Summary of Chapter 7

The main conclusion of the preceding chapters is that it is possible (at least conceptually) to design quantum circuits that implement, with some margin of error, any predefined unitary operator. In Chap.4, the main basic components are described, and, in Chap.6, synthesis methods are proposed. Large scale quantum computers are not yet available but might exist in the future. The question is "What kind of algorithm can be executed by a quantum circuit, with a better efficiency than with a digital circuit? "

Switching functions, for example arithmetic functions, are certainly not the type of algorithm that quantum circuits can execute in an efficient way, unless they are an essential part of another algorithm (Sec.5 and 6). Observe also that the emulation of a sequential machine with a quantum circuit is a difficult task. The sequential machine should be transformed into an equivalent combinational iterative circuit.

The superiority of quantum circuits with respect to digital circuits resides in the fact that the qubits can be set into superposition states. This means that an operation can be executed with different operands in an apparently simultaneous way. The weak point of quantum circuits is that the measurement of their final state generates one of the most likely results, not the whole final state description. This gives a clue about the type of algorithm that must be considered. Consider a function $f(x)$ where x belongs to a finite domain D. The computation of the value of f, at some particular point x of D, can be done with a digital circuit, but the computation of a characteristic of f that depends on the value of f at many points x of D could be more efficiently performed with a quantum circuit, taking advantage of the possibility of setting the qubits in a superposition

state. Obviously, the answer of the quantum circuit must be reasonably clear: one of the most likely final states should be the expected solution.

As an example, consider a function f, easy to compute but hard to invert, a so-called one-way function. This type of function is used in public-key cryptography. Given a message x, then $y = f(x)$ is the encrypted message. To decrypt the message y, without knowing the private key, it is necessary to find a value x such that $f(x) = y$. This could be done with a digital circuit including a circuit that computes $f(x)$ and a counter that generates all the values of x (Fig.1.a). A match between the computed value $f(x)$ and the encrypted message y will eventually be obtained, but, if x is a very long binary vector, it could be impossible to generate all the candidate values of x within a reasonable time. A quantum system composed of two registers (Fig.1.b) could be considered. Assume that the messages are n-bit vectors. The first register, composed of n qubits, is set into a superposition state $|\psi\rangle$ that includes all 2^n possible basic states $|x\rangle$ with the same probability. The second register is made up of ancillary qubits that are initially set to the ground state $|0\rangle$. Then, the problem can be stated as follows: find a unitary operation $U(f, y)$ such that, after its execution, the measurement of the final state of the first register gives a value z, with a relatively high probability that $f(z) = y$. If this probability is equal to p, resetting the system and executing the operation $U(f, y)$ about $1/p$ times, the value of x will be found. The Grove algorithm of Sec.6 could be seen as a first approximation to this problem.

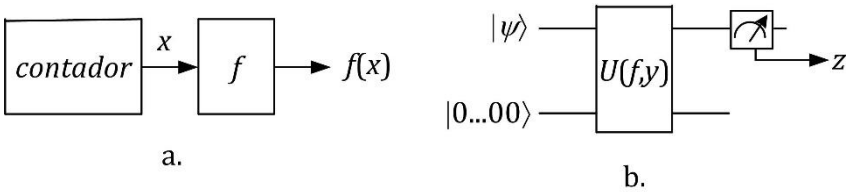

Figure 1 Inversion of a 1-way function f

Independently of the executed algorithm, an obvious difference between digital and quantum circuits is the amount of information that they store and process. Consider an *N*-qubit circuit and try to emulate its operations with a digital circuit. The quantum state must be represented by a register that store all the coefficients a_k of an expression such as

$$a_0|0\ldots00\rangle+a_1|0\ldots01\rangle+\ldots+a_{2^N-1}|1\ldots11\rangle.$$

Each coefficient is a complex number whose norm is smaller than or equal to 1. They could be represented by a pair of *n*-bit fixed-point numbers, so that this state register should store $n\cdot 2^{N+1}$ bits. Thus, to emulate a quantum circuit composed of e.g. 100 qubits, with 32-bit fixed-point numbers, the size of the state register is 2^{106} bits, an astronomical number.

1. Quantum parallelism

In order to accelerate the execution of an algorithm, one of the basic methods – apart from the modification of the algorithm – is the execution of operations in parallel.

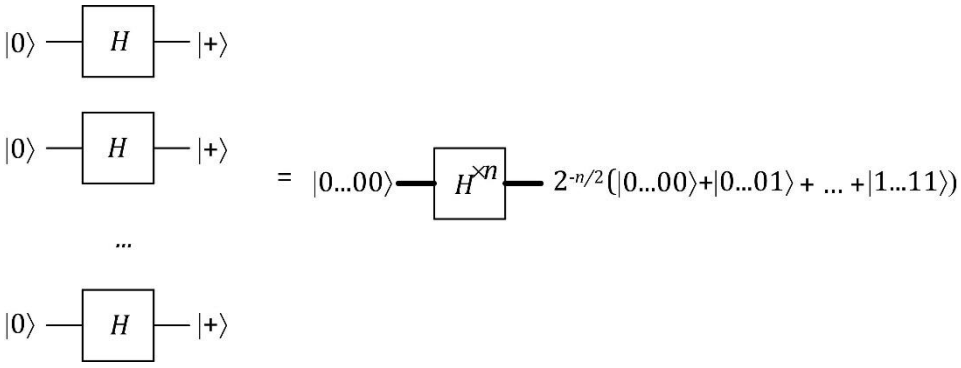

Figure 2 Superposition of 2^n basic states

A circuit made up of *n* Hadamard operators is shown in Fig.2. If the register is initially in state |00...0⟩, after execution of the Hadamard operations the state is

$$|\psi\rangle = |+\rangle \times |+\rangle \times \ldots \times |+\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=00\ldots0}^{x=11\ldots1} |x_0 x_1 \ldots x_{n-1}\rangle, \qquad (1)$$

(|+⟩ are |-⟩ defined in Chap.4, relation 44), that is, a superposition of all 2^n basic states, with the same probability $1/2^n$.

Consider a switching function $f(x)$, where x is an n-bit vector. It can be computed with a unitary operator U (Chap.6, Fig.1). By combining this unitary operator with the circuit of Fig.2, the circuit of Fig.3 is obtained. Its final state is

$$\frac{1}{\sqrt{2^n}} \sum_{x=00\ldots0}^{x=11\ldots1} |x_0 x_1 \ldots x_{n-1}\, f(x_0, x_1, \ldots, x_{n-1})\rangle. \qquad (2)$$

Thus, this circuit implicitly computes the value of $f(x)$ for all 2^n n-bit vectors. Unfortunately, the measure of the quantum state generates the value of only one of the pairs $(x, f(x))$. This circuit is not efficient for computing the value of f at some particular point a. However, it could be useful as part of a system that computes a characteristic of f depending on the values of $f(x)$ over the whole domain of x. An example is the Deutsch algorithm of Sec.2.

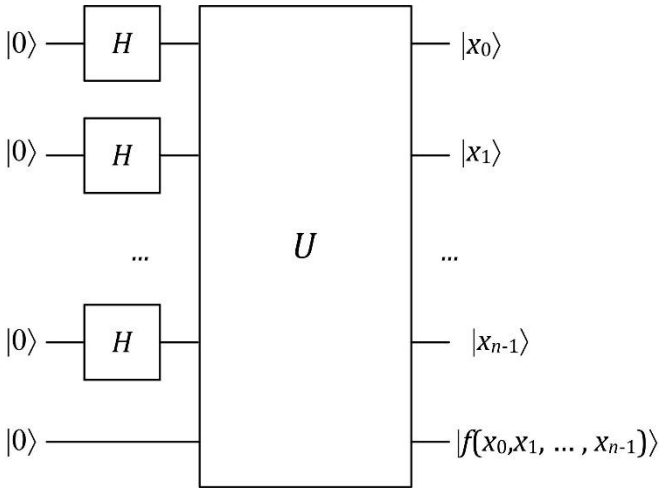

Figure 3 Computation of $f(x)$

2. Deutsch algorithm

This first example of quantum algorithm has no practical interest, but it serves to sense which is the type of algorithm that a quantum circuit can

execute more efficiently than a digital circuit. Consider a digital circuit (Fig.4.a) that computes a switching function $f(x)$ of a binary variable x, and a 2-qubit quantum circuit (Fig.4.b) that executes a unitary operation U that executes the transformation $U|x\ y\rangle = |x\ y \oplus f(x)\rangle$, for any basic state $|x\ y\rangle$. The key point (at the same time the unrealistic one) is that f is an unknown function. The circuits of Fig.4 are black boxes that, given an input x, compute $f(x)$. In computer sciences they are called "oracles". The problem to be resolved is the following: determine whether f is a constant function, or not. In other words, compute

$$z = f(0) \oplus f(1). \tag{3}$$

This is an example of a problem whose solution depends on the value of $f(x)$ at all points of the domain of x, in this case only two points.

