

Introduction à l'informatique quantique:

En lisant les articles de presse sur l'informatique quantique, je pensais que c'était un domaine extrêmement complexe, inaccessible et très abstrait, réservé au monde de la recherche. J'avais le vague sentiment qu'il existait très peu de choses concrètes pour développer en informatique quantique. En résumé, que c'était un écran de fumé publicitaire pour faire la promotion sur l'innovation informatique.

Eh bien, je me trompais en partie. C'est un domaine complexe mais tout à fait accessible pour un informaticien à condition d'avoir une bonne base scientifique et il existe déjà beaucoup d'outils permettant de commencer la pratique. Mais les progrès restent à faire en ce qui concerne la fabrication des ordinateurs quantiques, les réseaux quantiques, la sécurité, etc.

Je vais tenter à travers ce premier article de poser les bases de la compréhension de l'informatique quantique sans pour autant vous noyer dans les équations mathématiques et les théorèmes physiques, même si c'est indispensable pour comprendre les bases.

Mon exposé suivra le fil conducteur suivant:

1. L'informatique quantique: à quoi ça sert?
2. Les avantages et les inconvénients de l'ordinateur quantique vs l'ordinateur traditionnel
3. Les fondamentaux pour comprendre et développer en informatique quantique
4. La programmation quantique
5. D'autres explorations en informatique quantique à voir

1. L'informatique quantique: à quoi ça sert?

Olivier Ezratty a écrit un e-book très intéressant sur l'état actuel du marché de l'informatique quantique. Je vous encourage à le lire si vous souhaitez avoir un aperçu global de ce qu'on en fait. <https://www.oezratty.net/wordpress/wp-content/themes/Ezratty5/forcedownload.php?file=/Files/Publications/Comprendre%20Informatique%20Quantique%20Olivier%20Ezratty.pdf>

L'informatique quantique a de nombreuses applications:

Elle permet de:

- Réaliser les optimisations combinatoires (trajets, placements, cartes, finance) dans la logistique, les réseaux de distribution d'électricité, de gaz, ..., les cartographies et le domaine financier
- Faire des simulations moléculaires (matériaux et biologie) pour les modélisations
- Effectuer la factorisation de très grands nombres entiers (sécurité, cryptographie post quantique, ...)
- Enrichir les possibilités de l'intelligence artificielle (machine learning et deep learning)
- etc.

Tout est imaginable si on a une bonne compréhension de ce qu'est l'informatique quantique et ses limites.

2. Les avantages et les inconvénients de l'informatique quantique par rapport à l'informatique traditionnelle

Il y a bien sûr de nombreux avantages et inconvénients. Je ne vais pas vous faire une liste exhaustive, mais les principaux sont là, dans l'état actuel des choses.

Les avantages:

- Résoudre des problèmes complexes qu'un ordinateur traditionnel ne pourrait jamais résoudre dans un temps déterminé et acceptable, mais certains problèmes très très complexes restent irresolvables même avec l'informatique quantique.
- Possibilité de faire des calculs en parallèle en un temps inimaginable
- Consommation très faible d'énergie par rapport à sa puissance de calcul (vs les ordinateurs traditionnels de capacité moindre)
- Faible occupation de surface et de taille d'une machine quantique
- Les opérations sur un ordinateur quantique sont réversibles
- etc.

Les inconvénients:

- Les différentes solutions de l'informatique quantique aujourd'hui ne sont pas encore scalables et miniaturisables pour l'instant
- La durée de calcul (appelée le temps de cohérence) est faible
- La capacité de calcul est encore limitée par rapport au nombre de qubits que l'on est en mesure de fabriquer avec un ordinateur quantique (l'exploitation à plus grande échelle de l'informatique quantique reste à prouver)
- Certains types d'ordinateurs quantiques (ex: supraconducteurs) nécessitent de fonctionner à très basse température (proche du zéro absolu $-273,15^{\circ}\text{C}$)
- Le taux d'erreur est très élevé lors de l'exécution d'un programme quantique, cela nécessite l'exécution de plusieurs fois le même programme et d'intégrer des programmes de correction des erreurs pour s'assurer que le résultat obtenu soit correct.
- Les opérations sur l'ordinateur quantique ne conservent pas d'états, d'où il est indispensable d'utiliser un ordinateur traditionnel combiné à un ordinateur quantique pour l'exploiter correctement et le piloter.
- etc.

3. Les fondamentaux pour comprendre et développer en informatique quantique

Il existe différents types d'ordinateurs quantiques actuellement en R&D:



Selon les types d'ordinateurs, vous aurez souvent des outils associés (langages de programmation, plate-formes de logiciels, etc). Aujourd'hui, seul D-Weave commercialise des ordinateurs quantiques à recuit. Souvent ce sont des solutions complètement intégrées. D'autres fournissent des accès sous formes de services Cloud (AWS Braket, Azure Quantum, IBM Q, ...).

Dans la suite de cet article, je vais explorer un exemple d'IBM sur un modèle de programmation de bas niveau avec le langage QASM. Mais avant cela il faut comprendre les fondamentaux de l'informatique quantique et les compétences nécessaires.

Quels types de profils d'informaticien a-t-on-besoin?

Pour bien comprendre l'informatique quantique, il est indispensable pour l'informaticien d'avoir de bonnes bases:

- en probabilité
- sur les calculs matriciels
- sur les calculs vectoriels avec des nombres complexes
- en trigonométrie
- en algorithmique et programmation au moins dans un langage de l'ordinateur quantique
- en modélisation sur le type de problèmes à résoudre
- assimiler quelques concepts de base en quantique

Voici les concepts de base en informatique quantique.

Qubits:

Quelques soient les types d'ordinateurs quantiques, ils manipulent tous des qubits.

Un qubit est comme le bit dans l'ordinateur traditionnel. La différence est qu'un bit dans un ordinateur traditionnel à 2 états possibles: {0,1}. Sur un ordinateur quantique, un qubit à 3 états possibles: {0, 1, 0 et 1 (en même temps)}. Ce dernier état {0 et 1} est lié à la propriété physique quantique appelé la **superposition**. Dans un environnement quantique bien isolé dans un ordinateur quantique supra-conducteur par exemple refroidi à des températures proche de zéro absolu avec des blindages isolant les perturbations magnétiques, ..., on parvient à créer physiquement cet état quantique de superposition.

