Summary of Chapter 6

The synthesis problem can be stated as follows: given a $2^n \times 2^n$ unitary matrix U over the complex field, and a quantum computation system composed of N qubits ($N \geq n$), define a quantum circuit that executes, on a subset of n qubits, the operation defined by U. The circuit is made up of components such as those defined in Chap.4, for example, unary operators (Hadamard, rotations), binary operators (CX, $SWAP$) and ternary operators (C^2X, $CSWAP$).

1. Switching functions

A natural question is whether switching functions f: $\{0,1\}^n \rightarrow \{0, 1\}$ can be implemented by quantum circuits. The answer is yes, but it doesn't mean that quantum computation is an efficient alternative to digital computation if only the implementation of switching functions is considered. Nevertheless, some quantum algorithms include, as part of them, the computation of switching functions. Thus, the synthesis of switching functions with quantum gates is a relevant problem.

1.1. Unitary operator

To implement a switching function f, the first step is the definition of a unitary operation whose execution on a register of qubits emulates a digital circuit that computes f.

Consider a binary function f of n binary variables $x_0, x_1, \ldots, x_{n-1}$. Define an ($n$+1)-qubit register ($q_0, q_1, \ldots, q_{n-1}, q_n$) and an operator U_f that executes the following transformation of the basic register states

$$|x_0 x_1 \ldots x_{n-1}\rangle \times |x_n\rangle \xrightarrow{U_f} |x_0 x_1 \ldots x_{n-1}\rangle \times |x_n \oplus f(x_0, x_1, \ldots, x_{n-1})\rangle, \qquad (1)$$

being x_n an additional binary variable and $\oplus$ the *XOR* (mod 2 addition) function. It can easily be demonstrated that U_f is Hermitian ($U_f = U_f^+$) and unitary ($U_f U_f^+ = I$). In particular, observe that

$$|x_0 x_1 \dots x_{n-1}\rangle \times | x_n \oplus f(x_0, x_1, \dots , x_{n-1})\rangle \xrightarrow{U_f} |x_0 x_1 \dots x_{n-1}\rangle \times |x_n\rangle,$$

so that, whatever the basic state $|b_k\rangle = |x_0 x_1 \dots x_{n-1} x_n\rangle$, $U_f U_f|b_k\rangle = |b_k\rangle$ and, by linearity, this property extends to any quantum state $|\psi\rangle$.

In conclusion, any switching function $f(x_0, x_1, \dots , x_{n-1})$ can be computed by a quantum circuit (Fig.1) that implement the unitary operator U_f defined by (1):

- set the initial state of the register ($q_0, q_1, \dots , q_{n-1}, q_n$) to $|x_0\, x_1 \dots\ x_{n-1}\, 0\rangle$,
- execute U_f,
- measure the final state of q_n.

According to (1) the final state of q_n is $|f(x_0, x_1, \dots , x_{n-1})\rangle$, so that the measurement result is $f(x_0, x_1, \dots , x_{n-1})$.

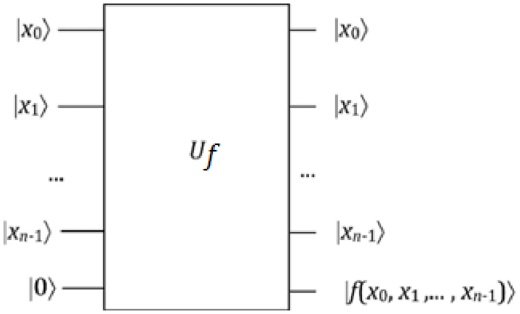

Figure 1 Switching function

1.2. Digital circuit emulation

Any switching function can be implemented by a digital circuit composed of logic gates (*NAND* gates, *NOR* gates, and so on). Thus, in order to compute switching functions using a quantum circuit, a straightforward method is to design a digital circuit composed of logic gates, and then to replace logic gates and connections by quantum gates.

1.2.1. Logic gates

Quantum circuits that compute the logic functions *INV*, *NAND*2, *NOR*2, *AND*2 and *OR*2 are shown in Fig.2. The states $|x\rangle$ and $|y\rangle$ are basic states $|0\rangle$ or $|1\rangle$. The *NOT* gate is implemented with a *CX* gate, and the other gates with *CCX* (Toffoli) operators.

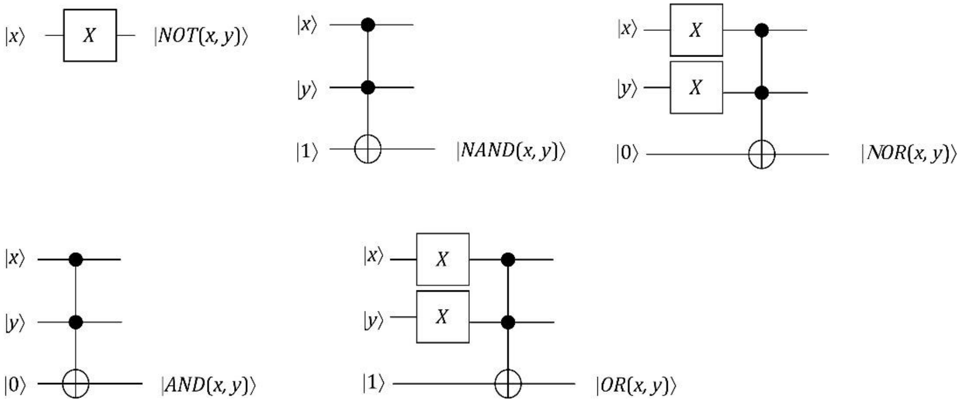

Figure 2 Logic gates

1.2.2. Connections

Remember that a quantum gate is not a component that can be connected to other components by means of a wire. It is a sequence of control signals that modify the corresponding qubit state. Connections, with fan out equal to 2 and 3, are shown in Fig.3. They are implemented with *CX* gates.

