武漢大學
WUHAN UNIVERSITY

量子计算实验手册

Quantum Computing

Experiment Manual

第一版

王浩冰，张寒子逸，梁超

二〇二一年三月

文档说明

《量子计算》是武汉大学开设的一门计算机专业选修课，由计算机学院的梁超副教授担任授课老师，弘毅学堂的王浩冰、张寒子逸两位同学担任课程助教。

本实验手册是《量子计算》课程配套实验的指导手册，其中助教王浩冰同学负责实验设计和各实验文档的初步编写工作，助教张寒子逸同学负责各实验文档的进一步删改优化和整合排版工作。

我们为《量子计算》课程配套实验撰写了本实验手册、实验配套 PPT、实验参考代码及实验答案文档。在开始所有实验前，学生需要基本熟悉 Circuit Composer 在线编程界面的操作方法，调试 Python 的运行环境；可以通过浏览本手册配套的 PPT 教案进行预备实验，熟悉本实验工具。在结束每个试验后，可以查看每个实验对应的参考代码进一步验证自己实验结果的正确性。

所有本课程相关内容（包括实验），都可通过进入我们的课程网页量子计算课程指南获取。

实验目录

实验 1

制造并观测单量子的叠加态

1.1 实验目的

(1) 深入理解量子比特与量子门的概念；
(2) 从数学角度与应用角度理解 H 门与 X 门的作用；
(3) 熟练掌握量子实验平台的可视化编程方法，了解 QASM 语言与 Qiskit 的基本编程操作与语法。

1.2 实验概述

(1) 在量子实验平台上运用可视化编程实现基本的单量子电路；
(2) 书面计算该量子电路的中间状态和结果；
(3) 将生成的程序代码下载到本地运行并得到仿真结果；
(4) 在量子实验平台上提交程序代码并得到量子计算机运行结果。

1.3 前提知识

1.3.1 量子比特

量子比特（Qubit）是量子信息的物理载体。它是比特的量子版本，它的量子态可以用一个二维向量来表示。对于传统比特最基本的两个状态“0”和“1”，其量子比特的表示方法如下：

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.1}$$

这一组状态，又被称为标准基（Standard Basis ）或 z 基（z Basis）。

IBM 量子实验平台提供的量子计算机中基于超导微电路从物理角度实现了使用

量子比特的计算和应用，感兴趣的同学可以深入 Transmon Qubit[1]的相关原理，这里不再赘述。

在之后的编程实验中，我们将仅从抽象角度（二维复向量角度）来计算和应用量子比特。

1.3.2 量子门

量子门和经典电路中的电路门类似，可以将输入的量子比特调整至其它状态。量子门可以用一个酉矩阵表示，而单比特的量子门就是一个二阶酉矩阵。量子门作用于量子比特，可以用矩阵与向量相乘表示，例如量子门U 作用于量子态为$|\psi\rangle$ 的量子比特，得到新量子态$|\psi'\rangle$，可以表示为：

$$U|\psi\rangle = |\psi'\rangle \tag{1.2}$$

在本次实验中，我们需要用到两个单比特量子门——X 门和 H 门。

其中，X 门可以用矩阵表示为：

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{1.3}$$

X 门像经典电路的非门一样，可以将“0”状态翻转为“1”状态，反之亦然：

$$X|0\rangle = |1\rangle \qquad X|1\rangle = |0\rangle \tag{1.4}$$

H 门可以用矩阵表示为：

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.5}$$

我们将在之后的实验中研究并探讨其作用。

1.4 实验步骤

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为一个，经典比特寄存器的数量调整为一个，这可以通过以下两种方法实现：

  (a) 菜单栏点击“Edit”→“Manage Registers”，在寄存器编辑窗口中“Number of qubits”栏输入 1，点击“Save” 完成，如图1.1所示。

  (b) 在右侧的“Code editor”窗口中的 QASM 代码中将 qreg 和 creg 的值改为 1。

[1]其全称为：Transmission Line Shunted Plasma Osillation Qubit。

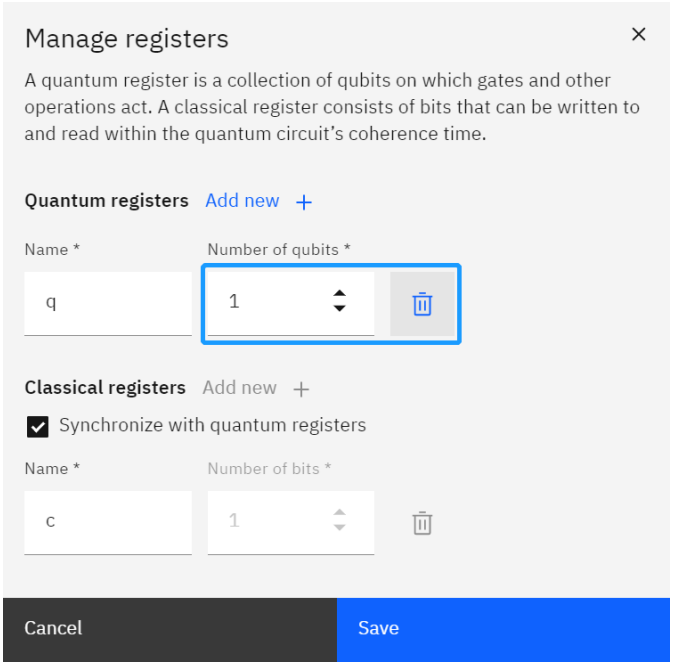


图 1.1: 寄存器编辑窗口

(3) 如图1.2所示，拖动工具栏中的量子门到电路中，搭建单量子比特 H 门电路。该电路包含两个门，分别是一个 H 门和一个观测操作[2]。全部拖动完毕后，右侧的“Code editor”窗口将会自动生成代码。如图1.3所示，在该窗口左上角选择“Qiskit”，可查看如例1.1所示的 Python 代码；选择“OpenQASM 2.0”，可查看如例1.2所示的 QASM 代码。

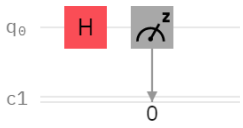


图 1.2: 一个简单的单量子电路

(4) 观察整个界面左下方的状态窗口，在“State vector”选择卡下可以看到通过计算机仿真结果，对应量子比特进行观测之前，其量子态为 [0.707+0j,0.707+0j]，即：

$$|\psi\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{1.6}$$

在“Measurement Probabilities”选择卡下可以看到通过计算机仿真算出的，对该量子比特进行观测之后，观测结果为“0”和“1”的可能性各占 50%。

[2]观测操作为一个形如图1.2中以灰色为底的电路符号，它将对单个量子比特的测量结果写入一个经典比特，是一个不可逆操作，不属于量子门。

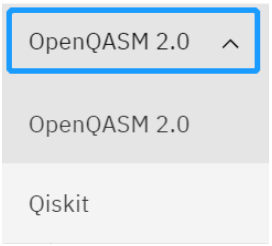


图 1.3: 编程语言选择栏

例 1.1: 自动生成的 Python 代码

```python
from qiskit import QuantumRegister, ClassicalRegister,
    QuantumCircuit
from numpy import pi

qreg_q = QuantumRegister(1, 'q')
creg_c = ClassicalRegister(1, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

circuit.h(qreg_q[0])
circuit.measure(qreg_q[0], creg_c[0])
```

例 1.2: 自动生成的 QASM 代码

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[1];
creg c[1];

h q[0];
measure q[0] -> c[0];
```

(5) 对刚才的仿真结果进行书面计算推演，检查计算结果和仿真结果是否一致。

(6) 在“Code editor”窗口左上角将编程语言调整为“Qiskit”，然后点击右侧的“ExportCode”下载代码到本机，后者位置如图1.4所示。

(7) 打开下载好的代码，在上方 from qiskit import 处添加 execute 和 Aer 两项，在下方添加本地仿真代码，运行该 Python 程序，得到计算机本地仿真结果的输出。在本实验结束后，还可以在所有实验结束后，通过在量子计算课程指南网页上下载 QLab1.py.ipynb 文档[3]，并用 Jupyter Notebook 打开并运行，从而得到仿真结果。其中需要添加的仿真代码及运行结果如例1.3所示。

[3]本手册内的每个实验都会给出一个对应的.ipynb 文档，用于对实验内容进行辅助理解。这些文档将在所有实验都结束之后统一放出，它们均可用 Jupyter Notebook 打开并运行。

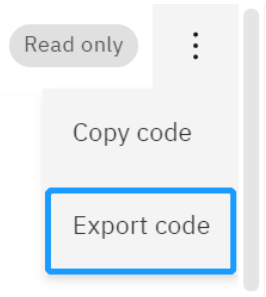


图 1.4: “ExportCode”位置

例 1.3: 仿真代码

```
# 电路运行和数据处理
# 对于小型实验，一般重复进行 1024 次：
total_shots = 1024
# 执行电路并将结果存入 job ：
job = execute(circuit, simulator, shots = total_shots)
# 从 job 中读取 0/1 字典：
result_dict = job.result().get_counts(circuit)