Pour imaginer ce phénomène, vous pourriez imaginer une pièce avec les 3 états: pile est l'état 0, face est l'état 1. Puis si vous faites tourner la pièce sur une table elle est pile et face tant que vous ne l'arrêtez pas.

On note alors les états du qubit (selon la notation du physicien Paul Dirac):

$|0\rangle$: l'état 0 (pile par exemple)

$|1\rangle$: l'état 1 (face par exemple)

$|S\rangle$: l'état de superposition 0 et 1 (pile et face en même temps)

Quelques notions de base en quantique:

Le **principe de mesure** d'un qubit, comme dans le phénomène des particules subatomiques : à l'instant où vous mesurez l'état d'un qubit, le phénomène de superposition disparaît. L'état mesuré est soit l'état 0 ou l'état 1. (C'est le fameux principe de la **dualité onde-corpusculaire** en physique quantique).

Pendant toute la durée où le qubit est en état de superposition (la pièce qui tourne), cette durée est appelé le **temps de cohérence**. Dès que vous prenez la mesure de l'état, le temps de cohérence s'arrête (c'est la décohérence). Le calcul dans l'ordinateur quantique ne peut s'opérer que pendant la durée de cohérence. Aujourd'hui les ordinateurs quantiques ne parviennent pas à avoir ce temps de cohérence suffisamment long pour pouvoir réaliser des super calculs complexes, peut-être que ça viendra avec les progrès techniques à venir.

La question est : comment peut-on exploiter le qubit en état de superposition si on ne sait pas le mesurer?

Heureusement Marx Born est là. Il a formulé le théorème suivant:

mesure

$|S\rangle = a|0\rangle + b|1\rangle$ -----> la probabilité d'avoir l'état $|0\rangle$ est $|a|^2$
la probabilité d'avoir l'état $|1\rangle$ est $|b|^2$

avec $|a|^2 + |b|^2 = 1$

On peut relier l'état de superposition d'un qubit après mesure par un calcul de probabilité.

On peut appliquer le même principe de calcul des états pour 2 qubits.

$|S_2\rangle = (a|0\rangle + b|1\rangle)(c|0\rangle + d|1\rangle)$

avec $(a|0\rangle + b|1\rangle)$ pour 1er qubits et $(c|0\rangle + d|1\rangle)$ pour le 2ème qubits
en remarquant avec 2 qubits on a 2^2 états possibles.

$|S_2\rangle = a|0\rangle + b|1\rangle(c|0\rangle + d|1\rangle) = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$

On peut aussi noter sous forme indexé, $a_{00} = ac$, $a_{01} = ad$, $a_{10} = bc$ et $a_{11} = bd$

$|S_2\rangle = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$

Pour n qubits, le théorème de Marx Born devient

$|S_n\rangle = \sum_{k=0}^{2^n-1} a_k |k\rangle$ avec $\sum_{k=0}^{2^n-1} |a_k|^2 = 1$

On peut lier les qubits entre eux en utilisant un autre phénomène quantique appelé **l'intrication quantique** ou enchevetrement ("entanglement" en anglais) :

Imaginons que vous disposiez de 2 pièces de monnaie complètement liées entre elles par le phénomène d'intrication quantique (même si cela n'est pas possible dans la vraie vie puisque le phénomène quantique s'applique qu'aux particules subatomiques électrons et photons).

Vous faites tourner les 2 pièces, l'une à Paris et l'autre à Lyon, en même temps (comme dans le cas de 2 qubits intriqués en superposition).

Du fait que les 2 pièces soient intriquées : au moment où vous arrêtez les deux pièces en même temps (comme si vous mesuriez l'état des 2 qubits), si la pièce à Paris est tombée sur la face Pile (l'état 0) alors la pièce de Lyon aura aussi la face Pile (à l'état 0) et vice versa et de manière instantanée.

Ce phénomène d'intrication est instantané (plus vite que la vitesse de la lumière, ce que Einstein pense impossible).

Cette propriété est souvent utilisée dans la transmission de messages en réseau, la cryptographie, la répartition des calculs sur plusieurs serveurs, ...

Remarques :

Il existe différentes manières de représenter les qubits et de faire les calculs :



- sous forme de combinaisons linéaires (comme décrit dans les exemples ci-dessus)
- sous forme géométrique (la sphère de Bloch)
- sous forme vectorielle (en utilisant les calculs matriciels)
- fonctions trigonométriques

Je ne vais pas décrire tous ces aspects car les calculs seraient trop fastidieux et trop longs pour les novices.

Les portes quantiques:

Les portes quantiques sont des opérateurs utilisés pour le calcul ou des opérations sur les qubits au même titre que les portes logiques dans l'informatique traditionnelle pour les opérations avec des bits.

Exemple de porte logique NOT : elle consiste à transformer un bit 0 en 1 et 1 en 0.
(cf. https://fr.wikipedia.org/wiki/Fonction_logique)

NON			$\neg A$	\overline{A}	<table><tr><th>Entrée</th><th>Sortie</th></tr><tr><td>A</td><td>NON A</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	Entrée	Sortie	A	NON A	0	1	1	0
Entrée	Sortie												
A	NON A												
0	1												
1	0												

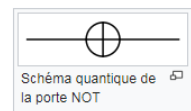
Exemples de porte quantique NOT :

Porte Pauli-X (= porte NOT) [modifier | modifier le code]

La porte Pauli-X agit sur un seul qubit. C'est l'équivalent quantique de la porte NOT pour les ordinateurs classiques (en respectant les standards de base $|0\rangle$, $|1\rangle$, qui privilégie la direction Z (Z-direction)). Cela équivaut à une rotation de la **sphère de Bloch** autour de l'axe X par π radians. Elle transforme $|0\rangle$ en $|1\rangle$ et $|1\rangle$ en $|0\rangle$, c'est pourquoi elle est parfois appelée **bit-flip**. Elle est représentée par la **matrice de Pauli**:

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

$NOT\ NOT = I$ où I est la matrice identité



(source : cf https://fr.wikipedia.org/wiki/Porte_quantique)

Processus de calcul :

données entrantes → traitement de données (calcul) → analyse des données résultats sortant

Le calcul quantique :

qubits entrants → opérations sur les qubits en appliquant les portes quantiques → mesure qubits résultats sortant

Réaliser un calcul sur des qubits consiste tout simplement à lui appliquer un certain nombre d'opérations sur les qubits en utilisant les portes quantiques et à mesurer le résultat sortant. Une fois le calcul effectué, le résultat doit être stocké en mémoire ou sur disque.

