

中華民國第 65 屆中小學科學展覽會

作品說明書

高級中等學校組 物理與天文學科

第二名

051814

真實量子運算中的錯誤與修復：Qiskit 的噪聲
模型與錯誤更正實驗

學校名稱：新興學校財團法人桃園市新興高級中等學
校

作者：	指導老師：
高三 林柔均	林柏成
高三 黃新	
高三 吳妤婕	

關鍵詞：量子電路、噪聲模型、Qiskit 錯誤更正

摘要

本研究旨在探討量子計算中由噪聲引發的兩種主要錯誤：位元翻轉錯誤（Bit-flip Error）與相位翻轉錯誤（Phase-flip Error）。我們首先利用 IBM Quantum 平台，透過即時模擬與圖像化操作，建構了一套重複更正碼系統，展示量子糾纏在錯誤校正中的應用。然而，當我們將錯誤更正碼送至真實的量子電腦運算時，發現額外的噪聲干擾使得錯誤更正碼的效率受到影響。因此，我們參考了 IBM 官方網站的資料，並結合在真實量子電腦上獲得的結果，使用 Google Colab 打造了一個最接近真實情況的噪聲模型，利用錯誤參數量化、分析真實量子電腦的出錯情形。

壹、研究動機

量子電腦被認為是未來運算的主力，其量子位元的特性使得大量資料能夠快速處理。然而，目前的技術僅能同時操控少數量子位元，且極易受到外界噪聲的干擾，這使得量子電腦尚未能廣泛應用於各大產業。因此，我們著手研究量子電腦的錯誤機制，試圖克服這一量子發展的瓶頸。

貳、研究目的

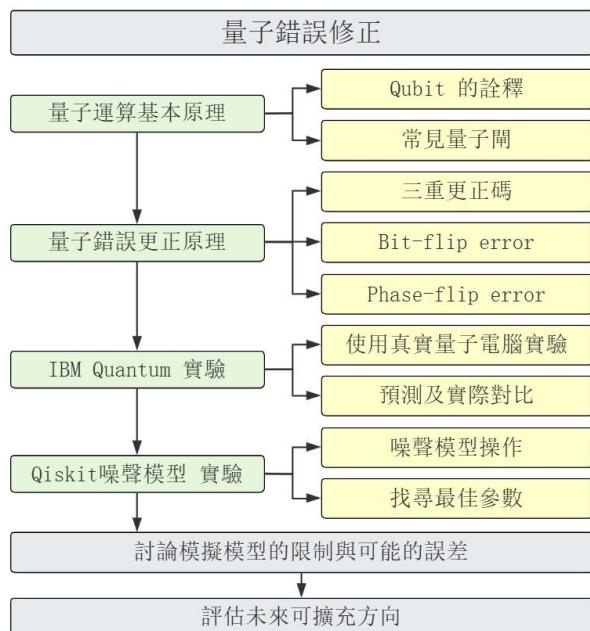
- 一、實作錯誤更正碼
- 二、建構噪聲模型，探討錯誤發生的物理機制
- 三、比較噪聲模型與真實量子電腦跑出來的結果。

參、研究設備及器材

表一：研究設備及軟體表（研究者自製）

	硬體設備	軟體設備		
名稱	電腦 (電子計算機)	IBM Quantum Composer	Google Colab	IBM Quantum Lab Qiskit 模組
圖片				

肆、研究過程或方法



圖一：研究架構圖（研究者自製）

一、研究架構

我們使用 IBM Quantum 學習量子錯誤更正碼和量子除錯時的機制，將實驗得到的真實結果，在 Google Colab 上使用 Qiskit 模組進行模擬，利用錯誤參數的調整找出最符合真實狀況的排列，進而推測真實量子電腦中所遇到的主要錯誤，再利用實作理解實驗數據與模擬差異。

二、研究平台

(一)、IBM Quantum Composer

透過 IBM Quantum Composer，能將工具列中的量子閘元件拖曳至電路上，將量子編程以圖像化方式呈現，下方可即時觀察狀態向量及結果的機率分布，由於顯示的結果皆為模擬，因此還需將編寫的程式提交至真實的量子電腦進行運算。

(二)、Google Colab 和 Qiskit 模組

Qiskit 模組是 IBM Quantum 開發的一個開源軟體工具包，提供使用者建構量子電路所需的工具，例如 NoiseModel 能夠提供噪聲模擬。我們利用 Google Colab 平台結合 Qiskit 建構一個量子模擬環境，並透過撰寫 Python 程式碼建構量子電路。

(三)、量子電腦

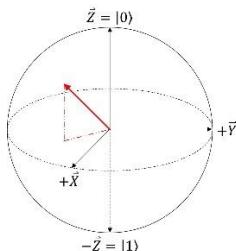
近年來興起的量子電腦，由量子位元（qubit）組成，量子位元是量子資訊的基本單位，具備可同時處於「0」與「1」狀態的疊加性，與只能是「0」或「1」的經典位元（bit）不同。當量子位元被觀測時，會塌縮至某個本徵態（如 $|0\rangle$ 或 $|1\rangle$ ）。

目前發展的量子位元實作方式包括光子偏振態、原子自旋態與超導量子位元等。量子電腦的目標並非取代傳統電腦，而是利用其特性處理經典電腦難解的問題，如模擬材料的量子行為、破解加密技術、優化演算法等，展現出廣泛應用潛力。

三、量子與量子邏輯閘

(一)、布洛赫球面

布洛赫球是一顆假想的球面，分布在 xyz 軸組成的空間，正 z 軸代表 0，負 z 軸代表 1。因量子具有疊加的特性，球面上的任意點都可以代表量子態（傳統位元只能是兩極的 0 或 1），也就是量子位元現在 $|0\rangle$ 和 $|1\rangle$ 的向量總和。



圖二：布洛赫球面示意圖（研究者自製）

(二)、狄拉克表示法

以數學式表示量子的狀態，公式為：

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

代表此時 $|\psi\rangle$ 的向量狀態，此表示法不是單一的， $\alpha|0\rangle$ 代表此量子態在 0 方向上的向量， $\beta|1\rangle$ 代表此量子態在 1 方向上的向量， α^2 和 β^2 則是當下 0 和 1 的機率，即： $\alpha^2 + \beta^2 = 1$ 。此表示法可以使量子態輕鬆的被運用在數學式中以便計算。

(三)、疊加和糾纏

1. 疊加

如同一個向量可以被分解成兩向量的加法，量子態可以表示為多種狀態的機率疊加，例如：

表二：疊加態數學表示法範例（研究者自製）

量子態	=	本徵態 1	+	本徵態 2
$ \psi\rangle$	=	$\frac{1}{\sqrt{2}} 0\rangle$	+	$\frac{1}{\sqrt{2}} 1\rangle$

代表其在被觀測而塌縮後有 50% 的機率是 0，50% 的機率是 1。此種表示方法有無限種，這意味著粒子可以同時存在於多個位置或能量狀態，此即量子的平行性，直到被觀測時才塌縮到某一本徵態，使得量子電腦可以同時計算一個複雜問題背後的龐大可能性，並利用干涉效應（機率波的相互抵銷）一個一個消除不正確的推測。

2. 紛纏

兩量子不管相隔多遠，當一方的狀態被觀測（塌縮至某一本徵態）時，另一方的性質就會在瞬間確定，速度比光速還快，愛因斯坦稱為「鬼魅般的超距作用」，這種性質讓量子電腦可以在同時處理許多資料。

(四)、使用到的量子邏輯閘

正如經典位元利用傳統邏輯閘（例如：AND、OR、NOT）進行運算，量子邏輯閘則對量子位元執行操作。透過將這些量子閘依序組合，可以充分利用疊加與糾纏的特性，從而提升計算效能。

1. Pauli-X 閘（X 閘）

也稱為 NOT 閘，是一個能讓量子位元圍繞 x 軸旋轉 180°的基本操作。簡單來說，就是將一個位元的值從 0 變成 1 或從 1 變成 0。

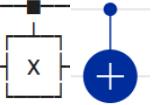
表三：Pauli-X 閘真值表、圖示和矩陣表示法（研究者自製）

(圖) 標示		輸入		X 閘	輸出	
		0⟩	1⟩		1⟩	0⟩
矩陣 表示	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	1⟩			0⟩	
			$\alpha 0\rangle + \beta 1\rangle$			$\alpha 1\rangle + \beta 0\rangle$

2. CNOT 閘 (CX 閘)

就是受控的 X 閘。CNOT 閘結合了兩個量子位元，其中一個作為「控制位元」，用以決定另一個「目標位元」的狀態改變。當控制位元處於|1⟩時，目標位元的狀態會被反轉；而若控制位元為|0⟩，目標位元則維持不變。而在這個操作中，控制位元的狀態不改變，而目標位元則根據邏輯規則發生翻轉，這正體現了量子電路的可逆性，與傳統電腦只呈現最終輸出結果的方式截然不同。

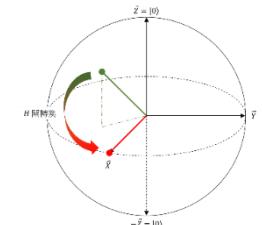
表四：CNOT 閘真值表、圖示和矩陣表示法（研究者自製）

(圖) 標示		輸入		CNOT 閘	輸出	
		A⟩	B⟩		A⟩	A⊕B⟩
矩陣 表示	$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	0⟩	0⟩		0⟩	0⟩
		0⟩	1⟩		0⟩	1⟩
		1⟩	0⟩		1⟩	1⟩
		1⟩	1⟩		1⟩	0⟩

3. H 閘 (Hadamard gate)

疊加態量子閘，可以將量子位元旋轉為機率 50% 是 0，50% 是 1 的狀態，在布洛赫球面上即將|0⟩或|1⟩轉向+X 或-X 軸。

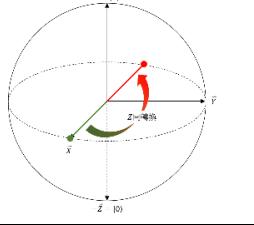
表五：H 閘真值表、圖示、矩陣表示法和布洛赫球面表示法（研究者自製）

(圖) 標示		輸入		H 閘	輸出		布洛赫球面表示
		任意狀態					
矩陣 表示	$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$				$\frac{1}{2} 1\rangle + \frac{1}{2} 0\rangle$		

4. Pauli-Z 閘 (Z 閘)

使量子態沿著 Z 軸旋轉 180 度，改變其相位角，為翻轉相位的功能。

表六：Pauli-Z 閘真值表、圖示、矩陣表示法和布洛赫球面表示法（研究者自製）

(圖) 標示		輸入		輸出	布洛赫球面表示
矩陣 表示	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$ 0\rangle$	Z 閘	$ 0\rangle$	
		$ 1\rangle$		$- 1\rangle$	
		$\frac{1}{\sqrt{2}}(0\rangle + 1\rangle)$		$\frac{1}{\sqrt{2}}(0\rangle - 1\rangle)$	

四、量子錯誤更正 (QEC) 與三重重複碼 (Three-qubit Bit-flip code)

(一)、重複碼

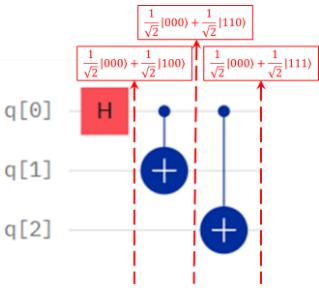
重複碼將想要傳達的資訊不斷的重複，使得接收端可以利用多數決來判斷正確資料，這是一種捨棄傳輸速度以確保正確性的方法。例如：我要傳達一個 0 的訊息，為了確保資料不要因外部因素產生的錯誤無法正確讀取，因此我將 0 重複三遍，變成 000，這樣當接收端收到訊息時，若其中一個位元出錯資料變成 100，仍可以看出正確的資訊為 000。

(二)、量子錯誤更正 (QEC)

因為量子位元的不可複製性及其容易受外界影響而塌縮的特性，我們不能直接將出錯的量子位元改正，否則將失去此位元內儲存的資料。因此將量子錯誤更正分為以下步驟：

1. 重複 (Repetition)

