Nota 2: *QRAM* y *QDAC*

En esta nota se describen dos circuitos cuánticos que, a su vez, son recursos básicos para la definición de otros circuitos. Las siglas hacen referencia a dos componentes comunes de los circuitos electrónicos: *RAM* (*Random Access Memory*) y *DAC* (*Digital to Analog Converter*).

1. *QRAM*

Considérese

- un conjunto de $N = 2^n$ **datos** de m bits $\{d_0, d_1, \ldots, d_{N-1}\}$,
- una base ortonormal $\{|v_0\rangle, |v_1\rangle, \ldots, |v_{N-1}\rangle\}$ del espacio $V = C^n$ de las n-tuplas de números complejos.

Una *QRAM* es un circuito cuántico compuesto de $n+m$ qubits (sin contar los qubits auxiliares) que ejecuta la siguiente operación:

$$\sum_j \alpha_j |v_j\rangle \times |00\ldots0\rangle \xrightarrow{QRAM(\{d_j\})} \sum_j \alpha_j |v_j\rangle \times |d_j\rangle. \quad (1)$$

A una superposición de vectores $|v_j\rangle$, interpretados como **direcciones**, este sistema asocia una superposición de pares enlazados dirección-dato $|v_j\rangle \times |d_j\rangle$.

Si se utiliza la base natural $\{|j_0 j_1 \ldots j_{n-1}\rangle \mid j_k \in \{0, 1\}, \forall k = 0, 1, \ldots, n\text{-}1\}$, una *QRAM* materializa m funciones binarias $f_0, f_1, \ldots, f_{m-1}$ de n variables binarias $j_0, j_1, \ldots, j_{n-1}$ definidas por la siguiente tabla de la verdad:

$j_0\, j_1 \ldots j_{n-1}$	$f_0\, f_1 \ldots f_{m-1}$
00 … 0	$d_{00}\, d_{01} \ldots d_{0\, m-1}$
00 … 1	$d_{10}\, d_{11} \ldots d_{1\, m-1}$
…	…
11 … 1	$d_{n-1\, 0}\, d_{n-1\, 1} \ldots d_{n-1\, m-1}$

Tabla 1 Tabla de la verdad

Se describe en Nota1 un operador unitario U_f, definido a partir de una tabla, que ejecuta la operación

$$|j_0 \ldots j_{n-1}\rangle \times |00 \ldots 0\rangle \xrightarrow{U_f} |j_0 \ldots j_{n-1}\rangle \times |f_0(j_0, \ldots, j_{n-1}) \ldots f_{m-1}(j_0, \ldots, j_{n-1})\rangle = |j\rangle \times |d_j\rangle.$$

Por linealidad, el operador U_f aplicado al estado initial

$$\sum_j \alpha_j |j\rangle \times |00 \ldots 0\rangle, \qquad (2)$$

ejecuta la operación (1), con $|v_j\rangle = |j\rangle$.

Supóngase que se pueda definir un operador U_d que genera un estado de superposición de direcciones

$$|\varphi\rangle = \sum_j \alpha_j |j\rangle \,. \qquad (3)$$

Entonces, el circuito de la Fig.1 genera el estado

$$\sum_j \alpha_j |j\rangle \times |d_j\rangle. \qquad (4)$$

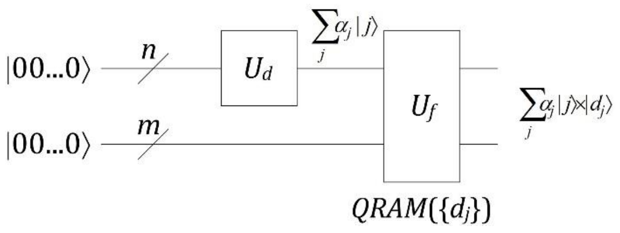

Figura 1 Generación del estado $\sum_j \alpha_j |j\rangle \times |d_j\rangle$.

Si todos los coeficientes α_j son iguales a $1/\sqrt{N}$, el operador que genera el estado $|\varphi\rangle$ es un conjunto de n puertas de Hadamard. En este caso, el estado $|\varphi\rangle$ es la superposición de las N posibles direcciones $|j\rangle$, todas con la misma probabilidad $1/\sqrt{N}$. Se podría interpretar que de ahí viene el nombre *QRAM*

(*Quantum Random Access Memory*): la probabilidad de acceso a un dato $|d_j\rangle$ no depende de su dirección $|v_j\rangle$.