Remarque :

Contenu du fait qu'un ordinateur quantique pur ne sait pas conserver un état dès qu'on prend la mesure, un ordinateur traditionnel est souvent utilisé avec un ordinateur quantique pour le piloter (activer les portes quantiques) et sauvegarder les résultats issus des calculs (en mémoire ou sur disque dur).

En combinant les portes quantiques, vous pourriez réaliser un certain nombre d'opérations :

- initialiser les qubits à zéro,
- créer les états de superposition en utilisant la porte Hadamard,
- créer des conditions Si Alors avec la porte CNOT
- créer des intrications qubits avec les portes Hadamard, CNOT
- permuter l'état des qubits avec une porte SWAP

- faire des additions en combinant des portes CNOT
- mesurer les résultats avec les portes de Mesures
- etc.

La diversité des portes quantiques dépend des fournisseurs, des outils de conception, ...

Algorithme quantique :

Pour concevoir un algorithme quantique, on peut utiliser un modèle de circuits quantiques qui se schématise comme dans le dessin ci-dessous:

(cf. source : p45 Radovanovic, Aleksandar. *Quantum Programming Illustrated*. Édition du Kindle.)

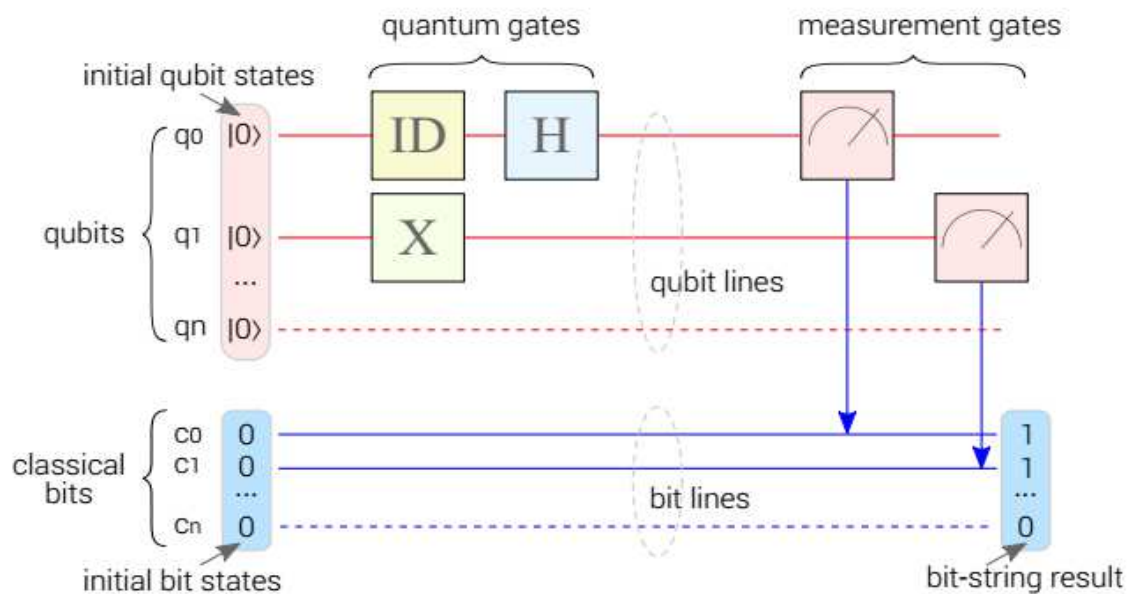
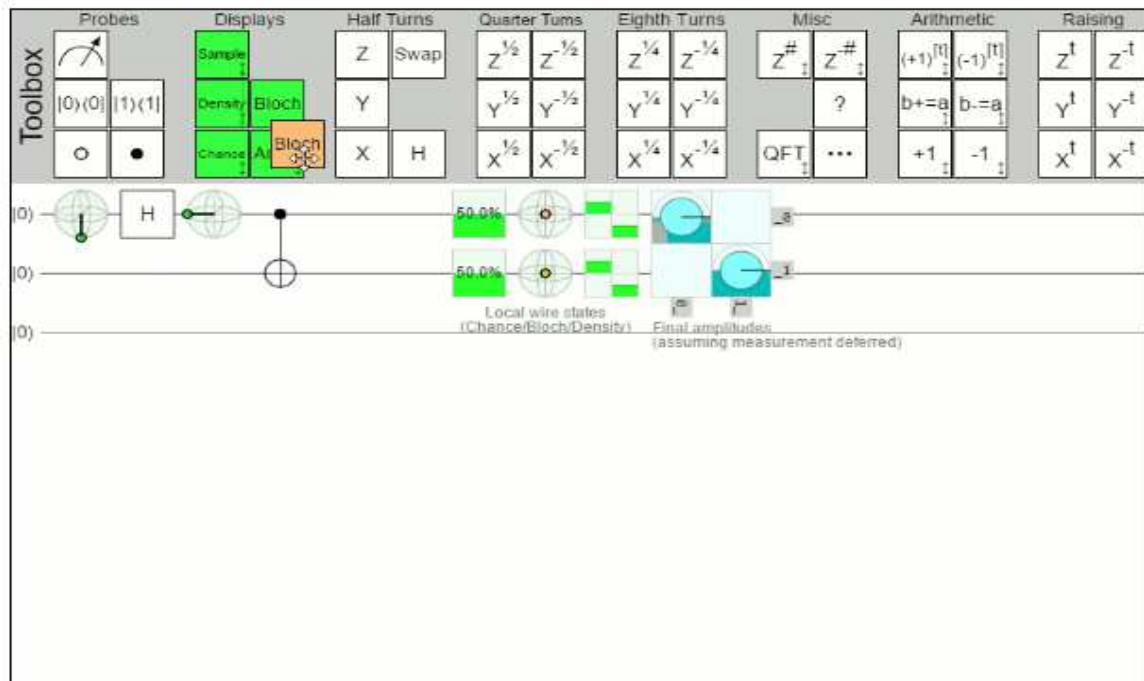


Figure 3-3. Quantum circuit model of computation.

