therefore, we select conditional rotation gates of the form $R_l = R_{i+j+s-2n}$, where $i+j+s-2n > 0$, to implement the QFT adder block controlled by b_j.

In this circuit, we have chosen the size of the ancillary register so that we get the exact value of $a \cdot b$ instead of a modular multiplication. We can vary the size of the ancillary register and modify the R_l gates accordingly to obtain any desired modular multiplication in moduli that are powers of two of the size of the ancillary register.

8 Controlled Weighted Sum

We can now implement a quantum circuit to compute the weighted sum

$$\sum_{m=1}^{N} a_m x_m, \tag{28}$$

in which each a_m is an input to the system that is stored in a quantum register and can be in a superposition of different values. To build such a circuit we can use an architecture similar to the QFT multiplication block introduced in Section 7. Each qubit of a_m controls how to add the contribution of each qubit of x_m. If we directly use the circuit of Figure 3, we compute the weighted sum for integer weights. However, the discrete weights a_m can be adjusted to any range of interest simply by introducing the appropriate factor in the corresponding CZ^F gates. We define a precision variable p so that the weights a_m are the integers represented by each binary string encoded in the weight qubits divided by 2^p. If we use q qubits to store each weight a_m, we can obtain weights a_m in a range $0 < a_m < 2^{q-p}$ with a precision 2^{-p}. We can define any desired range of values with the precision we require if we adjust the values of p and q.

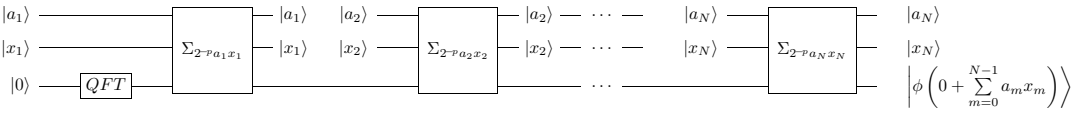

Fig. 4. Controlled weighted sum (block diagram). Each block adds a value $a_m x_m$ to an ancillary register.

The circuit is implemented using N variations of the QFT multiplication blocks as shown in Figure 4. We first compute the quantum Fourier Transform of $|0\rangle$. Then we apply the first multiplication block, that takes the input state

$$|\mathbf{a_1}\rangle |\mathbf{x_1}\rangle \otimes \cdots \otimes |a_N\rangle |x_N\rangle |\phi(0)\rangle \tag{29}$$

and returns the output state

$$|\mathbf{a_1}\rangle |\mathbf{x_1}\rangle \otimes \cdots \otimes |a_N\rangle |x_N\rangle |\phi(0 + \mathbf{a_1 x_1})\rangle \,. \tag{30}$$

The second multiplier acts in a similar manner, taking the input state

$$|a_1\rangle |x_1\rangle |\mathbf{a_2}\rangle |\mathbf{x_2}\rangle \otimes \cdots \otimes |a_N\rangle |x_N\rangle |\phi(0 + a_1 x_1)\rangle \tag{31}$$

and returning the output state

$$|a_1\rangle |x_1\rangle |\mathbf{a_2}\rangle |\mathbf{x_2}\rangle \otimes \cdots \otimes |a_N\rangle |x_N\rangle |\phi(0 + a_1 x_1 + \mathbf{a_2 x_2})\rangle \,. \tag{32}$$

After applying all the multipliers, we get the final output state

$$|a_1\rangle |x_1\rangle |a_2\rangle |x_2\rangle \otimes \cdots \otimes |a_N\rangle |x_N\rangle |\phi(0 + a_1 x_1 + a_2 x_2 + \ldots + a_N x_N)\rangle \,. \tag{33}$$

Figure 5 shows an example of the gates inside each of the multiplication blocks. The Figure describes the circuit controlled by the jth qubit of the m-th weight and how it controls the

R_l operations on the qubits of the mth value. The subindices are written for a system with weights that have q bits, input numbers x_m with n bits and that compute the weighted sum modulo 2^t. If we want the bare weighted sum and there are N values to be added, the ancillary register must have $t = \lceil \log_2(N2^{q+n}) \rceil = \lceil (q+n)\log_2(N) \rceil$ qubits.