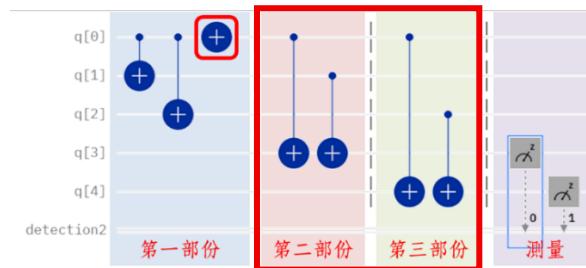
$q[0]$ 為主要進行運算的量子位元， $q[1]、q[2]$ 則是拿來複製資料的位元，他們的初始值皆為 $|0\rangle$ ，而我們可以用兩個 CNOT 閘來完成重複碼的設計：第一個 CNOT 閘：以 $q[0]$ 作為控制位元、 $q[1]$ 作為目標位元，得到 $\alpha|000\rangle + \beta|110\rangle$ ；第二個 CNOT 閘：以 $q[0]$ 作為控制位元、 $q[2]$ 作為目標位元，得到 $\alpha|000\rangle + \beta|111\rangle$ 。



圖三：利用糾纏態製作重複碼（研究者自製）

2. 偵錯 (Detection)

不直接測量主要位元（負責傳遞訊息），而是另外加上輔助位元，製造糾纏態，利用糾纏態的特性確認位元是否出錯。



圖四：利用糾纏態偵錯（研究者自製）

主要是以兩個問題來去做偵錯： $q[0]$ 、 $q[1]$ 測量值是否相同及 $q[0]$ 、 $q[2]$ 二測量值是否相同，並回傳二位元的錯誤類型代碼，如下表：

表七：偵錯產生的錯誤代碼（研究者自製）

	$q[0]$	$q[1]$	$q[2]$	$q[0]q[1]$	$q[0]q[2]$	detection
沒出錯	0	0	0	+1	+1	00
$q[0]$ 出錯	1	0	0	-1	-1	11
$q[1]$ 出錯	0	1	0	-1	+1	01
$q[2]$ 出錯	0	0	1	+1	-1	10

3. 除錯 (Correction)

根據錯誤類型代碼來做除錯。電路設計則利用 X 閘來翻轉位元達到除錯效果，如下表：

表八：錯誤代碼對應修正動作（研究者自製）

錯誤類型代碼	00	11	01	10
修正動作	不用修正	翻轉 q[0]	翻轉 q[1]	翻轉 q[2]

4. 測量 (Measurement)

五、位元翻轉錯誤 (Bit-flip error) 之偵測及除錯

位元翻轉錯誤，即將 0 與 1 互換，在傳統電腦上也可以見到。

(一)、利用 X 閘模擬出錯

在 $q[0]$ 多加上一個 X 閘，利用 X 閘可以翻轉量子位元的特性來模擬錯誤的發生：

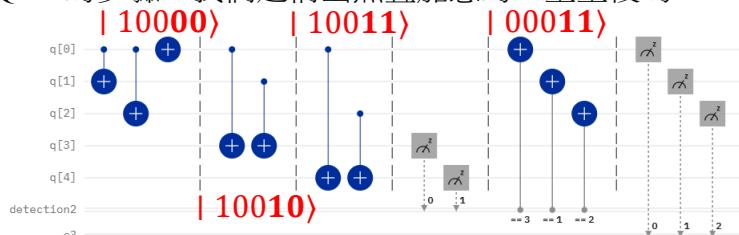
$$X |\psi\rangle = \alpha X |0\rangle + \beta X |1\rangle = \alpha |1\rangle + \beta |0\rangle$$

可以看出 1、0 被交換了。

(二)、位元翻轉錯誤實驗 (無疊加態)

$q[0]$ 、 $q[1]$ 、 $q[2]$ 皆初始化為 0，因並未放上 H 閘所以他們並不是疊加態。我們進行此實驗的目的是作為傳統電腦與量子電腦的銜接，認識重複碼如何進行錯誤更正。

照著 QEC 的步驟，我們建構出無疊加態的三重重複碼。



圖五：位元翻轉錯誤的糾錯電路圖（研究者自製）

1. 製造重複碼

$q[0]$ ($|0\rangle$) 作為控制位元，透過兩個 CNOT 閘分別控制 $q[1]$ 和 $q[2]$ ，將其狀態與 $q[0]$ 進行糾纏，製造重複碼 ($|000\rangle$)，雖然這三個量子位元不管有沒有使用 CNOT 閘輸出都是 000，但為了表現重複碼的意義我們仍放上 CNOT 閘。

2. 模擬出錯

在 $q[0]$ 上用 X 閘翻轉 $q[0]$ ，模擬固定錯誤的發生。

3. 偵錯

為了避免量子位元經測量後產生塌縮而失去資料完整性，我們將三個位元的狀態記錄在另外兩個位元 $q[3]$ 及 $q[4]$ 上， $q[3]$ 分別和 $q[0]$ 、 $q[1]$ 用 CNOT 閘連結，當 $q[0]$ 出錯時（變為 1）， $q[3]$ 在第一個 CNOT 閘受 $q[0]$ 控制，由初始值 0 轉為 1。第二個 CNOT 閘上， $q[1]$ 為 0 因此不改變 $q[3]$ 的狀態。經此兩個 CNOT 閘後 $q[3]$ 經測量後應可得 1。

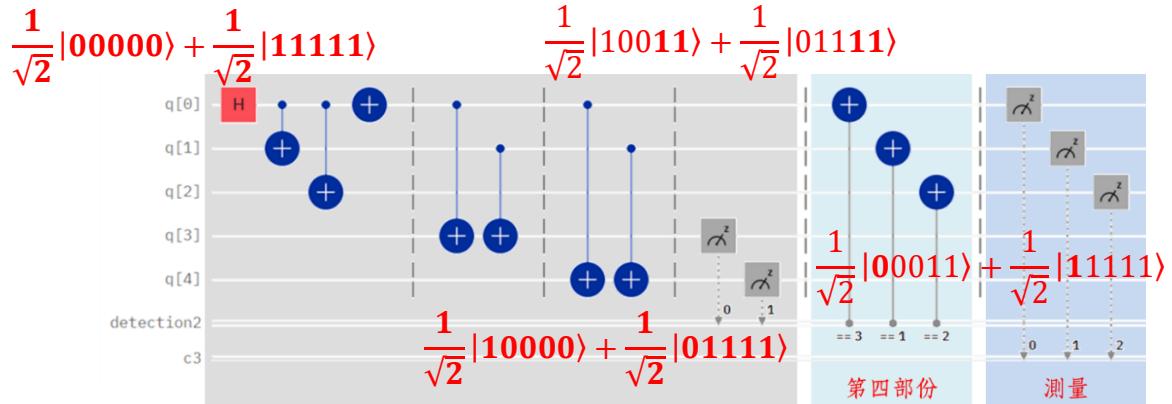
同理在第三部分中， $q[4]$ 分別和 $q[0]$ 、 $q[2]$ 連結，測量後結果應為 1。我們在 detection（經典位元）上測量 $q[3]$ 及 $q[4]$ 並將程式送至真實量子電腦進行 3 次（總數各 10000 次）的運算。

由下表可知，重複碼所運用的「多數決制」亦可能是錯誤無法被修正的原因，例如在三個位元的時候，1 出現兩次以上，就無法被辨識而被修復。

表九：不同數量及位置的位元出錯後 $q[3]q[4]$ 所產生錯誤代碼及修正方法（研究者自製）

	$q[0]$	$q[1]$	$q[2]$	$q[3]$	$q[4]$	detection	修正	結果
原設定	0	0	0	0	0	00	不用修正	000
$q[0]$ 出錯	1	0	0	1	1	11	翻轉 $q[0]$	000
$q[1]$ 出錯	0	1	0	1	0	01	翻轉 $q[1]$	000
$q[2]$ 出錯	0	0	1	0	1	10	翻轉 $q[2]$	000
$q[0]q[1]$ 出錯	1	1	0	0	1	10	翻轉 $q[2]$	111
$q[0]q[2]$ 出錯	1	0	1	1	0	01	翻轉 $q[1]$	111
$q[1]q[2]$ 出錯	0	1	1	1	1	11	翻轉 $q[0]$	111
全錯	1	1	1	0	0	00	不用修正	111

(三)、位元翻轉錯誤 (利用 H 閘做出疊加態)



圖六：疊加態位元翻轉錯誤的糾錯電路圖（研究者自製）

同初始值為 0 的 Bit-flip 實驗，我們利用 X 閘判斷並除錯，再將除錯結果紀錄至 c 並將程式送至真實量子電腦進行 3 次（總數各 10000 次）的運算。

在疊加態上，我們可以更加清楚的模擬真實輪自電路的情況。

六、相位翻轉錯誤 (Phase-flip error) 之偵測

(一)、以 Z 閘模擬 phase-flip error

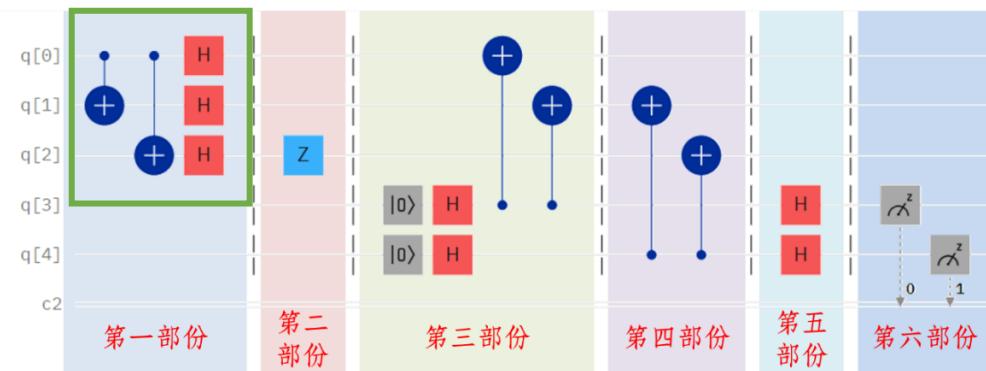
Phase-flip 改變量子位元狀態中的相位，通常用 Z 操作符表示：

$$Z |0\rangle = |0\rangle; Z |1\rangle = -|1\rangle$$

作用於一般的量子位元狀態 $|\Psi\rangle$ 上，可以寫作：

$$Z |\Psi\rangle = Z(\alpha |0\rangle + \beta |1\rangle) = \alpha Z |0\rangle + \beta Z |1\rangle = \alpha |0\rangle - \beta |1\rangle$$

(二)、利用 H 閘轉換 Phase-flip error



圖七：Phase-flip error 電路圖（研究者自製）

在傳統的重複碼中，會將位元重複多次以對抗 Bit-flip error，然而這個編碼對 Phase-flip error 不敏感，因為它只會改變相位。

但經過 H 閘的轉換，我們可以將 Phase-flip error 轉換成 Bit-flip error：

$$HZH = X$$

其中 Z 是 Phase-flip error 的操作，而 X 則對應 Bit-flip error，而此等式可以用數學上的矩陣來理解：

$$HZH = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \right) \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right)$$

即：

$$HZH = \frac{1}{2} \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

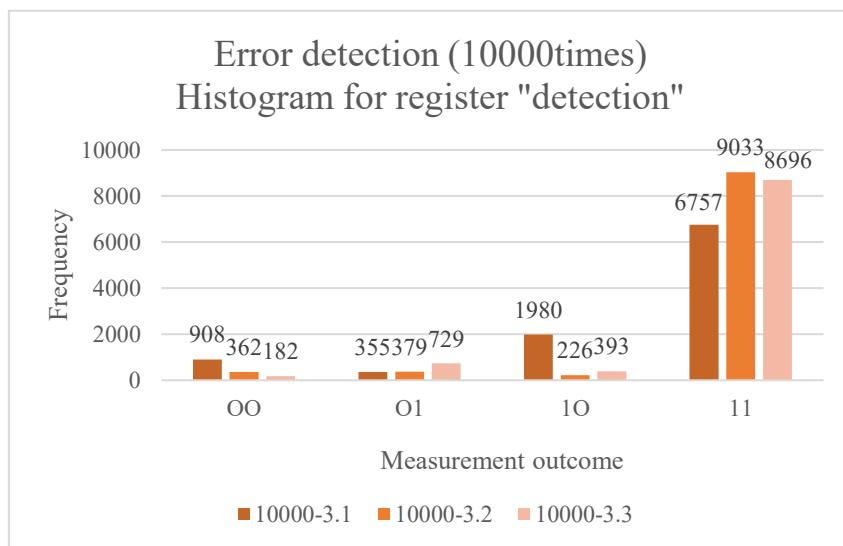
這種錯誤型態的轉換對於許多量子錯誤修正方案來說非常有用，因為重複碼等錯誤修正技術通常是針對位元翻轉錯誤而設計的。

