UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR

INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

MASTER UNIVERSITARIO EN INVESTIGACIÓN

EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Operaciones Aritméticas Básicas con Circuitos Cuánticos

Autor:

Dª. Lidia Ruiz Pérez

Tutor:

Dr. D. Juan Carlos García Escartín

Valladolid, 5 de septiembre de 2014

TÍTULO:	**Operaciones Aritméticas Básicas con Circuitos Cuánticos**
AUTOR:	**Dª. Lidia Ruiz Pérez**
TUTOR:	**Dr. D. Juan Carlos García Escartín**
DEPARTAMENTO:	**Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

Tribunal

PRESIDENTE:	**Dr. D. Alonso Alonso Alonso.**
VOCAL:	**Dr. D. César Palencia de Lara.**
SECRETARIO:	**Dr. D. Juan Ignacio Asensio Pérez.**
FECHA:	**10 de septiembre de 2014**
CALIFICACIÓN:	

Resumen del TFM

Los ordenadores que aprovechan las leyes de la Mecánica Cuántica son capaces de realizar de forma eficiente tareas inabordables para un ordenador clásico como, por ejemplo, la factorización de números. Ciertas operaciones de uso frecuente, como resolver sistemas de ecuaciones lineales, aunque son eficientes en ordenadores clásicos, se pueden resolver en menos tiempo con circuitos cuánticos. En este trabajo se estudiarán las posibles mejoras de eficiencia en el cálculo de operaciones aritméticas básicas. En particular, se estudiará un sumador cuántico y un multiplicador que operan trabajando con la transformada cuántica de Fourier.

Palabras clave

Computación cuántica, sumador, multiplicador, transformada cuántica de Fourier.

Abstract

Computers that take advantage of Quantum Mechanics are capable of doing tasks that are unapproachable for a classical computer in an efficient manner as, for example, number factoring. Common operations, like solving systems of linear equations, can be solved efficiently using a classical computer, but could be solve in less time making use of quantum circuits. In this work some improvements in the efficiency of basic arithmetic operations will be studied. In particular we will work on a quantum adder and a multiplier that makes use of the quantum Fourier transform.

Keywords

Quantum computation, adder, multiplier, quantum Fourier transform.

Agradecimientos

Quisiera agradecer a mi tutor, Juan Carlos, y a mi familia su infinita paciencia conmigo durante la elaboración de este TFM.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación y Objetivos

Un ordenador cuántico es una máquina es una máquina física que trabaja con estados de entrada que representan una superposición coherente de posibles entradas y las transforma en los correspondientes estados de salida que representan una superposición coherente de resultados [VBE96]. Como vemos, la función de los ordenadores cuánticos no difiere demasiado de la función de los ordenadores clásicos. Sin embargo, existe una diferencia entre ambos ordenadores, y es que los ordenadores cuánticos invocan intrínsicamente fenómenos mecánico-cuánticos [BCDP96] para operar sobre los datos de entrada.

Esta característica de los ordenadores cuánticos tiene entre sus consecuencias que el proceso de computación, es decir, la aplicación sobre los estados de entrada de una serie de operaciones unitarias, afecta de manera simultánea a todos los elementos que componen la superposición, lo que permite paralelizar el procesamiento de datos aunque se cuente con una única pieza de hardware cuántico [VBE96]. Esta capacidad de operar sobre todos los elementos de la superposición al mismo tiempo podría proporcionar, en algunos casos, mayor velocidad de computación con respecto a los ordenadores clásicos, así como la posibilidad de resolver problemas que en principio son imposibles de resolver con un ordenador clásico. De ahí nace parte del interés por la computación cuántica.

Los ordenadores cuánticos trabajan con unidades de información llamadas qubits. Los qubits el análogo cuántico al bit clásico, pero con alguna diferencia, pues los qubits son sistemas cuánticos que cuentan con dos estados accesibles pero que, además pueden existir en una superposición de ambos estados [DiV95], mientras que los bits clásicos se pueden encontrar o en el estado 0 o en el estado 1, pero no en una superposición de ambos.

Sobre los qubits se pueden realizar operaciones, tal y como sucede con los bits clásicos. Sin embargo, las operaciones que se apliquen sobre los qubits deben ser operaciones unitarias y reversibles, pues deben respetar las leyes de la mecánica cuántica, como explicaremos en el capítulo 2. Esta restricción conlleva que algunas de las operaciones lógicas que se realizan en computación clásica no puedan reproducirse de forma directa en computación cuántica, como es el caso de la operación lógica AND o la operación OR: estas puertas son irreversibles, pues leer un 1 a la salida no proporciona suficiente

información para conocer la entrada, que podría ser cualquiera de los pares $(0,1)$, $(1,0)$ o $(1,1)$ [VBE96]. Sin embargo, haciendo uso de las operaciones unitarias adecuadas, es decir, de la lógica cuántica, se puede definir una operación o puerta cuántica que dé como resultado la operación AND [DiV95].

Puesto que no todas las operaciones que se realizan en computación clásica se pueden reproducir de forma directa en computación cuántica es necesario definir la aritmética cuántica. Por ejemplo, en una operación sencilla como puede ser la suma el algoritmo clásico que la implementa no se puede trasladar de forma directa a un algoritmo cuántico. Esto es debido a que la suma se implementa mediante las operaciones lógicas AND y OR, que son irreversibles, por lo que no son operaciones permitidas en computación cuántica.

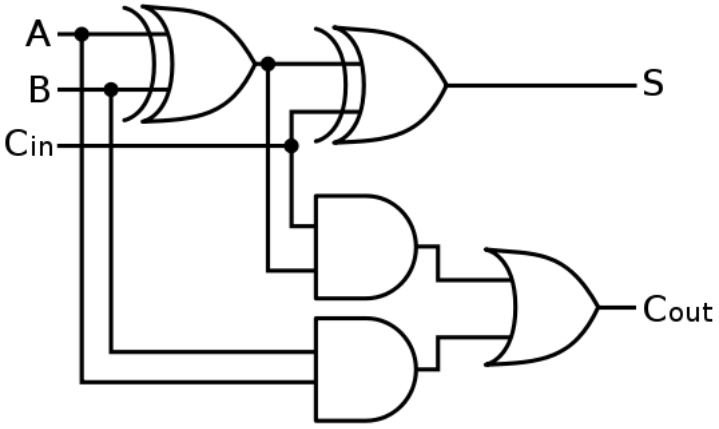


Figura 1.1: Sumador clásico.

En este sentido se han realizado numerosos trabajos que proponen circuitos cuánticos que realizan la suma de dos números, representados por su expansión binaria de n bits. Por ejemplo, en [BCDP96] se proponen varios circuitos cuánticos para realizar la factorización cuántica que incluyen un sumador, un sumador módulo N y un multiplicador módulo N entre otros, con $N = 2^n$. En [VBE96] se propone un sumador cuántico que emula el esquema clásico pero haciendo uso de la lógica cuántica. Otras aproximaciones implementan esquemas con técnicas de acarreo almacenado, como en [Gos98] y [Zal98]. Todos ellos están inspirados en los esquemas clásicos y requieren al menos $3n$ qubits para sumar dos números de n bits [Dra00].

En [Dra00] se propone un sumador cuántico que no se basa en el esquema clásico. Este esquema hace uso de la transformada cuántica de Fourier o QFT. Haciendo uso de este algoritmo, los números con los que se desea operar se codifican como desfases entre los elementos que componen el registro. De esta manera transforma a en $F(a)$ y luego lo hace evolucionar a $F(a+b)$. Realizando la QFT inversa se recupera el resultado de sumar a y b. Este circuito utiliza $2n$ qubits para sumar dos números de n bits. Este circuito se utiliza en [Bea02] para implementar el algoritmo de Shor para la factorización cuántica.

Sin embargo el sumador QFT, tal y como está definido en [Dra00], en realidad realiza la suma módulo N. Al hacer la QFT^{-1} de $F(a+b)$ recuperará o bien $a+b$ si $a+b \leq N$

o $N - (a + b)$ si $a + b > N$. Lo que hace pensar que puede ser posible extender el esquema para recuperar $a + b$. Por otra parte, la propuesta deja sin discutir aspectos como la respresentación de números con signo o la suma de más de dos números.

En lo que se refiere a los multiplicadores, existen pocos esquemas propuestos. En [VBE96] se propone un esquema de multiplicador modular que calcula la multiplicación módulo N de dos números a y b. Para ello se modifica el sumador propuesto en el mismo artículo para que realice la suma módulo N y se utiliza para realizar n sumas módulo N controladas. En [BCDP96] se demuestra que utilizando repetidas sumas modulares se puede construir un multiplicador modular. Por último, en [Bea02] se propone un circuito que realiza la multiplicación módulo N haciendo uso de la QFT. Vemos que existen algunos esquemas propuestos que realizan el cálculo $a \cdot b$ módulo N, pero no el producto $a \cdot b$.

En [KTR14] se propone un esquema que realiza la suma de productos parciales en paralelo, siguiendo el modelo de árbol binario. Este esquema se basa en sumadores cuánticos que emulan el sistema clásico, lo que hace pensar que se puede proponer un esquema que reduzca el número de qubits necesario para hallar el producto de dos números de n bits.

Una vez establecida la necesidad de diseñar operaciones aritméticas sencillas empleando circuitos cuánticos y definido el trabajo desarrollado en este aspecto, se define como objetivo de este TFM plantear esquemas de operaciones aritméticas básicas con puertas cuánticas. Para cumplir este objetivo, se definen los siguientes objetivos parciales:

- Realizar un estudio de las operaciones aritméticas ya implementadas.
- Mejorar los esquemas planteados para el sumador cuántico:
  - Completar el esquema del sumador QFT:
    - Extender el esquema para sumar k números.
    - Obtener la suma y no la suma módulo N.
    - Discutir la representación de números enteros con signo.
    - Implementar la media.
    - Implementar la suma ponderada.
- Implementar un multiplicador basado en el sumador QFT.

En la siguiente sección se explica la metodología propuesta para resolver los objetivos propuestos en este TFM.

1.2. Metodología

El trabajo desarrollado a lo largo de este TFM es fundamentalmente teórico. Esto quiere decir que propondremos circuitos que implementen operaciones aritméticas sencillas y presentaremos la base matemática que justifica el funcionamiento de los mismos, pero no vamos a probar los circuitos y medir su funcionamiento en términos de, por ejemplo, tiempo de ejecución.

Para cumplir con los objetivos propuestos en la sección anterior, estudiaremos primero la bibliografía con el objetivo de identificar los circuitos ya propuestos. Sobre estos circuitos, trabajaremos sobre aquellos en los que detectemos la posibilidad de completar el esquema propuesto y que nos ofrezcan la base para desarrollar nuevas operaciones algebraicas. Propondremos una solución a los problemas o debilidades detectadas, y se desarrollará la base matemática que pruebe que el circuito propuesto resuelve el problema detectado.

Para el desarrollo del circuito que realice la multiplicación, estudiaremos los circuitos propuestos y los distintos algoritmos que utilizamos habitualmente para hallar el producto de dos números. Después se propondrá un circuito que implemente la multiplicación. En este caso, propondremos un circuito que resuelva el producto mediante sumas secuenciales, pues podemos basarnos en el trabajo ya realizado sobre sumadores. Por último, mostraremos matemáticamente que el circuito realiza la multiplicación correctamente.

1.3. Estructura del Trabajo

Este trabajo está estructurado en los siguientes capítulos:

- El capítulo 2, que incluye los conocimientos previos necesarios para comprender el trabajo desarrollado a lo largo de este TFM. En la sección 2.1 se realiza una introducción de las unidades de información empleadas en computación cuántica, los qubits, incluyendo algunas de sus características más importantes y algunas consideraciones relacionadas con la notación empleada para representar los qubits. En la sección 2.2 se explican los estados formados por varios qubits. La sección 2.3 realiza una introducción sobre las operaciones que se pueden realizar en computación cuántica, los requisitos que estas operaciones deben cumplir y algunas de las operaciones más importantes que se pueden realizar sobre uno o varios qubits. Por último se explica un importante algoritmo cuántico, la Transformada Cuántica de Fourier, en la sección 2.4. El trabajo desarrollado está basado, principalmente, en este algoritmo.

- El capítulo 3 realiza una breve introducción a los circuitos cuánticos y a los circuitos sumadores con los que trabajaremos en este proyecto.

- En el capítulo 4 se estudia el esquema de sumador cuántico basado en la Transformada Cuántica de Fourier propuesto en [Dra00]. Tomando este esquema como punto de partida, se proponen varias mejoras que incluyen la suma de más de dos qubits en la sección 4.2, la suma aritmética que se presenta en la sección 4.3 y la media aritmética y la suma ponderada, expuestas en la sección 4.5. Además, en la sección 4.6 discutiremos la representación de números enteros con signo.

- En el capítulo 5 se presenta un esquema de multiplicador basado en QFT. El capítulo incluye diagramas circuitales del multiplicador, así como una justificación matemática del funcionamiento del mismo.

- El TFM concluye con el capítulo 6, en el que se presentan las conclusiones extraídas a lo largo de este trabajo y posibles líneas de investigación futuras.

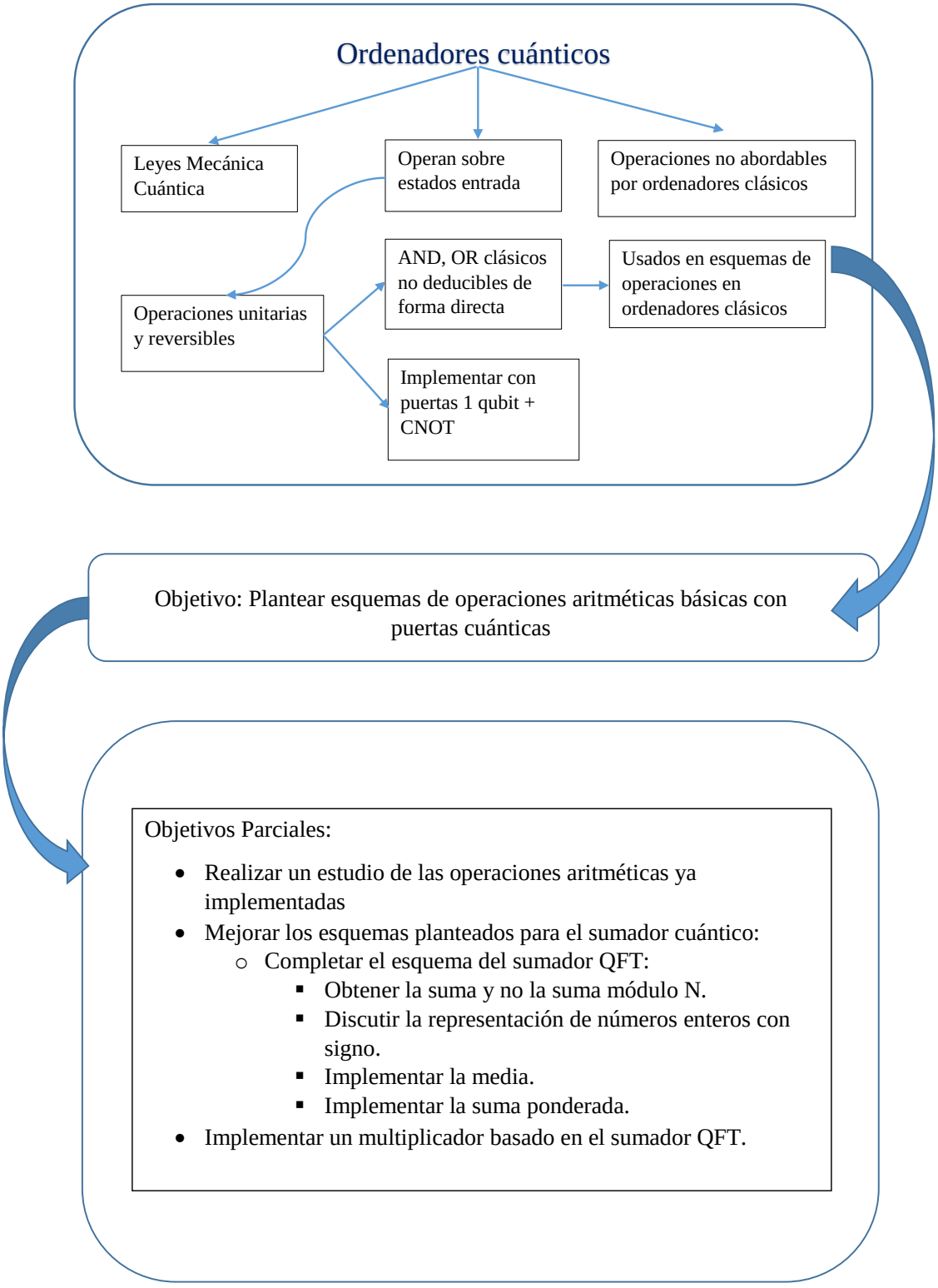


Figura 1.2: Contexto del TFM y objetivos.

Capítulo 2

Conocimientos Previos.

En el capítulo anterior hablamos de la necesidad de desarrollar algoritmos cuánticos que realicen operaciones aritméticas básicas, pues los algoritmos clásicos no siempre se pueden trasladar de forma directa al dominio cuántico. Dentro de las operaciones aritméticas nos hemos centrado en la suma y hemos revisado los algoritmos propuestos para ejecutar la suma en un ordenador cuántico. De estos esquemas hemos elegido trabajar sobre el sumador QFT propuesto en [Dra00]. Este esquema es óptimo en cuanto al número de qubits necesarios para codificar tanto los números que se desean multiplicar como el resultado de los mismos, pero deja sin discutir como calcular la llevada o la representación de números enteros con signo. Además, a partir de este esquema se pueden proponer otros circuitos que realicen operaciones como la suma de más de dos números, la media, la suma ponderada o la multiplicación.

Una vez establecidos tanto el contexto en el que se va a desarrollar la labor de investigación como los objetivos planteados para este TFM, debemos pasar a explicar el trabajo realizado. Pero antes es necesario explicar algunos conceptos básicos necesarios para entender dicho trabajo.

En este capítulo realizaremos una brevísima introducción a la computación cuántica explicando los conceptos básicos empleados a lo largo de este TFM. Hablaremos primero de la unidad de información empleadas en computación cuántica. Posteriormente explicaremos qué operaciones se pueden aplicar a las unidades de información e introduciremos algunas de las más empleadas en computación cuántica. Por último hablaremos de un algoritmo cuántico que computa la transformada cuántica de Fourier, pues en este algoritmo basaremos el trabajo desarrollado en el TFM.

2.1. Unidades de Información.

Si comparamos un ordenador clásico con un ordenador cuántico, encontraremos la primera diferencia en algo tan fundamental como la unidad de información. Como ya sabemos, un ordenador maneja la información en forma de bits, que pueden tomar dos valores, 0 y 1, tomando únicamente uno de estos valores en un momento dado. La información se codifica en cadenas con varios de estos bits, que será con lo que trabajen los or-

denadores. Para representar estas cadenas los ordenadores disponen de varios dispositivos físicos que solo se pueden encontrar en dos estados físicos no ambiguos y perfectamente distinguibles que se corresponderán con los estados 0 y 1. Éstos sistemas físicos pueden ser, por ejemplo, interruptores abiertos (0) o cerrados (1).

La unidad de información con la que se trabaja en computación cuántica es análoga al bit, pero se diferencia en que, en lugar de ser un sistema con dos posibles estados, es un sistema de dos niveles. Por ejemplo, tomemos como base el modelo atómico y supongamos que tenemos un átomo con un electrón que, a su vez, se puede hallar en dos estados de energía distintos. Sabemos que si le aplicamos luz con la energía adecuada y durante el periodo de tiempo apropiado, podemos hacer que el electrón cambie de órbita y pase, por ejemplo, del estado de reposo al estado excitado y viceversa. De la misma manera, pero reduciendo el tiempo de aplicación de la luz, podemos conseguir que el electrón se quede en un estado intermedio entre estos dos estados de energía [NC10] hallándose, de alguna manera, en ambos niveles al mismo tiempo. Si llamamos a estos dos posibles estados 0 y 1 o, como es más común al hablar de computación cuántica [Mer07], $|0\rangle$ y $|1\rangle$, la unidad de información podrá estar en el estado $|0\rangle$, en $|1\rangle$ o bien en un estado de superposición cuántica, es decir, un estado formado por la combinación lineal de ambos, de forma que:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

siendo α y β números complejos. Dicho de otra manera, el estado de esta unidad de información es un vector en un espacio vectorial bidimensional complejo. A esta unidad de información se le llama bit cuántico, qbit o qubit, que es la denominación más extendida.

Para denotar el estado de un sistema cuántico se emplea la notación de Dirac. Esta notación emplea la estructura $|\ \rangle$, llamada *ket*. Cuando se utiliza esta notación, los estados vienen representados como $|0\rangle, |1\rangle, |2\rangle ... |k\rangle$, donde $k \leq 2^n - 1$, siendo n el número de qubits del sistema. En ocasiones se emplea la expansión binaria de n dígitos. Por ejemplo, si empleamos la expansión binaria de $n = 3$ dígitos, el estado $|1\rangle$ también se puede representar utilizando la notación de Dirac como $|1\rangle_3$ o como $|001\rangle$. Por último, en ocasiones se emplearán letras para representar los qubits de forma simbólica. Así, los estados $|x\rangle$ e $|y\rangle$ representados de nuevo mediante notación de Dirac pueden representar, por ejemplo, el estado del qubit antes y después de que se le aplique una operación cuántica. A lo largo de este Trabajo de Fin de Máster trabajaremos con la representación simbólica de los estados, así como con el concepto de expansión binaria.

Si bien la notación de Dirac es una de las maneras más comunes de representar qubits, también se puede trabajar con ellos en forma de vectores columna. Los estados $|0\rangle$ y $|1\rangle$ se representan como vectores de la siguiente manera:

$$|0\rangle \longleftrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle \longleftrightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2.2}$$

Lo que implica que el estado general de un qubit (2.1) se puede expresar mediante el siguiente vector:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix}. \tag{2.3}$$

Para finalizar con las características de los qubits, vamos a comentar una particularidad de los mismos que no encuentra analogía en los bits clásicos. Mientras que en la computación clásica podemos examinar el bit para determinar si este se encuentra en el estado 0 o 1, no podemos realizar la misma operación para averiguar en qué estado cuántico se encuentran los qubits, es decir, no podemos calcular el valor de α y β al mismo tiempo. La operación que se emplea para examinar el qubit se denomina *medida*. Aunque más adelante hablaremos de la medida con más detalle, por ahora nos basta saber que si medimos el estado (2.3) en la base $(|0\rangle, |1\rangle)$, obtendremos o bien 0 con probabilidad $|\alpha|^2$, o bien 1 con probabilidad $|\beta|^2$, con $|\alpha|^2 + |\beta|^2 = 1$. Describiremos más características de la medida en las siguientes secciones del capítulo.

Una vez vistas las características de la unidad de información, podemos hablar de sistemas cuánticos compuestos de más de un qubit. Los sistemas de múltiples qubits se presentan en la siguiente sección.

