Informatique Quantique – Cours 1

Nicolas Boutry[1]

[1] Laboratoire de Recherche et Développement de l'EPITA (LRDE), France

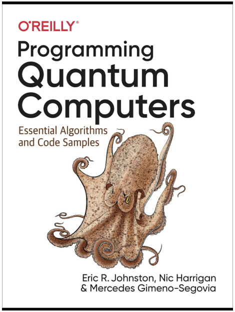
Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

Tout au long de ce cours, nous allons suivre en grande partie le plan du livre intitulé Programming Quantum Computers de Johnston *et al.* de la collection O'Reilly.



Un schéma de programme ressemble généralement à ceci :

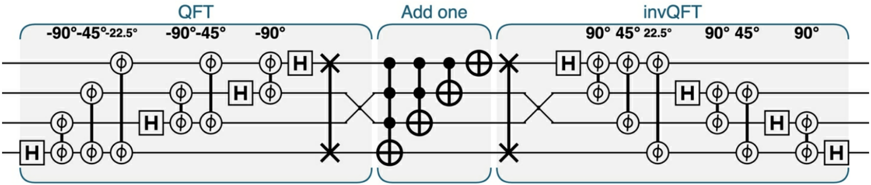


Figure P-1. Quantum programs can look a bit like sheet music

Remarquez les lignes correspondant aux qubits, les opérateurs reliant les lignes entre elles, et la symétrie dans ce schéma (tout ceci sera étudié plus tard).

Pour être plus précis, le programme ci-dessus est une suite de 3 primitives: transformée de Fourier quantique, ajout de la valeur 1, puis transformée de Fourier inverse.

Des programmes d'I.Q. seront exécutables sur leur site tout au long de ce cours à l'URL suivante: `https://oreilly-qc.github.io/` (si l'URL bug, chercher sur Google à "O'Reilly QCEngine").

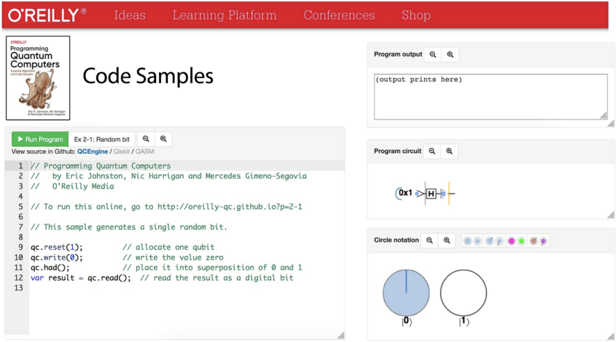


Figure 1-1. The QCEngine UI

La documentation est accessible sur : `https://oreilly-qc.github.io/docs/build/index.html`

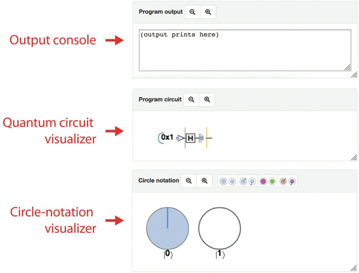


Figure 1-2. QCEngine UI elements for visualizing QPU results

Important, en mono-registre (un seul qubit), le qubit est représenté par deux disques $|0>$ et $|1>$, chaque disque représente un état. Ici, $|0\rangle$ est plein, et $|1\rangle$ est vide, on est donc dans l'état $|0\rangle$.

Via QCEngine, on peut faire défiler des programmes avec des registres à plusieurs qubits:

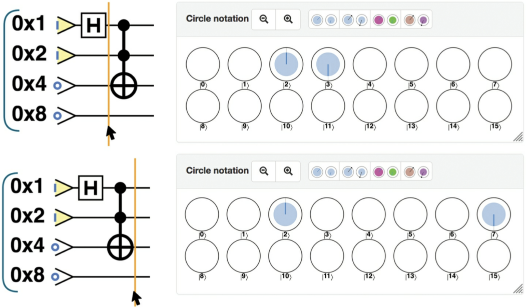
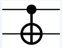
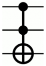


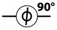
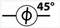
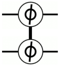
Figure 1-3. Stepping through a QCEngine program using the circuit and circle-notation visualizers

La barre jaune verticale représente l'instant d'exécution.
Attention : Si vous avez remarqué que l'on à 4 qubits $0x1$, ..., $0x8$, c'est bien, mais avez-vous remarqué les 16 états dessinnés sur la droite ? Pourquoi n'en a-t-on pas que 8 ? Ceci est lié à l'intrication ou entanglement en Anglais, on en parlera bientôt.

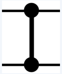
Voici les portes quantiques que nous allons voir tout au long du cours, remarquez juste que soit elles s'appliquent à un seul qubit (une ligne les traverse), soit elles s'appliquent a plusieurs qubits (et peuvent créer de l'intrication !).

Table 1-1. Essential QPU instruction set

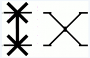
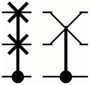
Symbol	Name	Usage	Description
	NOT (also x)	`qc.not(t)`	Logical bitwise NOT
	CNOT	`qc.cnot(t,c)`	Controlled NOT: `if (c) then NOT(t)`
	CCNOT (Toffoli)	`qc.cnot(t,c1|c2)`	`if (c1 AND c2) then NOT(t)`
	HAD (Hadamard)	`qc.had(t)`	Hadamard gate
	PHASE	`qc.phase(angle,c)`	Relative phase rotation

	Z	`qc.phase(180,c)`	Relative phase rotation by 180 °
	S	`qc.phase(90,c)`	Relative phase rotation by 90 °
	T	`qc.phase(45,c)`	Relative phase rotation by 45 °
	CPHASE	`qc.cphase(angle,c1|c2)`	Conditional phase rotation

Notez le petit "c" dans le nom de certaines portes (comme CPHASE), il signifie conditionnel, c'est-à-dire que selon un bit de contrôle/d'entrée, il appliquera une certaine opération sur le bit contrôlé.