A gauche vous avez des n qubits en entrée initialisé à l'état $|0\rangle$, vous lui faites appliquer des opérations de calculs avec les portes quantiques de gauche à droite dans une séquence temporelle. Vous mesurez les résultats grâce aux portes mesures, qui stockent les données sous forme de bits classiques.

Il existe des logiciels pour faire des simulations extrêmement pratiques dans la phase de conception : <https://algassert.com/2016/05/22/quirk.html>

Ce site présente plusieurs outils disponibles et leur comparaison



4. Programmation quantique

L'objectif n'est pas de faire un algorithme complexe mais de montrer comment on met en œuvre avec une plate-forme quantique.

Dans cet exemple, j'utilise le service IBM Quantum Experience. Il propose la possibilité d'utiliser un simulateur d'ordinateur quantique pour écrire des programmes quantiques.

Le pré-requis est d'avoir créé un compte IBM.

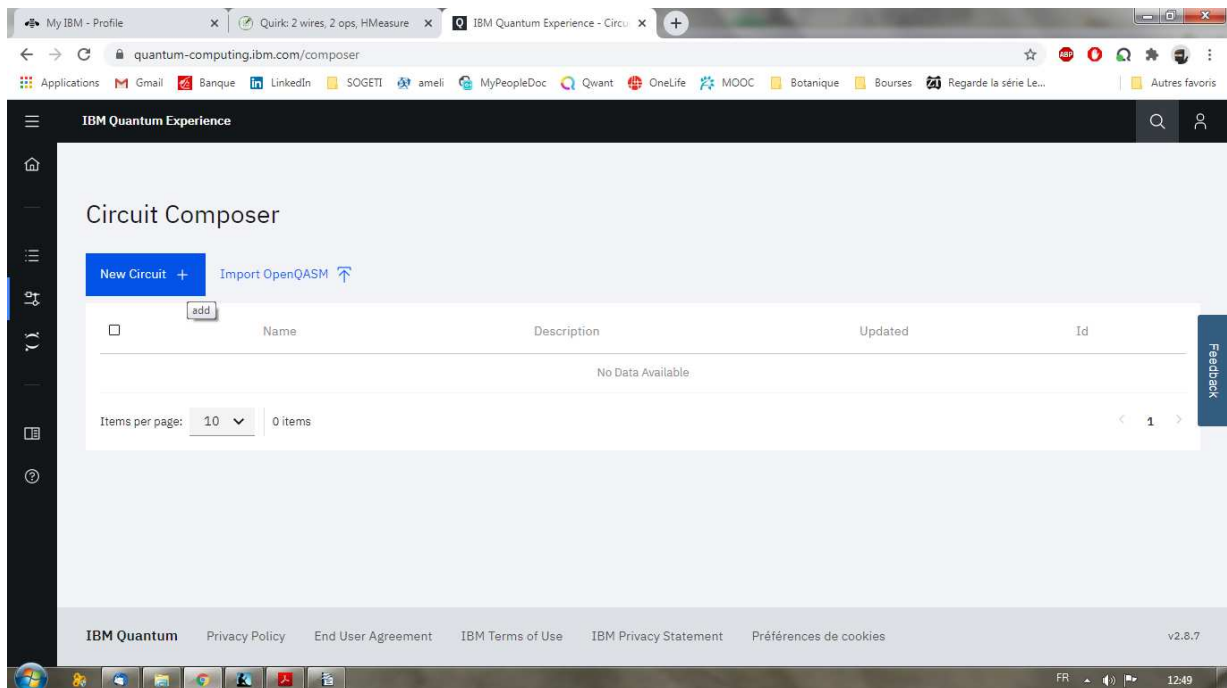
Ensuite utiliser l'outil de conception d'IBM Circuit Composer pour modéliser les opérations quantiques.

Puis on regarde le code QASM généré.

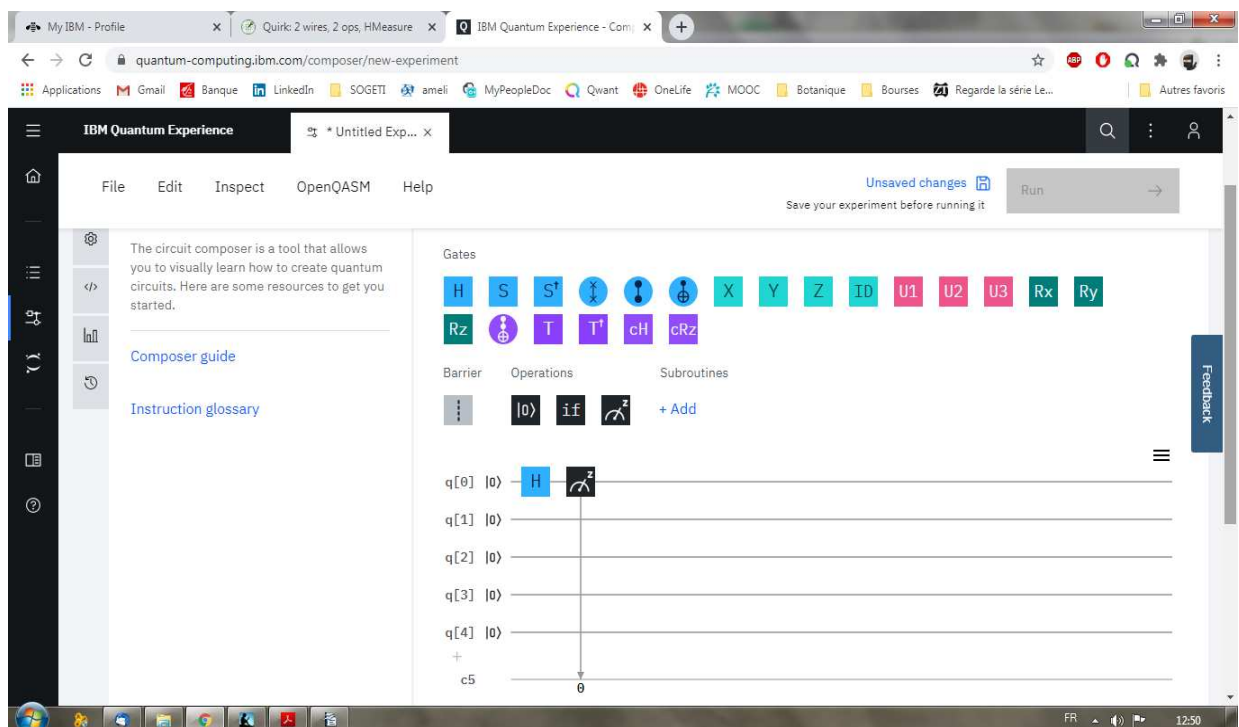
Un exemple très très simple de programmation quantique :