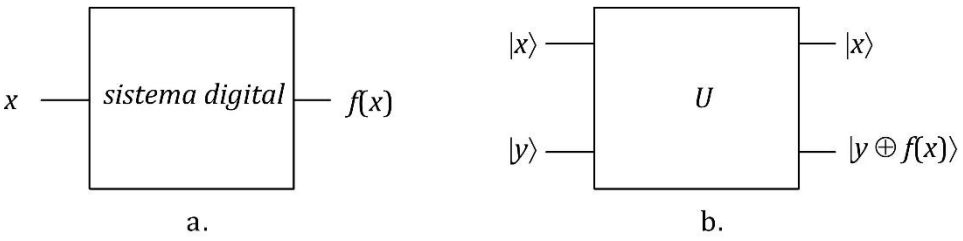

Figure 4 Digital and quantum circuits

The computation of (3), with the black box of Fig.4.a, must be executed in two steps, for example,

```
z = f(0)
z = z ⊕ f(1)
```

The same result can be obtained, in a single step, with the quantum circuit of Fig.5, that includes the unitary operator of Fig.4.b. The two qubits are initially set in state $|01\rangle$. Then, with Hadamard gates, they are set in superposition states:

$$|0\rangle \times |1\rangle \xrightarrow{H \times H} |+\rangle \times |-\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle),$$

and, with the unitary operator U, the following state is obtained:

$$|\psi\rangle = \frac{1}{2}(|0\, f(0)\rangle - |0\, \overline{f(0)}\rangle + |1\, f(1)\rangle - |1\overline{f(1)}\rangle). \quad (4)$$

The state $|\psi\rangle$ depends on the values of $f(0)$ and $f(1)$:

- if $f(0) = f(1) = 0$,

$$|\psi\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \frac{1}{2}(|00\rangle + e^{i\pi}|01\rangle + |10\rangle + e^{i\pi}|11\rangle); \quad (5)$$

- if $f(0) = f(1) = 1$

$$|\psi\rangle = \frac{1}{2}(|01\rangle - |00\rangle + |11\rangle - |10\rangle) = -\frac{1}{2}(|00\rangle + e^{i\pi}|01\rangle + |10\rangle + e^{i\pi}|11\rangle); \quad (6)$$

- if $f(0) = 0$ y $f(1) = 1$

$$|\psi\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = \frac{1}{2}(|00\rangle + e^{i\pi}|01\rangle + e^{i\pi}|10\rangle + |11\rangle); \quad (7)$$

- if $f(0) = 1$ y $f(1) = 0$

$$|\psi\rangle = \frac{1}{2}(|01\rangle - |00\rangle + |10\rangle - |11\rangle) = -\frac{1}{2}(|00\rangle + e^{i\pi}|01\rangle + e^{i\pi}|10\rangle + |11\rangle). \quad (8)$$

In conclusion, if $f(0) = f(1)$, then, according to (5) and (6), the relative phases of the coefficients of $|\psi\rangle$ are 0, π, 0, π, respectively, and if $f(0) \neq f(1)$, then, according to (7) and (8), the relative phases are 0, π, π, 0. Thus, the value of z is encoded by the relative phases. To decode this information, an additional H operator is used, so that the following transformations are executed (I is the identity operation):

$$\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \xrightarrow{H\times I}$$

$$\frac{1}{2\sqrt{2}}[(|00\rangle+|10\rangle)-(|01\rangle+|11\rangle)+(|00\rangle-|10\rangle)-(|01\rangle-|11\rangle)] =$$

$$\frac{1}{\sqrt{2}}(|00\rangle-|01\rangle); \quad (9)$$

$$\frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) \xrightarrow{H\times I}$$

$$\frac{1}{2\sqrt{2}}[(|00\rangle+|10\rangle)-(|01\rangle+|11\rangle)-(|00\rangle-|10\rangle)+(|01\rangle-|11\rangle)] = \frac{1}{\sqrt{2}}(|10\rangle-|11\rangle). \quad (10)$$

If $f(0) = f(1)$, then, according to (9), the result of the first qubit measurement is certainly 0, and if $f(0) \neq f(1)$, according to (10), the result is certainly 1.

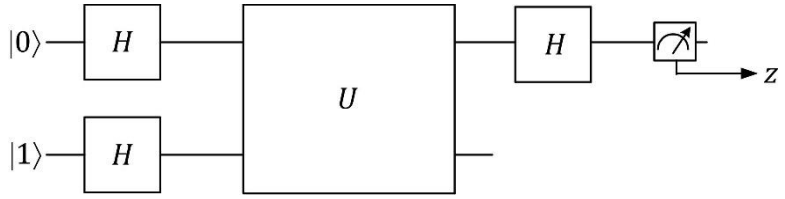

Figure 5 Deutsch algorithm

In conclusion, the computation of a characteristic of *f*, that depends on the value of $f(x)$ at all points of the domain of *x*, has been executed in a single step with a quantum circuit, while a digital circuit needs two steps to execute the same computation. Observe that the quantum circuit doesn't give any information about e.g. the value of *f* at point 0; it only says whether the value of *f* at point 0 is the same as at point 1, or not.

Comments 1

This first example is a simplified version of the Deutsch-Jozsa algorithm ([2]). It highlights several aspects of the quantum algorithms:

- Parallelism, thanks to the possibility of setting qubits in a superposition state.
- Definition of unitary operators that act on the relative phases rather than on the probabilities of measuring particular states. In this example, it could seem that *U* doesn't modify the state of the first qubit. In fact, if the state of the first qubit were measured after the execution of *U*, the result would be 0 or 1, with the same probabilities 0.5 as before the execution of *U*. However, observe that the execution of *U* makes the state $|y \oplus f(x)\rangle$ of the second qubit dependent on the state $|x\rangle$ of the first one. Thus, the first qubit acts as a control qubit while the other is the target qubit. The modification of the control qubit phase,

when a controlled operation is executed, is a quantum effect known as "phase kickback". The phase (0 o π) is not directly measurable, so that an additional *H* gate was necessary.

- Importance of the Hadamard operator. It encodes the basic states $|0\rangle$ and $|1\rangle$ into superposition states $|+\rangle$ and $|-\rangle$, and inversely.

3. Quantum Fourier transform

This section describes a generic algorithm that, in turn, is a computation primitive used in other algorithms.

The Fourier analysis is a tool widely used in mathematics, physics, electronics, telecommunications, information processing, and other engineering fields. A quantum version has been defined.

3.1. Definitions

Consider a quantum circuit composed of *n* qubits. Its 2^n basic states are $|0...00\rangle$, $|0...01\rangle$, ... , $|1...11\rangle$. The quantum Fourier transform (*QFT*), applied to a basic state $|j_1\, j_2 ...\, j_n\rangle$ of this *n*-qubit register executes the following operation:

$$QFT|j_1\, j_2 ...\, j_n\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk}\, |k_1 k_2 \ldots k_n\rangle, \qquad (11)$$

with

$$N = 2^n,\ j = j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0\ ,\ k = k_1 2^{n-1} + k_2 2^{n-2} + \ldots + k_n 2^0,\ \omega_N = e^{\frac{2\pi i}{N}}. \qquad (12)$$

This operation (11) transforms a basic state $|j_1\, j_2 ...\, j_n\rangle$ into a superposition of *N* basic states $|k_1\, k_2 ...\, k_n\rangle$, each of them with a coefficient equal to $\frac{1}{\sqrt{N}}\, \omega_N^{jk}$, so that they all have the same probability $\frac{1}{N}$. Thus, the binary vector j_1, j_2, ..., j_n has been encoded under the form of *N* relative phases $j \cdot k/N$.

The inverse quantum Fourier transform (*IQFT* or QFT^{-1}), applied to a basic state $|k_1\, k_2 ...\, k_n\rangle$ executes the following operation:

$$QFT^{-1}|k_1\, k_2 \ldots k_n\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{-kj}\, |j_1 j_2 \ldots j_n\rangle. \quad (13)$$

The transformations (12) and (13) have been defined in the case of basic states. By linearity, they can be applied to any quantum state. For example, if

$$|\psi\rangle = \sum_{j=0}^{N-1} a_j\, |j_1 j_2 \ldots j_n\rangle, \quad (14)$$

then

$$QFT|\psi\rangle = \sum_{k=0}^{N-1} b_k |k_1 k_2 \ldots k_n\rangle \quad (15)$$

with

$$b_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} a_j\, \omega_N^{jk}. \quad (16)$$

3.2. Circuits

The relation (11) can be slightly modified: ω_N is a *Nth* root of 1 so that

$$\omega_N^{jk} = \omega_N^{(jk) mod\, N}. \quad (17)$$

Given that $N = 2^n$, the mod N reduction is trivial:

$$jk \bmod 2^n = k_1(j_n 2^{n-1}) + k_2(j_{n-1} 2^{n-1} + j_n 2^{n-2})$$

$$+ \ldots + k_{n-1}(j_2 2^{n-1} + \ldots + j_n 2^1) + k_n(j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0). \quad (18)$$

Thus,

$$\omega_N^{jk} = e^{\frac{2\pi i jk}{N}} = e^{2\pi i k_1 (0.j_n)} \cdot e^{2\pi i k_2 (0.j_{n-1} j_n)} \cdot \ldots \cdot e^{2\pi i k_n (0.j_1 j_2 \ldots j_n)}, \quad (19)$$

where $0.j_n$, $0.j_{n-1}j_n$, ... , $0.j_1 j_2 \ldots j_n$ are fixed-point fractional numbers. From (11) and (19), a new expression is obtained:

$$QFT|j_1\, j_2 \ldots j_n\rangle = \quad (20)$$

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i k_1 (0.j_n)} \cdot e^{2\pi i k_2 (0.j_{n-1} j_n)} \cdot \ldots \cdot e^{2\pi i k_n (0.j_1 j_2 \ldots j_n)} |k_1 \ldots k_{n-1} k_n\rangle.$$

The second member of (20) can be decomposed into a product of quantum states:

$$QFT|j_1\, j_2 \ldots j_n\rangle = \tag{21}$$

$$\frac{1}{\sqrt{N}}\left(|0\rangle + e^{2\pi i(0.j_n)}|1\rangle\right)\times\left(|0\rangle + e^{2\pi i(0.j_{n-1}j_n)}|1\rangle\right)\times \ldots \times\left(|0\rangle + e^{2\pi i(0.j_1 j_2 \ldots j_n)}|1\rangle\right)$$

(compute the coefficient of $|k_1\, k_2 \ldots\, k_n\rangle$ in (21) and check that it is the same as in (20)).

