



Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y  
Tecnología

Máster en Computación Cuántica

Diseño optimizado de un  
circuito comparador para  
computación cuántica

**Trabajo Fin de Máster**

**Presentado por:** Francisco José Orts Gómez

**Dirigido por:** Rodrigo Gil-Merino y Rubio

**Ciudad:** Almería, España

**Fecha:** Julio de 2023



# Índice de contenidos

<b>Resumen</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	7
1.3. Estructura de la memoria . . . . .	10
<b>2. Contexto y estado de la técnica</b>	<b>12</b>
2.1. Circuitos y puertas cuánticas . . . . .	12
2.2. Puertas cuánticas que aparecen en este trabajo . . . . .	14
2.3. Puertas disponibles en la plataforma IBM Quantum . . . . .	25
2.4. Métricas . . . . .	27
2.5. Comparadores en la literatura cuántica . . . . .	31
<b>3. Objetivos</b>	<b>38</b>
3.1. Objetivo principal . . . . .	38
3.2. Objetivos secundarios . . . . .	39
3.3. Metodología de trabajo . . . . .	39
<b>4. Desarrollo del trabajo</b>	<b>42</b>
4.1. Algoritmo para la construcción del comparador . . . . .	44
<b>5. Análisis y comparativa</b>	<b>47</b>
5.1. Análisis del circuito propuesto . . . . .	47
5.2. Comparativa . . . . .	49
<b>6. Conclusiones</b>	<b>52</b>
<b>7. Trabajo futuro</b>	<b>55</b>

# Índice de ilustraciones

1.1. Esfera de Bloch. . . . .	3
1.2. Medio sumador clásico y cuántico. . . . .	6
1.3. Ejemplo de diseño digital de un comparador binario de dos dígitos. . . . .	9
1.4. Ejemplo de un medio comparador binario de dos dígitos clásico. . . . .	9
1.5. Ejemplo de un medio comparador binario de dos dígitos cuántico. . . . .	10
2.1. Puertas cuánticas del grupo Clifford+T. . . . .	13
2.2. Matrices Pauli. . . . .	14
2.3. Matriz de la puerta T. . . . .	15
2.4. Matriz de la inversa de la puerta T. . . . .	16
2.5. Matriz de la puerta S. . . . .	16
2.6. Matriz de la puerta Hadamard. . . . .	16
2.7. Matriz de la puerta CNOT. . . . .	17
2.8. Matriz de la puerta Toffoli. . . . .	19
2.9. Implementación de la puerta Toffoli. . . . .	20
2.10. Implementación de la puerta lógica-AND temporal. . . . .	21
2.11. Implementación de la puerta inversa de la lógica-AND temporal. . . . .	22
2.12. Matriz de la puerta SWAP. . . . .	23
2.13. Matriz de la puerta SWAP controlada. . . . .	24
2.14. Matriz de la puerta Peres. . . . .	24
2.15. Implementación de la puerta Peres. . . . .	25
2.16. Puertas cuánticas y su símbolo incluidas en la plataforma IBM Quantum. . . . .	26
2.17. Implementación de la puerta Toffoli utilizada por IBM. . . . .	27
2.18. Implementación de la puerta Toffoli utilizada por IBM. . . . .	27
2.19. Implementación de la puerta lógica-AND temporal. . . . .	31
2.20. Puerta CGC. . . . .	33
2.21. Comparador de Xia et al. . . . .	34
2.22. Otro comparador de Xia et al. . . . .	35
2.23. Comparador de Li et al. . . . .	35
2.24. Comparador de Orts et al. . . . .	37

4.1. Ejemplo de aplicación de la metodología de Pérez et al. . . . . .	43
4.2. Circuito propuesto. . . . .	46
4.3. Inversa del circuito propuesto. . . . .	46

# Índice de tablas

1.1. Tabla de verdad de un comparador binario de dos dígitos. . . . .	8
2.1. Tabla de verdad de la puerta CNOT. . . . .	18
2.2. Tabla de verdad de la puerta Toffoli. . . . .	20
2.3. Métricas del circuito propuesto. . . . .	32
5.1. Métricas del circuito propuesto. . . . .	49
5.2. Comparativa del circuitos. . . . .	49

# Resumen

En este trabajo se presenta un circuito comparador para computación cuántica. El circuito es capaz de determinar, dadas dos cadenas de bits  $A$  y  $B$  de cualquier longitud, si  $A$  es menor o igual a  $B$ , o bien si  $A$  es mayor que  $B$ . Aunque ya existen otros comparadores para computación cuántica, el circuito que se propone en este trabajo permite realizar dicha operación necesitando un menor número de cúbits. Los cúbits son la unidad mínima de información en computación cuántica. Como los dispositivos cuánticos actuales disponen de una cantidad muy limitada de cúbits, conseguir circuitos que optimicen su uso es crucial para este paradigma de computación. Típicamente, cada cúbit que se logra reducir del diseño de un circuito permite extender la longitud de las cadenas que el circuito puede comparar en una unidad, lo que también puede traducirse como un aumento del tamaño de los datos en un orden de magnitud en base 2. Pero lograr tal optimización no resulta trivial, pues los circuitos cuánticos tienen restricciones importantes tales como la imposibilidad de copiar valores o la exigencia de que la computación que en ellos se hace siempre debe ser reversible.

**Palabras clave:** computación cuántica, circuitos cuánticos, comparador cuántico, circuito medio comparador

# Abstract

This work presents a comparator circuit for quantum computing. The circuit is able to determine, given two bit strings  $A$  and  $B$  of any length, whether  $A$  is less than or equal to  $B$ , or whether  $A$  is greater than  $B$ . Although there are already other comparators for quantum computing, the circuit proposed in this work allows this operation to be performed with a smaller number of qubits. Qubits are the minimum unit of information in quantum computing. As current quantum devices have a very limited number of qubits, achieving circuits that optimize their use is crucial for this computing paradigm. Typically, each qubit that is successfully reduced from a circuit design allows extending the length of the strings that the circuit can compare by one unit, which can also translate as an increase in data size by an order of magnitude in base 2. But achieving such optimization is not trivial, as quantum circuits have important restrictions such as the impossibility of copying values or the requirement that the computation done in them must always be reversible.

**Keywords:** quantum computing, quantum circuits, quantum comparator, half comparator



# 1. Introducción

## 1.1. Motivación

La computación cuántica es capaz de resolver ciertos problemas de forma más eficiente que otros modelos de computación (Nielsen y Chuang, 2011). Los algoritmos de Shor y de Grover son posiblemente la mejor demostración de esta superioridad (Shor, 1999; Grover, 1996). El primero de ellos es capaz de encontrar los factores de un número en un tiempo polinomial, algo imposible para cualquier algoritmo clásico existente hoy en día. Por su parte, el algoritmo de Grover permite encontrar uno o varios elementos en una base de datos desordenada de forma más rápida que el resto de los algoritmos clásicos de búsqueda conocidos. Pero los algoritmos cuánticos que logran ventajas computacionales no se limitan a estos dos, sino que existe una gran variedad de ellos con aplicaciones en áreas tan distintas como la química (Bauer et al., 2020; Cao et al., 2019; Lee et al., 2021), el álgebra lineal (Berry et al., 2017; Harrow et al., 2009; Wossnig et al., 2018), o el aprendizaje automático (Biamonte et al., 2017; Huang et al., 2021; Y. Liu et al., 2021) entre otros.

Pese a que puede ofrecer ventajas computacionales con ejemplos como los mencionados algoritmos de Shor o de Grover, la computación cuántica está en pleno desarrollo y se enfrenta actualmente a ciertos problemas y limitaciones (Preskill, 2018). Uno de estos problemas es la escasa cantidad de recursos de los que disponen los dispositivos cuánticos actuales: poseen una limitada cantidad de cúbits (la unidad básica de información en computación cuántica), por lo que los problemas que pueden resolverse actualmente están limitados en tamaño por este número (Bharti et al., 2022). Afortunadamente, el número de cúbits del que disponen los computadores cuánticos es cada vez mayor, por lo que este problema se va reduciendo progresivamente (Chauhan et al., 2022). Pero, mientras tanto, los investigadores se ven obligados a buscar formas eficientes de representar sus algoritmos si quieren ser capaces de ejecutarlos en un número reducido de cúbits (Cerezo et al., 2021).

Otro problema, más grave aún que el anterior, es la extrema sensibilidad que tienen los computadores cuánticos actuales al ruido interno y externo (Bharti et al., 2022). El ruido interno es causado por los defectos e imprecisiones de los propios dispositivos, mientras que el externo es el causado por cualquier perturbación exterior. Los efectos del ruido afectan negativamente al estado y comunicación de los cúbits, provocando que los algoritmos cuánticos produzcan resultados erróneos. Desde el punto de vista de la implementación,

existe una intensa investigación enfocada en lograr mejores implementaciones de los cúbits, de forma que puedan reducirse los efectos del ruido (Gyenis et al., 2021; C. Wang et al., 2022). Pero hay otras formas de combatir los efectos del ruido. Por ejemplo, el propio diseño de los algoritmos (de lo que se hablará más adelante), puede realizarse teniendo en cuenta estos efectos y, hasta cierto punto, se puedan detectar e incluso corregir (Sharma et al., 2020; Xue et al., 2021). También hay otro factor importante relacionado que es la longitud del algoritmo: un algoritmo que se ejecute rápidamente estará poco tiempo expuesto al ruido y será menos propenso a sufrir sus efectos (Pérez-Salinas et al., 2020).

A partir de los dos problemas mencionados (escasez de recursos y ruido), podemos afirmar que dados dos algoritmos que resuelven un mismo problema, el más pequeño en términos del número de cúbits y operaciones necesarias será más eficiente que el otro (Asadi et al., 2020; Noorallahzadeh y Mosleh, 2019). Por supuesto, asumiendo que las operaciones sean del mismo orden. El algoritmo pequeño es más eficiente porque consumirá menos cúbits, siendo capaz de resolver con datos más grandes. Y a su vez, es más eficiente porque se verá menos afectado por el ruido por el simple hecho de que está menos tiempo expuesto a él (Preskill, 2018). En relación a la reducción del tiempo de ejecución, también puede resultar interesante paralelizar las operaciones cuando sea posible, pues también redundará en una reducción del tiempo total de ejecución y de la exposición del algoritmo al ruido (Ball et al., 2016; Takahashi y Kunihiro, 2008; Thapliyal et al., 2019).

Incluso con los inconvenientes que tienen los computadores cuánticos actuales, éstos ya se utilizan actualmente con éxito para resolver gran cantidad de problemas (Yarkoni et al., 2022). Uno de los principales responsables de ello es el propio cúbit. Su potencia es tal que, incluso con un número reducido de cúbits es posible alcanzar resultados sorprendentes si se utilizan representaciones de datos que permitan aprovechar su capacidad (del cúbit) de representar infinitos valores (Pérez-Salinas et al., 2020). Un bit puede contener uno de dos valores: 0 o 1. El bit puede representar cualquier sistema binario, como por ejemplo un interruptor que puede estar encendido o apagado. Por su parte, un cúbit puede definirse como cualquier ket unitario en  $\mathbb{C}^2$  (el conjunto de vectores columna, de tamaño 2, de números complejos), pudiendo por lo tanto tomar infinitos valores (Bernhardt, 2019). El término ket, acuñado por Paul Dirac (Dirac, 1939), es simplemente una forma de llamar a los vectores columna. Siguiendo la notación de Dirac, representamos un ket con un nombre cualquiera y una simbología propia  $|\dots\rangle$ . Dos ejemplos de estados posibles de un cúbit son

$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  y  $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ , siendo pues los nombres que hemos elegido para estos dos kets 0 y 1, respectivamente. De hecho, estos dos estados que aquí hemos notado como  $|0\rangle$  y  $|1\rangle$  son especialmente útiles, como se explica a continuación.

Cuando se trabaja con cúbits, llega un momento en que es necesario medirlos para recuperar sus valores. Se ha dicho que un cúbit puede tomar infinitos valores posibles. En la Figura 1.1, se muestra la llamada esfera de Bloch (Bloch, 1946). La esfera de Bloch se utiliza para describir gráficamente un cúbit. Se asume que la esfera tiene radio 1, siendo cualquier vector con origen en el centro de la esfera y extremo en la superficie de la misma un estado válido para el cúbit. Como hay infinitos puntos en la superficie de la esfera, tenemos infinitos vectores posibles. Sin embargo, aquí entran en juego las propiedades de la mecánica cuántica: pese a que el cúbit puede tener esos infinitos valores, al medirlo solo devolverá uno de dos valores posibles. Justo lo mismo que ocurre con los bits. En realidad, el cúbit no devolverá un valor binario, sino que el cúbit “colapsará” a uno de dos estados posibles, que nosotros interpretaremos como 0 o 1. Es decir, que al medir el cúbit, éste estará en uno de dos estados cuánticos posibles. ¿A qué dos estados puede colapsar un cúbit? Esto dependerá de cómo se mida. Para medir un cúbit se debe elegir una dirección de medida, lo que se traduce en utilizar una base ortonormal ordenada ( $|b_0\rangle, |b_1\rangle$ ) (Bernhardt, 2019). Una base de un espacio vectorial es un conjunto de vectores linealmente independientes que son capaces de generar a todos los vectores que forman parte de ese espacio vectorial. Los adjetivos ortonormal y ordenada se explican a continuación.

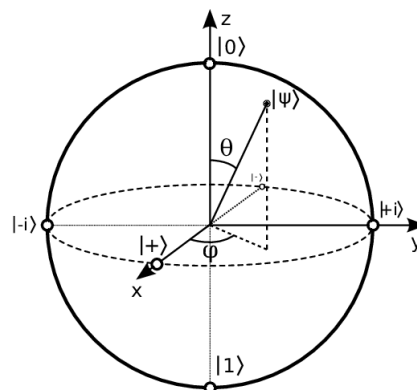


Figura 1.1: La llamada esfera de Bloch permite representar gráficamente un cúbit. Cualquier vector que lleve desde el centro de la esfera a la superficie de la misma representa un estado cuántico válido  $|\psi\rangle$ .

Una base ortonormal está compuesta por dos kets que son ortogonales (el producto interno de un vector de la base con cada uno de los otros vectores de la base es 0) y que tienen normal 1 (Nielsen y Chuang, 2011). El orden es importante (por eso lo de ordenada), puesto que si establecemos la base  $(|b_0\rangle, |b_1\rangle)$  asumimos que  $|b_0\rangle$  será el 0 y  $|b_1\rangle$  el 1 (como resultado de la medida), mientras que si la base es  $(|b_1\rangle, |b_0\rangle)$  entonces  $|b_1\rangle$  será el 0 y  $|b_0\rangle$  el 1. Todo estado  $|\psi\rangle$  de un cúbit puede expresarse como combinación lineal de una base  $(|b_0\rangle, |b_1\rangle)$ :

$$|\psi\rangle = \alpha |b_0\rangle + \beta |b_1\rangle \quad (1.1)$$

dónde  $\alpha$  y  $\beta$  son números complejos (Nielsen y Chuang, 2011). A esta idea -que el estado del cúbit puede expresarse como combinación lineal de una base- se le denomina superposición (Bernhardt, 2019), que es una de las propiedades más conocidas y características de la computación cuántica (heredada, por supuesto, de la mecánica cuántica). Al medir el cúbit en el estado  $|\psi\rangle$  y asumiendo que para medir utilizamos la base  $(|b_0\rangle, |b_1\rangle)$ , el cúbit colapsará a  $|b_0\rangle$  con una probabilidad  $\alpha^2$ , o a  $|b_1\rangle$  con una probabilidad  $\beta^2$  (Rieffel y Polak, 2011). Una base especialmente utilizada en la literatura por su simplicidad es la base  $(|0\rangle, |1\rangle)$ , de cuyos componentes ya hemos hablado anteriormente. Además, esta base es utilizada por los computadores cuánticos de IBM (Cruz et al., 2019), que son a los que tenemos acceso para realizar las ejecuciones en este trabajo. Por lo tanto, en el presente TFM se utilizará dicha base.

Existen diversas formas de programar un algoritmo en un computador cuántico, siendo la más habitual el diseño e implementación de circuitos cuánticos (Bernhardt, 2019; Nielsen y Chuang, 2011). Los circuitos cuánticos son similares en concepto a los circuitos clásicos, aunque existen diferencias importantes entre ambos tipos:

- Bit versus cúbit: La diferencia más obvia es que los circuitos clásicos trabajan con bits, mientras que los circuitos cuánticos trabajan con cúbits y sus infinitos estados posibles.
- Teorema de no clonación: Otra diferencia es que en los circuitos clásicos podemos copiar bits, lo que en electrónica digital se conoce como fan-out (Harris y Harris, 2015). Sin embargo, en computación cuántica no podemos clonar estados cuánticos (Wootters y Zurek, 2009). El origen de esta imposibilidad tiene su origen, una vez más, en la mecánica cuántica. El llamado teorema de no clonación afirma que no

podemos copiar un estado cuántico de un cúbit a otro (Park, 1970; Wootters y Zurek, 1982). La demostración de dicho teorema está fuera de los objetivos de este TFM, pero puesto que la computación cuántica tiene por objetivo emular las propiedades de la mecánica cuántica, queda claro que los circuitos cuánticos deben regirse por el teorema de no clonación.

- Reversibilidad: Una tercera diferencia entre circuitos clásicos y cuánticos es que estos últimos deben ser reversibles, también como consecuencia de la mecánica cuántica (Bernhardt, 2019).

El colapso al medir, la superposición, el principio de no clonación y la obligación de reversibilidad (que hace que nunca se pierda información) no son las únicas propiedades de la mecánica cuántica (Merzbacher, 1998), pero son las más importantes para el entendimiento de este trabajo.

Toda función que puede llevarse a cabo a través de un circuito clásico puede implementarse en un computador cuánticos (Deutsch, 1985). Por lo tanto, a la hora de construir un circuito cuántico que implemente una función clásica, puede resultar interesante consultar la literatura sobre electrónica digital clásica para encontrar diseños optimizados (Thomsen et al., 2010). Por ejemplo, los sumadores aritméticos son circuitos muy interesantes en computación cuántica, pues son necesarios en múltiples algoritmos cuánticos como el algoritmo de Shor (Shor, 1999). Estos circuitos sumadores están contruidos siguiendo la teoría de la electrónica digital clásica, que ofrece diseños que se han ido optimizado a lo largo de décadas de estudio (Harris y Harris, 2015; Patterson et al., 1990). Sin embargo, la requerida reversibilidad añade una dificultad extra al diseño de circuitos cuánticos (Thapliyal y Ranganathan, 2010). Como norma, esta reversibilidad implica que se necesiten más recursos para implementar cualquier función respecto a su análoga clásica. A modo de ejemplo, la Figura 1.2 muestra dos circuitos que realizan la suma de dos bits  $A$  y  $B$ , siendo el circuito de la izquierda clásico y el de la derecha cuántico. La suma de dos bits produce como resultado otros dos bits: la suma  $S = A + B$  y el acarreo  $C = A \oplus B$ . No es necesario en este momento conocer el funcionamiento de las puertas lógicas o cuánticas utilizadas, puesto que solo se pretende demostrar la necesidad de recursos extra en el caso cuántico. Por simplicidad para este ejemplo, en el circuito cuántico se asume que las entradas solo pueden ser  $|0\rangle$  y  $|1\rangle$ , representando ambos estados a los bits 0 y 1 respectivamente (y lo mismo ocurre con las salidas). Esta operación se conoce como “medio

suma” (Harris y Harris, 2015). El término “medio” se debe a que la operación no admite acarreo de entrada. Puede notarse, en esta operación tan sencilla, que el circuito cuántico, para mantener la reversibilidad, necesita una entrada más. También, por estos motivos, tiene una salida más que el clásico. Asimismo, son necesarias más operaciones en la versión cuántica que en la clásica.