2.2. Múltiples qubits.

Cuando un sistema cuántico está formado por más de un qubit decimos que éstos forman un registro cuántico, que se emplea para codificar información más compleja. En el caso de un registro formado por dos qubits, estos pueden encontrarse en los estados clásicos $|00\rangle, |01\rangle, |10\rangle$ y $|11\rangle$ o en una superposición cuántica de los cuatro estados:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \longleftrightarrow \begin{bmatrix}\alpha_{00}\\ \alpha_{01}\\ \alpha_{10}\\ \alpha_{11}\end{bmatrix}, \tag{2.4}$$

donde las amplitudes complejas deben cumplir la condición $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. De forma general, el estado de un registro de n qubits será la superposición normalizada de los 2^n estados clásicos y vendrá representado por la siguiente expresión:

$$|\psi\rangle = \sum_{0\leq x\leq 2^n-1} \alpha_x|x\rangle; \tag{2.5}$$

$$\sum_{0\leq x\leq 2^n-1} |\alpha_x|^2 = 1. \tag{2.6}$$

Este estado describe la situación en la que varios valores distintos del registro están presentes simultáneamente y, como con los qubits, no se encuentra un homólogo en computación clásica. En la ecuación (2.6), en lugar de representar cada estado por su expansión binaria, se representa por el valor decimal correspondiente. En el contexto de la computación cuántica, los 2^n estados clásicos, es decir, el conjunto de todos los posibles productos de los estados $|0\rangle$ y $|1\rangle$, se denomina base computacional.

Como en el caso de un único qubit, podremos representar el estado de un registro cuántico formado por n qubits (2.6) como vectores columna. De esta forma:

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}, \tag{2.7}$$

donde $N = 2^n$ representa el número de estados clásicos posibles.

La normalización de las amplitudes complejas hace que el estado del sistema sea un vector unitario en un espacio vectorial N-dimensional complejo. Este espacio vectorial, junto con el producto interno, forman un espacio de Hilbert.

Como en el caso del qubit sencillo, podemos aplicar la medida sobre los registros cuánticos de forma que si el sistema se encuentra en el estado $\sum_k \alpha_k|k\rangle$, donde k es un número decimal, obtendremos como resultado $|k\rangle$ con probabilidad $|\alpha_k|^2$. Pero, en este caso, puede ser que solo se quiera medir una parte de los qubits. Volvamos al ejemplo de dos qubits, donde el estado del registro viene representado por la ecuación (2.4). Si queremos medir el primer qubit, nos dará como resultado 0 con probabilidad $|\alpha_{00}|^2 + |\alpha_{01}|^2$. Dado que después de medir se debe cumplir la condición de normalización, es decir, $\sum_k |\alpha_k|^2 = 1$, el estado resultante será:

$$|\psi'\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}, \tag{2.8}$$

por lo que queda renormalizado por el factor $\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}$.

Otra característica de los estados es la fase. Si por ejemplo tenemos el estado $e^{i\theta}|\phi\rangle$, donde $|\phi\rangle$ es un vector de estado y θ un número real, se puede decir que los estados $e^{i\theta}|\phi\rangle$ y $|\phi\rangle$ son el mismo. Esto es debido a que la probabilidad de encontrar $|\phi\rangle$ y $e^{i\theta}|\phi\rangle$ es la misma. La fase $e^{i\theta}$ recibe el nombre de fase global y se suele ignorar al no afectar al estado desde el punto de vista de la medida [NC10].

Existe otro tipo de fase conocida como fase relativa. Para entender su relevancia lo mejor es ver un ejemplo con los estados:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{2.9}$$

La amplitud de $|0\rangle$ es la misma en ambos estados. La amplitud de $|1\rangle$ es igual en ambos estados salvo por un signo negativo, o una fase relativa $e^{i\pi}$. Si medimos en la base $(|0\rangle, |1\rangle)$, la fase relativa no varía la probabilidad de medir el estado $|1\rangle$. Pero si trabajamos

en la base $(|+\rangle, |-\rangle)$ donde, como ya vimos en (2.40):

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}; \tag{2.10}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.11}$$

la fase relativa implica que los estados, en esa base, ya no son equivalentes. Es decir, una fase relativa puede modificar la amplitud de un estado con respecto a otro y, en función de la base en la que se mida el qubit, puede implicar diferencias en la probabilidad de medida. Como consecuencia, los estados ya no son físicamente equivalentes [NC10].

Para finalizar, vamos a realizar una última analogía entre los bits clásicos y los qubits con el fin de explicar una última característica de las unidades de información usadas en computación cuántica. Como ya sabemos, dos bits clásicos se pueden encontrar en los estados $00, 01, 10$ y 11. Si representamos estos estados mediante vectores columna, como hacemos con los qubits en la expresión (2.2), obtendríamos que los posibles estados en los que se pueden hallar los dos bits son $|00\rangle, |01\rangle, |10\rangle$ y $|11\rangle$ que, en realidad, es una forma abreviada de representar el producto tensorial de los estados de los bits clásicos expresados mediante vectores columna. Si usamos una notación matemática más formal, estos estados vendrían expresados de la siguiente manera [Mer07]:

$$|0\rangle \otimes |0\rangle;\ |0\rangle \otimes |1\rangle;\ |1\rangle \otimes |0\rangle;\ |1\rangle \otimes |1\rangle. \tag{2.12}$$

Supongamos ahora que tenemos dos qubits, uno en el estado $|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ y el otro en el estado $|\theta\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, separados el uno del otro. Podemos generar un estado compuesto juntando ambos qubits. Usando una generalización del caso anterior, el estado del par se puede tomar como el producto tensorial de los estados de cada uno de los qubits [Mer07], es decir:

$$\begin{aligned} |\psi\rangle &= |\phi\rangle \otimes |\theta\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes |(\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle. \end{aligned} \tag{2.13}$$

Se debe cumplir que la suma de las amplitudes al cuadrado sea 1, es decir:

$$|\alpha_1 \cdot \beta_1|^2 + |\alpha_1 \cdot \beta_2|^2 + |\alpha_2 \cdot \beta_1|^2 + |\alpha_2 \cdot \beta_2|^2 = 1. \tag{2.14}$$

En este caso hay que tener en cuenta que el estado general de un sistema formado por dos qubits es de la forma (2.13) si y solo si $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$. Dado que la única condición que deben cumplir las amplitudes es la expresada en las ecuaciones (2.6) y (2.14), esta igualdad no tiene por qué cumplirse, por lo que, en general, y al contrario de lo que sucede con los bits, no siempre se podrá expresar el estado de un sistema de n qubits como el producto de los estados de cada uno de los qubits. Los qubits, como parte de un registro de n qubits, no estarán caracterizados por un estado definido sino que su estado dependerá de los estados del resto de qubits que forman el sistema.

Cuando el estado de un sistema compuesto no puede describirse como el producto tensor de los estados de los qubits que componen el sistema, se dice que dicho sistema está en un estado de entrelazamiento cuántico. Algunos estados entrelazados populares son los estados de Bell:

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \tag{2.15}$$

$$|\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}; \tag{2.16}$$

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}; \tag{2.17}$$

$$|\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}. \tag{2.18}$$

$$\tag{2.19}$$

Los estados entrelazados son clave en algunos protocolos de comunicación cuántica como la teleportación cuántica o la codificación superdensa.

2.3. Operaciones.

Al igual que cualquier operación clásica se puede expresar como una secuencia de operaciones lógicas sobre uno o varios bits, cualquier operación cuántica se puede expresar como una secuencia de puertas lógicas cuánticas que actúan sobre uno o varios qubits. En esta sección vamos a describir algunas características de estas operaciones, así como las puertas más importantes.

2.3.1. Puertas cuánticas que actúan sobre un qubit.

Las operaciones que un ordenador cuántico puede realizar sobre un qubit son aquellas transformaciones lineales que toman vectores unitarios y los transforman en vectores unitarios. Tomemos como ejemplo la puerta lógica NOT, que en computación clásica transforma un bit 0 en 1 y viceversa. En computación cuántica se define la puerta NOT que actúa linealmente sobre el estado del qubit, tomando el estado

$$\alpha|0\rangle + \beta|1\rangle, \tag{2.20}$$

y transformándolo en el estado

$$\alpha|1\rangle + \beta|0\rangle. \tag{2.21}$$

Para ver cómo actúa la puerta NOT sobre el estado de un qubit, vamos a trabajar con matrices. Definimos la matriz X, que representa la puerta cuántica NOT, como:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{2.22}$$

Si el estado del qubit en notación vectorial es:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2.23}$$

siendo α la amplitud correspondiente a $|0\rangle$ y β la amplitud correspondiente a $|1\rangle$, la salida de la puerta cuántica NOT será:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}. \tag{2.24}$$

Es decir, transforma el estado del qubit (2.20) en el estado (2.21). Se deduce, por tanto, que las puertas cuánticas que actúan sobre un único qubit se pueden representar mediante matrices 2×2 [NC10]. Ahora solo falta definir qué matrices pueden constituir una puerta cuántica válida y, dado que el estado de un qubit debe cumplir la condición de normalización, las puertas cuánticas deben operar de tal manera que el resultado también cumpla la condición de normalización, es decir, que el estado de salida $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ cumpla que $|\alpha'|^2 + |\beta'|^2 = 1$, lo que implica que la matriz debe ser unitaria. Si U representa una puerta cuántica genérica, debe cumplir que:

$$UU^\dagger = U^\dagger U = I. \tag{2.25}$$

Por tanto, cualquier transformada unitaria constituiría una puerta cuántica válida y puesto que toda transformación unitaria cuenta con una transformación unitaria inversa, resulta que estas operaciones son reversibles.

Algunas puertas interesantes que actúan sobre un qubit y que podemos ver en la imagen 2.1, son la puerta Y, que transforma $\alpha|0\rangle + \beta|1\rangle \to -\beta i|0\rangle + \alpha|1\rangle$:

$$Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \tag{2.26}$$

y la puerta Z, que cambia el signo de $|1\rangle$ y deja el signo de $|0\rangle$ sin alterar:

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.27}$$

Las puertas X, Y y Z, junto con la matriz identidad I, reciben el nombre de matrices de Pauli [NC10].

Matrices de Pauli

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{X} \longrightarrow \beta|0\rangle + \alpha|1\rangle$$

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{Y} \longrightarrow \beta i|0\rangle + \alpha|1\rangle$$

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{Z} \longrightarrow \alpha|0\rangle - \beta|1\rangle$$

Figura 2.1: Matrices de Pauli sobre un qubit.

A lo largo de este Trabajo de Fin de Máster trabajaremos también con la puerta (o transformada) Hadamard (figura 2.2), que viene definida por:

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.28}$$

Esta puerta transforma el estado $|0\rangle$ en $(|0\rangle+|1\rangle)/\sqrt{2}$ y el estado $|1\rangle$ en $(|0\rangle-|1\rangle)/\sqrt{2}$.

Transformada de Hadamard

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{H} \longrightarrow \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Figura 2.2: Transformada de Hadamard sobre el estado $\alpha|0\rangle + \beta|1\rangle$.

La última puerta con la que trabajaremos es la puerta de rotación o desplazamiento de fase. Esta puerta realiza la transformación $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle + \beta e^{i\theta}|1\rangle$, es decir, añade una fase a $|1\rangle$ mientras que deja al estado $|0\rangle$ sin alterar. Esta puerta no altera la probabilidad de medir $|0\rangle$ o $|1\rangle$, pero modifica la fase relativa entre los estados $|0\rangle$ y $|1\rangle$. La matriz que representa a esta puerta es:

$$R_\theta \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}. \tag{2.29}$$

Puerta de Rotación de Fase

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{R_\theta} \longrightarrow \alpha|0\rangle + \beta e^{i\theta}|1\rangle$$

Figura 2.3: Puerta de rotación de fase actuando sobre $\alpha|0\rangle + \beta|1\rangle$.

2.3.2. Puertas cuánticas que actúan sobre múltiples qubits.

De la misma manera que tenemos puertas que actúan sobre un qubit, podemos tener puertas que actúen sobre n qubits con la condición de que sean operaciones unitarias reversibles, que tengan tantos qubits de entrada como qubits de salida. Sin embargo, a la hora de diseñar algorimos cuánticos, las transformaciones unitarias que se emplean habitualmente están restringidas a aquellas que se pueden construir como el producto de un número finito de puertas que actúan sobre uno o dos qubits. La razón es que construir puertas para uno o dos qubits es más sencillo que construir puertas para más qubits. Esto hace que operaciones que podrían describirse mediante una única matriz unitaria, es decir, como una única puerta, acaben construyéndose como un conjunto finito de más de una puerta cuántica. El número de puertas empleadas en la implementación de los circuitos será uno de los criterios de bondad que emplearemos a lo largo de este Trabajo de Fin de Máster.

Un ejemplo de puerta que actúa sobre más de un qubit es las puerta swap. Estas son operaciones que intercambian los estados entre qubits. Están definidas por la matriz:

$$SWAP \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.30}$$

Otro tipo de puertas que actúan sobre más de un qubit son las puertas controladas. Estas puertas toman como entrada un qubit de control y un qubit "objetivo", y transforman el estado del qubit objetivo en función del estado del qubit de control, de forma que si el qubit de control es $|0\rangle$ el estado del qubit objetivo queda inalterado, mientras que si el qubit de control es $|1\rangle$ entonces la puerta actuará sobre el qubit objetivo y su estado cambiará.

Si definimos la puerta U que actúa sobre un qubit como

$$U \equiv \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix}, \tag{2.31}$$

La puerta U controlada que actúa sobre dos qubits de tal manera que el primero será el bit de control realizará la siguiente operación:

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle; \\ |01\rangle &\rightarrow |01\rangle; \\ |10\rangle &\rightarrow |1\rangle U|0\rangle = |1\rangle \left(x_{00}|0\rangle + x_{10}|1\rangle\right); \\ |11\rangle &\rightarrow |1\rangle U|0\rangle = |1\rangle \left(x_{01}|0\rangle + x_{11}|1\rangle\right); \end{aligned} \tag{2.32}$$

La matriz unitaria que representa a la puerta U controlada será

$$cU \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{01} \\ 0 & 0 & x_{10} & x_{11} \end{bmatrix}, \tag{2.33}$$

Puerta U controlada

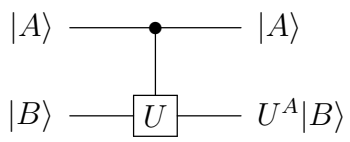


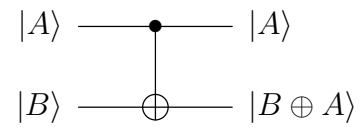
Figura 2.4: Puerta U controlada.

El ejemplo típico de puerta controlada que actúa sobre dos qubits es la puerta *controlled*-NOT o cNOT. Dado que la puerta NOT cambia $|0\rangle \rightarrow |1\rangle$ y $|1\rangle \rightarrow |0\rangle$, la puerta controlada cambiará el estado del qubit objetivo si el qubit de control es $|1\rangle$ y lo dejará inalterado si el qubit objetivo es $|0\rangle$, como se indica en (2.34).

$$|00\rangle \rightarrow |00\rangle;\ |01\rangle \rightarrow |01\rangle;\ |10\rangle \rightarrow |11\rangle;\ |11\rangle \rightarrow |10\rangle. \tag{2.34}$$

En la imagen 2.5 vemos la representación matricial y un diagrama de circuito de esta puerta. Otra manera de ver esta operación es como la generalización de la puerta lógica XOR empleada en computación clásica porque, como vemos en la imagen, su acción se puede resumir como $|A, B\rangle \rightarrow |A, A \oplus B\rangle$, donde $\oplus$ es la suma módulo dos, es decir, la operación lógica XOR [NC10]. La importancia de esta puerta radica en que cualquier puerta cuántica que actúa sobre n qubits se puede descomponer en un conjunto de puertas cNOT y de puertas para un qubit [NC10].

Puerta NOT controlada



$$cNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figura 2.5: Puerta cNOT [NC10].

Otra puerta que utilizaremos a lo largo del Trabajo de Fin de Máster es la puerta de rotación de fase controlada. Esta puerta cambiará la fase del qubit objetivo si el qubit de control es $|1\rangle$, y lo dejará inalterado si el qubit de control es $|0\rangle$. La matriz que representa a esta puerta es

$$cR_\theta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & e^{i\theta} \end{bmatrix} \tag{2.35}$$

Se pueden definir puertas controladas que actúen sobre tres qubits, de tal manera que dos de ellos actuarán como qubits de control, mientras que el tercero será el qubit objetivo. Esta puerta solo altera el estado del qubit objetivo si los dos qubits de control son $|1\rangle$. En caso contrario, el estado del qubit objetivo permanece inalterado.

Puerta U controlada

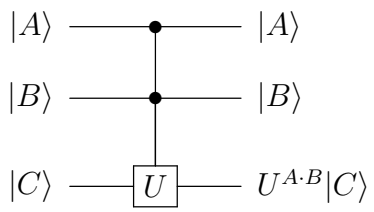


Figura 2.6: Puerta U que actúa sobre tres qubits.

La puerta de este tipo más importante es la puerta Toffoli, cuya representación circuital se puede ver en la figura 2.7. Como vemos, es una puerta NOT con dos controles, por lo que esta puerta cambiará el estado del qubit objetivo de $|0\rangle$ a $|1\rangle$ o de $|1\rangle$ a $|0\rangle$ si los dos qubits de control son $|1\rangle$ [NC10]. Esta puerta es importante en computación cuántica ya que cualquier circuito clásico puede ser reemplazado por un circuito cuántico equivalente construido con puertas Toffoli. La representación matricial de esta puerta se da en la expresión (2.36).

Puerta Toffoli

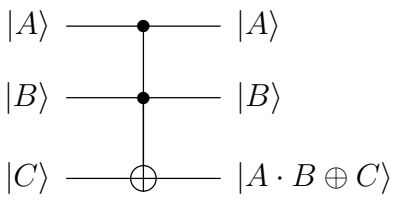


Figura 2.7: Puerta Toffoli.

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.36}$$

En este Trabajo de Fin de Máster utilizaremos la puerta de rotación de fase controlada por dos qubits, $|a\rangle$ y $|b\rangle$, de tal manera que transformará $|c\rangle \rightarrow (e^{i\theta})^{a\cdot b}|c\rangle$. Es decir, des-

plazará la fase del qubit solo cuando $|a\rangle = |b\rangle = |1\rangle$. La matriz que representa esta puerta será

$$ccR_\theta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\theta} \end{bmatrix}. \tag{2.37}$$

2.3.3. Medida

La última operación que realizaremos sobre qubits a lo largo de este Trabajo de Fin de Máster es la medida, que ya hemos introducido al describir las características de estas unidades de información. Para especificar el estado de un qubit debemos especificar las amplitudes α y β y, al contrario de lo que sucede con los bits clásicos, no lo podemos hacer con solo "mirar", es decir, no podemos extraer la información contenida en las amplitudes α_x de un conjunto de n qubits como se hace con los bits clásicos.

La herramienta de que se dispone en computación cuántica para extraer la información de los qubits es la medida. Por ejemplo, si medimos un único qubit que se encuentre en el estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, el resultado que tendremos será 0 con probabilidad $|\alpha|^2$ o 1 con probabilidad $|\beta|^2$. Extendido a un registro de n qubits [Mer07], la probabilidad $p(x)$ de medir el estado x vendrá dada por:

$$|\psi\rangle = \sum_{0 \leq x \leq 2^n - 1} \alpha_x |x\rangle \rightarrow p(x) = |\alpha_x|^2. \tag{2.38}$$

La medida es la única operación irreversible que se puede aplicar a los qubits ya que una vez medido, el qubit toma el estado resultante y no se puede recuperar el estado anterior. En el ejemplo de un qubit, si el resultado es 0, el qubit toma el estado $|0\rangle$ y se pierde tanto la información almacenada en la amplitud β como la posibilidad de volver al estado anterior. Además, es una operación no lineal.

Para medir los qubits se pueden utilizar distintas bases. Por ejemplo, para medir un único qubit se suele utilizar la base $\{|0\rangle, |1\rangle\}$ pero, dependiendo del número de qubits o de la información que se desee extraer, a veces se utilizan otras, como por ejemplo la base $\{|+\rangle, |-\rangle\}$ siendo

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}; \tag{2.39}$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.40}$$

¿Cómo podemos emplear esta base para obtener información sobre el estado de un único qubit? Pues podemos transformar el estado de un qubit de la siguiente manera:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha\frac{|+\rangle + |-\rangle}{\sqrt{2}} + \beta\frac{|+\rangle - |-\rangle}{\sqrt{2}} = \frac{\alpha+\beta}{\sqrt{2}}|+\rangle + \frac{\alpha-\beta}{\sqrt{2}}|-\rangle. \quad (2.41)$$

De esta manera, obtendremos como resultado $+$ con probabilidad $|\alpha+\beta|^2/2$ o $-$ con probabilidad $|\alpha-\beta|^2/2$ y los estados posibles después de medir serán $|+\rangle$ o $|-\rangle$. De manera más general, si tenemos cualquier base $\{|a\rangle, |b\rangle\}$, podemos expresar el estado del qubit como una combinación lineal de esos estados $\alpha|a\rangle + \beta|b\rangle$ y, si esa base es ortonormal, entonces podemos medir el qubit sobre esa base, dando como resultado a con probabilidad $|\alpha|^2$ o b con probabilidad $|\beta|^2$. Que los estados de la base sean ortornomales es necesario para que $|\alpha|^2 + |\beta|^2 = 1$ [NC10].

2.4. Transformada Cuántica de Fourier

Para finalizar el capítulo de conocimientos previos introduciremos una operación cuántica que se empleará en numerosas ocasiones a lo largo de este Trabajo de Fin de Máster, la transformada cuántica de Fourier.

Para explicar esta operación vamos a comenzar recordando su equivalente en computación cuántica, la transformada discreta de Fourier o DFT. Como sabemos, la DFT es un operador lineal que transforma un conjunto de muestras de una función en el tiempo en los coeficientes que, al ser aplicados a sinusoides complejas ordenadas por su frecuencia, nos dan el espectro muestreado de la función en el dominio de la frecuencia. La DFT opera sobre un vector complejo $x_0, \ldots, x_{N-1}$ con N elementos y lo transforma en otro vector complejo $y_0, \ldots, y_{N-1}$ de la siguiente manera [NC10]:

$$y_x \to \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}}. \quad (2.42)$$

Pues la transformada cuántica de Fourier o QFT (Quantum Fourier Transform), es una operación análoga a la DFT. Aplicada sobre una base ortonormal $|0\rangle, \ldots, |N-1\rangle$, la QFT es un operador lineal que realiza la siguiente transformación cuando actúa sobre un qubit $|j\rangle$ en la base ortonormal[NC10]:

$$|j\rangle \to \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle. \quad (2.43)$$

De forma equivalente, el efecto de la transformada sobre un qubit arbitrario se puede escribir como:

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_j |k\rangle, \quad (2.44)$$

donde las amplitudes y_k componen la transformada discreta de Fourier de las amplitudes x_j.

Veamos cómo construir un circuito formado por puertas cuánticas que realice la QFT de un qubit $|j\rangle$. Para ello vamos a realizar las siguientes consideraciones:

- Tomamos $N = 2^n$, donde n es un número entero. La base $|0\rangle, \ldots, |2^n - 1\rangle$ es, entonces, la base computacional de n qubits.
- Trabajaremos con la expansión binaria de j, es decir, $j = j_1 j_2 \ldots j_n$, de tal manera que $j = j_1 2^{n-1} + j_2 2^{n-2} + \ldots + j_n 2^0$.
- También trabajaremos con la notación $0.j_l j_{l+1} \ldots j_m$ para representar la fracción binaria $j_l/2^1 + j_{l+1}/2^2 + \ldots + j_m/2^m$.