Ejemplo 1

Considérese la siguiente tabla de la verdad:

I	D
000	1101
001	0111
010	1011
011	0110
100	1111
101	0111
110	0111
111	1011

Tabla 2 Ejemplo 1

Con puertas de Hadamard se genera el estado inicial

$$\sum_j \frac{1}{\sqrt{8}}|j\rangle \times |0000\rangle.$$

El programa `nota2_1.py` describe un sistema que ejecuta las operaciones

$$\sum_j |000\rangle \times |0000\rangle \xrightarrow{H^{\times 4}} \sum_j \frac{1}{\sqrt{8}}|j\rangle \times |0000\rangle \xrightarrow{QRAM(\{d_j\})} \sum_j \frac{1}{\sqrt{8}}|j\rangle \times |d_j\rangle.$$

La ejecución del programa genera el siguiente estado cuántico:

```
output vector:
0.354|0001101⟩ + 0.354|0010111⟩ + 0.354|0101011⟩ + 0.354|0110110⟩ +
0.354|1001111⟩ + 0.354|1010111⟩ + 0.354|1100111⟩ + 0.354|1111011⟩
```

2. *QDAC*

Considérese un conjunto de $n+m$ qubits preparados en el estado

$$|\psi\rangle = \sum_j \alpha_j |v_j\rangle \times |d_j\rangle, \tag{5}$$

donde

- los vectores $\{|v_j\rangle\}$ constituyen una base ortonormal del espacio $V = C^n$,
- los vectores $|d_j\rangle$ son estados básicos del espacio $V = C^m$ que representan datos de m bits.

El estado (5) podría haber sido generado por el circuito de la Fig.1, a partir de un estado de superposición

$$|\varphi\rangle = \sum_j \alpha_j |v_j\rangle \tag{6}$$

(con $|v_j\rangle = |j\rangle$), pero no es la única posibilidad. Por ejemplo, en el algoritmo *HHL* [2], se utiliza un circuito de estimación de la fase ([1], Sec.7.4) para generar un estado del tipo (5), siendo $\{|v_j\rangle\}$ una base ortonormal constituida por los vectores propios de una matriz unitaria y $\{|d_j\rangle\}$ un conjunto de datos asociados a los valores propios correspondientes. Sea cual sea el caso, llámese T la operación que genera el estado (5) a partir del estado (6), es decir,

$$\sum_j \alpha_j |v_j\rangle \times |00 \dots 0\rangle \xrightarrow{T(\{d_j\})} \sum_j \alpha_j |v_j\rangle \times |d_j\rangle. \tag{7}$$

Un *QDAC* es un circuito cuántico que ejecuta la siguiente operación:

$$|\psi\rangle = \sum_j \alpha_j |v_j\rangle \times |d_j\rangle \xrightarrow{QDAC} C \cdot \sum_j \alpha_j d_j |v_j\rangle, \tag{8}$$

donde el coeficiente C normaliza el estado final de tal manera que

$$\sum_j \left| C\alpha_j d_j \right|^2 = 1.$$

Este circuito convierte una información digital (un estado básico $|d_j\rangle$ enlazado con una dirección $|v_j\rangle$) en una información analógica, es decir, un coeficiente d_j asociado a la dirección $|v_j\rangle$. Para realizar esta operación se utilizan cuatro registros: un registro de n qubits que almacena las direcciones $|v_j\rangle$, un registro de m qubits que almacena los datos $|d_j\rangle$, un registro de p qubits que representa ángulos comprendidos entre 0 y π radianes, y un qubit auxiliar (ancillary qubit). El circuito se muestra en la figura 2.

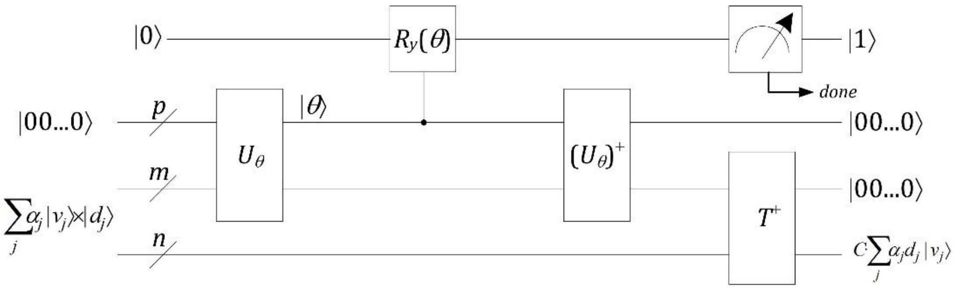

Figura 2 Operador *QDAC*

- El operador U_θ aplicado a un estado básico realiza la siguiente operación:

$$U_\theta|d_j\rangle\times|00...0\rangle = |d_j\rangle\times|\theta_j\rangle \quad (9)$$

donde θ_j es una estimación, con p-2 bits fraccionarios, de $2\cdot sin^{-1}(d_j/2^m)$. Por tanto, teniendo en cuenta que

$$0 \leq d_j/2^m < 1,$$

se deduce que

$$0 \leq 2\cdot sin^{-1}(d_j/2^m) < \pi.$$

En numeración binaria,

$$2\cdot sin^{-1}(d_j/2^m) \cong \theta_{j1}\theta_{j2}.\theta_{j3}\ ...\theta_{jp}. \quad (10)$$

El operador U_θ puede sintetizarse a partir de un circuito digital que calcula la función $2\cdot sin^{-1}(d_j/2^m)$ ([1], Sec.6.1.2 y 6.1.3). En el apéndice B de [4] se describe un método de síntesis basado en series de Taylor.

El operador $R_y(\theta_j)$, aplicado al qubit auxiliar, preparado en el estado $|0\rangle$, ejecuta la siguiente operación ([1], Equ.4.56):

$$R_y(\theta_j)\ |0\rangle = sin(\theta_j/2)|1\rangle + cos(\theta_j/2)|0\rangle = (d_j/2^m)|1\rangle + \sqrt{1-\left(\frac{d_j}{2^m}\right)^2}|0\rangle. \quad (11)$$

- El operador U_θ^+ invierte la operación (9).
- El operador T^+ invierte la operación (7).

El operador $R_y(\theta_j)$ puede materializarse con puertas $R_y(2^k)$ condicionales. Según (10),

$$\theta_j = \theta_{j1}\cdot 2^1 + \theta_{j2}\cdot 2^0 + \theta_{j3}\cdot 2^{-1} + \ldots + \theta_{jp}\cdot 2^{-(p-2)}.$$

Se usan puertas binarias $CR_y(q, 2^k)$ que ejecutan la operación

$$CR_y(q,\, 2^k)|\psi\rangle = R_y(2^k)|\psi\rangle \text{ si } q = |1\rangle \text{ y } CR_y(q,\, 2^k)|\psi\rangle = |\psi\rangle \text{ si } q = |0\rangle,$$

es decir, operadores unarios controlados ([1], Sec.4.3.2). Finalmente (Fig.3)

$$R_y(\theta_j) = CR_y(|\theta_{jp}\rangle, 2^{-(p-2)})\cdot \ldots \cdot CR_y(|\theta_{j3}\rangle, 2^{-1})\cdot CR_y(|\theta_{j2}\rangle, 2^0)\cdot CR_y(|\theta_{j1}\rangle, 2^1). \qquad (12)$$

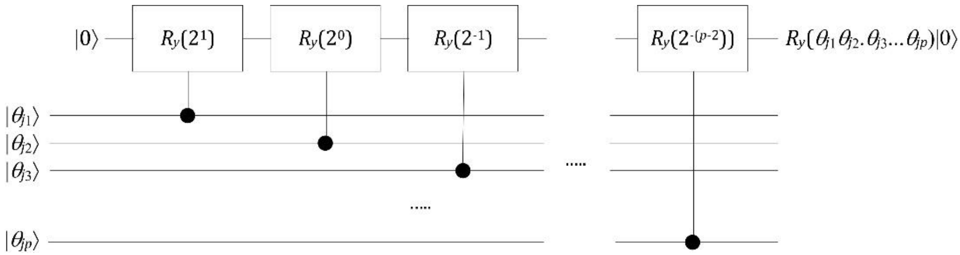

Figura 3 Operador $R_y(\theta_j)$

Las operaciones ejecutadas por el circuito de la Fig.2 son las siguientes (I representa el operador identidad sea cual sea el tamaño del registro):

$$\sum_j \alpha_j|v_j\rangle\times|d_j\rangle\times|00\ldots0\rangle\times|0\rangle \xrightarrow{I\times U_\theta\times I} \sum_j \alpha_j|v_j\rangle\times|d_j\rangle\times|\theta_j\rangle\times|0\rangle$$

