Shor’s Algorithm

第3讲：　量子逻辑电路及量子傅里叶变换

内 容

1. 不可逆计算与可逆计算
2. 量子加法器入门
3. 量子傅里叶变换

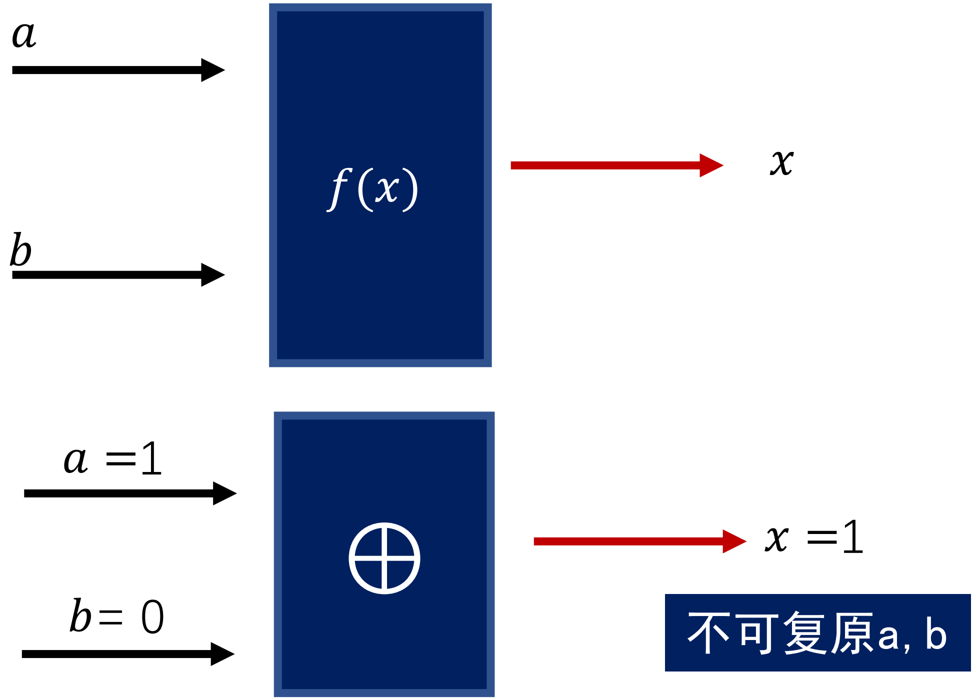
1. 不可逆计算与可逆计算

本源量子

本源量子

经典不可逆线路

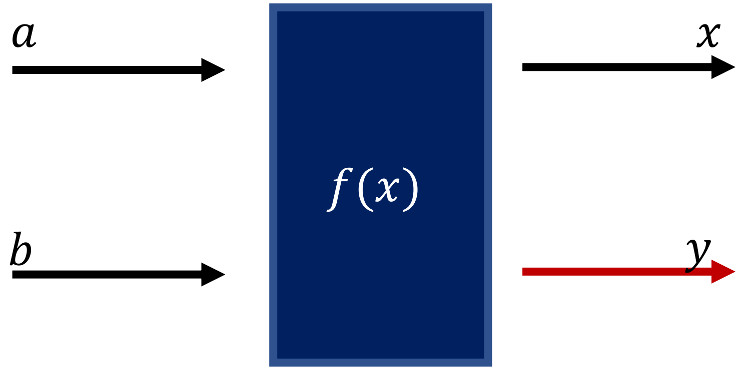
- 对于经典计算，可建立抽象的计算模型。
- 计算因为有信息擦出，从而导致，输出不可复原输入 。
- 这种不可复原输入的计算模型被称为不可逆计算。



本 源
量 子

经典可逆线路

- 任何经典不可逆计算都可以转化为可逆计算的形式！
- 可逆计算，可以通过逆计算恢复原来的输入。

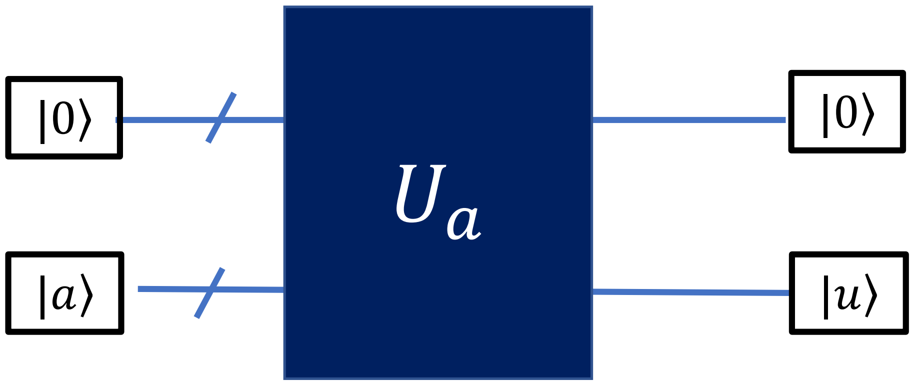


本源量子

量子线路

- 在量子计算里，西变换构成的线路是可逆的。
- 经典线路不可逆计算可以通过特殊的方式转换为量子线路。
- 通过构建黑盒子U_a来完成可逆计算，使用U_a^{-1}可以复原$|0\rangle$和$|a\rangle$

