EAFIT University
Master in Data Science and Analytics
Quantum Computing I Course
Student: B. Sc. Jonathan J. Montes C.
Physicist.
Professor: Juan Guillermo Lalinde
Semester 2022-2, Date: Oct, 2022

Assignment No.2

1 A bit of physics: Deutsch-Jotzsa Algorithm

Antes de describir cada circuito que se utilizó para describir el algoritmo de Deutsch-Jotzsa, me gustaría realizar un *background* teórico para ponernos en contexto y poder comprender bien los resultados. En la figura 1 podemos ver el algoritmo de Deutsch-Jotzsa. Este no es más que una generalización del algoritmo de Deutsch [**1**]. La idea de este algoritmo es el demostrar como los computadores cuánticos pueden solucionar ciertas funciones más rápido que las computadores clásicos.

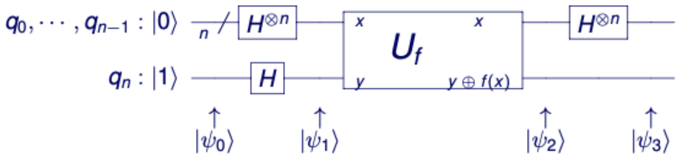

Figure 1: Generalización del Algoritmo de Deutsch

En este algoritmo se tienen como entrada **n** estados cuánticos en superposición por medio de compuertas Hadamard, y un último estado cuántico $|0\rangle$. Este último estado lo generé en los circuitos que desarrolle en Qiskit (IBM) por medio de una compuerta C-NOT. Se puede expandir las entradas de la figura 1 para ver las entradas **n** de los estados $|0\rangle$ en la figura 2.

El poder del algoritmo se basa en poder determinar en una sola pasada o compilación, si las salidas al ser medidas son todas ceros o unos, o la mitad ceros y la mitad uno, esto por medio de la probabilidad de los estados en la parte final del circuito, tema que explicaré más adelante por medio de la ecuación de probabilidad cuando se colapsan

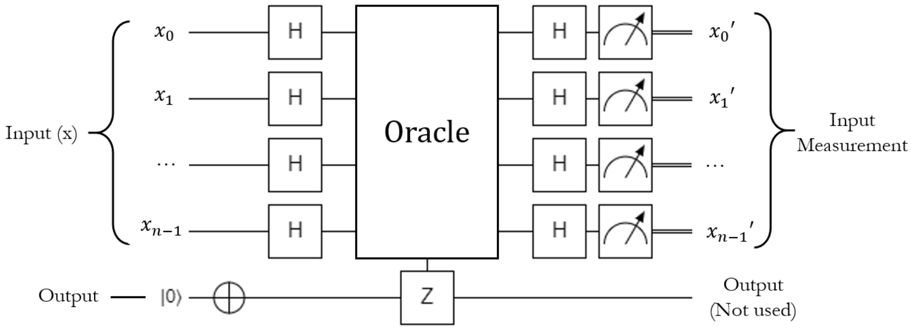

Figure 2: Esquema del Algoritmo de Deutsch-Jozsa

las funciones de onda de los estados cuánticos, i.e, se mide al final del circuito. La función oráculo, de la cual no tenemos información en detalle de su forma y que está representada por medio de una compuerta y/o operación unitaria $\hat{U}$ entre $|\psi_1\rangle$ y $|\psi_2\rangle$ será balanceada si obtenemos la mitad de estados de salida ceros y la otra mitad 1, y se define constante cuando las salidas son todas ceros o todas 1. El tiempo de ejecución de una computadora clásica viene determinada por

$$2^{n-1} + 1$$

La diferencia con la computación clásica y la que caracteriza este algoritmo tan particular, es que hace uso del principio de superposición de la mecánica cuántica para realizar el cálculo y determinar si la función oráculo es constante o balanceada en una sola iteración.

