量子计算
—算法篇

Quantum Computer

网址：www.qubits.top

作者：Calvin Tang

邮箱：179209347@qq.com

介绍

教程简介：
- 面向对象：量子计算初学者
- 依赖课程：线性代数，解析几何，量子力学（非必需）

知乎专栏：
https://www.zhihu.com/column/c_1501138176371011584

Github & Gitee 地址：
https://github.com/mymagicpower/qubits
https://gitee.com/mymagicpower/qubits

HHL量子算法背景概述

HHL量子算法（以三位创始人Aram Harrow，Avinathan Hassidim和Seth Lloyd命名）是一个用量子计算机解决线性问题（求解线性方程组）$Ax = b$ 最优解的算法，广泛的被应用于许多量子机器学习算法中（如支持向量机SVM，主成分分析PCA等等）。

线性方程组问题可定义为：

给定矩阵 $A \in C^{n\times n}$ 和向量 $b \in C^n$，找到 $x \in C^n$ 满足 $Ax = b$ 。

HHL算法相对于经典算法有着指数级的加速，但经典算法可以返回精确解，而HHL算法只能返回近似解。

对输入 A 和 b 的要求：

首先要求 $n\times n$ 的矩阵 A 是一个**厄米矩阵**（Hermitian Matrix），即自共轭矩阵（即A的共轭转置矩阵等于它本身），其次输入b是一个单位向量。(当 A 不是**厄米矩阵**时，原算法作者在文中也给出了构造**厄米矩阵**的方法)：

$$令\ C = \begin{bmatrix} 0 & A \\ A^\dagger & 0 \end{bmatrix} \quad 求解\ Cy = \begin{bmatrix} b \\ 0 \end{bmatrix} \quad 可得\ y = \begin{bmatrix} 0 \\ x \end{bmatrix}$$

Calvin, QQ: 179209347 Mail: 179209347@qq.com

HHL算法原理

以下内容中将默认A为自共轭矩阵。
将向量 b , x 分别归一化后采用编码到振幅上的方式映射到量子态 $|b\rangle, |x\rangle$ ，原问题转换为 $A|x\rangle = |b\rangle$.

对矩阵 A 进行谱分解有：

$$A = \sum_{j=0}^{N-1} \lambda_i |\mu_j\rangle\langle\mu_j|, \quad \lambda_j \in R$$

其中 λ_j , μ_j 为矩阵A特征值及相应的特征向量，将 $|b\rangle$ 以特征向量 $|\mu_j\rangle$ 为基，线性组合表示得到：

$$|b\rangle = \sum_{j=0}^{N-1} b_j |\mu_j\rangle, \quad b_j \in C$$

于是原方程组的解 $|x\rangle$ 为：

$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} b_j A^{-1} |\mu_j\rangle = \sum_{j=0}^{N-1} \frac{b_i}{\lambda_j} |\mu_j\rangle$$

解量子态 $|x\rangle$ ，可以转换为由量子态 $|b\rangle$ 构造。

$A|\mu_j\rangle = \lambda_j|\mu_j\rangle$
$\rightarrow A^{-1}A|\mu_j\rangle = A^{-1}\lambda_j|\mu_j\rangle$
$\rightarrow |\mu_j\rangle = A^{-1}\lambda_j|\mu_j\rangle$
$\rightarrow A^{-1}|\mu_j\rangle = \frac{1}{\lambda_j}|\mu_j\rangle$

