UNIVERSIDAD POTIFICIA CATÓLICA ARGENTINA

Computación cuántica: Análisis y ejecución de algorítmos cuánticos

Por

Felipe Rojo Amadeo

Prof. Ricardo Di Pasquale

Trabajo final presentado como parte del
título de grado

en

Ingeniería en Informática

Facultad de Ciencias Fisicomatemáticas e Ingeniería

17 de marzo de 2020

Declaración de autoría

Yo, Felipe Rojo Amadeo, declaro que el título de esta tesis, 'Computación cuántica: Análisis y ejecuciónde algorítmos cuánticos' y el trabajo presentado en el es de mi propiedad intelectual. Yo declaro que:

- Este trabajo fue realizado como parte de mi titulo de grado.
- Cuando he consultado el trabajo publicado de otros, esto siempre se atribuye claramente
- Donde he citado el trabajo de otros, siempre se da la fuente. Con la excepción de tales citas, esta tesis es completamente mi propio trabajo.

Firma:

Fecha:

"La mecánica cuántica describe la naturaleza como algo absurdo al sentido común. Pero concuerda plenamente con las pruebas experimentales. Por lo tanto espero que ustedes puedan aceptar a la naturaleza tal y como es: absurda."

Richard Feynman

Resumen

Las compañías más grandes en tecnología y software se encuentran focalizadas en un mismo objetivo: alcanzar la supremacía cuántica y operar con la mayor cantidad de qubits posible. Para entender la importancia que este desafío conlleva, bastará con entender, que alcanzarlo no solo será un logro a nivel científico, sino que tendrá un impacto muy fuerte en el ámbito político, económico y social. Es comparable a la carrera por llegar a la luna o por construir la primer bomba atómica.

Este trabajo esta dividido de la siguiente forma: En el capítulo 1 damos a conocer las principales áreas en donde la computación cuántica tendrá más impacto. En el capitulo 2 hacemos un breve repaso de la mecánica cuántica y las distintas formas de representar un procesador cuánticas, o al menos, en las que más se está trabajando hoy en día.

El siguiente capítulo (3) costa de postulados de computación cuántica, representación de los qubits y las compuertas que conforman sus circuitos. Continuando hacia el capítulo 4, describiremos los algoritmos de Deustch y Jozsa, el mismo generalizado, el de Simon y el de búsqueda de Grover.

Finalmente construiremos todos los algoritmos y los ejecutaremos en la plataforma de simulación de IBM.

No trataremos correlación de errores.

...

Agradecimientos

Aca van agradecimientos ...

Índice general

Índice de figuras

Abreviaciones

O_f	**O**ráculo
CNOT	**C**ontrolled **NOT**
CCN	**C**ontrolled-**C**ontrolled **NOT**
CSWAP	**C**ontrolled **SWAP**
H	**C**ontrolled **H**adamard
EQP	**E**qual **P**olinomial **T**ime
BPP	**B**ounded-error **P**robabilistic **P**olynomial **T**ime

Constantes Físicas

$$\text{Constante de Plank} \quad \hbar \quad = \quad 6{,}62607004 \times 10^{-34} \ \frac{m^2 kg}{s} \text{ (exacto)}$$

Símbolos

$\otimes$	Producto tensorial
$\oplus$	Exclusive OR
i	número imaginario $\sqrt{-1}$
$\perp$	Ortogonalidad entre vectores del espacio de Hilbert
.	Producto interno entre bits modulo 2
$U^\dagger$	Matriz Hermitiana o conjugada transpuesta
U^*	Matriz conjudada

Aca va dedicatoria...

Capítulo 1

Introducción

El 23 de octubre de 2019, la revista científica Nature publicó un artículo en el que Google afirma haber alcanzado la supremacía cuántica. Para apenas comenzar a comprender esto, podemos decir que la supremacía cuántica consiste en resolver un problema con una computadora exclusivamente cuántica, de modo tal que con una clásica sería imposible o tardaría miles de años. El experimento al que se hace referencia arrojó que una máquina cuántica requirió 200 segundos en resolver un problema que una computadora clásica, con 1 millón de núcleos, tardaría 10 mil años.

Esto es solo el comienzo. Estamos en el umbral de una nueva era y al igual que la computación clásica, se podrán resolver problemas de todas las disciplinas.

Destacamos las áreas que tendrán mayor impacto:

- Medicinal, biología y genética
  En estas áreas nos podremos enfrentar a ciertos problemas que hoy resultan imposibles atacarlos, debido a la complejidad de los mismos. Por ejemplo, en lo que se refiere a la cura del cáncer, se podrá simular el comportamiento de la infinidad de células malignas y sus complejas relaciones, para obtener distribuciones de cómo pueden esparcirse por el cuerpo para anticiparse y atacarlas de manera eficiente.

- Big Data, AI, Machine Learning
  En 2013, la cantidad de datos producidos fue de 4.4 zettabytes, siendo analizados solo el $0,5\,\%$ de los mismos. Cabe aclarar que solo el $22\,\%$ hubiese sido útil. Hoy se producen 44 zettabytes, 10 veces más que en 2013, y se esperan 163 zettabytes para 2025. Hoy se estima que un $37\,\%$ es información valiosa para ser analizada. El 97,2 % de las empresas invierten en Big Data y AI, y cada vez las apuestas son más fuertes. Como podemos observar, la disponibilidad de datos a ser analizados

es sobreabundante y las empresas están dispuestas a invertir grandes cantidades de dinero en estas tecnologías. En este campo, la computación cuántica jugará un papel fundamental, ya que permitirá analizar mayores volúmenes de información de manera más rápida y menos costosa. Tecnologías como AI y Machine Learning sacarán provecho de las nuevas computadoras para analizar, predecir, aprender, y procesar datos.

- Finanzas
  Hoy en día, cada avance en el mundo de las finanzas, cada modelo económico que se crea, se computaliza de manera inmediata, o directamente se desarrollan en computadora. Como hemos descrito en las disciplinas anteriores, la nueva computación traerá aparejado un nuevo potencial de cálculo, y al ser una disciplina tan digitalizada como lo son las finanzas, estamos seguros que va a ser candidata a que desaparezca la intervención humana. Se dejarán de lado las intuiciones y especulaciones, para darle lugar a los resultados sobre el análisis de gran cantidad y diversidad de datos. Por ejemplo, el banco español CaixaBank fue pionero en realizar proyectos con computación cuántica, en particular, adaptar un algoritmo cuántico para evaluar el riesgo en carteras hipotecarias y bonos del Tesoro.

Podríamos mencionar un sinfín de ciencias, ingenierías, tecnologías, disciplinas, entre otras áreas que potenciará la computación cuántica, pero también, sin lugar a duda, emergerán nuevas. Podría dar dos ejemplos que pertenecen a la naturaleza misma de la mecánica cuántica: la teletransportación[1] y la seguridad en redes. La primera se basa en dos principios fundamentales, el entrelazamiento cuántico y los estados de Bell, y la segunda, del principio de no clonación, los cuales veremos brevemente en el primer capítulo. Quiero aclarar, que aunque la seguridad en redes es un tema que lleva décadas de estudio, la seguridad en computación cuántica es algo totalmente diferente por la naturaleza física que esta conlleva.

En el capitulo 2, haremos un breve recorrido desde el surgimiento de la mecánica cúantica, hasta donde se empezó a evaluar la posibilidad de que los efectos cúanticos podrían traer algún beneficio si se pensaban computacionalmente. Luego describiremos brevemente que camino tomaron las compañías más importantes para construir el procesador cuántico.

En el tercer capítulo, expondremos una introducción a la computación cuántica, seguido de sus postulados y la forma de como se representación los qubits. Expondremos también las principales compuertas cúanticas de un circuito.

[1] La teletransportación como tal no existe, solo en la ficción. Pero a través de ciertos efectos cuánticos, podemos construir canales de comunicación donde la información se obtiene de manera instantánea

Una vez habiéndonos introducido en la computación cuántica, en el cuarto capítulo iremos analizando los principales algoritmos, partiendo de los más simples a los más complejos. En este capítulo será donde por primera vez, veremos el increíble poder computacional de una máquina cuántica, y cómo supera, en algunos casos, exponencialmente a la clásica.

No nos quedaremos solo con analizar los algoritmos de forma teórica, en el quinto y último capítulo, ejecutaremos todos los algoritmos expuestos, el de Deutsch, su generalización Deutsch-Jozsa, el de Simon y Grover. Quedara pendiente el algoritmo de Shor, desarrollado por Peter Shor para factorizar numero primos. Correremos cada algoritmo mas de una funcion en la plataforma IBM Quantum Experience. Elegí la plataforma de IBM, por dos razones, principalmente porque es intuitiva y simple de utilizar, pero además, IBM es una de las compañías con mas prestigio cunado hablamos de computación cuántica, aunque, en mi opinión, la delantera la lleva Google por el hecho que describimos al principio: haber alcanzado la supremacía cuántica.

Capítulo 2

Mecánica Cuántica y Procesadores Cuánticos

En este capítulo nos introduciremos a los principios elementales que gobiernan la física cuántica. Sin ir mas lejos, deduciremos la ecuación de Schrödinger, la cual representa el comportamiento de las partículas a escala atómica. Luego presentaremos alguna de las partículas subatómicas mas utilizadas para construir procesadores cuánticos.

2.1. Introducción a la mecánica cuántica

2.1.1. Historia

Haremos un breve recorrido de la historia de la física cuántica, marcando los eventos más relevantes hasta llegar al nacimiento de la computación cuántica.

- Enero de 1900
  Hasta esta fecha, el modelo atómico no era universalmente aceptado, el electrón había sido descubierto apenas 3 años antes y no se sabia como interactuaba con otras partículas, ni donde estaba localizado. Un importante problema se refería a los colores emitidos por los átomos en un tubo de descarga. Nadie podía entender por qué diferentes átomos de gas brillaban en diferentes colores. Este y otros interrogantes eran materia de investigación de la comunidad científica. [I]

- 14 de Diciembre de 1900

[I] Dan Styer. A Brief History of Quantum Mechanics

Fue el día que Max Plank presentó su trabajo sobre la cuantización de la energía, relacionada con una constante universal, la cual se denominó constante de Plank. Esta teoría ayudó a resolver fenómenos naturales previamente inexplicables, como el comportamiento del calor en los sólidos y la naturaleza de la absorción de luz a nivel atómico.
Años mas tarde obtuvo el premio Nobel.

- 1905
  Albert Einstein, a la edad de 26 años, postuló que el total de la energía de un haz de luz es cuatizable. Concepto que le permitió formular el efecto fotoeléctrico por lo que casi 2 décadas mas tarde obtuvo el premio Nobel.
  A partir de aquí, otros científicos también empezaron a tratar la cuantización de la energía para la velocidad, posición, momento, entre otras variables.

- 1925
  Erwin Schrödinger terminó de escribir la formula la cual describe de manera completa este sistema quántico y poder hacer predicciones

- 1980
  Aquí se gesta la computación cuántica. Paul Benioff publica una completa descripción de la maquina de Turing cuántica. Luego Richard Feynmanman y Yuri Manin sugirieron que la computación cuántica podría tener un potencial mayor por sobre la computación clásica.

2.1.2. El gato de Schrödinger

Una buena manera de dar el primer paso hacia la física cuántica, es comenzar por la famosa paradoja del gato de Schrödinger.

> Un gato está encerrado en una cámara de acero, junto con el siguiente dispositivo (que debe asegurarse contra la interferencia directa del gato): en un contador Geiger, hay una pequeña cantidad de sustancia radiactiva, tan pequeña, que tal vez en el curso de la hora uno de los átomos se descompone, pero también, con igual probabilidad, tal vez ninguno; Si sucede, el tubo contador se descarga y, a través de un relé, libera un martillo que rompe un pequeño matraz de ácido hidrocianico. Si uno ha dejado todo este sistema solo durante una hora, se podría decir que el gato aún vive si, mientras tanto, ningún átomo se ha descompuesto. La primera desintegración atómica lo habría envenenado. La función psi de todo el sistema lo expresaría al tener al gato vivo y muerto (perdón por la expresión) en partes iguales.

> Es típico de estos casos que una indeterminación originalmente restringida al dominio atómico se transforme en indeterminación macroscópica, que luego puede resolverse mediante observación directa. Eso nos impide aceptar tan ingenuamente como un "modelo borroso" válido para representar la realidad. En sí mismo, no representaría nada poco claro o contradictorio. Hay una diferencia entre una fotografía temblorosa o desenfocada y una instantánea de nubes y bancos de niebla. [II]

Básicamente, lo que Schrödinger quiso demostrar, es la diferente naturaleza entre un sistema físico clásico y cuántico. Clásicamente podríamos decir que el gato esta vivo o muerto al momento de abrir la caja, incluso antes de abrir la caja debería tener un estado determinado, pero cuánticamente, existe una probabilidad de que el gato este vivo y su complemento de que este muerto. Ahora si abrimos la caja, estaremos haciendo una medición, que hará que colapse el sistema, y así saber si el gato esta vivo o muerto [III].

2.1.3. La ecuación de Schrödinger

La paradoja anterior nos permite captar la contraintuitiva idea de que pueden coexistir dos o más estados en simultaneo y si queremos observar uno de ellos, debemos medirlo, lo cual perturbará el sistema y con cierta probabilidad quedará determinado en alguno de los estados.

Schrödinger fue capaz de expresar el comportamiento de un sistema cuántico en una ecuación, lo cual le valió el premio Nobel.

Derivaremos la ecuación dependiente del tiempo, siguiendo los pasos de David Morin en su libro "Waves", capítulo 10.2

Comenzamos por escribir la formula no relativista de la energía. La energía total del sistema es igual a la energía cinética más la energía potencial. Por el momento definimos la energía potencial en una sola dimensión:

$$E = K + V = \frac{1}{2}mv^2 + V(x) = \frac{p^2}{2m} + V(x) \tag{2.1}$$

Invocamos la ecuación de Broglie, la cual expresa la representación de partículas como una onda con frecuencia ω, y la relación Planck–Einstein, $E = \hbar\omega$ y el momento lineal dado por el número de onda k, $p = \hbar k$. Nos queda:

[II] Erwin Schrödinger, traducido por John D. Trimmer. THE PRESENT SITUATION IN QUANTUM MECHANICS: A TRANSLATION OF SCHRÖDINGER'S "CAT PARADOX PAPER". 1935

[III] Aquí seguimos la interpretación de Copenhague de la mecánica cuántica

$$\hbar\omega = \frac{\hbar^2 k^2}{2m} + V(x) \tag{2.2}$$

Una función de onda puede escribirse de forma exponencial así:

$$\psi(x,t) = Ae^{i(kx-\omega t)}$$

A continuación derivamos ψ con respecto al tiempo y luego derivamos dos veces con respecto a la posición:

$$\frac{\partial\psi}{\partial t} = -i\omega\psi \Rightarrow \omega\psi = \frac{i\partial\psi}{\partial t} \tag{2.3}$$

$$\frac{\partial^2\psi}{\partial x^2} = -k^2\psi \Rightarrow k^2\psi = -\frac{\partial^2\psi}{\partial x^2} \tag{2.4}$$

Multiplicamos la ecuación 2.2 por ψ

$$\hbar(\omega\psi) = \frac{\hbar^2(k^2\psi)}{2m} + V(x)\psi \tag{2.5}$$

Ahora reemplazamos ecuaciones 2.3 y 2.4 en 2.5

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t) \tag{2.6}$$

Como mencionamos previamente, la posición de esta ecuación esta representada en una sola dimensión, si quisiésemos expresarla en las dimensiones (x, y, z), deberíamos reemplazar $\frac{\partial^2\psi(x,t)}{\partial x^2}$ por el laplaciano $\nabla^2\psi$

Para llegar a esta ecuación, hemos asumido, entre otras cosas, el postulado de Broglie que establece que cada partícula tiene una función de onda asociada. Esto lo aceptamos como una verdad de fe. Lo que es importante destacar, es que mientras mas se experimenta, más se afirma la ecuación de Schrödinger en la teoría cuántica.

2.1.4. Principio de incertidumbre

En 1927, Werner Heisenberg probó que no es posible medir dos propiedades físicas en simultaneo de un mismo sistema cuántico. Si quisiésemos medir la velocidad y posición

de una partícula, mientras mas precisa es la medición de una de las propiedades, menos resulta la precisión de la otra. [IV]

Esto a nivel macro no pasa, de hecho, pasa lo contrario, ya que mientras mas exactas son las propiedades físicas que medimos en un sistema, más información obtenemos de las demás. Por ejemplo, es claro que podemos medir la velocidad y espacio recorrido de una moto sobre una carretera. Incluso, si mejoramos la medición de una variable, mas precisas se vuelven las otras.

El principio de incertidumbre no es un misterio de la naturaleza cuántica, lo que sucede es simplemente que al momento de medir una variable física, debemos perturbar la otra, por lo que si medimos esta última post-medición de la primera, obtendremos un resultado distorsionado.