(三)、再次翻轉

X 基底需再次通過 H 閘復原回 Z 基底，以便在結果能讓我們看出原先出錯的位元狀況，接著將程式送至真實量子電腦進行 3 次（總數各 10000 次）的運算。

在進行實驗前，我們定義準確率 (Accuracy) 是指實驗結果中，測得與理論預期相符之結果所佔比例；而錯誤率 (Error rate)：指實驗結果中，測得與理論預期不符之結果所佔比例。

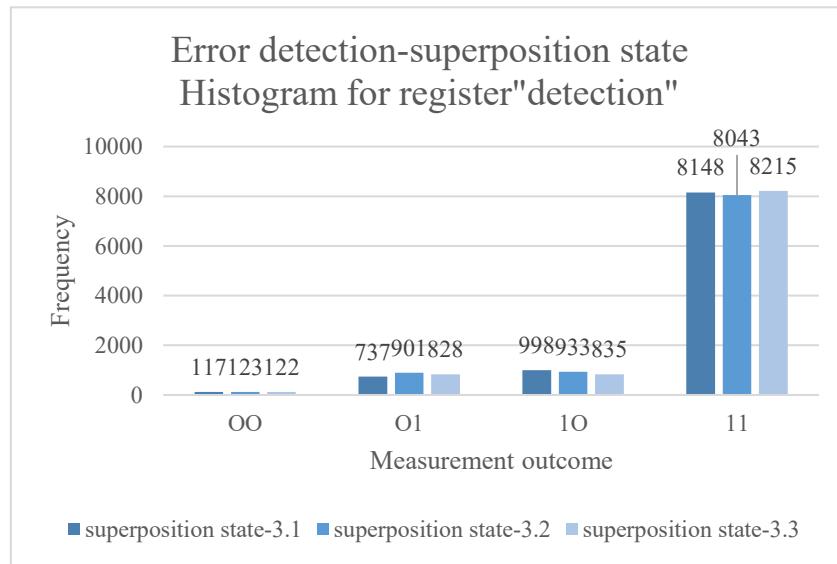
七、真實量子電腦偵錯結果探討



圖八：位元翻轉錯誤經偵錯後之運算結果（研究者自製）

理想：位元錯誤位置為 $q[0]$ ，偵錯碼為 11。

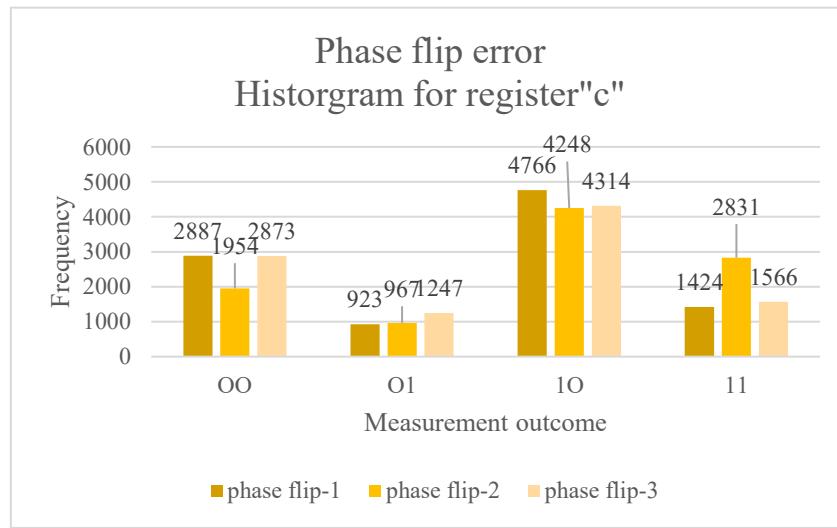
實際：正確率最高為 90.33%，最低僅 67.57%。



圖九：位元翻轉錯誤（疊加態）經偵錯後之運算結果（研究者自製）

理想：位元錯誤位置為 $q[0]$ ，偵錯碼為 11。

實際：正確率最高僅 82.15%。



圖十：相位翻轉錯誤經偵錯後之運算結果（研究者自製）

理想：位元錯誤位置為 $q[2]$ ，偵錯碼為 10。

實際：正確率最高僅 47.66%。

由圖八、圖九和圖十可知，雖然經過多次實驗降低機率波動的影響，但是仍然出現許多不同於我們所期望的結果，因此我們著手尋找讓量子電腦出現錯誤的原因。

八、噪聲模型（Noise Model）

(一)、初探噪聲模型

除了我們人工加入的錯誤（噪聲），真實的 IBM 機器似乎還具有不少其他的噪聲。但我們無法直接觀測量子電腦內部的實際運作，也難以全面掌握錯誤產生的根本機制。因此，我們利用 Qiskit 提供的噪聲模組（Noise Model），在模擬環境中構建噪聲模型，並透過數據觀察與分析，預測噪聲與量子運算中的錯誤行為的關連，為理解與修正量子錯誤提供依據與支撐。

(二)、Noise Model 簡介

Qiskit 的 NoiseModel 是一套可組合的噪聲模擬框架，透過設定不同的錯誤模型與參數，可以在模擬器中營造接近真實環境的噪聲條件，進而分析噪聲對量子電路行為的影響。

NoiseModel 主要針對量子電腦中多種錯誤來源進行模擬，而 Quantum gate errors、Readout errors 以及 Reset errors 是其中三種關鍵的錯誤來源。這三類錯誤涵蓋了量子電路在執行過程中不同階段可能遭遇的噪聲影響，能夠較完整的呈現真實硬體環境下的錯誤行為。

1. Quantum gate errors

發生在量子閘操作階段的錯誤。理想情況下，量子閘應該能精準控制 qubit 的狀態，但實際上由於硬體限制與環境干擾（如電磁波噪聲、控制脈衝不穩定），每次量子閘操作後量子位元的狀態都可能偏離預期軌跡。此類錯誤通常會在電路中隨量子閘的數量逐步累積，對運算正確性造成影響。

2. Readout errors

發生在量子電路執行結束後的量測（readout）階段。量測量子位元狀態時，需將量子態轉換為經典訊號，這一過程中易受到放大器噪聲、探測靈敏度不足等因素影響，導致量測結果與量子位元狀態不一致。

3. Reset errors

針對量子電路中量子位元重置（reset）操作的錯誤。在許多演算法中，為提

升量子位元資源利用率或進行多輪運算，常需將量子位元重置回初始狀態（通常為 $|0\rangle$ ）以供後續使用。然而實際硬體中的重置操作往往無法完全重置回理想狀態，進而影響後續運算過程。

4. Depolarizing error (去極化錯誤)

一種常見且簡單的模型，與其他只針對特定錯誤的模型不同，Depolarizing error 將量子閘分為三種類別，並且隨機將錯誤設置在閘上，無法預測量子位元最終會出現哪一種偏差。由於這種模型設計簡單，能夠概括各種潛在錯誤的整體效果，因此在模擬和測試量子演算法時，常被用作基礎參考，幫助評估演算法在不同噪聲強度下的穩定性與容錯能力。

(三)、簡單電路的設計及功用

為了能更好的說明噪聲模型會造成甚麼效果，我們設計了簡單的量子電路，只有一個 qubit，先用 H 閘把它從初始態轉成疊加態，然後直接做測量。在理想情況下，該電路應該產生 50% 機率的 $|0\rangle$ 及 50% 機率的 $|1\rangle$ ，然而實際硬體上 H gate 是不完美的，我們加入噪聲模擬真實情況。我們針對 H gate 引入 Depolarizing error，使每次執行 H gate 時，有一定機率 qubit 狀態被隨機擾動，而從理想模擬結果與噪聲模擬結果的比較中，可以清楚觀察到噪聲對量子狀態有顯著的影響。

我們的理想模擬結果 0 與 1 分別占比 50.006% 與 49.994%，而 Depolarizing error 的噪聲強度設為 0.5 時，模擬結果 0 與 1 則分別占比 50.21% 與 49.79%，由此可知，加上噪聲影響量子運算的準確率。

(四)、用模擬重現「error detection 電路」在實驗機器上的行為

我們參考 IBM 官方網站的資訊，試著搭建噪聲模型。我們共實驗兩種噪聲模型，第一種噪聲模型是以三組主要參數 p_reset、p_meas 及 p_gate 作為控制變數，分別對應於 Reset errors、Readout errors 與 Quantum gate errors，各參數值代表對應錯誤發生的機率，而第二種噪聲模型是 depolarizing error。研究主要可以分成兩部分，分別是尋找最佳參數與實際模擬。

我們首先載入從 IBM 平台獲取的 Qiskit 程式碼，以及實際量子電腦執行該電路所得到的運算結果，作為預期比較的基準。為量化模擬結果與真實硬體結果之間

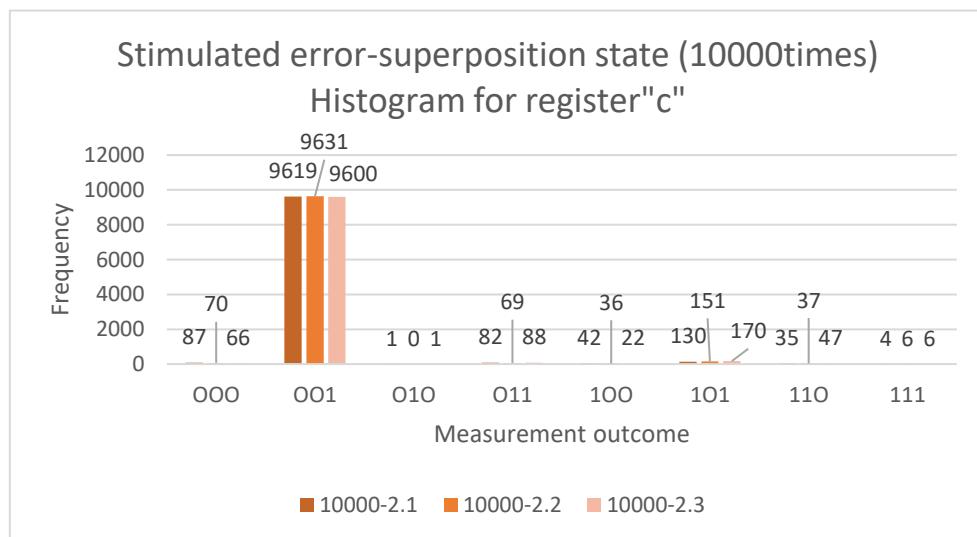
的差異，採用均方誤差（MSE）作為評估指標，均方誤差越小表示模擬結果越接近實驗基準結果。基於此原理，透過程式掃描不同噪聲參數組合，尋找最小均方誤差所對應的最佳參數組合，我們所求的參數估計值均是 50 次平均的結果，並使用該組合生成噪聲模型，將目標量子電路置於此模型下執行模擬。

