Quantum Computing

Chao Liang

School of Computer Science
Wuhan University

Review: Lecture 6

- Quantum Gates
  1. Bits and Qubits
     - Definitions and their relation
  2. Classical Gates
     - NOT, AND, OR, and NAND gates
     - 功能完备与通用门
     - Sequential and Parallel Operations
  3. Reversible Gates
     - Controlled-NOT, Toffoli, and Fredkin gates
  4. Quantum Gates
     - Definition
     - Phase shift, Controlled-U, and Deutsch gates
     - Limitations

Lecture 7: Quantum Algorithms

1

Deutsch's algorithm

- The Deutsch oracle problem
- Reversible and irreversible operators
- Deutsch’s algorithm
- Discussion

2

Deutsch-Jozsa algorithm

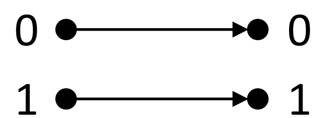
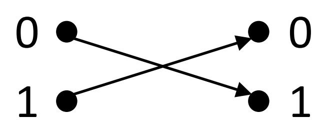
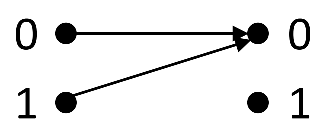
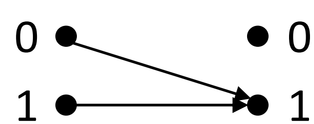
- Hadamard matrix and Kronecker product
- N-bit Deutsch oracle problem
- Deutsch-Jozsa algorithm

1. Deutsch's algorithm

- Basic framework of quantum algorithms
  - The system will start with the qubits in a particular classical state
  - From there the system is put into a superposition of many states
  - This is followed by acting on this superposition with several unitary operations
  - And finally, a measurement of the qubits

1. Deutsch's algorithm

■ Balanced and constant functions

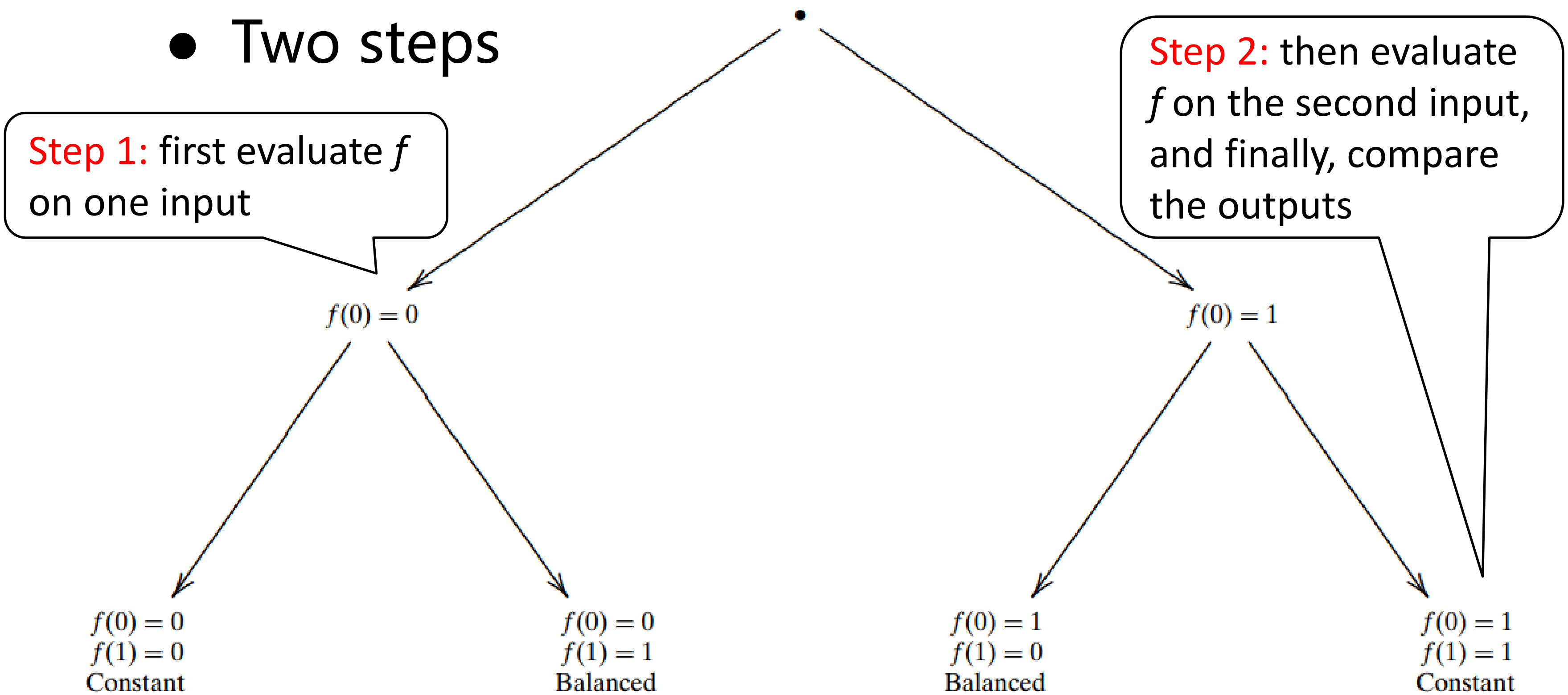
balanced	identity	$f(x) = x$		$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$
	negation (bit flip/X gate)	$f(x) = \neg x$		$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$
constant	constant-0	$f(x) = 0$		$\begin{pmatrix}1 & 1\\0 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1 & 1\\0 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$
	constant-1	$f(x) = 1$		$\begin{pmatrix}0 & 0\\1 & 1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0 & 0\\1 & 1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$

1. Deutsch's algorithm

- The Deutsch oracle problem
  - Given a function *f*: {0, 1} -> {0, 1} as a black box (BB), where one can evaluate an input, but cannot "look inside" and "see" how the function is defined, determine if the function is balanced or constant.

1. Deutsch's algorithm

- Solution with a classic computer
  - Two steps



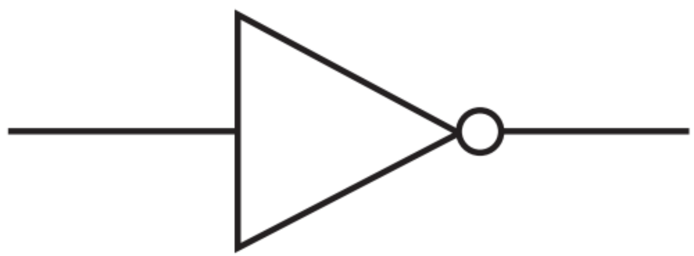
1. Deutsch's algorithm

- How about a quantum computer?
  - Quantum computer use only reversible operations
    - Given the operation and output value, find the input
  - Operations which permute（改序） are reversible
    - e.g., X gate, CNOT gate, H gate, identity and negation
  - Operations which erase & overwrite are irreversible
    - Constant-0 and constant-1 are not reversible

下面咱们先看可逆的操作（后面Deutsch algorithm要用到），再考虑不可逆的操作（咱们想办法把它对应到可逆门）

1. Deutsch's algorithm

- Reversible operators
  - NOT (X) gate
    - $|x\rangle \mapsto |\neg x\rangle$

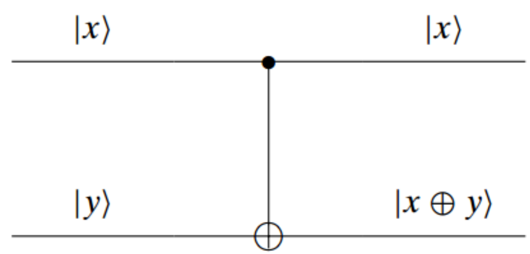


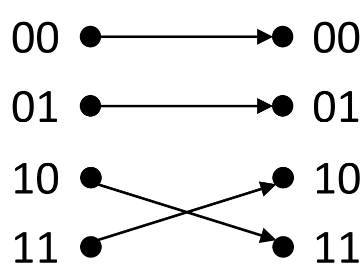
$$\mathrm{NOT} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

1. Deutsch's algorithm

- Reversible operators
  - CNOT (CNOT) gate
    - $|x,\ y\rangle \mapsto |x,\ x \oplus y\rangle$
      - Operation on a pair of bits
      - $|x\rangle$ is the control bit
      - $|y\rangle$ is the target bit





	00	**01**	**10**	**11**
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

1. Deutsch's algorithm

- Reversible operators
  - Hadamard (H) gate
    - Maps a 0- or 1-bit into exactly equal superposition, and back (operations are their own inverse!)

$$\mathbf{H}|0\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \Leftrightarrow \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

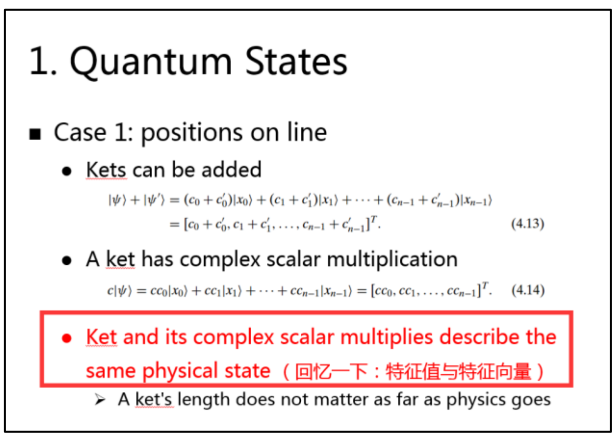
$$\mathbf{H}|1\rangle = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \Leftrightarrow \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- We can transition out of superposition without measurement
- We can structure quantum computation deterministically instead of probabilistically

We can transition into superposition from classic state

1. Deutsch's algorithm

- Reversible operators
  - The unit circle state machine
    - ➢ Unit circle
    - ➢ 8 states
    - ➢ 4 different states (why?)



1. Quantum States

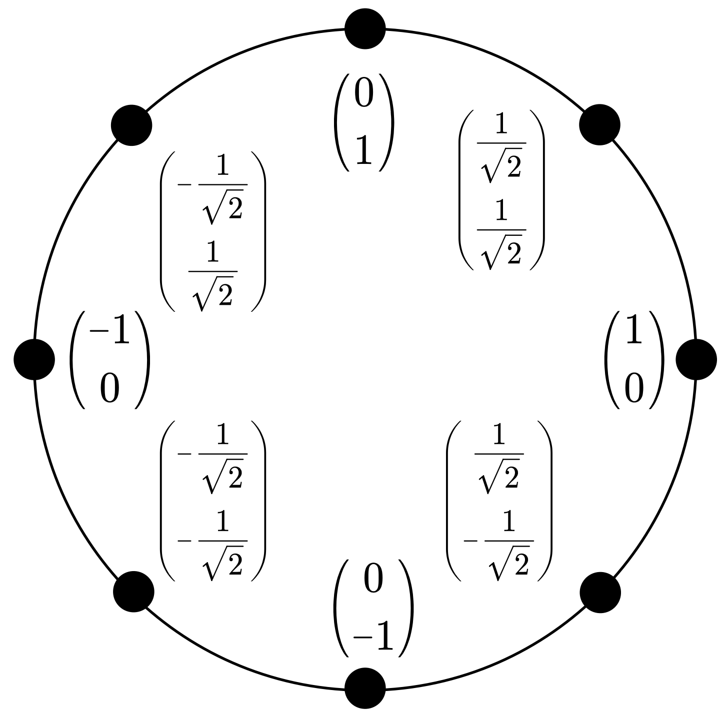
- Case 1: positions on line
  - Kets can be added

$$|\psi\rangle + |\psi'\rangle = (c_0 + c_0')|x_0\rangle + (c_1 + c_1')|x_1\rangle + \cdots + (c_{n-1} + c_{n-1}')|x_{n-1}\rangle = [c_0 + c_0', c_1 + c_1', \ldots, c_{n-1} + c_{n-1}']^T. \quad (4.13)$$

  - A ket has complex scalar multiplication

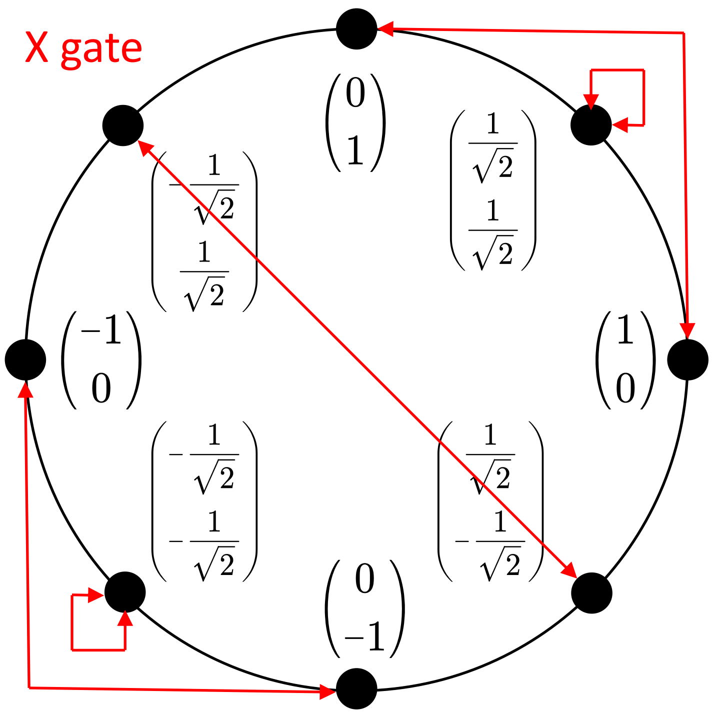
$$c|\psi\rangle = cc_0|x_0\rangle + cc_1|x_1\rangle + \cdots + cc_{n-1}|x_{n-1}\rangle = [cc_0, cc_1, \ldots, cc_{n-1}]^T. \quad (4.14)$$

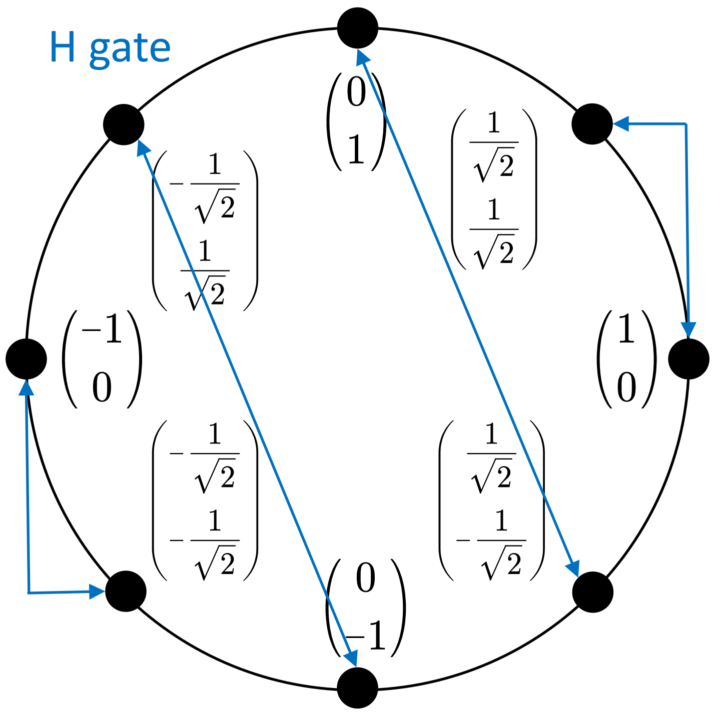
  - Ket and its complex scalar multiplies describe the same physical state（回忆一下：特征值与特征向量）
    - ➢ A ket's length does not matter as far as physics goes



1. Deutsch's algorithm

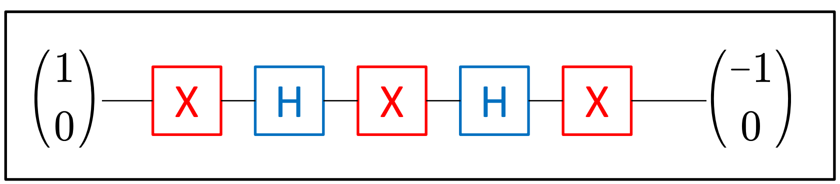
- Reversible operators
  - The unit circle state machine

