

中原大學

資訊工程學系

104 學年度專題實驗期末報告

**隨機性對於量子糾錯算法的重要性及適
用於 IBM Quantum 的量子糾錯算法**

組員

資訊四乙 曾品元 11027233

資訊四乙江庭瑄 11027235

資訊四乙王紫薰 11027238

指導教授

黃琮暉教授

張元翔教授

中華民國 113 年 11 月

摘要—隨著經典電腦技術的進步，它遇到了瓶頸。馮諾依曼架構下，資料儲存技術進步緩慢，經典電腦的運算能力難以大幅提升。相較之下，量子電腦雖然還比較不成熟，但其針對特定問題的時間複雜度卻遠遠超過了經典電腦。

在經典計算機中，一個位元只能處於狀態0或1。然而，迄今為止還沒有一個完美的量子電腦系統，量子位元總是存在錯誤。因此，糾錯變得極具挑戰性，需要與傳統方法完全不同的方法。

在我們的研究中，我們使用模擬器來實現 Shor 糾錯碼 (SRCC)，並針對位元翻轉錯誤問題開發了隨機方法 Shor 糾錯碼 (SRCC-R)。我們考慮錯誤率為 0.007 和 0.002 的 IBM 量子電腦系統。我們的結果表明，使用 SRCC，翻轉錯誤可以糾正少於 25 個受控非門，正確率為 0.97。在我們的方法中，SRCC-R 的翻轉錯誤可以修正超過 500 個受控非門，正確率為 0.95。

關鍵字—環狀糾錯、隨機打亂

目次

摘要及關鍵字	II
目次	III
圖目錄	IV
I. Shor's Error Correction Code	
A. Repetition Code	1
B. Phase Code	1
II. Reasons for Error Correction	
A. Common Quantum Errors	1
B. The Arrangement of the Quantum Computer Qubit Grid	2
III. Implementation and Analysis	
A. Three-Bit Repetition Code.....	3
B. Grouping Error Correction in Circle	3
C. Sampling According to Offset: [-2,0,1]	4
D. Write Back After All Bits Are Finished.....	5
E. Shuffle Randomly.....	6
IV. Conclusion	7
參考文獻	8

圖目錄

Fig. 1(Ⅱ段 A 節)	2
Fig. 2(Ⅲ段 B 節)	3
Fig. 3(Ⅲ段 B 節)	4
Fig. 4(Ⅲ段 C 節)	4
Fig. 5(Ⅲ段 D 節)	5
Fig. 6(Ⅲ段 E 節)	6
Fig. 7(Ⅲ段 E 節)	7
Fig. 8(Ⅲ段 E 節)	7

I. Shor's Error Correction Code

Shor's Error Correction Code 是量子糾錯碼的一種，由美國物理學家 Peter Shor 於 1995 年提出[1]。基本原理是將一個量子位元的資料分佈在多個量子位元上。

A. Repetition Code

重複碼用於將一個量子位元的資料擴展到三個量子位元。

對於狀態如下：

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

其編碼狀態為：

$$|\psi\rangle = \alpha|000\rangle + \beta|111\rangle \quad (2)$$

B. Phase Code

然後每個物理量子位元被編碼成三個量子位元。最終的編碼狀態是 9 個量子位元，每三個量子位元對應同一個狀態，代表一個初始邏輯量子位元。

短糾錯碼是量子運算領域的基礎技術之一。它將一個量子位元的資訊分佈到多個量子位元，並利用冗餘來檢測和修正量子計算過程中的常見錯誤。基於此演算法，形成了本文的構思。然而，值得一提的是，1996 年，Steane [2] 發現了 7 個量子位元程式碼中的通用程式碼。Calderbank 和 Shor [3] 以及 Steane [4] 的論文也展示了量子糾錯的一般理論。

