Ejercicios

41, 49, 50, 51, 52, 53, 57, 60, 61

Autoría:
Aparicio Morales, Álvaro Manuel
Módulo 3
I CERTIFICADO DE EXTENSIÓN UNIVERSITARIA EN COMPUTACIÓN CUÁNTICA (2024-25)
UNIVERSIDAD DE MÁLAGA

Ejercicio 41

$$H = U_1^{(c)}(\beta = \pi, \phi = \pi)\, U_1^{(c)}(\beta = \pi/2, \phi = -\pi/2)$$

Tomando como base $\{|g,0\rangle, |e,0\rangle\}$ podemos expresar la matriz de carrier transition como una matriz 2x2

$$U^{(c)} = \begin{pmatrix} \cos(\beta/2) & -ie^{-i\phi}\sin(\beta/2) \\ -ie^{i\phi}\sin(\beta/2) & \cos(\beta/2) \end{pmatrix}$$

Con esta matriz vamos a calcular primeramente $U_1^{(c)}(\beta=\pi, \phi=\pi)$

$$U_1^{(c)}(\beta=\pi, \phi=\pi) = \begin{pmatrix} \cos(\pi/2) & -ie^{-i\pi}\sin(\pi/2) \\ -ie^{-i\pi}\sin(\pi/2) & \cos(\pi/2) \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & -ie^{i\pi} \\ -ie^{i\pi} & 0 \end{pmatrix} \rightarrow \text{Aplicamos Euler} = \begin{pmatrix} 0 & -i(-1) \\ -i(-1) & 0 \end{pmatrix} =$$

$$e^{i\phi} = \cos(\phi) + i\sin(\phi)$$

$$e^{-i\pi} = \cos(\pi) - i\sin(\pi)$$

$$e^{i\pi} = \cos(\pi) + i\sin(\pi)$$

$$= \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}$$

Ahora vamos a calcular $U_1^{(c)}(\beta - \pi/2$ y $\phi = -\pi/2)$

$$U_1^{(c)}(\beta=\pi/2, \phi=-\pi/2) = \begin{pmatrix} \cos(\pi/4) & -ie^{i\pi/2}\sin(\pi/4) \\ -ie^{-i\pi/2}\sin(\pi/4) & \cos(\pi/4) \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{\sqrt{2}}{2} & -ie^{i\pi/2}\frac{\sqrt{2}}{2} \\ -ie^{-i\pi/2}\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} \rightarrow \begin{matrix}\text{Aplicando Euler} \\ \text{y simplificando}\end{matrix} = \frac{\sqrt{2}}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Teniendo ambas matrices, podemos hacer la comprobación

$$H = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} \cdot \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} =$$

$$= \frac{\sqrt{2}}{2} \begin{pmatrix} -i & i \\ i & i \end{pmatrix} = \frac{i\sqrt{2}}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}$$ (No coincide con la Hadamard)

Vamos a probar con $\phi = \pi/2$ en la segunda matriz. Por tanto:

$$U_1^{(c)}(\beta = \pi/2, \phi = \pi/2) = \begin{pmatrix} \cos(\pi/2) & -ie^{-i\pi/2} \operatorname{sen}(\pi/4) \\ -ie^{i\pi/2} \operatorname{sen}(\pi/4) & \cos(\pi/2) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\sqrt{2}}{2} & -ie^{-i\pi/2} \frac{\sqrt{2}}{2} \\ -ie^{i\pi/2} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix} = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & -ie^{-i\pi/2} \\ -e^{i\pi/2} & 1 \end{pmatrix} \rightarrow$$

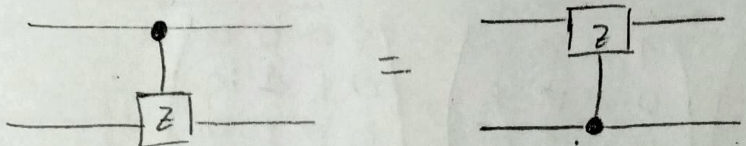
Aplicando Euler y simplificando $\rightarrow \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$

$$H = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \frac{\sqrt{2}}{2} = \frac{\sqrt{2}}{2} \begin{pmatrix} i & i \\ i & -i \end{pmatrix} =$$

$$= \frac{i\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = i \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Demostramos que $U_1^{(c)}(\beta = \pi, \phi = \pi)\, U_1^{(c)}(\beta = \pi/2, \phi = \pi/2) = H$ con una fase global