# 输出结果
# 直接输出该字典：
print(result_dict)
```

运行结果：

```
{'1': 513, '0': 511}
```

本地仿真的原理是先通过复数运算得到观测前的量子态，再通过计算机内部基于经典电路的算法产生的（伪）随机数，模拟对量子态观测所导致的（真）随机结果。

正确执行实验流程，最终的结果应该为“0”与“1”的频率各占 50% 左右。

(8) 回到 Circuit Composer 界面，在上方的工具栏中找到“Run Settings”并点开运行设置栏，在其中“System”菜单中选择后缀为“1q”的量子主机。*在进行后续实验时，请默认选择提供的量子比特数（见主机名称的后缀）不少于实验程序所需的基础上，量子比特数尽可能少的量子主机进行试验，以加快实验进度。*

输入合适的测试次数（小型实验一般取 1024 次）后，点击右上角的“Run on 量子主机名”蓝色按钮，将程序投入该主机的运行队列，全部操作按钮如图1.5所示。

(9) 等待程序排队运行完毕，期间可以点击右侧的“Jobs”按钮查看程序的排队/运行状态。运行完毕后，点击“Jobs”窗口中的对应程序，进入程序运行结果页面，所显示的结果即为该程序在量子计算机上运行所得到的实际频率分布 ci，如图1.6所示。

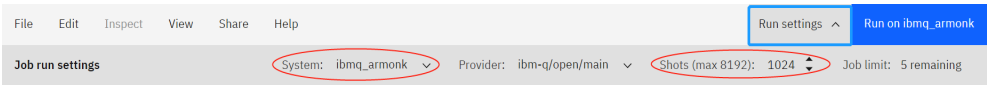


图 1.5: 提交代码操作面板

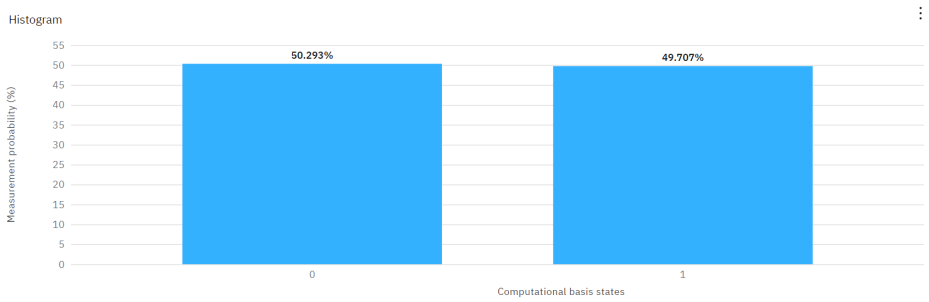


图 1.6: 一次运行结果

(10) 在上述步骤中，我们将大量的单量子比特初始化后通过一个 H 门，然后观测其状态，得到结果：有约 50% 的单量子比特显现出“0”状态、另外约 50% 的单量子比特显现出“1”状态。

不难从数学角度对此结果加以验算：

(a) 初始状态：

$$|\psi_0\rangle = |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1.7}$$

(b) 通过 H 门：

$$|\psi_1\rangle = H|\psi_0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{1.8}$$

(c) 观测：

$$\begin{aligned} P_0 &= |\langle 0|\psi_1\rangle|^2 = |\frac{1}{\sqrt{2}}|^2 = 50\% \\ P_1 &= |\langle 1|\psi_1\rangle|^2 = |\frac{1}{\sqrt{2}}|^2 = 50\% \end{aligned} \tag{1.9}$$

(11) 分别按下述方式对原电路进行些许调整，并重复步骤 (4)-(10)：

(a) 在原电路基础上，于 H 门前添加 X 门；

(b) 在原电路基础上，于 H 门前添加 H 门；

(c) 其它 X 门与 H 门的可能排列。

对仿真结果、运行结果进行分析，并从数学角度对其加以验算，总结 X 门与 H 门两种量子门对量子比特的作用规律。

1.5 补充知识

1.5.1 另外四种常见量子态

由“0”状态和“1”状态经过 H 门分别产生的两种叠加态，在后续的量子算法中会因其良好的性质而被经常用到。这两个量子态也被分别称作“+”态和“-”态，它们的数学形式如下所示：

$$
\begin{aligned}
|+\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\
|-\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle
\end{aligned}
\tag{1.10}
$$

这一组状态，又被称为 x 基（x Basis）。

$|0\rangle$、$|1\rangle$、$|+\rangle$、$|-\rangle$ 是实数空间内的四个重要的量子态。此外，我们在之后的实验中还将用到另外两个涉及复数的量子态“$\circlearrowleft$”态和“$\circlearrowright$”态，它们的数学形式如下所示：

$$
\begin{aligned}
|\circlearrowleft\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}i|1\rangle \\
|\circlearrowright\rangle &= \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}i|1\rangle
\end{aligned}
\tag{1.11}
$$

这一组状态，又被称为 y 基（y Basis），我们将在之后的实验中认识其性质与作用。

1.6 实验结果分析

1.6.1 思考题

(1) 如图1.7的三条单比特量子电路中，三个量子以$|0\rangle$ 态作为初始状态，运行结束后哪个量子的量子态与众不同？试从数学角度分析，为什么量子门的种类和数量都一样，但结果却可能不同？

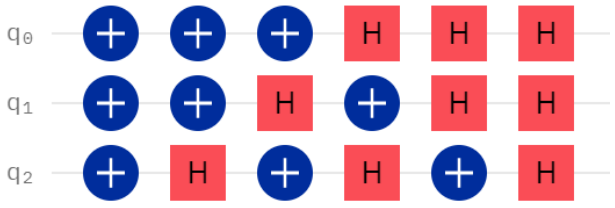


图 1.7: 三条不同的单比特量子电路

(2) IBMQ 中规定所有量子比特的矩阵表示均要遵从高位在左、低位在右的规则，即假如有量子比特序列$q_0, q_1, ..., q_{n-1}$，那么其矩阵表示应该写作：$|q_{n-1}, q_{n-2}, ...q_0\rangle = |q_{n-1}\rangle \otimes |q_{n-2}\rangle \otimes ... \otimes |q_0\rangle$。试分析这种表示规则有何好处？（提示: 研究$n = 2$时$|q_0, q_1\rangle$ 与$|q_1, q_0\rangle$ 向量的元素位置的差别）

1.6.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第1.6.1节中思考题的解答。

(2) 符合本实验要求的过程及结果。其中，

实验过程为：量子电路图、QASM 汇编代码、Python 代码三选一；

实验结果为：本地仿真结果、IBM 平台计算结果、本源平台计算结果三选一。

实验 2

探究量子相位及相关量子门作用

2.1 实验目的

(1) 理解量子相位的概念；
(2) 从数学角度与应用角度理解 P 门的作用。

2.2 实验概述

(1) 在量子实验平台上观察 P 门对单量子量子态的作用；
(2) 在量子实验平台上利用概率法验算一个自定 P 门的量子相位偏移；
(3) 在本地仿真中利用概率法测量并计算一个未知 P 门的量子相位偏移。

2.3 前提知识

2.3.1 量子相位

量子比特可以用二维复向量表示为：

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} r_1 e^{i\varphi_1} \\ r_2 e^{i\varphi_2} \end{pmatrix} \tag{2.1}$$

其中，c_1 和c_2 满足$|c_1|^2 + |c_2|^2 = r_1^2 + r_2^2 = 1$。

之前用 H 门和 X 门得到的“0”、“1”、“+”、“-”四种状态只是c_1 和c_2 为实数时的四个特殊状态而已，单量子比特可以做到的事情要丰富的多。

由于量子态乘以任意复数都不会改变该量子态，因此我们将式2.1中所示的$|\psi\rangle$ 看作$|\psi_0\rangle$，并进行如下简化：

$$|\psi\rangle = e^{-i\varphi_1}|\psi_0\rangle = e^{-i\varphi_1}\begin{pmatrix} r_1 e^{i\varphi_1} \\ r_2 e^{i\varphi_2} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 e^{i(\varphi_1-\varphi_2)} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 e^{i\varphi} \end{pmatrix} \tag{2.2}$$

其中，$|\psi\rangle$ 与$|\psi_0\rangle$ 是完全相同的量子态。可以看到，任一量子态$|\psi\rangle$ 的虚数部分都可以用$\varphi_1 - \varphi_2 = \varphi \in [0, 2\pi)$ 表示，我们称此φ 为式2.2中所示量子态的相位。

在对量子态进行观测时，有：

$$\begin{aligned} P_0 &= |\langle 0|\psi_1\rangle|^2 = |c_1|^2 = r_1^2 \\ P_1 &= |\langle 1|\psi_1\rangle|^2 = |c_2|^2 = r_2^2 \end{aligned} \tag{2.3}$$

可以看出，观测结果的概率分布与相位无关，因此可以得到结论：只改变量子态的相位不会影响观测结果的概率分布。

2.3.2 相移门

P 门又名 RZ 门[1]，代表了一类量子门——相移门[2]。部分改变特定相位的 P 门有专门的名字，如 T 门、S 门、Z 门等。此类量子门通用矩阵表示如下：

$$P(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \tag{2.4}$$

其可以让量子相位在原有基础上增加λ 弧度：

$$P(\lambda)|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 e^{i\varphi} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 e^{i(\varphi+\lambda)} \end{pmatrix} \tag{2.5}$$

以下为一些常用的 P 门：

$$\begin{aligned} T = P(\frac{\pi}{4}) \qquad S = P(\frac{\pi}{2}) \qquad Z = P(\pi) \\ T^\dagger = P(-\frac{\pi}{4}) \qquad S^\dagger = P(-\frac{\pi}{2}) \end{aligned} \tag{2.6}$$

尽管 P 门单独作用于单量子比特不会改变该比特的观测概率分布，但是与 H 门配合使用时情况会大不相同，我们将在之后的实验中研究并探讨其作用。

2.4 实验步骤

2.4.1 改变单量子比特的相位

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为一个，经典比特寄存器的数量调整为一个。

[1]后续实验文档中，在 P 门用于相位旋转单独出现时，使用“P 门”的表述；在其用于数学推演，与 RX 门、RY 门一起出现时，使用“RZ 门”进行表述并注明 RZ 门即为 P 门。

[2]相移门单独作用于一个量子比特时，只改变其相位，不改变其观测概率分布。通过相移门，我们可以表示出更多的量子态。

(3) 拖动工具栏中的 P 量子门到电路[3]此处以一个 S 门为例，电路如图2.1所示），然后观察下方“State vector”与“Measurement Probabilities”窗口，会发现两个窗口都没有任何变化。

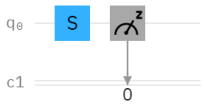


图 2.1: 单个 P 门电路示例

(4) 清空量子门，这次先添加一个 H 门，记录“State vector”与“Measurement Probabilities”窗口的结果；再添加一个 P 门，电路如图2.2所示，然后观察下方“State vector”与“Measurement Probabilities”窗口，对比观察到的结果与实验 1 中的只有一个 H 门时的结果，解释现象并计算佐证。

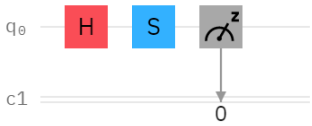


图 2.2: H-P 门电路示例

通过上述实验可以看出，处于$|0\rangle$与$|1\rangle$的叠加态的量子比特可以通过 P 门改变其量子相位，但只改变其量子相位并不会影响该叠加态对$|0\rangle$与$|1\rangle$下的观测结果。

那么问题来了：对于一个未知的 P 门，如何通过测量得到其偏转角？无论是让初始化的量子比特通过，还是让处于叠加态的量子比特通过，都不会改变其观测值，这时就需要运用 H-P-H 组合门了。

2.4.2 测量未知 P 门的相位偏移量

(1) 搭建如图2.3所示的量子电路，自定义中间的 P 门的相位偏移值，记录下方的“Measurement Probabilities”窗口中概率分布的数值。

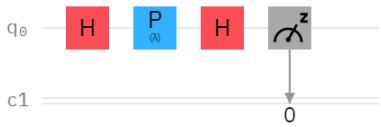


图 2.3: H-P-H 门电路示例

[3]在 Circuit Composer 界面中，所有 P 门都以浅蓝色为底。

(2) 通过计算不难发现概率分布P_0、P_1 与相位偏移值λ 的联系：

$$
\begin{aligned}
&|\psi_0\rangle = |0\rangle \\
&|\psi_1\rangle = H|\psi_0\rangle = |+\rangle \\
&|\psi_2\rangle = P(\lambda)|\psi_1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0\\ 0 & e^{i\lambda}\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ e^{i\lambda}\end{pmatrix} \\
&|\psi_3\rangle = H|\psi_2\rangle = \frac{1}{2}\begin{pmatrix}1 & 1\\ 0 & -1\end{pmatrix}\begin{pmatrix}1\\ e^{i\lambda}\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1+e^{i\lambda}\\ 1-e^{i\lambda}\end{pmatrix}
\end{aligned}
\tag{2.7}
$$

对$|\psi_3\rangle$ 进行测量，结果为[4]：

$$
\begin{aligned}
&P_0 = |\langle 0|\psi_3\rangle|^2 = |\frac{1}{2}(1+cos\lambda+isin\lambda)|^2 = \frac{1}{2}(1+cos\lambda) \\
&P_1 = |\langle 1|\psi_3\rangle|^2 = |\frac{1}{2}(1-cos\lambda-isin\lambda)|^2 = \frac{1}{2}(1-cos\lambda) \\
&\Delta P_x = P_0 - P_1 = cos\lambda
\end{aligned}
\tag{2.8}
$$

(3) 如图2.4所示，我们在 P 门$P(\lambda)$ 后加入一个 Sdg 门$S^\dagger$，由于两个量子门都只改变相位，二者可以合起来看作一个新的 P 门$P(\lambda)$，其中：

$$
\lambda' = \lambda - \frac{\pi}{2} \tag{2.9}
$$

仿照刚才的方法，我们可以测量出：

$$
\Delta P_y = cos\lambda' = cos(\lambda - \frac{\pi}{2}) = sin\lambda \tag{2.10}
$$

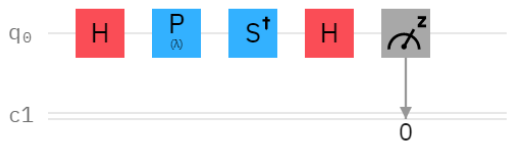


图 2.4: 调整相位测量正弦值的电路示例

这样我们就可以通过三角函数得到未知数λ 的取值了。

(4) 自拟一个相位偏移值λ，在量子实验平台上提交两个电路的有关代码，等待量子计算机运行完该程序并记录得到的频率分布结果，根据上述方法计算λ 的测量值，核验其与真实值是否近似。

在书面计算中，由于随机性会为测量带来误差，可能会导致$\Delta P_y^2 + \Delta P_x^2 \neq 1$ 的情况出现，此时可以使用下述公式对测量结果归一化：

[4]注意，复数模的平方等于复数自身乘以其共轭复数。

$$\bar{y} = \frac{\Delta P_y}{\sqrt{\Delta P_y^2 + \Delta P_x^2}}$$
$$\bar{x} = \frac{\Delta P_x}{\sqrt{\Delta P_y^2 + \Delta P_x^2}} \tag{2.11}$$

在计算机中，可以使用$atan2(\Delta P_y, \Delta P_x)$ 函数直接得出λ 的弧度值。

(5) 利用所学的 Qiskit 知识，在本地设计编写一个能够自动生成随机值λ，输出λ 的测量值和误差的程序，程序代码已在 QLab2_blank.py 文档中给好。

2.5 补充知识

2.5.1 U 门与布洛赫球面

在课堂上我们学习过使用布洛赫球面（Bloch Sphere）来表示量子态的方法，并证明了任何一个单量子的量子态均可以用球面上的一个点来表示。由于单量子的量子门可以看作由一个量子态到另一个量子态的映射，我们可以用量子态在布洛赫球面上的旋转来通用地表示所有量子门。通用量子门用 U 门[5]表示，其数学形式为：

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} cos(\frac{\theta}{2}) & -e^{i\lambda} sin(\frac{\theta}{2}) \\ e^{i\phi} sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)} cos(\frac{\theta}{2}) \end{pmatrix} \tag{2.12}$$

其中，$\theta, \phi, \lambda \in [\pi, -\pi]$。

固定其中一部分参数，我们可以分别得到象征着围绕 X、Y、Z 三个轴旋转的旋转量子门——RX 门、RY 门、RZ 门[6]，其数学形式如下：

$$R_x(\theta) = U_3(\theta, -\frac{\pi}{2}, \frac{\pi}{2}) = \begin{pmatrix} cos(\frac{\theta}{2}) & -isin(\frac{\theta}{2}) \\ -isin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix}$$
$$R_y(\theta) = U_3(\theta, 0, 0) = \begin{pmatrix} cos(\frac{\theta}{2}) & -sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix} \tag{2.13}$$
$$R_z(\lambda) = P(\lambda) = U_3(0, 0, \lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$$

三个特殊的弧度为π 的旋转门也分别简称为 X 门、Y 门、Z 门，其对量子态的影响也可以看作以坐标轴为中心对球面进行轴对称变换[7]：

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.14}$$

此外，H 门可做如下分解：

[5] U 门是 IBM 量子计算机上仅有的实际存在的非经典量子门，程序在实机运行前，会将所有量子门都转化成 U 门形式，因此 U 门也被称作物理门。

[6]不难看出,RZ 门与 P 门等效。我们也可以不严谨地说，RZ 门就是 P 门。

[7]不难看出，其中 X 门对经典电路的非门。事实上，Circuit Composer 也将 X 门称作 NOT 门。

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}cos(\frac{\pi}{4}) & -sin(\frac{\pi}{4})\\ sin(\frac{\pi}{4}) & cos(\frac{\pi}{4})\end{pmatrix} = XR_y(\frac{\pi}{2}) \tag{2.15}$$

故 H 门变换可以看作在布洛赫球面上绕 Y 轴旋转$\frac{\pi}{2}$ 后，绕 X 轴旋转π，如图2.5所示。

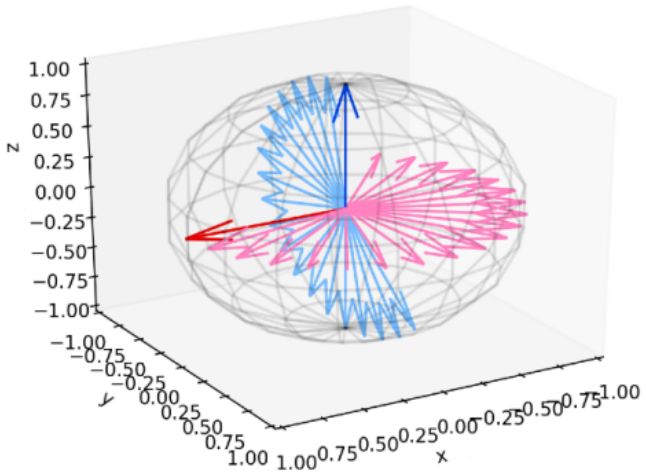


图 2.5: H 门对部分量子态的影响示意

其中，x 基为$|+\rangle$ 和$|-\rangle$；y 基为$|\circlearrowleft\rangle$ 和$|\circlearrowright\rangle$；z 基为$|0\rangle$ 和$|1\rangle$。

2.6 实验结果分析

2.6.1 思考题

(1) 如例2.1所示的是一个量子电路的 QASM 程序代码，量子的初始状态为$|0\rangle$，若要使观测结果确定（要么全部为 0、要么全部为 1），那么括号内可以填 _____（以弧度表示，结果如果不唯一可以借助 k）。

例 2.1: 一个量子电路的 QASM 程序代码