Ce premier programme consiste tout simplement à créer une superposition de qubit et de mesurer son état sortant.

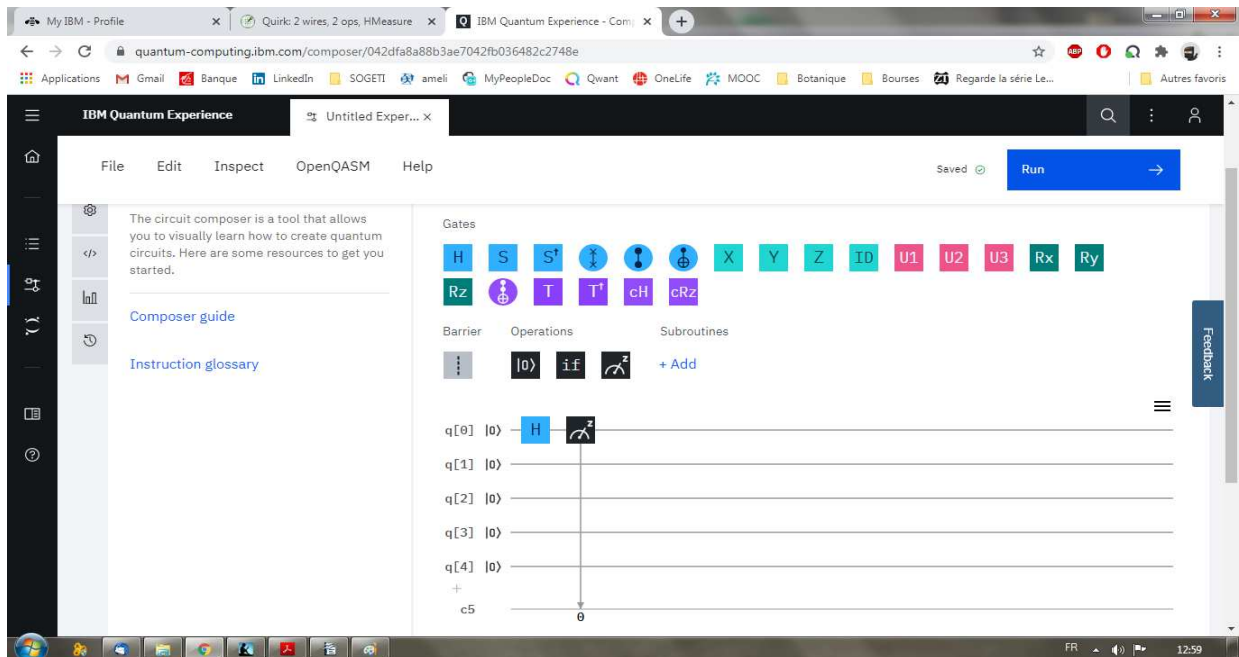
Allez sur **Circuit Composer** et créer un nouveau circuit :



Faites glisser les 2 portes quantiques qui nous intéressent : Hadamard et Mesure sur le qubit. Remarquez que le résultat est stocké sur un bit classique.