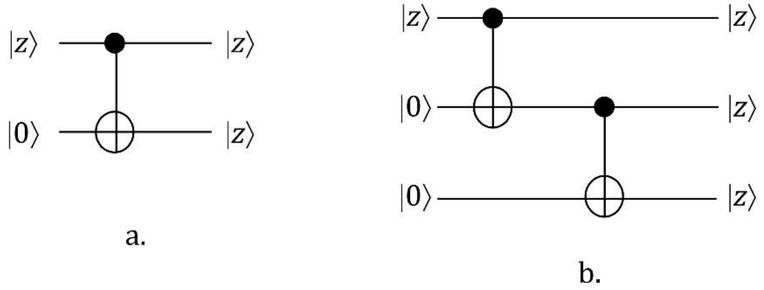

Figure 3 Connections

Example 1

The digital circuit of Fig. 4.a implements the switching function $f(z, x_0, x_1) = \bar{z}x_0 + zx_1$ (MUX2to1 gate). By replacing the inverter by an X gate, the $NAND$ gates by Toffoli gates, and the z input fan out by a CX gate, the quantum circuit of Fig.4 is obtained.

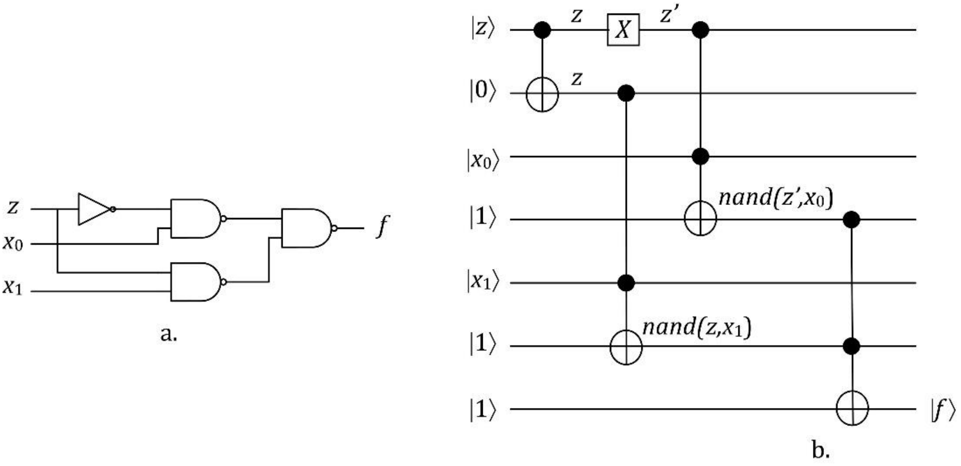

Figure 4 2 to1 multiplexer

1.3. Arithmetic circuits

Arithmetic operations are present in many information processing systems. As already commented before, the superiority of quantum circuits, with respect to digital circuits, is not in their capacity to compute switching functions. However, some quantum algorithms (Chap.7) include arithmetic operations that must be executed by the quantum circuit.

The CX and CCX quantum gates, that include the mod 2 addition, are basic building blocks of any quantum arithmetic circuit.

Example 2

Consider the digital adder of Fig.6.6.b. It is an iterative circuit composed of *n* full adders (FA) that can be implemented with logic gates (Fig.6.6.a).

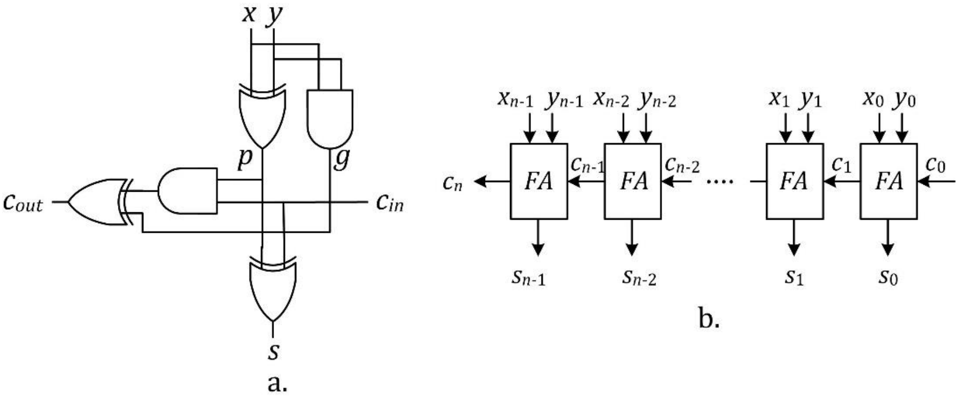

Figure 6 Digital adder

A quantum version of the adder is shown in Fig.7.

