Introduction

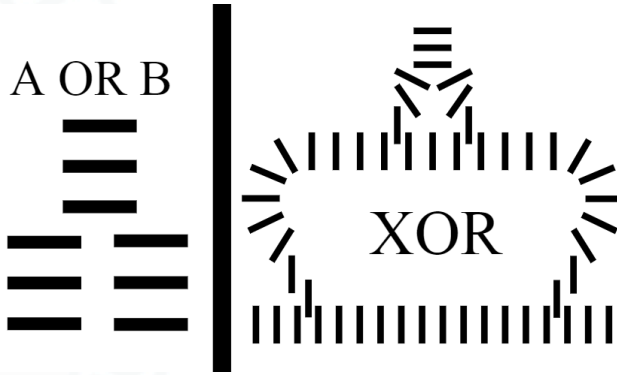
Dr. Fayyaz ul Amir Afsar Minhas
PIEAS Biomedical Informatics Research Lab
Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad, Pakistan
http://faculty.pieas.edu.pk/fayyaz/

Linear Algebra Basics

- Numbers
- Complex Numbers
  - Conjugate (lit. related but opposite in some way)
  - Polar and Cartesian Representations (Magnitude and Phase, real and imaginary components)
  - Complex number arithmetic (addition, multiplication)
- Vectors
  - Basis
  - Vector Space
  - Orthonormal basis $\langle a, b \rangle = 0, \|a\| = \|b\| = 1$
  - Linear Combination
  - Vector (addition, multiplication: dot product, cross product)
  - Transpose (lit. exchange places)
  - Complex vector spaces (Hermitian = conjugate transpose)
  - Transposition
  - Inner product spaces: In which inner products are defined $\langle a, a \rangle \geq 0, \langle a, b \rangle = \overline{\langle b, a \rangle}, \langle a, b + c \rangle = \langle a, b \rangle + \langle b, c \rangle$
- Matrices
  - Viewing matrices as operators on vectors
  - Matrix inverse
  - Matrix determinant (determines if matrix transform is invertible)
  - Eigen values and Eigen Vectors
- Probability
  - Probability of an event is a non-negative number less than or equal to one
  - The sum of all possibilities must always sum to one
  - If two events are independent, then the probability that one will occur or the other will occur will be the sum of the two probabilities and the probability that both will occur will be the product of their probabilities

Unconventional Computing

- Mechanical computing
- Physics
  - Optical, Spintronics, Atomtronics, Fluidics, Quantum Computing
- Chemistry
  - Molecular Computing
- Biochemistry
  - Peptide Computing, DNA Computing
- Biological computing
  - Neuroscience, Bioinspired computing
- Mathematic approaches
  - Analog computing, Reversible Computing, Stochastic Computing



What is a quantum computer?

- What is computing and what is computable?
  - Limits of computation
- Difference between quantum and classical computing
  - Quantum computers exploit quantum phenomenon (superposition, tunneling, entanglement, etc.) for computation
  - It's not a change of medium of computation
    - Gates in vacuum tubes
    - Gates in transistors
    - Gates in Ics (silicon)
    - Gates in biological systems
  - Quantum computers can solve problems that classical ones cannot or they can do them more efficiently
  - 1980: Paul Benioff, Yuri Manin, Richard Feynman, David Deutsch
- Why don't we have quantum computers
  - Sensitivity of quantum phenomenon to external factors
  - The equipment must be tolerant to such issues
  - Error correction is required
  - We do have (limited and rapidly growing) quantum compute capability

What do we have?

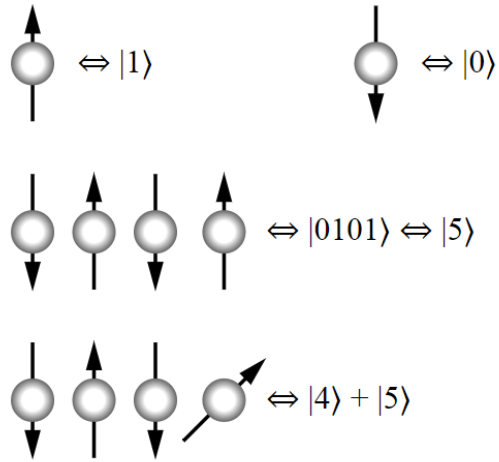
- IBM Quantum Experience
- D-Wave Systems
- Microsoft
- Rigetti Computing

Quantum Supremacy

- Solve problems that traditional computers cannot
  - Integer Factorization
  - Simulation of quantum many-body systems
  - Search Problems
- Classical computers can solve these problems with infinite resources
  - Impractical

Bits or Qbits?

- Classical computing is based on binary digits (bits)
  - A bit can be 1 or 0
- Quantum computing bits are called Qbits
  - They can be 1 or 0
  - Or a super-position of these
  - Two qubits can be in a superposition of 4 states
  - A quantum computer with n qbits can be in an superposition of upto 2^n different states
  - Fundamentally different
    - To characterize the s
    - Representing the state of an n-qubit system on a classical computer requires the storage of 2^n complex numbers



qubits can be in a superposition of all the clasically allowed states

A word of caution!

- Don't think that a n-qubit system can hold 2^n classical bits of information
  - It is important to remember that a quantum computer of n-qubits is in a probabilistic superposition of all 2^n states
  - When a final measurement is made, they will only be found in one of the possible configurations they were in before the measurement
- Also don't think that an n-qbit system is in a single state prior to measurement because of superposition

Another caveat!

- Quantum computing is inherently probabilistic
- Multiple measurements are required to determine the output

- The objective of quantum programming is to design the operations on the computer in a way that gives the required output with high probability!

Quantum Computing Operations

- Quantum computing operations are unitary matrices on vectors
  - Essentially rotations
  - Reversible
- Generalization of classical computation
- Measurement causes the superposition to collapse to one state

Applications

- Cryptography
  - Shor's algorithm can crack prime factors
    - RSA, Diffie-Hellman and Elliptic curve Diffie-Hellman could be broken
    - There are others that cannot be broken (https://en.wikipedia.org/wiki/Post-quantum_cryptography)
- Quantum Search
  - Grover's algorithm can search using quadratically fewer queries to the database than are required by classical algorithms

Applications

- Quantum Simulation
- Quantum Annealing and Adiabatic Optimization
- Solving Linear Equations (HHL Algorithm)

Developments

- Quantum Computing Models
  - Quantum gate array
  - One-way quantum computer
  - Adiabatic quantum computer
  - Topological quantum computer
- Physical realization
  - Superconducting quantum computing
  - Trapped ion quantum computer
  - Optical Lattices
  - Quantum Dot
  - …

Timeline

- 1959: “There is plenty of room at the bottom”
  - Richard Feynmann
- 1981: "Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.“
- 1984: BB84 quantum crypto protocol
- 1993: Quantum teleportation achieved
- 1994: Shor’s algorithm
- 2001: 15 = 5x3 on a 7-qubit NMR computer
- 2011: D-wave quantum anneler
- 2011: 143 factored using 4-qubits
- 2012: 1QBit - first company to focus exclusively on commercializing software applications for commercially available quantum computers
- 2013: Quantum AI Lab by Google
- 2014: Snowden “Penetrating Hard Targets” at NSA $79.7M
- 2016: IBM Quantum Experience Launched
- 2016: Reprogrammable quantum computer
- 2017: IBM Q – first initiative to build commercially available universal quantum computing system

Quantum Complexity Theory

- Problems that can be efficiently solved by quantum computers is called BQP "Bounded Error, Quantum, Polynomial time"
- Although quantum computers may be faster than classical computers for some problem types, those described above cannot solve any problem that classical computers cannot already solve.
  - Undecidable problem: Halting problem

How is computation modeled?

