الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Centre Universitaire Abdelhafid Boussouf Mila

Institut de Mathématique et Informatique

Mémoire préparé En vue de l'obtention du diplôme de Master en Informatique

Filière : Informatique

Spécialité : Sciences et Technologies de l'Information

Et de la Communication (STIC)

Conception d'une unité arithmétique et logique d'un ordinateur quantique utilisant des circuits et des portes logiques quantiques.

Préparé par : Aichour Narimane.

Soutenue devant le jury :

Président :	Attia Mourad.	C.U. Abdelhafid Boussouf.
Examinatrice :	Deffas Zineb.	C.U. Abdelhafid Boussouf.
Encadrant :	Slimani Ayyoub.	C.U. Abdelhafid Boussouf.
Co-encadrant :	Hettab Abdelkamel.	C.U. Abdelhafid Boussouf.

Année universitaire : 2023/2024

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Remerciement

D’abord je remercie dieu le tout puissant qui m’a donné la force, la volonté et le courage pour accomplir ce modeste travail.

Les mots manquent pour exprimer mon profond sentiment de gratitude envers mes superviseurs et mentors **M**. **Slimani Ayoub** et **M**. **Abdelkamel Hettab**, pour leur patience, leurs encouragements et leurs suggestions. Merci pour vos échanges scientifiques, vos conseils, votre rigueur et tous les efforts que vous avez fournis pour faciliter et aider à l'accomplissement de mon travail de fin d’études.

Je remercie également les membres du jury **M. Attia Mourad** et **Mme. Deffas Zineb** pour avoir offert son temps et ses efforts pour voir le travail.

Je n'oublie pas non plus mes enseignants, qui, tout au long du cycle d’études au Centre Universitaire de Mila, nous ont transmis leur savoir.

Enfin, j'exprime mes sincères remerciements à ma famille et à mes amis pour leur soutien et leurs encouragements tout au long de la réalisation de ce projet. Ils m'ont toujours soutenu moralement et financièrement tout au long de mes années scolaires. Merci à tous pour tout.

Dédicace

Je remercie tout d'abord Dieu tout-puissant pour la grande grâce qu'il m'a donnée.

A mes chers parents :

Salim & Leila

Merci pour votre confiance et votre encouragement constant ont été des piliers essentiels dans mon éducation. Votre soutien financier, votre disponibilité et votre dévouement ont fait de moi la personne que je suis aujourd'hui. Vos sacrifices et votre investissement pour mon éducation ont été une source d'inspiration constante.

Je tien à dédier ma famille surtout :

- ✓ Mes sœurs : - **Imane** et son Mari **Adel** et leurs enfants **Djana** et **Khalil.**

  -Wahiba.
- ✓ Mes frères : **Zouhir, Islam.**
- ✓ Ma grand-mère

Tous mes amis surtout :

- ✓ Yasmine, Bouchra, Manel, Selma et chaima.
- ✓ Ben Yahia Hassen, Bougueribia Bessam.

Aichour Narimane

Résumé

Un ordinateur quantique repose sur des principes quantiques pour effectuer des calculs, exploitant les propriétés uniques des particules quantiques appelées qubits. L'unité arithmétique et logique (UAL) quantique est un composant essentiel de ces systèmes, responsable de l'exécution d'opérations arithmétiques et logiques sur les qubits.

L'objectif principal de ce travail concentre sur la conception d'une unité arithmétique et logique (UAL) pour les ordinateurs quantiques, en utilisant des circuits et des portes logiques quantiques. Cette UAL capable de réaliser des opérations arithmétiques et logiques essentielles.

Nous commençons par explorer les concepts fondamentaux des qubits et des portes logiques quantiques, telles que les portes Hadamard, CNOT et Toffoli …etc. ainsi que leur rôle dans la construction des circuits quantiques.

Nous avons conçu un circuit d'unité arithmétique et logique quantique, puis nous avons comparé le circuit proposé aux conceptions précédentes en termes de coût quantique (QC), de délai quantique (QD) et de nombre d'opérations produites par le circuit. Le circuit proposé a montré sa supériorité sur son homologue, notamment dans le paramètre d'évaluation "délai quantique (QD)".

Pour valider nos conceptions, nous utilisons Qiskit, une bibliothèque open-source développée par IBM Quantum, pour simuler et évaluer les performances de nos circuits.

Les résultats de nos simulations montrent que notre UAL quantique peut réaliser des opérations avec précision et efficacité.

Mots clés : ordinateur quantique, portes logiques quantiques, circuit, unité arithmétique et logique quantique, Qiskit.

Abstract

A quantum computer relies on quantum principles to perform calculations, exploiting the unique properties of quantum particles called qubits. The quantum arithmetic and logic unit (ALU) is an essential component of these systems, responsible for executing mathematical and logical operations on qubits.

The primary objective of this work focuses on the design of an arithmetic and logic unit (ALU) for quantum computers, using quantum logic circuits and gates. This ALU is capable of performing essential arithmetic and logical operations.

We begin by exploring the fundamental concepts of qubits and quantum logic gates, such as Hadamard, CNOT, and Toffoli gates, and their roles in constructing quantum circuits.

We designed a quantum logic and arithmetic unit circuit, then we compared the proposed circuit with previous designs in terms of quantum cost (QC), quantum delay (QD) and the number of operations produced by the circuit. The proposed circuit showed superiority over its counterpart, especially in the parameter of evaluation "quantum delay (QD)".

To validate our designs, we use Qiskit, an open-source library developed by IBM Quantum, to simulate and evaluate the performance of our circuits.

The results of our simulations show that our quantum ALU can perform operations with precision and efficiency.

Keywords : Quantum computer, Quantum logic gates, Circuit, Quantum arithmetic and logic unit, Qiskit.

ملخص

يعتمد الحاسوب الكمي على المبادئ الكمية لأداء الحسابات، مستغلاً الخصائص الفريدة للجسيمات الكمية التي تُدعى الكيوبتات. تُعد وحدة الحساب والمنطق الكمية مكونًا أساسيًا لهذه الأنظمة، حيث تتولى تنفيذ العمليات الحسابية والمنطقية على الكيوبتات.

يركز الهدف الرئيسي من هذا العمل على تصميم وحدة حساب ومنطق لأجهزة الحاسوب الكمي،باستعمال الدارات و البوابات من خلال تنفيذ الدوائر والأبواب المنطقية الكمية. هذه الوحدة قادرة على تنفيذ العمليات الحسابية والمنطقية الأساسية.

نبدأ باستكشاف المفاهيم الأساسية للكيوبتات والأبواب المنطقية الكمية، مثل بوابات Hadamard ،CNOT وToffoli

ودورها في بناء الدارات الكمية.

قمنا بتصميم دارة وحدة الحساب المنطق الكمية، ثم قمنا بمقارنة الدارة المقترحة مع تصاميم سابقة من حيث الكلفة الكمية QC ، التاخير الكمي QD و عدد العمليات التي تنتجها الدارة ، حيث اظهرت الدارة المقترحة تفوقا على نظيرتها خاصة في معيار التقييم "التاخير الكمي QD".

للتحقق من صحة تصاميمنا، نستخدم Qiskit ، وهي مكتبة مفتوحة المصدر طورتها IBM Quantum، لمحاكاة وتقييم

أداء الدوائر.

تُظهر نتائج محاكاتنا أن وحدة الحساب والمنطق الكمية التي قمنا بتصميمها يمكنها تنفيذ العمليات بدقة وكفاءة.

الكلمات المفتاحية: حاسوب كمي، أبواب منطقية كمية، دائرة، وحدة حساب ومنطق كمية،Qiskit .

Liste des abréviations :

UAL : Unité Arithmétique et Logique.

QC : Quantum Cost.

QD : Quantum Delay.

QG : Quantum Garbage.

CNOT : Controled-Not.

C-V : Controled-V.

C-V+ : Controled-V+.

NCT: NOT, CNOT, Toffoli.

NCV: NOT, CNOT, Controled V ou Controled V+.

DPG: Double Peres Gate.

Table des matières

Listes des Tableaux

Chapitre 01 : De l'informatique classique vers l'informatique quantique

Chapitre 02 : Conceptions des circuits quantiques et d'UAL quantique.

Chapitre 03 : La conception proposée d'unité arithmétique et logique quantique et sa simulation dans Qiskit « la partie pratique »

Listes des figures

Chapitre 01 : De l'informatique classique vers l'informatique quantique

Introduction générale :

L'information quantique est un domaine de recherche novateur en plein essor, visant à manipuler des systèmes physiques en tirant parti des propriétés distinctives de la mécanique quantique, telles que la superposition et l'intrication. L'objectif est de traiter, stocker et transmettre l'information de manière plus efficace. Cette approche promet de fournir des solutions innovantes à divers problèmes informatiques et de conduire à des avancées technologiques révolutionnaires.

Depuis les années 1980, les physiciens ont réussi à manipuler et observer des objets quantiques élémentaires tels que les photons, les atomes et les ions de manière individuelle. Cela a rendu possible la création de bits quantiques et a donné naissance au domaine de l'information quantique. Pendant cette période, Richard Feynman, lauréat du Prix Nobel de physique en 1965, a souligné la difficulté de simuler des systèmes quantiques avec des ordinateurs classiques. En réponse, il a proposé de construire un ordinateur quantique capable de simuler efficacement le comportement des systèmes quantiques.

Actuellement, les composants électroniques des ordinateurs ont été développés grâce au traitement quantique de l'information, où le fonctionnement des transistors repose sur ce traitement, permettant ainsi de réduire la taille des composants à environ 50 nm. De plus, la miniaturisation des composants atteindra bientôt l'échelle nanométrique, comme le prédit la célèbre loi de Moore, selon laquelle les ordinateurs classiques cesseront de fonctionner efficacement d'ici une quinzaine d'années. Par conséquent, il devient essentiel de prendre en compte les effets quantiques dans les technologies de traitement de l'information pour le développement de l'informatique quantique.

Les ordinateurs quantiques se distinguent fondamentalement des ordinateurs classiques en utilisant des qubits pour effectuer des calculs. Contrairement aux ordinateurs traditionnels qui utilisent des bits binaires (0 ou 1), les qubits peuvent être en superposition, ce qui leur permet de traiter simultanément une multitude de valeurs. Quant à l'unité arithmétique et logique (UAL) dans un contexte classique, elle est remplacée dans les ordinateurs quantiques par des circuits quantiques qui manipulent des portes logiques quantiques comme la porte de Hadamard et la porte CNOT pour effectuer des opérations logiques avancées. Ces technologies ouvrent la voie à des capacités de calcul potentiellement révolutionnaires pour résoudre des problèmes complexes à une échelle inédite.

Notre projet de fin d'études est la conception d'une unité arithmétique et logique d'un ordinateur quantique utilisant des portes et des circuits quantiques, l'objectif essentielle de ce projet est la réalisations une UAL quantique effectuer plusieurs opérations grâce à des principes quantiques.

Ce travail est organisé en trois principaux chapitres comme suit :

- ✓ **Chapitre 01 :** De l'informatique classique vers l'informatique quantique.

- ✓ **Chapitre 02 :** Conceptions des circuits quantiques et d'UAL quantique.
- ✓ **Chapitre 03 :** La conception proposée d'unité arithmétique et logique quantique et sa simulation dans Qiskit « la partie pratique ».

Chapitre 01 :

De l’informatique classique vers l’informatique quantique

Introduction :

Dans ce chapitre, nous examinerons les notions de base de l'informatique classique, telles que les portes logiques et les limites de cette technologie. Nous passerons ensuite en revue les concepts fondamentaux de l'informatique quantique, notamment la superposition des états quantiques, l'intrication quantique, la décohérence, les qubits, les portes logiques quantiques et les circuits quantiques. Enfin, nous conclurons par une brève explication des algorithmes quantiques.

1 L'informatique classique :

Le domaine de l'informatique conventionnelle, qui repose sur la fiabilité absolue de la logique binaire, fonctionne à l'intérieur de la structure rigide des bits et des algorithmes déterministes. L'ordinateur classique, prouesse technologique de l'époque moderne, a constitué la pierre angulaire des innovations technologiques au cours des dernières décennies. Sa puissance réside dans sa formidable capacité à manipuler, conserver et récupérer des quantités colossales d'informations à une vitesse et une précision surprenante. (A. V. Navaneeth, 2021)

1.1 Bit classique :

Le terme « bit » est une abréviation de « binary digit », ce qui représente les chiffres binaires 0 ou 1. Il est utilisé pour désigner la plus petite unité d'information qui peut être manipulée par une machine numérique. (Reda Adjoudj, 2009)

1.1.1 Encodage d'information :

L'information dans un ordinateur est encodée en utilisant le système binaire, qui est basé sur la base 2. Les chiffres binaires sont appelés des bits. Un bit peut prendre la valeur de 0 ou 1, et l'information est représentée par une séquence de bits. Une séquence de 8 bits est communément appelée un « octet ». (S´everine Fratani, 23 septembre 2014)

1.1.2 Système binaire :

Le système binaire est le système de codage de l'information utilisé dans les ordinateurs. Ce système repose sur l'utilisation de deux états distincts, représentés par les chiffres 0 et 1, pour encoder les données. C'est grâce à ce système que les ordinateurs peuvent traiter et manipuler l'information de manière électronique. (Reda Adjoudj, 2009)

1.1.3 Code ASCII :

Le code ASCII (American Standard Code for Information Interchange) crée en 1961 associe un nombre `a un caractère. Chaque caractère est codé sur 7 bits. Il ne contient donc que 128 caractères différents (de 00 à 7F en hexadécimal). (S´everine Fratani, 23 septembre 2014)

1.2 Les portes logiques :

Les portes logiques sont des éléments essentiels des circuits logiques, fonctionnant selon des principes booléens. Elles sont constituées d'entrées (situées à gauche sur les schémas) et d'une sortie (située à droite). Les entrées reçoivent des signaux binaires (0 ou 1), et la porte logique effectue une opération spécifique sur ces signaux pour produire un résultat qui est émis par la sortie.

Chaque porte logique est associée à une table de vérité qui spécifie la sortie correspondante pour chaque combinaison d'entrées possible. (S´everine Fratani, 23 septembre 2014)

1.2.1 Portes à un bit :

- **Porte NON (NOT) :** l'opérateur d'inverse a une seule entrée et une seule sortie. La sortie est 1 seulement si l'entrée est 0, sinon la sortie est 0. En résumé, la sortie est l'inverse de l'entrée, sa forme algébrique : $NON\ (A) = \bar{A}$.
  La figure 1.1 montre la forme de la porte NON.
  Le tableau 1.1 est le tableau de vérité pour la porte NON.

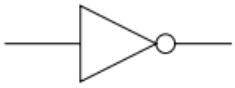

Figure 1.1 :la porte NON.

A	$\bar{A}$
0	1
1	0

Tableau 1.1 : Tableau de verité de la porte NON.

1.2.2 Portes à deux bits :

- **Porte ET(AND) :** met sa sortie à 1 uniquement lorsque ses deux entrées sont à 1, et donne une sortie de valeur 0 si au moins l'une des deux entrées est à 0, sa forme algébrique : F (A, B) = A ET B = A.B = AB.
  La figure 1.2 montre la forme de la porte ET.
  Le tableau 1.2 est le tableau de vérité pour la porte NON.

Figure 1.2 :la porte ET.

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

Tableau 1.2 : Tableau de verité de la porte ET

- **Porte OU(OR) :** Une porte logique qui opère sur deux variables d'entrée donne une valeur de 1 si l'une ou l'autre des entrées (ou les deux) est à 1. Elle donne une valeur de 0 si les deux entrées sont à 0, sa forme algébrique : F (A, B) = A OR B =A + B.

  La figure 1.3 montre la forme de la porte OU.

  Le tableau 1.3 est le tableau de vérité pour la porte OU.

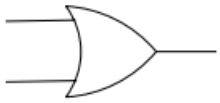

Figure 1.3 : la porte OU

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Tableau 1.3 : Tableau de verité de la porte OU

Porte OU-exclusif (XOR) : produit une sortie de valeur 1 lorsque seulement l'une des deux entrées est à 1, et une sortie de valeur 0 lorsque les deux entrées sont soit toutes les deux à 0, soit toutes les deux à 1, sa forme algébrique : F (A, B) = A XOR B =A$\bigoplus$B = $\bar{A}.B + A.\bar{B}$

La figure 1.4 montre la forme de la porte OU-exclusif.

Le tableau 1.4 est le tableau de vérité pour la porte OU-exclusif.

Figure 1.4 : la porte OU-exclusif

A	B	A$\bigoplus$B
0	0	0
0	1	1
1	0	1
1	1	0

Tableau 1.4 : Tableau de verité de la porte XOR.

1.2.3 Portes universelles :

- **Porte NON ET (NAND) :** Une porte logique qui opère sur deux variables d'entrée prend la valeur 1 lorsque au moins l'une des deux entrées est à 0.cet opérateur est l'inverse de l'opérateur ET, sa forme algébrique : F (A, B) = A NAND B = $\overline{A.B}$.

  La figure 1.5 montre la forme de la porte NON ET.

  Le tableau 1.5 est le tableau de vérité de la porte NON ET.

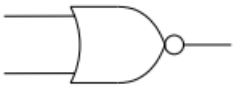

Figure 1.5 : la porte NON ET.

A	B	$\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

Tableau 1.5 : Tableau de vérité de la porte NON ET

Porte NON OU (NOR) : qui fonctionne sur deux variables d'entrée. Il est considéré comme l'inverse de l'opérateur OU, sa forme algébrique : F (A, B) = A NOR B = $\overline{A + B}$. (KHALFI, 2016).

La figure 1.6 montre la forme de la porte NON OU.