	CZ	`qc.cphase(180,c1|c2)`	Conditional phase rotation by 180 °
	READ	`val = qc.read(t)`	Read qubits, returning digital data
	WRITE	`qc.write(t,val)`	Write conventional data to qubits
	ROOTNOT	`qc.rootnot(t)`	Root-of-NOT operation

On verra qu'il peut arriver que des portes puissent être reconstruites à partir de combinaisons d'autres portes (comme la porte racine carrée que l'on verra plus tard).

	SWAP (EXCHANGE)	`qc.exchange(t1|t2)`	Exchange two qubits
	CSWAP	`qc.exchange(t1|t2, c)`	Conditional exchange: `if (c) then SWAP(t1,t2)`

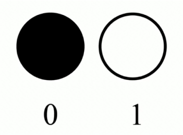
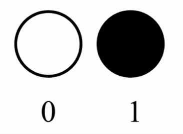
On verra aussi que plusieurs notations peuvent être utilisées et donc choisies selon les commodités.

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

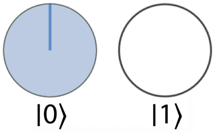
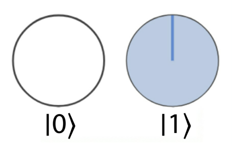
Revenons-en deux secondes au bits logiques : chaque bit est soit 0 soit a 1, de qui peut se représenter ainsi :

Table 2-1. Possible values of a conventional bit — a graphical representation

Possible values of a bit	Graphical representation
0	
1	

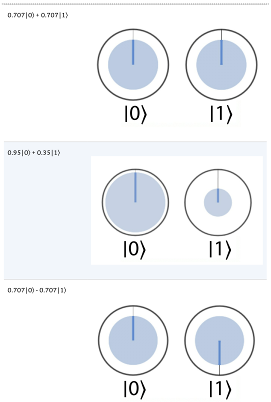
Repassons en quantique (observez les bra-kets): le bit du haut est à 0, celui du bas a 1 (bleu = rempli). Notez la barre verticale, elle montre la phase relative entre les deux états (on la redétaille plus tard).

Table 2-2. Some possible values of a qubit

Possible values of a qubit	Graphical representation
$\|0\rangle$	
$\|1\rangle$	

Introduisons maintenant la superposition d'états, elle représente que l'on n'a plus soit 0 soit 1, mais potentielement les deux en même temps !

L'aire de la surface en bleu correspond a l'amplitude, c'est à dire "à quel point" on aura l'état en question.



Quand les deux barres (une par état du qubit) sont alignées, c'est que la phase relative est de 0 degré. On a donc dans ces trois exemples: 0, 0, 180 degrés respectivement.

Comment passer de notre mode de pensée ”logique” a un mode de pensée quantique ? Observez le cas ”logique” sur ce schéma, les miroirs réfléchissent le photon de façon nulle ou totale :

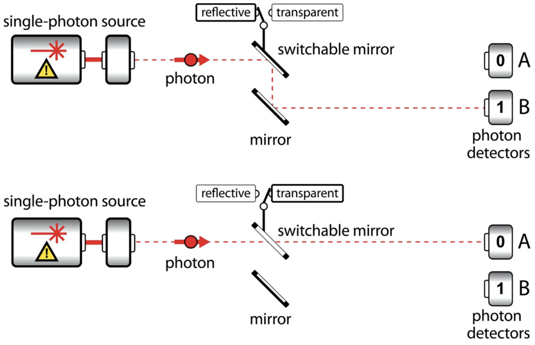


Figure 2-1. Using a photon as a conventional bit

Imaginons maintenant que la lumière passe "partiellement" (on parle de biréfringence), on peut avoir le schéma suivant:

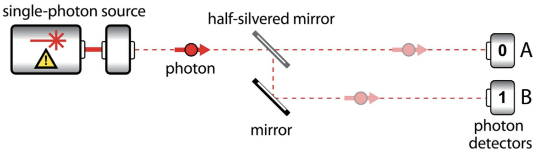


Figure 2-2. A simple implementation of one photonic qubit

Le photon n'est pas "coupé" en deux, ça serait physiquement faux, c'est en fait que la probabilité de **LIRE** un 0 et celle d'avoir un 1 passent a un demi.

Attention : on ne peut pas dire non plus que le photon passe soit par un chemin soit par l'autre, en quelque sorte il passe par les deux, mais surtout, il faut le voir en tant que probabilité, non pas parce que on ne sait pas où il est, mais parce qu'en quantique, on **pense** en probabilités. C'est une réalité physique.

Observez ces quelques exemples d'états et de phases (relatives) :

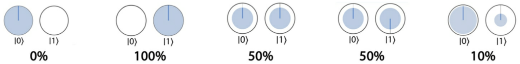


Figure 2-3. Probability of reading the value 1 for different superpositions represented in circle notation

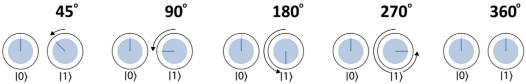


Figure 2-4. Example relative phases in a single qubit

Quand on dit que c’est la phase relative qui compte, cela signifie donc que il y a des paires d’états rigoureusement équivalents.

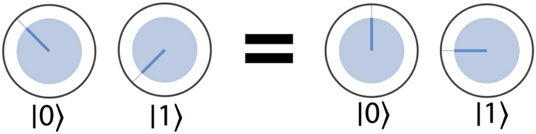


Figure 2-5. Only relative rotations matter in circle notation — these two states are equivalent because the relative phase of the two circles is the same in each case

Outline

1 Introduction

2 Programming for a QPU

3 Les portes quantiques à un seul qubit

4 Détection d'espion par informatique quantique

5 Registres multi-qubits

6 Opérations multi-qubits

7 Opérateurs multiqubits conditionnels
- Porte CNOT
- Paire de Bell
- Porte quantique de changement de phase conditionnelle
- Phase kickback
- Porte de Toffoli
- Portes Swap et CSwap

8 Le Swap test

9 Téléportation quantique

QPU Instruction: NOT

NOT is the quantum equivalent of the eponymous conventional operation. Zero becomes one, and vice versa. However, unlike its traditional cousin, a QPU NOT operation can also operate on a qubit in superposition.

