Introduction to Quantum Computing and its Applications to Cyber Security

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@wustl.edu

These slides and audio/video recordings of this class lecture are at:
http://www.cse.wustl.edu/~jain/cse570-19/
Overview

1. What is a Quantum and Quantum Bit?
2. Matrix Algebra Review
3. Quantum Gates: Not, And, or, Nand
4. Applications of Quantum Computing
5. Quantum Hardware and Programming
What is a Quantum?

- **Quantization**: Analog to digital conversion
- **Quantum** = Smallest discrete unit
- **Wave Theory**: Light is a wave. It has a frequency, phase, amplitude
- **Quantum Mechanics**: Light behaves like discrete packets of energy that can be absorbed and released
- **Photon** = One quantum of light energy
- Photons can move an electron from one energy level to next higher level
- Photons are released when an electron moves from one level to lower energy level
Probabilistic Behavior

- Young’s Double-Slit Experiment 1801

- The two waves exiting the slits interfere.
- Interference is constructive at some spots and destructive at others ⇒ Probabilistic
Quantum Bits

1. Computing bit is a binary scalar: 0 or 1
2. Quantum bit (Qubit) is a $2 \times 1$ vector: $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
3. Vector elements of Qubits are complex numbers $x+iy$
4. Modulus of a complex Number $|x+iy| = \sqrt{(x+iy)(x-iy)} = \sqrt{x^2 + y^2}$

Example: $|1 + 2i| = \sqrt{(1+2i)(1-2i)} = \sqrt{1+4} = \sqrt{5}$

5. Probability of each element in a qubit vector is proportional to its modulus squared

For $\begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$

$$P = \frac{|a_0|^2}{|a_0|^2 + |a_1|^2}$$

For $\begin{bmatrix} 1+2i \\ 1-i \end{bmatrix}$

$$|1+2i| = \sqrt{(1+2i)(1-2i)} = \sqrt{5} \Rightarrow P = \frac{5}{5+2} = \frac{5}{7}$$

$$|1-i| = \sqrt{(1-i)(1+i)} = \sqrt{2} \Rightarrow P = \frac{2}{5+2} = \frac{2}{7}$$
Polar Representation

- Complex numbers in polar coordinates:

\[(x + iy) = re^{i\theta} = r\left(\cos(\theta) + isin(\theta)\right)\]

\[r = \sqrt{x^2 + y^2}\]

\[\theta = \tan^{-1}(y / x)\]

\[
\begin{bmatrix}
1 + i \\
-1 + i
\end{bmatrix} = \begin{bmatrix}
\sqrt{2}e^{i\pi/4} \\
\sqrt{2}e^{3\pi/4}
\end{bmatrix} = \begin{bmatrix}
\sqrt{2} (\cos(\pi / 4) + i \sin(\pi / 4)) \\
\sqrt{2} (\cos(3\pi / 4) + i \sin(3\pi / 4))
\end{bmatrix}
\]

- Exercise: Find the complex and polar representation of C
Qubit Interpretation

If a single photon is emitted from the source, the photon reaches position A or B with some probability. 
⇒ Photon has a superposition (rather than position)

Each position has a different path length and, therefore, different amplitude and phase

Washington University in St. Louis http://www.cse.wustl.edu/~jain/ ©2019 Raj Jain
Bra-Ket Notation

- The vector $\psi$ is denoted in bra-kets $|\psi\rangle$
- Brackets: { }, [ ], < >
- Bra $<a|$  
- Ket $|a>$
- Example: Ket-zero and ket-one
  \[
  \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle
  \]
- Bra is the transpose of the complex-conjugate of a Ket.
  Example: Bra-zero and Bra-one
  \[
  \begin{bmatrix} 1 & 0 \end{bmatrix} = <0| \quad \begin{bmatrix} 0 & 1 \end{bmatrix} = <1|
  \]
Matrix Multiplication