2.2. Procesadores cuánticos

No todas las compañías han tomado la misma estrategia para construir procesadores cuánticos. Si bien, Google, IBM, Rigetti, IonQ, entre otras escogieron implementarlos en base a circuitos LC con superconductores y uniones de Josephson, otros han tomado un camino diferente. Intel utiliza quantum dot, Microsoft topológicas, D-Wave Annealing. A continuación describiremos brevemente las más relevantes:

2.2.1. Circuito LC con unión Josephson y Superconductores

La teoría de superconductividad fue propuesta por Bardeen, Cooper and Schrieffer en 1957. A diferencia de otros materiales, los materiales superconductores son aquellos en los que existe una temperatura crítica, para la cual, la resistencia es nula (Ver Figura 2.1). Esto se debe a efectos cuánticos, ya que en ciertos materiales y a partir de una temperatura crítica T_c, forman pares de electrones, llamados pares de Coopers. Los electrones tienen estados de spin como fermiones ($\frac{1}{2}, \frac{3}{2}$...), lo que implica que cada uno debe ocupar un estado diferente (principio de exclusión de Pauli). Los pares de Cooper, se comportan como Bosones, con spin enteros y no tienen que cumplir con el estado de exclusión de Pauli, lo que significa que los dichos pares pueden ocupar el mismo estado. Al ocupar el mismo estado, se dice que están condensados, como si fuesen una gran partícula, la cual se preservara ante pequeñas perturbaciones como el proceso de dispersión. En materiales no superconductores, los electrones no ocupan el mismo estado, como dijimos anteriormente, por lo cual se comportan de manera independiente y están expuestos a pequeñas

[IV] Werner Heisenberg. The Nobel Prize in Physics 1932. https://www.nobelprize.org/prizes/physics/1932/ceremony-speech/

perturbaciones debido al proceso de dispersión. Esto quiere decir que el momento M de un electrón puede cambiar a $M^{'}$, lo que se traduce en una degradación de la corriente.

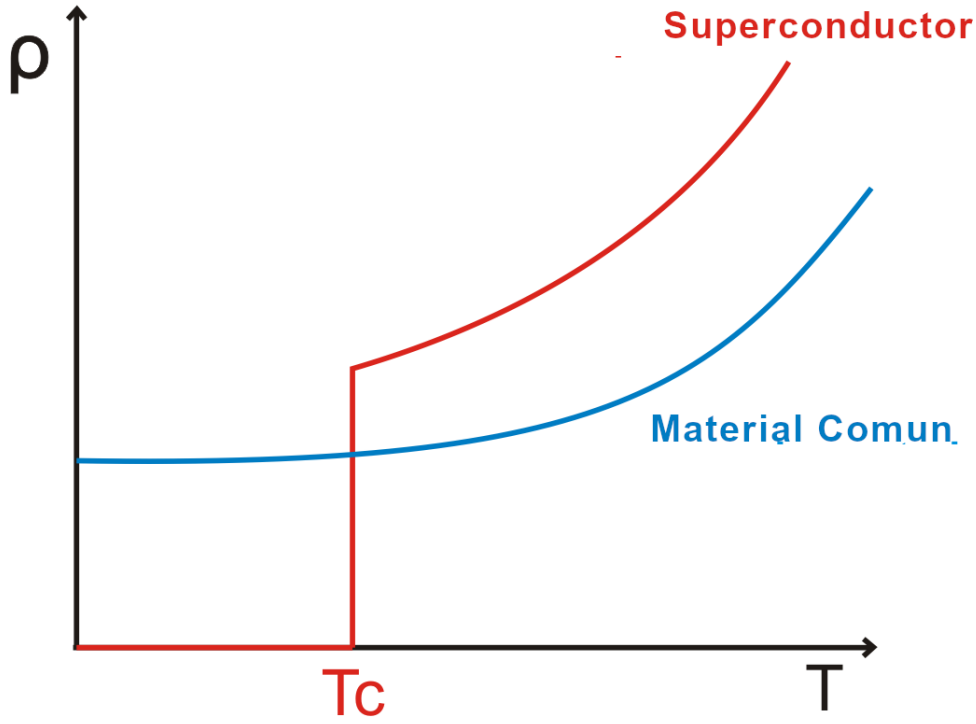

FIGURA 2.1: ρ : resistencia, T: Temperatura, T_c : Temperatura crítica

Los materiales superconductores son la base para construir nuestro procesador cuántico.

El circuito más usado por las compañías mencionadas al principio es el circuito LC, pero sin un elemento clave no iba a ser posible construir el procesador cuántico. Este elemento es la unión de Josephson, o efecto de Josephson.

Si miramos la figura 2.2, lo primero que notamos es que tenemos muchos niveles de energía equidistantes. Dichos niveles de energía pueden ser representados por un oscilador harmónico cuántico, análogo al oscilador clásico de un resorte y una masa en un movimiento periódico. Esto es un problema, ya que nosotros buscamos representar los bits 0 y 1 con el estado de energía mas bajo y el siguiente respectivamente. Lo que sucede es que el circuito puede tomar cualquier otro estado con igual probabilidad.

Aquí es donde entra en juego la teoría de Brian David Josephson, llamada efecto Josephson, representada en un circuito por la unión Josephson, que le valió el premio nobel en el año 1973 [v]. El fue el primero en predecir el efecto túnel de pares de Cooper en superconductores.

[v] B.D.Josephson. Possible new effects in superconductive tunnelling. https://www.sciencedirect.com/science/article/abs/pii/0031916362913690

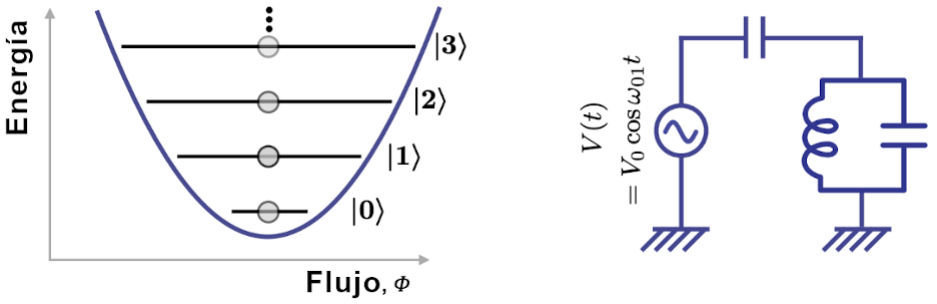

FIGURA 2.2: El el gráfico de la izquierda están representados los diferentes niveles de energía del circuito, en el otro el circuito

Para construir la unión de Josephson, bastará con tomar dos superconductores, aislados por una fina capa de aislante.

Incorporando la unión Josephson, rompemos con la linealidad del circuito y veremos representados los niveles de energía no equidistantes como vemos en la figura 2.3 y así poder controlar los estados $|0\rangle$ y $|1\rangle$ de manera no ambigua.

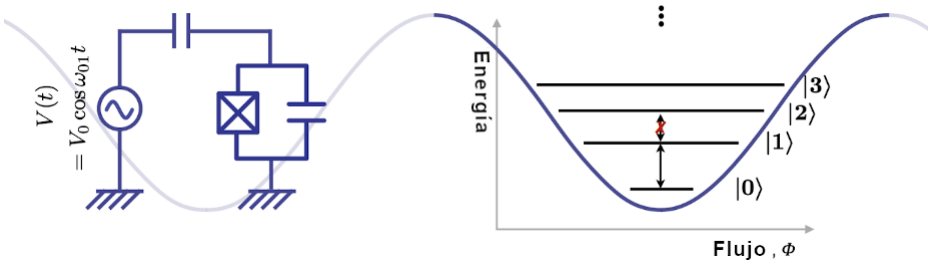

FIGURA 2.3: A la izquierda vemos el circuito con la unión Josephson y al otro lado, los distintos niveles de energía no lineales

Nuestro qubit quedará formado por el circuito y la frecuencia de transición de qubit depende de las capacitancias e inductancias en el circuito. Las operaciones cuánticas se realizan enviando impulsos electromagnéticos a frecuencias de microondas (alrededor de 4–6 KHz) al resonador acoplado al qubit. Esta frecuencia resuena con la separación de energía entre los niveles de energía para $|0\rangle$ y $|1\rangle$.

Existen distintas formas de representar los niveles de energía del circuito, estos son: carga Qubit, flujo Qubit y fase Qubit.

Una de las ventajas de utilizar superconductores para construir un "átomo" artificial, es que tenemos mucho conocimiento teórico y practico en ese área. La de-coherencia sigue siendo una barrera que relentiza el escalamiento cuántico, no solo en este tipo de procesadores, sino en la mayoría. El enfriamiento del circuito a muy bajas temperaturas es un proceso complejo a tener en cuenta.

2.2.2. Topológicos

Hoy, principalmente Microsoft se encuentra desarrollando esta tecnología. Tomó un gran desafío por su complejidad, pero muy prometedor.

La topología es una rama de las matemáticas que describe cómo un sistema experimenta cambios físicos como curvatura, estiramiento, compactación, o cualquier otra deformación, pero se mantienen ciertas propiedades de la forma original. Pequeños o suaves cambios en la forma no cambian la topología de un objeto, sin embargo, cambios violentos si. Por ejemplo, rebanar una pequeña parte de la superficie de una dona no alterará la topología, sin embargo, cortarla a la mitad, si. Esto aplicado a la computación cuántica, nos dice que las propiedades topológicas crean un nivel de protección que ayuda a retener información más allá de las condiciones del ambiente, o sea, es más tolerante a fallas que otros tipos de procesadores.

Los procesadores topológicos están formados por cuasipartículas llamadas anyons, las cuales ocurren solo en un sistema de dos dimensiones espaciales.

Intercambiando dos Anyons no abelianos, es equivalente a realizar una operación unitaria sobre el systema, lo que lleva a un cambio de fase, y en particular de Anyons no Abelianos, a un cambio de estado.

Si consideramos $2 + 1$ dimensiones, con el tiempo como tercera dimensión, veremos la evolución del sistema (Figura 2.4) en forma de trenzas, las cuales, si son topológicamente equivalentes, corresponderán a una misma operación unitaria sobre el sistema. Topológicamente equivalentes quiere decir que las trenzas pueden deformarse o estirarse, pero seguirán siendo topológicamente las mismas.

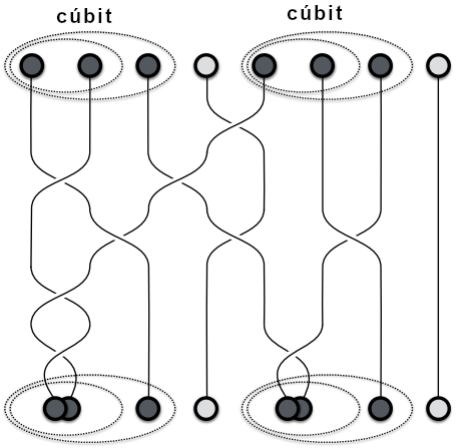

FIGURA 2.4: Evolución en el tiempo de dos qubits formados por cuatro anyons

Trenzando los Anyons de cada qubit alcanzamos la superposición de estados lo que implica que medir un qubit, afecta el estado de los demás. Para representar las diferentes

compuertas cuánticas, lo hacemos guardando las trenzas que sabemos que arrojaron el estado deseado.

Para medir el sistema, lo que hacemos es fusionar dos Anyons de cada qubit y observamos la salida de cada uno.

Unos de los modelos mas simples para representar grupos de Anyons, es el de Anyons de Fibonacci. Se representan los estados $|0\rangle$ y $|1\rangle$ como vacío (ausencia de la partícula) y el Anyon no trivial respectivamente.

La principal ventaja de un procesador topológico sobre los demás es la mayor estabilidad. Pequeñas perturbaciones que llevan a la de-coherencia de cualquier otro procesador cuántico, en este, no impacta en las trenzas del sistema, a menos que estas perturbaciones sean fuertes. Varias instituciones han comenzado a invertir en investigar las distintas piezas de conforman un procesador topológico. Últimamente se han hechos progresos significativos en el área, tanto teórico como práctico, lo que ha llevado a este tipo de procesador a ser una de las mayores promesas a mediano plazo.

2.2.3. Annealing

D-Wave posee un procesador cuántico Anneling, el cual corre algoritmos adiabáticos cuánticos. Los principales problemas que resuelve eficientemente son de búsqueda y encontrar el mínimo. Pero se destaca predominantemente en problemas donde hay muchas soluciones, de las cuales puede encontrar una suficientemente buena.

Problemas como obtener la ruta mas eficiente para un vendedor que visita muchas ciudades o ¿debería enviar mi paquete en este camión o en el siguiente? Son resueltos de manera eficiente con este tipo de computación cuántica, en donde encontrar el nivel mas bajo de energía (para un problema de optimización) o para otros problemas basta con encontrar algún mínimo local, no global.

La regla física fundamental que lidera este enfoque es que todo sistema tiende a buscar el mínimo nivel de energía. Esto lo vemos en un cuerpo caliente que se enfría, en una bola en la cima de una colina que rueda hacia abajo, lo mismo sucede en el mundo cuántico en términos de energía.

En problemas de muestreo, que en particular hoy se utiliza en machine learning, podemos obtener la forma de los distintos mínimos locales de energía que representan el estado actual del modelo en cuestión.

Para construir el procesador de Anneling, D-Wave implementa superconductores junto a un acoplador utilizado para entrelazar qubits y un bias que se usa para ejercer campos magnéticos direccionados para cambiar los estados de los qubits.

En la figura 2.5 vemos como evoluciona el sistema hasta alcanzar el estado desdado, o sea un mínimo, en este caso global.

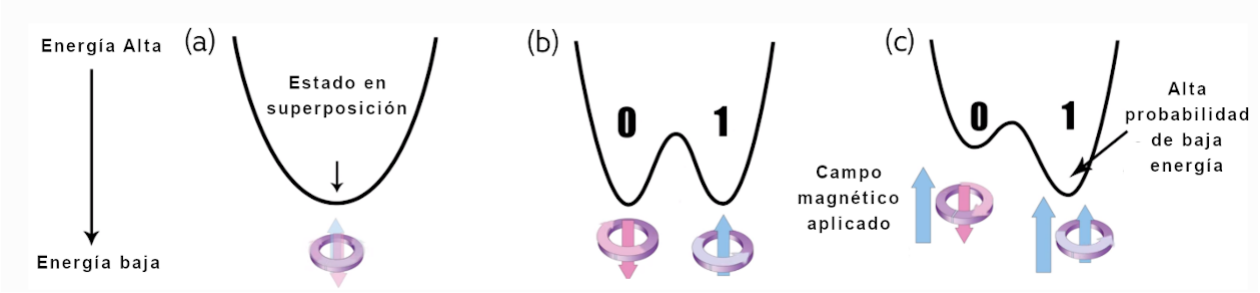

FIGURA 2.5: Evolución en el tiempo de un sistema compuesto por un qubit luego de aplicar un campo magnético (bias)

Para un sistema cuántico, un Hamiltoniano es una función que mapea ciertos estados, llamados eigenstates, a distintos niveles de energía. Cuando el sistema esta en un eigenstate del Hamiltoniano, decimos que se encuentra en un estado bien definido de energía, llamado eigenenergy.

El Hamilotniano es la suma de el Hamiltoniano inicial mas el Hamiltoniano final (al que queremos llegar). Entonces los pasos para resolver un determinado problema son:

- Hamiltoniano inicial: El estado de energía mas baja de este es cuando todos los qubits están en superposición $|0\rangle$ y $|1\rangle$. Este estado también es llamado Túnel Hamiltoniano.
- Hamiltoniano final: El nivel mas bajo de energía de este corresponde con la respuesta al problema planteado.

Idealmente, al final del proceso nos encontraremos en el nivel mas bajo de energía del Hamiltoniano final, obteniendo la respuesta a nuestro problema.

Entre el nivel mas bajo de energía y el siguiente, existe un espacio, el cual debemos evitar que el sistema salte a este ultimo. Hay varios factores que pueden hacer que se produzca un salto de energía, entre ellas esta las fluctuaciones térmicas y otra importante es que el proceso de Anneling se corra muy rápido. Un sistema que evoluciona en el tiempo de manera lenta sin interferencias de ningún tipo de fuente externa se denomina proceso Adiabático. Como no existe en la realidad un sistema totalmente aislado, Anneling es la contrapartida práctica del proceso teórico Adiabático.

2.2.4. Trapped ion

Trapped ions es otra forma de construir computadoras cuánticas de gran escala. Esta basada en ion trap, lo cual es un dispositivo que tiene como objetivo asilar iones o átomos con carga positiva suspendiéndolos en el espacio por medio de campos potenciales o magnéticos. La ion trap mas usada para implementar este procesador cuántico es Paul trap, la cual esta compuesta por campos eléctricos oscilatorios, producidos por electrodos a aproximadamente la frecuencia de radio, como podemos ver en la figura 2.6 Los campos son oscilatorios porque no es posible atrapar iones con campos estáticos.

