Chapitre 12

Information et calcul quantique

L'objectif de ce chapitre est d'effleurer quelques aspects de la théorie et de la pratique de l'information et du calcul quantique. Je souhaite vous montrer que ce cours vous permet déjà d'aborder certains aspects des technologies quantiques actuelles. En premier lieu, nous reviendrons sur la notion de *qubit*, par analogie au *bit* d'information classique, et nous utiliserons, pour les manipuler, certaines opérations unitaires rencontrées dans les chapitres précédents. En particulier, nous verrons comment manipuler les *qubit* : créer des superpositions et de l'intrication. Les notions abordées dans ce cours sont à maîtriser pour aborder la RP2.

12.1 Information classique : notion de *bit* et de porte logique

Bit classique L'électronique numérique et l'informatique classique utilisent des transistors comme composant de base pour traiter de l'information. Les transistors sont des composants électroniques non linéaires, qui peuvent être utilisés comme interrupteurs, commutant entre deux niveaux de tension. L'unité d'information classique naturelle est donc un ensemble de deux valeurs. La structure mathématique naturelle pour représenter cette information et la traiter est l'algèbre de Boole, composée des deux éléments $\{0, 1\}$, que l'on note parfois $\{\text{FAUX}, \text{VRAI}\}$, ou encore $\{\text{NON}, \text{OUI}\}$. En pratique chacun de ces deux éléments correspond à l'un des deux niveaux de tension électrique à la sortie du transistor.

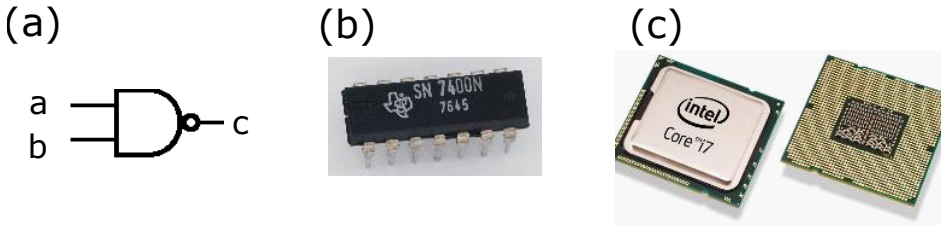


FIGURE 12.1 – (a) Schéma de la porte NAND. Cette porte a deux entrée dont les états sont notés a et b, et une sortie c. L'état en sortie s'obtient par l'opération logique suivante : $c = \overline{a.b}$. (b) Circuit intégré contenant 4 portes NAND. (c) CPU d'ordinateur contenant des centaines de millions de portes NAND.

Portes logiques En connectant différents transistors entre eux, on peut réaliser des opérations logiques, c'est à dire mettre en oeuvre des opérateurs de l'algèbre de Boole. Ces opérations logiques sont souvent appelées *portes logiques*. La plus simple est la porte NOT, ou porte NON en français. Cette porte bascule l'état d'un bit. Le 0 devient 1 et le 1 devient 0. La porte la plus utilisée, et la plus célèbre, car elle peuple majoritairement les puces de vos ordinateurs, est la porte NAND [1] (cf. Figure 1). La porte NAND possède deux entrées a et b et une sortie c. La valeur de c se déduit des valeurs de a et b via l'opération logique $c = \overline{a.b}$ (la barre supérieure est l'opération logique NON, et le point est

1. *Not And*, c'est-à-dire *Non Et* en français.

l'opération logique ET. On peut représenter la totalité des états possibles de cette portes grâce à une *table de vérité* :

a	b	c
0	0	1
1	0	1
0	1	1
1	1	0

En d'autres termes, la sortie vaut toujours un, sauf si les deux entrées valent 1. On peut montrer que la porte NAND est *universelle*, ce qui signifie que n'importe quelle opération logique peut être obtenue en combinant des portes NAND.