最終，透過比較噪聲模型模擬結果與真實量子電腦結果的相似度，評估所建構噪聲模型之合理性，並藉由最佳化參數的數值進一步推估潛在錯誤來源及其影響程度。

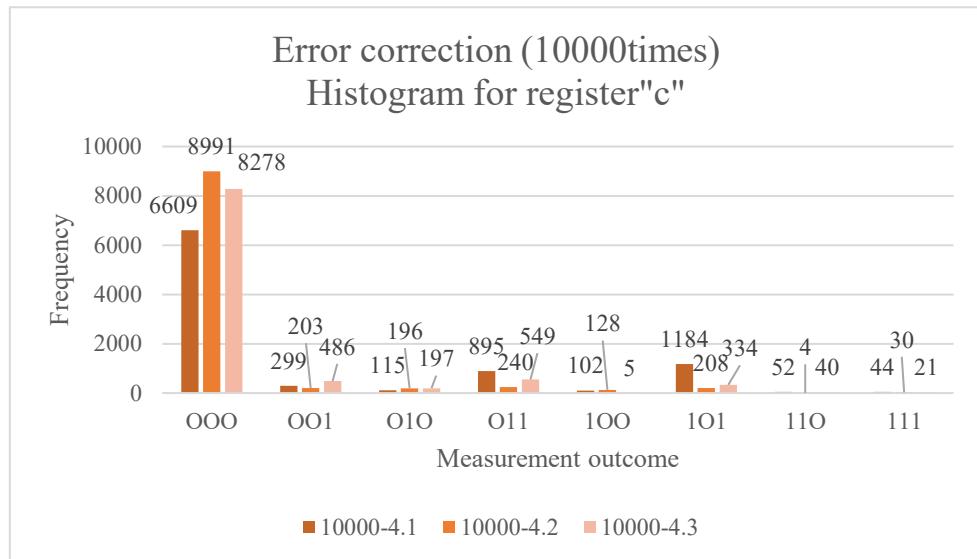
伍、研究結果

一、位元翻轉錯誤（Bit-flip error）。

(一)、初始值為 0



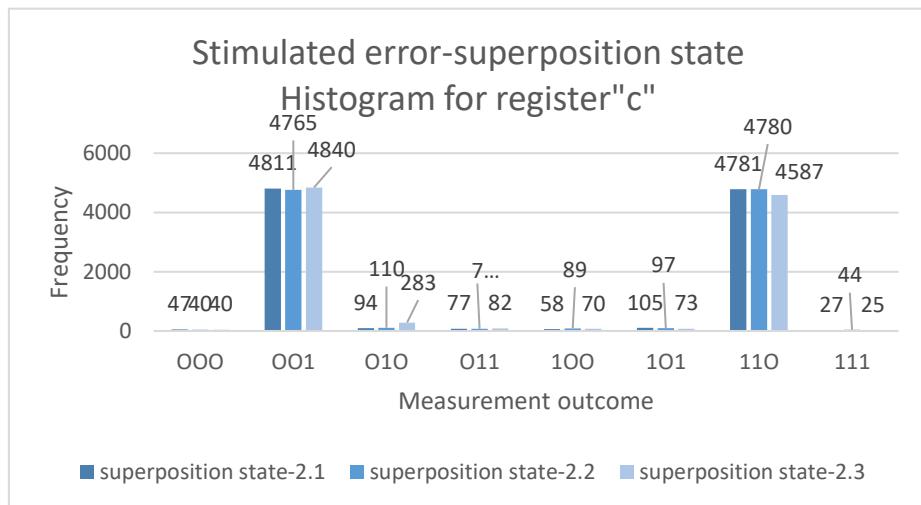
圖十一：模擬出錯之運算結果（研究者自製）



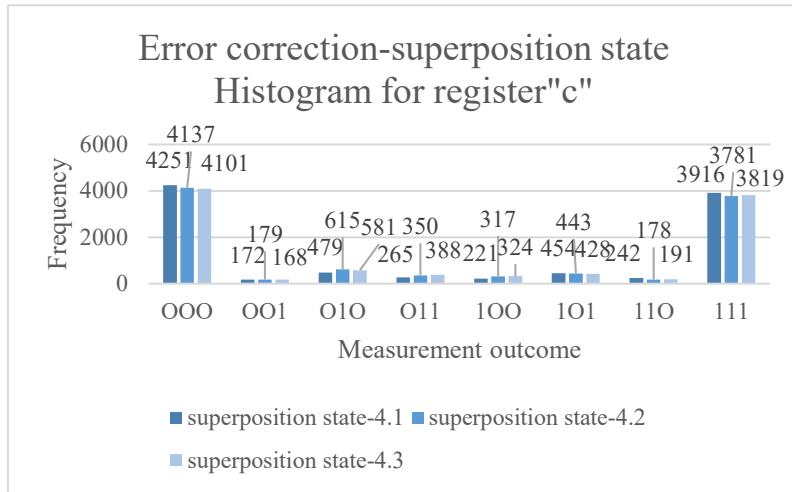
圖十二：位元翻轉錯誤經錯誤修正後之運算結果（研究者自製）

由上圖十一、圖十二可知，我們將模擬出錯後的量子態送入實際量子電腦進行運算，結果顯示在未進行除錯時，系統整體準確率約可達 96%。引入錯誤更正碼後，雖有效降低了錯誤發生率，驗證了錯誤更正機制的成效，卻也使最終的運算準確率有顯著下降。

(二)、利用 H 開做出疊加態



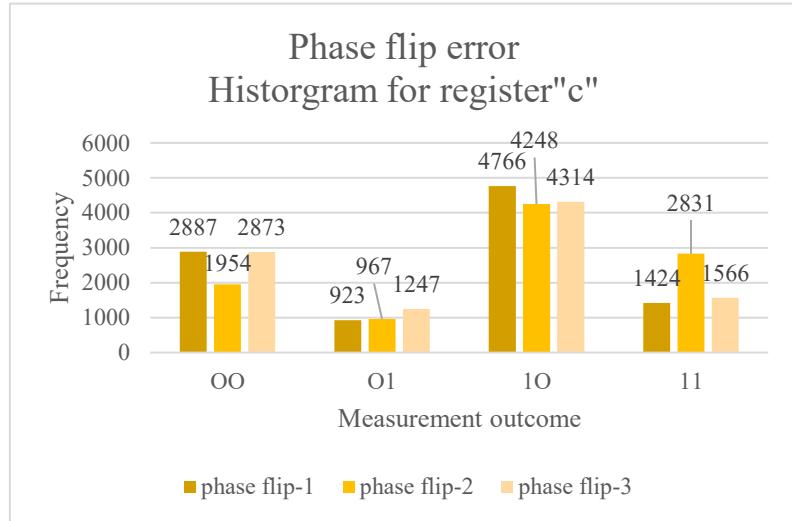
圖十三：模擬出錯（疊加態）之運算結果（研究者自製）



圖十四：位元翻轉錯誤（疊加態）經錯誤修正後之運算結果（研究者自製）

當我們在電路中加入 H 閘運算後觀察結果發現，於模擬錯誤情境下，真實量子電腦運算所得之正確率約為 95%，相較於未加 H 閘時略有下降。進一步施加錯誤更正碼後，雖有效降低錯誤發生率，但整體準確率仍下降至約 80%。

二、相位翻轉錯誤（Phase-flip error）



圖十五：相位翻轉錯誤經偵錯後之運算結果（研究者自製）

我們利用 Z 閘模擬相位翻轉（Phase-flip）錯誤。依據電路設計與理論分析，透過 H 閘轉換，可將 Phase-flip error 轉化為類似 Bit-flip error 的形式進行處理，從而套用相似的錯誤更正策略。根據 IBM 介面顯示之電路圖及模擬結果，當在 q[2] 施加 Z 閘模擬錯誤時，理想情況下偵錯電路應回應 $|10\rangle$ 量子態。實際於真實量子電腦執行實驗時，觀察結果確實以 $|10\rangle$ 為主要輸出態，約佔 44%；然而，其他三種態的出現比例亦相當明顯。

三、Qiskit 實驗結果

(一)、Pauli 錯誤模型

1. Phase-flip error

	Qiskit 模擬數據	真實量子電腦數據
參數	$p_{\text{reset}} = 0.17694$; $p_{\text{meas}} = 0.1701$; $p_{\text{gate1}} = 0.0186$	
直方 數據 統計 圖	<p>圖十八：Pauli 錯誤模型模擬 Phase-flip error 運算結果（研究者自製）</p>	<p>圖十九：Phase-flip error 真實量子電腦運算結果（研究者自製）</p>

2. Bit-flip error (初始為 0)

	Qiskit 模擬數據	真實量子電腦數據
參數	$p_{\text{reset}} = 0.01$; $p_{\text{meas}} = 0.01$; $p_{\text{gate1}} = 0.01$	
直方 數據 統計 圖	<p>圖二十：Pauli 錯誤模型模擬 Bit-flip error 運算結果（研究者自製）</p>	<p>圖二十一：Bit-flip error 真實量子電腦運算結果（研究者自製）</p>

3. Bit-flip error (放 H 閘)

	Qiskit 模擬數據	真實量子電腦數據																				
參數	$p_{\text{reset}} = 0.02472$; $p_{\text{meas}} = 0.22384$; $p_{\text{gate1}} = 0.01$																					
直方 數據 統計 圖	<table border="1"> <thead> <tr> <th>State</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>$0\rangle$</td> <td>755</td> </tr> <tr> <td>$1\rangle$</td> <td>1984</td> </tr> <tr> <td>$0\rangle 1\rangle$</td> <td>1939</td> </tr> <tr> <td>$1\rangle 0\rangle$</td> <td>5322</td> </tr> </tbody> </table>	State	Count	$ 0\rangle$	755	$ 1\rangle$	1984	$ 0\rangle 1\rangle$	1939	$ 1\rangle 0\rangle$	5322	<table border="1"> <thead> <tr> <th>Outcome</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>117</td> </tr> <tr> <td>01</td> <td>737</td> </tr> <tr> <td>10</td> <td>998</td> </tr> <tr> <td>11</td> <td>8148</td> </tr> </tbody> </table>	Outcome	Count	00	117	01	737	10	998	11	8148
State	Count																					
$ 0\rangle$	755																					
$ 1\rangle$	1984																					
$ 0\rangle 1\rangle$	1939																					
$ 1\rangle 0\rangle$	5322																					
Outcome	Count																					
00	117																					
01	737																					
10	998																					
11	8148																					

(二)、Depolarizing error 模型

1. Phase-flip error

	Qiskit �模擬數據	真實量子電腦數據																				
參數	$\text{depolarizing_error}=0.05$																					
直方 數據 統計 圖	<table border="1"> <thead> <tr> <th>State</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>$0\rangle$</td> <td>859</td> </tr> <tr> <td>$1\rangle$</td> <td>249</td> </tr> <tr> <td>$0\rangle 1\rangle$</td> <td>8056</td> </tr> <tr> <td>$1\rangle 0\rangle$</td> <td>836</td> </tr> </tbody> </table>	State	Count	$ 0\rangle$	859	$ 1\rangle$	249	$ 0\rangle 1\rangle$	8056	$ 1\rangle 0\rangle$	836	<table border="1"> <thead> <tr> <th>Outcome</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>2887</td> </tr> <tr> <td>01</td> <td>923</td> </tr> <tr> <td>10</td> <td>4766</td> </tr> <tr> <td>11</td> <td>1424</td> </tr> </tbody> </table>	Outcome	Count	00	2887	01	923	10	4766	11	1424
State	Count																					
$ 0\rangle$	859																					
$ 1\rangle$	249																					
$ 0\rangle 1\rangle$	8056																					
$ 1\rangle 0\rangle$	836																					
Outcome	Count																					
00	2887																					
01	923																					
10	4766																					
11	1424																					

2. Bit-flip error (初始為 0)

	Qiskit 模擬數據	真實量子電腦數據																				
參數	depolarizing_error=0.05																					
直方 數據 統計 圖	<table border="1"> <thead> <tr> <th>Bit String</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>252</td> </tr> <tr> <td>01</td> <td>242</td> </tr> <tr> <td>10</td> <td>259</td> </tr> <tr> <td>11</td> <td>9247</td> </tr> </tbody> </table> <p>圖二十五：depolarizing error 模型模擬 Bit-flip error 運算結果（研究者自製）</p>	Bit String	Count	00	252	01	242	10	259	11	9247	<table border="1"> <thead> <tr> <th>Bit String</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>362</td> </tr> <tr> <td>01</td> <td>379</td> </tr> <tr> <td>10</td> <td>226</td> </tr> <tr> <td>11</td> <td>9033</td> </tr> </tbody> </table> <p>圖二十一：Bit-flip error 真實量子電腦運算結果（研究者自製）</p>	Bit String	Count	00	362	01	379	10	226	11	9033
Bit String	Count																					
00	252																					
01	242																					
10	259																					
11	9247																					
Bit String	Count																					
00	362																					
01	379																					
10	226																					
11	9033																					

3. Bit-flip error (放 H 閘)

	Qiskit 模擬數據	真實量子電腦數據																				
參數	depolarizing_error=0.05																					
直方 數據 統計 圖	<table border="1"> <thead> <tr> <th>Bit String</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>272</td> </tr> <tr> <td>01</td> <td>228</td> </tr> <tr> <td>10</td> <td>248</td> </tr> <tr> <td>11</td> <td>9252</td> </tr> </tbody> </table> <p>圖二十六：depolarizing error 模型模擬 Bit-flip error (放 H 閘) 運算結果（研究者自製）</p>	Bit String	Count	00	272	01	228	10	248	11	9252	<table border="1"> <thead> <tr> <th>Bit String</th> <th>Count</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>117</td> </tr> <tr> <td>01</td> <td>737</td> </tr> <tr> <td>10</td> <td>998</td> </tr> <tr> <td>11</td> <td>8148</td> </tr> </tbody> </table> <p>圖二十三：Bit-flip error (放 H 閘) 真實量子電腦運算結果（研究者自製）</p>	Bit String	Count	00	117	01	737	10	998	11	8148
Bit String	Count																					
00	272																					
01	228																					
10	248																					
11	9252																					
Bit String	Count																					
00	117																					
01	737																					
10	998																					
11	8148																					