49) Circuito 14. 3. 3. 1

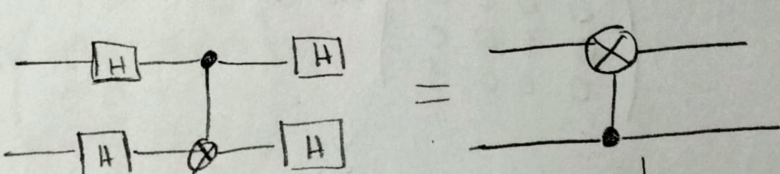


$$U_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Podemos observar que ambas matrices son idénticas. Si dos operadores cuánticos son idénticos, esto implica que su producto es igual a la matriz identidad

$$U_1 \cdot U_2 = I \rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = I$$

Circuito 14. 3. 3. 2



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Vamos a calcular la matriz de operación del circuito:

$$H \otimes H \cdot CX \cdot H \otimes H$$

$$H \otimes H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

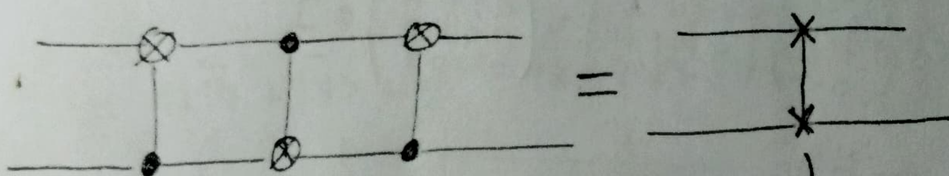
$$\frac{1}{2} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$= \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$U_1 = \begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix}$$

$$U_1 \cdot U_2 = I \rightarrow \begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix}\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix} = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{pmatrix}$$

14.3.3 3



$$\downarrow$$

$$\begin{pmatrix} 1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1 \end{pmatrix} \rightarrow U_2$$

$$CNot_{0,1} = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix}$$

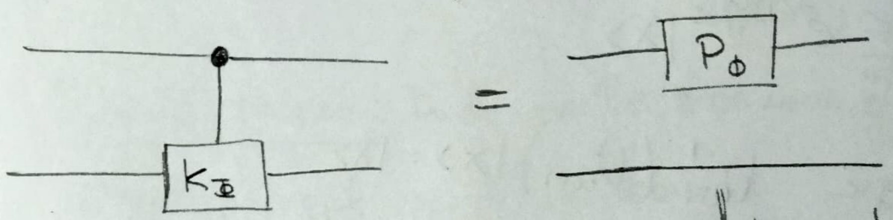
$$CNot_{1,0} = \begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix}$$

$$\overbrace{CX_{1,0} \circ CX_{01} \circ CX_{10}}^{U_1} = \begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix}\begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix}\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix} =$$

$$= \begin{pmatrix} 1&0&0&0\\0&0&1&0\\0&0&0&1\\0&1&0&0 \end{pmatrix}\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix} = \begin{pmatrix} 1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1 \end{pmatrix}$$

$$U_1 \cdot U_2 = \begin{pmatrix} 1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1 \end{pmatrix}\begin{pmatrix} 1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1 \end{pmatrix} = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{pmatrix}$$

14.3.3.4



La matriz operación asociada al sistema

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\Phi} & 0 \\ 0 & 0 & 0 & e^{i\Phi} \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\Phi} & 0 \\ 0 & 0 & 0 & e^{i\Phi} \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\Phi} & 0 \\ 0 & 0 & 0 & e^{i\Phi} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\Phi} & 0 \\ 0 & 0 & 0 & e^{i\Phi} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{2i\Phi} & 0 \\ 0 & 0 & 0 & e^{2i\Phi} \end{pmatrix} \rightarrow$$ Aplicando Euler y la identidad trigonométrica

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Bloque 3: Algoritmos Cuánticos

Alumno: Álvaro Manuel Aparicio Morales

I Certificado de Extensión Universitaria en Computación Cuántica (2024-2025)