Instead of synthesizing the operator (21), the following is synthesized:

$$QFT|j_1\, j_2 \ldots j_n\rangle = \tag{22}$$

$$\frac{1}{\sqrt{N}}\left(|0\rangle + e^{2\pi i(0.j_1 j_2 \ldots j_n)}|1\rangle\right)\times\left(|0\rangle + e^{2\pi i(0.j_2 \ldots j_n)}|1\rangle\right)\times \ldots \times\left(|0\rangle + e^{2\pi i(0.j_n)}|1\rangle\right).$$

The only difference is the order of the factors. If necessary, a final permutation might be applied.

The transformation (22) is executed in several steps. First, modify the first qubit state, without changing the other qubits states:

$$|j_1\, j_2 \ldots j_n\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1 j_2 \ldots j_n)}|1\rangle) \times |j_2 \ldots j_n\rangle. \tag{23}$$

This operation can be decomposed as follows:

$$H(|j_1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1)}|1\rangle),$$

$$CR_{\pi/2}[|j_2\rangle, \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1)}|1\rangle)] = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1 j_2)}|1\rangle),$$

$$CR_{\pi/4}[|j_3\rangle, \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1 j_2)}|1\rangle)] = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1 j_2 j_3)}|1\rangle), \tag{24}$$

and so on. Observe that only the first qubit state is modified. The others are control qubits. At the end of this step, the n-qubit register state is

$$|t_1\rangle \times |j_2 \ldots j_n\rangle \text{ with } |t_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_1 j_2 \ldots j_n)}|1\rangle). \tag{25}$$

The second step modifies the second qubit state, without modifying the other qubit state:

$$|t_1\rangle \times |j_2 \ldots j_n\rangle \rightarrow |t_1\rangle \times \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2 \ldots j_n)}|1\rangle) \times |j_3 \ldots j_n\rangle. \tag{26}$$

This operation can be decomposed as follows:

$$H(|j_2\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2)}|1\rangle),$$

$$CR_{\pi/2}[|j_3\rangle, \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2)}|1\rangle)] = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2 j_3)}|1\rangle),$$

$$CR_{\pi/4}[|j_4\rangle, \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2 j_3)}|1\rangle)] = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2 j_3 j_4)}|1\rangle), \quad (27)$$

and so on. At the end of this step, the n-qubit register state is

$$|t_1\rangle \times |t_2\rangle \times |j_3 \ldots j_n\rangle \text{ with } |t_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.j_2 \ldots j_n)}|1\rangle). \quad (28)$$

Thus, using the relations (24), (27), and so on, a circuit that implements the operation (22), composed of H gates and of controlled rotations is obtained. An example, with n = 4, is shown in Fig.6. It executes the *QFT* operation, without the final permutation. If it were necessary, depending on the application, the permutation might be implemented with *SWAP* gates.

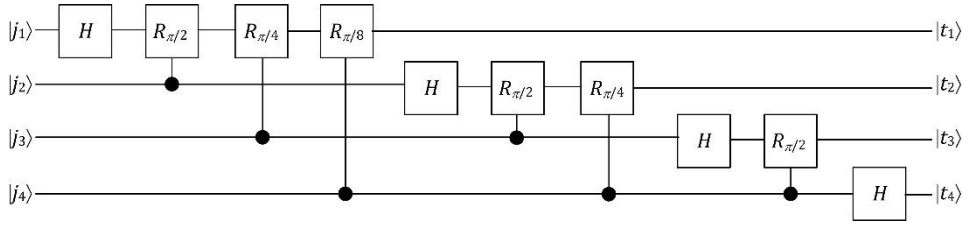

Figure 6 Quantum Fourier transform (n = 4)

The matrices that define the *QFT* and QFT^{-1} operators are deduced from the definitions (11) and (13):

$$QFT[j,k] = \frac{1}{\sqrt{N}}\omega_N^{(jk) \bmod N}, \quad QFT^{-1}[j,k] = \frac{1}{\sqrt{N}}\omega_N^{(-kj) \bmod N}. \quad (29)$$

To execute the inverse transform QFT^{-1}, the circuit of Fig.6 must be inverted, what means: executing the operations in reverse order and replacing R_φ by $R_{-\varphi}$. The circuit (n= 4) is shown in Fig.7.

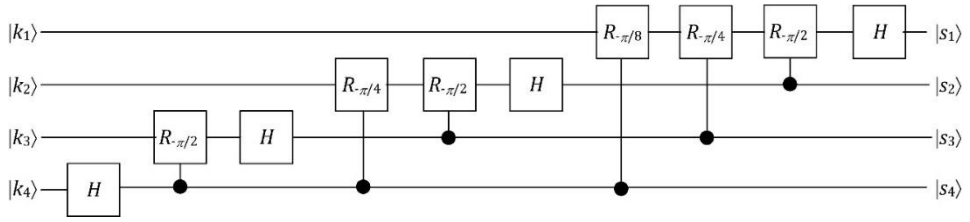

Figure 7 Inverse quantum Fourier transform (n = 4)

Comments 2

- The *QFT* is a generalization of the Hadamard operation. In other word, the operation H is a 1-qubit *QFT*. Applied to n qubits, it transforms basic states $|j_1 j_2 ... j_n\rangle$ into superposition states characterized by their relative phases $j \cdot k/N$, k = 0 to N-1.
- The *QFT* operator is used in circuits that execute algorithms such as the phase estimation (Sec.4) and the computation of the order of a natural (Sec.5). It can also be used in circuits that execute arithmetic operations: natural numbers can be encoded by angles, and rotations amounts to angle additions ([3], [4], [5]).

4. Phase estimation algorithm

This is another generic algorithm. It is used as a computation primitive in e.g. cryptography (Sec.5) or linear algebra algorithms [6].

4.1. Problem statement

Consider an m-qubit register. Assume that several circuits, able to execute operations on this register, are available (the black boxes of Fig.8):

- controlled operations CU^n, with n = 1, 2, 4, 8, etc., being U an unknown unitary operation,
- an initialization operator that sets the m-qubit register into a state $|u\rangle$ which is an eigenvector of the unknown unitary operation U.

The stated problem is the computation of the eigenvalue of $|u\rangle$. It is important to understand that U is an unknown transformation and that

the circuits of Fig.8 are black boxes (oracles). If U were a known linear transformation, the stated problem would amount to the computation of the eigenvalues of U, that is, a classical linear algebra problem.

ome

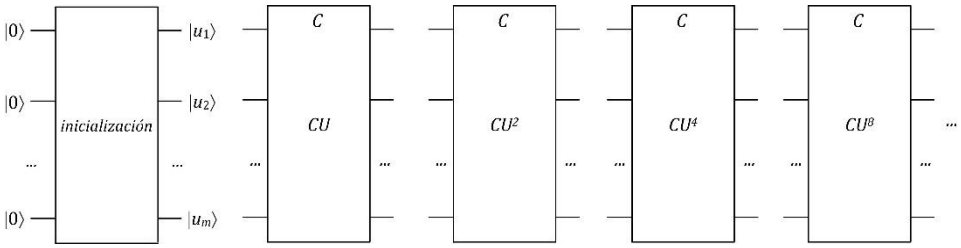

Figure 8 Black boxes

This problem, stated in terms of digital circuits, have a trivial solution (Fig.9.a). Use the initialization black box to generate an eigenvector u. Compute $U(u)$ with the other black box. By definition of the eigenvectors of a linear transformation, it remains to compute λ such that $U(u) = \lambda u$. It amounts to a division over the complex field.

If similar operations are executed with the quantum black boxes (Fig.9.b), some values z and w are obtained when the final quantum states are measured. Assume that

$$|u\rangle = \sum_{j=0}^{M-1} a_j \, |j_1 \dots j_{m-1} \, j_m\rangle, \tag{30}$$

($M = 2^m$). Then, taking into account that the eigen values λ of a unitary operator are unitary (complex numbers whose norm is equal to 1), $\lambda = e^{i\varphi}$ for some angle φ, so that

$$U|u\rangle = e^{i\varphi}|u\rangle = \sum_{j=0}^{M-1} e^{i\varphi} a_j \, |j_1 \dots j_{m-1} \, j_m \rangle. \tag{31}$$

Compare (30) and (31): the states $|u\rangle$ and $U|u\rangle$ only differ by the phases of their coefficients, not by the probabilities of measuring a particular value. Thus, the probability $|a_j|^2$ of measuring a particular value $(j_1, \dots, j_{m-1}, j_m)$ at the output w, is the same as the probability $|e^{i\varphi} a_j|^2 = |a_j|^2$ of measuring

the same value at the output *z*. The measurement of *w* and *z* doesn't give any information about the value of φ. In other words, in (31) φ is a global phase and is not observable. To make possible the measurement of φ, the global, unobservable, phase must be translated into relative, observable, phases, using the same technique as in the Deutsch algorithm. For that, controlled operators CU^n and the quantum Fourier transform are used.

