République Algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire

de fin d'études pour l'obtention du diplôme
de Master en Informatique
Option : Intelligence Artificielle

Thème

Correction d'erreur quantique:

Approche basée sur les codes stabilisateurs

Encadré par :

- ✓ M^r .Khalfaoui Khaled

Réalisé par :

- ✓ M^{elle} Boutadjine Amel
- ✓ M^{elle} Cherih Wafa

Promotion 2019.

République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université de Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire

de fin d'études pour l'obtention du diplôme
de Master en Informatique
Option : Intelligence Artificielle

Thème

Correction d'erreur quantique:

Approche basée sur les codes stabilisateurs

Encadré par :

- ✓ *M^r* .Khalfaoui Khaled

Réalisé par :

- ✓ M^elle Boutadjine Amel
- ✓ M^elle Cherih Wafa

Promotion 2019.

Table des matières

CHAPITRE II : ALGORITHMES QUANTIQUES

CHAPITRE III : CODES STABILISATEURS

CHAPITRE IV : IMPLEMENTATION ET APPLICATION

Remerciement

Avant tout, nous remercions ALLAH le tout puissant qui nous aide et nous donne la patience et le courage durant ces années d'étude, et qui sans lui ce mémoire n'aurait jamais vu le jour.

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire .Ces remerciements vont tout d'abord au Mr. Khalfaoui, notre encadrant de mémoire de fin d'étude, pour ses précieux conseils, son orientation ficelée et ainsi pour son temps tout au long de la réalisation de ce mémoire.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Nos remerciements s'étendent également au tous les membres de jury, pour l'honneur qu'ils nous ont fait en acceptant d'examiner ce travail.

Nous remercions Mr.Boudjdaa Tahar et Mr.Lakhlef Brahim pour leur aide précieuse et leurs efforts, on n'oublie pas nos chers parents pour leur contribution, leur soutien et leur patience et qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Nous adressons nos plus sincères remerciements à tous nos proches, amis et collègues, qui nous ont aidé et encouragées au cours de la réalisation de ce mémoire.

Enfin, nous remercions tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Merci à tous et à toutes.

Dédicace

A la mémoire de ma grand-mère.

A mes très chers parents, source de vie, d'amour et d'affection.

A mes chers frères Toufik, Seyf Eddine, Zakaria et Saber.

A mes belles sœurs Rafika, Nawel et Widad.

A toute ma famille, source d'espoir et de motivation.

A tous mes amis, tout particulièrement Soumia et Asma.

A mes collègues étudiants en informatique à l'université de Jijel, tout particulièrement en IA.

A Wafa, amie avant d'être binôme.

A vous cher lecteur,

Je dédie ce mémoire.

~ Amel ~

Dédicace

Je dédié ce modeste travail à mes chers parents et ma deuxième maman Rachida qui ont été toujours près de moi pour me soutenir me suivre et m'encourager et ont sacrifié leur vie pour faire en sorte que tout moyen soit valable pour mon éducation.

Puisse ce travail leur porter bonheur.

A ma sœur, mes frères et toute ma famille.

A tous mes amis, et mes collègues.

~ Wafa ~

Liste des tableaux :

CHAPITRE I : CALCUL QUANTIQUE

CHAPITRE II : ALGORITHMES QUANTIQUES

CHAPITRE III : CODES STABILISATEURS

Liste des figures :

CHAPITRE I : Calcul quantique

CHAPITRE II : Algorithmes quantiques

CHAPITRE III : Codes stabilisateurs

CHAPITRE IV : Implémentation et application

La physique quantique a mis en évidence des phénomènes dans le comportement des particules élémentaires, qui sont désormais considérés sous l'angle de leur exploitation pour représenter, traiter et communiquer l'information [1]. La rencontre entre cette discipline et les sciences de l'information débute dans les années 80. Les spécialistes de ce domaine ont confirmés qu'au monde microscopique, les lois de la physique de notre propre échelle ne peuvent plus s'appliquer mais celles de la physique quantique qui s'imposeront. Ils ont proposé d'intégrer la physique quantique dans la théorie de l'information et l'informatique et de l'utiliser comme support matériel du calcul. Cette intégration permettrait de tirer profit des phénomènes étranges et surprenants de la physique quantique telle que les superpositions d'états, l'intrication et l'interférence. Ces mécanismes semblent défier la logique et le bon sens, mais offrent des opportunités de calcul infiniment plus rapide que celui d'un calculateur classique. L'idée de base consiste à réaliser un ordinateur quantique analogue à l'ordinateur classique :

- L'unité d'information classique ''bit '' remplacé par le '' bit quantique''
- Les circuits et les portes logiques par des portes quantiques.
- L'aspect du calcul basé sur les fonctions booléennes est remplacé par un calcul spécifique basé sur les opérations algébriques linéaires.

Dans ce mode de calcul, les erreurs sont principalement dues à l'interaction du système quantique avec son environnement. Plutôt que de tenter d'affronter de face cet obstacle inévitable, il est possible d'utiliser des techniques de correction. Généralement, les chercheurs ont proposé des solutions basées sur la redondance. C'est une adaptation des correcteurs classiques en tirant profit de l'intrication afin de délocaliser sur plusieurs systèmes physiques l'information encodée.

L'objectif de ce travail est d'étudier la correction d'erreur quantique basée sur les codes stabilisateurs. D'une façon plus générale, le processus de correction d'erreur est basé sur la mesure d'un ensemble de syndrome utilisé pour discriminer l'erreur réelle de toutes les erreurs possibles. Ce manuscrit est constitué de quatre chapitres :

- Chapitre 1 : C'est une introduction au calcul quantique avec un rappel sur les notions mathématiques nécessaires. Il présente les portes et circuits quantiques en donnant les matrices de transformation correspondantes.

- Chapitre 2 : Ce chapitre est dédié aux algorithmes quantiques. Plus précisément, nous détaillons la téléportation quantique, la factorisation des entiers ainsi que la correction d'erreurs basée sur la redondance.
- Chapitre 3 : Dans ce chapitre, nous introduisons la correction d'erreurs basée sur les codes stabilisateurs. Un intérêt particulier sera porté sur l'étape d'encodage et le calcul des syndromes.
- Chapitre 4 : Dans ce chapitre, nous présentons une implémentation de cette technique. Afin de valider notre travail, des exemples d'applications seront présentés.

Enfin, nous terminons par une conclusion générale qui résume l'apport de notre travail et présente quelques perspectives.

Chapitre I : Calcul quantique

1.Introduction
2.Notion de base
3.Postulats
4.Portes quantique
5.Circuit quantique
6.Conclusion

I.1. Introduction :

La physique quantique est une théorie qui décrit le monde microscopique des atomes et des particules. Cette discipline a été développée au début du XXe siècle. Des physiciens et des mathématiciens ont montré que les phénomènes quantiques, tels qu'ils sont formulés par la mécanique quantique, peuvent être exploités pour représenter, traiter et communiquer l'information [1].

En calcul classique, l'information élémentaire manipulée est le bit. Il ne peut prendre que l'un des deux états : haut et bas. Cette formulation est suffisante pour la résolution de la plupart des problèmes. Mais dans quelques domaines particuliers tels que l'optimisation combinatoire, elle est largement limitée. Les solutions proposées nécessitent des temps de calcul considérables. Pour remédier à ce problème, le calcul quantique constitue un très bon candidat. La superposition des états quantiques ainsi que les mécanismes offerts par ce calcul permettent un parallélisme parfait [2].

Ce chapitre est dédié aux concepts de base de l'informatique quantique. Comme nous sommes confrontés à un domaine complètement nouveau, nous commençons d'abord par une courte présentation des notions physiques et mathématiques nécessaires. Ensuite, nous introduisons les différentes opérations quantiques. Enfin, nous donnons un aperçu sur les circuits quantiques.

I.2. Notions de base :

I.2.1. Bit quantique (Qbit) :

L'élément de base de l'informatique quantique est le bit quantique dit Qbit. Il représente un système basé sur deux états fondamentaux 0 et 1. A la différence du bit classique qui peut être seulement 0 ou 1, le bit quantique peut être dans plusieurs états en même temps, ou plus précisément dans une superposition de ces deux états fondamentaux.

$$\alpha|0\rangle + \beta|1\rangle$$

Tels que α et β sont deux nombres complexes appelés amplitudes et vérifiant la contrainte :

$$|\alpha|^2 + |\beta|^2 = 1$$

Il est représenté par le vecteur : $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$

L'observation de cet état donne :

- la valeur 0 avec une probabilité $|\alpha|^2$,
- la valeur 1 avec une probabilité $|\beta|^2$.

I.2.2. Etat quantique :

Un état quantique est un registre à n Qbits. Il s'agit d'une superposition donnée par :

$$|X\rangle = \sum_{x=0}^{2^n-1} c_x |X\rangle \tag{1.1}$$

Les amplitudes c_x gouvernant les probabilités des différents états doivent satisfaire la propriété suivante :

$$\sum_{x=0}^{2^n-1} |c_x|^2 = 1 \tag{1.2}$$

Dans un système classique, un état de n bit représente une seule configuration parmi les 2^n possibles. Par contre, un état quantique à n Qbit représente une superposition de toutes les 2^n configurations en même temps [2].

I.2.3. Espace de Hilbert :

L'espace de Hilbert H est un espace vectoriel complexe muni d'un produit scalaire. C'est un cadre mathématique approprié pour décrire les concepts, principes, processus et les lois de la mécanique quantique. Les états purs des systèmes quantiques sont considérés comme des vecteurs. Pour faire des calculs, les spécialistes utilisent plusieurs notations. La plus efficace est celle proposée par Dirac [3, 4].

I.2.4. Notation de Dirac :

En mécanique quantique, la notation de Dirac est utilisée afin de faciliter l'écriture des équations. Elle est encore appelée notation ''Bra-Ket'' tel que :

- Bra : un vecteur ligne.
- Ket : un vecteur colonne.

Vecteur de Ket :

Dans la notation de Dirac, un état ψ est décrit par une matrice colonne appelée Ket notée $|\psi>$ tel que :

$$|\psi\rangle = \sum_{i=1}^{n} a_i \, |i\rangle = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \quad (1.3)$$

Avec $a_i \in \mathbb{C}$.

Vecteur Bra :

La matrice ligne obtenue par la conjugaison complexe des éléments de $|\psi\rangle$ est appelée Bra. Elle est notée $\langle\psi|$ et calculée comme suit :

$$\langle\psi| = \sum_{i=1}^{n} a_i^* \langle i| = (a_1^*, a_2^*, \dots a_n^*) \quad (1.4)$$

I.2.5. Produit scalaire Bra-ket :

Etant donné un vecteur Bra $\langle\varphi|$ et un vecteur Ket $|\psi\rangle$:

$$\langle\varphi| = \sum_{i=1}^{n} b_i^* \langle i| \quad (1.5)$$

$$|\psi\rangle = \sum_{i=1}^{n} a_i \, |i\rangle \quad (1.6)$$

Le produit de ces deux vecteurs dans cet ordre noté $\langle\varphi|\psi\rangle$ est appelé produit scalaire Bra-ket. C'est un nombre complexe donné par l'équation suivante :

$$\langle\varphi|\psi\rangle = \sum_{i=1}^{n} a_i b_i^* \langle i|i\rangle \quad (1.7)$$

I.2.6. Produit tensoriel :

En mécanique quantique, le produit tensoriel est un opérateur très intéressant. Il est utilisé pour combiner deux espaces de vecteurs afin d'obtenir un nouveau espace plus large. Etant donné deux vecteurs Ket $|\psi\rangle$ et $|\phi\rangle$ tel que :

$$|\psi\rangle=\sum_{i=1}^{n} a_i\,|i\rangle \tag{1.8}$$

$$|\phi\rangle=\sum_{i=1}^{m} b_i\,|i\rangle \tag{1.9}$$

Le produit tensoriel de ces deux vecteurs dans cet ordre noté $|\psi\rangle\otimes|\phi\rangle$ est calculé comme suit :

$$|\psi\rangle\otimes|\phi\rangle=\begin{pmatrix}a_1\\a_2\\\vdots\\a_n\end{pmatrix}\otimes\begin{pmatrix}b_1\\b_2\\\vdots\\b_m\end{pmatrix}=\begin{pmatrix}a_1\begin{pmatrix}b_1\\b_2\\\vdots\\b_m\end{pmatrix}\\a_2\begin{pmatrix}b_1\\b_2\\\vdots\\b_m\end{pmatrix}\\\vdots\\a_n\begin{pmatrix}b_1\\b_2\\\vdots\\b_m\end{pmatrix}\end{pmatrix}=\begin{pmatrix}a_1.b_1\\a_1.b_2\\\vdots\\a_1.b_m\\a_2.b_1\\\vdots\\a_{n-1}.b_m\\a_n.b_1\\\vdots\\a_nb_m\end{pmatrix} \tag{1.10}$$

I.3. postulats:

I.3.1. Superposition :

A tout moment, l'état d'un registre quantique à n Qbits est un vecteur dans un espace vectoriel complexe de dimension 2^n, c'est-à-dire un vecteur avec au plus 2^n composantes complexes. La base de cet espace vectoriel comprend les 2^n vecteurs $|i\rangle$, pour i dans $\{0,1\}^n$. Cette caractéristique permet principalement un traitement parallèle de toute cette combinaison en même temps [5]. D'où la puissance du calcul quantique en terme de complexité algorithmique.

I.3.2. Mesure :

Etant donné un état $|\psi\rangle$ tel que :

$$|\psi\rangle=\alpha|0\rangle+\beta|1\rangle \tag{1.11}$$

La mesure de cet état est une projection dans une base $\{|0\rangle, |1\rangle\}$.

- L'amplitude de probabilité d'obtenir l'état $|0\rangle$ après mesure est la suivante :

$$|\alpha|^2=|\langle 0|\psi\rangle|^2 \tag{1.12}$$

- L'amplitude de probabilité d'obtenir l'état $|1\rangle$ après mesure est la suivante :

$$|\beta|^2=|\langle 1|\psi\rangle|^2 \tag{1.13}$$

Dans une mesure, le point le plus important à retenir est que cette opération est irréversible. Apres la mesure, les coefficients α et β seront perdus. C'est une projection dans la base $\{|0\rangle, |1\rangle\}$.

I.3.3. Evolution :

L'évolution d'un système quantique est décrite par une transformation unitaire. Cette évolution de l'état provient de l'application d'un opérateur linéaire, nommé opérateur d'évolution [6].

$$|\psi'\rangle=U|\psi\rangle \tag{1.14}$$

I.4. Portes quantiques :

Dans le monde quantique, il existe aussi les analogues des portes classiques: les portes logiques quantiques. On dispose d'une bibliothèque de transformation très riche mais les traitements sont plus complexes. Dans le cas général, une porte qui agit sur n Qbits est représentée par une matrice de taille $2^n \times 2^n$ [7,8].

I.4.1. Portes unaires :

Ce sont les portes les plus simples. Elles sont représentées par des matrices carrées d'ordre deux.