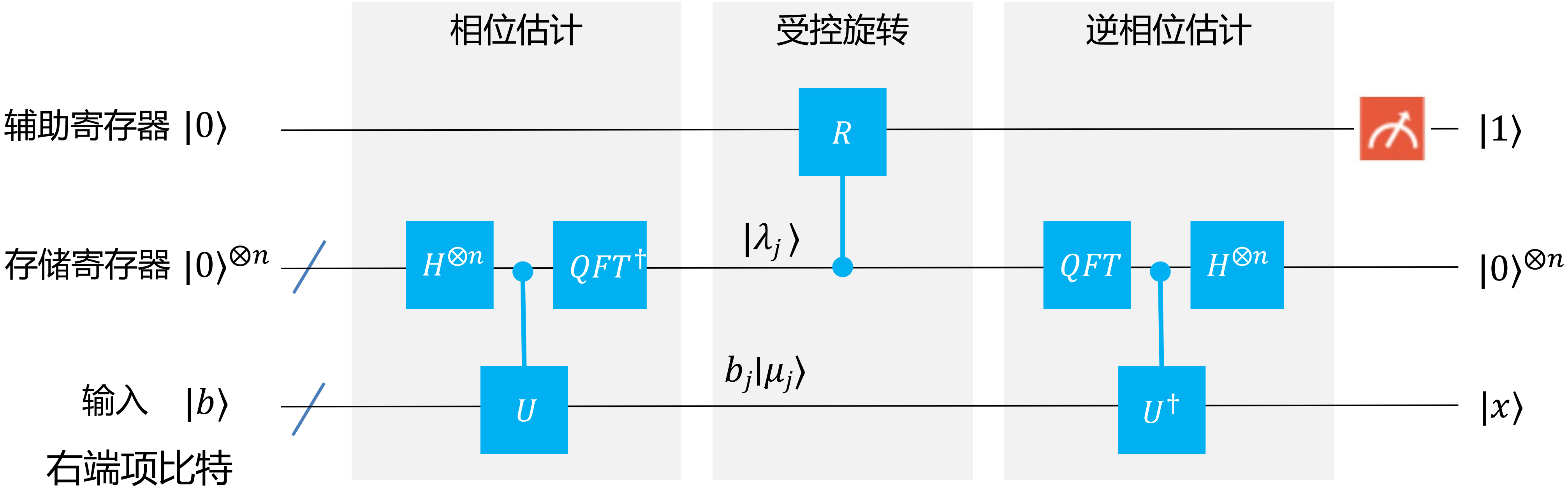
HHL算法原理

HHL算法主要包含了三大步骤，并需要使用三个寄存器：

- **相位估计**，将矩阵 A 的整数形式特征值全部转移到存储比特的基向量中。
- **受控旋转**，利用受控旋转门将 λ_j 将特征值从存储比特的基向量转移到振幅上，其中 c 是可调参数。
- **逆相位估计**，对特征存储比特及右端项比特进行逆相位估计，将存储比特振幅上的特征值合并到右端项比特上，当辅助比特测量得到特定状态时，在右端项比特上可得到解的量子态。

HHL算法量子线路

输入一个 $n\times n$ 的矩阵 $A\in\ C^{n\times n}$ ，一个 n 维向量 $b\in C^{n}$
输出 n 维向量 $x\in C^{n}$ ，$A\,x=b$
* 矩阵 A 进行谱分解 $A=\sum_{j=0}^{N-1}\lambda_i|\mu_j\rangle\langle\mu_j|,\quad \lambda_j\in R$