陸、討論

一、錯誤更正碼實驗

我們首先以量子初始的狀態(初始值為 0)進行實驗，並將模擬出錯後的量子態送入真實量子電腦進行運算。實驗結果顯示，在未進行錯誤更正的情況下，系統整體準確率約為 96%；而引入錯誤更正碼後，雖然有效降低了錯誤發生率，驗證了錯誤更正機制具備實質效果，然而最終整體運算準確率卻顯著下降。推測此現象主要源自於為實現錯誤更正所需額外增加的量子閘操作，進而提升了電路對環境噪聲與不可預期干擾的敏感性。

當我們於電路中加入 H 閘後進行觀測，發現於模擬錯誤情境下，真實量子電腦的運算正確率約為 95%，相較於未加 H 閘時略有下降。在進一步引入錯誤更正碼後，雖然成功降低了錯誤發生率，但整體運算準確率卻進一步下降至約 80%。此結果延續了先前實驗所觀察到的趨勢——量子閘數量的增加對實驗結果具有顯著影響，特別是在真實硬體環境中，較多的閘操作會引入額外噪聲與干擾，進而影響計算精度。這一現象再次突顯，目前量子硬體中，量子閘數量與計算準確性之間仍存在明顯的權衡關係。

此外，我們亦觀察到，真實量子硬體的運算表現受到目前尚難完全預測或建模的背景噪聲與動態干擾所限制，導致與理想模擬結果之間出現明顯差距，顯示在 blt flip error 偵測電路中，推測因為電路結構相對簡單，使用的閘較少，故兩者模型皆具備了解釋數據的能力。

二、噪聲模型實驗

在 Phase-flip 錯誤的模擬中，使用 Pauli 錯誤模型所擬合出的結果與真實量子電腦運算結果高度相似。呈現最佳擬合狀態時，錯誤模型的三個參數分別為 $p_{\text{reset}} = 0.17694$ 、 $p_{\text{meas}} = 0.1701$ 、 $p_{\text{gate1}} = 0.0186$ 。由此可推測，Phase-flip 錯誤較有可能主要發生於量子位元的重置階段與測量階段，而在量子閘操作中的影響則相對較小。

另一方面，從圖中可觀察到，若僅套用 Depolarizing error 模型，無法良好擬

合實驗數據，說明單純依賴 Depolarizing 模型難以充分描述 IBM 機器中實際主導 Phase-flip 行為的噪聲來源。

兩相比較，前述的 Pauli 錯誤模型能更有效擬合實驗結果，故可初步推論，在目前實驗所設計之 Phase-flip error detection 電路中，IBM 機器受環境影響的行為，主要受到三個參數模型中的噪聲主導，單純使用 Depolarizing 模型無法完整捕捉此特性，需採用更具針對性的錯誤模型進行描述與分析。

而在 bit-flip 錯誤模擬中，我們分別測試了以初始態為 $|0\rangle$ 以及加入 H gate 產生疊加態兩種不同的輸入設定。從實驗結果可觀察到，加入 H gate 的電路，其模擬結果與實驗預期之間的差距較大。造成此差異的原因之一，是 H gate 本身在模擬時同樣會受到噪聲模型的作用；而在實際量子電路中，每增加一個量子閘操作都會引入額外的損失與噪聲，因此透過 H gate 轉換後的輸出，噪聲影響也相對增加。

此外，在 bit-flip 電路的模擬中，即使僅使用單一類型的 Depolarizing error 模型，亦能達到相當良好的擬合效果；而採用三參數噪聲模型也能大致展現出良好的擬合結果，顯示兩種模型皆具備解釋實驗數據的能力。

三、綜合考量與未來展望

在量子計算的研究與實驗中，真實硬體不可避免會受到雜訊（Noise）和錯誤（Error）的影響。利用 Qiskit 提供的噪聲模型，我們得以模擬實驗環境中各種可能的干擾，從而探索如何藉由理論方法使模擬結果更貼近真實實驗現象。這種「用理論接近實驗」的嘗試，讓我們能夠在理論與實踐間架起一道橋樑，對量子錯誤更正與量子模擬有了更全面的認識。

在這段探索過程中，我們首先展示了量子錯誤更正和噪聲模型的完整流程。透過 Qiskit 的各項功能，我們能夠依據硬體特性和噪聲參數建立符合實際情況的模擬環境。例如，利用機率波動、閘操作中產生的隨機錯誤以及其他物理現象，可以精準地模擬量子比特在實際運作時所受的干擾。這個過程中，我們體認到雖然模型並非唯一，但一個設計良好的噪聲模型能夠幫助我們更好理解背後隱藏的物理機

制，並為實驗數據提供合理的解釋與預測。

同時，我們在進行噪聲模型分析時，也深刻體會到理論與實驗之間的微妙平衡。透過這些模擬，我們學到的不僅是理論公式和流程，更認識到在實際系統中，每一個微小的參數變化都可能對最終結果產生驚人的影響。這種從模擬中獲得的啟發，正是推動我們持續改進量子演算法與錯誤更正技術的重要動力。

1. 探索更多噪聲類型與通道

除了基本的衰變和位相錯誤外，未來可以進一步考慮其他更複雜的噪聲來源，如交叉耦合、非高斯噪聲及環境干擾等。這將使得所建立的噪聲模型更貼近真實硬體的行為，並有助於設計出針對性的錯誤更正策略。

2. 自動化 fit 與參數優化

在目前的手動調整中，我們已經能夠獲得相對合理的模擬效果；然而，未來可採用自動化工具以机器學習或優化算法來自動調整噪聲參數，使得模型能夠根據實驗數據自我校正，進一步縮小模擬與實驗之間的落差。這種自動化流程將大大提高模擬效率，並幫助我們及時捕捉硬體變動帶來的影響。

3. 綜合以上心得，利用 Qiskit 的噪聲模型不僅增強了我們對量子錯誤更正理論的理解，也促進了從模擬到實驗的無縫銜接。未來的發展將依賴於更精細的噪聲分析、自動化參數 fit 技術以及與錯誤抑制策略的結合。這一切不僅預示著量子計算領域的挑戰，更展現出一個充滿潛力與創新的未來局面。每一個進步都將帶來對量子世界更深層次的洞察，並促使我們在理論與實踐的交界上，不斷追求更高的精度和更穩定的運算結果。

4. 未來，或許我們可以探討如何利用最新的機器學習技術來預測噪聲變化，或進一步將這些模擬結果與實際量子硬體動態結合，形成一個閉環系統，獲得更即時、精準的反饋與校正。這樣的跨界嘗試與技術創新，無疑是推動量子計算走向成熟的重要一步。

柒、結論

一、總結

本研究以 Qiskit 為工具，結合錯誤更正碼設計與噪聲模型模擬，系統性探討了當前真實量子電腦中錯誤行為的特性與影響，並嘗試驗證不同錯誤更正與噪聲建模策略的適用性。透過多組實驗結果，我們觀察到量子閘數量與計算準確性間存在明顯權衡，特別是在引入錯誤更正碼後，儘管邏輯錯誤率有所降低，但額外增加的量子閘操作也使電路對噪聲與環境干擾更加敏感，整體運算準確率反而下降。

此外，針對不同錯誤類型的噪聲建模結果亦呈現出差異性。對於 Phase-flip 錯誤，單一 Depolarizing 模型難以完整描述實際硬體中的複雜噪聲行為，需藉由更細緻的 Pauli 錯誤模型方能較準確擬合實驗觀察；而針對 Bit-flip 錯誤，Depolarizing 模型則已能夠合理擬合出與實驗數據相符的結果，顯示此類錯誤在目前實驗條件下，其噪聲來源較為單純且均勻。

二、實驗省思

綜合而言，理論模擬結果與真實實驗間仍存在一定差距，主要原因之一在於，當前模擬中所採用的噪聲模型雖可近似描述部分常見錯誤來源，但實際量子硬體中仍存在許多難以完全預測或量化的背景噪聲與瞬態干擾；而量子位元本身又極度敏感，易受到環境中微小變化的影響，導致運算結果出現不可忽視的誤差。這一現象提示，未來量子錯誤更正技術的發展，不僅需要改良演算法本身，更需配合更高精度、更能即時反映硬體狀態的噪聲模型，共同提升量子計算的可靠性。

三、實驗擴展

展望未來，隨著量子硬體技術與模擬工具的進步，本研究亦揭示數個潛在的發展方向，包括探索更多元化噪聲來源、導入機器學習技術進行噪聲參數自動擬合，以及發展能即時回饋硬體狀態的閉環優化架構。這些創新將有望縮小理論模擬與實驗結果之間的差距，並推動量子錯誤更正與量子演算法設計朝向更高效、更穩定的方向邁進。

最終，我們期望藉由這類跨理論與實作的系統化研究，不僅能深化對量子計算中噪

聲與錯誤行為的理解，亦為未來大規模、實用化量子運算鋪設更穩固的基礎。

捌、參考資料及其他

- 一、Oxford University Press. (n.d.). An Introduction to Quantum Computing.
- 二、IT Support Guides. (n.d.). What is a bit flip? Causes, consequences and prevention.
- 三、Science ABC. (n.d.). What are bit flips and how are spacecraft protected from them?
- 四、QubitLab. (n.d.). Chapter 03.
- 五、Nielsen, M. A., & Chuang, I. L. (2000). Quantum computation and quantum information. Cambridge University Press.
- 六、Preskill, J. (1998). Lecture notes for physics 229: Quantum information and computation. California Institute of Technology.
- 七、Haroche, S., & Raimond, J.-M. (2006). Exploring the quantum: Atoms, cavities, and photons. Oxford University Press.
- 八、DiVincenzo, D. P. (2000). The physical implementation of quantum computation. *Fortschritte der Physik*, 48, 771–783

【評語】051814

本作品探討量子計算中由噪聲引發的位元翻轉錯誤與相位翻轉錯誤解決方法議題，研究執行使用 IBM Quantum 平台的 Qiskit 工具，在基本運算中，結合錯誤修正設計模型，驗證錯誤修正策略於降噪運算的適用性能。研究者理解量子資訊的理論基礎，並能應用開源工具進行實驗，顯示其具有編程與資訊分析能力，且具有科學探索能力與精神。

理論上，引入錯誤更正機制，可降低邏輯錯誤，但實驗結果，顯示可能錯誤更正電路本身導入額外噪聲，建議可進一步評估量子電路深度與閘操作數對總體運算精度的影響，嘗試設計更簡化的除錯架構以減低額外誤差，以提升模型精度與可擴展性。