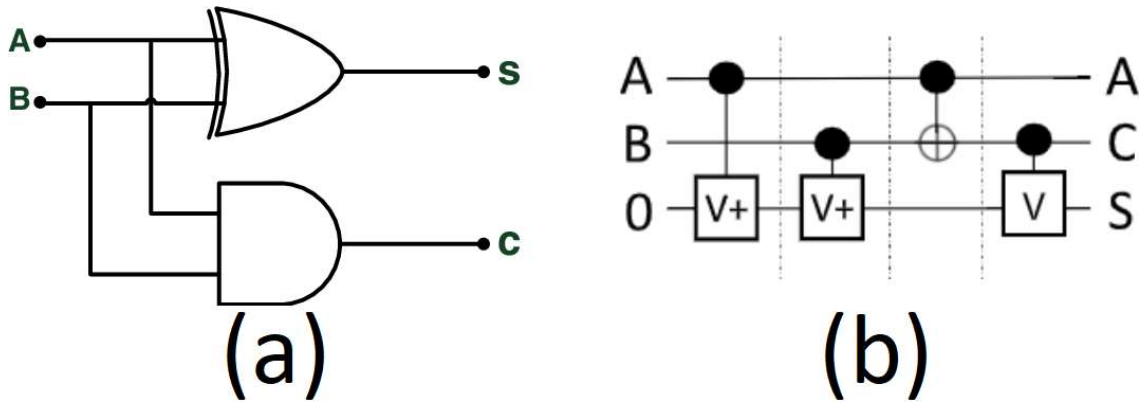


Figura 1.2: (a) Esquema, a nivel digital, de un medio sumador clásico (Yaghoubi et al., 2014). (b) Esquema de un medio sumador cuántico (Hung et al., 2006).

En base a todo lo expuesto, podemos concluir que todo circuito cuántico que implemente una función útil, y que sea pequeño, será un recurso valioso incluso aunque no presente por sí mismo ninguna ventaja cuántica (Pérez-Salinas et al., 2020). Será valioso porque se puede utilizar en algoritmos y circuitos cuánticos mayores que sí ofrecen algún tipo de beneficio. Volviendo al caso de los sumadores, un circuito cuántico que realice la suma aritmética, como los expuestos anteriormente, no ofrece ninguna ventaja por sí solo. Sin embargo, tal y como hemos visto, son necesarios para poder implementar algoritmos cuánticos útiles como el de Shor (Orts et al., 2020). Un sumador optimizado redundará en el beneficio total del circuito que implemente el algoritmo de Shor. Si, por ejemplo, emplea pocos cúbits, dejará libres más cúbits para que otras operaciones del algoritmo puedan utilizarlos y así resolver la factorización para números mayores. Lo mismo ocurre con muchas otras operaciones, incluida la comparación (Xia et al., 2019; Xia et al., 2020).

En este trabajo se propone un circuito medio comparador para computación cuántica. El funcionamiento de un medio comparador se detalla en la siguiente Sección, pero de forma resumida se puede decir que comprueba si un valor  $A$  está por encima de un determinado umbral, o no (Chu y Current, 1999). La necesidad de este tipo de circuitos en computación

cuántica se justifica por ser una operación que resulta fundamental para gran cantidad de algoritmos cuánticos de probada eficacia en, por ejemplo, el tratamiento de imágenes (Du et al., 2022; Orts et al., 2021; Yan et al., 2017) o el cálculo matemático (Bodasingi et al., 2022; H. Li, 2022; Zhao et al., 2022). El circuito propuesto está optimizado en el número de cúbits necesario para su implementación, y en el de operaciones necesarias para realizar la mencionada operación.

## 1.2. Planteamiento del problema

La comparación es una operación básica de cualquier sistema que opere con números, incluidos los computadores clásicos (Floyd, 2011). Dadas dos cadenas de bits  $A$  y  $B$  que representan cada una un número binario, un circuito comparador devuelve el resultado de la comparación a través de dos o tres bits. En el caso de funcionar con dos bits, el resultado se devuelve como un número binario. Por ejemplo, 00 o 11 si  $A = B$  (y valen 0 o 1, respectivamente), 10 si  $A > B$ , y 01 si  $A < B$ . Para mayor claridad, se muestra este supuesto en la Tabla 1.1. En dicha tabla,  $A_1$  y  $A_0$  son los dígitos de  $A$ ,  $B_1$  y  $B_0$  los de  $B$ , y los dígitos  $S_1$  y  $S_0$  se corresponden con la salida de la comparación. No obstante, para evitar tener que interpretar el resultado, es habitual proporcionar tres salidas: una para cada caso posible. Por ejemplo, para el caso  $A = 1$  y  $B = 0$ , la salida correspondiente a  $A > B$  será 1, y las restantes salidas serán 0. Si se prefiere trabajar con lógica inversa (Harris y Harris, 2015), los resultados serán los opuestos, lo importante es poder distinguir el resultado correcto de los restantes. Un ejemplo de un comparador de solo dos dígitos se muestra en la Figura 1.3.

Existe una versión reducida de estos comparadores, llamada medio comparador (Xia et al., 2018). El medio comparador realiza una función similar a la del anterior comparador (al que llamaremos comparador completo a partir de ahora, para distinguir claramente a ambos). Sin embargo, el medio comparador solo identifica si  $A > B$  o no (Angulo et al., 2007). Es decir, devolverá 1 si  $A > B$ , o bien 0 si  $A \leq B$ . El medio comparador es de gran utilidad para una gran cantidad de aplicaciones en las que es necesario comprobar si un valor determinado supera (o no) un umbral, sin necesidad de distinguir el caso  $A = B$  del caso  $A < B$  (Harris y Harris, 2015). Por supuesto, también puede diseñarse de forma que compruebe si el valor está por debajo, o no, de un valor umbral en lugar de comprobar si está por encima de él (Harris y Harris, 2015). Además, el medio comparador tiene la

$A_1$	$A_0$	$B_1$	$B_0$	$S_1$	$S_0$
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

Tabla 1.1: Tabla de verdad de un comparador binario de dos dígitos. Se comparan dos cadenas de dos bits  $A$  y  $B$ , y se produce una salida de dos bits  $S$  que será 00 cuando  $A$  y  $B$  son iguales, 01 cuando  $A < B$ , y 10 cuando  $A > B$ .

ventaja, respecto al comparador completo, de que necesita menos recursos en forma de puertas lógicas, lo que se traduce en una implementación física más económica y reducida en espacio, y un solo bit de salida. En la Figura 1.4 se muestra un medio comparador de números de un solo dígito. Puede apreciarse a simple vista que necesita menos recursos que el comparador completo de la Figura 1.3. Conforme aumente el número de dígitos (bits) de  $A$  y de  $B$  en ambos casos, la diferencia será aún mayor.

Pasando ya a computación cuántica, el concepto de medio comparador es exactamente igual que el expuesto hasta ahora: dados dos números  $A$  y  $B$ , se requiere un circuito que determine si  $A > B$  o no. Es importante remarcar que no se comparan estados cuánticos, operación que sin duda resulta más compleja (Andersson et al., 2006), sino que se busca una forma de representar dígitos binarios utilizando cúbits, y devolver una salida interpretable



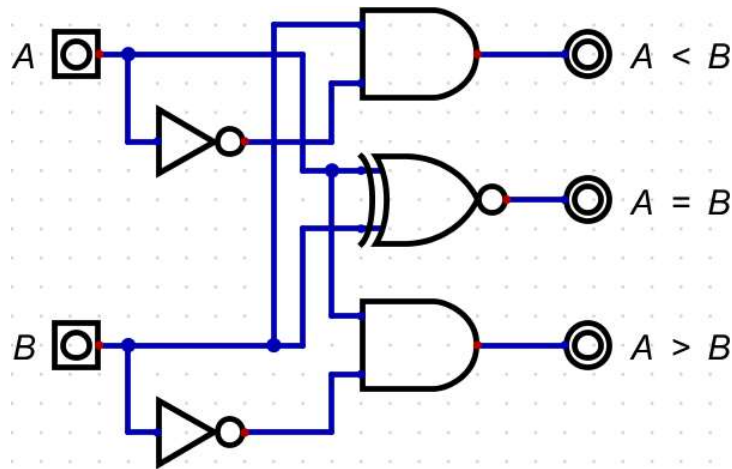


Figura 1.3: Ejemplo de diseño digital de un comparador binario de dos dígitos. Hay dos entradas,  $A$  y  $B$ , y tres salidas posibles. Solo una de las salidas devolverá un 1 en función de los valores de entrada, siendo el valor de las dos salidas restantes 0. Fuente: elaboración propia utilizando el software Logisim (Burch, 2002).

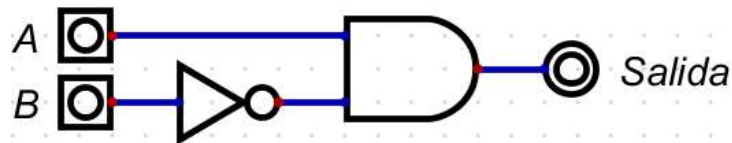


Figura 1.4: Ejemplo de diseño digital de un medio comparador binario de dos dígitos. Hay dos entradas,  $A$  y  $B$ , y una sola salida posible. La salida será 1 si  $A > B$ , y 0 en caso contrario. Fuente: elaboración propia utilizando el software Logisim (Burch, 2002).

a través de una medida o bien por otro circuito cuántico como potencial entrada (H. Li et al., 2020). Todo ello cumpliendo con los requisitos impuestos por la mecánica cuántica, como todos los circuitos cuánticos (Nielsen y Chuang, 2011). Es directo comprobar que los circuitos mostrados en las Figuras 1.3 y 1.4 no son circuitos cuánticos válidos por diversos motivos: realizan unión de cables, hay diferente número de entradas que de salidas, las puertas lógicas utilizadas poco tienen que ver con las puertas cuánticas, y el circuito no resulta reversible. Por ejemplo, si la puerta AND de la Figura 1.4 devuelve 0, no es posible recuperar los valores originales de  $A$  y  $B$  que ha provocado este valor. Obtener un circuito equivalente a este en términos cuánticos es sencillo y puede realizarse con tres cúbits que solo trabajen con la base  $|0\rangle$  y  $|1\rangle$  (y como ya hemos dicho, interpretando tales estados como 0 y 1) y utilizando solo tres puertas cuánticas. Se muestra este circuito en la Figura 1.5. Sin embargo, obtener un medio comparador eficiente para el caso general de números de  $N$

bits no resulta trivial (Xia et al., 2018), como lo demuestra la gran cantidad de alternativas expuesta en el Capítulo 2.

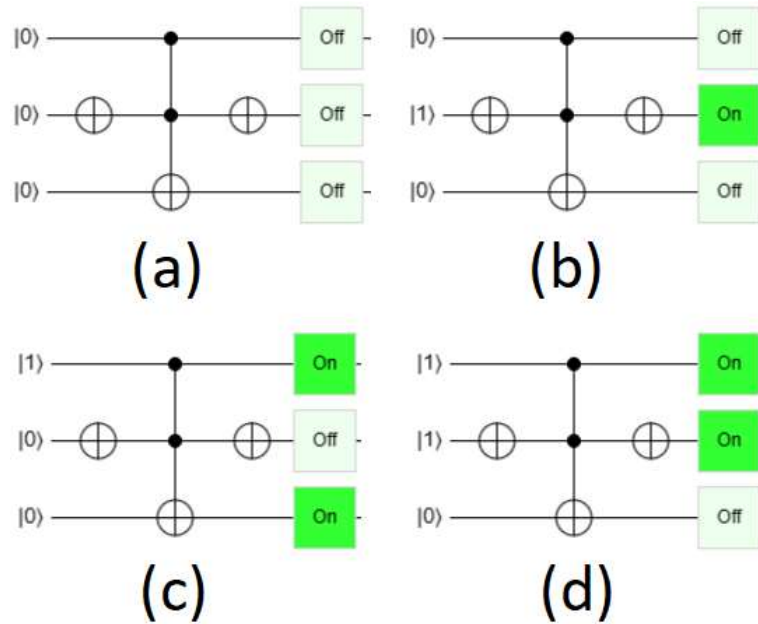


Figura 1.5: Diseño equivalente al mostrado en la Figura 1.4, pero para computación cuántica. Por motivos de claridad, se muestran los cuatro casos posibles en función de las entradas. Dados tres cúbits  $A$ ,  $B$ , y  $C$  (empezando por arriba), el circuito devuelve  $C = |1\rangle$  si  $A > B$  (subfigura c), y  $C = |0\rangle$  en caso contrario (subfiguras a, b y d). Fuente: elaboración propia utilizando el simulador Quirk (Gidney, 2023).

### 1.3. Estructura de la memoria

El resto del trabajo se estructura como se indica a continuación:

- El Capítulo 2 introduce las métricas utilizadas para medir un circuito cuántico y hace una revisión de los trabajos existentes en la literatura sobre medio comparadores para computación cuántica.
- El Capítulo 3 expone de forma clara los objetivos perseguidos en este trabajo.
- El Capítulo 4 presenta el circuito propuesto y explica en detalle cómo es posible reproducirlo en cualquier simulador o dispositivo cuántico real.
- El Capítulo 5 realiza un análisis del circuito propuesto en base a las métricas indicadas en el Capítulo 2, y realiza una comparativa entre el circuito propuesto y los

disponibles en la literatura.

- El último Capítulo presenta las conclusiones y los posibles trabajos futuros.
- Finalmente, el Anexo A incluye un resumen del trabajo realizado en formato de artículo de investigación.

## 2. Contexto y estado de la técnica

### 2.1. Circuitos y puertas cuánticas

Ya se ha mencionado que la forma más habitual de programar un computador cuántico actual es utilizando circuitos (Combarro y Gonzalez-Castillo, 2023). De forma similar a como ocurre con los circuitos clásicos, los cuánticos se construyen utilizando puertas (Brylinski y Brylinski, 2002). Las puertas “cuánticas” son la extensión natural de las puertas lógicas al mundo cuántico (Bernhardt, 2019). Toda puerta cuántica puede representarse matemáticamente como una matriz unitaria compleja (Bernhardt, 2019). Las puertas cuánticas operan sobre los cúbits, modificando sus estados (Nielsen y Chuang, 2011). Volviendo a la esfera de Bloch (Figura 1.1), una puerta cuántica que actúe sobre un cúbit modificará el estado actual de dicho cúbit, aunque por exactitud sería correcto afirmar que también podría no tener efecto alguno sobre dicho cúbit (Bernhardt, 2019).

En electrónica digital, existe un número limitado de puertas lógicas. Para trabajar sobre un bit, solo existen dos puertas lógicas posibles: la identidad y la NOT (Patterson et al., 1990). La primera deja el bit como está, y la segunda lo invierte. A su vez, existe un pequeño conjunto de puertas que actúan sobre dos bits: la AND, la OR, la XOR, y sus versiones negadas (Patterson et al., 1990). En computación cuántica, existen infinitas puertas cuánticas posibles para trabajar sobre un solo cúbit (Bernhardt, 2019; Nielsen y Chuang, 2011). Afortunadamente, existen conjuntos de puertas cuánticas universales que permiten aproximar a todas las demás (Bernhardt, 2019). Un ejemplo de conjunto universal nos lo ofrece el llamada grupo Clifford+T. En la Figura 2.1 se muestran algunas de las puertas pertenecientes al grupo Clifford+T. En realidad, hay otras puertas pertenecientes al grupo, pero pueden ser generadas utilizando las mostradas en la Figura 2.1.

Para entender el funcionamiento de las puertas, supongamos un sencillo ejemplo. Tenemos un único cúbit que inicialmente está en el estado  $|0\rangle$ . Si aplicamos a este cúbit la puerta cuántica mostrada en la Figura 2.1 (f), llamada Pauli-X y cuyo funcionamiento detallaremos más adelante (de momento, solo nos interesa la matriz mostrada en dicha figura), realizaremos la siguiente operación:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (2.1)$$



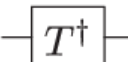

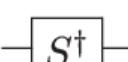


(a)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
(b)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i \cdot \frac{\pi}{4}} \end{bmatrix}$
(c)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{-i \cdot \frac{\pi}{4}} \end{bmatrix}$
(d)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
(e)		$\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$
(f)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
(g)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Figura 2.1: Puertas cuánticas del grupo Clifford+T: (a) Puerta Hadamard, (b) puerta T, (c) puerta adjunta de la T, (d) puerta S, (e) puerta adjunta de la S, (f) puerta Pauli-X, y (g) puerta CNOT. Se muestra, para cada puerta, su símbolo y su matriz (Yan et al., 2014).

Podemos ver que el resultado de aplicar la puerta Pauli-X sobre un cúbit en el estado  $|0\rangle$  es que el cúbit pasa a estar e el estado  $|1\rangle$ . Puertas cuánticas diferentes producirán resultados diferentes en un mismo cúbit. Incluso la misma puerta cuántica producirá resultados diferentes sobre estados cuánticos diferentes ([bernhardt2019quantum0](#)). Por ejemplo:

$$X |0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (2.2)$$

En base a los resultados mostrados por la puerta Pauli-X en las Ecuaciones 2.1 y 2.4, puede entenderse el por qué a la puerta Pauli-X se la llama puerta comúnmente puerta NOT por algunos autores (Schmidt-Kaler et al., 2003; Smolin y DiVincenzo, 1996).

## 2.2. Puertas cuánticas que aparecen en este trabajo

Se identifican y explican a continuación las puertas cuánticas utilizadas en este trabajo. Algunas de estas puertas se utilizan en el circuito propuesto, mientras que otras se utilizan en los circuitos estudiados. A su vez, hay algunas puertas que no se utilizan directamente en tales circuitos, pero que son necesarias para implementar a otras puertas cuánticas.

- **Puertas Pauli:** Las cuatro matrices Pauli son matrices de tamaño  $2 \times 2$  (Pauli, 1988), tal y como se muestra en la Figura 2.2. Se pueden aplicar pues a un único cúbit. Aunque las cuatro matrices resultan de gran interés para la computación cuántica (Nielsen y Chuang, 2011), en este trabajo solo se utiliza la Pauli-X (llamada X en la Figura 2.2), y muy casualmente la Pauli-Z (llamada Z en la Figura 2.2) como parte de la implementación de otra puerta cuántica descrita más adelante. Se describe a continuación la puerta Pauli-X, que es por tanto la que tiene interés para este trabajo.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figura 2.2: Las matrices Pauli I, X, Y, y Z son posiblemente las matrices ampliamente utilizadas en computación cuántica e información cuántica (Nielsen y Chuang, 2011).