$$|x\rangle=\sum_{j=0}^{N-1}\frac{b_j}{\lambda_j}|\mu_j\rangle$$

输出 x 和输入 b 是在同一个寄存器中

Calvin，QQ: 179209347 Mail: 179209347@qq.com

HHL算法过程 – 初态制备（1/4）

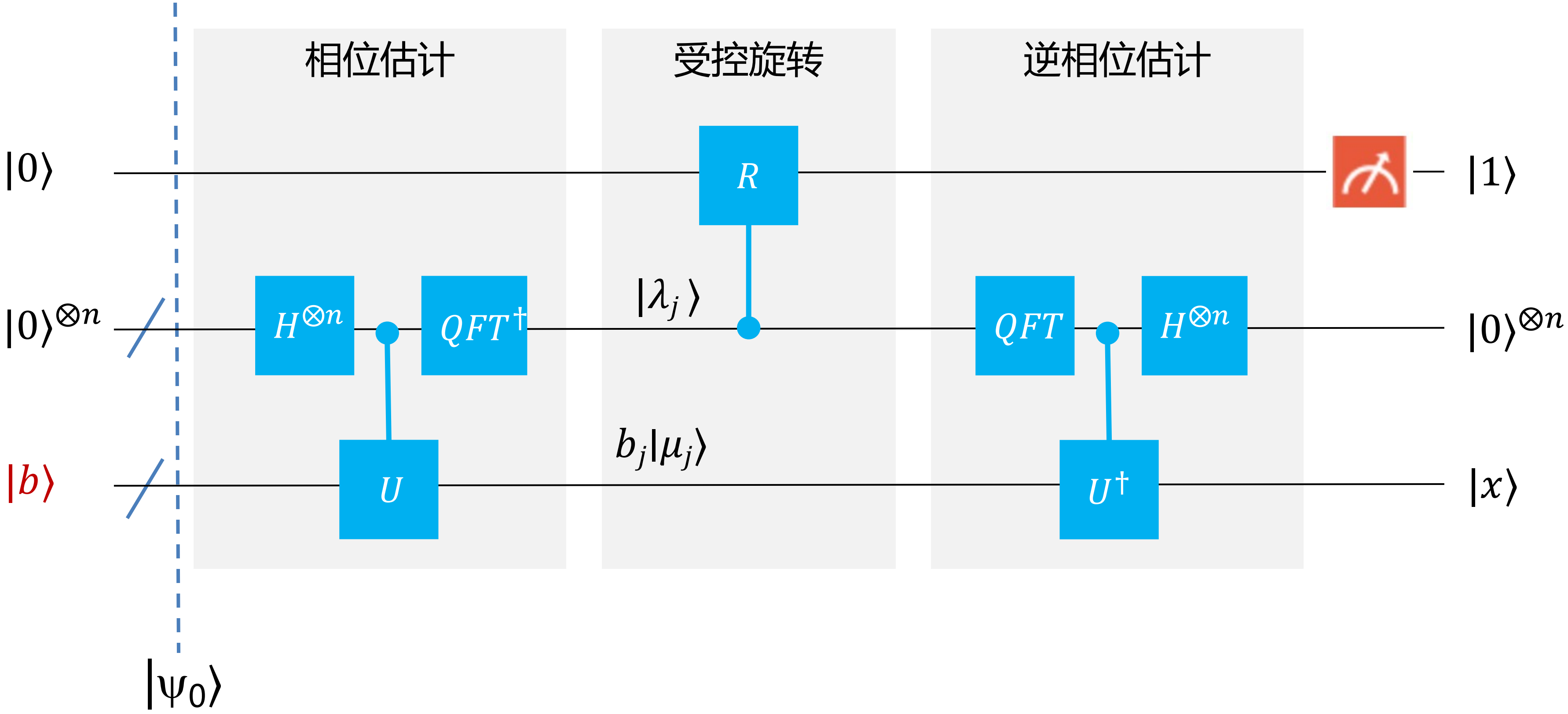
1. 初态制备：

$|b\rangle = \sum_{j=0}^{N-1} b_j \, |j\rangle$

其中：

$b = (b_0, \ldots b_{N-1})$

$\sum_{j=0}^{N-1} |b_j|^2 = 1$



HHL算法过程 - 相位估计（2/4）

通过QPE提取特征值

为了将矩阵 A 的特征值提取到解量子态的振幅，首先需要完成特征值的提取。 QPE 量子线路可以用于特征值提取。

对 $|0\rangle^{\otimes n}|b\rangle$ 进行一次 QPE 操作，得到：

$$QPE(|0\rangle^{\otimes n}|b\rangle) = \sum_{j=0}^{N-1} \mathrm{b}_j\, |\widetilde{\lambda}_j\rangle|\mu_j\rangle$$