Le tableau 1.6 est le tableau de vérité de la porte NON OU.

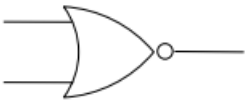

Figure 1.6 : la porte NON OU

A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

Tableau 1.6 : Tableau de verité de la porte NON OU

1.3 Les circuits logiques :

1.3.1 Les circuits combinatoires :

Un circuit combinatoire est composé d'un ensemble de portes logiques interconnectées. Les sorties de ces portes sont reliées aux entrées d'autres portes, formant ainsi un réseau

de connexions unidirectionnelles. Ce réseau de connexions garantit que le circuit est acyclique, c'est-à-dire qu'il n'y a pas de boucles. De cette manière, un circuit combinatoire peut être considéré comme une porte logique avec plusieurs sorties. (S´everine Fratani, 23 septembre 2014)

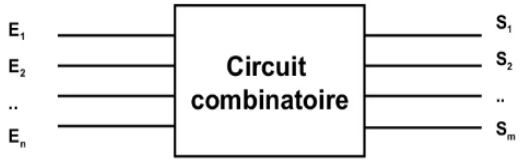

Figure 1.7 : Schéma des circuits logiques combinatoires.

1.3.2 Principaux circuits MSI (Medium Scale Integration):

- **Le multiplexeur** : Le multiplexeur est un dispositif de sélection de données ou d'aiguillage convergent. Il permet de convertir une information représentée par un ensemble de n bits en parallèle en une information séquentielle de n bits. Une voie d'entrée E est sélectionnée à l'aide d'une adresse A, et cette voie est ensuite reliée à la sortie S. En utilisant une adresse spécifique, il est possible de choisir l'une des N voies d'entrée pour acheminer les données vers la sortie S.
- **Le démultiplexeur :** Le démultiplexeur effectue l'opération inverse du multiplexeur. Il dispose d'une entrée d'information ou de données (E), de N entrées de sélection et de $S = 2^n - 1$ sorties. En utilisant une adresse sur les entrées de sélection, on peut choisir la sortie vers laquelle l'entrée sera acheminée. De plus, le démultiplexeur peut être équipé d'une entrée de validation des sorties, similaire au multiplexeur.
- **Décodeur :** Le décodeur est un opérateur à n entrées et 2^n sorties. Chaque sortie est activée en fonction de la configuration du code présente aux entrées. Il peut décoder toutes les configurations possibles du code ou seulement une partie. Une seule sortie est activée à la fois. Certains décodeurs peuvent également avoir des entrées de validation en plus des entrées principales.

1.3.3 Les circuits de calculs :

- ✓ **L'additionneur :** L'addition est une opération arithmétique cruciale dans les systèmes numériques, et les additionneurs sont d'une grande importance pour les opérations de multiplication et de division. Ils permettent de réaliser efficacement ces opérations en manipulant les nombres binaires.
- ✓ **Demi additionneur :** Un demi-additionneur est un circuit logique qui effectue l'addition de deux bits A et B. Il produit une somme S, qui correspond au résultat de

l'addition des deux bits, et une retenue C, qui indique si une retenue est générée lors de l'addition.

- ✓ **Additionneur complet :** Pour additionner deux mots de n bits, il faut utiliser n additionneurs. La retenue se transmet des bits de poids faible vers les bits de poids élevé. (KHALFI, 2016)
- ✓ **L'unité arithmétique et logique :** L'unité Arithmétique et Logique (UAL), qui est le cerveau mathématique d'un processeur, se charge de toutes les opérations de base sur les nombres entiers. À l'époque des premiers ordinateurs, elle était le composant le plus essentiel du processeur. Aujourd'hui, les processeurs contiennent souvent plusieurs UAL, bien que leur rôle reste crucial, ils ne représentent plus qu'une fraction minime de la superficie totale du processeur. (KHALFI, 2016)

2 Limites de l'informatique classique :

La demande croissante d'applications informatiques avancées a nécessité une mise à jour technologique afin de rester en phase avec ces besoins. Les progrès technologiques ont permis le développement d'applications complexes telles que le cloud computing, le traitement en temps réel de vastes bases de données et les calculs biotechnologiques. Au fil des années, les avancées technologiques en termes de fréquence de fonctionnement accrue et de réduction de la taille des puces ont entraîné une augmentation significative de la puissance de calcul, ce qui a favorisé cette croissance.

2.1 Loi de Moore :

La loi de Moore, formulée par Gordon Moore dans les années 1960, a confirmé cette tendance selon laquelle le nombre de transistors dans une puce double environ tous les dix-huit mois en moyenne.

Gordon Moore illustre l'augmentation du nombre de transistors dans la figure 1.8. Selon l'ITRS (International Technology Roadmap for Semiconductors), il est prévu que la taille des transistors atteigne l'échelle atomique d'ici 2050.

La réduction de la taille des transistors a entraîné un ensemble de défis opérationnels tels que la dissipation thermique, la nécessité de faisceaux laser fins, la distribution de l'horloge, et bien d'autres.

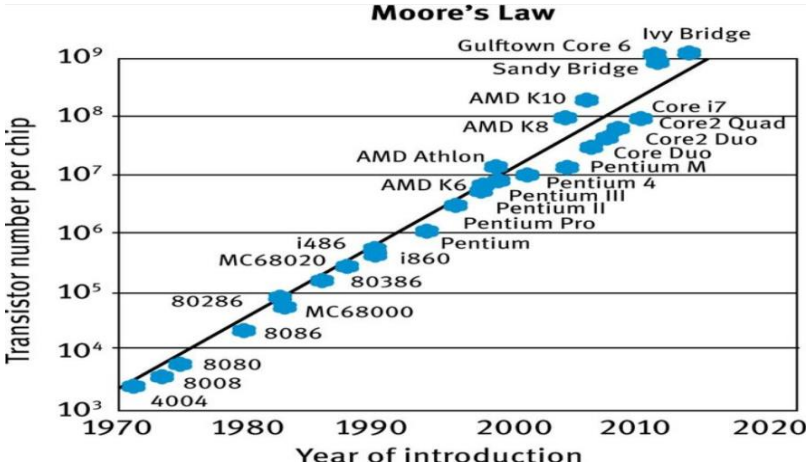

Figure 1.8 : Evolution du nombre de transistors dans le microprocesseur Intel

Pour faire face aux limites des technologies actuelles, une alternative possible pour l'avenir est l'informatique réversible. Cette approche permet de concevoir des systèmes informatiques de nouvelle génération qui préservent les informations précédentes et réduisent la dissipation de chaleur lors de leur fonctionnement.

Pendant plus de 30 ans, la technologie conventionnelle a dominé le monde de l'informatique. Cependant, cette technologie présente une limitation importante : les calculs sont irréversibles. Cela signifie que l'entrée ne peut pas être récupérée à partir de la sortie. Parfois, certaines informations sont perdues ou inversées avant d'atteindre la sortie. Le bit irréversible des calculs dans l'informatique conventionnelle pose plusieurs problèmes, notamment en ce qui concerne les calculs à haute vitesse. Cette limitation remet en question la viabilité de l'informatique conventionnelle pour les générations futures.

Les ressources clés prises en compte dans l'informatique sont le temps, l'espace, le coût de fabrication et l'énergie. Étant donné la demande croissante de vitesse de calcul dans les applications scientifiques, il a été constaté que la logique irréversible des portes conventionnelles n'est pas une technologie viable. En effet, elle entraînerait un certain nombre de problèmes. Les problèmes qui peuvent se produire dans les systèmes informatiques peuvent être regroupés comme suit :

1.1.1 Problèmes physiques : Les dispositifs utilisant des portes classiques irréversibles peuvent entraîner une inefficacité physique dans les systèmes informatiques conventionnels. Voici quelques-uns de ces problèmes courants :

- ✓ **Dissipation de la chaleur :** Selon Landauer (1961), dans la logique irréversible, le fonctionnement d'une porte logique entraîne inévitablement une perte d'énergie, quelle que soit la technologie utilisée. Cette perte d'énergie est quantifiée par la formule kT.Ln2 Joule, où k représente la constante de Boltzmann et T correspond à

la température du système. Cette quantité d'énergie dissipée est associée à chaque bit d'information « perdu » pendant le calcul. L'énergie dissipée est exprimée en fonction de la température ambiante (kT) et est proportionnelle au nombre de réseaux d'informations associés. La perte thermique deviendra significative d'ici 2035, lorsque les transistors deviendront de plus en plus petits. Cependant, en utilisant des opérations réversibles, il est possible d'éviter la dissipation d'énergie de l'ordre de kT par bit. Ainsi, la perte due à la manipulation de bits réversibles pourrait continuer à diminuer à l'avenir.

- ✓ **Incapacité de répondre aux exigences de taille :** Selon la loi de Moore, il est observé, d'après la courbe de l'ITRS (International Technology Roadmap for Semiconductors), que la complexité des calculs augmente tandis que la taille des transistors diminue. De nos jours, les ordinateurs utilisent des puces en silicium. Plus ces puces deviennent petites et rapides, plus la densité d'emballage des puces augmente. On prévoit que d'ici 2050, la taille des transistors nécessaires atteindra l'échelle atomique. Cependant, en raison de la dissipation thermique, il sera difficile d'atteindre une densité d'emballage élevée, car il existe une limite de distance entre les périphériques des bits adjacents. En d'autres termes, les ordinateurs conventionnels ne seront pas en mesure de répondre aux exigences de taille dans un avenir proche.

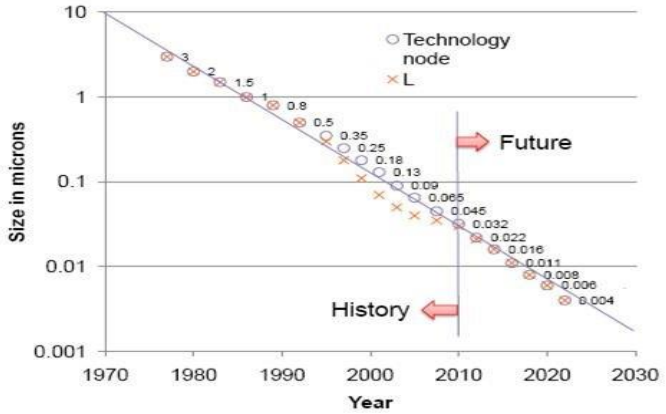

Figure 1.9 : Evolution de la taille de transistors dans le temps.

2.1.2 Problème du temps et de la taille du calcul : L'informatique quantique a la capacité de résoudre des problèmes auxquels l'informatique traditionnelle ne peut plus faire face de manière viable. Ces problèmes sont de nature exponentielle, leur complexité augmente en fonction du nombre de données à traiter. Par exemple : la résolution d'un problème de parcours d'un ensemble de véhicules distributeurs, où chaque véhicule possède un grand nombre de possibilités d'optimisation, dépasse les capacités des algorithmes classiques utilisés dans les ordinateurs traditionnels.

Un autre exemple de problème que l'informatique quantique est capable de résoudre est la factorisation de très grands nombres. Contrairement à l'informatique traditionnelle, l'informatique quantique offre des possibilités de traitement efficace pour ce type de problème. Revenons à notre premier exemple, il s'agit d'un système soumis à de nombreuses contraintes, où l'on dispose de l'ensemble des données associées. Dans ce cas, il est nécessaire d'optimiser théoriquement l'itinéraire individuel de chaque véhicule, en tenant compte de leur point de départ et de leur destination. Les ordinateurs traditionnels, qui utilisent des algorithmes classiques, sont généralement capables de répondre à nos besoins lorsqu'il s'agit d'un nombre limité de véhicules et de trajets. Cependant, dès que le nombre de véhicules et de trajets dépasse quelques centaines, les capacités de calcul des ordinateurs traditionnels atteignent rapidement leurs limites, et ils deviennent saturés, ce qui les rend inefficaces.

La figure 1.10 fournie illustre de manière concrète l'importance de l'informatique quantique en comparant les temps de calcul pour des problèmes extrêmement complexes entre les approches classique et quantique.

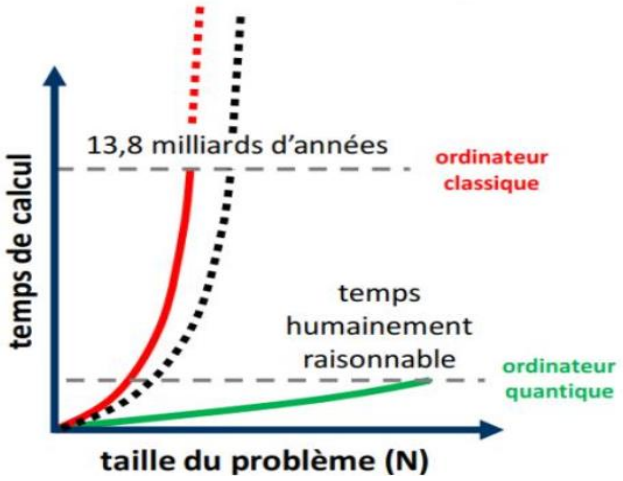

Figure 1.10 : Variation du temps du calcul en fonction de la taille du problème pour le cas classique et celui quantique.

En se basant sur les données présentées dans les figures 1.10 et 1.11, il est clair que les calculateurs d'ordinateurs classiques nécessiteraient un temps de calcul dépassant 13,85 milliards d'années pour résoudre des problèmes plus complexes. Cela est pratiquement impossible. Les temps de calcul pour les problèmes exponentiels restent exponentiels, même avec une hypothétique amélioration des performances des machines tous les 18 mois à deux ans.

En revanche, l'introduction d'un simple qubit, que nous expliquerons en détail plus tard, permettrait théoriquement de doubler les performances d'un ordinateur quantique. En comparaison, les ordinateurs quantiques pourraient théoriquement, un jour, résoudre ces mêmes problèmes dans un laps de temps raisonnable à l'échelle d'une vie humaine, que ce soit en heures, jours, semaines ou mois.

L'un des objectifs principaux du calcul quantique est de réduire considérablement les échelles de temps nécessaires pour résoudre un problème.

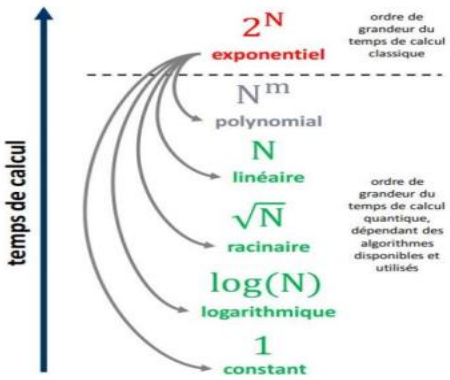

Figure1.11 : Comparaison du temps du calcul de problèmes très complexes entre calcul classique et calcul quantique.

1.2.3 Problème économique : En ce qui concerne l'aspect économique, il est important de noter que l'augmentation de la complexité des calculs entraîne généralement des coûts plus élevés. Cela signifie que plus un problème de calcul est complexe, plus il est coûteux à résoudre. La figure 1.12 présente le nombre d'opérations sur les bits par dollar de calcul pour les calculs irréversibles et réversibles.

La figure 1.12 met en évidence la relation entre la complexité des calculs, les coûts associés et les avantages économiques potentiels de l'informatique réversible. Elle suggère que l'utilisation de l'informatique quantique, avec ses opérations réversibles, peut offrir des efficiences économiques supérieures pour les problèmes complexes par rapport aux calculs classiques irréversibles. (ANDALOUSSI, 2022)

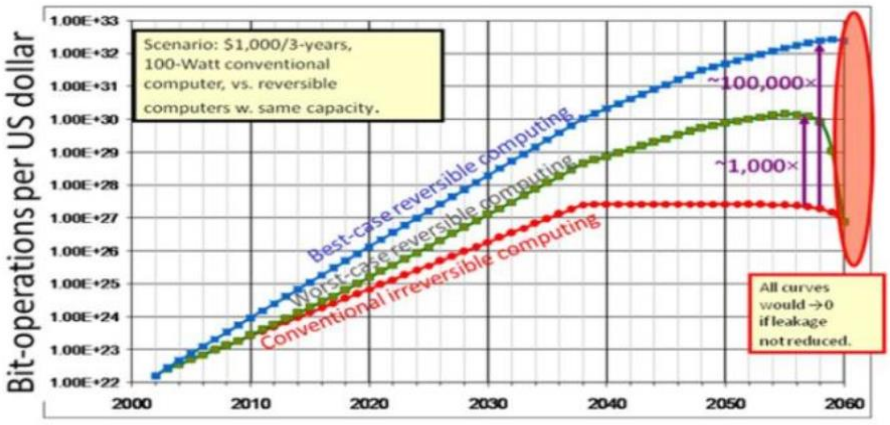

Figure 1.12 : Nombre des opérations sur les bits par dollar de calculs irréversibles et réversibles

3 L'informatique quantique :

L'informatique quantique est un domaine émergent qui combine plusieurs disciplines telles que la physique, le génie, la chimie, l'informatique et les mathématiques. Son objectif principal est de réaliser un ordinateur quantique et d'utiliser cet ordinateur pour effectuer des calculs beaucoup plus rapidement que les ordinateurs classiques traditionnels.