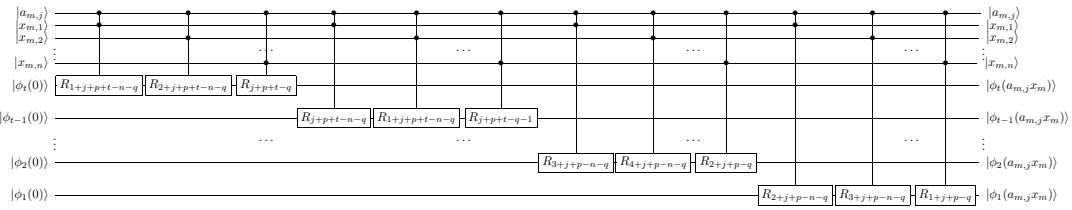

Fig. 5. Example block of a controlled weighted sum. The jth qubit of the weight a_m and the qubits of the value x_m control the phase shift induced in an ancillary register.

The block on Figure 5 must be repeated for all the qubits of $|a_m\rangle$. The gate acting on the uth qubit of the ancillary register is controlled by the ith qubit of $|x_m\rangle$ and the jth qubit of $|a_m\rangle$ and must produce a phase shift

$$e^{i\frac{2\pi x_i 2^{n-i} a_j 2^{q-p-j} k_u 2^{t-u}}{2^t}} = e^{i\frac{2\pi x_i a_j k_u}{2^{i+j+u+p-n-q}}}, \tag{34}$$

which corresponds to the conditional phase rotation gate $R_l = R_{i+j+u+p-n-q}$.

9 Discussion

The Quantum Fourier Transform offers a versatile way to perform modular and non-modular arithmetic on a quantum computer. We have discussed how QFT adders and multipliers provide compact circuits for quantum arithmetic and have given a few modifications to accommodate signed numbers and different moduli.

We have also shown that certain operations, like the arithmetic mean or any weighted average, can be implemented with the same number of gates as a basic QFT addition. If the elementary gates can be classically programmed, this allows for a flexible quantum weighted sum calculator that avoids computing each weight-value product.

Additionally, we have presented a quantum circuit that computes the weighted sum for both quantum weights and values. The circuit for N numbers takes as many gates as N multipliers but needs no additions. In all the cases, there is an overhead in the form of the direct and inverse QFT.

The controlled quantum weighted adder opens many applications. Optimizing weighted sums is a problem that appears in data processing and network planning among others. A quantum weighted adder that has a uniform superposition of all the possible discrete weights for a given register size as its input can be combined with the quantum algorithm for finding the minimum [15] to obtain a quadratic speedup in the weight optimization problem. For a good enough weight precision this can be very helpful.

References

1. P. W. Shor (1997) *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, 26(5), 1484.

2. V. Vedral, A. Barenco, and A. Ekert (1996) *Quantum networks for elementary arithmetic operations*. Physical Review A, 54(1), 147–153.
3. D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill (1996) *Efficient networks for quantum factoring*. Physical Review A, 54, 1034–1063.
4. S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton (2004) *A new quantum ripple-carry addition circuit*. arXiv preprint quant-ph/0410184.
5. R. Van Meter and K. M. Itoh (2005) *Fast quantum modular exponentiation*. Physical Review A, 71, 052320.
6. T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore (2006) *A Logarithmic-depth Quantum Carry-lookahead Adder*. Quantum Information & Computation, 6(4), 351–369.
7. J. J. Álvarez-Sánchez, J. V. Álvarez-Bravo, and L. M. Nieto (2008) *A quantum architecture for multiplying signed integers*. Journal of Physics: Conference Series, 128(1), 012013.
8. R. V. Meter, W. J. Munro, K. Nemoto, and K. M. Itoh (2008) *Arithmetic on a Distributed-memory Quantum Multicomputer*. Journal of Emerging Technologies in Computing Systems, 3(4), 2:1–2:23.
9. N. Wiebe and M. Roetteler (2014) *Quantum arithmetic and numerical analysis using Repeat-Until-Success circuits*. Techical Report MSR-TR-2014-103, Microsoft Research.
10. B.-S. Choi and R. Van Meter (2012) *A $\Theta(\sqrt{n})$-depth Quantum Adder on the 2D NTC Quantum Computer Architecture*. Journal of Emerging Technologies in Computing Systems, 8(3), 24:1–24:22.
11. T. G. Draper (2000) *Addition on a quantum computer*. arXiv preprint quant-ph/0008033.
12. S. Beauregard (2003) *Circuit for Shor's Algorithm Using 2n+3 Qubits*. Quantum Information & Computation, 3(2), 175–185.
13. A. Pavlidis and D. Gizopoulos (2014) *Fast Quantum Modular Exponentiation Architecture for Shor's Factoring Algorithm*. Quantum Information & Computation, 14(7& 8), 649–682.
14. C. Maynard and E. Pius (2014) Quantum Information Processing, 13(5).
15. C. Dürr and P. Hoyer (1996) *A Quantum Algorithm for Finding the Minimum*. eprint arXiv:quant-ph/9607014.