An introduction to
Quantum computing

- Quantum computing: A brief historical journey
- States, qubits, superposition, entanglement
- Existing packages: Cirq (main), Quiskit, Forest
- Examples

Resources:

1 "Quantum Computation and Quantum Information" 10th Anniversary Edition, by Michael A. Nielsen & Isaac L. Chuang Cambridge University Press.

2 J. D. Hidary "Quantum computing: An applied approach." Springer, 2019

3 Arxiv Article: "Quantum Algorithm Implementations for Beginners", P. J. Coles et al. arXiv:1804.03719v1 [cs.ET] 10-Apr. 2018

4 Austin Gilliam, Charlene Venci, Sreraman Muralidharan, Vitaliy Dorum, Eric May, Rajesh Narasimhan, and Constantin Gonciulea Foundational Patterns for Efficient Quantum Computing

5 Eleanor G. Rieffel, Wolfgang Polak "An Introduction to Quantum Computing for Non-Physicists", arXiv:quant-ph/9809016

Historical perspective

Motivation: Moore's law: harder and harder to gain speed out of traditional computers

➤ The Church-Turing thesis: *Any algorithmic process can be simulated efficiently using a Turing machine.*

➤ However some types of computations may be difficult/ impossible to solve *efficiently* on standard computers ...

➤ ... but can be solved *efficiently* on non-standard computers – e.g. "Analogue computers"

➤ Question: How about trying to exploit properties of the quantum world to solve 'hard problems'?

➤ Question asked by David Deutsch in 1985 - answered the question positively

➤ Breakthrough: Shor's algorithm [1994]: demonstration of how to find prime factors of large integers – main ingredient of encryption

➤ Currently: Huge regain of interest from governments and private sector

➤ Note: IBM has an experimental quantum computer ('Q' computer, 53 qubits) as does Google ('Sycamore' also 53 qubits),

➤ Caveat emptor: No one knows if QC will succeed in becoming general purpose platforms that will eventually replace current computers..

A few nanoseconds worth of quantum mechanics

- *At the end of the 19th century it was discovered that classical mechanics does not provide an accurate picture of the microscopic world. A few discoveries made in those days set in motion one of the most important and fascinating chapters of physics. See: "30 years that shook physics" - by George Gamov, Dover for an interesting account.*

➤ The quantum world is very different from classical one. Can be counter-intuitive.

➤ If one observes a quantum object it looks like a particle, but when it is not being observed it behaves like a wave.

➤ Wave-particle duality $\longrightarrow$ many interesting physical phenomena.

➤ Example: quantum objects can exist in multiple states at once. Superposition of these objects interfere like waves to define a quantum state. The main property that gives quantum computing its power: *superposition of states*

Superposition

" Imagine a pot with water in it. When you have water in a pot with a top on it, you don't know if it's boiling or not. Real water is either boiling or not; looking at it doesn't change its state. But if the pot was in the quantum realm, the water (representing a quantum particle) could both be boiling and not boiling at the same time or any linear superposition of these two states. If you took the lid off of that quantum pot, the water would immediately be one state or the other. The measurement forces the quantum particle (or water) into a specific observable state."

➤ The state of a quantum-mechanical system is described by a wavefunction ψ - a function of the coordinates of each particle.. This function is a solution of the Schrödinger equation.

➤ The wavefunction ψ lies in a complex Hilbert space [think of this $\mathbb{C}^n$ where $n = \infty$]

➤ The wavefunction ψ is a linear combination of some orthonormal basis functions (e.g. the eigenstates of the Hamiltonian)

Schrödinger equation

$$i\hbar\frac{\partial\Psi}{\partial t} = H\Psi$$

➤ The Hamiltonian in its original form is very complex:

$$H = -\frac{h^2}{2m}\sum_i \nabla^2_{\vec{r}_i} + \sum_{i,j}\frac{e^2}{|\vec{r}_i - \vec{r}_j|^2} - \sum_i\sum_k \frac{Z_k e^2}{|\vec{r}_i - \vec{R}_k|^2}$$

$$-\frac{h^2}{2M}\sum_k \nabla^2_{\vec{R}_k} + \sum_{k,l}\frac{e^2}{|\vec{R}_k - \vec{R}_l|^2}$$

➤ Involves sums over all electrons / nuclei and their pairs in terms involving Laplaceans, distances betweens electrons /nuclei.

➤ When we observe the state we see only one component. If we repeat the experiment we may observe another state.. But the states appear with probabilities given by the amplitudes = | coefficients | squared.