- ***Porte X :***

  Cette porte correspond à la porte NOT, sa matrice est donnée par :

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.15}$$

Sa table de vérité est donnée comme suit :

Entrée	Sortie
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$
$\alpha\|0\rangle + \beta\|1\rangle$	$\beta\|0\rangle + \alpha\|1\rangle$

Tab 1.1 : *Table de vérité de la porte quantique «X ».*

- ***Porte Y:***

La porte Y est représentée par la matrice carrée qui suit :

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{1.16}$$

Le tableau suivant représente la table de vérité de cette porte :

Entrée	Sortie
$\|0\rangle$	$i\|1\rangle$
$\|1\rangle$	$- i \|0\rangle$
$\alpha\|0\rangle + \beta\|1\rangle$	$-\beta i\|0\rangle + \alpha i\|1\rangle$

Tab1.2 : *Table de vérité de la porte quantique «Y».*

- ***Porte Z:***

Cette porte laisse inchangée le premier coefficient de l'entrée mais change le signe du second, sa matrice peut s'écrit comme suit :

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{1.17}$$

Entrée	Sortie
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$-\|1\rangle$
$\alpha\|0\rangle + \beta\|1\rangle$	$\alpha\|0\rangle - \beta\|1\rangle$

Tab 1.3 : *Table de vérité de la porte quantique «Z».*

Ces trois portes forment un groupe dit groupe de Pauli. Elles constituent une base permettant de générer tous les transformations possibles agissant sur un seul Qbit. Parmi les plus intéressantes, l'operateur de **Hadamard** noté H :

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.18}$$

La table de vérité de Hadamard est donnée par :

Entrée	Sortie
$\|0\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
$\|1\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$
$\alpha\|0\rangle + \beta\|1\rangle$	$\frac{\alpha+\beta}{\sqrt{2}}\|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}\|1\rangle$

Tab 1.4 : *Table de vérité de la porte quantique de « Hadamard ».*

I.4.2. Portes multi-Qbits :

- **Porte SWAP :**

  Cette porte permet la permutation des positions de deux Qbits. Elle est donnée par la matrice suivante :

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

Entrée	Sortie
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|10\rangle$
$\|10\rangle$	$\|01\rangle$
$\|11\rangle$	$\|11\rangle$
$\alpha\|00\rangle + \beta\|01\rangle + \gamma\|10\rangle + \delta\|11\rangle$	$\alpha\|00\rangle + \gamma\|01\rangle + \beta\|10\rangle + \delta\|11\rangle$

Tab 1.5 **:** *Table de vérité de la porte quantique « Swap ».*

Sa représentation graphique est la suivante :

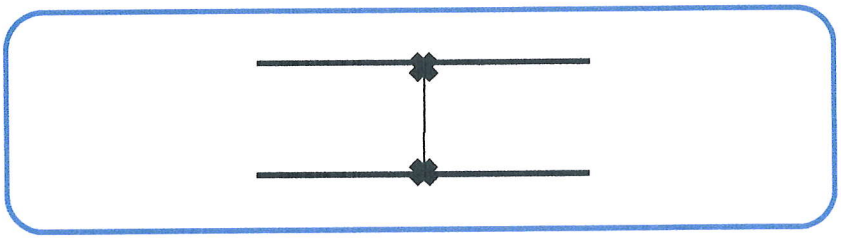


Fig.1.1 : *Porte Swap.*

➢ **Portes contrôlées :**

Ces portes agissent généralement sur plusieurs Qbits. Un Qbit de contrôle et des Qbits cibles.

- Si la valeur du Qbit de contrôle satisfait une certaine condition, on applique une transformation sur les Qbits cibles.
- Sinon, rien à faire.

Il est à noter que le Qbit de contrôle reste invariant. La représentation graphique d'une porte contrôlée est la suivante:

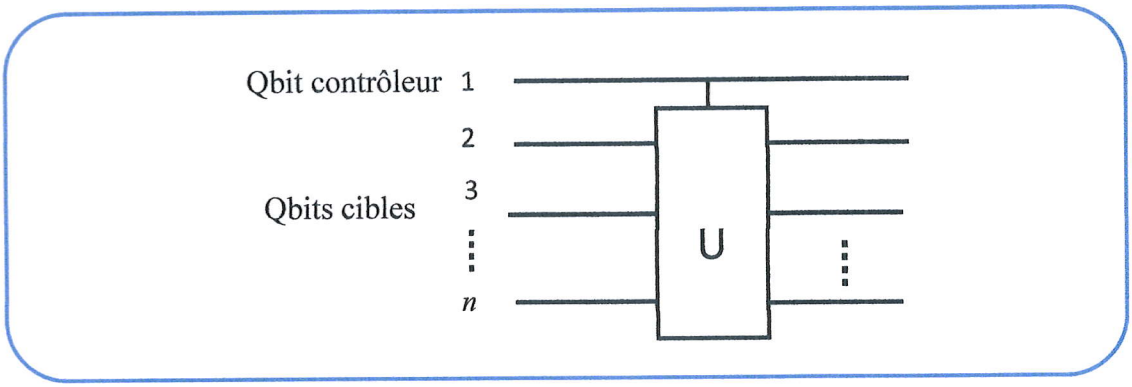


Fig.1.2 : *Porte contrôlée.*

Pour n= 2, la matrice de transformation correspondante à une porte contrôlée est donnée par la relation suivante :

$$C_U = I \oplus U \tag{1.19}$$

Remarque: $\oplus$ est l'opérateur somme directe.

- **Le Not contrôlé (CNot) :**

Le CNot est un cas particulier des portes contrôlées. C'est l'analogue quantique de la porte XOR classique. Il s'agit d'une transformation X contrôlée :

$$CNot \equiv I \oplus X \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Entrée	Sortie
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$
$\alpha\|00\rangle + \beta\|01\rangle + \gamma\|10\rangle + \delta\|11\rangle$	$\alpha\|00\rangle + \beta\|01\rangle + \delta\|10\rangle + \gamma\|11\rangle$

Tab 1.6 : *Table de vérité de porte « CNot ».*

Sa représentation graphique est la suivante :

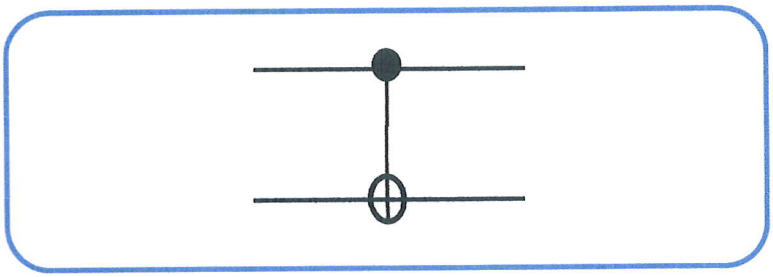


Fig.1.3 : *Porte quantique « CNOT ».*

- **La porte Toffoli :**

C'est une porte X contrôlée par deux Qbits de contrôle.

$$\text{TOFFOLI} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Entrée	Sortie
|000⟩	|000⟩
|001⟩	|001⟩
|010⟩	|010⟩
|011⟩	|011⟩
|100⟩	|100⟩
|101⟩	|101⟩
|110⟩	|111⟩
|111⟩	|110⟩
$a_1\|000\rangle + a_2\|001\rangle + a_3\|010\rangle + a_4\|011\rangle + a_5\|100\rangle + a_6\|101\rangle + a_7\|110\rangle + a_8\|111\rangle$	$a_1\|000\rangle + a_2\|001\rangle + a_3\|010\rangle + a_4\|011\rangle + a_5\|100\rangle + a_6\|101\rangle + a_8\|110\rangle + a_7\|111\rangle$

Tab 1.7 : *Table de vérité de « Toffoli ».*

I.5. Circuit quantique :

Un circuit quantique manipule un ensemble de Qbits. Il possède autant d'entrées que de sorties. Il est constitué d'un ensemble de portes qui agissent comme des transformations unitaires. Ces dernières peuvent être combinées en série ou en parallèle selon le traitement souhaité.

I.5.1. Composition en série :

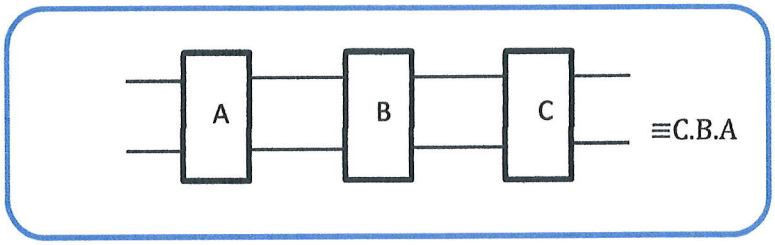
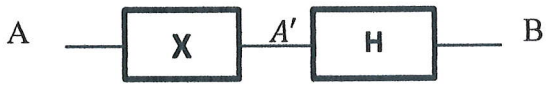


Fig.1.4 : *Circuit avec composition en série.*

- La matrice totale est calculée par le produit cartésien des matrices correspondantes aux portes mais dans l'ordre inverse [9].

Exemple :



Dans ce circuit :

$$|A'\rangle = X\ |A\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}$$

$$|B\rangle = H\ |A'\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} \beta+\alpha \\ \beta-\alpha \end{pmatrix}$$

La matrice de transfert globale T est donnée par:

$$T = H.X = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Tel que : $|B\rangle = T\ |A\rangle$

$$|B\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} \beta+\alpha \\ \beta-\alpha \end{pmatrix}$$

I.5.2. Composition parallèle :

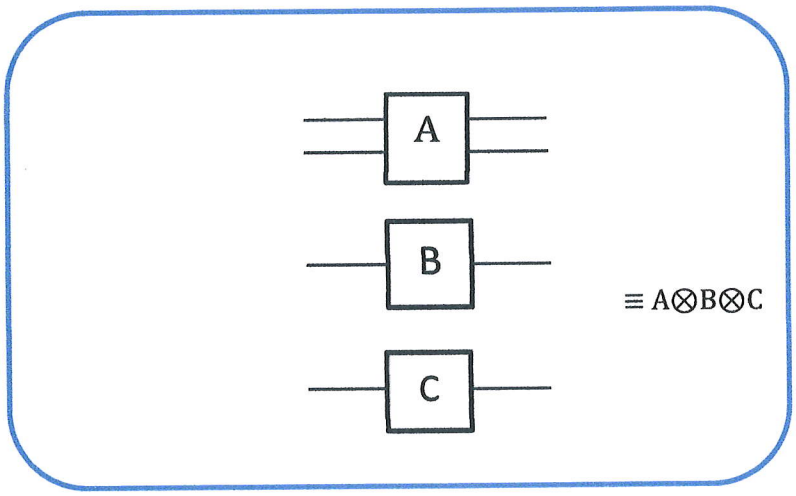
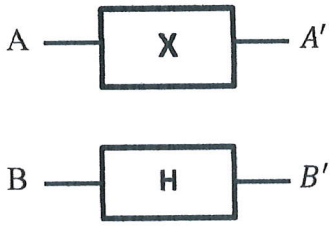


Fig.1.5 : *Circuit avec composition parallèle.*

- Dans ce cas, la matrice globale est obtenue par un produit tensoriel des matrices correspondantes aux portes utilisées [9].

Exemple :



On a : $|A'\rangle = X\ |A\rangle = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \begin{pmatrix}\beta\\ \alpha\end{pmatrix}$

$|B'\rangle = H\ |B\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}\gamma\\ \delta\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\gamma+\delta\\ \gamma-\delta\end{pmatrix}$

Le traitement séparé de ces deux Qbits donne comme résultat :

$$|A'B'\rangle = \begin{pmatrix}\beta\\ \alpha\end{pmatrix} \otimes \frac{1}{\sqrt{2}}\begin{pmatrix}\gamma+\delta\\ \gamma-\delta\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}\beta\gamma+\beta\delta\\ \beta\gamma-\beta\delta\\ \alpha\gamma+\alpha\delta\\ \alpha\gamma-\alpha\delta\end{pmatrix}$$

Passant maintenant au calcul de la matrice globale T, on a :

$$|A'B'\rangle = T\,|A\,B\rangle$$

Tel que :

$$T = X \otimes H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}$$

$$|A\,B\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \otimes \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix}$$

Donc :

$$|A'B'\rangle = T\,|A\,B\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{pmatrix} = \begin{pmatrix} \beta\gamma + \beta\delta \\ \beta\gamma - \beta\delta \\ \alpha\gamma + \alpha\delta \\ \alpha\gamma - \alpha\delta \end{pmatrix}$$

I.6. Conclusion :

Ce chapitre était dédié à la présentation des principes du calcul quantique. Nous avons donné un aperçu sur le formalisme mathématique ainsi que les postulats contrôlant cette théorie. On a constaté que c'est un domaine spécifique caractérisé par quelques phénomènes très étranges par rapport au calcul classique tels que la superposition et la mesure.

- La superposition: Un état quantique est une combinaison linéaire d'un ensemble états fondamentaux.
- La mesure : La mesure d'un état superposé d'un ensemble états fondamentaux est une projection sur l'un de ces états. C'est une opération irréversible.

En termes de calcul, on a présenté les différentes portes quantiques en donnant les matrices de transformation correspondantes. Aussi, on a donné un aperçu sur les circuits quantiques.

Le chapitre suivant sera consacré à la présentation de quelques algorithmes quantiques.

Chapitre II : Algorithmes quantiques

1. Introduction
2. Téléportation quantique
3. Algorithme de factorisation de shor
4. Correction d'erreurs quantiques
5. Algorithme de shor
6. Conclusion

II.1. Introduction :

Un algorithme quantique est une succession de transformations quantiques appliquées sur un ensemble de Qbits. Il est constitué d'un ensemble de portes simples et contrôlées spécifiées dans un ordre bien défini. Comparés aux algorithmes classiques, les algorithmes quantiques offrent une complexité nettement meilleure. Cela est dû principalement au parallélisme offert par le mécanisme superposition d'états. Cette puissance a attiré la tension de beaucoup de chercheurs.

Dans ce chapitre nous détaillons trois algorithmes quantiques très célèbres.

II.2. Téléportation quantique :

Imaginons deux amis, nommés Alice et Bob, éloignés géographiquement mais autorisés à transmettre de l'information classique [10]. Alice dispose du Qbit suivant :

$$|\psi> = \alpha|0> + \beta|1> \tag{2.1}$$

La téléportation quantique consiste à transférer l'état de ce Qbit d'Alice à Bob. Pour ce faire, ces deux personnages doivent partager la paire de Qbits suivante :

$$|\psi> = \frac{1}{\sqrt{2}}(|0>|0> + |1>|1>) \tag{2.2}$$

Donc, Alice possède deux Qbits indépendants. A ce niveau, le registre quantique contenant les trois Qbits est dans l'état :

$$|\psi>_a |\varphi>_{ab} = (\alpha|0>_a + \beta|1>_a)\frac{1}{\sqrt{2}}(|0>_a |0>_b + |1>_a |1>_b) \tag{2.3}$$

- les indices a et b identifient les Qbits possédés respectivement par Alice et Bob.

La figure suivante présente le circuit quantique permettant de réaliser cette opération.

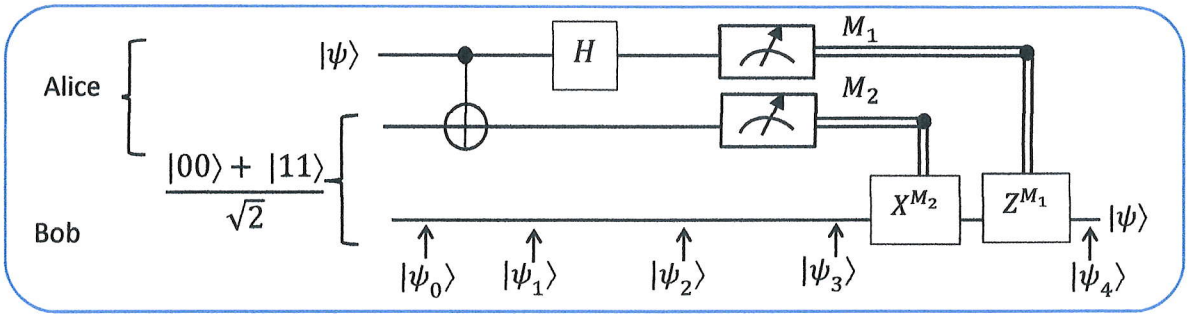


Fig2.1 : *Circuit quantique de téléportation* d'état.