Analicemos matemáticamente y físicamente cada parte del algoritmo de la figura 1:

1. El estado $|\psi_0\rangle$ es $|0\rangle^{\otimes n}|1\rangle$. $\otimes n$ significa que se están multiplicando **n** ceros que están en paralelo. Al pasar por las compuertas Hadamard $H^{\otimes n+1}$ el estado es el que se obtiene en $|\psi_1\rangle$ dado por

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

donde **n** se refiere al número de qubits y $|x\rangle$ son los posibles estados de x dentro de la función oráculo. Por ejemplo, si $n = 2$, x tomará los valores $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. Si por ejemplo tenemos que $|y\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \hat{H}|1\rangle$ se tiene que:

$$(-1)^{f(x)}|x\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \to \text{es equivalente a}\frac{|xf(x)\rangle - |x!f(x)\rangle}{\sqrt{2}},$$

siendo $!f(x)$ el opuesto a f(x). Por ejemplo, si f(x)=0 tenemos:

$$\frac{|x0\rangle - |x1\rangle}{\sqrt{2}},$$

y con f(x)=1:

$$\frac{|x1\rangle - |x0\rangle}{\sqrt{2}}.$$

Ahora, recordemos que siendo $|x\rangle$ un estado de superposición, al aplicar $\hat{U}_f$, se evalúan de manera simultánea toods los posibles valores de $f(x)$:

$$|x\rangle |0\rangle \xrightarrow{U_f} |x\rangle |f(x)\rangle \tag{2}$$

$$|x\rangle |1\rangle \xrightarrow{U_f} |x\rangle |1 \oplus f(x)\rangle . \tag{3}$$

Y por tanto:

$$|x\rangle |-\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle |-\rangle . \tag{4}$$

En el algoritmo de Deutch se puede tener en la función oráculo la compuerta de control X entre dos Hadamards para obtener la compuerta Z, la cual en función del input activa el control o no.

2. Al aplicar $\hat{U}_f$ a $|\psi_1\rangle$, i.e, en la ecuación 1, se obtiene, usando 4:

$$|\psi_2\rangle = \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \frac{|x\rangle}{\sqrt{2}^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right). \tag{5}$$

Con $H|x\rangle = \sum_{z\in\{0,1\}} \frac{(-1)^{x\cdot z}|z\rangle}{\sqrt{2}}$, se tiene entonces de forma general que:

$$H^{\otimes n} |x\rangle = \sum_{z\in\{0,1\}^n} \frac{(-1)^{x\cdot z} |z\rangle}{\sqrt{2}^n} \tag{6}$$

Y por tanto al aplicar 5 a 6 queda finalmente:

$$\sum_{x\in\{0,1\}}(-1)^{f(x)}\frac{\sum_{z\in\{0,1\}^n}\frac{(-1)^{x\cdot z}|z\rangle}{\sqrt{2}^n}}{\sqrt{2}^n}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right). \tag{7}$$

y finalmente:

$$|\psi_3\rangle=\sum_{x,z\in\{0,1\}^n}\frac{(-1)^{f(x)+x\cdot z}\,|z\rangle}{2^n}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right). \tag{8}$$

Podemos ver que esta última ecuación para nuestro caso de estudio es similar a:

$$|\psi_3\rangle=\frac{1}{2^n}\sum_{y=0}^{2^n-1}\left[\sum_{y=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle\,. \tag{9}$$

Con amplitud de probabilidad:

$$|\langle\psi_y|\psi_y\rangle|^2=\psi_y=\frac{1}{2^n}\left|\sum_{y=0}^{2^n-1}(-1)^{f(x)}\right|^2, \tag{10}$$

por tanto si $\psi_y=0$ es porque se anulan todos y queda $Pr(y=0^n)$, de manera que se tienen valores de $f(x)$ la mitad unos y y la otra mitad ceros, por tanto si la función oráculo será balanceada si nuestro resultado en términos de probabilidades es igual a cero . Esta última ecuación 10 mide las amplitudes de probabilidad de cuáles serán los *outputs* o *qubits*. Si la función es balanceada los *outputs* se cancelan unos a otros y tenemos probabilidad cero, de lo contrario, si es constante, i.e, $f(x)=0$ o $f(x)=1$ para $n\in\{0,1\}$ se tiene $\frac{|1+1+1+1|}{2^2}=1$ y $\frac{|1-1-1-1|}{2^2}=1$, por tanto si la función es constante se obtendrá probabilidad 1.

2 Primer Circuito: Problema de n-tuplas con $n=2$

El primer circuito lo elaboré para el primer caso $n=2$, i.e, una tupla con dos entradas o dos qubits 0 y 1 en el que se cumple para una función (el caso de estudio más simple):

$$f(x)=\begin{cases}1 & \text{si } x=0\\ 0 & \text{si } x=1\end{cases}$$

El primer circuito, el más simple para $n=2$ está en el jupyter adjunto llamado *Deutch_Jotzsa_constant_oracle_function_for_tuples_with_n=2.ipynb*, representado en la figura

3. Podemos observar que $\hat{U}$ está determinada por una compuerta C-NOT, donde se tiene que el qubit de control es el $\hat{H}\,|0\rangle$ y el *target* es el qubit $\hat{H}\,|1\rangle$.