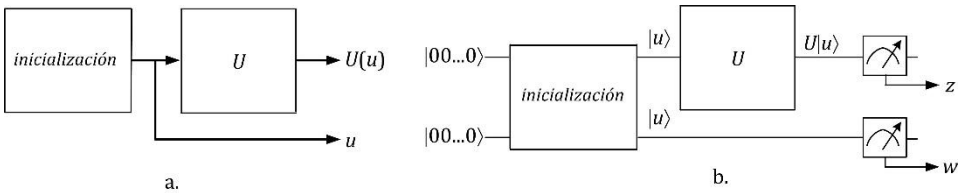

Figure 9 Comparison between digital and quantum circuit

4.2. Solutions

A first solution is shown in Fig.10, with m+1 qubits and only one controlled CU operator. The first qubit is set into state $|+\rangle$ with a Hadamard operator. Then the controlled CU operator executes the following transformation:

$$|+\rangle\times|u\rangle = \frac{1}{\sqrt{2}}|0\rangle\times|u\rangle + \frac{1}{\sqrt{2}}|1\rangle\times|u\rangle \xrightarrow{CU} \frac{1}{\sqrt{2}}|0\rangle\times|u\rangle + \frac{1}{\sqrt{2}}|1\rangle\times U|u\rangle =$$

$$(\frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\varphi}}{\sqrt{2}}|1\rangle)\times|u\rangle. \qquad (32)$$

Thus, the final state of the first qubit is (Fig.10.a)

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\varphi}}{\sqrt{2}}|1\rangle. \qquad (33)$$

In order to observe the relative phase φ, and additional Hadamard operator is used and the first qubit state is measured (Fig.10.b):

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\varphi}}{\sqrt{2}}|1\rangle \xrightarrow{H} \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{e^{i\varphi}}{2}|0\rangle - \frac{e^{i\varphi}}{2}|1\rangle = \frac{1+e^{i\varphi}}{2}|0\rangle + \frac{1-e^{i\varphi}}{2}|1\rangle. \quad (34)$$

Define

$$p(0) = \left|\frac{1+e^{i\varphi}}{2}\right|^2 = \frac{1+\cos\varphi}{2},\ p(1) = \left|\frac{1-e^{i\varphi}}{2}\right|^2 = \frac{1-\cos\varphi}{2}. \qquad (35)$$

The measurement result is 0, with probability $p(0)$, and 1, with probability $p(1)$. If the circuit of Fig.10.b is reset N times, and if the result 0 is obtained N_0 times and the result 1 is obtained N_1 times, then

$$\varphi \cong \cos^{-1}\left(\frac{2N_0}{N} - 1\right). \qquad (36)$$

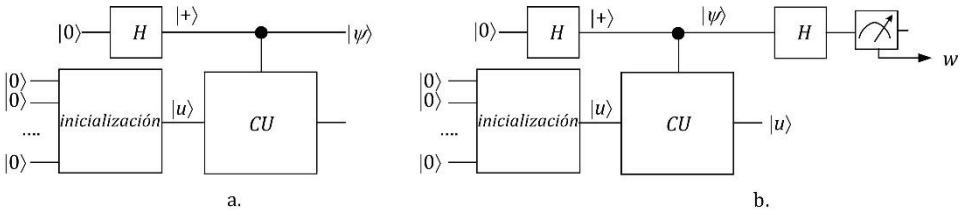

Figure 10 Relative phase generation

As in the case of the Deutsch algorithm (Sec.2, comments 1), the first qubit is a control qubit and the others are the targets. The execution of $U|u\rangle$, under the control of the first qubit, initially set into a superposition state, modifies the relative phases of this qubit. This is another example of the phase kickback effect.

If, as initially assumed (Fig.8), the controlled operators CU^2, CU^4, CU^8, ... , are available, then the estimation of φ can be performed in a single step. Define

$$\phi = \varphi/2\pi, \qquad (37)$$

so that ϕ is a non-negative real number smaller than 1. Assume that it can be expressed with t fractional bits:

$$\phi = 0.\phi_1\phi_2 \ldots \phi_t,\ 2^t\phi = \phi_1 2^{t-1} + \phi_2 2^{t-2} + \ldots + \phi_{t-1}2 + \phi_t. \qquad (38)$$

The quantum circuit of Fig.11 computes ϕ. It is composed of a t-qubit register, initially set to $|00 \ldots 0\rangle$, and an m-qubit register initially prepared in state $|u\rangle$ with the initialization operator of Fig.8. The t-qubit register is set into a superposition of all 2^t basic states with Hadamard operators.

Then, using the other operators of Fig.8, unitary operations U, U^2, U^4, … , are executed on the m-qubit register, under the control of the first register qubits. The following transformations are executed:

$$|00\ldots0\rangle\times|u\rangle \xrightarrow{H^{\times t}} \frac{1}{\sqrt{2^t}}[(|0\rangle + |1\rangle) \times \ldots \times (|0\rangle + |1\rangle) \times (|0\rangle + |1\rangle)]\times|u\rangle$$

$$\xrightarrow{CU} \frac{1}{\sqrt{2^t}}[(|0\rangle + |1\rangle) \times \ldots \times (|0\rangle + |1\rangle) \times (|0\rangle + e^{2\pi i\phi}|1\rangle)]\times|u\rangle$$

$$\xrightarrow{CU^2} \frac{1}{\sqrt{2^t}}[(|0\rangle + |1\rangle) \times \ldots \times (|0\rangle + e^{4\pi i\phi}|1\rangle) \times (|0\rangle + e^{2\pi i\phi}|1\rangle)]\times|u\rangle$$

…

$$\xrightarrow{CU^{2^{t-1}}} \frac{1}{\sqrt{2^t}}[(|0\rangle + e^{2^t\pi i\phi}|1\rangle) \times \ldots \times (|0\rangle + e^{4\pi i\phi}|1\rangle) \times (|0\rangle + e^{2\pi i\phi}|1\rangle)]\times|u\rangle. \quad (39)$$

Define $k = k_1 2^{t-1} + k_2 2^{t-2} + \ldots + k_t 2^0$ and observe that the state $|\psi\rangle$ of the t-qubit register is

$$|\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i k\phi} |k_1 \ldots k_{t-1}\, k_t\rangle = \quad QFT|\phi_1\phi_2 \ldots \phi_t\rangle. \quad (40)$$

Thus, the controlled operation U^n, applied to the eigenvector $|u\rangle$ under the control of the t-qubit register, have generated ϕ under the form of relative phases $2\pi k\phi$, with k = 0 to 2^{t-1}. This is information is not directly measurable as all the coefficients of (40) have the same norm. To decode it, the inverse quantum Fourier transform is used:

$$|\phi_1\phi_2 \ldots \phi_t\rangle = QFT^{-1}|\psi\rangle. \quad (41)$$

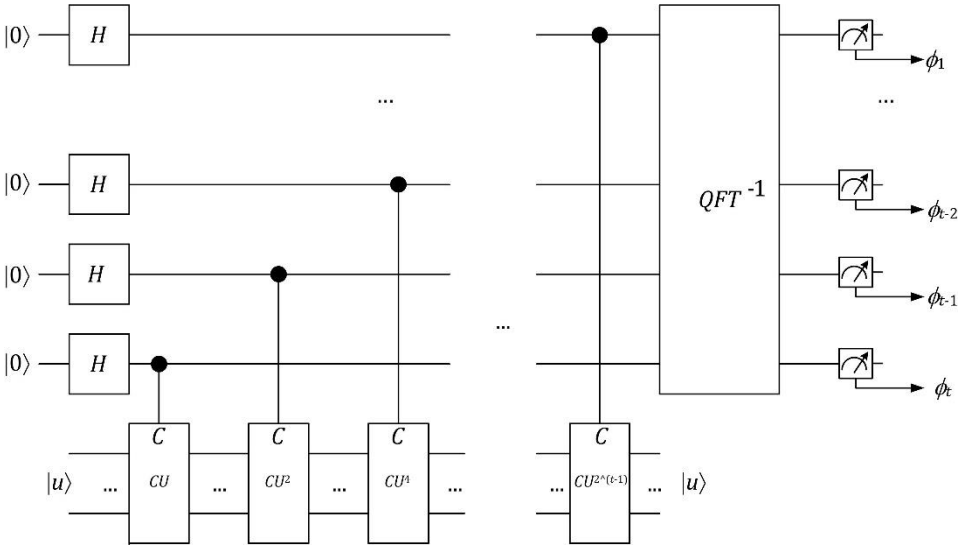

Figure 11 Phase estimation circuit

If ϕ cannot be expressed with t fractional qubits, that is, if

$$\phi \cong \bar{\phi} = 0.\phi_1\phi_2 \ldots \phi_t,\ 2^t\bar{\phi} = \phi_1 2^{t-1} + \phi_2 2^{t-2} + \ldots + \phi_{t-1}2 + \phi_t,$$

then the final state of the t-qubit register is no longer a basic state like (40). It can be demonstrated that

$$QFT^{-1}|\psi\rangle = \sum_{j=0}^{2^t-1} a_j |j_1 \ldots j_{t-1}\, j_t\rangle, \tag{42}$$

where

$$a_j = \frac{1}{2^t}\left(\sum_{k=0}^{2^t-1} \omega_{2^t}^{k(2^t\phi - j)}\right). \tag{43}$$

The final state is a superposition state. The measurement can generate several values of $\bar{\phi} = 0.\phi_1\phi_2 \ldots \phi_t$, with different probabilities. A detailed analysis shows that to get an estimation of ϕ, with a precision of s fractional bits, and with a probability greater than $1 - \varepsilon$, the number t of qubits of the first register must satisfy the following condition

$$t \geq s + log_2\left(2 + \frac{1}{2\varepsilon}\right). \tag{44}$$

Comment 3

The circuit of Fig.11 computes an eigenvalue of U if the second register can be initialized in a state $|u\rangle$ which is an eigenvector of U. It doesn't solve a more general problem that is the computation of an eigenvector of U and of the corresponding eigenvalue. However, the following property has been demonstrated: if the initialization operator generates a state $|u\rangle$ that is a linear combination of eigenvectors, that is,

$$|u\rangle = \sum_j c_j |u_j\rangle, \tag{45}$$

where each state $|u_j\rangle$ is an eigenvector of U whose corresponding eigenvalue is $e^{2\pi i \phi_j}$, then the circuit of Fig.11 generates an estimation of ϕ_j, with a probability equal to

$$p = |c_j|^2 (1 - \varepsilon), \tag{46}$$

and a precision of s fractional bits if the number t of qubits of the first register satisfies the condition (44).