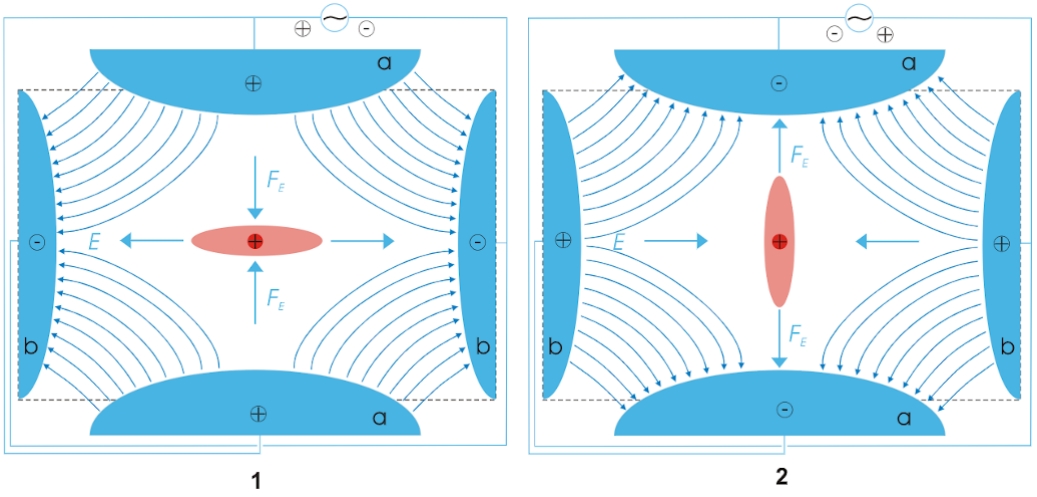

FIGURA 2.6: Del gráfico 1 al 2 vemos como se alternan oscilatoriamente los campos eléctricos para suspender el ion en el centro

Una vez atrapados son enfriados con una combinación de las técnicas Doppler cooling y Resolved sideband cooling. A bajas temperaturas es posible cuantificar la energía en fotones.

Hay dos maneras de formar un qubit

- qubits hiperfinos: Dos niveles hiperfinos en estado fundamental.
- qubits ópticos: Un nivel de estado fundamental y un nivel excitado.

Ambos enfoques tienen la ventaja de ser estables y longevos.

Para inicializar qubits, se utiliza el proceso llamado Optical pumping. Luego, el proceso de medición es con un láser apuntado al ion, cuando este colapsa, libera fotones y hemos medido un estado, si no emite fotones es porque se encuentra en el estado contrario.

Capítulo 3

Representación, enunciados y operaciones cuánticas

La computación cuántica explota ciertos efectos y leyes de la mecánica cuántica, así como análogamente la computación clásica el comportamiento de la fuerza electromagnética a gran escala. Cabe aclarar, que cuando hablamos del modelo clásico, hacemos referencia a la Maquina de Turing, la cual puede ser implementada de muchas maneras, sin embargo la mas famosa es el ordenador digital. Lo mismo sucede en el nuevo modelo, tenemos la Maquina de Turing Cuántica como modelo abstracto, pero esta puede ser representada por fotones, electrones, iones, entre otras partículas a escala subatómica.

3.1. Qubits y postulados de la Computación Cuántica

3.1.1. Qubit y su representación

Un qubit está en estado superpuesto de 0 y 1

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$$

Cuando el qubit es medido, tomará el valor de 0 o 1, dependiendo de los valores a_0 y a_1. Es una medición probabilística.

Todos los sistemas cuánticos están descriptos por un vector de estado representado por notación de Dirlac KET.

$$|\psi\rangle = a_0 |\psi_0\rangle + a_1 |\psi_1\rangle + ... + a_{N-1} |\psi_{N-1}\rangle = \sum_{i=0}^{N-1} a_i |\psi_i\rangle$$

También podemos describir el vector de estado como un vector de sus coeficientes complejos:

$$|\psi\rangle \equiv \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{bmatrix}$$

El estado conjugado se describe con notación Dirlac BRA y sus coeficientes conjugados:

$$\langle\psi| = \sum_{i=0}^{N-1} a_i^* \langle\psi_i| \equiv \begin{bmatrix} a_0^* & a_1^* & \dots & a_{N-1}^* \end{bmatrix}$$

Al medir el sistema, obtendremos un solo valor, 0 o 1, con las siguientes probabilidades:

$$p(0) = a_0^* \cdot a_0$$

$$p(1) = a_1^* \cdot a_1$$

$$p(0) + p(1) = a_0^* \cdot a_0 + a_1^* \cdot a_1 = 1$$

Luego de la medición, el sistema permanecerá en el estado asociado con el resultado obtenido.

3.1.2. Transformaciones unitarias

Para pasar de un estado a otro utilizamos algún operador unitario y lo podemos representar como una matriz cuadrada

$$U |\psi\rangle = |\psi'\rangle$$

Siendo U

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ u_{N1} & u_{N2} & \dots & u_{NN} \end{bmatrix}$$

La matriz de transformación unitaria satisface las condiciones

$$U^\dagger U = I$$

$$U^\dagger_{ij} = U^*_{ji}$$

Una propiedad importante es que las transformaciones unitarias son siempre reversibles.

3.1.3. Sistemas compuestos por el producto tensorial

Para unir dos sistemas o más, necesitamos:

1. Formar las bases orto-normales para el nuevo sistema
2. Determinar las amplitudes del nuevo sistema

Para satisfacer estas dos condiciones utilizamos el producto tensorial:

$$|\omega\rangle = |\psi\rangle \otimes |\phi\rangle$$

Supongamos que el sistema $|\psi\rangle$ es un registro de N bits y el sistema $|\phi\rangle$ de M bits, luego el sistema compuesto por ambos será:

$$|\psi\rangle \otimes |\phi\rangle = \sum_{ij}^{N+M} a_i \cdot b_j \, |\psi_i \phi_j\rangle$$

3.1.4. Entrelazamiento

Es un efecto de la física cuántica, donde dos o mas partículas se combinan para formar un único sistema. El cambio en una parte del sistema afecta a todo el sistema. Es la propiedad fundamental del que se nutren los algoritmos cuánticos

3.1.5. Estados de Bell

Los estados de Bell tienen la particularidad de que al medir uno de los qubits, obtendremos el otro con 100 % de certeza. Es posible construir estos estados a través de compuertas, por ejemplo para el primer estado con la compuerta Hadamard y Controlled Not llegamos al mismo. Existen cuatro estados de Bell. En este caso representado por 2 qubits:

- $|\psi_1\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$
- $|\psi_2\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}}$
- $|\psi_3\rangle = \frac{|10\rangle+|01\rangle}{\sqrt{2}}$
- $|\psi_4\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}}$

En el primer caso, si medimos el primer qubit y es 0, instantáneamente, el segundo qubit también será 0.

3.1.6. Teorema de no clonación

Supongamos que quisiéramos copiar un registro en otro:

$$U(|\psi\rangle\,|\phi\rangle) = |\psi\rangle\,|\psi\rangle$$

Para esto deberíamos encontrar una matriz U unitaria que satisfaga esta ecuación. Desarrollando la formula anterior podríamos verificar que no es posible obtener dicha matriz que copie un registro de qubits en otro.

3.2. Compuertas Cuánticas

En esta sección vamos a presentar las principales compuertas cuánticas y en especial las necesarias para construir los algoritmos cuánticos que presentaremos en el próximo capítulo. Todas las compuertas, expresadas como matrices unitarias, son reversibles.

3.2.1. Compuerta NOT

La compuerta NOT, tal como la conocemos en computación clásica, no es reversible. Cuando hablamos en computación cuántica sobre la compuerta NOT, nos referimos a la matriz unitaria aplicada a un qubit:

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

Esta transformación aplicada a un qubit:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_0 \end{bmatrix}$$

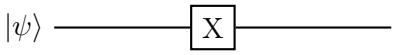

FIGURA 3.1: Compuerta NOT

3.2.2. Compuerta CNOT

La compuerta Controlled NOT o CNOT es una matriz unitaria que opera sobre dos qubits:

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

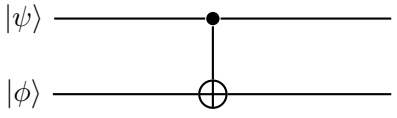

FIGURA 3.2: Compuerta CNOT

3.2.3. Compuerta CCN o Toffoli

Compuerta Controlled-Controlled NOT, en inglés:

$$N_{CC} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

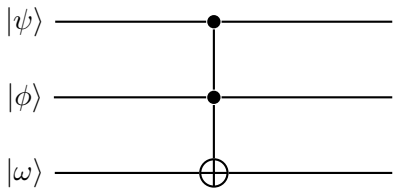

FIGURA 3.3: Compuerta CCNOT o Toffoli

3.2.4. Compuerta CSWAP o Fredkin

Inventada por Fredkin, la compuerta Controlled Exchange es similar a la anterior, solo que intercambia los penúltimos y antepenúltimos bits.

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

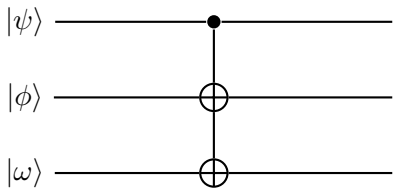

FIGURA 3.4: Compuerta CCNOT

3.2.5. Compuertas X, Y y Z o Matrices de Pauli

Las Matrices de Pauli para 1 qubit, tomando en cuenta la matriz X que presentamos en 2.2.1 como la compuerta NOT para 1 qubit, se representan de la siguiente forma:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad y \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

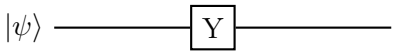

FIGURA 3.5: Compuerta Y

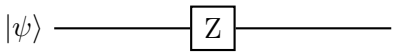

FIGURA 3.6: Compuerta Z

3.2.6. Compuerta Hadamard

Es una de las matrices de transformación unitaria más importantes y usadas:

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

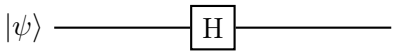

FIGURA 3.7: Compuerta Hadamard

3.2.7. Compuertas Phase Gate, T y $\mathbf{R_k}$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad y \qquad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

La compuerta R_k es más genérica con la característica de que $R_2 = S$ y $R_3 = T$

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

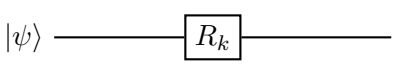

FIGURA 3.8: Compuerta R_k

Capítulo 4

Principales algoritmos cuánticos

4.1. Algoritmo de Deustch y Jozsa

El primer y más simple algoritmo cuántico que demuestra un incremento en el poder de cálculo con respecto a los algoritmos clásicos fue desarrollado por David Deutsch y Richard Jozsa en 1992. El algoritmo propone averiguar si una función es constante o balanceada. Esto, en un algoritmo clásico, para una función $f : (0, 1) \to (0, 1)$ requiere evaluar como mínimo dos valores. Deustch y Jozsa lograron implementar y alcanzar el mismo resultado haciendo una sola medición.

Veamos la siguiente función: f

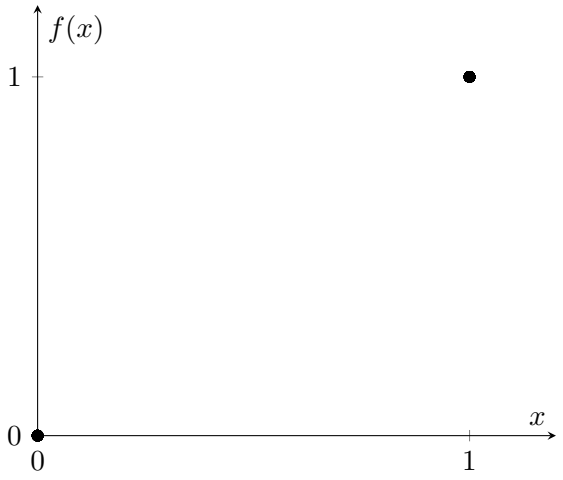

FIGURA 4.1: Ejemplo de función balanceada

A simple vista se puede observar que es una función balanceada, pero algorítmico deberíamos calcular $f(0) = 0$ y $f(1) = 1$ y concluir que al ser $f(0) \neq f(1)$, la función es balanceada.

Nuestra función queda definida como

$$f : (0,1) \to (0,1) = \begin{cases} Balanceada & si \quad f(0) \neq f(1) \\ \\ Constante & si \quad f(0) = f(1) \end{cases}$$

El algoritmo de Deustch y Jozsa es el siguiente:

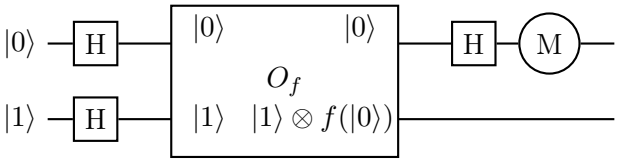

FIGURA 4.2: Circuito de Deustch y Jozsa

En primer lugar, tomamos dos qubits $|0\rangle\, y\, |1\rangle$

Al comienzo del circuito, el estado del sistema es:

$$|\omega_0\rangle = |0\rangle \otimes |1\rangle$$

Luego le aplicamos a cada qubit la compuerta Hadamard:

$$|\omega_1\rangle = (\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes (\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle)$$

y lo escribimos como

$$|\omega_1\rangle = \frac{1}{2}|0\rangle \otimes (|0\rangle - |1\rangle) + \frac{1}{2}|1\rangle \otimes (|0\rangle - |1\rangle)$$

Ahora es hora de aplicar la transformación Oráculo (O_f), la cual actúa sobre un solo qubit.

$$O_f\,|\phi\rangle\,|\psi\rangle \to |\phi\rangle\,|\psi \oplus f(\phi)\rangle$$

Como se puede comprobar fácilmente, la operación definida por O_f es reversible gracias a la operación $\oplus$. De hecho, una compuerta clásica equivalente también sería reversible.

Continuando con el desarrollo, aplicamos O_f a nuestro sistema:

$$|\omega_2\rangle = \frac{1}{2}|0\rangle \otimes (|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + \frac{1}{2}|1\rangle \otimes (|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)$$

Lo que es importante destacar aquí, es que la función f está involucrada en las amplitudes de los qubits, pero todavía de nada nos sirve porque para saber que tipo de función es deberíamos medir ambos qubits.

Como podemos observar, $|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle$ tomará siempre distintos valores, por lo cual, es equivalente a $(-1)^{f(0)}(|0\rangle - |1\rangle)$ El mismo procedimiento podemos aplicar al termino $|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle$

Entonces reescribimos:

$$|\omega_2\rangle = \frac{1}{2}(-1)^{f(0)}|0\rangle \otimes (|0\rangle - |1\rangle) + \frac{1}{2}(-1)^{f(1)}|1\rangle \otimes (|0\rangle - |1\rangle)$$

y lo podemos separar como:

$$|\omega_2\rangle = \frac{1}{\sqrt{2}}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Como podemos observar, hemos alcanzado la parte más tediosa del algoritmo: lograr que la amplitud donde actúa la función quede en un solo qubit. Este fenómeno se denomina "phase kick-back".

Del primer qubit deducimos lo siguiente:

$$\text{del } 1^\circ \text{ qubit} = \begin{cases} |u_c\rangle = \frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) \quad si \quad f(0) = f(1) \\ \\ |u_b\rangle = \frac{\beta}{\sqrt{2}}(|0\rangle - |1\rangle) \quad si \quad f(0) \neq f(1) \end{cases}$$

con α y $\beta \in \{-1, 1\}$

$|u_c\rangle \perp |u_b\rangle$

Finalmente le aplicamos la compuerta Hadamard como indica el circuito.

Si f es constante:

$$H|u_c\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\frac{\alpha}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = \alpha|0\rangle$$

Si f es balanceada:

$$H\left|u_b\right\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\frac{\beta}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \begin{bmatrix}0\\ \beta\end{bmatrix} = \beta\left|1\right\rangle$$

Luego, si medimos el primer qubit y observamos el estado $\alpha\left|0\right\rangle$ podemos asegurar que la función es constante, de lo contrario es balanceada.

4.2. Algoritmo de Deutsch–Jozsa generalizado

El caso anterior es un caso particular del algoritmo de Deustsch-Jozsa generalizado. En general, si tenemos una función $f : (0,1)^n \to (0,1)$, para saber si es constante o balanceada, en un algoritmo clásico necesitaríamos evaluar $2^{n-1}+1$ puntos del dominio. Mediante un algoritmo cuántico se realizará con una medición de la función f. Cabe aclarar que en 1985 David Deustsh y Richard Jozsa llegaron a la solución con dos mediciones de f. No fue hasta 1998 que Richard Cleve, Artur Ekert, Chiara Macchiavello, y Michele Mosca llegaron a la versión actual del algoritmo.
En complejidad computacional, veremos que el algoritmo pertenece a la clase EQP, esto quiere decir que resuelve el problema en tiempo polinomial con probabilidad igual a 1. Es el caso análogo a la clase P en computación clásica.