Ejercicio 49

Para la resolución de este ejercicio, se ha desarrollado una función para comprobar que dos operadores unitarios son equivalentes mediante el producto de un operador del circuito 1 con el traspuesto conjugado del circuito 2 y la comprobación de que esa multiplicación es igual a la identidad.

```
from qiskit import QuantumCircuit, quantum_info
import numpy as np
```

```
def operators_equivalence(qc_1,qc_2):
    unitary_operator_qc1 = quantum_info.Operator(qc_1)
    unitary_operator_qc2 = quantum_info.Operator(qc_2)

    # Para comprobar si ambos operadores son idénticos el producto del operador del circuito 1
    # con el traspueto conjugado del circuito 2, debe dar la identidad

    unitary_operator_matrix_qc1 = unitary_operator_qc1.data
    unitary_operator_matrix_qc2 = unitary_operator_qc2.data

    print(unitary_operator_matrix_qc1)
    print(unitary_operator_matrix_qc2)

    identity = np.identity(unitary_operator_matrix_qc1.shape[0]) # Obtenemos el tamaño de la matriz para generar la identidad
    operators_products = unitary_operator_matrix_qc1 @ (np.conjugate(unitary_operator_matrix_qc2.T))
    equivalence = np.allclose(identity, operators_products,atol=1e-9)
    return equivalence
```

```
# Circuito 14.3.3.1
qc1_1 = QuantumCircuit(2)
qc1_1.cz(0,1)
qc1_2 = QuantumCircuit(2)
qc1_2.cz(1,0)
equivalence = operators_equivalence(qc_1=qc1_1, qc_2=qc1_2)
print("Result of equivalence is: ", equivalence)
```

```
[[ 1.+0.j  0.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  1.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  0.+0.j -1.+0.j]]
[[ 1.+0.j  0.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  1.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  0.+0.j -1.+0.j]]
Result of equivalence is:  True
```

```
# Circuito 14.3.3.2
qc2_1 = QuantumCircuit(2)
qc2_1.h(0)
qc2_1.h(1)
qc2_1.cx(0,1) # cnot
qc2_1.h(0)
qc2_1.h(1)

qc2_2 = QuantumCircuit(2)
qc2_2.cx(1,0) # cnot()
equivalence = operators_equivalence(qc_1=qc2_1, qc_2=qc2_2)
print("Result of equivalence is: ", equivalence)
```

```
[[1.+0.j 0.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 0.+0.j 1.+0.j]
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]]
[[1.+0.j 0.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 0.+0.j 1.+0.j]
```

```
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]]
Result of equivalence is:  True
```

```
# Circuito 14.3.3.3
qc3_1 = QuantumCircuit(2)
qc3_1.cx(1,0)
qc3_1.cx(0,1)
qc3_1.cx(1,0)
qc3_2 = QuantumCircuit(2)
qc3_2.swap(0,1)
equivalence = operators_equivalence(qc_1=qc3_1, qc_2=qc3_2)
print("Result of equivalence is: ", equivalence)
```

```
[[1.+0.j 0.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 0.+0.j 1.+0.j]]
[[1.+0.j 0.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 0.+0.j 1.+0.j]]
Result of equivalence is:  True
```

```
# Circuito 14.3.3.4
from qiskit.circuit.library import U1Gate

def get_ckmatrix (phi):
    ckmatrix = np.array([
        [1, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, np.exp(1j * phi), 0],
        [0, 0, 0, np.exp(1j * phi)]
    ])
    return ckmatrix

phase = np.pi

qc4_1 = QuantumCircuit(2)
ck_operator = quantum_info.Operator(get_ckmatrix(phase)) # Construimos el operador CKphi para cualquier fase
qc4_1.append(ck_operator,[0,1])

qc4_2 = QuantumCircuit(2)
qc4_2.append(U1Gate(phase),[1])

equivalence = operators_equivalence(qc_1=qc4_1, qc_2=qc4_2)
print("Result of equivalence is: ", equivalence)
```

```
[[ 1.+0.0000000e+00j  0.+0.0000000e+00j  0.+0.0000000e+00j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  1.+0.0000000e+00j  0.+0.0000000e+00j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  0.+0.0000000e+00j -1.+1.2246468e-16j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  0.+0.0000000e+00j  0.+0.0000000e+00j
  -1.+1.2246468e-16j]]
[[ 1.+0.0000000e+00j  0.+0.0000000e+00j  0.+0.0000000e+00j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  1.+0.0000000e+00j  0.+0.0000000e+00j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  0.+0.0000000e+00j -1.+1.2246468e-16j
   0.+0.0000000e+00j]
 [ 0.+0.0000000e+00j  0.+0.0000000e+00j  0.+0.0000000e+00j
  -1.+1.2246468e-16j]]
Result of equivalence is:  True
```