La puerta que implementa a la matriz X ya ha sido mencionada en la Sección anterior, en la Ecuación 2.1, donde además se mencionó que es común encontrar trabajos en los que se le denomina puerta NOT debido a que si se aplica sobre un cúbit con el estado  $|0\rangle$  se obtiene el estado  $|1\rangle$ , y viceversa. Sin embargo, la analogía con la puerta clásica NOT solo se cumple si trabajamos con estas bases. Lo que en realidad hace la puerta Pauli-X es intercambiar las amplitudes de los estados base sobre los que trabaja. Dicho formalmente, dado un cúbit en el siguiente estado:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (2.3)$$

el efecto de aplicar una puerta Pauli-X sobre dicho cúbit será el siguiente:

$$|\psi\rangle = \beta |0\rangle + \alpha |1\rangle \quad (2.4)$$

Lo que hace, pues, la puerta Pauli-X es intercambiar las amplitudes de los estados base, y no una negación de estados tal y como la entendemos en computación clásica, que implica transformar un 1 en 0, o un 0 en 1.

- **T:** La llamada puerta T introduce una fase de  $\frac{\Pi}{4}$  en el estado del cúbit (Barenco et al., 1995). Su matriz  $2 \times 2$  se muestra en la Figura 2.3. La puerta T no se usa directamente en este trabajo, pero sí es requerida en la implementación de otras puertas que sí forman parte de los circuitos aquí estudiados, como por ejemplo la puerta Toffoli (se estudia más adelante en esta misma Sección). Una peculiaridad de esta puerta es que hoy en día no es posible su simulación de forma eficiente en un computador clásico (Yuan et al., 2022). Además, el coste de su implementación es más elevado que el del resto de puertas cuánticas de su mismo tamaño (Litinski, 2019). Este coste extra ha generado gran interés en el mundo de la investigación para conseguir circuitos cuánticos que reduzcan el número de puertas T necesarias para su implementación, lo que supone un ahorro en el coste de dichos circuitos (H. Li et al., 2020; Orts et al., 2021; Thapliyal et al., 2019).

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Figura 2.3: La matriz de la puerta T tiene un tamaño de  $2 \times 2$ . Es una puerta que no puede ser simulada de forma eficiente por los computadores clásicos, además de tener un coste superior al del resto de puertas (Litinski, 2019).

- **Inversa de la T:** No causa sorpresa que la puerta inversa de la puerta T, que se denotará como T', introduzca una fase de  $-\frac{\Pi}{4}$  (Barenco et al., 1995). Esta puerta comparte las características de coste elevado e imposibilidad de simulación clásica eficiente con la puerta T (Litinski, 2019). Se muestra su matriz en la Figura 2.4.
- **S:** Similar a la puerta T, pero en este caso introduce una fase de  $\frac{\Pi}{2}$  (Barenco et al., 1995). Al igual que las dos puertas anteriores, no se utiliza directamente en los comparadores estudiados o en el propuesto, pero sí en otras puertas cuánticas

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}$$

Figura 2.4: La matriz de la inversa de la puerta T permite revertir la fase inducida por la puerta T.

involucradas en ellos. La puerta S no tiene los problemas vistos para las puertas T y T'. Se muestra su matriz en la Figura 2.5.

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

Figura 2.5: La matriz de la puerta S realiza un giro de 90 grados alrededor del eje Z de la esfera de Bloch.

- **Hadamard:** La puerta Hadamard, a veces llamada simplemente H, es de gran importancia para la computación cuántica (Nielsen y Chuang, 2011), aunque en este trabajo su uso se limita al de componente para la construcción de otras puertas de mayor tamaño. Se muestra su matriz en la Figura 2.6. Esta puerta se utiliza fundamentalmente para poner los cúbits en superposición (Bernhardt, 2019).

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

Figura 2.6: Matriz de la puerta Hadamard. La puerta Hadamard permite poner en superposición el estado de un cúbit.

- **CNOT:** El término CNOT, del inglés Controlled-NOT, hace referencia a una puerta Pauli-X controlada (Bernhardt, 2019). Este término es ampliamente utilizado en la literatura sobre circuitos cuánticos (H. Li et al., 2022; P. Li et al., 2020; Thapliyal et al., 2019), por lo que lo mantendremos en este trabajo. Es una puerta controlada de dos cúbits (Nielsen y Chuang, 2011). Una puerta controlada siempre tiene uno o más cúbits de control, y un único cúbit objetivo. Sobre el cúbit objetivo se aplicará una operación  $U$  (una puerta cuántica  $2 \times 2$ , como por ejemplo la anterior Pauli-X), pero solamente si el estado del cúbit o cúbits de control es  $|1\rangle$ . En el caso concreto



de la puerta CNOT, se tiene un cúbit de control  $c$  y un cúbit objetivo  $o$ , de forma que se aplica una operación equivalente a aplicar una puerta Pauli-X sobre  $o$  pero solo si  $c = |1\rangle$ . Esta operación se nota como  $|c\rangle \oplus |o\rangle$  (Nielsen y Chuang, 2011). La matriz asociada a la puerta CNOT se muestra en la Figura 2.7).

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

Figura 2.7: Matriz de la puerta CNOT. La puerta CNOT es una puerta  $4 \times 4$  con un cúbit de control y un cúbit objetivo. Está íntimamente relacionada con la puerta Pauli-X, y de manera simplificada se puede decir que aplica una puerta Pauli-X sobre el cúbit objetivo siempre que el cúbit de control esté en el estado  $|1\rangle$ , y no hace nada en caso de estar en el estado  $|0\rangle$ .

Para mayor claridad, se muestra la tabla de verdad (Tabla 2.1) de la puerta CNOT sobre dos cúbits considerando las bases canónicas de  $C^2$ . Pero es importante notar que esta tabla solo sirve para dar una idea intuitiva del funcionamiento de la puerta, pues la CNOT podría aplicarse a los infinitos estados cuánticos posibles de esos cúbits. Esta puerta resulta muy similar a la puerta clásica XOR, que devuelve 0 cuando las dos entradas son iguales, y 1 cuando son diferentes (Harris y Harris, 2015). No obstante, la puerta XOR tiene una única salida, mientras que la CNOT tiene una salida (cúbit, en este caso) extra por motivos de reversibilidad. En realidad, la salida extra es simplemente la entrada  $c$ , que mantiene su valor. Hay una situación que puede resultar contraintuitiva al no ocurrir nada parecido en computación clásica (Nielsen y Chuang, 2011): en ciertas situaciones, el cúbit de control podría cambiar su valor tras aplicarle esta puerta. Esta situación no puede deducirse a partir de la Tabla 2.1, y es por ello que antes se ha remarcado su incompletitud como descriptora de la puerta CNOT. Se dan más detalles de este curioso fenómeno a continuación.

Es importante también mencionar que esta puerta está íntimamente relacionada con el entrelazamiento (Bernhardt, 2019). Una vez remarcado que la puerta se puede aplicar sobre cualquier estado cuántico, un ejemplo interesante viene al aplicarla

$c$	$o$	$c'$	$o' = c \oplus o$
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

Tabla 2.1: Tabla de verdad de la puerta CNOT.  $c$  y  $o$  son los cúbits de control y objetivo, respectivamente.  $c'$  y  $o'$  muestran el estado de los cúbits  $c$  y  $o$  tras aplicar la puerta CNOT. Como todas las puertas cuánticas, la puerta CNOT puede aplicarse sobre cualquier estado cuántica, y no únicamente sobre los estados  $|0\rangle$  y  $|1\rangle$ . Por lo tanto, esta tabla de verdad puede utilizarse para ayudar a explicar los efectos que tiene esta puerta, pero entendiendo que su aplicación no está limitada a los casos mostrados en dicha tabla.

sobre los cúbits con el estado  $c = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  y  $o = |0\rangle$ . Si ahora se aplica sobre ellos una puerta CNOT (respetando la nomenclatura que hemos usado hasta ahora,  $c$  será el cúbit de control y  $o$  el cúbit objetivo), se llega a un estado cuántico en el que no podemos describir por separado el estado de cada cúbit:  $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ . Este estado es uno de los famosos estados de Bell (Bell, 1964), cuyas propiedades son de especial interés para la comunidad científica (Y.-H. Kim et al., 2001; Tittel et al., 2000).

- **Toffoli:** La puerta Toffoli es una extensión de la puerta CNOT (Toffoli, 1980). De hecho, en algunas fuentes se le denomina CCNOT (Reed et al., 2012). Es similar en comportamiento a la puerta CNOT, pero con dos cúbits de control. Dados dos cúbits  $c_1$  y  $c_2$  que actuarán como cúbits de control, y un tercer cúbit  $o$  que actuará como cúbit objetivo, se invertirán las amplitudes de  $|0\rangle$  y  $|1\rangle$  del estado de  $o$  pero solamente si tanto  $c_1$  como  $c_2$  están en el estado  $|1\rangle$ . Se muestra la matriz correspondiente a esta puerta en la Figura 2.8. Se puede ver que su tamaño es de  $8 \times 8$ . Esto es debido al producto tensorial de las bases de los tres cúbits involucrados (Bernhardt, 2019). Existe en computación clásica una puerta lógica llamada NAND, que acepta dos entradas y devuelve siempre un 1, salvo cuando ambas entradas valen 0, en cuyo caso devuelve un 0 (Harris y Harris, 2015). Esta puerta NAND es una puerta universal: con ella, y solo con ella, se puede implementar cualquier función posible en computación clásica (Roth Jr et al., 2020). La puerta Toffoli es capaz de reproducir

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

Figura 2.8: La matriz de la puerta Toffoli es de tamaño  $8 \times 8$ . Puede verse el crecimiento de las matrices viendo los tamaños de las matrices Pauli, la CNOT, y ahora la Toffoli, conforme aumenta el número de cúbits.

el comportamiento de la puerta NAND, como puede verse en su tabla de verdad (Tabla 2.2). Si se observan aquellos casos en los que  $o$  vale inicialmente 1, puede verse como la puerta Toffoli se comporta de forma similar a lo descrito para la puerta NAND:  $o'$  valdrá siempre 1 ( $|1\rangle$ ), excepto cuando  $c_1$  y  $c_2$  valgan ambos 1. Este resultado es especialmente importante: si una puerta cuántica es capaz de reproducir una puerta clásica universal, entonces es posible implementar cualquier función clásica en computación cuántica.

La puerta es ampliamente utilizada en circuitos cuánticos de todo tipo, y es actualmente objeto de un intenso estudio para tratar de optimizar su implementación (Y. Kim et al., 2022). De entre las diversas implementaciones propuestas, posiblemente la más utilizada en la literatura (Bharti et al., 2022; Campbell et al., 2017; Murali et al., 2019) es la propuesta por Amy et al. (Amy et al., 2013). Como puede verse en la Figura 2.9, está compuesta de varias puertas CNOT, además de otras puertas de un solo cúbit ya estudiadas en esta Sección. Consta pues de 16 puertas cuánticas de tamaños  $2 \times 2$  o  $4 \times 4$ . Existen otras implementaciones para la puerta Toffoli (Orts et al., 2022), pero no son de interés para este trabajo puesto que como se verá en la Sección siguiente, se terminará utilizando una versión concreta de dicha puerta como consecuencia de la plataforma de trabajo elegida.

Por motivos de claridad, conviene resaltar que las diversas implementaciones de la puerta Toffoli no modifican su funcionamiento, sino que permiten realizar su funcio-

$c_1$	$c_2$	$o$	$c'_1$	$c'_2$	$o' = c_1c_2 \oplus o$
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 0\rangle$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

Tabla 2.2: Tabla de verdad de la puerta Toffoli.  $c_1$  y  $c_2$  son los cúbits de control, y  $o$  el cúbit objetivo.  $c'_1, c'_2$  y  $o'$  muestran el estado de los cúbits tras aplicar la puerta Toffoli. Si  $o$  se establece inicialmente a  $|0\rangle$ , la puerta Toffoli se comporta como una puerta AND clásica que realiza la operación  $c_1c_2$ , mientras que si  $o$  se establece a  $|1\rangle$  se comporta como una puerta NAND clásica (Roth Jr et al., 2020). Este último comportamiento es excepcionalmente importante, pues demuestra que la puerta Toffoli es una puerta universal para computación clásica en computadores cuánticos.

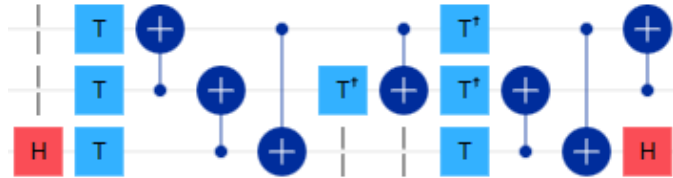


Figura 2.9: Esta implementación de la puerta Toffoli es la más eficiente disponible actualmente en términos de número de puertas necesarias (Amy et al., 2013). Las puertas más complejas, como esta, no suelen hacerse con una sola operación, sino que involucran operaciones (puertas) más pequeñas.

alidad utilizando menos recursos o de forma más rápida. Resulta obvio decir que, si se modifica el comportamiento de la puerta Toffoli, ya no sería la puerta Toffoli. Un ejemplo sencillo de esto es la puerta Lógica-AND temporal, descrita justo a continuación.

- **Logica-AND temporal:** La puerta Lógica-AND temporal (Gidney, 2018) realiza casi la misma función que la puerta Toffoli, pero presenta ciertas diferencias en su

comportamiento que la hacen interesante. Dicha implementación se muestra en la Figura 2.10. Las dos primeras puertas, identificadas como H y T respectivamente (se detalla su funcionamiento un poco más adelante), ponen al tercer cúbit en el estado  $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$ . Para que la puerta Lógica-AND temporal funcione correctamente, este cúbit tiene que estar inicialmente en el estado  $|0\rangle$ , para que al aplicar las puertas H y T se llegue al estado mencionado. Vemos una primera diferencia con respecto a la puerta Toffoli: esta otra puerta no puede partir con un cúbit objetivo en el estado  $|0\rangle$ , con las implicaciones que ello tiene. Por ejemplo, no podrá simular a la puerta NAND clásica ni por tanto implementarse cualquier función lógica utilizando esta puerta. Una pequeña ventaja de la puerta Lógica-AND temporal frente a la puerta Toffoli sería que necesita menos puertas de tamaños  $2 \times 2$  o  $4 \times 4$  para su implementación: 13 puertas frente a las 16 de la puerta Toffoli. Sin embargo, la principal ventaja de esta puerta y que la hace tan atractiva en gran cantidad de trabajos de investigación es su capacidad para revertirse de forma muy rápida y utilizando muy pocas puertas (Cao et al., 2019; Daley et al., 2022; Gidney y Ekerå, 2021). Pero para entender esto, es necesario introducir otra puerta: la puerta que revierte a la Lógica-AND temporal.

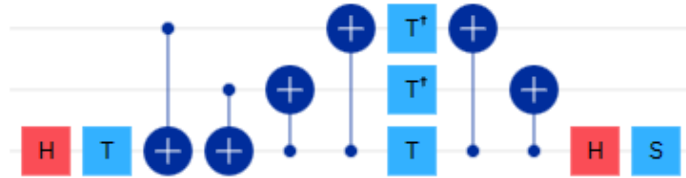


Figura 2.10: Implementación de la puerta llamada lógica-AND temporal propuesta por Gidney (Gidney, 2018). Para que funcione, el cúbit identificado como  $C$  debe inicializarse en el estado  $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle)$ . Puede verse que consta de 11 puertas de uno o dos cúbits, que tiene un retraso de  $9\Delta$ , un cúbit auxiliar ( $C$ , puesto que es un valor de entrada constante), y que no tiene salidas basuras puesto que al finalizar  $C$  contiene el resultado, y  $A$  y  $B$  tienen sus valores originales.

- **Inversa de la puerta Lógica-AND temporal:** Como norma, se considera una buena práctica revertir las operaciones realizadas en computación cuántica una vez han cumplido su cometido (Mohammadi y Eshghi, 2009). Los motivos de ello se explican en detalle en una Sección posterior. Baste ahora decir que el objetivo de revertir operaciones es dejar los cúbits en su estado original para que puedan ser utilizados por otras operaciones (Bennett, 1973). En el caso de la puerta Toffoli,

revertir dicha operación requiere aplicar otra puerta Toffoli. Resulta trivial que el coste en términos de puertas cuánticas será el doble: 16 puertas para implementar la puerta Toffoli, y otras 16 para revertir la operación y dejar libres esos cúbits.

En el caso de la puerta lógica-AND temporal, su autor propuso una segunda puerta (que llamaremos inversa de la lógica-AND temporal) que permite revertirla sin necesidad de tener que aplicar otra vez dicha puerta (Gidney, 2018). Para ello, utilizó una estrategia definida en un trabajo anterior (Jones, 2013) para simplemente medir el cúbit objetivo y, una vez medido, realizar una sencilla operación mediante una puerta Z. Queda fuera del ámbito de este trabajo detallar el funcionamiento de esta técnica, pero como las puertas lógica-AND temporal y su inversa aparecen en uno de los circuitos estudiados, resulta necesario tener nociones fundamentales sobre dichas puertas. Es suficiente saber que revertir esta puerta tendrá un coste de tan solo dos puertas cuánticas (y una medición). En la Figura 2.11 se muestra la puerta inversa de la lógica-AND temporal.

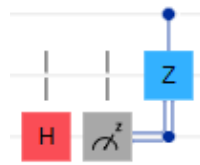


Figura 2.11: Implementación de la puerta inversa de la lógica-AND temporal (Gidney, 2018). En lugar de tener que invertirse aplicando una inversión literal del circuito de la Figura 2.10, esta puerta utiliza una aproximación diferente para logrando utilizando muchos menos recursos (Jones, 2013).

- SWAP:** La puerta swap (Fredkin y Toffoli, 1982) es una puerta cuántica que actúa sobre dos cúbits, es decir, se corresponde con una matriz  $4 \times 4$  (Figura 2.12), y puede implementarse fácilmente utilizando tres puertas CNOT. A diferencia de las puertas multicúbit anteriores, esta puerta no es una puerta controlada. Es decir, que no hay un cúbit controlando que se aplique o no una operación sobre un cúbit objetivo. En este caso, la operación a realizar es el intercambio del estado cuántico de dos cúbits. Sean  $A$  y  $B$  dos cúbits, cada uno con el siguiente estado:

$$|A\rangle = \alpha |0\rangle + \beta |1\rangle, |B\rangle = \gamma |0\rangle + \theta |1\rangle \quad (2.5)$$

La puerta SWAP provocará el siguiente resultado:

$$|A\rangle = \gamma|0\rangle + \theta|1\rangle, |B\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.6)$$

Es importante matizar que esta puerta no esta copiando estados cuánticos, algo que ya se ha dicho que no es posible en computación cuántica. Lo que hace la puerta SWAP es intercambiar el valor de los dos cúbits involucrados, sin ningún tipo de copia o creación de información (Nielsen y Chuang, 2011). Pese a que inicialmente podría parecer que una puerta que solo intercambia valores es poco útil, lo cierto es que es muy utilizada en la literatura (Lye et al., 2015; Sangouard et al., 2005; Wille et al., 2014).

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

Figura 2.12: La matriz de la puerta SWAP es de tamaño  $4 \times 4$ , y en caso de aplicarse sobre dos cúbits intercambiará el valor de tales cúbits.

