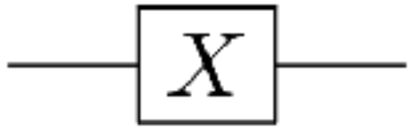
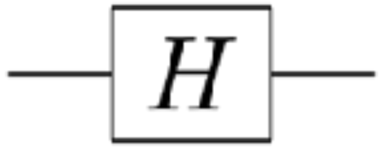
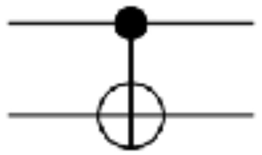
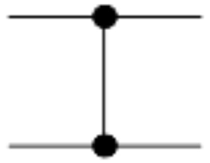
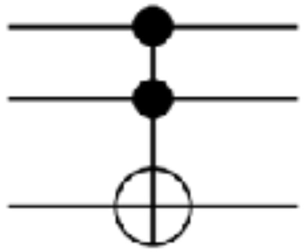
Quantum Algorithms (2)

Hung-Wei Tseng

Recap: Qiskit & quantum gates

Gate	Input/Output bits	Symbol	Transition Matrix	Qiskit Method
Pauli-X / NOT / Bit-flip	1-bit		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	QuantumCircuit.x
Pauli Y	1-bit		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	QuantumCircuit.y
Pauli Z / Phase flip	1-bit		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	QuantumCircuit.z
Hadamard	1-bit		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	QuantumCircuit.h
Controlled NOT	2-bit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	QuantumCircuit.cx
Controlled Z	2-bit		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	QuantumCircuit.cz
Toffoli	3-bit		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	QuantumCircuit.ccx

Recap: The Deutsch-Jozsa Problem

- Given a hidden Boolean function f, which takes as input a string of bits, and returns either $\mathbf{0}$ or $\mathbf{1}$, that is:

$f(\{x_0, x_1, x_2, \ldots\}) \rightarrow 0$ or 1 , where x_n is 0 or 1

- The given Boolean function is that it is guaranteed to either be balanced or constant
  - A constant function returns all 0s or all 1s for any input
  - A balanced function returns 0s for exactly half of all inputs and 1s for the other half
- Our task is to determine whether the given function is balanced or constant

David Deutsch and Richard Jozsa (1992). "Rapid solutions of problems by quantum computation". Proceedings of the Royal Society of London A. 439: 553–558. doi:10.1098/rspa.1992.0167

Recap: The classical solution

- Let's start by choosing two numbers and test their outputs
  - if $f(0,0,0,\ldots) \to 0$ and $f(1,0,0,\ldots) \to 1$, then we know the given one is a balanced one!
  - What if $f(0,0,0,\ldots) \to 0$ and $f(1,0,0,\ldots) \to 0$? We have to try one more run…
- The worst case will need to go through exactly half of the input space + 1, that is $2^{n-1} + 1$, numbers
- The classical solution is therefore $O(2^n)$

Recap: An overview of Deutsch-Jozsa Algorithm

- Initialize n + 1 qubits
- Transform these qubits into Hadamard basis: making each qubit 50%-50% of being 0 or 1

$$|\psi_1 = \sum_{x\in\{0,1\}^n} H|x\rangle = \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}}|x\rangle(|0\rangle - |1\rangle)$$

- Encode the given function as an unitary matrix (i.e., oracle)

$$U_f|\psi_1\rangle = |\psi_2 = \sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}}(-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)$$

- Return the qubits for measurements

- Measure the qubits to obtain the solution

$$|\psi_3\rangle = \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\left[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right]$$

$$= \frac{1}{2^n}\sum_{y=0}^{2^n-1}\left[\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle$$

- This algorithm is simply $O(n)$

The Bernstein-Vazirani Algorithm

The Bernstein-Vazirani Problem

- Given a hidden Boolean function f, which takes as input a string of bits, and returns either $\mathbf{0}$ or $\mathbf{1}$, that is:

  $f(\{x_0, x_1, x_2, \ldots\}) \to 0$ or 1 , where x_n is 0 or 1
- The given function is guaranteed to return the bitwise product of the input with some string, s . In other words, given an input x , $f(x) = s \cdot x \pmod 2$.
- We are expected to find s .