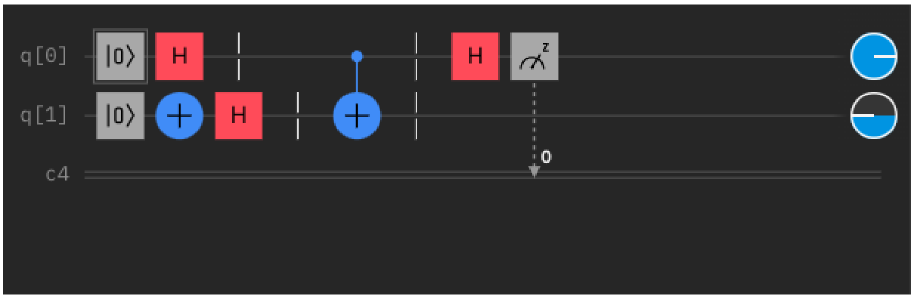

Figure 3: Algoritmo de Deutsch-Jozsa para dos entradas, i.e, $n = 2$, con $|0\rangle$ y $|1\rangle$

Se obtiene una probabilidad de como puede observase en la figura 4, donde se realizaron 1024 *shots*, que vienen por *default* en el *composer* de IBM, el cuál exporté en un jupyter para ser compilado con la librería de qiskit.

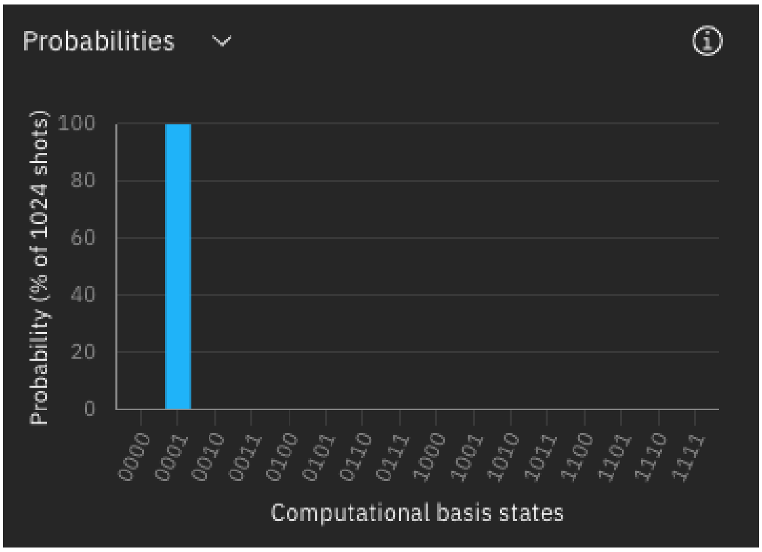

Figure 4: Probabilidad medida con $n = 2$

Dado el valor de las probabilidad del estado para el valor 1, i.e, $|0001\rangle$, tenemos que esta es una función constante.

3 Segundo Circuito: de n-tuplas con $n = 3$

El circuito se puede observar en la figura 5. Sus probabilidades se pueden ver en la figura 6 dando como resultado una función balanceada, ya que la salida para el valor 0 da % de probabilidad por tanto se concluye que es una función balanceada.