- **SWAP controlada:** La puerta SWAP controlada (Fredkin y Toffoli, 1982) es una puerta que involucra tres cúbits ( $8 \times 8$ , ver Figura 2.13). Es similar a la puerta anterior, pero añadiendo un cúbit de control. y utilizando dos puerta CNOT y una puerta Toffoli en lugar de tres puertas CNOT. Resulta trivial explicar que en este caso la operación de intercambio de valores entre dos cúbits solo se realizará si el cúbit de control está en el estado  $|1\rangle$ . Algunos ejemplos de uso de esta puerta incluyen comparadores (Xia et al., 2019) o circuitos para el tratamiento cuántico de imágenes (Xia et al., 2020).
- **Peres:** De forma simplificada, la puerta Peres produce un efecto similar al que producirían una puerta Toffoli aplicada sobre tres cúbits  $A, B$  y  $C$  (siendo este último el cúbit objetivo), y una puerta CNOT que realice la operación  $A \oplus B$  (Peres, 1985). Es decir, realiza las operaciones  $AB \oplus C$  y  $A \oplus B$ . No obstante, esta puerta implementa de una vez ambas operaciones, por lo que en caso de tener que realizar la sucesión de puertas descrita (Toffoli y CNOT), el uso de una puerta Peres permitirá hacer dichas operaciones con una sola puerta. La matriz de la puerta Peres puede verse

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

Figura 2.13: La matriz de la puerta SWAP controlada, de tamaño  $8 \times 8$ , permite aplicar una operación SWAP entre dos cúbits solamente si un tercer cúbit de control está en el estado  $|1\rangle$ .

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

Figura 2.14: La matriz de la puerta Peres, de tamaño  $8 \times 8$ . Esta matriz produce el mismo efecto que aplicar de forma consecutiva una matriz Toffoli y una CNOT sobre los cúbits de control de la Toffoli. Sin embargo, ambas operaciones pueden integrarse en una sola, como puede verse en esta matriz (Peres, 1985).

en la Figura 2.14. Esta puerta también permite emular a la puerta NAND (resulta trivial demostrarlo, pues se ha dicho que una de las operaciones que realiza la Peres es  $AB \oplus C$ ), por lo que también es una puerta para computación clásica.

Existen diversas implementaciones propuestas para la puerta Peres, siendo la más eficiente en número de puertas cuánticas la ofrecida por Li et al., mostrada en la Figura 2.15 (H. Li et al., 2020). Esta implementación requiere de 15 puertas cuánticas, una menos que la puerta Toffoli. En realidad, el ahorro es de dos puertas, puesto que



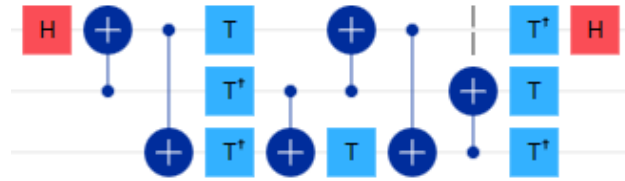


Figura 2.15: Implementación de la puerta Peres (H. Li et al., 2020). La puerta Peres permite realizar sobre tres cúbits  $A, B$  y  $C$ , de una sola vez, las operaciones  $AB \oplus C$  y  $A \oplus C$ .

en el caso de la puerta Toffoli habría que aplicar una puerta CNOT extra, operación que en la puerta Peres ya está incluida. La puerta Peres se aplica, precisamente, cuando se aplican numerosas puertas Toffoli seguidas de puertas CNOT. En su lugar, se utiliza la Peres para conseguir un ahorro de puertas cuánticas. El ahorro será mayor cuantas más puertas Toffoli y CNOT se puedan reemplazar por puertas Peres (H. Li et al., 2020).

### 2.3. Puertas disponibles en la plataforma IBM Quantum

Los circuitos a estudiar, así como el circuito propuesto, se van a implementar en la plataforma IBM Quantum (IBM, 2023) como parte de este trabajo. Por lo tanto, es importante verificar que las puertas cuánticas estudiadas en la Sección anterior están disponibles para su implementación en dicha plataforma, ya sea de forma directa, o bien mediante el uso de otras puertas cuánticas. En la Figura 2.16 se muestran aquellas puertas (y su símbolo) que están directamente incluidas en la plataforma IBM Quantum, y que por tanto se pueden utilizar directamente. Estas puertas no son todas las disponibles en la plataforma, sino que es un listado de las puertas que se necesitan para implementar los circuitos recogidos en este trabajo.

En base a dicha lista, queda patente que las siguientes puertas cuánticas necesarias no están incluidas en la plataforma y será necesario recurrir a implementaciones a partir de otras puertas:

- Lógica-AND temporal: se usará la implementación mostrada en laa Figura 2.10.
- Inversa de la lógica-AND temporal: se usará la implementación mostrada en laa Figura 2.11.











Puerta	Símbolo	Puerta	Símbolo
Pauli-X		T	
Pauli-Z		T'	
Hadamard		S	
CNOT		Swap	
Toffoli		Swap controlada	

Figura 2.16: Nombre y símbolo de las puertas cuánticas necesarias en este trabajo incluidas en la plataforma IBM Quantum.

Por lo demás, solo un detalle requiere de atención. Aunque la puerta Toffoli está disponible aparentemente de forma nativa, en realidad está implementada utilizando el diseño mostrado en la Figura 2.17. El motivo de que utilicen este diseño y no el descrito por Amy et al. mostrado en la Sección anterior (Amy et al., 2013) no se detalla. Sí puede verse que este otro diseño requiere una puerta cuántica menos. Por sencillez, en este trabajo se utilizará el diseño disponible en la plataforma de IBM.

De igual modo que ocurre con la puerta Toffoli, la implementación utilizada por la plataforma de IBM para la puerta Swap controlada es la mostrada en la Figura 2.18. Simplemente utiliza una puerta CNOT, una Toffoli, y otra CNOT. La puerta Toffoli está implementada conforme lo descrito en la Figura 2.17. En este trabajo se utilizará esta versión de la puerta SWAP controlada, que en este caso sí coincide con la descrita en la

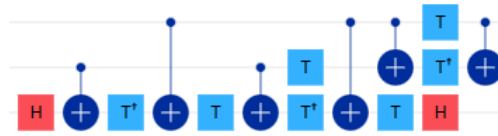


Figura 2.17: Implementación de la puerta Toffoli que ofrece la plataforma IBM Quantum. Esta es el diseño que se utilizará si utilizamos la puerta Toffoli incluida en dicha plataforma.

Sección anterior, pero con la diferencia de implementaciones para la puerta Toffoli.



Figura 2.18: Implementación de la puerta Toffoli que ofrece la plataforma IBM Quantum. Esta es el diseño que se utilizará si utilizamos la puerta Toffoli incluida en dicha plataforma.

## 2.4. Métricas

A la hora de medir un circuito clásico, es común describirlo en términos tales como su coste de fabricación, su tiempo de ejecución, su consumo energético, o incluso el espacio que ocupa (Hasan et al., 2019; Soumya et al., 2019; Zhuang y Wu, 1992). De esta forma, si tenemos dos circuitos diferentes que realizan la misma función, resulta sencillo compararlos si previamente los hemos medido utilizando estas métricas (Harris y Harris, 2015). Si uno de los circuitos es más rápido que el otro, diremos que es mejor en tales términos. No obstante, el circuito más lento podría ser más barato, resultando más adecuado que el otro circuito en un contexto de ahorro económico. De hecho, es habitual encontrar diseños optimizados en una o varias métricas que a menudo sacrifican el resto de las métricas en favor de las prioritarias (Bruce et al., 2002; Han y Carlson, 1987; Naseri y Timarchi, 2018). Por ejemplo, los llamados sumadores de acarreo anticipado son sumadores especialmente rápidos, pero cuyo coste es bastante más elevado que el de otros tipos de sumadores (Harris y Harris, 2015). Precisamente tienen un coste mayor porque emplean más recursos para conseguir tal velocidad (Liu et al., 2019). Estos sumadores serán más valiosos que los sumadores más lentos cuando la velocidad sea un factor importante. Sin embargo, si la prioridad es reducir costes, estos sumadores no deberían ser la primera opción. El que un circuito sea mejor que otro dependerá pues de la utilidad que se le quiera dar, pero no

podemos establecer que una o varias métricas sean mejores que el resto.

En computación cuántica también existen métricas para medir la bondad de los circuitos en determinados aspectos (Mohammadi y Eshghi, 2009). Existen métricas para medir los circuitos desde el punto de vista físico y analógico (Paredes-Barato y Adams, 2014), y otras métricas más adecuadas para medirlos a nivel digital (Thapliyal y Ranganathan, 2010). Los dispositivos cuánticos actuales están contruidos utilizando diferentes tecnologías: superconductores (Houck et al., 2008), iones (Leibfried et al., 2003), y otras propuestas (Kok et al., 2007) compiten por ser las que consigan máquinas cuánticas más avanzadas (Jazaeri et al., 2019). Comparar circuitos centrados en diferentes tecnologías no resulta especialmente útil, puesto que lo normal es que cada uno de esos circuitos se centre en las prioridades de su tecnología. Por citar un ejemplo, en computación cuántica óptica (Kok et al., 2007) existen grandes dificultades para construir puertas controladas, por lo que se priorizan los diseños de circuitos que permitan minimizar su uso (Lemr et al., 2015). Este problema no ocurre con otras tecnologías, por lo que dicha métrica (el número de puertas controladas) solo resultará útil en un contexto de computación cuántica óptica.

Puesto que no existe una única tecnología cuántica, resulta útil proponer diseños digitales para los circuitos en lugar de centrarse en una tecnología, de forma que estos sirvan para cualquier dispositivo cuántico disponible. Por supuesto, construir circuitos adaptados a la tecnología a usar es fundamental, pero es una tarea que debería hacerse a partir de diseños digitales optimizados, de igual forma a como se hace en circuitos clásicos (Harris y Harris, 2015). En la literatura sobre circuitos cuánticos es frecuente este enfoque de diseño de circuitos digitales, habiendo abundancia de trabajos (Häner et al., 2020; Y. Kim et al., 2023; H. Li et al., 2020; H. Wang et al., 2022). En este nivel, existen varias métricas empleadas por estos trabajos para medir y comparar circuitos. Mohammadi y Eshghi definieron un marco de trabajo para medir circuitos cuánticos (Mohammadi y Eshghi, 2009), ampliamente utilizadas. El trabajo de Mohammadi y Eshghi permite, de forma sencilla, analizar y comparar circuitos cuánticos a un nivel digital, y es sido ampliamente utilizado en la literatura (Gaur et al., 2019; Majumdar y Sur-Kolay, 2020; Noorallahzadeh et al., 2021). En particular, este trabajo propone cuatro parámetros para medir y analizar la bondad de un circuito:

- Coste cuántico: El coste cuántico nos permite hacernos una idea aproximada de la complejidad de un circuito. El coste cuántico indica el número de puertas de uno o dos cúbits que componen el circuito. Existen algunos trabajos en la literatura (Cha-

krabarti y Sur-Kolay, 2008) que utilizan como métrica el número de puertas (de cualquier tamaño) que componen un circuito, pero esta comparación puede resultar engañosa. En caso de comparar dos circuitos  $A$  y  $B$ , el primero compuesto por una puerta Toffoli, y el segundo compuesto por tres puertas Pauli-X, comparar directamente su número de puertas nos llevaría a la errónea idea de que  $B$  es un circuito más complejo que  $A$ . Es por ello por lo que el coste cuántico identifica el número de puertas de uno y dos cúbits, y no las de mayor tamaño. En caso de una puerta de 3 o más cúbits, su coste cuántico vendrá dado por el número de puertas de uno y dos cúbits que la componen.

- Retraso: El retraso (delay, en inglés) es una métrica relacionada con la velocidad del circuito. La velocidad de un circuito cuántico está íntimamente relacionada con la tecnología utilizada y las diversas implementaciones tanto de los cúbits como de las puertas cuánticas. En lugar de medir tiempos, lo que hace el retraso es indicar el número de puertas de uno y dos cúbits que se ejecutan secuencialmente en el camino crítico del circuito. De forma similar, el retraso de una puerta multi-cúbit vendrá determinado por el número de puertas de uno y dos cúbits que deben ejecutarse secuencialmente. Se define la unidad delta ( $\Delta$ ) como unidad del retraso.
- Cúbits auxiliares: Mohammadi y Eshghi denominan cúbit auxiliar (ancilla qubit, en inglés) a todo cúbit que inicialmente tiene un valor constante. Dicho de otra manera, se considera cúbit auxiliar a todo cúbit que no se utiliza para introducir en el circuito los valores externos necesarios para que este pueda realizar su función. En el caso de este trabajo, que realiza la comparación entre dos números  $a$  y  $b$ , se considerará un cúbit auxiliar cualquier cúbit que inicialmente no se utilice para introducir los dígitos de  $a$  y de  $b$ . Los ejemplos más comunes de cúbits auxiliares son cúbits utilizados para realizar operaciones auxiliares o para contener la salida/s del circuito (Thapliyal y Ranganathan, 2010).
- Salidas basura: Se denomina salida basura a cualquier salida del circuito que cumpla estas dos condiciones: 1) no contiene un valor que forme parte de la solución del problema resuelto por el circuito (y que por tanto es útil para una operación posterior), y 2) no contiene el estado cuántico al que se inicializó al comienzo del circuito. Los cúbit que no se restauran a su valor original no pueden ser utilizados por circuitos posteriores al no conocerse su estado. Con los dispositivos cuánticos actuales tenien-

do recursos limitados, revertir estos cúbits a su estado original resulta fundamental para aprovechar correctamente los recursos disponibles y poder así construir aplicaciones mayores. Para revertir estos cúbits a su estado debe realizarse un proceso adecuado que mantenga la reversibilidad del circuito completo, como por el ejemplo el propuesto por Bennett, 1973. De forma simplificada, el método de Bennet consiste en aplicar un circuito inverso. Este circuito es exactamente igual que el anterior, pero todas las puertas cuánticas se aplican en el orden exactamente inverso. La primera puerta será la última, y viceversa. Aunque algunos dispositivos cuánticos ofrecen operaciones de reseteo que son inmediatas, como es el caso de los computadores de IBM, estas operaciones no son reversibles y pueden provocar errores en caso de que los circuitos se deban integrar dentro de otros.

A modo de ejemplo de estas métricas, se va a analizar el circuito mostrado en la Figura 2.19. Este circuito es la ya descrita puerta cuántica lógica-AND temporal, que establece un qubit  $C$  a  $|1\rangle$  sí y solo sí otros dos cúbits  $A$  y  $B$  están en el estado  $|1\rangle$  (Gidney, 2018). La Figura 2.19 es ligeramente diferente a la Figura 2.10 que se mostró para ilustrar su funcionamiento. Esto es debido a que la Figura 2.10 muestra no solo la puerta en sí, sino también los preparativos necesarios para que funcione. Ya se mencionó que para que esta puerta funcione correctamente, es necesario inicializar  $C$  en el estado  $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$ . Pero en la Figura 2.19, que es similar a como es descrita por su autor en la fuente original (Gidney, 2018), no se muestran dichos preparativos. Precisamente se remarca esto ahora para entender que es necesario conocer las puertas para conocer sus costes reales. Por lo demás, podemos determinar que el circuito tiene un cúbit auxiliar, puesto que el objetivo  $C$  siempre está preparado en el estado indicado. El resultado se almacena en el cúbit  $C$ , mientras que los otros dos cúbits  $A$  y  $B$  mantienen su valor original. Por lo tanto, no hay salidas basura. Determinar el coste cuántico de la puerta lógica-AND temporal resulta trivial una vez sabemos que tenemos que preparar el estado cuántico indicado para  $C$  utilizando una puerta H y una puerta T (ver Sección 2.2): el circuito consta de 11 puertas cuánticas de uno y dos cúbits, mas esas dos puertas, por lo que su coste cuántico será de 13. Finalmente, el retraso vendrá dado por el número de puertas que deben ejecutarse secuencialmente, dando un resultado de  $11\Delta$ . Para mayor claridad, en la Figura 2.19 se han identificado numéricamente los 9 intervalos de  $1\Delta$  cada uno. Habría que añadir las puertas H y T que se ejecutan de forma secuencial para preparar  $C$ , y así obtenemos el valor de  $11\Delta$ .

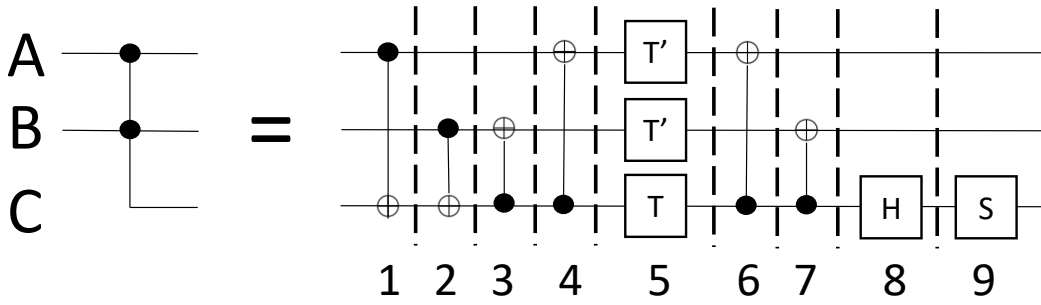


Figura 2.19: Símbolo e implementación de la puerta llamada lógica-AND temporal (Gidney, 2018). Para que funcione, el cúbit identificado como  $C$  debe inicializarse en el estado  $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}}|1\rangle)$  utilizando una puerta H y una puerta T (no mostradas en la imagen). Puede verse que consta de 11 puertas de uno o dos cúbits, que tiene un retraso de  $9\Delta$ , un cúbit auxiliar ( $C$ , puesto que es un valor de entrada constante), y que no tiene salidas basuras puesto que al finalizar  $C$  contiene el resultado, y  $A$  y  $B$  tienen sus valores originales.

Para una total claridad de cara al análisis de los circuitos, se muestra en la Tabla 2.3 el coste cuántico y retraso de todas las puertas cuánticas involucradas en este trabajo. Estos valores corresponden a su uso en los dispositivos cuánticos reales de IBM en los términos que se han descrito en esta Sección. Es importante aclarar que se considerará la misma implementación de las puertas para todos los circuitos analizados (pues las pruebas y medidas se realizarán en la plataforma de IBM) con independencia de la implementación escogida por los autores en los trabajos originales.

## 2.5. Comparadores en la literatura cuántica

Dada la importancia de la comparación, ya existen trabajos previos en los que se proponen circuitos cuánticos para realizarla (H. Li et al., 2020; Orts et al., 2021; Xia et al., 2018, 2019; Xia et al., 2020). En esta Sección se hace una revisión de los comparadores más recientes publicados en la literatura sobre computación cuántica. No se incluyen en esta revisión los comparadores completos, puesto que comparar los recursos de este tipo de comparadores con los utilizados por los medio comparadores no tendría sentido, al asumirse que los primeros necesitan más recursos al realizar más operaciones y ofrecer una funcionalidad ampliada. Por lo tanto, la revisión se centra solo en los medio comparadores.