- How can we describe the process of computation?
  - Using logic gates and Boolean algebra
  - Based on the model we do not care how the gate is implemented and we can reason about the structure and make programs for classical computing
- Alternative modeling mechanism
  - Using linear algebra models
    - Can model both classical computing and quantum computing
    - We will discuss models of quantum computing

Unit of computation

- Classical bits (Cbits)
  - Can be 1 or 0: this is the state of the cbit
  - Realized by a physical state: Switch On or Switch Off
- Quantum bits (Qbits)

Dirac Notation

- Let's imagine that bits live in little boxes
  - For example: $|0\rangle$, $|1\rangle$
    - So two bits can have the following values
      - $|0\rangle|0\rangle$, $|0\rangle$ $|1\rangle$, $|1\rangle|0\rangle$,$|1\rangle|1\rangle$ or $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ or $|0\rangle$, $|1\rangle$, $|2\rangle$, $|3\rangle$
    - But why bother writing it this way
      - Difference between state and physical realization, e.g., the state $|3\rangle$ of our 2-cbit system is actually realized possibly by two switches
      - Enables writing products (which will become apparent later)
    - Bra-ket notation was introduced by Dirac as a way of representing vectors (or "kets")

Cbits as vectors

- Consider a single cbit system: we can represent the states of this system in terms of two orthonormal vectors

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

- How many dimensions are needed for a two-cbit system?
  - Four
  - What will be the 4 dimensional representation of 00, 01, 10, 11?
- What about 3-cbit system?

- Vector representation of a 2-cbit system

$$|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle$$

- Or $|0\rangle|0\rangle,\ |0\rangle|1\rangle,\ |1\rangle|0\rangle,\ |1\rangle|1\rangle$

- The dirac notation is written in terms of multiplication (tensor-product)

Tensors and Tensor Products

- Tensor: is a multidimensional vector or matrix

- Tensor product:
  - Product of all possible components written as a vector
  - $|2\rangle = |1\rangle|0\rangle = |1\rangle \otimes |0\rangle = \binom{0}{1} \otimes \binom{1}{0} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
  - The Dirac notation allows us to directly write multi-qubit systems as tensor products of single (or lower) ones
- The tensor-product structure of n-Cbit states is just what one needs in order for the 2^n-dimensional column vector representing the state $|m\rangle_n$ to have all its entries zero except for a single 1 in the m^{th} position down from the top
- Since it is a mean of representation, we will "forget" about column vectors and deal directly with deal directly with their abstract form, e.g., $|2\rangle = |1\rangle|0\rangle = |1\rangle \otimes |0\rangle$

Computation by operation on cbits

- Operations
  - Single bit operations
  - IDENTITY, NOT, SET-TO-ZERO, SET-TO-ONE, MEASURE
    - IDENTITY: $1 \longrightarrow 1, 0 \longrightarrow 0$
    - ERASE: $1 \longrightarrow 0, 0 \longrightarrow 0$
    - SET-TO-ONE: $1 \longrightarrow 1, 0 \longrightarrow 1$
    - NOT: $1 \longrightarrow 0, 0 \longrightarrow 1$
- Reversible Operations
  - Operations that, if performed in reverse, give you the input that produced that result
    - Which operations are reversible?
      - IDENTITY and NOT

Reversible Ops

- The only non-reversible operation in quantum computing is MEASURE
  - We typically do not think of finding the value of a cbit (measurement) as an operation
  - But in quantum computing, it is an operation that maps a qubit to zero or one and is thus non-reversible
  - This will become apparent later.

Computing operations as matrix operations

- We have seen that cbits can be represented as multi-dimensional vectors
- We will now view operations as matrix operations
  - For example, the not operation or X can be written as

$$\mathbf{X} : |x\rangle \rightarrow |\tilde{x}\rangle; \quad \tilde{1} = 0, \quad \tilde{0} = 1.$$

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad X|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

Reversible Multi-cbit operations

- What are different "reversible" multi-cbit operations?
  - AND?
  - SWAP $\mathbf{S}_{10}|xy\rangle = |yx\rangle$
    - $S_{10}|2\rangle = S_{10}|10\rangle = S_{10}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}\begin{pmatrix}0\\0\\1\\0\end{pmatrix} =$

      $\begin{pmatrix}0\\1\\0\\0\end{pmatrix} = |01\rangle = |1\rangle$
    - Prove that this operation is reversible

Controlled-NOT (CNOT)

- The operator is denoted by C_{ij}
  - The i corresponds to the the ith bit and is called the control bit and always remains unchanged
  - The j corresponds to the the jth bit and is called the target bit
    - If the control bit is 0, then the target bit is passed unchanged
    - If the control bit is 1, then the target bit is "inverted"
  - Example:
    - $C_{10}|0\rangle = C_{10}|00\rangle = |00\rangle = |0\rangle$
    - $C_{10}|1\rangle = C_{10}|01\rangle = |01\rangle = |1\rangle$
    - $C_{10}|2\rangle = C_{10}|10\rangle = |11\rangle = |3\rangle$
    - $C_{10}|3\rangle = C_{10}|11\rangle = |10\rangle = |2\rangle$

$$\mathbf{C}_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Controlled-NOT (CNOT)

- $C_{01}|0\rangle = ?$
- $C_{01}|1\rangle = ?$
- $C_{01}|2\rangle = ?$
- $C_{01}|3\rangle = ?$

$$\mathbf{C}_{01} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

– We can write these operations as with $\oplus$ denoting XOR (or sum modulo 2)

$$\mathbf{C}_{10}|x\rangle|y\rangle = |x\rangle|y \oplus x\rangle, \qquad \mathbf{C}_{01}|x\rangle|y\rangle = |x \oplus y\rangle|y\rangle$$