- **Matrix multiplication ×**: 

\[
\begin{bmatrix}
  a_{00} & a_{01} & a_{02} \\
  a_{10} & a_{11} & a_{12}
\end{bmatrix} \times \begin{bmatrix}
  b_{01} & b_{01} \\
  b_{10} & b_{11} \\
  b_{20} & b_{21}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
  a_{00}b_{00} + a_{01}b_{10} + a_{02}b_{20} & a_{00}b_{01} + a_{01}b_{11} + a_{02}b_{21} \\
  a_{10}b_{00} + a_{11}b_{10} + a_{12}b_{20} & a_{10}b_{01} + a_{11}b_{11} + a_{12}b_{21}
\end{bmatrix}
\]

- **Example**: 

\[
\begin{bmatrix}
  0 & 1 \\
  1 & 1 \\
  0 & 0
\end{bmatrix} \times \begin{bmatrix}
  1 & 0 & 1 \\
  1 & 1 & 1
\end{bmatrix} = \begin{bmatrix}
  1 & 1 & 1 \\
  2 & 1 & 2 \\
  0 & 0 & 0
\end{bmatrix}
\]

\(3\times2 \quad 2\times3 \quad 3\times3\)
Tensor Product

Tensor Product $\otimes: m \times n \otimes k \times l$ results in $mk \times nl$ matrix

$$A = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \quad B = \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix}$$

$$A \otimes B = \begin{bmatrix} a_{00} \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} & a_{01} \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} \\ a_{10} \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} & a_{11} \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} a_{00}b_{00} & a_{00}b_{01} & a_{00}b_{02} & a_{01}b_{00} & a_{01}b_{01} & a_{01}b_{02} \\ a_{00}b_{10} & a_{00}b_{11} & a_{00}b_{12} & a_{01}b_{10} & a_{01}b_{11} & a_{01}b_{12} \\ a_{00}b_{20} & a_{00}b_{21} & a_{00}b_{22} & a_{01}b_{20} & a_{01}b_{21} & a_{01}b_{22} \\ a_{10}b_{00} & a_{10}b_{01} & a_{10}b_{02} & a_{11}b_{00} & a_{11}b_{01} & a_{11}b_{02} \\ a_{10}b_{10} & a_{10}b_{11} & a_{10}b_{12} & a_{11}b_{10} & a_{11}b_{11} & a_{11}b_{12} \\ a_{10}b_{20} & a_{10}b_{21} & a_{10}b_{22} & a_{11}b_{20} & a_{11}b_{21} & a_{11}b_{22} \end{bmatrix}$$
Tensor Product (Cont)

Example 1:

\[
\begin{bmatrix}
a_{00} \\
a_{10} \\
a_{20}
\end{bmatrix} \otimes \begin{bmatrix}
b_{00} \\
b_{10}
\end{bmatrix} = \begin{bmatrix}
a_{00} & b_{00} \\
a_{00} & b_{10} \\
a_{10} & b_{00} \\
a_{10} & b_{10} \\
a_{20} & b_{00} \\
a_{20} & b_{10}
\end{bmatrix} = \begin{bmatrix}
a_{00}b_{00} \\
a_{00}b_{10} \\
a_{10}b_{00} \\
a_{10}b_{10} \\
a_{20}b_{00} \\
a_{20}b_{10}
\end{bmatrix}
\]

\[3 \times 1 \quad 2 \times 1 \quad 6 \times 1\]

Example 2:

\[
\begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix} \otimes \begin{bmatrix}
1 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1
\end{bmatrix}
\]

\[2 \times 2 \quad 1 \times 3 \quad 2 \times 6\]
Multiple Qubits and QuBytes

One Qbit: \[ |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

Two Qbits: 
\[
|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \]

|01\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |01\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \]

|10\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \]

|11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \]

- In a k-qubit register, each of the \(2^k\) positions can be any complex number
- QuByte=8-Qubits = 256-element vector

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/  ©2019 Raj Jain
Homework 19A

Given two matrices:

\[ A = \begin{bmatrix} 1+i & 1 \\ 1-i & i \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

Compute: \( A \times B, A \otimes B \)