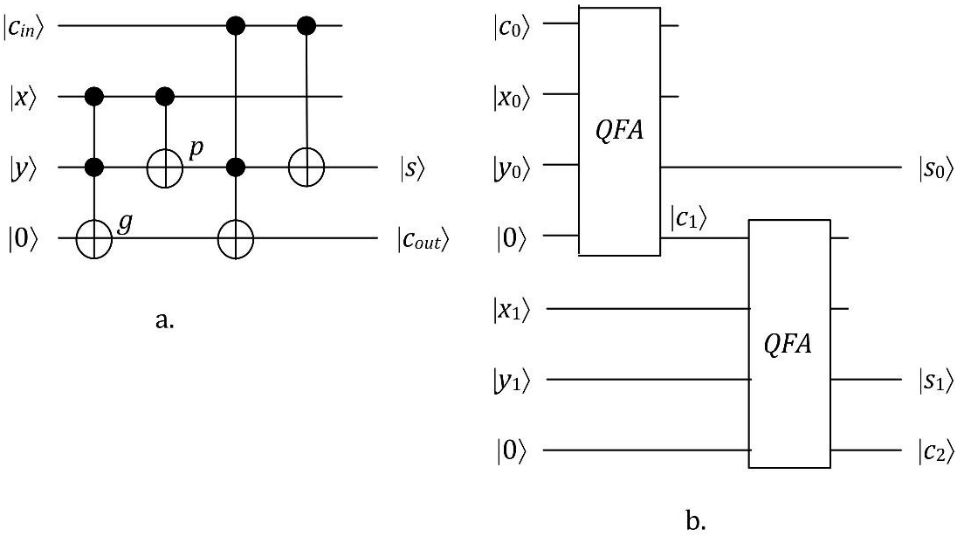

Figure 7 Quantum adder

Observe that in the quantum circuits of Fig.6 and Fig.7, if the initial qubit states are not superposition states, the next states won't be superposition states. This is because the used gates (X, CX and CCX) transform basic states into basic states. In particular, the connection circuit of Fig.3.a transforms the initial state $|z\rangle\times|0\rangle$ into $|z\rangle\times|z\rangle$ only if $|z\rangle$ is a basic state, $|0\rangle$ or $|1\rangle$. If $|z\rangle = a_0|0\rangle + a_1|1\rangle$, the initial state of the 2-qubit register is $a_0|00\rangle + a_1|10\rangle$, and the final state is $a_0|00\rangle + a_1|11\rangle$. Thus, the qubits are entangled, and the final state is not $|z\rangle\times|z\rangle = a_0^2|00\rangle + a_1^2|11\rangle + a_0a_1(|0\rangle + |1\rangle)$, unless $a_0 = 0$ or $a_1= 0$. This is a particular case of a quantum computation principle, known as the no-cloning theorem.

2. Unitary operators

As already mentioned above, the synthesis problem can be stated as follows: given a $2^n\times2^n$ unitary matrix U over the complex field – a functional specification - , define a quantum circuit consisting of N qubits ($N \geq n$) controlled by quantum gates that belong to a list of predefined components (Chap.4).

2.1. Two-level unitary operators

Two-level unitary operators are a particular class of unitary operators. Any unitary matrix can be decomposed into a product of two-level unitary matrices, so that the general synthesis problem amounts to the synthesis of this particular class of operators.

Definition 1

Consider a d-dimensional space V and a base

$$b = \{|0\rangle, |1\rangle, \dots , |d\text{-}1\rangle\}. \quad (2)$$

A linear transformation A is a two-level operator if it acts non-trivially on only two or fewer base vectors. More precisely, there are two vectors $|i\rangle$ and $|j\rangle$ of b and four coefficients c_{00}, c_{01}, c_{10} y c_{11} such that

$$A|i\rangle = c_{00}|i\rangle + c_{10}|j\rangle, A|j\rangle = c_{01}|i\rangle + c_{11}|j\rangle,$$

$$A|k\rangle = |k\rangle, \forall\ |k\rangle \neq |i\rangle \text{ and } |k\rangle \neq |j\rangle. \qquad (3)$$

A two-level operator A is unitary iff (if and only if) the submatrix defined by the coefficients c_{00}, c_{01}, c_{10} y c_{11}, that is,

$$A_{ij} = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}, \qquad (4)$$

is unitary. For that, the rows and columns of A_{ij} must be orthonormal:

$$[c_{00}\, c_{01}]\ [c_{00}\, c_{01}]^{+} = [c_{10}\, c_{11}]\ [c_{10}\, c_{11}]^{+} = 1, [c_{00}\, c_{01}]\ [c_{10}\, c_{11}]^{+} = 0,$$

$$[c_{00}\, c_{10}]\ [c_{00}\, c_{10}]^{+} = [c_{01}\, c_{11}]\ [c_{01}\, c_{11}]^{+} = 1, [c_{00}\, c_{10}]\ [c_{01}\, c_{11}]^{+} = 0. \qquad (5)$$

Property 1

If U is a $p\times p$ unitary matrix over the complex field, then U can be represented as a product of $N \leq p(p\text{-}1)/2$ two-level unitary matrices V_k.

The demonstration is based on the recurrent use of an algorithm that computes, for any $m\times m$ unitary matrix A, a set of two-level unitary matrices $A_1, A_2, \ldots, A_{m\text{-}1}$ such that

$$A_{m\text{-}1} \ldots A_2 A_1 A = \begin{bmatrix} 1\ 0 \ldots 0 & \\ 0 & \\ \ldots & Z \\ 0 & \end{bmatrix}, \qquad (6\ )$$

where Z is an $(m\text{-}1)\times(m\text{-}1)$ unitary matrix.

Algorithm 1

Assume that

$$A = \begin{bmatrix} a_{00} & a_{01} & \cdots \\ a_{10} & a_{11} & \cdots \\ & \cdots & \end{bmatrix}. \qquad (7)$$

- If $a_{10} = 0$, then $A_1 = I_m$ (the $m\times m$ identity matrix). If $a_{10} \neq 0$, define the following two-level unitary matrix

$$A_1 = \begin{bmatrix} a^* & b^* & \mathbf{0} \\ b & -a & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{m-2} \end{bmatrix}, \tag{8}$$

where

$$a = \frac{a_{00}}{\sqrt{|a_{00}|^2+|a_{10}|^2}}, b = \frac{a_{10}}{\sqrt{|a_{00}|^2+|a_{10}|^2}}, \tag{9}$$

$I_{\mathrm{m}-2}$ is the $(m\text{-}2)\times(m\text{-}2)$ identity matrix, and the other coefficients are equal to 0. Relations (9) are the kernel of the algorithm. The first two rows and columns of A_1 are orthonormal, so that it is a unitary matrix.

- Compute $B = A_1A$. It is unitary and $b_{10} = 0$. Thus

$$B = \begin{bmatrix} b_{00} & b_{01} & b_{02} & \cdots \\ 0 & b_{11} & b_{12} & \cdots \\ b_{20} & b_{21} & b_{22} & \cdots \\ & \cdots & & \end{bmatrix}. \tag{10}$$

- The next steps are similar: replace rows and columns number 0 and 1 by rows and columns number 0 and 2, 0 and 3, and so on. For example, after the second step, a matrix

$$A_2 = \begin{bmatrix} c^* & 0 & d^* & \mathbf{0} \\ 0 & 1 & 0 & \mathbf{0} \\ d & 0 & -c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{m-3} \end{bmatrix},$$

is defined. Then, compute $C = A_2B = A_2A_1A$. It is unitary and $c_{10} = c_{20} = 0$.