量子可逆线路

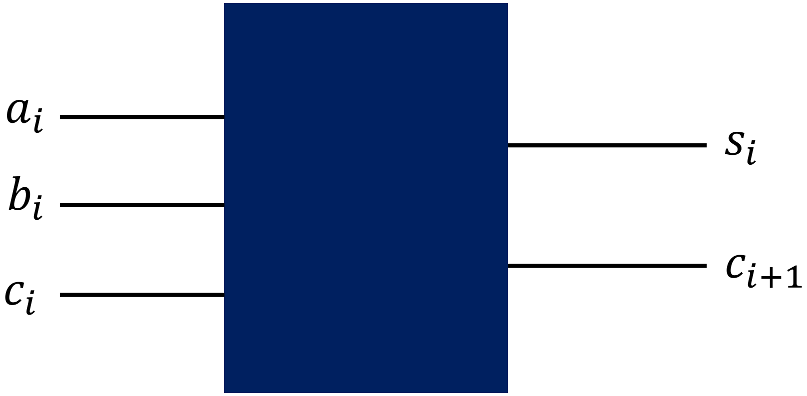


2. 量子加法器入门

本 源 量 子

本源量子

经典加法器模型



$$s_i = a_i \oplus b_i \oplus c_i$$

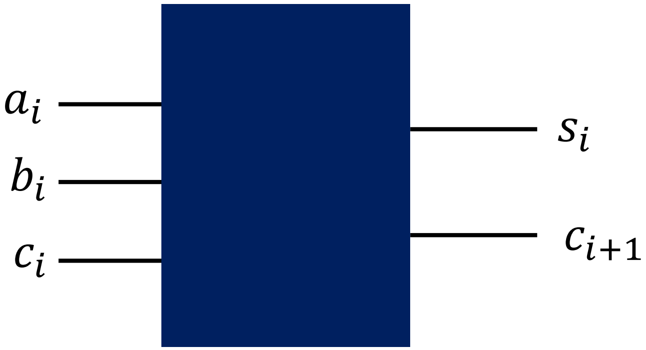
$$c_{i+1} = a_i b_i \oplus b_i c_i \oplus a_i c_i$$

输入			输出	
a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	1	1	1	1
1	0	1	0	1
1	1	0	0	1
0	1	1	0	1

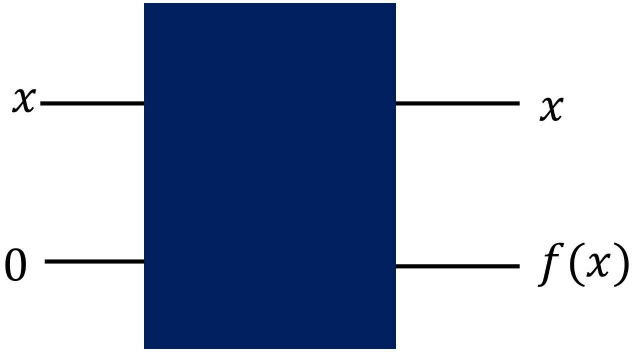
真值表

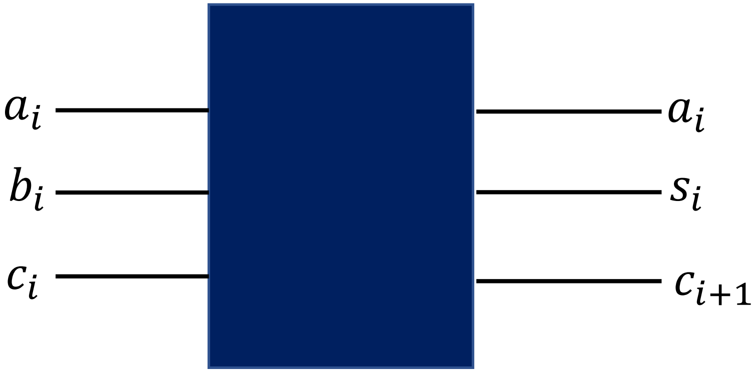
本源
量子

量子加法器假想模型



不可逆变换，不可实现复原操作



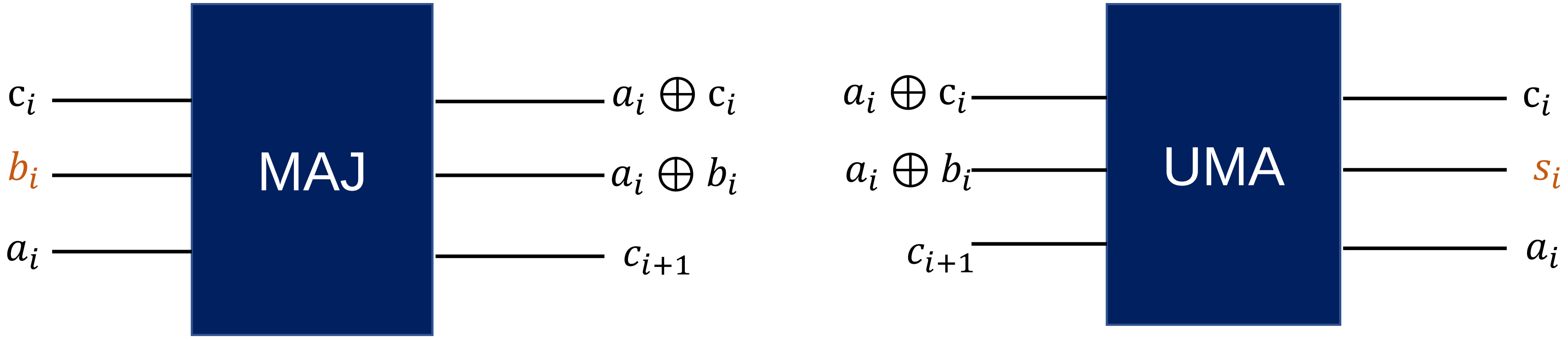


构建，通过一次酉变换，同时得到c_{i+1}与s_i!