- CNOT allows us to make SWAP as:

$$\mathbf{S}_{ij} = \mathbf{C}_{ij}\mathbf{C}_{ji}\mathbf{C}_{ij}$$

A very common kind of 2-Cbit operator consists of the tensor product $\otimes$ of two 1-Cbit operators:

$$(\mathbf{a} \otimes \mathbf{b})|xy\rangle = (\mathbf{a} \otimes \mathbf{b})|x\rangle \otimes |y\rangle = \mathbf{a}|x\rangle \otimes \mathbf{b}|y\rangle, \tag{1.26}$$

from which it follows that

$$(\mathbf{a} \otimes \mathbf{b})(\mathbf{c} \otimes \mathbf{d}) = (\mathbf{ac}) \otimes (\mathbf{bd}). \tag{1.27}$$

Writing Multibit Operations

- If A is a single bit operator that acts on the j^{th} bit of a multi-bit string then we will denote it as A_j
- Projection Operations
  - Project onto $|1\rangle$ or $|0\rangle$ $\mathbf{n} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \tilde{\mathbf{n}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
  - Properties

$$\mathbf{n}^2 = \mathbf{n}, \qquad \tilde{\mathbf{n}}^2 = \tilde{\mathbf{n}}, \qquad \mathbf{n}\tilde{\mathbf{n}} = \tilde{\mathbf{n}}\mathbf{n} = \mathbf{0}, \qquad \mathbf{n} + \tilde{\mathbf{n}} = \mathbf{1}$$

  - Controlled NOT can be written as

$$\mathbf{C}_{ij} = \tilde{\mathbf{n}}_i + \mathbf{X}_j \mathbf{n}_i$$

Z and H operator

- The Z operator has no equivalent or use in classical computing but it is very important for quantum computing

$$\mathbf{Z} = \tilde{\mathbf{n}} - \mathbf{n} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- Another is the Hadamard Operator

$$\mathbf{H} = \frac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

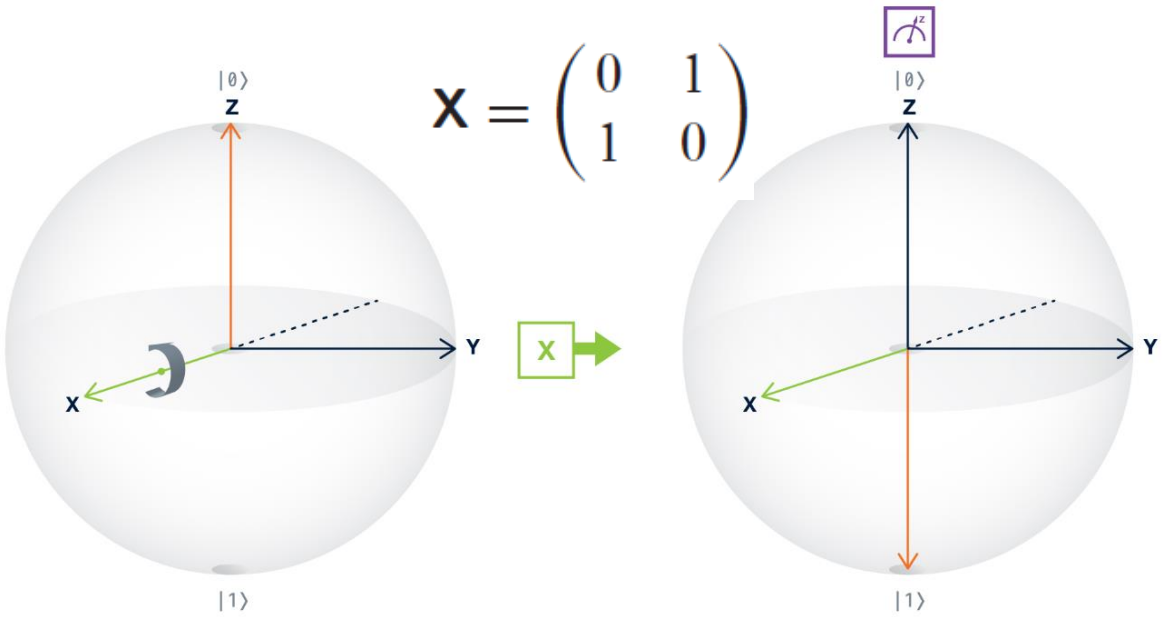
$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad \mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Hadamard Operator

- Note that the Hadamard Operator does not define any useful operation on cbits
  - Why?
    - Because: $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$
      - And we have no physical realization of the state $\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ in a classical 1-bit computer
      - However, this is (easily!) possible in Quantum Computing

The X, Y and Z Gates

$$\mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$



$$\mathbf{Y} = i\mathbf{XZ} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\mathbf{Z} = \tilde{\mathbf{n}} - \mathbf{n} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Quantum Bits vs. Classical Ones

- Before we even begin to describe what quantum bits are
  - We can (begin to) see that the quantum representation is more flexible and powerful
- The Cbit has been represented as a two-dimensional vector but practically only two realizations are possible from the whole space of two dimensional vector
- Qbits do not suffer from this limitation

Qbit

- A Qbit is said to be associated with the state

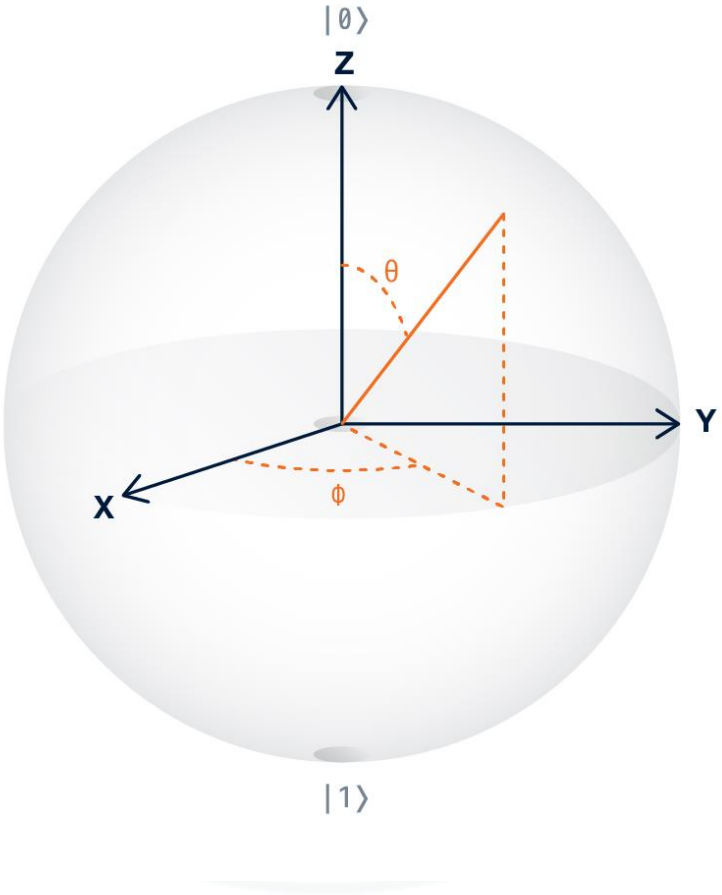
$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \quad |\alpha_0|^2 + |\alpha_1|^2 = 1$$

