Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Elementos y conceptos de computación cuántica

Autor: Diego Jesús Benjumea Gayango

Tutor: Pedro Pérez Fernández

Dep. Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Elementos y conceptos de computación cuántica

Autor:
Diego Jesús Benjumea Gayango

Tutor:
Pedro Pérez Fernández
Profesor Ayudante Doctor

Dep. Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Elementos y conceptos de computación cuántica

Autor: Diego Jesús Benjumea Gayango
Tutor: Pedro Pérez Fernández

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

En primer lugar quería agradecer a mi profesor Pedro, que me ha estado siguiendo y guiando desde el principio del curso, por todo el trabajo y preocupación que ha mostrado para que lograse mi objetivo final. También quería agradecer a todos mis compañeros y amigos por el apoyo y compresión que han tenido. Pero si hay alguien de quien realmente estoy orgulloso y agradecido esos son mis padres, que sin pedir nada a cambio me lo han dado todo, han hecho posible lo imposible para que finalizase mis estudios, y aunque no lo admitan sé cuánto han sufrido. Es por esto por lo que les estaré eternamente agradecido.

Diego Jesús Benjumea Gayango
Sevilla, 2018

Resumen

En los últimos años, la computación cuántica ha supuesto un gran desafío para los ingenieros y científicos, ya que plasmar por primera vez la teoría cuántica aplicada a la computación desde el papel a la vida real es una ardua tarea. En este texto estudiaremos los elementos y conceptos necesarios para entender cómo funciona la computación cuántica, así como una aplicación directa de la misma para factorizar un número en factores primos.

Abstract

In the last years, quantum computation has meant a large challenge for engineers and scientists, since capturing the quantum theory applied to computation from the paper to the real life for the first time is a hard task. In this paper we shall study the required elements and concepts for understanding how quantum computation works, as well as a direct application of it to factorize a number in prime factors.

Índice

Notación

$\mathbb{R}$	Cuerpo de los números reales
$\mathbb{C}$	Cuerpo de los números complejos
$\mathcal{H}$	Espacio de Hilbert
$\frac{d}{dt}$	Derivada temporal
A	Operador A
$A^\dagger$	Adjunto (conjugado y transpuesto) del operador A
$\det \lvert A \rvert$	Determinante del operador A
$\xrightarrow{A}$	Se aplica la operación A
$\vec{v}$	Vector v
$\lvert\psi\rangle$	Estado cuántico ψ (estado ket)
$\langle\phi\rvert$	Estado cuántico ϕ (estado bra)
$\langle\phi\rvert\psi\rangle$	Producto interno entre $\lvert\phi\rangle$ y $\lvert\psi\rangle$
i	Unidad imaginaria
α^*	Conjugado del número complejo α
e	Número e
$e^{i\phi}$	Exponencial compleja
$\lvert\alpha\rvert$	Módulo del número complejo α
$\sin x$	Función seno
$\cos x$	Función coseno
$\tan x$	Función tangente
■	Como queríamos demostrar
$\oplus$	Suma de módulo dos
$\otimes$	Producto tensorial
$\equiv$	Equivalente
$\log n$	Logaritmo natural
$\log_2 n$	Logaritmo en base dos
$O(a)$	Orden de magnitud de a
$a = b \pmod n$	a es congruente con b módulo n
$\text{mcd}(a,\, b)$	Máximo común divisor entre a y b
QFT	Transformada cuántica de Fourier
$\lim_{x \to x_0} f(x)$	Límite cuando x tiende a x_0 de $f(x)$
$p(x)$	Probabilidad de x
$\lfloor x \rfloor$	Función suelo
$\lceil x \rceil$	Función techo
$\perp$	Perpendicular

Introducción

En el presente texto vamos a tratar los elementos y los conceptos empleados en el estudio de la computación cuántica. Antes de profundizar en el tema, primero vamos a introducir el término de *computación*. Según la Real Academia Española, este significa: *"Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras"*, es decir, el principal objetivo de la computación es procesar información de forma automática. Para ello necesitamos una máquina o computador que realice esta tarea, partiendo de una base teórica que fundamente su comportamiento, así como de una técnica apropiada, viable y eficiente para llevarlo del mundo teórico al real. Los computadores actuales emplean modelos matemáticos complejos y están formados por millones de transistores, rondando el tamaño de los mismos entorno a los 7 nm (7×10^{-9} m). Para ganar potencia de cálculo lo que se hace es introducir más transistores en el chip, esto se hace reduciendo el tamaño de los mismos, pero llegará tal punto en que no será posible reducirlo más, ya que a escalas tan pequeñas empezarán a aparecer efectos cuánticos. Una de las posibles opciones para solucionar esto es el diseño de un nuevo modelo de computador: el computador cuántico.

Actualmente, empresas como IBM o D-Wave han apostado por el desarrollo de ordenadores cuánticos, siendo las mayores potencias mundiales en esta tecnología. Por ejemplo, el D-Wave 2000Q^{TM}, con una cifra que asciende a los 2000 qubits a fecha de Julio de 2018, es el ordenador cuántico con más qubits del mundo, pero este no es genérico, es decir, no es capaz de ejecutar cualquier algoritmo. Por otro lado IBM posee ordenadores de 20 qubits genéricos y está desarrollando otro de 50 que superará a los actuales. Un dato importante es que IBM ofrece a cualquier usuario la posibilidad de programar un ordenador cuántico desde su aplicación web *IBM Q Experience Composer* (figura 0.1) de forma gratuita, pero solo permite usar su computador cuántico de 5 qubits y un número limitado de veces. También ofrece la posibilidad de hacer una simulación, la cual el usuario elige el número de qubits deseados y sin ninguna otra restricción.

A continuación se detallan los diferentes capítulos que integran este trabajo. En el capítulo 1 se establecerán las bases de la computación cuántica, partiendo de una breve introducción a la mecánica cuántica. Posteriormente se introducirá el elemento más básico de información: el *qubit*, y se estudiarán sus propiedades básicas, así como su representación en la esfera de Bloch y correspondencia con el espín. A continuación se definirá el producto tensorial para trabajar con sistemas de múltiples qubits y finalmente se hablará sobre entrelazamiento cuántico, un fenómeno tanto extraño como impresionante al mismo tiempo.

En el capítulo 2 hablaremos de cómo realizar operaciones con qubits, tanto con uno solo, como con varios, e introduciremos algunas puertas cuánticas para implementar dichas operaciones. También introduciremos el concepto de sistema reversible y mostraremos que

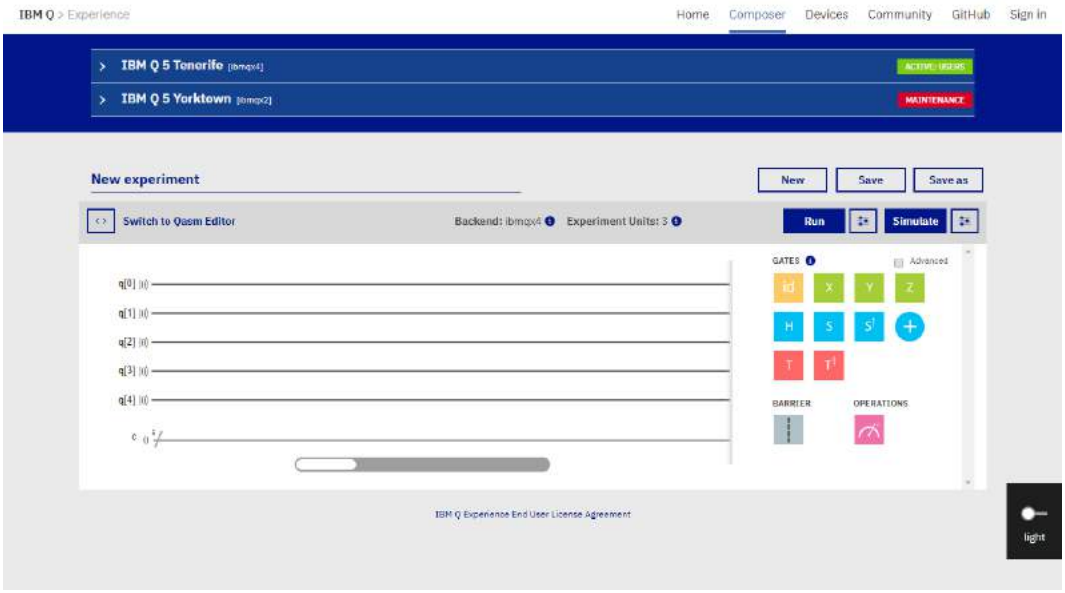


Figura 0.1 Editor online de IBM.

no existe ninguna puerta cuántica capaz de clonar un estado cualquiera. Concluiremos el capítulo con los ejemplos de superdense coding y teletransporte cuántico, que son formas complementarias de codificar información.

En el capítulo 3 veremos una aplicación directa de la computación cuántica: el algoritmo de Shor. Este algoritmo se basa en la idea de que el problema de hacer una factorización en números primos se puede reducir simplemente a encontrar el período de una cierta función, pudiéndose implementar eficientemente en un ordenador cuántico usando el algoritmo de *period findig*. También hablaremos sobre la versión cuántica de la transformada de Fourier, ya que juega un papel muy importante a la hora de encontrar el período.

1 Conceptos básicos

If you think you understand quantum mechanics, you don't understand quantum mechanics.

RICHARD FEYNMAN

1.1 Introducción a la mecánica cuántica

La mecánica cuántica es una teoría física que sirve para describir los fenómenos observados a escala microscópica que no podían ser descritos mediante la mecánica clásica. Surge a comienzos del siglo XX, cuando el físico alemán Max Planck estaba tratando de encontrar una solución a la *catástrofe ultravioleta*. Un cuerpo negro[1] emite radiación electromagnética por el simple hecho de estar a una temperatura, lo que se conoce como radiación térmica. Si se usaban las ecuaciones de la electrodinámica clásica para calcular la energía de esta radiación, sumándola para todas las frecuencias, se obtenía que la energía emitida era infinita. Planck tuvo la idea de, en el proceso aritmético, sustituir la integral por una suma discreta, obteniendo de este modo un resultado finito. Llegó a la conclusión de que la energía no se transmite de forma continua, sino mediante pequeños paquetes o cuantos de energía llamados fotones. La energía que porta un fotón es directamente proporcional a su frecuencia,

$$E = h\nu, \tag{1.1}$$

con h la constante que lleva el nombre de su descubridor, Planck. Años mas tardes de que Planck plantease esta hipótesis, fue usada por Albert Einstein para explicar el efecto fotoeléctrico, y posteriormente por Niels Bohr al formular un modelo atómico con niveles de energía cuantizados.

Al igual que en mecánica racional existen magnitudes como la posición, velocidad, energía, etc. que determinan el estado de un cierto sistema físico, en mecánica cuántica existe una magnitud especial llamada *vector de estado* o *función de onda*. La función de onda de un sistema se representa con la letra griega ψ encerrada entre una barra vertical y una especie de paréntesis en ángulo: $|\psi\rangle$. A esta notación se le llama **notación de Dirac**.

[1] Un cuerpo negro es un objeto teórico o ideal que absorbe toda la luz y toda la energía radiante que incide sobre él. Nada de la radiación incidente se refleja o pasa a través del cuerpo negro.

1.1.1 Notación de Dirac

La notación de Dirac es una forma estándar de representar estados cuánticos que fue introducida en 1939 por Paul Dirac [4]. Esta notación también es conocida como notación *bra-ket*, así el estado $|\psi\rangle$ se entiende como *ket*, y el estado $|\psi\rangle$ como *bra*, residentes en un espacio vectorial especial llamado *espacio de Hilbert*. De este espacio solo necesitamos saber que se trata de una generalización del espacio euclídeo, de dimensión no necesariamente finita donde se define un producto interno y, que a demás de vectores, también pueden residir funciones. Estos estados no son más que vectores complejos de dimensión igual a la dimensión del espacio de Hilbert donde residen.

Los estados $c|\psi\rangle$ y $c^*\langle\psi|$ son estados duales, es decir,

$$c|\psi\rangle = \Big(c^*\langle\psi|\Big)^\dagger, \tag{1.2}$$

con † el transpuesto conjugado y c un número complejo. A la función de onda también se le puede aplicar operadores lineales, por ejemplo, sea A un operador lineal y $|\psi\rangle$ un vector de estado, $A|\psi\rangle$ es dual a $A^\dagger\langle\psi|$, es decir,

$$A|\psi\rangle = \Big(A^\dagger\langle\psi|\Big)^\dagger. \tag{1.3}$$

La multiplicación de dos funciones de onda se realiza mediante el producto interno. Este se expresa de la forma *bra-ket* $\langle\phi|\psi\rangle$. Este producto es parecido al producto escalar convencional pero con vectores complejos. Las propiedades más significativas del producto interno son las siguientes:

1. Dados cualquier bra $\langle\phi|$, ket $|\psi_1\rangle$ y $|\psi_2\rangle$, y números complejos c_1 y c_2, entonces,

$$\langle\phi|\Big(c_1|\psi_1\rangle + c_2|\psi_2\rangle\Big) = c_1\langle\phi|\psi_1\rangle + c_2\langle\phi|\psi_2\rangle. \tag{1.4}$$

2. Dados cualquier ket $|\psi\rangle$, bra $\langle\phi_1|$ y $\langle\phi_2|$, y números complejos c_1 y c_2, entonces,

$$\Big(c_1\langle\phi_1| + c_2\langle\phi_2|\Big)|\psi\rangle = c_1\langle\phi_1|\psi\rangle + c_2\langle\phi_2|\psi\rangle. \tag{1.5}$$

3. En el producto interno, en general, la propiedad conmutativa no se cumple, de hecho, dados cualquier bra $\langle\phi|$ y ket $|\psi\rangle$ se cumple:

$$\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*. \tag{1.6}$$

1.1.2 Postulados de la mecánica cuántica

Una de las singularidades de la mecánica cuántica es que un sistema puede estar en varios estados al mismo tiempo [12], veamos un ejemplo. Supongamos que tenemos un electrón. Al ser este una partícula de espín 1/2, el valor del espín a lo largo del eje z (por ejemplo) puede tomar las cantidades $+\hbar/2$ o bien $-\hbar/2$, con $\hbar$ la constante de Planck reducida ($\frac{h}{2\pi}$). La mecánica cuántica nos dice que antes de ser medido, el valor del espín se encuentra en un estado de incertidumbre, es decir, no tiene ningún valor concreto y se encuentra en una superposición de estados. Es cuando decidimos medirlo cuando este se ve forzado a optar por uno de los valores. Por tanto, para este ejemplo, la función de onda sería una combinación

lineal de los vectores $|\uparrow\rangle$ y $|\downarrow\rangle$, que representan respectivamente los valores $+\hbar/2$ y $-\hbar/2$ de la proyección del espín a lo largo del eje z:

$$|\psi\rangle = \alpha|\uparrow\rangle + \beta|\downarrow\rangle, \quad \alpha, \beta \in \mathbb{C}. \tag{1.7}$$

La formulación matemática rigurosa de la mecánica cuántica se fundamenta en una serie de postulados. El número de postulados varía en función del libro donde se enuncien. En este texto vamos a usar la formulación de [1] y parte de [10].

Postulado I *El estado de un sistema físico S queda completamente descrito por un vector unitario $|\psi\rangle$, llamado vector de estado, o función de onda, y reside en el espacio de Hilbert $\mathcal{H}_S$ asociado al sistema.*

El postulado I propone el espacio de trabajo de la mecánica cuántica, que es el espacio de Hilbert. A su vez nos dice que todo sistema tiene asociado un vector de estado que lo describe, y que debe estar normalizado, esto es $\langle\psi|\psi\rangle = 1$. Esto es así debido a que la mecánica cuántica es una teoría probabilista y la probabilidad de encontrarnos el sistema en el estado $|\psi\rangle$ es 1. De forma general, dado el estado

$$|\psi\rangle = \sum_j \alpha_j |j\rangle, \tag{1.8}$$

con $|\alpha_j|^2$ la probabilidad de encontrar el sistema en el estado $|j\rangle$, se cumple que

$$\langle\psi|\psi\rangle = \sum_j \alpha_j^* \alpha_j = \sum_j |\alpha_j|^2 = 1, \tag{1.9}$$

ya que la suma total de probabilidades tiene que ser 1.

Postulado II *La evolución de un sistema cuántico **cerrado** viene dada por una transformación unitaria. Es decir, el estado $|\psi\rangle$ del sistema en el instante t_1 está relacionado con el estado $|\psi'\rangle$ en el instante t_2 por medio de un operador unitario U, el cual depende exclusivamente de los instantes de tiempos t_1 y t_2.*

$$|\psi'\rangle = U|\psi\rangle. \tag{1.10}$$

El postulado II considera que la evolución temporal del sistema se produce en un salto finito de tiempo, entre los instantes t_1 y t_2. Ya que lo que ocurre en ese intervalo de tiempo no es relevante para nosotros, esta definición es más que suficiente. Sin embargo es posible conocer el estado $|\psi\rangle$ entre t_1 y t_2 resolviendo la ecuación diferencial que gobierna el sistema, esta es la conocida ecuación de Schrödinger:

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle, \tag{1.11}$$

con H el Hamiltoniano del sistema cuántico, un observable que representa la energía total que posee el sistema.