En 2018, Xia et al. propusieron dos comparadores (Xia et al., 2018). El primero de ellos es un medio comparador, y el segundo un comparador completo, por lo que solo

Puerta	Coste cuántico	Retraso ( $\Delta$ )
Pauli-X	1	1
Pauli-Z	1	1
T	1	1
T'	1	1
S	1	1
CNOT	1	1
Swap	3	3
Lógica-AND temporal	13	11
Toffoli	15	11
Swap controlada	17	13

Tabla 2.3: Métricas del circuito propuesto, sin revertir las salidas basura, y con las salidas basura ya eliminadas. Se muestra el coste cuántico, el retraso, en número total de cúbits, y el número de salidas basura. Revertir las salidas basura tiene un coste alto en términos de coste cuántico y retraso, pero permite dejar libres  $2N - 1$  cúbits.

el primero es de interés para este trabajo. Para la construcción del comparador, Xia et al. construyeron dos subcircuitos llamados *CGC* y *ICGC*. El primero devuelve el acarreo generado al sumar dos dígitos, o tres si se le pasa un acarreo anterior (Figura 2.20), mientras que el segundo es igual que el primero pero en orden inverso (para revertir los cúbits a su estado original). Cada uno de estos circuitos consta de dos puertas Pauli-X, dos puertas CNOT, y una puerta Toffoli. Para construir el comparador, se utilizan  $N$  circuitos *CGC*, una puerta CNOT, y  $N$  circuitos *ICGC*. El circuito tiene dos cúbits auxiliares: el acarreo del primer circuito *CGC*, y un cúbit final para guardar el resultado. El comparador está libre de salidas basura. Se muestra un ejemplo completo del circuito, para el caso  $N = 4$ , en la Figura 2.21.

En 2019, los mismos autores del trabajo anterior publicaron un trabajo presentando varios comparadores, entre ellos un medio comparador de  $N$  bits (Xia et al., 2019). En este circuito, se comparan dos números  $A$  y  $B$  con igual cantidad de cifras binarias, empezando por los pares de dígitos menos significativos y terminando en los mas significativos. El circuito se construye mediante puertas Pauli-X, puertas CNOT, y puertas SWAP controladas, basándose en un diseño previo presentado en ese mismo trabajo de un comparador de





Figura 2.20: Puerta *CGC* propuesta por Xia et al. para generar el acarreo de dos bits  $A$  y  $B$  (Xia et al., 2018) y un posible acarreo de entrada.  $q[0]$  contiene el posible acarreo inicial, y al finalizar el circuito contendrá el resultado.  $q[1]$  y  $q[2]$  contendrán los dígitos a sumar. Al circuito inverso lo denominaron *ICGC*.

dos bits. Se muestra un ejemplo de este circuito, para el caso  $N = 4$ , en la Figura 2.22. Los autores afirman que el circuito necesita un único cúbit auxiliar, pero en realidad necesita dos:  $q[0]$  y  $q[9]$ . Ambos cúbits deben comenzar siempre en el estado  $|0\rangle$ , por lo que deben considerarse constantes y auxiliares. Para construir el circuito se necesitan  $2N$  puertas Pauli,  $2N + 1$  puertas CNOT, y  $2N$  puertas SWAP, así como  $2N + 2$  cúbits en total. El circuito revierte correctamente todos los cúbits, por lo que no presenta salidas basura.

En 2020, Li et al. presentaron un trabajo sobre operaciones aritméticas para computación cuántica y su aplicación en el tratamiento de imágenes (H. Li et al., 2020). Como parte de este trabajo, se definieron diversos circuitos de diseño propio para la suma, resta, y multiplicación. También se presentó un medio comparador, construido utilizando un sumador modular presentado en el mismo trabajo. Este sumador modular se implementa utilizando la puerta Peres (Peres, 1985), que a su vez se puede implementar mediante una puerta Toffoli y una puerta CNOT. Se reproduce el circuito de Li et al. en la Figura 2.23, para el caso de números de 4 dígitos. Por claridad, las puertas Peres se muestra explícitamente como una puerta Toffoli y una puerta CNOT. En realidad, hay un pequeño error de definición en el trabajo de Li et al.: la puerta Peres debe realizar la operación correspondiente a una puerta Toffoli seguida de una puerta CNOT sobre los dos cúbits de control de la puerta Toffoli. Es decir, que la CNOT no tiene efecto sobre la operación realizada por la puerta Toffoli, puesto que se aplica después. Sin embargo, para que el circuito de Li et al. funcione correctamente hay que aplicar primero la puerta CNOT y luego la puerta Toffoli (ver Figura 2.23), por lo que técnicamente esta puerta no es una puerta Peres. Omitiendo ahora la puerta Peres y centrando nuestra atención en el resto, el

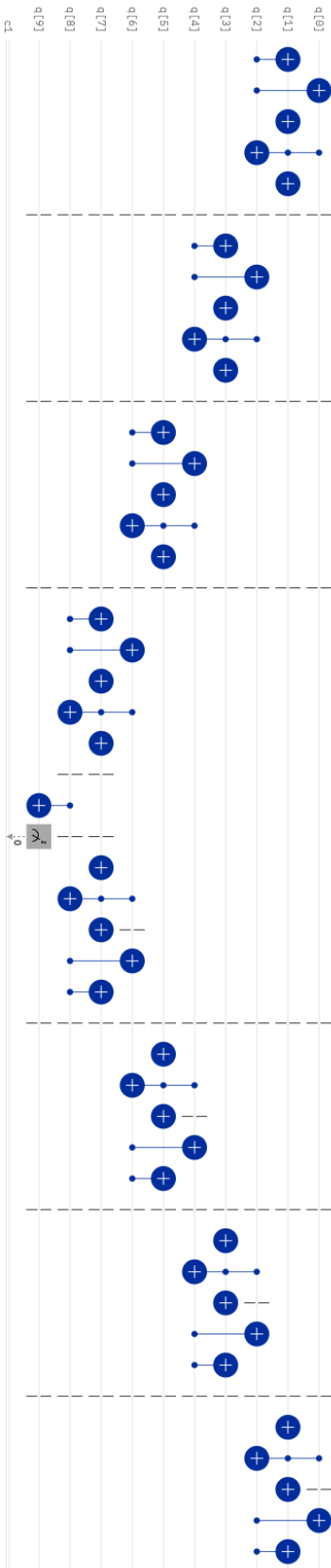


Figura 2.21: Comparador propuesto por Xia et al para la comparación de dos números  $A$  y  $B$ . Se muestra aquí un ejemplo para el caso  $N = 4$  dígitos. Puede apreciarse que se construye con subcircuitos  $CGC$  (Figura 2.20), una  $CNOT$ , y circuitos  $ICGC$  (el inverso de  $CGC$ ).  $q[0]$  y  $q[9]$  son cúbits auxiliares inicializados a  $|0\rangle$ . A partir de  $q[1]$ , cada par de cúbits representan (inicialmente) los dígitos de  $A$  y  $B$  ( $A_0, B_0, A_1, B_1, A_2, B_2, A_3, B_3$ ). Todos los cúbits se revierten a su valor original, por lo que no hay salidas basura. Se muestra un medidor para indicar dónde debe medirse el valor resultado.

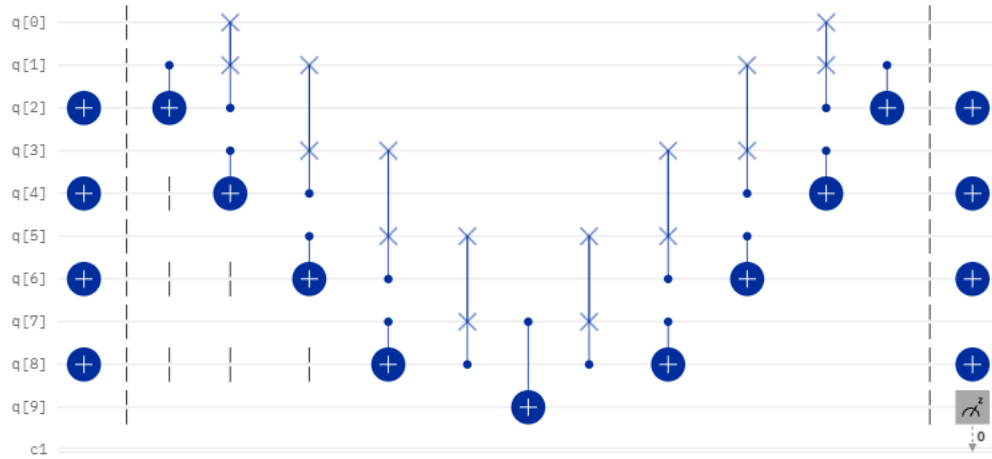


Figura 2.22: Circuito propuesto por Xia et al. para comparar dos números  $A$  y  $B$ , con  $N = 4$  (Xia et al., 2019). En el circuito,  $q[0]$  y  $q[9]$  son cúbits auxiliares. Los cúbits impares representan los dígitos de  $A$ , y los pares los dígitos de  $B$ . Se muestra un medidor en el cúbit que contiene el resultado tras la ejecución del circuito.

circuito constará de  $6N - 5$  puertas CNOT, y de  $2N - 1$  puertas Toffoli. Asimismo, tendrá una única entrada auxiliar (el cúbit superior, en la Figura 2.23), y ninguna salida basura.

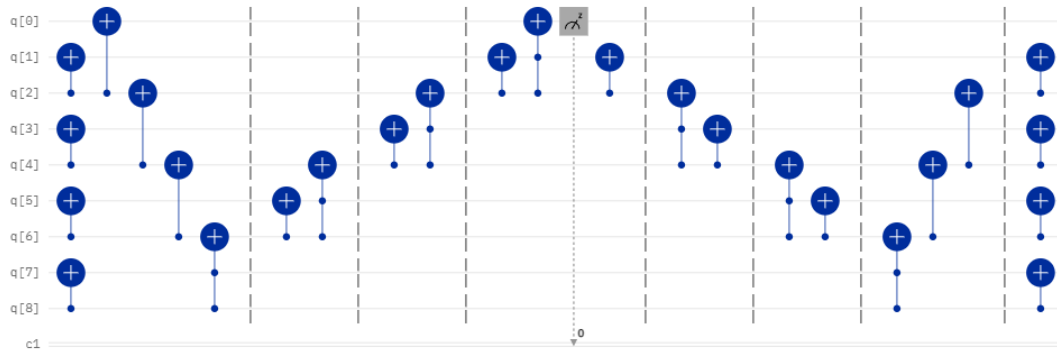


Figura 2.23: Circuito propuesto por Li et al., para el caso  $N = 4$  (H. Li et al., 2020). El cúbit  $q[0]$  debe estar inicializado a 0. Los cúbits impares contendrán los dígitos de  $B$  (desde  $q[1] = B_{N-1}$  hasta  $q[7] = B_{N-1}$ ). Por su parte, los cúbits pares contendrán los dígitos de  $A$  (desde  $q[2] = A_{N-1}$  hasta  $q[8] = A_{N-1}$ ). A diferencia de los anteriores comparadores, este comienza la comparación desde los dígitos más significativos en lugar de hacerlo desde lo menos significativos.

En el mismo año, Xia et al. propusieron un trabajo sobre comparadores y su aplicabi-

lidad en la binarización de imágenes en computación cuántica. Sin embargo, la aportación de este trabajo es un circuito binarizador (da a cada píxel de una imagen el valor blanco o negro en función de que superen o no un determinado valor umbral), siendo el circuito comparador utilizado el mismo que ya propusieron en 2019. En 2021, Orts et al. continuaron el trabajo de Xia et al. y su binarizador, pero aportando esta vez un medio comparador nuevo (Orts et al., 2021). El comparador de Orts et al. se centra en reducir el número de puertas T que el circuito necesita respecto a los comparadores anteriores. La puerta T tiene un coste superior al resto de puertas (Litinski, 2019), por lo que es frecuente en la literatura encontrar trabajos centrados en tratar de reducir el número de puertas T que posee un circuito (Gidney, 2018; Muñoz-Coreas y Thapliyal, 2018; Thapliyal et al., 2019). Reducir este número es muy útil en trabajos de bajo nivel que ofrecen circuitos pensando en los niveles inferiores de implementación. En la Figura 2.24 se muestra un ejemplo del circuito, para el caso  $N = 4$ . Por simplicidad, se utiliza la representación de la puerta Toffoli en la imagen para la operación *AND*. Sin embargo, este circuito no utiliza la puerta Toffoli, sino que utiliza la puerta lógica-AND temporal (Figura 2.19), que permite reducir el número de puertas T de 7 a 4 y, lo que es aún más importante, ser revertida sin necesitar para ello ninguna puerta T (la Toffoli necesitaría otras 7 puertas T). Para un tamaño  $N$ , el circuito necesitará  $2N$  puertas Pauli-X,  $6N - 1$  puertas CNOT,  $N$  puertas lógica-AND temporal, y  $N - 1$  puertas para revertir a la puerta lógica-AND temporal. Aunque en las métricas utilizadas en este trabajo no sea competitivo, en términos de número de puertas T es el mejor medio comparador actualmente disponible en la literatura según el análisis realizado en dicho trabajo (Orts et al., 2021). En su defecto, logra tal número a costa de sacrificar cúbits: necesita  $N$  cúbits auxiliares.

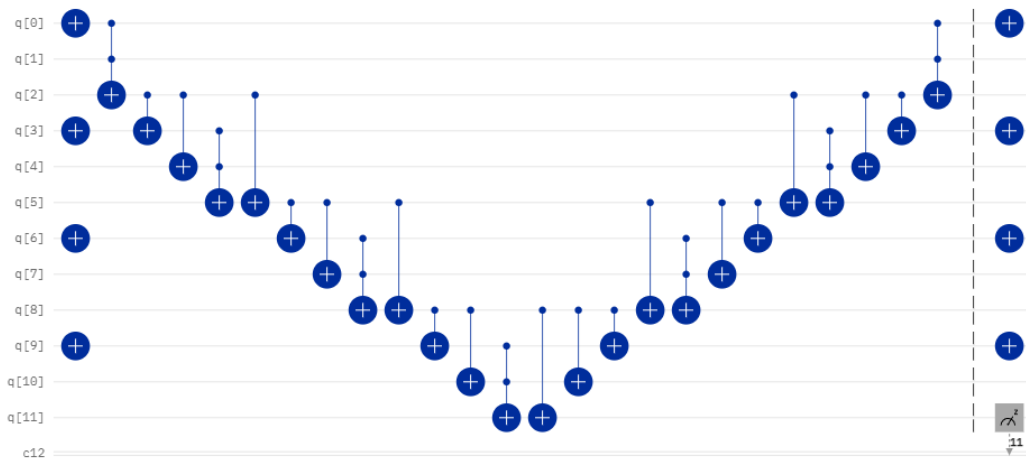


Figura 2.24: Circuito de 2020 propuesto por Orts et al., para el caso  $N = 4$  (Orts et al., 2021). Esta implementación está centrada en reducir el número de puertas T. Cabe destacar que aquí el símbolo habitual de la Toffoli no denota a esta puerta, sino a la lógica-AND temporal (Figura 2.19). Los cúbits van conteniendo sucesivamente pares de dígitos ( $q[0] = A_0, q[1] = B_1, \dots$ ). Sin embargo, los cúbit objetivo de todas las puertas lógica-AND temporal son auxiliares preparados en el estado  $\frac{1}{\sqrt{2}}(|0\rangle + e^{\frac{i\pi}{4}} |1\rangle)$ .

# 3. Objetivos

## 3.1. Objetivo principal

El objetivo principal de este trabajo es el diseño e implementación de un circuito comparador para computación cuántica que permita reducir el número necesario de cúbits respecto a los trabajos actualmente disponibles en la literatura sobre comparadores.

Se ha explicado en la Introducción (Capítulo 1) cómo los computadores cuánticos actuales disponen de pocos cúbits (Preskill, 2018). Por este motivo, es importante buscar formas de optimizar el número de cúbits de los circuitos. En el caso concreto de la comparación, se ha visto que el comparador más eficiente en estos términos es el de Li et al. (Figura 2.23). Este circuito necesita  $2N + 1$  cúbits para realizar la comparación de dos cadenas de  $N$  bits. Este número de cúbits,  $2N + 1$ , es el límite inferior teórico con el que puede hacerse un circuito comparador utilizando el método de codificación directo de los circuitos comparadores descritos en la Sección 2.5. Según este método de codificación, cada cúbit representa un bit (necesitándose por lo tanto  $2N$  cúbits para representar dos cadenas de  $N$  bits), no siendo posible realizar la comparación con un circuito que tenga  $2N$  cúbits o menos por motivos de reversibilidad (Nielsen y Chuang, 2011). Lo contrario implicaría que con  $2N$  cúbits (o menos) se puede representar una tabla de verdad que va a contener salidas totalmente idénticas para dos o más entradas diferentes, algo que no es posible en computación cuántica. Un ejemplo sencillo: Supónganse tres pares de cadenas:  $\{00, 11\}$ ,  $\{01, 11\}$  y  $\{10, 11\}$ . Un hipotético circuito destinado a comparar estas cadenas podría tener solo 2 cúbits, pues no se necesitan más cúbits para representar tales valores. El circuito debería devolver una salida (un solo cúbit, el otro lo podemos utilizar para intentar mantener la reversibilidad) que represente si la primera cadena es menor o igual que la segunda o no, valiendo dicha salida  $|0\rangle$  o  $|1\rangle$ , respectivamente. Puede verse a simple vista que resulta imposible: uno de los cúbits (la salida) siempre tiene que ser  $|0\rangle$ , puesto que en los tres casos del ejemplo la primera cadena es menor que la segunda. No podemos, con el otro cúbit, dar un valor diferente para los tres casos, por lo que no es posible mantener la reversibilidad con solo este número de cúbits. Sin embargo, en este trabajo se ha podido implementar la comparación con solo  $2N$  cúbits gracias al uso de una codificación alternativa explicada en el Capítulo siguiente, demostrándose que se puede superar el límite de  $2N + 1$  si se codifican eficientemente los datos.

### 3.2. Objetivos secundarios

Para lograr la consecución del objetivo marcado como principal, se hace necesario establecer una serie de objetivos secundarios.

1. Estudiar el funcionamiento de los diferentes tipos de circuitos comparadores en computación clásica, y definir claramente las entradas, salidas y pasos necesarios para realizar dicha comparación.
2. Analizar los circuitos comparadores para computación cuántica publicados en la literatura científica, verificando su correcto funcionamiento y analizando su metodología.
3. Identificar las puertas cuánticas más significativas y sus implementaciones más eficientes, así como todas las puertas cuánticas utilizadas en los circuitos estudiados.
4. Estudiar diferentes técnicas y formas de representación de la información en computación cuántica que puedan permitir reducir el número de cúbits necesarios para implementar un circuito comparador.
5. Reproducir los circuitos comparadores existentes en la literatura en la plataforma IBM Quantum.
6. Identificar y estudiar las técnicas más adecuadas para el análisis y medición de circuitos cuánticos, seleccionando un conjunto de ellas que resulte apropiado para la medición de circuitos comparadores.
7. Utilizar el conocimiento adquirido en los objetivos anteriores para diseñar e implementar un circuito de aportación propia que permita mejorar a los actualmente disponibles en los términos indicados por el objetivo principal.
8. Realizar una comparativa objetiva entre el circuito propuesto y los circuitos existentes para demostrar que se ha alcanzado el objetivo principal.