作品海報

真實量子運算中的錯誤與修復

QISKit的噪聲模型與錯誤更正實驗



摘要

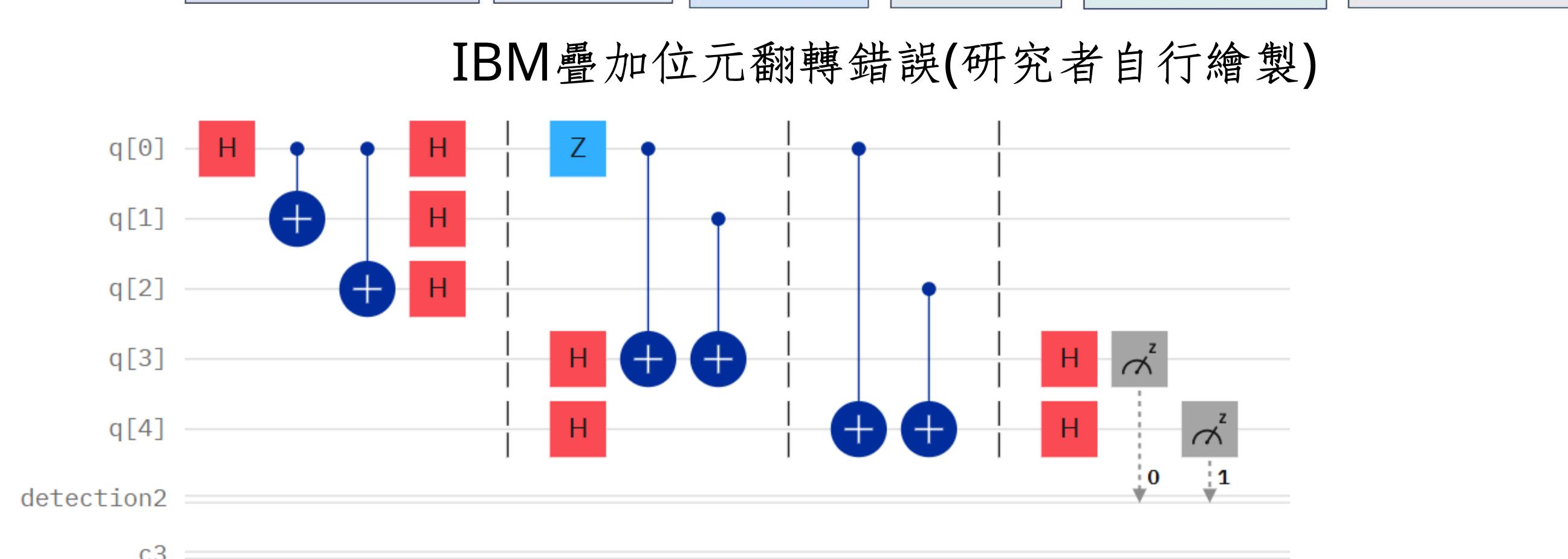
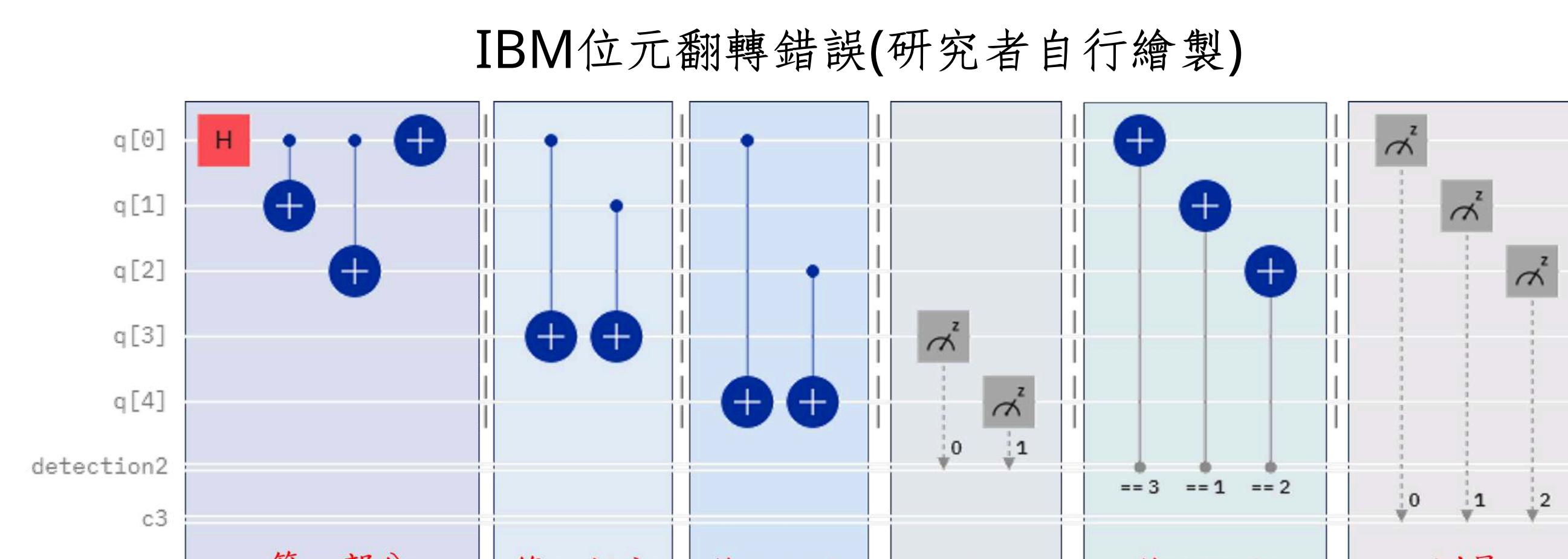
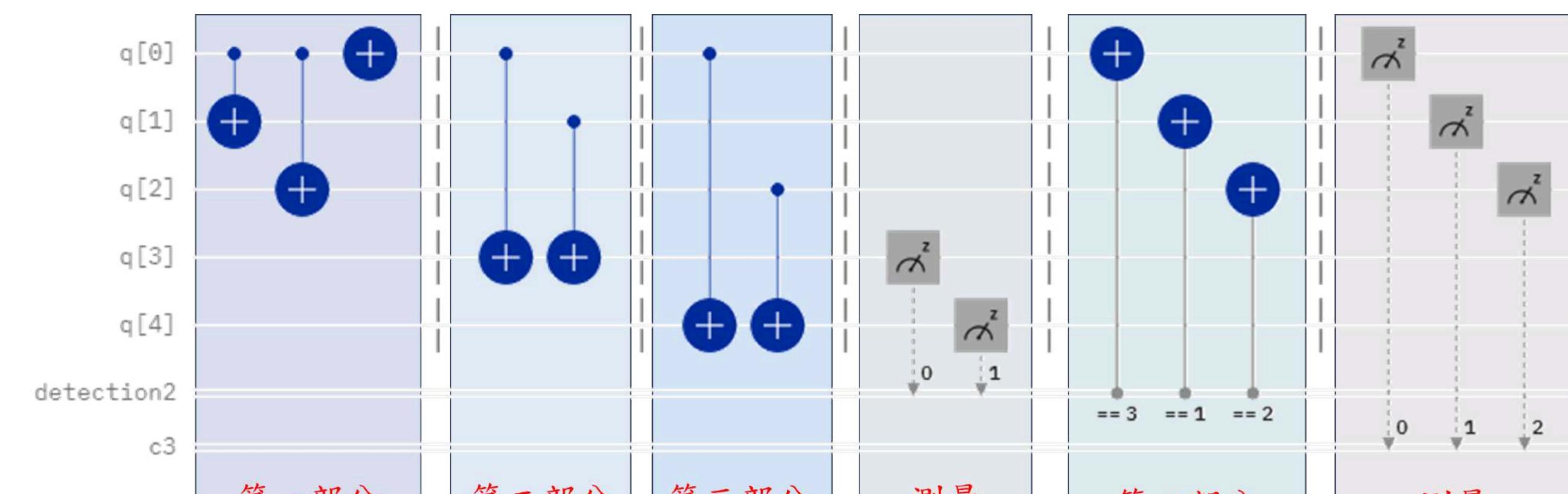
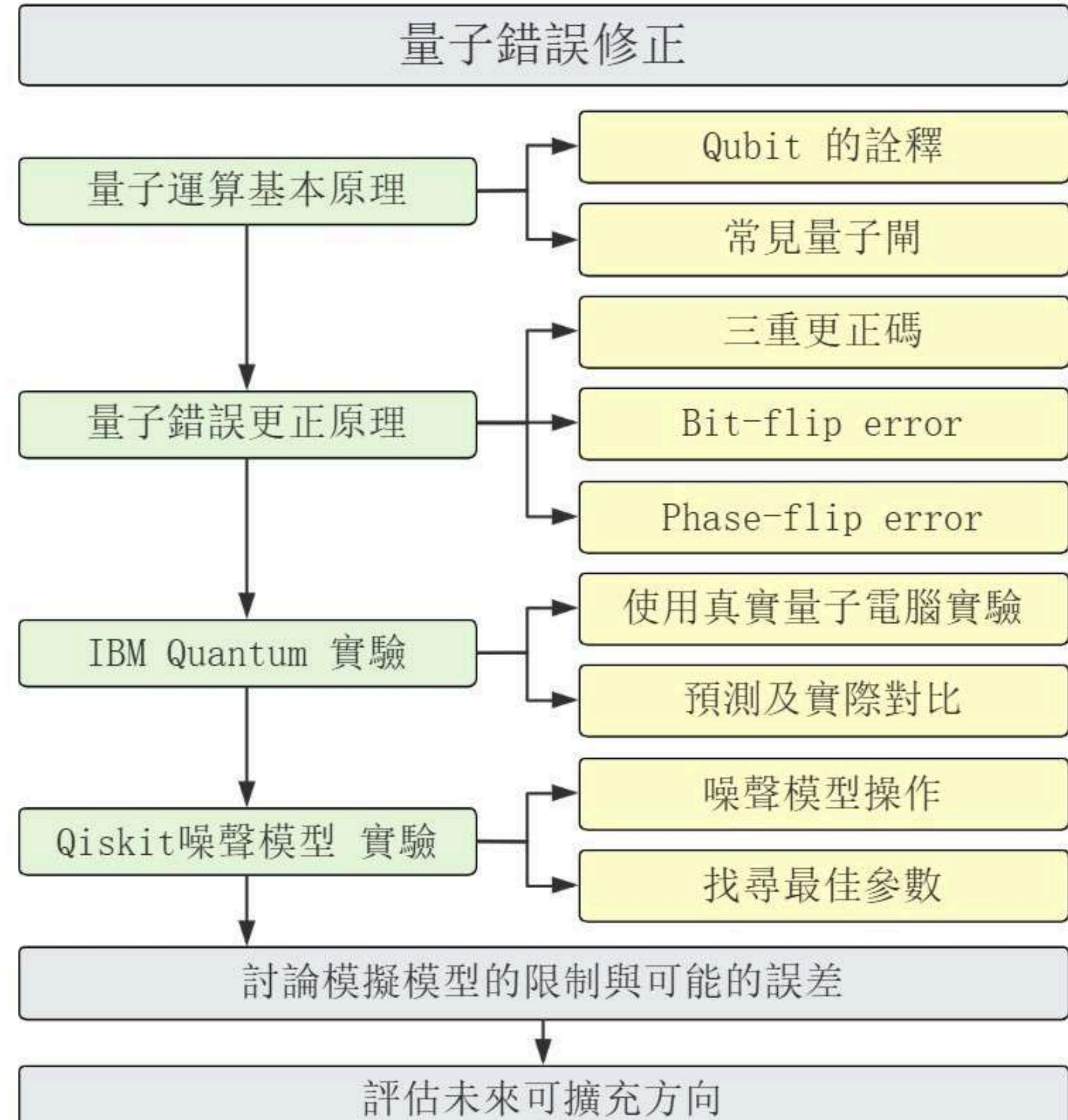
本研究首先探討量子計算中由噪聲引發的兩種主要錯誤：位元翻轉錯誤（Bit-flip Error）與相位翻轉錯誤（Phase-flip Error）。利用 IBM Quantum 平台，透過即時模擬與圖像化操作，建構了一套重複更正碼系統，展示量子糾纏在錯誤校正中的應用，並且確認我們的錯誤更正碼是有效的。不過從實驗結果中，我們發現出現了額外的噪聲干擾使得更正的正確率受到影響。因此，我們結合在真實量子電腦上獲得的結果，使用 Qiskit 試著尋找最接近真實情況的噪聲模型，並分析其中的量子行為。

研究動機與目的

量子電腦被認為是未來運算的主力，然而目前的技術僅能同時操控少數量子位元，且極易受到外界噪聲的干擾，這使得量子電腦尚未能廣泛應用於各大產業。因此，我們著手研究量子電腦的錯誤機制，試圖克服這一量子發展的瓶頸。

- 一、實作錯誤更正碼
- 二、建構並優化噪聲模型
- 三、分析量子錯誤行為機制

研究過程與方法



X閘圖示、矩陣表示、改變之量子態

(圖) 標示		輸入	輸出
		$ 0\rangle$	$ 1\rangle$
矩陣表示	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$ 1\rangle$	$ 0\rangle$
		$\alpha 0\rangle + \beta 1\rangle$	$\alpha 1\rangle + \beta 0\rangle$

(研究者自行繪製)

H閘圖示、矩陣表示、改變之量子態

(圖) 標示		輸入	輸出	布洛赫球面表示
矩陣表示	$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	任意狀態	$\frac{1}{2} 1\rangle + \frac{1}{2} 0\rangle$	

(研究者自行繪製)

CNOT閘圖示、矩陣表示、改變之量子態 (研究者自行繪製)