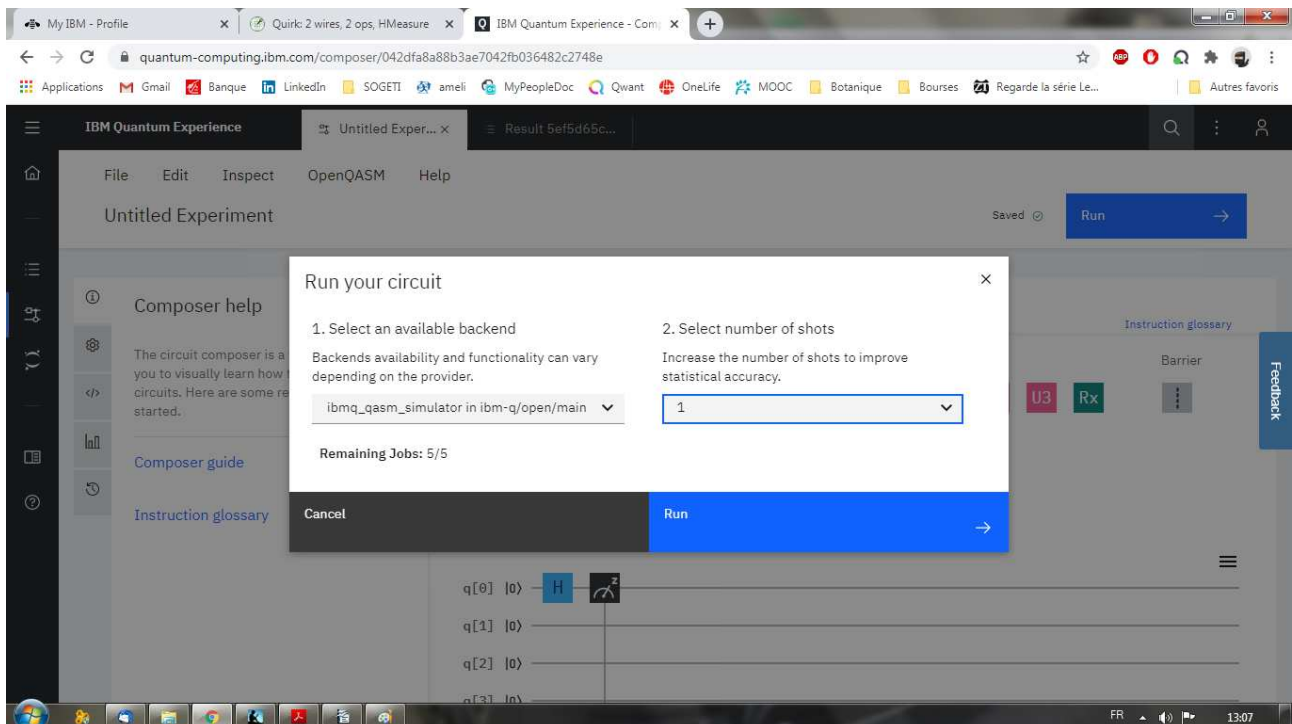


Pour exécuter le programme, il faut d'abord sauvegarder en cliquant sur l'icône disquette en haut à droite. Ensuite le bouton à droite de l'icône Run devient bleu.

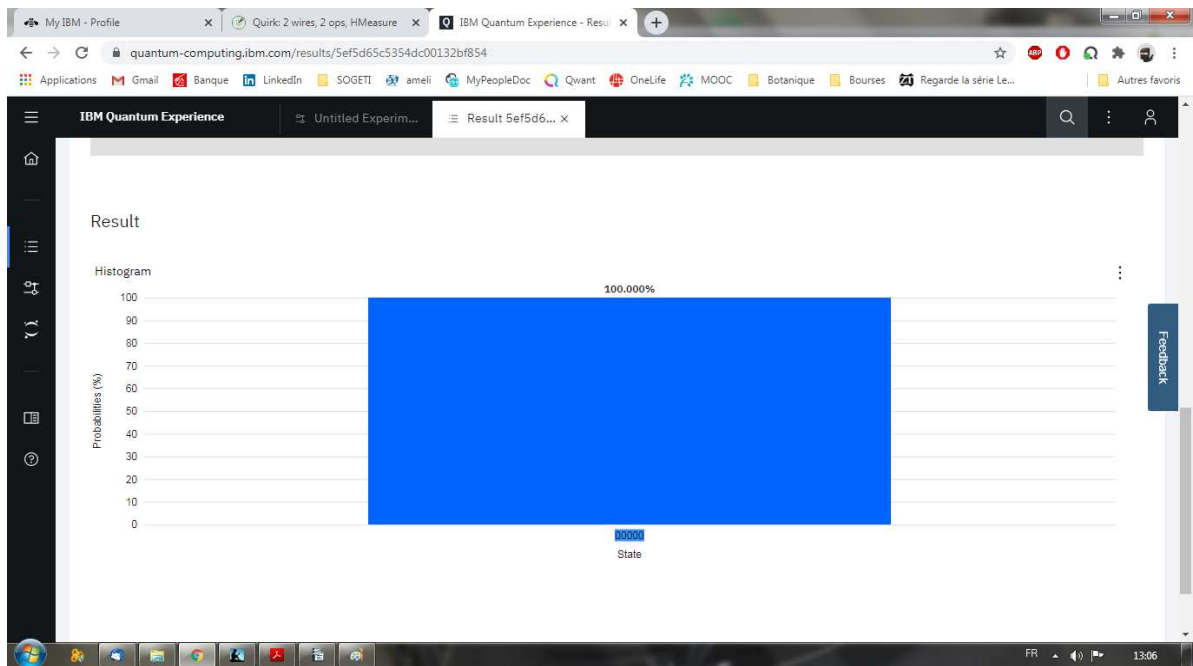


En cliquant Run, vous sélectionnez quels sont les serveurs disponibles à gauche et le nombre d'exécution pour valider votre résultat.