➤ Two or more quantum states in a system can be strongly linked: measurement of one dictates the possible measurement outcomes for another – regardless of the distance between the two objects.

➤ The property underlying this phenomenon is known as entanglement and it at the core of the huge potential power of QC.

Entanglement

Two qubits are entangled if they cannot act independently from one another: They are 100% correlated. This situation is physical: the counter-intuitive fact is that the correlation persists even when the particles are physically far apart from each other.

Q: How does QC work?

Answer: one can design quantum circuits that can be manipulated with, e.g., energy fields – You design the circuit [this is like coding in classical computing] - then the hardware will run the circuit and you observe some output.. need to repeat and average. [one observation by itself is useless]

Quantum computing: Notation

- Linear algebra notation: $\psi = a_0\psi_0 + a_1\psi_1 + \cdots + a_j\psi_j + \cdots$
- Quantum mechanics notation: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle + \cdots + a_j|j\rangle + \cdots$

➤ Think of $|\psi\rangle$ as the column vector $\longrightarrow$ $\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_j \\ \vdots \end{pmatrix}$

➤ Then $\langle\psi|$ will be the transpose conjugate of this vector

➤ $\langle u|v\rangle$ is the (complex) inner product of u and v - (a scalar).

➤ ... $|u\rangle\langle v|$ is the 'outer product' of u and v – a matrix (uv^H in standard LA notation)

➤ $|\psi|^2$ represents a probability. Its integral over space is 1, i.e., $\langle\psi|\psi\rangle = 1$

➤ The energy of a system is governed by a Hamiltonian

$$E(\psi) = \langle\psi|H|\psi\rangle$$

➤ Ground state: Minimum energy (i.e., ψ minimizes $E(\psi)$)

➤ This leads to an eigenvalue problem: (time-independent Schrödinger equation)

$$H\Psi = E\Psi$$

➤ Feynman suggested to use a quantum-mechanical system to actually compute the wavefunction

L. K. Glover *Perhaps the most surprising thing about quantum computing is that it was so slow to get started. Physicists have known since the 1920s that the world of subatomic particles is a realm apart, but it took computer scientists another half-century to begin wondering whether quantum effects might be harnessed for computation. The answer was far from obvious.*

Early work:

➤ Charles Bennetts [physicist, IBM Watson]

➤ Paul Benioff [Physicist, Argonne Nat. lab]

➤ Richard Feynman [Physicist, Caltech]

bits and qubits

➤ Standard computers use bits. A bit can take the value 0 or 1.

➤ A quantum bit or 'qubit' stores a combination of zero and one. Its state is represented by

$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle$$

where a_0, a_1 are complex and

$$|a_0|^2 + |a_1|^2 = 1$$

➤ Difference with classical computing: if we 'observe' state $|\psi\rangle$ we will see either $|0\rangle$ (probability $|a_0|^2$) or $|1\rangle$ (probability $|a_1|^2$)

The Bloch sphere

* State of a single qubit: $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$

* a_1, a_2 are complex. So in principle we would need 4 real variables

* Also recall that we must have $|a_0|^2 + |a_1|^2 = 1$

* First consider *real* combinations of the two base states. Write in the form:

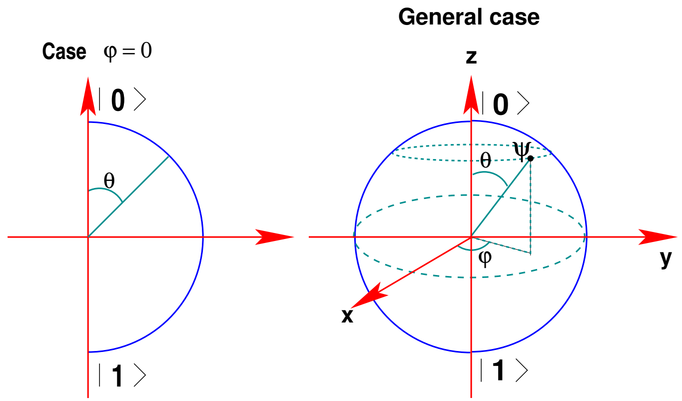
$$\cos\left(\frac{\theta}{2}\right)\ |0\rangle + \sin\left(\frac{\theta}{2}\right)\ |1\rangle$$

* Note: for $\theta = 0$ we get $|0\rangle$ and for $\theta = \pi$ we get $|1\rangle$

* Add complex phase to the 2nd term (only) [keeping a_0 real]:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)\ |0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)\ |1\rangle$$

➤ A qubit state can be represented on a so-called Bloch Sphere.