L'accélération des calculs est rendue possible en exploitant les phénomènes quantiques tels que les superpositions d'états, l'intrication et l'interférence quantique. (Blais, decembre 2002)

3.1 La technologie de l'information et la mécanique quantique :

La science et technologie de l'information quantique (STIQ) est un domaine qui fusionne la mécanique quantique et la technologie de l'information. En exploitant les propriétés quantiques des particules subatomiques, la STIQ permet de réaliser des calculs et des communications d'une manière radicalement différente de celle des technologies classiques. Elle ouvre de nouvelles perspectives pour des capacités de calcul massivement parallèles et des systèmes de communication sécurisés. (quantique, 2022)

3.2 Notion de base de l'informatique quantique :

1.1.1 Qubit :

En informatique quantique, le qubit (quantum bit) est l'unité de base pour le support de l'information quantique, noté qubit ou q-bit, pour « Quantum bit ». Le qubit n'a pas de valeur définie, mais plutôt un état quantique. Il peut exister simultanément dans plusieurs états superposés. $|0\rangle$ et $|1\rangle$ prononcé « ket zéro » et « ket un ». (Thomas Cluzel)

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad (1)$$

Forme vectorielle :

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \quad \text{et } |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \qquad (2)$$

Un qubit est représenté par un vecteur :

$$|\psi\rangle = \begin{pmatrix}\alpha\\\beta\end{pmatrix} \qquad (3)$$

En effet :

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix} \qquad (4)$$

Il y a plusieurs types de qubits. Certains sont naturels, d'autres artificiels. Voici quelques-uns des types les plus répandus :

- ✓ **Spin :** La plupart des particules quantiques ont une propriété appelée spin, qui fait qu'elles se comportent comme de petits aimants. Le spin d'une particule peut pointer soit vers le haut, soit vers le bas, mais jamais entre les deux. On peut ainsi coder l'information quantique en utilisant ces deux états de spin : spin up et spin down.
  0 = vers le haut, 1 = vers le bas

Figure 1.13 : États de spin de l'électron.

- ✓ **Atomes et ions piégés :** Les niveaux énergétiques quantifiés des électrons dans des atomes ou ions permettent de coder des qubits : l'état fondamental définit $|0\rangle$ et un état excité défini $|1\rangle$. Des impulsions laser contrôlent alors les transitions entre ces niveaux pour manipuler ces qubits.
  0 = état de faible énergie, 1 = état de forte énergie

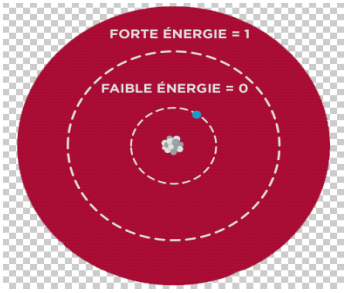

Figure 1.14 : Niveaux d'énergie atomique.

- ✓ **Photons :** Les photons, particules de lumière individuelles, fournissent plusieurs modes de codage possible pour des qubits :

  Qubits de polarisation : Chaque photon possède un champ électromagnétique polarisé selon une orientation définie dans l'espace. Les qubits photoniques s'encodent sur les états de polarisation horizontale et verticale des photons.

  0 = horizontale, 1 = verticale

  Qubits de trajectoire : On peut encoder un qubit sur la superposition quantique de trajets optiques d'un photon unique créée par des séparateurs de faisceaux.

  0 = trajectoire du haut, 1 = trajectoire du bas

  Qubits de moment d'arrivée : Des qubits peuvent être encodés sur la superposition des temps d'arrivée « de manière hâtive » et « de manière tardive » des photons uniques.

  0 = arrivée hâtive du photon, 1 = arrivée tardive du photon.
- ✓ **Circuits supraconducteurs :** Refroidis suffisamment, des matériaux dits supraconducteurs permettent au courant électrique de les traverser sans résistance. Il est possible de concevoir des circuits électriques utilisant des supraconducteurs de manière à ce qu'ils se comportent comme des qubits.

  0 = courant dans le sens horaire, 1 = courant dans le sens antihoraire.

Figure 1.15 : États de rotation quantique.

3.2.2 Dualité onde-corpuscule :

La dualité onde-corpuscule est le concept qui stipule que la matière et la lumière peuvent exhiber à la fois des caractéristiques d'ondes et de particules. La fonction d'onde est utilisée pour décrire la nature ondulatoire et corpusculaire des objets quantiques. Contrairement à une onde d'énergie telle qu'une onde sonore ou une vague, la fonction d'onde décrit une onde de probabilité.

3.2.3 Superposition quantique :

La superposition est une propriété des ondes selon laquelle l'addition d'ondes donne une nouvelle onde. Comme les particules quantiques ont une nature ondulatoire, elles peuvent être dans une superposition quantique de différents états. Par exemple, si une particule est dans une superposition de 2 endroits, elle agit comme si elle était aux 2 endroits en même temps comme si elle était en même temps « ici » et « là », ou « en haut » et « en bas ».

3.2.4 Effet tunnel :

Dans l'expérience de faire rouler une balle sur un monticule, si la balle n'est pas lancée avec une vitesse suffisante, elle ne peut pas passer de l'autre côté. Cependant, si la balle était une particule quantique, elle aurait une faible probabilité de traverser le monticule en raison de son comportement ondulatoire. C'est ce qu'on appelle l'effet tunnel quantique.

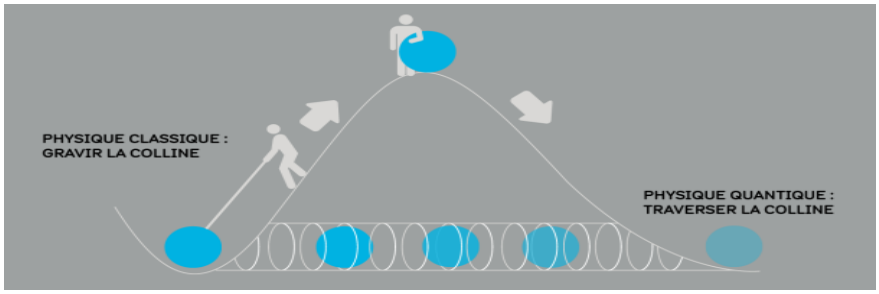

Figure 1.16 : Effet tunnel : classique vs quantique

3.2.5 Décohérence :

Les états quantiques sont sensibles aux interactions non souhaitées avec leur environnement, ce qui peut entraîner une perte de superposition et un effondrement de l'état quantique, similaire à une mesure. Ce phénomène est connu sous le nom de décohérence.

Figure 1.17 : Illustration de la décohérence.

3.2.6 L'intrication :

L'intrication quantique est une corrélation forte entre deux particules qui les décrit par une même fonction d'onde. Elle permet de prédire de manière prévisible les propriétés des particules mesurées ensemble, même à distance. Par exemple, la position de deux électrons intriqués peut être instantanément connue lorsque l'on mesure la position de l'un d'eux. (quantique, 2022)

3.2.7 Sphère de Bloch :

Mathématiquement, l'état d'un qubit est décrit à l'aide de nombres complexes et représenté géométriquement par un point à la surface d'une sphère, appelée sphère de Bloch, qui permet de visualiser cet état quantique. On représente l'état d'un qubit par un vecteur unitaire à deux dimensions dans un espace vectoriel complexe. Ce vecteur d'état du qubit $|\psi\rangle$ est une combinaison linéaire des états de base $|0\rangle$ et $|1\rangle$, avec des coefficients complexes α et β. Dans la sphère de Bloch, l'état $|0\rangle$ d'un qubit correspond à un vecteur unitaire dirigé vers le pôle Nord, tandis que l'état $|1\rangle$ est représenté par un vecteur unitaire vers le pôle Sud. Dans la représentation géométrique par la sphère de Bloch, les états superposés d'un qubit sont figurés par des vecteurs unitaires orientés de façon arbitraire à partir du centre, définis par un angle par rapport à l'axe z (vertical) et un angle azimutal par rapport au plan équatorial x-y.

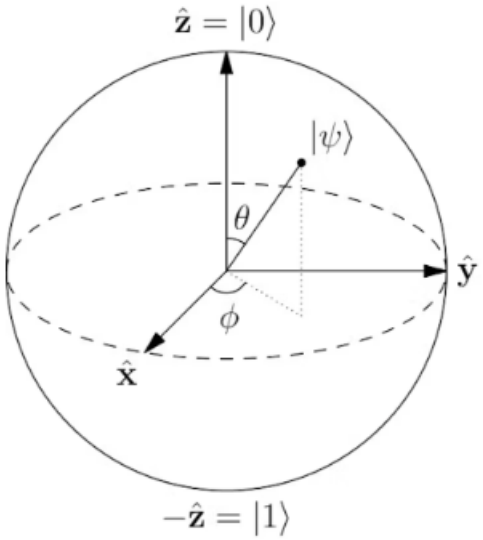

Figure 1.18 : Sphère de Bloch.

3.2.8 Règles de Max Born :

Dans l'équation d'état d'un qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, les nombres complexes α et β déterminent les probabilités de mesurer les états $|0\rangle$ et $|1\rangle$ respectivement : $|\alpha|^2$ est la probabilité d'obtenir $|0\rangle$ et $|\beta|^2$celle d'obtenir $|1\rangle$.

La somme des probabilités des deux états doit donner 1. Ce n'est pas α+β mais $\alpha^2+\beta^2$qui donnent 1. Parce que ce modèle de Born pour l'interprétation de la fonction d'onde, élaboré en 1926, attribue une signification probabiliste au carré du module de la fonction d'onde : il représente la densité de probabilité de présence d'une particule. Ce modèle s'applique aux qubits, bien que leurs états $|0\rangle$ et $|1\rangle$ ne codent pas une position spatiale mais deux états quantiques possibles. La fonction d'onde de Schrödinger décrit la distribution de probabilité temporelle et spatiale de ces états. Ici, on l'utilise pour représenter la probabilité des états $|0\rangle$ et $|1\rangle$.

L'état d'un qubit est représenté par un vecteur à deux dimensions dans un espace de Hilbert, ce qui encode beaucoup plus d'informations qu'un simple 0 ou 1 classique. Ce vecteur est défini par les amplitudes complexes α et β qui représentent les probabilités des états de base. Par convention, on peut choisir α réel et positif, la phase globale du vecteur n'affectant pas la physique. Un qubit porte donc une information bien plus riche qu'un bit classique.

Selon la règle de Max Born, la relation entre α et β est liée à la fonction d'onde de Schrödinger, qui définit les états $|0\rangle$ et $|1\rangle$ d'un qubit. L'état du qubit est représenté par $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, où α est l'amplitude de l'état $|0\rangle$ et β est l'amplitude de l'état $|1\rangle$. La condition $|\alpha|^2 + |\beta|^2 = 1$ doit être satisfaite, ce qui signifie que la somme des carrés des amplitudes doit être égale à 1.

En utilisant la formule d'Euler, on peut exprimer l'exponentielle de $|\psi\rangle$ en termes d'un nombre complexe avec un cosinus et un sinus de l'angle φ.

La forme classique de l'état $|\psi\rangle$ est donnée par $|\psi\rangle = cos\frac{\theta}{2}|0\rangle + sin\frac{\theta}{2}\, e^{i\varphi}|1\rangle$, où θ est l'angle et φ est la phase.

La forme symétrique de l'état $|\psi\rangle$ est donnée par $|\psi\rangle = cos\frac{\theta}{2}.e^{\frac{-i\varphi}{2}}|0\rangle + sin\frac{\theta}{2}.e^{\frac{i\varphi}{2}}|1\rangle$, où θ est l'angle et φ est la phase.

La différence entre ces deux formes réside dans la multiplication par une phase globale ($e^{\frac{-i\varphi}{2}}$), qui n'a pas d'impact sur la mesure de l'état du qubit. Seule la phase relative entre les deux états de base est importante. En d'autres termes, dans la représentation de la sphère de Bloch, cette phase relative entre $|0\rangle$ et $|1\rangle$ est ce qui compte.

La figure 1.19 présente la sphère de Bloch, qui est un modèle mathématique probabiliste utilisé pour une meilleure compréhension des états quantiques.

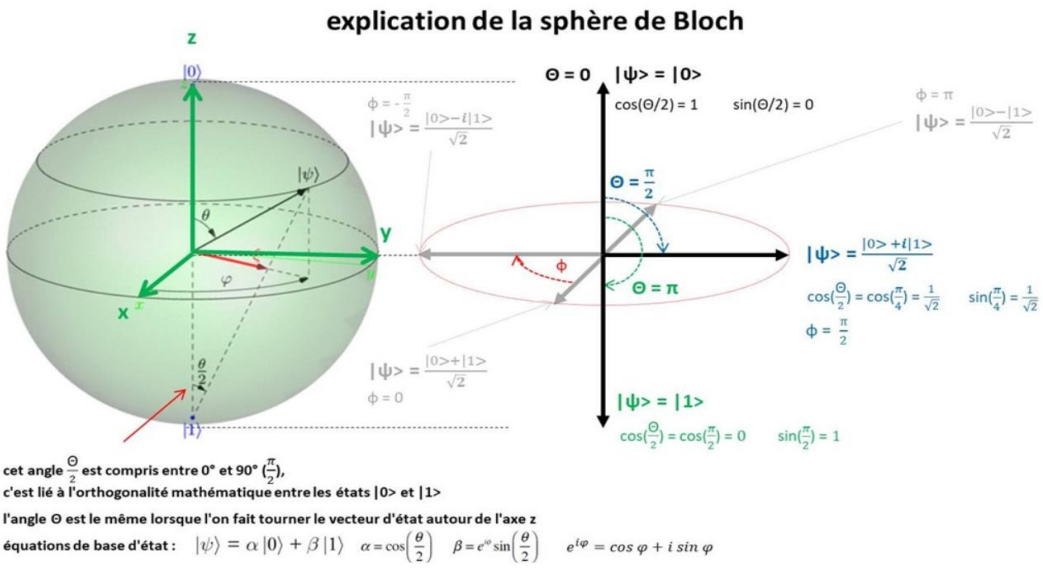

Figure 1.19 : Modèle mathématique probabiliste de la sphère du Bloch.

La figure 1.20 illustre une coupe équatoriale de la sphère de Bloch. Cependant, il est important de noter que cette représentation n'est pas un modèle physique directement lié à des phénomènes tels que l'angle de polarisation d'un photon ou le spin d'un électron, bien qu'il puisse y avoir des similitudes.

Lorsque le vecteur d'état du qubit est aligné horizontalement sur la sphère de Bloch, c'est-à-dire qu'il se situe sur l'équateur, cela correspond à un état de superposition entre l'état |0⟩ et l'état |1⟩, avec une phase relative entre ces deux états qui est liée à l'angle horizontal θ du vecteur par rapport à l'axe z, comme illustré dans le schéma ci-dessous.

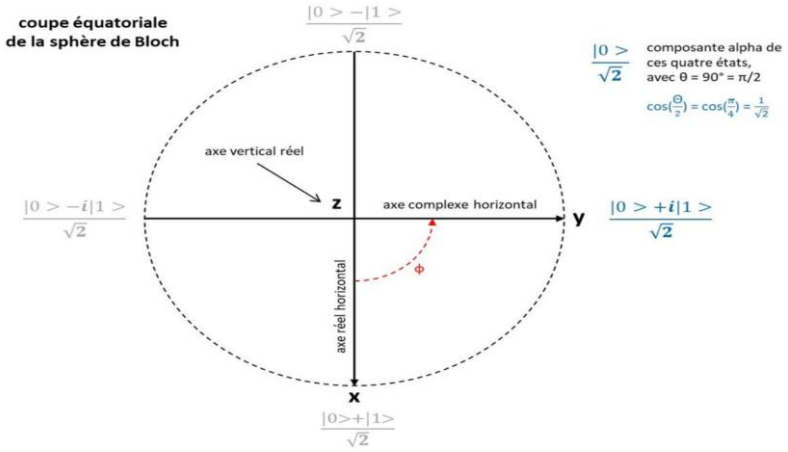

Figure 1.20 : Coupe équatoriale de la sphère du Bloch.

Les portes quantiques permettent de modifier l'information contenue dans un qubit. L'angle (θ/2), qui varie de 0° à 90° (π/2), est lié à l'orthogonalité mathématique entre les états |0⟩ et |1⟩. Peu importe la rotation du vecteur d'état autour de l'axe z, cet angle reste constant. En effet, les équations des états quantiques sont données par $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, où

$\alpha = cos\frac{\theta}{2}$ et $\beta = e^{i\varphi}.sin\frac{\theta}{2}$. Une porte quantique unitaire, qui agit sur un seul qubit, effectue une rotation de l'état du qubit dans la sphère de Bloch. Cette rotation dans la sphère de Bloch est effectuée en utilisant une matrice unitaire de dimension 2x2. Cette matrice est composée de 4 nombres complexes et est orthogonale, ce qui signifie qu'elle conserve la norme du vecteur d'état après son application. Ainsi, la longueur du vecteur d'état reste toujours égale à 1.

En général, les portes quantiques n'effectuent pas toutes les rotations possibles dans la sphère de Bloch. Elles réalisent souvent des rotations d'un demi-tour ou d'un quart de tour.

Les points cardinaux de la sphère de Bloch sont les points les plus couramment utilisés, notamment le pôle nord (état $|0\rangle$), le pôle sud (état $|1\rangle$), ainsi que les quatre points situés sur l'équateur de la sphère qui correspondent aux superpositions des états $|0\rangle$ et $|1\rangle$. Cependant, pour exploiter pleinement la puissance de calcul quantique, il est nécessaire d'effectuer des rotations plus petites que des quarts de tours en utilisant des portes R à phase variable, que nous aborderons ultérieurement.

Un autre point clé est que les vecteurs opposés sur la sphère de Bloch sont toujours mathématiquement orthogonaux. C'est le cas des états $|+\rangle$ et $|-\rangle$ qui se situent sur l'équateur de la sphère.