Compute the probabilities of each element of \( A \times B \)
Quantum Gates

1. Quantum NOT Gate
2. Quantum AND Gate
3. Quantum OR Gate
4. Quantum NAND Gate
5. Quantum √NOT Gate
Quantum NOT Gate

- **NOT:**

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\end{bmatrix} \times \begin{bmatrix}
0 \\
1 \\
\end{bmatrix} = \begin{bmatrix}
1 \\
0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 \\
1 & 0 \\
\end{bmatrix} \times \begin{bmatrix}
1 \\
0 \\
\end{bmatrix} = \begin{bmatrix}
? \\
? \\
\end{bmatrix}
\]

\[
NOT \; |1> = |0> \quad NOT \; |0> = |?>
\]

- **Exercise:** Fill in the ?’s
Quantum AND Gate

AND:

\[
\text{AND} = \begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \times \begin{bmatrix}
1 \\
0 \\
0 \\
0
\end{bmatrix} = \begin{bmatrix}
1 \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[
\text{AND} \quad \begin{bmatrix} 00 > \\ 01 > \\ 10 > \\ 11 > \end{bmatrix} = \begin{bmatrix} 0 > \\ 0 > \\ 0 > \\ ? > \end{bmatrix}
\]

Exercise: Fill in the ?’s
Quantum OR Gate

- **OR:**

\[
OR = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} \times
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} =
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} \times
\begin{bmatrix}
0 & 1 & 1 & 1
\end{bmatrix}
\]

- **OR**

\[
|00\rangle |01\rangle |10\rangle |11\rangle =
|0\rangle |1\rangle |1\rangle |1\rangle
\]
Quantum NAND Gate

- **NAND**: 

\[
\begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix} \times \begin{bmatrix}
1 \\
0 \\
0 \\
0
\end{bmatrix} \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}
\]

NAND \[ |00> |01> |10> |11> = |1> |1> |1> |0> \]
Quantum $\sqrt{\text{NOT}}$ Gate

- **$\sqrt{\text{NOT}}$:** $\sqrt{\text{NOT}} \times \sqrt{\text{NOT}} = \text{NOT}

$$\sqrt{\text{NOT}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$|1\rangle$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$|0\rangle$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \times \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$|1\rangle$
Controlled NOT Gate

- **CNOT**: If the control bit is 0, no change to the 2\textsuperscript{nd} bit. If control bit is 1, the 2\textsuperscript{nd} bit is complemented.

<table>
<thead>
<tr>
<th>CNOT</th>
<th>(00)</th>
<th>(01)</th>
<th>(10)</th>
<th>(11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(00)</td>
<td>1 0 0 0</td>
<td>0 1 0 0</td>
<td>0 0 0 1</td>
<td>0 0 1 0</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\times
\begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\]

- Controlled NOT gate can be used to produce two bits that are **entangled** ⇒ Two bits behave similarly even if far apart ⇒ Can be used for teleportation of information
# Quantum Gates: Summary

- The first 4 gates above are similar to the classical gates. The last two are non-classical gate.
- There are many other classical/non-classical quantum gates, e.g., Rotate, Copy, Read, Write, …
- Using such gates one can design **quantum circuits**

<table>
<thead>
<tr>
<th>Gate</th>
<th>Classical</th>
<th>Non-Classical</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT</td>
<td>[\begin{bmatrix} 0 &amp; 1 \ 1 &amp; 0 \end{bmatrix}]</td>
<td>[\sqrt{\text{NOT}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 &amp; -1 \ 1 &amp; 1 \end{bmatrix}]</td>
</tr>
<tr>
<td>AND</td>
<td>[\begin{bmatrix} 1 &amp; 1 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \ 0 &amp; 0 &amp; 1 &amp; 0 \end{bmatrix}]</td>
</tr>
<tr>
<td>OR</td>
<td>[\begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 1 &amp; 1 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 0 &amp; 0 &amp; 0 &amp; 1 \ 1 &amp; 1 &amp; 1 &amp; 0 \end{bmatrix}]</td>
</tr>
<tr>
<td>NAND</td>
<td>[\begin{bmatrix} 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}]</td>
</tr>
<tr>
<td>CNOT</td>
<td>[\begin{bmatrix} 0 &amp; 0 &amp; 0 &amp; 1 \ 1 &amp; 1 &amp; 1 &amp; 0 \end{bmatrix}]</td>
<td>[\begin{bmatrix} 1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \ 0 &amp; 0 &amp; 1 &amp; 0 \end{bmatrix}]</td>
</tr>
</tbody>
</table>