Ejercicio 50

Para demostrar la equivalencia entre estos dos circuitos vamos a ver que le ocurre al tercer qubit con los diferentes posibles valores de entrada

Comenzamos con el circuito de puerta multicontrolada U

Entrada	Tercer qubit ($\|\psi\rangle$)
$\|00\rangle$	$\|\psi\rangle$
$\|01\rangle$	$\|\psi\rangle$
$\|10\rangle$	$\|\psi\rangle$
$\|11\rangle$	$U\|\psi\rangle$

Ahora para el segundo circuito

Entrada	Tercer qubit ($\|\psi\rangle$)
$\|00\rangle$	$\|\psi\rangle$
$\|01\rangle$	$V^{+}V\|\psi\rangle = I\|\psi\rangle$
$\|10\rangle$	$VV^{+}\|\psi\rangle = I\|\psi\rangle$
$\|11\rangle$	$VV\|\psi\rangle = V^{2}\|\psi\rangle$

Por tanto, si se verifica que $U = V^{2}$, entonces ambos circuitos son equivalentes.

51) Vamos a ir desglosando las transiciones de estados que se producen en el circuito

Estado inicial: Dado que los 2 primeros qubits son $|0\rangle$. El estado inicial se define como:

$$|0\rangle \otimes |0\rangle \otimes |\psi\rangle$$

Aplicamos H:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|0\rangle \otimes H|0\rangle \otimes |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |\psi\rangle$$

$$= \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes |\psi\rangle =$$

$$= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \otimes |\psi\rangle$$

Aplicamos CCNot (Toffoli)

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \otimes |\psi\rangle =$$

$$= \frac{1}{2}(|00\rangle \otimes |\psi\rangle + |01\rangle \otimes |\psi\rangle + |10\rangle \otimes |\psi\rangle + |11\rangle \otimes |\psi\rangle)$$

→ Aplicación CNot

$$\frac{1}{2}(|00\rangle \otimes |\psi\rangle + |01\rangle \otimes |\psi\rangle + |10\rangle \otimes |\psi\rangle + |11\rangle \otimes T|\psi\rangle)$$

Aplicamos S

$$\frac{1}{2}((|00\rangle + |01\rangle + |10\rangle) \otimes S|\psi\rangle + |11\rangle \otimes S(T|\psi\rangle)) -$$

Aplicamos CCNOT (toffoli)

$$\frac{1}{2}\left[(|00\rangle + |01\rangle + |10\rangle) \otimes S|\psi\rangle + |11\rangle \otimes T(S(T|\psi\rangle))\right]$$

Aplicamos Hadamard

$$\frac{1}{2}\left[(|00\rangle + |01\rangle + |10\rangle)\otimes S|\psi\rangle + |11\rangle \otimes T(S(T|\psi\rangle))\right]$$

$$\Big\downarrow \text{Hadamard}$$

$$\frac{1}{2}\left[(H\otimes H|00\rangle + H\otimes H|01\rangle + H\otimes H|10\rangle)\otimes S|\psi\rangle + H\otimes H|11\rangle \otimes T(S(T|\psi\rangle))\right]$$

$$H\otimes H|00\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$H\otimes H|01\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

$$H\otimes H|10\rangle = \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

$$\frac{1}{2}(3|00\rangle + |01 + |10\rangle - |11\rangle)$$

$$H\otimes H|11\rangle = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

Agrupamos los resultados

$$\frac{1}{2}\left[\frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)\otimes S|\psi\rangle + \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)\right.$$

$$\left.\otimes T(S(T|\psi\rangle))\right] =$$

$$= \frac{1}{4}\left[(|00\rangle - |01\rangle - |10\rangle + |11\rangle)\otimes S|\psi\rangle + (|00\rangle - |01 - |10\rangle + |11\rangle)\cdot\right.$$

$$\left.\otimes T(S(T|\psi\rangle))\right]$$

Si al medir, los 2 primeros qubits valen 0 el estado es:

$$\frac{1}{4}(S|\psi\rangle \otimes T(S(T|\psi\rangle))$$

$q_0 = 0 \quad q_1 = 0$ Debemos demostrar que

$$\frac{1}{4}(S|\psi\rangle \otimes TST|\psi\rangle) = R_z((\phi)|\psi\rangle \quad \text{, con } \cos\phi = 3/5$$