- The state $|\psi\rangle$ is said to be a *superposition* of the states $|0\rangle$ and $|1\rangle$ with *amplitudes* α_0 and α_1

$$|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$$

Bloch Sphere Representation



Superposition

- Rotation + Translation
  = Rolling

Superposition of almost plane waves (diagonal lines) from a distant source and waves from the wake of the ducks. Linearity holds only approximately in water and only for waves with small amplitudes relative to their wavelengths.

- The state of a pair of two qbits can be written as their tensor product

$$
\begin{aligned}
|\Psi\rangle = |\psi\rangle \otimes |\phi\rangle &= \big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) \otimes \big(\beta_0|0\rangle + \beta_1|1\rangle\big) \\
&= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \\
&= \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}.
\end{aligned}
$$

- The general form of a two-qbit state shown earlier is

$$
|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}
$$

$$
|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1
$$

major difference between qbit & cbit

- The general 2-qbit state is not a product of two 1-qbit states
- Why?
  - Normalization
  - Only possible when $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$
- Individual Qbits making up a multi-Qbit system, in contrast to individual Cbits, cannot always be characterized as having individual states of their own.

Example

- Consider the state of a 2-qbit system $\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ 0 \end{pmatrix}$
- Let's represent the two q-bits as
  - $\binom{a}{b}, \binom{c}{d}$ so if we the state is a tensor product of these states then we require

$$\begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

  - This implies bd = 0 but neither b nor d can be 0 because $ad = \frac{1}{\sqrt{2}}, bc = \frac{-1}{\sqrt{2}}$
  - Thus, the two qbits forming the state are entangled and cannot be taken independently. Such a state is not possible in a 2-cbit system but is possible in a 2-qbit system

How is it possible?

- This is possible if the states aren't pre-written
- Suppose Alice is an observer for system $A \equiv \binom{a}{b}$, and Bob is an observer for system $B \equiv \binom{c}{d}$. If in the entangled state given above Alice makes a measurement, there are two possible outcomes
  - Alice measures 0, i.e., $\binom{a}{b} = \binom{1}{0}$, and the state of the system collapses to $|0\rangle|1\rangle$ with $\binom{c}{d} = \binom{0}{1}$
    - Any subsequent measurement performed by Bob, in the same basis, will always return 1
  - Alice measures 1, i.e., $\binom{a}{b} = \binom{0}{1}$, and the state of the system collapses to $|1\rangle|0\rangle$ with $\binom{c}{d} = \binom{1}{0}$
    - Any subsequent measurement performed by Bob, in the same basis, will always return 0
- Thus, system B has been altered by Alice performing a local measurement on system A. This remains true even if the systems A and B are spatially separated.

Entanglement

- Two qbits are entangled if the state of the two q-bit system cannot be expressed as the product of their individual states
  - It is as if these qbits cannot behave independently anymore and only in relation to each other
  - Spooky action at a distance
    - If two entangled qbits are separated by a large distance, an operation on one will affect the other

https://youtu.be/CC_XES4xQD4

Reversible operations on qbits

- The reversible operations that a quantum computer can perform upon a single Qbit are represented by the action on the state of the Qbit of any *linear* transformation that takes unit vectors into unit vectors.
- Such transformations **u** are called *unitary* and satisfy the condition (norm preserving)

$$\mathbf{u}\mathbf{u}^{\dagger} = \mathbf{u}^{\dagger}\mathbf{u} = 1$$

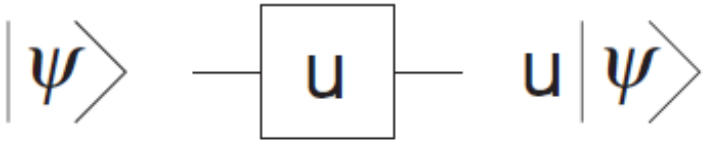


Fig 1.1 A circuit diagram representing the action on a single Qbit of the 1-Qbit gate **u**. Initially the Qbit is described by the input state $|\psi\rangle$ on the left. The thin line (wire) represents the subsequent history of the Qbit. After emerging from the box representing **u**, the Qbit is described on the right by the final state $\mathbf{u}|\psi\rangle$.

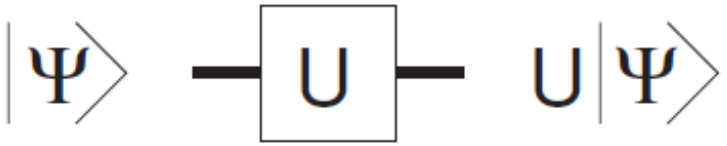
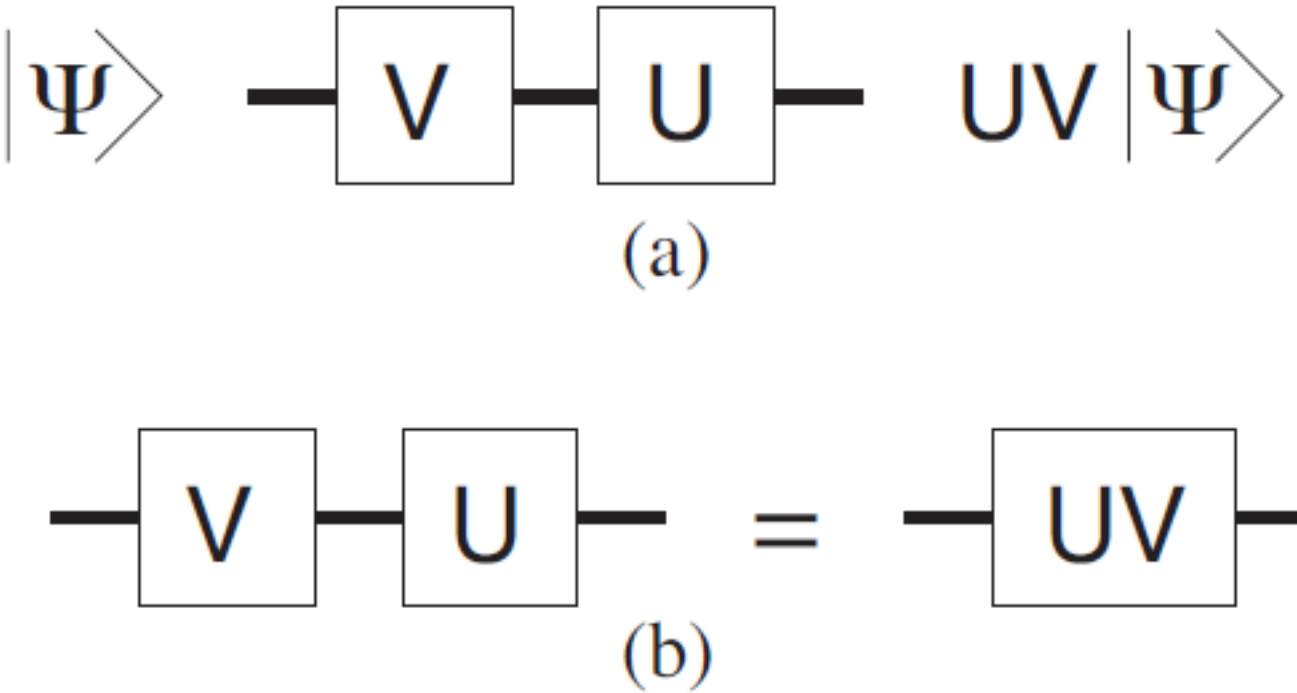