- After at most m-1 steps, a matrix D is obtained, such that $D = A_{m-1}\ldots A_2A_1A$, where all matrices A_k are two-level unitary matrices, and D has the structure defined by (6).

This algorithm permits to decompose a unitary matrix U into a product of two-level unitary matrices. Consider a $p\times p$ unitary matrix U. A first execution of Algorithm 1 generates p-1 two-level unitary matrices A_1, A_2, ... , A_{p-1} such that

$$A_{p-1}\ldots A_1U = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & U_{p-1} \end{bmatrix}, \tag{11}$$

where $U_{p\text{-}1}$ is a (p-1)×(p-1) unitary matrix. Then, apply Algorithm 1 to the submatrix $U_{p\text{-}1}$. It generates p-2 two-level unitary matrices B_1, B_2, ... , $B_{p\text{-}2}$ such that

$$B_{p\text{-}2}....B_1A_{p\text{-}1}....A_1U = \begin{bmatrix} \boldsymbol{I}_2 & \mathbf{0} \\ \mathbf{0} & U_{p-2} \end{bmatrix}, \tag{12}$$

where I_2 is the 2×2 identity matrix and $U_{p\text{-}2}$ is a (p-2)×(p-2) unitary matrix.

Finally, a set of two-level unitary matrices, renamed W_1, W_2, ... , $W_{N\text{-}1}$, is obtained, such that

$$W_N = W_{N\text{-}1} \dots W_2W_1U = \begin{bmatrix} \boldsymbol{I}_{p-2} & \mathbf{0} \\ \mathbf{0} & U_2 \end{bmatrix}, \tag{13}$$

where $I_{p\text{-}2}$ is the (p-2)×(p-2) identity matrix and U_2 is a 2×2 matrix. Thus, W_N is a two-level unitary matrix and U can be expressed as follows:

$$U = V_1 \dots V_{N\text{-}1}V_N, \text{ with } V_1 = W_1^+, \dots, V_{N\text{-}1} = W_{N-1}^+, V_N = W_N. \tag{14}$$

All matrices V_k are two-level unitary matrices and the maximum value of N is (p-1) + (p-2) + ... +1 = p(p-1)/2.

Example 3

Consider the unitary matrix

$$U = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & -i & 0 & -i \\ i & 0 & i & 0 \\ 0 & -i & 0 & i \\ i & 0 & -i & 0 \end{bmatrix}, \tag{15}$$

Then, using (8) and (9), compute

$$A_1 = \begin{bmatrix} 0 & -i & 0 & 0 \\ i & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_1U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -i & 0 & i \\ i & 0 & -i & 0 \end{bmatrix}. \tag{16}$$

The same operations applied to A_1U generate A_2:

$$A_2 = \begin{bmatrix} 1/\sqrt{2} & 0 & 0 & -i/\sqrt{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ i/\sqrt{2} & 0 & 0 & -1/\sqrt{2} \end{bmatrix}, A_2A_1U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{1/\sqrt{2}} & \mathbf{0} & \mathbf{1/\sqrt{2}} \\ 0 & \boldsymbol{-i/\sqrt{2}} & \mathbf{0} & \boldsymbol{i/\sqrt{2}} \\ 0 & \mathbf{0} & \boldsymbol{i} & \mathbf{0} \end{bmatrix}. \quad (17)$$

The preceding matrix has the structure defined by (6). Apply the same method to the submatrix defined by the rows and columns number 2 to 4 of A_2A_1U. The following matrix A_3 is obtained:

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{1/\sqrt{2}} & \boldsymbol{i/\sqrt{2}} & \mathbf{0} \\ 0 & \boldsymbol{-i/\sqrt{2}} & \mathbf{-1/\sqrt{2}} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, A_4 = A_3A_2A_1U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -i \\ 0 & 0 & i & 0 \end{bmatrix}. \quad (18)$$

The preceding matrix has the structure defined by (13). Thus, according to (14),

$$U = A_1^+A_2^+A_3^+A_4 = A_1A_2A_3A_4.$$

The conclusion of this section is that a unitary operator that actuates on an n-qubit register can be decomposed into a product of N two-level unitary operators, where N is smaller than $2^n(2^n - 1)/2 \cong 2^{2n-1}$.

2.2. Synthesis of two-level unitary operators

Two-level unitary operators can be synthesized with CX gates and with unary unitary operators U.

2.2.1. Synthesis with X and $C^{n-1}U$ operators

Consider a two-level operator that executes the operation (3), where the base (2) is the set of $p = 2^n$ basic states $|00...0\rangle$, ... , $|11 \dots 1\rangle$. Assume that the states $|i\rangle$ and $|j\rangle$, in (3), differ in only one bit; for example:

$$|i\rangle = |i_0\, i_1 \dots i_{m-1}\, 0\, i_{m+1} \dots i_{n-1}\rangle,\ |j\rangle = |i_0\, i_1 \dots i_{m-1}\, 1\, i_{m+1} \dots i_{n-1}\rangle. \quad (19)$$

Then, A executes the following transformations:

- if $|k\rangle \neq |i_0\, i_1 \dots i_{m-1}\, 0\, i_{m+1} \dots i_{n-1}\rangle$ and $|k\rangle \neq |i_0\, i_1 \dots i_{m-1}\, 1\, i_{m+1} \dots i_{n-1}\rangle$, then $A|k\rangle = |k\rangle$;

- if $|k\rangle = |i_0\, i_1 \dots\, i_{m\text{-}1}\; 0\; i_{m+1} \dots i_{n\text{-}1}\rangle$, then $A|k\rangle = |i_0\, i_1 \dots\, i_{m\text{-}1}\rangle \times (c_{00}|0\rangle + c_{10}|1\rangle) \times |i_{m+1} \dots i_{n\text{-}1}\rangle$;
- if $|k\rangle = |i_0\, i_1 \dots\, i_{m\text{-}1}\; 1\; i_{m+1} \dots i_{n\text{-}1}\rangle$, then $A|k\rangle = |i_0\, i_1 \dots\, i_{m\text{-}1}\rangle \times (c_{01}|0\rangle + c_{11}|1\rangle) \times |i_{m+1} \dots i_{n\text{-}1}\rangle$.