Para el resto de los casos hay que demostrar que:

$$\frac{1}{4}\left[(-|01\rangle - |10\rangle + |11\rangle) \otimes S|\psi\rangle + (-|01\rangle - |10\rangle + |11\rangle) \otimes TST|\psi\rangle = \right.$$

$$= Z|\psi\rangle$$

$$\frac{1}{4}\left[(-|01\rangle - |10\rangle + |11\rangle) \otimes (S|\psi\rangle + TST|\psi\rangle)\right] = Z|\psi\rangle$$

$$\frac{1}{4}\left[(-|01\rangle - |10\rangle + |11\rangle) \otimes ((S+TST)|\psi\rangle)\right] = Z|\psi\rangle$$

Sabiendo que en cualquiera de los estados de los dos primeros qubits debe dar $Z|\psi\rangle$, se debe cumplir que

$$(S + XSX) = Z$$

(T se sustituye por X porque ya no influyen los qubits)

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$XSX = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ i & 0 \end{pmatrix} = \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix}$$

$$S + XSX = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} + \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1+i & 0 \\ 0 & i+1 \end{pmatrix} = \underbrace{(1+i)}_{\text{Fase Global}}\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}}_{Z}$$

Bloque 3: Algoritmos Cuánticos

Alumno: Álvaro Manuel Aparicio Morales

I Certificado de Extensión Universitaria en Computación Cuántica (2024-2025)

Ejercicio 52

```
from qiskit import QuantumCircuit
import numpy as np
```

```
# Circuito de 16 qubits
qc = QuantumCircuit(16)
qc.h(0)
for q in range(1,qc.num_qubits):
    qc.cx(q-1,q)
print(qc)
```

```
      ┌───┐                                                                 »
 q_0: ┤ H ├──■──────────────────────────────────────────────────────────────»
      └───┘┌─┴─┐                                                            »
 q_1: ─────┤ X ├──■─────────────────────────────────────────────────────────»
           └───┘┌─┴─┐                                                       »
 q_2: ──────────┤ X ├──■────────────────────────────────────────────────────»
                └───┘┌─┴─┐                                                  »
 q_3: ───────────────┤ X ├──■───────────────────────────────────────────────»
                     └───┘┌─┴─┐                                             »
 q_4: ────────────────────┤ X ├──■──────────────────────────────────────────»
                          └───┘┌─┴─┐                                        »
 q_5: ─────────────────────────┤ X ├──■─────────────────────────────────────»
                               └───┘┌─┴─┐                                   »
 q_6: ──────────────────────────────┤ X ├──■────────────────────────────────»
                                    └───┘┌─┴─┐                              »
 q_7: ───────────────────────────────────┤ X ├──■───────────────────────────»
                                         └───┘┌─┴─┐                         »
 q_8: ────────────────────────────────────────┤ X ├──■──────────────────────»
                                              └───┘┌─┴─┐                    »
 q_9: ─────────────────────────────────────────────┤ X ├──■─────────────────»
                                                   └───┘┌─┴─┐               »
q_10: ──────────────────────────────────────────────────┤ X ├──■────────────»
                                                        └───┘┌─┴─┐          »
q_11: ───────────────────────────────────────────────────────┤ X ├──■───────»
                                                             └───┘┌─┴─┐     »
q_12: ────────────────────────────────────────────────────────────┤ X ├──■──»
                                                                  └───┘┌─┴─┐»
q_13: ─────────────────────────────────────────────────────────────────┤ X ├»
                                                                       └───┘»
q_14: ──────────────────────────────────────────────────────────────────────»
                                                                            »
q_15: ──────────────────────────────────────────────────────────────────────»
                                                                            »
«
« q_0: ──────────
«
« q_1: ──────────
«
« q_2: ──────────
«
« q_3: ──────────
«
« q_4: ──────────
«
« q_5: ──────────
«
« q_6: ──────────
«
« q_7: ──────────
«
« q_8: ──────────
«
« q_9: ──────────
«
«q_10: ──────────
«
«q_11: ──────────
«
```

Para poder optimizar el circuito anterior, podemos trabajar con el circuito mediante la apliación de puertas en paralelo. Para lograrlo se puede seguir la estructura de un árbol binario donde cada nivel es una etapa donde se pueden aplicar puertas cnots paralelas.