En appliquant les premières opérations dans l'ordre spécifié, les états quantiques intermédiaires obtenus sont :

$$|\psi\rangle_0 = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle).$$

$$|\psi\rangle_1 = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle).$$

$$|\psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle).$$

$$|\psi_3\rangle = \frac{1}{2}[(|00\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle).$$

$$|\psi_4\rangle = \frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

Ensuite, Alice effectue une mesure sur ses deux Qbits. Cette opération se répercute immédiatement sur le Qbit de Bob.

$$\text{Mesure d'Alice} \quad \begin{matrix} m_1 & m_2 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} \begin{cases} \rightarrow |\psi_3\rangle = \alpha|\mathbf{0}\rangle + \beta|\mathbf{1}\rangle \\ \rightarrow |\psi_3\rangle = \alpha|\mathbf{1}\rangle + \beta|\mathbf{0}\rangle \\ \rightarrow |\psi_3\rangle = \alpha|\mathbf{0}\rangle - \beta|\mathbf{1}\rangle \\ \rightarrow |\psi_3\rangle = \alpha|\mathbf{1}\rangle - \beta|\mathbf{0}\rangle \end{cases}$$

Alice communique à Bob, via un canal de transmission classique, les valeurs de m_1 et m_2.
Bob retrouve alors $|\psi>$ sur son Qbit $(\alpha|0\rangle + \beta|1\rangle)$ en appliquant les portes $X = |0\rangle\langle 1| + |1\rangle\langle 0|$ et $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ si nécessaire à $|\psi_3\rangle$:

$$\boxed{\alpha|0\rangle + \beta|1\rangle = Z^{m1}X^{m2}|\psi_3\rangle} \qquad (2.4)$$

II.3. Algorithme de factorisation de Shor :

Dans les transactions importantes, les données sont souvent sécurisées via des protocoles cryptographiques puissants. Ils sont fondés principalement sur la difficulté de factoriser des nombres entiers de grandes tailles. Cette opération est très couteuse en termes de complexité algorithmiques [11].

En 1994, Peter Shor [12, 13] a publié un article qui a suscité beaucoup de réactions. Il a proposé un algorithme très intéressant permettant de factoriser les grands nombres entiers en un temps polynomial. Il est basé sur une procédure quantique dite transformée de Fourier quantique.

Algorithme :

a) Choisir a au hasard, $1< a <P$.

b) Si $PGCD\ (a, P) = 1$, continuer.
Sinon, le problème est résolu !

c) Utiliser un circuit quantique pour trouver r, la période de la fonction suivante :

$$f_a(k) = a^k \bmod P.$$

On a alors :

$$a^r = 1 \bmod P.$$

d) Si r est pair, alors :

$$(a^{r/2} + 1)\ (a^{r/2} - 1) = 0 \bmod P.$$

Si r est aussi tel que $a^{\frac{r}{2}} \neq \pm 1 \bmod P$, alors :

$$PGCD\ (a^{r/2}+1, P)$$

Et

$$PGCD\ (a^{r/2} - 1, P) \text{ sont des facteurs de } P : \text{stop !}$$

Sinon, retourner au pas a.

Dans cet algorithme, la partie quantique calculant la période est réalisée par le circuit suivant:

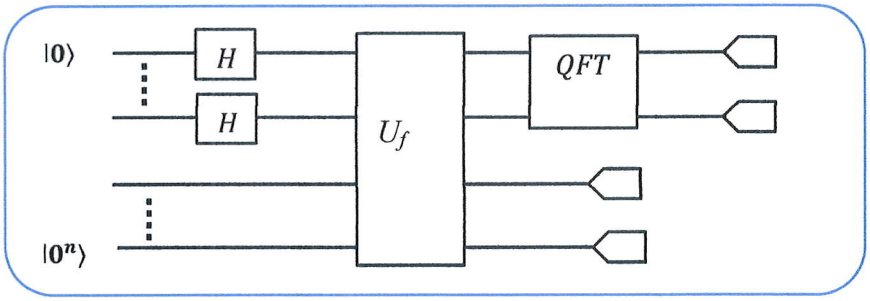


Fig.2.2 : *Le circuit quantique calculant la période.*

Exemple :

N = 21, a = 2

- On a besoin de deux registres :
  - Le premier comprend m Qbits tel que : $N^2 \leq 2^m < 2\,N^2$

    $m = 9$
  - Le deuxième registre comprend n Qbits où : n = Log (N)

    $n = 5$

  Les deux registres sont initialisés dans les états de base:

  $|000000000>$

  $|00000>$

- Par abréviation, on a l'état initial : $|\psi_0> = |0>|0>$
- On applique m porte Hadamard sur les Qbits du premier registre, On aura :

$$|\psi_1> = \frac{1}{\sqrt{512}}\sum_{j=0}^{512}|j>|0>$$

- On applique l'opérateur U_f :

$$|\psi_2> = \frac{1}{\sqrt{512}}[(|0>+|6>\cdots+|510>)|1>+(|1>+|7>+|13>+..+|51>)$$
$$|2>+(|2>+|8>+|14>+\cdots+|506>)|4>+(|3>+|9>+|15>+\cdots+$$
$$|507>)|8>+(|4>+|10>+|16>+\cdots+|508>)|16>+(|5>+|11>+$$
$$|17>+\cdots+|509>)|11>]$$

Le premier registre est à présent formé d'une superposition d'états, dont les numéros d'ordre forment une suite périodique de période précisément égale à r

- On mesure l'état du second registre, supposons que le résultat de mesure est 2.
- Le nouvel état est suivant :

$$|\psi_3> = \frac{1}{\sqrt{86}}((|1>+|7>+|13>+..+|511>)|2>) = \frac{1}{\sqrt{86}}\sum_{l=0}^{85}|6l+1>|2>.$$

- Le premier registre est à présent formé d'une superposition d'états dont les numéros d'ordre forment une suite périodique de période égale à r
- On applique la QFT sur le premier registre.

Le résultat est le suivant :

$$|\psi_4> = (0.11|85>+0.11|171>+0.17|256>+0.11|341>+0.11|427>)$$

- En théorie, la période cherchée vaut l'inverse de la fréquence la plus basse :
  - $r = 1 / \nu_0 = 512 / 85 = 6.023$ $(256 = 2^m = 2^9)$

- Mais la mesure est aléatoire. Supposant qu'on a le résultat : 427
  - le calcul de r se fait sur base du développement en fractions continues

de la valeur trouvée pour la fréquence

$V_4 = 427$

$$V_4 = \frac{427}{512} = \frac{k}{r} = \frac{1}{1 + \frac{1}{5 + \frac{1}{42 + \frac{1}{2}}}}$$

Soit la suite des approximations, 1/1, ***5/6***, 211/253, 427/512.

- On trouve la période cherchée, r, comme le dénominateur de l'approximant le plus précis dont le dénominateur n'excède toutefois pas N (en effet, r <21).

$$r = 6.$$

Les facteurs de N sont : $PGCD(a^{\frac{r}{2}} \pm 1,\ N)$.

- Facteur 01 : $PGCD(7, 21) = 7.$
- Facteur 02 : $PGCD(9, 21) = 3.$

21 = 7 * 3

II.4. Correction d'erreurs quantiques :

II.4.1. Particularités du cas quantique :

Dans le cas classique, les techniques de correction d'erreurs sont basées sur l'ajout d'informations redondantes (codage). Cela permet de tester si le message codé a été perturbé et de corriger les erreurs. Parmi ces codes, le plus simple est le codage à trois bits. Il consiste à tripler chaque bit d'information :

- $0 \rightarrow 000.$
- $1 \rightarrow 111.$

On suppose que la probabilité d'erreur est suffisamment faible pour que la probabilité que deux erreurs surviennent simultanément soit négligeable. Donc après avoir traversé un canal susceptible de créer une erreur, le triplet de bits peut se retrouver avec au plus un bit inversé. La détection de l'erreur se fait en testant si tous les bits sont égaux ou non. En cas d'inégalités, on utilise la règle de

la majorité pour rétablir la bonne valeur logique associée au triplet. C'est ce qu'on appelle le décodage.

Mais l'information quantique se heurte à des difficultés non triviales qui n'ont pas d'analogues dans le traitement classique de l'information:

- la forme du Qbit est différente de celle du bit classique. Dans le cas classique, un bit est soit un 0 ou un 1, alors que dans le cas quantique, c'est un état de la forme :

  $$|\psi> = \alpha|0> + \beta|1>$$

  C'est-à-dire deux nombres complexes.
- Les erreurs quantiques possibles sont plus nombreuses que celles classiques. Tandis que classiquement une erreur sur un bit est une simple interversion, une erreur sur un Qbit peut être n'importe quelle transformation unitaire.
- Pas de clonage: il est impossible de dupliquer des états quantiques pour obtenir le code de répétition.
- Les erreurs sont permanentes: un continuum d'erreurs différentes peut se produire sur un seul Qbit, déterminer quelle erreur s'est produite afin de la corriger nécessiterait une précision infinie.
- La mesure détruit les informations quantiques: les informations classiques peuvent être observées sans être détruites puis décodées, mais les informations quantiques sont détruites par la mesure et ne peuvent pas être récupérées.

Malgré toutes ces difficultés, la correction d'erreurs quantiques est possible. En plus, dans le cadre quantique, les recherches affirment que si nous pouvons corriger à la fois les erreurs X, Y et Z, alors nous pouvons corriger du même coup une très large classe d'erreurs à un Qbit. C'est une conséquence directe des propriétés du groupe de Pauli.

II.4.2. Codes de correction à répétition :

II.4.2.1. Correction d'erreur de type X ou Bit-flip :

Le circuit qui permet de corriger une erreur de type X est le suivant [14] :

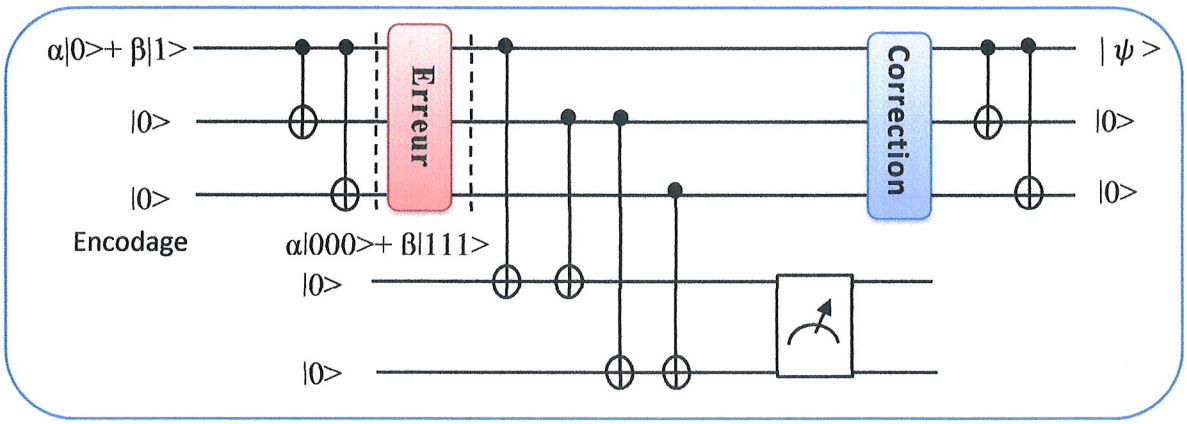


***Fig.2.3** : Circuit de correction d'erreurs de type « X ».*

Encodage : Soit le Qbit $|\psi>$ défini par : $|\psi> = \alpha|0> + \beta|1>$.

L'étape d'encodage est basée sur une redondance spécifique. Elle consiste à l'ajout de deux Qbits intriqués au Qbit initial. Elle est réalisée en appliquant deux portes CNot.

- L'état initial : $|\psi_1> = \alpha|000> + \beta|100>$
- L'application du CNot (1,2) donne l'état : $|\psi_2> = \alpha|000> + \beta|110>$
- L'application du CNot (1,3) donne l'état : $|\psi_3> = \alpha|000> + \beta|111>$

L'encodage d'un Qbit : $|\psi> = \alpha|0> + \beta|1>$ donne un état quantique $|\psi_3>$ de trois Qbits tel que:

$|\psi_3\rangle = \alpha|000\rangle + \beta|111\rangle$

Détection d'erreurs : La détection d'erreurs X est basée sur l'utilisation de deux Qbits supplémentaires initialisés à $|0\rangle$ comme syndrome. Ce dernier est calculé par quatre portes CNot. Dans la suite, nous présentons les quatre cas possibles.

❖ **Cas 01 : sans erreur :**

- $|\psi_4\rangle = \alpha|000> + \beta|111>$

- L'ajout des deux Qbits du syndrome donne l'état :

$$|\psi_5\rangle = \alpha|00000\rangle + \beta|11100\rangle$$

- L'application des portes CNot donne les états suivants :

$C_{[1,4]} \Longrightarrow |\psi\rangle = \alpha|00000\rangle + \beta|11110\rangle$

$C_{[2,4]} \Longrightarrow |\psi_7\rangle = \alpha|00000\rangle + \beta|11100\rangle$

$C_{[2,5]} \Longrightarrow |\psi_8\rangle = \alpha|00000\rangle + \beta|11101\rangle$

$C_{[3,5]} \Longrightarrow |\psi_9\rangle = \alpha|00000\rangle + \beta|11100\rangle$

Dans ce cas : $|\psi_9\rangle = (\alpha|000\rangle + \beta|111\rangle)\,|00\rangle$

❖ Cas 02 : Erreur X sur le 1er Qbit :

- $|\psi_4\rangle = \alpha|100\rangle + \beta|011\rangle$
- L'ajout des deux Qbits du syndrome donne l'état :

$$|\psi_5\rangle = \alpha|10000\rangle + \beta|01100\rangle$$

- L'application des portes CNot donne les états suivants :

$C_{[1,4]} \Longrightarrow |\psi_6\rangle = \alpha|10010\rangle + \beta|01100\rangle$

$C_{[2,4]} \Longrightarrow |\psi_7\rangle = \alpha|10010\rangle + \beta|01110\rangle$

$C_{[2,5]} \Longrightarrow |\psi_8\rangle = \alpha|10010\rangle + \beta|01111\rangle$

$C_{[3,5]} \Longrightarrow |\psi_9\rangle = \alpha|10010\rangle + \beta|01110\rangle$

Dans ce cas : $|\psi_9\rangle = (\alpha|100\rangle + \beta|011\rangle)\,|10\rangle$

❖ Cas 03 : Erreur X sur le 2eme Qbit :

- $|\psi_4\rangle = \alpha|010\rangle + \beta|101\rangle$
- L'ajout des deux Qbits du syndrome donne l'état :

$$|\psi_5\rangle = \alpha|01000\rangle + \beta|10100\rangle$$

- L'application des portes CNot donne les états suivants :

$C_{[1,4]} \Longrightarrow |\psi_6\rangle = \alpha|01000\rangle + \beta|10110\rangle$

$C_{[2,4]} \Longrightarrow |\psi_7\rangle = \alpha|01010\rangle + \beta|10110\rangle$

$C_{[2,5]} \Longrightarrow |\psi_8\rangle = \alpha|01011\rangle + \beta|10110\rangle$

$C_{[3,5]} \Longrightarrow |\psi_9\rangle = \alpha|01011\rangle + \beta|10111\rangle$

Dans ce cas : $|\psi_9\rangle = (\alpha|010\rangle + \beta|101\rangle)\,|11\rangle$

❖ Cas 04 : Erreur X sur le 3eme Qbit :

- $|\psi_4\rangle = \alpha|001> + \beta|110>$
- L'ajout des deux Qbits du syndrome donne l'état :

$$|\psi_5\rangle = \alpha|00100> + \beta|11000>$$

- L'application des portes CNot donne les états suivants :

$C_{[1,4]} \Longrightarrow |\psi_6\rangle = \alpha|00100\rangle + \beta|11010\rangle$

$C_{[2,4]} \Longrightarrow |\psi_7\rangle = \alpha|00100\rangle + \beta|11000\rangle$

$C_{[2,5]} \Longrightarrow |\psi_8\rangle = \alpha|00100\rangle + \beta|11001\rangle$

$C_{[3,5]} \Longrightarrow |\psi_9\rangle = \alpha|00101\rangle + \beta|11001\rangle$

Dans ce cas : $|\psi_9\rangle = (\alpha|001\rangle + \beta|110\rangle)\,|01\rangle$

- **La mesure :** Dans cette étape, les deux Qbits du syndrome sont mesurés. Les valeurs obtenues sont suffisantes pour la correction.
- **Correction :** On applique l'opérateur unitaire X_i^{-1} sur le Qbit erroné afin de retrouver son état initial. Comme l'opérateur X est unitaire alors : $X_i^{-1} = X_i$

Cette correction est résumée dans le tableau suivant :

$\|Q_4\ Q_5\rangle$	Correction
$\|\mathbf{00}\rangle$	Ne rien faire
$\|\mathbf{01}\rangle$	X (3)
$\|\mathbf{10}\rangle$	X (1)
$\|\mathbf{11}\rangle$	X (2)

Tab2.1 : *Correction d'erreur de type « X ».*

Ce qui permet de revenir dans tous les cas à l'état : $|\psi_9\rangle = \alpha|000\rangle + \beta|111\rangle$

- **Décodage :** Le décodage sert à isoler le Qbit initial des deux Qbits d'intrication. Il est réalisé par application de l'inverse des portes utilisées dans l'étape l'encodage.
- L'état initial : $|\psi_9> = \alpha|000> + \beta|111>$
- L'application du CNot (1,2) donne l'état :

$|\psi_{10}> = \alpha|000> + \beta|101>$

➢ L'application du CNot (1,3) donne l'état :

$$|\psi_{11}> = \alpha|000> + \beta|100> = |\psi>|00>$$

II.4.2.2. Correction d'erreur de type Z ou Phase-flip :

Comme vu dans le chapitre précédent, les portes X, Z et H sont définies par les matrices unitaires suivantes [14] :

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

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Calculant la matrice correspondante à la transformation : HZH.