Fig 1.2 A circuit diagram representing the action on n Qbits of the n-Qbit gate **U**. Initially the Qbits ares described by the input state $|\Psi\rangle$ on the left. The thick line (bar) represents the subsequent history of the Qbits. After emerging from the box representing **U**, the Qbits are described on the right by the final state $\mathbf{U}|\Psi\rangle$.

Fig 1.3 (a) A circuit diagram representing the action on n Qbits of two n-Qbit gates. Initially the Qbits are described by the input state $|\Psi\rangle$ on the left. They are acted upon first by the gate **V** and then by the gate **U**, emerging on the right in the final state **UV**$|\Psi\rangle$. Note that the order in which the Qbits encounter unitary gates in the figure is opposite to the order in which the corresponding symbols are written in the symbol for the final state on the right. (b) This emphasizes the unfortunate convention that, because gates on the left act before gates on the right in a circuit diagram, a circuit showing **V** on the left and **U** on the right represents the operation conventionally denoted by **UV**.



Measurement: Born Rule

- The value of each bit in a c-bit system can be simply viewed and precisely determined
- However, for quantum systems things are different and probabilistic
- Given a n-qbit state $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_x |x\rangle$, probability that the zeros and ones resulting from measurements of all the Qbits will give the binary expansion of the integer x is $p(x) = |\alpha_x|^2$
  - The reason we have normalization is to be able to write the probability without normalization
- Example:
  - The probability that the output is 1 or 2 is 50%

$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Three Polarizer Paradox

- The light passing through the first polarizer becomes horizontally polarized and can be represented as: $|\rightarrow\rangle$
- The second polarizer is oriented at 45 degrees so its preferred axis is $|\nearrow\rangle$
- We know that $|\rightarrow\rangle = \frac{1}{\sqrt{2}}|\nearrow\rangle - \frac{1}{\sqrt{2}}|\nwarrow\rangle$
- Thus, the chances that the photon will pass through the second polarizer is ½
- Any photons that have passed through the second polarizer now have polarization $|\nearrow\rangle = \frac{1}{\sqrt{2}}|\rightarrow\rangle + \frac{1}{\sqrt{2}}|\uparrow\rangle$
- Thus, the chances that it passes through the third polarizer with preferred axis $|\uparrow\rangle$ is ½

Measurement

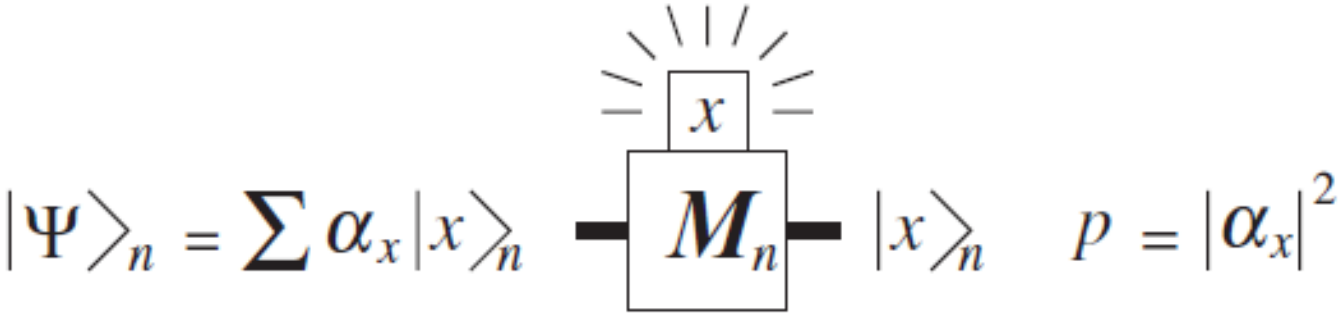


Fig 1.4 A circuit diagram representing an n-Qbit measurement gate. The Qbits are initially described by the n-Qbit state

$$|\Psi\rangle_n = \sum_{0 \le x < 2^n} \alpha_x |x\rangle_n,$$

on the left. After the measurement gate M_n has acted, with probability $p = |\alpha_x|^2$ it indicates an integer x, $0 \le x < 2^n$, and the Qbits are subsequently described by the state $|x_n\rangle$ on the right.

Measurement

- The action of a measurement gate cannot be
- undone:
  - given the final state $|x\rangle$, there is no way of reconstructing the initial state $|\psi\rangle$.
  - Measurement is irreversible.
  - The action of a measurement gate in not linear.
- To the extent that it suggests that some preexisting property is being revealed, “measurement” is a dangerously misleading term
- While measurement in quantum mechanics is not at all like measuring somebody’s weight, it does have some resemblance to measuring Alice’s IQ , which, one can argue, reveals no preexisting numerical property of Alice, but only what happens when she is subjected to an IQ test.
- Measurement of a qbit is simply the output of a measurement gate when a qbit is subjected to it