Note $\quad 0 \le \theta \le \pi \qquad 0 \le \varphi < 2\pi$

✍1 How did we manage to use a sphere (3 parameters) in 3 dimensions while we started off with 4 (real) parameters?

- Answer : we sacrified one phase because it made no difference - normally:

$$|\psi\rangle = e^{i\alpha_0}\cos\left(\frac{\theta}{2}\right)\ |0\rangle + e^{i\alpha_1}\sin\left(\frac{\theta}{2}\right)\ |1\rangle$$
$$= e^{i\alpha_0}\left[\cos\left(\frac{\theta}{2}\right)\ |0\rangle + e^{i(\alpha_1-\alpha_0)}\sin\left(\frac{\theta}{2}\right)\ |1\rangle\right]$$

The factor $e^{i\alpha_0}$ makes no physical difference (all that matters is the 2-norm of $\binom{a_0}{a_1}$ which is the same). So we can set it to 1 to make a_0 real. Then we set $\varphi = \alpha_1 - \alpha_0$ and discard the first phase term.

✍2 What are all 6 states that correspond to the 6 points where the sphere touches the 3 axes (x, y, z axes). [Hint: 2 of these are obvious. For the others determine θ and φ....]

✍3 Take a state represented in the form $\begin{pmatrix}\cos(\theta/2)\\ \sin(\theta/2)e^{i\varphi}\end{pmatrix}$. What are the values of x, y, and z on the sphere?

One-qubit Quantum operators

- Operators that act on one qubit in a certain state (to produce one qubit in a certain state)
- Each operartor is a mapping from $\textbf{span}\{|0\rangle, |1\rangle\}$ to itself
- We use the basis: $\{|0\rangle, |1\rangle\}$.
- In this basis $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$.
- With this: Each operator can be viewed as a mapping from $\mathbb{C}^2$ to itself $\longrightarrow$ Can be expressed as a 2×2 matrix.

• Note: Each of them is unitary [in particular it preserve length]

✍04 Why is this property required?

- Next w'll see a few of the most important ones

The NOT operator ['Pauli-X' operator]

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

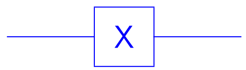
If we apply X to the state $|0\rangle$ we get

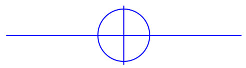
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

- Note: for $j \in \{0, 1\}$ we have:

$$X|j\rangle = |j \oplus 1\rangle$$

where $\oplus$ is the exclusive or.

DIAGRAM: 

or 

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{or} \quad |0\rangle \longrightarrow |1\rangle$$

✍05 What does this operation do to a point on the Bloch sphere?

Sol: The phase φ makes no difference. Assume it is 0.

$$\begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2) \end{pmatrix} \longrightarrow \begin{pmatrix} \sin(\theta/2) \\ \cos(\theta/2) \end{pmatrix} = \begin{pmatrix} \cos(\frac{\pi-\theta}{2}) \\ \sin(\frac{\pi-\theta}{2}) \end{pmatrix}$$

- Verification : when applied to $|+\rangle = \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$ you get the same result. The point is invariant - as expected.
- $\theta \to \pi - \theta \quad \to$: *Symmetry about the x, y plane.*

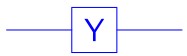
✍06 What about the general cases when $\varphi \neq 0$?

The Y operator

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

Example:

$$Y|j\rangle = (-1)^j i|1 \oplus j\rangle$$

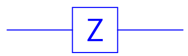
DIAGRAM: 

The Z operator

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

Example:

$$Z|j\rangle = (-1)^j|j\rangle$$

DIAGRAM 

The R_φ operator

$$R_\varphi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$$

Example:

$$R_\varphi |1\rangle = e^{i\varphi}|1\rangle$$

DIAGRAM: —[R_φ]—

R_φ = phase shift op.