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

Ce résultat affirme que la correction d'une erreur de type Z se ramène à celle d'une erreur de type X. Il suffit d'ajouter une transformation H à la fin de l'encodage et son inverse (H) avant la détection d'erreur. Donc, le circuit de correction est le suivant :

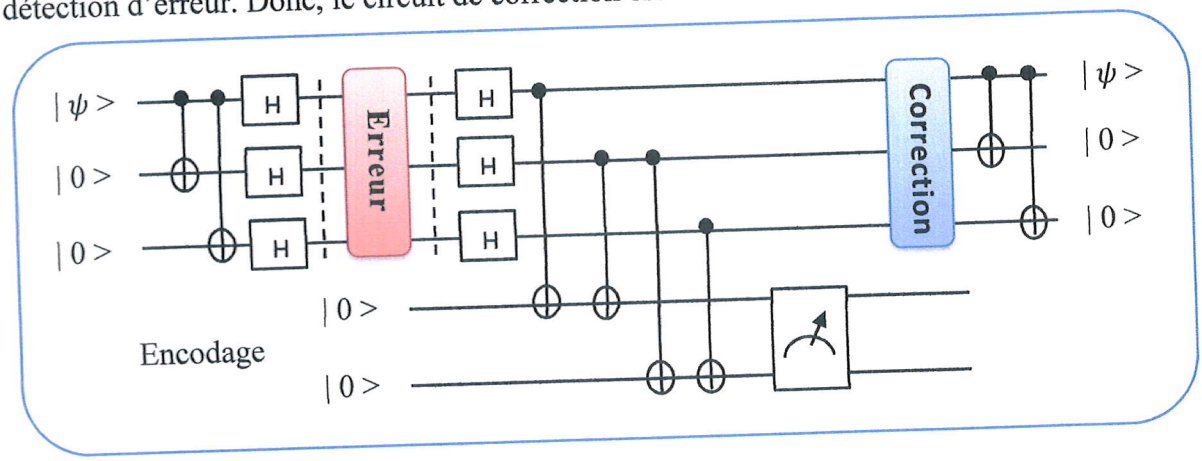


Fig.2.4 : *Circuit de transformation du code phase-flip à trois Qbit.*

Remarques :

- Une erreur de type Z sur un Qbit est transformée en une erreur de type X sur le même Qbit.
- Les étapes suivantes sont les mêmes vues dans le traitement des erreurs X.

II.4.2.3. Correction d'erreur de type Y:

De même, Calculant :

$$iZX \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = Y$$

Donc, une erreur Y peut être corrigée par application d'un correcteur d'erreurs Z suivi d'un correcteur d'erreurs X [14].

II.5. Algorithme de Shor :

Un système quantique pourrait avoir des erreurs de type X, Y et Z. Shor a développé un code de correction universel capable de corriger ces trois types d'erreurs à la fois [14]. L'idée principale est de "concaténer" les deux codes de répétition présentés dans le paragraphe précédent :

- Un correcteur protégeant le Qbit $|\psi>$ des erreurs de type Z. Il utilise deux Qbits supplémentaires.
- Chacun de ces trois Qbits est protégé à son tour par un correcteur des erreurs de type X.
- Donc, c'est un correcteur à 9 Qbits. Le schéma présenté dans la page suivante donne l'architecture globale de cette solution :

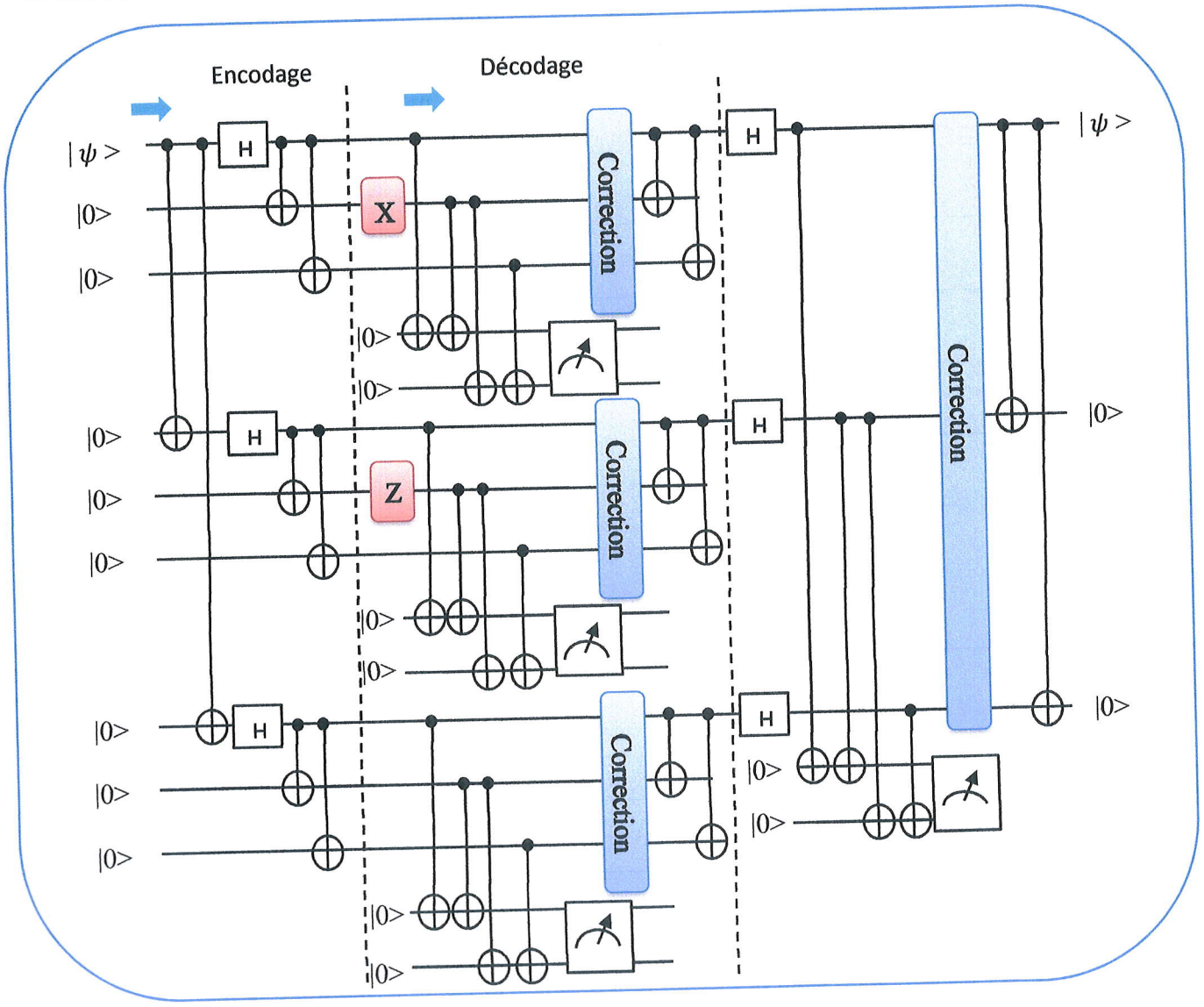


Fig.2.5 : Circuit de correction de Shor.

II.6. Conclusion :

En algorithmique quantique, les progrès réalisés sont relativement lents, il existe peu d'algorithmes quantiques. Dans ce chapitre, nous avons présenté les plus célèbres. Le premier est utilisé pour la téléportation d'un état quantique entre deux localités distantes dans l'espace. Le second est celui de la factorisation des grands nombres entiers. Enfin, le dernier corrige les erreurs quantiques. Il est basé sur un codage à 9 Qbits.

Les recherches affirment qu'un système quantique est très sensible aux erreurs du à l'interaction avec l'environnement. De ce fait, la correction d'erreurs quantiques est inévitable et joue un rôle très important. Dans le chapitre suivant, nous nous intéressons à une solution plus optimale nécessitant uniquement 5 Qbits pour l'encodage.

Chapitre III : Codes stabilisateurs

1. Introduction
2. Idée de base
3. Code à neuf Qbits
4. Code à sept Qbits
5. Code à cinq Qbits
6. Conclusion

III.1. Introduction :

Ces dernières années, des évolutions importantes ont eu lieu et qui semblent prometteuses au niveau du codage de l'information quantique notamment dans le domaine de la correction d'erreurs. Un grand travail théorique a donné naissance à des nouvelles familles de codes. Généralement, les solutions proposées utilisent la redondance d'informations comme outil de détection. C'est une adaptation des correcteurs classiques en tirant profit de l'intrication afin de délocaliser sur plusieurs systèmes physiques l'information encodée.

Dans ce chapitre nous nous intéressons aux codes stabilisateurs [15,16]. Ils sont basés principalement sur les algèbres des opérateurs de Pauli. Dans la suite, nous présentons trois variantes : Un code à 9 Qbits équivalent à celui de Shor, un autre à 7 Qbits et enfin un dernier à 5 Qbits.

III.2. Idée de base :

Afin d'introduire les codes stabilisateurs, prenons un cas très simple. Soit le Qbit ψ défini par :

$$|\psi> = \alpha|0> + \beta|1> \qquad (3.1)$$

Pour protéger ce Qbit des erreurs, on commence d'abord par une étape d'encodage basée sur la redondance. Elle consiste à l'ajout de deux Qbits supplémentaires intriqués au Qbit initial. Elle est réalisée en appliquant deux portes CNot.

- L'état initial : $\alpha|000> + \beta|100>$
- L'application du CNot (1,2) donne l'état :

$$\alpha|000> + \beta|110>$$

- L'application du CNot (1,3) donne l'état :

$$|\psi_c> = \alpha|000> + \beta|111>$$

Supposons maintenant que le seul type d'erreur qui endommage le Qbit : $|\psi_c>$ soit un bit-flip (réalisée par l'action de l'opérateur X); il y a trois possibilités :

$$X_1[\alpha|000> + \beta|111>] = \alpha|100> + \beta|011>$$

$$X_2[\alpha|000> + \beta|111>] = \alpha|010> + \beta|101>$$

$$X_3[\alpha|000> + \beta|111>] = \alpha|001> + \beta|110>$$

La technique basée sur les codes stabilisateurs procède de la manière suivante :

- On définit des opérateurs de syndrome qui ont la particularité d'avoir comme états propres les états erronés ainsi que l'état non perturbe.
- On mesure ces operateurs; les états, erronés ou non, ne sont pas modifiés puisqu'ils sont états propres et le résultat de la mesure (les valeurs propres) signent la position de l'erreur. Il suffit alors d'apporter la correction.

Dans notre cas, ces opérateurs de syndrome sont : $S_1 = Z_1Z_2$ et $S_2 = Z_2Z_3$. Il est facile de vérifier que :

- Les états erronés ainsi que l'état non perturbe sont états propres de ces deux opérateurs.
- De plus, le couple de valeurs propres signe sans ambiguïté la position de l'erreur. Le tableau ci-dessous indique les valeurs propres du couple (S_1, S_2) pour chacun des états.

$$|\psi_0> = |\psi> = \alpha|000> + \beta|111> \rightarrow (+1,+1)$$
$$|\psi_1> = X_1|\psi> = \alpha|100> + \beta|011> \rightarrow (-1,+1)$$
$$|\psi_2> = X_2|\psi> = \alpha|010> + \beta|101> \rightarrow (-1,-1)$$
$$|\psi_3> = X_3|\psi> = \alpha|001> + \beta|110> \rightarrow (+1,-1)$$

Donc, la mesure du syndrome permet de détecter le type d'erreur. La mesure quantique de chacune des valeurs propres est mise en œuvre par le circuit suivant :

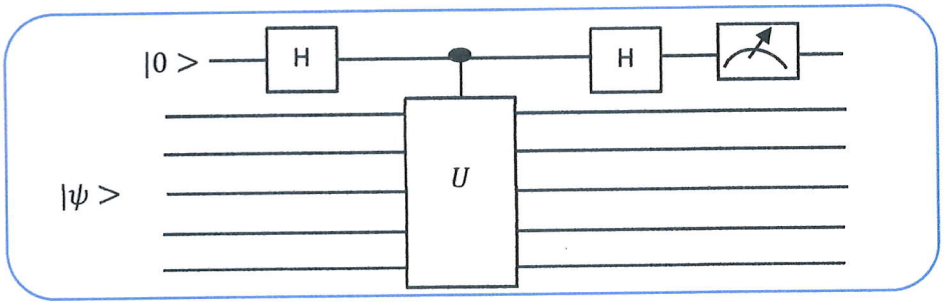


Fig.3.1 : *Circuit quantique pour la mesure des syndromes.*

- On utilise un Qbit auxiliaire $|0>$ que l'on fait passer une porte Hadamard, l'état est alors de

$$\frac{|0>+|1>}{\sqrt{2}} \otimes |\psi>$$

- On effectue une porte U contrôlée par le Qbit auxiliaire conduisant :

$$\frac{1}{\sqrt{2}}(|0>|\psi>+|1>U|\psi>)$$

- On effectue de nouveau une porte Hadamard sur le Qbit auxiliaire ce qui donne :

$$\frac{1}{\sqrt{2}}\left(\frac{|0>+|1>}{\sqrt{2}}\left|\psi>+\frac{|0>-|1>}{\sqrt{2}}U\right|\psi>\right) = |0>\frac{|\psi>+U|\psi>}{2}+|1>\frac{|\psi>-U|\psi>}{2}$$

- On mesure ensuite le Qbit auxiliaire.

  - ✓ Si on obtient 0 la valeur propre est 1.
  - ✓ Par contre, si on obtient 1 la valeur propre est -1.

Remarques :

- Les trois états erronés ainsi que l'état sans erreur sont des vecteurs propres de deux opérateurs Z_1Z_2 et Z_2Z_3.
- Seul le mot de code valide est un vecteur propre des deux opérateurs Z_1Z_2 et Z_2Z_3 avec une valeur propre +1.
- Les deux opérateurs Z_1Z_2 et Z_2Z_3 forment un groupe S, appelé stabilisateur du code.
- Z_1Z_2 et Z_2Z_3 sont appelés générateurs de ce groupe.
- Z_1Z_2 et Z_2Z_3 commutent entre eux.
- Le couple de valeurs propres calculées pour chaque cas est appelée syndrome.
- Par linéarité des opérations quantique, l'encodage d'un état :

  $|\psi> = \alpha|0> + \beta|1>$

  Donne :

  $|\psi_c> = \alpha|0\rangle_c + \beta|1\rangle_c$

  Cette notation sera utilisée dans le reste de ce mémoire.
- Les opérations quantiques nécessaires dans l'étape de décodage sont les mêmes utilisées dans le processus d'encodage, mais elles doivent être appliquées dans l'ordre inverse.

III.3. Code à neuf Qbits :

Encodage :

$$|0\rangle_c = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$

$$= \frac{1}{2\sqrt{2}}(|000000000\rangle + |000000111\rangle + |000111000\rangle + \dots)$$

$$|1\rangle_c = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

$$= \frac{1}{2\sqrt{2}}(|000000000\rangle - |000000111\rangle - |000111000\rangle + |000111111\rangle + \cdots)$$

- Le circuit d'encodage correspondant est donné dans le chapitre précèdent.