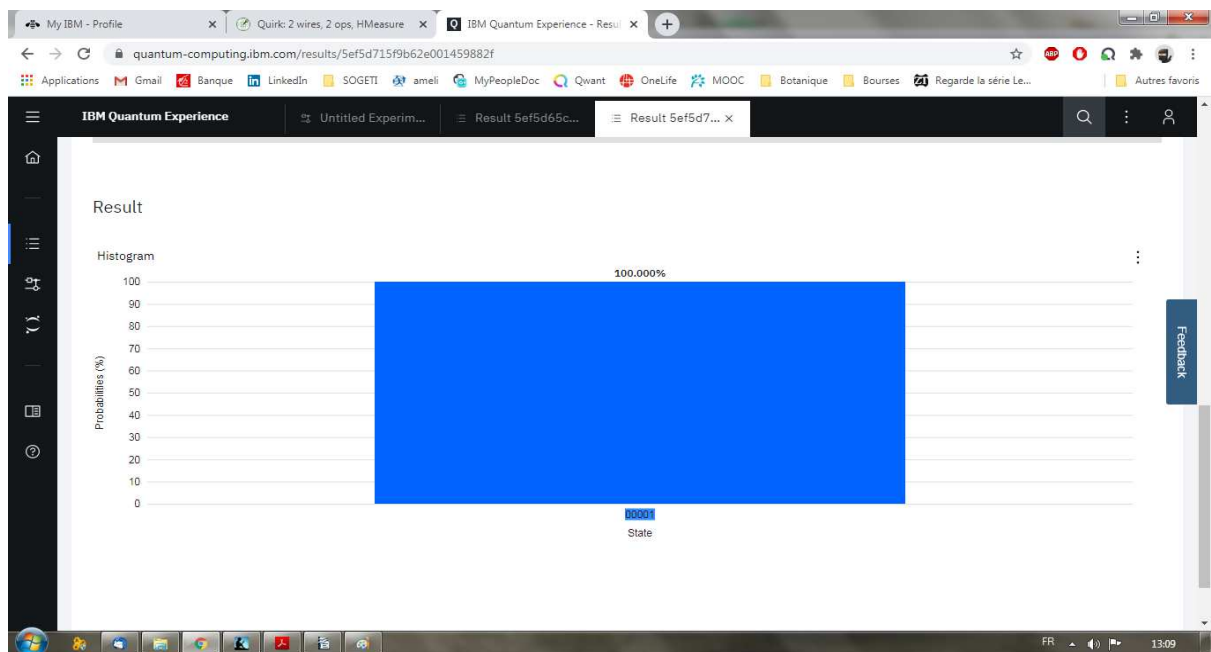
Choisissons le nombre d'exécution 1



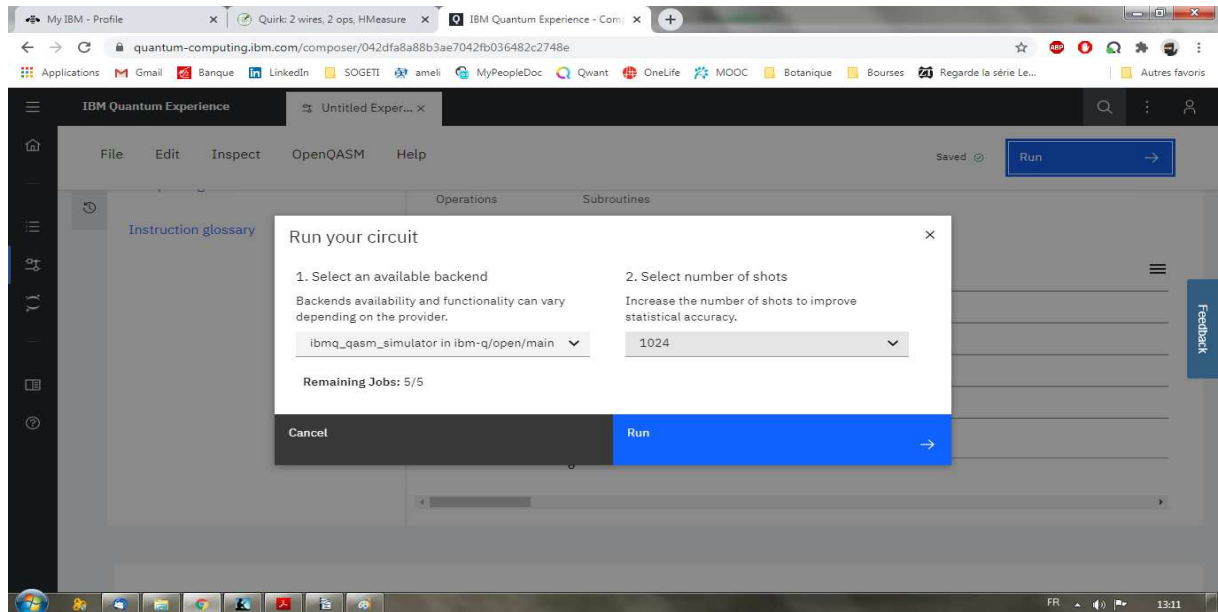
Résultat : on a 100% valeur 0000. Le résultat semble ne pas être conforme aux attentes. L'état du qubit en superposition doit être 0,5 soit 50% de chance d'avoir 0000 et 50% de chance d'avoir 0001 or on a 100% à 0000. (cf la vérification dans les paragraphes suivants)



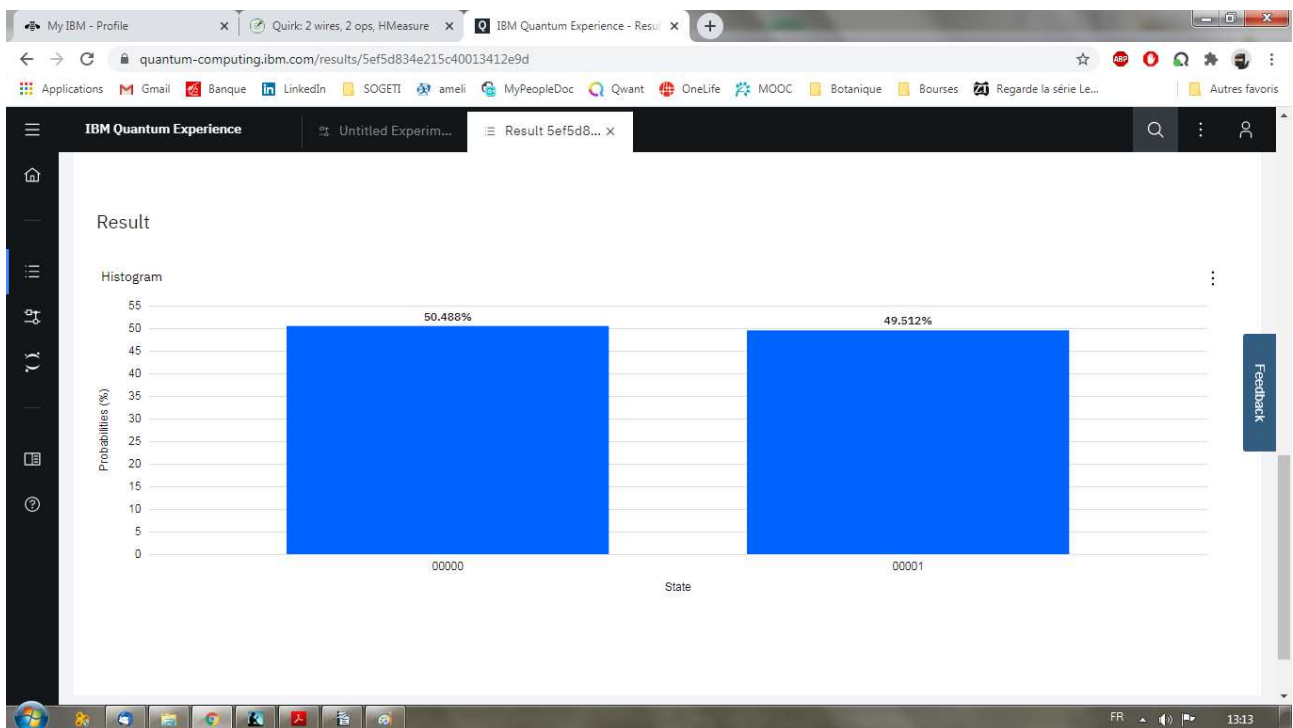
On réitère l'opération une 2ème fois, on a 100% valeur 0001. Idem, le résultat semble ne pas être conforme aux attentes.