La représentation géométrique des états quantiques à deux niveaux sur une sphère (sphère de Bloch en mécanique quantique, sphère de Poincaré en optique) trouve son origine dans trois contributions majeures : la fonction d'onde de Schrödinger (1926), l'interprétation probabiliste de Born (1926), et la proposition de Bloch (1946) de représenter un spin ½ sur une sphère. Cette sphère permet de visualiser complètement l'état à deux dimensions d'un tel système quantique. En optique, elle porte le nom de Poincaré pour décrire la polarisation des photons. (ANDALOUSSI, 2022)

3.3 Les portes logiques quantiques :

En informatique quantique, les qubits sont manipulés à l'aide de portes quantiques (également connues sous le nom de portes logiques quantiques). Ces portes quantiques, telles que décrites par Barenco et al. En 1995, permettent d'effectuer des opérations sur les qubits, ce qui permet de modifier l'état superposé d'un système quantique sans provoquer de décohérence.

L'état quantique de n qubits peut être décrit par un vecteur à 2n dimensions dans un espace de Hilbert, avec des composantes complexes.

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

À la différence des portes logiques classiques définies par leur table de vérité, Les portes quantiques sont définies par des matrices unitaires de dimension 2^n, où n est le nombre de qubits manipulés. (Thomas Cluzel)

1.1.2 Les portes à un qubit :

Les portes quantiques à un qubit sont définies comme des matrices unitaires 2x2 agissant sur les vecteurs d'états des qubits dans l'espace d'Hilbert C^2. Quelques-unes de ces portes à un qubit jouent un rôle particulièrement crucial.

- ✓ **Les matrices de Pauli :** Les matrices de Pauli sont des matrices unitaires 2x2 hermitiennes, couramment employées en informatique quantique pour transformer les états des qubits. Appelées aussi portes de Pauli, elles sont très utiles pour faire tourner les vecteurs d'états dans l'espace de Hilbert, et donc manipuler les qubits.

- ✓ **La porte de Pauli X :** La matrice de Pauli X = $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ fait tourner de π radians le vecteur d'état d'un qubit sur la sphère de Bloch autour de l'axe x. Elle correspond donc à l'équivalent quantique de la porte logique classique NON (NOT) : elle fait tourner $|0\rangle$ en $|1\rangle$ et vice-versa.

  La figure 1.21 montre la forme de la porte X.

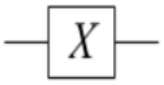

Figure 1.21 : la porte X.

- ✓ **La porte de Pauli Y :** La matrice de Pauli Y= $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ applique une rotation de π radians du vecteur d'état d'un qubit sur la sphère de Bloch selon l'axe y. Cette opération convertit l'état $|0\rangle$ en $i|1\rangle$ et $|1\rangle$ en $-i|0\rangle$, avec i le nombre imaginaire.
  La figure 1.22 montre la forme de la porte Y.

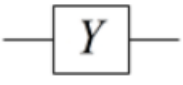

Figure 1.22 : la porte Y

✓ **La porte de Pauli Z :** La matrice de Pauli Z= $\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$ applique une rotation de π radians du vecteur d'état d'un qubit le long de l'axe z sur la sphère de Bloch. Cette opération laisse inchangé l'état |0⟩ et transforme l'état |1⟩ en -|1⟩.

La figure 1.23 montre la forme de la porte Z.

Figure 1.23 : La porte Z.

✓ **Porte de Hadamard :** La porte de Hadamard est considérée comme l'une des portes les plus fondamentales en informatique quantique. Elle permet de transformer les états de base |0⟩ en une superposition $\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$ et |1⟩ en une superposition $\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}$. Pour ces deux états remarquables, la probabilité de mesurer les états |0⟩ et |1⟩ est la même. Géométriquement, cela correspond à une rotation de π autour de l'axe défini par le vecteur $\vec{n} = \frac{(\vec{x}+\vec{z})}{\sqrt{2}}$, où $\vec{x}$ et $\vec{z}$ sont les vecteurs unitaires le long des axes x et z respectivement, sa forme matricielle est :H = $\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$. (N, 2021)

La figure 1.24 montre la forme de la porte Hadamard.

Figure 1.24 : la porte Hadamard

✓ **Les portes V et V^+ :** Ont été développées par Hung et ses collègues en 2006. La porte V est une porte de dimension 1×1 qui applique l'opération $V = \frac{1+i}{2}\begin{pmatrix}1 & -i\\ -i & 1\end{pmatrix}$ À la valeur d'entrée.

En comparaison, la porte V+ (également de dimension 1×1) applique l'opération$V^+ = \frac{1-i}{2}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix}$.

La figure 1.25 montre la forme des portes V et V^+

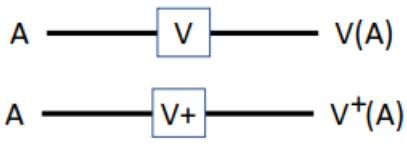

Figure 1.25 : les portes V et V^+

Dans cette étude, seules les bases standard 0 et 1 sont utilisées, conduisant ainsi à quatre scénarios possibles : $V\,(0)$, $V\,(1)$, $V^{+}(0)$, et $V^{+}(1)$. Les résultats de chacun de ces configurations sont exposés ci-dessous :

$$V(0) = \frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{(1+i)}{2}\begin{pmatrix} 1 \\ -i \end{pmatrix}$$

$$V(1) = \frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1+i}{2}\begin{pmatrix} -i \\ 1 \end{pmatrix}$$

$$V^{+}(0) = \frac{1-i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1+i}{2}\begin{pmatrix} 1 \\ i \end{pmatrix}$$

$$V^{+}(1) = \frac{1-i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1+i}{2}\begin{pmatrix} i \\ 1 \end{pmatrix}$$

En considérant ces possibilités, on peut déduire trois propriétés essentielles des portes V et V^{+} :

(1) $V(A) \times V(A) = NOT\ (A)$

(2) $V^{+}(A) \times V^{+}(A) = NOT(A)$

(3) $V(A) \times V^{+}(A) = V^{+}(A) \times V(A) = A$

Ces propriétés sont largement exploitées dans plusieurs études pour simplifier et réduire les circuits quantiques. (Francisco ORTS, (2019))

3.3.2 Les portes à n qubits :

La manipulation de systèmes à plusieurs qubits est l'une des parties les plus cruciales de l'information quantique. Cela implique d'effectuer des opérations sur plusieurs qubits simultanément. (Sujata S. Chiwande, 2011).

- ✓ **Les portes Controlled-V et Controlled-V^{+} :** Ces portes ressemblent aux portes V et V^{+}, mais elles incluent un qubit de contrôle, comme expliqué dans la porte CNOT. Elles restent des portes 2×2 et conservent les propriétés des portes V et V^{+}, mais peuvent être activées ou désactivées à l'aide du qubit de contrôle de manière pratique. (Francisco ORTS, (2019)), La formule matricielle pour les deux portes C-V et C-V^{+} s'écrit comme suit :

$$V = \frac{1+i}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 0 & i & 1 \end{pmatrix} \qquad V^{+} = \frac{1-i}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & i \\ 0 & 0 & i & 1 \end{pmatrix}$$

La figure 1.26 Montre la forme de la porte C-V et C-V+

Figure 1.26 : les portes C- V et C-V^{+}

✓ **Porte Controlled-NOT (CNOT)** : La porte CNOT ou Feynman est une porte logique quantique réversible à deux qubits. Ses entrées sont les bits A et B, et ses sorties sont les bits P et Q définis comme suit : P = A et Q = A ⊕ B.

Le coût quantique associé à une porte de Feynman est de 1. (Sujata S. Chiwande, 2011)

Sa forme matricielle est donne par :

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

La figure 1.27 montre la forme de la porte CNOT.

Le tableau 1.7 est le tableau de vérité de la porte CNOT.

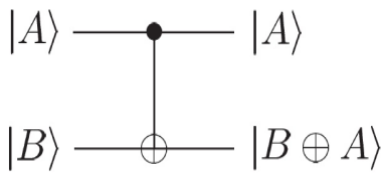

Figure 1.27 : La porte CNOT

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Tableau 1.7 : Tableau de vérité de la porte CNOT.

✓ **Porte SWAP :** La porte SWAP est une porte quantique qui permet d'effectuer l'échange ou la permutation de deux qubits, sa forme matricielle est donne par :

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La figure 1.28 montre la forme de la porte SWAP.

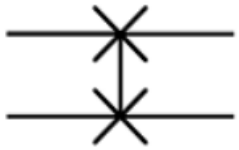

Figure 1.28 : La porte SWAP.

- ✓ **Porte de Fredkin :** La porte de Fredkin est une porte logique réversible à trois entrées et trois sorties, Ses entrées sont les bits A, B et C, et ses sorties sont les bits P, Q et R définis comme suit : P=A, Q = $\bar{A}B \oplus AC$ et R = $AB \oplus \bar{A}C$.

Le coût quantique associé à une porte de Fredkin est de 5. (Sujata S. Chiwande, 2011)

Sa forme matricielle est donne par :

$$\text{Fredkin} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

La figure 1.29 montre la forme de la porte Fredkin.

Le tabeau 1.8 est le tableau de vérité de la porte Fredkin.

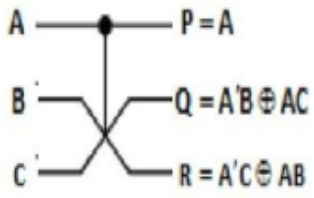

Figure 1.29 : La porte Fredkin.

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Tableau 1.8 : Tableau de vérité de la porte Fredkin

- **Implémentation quantique de la porte Fredkin :** le coût de la porte de Fredkin est constitué de 2 rectangles en pointillés, 1 porte Controlled-V et 2 portes CNOT, aboutissant à un coût quantique de 5. (HIMANSHU THAPLIYA, (2010))

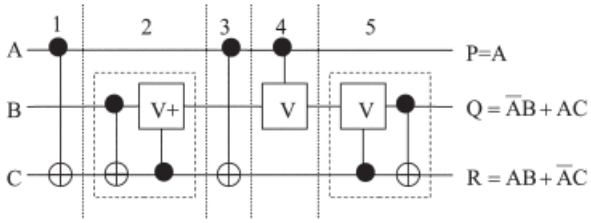

Figure 1.30 : Implémentation quantique de la porte Fredkin.

✓ **Porte Toffoli :** La porte de Fredkin est une porte logique réversible à trois entrées et trois sorties, Ses entrées sont les bits A, B et C, et ses sorties sont les bits P, Q et R définis comme suit : $P = A$, $Q = B$ et $R = AB \oplus C$.

Le coût quantique associé à une porte de Toffoli est de 5. (Sujata S. Chiwande, 2011)

Sa forme matricielle est donnée par :

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

La figure 1.31 montre la forme de la porte Toffoli.
Le tableau 1.9 est le tableau de vérité de la porte Toffoli.

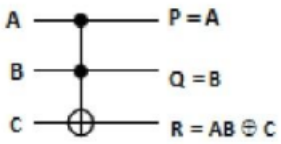

Figure 1.31 : La porte Toffoli.

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Tableau 1.9 : Tableau de vérité de la porte Toffoli

- **Implémentation quantique de la porte Toffoli :** La porte Toffoli a un coût quantique de 5, étant donné qu'elle requiert l'utilisation de 2 portes Controlled-V, 1 porte Controlled-V+ et 2 portes CNOT pour son implémentation. (HIMANSHU THAPLIYA, (2010))

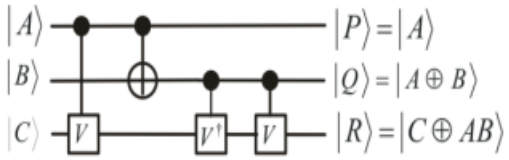

Figure 1.32 : Implémentation quantique de la porte Toffoli

✓ **Porte Peres :** La porte de Peres est une porte logique réversible à trois entrées et trois sorties, Ses entrées sont les bits A, B et C, et ses sorties sont les bits P, Q et R définis comme suit : P = A, Q =A ⊕ B et R = AB ⊕ C.

Cette porte se construit à partir d'une porte de Toffoli et d'une porte de CNOT, d'où un coût quantique de 4. (Sujata S. Chiwande, 2011).

Cette porte se construit à partir d'une porte de Toffoli et d'une porte de CNOT, d'où un coût quantique de 4. (Sujata S. Chiwande, 2011).

La figure 1.33 montre la forme de la porte Peres.

Le tableau 1.10 est le tableau de vérité de la porte Peres.

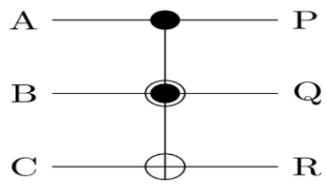

Figure 1.33 : La porte Peres

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Tableau 1.10 : Tableau de vérité de la porte Peres

- **Implémentation quantique de la porte Peres :** Le coût quantique de la porte de Peres est de 4, car sa mise en œuvre implique l'utilisation de 2 portes Controlled-V+, 1 porte Controlled-V, et 1 porte CNOT. (HIMANSHU THAPLIYA, (2010))

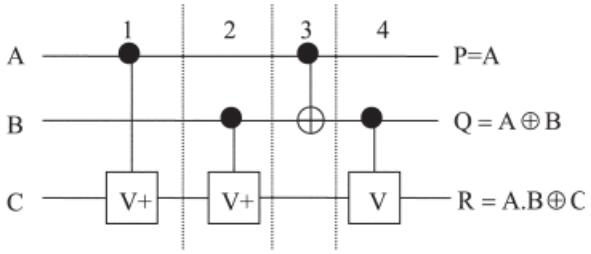

Figure 1.34 : Implémentation quantique de la porte Peres.

3.4 Les algorithmes quantiques :

Les algorithmes quantiques sont des programmes informatiques conçus pour être exécutés sur des ordinateurs quantiques. Ils exploitent les principes de la mécanique quantique, tels que la superposition et l'intrication, pour résoudre certains problèmes beaucoup plus rapidement que les algorithmes classiques. Les exemples célèbres incluent l'algorithme de Shor pour la factorisation des nombres et l'algorithme de Grover pour la recherche dans des bases de données non structurées. Ces algorithmes promettent des avancées significatives dans des domaines tels que la cryptographie, l'optimisation et la simulation de systèmes quantiques complexes.

3.4.1 Algorithme de shor :

L'algorithme de Shor est un algorithme quantique célèbre conçu par le mathématicien Peter Shor en 1994. Il est particulièrement important en cryptographie car il permet de factoriser les grands nombres entiers en leurs facteurs premiers de manière exponentiellement plus rapide que les meilleurs algorithmes classiques connus.

3.4.2 Algorithme de Grover :

L'algorithme de Grover, introduit par Lov Grover en 1996, est un algorithme de recherche d'un élément dans une liste qui est plus efficace que les algorithmes classiques. Son principe est simple, même si sa mise en œuvre est un peu complexe. L'algorithme de Grover ne fournit pas un résultat sûr à 100 %, mais une réponse qui a de grandes chances d'être la bonne. (N, 2021)

Conclusion :

Dans ce chapitre, nous avons passé en revue l'informatique classique et ses limites, ainsi que l'informatique quantique et ses portes quantiques, qui sont les concepts de base pour réaliser un circuit quantique. Dans le chapitre suivant, nous verrons comment réaliser un circuit quantique en combinant des portes quantiques et comment concevoir une UAL quantique.

Chapitre 02 :

Conception des circuits quantiques et d'UAL quantique

Introduction :

L'unité arithmétique et logique (UAL) est la composante fondamentale de l'unité centrale de traitement (CPU), essentielle à tous les ordinateurs actuels. Elle agit comme le cœur permettant de réaliser toutes les opérations arithmétiques et logiques possibles, telles que l'addition, la soustraction et les opérations logiques.

Dans ce chapitre, nous examinerons les circuits d'unité arithmétique et logique (UAL) les plus réputés et les plus performants. Nous traiterons naturellement du circuit de l'additionneur, qui est un élément clé de ces unités. En conclusion, nous évaluerons l'efficacité de chaque circuit en utilisant les critères mentionnés au début de ce chapitre, selon les normes d'optimisation. Chaque circuit sera étudié individuellement, nous calculerons les indicateurs de performance (QC, QD, QG) pour chacun, et nous ajouterons un autre facteur important pour la qualité du circuit : le nombre d'opérations effectuées par chaque UAL.

1 Circuit quantique :

Un circuit logique représente une fonction booléenne qui prend des bits en entrée et produit des bits en sortie. En informatique quantique, les qubits ne se déplacent pas à travers des portes, ce sont les portes quantiques qui sont appliquées aux systèmes de qubits. Une porte quantique ne génère pas de sortie, mais modifie plutôt ses entrées, ce qui signifie qu'un circuit quantique a toujours le même nombre de sorties que d'entrées. (Thomas Cluzel)

1.1 Circuit quantique réversible :

1.1.1 Bibliothèque NCT (Library NCT) : Le circuit réversible est une structure en cascade de portes réversibles, et la combinaison de différentes portes réversibles donne lieu à une bibliothèque variée de portes. Parmi les portes réversibles fondamentales, on trouve la porte NOT (similaire aux portes conventionnels), la porte CNOT ou la porte FEYNMAN, et la porte TOFFOLI. La famille de ces portes est connue sous le nom de bibliothèque NCT.

1.1.2 Bibliothèque NCV (Library NCV) : La bibliothèque NCV, introduite par Barenco et ses collaborateurs, est essentielle dans les circuits quantiques. Grâce à cette bibliothèque, il est possible de réaliser n'importe quelle fonction booléenne réversible, ce qui lui confère le statut de bibliothèque universelle. Les portes quantiques de cette bibliothèque comprennent les portes NOT, CNOT, C-V et C-V+ contrôlée. (Handique, 2017)

1.1.3 Les termes de base de la logique réversible : Il existe de nombreux paramètres pour déterminer la performance et la complexité des circuits.