- Two particular cases:

$\varphi = \pi/2$ $\rightarrow$ S operator

rotates state by $\frac{\pi}{2}$ around z-axis

$\varphi = \pi/4$ $\rightarrow$ T operator

rotates state by $\frac{\pi}{4}$ around z-axis

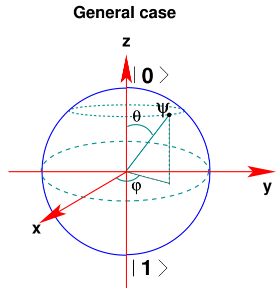
Note that $S = T^2$

➤ Alternative – and equivalent on the Boch sphere – to R_φ is:

$$R_z(\varphi) = \begin{pmatrix} e^{-i\frac{\varphi}{2}} & 0 \\ 0 & e^{i\frac{\varphi}{2}} \end{pmatrix}$$

✍07 Explain why on Bloch sphere, R_φ is equivalent to $R_z(\varphi)$

➤ Take a look at the Bloch sphere:



$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix} \begin{pmatrix} \cos(\theta/2) \\ e^{i\varphi_0}\sin(\theta/2) \end{pmatrix} = \begin{pmatrix} \cos(\theta/2) \\ e^{i(\varphi_0+\varphi)}\sin(\theta/2) \end{pmatrix}$$

➤ Rotation of angle φ around z axis.

➤ The other two rotations $R_x(\theta)$ and $R_y(\theta)$ of angle θ around the x and y axes respectively are:

✍08 Bloch sphere: What actions do you get when $\varphi = 0$?

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

➤ Note: It can be shown that

$$R_x(\theta) = \exp\left(-i\frac{\theta}{2}X\right)$$

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right)$$

$$R_z(\theta) = \exp\left(-i\frac{\theta}{2}Z\right)$$

The Hadamard operator

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Example:

$$H|0\rangle = \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$$

DIAGRAM: —[H]—

Properties

✍09 HXH =?

✍10 $HZH = X$

✍11 HYH =?

✍12 H^{-1} =?

✍13 H^2 =?

✍14 S^2 =?

➤ Later, we will exploit the relation $HZH = X$

➤ The Hadamard gate plays a very important role in QC.

✍15 Visualize the effect of the H gate on the Bloch sphere

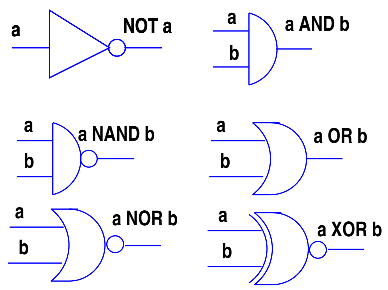
Note:

$\alpha|0\rangle + \beta|1\rangle$ —[X]— $\beta|0\rangle + \alpha|1\rangle$

$\alpha|0\rangle + \beta|1\rangle$ —[Z]— $\alpha|0\rangle - \beta|1\rangle$

$\alpha|0\rangle + \beta|1\rangle$ —[H]— $\frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$

➤ Classical setting: a gate acts on 1 bit (e.g., the NOT gate) or 2 bits (e.g., the AND gate) to yield one bit.
➤ Question: can we represent all the QC single qubit gates from combining a few basic ones?



Gates: Universality

Recall: In classical setting, only *one* gate is needed to implement any function of bits - the NAND gate

a	b	a AND b	a NAND b
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Quantum setting: Any n-qubit gate can be made from 2-qubit gates. Specifically: Any multiple qubit logic gate may be composed from CNOT and single qubit gates.

➤ This is because: Any unitary $n \times n$ can be decomposed as a product of 2-level unitary matrices, i.e., unitary matrices that act only on two-or-fewer vector components.

[essentially: rotations, and complex scalings]

Two qubits

➤ Let q_0, q_1 be two qubits.

➤ $|ij\rangle$ means: q_0 is in state $|i\rangle$ and q_1 is in state $|j\rangle$

➤ A 2-qubit register is a combination of 4 states

$$|\psi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle$$

➤ The space of these 4 states is $\mathbb{C}^2 \otimes \mathbb{C}^2$

➤ $|ij\rangle$ also represents: $|i\rangle \otimes |j\rangle$. We will often just write $|i\rangle|j\rangle$

➤ If $f = \alpha|0\rangle + \beta|1\rangle$ and $g = \gamma|0\rangle + \delta|1\rangle$, what is $|f\rangle \otimes |g\rangle$?

➤ In what follows e_1, e_2, e_3, e_4 are the 4 canonical basis vectors of $\mathbb{C}^4$,i.e., the 4 columns of the identity matrix.

➤ By convention the basis of the resulting space is

$$|\psi_1\rangle = |00\rangle \quad = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \quad = e_1$$

$$|\psi_2\rangle = |01\rangle, \quad = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \quad = e_2$$

$$|\psi_3\rangle = |10\rangle, \quad = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \quad = e_3$$

$$|\psi_4\rangle = |11\rangle, \quad = \begin{pmatrix}0\\1\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} \quad = e_4$$

So for example $|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$, and $|01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$.