其中 $\widetilde{\lambda}_j$ 是对应特征值 λ_j 的近似整数。 于是矩阵A的特征值信息存入到了基向量 $|\widetilde{\lambda}_j\rangle$ 中。

参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/HHL.html

HHL算法过程 - 相位估计（2/4）

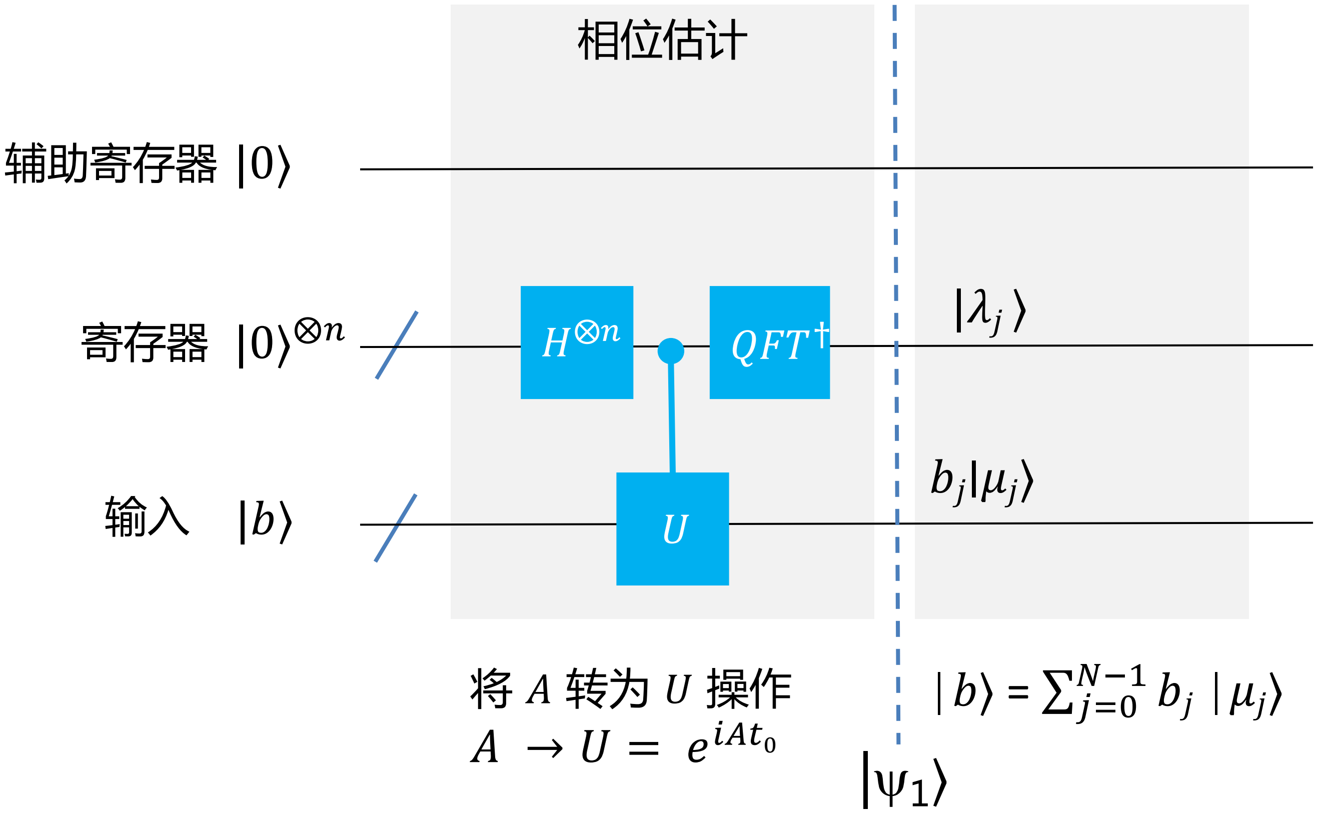
2. 相位估计（QPE）：

- 中间的寄存器会存储一系列的特征值 λ_j（存储在基态 $|\lambda_j\rangle$ 中）
- 而底部寄存器存储的输入 $|b\rangle$ 会在 A 的特征空间上进行分解，表示为：