$$\xrightarrow{I\times I\times R_y(\theta_j)} \sum_j \alpha_j|v_j\rangle\times|d_j\rangle\times|\theta_j\rangle\times\left(\frac{d_j}{2^m}|1\rangle + \sqrt{1-\left(\frac{d_j}{2^m}\right)^2}\,|0\rangle\right)$$

$$\xrightarrow{I\times U_\theta^+\times I} \sum_j \alpha_j|v_j\rangle\times|d_j\rangle\times|00\ldots0\rangle\times\left(\frac{d_j}{2^m}|1\rangle + \sqrt{1-\left(\frac{d_j}{2^m}\right)^2}\,|0\rangle\right)$$

$$\xrightarrow{T^+\times I\times I} \sum_j \alpha_j|v_j\rangle\times|00\ldots0\rangle\times|00\ldots0\rangle\times\left(\frac{d_j}{2^m}|1\rangle + \sqrt{1-\left(\frac{d_j}{2^m}\right)^2}\,|0\rangle\right).$$

Finalmente, se hace una medición del qubit auxiliar. Se repite la ejecución del algoritmo hasta obtener una medición igual a 1. Entonces, el estado final será

$$C \cdot \sum_j \alpha_j d_j |v_j\rangle \times |00 \ldots 0\rangle \times |00 \ldots 0\rangle \times |1\rangle,$$

es decir, el resultado de la operación (8) si solo se tiene en cuenta el primer registro ya que los otros están en un estado básico desenlazado del primero.

Ejemplo 2

Añádase al programa del ejemplo 1 la descripción del circuito de la Fig.2. Se obtiene el programa `nota2_2.py`. A efectos de simulación, para el operador U_θ se ha definido una tabla `truth_table` y el operador `Uf2` asociado. Se ejecuta el programa hasta obtener una medición del qubit auxiliar igual a 1. Este es el resultado:

```
measurements:
q(0),q(4),q(5),q(6),q(7),q(8),q(9),q(10),q(11),q(12)=1000000000

qubits: (cirq.LineQubit(0),)
output vector: |1⟩

qubits: (cirq.LineQubit(1), cirq.LineQubit(2), cirq.LineQubit(3))
output vector:
0.471|000⟩ + 0.225|001⟩ + 0.389|010⟩ +
0.189|011⟩ + 0.536|100⟩ + 0.225|101⟩ +
0.225|110⟩ + 0.389|111⟩
```

En este programa, el qubit auxiliar es $q(0)$ y el registro que contiene las direcciones $|j\rangle$ es $(q(1), q(2), q(3))$. Según la tabla 2 el resultado correcto es

$$C\cdot(13|000\rangle+7|001\rangle+11|010\rangle+6|011\rangle+15|100\rangle+7|101\rangle+7|110\rangle+11|111\rangle).$$

El programa siguiente calcula el valor de C y genera el estado normalizado:

```
X = [13,7,11,6,15,7,7,11]
def normalize(X):
    suma = 0
    for i in range(8):
        suma = suma + (X[i])**2
    N = suma**0.5
    NX = [(x/N) for x in X]
    return NX
print(normalize(X))
```

El resultado de la normalización es

```
[0.45425676257949793,      0.24459979523511427,      0.3843711067980367,
0.20965696734438366,      0.5241424183609591,      0.24459979523511427,
0.24459979523511427, 0.3843711067980367]
```

Desde luego se observan diferencias con el resultado de la simulación del circuito, pero son esperables teniendo en cuenta la poca precisión (5 bits) elegida para la definición del ángulo θ_j.

Comentarios

- Si la probabilidad de que la medición del qubit auxiliar dé como resultado 1 es igual a p, entonces, como promedio, será necesario ejecutar el algoritmo aproximadamente $1/p$ veces. Podría ser un número muy grande. Se ha definido un método (*quantum amplitude amplification*, [3]) que permite reducir el tiempo de ejecución de algoritmos de este tipo. Se basa en una generalización del algoritmo de Grove ([1], Sec.7.6).
- Si se interpreta el vector $|d_j\rangle$ como la representación de un entero en complemento a 2, es decir, $-2^{m-1} \leq d_j < 2^{m-1}$, con lo cual

  $$-0.5 \leq d_j/2^m < 0.5,$$

  y si se define el ángulo θ_j como siendo una estimación, con p-2 bits fraccionarios, de $2\cdot sin^{-1}(d_j/2^m)$, entonces