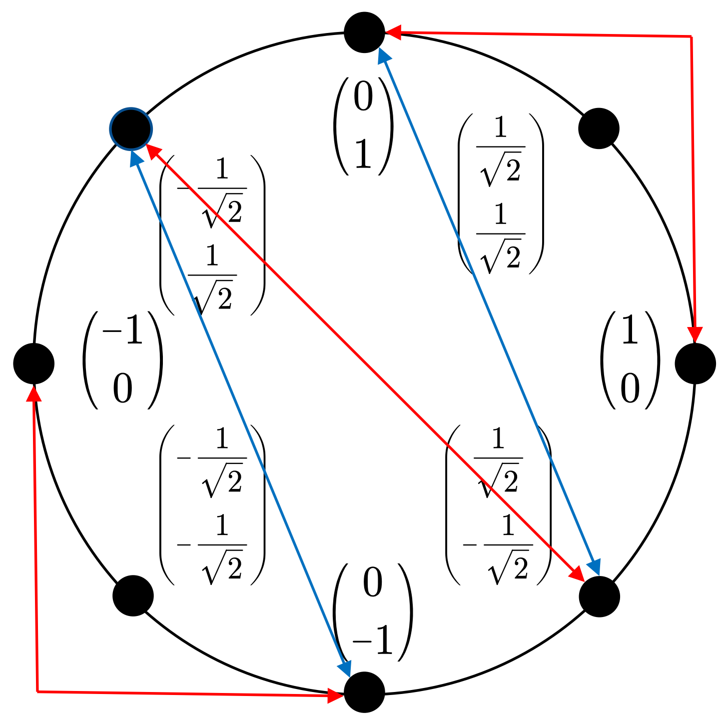




1. Deutsch's algorithm

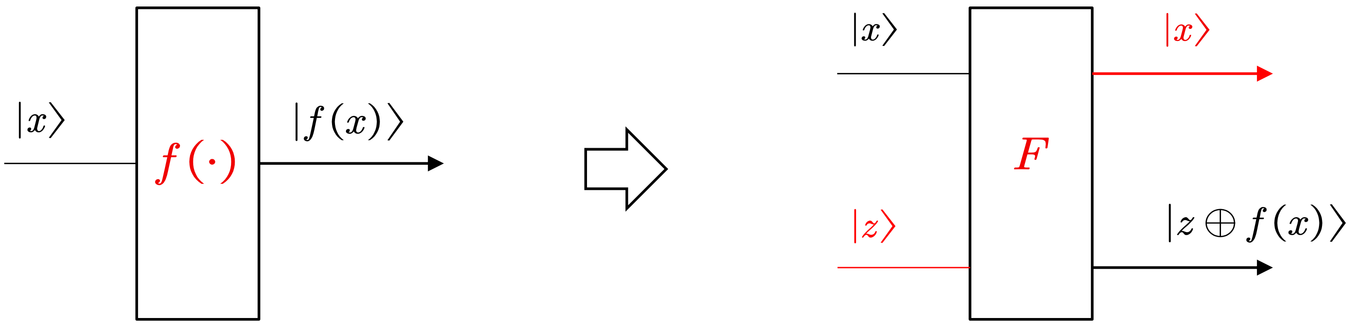
- Reversible operators
  - The unit circle state machine





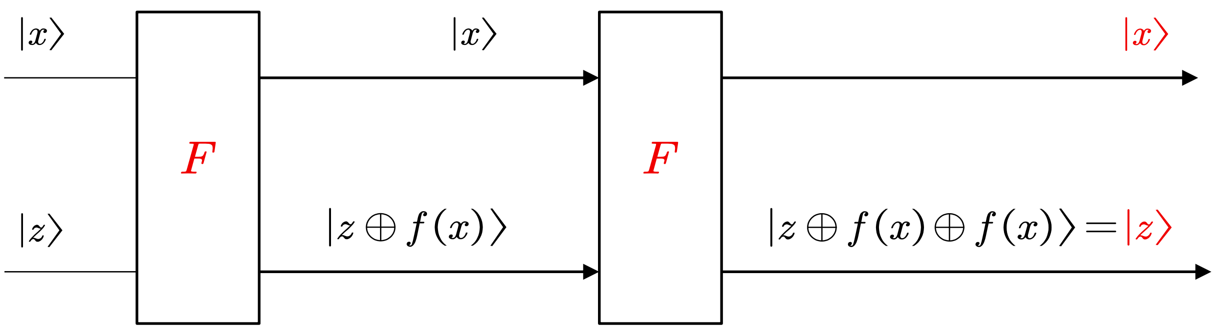
1. Deutsch's algorithm

- Irreversible operators
  - Conversion to reversible operators
    - Add an additional output qubit, output $|x\rangle$, which recovers input $|x\rangle$
    - Add an additional input $|z\rangle$, which can be recovers by applying $|z\rangle = |z \oplus f(x) \oplus f(x)\rangle$ given the operation $f(\cdot)$ and output $|x\rangle$



补充材料

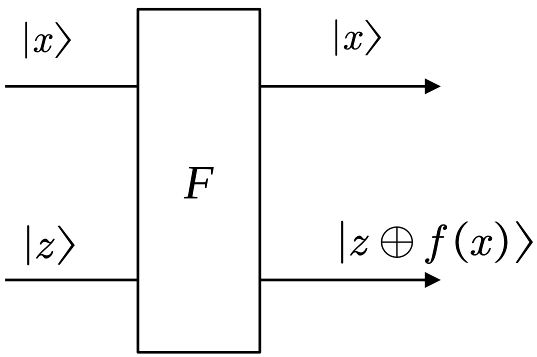
- 为什么从 f 转换到 F 就可逆了?
  - 理由一



来源于：《人人可懂的量子计算》，**Chris Bernhardt著，邱道文等译，机械工业出版社，2020年**

补充材料

- 为什么从 f 转换到 F 就可逆了?
  - 理由二



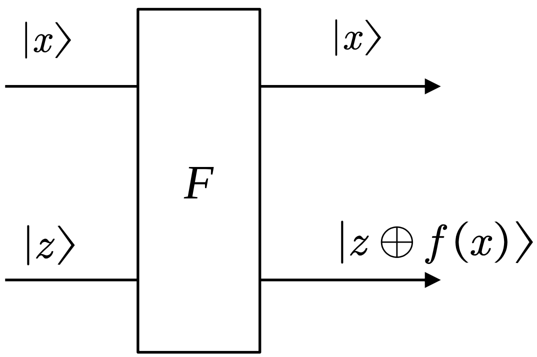
输入	输出	
$\|0\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|f(0)\rangle$	互斥
$\|0\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|f(0) \oplus 1\rangle$	
$\|1\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|f(1)\rangle$	互斥
$\|1\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|f(1) \oplus 1\rangle$	

- 对于任意 f，$f(0)$ 与 $f(0) \oplus 1$ 一个为0，一个为1
- 对于任意 f，$f(1)$ 与 $f(1) \oplus 1$ 一个为0，一个为1

来源于：《人人可懂的量子计算》，Chris Bernhardt著，邱道文等译，机械工业出版社，2020年

补充材料

- 为什么从 f 转换到 F 就可逆了?
  - 理由二



输入	输出
$\|0\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|f(0)\rangle$
$\|0\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|f(0) \oplus 1\rangle$
$\|1\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|f(1)\rangle$
$\|1\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|f(1) \oplus 1\rangle$

  - 输入与输出均为标准基， 是一个标准基变换矩阵
  - 表明门是正交的（酉矩阵），量子系统物理可实现

来源于：《人人可懂的量子计算》，Chris Bernhardt著，邱道文等译，机械工业出版社，2020年

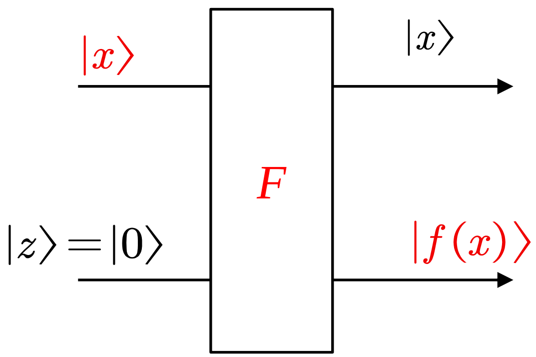
补充材料

- 证明：标准正交基变换矩阵 $\mathbf{M}$ 是正交的
  - 给定两个标准正交基 $\mathbf{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n]$ 和 $\mathbf{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n]$
  - 假设基变换矩阵 $\mathbf{M}:\mathbf{U}\rightarrow\mathbf{V}$，即 $\mathbf{V} = \mathbf{UM}$
    - 有 $\mathbf{I}=\mathbf{V}^{\dagger}\mathbf{V}=(\mathbf{UM})^{\dagger}(\mathbf{UM})$

      $=\mathbf{M}^{\dagger}\mathbf{U}^{\dagger}\mathbf{UM}=\mathbf{M}^{\dagger}\mathbf{IM}=\mathbf{M}^{\dagger}\mathbf{M}$
    - 即 标准基变换矩阵是正交的（即为酉矩阵）
  - 假设一个向量 $\boldsymbol{s}$ 在 $\mathbf{U}$ 和 $\mathbf{V}$ 下的坐标分别为 $\boldsymbol{x}$ 和 $\boldsymbol{y}$
    - 有 $\boldsymbol{s}=\mathbf{U}\boldsymbol{x}=\mathbf{V}\boldsymbol{y}=\mathbf{UM}\boldsymbol{y}\rightarrow\boldsymbol{x}=\mathbf{M}\boldsymbol{y}$

补充材料

- 从不可逆函数到可逆门（仅看右表一、三行）



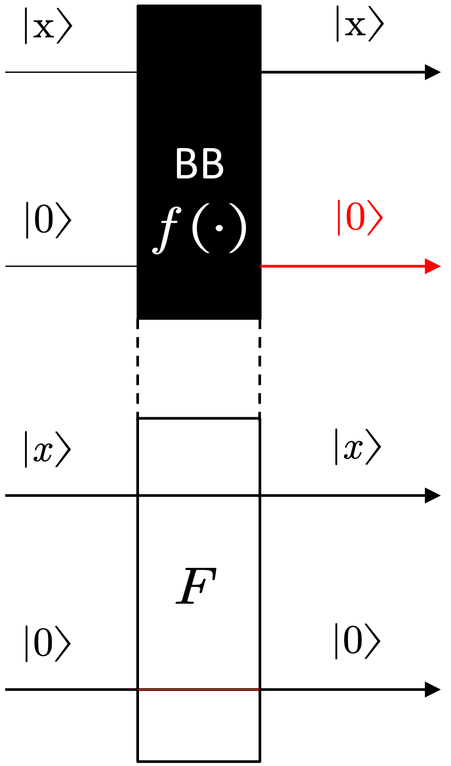
输入	输出
$\|0\rangle \otimes \|0\rangle$	$\|0\rangle \otimes \|f(0)\rangle$
$\|0\rangle \otimes \|1\rangle$	$\|0\rangle \otimes \|f(0) \oplus 1\rangle$
$\|1\rangle \otimes \|0\rangle$	$\|1\rangle \otimes \|f(1)\rangle$
$\|1\rangle \otimes \|1\rangle$	$\|1\rangle \otimes \|f(1) \oplus 1\rangle$

- 当 $|z\rangle = |0\rangle$ 时， $F(|x\rangle, |0\rangle) = (|x\rangle, |f(x)\rangle)$
- 不可逆函数 $f(x)$ 转变为 可逆函门 F ，两者一一对应

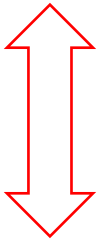
来源于：《人人可懂的量子计算》，Chris Bernhardt著，邱道文等译，机械工业出版社，2020年