$$|b\rangle = \sum_{j=0}^{N-1} b_j\,|\mu_j\rangle$$

$|\mu_j\rangle$ 是 A 的特征向量，λ_j 为对应特征值。

输入 b 存放在底部寄存器中，输入 A 作为相位估计中酉算子的一个组成部分。



$$|\psi_1\rangle = \sum_{j=0}^{N-1} |0\rangle|\lambda_j\rangle b_j|\mu_j\rangle$$

HHL算法过程 - 受控旋转（3/4）

3. 受控旋转：通过受控旋转转移特征值

相位估计的第二阶段，把存储在概率幅上的值提取到了基态上。而在HHL算法的第二个阶段用到了类似的技巧（不过是反向的），即通过受控旋转操作，**将基态中的值提取到了概率幅上。**

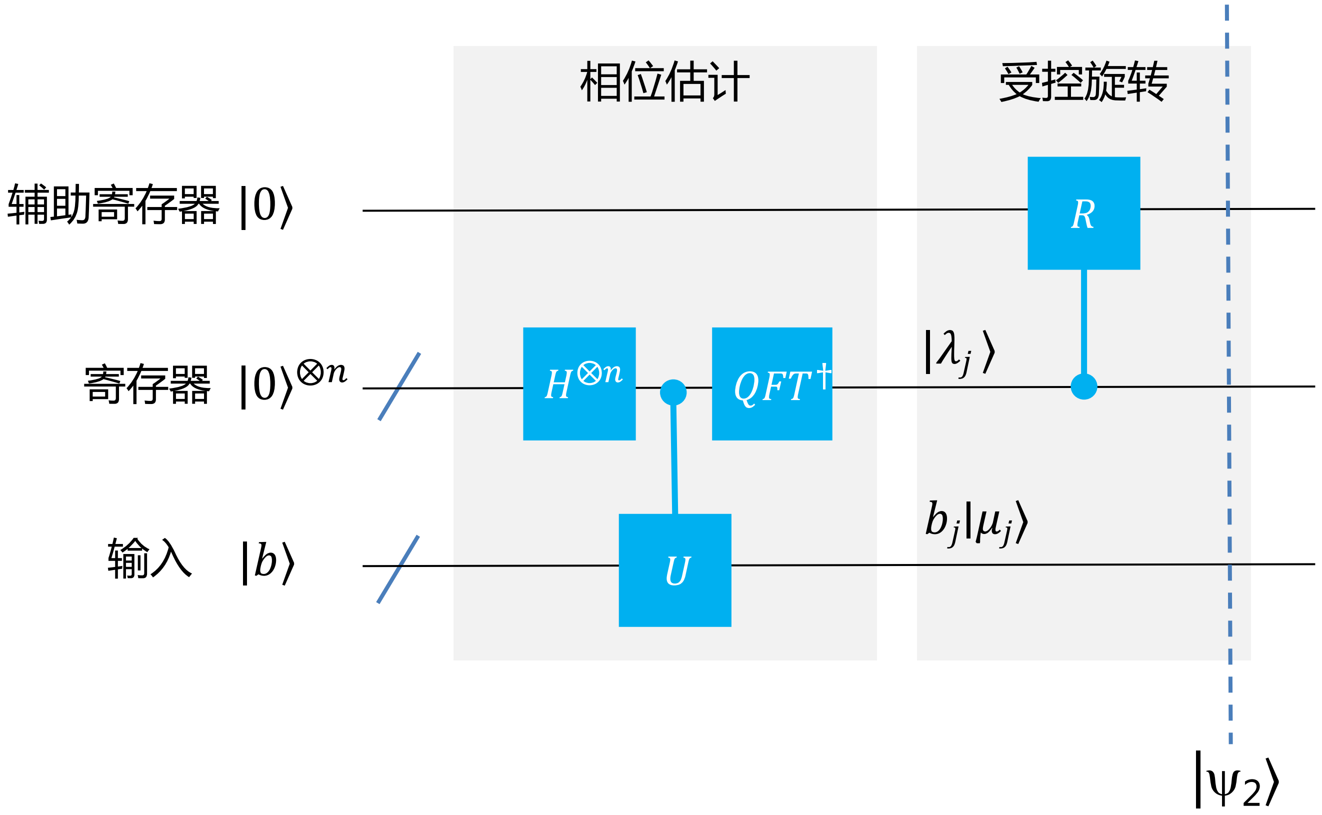
以 $|\lambda_j\rangle$ 作为控制比特对辅助量子比特进行旋转，实现了将基态值的倒数按比例提取到了对应基态的概率幅上。