The corresponding circuit is shown in Fig.8. The operator $X(p)$ is the identity operator I_1 if $i_p = 1$ and the operator X if $i_p = 0$. This set of X operators is similar to an address decoder that detects the state $|i_0\, i_1 \dots\, i_{m\text{-}1}\rangle \times |i_{m+1} \dots i_{n\text{-}1}\rangle$ and, if detected, sets the qubits number 0 to m-1 and m+1 to n-1 in state $|1\rangle$. The unitary operator U acts on the qubit number m under the control of the other qubits. It is a generalization of the C^2U operator (Sec.3.3.1) with n-1 control qubits instead of 2. The operation U is executed iff the n-1 control qubits are in state $|1\rangle$. It executes the following transformations of qubit number m basic states:

$$|0\rangle \overset{U}{\rightarrow} (c_{00}|0\rangle + c_{10}|1\rangle),\ |1\rangle \overset{U}{\rightarrow} (c_{01}|0\rangle + c_{11}|1\rangle). \qquad (20)$$

The $X(p)$ operators, on the right side of Fig.8, reset all the qubits, but the target qubit, to their initial state.

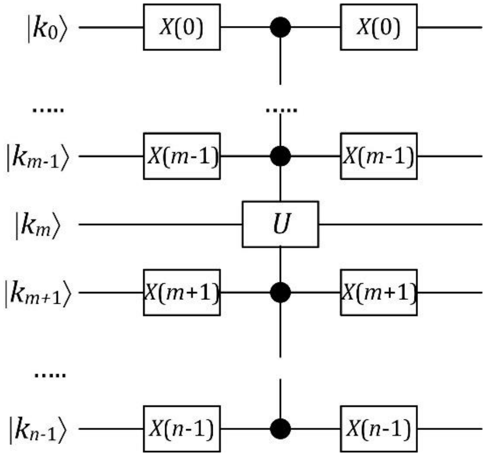

Figure 8 Two-level operator synthesis

If the vicinity condition (19) doesn't hold, additional permutation operators must be used. Consider a permutation P that executes the following register state transformation:

- $|i_0\ i_1 \ldots\ i_{m\text{-}1}\ x\ i_{m+1} \ldots i_{n\text{-}1}\rangle \xrightarrow{P} |i_0\ i_1 \ldots\ i_{m\text{-}1}\ \bar{x}\ i_{m+1} \ldots i_{n\text{-}1}\rangle$, $\forall x \in \{0,1\}$,
- if $|k\rangle \neq |i_0\ i_1 \ldots\ i_{m\text{-}1}\ 0\ i_{m+1} \ldots i_{n\text{-}1}\rangle$ and $|k\rangle \neq |i_0\ i_1 \ldots\ i_{m\text{-}1}\ 1\ i_{m+1} \ldots i_{n\text{-}1}\rangle$, then $P|k\rangle = |k\rangle$. (21)

It transforms the state $|i_0\ i_1 \ldots\ i_{m\text{-}1}\ 0\ i_{m+1} \ldots i_{n\text{-}1}\rangle$ into the state $|i_0\ i_1 \ldots\ i_{m\text{-}1}\ 1\ i_{m+1} \ldots i_{n\text{-}1}\rangle$, and inversely, without modifying the other states. It can be executed by the circuit of Fig.8, with $U = X$. If the states $|i\rangle$ and $|j\rangle$, in (3), differ in more than one bit, a series of permutations (21) permits to interchange a basic state with another basic state that is at distance 1 from a given basic state, without modifying the others. For example (n = 3), if $|i\rangle = |001\rangle$ and $|j\rangle = |110\rangle$, then define two permutations P_1 and P$_0$:

$$|001\rangle \xrightarrow{P_1} |011\rangle, |011\rangle \xrightarrow{P_1} |001\rangle, |k\rangle \xrightarrow{P_1} |k\rangle \text{ si } |k\rangle \neq |001\rangle \text{ y } |k\rangle \neq |011\rangle;$$

$$|011\rangle \xrightarrow{P_0} |111\rangle, |111\rangle \xrightarrow{P_0} |011\rangle, |k\rangle \xrightarrow{P_0} |k\rangle \text{ si } |k\rangle \neq |011\rangle \text{ y } |k\rangle \neq |111\rangle. (22)$$

By successively applying P_1 and P_0, the following transformation is executed:

$$|001\rangle \xrightarrow{P_1} |011\rangle \xrightarrow{P_0} |111\rangle,$$

so that $|i\rangle = |001\rangle$ is replaced by a state $|111\rangle$which is at distance 1 from $|j\rangle = |110\rangle$.

To summarize, the synthesis of a two-level unitary operator A, defined by (3), is performed as follows:

- Execute a permutation that transforms the basic state $|i\rangle$ into a basic state at distance 1 from $|j\rangle$. For that, a series of permutations (21), implemented with gates X and $C^{n\text{-}1}X$, are executed.
- Execute the operation $C^{n\text{-}1}U$ (Fig.8).
- Execute the same permutations as before, but in reverse order.

Example 4

The following two-level matrix

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{a} & 0 & 0 & 0 & 0 & \boldsymbol{b} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \boldsymbol{c} & 0 & 0 & 0 & 0 & \boldsymbol{d} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{23}$$

with $|i\rangle = |001\rangle$ and $|j\rangle = |110\rangle$, is unitary if the submatrix U, defined by a, b, c and d, is unitary. Then, using the permutations (22), interchange $|i\rangle = |001\rangle$ with $|i'\rangle = |111\rangle$ at distance 1 from $|j\rangle = |110\rangle$. Execute C^2U (Fig.8 with $n = 3$, $m = 2$) where U executes the following transformation:

$$|0\rangle \overset{U}{\rightarrow} d|0\rangle + b|1\rangle, |1\rangle \overset{U}{\rightarrow} c|0\rangle + a|1\rangle.$$

Finally, execute the permutations (22) in reverse order. The circuit is shown in Fig.9, assuming that $U = H$.