A continuación generamos el circuito optimizado:

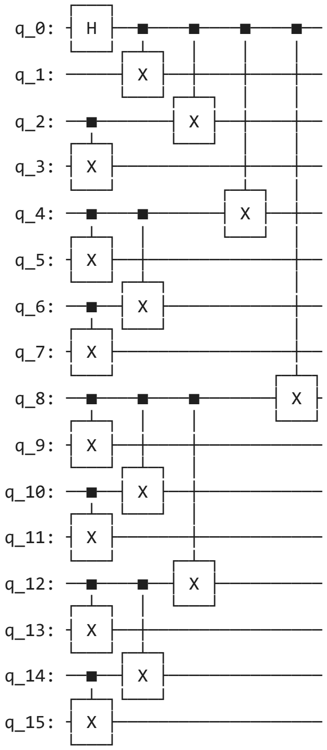
```
# Crear un nuevo circuito
qc_optimized = QuantumCircuit(16)

# 1. Aplicar la puerta Hadamard al primer qubit
qc_optimized.h(0)

# 2. Usar CNOT en paralelo dividiendo los qubits
step = 1
while step < qc.num_qubits:
    for i in range(0, qc.num_qubits - step, 2 * step):
        qc_optimized.cx(i, i + step)
    step *= 2

# Dibujar el circuito optimizado
print("Circuito optimizado con profundidad 5:")
print(qc_optimized)
```

```
Circuito optimizado con profundidad 5:
```



Ejercicio 53

a) Quantum Volume: El quantum volume, es una metrica utilizada para evaluar la capacidad y los ratios de error de un computador cuántico. Esta metrica permite realizar comparaciones entre distintas máquinas cuánticas.

Esta medida expresa el tamaño máximo de un circuito cuántico cuadrado (circuito cuya profundidad es igual al número de cubits del circuito) que puede ser implementado satisfactoriamente.

El "Quantum Volume", se calcula como:

$QV = 2^d$, siendo d la máxima profundidad de un circuito cuadrado que haya podido ser ejecutado con una fidelidad mayor que un umbral.

Los 3 procesadores con mayor "Quantum Volume" a fecha de 23 de noviembre son:

Computador Cuántico	Quantum Volume
"Quantinuum System Model H2-1"	2^{21}
"Quantinuum System Model H1-1"	2^{20}
"Quantinuum System Model H2"	2^{16}

b) EPLG → Error-Per-Layered Gate, es una métrica que permite medir la calidad de un circuito en base al error producido en cada capa del circuito.

Una capa, es un conjunto de puertos que se ejecutan simultaneamente.

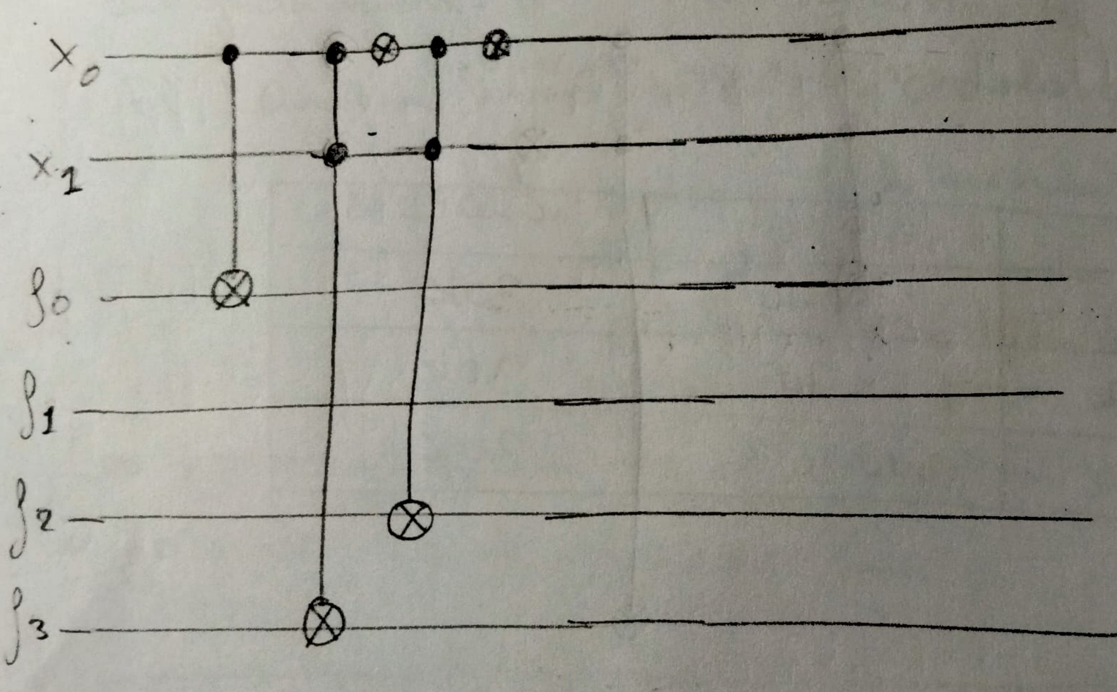
CLOPS → Circuit-Layer Operations Per Second, es una métrica para medir el rendimiento de una maquina cuantica de IBM. Esta métrica, mide el numero de capas de circuitos cuánticos que el ordenador cuantico puede ejecutar por segundo.

