The Quantum Fourier Transform and Its Applications
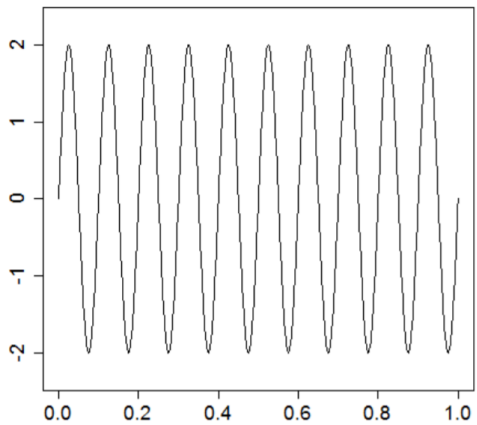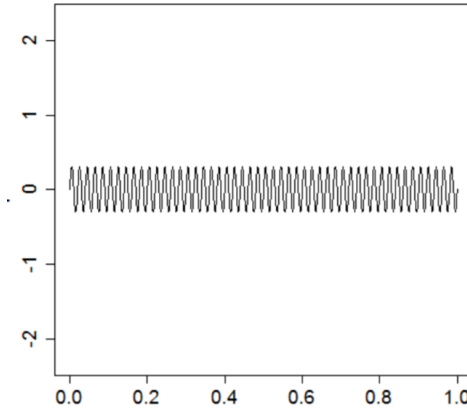
Claire Yang　　　2019/02/14

Discrete Fourier Transform
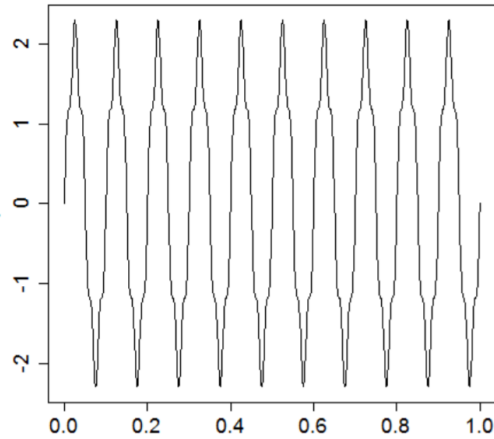
Input signal
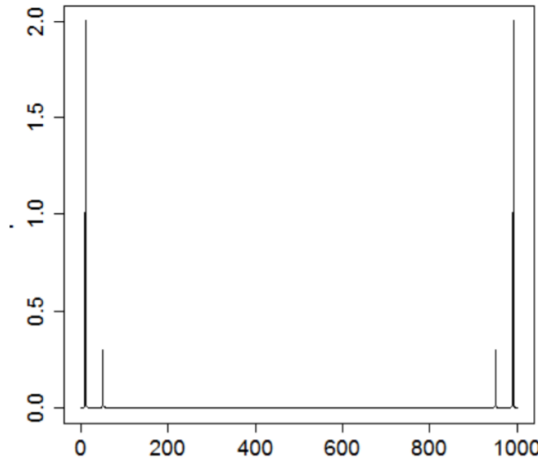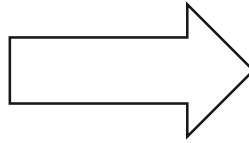
Input signal in frequency domain

True signal: f=10, A=2

Noise signal: f=50, A=0.3

QFT is the quantum analogue of DFT:

— Map signal from **time** domain to **frequency** domain on quantum computer with a faster speed

— Basis for **quantum phase estimation**

Pictures: https://arxiv.org/pdf/1804.10068.pdf

Quantum Fourier Transform

Input

$x_1, x_2, \ldots, x_N$

Length: N

DFT: $y_k \equiv \dfrac{1}{\sqrt{N}} \displaystyle\sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$

Output

$y_1, y_2, \ldots, y_N$

Length: N

$|x_1, x_2, \ldots, x_n\rangle$

Length: n

QFT: $|j\rangle \longrightarrow \dfrac{1}{\sqrt{N}} \displaystyle\sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle$

