

Тема по квантова информатика:
Алгоритъм за засичане и коригиране на грешки в квантови
системи
Quantum Error Correction

Цветелин Цецков ФН:82130

10 февруари 2021 г.

Съдържание

| | | |
|----------|-------------------------------------|-----------|
| 1 | Класически подход | 2 |
| 2 | Преход към квантова система | 2 |
| 2.1 | Кодировки на един кубит | 2 |
| 2.2 | За шума | 6 |
| 2.3 | Декодиране на състоянието | 7 |
| 3 | Недостатъци на метода | 10 |
| 4 | Кодировка на Шор | 10 |
| 5 | Заключение | 11 |

Абстракт

При квантовите операции и пренос на състояния винаги е възможно да настъпи т.н. decoherence - от чисти състояния, представени с вектори, до нечисти състояния, представени от матрици на плътността. Или да останем в чисто състояние, но околната среда да "измери" състоянието вместо нас. Тогава спрямо Проекционният постулат системата ще премине със скок в ново състояние спрямо изчислителния базис.

1 Класически подход

Можем да заемем подход от един от първите подходи за корекции на грешки в класическите компютри - да повторим бит няколко пъти¹. Това не е проблем, ако нашето състояние е едно от базисните $|0\rangle, |1\rangle, |+\rangle$ или $|-\rangle$.

Ако кубитът, който кодираме е $|\psi\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$. То нашата кодировка с повторение би изглеждала по следния начин:

$$|\phi\rangle = |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \quad (1)$$

Тази кодировка се основава на факта, че ако p е вероятността за обръщане на един бит при преноса(или изчислението). То вероятността за:

1 грешка е от порядъка на $p \cdot (1-p) \cdot (1-p) \rightarrow p$

2 грешки е от порядъка на $p \cdot p \cdot (1-p) \rightarrow p^2$

3 грешки е от порядъка на $p \cdot p \cdot p \rightarrow p^3$

2 Преход към квантова система

Но обаче ако имаме произволно състояние $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, то за него не бихме могли да приложим (1)²

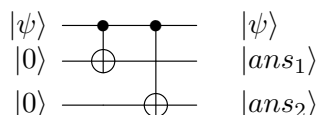
Трябва да се измисли начин, по който да можем да поправяме грешки в кодировката без да се налага да измерваме системата, или поне не директно, защото тогава ще се приложи Проекционният постулат и ще загубим състоянието.

Вместо да правим:

$$|\phi\rangle = |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \quad (2)$$

2.1 Кодировки на един кубит

Една от най-простите кодировки би била да използваме помощни битове, за да запазим състоянието.



Фигура 1: Примерно кодиране на неизвестно състояние $|\psi\rangle$

Ако $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, то след кодировката получаваме:

$$|\phi\rangle = \alpha|000\rangle + \beta|111\rangle \neq |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle \quad (3)$$

Нека за пълнота приемем $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ и изпишем матрицата на кодировката.

¹Класическите [Кодировки с повторение](#). Основен принци, на които е повторението на един и същи бит разчитайки на факта, че повече на брой грешки са по-малко вероятни.

²Защото е в сила [Теоремата](#), че не можем да клонираме произволно състояние

Нека преди това да припомним матриците на трансформациите Както и операциите, които прилагаме към трансформациите, за да можем да ги приложим към нашия "квантов регистър"³

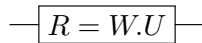
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Матриците на използваните операции в Фигура 1(на нея се вижда само $CNOT$ операцията, но когато изпишем алгоритъма в явен вид ще стане ясно, че се ползва и $SWAP$ операцията)



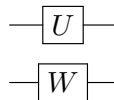
Фигура 2: Последнователно прилагане на операции

Тогава тази операция може да се замени с една единствена получена по правилото: $R = W.U$



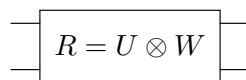
Фигура 3: Последнователно прилагане на операции

Когато обаче се случи да имаме операции, които действат само върху част от квантовия регистър:



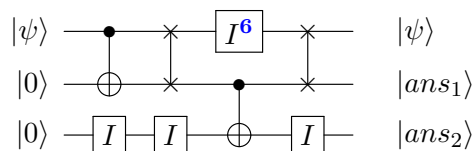
Фигура 4: Паралелно прилагане на операции

Тогава можем да заместим двете с трансформация която действа едновременно и на двата кубита по правилото: $R = U \otimes W$ ⁴



Фигура 5: Паралелно прилагане на операции

Явен вид на кодировката Сега е необходимо да изпишем по-подробно предишната схема от Фигура 1 във Фигура 6.