- **Coût quantique :** Le coût d'un circuit en termes du nombre de portes primitives (1x1, 2x2) est appelé son coût quantique, nous le symbolisons par le symbole QC, abréviation du mot « Quantum Cost »
- **Déchets quantiques :** Dans un circuit logique réversible, le nombre de sorties inutilisées ne peut pas être évité. Elles sont donc essentielles pour maintenir la réversibilité et sont appelées sorties inutilisées, Nous le symbolisons par le symbole QG, abréviation du mot « Quantum Garbage »
- **Délai :** est le nombre maximum de portes quantiques dans un chemin critique depuis n'importe quelle ligne d'entrée vers n'importe quelle ligne de sortie. Chaque porte quantique élémentaire effectue le calcul en une unité de temps. Il représente la profondeur logique du circuit. Les retards des portes quantiques 1*1 et 2*2 sont égaux à un. Nous le symbolisons par le symbole QD, abréviation du mot « Quantum Delay ». (Surekha, 2017)

1.2 Circuit quantique arithmétique :

1.2.1 Additionneur complet : Pour implémenter un additionneur complet (full adder) en circuit quantique réversible, (Poornashree S J, 2021)on définit une fonction à 4 qubits, les entrées sont : A, B, Carry in (la retenue entrante) et $|0\rangle$ (qubit auxiliaire initialisé a 0) et les sorties sont A(inchangé), B(inchangé), Sum et Carry Out (la retenue sortante) ou :

$$\begin{cases} Sum = A \oplus B \oplus C_{in} & (2.1) \\ Carry_{out} = C_{in}(A \oplus B) \oplus AB & (2.2) \end{cases}$$

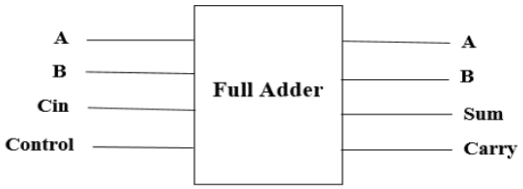

Figure 2.1 : Un diagramme de blocs d'un additionneur quantique complet.

1.2.2 Additionneur complet simple : Cet additionneur repose sur deux circuits quantiques fondamentaux, les circuits SUM et CARRY, qui calculent la somme de deux qubits et conservent le résultat de leur addition.

- **Circuit SUM** :

- ✓ Calcule la somme des deux qubits d'entrée.
- ✓ Utilise deux portes logiques quantiques telles que CNOT pour réaliser ce calcul.
- ✓ Le résultat est la valeur de la somme.

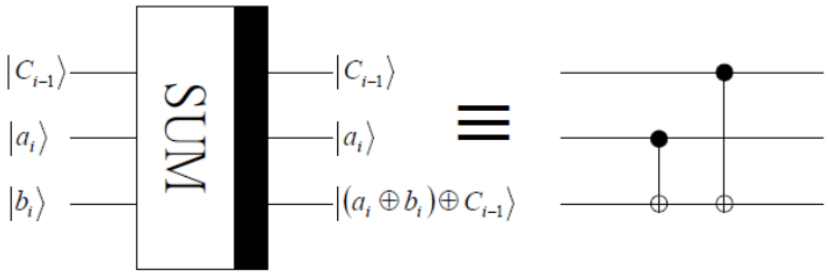

Figure 2.2 : La porte quantique SUM et sa représentation quantique.

- **Circuit CARRY** :

Ce circuit calcule la retenue sortant résultant C_{i+1} de l'addition des deux nombres A et B avec la retenue entrante C_i, comme le montre la figure 2.3.

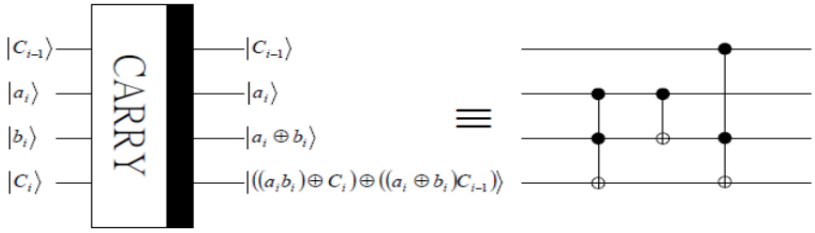

Figure 2.3 : La porte quantique CARRY et sa représentation quantique.

L'additionneur quantique simple est conçu en connectant les circuits précédents de manière séquentielle, comme illustré par la figure 2.4 qui représente l'additionneur quantique pour quatre qubits. À chaque niveau de qubit, la somme (SUM) et la retenue (CARRY) sont calculés. Chaque fois que la retenue est calculée, il est transmis au qubit suivant par le circuit.

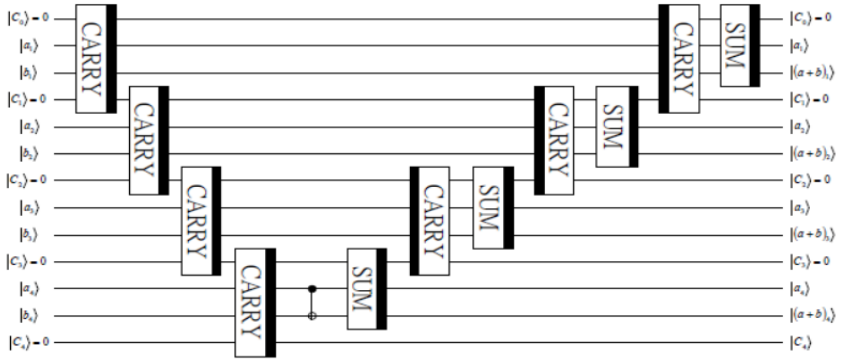

Figure 2.4 : Circuit d'additionneur quantique avec 4 qubits

La figure précédente met en évidence les coûts élevés de construction, car pour chaque niveau de qubit, il faut utiliser 4 portes Toffoli et 1 porte CNOT. Ainsi, le coût pour un seul qubit (QC) est de 23, (QD) est de 23 et (QG) est de 2. Par conséquent, le coût total du circuit

d'additionneur quantique simple à quatre qubits est de QC = 88. Comme il n'y a pas de parallélisme dans l'exécution des opérations, QD est égal à QC, soit 88. (Vedral, 1996)

1.2.3 Conception Améliorée d'un Circuit d'Additionneur Quantique Simple :

Comme mentionné précédemment, le coût du circuit de l'additionneur quantique simple est relativement élevé. Pour optimiser ce coût en termes de QD, QC et le nombre de portes utilisées, un nouveau modèle amélioré est proposé. Ce modèle utilise une combinaison de portes réversible pour réduire le coût global. Voici une explication détaillée de ce modèle. (Zhou, 2015)

- **Additionneur complet pour un qubit :**

Le modèle proposé pour l'additionneur complet pour un qubit unique calcule la somme (Sum) et la retenue (Carry). Ce circuit est composé d'une porte Toffoli, d'une porte CNOT, et d'une porte Peres, comme illustré dans la figure 2.5 :

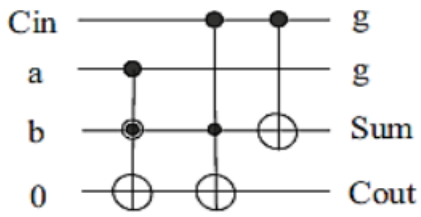

Figure 2.5 : Circuit de l'Additionneur Quantique Amélioré.

Modèle général amélioré de l'additionneur quantique :

La figure 2.6 montre le modèle général amélioré de l'additionneur quantique simple. Ce modèle se compose de n portes Toffoli, n portes CNOT et n portes Peres, par conséquent le cout global est de QC = 10n, QD = 10n et QG = n + 1. Nous pouvons en déduire les valeurs de coût Pour un additionneur à 4 qubits : QC = 40, QD = 40 et QG = 5

En comparant ces résultats avec les précédents, on peut clairement observer une amélioration significative.

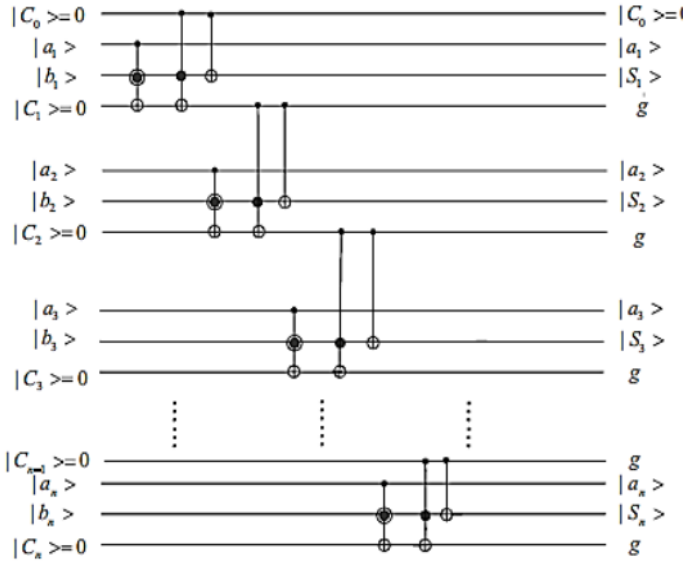

Figure 2.6 : Circuit de l'Additionneur Quantique Amélioré (Modèle Général)

- **Conversion de l'addition en soustraction :**

Comme mentionné précédemment, la soustraction peut être dérivée de l'addition. Pour ce modèle, les équations (2.1) et (2.2) sont transformées en équations (2.3) et (2.4) comme suit :

$$B_i = \bar{a}_i(b_i \oplus B_{i-1}) \oplus b_i B_{i-1} = \bar{a}_i b_i \oplus \bar{a}_i B_{i-1} \oplus b_i B_{i-1} = \bar{a}_i b_i \oplus (\bar{a}_i \oplus b_i) B_{i-1} \quad (2.3)$$

$$D_i = \overline{\bar{a}_i \oplus b_i \oplus B_{i-1}} \quad (2.4)$$

Sur la base des équations (2.3) et (2.4), on peut obtenir l'opération de soustraction à partir de l'addition en inversant $\overline{|B\rangle}$ puis en inversant la somme.

Pour réaliser cette négation, deux portes CNOT sont ajoutées à chaque niveau de qubit : l'une pour inverser l'un des qubits et l'autre pour inverser la somme. Ces deux portes sont contrôlées par les fils de contrôle C_{SUB} et C_{SNOT} .(Zhou, 2015)

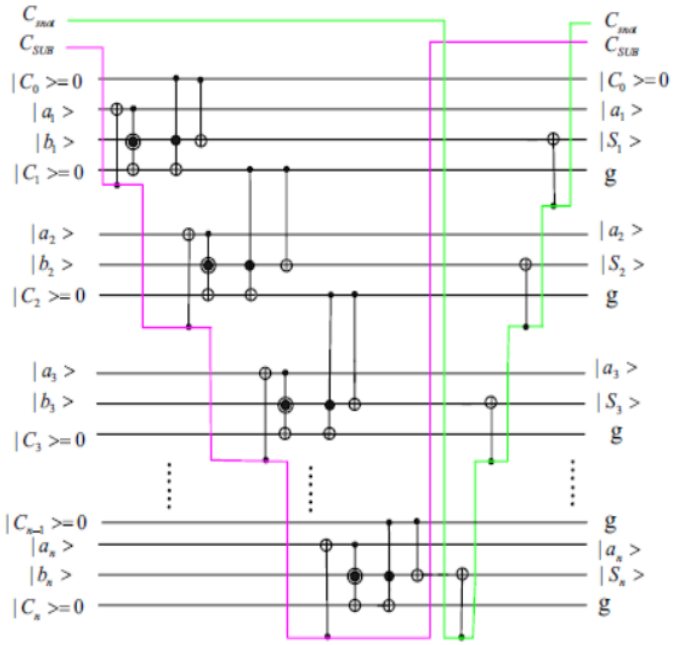

Figure 2.7 : Soustraction en Ajoutant des Fils de Contrôle

L'opération de soustraction se produit lorsque ces fils de contrôle sont dans l'état$|1\rangle$.

1.2.4 L'additionneur complet (modèle de CDKM) :

L'additionneur complet (l'additionneur pour un seul qubit) tel qu'illustré dans la figure 2.8, qui est composé de deux portes logiques principales : la porte MAJ (figure 2.9) et la porte UMA (figure 2.10).

Chacune des deux portes est composée de portes de Toffoli et deux portes CNOT. Ainsi, le coût total d'additionneur complet est de QD = 14, QC = 14.

La porte UMA transforme les entrées $(|a_i\rangle, |b_i\rangle, |c_i\rangle)$ en sorties $(|a_i \oplus c_i\rangle, |a_i \oplus b_i\rangle, |c_{i-1}\rangle)$

La porte MAJ transforme les entrées $(|a_i \oplus c_i\rangle, |a_i \oplus b_i\rangle, |c_{i-1}\rangle)$ en sorties $(|a_i\rangle, |b_i\rangle, |c_i\rangle)$.

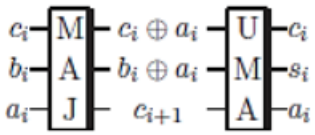

Figure 2.8 : Circuit d'additionneur complet

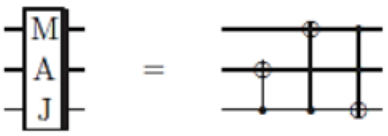

Figure 2.9 : La porte quantique MAJ

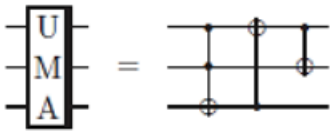

Figure 2.10 : La porte quantique UMA

Le cout du circuit pour un additionneur à un qubit est le cout d'un additionneur complet multipliée par 6 plus une porte CNOT (+1). Par conséquent, nous pouvons déduire le cout de l'additionneur quantique (modèle CDKM) pour quatre qubits comme suit : QG = 1, QD = 57, et QC = 57.

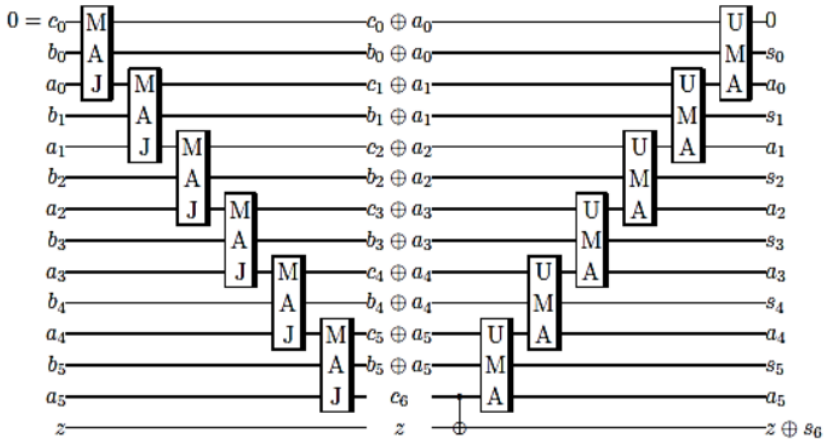

Figure 2.11 : Circuit d'additionneur Complet (CDKM)

1.2.5 L'additionneur complet (modèle de VanRentergem) :

Ce modèle est considéré comme l'un des meilleurs en termes de nombre de portes utilisées, comme illustré à la figure suivante :

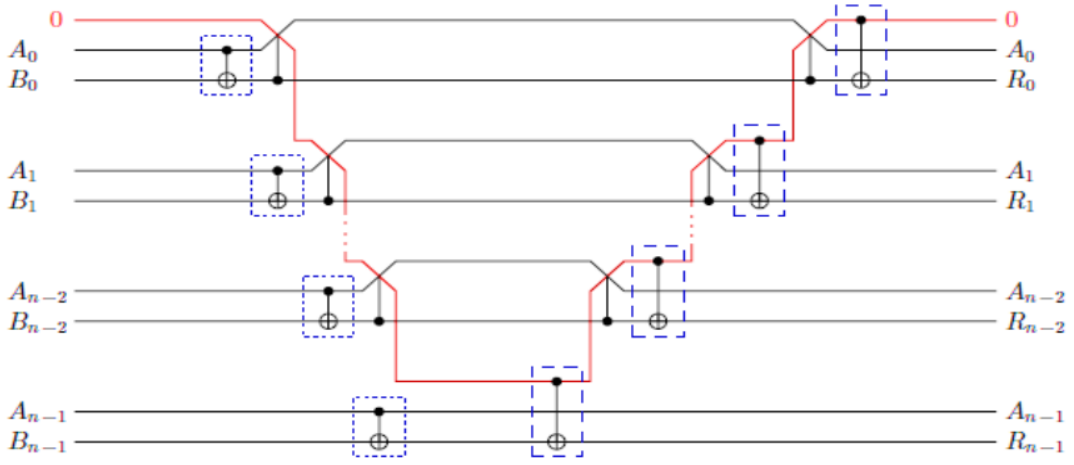

Figure 2.12 : Circuit d'additionneur quantique (VanRentergem)

L’additionneur complet dans ce modèle utilise deux portes Fredkin et deux portes CNOT. Le coût du circuit pour un additionneur complet est donc : QD = 12, QC = 12.

La figure 2.12 montre que l'additionneur quantique pour n qubits, où les deux portes Fredkin sont supprimées du dernier qubit du circuit car elles ne sont pas nécessaires. Ainsi, le cout général de l'additionneur est la suivante : QC = 12n - 10, QD = 10n - 4, et QG=1.