Operando sobre la expresión (2.43), podemos transformar el sumatorio en una representación en forma de producto de la siguiente manera:

$$|j\rangle \rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i j k/2^n} |k\rangle \tag{2.45}$$

$$= \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \left(\sum_{k_l=1}^{n} k_l 2^{-l}\right)} |k_1 \ldots k_n\rangle \tag{2.46}$$

$$= \frac{1}{2^{\frac{n}{2}}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \left(\bigotimes_{l=1}^{n} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right) \tag{2.47}$$

$$= \frac{1}{2^{\frac{n}{2}}} \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2\pi i j k_l 2^{-l}} |k_l\rangle \right] \tag{2.48}$$

$$= \frac{1}{2^{\frac{n}{2}}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i j 2^{-l}} |1\rangle \right] \tag{2.49}$$

$$= \frac{1}{2^{\frac{n}{2}}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i \frac{j}{2^l}} |1\rangle \right] \tag{2.50}$$

$$= \frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_2\ldots j_{n-1}j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_1 j_2\ldots j_{n-1}j_n}|1\rangle\right)}{2^{\frac{n}{2}}}, \tag{2.51}$$

donde $\otimes$ representa el producto tensorial. Se puede construir un circuito cuántico que realize la operación de la expresión (2.51). Para ello necesitaremos definir la siguiente puerta de rotación de fase:

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}. \tag{2.52}$$

Para la construcción del circuito también necesitaremos emplear puertas de Hadamard. Esta puerta realizará la siguiente transformación en un qubit en la base computacional:

$$|j_i\rangle \rightarrow \frac{|0\rangle + e^{2\pi i 0.j_i}|1\rangle}{\sqrt{2}}, \tag{2.53}$$

de tal manera que si $j_i = 0$, $e^{2\pi i 0.j_i} = 1$ y si $j_i = 1$, $e^{2\pi i 0.j_i} = -1$.

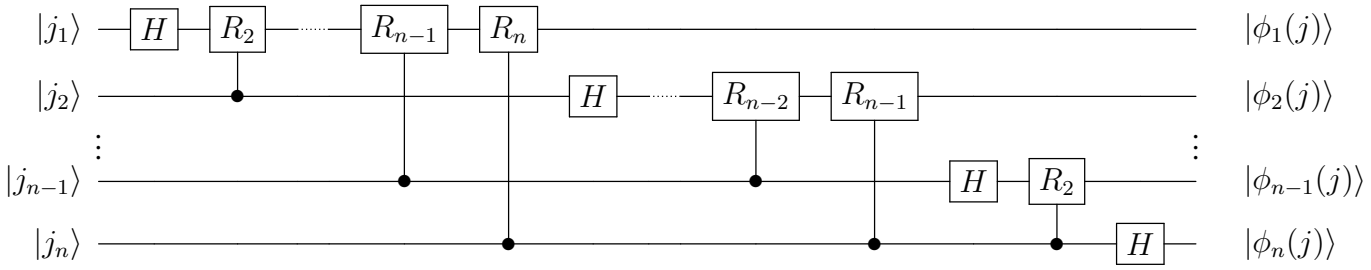


Figura 2.8: Transformada Cuántica de Fourier.

Teniendo en cuenta estas consideraciones, construimos el circuito que se muestra en la figura 2.8. Veamos como funciona el circuito, para lo cual comenzaremos estudiando cómo evoluciona el qubit $|j_1\rangle$.

La primera operación que se realiza sobre $|j_1\rangle$ es aplicar una puerta de Hadamard. Teniendo en cuenta la expresión (2.53), el qubit se transforma en

$$|j_1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i \frac{j_1}{2^1}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1}|1\rangle\right). \tag{2.54}$$

Posteriormente se le aplica una puerta de rotación de fase R_2, controlada por $|j_2\rangle$. Esto implica que se va a aplicar la rotación de fase

$$R_2 \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^2}} \end{bmatrix}, \tag{2.55}$$

siempre y cuando el qubit de control, $|j_2\rangle$, sea igual a 1. El resultado de aplicar esta operación será:

$$|j_1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_1}{2^1}+\frac{j_2}{2^2}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}|1\rangle\right). \tag{2.56}$$

Continuamos aplicando puertas de rotación $R_3, R_4, \ldots R_n$ controladas por los qubits $|j_3\rangle, |j_4\rangle, \ldots, |j_n\rangle$ respectivamente. El resultado final será:

$$|j_1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_1}{2^1}+\frac{j_2}{2^2}+\frac{j_3}{2^3}+\ldots+\frac{j_n}{2^n}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 j_3 \ldots j_n}|1\rangle\right), \tag{2.57}$$

y el circuito habrá producido el estado

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 j_3 \ldots j_n}|1\rangle\right)|j_2 \ldots j_n\rangle. \tag{2.58}$$

A continuación, se ejecuta un proceso similar sobre el siguiente qubit, $|j_2\rangle$. Primero aplicamos la transformada de Hadamard:

$$|j_2\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i \frac{j_2}{2^1}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_2}|1\rangle\right). \tag{2.59}$$

A continuación se aplica una puerta R_2 controlada por el qubit $|j_3\rangle$:

$$|j_2\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_2}{2^1}+\frac{j_3}{2^2}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_2 j_3}|1\rangle\right). \tag{2.60}$$

De nuevo, se aplican puertas de rotación de fase controladas R_3, R_4, hasta R_{n-1}. Estas puertas estarán controladas por los qubits $|j_4\rangle, |j_5\rangle, \ldots, |j_n\rangle$ respectivamente. El resultado de este proceso será:

$$|j_2\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{j_2}{2^1}+\frac{j_3}{2^2}+\frac{j_4}{2^3}+\ldots+\frac{j_n}{2^{n-1}}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_2 j_3 j_4 \ldots j_n}|1\rangle\right), \tag{2.61}$$

mientras que el circuito habrá producido el estado

$$\frac{1}{\sqrt{2^2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 j_3 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 j_3 j_4 \ldots j_n}|1\rangle\right)|j_3 \ldots j_n\rangle. \tag{2.62}$$

El circuito opera de la misma manera sobre el resto de los qubits hasta llegar a $|j_n\rangle$, sobre el que solo aplica una transformada de Hadamard. Este qubit quedará transformado según la expresión:

$$|j_n\rangle \to \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i \frac{j_n}{2^1}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right) \tag{2.63}$$

y el circuito habrá producido el estado final

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 j_3 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 j_3 j_4 \ldots j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right), \tag{2.64}$$

Por último se deben aplicar puertas swap. Estas puertas intercambian los estados entre dos qubits. Las puertas actuarán sobre los pares $(|\phi_1(j)\rangle, |\phi_n(j)\rangle)$, $(|\phi_2(j)\rangle, |\phi_{n-1}(j)\rangle)$, y así sucesivamente hasta llegar a $\left(\left|\phi_{\frac{n}{2}}(j)\right\rangle, \left|\phi_{\frac{n}{2+1}}(j)\right\rangle\right)$.

El resultado será:

$$\begin{aligned}
|j_1\rangle &\to \frac{1}{2}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right); \\
|j_2\rangle &\to \frac{1}{2}\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right); \\
&\vdots \\
|j_{n-1}\rangle &\to \frac{1}{2}\left(|0\rangle + e^{2\pi i 0.j_2 \ldots j_{n-1}j_n}|1\rangle\right); \\
|j_n\rangle &\to \frac{1}{2}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_{n-1}j_n}|1\rangle\right),
\end{aligned} \tag{2.65}$$

produciendo el estado final

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_2\ldots j_{n-1}j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_1 j_2\ldots j_{n-1}j_n}|1\rangle\right), \tag{2.66}$$

que coincide con la expresión (2.51). Además, se demuestra que la QFT es una operación unitaria, pues todas las operaciones empleadas en la implementación del algoritmo son, a su vez, unitarias.

La QFT es clave en numerosos algoritmos cuánticos como el algoritmo de estimación de fase. En este Trabajo de Fin de Máster, emplearemos la QFT para realizar operaciones aritméticas como la suma o la multiplicación en el dominio transformado.

Capítulo 3

Operaciones Aritméticas

En el capítulo anterior hemos introducido algunos conceptos básicos sobre la computación cuántica. En este capítulo vamos a introducir los circuitos cuánticos, que es el modelo seguido para explicar el proceso de computación en un ordenador cuántico y con el que vamos a trabajar para diseñar algoritmos que realicen operaciones aritméticas.

3.1. Circuitos Cuánticos.

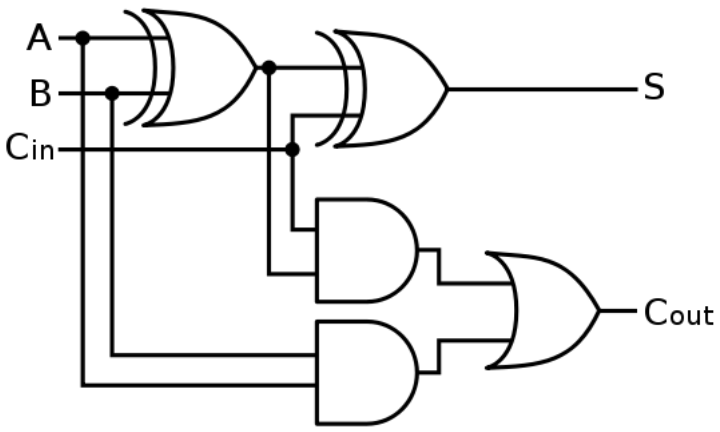


Figura 3.1: Sumador clásico.

El modelo de circuitos cuánticos es el lenguaje que se utiliza para describir algoritmos cuánticos. Estos son conjuntos de componentes conectados entre sí que describen un proceso computacional [NC10]. Es un concepto análogo a representar un proceso computacional en un ordenador clásico mediante un circuito compuesto de puertas lógicas como la puerta AND o la puerta OR.

Los circuitos cuánticos van a actuar sobre estados. Estos estados estarán representados por n qubits que constituirán las entradas del circuito. El espacio de estados será un

espacio complejo de Hilbert de dimensión 2^n. $|x\rangle$ será el estado y estará representado por los qubits $|x_1\rangle, \ldots |x_n\rangle$ de tal manera que:

$$|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes, \ldots, \otimes |x_n\rangle. \tag{3.1}$$

Si además $x_i = 0, 1$, estos qubits se conocen como estados en la base computacional [NC10]. A lo largo de este TFM trabajaremos con estados en la base computacional.

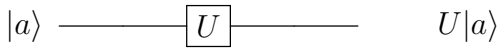


Figura 3.2: Puerta cuántica.

Una vez que tenemos preparados el conjunto de qubits sobre el que queremos operar, el siguiente paso es aplicar puertas cuánticas. En el capítulo anterior vimos que las operaciones válidas que se pueden aplicar sobre los qubits son aquellas operaciones reversibles que se pueden describir mediante una matriz unitaria. Representaremos estas operaciones mediante puertas cuánticas, como vemos en la imagen 3.2.

Estas puertas estarán conectadas entre sí mediante cables. Los cables no tienen por qué corresponderse con cables físicos, sino que pueden representar el paso del tiempo, o una partícula física como puedan ser los fotones moviéndose de un punto a otro en el espacio [NC10].

Por último, los circuitos suelen incluir una medida en la base computacional para extaer información de los qubits. Pero en este TFM dibujaremos circuitos que no incluyen esas puertas ya que no son necesarias para explicar el trabajo que se está realizando.

Es importante hacer notar que las puertas cuánticas tienen que tener al menos tantos estados de salida como estados de entrada. De no ser así, la operación no será reversible. Por este motivo, algunos algoritmos clásicos para realizar operaciones cuánticas como la suma no pueden ser trasladados al dominio cuántico de forma directa. Si observamos la figura 3.1, que muestra un sumador con llevada, vemos que este cuenta con tres entradas, A, B y C y dos salidas $A + B$ y C. Si diseñáramos un circuito cuántico con ese número de estados de entrada y salida, perderíamos la información de uno de los estados y la operación no se podría revertir.

En cambio, se puede diseñar un sumador como el mostrado en la figura 3.3. Este sumador tiene dos estados de entrada, $|a\rangle$ y $|b\rangle$, y dos estados de salida, $|a\rangle$ y $|a + b\rangle$. Con esa información de salida, podremos realizar una operación que nos permita recuperar los $|a\rangle$ y $|b\rangle$ originales. Y si queremos calcular la llevada, podemos diseñar un circuito como el mostrado en la figura 3.4, en el que se añade el estado $|c\rangle$ para incorporar este término.

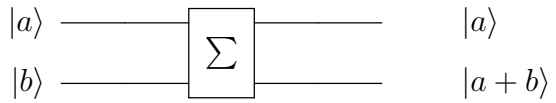


Figura 3.3: Sumador $a + b$.

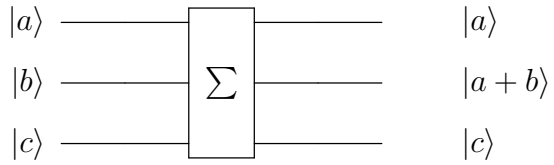


Figura 3.4: Sumador $a + b$.

3.1.1. Sumador Cuántico

Teniendo en cuenta las restricciones que se deben observar a la hora de diseñar circuitos cuánticos, es necesario crear algoritmos que puedan ejecutar la suma en un ordenador cuántico, habida cuenta que el algoritmo clásico no se puede trasladar de forma directa.

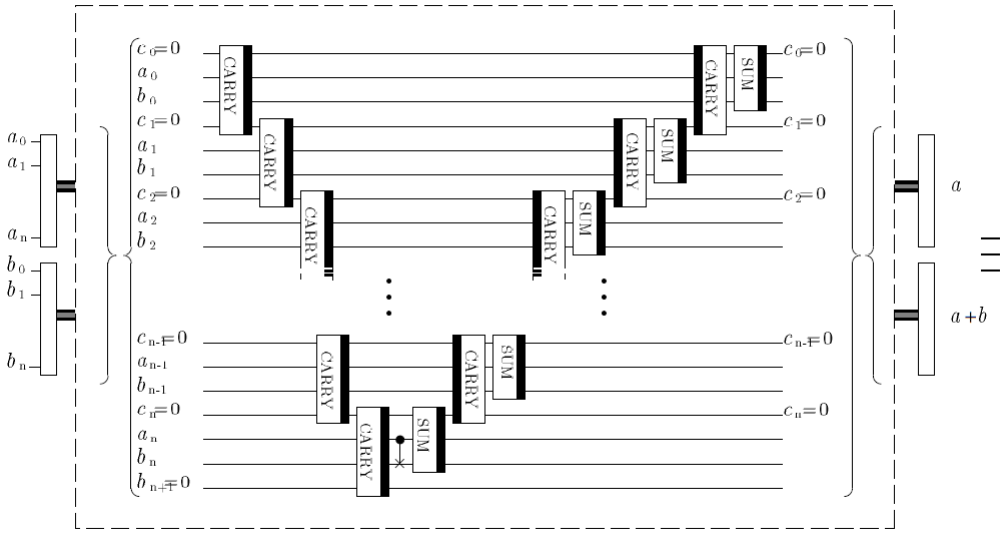


Figura 3.5: Sumador [VBE96].

Uno de los primeros esquemas creados para dar respuesta a esta necesidad es el presentado en [VBE96]. Este esquema suma los números a y b. Para ello los representa mediante n qubits $|a_1\rangle, \ldots, |a_n\rangle$ y $|b_1\rangle, \ldots, |b_n\rangle$. El esquema calcula primero todos las llevadas hasta que el último bloque calcula el dígito más significativo del resultado. Para ello emplea n qubits $|c_1\rangle, \ldots, |c_n\rangle$. Posteriormente deshace las operaciones y calcula la suma bit a bit. hace la suma bit a bit del bit menos significativo hasta el bit más significativo. La figura 3.5 muestra el esquema del sumador. Los bloques con el borde negro a la izquierda representa la secuencia de puertas que forman el bloque pero en sentido inverso.

Para realizar la suma define dos bloques, un sumador y uno de llevada, compuestas de puertas de Toffoli y puertas XOR. El esquema del bloque sumador se muestra en la figura 3.6, y el esquema del bloque que calcula la llevada se muestra en 3.7.

El sumador necesita $3n$ qubits para realizar la suma de dos números de n bits. Además el bloque sumador y el bloque de llevada emulan las puertas AND y OR. Es por ello que se

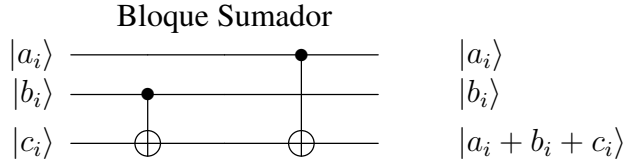


Figura 3.6: Sumador $a + b$.

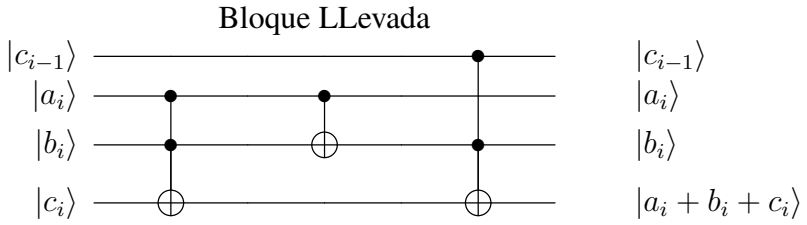


Figura 3.7: Sumador $a + b$.

dice que está inspirado en el algoritmo clásico, pues intenta emular ese esquema haciendo uso de las operaciones permitidas en computación cuántica.

Pero es posible diseñar un circuito que haga uso de operaciones y algoritmos cuánticos para calcular la suma. Sobre este circuito basaremos nuestro trabajo, que presentaremos en los capítulos siguientes.

Capítulo 4

Sumadores con QFT

Tradicionalmente, los algoritmos para implementar la suma diseñados para ordenadores cuánticos siguen los esquemas implementados para ordenadores clásicos, con las correspondientes extensiones que permitan realizar cálculos reversibles. Sin embargo, se pueden diseñar algoritmos cuánticos para implementar operaciones aritméticas sin necesidad de copiar el esquema clásico del algoritmo.

En [Dra00] se presenta un sumador cuántico que hacer uso de la transformada cuántica de Fourier (en adelante, QFT). Este sumador reduce el número de qubits necesarios para realizar la suma al no utilizar bits para la llevada. El esquema presentado favorece además la paralelización en la ejecución de la suma y permite sumar un número clásico a una superposición cuántica sin codificar el número clásico en un registro cuántico.

Partiendo de este esquema, propondremos varias mejoras que incluyen la suma de más de dos qubits, la suma, la media aritmética y la suma ponderada. Además discutiremos la representación de números enteros con signo.

4.1. El Sumador QFT.

Comencemos estudiando el sumador QFT propuesto en [Dra00]. El circuito realiza la suma de dos números a y b. El esquema básico del sumador se presenta en la figura 4.1.

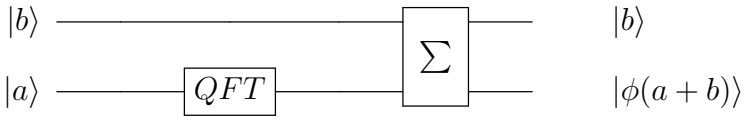


Figura 4.1: Sumador QFT.

Como vemos en la figura 4.1, la primera operación que realiza el sumador es la transformada cuántica de Fourier (o QFT) de $|a\rangle$, que es un estado que contiene la información relativa al número a. Esta operación realiza la transformación:

$$|a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i a k}{N}} |k\rangle, \tag{4.1}$$

donde $N = 2^n$ y n es el número de qubits que codifican el estado $|a\rangle$. El resultado es la QFT codifica el número en los desfases entre los términos $|k\rangle$. El segundo bloque, representado por $\sum$, es el bloque sumador. El bloque está compuesto por varias puertas de rotación de fase controladas por el estado $|b\rangle$, que codifica la información relativa a b. El resultado es que añade una fase a la QFT del estado $|a\rangle$ de la siguiente forma:

$$|a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i a k}{N}} e^{\frac{2\pi i b k}{N}} |k\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i (a+b)\cdot k}{N}} |k\rangle. \tag{4.2}$$

Es decir, realizamos la suma $a + b$ y la codificamos como desfases entre los términos $|k\rangle$. Tras este proceso bastaría realizar la QFT^{-1} de este resultado y obtendríamos la suma $a + b$. El esquema emplea n qubits para representar a y otros n para representar b a la entrada del circuito. A la salida del circuito, n qubits reprentarán el número b y otros n qubits representarán la suma $a + b$. Es decir, para sumar dos números de n bits utiliza $2n$ qubits y no emplea ningún qubit para almacenar la llevada.

Si $a + b \leq N$, al realizar la transformada cuántica de Fourier inversa o QFT^{-1} recuperamos la suma $a + b$, pero si $a + b > N$, la diferencia $a + b - N$ genera en el estado la misma fase que $a + b$, por tanto, no se puede distinguir entre $a + b$ y $a + b - N$ y se pierde información. Así que a la hora de recuperar la información de la suma $a + b$, en realidad que obtenemos la suma módulo N. Veremos cómo extender el esquema para obtener la suma.

En las siguientes subsecciones se mostrará cómo construir la QFT y el bloque sumador.

4.1.1. Bloque QFT

En esta sección vamos a explicar cómo construir el bloque QFT. Partimos del algoritmo presentado en el capítulo 2. Ya sabemos que el algoritmo, aplicado sobre el estado $|a\rangle$, realiza la transformación:

$$|a\rangle \rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{2\pi i a k/2^n} |k\rangle, \tag{4.3}$$

y que operando sobre esta expresión como se muestra en las ecuaciones (2.46)-(2.49) se puede transformar en:

$$\frac{1}{2^{\frac{n}{2}}} \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2\pi i \frac{a}{2^l}} |1\rangle \right]. \tag{4.4}$$

Como vemos, la fase de $|1\rangle$ contiene al número a dividido por el término 2^l. Recordemos que la conversión al sistema decimal de números binarios fraccionarios viene dada por la expresión:

$$(0.a_1 a_2 \ldots a_n)_2 = (a_1 \cdot 2^{-1} + a_2 \cdot 2^{-2} + \ldots + a_n \cdot 2^{-n})_{10}, \tag{4.5}$$

donde el subíndice 2 indica la expansión del número en el sistema binario, mientras que el subíndice 10 indica la expansión del número en el sistema decimal. Por tanto, la expresión (4.4) se puede reescribir como:

$$|a\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.a_n}|1\rangle)\,(|0\rangle + e^{2\pi i 0.a_n a_{n-1}}|1\rangle)\ldots(|0\rangle + e^{2\pi i 0.a_1 a_2 \ldots a_n}|1\rangle)}{2^{\frac{n}{2}}}. \tag{4.6}$$

Si llamamos

$$|\phi_k(a)\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\frac{2\pi i a}{2^k}}|1\rangle\right), \tag{4.7}$$

la expresión (4.6) se puede factorizar según [Dra00]:

$$\sum_{k=0}^{2^n-1} e^{2\pi i a k/2^n}|k\rangle = |\phi_1(a)\rangle \otimes |\phi_2(a)\rangle \otimes \ldots \otimes |\phi_{n-1}(a)\rangle \otimes |\phi_n(a)\rangle. \tag{4.8}$$

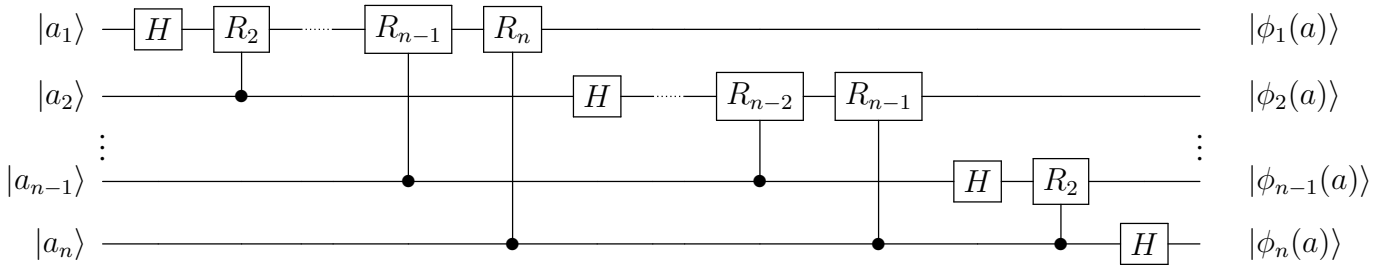


Figura 4.2: Transformada Cuántica de Fourier.

