# Nota 6: Multiplicador

En esta nota se describe un circuito que materializa un algoritmo convencional de multiplicación de números naturales.

## 1. Algoritmo

Considérense tres números naturales expresados en numeración binaria con $n$ bits:

$$a = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2^1 + a_0 2^0,$$
$$z = z_{n-1}2^{n-1} + z_{n-2}2^{n-2} + \ldots + z_1 2^1 + z_0 2^0,$$
$$x = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \ldots + x_1 2^1 + x_0 2^0.$$

El cálculo de

$$p = x + a \cdot z \bmod 2^n \qquad (1)$$

puede ejecutarse según el siguiente esquema:

$p_0 = (a_0 z_{n-1}2^{n-1} + a_0 z_{n-2}2^{n-2} + \ldots + a_0 z_1 2^1 + a_0 z_0 2^0),$

$p_1 = (a_1 z_{n-2}2^{n-1} + \ldots + a_1 z_1 2^2 + a_1 z_0 2^1),$

…

$p_{n-2} = (a_{n-2} z_1 2^{n-1} + a_{n-2} z_0 2^{n-2})$ +

$p_{n-1} = a_{n-1} z_0 2^{n-1}$,l

$p = (\ldots((x + p_0) + p_1) + \ldots + p_{n-2}) + p_{n-1},\ \bmod 2^n.$

Este esquema se reduce al cálculo de $n(n+1)/2$ productos binarios $a_i \cdot z_j$ con $i+j < n$, y de $n$ sumas mod $2^n$ de dos números naturales, es decir, sin tener en cuenta los acarreos finales.

En la Fig.1 se muestra un multiplicador cuántico en el cual los productos $a_i \cdot z_j$ se ejecutan con operadores $CCX$ (Toffoli) y la suma $p$ se ejecuta con sumadores del tipo propuesto en [2] y descrito en la nota 4 (Fig.3 y 4), sin el último qubit (acarreo de salida).

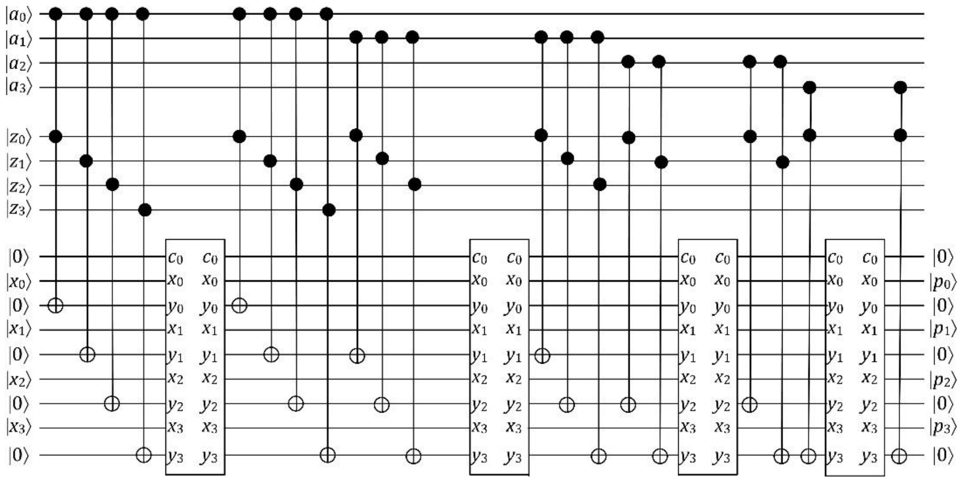

**Figura 1** Multiplicador módulo $2^n$ ($n$ = 4)

De forma general, un multiplicador de $n$ bits diseñado según el esquema de la Fig.1, constaría de $4n$ qubits, $n(n+1)$ operadores de Toffoli y $n$ sumadores. Los operadores de Toffoli calculan dos veces cada producto $a_i \cdot z_j$ para que los qubits $y_i$ estén en el estado $|0\rangle$ antes de cada etapa.

**Ejemplo 1**

El programa `nota6_1.py` describe el circuito de la Fig.1. Se utilizan las clases `CY` y `XOR` descritas en `nota4_1.py`, y se define una nueva clase `ADDER` que describe el sumador:

```
class ADDER(cirq.Gate):
    def __init__(self):
        super(ADDER, self)
    def _num_qubits_(self):
        return 9
    def _decompose_(self, qubits):
        c0,x0,y0,x1,y1,x2,y2,x3,y3 = qubits
        yield cy(c0,x0,y0)
        yield cy(y0,x1,y1)
        yield cy(y1,x2,y2)
        yield cy(y2,x3,y3)
        yield xor(y2,x3,y3)
        yield xor(y1,x2,y2)
        yield xor(y0,x1,y1)
        yield xor(c0,x0,y0)
    def _circuit_diagram_info_(self, args):
        return ["ADDER"] * self.num_qubits()
```

## 2. Comentarios

Compárense los circuitos de la Fig.6 de la nota 5 y de la Fig.1 de esta nota.

- El primero (basado en la *QFT*) utiliza el número mínimo $3n$ de qubits ($n$ qubits para cada uno de los tres operandos). En cambio, el segundo necesita $n$ qubits auxiliares, con lo cual, en total utiliza $4n$ qubits.
- El número de operadores de Toffoli, para el cálculo de los productos $a_i \cdot x_j$ o $a_i \cdot z_j$, es el mismo: $n(n+1)$.
- El multiplicador basado en la *QFT* utiliza rotaciones $R_\varphi$ controladas, con $\varphi = \pi/2^k$, $k$ = 0, 1, …, $n$-1, tanto para la ejecución de la transformada como para las sumas. En cambio, el multiplicador de la Fig.1 solo utiliza operadores *CX* y *CCX* (nota 4, Fig.3 y 4).

La ventaja del circuito basado en la *QFT* es que no necesita qubits auxiliares. Ello se consigue sustituyendo sumas binarias por rotaciones con ángulos perteneciendo a un conjunto finito de $n$ valores predefinidos. La ventaja de un multiplicador más convencional, como el de la Fig.1, es que solo necesita operadores *CX* y *CCX*. Además, este último puede sintetizarse con operadores *CX* y operadores $H$, $S = R_{\pi/2}$, $T = R_{\pi/4}$ y $T^+$ ([1], Fig.6.16).

Desde luego, sea cual sea el tipo de multiplicador, el diseño (*layout*) del circuito es un tema complejo: la distancia entre el qubit de control y el qubit objetivo de una puerta *CX* puede ser larga y, ello puede obligar a añadir qubits y puertas adicionales ([1], Fig.6.18).

## Referencias

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

[2] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, “A new quantum ripple-carry addition circuit”, 2004, *arXiv:quant-ph/0410184*.