5. Order of a natural number

The computation of the order of a natural is a problem that a quantum circuit can solve more efficiently than a digital computer. Its practical interest is that this operation is part of a method of factorization of natural numbers which, in turn, is the base of some ciphering algorithms.

5.1. Factorization of natural numbers

The factorization problem is stated as follows: given a non-prime natural number N, find a non-trivial divider of N (neither 1, nor N). As a previous step, the following conditions are checked: N must be odd, and there is no natural $s \geq 3$ such that $N = s^k$ for some natural k. If it were the case, then, as N is odd, the following conditions would hold: $k = log_s N$, $s \geq 3$, so that $k \leq log_3 N$. To summarize, the previous step consists in checking that

- N mod 2 = 1,

- $s = \sqrt[k]{N}$ is not an integer number, whatever the natural k belonging to the interval $2 \le k \le log_3 N$.

If N is even, then 2 is a factor of N, and if $N = s^k$ with $s \ge 3$, then s is a factor of N. If none of the two preceding conditions hold, then N has at least two different prime factors p_1 and p_2, so that

$$N = p_1^{n_1} p_2^{n_2} q \tag{47}$$

where q includes all the other prime factors. Thus, N can be expressed as a product of two coprime factors:

$$N = \left(p_1^{n_1} q_1\right) \cdot \left(p_2^{n_2} q_2\right) \tag{48}$$

where q_1 and q_2 have no common factors.

Define the set $Z_N^{\times}$ of all natural numbers smaller than N and coprime with N:

$$Z_N^{\times} = \{a: 1 \le a < N, gcd(a, N) = 1\}. \tag{49}$$

It is easy to demonstrate that if a_1 and a_2 are elements of $Z_N^{\times}$, then $a_1 \cdot a_2 \bmod N$ is also an element of $Z_N^{\times}$. Thus, $Z_N^{\times}$ is a commutative group. The number of elements of this group is the Euler φ function.

Let a be an element of $Z_N^{\times}$. Its order is the smallest positive integer such as

$$a^r \bmod N = 1. \tag{50}$$

A general property of finite groups is that the order of an element is a divider of the number of group elements. In this case, the order r of any element a is a divider of $\varphi(N)$. In particular,

$$r < \varphi(N). \tag{51}$$

The previous relation reduces the range of possible values of r. The Shor algorithm is based on a statistical property: the probability that a randomly selected element a of $Z_N^{\times}$ satisfies the conditions

$$r \bmod 2 = 0,\ a^{r/2} + 1 \bmod N \neq 0, \tag{52}$$

being r the order of a, is high. If a satisfies the preceding conditions (52), then $r/2$ is an integer number and, by definition of the order r of a,

$$(a^{r/2} + 1)(a^{r/2} - 1) = a^r - 1 \equiv 0 \bmod N, \tag{53}$$

so that there exists an integer m such that

$$(a^{r/2} + 1)(a^{r/2} - 1) = mN. \tag{54}$$

Furthermore, $a^{r/2} - 1 \bmod N \neq 0$; in the contrary case, the order of a is $r/2$ instead of r. As none of the natural numbers $a^{r/2} + 1$ and $a^{r/2} - 1$ is a multiple of N, then, according to (54), the prime factors of N are distributed between $a^{r/2} + 1$ y $a^{r/2} - 1$. In conclusion, the greatest common dividers $gcd(a^{r/2} + 1, N)$ and $gcd(a^{r/2} - 1, N)$ are non-trivial dividers of N: they divide N and they don't include all the prime factors of N.

As an example, compute a divider of 45. The set $Z_{45}^{\times}$ defined by (49) is

$$Z_{45}^{\times} = \{1, 2, 4, 8, 16, 32, 7, 14, 28, 11, 22, 44, 13, 26, 17, 34, 19, 38, 23, 29, 31, 37, 41, 43\},$$

so that $\varphi(45) = 24$. Choose $a = 2$. The order of 2 is 12 ($2^{12} = 4096 = 91 \cdot 45 + 1$). Check that $2^6 + 1 = 65$ is not a multiple of 45 and that the conditions (52) are satisfied. Compute

$$gcd(2^6 + 1, 45) = 5, \ gcd(2^6 - 1, 45) = 9, \tag{55}$$

and check that 5 and 9 are non-trivial dividers of 45.

To summarize, the Shor algorithm [7] executes the following operations:

- Check that $N \bmod 2 = 1$. If not, 2 and $N/2$ non-trivial factors.
- Check that $N \neq s^k$ con $2 \leq k \leq log_3 N$. If not, s and s^{k-1} are non-trivial factors.
- Randomly select a natural number a.
- If the greatest common divider of a and N is greater than 1, then $gcd(a, N)$ and $N/gcd(a, N)$ are non-trivial dividers.

- Compute the order r of a. If the conditions (52) hold, $gcd(a^{r/2} + 1, N)$ and $gcd(a^{r/2} - 1, N)$ are non-trivial dividers. If not, choose another value of a and go back to the penultimate step (random selection of a).

For all steps, but the computation of the order of a, algorithms with a polynomial time complexity have been defined. For example, the greatest common divider can be computed with the Euclidean algorithm. However, there is no known algorithm that computes the order of a group element with a polynomial time complexity. In the next section, a quantum circuit that computes the order of an element of $Z_N^{\times}$, with a delay proportional to n^3, being n the number of bits of N, is described. Thus, using this quantum circuit as a coprocessor, it might be possible to solve the factorization problem in a polynomial time.

5.2 Order of a natural

Let N be a natural and $Z_N^{\times}$ the multiplicative group defined by (49). Assume that $N \leq 2^n$ so that the elements of $Z_N = \{0, 1, \ldots, N\text{-}1\}$, a set that contains $Z_N^{\times}$, are represented with n bits. Define a quantum circuit composed of n qubits whose basic states $|x_1\, x_2 \ldots x_n\rangle$ represent the elements of Z_N. Let U be an operator that executes the following transformation of the basic states:

$$U|x_1\, x_2 \ldots x_n\rangle = |y_1\, y_2 \ldots y_n\rangle, \tag{56}$$

where

$$y_1 2^{n-1} + y_2 2^{n-2} + \ldots + y_n 2^0 = a(x_1 2^{n-1} + x_2 2^{n-2} + \ldots + x_n 2^0) \bmod N, \tag{57}$$

being a the element of $Z_N^{\times}$ whose order must be computed. In a concise way:

$$U|x\rangle = |ax \bmod N\rangle, \forall x \in Z_N. \tag{58}$$

It can be proven (properties 7.2 and 7.3 of [1]) that U is a unitary operator and that, if the order of a is equal to r, then the vectors

$$|u_j\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega_r^{-kj} \, |a^k \bmod N\rangle, j \in \{0, 1, \ldots, r\text{-}1\}, \omega_r = e^{\frac{2\pi i}{r}}, \quad (59)$$

are eigenvectors of U with eigenvalues equal to

$$\omega_r^j = e^{\frac{2\pi i}{r} j} = e^{2\pi i \phi_j} \text{ con } \phi_j = j/r. \quad (60)$$

Being U a unitary operator, it is possible to define a quantum circuit that executes U as well as all the controlled operations CU, CU^2, CU^4, etc. Therefore, the circuit of Fig.11 can be implemented with quantum gates. If it were possible to prepare the system into the state $|00...0\rangle \times |u_j\rangle$, then a t-bit estimation of $\phi_j = j/r$ could be obtained, but this is obviously impossible given that the value of r is included within the definition (59) of $|u_j\rangle$. However, it can be demonstrated that

$$\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |u_j\rangle = |00 \ldots 01\rangle. \quad (61)$$

Thus, according to the comment 3 and to the relations (45) and (46), if the initial state of the circuit of Fig.11 is $|00 \ldots 00\rangle \times |00 \ldots 01\rangle$, then an estimation $\bar{\phi}$ of one of the phases $\phi_j = j/r$ is obtained (Fig.12). To deduce the value of r from the measurement of the t-qubit register, the following steps are performed:

- If $\bar{\phi} = 0$, that is, if the measured eigenvalue corresponds to the eigenvector $|u_0\rangle$, repeat the quantum algorithm execution until $\bar{\phi} \neq 0$.
- If $\bar{\phi} \neq 0$, then $\bar{\phi}$ is an estimation of j/r. The probability of measuring a particular value of j/r is the same, $(1 - \varepsilon)/r$, whatever the value of j. So, no information about the value of j can be deduced. Nevertheless, taking into account that $\bar{\phi} = 0.\phi_1\phi_2 \ldots \phi_t$ is an estimation of a quotient j/r, where j and r are integers belonging to the range $0 < j < r < N$, it is possible to compute two naturals x and z that satisfy the conditions

  $$0 < x < z < N, x/z \cong \bar{\phi}, \; gcd(x, z) = 1. \quad (62)$$

  For that, an algorithm $cf(y,Q,N)$, based on the computation of continuous fractions is used ([8], ejemplo7.9.py). It returns two

integers that satisfy the conditions $0 < x < z < N$, $x/z \cong y/Q$, $gcd(x, z) = 1$. Thus, $cf(2^t\bar{\phi}, 2^t, N)$ generates x and z that satisfy (62). The parameter N could even be replaced by $\varphi(N)$.