12.2 Information quantique : notion de *qubit* et porte quantique

Quantum bit ou *qubit* Les système d'information quantique utilisent majoriatirement des *qubits*. Un qubit est un système à deux états $|0\rangle$ et $|1\rangle$. Il peut être de natures diverses (spin 1/2, NH3, etc). La différence avec le *bit* classique est la possibilité de créer des superpositions à coefficients complexes des deux états, ainsi que la possibilité d'intriquer plusieurs *qubits*. Rappelons que le 0 et le 1 notés à l'intérieur du ket ne sont pas des valeurs propres d'observable, mais simplement des étiquettes choisie par analogie au *bit* classique. **La valeur propre associée à l'état $|0\rangle$ est 1, celle associée à l'état $|1\rangle$ est -1** (valeurs propres des matrices de Pauli [2]).

Portes quantiques En information classique les opérations sur les *bit* sont irréversibles : par exemple dans le cas de la porte NAND, on a deux entrées et une sortie, et il est impossible de retrouver a et b connaissant c. En information quantique, les portes logiques sont des *opérations unitaires* réversibles. Les opérateurs unitaires d'un espace de Hilbert vérifient

$$\widehat{U}^\dagger \widehat{U} = \widehat{U}\widehat{U}^\dagger = \hat{\mathbb{1}}, \tag{12.1}$$

que l'on peut également écrire

$$\widehat{U}^\dagger = \widehat{U}^{-1}. \tag{12.2}$$

Un opérateur unitaire est inversible, et son inverse est égal à son adjoint. Vous avez déjà rencontré des opérations unitaires dans ce cours : les changements de base. Un changement de base est une opération inversible (on peut faire le changement de base dans les deux sens), et si vous retournez voir les différents changements de base introduits dans les chapitres précédents, vous constaterez que l'inverse de la matrice est toujours égale à sa transconjuguée. Ces opérations unitaires ont pour propriété de conserver la norme des vecteurs. En effet, un vecteur d'une base orthonormé a pour image un vecteur d'une autre base orthonormée.

12.2.1 Portes quantiques à 1 *qubit*

Les portes quantiques à 1 *qubit* sont les opérations unitaires de l'espace des états d'un *qubit*. Elles sont donc représentées par des matrices 2x2 qui vérifient l'équation 12.2. Les exemples les plus importants et les plus simples sont les matrices de Pauli. Ces matrices sont à la fois hermitiennes et unitaires. Elles peuvent donc représenter des observables et portes quantiques. Dans toute la suite nous écrirons toutes les matrices dans la base dite *computationnelle* des états propres de σ_z : $\{|0\rangle, |1\rangle\}$.

2. En théorie de l'information quantique on ne donne pas une dimension physique aux valeurs propres associées à ces états propres.

Porte X (bit flip, porte NON, Pauli-X) :

$$X \equiv (\sigma_x) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{12.3}$$

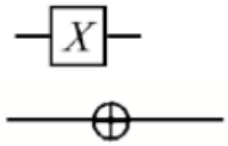
Action sur les vecteur de la base computationnelle :

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$
$$X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

La table de vérité est donc :

entrée	**sortie**
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

Cette porte a pour effet d'échanger les deux états de la base (on dit aussi qu'elle retourne le qubit, d'où le nom anglais de *bit-flip*). C'est l'équivalent quantique de la porte NON (NOT gate en anglais) qui change 0 en 1 et 1 en 0. On la représente graphiquement par l'un ou l'autre des symboles



Porte Z (phase-flip, porte π, Pauli-Z) :

$$Z \equiv (\sigma_z) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{12.4}$$

Action sur les vecteurs de la base computationnelle :

$$Z|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$
$$Z|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = -|1\rangle = e^{i\pi}|1\rangle$$

La table de vérité est donc :

entrée	**sortie**
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$e^{\mathrm{i}\pi}\|1\rangle$

Cette porte a pour effet d'ajouter une phase π sur le second état de la base computationnelle. Pour bien comprendre son action, appliquons-là à une superposition d'états :

$$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{\mathrm{i}\pi}\beta|1\rangle = \alpha|0\rangle - \beta|1\rangle \tag{12.5}$$