Et donc, on peut déduire le cout de l'additionneur quantique pour quatre qubits : QD = 36, QC = 38, et QG = 1. Cela est considéré comme l'un des meilleurs résultats par rapport aux résultats précédents.

Ces valeurs montrent une efficacité supérieure comparée aux résultats précédents.

Nous remarquons que le premier fil est représenté en rouge car il sera utilisé plus tard dans le circuit de l'unité arithmétique et logique quantique comme fil de contrôle, ce qui rend le circuit sans interférences. (Candidato, 2020)

2 L’Unité Arithmétique et Logique Quantique :

L’Unité Arithmétique et Logique Quantique (UAL) est l’équivalent quantique de l’UAL classique dans les systèmes informatiques quantiques. Tout comme son homologue classique, l’UAL quantique est chargée d’effectuer une variété d’opérations, mais dans le cadre de la mécanique quantique.

Contrairement à l’UAL classique qui traite des données classiques sous forme de bits, l’UAL quantique manipule des qubits, les unités fondamentales d’information quantique. Cela lui permet d’effectuer des opérations sur des superpositions de valeurs et d’exploiter les phénomènes quantiques tels que l’interférence et l’entrelacement pour réaliser des calculs complexes de manière simultanée et parallèle.

L’UAL quantique joue un rôle crucial dans les ordinateurs quantiques et les systèmes de traitement de l’information quantique, où elle est utilisée pour exécuter des algorithmes quantiques et des calculs sur des données quantiques. Son développement est essentiel pour l’avancement de l’informatique quantique et l’exploitation des capacités uniques offertes par le monde quantique. (Khaled El-Wazan, 2021).

2.1 Unité arithmétique et logique quantique (modèle TGA) :

Le modèle TGA de l'unité arithmétique et logique quantique est basé sur le circuit de l'additionneur quantique (modèle Vanrentergem), car il est considéré comme l'un des meilleurs

modèles en termes de nombre de portes utilisées et de quantité de déchets quantiques, ce qui rend ce circuit d'additionneur très élégant.

Pour transformer le circuit de l'additionneur en une ALU, il faut ajouter une unité de contrôle, qui consiste en des fils ajoutés au circuit de l'additionneur et utilisés pour contrôler l'opération souhaitée. Il est également possible d'ajouter ou de remplacer certaines portes pour faciliter le contrôle.

La figure 2.13 montre le circuit de l'unité arithmétique et logique (modèle TGA), où les lignes colorées représentent l'unité de contrôle et les lignes noires représentent le circuit de l'additionneur (figure 2.12). Les lignes de contrôle permettent de déduire la tâche qu'elles accomplissent à partir du nom de la ligne indiqué dans la figure 2.13 comme suit. (Thomsen, 2010)

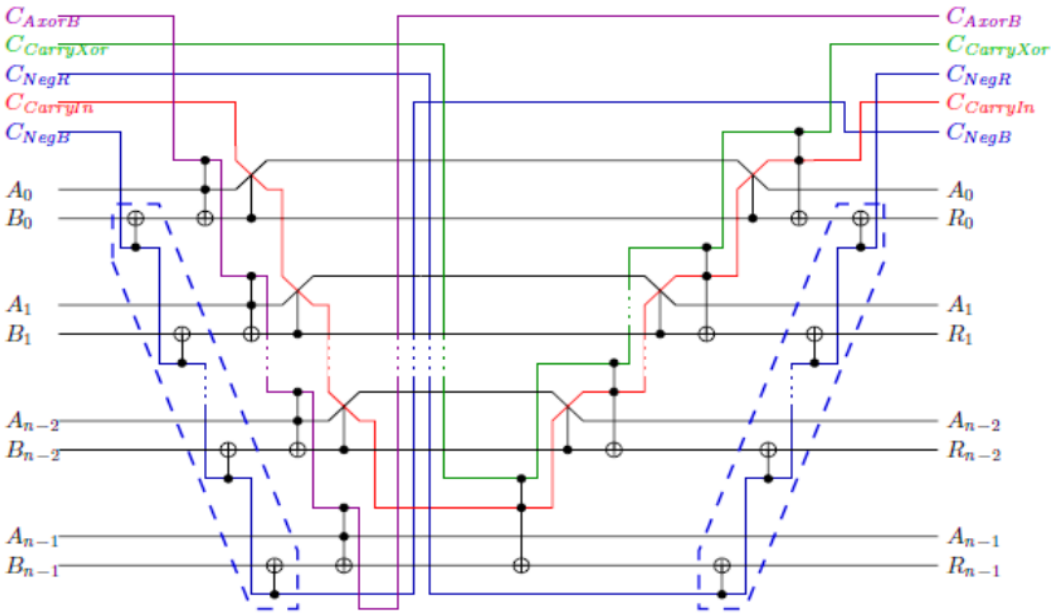

Figure 2.13 : Circuit d'UAL (modèle TGA)

- ✓ $\boldsymbol{C_{NegB}}$ **:** C'est un fil de contrôle pour la porte CNOT ajoutée à chaque qubit pour qui inverse le qubit $|B\rangle$, si l'état d'entrée de ce fil est à $|1\rangle$, Ce fil est nécessaire pour convertir l'opération d'addition en soustraction par processus d'inversion de qubit $|B\rangle$.
- ✓ $\boldsymbol{C_{NegR}}$ **:** Ce fil de contrôle pour la porte CNOT ajoutée à chaque niveau sert à inverser la somme R si l'état d'entrée de ce fil est à $|1\rangle$, Nous avons besoin de ce fil pour transformer l'opération d'addition en soustraction lorsque la somme est inversée.
- ✓ $\boldsymbol{C_{CarryXor}}$ **:** Ce fil de contrôle est pour la porte Toffoli, qui remplace la porte CNOT dans le circuit. Si l'entrée de ce fil est à l'état $|1\rangle$, il permet d'effectuer l'opération XOR entre C et $A \oplus$ B.

Si l'entrée est à l'état $|0\rangle$, il n'autorise pas l'opération XOR, donc la sortie reste inchangée $A \oplus$ B.

- ✓ $C_{A\ xor\ B}$ **:** Ce fil de contrôle est pour la porte Toffoli, qui agit comme la porte CNOT dans le circuit.

  Si l'entrée de ce fil est à l'état $|0\rangle$, il permet l'opération $A \oplus$ B. Si l'entrée est à l'état 0, il n'autorise pas cette opération, donc la sortie reste B.
- ✓ $C_{Carry\ In}$ **:** Ce fil de contrôle est pour la porte Fredkin. Nous avons besoin de ce fil pour annuler l'opération de soustraction si dans l'état $|1\rangle$.

Ce modèle peut réaliser une gamme de 10 opérations logiques et arithmétiques.

Opérations résultantes d'ALU **Modele (TGA)**	C_{NegB}	$C_{Carry\ In}$	$C_{A\ xor\ B}$	$C_{CarryXo}$	C_{NegR}
Addition A+B	0	0	1	1	0
Soustraction A−B	1	0	1	1	1
Soustraction négative A−B	1	1	1	1	0
Opération XOR A⊕B	0	0	1	0	0
Pas d'opération B	0	0	0	0	0
B+A+1	0	1	1	1	0
B−A−1	1	1	1	1	1
A−B−1	1	0	1	1	0
$\overline{A \oplus B}$	0	0	1	0	1
$\overline{B}$	0	0	0	0	1

Tableau 2.1 : Les opérations d'UAL modèle TGA

Étant donné que ce modèle utilise 2n portes CNOT, 2n portes Toffoli et 2n-2 portes Fredkin, on calcule son coût total en fonction du nombre de qubits n comme suit :

QC= 2n*5+ 2n*1+(2n-2)*5= 22n-10

Et le délai quantique : QD= 1+5+(n-1)*5+10n-5+1= 15n-3

Et le déchet quantique : QG= 0

Nous pouvons calculer le coût du circuit UAL pour quatre qubits de la manière suivante : QC=78, QD= 57 et QG=0.

2.2 Unité arithmétique et logique quantique (Modèle ZLZH) :

Le modèle (ZLZH) Utilise la même méthode que le modèle précédent où des fils colorés appelés unités de contrôle ont été ajoutés au circuit de l'additionneur (figure 2.15), ainsi que Des portes CNOT

ont été ajoutées et certaines portes CNOT ont été remplacées par des portes Toffoli. (Zhou, 2015) De nouvelles portes ont également été ajoutées en raison de l'ajout d'un qubit de contrôle à une porte Toffoli, la transformant en une porte à 4 entrées et 4 sorties, comme illustré dans (la figure 2.14)

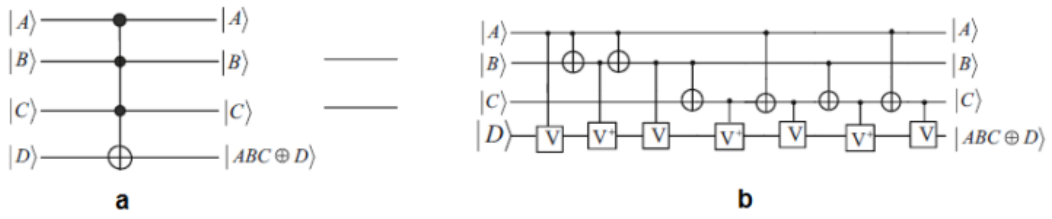

Figure 2.14 : (a) la porte Toffoli (4*4) et (b) son implémentation quantique.

Nous pouvons calculer le coût d'une porte Toffoli (4*4) comme suit : QC= 13, QD= 13.

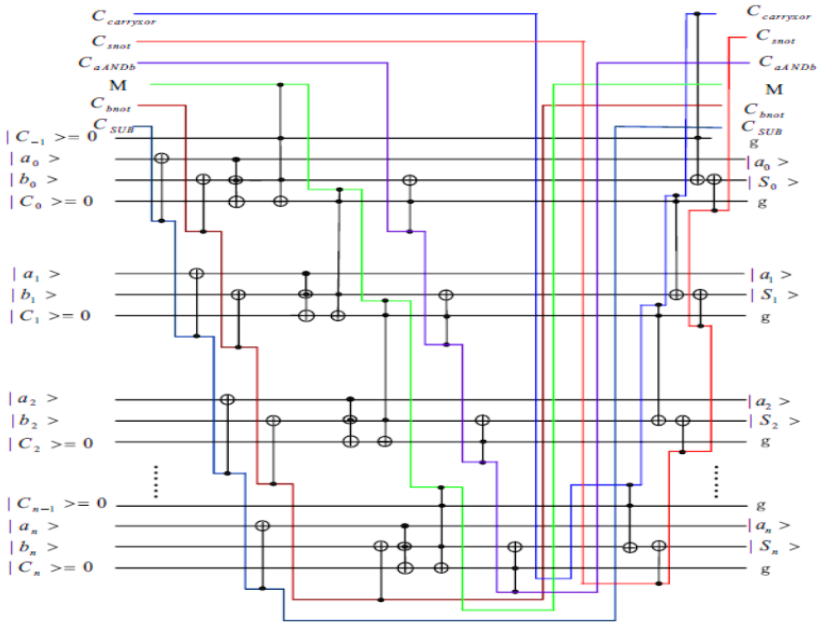

Figure 2.15 : Circuit d'UAL (modèle ZLZH).

Le circuit de l'unité arithmétique et logique possède une unité de contrôle composée de 6 fils, dont les entrées varient entre 0 et 1 pour obtenir des opérations logiques et arithmétiques. Les fils de contrôle sont disposés comme suit : $C_{carry\ xor}, C_{snot}, C_{aANDb}, C_{bnot}, C_{SUB}$, M.

Le fil de contrôle M détermine les opérations produites par le circuit ALU. Si M = 0, le circuit produit des opérations logiques. En revanche, si M = 1, le circuit produit des opérations arithmétiques.

Le modèle ZLZH est l'un des meilleurs modèles en termes du nombre d'opérations produites par le circuit, comme résumé dans le tableau 2.2. Ce circuit produit 18 opérations logiques et arithmétiques, mais son coût sera également élevé.

Opérations résultantes du côté ALU	C_{SUB}	C_{bnot}	C_{aANDb}	C_{snot}	$C_{carry\ xor}$	**M**
A plus B	0	0	0	0	1	1
$\overline{A\ plus\ B}$	1	0	1	1	1	1
A minus B	1	0	1	1	1	1
B minus A	1	1	1	1	1	1
$A + B + 1$	1	1	1	0	1	1
$A - B - 1$	0	1	1	0	1	1
$B - A - 1$	0	0	1	0	1	1
$\overline{A}$ **plus =** $\overline{B}$	0	1	1	0	1	1
$A \oplus B$	0	0	0	1	0	0
$\overline{A \oplus B}$	1	0	0	1	0	0
$A + B$	1	0	0	1	0	0
$\overline{A + B}$	0	0	0	1	0	0
$\overline{A} + B$	0	0	0	1	0	0
$A + \overline{B}$	0	1	0	1	0	0
$A \cdot B$	0	1	0	1	0	0
$\overline{A} \cdot B$	0	0	0	1	0	0
$A \cdot \overline{B}$	0	1	0	1	0	0
$\overline{A} + \overline{B}$	0	1	0	0	0	0
$\overline{A}$	1	0	0	0	0	0

Tableau 2.2 : Les opérations d'UAL modèle ZLZH.

La raison de l'augmentation du coût du circuit ALU modèle HLZH réside dans l'utilisation d'un nombre important de portes, en particulier les portes Toffoli à 4 entrées et 4 sorties, dont le coût individuel peut atteindre jusqu'à 13.

Le circuit sera construit avec n portes Toffoli (4*4), en plus 2n de portes Toffoli (3*3), 3n portes CNOT et n portes Peres par conséquent, le coût total serait le suivant :

QC = 13*n + 5*2n +1*3n+ 4*n = 30n.

Le délai quantique : QD = 1+1+4+13n+10+5n+1 = 18n+17

Le déchet quantique : QG=n+2

Alors le coût du circuit ALU pour quatre qubits est le suivant : QC = 120, QD = 89 et QG= 6

2.3 Unité arithmétique et logique quantique (Modèle HA) :

Ce modèle, contrairement aux autres modèles précédents, se base sur la conception d'un seul circuit UAL pour un qubit, puis construit le circuit pour n qubits en connectant séquentiellement ces UAL pour un qubit. Dans ce travail, les chercheurs ont proposé trois conceptions d'UAL pour un qubit. Nous choisirons la meilleure, qui est la conception numéro 2 comme illustrée dans l'image. (Haghparast, 2016)

L'unité de contrôle dans ce modèle est composée des fils S_0 à S_4, et les autres fils constituent le circuit quantique. Cette 1-qubit UAL produit 10 opérations logiques et arithmétiques en sortie, résumées dans le tableau (2.3).

Dans ce modèle utilise 6 portes CNOT, 4 Controlled-V et 2 Controlled-V.
Par conséquent, le coût total de cette porte est le suivant : QC = 12n, QD = 12n, QG = 0.

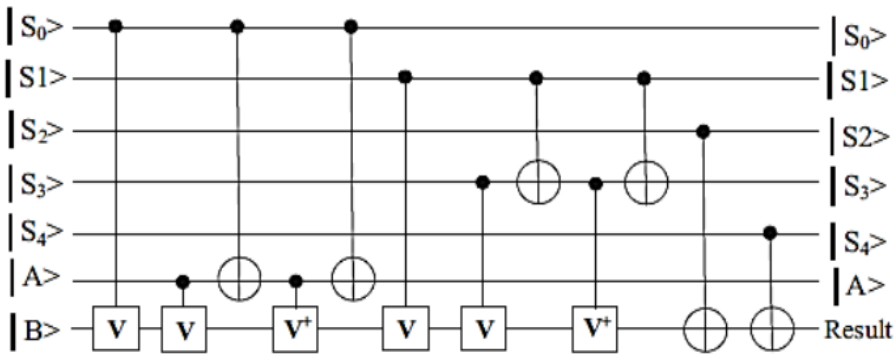

Figure 2.16 : Circuit d'UAL (modèle HA)

Opérations résultantes d'ALU (modèle HB)	S_0	S_1	S_2	S_3	S_4
Addition $A + B$	1	1	0	0	0
Soustraction $A - B$	1	1	1	0	1
Soustraction négative$B - A$	1	1	0	1	1
A⊕B	1	0	0	0	0
Pas d'opération B	0	0	0	0	0
$A + B + 1$	1	1	0	1	0
$A - B - 1$	1	1	1	1	1
$B - A - 1$	1	1	0	0	1
$A + B$	0	0	1	0	0
B	0	0	1	0	0

Tableau 2.3 : Les opérations d'UAL modèle HA.

Conclusion :

Dans ce chapitre, nous avons examiné les circuits d'unité arithmétique et logique (UAL) les plus réputés et performants, en explorant tous les concepts nécessaires à leur réalisation. Nous avons aussi calculé les paramètres de performance (QC, QD et QG) pour chaque modèle, nécessaires à utiliser dans le chapitre suivant.

Dans le chapitre suivant, nous présenterons une conception d'UAL proposé et tenterons de la simuler dans Qiskit.

Chapitre 3 :

La conception proposée d'unité arithmétique et logique quantique et sa simulation dans

Qiskit « la partie pratique »

Introduction :

Dans ce chapitre on va présenter la conception sophistiquée d'une ALU réversible à 4-qubits basée sur un additionneur quantique complet contrôlé composée par une porte HGN et une porte de Feynman avec un arrangement séquentiel d'opérations. L'ALU réversible proposée vise à implémenter l'ensemble d'opérations de base suivant, nous supposons les deux 4-qubits comme $|A> = |a_0 a_1 a_2 a_3> et\ |B> = |b_0 b_1 b_2 b_3>$:

1-l'opération (ADD): $|A + B>$

2- l'opération (SUB): $|A - B>$

3- L'opération (Négative-SUB): $|B - A>$

4-L'opération (OU exclusif): $|A \oplus B>$

5-L'opération (OR): $|A\ or\ B>$

6-L'opération (AND): $|A.B>$.

