Jean-Michel Torres
IBM Quantum France
IBM Systems Center | Montpellier

Jouer avec des qubits : superposez les initiales de votre nom avec un algorithme quantique !

Abordons quelques notions et techniques de base du calcul quantique pour les mettre à l'œuvre sous la forme d'un petit jeu, avec presque pas de mathématique ni de physique quantique.

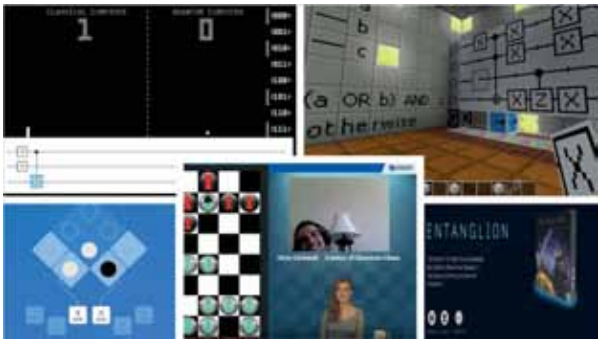
Le jeu dans les technologies informatiques.

Lorsqu'une technologie progresse il est un signe qui ne trompe pas : les jeux fleurissent. Qu'il s'agisse d'exercices de style ou de programmation, de support de communication et de vulgarisation ou carrément d'exploitation commerciale, les exemples sont nombreux.
Mieux : l'industrie du jeu devient un moteur d'innovation (la PlayStation 3 de Sony, sortie en 2006 et vendue à 86 millions d'exemplaires embarque un processeur de type Cell à 10 cœurs qui fait figure de prouesse technologique à l'époque). Ce n'est qu'un exemple de la riche lignée des consoles de jeu. Le jeu Pong sort sur arcade en 1972, et la console HomePong est commercialisée par Atari la même année, à l'origine il s'agissait d'un exercice de programmation pour son concepteur.
Et pourtant, le premier micro-ordinateur n'a pas encore vu le jour (d'ailleurs il est français, s'appelle Micral et sort en 1973, avant Amstrad, IBM PC et Apple II (voir « Technosaures #1 » de François Tonic !)). Plus tard, les ordinateurs et les consoles ne cesseront de rivaliser dans le domaine du jeu, chacun essayant de tirer parti de ses avantages, et forçant l'autre plateforme à innover sans cesse, les ventes se comptent en dizaines de millions d'unités.

Le calcul quantique se présente donc comme un nouvel acteur : depuis 2017 on peut jouer à « Entanglion » un jeu de plateau imaginé par des scientifiques d'IBM pour vulgariser quelques notions et un peu de vocabulaire, il s'agit de diriger son vaisseau spatial au travers d'un univers étrange où certains phénomènes quantiques se manifestent à grande échelle.

A la même époque, un jeu d'échec dont le comportement des pièces illustrent les propriétés quantiques apparaît, une partie contre une championne d'échec est à voir sur Youtube : « Anna Rudolf plays Quantum Chess ».
Sur l'Apple Store le jeu Hello Quantum permet de résoudre de petits puzzles à l'aide d'une logique inspirée des portes quantiques utilisées dans le calcul quantique, et il y a d'autres exemples : une version QPONG (disponible sur github) et on peut construire des circuits quantiques à résoudre dans Minecraft…

Je propose ici un petit défi consistant à superposer les initiales de votre nom et de votre prénom, c'est un prétexte pour apprendre quelques notions de base et programmer vos premiers circuits.



Nous allons superposer les valeurs « binaires » des code ASCII des initiales de votre prénom et de votre nom. Il s'agit donc de superposer des 1 et des 0 ou plus précisément les états quantiques de quelques qubits (quantum bits) pour former le code ASCII de deux caractères à la fois.

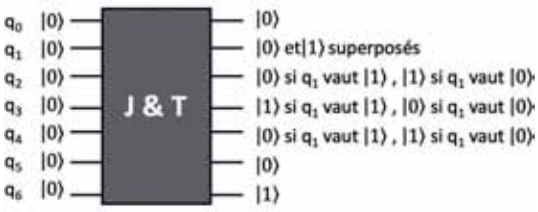
Le qubit est au calculateur quantique ce qu'un bit est à l'ordinateur classique : la plus petite quantité d'information traitée (en « classique », on utilise souvent les groupes de 8 bits (octets) et on les rencontre en général en très grandes quantités : GigaOctets, TeraOctets).
En calcul quantique, on compte encore les qubits sur les doigts de quelques mains, mais contrairement au bit qui ne peut valoir que 0 ou 1, un qubit peut se trouver dans l'état 0 ou 1, on note $|0\rangle$ et $|1\rangle$, mais aussi dans toute combinaison de ces deux états.