Ethan Bernstein and Umesh Vazirani (1997) "Quantum Complexity Theory" SIAM Journal on Computing, Vol. 26, No. 5: 1411-1473, doi:10.1137/S0097539796300921.

The classical solution

- Given an input x . Thus, the hidden bit string s can be revealed by querying the oracle with the sequence of inputs $\{0^m10^k\}$, m+k+1=n
- This means we would need to call the function $f_s(x)$, n times.

Hadamard gate

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

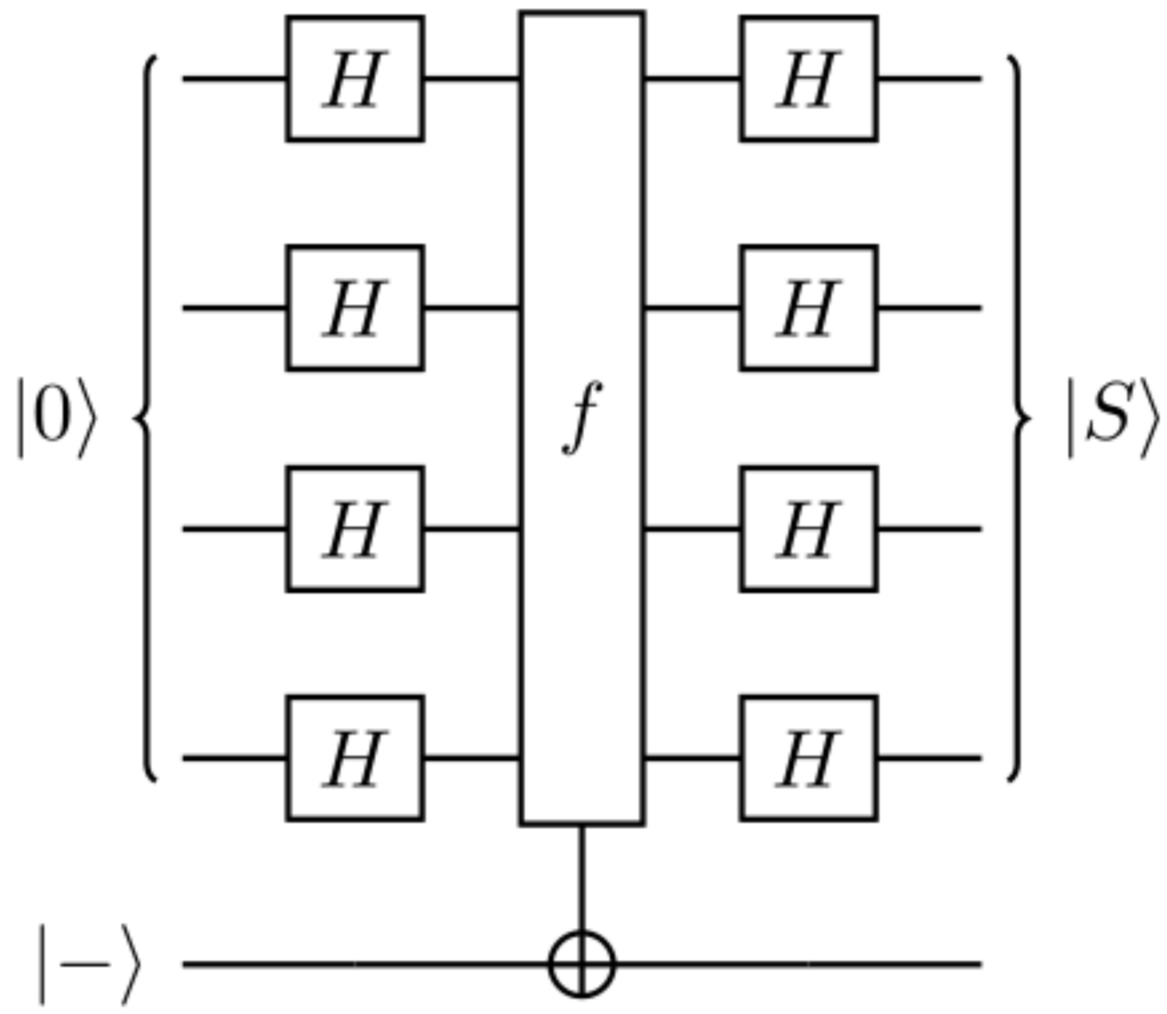
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} = |-\rangle$$

