Algoritmo de Deutsch y Deutsch-Jozsa

Ciencias Naturales y Tecnología

Sebastián. Gómez[1]

[1] *Ingeniería de Sistemas, Escuela Colombiana de Ingeniería Julio Garavito, Bogotá DC, Colombia*

Fecha: 29/11/2019

Resumen— Los computadores clásicos le han permitido a la humanidad desarrollar algoritmos bastante complejos, pero gracias al avance del uso de tecnología poco a poco se quedan limitados por la forma en que operan los computadores. Desde hace mucho tiempo una nueva alternativa se esta abriendo camino, la *computación cuántica*. Con la creación de este nuevo paradigma de computación llegan nuevos algoritmos de la mano de Deutsch o Jozsa.

Palabras clave—Mecánica Cuántica, Deutsch, Deutsch-Jozsa, Algoritmo

Keywords— Quantum Computing, Deutsch, Deutsch-Jozsa, Algorithm

Objetivos

1. Verificar, en simuladores, el correcto funcionamiento de dos algoritmos cuánticos: el algoritmo de Deutsch y Deutsch-Jozsa
2. Observar las variaciones de resultados al realizar la simulación de los algoritmos en diferentes simuladores y diferentes shots

Problemas Deutsch

1. Implemente las 4 funciones posibles de 0,1 a 0,1 usando el computador cuántico de IBM.
   - Dibujo de función
   - Matriz correpondiente
   - Circuito correspondiente
   - Resultados de las 4 pruebas
2. Verifique que el algoritmo de Deutsch funciona para comprobar cuáles de estas funciones son balanceadas o constantes.
   - Circuito
   - Resultados

Solucion

En primero lugar se hacen los dibujos de las funciones con sus respectivas matrices como se ve en las figuras 1, 2, 3 y 4.

$\vert x> \vert y>$	f(x)	y+f(x)	$\vert x> \vert y+f(x)>$
$\vert 0> \vert 0>$	0	0	$\vert 0> \vert 0>$
$\vert 0> \vert 1>$	0	1	$\vert 0> \vert 1>$
$\vert 1> \vert 0>$	0	0	$\vert 1> \vert 0>$
$\vert 1> \vert 1>$	0	1	$\vert 1> \vert 1>$

$$U0 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Fig. 1: U0 tabla y matriz

$\vert x> \vert y>$	f(x)	y+f(x)	$\vert x> \vert y+f(x)>$
$\vert 0> \vert 0>$	1	1	$\vert 0> \vert 1>$
$\vert 0> \vert 1>$	1	0	$\vert 0> \vert 0>$
$\vert 1> \vert 0>$	1	1	$\vert 1> \vert 1>$
$\vert 1> \vert 1>$	1	0	$\vert 1> \vert 0>$

$$U1 = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

Fig. 2: U1 tabla y matriz

$\vert x> \vert y>$	f(x)	y+f(x)	$\vert x> \vert y+f(x)>$
$\vert 0> \vert 0>$	1	1	$\vert 0> \vert 1>$
$\vert 0> \vert 1>$	1	0	$\vert 0> \vert 0>$
$\vert 1> \vert 0>$	0	0	$\vert 1> \vert 0>$
$\vert 1> \vert 1>$	0	1	$\vert 1> \vert 1>$

$$Uc = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

Fig. 3: Uc tabla y matriz

|x> |y>	f(x)	y+f(x)	|x> |y+f(x)>
|0> |0>	0	0	|0> |0>
|0> |1>	0	1	|0> |1>
|1> |0>	1	0	|1> |0>
|1> |1>	1	1	|1> |1>

$$Ui = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$$

Fig. 4: Ui tabla y matriz

Después de tener la información de las funciones en la parte teórica se realizaron los circuitos correspondientes, en el simulador de IBM, a cada una de las funciones. Los circuitos se muestran en las figuras 5, 6, 7 y 8.