A partir de la expresión (4.6) podemos construir un circuito cuántico que realice la QFT. Este circuito se muestra en la figura 4.2. Para construir el esquema trabajamos con la expansión binaria de a, es decir, con $a_1, a_2, \ldots, a_n$, de tal manera que:

$$a = a_1 \cdot 2^0 + a_2 \cdot 2^1 + \ldots + a_{n-1} \cdot 2^{2n-2} + a_n \cdot 2^{2n-1}. \tag{4.9}$$

Empleando la expansión binaria de a, el estado $|a\rangle$ que codifica la información relativa a a se puede describir según:

$$|a\rangle = |a_1\rangle \otimes |a_2\rangle \otimes \ldots \otimes |a_{n-1}\rangle \otimes |a_n\rangle, \tag{4.10}$$

es decir, se representa como el producto tensorial de los qubits $|a_1\rangle, \ldots, |a_n\rangle$.

Una vez que tenemos codificada la información en qubits, procedemos a construir el circuito. El primer paso es aplicar una puerta de Hadamard al qubit $|a_1\rangle$. Esta puerta realizará la transformación:

$$|a_1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\frac{2\pi i a_1}{2^1}}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1}|1\rangle\right), \tag{4.11}$$

de forma que si $a_1 = 0$, el qubit se transforma en $1/\sqrt{2}(|0\rangle + |1\rangle)$ y si $a_1 = 0$, se transforma en $1/\sqrt{2}(|0\rangle - |1\rangle)$. A continuación se aplica una puerta de rotación R_2 controlada por $|a_2\rangle$. La puerta de rotación de fase general controlada por $|a_k\rangle$ se define según

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix} \tag{4.12}$$

y el resultado es que añade una fase relativa del tipo $e^{\frac{2\pi i a_k}{2^k}}$. Es decir, transforma el qubit de la siguiente manera:

$$\begin{aligned} |a_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\frac{2\pi i a_1}{2^1}} e^{\frac{2\pi i a_2}{2^2}} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1} e^{2\pi i 0.0a_2} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.a_1 + 0.0a_2)} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1 a_2} |1\rangle\right). \end{aligned} \tag{4.13}$$

A continuación se aplica la puerta R_3 controlada por $|a_3\rangle$, que transforma al qubit según:

$$\begin{aligned} |a_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\frac{2\pi i a_1}{2^1}} e^{\frac{2\pi i a_2}{2^2}} e^{\frac{2\pi i a_3}{2^3}} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1} e^{2\pi i 0.0a_2} e^{2\pi i 0.00a_3} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i (0.a_1 + 0.0a_2 + 0.00a_3)} |1\rangle\right) \\ &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1 a_2 a_3} |1\rangle\right). \end{aligned} \tag{4.14}$$

Procedemos de igual manera hasta llegar a la puerta R_n controlada por $|a_n\rangle$. El resultado final será:

$$|a_1\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1 a_2 a_3 \ldots a_n} |1\rangle\right). \tag{4.15}$$

De acuerdo con la expresión (4.7, podemos reescribir la expresión (4.17) según:

$$|a_1\rangle \rightarrow |\phi_1(a)\rangle. \tag{4.16}$$

Ahora procedemos de igual manera con $|a_2\rangle$. Primero aplicamos una puerta Hadamard, que transforma al qubit según $1/\sqrt{2}\left(|0\rangle + e^{\frac{2\pi i a_2}{2^1}} |1\rangle\right)$. A continuación aplicamos la puerta R_2 controlado por $|a_3\rangle$, después la puerta R_3 controlada por $|a_4\rangle$, y así hasta llegar a la puerta R_{n-1} controlada por $|a_n\rangle$. El resultado será:

$$|a_2\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_2 a_2 a_3 \ldots a_n} |1\rangle\right) = |\phi_2(a)\rangle. \tag{4.17}$$

Realizamos el mismo proceso con el resto de los qubits, hasta llegar a $|a_n\rangle$ sobre el que solo actúa una puerta de Hadamard, por ser el último qubit. El resultado será:

$$|a_n\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_n}|1\rangle\right) = |\phi_n(a)\rangle. \tag{4.18}$$

El último paso, que no se muestra en la figura 4.2 es aplicar puertas swap, que intercambian el estado entre dos qubits. Estas puertas se aplican entre $(|\phi_1(a)\rangle, |\phi_n(a)\rangle)$, $(|\phi_2(a)\rangle, |\phi_{n-1}(a)\rangle)$, …, $\left(\left|\phi_{\frac{n}{2}}(a)\right\rangle, \left|\phi_{\frac{n}{2+1}}(a)\right\rangle\right)$. El resultado es:

$$\begin{aligned}
|a_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_n}|1\rangle\right); \\
|a_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_{n-1}a_n}|1\rangle\right); \\
&\vdots \\
|a_{n-1}\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_2\ldots a_{n-1}a_n}|1\rangle\right); \\
|a_n\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_1a_2\ldots a_{n-1}a_n}|1\rangle\right);
\end{aligned} \tag{4.19}$$

El estado $|a\rangle$ quedará transformado según:

$$\begin{aligned}
|a\rangle &\rightarrow |\phi(a)\rangle \\
&= |\phi_1(a)\rangle \otimes |\phi_2(a)\rangle \otimes \ldots \otimes |\phi_n(a)\rangle \\
&= \frac{\left(|0\rangle + e^{2\pi i 0.a_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.a_{n-1}a_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.a_2\ldots a_{n-1}a_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.a_1a_2\ldots a_{n-1}a_n}|1\rangle\right)}{2^{\frac{n}{2}}}.
\end{aligned} \tag{4.20}$$

4.1.2. Bloque sumador

El segundo bloque que compone el sumador QFT es, precisamente, el bloque sumador. Este bloque va a continuación del bloque QFT y, en esencia, añade una fase del tipo $e^{\frac{2\pi i b}{2^n}}$ a la QFT del estado $|a\rangle$:

$$\begin{aligned}
|a\rangle &\rightarrow \frac{1}{\sqrt{2^n}}\sum_{k=0}^{N-1} e^{\frac{2\pi i a k}{2^n}} e^{\frac{2\pi i b k}{2^n}}|k\rangle \\
&= \frac{1}{\sqrt{2^n}}\sum_{k=0}^{N-1} e^{\frac{2\pi i (a+b)\cdot k}{2^n}}|k\rangle.
\end{aligned} \tag{4.21}$$

Para construir el bloque es necesario trabajar con la expansión binaria de $b, b_1, b_2, \ldots b_n$, de tal manera que:

$$b = b_1 \cdot 2^0 + b_2 \cdot 2^1 + \ldots + b_{n-1} \cdot 2^{2n-2} + b_n \cdot 2^{2n-1}. \tag{4.22}$$

El estado $|b\rangle$, que codifica la información relativa a b se puede describir según:

$$|b\rangle = |b_1\rangle \otimes |b_2\rangle \otimes \ldots \otimes |b_{n-1}\rangle \otimes |b_n\rangle. \tag{4.23}$$

El bloque sumador se construye empleando una serie de puertas de rotación de fase controladas por los qubits $|b_1\rangle, \ldots, |b_n\rangle$, como se muestra en la figura 4.3. La puerta de rotación de fase controlada se construye de acuerdo con la expresión 4.12. Estas matrices son conmutativas entre sí [Dra00].

Las puertas se deben elegir de tal manera que se cumpla la siguiente expresión:

$$\begin{aligned} |a\rangle \quad &\rightarrow \quad \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i(a+b)\cdot k}{2^n}} |k\rangle = \\ &= \quad \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i}{2^n} 2^{n-j}(a_j+b_j) \sum_{s=1}^{n} k_s 2^{n-s}} |k\rangle. \end{aligned} \tag{4.24}$$

Esto implica que para cada k_s, que representa a ϕ_1,. . ., ϕ_n, tenemos que elegir puertas de rotación de fase que introduzcan la fase:

$$e^{\frac{2\pi i}{2^n} 2^{n-j} 2^{n-s}(a_j+b_j)\cdot k_s} = e^{\frac{2\pi i}{2^{j+s-n}}(a_j+b_j)\cdot k_s}, \tag{4.25}$$

es decir, deben introducir la fase $e^{\frac{2\pi i}{2^{j+s-n}}}$, por lo que las puertas de rotación de fase a utilizar son $R_k = R_{j+s-n}$.

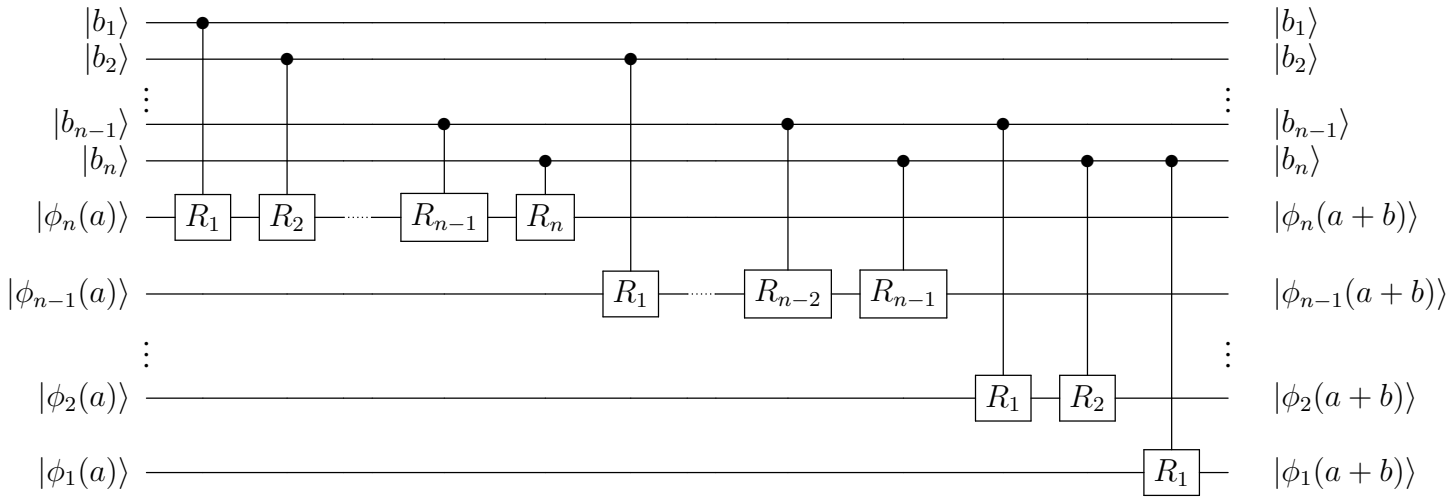


Figura 4.3: Suma de dos qubits.

Para ver cómo actúan estas puertas, comenzaremos analizando qué sucede al aplicar las rotaciones condicionales de fase sobre $|\phi_n(a)\rangle$. Las puertas que actúan sobre este qubit son las puertas $R_{1+n-n} = R_1$ controlada por $|b_1\rangle$, $R_{2+n-n} = R_2$ controlada por $|b_2\rangle$ y así sucesivamente hasta llegar a la puerta $R_{n+n-n} = R_n$, controlada por $|b_n\rangle$, como vemos en el esquema 4.3.

La puerta de rotación de fase R_1 controlada por $|b_1\rangle$ realiza la siguiente operación:

$$|\phi_n(a)\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\left(\frac{a_1}{2^1}+\frac{a_2}{2^2}+\ldots+\frac{a_{n-1}}{2^{n-1}}+\frac{a_n}{2^n}+\frac{b_1}{2^1}\right)}|1\rangle). \tag{4.26}$$

O lo que es lo mismo:

$$|\phi_n(a)\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_1a_2\ldots a_n+0.b_1)}|1\rangle). \tag{4.27}$$

La siguiente puerta empleada es una rotación de fase R_2 controlada por $|b_2\rangle$, que actúa de la siguiente manera:

$$|\phi_n(a)\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_1a_2\ldots a_n+0.b_1b_2)}|1\rangle). \tag{4.28}$$

Se continúa aplicando rotaciones de fase controladas hasta llegar a la puerta R_n controlada por $|b_n\rangle$, y obtendremos como resultado:

$$|\phi_n(a)\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_1a_2\ldots a_n+0.b_1b_2\ldots b_n)}|1\rangle) = |\phi_n(a+b)\rangle. \tag{4.29}$$

Con $|a_{n-1}\rangle$ se realiza el mismo proceso. Las puertas que actúan sobre este qubit son $R_{1+(n-1)-n} = R_0$ controlada por $|b_1\rangle$, $R_{2+(n-1)-n} = R_1$ controlada por $|b_2\rangle$ y así hasta la puerta $R_{n+(n-1)-n} = R_{n-1}$ controlada por $|b_n\rangle$. Como R_0 introduce la fase $e^{2\pi i} = 1$, esta deja al qubit sin alterar y en la práctica no se incluye en el circuito, de ahí que no aparezca en la figura 4.3. Sucederá lo mismo cada vez que la puerta introduzca una fase del tipo $e^{\frac{2\pi i}{2^k}}$ donde $k \leq 0$.

Tras aplicar la transformada cuántica de Fourier, $|a_2\rangle$ queda transformado según:

$$|a_2\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.a_2a_3\ldots a_n}|1\rangle\right) = |\phi_{n-1}(a)\rangle. \tag{4.30}$$

Posteriormente se aplican las puertas de rotación de fase controladas por $b_2, b_3, \ldots, b_n$. El resultado será:

$$\begin{aligned}
|\phi_{n-1}(a)\rangle &\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\left(\frac{a_2}{2^1}+\frac{a_3}{2^2}+\ldots+\frac{a_{n-1}}{2^{n-2}}+\frac{a_n}{2^{n-1}}+\frac{b_2}{2^1}\right)}|1\rangle) \\
&\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\left(\frac{a_2}{2^1}+\frac{a_3}{2^2}+\ldots+\frac{a_{n-1}}{2^{n-2}}+\frac{a_n}{2^{n-1}}+\frac{b_2}{2^1}+\frac{b_3}{2^2}\right)}|1\rangle) \\
&\rightarrow \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\left(\frac{a_2}{2^1}+\frac{a_3}{2^2}+\ldots+\frac{a_{n-1}}{2^{n-2}}+\frac{a_n}{2^{n-1}}+\frac{b_2}{2^1}+\frac{b_3}{2^2}+\ldots+\frac{b_{n-1}}{2^{n-2}}+\frac{b_n}{2^{n-1}}\right)}|1\rangle) \\
&= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.a_2a_3\ldots a_{n-1}a_n+b_2b_3\ldots b_{n-1}b_n)}|1\rangle) = |\phi_2(a+b)\rangle.
\end{aligned} \tag{4.31}$$

El sumador lleva a cabo el mismo procedimiento con $|\phi_{n-2}(a)\rangle\ldots|\phi_1(a)\rangle$, de forma que a la salida se obtienen los estados $|\phi_{n-2}(a+b)\rangle\ldots|\phi_1(a+b)\rangle$. Tras ello bastaría realizar una transformada cuántica de Fourier inversa y obtendríamos como resultado la suma $|a+b\rangle$.

El número de puertas cuánticas que necesitaremos para implementar el esquema es el siguiente:

	T. Hadamard	Rotaciones de fase
QFT	n	$\frac{n(n-1)}{2}$
QFT^{-1}	n	$\frac{n(n-1)}{2}$
$Suma$	0	$\frac{n(n+1)}{2}$
Total	$2n$	$\frac{3n^2-n}{2}$

Tabla 4.1: Número de puertas necesarias para implementar el sumador cuántico con QFT.

Una vez analizado el funcionamiento del sumador con QFT, pasaremos a estudiar cómo se puede ampliar este esquema para implementar sumas de más de dos números.

4.2. Sumador Cuántico de más de dos Números.

A partir del esquema propuesto en [Dra00] se puede realizar de manera sencilla una extensión para calcular sumas de más de dos números. Por ejemplo, si deseamos sumar k qubits $|a\rangle + |b_1\rangle + \ldots + |b_{k-1}\rangle$, bastaría con realizar la transformada cuántica de Fourier de uno de los estados, por ejemplo $|a\rangle$ como en el esquema de [Dra00] y aplicar sobre el estado transformado rotadores de fase condicionales que vengan controlados por el resto de estados que se desean sumar, en un proceso similar al descrito en el apartado anterior, de tal manera que:

$$|a\rangle \rightarrow \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i a k}{2^n}} e^{\frac{2\pi i k \sum b_i}{2^n}} |k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i (a+\sum b_i)\cdot k}{2^n}} |k\rangle. \tag{4.32}$$

El esquema propuesto se muestra en la figura 4.4.

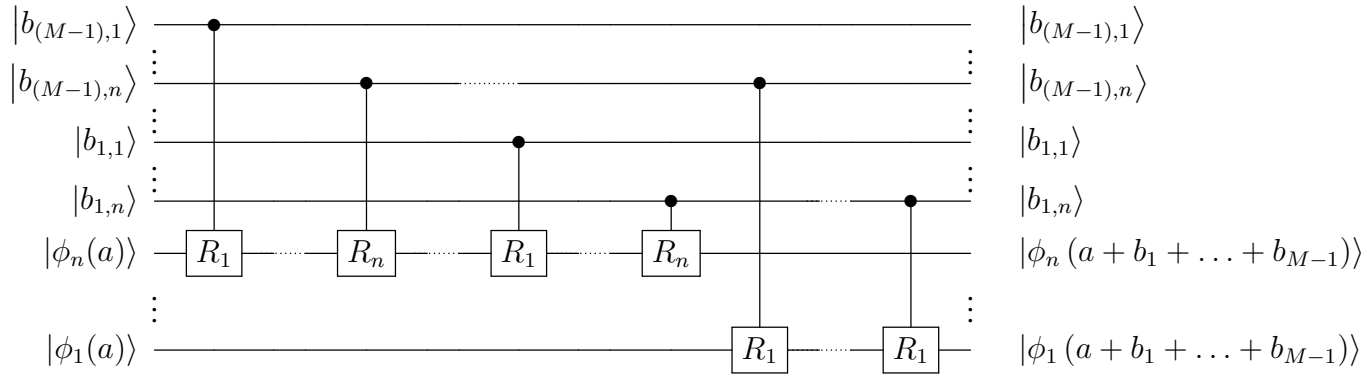


Figura 4.4: Sumador de k números.

Cada nuevo estado $|x\rangle$ representado mediante la expansión binaria $|x_1 x_2 \ldots x_{n-1} x_n\rangle$ necesitaría $n(n+1)/2$ nuevas puertas de rotación controlada de fase, por lo que si a $|a\rangle$ le

sumamos $k-1$ números representados por n dígitos binarios, el número total de puertas de rotación de fase necesarios para implementar el bloque sumador vendrá dado por la expresión:

$$(k-1)\frac{n(n+1)}{2}, \tag{4.33}$$

mientras que seguiremos necesitando n transformadas de Hadamard y $n(n-1)/2$ rotadores de fase para implementar la QFT y el mismo número de puertas para construir la QFT^{-1}.

Una vez visto como ampliar el esquema para sumar más de dos qubits, discutiremos cómo modificar el sumador para que realice operaciones de suma con llevada.

4.3. Suma Aritmética.

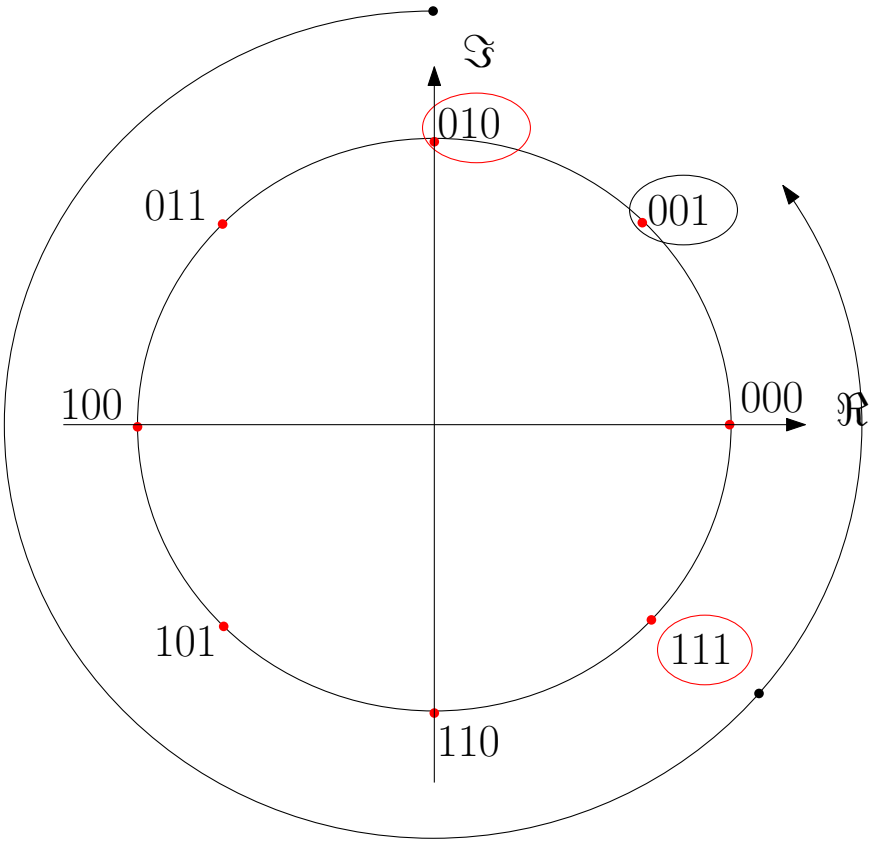


Figura 4.5: Suma empleando el esquema [Dra00].

El sumador QFT que hemos presentado lleva a cabo, en realidad, la suma módulo N, donde $N = 2^n$ y n el número de dígitos que componen la expansión binaria de a y b. Es

decir, estamos realizando la operación:

$$\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i(a+b)\cdot k}{2^n}} |k\rangle. \tag{4.34}$$

El resultado que se recupera en caso de que $a + b \leq N$ es, precisamente, la suma de ambos términos. Pero si $a + b > N$, el resultado que se recupera es $a + b \mod N$.

Veamos un ejemplo numérico en la siguiente sección.

4.3.1. Suma módulo N

Supongamos que queremos sumar los números $a = 2$ y $b = 7$. Para ello utilizamos la expansión binaria de $n = 3$ bits, con $N = 2^3 = 8$, de manera que

$$\begin{aligned} N &= 1000; \\ a &= 010; \\ b &= 111. \end{aligned} \tag{4.35}$$

El sumador debería realizar la operación:

$$\begin{array}{r} 010 \\ +\,111 \\ \hline 1001 \end{array} \tag{4.36}$$

En cambio, lo que obtenemos es $1001 - 1000 = 001 = 1$, en lugar de 9. Es decir, obtenemos la suma módulo N.

La figura 4.5 representa en el plano complejo los números todos los números binarios de tres bits, incluidos $a = 010$ y $b = 111$. La secuencia que representamos comienza en 000 y se incrementa en una unidad hasta llegar a 111. A partir de este número, la secuencia se repite. Es decir, donde debería estar representado 1000 tenemos 000, donde debería encontrarse 1001 se encuentra en realidad 001 y así sucesivamente. Esta secuencia está representada sobre una circunferencia en el plano complejo, y para incrementar cada número en una unidad, basta con dar un salto en sentido antihorario. Si queremos incrementar 111 en dos unidades, deberemos dar dos saltos en la circunferencia en sentido antihorario. Deberíamos llegar a 1001, pero como la circunferencia sólo representa hasta 111 y después se repite la secuencia, el resultado obtenido es 001.