1. Deutsch's algorithm

- Reversible gate for constant-0 function



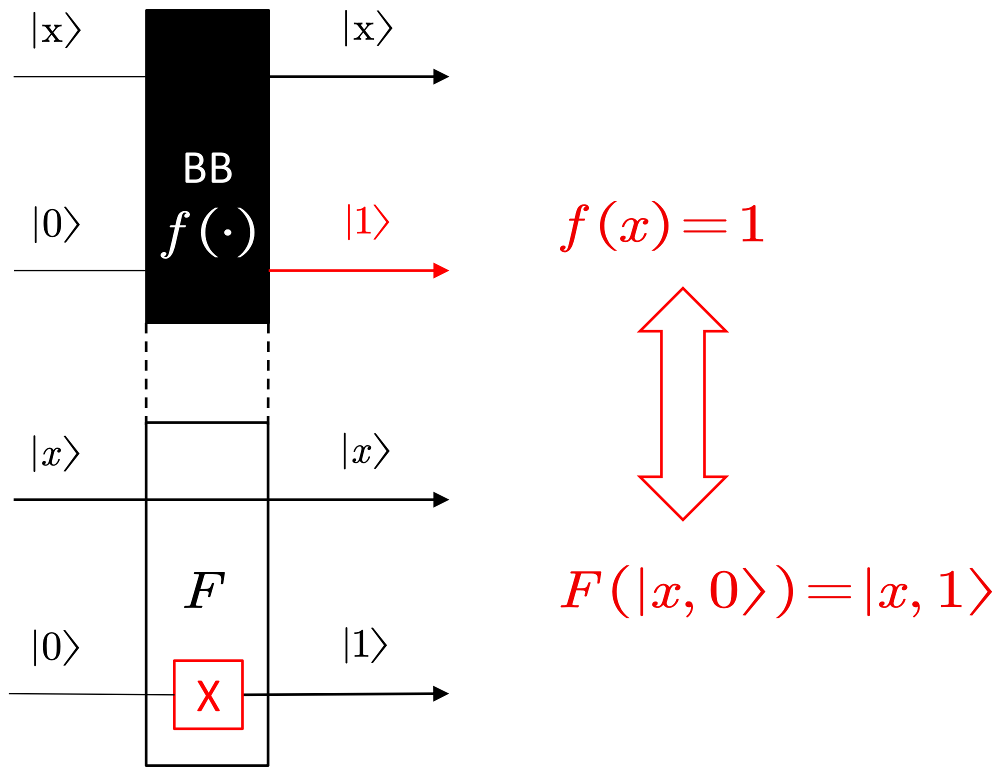
$f(x) = 0$



$F(|x, 0\rangle) = |x, 0\rangle$

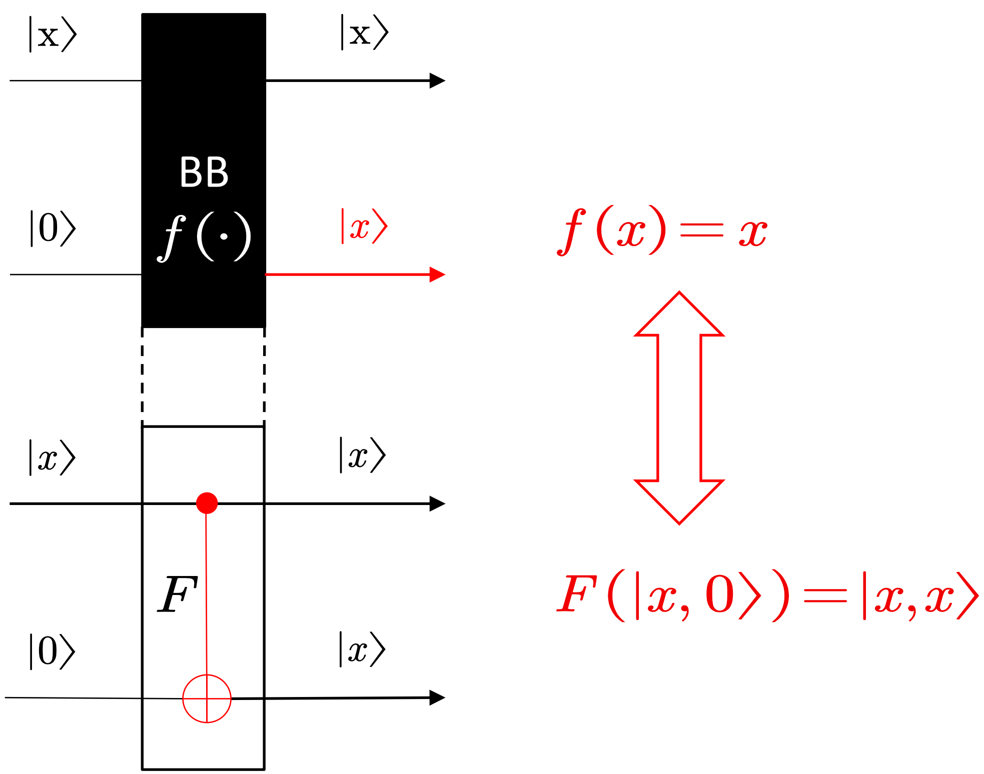
1. Deutsch's algorithm

- Reversible gate for constant-1 function



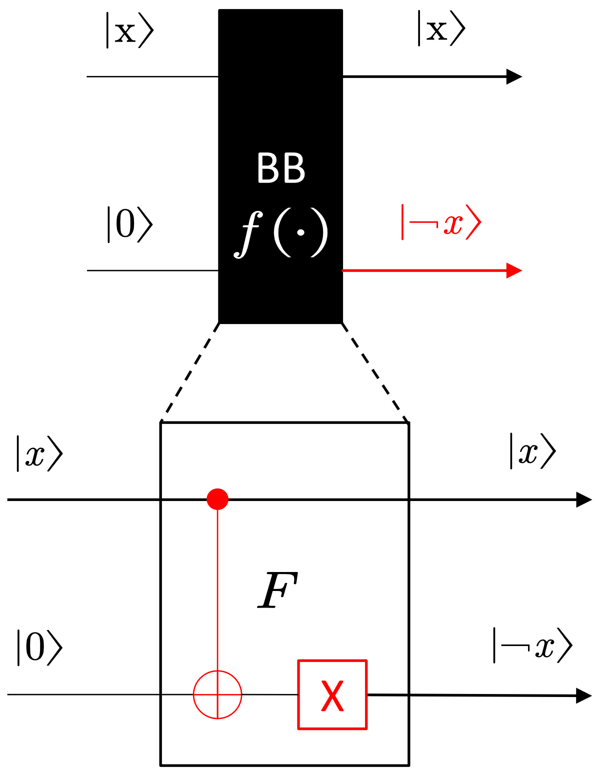
1. Deutsch's algorithm

- Reversible gate for identity function



1. Deutsch's algorithm

- Reversible gate for negation function



$f(x) = \neg x$

$\Updownarrow$

$F(|x, 0\rangle) = |x, \neg x\rangle$

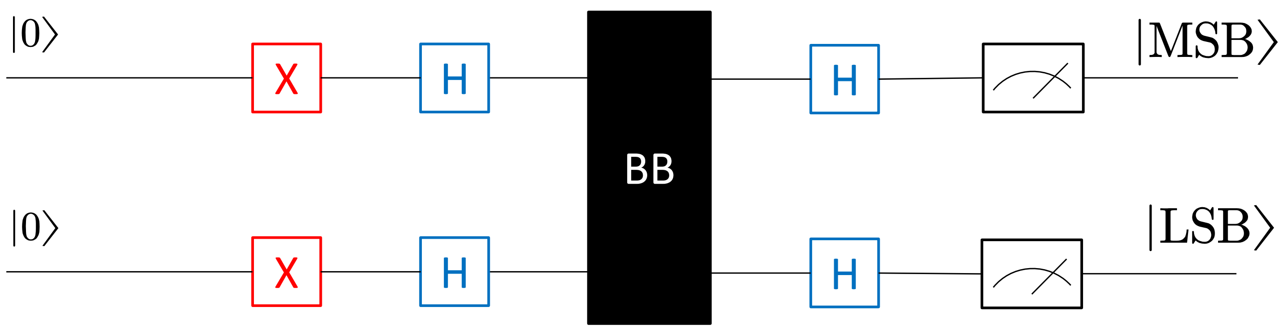
补充材料

- 多伊奇先知问题的量子计算版本
  - 给定这四个门中任意一个 F_i，要用这个门多少次才能确定对应的函数 f_i 是常值函数还是平衡函数?
  - 如果限制输入为经典比特 $|0\rangle$或 $|1\rangle$，则必须使用这个门两次
  - 如果允许输入包含 $|0\rangle$和 $|1\rangle$ 的叠加态，则只需要使用这个门一次

来源于：《人人可懂的量子计算》，**Chris Bernhardt**著，邱道文等译，机械工业出版社，**2020**年

1. Deutsch's algorithm

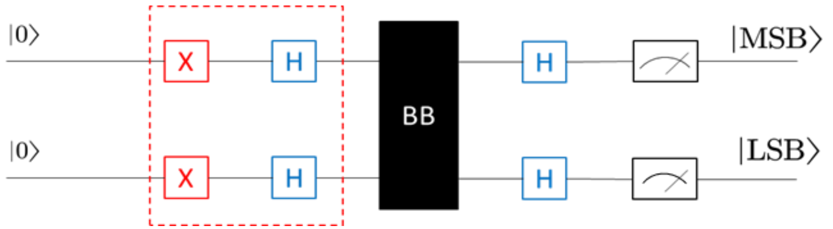
- Deutsch's algorithm



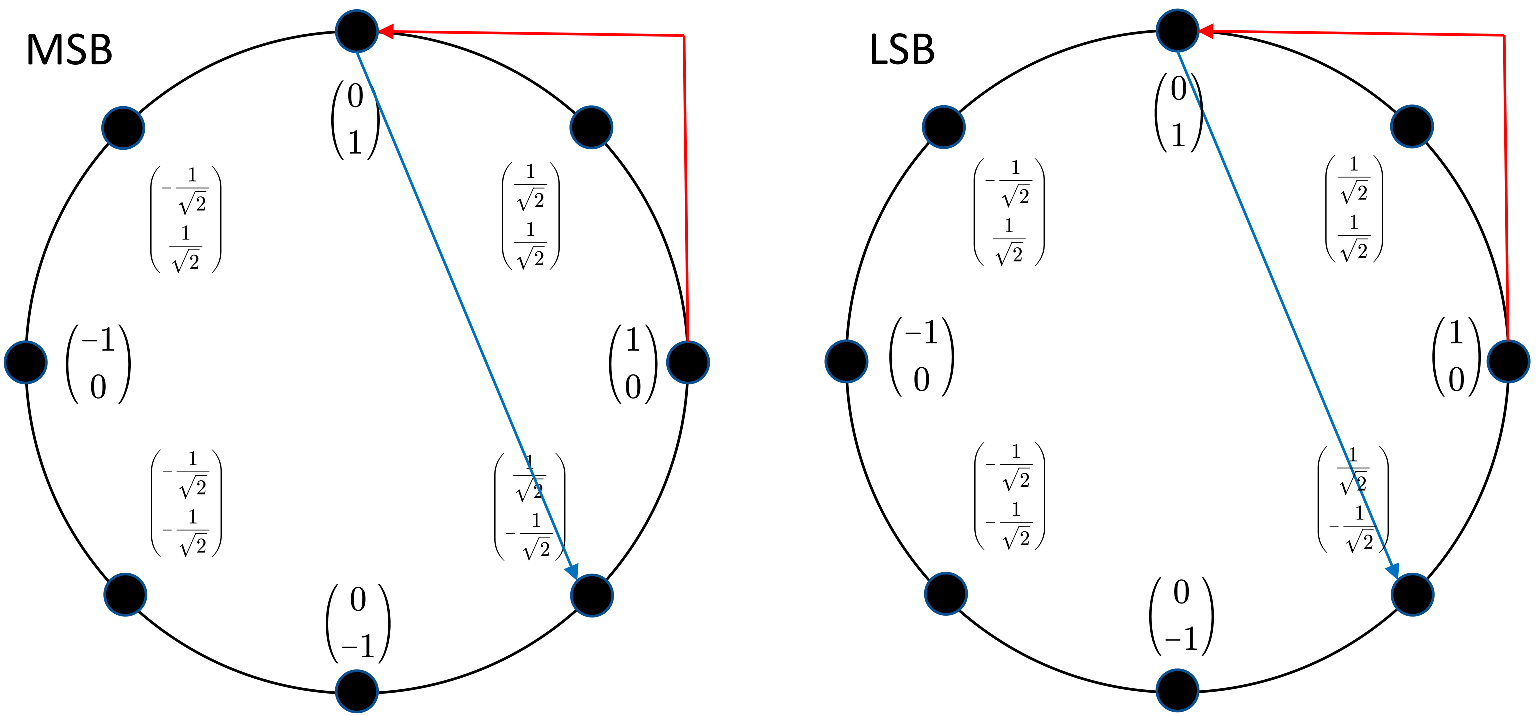
- If the BB function is constant, measurement result would be $|\mathrm{MSB},\ \mathrm{LSB}\rangle = |11\rangle$
- If the BB function is balanced, measurement result would be $|\mathrm{MSB},\ \mathrm{LSB}\rangle = |01\rangle$

（感谢弘毅学堂2020级李宇尧同学纠正LSB支路观测符号拼写错误）

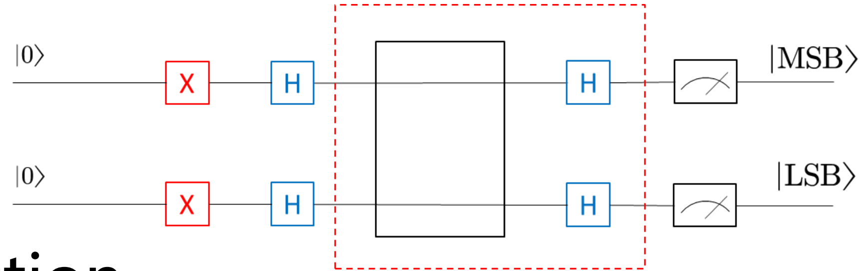
1. Deutsch's algorithm



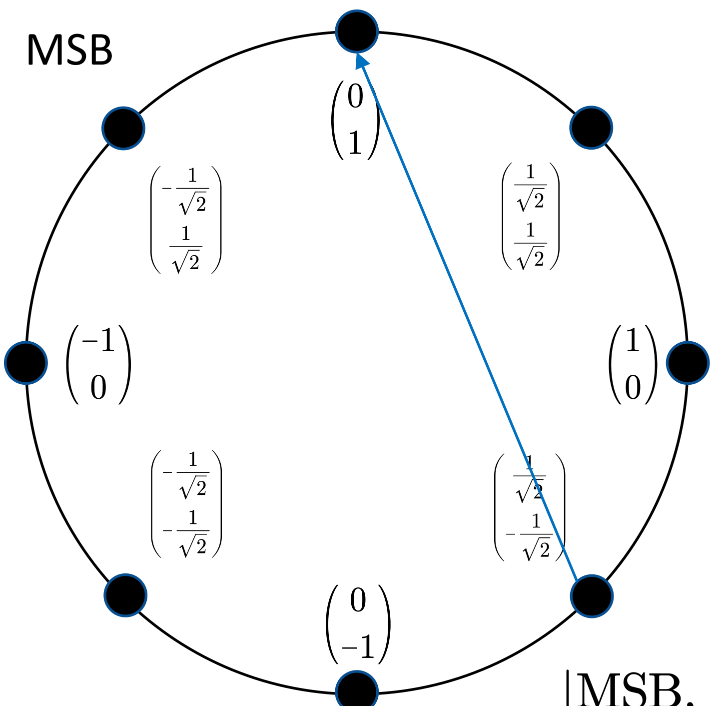
- Deutsch's algorithm
  - preprocessing

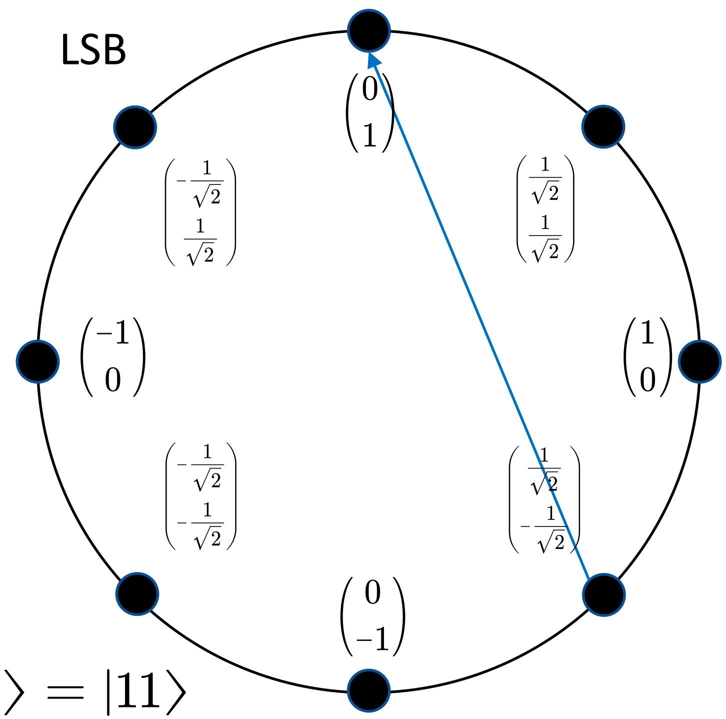


1. Deutsch's algorithm



- Deutsch's algorithm
  - BB is constant-0 function

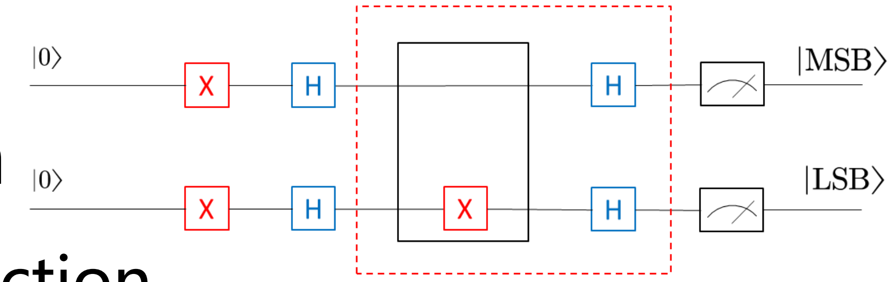




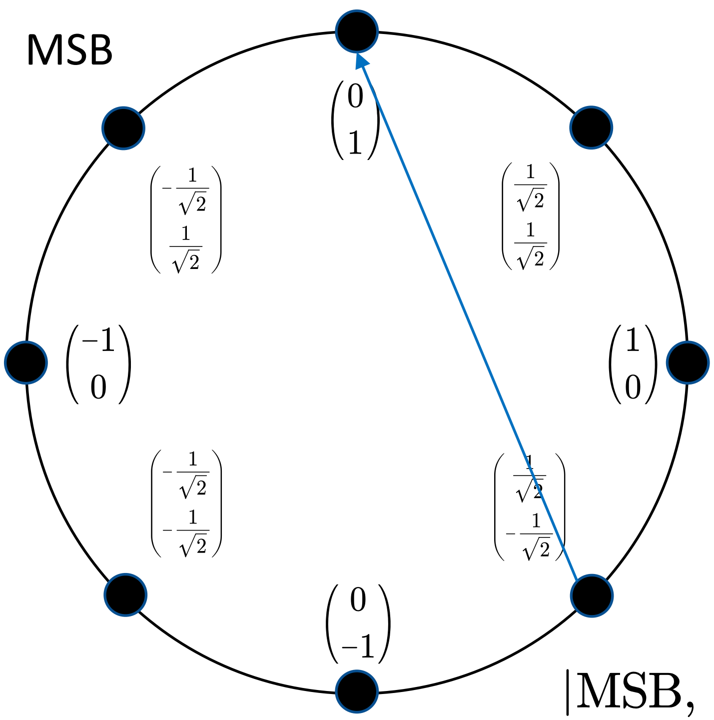
$$|\mathrm{MSB},\ \mathrm{LSB}\rangle = |11\rangle$$