$|y_1, y_2, \ldots, y_n\rangle$

Length: n

$$N = 2^n$$

Quantum Fourier Transform

QFT: $|j\rangle \longrightarrow \dfrac{1}{\sqrt{N}} \displaystyle\sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle$

$$|x_1, x_2, ..., x_n\rangle \Rightarrow \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \cdots & \omega_n^{N-1} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \cdots & \omega_n^{2(N-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \cdots & \omega_n^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_n^{N-1} & \omega_n^{2(N-1)} & \omega_n^{3(N-1)} & \cdots & \omega_n^{(N-1)(N-1)} \end{bmatrix} \Rightarrow |y_1, y_2, ..., y_n\rangle$$

Length: n   ⟵ (under $|x_1, x_2, ..., x_n\rangle$)

N×N matrix

Length: n   ⟵ (under $|y_1, y_2, ..., y_n\rangle$)

where $\omega_n := e^{\frac{2\pi i}{2^n}}$
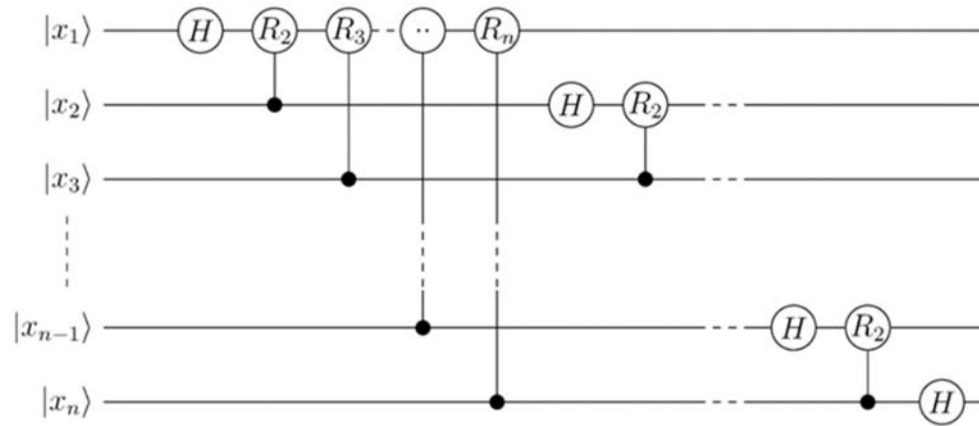
Example (2 qubits):

$$F_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

Quantum Fourier Transform

QFT: $|j\rangle \longrightarrow \dfrac{1}{\sqrt{N}} \sum\limits_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle$

$|x_1, x_2, \dots, x_n\rangle \Rightarrow$
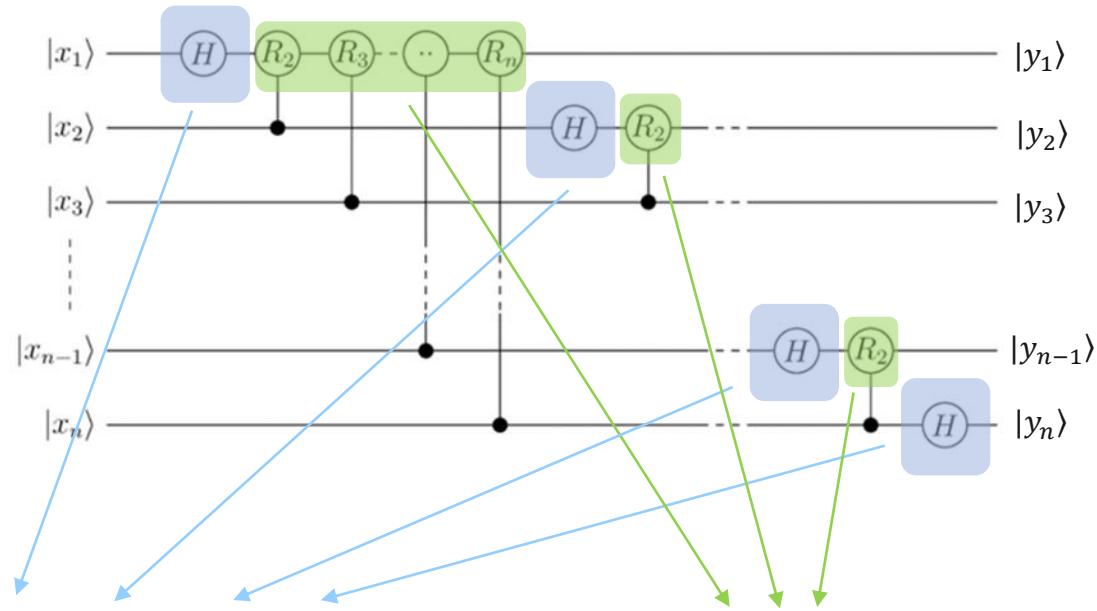
Length: n
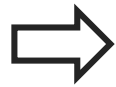
Quantum circuit for QFT