Générateurs du groupe stabilisateur :

M_1	Z	Z	I	I	I	I	I	I	I
M_2	Z	I	Z	I	I	I	I	I	I
M_3	I	I	I	Z	Z	I	I	I	I
M_4	I	I	I	Z	I	Z	I	I	I
M_5	I	I	I	I	I	I	Z	Z	I
M_6	I	I	I	I	I	I	Z	I	Z
M_7	X	X	X	X	X	X	I	I	I
M_8	X	X	X	I	I	I	X	X	X

Tab 3.1 : *Le stabilisateur pour le code de Shor à 9 Qbits.*

Syndromes :

Le syndrome associé à chaque type d'erreur est donné par le circuit suivant :

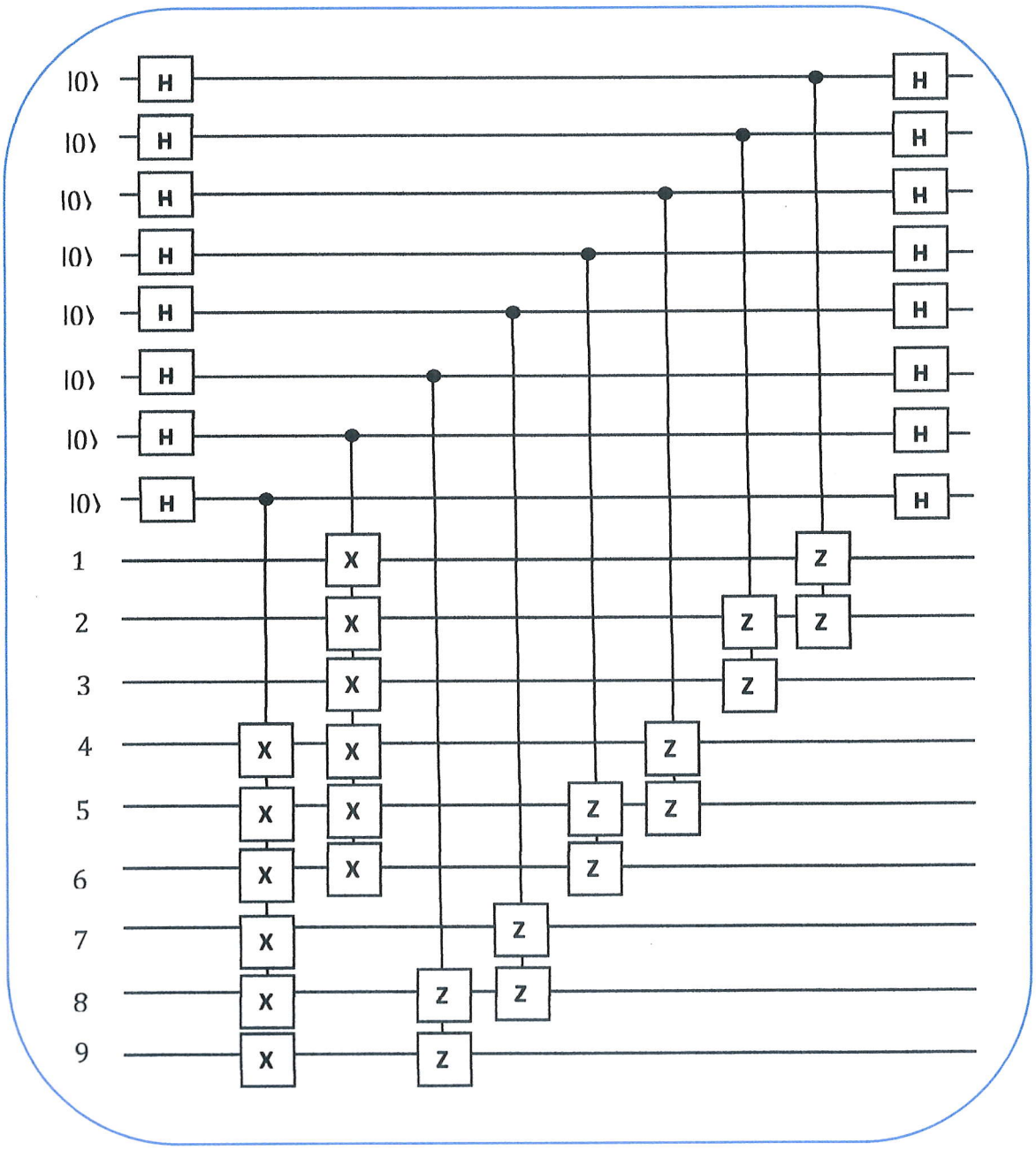


Fig.3.2 : Circuit pour la mesure du syndrome d'erreur du code correcteur à 9 Qbits de Shor.

Décodage :

- De même, le circuit décodage correspondant est donné dans le chapitre précèdent.

III.4. Code à sept Qbits :

Encodage :

$$|0\rangle_c = \frac{1}{\sqrt{8}}(1 + M_0)(1 + M_1)(1 + M_2)|0000000\rangle \tag{3.2}$$

$$|1\rangle_c = \frac{1}{\sqrt{8}}(1 + M_0)(1 + M_1)(1 + M_2)|1111111\rangle \tag{3.3}$$

Tel que :

- $M_0 = X_0X_4X_5\,X_6$
- $M_1 = X_1X_3X_5\,X_6$
- $M_2 = X_2X_3X_4\,X_6$

Le circuit d'encodage correspondant est le suivant [10]:

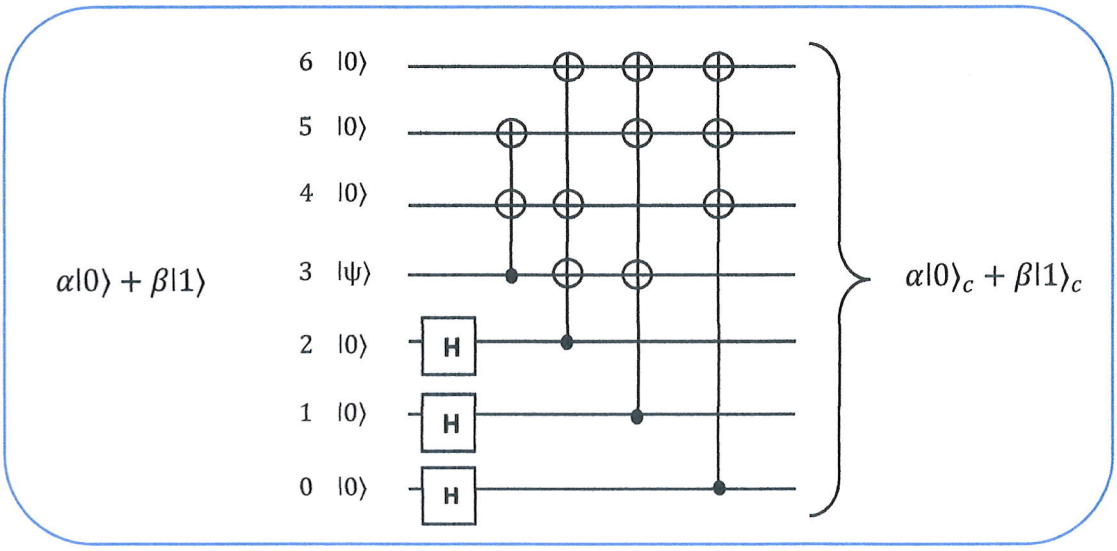


Fig.3.3 **:** *Circuit quantique pour l'encodage d'un Qbit en 7 Qbits.*

Générateurs du groupe stabilisateur :

$M_0 = X_0X_4X_5\,X_6$ $\quad$ $N_0 = Z_0Z_4Z_5Z_6$
$M_1 = X_1X_3X_5\,X_6$ $\quad$ $N_1 = Z_1Z_3Z_5Z_6$
$M_2 = X_2X_3X_4\,X_6$ $\quad$ $N_2 = Z_2Z_3Z_4Z_6$

6	5	4	3	2	1	0
X	X	X	I	I	I	X
X	X	I	X	I	X	I
X	I	X	X	X	I	I
Z	Z	Z	I	I	I	Z
Z	Z	I	I	Z	I	Z
Z	I	Z	Z	Z	I	I

Tab 3.2 : *Le stabilisateur pour le code à 7 Qbits.*

Syndromes :

Les syndromes d'erreurs pour le code à sept Qbits sont résumés dans le tableau qui suit :

	$X_0Y_0Z_0$	$X_1Y_1Z_1$	$X_2Y_2Z_2$	$X_3Y_3Z_3$	$X_4Y_4Z_4$	$X_5Y_5Z_5$	$X_6Y_6Z_6$	1
$M_0 = Z_1Z_2Z_3Z_7$	− − +	+ + +	+ + +	+ + +	− − +	− − +	− − +	+
$M_1 = Z_1Z_2Z_4Z_6$	+ + +	− − +	+ + +	− − +	+ + +	− − +	− − +	+
$M_2 = Z_1Z_4Z_5Z_6$	+ + +	+ + +	− − +	− − +	− − +	+ + +	− − +	+
$M_3 = X_1X_4X_5X_6$	+ − −	+ + +	+ + +	+ + +	+ − −	+ − −	+ − −	+
$M_4 = X_1X_2X_4Z_6$	+ + +	+ − −	+ + +	+ − −	+ + +	+ − −	+ − −	+
$M_5 = X_1X_4X_5X_6$	+ + +	+ + +	+ − −	+ − −	+ − −	+ + +	+ − −	+

Tab 3.3 : *Syndromes d'erreur pour le code à 7 Qbits.*

Ces valeurs peuvent être calculées en utilisant le circuit suivant [10]:

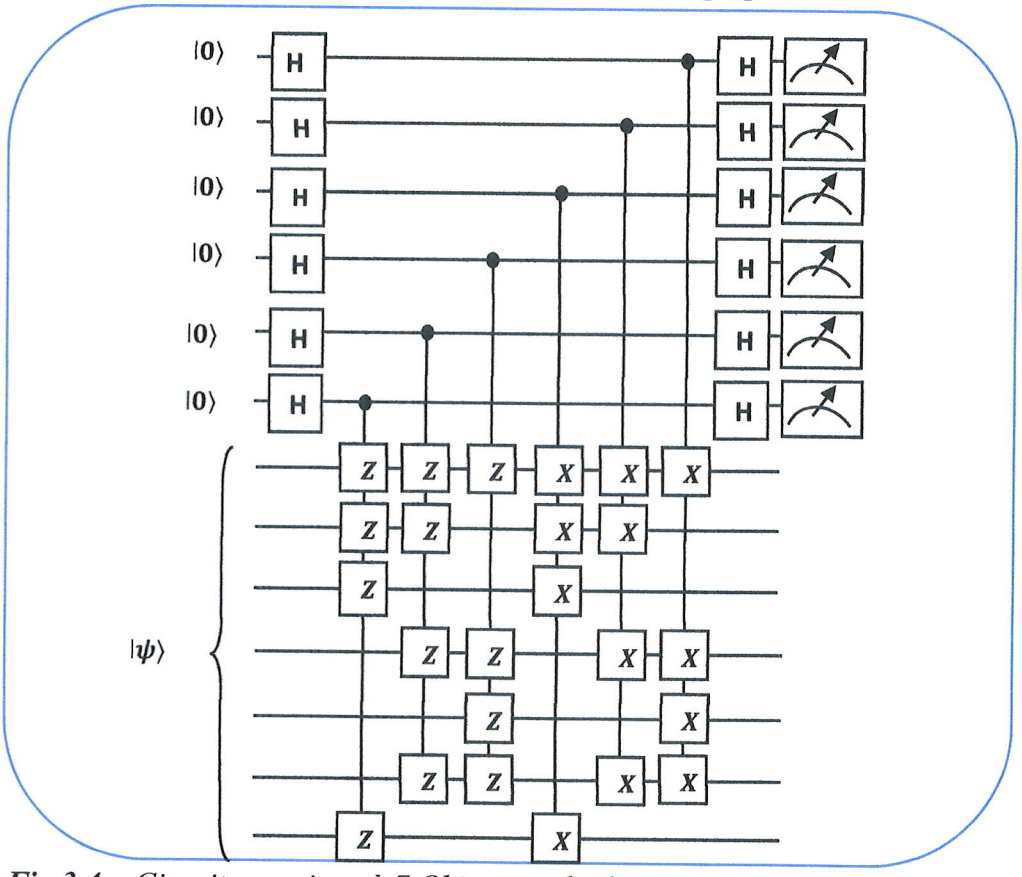


Fig.3.4 : *Circuit quantique à 7 Qbits pour la détection des syndromes d'erreur.*

Décodage :

Le circuit de décodage est le suivant :

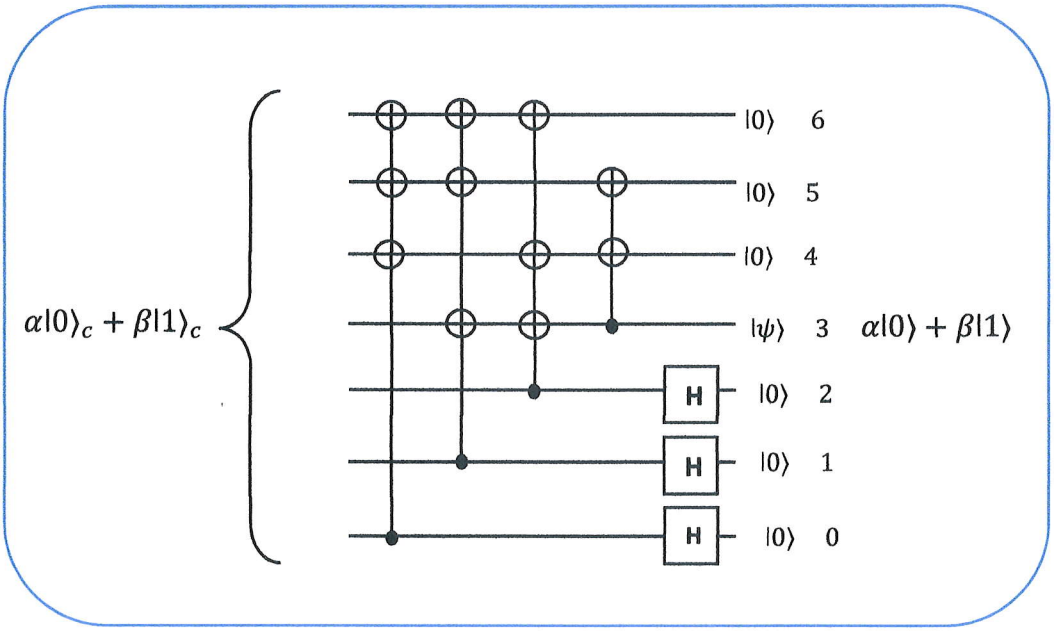


Fig.3.5 **:** *Circuit quantique pour le décodage de 7 Qbits.*

III.5. Code à Cinq Qbits :

Encodage :

$$|0\rangle_c = \frac{1}{4}(1+M_0)(1+M_1)(1+M_2)(1+M_3)|00000\rangle \quad (3.4)$$

$$|1\rangle_c = \frac{1}{4}(1+M_0)(1+M_1)(1+M_2)(1+M_3)|11111\rangle \quad (3.5)$$

Tel que :

- $M_0 = Z_1X_2X_3Z_4$
- $M_1 = Z_2X_3X_4Z_0$
- $M_2 = Z_3X_4X_0Z_1$
- $M_3 = Z_4X_0X_1Z_2$

Le circuit correspondant est le suivant [10]:

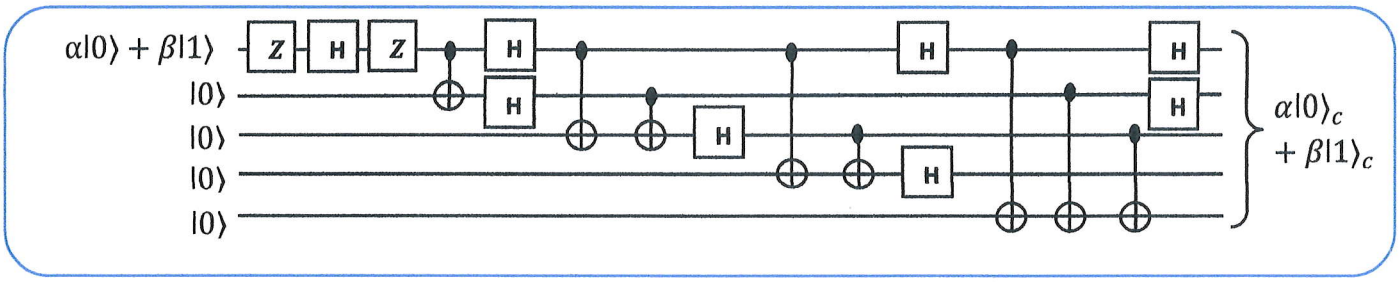


Fig.3.6 : *Circuit pour coder un Qbit sur 5 Qbits.*

Générateurs du groupe stabilisateur :

Le stabilisateur de code à cinq Qbits est donné dans le tableau 3.4 [10].