Postulado III *A cualquier observable A se le asocia un operador autoadjunto A en el espacio de Hilbert $\mathcal{H}_S$. Los únicos posibles resultados de la medida del observable A son los autovalores del operador A. Escribiendo la ecuación de autovalores para el operador A,*

$$A|j\rangle = a_j|j\rangle, \tag{1.12}$$

donde $|j\rangle$ es una base ortonormal de autoestados del operador A y a_j sus respectivos autovalores. Si podemos expresar la función de onda $|\psi(t)\rangle$ sobre esta base

$$|\psi(t)\rangle = \sum_j c_j(t)|j\rangle, \tag{1.13}$$

con $c_j(t)$ ciertos coeficientes complejos, entonces la probabilidad de que al medir el observable A en el instante t el resultado sea a_j es

$$p_j(t) = p(a = a_j|t) = |\langle j|\psi(t)\rangle|^2 = |c_j(t)|^2. \tag{1.14}$$

Un observable es cualquier propiedad del estado de un sistema que pueda ser determinada u "observada" por procedimientos físicos, al cual se le asocia un operador autoadjunto.

Postulado IV *Si un sistema está descrito por la función de onda $|\psi\rangle$ y medimos el observable A, obteniendo el resultado a_n, entonces inmediatamente después de la medida el estado del sistema viene dado por*

$$\frac{P_n|\psi\rangle}{\sqrt{\langle\psi|P_n|\psi\rangle}}, \tag{1.15}$$

donde P_n es el operador proyector sobre el subespacio asociado a a_n.

El postulado IV describe el colapso de la función de onda cuando esta es observada. Esto implica que tras una medición, el estado del sistema ya está completamente determinado, y si volvemos a medirlo, antes de que evolucione en el tiempo, obtendremos el mismo valor.

1.2 Qubit

Para comenzar a estudiar la computación cuántica, primero vamos hacer una pequeña introducción a su elemento más básico. En computación clásica, el elemento fundamental se conoce tradicionalmente como *bit*, que es la unidad mínima de información. El estado de un bit está completamente definido y puede tomar dos valores: 0 o 1. En computación cuántica esta unidad mínima se denomina *bit cuántico* o *qubit* [1, 10]. Un qubit puede ser cualquier sistema de dos niveles que reside en un espacio de Hilbert de dimensión dos. Los dos estados posibles de un qubit son $|0\rangle$ y $|1\rangle$, análogos al 0 y 1 en computación clásica. Para que estos estados formen una base de $\mathbb{C}^2$, podríamos coger dos vectores ortonormales entre sí:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.16}$$

Donde empiezan a aparecer los fenómenos cuánticos del qubit es en el hecho de que $|0\rangle$ y $|1\rangle$ no son los únicos estados posibles, un qubit puede estar en todo el rango continuo de estados comprendidos entre $|0\rangle$ y $|1\rangle$, es decir, en una superposición de ambos estados, esto es

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.17}$$

donde α y β son números complejos denominados amplitudes. A su vez, $|\alpha|^2$ y $|\beta|^2$ determinan las probabilidades de que al medir el qubit, este se encuentre en el estado $|0\rangle$ o $|1\rangle$ respectivamente. Como tiene que cumplir la condición de normalización para que la suma total de probabilidades sea 1, entonces

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.18}$$

1.2.1 Esfera de Bloch

Como veremos en breve, el conjunto de todos los estados posibles de un qubit corresponde a la superficie de una esfera de radio unidad en el espacio tridimensional $\mathbb{R}^3$. Esta esfera se conoce como *esfera de Bloch*, en honor al físico Felix Bloch, y nos facilita la visualización de los estados de un qubit.

Partiendo de que α y β son números complejos, pueden ser expresados usando la notación de Euler: $\alpha = r_\alpha e^{i\varphi_\alpha}$, $\beta = r_\beta e^{i\varphi_\beta}$, entonces la expresión (1.17) resulta

$$|\psi\rangle = r_\alpha e^{i\varphi_\alpha}|0\rangle + r_\beta e^{i\varphi_\beta}|1\rangle. \tag{1.19}$$

Dado que a efectos observables, las probabilidades de medir un cierto estado son $|\alpha|^2$ y $|\beta|^2$, $|\psi\rangle$ se puede multiplicar por un factor cualquiera $e^{i\gamma}$ (fase global), ya que estas probabilidades no van a cambiar.

Demostración.

$$|\alpha e^{i\gamma}|^2 = (\alpha e^{i\gamma})^*(\alpha e^{i\gamma}) = \alpha^* e^{-i\gamma}\alpha e^{i\gamma} = \alpha^*\alpha = |\alpha|^2. \tag{1.20}$$

■

Entonces, si multiplicamos (1.19) por $e^{-i\varphi_\alpha}$ no afectaría al resultado:

$$|\psi\rangle = r_\alpha|0\rangle + r_\beta e^{i\varphi}|1\rangle, \tag{1.21}$$

donde $\varphi = \varphi_\beta - \varphi_\alpha$. Hemos pasado de tener cuatro parámetros a tres. Podemos expresar $r_\beta e^{i\varphi}$ en coordenadas cartesianas:

$$|\psi\rangle = r_\alpha|0\rangle + (x + iy)|1\rangle. \tag{1.22}$$

Dado que $|\psi\rangle$ tiene que cumplir la condición de normalización ($\langle\psi|\psi\rangle = 1$),

$$r_\alpha^2 + |x + iy|^2 = 1 \longrightarrow r_\alpha^2 + x^2 + y^2 = 1. \tag{1.23}$$

Esto es la ecuación de una esfera de radio unidad en el espacio tridimensional $\mathbb{R}^3$, lo que sugiere una representación en coordenadas esféricas. Teniendo en cuenta que el cambio de coordenadas cartesianas a esféricas es

$$x = \sin(\theta)\cos(\varphi), \quad y = \sin(\theta)\sin(\varphi), \quad z = \cos(\theta), \tag{1.24}$$

obtenemos por tanto que

$$\begin{aligned} |\psi\rangle &= \cos(\theta)|0\rangle + [\sin(\theta)\cos(\varphi) + i\sin(\theta)\sin(\varphi)]|1\rangle \\ &= \cos(\theta)|0\rangle + \sin(\theta)e^{i\varphi}|1\rangle. \end{aligned} \tag{1.25}$$

Observamos que para $\theta = 0$ el estado $|\psi\rangle$ es el $|0\rangle$, y para $\theta = \frac{\pi}{2}$ el estado es $e^{i\varphi}|1\rangle$, lo que hace que sólo nos haga falta los valores $0 \leq \theta \leq \frac{\pi}{2}$ para obtener todos los estados posibles de superposición. Es más, si tomamos un estado $|\psi'\rangle$ en el lado opuesto de la esfera con coordenadas $(1, \pi - \theta, \varphi + \pi)$, tenemos que $|\psi'\rangle = -|\psi\rangle$.

Demostración.

$$\begin{aligned}|\psi'\rangle &= \cos(\pi-\theta)|0\rangle + \sin(\pi-\theta)e^{i(\varphi+\pi)}|1\rangle \\ &= -\cos(\theta)|0\rangle + \sin(\theta)e^{i(\varphi+\pi)}|1\rangle \\ &= -\cos(\theta)|0\rangle - \sin(\theta)e^{i\varphi}|1\rangle \\ &= -|\psi\rangle \end{aligned} \tag{1.26}$$

■

Por tanto observamos que las semiesferas superior e inferior son iguales pero con signo opuesto. Es por esto por lo que reemplazamos la expresión (1.25) por

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + \sin\left(\frac{\theta}{2}\right)e^{i\varphi}|1\rangle, \tag{1.27}$$

o lo que es lo mismo,

$$|\psi\rangle = \left(\cos\left(\frac{\theta}{2}\right),\ \sin\left(\frac{\theta}{2}\right)e^{i\varphi}\right), \tag{1.28}$$

con $\theta \in [0,\, \pi]$ y $\varphi \in [0,\, 2\pi]$. Si ahora calculamos un estado ortogonal a este, sabiendo que $\langle\psi|\psi_\perp\rangle = 0$,

$$|\psi_\perp\rangle = \left(-\sin\left(\frac{\theta}{2}\right),\ \cos\left(\frac{\theta}{2}\right)e^{i\varphi}\right), \tag{1.29}$$

obtenemos un par de vectores ortonormales, conocidos como *estados de Bloch*. Como veremos más adelante, los estados $|\psi\rangle$ y $|\psi_\perp\rangle$ corresponden, respectivamente, con los autoestados asociados a los autovalores $+1/2$ y $-1/2$ de la proyección S_n del operador de espín.

La figura 1.1 muestra una representación gráfica de la esfera de Bloch de un qubit.

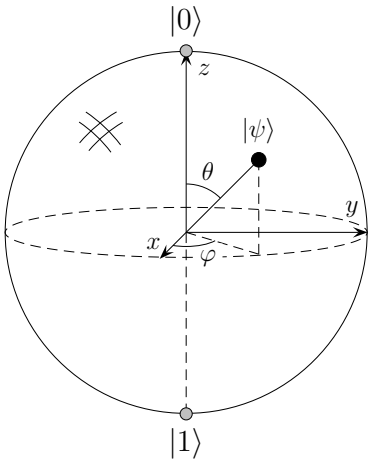


Figura 1.1 Representación gráfica de la esfera de Bloch de un qubit.

1.2.2 El espín

El *espín* es una propiedad física de las partículas elementales, al igual que la carga eléctrica o la masa. Se dice que las partículas tienen un momento angular intrínseco asociado de valor fijo, pero en oposición a la mecánica clásica este momento angular no se define como una rotación sobre su eje, más bien es un término abstracto que añade un grado de libertad cuántico. Sin embargo, para una mayor compresión, se puede visualizar como una rotación propiamente dicha de la partícula sobre sí misma.

El valor que puede tomar el espín está cuantizado, por ejemplo, para fermiones de espín $1/2$ este puede tomar los valores $+\hbar/2$ o bien $-\hbar/2$, es por ello por lo que resulta interesante como sistema físico en computación cuántica, por los que se usa el estado $|0\rangle$ para representar la proyección de espín $+\hbar/2$ y $|1\rangle$ para la proyección de espín $-\hbar/2$.

El vector de espín se define como $\vec{S} = (S_x, S_y, S_z)$, donde S_x, S_y y S_z son las proyecciones del observable espín sobre los ejes x, y y z. Estas proyecciones se definen como:

$$S_x = \frac{\hbar}{2}\sigma_x, \quad S_y = \frac{\hbar}{2}\sigma_y, \quad S_z = \frac{\hbar}{2}\sigma_z, \tag{1.30}$$

siendo σ_x, σ_y y σ_z las matrices de Pauli:

$$\sigma_x \equiv X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.31}$$

Usualmente, para ahorrar en notación, se suele usar unidades naturales, esto es $\hbar = 1$, tomando $\hbar$ este valor en los lugares donde aparece. El espín se puede proyectar sobre un vector **unitario** $\vec{n}$ de componentes (n_x, n_y, n_z):

$$S_n = \vec{S} \cdot \vec{n} = S_x n_x + S_y n_y + S_z n_z. \tag{1.32}$$

Si expresamos $\vec{n}$ en coordenadas esféricas, tenemos que

$$\vec{n} = (\sin(\theta)\cos(\varphi),\ \sin(\theta)\sin(\varphi),\ \cos(\theta)), \tag{1.33}$$

y por tanto:

$$\begin{aligned} S_n &= \frac{1}{2}\left[\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\sin(\theta)\cos(\varphi) + \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\sin(\theta)\sin(\varphi) + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\cos(\theta)\right] \\ &= \frac{1}{2}\begin{pmatrix} \cos(\theta) & \sin(\theta)\cos(\varphi) - i\sin(\theta)\sin(\varphi) \\ \sin(\theta)\cos(\varphi) + i\sin(\theta)\sin(\varphi) & -\cos(\theta) \end{pmatrix} \\ &= \frac{1}{2}\begin{pmatrix} \cos(\theta) & \sin(\theta)e^{-i\varphi} \\ \sin(\theta)e^{i\varphi} & -\cos(\theta) \end{pmatrix}. \end{aligned} \tag{1.34}$$

Según el Postulado III, los posibles resultados de la medida de un observable son los autovalores del operador asociado. Si calculamos los autovalores de S_n mediante el polinomio característico,

$$\det|S_n - \lambda I| = 0, \tag{1.35}$$

obtenemos que son $+1/2$ y $-1/2$, como era de esperar, ya que son los dos únicos valores posibles de proyección del espín. Los autoestados asociados a dichos autovalores se calculan resolviendo el siguiente sistema de ecuaciones:

$$(S_n - \lambda I)v = 0, \tag{1.36}$$

con $v = (v_x,\, v_y)$, nuestra incógnita. Dado que $\det|S_n - \lambda I| = 0$, sólo tenemos una ecuación linealmente independiente:

$$\frac{\sin(\theta)e^{i\varphi}}{2}v_x - \left(\frac{\cos(\theta)}{2} + \lambda\right)v_y = 0, \tag{1.37}$$

$$v_x = \frac{\cos(\theta) + 2\lambda}{\sin(\theta)e^{i\varphi}}v_y. \tag{1.38}$$

Usando la parametrización $v_y = \sin(\theta)e^{i\varphi}$, obtenemos $v_x = \cos(\theta) + 2\lambda$, entonces los autoestados asociados a los autovalores $+1/2$ y $-1/2$ son, respectivamente

$$|v_1\rangle = \begin{pmatrix} \cos(\theta) + 1 \\ \\ \sin(\theta)e^{i\varphi} \end{pmatrix}, \quad |v_2\rangle = \begin{pmatrix} \cos(\theta) - 1 \\ \\ \sin(\theta)e^{i\varphi} \end{pmatrix}. \tag{1.39}$$

Debido a que por definición todo estado tiene que estar normalizado, hay que dividirlos por su módulo $|v| = \sqrt{\langle v|v\rangle}$. El módulo de $|v_1\rangle$ es

$$\begin{aligned} |v_1| &= \sqrt{(\cos(\theta) + 1)^2 + \sin^2(\theta)} \\ &= \sqrt{\cos^2(\theta) + 2\cos(\theta) + 1 + \sin^2(\theta)} \\ &= \sqrt{2\cdot(1 + \cos(\theta))} = \sqrt{2\cdot 2\cdot \underbrace{\frac{(1 + \cos(\theta))}{2}}_{\cos^2(\theta/2)}} \\ &= 2\cos\left(\frac{\theta}{2}\right). \end{aligned} \tag{1.40}$$

De forma análoga se procede para calcular $|v_2|$, obteniéndose en este caso $|v_2| = 2\sin\left(\frac{\theta}{2}\right)$. Por tanto, los autoestados normalizados $|v_1\rangle$ y $|v_2\rangle$ nos quedan de la siguiente forma:

$$|v_1\rangle = \frac{1}{2\cos\left(\frac{\theta}{2}\right)}\begin{pmatrix} \cos(\theta) + 1 \\ \\ \sin(\theta)e^{i\varphi} \end{pmatrix}, \quad |v_2\rangle = \frac{1}{2\sin\left(\frac{\theta}{2}\right)}\begin{pmatrix} \cos(\theta) - 1 \\ \\ \sin(\theta)e^{i\varphi} \end{pmatrix}. \tag{1.41}$$

Si aplicamos las identidades trigonométricas

$$\cos(\theta) = 2\cos^2\left(\frac{\theta}{2}\right) - 1, \tag{1.42}$$

$$\sin\left(\frac{\theta}{2} + \frac{\theta}{2}\right) = 2\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right), \tag{1.43}$$

llegamos a que

$$|v_1\rangle = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \\ \sin\left(\frac{\theta}{2}\right)e^{i\varphi} \end{pmatrix}, \quad |v_2\rangle = \begin{pmatrix} -\sin\left(\frac{\theta}{2}\right) \\ \\ \cos\left(\frac{\theta}{2}\right)e^{i\varphi} \end{pmatrix}. \tag{1.44}$$

Nótese que coinciden con los estados de Bloch (1.28) y (1.29).

1.3 Producto tensorial

Por el momento solo hemos estudiado el qubit en sí como unidad, separado del resto del universo. El *producto tensorial* no es más que una herramienta matemática para poder manejar sistemas cuánticos con más de un constituyente, con el cual poder tratar sistemas de dos qubits o más [10]. Para estudiar sistemas compuestos es necesario introducir este concepto de producto tensorial, cuyas propiedades se revisan a continuación.

1.3.1 Producto tensorial de vectores

Dado dos espacios de Hilbert V y W de dimensión m y n respectivamente, donde residen dos vectores cualesquiera $|v\rangle$ y $|w\rangle$, se define el producto tensorial de V y W como $V \otimes W$. Este nuevo espacio es de dimensión mn y el vector dado por $|v\rangle \otimes |w\rangle$ pertenece a dicho espacio. Comúnmente el producto tensorial $|v\rangle \otimes |w\rangle$ se suele abreviar por $|v\rangle|w\rangle$, $|v, w\rangle$ o incluso por $|vw\rangle$.

Algunas propiedades básicas de este producto son:

1. Para un escalar arbitrario z y dos vectores $|v\rangle$ de V y $|w\rangle$ de W,

$$z(|v\rangle \otimes |w\rangle) = z(|v\rangle) \otimes |w\rangle = |v\rangle \otimes z(|w\rangle). \tag{1.45}$$

2. Para dos vectores cualesquiera $|v_1\rangle$ y $|v_2\rangle$ pertenecientes a V y otro $|w\rangle$ perteneciente a W,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle. \tag{1.46}$$

3. Para dos vectores cualesquiera $|w_1\rangle$ y $|w_2\rangle$ pertenecientes a W y otro $|v\rangle$ perteneciente a V,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle. \tag{1.47}$$

Genéricamente, para un sistema de n componentes $|v_1\rangle$, $|v_2\rangle$ $\ldots$ $|v_n\rangle$ residiendo en V_1, V_2 $\ldots$ V_n, su producto tensorial viene dado por

$$|v\rangle = |v_1\rangle \otimes |v_2\rangle \otimes \cdots \otimes |v_n\rangle = \bigotimes_{i=1}^{n} |v_i\rangle, \tag{1.48}$$

donde $|v\rangle$ es un vector del espacio producto $V = V_1 \otimes V_2 \otimes \cdots \otimes V_n$.