Entanglement: An example

Case 1: $|\psi\rangle = |00\rangle$ Measuring $|\psi\rangle$ we will find with 100% probability that the first qubit q_0 is $|0\rangle$ and similarly that q_1 is $|0\rangle$.

Case 2: $|\psi\rangle = \frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$

* 50% chance of observing $|00\rangle$ and 50% chance of observing $|11\rangle$

* However, if we measure q_0 and find that $q_0 = |0\rangle$ then we know that the outcome must be $|00\rangle$ therefore $q_1 = |0\rangle$ also

* If we measure q_0 and find that $q_0 = |1\rangle$ then we know that the outcome must be $|11\rangle$ therefore $q_1 = |1\rangle$ also

* In case 2, the two qubits are 100% correlated. They are entangled

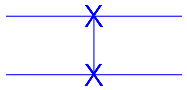
A few important binary operators

➤ Input: 2 qubits – out 2 qubits

SWAP

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

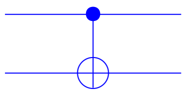
DIAGRAM:



CNOT

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

DIAGRAM:



➤ CNOT stands for controled not. Very important in quantum logic

➤ First input qubit q_0 plays the role of a control qubit.

➤ Second qubit is the target qubit.

➤ On output top qubit remains the same. Lower one is flipped ('Not' applied to it) when (and only when) control bit is $|1\rangle$.

The following exercise will help you understand this

✍16 Determine the output states for each of all 4 possible inputs states. Use the CNOT diagram to illustrate this.

Logical operation of CNOT gate: *if a is in state $|1\rangle$ flip qubit b*

```
a----*----a
     |
b---(+)----b'
```

$\|ab\rangle$	$\|ab'\rangle$
$\|00\rangle$	$\|00\rangle$
$\|01\rangle$	$\|01\rangle$
$\|10\rangle$	$\|11\rangle$
$\|11\rangle$	$\|10\rangle$

✍17

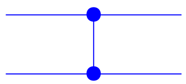
```
0----*----?        0----*----?
     |                  |
0---(+)---?        1---(+)---?

1----*----?        1----*----?
     |                  |
0---(+)---?        1---(+)---?
```

CZ

$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

DIAGRAM:



- Controlled Z operator
- q_0 = control qubit, q_1 = target
- Z operator applied to q_1 iff $q_0 = |1\rangle$

Note: CZ is symmetric, i.e., contol-target roles of q_0, q_1 can be exchanged

The Bell State

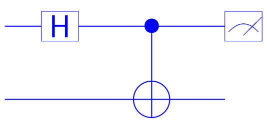
1 Start with $q_0 := |0\rangle$ and $q_1 := |0\rangle$

2 Apply Hadamard to q_0 $\longrightarrow$

$$\boxed{q_0 := H|0\rangle = |+\rangle}$$

3 Apply CNOT gate to q_0 and q_1: the 2 qbits are now entangled

DIAGRAM



- The resulting entangled state is the state $|\psi\rangle = \frac{1}{\sqrt{2}}[|00\rangle + |11\rangle]$ of case 2 seen before. It is called a Bell State. In quantum physics this involves two particles that form a so-called EPR pair. [EPR stands for Einstein, Podolsky and Rosen]

- It is known that Einstein was very skeptical about quantum mechanics ("God does not play dice" he once stated). In a 1935 article, Einstein, Podolsky and Rosen, tried to show that quantum mechanics would lead to a contradiction.. – it was a contradiction to our logic of thinking. But the nano world is different.

L. K. Glover

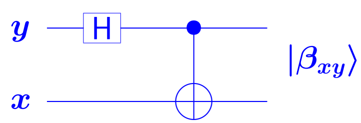
On the power of quantum computing:

Thus from, say, 500 particles you could, in principle, create a quantum system that is a superposition of as many as 2^{500} states. Each state would be a single list of 500 1's and 0's. Any quantum operation on that system-a particular pulse of radio waves, for instance, whose action was, say, to execute a controlled-NOT operation on the 175th and 176th qubits-would simultaneously operate on all 2^{500} states. Hence with one machine cycle, one tick of the computer clock, a quantum operation could compute not just on one machine state, as serial computers do, but on 2^{500} machine states at once! That number, which is approximately equal to a 1 followed by 150 zeros, is far larger than the number of atoms in the known universe. Eventually, of course, observing the system would cause it to collapse into a single quantum state corresponding to a single answer, a single list of 500 1's and 0's – but that answer would have been derived from the massive parallelism of quantum computing.