Фигура 6: Примерно кодиране на неизвестно състояние $|\psi\rangle$

Това ще ни помогне да видим кои операции ще са ни необходими(ако построяваме и физически) и какви операции трябва да приложим между матриците на трансформациите. В случая

³В аналог с класическите компютри този термин ще бъде използван за съставна система от 2 или повече кубита

⁴С този символ означаваме тензорно произведение между 2 матрици. Още наричано [произведение на Кронекер](#)

⁶Използваме I като операцията идентитет - не осъществява никаква трансформация и има матрица $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

множеството $\{I, CNOT, SWAP\}$ е напълно достатъчно, за да построим нашият алгоритъм. Така можем да изведем формула за нашата кодировка⁷:

$$ENCODE_1 = \underbrace{(SWAP \otimes I)}_{\text{Четвърта колона}} \cdot \underbrace{(I \otimes CNOT)}_{\text{Трета колона}} \cdot \underbrace{(SWAP \otimes I)}_{\text{Втора колона}} \cdot \underbrace{(CNOT \otimes I)}_{\text{Първа колона}} \quad (5)$$

(5) представлява формула за изчисление на матрицата на нашата кодировка (за един кубит). Замествайки стойностите от (4) в нея получаваме "кодировъчната матрица":

$$ENCODE_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

От сега нататък $ENCODE_1$ ще бъде използвана като кодировка на 3 кубита по метода обяснен по-горе.

Интересното на тази кодировка е, че посредством тези $CNOT$ операции, цялата система става "entangled" (преплетена). Именно от този феномен идва всеизветната фраза на Айнщайн - "spooky action at a distance" тъй като ако измерим един кубит на системата веднага ще знаем стойността на другите два. [2]

Примерни изчисления с кодировъчната матрица Ще извършим кодировка на всеки от базисните вектори на състоянията и в правоъгълния базис⁸, и диагоналния⁹

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (7)$$

НЕ можем просто така да приложим $ENCODE_1$ върху $|0\rangle$, защото само един кубит, докато $ENCODE_1$ действа върху 3. Затова конструираме състояние $|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle$

$$|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (8)$$

$$ENCODE_1 \cdot |000\rangle = ENCODE_1 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |000\rangle \quad (9)$$

⁷Избраното име на трансформацията идва от англ. "encode кодирам, от там и кодировка. Прието е да се изписват имената на трансформациите на латиница с латински букви, за да има консистентност поне между чертежи, докато текстовете са различни.

⁸Прието е правоъгълният базис да се състои от $\{|1\rangle, |0\rangle\}$

⁹Диагоналният базис се състои от $\{|+\rangle, |-\rangle\}$, който може да се получи от правоъгълния по следния начин $|\pm\rangle = \frac{1}{\sqrt{2}}|0\rangle \pm \frac{1}{\sqrt{2}}|1\rangle$

Разглеждаме случая за $|1\rangle$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (10)$$

Аналогично не можем да приложим кодирувчната матрица, затова конструираме състоянието $|4\rangle^{10} = |100\rangle = |1\rangle \otimes |0\rangle \otimes |0\rangle$

$$|4\rangle = |100\rangle = |1\rangle \otimes |0\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (11)$$

$$ENCODE_1 \cdot |100\rangle = ENCODE_1 \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |111\rangle = |7\rangle \quad (12)$$

Сега за диагоналния базис:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (13)$$

$$|+00\rangle = |+\rangle \otimes |0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} |000\rangle + \frac{1}{\sqrt{2}} |001\rangle \quad (14)$$

$$ENCODE_1 \cdot |+00\rangle = ENCODE_1 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |+00\rangle \quad (15)$$

¹⁰С цел по-кратък запис можем да използваме десетичния запис на квантовия регистър когато това е възможно, напр $|101\rangle = |5\rangle$

И последно случая:

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad (16)$$

$$|-00\rangle = |-\rangle \otimes |0\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (17)$$

$$\begin{aligned} ENCODE_1. |-00\rangle &= ENCODE_1. \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = |-00\rangle \end{aligned} \quad (18)$$

Интересно нещо, което можем да забележим е, че диагоналните базиси и $|0\rangle$ остават "същите докато само $|1\rangle$ променя стойността си(вж. (12))

2.2 За шума

Той може да бъде под най-различна форма и да "разстрои"нашето състояние по много начини. Може да бъде породен от множество фактори. Околната среда да "измери"нашето състояние и чрез проекционния посулат да загубим подготвеното състояние. Или пък от чисто състояние да го превърне в матрица на плътността, с която не е удобно да се работи за изчисления. Дори физическите трансформации, които прилагаме на кубита може да разстроят неговото състояние и да загубим нашите данни.