1.3.2 Producto tensorial de operadores lineales

El producto tensorial también se define para operadores lineales. Esto es útil para cuando queramos aplicar operadores definidos para un qubit en sistemas de múltiples qubits. Esto es, si dado dos operadores lineales A y B definidos en V y W respectivamente, y $|v\rangle$ y $|w\rangle$ dos vectores pertenecientes de dichos espacios, se define el operador lineal $A \otimes B$ actuando sobre $V \otimes W$ como

$$(A \otimes B)(|v\rangle \otimes |w\rangle) \equiv A|v\rangle \otimes B|w\rangle. \tag{1.49}$$

Si ahora tomamos una base de V, $\{|v_1\rangle, |v_2\rangle \ \ldots \ |v_n\rangle\}$, y otra de W, $\{|w_1\rangle, |w_2\rangle \ \ldots \ |w_m\rangle\}$, el conjunto de vectores resultado de multiplicar cada elemento V por cada elemento de W genera el espacio producto $V \otimes W$, es decir, forma una base vectorial de tal espacio.

Aplicando esto, la aplicación del operador lineal $A \otimes B$ puede ser extendida a todos los elementos de $V \otimes W$:

$$(A \otimes B)\left(\sum_i a_i |v_i\rangle \otimes |w_i\rangle\right) = \sum_i a_i A|v_i\rangle \otimes B|w_i\rangle. \tag{1.50}$$

1.4 Entrelazamiento cuántico

El fenómeno cuántico más espectacular pero a la vez el más contraintuitivo es el entrelazamiento [1]. En varias partículas entrelazadas, el simple hecho de hacer una medida en una implica que el estado del resto colapsa instantáneamente, sin importar las distancias que las separen. Es por ello por lo que Einstein llamó a este fenómeno *"espeluznante acción a distancia"*.

Formalmente se define de la siguiente forma:

Definición 1.4.1 *Dado un sistema S compuesto de n qubits, su espacio de Hilbert asociado $\mathcal{H}$ es el producto tensorial de los espacios de Hilbert $\mathcal{H}_i$ asociados a cada uno de los componentes i del sistema S,*

$$\mathcal{H} = \bigotimes_{i=1}^{n} \mathcal{H}_i. \tag{1.51}$$

Se dice que S se encuentra en un estado entrelazado si este no es posible expresarlo como producto de los estados de sus componentes, es decir, si no es factorizable.

Para poder entender con claridad este concepto vayámonos al caso más sencillo de todos, el de un sistema sólo formado por dos qubits. El espacio de Hilbert donde reside dicho sistema es

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2. \tag{1.52}$$

Si $\{|0\rangle_1, |1\rangle_1\}$ y $\{|0\rangle_2, |1\rangle_2\}$ forman una base de $\mathcal{H}_1$ y $\mathcal{H}_2$ respectivamente, entonces una base que genera $\mathcal{H}$ es

$$\{|0\rangle_1 \otimes |0\rangle_2,\ |0\rangle_1 \otimes |1\rangle_2,\ |1\rangle_1 \otimes |0\rangle_2,\ |1\rangle_1 \otimes |1\rangle_2\}. \tag{1.53}$$

Por motivos de simplicidad, el producto tensorial $|a_1\rangle_1 \otimes |a_2\rangle_2 \otimes \cdots \otimes |a_n\rangle_n$ se suele denotar como $|a_1 a_2 \cdots a_n\rangle$, por lo que la expresión anterior resulta

$$\{|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle\}. \tag{1.54}$$

El estado $|\psi\rangle$ del sistema S vendrá dado por

$$|\psi\rangle = \alpha_1|00\rangle + \alpha_2|01\rangle + \alpha_3|10\rangle + \alpha_4|11\rangle. \tag{1.55}$$

con $\sum_i |\alpha_i|^2 = 1$. Si el sistema no está entrelazado, su estado se puede expresar de la siguiente forma:

$$\begin{aligned} |\psi\rangle &= (c_1|0\rangle + c_2|1\rangle) \otimes (d_1|0\rangle + d_2|1\rangle) \\ &= c_1 d_1|00\rangle + c_1 d_2|01\rangle + c_2 d_1|10\rangle + c_2 d_2|11\rangle. \end{aligned} \tag{1.56}$$

Comparando las ecuaciones (1.55) y (1.56) se tiene el sistema de ecuaciones

$$\begin{aligned}\alpha_1 &= c_1 d_1,\\ \alpha_2 &= c_1 d_2,\\ \alpha_3 &= c_2 d_1,\\ \alpha_4 &= c_2 d_2.\end{aligned} \tag{1.57}$$

Si el estado $|\psi\rangle$ es tal que se verifica cada una de las ecuaciones, entonces el sistema es separable y no estará entrelazado. En caso contrario, se dice que el sistema no es separable, y por consiguiente estará entrelazado.

A continuación vamos a ver un ejemplo de entrelazamiento cuántico, en el cual estudiaremos dos casos. Primero, un sistema de dos qubits no entrelazados, y segundo, otro que sí se encuentra en un estado de entrelazamiento cuántico. Estos dos estados son $|\psi_1\rangle$ y $|\psi_2\rangle$,

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |11\rangle), \qquad |\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{1.58}$$

Para $|\psi_1\rangle$ tenemos que $\alpha_2 = \alpha_4 = 1/\sqrt{2}$ y $\alpha_1 = \alpha_3 = 0$, entonces:

$$\begin{aligned}0 &= c_1 d_1,\\ 1 &= c_1 d_2,\\ 0 &= c_2 d_1,\\ 1 &= c_2 d_2.\end{aligned} \tag{1.59}$$

De la segunda y cuarta ecuación obtenemos $c_1 = d_2 = c_2 = 1$, de la primera $d_1 = 0$ y finalmente se cumple la tercera. Nos queda por tanto:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle, \tag{1.60}$$

luego no está entrelazado. Para $|\psi_2\rangle$ tenemos $\alpha_1 = \alpha_4 = 1$ y $\alpha_2 = \alpha_3 = 0$:

$$\begin{aligned}1 &= c_1 d_1,\\ 0 &= c_1 d_2,\\ 0 &= c_2 d_1,\\ 1 &= c_2 d_2.\end{aligned} \tag{1.61}$$

Observamos que $c_1 = d_1 = c_2 = d_2 = 1$. Si esto es así, las ecuaciones dos y tres no se pueden satisfacer, y tenemos por tanto, que $|\psi_2\rangle$ está entrelazado.

El estado $|\psi_2\rangle$ mostrado en este ejemplo es uno de los cuatro *estados de Bell*, los cuales fundamentan la base de la *teleportación cuántica*. Este concepto es muy importante, ya que posee una característica clave para la computación cuántica. Hablaremos con más profundidad sobre los estados de Bell y teleportación en el capítulo 2.

2 Puertas y circuitos elementales

A classical computation is like a solo voice —one line of pure tones succeeding each other. A quantum computation is like a symphony —many lines of tones interfering with one another.

SETH LLOYD

Hasta ahora hemos tratado de explicar en qué consiste un qubit y cuáles son sus propiedades, pero lo verdaderamente interesante para la computación cuántica es el hecho de manipularlo según requieran nuestras necesidades. Para ello se le hace pasar por determinadas *puertas* con el objetivo de aplicarle una serie de transformaciones. Dichas transformaciones pueden aplicarse a un solo qubit o a varios, en función del tipo de puerta que estemos usando. A continuación veremos cómo se construyen y cuáles son sus características, tanto las de un qubit como las de varios.

2.1 Operaciones sobre un qubit

Dado que los qubits residen en un espacio de Hilbert de dimensión dos, cualquier matriz 2×2 puede actuar como puerta, siempre y cuando se siga cumpliendo la condición de normalización, es decir, debido a que las amplitudes al cuadrado determinan la probabilidad de medir un cierto estado, la suma total de probabilidades tiene que ser 1. Si por ejemplo queremos aplicarle el operador A a un estado cuántico cualquiera $|\psi\rangle$, el resultado de dicha operación es

$$A|\psi\rangle = |\psi'\rangle. \tag{2.1}$$

Para cumplir la condición de normalización se debe satisfacer que $\langle\psi'|\psi'\rangle = 1$. Sabiendo que $\langle\psi'| = \langle\psi|A^\dagger$, entonces

$$\langle\psi'|\psi'\rangle = \langle\psi|A^\dagger A|\psi\rangle = 1. \tag{2.2}$$

Esta expresión solo vale 1 si, y solo si, $A^\dagger A = I$, es decir, si A es un operador unitario. Esta es la única restricción para que un operador pueda actuar sobre un qubit.

Como acabamos de ver, cualquier matriz unitaria 2×2 puede ser utilizada para manipular un solo qubit, pero a continuación se detallará algunas de las puertas más importantes debido a su interés físico.

2.1.1 Puerta NOT

Como bien sabemos, la única operación – no trivial – que se le puede efectuar a un bit es pasarlo de 1 a 0 y de 0 a 1, esto es la negación lógica. En el caso cuántico también existe tal operación que transforma el $|0\rangle$ en $|1\rangle$ y viceversa. Si recordamos la representación vectorial de estos estados,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{2.3}$$

y le aplicamos el operador

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.4}$$

a cada uno de ellos, obtenemos $|1\rangle \rightarrow |0\rangle$ y $|0\rangle \rightarrow |1\rangle$, por tanto el operador X es el operador de negación o la puerta NOT [10]. De forma general, si al estado cuántico $\alpha|0\rangle + \beta|1\rangle$ le aplicamos la puerta NOT tenemos que

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}. \tag{2.5}$$

2.1.2 Puertas Phase-Flip y Hadamard

También existen otras puertas no triviales. Las más importantes son la Z, también conocida como *phase-flip* (debido que introduce una fase de 180º), y la puerta de *Hadamard*, o H. La puerta Z deja a $|0\rangle$ sin modificar, y le cambia el signo a $|1\rangle$, obteniendo $-|1\rangle$. Sus expresiones matriciales vienen dadas por:

$$Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.6}$$

Geométricamente, la puerta Z efectúa una rotación de la esfera de Bloch de 180º alrededor del eje z. Esta rotación se puede ver en la figura 2.1.

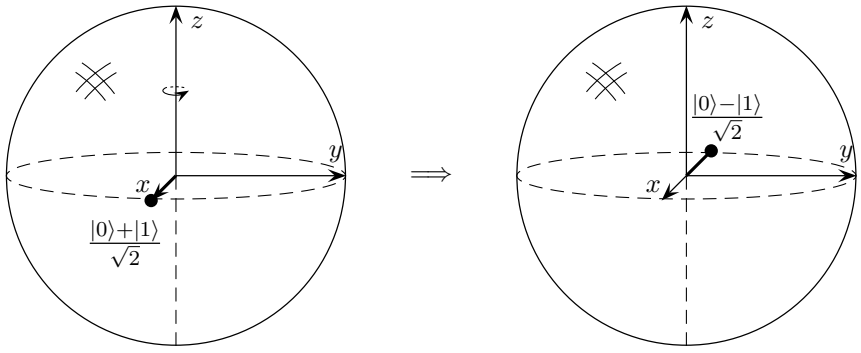


Figura 2.1 Visualización de la puerta Z sobre la esfera de Bloch, actuando sobre el estado $(|0\rangle + |1\rangle)/\sqrt{2}$.

La acción de la puerta de Hadamard también puede visualizarse en la esfera de Bloch como un giro de 90º alrededor del eje y y posteriormente otro giro de 180º alrededor del eje x (o equivalentemente una reflexión a través del plano xy). Estas transformaciones pueden verse en la figura 2.2.

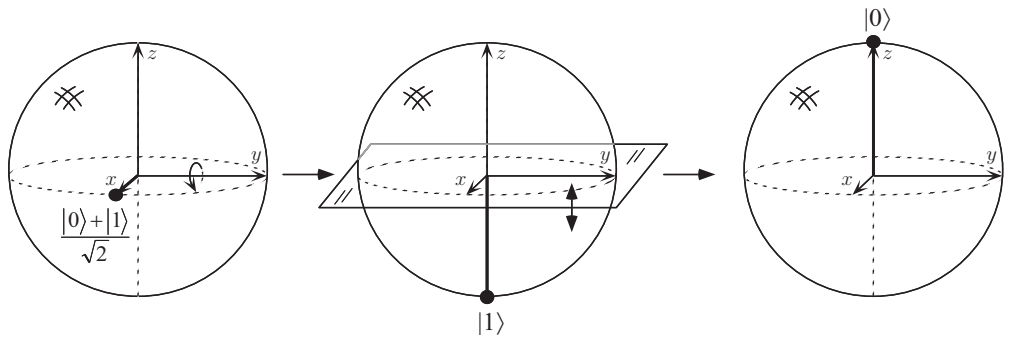


Figura 2.2 Visualización de la puerta de Hadamard sobre la esfera de Bloch, actuando sobre el estado $(|0\rangle + |1\rangle)/\sqrt{2}$.

Por tanto, al aplicar la transformación de Hadamard a $|0\rangle$ se obtiene $(|0\rangle + |1\rangle)/\sqrt{2}$, y al aplicarla a $|1\rangle$, $(|0\rangle - |1\rangle)/\sqrt{2}$. Por conveniencia, estos estados resultantes suelen denotarse respectivamente por

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{y} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.7}$$

los cuales toman vital importancia para la computación cuántica. Cabe destacar la relevancia de la puerta de Hadamar, debido a que partiendo de un estado conocido (ya sea $|0\rangle$ o $|1\rangle$), genera una superposición de estados con probabilidad $1/2$ cada uno.

La figura 2.3 muestra a modo de resumen el diagrama conmutativo de las transformaciones necesarias para obtener cada uno de los estados vistos hasta ahora y cómo se relacionan a través de las puertas estudiadas.

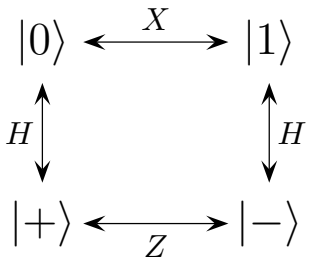


Figura 2.3 Diagrama de transformaciones de un qubit.

2.1.3 Circuito de medida

Otra operación a tener en cuenta es la de medida, que es representada por el símbolo de un medidor, según se muestra en la figura 2.4. Esta codifica un estado cuántico $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ usando un bit de información clásica M, obteniendo un resultado de 0 o 1 con una probabilidad de $|\alpha|^2$ o $|\beta|^2$ respectivamente. Justo después de esta operación la función de onda colapsa hacia $|0\rangle$ si M es 0, o hacia $|1\rangle$ si es 1. El bit clásico se representa mediante una línea doble.

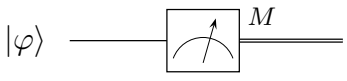


Figura 2.4 Símbolo para la medida cuántica.

2.1.4 Otras puertas

También existen otras transformaciones unitarias que actúan sobre un qubit. Por ejemplo la matriz Y de Pauli o la puerta *phase-shift*, que se definen respectivamente como

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad R_\delta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix}. \tag{2.8}$$

La puerta Y efectúa una rotación de la esfera de Bloch de 180º alrededor del eje z, seguida de otra rotación de 180º alrededor del eje Y, y finalmente introduce una fase global de 90º. Por otro lado, la puerta R_δ deja al vector $|0\rangle$ inmutado e introduce una fase global multiplicando $|1\rangle$ por $e^{i\delta}$. Observamos que R_δ se trata de una generalización de la puerta Z, ya que esta última hace una rotación de π radianes alrededor del eje z y R_δ hace una rotación de δ radianes también alrededor del eje z, es decir, $Z = R_\pi$.

También es conveniente definir las operaciones de rotación alrededor de los ejes y y z:

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \quad R_z(\alpha) = \begin{pmatrix} e^{i\alpha/2} & 0 \\ 0 & e^{-i\alpha/2} \end{pmatrix}, \tag{2.9}$$

las cuales efectúan una rotación de θ y α alrededor de los ejes y y z respectivamente. Para más información se recomienda consultar [1] y [10].

2.2 Operaciones sobre múltiples qubits

2.2.1 Puerta CNOT

La puerta de dos qubits más básica de todas es la NOT controlada o CNOT [10]. El primer qubit de esta puerta se denomina *'control qubit'*, ya que como veremos ahora, controla la salida del segundo qubit, denominado *'target qubit'*. El control qubit permanece inmutable a lo largo de su paso por la puerta CNOT mientras que el target qubit cambia de estado si el control qubit es $|1\rangle$ y permanece igual si es $|0\rangle$. En la tabla 2.1 se muestra la tabla de verdad de esta puerta.

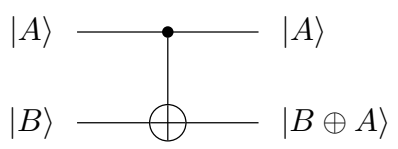
Tabla 2.1 Tabla de verdad de la puerta CNOT.

Control qubit	Target qubit	Output
$\|0\rangle$	$\|0\rangle$	$\|00\rangle$
$\|0\rangle$	$\|1\rangle$	$\|01\rangle$
$\|1\rangle$	$\|0\rangle$	$\|11\rangle$
$\|1\rangle$	$\|1\rangle$	$\|10\rangle$

La puerta CNOT puede verse como una generalización de la puerta XOR clásica, ya que lo que hace es $|A, B\rangle \to |A, B \oplus A\rangle$, donde $\oplus$ denota suma de módulo dos, que es exactamente lo que hace la puerta XOR. Otra forma de verla es mediante su representación matricial, donde definiendo la base de estados ortonormal del espacio de Hilbert producto

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{2.10}$$

existe una matriz unitaria U_{CN}, tal que aplicándola a dichos estados, se obtiene $U_{CN}|A, B\rangle = |A, B \oplus A\rangle$.