1. Deutsch's algorithm

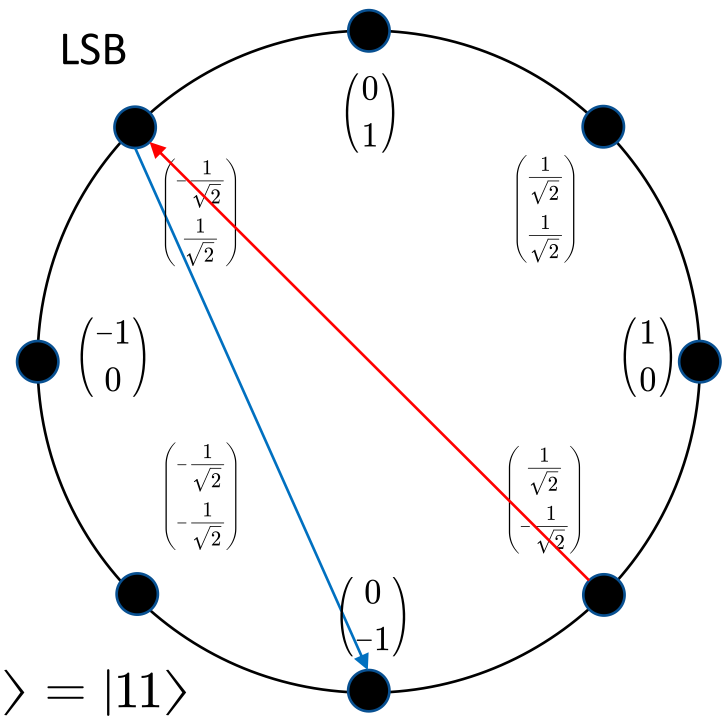
- Deutsch's algorithm
  - BB is constant-1 function



MSB



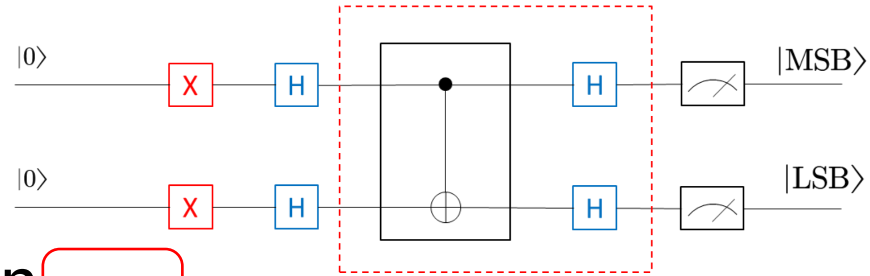
LSB

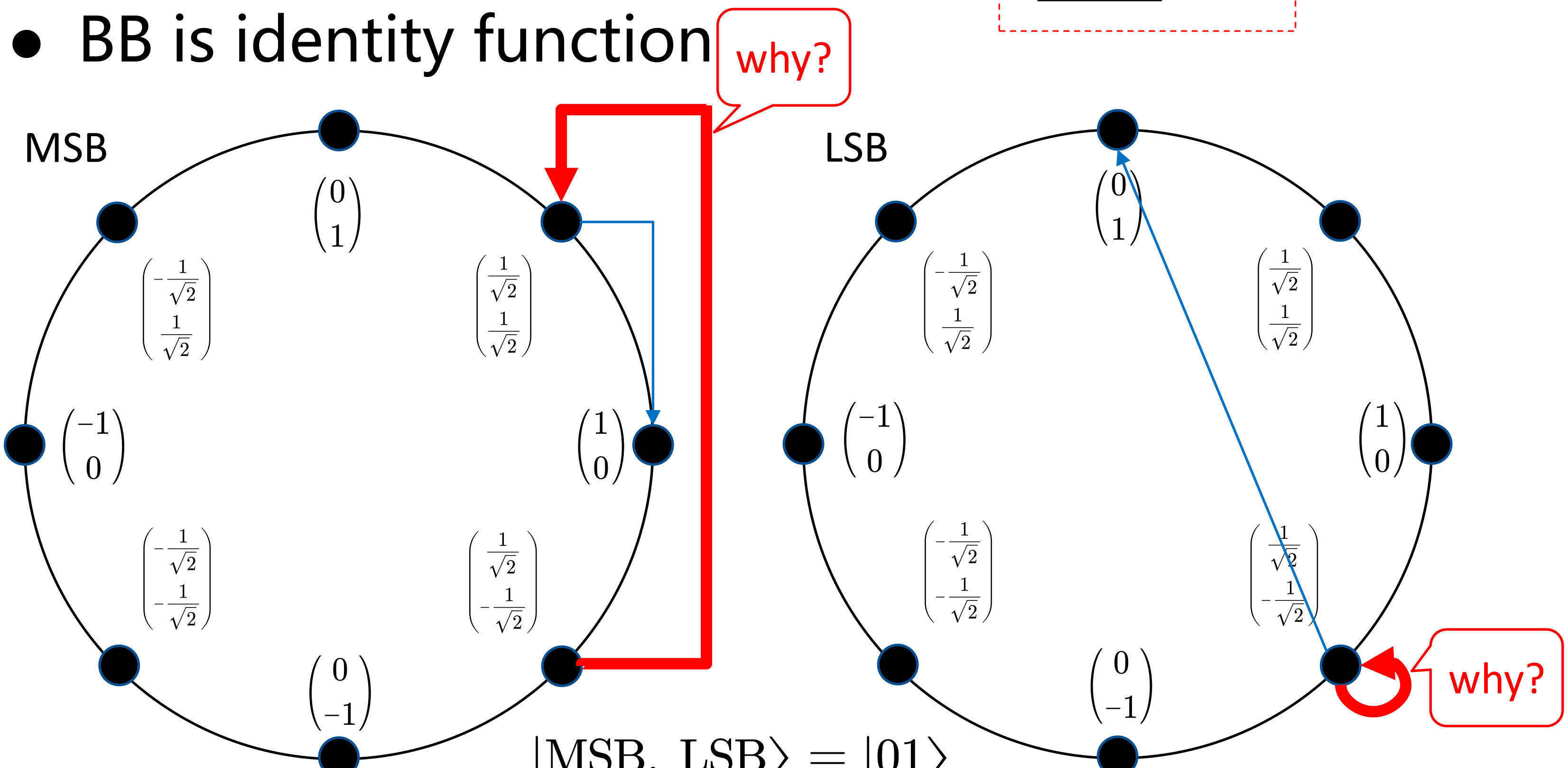


$$|\mathrm{MSB},\ \mathrm{LSB}\rangle = |11\rangle$$

1. Deutsch's algorithm

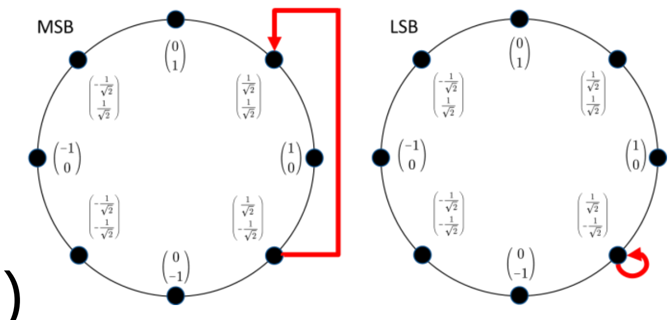
- Deutsch's algorithm
  - BB is identity function





$$|\text{MSB, LSB}\rangle = |01\rangle$$

1. Deutsch's algorithm

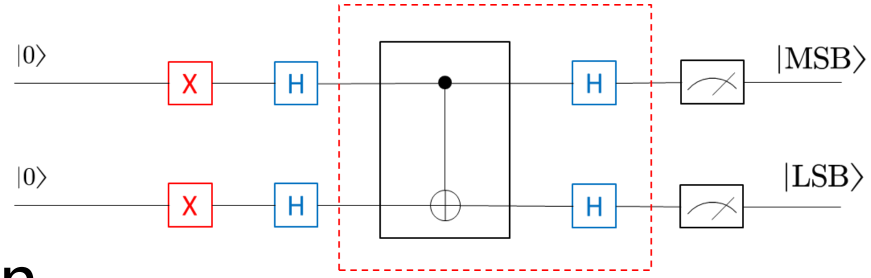


- Deutsch's algorithm
  - BB is identity function (cont.)

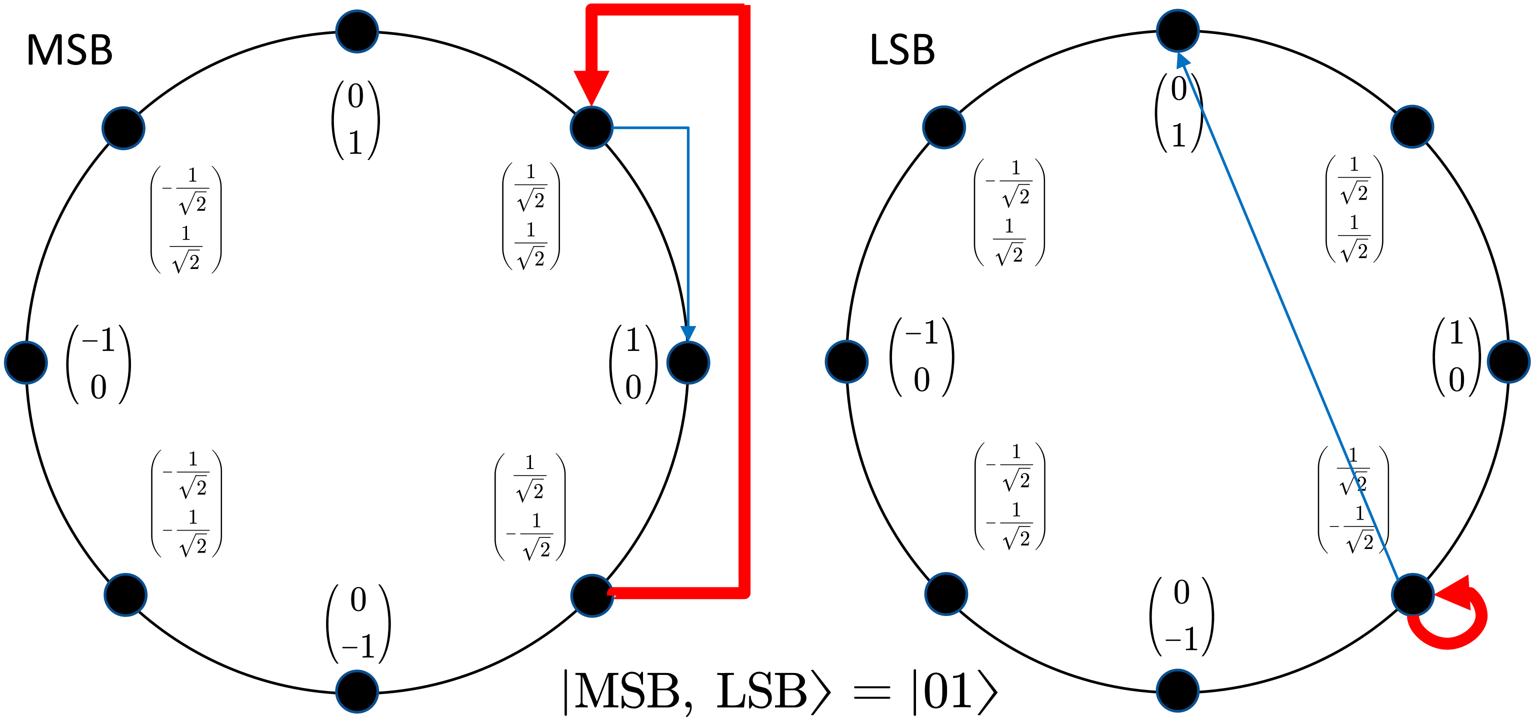
$$\mathrm{CNOT}\left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}\right)=\mathrm{CNOT}\begin{pmatrix}\frac{1}{2}\\ -\frac{1}{2}\\ -\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\-1\\-1\\1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}=\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}$$

$|\mathrm{MSB}\rangle$ $|\mathrm{LSB}\rangle$

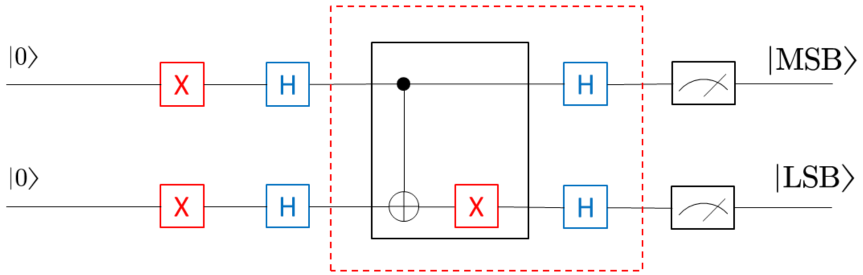
1. Deutsch's algorithm



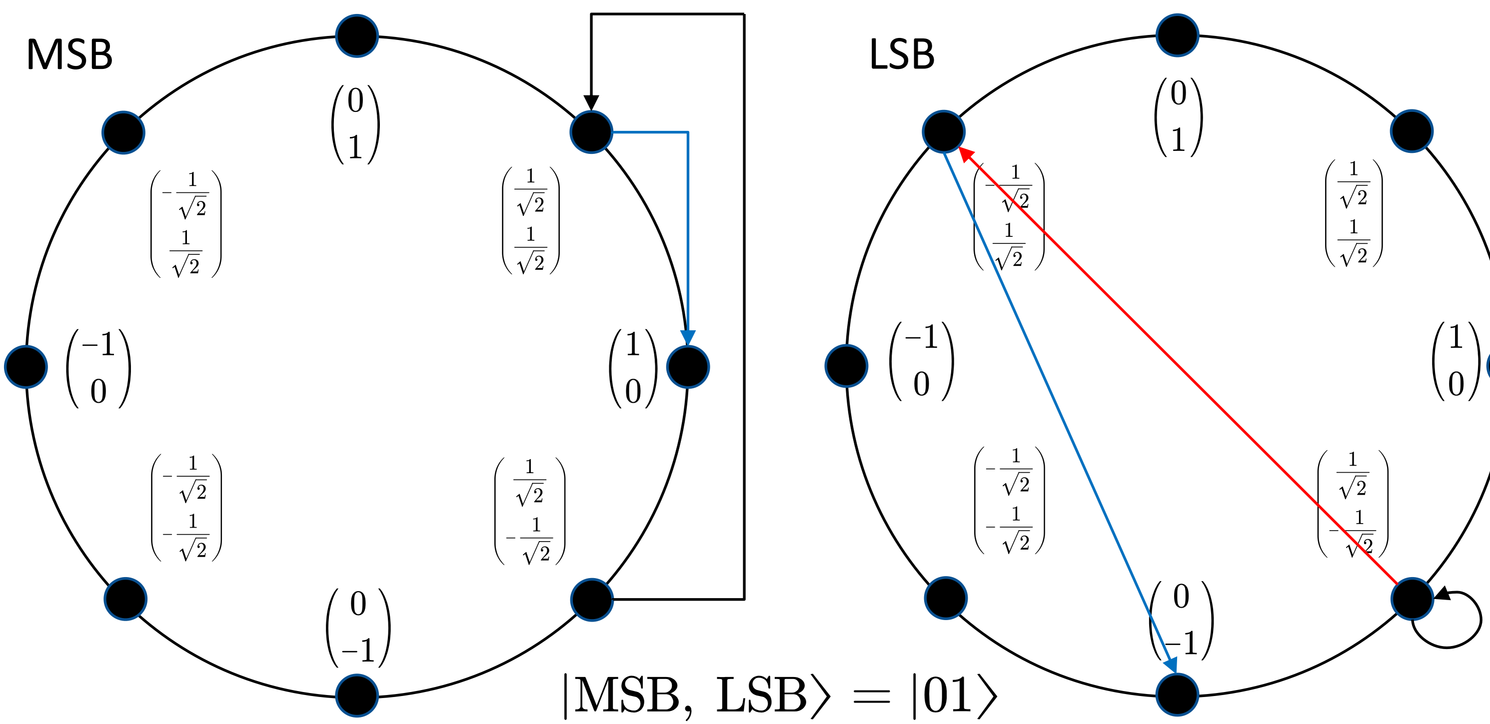
- Deutsch's algorithm
  - BB is identity function



1. Deutsch's algorithm



- Deutsch's algorithm
  - BB is negation function



$$|\mathrm{MSB},\ \mathrm{LSB}\rangle = |01\rangle$$

1. Deutsch's algorithm

- Discussion
  - We did it! But why?
    - Problem 1: Why it is so efficient?
    - Problem 2: Why it is effective?