А относно каква форма могат да приемат тези изменения, те могат да бъдат

1. Обръщане на бит¹¹
2. Обръщане на фаза¹²
3. Линейна комбинация комбинация на 1 и 2

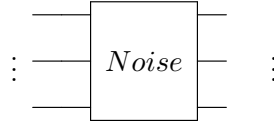
В известен смисъл построения $ENCODE_1$ ни защитава само от 1. Обаче ако се загледаме хубаво в обръщането на фазата можем да забележим, че обръщането на фазата в правоъгълния базис съответства на обръщане на бита в диагоналния, така че можем да продължим да изграждаме

¹¹т.нар. bit-flip от класическите системи, където например дадена система $\alpha_1\alpha_2\alpha_3$ преминава в системата: $\alpha_1\overline{\alpha_2}\alpha_3$

¹²т. нар. phase-flip, когато състоянието $\alpha|0\rangle + \beta|1\rangle$ преминава в $\alpha|0\rangle - \beta|1\rangle$.

алгоритъма за корекция на грешки в правоъгълния базис и в следствие да го допълним и до диагонален базис, за да коригираме грешка и в него.

В крайната диаграма ще използваме следното означение за шума от околната среда и грешката по време на операциите



Фигура 7: Означението, че системата преминава през шум

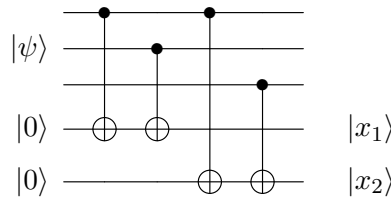
2.3 Декодирание на състоянието

В общия случай получаваме неизвестно състояние

$$|\psi\rangle = \alpha_1 |0\rangle + \alpha_2 |1\rangle + \alpha_3 |2\rangle + \alpha_4 |3\rangle + \alpha_5 |4\rangle + \alpha_6 |5\rangle + \alpha_6 |6\rangle + \alpha_7 |7\rangle \quad (19)$$

След като знаем, че идващото състояние е получено чрез нашата кодировка, можем да сме сигурни, че е разложимо(или поне трябва да бъде). Затова можем да си го мислим като получено от даден кубит $|\phi\rangle = \alpha |0\rangle + \beta |1\rangle$

Но не можем просто да започнем да измерваме, тъй като ще загубим информацията заради проекционния постулат. Затова е необходимо отново да включим т.нар. ancilla bits¹³. Ще ги използваме да засечем грешките, които са възникнали и ще измерваме само тях, без да докосваме състоянието.



Фигура 8: Декодирание на полученото състояние

Сега можем да разсъждаваме за $|x_1\rangle$ и $|x_2\rangle$ ¹⁴.

$$|x_1, x_2\rangle = \begin{cases} |00\rangle \rightarrow \text{Не правим нищо.} \\ |01\rangle \rightarrow \text{Прилагаме Pauli-X}^{15} \text{ на третия кубит.} \\ |10\rangle \rightarrow \text{Прилагаме Pauli-X на втория кубит.} \\ |11\rangle \rightarrow \text{Прилагаме Pauli-X на първия кубит.} \end{cases}$$

И това би проработило и би ни дало

верен резултат. Нека разпишем случаите и да се убедим:

1. $|x_1, x_2\rangle = |00\rangle : |x_1\rangle = 0 \rightarrow |\psi_1\rangle = |\psi_2\rangle$ и $|x_2\rangle = 0 \rightarrow |\psi_2\rangle = |\psi_3\rangle$ ¹⁶
2. $|x_1, x_2\rangle = |01\rangle : |x_1\rangle = 0 \rightarrow |\psi_1\rangle = |\psi_2\rangle$, но $|x_2\rangle = 1 \rightarrow |\psi_1\rangle \neq |\psi_3\rangle \rightarrow$ третия се различава от останалите
3. $|x_1, x_2\rangle = |10\rangle : |x_1\rangle = 1 \rightarrow |\psi_1\rangle \neq |\psi_2\rangle$ и $|x_2\rangle = 0 \rightarrow |\psi_1\rangle = |\psi_3\rangle \rightarrow$ втория се различава от останалите

¹³Бит, който не допринася за изчислението, а е необходим само за засичането и поправянето на грешки. [Ancilla bit](#)

¹⁴Тези разсъждения са взети от лекциите с цел пълнота на представянето.