Nombre	EPLG	CLOPS
ibm-kyiv	$1.57 \cdot 10^{-2}$	30k
ibm-sherbrook	$1.74 \cdot 10^{-2}$	30k
ibm-brisbane	$2.16 \cdot 10^{-2}$	30K

57) $x \in \{0, 1, 2, 3\}$ $\qquad f(x) = x^2$

Tabla de verdad

x	$x_1 x_0$	$f(x)$	$f_3 f_2 f_1 f_0$
0	00	0	0000
1	01	1	0001
2	10	4	0100
3	11	9	1001



60) Siendo $U_{QFT}|x\rangle = \frac{1}{\sqrt{N}} \cdot \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |x\rangle$

$$U_{QFT}^{-1}|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{-2\pi i x y}{N}} |x\rangle$$

Vamos a probar que $U_{QFT}^{-1}(U_{QFT})|x\rangle = |x\rangle$

- Sustituimos

$$\frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{-2\pi i x y}{N}} \frac{1}{\sqrt{N}} \sum_{z=0}^{N-1} e^{\frac{2\pi i x z}{N}} |x\rangle =$$

$$= \frac{1}{N} \sum_{y=0}^{N-1} e^{\frac{-2\pi i x y}{N}} \sum_{z=0}^{N-1} e^{\frac{2\pi i x z}{N}} |x\rangle =$$

$$= \frac{1}{N} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} e^{\frac{-2\pi i x y}{N}} \cdot e^{\frac{2\pi i x z}{N}} |x\rangle = \frac{1}{N} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} e^{\frac{2\pi i x y}{N} - \frac{2\pi i x z}{N}} |x\rangle$$

$$= \frac{1}{N} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} e^{\frac{2\pi i x (y-z)}{N}} |x\rangle = \sum_{y=0}^{N-1} \underbrace{\frac{1}{N} \sum_{z=0}^{N-1} e^{\frac{2\pi i x (y-z)}{N}}}_{\text{Expresión de delta de Kronecker}}$$

$$\boxed{\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i j k / N} = \delta_{j0 \bmod N}}$$

Que resulta 1 cuando

$\delta_{ij} = 1$ si $i = j$

$$\rightarrow \sum_{y=0}^{N-1} \delta_{yz} |x\rangle$$

Para que $U_{QFT}^{-1}(U_{QFT})|x\rangle = |x\rangle$

$$\sum_{y=0}^{N-1} \delta_{y,z} |x\rangle = |x\rangle$$

Por tanto $\sum_{y=0}^{N-1} \delta_{y,z} = 1$, por lo que $\boxed{y = z}$

61) Debemos demostrar que $\frac{1}{\sqrt{r}}\sum_{j=0}^{r-1}|u_j\rangle = |1\rangle$

Sabemos que $|u_j\rangle$ podemos expresarlo como:

1 $$\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i k \frac{j}{r}}|a^k \bmod N\rangle$$

Por tanto debemos demostrar que:

$$\frac{1}{\sqrt{r}}\sum_{j=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{-2\pi i k \frac{j}{r}}|a^k \bmod N = |1\rangle$$

Tal y como se indica $a = 7$ y $N = 15$. Por tanto vamos a calcular el periodo de la función. que será el menor r que cumple que $\boxed{|a^r \bmod N = 1|}$