$\Rightarrow |y_1, y_2, \dots, y_n\rangle$

Length: n

Picture: https://www.wikiwand.com/en/Quantum_Fourier_transform

Quantum Fourier Transform

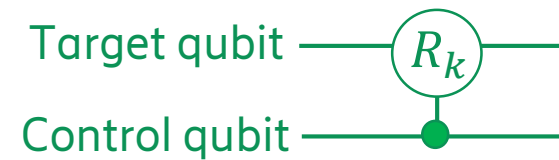## Quantum circuit for QFT

Hadamard gate: create superposition

$$|0\rangle \rightarrow \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|1\rangle \rightarrow \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Controlled phase gate

Target qubit —— R_k ——

Control qubit ——●——

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{k}} \end{bmatrix}$$

Picture: https://www.wikiwand.com/en/Quantum_Fourier_transform

Implementation and comparison with iFFT

$$QFT = iFFT, \quad iQFT = FFT$$

000	⇔ 0
001	⇔ 1
010	⇔ 2
011	⇔ 3
100	⇔ 4
101	⇔ 5
110	⇔ 6
111	⇔ 7

Python

(8-bit)
0100 0000 → **iFFT Algorithm** ⟶ 2.96 μs

One-hot encoding

|001⟩
(3-qubit) → **QFT Algorithm**

IBM QISKit

Local Simulator

IBM Quantum Chip

Same output (wave function)

$$t_{measure} = t_{queue} + t_{run}$$

11.7s   11.2s
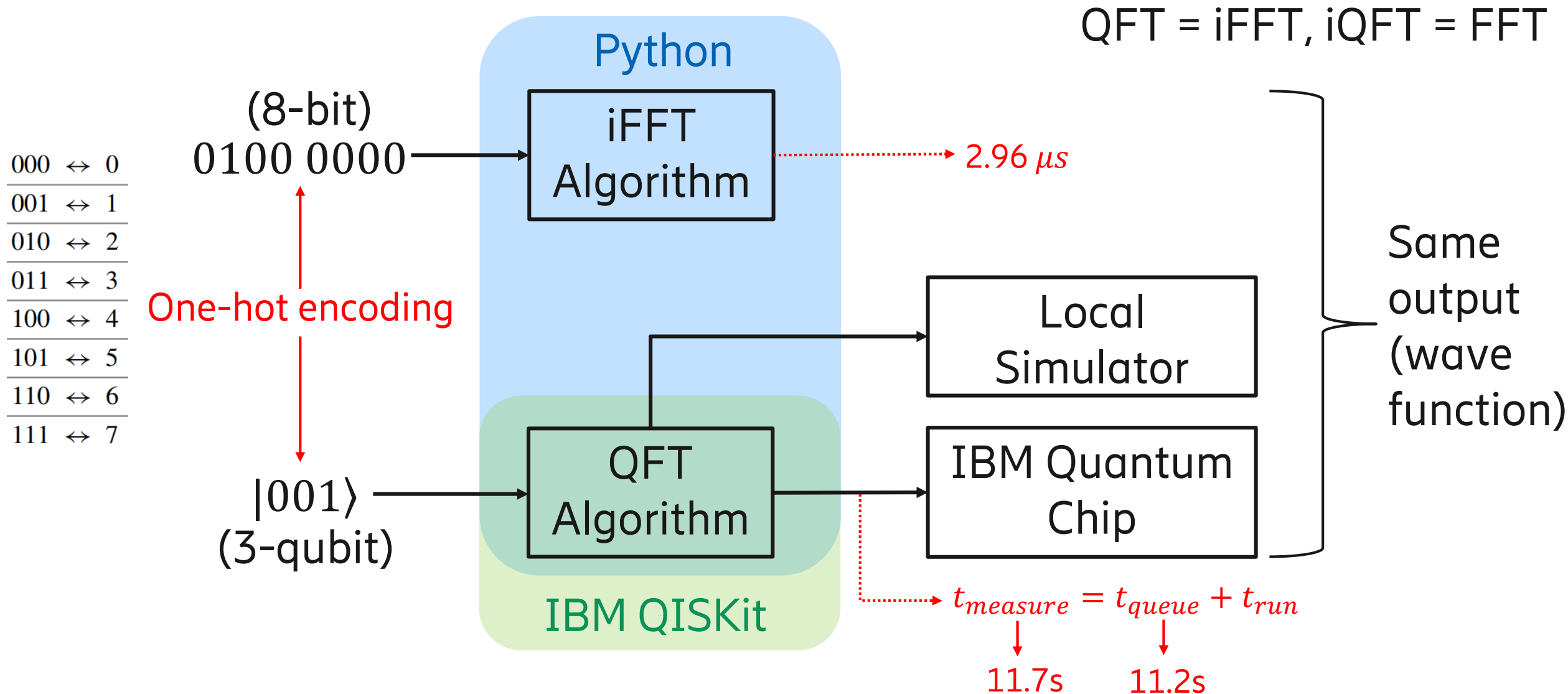
Estimation of the QFT running time