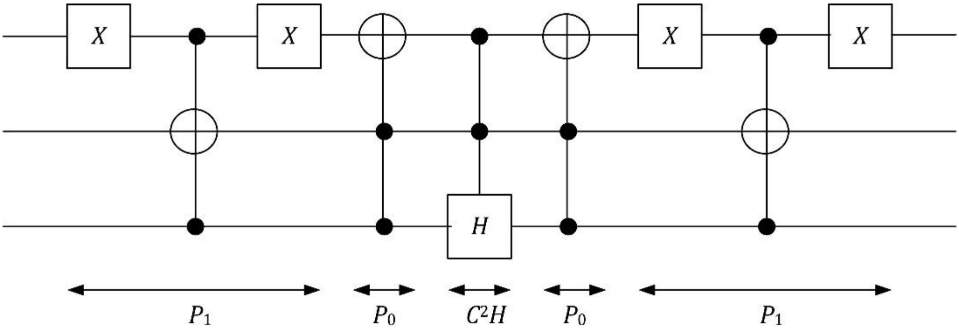

Figure 9 Operator (23) with $a = -1/\sqrt{2}$, $b = c = d = 1/\sqrt{2}$

2.2.2. Synthesis of $C^{n-1}U$ operators

An operator $C^{n-1}U$, that is, a unary unitary operator U controlled by n-1 qubits, can be synthesized with CU and C^2X (Toffoli) operators. A circuit that implements a C^4U operator is shown in Fig.10. This structure can obviously be generalized to any number n-1 of control qubits. Apart from the target qubit t and the four control qubits c_0 to c_3, the circuit includes three ancillary qubits a_0, a_1 and a_2. Observe that the qubit a_2 is in state $|1\rangle$ iff the control qubits are in state $|1\rangle$. Thus, as a_2 controls the operation U, the circuit executes the operation C^4U. After executing the main operation, the ancillary qubits are reset to their initial state. This is a common practice. Resetting the ancillary qubits permits to use them later, to execute other operations. Furthermore, the reduction of the whole system energy makes it more stable and less sensitive to external perturbations.

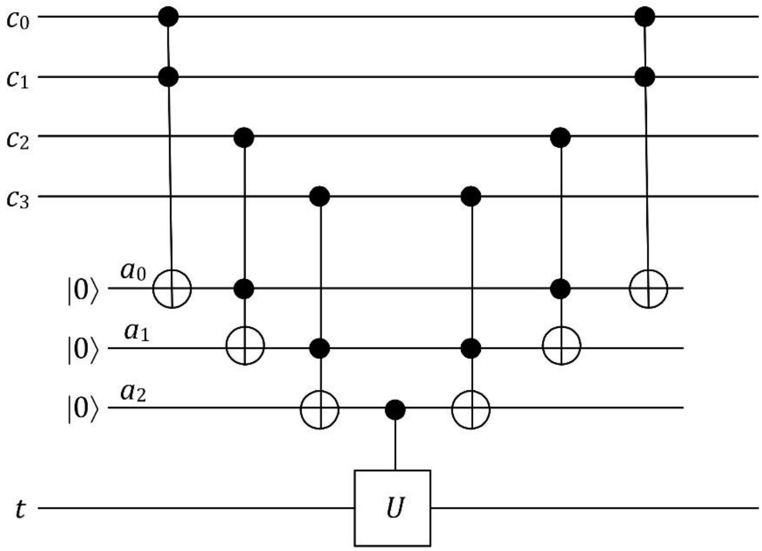

Figure 10 Operator C^4U

2.2.3. Synthesis of *U* and *CU* operators

The implementation of unitary operators that act on a single qubit is certainly an important research topic in quantum computation. In this section it is demonstrated that any unary unitary operator U can be synthesized with rotation operators, and that, with additional CX gates, the corresponding CU operator can be implemented.

Theorem 1

If U is a unary unitary operator, there exist real numbers a, b, c and d (angles in radians) such that

$$U = e^{ia}\, R_z(b)\, R_y(c)\, R_z(d). \tag{24}$$

Proof

Assume that

$$U = \begin{bmatrix} u_{00}e^{i\alpha} & u_{01}e^{i\beta} \\ u_{10}e^{i\gamma} & u_{11}e^{i\delta} \end{bmatrix} \tag{25}$$

where all u_{jk} are non-negative real numbers. As U is unitary, $U^{+}U = I$, and

$$u_{00}^2 + u_{01}^2 = u_{10}^2 + u_{11}^2 = u_{00}^2 + u_{10}^2 = u_{01}^2 + u_{11}^2 = 1.$$

Thus, $u_{01} = u_{10}$, $u_{00} = u_{11}$, and there exists an angle c, between 0 and π radians, such that

$$u_{00} = u_{11} = \cos(c/2) \text{ y } u_{01} = u_{10} = \sin(c/2). \tag{26}$$

Define

$$a = \frac{\alpha+\delta}{2},\ b = \gamma - \alpha,\ d = \delta - \gamma, \tag{27}$$

and c such that

$$u_{00} = u_{11} = \cos(c/2) \text{ y } u_{01} = u_{10} = \sin(c/2). \tag{28}$$

Furthermore, by orthogonality of rows and columns,

$$u_{00}e^{i\alpha}\, u_{10}e^{-i\gamma} + u_{01}e^{i\beta}\, u_{10}e^{-i\delta} = 0,$$

so that

$$\cos(c/2)\sin(c/2)(e^{i(\alpha-\gamma)} + e^{i(\beta-\delta)}) = 0, \tag{29}$$

and, excepted if $c = 0$ or $c = \pi$,

$$\beta = \pi + \alpha - \gamma + \delta = \pi + a - b/2 + d/2 \ (\mathrm{mod}\ 2\pi). \tag{30}$$

Thus (Chap.4, relations 36), U can be represented as

$$U = \begin{bmatrix} e^{i(a-\frac{b}{2}-\frac{d}{2})}\, cos\frac{c}{2} & -e^{i(a-\frac{b}{2}+\frac{d}{2})}\, sin\frac{c}{2} \\ e^{i(a+\frac{b}{2}-\frac{d}{2})}\, sin\frac{c}{2} & e^{i(a+\frac{b}{2}+\frac{d}{2})}\, cos\frac{c}{2} \end{bmatrix} = e^{ia} R_z(b)\, R_y(c)\, R_z(d). \quad (31)$$

If $c = 0$, then

$$U = \begin{bmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\delta} \end{bmatrix} = e^{i\frac{\alpha+\delta}{2}} R_z(\delta\text{-}\alpha),$$

and if $c = \pi$,

$$U = \begin{bmatrix} 0 & e^{i\beta} \\ e^{i\gamma} & 0 \end{bmatrix} = ie^{i\frac{\beta+\gamma}{2}} R_z(\gamma\text{-}\beta)\, R_y(\pi)\, R_z(\pi).$$

Thus, any unary unitary operator can be synthesized with rotations about the axis $0y$ and $0z$.