Nous verrons plus tard que l'UAL réversible finale permet plus que ces six opérations, Il est très important de minimiser les coûts de ressources pour la conception d'une UAL réversible et d'identifier un ensemble d'instructions de données d'entrée de contrôle expressives pour la programmation informatique.

1 La conception proposée d'additionneur quantique :

Il existe de nombreuses portes réversibles fonctionnant comme un additionneur complet réversible, telles que DPG et HGN. (H.R.BHAGYALAKSHMI, 2010)D'après l'expression de l'additionneur complet classique dans eq.1 et eq.2, il est facile d'obtenir ce HGN en tant qu'additionneur complet réversible est représenté sur la Figure 3.1 :

$$Sum = A \oplus B \oplus C \qquad (1)$$

$$Carry = AB \oplus BC \oplus AC = AB \oplus C(A \oplus B) \qquad (2)$$

Soit $|a_i>, |b_i>, |c_i>$ et $|S_i>$ le ième Qubit of $|A>, |B>, |Carry>$ et $|Sum>$,respectivement.

$$S_i = a_i \oplus b_i \oplus c_{i-1} \qquad (3)$$

$$c_i = a_i b_i \oplus (a_i \oplus b_i) c_{i-1} \qquad (4)$$

Un additionneur quantique de 4-qubits est mis en cascade par l'additionneur complet illustré à la Figure 3.1. Il est résumé ici sous la forme Figure 3.2.

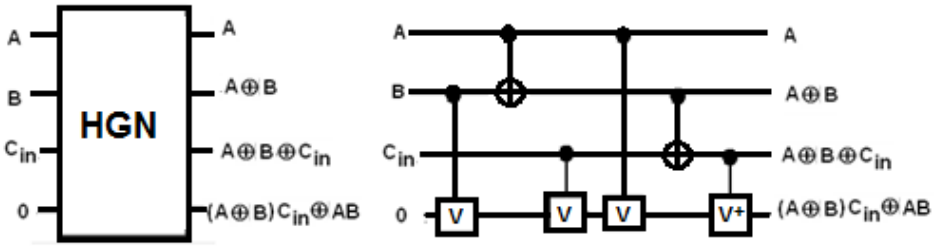

Figure 3.1 : La porte HGN et sa représentation quantique.

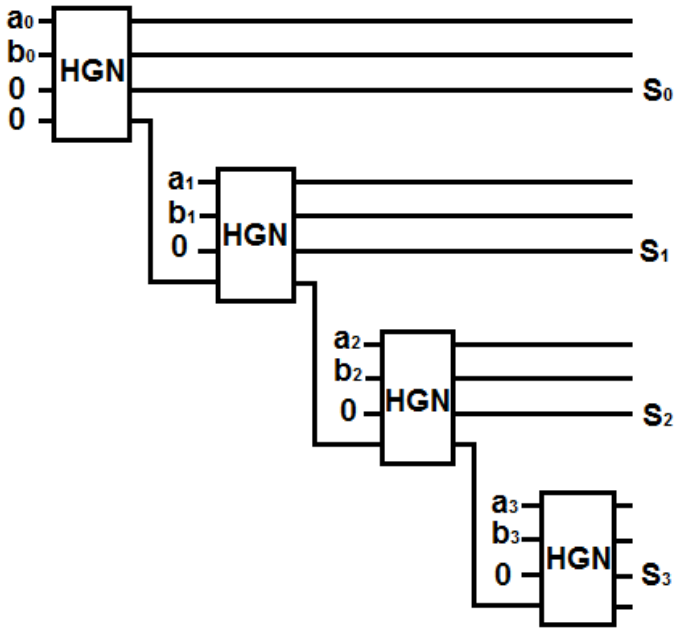

Figure 3.2 : la conception proposée d'additionneur quantique de 4-qubits.

La porte HGN consomme : QC=6 et QD=5.

Donc le circuit de l'additionneur quantique proposée composée de 4 HGN consomme :

QC = 24, QD = 20 et QG = 5.

2 La conception proposée de L'unité arithmétique réversible :

Pour réaliser des opérations arithmétiques supplémentaires comme la soustraction (SUB) et la négation de la soustraction (Négative-SUB), des lignes de contrôle doivent être intégrées au circuit de l'additionneur précédent. De plus, l'ajout de portes réversibles telles que la porte de Toffoli et la porte de Feynman est nécessaire.

Sur la Figure 3.3 nous présentons un nouveau circuit réversible et quantique capable d'effectuer plusieurs opérations arithmétiques produisant en sortie $|R>=|R_0R_1R_2R_3> et\ C_{out}$.

L'expressions logique de le Ième R_i et $C_{out\,i}$ s'écrit comme suit :

$$R_i = (a_i \oplus S_0) \oplus (b_i \oplus S_1) \oplus S_2 c_{i-1} \oplus S_3 \quad (5)$$

$$C_{out\,i} = \big((a_i \oplus S_0) \oplus (b_i \oplus S_1)\big) S_2 c_{i-1} \oplus (a_i \oplus S_0)(b_i \oplus S_1) \quad (6)$$

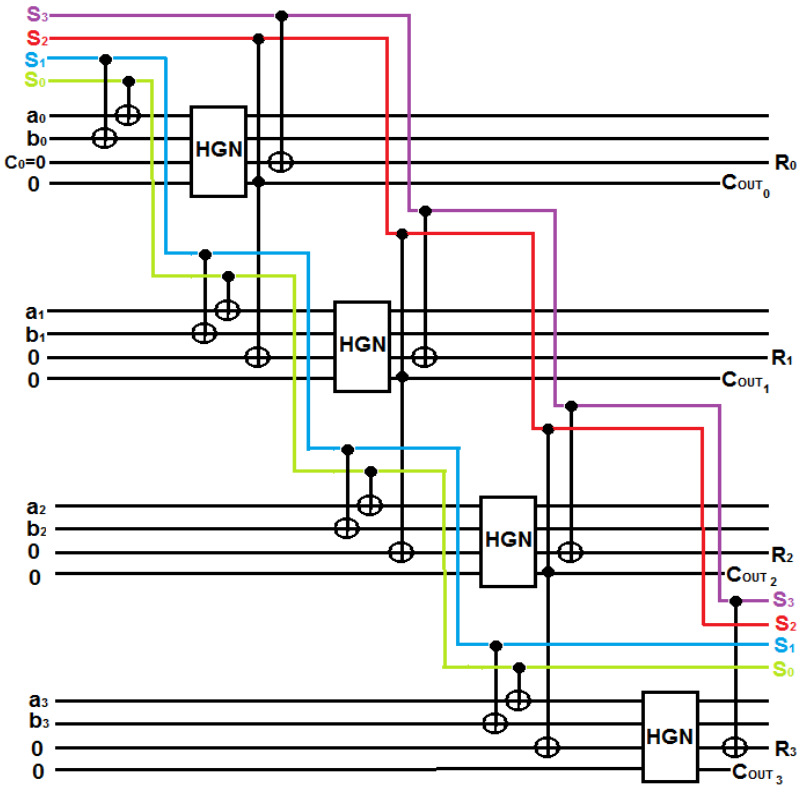

Figure 3.3 : La conception proposée de l'unité arithmétique réversible de 4-qubits.

Exemple :

- ✓ Pour effectuer l'opération d'addition "A plus B" toutes les lignes de contrôle doivent être configurées en position "0 et 1" $S_0 = S_1 = S_3 = 0\ et\ S_2 = 1$.

  Alors : Les équations 4 et 5 sont modifiées comme suit :

$$R_i = (a_i) \oplus (b_i) \oplus c_{i-1} \quad (7)$$

$$C_{out\,i} = \big((a_i) \oplus (b_i)\big) c_{i-1} \oplus (a_i)(b_i) \quad (8)$$

- ✓ Pour effectuer l'opération de soustraction "A minus B" toutes les lignes de contrôle doivent être configurées en position $S_0 = S_1 = S_3 = 1\ et\ S_1 = 0$.

  Alors : Les équations 4 et 5 sont modifiées comme suit :

$$R_i = \overline{(\bar{a_i}) \oplus (b_i) \oplus c_{i-1}} \quad (9)$$

$$C_{out\,i} = \big((\bar{a_i}) \oplus (b_i)\big) c_{i-1} \oplus (\bar{a_i})(b_i) \quad (10)$$

À partir de l'équation 5, nous pouvons résumer toutes les opérations arithmétiques effectuées par le circuit de l'unité arithmétique proposé dans le tableau Suivant :

Numéro d'opération	S_0	S_1	S_2	S_3	les opérations arithmétiques produites par l'unité arithmétique proposée
(1)	0	0	1	0	$\boldsymbol{A\ plus\ B}$
(2)	0	0	1	1	$\overline{\boldsymbol{A\ plus\ B}}$
(3)	1	0	1	1	$\boldsymbol{A\ minus\ B}$
(4)	0	1	1	1	$\boldsymbol{B\ minus\ A}$
(5)	1	1	1	1	$\boldsymbol{A\ plus\ B\ plus\ 1}$
(6)	1	0	1	0	$\boldsymbol{A\ minus\ B\ minus\ 1}$
(7)	0	1	1	0	$\boldsymbol{B\ minus\ A\ minus\ 1}$
(8)	1	1	1	0	$\boldsymbol{\bar{A}\ plus = \bar{B}}$
(9)	0	0	0	0	$\boldsymbol{A \oplus B}$
(10)	0	0	0	1	$\overline{\boldsymbol{A \oplus B}}$

Tableau 3.3: les opérations arithmétiques produites par l'UAL quantique proposée.

3 La conception proposé de l 'unité arithmétique et logique (UAL) :

En informatique classique, l'unité logique est présentée comme une unité indépendante de l'unité arithmétique. En effet, l'informatique classique est irréversible car les informations sont effacées. Étant donné que l'informatique quantique est réversible et qu'aucune information n'est effacée, cela signifie que nous pouvons utiliser l'unité arithmétique pour créer une unité arithmétique et logique combinée, et cela se fait en ajoutant d'autres portes quantiques à l'unité arithmétique.

On remarque dans le circuit de l'unité arithmétique précédent (Figure3.3) qu'une des sorties est $C_{out\ i}$ Qui peut être exploité pour produire des opérations logiques Cela se fait en ajoutant la porte Feynman (Figure 3.4) de sorte que l'équation de cette sortie devienne la suivante :

$$C_{out\ i} = \big((a_i \oplus S_0) \oplus (b_i \oplus S_1)\big)S_2 c_{i-1} \oplus (a_i \oplus S_0)(b_i \oplus S_1) \oplus S_3 \qquad (11)$$

Exemple :

- ✓ Pour effectuer l'opération d'addition "A AND B" toutes les lignes de contrôle doivent être configurées en position 0 $S_0 = S_1 = S_2 = S_3 = 0$.

  Alors : L'équations 5 sont modifiée comme suit :

$$C_{out\ i} = (a_i).(b_i) = (a_i)AND(b_i) \qquad (12)$$

✓ Pour effectuer l'opération d'addition "A OR B" toutes les lignes de contrôle doivent être configurées en position 0 et 1 $S_2 = 0$ et $S_0 = S_1 = S_3 = 1$.

Alors : l'équation précédente devient la suivante:

$$C_{out\ i} = \overline{(\overline{a_i}).(\overline{b_i})} = \overline{(\overline{a_i})AND(\overline{b_i})} = (a_i)OR\ (b_i) \qquad (13)$$

Le reste des opérations logique est résumé dans le tableau Suivant :

Les opérations arithmétiques et logiques produites par l'ALU proposée	**S_0**	**S_1**	**S_2**	**S_3**	**S_4**	**Numéro d'opération**
A plus B	0	0	1	0	0	**(1)**
$\overline{\text{A plus B}}$	0	0	1	1	0	**(2)**
A minus B	1	0	1	1	0	**(3)**
B minus A	0	1	1	1	0	**(4)**
A plus B plus 1	1	1	1	1	0	**(5)**
A minus B minus 1	1	0	1	0	0	**(6)**
B minus A minus 1	0	1	1	0	0	**(7)**
$\overline{\text{A}}$ plus = $\overline{\text{B}}$	1	1	1	0	0	**(8)**
A $\oplus$ B	0	0	0	0	0	**(9)**
$\overline{\text{A} \oplus \text{B}}$	0	0	0	1	0	**(10)**
A (AND) B	0	0	0	0	1	**(11)**
A (AND) $\overline{\text{B}}$	0	1	0	0	1	**(12)**
$\overline{\text{A}}$ (AND) B	1	0	0	0	1	**(13)**
$\overline{\text{A}}$ (AND) $\overline{\text{B}}$	1	1	0	0	1	**(14)**
$\overline{\text{A}}$ (OR) $\overline{\text{B}}$	0	0	0	1	1	**(15)**
A (OR) $\overline{\text{B}}$	1	0	0	1	1	**(16)**
$\overline{\text{A}}$ (OR) B	0	1	0	1	1	**(17)**
A (OR) B	1	1	0	1	1	**(18)**

Tableau 3.4: les opérations arithmétiques et logiques produites par l'UAL quantique proposée.

Enfin, nous pouvons ajouter la porte de Fredkin, qui est considérée comme une multiplexer quantique qui combine les deux sorties $R_i\ et\ C_{out\ i}$ sur une seule ligne (Figure 3.4), Nous l'avons également appelé R, abréviation du mot « résultat », et c'est la sortie finale du circuit de l'unité arithmétique et logique quantique, dont l'équation s'écrit comme suit :

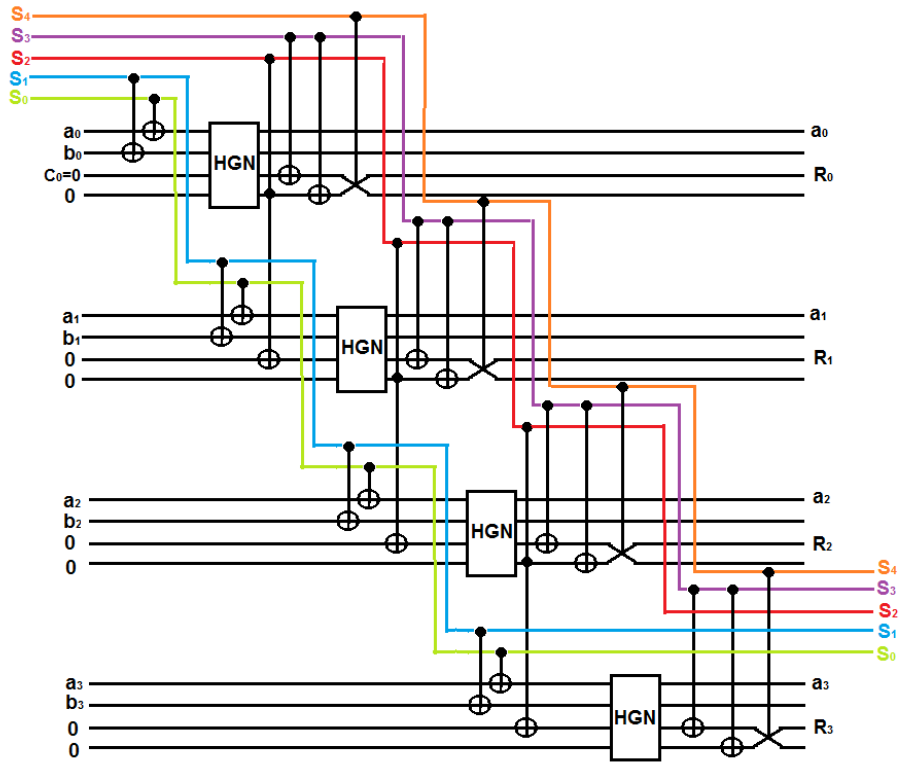

Figure 3.4 : La conception proposé du circuit arithmétique et logique réversible.

$$R_i = [(a_i \oplus S_0) \oplus (b_i \oplus S_1) \oplus S_2 c_{i-1} \oplus S_3]\overline{S_4} \oplus \left[\left((a_i \oplus S_0) \oplus (b_i \oplus S_1)\right)S_2 c_{i-1} \oplus (a_i \oplus S_0)(b_i \oplus S_1) \oplus S_3\right]S_4 \qquad (14)$$

Puisque la porte de Fredkin est contrôlée par la ligne S4, cela signifie que si l'on veut produire des opérations logiques, on met S_4 en position « 1 », et si l'on veut produire des opérations arithmétiques, on met $S_4 = 0$.

La figure 3.4 montre que le circuit proposé de 4 qubits utilise 5 lignes de contrôle et se compose de : 4(4CNOT + HGN + 4Fredkin) + 3Toffoli

Alors :

Le cout quantique : QC = ((4*1) +6+5) *4+(5*3)) = 75.

Le delay quantique : QD = 50

Le déchet quantique : QG = 8

4 Implémentation et Simulation quantique d'UAL proposée dans le « QISKIT » :

Qiskit est un kit de développement logiciel (SDK) open source permettant de travailler avec des ordinateurs quantiques au niveau des circuits et des algorithmes. Il fournit des outils pour créer et

manipuler des programmes quantiques et les exécuter sur des prototypes de dispositifs quantiques sur IBM Quantum Platform ou sur des simulateurs sur un ordinateur local.