— Add up the delays on the longest path of the compiled circuit

— Single qubit gate: 10 ns, two qubit gate: 100 ns

SWAP gate

$$|x_1\rangle \quad\text{---}\,H\,\text{---}\,R_2\,\text{---}\,R_3\,\text{---}\times\quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_1 x_2 x_3]}|1\rangle\right)$$

$$|x_2\rangle \quad\text{---}\,H\,\text{---}\,R_2\,\text{---}\quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_2 x_3]}|1\rangle\right)$$

$$|x_3\rangle \quad\text{---}\,H\,\text{---}\times\quad \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i[0.x_3]}|1\rangle\right)$$
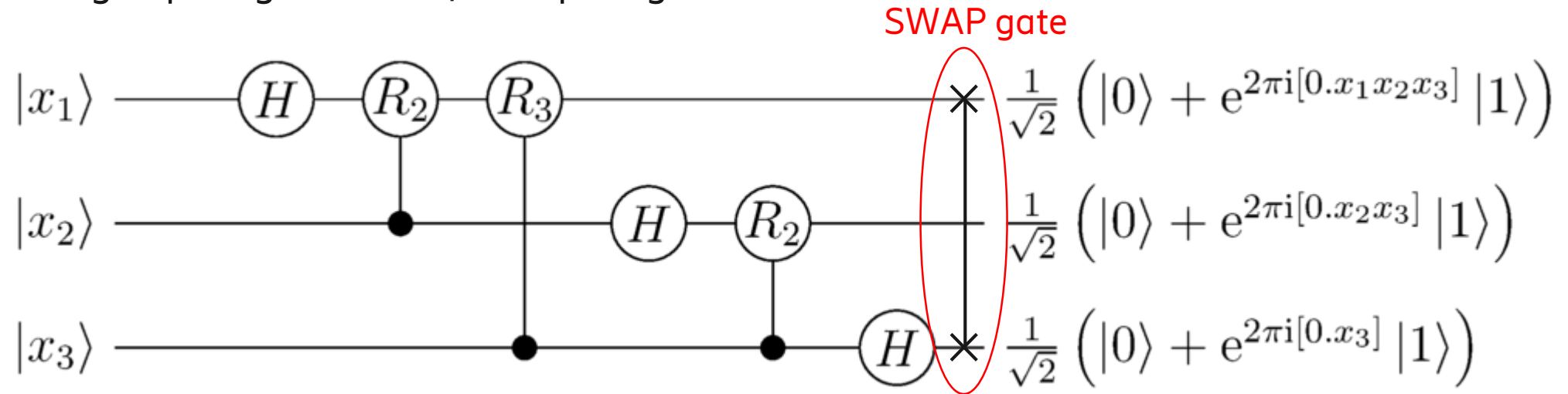
— e.g. For 3-qubit circuit, the estimated running time is:

6 single qubit gates + 1 two qubits gate (SWAP gate) = 160ns

— Computational cost for n qubits (or for 2^n bits): QFT $\Theta(n^2)$, FFT $\Theta(n \cdot 2^n)$