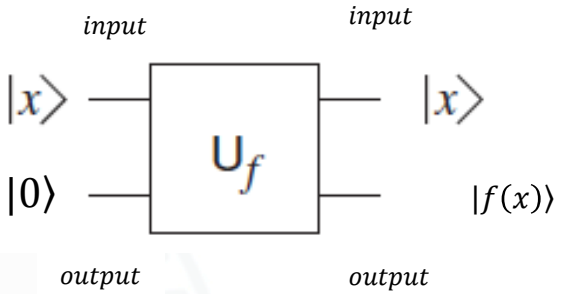
Measurement

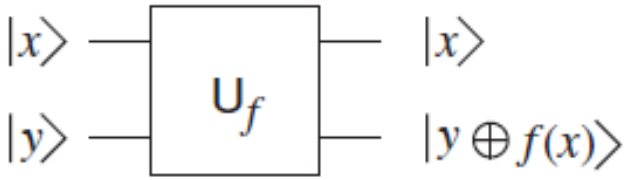
- Once you send *n* Qbits through an *n*-Qbit measurement gate, you remove the possibility of extracting any further information about their original state
- After such a measurement of five Qbits, if the
- result is 01100, then the post-measurement state associated with the Qbits is no longer $|\psi\rangle$, but $|01100\rangle$.
- This change of state attendant upon a measurement is often referred to as a *reduction* or *collapse* of the state.
- It shouldn't be said that if the measurement came out to be 0, then the qbit was in state 0 prior to measurement because it may not have been in state 0

	Cbits	Qbits
States of n Bits	$\|x\rangle_n, \quad 0 \leq x < 2^n$	$\sum \alpha_x \|x\rangle_n, \sum \|\alpha_x\|^2 = 1$
Subsets of n Bits	Always have states	Generally have no states
Reversible operations on states	Permutations	Unitary transformations
Can state be learned from Bits?	Yes	No
To learn state of Bits	Examine them	Go ask Alice
To get information from Bits	Just look at them	Measure them
Information acquired	x	x with probability $\|\alpha_x\|^2$
State after information acquired	Same: still $\|x\rangle$	Different: now $\|x\rangle$

Quantum Computer Model

- Two registers
  - Input register (n-qbit)
  - Output register (m-qbit)
- All computations are reversible (except measurement)
- Apply a function f(x) as a unitary transformation $\mathbf{U}_f$
- Standard reversible quantum computing protocol models computation as





$$\mathbf{U}_f\left(|x\rangle_n|y\rangle_m\right) = |x\rangle_n|y \oplus f(x)\rangle_m$$

Quantum Computer Model

- If the initial output is zero, the output will then end up with f(x) in the output register

$$\mathbf{U}_f\left(|x\rangle_n|y\rangle_m\right) = |x\rangle_n|y \oplus f(x)\rangle_m$$

$$\mathbf{U}_f\left(|x\rangle_n|0\rangle_m\right) = |x\rangle_n|f(x)\rangle_m$$

- It is reversible

$$\mathbf{U}_f\mathbf{U}_f\left(|x\rangle|y\rangle\right) = \mathbf{U}_f\left(|x\rangle|y \oplus f(x)\rangle\right)$$
$$= |x\rangle|y \oplus f(x) \oplus f(x)\rangle = |x\rangle|y\rangle$$

The magic of quantum computing

- Let's apply an H-gate to all n input qbit state $|00\rangle$
  - Recall that $H|0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$$
\begin{aligned}
(\mathbf{H}\otimes\mathbf{H})(|0\rangle\otimes|0\rangle) = \mathbf{H}_1\mathbf{H}_0|0\rangle|0\rangle &= (\mathbf{H}|0\rangle)(\mathbf{H}|0\rangle)\\
&= \tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\\
&= \tfrac{1}{2}(|0\rangle|0\rangle+|0\rangle|1\rangle+|1\rangle|0\rangle+|1\rangle|1\rangle)\\
&= \tfrac{1}{2}(|0\rangle_2+|1\rangle_2+|2\rangle_2+|3\rangle_2).
\end{aligned}
$$

  - This generates a superposition of all possible integers

The magic of quantum computing

- For the n-qbit state $|0\rangle_n$

$$\mathbf{H}^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} |x\rangle_n$$

- So if the initial state of the input register is $|0\rangle_n$ and we apply an *n*-fold Hadamard transformation to that register, its state becomes an equally weighted superposition of all possible *n*-Qbit inputs.

The magic of quantum computing

- If we then apply $\mathbf{U}_f$ to that superposition, with 0 initially in the output register, then due to linearity we get

$$\mathbf{U}_f\left(\mathbf{H}^{\otimes n} \otimes \mathbf{1}_m\right)\left(|0\rangle_n |0\rangle_m\right) = \frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} \mathbf{U}_f\left(|x\rangle_n |0\rangle_m\right)$$

$$= \frac{1}{2^{n/2}} \sum_{0 \le x < 2^n} |x\rangle_n |f(x)\rangle_m .$$

quantum parallelism

- If before letting $\mathbf{U}_f$ act, we merely apply a Hadamard transformation to every Qbit of the input register, initially in the standard state $|0\rangle_n$, the result of the computation is described by a state whose structure cannot be explicitly specified without knowing the result of all 2^n evaluations of the function f .
- So if we have a mere hundred Qbits in the input register, initially all in the state $|0\rangle_{100}$ (and m more in the ouput register), if a hundred Hadamard gates act on the input register before the application of $\mathbf{U}_f$, then the form of the final state contains the results of $2^{100} \approx 10^{30}$ evaluations of the function f .
- A billion billion trillion evaluations! This apparent miracle is called ***quantum parallelism.***

Note

- It is important to note that although all possible inputs have been mapped to all possible outputs (i.e., computation has taken place), we can only measure one input and output according to born rule (and that too at random)
- After the measurement the state of the registers reduces to $|x\rangle_n \, |f(x)\rangle_m$
- So this is no better than a classical computer?

- If this were the full story, nobody but a few philosophers would be interested in quantum computation. The National Security Agency of the United States of America is interested because there are more clever things one can do.
- Typically these involve applying additional unitary gates to one or both of the input and output registers before and/or after applying $\mathbf{U}_f$, sometimes intermingled with intermediate measurement gates acting on subsets of the Qbits.
- All these additional gates are cunningly chosen so that when one finally does measure all the Qbits, one extracts useful information about *relations* between the values of *f* for several different values of *x*, which a classical computer could get only by making several independent evaluations.
- The price one inevitably pays for this relational information is the loss of the possibility of learning the actual value *f* (*x*) for any individual *x*. This tradeoff of one kind of information for another is typical of quantum computation, and typical of quantum physics in general, where it is called the *uncertainty principle*.

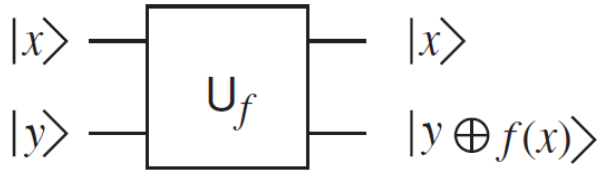
Deutsch Problem

- Deutsch's problem is the simplest example of a quantum tradeoff that sacrifices particular information to acquire relational information.
- Let both input and output registers each contain only one Qbit, so we are exploring functions *f* that take a single bit into a single bit.