1. Deutsch's algorithm

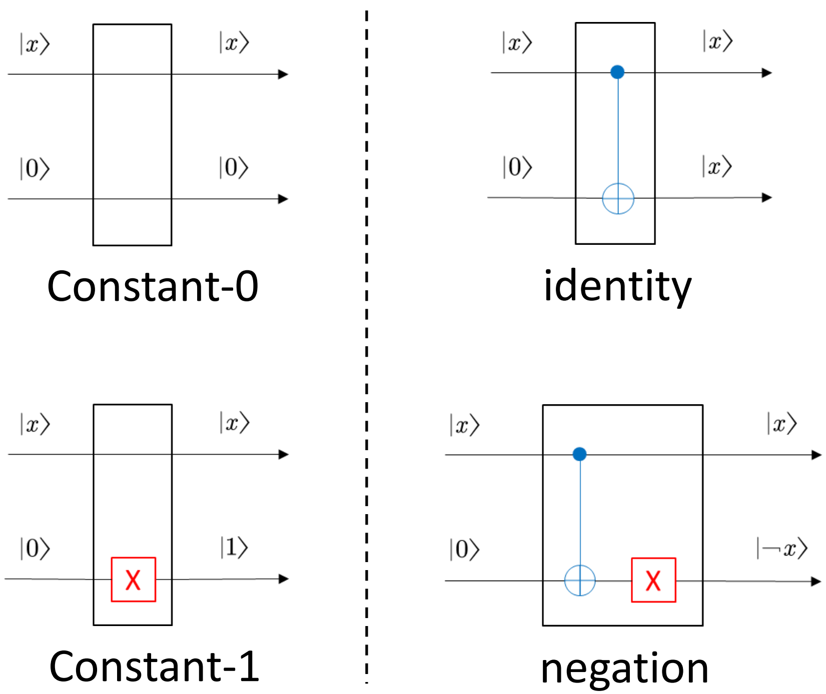
- Discussion
  - Problem 1: Why it is so efficient?
    - Palm civet for prince（狸猫换太子）
      - Irreversible functions -> reversible gates
      - Classic bits -> qubits
    - Qubits
      - Superposition
      - Parallel computation

1. Deutsch's algorithm

- Discussion
  - Problem 2: Why it is effective?
    - The difference within categories (negation) was neutralized
    - The difference between categories (CNOT) was magnified

1. Deutsch's algorithm

- Discussion
  - We did it! But why?

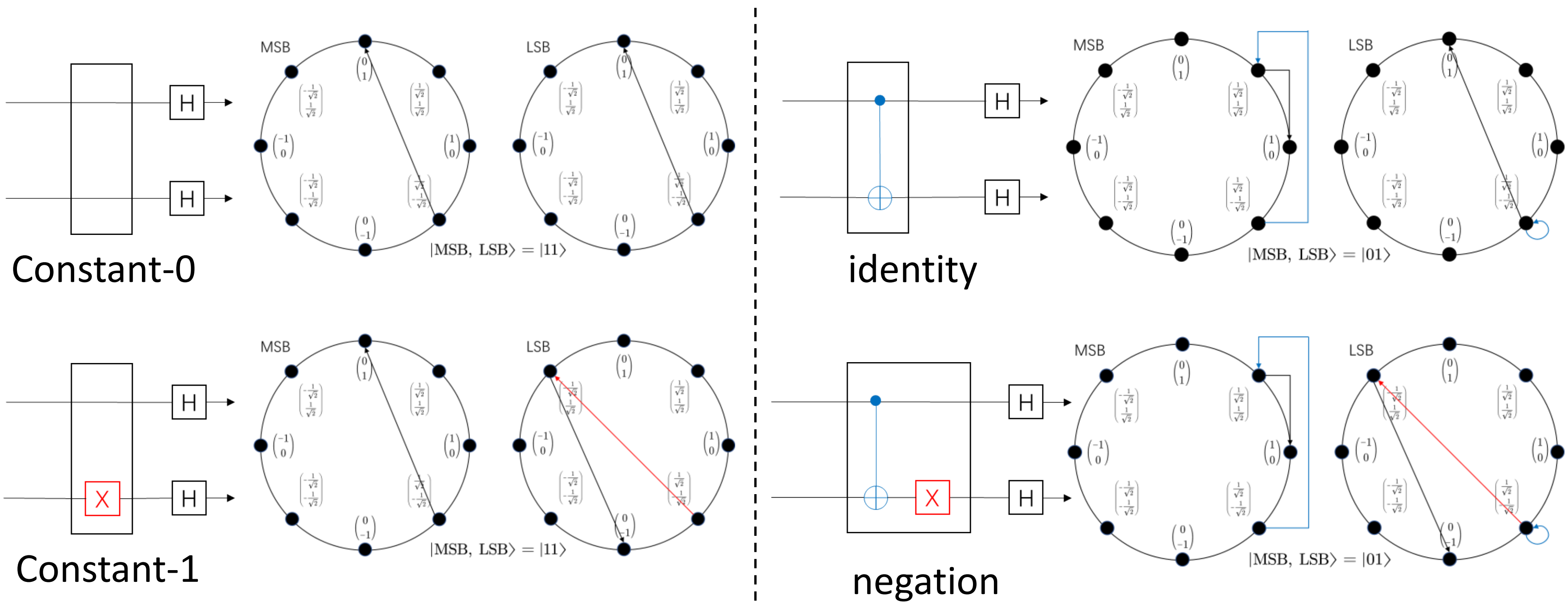


1. Difference within categories is negation
2. Difference between categories is CNOT

1. Deutsch's algorithm

- Discussion
  - We did it! But why?

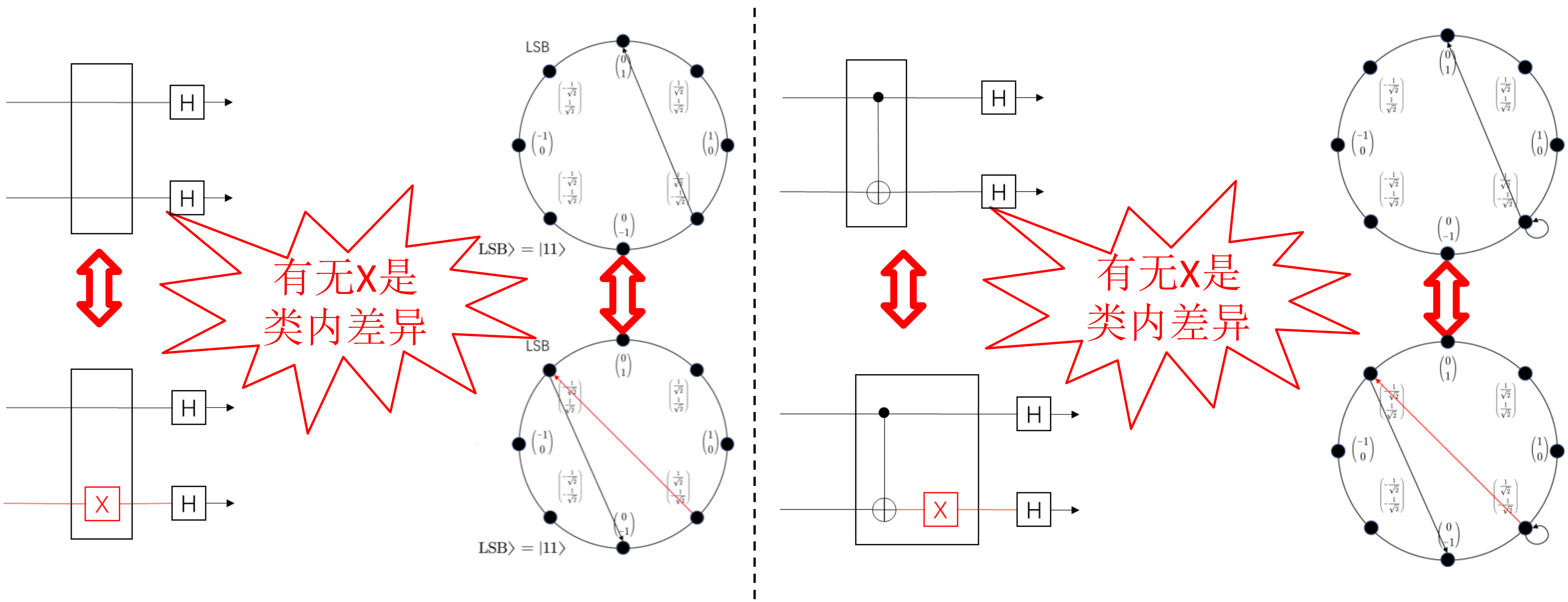
1. Difference within categories, negation, is neutralized
2. Difference between categories, CNOT, is magnified



1. Deutsch's algorithm

- Discussion
  - We did it! But why?

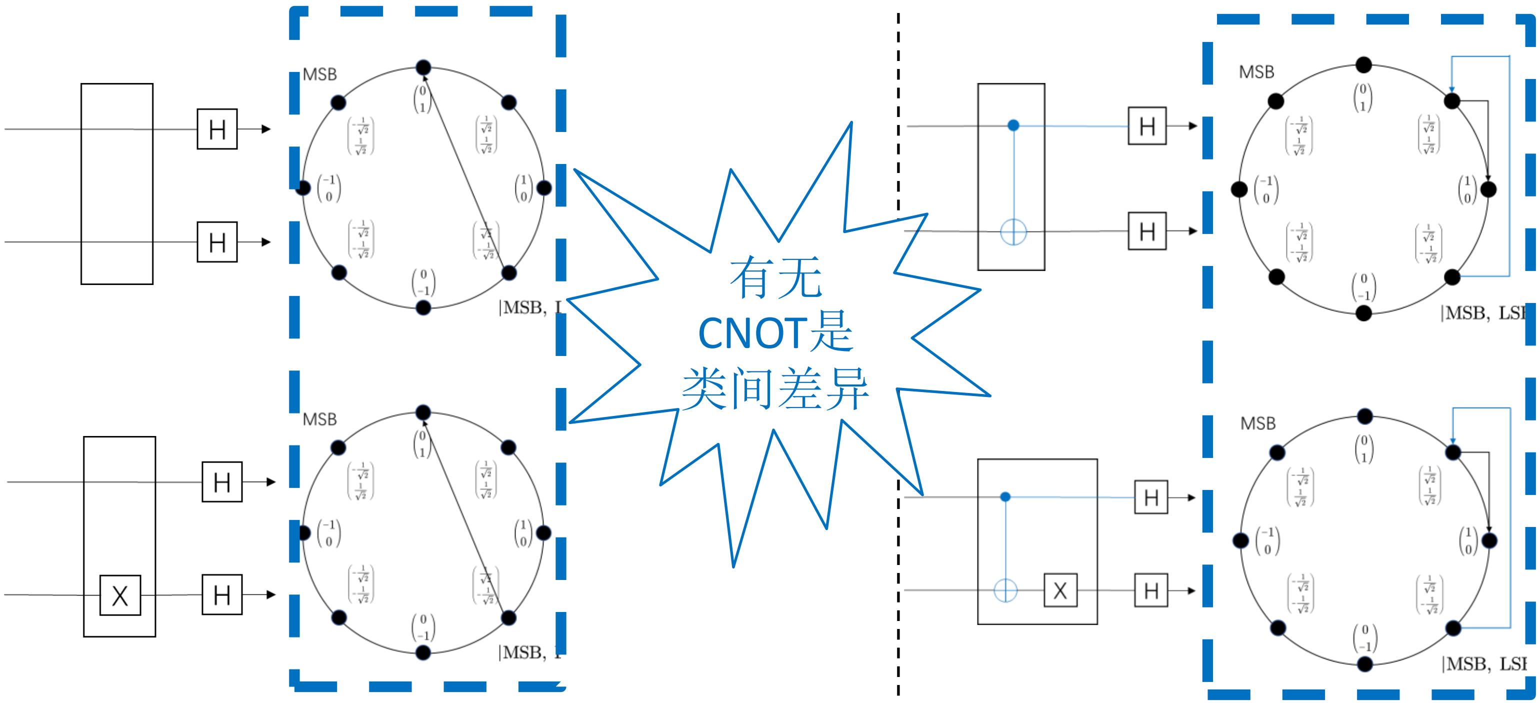
1. Difference within categories, negation, is neutralized
2. Difference between categories, CNOT, is magnified



1. Deutsch's algorithm

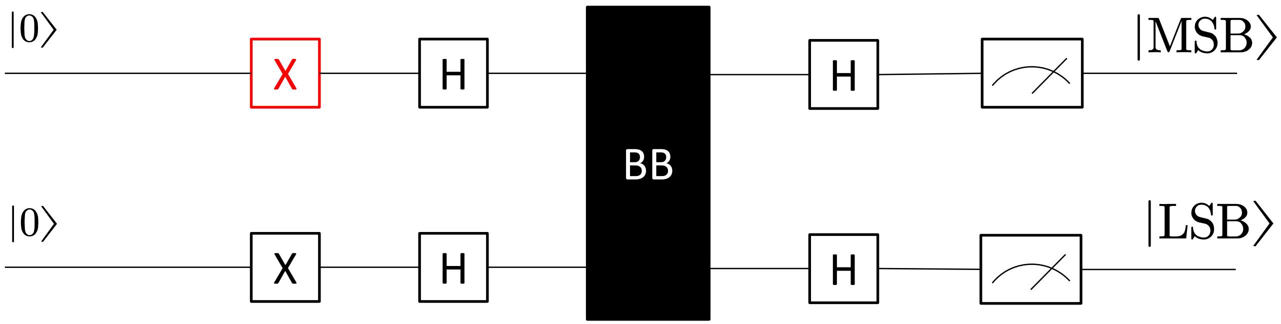
- Discussion
  - We did it! But why?

1. Difference within categories, negation, is neutralized
2. Difference between categories, CNOT, is magnified



补充资料：X门的必要性

- 思考题
  - Deutsch算法中第一条支路中的X是必要的吗?



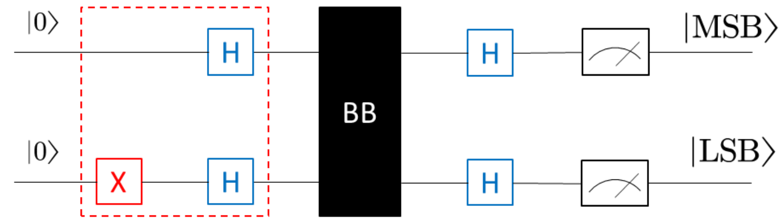
  - 如果不是必要的，则算法结果有何更改?