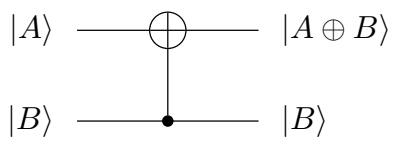


$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figura 2.5 Puerta CNOT. Símbolo en circuito (izquierda) y representación matricial (derecha) en $\mathbb{C}^4$. La línea superior representa el control qubit y la inferior el target qubit.

Una de las propiedades más llamativas de esta puerta es su capacidad para entrelazar dos qubits, cosa que aplicaremos más adelante para diseñar circuitos cuánticos.

Otra variante es como se muestra en la figura 2.6. Aquí se han intercambiado los control y target qubits, o sea, se ha volteado la puerta. Ahora A se encuentra en el target qubit y B en el control qubit.



$$\tilde{U}_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Figura 2.6 Puerta CNOT volteada. Símbolo en circuito (izquierda) y representación matricial (derecha) en $\mathbb{C}^4$.

2.2.2 Puerta SWAP

Una combinación de tres puertas CNOT da la puerta SWAP, tal y como se puede ver en la figura 2.7. Lo que se consigue con esto es intercambiar el estado del primer qubit con el del segundo qubit – de ahí el nombre *swap* en inglés.

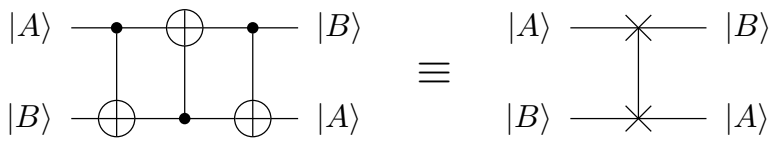


Figura 2.7 Puerta SWAP.

Para analizar su comportamiento, tomamos dos estados cualesquiera $|A\rangle$ y $|B\rangle$ expresados en la base computacional $|0\rangle$ y $|1\rangle$ y los hacemos pasar por la puerta SWAP. Entonces,

obtenemos la siguiente secuencia:

$$\begin{aligned} |A,B\rangle &\xrightarrow{U_{CN}} |A, B\oplus A\rangle \\ &\xrightarrow{\tilde{U}_{CN}} |A\oplus(B\oplus A), B\oplus A\rangle = |B, B\oplus A\rangle \\ &\xrightarrow{U_{CN}} |B, (B\oplus A)\oplus B\rangle = |B, A\rangle. \end{aligned} \tag{2.11}$$

Podemos comprobar que efectivamente permuta los estados de ambos qubits.

2.2.3 Puerta universal de dos qubits (U controlada)

En este apartado veremos que cualquier operación en el espacio de Hilbert de dos qubits puede ser descompuesta en puertas de un único qubit y la puerta CNOT.

La operación U controlada se define como, dado el operador unitario U, este se aplica al target qubit si el control qubit está establecido a 1,

$$|p\rangle|q\rangle \longrightarrow |p\rangle U^p|q\rangle. \tag{2.12}$$

Dado que una matriz U es unitaria si, y solo si, sus filas y sus columnas son ortonormales entre sí, resulta que cualquier matriz unitaria 2×2 puede ser escrita como

$$U = \begin{pmatrix} e^{i(\delta+\alpha/2+\beta/2)}\cos\theta/2 & e^{i(\delta+\alpha/2-\beta/2)}\sin\theta/2 \\ -e^{i(\delta-\alpha/2+\beta/2)}\sin\theta/2 & e^{i(\delta-\alpha/2-\beta/2)}\cos\theta/2 \end{pmatrix}, \tag{2.13}$$

donde δ, α, β y θ son parámetros reales. Por lo tanto, es posible descomponer U de la siguiente forma:

$$U = R_z(\alpha)R_y(\theta)R_z(\beta). \tag{2.14}$$

De hecho, para cualquier U expresado como en la ecuación (2.13), existen tres matrices unitarias A, B y C

$$A = R_z(\alpha)R_y\left(\frac{\theta}{2}\right), \tag{2.15a}$$

$$B = R_y\left(-\frac{\theta}{2}\right)R_z\left(-\frac{\alpha+\beta}{2}\right), \tag{2.15b}$$

$$C = R_z\left(\frac{\beta-\alpha}{2}\right), \tag{2.15c}$$

tales que

$$ABC = I \qquad \text{y} \qquad AXBXC = U, \tag{2.16}$$

con X la puerta NOT. Por lo tanto si el control qubit es 0, el target qubit se deja como está. Sin embargo, si el control qubit es 1, al target qubit se le aplica la transformación U. El circuito cuántico para esta operación se muestra en la figura 2.8 [1].

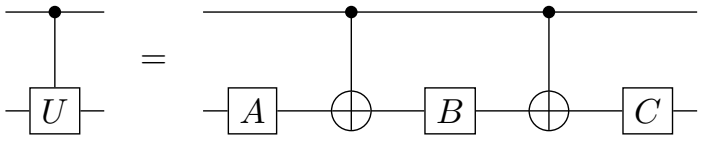


Figura 2.8 Generalización de la puerta U controlada.

2.3 Reversibilidad de un sistema

Antes de seguir vamos a introducir un concepto de gran importancia: la reversibilidad de un sistema. Esta se da cuando en un sistema – ya sea cuántico o clásico – del cual se conoce la operación realizada sobre el mismo y el estado final que produce dicha operación, es posible determinar el estado de partida. Normalmente los sistemas clásicos de computación son irreversibles. Por ejemplo tomemos la puerta AND. Si a un cierto estado inicial lo hacemos pasar por una puerta AND y obtenemos 1 en la salida decimos que ambas entradas de la puerta son 1. En este caso concreto podríamos recuperar el estado inicial, pero tomemos ahora que la salida es 0, en este caso las entradas podrían ser 00, 01 o 10, lo cual no queda completamente determinada. Lo mismo suceden con las demás puertas como la OR o la NAND.

Sin embargo los sistemas cuánticos **cerrados** siempre son reversibles. Que sea cerrado implica que no exista contacto alguno con otro sistema, y más aún que no se realice ninguna medida del sistema, ya que este acto implica que la función de onda colapse inmediatamente, y lo hace irreversiblemente, debido a la naturaleza irreversible de este proceso [3].

Supongamos un sistema cerrado que parte inicialmente del estado $|\psi\rangle$, si le aplicamos una transformación unitaria U obtenemos un nuevo estado $|\psi'\rangle$:

$$U|\psi\rangle = |\psi'\rangle. \tag{2.17}$$

Pues bien, si multiplicamos a ambos lados de la igualdad por $U^\dagger$ podemos obtener nuevamente el estado de partida, ya que al ser U unitario, $U^\dagger U = I$:

$$|\psi\rangle = U^\dagger|\psi'\rangle. \tag{2.18}$$

Por tanto, en sistemas cuánticos cerrados, conociendo el estado final que se ha alcanzado en un qubit y la operación que le hemos hecho, podemos conocer el estado de partida de dicho qubit.

2.4 Teorema de no clonación

En computación clásica es posible realizar una copia de un estado x, desconocido a priori, de hecho es lo que hace una puerta XOR, tal y como se ilustra en la figura 2.9. En este circuito básico, se recibe el estado x y un 0, devolviendo el estado inicial y una copia del mismo. Como vimos anteriormente, el equivalente cuántico de la puerta XOR es la CNOT, entonces, ¿qué sucede si es aplicada a un estado desconocido $|\psi\rangle = a|0\rangle + b|1\rangle$ y a otro conocido $|0\rangle$? ¿Realizaríamos una copia de $|\psi\rangle$, es decir, obtendríamos a la salida $(a|0\rangle + b|1\rangle)(a|0\rangle + b|1\rangle)$? La respuesta es que no es posible realizar una copia de un estado arbitrario. Veámoslo con un sencillo ejemplo.

Para que el estado $|\psi\rangle = a|0\rangle + b|1\rangle$ sea clonado, el estado resultante de hacer pasar $|\psi\rangle$ por una puerta CNOT con el target qubit $|0\rangle$ debería ser igual a $|\psi\rangle \otimes |\psi\rangle$. Si hacemos los cálculos,

$$(a|0\rangle + b|1\rangle) \otimes |0\rangle = a|00\rangle + b|10\rangle \xrightarrow{CNOT} a|00\rangle + b|11\rangle. \tag{2.19}$$

Pero, ¿este resultado es lo mismo que $|\psi\rangle \otimes |\psi\rangle$? Veámoslo:

$$\begin{aligned} |\psi\rangle \otimes |\psi\rangle &= (a|0\rangle + b|1\rangle) \otimes (a|0\rangle + b|1\rangle) \\ &= a^2|00\rangle + ab|01\rangle + ab|10\rangle + b^2|11\rangle. \end{aligned} \tag{2.20}$$

Como acabamos de comprobar, si $ab \neq 0$ aparecen los estados $|01\rangle$ y $|10\rangle$, y por tanto $|\psi\rangle \otimes |\psi\rangle \neq a|00\rangle + b|11\rangle$. Si en cambio $ab = 0$ estos ya no aparecen, pero implica que o bien a o b tiene que ser cero, obteniendo un resultado de $|\psi\rangle \otimes |\psi\rangle = b^2|11\rangle$ para $a = 0$ o $|\psi\rangle \otimes |\psi\rangle = a^2|00\rangle$ para $b = 0$, que es bien distinto del resultado que esperábamos. Es por esto que, de forma general, no es posible realizar una clonación de estados.

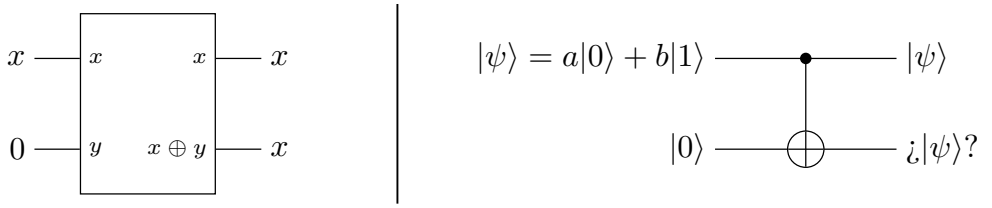


Figura 2.9 Circuitos para copiar un bit usando una puerta XOR (izquierda) y un qubit usando una puerta CNOT(derecha).

Imaginemos por un momento que existe una máquina que partiendo de un estado inicial $|\psi\rangle|s\rangle$[1] es capaz de evolucionar en el tiempo hasta llegar a otro estado $|\psi\rangle|\psi\rangle$. Según enuncia el postulado II debe existir un operador unitario U tal que

$$U|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle. \tag{2.21}$$

Para que esta máquina sea genérica tiene que actuar del mismo modo ante cualquier entrada, de forma que la expresión

$$U|\phi\rangle|s\rangle = |\phi\rangle|\phi\rangle \tag{2.22}$$

también se debería cumplir. Si calculamos el producto interno de las dos expresiones anteriores tenemos que

$$(\langle\psi|\langle s|)\underbrace{U^\dagger U}_{I}(|\phi\rangle|s\rangle) = (\langle\psi|\langle\psi|)(|\phi\rangle|\phi\rangle), \tag{2.23}$$

$$\langle\psi|\phi\rangle\cancelto{1}{\langle s|s\rangle} = \langle\psi|\phi\rangle\langle\psi|\phi\rangle, \tag{2.24}$$

$$\langle\psi|\phi\rangle = |\langle\psi|\phi\rangle|^2. \tag{2.25}$$

Observamos que las dos únicas soluciones de la ecuación anterior son $\langle\psi|\phi\rangle = 0$ o bien $\langle\psi|\phi\rangle = 1$, es decir, como el producto interno de dos vectores indica el coseno del ángulo que forman, se traduce a que las dos únicas soluciones son para tales $|\psi\rangle$ y $|\phi\rangle$ de modo que sean el mismo estado o estados ortogonales entre sí, lo que hace que una función de onda no pueda ser clonada en cualquier situación, sino en estos casos muy concretos. Concluimos por tanto que las leyes de la mecánica cuántica prohíben la existencia de una máquina capaz de clonar estados cuánticos cualesquiera [10, 14].

[1] El estado $|s\rangle$ es un estado genérico normalizado que, a priori, no tiene por qué ser conocido. Se podría usar $|0\rangle$ pero de esta forma se consigue una demostración más genérica.

2.5 Estados de Bell

Los *estados de Bell* o también conocidos como *pares EPR* [1, 10], son estados de máximo entrelazamiento. Aparecieron por primera vez en la sección 1.4 y ahora tenemos las herramientas necesarias para estudiarlos. Son interesante para numerosas aplicaciones que necesiten la propiedad cuántica de entrelazamiento, como por ejemplo el *superdense coding* y el *teletransporte cuántico*, que veremos posteriormente. Los estados de Bell se definen como

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{2.26a}$$

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \tag{2.26b}$$

$$|\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \tag{2.26c}$$

$$|\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \tag{2.26d}$$

los cuales forman una base ortonormal de $\mathbb{C}^4$. Para verificar esto, hay que comprobar que son ortogonales dos a dos y de norma la unidad. Expresándolos en forma vectorial

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}, \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\-1\end{pmatrix}, \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\1\\0\end{pmatrix}, \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}0\\1\\-1\\0\end{pmatrix}, \end{aligned} \tag{2.27}$$

observamos que efectivamente todos tienen norma uno, y si calculamos el producto interno dos a dos:

$$\begin{aligned} &\langle\Phi^+|\Phi^-\rangle = 0, \qquad \langle\Phi^+|\Psi^+\rangle = 0, \qquad \langle\Phi^+|\Psi^-\rangle = 0, \\ &\langle\Phi^-|\Psi^+\rangle = 0, \qquad \langle\Phi^-|\Psi^-\rangle = 0, \qquad \langle\Psi^+|\Psi^-\rangle = 0, \end{aligned} \tag{2.28}$$

concluimos por tanto que forman una base ortonormal.

Los estados de Bell pueden ser obtenidos mediante circuitos cuánticos usando las puertas que vimos anteriormente. El más sencillo de generar es $|\Phi^+\rangle$. Este se construye haciendo pasar el estado $|+\rangle|0\rangle$ por una puerta CNOT. Para ello, partimos del estado $|0\rangle|0\rangle$ y aplicamos la transformación de Hadamard en el primer qubit:

$$|0\rangle|0\rangle \xrightarrow{H\otimes I} H|0\rangle|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = |+\rangle|0\rangle. \tag{2.29}$$

Posteriormente se aplica al estado que acabamos de obtener la puerta CNOT:

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |\Phi^+\rangle. \tag{2.30}$$

En el caso de $|\Phi^-\rangle$ se parte de $|-\rangle|0\rangle$, luego además de la puerta de Hadamard también hay que aplicar la puerta Z en el primer qubit:

$$\begin{aligned} |0\rangle|0\rangle &\xrightarrow{H\otimes I} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = |+\rangle|0\rangle \\ &\xrightarrow{Z\otimes I} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)|0\rangle = |-\rangle|0\rangle \\ &\xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |\Phi^-\rangle. \end{aligned} \tag{2.31}$$

Los dos casos restantes se construyen igual que los anteriores, excepto que ahora el segundo qubit tiene que ser $|1\rangle$ antes de llegar a la puerta CNOT. Por consiguiente, hay que transformarlo con una puerta X. La figura 2.10 muestra los cuatro circuitos empleados para generar cada uno de los estados de Bell.

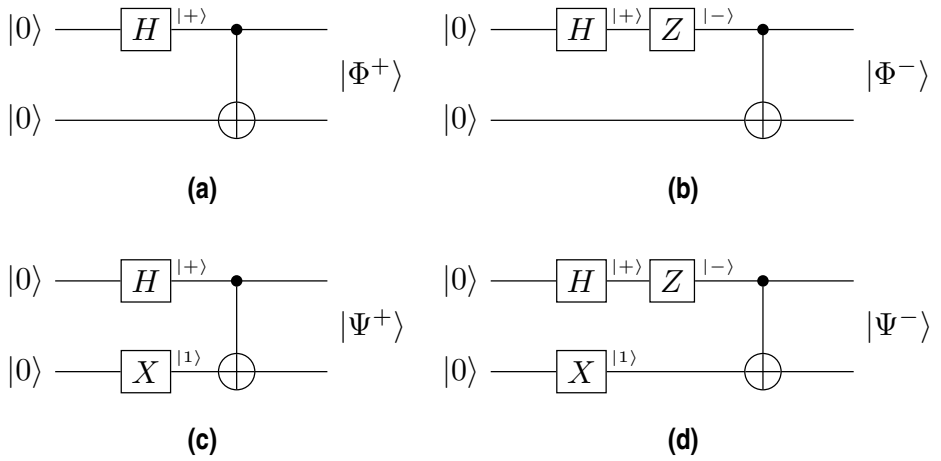


Figura 2.10 Circuitos cuánticos para obtener los estados de Bell.

2.6 Superdense coding y teleportación cuántica

Como veremos ahora, superdense coding y teleportación cuántica son dos caras de una misma moneda. Ambos usan el concepto de entrelazamiento para enviar información de un lugar a otro, existiendo entre emisor y receptor un canal de comunicación cuántico para el caso de superdense coding y clásico para la teleportación. Pasemos a analizar estos dos conceptos en profundidad.