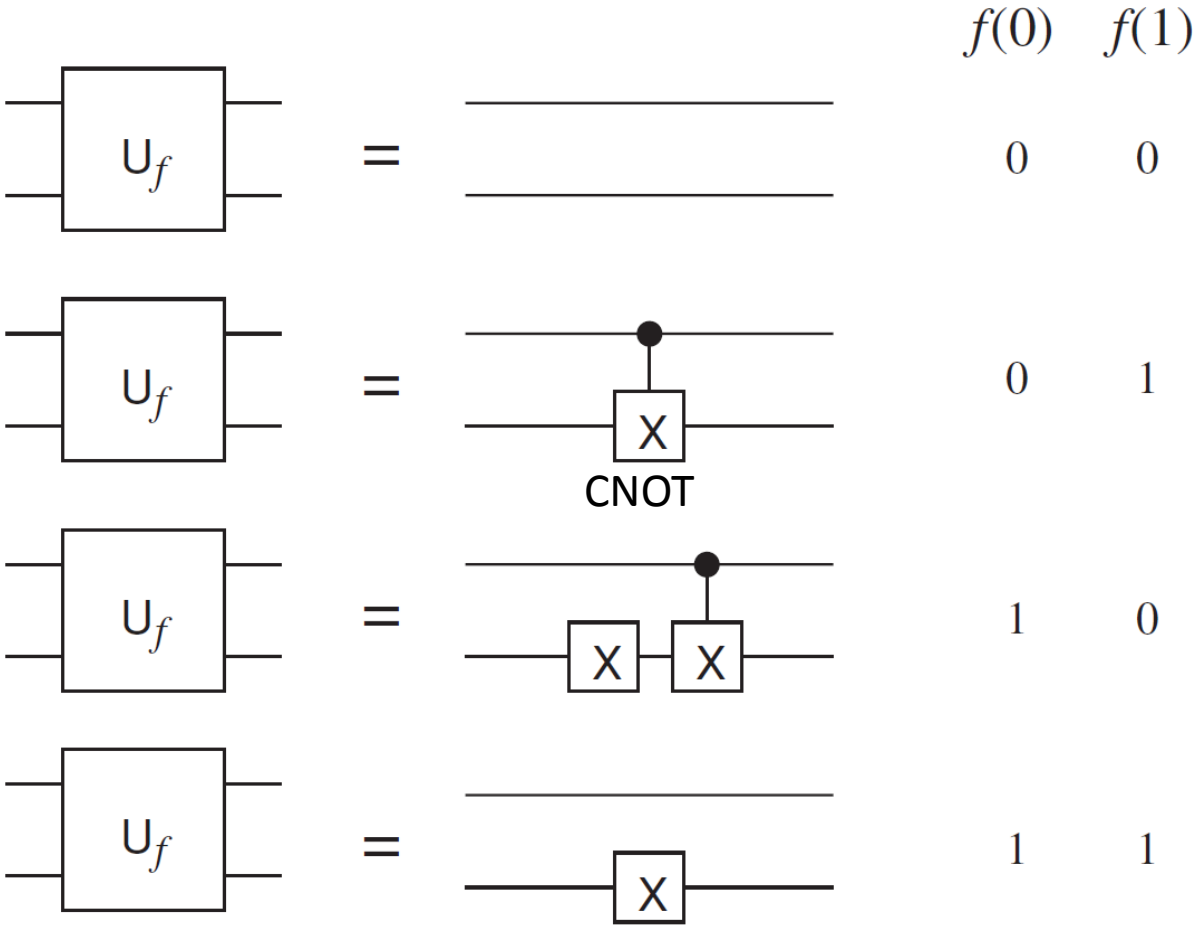
	$x = 0$	$x = 1$
f_0	0	0
f_1	0	1
f_2	1	0
f_3	1	1

Deutsch Problem

- Note
  - Output is zero initially



	$x = 0$	$x = 1$
f_0	0	0
f_1	0	1
f_2	1	0
f_3	1	1

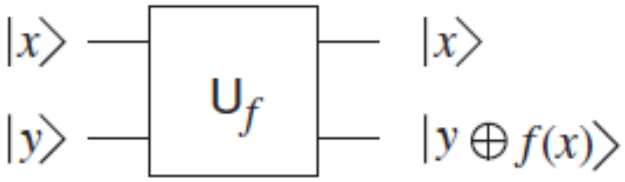


$$\mathbf{U}_{f_0} = \mathbf{1}, \qquad \mathbf{U}_{f_1} = \mathbf{C}_{io}$$
$$\mathbf{U}_{f_2} = \mathbf{C}_{io}\mathbf{X}_o, \qquad \mathbf{U}_{f_3} = \mathbf{X}_o$$

- Suppose that we are given a black box that executes $\mathbf{U}_f$ for one of the four functions, but are not told which of the four operations the box carries out
- How can you find out what is in the black box?
- You can only give a single input to the block
- What information can you get?

- If we choose to learn the value of $f(0)$
  - we can restrict f to being either f_0 or f_1 (if $f(0) = 0$) or to being either f_2 or f_3 (if $f(0) = 1$)
- If we choose to learn the value of $f(1)$
  - we can restrict f to being either f_0 or f_2 (if $f(1) = 0$) or to being either f_1 or f_3 (if $f(1) = 1$)
- Suppose, however, that we want to learn whether f is constant ($f(0) = f(1)$, satisfied by f_0 and f_3) or not constant ($f(0) \neq f(1)$, satisfied by f_1 and f_2)
  - We then have no choice with a classical computer but to evaluate both $f(0)$ and $f(1)$ and compare them.
- However, a quantum computer can do this in a single run!

- If we apply the following operations before and after U_f and measure the input bit afterwards
- Note: We do not know that exact function values



$$
\begin{aligned}
(\mathbf{H}\otimes\mathbf{H})(\mathbf{X}\otimes\mathbf{X})(|0\rangle|0\rangle) &= (\mathbf{H}\otimes\mathbf{H})(|1\rangle|1\rangle) \\
&= \left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right)\left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right) \\
&= \tfrac{1}{2}\left(|0\rangle|0\rangle - |1\rangle|0\rangle - |0\rangle|1\rangle + |1\rangle|1\rangle\right)
\end{aligned}
$$

$$
(\mathbf{H}\otimes 1)\mathbf{U}_f(\mathbf{H}\otimes\mathbf{H})(\mathbf{X}\otimes\mathbf{X})(|0\rangle|0\rangle)
$$

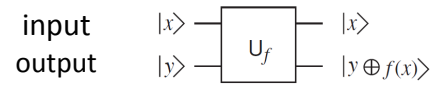
$$
= \begin{cases} |1\rangle\frac{1}{\sqrt{2}}\left(|f(0)\rangle - |\tilde{f}(0)\rangle\right), & f(0) = f(1), \\ |0\rangle\frac{1}{\sqrt{2}}\left(|f(0)\rangle - |\tilde{f}(0)\rangle\right), & f(0) \neq f(1). \end{cases}
$$

Significance?

