Quantum computing and quantum communication

Rakesh P. Tiwari

rakesh.tiwari@unibas.ch

November 30, 2016

UNI BASEL

What will we learn ?

- elements of quantum information
  - qubits
  - superposition and entanglement
  - 1- and 2-qubit gates
  - no-cloning theorem
  - Deutsch algorithm
- error correction, encryption, teleportation
- "hardware" for quantum computers

references:

N.D. Mermin, Quantum computer science, Cambridge University Press

M.A. Nielsen and I.L. Chuang, Quantum computation and quantum information, Cambridge University Press

Lecture notes by C. Bruder

What are quantum bits ?

- A classical computer manipulates bits: possible states 0 or 1
- A quantum computer manipulates qubits $\equiv$ quantum 2-level systems: possible states $(\alpha|0\rangle + \beta|1\rangle)$
- α, β are complex numbers with $|\alpha|^2 + |\beta|^2 = 1$.

Reminder

- operators, e.g., Hamiltonian operator, act on states
- Schrödinger equation: $H|\psi\rangle = E|\psi\rangle$
- states can be written as linear combination of basis states $|\psi\rangle = \sum_n \alpha_n |n\rangle$
- example: spin $\frac{1}{2}$; each state may be expressed as linear combination of $|\uparrow\rangle$ and $|\downarrow\rangle$

Examples of 2-level systems

- all 2-level systems are mathematically equivalent!
- example: spin $\frac{1}{2}$
- physical state $|\uparrow\rangle \rightarrow$ logical state $|0\rangle$
- physical state $|\downarrow\rangle \rightarrow$ logical state $|1\rangle$
- In the basis of eigenstates of $\hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- All operators acting on one qubit are 2×2 matrices

2-qubit states

- 2 qubits $\Rightarrow$ 4 basis states
- $|0\rangle_1|0\rangle_2$
- $|0\rangle_1|1\rangle_2$
- $|1\rangle_1|0\rangle_2$
- $|1\rangle_1|1\rangle_2$
- we omit the indices 1,2 and write $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$
- similarly, we define 3-qubit states, 4-qubit states, ... N-qubit states

Entanglement I

- Apart from the possibility to form superpositions of states, there is another crucial additional resource in a quantum computer: entanglement
- Classical 2-bit state can be ‘factorized’
- Example: state (11)
- Bit 1 is in state “1”, bit 2 is in state “1”

Entanglement II

- In contrast, the entangled 2-qubit state $\left[\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right]$ cannot be factorized
- What happens if we measure qubit 1 and qubit 2?
- Corresponds to measuring the operator $\hat{\sigma}_z$

Entanglement III

- EITHER we get 0 for qubit 1 and 0 for qubit 2 (probability $\frac{1}{2}$)
- OR we get 1 for qubit 1 and 1 for qubit 2 (probability $\frac{1}{2}$)
- But never any 'mixed' result (regardless in which direction we measure)
- This explains the expression 'cannot be factorized'

Superposition vs. entanglement

- $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ superposition of two 1-qubit states
- $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ entangled superposition of two 2-qubit states
- $\frac{1}{2}(|00\rangle + |10\rangle + |01\rangle + |11\rangle)$
- superposition?
- entangled state ?
- $= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ not-entangled superposition of four 2-qubit states

1-qubit gates

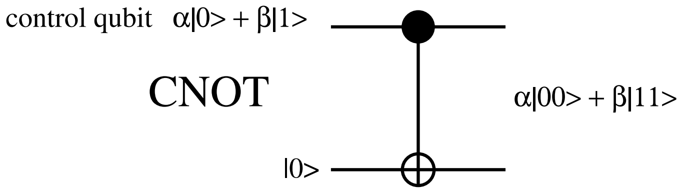
- Example: NOT gate $\hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $\hat{\sigma}_x|0\rangle = \hat{\sigma}_x \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$
- And vice versa
  $\Rightarrow \hat{\sigma}_x$ is the NOT gate
- General 1-qubit gate: unitary 2×2 matrix
- Reminder: A unitary means $AA^\dagger = 1$

Hadamard gate

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

2-qubit gates: CNOT

- 2-qubit gates, e.g., controlled-NOT
- Basis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$
- $\mathrm{CNOT} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}$
- second qubit is flipped if the first one (control qubit) is 1
- $|00\rangle \rightarrow |00\rangle$; $|01\rangle \rightarrow |01\rangle$; $|10\rangle \rightarrow |11\rangle$; $|11\rangle \rightarrow |10\rangle$