- Two integers x and z have been computed, such that $x/z \cong j/r$. It is likely that either $r = z$ or that r is a multiple of z smaller than $\varphi(N)$.
- If neither z, nor any multiple of z smaller than N, is the order of a, repeat the execution of the algorithm.

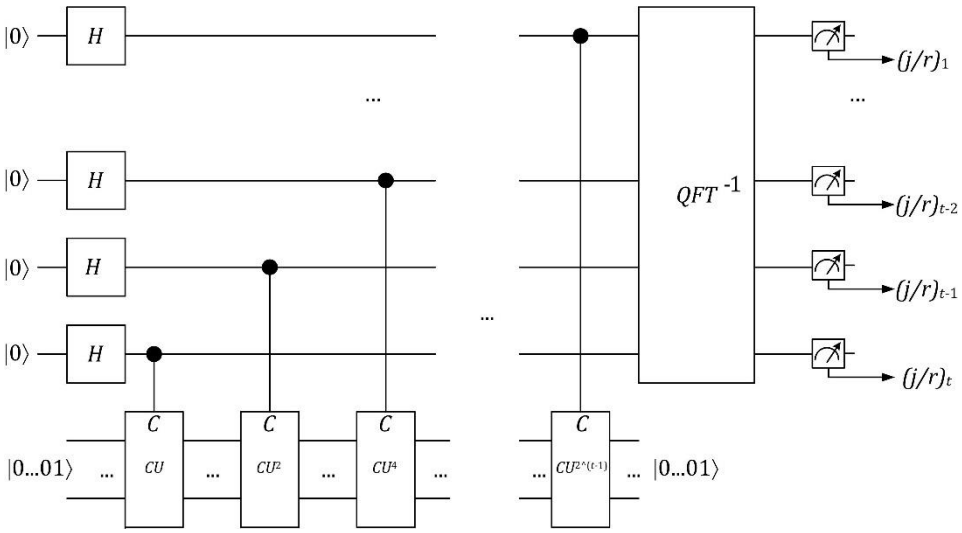

Figure 12 Estimation o j/r

Consider an example. The circuit of Fig.12, with $a = 29$, $t = 8$, $N = 45$, $\varphi(N) = 24$ and $n = 6$, has been simulated. The most likely non-zero measurement results are

$$2^t\bar{\phi} = 128, 85, 43, 213 \text{ and } 171.$$

Then, the execution of the function $cf(2^t\bar{\phi}, 2^t, N)$ gives the following results:

```
cf(128,256,24)=(1,2)
cf(85,256,24)=(1,3)
cf(43,256,24)=(1,6)
cf(213,256,24)=(5,6)
```

```
cf(171,256,24)=(2,3)
```

The candidate values of r are 2, 3, 6, and their multiples smaller than $\varphi(45) = 24$. Actually, the order of 29 mod 45 is 6:

$$29^6 = 594.823.321 = 45 \cdot 13.218.296 + 1.$$

5.3. Complexity

Analize the time complexity of the circuit of Fig.12. It is made up of two subcircuits: the inverse quantum Fourier transform over the t-qubit register and the U^k operations on the n-qubit register under the control of the t-qubit register. The circuit of Fig.7 executes $1 + 2 + ... + t = t(t\text{-}1)/2$ unary and binary operations. The corresponding time complexity is proportional to t^2. Let $|x_1\, x_2\, ...\, x_n\rangle$ and $|c_1\, c_2\, ...\, c_t\rangle$ be the basic states of the n-qubit and t-qubit registers, respectively. Denote by x is the natural represented by $x_1\ ...\ x_{n\text{-}1}\ x_n$. According to (58), the transformations executed by the CU^k operators are

$$|x_1\, x_2\, ...\, x_n\rangle \times |c_1\rangle \xrightarrow{CU} |a^{c_1} x \bmod N\rangle \times |c_1\rangle,$$

$$|x_1\, x_2\, ...\, x_n\rangle \times |c_2\rangle \xrightarrow{CU^2} |(a^{c_2})^2 x \bmod N\rangle \times |c_2\rangle,$$

$$...$$

$$|x_1\, x_2\, ...\, x_n\rangle \times |c_t\rangle \xrightarrow{CU^{2^{t-1}}} |(a^{c_t})^{2^{t-1}} x \bmod N\rangle \times |c_t\rangle. \qquad (63)$$

Thus, the set of CU^k operators executes the following transformation:

$$|x_1\, x_2\, ...\, x_n\rangle \times |c_1\, c_2\, ...\, c_t\rangle \xrightarrow{CU^{2^{t-1}} ... CU^2 CU}$$

$$|(a^{c_t})^{2^{t-1}} ... (a^{c_2})^{2^1} (a^{c_1})^{2^0} x \bmod N\rangle \times |c_1\, c_2\, ...\, c_t\rangle = |a^c \cdot x \bmod N\rangle \times |c\rangle, \qquad (64)$$

where c is the integer represented by $c_t\ ...\ c_2\ c_1$. The following scheme computes $a^c \cdot x \bmod N$:

- first compute

$$a_2 = a \cdot a \bmod N,\ a_4 = a_2 \cdot a_2 \bmod N,\ \ldots,\ a_{2^{t-1}} = a_{2^{t-2}} \cdot a_{2^{t-2}} \bmod N; \quad (65)$$

- then compute

$$a^c \cdot x \bmod N = a^{c_1}\ a_2{}^{c_2}\ a_4{}^{c_3}\ \ldots\ a_{2^{t-1}}{}^{c_t}\ x \bmod N. \quad (66)$$

The computation of (65) consists of t mod N squarings, and the computation of (66) consists of t mod N products. Those operations (squaring, product, mod N reduction) on n-bit numbers, have a time complexity proportional to n^2, and the computation of (66) has a time complexity proportional to $t \cdot n^2$. Assuming that a quantum circuit can compute (66) with a time complexity similar to that of a digital circuit (Chap.6, Sec.1), the conclusion is that the circuit of Fig.12 has a time complexity proportional to $\alpha \cdot t^2 + \beta \cdot t \cdot n^2$, being α and β coefficients that correspond to the inverse transform and to the computation of (66), respectively.

As regards the function *cf* (continuous fractions), it can be demonstrated that it must be executed using an estimation $\bar{\phi}$ of j/r with $2n+1$ fractional bits. Thus, according to (44),

$$t \geq\ 2n + 1 +\ log_2\left(2 + \frac{1}{2\varepsilon}\right). \quad (67)$$

In conclusion, for great values of n, the value of t is proportional to n, so that the time complexity of the circuit is proportional to n^3, that is, a polynomial complexity.

6. Search algorithm

Consider a function f: {0, 1, 2, ... , N-1} $\rightarrow$ {0, 1} and a digital circuit that computes f. The problem statement is the following: find a solution x of $f(x) = 1$. It could seem trivial. Nevertheless, if f is an unknown function and if the only resource is the circuit itself (an oracle), the solution is the execution of an algorithm such as

```
for x in {0, 1, 2, ... , N-1}:
   if f(x) = 1:
      return x
      exit
```

The average computation time is proportional to $N/2$. This problem could be stated in a slightly different way. Let f: {0, 1, 2, ... , N-1} $\rightarrow$ {0, 1} be a one-way function, relatively easy to compute but practically impossible to invert. To find a solution of $f(x) = 1$, a circuit that computes f is implemented, and the preceding algorithm is executed. In this section, a quantum algorithm that computes a solution of $f(x) = 1$ and has a time complexity proportional to $N^{0.5}$, instead of N, is proposed. Its execution needs the previous knowledge of the number M of solutions.

6.1. Grove algorithm

Let $N = 2^n$. Consider an $(n+1)$-qubit circuit whose basic states are $|x_1\, x_2 \ldots x_n\, x_{n+1}\rangle$. Define a unitary operator U_f that executes the following transformation of the basic states (Chap.6, Sec.1.1):

$$|x_1\, x_2 \ldots x_n\, x_{n+1}\rangle \xrightarrow{U_f} |x_1\, x_2 \ldots x_n\rangle \times |x_{n+1} \oplus f(x_1, x_2, \ldots, x_n)\rangle. \qquad (68)$$

The circuit that executes the Grove algorithm is shown in Fig.13. The n first qubits constitute the *data* register, initially in state $|00 \ldots 0\rangle$, and set into the superposition state $|+\rangle \times |+\rangle \times \ldots |+\rangle$ with H gates. Thus, according to relation (1), the state $|\psi\rangle$ of the register is

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=00\ldots0}^{x=11\ldots1} |x_1 x_2 \ldots x_n\rangle, \qquad (69)$$

that is a superposition of basic states, all with the same coefficient $\sqrt{1/N}$ and the same probability $1/N$. The last qubit (*target*), initially in state $|1\rangle$, is set to the superposition state $|-\rangle$ with another H gate. The operation G and the number R of iterations are defined later on.