### 3.3. Metodología de trabajo

El desarrollo del presente trabajo se ha desarrollado en tres fases diferenciadas, que de forma resumida pueden identificarse como análisis del estado de la técnica, diseño e implementación y demostración de que se ha alcanzado el objetivo. Más en detalle, estas tres fases son:

1. Estudio de la literatura: la primera fase ha consistido en estudiar el conocimiento ya disponible en las materias relacionadas. Estas materias han sido, fundamentalmente, electrónica digital (clásica) y las bases necesarias de computación cuántica para poder interpretar circuitos cuánticos. La parte temporalmente más costosa de esta primera fase ha sido el estudio de los circuitos cuánticos comparadores ya disponibles. Relacionado con este último punto, ha estado el estudio de la plataforma IBM Quantum para poder implementar y probar tales circuitos, así como el uso de Github (Github, 2020) para clonar los circuitos y ofrecer prototipos de los mismos como aportación de este trabajo.

Objetivos cubiertos en esta fase:

- Estudiar el funcionamiento de los diferentes tipos de circuitos comparadores en computación clásica, y definir claramente las entradas, salidas y pasos necesarios para realizar dicha comparación.
- Analizar los circuitos comparadores para computación cuántica publicados en la literatura científica, verificando su correcto funcionamiento y analizando su metodología.
- Identificar las puertas cuánticas más significativas y sus implementaciones más eficientes, así como todas las puertas cuánticas utilizadas en los circuitos estudiados.
- Estudiar diferentes técnicas y formas de representación de la información en computación cuántica que puedan permitir reducir el número de cúbits necesarios para implementar un circuito comparador.
- Reproducir los circuitos comparadores existentes en la literatura en la plataforma IBM Quantum.
- Identificar y estudiar las técnicas más adecuadas para el análisis y medición de circuitos cuánticos, seleccionando un conjunto de ellas que resulte apropiado para la medición de circuitos comparadores.

2. Diseño e implementación: la segunda fase del trabajo se ha dedicado a utilizar el conocimiento adquirido en la fase anterior para diseñar e implementar un circuito cuántico comparador que mejore en términos de cúbits necesarios a los circuitos actualmente disponibles. Se han identificado las técnicas que se han considerado



más adecuadas de los paradigmas de computación clásico y cuántico para establecer el diseño. De forma similar a como se ha hecho con los circuitos estudiados, se ha implementado el circuito propuesto en la plataforma IBM Quantum y se ha subido un prototipo a Github<sup>1</sup>, que también se ofrece públicamente para que cualquier investigador interesado pueda probarlo rápidamente.

Objetivo cubierto en esta fase:

- Utilizar el conocimiento adquirido en los objetivos anteriores para diseñar e implementar un circuito de aportación propia que permita mejorar a los actualmente disponibles en los términos indicados por el objetivo principal.
3. Comparativa de circuitos: la tercera fase ha consistido en un análisis comparativo (utilizando las métricas estudiadas en la fase 1) entre el circuito propuesto y los circuitos disponible en la literatura (también estudiados en la fase 1). Se ha demostrado en este análisis que el circuito comparador propuesto ha alcanzado el objetivo principal, siendo el mejor de su categoría en términos de números de cúbits.

Objetivo cubierto en esta fase:

- Realizar una comparativa objetiva entre el circuito propuesto y los circuitos existentes para demostrar que se ha alcanzado el objetivo principal.

---

<sup>1</sup><https://github.com/2forts/QuantumComparator>

## 4. Desarrollo del trabajo

El comparador que se va a proponer se fundamenta en dos ideas principales. En primer lugar, se utiliza la metodología del comparador de Li et al (H. Li et al., 2020), que ya ha sido descrita en la Sección 2.5. En segundo lugar, se utiliza la idea propuesta por Pérez-Salinas et al. de introducir valores a un circuito, no como estados de los cúbits, sino como operaciones sobre ellos (Pérez-Salinas et al., 2020). Dicho de otra forma, en lugar de asignar valores iniciales a los cúbits como se hace en el resto de los comparadores estudiados en este trabajo, se utilizan puertas cuánticas a lo largo del circuito en función de estos valores. Es habitual en todo circuito cuántico inicializar el estado de los cúbits con puertas cuánticas, pero Pérez-Salinas et al. no se limitaron a eso, sino que reintroducen a lo largo del circuito esos valores. La ventaja principal es que se obtiene un ahorro de cúbits. Si se hace con cuidado, es incluso posible reducir el coste cuántico del circuito. En el caso de Pérez-Salinas et al., les permitió implementar un clasificador cuántico con un solo cúbit. Además, Pérez-Salinas et al. justifican que esta reintroducción de datos en circuitos cuánticos es una forma natural en computación cuántica de compensar la imposibilidad de copiar valores que sí existe en computación clásica.

Al igual que los comparadores anteriores, el circuito que se va a proponer comparará dos cadenas de bits  $A$  y  $B$ . Los dígitos de  $B$ , esto es,  $b_{N-1}...b_0$ , se introducen de forma similar a lo explicado en los circuitos anteriores. Es decir, se necesitarán  $N$  cúbits para representar  $B$ . Sin embargo, para introducir  $A$  se utiliza la metodología de Pérez-Salinas et al. En lugar de utilizar cúbits inicializados con los dígitos  $a_{N-1}...a_0$ , se aplicarán tales dígitos mediante puertas cuánticas. Se muestra en la Figura 4.1 un ejemplo sencillo de cómo funciona esta idea, antes de explicar el circuito propuesto. Supongamos dos bits  $a_0$  y  $b_0$ . Si se quiere realizar la operación  $a_0 \oplus b_0$ , el proceder habitual en un circuito cuántico es el mostrado en la Figura 4.1(a). Sin embargo, es posible realizar esta misma operación utilizando un único cúbit, tal y como se muestra en la Figura 4.1(b). Aquí, la puerta roja etiquetada como  $a_0$  será una puerta Pauli-X si  $a_0 = 1$ , o bien una puerta identidad (o simplemente una ausencia de puerta) en caso contrario. El porqué se deben utilizar estas puertas y no otras se explica justo a continuación. Por lo demás, en este sencillo ejemplo ya se ponen de manifiesto las ventajas y desventajas de esta metodología (que se analizarán con detalle en el siguiente Capítulo): se ha reducido el número de cúbits, pero será necesario construir un circuito diferente cada vez que cambie  $a_0$ .

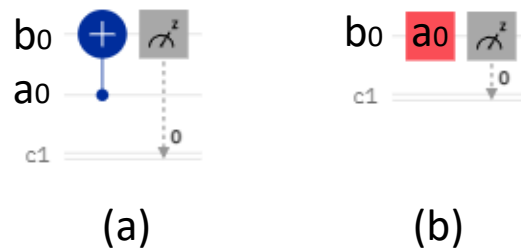


Figura 4.1: (a) Operación  $a_0 \oplus b_0$  utilizando la puerta CNOT. (b) La misma operación, pero utilizando una puerta de un solo cúbit para representar a  $a_0$ . Esta puerta será una puerta identidad si  $a_0 = 0$ , o una puerta Pauli-X en caso contrario.

En el circuito que se va a proponer, al igual que en el resto de comparadores, se trabaja con las bases  $\{|0\rangle, |1\rangle\}$  que el usuario del circuito interpreta como los dígitos 0 y 1 respectivamente. Un vistazo rápido al circuito de Li et al. (Figura 2.23) permite constatar que dicho circuito está compuesto únicamente de puertas CNOT y Toffolis. Explicado de manera simplificada, tales puertas suponen que se aplique una rotación de la matriz de Pauli (sobre X) cuando uno o dos cúbits de control (para la CNOT y la Toffoli, respectivamente) están en el estado  $|1\rangle$ , que es el estado que tendrá cada cúbit cuando el dígito al que representa está puesto a 1. En el ejemplo mostrado en la Figura 4.1 se ha indicado que se debe utilizar una puerta Pauli-X si  $a_0 = 1$  precisamente por este motivo. Y en caso contrario, no se debe utilizar ninguna puerta o, por simple motivo de claridad, utilizar una puerta identidad.

Por supuesto, el ejemplo de la Figura 4.1 muestra el caso más sencillo. Pero otras múltiples operaciones pueden simplificarse de esta manera. En este trabajo, además de la mencionada en el ejemplo, se utilizan principalmente estas otras:

- Supongase que se necesita el valor  $a_0 \oplus b_0$  de la Figura 4.1(b) para, a su vez, actuar como cúbit de control sobre otro cúbit  $C$ . Una vez realiza esta operación, podemos volver a aplicar la misma puerta  $a_0$  (una Pauli-X si  $a_0 = 1$ , o una identidad en caso contrario) para revertir el estado actual del cúbit y que vuelva a contener a  $b_0$  si volvemos a necesitar el uso de este valor.
- Se puede reutilizar varias veces un cúbit auxiliar (ver definición en la Sección 2.4) para guardar temporalmente diferentes valores. Por ejemplo, si en un momento dado se necesita utilizar una puerta Toffoli controlada por  $b_0$  y  $a_0$ , y posteriormente se quiere aplicar una segunda puerta Toffoli esta vez controlada por  $b_0$  y  $a_1$ , podemos

introducir  $a_0$  mediante puerta en el cúbit auxiliar, aplicar la puerta Toffoli, revertir el valor del cúbit auxiliar mediante otra puerta  $a_0$ , y finalmente introducir  $a_1$  mediante puerta para poder aplicar la segunda Toffoli.

- Dos puertas  $a_i$  y  $a_j$  consecutivas en un mismo cúbit auxiliar realizan la operación  $a_i \oplus a_j$ .

Con estas cuatro sencillas ideas, se puede simplificar el circuito de Li et al. de forma que se reduzca en uno el número de cúbits necesarios, **logrando así el único comparador en la literatura sobre circuitos cuánticos que solo necesita  $2N$  cúbits.**

## 4.1. Algoritmo para la construcción del comparador

Para construir el circuito para cualquier tamaño de dígito  $N$ , basta con seguir estos pasos:

1. Preparar  $2N$  cúbits. De ellos,  $N$  se utilizarán para codificar los dígitos de  $B$  ( $b_{N-1} \dots b_0$ ), y el resto se inicializarán a  $|0\rangle$  y serán utilizados como cúbits auxiliares. Los cúbits que contienen los dígitos de  $B$  estarán en el estado  $|0\rangle$  si el dígito al que representa es 0, y en el estado  $|1\rangle$  en caso contrario. No serán necesarios más cúbits para implementar el comparador.
2. Aplicar una puerta Pauli-X en el primer cúbit auxiliar si  $a_0 = 1$ , o una puerta identidad si  $a_0 = 0$ . A partir de ahora, a este tipo de operación en la que aplicamos una puerta Pauli-X si  $a_i = 1$  o una puerta identidad si  $a_i = 0$  la etiquetaremos como puerta  $a_i$ . También se aplicará una puerta  $a_1$  sobre el segundo cúbit auxiliar, y una puerta  $a_j$  sobre cada uno de los cúbits utilizados para codificar  $B$  a excepción de  $b_0$  (es decir, desde  $j = N - 1$  hasta 1). A continuación, se aplica una puerta CNOT sobre el primer cúbit auxiliar y  $b_0$  para realizar la operación  $a_0 \oplus b_0$ , y una puerta Toffoli sobre  $b_0$  y los dos cúbits auxiliares utilizados hasta ahora de forma que se realice la operación  $(a_0 \oplus b_0)a_0 \oplus a_1$ . Finalmente, se aplica una puerta  $a_0$  sobre el primer cúbit auxiliar para revertirlo y que quede libre.
3. Sobre el cúbit auxiliar que hemos liberado anteriormente, aplicar una puerta  $a_1$  y una puerta  $a_2$ , de forma que se implemente la operación  $a_1 \oplus a_2$ . Aplicar una puerta CNOT cuyo cúbit de control es el segundo cúbit auxiliar y cuyo objetivo es  $b_1$ , y

finalmente una puerta Toffoli cuyos cúbits de control son los cúbits involucrados en la CNOT, y cuyo objetivo es el cúbit que contiene  $a_1 \oplus a_2$ .

4. Desde  $i = 2$  hasta  $i = N - 2$ , repetir:
  - Sobre un cúbit auxiliar aún no utilizado anteriormente, realizar la operación  $a_{i+1} \oplus a_i$  aplicando dos puertas  $a_{i+1}$  y  $a_i$ . Aplicar una puerta CNOT, siendo su cúbit de control el cúbit objetivo de la última puerta Toffoli aplicada, y cuyo cúbit objetivo (de la CNOT) es  $b_i$ . Finalmente, aplicar una puerta Toffoli cuyos cúbits de control son los dos cúbits involucrados en la CNOT previa, y cuyo cúbit objetivo es el cúbit auxiliar que contiene la operación  $a_{i+1} \oplus a_i$ .
5. En el último paso para construir el circuito se debe aplicar una puerta CNOT, siendo su cúbit de control el cúbit objetivo de la última puerta Toffoli aplicada, y cuyo cúbit objetivo (de la CNOT) es  $b_{N-1}$ . Posteriormente, se aplica una puerta Toffoli cuyos cúbits de control son los dos cúbits involucrados en la CNOT previa, y cuyo cúbit objetivo es el único cúbit auxiliar que aún no se ha utilizado. Sobre este cúbit, se aplica la operación  $a_{N-1}$ . Este cúbit contendrá el resultado del comparador.

En la Figura 4.2 se muestra un ejemplo del circuito propuesto para el caso  $N = 4$ . Tal y como se ha explicado, se necesitan  $2N = 8$  cúbits para su implementación. Puede verse cómo los dígitos de  $B$  se introducen en los cúbits (marcados como  $b_3...b_1$ ), mientras que los dígitos de  $A$  se introducen en forma de puertas cuánticas de color rojo. Las puertas rojas con un número  $i$  hacen referencia a las diferentes operaciones  $a_i$ . Asimismo, por motivos de claridad cada etapa se muestra separada de las demás por una barrera que cubre todos los cúbits. Las barreras pequeñas que no cubren todos los cúbits se utilizan para mantener el orden visual explicado en el algoritmo de construcción del circuito.

Sin embargo, el circuito en su estado actual presenta salidas basura que deben revertirse para liberar esos cúbits. Para revertir dichas salidas basura basta con aplicar el esquema de Bennett (Bennett, 1973) al circuito anterior como ya se dijo en la Sección 2.4, respetando el cúbit que contiene el resultado. En la Figura 4.3 se puede ver un ejemplo del circuito necesario para revertir las salidas basura del circuito propuesto, para el caso  $N = 4$ . Puede verse como en este segundo circuito se respeta en todo momento el cúbit que contiene el resultado, mientras que el resto de cúbits es devuelto a su estado inicial y son libres de poder ser utilizados por otros circuitos cuánticos. Puede verse en la Figura 4.3 como,

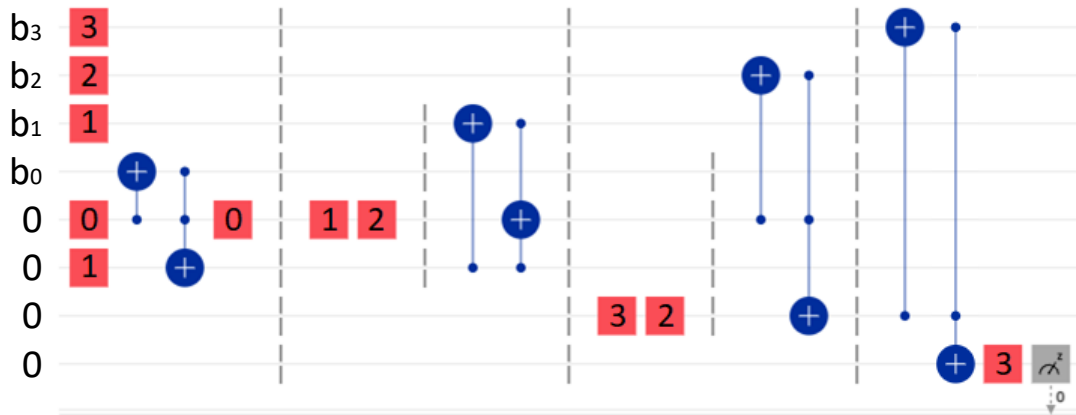


Figura 4.2: Circuito propuesto, para el caso  $N=4$ . Puede verse a simple vista que reduce el número de cúbits necesarios respecto a los comparadores estudiados. Los dígitos de  $B$  se representan utilizando los cúbits marcados como  $b_i$ . Por su parte, los dígitos de  $A$  se introducen mediante puertas cuánticas (indicadas en color rojo) en la propia implementación del circuito. El número de cada puerta roja indica el índice del dígito de  $A$ . Por ejemplo, un 3 hace referencia a  $a_3$ . Esta puerta será una Pauli-X si dicho dígito es un 1, o una puerta identidad (o una ausencia de puerta) en caso contrario.

efectivamente, los cúbits terminan conteniendo el estado  $|0\rangle$  en el caso de los auxiliares, y el estado asociado a cada dígito de  $B$  para el resto.

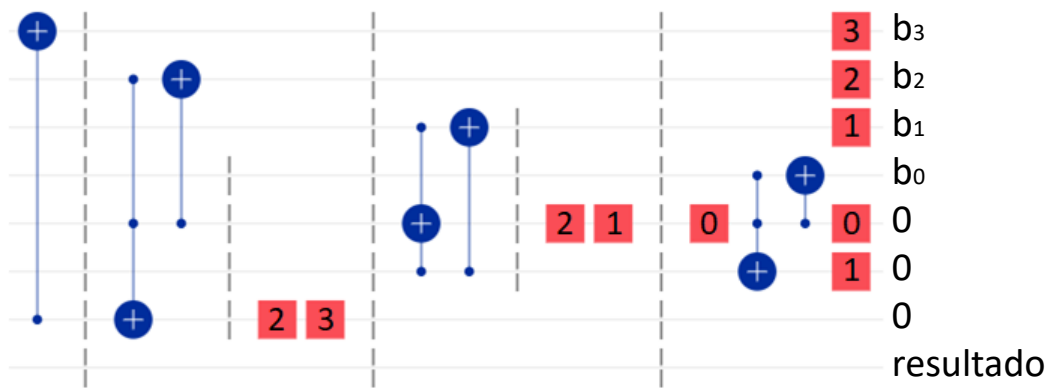


Figura 4.3: Inversa del circuito propuesto para revertir todos los cúbits (excepto el que contiene el resultado) a su estado original y evitar salidas basura.

# 5. Análisis y comparativa

En la primera Sección de este Capítulo se realiza la medición del circuito propuesto utilizando las métricas descritas en la Sección 2.4. En la segunda Sección se realiza una comparativa entre el circuito propuesto y los disponibles en la literatura utilizando dichas métricas.