El circuito de compuertas cuánticas es el siguiente:

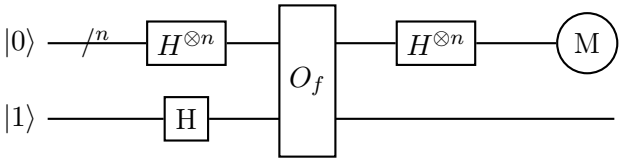

FIGURA 4.3: Circuito de Deutsch–Jozsa generalizado

Inicialmente nuestro sistema es:

$$\left|\omega_o\right\rangle = (\left|0_0\right\rangle \otimes \left|0_1\right\rangle \otimes ... \otimes \left|0_n\right\rangle) \otimes \left|1\right\rangle = \left|0\right\rangle^{\otimes n} \otimes \left|1\right\rangle$$

Primero aplicamos la compuerta Hadamard $H^{\otimes n}$ para el primer registro de n qubits y luego, H para el qubit $\left|1\right\rangle$

$$\left|\omega_1\right\rangle = H^{\otimes n}\left|0\right\rangle^{\otimes n} \otimes H\left|1\right\rangle$$

$$|\omega_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes (|0\rangle - |1\rangle)$$

A continuación aplicaremos la compuerta Oráculo con la función f.

$$|\omega_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x\in\{0,1\}^n} |x\rangle \otimes (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle)$$

Recordemos que $0 \oplus f(x) = f(x)$ y que $|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle$ puede valer $|0\rangle - |1\rangle$ o $|1\rangle - |0\rangle$, dependiendo si $f(x)$ es constante o balanceada. Reescribiendo nos queda:

$$|\omega_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

A partir de este momento, dejamos de lado el último qubit, ya que como indica el circuito, debemos aplicar la compuerta Hadamard al primer registro:

$$|\omega_2\rangle = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \left[\sum_{y\in\{0,1\}^n} (-1)^{x\cdot y} |y\rangle \right]$$

Ahora $|y\rangle$ es nuestro registro de n bits y depende de la alternancia de ± 1 según $x \cdot y$. La transformación Hadamard modifica el signo y se puede expresar como la suma de la multiplicación bit a bit:

$$x_0 \cdot y_0 \oplus x_1 \cdot y_2 \oplus ... \oplus x_{n-1} \cdot y_{n-1}$$

Reordenando los términos de la sumatoria:

$$|\omega_2\rangle = \frac{1}{2^n} \sum_{y\in\{0,1\}^n} \left[\sum_{x\in\{0,1\}^n} (-1)^{f(x)} (-1)^{x\cdot y} \right] |y\rangle$$

Finalmente, calculamos la probabilidad de medir: $|0\rangle^{\otimes n}$.

$$\left| \frac{1}{2^n} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \right|^2$$

Solo cuando $|y\rangle = |0\rangle^{\otimes n}$, el valor $(-1)^{x \cdot y} = 1$ y si además la función es constante, obtendremos $|y\rangle = |0\rangle^{\otimes n}$ con probabilidad 1. Si la función es balanceada, la probabilidad de obtener $|y\rangle = |0\rangle^{\otimes n}$ es 0 y obtendremos un valor distinto de $|y\rangle = |0\rangle^{\otimes n}$.

$$\text{La probabilidad de medir } |0\rangle^{\otimes n} = \begin{cases} 1 \quad si \quad f \text{ es constante} \\ \\ 0 \quad si \quad f \text{ es balanceada} \end{cases}$$

Como podemos observar, hemos hecho una sola medición.

4.3. Algoritmo de Simon

Daniel R. Simon en 1994 exhibió un algoritmo cuántico que resuelve el problema que presentaremos a continuación, en tiempo polinomial.

Dada una función $f : \{0,1\}^n \to \{0,1\}^n$ y que satisface la propiedad que $\forall x, y \in \{0,1\}^n, f(x) = f(y) \iff x \oplus y = s$ para algún $s \in \{0,1\}^n$

Desde la teoría de la complejidad computacional decimos que este problema se encuentra incluido en la clase BQP[1], si lo consideramos resoluble desde una computadora cuántica. Es exponencialmente mas rápido que cualquier máquina clásica determinística o probabilística.
El problema que expondremos, junto con el siguiente, son un caso especial de abelian hidden subgroup problem.

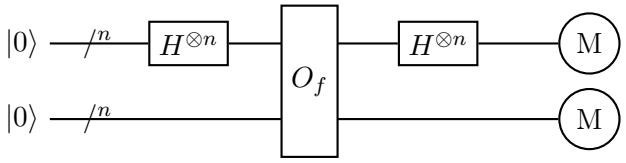

FIGURA 4.4: Circuito de Simon

Como observamos en el circuito, en primer lugar tomamos dos registros de n qubits cada uno inicializados en $|0\rangle$:

$$|\omega_o\rangle = (|0_0\rangle \otimes |0_1\rangle \otimes ... \otimes |0_n\rangle) \otimes (|0_0\rangle \otimes |0_1\rangle \otimes ... \otimes |0_n\rangle) = |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$$

Luego aplicamos la compuerta Hadamard $H^{\otimes n}$ al primer registro.

[1] $BPP \in BQP$

$$|\omega_1\rangle = H^{\otimes n} |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n}$$

$$|\omega_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |0\rangle$$

A continuación, ambos registros pasan por una función Oráculo [11]. Como ya hemos mencionado, la función solo afectará al segundo registro.

$$|\omega_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |(|0 \oplus f(x)\rangle)\rangle$$

$$|\omega_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |f(x)\rangle$$

En este paso medimos el segundo registro, el cual colapsará en $f(x_0)$ para algún $x_0 \in \mathbb{Z}_2^n$ Inmediatamente vemos que, para que se cumpla la condición de $\iff$, se debe cumplir la condición: $x \oplus y = s$, y para incluir a s en en algoritmo escribimos $y_0 = x_0 \oplus s$. Nuestro sistema quedará reducido a la siguiente expresión:

$$|\omega_3\rangle = \frac{1}{\sqrt{2}}(|x_0\rangle + |x_0 \oplus s\rangle)$$

La última compuerta que aplicaremos es Hadamard sobre el primer registro:

$$|\omega_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} ((-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus s) \cdot y}) |y\rangle$$

Sacamos factor común entre $((-1)^{x_0 \cdot y}$ y $((-1)^{x_0 \cdot y \oplus s \cdot y}$ y nos queda:

$$|\omega_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} (-1)^{x_0 \cdot y} (1 + (-1)^{y \cdot s}) |y\rangle$$

Ahora estamos en condiciones de obtener mucha información sobre s. Si se da el caso trivial en que $s = 0^n$, decimos que f es inyectiva y esto habrá ocurrido con la siguiente probabilidad:

[11] O_f es una trasformación unitaria con los efectos de la función definida f

$$\left| \frac{1}{\sqrt{2^n}} |x_0\rangle \otimes f(x_0) \right|^2 = \frac{1}{2^n}$$

en cualquier otro caso obtenemos:

$$\left| \frac{1}{\sqrt{2^{n+1}}} (-1)^{x_0 \cdot y} (1 + (-1)^{y \cdot s}) \right|^2 =$$

$$\text{La probabilidad p} = \begin{cases} \frac{1}{2^{n-1}} & si \quad s \cdot y = 0 \\ \\ 0 & si \quad s \cdot y = 1 \end{cases}$$

Entonces, para un $s \neq 0^n$, el resultado siempre es para un $s \cdot y = 0$. A partir de esto vamos a tratar de determinar s. Los siguientes pasos corresponden a un procesamiento clásico. Como demostraremos a continuación, una vez obtenido y, es suficiente información como para obtener s.

Para poder resolver la ecuación $y \cdot s = 0$ vamos a necesitar $n-1$ ecuaciones linealmente independientes. Decimos que necesitamos $n-1$ y no n ya que el caso trivial $s = 0^n$ satisface las ecuaciones.

$$\begin{aligned} & y_1 \cdot s = y_{11} \cdot s_1 + y_{12} \cdot s_2 + ... + y_{1n} \cdot s_n = 0 \\ & y_1 \cdot s = y_{11} \cdot s_1 + y_{12} \cdot s_2 + ... + y_{1n} \cdot s_n = 0 \\ & \vdots \\ & y_{n-1} \cdot s = y_{(n-1)1} \cdot s_1 + y_{(n-1)2} \cdot s_2 + ... + y_{(n-1)n} \cdot s_n = 0 \end{aligned}$$

La probabilidad de fallar al obtener la primer ecuación lineal es igual a obtener $P_{f1}(y_1 = 0)$[III] $= 2^{-n}$, por el complemento, la probabilidad de obtener exitosamente la primer ecuación lineal es $P_{e1}(y_1 \neq 0) = 1 - 2^{-n}$. Para la segunda ecuación vamos a necesitar que el algoritmo no nos arroje $y_2 = 0$ o $y_2 = y_1$, entonces la probabilidad de fracaso es $P_{f2} = 2^{-n+1}$ y la de éxito $P_{e2} = 1 - 2^{-n+1}$. La probabilidad de obtener $n-1$ ecuaciones lineales queda expresado de la siguiente manera:

$$P_e = \left(1 - \frac{1}{2^n}\right) \left(1 - \frac{1}{2^{n-1}}\right) ... \left(1 - \frac{1}{2^{n-(n-2)}}\right)$$

[III] Aquí llamaremos P_{f1} a la probabilidad de fracaso de obtener la primer ecuación y la de éxito P_{e1}, y así sucesivamente

Para acotar esta ecuación utilizaremos la desigualdad $(1-x_1)(1-x_2)...(1-x_n-1) \geq 1-x_1-x_2-...-x_n \forall\ 0 \leq x_i \leq 1$ con $ide1..n$

Escribimos:

$$P_e \geq 1 - \sum_{i=2}^{n} \frac{1}{2^i}$$

Si utilizamos la serie geométrica obtenemos una expresión

$$P_e \geq \frac{1}{2} + \frac{1}{2^n} \approx \frac{1}{2}$$

que nos dice que luego de correr el procedimiento $n-1$ veces, tendremos éxito en encontrar todas ecuaciones linealmente independientes con una probabilidad de un poco más de $\frac{1}{2}$, que aproximaremos a $\frac{1}{2}$ para n grande.

Si corremos el algoritmo $2k$ veces, la probabilidad de que falle es:

$$\left(1 - \frac{1}{2}\right)^{2k} < e^{-k}$$

4.4. Algoritmo de Grover

El algoritmo de Grover es un algoritmo de búsqueda no estructurada que se basa en la técnica amplificación de la amplitud. A continuación describiremos como se configura la búsqueda y desarrollaremos en detalle esta técnica de la que se nutre el algoritmo.

Supongamos que tenemos una función $f : \{1....N\} \rightarrow 0,1$ y existe un solo elemento $a : \{1....N\}$, tal que $f(a) = 1$, para todos los demás, con $x : \{1....N\}$, $f(x \neq a) = 0$

Para una solución de este tipo en una máquina clásica tenemos una complejidad $\mathcal{O}(N)$, mientras que en una computadora cuántica, $\mathcal{O}(\sqrt{N})$. Esto nos da una mejora cuadrática.

Desarrollaremos el algoritmo, descomponiéndolo en 3 partes, a la que a cada una, le dedicaremos una sección y haremos referencia para que el lector pueda profundizar en ellas.

4.4.1. El Algoritmo

Si buscamos en una lista de N elementos, sabemos que cuánticamente podemos representar todos los elementos en $n : \log_2 N$ qubits.

Definimos los n qubits de la siguiente manera:

$$|\psi_1\rangle = \sum_{i=1}^{N} |0\rangle \tag{4.1}$$

Luego los inicializaremos (ver sección 4.4.2) con la compuerta Hadamard para obtener amplitudes uniformes:

$$|\psi_2\rangle = H\,|\psi_1\rangle = \sum_{i=1}^{N} \alpha_i\,|x\rangle \tag{4.2}$$

con $\alpha_i = \frac{1}{\sqrt{N}}$

El siguiente paso es invertir la fase (ver sección 4.4.3), para lo cual utilizaremos la función oráculo f como describimos al comienzo. Necesitamos construir un operador unitario U_f tal que invierta la fase del elemento que estamos buscando:

$$|\psi_3\rangle = U_f\,|\psi_2\rangle = \sum_{i=1}^{N} \alpha_i(-1)^{f(x)}\,|x\rangle \tag{4.3}$$

Una vez que hemos invertido la fase, lo siguiente es ampliar dicha fase. Esto se hace con lo que Grover llamó Operador de difusión o también conocido como inversión sobre la media o simplemente amplificación de la amplitud.

A este operador le llamaremos U_s, el cual demostraremos en la sección (4.4.4) como efectivamente multiplica la amplitud del elemento a que buscamos. Definimos el operador $|s\rangle$ de la siguiente manera

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |x\rangle \tag{4.4}$$

$$U_s = 2\,|s\rangle\,\langle s| - I \tag{4.5}$$

Luego aplicamos el operador U_s a nuestros qubits:

$$|\psi_4\rangle = U_s\,|\psi_3\rangle \tag{4.6}$$

Hasta aquí hemos terminado la primera interacción. El resto de las iteración será aplicar el operador U_f seguido de U_s. En total serán $\frac{\pi\sqrt{N}}{4}$ iteraciones.

El algoritmo nos queda:

1. Entrada del algoritmo
   - Estado base $\sum_{i=1}^{N} |0\rangle$
   - Operador Oráculo U_f
2. Inicializar estado base $\sum_{i=1}^{N} \alpha_i\,|x\rangle$
3. De 1 a $\sqrt{N}$
   a) Aplicar operador U_f
   b) Aplicar operador U_s
4. Medir

4.4.2. Inicialización

Luego de aplicar la compuerta Hadamard, todas las amplitudes se distribuyen uniformemente. Es evidente que el promedio μ es igual a cualquier amplitud, incluso del elemento buscado a.

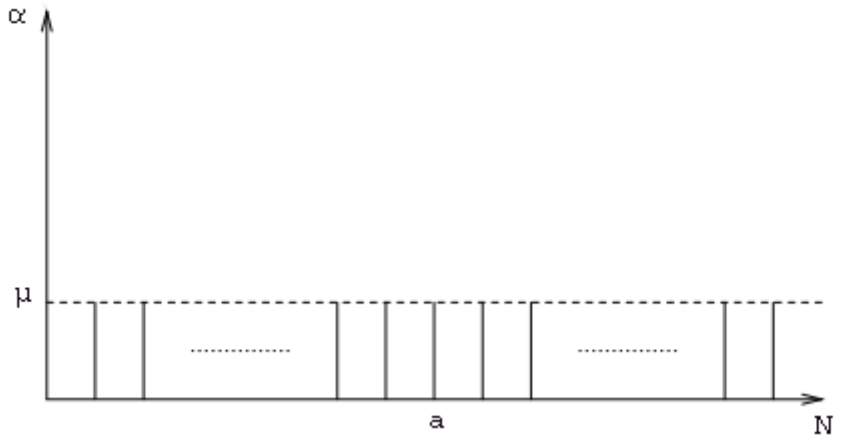

FIGURA 4.5: α es la amplitud, μ es la media de todas las amplitudes, a es el elemento que buscamos y N es el numero de elementos

4.4.3. Inversión de fase

Como mencionamos al principio del capítulo, solo para a se invertirá la fase ya que $(-1)^{f(a)=1} = -1$, para todos los demás la fase no se habrá alterado.

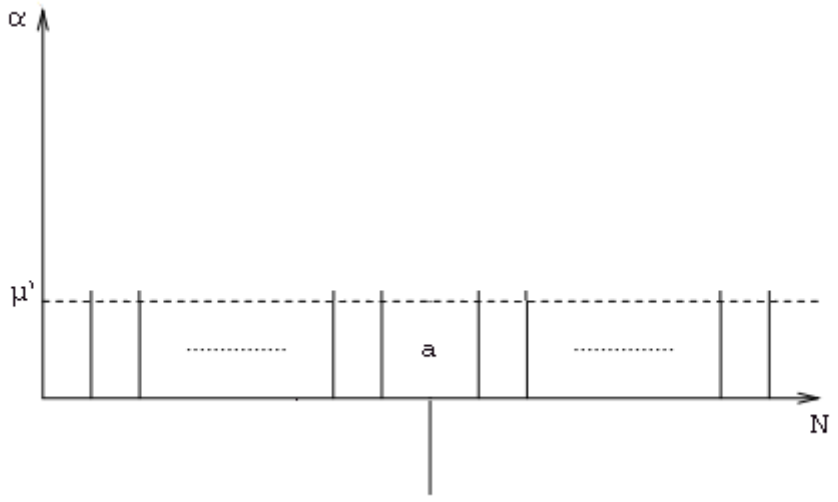

FIGURA 4.6: α es la amplitud, $\mu^{'}$ es la media de todas las amplitudes, a es el elemento que buscamos, con la amplitud invertida y N es el numero de elementos

4.4.4. Operador de difusión de Grover

Como podemos observar en la Figura 4.7, luego de aplicar el operador difusor, ha aumentado la amplitud del elemento buscado.