In circle notation this results, very simply, in the swapping of the $|0\rangle$ and $|1\rangle$ circles, as in Figure 2-6.

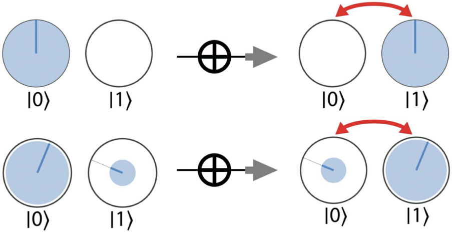


Figure 2-6. The NOT operation in circle notation

Reversibility: Just as in digital logic, the NOT operation is its own inverse; applying it twice returns a qubit to its original value.

QPU Instruction: HAD

The `HAD` operation (short for Hadamard) essentially creates an equal superposition when presented with either a $|0\rangle$ or $|1\rangle$ state. This is our gateway drug into using the bizarre and delicate parallelism of quantum superposition! Unlike `NOT`, it has no conventional equivalent.

In circle notation, `HAD` results in the output qubit having the same amount of area filled-in for both $|0\rangle$ and $|1\rangle$, as in Figure 2-7.

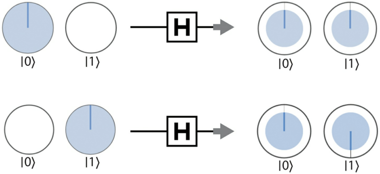


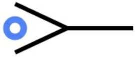
Figure 2-7. Hadamard applied to some basic states

This allows `HAD` to produce *uniform superpositions* of outcomes in a qubit; i.e., a superposition where each outcome is equally likely. Notice also that `HAD`'s action on qubits initially in the states $|0\rangle$ and $|1\rangle$ is slightly different: the output of acting `HAD` on $|1\rangle$ yields a nonzero rotation (relative phase) of one of the circles, whereas the output from acting it on $|0\rangle$ doesn't.

QPU Instruction: READ

The READ operation is the formal expression of the previously introduced *readout* process. READ is unique in being the only part of a QPU's instruction set that potentially returns a *random* result.

QPU Instruction: WRITE



The WRITE operation allows us to initialize a QPU register before we operate on it. This is a deterministic process.

Attention : Quand on lit un bit quantique, on le détruit en le transformant en bit logique.
Note : On écrit évidemment un bit logique, ici 0, qui devient quantique (cf. la notation en cercles).

Quand le bleu remplit 50 % de l'aire des disques, on a autant de chances de lire un 0 qu'un 1 (système aléatoire parfait).

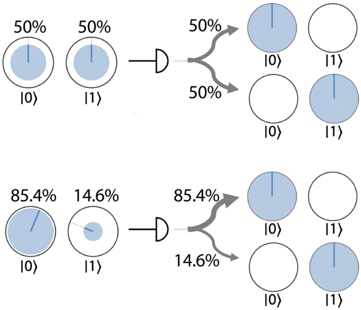


Figure 2-8. The READ operation produces random results

On raisonne en aire car c'est le carré de l'amplitude, et que la somme des carrés des amplitudes fait 1 (voir cours d'Edouard).

Voici le ”Hello World” quantique, le programme le plus simple qui existe en info quantique, il tire profit de l’aspect parfaitement aléatoire en générant aléatoirement des 0 et des 1 à chaque exécution.

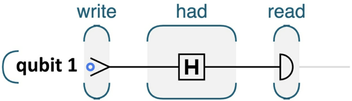


Figure 2-9. Generating a perfectly random bit with a QPU

SAMPLE CODE

Run this sample online at http://oreilly-qc.github.io?p=2-1.

Example 2-1. One random bit

```
qc.reset(1);          // allocate one qubit
qc.write(0);          // write the value zero
qc.had();             // place it into superposition of 0 and 1
var result = qc.read();  // read the result as a digital bit
```

Vous pouvez l’exécuter sur le QCEngine (mais ça ne sera qu’une simulation d’ordinateur quantique, donc l’aléatoire ne sera pas si parfait).

Et voici comment générer un octet aléatoire.

Remarquer qu'on peut aussi simplement exécuter 8 fois le programme précédent avec le même bit.

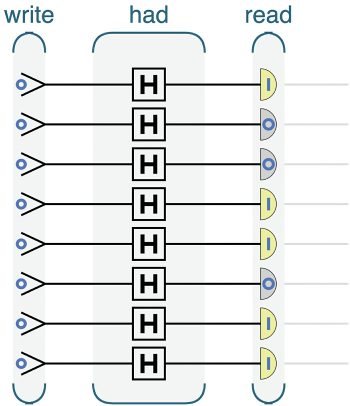


Figure 2-10. Generating one random byte

Voici la porte Phase:

QPU Instruction: PHASE(θ)

The `PHASE(`θ`)` operation also has no conventional equivalent. This instruction allows us to directly manipulate the *relative phase* of a qubit, changing it by some specified angle. Consequently, as well as a qubit to operate on, the `PHASE(`θ`)` operation takes an additional (numerical) input parameter — the angle to rotate by. For example, `PHASE(45)` denotes a `PHASE` operation that performs a 45° rotation.

In circle notation, the effect of `PHASE(`θ`)` is to simply rotate the circle associated with $|1\rangle$ by the angle we specify. This is shown in Figure 2-11 for the case of `PHASE(45)`.

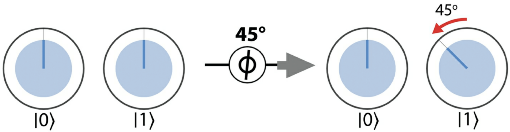


Figure 2-11. Action of a PHASE(45) operation

Bien se rappeler que c'est l'état $|1\rangle$ que l'on tourne !

Pour des questions de commodités, certains états récurrents en informatique quantique ont été nommés ainsi.