La superposition acquiert une phase relative π, d'où le nom *porte* π. Le ket fait un demi tour autour de l'axe O_z dans la sphère de Bloch, d'où l'expression *phase-flip*. Cette porte n'a pas d'équivalent classique, puisqu'il n'y a aucune notion de phase dans l'algèbre de Boole. On représente cette porte par le symbole

Porte Y (phase-flip + bit-flip, Pauli-Y) :

$$Y \equiv (\sigma_y) = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} \tag{12.6}$$

Je vous invite à vérifier par vous-même cette propriété remarquable des matrices de Pauli : $\sigma_y = -\mathrm{i}\sigma_z\sigma_x$. La porte Y est donc l'application successive d'une porte X et d'une porte Z. Il s'agit d'un *bit-flip* suivi d'un *phase-flip*. On la représente graphiquement par le symbole

Porte Hadamard : Cette porte est la matrice de changement de base entre les bases couplées et non couplées de la molécule d'ammoniac, c'est également la matrice de passage entre les bases propres $\mathcal{B}_\mathrm{x}$ et $\mathcal{B}_\mathrm{z}$ des composantes du spin 1/2. Elle a donc été largement utilisée dans les chapitres précédents.

$$(H) \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{12.7}$$

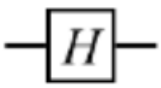
Action sur les vecteur de la base computationnelle :

$$H\,|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right)$$
$$H\,|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle\right)$$

Sans surprise on trouve les vecteur propre de σ_x. Par ailleurs, on constate que cette porte crée des superpositions d'états. Elle n'a donc pas d'équivalent classique. Sa table de vérité est :

entrée	**sortie**
$\|0\rangle$	$\frac{1}{\sqrt{2}}\left(\|0\rangle + \|1\rangle\right)$
$\|1\rangle$	$\frac{1}{\sqrt{2}}\left(\|0\rangle - \|1\rangle\right)$

Le symbole associé à la porte Hadamard est :



Attention, la porte Hadamard est notée « H ». **Cette porte n'a rien à voir avec un hamiltonien.** Mais dans ce contexte comme on n'écrit jamais d'hamiltonien cela n'est pas dérangeant.

Dernière remarque : en pratique pour réaliser une porte Hadamard, on peut utiliser des oscillations de Rabi.

12.2.2 Portes quantiques à 2 *qubit* : la CNOT

Les portes à un *qubit* ne permettent pas de coupler plusieurs *qubit*. En particulier, elle ne permettent pas de générer à elles-seules de l'intrication. Au minimum, pour obtenir des fonctionnalités intéressantes, il faut donc au moins une porte à 2 *qubit*. La plus connue est la porte CNOT (Controlled

NOT). Cette porte agit dans un espace de Hilbert $\mathcal{E}$ produit tensoriel entre deux espaces de Hilbert $\mathcal{E}_1$ et $\mathcal{E}_2$ de dimension 2. Sa dimension est donc 2x2=4. Une matrice unitaire représentant une porte à 2 *qubit* est donc une matrice 4x4. La base computationnelle de l'espace de Hilbert à 2 *qubit* est composée de tous les produits tensoriels possibles des états des bases computationelles de $\mathcal{E}_1$ et $\mathcal{E}_2$, c'est-à-dire :

$$\{|0\rangle_1 \otimes |0\rangle_2\,,\ |0\rangle_1 \otimes |1\rangle_2\,,\ |1\rangle_1 \otimes |0\rangle_2\,,\ |1\rangle_1 \otimes |1\rangle_2\}\,,$$

souvent notée de manière plus compacte et plus lisible

$$\{|00\rangle\,,\ |01\rangle\,,\ |10\rangle\,,\ |11\rangle\}\,.$$

Pour vraiment clarifier, voici l'écriture matricielle des vecteurs de cette base, exprimés dans cette base :

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix},\ |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix},\ |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix},\ |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