$$s_i \quad = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i b_i \oplus b_i c_i \oplus a_i c_i$$

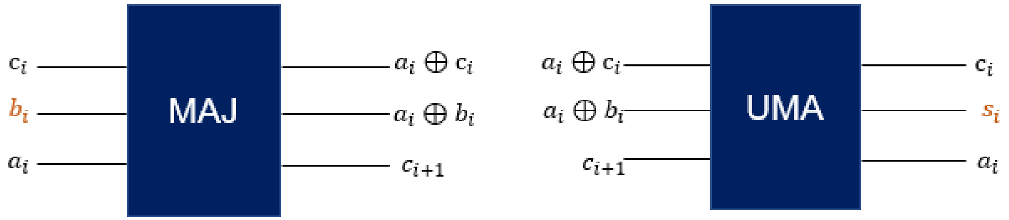
量子加法器模型

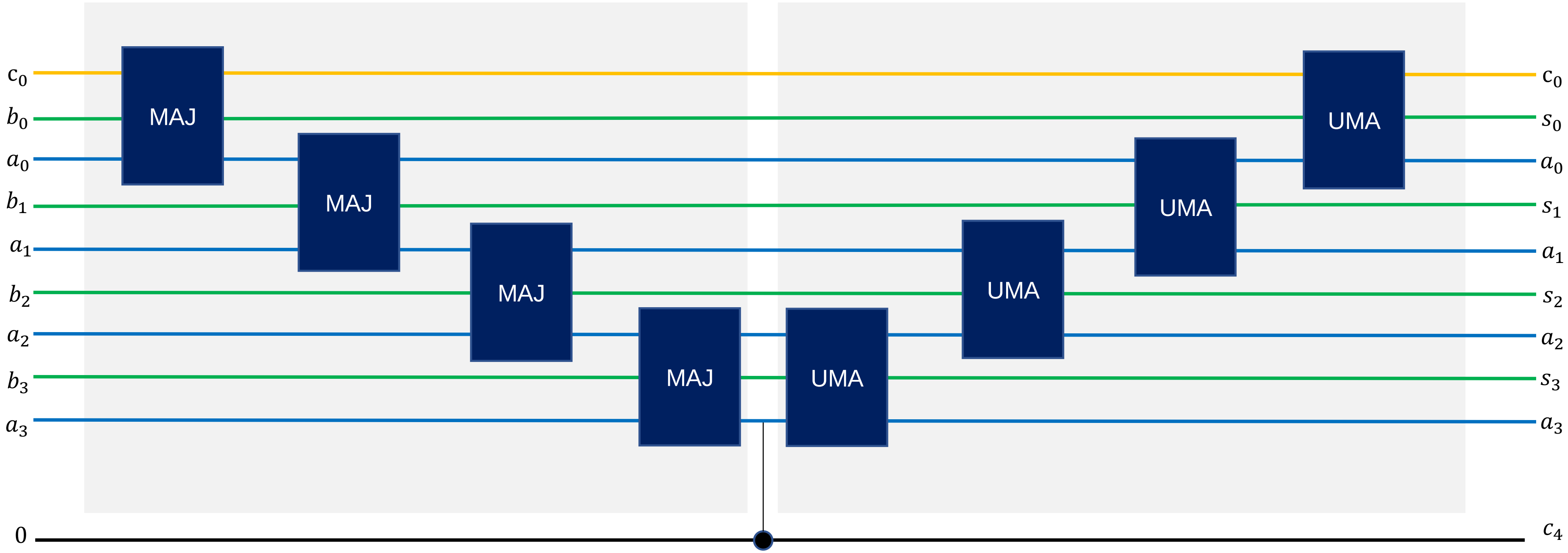


量子加法器里的MAJ模块和UMA模块

本 源
量 子

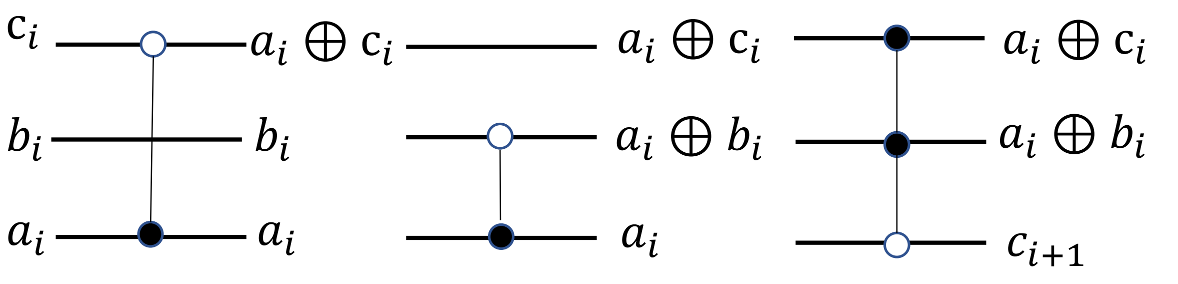
量子加法器模型



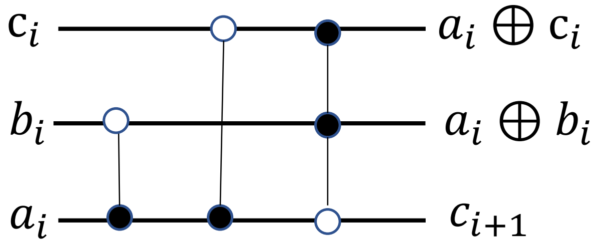


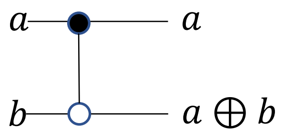
本源量子

MAJ单元的实现

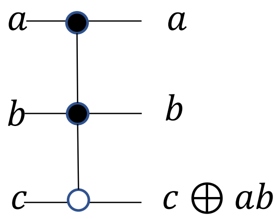


$$
\begin{aligned}
c_{i+1} &= a_ib_i \oplus b_ic_i \oplus c_ia_i \\
&= a_i \oplus a_ia_i \oplus a_ib_i \oplus b_ic_i \oplus c_ia_i \\
&= a_i \oplus (a_i \oplus c_i)(a_i \oplus b_i)
\end{aligned}
$$