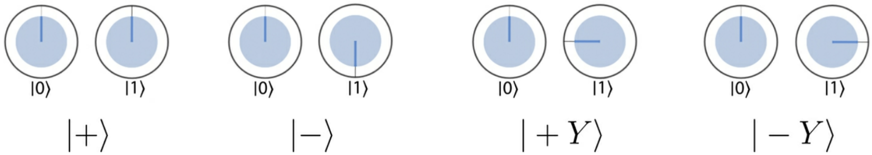


Figure 2-12. Four very commonly used single-qubit states

Certains opérateurs que l'on retrouve moins souvent sont néanmoins a connaître:

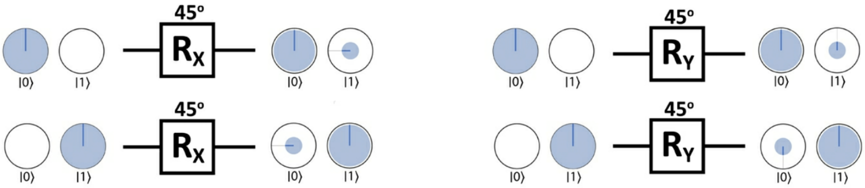


Figure 2-13. ROTX and ROTY actions on 0 and 1 input states

Pour information, l'opérateur Phase est aussi parfois appelé ROTZ.

Voici comment décomposer certaines opérations classiques en d'autres opérations classiques:

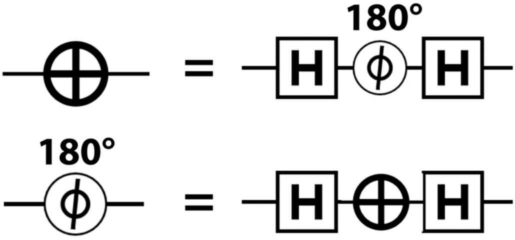
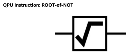


Figure 2-14. Building equivalent operations

Remarquer que comme le HAD est son propre inverse, on retrouve la ligne 1 à partir de la ligne 2, et inversement.

Voici un opérateur très anti-intuitif au départ, l'opérateur racine: il a été appelé ainsi car quand on l'applique deux fois de suite, on retrouve l'opérateur NOT (cette dénomination est discutable mais il faudra faire avec).



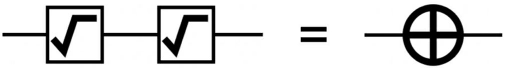


Figure 2-15. An impossible operation for conventional bits

There's more than one way to construct this operation, but Figure 2-16 shows one simple implementation.

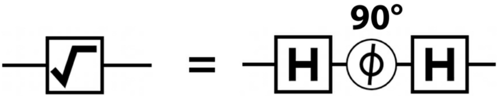


Figure 2-16. Recipe for ROOT-of-NOT

Elle se construit par la séquence vue ci-dessus, ce qui se justifie facilement.

Voici deux exemples d'application de la porte quantique ROOT, où on voit qu'en l'appliquant deux fois, on retrouve l'opérateur NOT.

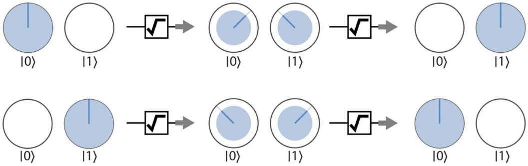


Figure 2-17. Function of the ROOT-of-NOT operation

Il existe aussi l'inverse de l'opérateur ROOT, noté astucieusement par l'opérateur à l'envers.

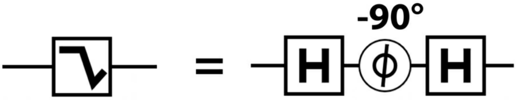


Figure 2-18. Inverse of RNOT

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

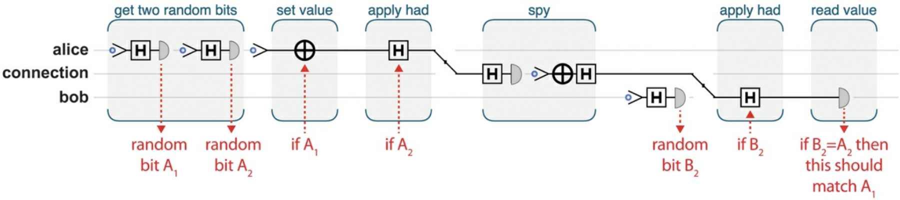


Figure 2-19. The quantum spy hunter program

Il s'agit de la version schématisée avec des portes quantiques de Alice et Bob qui voudraient se transmettre un message sur une ligne à 1 qubit (cf **connection**).

Détails ci-dessous.

Le bit random A_1 à 1 signifie "Alice transmets un 1", sinon elle transmet un 0. Le bit random A_2 signifie "Alice applique un HAD", sinon elle ne l'applique pas (équivalent du changement de polarisation du cours d'Edouard).

L'espion, par défaut, applique un HAD, lit, il lit la valeur (donc la détruit par la même occasion), réécrit la valeur lue, et ré-applique le HAD. Ainsi, dans le cas où il aurait bien choisi la polarisation et donc aurait bien lu le qubit, il n'est pas intercepté.

Bob fait comme Alice avec ses deux bits randoms puis lit le qubit résultant.

Au final ils comparent leurs polarisations (HAD ou non) et les qubits qui ont été lus, et peuvent en déduire si ils ont été espionnés avec une grande fiabilité.

En effet, tout espion qui lira un qubit aura 25 pourcents de chance d'être attrapé. Autrement dit, si Alice et Bob mettent 50 qubits de détection d'espion, alors il y aura moins d'une chance sur un million que l'espion ne soit pas détecté.

Le programme en question est disponible sur le QCEngine si vous voulez faire des simulations.

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

- ”un qubit” veut dire 2 états possibles,
- ”deux qubits” veut dire $00 = |0\rangle$, $01 = |1\rangle$, $10 = |2\rangle$, et $11 = |3\rangle$
- ...
- donc N qubits donnent 2^N états possibles.