2.6.1 Superdense coding

La idea primordial del superdense coding es el hecho de poder transmitir dos bits clásicos de información codificándolos en un solo qubit. Aunque la mecánica cuántica nos dice que un qubit puede estar en un rango continuo de estados, podríamos pensar que puede almacenar infinitos valores de información, pero el simple hecho de que tengamos que realizar una medición para conocer en qué estado se encuentra el qubit y que nos sea de utilidad, implica que este colapse inmediatamente, imposibilitando el almacenamiento de más de un valor. Teóricamente se comprueba que el número máximo de bits que se puede trasmitir usando un

qubit es uno, pero existe la posibilidad de usar los efectos del entrelazamiento para aumentar este número máximo a dos [6].

Se quiere enviar dos bits de información desde un punto A, hasta otro B. Históricamente se le ha llamado 'Alice' a la persona que quiere enviar la información y 'Bob' a la que la recibe. Para que sea posible enviar los dos bits, inicialmente Alice y Bob deben de compartir un par EPR

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{2.32}$$

generado como hemos visto en la sección anterior, siendo el primer qubit perteneciente a Alice y el segundo a Bob. Alice quiere transmitir $b_1 b_2$ a Bob, para ello, Alice no hará nada si los valores a enviar son 00, aplicará la puerta X si los valores a enviar son 01, aplicará la puerta Z si los valores a enviar son 10 o aplicará la puerta X y después la Z si los valores a enviar son 11. En la tabla 2.2 se puede ver un resumen de todas estas operaciones.

Tabla 2.2 Operaciones a realizar por Alice para codificar los bits en su qubit.

Bit 1	Bit 2	Operación
0	0	I
0	1	X
1	0	Z
1	1	ZX

Por tanto, en cada caso, el estado que se obtendrá será

$$00 : |\psi\rangle \to \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{2.33a}$$

$$01 : |\psi\rangle \to \frac{|10\rangle + |01\rangle}{\sqrt{2}}, \tag{2.33b}$$

$$10 : |\psi\rangle \to \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \tag{2.33c}$$

$$11 : |\psi\rangle \to \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \tag{2.33d}$$

Nótese que, respectivamente, son los estados de Bell $|\Phi^+\rangle$, $|\Psi^+\rangle$, $|\Phi^-\rangle$ y $|\Psi^-\rangle$ vistos anteriormente.

Una vez que Alice ha efectuado las transformaciones correspondientes a su qubit, se lo envía a Bob y este decodifica ambos qubits. Analicemos como se hace esta decodificación: en primer lugar se hace pasar por una puerta CNOT siendo el control qubit el qubit procedente de Alice y el target qubit el de Bob. Tras esto quedaría

$$00 : \frac{|00\rangle + |10\rangle}{\sqrt{2}}, \tag{2.34a}$$

$$01 : \frac{|11\rangle + |01\rangle}{\sqrt{2}}, \tag{2.34b}$$

$$10 : \frac{|00\rangle - |10\rangle}{\sqrt{2}}, \tag{2.34c}$$

$$11 : \frac{|01\rangle - |11\rangle}{\sqrt{2}}. \tag{2.34d}$$

Posteriormente se aplica la transformación de Hadamard al qubit procedente de Alice, obteniendo

$$00 : |00\rangle, \qquad 01 : |01\rangle, \qquad 10 : |10\rangle, \qquad 11 : |11\rangle, \tag{2.35}$$

y al efectuar la medición se consiguen los bits enviados por Alice [2, 10]. Este protocolo se describe en la figura 2.11.

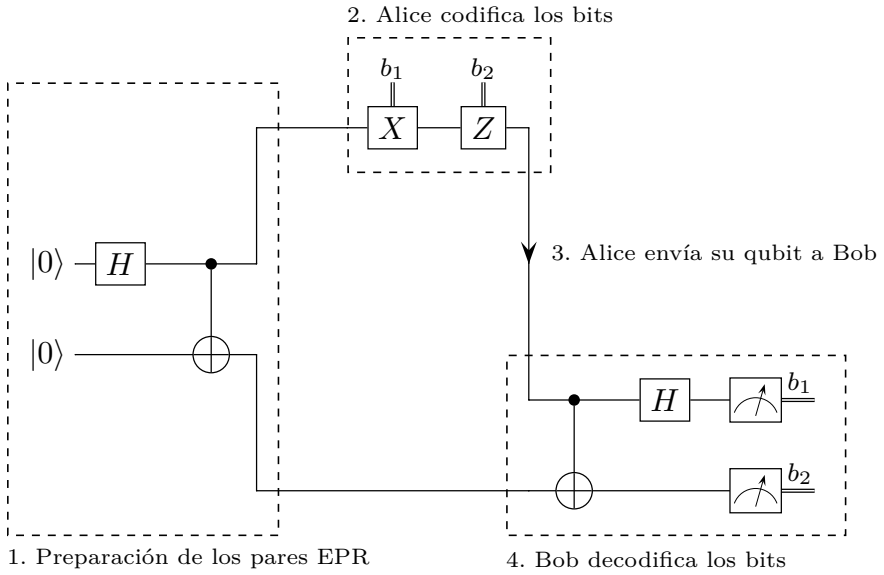


Figura 2.11 Esquema del protocolo superdense coding. El paso número 1 se considera previo y no necesariamente debe realizarlo Alice o Bob, partiendo de la suposición de que inicialmente ya comparten el par EPR.

2.6.2 Teleportación cuántica

Imaginemos que ahora Alice tenga en su posesión un qubit en el estado $|\psi\rangle = a|0\rangle + b|1\rangle$, el cual ha sido preparado durante meses, pero para continuar con su investigación, Alice no dispone de la tecnología necesaria. Por suerte Bob si dispone de ella, pero actualmente su laboratorio se encuentra a kilómetros de distancia. La cuestión es, ¿podrá Alice enviar el estado de su qubit $|\psi\rangle$ hasta Bob para continuar su investigación usando un canal clásico de comunicación? La teleportación cuántica responde a esta pregunta [10].

Realmente lo que se "teletransporta" no es la materia en sí, como muchos podrían pensar, más bien lo que sucede es que el estado cuántico de un qubit viaja hacia otro qubit, de ahí el nombre de "teletransporte cuántico". El esquema de este protocolo se puede ver en la figura 2.12, veamos como funciona.

El único requisito que tienen que cumplir Alice y Bob es que previamente tienen que compartir un par EPR, al igual que en el superdense coding. Inicialmente, el estado de entrada al circuito es

$$|\psi_0\rangle = |\psi\rangle|\Phi^+\rangle \tag{2.36a}$$

$$= \frac{1}{\sqrt{2}}\left[a|0\rangle(|00\rangle + |11\rangle) + b|1\rangle(|00\rangle + |11\rangle)\right], \tag{2.36b}$$

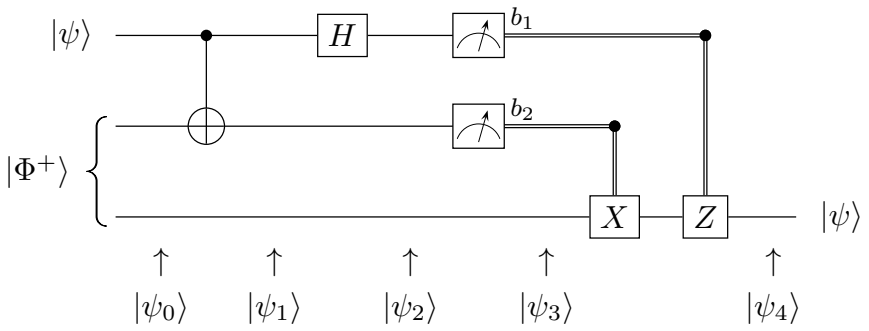


Figura 2.12 Esquema del protocolo para el teletransporte cuántico.

donde se ha tomado que los dos primeros qubits pertenecen a Alice y el tercero a Bob. Al aplicar la puerta CNOT a los dos qubits de Alice obtenemos

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left[a|0\rangle(|00\rangle + |11\rangle) + b|1\rangle(|10\rangle + |01\rangle)\right]. \tag{2.37}$$

Posteriormente Alice manda su qubit a teletransportar hacia una puerta de Hadamard, con lo cual el estado resultante es

$$|\psi_2\rangle = \frac{1}{2}\left[a(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + b(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)\right]. \tag{2.38}$$

Desarrollando la expresión anterior se llega a

$$\begin{aligned} |\psi_2\rangle = \frac{1}{2}[&a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) \\ &+b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)], \end{aligned} \tag{2.39}$$

y reagrupando términos

$$\begin{aligned} |\psi_2\rangle = \frac{1}{2}[&|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) \\ &+|10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)]. \end{aligned} \tag{2.40}$$

Por tanto, si al medir sus dos qubits, Alice obtiene 00, le dirá a Bob que no haga nada, si obtiene 01, le dirá que aplique la puerta X, si obtiene 10, le dirá que aplique la puerta Z, y si obtiene 11, le dirá que aplique X y después Z.

Debido a que se utiliza un canal clásico para transmitir información, este proceso no es instantáneo; como mucho, la velocidad máxima a la que se transmite la información es la velocidad de la luz, es por esto por lo que no se viola la teoría de la relatividad especial. Otro punto importante a tener en cuenta es que también se cumple el teorema de no clonación, porque al fin y al cabo, para poder teletransportar el estado $|\psi\rangle$, Alice ha tenido que medirlo, destruyéndolo inmediatamente.

3 Algoritmo de Shor

If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our email,
Till we get crypto that's quantum, and daunt 'em.

JENNIFER Y PETER SHOR

Una de las aplicaciones de la computación cuántica es el problema de la factorización de un número en factores primos, el cual se convierte en una tarea muy difícil de resolver con los ordenadores convencionales, ya que con el algoritmo más eficiente elaborado hasta la fecha se consigue factorizar en tiempos de orden exponencial [9], el cual para factorizar un cierto entero n lleva un tiempo asintótico de cálculo de $\exp(c(\log n)^{1/3}(\log\log n)^{2/3})$, para cierta constante c, cosa que hace inviable el proceso para números muy grandes. Es por esto por lo que los sistemas de encriptación actuales como el RSA [11] hacen uso de este concepto para generar claves de seguridad multiplicando dos números primos de longitudes considerables, haciendo que sea imposible romperlas. Sin embargo, el algoritmo de Shor promete tiempos de cálculo de orden polinómico, siendo necesario solo del orden de $O((\log n)^2(\log\log n)(\log\log\log n))$ pasos en un ordenador cuántico [13], haciendo posible resolver este problema en períodos de tiempo razonables usando ordenadores cuánticos como veremos en este capítulo.

3.1 Introducción a la aritmética modular

Antes de estudiar en profundidad el tema principal de este capítulo, es importante hacer una breve introducción a la aritmética modular ya que esta juega un papel primordial en el algoritmo de Shor.

Esta herramienta matemática fue introducida por primera vez en 1801 por Carl Friedrich Gauss en su libro *Disquisitiones Arithmeticae*. Esta establece una relación de congruencia entre enteros de tal forma que para un módulo n, se define como:

Definición 3.1.1 *Dados dos enteros a y b, se dice que se encuentran en la misma "clase de congruencia" módulo n, si ambos dejan el mismo resto al ser divididos entre n, o, equivalentemente, si $a - b$ es un múltiplo de n.*

Según la notación de Gauss [5], esta relación se puede expresar como

$$a = b \,(\mathrm{mod}\, n), \tag{3.1}$$

y se lee "a es congruente con b módulo n". Por tanto se establece una secuencia de enteros que se repite tras alcanzar el módulo, de ahí que a veces se llame "aritmética de reloj". Por ejemplo, para $n = 3$ la secuencia sería:

$$\begin{aligned} 1 &= 1 \,(\mathrm{mod}\, 3) \\ 2 &= 2 \,(\mathrm{mod}\, 3) \\ 3 &= 0 \,(\mathrm{mod}\, 3) \\ 4 &= 1 \,(\mathrm{mod}\, 3) \\ 5 &= 2 \,(\mathrm{mod}\, 3) \\ &\vdots \end{aligned}$$

3.2 Encontrar los factores a partir del período

El algoritmo de Shor consta de dos partes bien diferenciadas. La primera de ellas es clásica y puede ejecutarse en un ordenador convencional, mientras que la segunda utiliza elementos de computación cuántica.

Partimos de que tenemos un entero impar[1] N producto de dos primos, p y q, los cuales queremos hallar. En esta primera fase se adapta el problema inicial de forma que sea posible utilizar el procedimiento conocido como *period finding*, el cual permite usar la potencia de la computación cuántica. Esta técnica se emplea para encontrar el período r de una función que se repite en el tiempo, dicha función será en nuestro caso $f(x) = a^x \,(\mathrm{mod}\, N)$, con $a < N$.

El primer paso es generar un entero $a < N$ de forma aleatoria y comprobar que $\mathrm{mcd}(a,\ N) = 1$, donde $\mathrm{mcd}(a,\ b)$ denota *máximo común divisor* y puede ser calculado en tiempo polinómico usando el algoritmo de Euclides [8]. Si esta condición no se cumple quiere decir que a y N tienen un factor común, terminando aquí el algoritmo con valor $\mathrm{mcd}(a,\ N)$. El segundo paso es usar la subrutina cuántica para calcular el período r de la función $f(x) = a^x \,(\mathrm{mod}\, N)$, es decir, el entero más pequeño r para el cual $a^r = 1 \,(\mathrm{mod}\, N)$.

Si en esta expresión restamos 1 a ambos lados obtenemos $a^r - 1 = 0 \,(\mathrm{mod}\, N)$, que es lo mismo que

$$(a^{r/2} + 1)(a^{r/2} - 1) = 0 \,(\mathrm{mod}\, N), \tag{3.2}$$

obteniendo ya dos factores de N. Obviamente para que esto se pueda hacer, r tiene que ser divisible por 2, y ni $(a^{r/2} + 1)$ ni $(a^{r/2} - 1)$ pueden ser $0 \,(\mathrm{mod}\, N)$, ya que obtendríamos la relación $0 = 0 \,(\mathrm{mod}\, N)$ y esto no nos serviría de mucho, es decir, $a^{r/2} \neq \pm 1 \,(\mathrm{mod}\, N)$. En realidad solo es necesario comprobar que $a^{r/2} \neq -1 \,(\mathrm{mod}\, N)$ debido a que como hemos calculado una r tal que $a^r = 1 \,(\mathrm{mod}\, N)$, $a^{r/2}$ nunca va a ser congruente con 1 módulo N. Por tanto, en este punto del algoritmo hay que verificar dos cosas: $r = 0 \,(\mathrm{mod}\, 2)$ – que sea par –, y que $a^{r/2} \neq -1 \,(\mathrm{mod}\, N)$; si alguna de estas condiciones no se cumple hay que volver a generar un nuevo a y repetir el proceso. Finalmente los factores de N son el $\mathrm{mcd}(a^{r/2} \pm 1,\ N)$.

A continuación, antes de estudiar el algoritmo de period finding, vamos a introducir un elemento de vital importancia y del que haremos uso en dicho algoritmo: la transformada cuántica de Fourier.

[1] Si fuese par, una solución trivial al problema sería el número 2, obteniendo ya un factor inmediato.

3.3 Transformada cuántica de Fourier (QFT)

Hasta ahora hemos representado los sistemas de qubits usando la notación $|j_1 j_2 \dots j_n\rangle$. Esto se conoce como representación binaria. Cualquier número puede ser representado en binario, y por tanto ser codificado por un registro de qubits, pero resulta muy tedioso andar trabajando en binario, ya que para números grandes, representa una gran cantidad de ceros y unos. Para que nos sea más cómodo trabajar con registros de qubits, en vez de la representación binaria vamos a usar la decimal. Sea un estado representado mediante notación binaria $|j_1 j_2 \dots j_n\rangle$. Como es conocido, para pasar de notación binaria a decimal se tiene que

$$j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0 = \sum_{\ell=1}^{n} j_\ell 2^{n-\ell}, \tag{3.3}$$

con j_1, $j_2 \dots j_n$ los estados individuales de cada uno de los qubits del registro.

Por tanto, se usará la notación $|j\rangle_n$ para expresar que un estado j se encuentra almacenado en un registro de n qubits. En el caso de que aparezca el estado sin subíndice, significa que se tomará un registro con los qubits necesarios para que dicho estado pueda almacenarse en él. Como mínimo, $n \geq \lceil \log_2 j \rceil$.

Vamos a ver un pequeño ejemplo del uso de esta notación. Tenemos un estado cuántico cualquiera, como por ejemplo $|10010\rangle$. Si aplicamos la ecuación (3.3), obtenemos que

$$j = 1 \cdot 2^{5-1} + 0 \cdot 2^{5-2} + 0 \cdot 2^{5-3} + 1 \cdot 2^{5-4} + 0 \cdot 2^{5-5} = 18. \tag{3.4}$$

Por tanto el estado $|10010\rangle$ es equivalente a $|18\rangle$.

Una vez hecha esta pequeña introducción pasemos a estudiar la transformada cuántica de Fourier. La transformada cuántica de Fourier (en adelante QFT, de sus siglas en inglés *Quantum Fourier Transform*) es un elemento importante en computación cuántica debido a sus propiedades. Esta es una adaptación cuántica de la versión discreta. En esta última se toma un vector complejo $\vec{x} = (x_0, x_1, \dots, x_{N-1})$ de longitud N y lo transforma en otro vector complejo $\vec{y} = (y_0, y_1, \dots, y_{N-1})$ según la expresión

$$y_k \longrightarrow \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} x_j, \tag{3.5}$$

con $\omega = e^{2\pi i/N}$. Por el contrario, la QFT actúa como un operador lineal sobre una base ortonormal $|0\rangle, |1\rangle, \dots, |N-1\rangle$, transformando el estado $|j\rangle$ en el estado

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle. \tag{3.6}$$

De hecho, si tomamos un estado arbitrario $|\alpha\rangle = \alpha_0|0\rangle + \cdots + \alpha_{N-1}|N-1\rangle$ y le aplicamos la QFT, obtenemos un nuevo estado $|\beta\rangle = \beta_0|0\rangle + \cdots + \beta_{N-1}|N-1\rangle$, donde cada elemento β_j es la transformada discreta de α_j [10].