4.4. Extensión del esquema

Para implementar la suma con llevada, se puede extender el esquema añadiendo un qubit adicional, que en este caso será $|0\rangle$. La solución es análoga a las empleadas en los procesadores clásicos en los que, para almacenar la llevada y prevenir problemas de desbordamiento, se añaden bits adicionales.

El esquema realizará la transformada cuántica de Fourier de $|0a_1a_2\dots a_n\rangle$ como se muestra en la figura 4.6. Como vemos, es equivalente a representar a con $n+1$ bits y en realidad estaremos haciendo la QFT de $n+1$ qubits que dará como resultado:

$$\begin{aligned} |a\rangle &\rightarrow \frac{1}{\sqrt{2^{n+1}}}\sum_{k=0}^{2^{n+1}-1} e^{\frac{2\pi i a\cdot k}{2^{n+1}}}|k\rangle = \\ &= \frac{1}{\sqrt{2N}}\sum_{k=0}^{2N-1} e^{\frac{2\pi i a\cdot k}{2N}}|k\rangle, \end{aligned} \tag{4.37}$$

donde $N=2^n$. Por tanto, las puertas de rotación de fase realizarán la operación:

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{k=0}^{2^{n+1}-1} e^{\frac{2\pi i}{2^{n-1}}2^{n-j}(a_j+b_j)\sum_{s=1}^{n}k_s2^{n-s}}|k\rangle, \tag{4.38}$$

y estaremos haciendo la suma módulo $2^{n+1}=2N$. La fase que deben introducir las puertas de rotación es:

$$e^{\frac{2\pi i}{2^{n+1}}(a_j+b_j)\cdot k_s\cdot 2^{(n-j)+(n+1-s)}} = e^{2\pi i(a_j+b_j)\cdot k_s\cdot 2^{2n-j-s+1-n-1}} = e^{\frac{2\pi i}{2^{j+s-n}}}. \tag{4.39}$$

La puerta de rotación de fase que aplicaremos será:

$$R_{k'} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{k'}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{k+1}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{2^k}}} \end{bmatrix}, \tag{4.40}$$

y las puertas vendrán dadas por R_{j+s-n}. El esquema del bloque sumador resultante se muestra en la figura 4.7.

Veamos qué ocurre matemáticamente. La transformada de Hadamard convierte $|0\rangle$ en $\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$, que también se puede escribir como $\frac{1}{\sqrt{2}}(|0\rangle+e^{\frac{2\pi i 0}{2}}|1\rangle)$. Posteriormente se aplica una puerta de rotación de fase R_2 controlada por $|a_1\rangle$, que tendrá el siguiente resultado:

$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle+e^{\frac{2\pi i 0}{2}}|1\rangle\right) \\ &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle+e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2^2}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle+e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2\cdot 2^1}\right)}|1\rangle\right). \end{aligned} \tag{4.41}$$

Tras aplicar todas las puertas de rotación de fase, el resultado será:

$$
\begin{aligned}
|0\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{\frac{2\pi i 0}{2}}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2^2}+\ldots+\frac{a_{n-1}}{2^n}+\frac{a_n}{2^{n+1}}\right)}|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2\cdot 2^1}+\ldots+\frac{a_{n-1}}{2\cdot 2^{n-1}}+\frac{a_n}{2\cdot 2^n}\right)}|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0a_1\ldots a_{n-1}a_n}|1\rangle\right) = |\phi_1(a)\rangle. \qquad (4.42)
\end{aligned}
$$

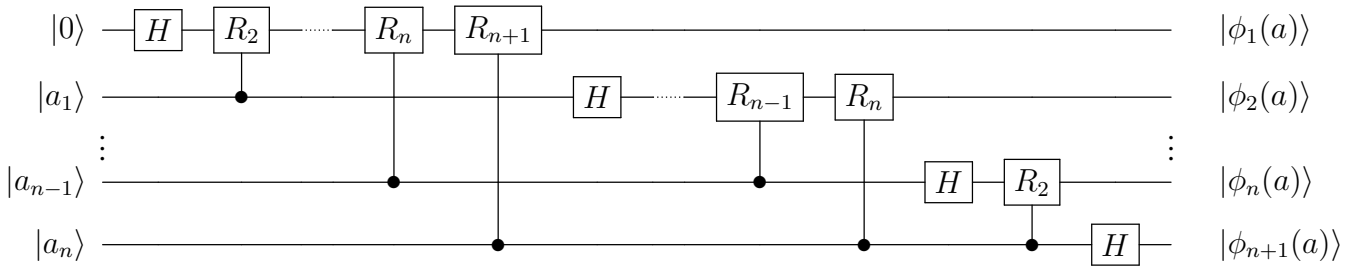


Figura 4.6: Transformada Cuántica de Fourier de $|0a_1 \ldots a_n\rangle$.

De nuevo, aunque no se muestra en la figura 4.6, se aplican al final del circuito puertas swap, de forma que el resultado será:

$$
\begin{aligned}
|\phi_1(a)\rangle &= \frac{1}{2}\left(|0\rangle + e^{2\pi i 0.an}|1\rangle\right); \\
&\vdots \\
|\phi_{n+1}(a)\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0a_1\ldots a_n}|1\rangle\right). \qquad (4.43)
\end{aligned}
$$

A continuación implementamos el bloque sumador. Para ello aplicamos las puertas de rotación de fase controladas por $|b_1\rangle \ldots |b_n\rangle$. El resultado será:

$$
\begin{aligned}
|0\rangle &\rightarrow |\phi_{n+1}(a)\rangle \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2^{2^1}}+\ldots+\frac{a_n}{2^{2^n}}+\frac{b_1}{2^{2^1}}\right)}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2}+\frac{a_1}{2^{2^1}}+\ldots+\frac{a_n}{2^{2^n}}++\frac{b_1}{2^{2^1}}+\ldots+\frac{b_n}{2^{2^n}}\right)}|1\rangle\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i(0.0a_1\ldots a_n+0.0b_1\ldots b_n)}|1\rangle\right). \qquad (4.44)
\end{aligned}
$$

De esta manera estaremos implementando la suma módulo $2N$ que, para sumar dos números a y b, utiliza un qubit adicional para almacenar la llevada.

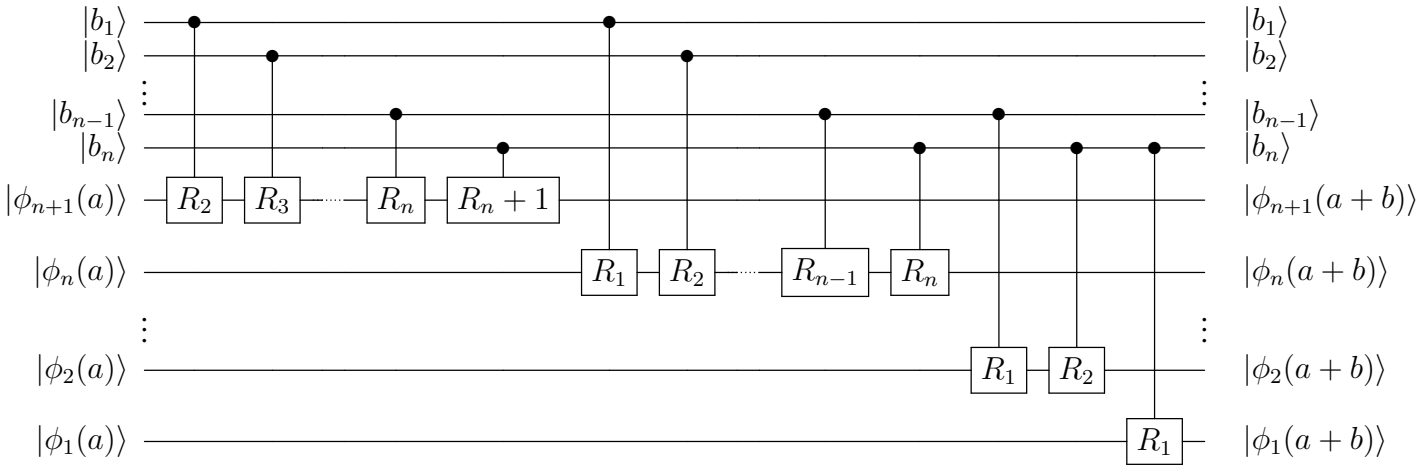


Figura 4.7: Suma con llevada.

4.4.1. Ejemplo de suma con llevada

En el ejemplo presentado en la subsección 4.3.1 sumamos $a = 2$ y $b = 7$ empleando la expansión binaria de $n = 3$ dígitos. El resultado es el número 1001, pero seguimos trabajando con la expansión binaria de tres dígitos, por lo que el resultado recuperado será 001.

Para resolver el problema, podemos hacer la suma módulo $2N$. Puesto que $N = 2^n$, $2N = 2 \cdot 2^n = 2^{n+1}$, por lo que trabajamos con números cuya expansión binaria será de $n + 1 = 4$ dígitos. Con ello podremos recuperar el número 1001. Podemos ver una representación gráfica en la figura 4.8

En este caso representamos la secuencia de 0000 a 1111. El número 010 se puede representar como 0010, mientras que 111 se puede representar como 0111. Como queremos incrementar 0111 en dos unidades, sólo tenemos que saltar dos posiciones en la circunferencia desde este número en sentido antihorario. El resultado es 1001, que en decimal se corresponde con 9, el resultado que deseamos recuperar.

4.4.2. Número de puertas

El número de qubits que tendremos que añadir para almacenar la llevada dependerá tanto de la cantidad de números que se van a sumar como del tamaño n de los mismos.

En caso de sumar dos números de n bits, basta con añadir un bit más para dejar espacio para almacenar la llevada. El resultado final tendrá un tamaño de $n + 1$ dígitos binarios. Por tanto el esquema añadirá un qubit más para almacenar la llevada, haciendo la QFT de $n + 1$ qubits $|0\rangle, |a_1\rangle, \ldots, |a_n\rangle$.

Cuando se trabaja con dos números a y b de expansiones binarias de n y m dígitos respectivamente, con $n \neq m$, el riesgo es que se produzca un desbordamiento con respecto al número de mayor tamaño. Para prevenir dicho riesgo, basta con añadir un bit más con respecto al número de mayor tamaño. El resultado final estará compuesto por $max(m, n)+1$ dígitos binarios. En nuestro esquema supone hacer la QFT $max(m, n) + 1$ qubits, de los

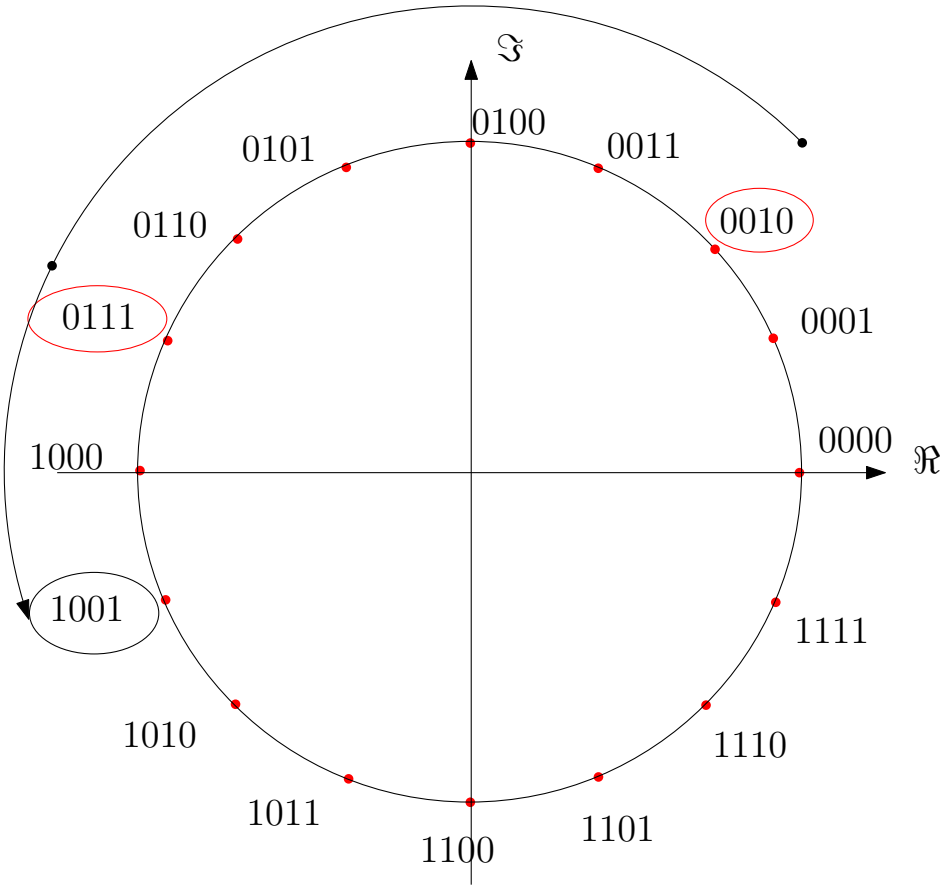


Figura 4.8: Representación gráfica de la suma con llevada.

cuales uno será $|0\rangle$, mientras que los restantes codificarán al número de representación binaria de mayor tamaño.

Por último hay que definir el número de qubits necesarios para almacenar la suma de k enteros representados por n bits. El número de bits que compondrán el resultado final será:

$$log_2(k \cdot 2^n) = log_2(k) + log_2(2^n) = log_2(k) + n. \tag{4.45}$$

Puesto que inicialmente los números venían representados con n dígitos binarios, el número de qubits necesarios para almacenar la llevada será:

$$log_2(k) + n - n = log_2(k). \tag{4.46}$$

Como la expresión 4.46 no siempre dará como resultado un número entero, eligiendo $\lceil log_2(k) \rceil$ qubits se garantizará habilitar espacio suficiente para almacenar la llevada.

Modificar el sumador con QFT para que realice la suma con llevada de k números supone un incremento de las puertas necesarias para implementar el circuito, debido a que se opera con más qubits en el dominio transformado y a que se suma una cantidad mayor

de números. A continuación veremos el número de puertas necesarias para implementar el sumador teniendo en cuenta que:

- Se implementa la suma $a + b_1 + \ldots + b_{k-1}$, es decir, se suman k números.
- Tanto a como los distintos b_i vienen representados por n dígitos binarios como hasta ahora.
- Se añaden $\lceil log_2(k) \rceil$ qubits para almacenar la llevada, que hay que llevar al dominio transformado.

Puesto que el número de qubits a transformar es $\lceil log_2(k) \rceil + n$, el número de puertas necesarias para implementar la QFT y la QFT^{-1} será:

$$2\left(\lceil log_2(k) \rceil + n\right) + \left(\lceil log_2(k) \rceil + n\right)\left(\lceil log_2(k) \rceil + n - 1\right), \tag{4.47}$$

como se muestra en la tabla 4.2.

	T. Hadamard	Rotaciones de fase
QFT	$\lceil log_2(k) \rceil + n$	$\frac{(\lceil log_2(k) \rceil + n)(\lceil log_2(k) \rceil + n - 1)}{2}$
QFT^{-1}	$\lceil log_2(k) \rceil + n$	$\frac{(\lceil log_2(k) \rceil + n)(\lceil log_2(k) \rceil + n - 1)}{2}$
Total	$2\left(\lceil log_2(k) \rceil + n\right)$	$\left(\lceil log_2(k) \rceil + n\right)\left(\lceil log_2(k) \rceil + n - 1\right)$

Tabla 4.2: Número de puertas para implementar la suma con llevada de k qubits.

El número de rotaciones de fase necesarias para sumar los $k - 1$ números restantes será el siguiente:

$$\frac{(k-1)(n+1)n}{2} + n(k-1)\lceil log_2(k) \rceil . \tag{4.48}$$

Donde $\frac{(k-1)(n+1)n}{2}$ es el número de puertas de rotación de fase que suman los distintos $|b_i\rangle$ a $|a\rangle$. Además se añade el término $n(k-1)\lceil log_2(k) \rceil$ que suman $|b_m\rangle$ a los qubits que almacenan la llevada. Como resultado, el número final de puertas necesarias para implementar el sumador QFT para k qubits, es decir, para sumar $a + \sum_{i=1}^{k-1} b_i$ con $\lceil log_2(k) \rceil$ qubits de llevada es:

$$n^2\frac{k+1}{2} + n\left[\frac{k+1}{2} + (k+1)\lceil log_2(k) \rceil\right] + \lceil log_2(k) \rceil^2 + \lceil log_2(k) \rceil . \tag{4.49}$$

Podemos comparar el número de puertas necesario para implementar el sumador con QFT y para implementar el sumador que emula el esquema clásico de la suma con llevada presentado en [VBE96].

El esquema realiza la suma con llevada de dos números a y b. De nuevo, se recurre a la representación binaria de dichos números, es decir, $a_1 \dots a_n$ y $b_1 \dots b_n$ y además se incluye el bit de llevada para cada suma elemental, representado por $c_1 \dots c_n$. En cada bit se calcula primero el bit de llevada que se empleará para realizar la suma de los bits siguientes. Posteriormente, se deshace esta operación y se realiza la suma. En total, el esquema utiliza cuatro puertas de Toffoli y cuatro puertas XOR controladas, es decir, ocho puertas, para implementar la suma de cada par de bits menos significativos. Para sumar el par de bits más significativos, emplea dos puertas de Toffoli y cuatro puertas XOR controladas, es decir, un total de seis puertas. El número total de puertas en función de n y de k será:

$$(k-1)[(n-1)8+6], \tag{4.50}$$

Veamos en qué casos el sumador QFT se implementa con menos puertas que el sumador con llevada que emula el esquema clásico. Se tiene que cumplir que (4.49)<(4.50). Si resolvemos n en función de k, esta condición se traduce en que:

$$n^2\frac{k+1}{2}+n\left[\frac{k+1}{2}+(k+1)\left\lceil log_2(k)\right\rceil\right]+\left\lceil log_2(k)\right\rceil^2+\left\lceil log_2(k)\right\rceil-(k-1)[(n-1)8+6]<0. \tag{4.51}$$

Resolviendo, obtenemos que las raíces de esta expresión vienen dadas por:

$$\begin{aligned} n &= \frac{15k-2(k-1)\lceil log_2(k)\rceil-17}{2(k+1)} \pm \\ &\pm \frac{\sqrt{4(k^2-1)\lceil log_2(k)\rceil^2-60(k-1)^2\lceil log_2(k)\rceil+209k^2-510k+305}}{2(k+1)}. \end{aligned} \tag{4.52}$$

En general, las soluciones de estas expresiones son o bien números imaginarios o bien resultan en $n < 1$, lo cual no es un resultado posible. Por tanto, el sumador QFT requerirá un número mayor o igual de puertas que el sumador clásico.

El sumador QFT no es más eficiente en términos de número de puertas que el sumador que emula el esquema clásico. Sin embargo, permite implementar operaciones que con el sumador clásico no se pueden realizar de forma directa, como la media o la suma ponderada. En el siguiente apartado discutiremos como implementar estas operaciones con el sumador QFT.

4.5. Media y Suma Ponderada

Implementar la suma mediante la QFT y rotaciones condicionales de fase nos ofrece como ventaja poder variar el factor por el que va a rotar la fase, ofreciéndonos la posibilidad de implementar otras operaciones aparte de la suma.

Supongamos que queremos operar con M números, $a, b_1, b_2, \dots, b_{M-1}$, representados por n dígitos binarios cada uno. Realizamos la transformada cuántica de Fourier de n

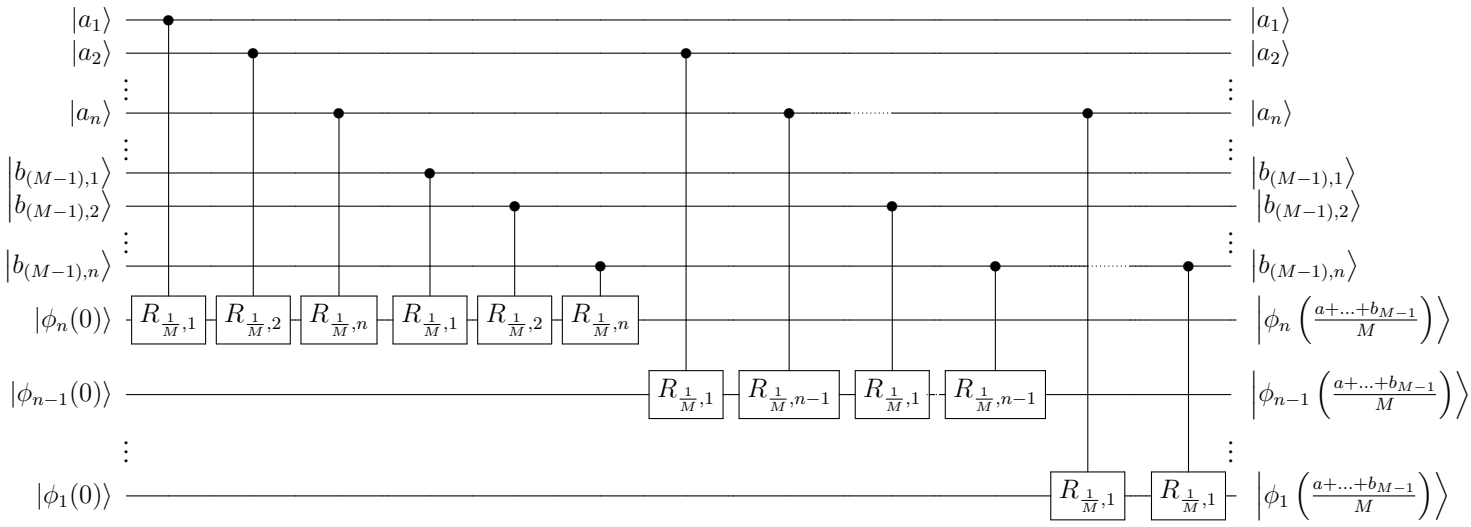


Figura 4.9: Media de M números.

qubits $|0\rangle$, que se transformarán en los estados $|\phi_1(0)\rangle \ldots |\phi_n(0)\rangle$, donde:

$$\begin{aligned}
|\phi_1(0)\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0_n}|1\rangle\right); \\
|\phi_2(0)\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0_n 0_{n+1}}|1\rangle\right); \\
&\vdots \\
|\phi_{n-1}(0)\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0_2 0_3 \ldots 0_n 0_{n+1}}|1\rangle\right); \\
|\phi_n(0)\rangle &= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0_1 0_2 \ldots 0_n 0_{n+1}}|1\rangle\right).
\end{aligned} \tag{4.53}$$

A continuación diseñamos la puerta de rotación de fase general controlada:

$$R_{\frac{1}{M},k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{M 2^k}} \end{bmatrix} \tag{4.54}$$

y la aplicamos a $|\phi_1(0)\rangle \ldots |\phi_{n+1}(0)\rangle$ siguiendo el esquema mostrado en la figura 4.9, es decir, aplicamos dichas puertas controladas por $|a_1\rangle \ldots |a_n\rangle$ y $|b_1\rangle \ldots |b_n\rangle$. El resultado obtenido será:

$$
\begin{aligned}
|\phi_1(0)\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{a_n}{M2^1}+\frac{b_{1,n}}{M2^1}+\ldots+\frac{b_{M-1,n}}{M2^1}\right)}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{0.a_n+\ldots+0b_{1,n}+0.b_{(M-1),n}}{M}}|1\rangle\right) \\
&\rightarrow \left|\phi_1\left(\frac{a+b_1+\ldots+b_{M-1}}{M}\right)\right\rangle; \\
&\vdots \qquad (4.55)
\end{aligned}
$$

$$
\begin{aligned}
|\phi_{n-1}(0)\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{a_2}{M2^1}+\ldots+\frac{a_n}{M2^{n-1}}+\frac{b_{1,2}}{M2^1}+\ldots+\frac{b_{1,n}}{M2^{n-1}}+\ldots+\frac{b_{(M-1),2}}{M2^1}+\ldots+\frac{b_{(M-1),n}}{M2^{n-1}}\right)}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{0.a_2\ldots a_n+0.b_{1,2}\ldots b_{1,n}+\ldots+0.b_{(M-1),2}\cdots b_{(M-1),n}}{M}}|1\rangle\right) \\
&\rightarrow \left|\phi_2\left(\frac{a+b_1+\ldots+b_{M-1}}{M}\right)\right\rangle; \qquad (4.56)
\end{aligned}
$$

$$
\begin{aligned}
|\phi_n(0)\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{a_1}{M2^1}+\ldots+\frac{a_n}{M2^n}+\frac{b_{1,1}}{M2^1}+\ldots+\frac{b_{1,n}}{M2^n}+\ldots+\frac{b_{m-1,1}}{M2^1}+\ldots+\frac{b_{M-1,n}}{M2^n}\right)}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\frac{0.a_1\ldots a_n+0.b_{1,1}\ldots b_{1,n}+\ldots+0.b_{(M-1),1}\cdots b_{(M-1),n}}{M}}|1\rangle\right) \\
&\rightarrow \left|\phi_n\left(\frac{a+b_1+\ldots+b_{M-1}}{M}\right)\right\rangle. \qquad (4.57)
\end{aligned}
$$

Es decir, habremos calculado la media de los M qubits $|a\rangle, |b_1\rangle, \ldots, |b_{M-1}\rangle$. La media no requiere qubits auxiliares para dejar espacio para la llevada, puesto que el resultado se va a encontrar entre 0 y $2^n - 1$.