$$\sqrt{1-\frac{C^2}{{\lambda_j}^2}}\ |0\rangle+\frac{C}{\lambda_j}|1\rangle$$

C为归一化系数

令 $f(\lambda_j)=C/\lambda_j$，将辅助量子比特由基态 $|0\rangle$ 映射到 $|0\rangle$ 和 $|1\rangle$ 的叠加态上，同时将函数值 $f(\lambda_j)$ 提取到了基态 $|1\rangle$的概率幅上，如下所示：

$$R\ |0\rangle=\sqrt{1-f(\lambda_j)^2}\ |0\rangle+f(\lambda_j)|1\rangle$$



$$|\psi_2\rangle=\sum_{j=0}^{N-1}\left(\sqrt{1-\frac{C^2}{{\lambda_j}^2}}\ |0\rangle+\frac{C}{\lambda_j}|1\rangle\right)|\lambda_j\rangle b_j|\mu_j\rangle$$

HHL算法过程 - 逆相位估计（4/4）

4. 逆相位估计：通过逆QPE输出结果量子态

理论上，受控旋转后的量子态已经可以通过测量得到解量子态 $|x\rangle$。

但为了避免出现 $|\mu_j\rangle$ 相同但 $|\tilde{\lambda}_j\rangle$ 不同的需要合并的量子态 $\frac{C}{\tilde{\lambda}_j} b_j|1\rangle|\tilde{\lambda}_j\rangle|\mu_j\rangle$，应当选择逆 QPE 操作来得到形如 $\frac{C}{\tilde{\lambda}_j} b_j|1\rangle|0\rangle|\mu_j\rangle$ 的结果量子态。对旋转结果进行逆 QPE，有：

$$|\psi_3\rangle = \left(I \otimes QPE^{\dagger}\right) \sum_{j=0}^{N-1} \left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_j{}^2}}\,|0\rangle + \frac{C}{\tilde{\lambda}_j}|1\rangle\right) |\tilde{\lambda}_j\rangle b_j|\mu_j\rangle$$

$$= \sum_{j=0}^{N-1} \left(b_j\sqrt{1-\frac{C^2}{\tilde{\lambda}_j{}^2}}\,|0\rangle|0\rangle|\mu_j\rangle + \frac{C}{\tilde{\lambda}_j} b_j|1\rangle|0\rangle|\mu_j\rangle\right)$$

事实上即使是这种形式的结果量子态，由于误差的存在，依然无法在第一个和第二个量子寄存器分别为 $|1\rangle, |0\rangle$ 的情况下以概率 1 得到解量子态：

$$|x\rangle = A^{-1}|b\rangle = \sum_{j=0}^{N-1} b_j A^{-1}|\mu_j\rangle = \sum_{j=0}^{N-1} \frac{b_j}{\lambda_j}|\mu_j\rangle$$