```
1 h q[0];
2 p() q[0];
3 h q[0];
```

(2) 在“测量未知 P 门的相位偏移量”实验中，实验结果记录如表2.1所示：
则待求角度最可能是 _____。
A.π　　B.$\frac{\pi}{2}$　　C.$\frac{\pi}{3}$　　D.$\frac{\pi}{4}$

	统计结果		
量子电路结构	0	1	合计
H-P-H	3112	984	4096
H-P-Sdg-H	3783	313	4096

表 2.1: 某实验记录表

2.6.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第2.6.1节中思考题的解答。

(2) 符合本实验要求的过程及结果。其中，

实验过程为：量子电路图、QASM 汇编代码、Python 代码三选一；

实验结果为：对表2.2的填写结果，及本地仿真结果、IBM 平台计算结果、本源平台计算结果三选一。

	观测结果			估算概率			
电路结构	0	1	总数	p(0)	p(1)	概率差	估算角度θ
H-P-H						$cos\theta =$	
H-P-Sdg-H						$sin\theta =$	

表 2.2: 实验 2 结果分析表

实验 3

典型的双量子与三量子纠缠态

3.1 实验目的

(1) 理解量子纠缠的概念；
(2) 从数学角度和应用角度理解 C 门的作用；
(3) 能够用线性代数完成简单的书面多量子比特电路跟踪。

3.2 实验概述

(1) 学习 CX 门的使用方法并利用其实现双量子比特的 Bell 态和三量子比特的 GHZ 态，完成电路搭建和数学推演；
(2) * 选做：应用所学知识并根据给定材料，补充残缺的量子电路图并辅以数学计算，实现所求三量子比特的特殊量子纠缠态。

3.3 前提知识

3.3.1 量子纠缠

多个量子组成整体后如果处于量子纠缠状态，则意味着无法通过单独描述各个粒子的性质来描述整体，即无法拆解为单个粒子状态的多量子态被称为量子纠缠。例如，双量子态$|0,0\rangle$ 可以拆解为两个单量子态$|0\rangle$ 和$|0\rangle$ 的张量积[1]：

[1]在本文档及后续文档中，只要没有特别说明，所有的矩阵张量积一律采用高位在前、低位在后的表示方法。例如，三个量子比特q_0、q_1、q_2 的状态用张量积写为$|q_2,q_1,q_0\rangle$，如果它们不处于纠缠态，则其状态应分解为：$|q_2,q_1,q_0\rangle = |q_2\rangle \otimes |q_1\rangle \otimes |q_0\rangle$。
分析量子电路图时，只要没有特别说明，一律以靠上线路为低位、靠下线路为高位。

$$|0,0\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\times\begin{pmatrix}1\\0\end{pmatrix}\\0\times\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix} = |0\rangle\otimes|0\rangle \tag{3.1}$$

但双量子态$\frac{1}{\sqrt{2}}(|0,0\rangle+|1,1\rangle)$却无法通过张量积拆解为更基础的单量子态。若试图拆分：

$$\frac{1}{\sqrt{2}}(|0,0\rangle+|1,1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix} = \begin{pmatrix}a\\b\end{pmatrix}\otimes\begin{pmatrix}c\\d\end{pmatrix} \tag{3.2}$$

则会发现$ac\neq 0$，$bd\neq 0$，$ad=bc=0$，式3.2无解。

从物理角度，这个状态代表：对这两个量子比特进行观测，它们有 50% 的可能性都是“0”态，有 50% 的可能性都是“1”态。如果观测一个量子比特，发现它处于“0”态，则另一个一定也处于“0”态——无论这两个量子比特距离有多远。

这两个量子比特所处的奇特的状态被称作 Bell 态（Bell-State），我们将在稍后探究其作用。

3.3.2 CX 门（CNOT 门）

单比特的量子门可以用一个二阶酉矩阵表示，而 n 比特的量子门则可以用2^n 阶酉矩阵表示。CX 门是一种特殊的双比特经典门，其矩阵表示如下：

$$CX_{q_0\to q_1} = I\otimes|0\rangle\langle 0| + X\otimes|1\rangle\langle 1| = \begin{pmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{pmatrix} \tag{3.3}$$

其作用可以理解为“如果q_0 为$|1\rangle$, 则对q_1 应用 X 门”，对经典比特的操作如表3.1所示：

操作前		操作后	
q_0	q_1	q_0	q_1
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

表 3.1: CNOT 门操作表

其中，不变的q_0 是控制位，而对q_1 的操作则受到了q_0 值的控制——当q_0 为$|1\rangle$ 时翻转q_1，而当q_0 为$|0\rangle$ 时什么都不做。

那么，如果q_0 处于$|0\rangle$ 与$|1\rangle$ 叠加态会发生什么呢?

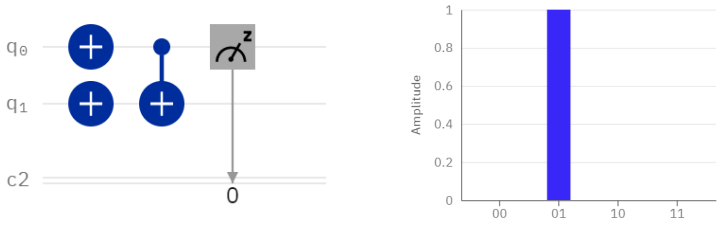
3.4 实验步骤

3.4.1 利用 CX 门构造 Bell 态和 GHZ 态

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为两个，经典比特寄存器的数量调整为两个。

(3) 构建如图3.1(a) 所示电路，在观察下方“State Vector”窗口中显示的量子态，可以看到如图3.1(b) 所示结果：当前量子态为$|0,1\rangle = |0\rangle \otimes |1\rangle$，表示$q_1$ 处于$|0\rangle$ 态、q_0 处于$|1\rangle$ 态[2]。



(a) 电路图 (b) “State Vector”窗口结果

图 3.1: CX 门电路一及其量子态

(4) 通过增减 X 门来改变电路的输入，观察下方“State Vector”窗口的变化，验证实验用 CX 门对量子比特的影响与上文中的状态表格是否吻合。

(5) 更改 CX 门的方向，使得q_1 成为控制位，观察”State Vector”窗口的变化并解释原因。本操作可以通过以下两种方法实现:

(a) 双击电路上的 CX 门符号并拖动“Qubits connections”栏下的连线，交换两个输入端到输出端的连接，如图3.2所示。

图 3.2: CX 门的拖动连线

[2]从今往后，若无特殊说明，输出格式均为“高位→ 低位”。

(b) 在右侧的 QASM 编程窗口中直接将“cx q[0],q[1];”改为“cx q[1],q[0];”，左侧会自动根据程序代码重新生成电路示意图。

(6) 在 CX 门前放置 H 门，搭建如图3.3所示电路，再观察“State Vector”窗口，会发现此时的量子态处于$\frac{1}{\sqrt{2}}(|0,0\rangle+|1,1\rangle)$状态，即发生了量子纠缠。我们称这种纠缠态为“Bell 态”。

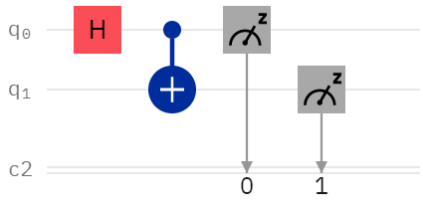


图 3.3: Bell 态电路

(7) 观察“Measurement Probabilties”窗口，会发现此时对两个量子进行观察，有 50% 可能性得到“00”，有 50% 可能性得到“11”。

(8) 通过计算不难验证上文实验中的结果：

(a) 初始状态：

$$|\psi_{q_0}\rangle=|\psi_{q_1}\rangle=|0\rangle=\begin{pmatrix}1\\0\end{pmatrix} \tag{3.4}$$

(b) q_0 通过 H 门：

$$H|\psi_{q_0}\rangle=|+\rangle \tag{3.5}$$

(c) q_0、q_1 整体的量子态可以写为：

$$|0\rangle\otimes|+\rangle=\frac{1}{\sqrt{2}}\begin{pmatrix}1\times\begin{pmatrix}1\\1\end{pmatrix}\\0\times\begin{pmatrix}1\\1\end{pmatrix}\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\\0\\0\end{pmatrix} \tag{3.6}$$

(d) 该整体通过 CX 门后的量子态为：

$$CX(|0\rangle\otimes|+\rangle)=\frac{1}{\sqrt{2}}\begin{pmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{pmatrix}\begin{pmatrix}1\\1\\0\\0\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix} \tag{3.7}$$

CX 门对处于叠加态的量子比特的作用还可以从以下两个比较直观但并不太严谨的角度理解：

(a) 处于控制位的量子比特既然处于叠加态，就同时具有“处于‘0’态的部分”和“处于‘1’态的部分”；而控制位通过 CX 门后，“处于‘0’态的部分”没有影响，“处于‘1’态的部分”则将被控制位的“一部分”反转到了 1 态——这让被控制位成为了和控制位同步的叠加态。

(b) 观察 CX 门的矩阵可以发现这是一个置换矩阵——其不改变向量的值，只交换向量部分位的位置；CX 门作用于量子比特的整体也可以看作 CX 矩阵作用于象征量子态的向量——矩阵通过置换将向量从一个可分解的状态变成了一个不可分解的状态。

(9) 同理，我们可以搭建三个量子比特的 GHZ 态（GHZ-State）电路，如图3.4所示。

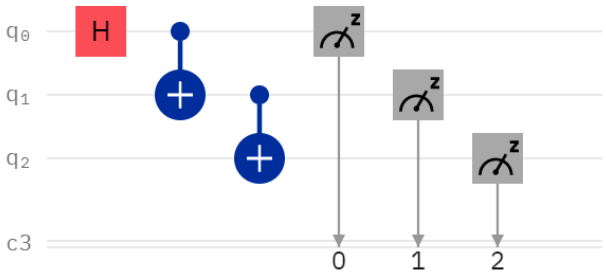


图 3.4: GHZ 态电路

(10) 完成电路搭建后，观察”State vector”、“Measurement Probabiities”等数据窗口，解释原因并对中间结果进行数学推演和验算。

3.4.2 * 选做：三量子比特的另一种特殊量子纠缠态

对于三个量子比特来说，除了 GHZ 态以外，还有一种常用的量子纠缠态，名为 W 态（W-State），其数学形式为：

$$\frac{1}{\sqrt{3}}(|0,0,1\rangle + |0,1,0\rangle + |1,0,0\rangle) = \frac{1}{\sqrt{3}}\begin{pmatrix}0\\1\\1\\0\\1\\0\\0\\0\end{pmatrix} \tag{3.8}$$

对处于这种状态的三个量子比特进行观测，将会分别有三分之一的概率得到“001”、“010”、“100”的结果。

阿 Q 为了在量子计算机中构造此叠加态，用一个 RY 门和一个 CRY 门搭建了如图3.5所示的电路。