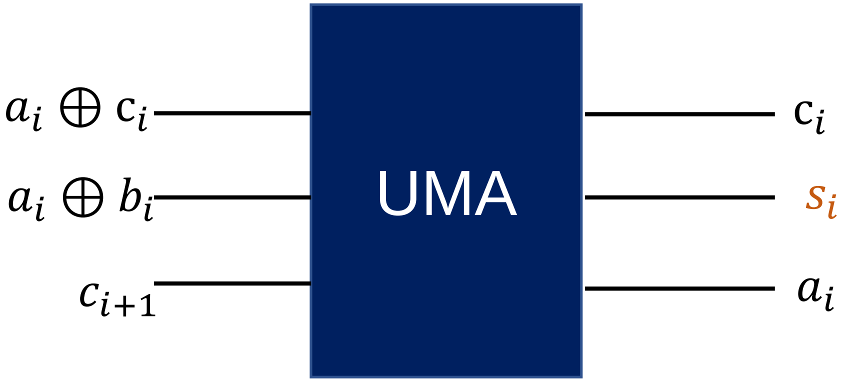
CNOT门

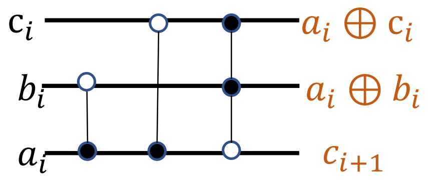


Toffoli门

本源
量子

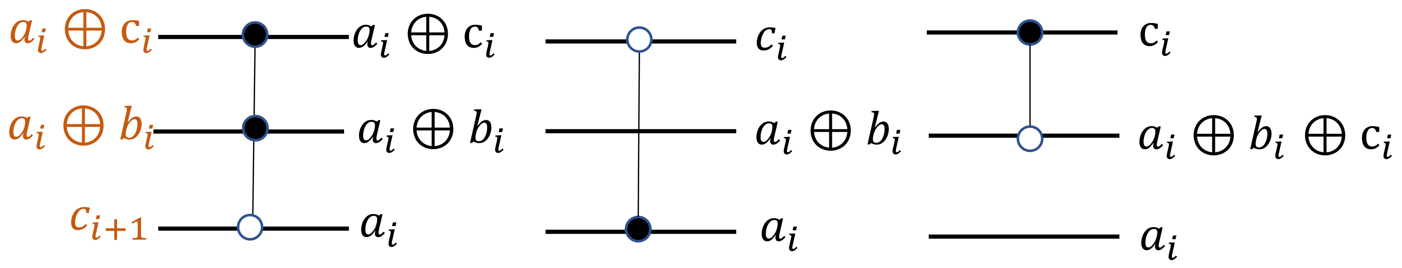
UMA单元的实现

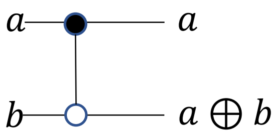




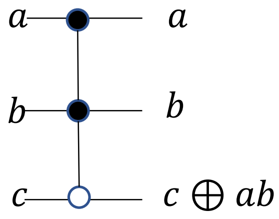
MAJ单元

$$
\begin{aligned}
c_{i+1} &= a_ib_i \oplus b_ic_i \oplus c_ia_i \\
&= a_i \oplus a_ia_i \oplus a_ib_i \oplus b_ic_i \oplus c_ia_i \\
&= a_i \oplus (a_i \oplus c_i)(a_i \oplus b_i)
\end{aligned}
$$