	4	3	2	1	0
M_0	Z	X	X	Z	I
M_1	X	X	Z	I	Z
M_2	X	Z	I	Z	X
M_3	Z	I	Z	X	X

Tab 3.4 : *Le stabilisateur pour le code à 5 Qbits.*

Syndromes :

Les valeurs propres associées à chaque cas sont les suivantes :

	$X_0Y_0Z_0$	$X_1Y_1Z_1$	$X_2Y_2Z_2$	$X_3Y_3Z_3$	$X_4Y_4Z_4$	1
$M_0 = Z_1X_2X_3Z_4$	$+++$	$--+$	$+--$	$+--$	$--+$	$+$
$M_1 = Z_2X_3X_4Z_0$	$--+$	$+++$	$--+$	$+--$	$+--$	$+$
$M_2 = Z_3X_4X_0Z_1$	$+--$	$--+$	$+++$	$--+$	$+--$	$+$
$M_3 = Z_4X_0X_1Z_2$	$+--$	$+--$	$--+$	$+++$	$--+$	$+$

Tab 3.5 : *Syndromes d'erreur pour le code à 5 Qbits.*

Le circuit quantique permettant de réaliser ces calculs est le suivant [10]:

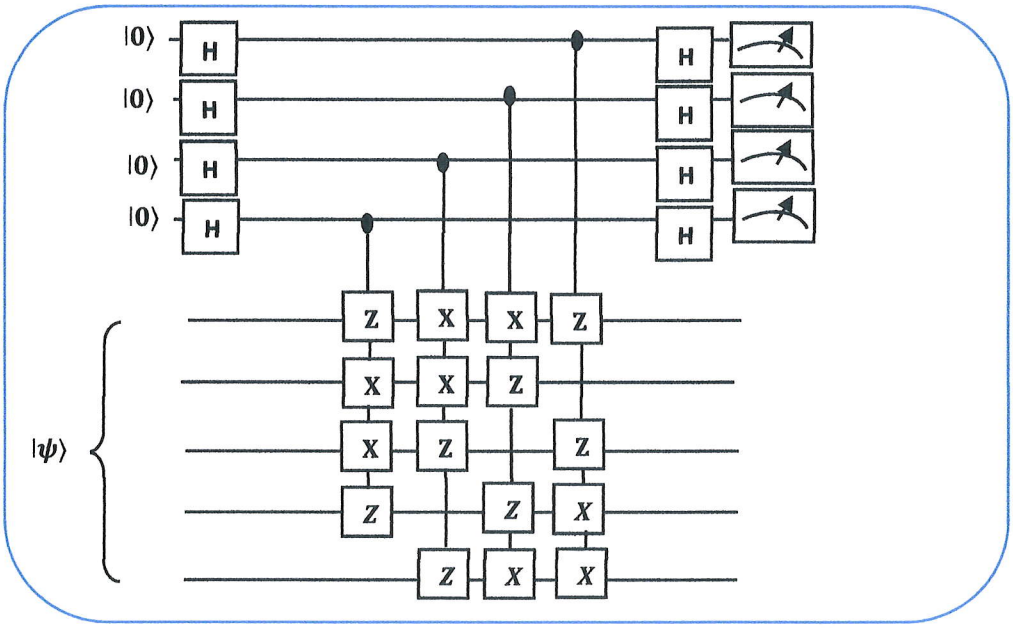


Fig.3.7 : *Circuit quantique pour la détection des syndromes d'erreur.*

Décodage :

Le circuit de décodage est le suivant :

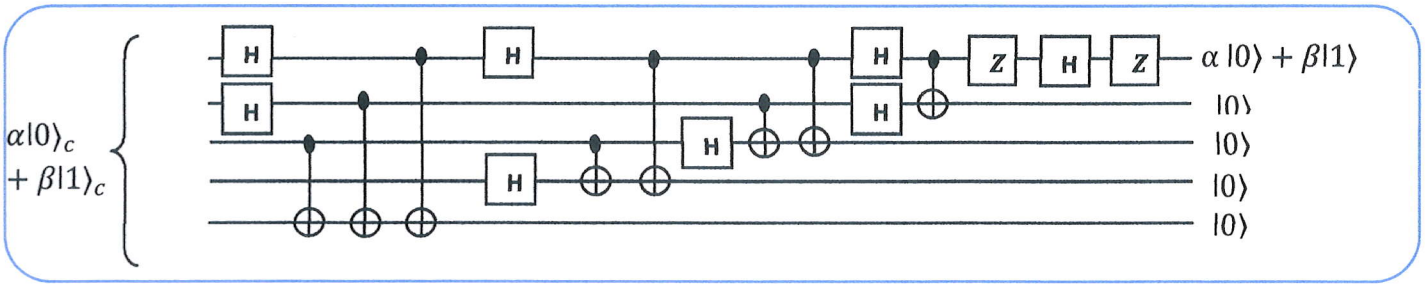


Fig.3.8 **:** *Circuit quantique pour le décodage de 5 Qbits.*

Remarques :

- Le code à cinq Qbits est le code quantique permettant la correction d'erreurs le plus optimal. C'est le code le plus court et donc il est d'un intérêt immense [17,18].
- Il n'y a pas de $M_4 = Z_0X_1X_2Z_3$ car $M_4 = M_0M_1M_2M_3$
- $M_i^2 = I$ pour tous car : $X_k^2 = Z_k^2 = I$.
- $M_iM_j = M_jM_i$ donc tout couple M_i, M_j commute.
- $|0\rangle_c$ et $|1\rangle_c$ sont des vecteurs propres de tous les M_i de valeur propre +1; ceci est dû au fait que tous les M_i commutent et $M_i(1 + M_i) = (1 + M_i)$.

- Il est possible de vérifier qu'en appliquant une erreur X_k (Y_k ou Z_k), les $X_k|\psi\rangle$, $Y_k|\psi\rangle$, et $Z_k|\psi\rangle$ sont également des vecteurs propres de tous les M_i, mais avec des ensembles différents d'états propres .

III.6. Conclusion :

Dans ce chapitre, on a présenté quelques algorithmes de correction d'erreurs basés sur les codes stabilisateurs. Le plus intéressant est celui à 5 Qbits. Il ne nécessite que 4 Qbits supplémentaires et permet la détection et la correction des erreurs de type X, Y et Z. L'idée de base est de trouver un encodage tel que :

- On peut définir des opérateurs de syndrome qui ont la particularité d'avoir comme vecteurs propres les états erronés ainsi que l´état sans erreur.
- La mesure de ces opérateurs ne modifie pas l'état traité.
- Le résultat de la mesure (les valeurs propres) signe sans ambiguïté la position de l'erreur.

Une fois l'erreur détectée, il suffit d'apporter la correction.

Chapitre IV : Implémentation et application

1. Introduction
2. Environnement de développement
3. Simulation quantique
4. Exemples illustratifs

IV.1. Introduction :

Ce chapitre constitue la dernière partie de ce mémoire. C'est une démonstration de la correction d'erreurs basée sur les codes stabilisateurs. Dans un premier temps, nous introduisons les différentes fonctions implémentées pour simuler un calculateur quantique. Ensuite, nous présentons quelques études de cas. Pour plus de clarté, tous les états intermédiaires des registres quantiques manipulés seront affichés.

IV.2. Environnement de développement :

La machine utilisée dans notre implémentation est caractérisée par :

- Processeur : Intel(R) Celeron(R) CPU N2840 @ 2.16GHz.
- Mémoire installée (RAM) : 4,00 GO.
- Disque dur : 500 GO.
- Type du système : système d'exploitation 64 bits.

Environnement logiciel :

Java : le langage java possède de nombreux points forts qui expliquent pourquoi c'est l'un des principaux langages de programmation utilisés en entreprise aujourd'hui :

- ✓ Langage statique et détaillé - Grâce à la nature statique et robuste de Java, il est facile à maintenir et à lire. Java permet de renvoyer plusieurs types de données et on peut facilement les utiliser dans diverses applications de même domaine.
- ✓ Grande compatibilité avec les outils open source, des applications, des serveurs, etc. (Apache)
- ✓ Portabilité, Facile à exécuter, facile à écrire - une fois un code est écrit en java, il est possible de l'exécuter presque n'importe où et à tout moment. on peut l'écrire sous Windows et l'exécuter sous Linux, sur des serveurs / téléphones mobiles, etc. avec le même comportement. C'est la force de la pierre angulaire de Java.

JDK : nécessaire pour le développement et l'implémentation de notre application. Java Development Kit, est le kit de développement qui offre un ensemble d'outils permettant de développer des logiciels et applications java et qui est proposé gratuitement par Oracle.
Netbeans: Un environnement de développement intégré avec un ensemble d'outils, venant aider les programmeurs, spécialisé dans le domaine Java [19].
JavaFX : Une API dédiée à la création d'interfaces graphique. JavaFX est qualifié comme le remplaçant de Swing, développée par Sun Microsystems et racheté par Oracle.

IV.3. Simulation quantique :

Pour simuler le fonctionnement des codes stabilisateurs, une version simplifiée d'un simulateur quantique est développée, avec les principales fonctionnalités requises. Les classes principales sont les suivantes :

- **Etat_Base :** Cette classe concerne la création et la manipulation des états élémentaires. Chaque objet de cette classe se caractérise par les attributs suivants :
  - Vecteur : chaine de caractère.
  - Param : le coefficient de cet état élémentaire, représenté comme un réel.

  Les méthodes de cette classe sont :

  - Constructeur (Param , Vect).
  - Afficher ( ).

- **Etat_Quantique :** Cette classe sert à la manipulation des états quantique. Il s'agit d'un vecteur contenant la combinaison linéaire des états de la base élémentaires. Pour des fins d'optimisation, les éléments avec des coefficients nuls sont ignorés. Cette classe n'a pas d'attributs mais elle est munie des méthodes suivantes :
  - Constructeur (alpha , Beta).
  - ProduitTensoriel (Etat_Quant_01 , Etat_Quant_02).
  - X (PositionQbitCible).
  - Y (PositionQbitCible).
  - Z (PositionQbitCible).
  - H (PositionQbitCible).
  - CNot (PositionQbitControle , PositionQbitCible).
  - CZ (PositionQbitControle , PositionQbitCible).
  - Mesure(PositionQbitCible).
  - Afficher ( ).

  Ces méthodes sont implémentées en se basant sur les pseudo-codes qui suivent :

```
ProduitTensoriel ( EtatQ1 : EtatQuantique, EtatQt2 : EtatQuantique ) : Etat_Quantique

Début :
    Pour ( i=0 ; i < Taille (EtatQ1 ) ) faire :
        Pour ( j = 0 ; j < Taille ( EtatQ2 ) ) faire :
            Param_R = (EtatQ1 [ i ]. param) * (EtatQ [ j ]. param) ;
            Vect_R = (EtatQ1 [ i ]. vect) + (EtatQ2 [ j ]. vect) ;
            Etat_Base_R= Etat_Base (Param_R, Vect_R) ;
            EtatQ_Resultat .ajouter (Etat_Base_R) ;
        Fin Pour.
    Fin Pour.
    Return ( EtatQ_Resultat) ;
Fin.
```

```
OpérateurX (EtatQ : EtatQuantique , PositionQbitCible :entier ) : EtatQuantique

Début :
    Pos_QbitCible = Pos_QbitCible – 1 ;
    Pour ( i = 0 ; i < Taille (EtatQ)) faire :
        vectCourant = Etat_Quant [i] . vect ;
        QbitCible = vect_courant [Pos_QbitCible] ;
        Si (QbitCible = '0') alors :
            nouv_vect = vect_courant [0, Pos_QbitCible];
            nouv_vect = nouv_vect + '1' ;
            nouv_vect = nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ];
        Sinon :
            nouv_vect = vect_courant [0, Pos_QbitCible];
            nouv_vect = nouv_vect + '0' ;
            nouv_vect = nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ];
        Fin Si.
        Nouv_EtatBase = Etat_Base (Etat_Quant [ i ] .param , nouv_vect ) ;
        EtatQ_Resultat. Ajouter (Nouv_EtatBase);
    Fin Pour.
    Return ( EtatQ_Resultat) ;
Fin.
```

```
OpérateurZ (EtatQ : EtatQuantique , PositionQbitCible : entier ) : EtatQuantique

Début :
    Pos_QbitCible = Pos_QbitCible – 1 ;
    Pour ( i = 0 ; i <  Taille (EtatQ)) faire :
        VecteurCourant  =  Etat_Quant [i] . vect ;
        NouveauxVecteur  =  VecteurCourant ;
        QbitCible  =  VecteurCourant[PositionQbitCible] ;
        Si (QbitCible  =  '0') alors :
            Nouveaux Paramaitre  =  EtatQ [ i ] . param ;
        Sinon :
            Nouveaux Paramaitre  =  -  EtatQ [ i ] . param ;
        Fin Si.
        NouveauxEtatBase  =  Etat_Base (Nouveaux Paramaitre  ,  NouveauxVecteur ) ;
        EtatQ_Resultat. Ajouter (NouveauxEtatBase);
    Fin Pour.
    Return ( EtatQ_Resultat)  ;
Fin.
```

```
OpérateurCNOT (EtatQ : EtatQuantique , PositionCible : entier , PositionControl : entier) : EtatQuantique

Début :
    PositionCible  =  PositionCible  –  1 ;
    PositionControl  =  PositionControl  –  1 ;
    Pour ( i = 0 ; i <  Taille (EtatQ)) faire :
        Nouveaux_param  =  EtatQ [ i ] . param ;
        VecteurCourant  =  EtatQ [i] . vect ;
        QbitControl  =  VecteurCourant [PositionControl] ;
        Si (QbitControl  =  '0') alors :
            NouveauxVecteur  =  VecteurCourant ;
        Sinon :
            Qbit_cible= vect_courant [Pos_QbitCible] ;
            Si (Qbit_cible= '0') alors :
                nouv_vect = vect_courant [0, Pos_QbitCible];
                nouv_vect = nouv_vect + '1' ;
                nouv_vect =  nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ] ;
```

```
        Sinon
            nouv_vect = vect_courant [0, Pos_QbitCible];
            nouv_vect = nouv_vect + '1' ;
            nouv_vect =  nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ];
        Fin Si.
    Fin Si.
    Nouv_EtatBase  =  Etat_Base (nouv_par, nouv_vect) ;
    EtatQ_Resultat. Ajouter (Nouv_EtatBase);
  Fin Pour.
  Return ( EtatQ_Resultat)  ;
Fin.
```