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

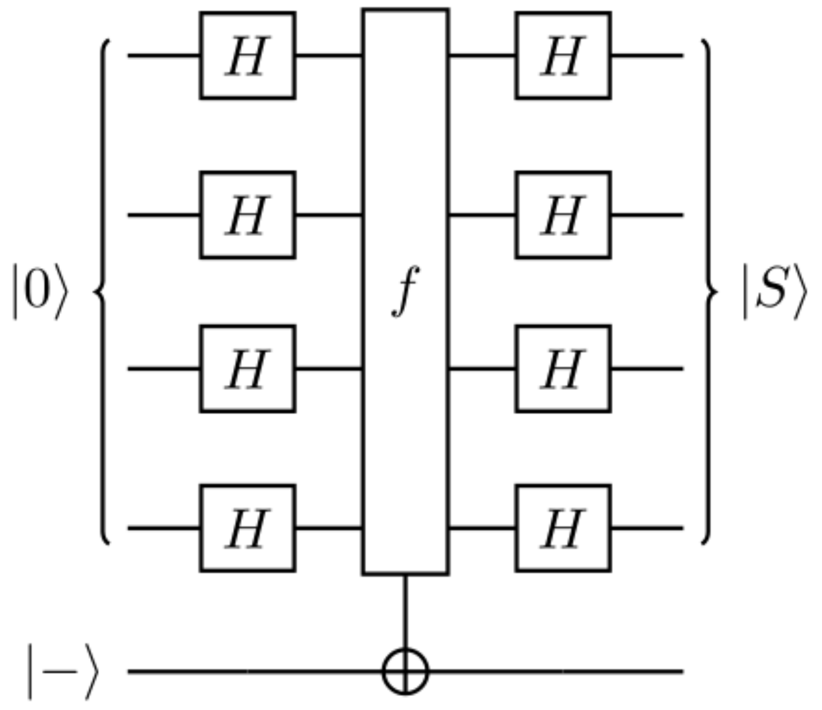
$$H|-\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$$

The quantum solution

- Initialize the inputs qubits to the $|0\rangle^{\otimes n}$ state, and output qubit to $|-\rangle$.
- Apply Hadamard gates to the input register
- Query the oracle
- Apply Hadamard gates to the input register
- Measure
- we can solve this problem with 100% confidence after only one call to the function $f(x)$



The quantum solution



- Initialize the inputs qubits to the $|0\rangle^{\otimes n}$ state, and output qubit to $|-$
- Apply Hadamard gates to the input register

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

- Query the oracle f that use the same phase kickback trick from the Deutsch-Jozsa algorithm and act on a qubit in the state $|-\rangle$

$$|x\rangle \xrightarrow{f_s} (-1)^{s\cdot x}|x\rangle$$

- Apply Hadamard gates to the input register

$$\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{s\cdot x}|x\rangle \xrightarrow{H^{\otimes n}} |s\rangle$$

- Measure
- we can solve this problem with 100% confidence after only one call to the function $f(x)$