A veces es útil usar la representación matricial de la QFT, ya que al ser un operador unitario, se puede diseñar un circuito cuántico para aplicarlo a un sistema de n qubits. Esta

representación matricial viene dada por una matriz de dimensión $N = 2^n$

$$QFT_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2N-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3N-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2N-2} & \omega^{3N-3} & \cdots & \omega^{(N-1)^2} \end{pmatrix}. \tag{3.7}$$

Como se puede observar, el elemento jk-ésimo de la matriz se define como $\omega^{jk} = e^{2\pi ijk/N}$ (donde se ha omitido el término de normalización).

Cabe destacar que para el caso particular donde $N = 2$ nuestra matriz QFT_2 es

$$QFT_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & \omega \end{pmatrix}. \tag{3.8}$$

Puesto que N vale 2, $\omega = e^{2\pi i/2} = e^{\pi i} = -1$, y entonces QFT_2 nos queda

$$QFT_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3.9}$$

Observamos que es la puerta de Hadamard que estudiamos anteriormente. Esto es así ya que la transformada cuántica de Fourier es una generalización de la puerta de Hadamard, solo se diferencian en que la QFT introduce una fase, que a efectos prácticos de observación, es indistinguible.

3.3.1 Propiedad de invariancia a desplazamientos lineales

Una de las razones por la que la transformada cuántica de Fourier es de gran utilidad en computación cuántica es debido a que es invariante a desplazamientos lineales, es decir, sea la transformación

$$\sum_{j=0}^{N-1} \alpha_j |j\rangle \xrightarrow{QFT} \sum_{k=0}^{N-1} \beta_k |k\rangle, \tag{3.10}$$

supongamos que existe un operador unitario U_d tal que sea capaz de introducir un desplazamiento d en el estado $|j\rangle$

$$U_d|j\rangle = |j + d\rangle. \tag{3.11}$$

El resultado de aplicar la QFT

$$U_d \sum_{j=0}^{N-1} \alpha_j |j\rangle = \sum_{j=0}^{N-1} \alpha_j |j + d\rangle \xrightarrow{QFT} \sum_{k=0}^{N-1} e^{2\pi idk/N} \beta_k |k\rangle, \tag{3.12}$$

tiene la propiedad de que el valor de la amplitud de $|k\rangle$ es el mismo independientemente del valor de d. Esto es así porque el módulo de $e^{2\pi idk/N}$ siempre es uno, y en el proceso de medida, este no afecta al valor obtenido [10].

3.3.2 Diseño de circuito cuántico

Como hemos mencionado anteriormente, la QFT es una transformación unitaria, y entonces se puede construir un circuito cuántico que la implemente de forma eficiente. En este apartado veremos las técnicas empleadas para elaborar dicho circuito cuántico usando las puertas

vistas en el capítulo 2. Partiendo del estado $|j\rangle$, le aplicamos la QFT:

$$|j\rangle \xrightarrow{QFT} |j'\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle. \tag{3.13}$$

Como queremos diseñar un circuito cuántico, tenemos que volver a usar la representación binaria del estado del sistema. Para ello cambiamos el estado $|k\rangle$ por $|k_1 \dots k_n\rangle$ y entonces la suma en k se transforma en n sumas de todos los posibles estados de cada uno de los qubits. Por otro lado, el $k/2^n$ del exponente, recordando la ecuación (3.3), se transforma en

$$\frac{k}{2^n} = \sum_{l=1}^{n} \frac{k_l 2^{n-l}}{2^n} = \sum_{l=1}^{n} \frac{k_l}{2^l}. \tag{3.14}$$

Por tanto, el estado $|j'\rangle$ nos queda

$$|j'\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{l=1}^{n} k_l/2^l\right)} |k_1 \dots k_n\rangle. \tag{3.15}$$

Haciendo algunos cálculos básicos y reagrupando:

$$\begin{aligned} |j'\rangle &= \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi i j k_l/2^l} |k_l\rangle \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l/2^l} |k_l\rangle \right] \\ &= \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j/2^l} |1\rangle \right]. \end{aligned} \tag{3.16}$$

Vayámonos al caso en el que $l = 1$. En este caso la exponencial que multiplica al estado $|1\rangle$ es $e^{2\pi i j/2}$. Si usamos la representación binaria de j vemos que $j = j_1 j_2 \cdots j_n$, al dividirlo entre 2 se desplaza cada uno de los dígitos un lugar hacia la derecha obteniendo $j/2 = j_1 j_2 \cdots j_{n-1} . j_n$[2], lo mismo que ocurre al dividir un número en la base decimal por 10. Por tanto, esta exponencial nos queda

$$e^{2\pi i [j_1 j_2 \cdots j_{n-1} . j_n]} = e^{2\pi i [j_1 j_2 \cdots j_{n-1}]} e^{2\pi i [0 . j_n]} = e^{2\pi i [0 . j_n]}. \tag{3.17}$$

Esto es así ya que $j_1 j_2 \cdots j_{n-1}$ es un número entero y entonces la fase va a ser un múltiplo de 2π, lo que lleva a $e^{2\pi i [j_1 j_2 \cdots j_{n-1}]} = 1$.

Para el resto de valores de l se procede de forma completamente idéntica, salvo que en vez de una vez, los bits se desplazan l veces hacia la derecha. Finalmente la transformada cuántica de Fourier resulta:

$$|j\rangle \to |j'\rangle = \frac{1}{\sqrt{2^n}} \left(|0\rangle + e^{2\pi i 0 . j_n} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i 0 . j_{n-1} j_n} |1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i 0 . j_1 j_2 \cdots j_n} |1\rangle\right). \tag{3.18}$$

[2] En esta expresión, el símbolo "." denota el punto decimal. Por ejemplo, para un estado cualquiera de 5 qubits en el que $j = 01101$, la división de j entre 2 sería de la forma $j/2 = 0110.1$, y entre 4 sería $j/4 = 011.01$.

Una vez en este punto es fácil construir el circuito. Empecemos por el qubit número n:

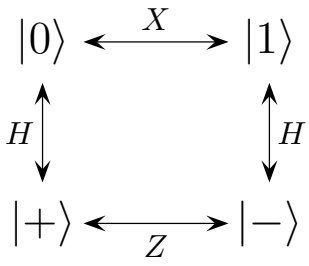
$$|j_1'\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.j_n]}|1\rangle\right), \tag{3.19}$$

esto es lo mismo que

$$|j_1'\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i j_n/2}|1\rangle\right) \tag{3.20}$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i j_n}|1\rangle\right). \tag{3.21}$$

Por tanto tenemos dos casos. En primer lugar, si $j_n = 1$ entonces $e^{\pi i j_n} = -1$, si por el contrario $j_n = 0$ entonces $e^{\pi i j_n} = +1$, es decir, tenemos los estados $|-\rangle$ y $|+\rangle$ respectivamente. Si recordamos la figura 2.3:



Observamos por tanto que esta operación se corresponde con la puerta de Hadamard. Para el qubit número $n-1$ seguimos el mismo procedimiento:

$$|j_2'\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.j_{n-1}j_n]}|1\rangle\right) \tag{3.22}$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i j_{n-1}} e^{\pi i j_n/2}|1\rangle\right). \tag{3.23}$$

Como acabamos de ver, el término $e^{\pi i j_{n-1}}$ corresponde con aplicarle la puerta de Hadamard. Como se observa, el término $e^{\pi i j_n/2}$ se aplica si, y solo si, $j_n = 1$, debido a que si es 0, la exponencial vale 1. Esto es aplicar de forma controlada la puerta de desplazamiento de fase (*phase-shift*)

$$R_{\pi/2} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2} \end{pmatrix}, \tag{3.24}$$

con j_n el qubit de control. De forma general, para el qubit número $n-\ell$ (con $\ell = 0,1,2,\ldots,n-1$) se tiene:

$$|j_{\ell+1}'\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.j_{n-\ell}j_{n-\ell+1}\ldots j_n]}|1\rangle\right) \tag{3.25}$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\pi i j_{n-\ell}} \cdot e^{\frac{\pi}{2} i j_{n-\ell+1}} \cdots e^{\frac{\pi}{2^{\ell-1}} i j_n}|1\rangle\right). \tag{3.26}$$

Por consiguiente, hay que aplicar una puerta de Hadamard seguido de ℓ puertas de desplazamiento de fase controladas, siendo un total de $\ell + 1$ puertas por qubit. Si hacemos la suma para cada uno de ellos:

$$\sum_{\ell=0}^{n-1}(\ell+1) = n + \sum_{\ell=0}^{n-1}\ell = \frac{n(n+1)}{2}, \tag{3.27}$$

obtenemos un total aproximado de $O(n^2)$ puertas cuánticas. En la figura 3.1 se observa una representación gráfica del circuito cuántico. Nótese que al final se han utilizado $\lfloor n/2 \rfloor$ puertas SWAP para alterar el orden de los qubits, ya que al aplicar la QFT se ha invertido.

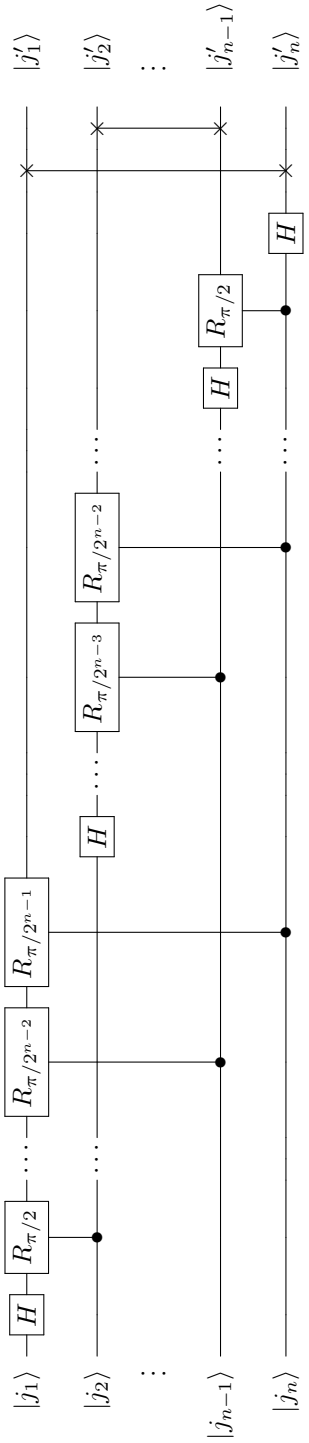


Figura 3.1 Transformada cuántica de Fourier implementada en circuito cuántico.

3.4 Period Finding

Encontrar el período de una función dada es una tarea bastante costosa para un ordenador convencional, ya que a día de hoy no existe ningún algoritmo capaz de resolver este problema de forma eficiente y con tiempos razonables de cómputo para que sea de utilidad [9].

Queremos encontrar el período de la función $f(x) = a^x \pmod N$. Como comentamos anteriormente, este período es el menor entero r para el cual $a^r = 1 \pmod N$. Por ejemplo, para un valor de $a = 2$ y $N = 5$ tenemos que

$$\begin{aligned} 2^1 &= 2 \pmod 5 \\ 2^2 &= 4 \pmod 5 \\ 2^3 &= 3 \pmod 5 \\ 2^4 &= 1 \pmod 5 \\ 2^5 &= 2 \pmod 5 \\ 2^6 &= 4 \pmod 5 \\ &\vdots \end{aligned}$$

Observamos que en este caso el período es 4.

Para comenzar con el algoritmo de Period Finding, dados a y N, el primer paso es encontrar q, una potencia de 2 ($q = 2^n$, con n el número de qubits) tal que $N^2 \leq q < 2N^2$. Esto es así porque necesitamos que la función se repita un número suficiente de veces para minimizar el error cometido. Ahora necesitamos dos registros de qubits. En uno de ellos será donde almacenemos los valores de x que le pasaremos a nuestra función $f(x)$, y en el segundo almacenaremos los valores que devuelve dicha función. El número de qubits que tendrá nuestro sistema será por tanto $n = \lceil \log_2(N^2) \rceil$ en el primer registro y $m = \lceil \log_2(N) \rceil$ en el segundo.

Inicialmente nuestro sistema se encuentra en el estado $|0\rangle|0\rangle$ (para ahorrar espacio, los estados de los registros se expresan en la base decimal en lugar de en binario). A continuación se crea una superposición en el primer registro, aplicando en este la puerta de Hadamard a cada uno de los qubits individualmente

$$|0\rangle|0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|0\rangle. \tag{3.28}$$

Lo que tenemos ahora es una superposición uniforme de todos los valores comprendidos entre 0 y $q - 1$. El siguiente paso consiste en calcular $f(x)$ en el segundo registro. Esto se consigue aplicando la transformación unitaria U_f. Cómo es esta transformación internamente no nos concierne ya que será diferente para cada función a la que queramos calcularle el período, y su diseño se escapa del propósito de este texto. Podemos verlo como una "caja negra" a la que le entra un valor x y sale $f(x)$, como se observa en la figura 3.2. Por tanto, ahora tenemos nuestro sistema en el estado

$$\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|f(x)\rangle. \tag{3.29}$$

Si medimos el segundo registro, este colapsa a un determinado estado $|f(x_0)\rangle$. Entonces el primer registro colapsará a los valores de x tal que $f(x) = f(x_0)$. Como $f(x)$ es periódica de período r estos valores serán x_0, $x_0 + r$, $x_0 + 2r$, $\ldots$, $x_0 + (\frac{q}{r} - 1)r$, dejando el estado

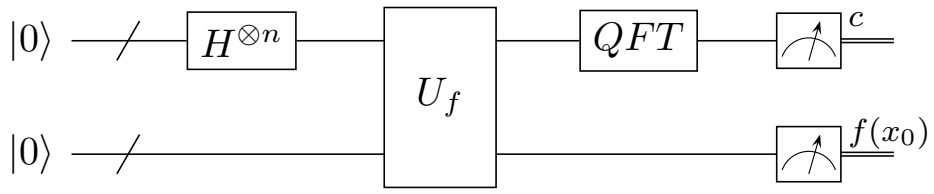


Figura 3.2 Circuito cuántico para Period Finding.

cuántico de la siguiente forma:

$$\sqrt{\frac{r}{q}} \sum_{j=0}^{\frac{q}{r}-1} |x_0 + jr\rangle |f(x_0)\rangle. \tag{3.30}$$

Hemos conseguido una superposición en la que el estado se repite cada cierto valor r, que es el período que estamos buscando. Sin embargo, si realizamos una medición directamente, no nos va a ser de gran utilidad debido al término de desplazamiento aleatorio x_0. Para solucionar esto aplicamos la QFT al primer registro ya que esta es invariante respecto a un desplazamiento lineal, de esta forma se elimina el término x_0,

$$\sqrt{\frac{r}{q}} \frac{1}{\sqrt{q}} \sum_{k=0}^{q-1} \sum_{j=0}^{\frac{q}{r}-1} e^{2\pi i(x_0+jr)k/q} |k\rangle. \tag{3.31}$$

El factor $e^{2\pi i x_0 k/q}$, al no depender de j, se puede sacar fuera de la suma, resultando:

$$\frac{\sqrt{r}}{q} \sum_{k=0}^{q-1} e^{2\pi i x_0 k/q} \sum_{j=0}^{\frac{q}{r}-1} e^{2\pi i jrk/q} |k\rangle. \tag{3.32}$$

Observamos que la suma en j corresponde a una serie geométrica de razón $\rho = e^{2\pi irk/q}$. Se puede demostrar que el valor de esta suma vale

$$\sum_{j=0}^{n-1} \rho^j = \frac{1-\rho^n}{1-\rho}. \tag{3.33}$$

Aplicando este resultado a nuestra suma obtenemos

$$S(k) = \sum_{j=0}^{\frac{q}{r}-1} \left(e^{2\pi irk/q}\right)^j = \frac{1-e^{2\pi ik}}{1-e^{2\pi ikr/q}}. \tag{3.34}$$

$S(k)$ vale cero para los valores de k tales que $e^{2\pi ik} = 1$. Debido a que k es un número entero, el ángulo que forma este número complejo con el eje real siempre va a ser un múltiplo de 2π, por tanto, esta igualdad se va a cumplir siempre. Sin embargo, para los valores en los que k sea un múltiplo entero de q/r, el denominador también se anula, obteniendo una

indeterminación del tipo $0/0$. Para resolverla aplicamos la regla de l'Hôpital:

$$\begin{aligned}\lim_{k\to\frac{q}{r}} S(k) &= \lim_{k\to\frac{q}{r}} \frac{1-e^{2\pi ik}}{1-e^{2\pi ikr/q}} \\ &= \lim_{k\to\frac{q}{r}} \frac{2\pi i e^{2\pi ik}}{2\pi i\frac{r}{q}e^{2\pi ikr/q}} \\ &= \frac{qe^{2\pi iq/r}}{re^{2\pi i}} = \frac{q}{r}. \end{aligned} \tag{3.35}$$

Finalmente concluimos que $S(k)$ vale cero siempre, excepto para los valores $k = \lambda q/r$ con $\lambda \in \{0, 1, 2, \ldots, r-1\}$, que vale q/r. Entonces la ecuación (3.32) nos queda:

$$\frac{1}{\sqrt{r}} \sum_{\lambda=0}^{r-1} e^{2\pi i x_0 \lambda/r} |\lambda\frac{q}{r}\rangle. \tag{3.36}$$

Ahora realizamos una medición y obtenemos el valor c, poco útil a priori debido a la aleatoriedad de λ, pero sabemos que

$$c = \lambda\frac{q}{r} \longrightarrow \frac{c}{q} = \frac{\lambda}{r}. \tag{3.37}$$

Si ni λ ni r tienen factores comunes, c/q puede ser expresado como una fracción irreducible usando la expansión en fracciones continuas [7], obteniendo por tanto ambos valores, λ y r. Esto ocurre con probabilidad $1/\log\log r$. En cualquier otro caso, el algoritmo falla y hay que volver a repetirlo [1, 13]. En el apéndice A se explica en qué consiste el algoritmo de fracciones continuas.