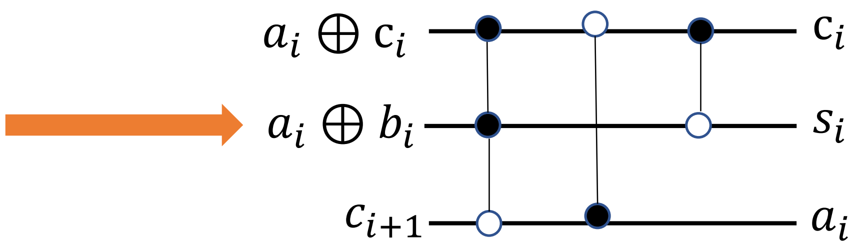




CNOT门

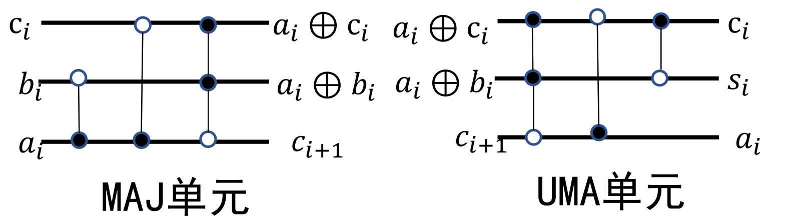


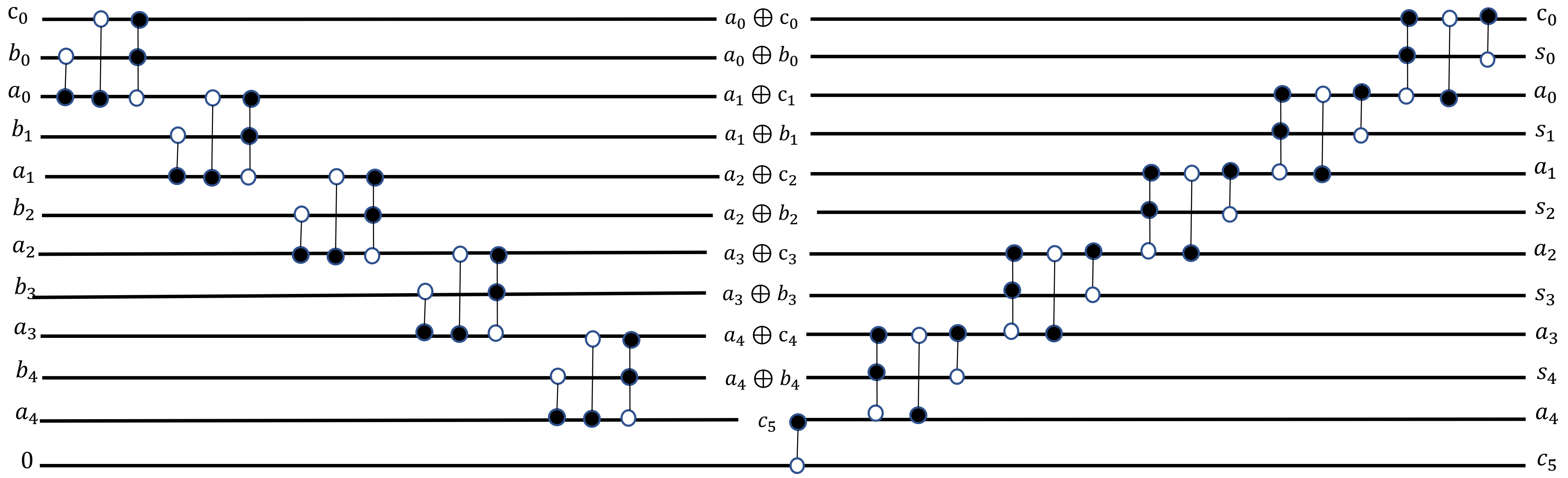
Toffoli门



本源量子

量子加法器电路





$$b_0 = a_0 \oplus b_0 \rightarrow c_0 = c_0 \oplus a_0 \rightarrow a_0 = c_{i+1} \rightarrow b_1 = a_1 \oplus b_1 \rightarrow c_1 = c_1 \oplus a_1 \rightarrow a_1 = c_2 \ldots$$

完成n位的加法器，需要长度为6n+1的时序电路

3. 量子傅里叶变换

本源量子

本源
量子

快速傅里叶变换（FFT）

信号

$$f(\mathrm{t}) \xrightarrow{\text{FFT}} F(\mathrm{t})$$

频域

Frequency domain

时域

Time domain

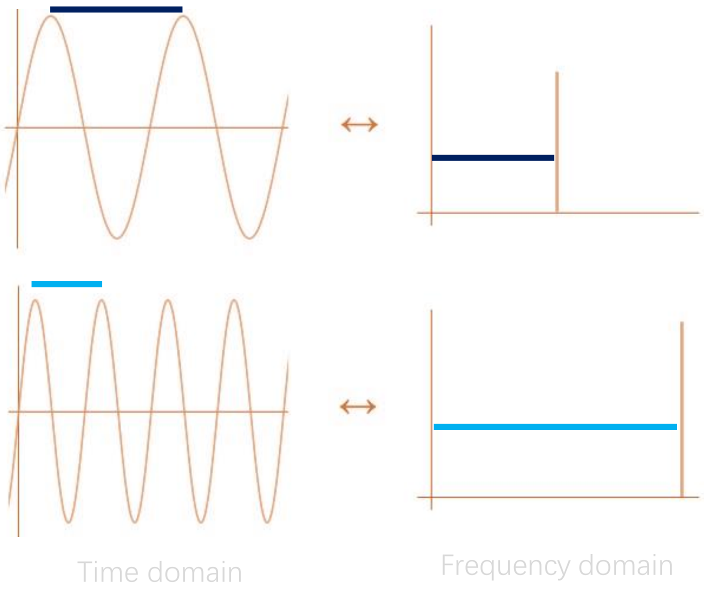
快速傅里叶变换（FFT）

信号

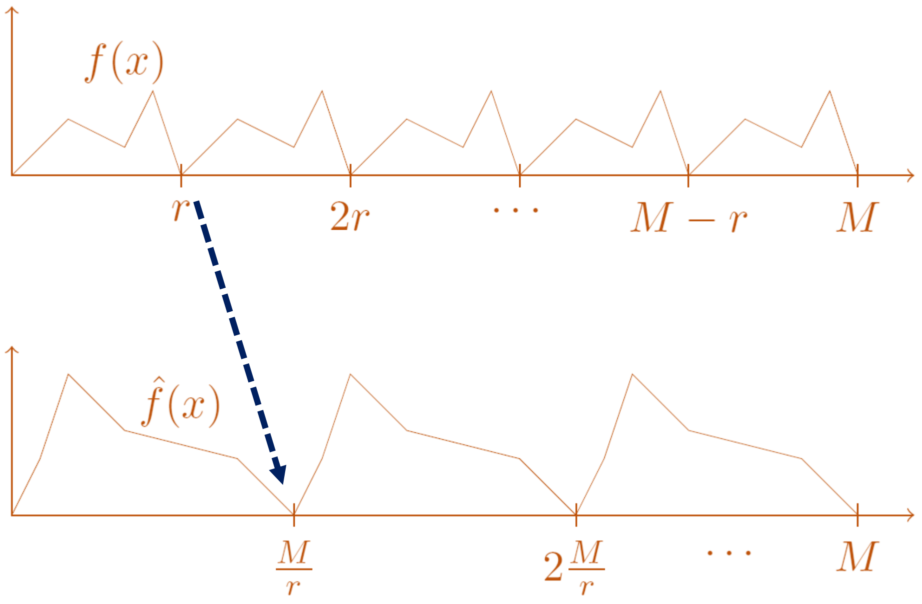
f(t) $\xrightarrow{\text{FFT}}$ F(t)