## 5.1. Análisis del circuito propuesto

Para analizar el número de cúbits que necesita el circuito propuesto, así como su coste cuántico, retraso, y presencia (o no) de salidas basura, se va a analizar paso a paso el algoritmo descrito en el Capítulo anterior para la implementación de dicho circuito:

- En el paso 1 se indica que se necesitan  $2N$  cúbits, de los cuales  $N$  se utilizan para codificar  $B$ , y los otros  $N$  tienen fines auxiliares. Por lo tanto, el circuito propuesto necesita  $2N$  cúbits, de los cuales  $N$  son cúbits auxiliares.
- En el paso 2 se utilizan dos puertas Pauli-X o dos puertas identidad sobre dos cúbits auxiliares (el coste cuántico de cada una de estas es 1, y su retraso es  $1\Delta$ ),  $N - 1$  puertas Pauli-X o identidad sobre los cúbits de  $B$ , una puerta CNOT (coste cuántico 1, retraso  $1\Delta$ ), una puerta Toffoli (coste cuántico 15, retraso  $11\Delta$ ), y una puerta Pauli-X o identidad (coste cuántico 1, retraso  $1\Delta$ ). El coste cuántico de este paso es  $1 + 1 + (N - 1) + 1 + 15 + 1 = N + 18$ . Por su parte, el retraso es  $1\Delta + 1\Delta + 11\Delta + 1\Delta = 14\Delta$  (todas las puertas  $a_i$ , salvo la última, pueden computarse en paralelo).
- En el paso 3 se utilizan dos puertas Pauli-X o identidad, una puerta CNOT, y una puerta Toffoli. El coste cuántico es  $1 + 1 + 1 + 15 = 18$ , y el retraso es  $1\Delta + 1\Delta + 1\Delta + 11\Delta = 14\Delta$ .
- En el paso 4 se realizan  $N - 3$  iteraciones. Cada una de estas iteraciones involucra dos puertas Pauli-X o identidad, una puerta CNOT, y una puerta Toffoli. El coste cuántico por iteración es  $1 + 1 + 1 + 15 = 18$ . El retraso por iteración es  $1\Delta + 1\Delta + 11\Delta + 1\Delta = 14\Delta$ . Teniendo en cuenta las  $N - 3$  iteraciones, el coste cuántico será de  $18N - 54$ , y el retraso de  $(14N - 42)\Delta$ .

- En el paso 5 se aplica una puerta CNOT, una puerta Toffoli, y una puerta PauliX o identidad. El coste cuántico es  $1 + 15 + 1 = 17$ , y el retraso es  $1\Delta + 11\Delta + 1\Delta = 13\Delta$ .

El coste cuántico del circuito en este punto es de  $N + 18 + 18 + 18N - 54 + 17 = 19N - 1$ . Por su parte, el retraso es de  $14 + 14 + 14N - 42 + 13 = (14N - 1)\Delta$ . Sin embargo, aún no se han revertido las salidas basura y el circuito tiene  $2N - 1$  salidas basura. Para eliminar las salidas basura se debe ejecutar el circuito inverso, que supone:

- Invertir el paso 5 solo requiere una puerta CNOT. Supone un coste cuántico y retraso de 1 y  $1\Delta$ , respectivamente.
- El paso 4 debe repetirse entero, pero a la inversa. Se añade pues un coste cuántico de  $18N - 54$  y un retraso extra de  $(14N - 42)\Delta$ .
- El paso 3 también debe ejecutarse a la inversa. Supone un coste cuántico de 18, y un retraso de  $14\Delta$ .
- El paso 2, de forma inversa, añade un coste cuántico extra de  $N + 18$ , y un retraso de  $14\Delta$ .
- No se requiere ninguna acción para revertir el paso 1. Por lo tanto, no se añaden costes extras en este paso.

El coste cuántico del circuito inverso es de  $1 + 18N - 54 + 18 + N + 18 = 19N - 17$ . Su retraso es de  $1 + 14N - 42 + 14 + 14 = 14N - 13$ . Estos valores deben sumarse a los anteriores, obteniéndose un coste cuántico total de  $38N - 20$ , y un retraso total de  $(28N - 14)\Delta$ . El circuito no tiene ninguna salida basura, y se han necesitado  $2N$  cúbits, de los cuales  $N$  son auxiliares. Para una mejor visualización, se muestran las métricas del circuito sin revertir las salidas basuras y de la versión final (con todos los cúbits ya revertidos) en la Tabla 5.1.

El principal logro del circuito propuesto es haber conseguido un circuito cuántico que realice la comparación de  $2N$  bits con solamente  $2N$  cúbits. Según lo expuesto en la Sección 3.1, no es posible implementar una comparación con menos de  $2N + 1$  cúbits si se representa cada bit de las cadenas utilizando un cúbit. Sin embargo, el uso de puertas cuánticas para codificar los bits en lugar de utilizar cúbits ha permitido vencer dicho límite.



Circuito	Coste cuántico	Retraso ( $\Delta$ )	Cúbits	Salidas basura
Con salidas basura	$19N - 1$	$14N - 1$	$2N$	$2N - 1$
Sin salidas basura	$38N - 20$	$28N - 14$	$2N$	0

Tabla 5.1: Métricas del circuito propuesto, sin revertir las salidas basura, y con las salidas basura ya eliminadas. Se muestra el coste cuántico, el retraso, en número total de cúbits, y el número de salidas basura. Revertir las salidas basura tiene un coste alto en términos de coste cuántico y retraso, pero permite dejar libres  $2N - 1$  cúbits.

## 5.2. Comparativa

Para poder realizar una comparativa que resulte útil, se han medido el resto de circuitos descritos en la Sección 2.5 utilizando las mismas métricas que con el circuito propuesto. Las métricas para todos los circuitos, incluido el propuesto, se muestran en la Tabla 5.2. Aunque en las métricas hemos hablado inicialmente de cúbits auxiliares, en su lugar se muestra aquí el número de cúbits totales del circuito. El motivo es sencillo: los circuitos de la literatura utilizan una metodología en la que indicando el número de cúbits auxiliares permite calcular de forma trivial el número de cúbits totales, pero esto no ocurre con el circuito propuesto, y puede dar lugar a confusión. Si, por ejemplo, indicamos que el circuito de Orts et al. tiene  $N$  cúbits auxiliares y el circuito propuesto en este trabajo tiene  $N - 1$  cúbits auxiliares, parecería que el propuesto solo mejora al otro en un cúbit, cuando la realidad es que lo mejora en  $N$  cúbits. Por lo demás, se han considerado, para cada puerta cuántica, los mismos valores de coste cuántico y retraso (indicados en la Sección 2.4).

Circuito	Coste cuántico	Retraso ( $\Delta$ )	Cúbits	Salidas basura
Xia et al., 2018	$38N + 1$	$26N$	$2N + 2$	0
Xia et al., 2019	$39N + 1$	$28N + 3$	$2N + 2$	0
H. Li et al., 2020	$36N - 20$	$26N - 14$	$2N + 1$	0
Orts et al., 2021	$32N - 16$	$28N - 25$	$3N$	0
Propuesta	$38N - 20$	$28N - 14$	$2N$	0

Tabla 5.2: Comparativa en términos de coste cuántico, retraso, número de cúbits, y número de salidas basura, entre el circuito propuesto y los circuitos revisados en la Sección 2.5.

En términos de coste cuántico, el mejor circuito es el de Orts. et al., con un coste

cuántico de solo  $32N - 16$ . El uso de la puerta logica-AND temporal en sustitución de la puerta Toffoli le permite un ahorro importante de coste cuántico, hasta el punto de que es el mejor circuito en tales términos. Sin embargo, como consecuencia del uso de dicha puerta se incrementa el número de cúbits que necesita (cada puerta logica-AND temporal supone añadir un cúbit auxiliar). En contraste, el peor circuito en términos de coste cuántico es el de Xia et al. de 2019. El circuito propuesto tiene un coste cuántico de  $38N - 20$ . Este coste superior en  $6N$  al coste del circuito de Orts et al., pero necesita  $N$  cúbits menos.

En términos del retraso, el mejor circuito (en el caso general) es el de Li et al. con un valor de  $26N - 14\Delta$ , seguido del primer circuito de Xia et al. ( $26N\Delta$ ) y del circuito de Orts et al. ( $28N - 25$ ). De hecho, el circuito de Orts et al. mejora al circuito de Li et al. para valores de  $N$  menores que 6. En el lado contrario, el circuito más lento es el segundo de Xia et al., con un delay de  $28N + 3\Delta$ . Por su parte, el circuito propuesto tiene un retraso de  $28N - 14\Delta$ . Una diferencia, respecto al más rápido, de  $2N\Delta$ .

En términos de número de cúbits es donde más destaca el circuito propuesto: solamente necesita  $2N$  cúbits para implementar una comparación entre cadenas de  $N$  bits. Es el único comparador actualmente disponible en la literatura cuántica que tiene este número de cúbits. Conseguir tal número de cúbits no era sencillo, como hemos demostrado en la Sección anterior, y ha sido posible gracias al uso de una metodología de codificación alternativa. El segundo mejor circuito es el de Li et al, que necesita  $2N + 1$  cúbits. El peor circuito en términos de cúbits es el de Orts et al., que ya se ha visto como sacrifica cúbits para reducir el valor de otras métricas.

La última columna de la Tabla 5.2 muestra que todos los circuitos carecen de salidas basura, por lo que están optimizados en tales términos y permiten ser fácilmente combinados con otros circuitos cuánticos.

Destacando los circuitos mejores en cada métrica, obtenemos:

- Mejor coste cuántico:  $32N - 16$  (Orts et al., 2021)
- Mejor retraso:  $26N - 14\Delta$  (H. Li et al., 2020) y  $28N - 25$  (Orts et al., 2021). El primero es mejor para  $N > 5$ , el segundo es mejor para  $N \leq 5$ .
- Mejor número de cúbits:  $2N$  (circuito propuesto).
- Mejor número de salidas basura: 0 (todos los circuitos).

Un resultado interesante es que no hay un circuito mejor en términos absolutos, sino que cada métrica tiene su propio ganador. Por lo tanto, la elección de qué comparador debe utilizarse dependerá de la métrica o métricas que se desean optimizar. Si el objetivo es reducir el coste cuántico y hay cúbits suficientes, la mejor opción es el circuito de Orts et al. Si se busca un menor retraso, se debería elegir el circuito de Lit et al. para cadenas de más de 5 bits, o bien el circuito de Orts et al. para las cadenas más pequeñas de 5 o menos cúbits. En caso de que la prioridad sea el número de cúbit, que es el recurso más escaso en los computadores cuánticos actuales (Preskill, 2018), el circuito más eficiente es el comparador propuesto en este trabajo. En términos de salidas basura, todos los circuitos aquí analizados son óptimos, por lo que se puede elegir cualquier otra métrica como prioritaria sabiendo que, al final, todos los circuitos devolverán el resultado en un solo cúbit dejando el resto en su estado inicial.

## 6. Conclusiones

En este trabajo se ha propuesto un circuito cuántico para realizar la comparación entre dos cadenas de bits de una determinada longitud. El circuito propuesto mejora, en número de cúbits necesarios para su implementación, a los comparadores actualmente publicados en la literatura científica cuántica. Para alcanzar este objetivo, se ha utilizado el conocimiento adquirido durante el Máster en Computación Cuántica para establecer los conceptos necesarios que permitiesen implementar un circuito cuántico, incluyendo entre otros la codificación de la información (asignaturas de Información Cuántica y Matemáticas de la Información), el uso de puertas cuánticas (asignatura de Computación Cuántica), y conceptos generales tanto de álgebra lineal como de mecánica cuántica (en las asignaturas homónimas del Máster).

Todos los circuitos estudiados en este trabajo, incluido por supuesto el circuito propuesto, han sido implementados en la plataforma IBM Quantum (cuya enseñanza de uso se cubre en las asignaturas de Computación Cuántica y de Algoritmos Cuánticos del Máster) para comprobar su correcto funcionamiento así como la veracidad de las métricas. De hecho, en uno de los circuitos estudiados se han encontrado errores de definición (que no de funcionamiento) tal y como se comenta en la Sección correspondiente al estudio de tales circuitos. Como complemento al trabajo, se ofrece una replica de cada uno de los circuitos estudiados (y del propuesto) para que cualquier persona interesada pueda probarlos directamente. Tales circuitos se encuentran en el siguiente repositorio: <https://github.com/2forts/QuantumComparator>.

Previamente a la implementación de los circuitos publicados, se han estudiado las puertas cuánticas necesarias para su implementación. Se han identificado tales puertas (incluyendo las puertas necesarias para la construcción de las puertas más grandes, cuando no se trataba de puertas básicas o disponibles de forma directa en la plataforma utilizada. Todas las puertas estudiadas pertenecen al grupo Clifford+T (o se pueden implementar mediante puertas de dicho grupo). El propio circuito propuesto en este trabajo está implementado exclusivamente utilizando puertas del grupo Clifford+T, lo que lo hace compatible con los códigos de detección y corrección de errores estudiados en la asignatura de Información Cuántica.

Para realizar una medición robusta del circuito propuesto, así como una comparativa seria y útil entre el circuito propuesto y el resto de comparadores para computación

cuántica, se ha buscado y elegido un conjunto de métricas ampliamente utilizadas en la literatura. En tales términos, se han analizado todos los circuitos recogidos en este trabajo y se ha mostrado la información resultante en forma de tabla para que cualquier investigador o investigadora interesado en la materia pueda, de forma cómoda y rápida, elegir el circuito que más útil le sea para sus intereses.

El circuito propuesto no se ha definido para un tamaño fijo de cadenas a comparar, sino que es fácilmente personalizable a cadenas de cualquier tamaño. Se ha proporcionado un algoritmo que permite su rápida reconstrucción para cualquier tamaño incluso aunque no se entienda el funcionamiento del circuito, y se ha utilizado dicho algoritmo para demostrar la veracidad de las métricas asignadas en el análisis. El circuito es totalmente reversible de acuerdo a las reglas de la computación cuántica, y además carece de salidas basura al igual que todos sus competidores.

Aunque la mejora en términos de cúbits pueda parecer pequeña (estamos hablando de un solo cúbit frente al segundo mejor circuito en tales términos), se ha conseguido romper la barrera de los  $2N + 1$  cúbits necesarios para realizar la comparación entre dos cadenas de  $N$  bits impuesta por la necesidad de mantener una computación reversible que no pierda nunca información. Para ello se ha recurrido a una forma alternativa de codificar la información de manera más eficiente que la utilizada por los comparadores cuánticos actualmente disponibles. En lugar de utilizar un cúbit para representar cada bit, algunos de estos bits se han introducido al circuito mediante puertas cuánticas. Este tipo de computación abre la puerta a posibles implementaciones que podrían conseguir reducir aún más el número necesario de cúbits no solo en comparadores, sino en todo tipo de circuitos aritméticos para computación cuántica. Posiblemente la conclusión más grande que se puede sacar de este trabajo es esta posibilidad de utilizar codificaciones alternativas en circuitos en los que resulta totalmente contraintuitivo, como es el caso de las operaciones aritméticas en las que la humanidad lleva siglos utilizando las codificaciones habituales.

Una conclusión interesante que se puede sacar de este trabajo es ver cómo la computación cuántica es capaz de utilizar sus propias herramientas para implementar cualquier función lógica, en este caso la comparación de bits. En los circuitos, se ha hablado de bits, pero en realidad se han manejado estados  $|0\rangle$  y  $|1\rangle$  para representar tales bits. Se han tenido que buscar alternativas a la copia clásica de valores, concretamente utilizando entrelazamiento y cúbits auxiliares. Se han realizado operaciones clásicas como la AND o la OR mediante puertas cuánticas y sus rotaciones en los cúbits. Incluso se han introducido

valores clásicos (en realidad, rotaciones que interpretamos como tales valores) a través de puertas cuánticas. Y aunque estos circuitos no utilizan la superposición en este trabajo, la pueden soportar perfectamente de forma que puedan, de una sola iteración, computar la comparación entre todos los valores que es posible representar con  $N$  bits (o un subconjunto de ellos) y actuar como parte de otros algoritmos cuánticos mayores.

## 7. Trabajo futuro

Como trabajo futuro obvio está la búsqueda de un circuito comparador que reduzca aún más el número de cúbits necesarios para su implementación. Ahora que se ha roto la barrera de los  $2N + 1$  cúbits, ¿existen diseños alternativos que permitan reducir aún más el circuito? ¿Se pueden representar algunos de los cúbits de  $B$  también mediante puertas cuánticas? ¿Es posible alguna simplificación en las operaciones del circuito que permita, tal vez con otras puertas o mediante otras operaciones auxiliares, reducir su complejidad? Y lo que es igualmente interesante: ¿se pueden hacer sumadores y otro tipo de circuitos aritméticos de forma más eficiente utilizando esta metodología?

Un estudio bastante más sencillo que el anterior pero igualmente útil e interesante sería el de comprobar las adaptaciones de los circuitos estudiados en este trabajo a las diferentes arquitecturas cuánticas actualmente disponibles. En este trabajo no se ha cubierto el hecho de que las implementaciones físicas de los dispositivos cuánticos no presentan uniones uno a uno entre todos los cúbits (Preskill, 2018). Esto produce que, si en el diseño de un circuito dos cúbits están conectados entre sí, pero estos dos cúbits están mapeados sobre dos cúbits físicos que no están físicamente conectados, sea necesario recurrir a operaciones swap que provocan un evidente aumento del coste cuántico y del retraso (O’Gorman et al., 2019; Wille et al., 2014). Resultaría muy interesante estudiar los métodos más eficientes para mapear circuitos cuánticos, así como su aplicabilidad en los circuitos estudiados de forma que pudiera estudiarse cómo de bien se pueden adaptar a las diferentes arquitecturas.

También resultaría sumamente interesante realizar un análisis de los circuitos propuestos en métricas específicas de las diferentes tecnologías estudiadas en la asignatura de Implementación Física de un Procesador Cuántico. Cada tecnología tiene sus propias ventajas y desventajas, por lo que las métricas estudiadas en este trabajo podrían resultar irrelevantes en algunas tecnologías concretas. Por ejemplo, se ha mencionado anteriormente que en óptica lineal es fundamental tratar de reducir lo máximo posible el número de puertas controladas ya que la puerta CNOT solo puede implementarse actualmente de forma probabilística (Lemr et al., 2015).

Finalmente, también podría resultar útil utilizar las diferentes técnicas y algoritmos estudiados en la asignatura Algoritmos Cuánticos para tratar de obtener circuitos comparadores mediante técnicas de inteligencia artificial, circuitos variaciones, y un largo etcetera. Por ejemplo, existen circuitos que permiten realizar la suma de manera aproximada que

han sido diseñados mediante estas técnicas (Gyongyosi y Imre, 2019). Aunque sería necesario considerar si una comparación aproximada puede resultar en algún contexto, si se consigue con la suficiente precisión para que el error sea relativamente despreciable sí podría resultar de utilidad.



# Bibliografía

- Amy, M., Maslov, D., Mosca, M., & Roetteler, M. (2013). A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6), 818-830.
- Andersson, E., Curty, M., & Jex, I. (2006). Experimentally realizable quantum comparison of coherent states and its applications. *Physical Review A*, 74(2), 022304.
- Angulo, J., Angulo, I., & Garcia, J. (2007). *Sistemas digitales y tecnología de computadores*. Ediciones Paraninfo, SA.
- Asadi, M.-A., Mosleh, M., & Haghparast, M. (2020). An efficient design of reversible ternary full-adder/full-subtractor with low quantum cost. *Quantum Information Processing*, 19(7), 204.
- Ball, H., Stace, T. M., Flammia, S. T., & Biercuk, M. J. (2016). Effect of noise correlations on randomized benchmarking. *Physical Review A*, 93(2), 022303.
- Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., & Weinfurter, H. (1995). Elementary gates for quantum computation. *Physical review A*, 52(5), 3457.
- Bauer, B., Bravyi, S., Motta, M., & Chan, G. K.-L. (2020). Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22), 12685-12717.
- Bell, J. S. (1964). On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1(3), 195.
- Bennett, C. H. (1973). Logical reversibility of computation. *IBM journal of Research and Development*, 17(6), 525-532.
- Bernhardt, C. (2019). *Quantum computing for everyone*. MIT Press.
- Berry, D. W., Childs, A. M., Ostrander, A., & Wang, G. (2017). Quantum algorithm for linear differential equations with exponentially improved dependence on precision. *Communications in Mathematical Physics*, 356, 1057-1081.
- Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., et al. (2022). Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1), 015004.
- Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, 549(7671), 195-202.