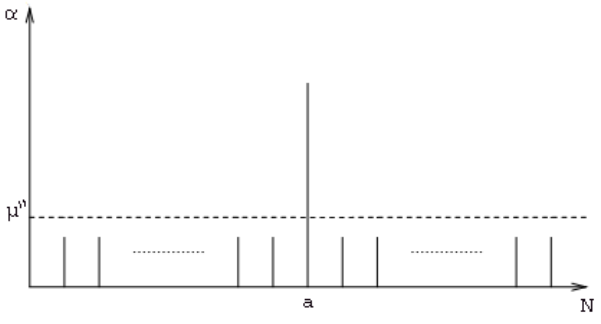

FIGURA 4.7: α es la amplitud, $\mu^{''}$ es la media de todas las amplitudes, a es el elemento que buscamos, cuya amplitud ha crecido, a su vez, las demás amplitudes se han reducido y N es el numero de elementos

Para demostrar por que U_s es el operador que incremente la amplitud de el elemento buscado, partimos de las siguientes matrices:

$$U_s = H_N \begin{pmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} H_N$$

Extraemos la matriz identidad I y la sumamos para mantener la igualdad

$$= H_N \left(\begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} + I \right) H_N$$

Aquí solo acomodo los términos, básicamente saco la identidad afuera como suma

$$= H_N \begin{pmatrix} -2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} H_N + I$$

Multiplico las Matrices Hadamard a ambos lados

$$= \begin{pmatrix} -2/N & -2/N & \cdots & -2/N \\ -2/N & -2/N & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N \end{pmatrix} + I$$

Finalmente obtenemos

$$= \begin{pmatrix} -2/N+1 & -2/N & \cdots & -2/N \\ -2/N & -2/N+1 & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N+1 \end{pmatrix}$$

Si nuestro estado de la primera iteración, luego de aplicar la inversión de la fase, nuestro elemento a buscar α_i tendrá signo opuesto con respecto a los demás elementos

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ -\alpha_i \\ \vdots \\ \alpha_N \end{pmatrix}$$

Apliquemos el operador U_s para cada elemento de nuestro estado:

$$= \begin{pmatrix} -2/N+1 & -2/N & \cdots & -2/N \\ -2/N & -2/N+1 & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N+1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ -\alpha_i \\ \vdots \\ \alpha_N \end{pmatrix}$$

Al ser la primera iteración, sabemos que todas las amplitudes serán $\frac{1}{\sqrt{N}}$, con la amplitud i con tendrá signo opuesto.

$$= \begin{pmatrix} -2/N+1 & -2/N & \cdots & -2/N \\ -2/N & -2/N+1 & \cdots & -2/N \\ \vdots & \vdots & \ddots & \vdots \\ -2/N & -2/N & \cdots & -2/N+1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \vdots \\ -\frac{1}{\sqrt{N}} \\ \vdots \\ \frac{1}{\sqrt{N}} \end{pmatrix}$$

El resultado nos queda

$$= \begin{pmatrix} -\left(\frac{1}{\sqrt{N}} - \frac{2}{N\sqrt{N}}\right)_1 \\ \vdots \\ -\left(\frac{1}{\sqrt{N}} + \frac{2N-4}{N\sqrt{N}}\right)_i \\ \vdots \\ -\left(\frac{1}{\sqrt{N}} - \frac{2}{N\sqrt{N}}\right)_N \end{pmatrix}$$

Vemos que para el término i se incrementa su amplitud en $\frac{2N-4}{N\sqrt{N}}$, mientas que en el resto de los términos, sus amplitudes se ven reducidas en $\frac{2}{N\sqrt{N}}$

Si del elemento i desprecio el factor 4, nos queda un incremento de amplitud de $\sim \frac{3}{\sqrt{N}}$. Lo que quiere decir, que en cada iteración tendremos aproximadamente un incremento de amplitud de $\frac{2}{\sqrt{N}}$

Luego de $\sim \sqrt{N}$ iteraciones, tendremos una amplitud cercana a 1, por lo cual, con alta probabilidad la medición será exitosa.

Existe otra forma de representar cada iteración además de las que vimos en la figura 4.6 y 4.7, y es a través de reflexiones.

4.4.5. Interpretación geométrica

Hay una representación geométrica muy intuitiva del algoritmo. Como podemos observar en la figura 4.8, las reflexiones del vector $|s\rangle$ corresponden a la primera iteración del algoritmo.

Definimos $|s\rangle$ como en la ecuación 4.4. Este estado es el resultado de aplicar la compuerta Hadamard sobre el estado inicial $|0\rangle^n$, siendo n la cantidad de qubits.

El vector a corresponde al estado que estamos buscando. Generamos un plano definido por ambos vectores, $|s\rangle$ y $|a\rangle$, con un ángulo β entre ellos.

Por otro lado, vamos a definir un estado $|s'\rangle$ que será perpendicular al vector $|a\rangle$ y nos permitirá escribir cualquier estado $|s\rangle$ como

$$\langle s| = \sin\theta\,|a\rangle + \cos\theta\,|s'\rangle$$

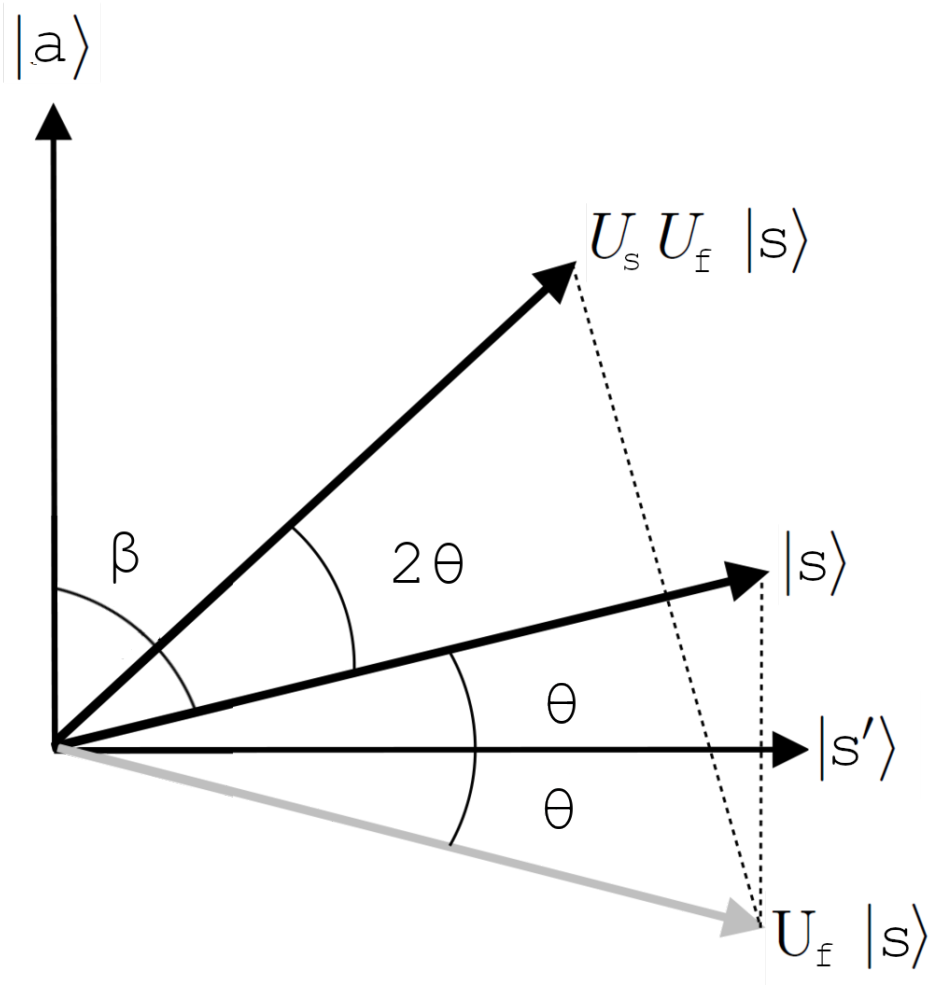

FIGURA 4.8: Representación gráfica de la primera iteración del algoritmo de Grover. $|a\rangle$ es el estado que buscamos, $|s\rangle$ el estado inicial y, U_f y U_s las dos transformaciones.

Por trigonometría sabemos que la fórmula para calcular el ángulo entre dos vectores, y en particular nuestro ángulo β es:

$$\cos\beta = \frac{\langle a|\,|s\rangle}{\|a\|\,\|s\|} \tag{4.7}$$

Notamos que $\langle a|\,|s\rangle$ es el producto interno entre vectores y, nuestro vector a, es el estado que estamos buscando, por lo cual, la posición del vector que representa dicho estado tendrá una amplitud ~ 1 y todas las demás ~ 0, ya que recordemos que la suma del cuadrado debe dar 1. Por ejemplo si tengo dos qubits, el espacio de Hilbert sera de $2^2 = 4$, y si mi elemento a buscar es el 10 [IV], me queda:

$$\langle a| = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

[IV] Para un vector de 4 dimensiones, la posición 0 equivale al numero binario 00, la siguiente posición al valor 01, la siguiente al 10 y por ultimo 11

El estado $|s\rangle$ es la superposición de estados uniformemente distribuida, cada uno con amplitud $\frac{1}{\sqrt{N}}$, o en nuestro ejemplo $\frac{1}{\sqrt{4}}$:

$$\langle s| = \left(\frac{1}{\sqrt{N}} \quad \frac{1}{\sqrt{N}} \quad \frac{1}{\sqrt{N}} \quad \frac{1}{\sqrt{N}}\right)$$

El resultado del producto interno entre estos dos vectores nos da $\frac{1}{\sqrt{N}}$ y $\frac{1}{\sqrt{4}}$ para nuestro ejemplo:

$$\langle a| \, |s\rangle = 0.\frac{1}{\sqrt{N}} + 0.\frac{1}{\sqrt{N}} + 1.\frac{1}{\sqrt{N}} + 0.\frac{1}{\sqrt{N}} = \frac{1}{\sqrt{N}}$$

La magnitud de ambos vectores es 1, ya que por definición sabemos que la suma al cuadrado de todas las amplitudes debe ser 1.

La ecuación 4.7 para calcular el ángulo β nos queda:

$$\cos\beta = \frac{1}{\sqrt{N}}$$

$$\beta = \arccos\frac{1}{\sqrt{N}}$$

Pero nos interesa más saber el ángulo θ, por lo cual vamos a usar la siguiente relación

$$\cos\beta = \sin\left(\frac{\pi}{2} - \beta\right)$$

Pero hemos definido a θ como $\frac{\pi}{2} - \beta$, por lo cual reemplazando nos queda:

$$\sin\theta = \frac{1}{\sqrt{N}}$$

$$\theta = \arcsin\frac{1}{\sqrt{N}}$$

Lo siguiente es aplicar el operador U_f sobre nuestro estado $|s\rangle$, lo que nos nos invertirá la fase, como vimos al principio de esta sección. La reflexión sera sobre el vector $|s'\rangle$, una magnitud θ. Pero para verlo mas claro, lo que realmente ocurre es que se invierte la dirección de $|s\rangle$ en en eje donde esta definido $|a\rangle$. Recordemos que en la primera iteración, la transformación U_f tan solo invierte el signo de la posición del elemento que estoy buscando.

Para finalizar la primera iteración, se aplica el operador U_s, el cual hará una reelección sobre $|s\rangle$, que a diferencia de U_f, invierte todos los estados y además, como vimos en la sección 4.4.4, duplica la amplitud inicial.

Por último, de esta representación se puede estimar la cantidad exacta de iteraciones que necesita el algoritmo. No nos detendremos a hacer este análisis pero diremos que el objetivo es acercar $|s\rangle$ a $|a\rangle$. En el plano que hemos definido, se traduce que el ángulo θ se aproxime lo máximo posible a $\frac{\pi}{2}$ [v]. El resultado equivaldrá a $\frac{\pi\sqrt{N}}{4}$ iteraciones.

[v] Si tuviésemos más de una entrada a buscar, la cantidad de iteraciones se vera reducida a $\frac{\pi}{4}\sqrt{\frac{N}{k}}$, siendo k esta cantidad de elementos que cumplen con la condición de búsqueda.

Capítulo 5

Experimentación Cuántica

En este capitulo analizaremos y ejecutaremos los algoritmos cuánticos presentados en el capítulo 4. Propondremos diferentes funciones Oráculo para completar cada algoritmo. Utilizaremos la plataforma IBM Quantum Experience para construir el circuito, junto con el lenguaje OPENQASM 2.0. También haremos parcialmente el desarrollo matricial de los operadores Oráculo para tener una mayor visibilidad de cómo esta transforma los estados.

5.1. Algoritmo de Deutsch y Jozsa: 2 qubit

En esta sección plantearemos algunas funciones para resolver a través del algoritmo de Deutsch y Jozsa.
Como vimos en el capitulo anterior, este algoritmo resuelve si una función es constante o balanceada. Para eso, debemos crear la transformación correspondiente a la función oráculo.

Plantearemos 4 funciones básicas, dos constantes y otras dos balanceadas, y para cada una de ellas construiremos su operador oráculo, para luego encontrar la solución mediante el circuito ejecutándolo en la plataforma de IBM.
Las funciones quedan definidas como $f : \{0, 1\} \to 0, 1$ con: $x : \{0, 1\}$ y para cada una en particular:

Funciones constantes:

$$f(x) = 0$$

$$f(x) = 1$$

Funciones balanceadas:

$$f(x) = x$$

$$f(x) = 1 - x$$

5.1.1. Función constante $f(|x\rangle) = 0$

La siguiente tabla nos muestra la función f:

$\|x\rangle$	$f(\|x\rangle)$
0	0
1	0

Como podemos observar, en la figura 5.2, nuestro $O(f)$ es equivalente a no aplicar ninguna transformación. Lo que buscamos es que el primer qubit pase sin modificarse, mientras que la salida para el segundo qubit sea: $|x\rangle \oplus f(|x\rangle)$

En bits, luego de aplicar $O(f)$ nos queda:

$\|x\rangle \|y\rangle \rightarrow O(f(\|x\rangle), \|y\rangle)$
$00 \rightarrow 00$
$01 \rightarrow 01$
$10 \rightarrow 10$
$11 \rightarrow 11$

A partir de $O(f)$, obtenemos los valores de $f(|x\rangle)$, ya que de la tabla anterior, tenemos el resultado de $|x\rangle \oplus f(|x\rangle)$, tan solo con mirar el resultado del segundo qubit luego de aplicar $O(f)$

$\|x\rangle$	$\|y\rangle$	$\|y\rangle \oplus f(\|x\rangle)$
0	0	$0 \oplus [f(0) = 0] = 0$
0	1	$1 \oplus [f(0) = 0] = 1$
1	0	$0 \oplus [f(1) = 0] = 0$
1	1	$1 \oplus [f(1) = 0] = 1$

Nuestro circuito quedaría simplemente:

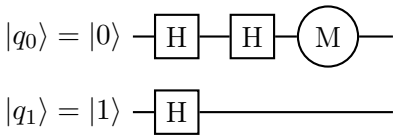

FIGURA 5.1: Circuito de Deutsch

A su vez mostramos como queda representado en la plataforma de IBM:

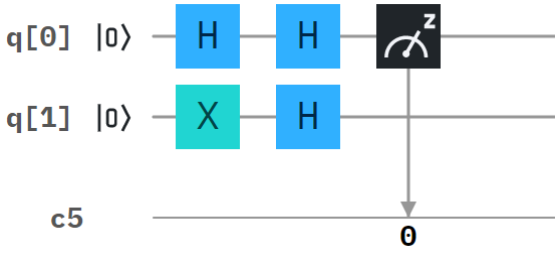

FIGURA 5.2: Circuito de Deutsch en la plataforma de IBM. Aplicamos una compuerta NOT al segundo qubit ya que este se debe inicializar en $|1\rangle$

En la figura 5.2 mostramos el resultado luego de ejecutar correr el circuito, como era de esperar, el resultado de medir el primer qubit es 0, lo cual implica que nuestra función es constante.