II. Reasons for Error Correction

A. Common Quantum Errors

量子誤差來自硬件，包括準備初始狀態[5]、相干時間[6]、閘操作誤差[7]、測量誤差[8]等。量子誤差有很多種。以下是常見的量子錯誤。

- *Bit Flip Errors*：當量子位元 (qubit) 在計算過程中因外部幹擾或其他原因從 0 翻轉到 1，或從 1 翻轉到 0 時，就會出現這種錯誤。
- *Phase Flip Errors*：這個誤差涉及量子位元的相位訊息，量子位元在計算過程中可能會受到外在因素的影響，導致其相位不正確。
- *Gate Operation Errors*：量子計算機使用閘運算來執行計算。這些操作可能會受到外界環境的干擾，也可能在實際實現中存在一些不完美的表現，導致門操作錯誤。
- *Qubit Crosstalk Errors*：不同量子位元之間的相互作用可能導致它們之間的訊息傳輸錯誤，這可能發生在相鄰量子位元之間。

- *Phase Correction Errors*：這些錯誤在嘗試校正相位誤差時發生，並可能導致進一步的計算錯誤。
- *Connectivity Errors*：由於量子位元之間的連接關係可能是有限的，因此在計算過程中可能會出現連接錯誤，影響計算的正確執行。
- *Timing Errors*：這些錯誤可能是由於計算過程中計時不準確造成的，導致閘門操作和量子位元的同步不正確。
- *Measurement Errors*：在量子計算結束時，測量操作可能會受到干擾，導致計算結果出現錯誤。

在我們的研究中，我們將重點放在位翻轉錯誤。有幾個原因。首先，比特的資訊是最重要的計算資訊。其次，雖然相位資訊在量子計算中也很重要，但是，它是使用相同的糾錯方法透過改變測量基礎來修正相位資訊。第三，我們想要找出量子電腦產生誤差的能力，因此我們不考慮硬體的觀點來減少特定量子電腦中的誤差以提高保真度。

B. The Arrangement of the Quantum Computer Qubit Grid

量子糾錯演算法不僅要考慮糾錯後的精確度，用於糾錯的二量子位元閘也必須符合量子電腦量子位元網格的排列。此外，考慮硬體結構也很重要。因此，我們選擇 IBM Heron r1 處理器[9]，如圖 1 所示，並開發了量子糾錯線來對應系統中量子位元的排列。

IBM Heron r1 processor:

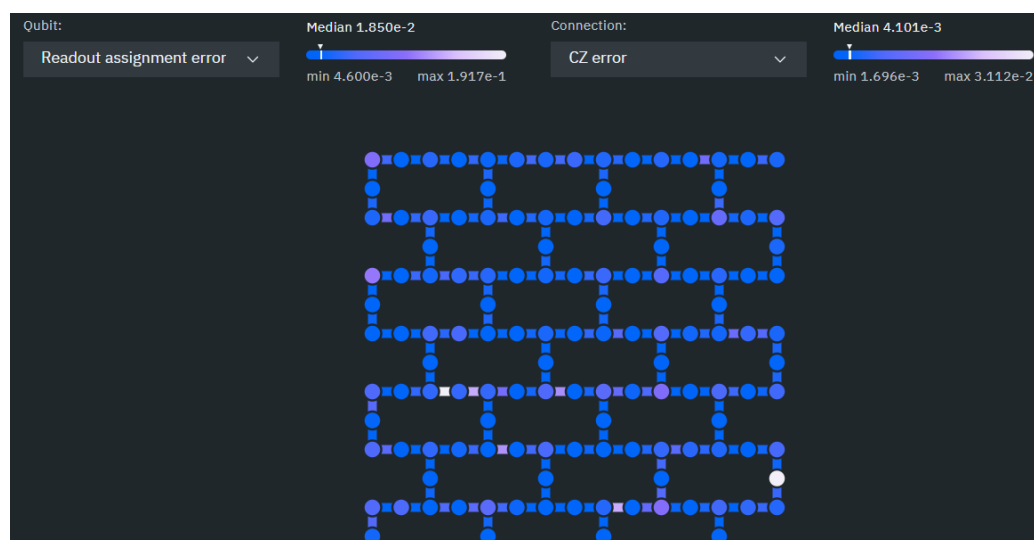


Fig. 1. The arrangement of the qubits and the current error rate map of IBM Heron r1 processor[9]