Podemos realizar un procedimiento similar para modificar las puertas de rotación de fase controladas para que, en lugar de calcular la media, calculen la suma ponderada $\sum_i \delta_i |x_i\rangle$. Basta con diseñar puertas de rotación de fase de este tipo:

$$
R_{\delta_i k} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\delta_i \frac{2\pi i}{2^k}} \end{bmatrix}, \qquad (4.58)
$$

donde δ_i representa el peso por el que queremos multiplicar el qubit. Suponiendo que queremos sumar $\delta_1|a\rangle + \delta_2|b_1\rangle + \ldots + \delta_M|b_{M-1}\rangle$ habremos de seguir un procedimiento similar al mostrado para calcular la media de M números. Primero realizaremos la transformada cuántica de Fourier de $|0\rangle$ y posteriormente le sumaremos los qubits $|a\rangle, |b_1\rangle, \ldots, |b_{M-1}\rangle$ empleando las puertas de rotación de fase $R_{\delta_1 k}, R_{\delta_2 k}, \ldots, R_{\delta_M k}$. El resultado será:

$$
\begin{aligned}
|\phi_n(0)\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\delta_1\left(\frac{a_1}{2^1}+\ldots+\frac{a_n}{2^n}\right)+\delta_2\left(\frac{b_{1,1}}{2^1}+\ldots+\frac{b_{1,n}}{2^n}\right)+\ldots+\delta_M\left(\frac{b_{(M-1),1}}{2^1}+\ldots+\frac{b_{(M-1),n}}{2^n}\right)\right)}|1\rangle\right) \\
&\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\delta_1 0.a_1a_2\ldots a_{n-1}a_n+\delta_2 0.b_{1,1}\ldots b_{1,n}+\ldots+\delta_M 0.b_{(M-1),1}\ldots b_{(M-1),n}\right)}|1\rangle\right) \\
&= |\phi_n\left(\delta_1 a+\delta_2 b_1+\ldots+\delta_M b_{M-1}\right)\rangle, \qquad (4.59)
\end{aligned}
$$

si $\sum_i \delta_i = 1$. En caso contrario, deberemos añadir qubits auxiliares para almacenar la llevada. Si el resultado final tiene un tamaño $log_2(\sum \delta_i(2^n))$, tendremos que añadir $log_2(\sum \delta_i)$ qubits auxiliares y modificar las puertas de rotación de forma conveniente para asegurar que no se dé desbordamiento y se recupere el resultado deseado.

4.6. Representación de Números con Signo.

Hasta el momento hemos trabajado con qubits que representaban números enteros en el sistema binario, pero no hemos tenido en cuenta cuestiones como el signo de dichos números. Pero, ¿qué sucedería si quisiéramos operar con un número entero negativo?

Podemos inspirarnos en algunas de las representaciones empleadas en los ordenadores clásicos, como la representación signo - magnitud. En esta representación, de los n bits que representan un número, se utiliza el situado más a la izquierda para representar el signo, de tal manera que si el bit es 0, el número es positivo, mientras que si es 1 el número es negativo [Knu14].

De la misma manera, se puede añadir un dígito adicional a los n empleados hasta ahora para representar los qubits con los que se quiere operar. Este qubit sería $|0\rangle$ para representar el signo positivo y $|1\rangle$ para representar el signo negativo. La ventaja de este sistema de representación es la codificación natural para nuestro sumador módulo N.

Veamos un ejemplo gráfico. La figura 4.10 representa números binarios de tres dígitos en el plano complejo. Tomemos como ejemplo los números 010 y 110, que se corresponden con los números 2 y 6. Podemos ver que entre ambos números existe un desfase de π, es decir, que tenemos los números $|2|e^{i0}$ y $|2|e^{i\pi}$, que también equivalen a 2 y -2. Por tanto, los números que comienzan con 0 representarían enteros positivos y si giramos π en el plano complejo encontramos el número negativo correspondiente, que comenzaría con 1.

En el caso de sumar dos qubits, incluir un dígito para representar el signo requeriría una nueva transformada de Hadamard así como n rotaciones condicionales de fase tanto para la transformada cuántica de Fourier como para la Transformada Inversa, de tal manera que necesitaremos $2(n+1)$ Transformadas de Hadamard y $n(n+1)$ rotadores de fase. Mientras que el número de puertas de rotación de fase necesarias para implementar la suma será $\frac{(n+1)(n+2)}{2}$.

Este tipo de representación, aún siendo sencilla de implementar, tiene un efecto que es necesario corregir. Supongamos que queremos sumar dos números negativos, -1 y -2.

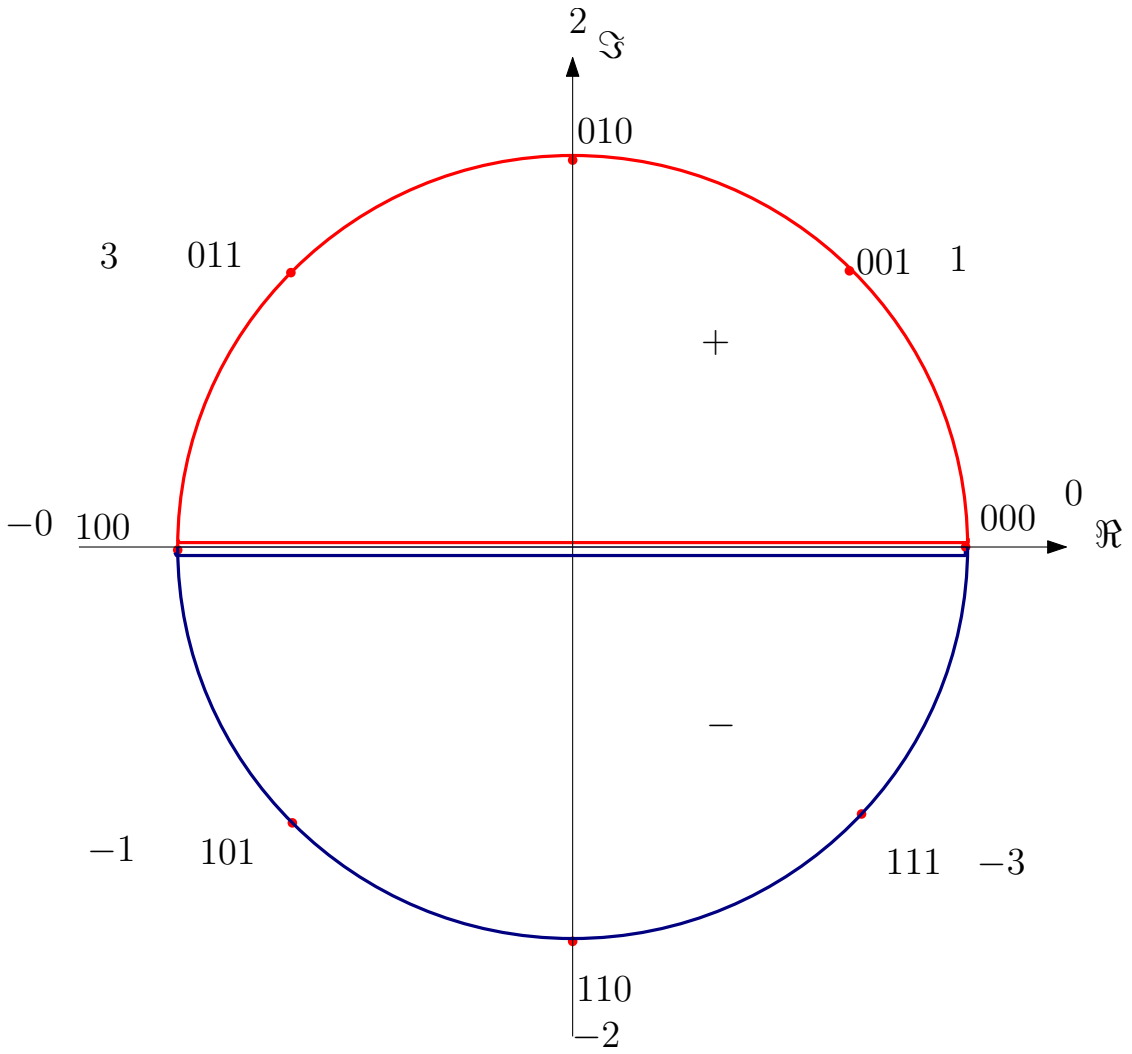


Figura 4.10: Representación en la esfera de números positivos y negativos.

La representación, empleando dos bits y un tercero de signo, vendría dada por 101 y 110. Si sumamos empleando el esquema de [Dra00], el resultado será:

$$101 + 110 = 011, \tag{4.60}$$

es decir, obtendremos como resultado un número positivo, 3, en lugar del número negativo que deberíamos obtener. Como resultado perdemos la información relativa al signo del resultad. Y esto es debido a que el circuito presentado en [Dra00] realiza la suma módulo N, exactamente el mismo resultado que estamos obteniendo en la expresión (4.60).

Para solucionar el problema podemos ampliar el esquema para que realice la suma módulo $2N$, siguiendo el procedimiento mostrado en la sección 4.3. Es decir, si tenemos dos números a y b, cuya expansión binaria consta de n dígitos de los cuales uno representa el signo, en nuestro sumador representaremos b mediante n qubits y haremos la QFT y $n+1$ qubits $|0\rangle, |a_1\rangle, \ldots, |a_n\rangle$. Esto es equivalente a representar a con $n+1$ bits. Poste-

riormente aplicaremos el bloque sumador como hemos explicado en el apartado 4.3. De esta manera se podrá calcular la suma de dos enteros con signo sin perder la información del mismo.

Capítulo 5

Multiplicador QFT

En el capítulo anterior hemos presentado el sumador QFT y propuesto una serie de ampliaciones que permitan realizar operaciones como la suma de k números, la media o la suma ponderada. También hemos visto que el esquema realiza la suma módulo 2^n, pero que se puede ampliar con qubits auxiliares para aumentar el módulo y que recupere la suma aritmética. En este capítulo vamos a tomar como punto de partida el sumador para implementar otra operación aritmética, la multiplicación.

5.1. Multiplicación

Se propone crear un multiplicador cuántico que emule el proceso de la multiplicación binaria de dos números sin signo. En el sistema binario, el resultado de multiplicar dos bits solo da como resultado 1 cuando ambos bits son 1. Teniendo en cuenta esta premisa, veamos cómo multiplicar dos números binarios de n bits.

Supongamos que tenemos dos números binarios de $n = 3$ bits, por ejemplo, $a = a_1a_2a_3 = 110$ y $b = b_1b_2b_3 = 110$, que en el sistema decimal equivalen a 6 y 6. El producto $a \cdot b$ se puede calcular mediante el siguiente proceso:

$$\begin{array}{r} 110 \\ \times\, 110 \\ \hline 000 \\ 110 \\ +\, 110 \\ \hline 100100 \end{array} \tag{5.1}$$

Es decir, primero se calcula el producto ab_3 y posteriormente se le suma el producto ab_2, desplazado una posición a la izquierda, y el producto ab_1, desplazado dos posiciones a la izquierda. El resultado obtenido es 100100, que en el sistema decimal se corresponde con el número 36. Comprobamos que el resultado final está compuesto por $2n = 6$ dígitos binarios. Otra manera de explicar como se realiza el producto es mediante la siguiente suma:

$$\begin{array}{r} 110 \\ \times\ 110 \\ \hline 000 \\ 1100 \\ +\ 11000 \\ \hline 100100 \end{array} \tag{5.2}$$

El primer sumando es 000, que es el resultado de multiplicar 110×0. El segundo sumando es 1000. Este sumando es el resultado de multiplicar 100×1, añadiendo un 0 al final. Por tanto, lo que estamos haciendo es:

$$(110 \times 1) \times 10 = 1000, \tag{5.3}$$

es decir, estamos multiplicando el número por 10, que en el sistema decimal equivale a multiplicarlo por 2. El tercer sumando es 10000, que se obtiene de multiplicar:

$$(110 \times 1) \times 100 = 100100, \tag{5.4}$$

lo que en el sistema decimal implica multiplicar por 4. Realizamos las tres sumas y comprobamos que al dígito situado más a la izquierda, que es un 1, hay que sumarle otro 1 correspondiente a la llevada generada por la suma de los términos situados más a la derecha. Por tanto, el tamaño del resultado final es $2n = 6$.

Podemos generalizar este proceso para multiplicar dos números de n bits $a_1 \dots a_n$ y $b_1 \dots b_n$. El resultado vendrá dado por:

$$\begin{array}{rccccccc}
 & & & & & a_1 & a_2 & a_3 \ \dots\ a_{n-1}\ a_n \\
 & & & & & \times\ b_1 & b_2 & b_3 \ \dots\ b_{n-1}\ b_n \\
\hline
 & & & & a_1 \cdot b_n & a_2 \cdot b_n & a_3 \cdot b_n & \dots \quad a_{n-1} \cdot b_n \quad a_n \cdot b_n \\
 & & & a_1 \cdot b_{n-1} & a_2 \cdot b_{n-1} & a_3 \cdot b_{n-1} & \dots & a_{n-1} \cdot b_{n-1} \quad a_n \cdot b_{n-1} \\
 & & & & & \dots & & \\
 & & & & & \dots & & \\
 & & & & & \dots & & \\
 & & & a_1 \cdot b_3 & a_2 \cdot b_3 & a_3 \cdot b_3 & \dots \quad a_{n-1} \cdot b_3 & a_n \cdot b_3 \\
 & & a_1 \cdot b_2 & a_2 \cdot b_2 & a_3 \cdot b_2 & \dots \quad a_2 \cdot b_2 & a_n \cdot b_2 & \\
+ & a_1 \cdot b_1 & a_2 \cdot b_1 & a_3 \cdot b_1 & \dots \quad a_{n-1} \cdot b_1 & a_n \cdot b_1 & & \\
\hline
 & c_{p_1}(a_1 \cdot b_1 + c_{p_3}) & (a_1 \cdot b_2 + a_2 \cdot b_1 + c_{p_4}) & & \dots & (a_{n-1} \cdot b_n + a_n \cdot b_{n-1} + c_{p_n}) & & (a_n \cdot b_n)
\end{array} \tag{5.5}$$

Donde el término c_{p_i} que aparece en cada término representa la llevada generada en el término anterior. La expresión (5.5) es equivalente a realizar la siguiente suma:

$$
\begin{array}{r}
a_1 \;\; a_2 \;\; a_3 \;\; \dots \;\; a_{n-1} \; a_n \\
\times\, b_1 \;\; b_2 \;\; b_3 \;\; \dots \;\; b_{n-1} \;\; b_n \\
\hline
a_1 \cdot b_n \;\; a_2 \cdot b_n \;\; a_3 \cdot b_n \qquad \dots \qquad a_{n-1} \cdot b_n \;\; a_n \cdot b_n \\
a_1 \cdot b_{n-1} \;\; a_2 \cdot b_{n-1} \;\; a_3 \cdot b_{n-1} \;\; \dots \;\; a_{n-1} \cdot b_{n-1} \;\; a_n \cdot b_{n-1} \qquad 0 \\
\dots \qquad\qquad\qquad\qquad\qquad \\
\dots \qquad\qquad\qquad\qquad\qquad \\
\dots \qquad\qquad\qquad\qquad\qquad \\
a_1 \cdot b_3 \;\; a_2 \cdot b_3 \;\; a_3 \cdot b_3 \;\; \dots \;\; a_{n-1} \cdot b_3 \;\; a_n \cdot b_3 \qquad 000000\dots0000000000 \\
a_1 \cdot b_2 \;\; a_2 \cdot b_2 \;\; a_3 \cdot b_2 \;\; \dots \;\; a_2 \cdot b_2 \;\; a_n \cdot b_2 \qquad 0 \qquad 000000\dots0000000000 \\
+\, a_1 \cdot b_1 \;\; a_2 \cdot b_1 \;\; a_3 \cdot b_1 \;\; \dots \;\; a_{n-1} \cdot b_1 \;\; a_n \cdot b_1 \qquad 0 \qquad 0 \qquad 000000\dots0000000000 \\
\hline
c_{p_1} (a_1 \cdot b_1 + c_{p_3}) \;\; (a_1 \cdot b_2 + a_2 \cdot b_1 + c_{p_4}) \;\; \dots \;\; (a_{n-1} \cdot b_n + a_n \cdot b_{n-1} + c_{p_n}) \;\; (a_n \cdot b_n)
\end{array}
\tag{5.6}
$$

Donde el primer sumando está multiplicado por $1_2 = 2^0_{10}$, el segundo está multiplicado por $10_2 = 2^1_{10}$, el tercero está multiplicado por $100_2 = 2^2_{10}$ y así sucesivamente hasta llegar al n-ésimo sumando, que estará multiplicado por el equivalente en el sistema binario del número decimal 2^{n-1}, que será un número del tipo $100\dots00$, es decir, un 1 seguido de n 0s.

La suma se realiza normalmente, es decir, por filas comenzando por los términos situados más a la derecha hasta finalizar con los términos situados más a la izquierda. En caso de que en alguna de las filas la suma genere un término de llevada, este se suma a los términos situados inmediatamente a la izquierda.

El producto de dos números de n bits sin bit de signo dará como resultado un número que puede ocupar $2n - 1$ bits o $2n$ bits. Esto dependerá de si la suma $(a_1 b_2 + a_2 b_1)$ y del valor de $a_1 b_1$. Si la suma es tal que la llevada es 1 y $a_1 b_1 = 1$, al sumar ambos términos también nos llevamos 1, de forma que el resultado final constará de $2n$ bits. Por tanto, para evitar desbordamientos, fijaremos el tamaño de la multiplicación en $2n$ bits, de tal manera que el resultado final del producto de dos números de n bits será un número $p = p_1 p_2 \dots p_{2n-1} p_{2n}$, con

$$
\begin{aligned}
p_1 &= c_{p_2}; \\
p_2 &= a_1 b_1 + c_{p_3}; \\
p_3 &= a_1 b_3 + a_2 b_2 + a_3 b_1 + c_{p_4}; \\
&\vdots \\
p_{2n-1} &= a_{n-1} b_n + a_n b_{n-1}; \\
p_{2n} &= a_n b_n,
\end{aligned}
\tag{5.7}
$$

donde c_{p_i} denota el término de llevada generado en cada p_i. En adelante, al término p_1 se le llamará término o bit de llevada.

Proponemos implementaremos un multiplicador QFT que genere el producto de dos números a y b siguiendo esta aproximación de sumas secuenciales. Para ello codificaremos los números a multiplicar en los estados $|a\rangle$ y $|b\rangle$. Estos estados estarán formados por

n qubits $|a_1\rangle, \ldots, |a_n\rangle$ y $|b_1\rangle, \ldots, |b_n\rangle$, que se corresponderán con la expansión binaria de n dígitos de a y b. El resultado final será $|p\rangle = |p_1 p_2 \ldots p_{2n-1} p_{2n}\rangle$, donde cada $|p_s\rangle$ se corresponderá con la siguiente expresión:

$$c_{p_2} \quad , \quad s = 1; \tag{5.8}$$

$$c_{s+1} \quad + \sum_{j=1}^{s-1} |a_j\rangle |b_{s-j}\rangle, \quad 2 \leq s \leq n; \tag{5.9}$$

$$c_{s+1} \quad + \sum_{j=s-n}^{n} |a_j\rangle |b_{s-j}\rangle, \quad n < s \leq 2n. \tag{5.10}$$

En el siguiente apartado se presentará un esquema que implementa el proceso de multiplicación descrito mediante puertas QFT y puertas de rotación de fase controladas.

5.2. Esquema Propuesto

Una vez descrito el proceso de multiplicación que se quiere emular, pasamos a describir el circuito propuesto para implementar la multiplicación mediante puertas QFT y puertas de fase controladas. Este esquema puede verse en la figura 5.2.

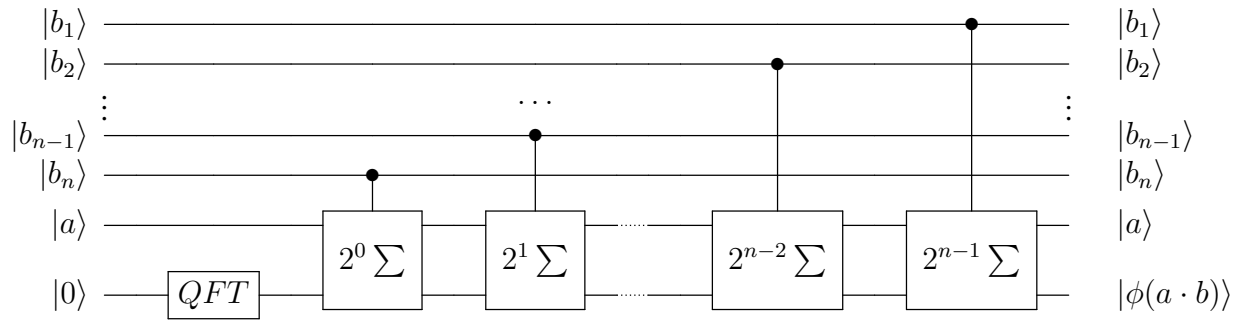


Figura 5.1: Multiplicador QFT.