- Bloch, F. (1946). Nuclear induction. *Physical review*, 70(7-8), 460.
- Bodasingi, N., Varasala, K., Saladi, S., Chalumuri, A. N., Jammu, B. R., & Veeramachane-  
ni, S. (2022). Modified priority encoder based hardware efficient N-bit comparator.  
*International Journal of Electronics Letters*, 1-13.
- Bruce, J., Thornton, M. A., Shivakumaraiah, L., Kokate, P., & Li, X. (2002). Efficient adder  
circuits based on a conservative reversible logic gate. *Proceedings IEEE Computer  
Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design.  
ISVLSI 2002*, 83-88.
- Brylinski, J.-L., & Brylinski, R. (2002). Universal quantum gates. En *Mathematics of  
quantum computation* (pp. 117-134). Chapman; Hall/CRC.
- Burch, C. (2002). Logisim: A graphical system for logic circuit design and simulation.  
*Journal on Educational Resources in Computing (JERIC)*, 2(1), 5-16.
- Campbell, E. T., Terhal, B. M., & Vuillot, C. (2017). Roads towards fault-tolerant universal  
quantum computation. *Nature*, 549(7671), 172-179.
- Cao, Y., Romero, J., Olson, J. P., Degroote, M., Johnson, P. D., Kieferová, M., Kivlichan,  
I. D., Menke, T., Peropadre, B., Sawaya, N. P., et al. (2019). Quantum chemistry  
in the age of quantum computing. *Chemical reviews*, 119(19), 10856-10915.
- Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean,  
J. R., Mitarai, K., Yuan, X., Cincio, L., et al. (2021). Variational quantum algo-  
rithms. *Nature Reviews Physics*, 3(9), 625-644.
- Chakrabarti, A., & Sur-Kolay, S. (2008). Designing quantum adder circuits and evaluating  
their error performance. *2008 International Conference on Electronic Design*, 1-6.
- Chauhan, V., Negi, S., Jain, D., Singh, P., Sagar, A. K., & Sharma, A. K. (2022). Quantum  
Computers: A Review on How Quantum Computing Can Boom AI. *2022 2nd  
International Conference on Advance Computing and Innovative Technologies in  
Engineering (ICACITE)*, 559-563.
- Chu, W.-S., & Current, K. W. (1999). A CMOS voltage comparator with rail-to-rail input-  
range. *Analog integrated circuits and signal processing*, 19, 145-149.
- Combarro, E. A., & Gonzalez-Castillo, S. (2023). *A Practical Guide to Quantum Ma-  
chine Learning and Quantum Optimization: Hands-On primer to Quantum Com-  
puting: From Qubits to Quantum Machine Learning and Beyond* (Vol. 1). Packt  
Publishing.

- Cruz, D., Fournier, R., Gremion, F., Jeannerot, A., Komagata, K., Tasic, T., Thiesbrummel, J., Chan, C. L., Macris, N., Dupertuis, M.-A., et al. (2019). Efficient quantum algorithms for GHZ and W states, and implementation on the IBM quantum computer. *Advanced Quantum Technologies*, 2(5-6), 1900015.
- Daley, A. J., Bloch, I., Kokail, C., Flannigan, S., Pearson, N., Troyer, M., & Zoller, P. (2022). Practical quantum advantage in quantum simulation. *Nature*, 607(7920), 667-676.
- Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818), 97-117.
- Dirac, P. A. M. (1939). A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 35(3), 416-418.
- Du, S., Luo, K., Zhi, Y., Situ, H., & Zhang, J. (2022). Binarization of grayscale quantum image denoted with novel enhanced quantum representations. *Results in Physics*, 39, 105710.
- Floyd, T. L. (2011). *Digital fundamentals 10th Edition*. Pearson Education.
- Fredkin, E., & Toffoli, T. (1982). Conservative logic. *International Journal of theoretical physics*, 21(3-4), 219-253.
- Gaur, H., Singh, A., Mohan, A., & Pradhan, D. (2019). Computational analysis and comparison of reversible gates for design and test of logic circuits. *International Journal of Electronics*, 106(11), 1679-1693.
- Gidney, C. (2018). Halving the cost of quantum addition. *Quantum*, 2, 74.
- Gidney, C. (2023). Quirk Simulator. <https://algassert.com/quirk>
- Gidney, C., & Ekerå, M. (2021). How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5, 433.
- Github. (2020). GitHub. <https://github.com/>
- Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 212-219.
- Gyenis, A., Di Paolo, A., Koch, J., Blais, A., Houck, A. A., & Schuster, D. I. (2021). Moving beyond the transmon: Noise-protected superconducting quantum circuits. *PRX Quantum*, 2(3), 030101.

- Gyongyosi, L., & Imre, S. (2019). A survey on quantum computing technology. *Computer Science Review*, 31, 51-71.
- Han, T., & Carlson, D. A. (1987). Fast area-efficient VLSI adders. *1987 IEEE 8th symposium on computer arithmetic (ARITH)*, 49-56.
- Häner, T., Jaques, S., Naehrig, M., Roetteler, M., & Soeken, M. (2020). Improved quantum circuits for elliptic curve discrete logarithms. *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11*, 425-444.
- Harris, S. L., & Harris, D. (2015). *Digital design and computer architecture*. Morgan Kaufmann.
- Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15), 150502.
- Hasan, M., Saha, U. K., Sorwar, A., Dipto, M. A. Z., Hossain, M. S., & Zaman, H. U. (2019). A novel hybrid full adder based on gate diffusion input technique, transmission gate and static CMOS logic. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-6.
- Houck, A., Schreier, J., Johnson, B., Chow, J., Koch, J., Gambetta, J., Schuster, D., Frunzio, L., Devoret, M., Girvin, S., et al. (2008). Controlling the spontaneous emission of a superconducting transmon qubit. *Physical review letters*, 101(8), 080502.
- Huang, H.-Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021). Power of data in quantum machine learning. *Nature communications*, 12(1), 2631.
- Hung, W. N., Song, X., Yang, G., Yang, J., & Perkowski, M. (2006). Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis. *IEEE transactions on Computer-Aided Design of integrated circuits and Systems*, 25(9), 1652-1663.
- IBM. (2023). IBM Quantum. <https://quantum-computing.ibm.com/>
- Jazaeri, F., Beckers, A., Tajalli, A., & Sallese, J.-M. (2019). A review on quantum computing: From qubits to front-end electronics and cryogenic MOSFET physics. *2019 MIXDES-26th International Conference "Mixed Design of Integrated Circuits and Systems"*, 15-25.

- Jones, C. (2013). Low-overhead constructions for the fault-tolerant Toffoli gate. *Physical Review A*, 87(2), 022328.
- Kim, Y.-H., Kulik, S. P., & Shih, Y. (2001). Quantum teleportation of a polarization state with a complete Bell state measurement. *Physical Review Letters*, 86(7), 1370.
- Kim, Y., Morvan, A., Nguyen, L. B., Naik, R. K., Jünger, C., Chen, L., Kreikebaum, J. M., Santiago, D. I., & Siddiqi, I. (2022). High-fidelity three-qubit Toffoli gate for fixed-frequency superconducting qubits. *Nature Physics*, 18(7), 783-788.
- Kim, Y., Wood, C. J., Yoder, T. J., Merkel, S. T., Gambetta, J. M., Temme, K., & Kandala, A. (2023). Scalable error mitigation for noisy quantum circuits produces competitive expectation values. *Nature Physics*, 1-8.
- Kok, P., Munro, W. J., Nemoto, K., Ralph, T. C., Dowling, J. P., & Milburn, G. J. (2007). Linear optical quantum computing with photonic qubits. *Reviews of modern physics*, 79(1), 135.
- Lee, J., Berry, D. W., Gidney, C., Huggins, W. J., McClean, J. R., Wiebe, N., & Babbush, R. (2021). Even more efficient quantum computations of chemistry through tensor hypercontraction. *PRX Quantum*, 2(3), 030305.
- Leibfried, D., DeMarco, B., Meyer, V., Lucas, D., Barrett, M., Britton, J., Itano, W. M., Jenlenković, B., Langer, C., Rosenband, T., et al. (2003). Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate. *Nature*, 422(6930), 412-415.
- Lemr, K., Bartkiewicz, K., Černoč, A., Dušek, M., & Soubusta, J. (2015). Experimental implementation of optimal linear-optical controlled-unitary gates. *Physical Review Letters*, 114(15), 153602.
- Li, H. (2022). The optimization and application of 3-bit Hermitian gates and multiple control Toffoli gates. *IEEE Transactions on Quantum Engineering*, 3, 1-15.
- Li, H., Fan, P., Xia, H., & Long, G.-L. (2022). The circuit design and optimization of quantum multiplier and divider. *Science China Physics, Mechanics & Astronomy*, 65(6), 260311.
- Li, H., Fan, P., Xia, H.-Y., Peng, H., & Long, G.-L. (2020). Efficient quantum arithmetic operation circuits for quantum image processing. *Science China Physics, Mechanics & Astronomy*, 63, 1-13.
- Li, P., Shi, T., Zhao, Y., & Lu, A. (2020). Design of threshold segmentation method for quantum image. *International Journal of Theoretical Physics*, 59(2), 514-538.

- Litinski, D. (2019). Magic state distillation: Not as costly as you think. *Quantum*, 3, 205.
- Liu, Y., Arunachalam, S., & Temme, K. (2021). A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9), 1013-1017.
- Liu, Zheng, L., Wang, G., Shen, Y., & Liang, Y. (2019). A carry lookahead adder based on hybrid CMOS-memristor logic circuit. *IEEE Access*, 7, 43691-43696.
- Lye, A., Wille, R., & Drechsler, R. (2015). Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits. *The 20th Asia and South Pacific Design Automation Conference*, 178-183.
- Majumdar, R., & Sur-Kolay, S. (2020). Approximate ternary quantum error correcting code with low circuit cost. *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, 34-39.
- Merzbacher, E. (1998). *Quantum mechanics*. John Wiley & Sons.
- Mohammadi, M., & Eshghi, M. (2009). On figures of merit in reversible and quantum logic designs. *Quantum Information Processing*, 8, 297-318.
- Muñoz-Coreas, E., & Thapliyal, H. (2018). Quantum circuit design of a T-count optimized integer multiplier. *IEEE Transactions on Computers*, 68(5), 729-739.
- Murali, P., Baker, J. M., Javadi-Abhari, A., Chong, F. T., & Martonosi, M. (2019). Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers. *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, 1015-1029.
- Naseri, H., & Timarchi, S. (2018). Low-power and fast full adder by exploring new XOR and XNOR gates. *IEEE transactions on very large scale integration (VLSI) systems*, 26(8), 1481-1493.
- Nielsen, M. A., & Chuang, I. L. (2011). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.
- Noorallahzadeh, M., & Mosleh, M. (2019). Efficient designs of reversible latches with low quantum cost. *IET Circuits, Devices & Systems*, 13(6), 806-815.
- Noorallahzadeh, M., Mosleh, M., & Ahmadpour, S.-S. (2021). Efficient designs of reversible synchronous counters in nanoscale. *Circuits, Systems, and Signal Processing*, 40(11), 5367-5380.
- O’Gorman, B., Huggins, W. J., Rieffel, E. G., & Whaley, K. B. (2019). Generalized swap networks for near-term quantum computing. *arXiv preprint arXiv:1905.05118*.

- Orts, F., Ortega, G., Cucura, A., Filatovas, E., & Garzón, E. M. (2021). Optimal fault-tolerant quantum comparators for image binarization. *The Journal of Supercomputing*, 77, 8433-8444.
- Orts, F., Ortega, G., Combarro, E. F., & Garzón, E. M. (2020). A review on reversible quantum adders. *Journal of Network and Computer Applications*, 170, 102810.
- Orts, F., Ortega, G., & Garzón, E. M. (2022). Studying the cost of N-qubit Toffoli gates. *Computational Science–ICCS 2022: 22nd International Conference, London, UK, June 21–23, 2022, Proceedings, Part IV*, 122-128.
- Paredes-Barato, D., & Adams, C. (2014). All-optical quantum information processing using Rydberg gates. *Physical review letters*, 112(4), 040501.
- Park, J. L. (1970). The concept of transition in quantum mechanics. *Foundations of physics*, 1(1), 23-33.
- Patterson, D. A., Hennessy, J. L., & Goldberg, D. (1990). *Computer architecture: a quantitative approach* (Vol. 2). Morgan Kaufmann San Mateo, CA.
- Pauli, W. (1988). *Zur quantenmechanik des magnetischen elektrons*. Springer.
- Peres, A. (1985). Reversible logic and quantum computers. *Physical review A*, 32(6), 3266.
- Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. (2020). Data reuploading for a universal quantum classifier. *Quantum*, 4, 226.
- Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum*, 2, 79.
- Reed, M. D., DiCarlo, L., Nigg, S. E., Sun, L., Frunzio, L., Girvin, S. M., & Schoelkopf, R. J. (2012). Realization of three-qubit quantum error correction with superconducting circuits. *Nature*, 482(7385), 382-385.
- Rieffel, E. G., & Polak, W. H. (2011). *Quantum computing: A gentle introduction*. MIT Press.
- Roth Jr, C. H., Kinney, L. L., & John, E. B. (2020). *Fundamentals of logic design*. Cengage Learning.
- Sangouard, N., Lacour, X., Guerin, S., & Jauslin, H. (2005). Fast SWAP gate by adiabatic passage. *Physical Review A*, 72(6), 062309.
- Schmidt-Kaler, F., Häffner, H., Riebe, M., Gulde, S., Lancaster, G. P., Deuschle, T., Becher, C., Roos, C. F., Eschner, J., & Blatt, R. (2003). Realization of the Cirac–Zoller controlled-NOT quantum gate. *Nature*, 422(6930), 408-411.
- Sharma, K., Khatri, S., Cerezo, M., & Coles, P. J. (2020). Noise resilience of variational quantum compiling. *New Journal of Physics*, 22(4), 043006.

- Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2), 303-332.
- Smolin, J. A., & DiVincenzo, D. P. (1996). Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. *Physical Review A*, 53(4), 2855.
- Soumya, N., Kumar, K. S., Rao, K. R., Rooban, S., Kumar, P. S., & Kumar, G. N. S. (2019). 4-bit multiplier design using CMOS gates in electric VLSI. *International Journal of Recent Technology and Engineering*, 8(2), 1172-1177.
- Takahashi, Y., & Kunihiro, N. (2008). A fast quantum circuit for addition with few qubits. *Quantum Information & Computation*, 8(6), 636-649.
- Thapliyal, H., Munoz-Coreas, E., Varun, T., & Humble, T. S. (2019). Quantum circuit designs of integer division optimizing T-count and T-depth. *IEEE transactions on emerging topics in computing*, 9(2), 1045-1056.
- Thapliyal, H., & Ranganathan, N. (2010). Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 6(4), 1-31.
- Thomsen, M. K., Glück, R., & Axelsen, H. B. (2010). Reversible arithmetic logic unit for quantum arithmetic. *Journal of Physics A: Mathematical and Theoretical*, 43(38), 382002.
- Tittel, W., Brendel, J., Zbinden, H., & Gisin, N. (2000). Quantum cryptography using entangled photons in energy-time Bell states. *Physical review letters*, 84(20), 4737.
- Toffoli, T. (1980). Reversible computing. *Automata, Languages and Programming: Seventh Colloquium Noordwijkerhout, the Netherlands July 14-18, 1980* 7, 632-644.
- Wang, C., Li, X., Xu, H., Li, Z., Wang, J., Yang, Z., Mi, Z., Liang, X., Su, T., Yang, C., et al. (2022). Towards practical quantum computers: Transmon qubit with a lifetime approaching 0.5 milliseconds. *npj Quantum Information*, 8(1), 3.
- Wang, H., Ding, Y., Gu, J., Lin, Y., Pan, D. Z., Chong, F. T., & Han, S. (2022). Quantumnas: Noise-adaptive search for robust quantum circuits. *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 692-708.
- Wille, R., Lye, A., & Drechsler, R. (2014). Optimal SWAP gate insertion for nearest neighbor quantum circuits. *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 489-494.
- Wootters, W. K., & Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature*, 299, 802-803.



- Wootters, W. K., & Zurek, W. H. (2009). The no-cloning theorem. *Physics Today*, 62(2), 76-77.
- Wossnig, L., Zhao, Z., & Prakash, A. (2018). Quantum linear system algorithm for dense matrices. *Physical review letters*, 120(5), 050502.
- Xia, H.-Y., Li, H., Zhang, H., Liang, Y., & Xin, J. (2018). An Efficient Design of Reversible Multi-Bit Quantum Comparator Via Only a Single Ancillary Bit. *International Journal of Theoretical Physics*, 57(12), 3727-3744.
- Xia, H.-Y., Li, H., Zhang, H., Liang, Y., & Xin, J. (2019). Novel multi-bit quantum comparators and their application in image binarization. *Quantum Information Processing*, 18(7), 229.
- Xia, H.-Y., Zhang, H., Song, S.-X., Li, H., Zhou, Y.-J., & Chen, X. (2020). Design and simulation of quantum image binarization using quantum comparator. *Modern Physics Letters A*, 35(09), 2050049.
- Xue, C., Chen, Z.-Y., Wu, Y.-C., & Guo, G.-P. (2021). Effects of quantum noise on quantum approximate optimization algorithm. *Chinese Physics Letters*, 38(3), 030302.
- Yaghoubi, E., A. Bakhtiar, L., Adami, A., Hamidi, S., & Hosseinzadeh, M. (2014). All optical OR/AND/XOR gates based on nonlinear directional coupler. *Journal of Optics*, 43, 146-153.
- Yan, F., Ilyasu, A. M., & Jiang, Z. (2014). Quantum computation-based image representation, processing operations and their applications. *Entropy*, 16(10), 5290-5338.
- Yan, F., Ilyasu, A. M., & Le, P. Q. (2017). Quantum image processing: a review of advances in its security technologies. *International Journal of Quantum Information*, 15(03), 1730001.
- Yarkoni, S., Raponi, E., Bäck, T., & Schmitt, S. (2022). Quantum annealing for industry applications: Introduction and review. *Reports on Progress in Physics*.
- Yuan, S., Gao, S., Wen, C., Wang, Y., Qu, H., & Wang, Y. (2022). A novel fault-tolerant quantum divider and its simulation. *Quantum Information Processing*, 21(5), 182.
- Zhao, S., Li, H., Li, G., & Tang, X. (2022). The implementation of the enhanced quantum floating-point adder. *Modern Physics Letters A*, 37(26), 2250169.
- Zhuang, N., & Wu, H. (1992). A new design of the CMOS full adder. *IEEE journal of solid-state circuits*, 27(5), 840-844.