频域 → 时域

Frequency domain → Time domain



快速傅里叶变换（FFT）



如果函数在时域中具有周期r，则变换函数在频域中具有$\frac{1}{r}$的周期变化。

傅里叶变换在数学上定义为：

$$y_k = \sum_{j=0}^{N-1} e^{\frac{2\pi ikj}{N}} x_j.$$

由此可见，我们可以用量子计算中的一些相位门来表达傅里叶变换。

如：

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix}$$

量子傅里叶变换（QFT）

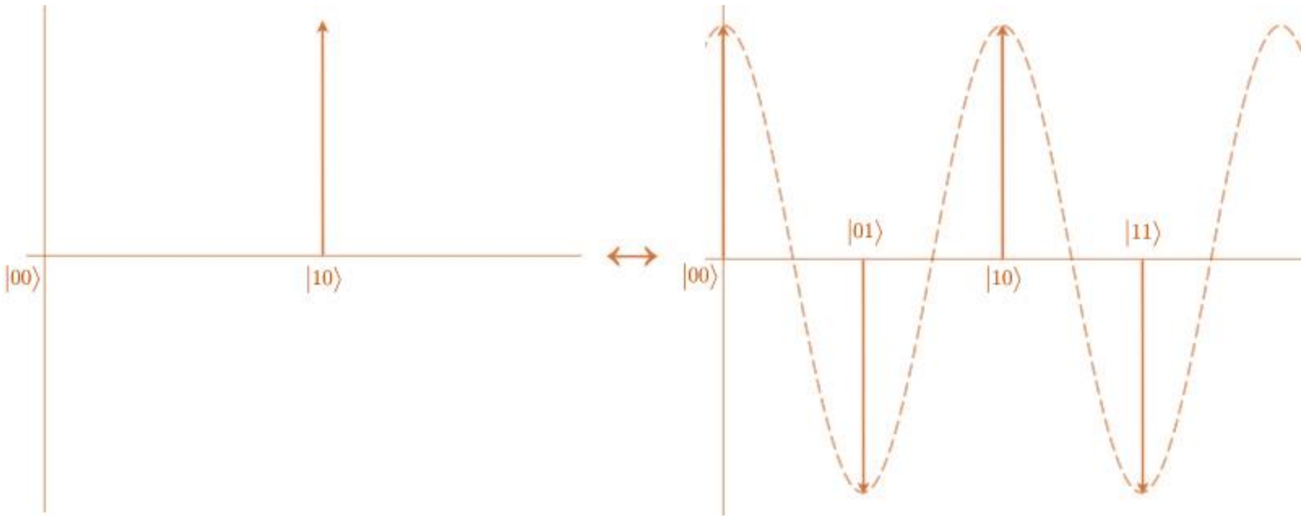
量子傅里叶变换定义为：

$$\sum_j \alpha_j |j\rangle \rightarrow \sum_k \tilde{\alpha}_k |k\rangle$$

其中，

$$\tilde{\alpha}_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jk/N} \alpha_j$$

量子傅里叶变换是可逆的！



$$|10\rangle \leftrightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

将QFT表示为线性算子

$$QFT = \frac{1}{\sqrt{M}}\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{M-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2M-2} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3M-3} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{M-1} & \omega^{2M-2} & \omega^{3M-3} & \cdots & \omega^{(M-1)(M-1)} \end{pmatrix}$$

比如: $M = 4$， $\omega^0 = 1, \omega^1 = i, \omega^2 = -1, \omega^3 = -i,$

$$\frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle) = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

$$\left|\hat{f}\right\rangle = \frac{1}{4}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & i\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$\omega^4 = 1$$

$$\frac{1}{4}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & i\end{pmatrix}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

$$\frac{1}{2}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & i\end{pmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

让我们用不同的输入，重复计算，结果如下

本源量子

量子傅里叶变换

量子版本,介绍一下符号:

$$j = j_1 j_2 \cdots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n$$

$$0.j_l j_{l+1} \cdots j_m = j_l/2 + j_{l+1}/4 + \cdots + j_m/2^{m-l+1}$$

Eg: $j = 2$. 使用二进制表达为10, $j_1 = 1, j_2 = 0$.

10　　2

$$j = j_1 j_2 \cdots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n$$

Eg: $j = 0.5$(二进制0.10).