El multiplicador trabajará con los estados $|a\rangle$, $|b\rangle$ y $|k\rangle$. El estado $|k\rangle$ se representará por su expansión binaria de $2n$ bits, mientras que los estados $|a\rangle$ y $|b\rangle$ estarán compuestos por n qubits $|a_1\rangle, \ldots, |a_n\rangle$ y $|b_1\rangle, \ldots, |b_n\rangle$, de tal manera que $a_1, \ldots, a_n$ y $b_1, \ldots, b_n$

forman la expansión binaria de n dígitos de a y b. Por tanto,

$$k = k_1 \cdot 2^{2n-1} + k_2 \cdot 2^{n-2} + \ldots + k_n \cdot 2^0 = \sum_{s=1}^{2n} k_s \cdot 2^{2n-s}; \tag{5.11}$$

$$a = a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_0 \cdot 2^0 = \sum_{j=1}^{n} a_j \cdot 2^{n-j}; \tag{5.12}$$

$$b = b_1 \cdot 2^{n-1} + b_2 \cdot 2^{n-2} + \ldots + b_0 \cdot 2^0 = \sum_{h=1}^{n} b_h \cdot 2^{n-h}, \tag{5.13}$$

$$\tag{5.14}$$

y los estados vienen representados por:

$$|0\rangle \rightarrow |0_1 0_2 \ldots 0_{2n-1} 0_{2n}\rangle; \tag{5.15}$$

$$|a\rangle \rightarrow |a_1 a_2 \ldots a_{n-1} a_n\rangle; \tag{5.16}$$

$$|b\rangle \rightarrow |b_1 b_2 \ldots b_{n-1} b_n\rangle. \tag{5.17}$$

Como vemos en la figura 5.2, el multiplicador consta de un primer bloque QFT que realiza la Transformada Cuántica de Fourier de $2n$ qubits $|0\rangle$. El resultado será:

$$|0\rangle^{\otimes 2n} \rightarrow \frac{1}{\sqrt{2^{2n}}} \sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}}(0 \cdot k)} |k\rangle, \tag{5.18}$$

transformando $2n$ qubits 0 que componen el estado $|0\rangle$ en $2n$ qubits $|0_1\rangle \ldots |0_{2n}\rangle$ en $|\phi_1(0)\rangle \ldots |\phi_{2n}(0)\rangle$ que componen el estado $|\phi(0)\rangle$, es decir, la QFT del estado $|0\rangle$.

La segunda parte del multiplicador implementará la parte de las sumas secuenciales que darán como resultado el producto de $a \cdot b$, mediante puertas de rotación de fase controladas, de tal manera que:

$$\begin{aligned} |\phi(0)\rangle &\rightarrow |\phi(a \cdot b)\rangle \\ &= \frac{1}{\sqrt{2^{2n}}} \sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}} \left(0 + a \cdot \sum_{h=1}^{n} b_h \cdots 2^{n-h}\right) \cdot k} |k\rangle. \end{aligned} \tag{5.19}$$

Como tenemos que hacer n sumas, emplearemos n bloques sumadores compuestos por puertas de rotación de fase. El primer sumador estará controlado por el qubit de b menos significativo, es decir, por $b_n \cdot 2^0$. Se debe construir de tal manera que implemente la siguiente suma:

$$0 + a = 0 + a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0. \tag{5.20}$$

Como al controlar la puerta por b_n solo se realiza la suma si $b_n = 1$, la expresión (5.21 es equivalente a:

$$\begin{aligned} suma_1 &= 0 + a \cdot b_n \cdot 2^0 = \\ &= 0 + b_n \cdot 2^0 \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right) = \\ &= 0 + 1 \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right). \end{aligned} \tag{5.21}$$

El segundo bloque estará controlado por el segundo qubit de b menos significativo, es decir, b_{n-1}. Por tanto, el bloque implementará la operación:

$$\begin{aligned} suma_2 &= suma_1 + a \cdot b_{n-1} \cdot 2^{n-1} = \\ &= suma_1 + a \cdot b_{n-1} \cdot 2^1 \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right). \end{aligned} \tag{5.22}$$

Si $b_{n-1} = 0$, la anterior expresión será 0. Si $b_{n-1} = 1$, la expresión anterior se puede reescribir como:

$$suma_2 = suma_1 + 2^1 \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right), \tag{5.23}$$

que, como podemos observar, es la expresión (5.21) multiplicada por 2^1. El tercer bloque estará controlado por el tercer bit menos significativo de b, es decir, $|b_{n-2}\rangle$ e implementará la suma:

$$\begin{aligned} suma_3 &= suma_2 + a \cdot b_{n-2} \cdot 2^2 = \\ &= suma_2 + b_{n-2} \cdot 2^2 \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right) = \end{aligned} \tag{5.24}$$

$$= suma_2 + 2^2 \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right), \tag{5.25}$$

de nuevo podemos comprobar que la expresión (5.25) es la expresión (5.21) por $2^2 = 4$. Con el resto de puertas sucede lo mismo, de tal manera que al llegar al bloque controlado por el qubit más significativo, es decir, $|b_1\rangle$, este debe implementar la suma:

$$\begin{aligned} suma_n &= suma_{n-1} + a \cdot b_1 \cdot 2^{n-1} = \\ &= suma_{n-1} + b_{n1} \cdot 2^{n-1} \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right) \end{aligned} \tag{5.26}$$

$$= suma_{n-1} + 2^{n-1} \cdot \left(a_1 \cdot 2^{n-1} + a_2 \cdot 2^{n-2} + \ldots + a_{n-1} \cdot 2^1 + a_n \cdot 2^0\right), \tag{5.27}$$

que es la suma que implementa el bloque controlado por el bit menos significativo multiplicada por 2^{n-1}.

Al final del circuito obtendremos el siguiente resultado:

$$0 + a \cdots b_n \cdots 2^0 + a \cdots b_{n-1} \cdots 2^1 + \ldots + a \cdots b_1 \cdots 2^0 = 0 + a \sum_{h=1}^{n} b_h \cdots 2^{n-h}. \tag{5.28}$$

Tras aplicar la QFT y los bloques sumadores, el resultado será:

$$
\begin{aligned}
|0_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.c_{p_2}(a_1\cdot b_1+c_{p_3})(a_1\cdot b_2+a_2\cdot b_1+c_{p_4})\ldots(a_{n-1}\cdot b_n+a_n\cdot b_{n-1}+c_{p_{2n}})(a_n\cdot b_n)}|1\rangle\right);\\
|0_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_1\cdot b_1+c_{p_3})(a_1\cdot b_2+a_2\cdot b_1+c_{p_4})\ldots(a_{n-1}\cdot b_n+a_n\cdot b_{n-1}+c_{p_{2n}})(a_n\cdot b_n)}|1\rangle\right);\\
|0_3\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_1\cdot b_2+a_2\cdot b_1+c_{p_4})\ldots(a_{n-1}\cdot b_n+a_n\cdot b_{n-1}+c_{p_{2n}})(a_n\cdot b_n)}|1\rangle\right);\\
&\vdots\\
|0_{2n-1}\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_{n-1}\cdot b_n+a_n\cdot b_{n-1}+c_{p_{2n}})(a_n\cdot b_n)}|1\rangle\right);\\
|0_{2n}\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_n\cdot b_n)}|1\rangle\right),
\end{aligned}
\tag{5.29}
$$

donde c_{p_i} es la llevada de la suma en el término situado inmediatamente a la derecha. Realizando al QFT^{-1} se recuperaría el resultado de multiplicar a y b. En las siguientes secciones veremos como construir cada bloque sumador.

5.3. Bloque Sumador

Hemos visto en la imagen 5.2 que el esquema general del multiplicador QFT está compuesto por varios bloques $2^{n-h}\sum$, que representan los bloques sumadores ponderados por el peso 2^{n-h} donde h se corresponde con el índice del qubit de control del bloque. Estos bloques sumadores están compuestos por varias puertas de rotación de fase controladas por $|a_1\rangle, |a_2\rangle, \ldots, |a_n\rangle$. El objetivo es implementar estos bloques de tal manera que realicen la operación

$$
\begin{aligned}
0 + a_1\cdot 2^{n-1} + a_2\cdot 2^{n-2} + \ldots + a_n\cdot 2^0 = \\
= 0 + \sum_{j=1}^{n} a_j\cdot 2^{n-j}.
\end{aligned}
\tag{5.30}
$$

Las puertas de rotación de fase deben estar diseñadas de tal manera que al aplicarlas sobre $|\phi(0)\rangle$ obtengamos el siguiente resultado:

$$
|0\rangle \rightarrow \frac{1}{\sqrt{2^{2n}}}\sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}}\left(a_j\cdot 2^{n-j}+0\right)\cdot k}|k\rangle,
\tag{5.31}
$$

pero como sabemos, $k = k_1\cdot 2^{2n-1} + k_2\cdot 2^{2n-2} + \ldots + k_n 2^0$, por lo que la expresión anterior se puede reescribir como:

$$
|0\rangle \rightarrow \frac{1}{\sqrt{2^{2n}}}\sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}}a_j\cdot 2^{n-j}\left(\sum_{s=1}^{2n} k_s\cdot 2^{2n-s}\right)}|k\rangle,
\tag{5.32}
$$

Así que emplearemos las puertas de rotación de fase:

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}. \tag{5.33}$$

Pero en este esquema cada bloque sumador está controlado, a su vez, por $|b_1\rangle, \ldots, |b_n\rangle$. Esto implica que las puertas de rotación cuentan en realidad con dos controles, en lugar de con un control como hemos visto hasta ahora. Las puertas con múltiples controles solo actúan sobre el qubit objetivo si todos los qubits de control son 1. En el caso que nos ocupa, estas puertas estarán controladas por los pares $(|a_1\rangle, |b_j\rangle), (|a_2\rangle, |b_j\rangle, \ldots, (|a_n\rangle, |b_j\rangle)$ respectivamente. Las puertas solo actuarán sobre el qubit objetivo si $|a_i\rangle = |b_j\rangle = |1\rangle$. Si representamos esto mediante una tabla de verdad:

$\|a_j\rangle$	$\|b_h\rangle$	$\|a_j\rangle\|b_h\rangle$
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 5.1: Producto de dos qubits.

vemos que la puerta estará controlada por el resultado de multiplicar los qubits $|a_j\rangle$ y $|b_h\rangle$. El efecto sobre el qubit objetivo será

$$|0\rangle \to \frac{1}{\sqrt{2^{2n}}} \sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}} a_j \cdot 2^{n-j} b_h \cdot 2^{n-h} \left(\sum_{s=1}^{2n} k_s \cdot 2^{2n-s} \right)} |k\rangle, \tag{5.34}$$

Por tanto, para recuperar el producto tendremos que introducir una fase a $|\phi_s(0)\rangle$ controlada por a_j y b_h del tipo:

$$e^{\frac{2\pi i a_j \cdot b_h \cdot k_s \cdot 2^{4n-j-h-s}}{2^{2n}}} = e^{2\pi i a_j \cdot b_h \cdot k_s \cdot 2^{2n-j-h-s}} = e^{\frac{2\pi i a_j \cdot b_h \cdot k_s}{2^{j+s+h-2n}}}. \tag{5.35}$$

Teniendo en cuenta que las puertas de rotación de fase, definidas por la expresión (5.33), introducen una fase del tipo $e^{\frac{2\pi i}{2^k}}$, las puertas que emplearemos vendrán dadas por $R_k = R_{j+s+h-2n}$.

Veamos ahora cómo construir los bloques. Para ello comenzamos explicando cómo se construye el bloque $2^0 \sum$, es decir, el bloque controlado por el bit menos significativo, b_n. Llamaremos a este bloque sumador básico, pues todos los demás se pueden obtener a partir de este bloque.

Calculamos primero las puertas controladas por a_1. La puerta que actúe sobre k_1 debemos obtener la expresión

$$\frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{2n}} a_1 \cdot 2^{n-1} b_n \cdot 2^0 k_1 \cdot 2^{2n-1}} |k\rangle = \frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{h+s-n}} a_1 \cdot k_1} |k\rangle = \frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{2-n}} a_1 \cdot b_n \cdot k_1} |k\rangle, \tag{5.36}$$

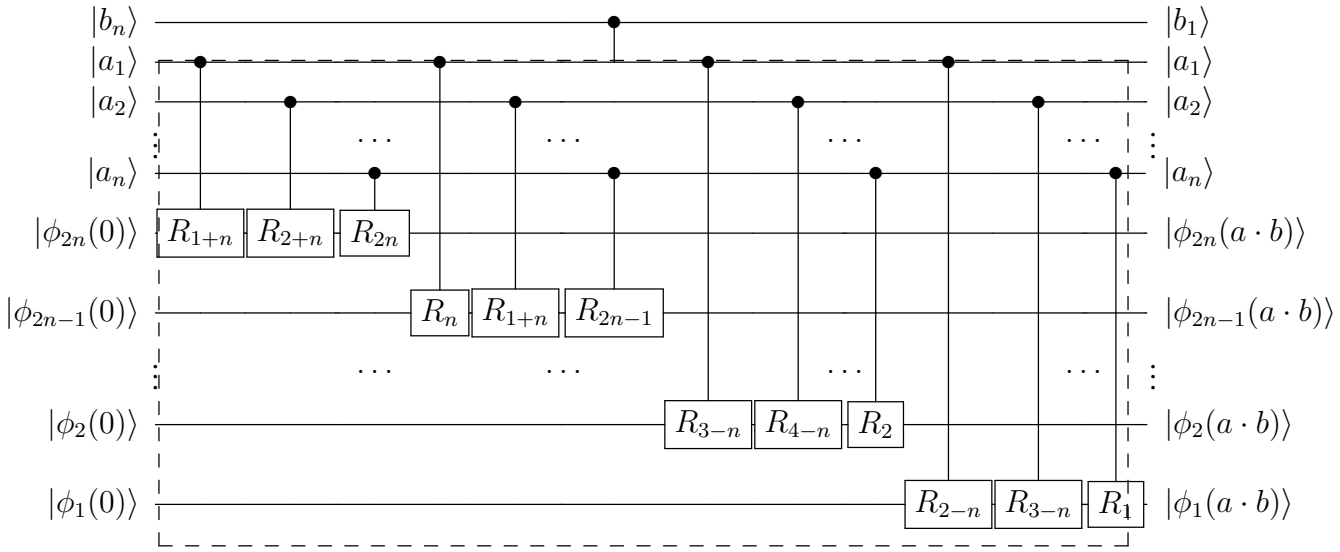


Figura 5.2: Bloque sumador básico.

donde $j = s = 1$ y $h = n$. R_k debe ser:

$$R_k = R_{j+s+h-2n} = R_{2-n} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{2-n}}} \end{bmatrix}, \tag{5.37}$$

Sobre k_2 debe actuar una puerta controlada por a_1 que dé como resultado:

$$\frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{j+s-n}} a_1 \cdot b_n \cdot k_2} |k\rangle = \frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{3-n}} a_1 \cdot b_n \cdot k_2} |k\rangle, \tag{5.38}$$

donde $j = 1$, $s = 2$ y $h = n$. La puerta de rotación de fase será:

$$R_k = R_{j+s+h-2n} = R_{3-n} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{3-n}}} \end{bmatrix}. \tag{5.39}$$

Y así sucesivamente, de tal manera que en k_{2n} debe actuar la puerta

$$R_k = R_{j+s+h-2n} = R_{1+n} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{1+n}}} \end{bmatrix}, \tag{5.40}$$

ya que en este caso $j = 1$, $s = 2n$ y $h = n$. El resultado será:

$$\frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{j+s-n}} a_1 \cdot b_n \cdot k_{2n}} |k\rangle = \frac{1}{\sqrt{2^{2n}}} e^{\frac{2\pi i}{2^{1+n}} a_1 \cdot b_n \cdot k_{2n}} |k\rangle. \tag{5.41}$$

Procedemos de igual manera para calcular las puertas de rotación de fase controladas por a_2. Como $R_{j+s+h-2n} = R_{2+s+n-2n} = R_{2+s-n}$, las puertas tienen que ser:

$R_{2+1-n} = R_{3-n}$ que actúa sobre k_1.

$R_{2+2-n} = R_{4-n}$ que actúa sobre k_2.

...

$R_{2+2n-n} = R_{2+n}$ que actúa sobre k_{2n}

Y seguimos el mismo procedimiento hasta llegar a a_n. Las puertas controladas por a_n serán:

$R_{n+1-n} = R_1$ que actúa sobre k_1.

$R_{n+2-n} = R_2$ que actúa sobre k_2.

...

$R_{n+2n-n} = R_{2n}$ que actúa sobre k_{2n}

El bloque sumador básico se muestra en la figura 5.2. Puede suceder que $j + s + h - 2n > 0$. En ese caso, la fase que tendremos será del tipo $e^{2\pi 2^{2n-j-s-h}i}$. Es decir, tendremos una fase múltiplo de $e^{2\pi i} = 1$, por lo que la puerta que actuará sobre el qubit en realidad será la matriz I y no tendrá ningún efecto sobre el qubit. A efectos prácticos, esta puerta no se implementaría.

A continuación se construye el bloque sumador $2^1 \sum$, que es el controlado por el segundo bit de b menos significativo, b_{n-1}, por lo que $h = n - 1$. Este bloque debe implementar la suma:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2^{2n}}} \sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{2n}} a_j \cdot 2^{n-j} b_{n-1} \cdot 2^1 \left(\sum_{s=1}^{2n} k_s \cdot 2^{2n-s} \right)} |k\rangle, \tag{5.42}$$

es decir, se implementa la misma suma que en el bloque controlado por el bit menos significativo, multiplicando las fases por 2. Por tanto, la expresión anterior es equivalente a:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2^{2n}}} \sum_{k=1}^{2n} e^{\frac{2\pi i}{2^{j+s-n-1}} a_j \cdot b_{n-1} \cdot \left(\sum_{s=1}^{2n} k_s \cdot 2^{2n-s} \right)} |k\rangle. \tag{5.43}$$

Y las puertas ha utilizar serán del tipo $R_k = R_{j+s-n-1}$. Por el mismo motivo, el tercer bloque, controlado por b_{n-2}, debe realizar una suma equivalente a multiplicar la fase de la suma que debe ejecutar el sumador básico por el término 2^2. Por tanto, las puertas que habrá que utilizar para implementar el sumador serán del tipo $R_k = R_{j+s+(n-2)-2n} = R_{j+s-n-2}$. El cuarto bloque deberá emplear puertas $R_k = R_{j+s-n-3}$ y así sucesivamente hasta llegar al n-ésimo bloque, que se construirá con puertas $R_k = R_{j+s-2n+1}$

5.3.1. Ejemplo

Vamos a construir un multiplicador que realice la operación $a \cdot b$. Suponemos que la expansión binaria de a y b está compuesta de $n = 2$ dígitos. Sumaremos dos términos:

$$b_2 \quad \cdot 2^0 \left(a_1 \cdot 2^1 + a_2 \cdot 2^0 \right) \quad y \tag{5.44}$$

$$b_1 \quad \cdot 2^1 \left(a_1 \cdot 2^1 + a_2 \cdot 2^0 \right). \tag{5.45}$$

El multiplicador se muestra en la figura 5.3. El primer paso es realizar la QFT de tamaño $2n$ del estado $|0\rangle$, es decir, de los $2n$ qubits 0 con los que se construye el estado $|0\rangle$, con los circuitos cuánticos descritos en los capítulos 2 y 4. Como sabemos esos esquemas realizan la transformación:

$$\begin{aligned}
|0_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}+\frac{0}{2^4}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0000}|1\rangle\right);\\
|0_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.000}|1\rangle\right);\\
|0_3\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.00}|1\rangle\right);\\
|0_4\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0}|1\rangle\right);
\end{aligned} \tag{5.46}$$

Posteriormente utilizamos puertas swap, que intercambian el estado entre dos qubits, para ordenar la salida. El resultado que obtenemos es:

$$\begin{aligned}
|0_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0}|1\rangle\right);\\
|0_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.00}|1\rangle\right);\\
|0_3\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.000}|1\rangle\right);\\
|0_4\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}+\frac{0}{2^3}+\frac{0}{2^4}\right)}|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0000}|1\rangle\right);
\end{aligned} \tag{5.47}$$

Ahora construimos el primer sumador, controlado por b_2 que es el bit de b menos significativo. Sobre $k_1 = 0_1$ actúan las puertas R_{j+s-n}, que se muestran en la tabla 5.2.

Rotaciones de fase que componen el sumador básico		
	a_1	a_2
$k_1 = 0_1$	$R_{1+1-2} = R_0$	$R_{2+1-2} = R_1$
$k_2 = 0_2$	$R_{1+2-2} = R_1$	$R_{2+2-2} = R_2$
$k_3 = 0_3$	$R_{1+3-2} = R_2$	$R_{2+3-2} = R_3$
$k_4 = 0_4$	$R_{1+4-2} = R_3$	$R_{2+4-2} = R_4$

Tabla 5.2: Puertas de rotación de fase que constituyen el sumador básico.

Como vemos en la tabla, el bit a_1 controla una puerta de rotación de fase sobre k_1 dada por R_0. Esta puerta introduce la fase $e^{2\pi i a_1} = 1$. Por tanto no realiza ningún cambio sobre el qubit y en la práctica esta puerta no se llega a aplicar. El resultado de aplicar estas puertas es:

$$
\begin{aligned}
|0_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(1\cdot a_1\cdot b_2+\frac{0+a_2\cdot b_2}{2^1}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_2\cdot b_2)}|1\rangle\right); \\
|0_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0+a_1\cdot b_2}{2^1}+\frac{0+a_2\cdot b_2}{2^2}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_1\cdot b_2)(a_2\cdot b_2)}|1\rangle\right); \\
|0_3\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0+a_1\cdot b_2}{2^2}+\frac{0+a_2\cdot b_2}{2^3}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.0(a_1\cdot b_2)(a_2\cdot b_2)}|1\rangle\right); \\
|0_4\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1}+\frac{0}{2^2}+\frac{0+a_1\cdot b_2}{2^3}+\frac{0+a_2\cdot b_2}{2^4}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.00(a_1\cdot b_2)(a_2\cdot b_2)}|1\rangle\right); \qquad (5.48)
\end{aligned}
$$

Una vez construido el bloque básico, pasamos a diseñar el segundo bloque sumador, controlado por b_1. Este se construye multiplicando las fases que introducen las puertas de rotación de fase por un término $2^1 = 2$, lo que supone que las puertas a utilizar vienen dadas por $R_{j+s-n-1}$. Las puertas con las que se debe implementar el sumador se encuentran en la tabla 5.3.

Rotaciones de fase que componen el segundo sumador		
	a_1	a_2
$k_1 = 0_1$	$R_{1+1-2-1} = R_{-1}$	$R_{2+1-2-1} = R_0$
$k_2 = 0_2$	$R_{1+2-2-1} = R_0$	$R_{2+2-2-1} = R_1$
$k_3 = 0_3$	$R_{1+3-2-1} = R_1$	$R_{2+3-2-1} = R_2$
$k_4 = 0_4$	$R_{1+4-2-1} = R_2$	$R_{2+4-2-1} = R_3$

Tabla 5.3: Puertas de rotación de fase que constituyen el sumador controlado por b_1.