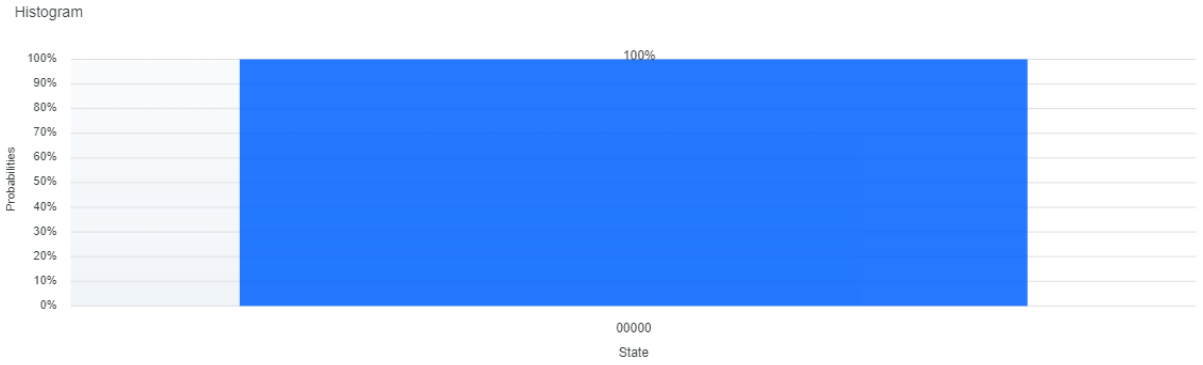

FIGURA 5.3: Resultado: 0

5.1.1.1. El algoritmo en OPENQASM 2.0

El algoritmo queda definido en el lenguaje OPENQASM 2.0 de la plataforma de simulación de IBM:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[2]; creg c[5];

h q[0]; x q[1]; h q[1];
h q[0];
measure q[0] ->c[0];
```

5.1.2. Función constante $f(|x\rangle) = 1$

La siguiente función es tambien constante y queda representada en la siguiente tabla:

$\|x\rangle$	$f(\|x\rangle)$
0	1
1	1

Nuestro Oráculo, $O(f)$, es equivalente a aplicar la transformación de una compuerta NOT sobre el segundo qubit.

Mostramos en la siguiente tabla que arrojaria nuestro operador Oraculo $O(f)$:

$\|x\rangle \|y\rangle \rightarrow O(f(\|x\rangle), \|y\rangle))$
$00 \rightarrow 01$
$01 \rightarrow 00$
$10 \rightarrow 11$
$11 \rightarrow 10$

A partir de $O(f)$, podemos deducir los valores de $|y\rangle \otimes f(|x\rangle)$, como hicimos anteriormente:

$\|x\rangle$	$\|y\rangle$	$\|y\rangle \otimes f(\|x\rangle)$
0	0	$0 \otimes [f(0) = 1] = 1$
0	1	$1 \otimes [f(0) = 1] = 0$
1	0	$0 \otimes [f(1) = 1] = 1$
1	1	$1 \otimes [f(1) = 1] = 0$

Nuestro circuito nos queda conformado así:

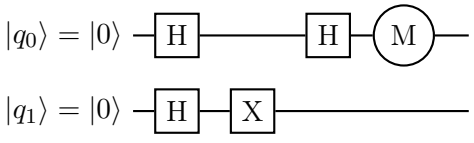

FIGURA 5.4: Circuito de Deutsch y Jozsa

Nuevamente, en la siguiente figura (5.17) mostramos el circuito construido en la plataforma de IBM

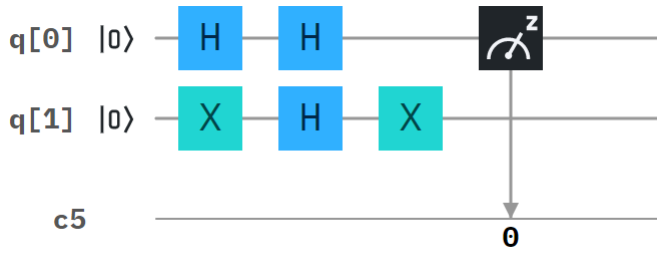

FIGURA 5.5: Circuito de Deutsch y Jozsa en la plataforma de IBM

Al igual que en el algoritmo anterior, obtenemos 0 al medir el primer qubit como podemos observar en la figura 5.18

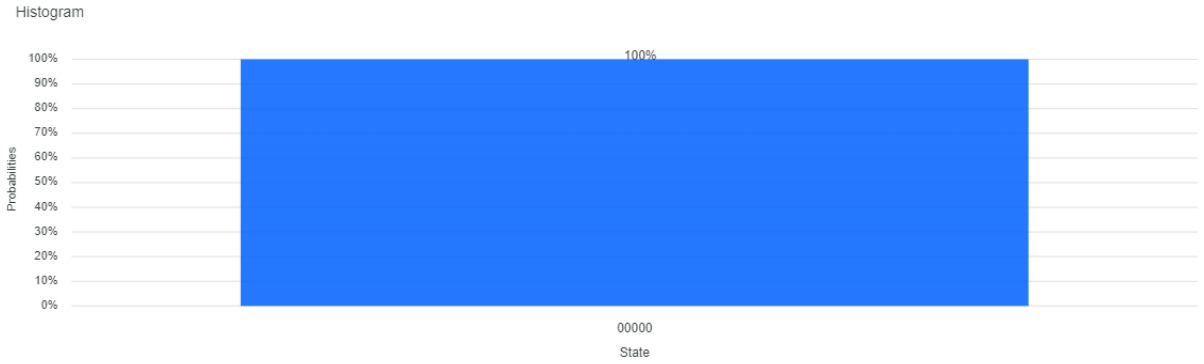

FIGURA 5.6: Resultado: 0

5.1.2.1. El algoritmo en OPENQASM 2.0

El algoritmo en el lenguaje OPENQASM 2.0:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[2]; creg c[5];

h q[0]; x q[1]; h q[1];
x q[1]; h q[0];
measure q[0] ->c[0];
```

5.1.3. Función balanceada $f(x) = x$

Aquí nos encontramos con una función balanceada, la cual esta representada en la siguiente tabla:

$\|x\rangle$	$f(\|x\rangle)$
0	0
1	1

La transformación Oráculo, $O(f)$, es equivalente a aplicar la transformación de una compuerta CNOT. Esto queda expresado en la tabla a continuación:

$\|x\rangle \|y\rangle \rightarrow O(f(\|x\rangle), \|y\rangle))$
$00 \rightarrow 00$
$01 \rightarrow 01$
$10 \rightarrow 11$
$11 \rightarrow 10$

Finalmente mostramos como deducimos $|y\rangle \otimes f(|x\rangle$

$\|x\rangle$	$\|y\rangle$	$\|y\rangle \otimes f(\|x\rangle)$
0	0	$0 \otimes [f(0) = 0] = 0$
0	1	$1 \otimes [f(0) = 0] = 1$
1	0	$0 \otimes [f(1) = 1] = 1$
1	1	$1 \otimes [f(1) = 1] = 0$

El circuito nos queda:

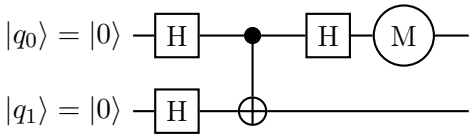

FIGURA 5.7: Circuito de Deutsch y Jozsa

Mismo circuito en la plataforma de IBM:

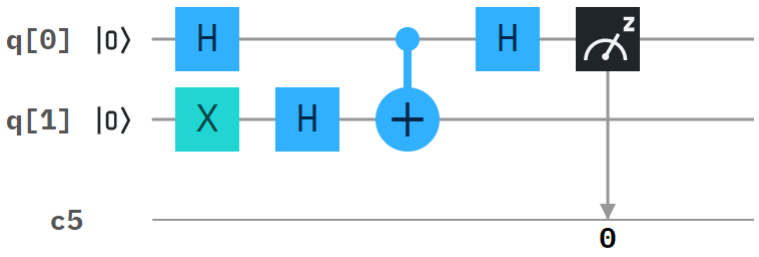

FIGURA 5.8: Circuito de Deutsch y Jozsa en la plataforma de IBM

Obtuvimos un resultado diferente al de los circuitos anteriores. El estado que se muestra en la figura 5.18 implica que efectivamente nuestra funcion f es balanceada

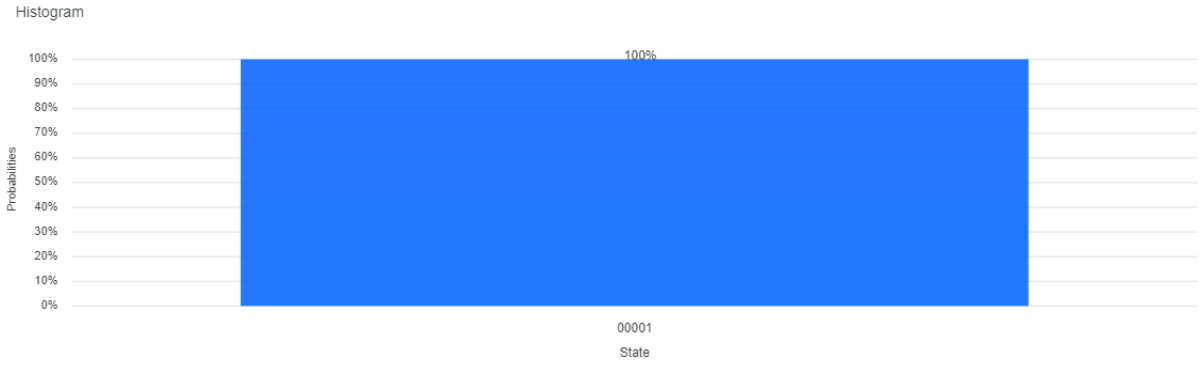

FIGURA 5.9: Resultado: 1

5.1.3.1. El algoritmo en OPENQASM 2.0

El algoritmo en el lenguaje OPENQASM 2.0:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[2]; creg c[5];
h q[0]; h q[1];
cx q[0],q[1];
h q[0];
measure q[0] ->c[0];
```

5.1.4. Función balanceada $f(|x\rangle) = 1 - |x\rangle$

Construiremos un último circuito de dos qubits para una función balanceada. La función arroja los siguientes valores mostrados en la siguiente tabla:

$\|x\rangle$	$f(\|y\rangle)$
0	1
1	0

El operador Oráculo lo representamos por una compuerta CNOT, lo cual genera la siguiente transformación sobre los qubits:

$\|x\rangle \|y\rangle \rightarrow O(f \|x\rangle), \|y\rangle)$
$00 \rightarrow 01$
$01 \rightarrow 00$
$10 \rightarrow 10$
$11 \rightarrow 11$

A partir de $O(f)$, deducimos $|y\rangle \otimes f(|x\rangle)$ como muestra la tabla a continuación:

$\|x\rangle$	$\|y\rangle$	$\|y\rangle \otimes f(\|x\rangle)$
0	0	$0 \otimes [f(0) = 1] = 1$
0	1	$1 \otimes [f(0) = 1] = 0$
1	0	$0 \otimes [f(1) = 0] = 0$
1	1	$1 \otimes [f(1) = 0] = 1$

El circuito es el siguiente:

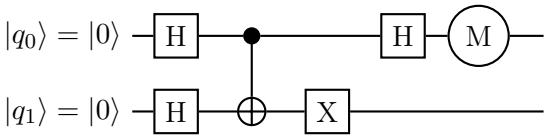

FIGURA 5.10: Circuito de Deutsch y Jozsa

En la figura 5.17 mostramos el mismo circuito en plataforma de IBM :

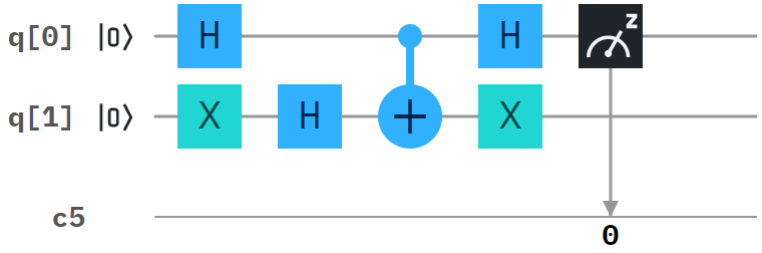

FIGURA 5.11: Circuito de Deutsch y Jozsa en la plataforma de IBM

Obtenemos el mismo resultado que la ejecución del algoritmo anterior. Obtener como resultado 1 implica que la función es balanceada

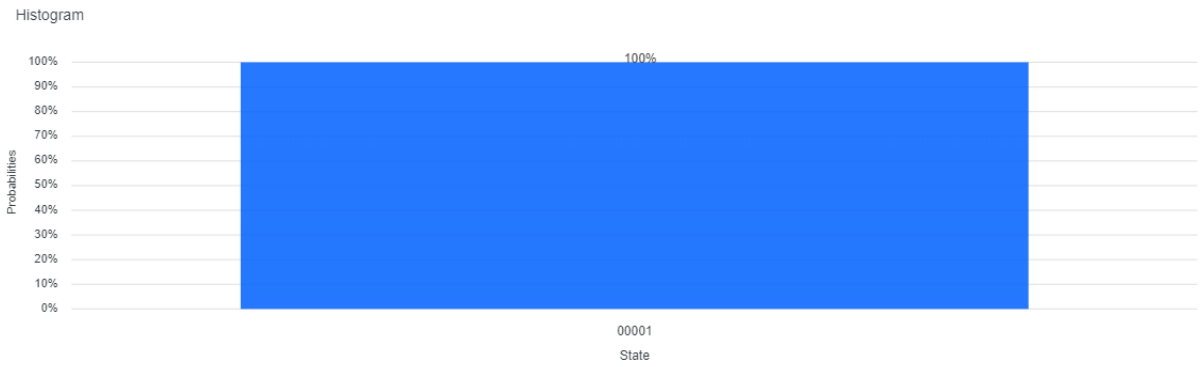

FIGURA 5.12: Resultado: 1

5.1.4.1. El algoritmo en OPENQASM 2.0

El algoritmo en el lenguaje OPENQASM 2.0:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[2]; creg c[5];
h q[0]; x q[1]; h q[1];
cx q[0],q[1];
h q[0]; x q[1];
measure q[0] ->c[0];
```

5.2. Algoritmo de Deutsch-Jozsa: n-qubits

Construiremos el algoritmo para una función constante y otra balanceada.
Utilizaremos n qubits + 1, por lo cual, en este caso, el algoritmo utilizará 3 qubits.
Sabemos que las matrices de transformación para 3 qubits son de 2^3, con 3 igual a la cantidad de qubits requeridos; y es por esta razón que nos limitaremos a construir estos algoritmos con un máximo de 3 qubit para que no se hagan engorrosas las operaciones.

Función constante:

$$f(x) = 0$$

Función balanceada:

$$f(x) = x$$

5.2.1. Función constante $f(|x\rangle\,|y\rangle) = 0$

La tabla de la función f queda definida de la siguiente manera:

$\|x\rangle$	$\|y\rangle$	$f(\|x\rangle \|y\rangle)$
0	0	0
0	1	0
1	0	0
1	1	0

Al igual que el primer ejercicio de este capítulo, nuestro Oráculo es equivalente a no aplicar ninguna transformación.

$\|x\rangle \|y\rangle \|z\rangle \rightarrow O(f(\|x\rangle \|y\rangle), \|z\rangle))$
$000 \rightarrow 000$
$001 \rightarrow 001$
$010 \rightarrow 010$
$011 \rightarrow 011$
$100 \rightarrow 100$
$101 \rightarrow 101$
$110 \rightarrow 110$
$111 \rightarrow 111$

A partir de $O(f)$, obtenemos los valores de $|z\rangle \otimes f(|x\rangle |y\rangle)$

[H]

$\|x\rangle \|y\rangle$	$\|z\rangle$	$\|z\rangle \otimes f(\|x\rangle \|y\rangle)$
00	0	$0 \otimes [f(00) = 0] = 0$
00	1	$1 \otimes [f(00) = 0] = 1$
01	0	$0 \otimes [f(01) = 0] = 0$
01	1	$1 \otimes [f(01) = 0] = 1$
10	0	$0 \otimes [f(10) = 0] = 0$
10	1	$1 \otimes [f(10) = 0] = 1$
11	0	$0 \otimes [f(11) = 0] = 0$
11	1	$1 \otimes [f(11) = 0] = 1$

El circuito nos queda:

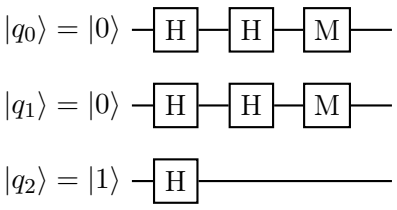

FIGURA 5.13: Circuito de Deutsch y Jozsa

En la plataforma de IBM nos quedó de la siguiente manera:

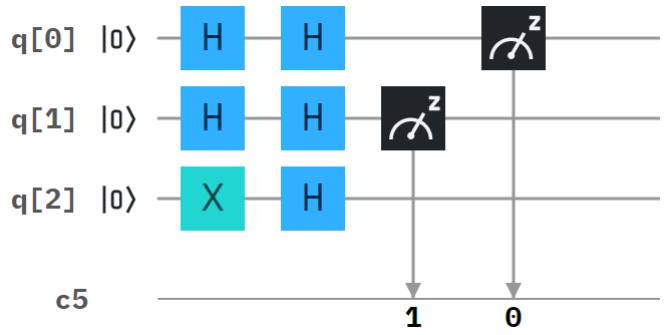

FIGURA 5.14: Circuito de Deutsch y Jozsa en la plataforma de IBM

Como era de esperar, el resultado de medir los primeros dos qubit es 0, lo cual implica que nuestra función es constante.