$$0.j_l j_{l+1} \cdots j_m = j_l/2 + j_{l+1}/4 + \cdots + j_m/2^{m-l+1}$$

0.10　　0.5

通过证明,可以迭代执行QFT为:

$$|j_1 \cdots j_n\rangle$$

$$\updownarrow$$

$$\frac{\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)}{2^{n/2}}$$

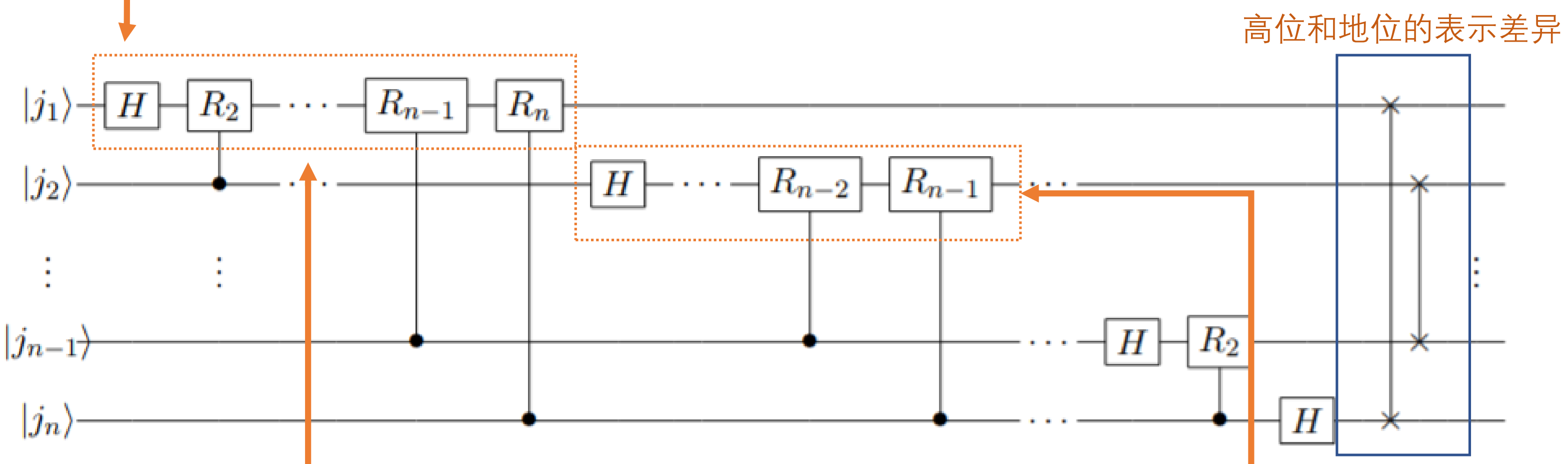
CR量子门在控制位为$|1\rangle$时做控制相位变换操作,受控运算符的矩阵形式

$$\hat{R}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

本源
量子

QFT可以通过一系列受控R门实现，如下所示：

$$\frac{1}{2^{1/2}}\left(\boxed{|0\rangle + e^{2\pi i 0.j_1}|1\rangle}\right)|j_2\dots j_n\rangle$$



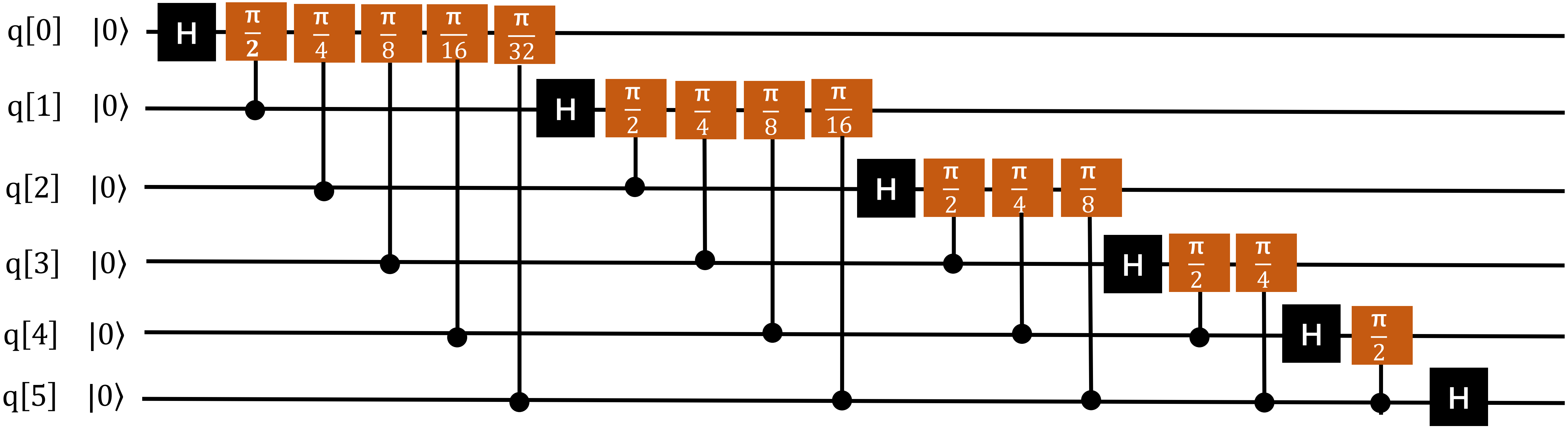
$$\frac{1}{2^{1/2}}\left(\boxed{|0\rangle + e^{2\pi i 0.j_1 j_2\dots j_n}|1\rangle}\right)|j_2\dots j_n\rangle$$

$$\frac{1}{2^{2/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2\dots j_n}|1\rangle\right)\left(\boxed{|0\rangle + e^{2\pi i 0.j_2\dots j_n}|1\rangle}\right)|j_3\dots j_n\rangle$$