Dans cette partie de la recherche, on va étudier l'efficacité du circuit d'unités arithmétiques et logiques proposé en termes de réussite de toutes les opérations, sachant qu'on va utiliser pour cela le programme Qiskit pour réaliser ce circuit.

```
!pip install qiskit

!pip install Aer

!pip install qiskit_Aer
```

```
from qiskit import QuantumCircuit
from qiskit.circuit.library import  SXGate , SXdgGate
from qiskit.quantum_info import Operator
from qiskit import transpile, assemble
from qiskit.visualization import plot_histogram, plot_state_qsphere
from qiskit_aer import  Aer

import matplotlib.pyplot as plt
import numpy as np
```

```
[ ]  qc= QuantumCircuit(21, 4)

     csx_gate = SXGate(label="v").control()

     csxdg_gate = SXdgGate(label="v+").control()
     ##########S4
     #qc.x(0)
     ##########S3
     #qc.x(1)
     ##########S2
     #qc.x(2)
     ##########S1
     #qc.x(3)
     ##########S0
     #qc.x(4)
     ##########a0
     #qc.x(5)
     ##########b0
     #qc.x(6)
     ##########a1
     #qc.x(9)
     ##########b1
     #qc.x(10)
     ##########a2
     #qc.x(13)
```

Figure 3.5: Partie du code utilisé pour simuler de l'UAL proposé.

Nous aurions pu implémenter ce circuit avec la bibliothèque NCT dans le qiskit, mais nous avons préféré l'implémenter avec la bibliothèque NCV pour obtenir les améliorations offertes par qiskit comme le délai.

La figure 3.6 montre le processus de dessin du circuit à l'aide du qiskit, où il a amélioré le circuit en réduisant le délai à QD=31 et QC=71.

Pour tester ce circuit, vous devez d'abord choisir l'opération à effectuer, déterminer la position de l'unité de contrôle (Tableau 3), puis sélectionner deux nombres binaires, A et B, sur lesquels l'opération sera réalisée.

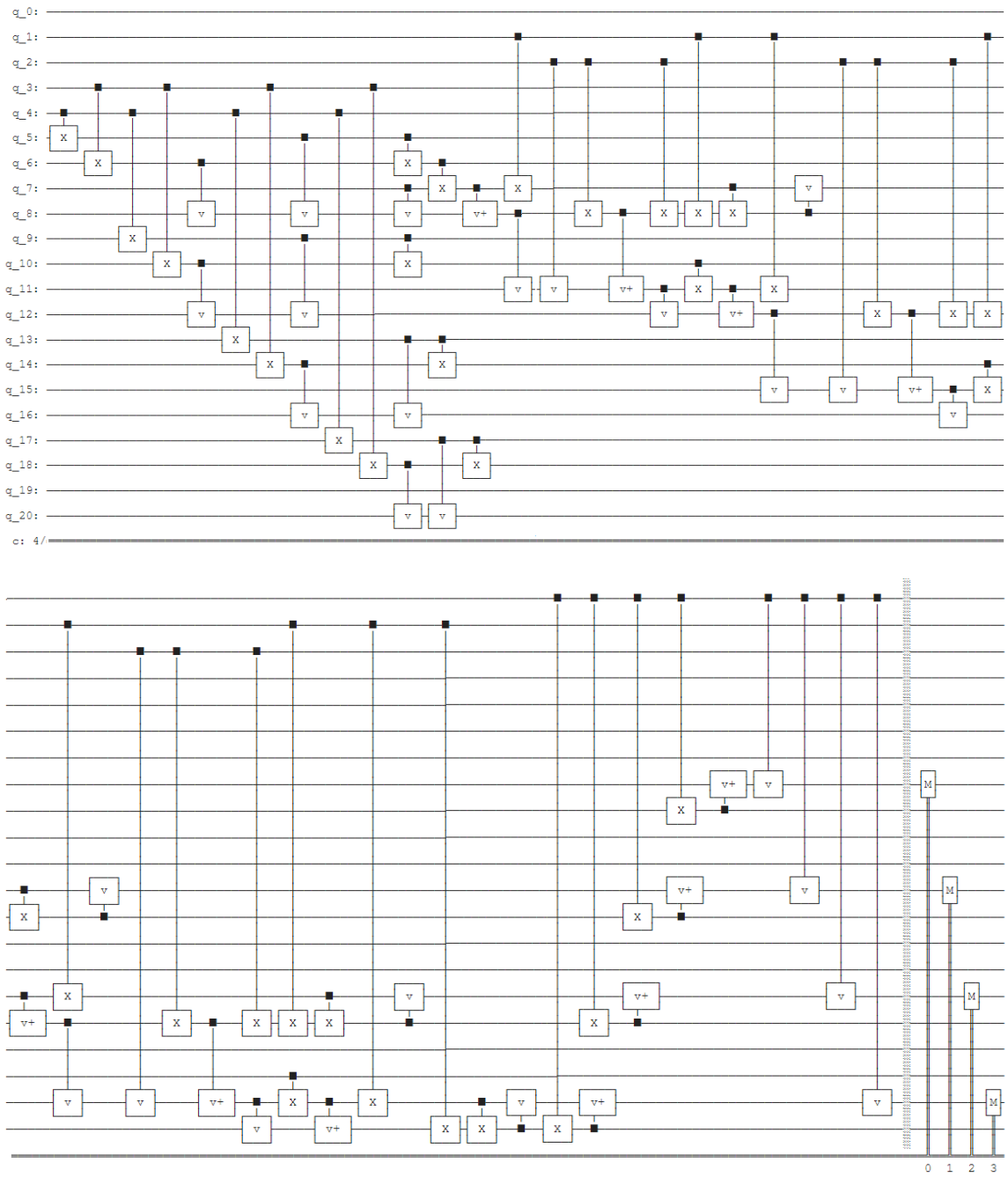

Figure 3.6 : l'affichage du circuit proposé on qiskit.

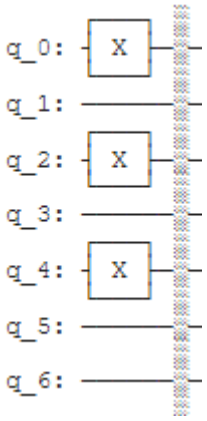

Figure 3.7 : La méthode pour convertir les qubits de 0 à 1.

Dans la Figure 3.7 tous les qubits sont présentés dans l'état 0. Ainsi, pour les transformer en état 1, il est nécessaire d'appliquer une porte NOT « X » à chaque qubit individuel.

Exemple :

Si nous prenons deux nombres A=5 et B=2, cela peut être exprimé en binaire comme suit :

$|A>=|a_0=1,a_1=0,a_2=1,a_3=0>et\ |B>=|b_0=0,b_1=1,b_2=0,b_3=0>$

Pour effectuer les opérations 1, 2, 11 et 18 (voir tableau 2), nous devons ajuster l'unité de contrôle en mode :

$$op1\ \rightarrow |S_0=0,S_1=0,S_2=1,S_3=0,S_4=0\ >$$

$$op3\ \rightarrow |S_0=1,S_1=0,S_2=1,S_3=1,S_4=0\ >$$

$$op11\ \rightarrow |S_0=0,S_1=0,S_2=0,S_3=0,S_4=1\ >$$

$$op18\ \rightarrow |S_0=1,S_1=1,S_2=0,S_3=1,S_4=1\ >$$

Avant d'implémenter le circuit dans qiskit, nous connaissons avec certitude les résultats des opérations suivantes :

Pour l'addition (opération 1) des deux nombres A et B : $\mathrm{R(op1)=A+B=5+2=7=|0111>}$

Pour la soustraction (opération 3) : $\mathrm{R(op3)=A-B=5-2=3=|0011>}$

Pour l'opération logique "ET" entre A et B : $\mathrm{R(op11)=A\ AND\ B=|0000>}$

Pour l'opération logique "OU" entre A et B : $\mathrm{R(op18)=A\ OR\ B=|0111>}$

Les résultats des quatre opérations exécutées dans qiskit sont présentés dans la figure 3.8 confirmant leur correspondance avec les résultats calculés manuellement.

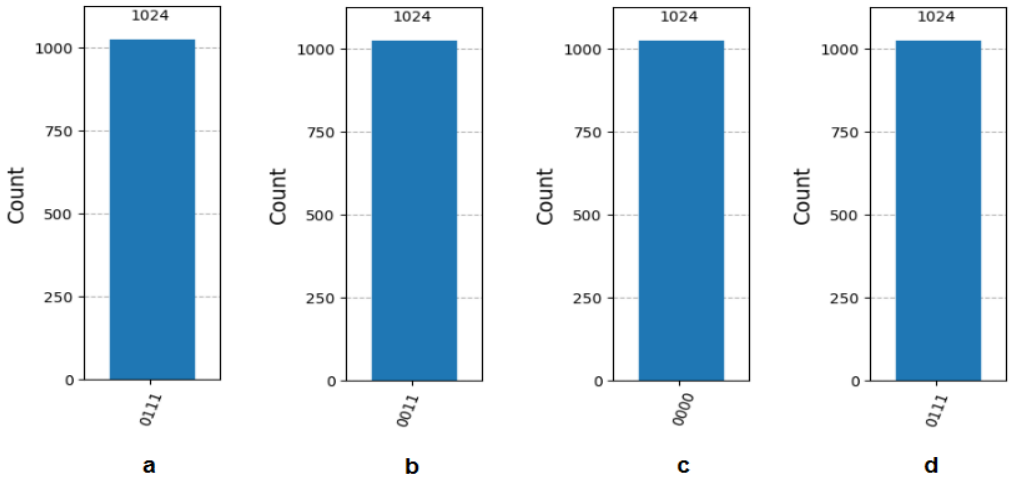

Figure 3. 6 : Les résultats de l'exécution dans le qiskit

a operation 1 (addition) **b.** operation 3 (soustraction) **c.** operation 11 (AND) d. operation 18 (OR).

4 Comparaison des résultats :

Le tableau suivant compare les UALs quantiques proposées avec d'autres circuits conçus et mentionnés dans le chapitre 02.

Paramètres d'évaluations / Les conceptions D'UAL	Nombre d'Operations	QC	QD	QG
Modèle TGA	10	78	57	0
Modèle ZLZH	18	120	65	6
Modèle HA	10	48	48	0
Conception proposé	18	69	31	8

Tableau 3.3 : Comparer les résultats de l'UAL proposée avec les résultats précédents.

L'unité arithmétique et logique proposé présentent trois avantages majeurs par rapport aux travaux précédents :

- ✓ Les circuits présentés ont obtenu le coût quantique le plus réduit (QC), ce qui est un critère essentiel dans l'évaluation des circuits quantiques.
- ✓ Le nombre de fonctions produites (18 opérations).
- ✓ Le coût quantique (QC) est le Delay quantique (QD) Avec un QD = 31, cette amélioration représente une avancée significative pour le circuit de l'UAL.

Mais pour confirmer que la conception proposé est plus rapide que tous les circuits précédents, même ceux qui exécutent un nombre limité d'opérations. Nous avons proposé un nouveau standard pour évaluer les meilleures conceptions. Ce standard, nommé coefficient d'amélioration(If), unifie quatre critères (nombre d'opérations, QD, QC et QG) en un seul coefficient. Plus ce coefficient (If) est élevé pour une certaine conception de circuit, meilleure est cette conception par rapport aux autres. Cette formule est détaillée dans l'équation suivant :

$$If = \frac{nombres\ des\ opérations}{QC + QD + QG}$$

Pour déterminer comment le coefficient d'amélioration identifie quelles conceptions sont meilleures, nous l'expliquons dans les points suivants :

- ✓ Plus le nombre d'opérations produites dans une conception est élevé, meilleure est la performance du circuit. C'est pourquoi nous avons placé le nombre d'opérations dans le numérateur. Ainsi, à mesure que ce nombre augmente, le coefficient d'amélioration augmente également.
- ✓ Plus les valeurs des critères QD, QC et QG sont petites, meilleure est la conception. C'est pourquoi nous les avons placés dans le dénominateur. Par conséquent, à mesure que le nombre de QD, QC et QG diminue, le coefficient d'amélioration augmente.

Les résultats de la comparaison entre les conceptions précédentes et la conception proposée sont présentés dans le tableau suivant :

Les conceptions D'UAL / L'amélioration	TGA	ZLZH	HA	Conception proposé
Coefficient d'amélioration	0.07	0.09	0.10	0.16

Tableau 3.4 : comparaison de la conception proposée avec les conceptions précédentes en termes de coefficient d'amélioration.

Alors comme indiqué dans le tableau, le circuit proposé est plus rapide que tous les circuits précédents, même ceux qui exécutent un nombre limité d'opérations.

Conclusion :

Dans ce chapitre, nous avons conçu une UAL réversible à 4 qubits basée sur un additionneur quantique complet contrôlé, utilisant une combinaison de portes logiques quantiques. Ainsi que nous avons également simulé le circuit proposé.

Conclusion générale :

Dans cette étude, nous avons abordé un domaine crucial de l'ingénierie informatique quantique, en mettant en lumière le rôle central du processeur quantique. L'objectif principal était d'améliorer les performances des circuits de l'unité arithmétique et logique quantique réversible. Étant donné la nouveauté de ce domaine, les conceptions présentées ici sont semblables, dans une certaine mesure, aux conceptions traditionnelles d'ordinateurs, adaptant les principes et les circuits clés utilisées en informatique classique.

En tant que domaine basé sur les principes de la mécanique quantique, l'ordinateur quantique repose sur des concepts que nous avons explorés et résumés pour la création de circuits quantiques et de conceptions logiques. Il existe plusieurs modèles pour convertir les circuits classiques en circuits quantiques réversibles, tels que les systèmes NCT et NCV. Dans ce travaille, nous avons utilisé le système NCV pour transformer toutes les portes classiques réversibles en versions quantiques réversibles. Ces portes sont assemblées en circuits qui accomplissent diverses tâches en fonction des portes utilisées dans leur construction.

En outre, nous avons amélioré les conceptions pour développer un circuit d'unité arithmétique et logique quantique, en utilisant des critères d'amélioration tels que le nombre d'opérations réalisées, QD, QC et QG. Le circuit proposé dans cette thèse exécute 18 opérations, facilitant ainsi le fonctionnement du processeur quantique.

En complément, nous avons également procédé à la simulation de notre circuit dans Qiskit, suivie de tests rigoureux visant à confirmer son efficacité. Ce processus nous a permis de vérifier la performance et la fiabilité du circuit dans des conditions de simulation avant d'envisager une éventuelle implémentation physique.

Enfin, cette conception offre des coûts réduits en termes de QD, QC et QG par rapport aux précédentes, en faisant du circuit proposé une solution hautement efficace. Il est donc prévu que cette conception joue un rôle clé dans l'avenir de la réalisation des ordinateurs quantiques.

Bibliography

A. V. Navaneeth, M. R. (2021). A Study and Analysis of Applications of Classical Computing and Quantum Computing: A Survey. p. 12.

ANDALOUSSI, I. (2022, December). Optimisation des critères de performancesdes circuits réversibles. p. 127. Récupéré sur https://www.researchgate.net/publication/366085988

Blais, A. (decembre 2002). Algorithmes et architectures pour ordinateurs quantiques supraconducteurs.

Candidato. (2020). Arithmetic circuits for quantum computing: a software library.

Francisco ORTS, G. O. ((2019)). A Faster Half Subtractor Circuit Using Reversible. p. 13. Retrieved from https://doi.org/10.22364/bjmc.2019.7.1.08

H.R.BHAGYALAKSHMI, M. (2010). AN IMPROVED DESIGN OF A MULTIPLIER USING REVERSIBLE LOGIC GATES. *International Journal of Engineering Science and Technology*, 9.

Haghparast, M. a. (2016). Optimization approaches for.

Handique, M. (2017). an extended aproach for mapping reversible circuits to quantumcircuits using NCV library.

HIMANSHU THAPLIYA, N. R. ((2010), December). Design of Reversible Sequential Circuits. p. 31. Retrieved from http://doi.acm.org/10.1145/1877745.

Khaled El-Wazan, M. O. (2021). Efficient Designs of Quantum Adder/Subtractor Using Universal Reversible Gate on IBM Q. p. 15.

KHALFI, M. F. (2016, Mars). Cours Structure Machine. p. 147. Récupéré sur https://www.researchgate.net/publication/336957032

N, A. R. (2021). *UN P E U D E M AT H É M AT I Q U E S P O U R L'I N F O R M AT I Q U E Q U A N T I Q U E.*

Poornashree S J, P. K. (2021). Design and Implementation of Quantum half Adder and Full adder Using IBM Quantum Experience. *Advanced Trends in Computer Science and Engineering*, 4.

quantique, I. d. (2022). Physique quantique 101. Récupéré sur uwaterloo.ca/iqc

Reda Adjoudj, M. M. (2009, January). Architecture générale d'un ordinateur. p. 82. Récupéré sur https://www.researchgate.net/publication/289649079

S´everine Fratani, P. N. (23 septembre 2014). Cours d'Architecture des ordinateurs.

Sujata S. Chiwande, P. .. (2011). Introduction to Reversible Logic Gates & its Application. *2nd National Conference on Information and Communication Technology (NCICT)* (p. 5). International Journal of Computer Applications® (IJCA).

Surekha, M. (2017). Efficient Approaches for Designing Quantum Costs of Various Reversible Gates. 22. Retrieved from https://dx.doi.org/10.37622/IJES/9.1.2017.57-78

Thomas Cluzel, C. M. (s.d.). *Découverte de l'informatique quantique, état de l'art.* 1 rue de la Chébarde, FRANCE.

Thomsen, M. K. (2010). Reversible arithmetic logic unit for quantum arithmetic.

Vedral, V. A. (1996). *Quantum networks for.*

Zhou, R. e. (2015). *Novel design for reversible arithmetic logic.*