HHL算法充分利用了量子相位估计提取特征值信息的功能，巧妙构造了受控旋转门从存储比特的基向量中抓取特征值存入振幅， 最后利用逆相位估计还原存储量子比特，从而得到了振幅含特征值的方程解。

参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/HHL.html

HHL算法过程 - 逆相位估计（4/4）

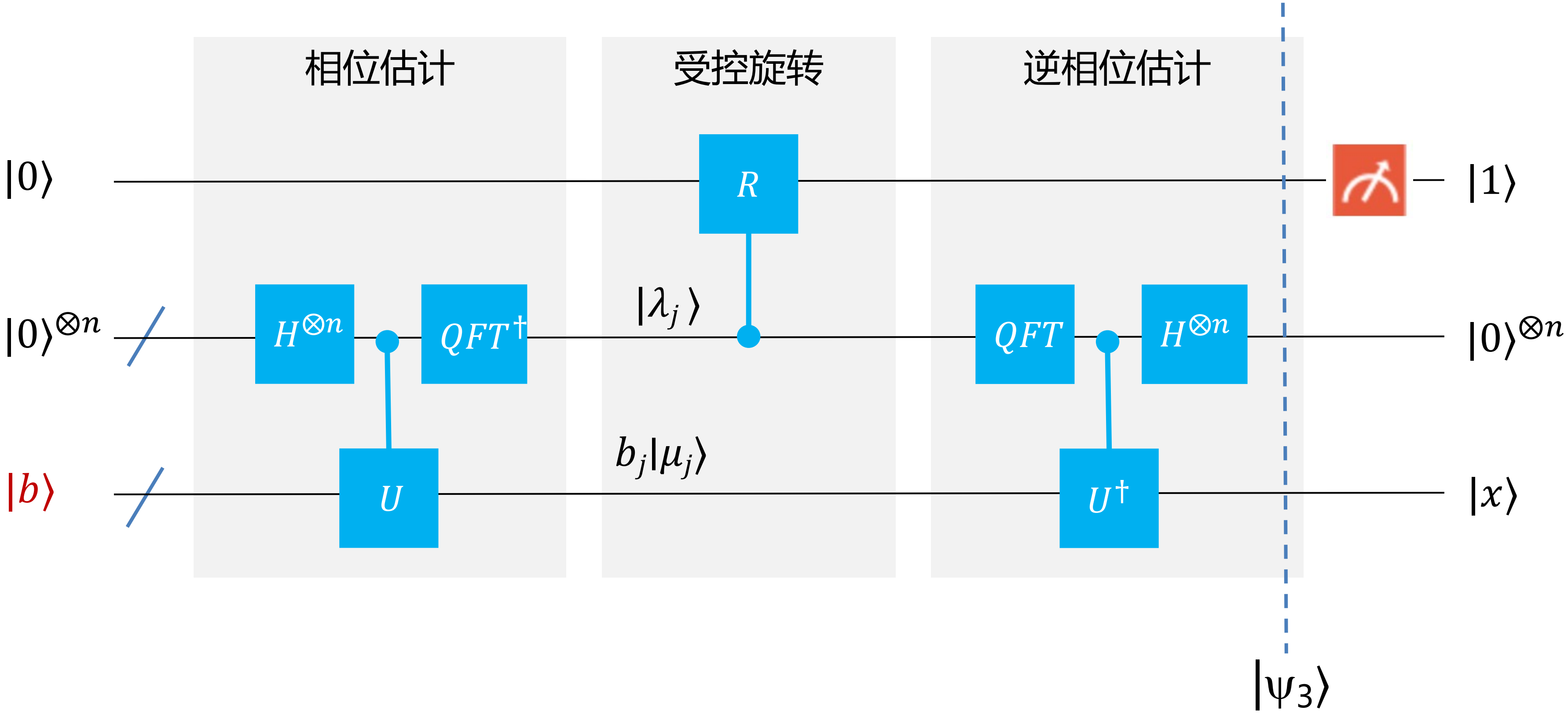
4. 逆相位估计：

执行逆相位估计，将 $|\lambda_j\rangle$ 转换为 $|0\rangle^{\otimes n}$ 。

对辅助量子比特进行测量，当测量得到 1 时，原来的寄存器由一系列 $|\lambda_j\rangle$ 的叠加变为 $f(\lambda_j)|1\rangle$ 的叠加。