Volviendo a la ecuación (3.32), observamos que la probabilidad de medir un cierto estado $|k\rangle$ es

$$p(k) = |\langle k|\psi\rangle|^2 = \frac{r}{q^2}\left|\sum_{j=0}^{\frac{q}{r}-1} e^{2\pi ijrk/q}\right|^2. \tag{3.38}$$

En la figura 3.3 se observa una representación de esta probabilidad para el caso particular de $N = 33$ y $a = 5$. Para poder ver mejor lo que ocurre, se ha tomado k como una variable continua en vez de una discreta y se ha representado en el rango $[610, 619]$. Se aprecia que para el valor de $k = \lambda q/r$ se produce un máximo. Debido a que estamos trabajando con variables discretas, si q/r no es un número entero, la relación (3.37) no se cumple y el algoritmo de expansión en fracciones continuas no es capaz de encontrar el período correctamente porque el valor c que hemos medido no es realmente $\lambda q/r$ sino una aproximación al entero más cercano.

Si $\lambda q/r - \lfloor \lambda q/r \rfloor < 1/2$, es decir, si está más próximo al entero inferior, este tendrá mayor probabilidad de ser medido frente al superior, y por ende $\lambda q/r \leq c + 1/2$. Si por el contrario sucede que $\lambda q/r - \lfloor \lambda q/r \rfloor > 1/2$, entonces quien tiene ahora mayor probabilidad de ser medido es el superior, teniendo que $\lambda q/r \geq c - 1/2$. Observamos por tanto que se cumple la siguiente desigualdad:

$$c - \frac{1}{2} \leq \lambda\frac{q}{r} \leq c + \frac{1}{2}. \tag{3.39}$$

Restando c:

$$-\frac{1}{2} \leq \lambda\frac{q}{r} - c \leq \frac{1}{2}. \tag{3.40}$$

Si ahora dividimos por q y tomamos valor absoluto:

$$\left|\frac{c}{q} - \frac{\lambda}{r}\right| \leq \frac{1}{2q}. \tag{3.41}$$

Esto quiere decir que el error cometido al aproximar c/q a λ/r es menor o igual a $1/(2q)$. Por tanto, si establecemos esta tolerancia en el algoritmo de expansión en fracciones continua para que deje de iterar nada más alcance dicho error, obtendremos el valor correcto del período.

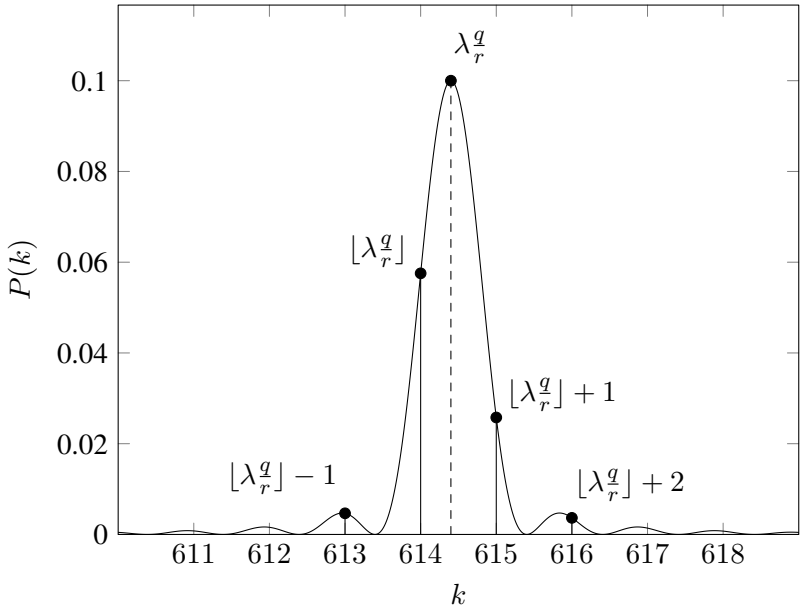


Figura 3.3 Probabilidad de observar un cierto estado $|k\rangle$ en el primer registro una vez aplicada la QFT para $q = 2048$, $r = 10$ y $\lambda = 3$.

Cabe mencionar que la computación cuántica es mas cara y costosa que la computación convencional. Es por esto por lo que es preferible invertir algo más de tiempo en realizar un procesado posterior de los resultados obtenidos que volver a repetir el algoritmo cuántico. Por ejemplo, conviene probar con valores de $r' = r,\ 2r, 3r \ldots$ por si λ/r tuviesen algún factor común, o con valores de $c' = c, c \pm 1, c \pm 2 \ldots$, por si hubiésemos medido algún valor cercano al máximo.

3.5 Resumen del algoritmo

En esta sección vamos describir el algoritmo de Shor paso a paso para factorizar el número N.

Paso 1: Generar un número aleatorio a tal que $1 < a < N$. Posteriormente calculamos el máximo común divisor de a y N, $p = \text{mcd}(a,\ N)$. Pueden ocurrir dos cosas:

- $p > 1$: esto quiere decir que a es un factor no trivial de N. En este caso ya tenemos uno de los factores, el otro se consigue calculando $q = N/p$.
- $p = 1$: vamos al paso 2.

Paso 2: Hay que encontrar el período r de la función $f(x) = a^x \pmod N$. Esta es la parte cuántica del algoritmo y vamos a usar la subrutina de *period-findig*.

Parte cuántica para encontrar el período r

2.1) Inicializar dos registros de tamaño $n = \lceil \log_2(N^2) \rceil$ y $m = \lceil \log_2(N) \rceil$ en el estado

$$|\psi_0\rangle = |0\rangle_n |0\rangle_m. \tag{3.42}$$

2.2) Crear una superposición cuántica en el primer registro, aplicando la transformación QFT:

$$|\psi_1\rangle = QFT|\psi_0\rangle = QFT|0\rangle_n|0\rangle_m = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n |0\rangle_m. \tag{3.43}$$

2.3) Aplicar la función $f(x) = a^x \pmod N$ en el segundo registro mediante la transformación unitaria U_f:

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle_n |a^x \pmod N\rangle_m. \tag{3.44}$$

2.4) Se mide el segundo registro y se obtiene el estado $|f(x_0)\rangle$, por tanto el primer registro colapsa. El nuevo estado de nuestro sistema es

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n/r}} \sum_{j=0}^{\frac{2^n}{r}-1} |x_0 + rj\rangle |f(x_0)\rangle \tag{3.45}$$

2.5) Se aplica la transformación QFT al primer registro, quedando por tanto

$$|\psi_4\rangle = \frac{1}{\sqrt{2^n/r}} \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i k/2^n} \sum_{j=0}^{2^n/r-1} e^{2\pi i r j k/2^n} |k\rangle |f(x_0)\rangle. \tag{3.46}$$

Entonces la probabilidad de medir un cierto $|k\rangle$ viene dada por

$$p(k) = |\langle k|\psi_4\rangle|^2 = \frac{r}{2^n \times 2^n} \left| \sum_{j=0}^{2^n/r-1} e^{2\pi i r j k/2^n} \right|^2. \tag{3.47}$$

Al medir el primer registro se obtiene algún valor $c = \lambda q/r \ \ \forall \lambda \in \{0, 1, 2, \ldots, r-1\}$. Según la ecuación (3.41), el error de aproximar λ/r a c/q es menor o igual a $1/(2q)$, si establecemos esta tolerancia en el algoritmo de expansión en fracciones continuas obtendremos r. Comprobamos si efectivamente el período calculado es el correcto. También podemos probar con valores del período tales que $r' = r,\ 2r,\ 3r \ldots$ Si el período es el correcto, vamos al paso 3, en caso contrario regresamos al paso 2.1.

Paso 3: Una vez hemos calculado el período hay que comprobar que sea par. Si es par, vamos al paso 4, si no, volvemos al 1.

Paso 4: Comprobamos que $a^{r/2} + 1 \neq 0 \pmod N$. Si es distinto, vamos al paso 5, en caso contrario regresamos al paso 1.

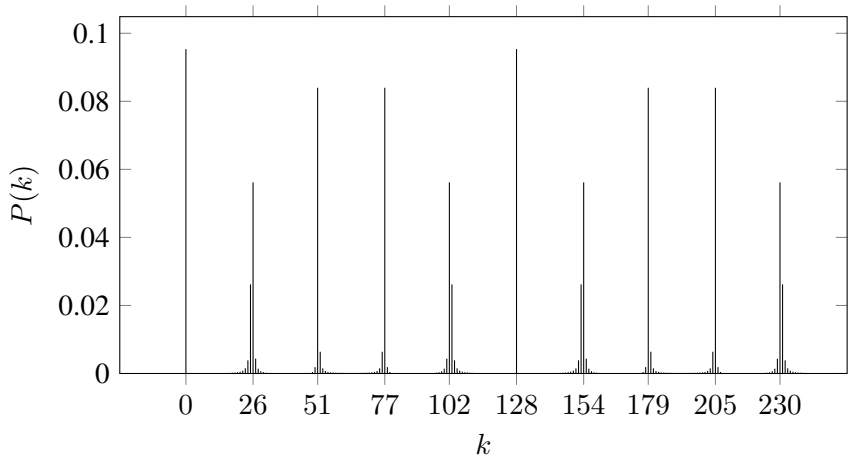


Figura 3.4 Probabilidad de observar un cierto estado $|k\rangle$ en el primer registro una vez aplicada la QFT, con $q = 256$ y $r = 10$.

Paso 5: Los factores de N son $p = \text{mcd}(a^{r/2} - 1,\ N)$ y $q = \text{mcd}(a^{r/2} + 1,\ N)$.

3.6 Ejemplo de funcionamiento

En esta sección vamos a mostrar un ejemplo para comprobar el funcionamiento del algoritmo. Vamos a factorizar un número sencillo, ya que vamos a hacerlo a mano, y para demostrar el funcionamiento básico del algoritmo es suficiente. Por ejemplo, queremos factorizar el número $N = 33$. A priori sabemos que es 3×11, pero vamos a aplicar el procedimiento explicado en este capítulo:

Paso 1: Generar un número aleatorio a tal que $1 < a < N$. Posteriormente calculamos el máximo común divisor de a y N, $p = \text{mcd}(a,\ N)$. Pueden ocurrir dos cosas:

- $p > 1$: esto quiere decir que a es un factor no trivial de N. En este caso ya tenemos uno de los factores, el otro se consigue calculando $q = N/p$.
- $p = 1$: vamos al paso 2.

En nuestro caso tenemos que $a = 5$ y $\text{mcd}(5,\ 33) = 1$, luego vamos al paso 2.

Paso 2: Hay que encontrar el período r de la función $f(x) = a^x \pmod N$. Esta es la parte cuántica del algoritmo y vamos a usar la subrutina de *period-findig*. En el caso que nos conlleva es fácil ver que

$$\begin{aligned} 5^1 &= 5 \pmod{33} \\ 5^2 &= 25 \pmod{33} \\ 5^3 &= 26 \pmod{33} \\ 5^4 &= 31 \pmod{33} \\ &\vdots \\ 5^{10} &= 1 \pmod{33} \\ &\vdots \end{aligned}$$

así que el período es 10. Pero esto hay que calcularlo usando computación cuántica.

Parte cuántica para encontrar el período r

2.1) Inicializar dos registros de tamaño $n = \lceil \log_2(N^2) \rceil$ y $m = \lceil \log_2(N) \rceil$ en el estado

$$|\psi_0\rangle = |0\rangle_n |0\rangle_m. \tag{3.48}$$

En nuestro caso $n = 11$, $m = 6$ y por tanto $q = 2048$.

2.2) Crear una superposición cuántica en el primer registro, aplicando la transformación QFT:

$$|\psi_1\rangle = QFT|\psi_0\rangle = QFT|0\rangle_{11}|0\rangle_6 = \frac{1}{\sqrt{2^{11}}} \sum_{x=0}^{2^{11}-1} |x\rangle_{11}|0\rangle_6. \tag{3.49}$$

2.3) Aplicar la función $f(x) = 5^x \pmod{33}$ en el segundo registro mediante la transformación unitaria U_f:

$$|\psi_2\rangle = U_f|\psi_1\rangle = \frac{1}{\sqrt{2^{11}}} \sum_{x=0}^{2^{11}-1} |x\rangle_{11}|5^x \pmod{33}\rangle_6. \tag{3.50}$$

Si desarrollamos la suma:

$$\begin{aligned}|\psi_2\rangle = \frac{1}{\sqrt{2^{11}}}(&|0\rangle|1\rangle + |1\rangle|5\rangle + |2\rangle|25\rangle + |3\rangle|26\rangle + |4\rangle|31\rangle + |5\rangle|23\rangle \\ &+ |6\rangle|16\rangle + |7\rangle|14\rangle + |8\rangle|4\rangle + |9\rangle|20\rangle + |10\rangle|1\rangle + |11\rangle|5\rangle + \cdots + |2047\rangle|14\rangle),\end{aligned} \tag{3.51}$$

observamos que los valores del segundo registro se repiten con un período de 10.

2.4) Se mide el segundo registro y se obtiene el estado $|26\rangle$, por tanto el primer registro colapsa. El nuevo estado de nuestra máquina es

$$|\psi_3\rangle = \frac{1}{\sqrt{2^{11}/10}} \left(|3\rangle + |13\rangle + |23\rangle + \cdots + |2043\rangle\right)|26\rangle \tag{3.52}$$

$$= \frac{1}{\sqrt{2^{11}/10}} \sum_{j=0}^{\frac{2^{11}}{10}-1} |3 + 10j\rangle|26\rangle \tag{3.53}$$

2.5) Se aplica la transformación QFT al primer registro, quedando por tanto

$$|\psi_4\rangle = \frac{1}{\sqrt{2^{11}/10}} \frac{1}{\sqrt{2^{11}}} \sum_{k=0}^{2^{11}-1} e^{2\pi i k/2^{11}} \sum_{j=0}^{2^{11}/10-1} e^{2\pi i 10jk/2^{11}} |k\rangle|26\rangle. \tag{3.54}$$

Entonces la probabilidad de medir un cierto $|k\rangle$ viene dada por

$$p(k) = |\langle k|\psi_4\rangle|^2 = \frac{10}{2^{11} \times 2^{11}} \left| \sum_{j=0}^{2^{11}/10-1} e^{2\pi i 10jk/2^{11}} \right|^2. \tag{3.55}$$

Al medir el primer registro se obtiene algún valor $c = \lambda q/r \ \forall \lambda \in \{0, 1, 2, \ldots, r-1\}$. Suponemos que hemos obtenido el valor de $c = 819$. Si calculamos c/q tenemos que

$819/2048 = 0.399902$. Según la ecuación (3.41), el error de aproximar λ/r a c/q es menor o igual a $1/(2q) = 0.0002$, por tanto podemos tomar $\lambda/r = 0.4$. Aplicando el algoritmo de expansión en fracciones continuas obtenemos que $0.4 = 2/5$. Comprobamos si efectivamente el período es 5 y obtenemos que $5^5 = 23 \pmod{33} \neq 1 \pmod{33}$, por lo que vemos que no es el período. Si probamos con $r' = 2 \times 5 = 10$ tenemos $5^{10} = 1 \pmod{33}$, luego el período es 10.

Paso 3: Una vez hemos calculado el período hay que comprobar que sea par. Si es par, vamos al paso 4, si no, volvemos al 1. Como nuestro r es par vamos al paso 4.

Paso 4: Comprobamos que $a^{r/2} + 1 \neq 0 \pmod{N}$. Si es distinto, vamos al paso 5, en caso contrario regresamos al paso 1. En nuestro caso $5^5 + 1 = 24 \pmod{33}$, luego vamos al paso 5.

Paso 5: Los factores de N son $p = \text{mcd}(a^{r/2} - 1,\, N)$ y $q = \text{mcd}(a^{r/2} + 1,\, N)$. En nuestro caso $p = \text{mcd}(5^5 - 1,\, 33) = 11$ y $q = \text{mcd}(5^5 + 1,\, 33) = 3$. Concluimos que $N = 3 \times 11 = 33$.

3.7 Algoritmo de Matlab

Para finalizar el trabajo, se ha escrito un código en Matlab que simula un computador cuántico para ejecutar el algoritmo de Shor estudiado en el capítulo 3. Todo el código fuente se puede encontrar en el apéndice B. La función principal se llama *factoriza()*. Esta recibe un número (es importante que sea producto de dos números primos) y devuelve un vector con los factores. Si por ejemplo le pasamos el número 33 obtenemos la traza mostrada en la figura 3.5.