Pour « superposer » vos initiales, il faut d'abord connaître leur valeur ASCII (si la table des code ASCII n'est plus affichée dans votre bureau, vous la trouverez facilement sur le web). Dans cet exemple je vais utiliser « J » et « T » : dont les valeurs respectives sont 0x4A et 0x54, c'est-à-dire, en binaire : 100 1010 et 101 0100 ce sont les deux valeurs que je souhaite « superposer » :

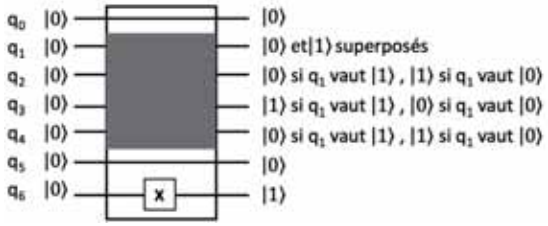
Voici le problème résumé dans ce tableau avec en vert les bits qui ont la même valeur pour le J et le T, en jaune ceux qui ont une valeur différente :

	q_6	q_5	q_4	q_3	q_2	q_1	q_0
J	1	0	0	1	0	1	0
T	1	0	1	0	1	0	0

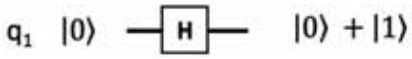
Il s'agit donc de créer un algorithme ou un circuit quantique utilisant 7 qubits (partant tous de l'état $|0\rangle$), dont les entrées et sorties peuvent être représentées de la manière suivante (pour le cas J et T) :



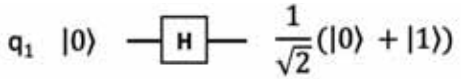
Pour le qubit 0 : l'entrée est à $|0\rangle$, je veux $|0\rangle$ en sortie, il suffit de ne rien faire
Pour le qubit 5 (second à gauche) : c'est la même chose
Pour le qubit 6 (le plus à gauche) : je veux toujours la valeur $|1\rangle$, comme il est à $|0\rangle$ au début, il faut lui appliquer une porte NOT (dans la panoplie des portes quantiques, elle s'appelle X).
On a fait le plus simple et presque la moitié du chemin :



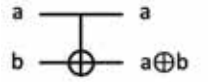
Pour les quatre autres (q_1 à q_4) : si j'obtiens une superposition de J et T, c'est que le qubit q_1 est « à moitié » dans l'état $|0\rangle$ et « à moitié » dans l'état $|1\rangle$. Il existe justement un opérateur (autrement dit une porte quantique), qui permet de réaliser ceci : la porte de Hadamard notée H (d'après le mathématicien français Jacques Hadamard). On aura donc :



En réalité, pour des raisons de propriété des états de qubit, il y a un facteur $\frac{1}{\sqrt{2}}$, le schéma correct est :

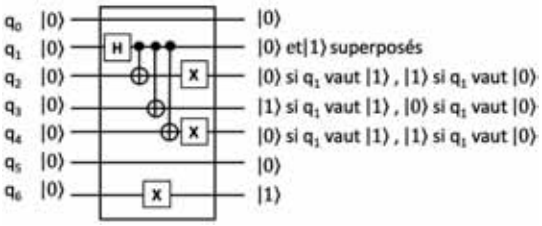


L'idée est que l'on a un état « avec autant de 0 que de 1 dedans » et quand on va le mesurer, comme on revient dans un monde « classique » et que ceci n'a pas de sens, on va trouver comme résultat de mesure tantôt $|0\rangle$ et tantôt $|1\rangle$ sans qu'il soit possible de le prédire, et avec une probabilité ½ pour chaque cas.
Alors, les autres qubits (q_2, q_3, q_4) doivent également être superposés mais dans un état qui dépendra de l'état de q_1 (on ne veut pas avoir q_2 à $|1\rangle$ si q_1 est à $|1\rangle$, cf. le tableau ci-dessus).
Là aussi il existe une porte quantique, sur deux qubits, qui permet d'obtenir ce résultat il s'agit de la porte Control-Not (CNOT), elle va inverser l'état d'un qubit selon l'état de l'autre :



$a \oplus b$ représente le « ou exclusif » pour les valeurs de a et b.
Si je fais passer q_1 et q_3 dans une telle porte, j'aurai en sortie le résultat escompté : la sortie en face de q_3 vaudra $|1\rangle$ si q_1 vaut $|1\rangle$ et $|0\rangle$ si q_1 vaut $|0\rangle$.
Pour q_2 et q_4 c'est la même chose, sauf qu'il faudra inverser le résultat avec une porte X.

Finalement voici mon circuit (pour J et T) :



On vérifie facilement que l'on obtient soit J soit T, selon la valeur de la sortie en face de q_1.
On peut alors utiliser la bibliothèque de programmation quantique « QISKit » pour Python, en utilisant par exemple un notebook Jupyter.
Import des objets et bibliothèques nécessaires :

```
from qiskit import ClassicalRegister, QuantumRegister
from qiskit import QuantumCircuit, execute
from qiskit.tools.visualization import plot_histogram
from qiskit import IBMQ, BasicAer
from qiskit.tools.jupyter import *
import matplotlib.pyplot as plt
%matplotlib inline
```

Construction du circuit :

```
# set up les registres et le circuit
qr = QuantumRegister(7)
cr = ClassicalRegister(7)
qc = QuantumCircuit(qr, cr)

# x sur les qubit qui valent 1 dans les deux cas
qc.x(qr[6])

# h sur le qubit de poids le plus faible qui change
qc.h(qr[1])
# cnot du qubit précédent sur tous ceux qui changent également
qc.cx(qr[1],qr[2])
qc.cx(qr[1],qr[3])
qc.cx(qr[1],qr[6])

# not sur ceux qui changent de manière opposée au premier
qc.x(qr[2])
qc.x(qr[4])


qc.barrier()
# mesures et dessin
qc.measure(qr, cr)
qc.draw(output='mpl')
```

Exécution :

A votre tour ! Saurez-vous reconstruire le circuit et l'exécuter pour vos propres initiales ?
Si vous avez un nom composé ou plusieurs prénoms, c'est plus compliqué, contactez l'auteur !