（感谢弘毅2018级王浩冰同学指出第一条支路X门的必要性问题）

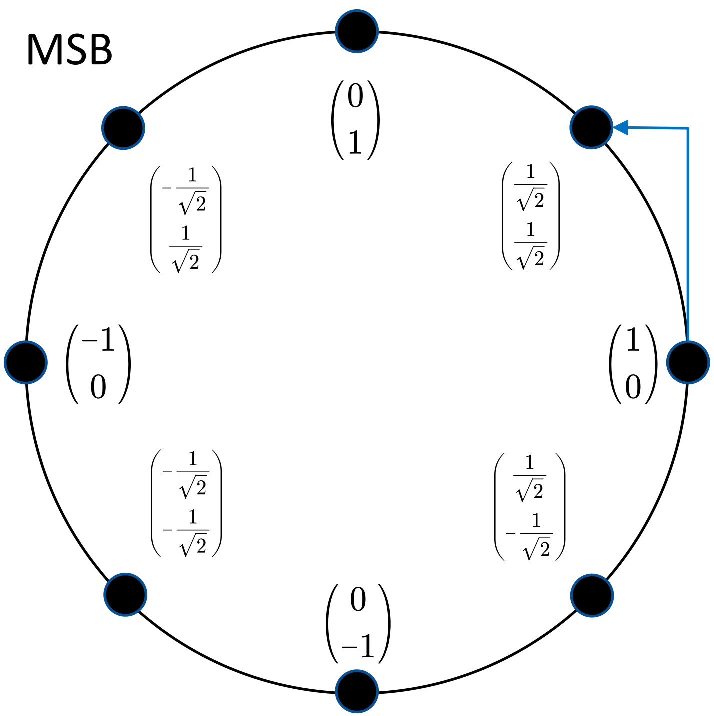
补充资料： X门的必要性

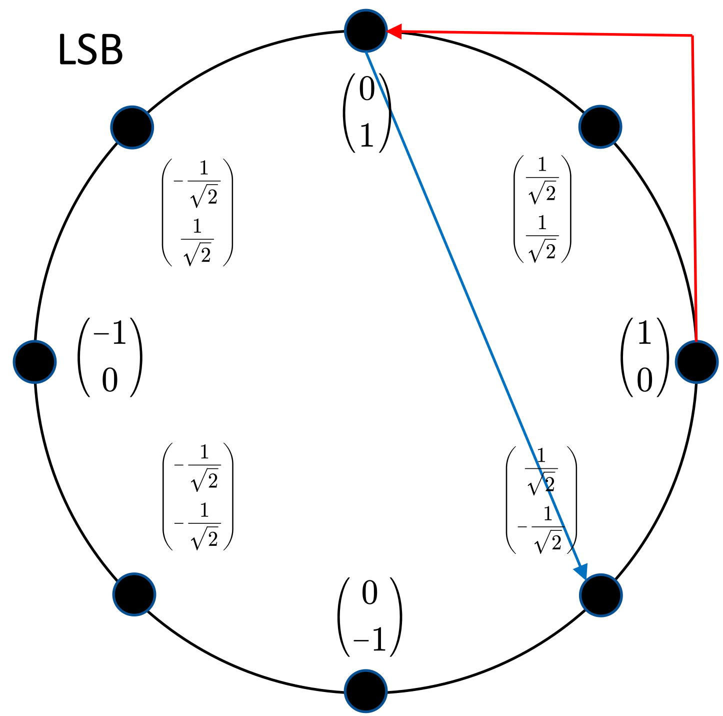
- 思考题答案
  - Deutsch算法中第一条支路中的X是必要的吗?
    - 不是必要的。依然可以对两类函数进行区分，但结果相反。
  - 如果不是必要的，则算法结果有何更改?
    - If the BB function is constant, measurement result would be $|\mathrm{MSB},\ \mathrm{LSB}\rangle = |01\rangle$
    - If the BB function is balanced, measurement result would be $|\mathrm{MSB},\ \mathrm{LSB}\rangle = |11\rangle$

补充资料： X门的必要性

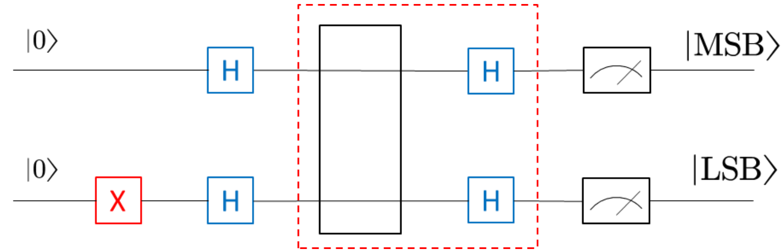


- 原因分析
  - preprocessing

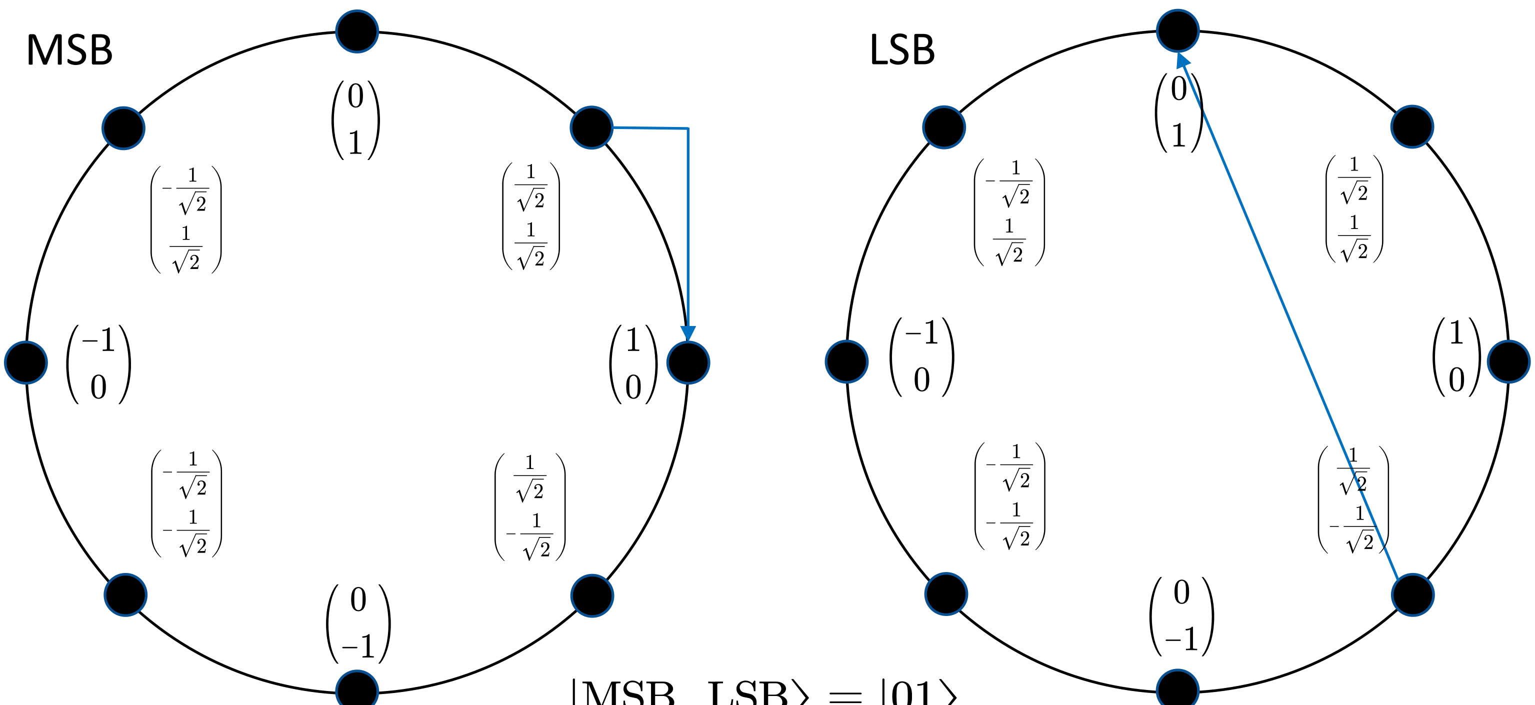




补充资料： X门的必要性

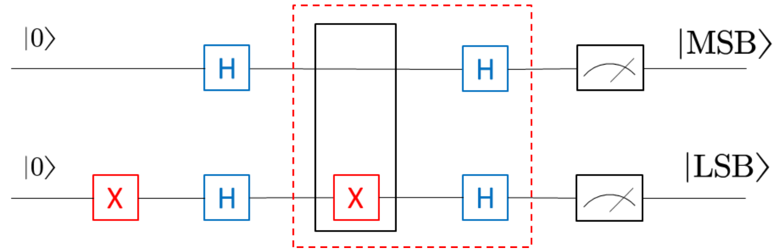


- 原因分析
  - BB is constant-0 function

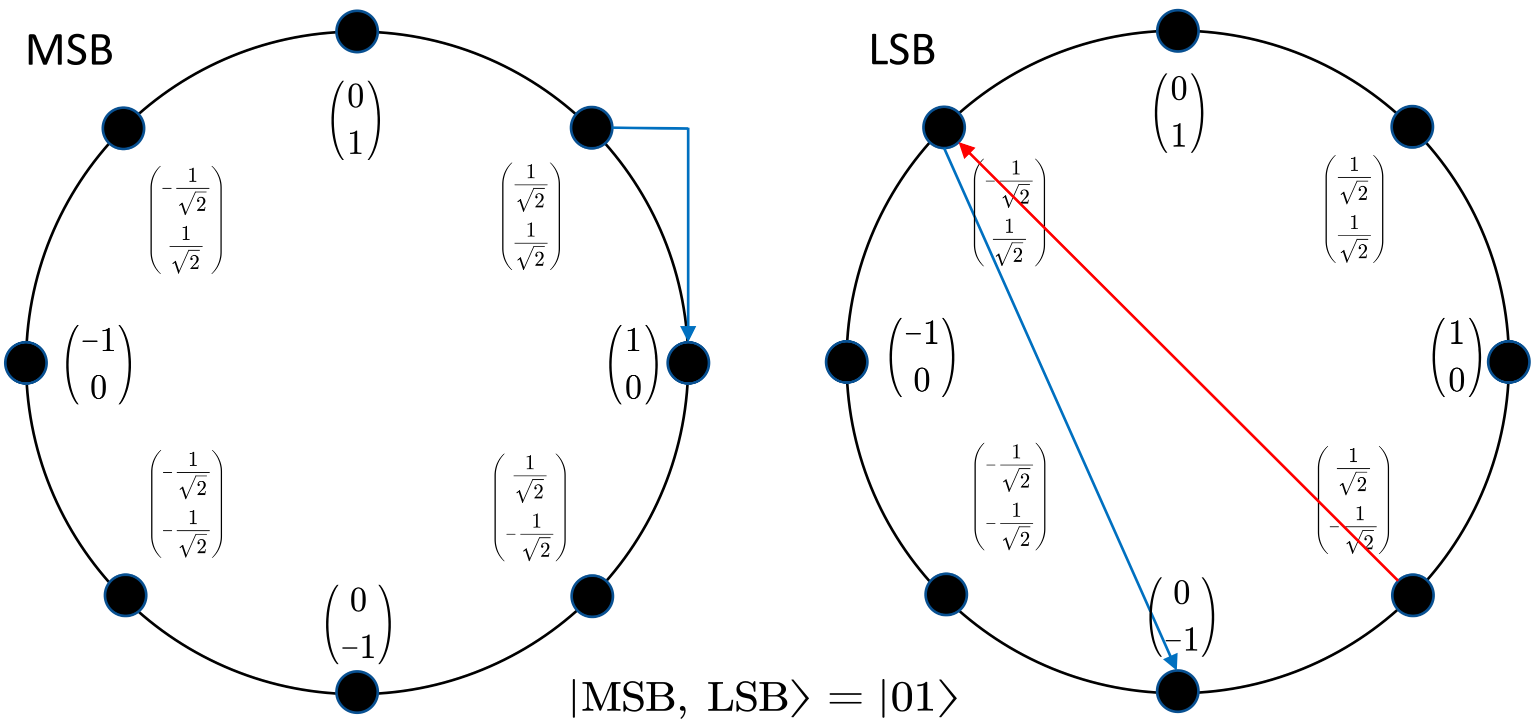


$$|\text{MSB, LSB}\rangle = |01\rangle$$

补充资料： X门的必要性

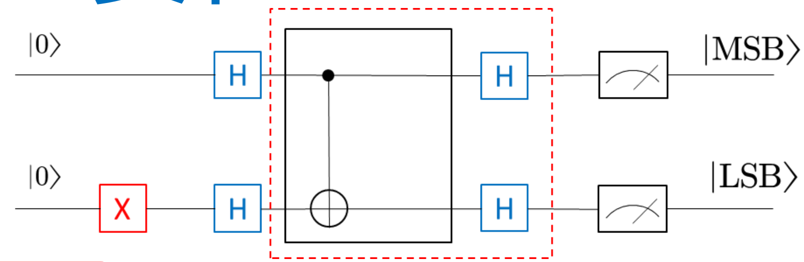


- 原因分析
  - BB is constant-1 function

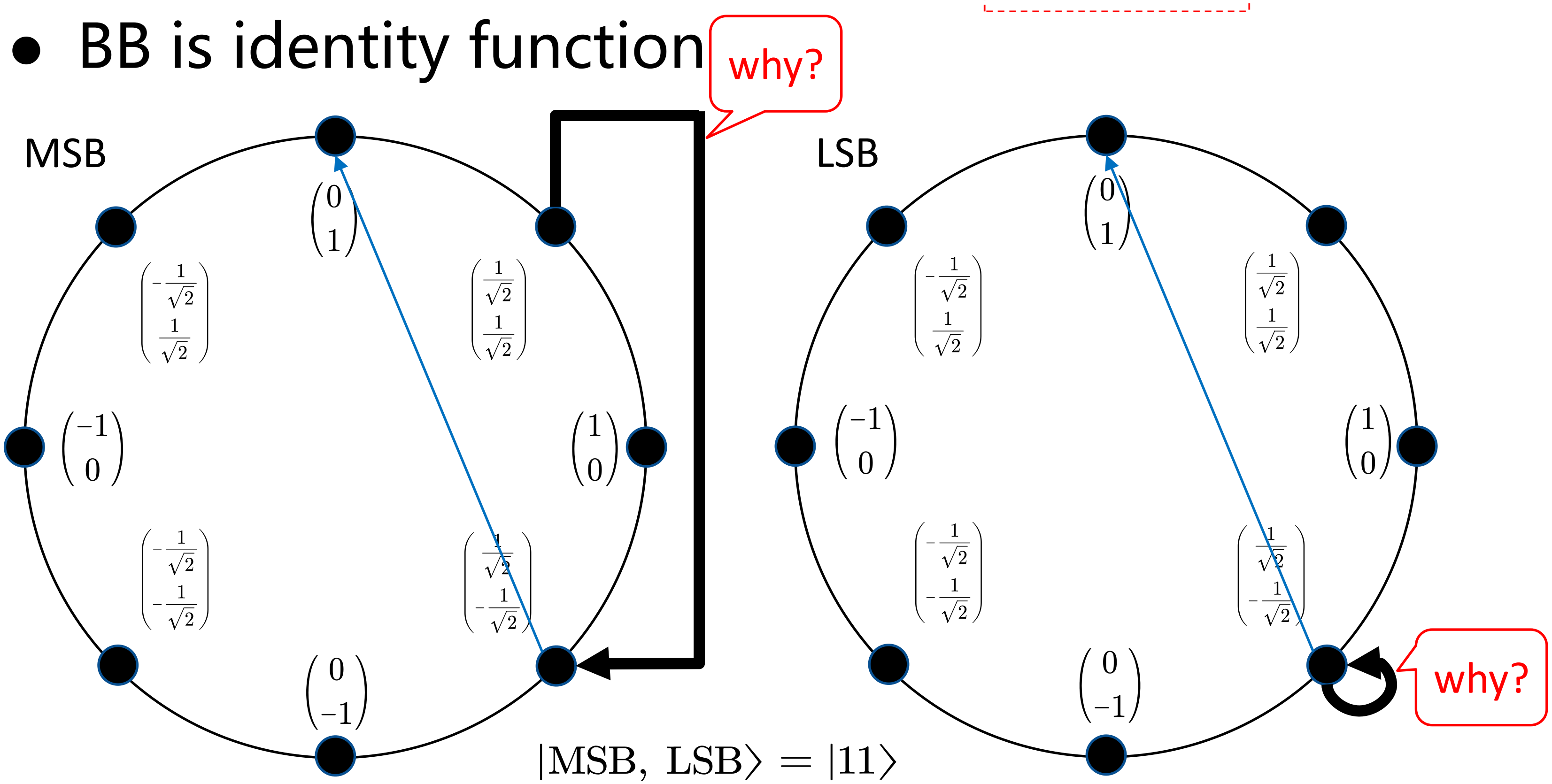


$$|\mathrm{MSB},\ \mathrm{LSB}\rangle = |01\rangle$$

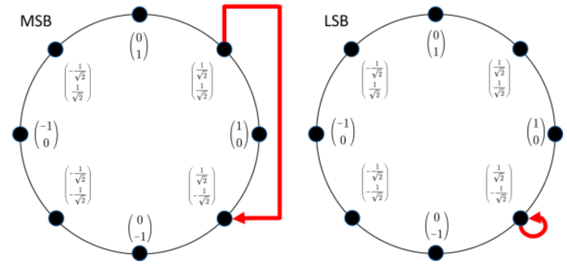
补充资料： X门的必要性



- 原因分析
  - BB is identity function

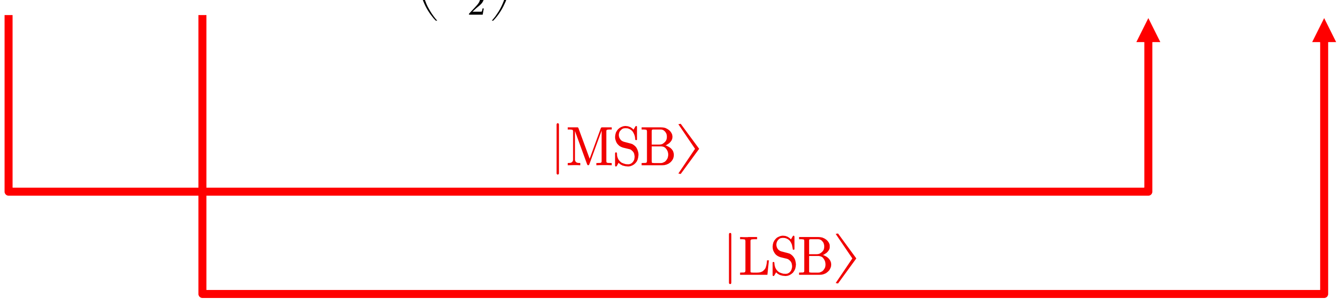


补充资料： X门的必要性

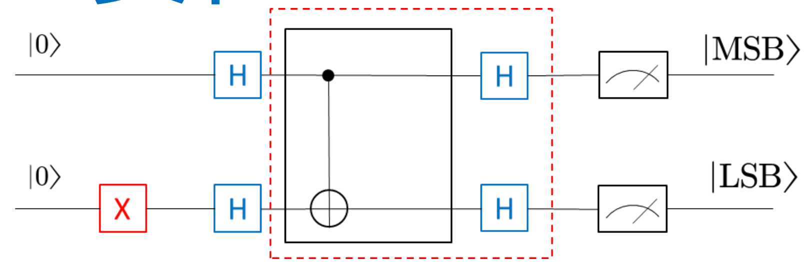


- 原因分析
  - BB is identity function (cont.)