Simon's Algorithm

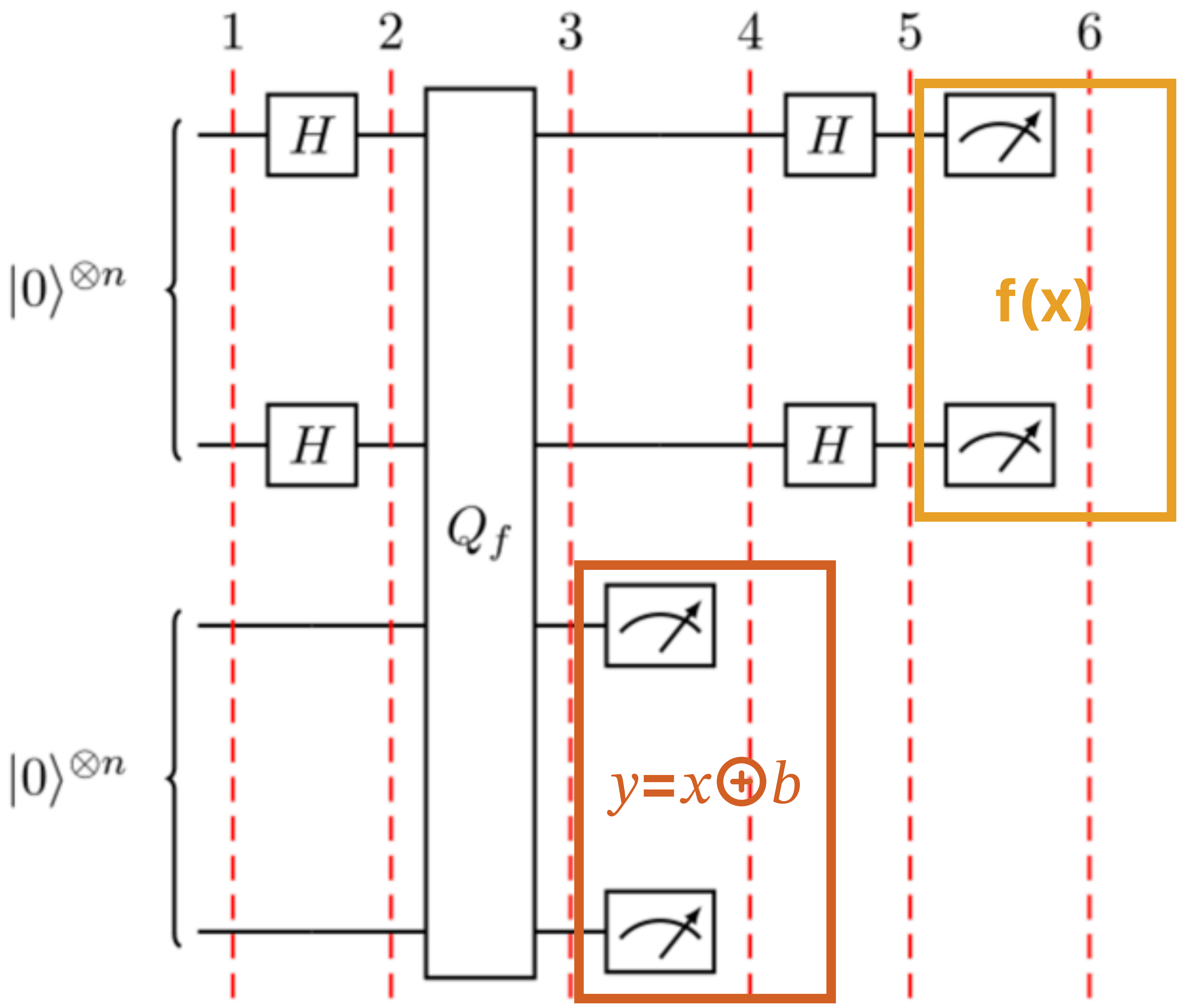
Simon's Problem

- Given an unknown blackbox function f, which is guaranteed to be either one-to-one (1:1) or two-to-one (2:1), where one-to-one and two-to-one functions have the following properties
  - one-to-one: maps exactly one unique output for every input. An example with a function that takes 4 inputs is: $f(1) \to 1, f(2) \to 2, f(3) \to 3, f(4) \to 4$
  - two-to-one: maps exactly two inputs to every unique output. An example with a function that takes 4 inputs is: $f(1) \to 1, f(2) \to 2, f(3) \to 1, f(4) \to 2$
    - This two-to-one mapping is according to a hidden bitstring, b, where: given $x_1, x_2 : f(x_1) = f(x_2)$ it is guaranteed : $x_1 \oplus x_2 = b$
- Given this blackbox f, how quickly can we determine if f is one-to-one or two-to-one? Then, if f turns out to be two-to-one, how quickly can we determine b? As it turns out, both cases boil down to the same problem of finding b, where a bitstring of $b = 000\ldots$ represents the one-to-one f.