(圖) 標示		輸入		輸出	
		$ A\rangle$	$ B\rangle$	$ A\rangle$	$ A \oplus B\rangle$
矩陣表示	$CNOT$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
		$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
		$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 0\rangle$
		$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

(研究者自行繪製)

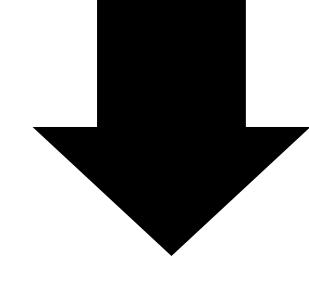
Z閘圖示、矩陣表示、改變之量子態

(圖) 標示		輸入	輸出	布洛赫球面表示
矩陣表示	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$ 1\rangle$	$- 1\rangle$	

(研究者自行繪製)

經過IBM上的實驗我們發現：

除了我們人工加入的錯誤，真實的IBM機器似乎還有不少其他的噪聲，影響量子電腦運算準確率

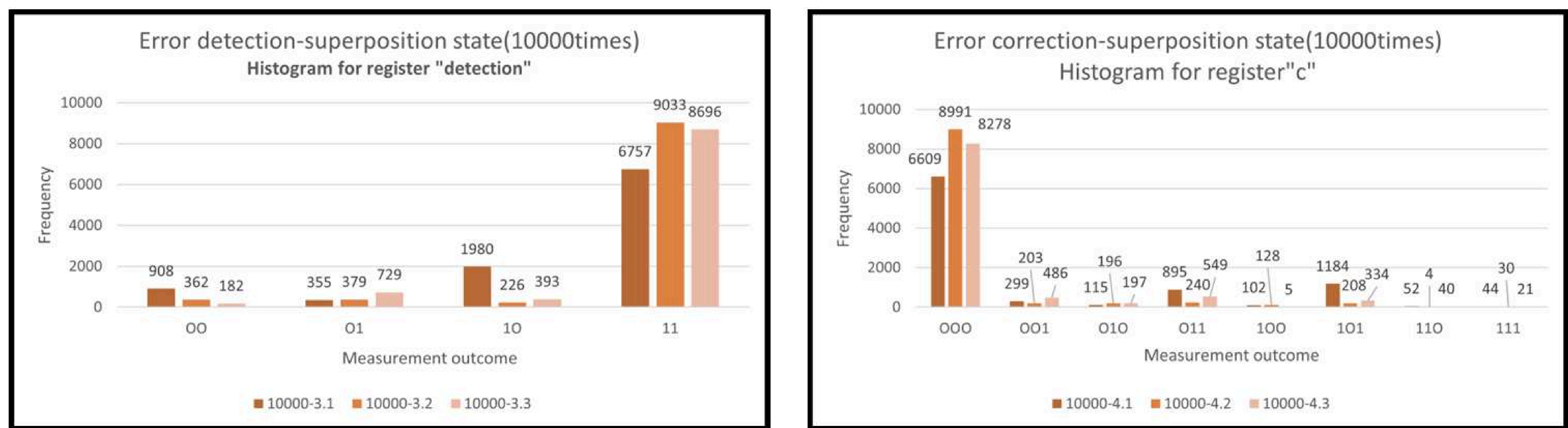


利用均方誤差來找出最接近真實情況的參數組合，建構出噪聲模型：共實驗兩種噪聲模型，第一種噪聲模型是以三組主要參數 p_{reset} 、 p_{meas} 及 p_{gate} 作為控制變數，分別對應於 Reset errors、Readout errors 與 Quantum gate errors，各參數值代表對應錯誤發生的機率，而第二種噪聲模型是 depolarizing error。最終，透過比較噪聲模型模擬結果與真實量子電腦結果的相似度，評估所建構噪聲模型之合理性，並藉由最佳化參數的數值進一步推估潛在錯誤來源及其影響程度。

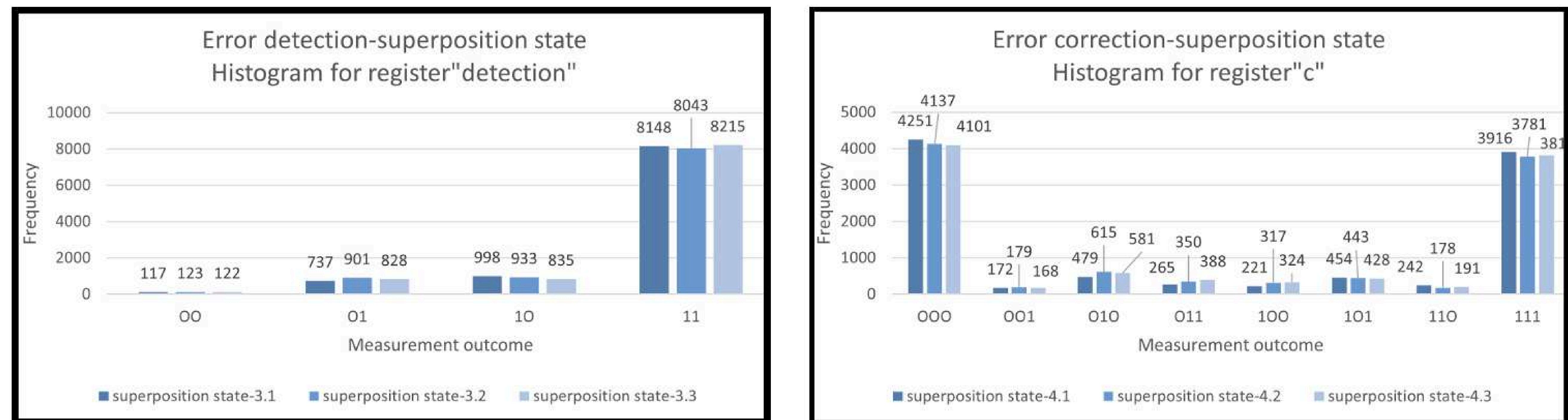
- 1.Depolarizing error**：將量子閘分為三種類型，並且隨機將錯誤設置在閘上，無法預測量子位元最終會出現哪一種偏差。
- 2.Quantum gate error**：每次量子閘操作後量子位元的狀態都可能偏離預期軌跡。此類錯誤會隨量子閘的堆疊而逐步放大，對最終運算結果的精確性產生顯著影響。
- 3.Readout error**：量測量子位元狀態時，需將量子態轉換為經典訊號，這一過程中易受到放大器噪聲、探測靈敏度不足等因素影響，導致量測結果與量子位元狀態不一致。
- 4.Reset error**：量子位元重置時未能準確回到初始態，使得接下來的運算結果受到影響。

實驗結果

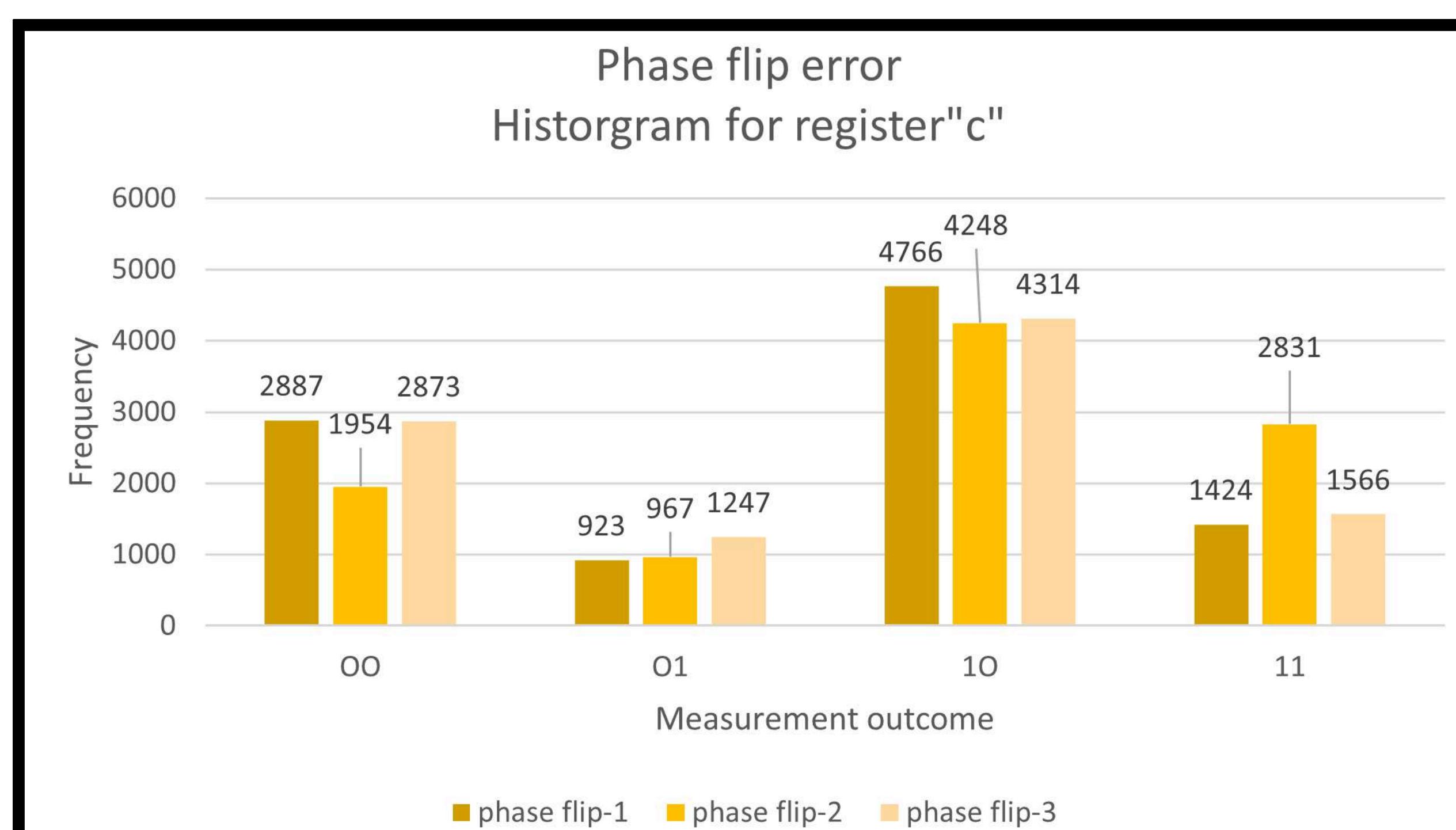
一、IBM位元翻轉錯誤(研究者自行繪製)



二、IBM疊加態位元翻轉錯誤(研究者自行繪製)