Application: Quantum Phase Estimation

Quantum phase estimation: to calculate the eigenvalue of U

Controlled-U gate

Inverse quantum Fourier transform:
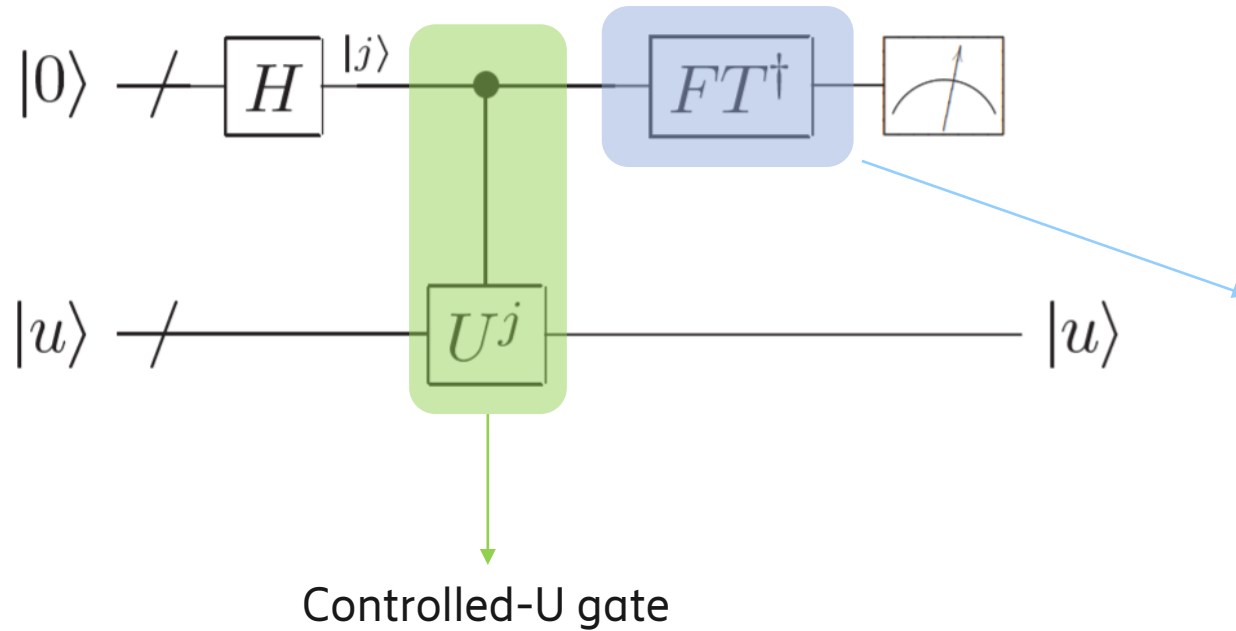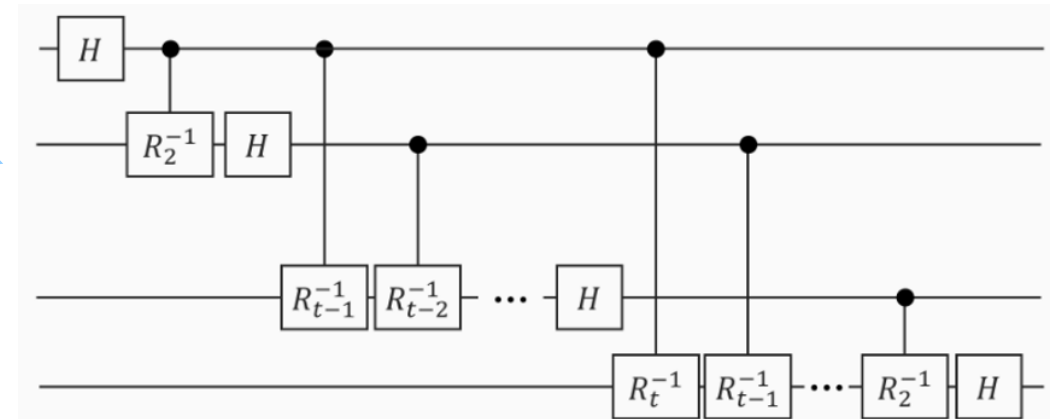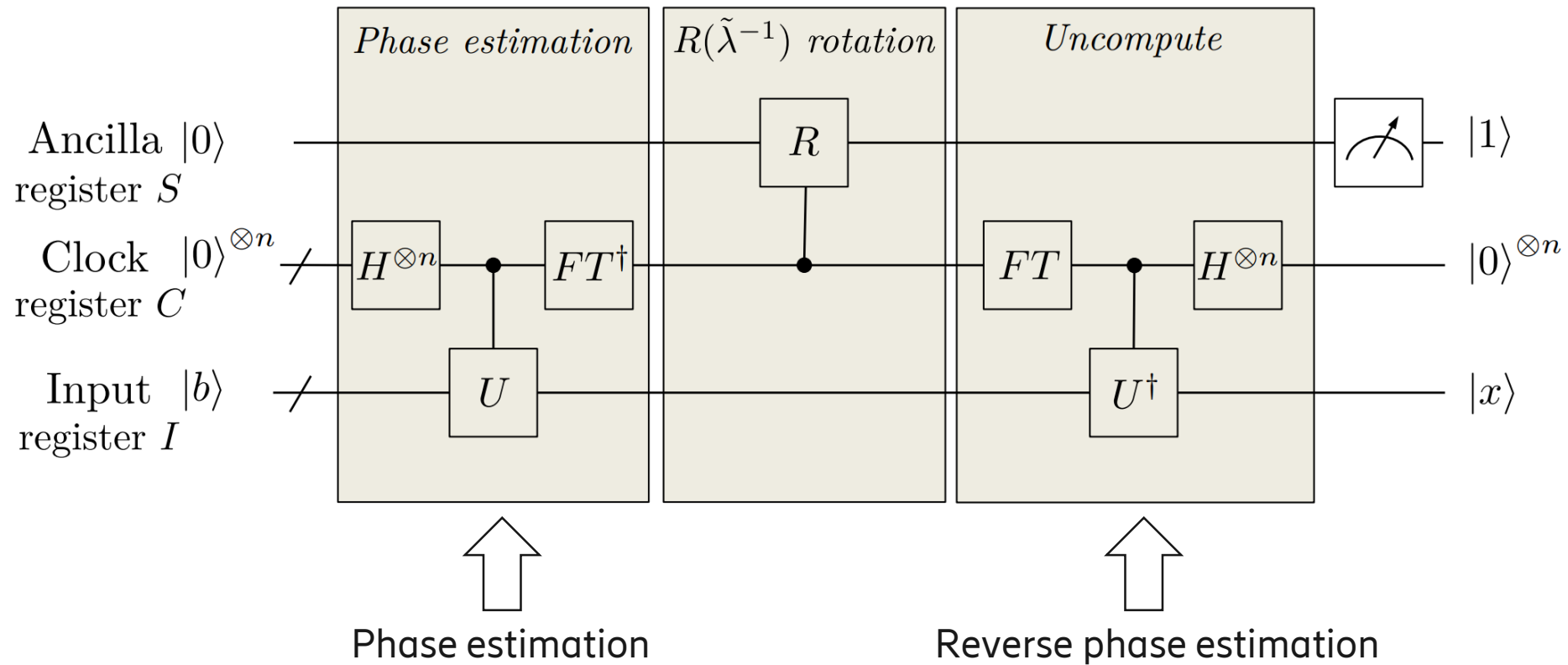