¹⁵[Pauli-X](#) трансформацията, кръстена на Волфганг Паули. Представява класическата *NOT* операция и има матрица $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

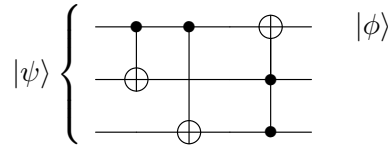
¹⁶Използваме $|\psi_i\rangle$ за означение на i -тия кубит в системата

4. $|x_1, x_2\rangle = |11\rangle : |x_1\rangle = 1 \rightarrow |\psi_1\rangle \neq |\psi_2\rangle$ и $|x_2\rangle = 0 \rightarrow |\psi_1\rangle \neq |\psi_3\rangle \rightarrow$ първия се различава от останалите

Ако измерим кое да е от гореспоменатите спрямо проекционния постулат $|\psi\rangle$ ще премине със скок в съответното състояние, което предполагаемо имаме.

Например, ако измерим $|00\rangle$ ще имаме информация, че всички битове са еднакви следователно ще имаме $\langle 000 + 111|\psi\rangle$, аналогично ако $|x_1, x_2\rangle = |01\rangle$ то ще знаем¹⁷, че първия и третия кубит на системата се различават и можем да проектираме $\langle 110 + 001|\psi\rangle$ и вече се вижда всъщност от къде идва решението за прилагане на *Pauli – X* към третия [1]. Аналогично и с останалите случаи: проектираме върху линейната комбинация, за която сме сигурни, че представлява състоянието на квантовия регистър.

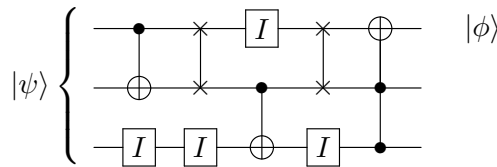
Малка хитрина Можем да спестим няколко операции като всъщност се замислим какво ни трябва, а именно кубитът $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ кодиран чрез $ENCODE_1$ и получен след шум като $|\psi\rangle$. Ако се интересуваме само от него можем да приложим операцията на Тофоли¹⁸ и в крайна сметка да елимираме участието на ancilla битове и да обръщаме само $|\psi\rangle$ (първия кубит в системата $|\psi\rangle$), ако е необходимо. И тогава нашият алгоритъм схематично ще изглежда така:



Фигура 9: Декодирание чрез операцията на Тофоли

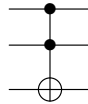
Добавен бонус към това декодиране е, че правим по-малко операции, съответно намалява акумулирания шанс за грешка при прилагане на операциите.

Сега за целите на изложението е необходимо да изпишем алгоритъма в явен вид и да пресметнем неговата матрица. Разписваме схемата Фигура 9 и получаваме:



Фигура 10: Декодирание чрез операцията на Тофоли(разписна версия)

Въпреки, че операцията на Тофоли е дефинирана при:



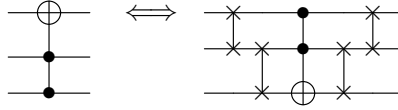
Фигура 11: Стандартната операция на Тофоли

¹⁷Под "знаем" се има предвид, че можем да разсъждаваме за нещо като за дадено.