此量子處理器主要使用 CZ-gate。中位數錯誤率為 0.004101。由於量子誤差不僅是位元翻轉誤差，還包括相位誤差，所以這個 0.004101 的誤差率太大，經過十幾次運算就無法得到正確答案。因此，量子糾錯演算法對於量

子電腦的成功運作非常重要。

III. Implementation and Analysis

A. Three-Bit Repetition Code

首先，根據 SRCC 的說法，重複一個量子位元操作三次應該會得到三個相同的結果。然而，操作過程中可能會出現錯誤，導致三個量子位元不同。三位元重複碼就是將這三個量子位元統一校正為多數答案，也就是說將這三個量子位元放入全加器中並取進位元位元。

B. Grouping Error Correction in Circle (shown as Fig.2)

為了考慮 IBM 量子電腦系統的結構，我們考慮循環結構的肖爾糾錯碼。圖 2 示範如何將 6 位元組成一個環並分組以進行糾錯。執行六次要修正的量子位元操作，並將這些結果循環分組以進行糾錯。將 012、123、234……等分配為一組，組成三位重複碼並進行糾錯。但不一定要 6 位才能組成環，所以後續實驗會在 20 位上進行。

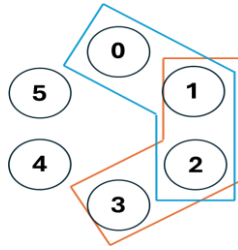


Fig. 2. 6 bits grouping error correction in circle

但結果並不好，很明顯，即使在錯誤率較低的情況下，經過 100 次受控非門操作後，正確率幾乎下降到 0.6 以下，如圖 3 所示。我們假設這是由連續位錯誤引起的。假設我們的目標是將所有量子位元修正為 0。也就是說，如果連續出現錯誤，就沒有辦法修正。兩個量子位元連續出錯是很容易發生的，尤其是 cx-gate（或其他雙量子位元閘）。因此，我們假設要打破無限循環，必須打亂採樣序列。為了檢查假設，我們設計了 c 和 d 小節。

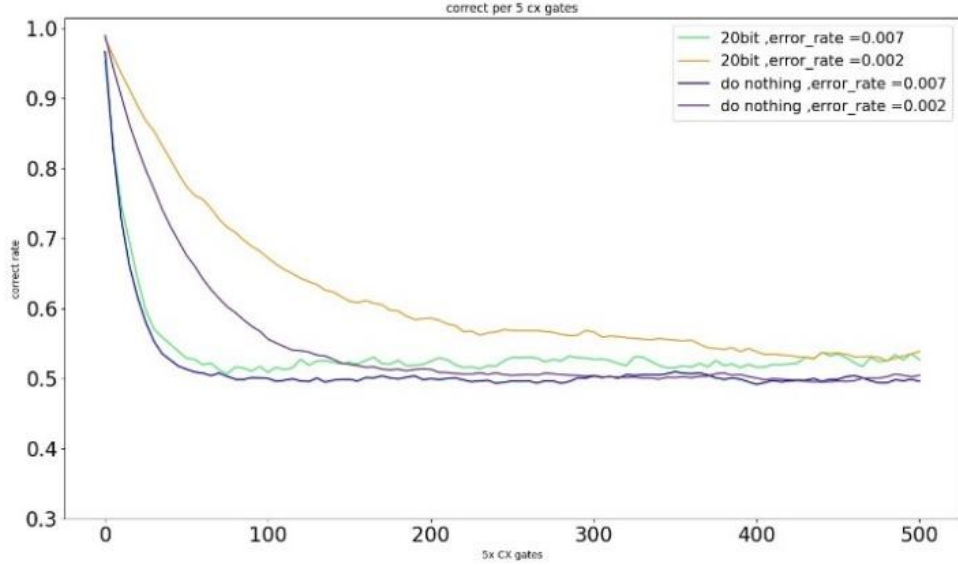


Fig. 3. Results of 6-Bit Grouping Error Correction in Circle. Consider the error rate equal to 0.007 and 0.002 in without error correction (dark blue and green) and SRCC error correction (purple and yellow).