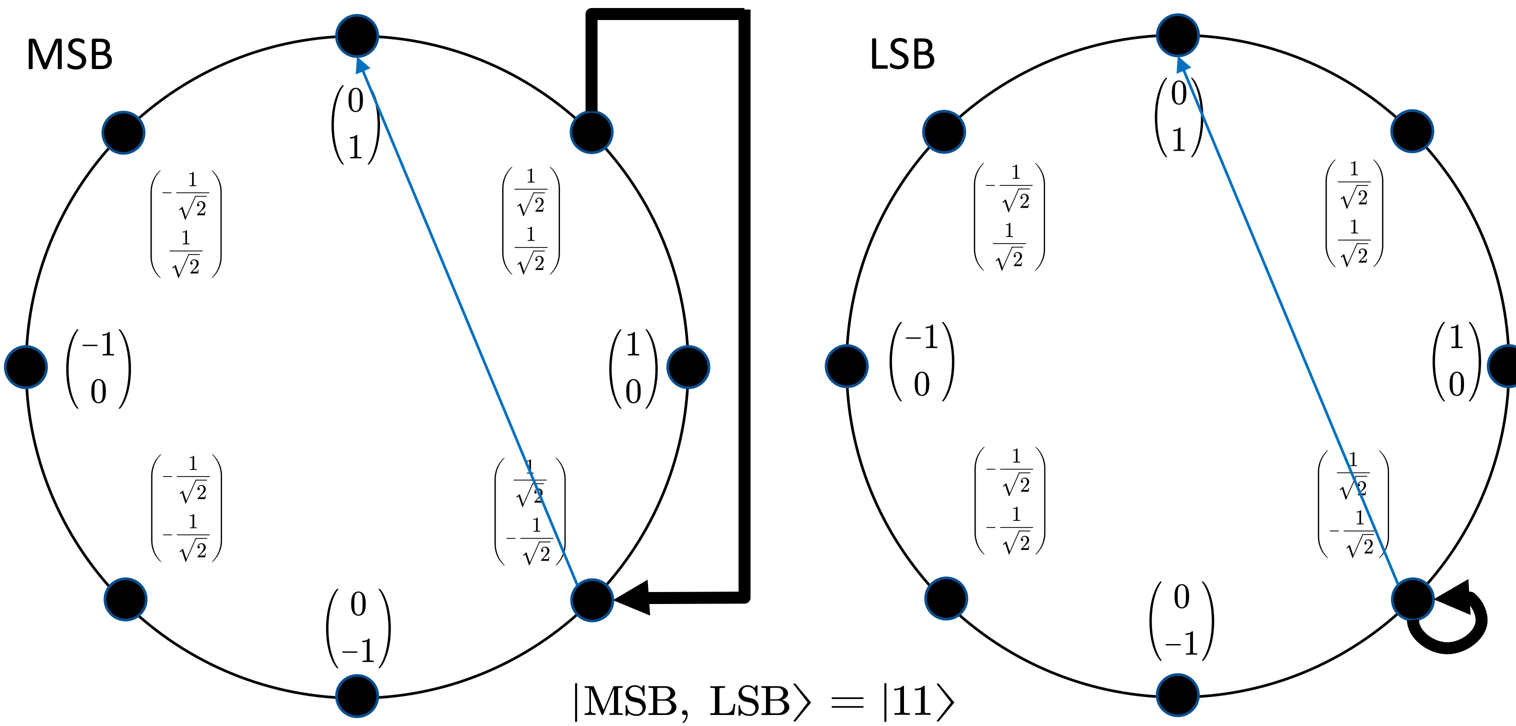
$$\mathrm{CNOT}\left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}\right)=\mathrm{CNOT}\begin{pmatrix}\frac{1}{2}\\ -\frac{1}{2}\\ \frac{1}{2}\\ -\frac{1}{2}\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1\\-1\\-1\\1\end{pmatrix}=\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ -\frac{1}{\sqrt{2}}\end{pmatrix}$$



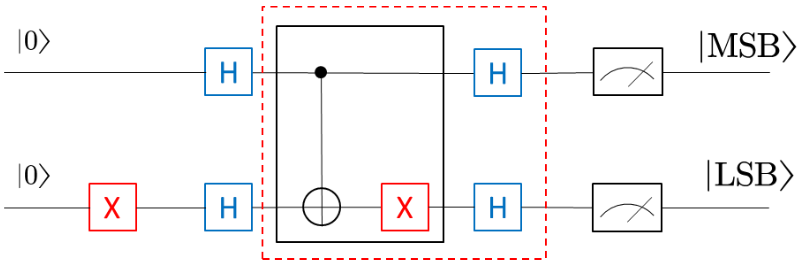
补充资料： X门的必要性



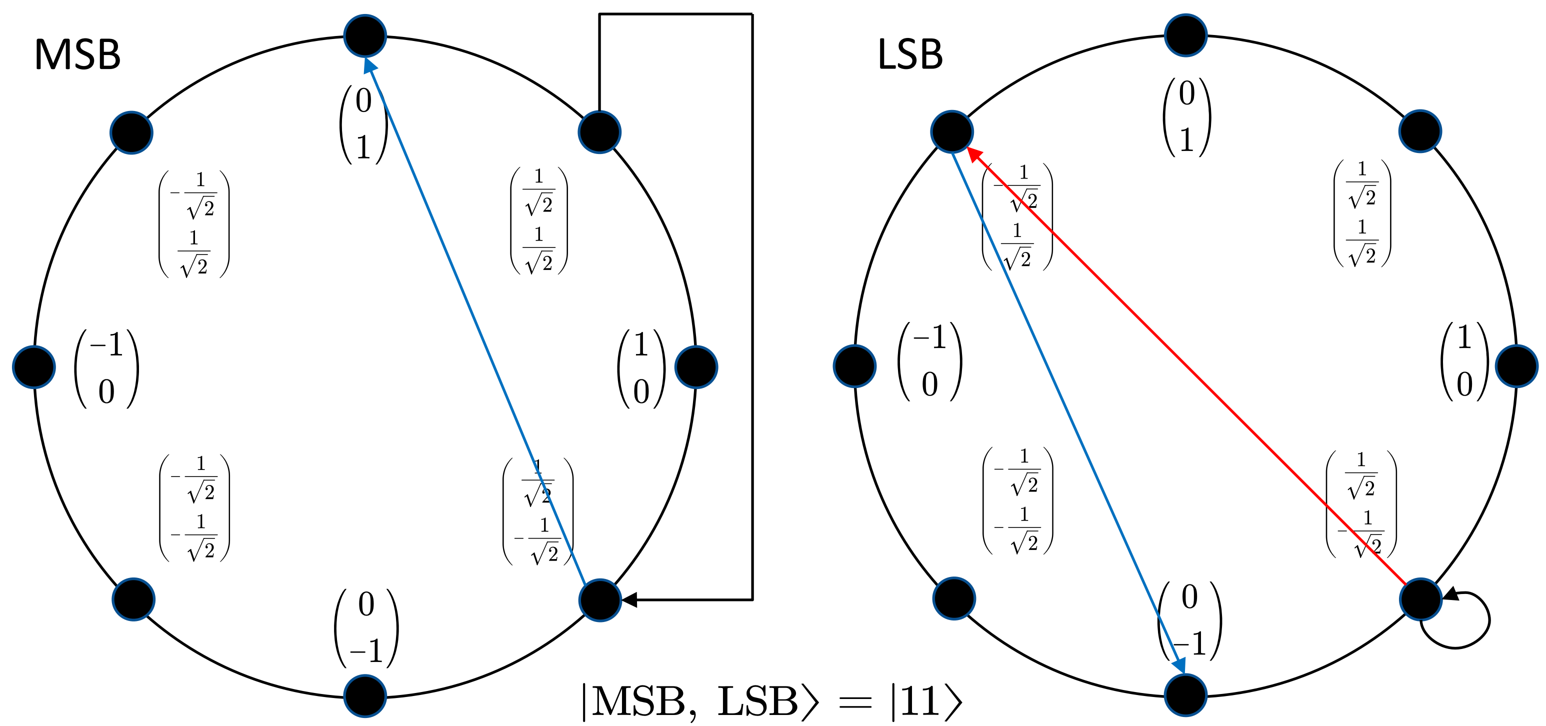
- 原因分析
  - BB is identity function



补充资料： X门的必要性

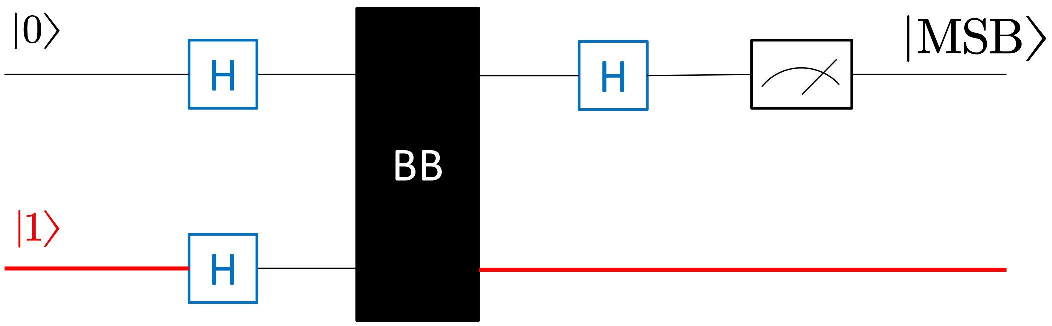


- 原因分析
  - BB is negation function



补充资料：还能进一步简化吗？

- 思考题
  - Deutsch算法电路还能进一步简化吗?
    - 小提示：下面的电路可行吗?



    - 为什么?

来源于：《人人可懂的量子计算》，**Chris Bernhardt著，邱道文等译，机械工业出版社，2020年**

1. Deutsch's algorithm

- Discussion
  - This problem seems pretty contrived
    - A generalized version with n-bit BB is solved by Deutsch-Josza algorithm
    - A variant of the generalized version was an inspiration of Shor's algorithm

2. Deutsch-Jozsa algorithm

- Hadamard matrix

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

  - Operation on single qubit

$$\boldsymbol{H}(|0\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$\boldsymbol{H}(|1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

2. Deutsch-Jozsa algorithm

- Hadamard matrix
  - Operation on double qubits

$|0\rangle \otimes |0\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)=\frac{1}{\sqrt{2}}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$$

$|0\rangle \otimes |1\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle\right)=\frac{1}{\sqrt{2}}(|00\rangle-|01\rangle+|10\rangle-|11\rangle)$$

2. Deutsch-Jozsa algorithm

- Hadamard matrix
  - Operation on double qubits

$|1\rangle \otimes |0\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

$|1\rangle \otimes |1\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

2. Deutsch-Jozsa algorithm

- Hadamard matrix
  - Vector representation of double qubits

$|0\rangle\otimes|0\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle) \Rightarrow \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

$|0\rangle\otimes|1\rangle$ 变换成

$$\left(\frac{1}{\sqrt{2}}|0\rangle+\frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle\right)=\frac{1}{2}(|00\rangle-|01\rangle+|10\rangle-|11\rangle) \Rightarrow \begin{bmatrix}0\\1\\0\\0\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}$$

（感谢gitee网友酹江月指出此页向量表示系数的错误）

2. Deutsch-Jozsa algorithm

- Hadamard matrix
  - Basis transformation matrix

$$\begin{bmatrix}1\\0\\0\\0\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix} \qquad \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\0\\0\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix} \qquad \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \text{变换成} \frac{1}{2}\begin{bmatrix}1\\-1\\-1\\1\end{bmatrix}$$

$$\Rightarrow \quad \boldsymbol{H}^{\otimes 2} = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\1 & -1 & 1 & -1\\1 & 1 & -1 & -1\\1 & -1 & -1 & 1\end{bmatrix}$$

2. Deutsch-Jozsa algorithm

- Kronecker product
  - Hadmard gate for two qubits

$$H^{\otimes 2} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} & \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} & -\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{H} & \boldsymbol{H} \\ \boldsymbol{H} & -\boldsymbol{H} \end{bmatrix}$$

2. Deutsch-Jozsa algorithm

- Kronecker product
  - Hadamard gate for three qubits

$$H^{\otimes 3} = \frac{1}{\sqrt{2}}\begin{bmatrix} H^{\otimes 2} & H^{\otimes 2} \\ H^{\otimes 2} & -H^{\otimes 2} \end{bmatrix} = \frac{1}{2\sqrt{2}}\begin{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} & -\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \end{bmatrix}$$

2. Deutsch-Jozsa algorithm

- Kronecker product
  - Hadamard gate for n qubits

$$H^{\otimes n} = \frac{1}{\sqrt{2}} \begin{bmatrix} H^{\otimes(n-1)} & H^{\otimes(n-1)} \\ H^{\otimes(n-1)} & -H^{\otimes(n-1)} \end{bmatrix}$$

（感谢弘毅学堂2020级李天羽同学纠正Hadamard单词拼写错误）

2. Deutsch-Jozsa algorithm

- N-bit Deutsch oracle problem
  - $f(b_1, b_2, \cdots, b_n)$ where $b_i \in \{0, 1\}$
  - f is either constant function (always output 0 or 1) or balanced function (half output 0 and half output 1)
  - How many steps to classify the type of f