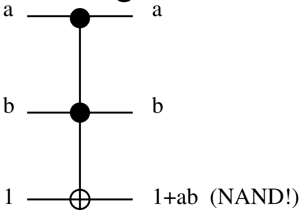


- $$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & {\color{red}1} \\ & & {\color{red}1} & \end{bmatrix} \begin{bmatrix} \alpha \\ 0 \\ \beta \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ 0 \\ \beta \end{bmatrix}$$

Toffoli gate

- 3-qubit gate, basis $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, $|111\rangle$
- $$\text{Toffoli} := \mathcal{T} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & & {\color{red}1} \\ & & & & & & {\color{red}1} & \end{bmatrix}$$
- Third bit (*target*) is flipped if the first two (*control*) bits are 1

Toffoli gate



input			output		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

- (a, b, c)$\rightarrow$ (a,b,c$\oplus$ab) $\rightarrow$ (a,b,c)
- Reversible gate, its inverse is itself
- Simulates classical NAND gate

Can we simulate classical logic circuit using quantum circuit ?

- Of course (world around us is quantum !!)
- All unitary quantum logic gates are inherently *reversible* [each output corresponds to unique input]
- Classical logic gates, such as NAND is inherently irreversible
- All classical logic gates can be assembled from only binary NAND gates
- $\Rightarrow$ using Toffoli gate any classical algorithm can be executed on a quantum computer
- Universal quantum computer needs the CNOT, H, phase gate, $\pi/8$ gate

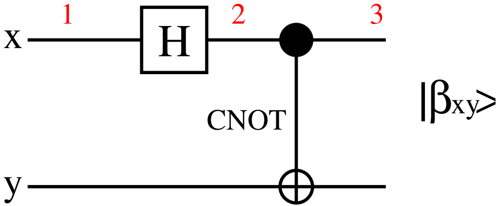
No-cloning theorem

- Copying a state is impossible (no-cloning theorem); however, recreating a state in one location is possible at the expense of destroying it in another (teleportation)
- Assuming there is a "cloning operator" A: $A|\alpha\rangle|0\rangle = |\alpha\rangle|\alpha\rangle$ for any $|\alpha\rangle$
- Now take $|\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- Hence $A|\alpha\rangle|0\rangle = \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)$
- On the other hand, because of linearity, $A\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(A|0\rangle|0\rangle + A|1\rangle|0\rangle)$
- $A\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$
- CONTRADICTION!

Examples: Bell states - circuit to create the Bell states

- $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- $|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$
- $|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$
- $|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
- general expression: $\boxed{|\beta_{xy}\rangle = \frac{1}{\sqrt{2}}(|0y\rangle + (-1)^x|1\bar{y}\rangle)}$

Examples: Bell states - circuit to create the Bell states

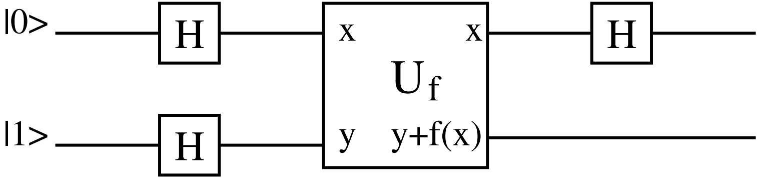


- **1**: input state $|xy\rangle = |00\rangle$
- **2**: $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ (Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$)
- **3**: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \beta_{00}$

Deutsch's algorithm I

- $f(x) : \{0,1\} \rightarrow \{0,1\}$ classical function
- $U_f : |x, y\rangle \rightarrow |x, y + f(x)\rangle$ quantum circuit that implements $y + f(x)$
- input $x = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $y = |0\rangle$ leads to $[\frac{|0,f(0)\rangle + |1,f(1)\rangle}{\sqrt{2}}]$
- $\Rightarrow$ one "application" of f results in both $f(0)$, $f(1)$!
- However...measurement of the final state gives either $|0, f(0)\rangle$ or $|1, f(1)\rangle$
- so, quantum parallelism does not help ...?

Deutsch's algorithm II



- results in $|f(0) \oplus f(1)\rangle[\frac{|0\rangle - |1\rangle}{\sqrt{2}}]$
- $\Rightarrow$ global property of f, namely $f(0) \oplus f(1)$, using only one evaluation of $f(x)$!
- impossible on a classical computer

The power of quantum computing

- Computation = unitary time evolution of a system of qubits generated by a suitable Hamiltonian
- Hamiltonian acts on superposition of entangled input states $\Rightarrow$ high degree of parallelism
- Quantum computer can factorize N-digit numbers in a time that grows polynomially with N using Shor algorithm
- Classical computer: presumably exponentially!