C. Sampling According to Offset: $[-2, 0, 1]$

首先，我們需要在模擬器中找出最佳結果。經過實驗，我們發現依照偏移量取樣： $[-2, 0, 1]$ 糾錯效果最好（注意這裡的 $[-2, 0, 1]$ 定義為三個索引除以六的餘差，或將正號和負號分別視為順時針和逆時針。例如，對應圖 2，應選擇 $[4, 0, 1]$ 作為一組。結果如圖 4 所示。顯然，當錯誤率低於 0.002 且處於最佳指標時，SRCC 在循環結構中是有用的。然而，它很難在真實設備中實現，因為它無法找到真實設備中的最佳索引。

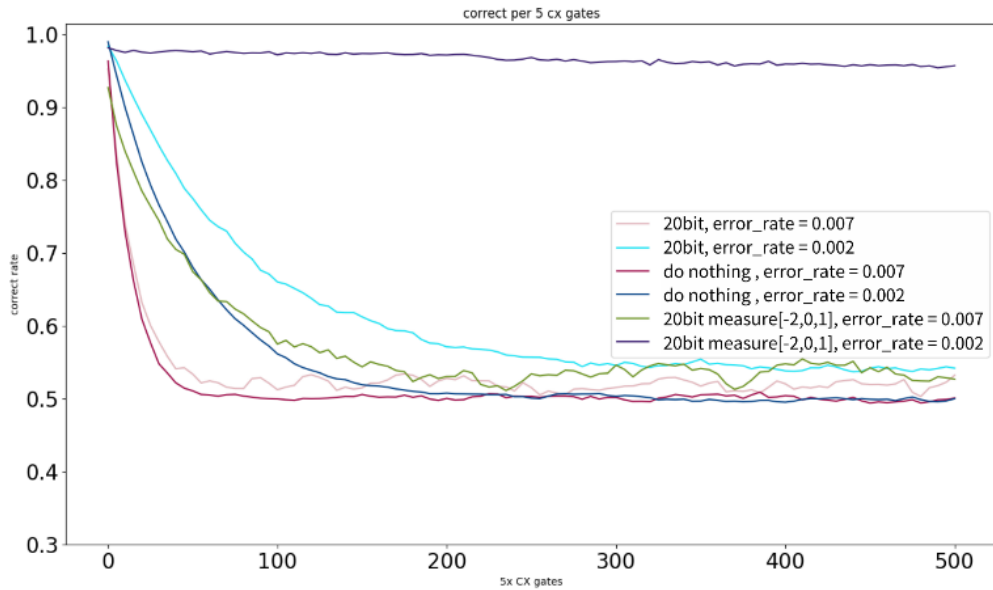


Fig. 4. Results of Sampling according to offset: $[-2, 0, 1]$

D. Write Back After all bits are finished

另一方面，儘管我們要求最好的結果並寫回所有量子位元，但錯誤仍然出現。很容易檢查改變的量子位元越多，發生的錯誤就越多，如下所示：

100001 -> 110001 -> 111001...

原因是，如果第一組三位重複碼的糾錯結果仍然錯誤，則寫回 0 量子位元後，將由第二組三位重複碼取樣進行糾錯。這顯然會導致錯誤的步驟和錯誤的結果。

更好的做法是將每組三位重複碼的糾錯結果暫時儲存到另一組與 6 量子位元索引對應的量子位元中，然後對下一組三位重複碼進行取樣。原來未修正的錯誤避免了各個修正錯誤互相影響。所有糾錯完成後，將結果寫回原來的 q_0 。「全部位寫完後回寫」的結果如圖 5 所示。很明顯，正確率比之前的結果好。使用 500 次受控非閘操作後，即使錯誤率等於 0.007，正確率也高於 0.8。需要提到的是，在第 100 次操作中有一個北斗，因為我們在這次操作中放入了巨大的誤差，誤差等於 CNOT 閘誤差的 20 倍。很明顯，使用我們的方法可以恢復正確率。