OpérateurCZ (EtatQ : EtatQuantique , PositionCible : entier , PositionControl : entier) :

```
Début :
  Pos_QbitCible = Pos_QbitCible – 1 ;
  Pour ( i = 0  ;  i <  Taille (EtatQ)) faire :
    VecteurCourant  =  Etat_Quant [i] . vect ;
    NouveauxVecteur  =  VecteurCourant ;
    QbitCible  =  VecteurCourant[PositionCible] ;
    QbitControl  =  VecteurCourant[PositionControl] ;
    Si (QbitCible  =  '1') et (QbitCible  =  '1')  alors :
        NouveauxParamaitre  =  -  EtatQ [ i ] . param ;
    Sinon :
        Nouveaux Paramaitre  =  EtatQ [ i ] . param ;
    Fin Si.
    NouveauxEtatBase  =  Etat_Base (Nouveaux Paramaitre  ,  NouveauxVecteur ) ;
    EtatQ_Resultat. Ajouter (NouveauxEtatBase);
  Fin Pour.
  Return ( Etat_Qt_Resultat)  ;
Fin.
```

```
OpérateurH (EtatQ : EtatQuantique , PositionCible : entier) : EtatQuantique

Début :
    PositionCible = PositionCible – 1 ;
    Pour ( i = 0 ; i < Taille (EtatQ)) faire :
        vect_courant = Etat_Quant [ i ] . vect ;
        nouv_vect = vect_courant [0, Pos_QbitCible];
        nouv_vect_0 = nouv_vect + '0' ;
        nouv_vect_1 = nouv_vect + '1' ;
        nouv_vect_0 = nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ];
        nouv_vect_1 = nouv_vect + vect_courant [ Pos_QbitCible + 1 , Taille (vect_courant) ];
        Qbit_cible = vect_courant [Pos_QbitCible] ;
        Si (QbitCible = '0') alors :
            Nouv_EtatBase_0 = Etat_Base (Etat_Quant [ i ] . par / sqrt(2), nouv_vect_0) ;
            Nouv_EtatBase_1 = Etat_Base (Etat_Quant [ i ] . par / sqrt(2), nouv_vect_1) ;
        Sinon :
            Nouv_EtatBase_0 = Etat_Base (Etat_Quant [ i ] . par / sqrt(2), nouv_vect_0) ;
            Nouv_EtatBase_1 = Etat_Base ( - Etat_Quant [ i ] . par / sqrt(2), nouv_vect_1) ;
        Fin Si.
        Etat_Qt_Resultat. Ajouter ( Nouv_EtatBase_0 );
        EtatQ_Resultat. Ajouter ( Nouv_EtatBase_1 );
    Fin Pour.
    Return ( EtatQ_Resultat) ;
Fin.
```

- **Interface graphique :**

  Afin d'illustrer notre implémentation, on a proposé l'interface graphique suivante :

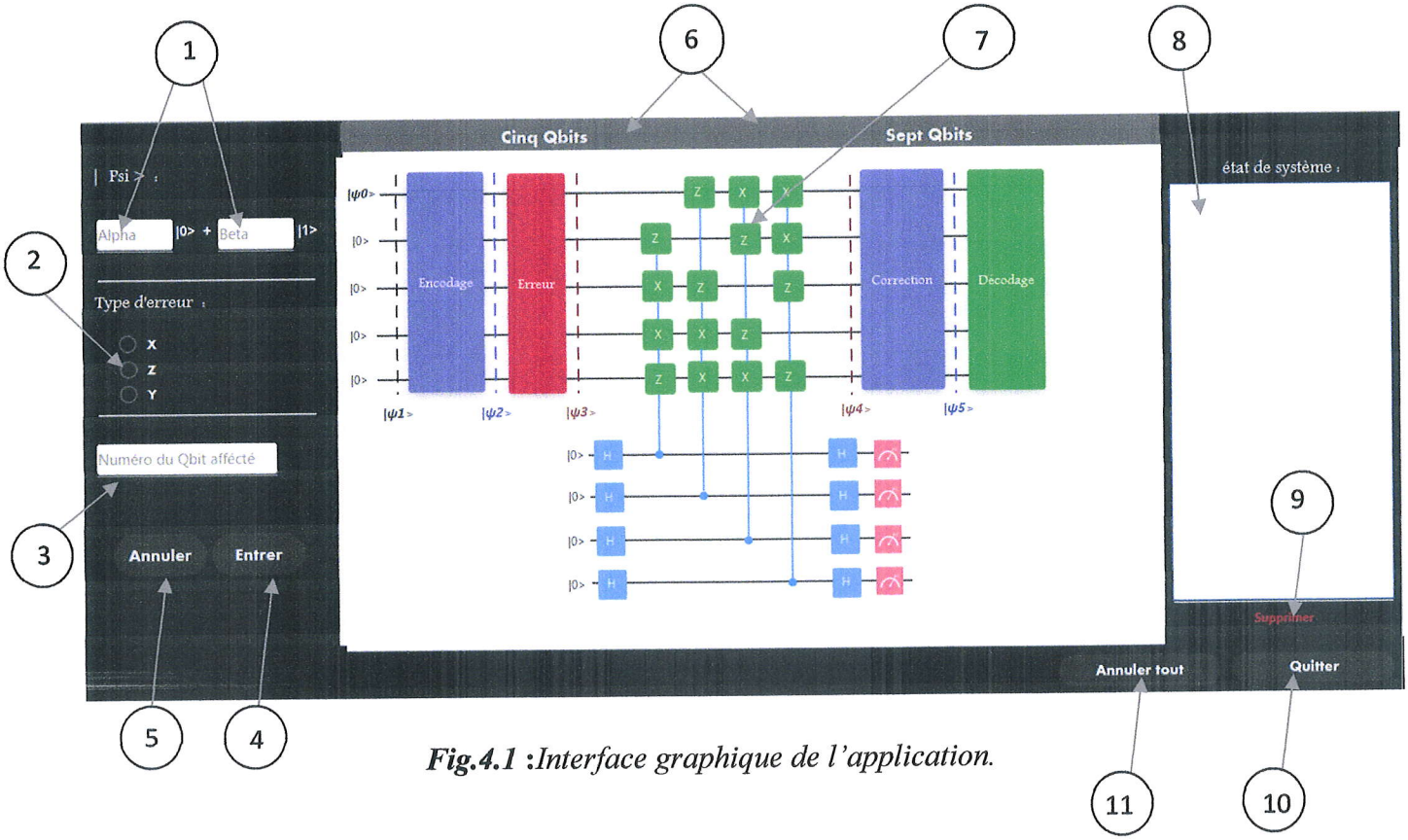


***Fig.4.1** :Interface graphique de l'application.*

1) Champs utilisés pour récupérer les paramètres alpha et beta du Qbit manipulé.
2) Radio Boutons permettant de choisir le type d'erreur traitée : X, Z ou Y.
3) Champs utilisé pour saisir le numéro du Qbit affecté par l'erreur.
4) Bouton pour initialiser tous les paramètres de champs de saisie.
5) Bouton pour réinitialiser les paramètres des champs de saisie.
6) Boutons pour sélectionner la variante étudiée.
7) Circuit quantique.
8) Zone d'affichage des résultats.
9) Boutons pour effacer le contenu de la zone d'affichage.
10) Bouton pour quitter l'application.
11) Bouton pour arrêter le processus de circuit quantique.

IV.4. Exemples illustratifs : Soit le Qbit Psi avec un état initial comme suit :

$$Psi = alpha\ |\ 0 > + beta\ |\ 1 >$$

Tel que : alpha = 0.35 ; beta = 0.93675 ;

❖ Application du Code stabilisateur à cinq Qbits :

✓ Création du Qbit à l'état initial:

$$|\psi\rangle = 0.35|0> + 0.93675|1>$$

✓ Ajout des quatre Qbits d'intrication:

$$|\psi_1\rangle = 0{,}35|00000> + 0.93675|10000>$$

✓ Etape d'encodage:

- Application de la porte Z sur le premier Qbit :

$|\psi_2\rangle = 0{,}35|00000> - 0.93675|10000>$

- Application de la porte H sur le premier Qbit :

$|\psi_2\rangle = - 0.415|00000> + 0.909|10000>$

- Application de la porte Z sur le premier Qbit :

$|\psi_2\rangle = - 0.415|00000> - 0.909|10000>$

- Application de la porte CNot (1, 2):

$|\psi_2\rangle = - 0.415|00000> - 0.909|11000>$

- Application de la porte H sur le premier Qbit :

$|\psi_2\rangle = - 0.293|00000> - 0.293|10000> - 0.642|01000> + 0.\ 642|11000>$

- Application de la porte H sur le deuxième Qbit :

$|\psi_2\rangle = - 0.663|00000> + 0.249|01000> + 0.249|10000> - 0.663|11000>$

- Application de la porte CNot (1, 3):

$|\psi_2\rangle = - 0.663|00000> + 0.249|01000> + 0.249|10100> - 0.663|11100>$

- Application de la porte CNot (2, 3):

$|\psi_2\rangle = - 0.663\ |00000> + 0.249|01100> + 0.249|10100> - 0.663|11000>$

- Application de la porte H sur le troisième Qbit :

$|\psi_2\rangle = - 0.468|00000> + 0.468|00100> + 0.176|01000> - 0.176|01100> + 0.176|10000> - 0.176|10100> - 0.468|11000> - 0.468|11100>$

- Application de la porte CNot (1, 4):

$|\psi_2\rangle$ = - 0.468|00000> + 0.468|00100> + 0.176|01000> - 0.176|01100> + 0.176|10010> - 0.176|10110> - 0.468|11010> - 0.468|11110>

- Application de la porte CNot (3, 4):

$|\psi_2\rangle$ = -0.468 |00000> +0. 468|00110> + 0.176|01000> -0. 176|01110> +0.176|10010> - 0.176|10100> - 0.468|11010> - 0.468|11100>

- Application de la porte H sur le premier Qbit :

$|\psi_2\rangle$ = - 0.33|00000> - 0.33|10000> + 0. 33|00110> + 0. 33|10110> + 0.124|01000> + 0.124|11000> - 0.124|01110> - 0.124|11110> + 0.124|00010> - 0.124|10010> - 0.124|00100> + 0.124|10100> - 0.33|01010> + 0.33|11010> - 0.33|01100> + 0.33|11100>

- Application de la porte H sur le quatrième Qbit :

$|\psi_2\rangle$ = - 0.146|00000> - 0.32 |00010> - 0. 32 |10000> - 0. 146|10010> + 0.146|00100> - 0.32|00110> + 0.32|10100> - 0.146|10110> - 0.146|01000> + 0.32|01010> + 0.32|11000> - 0.146|11010> - 0.32|01100> + 0.087|01110> + 0.146|11100> + 0.32|11110>

- Application de la porte CNot (1, 5):

$|\psi_2\rangle$ = - 0.146|00000> - 0.32|00010> - 0.32 |10001> - 0.146|10011> + 0.146|00100> - 0.32|00110> + 0.32|10101> - 0.146|10111> - 0.146|01000> + 0.32|01010> + 0.32|11001> - 0.146|11011> - 0.32|01100> + 0.087|01110> + 0.146|11101> + 0.32|11111>

- Application de la porte CNot (2, 5):

$|\psi_2\rangle$ = - 0.146|00000> - 0.32|00010> - 0.32|10001> - 0.146|10011> + 0.146|00100> - 0.32|00110> + 0.32|10101> - 0.146|10111> - 0.146|01001> + 0.32|01011> + 0.32|11000> - 0.146|11010> - 0.32|01101> + 0.087|01111> + 0.146|11100> + 0.32|11110>

- Application de la porte CNot (3, 5):

$|\psi_2\rangle$ = - 0.146|00000> - 0.32|00010> - 0.32|10001> - 0.146|10011> + 0.146|00101> -0. 32|00111> + 0.32|10100> - 0.146|10110> - 0.146|01001> + 0. 32|01011> + 0.32|11000> - 0.146|11010> - 0.32|01100> + 0.087|01110> + 0.146|11101> + 0.32|11111>

- Application de la porte H sur le premier Qbit :

$|\psi_2\rangle$ = - 0.10|00000> - 0.10|10000> - 0.22|00010> - 0.22|10010> - 0. 22|00001> + 0.22|10001> - 0.10|00011> + 0.10|10011> + 0.10|00101> + 0.10|10101> - 0.22|00111> - 0.22|10111> + 0. 22|00100> - 0. 22|10100> - 0.10 |00110> + 0.10|10110> - 0.10|01001> - 0.10|11001> + 0.22|01011> + 0.22|01000> - 0.22|11000> - 0.10|01010> + 0.10|11010> - 0.22|01100> - 0.22|11100> + 0.061|01110> + 0.061|11110> + 0.10 |01101> - 0.10|11101> + 0.22|01111> - 0.22|11111>

- Application de la porte H sur le deuxième Qbit, on obtient le psi codé comme suit:

$|\psi_2\rangle$ = - 0.085|00000> - 0.23|01000> - 0.23|10000> + 0.085|11000> - 0.23|00010> - 0.085|01010> - 0.085|10010> - 0.23|11010> + 0.085|01100> - 0.085|01001> + 0.085|10001> - 0.085 |00011> - 0.23|01011> + 0.23 |10011> - 0.085|11011> - 0.085|00101> - 0.23|01101> - 0.23|10101> - 0.085|11101> - 0.23|00111> - 0.085|01111> - 0.085|10111> + 0.23|11111> - 0.23|00100> - 0.085|10100> + 0.085|00110> + 0.23|01110> - 0.23|10110> - 0.085|11110> - 0.23|00001> + 0.23|11001> + 0.23|01110>

- **Correction d'une erreurs de type X (Bit-flip)**

✓ Injecter une Erreur X sur le deuxième Qubit :

$|\psi_3\rangle$ = - 0.085|01000> - 0.23|00000> - 0.23|11000> + 0.085|10000> - 0.23|01010> - 0.085|00010> - 0.085|11010> - 0.23|10010> + 0.085|00100> - 0.085|00001> + 0.085|11001> - 0.085|01011> - 0.23|00011> + 0.23|11011> - 0.085|10011> - 0.085|01101> - 0.23|00101> - 0.23|11101> - 0.085 |10101> - 0.23|01111> - 0.085|00111> - 0.085|11111> + 0.23|10111> - 0.23|01100> - 0.085|11100> + 0.085|01110> + 0.23|00110> - 0.23|11110> - 0.085|10110> - 0.23|01001> + 0.23|10001> + 0.23|00110>

✓ Ajout des quatre Qbits du syndrome:

$|\psi_3\rangle$ = - 0.085|010000000> - 0.23|000000000> - 0.23|110000000> + 0.085|100000000> - 0.23|010100000> - 0.085|000100000 > - 0.085|110100000> - 0.23|100100000> + 0.085|001000000> - 0.085|000010000> + 0.085|110010000> - 0.085|010110000> - 0.23|000110000> + 0.23|110110000> - 0.085|100110000> - 0.085|011010000> - 0.23|001010000> - 0.23|11101 0000> - 0.085|101010000> - 0.23|011110000> - 0.085|001110000> - 0.085|111110000> + 0.23|101110000> - 0.23|011000000> - 0.085|111000000> + 0.085|011100000 > + 0.23|00110 0000> - 0.23|111100000> - 0.085|101100000> - 0.23|010010000> + 0.23|100010000> + 0.23|001100000>

✓ Après application du circuit, on obtient :

$|\psi_3\rangle$ = 0.075|001001**010**> + 0.23|001101**010**> + 0.075|010001**010**> + 0.075|010111**010**> + 0.075|011101**010**> + 0.23|011111**010**> + 0.075|100001**010**> + 0.23|100011**010**> + 0.23|101001**010**> + 0.23|101111**010**> + 0.075|110011**010**> + 0.23|110111**010**>

Résultat de mesure = 1010 , donc erreur de type X2.