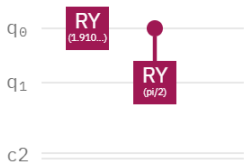


图 3.5: 阿 Q 的部分电路

其中 RY 门的参数为$arccos(-\frac{1}{3}) \approx 1.9106332362490184$；为了创建 CRY 门，需要将一个控制门修饰符[3]按如图3.6所示拖动到 RY 门上。

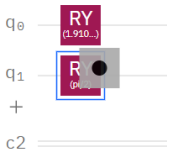


图 3.6: CRY 门创建方式

经过了这两个量子门后，量子比特q_0、q_1 整体的量子态可以写为：$\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\0\\1\end{pmatrix}$，而算上$q_2$ 后三个量子比特的整体状态为：$\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\0\\1\\0\\0\\0\\0\end{pmatrix}$，已经十分接近最终目标。

观察目前的量子态向量与目标量子态向量，不难发现二者概率幅的数值都是一样的，区别只在分布的位置上。那么你能否运用所学的量子电路知识帮助阿 Q，补充此电路以实现这种量子叠加态？要求：

(1) 在阿 Q 的部分电路之后添加量子门，使得最终量子态为：

$$\frac{1}{\sqrt{3}}(|0,0,1\rangle + |0,1,0\rangle + |1,0,0\rangle)$$

(2) 运用数学计算验证电路的结果。

[3]控制门修饰符为一个形如图3.6中以灰色为底的电路符号，和观测操作一样是不可逆操作，不属于量子门。

提示: 由于现有状态与目标状态的差别仅在概率幅分布上，所以使用若干经典门对叠加态的整体或部分进行置换变换即可达到目标。答案不唯一，其中一种将在本实验的结尾处放出，仅供参考。

3.5 补充知识

3.5.1 控制门

几乎每个单比特量子门都有其对应的控制门。在 Circuit Composer 界面中，只要将控制门修饰符拖动到所需量子门上，即可得到对应的控制门。控制门的含义在于：对单比特量子门 A 而言，CA 门意味着“如果控制位为 1 则对被控制位施用 A 门”，从数学角度可以写为:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$CA_{q_0 \to q_1} = I \otimes |0\rangle\langle 0| + A \otimes |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & c & 0 & d \end{pmatrix} \tag{3.9}$$

3.6 实验结果分析

3.6.1 思考题

(1) 将所有只由$|0\rangle$ 和$|1\rangle$ 组成的量子态称为基本量子态，判断正误：
   (a) 所有量子态都可以写作基本量子态在复空间的线性组合；
   (b) 纠缠态一定不是基本量子态；
   (c) 不是基本量子态的就是纠缠态。

(2) 试证明$CA_{q_0 \to q_1} = CA_{q_1 \to q_0}$ 当且仅当$A = P(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$。

提示: 参考控制门通用方程:

$$CA_{q_0 \to q_1} = I \otimes |0\rangle\langle 0| + A \otimes |1\rangle\langle 1| \quad CA_{q_1 \to q_0} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes A$$

3.6.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第3.6.1节中思考题的解答。

附加挑战：只使用三个量子门解答，其中，每个量子门只能是 X 门或 CX 门。请尝试枚举尽可能多的不同解答吧！

(2) 符合本实验要求的过程及结果。其中，

实验过程为：量子电路图、QASM 汇编代码、Python 代码三选一；

实验结果为：本地仿真结果、IBM 平台计算结果、本源平台计算结果三选一。

实验 4

Deutsch-Jozsa 算法的实现

4.1 实验目的

(1) 深入理解 Deutsch-Jozsa 算法的思路和原理；
(2) 掌握运用辅助位设计$(-1)^{f(X)}$ 型预言机电路的方法。

4.2 实验概述

(1) 完成单量子 Deutsch 算法的电路搭建；
(2) 完成双量子 Deutsch-Jozsa 算法的电路搭建。

4.3 前提知识

4.3.1 预言机电路（Oracle）

考察如图4.1所示的电路。

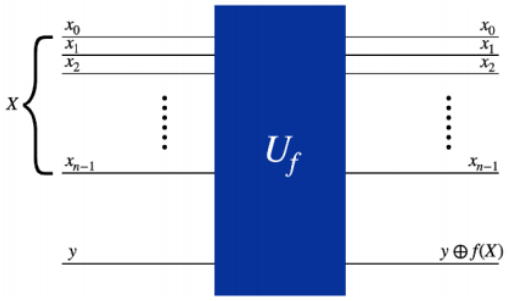


图 4.1: 预言机核心电路

$f(X)$ 是一个定义域为$\{0,1\}^n$、值域为$\{0,1\}$ 的函数。当输入$x_0, x_1, ..., x_{n-1}$ 为一串经典比特时，若$f(X)=0$，则输出与输入一致；若$f(X)=1$，则翻转 y。

对于任何$X \in \{0,1\}^n$：

当$Y = |0\rangle$ 时，输出为$f(X)X$；

当$Y = |1\rangle$ 时，输出为$f(\bar{X})X$。

因此可以得出：

$$U_f = \sum_{X\in\{0,1\}^n} (|f(X)X\rangle\langle 0X| + |\overline{f(X)}X\rangle\langle 1X|) \tag{4.1}$$

现在，我们让$X = |+,+,...,+\rangle$，$Y = |-\rangle$，考察其输出量子态：

$$\begin{aligned} U_f|-,+,+,...,+\rangle &= \frac{1}{\sqrt{2^{n+1}}} \cdot \sum_{X\in\{0,1\}^n} (|f(X)X\rangle - |\overline{f(X)}X\rangle) \\ &= \frac{1}{\sqrt{2^{n+1}}} \cdot \sum_{X\in\{0,1\}^n} (|f(X)\rangle - |\overline{f(X)}\rangle) \otimes |X\rangle \\ &= \frac{1}{\sqrt{2^n}} \cdot \frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \sum_{X\in\{0,1\}^n} (-1)^{f(X)}|X\rangle \\ &= |-\rangle \otimes (\frac{1}{\sqrt{2^n}} \cdot \sum_{X\in\{0,1\}^n} (-1)^{f(X)}|X\rangle) \end{aligned} \tag{4.2}$$

不难发现，经过U_f 之前：

$$U_{before} = |+,+,...,+\rangle = \frac{|0,0,...,0\rangle + |0,0,...,1\rangle + |1,1,...,1\rangle}{2^{\frac{n}{2}}} \tag{4.3}$$

经过U_f 之后，X 的形式没有改变，但所有让$f(X) = 1$ 的X 前的符号由加号变成了减号。这种可以在一次实验中标记未知函数的全部结果的电路被称作预言机（Oracle）。预言机通常作为题目给出的未知黑箱，而量子算法则体现为检验黑箱的特定性质并将其分类。

4.3.2 Deutsch-Jozsa 算法解决思路

如果对于任何X，$f(X) = 0$，那么$X_{after} = X_{before}$，最后输出$|+,+,...,+\rangle$；

如果对于任何X，$f(X) = 1$，那么$X_{after} = -X_{before}$，由于整体乘任何复数量子态不变，因此最后仍然输出$|+,+,...,+\rangle$。

依题意，U_f 只能是常值函数或平衡函数。构造如图4.2所示的完整电路，则如果$f(x)$ 是常值函数，结果一定为$|0,0,...,0\rangle$；同理也可以推出，如果$f(x)$ 是平衡函数，结果一定不可能包含$|0,0,...,0\rangle$。

4.4 实验步骤

4.4.1 单量子 Deutsch 算法的电路搭建

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

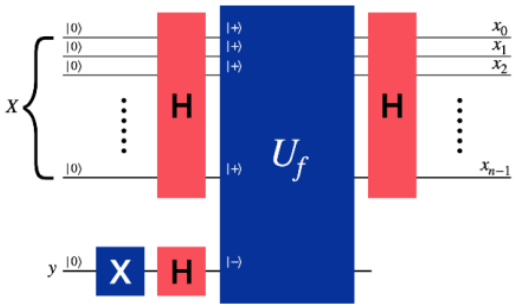


图 4.2: 算法完整电路

(2) 将量子比特寄存器的数量调整为 2 个，经典比特寄存器的数量调整为 1 个。
(3) 搭建如图4.3所示的预备电路，其中q_0 为X, q_1 为y。

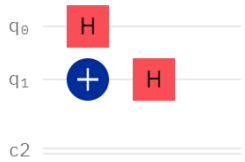


图 4.3: 单比特预备电路

(4) 根据函数$f(X)$ 设计预言机U_f, 这里将以常值函数$f(X) = 1$ 为例：
(a) 枚举X 和y 在经典比特时的输入输出，如表4.1所示。

输入		函数	输出	
y	x_0	$f(X)$	$y' = y \oplus f(X)$	$x_0' = x_0$
0	0	1	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	0	1

表 4.1: 单量子 Deutsch 算法电路输入输出表

(b) 由$U_f|yX\rangle = |y'X'\rangle$，我们可以得到$U_f$ 的数学形式：

$$U_f = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = X \otimes I = X_1 \tag{4.4}$$

分解U_f 可知，应对q_1 施用 X 门，不用对q_0 进行任何操作。

(c) 故可在原图基础上添加设计好的电路，如图4.4所示。

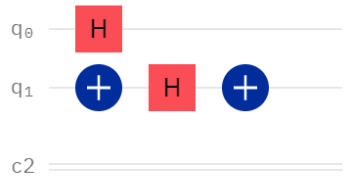


图 4.4: 单量子预备 -预言电路

(5) 最后将X 整体通过 H 门后观测，对单量子实验而言，这意味着对q_0 施用 H 门，然后观测。最终完整电路如图4.5所示。

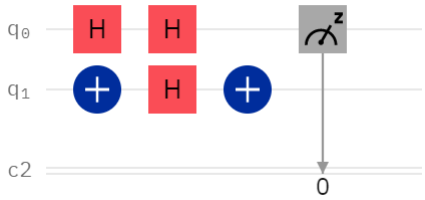


图 4.5: 完整单量子 Deutsch 算法电路

(6) 观察“Measurement Probabilties”窗口，可以看到对X 进行观测结果为“0”的概率是 100%，证明$f(X)$ 是常值函数，符合预期。

(7) 将$f(X)$ 改为以下两个平衡函数中的一种：$f(X) = x_0$ 或$f(X) = \overline{x_0}$（学生需要按自身学号的 hash 码选择改成哪一种，具体参见第4.6.2节末尾的表4.4），重复步骤 (3)-(6)，观察此时对X 进行观测的结果仿真是否符合预期。

(8) 以下为可能用到的矩阵分解：

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} = CX_{0\to 1} \qquad \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = X_1 \cdot CX_{0\to 1} \tag{4.5}$$