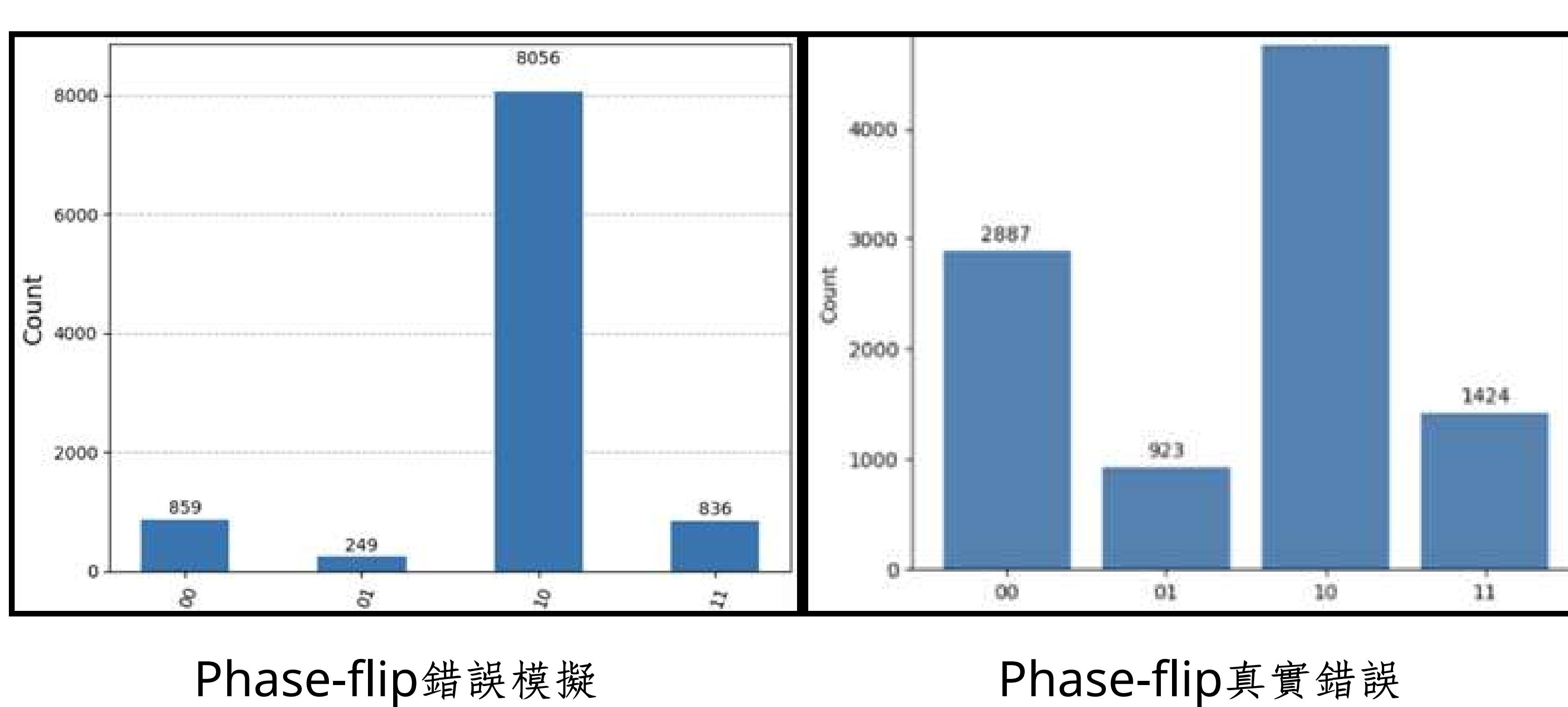


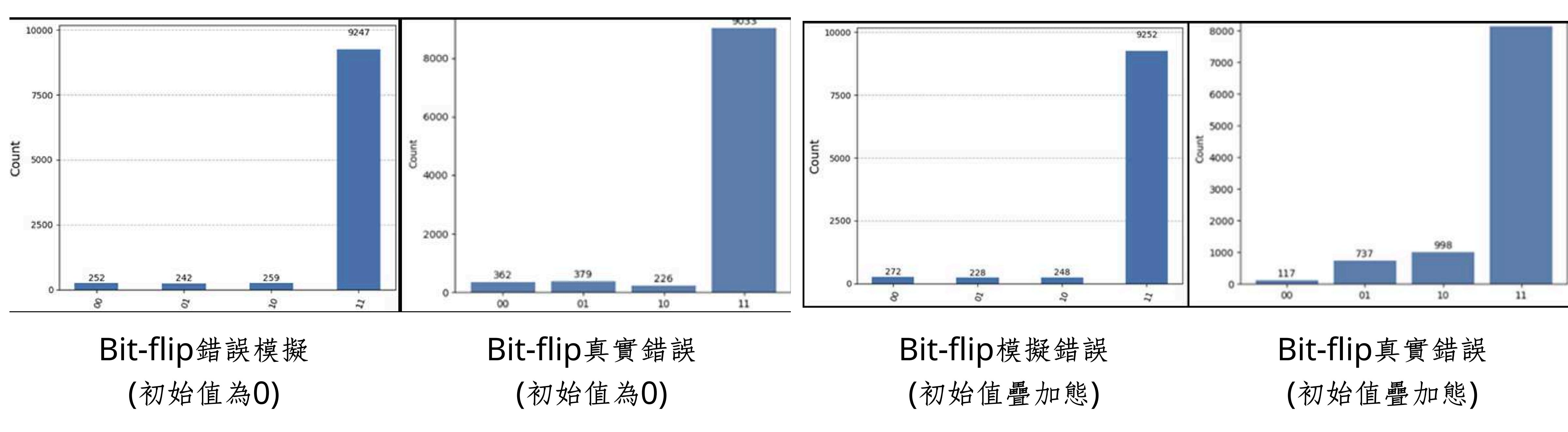
三、IBM相位翻轉錯誤(研究者自行繪製)



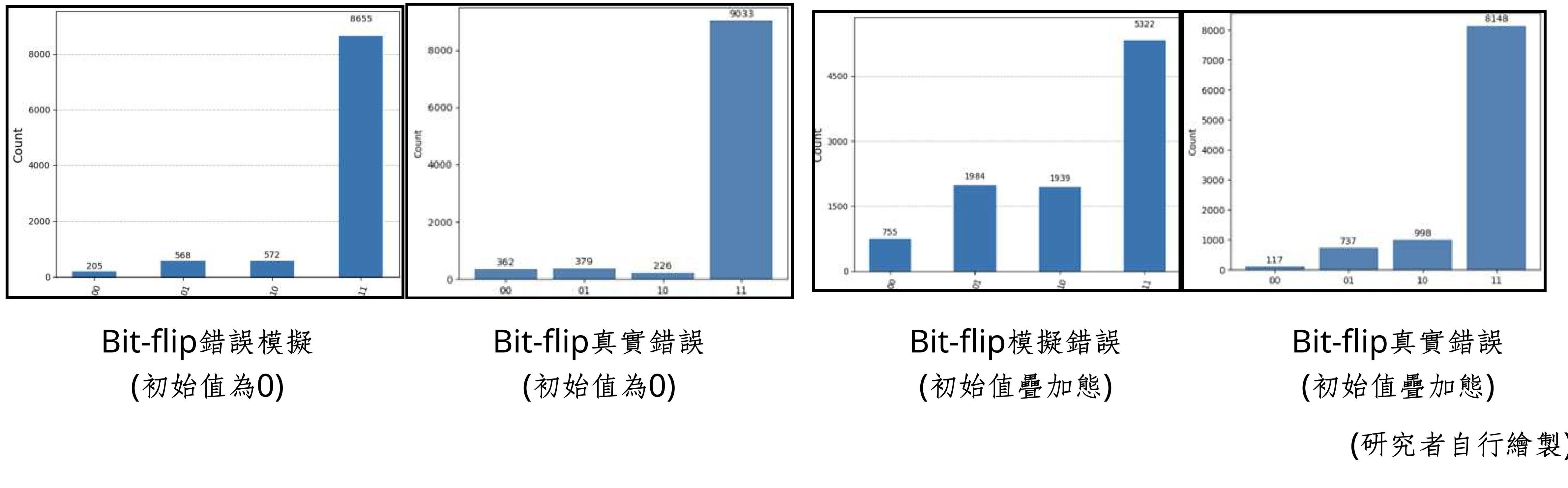
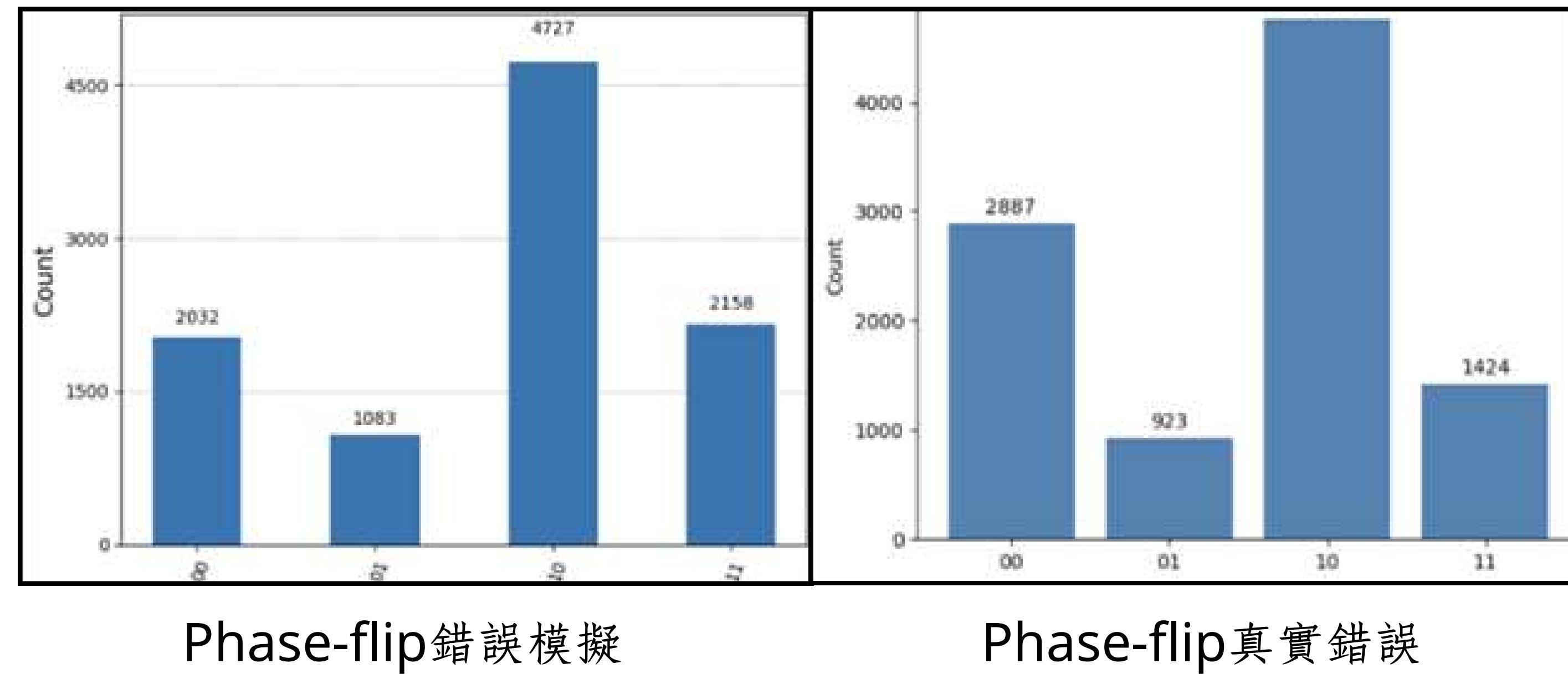
四、Qiskit實驗結果

1.Depolarizing錯誤





2.Pauli錯誤



討論

一、錯誤更正碼實驗：錯誤更正碼能有效修正錯誤，但額外增加的量子閘操作會讓電路更容易受環境干擾，而這些噪聲目前無法預測或消除。

二、噪聲模型實驗：

- 在 Phase-flip 錯誤模擬中，Pauli 錯誤模型能精準擬合實驗結果，顯示錯誤主要來自量子位元的重置與測量階段；相對地，單純的 Depolarizing 模型無法完整解釋真實噪聲行為。
- 在 Bit-flip 模擬中，無論使用 Depolarizing 或 Pauli 模型，都能達到良好擬合，而在加入 H gate 後，噪聲影響更為明顯，凸顯量子閘數量對運算準確度的重大影響。整體顯示 Pauli 模型較能捕捉 Phase-flip 的錯誤特性，Depolarizing 模型則能有效描述 Bit-flip 下的行為。

三、綜合考量與展望

結論

我們成功證明錯誤更正能有效降低 Bit-flip 與 Phase-flip 的錯誤率，展現此方法在真實量子電腦上的實質效用；此外，我們利用 Qiskit 噪聲模型建立了接近真實硬體的模擬，找出錯誤來源並推測其物理機制，雖無法直接修正噪聲，但能用於分析與理解量子錯誤行為。結合錯誤更正與噪聲模型的研究，能幫助全面掌握量子電腦中錯誤的成因與特性，為未來提升量子計算準確度與穩定性奠定基礎。未來將持續探索更多噪聲類型，並結合機器學習自動化優化噪聲參數，發展能即時反映硬體狀態的系統，進一步提升量子運算的穩定性與精確度。

參考文獻

- Oxford University Press. (n.d.). An Introduction to Quantum Computing.
- IT Support Guides. (n.d.). What is a bit flip? Causes, consequences and prevention.
- Science ABC. (n.d.). What are bit flips and how are spacecraft protected from them?
- QubitLab. (n.d.). Chapter 03.
- Nielsen, M. A., & Chuang, I. L. (2000). Quantum computation and quantum information. Cambridge University Press.
- Preskill, J. (1998). Lecture notes for physics 229: Quantum information and computation. California Institute of Technology.
- Haroche, S., & Raimond, J.-M. (2006). Exploring the quantum: Atoms, cavities, and photons. Oxford University Press.
- DiVincenzo, D. P. (2000). The physical implementation of quantum computation. Fortschritte der Physik, 48, 771–783

Cambridge