En informatique classique, on aurait ”1 contexte avec N bits” = ”1 seul état activé correspondant à une combinaison particulière de ces états”. Or en informatique quantique, tous les états peuvent être actifs en même temps (superposition d'états).

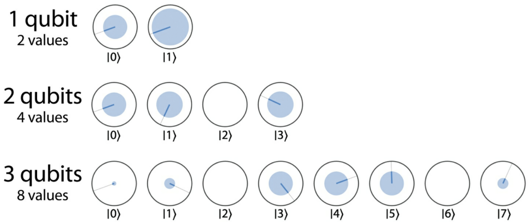


Figure 3-1. Circle notation for various numbers of qubits

On a donc bien $2^3 = 8$ états activés en même temps avec seulement 3 qubits (et non pas 6 cercles !).

En réalité, il y a deux représentations possibles !

Si on a N qubits, on a $2N$ cercles (représentation séparée, suppose qu'il n'y a pas d'intrication entre les qubits), et 2^N états (représentation groupée).

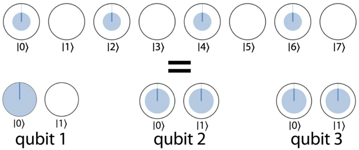


Figure 3-2. Some multi-qubit quantum states can be understood in terms of single-qubit states

Indice de comment passer d'une représentation à l'autre (quand c'est possible). Prenons la représentation groupée (ligne du haut), les états $|0\rangle$, $|2\rangle$, $|4\rangle$, $|6\rangle$, c'est-à-dire $|000\rangle$, $|010\rangle$, $|100\rangle$, et $|110\rangle$, donc tous les états de la forme $XX0$ activés, donc le premier qubit est à 0. Les autres, les $XX1$, sont tous à la même valeur non nulle, donc autant de chances d'avoir 0 ou 1 pour chaque état des qubits 2 et 3.

Attention, cela n'est pas toujours possible d'appliquer un tel raisonnement, on ne peut pas représenter les qubits séparément quand il y a intrication (les qubits sont "liés" entre eux par un phénomène quantique).

L'opération pour passer de la représentation séparée à la représentation groupée est toujours possible.

Exemple de cas intriqué, où on ne peut donc pas représenter autrement l'état du registre multi-qubits.

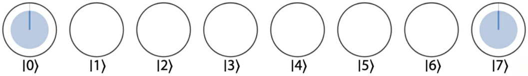


Figure 3-3. Quantum relationships between multiple qubits

This represents a state of three qubits in equal superposition of $|0\rangle$ and $|7\rangle$. Can we visualize this in terms of what each individual qubit is doing like we could in Figure 3-2? Since 0 and 7 are 000 and 111 in binary, we have a superposition of the three qubits being in the states $|0\rangle|0\rangle|0\rangle$ and $|1\rangle|1\rangle|1\rangle$. Surprisingly, in this case, there is no way to write down circle representations for the individual qubits! Notice that reading out the three qubits always results in us finding them to have the *same* values (with 50% probability that the value will be 0 and 50% probability it will be 1). So clearly there must be some kind of link between the three qubits, ensuring that their outcomes are the same.

This link is the new and powerful *entanglement* phenomenon. Entangled multi-qubit states cannot be described in terms of individual descriptions of what the constituent qubits are doing, although you're welcome to try! This entanglement link is only describable in the configuration of the *whole* multi-qubit register. It also turns out to be impossible to produce entangled states from only *single-qubit* operations. To explore entanglement in more detail, we'll need to introduce multi-qubit operations.

Schéma d'un registre multi qubits: on écrit en hexadécimal, d'où le ”x”, et on multiplie par 2 a chaque fois qu'on passe au qubit suivant (on a donc jamais de ”F” dans la valeur du qubit même si il existe bien en hexadécimal, du fait que l'on travaille avec les puissances de 2 !).

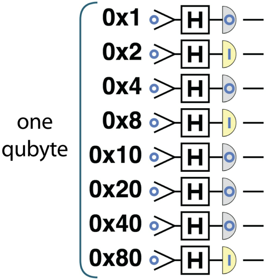


Figure 3-4. Labeling qubits in a qubyte

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

Imaginons un registre de 3 qubits, donc 8 états en représentation groupée. La logique de l'opérateur NOT sur ce registre est décrite ci-dessous.

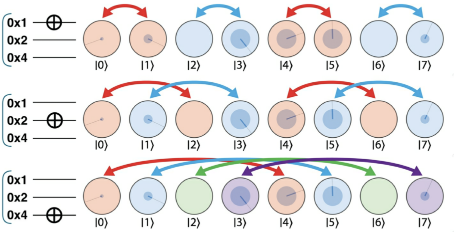


Figure 3-5. The NOT operation swaps values in each of the qubit's operator pairs; here, its action is shown on an example multi-qubit superposition

Explication : NOT échange les états $|0\rangle$ et $|1\rangle$ du *premier* qubit dans la représentation séparée, ainsi $000 \leftrightarrow 001$, $010 \leftrightarrow 011$, etc. Traduit en numéros d'états, ca fait que l'on échange $|0\rangle$ avec $|1\rangle$, $|2\rangle$ avec $|3\rangle$, etc. C'est le même raisonnement si l'on passe à la deuxième ligne: on applique NOT sur le deuxième qubit, donc on échange $000 \leftrightarrow 010$, $001 \leftrightarrow 011$, etc., donc $|0\rangle$ avec $|2\rangle$, $|1\rangle$ avec $|3\rangle$, etc. En résumé, si on travaille sur le qubit n, on apparie les états en faisant des sauts de 2^{n-1}.