The next corollary indicates how CU operators can be synthesized.

Corollary 1

If U is a unary unitary operator, then there exist three unary unitary operators A, B and C and a real number a such that

$$ABC = I,\ U = e^{ia}AXBXC. \quad (32)$$

Proof

Define

$$A = R_z(b)R_y(c/2),\ B = R_y(\text{-}c/2)R_z(\text{-}(d+b)/2),$$
$$C = R_z((d\text{-}b)/2), \quad (33)$$

and check that both relations (32) hold true.

Using the preceding corollary, a CU operator can be implemented by the circuit of Fig.11. Assume that the initial state of the target qubit t is $|\psi\rangle$. The circuit executes the following operations:

$$|0\rangle\times|\psi\rangle \rightarrow R_\alpha|0\rangle\times ABC|\psi\rangle = |0\rangle\times I|\psi\rangle = |0\rangle\times|\psi\rangle,$$

$$|1\rangle\times|\psi\rangle \rightarrow R_\alpha|1\rangle\times AXBXC|\psi\rangle = e^{i\alpha}|1\rangle\times AXBXC\,|\psi\rangle = |1\rangle\times U|\psi\rangle.$$

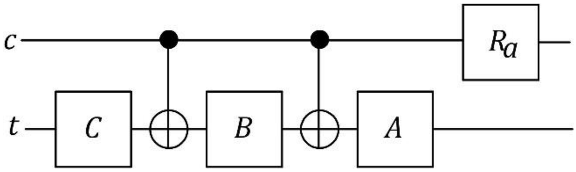

Figure 11 Synthesis of *CU*

In conclusion, any unitary operator, acting on an n-qubit register, can be synthesized with operators X, CX, C^2X (Toffoli) and rotations about the axis $0y$ and $0z$.

2.2.4. Synthesis of ternary operators

The circuit of Fig.12 executes the operation C^2U where U is equal to $V^2 = V{\cdot}V$ and V is unitary. Assume that, initially, q_2 is in state $|\psi\rangle$. Then, in function of the states of q_0 and q_1, the following transformations are executed:

- $|00\rangle\times|\psi\rangle \xrightarrow{CV} |00\rangle\times|\psi\rangle \xrightarrow{CX} |00\rangle\times|\psi\rangle \xrightarrow{CV^+} |00\rangle\times|\psi\rangle \xrightarrow{CX,} |00\rangle\times|\psi\rangle$ $\xrightarrow{CV} |00\rangle\times|\psi\rangle$,
- $|01\rangle\times|\psi\rangle \xrightarrow{CV} |01\rangle\times V|\psi\rangle \xrightarrow{CX} |01\rangle\times V|\psi\rangle \xrightarrow{CV^+} |01\rangle\times V^+V|\psi\rangle \xrightarrow{CX,} |01\rangle\times V^+V|\psi\rangle$ $\xrightarrow{CV} |01\rangle\times V^+V|\psi\rangle = |01\rangle\times|\psi\rangle$,
- $|10\rangle\times|\psi\rangle \xrightarrow{CV} |10\rangle\times|\psi\rangle \xrightarrow{CX} |11\rangle\times|\psi\rangle \xrightarrow{CV^+} |11\rangle\times V^+|\psi\rangle \xrightarrow{CX,} |10\rangle\times V^+|\psi\rangle$ $\xrightarrow{CV} |10\rangle\times VV^+|\psi\rangle = |10\rangle\times|\psi\rangle$,
- $|11\rangle\times|\psi\rangle \xrightarrow{CV} |11\rangle\times V|\psi\rangle \xrightarrow{CX} |10\rangle\times V|\psi\rangle \xrightarrow{CV^+} |10\rangle\times V|\psi\rangle \xrightarrow{CX,} |11\rangle\times V|\psi\rangle$

$\xrightarrow{CV} |11\rangle\times VV|\psi\rangle = V^2|\psi\rangle$.

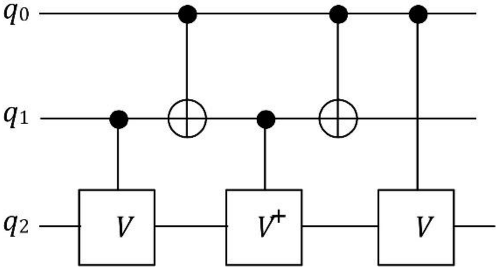

Figure 12 Operator C^2U with $U = V^2$

Given a unitary operator U, the computation of V, such that $V^2 = U$, amounts to the computation of the square root of the eigenvalues of U. For example, assume that $U = X$. The eigenvalues of X are 1 and -1 and its spectral representation is

$$X = 1\cdot\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + (-1)\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Thus,

$$V = X^{1/2} = 1\cdot\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + i\begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}.$$

Using (24), the following decomposition of V is obtained:

$$V = e^{\frac{i\pi}{4}} R_z(-\tfrac{\pi}{2})\ R_y(\tfrac{\pi}{2})\ R_z(\tfrac{\pi}{2}).$$

To synthesize the controlled operator CV use the circuit of Fig.11 with, according to (33),

$$A = R_z(-\tfrac{\pi}{2})R_y(\tfrac{\pi}{4}),\ B = R_y(-\tfrac{\pi}{4})R_z(0),\ C = R_z(\tfrac{\pi}{2}).$$

Thus, the circuit of Fig.13 executes the CV operator.