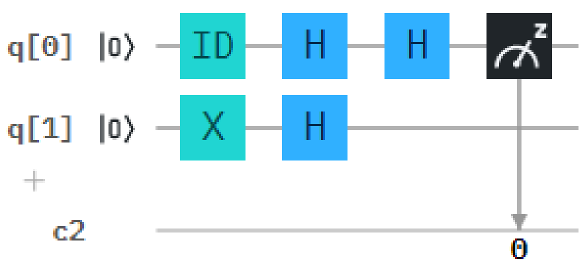

Fig. 5: Ciurcuito de la funcion U0.

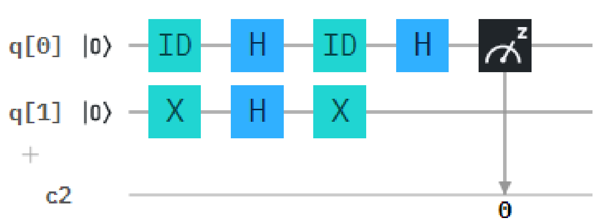

Fig. 6: Ciurcuito de la funcion U1.

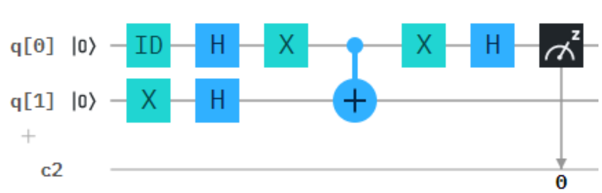

Fig. 7: Ciurcuito de la funcion Uc.

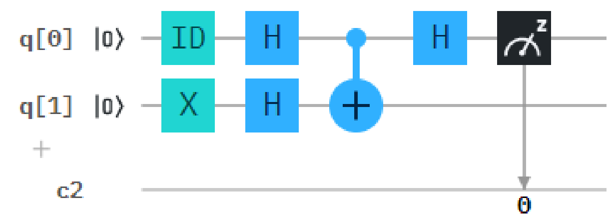

Fig. 8: Ciurcuito de la funcion Ui.

Con los circuitos definidos lo último que queda es correr la simulación del circuito. Para cada uno se realizaron dos simulaciones en dos computadores diferentes. La primera simulación de cada circuito se realizó en el computador 'ibmq qasm simulator' y en este solo se realizó un shot. Y la segunda simulación se realizó en el computador 'ibmq 16 melbourne' en el cual se enviaron 1024 shots por prueba.

Resultado U0

Los resultados de las pruebas para U0 se muestran a continuación.

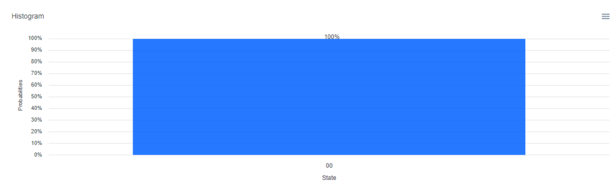

Fig. 9: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 00.

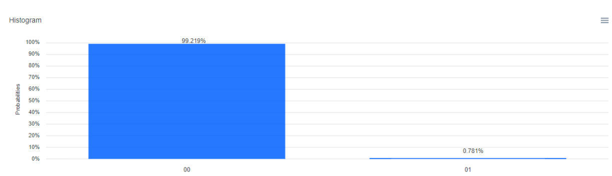

Fig. 10: Simulacion de 1024 shots

El resultado muestra que con un 99.21 % de probabilidad el sistema queda en el estado 00.
De estos resultados podemos concluir que el algoritmo nos está dando como respuesta en el alambre de arriba el estado 0, lo cual significa que la función U0 no es balanceada por lo tanto es constante, lo cual concuerda con nuestra función U0 la cual da todos los valores a 0.

Resultado U1

Ahora revisemos los resultados de las pruebas para U1.

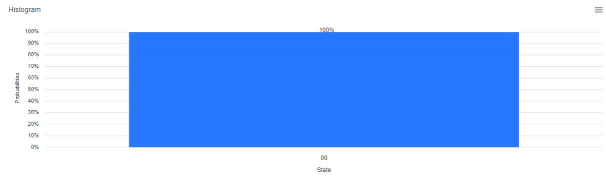

Fig. 11: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 00.