結果可以發現，當錯誤率稍大時，糾錯率會明顯下降。

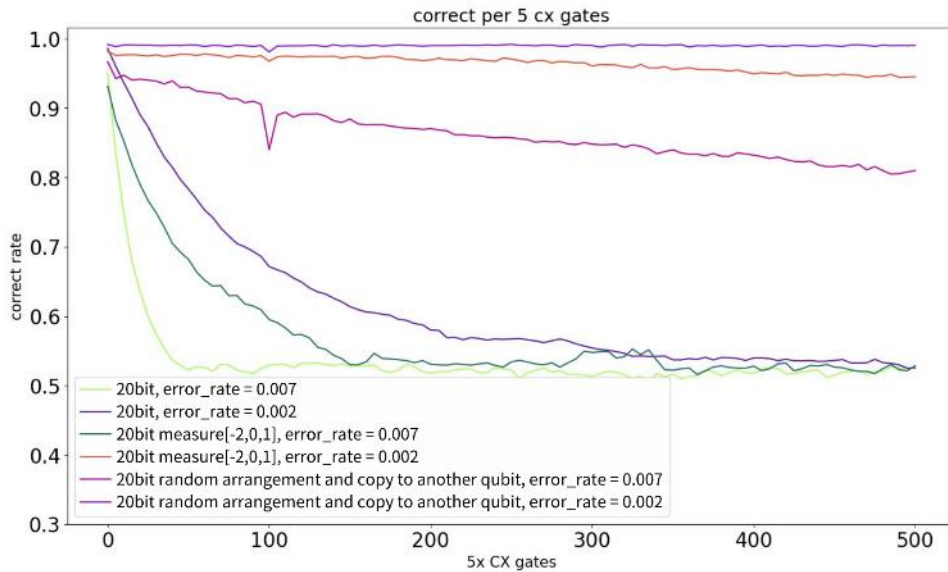


Fig. 5. The results of “write back after all bits are finished”. It is obviously that the correct rate is enhanced. It should be mentioned that we give a huge error in 100th controlled-not gates operations in the system. In our method the correct rate will be recovered.

E. Shuffle Randomly

根據上述思路，如果我們能夠對隨機選擇的索引量子位元進行打亂，糾錯將會更有效率。考慮到量子位元的實際排列，雖然這種隨機改組的方法在現實中並不存在，但我們可以觀察「隨機改組」是否能夠幫助極大地提高糾錯能力，並且發現確實如此。

以 IBM 布里斯班量子計算機的量子比特排列為例，示範隨機洗牌的步驟：

1) 橙色框中的量子位元是要修正的數據，紅色框中的量子位元是用於儲存糾錯結果的附加量子位元空間。圖中，橘色框中的數字 40、41 和 42 是相鄰的量子位元。先取這三個量子位元進行糾錯（放入加法器得到進位位），結果存入編號為 60 的量子位元中。比特號 64 中（如圖 6）。

2) 在將糾錯結果儲存在編號 53 和 54 之後，將要修正的資料的量子位元位置成對交換。交換方式如圖 2 中藍框線所示。加擾後，得到編號 40、41、42 的新量子位元。這三個量子位元用於糾錯，結果儲存在編號為 60 的量子位元中。到儲存空間直到填滿空間，即得到所有糾錯結果。（如圖 7）

Fig. 8 顯示我們的結果。.