¹⁸Гейта на Тофоли е двойно контролирана NOT операция с матрица: $CCNOT =$

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

Но много лесно можем да разпределяме "контролите"¹⁹, където поискаме чрез умножение по *SWAP* матрицата (отляво или отдясно) дадена в (4). Често това не се отразява експлицитно, защото се счита, че винаги можем да преномерираме кубитите в края и да получим желания резултат. С цел експлицитност на изложението ще извършвам и тези операции, тъй като това допринася за яснота на цялостното разбиране. В случая резултата е очевиден:



Фигура 12: Разписана модифицираната операция на Тофоли от Фигура 9

$$NOTCC^{20} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (20)$$

След като за стандартната трансформация използвахме *CCNOT* и нейните контроли са към "горните" кубитове, то логично можем да заключим, че можем да смятаме наименованието като всяка паралелна операция се поставя след "горната". Щом *CCNOT* е Controlled-Controlled-NOT, то най-логичното наименование на получената матрица е *NOTCC* - NOT-Controlled-Controlled. След като имаме (4) и (20) можем да пресметнем матрицата на *DECODE*₁ операцията от Фигура 10

$$DECODE_1 = NOTCC.(SWAP \otimes I).(I \otimes CNOT).(SWAP \otimes I).(CNOT \otimes I) \quad (21)$$

Сега извършваме операциите в (21) и получаваме:

$$DECODE_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (22)$$

Примерни изчисления с декодиращата матрица(проверка) Сега, за да се уверим, че наистина това е матрицата на декодирането ще я приложим към резултатите от: (9), (12), (15), (18)

*DECODE*₁ приложено на резултата от (9)

$$DECODE_1. |000\rangle = |000\rangle \quad (23)$$

$$DECODE_1. |111\rangle = |100\rangle \quad (24)$$

¹⁹Контролите на една операция са изобразени като точки на схемите и служат за обозначаване на това, че дадена операция се контролира от кубитът в линията, където е точката.

²⁰Прави ни впечатление, че всъщност матрицата на *NOTCC* не е много по-различна от тази на *CCNOT*, дори можем да забележим, че е просто "транспонирана по обратния диагонал"

$$DECODE_1. | +00 \rangle = | +00 \rangle \quad (25)$$

$$DECODE_1. | -00 \rangle = | -00 \rangle \quad (26)$$

Уверихме се, че декодира правилно изчислените състояния, какво ще получим обаче, ако получим системата $|\psi\rangle = |011\rangle$, бихме очаквали да има шум в първия кубит от системата (именно този, който ни интересува), затова прилагаме $DECODE_1$ и получаваме

$$DECODE_1. |011\rangle = |111\rangle \quad (27)$$

И както можем да забележим първия кубит се обръща и получаваме това, което сме очаквали. Нека разгледаме по-особен случай $|\psi\rangle = \alpha |010\rangle + \beta |101\rangle$

$$DECODE_1. |\psi\rangle = \alpha |010\rangle + \beta |110\rangle \quad (28)$$

Както е видно системата не прие състоянието, което бихме очаквали при пълно декодиране, в случая получихме частично декодиране (само върху първия кубит в системата).

3 Недостатъци на метода

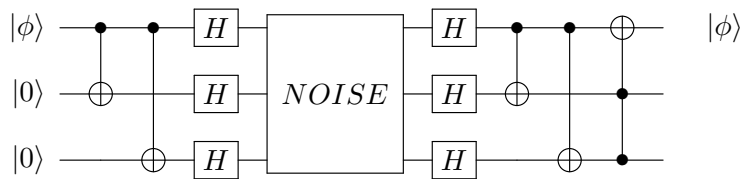
Phase-flip errors Горейзложения метод (вж. Фигура 1 и Фигура 9) могат да засичат и коригират грешки от т. нар. тип "bit-flip errors" но могат да настъпят и "phase-flip errors"¹². Затова е необходимо разглеждане на втори алгоритъм именно за този вид грешки. Ако разгледаме какво всъщност представляват "phase-flip" грешките:

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{\text{Phase flip}} \alpha |0\rangle - \beta |1\rangle \quad (29)$$

В частност:

$$|+\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \xrightarrow{\text{Phase flip}} \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = |-\rangle \quad (30)$$

От (30) можем да забележим, че "phase-flip" грешката в правоъгълния базис може да се разглежда като "bit-flip" грешка в диагоналия. Следователно нашият алгоритъм трябва да "премине" от единия базис в другия. Можем да използваме квантовата Фурие трансформация²¹, но за целите на кодиране и декодиране можем да използваме опростена схема с Хадамар трансформацията.