请特别注意量子门的编排顺序与矩阵的点乘顺序相反。

4.4.2 双量子 Deutsch-Jozsa 算法的电路搭建

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为 3 个，经典比特寄存器的数量调整为 2 个。

(3) 搭建如图4.6所示的预备电路，其中q_0，q_1 为X，q_2 为y。

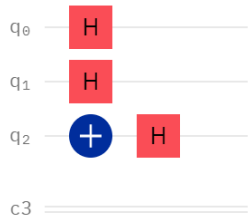


图 4.6: 双量子预备电路

(4) 根据函数$f(X)$ 设计预言机U_f，此处将以平衡函数$f(X) = x_0 \oplus x_1$ 为例。

(a) 枚举 X 和 y 在经典比特时的输入输出，如表4.2所示。

输入			函数	输出		
	X				X'	
y	x_1	x_0	$f(X)$	$y' = y \oplus f(X)$	$x'_1 = x_1$	$x'_0 = x_0$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	1	1	0
0	1	1	0	0	1	1
1	0	0	0	1	0	0
1	0	1	1	0	0	1
1	1	0	1	0	1	0
1	1	1	0	1	1	1

表 4.2: 双量子 Deutsch-Jozsa 算法电路输入输出表

(b) 由$U_f|yX\rangle = |y'X'\rangle$，我们可以得到$Ug_f$ 的数学形式：

$$U_f = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} = CX_{0\to 2} \cdot CX_{1\to 2} \tag{4.6}$$

分解U_f 可知，应对先对q_1，q_2 施用 CX 门，再对q_0，q_2 施用 CX 门[1]。故

[1]一般矩阵最优分解问题已被证明是 QMA 完全问题，这里直接查表完成分解，结果不唯一。

可在原图基础上添加设计好的电路，结果如图4.7所示。

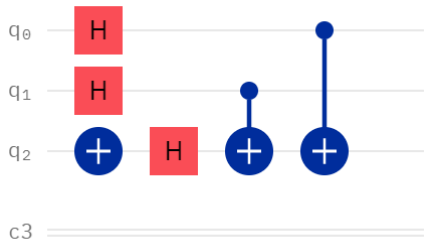


图 4.7: 双量子预备 -预言电路

(5) 最后将X 整体通过 H 门后观测，即对q_0，q_1 均施用 H 门后观测。最终完整电路如图4.8所示。

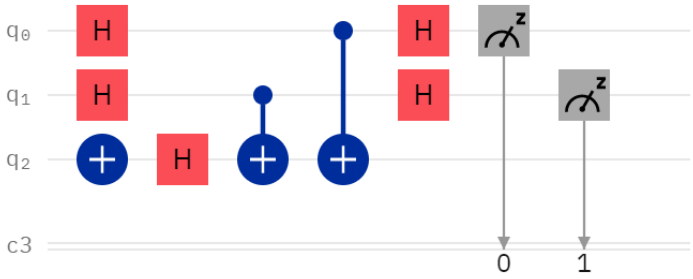


图 4.8: 完整双量子 Deutsch-Jozsa 算法电路

(6) 观察“Measurement Probabilties”窗口，可以看到对X 进行观测结果为“11”的概率是 100%，即结果并非全零，证明$f(X)$ 不是常值函数，符合预期。

(7) 将$f(X)$ 改为以下两个常值函数中的一种：$f(X)=0$ 或$f(X)=1$（学生需要按自身学号的 hash 码选择改成哪一种，具体参见第4.6.2节末尾的表4.4），重复步骤(3)-(6)，观察此时对X 进行观测的结果仿真是否符合预期。

4.5 补充知识

表4.3为 16 个预言机置换矩阵的量子门构造（假设原排列为“1 2 3 4 5 6 7 8”）。

4.6 实验结果分析

4.6.1 思考题

DJ 算法能够帮助我们能够判断一个函数是常值函数还是平衡函数，但这种功能基于一个假设：“函数只可能是常值函数或者平衡函数”。考虑这种情况：

排列								量子门排列顺序（已按时序排列）
1	2	3	4	5	6	7	8	-
1	2	3	8	5	6	7	4	$CCX_{01\to 2}$
1	2	7	4	5	6	3	8	$CCX_{01\to 2}, CX_{1\to 2}$ 或$X_0, CCX_{01\to 2}, X_0$
1	2	7	8	5	6	3	4	$CX_{1\to 2}$
1	6	3	4	5	2	7	8	$CCX_{01\to 2}, CX_{0\to 2}$ 或$X_1, CCX_{01\to 2}, X_1$
1	6	3	8	5	2	7	4	$CX_{0\to 2}$
1	6	7	4	5	2	3	8	$CX_{1\to 2}, CX_{0\to 2}$
1	6	7	8	5	2	3	4	$CCX_{01\to 2}, CX_{1\to 2}, CX_{0\to 2}$
5	2	3	4	1	6	7	8	$CCX_{01\to 2}, CX_{1\to 2}, CX_{0\to 2}, X_2$ 或 $X_1, X_0, CCX_{01\to 2}, X_1, X_0$
5	2	3	8	1	6	7	4	$CX_{1\to 2}, CX_{0\to 2}, X_2$
5	2	7	4	1	6	3	8	$CX_{0\to 2}, X_2$
5	2	7	8	1	6	3	4	$CCX_{01\to 2}, CX_{0\to 2}, X_2$
5	6	3	4	1	2	7	8	$CX_{1\to 2}, X_2$
5	6	3	8	1	2	7	4	$CCX_{01\to 2}, CX_{1\to 2}, X_2$
5	6	7	4	1	2	3	8	$CCX_{01\to 2}, X_2$
5	6	7	8	1	2	3	4	X_2

表 4.3: 预言机置换矩阵的量子门构造表

$f(X) = x_0 \wedge x_1$，当且仅当$x_0 = x_1 = 1$ 时有$f(X) = 1$，使用相同的算法是否有办法将其与平衡函数、常值函数加以区别？对于任意一个双比特函数$f(x_1x_0)$，构造出的 DJ 算法的电路可能产生哪些结果？尝试通过实验总结规律。

4.6.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第4.6.1节中思考题的解答。

(2) 符合本实验要求的过程及结果。其中，

实验过程为：量子电路图、QASM 汇编代码、Python 代码三选一；

实验结果为：本地仿真结果、IBM 平台计算结果、本源平台计算结果三选一。

实验类别	学号 hash 码的位	偶数	奇数
单量子 DJ 算法电路实验	第 1 位	$f(X) = x_0$	$f(X) = \overline{x_0}$
双量子 DJ 算法电路实验	第 2 位	$f(X) = 0$	$f(X) = 1$

表 4.4: $f(X)$ 选取规则表

实验 5

BB84 协议模拟实验

5.1 实验目的

(1) 理解构建 BB84 协议的理论基础；
(2) 掌握 BB84 协议的实现流程。

5.2 实验概述

(1) 根据前提知识，了解 BB84 协议，并完成通过该协议将 8 位初始值生成密钥的流程演算；
(2) 根据提供的代码，理解通过 BB84 协议将 1024 位初始值生成密钥的流程。

5.3 前提知识

5.3.1 BB84 协议

BB84 协议是量子密码学中第一个密钥分发协议，由 Bennett 和 Brassard 在 1984 年提出，也是使用和实验最多的量子密钥分发方案之一。BB84 协议通过光子的 4 种偏振态来进行编码：线偏振态和，圆偏振和，如图5.1所示。其中，线偏振光子和圆偏振光子的两个状态各自正交，但是线偏振光子和圆偏振光子之间的状态互不正交。

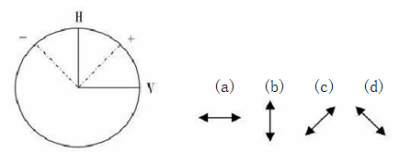


图 5.1: 光子的 4 种偏振态

BB84 协议的实现需要两个信道：经典信道和量子信道。经典信道要确保发送方 Alice 和接收方 Bob 之间能进行一些必要信息的交换，而量子信道用于传输携带信息的或者随机的量子态。

BB84 协议的实现思路如下（以 8 位初始值为例）：

(1) 发送方 Alice 随机产生一组二进制序列sA，假定数值为$|0,1,1,0,0,1,0,1\rangle$，然后，Alice 再生成另一组相同长度的随机序列mA，假定数值为$|b,a,b,b,b,b,a,a\rangle$，其中，a 表示线偏振，b 表示圆偏振。

(2) 根据这两个序列，调制产生 8 个光子，这些光子将被通过量子信道传输到 Bob 处。根据如表5.1所示的关系确定如何调制每个光子的状态。

初始值sA	发送基mA	发送基表示法	光子状态	光子表示法
0	线偏振	a	如图5.1(a)	0
0	圆偏振	b	如图5.1(c)	+
1	线偏振	a	如图5.1(b)	1
1	圆偏振	b	如图5.1(d)	-

表 5.1: 光子状态调制方法表

(3) 由于接收方 Bob 并不知道应该用哪组基进行测量，所以 Bob 生成一个随机序列用来选择测量基，假定称之为测量基序列mB，假设数值为$|a,a,b,a,b,a,b,a\rangle$，Bob 对粒子进行测量。

(4) 之后，Bob 通过经典信道告知 Alice 测量基序列mB。Alice 比较 Bob 的测量基序列mB 和她自己保留的发送基序列mA，并通知 Bob 所采用的测量基中哪些测量选择正确并予以保留，这样 Alice 和 Bob 就拥有一组所选的基相一致的随机二进制序列，称为筛选码。计算过程如表5.2所示，筛选码为第 1、2、4、7 位。

sA	mA	mB	结果
0	b	a	0 或 1（舍弃）
1	a	a	1
1	b	b	1
0	b	a	0 或 1（舍弃）
0	b	b	0
1	b	a	0 或 1（舍弃）
0	a	b	0 或 1（舍弃）
1	a	a	1

表 5.2: 筛选码表

(5) Alice 和 Bob 从筛选码中随机选取 n 位并公开宣布其值和所选的基，以估计信道参数和量子比特误码率。此处选取所有位，若无窃听者，理论上，最终生成密钥序列为$|1,1,0,1\rangle$。

(6) 若误码较高（即，有过多本应得到精确测量结果的比特，测量结果与理论值不符），说明存在窃听方 Eve，且 Eve 获得了较多信息，那么就放弃本次通信。若误码率较低，那么通过误码纠错和私钥放大等后处理方法就可除去 Eve 获得的信息，得到 Eve 一无所知的、Alice 和 Bob 共有的随机密钥。

在 BB84 协议中，所采用的线偏振和圆偏振是共扼态，满足测不准原理。根据测不准原理，线偏振光子的测量结果越精确意味着对圆偏振光子的测量结果越不精确。因此，测量必然造成量子状态的改变。理论上，光子被发送之后只会被 Bob 测量，只有 Alice 的发送基mA 和 Bob 的测量基mB 一致时，测量结果才准确。而如果存在窃听方 Eve，其测量必定会对原来量子状态产生改变，这相当于发送的序列被改变，此时 Bob 的测量结果将与理论上的结果不符。随后，Alice 和 Bob 在经典讯道通讯并公开对比选择的基时，可以根据测不准原理检测出该扰动，从而检测出是否存在窃听。另外，线偏振态和圆偏振态是非正交的，因此它们是不可区分的，由量子不可克隆定理，Eve 不可能精确地测量所截获的每一个量子态，也就不可能制造出相同的光子来冒充。就算有巧合产生同样的光子，只要发送的光子序列足够长，Eve 的存在就一定会被发现。测不准原理和量子不可克隆定理保证了 BB84 协议量子通信的无条件安全性。

5.4 实验步骤

本次实验我们使用 python 语言来模拟 1024 位的 BB84 协议通信。该协议可以完全在代码层面模拟。实验代码写成 QLab5.py.ipynb, 详细分析如下：

(1) 首先假定发送方调制光子的态分别为“0”，“1”，“+”，“-”四种，用一个字典 basis 来表示，如例5.1所示。

例 5.1: 发送方光子态调制字典