2. Deutsch-Jozsa algorithm

- N-bit Deutsch oracle problem
  - Example: 3-bit condition

$$(0,0,0),(0,0,1),(0,1,0),(0,1,1),(1,0,0),(1,0,1),(1,1,0),(1,1,1)$$

  - Best case: 2 times

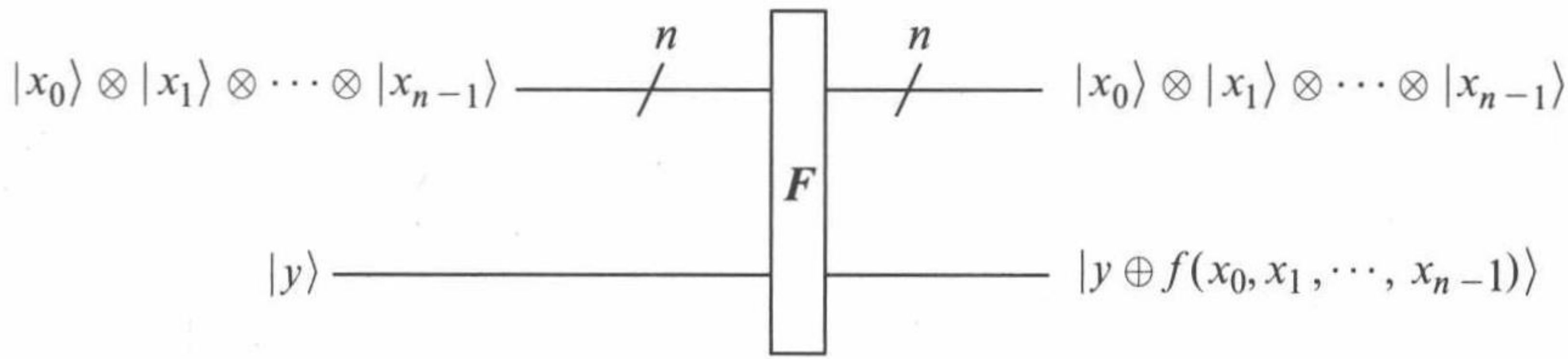
$$f(0,0,0)=1 \text{ and } f(0,0,1)=0$$

  - Worst case: $2^{n-1}+1$ times

$$f(0,0,0)=1,\ f(0,0,1)=1,\ f(0,1,0)=1,\ f(0,1,1)=1$$

2. Deutsch-Jozsa algorithm

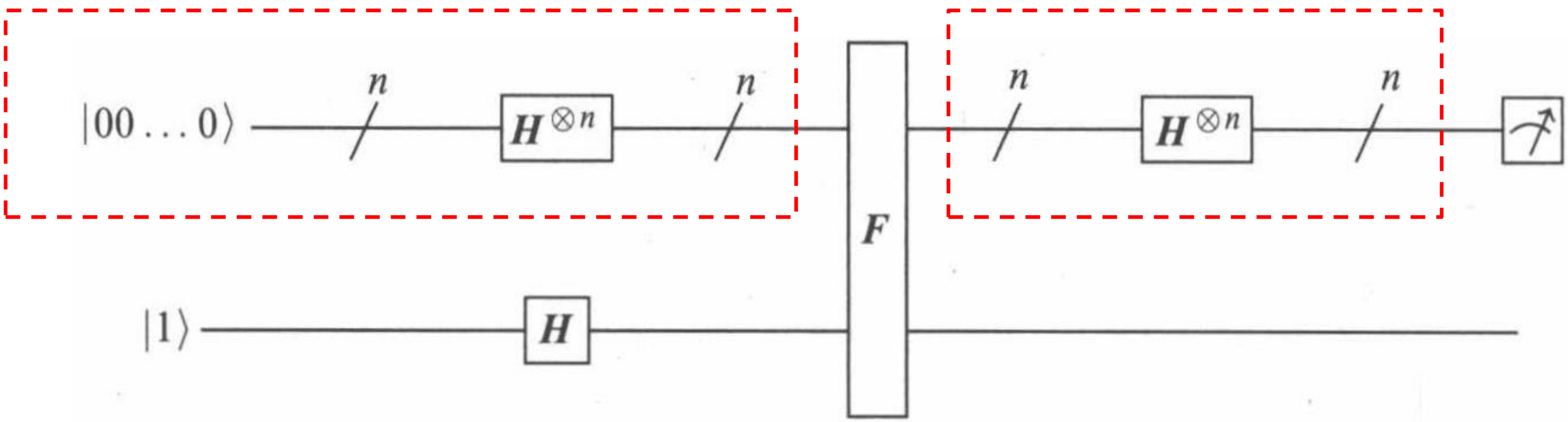
- ***F*** gate for n-bit function



这个电路告诉我们：每个量子比特 $|x_i\rangle$（要么是 $|0\rangle$ 要么是 $|1\rangle$）会如何变化。输入由 $n+1$ 个 ket 组成——$|x_0\rangle \otimes |x_1\rangle \otimes \cdots \otimes |x_{n-1}\rangle$ 和 $|y\rangle$，其中前 n 个 ket 对应于函数变量。输出也由 $n+1$ 个 ket 组成，其中前 n 个 ket 的输出与前 n 个 ket 的输入完全相同。如果 $y=0$，最后一位输出是 $|f(x_0,x_1,\cdots,x_{n-1})\rangle$；如果 $y=1$，最后一位输出是 $|f(x_0,x_1,\cdots,x_{n-1})\rangle$ 的相反值。

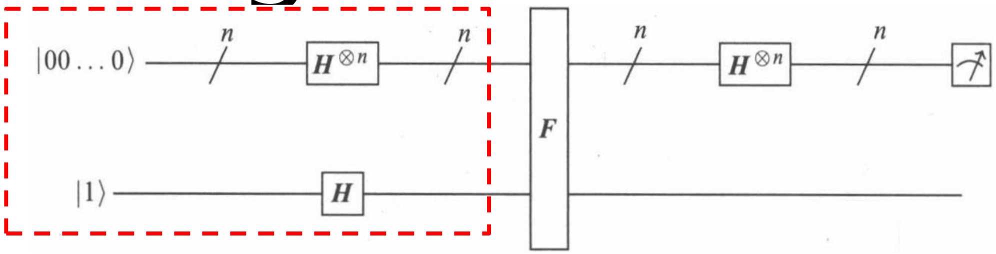
2. Deutsch-Jozsa algorithm

■ Deutsch-Jozsa algorithm



- 所有顶部的量子比特都通过***F***门两侧的Hadamard门

2. Deutsch-Jozsa algorithm



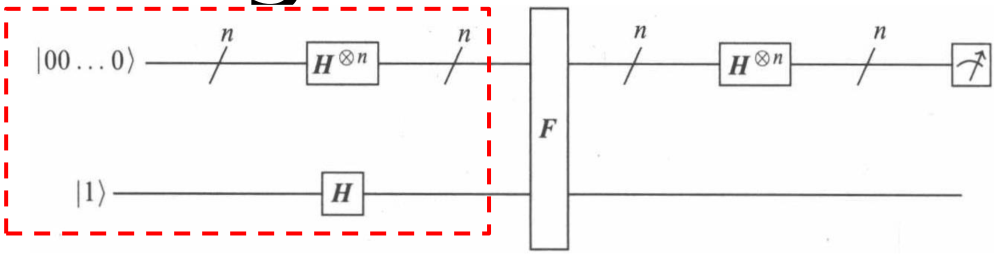
- D-J algorithm
  - Step 1: (3) qubits pass through Hadamard gate

$$H^{\otimes 2}(|00\rangle)=\frac{1}{2}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}=\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)$$

所有基态的叠加，且每个基态具有相同概率幅：$\left(\frac{1}{\sqrt{2}}\right)^n$

$$H(|1\rangle)=\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\end{bmatrix}-\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\end{bmatrix}=\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle$$

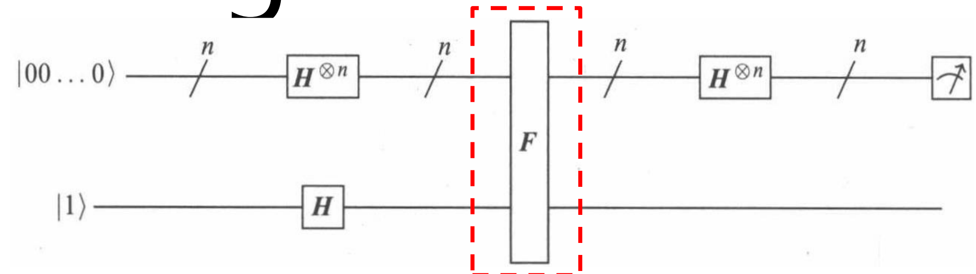
2. Deutsch-Jozsa algorithm



- D-J algorithm
  - Step 1: (3) qubits pass through Hadamard gate
    - Output state

$$\frac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle-\frac{1}{\sqrt{2}}|1\rangle\right) \Leftrightarrow \begin{aligned}&\frac{1}{2\sqrt{2}}|00\rangle\otimes(|0\rangle-|1\rangle)\\&+\frac{1}{2\sqrt{2}}|01\rangle\otimes(|0\rangle-|1\rangle)\\&+\frac{1}{2\sqrt{2}}|10\rangle\otimes(|0\rangle-|1\rangle)\\&+\frac{1}{2\sqrt{2}}|11\rangle\otimes(|0\rangle-|1\rangle)\end{aligned}$$

2. Deutsch-Jozsa algorithm

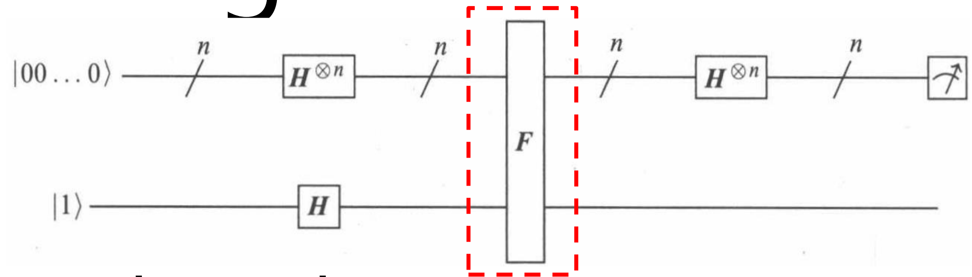


- D-J algorithm
  - Step 2: (3) qubits pass through ***F*** gate
    - Output state

$$
\begin{aligned}
&\frac{1}{2\sqrt{2}}|00\rangle\otimes(|f(0,0)\rangle-|f(0,0)\oplus 1\rangle)\\
&+\frac{1}{2\sqrt{2}}|01\rangle\otimes(|f(0,1)\rangle-|f(0,1)\oplus 1\rangle)\\
&+\frac{1}{2\sqrt{2}}|10\rangle\otimes(|f(1,0)\rangle-|f(1,0)\oplus 1\rangle)\\
&+\frac{1}{2\sqrt{2}}|11\rangle\otimes(|f(1,1)\rangle-|f(1,1)\oplus 1\rangle)
\end{aligned}
\Rightarrow
\begin{aligned}
&(-1)^{f(0,0)}\frac{1}{2}|00\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\
&+(-1)^{f(0,1)}\frac{1}{2}|01\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\
&+(-1)^{f(1,0)}\frac{1}{2}|10\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\\
&+(-1)^{f(1,1)}\frac{1}{2}|11\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)
\end{aligned}
$$

当 $a=0$ 或 $a=1$ 时，我们有如下事实：$|a\rangle-|a\oplus 1\rangle=(-1)^a(|0\rangle-|1\rangle)$

2. Deutsch-Jozsa algorithm



- D-J algorithm
  - Step 2: (3) qubits pass through ***F*** gate
    - Output state

此处存在书写错误，应该为指数（后面更正）

非纠缠的。顶部的两个量子比特有如下状态

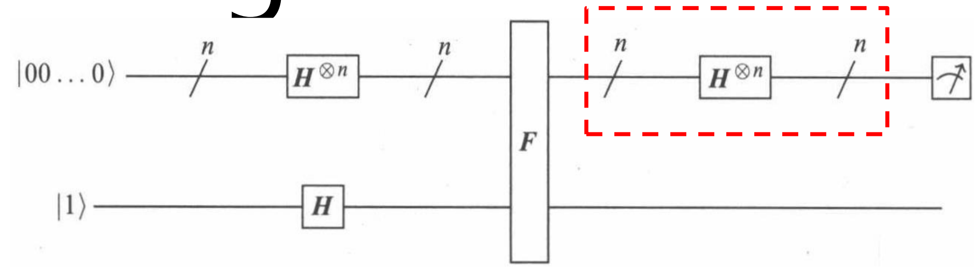
$$\frac{1}{2}((-1)f^{(0,0)}|00\rangle+(-1)f^{(0,1)}|01\rangle+(-1)f^{(1,0)}|10\rangle+(-1)f^{(1,1)}|11\rangle)$$

（对于一般的 n，该论证也成立。这时你有一个包含所有基态的叠加态，任意基态 $|x_0x_1\cdots x_{n-1}\rangle$ 对应的系数为 $\left(\frac{1}{\sqrt{2}}\right)^n(-1)^{f(x_0,x_1,\cdots,x_{n-1})}$。）

$$(-1)^{f(0,0)}\frac{1}{2}|00\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(0,1)}\frac{1}{2}|01\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(1,0)}\frac{1}{2}|10\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$
$$+(-1)^{f(1,1)}\frac{1}{2}|11\rangle\otimes\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$$

顶部两个量子比特与底部量子比特非纠缠

2. Deutsch-Jozsa algorithm

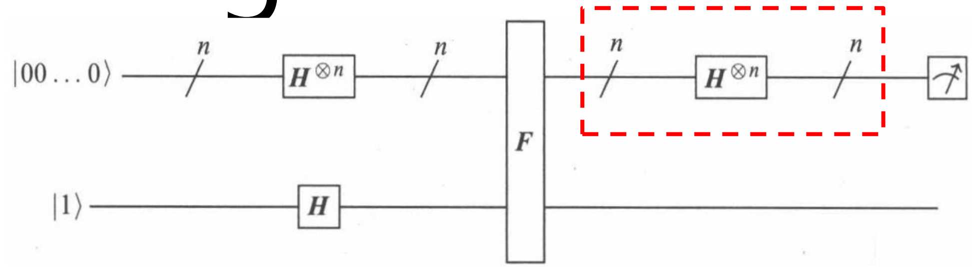


- D-J algorithm
  - Step 3: upper (2) qubits pass through Hadamard gate
    - Output state

$$\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}(-1)^{f(0,0)}\\ (-1)^{f(0,1)}\\ (-1)^{f(1,0)}\\ (-1)^{f(1,1)}\end{bmatrix}$$

    - 顶部元素对应状态 $|00\rangle$ 的振幅

2. Deutsch-Jozsa algorithm



- D-J algorithm
  - Step 3: upper (2) qubits pass through Hadmard gate

$$\frac{1}{4}((-1)^{f(0,0)}+(-1)^{f(0,1)}+(-1)^{f(1,0)}+(-1)^{f(1,1)})$$

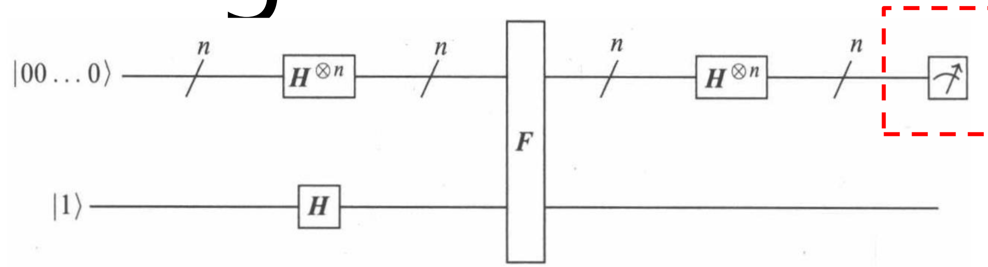
这是 $|00\rangle$ 的概率振幅。我们计算两种函数的 $|00\rangle$ 的概率振幅的结果：

如果 f 是常值函数，任意输入对应的输出都是 0，那么概率振幅为 1。

如果 f 是常值函数，任意输入对应的输出都是 1，那么概率振幅为 −1。

如果是平衡函数，那么概率振幅为 0。

2. Deutsch-Jozsa algorithm



- D-J algorithm
  - Step 4: measure upper (2) qubits

当我们测量顶部的量子比特时，会得到 00、01、10 或 11 中的一个。问题变成了“我们是否可以得到 00？”如果函数是常值函数，那么我们得到 00 的概率是 1；如果函数是平衡函数，我们得到 00 的概率是 0。因此，当测量结果是 00 时，函数就是常值函数；否则，就是平衡函数。

2. Deutsch-Jozsa algorithm

- D-J algorithm
  - Discussion

因此，无论 n 的取值是多少，仅需查询一次 oracle，我们就可以解决 Deutsch-Jozsa 问题。回想一下经典的例子，最坏的情况需要查询 $2^{n-1}+1$ 次，所以改进是巨大的。

Conclusion

- Deutsch's algorithm
  - Deutsch's oracle problem
  - Reversible and irreversible operators
  - Deutsch's algorithm
- Deutsch-Jozsa algorithm
  - Hadamard matrix and Kronecker product
  - N-bit Deutsch oracle problem
  - Deutsch-Jozsa algorithm