The classical solution

- Checking just over half of all the possible inputs until we find two cases of the same output
- Worst case $O(2^{n-1} + 1)$

The quantum solution



The Quantum Solution

- Two n-qubit input registers are initialized to the zero state
  $|\psi_1\rangle = |0\rangle^{\otimes n}|0\rangle^{\otimes n}$
- Apply a Hadamard transform to the first register
  $$|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|0\rangle^{\otimes n}$$
- Apply the query function Q$_f$
  $$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle|f(x)\rangle$$
- Measure the second register. A certain value of $f(x)$ will be observed. Because of the setting of the problem, the observed value $f(x)$ could correspond to two possible inputs: x and $y=x\oplus b$. Therefore the first register becomes
  $$|\psi_4\rangle = \frac{1}{\sqrt{2}}\left(|x\rangle + |y\rangle\right)$$
- Apply Hadamard on the first register
  $$|\psi_5\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{z\in\{0,1\}^n} \left[(-1)^{x\cdot z} + (-1)^{y\cdot z}\right]|z\rangle$$

The Quantum Solution

- Measuring the first register will give an output only if $(-1)^{x \cdot z} = (-1)^{y \cdot z}$, which means
  $x \cdot z = y \cdot z$

  $x \cdot z = \left(x \oplus b\right) \cdot z$

  $x \cdot z = x \cdot z \oplus b \cdot z$

  $b \cdot z = 0$ (mod 2)

  A string z will be measured, whose inner product with b=0 . Thus, repeating the algorithm $\approx n$ times, we will be able to obtain n different values of z

Quantum Fourier Transform (QFT)

DFT and QFT

- The discrete Fourier transform acts on a vector $(x_0,\ldots,x_{N-1})$ and maps it to the vector $(y_0,\ldots,y_{N-1})$ according to the formula
  $y_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j\omega_N^{jk}$, where $\omega_N^{jk} = e^{2\pi i\frac{jk}{N}}$
- the quantum Fourier transform acts on a quantum state $|X\rangle = \sum_{j=0}^{N-1} x_j|j\rangle$ and maps it to the quantum state
  $|Y\rangle = \sum_{k=0}^{N-1} y_k|k\rangle$ according to the formula
  $y_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j\omega_N^{jk}$
  - Only the amplitudes of the state were affected by this transformation.
  - $|j\rangle \mapsto \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_N^{jk}|k\rangle$
  - $U_{QFT} = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\omega_N^{jk}|k\rangle\langle j|$

QFT

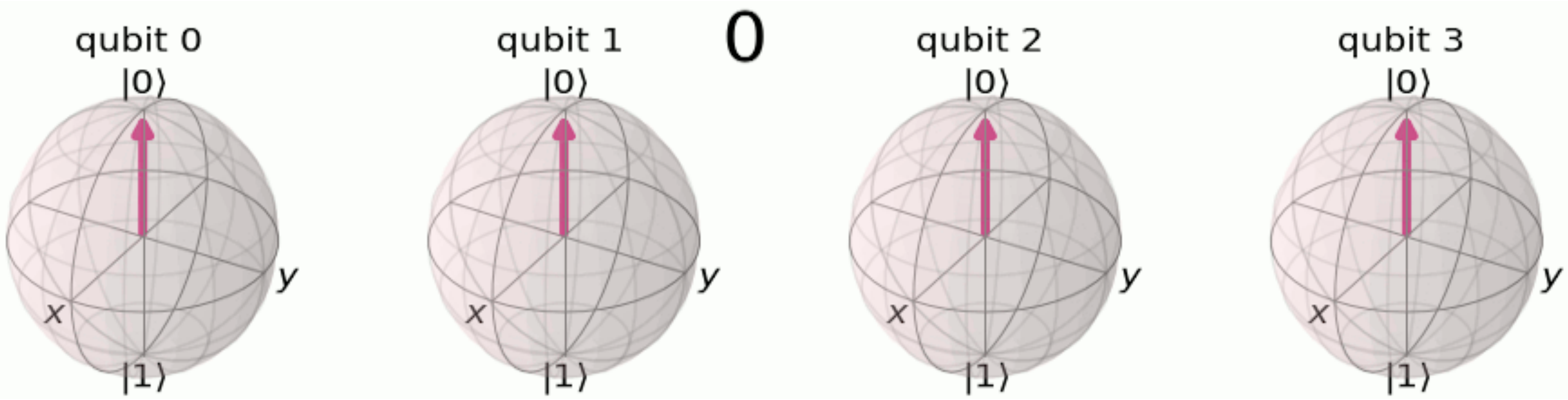
- Transforms between two bases, the computational (Z) basis, and the Fourier basis.