$$|j\rangle \rightarrow \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} e^{-2\pi ijk/2^n} |k\rangle$$

$$R_k{}^{-1} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{2\pi i}{k}} \end{bmatrix}$$

Picture: https://www.qtumist.com/info/PE-H5/index.html

Application: HHL Algorithm - Quantum algorithm for linear systems of equations

Solve $\vec{A}x = \vec{b}$ using a quantum computer

Picture: https://arxiv.org/pdf/1802.08227.pdf

Application: HHL Algorithm- Milestone for QML

Harrow, Hassidim & Lloyd *propose a quantum algorithm to solve linear systems of equations*

Quantum Machine Learning

Quantum Computing

1996

1994

2009   2013

TIME

Peter Shor *proposes a quantum algorithm for prime factoring*

Luv Grover *proposes a quantum algorithm for unordered search*
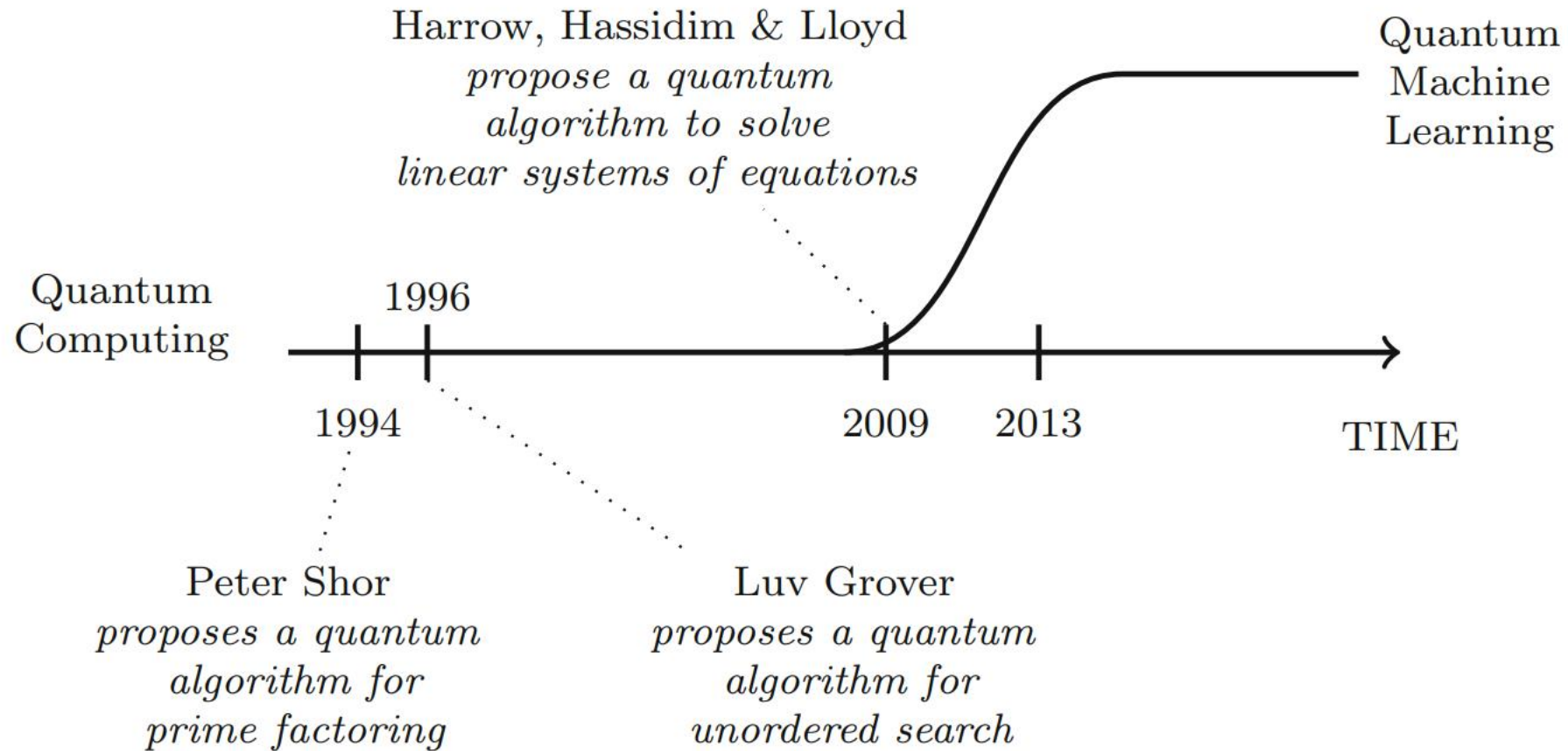
Usage: quantum machine learning algorithms including bayesian inference, qSVM, qPCA...

Picture: M. Schuld and F. Petruccione. Supervised learning with quantum computers. Springer, 2018.

Materials

— Books:

       Quantum Computation and Quantum Information, 2002

       Quantum Machine Learning: What Quantum Computing Means to Data Mining, 2014

       Supervised Learning with Quantum Computers, 2018

— Online courses:

       edX & University of Toronto: Quantum Machine Learning

       edX & TU Delft: Quantum Cryptography