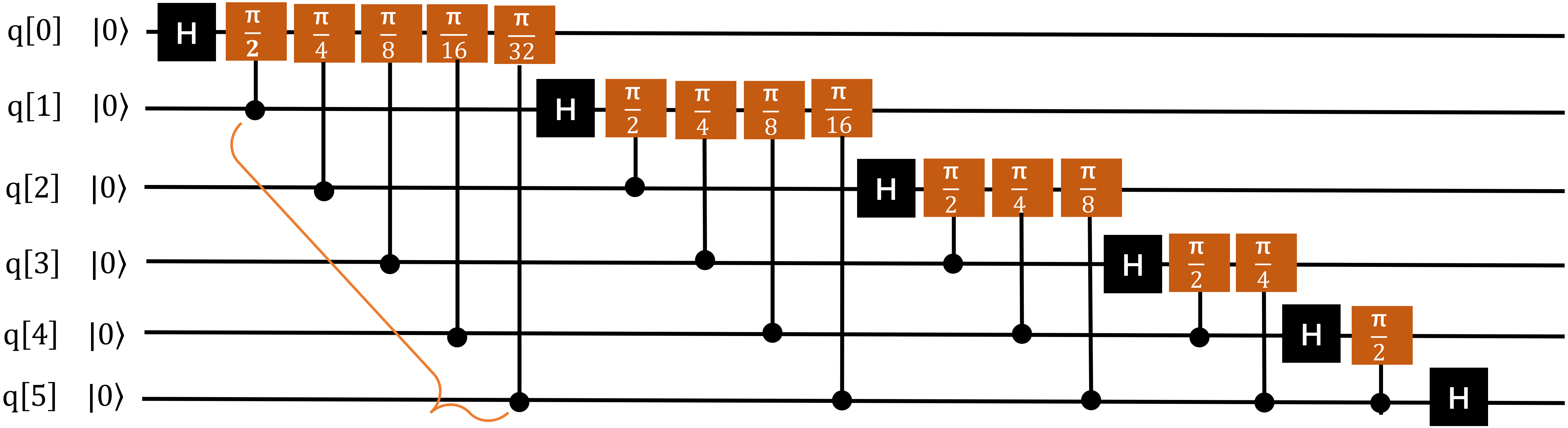
$$\underbrace{\begin{pmatrix}1 & 0\\ 0 & e^{2\pi i/2^2}\end{pmatrix}}_{\hat{R}_2}\begin{pmatrix}1\\ j_2\end{pmatrix}=\begin{pmatrix}1\\ e^{2\pi i\,\mathbf{0.0j_2}}\end{pmatrix}$$

考虑j_2使用多个R_2

$$\hat{R}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$$

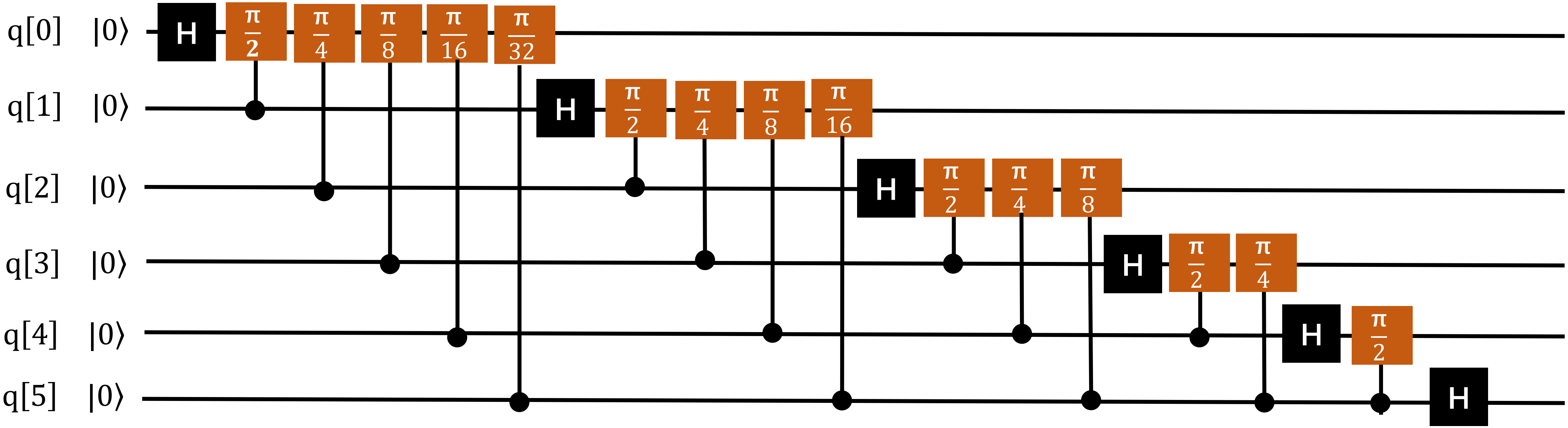


本源量子



初始化如果都是0，则控制不工作。线路等价于对所有比特做H门操作。

本源
量子


q[0] |0⟩
q[1] |0⟩
q[2] |0⟩
q[3] |0⟩
q[4] |0⟩
q[5] |0⟩
H
π/2
π/4
π/8
π/16
π/32
H
π/2
π/4
π/8
π/16
H
π/2
π/4
π/8
H
π/2
π/4
H
π/2
H

本 源
量 子

PyQPanda演示

获取QFT所需的比特长度

分别对每个比特位执行操作

使用SWAP门操作交换高低位

```
def qft(qlist):
    circ = QCircuit()

    qnum = len(qlist)
    for i in range(0, qnum):
        circ.insert(H(qlist[qnum-1-i]))
        for j in range(i + 1, qnum):
            circ.insert(CR(qlist[qnum-1-j], qlist[qnum-1-i], m.pi/(1 << (j-i))))

    for i in range(0, qnum//2):
        circ.insert(CNOT(qlist[i], qlist[qnum-1-i]))
        circ.insert(CNOT(qlist[qnum-1-i], qlist[i]))
        circ.insert(CNOT(qlist[i], qlist[qnum-1-i]))

    return circ
```

本源
量子

追本溯源 高掌远跖

https://www.originqc.com.cn