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/  ©2019 Raj Jain
Quantum Applications

- It has been shown that quantum computation makes several problems easy that are hard currently. Including:
  - Fourier Transforms
  - Factoring large numbers
  - Error correction
  - Searching a large unordered list

- There are some new methods:
  - Quantum Key Exchange
  - Quantum Teleportation (transfer states from one location to another)

Quantum-Safe Cryptography is being standardized
Fourier Transforms

Time Domain

- $A \sin(2\pi ft)$
- $(A/3) \sin(6\pi ft)$
- $A\sin(2\pi ft) + (1/3) \sin(6\pi ft)$

Frequency Domain

- Amplitude
  - $A$
  - $A/3$

- Frequency
  - $f$
  - $3f$
Quantum Fourier Transform (QFT)

- Fourier transform is used to find periodic components of signals
- Conventional computing requires $O(n2^n)$ gates, $n = \# \text{ of bits in the input register} = \text{Size of input numbers}$  
  $\Rightarrow$ Exponential in $n$
- Quantum computing allows Fourier transforms using $O(m^2)$ quantum gates, $m = \# \text{ of qubits in the q-registers}$  
  $\Rightarrow$ Polynomial in $m$
- QFT is faster than classical FT for large inputs
GCD

- Greatest Common Divisor of any two numbers
  - Divide the larger number with the smaller number and get the remainder less than the divisor
  - Divide the previous divisor with the remainder
  - Continue this until the remainder is zero.
    The last divisor is the GCD

\[
\begin{array}{c}
15) & 35 & \text{(2)} \\
\hline
30 & 05) & 15 & \text{(3)} \\
\hline
0 & & 15 \\
\hline
& & 0
\end{array}
\]

gcd
Shor’s Factoring Algorithm

- Peter Shor used QFT and showed that Quantum Computers can find prime factors of large numbers exponentially faster than conventional computers

- **Step 1:** Find the period of \( a^i \mod N \) sequence.
  Here \( a \) is co-prime to \( N \) \( \Rightarrow \) \( a \) is a prime such that \( \gcd(a, N) = 1 \)
  \( \Rightarrow \) \( a \) and \( N \) have no common factors.

  - Example: \( N=15 \), \( a=2 \);
    \[ 2^i \mod 15 \text{ for } i=0, 1, 2, \ldots \]
    \[ = 1, 2, 4, 8, 1, \ldots \Rightarrow p=4 \]

  - This is the classical method for finding period.
    QFT makes it fast.

- **Step 2:** Prime factors of \( N \) might be \( \gcd(N, a^{p/2}+1) \) and \( \gcd(N, a^{p/2}-1) \)

  - Example: \( \gcd(15, 2^2-1) = 3; \ \gcd(15, 2^2+1) = 5; \)
Homework 19B

- Find factors of 35 using Shor’s algorithm. Show all steps.
- Optional: Try factoring 407 (Answer: $11 \times 37$)
Quantum Machine Learning (QML)

- Quantum for solving systems of linear equation
- Quantum Principal Component Analysis
- Quantum Support Vector Machines (QSVM)
  - Classical SVM has runtime of $O(\text{poly}(m,n))$, $m$ data points, $n$ features
  - QSVM has runtime of $O(\log(mn))$
    - Currently limited to data that can be represented with small number of qubits
- QML can process data directly from Quantum sensors with full range of quantum information

Building Quantum Computers

1. **Neural Atom**: Group of cesium or rubidium atoms are cooled down to a few degree Kelvin and controlled using lasers

2. **Nuclear Magnetic Resonance (NMR)**

3. **Nitrogen-Vacancy Center-in-Diamond**: Some carbon atoms in diamond lattice are replaced by nitrogen atoms

4. **Photonics**: Mirrors, beam splitters, and phase shifters are used to control photons

5. **Spin Qubits**: Using semiconductor materials

6. **Topological Quantum Computing**: Uses Anyon which are quasi-particles different from photons or electrons