✓ La correction : appliquer la porte X sur le deuxième Qbit :

$|\psi_4\rangle$ = 0.075|001001010> + 0.23|001101010> + 0.075|010001010> + 0.075|010111010> + 0.075|011101010> + 0.23|011111010> + 0.075|100001010> + 0.23|100011010> + 0.23|101001010> + 0.23|101111010> + 0.075|110011010> + 0.23|110111010>

✓ Suppression des quatre Qbits du syndrome:

$|\psi_4\rangle$ = 0.075|00100> + 0.23|00110> + 0.075|01000> + 0.075|01011> + 0.075|01110> + 0.23|01111> + 0.075|10000> + 0.23|10001> + 0.23|10100> + 0.23|10111> + 0.075|11001> + 0.23|11011>

✓ Etape de décodage :

- Application de circuit de décodage :

$|\psi_5\rangle$ = 0.349999|00000> + 0.936749|10000>

- Suppression des Qbits auxiliaires :

$|\psi\rangle$ = 0.349999|0> + 0.936749|1>

• Correction d'erreurs de type Z (Phase-flip)

✓ Injecter une Erreur Z sur le premier Qubit :

$|\psi_3\rangle$ = - 0.085|01000> - 0.23|00000> + 0.23|11000> - 0.085|10000> - 0.23|01010> - 0.085|00010> + 0.085|11010> + 0.23|10010> + 0.085|00100> - 0.085|00001> - 0.085|11001> - 0.085|01011> - 0.23|00011> - 0.23|11011> - 0.085|10011> - 0.085|01101> - 0.23|00101> + 0.23|11101> + 0.085|10101>- 0.23|01111> - 0.085|00111> + 0.085|11111> - 0.23|10111> - 0.23|01100> + 0.085|11100 > + 0.085|01110> + 0.23|00110> + 0.23|11110> + 0.085|10110> - 0.23|01001> - 0.23|10001> + 0.23|00110>

✓ Ajout des quatre Qbits du syndrome:

$|\psi_3\rangle$ = - 0.085|010000000> - 0.23|000000000> + 0.23|110000000> - 0.085|100000000> - 0.23|010100000> - 0.085|000100000> + 0.085|110100000> + 0.23|100100000> + 0.085|001000000> - 0.085|000010000> - 0.085|110010000> - 0.085|010110000> - 0.23|000110000> - 0.23|110110000> - 0.085|100110000> - 0.085|011010000> - 0.23|001010000> + 0.23|111010000> + 0.085|101010000> - 0.23|011110000> - 0.085|001110000> + 0.085|111110000> - 0.23|101110000> - 0.23|011000000> + 0.085|111000000> + 0.085|011100000> + 0.23|001100000> + 0.23|111100000> + 0.085|101100000> - 0.23|010010000> - 0.23|100010000> + 0.23|00110 0000 >

✓ Après application du circuit, on obtient :

$|\psi_4\rangle$ = 0.074|000000**0011**> - 0.23|000010**0011**> - 0.23|000100**0011**> + 0.074|000110**0011**>
- 0.23|001000**0011**> - 0.074|001010**0011**> + 0.074|001100**0011**> + 0.23|001110**0011**>
- 0.23|010000**0011**> - 0.074|010010**0011**> - 0.074|010100**0011**> - 0.23|010110**0011**>
+ 0.074|011000**0011**> - 0.23|011010**0011**> + 0.23|011100**0011**> - 0.074|011110**0011**>
+ 0.23|100000**0011**> - 0.074|100010**0011**> + 0.074|100100**0011**> - 0.23|100110**0011**>
+ 0.074|101000**0011**> + 0.23|101010**0011**> + 0.23|101100**0011**> + 0.074|101110**0011**>
- 0.074|110000**0011**> - 0.23|110010**0011**> + 0.23|110100**0011**> + 0.074|110110**0011**>
- 0.23|111000**0011**> + 0.074|111010**0011**> + 0.074|111100**0011**> - 0.23|111110**0011**>

Résultat de mesure = 0011, donc erreur de type Z1

✓ **La correction** : appliquer la porte Z sur le premier Qbit :

$|\psi_5\rangle$ = 0.074|000000011> - 0.23|000010011> - 0.23|000100011> + 0.074|000110011> - 0.23|001000011> - 0.074|001010011> + 0.074|001100011> + 0.23|001110011> - 0.23|010000011> - 0.074|010010011> - 0.074|010100011> -0.23|010110011> + 0.074|011000011> - 0.23|011010011> + 0.23|011100011> -0.074|011110011> - 0.23|100000011> + 0.074|100010011> - 0.074|100100011> + 0.23|100110011> - 0.074|101000011> - 0.23|101010011> - 0.23|101100011> -0.074|101110011> + 0.074|110000011> + 0.23|110010011> - 0.23|110100011> -0.074|110110011> + 0.23|111000011> - 0.074|111010011> -0.074|111100011> + 0.23|111110011>

✓ Suppression des quatre Qbits du syndrome:

$|\psi_5\rangle$ = 0.074|00000> - 0.23|00001> - 0.23|00010> + 0.074|00011> - 0.23|00100> - 0.074|00101> + 0.074|00110> + 0.23|00111> - 0.23|01000> - 0.074|01001> - 0.074|01010> - 0.23|01011> + 0.074|01100> - 0.23|01101> + 0.23|01110> - 0.074|01111> - 0.23|10000> + 0.074|10001> - 0.074|10010> + 0.23|10011> -0.074|10100> - 0.23|10101> - 0.23|10110> - 0.074|10111> + 0.074|11000> + 0.23|11001> - 0.23|11010> - 0.074|11011> + 0.23|11100> - 0.074|11101> -0.074|11110> + 0.23|11111>

✓ Etape de décodage :

- Application de circuit de décodage :

|Psi⟩ = 0.349999|00000> + 0.936749|10000>

- Suppression des Qbits auxiliaires :

|Psi⟩ = 0.349999|0> + 0.936749|1>

- **Correction d'erreurs de type Y :**

✓ Injecter une Erreur Y sur le cinquième Qubit :

$|\psi_3\rangle$ = 0.085|00001> + 0.23|01001> + 0.23|10001> - 0.085|11001> + 0.23|00011> + 0.085|01011> + 0.085|10011> + 0.23|11011> - 0.085|01101> - 0.085|01000> + 0.085|10000> - 0.085|00010> - 0.23|01010> + 0.23|10010> - 0.085|11010> - 0.085|00100 > - 0.23|01100> - 0.23|10100> - 0.085|11100> - 0.23|00110> - 0.085|01110> - 0.085|10110> + 0.23|11110> + 0.23|00101> + 0.085|10101> -0.085|00111 > - 0. 23|01111> + 0.23|10111> + 0.085|11111> - 0.23|00000> + 0. 23|11000> + 0.23|01111>

✓ Ajout des quatre Qbits du syndrome:

$|\psi_3\rangle$ = 0.085|000010000> + 0.23|010010000> + 0.23|100010000> - 0.085|110010000> + 0.23|000110000> + 0.085|010110000> + 0.085|100110000> + 0.23|110110000> - 0.085|011010000> - 0.085|010000000> + 0.085|100000000> - 0.085|000100000> - 0.23|010100000> + 0.23|100100000> - 0.085|110100000> - 0.085|001000000> - 0.23|011000000> - 0.23|101000000> - 0.085|111000000> - 0.23|001100000> - 0.085|011100000> - 0.085|101100000> + 0.23|111100000> + 0.23|001010000> + 0.085|101010000> - 0.085|001110000> - 0.23|011110000> + 0.23|101110000> + 0.085|111110000> - 0.23|000000000> + 0.23|110000000> + 0.23|011110000>

✓ Après application du circuit, on obtient :

$|\psi_4\rangle$ = -0.2374|000001**1111**> - 0.0749|000011**1111**> + 0.0749|000101**1111**> + 0.2374|000111**1111**> - 0.0749|001001**1111**> + 0.2374|001011**1111**> + 0.2374|001101**1111**> - 0.0749|001111**1111**> - 0.0749|010001**1111**> + 0.2374|010011**1111**> -0.2374|010101**1111**> + 0.0749|010111**1111**> - 0.2374|011001**1111**> - 0.0749|011011**1111**> - 0.0749|011101**1111**> - 0.2374|011111**1111**> + 0.0749|100001**1111**> + 0.2374|100011**1111**> + 0.2374|100101**1111**> + 0.0749|100111**1111**> - 0.2374|101001**1111**> + 0.0749|101011**1111**> - 0.0749|101101**1111**> + 0.2374|101111**1111**> + 0.2374|110001**1111**> - 0.0749|110011**1111**> - 0.0749|110101**1111**> + 0.2374|110111**1111**> - 0.0749|111001**1111**> - 0.2374|111011**1111**> + 0.2374|111101**1111**> + 0.0749|111111**1111**>

Résultat de mesure = 1111 **, donc erreur de type Y5**

✓ **La correction** :

- appliquer la porte Y sur le cinquième Qbit :

$|\psi_5\rangle$ = - 0.2374|000011111> + 0.0749|000001111> + 0.0749|000111111> - 0.2374|000101111> - 0.0749|001011111> - 0.2374|001001111> + 0.2374|001111111> + 0.0749|001101111> - 0.0749|010011111> - 0.2374|010001111> - 0.2374|010111111> -

0.0749|010101111> - 0.2374|011011111> + 0.0749|011001111> - 0.0749|011111111> + 0.2374|011101111> + 0.0749|100011111> - 0.2374|100001111> + 0.2374|100111111> - 0.0749|100101111> - 0.2374|101011111> - 0.0749|101001111> - 0.0749|101111111> - 0.2374|101101111> + 0.2374|110011111> + 0.0749|110001111> - 0.0749|110111111> - 0.2374|110101111> - 0.0749|111011111> + 0.2374|111001111> + 0.2374|111111111> - 0.0749|111101111>

✓ Suppression des quatre Qbits du syndrome:

$|\psi_5\rangle$ = - 0.2374|00001> + 0.0749|00000> + 0.0749|00011> - 0.2374|00010> - 0.0749|00101> - 0.2374|00100> + 0.2374|00111> + 0.0749|00110> - 0.0749|01001> - 0.2374|01000> - 0.2374|01011> - 0.0749|01010> - 0.2374|01101> + 0.0749|01100> - 0.0749|01111> + 0.2374|01110> + 0.0749|10001> - 0.2374|10000> + 0.2374|10011> - 0.0749|10010> - 0.2374|10101> - 0.0749|10100> - 0.0749|10111> - 0.2374|10110> + 0.2374|11001> + 0.0749|11000> - 0.0749|11011> - 0.2374|11010> -0.0749|11101> + 0.2374|11100> + 0.2374|11111> - 0.0749|11110>

✓ Application de circuit de décodage :

|Psi⟩ = 0.349999|00000> + 0.936749|10000>

✓ Suppression des Qbits auxiliaires :

|Psi⟩ = 0.349999|0> + 0.936749|1>

✓ Etape de décodage :

- Application de circuit de décodage :

|Psi⟩ = 0.349999|00000> + 0.936749|10000>

- Suppression des Qbits auxiliaires :

|Psi⟩ = 0.349999|0> + 0.936749|1>

Les énormes difficultés qu'il y a à préserver les Qbits des influences du milieu environnant représentent l'entrave majeure à la réalisation d'un prototype d'ordinateur quantique. Le problème est si aigu que la solution n'est nullement écartée de l'adoption d'un protocole massif de corrections d'erreurs.

La correction automatique d'erreurs quantiques se fait sur le même principe qu'en théorie classique de l'information en recourant à des Qbits supplémentaires qui créent la redondance nécessaire. Mais plusieurs particularités du cas quantique rendent impossible l'adaptation directe des stratégies classiques. Le problème se complique du fait que :

- La structure du Qbit est différente de celle du bit classique.
- Certaines opérations classiques sont impossibles à réaliser de manière quantique, par exemple le clonage pur et simple est impossible.
- La mesure détruit l'état quantique. En plus, c'est une opération irréversible.
- Les erreurs quantiques possibles sont plus nombreuses que celles classiques.

Dans ce travail, nous nous sommes intéressés à la solution basée sur les codes stabilisateurs. Les stabilisateurs sont les analogues des lignes de la matrice de parité. Chaque équation aux valeurs propres satisfaite par un état du code est l'analogue d'une contrainte de parité. Les valeurs propres des stabilisateurs sont l'analogue du syndrome. Dans un premier temps, nous avons présenté une étude de cette solution. Ensuite, on a proposé une implémentation avec des exemples d'application. Cet algorithme permet la correction des erreurs de types X, Y et Z. Comme ces trois opérateurs, dits groupe de Pauli, constituent une base de l'espace des matrices unitaires, n'importe quelle anomalie est donc corrigeable. Cet algorithme est très efficace dans la correction. En termes de performance, n'utilise que quatre Qbits supplémentaires dans la phase d'encodage.

Ce travail peut être étendu selon plusieurs directions. Une des plus prometteuses, la recherche des codes stabilisateurs pour un ensemble de Qbits. Le traitement collectif peut diminuer les Qbits supplémentaires nécessaires.

Bibliographie

[1] S. Perdrix, " Modèles formels du calcul quantique: ressources, machines abstraites et calcul par mesure ", Institut National Polytechnique de Grenoble, Décembre 2006.

[2] H. Talbi, "Algorithmes évolutionnaires quantiques pour le recalage et la segmentation multi-objectif d'images ", Thèse de doctorat en sciences, Université Mentouri Constantine, 2009.

[3] J.Gruska, "Quantum computing ", McGraw-Hill, 1999.

[4] E.Strubell, "An Introduction to Quantum Algorithms", COS498 – Chawathe, 2011.

[5] P. Jorrand, " A Programmer's Survey of the Quantum Computing Paradigm".

[6] B. Lucas, H. Jaffali, I. Nounouh, " Principes fondamentaux de l'information quantique", Université Technologique de Belfort Montbéliard, 2014.

[7] S.MAGNIN-FEYSOT, " Cryptographie : De RSA à l'algorithme de Shor ", 8 janvier 2014.

[8] X. Lacour, "Information Quantique par Passage Adiabatique: Portes Quantiques et décohérence", Université de Bourgogne, 03 octobre 2007.

[9] Dr. Ramazan KOC, "chapter4 Quantum circuits", Université Gaziantep.

[10] D. Mermin," Calcul et algorithmes quantique : méthodes et exemples", CNRS EDITIONS.Nice, février 2010.

[11] G. Brassard "Cryptology column -quantum computing : The end of classical cryptography", ACM SIGACT News 25(4)15-21 décembre 1994.

[12] P. Shor "Algorithms for quantum computation Discrete log and factoring Proceedings of the 35-th Annual Symposium on Foundations of Computer Science".

[13] P. Shor "Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM Journal on Computing 26(5) :1484-1509 octobre 1997.

[14] A. Marin, " Borne inférieure sur la capacité d'un canal quantique ", Rapport de stage, Centre de recherche Inria-Rocquencourt, 2009.

[15] D. Gottesman, "A class of quantum error-correcting codes saturating the quantum Hamming bound", Phys. Rev. A, 1996.

[16] D. Gottesman, "Stabilizer Codes and Quantum Error Correction ", California Institute of Technology, Pasadena, California, 2008.

[17] M. Ibnouhsein," Corrélations quantiques et structures causales", Thèse de Doctorat, Université Paris-Sud, 2014.

[18] Y.Leroyer, G.Sénizergues, "Introduction à l'information quantique" Cours pour ENSEIRB 2ème année, Université de Bordeaux 1 Sciences Technologies, 2011.

[19] https://fr.netbeans.org/edi/articles/concours/presentation-netbeans40-1.html.

Résumé

L'informatique quantique est un sous-domaine de l'informatique basé sur les principes de la mécanique quantique. Dans ce mode de calcul, les erreurs sont principalement dues à l'interaction du système quantique avec son environnement. Plutôt que de tenter d'affronter de face cet obstacle inévitable, il est possible d'utiliser des codes correcteurs.

Le but de ce projet est l'étude et l'implémentation des solutions basées sur les codes stabilisateurs. Elles sont basées principalement sur l'algèbre des opérateurs de Pauli. Pour des fins de validation, plusieurs exemples d'application sont présentés.

Mots clés : Informatique quantique, calcul quantique, codes correcteurs, codes stabilisateurs.

Abstract

Quantum information is a subdomain of computer science based on the principles of quantum mechanics. In this mode of calculation, the errors are mainly due to the interaction of the quantum system with its environment. Rather than trying to face this unavoidable obstacle face to face, it is possible to use error correcting codes.

The purpose of this project is the study and implementation of solutions based on stabilizer codes. They are based mainly on the algebra of Pauli operators. For validation purposes, several application examples are presented.

Key words : Quantum information, Quantum computing, error correcting codes, stabilizer codes.