  $$-\pi/3 \leq 2\cdot sin^{-1}(d_j/2^m) < \pi/3.$$

  En numeración binaria,

  $$2\cdot sin^{-1}(d_j/2^m) \cong \theta_{j1}\theta_{j2}.\theta_{j3}\ \ldots\theta_{jp}, \tag{13}$$

  siendo θ_{j1} el bit de signo. Por tanto, la relación (12) se sustituye por

  $$R_y(\theta_j) = CR_y(|\theta_{jp}\rangle,2^{-(p-2)})\cdot\ \ldots\ \cdot CR_y(|\theta_{j3}\rangle,2^{-1})\cdot CR_y(|\theta_{j2}\rangle,2^{0})\cdot CR_y(|\theta_{j1}\rangle,-2^{1}). \tag{14}$$

3. *QDAC funcional*

Se puede generalizar el concepto de convertidor *QDAC*. Considérese de nuevo un conjunto de $n+m$ qubits preparados en el estado (5), así como una función

$$f\colon \{0, 1\}^m \rightarrow \{0, 1\}^s. \tag{15}$$

Entonces, se puede definir un circuito cuántico que ejecuta la operación

$$|\psi\rangle = \sum_j \alpha_j |v_j\rangle \times |d_j\rangle \rightarrow C \cdot \sum_j \alpha_j f(d_j) |v_j\rangle. \tag{16}$$

El circuito es el mismo que el de la Fig.3. Solo debe modificarse la definición del operador U_θ:

- El operador U_θ realiza la operación (9) donde θ_j es una estimación, con p-2 bits fraccionarios, de $2 \cdot sin^{-1}(f(d_j)/2^s)$. Por tanto, teniendo en cuenta que

  $$0 \leq f(d_j)/2^s < 1,$$

  se deduce que

  $$0 \leq 2 \cdot sin^{-1}(f(d_j)/2^s) < \pi,$$

  con lo cual, en numeración binaria,

  $$2 \cdot sin^{-1}(f(d_j)/2^s) \cong \theta_{j1}\theta_{j2}.\theta_{j3} \ldots \theta_{jp}. \tag{17}$$

Las operaciones ejecutadas por el circuito de la Fig.3, con la nueva definición de U_θ, son las siguientes:

$$\sum_j \alpha_j |v_j\rangle \times |d_j\rangle \times |00 \ldots 0\rangle \times |0\rangle \xrightarrow{I \times U_\theta \times I} \sum_j \alpha_j |v_j\rangle \times |d_j\rangle \times |\theta_j\rangle \times |0\rangle$$

$$\xrightarrow{I \times I \times R_y(\theta_j)} \sum_j \alpha_j |v_j\rangle \times |d_j\rangle \times |\theta_j\rangle \times \left(\frac{f(dj)}{2^s} |1\rangle + \sqrt{1 - \left(\frac{f(dj)}{2^s}\right)^2}\, |0\rangle \right)$$

$$\xrightarrow{I \times U_\theta^+ \times I} \sum_j \alpha_j |v_j\rangle \times |d_j\rangle \times |00 \ldots 0\rangle \times \left(\frac{f(dj)}{2^s} |1\rangle + \sqrt{1 - \left(\frac{f(dj)}{2^s}\right)^2}\, |0\rangle \right)$$

$$\xrightarrow{QRAM^+ \times I \times I} \sum_j \alpha_j |v_j\rangle \times |00 \ldots 0\rangle \times |00 \ldots 0\rangle \times \left(\frac{f(dj)}{2^s} |1\rangle + \sqrt{1 - \left(\frac{f(dj)}{2^s}\right)^2}\, |0\rangle \right). \tag{18}$$

Finalmente, se hace una medición del qubit auxiliar. Se repite la ejecución del algoritmo hasta obtener una medición igual a 1. Entonces, el estado final será

$$C \cdot \sum_j \alpha_j f(d_j) |v_j\rangle \times |00 \ldots 0\rangle \times |00 \ldots 0\rangle \times |1\rangle,$$

es decir, el resultado de la operación (16) si solo se tiene en cuenta el primer registro ya que los otros están desenlazados del primero.

Referencias

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

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

[3] G. Brassard, P. Hoyer, M. Mosca, A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. 305 (2002) 53–74.

[4] I. Cong, L. Duan, Quantum discriminant analysis for dimensionality reduction and classification, New J. Phys. 18(7) (2016) 073011.