```
1  import random
2
3  basis = {'A':('0','1'),'B':('+','−')}
4
5  # basis['A'] = ('0','1')
6  # basis['A'][0] = '0'
7  # basis['A'][1] = '1'
8
9  # basis['B'] = ('+','−')
10 # basis['B'][0] = '+'
11 # basis['B'][1] = '−'
```

(2) 定义接收方测量的方式：表现为一个函数 measure()，如例5.2所示，该函数接受两个参数。接收到的光子 q 和对应的测量基 b；如果 q 的发送基和测量基相同，则返回它原本对应的二进制序列值，否则返回一个随机值，说明该值损坏。

例 5.2: 定义接收方测量方式

```
def measure(q,b):
if q in basis[b]:
 # basis['A']|basis['B']=>q in ()=>q 是否属于这个 ()
return basis[b].index(q)       # 返回 q 在 () 里的索引
else:
return random.choice( (0,1) ) # 输入一个元组作为随机数选择来源
```

(3) 生成随机的二进制序列和发送基，如例5.3所示，分别是 aliceibits 和 aliceibasis。采用 random.choice() 函数来进行随机选择，该函数接受一个元组，随机返回一个存在于该元组的元素。

例 5.3: 生成随机的二进制序列和发送基

```
N = 1024

# Alice
alice_bits  = [random.choice((0,1)) for i in range(N)]
 # alice_bits = [] 创建一个空 list
 # for i in range(N):
 # range(N) 是个函数，它返回一个 [0,1,...,N-1] 的 list
 # alice_bits[i] = random.choice((0,1)):
 # 类似于 C 中的 for 循环： for (int i = 0;i < N;i++) {}
alice_basis = [random.choice(('A','B')) for i in range(N)]
 # 创建一个长度为 N 的，其中每个元素都从 ('A','B') 中选择的list
```

(4) 模拟在量子信道上传送，生成调制后的光子流，表现为列表 sending，如例5.4所示。对每一组二进制序列和发送基的对应位，采用 basis 字典规定的映射方式进行调制并加入 sending。

例 5.4: 量子信道传输

```
# Q-Channel
sending = [basis[b][i] for i,b in zip(alice_bits,alice_basis)]
 # 创建一个 list ，它有如下特点：
 # zip(alice_bits,alice_basis) 返回一个
 # 形如 [(1,'A'),(0,'B'),...,(1,'A')] ，长为 1024 的， list
 # 其中每个元组都作为 (i,b) 来向 basis 查找值，然后填充入 sending

# zip() 的解释：
# a = [1,2,3]
```

```
# b = [4,5,6]
# zipped = zip(a,b)      # 打包为元组的列表
# [(1, 4), (2, 5), (3, 6)]

# zip() 的运用：
# alice_bits = [1,0,...,1]
# alice_basis = ['A','B',...,'A']
# zip => [(1,'A'),(0,'B'),...,(1,'A')] 长度为 1024
```

(5) 模拟 Bob 的接收和可能出现的 Eve 的窃听，如例5.5所示。信号首先被 Eve 窃听，Eve 生成一组随机测量基，然后用这组测量基对 sending 中每个光子进行 measure 操作，并且由于测量光子之后就出现了信息损坏，sending 信号被更改了；Bob 接收时，生成一组随机的测量基，然后用这组测量基对修改后的 sending 中每个光子进行 measure 操作。

例 5.5: 窃听/接收光子序列

```
# Eve
eve_basis = [random.choice(('A','B')) for i in range(N)]
eve_bits  = [measure(q,b) for q,b in zip(sending,eve_basis)]
sending   = [basis[b][i] for i,b in zip(eve_bits,eve_basis)]

# Bob
bob_basis = [random.choice(('A','B')) for i in range(N)]
bob_bits  = [measure(q,b) for q,b in zip(sending,bob_basis)]
```

(6) 模拟在物理信道上进行传输，如例5.6所示。Bob 向 Alice 发送自己的测量基，Alice 将其与自己的基进行比较，形成一串表示每个基是否相等的二进制串，然后将其分别与 aliceibits，bobibits 进行相与，取得密钥 aliceikey 和 bobikey。

例 5.6: 物理信道传输

```
# P—Channel
conf_basis  = [a==b for a,b in zip(alice_basis,bob_basis)]
 # 比较 alice_basis 及 bob_basis 的每个分量，
 # 对应分量相同时结果为 1 ，否则为 0
 # 比较结果形如 [1,0,1,...,1] ，长为 1024 位

# Alice
alice_key   = [a for a,c in zip(alice_bits,conf_basis) if c]
 # 当 confibasis 当前分量为 1 时，
 # 才将对应 alice_bits 分量放入 alice_key 中
 # alice_key 的结果不一定为 1024 位

# Bob
```

```
14 bob_key      = [b for b,c in zip(bob_bits,conf_basis) if c]
15  # 同上，结果不一定为 1024 位
```

(7) 进行校验。比较两个密钥的前半部分，形成一串表示每个密钥分量是否相等的二进制串 confikey。假定误码率阈值为 0.03，计算 confikey 中 1 的个数除以 confikey 的长度得到的值，该数值即为 1-误码率，然后与阈值进行比较，判断是否存在窃听，如果存在窃听，输出 1-误码率，否则输出 bobikey 的后半部分。

例 5.7: 校验密钥

```
1  confikey    = [a==b for a,b in zip(alice_key[:len(alice_key)
    //2],bob_key[:len(bob_key)//2])]
2   # / 是浮点数除法， // 是整数除法
3   # alice_key[:len(alice_key)//2]:
4   # alice_key 第 0 项到第 len(alice_key)//2 项组成的列表（去尾法）
5  alpha       = 0.97
6  corr_rate   = conf_key.count(True)/len(conf_key)
7   # 数出列表中为 True|1 的数量，然后除以总长（浮点除）
8  if corrirate < alpha:
9  print('Eve␣detected␣with␣corr_rate␣:',corr_rate)
10 else:
11 print('Key␣:',bob_key[len(bob_key)//2:])
12  # bob_key 的后半部分（去尾法）
```

(8) 某次运行上述代码，得到如下结果：