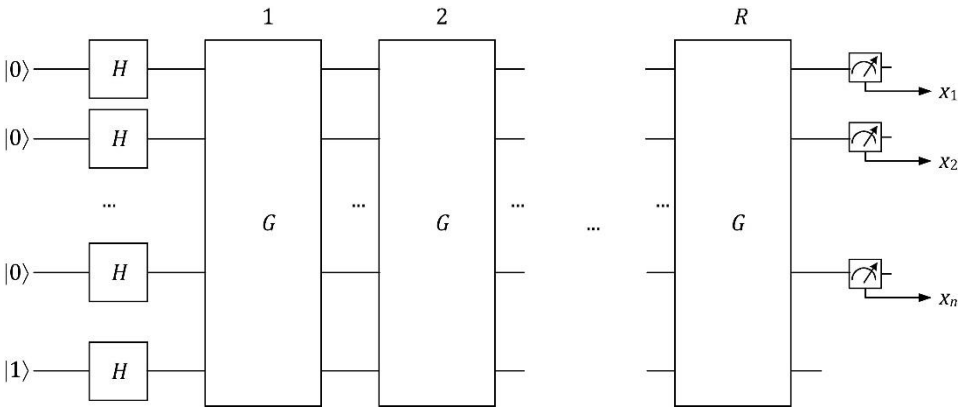

Figure 13 Grove algorithm

Assume that equation $f(x_1, x_2, \ldots, x_n) = 1$ has M solutions, and define two particular states of the data register:

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{f(x_1,x_2,\ldots,x_n)=0} |x_1 x_2 \ldots x_n\rangle,$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{f(x_1,x_2,\ldots,x_n)=1} |x_1 x_2 \ldots x_n\rangle. \qquad (70)$$

The initial state (69) can be expressed as a linear combination of $|\alpha\rangle$ y $|\beta\rangle$:

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle. \qquad (71)$$

The coefficients of $|\alpha\rangle$ and $|\beta\rangle$ are real numbers and the sum of their squares is equal to 1. According to (70) and (71), The result of the measurement of the data register state $|x_1\ x_2\ \ldots\ x_n\rangle$ is a solution of $f(x_1, x_2, \ldots, x_n) = 0$ with a probability equal to $(N\text{-}M)/N$ and a solution of $f(x_1, x_2, \ldots, x_n) = 1$ with a probability equal to M/N. The data register state $|\psi\rangle$ belongs to the dimension-2 subspace W generated by $|\alpha\rangle$ and $|\beta\rangle$. The Grove algorithm executes a series of rotations of $|\psi\rangle$, within W, in such a way that the final state becomes $a|\alpha\rangle + b|\beta\rangle$ with $b^2 >> a^2$, so that the result of the measurement of the data register state $|x_1\ x_2\ \ldots\ x_n\rangle$ is a solution of $f(x_1, x_2, \ldots, x_n) = 1$ with a high probability equal to b^2.

First, compute the action of U_f on the vectors of $W\times|-\rangle$. Given that $|\alpha\rangle$ is a superposition of all basic states $|x_1\, x_2 \ldots x_n\rangle$ such that $f(x_1, x_2, \ldots, x_n) = 0$, and $|\beta\rangle$ a superposition of all basic states such that $f(x_1, x_2, \ldots, x_n) = 1$, then, by definition of U_f (Chap.6, relation 1),

$$U_f|\alpha\rangle\times|-\rangle = |\alpha\rangle\times\left(\frac{1}{\sqrt{2}}|0\oplus 0\rangle - \frac{1}{\sqrt{2}}|1\oplus 0\rangle\right) = |\alpha\rangle\times|-\rangle, \qquad (72)$$

$$U_f|\beta\rangle\times|-\rangle = |\beta\rangle\times\left(\frac{1}{\sqrt{2}}|0\oplus 1\rangle - \frac{1}{\sqrt{2}}|1\oplus 1\rangle\right) = -|\beta\rangle\times|-\rangle. \qquad (73)$$

Thus, the transformation executed by U_f on an element $a|\alpha\rangle + b|\beta\rangle$ of W, is

$$(a|\alpha\rangle + b|\beta\rangle)\times|-\rangle \xrightarrow{U_f} (a|\alpha\rangle - b|\beta\rangle)\times|-\rangle. \qquad (74)$$

The target qubit state is unchanged and the data register state transformation is shown in Fig.14.

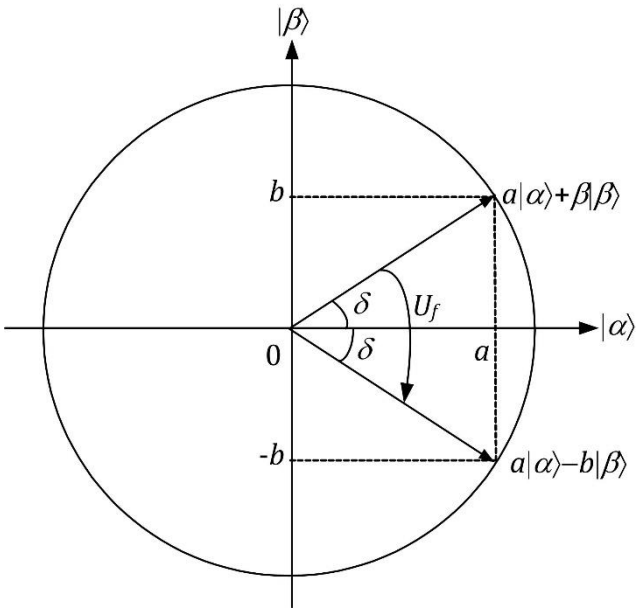

Figure 14 Action of U_f on the data register state

As U_f only transforms the data register state, the relation (74) is rewritten as

$$U_f\,(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle. \qquad (75)$$

The G operation of Fig.13 is graphically defined by Fig.14 and 15:

- first (Fig.7.14), replace $a|\alpha\rangle + b|\beta\rangle$ by $U_f(a|\alpha\rangle + b|\beta\rangle) = a|\alpha\rangle - b|\beta\rangle$;
- then, replace $a|\alpha\rangle - b|\beta\rangle$ by $K|\psi\rangle - (a|\alpha\rangle - b|\beta\rangle)$ where K is a real number such that the resulting vector is unitary.

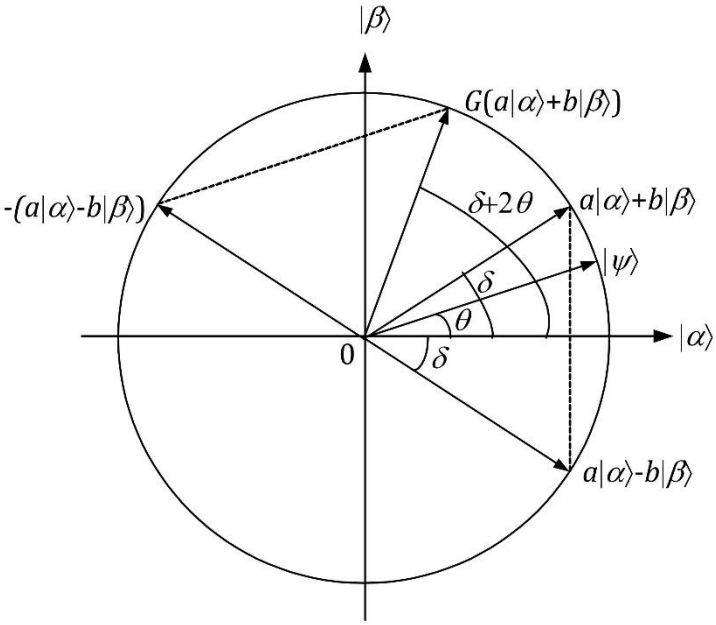

Figure 15 Operation G

It is easy to prove that G is unitary. In particular, by executing on $G(a|\alpha\rangle + b|\beta\rangle)$ the same operations, in reverse order, the initial value $a|\alpha\rangle + b|\beta\rangle$ is obtained. Later, a decomposition of G into unitary operations is described.

Elementary geometric properties show that the angle between $G(a|\alpha\rangle + b|\beta\rangle)$ and $|\alpha\rangle$ is equal to $\delta+2\theta$. The angular distance between $a|\alpha\rangle + b|\beta\rangle$ and $|\beta\rangle$ is equal to $\pi/2 - \delta$, while the distance between $G(a|\alpha\rangle + b|\beta\rangle)$ and $|\beta\rangle$ is equal to $\pi/2 - (\delta+2\theta)$. Thus, the distance has been reduced by 2θ radians, where (71)

$$\theta = \cos^{-1}\sqrt{\frac{N-M}{N}}. \tag{76}$$

The transformations of the data register state, during the algorithm execution, are shown in Fig.16. The initial state $|\psi\rangle$ is defined by (71), so that the initial values of a and b are (Fig.16.a)

$$a = \cos\theta = \sqrt{\frac{N-M}{N}},\ b = \sin\theta = \sqrt{\frac{M}{N}}$$

Each G operation executes a rotation by 2θ radians (Fig.16.b). The number R of steps is chosen in such a way that the final data register state is approximately equal to $|\beta\rangle$, that is,

$$R \cong \frac{\pi}{4\theta} - \frac{1}{2}. \tag{77}$$

In this way, the angular difference ε between $G^R|\psi\rangle$ and $|\beta\rangle$ is smaller than θ , and the probability that the final measurement is a solution of $f(x_1, x_2, \ldots, x_n) = 1$ is

$$p = \cos^2\varepsilon = 1 - \sin^2\varepsilon \geq 1 - \sin^2\theta = 1 - M/N. \tag{78}$$

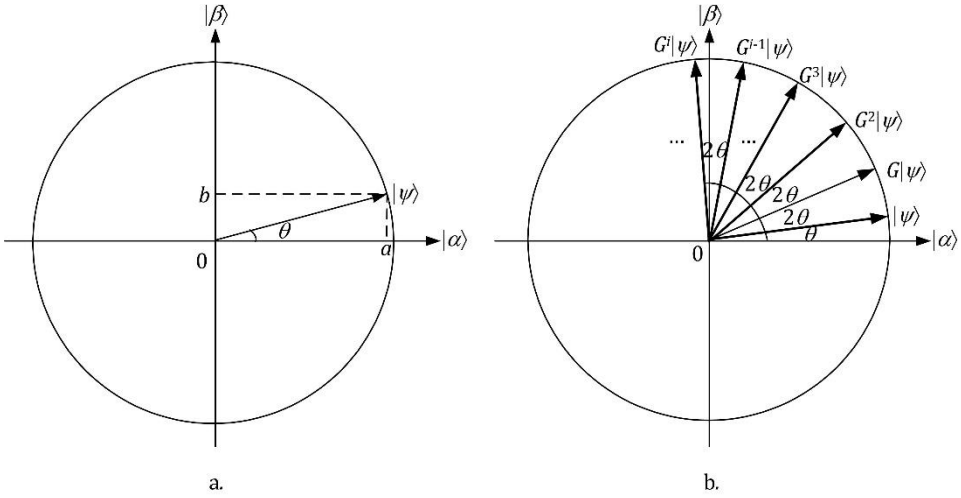

Figure 16 Successive data register states, $a = \sqrt{\frac{N-M}{N}}$ and $b = \sqrt{\frac{M}{N}}$