Appliquons maintenant l'opérateur phase, on rappelle qu'il ne fait tourner de ϕ que le 1 du qubit concerné, on obtient donc le schéma suivant:

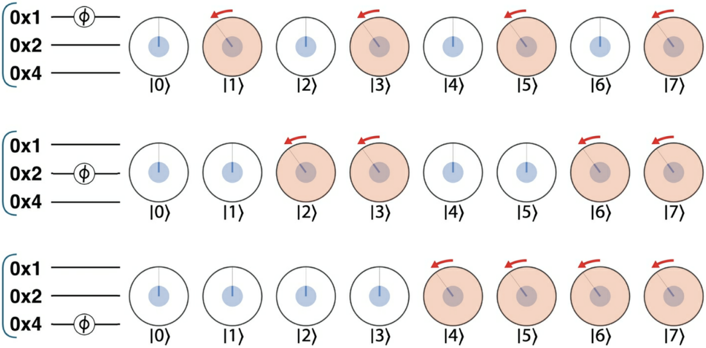


Figure 3-6. Single-qubit phase in a multi-qubit register

Évidemment, les représentations multi-qubits se complexifient vite quand on augmente le nombre de qubits, ce qui les rend moins lisibles.

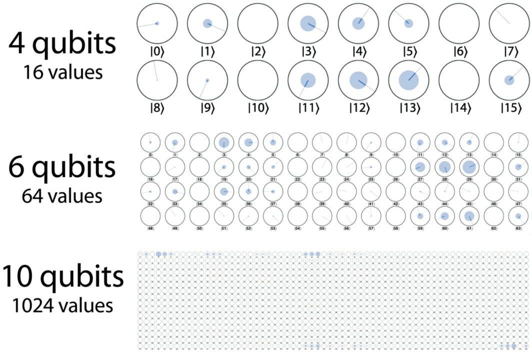
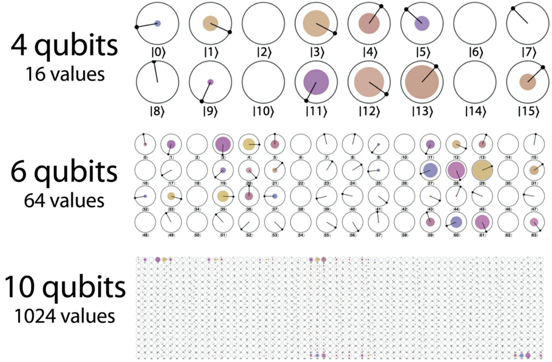


Figure 3-8. Circle notation for larger qubit counts

Il existe donc une représentation colorée qui met en valeur les amplitudes et les phases pour favoriser la lecture:



Outline

1 Introduction

2 Programming for a QPU

3 Les portes quantiques à un seul qubit

4 Détection d'espion par informatique quantique

5 Registres multi-qubits

6 Opérations multi-qubits

7 Opérateurs multiqubits conditionnels
- Porte CNOT
- Paire de Bell
- Porte quantique de changement de phase conditionnelle
- Phase kickback
- Porte de Toffoli
- Portes Swap et CSwap

8 Le Swap test

9 Téléportation quantique

Porte CNOT

1 Introduction

2 Programming for a QPU

3 Les portes quantiques à un seul qubit

4 Détection d'espion par informatique quantique

5 Registres multi-qubits

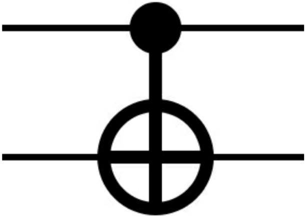
6 Opérations multi-qubits

7 Opérateurs multiqubits conditionnels
- Porte CNOT
- Paire de Bell
- Porte quantique de changement de phase conditionnelle
- Phase kickback
- Porte de Toffoli
- Portes Swap et CSwap

8 Le Swap test

L'instruction CNOT signifie: "quand le bit de contrôle est à 1, applique NOT sur le bit target".

QPU Instruction: CNOT



La logique paraît simple, mais le bit de contrôle peut être en superposition d'état ! (on détaillera plus tard).

Voici la différence de comportement entre un NOT et un CNOT:

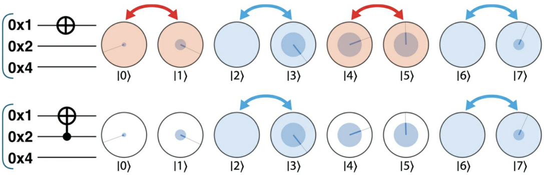


Figure 3-10. NOT versus CNOT in operation

On a déjà vu pour le NOT.

Procédure du CNOT lorsque le qubit de contrôle est le $0x2$, le qubit target est le $0x1$: pour tous les états correspondant à un 1 du qubit $0x2$, càd les états $010, 011, 110, 111$ (donc 2,3,6,7), on va appliquer NOT sur le 1er qubit. On ne fait donc des échanges que des états $|3\rangle$ avec $|4\rangle$, et de $|6\rangle$ avec $|7\rangle$ (rappel, on fait des appariements avec des sauts de 1 puisque c'est sur le 1er qubit qu'on applique l'opérator NOT).

Paire de Bell

1 Introduction

2 Programming for a QPU

3 Les portes quantiques à un seul qubit

4 Détection d'espion par informatique quantique

5 Registres multi-qubits

6 Opérations multi-qubits

7 Opérateurs multiqubits conditionnels
- Porte CNOT
- Paire de Bell
- Porte quantique de changement de phase conditionnelle
- Phase kickback
- Porte de Toffoli
- Portes Swap et CSwap

8 Le Swap test

Voici ce qu'on appelle la paire de Bell et voyons étape par étape le résultat qui en découle.

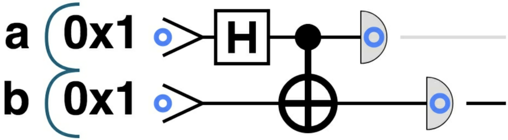


Figure 3-11. CNOT with a control qubit in superposition

On initialise les états séparément à 0, donc le registre est dans l'état $|0\rangle$.