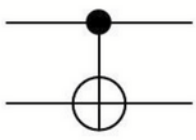
La matrice de la porte CNOT dans cette base est

$$\text{CNOT} = \begin{pmatrix}\hat{\mathbb{1}} & 0\\ 0 & X\end{pmatrix} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix} \tag{12.8}$$

Sa table de vérité est :

entrée	**sortie**
$\|0\rangle\,\|0\rangle$	$\|0\rangle\,\|0\rangle$
$\|0\rangle\,\|1\rangle$	$\|0\rangle\,\|1\rangle$
$\|1\rangle\,\|0\rangle$	$\|1\rangle\,\|1\rangle$
$\|1\rangle\,\|1\rangle$	$\|1\rangle\,\|0\rangle$

Premièrement, on constate que l'état du premier *qubit* n'est jamais modifié par cette porte. Ensuite, sur les deux première lignes, lorsque l'état du premier *qubit* est $|0\rangle$, on voit que l'état du second *qubit* est inchangé. Sur les deux dernières lignes, lorsque l'état du premier *qubit* est $|1\rangle$, on voit que le second *qubit* subit un *bit-flip*. D'où le nom de la porte Le second *qubit* subit une porte NOT si l'état du premier *qubit*, appelé *qubit de contrôle*, est $|1\rangle$. En d'autres termes l'application de la porte NOT au second *qubit* est contrôlée par l'état du premier *qubit*. On représente cette porte par le symbole

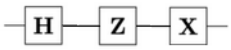


La première ligne horizontale représente l'état du premier *qubit*, et la seconde ligne l'état du second *qubit*. On reconnaît le symbole de la porte NOT, appliquée sur le second *qubit*. La ligne verticale qui connecte les deux *qubit* indique que la porte NOT est contrôlée par l'état du premier *qubit*.

12.3 Algorithme quantique

On nomme *algorithme quantique* l'application successive d'un certain nombre de portes quantique à un ou plusieurs *qubit*, en général suivie des mesures projectives de tous les *qubit*s dans la base computationelle.

Algorithme à un *qubit* : voici par exemple la représentation diagrammatique d'un algorithme à un *qubit* :



Ce schéma représente l'application successive des portes H, Z puis X à un *qubit* (traditionnellement l'état initial du *qubit* est représenté à gauche et l'état final à droite. Il faut voir l'axe horizontal comme un axe des temps). On peut donc déterminer l'opération unitaire équivalente à cette succession de portes, il suffit de calculer le produit

$$XZH = \frac{1}{\sqrt{2}}\begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Si le *qubit* est initialement dans l'état $|0\rangle$, alors l'état en sortie est

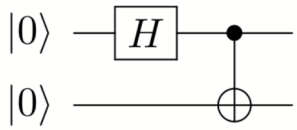
$$|\psi_f\rangle = \mathrm{XZH}\,|0\rangle = \frac{1}{\sqrt{2}}\left(-|0\rangle + |1\rangle\right).$$

Si l'on effectue ensuite une mesure de l'état du *qubit* dans la base computationnelle, on peut trouver les valeurs propres -1, et 1 avec les probabilités

$$\begin{aligned} p(-1) &= |\langle 1|\psi_f\rangle|^2 = \frac{1}{2} \\ p(1) &= |\langle 0|\psi_f\rangle|^2 = \frac{1}{2}. \end{aligned}$$

les algorithmes à 1 *qubit* ne sont pas très riches car il n'y a pas d'intrication.