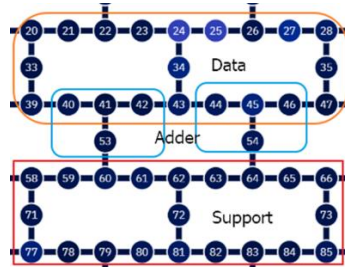


Fig. 6. Step1

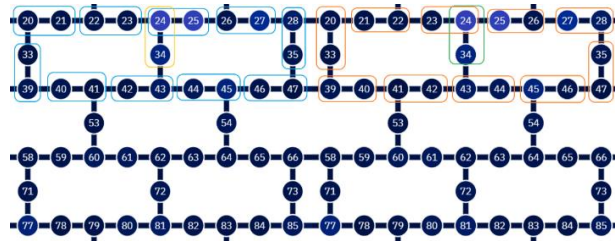


Fig. 7. Step2

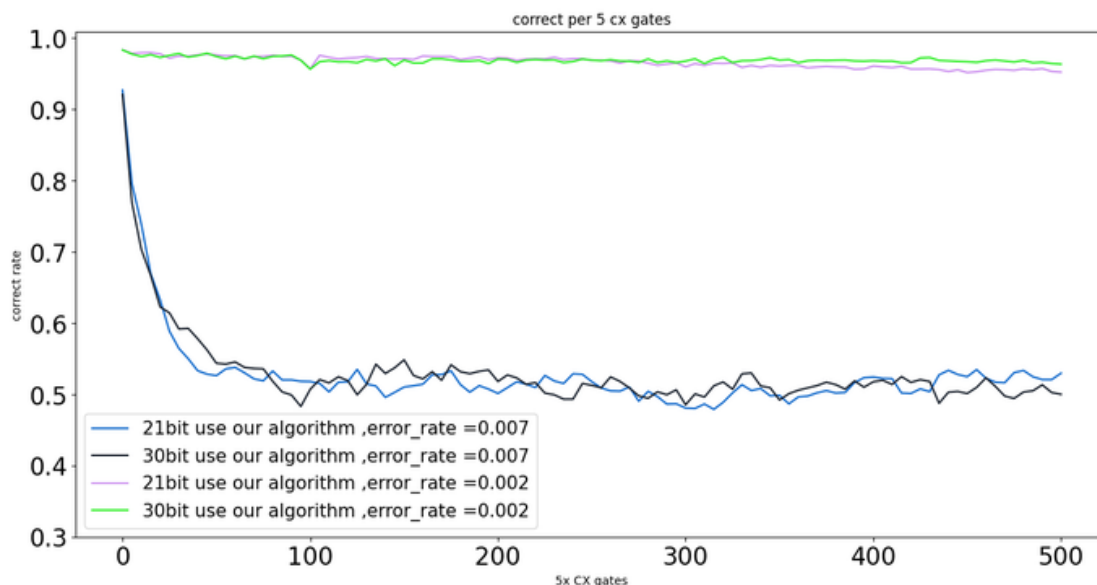


Fig. 8. The results of our algorithm. We can see that even more than 20 bits such as 21 bits and 30 bits can have a good effect when the error rate is 0.002

IV. Conclusion

總結來說，量子錯誤更正是一個複雜的問題。雖然我們提出的所有方法在錯誤率為 0.007 以下時無法保持穩定的準確率，但如果樂觀地採用 IBM 最新量子電腦的錯誤率（0.002）進行實驗，我們的方法確實可以在某一水準上保持準確率（0.95）。在這個過程中，我們還在特定時間施加了較大的錯誤率，以觀察我們的錯誤更正方法是否能靈活地將準確率恢復至原本的水準。結果顯示非常出色，我們的錯誤更正方法在 IBM 的量子電腦上能保持一定的準確率。此外，我們也理解了隨機排列對所有錯誤更正算法的重要性。在未來開發錯誤更正算法時，應特別關注該方法是否會導致隨機排列的出現。

參考文獻

- [1] Shor P. W., Phys. Rev. A, 52 (1995) R2493-R2496
- [2] Steane A. M., Phys. Rev. Lett., 77 (1996) 793-767.
- [3] Calderbank A. R. and Shor P. W., Phys. Rev. A, 54 (1996) 1098-1105
- [4] Steane A. M., Proc. Roy. Soc. Lond. A, 452 (1996) 2551-2577.
- [5] I. H. Kim, Noise-resilient preparation of quantum many-body ground states, arXiv:1703.00032 (2017).
- [6] Aaron Somoroff, Quentin Ficheux, Raymond A. Mencia, Haonan Xiong, Roman Kuzmin, and Vladimir E. Manucharyan, Phys. Rev. Lett. 130, 267001
- [7] D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen, Phys. Rev. A 96. 062306
- [8] Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, Paolo Stornati, and Xiaoyang Wang, Phys. Rev. A 105, 062404
- [9] <https://www.ibm.com/quantum>