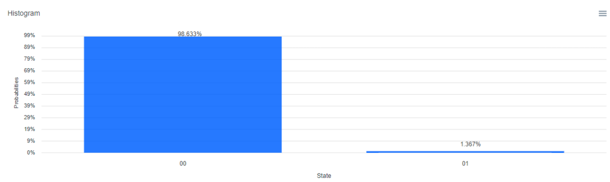

Fig. 12: Simulacion de 1024 shots

El resultado muestra que con un 98.63 % de probabilidad el sistema queda en el estado 00.
De estos resultados podemos concluir que el algoritmo nos está dando como respuesta en el alambre de arriba el estado 0, lo cual significa que la función U1, al igual que U0, es constante, lo cual concuerda con nuestra función U1 la cual da todos los valores a 1.

RESULTADO UC

Ahora revisemos los resultados de las pruebas para Uc.

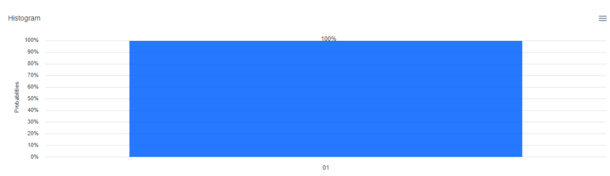

Fig. 13: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 01.

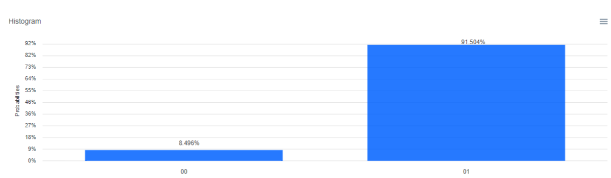

Fig. 14: Simulacion de 1024 shots

El resultado muestra que con un 91.50 % de probabilidad el sistema queda en el estado 0.

De estos resultados podemos concluir que el algoritmo nos está dando como respuesta en el alambre de arriba el estado 01, lo cual significa que la función UcEs balanceada, lo cual concuerda con nuestra función Uc la cual da como resultado f(0) = 1 y f(1) = 0.

RESULTADO UI

Los resultados de las pruebas para Ui se muestran a continuación.

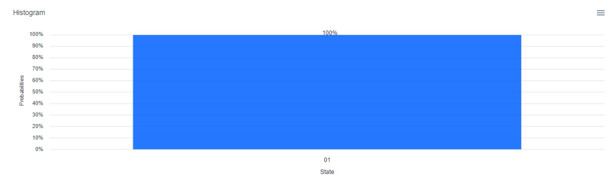

Fig. 15: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 01.

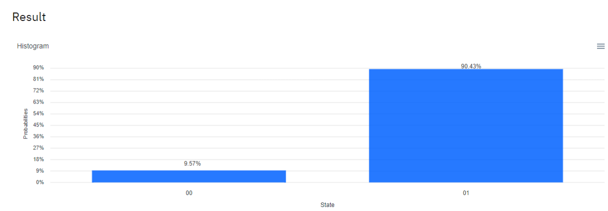

Fig. 16: Simulacion de 1024 shots

El resultado muestra que con un 90.43 % de probabilidad el sistema queda en el estado 01.
De estos resultados podemos concluir que el algoritmo nos está dando como respuesta en el alambre de arriba el estado 1, lo cual nos quiere decir que la funcion es balanceada ya que, si revisamos nuestra función Ui inicial da como resultado f(0) = 0 y f(1) = 1.

PROBLEMAS DEUTSCH-JOZSA

1. Implemente al menos 2 funciones con n= 2 para probar el funcionamiento del algoritmo Deustch-Jozsa
   - Dibujo de función
   - Matriz correpondiente
   - Circuito correspondiente
   - Resultados de las pruebas

SOLUCION

Como primer paso se seleccionaron dos funciones las cuales llamaremos U0 y Uc. esats funciones se definene en los siguientes grafos.

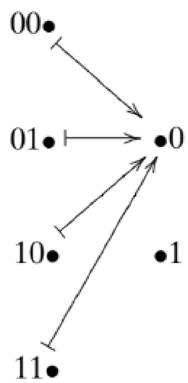

Fig. 17: Gráfo para la funcion U0

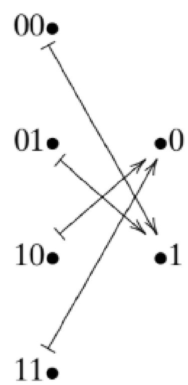

Fig. 18: Gráfo para la funcion Uc

Partiendo de estos gráfo tenemos que las matrices de estos quedan expresadas según lo mustran las figuras 19 y 20.