Mostramos el resultado obtenido luego de la ejecución en el simulador de IBM

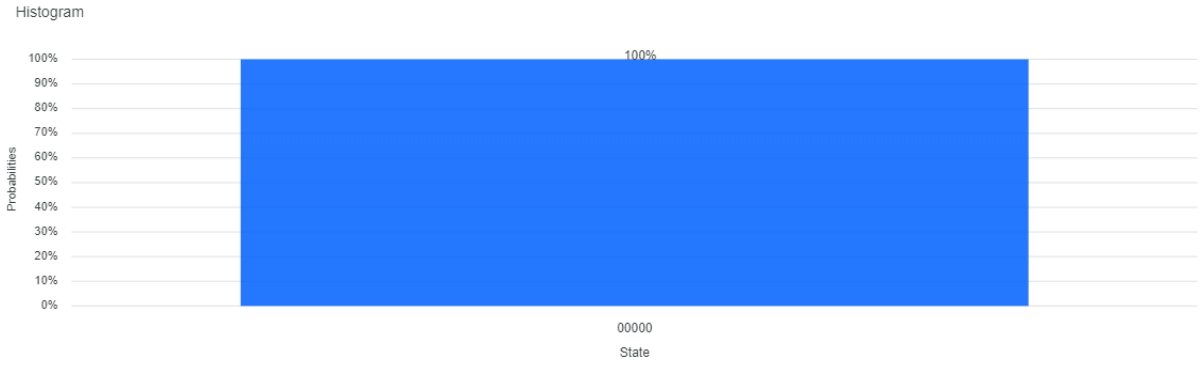

FIGURA 5.15: Resultado: 00

5.2.1.1. El algoritmo en OPENQASM 2.0

El algoritmo en el lenguaje OPENQASM 2.0:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[3]; creg c[5];
h q[0]; h q[1]; x q[2];
h q[0]; h q[1]; h q[2];
measure q[1] ->c[1];
measure q[0] ->c[0];
```

5.2.2. Función balanceada $f(|x\rangle, |y\rangle) = |x\rangle \oplus |y\rangle$

La tabla de verdad de la función f:

$\|x\rangle$	$\|y\rangle$	$f(\|x\rangle \|y\rangle)$
0	0	0
0	1	1
1	0	1
1	1	0

Nuestro Oráculo $O(f)$ lo construimos con dos compuertas Controllerd NOT, CX o CNOT, aplicadas, una sobre el primer y tercer qubit y la otra sobre el segundo y tercer qubit (ver figura 5.17).

$O(f)$ nos queda:

$\|x\rangle \|y\rangle \|z\rangle \rightarrow O(f(\|x\rangle \|y\rangle), \|z\rangle))$
$000 \rightarrow 000$
$001 \rightarrow 001$
$010 \rightarrow 011$
$011 \rightarrow 010$
$100 \rightarrow 101$
$101 \rightarrow 100$
$110 \rightarrow 110$
$111 \rightarrow 111$

A partir de $O(f)$, obtenemos los valores de $|z\rangle \otimes f(|x\rangle |y\rangle)$

$\|x\rangle \|y\rangle$	$\|z\rangle$	$\|z\rangle \otimes f(\|x\rangle \|y\rangle)$
00	0	$0 \otimes [f(00) = 0] = 0$
00	1	$1 \otimes [f(00) = 1] = 1$
01	0	$0 \otimes [f(01) = 1] = 1$
01	1	$1 \otimes [f(01) = 0] = 0$
10	0	$0 \otimes [f(10) = 1] = 1$
10	1	$1 \otimes [f(10) = 0] = 0$
11	0	$0 \otimes [f(11) = 0] = 0$
11	1	$1 \otimes [f(11) = 1] = 1$

El circuito nos conformado de la siguiente manera:

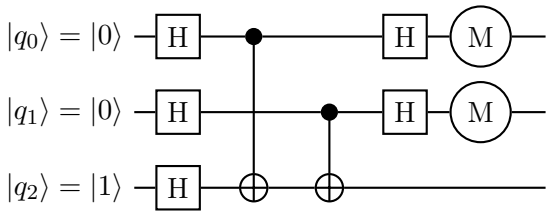

FIGURA 5.16: Circuito de Deutsch y Jozsa

En la figura 5.17 vemos el mismo circuito en la plataforma de IBM

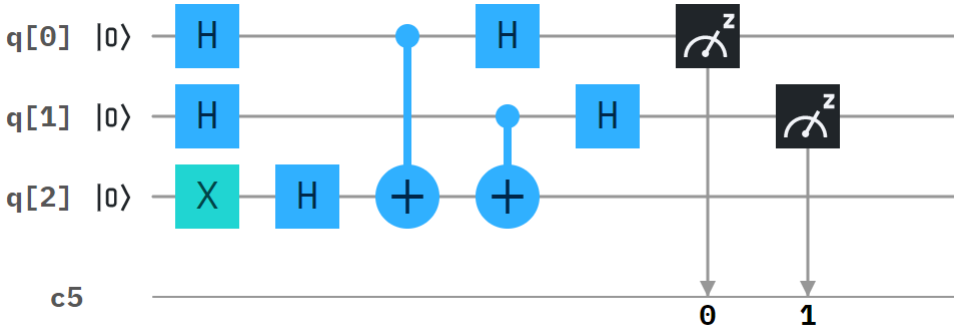

FIGURA 5.17: Circuito de Deutsch y Jozsa en la plataforma de IBM

En la figura siguiente (5.18) vemos que el resultado equivale a que nuestra función es balanceada

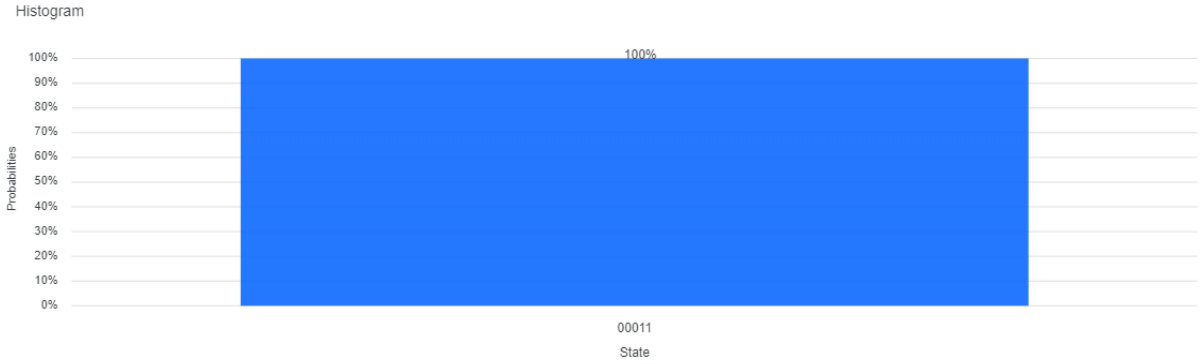

FIGURA 5.18: Resultado: 11

5.2.2.1. El algoritmo en OPENQASM 2.0

El algoritmo en el lenguaje OPENQASM 2.0:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[3]; creg c[5];
h q[0]; h q[1]; x q[2];
h q[2];
cx q[0],q[2];
cx q[1],q[2];
h q[0]; h q[1];
measure q[0] ->c[0];
measure q[1] ->c[1];
```

5.3. Algoritmo de Simon

Como vimos en el capítulo anterior, el algoritmo de Simon resuelve el siguiente problema: Dada una función $f : \{0,1\}^n \to \{0,1\}^n$ y que satisface la propiedad que $\forall x, y \in \{0,1\}^n, f(x) = f(y) \iff x \oplus y = s$ para algún $s \in \{0,1\}^n$ Nuestra función f será la siguiente:

$\|0\rangle$	$\|0\rangle$	$f(\|0\rangle \|0\rangle)$
0	0	00
0	1	11
1	0	11
1	1	00

Llamaremos Reg_1 a la primera mitad de los qubits del circuito y Reg_2 a la restante. La tabla de la transformación $O(f)$, queda definida en la siguiente tabla:

$Reg_1 Reg_2 \rightarrow O(f))$
$0000 \rightarrow 0000$
$0000 \rightarrow 0000$
$0000 \rightarrow 0000$
$0000 \rightarrow 0000$
$0100 \rightarrow 0111$
$0100 \rightarrow 0111$
$0100 \rightarrow 0111$
$0100 \rightarrow 0111$
$1000 \rightarrow 1011$
$1000 \rightarrow 1011$
$1000 \rightarrow 1011$
$1000 \rightarrow 1011$
$1100 \rightarrow 1100$
$1100 \rightarrow 1100$
$1100 \rightarrow 1100$
$1100 \rightarrow 1100$

A partir de $O(f)$, obtenemos los valores de $f(|0\rangle\,|0\rangle)$. A diferencia del algoritmo de Deutsch-Jozsa, la salida del segundo registro es la función f.

El circuito nos queda:

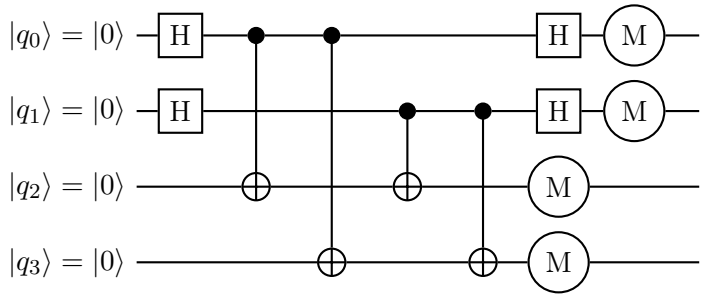

FIGURA 5.19: Circuito de Simon

Mostramos el circuito en la plataforma de IBM.

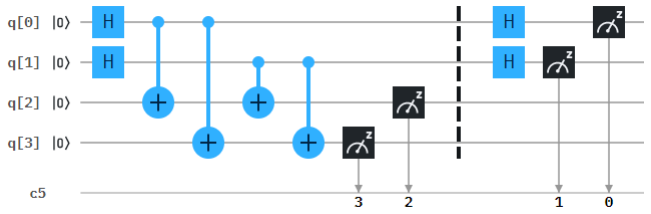

FIGURA 5.20: Circuito de Simon en la plataforma de IBM

Antes de medir el segundo registro, tenemos 50 % de probabilidad de obtener $|00\rangle$ y 50 % de probabilidad de $|11\rangle$. Eso se ve claramente en la tabla anterior.

Caso 1 Supongamos que de la medición de registro 2 resulta $|00\rangle$

El primer registro nos queda:

$$|\omega_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Luego, al aplicar Hadamard sobre el primer registro:

$$|\omega_5\rangle = \frac{1}{\sqrt{2}}[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) + (|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle)]$$

$$|\omega_5\rangle = \frac{1}{2\sqrt{2}}(|00\rangle + |01\rangle + |10\rangle + |11\rangle + |00\rangle - |01\rangle) - |10\rangle - |11\rangle)$$

Simplificando

$$|\omega_5\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Finalmente, tenemos 50 % de obtener $|00\rangle$ y 50 % de obtener $|11\rangle$ sobre el primer registro :

En caso de obtener $|00\rangle$ la ecuación:

$s.00 = 0$

En caso de obtener $|11\rangle$, nos queda

$s.11 = 0$

El sistema de ecuaciones nos queda:

$$s_1 \cdot 0 + s_2 \cdot 0 = 0 \quad \text{mód } 2$$

$$s_1 \cdot 1 + s_2 \cdot 1 = 0 \quad \text{mód } 2$$

El resultado que satisface estas ecuaciones es s = 11

En la siguiente tabla vemos que se cumple las condiciones de la función f:

$\|Reg_1\rangle$	$f(\|Reg_1\rangle$	$s = \|Reg_1\rangle \oplus \|Reg_2\rangle \, donde f(\|Reg_1\rangle) = f(\|Reg_2\rangle)$
00	00	$00 \oplus 11 = 11$
01	11	$01 \oplus 10 = 11$
10	11	$10 \oplus 01 = 11$
11	00	$11 \oplus 00 = 11$

Caso 2

Medición de registro 2 = $|11\rangle$

El primer registro nos queda:

$$|\omega_1\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Luego, al aplicar Hadamard sobre el primer registro:

$$|\omega_5\rangle = \frac{1}{\sqrt{2}}[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle) + (|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)]$$

$$|\omega_5\rangle = \frac{1}{2\sqrt{2}}(|00\rangle - |01\rangle + |10\rangle - |11\rangle + |00\rangle + |01\rangle) - |10\rangle - |11\rangle)$$

Simplificando

$$|\omega_5\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Finalmente, al igual que el caso anterior, tenemos 50 % de obtener $|00\rangle$ y 50 % de obtener $|11\rangle$ sobre el primer registro, por lo cual el resultado será el mismo que el caso 1

Finalmente mostramos el resultado que obtuvimos en la plataforma de IBM luego de ejecutar el algoritmo de la figura 5.20.

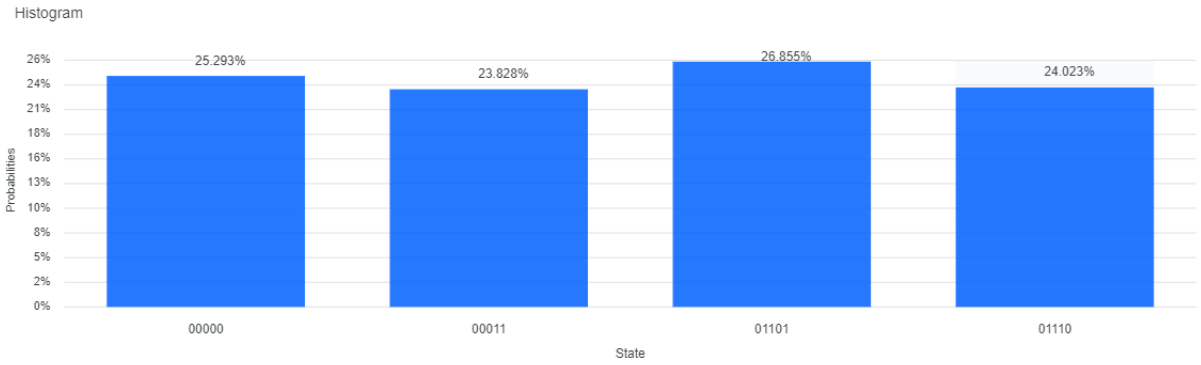

FIGURA 5.21: Resultado del algoritmo de la figura 5.20

5.3.0.2. El algoritmo en OPENQASM 2.0

El algoritmo queda definido en el lenguaje OPENQASM 2.0 de la plataforma de simulación de IBM:

```
OPENQASM 2.0;

include "qelib1.inc";
qreg q[4]; creg c[5];
h q[0]; h q[1];
cx q[0],q[2]; cx q[0],q[3];
cx q[1],q[2]; cx q[1],q[3];
measure q[3] ->c[3];
measure q[2] ->c[2];
barrier q[0],q[1],q[2],q[3];
h q[0]; h q[1];
measure q[1] ->c[1];
measure q[0] ->c[0];
```

5.4. Algoritmo de Grover: 3 qubit

Hemos preparado un algoritmo de 3 qubits cuya función Oraculo la definimos en la siguiente tabla:

$\|x\rangle \|y\rangle \|z\rangle \rightarrow O(f))$
$000 \rightarrow 0$
$001 \rightarrow 0$
$010 \rightarrow 0$
$011 \rightarrow 0$
$100 \rightarrow 0$
$101 \rightarrow 0$
$110 \rightarrow 0$
$111 \rightarrow 1$

Tal como esta representado en la tabla, el elemento buscado es 111 y es único, por lo cual la cantidad de iteraciones del algoritmo será $\frac{4\sqrt{8}}{4} = 2{,}221...$, según vimos en el capitulo

4., $N = 2^n$, donde n es la cantidad de qubits, en este caso 3.
Entonces la cantidad de iteraciones es igual a 2

Definimos la función Oráculo como la que queda encerrada entre las primeras dos barreras según la figura 5.22

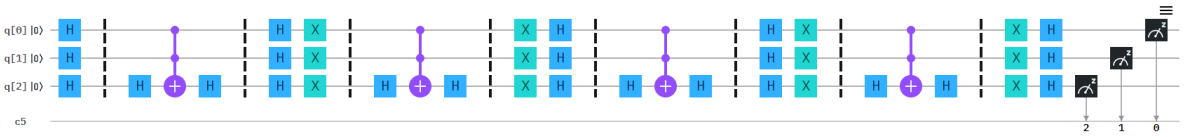

FIGURA 5.22: Algoritmo de Grover cuyo elemento a buscar es 111

El resultado del simulador es la figura 5.23

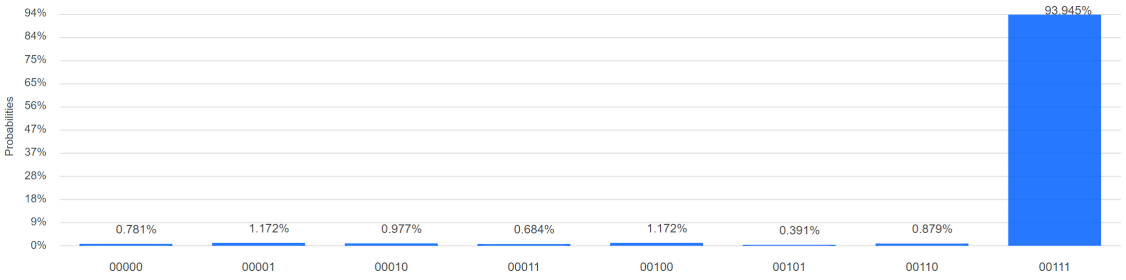

FIGURA 5.23: Resultado de la ejecución del algoritmo de la figura 5.22

5.4.0.3. Desarrollo matricial

Comenzaremos por expresar matricialmente la función Oráculo.

Se aplica la compuerta Hadamard sobre q[3], luego sobre todos los qubits la compuerta Toffoli y nuevamente la compuerta Hadamard sobre q[3]

Las operaciones se realizan de derecha a izquierda, aunque en este caso, el operador Oráculo es simétrico.