Maintenant, on choisit une exécution de 1024 fois :



On obtient le résultat suivant : 50,488% 0000 et 49,512% 0001



Comme l'informatique quantique est basé sur un modèle probabiliste (non déterministe), il est indispensable d'exécuter plusieurs fois le programme pour valider les résultats.

Cette fois le resultat est conforme.

Vérification :

L'état en sortie d'un qubit en superposition doit être selon le théorème de Born :

mesure

$|S\rangle = a|0\rangle + b|1\rangle$ -----> la probabilité d'avoir l'état $|0\rangle$ est $|a|^2$

la probabilité d'avoir l'état $|1\rangle$ est $|b|^2$

avec $|a|^2 + |b|^2 = 1$

Déterminons les valeurs de a et b

si $a=1/\sqrt{2}$ et $b=1/\sqrt{2}$ alors

$|S\rangle = 1/\sqrt{2} |0\rangle + 1/\sqrt{2} |1\rangle$ avec $|1/\sqrt{2}|^2 + |1/\sqrt{2}|^2 = 1/2 + 1/2 = 1$

La probabilité d'avoir l'état 0 est : $P(|0\rangle) = |a|^2 = |1/\sqrt{2}|^2 = 1/2$

Et la probabilité d'avoir l'état 1 est : $P(|1\rangle) = |b|^2 = |1/\sqrt{2}|^2 = 1/2$

dans le cas d'un ordinateur quantique théorique.

Le résultat obtenu après 1024 exécutions est approximativement de :

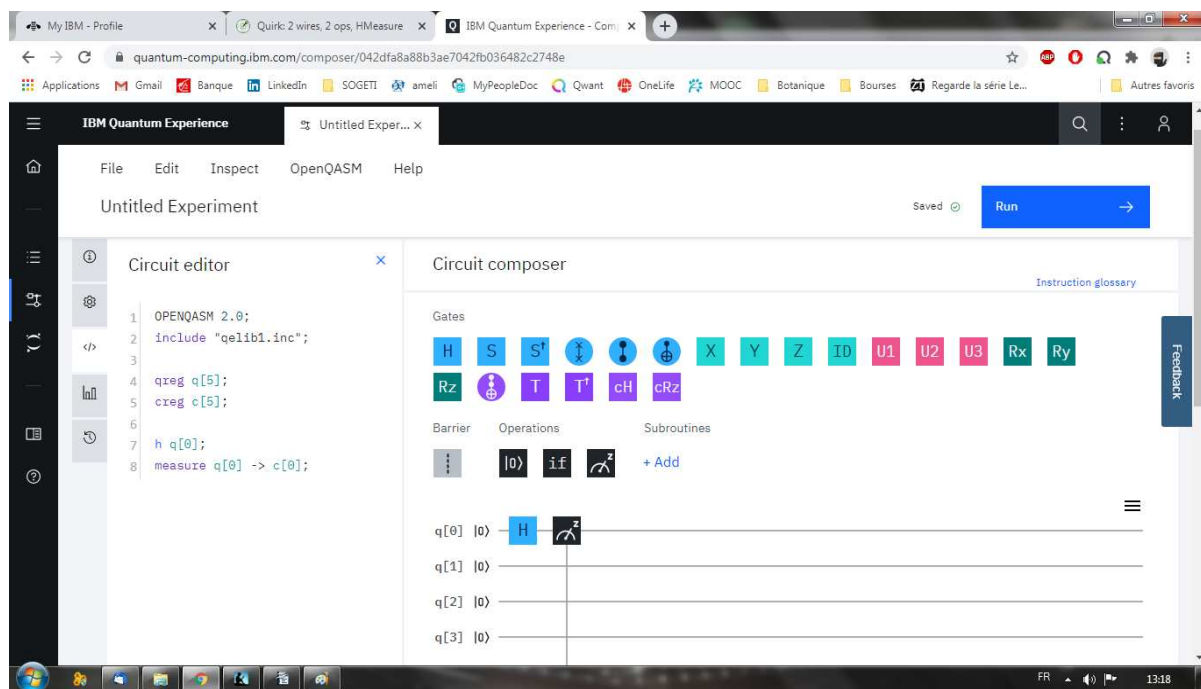
$P(|0\rangle) \approx 50,488\%$

$P(|1\rangle) \approx 49,512\%$

D'où il existe des programmes de correction d'erreur pour rendre les résultats fiables. Certains fournisseurs les intègrent dans leur solution.

Maintenant, nous allons explorer le code.

Pour regarder le code généré, il suffit de cliquer sur l'icône `</>` sur le menu à gauche :



Le code s'affiche sur le Circuit editor sous forme de texte.

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

qreg q[5];
creg c[5];

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

Description du programme :

La première ligne indique la version du langage QASM.

La deuxième ligne la librairie à utiliser pour ce programme.

qreg q[5]; déclare un registre avec 5 qubits.

creg c[5]; déclare un registre avec 5 bits classiques

h q[0]; applique la porte Hadamard sur le qubit q[0]

measure q[0] -> c[0]; mesure le qubit q[0] et sauvegarde du résultat dans un bit classiques c[0].

5. D'autres explorations en informatique quantique à voir

Pour ce premier article, j'ai voulu focaliser sur la compréhension générale et simpliste de ce qu'est l'informatique quantique. Mais pour approfondir sur l'élaboration des algorithmes quantiques, il faut entrer plus en profondeur sur les calculs mathématiques, sur les matrices de transformation, sur les vecteurs avec les nombres complexes. En effet, les qubits peuvent être mathématiquement représentés par des vecteurs dans l'espace de Hilbert ce qui lui confère certaines propriétés qui va faciliter les calculs matriciels. On aurait l'occasion d'aller plus loin dans les prochains articles.

Évidemment, je souhaiterais vous exposer d'autres approches que celle d'IBM lors de prochaines présentations et aussi d'autres langages et plate-formes.

J'espère que cet article vous a permis d'approcher les concepts de l'informatique quantique.