$$|\text{State in Computational Basis}\rangle \xrightarrow{\text{QFT}} |\text{State in Fourier Basis}\rangle$$

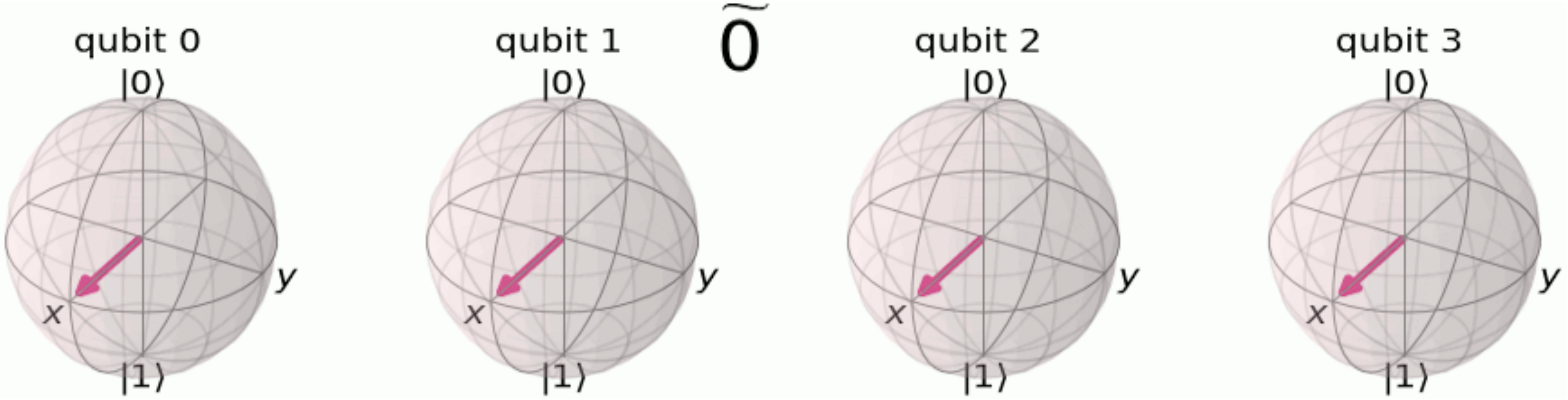
$$\text{QFT}\,|x\rangle = |\widetilde{x}\rangle$$

Counting in different basis

0



$\tilde{0}$



1-qubit QFT

- $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, x_0 = \alpha, x_1 = \beta$, and $N = 2$.
- Then,

$$y_0 = \frac{1}{\sqrt{2}}\left(\alpha \exp\left(2\pi i \frac{0\times 0}{2}\right) + \beta \exp\left(2\pi i \frac{1\times 0}{2}\right)\right) = \frac{1}{\sqrt{2}}(\alpha+\beta)$$

and

$$y_1 = \frac{1}{\sqrt{2}}\left(\alpha \exp\left(2\pi i \frac{0\times 1}{2}\right) + \beta \exp\left(2\pi i \frac{1\times 1}{2}\right)\right) = \frac{1}{\sqrt{2}}(\alpha-\beta)$$

- $U_{QFT}|\psi\rangle = \frac{1}{\sqrt{2}}(\alpha+\beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|1\rangle$

$$= H|\psi\rangle = H|\alpha|0\rangle + \beta|1\rangle = \frac{1}{\sqrt{2}}(\alpha+\beta)|0\rangle + \frac{1}{\sqrt{2}}(\alpha-\beta)|1\rangle \equiv \tilde{\alpha}|0\rangle + \tilde{\beta}|1\rangle$$