Finalmente obtenemos con éxito en una sola iteración que los factores de 33 son 3 y 11.

```
>> factores = factoriza(33)

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
COMENZANDO PARTE CLASICA:

factoriza: Nueva iteracion. a = 20

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
COMENZANDO PARTE CUANTICA:
encuentra_periodo: n = 11, m = 6

encuentra_periodo: Iteracion numero 1.
encuentra_periodo: Preparando estado inicial.
encuentra_periodo: Creando superposicion en primer registro.
encuentra_periodo: Aplicando funcion a^x (mod N) en segundo registro.
encuentra_periodo: Midiendo el segundo registro.
encuentra_periodo: Aplicando la QFT al primer registro.
encuentra_periodo: Midiendo el primer registro.
encuentra_periodo: Proceso terminado con 1 iteraciones: q = 2048, a = 20, c = 1229, r↙
= 10 (periodo real 10)

factoriza: He usado el periodo r = 10

factores =

    11     3

>>
```

Figura 3.5 Traza de Matlab al factorizar el número 33.

Resumen y conclusiones

A lo largo de este texto, hemos ido estudiando los elementos y conceptos de la computación cuántica, desde una breve introducción a la mecánica cuántica, hasta el análisis de un algoritmo cuántico para factorizar números en factores primos. Es en esta última aplicación donde se ve la gran potencia de la computación cuántica. Una tarea fácil de formular, se convierte en un gran problema cuando tratamos de factorizar números muy grandes usando ordenadores clásicos, y es prácticamente imposible de resolver puesto que nos llevaría millones de años de cálculo. Sin embargo, debido a la propiedad de superposición de estados de un ordenador cuántico, podemos realizar todos estos cálculos en una sola iteración, brindando un gran abanico de aplicaciones en el futuro, como por ejemplo la encriptación cuántica. Es por esto por lo que numerosas empresas se encuentran en investigación y desarrollo de esta tecnología para ser capaces de construir ordenadores cuánticos de cada vez más qubits.

Para poder realizar este trabajo, he tenido que estudiar, entender y comprender los conceptos básicos de la mecánica cuántica, ya que en el Grado de Ingeniería de las Tecnologías Industriales no se ve en ningún momento, así como la notación de Dirac, y el producto interno, que antes no sabía que existían. He tenido que leer decenas de fuentes para poder entenderla. Una vez alcanzados los conocimientos básicos sobre mecánica cuántica, empecé a profundizar en computación cuántica, comenzando por entender el qubit, qué es, sus propiedades y cómo se representa en la esfera de Bloch, así como la asociación con el espín. Para entender como se proyecta el espín en determinadas direcciones, tuve que mirar y comprender el experimento de Stern Gerlach. Posteriormente, para tratar sistemas de múltiples qubits, había que estudiar el producto interno y sus propiedades. Finalmente, para concluir el capítulo 1, tuve que comprender cómo un conjunto de elementos, habiendo interactuado entre sí en algún momento, la acción sobre uno individual puede provocar una reacción inmediata en el resto, aunque ya no formen parte del mismo sistema. Esto es lo que se llama entrelazamiento cuántico.

En el capítulo 2 estudiamos cómo realizar operaciones con qubits, tanto con uno solo, como con varios, e introdujimos algunas puertas cuánticas para implementar dichas operaciones. También introdujimos el concepto de sistema reversible y mostramos que no existe ninguna puerta cuántica capaz de clonar un estado cualquiera. Concluimos el capítulo con los ejemplos de superdense coding y teletransporte cuántico, que son formas complementarias de codificar información.

En el capítulo 3 vimos una aplicación directa de la computación cuántica: el algoritmo de Shor. Este algoritmo se basa en la idea de que el problema de hacer una factorización en números primos se puede reducir simplemente a encontrar el período de una cierta función, pudiéndose implementar eficientemente en un ordenador cuántico usando el algoritmo de *period findig*. También hablamos sobre la versión cuántica de la transformada de Fourier,

ya que juega un papel muy importante a la hora de encontrar el período. En este capítulo tuve que entender la aritmética modular, ya que es la herramienta principal para encontrar el período, así como recordar la transformada de Fourier para poder aplicarla a la computación cuántica y elaborar un circuito cuántico que la implementase. Para comprobar que todo esto funcionase he hecho un programa en Matlab que simula un ordenador cuántico ejecutando el algoritmo de Shor. Me llevó mucho tiempo programarlo, ya que al principio no me di cuenta de que al calcular las fracciones continuas para sacar el período, había que hacer una aproximación, pasándole a la función *rat()* de Matlab una tolerancia. Lo que hacia es calcular el número exacto, y en los casos en los que el período no dividía a 2^n, el algoritmo no era capaz de encontrar el período correcto. Finalmente me di cuenta de este pequeño detalle y logré que funcionase tal y como era de esperar. Evidentemente, al tratarse de una simulación y no ser un algoritmo especialmente optimizado para un ordenador convencional, los tiempos de cálculo son relativamente grande, pero factorizando números no más de tres cifras lo hace en un tiempo razonable.

Apéndice A

Algoritmo de fracciones continuas

La idea fundamental del algoritmo de fracciones continuas es en describir número reales usando solamente números enteros, mediante expresiones de la forma

$$[a_0,\ldots,a_M] \equiv a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \cfrac{1}{a_M}}}}, \tag{A.1}$$

donde $[a_0,\ldots,a_M]$ son enteros positivos. Si paramos de iterar antes de llegar al final del algoritmo, definimos la aproximación *m-ésima convergente* de esta fracción continua como $[a_0,\ldots,a_m]$, con $0 \leq m \leq M$. El algoritmo de fracciones continuas es un método para determinar la expansión en fracciones continuas de un número real arbitrario. Por ejemplo, supongamos que queremos descomponer $31/13$ como una fracción continua. El primer paso del algoritmo es separar $31/13$ en parte entera y decimal,

$$\frac{31}{13} = 2 + \frac{5}{13}. \tag{A.2}$$

Posteriormente invertimos la parte decimal, obteniendo

$$\frac{31}{13} = 2 + \frac{1}{\frac{13}{5}}. \tag{A.3}$$

Estos pasos de separar en parte entera y decimal e invertir la parte decimal, se repite ahora con $13/5$, resultando

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{3}{5}} = 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}}. \tag{A.4}$$

Después, separamos e invertimos $5/3$

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{2}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{\frac{3}{2}}}}. \tag{A.5}$$

La descomposición en fracciones continuas finaliza al obtener un 1 en el numerador de la parte decimal sin ser necesario invertirla. En nuestro caso observamos que

$$\frac{3}{2} = 1 + \frac{1}{2}, \tag{A.6}$$

Finalmente terminamos con el siguiente resultado

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1+\frac{1}{1+\frac{1}{2}}}} = [2,2,1,1,2]. \tag{A.7}$$

Queda completamente claro que este algoritmo converge para cualquier número siempre que sea racional. Esta expansión permitirá conocer la fracción irreducible para cualquier número real x.

Si establecemos una tolerancia ϵ, y paramos de iterar nada más se cumpla que

$$\left|x - [a_0, \ldots, a_m]\right| \leq \epsilon, \tag{A.8}$$

No tendremos que completar el algoritmo hasta el final. De hecho, usando esta tolerancia podremos calcular fracciones aproximadas de números irracionales, siendo todo lo preciso que uno quiera [10].

Apéndice B
Uso de Matlab para implementar el algoritmo de Shor

Para comprobar el algoritmo de Shor se ha escrito una función de Matlab para encontrar los factores de números producto de dos primos. La parte clásica se muestra en el código B.1, la cual está implementada en la función *factoriza*. Esta función recibe el número que queremos factorizar y devuelve un vector con los factores. A su vez, es esta función la que llama a la subrutina cuántica *encuentra_periodo* para calcular el período.

Código B.1 Función para encontrar los factores de N.

```
function f = factoriza (N)
% Factoriza el numero N en producto de dos primos.

% Comprueba que sea impar
if mod(N, 2) == 0
    f(1) = 2;
    f(2) = N / 2;
    fprintf(' factoriza : N es par\n')
    return
end

% Elige un numero aleatorio y comprueba que x y N sean coprimos
i = 1;
while i < N - 1
    a = randi([2, N-1]);
    fprintf('\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n')
    fprintf('COMENZANDO PARTE CLASICA:\n')
    fprintf('\nfactoriza : Nueva iteracion . a = %d\n', a)
    p = gcd(a, N);
    if p > 1
        f(1) = p;
        f(2) = N / p;
        fprintf(' factoriza : He tenido suerte y mcd(%d, %d) = %d es un factor de N!\n', a, N, p)
        return
    end

    % Encuentra el orden
    r = encuentra_periodo(a, N);
    if mod(r, 2) == 0
        if mod(a^(r/2) + 1, N) ~= 0
            p = gcd(a^(r/2) - 1, N);
```

```
            q = gcd(a^(r/2) + 1, N);
            if p ~= 1 && q ~= 1
                f(1) = p;
                f(2) = q;
                fprintf('factoriza: He usado el periodo r = %d\n', r)
                return
            end
        else fprintf('factoriza: Periodo no valido porque a^(r/2) = -1 (mod N)\n')
        end
    else fprintf('factoriza: Periodo no valido porque r es impar\n')
    end
i = i + 1;
end

f(1) = N;
f(2) = 1;
```

Código B.2 Función para encontrar el período.

```
function r = encuentra_periodo(a, N)
% Encuentra el periodo de la funcion f para los valores N y a dados.

n = ceil(log2(N^2));
q = 2^n;
m = ceil(log2(N));
fprintf('\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n')
fprintf('COMENZANDO PARTE CUANTICA:\n')
fprintf('encuentra_periodo: n = %d, m = %d\n', n, m)

repetir = 1;
iter = 0;
while repetir
    iter = iter + 1;
    % Preparamos estado inicial |psi_0> = |0>|0>
    fprintf('\nencuentra_periodo: Iteracion numero %d.\n', iter)
    fprintf('encuentra_periodo: Preparando estado inicial .\n')
    psi_0 = [1, zeros(1, q-1)];

    % Creamos superposicion en el primer registro
    fprintf('encuentra_periodo: Creando superposicion en primer registro .\n')
    psi_1 = ones(size(psi_0));
    psi_1 = psi_1/sqrt(sum(psi_1));
    clear psi_0;
    r1 = 0:q-1;

    % Aplicamos f(x) = a^r1 (mod N) en el segundo registro
    fprintf('encuentra_periodo: Aplicando funcion a^x (mod N) en segundo registro .\n')
    psi_2 = psi_1;
    clear psi_1;
    r2 = f(a, r1, N);

    % Medimos el segundo registro
    fprintf('encuentra_periodo: Midiendo el segundo registro .\n')
    fx_0 = dot(r2, measure(psi_2));
    clear psi_2;
    psi_3 = r2 == fx_0;
    psi_3 = psi_3 / sqrt(sum(psi_3));

    % Aplicamos QFT en el primer registro
    fprintf('encuentra_periodo: Aplicando la QFT al primer registro .\n')
```

```
    psi_4 = QFT(psi_3);
    clear psi_3;
    bar(0:q-1, abs(psi_4).^2);

    % Guarda los datos en un fichero
    fic = fopen('barras.dat', 'w');
    for k = 0:q-1
        fprintf(fic, '%d\t%f\n', k, abs(psi_4(k+1))^2);
    end
    fclose(fic);


    % Medimos el primer registro
    fprintf('encuentra_periodo: Midiendo el primer registro.\n')
    c = dot(r1, measure(psi_4));
    clear psi_4;

    % Aplicamos algoritmo de expansion en fracciones continuas para obtener
    % el periodo. A la funcion rat() se le introduce una tolerancia de
    % 1/(2*q) para aproximar c/q a lambda/r
    rf = periodo(r2);

    [~, r_candidato] = rat(c/q, 1/(2*q));
    for k = 1:9
        r = k*r_candidato;
        if f(a, r, N) == f(a, 2*r, N)
            repetir = 0;
            break
        end
    end

    if repetir
        fprintf('encuentra_periodo: %d no es el periodo (periodo real %d).\n', r_candidato, rf)
    end
end

fprintf('encuentra_periodo: Proceso terminado con %d iteraciones: q = %d, a = %d, c = %d, r =
    %d (periodo real %d)\n\n', iter, q, a, c, r, rf)
```

Debido a que el valor de la función a^x crece exponencialmente con x, no se puede usar la función *mod()* de Matlab para calcular los valores de $f(x)$, ya que para x suficientemente grande, Matlab lo considera infinito. Es por esto por lo que se ha empleado una librería de java para hacer la operación módulo sin calcular a^x directamente. En el código B.3 se puede visualizar la función que calcula $f(x)$.

Código B.3 Función para calcular $f(x) = a^x \pmod N$.

```
function y = f(a, x, N)

a_big = java.math.BigInteger(num2str(a));
N_big = java.math.BigInteger(num2str(N));
[rows, columns] = size(x);
y = zeros(rows, columns);
for i = 1:length(x)
    x_big = java.math.BigInteger(num2str(x(i)));
    y(i) = double(a_big.modPow(x_big, N_big));
end
```

Para calcular la transformada cuántica de Fourier, primero se planteó construir la matriz QFT_N y multiplicarla por el estado a transformar, pero volvía a ocurrir que el sistema se

quedaba sin memoria para valores de N relativamente grandes. Entonces se optó por crear una función que aceptase un vector y devolviese su transformado, solo siendo necesario almacenar en memoria una fila de la matriz QFT_N en lugar de todas. El vector transformado se calcula componente a componente, y cada una de ellas es el producto escalar entre la fila j y el vector de entrada a la función.

Código B.4 Función para calcular la QFT de un estado dado.

```
function qft = QFT(s)
% Calcula la componente j+1 del vector de salida como el producto escalar
% del vector s de entrada y la fila j+1 de la matriz QFT

N = length(s);
w = exp(2*pi*1i/N);

qft = ones(size(s));
for j = 0:N-1
        qft(j+1) = dot(s, w.^(j * (0:N-1))) / sqrt(N);
end
```

Finalmente la función *measure* lo que hace es tomar un vector de estado y le aplica la medición, devolviendo el estado colapsado. Se basa en la idea de que tiene que coger una componente del vector aleatoriamente teniendo en cuenta los pesos de cada una de ellas.

Código B.5 Función para realizar la medida cuántica.

```
function y = measure(psi)
% Realiza la medida cuantica. Dada la funcion de onda psi devuelve la
% funcion de onda colapsada.

p = (abs(psi)).^2;
[sp,ip]=sort(p);
sp=cumsum(sp);
r = rand;
i=1;
while r>sp(i)
    i=i+1;
end
obs=ip(i);
y = zeros(size(psi));
y(obs) = 1;
```

Con el fin de comprobar el correcto comportamiento de esta función, también se ha escrito un pequeño código de prueba, que lo que hace es crear un vector de probabilidades de cuatro componentes y se lo pasa a la función **measure()**. Esta devuelve el estado colapsado. Por tanto, según se muestra en el código B.6, la probabilidad de medir los estados

$$x_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad x_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{B.1}$$

es, respectivamente 0.4, 0.1, 0.3 y 0.2.

Si realizamos esta operación un número muy grande de veces y calculamos la probabilidad de medir un cierto estado como el número de veces que ha salido entre el número de experimentos realizados, debe salir las probabilidades predefinidas.

Código B.6 Código para probar la función de medida.

```
clear, clc
p = [0.4  0.1  0.3  0.2];
x = 1:length(p);
p = sqrt(p);

for i = 1:10000
    y(i) = dot(x, measure(p));
end

for i = 1:length(x)
    s(i) = sum(y == x(i)) / length(y);
end

s
```

Si ejecutamos este código en Matlab, realizando 10000 mediciones, obtenemos que las probabilidades son 0.4035, 0.0959, 0.2978 y 0.2028, observando que aproximadamente coinciden con los valores de probabilidad previamente establecidos. El error tiende a cero cuando el número de mediciones tiende a infinito.

Índice de Figuras

Índice de Tablas

Bibliografía

[1] Giuliano Benenti, Giulio Casati, and Giuliano Strini, *Principles of quantum computation and information*, vol. 1: Basic concepts, World Scientific, 2004.

[2] Charles H. Bennett and Stephen J. Wiesner, *Communication via one- and two-particle operators on einstein-podolsky-rosen states*, Phys. Rev. Lett. **69** (1992), 2881–2884.

[3] Francois David, *The formalisms of quantum mechanics*, Lecture Notes in Physics **893** (2015), 42–43.

[4] Paul Adrien Maurice Dirac, *A new notation for quantum mechanics*, Mathematical Proceedings of the Cambridge Philosophical Society, vol. 35, Cambridge University Press, 1939, pp. 416–418.

[5] Carl Friedrich Gauss and translated by Arthur A. Clarke, *Disquisitiones arithmeticae*, english ed., ch. 1 «Numbers congruences in general», Springer-Verlag, New York, 1986.

[6] Robert B. Griffiths, *Nature and location of quantum information*, Phys. Rev. A **66** (2002), 012311.

[7] Godfrey Harold Hardy and Edward Maitland Wright, *An introduction to the theory of numbers*, ch. X, Oxford university press, 1979.

[8] Donald E. Knuth, *The art of computer programming*, second ed., vol. 2: Seminumerical Algorithms, Addison-Wesley, 1981.

[9] Arjen K. Lenstra, Hendrik W. Lenstra, Mark S. Manasse, and John M. Pollard, *The number field sieve*, The development of the number field sieve, Springer, 1993, pp. 11–42.

[10] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, Cambridge, U.K., 2010.

[11] Ronald L. Rivest, Adi Shamir, and Leonard Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM **21** (1978), no. 2, 120–126.

[12] J. J. Sakurai, *Modern quantum mechanics*, revised ed., Addison-Wesley, 1994.

[13] Peter W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.*, SIAM review **41** (1995), no. 2, 303–332.

[14] W. K. Wootters and W. H. Zurek, *A single quantum cannot be cloned.*, Nature **299** (1982), no. 5886, 802.