$\vert x\rangle\ \vert y\rangle$	f(x)	y+f(x)	$\vert x\rangle\ \vert y+f(x)\rangle$
$\vert 00\rangle\ \vert 0\rangle$	0	0	$\vert 00\rangle\ \vert 0\rangle$
$\vert 00\rangle\ \vert 1\rangle$	0	1	$\vert 00\rangle\ \vert 1\rangle$
$\vert 01\rangle\ \vert 0\rangle$	0	0	$\vert 01\rangle\ \vert 0\rangle$
$\vert 01\rangle\ \vert 1\rangle$	0	1	$\vert 01\rangle\ \vert 1\rangle$
$\vert 10\rangle\ \vert 0\rangle$	0	0	$\vert 10\rangle\ \vert 0\rangle$
$\vert 10\rangle\ \vert 1\rangle$	0	1	$\vert 10\rangle\ \vert 1\rangle$
$\vert 11\rangle\ \vert 0\rangle$	0	0	$\vert 11\rangle\ \vert 0\rangle$
$\vert 11\rangle\ \vert 1\rangle$	0	1	$\vert 11\rangle\ \vert 1\rangle$

$$U = \begin{vmatrix} 1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{vmatrix}$$

Fig. 19: Matriz para U0

$\vert x\rangle\ \vert y\rangle$	f(x)	y+f(x)	$\vert x\rangle\ \vert y+f(x)\rangle$
$\vert 00\rangle\ \vert 0\rangle$	1	1	$\vert 00\rangle\ \vert 1\rangle$
$\vert 00\rangle\ \vert 1\rangle$	1	0	$\vert 00\rangle\ \vert 0\rangle$
$\vert 01\rangle\ \vert 0\rangle$	1	1	$\vert 01\rangle\ \vert 1\rangle$
$\vert 01\rangle\ \vert 1\rangle$	1	0	$\vert 00\rangle\ \vert 0\rangle$
$\vert 10\rangle\ \vert 0\rangle$	0	0	$\vert 10\rangle\ \vert 0\rangle$
$\vert 10\rangle\ \vert 1\rangle$	0	1	$\vert 10\rangle\ \vert 1\rangle$
$\vert 11\rangle\ \vert 0\rangle$	0	0	$\vert 11\rangle\ \vert 0\rangle$
$\vert 11\rangle\ \vert 1\rangle$	0	1	$\vert 11\rangle\ \vert 1\rangle$

$$U = \begin{vmatrix} 0&1&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{vmatrix}$$

Fig. 20: Matriz para Uc

Con las matrices hechas el siguiente paso para la comprobación del algoritmo es realizar los circuitos. Para esta tarea nos apoyaremos nuevamente en el simulador cuántico de IBM. El circuito para U0 queda como lo muestra la figura 21.

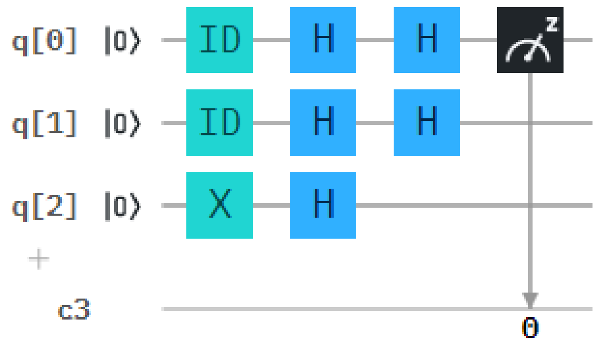

Fig. 21: Circuito para U0

Y el circuito para la función Uc queda según lo muestra la figura 22.