Generalized QFT

- Given $|x\rangle = |x_1 \ldots x_n\rangle$

$$QFT_N|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{xy} |y\rangle$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i xy/2^n} |y\rangle$$ since $\omega_N^{xy} = e^{2\pi i \frac{xy}{N}}$ and $N = 2^n$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i \left(\sum_{k=1}^{n} y_k/2^k \right) x} |y_1 \ldots y_n\rangle$$ rewriting in fractional binary notation $y = y_1 \ldots y_n$, $y/2^n = \sum_{k=1}^{n} y_k/2^k$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{2\pi i x y_k/2^k} |y_1 \ldots y_n\rangle$$ after expanding the exponential of a sum to a product of exponentials

$$= \frac{1}{\sqrt{N}} \bigotimes_{k=1}^{n} \left(|0\rangle + e^{2\pi i x/2^k} |1\rangle \right)$$ after rearranging the sum and products, and expanding $\sum_{y=0}^{N-1} = \sum_{y_1=0}^{1} \sum_{y_2=0}^{1} \ldots \sum_{y_n=0}^{1}$

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2\pi i}{2} x} |1\rangle \right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^2} x} |1\rangle \right) \otimes \ldots \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^{n-1}} x} |1\rangle \right) \otimes \left(|0\rangle + e^{\frac{2\pi i}{2^n} x} |1\rangle \right)$$

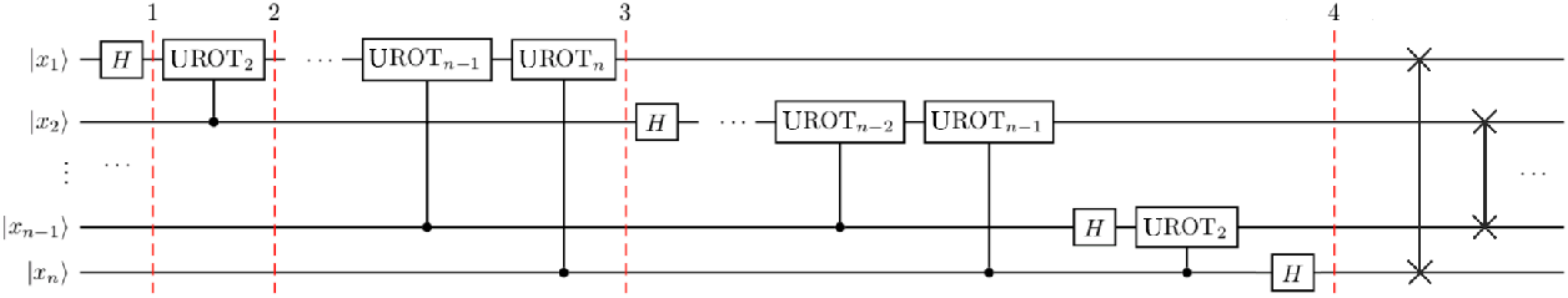
Implementing QFT

- $H|x_k\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + \exp\left(\frac{2\pi i}{2}x_k\right)|1\rangle\right)$

$CROT_k = \begin{bmatrix} I & 0 \\ 0 & UROT_k \end{bmatrix}$, where $UROT_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi i}{2^k}\right) \end{bmatrix}$

$CROT_k|0x_j\rangle = |0x_j\rangle$ and $CROT_k|1x_j\rangle = \exp\left(\frac{2\pi i}{2^k}x_j\right)|1x_j\rangle$

The quantum circuit



The implementation of rotation

P-Gate

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$$

S-Gate

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{2}} \end{bmatrix}, \quad S^{\dagger} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{2}} \end{bmatrix}$$

T-Gate

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}, \quad T^{\dagger} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{i\pi}{4}} \end{bmatrix}$$

U-Gate

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\phi}\sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)}\cos(\frac{\theta}{2}) \end{bmatrix}$$

$$U(\frac{\pi}{2},0,\pi) = \frac{1}{\sqrt{2}} \begin{bmatrix} 11 \\ 1-1 \end{bmatrix} = H \qquad U(0,0,\lambda) = \begin{bmatrix} 10 \\ 0e^{i\lambda} \end{bmatrix} = P$$