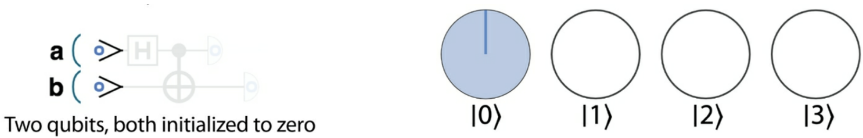


Figure 3-12. Bell pair step 1

On met en superposition les états $|0\rangle$ et $|1\rangle$ par un Hadamard.

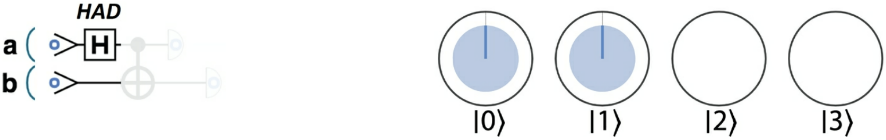


Figure 3-13. Bell pair step 2

On applique un CNOT (notez qu'il n'est pas positionné comme toute à l'heure !).

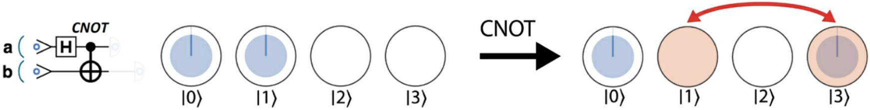


Figure 3-14. Bell pair step 3

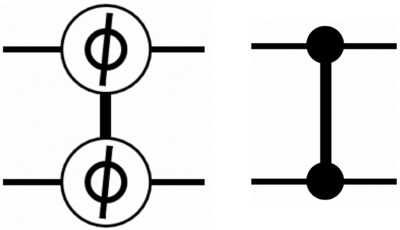
Le NOT se fait par sauts de 2, vu que le NOT est sur le deuxième qubit, et il se fait pour les états où le premier bit est à 1, donc on échange $|1\rangle$ et $|3\rangle$. On a créé un état intriqué grâce à une porte quantique multiqubits !

Porte quantique de changement de phase conditionnelle

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test

Avec une porte de changement de phase, on tourne les 1 quand on est à 1, mais attention, ces portes sont symétriques dans le sens où chaque bit target est bit de contrôle et inversement ! (notez la symétrie dans le dessin de la porte)

QPU Instructions: CPHASE and CZ



Another very common two-qubit operation is `CPHASE(θ)`. Like the `CNOT` operation, `CPHASE` employs a kind of entanglement-generating conditional logic. Recall from Figure 3-6 that the single-qubit `PHASE(θ)` operation acts on a register to rotate (by angle θ) the $|1\rangle$ values in that qubit's operator pairs. As `CNOT` did for `NOT`, `CPHASE` restricts this action on some target qubit to occur only when another control qubit assumes the value $|1\rangle$. Note that `CPHASE` only acts when its control qubit is $|1\rangle$, and when it does act, it only affects target qubit states having value $|1\rangle$. This means that a `CPHASE(θ)` applied to, say, qubits `0x1` and `0x4` results in the rotation (by θ) of all circles for which *both* these two qubits have a value of $|1\rangle$. Because of this particular property, `CPHASE` has a symmetry between its inputs not shared by `CNOT`. Unlike with most other controlled operations, it's irrelevant which qubit we consider to be the target and which we consider to be the control for `CPHASE`.

Attention à l'erreur dans le schéma, le $|1\rangle$ tourne aussi !

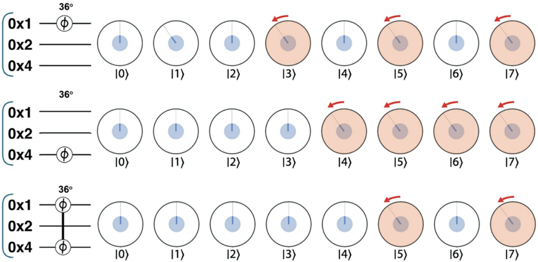


Figure 3-16. Applying CPHASE in circle notation

La logique par rapport aux bits de controle et de target est la même que d'habitude.

Notez l'égalité remarquable entre ces portes:

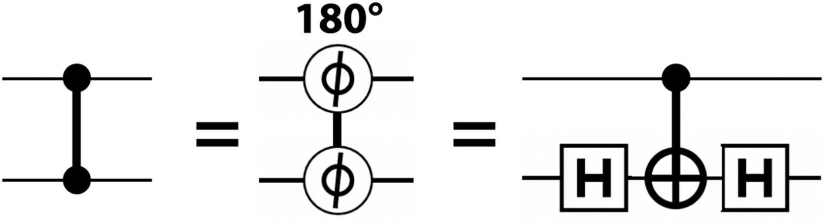


Figure 3-17. Three representations of CPHASE(180)

Phase kickback

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - **Phase kickback**
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test

Imaginons le circuit suivant :

QPU Trick: Phase Kickback

Once we start thinking about altering the phase of one QPU register *conditioned* on the values of qubits in some other register, we can produce a surprising and useful effect known as *phase kickback*. Take a look at the circuit in Figure 3-18.

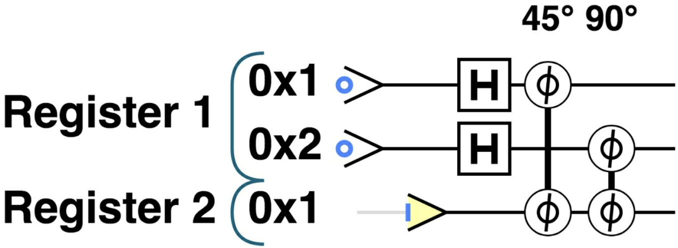


Figure 3-18. Circuit for demonstrating phase-kickback trick

On obtient cet effet étonnant (il nous servira plus tard pour résoudre des systèmes d'équations linéaires, pour estimer des phases, etc.):

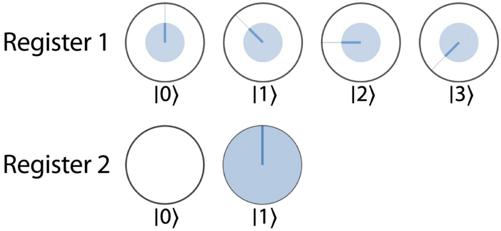


Figure 3-19. States of both registers involved in phase kickback

Le code est disponible à l'endroit habituel :

SAMPLE CODE

Run this sample online at http://oreilly-qc.github.io?p=3-3.