Фигура 13: Схема за засичане и коригиране на phase-flip грешки

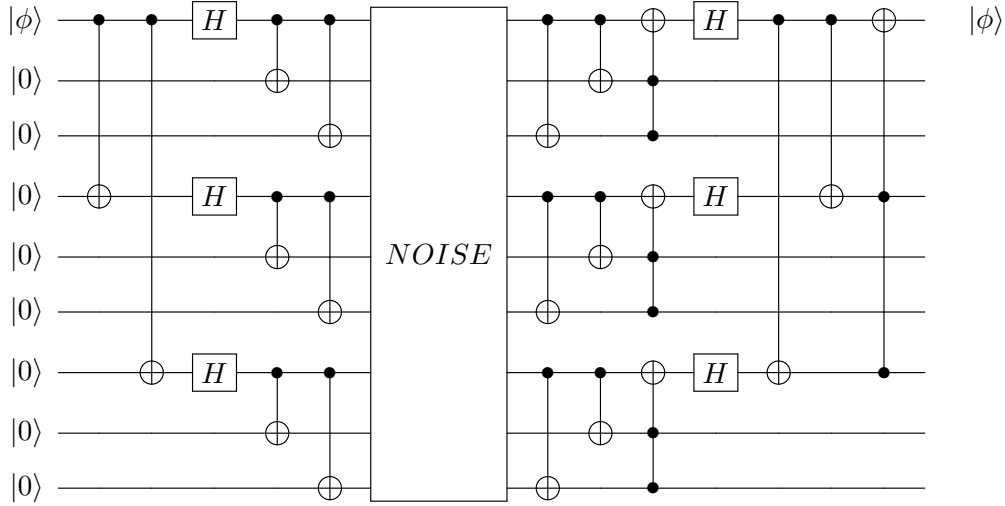
Отново сме се възползвали от идеята, че се интересуваме само от първия кубит на системата, а не от ancilla битовете. Матриците на $ENCODE_{phase}$ и $DECODE_{phase}$ могат тривиално да бъдат изписани от горната схема, чрез пресмятания подобни на (5) и (21)

4 Кодировка на Шор

[3]

²¹ Квантовата Фурие трансформация ни позволява да преминаваме от единия базис в другия като ни предоставя формула за матрицата на трансформацията

История Кодировката разработена от [Питър Шор](#). Чрез нея могат да се коригират произволни линейни комбинации грешки върху *единствен* кубит. Подхода е подобен на съвременни кодировки, а именно подхода на разделяне между логически и физически еденици. При кодировката на Шор един логически кубит(с който правим изчисления) се представя чрез 9 физически кубита. Той съчетава двете гореизложени схеми. Неговата схема изглежда така:



Фигура 14: Кодировка на Шор

Интересното, което може да се отбележи за тази кодировка е, че кубитите на местата 0,3,6 се използват за засичане и коригиране на "phase-flip" грешки, а кубитите по групи - (0,1,2), (3,4,5), (6,7,8) аналогично за "bit-flip" грешки. В общия случай кодировката на Шор може да се използва за грешки от вида на Фигура 7:

$$NOISE = c_1.I + c_2.\sigma_x + c_3.\sigma_y + c_4.\sigma_z \quad (31)$$

Където $c_1, c_2, c_3 \in \mathbb{C}$ и

$$\begin{aligned} \sigma_x &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \sigma_y &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \\ \sigma_z &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{aligned} \quad (32)$$

Съответно разглеждани: σ_x - bit-flip, σ_y - phase-flip, σ_z - и двете

5 Заключение

Настоящата разработка разглежда кодировки на един логически кубит и подходи за откриване и поправяне на единични грешки в системите на същия този логически кубит. Необходимо е да се отбележи, че тази област е обект на продължителни разработки, съответно изложените методи могат да търпят промени с течение на времето и да бъдат усъвършенствани.

Изчисленията и проверките на операциите бяха проведени на симулация на квантов компютър [4]. Също така матричните изчисления бяха улеснени от математическия модул на същата тази симулация.

Цитирани ресурси

- [1] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.
- [2] Albert Einstein и Max Born. *The Born-Einstein letters: Correspondence between Albert Einstein and Max and Hedwig Born from 1916-1955, with commentaries by Max Born*. Macmillan, 1971. ISBN: 9788806379117.
- [3] Peter Shor. “Quantum Computing”. В: (1998), с. 20. URL: <http://www-math.mit.edu/~shor/papers/ICM.pdf>.
- [4] Tsvetelin Tsetskov. *Simulation-Q*. URL: <https://github.com/TsvetelinKostadinov/Simulation-Q>.