$k = 1 \rightarrow 7^1 \bmod 15 = 7$

$k = 2 \rightarrow 7^2 \bmod 15 = 4$

$k = 3 \rightarrow 7^3 \bmod 15 = 13$

$k = 4 \rightarrow 7^4 \bmod 15 = 1 \rightarrow$ El periodo es 4

Tras obtener este valor sustituimos en nuestro sumatorio.

$$\frac{1}{\sqrt{4}}\sum_{j=0}^{3}\frac{1}{\sqrt{4}}\sum_{k=0}^{3} e^{-2\pi i k \frac{j}{4}}|7^k \bmod 15\rangle =$$

$$= \frac{1}{2}\sum_{j=0}^{3}\frac{1}{2}\sum_{k=0}^{3} e^{-\pi i k \frac{j}{2}}|7^k \bmod 15\rangle = \frac{1}{4}\sum_{j=0}^{3}\sum_{k=0}^{3} e^{-\pi i k \frac{j}{2}}|7^k \bmod 15\rangle$$

$$= \frac{1}{4}\Big(e^{-\pi i 0 \cdot \frac{0}{2}}|1\rangle + e^{-\pi i 1 \cdot \frac{0}{2}}|7\rangle + e^{-\pi i 2 \cdot \frac{0}{2}}|4\rangle + e^{-\pi i 3 \cdot \frac{0}{2}}|13 +$$

$$e^{-\pi i 0 \cdot \frac{1}{2}}|1\rangle + e^{-\pi i 1 \cdot \frac{1}{2}}|7 + e^{-\pi i 2 \cdot \frac{1}{2}}|4\rangle + e^{-\pi i 3 \frac{1}{2}}|13 +$$

$$e^{-\pi i 0 \frac{2}{2}}|1\rangle + e^{-\pi i 1 \cdot \frac{2}{2}}|7\rangle + e^{-\pi i 2 \cdot \frac{2}{2}}|4\rangle + e^{-\pi i 3 \cdot \frac{2}{2}}|13 +$$

$$e^{-\pi i 0 \frac{3}{2}}|1\rangle + e^{-\pi i 1 \cdot \frac{3}{2}}|7\rangle + e^{-\pi i 2 \cdot \frac{3}{2}}|4\rangle + e^{-\pi i 3 \cdot \frac{3}{2}}|13\Big)$$

61) Continuación

$$= \frac{1}{4}\left(e^{0}|1\rangle + e^{0}|7\rangle + e^{0}|4\rangle + e^{0}|13\rangle + e^{0}|1\rangle + e^{-\frac{\pi i}{2}}|7\rangle + e^{-\pi i}|4\rangle + e^{-\frac{3\pi i}{2}}|13\rangle + e^{0}|1\rangle + e^{-\pi i}|7\rangle + e^{-2\pi i}|4\rangle + e^{-3\pi i}|13\rangle + e^{0}|1\rangle + e^{-\frac{3\pi i}{2}}|7\rangle + e^{-3\pi i}|4\rangle + e^{-\frac{9\pi i}{2}}|13\rangle\right) = \frac{1}{4}\left(4|1\rangle + |7\rangle + e^{-\frac{\pi i}{2}}|7\rangle + e^{-\pi i}|7\rangle + e^{-\frac{3\pi i}{2}}|7\rangle + |4\rangle + e^{-\pi i}|4\rangle + e^{-2\pi i}|4\rangle + e^{-3\pi i}|4\rangle + |13\rangle + e^{-\frac{3\pi i}{2}}|13\rangle + e^{-3\pi i}|13\rangle + e^{-\frac{3\pi i}{2}}|13\rangle\right) =$$

$$= \frac{1}{4}\left(4|1\rangle + |7\rangle + e^{-3\pi i}|7\rangle + |4\rangle + e^{-5\pi i}|4\rangle + |13\rangle + e^{-6\pi i}|13\rangle\right)$$

$\downarrow$ Aplicando Euler

$e^{\phi i} = \cos\phi + i\,\text{sen}\,\phi$

$$= \frac{1}{4}\left(4|1\rangle + |7\rangle - |7\rangle + |4\rangle - |4\rangle + |13\rangle - |13\rangle\right) =$$

$$= |1\rangle$$

Corroboramos que: $\boxed{\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle}$