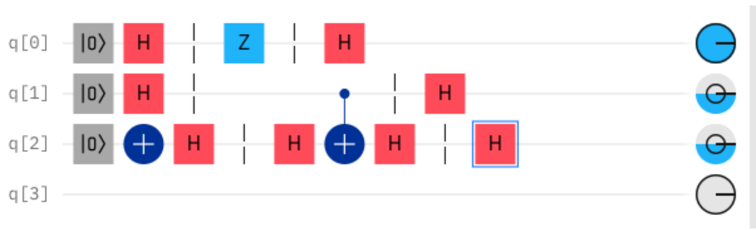

Figure 5: Algoritmo de Deutsch-Jozsa para tres qubits, i.e, $n = 3$

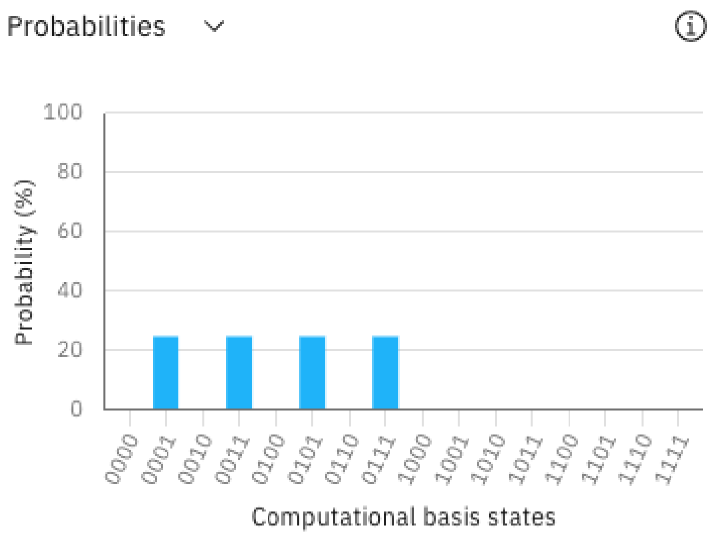

Figure 6: Probabilidad con $n = 3$

4 Tercer Circuito: de n-tuplas con $n = 3$

Este último circuito se puede observar en la figura 7. Es una variación al que presenta Qiskit en su página sobre el algoritmo de Deutch-Jozsa [**2**]. Sus probabilidades se pueden ver en la figura 8. Se obtiene un 50 % de probabilidad para los valores 1 y 0, por tanto se concluye que la función es balanceada. Este circuito lo presento en el *notebook* de

Colab llamado *Deutsch_Jozsa_Algorithm_3.ipynb*. Aquí está en detalle cómo construí este algoritmo.

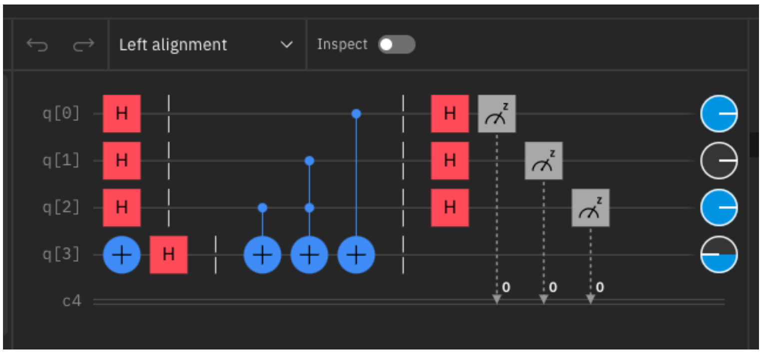

Figure 7: Algoritmo de Deutsch-Jozsa para tres qubits, i.e, $n = 4$

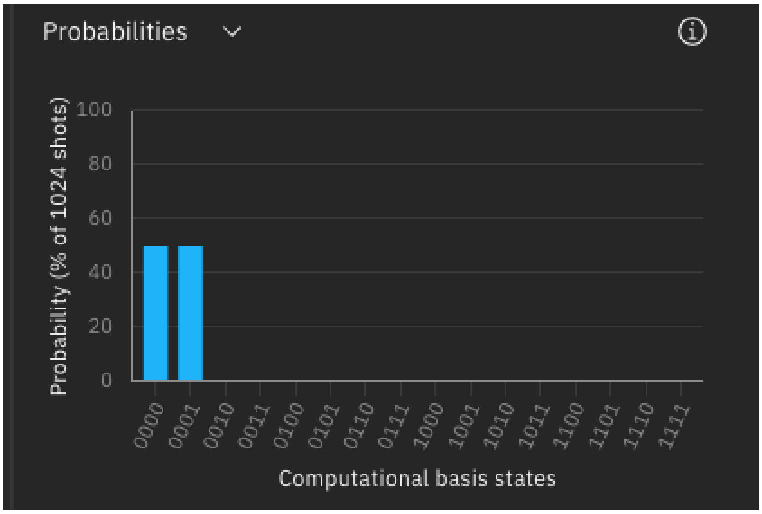

Figure 8: Probabilidad con $n = 4$

References

[1] Quantum Computation and Quantum Information. Michael A. Nielsen and Isaac L.Chuang. Tenth Edition. Editor: Cambridge. Year 2012.

[2] https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html