Como sucedía en el sumador básico, sobre k_1 actúan dos puertas R_{-1} y R_0, que introducen las fases $e^{2\pi 2a_1} = 1$ y $e^{2\pi a_2} = 1$. Como estas puertas en realidad no hacen evolucionar el estado del qubit, no se implementan en la práctica. El resultado de aplicar

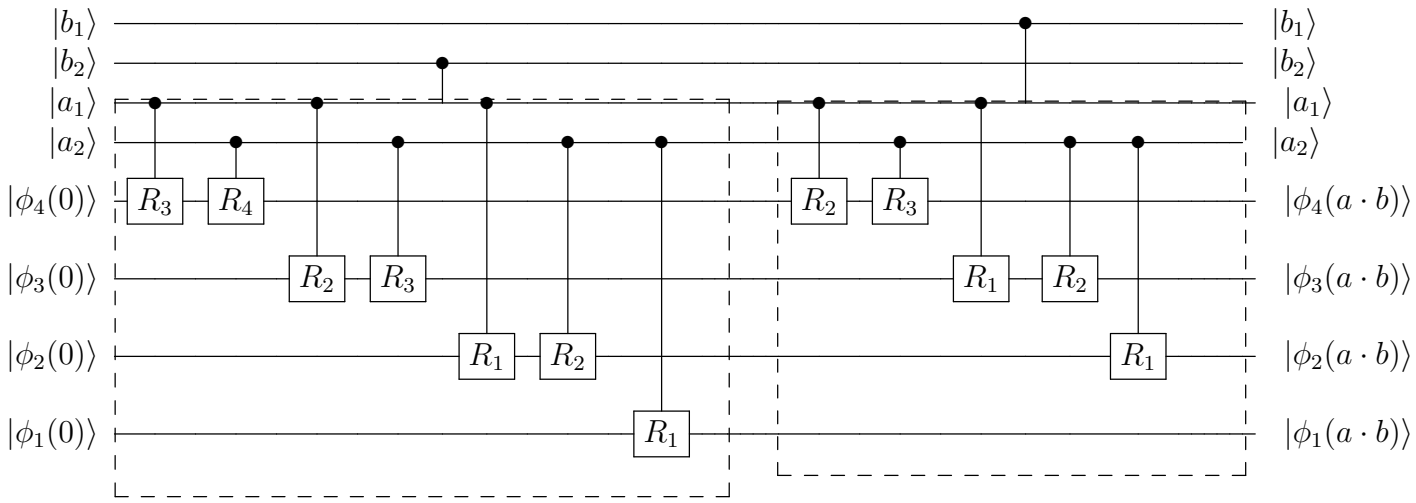


Figura 5.3: Multiplicador de dos números a y b de $n = 2$ bits.

estas puertas será:

$$
\begin{aligned}
|0_1\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(2\cdot a_1\cdot b_1 + 1\cdot(a_1\cdot b_2 + a_2\cdot b_1 + \frac{0+a_2\cdot b_2}{2^1}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_2\cdot b_2)}|1\rangle\right); \\
|0_2\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(1\cdot a_1\cdot b_1 + \frac{0+a_1\cdot b_2 + a_2\cdot b_1}{2^1} + \frac{0+a_2\cdot b_2}{2^2}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_1\cdot b_2 + a_2\cdot b_1 + c_{p_4})(a_2\cdot b_2)}|1\rangle\right); \\
|0_3\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0+a_1\cdot b_1}{2^1} + \frac{0+a_1\cdot b_2 + a_2\cdot b_2}{2^2} + \frac{0+a_2\cdot b_2}{2^3}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.(a_1\cdot b_1 + c_{p_3})(a_1\cdot b_2 + a_2\cdot b_1 + c_{p_4})(a_2\cdot b_2)}|1\rangle\right); \\
|0_4\rangle &\rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\left(\frac{0}{2^1} + \frac{0+a_1\cdot b_1}{2^2} + \frac{0+a_1\cdot b_2 + a_2\cdot b_1}{2^3} + \frac{0+a_2\cdot b_2}{2^4}\right)}|1\rangle\right) = \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i 0.c_{p_2}(a_1\cdot b_1 + c_{p_3})(a_1\cdot b_2 + a_2\cdot b_1 + c_{p_4})(a_2\cdot b_2)}|1\rangle\right);
\end{aligned}
\tag{5.49}
$$

Tras realizar la QFT inversa, el resultado que debemos recuperar es $c_{p_2}, (a_1 \cdot b_1 + c_{p_3})$,$(a_1 \cdot b_2 + a_2 \cdot b_1 + c_{p_4})$ y $(a_2 \cdot b_2)$. Veamos cual es el resultado que obtendríamos realizando la multiplicación manualmente:

$$
\begin{array}{r}
a_1 \;\; a_2 \\
\times \; b_1 \;\; b_2 \\
\hline
a_1 \cdot b_2 \;\; a_2 \cdot b_2 \\
+ \; a_1 \cdot b_1 \;\; a_2 \cdot b_2 \\
\hline
c_{p_2} \;\; a_1 \cdot b_1 + c_{p_3} \;\; a_1 \cdot b_2 + a_2 \cdot b_1 + c_{p_4} \;\; a_2 \cdot b_2
\end{array}
\tag{5.50}
$$

Es decir, que el resultado obtenido con el multiplicador QFT es el mismo que obtenemos haciendo la multiplicación manualmente.

5.4. Número de Puertas

Calculemos ahora el número de puertas necesarias para implementar el multiplicador. Cada bloque sumador consta de $2n \cdot n = 2n^2$ puertas, de las cuales ya sabemos que algunas en realidad no realizan ningún cambio sobre el qubit en el que actúan, por lo que en realidad sobran. La cuestión es calcular cuantas puertas no se utilizan en cada sumador, para lo cual vamos a comenzar calculando las puertas que sobran en el sumador básico.

En el primer qubit del sumador básico se implementan n puertas de las cuales $n-1$ no realizan ninguna acción. Sobre el segundo qubit, son $n-2$ los qubits que sobran. Así sucesivamente hasta llegar al qubit $n-1$ en el que sobra una puerta. En total, el número de puertas que no realizan ninguna acción es:

$$(n-1) + (n-2) + \ldots + 1 = \frac{n(n-1)}{2}. \tag{5.51}$$

Pasemos al segundo sumador, el controlado por el segundo bit menos significativo. Sobre el primer qubit de dicho sumador se implementan n puertas y ninguna de ellas realiza acción alguna sobre el qubit, por lo que sobran n puertas. Sobre el segundo qubit actúan n puertas de las cuales $n-1$ dejan al qubit sin alterar, sobre el tercer qubit son $n-1$ puertas las que no realizan ninguna acción y así sucesivamente hasta llegar al qubit n, en el que sobra una puerta. El número de puertas que sobran en el segundo bloque es:

$$n + (n-1) + (n-2) + \ldots + 1 = \frac{n(n+1)}{2}. \tag{5.52}$$

En el tercer bloque se repite la misma dinámica. Sobre el primer y segundo qubits se implementan n puertas respectivamente de las cuales ninguna opera sobre el qubit. A partir del tercer qubit son $n-1$ las puertas que no actúan sobre el mismo, en el cuarto son $n-2$ y así sucesivamente hasta llegar al qubit $n+1$ en el que sobra una puerta. En total:

$$n + (n-1) + (n-2) + \ldots + 1 = 2n + \frac{n(n-1)}{2}. \tag{5.53}$$

El proceso se repite en todos los bloques. El número total de puertas que sobra será:

$$\begin{aligned} n\frac{n(n-1)}{2} + n + 2n + \ldots + (n-1)n &= \\ &= \frac{n^2(n-1)}{2} + n(1 + 2 + \ldots + (n-1)) = \\ &= \frac{n^2(n-1)}{2} + \frac{n(n-1)}{2} = \\ &= n^2(n-1). \end{aligned} \tag{5.54}$$

Con lo que el número de puertas necesarias para implementar los n bloques sumadores será:

$$n \cdot 2n^2 - n^2(n-1) = 2n^3 - n^3 + n^2 = n^3 + n = n^2(n+1). \tag{5.55}$$

Mientras que el número de puertas necesarias para implementar la QFT viene dado por:

$$2n + \frac{2n(2n-1)}{2} = n(2n+1). \tag{5.56}$$

Por tanto, para realizar la QFT y la QFT inversa emplearemos $2n(2n+1)$ puertas. En total necesitamos $n^2(n+1) + 2n(2n+1) = n^3 + 5n^2 + 2n$ puertas.

El número de puertas es mayor que el número de puertas empleado en el multiplicador propuesto por Vedral en [VBE96]. Este multiplicador sigue la misma filosofía de implementar n sumas controladas por diferentes bits de b. El número de puertas empleadas en cada bloque es del orden de n, por lo que el número total será del orden de n^2. Por otra parte, el esquema de Vedral implementa en realidad la multiplicación módulo N por lo que en realidad no es justo comparar ambos esquemas. Sin embargo, podemos hacer que nuestro multiplicador calcule la multiplicación módulo 2^n haciendo la QFT de n qubits $|0\rangle$, es decir, sin añadir los n qubits 0 auxiliares que hemos empleado en el esquema. El resto del circuito se mantiene, es decir, se realizan n sumas mediante n bloques sumadores compuestos por puertas de rotación de fase $R_{j+s+h-2n}$.

En este caso es más sencillo calcular directamente el número de puertas que utiliza cada bloque sumador, en lugar de calcular las puertas que sobrar y restarlas del número total de puertas que se implementan.

El bloque sumador controlado por el bit más significativo, b_1, hace uso de una puerta. El sumador controlado por el segundo bit más significativo hace uso de $1+2=3$ puertas. El sumador controlado por el tercer bit más significativo, b_3, hace uso de $1+2+3=6$ puertas. Como vemos, En cada bloque se hace uso de $1+2+\ldots+h$ puertas, es decir:

$$1 + 2 + \ldots + h = \frac{h(h+1)}{2}. \tag{5.57}$$

Por tanto, el número total de puertas empleadas para construir los sumadores es:

$$\sum_{h=1}^{n} \frac{h(h+1)}{2} = \frac{n(n+1)(n+2)}{6}. \tag{5.58}$$

Además ahora hacemos la QFT y la QFT inversa de n qubits, lo que supone un total de puertas dado por:

$$2\left(n + \frac{n(n-1)}{2}\right) = n(n+1). \tag{5.59}$$

Por lo que el número total de puertas para implementar el multiplicador será:

$$n(n+1) + \frac{n(n+1)(n+2)}{6} = n(n+1)\frac{n+8}{6}. \tag{5.60}$$

Comparemos el esquema con el multiplicador de Vedral. El esquema está compuesto por n sumadores módulo $N = 2^n$. Cada uno de estos sumadores módulo N está a su vez compuesto por tres bloques sumadores y dos bloques restadores. Primero se aplica un bloque sumador que realiza la operación $a + b$. Posteriormente se aplica un restador que realiza la operación $a+b-N$. El resultado se almacena en un qubit auxiliar. Por último se aplican un sumador, un restador y otro sumador que restituyen el valor del qubit auxiliar y que dan como resultado $a + b \mod N$. Los bloques restadores son bloques sumadores colocados en orden inverso. Cada bloque sumador utiliza $8(n-1)+6$ puertas, por lo que un bloque sumador módulo N está compuesto por el siguiente número de puertas:

$$5(8(n-1)+6) = 40(n-1)+30. \tag{5.61}$$

Y utilizan $4n + 1$ qubits para realizar la operación, de los cuales $4n$ se utilizan para representar$a$, b, N y la llevada y el último se emplea como qubit auxiliar.

El multiplicador módulo N consta de n bloques sumadores módulo N controlados por $b_1, \ldots, b_n$ y por un qubit auxiliar, con el objetivo de diseñar un bloque multiplicador controlado. Es decir, cada sumador cuenta con dos controles, pero eso no afecta a la comparación que estamos realizando, pues el número de puertas que componen el circuito y nosotros podríamos hacer que nuestras puertas, que están controladas por dos qubits, pasasen a estar controladas por tres qubits. Por tanto emplea el siguiente número de puertas:

$$40n(n-1) + 30n. \tag{5.62}$$

Empleando para ello $5n + 2$ qubits, de los cuales $4n + 1$ se emplean en las sumas modulares y se añaden otros n qubits para el producto y un qubit auxiliar. Para que el multiplicador modular QFT emplee menos puertas que el multiplicador de Vedral, se debe cumplir que:

$$n(n+1)\frac{n+8}{6} - 40n(n-1) - 30n \leq 0. \tag{5.63}$$

Esta ecuación sólo tiene tres soluciones, $n = 0$, $n = 0.2$ y $n = 230.7053$. Es decir, para $n \leq 230$ el multiplicador modular QFT emplea menos puertas que el multiplicador de Vedral, mientras que para $n \geq 231$ es el multiplicador de Vedral el que emplea menos puertas, como se muestra en la figura 5.4. Sin embargo, siempre requerirá menos qubits el multiplicador modular QFT.

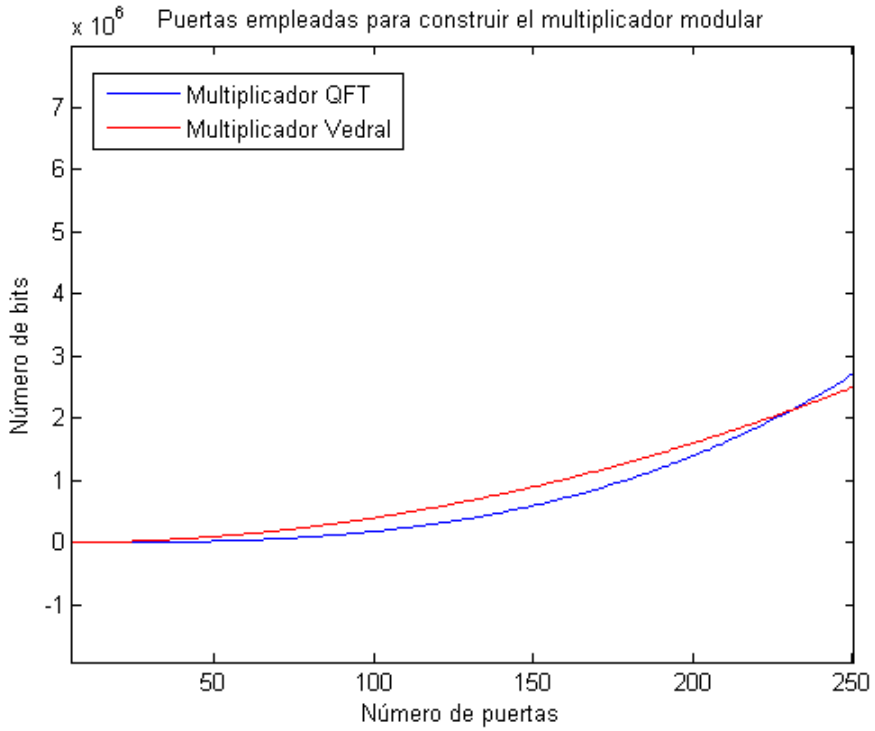


Figura 5.4: Número de puertas empleadas por el multiplicador modular QFT y el multiplicador propuesto en [VBE96].

Capítulo 6

Conclusiones

En este proyecto hemos trabajado en la implementación de operaciones aritméticas básicas con puertas cuánticas. El motivo es que existen operaciones aritméticas necesarias para la computación cuyo algoritmo clásico no se puede trasladar directamente al dominio cuántico, generalmente debido a que las operaciones lógicas que los componen no son operaciones válidas en computación cuántica.

Hemos comenzado revisando la bibliografía relacionada con los esquemas cuánticos que implementan operaciones aritméticas, centrándonos en sobre todo en los esquemas de suma y multiplicación. Hemos comprobado que la mayoría de los sumadores cuánticos propuestos se basan en el esquema clásico, en tanto que realizan la suma bit a bit utilizando bloques que emulan las operaciones AND y OR con los que se construyen los sumadores clásicos.

Sin embargo existe un esquema que emplea una aproximación distinta y utiliza un algoritmo cuántico que realiza la operación equivalente a la transformada discreta de Fourier, la llamada transformada cuántica de Fourier, para construir un sumador cuántico. Es lo que a lo largo del trabajo hemos llamado sumador QFT y fue propuesto por Thomas Draper en [Dra00].

El esquema suma dos números a y b. Para ello, se preparan dos estados $|a\rangle$ y $|b\rangle$ compuestos por n qubits cada uno de tal manera que $|a\rangle = |a_1\rangle \otimes \ldots \otimes |a_n\rangle$ y $|b\rangle = |b_1\rangle \otimes \ldots \otimes |b_n\rangle$. Por otra parte, $a_1, \ldots, a_n$ y $b_1, \ldots, b_n$ componen la expansión binaria de los números a y b respectivamente. Posteriormente realiza la QFT de $|a\rangle$, codificándolo como desfases entre los términos que conforman la QFT de $|a\rangle$, que llamaremos $|\phi(a)\rangle$. Por último, aplicamos rotadores de fase controlados por $|b\rangle$. Estos rotadores multiplican $|\phi(a)\rangle$ por una fase del tipo $e^{\frac{2\pi ibk}{2^k}}$. El resultado es que las fases de los términos son del tipo $e^{\frac{2\pi i(a+b)k}{2^k}}$, sumando los números $a + b$. Posteriormente se realiza la QFT inversa y se recupera $a + b$.

Este esquema realiza la suma módulo N, donde $N = 2^n$. Además no se discute ninguna ampliación para sumar k números ni como trabajar con números con signo. Por tanto, en este trabajo hemos propuesto extensiones al esquema que den respuesta a estos aspectos.

En primer lugar proponemos una extensión del esquema para sumar k números. Este

esquema realizaría la QFT únicamente de uno de los estados, y sumaría los demás utilizando puertas de rotación de fase con los índices empleados en el sumador QFT de dos números.

Como ya hemos dicho, el esquema realiza la suma módulo N, pero se podría realizar la suma añadiendo qubits para almacenar la llevada. Pero hay que determinar cuántos qubits se necesita añadir al esquema para evitar desbordamientos y recuperar la suma con llevada.

Si sumamos dos números de tamaño n, bastará con añadir un qubit más, que será un qubit 0. Hacemos la QFT de $|0a_1 \dots a_n\rangle$ y sumamos el otro número utilizando puertas de rotación de fase que vendrán dadas por:

$$R_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^{k+1}}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2 \cdot 2^k}} \end{bmatrix}. \tag{6.1}$$

Dado que el tamaño del resultado de la suma será de $n+1$ qubits, estaremos haciendo la suma módulo $2^{n+1} = 2N$. El número total de qubits que necesita el esquema para realizar la suma con llevada de dos números de tamaño n será $2n+1$.

En caso de sumar k números de tamaño n, el tamaño del resultado final vendrá dado por

$$log_2(k \cdot 2^n) = log_2(k) + log_2(2^n) = log_2(k) + n, \tag{6.2}$$

por lo que tendremos que añadir $log_2(k)$ qubits para habilitar espacio para la llevada. El esquema realizará la suma módulo $2^{n+log_2(k)}$ y empleará para ello $2n + log_2(k)$ qubits.

Si sumamos dos números de tamaños m y n con $n \neq m$, el resultado tendrá el tamaño $max(m,n)+1$. Así que tendremos que añadir un qubit 0 al número de mayor tamaño, y realizar la QFT de $|0a\rangle$. El sumador realizará la suma módulo $2^{max(n,m)+1}$ y empleará para ello $n+m+1$ qubits.

Hemos extendido el esquema para realizar la media y la suma ponderada. Se pueden realizar estas operaciones gracias a que podemos diseñar las puertas de rotación de fase controladas para que introduzcan el desfase que deseemos. Por tanto pueden introducir un desfase del tipo $e^{\frac{2\pi i}{M \cdot 2^k}}$ con las que se puede diseñar un circuito que calcule la media. También se puede diseñar una puerta que introduzca un desfase del tipo $e^{\frac{2\pi i \delta_i}{2^k}}$. Con este desfase se obtiene la suma ponderada.

Por último hemos discutido la representación de números con signo. Creemos que la representación signo-magnitud es la que más se adecúa para trabajar con el sumador QFT por su simplicidad y por ser la codificación natural. Pero si queremos operar con números con signo es necesario emplear el esquema extendido de suma con llevada, pues con el sumador QFT original se podría dar el caso de que se perdiera la información de signo por desbordamiento.

6.1. Multiplicador

Tomando como punto de partida el sumador QFT, se propone un circuito cuántico que realiza la multiplicación de dos números a y b. El circuito realiza la QFT de $2n$ qubits 0

que dan como resultado $|\phi_1(0)\rangle$,. . ., $|\phi_{2n}(0)\rangle$. De esta manera establecemos que el tamaño del resultado de multiplicar dos números de n bits sea $2n$.

En el siguiente paso, el circuito realiza n sumas $b_h \cdot 2^{n-h} \cdot \sum_{j=1}^{n} a_j \cdot 2^{n-j}$, donde n es el número de dígitos que componen la expansión binaria de a y b. Para realizar la suma emplearemos n bloques sumadores.

Definiremos un bloque básico que estará controlado por el qubit de b menos significativo, $|b_n\rangle$. Este bloque estará compuesto por puertas de rotación de fase controladass por $|a_1\rangle$, . . ., $|a_n\rangle$, de tal manera que las puertas de rotación que actúan sobre cada qubit $|\phi_s(0)\rangle$ están definidas por $R_k = R_{j+s-n}$, donde

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}. \tag{6.3}$$

Puesto que el bloque estará controlado por $|b_n\rangle$ solo funcionará si $b_n = 1$. Por otra parte, cada rotación de fase solo afectará a la fase del qubit sobre el que actúa si $a_j = 1$, siendo $|a_j\rangle$ el qubit que lo controla. Por tanto, cada rotación funcionará si $b_n \cdot a_j = 1$.

Los siguientes bloques sumadores se construyen a partir del bloque básico. Para ello hay que multiplicar cada fase introducida por 2^{n-h} siendo h el índice del qubit $|b_h\rangle$ que controla al bloque sumador. El resultado final es el producto $a \cdot b$, implementado realizando n sumas módulo 2^{2n}.

Este esquema se puede modificar para realizar la multiplicación módulo $N = 2^n$. Basta con realizar la QFT de n qubits 0 en lugar de utilizar $2n$ qubits auxiliares. Comparando el esquema con el multiplicador de Vedral [VBE96], el multiplicador modular QFT emplea menos puertas siempre que $n \leq 230$ y requiere menos qubits, $3n$ por los $5n + 2$ que utiliza el esquema de Vedral.

6.2. Líneas Futuras

En este trabajo hemos trabajado sobre operaciones aritméticas básicas realizadas con circuitos cuánticos. Hemos trabajado sobre la base del sumador QFT para proponer esquemas que sumen k números, que calculen la llevada y que realicen la media y la suma ponderada y posteriormente hemos propuesto un multiplicador basado en el sumador QFT. Pero no hemos discutido algunos aspectos como la complejidad de estos circuitos ni hemos estudiado si se pueden aplicar técnicas que permitan reducir el número de operaciones necesarias para implementar dichos circuitos. Estos aspectos se podrían estudiar en trabajos futuros.

Bibliografía

[BCDP96] David Beckman, Amalavoyal N Chari, Srikrishna Devabhaktuni, and John Preskill. Efficient networks for quantum factoring. *Physical Review A*, 54(2):1034, 1996.

[Bea02] Stephane Beauregard. Circuit for shor's algorithm using 2n+ 3 qubits. *arXiv preprint quant-ph/0205095*, 2002.

[DiV95] David P DiVincenzo. Quantum computation. *Science*, 270(5234):255–261, 1995.

[Dra00] Thomas G. Draper. Addition on a Quantum Computer. *arXiv preprint quant-ph/0008033.*, 2000.

[Gos98] Phil Gossett. Quantum carry-save arithmetic. *arXiv preprint quant-ph/9808061*, 1998.

[Knu14] Donald E Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms, The*. Addison-Wesley Professional, 2014.

[KTR14] S. Kotiyal, H. Thapliyal, and N. Ranganathan. Circuit for reversible quantum multiplier based on binary tree optimizing ancilla and garbage bits. In *VLSI Design and 2014 13th International Conference on Embedded Systems, 2014 27th International Conference on*, pages 545–550, Jan 2014.

[Mer07] N. David Mermin. *Quantum Computer Science. An Introduction.* Cambridge University Press, 2007.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2010.

[VBE96] Vlatko Vedral, Adriano Barenco, and Artur Ekert. Quantum networks for elementary arithmetic operations. *Physical Review A*, 54(1):147, 1996.

[Zal98] Christof Zalka. Fast versions of shor's quantum factoring algorithm. *arXiv preprint quant-ph/9806084*, 1998.