中原大學

資訊工程學系

112學年度專題實驗進度報告

**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**

資訊三乙 11027233 曾品元

資訊三乙 11027235 江庭瑄

資訊三乙 11027238 王紫薰

指導教授

黃琮暐教授

張元翔教授

中華民國113年6月18日

摘要及關鍵詞

隨著古典電腦技術的發展，其計算能力面臨瓶頸，尤其是資料儲存技術進展緩慢。而量子電腦在某些特定問題上具有顯著的優勢，能以極快的速度處理計算。量子電腦的位元（qubit）可以同時處於1和0的狀態，使其能夠在搜尋計算方面超越古典電腦。然而，量子電腦的錯誤率高，糾錯變得極具挑戰性，需採用不同的糾錯方法。本研究探討了three-bit repetition code和環狀分組糾錯探討。三位重複碼將一個qubit的運算重複三次，並將結果統一更正為佔多數的答案。環狀分組糾錯則是將qubit運算進行六次，並依序組成three-bit repetition code進行糾錯。然而，這些方法可能會遇到死循環問題，導致糾錯失敗。為了解決這些問題，研究提出了一種新的取樣方法，經實驗發現，隨機打亂qubit排列能大幅提升糾錯能力。理想中的隨機打亂可在量子電腦上實現，具體步驟包括對qubit位置進行交換，再進行糾錯。結果顯示，當量子電腦本身錯誤率較低時，糾錯演算法效果更佳。為處理相位問題，可先將相位旋轉映射到x軸上，完成糾錯後再反向旋轉回去。實驗使用了三個不同的錯誤率，分別為0.007395、0.002795、0.001433，顯示了這些糾錯演算法在未來量子電腦中的潛力。

目次

摘要……………………………………………………II

目次……………………………………………………III

圖目錄……………………………………………………IV

表目錄……………………………………………………V

糾錯演算法的實作與析……………………………………………………1

參考文獻……………………………………………………5

圖目錄

圖A：6-qubit環狀分組示意圖

圖B：6-qubit環狀糾錯成果

圖C：[-2, 0, 1]取樣示意圖

圖D：當量子電腦本身的錯誤率為0.002時的糾錯成果

圖E：當量子電腦本身的錯誤率為0.007時的糾錯成果

圖F：錯誤率0.001433下的糾錯率圖

圖一：IBM brisbane的量子電腦的qubit排列方式，示範隨機打亂的分組情形

圖二：IBM brisbane的量子電腦的qubit排列方式，示範隨機打亂的分組情形

圖三：IBM brisbane的量子電腦的qubit排列方式，示範隨機打亂的分組情形

表目錄

**糾錯演算法的實作與析**

**3-bit重複編碼**

三比特重複編碼（three-bit repetition code）是將一個待糾錯的qubit運算重複三次，理應獲得三個一模一樣的結果，然而運算過程可能會出錯，導致三個qubit不全相同。這個編碼技術即是將這三個qubit統一更正為佔多數的答案，即取三個qubit中的多數票。若結果為000, 001, 010, 100，則統一更正為0；若結果為111, 110, 101, 011，則統一更正為1。

A blue and orange lines on a black background

Description automatically generated**環狀分組糾錯**

將一個待糾正的qubit運算進行六次，並運用六個結果循環分組進行糾錯。依序將012, 123, 234, 345, 450, 501分為一組形成three-bit repetition code並執行糾錯。

右圖A

A graph with a line

Description automatically generated with medium confidence

**6-qubit環狀糾錯成果**

雖然此方法運用循環分組進行糾錯，但有時正確率甚至不如糾錯前。如下：

錯誤情形：000110 -> 000110 (Error)

假設目的是將所有qubit糾正為0，則當遇到連續兩個1時會發生死循環，無論糾錯幾次都無法修正為0。意即，只要有連續的錯誤，糾錯就會失敗。連續兩個qubit同時壞掉的情況是很容易發生的，尤其在cx-gate或其他雙qubit gate操作中。為解開死循環，需將取樣順序打亂。

上圖B

**如何取樣**

A screenshot of a computer

Description automatically generated經實驗發現，依照Index：[-2, 0, 1]取樣的糾錯效果最佳。假設欲修正index為4的qubit，則取index為2(4-2)、4(4+0)與5(4+1)的qubit組成three-bit repetition code，獲得糾錯結果後寫回index為4的qubit。

右圖C

**量子電腦的錯誤率很關鍵**

以下為當量子電腦本身的錯誤率為0.002與0.007時的糾錯成果：

A graph of a graph

Description automatically generated with medium confidence

左圖D

A graph with different colored lines

Description automatically generated

左圖E

可以看到當量子電腦本身的錯誤率為0.007時，效果遠不如錯誤率為0.002的量子電腦。由此可知，量子電腦本身的錯誤率很大程度上影響糾錯演算法的成功率。

**新的死循環**

假設欲將6-qubit以index=[-2,0,1]取樣糾錯以修正為全0，而當錯誤情形如下時，糾錯反而越錯越多：

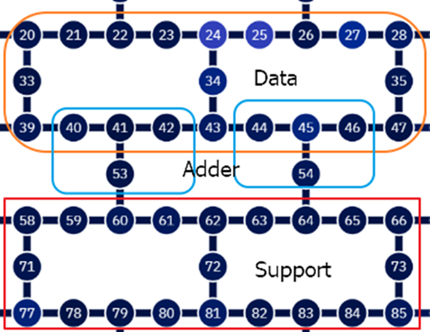
100001 -> 100001 -> 110001 -> 111001 …

原因是若第一組three-bit repetition code糾錯出來的結果仍是錯的，寫回第0-qubit後，會在被第二組three-bit repetition code取樣進行糾錯，導致一步錯，步步錯的結果。所以應該要將每一組three-bit repetition code糾錯出來的結果先暫存至另一組6-qubit相對應index的qubit中，取樣下一組three-bit repetition code時取的都是原始未糾錯過的，以避免每次糾錯互相影響。待所有糾錯都完成，再將結果寫回原本的q0。

**隨機打亂**

同樣錯誤率為0.002與0.007的糾錯結果，增加了另一種作法的觀察——模擬qubits隨機重新排列（灰色）。

以下以IBM brisbane的量子電腦的qubit排列方式為例，示範隨機打亂：



圖一：橘色框內的qubit為待糾錯的資料，紅色框內的qubit為欲儲存糾錯結果的額外qubit空間。圖中橘框內編號40、41、42為相鄰qubit，首先取這三個qubit進行糾錯（放進加法器取carry位），將結果存放到編號60的qubit。

圖二、圖三：編號53、54已存放糾錯後的結果後，將待糾錯資料的qubit兩兩位置交換，交換方式如圖二的藍色框線所示，完成後再如圖三的橘線所示兩兩互換位置，即完成一次打亂。打亂後得到新的編號40、41、42的qubit，取此三個qubit進行糾錯得結果存入編號60的qubit。以此類推。

A diagram of numbers and a number

Description automatically generated with medium confidenceA diagram of numbers and a number

Description automatically generated with medium confidence

錯誤率0.001433下的糾錯率圖：**A graph of a line

Description automatically generated with medium confidence**

左圖F

參考資料

### <https://zh.wikipedia.org/zh-tw/%E9%87%8F%E5%AD%90%E9%96%98>

### <https://www.researchgate.net/figure/a-Toffoli-gate-b-Swap-gate-c-Quantum-gate-d-Quantum-gate-stored-in-a-superposition_fig13_257641933>