7. **Superconducting Qubits**: Requires cooling down to 10mK

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/
Quantum Hardware

- IBM Q Experience: 5-Qubit quantum processor
  Open to public for experiments using their cloud,

Washington University in St. Louis http://www.cse.wustl.edu/~jain/
Quantum Hardware (Cont)

- Google’s Quantum computer in Santa Barbara Lab

Ref: https://www.nbcnews.com/mach/science/google-claims-quantum-computing-breakthrough-ibm-pushes-back-ncna1070461

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/
Quantum Simulators

- QCEngine: [https://oreilly-qc.github.io/](https://oreilly-qc.github.io/)
- Qiskit, [https://qiskit.org/](https://qiskit.org/)
  - Qiskit OpenQASM (Quantum Assembly Language), [https://github.com/QISKit/openqasm/blob/master/examples/generic/adder.qasm](https://github.com/QISKit/openqasm/blob/master/examples/generic/adder.qasm)
- Q# (Qsharp), [https://docs.microsoft.com/en-gb/quantum/?view=qsharp-preview](https://docs.microsoft.com/en-gb/quantum/?view=qsharp-preview)
- Forest, [https://www.rigetti.com/forest](https://www.rigetti.com/forest)
- List of QC Simulators, [https://quantiki.org/wiki/list-qc-simulators](https://quantiki.org/wiki/list-qc-simulators)
- See the complete list at: [https://en.wikipedia.org/wiki/Quantum_programming](https://en.wikipedia.org/wiki/Quantum_programming)

Washington University in St. Louis [http://www.cse.wustl.edu/~jain/](http://www.cse.wustl.edu/~jain/)
Quantum Supremacy

- Quantum Supremacy: Solve a problem on quantum computer that can not be solved on a classical computer
- Google announced it has achieved Quantum Supremacy on October 23, 2019
  - Google built a 54-qubit quantum computer using programmable superconducting processor
- Vendors: IBM, Microsoft, Google, Alibaba Cloud, D-Wave Systems, 1QBit, QC Ware, QinetiQ, Rigetti Computing, Zapata Computing
- Global Competition: China, Japan, USA, EU are also competing

Summary

1. Qubits are two element vectors. Each element is a complex number that indicate the probability of that level.
2. Multi-qubits are represented by tensor products of single-qubits.
3. Qubit operations are mostly matrix operations. The number of possible operations is much larger than the classic computing.
4. Shor’s factorization algorithm is an example of algorithms that can be done in significantly less time than in classic computing.
5. Quantum computing is here. IBM, Microsoft, Google all offer platforms that can be used to write simple quantum computing programs and familiarize yourself.
6. Quantum-Safe Crypto is in standardization.
Reading List

References

- Quantum Algorithm Zoo, (Compiled list of Quantum algorithms), http://quantumalgorithmzoo.org/
Wikipedia Links

- https://en.wikipedia.org/wiki/Bra%E2%80%93ket_notation
- https://en.wikipedia.org/wiki/Complex_number
- https://en.wikipedia.org/wiki/Controlled_NOT_gate
- https://en.wikipedia.org/wiki/Greatest_common_divisor
- https://en.wikipedia.org/wiki/Matrix_multiplication
- https://en.wikipedia.org/wiki/Polar_coordinate_system
- https://en.wikipedia.org/wiki/Quantum_error_correction
Wikipedia Links (Cont)

- https://en.wikipedia.org/wiki/Shor%27s_algorithm
Classic Papers on Quantum Computing


Classic Papers (Cont)

Related Modules

CSE567M: Computer Systems Analysis (Spring 2013),
https://www.youtube.com/playlist?list=PLjGG94etKypJEKjNAa1n_1X0bWWNyZcof

CSE473S: Introduction to Computer Networks (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypJWOSPMh8Azcg5y5e_10TiDw

Wireless and Mobile Networking (Spring 2016),
https://www.youtube.com/playlist?list=PLjGG94etKypKeboNzyN9tSs_HCd5c4wXF

CSE571S: Network Security (Fall 2011),
https://www.youtube.com/playlist?list=PLjGG94etKypKvzfVtutHcPFJXumyyg93u

Video Podcasts of Prof. Raj Jain's Lectures,
https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQM-S-8NUw