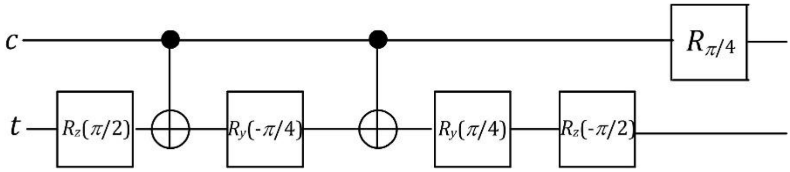

Figure 13 Operator CV with $V^2 = X$

In conclusion, according to Fig.12 and Fig.13, a C^2X (Toffoli) gate can be synthesized with unary operators (rotations) and CX gates.

In fact, more specific and efficient implementations of the Toffoli operator have been proposed. An example is shown in Fig.14. This circuit includes unary operators H, $S = R_{\pi/2}$ and $T = R_{\pi/4}$, and CX operators.

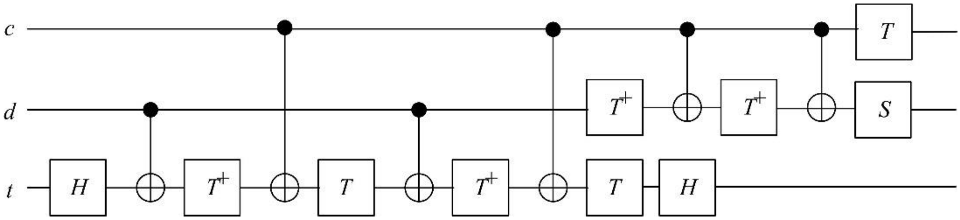

Figure 14 Toffoli operator

2.3. Final comments

- The conclusion of the preceding sections is that any unitary operator U that acts on an n-qubit register can be decomposed into a product of unary rotation operators about (e.g.) the axis $0y$ and $0z$, and CX operators.
- The synthesis method proposed in Sec.2.1 is based on the decomposition of U into two-level unitary operators. Another method uses the CS (cosine – sine) decomposition [2]. The conclusions are similar. The complexity of the so-obtained circuit, in terms of quantum gates, is $O(4^n)$ [3]. Observe that this value is compatible with the property 1 ($p = 2^n$). More details are given in [4].
- Apart from the axis $0x$, $0y$ and $0z$, other coordinate axis can be defined. Let n_x, n_y and n_z be the coordinates of a point n on the Bloch sphere.

Then, rotations $R_n(\gamma)$ about the axis $0n$ can be defined. The corresponding matrix is

$$R_n(\gamma) = \begin{bmatrix} \cos\frac{\gamma}{2} - in_z\sin\frac{\gamma}{2} & -(n_y + in_x)\sin\frac{\gamma}{2} \\ (n_y - in_x)\sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} + in_z\sin\frac{\gamma}{2} \end{bmatrix}.$$

- Consider another point m on the Bloch sphere, whose coordinates are n_x, $-n_y$ and n_z. It can be demonstrated that $R_m(\gamma) = HR_n(\gamma)H$. The property 1 can be generalized, replacing the axis $0x$ and $0y$ by $0n$ and $0m$. Thus, any unary unitary matrix can be decomposed into rotations about the axis $0n$ and $0m$. Furthermore, it can also be demonstrated that any rotation $R_n(\gamma)$, about the axis $0n$, is approximately equal to $(THTH)^A$, where A is a natural whose value depends on γ and on the desired accuracy. Thus, taking into account the generalized property 1, the expression of $R_m(\gamma)$ in function of $R_n(\gamma)$, and the approximation of $R_n(\gamma)$ with T and H gates, the conclusion is that any unary unitary operation U can be synthesized with T and H gates.
- Taking into account the first and the latter comment, any unitary operator that acts on an n-qubit register can be decomposed into H and T unary gates, and CX gates. As an example, observe that the Toffoli operator of Fig.14 is made up of T, $S = T^2$, H and CX gates. Intuitively, this means that any unitary operations can be implemented with rotations (T gates), superpositions (H gates) and entanglements (CX gates).
- The physical implementation of CX gates, applied to non-adjacent qubits, could be impossible. In this case, more complex circuits must be considered. An example is given in Fig.15: the CX gate of Fig.15.a, in which the control qubit is separated from the target qubit by other two qubits, can be implemented as shown in Fig.15.b.

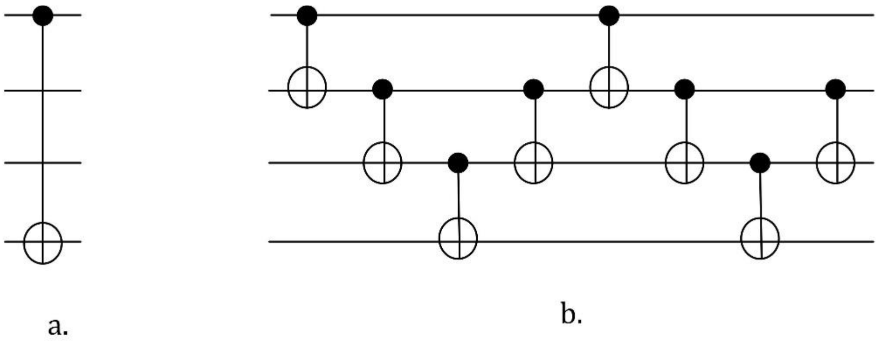

Figure 15 Equivalent circuits

References

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

[2] Sutton, B.D. Computing the complete CS decomposition. Numer. Algor. 2009 50, 33–65.

[3] Shende, V.; Bullock, S.; Markov, I. Synthesis of Quantum Logic Circuits. Comput.-Aided Des. Integr. Circuits Syst. IEEE Trans. 2006, 25, 1000–1010

[4] https://github.com/Marcombo/Circuitos-y-algoritmos-cuanticos, Nota 7: Descomposición *CS* de matrices, 2023.