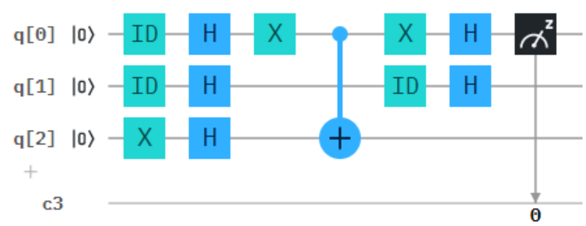

Fig. 22: Circuito para Uc

Con ayuda del cuircuito montado en IBMQ se realizaron dos simulaciones para cada circuito. La primera fue en el computador 'ibmq qasm simulator' y en este solo se realizó un shot. La segunda simulación se realizó en el computador 'ibmq 16 melbourne' en el cual se enviaron 1024 shots por prueba.

Resultados para U0

En simulacion en el computador 'ibmq qasm simulator' para la función U0 se obtuvieron los siguientes resultados.

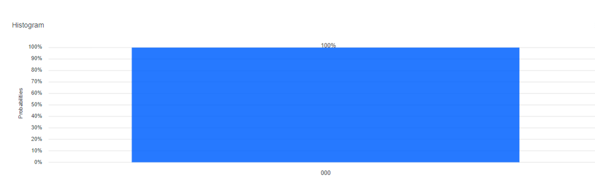

Fig. 23: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 000.
En a simulacion de 1024 shots muestra los siguientes resultados.

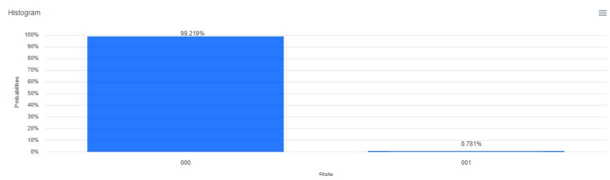

Fig. 24: Simulacion de 1024 shot

El resultado muestra que con un 99.22 % de probabilidad el sistema queda en el estado 000.

Estos resultados nos dejan ver que como resultado de la prueba en el alambre 1 se tiene el estado 0 lo cual quiere decir que el algoritmo de Deustch-Jozsa nos dice que la función es constante y según el grafo que planteamos al inicio es correcto el resultado.

RESULTADOS PARA UC

En simulacion de un solo shot para la función Uc se obtuvieron los siguientes resultados.

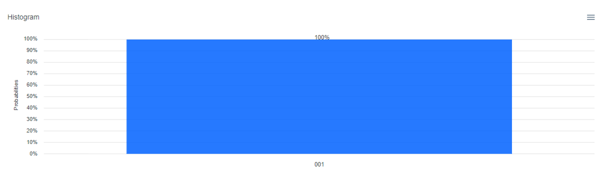

Fig. 25: Simulacion de un solo shot

El resultado muestra que con un 100 % de probabilidad el sistema queda en el estado 001.
En a simulacion de 'ibmq 16 melbourne' con 1024 shots muestra los siguientes resultados.

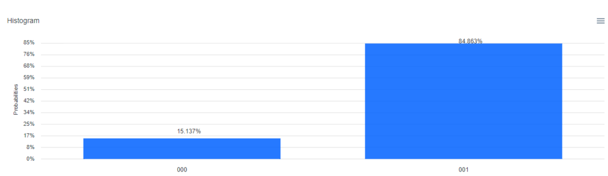

Fig. 26: Simulacion de 1024 shot

El resultado muestra que con un 84.86 % de probabilidad el sistema queda en el estado 001.

Estos resultados nos dejan ver que como conclusión de la prueba en el alambre 1 se tiene el estado 1 lo cual quiere decir, según el algoritmo de Deustch-Jozsa, que la función es es balanceada y según el grafo que planteamos al inicio es correcto el resultado.

CONCLUSIONES

- Al realizar la implementación de los algoritmos podemos evidenciar la gran importancia de este nuevo paradigma de programación como es la computación cuántica, la cual no permite crear y expandir nuevas áreas de conocimiento las cuales podremos aplicar en un corto tiempo a actividades cotidianas.

- Cuando se revisan los resultados de las pruebas de los algoritmos que se ejecutaron en diferentes simuladores podemos evidenciar una gran diferencia con los algoritmos clásicos, la cual deja en evidencia que en la computación cuántica no podemos tener un 100 % de certeza en los resultados lo cual no pasa con los algoritmos ni los computadores clásicos.