Las dos compuertas Hadamards tenemos que llevarlas a una matriz de $8x8$. Para lograr esto lo que haremos es utilizar el producto tensorial de las matrices $2x2$ de $I \oplus I \oplus H$.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Desarrollando las primeras dos matrices identidad nos queda:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

A este resultado llamémosle R_1

$$R_1 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Luego expresamos la matriz Toffoli o CCX:

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

Ahora si, calculemos el operador Oráculo como $R_1.CCX.R_1$:

$$O_f = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Multiplicamos las dos primeras matrices

$$O_f = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

El resultado es la siguiente matriz donde se aprecia claramente que se altera el signo de la ultima fila de la ultima columna. Luego de aplicarle este operador a nuestro estado, claramente te modificará el signo del ultimo elemento, el que representa el 111

$$O_f = \frac{1}{4} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Como podemos observar, a partir de 3 qubits las operaciones se vuelven engorrosas, al menos para desarrollarlas en este trabajo. Nos detendremos aquí y no construiremos

matricialmente el operador U_s ya que nuestra intención fue mostrar como se desarrollan las operaciones entre compuertas.

5.4.0.4. El algoritmo en OPENQASM 2.0

El algoritmo queda definido en el lenguaje OPENQASM 2.0 de la plataforma de simulación de IBM

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3]; creg c[5];
h q[0]; h q[1]; h q[2];
barrier q[0],q[1],q[2];
h q[2];
ccx q[0],q[1],q[2];
h q[2];
barrier q[0],q[1],q[2];
h q[0]; h q[1]; h q[2];
x q[0]; x q[1]; x q[2];
barrier q[0],q[1],q[2];
h q[2];
ccx q[0],q[1],q[2];
h q[2];
barrier q[0],q[1],q[2];
x q[0]; x q[1]; x q[2];
h q[0]; h q[1]; h q[2];
barrier q[0],q[1],q[2];
h q[2];
ccx q[0],q[1],q[2];
h q[2];
barrier q[0],q[1],q[2];
h q[0]; h q[1]; h q[2];
x q[0]; x q[1]; x q[2];
barrier q[0],q[1],q[2];
h q[2];
ccx q[0],q[1],q[2];
h q[2];
barrier q[0],q[1],q[2];
x q[0]; x q[1]; x q[2];
h q[0]; h q[1]; h q[2];
measure q[2] ->c[2];
measure q[1] ->c[1];
measure q[0] ->c[0];
```

5.5. Algoritmo de Grover: 3 qubit, con dos objetivos

En este caso, implementaremos una funcion oraculo tal que $f(x) = 1$ para dos valores de x. Veamos la siguiente tabla:

$\|x\rangle \|x\rangle \|x\rangle \rightarrow O(f))$
$000 \rightarrow 0$
$001 \rightarrow 0$
$010 \rightarrow 0$
$011 \rightarrow 1$
$100 \rightarrow 0$
$101 \rightarrow 1$
$110 \rightarrow 0$
$111 \rightarrow 0$

Nuestra función Oráculo esta compuesta por dos compuertas Controlled Z, como vemos entre las dos primeras barreras de la figura 5.24.
En este caso, ha diferencia del ejercicio anterior, hay dos estados que satisfacen la búsqueda, por lo cual, la cantidad de interacciones vendrá representada por: $\frac{\pi}{4}\sqrt{\frac{N}{k}}$, siendo k esta cantidad de elementos. Entonces la cantidad de iteraciones, con N=8 y k=2 :

$$\frac{\pi}{4}\sqrt{\frac{8}{2}} = 1{,}5707... \sim 1^{[1]}$$

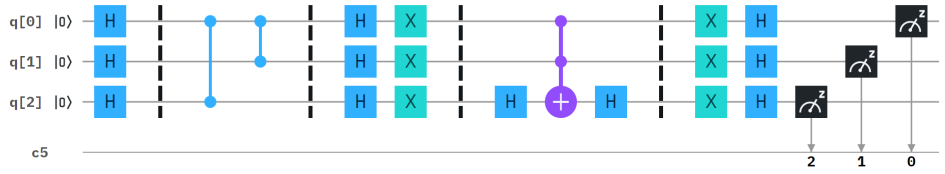

FIGURA 5.24: Algoritmo de Grover cuyos elementos a buscar son 110 y 101

El resultado podemos observarlo en la figura 5.25.

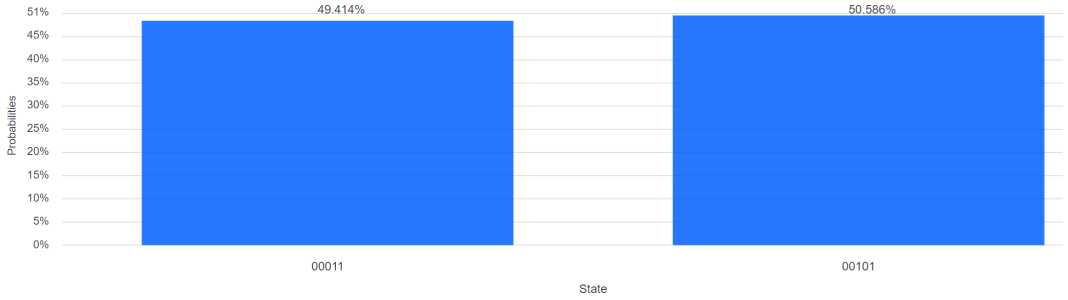

FIGURA 5.25: Resultado de la ejecución del algoritmo de la figura 5.24

5.5.0.5. Desarrollo matricial

Como observamos en la figura 5.24, la primer compuerta del operador Oráculo es Controlled Z aplicado al primer qubit y al último, sin involucrar el segundo, por lo cual ya no podemos utilizar el producto tensorial como lo veníamos haciendo. Para ello utilizaremos la siguiente estructura:

$$CU_n = \begin{bmatrix} I_{\frac{M}{2}} & O_{\frac{M}{2}} \\ O_{\frac{M}{2}} & I_{\frac{M}{4}} \otimes U \end{bmatrix}$$

[1]En una computadora clásica, encontraríamos estos dos elementos en 8 pasos, mientras que cuánticamente en 1 iteración hayamos los resultados

Donde $M = 2^{n+2}$ y n es la cantidad de qubits entre el qubit de control y el qubit objetivo, O es la matriz nula e I la matriz identidad. En nuestro caso $M = 2^{1+2} = 8$, con $n = 1$ya que la cantidad de qubits entre q[0] y q[2] es 1.

Resolvemos el producto tensorial de $I_{\frac{M}{4}} \otimes U$, donde $U = Pauli - Z$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

Ahora podemos escribir CU_1

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

La siguiente compuerta es también una compuerta Controlled Z pero aplicado sobre q[0] y q[1], por lo cual, luego bastará con aplicar el producto tensorial con la matriz identidad I de $2x2$ y de esta manera obtenemos la compuerta en $8x8$ dimensiones.

$$CZ \otimes I_{2x2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

Finalmente multiplicamos ambas compuertas (de derecha a izquierda) y obtenemos el operador Oraculo:

$$O_f = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Como podemos observar en la matriz final, el operador intercambiara los signos de los estados 101 y 110

5.5.1. El algoritmo en OPENQASM 2.0

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[3]; creg c[5];
h q[0]; h q[1]; h q[2];
barrier q[0],q[1],q[2];
cz q[2],q[0]; cz q[1],q[0];
barrier q[0],q[1],q[2];
h q[0]; h q[1]; h q[2];
x q[0]; x q[1]; x q[2];
barrier q[0],q[1],q[2];
h q[2];
ccx q[0],q[1],q[2];
h q[2];
barrier q[0],q[1],q[2];
x q[0]; x q[1]; x q[2];
h q[0]; h q[1]; h q[2];
measure q[2] ->c[2];
measure q[1] ->c[1];
measure q[0] ->c[0];
```

Conclusiones

La física cuántica surge a comienzos del XX con Max Planck, 80 años más tarde, se da comienzo a la teoría de la computación cuántica y no es hasta 2011 que D-Wave introduce D-Wave One, la primera computadora comercial. Con esta breve linea del tiempo quiero enfatizar lo exponencialmente rápido que avanzó la tecnología en los últimos 120 años. A este ritmo, podemos especular que en un mediano plazo el mundo cambiará rotundamente tal como lo conocemos. Como hemos mencionado al principio del capitulo, todas las áreas serán influenciadas por la computación cuántica.

En el capítulo 2, esbozamos una breve introducción a la física cuántica y las distintas direcciones que cada compañía ha tomado. Es importante resaltar que mi intención no solo fue mostrar las distintas tecnologías, sino también que estas llevan una complejidad altísima y solo pueden ser llevadas adelante con el trabajo mancomunado de toda al comunidad científica e ingenieril.

La teoría de la computación es un camino mas llevadero, partiendo del modelo de Turing cuántico, personas como Paul Benioff, Peter Shor, Scott Aaronson, Lov Grover, Daniel Simon, entre muchos otros han legado un avance muy grande a la computación cuántica. En el capitulo 3 y 4 vimos con cierto detalle algunos conceptos y algoritmos en los que demostramos teóricamente el poder que esta máquina conlleva.

Por último hemos ejecutamos algunos algoritmos y si bien llevó trabajo encontrar y construir las funciones Oráculo, a medida que uno rompe con la intuición clásica y empieza a imaginarse la realidad ”cuánticamente” , se allana el camino.

En general, concluimos que este trabajo es una invitación para aquellos que están buscando iniciarse en la computación cuántica, brindándoles, además de una breve introducción teórica, un paso al plano práctico. Creo que es fundamental empezar a preparar y educar a personas en esta área porque es el futuro.

Bibliografía

[1] David Morin. *Waves (Draft), Capítulo 10*. URL https://scholar.harvard.edu/david-morin/waves.

[2] Barton Zwiebach. Multiparticle states and tensor products. *Parte del curso Física Cuántica II del Instituto Tecnológico de Massachusetts*, Diciembre 2013. URL https://ocw.mit.edu/courses/physics/8-05-quantum-physics-ii-fall-2013/lecture-notes/MIT8_05F13_Chap_08.pdf.

[3] Alexandre Blais. Superconducting qubits. *Université de Sherbrooke, Québec, Canada*, 2015. URL http://www.quantum-lab.org/qip2015/slides/QIP2015-Alexandre%20Blais.pdf.

[4] Microsoft quantum team. Developing a topological qubit. *Microsoft blog*, Septiembre 2018. URL https://cloudblogs.microsoft.com/quantum/2018/09/06/developing-a-topological-qubit/.

[5] V.T. Lahtinen1. J.K. Pachos. A short introduction to topological quantum computation. *Freie Universitat Berlin, Arnimallee 14, 14195 Berlin, Germany*, Septiembre 2017. URL https://arxiv.org/abs/1705.04103.

[6] Bernard Field. Tapio Simula. Introduction to topological quantum computation with non-abelian anyons. *School of Physics and Astronomy, Monash University, Victoria 3800, Australia*, Abril 2018. URL https://arxiv.org/abs/1802.06176.

[7] Alexander Li. Topological quantum computing. *Medium blog*, Mayo 2019. URL https://medium.com/swlh/topological-quantum-computing-5b7bdc93d93f.

[8] D-Wave System team. Getting started with the system. *D-Wave System Documentation, Quantum Anneling*. URL https://docs.dwavesys.com/docs/latest/doc_getting_started.html.

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

[10] Umesh Vazirani. Quadratic speedup for unstructured search - grover's algorithm. *UC Berkeley, course CS 294-2, Lecture 11*, 2007. URL https://people.eecs.berkeley.edu/~vazirani/quantum.html.

[11] Craig Gidney. Grover's quantum search algorithm. *Twisted Oak Studios Blog*, Marzo 2013. URL http://twistedoakstudios.com/blog/Post2644_grovers-quantum-search-algorithm.

[12] Abraham Asfaw. Luciano Bello. Yael Ben-Haim. Sergey Bravyi. Lauren Capelluto. Almudena Carrera Vazquez. Jack Ceroni. Francis Harkins. Jay Gambetta. Shelly Garion. Leron Gil. Salvador De La Puente Gonzalez. David McKay. Zlatko Minev. Paul Nation. Anna Phan. Arthur Rattew. Joachim Schaefer. Javad Shabani. John Smolin. Kristan Temme. Madeleine Tod. James Wootton. Learn quantum computation using qiskit. *Textbook in collaboration with IBM Research as a university quantum algorithms/computation course supplement based on Qiskit.*, 2020. URL https://qiskit.org/textbook/preface.html.

[13] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Bell Labs, Murray Hill NJ 07974*, Mayo 1996. URL https://arxiv.org/abs/quant-ph/9605043.

[14] the free encyclopedia Wikipedia. Grover's algorithm. Marzo 2020. URL https://en.wikipedia.org/wiki/Grover%27s_algorithm.

[15] Scott Aaronson. Lecture 10: Quantum computing. URL https://www.scottaaronson.com/democritus.

[16] Sanjeev Arora and Boaz Barak. Computational complexity: A modern approach. *Princeton University*, Enero 2007. URL https://arxiv.org/abs/quant-ph/9605043.

[17] David Reinsel. John Gantz. John Rydning. The digitization of the world. from edge to core. *Seagate*, Noviembre 2018. URL https://www.seagate.com/la/es/our-story/data-age-2025/.

[18] IDC: Executive Summary. Data growth, business opportunities, and the it imperatives. *IDC*, Abril 2014. URL https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm.

[19] Dmitry Solenov. Jay Brieler. Jay Brieler. The potential of quantum computing and machine learning to advance clinical research and change the practice of medicine. Octubre 2018. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6205278/.

[20] Dan Styer. A brief history of quantum mechanics. *Appendix A of The Strange World of Quantum Mechanics*, 1999.

[21] Bruce MacLennan. Unconventional computation. *Department of Electrical Engineering and Computer Science University of Tennessee, Knoxville*, Noviembre 2018. URL http://web.eecs.utk.edu/~bmaclenn/Classes/494-594-UC/handouts/.

[22] Scott Aaronson. Doing my oracle duty. *Scott Aaronson Blog*, 1999. URL https://www.scottaaronson.com/blog/?p=451.

[23] Jack Simons. An introduction to theoretical chemistry. *Cambridge University Press*, Abril 2003. URL http://simons.hec.utah.edu/ITCSecondEdition/TableofContents.html.

[24] Wikipedia. Adiabatic quantum computation. 2020. URL https://en.wikipedia.org/wiki/Adiabatic_quantum_computation.

[25] Wikipedia. Squid. 2020. URL https://en.wikipedia.org/wiki/SQUID.

[26] Wikipedia. Josephson effec. 2020. URL https://en.wikipedia.org/wiki/Josephson_effect.

[27] Wikipedia. Pauli exclusion principle. 2020. URL https://en.wikipedia.org/wiki/Pauli_exclusion_principle.

[28] Wikipedia. Pauli exclusion principle. 2020. URL https://en.wikipedia.org/wiki/Pauli_exclusion_principle.

[29] Wikipedia. Cooper pair. 2020. URL https://en.wikipedia.org/wiki/Cooper_pair.

[30] Giacomo Nannicini. Quantum computing, lecture 5. *IBM T.J. Watson, Yorktown Heights, NY nannicini@us.ibm.com*, Octubre 2019. URL https://researcher.watson.ibm.com/researcher/files/us-nannicini/8100_lecture_5.pdf.

[31] D-Wave. Meet d-wave. 2020. URL https://www.dwavesys.com/our-company/meet-d-wave.

[32] Wikipedia. Quadrupole ion trap. 2020. URL https://en.wikipedia.org/wiki/Quadrupole_ion_trap.

[33] Diana Prado Lopes Aude Craik. Trapping ions for quantum computing. *Conferencia: Harvard University. Center for Integrated Quantum Materials*, Abril 2019. URL https://www.youtube.com/watch?v=j1SKprQIkyE.

[34] Steve Simon. Topological quantum computing. *Conferencia: Department of Physics at Oxford University*, Junio 2012. URL https://www.youtube.com/watch?v=FAiiXp9IoBk.

[35] Jonathan Hui. Qc — how to build a quantum computer with superconducting circuit? *Medium Blog*, Enero 2019. URL https://medium.com/@jonathan_hui/qc-how-to-build-a-quantum-computer-with-superconducting-circuit-4c30b1b296cd.

[36] Alexandre Blais. Superconducting qubits. *Université de Sherbrooke, Québec, Canada*, 2015. URL http://www.quantum-lab.org/qip2015/slides/QIP2015-Alexandre%20Blais.pdf.

[37] Frank Arute. Kunal Arya. [...] John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature magazine*, Octubre 2019. URL https://www.nature.com/articles/s41586-019-1666-5.

[38] CaixaBank. Caixabank develops the first r&d projects aimed at applying quantum computing to financial activity in spain. *CaixaBank Blog*, Agosto 2019. URL https://www.caixabank.com/comunicacion/noticia/caixabank-develops-the-first-applying-quantum-computing-to-financial-activity-in-s
en.html?id=41876.