Example 3-3. Phase kickback

```
qc.reset(3);
// Create two registers
var reg1 = qint.new(2, 'Register 1');
var reg2 = qint.new(1, 'Register 2');
reg1.write(0);
reg2.write(1);
// Place the first register in superposition
reg1.had();
// Perform phase rotations on second register,
// conditioned on qubits from the first
qc.phase(45, 0x4, 0x1);
qc.phase(90, 0x4, 0x2);
```

Porte de Toffoli

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test

Il existe une porte extrêmement connue, la porte de Toffoli, qui comporte deux bits de contrôle !

QPU Instruction: CCNOT (Toffoli)

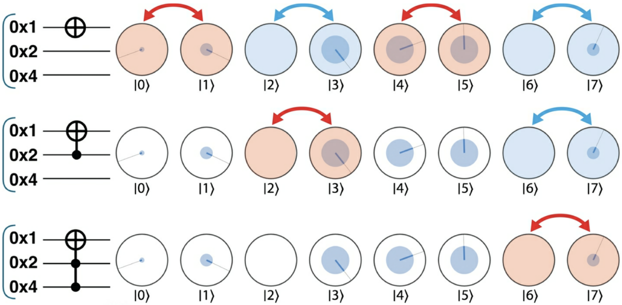


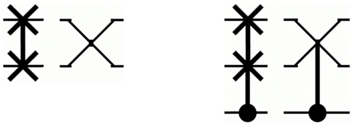
Figure 3-20. Adding conditions makes NOT operations more selective

A vous de comprendre sa logique (en exercice), quelle opération connue réalise-t-elle entre ses qubits de contrôle ?

Portes Swap et CSwap

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test

QPU Instructions: SWAP and CSWAP



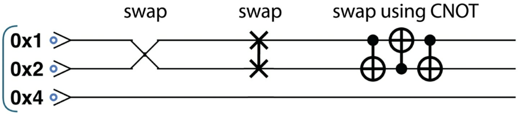


Figure 3-21. SWAP can be made from CNOT operations

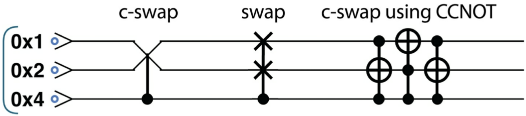


Figure 3-22. CSWAP constructed from CCNOT gates

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

The Swap Test

SWAP operations allow us to build a very useful circuit known as a *swap test*. A swap test circuit solves the following problem: if you're given two qubit registers, how do you tell if they are in the *same* state? By now we know only too well that in general (if either register is in superposition), we can't use the destructive READ operation to completely learn the state of each register in order to make the comparison. The SWAP operation does something a little sneakier. Without telling us what either state is, it simply lets us determine whether or not they're equal.

In a world where we can't necessarily learn precisely what's in an output register, the swap test can be an invaluable tool. Figure 3-23 shows a circuit for implementing a swap test, as demonstrated in Example 3-4.

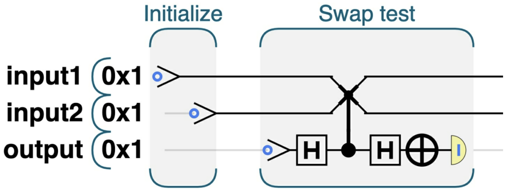


Figure 3-23. Using the swap test to determine whether two registers are in the same state

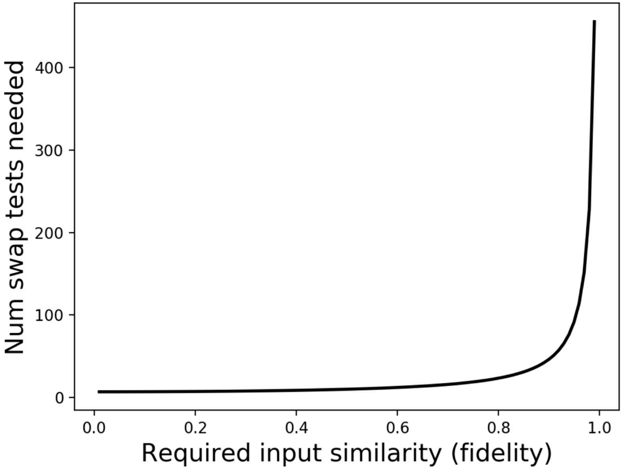


Figure 3-25. Number of swap tests that would need to return an outcome of 1 for us to be 99% confident inputs are identical

Outline

1. Introduction
2. Programming for a QPU
3. Les portes quantiques à un seul qubit
4. Détection d'espion par informatique quantique
5. Registres multi-qubits
6. Opérations multi-qubits
7. Opérateurs multiqubits conditionnels
   - Porte CNOT
   - Paire de Bell
   - Porte quantique de changement de phase conditionnelle
   - Phase kickback
   - Porte de Toffoli
   - Portes Swap et CSwap
8. Le Swap test
9. Téléportation quantique

Téléportation quantique :

Croyance : Il s'agirait de transporter instantanément un objet d'un point à un autre.

En vérité : Il s'agit juste de transmettre un état d'un qubit à un autre.

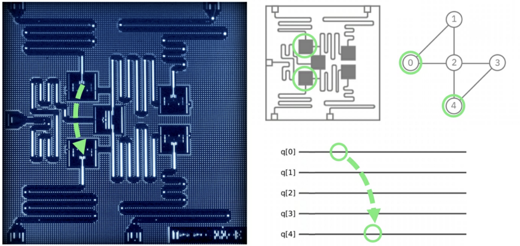


Figure 4-1. The IBM chip is very small, so the qubit does not have far to go; the image and schematics show the regions on the QPU we will teleport between[1]