An upper limit of R is given by the following relations:

$$R = \left\lceil \frac{\pi}{4\theta} \right\rceil \leq \left\lceil \frac{\pi}{4\sin\theta} \right\rceil = \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil. \quad (79)$$

6.2. Operator *G*

It remains to define a circuit that executes the operation G. First define an operator A that transforms the data register basic states as follows:

$$A|00...0\rangle = |00...0\rangle,\ A|x_1\ x_2 ...\ x_n\rangle = -|x_1\ x_2 ...\ x_n\rangle\ \forall |x_1\ x_2 ...\ x_n\rangle \neq |00...0\rangle. \quad (80)$$

To this operator corresponds a diagonal matrix whose first coefficient is equal to 1 and the others are equal to -1. In terms of outer products

$$A = 2|00...0\rangle\langle 00...0| - I. \quad (80)$$

The circuit of Fig.17 implements $-A = e^{j\pi}A$. Consider a basic state $|i_1\ i_2 ...\ i_{n-1}\ i_n\rangle$:

- if $|i_1\ i_2 ...\ i_{n-1}\rangle \neq |0\ 0\ ...\ 0\rangle$, then
  $$|j_1\ j_2 ...\ j_{n-1}\ j_n\rangle = |i_1\ i_2 ...\ i_{n-1}\ i_n\rangle \text{ given that } XX = HH = I;$$
- if $|i_1\ i_2 ...\ i_{n-1}\rangle = |0\ 0\ ...\ 0\rangle$, then, taking into account that $X|-\rangle = -|-\rangle$ and $X|+\rangle = |+\rangle$, the following transformations are executed:

$$|0\ 0\ ...\ 0\ 0\rangle \xrightarrow{X^n} |1\ 1\ ...\ 1\ 1\rangle \xrightarrow{I^{n-1}\times H} |1\ 1\ ...\ 1\rangle\times|-\rangle \xrightarrow{C^{n-1}X} -|1\ 1\ ...\ 1\rangle\times|-\rangle \xrightarrow{I^{n-1}\times H}$$

$$-|1\ 1\ ...\ 1\ 1\rangle \xrightarrow{X^n} -|0\ 0\ ...\ 0\ 0\rangle,$$

$$|0\ 0\ ...\ 0\ 1\rangle \xrightarrow{X^n} |1\ 1\ ...\ 1\ 0\rangle \xrightarrow{I^{n-1}\times H} |1\ 1\ ...\ 1\rangle\times|+\rangle \xrightarrow{C^{n-1}X} |1\ 1\ ...\ 1\rangle\times|+\rangle \xrightarrow{I^{n-1}\times H}$$

$$|1\ 1\ ...\ 1\ 0\rangle \xrightarrow{X^n} |0\ 0\ ...\ 0\ 1\rangle.$$

To summarize, if $|i_1\ i_2 ...\ i_{n-1}\ i_n\rangle \neq |0\ 0\ ...\ 0\ 0\rangle$ the state remains unchanged, and if $|i_1\ i_2 ...\ i_{n-1}\ i_n\rangle = |0\ 0\ ...\ 0\ 0\rangle$ the state becomes $-|0\ 0\ ...\ 0\ 0\rangle$.

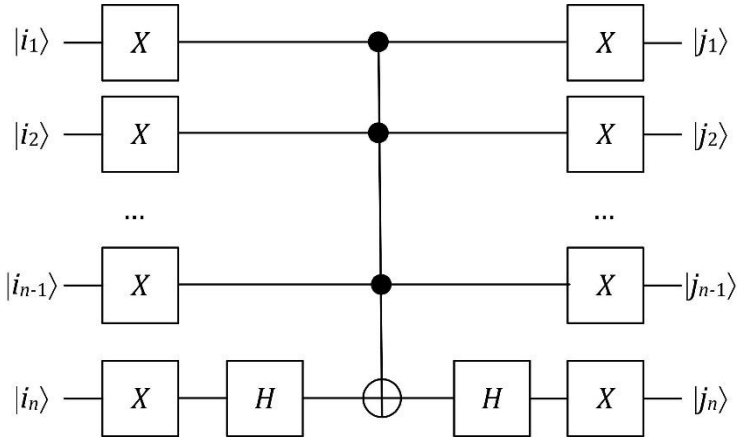

Figure 17 Operator A

Given that the global phase $e^{j\pi}$ is not observable, the circuit of Fig.17 actually implements the operator A. Taking into account that $|\psi\rangle = H^{\times n}|00...0\rangle$ (Fig.12), then, by associativity of the matrix product,

$$H^{\times n}AH^{\times n} = 2H^{\times n}|00...0\rangle\langle 00...0|H^{\times n} - H^{\times n}IH^{\times n} = 2|\psi\rangle\langle\psi| - I. \quad (81)$$

Define

$$G = H^{\times n}AH^{\times n}U_f. \quad (82)$$

This operator executes the following transformation of $a|\alpha\rangle + b|\beta\rangle$:

$$H^{\times n}AH^{\times n}U_f\,(a|\alpha\rangle + b|\beta\rangle) = (2|\psi\rangle\langle\psi| - I)\,(a|\alpha\rangle - b|\beta\rangle) =$$

$$2a|\psi\rangle\langle\psi|\alpha\rangle - 2b|\psi\rangle\langle\psi|\beta\rangle - (a|\alpha\rangle - b|\beta\rangle). \quad (83)$$

The inner products $\langle\psi|\alpha\rangle$ and $\langle\psi|\beta\rangle$ are the projections of $|\psi\rangle$ on $|\alpha\rangle$ and $|\beta\rangle$. According to Fig.16.a and to the definition (76) of θ,

$$\langle\psi|\alpha\rangle = \sqrt{\frac{N-M}{N}},\ \langle\psi|\beta\rangle = \sqrt{\frac{M}{N}}. \quad (84)$$

Thus,

$$G\,(a|\alpha\rangle + b|\beta\rangle) = K|\psi\rangle - (a|\alpha\rangle - b|\beta\rangle), \quad (85)$$

where

$$K = 2(a\sqrt{\frac{N-M}{N}} - b\sqrt{\frac{M}{N}}). \quad (86)$$

From (71) and (85), the values of two coefficients a' and b', such that $G(a|\alpha\rangle + b|\beta\rangle) = a'|\alpha\rangle + b'|\beta\rangle$, can be deduced. The operator G is composed of unitary operators (82), so that it is unitary, and $a'^2 + b'^2 = 1$. This justifies the graphical construction of Fig.15. The circuit that executes the operation G is shown in Fig.18.

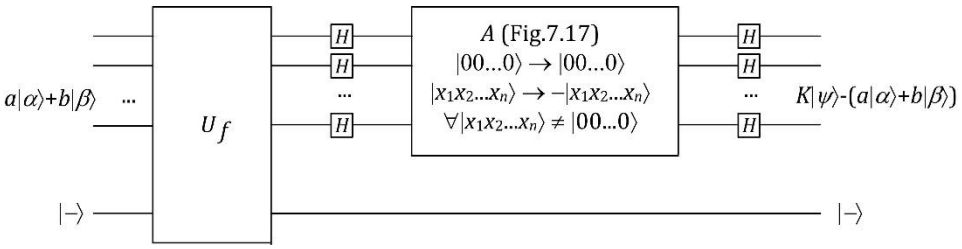

Figure 18 Operator G

Comments 4

- This algorithm is particularly efficient for small values of M/N, as the error probability (78) is smaller than M/N. However, the number R of iteration (79) is greater.
- The time complexity, that is, the number R of iterations, multiplied by the computation time of f (a relatively simple operation), is proportional to $N^{0.5}$. This is the advantage of the quantum circuit with respect to a digital circuit whose complexity would be proportional to N.
- The synthesis of U_f can be done as proposed in Chap.6, Sec.1: first define a digital circuit composed of logic gates, and then translate the digital circuit to a quantum circuit.

[1] J.P.Deschamps, Computación Cuántica, Marcombo, Barcelona, 2023.

[2] Deutsch, D. y Jozsa, R. (1992), «Rapid solution of problems by quantum computation», *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* n.º 439(1907), pp. 553-558.

[3] T. G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033v1, 2000.

[4] L.Ruiz-Perez and J.C.Garcia-Escartin, Quantum arithmetic with the Quantum Fourier Transform, https://arxiv.org/pdf/1411.5949.pdf, 2017.

[5] https://github.com/Marcombo/Circuitos-y-algoritmos-cuanticos, Nota 5.

[6] A.W. Harrow, A. Hassidim, S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. 103(15) (2009) 150502.

[7] Shor, P. (1994), «Algorithm for quantum computation: discrete logarithms and factoring», *Proceedings of the 35th Annual Symposium on Foundations of Computer Science,* IEEE Computer Society, pp. 124-134.

[8] https://github.com/Marcombo/Circuitos-y-algoritmos-cuanticos, ejemplo7.9.py.