基态 $|\lambda_j\rangle$ 中的值按照 $f(\lambda_j)$ 的比例被提取到了基态 $|\lambda_j\rangle$ 的概率幅上。

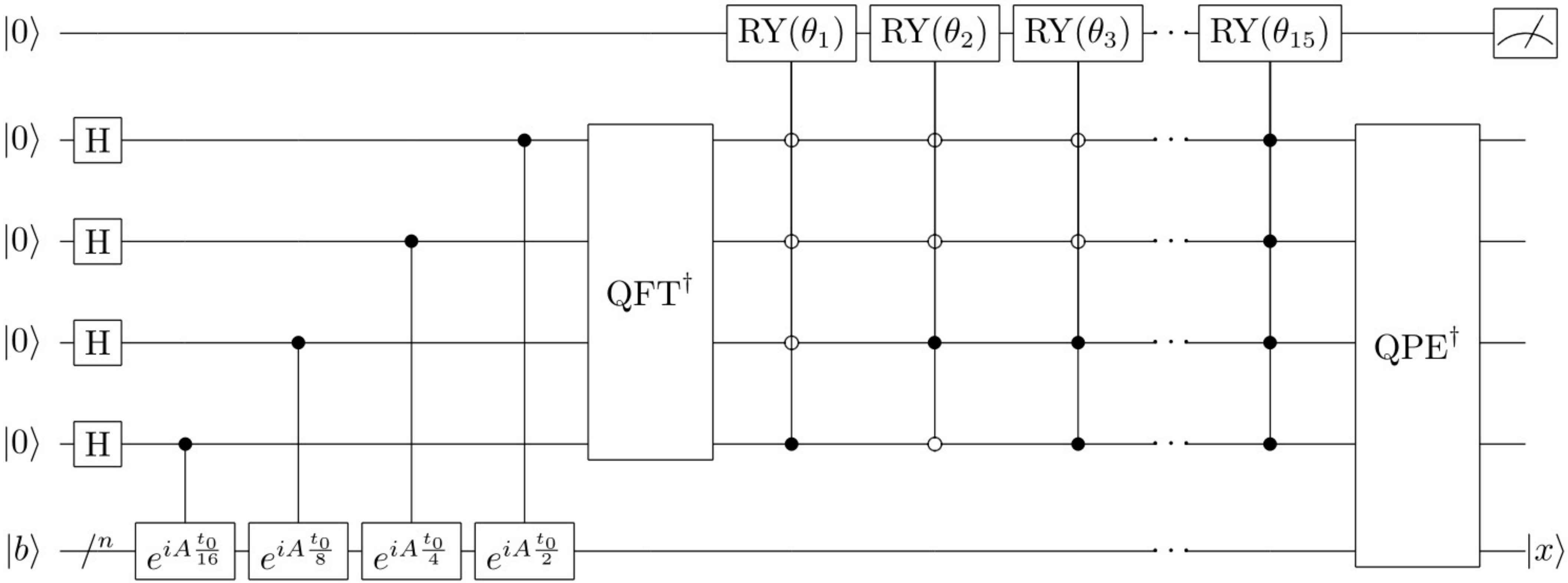
此时：$|x\rangle = \sum_{j=0}^{N-1} \frac{b_j}{\lambda_j} |\mu_j\rangle$



输出 x 和输入 b 是在同一个寄存器中

量子线路图例子

HHL算法的量子线路图如下所示



参考来源：https://pyqpanda-toturial.readthedocs.io/zh/latest/HHL.html

Thank

You