```
Eve detected with corr_rate : 0.7375
```

由图可知，能够正确找到 Eve 的存在，并给出误码率为 0.27 左右，这是一个比较高的数值。重复实验，得到另外两个结果，印证了结论的正确性：

```
Eve detected with corr_rate : 0.8008
```

```
Eve detected with corr_rate : 0.7585
```

作为对照，我们将步骤 (5) 中 Eve 窃听的三行代码注释掉，然后重新运行，查看结果：

```
Key:[0,0,0,0,1,1,0,0,1,0,1,0,0,1,0,0,0,0,1,0,0,0,0,1,0,1,1,1,1,0,
0,0,0,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,0,0,0,1,0,0,0,0,0,0,1,0,0,
0,0,1,0,1,0,1,0,0,0,0,1,1,0,0,1,0,0,1,1,0,1,1,0,0,1,1,0,1,1,0,
1,0,1,0,0,0,0,1,1,0,1,0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,1,1,1,0,1,
1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,0,0,1,0,0,1,1,1,1,1,1,0,1,0,1,0,
0,0,0,1,1,0,1,0,0,1,1,1,0,1,1,1,0,1,1,0,0,1,1,0,1,1,0,1,1,1,0,
0,0,0,1,0,1,0,0,0,1,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0,0,0,1,0,0,0,
0,1,1,0,1,0,0,0,0,1,1,1,1,0,0,1,1,1,1,1,0,1,1,0,0,0,1,1,0,0,1,
1,0,1]
```

程序正确输出了我们想要的密钥，经重复实验，可进行复现，说明程序能够正确判断出误码率是否小于 3%。

5.5 实验结果分析

5.5.1 思考题

(1) 观察 BB84 协议的全部过程，说明：为什么在经典计算机上模拟 BB84 协议并不能产生安全的密钥？哪些环节会受到攻击？量子通信是如何避免这些攻击的？

(2) 在仿真中，我们并没有模拟随机误差可能引起的比特错误，这使得在无人窃听时正确率始终是 100%。修改你的程序，引入一个随机变量ε 代表量子位出错（反转）的概率，重复实验并分析观察到的结果，然后说明：量子位出错的概率都可能从哪些方面对 BB84 协议产生影响？

5.5.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第5.5.1节中思考题的解答。

(2) 符合本实验要求的过程及结果。其中，

实验过程为：Python 代码；

实验结果为：Python 代码运行结果。

实验 6

* 选做：Grover 算法的实现

6.1 实验目的

(1) 深入理解 Grover 算法的思路和原理；
(2) 掌握运用辅助位设计$2|s\rangle\langle s| - I$ 型放大机电路的方法。

6.2 实验概述

(1) * 选做：完成双量子 Grover 算法的电路搭建；
(2) * 选做：完成三量子 Grover 算法的电路搭建。

6.3 前提知识

6.3.1 放大机电路简介

设向量$|s\rangle = \frac{1}{\sqrt{2^n}} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$，考察矩阵$U_s = 2|s\rangle\langle s| - I$，通过计算不难得出其矩阵形式：

$$U_s = \frac{1}{2^{n-1}} \begin{pmatrix} 1-2^{n-1} & 1 & \cdots & 1 & 1 \\ 1 & 1-2^{n-1} & \cdots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1-2^{n-1} & 1 \\ 1 & 1 & \cdots & 1 & 1-2^{n-1} \end{pmatrix} \tag{6.1}$$

这个矩阵具有如下几个重要性质（相关证明与讲解可以参考课本）：
(1) 这是一个酉矩阵——这意味着它可以作为量子门；

(2) 该矩阵作用于任一向量时，等价于将向量中每个元素值依整体平均值翻转。
例如，取$n=3$，此时：

$$U_s = \frac{1}{4}\begin{pmatrix} -3 & 1 & \cdots & 1 & 1 \\ 1 & -3 & \cdots & 1 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & -3 & 1 \\ 1 & 1 & \cdots & 1 & -3 \end{pmatrix} \tag{6.2}$$

设向量$v=(4,4,4,4,4,-4,4,4)^T$，其平均值为$\frac{4\times7-4}{8}=3$，计算$U_s v=(2,2,2,2,2,10,2,2)^T$，恰好等价于每一项依平均值翻转。

这种电路由于具有放大标记值的作用（如例中的“-4”），被称作放大机。

6.3.2 放大假电路的一种解决方法

在这里将不加证明地给出放大机矩阵$U_s=2|s\rangle\langle s|-I$ 的一种分解方法：

$$U_s = H^{\otimes n}\cdot X^{\otimes n}\cdot \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & -1 \end{pmatrix} \cdot X^{\otimes n}\cdot H^{\otimes n} \tag{6.3}$$

观察中间的矩阵，发现它就是$f(X)=\begin{cases}1 & ,X=|1,1,...,1\rangle \\ 0 & ,Otherwise\end{cases}$ 时的预言机U_f（忽略临时位y）的矩阵形式，故我们可以直接用U_f 电路和 H 门、X 门来实现U_s 电路，如图6.1所示。

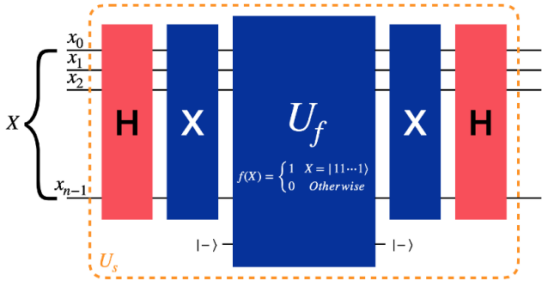


图 6.1: 放大机电路的一种实现方法

6.3.3 运用预言机和放大机电路实现 Grover 算法

如果一个函数有 N 种可能的求解结果，而其中只有一种是正确的，那么 Grover 算法就可以有效地从中找到那个求解结果。从数学上，我们可以说，如果有一个函数

满足：

$$f(X) = \begin{cases} 1 & ,X = |1,1,...,1\rangle \\ 0 & ,Otherwise \end{cases} \tag{6.4}$$

在经典算法中，如果想找到X_{target}，需要依次枚举所有可能的X 计算$f(X)$；但使用 Grover 量子算法，我们可以在极短时间内快速找到它。

具体推导可以参考课本，这里将直接给出 Grover 算法的电路实现思路。首先，利用 H 门构造$|+,+,...,+\rangle$，然后将其通过对应未知函数的预言机，最后将其通过放大机，不断重复“预言机 -放大机”的循环直到X_{target} 对应的概率幅度满足要求。电路设计的思路如图6.2所示。

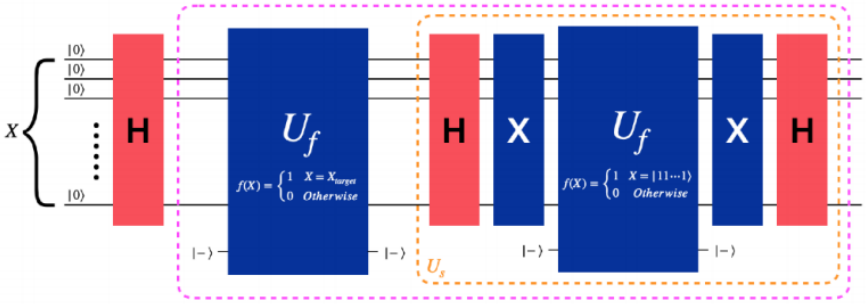


图 6.2: Grover 算法完整电路

6.4 实验步骤

6.4.1 * 选做：双量子 Grover 算法的电路搭建

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为 3 个，经典比特寄存器的数量调整为 2 个。

(3) 搭建预备电路如图6.3所示的电路，其中q_0、q_1 为X，q_2 为y。

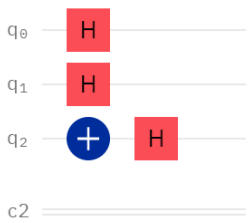


图 6.3: 双比特 Grover 算法预备电路

(4) 参考 4.4.2 中的步骤 (4)，设计预言机U_f。此处以$X_{target} = |1,1\rangle$，即$f(X) = \begin{cases} 1 & ,X = |1,1\rangle \\ 0 & ,Otherwise \end{cases}$为例。使用实验 4 的列表法可以发现$U_f$ 的置换矩阵排列为：

$$U_f = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = CCX_{0,1\to 2} \tag{6.5}$$

得到的量子电路如图6.4所示。

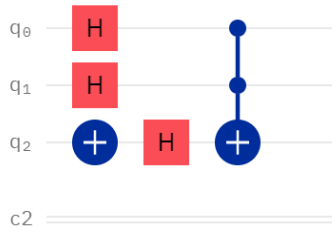


图 6.4: 双比特 Grover 算法预备电路与预言机

表6.1为X_{target} 取不同值时预言机电路的构造方法。观察表格规律不难发现：对

X_{target}	量子门顺序
$\|0,0\rangle$	$CCX_{0,1\to 2}, X_0, X_1$
$\|0,1\rangle$	$CCX_{0,1\to 2}, X_0$
$\|1,0\rangle$	$CCX_{0,1\to 2}, X_1$
$\|1,1\rangle$	$CCX_{0,1\to 2}$

表 6.1: 双量子X_{target} 取值与对应量子门表

于某个X_{target} 取值的预言机电路，只需要首先以X_{target} 作为控制位、y 作为被控制位对$|yX\rangle$ 应用 CCX 门[1]；然后对所有X_{target} 中取值为 0 的比特位应用 X 门即可。

(5) 根据如图6.5所示的设计图添加放大机电路和观测电路。

[1]CCX 门在 Circuit Composer 界面中的英文名为 Toffoli gate。

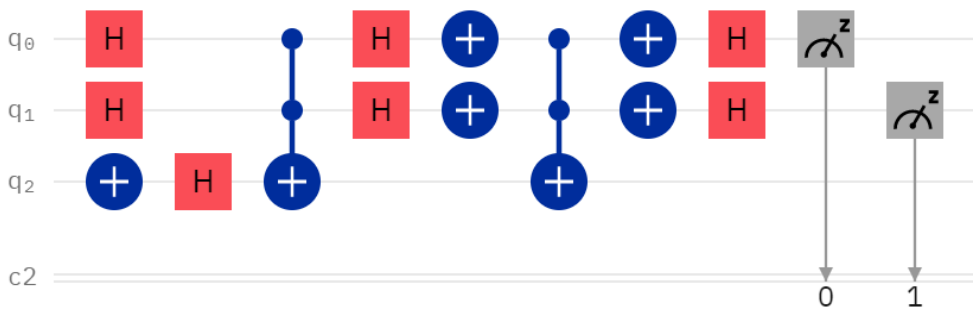