- How many queries does it take to find the exact function?
  - With classical computing: 2
  - With quantum computing: ?

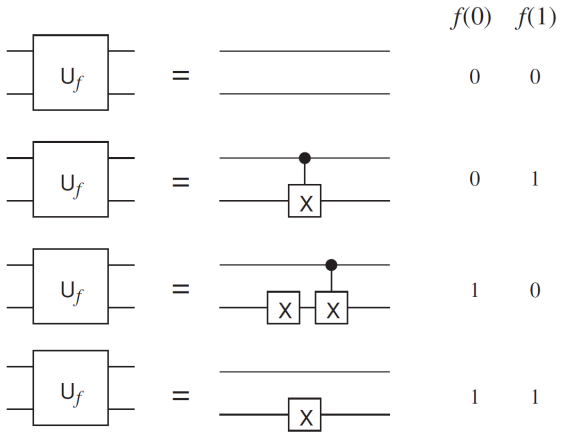
```
// Name of Experiment: Deutsch v1OPENQASM 2.0;
include "qelib1.inc";
qreg q[5];
creg c[5];
x q[0];
x q[1];
h q[0];
h q[1];
cx q[1],q[0];
x q[1];h
q[1];
measure q[1] -> c[1];
measure q[0] -> c[0];
```

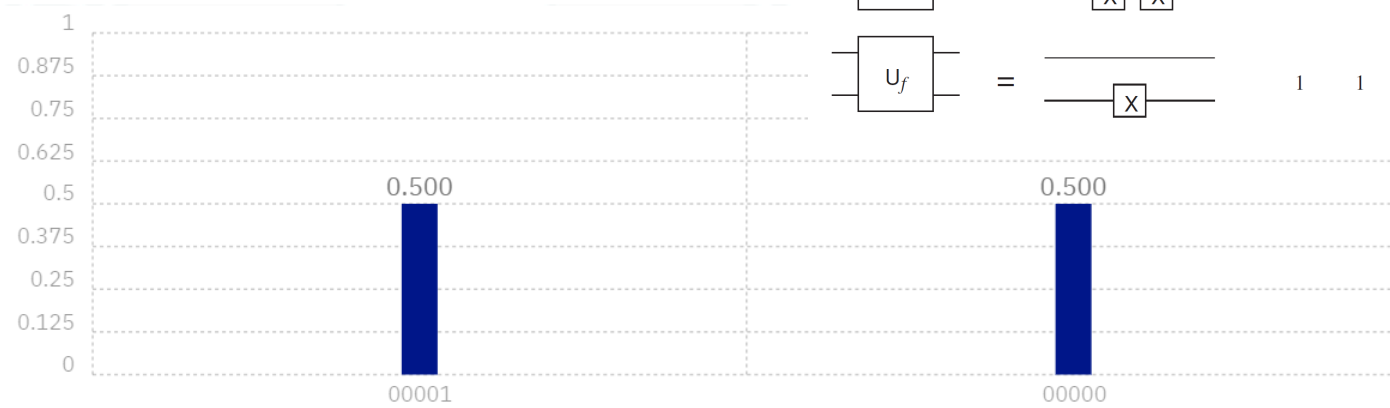
Assignment

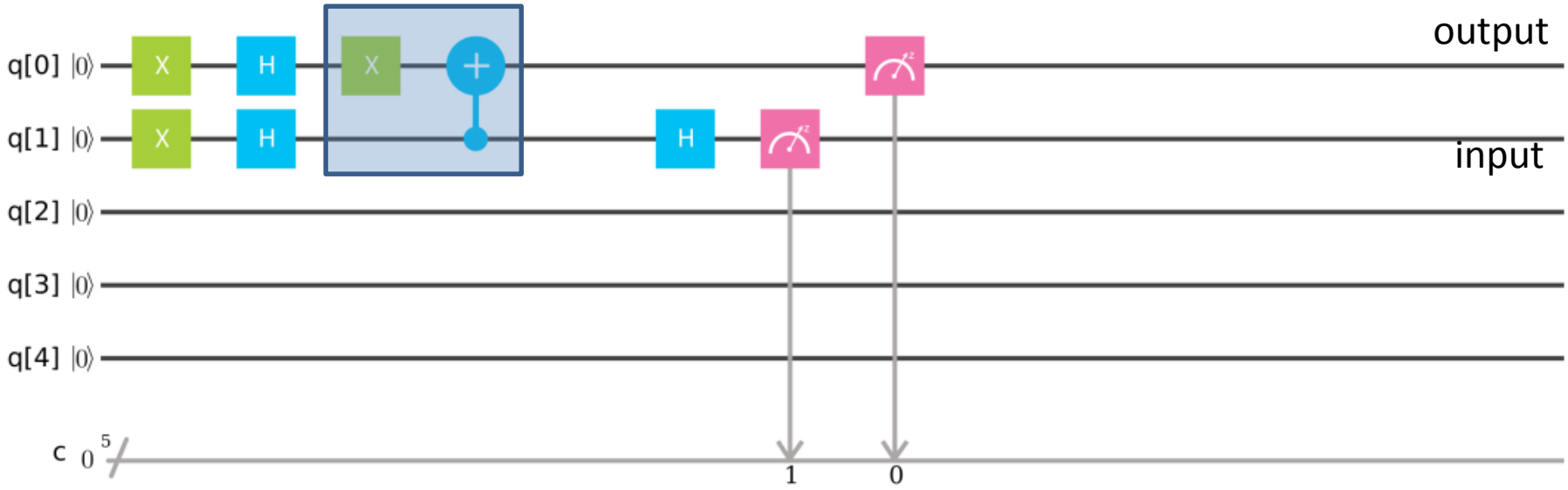


$$(\mathbf{H}\otimes 1)\mathbf{U}_f(\mathbf{H}\otimes\mathbf{H})(\mathbf{X}\otimes\mathbf{X})(|0\rangle|0\rangle)$$

$$= \begin{cases} |1\rangle\frac{1}{\sqrt{2}}\left(|f(0)\rangle - |\tilde{f}(0)\rangle\right), & f(0) = f(1), \\ |0\rangle\frac{1}{\sqrt{2}}\left(|f(0)\rangle - |\tilde{f}(0)\rangle\right), & f(0) \neq f(1). \end{cases}$$







End of Lecture-1

We want to make a machine that will be proud of us.

- Danny Hillis