Algorithme à 2 *qubit* : intrication Considérons l'algorithme suivant :



Les deux *qubit* sont initialisés dans l'état $|0\rangle$. On applique alors la porte Hadamard au premier *qubit*, suivi d'une porte CNOT aux deux *qubit*, le premier *qubit* étant le *qubit* de contrôle. Écrivons étape après étape l'état du système :

L'état initial à deux *qubit* est

$$|\psi_i\rangle = |0\rangle_1 \otimes |0\rangle_2 \equiv |0\rangle_1 |0\rangle_2 \equiv |00\rangle$$

Après application de la porte Hadamard au premier *qubit*, l'état du système est

$$\left(H \otimes \hat{\mathbb{1}}_{\mathcal{E}_1}\right)|0\rangle_1 \otimes |0\rangle_2 = \left(H\,|0\rangle_1\right) \otimes \left(\hat{\mathbb{1}}_{\mathcal{E}_1}\,|0\rangle_2\right) = \frac{1}{\sqrt{2}}\left(|0\rangle_1 + |1\rangle_1\right) \otimes |0\rangle_2 = \frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right)$$

Cet état est un état factorisable. Il n'est pas intriqué. Appliquons maintenant la porte CNOT, qui a pour effet de basculer l'état du *qubit* 2, si le *qubit* 1 est dans l'état $|1\rangle$. L'état final est :

$$|\psi_f\rangle = \mathrm{CNOT}\frac{1}{\sqrt{2}}\left(|00\rangle + |10\rangle\right) = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right).$$

Cet état est intriqué ! Notez qu'on aurait pu déterminer l'état final en utilisant un calcul matriciel direct. En effet cet algorithme est l'opération unitaire de l'espace $\mathcal{E} = \mathcal{E}_1 \otimes \mathcal{E}_2$ suivante

$$\mathrm{CNOT}(\mathrm{H} \otimes \hat{\mathbb{1}}_{\mathcal{E}_2}) = \begin{pmatrix} \hat{\mathbb{1}} & \\ & X \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{\mathbb{1}} & \hat{\mathbb{1}} \\ \hat{\mathbb{1}} & -\hat{\mathbb{1}} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \hat{\mathbb{1}} & \hat{\mathbb{1}} \\ X & -X \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{pmatrix} \tag{12.9}$$

Vous pouvez vérifier par vous même qu'appliquer cette matrice au ket initial $|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ donne le même état intriqué.

Dans la RP2, vous allez mettre en oeuvre les portes à un et deux *qubit* vues dans ce chapitre sur un système réel. En particulier, vous allez faire la tomographie de l'état d'un *qubit* (cf TD8), et créer un état intriqué avec l'algorithme ci-dessus.

Exemple d'algorithme : Pour finir, voici un exemple d'algorithme utile. Il s'agit de l'algorithme de Grover. Cet algorithme permet de chercher une donnée répondant à certains critères, dans un ensemble de données. Le diagramme ci-dessous montre son implémentation pour 8 *qubit*.

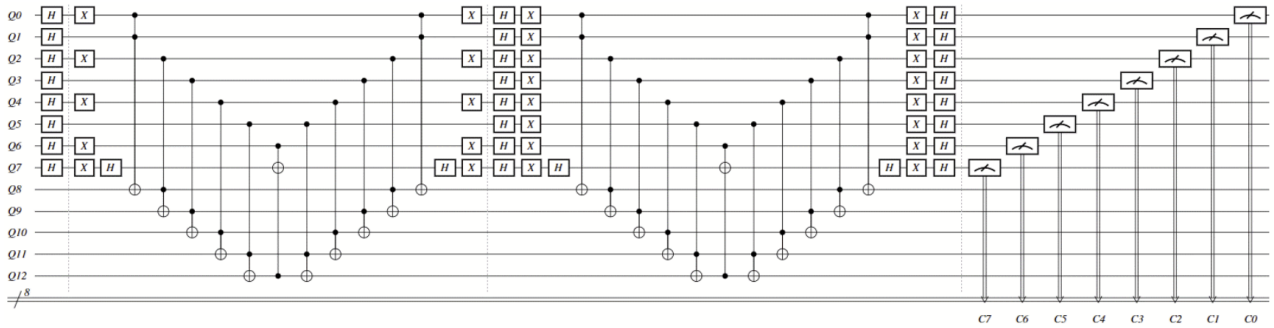


FIGURE 12.2 – Algorithme de Grover à 8 *qubit*.