图 6.5: 双比特 Grover 算法完整电路

(6) 观察下方“Measurement Probabiities”窗口，可以看到，仿真观测结果为“11”的概率为 100%，与设定的$X_{target} = |1,1\rangle$ 吻合。

(7) 改变X_{target} 的取值（学生需要按自身学号的 hash 码选择改成哪一种，具体参见第6.6.2节末尾的表6.3），并依据上文表格修改预言机电路的构造，重复步骤(4)-(6)，检查仿真观测结果与所设X_{target} 是否一致。

6.4.2 * 选做：三量子 Grover 算法的电路搭建

(1) 登入 IBM Quantum Experience 平台，进入 Circuit Composer 界面，创建一个新电路。

(2) 将量子比特寄存器的数量调整为 5 个，经典比特寄存器的数量调整为 3 个。

(3) 搭建预备电路如图6.6所示的电路，其中q_0、q_1、q_2 为X，q_3 为临时位，q_4 为y。

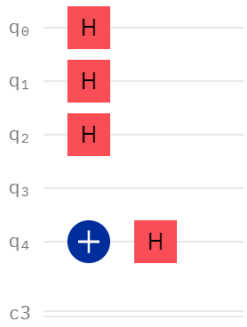


图 6.6: 三比特 Grover 算法预备电路

(4) 搭建预言机电路U_f，这里以$X_{target} = |1,1,1\rangle$ 为例。分析置换矩阵可以看出，

$X_{target} = |1,1,...,1\rangle = |1\rangle^{\otimes n}$ 时，有：

$$U_f = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 1 & \cdots & 0 & 0 \end{pmatrix} \tag{6.6}$$

可以看出，U_f 相当于一个单位矩阵的第$\frac{n}{2}$ 行与第n 行互换。

故此时U_f 矩阵等价于以X 整体作为控制位、y 作为被控制位的 C^nX 量子门。在本例中，$U_f = C^3X_{0,1,2\to4}$。对于超过两比特的 C^nX 量子门，仅使用给定数量比特的实现方法较为复杂，故这里采用借助临时位等价实现的思路，该思路如下：我们知道 $C^nX_{x_0,x_1,...,x_{n-1}\to y}$ 量子门的含义在于，当且仅当$x_0, x_1, ..., x_{n-1}$ 均为 1 时对y 应用 X 门，而逻辑与运算具有结合律。我们可以使用临时位作为草稿存储与运算的结果，并将之与后续比特（或其运算结果）进一步结合。计算完毕后，再将临时位还原。

本例中，我们使用q_3 作为临时位存储$q_0\&q_1$ 的结果，即:$q_0\&q_1\&q_2 = (q_0\&q_1)\&q_2 = q_3\&q_2 = q_4$。

$U_f = C^3X_{0,1,2\to4}$ 实现方法如图6.7所示。

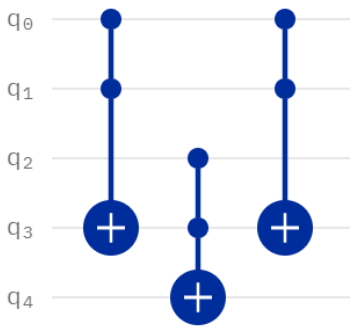


图 6.7: 三比特控制非门的实现

得到电路如图6.8所示。表6.2为X_{target} 取不同值时预言机电路的构造方法。

(5) 根据设计图添加放大机电路和观测电路，如图6.9所示。

(6) 观察下方“Measurement Probabiities”窗口，可以看到，仿真观测结果为“111”的概率为 78.125%，占绝对多数，与设定的$X_{target} = |1,1,1\rangle$ 吻合。

(7) 改变X_{target} 的取值（学生需要按自身学号的 hash 码选择改成哪一种，具体参见第6.6.2节末尾的表6.3），并依据上文表格修改预言机电路的构造，重复步骤(4)-(6)，检查仿真观测结果的绝对多数与所设X_{target} 是否一致。

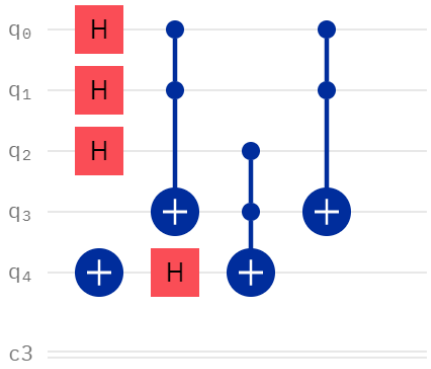


图 6.8: 三比特 Grover 算法预备电路与预言机

X_{target}	量子门顺序
$\|0,0,0\rangle$	$CCX_{0,1,2\to4}, X_0, X_1, X_2$
$\|0,0,1\rangle$	$CCX_{0,1,2\to4}, X_1, X_2$
$\|0,1,0\rangle$	$CCX_{0,1,2\to4}, X_0, X_2$
$\|0,1,1\rangle$	$CCX_{0,1,2\to4}, X_2$
$\|1,0,0\rangle$	$CCX_{0,1,2\to4}, X_0, X_1$
$\|1,0,1\rangle$	$CCX_{0,1,2\to4}, X_1$
$\|1,1,0\rangle$	$CCX_{0,1,2\to4}, X_0$
$\|1,1,1\rangle$	$CCX_{0,1,2\to4}$

表 6.2: 三量子X_{target} 取值与对应量子门表

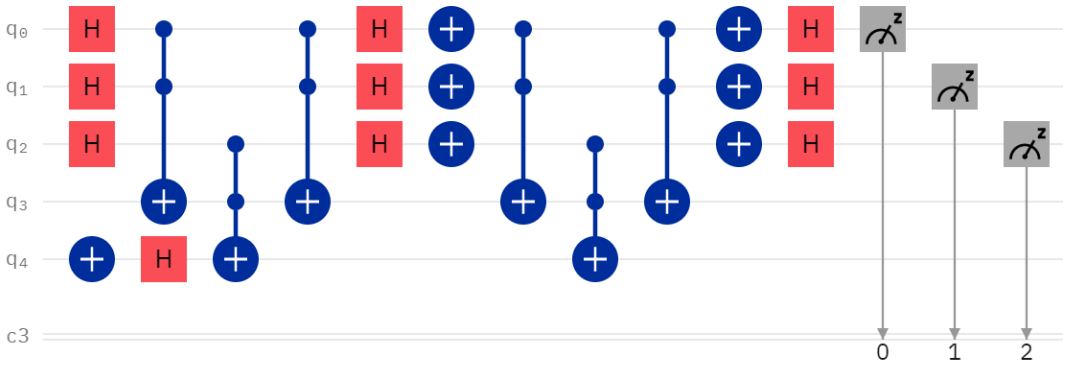


图 6.9: 三比特 Grover 算法完整电路

6.5 实验结果分析

为什么我们在双量子比特实验中得到正确答案的概率为 100%，而在三量子比特实验中得到正确答案的概率却只有约 78% 呢？这就和预言机与放大机的工作原理有关了。不妨再回顾一下二者的作用：

预言机：输入全部为$|+\rangle$ 的量子，可以将指定X_{target} 组合的概率幅变成负值，例如$X_{target} = |1,1,1\rangle$ 时，三个量子通过预言机的变化如图6.10所示。

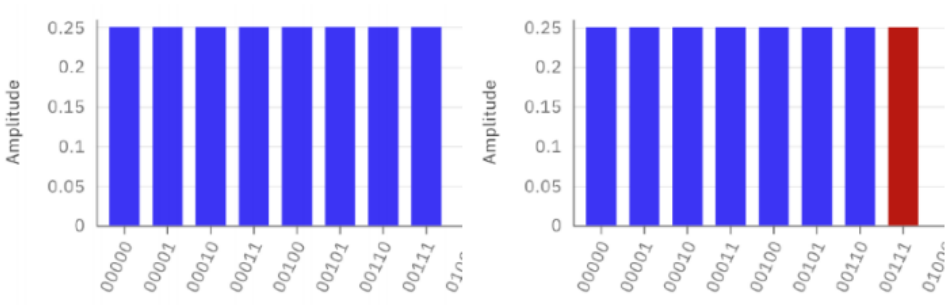


图 6.10: 预言机对全叠加态的影响

放大机：输入量子叠加态，将每个叠加态的概率幅依整体概率的平均值翻转。

(1) 两个比特时，叠加态由 4 种组合等幅组成：

$$|+,+\rangle = \frac{1}{2}(|0,0\rangle + |0,1\rangle + |1,0\rangle + |1,1\rangle)$$

这里简写为$(1,1,1,1)$。

假设$X_{target} = |1,1\rangle$，经过预言机的标记负值后，现在两个量子的叠加态变成了$(1,1,1,-1)$，整体平均值为$\frac{1}{2}$。

现在使用放大机对每个概率幅依平均值翻转，就可以得到$(0,0,0,2)$，即此时的量子态为$0 \times |0,0\rangle + 0 \times |0,1\rangle + 0 \times |1,0\rangle + \frac{1}{2} \times 2 \times |1,1\rangle = |1,1\rangle$。

对整体进行观测，有 100% 的可能得到“11”的结果。

(2) 三个比特时，叠加态由 8 种组合等幅组成：

$$|+,+,+\rangle = \frac{1}{2\sqrt{2}}(|0,0,0\rangle+|0,0,1\rangle+|0,1,0\rangle+|0,1,1\rangle+|1,0,0\rangle+|1,0,1\rangle+|1,1,0\rangle+|1,1,1\rangle)$$

这里简写为$(1,1,1,1,1,1,1,1)$。

假设$X_{target} = |1,1,1\rangle$，经预言机标记后，叠加态变成了$(1,1,1,1,1,1,1,-1)$，整体平均值为$\frac{3}{4}$。

使用放大机对概率幅依平均值翻转得$(\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{1}{2},\frac{5}{2})$，即此时量子态为：$\frac{1}{4\sqrt{2}}(|0,0,0\rangle+|0,0,1\rangle+|0,1,0\rangle+|0,1,1\rangle+|1,0,0\rangle+|1,0,1\rangle+|1,1,0\rangle+5|1,1,1\rangle)$。

对此叠加态进行观测，由于$|1,1,1\rangle$ 的概率幅为$\frac{5}{4\sqrt{2}}$，可以得到:

$$P_{1,1,1} = |\langle 1,1,1|\psi\rangle|^2 = |\frac{5}{4\sqrt{2}}|^2 = \frac{25}{32} = 78.125\%$$

总结上面两个例子，可以推测：随着量子比特数的增多，经过一次“预言机 -放大机”电路所能得到正确结果的概率会减少。面对这个问题，一方面我们可以使用重复实验来用频率逼近概率；另一方面我们可以在原电路基础上叠加“预言机 -放大机”电路。

当然，如果迭代次数超过理想次数，可能会适得其反——读者不妨自己根据“依平均值翻转”的原理推算一下关于比特数n 的理想迭代次数函数$F(n)$。

6.6 实验结果分析

6.6.1 思考题

Grover 算法的预言机电路无一例外地涉及到了一类重要的量子门：CNX 门，即前 N-1 个比特作为控制位，最后一个比特作为被控制位，若前面的量子位全为$|1\rangle$ 则反转最后一位。CX 门、Toffoli 门可以分别看作 C1X、C2X 门。在三比特 Grover 算法实验中，为了实现 C3X 门，我们引入了一个临时量子比特，然后使用了三个 Toffoli 门。试证明这两种电路等价，然后回答：不使用额外的临时量子比特并且只用 X 门、CX 门、Toffoli 门三种经典量子门能否实现 C3X 门？如果能，给出一种实现方法；如果不能，说明原因。

6.6.2 实验报告提交要求

完成本实验后，根据要求向助教提交相应的实验报告。该报告须包含以下内容：

(1) 对第6.6.1节中思考题的解答。

(2) 符合本实验要求的过程及结果。其中，

实验过程为：量子电路图、QASM 汇编代码、Python 代码三选一；

实验结果为：本地仿真结果、IBM 平台计算结果、本源平台计算结果三选一。

实验类别	X_{target}
双量子 Grover 算法电路实验	学号 hash 码第 3-4 位模 3
三量子 Grover 算法电路实验	学号 hash 码第 5-6 位模 7

表 6.3: $f(X)$ 选取规则表