The 4 Bell States

- In the form of an exercice

✍18 Determine the four possible outputs when the inputs are in the 4 possible base states $|00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$.



- The 4 resulting states are called the 4 Bell states and denoted by $\beta_{00}, \beta_{01}, \beta_{10}, \beta_{11}$, respectively
- These are also called 'EPR pairs' or 'EPR states'

Three qubits

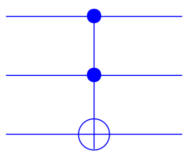
➤ We now have 3 input qubits and 3 ouputs. Operators are 8×8 matrices

➤ State represented by eight vectors $e_1, e_2, \cdots, e_8$

Toffoli

Matrix $= 8 \times 8$ Identity with last 2 columns swapped.

DIAGRAM



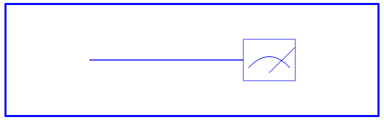
➤ q_0 and q_1 are both control qubits; q_2 = target

➤ NOT operator applied to q_2 iff $q_0 = |1\rangle$ AND $q_1 = |1\rangle$

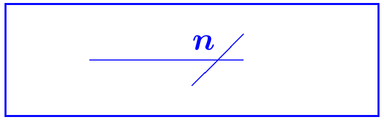
✍19 Determine each of the ouput states for all 8 possible inputs

Other symbols used

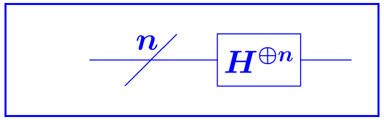
- Measurement symbol



- n qubit inputs

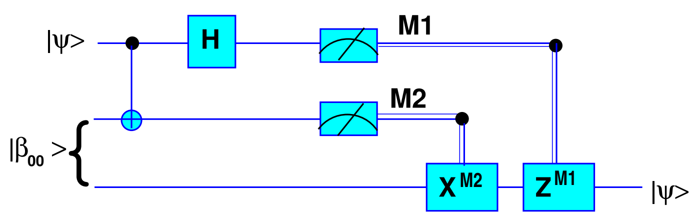


- Apply operator n qubits



Quantum teleportation (outline of an example)

- Bob and Alice now live far apart. When together they generated an EPR pair and each took one qubit of the pair before separating.
- Alice wants to send a qubit $|\psi\rangle$ to Bob by sending *classical information*
- Difficulty: measuring $|\psi\rangle$ not possible [will yield one state]
- Solution: Interact the $|\psi\rangle$ with her half of the EPR state. Measure the 2 qubits. Result one of 00, 01, 10, 11.
- Send this (classical info) info to Bob.
- Bob performs one of 4 operations [depending on what he received from Alice]
- Bob recovers $|\psi\rangle$



➤ Notes: double lines carry classical information. Top 2 lines: Alice, Bottom: Bob.

✍20 Details to be added

Resources: IBM and qiskit

- https://www.research.ibm.com/ibm-q/
- https://www.research.ibm.com/ibm-q/network/
- https://www.research.ibm.com/ibm-q/technology/devices/
- https://www.research.ibm.com/ibm-q/technology/simulator/
- https://qiskit.org/
- https://qiskit.org/aqua
- https://www.research.ibm.com/ibm-q/learn/what-is-quantum-computing/
- https://quantumexperience.ng.bluemix.net/qx/editor

Resources: cirq and Forest

Cirq

- https://github.com/quantumlib/Cirq
- https://cirq.readthedocs.io/en/stable

Forest

- https://github.com/rigetti/pyquil
- pyquil.readthedocs.io/en/latest

see

https://quantum-computing.ibm.com/support

Example: The Deutsch-Jozsa algorithm

➤ One of the first algorithms to demonstrate usefulness of QC

Problem: given a function f from $\{0, 1\}$ to itself determine whether f is a constant function.

➤ The function is constant when $f(x) \equiv 0 \; \forall x$ or $f(x) \equiv 1 \; \forall x$ ($\forall$= for all). It is balanced otherwise.

➤ Here are all possible 2-bit functions:

➤ Constant: f_0, f_1, balanced: f_x, $f_{\bar{x}}$

x	f_0	f_1	f_x	$f_{\bar{x}}$
0	0	1	0	1
1	0	1	1	0

➤ Normally we need 2 evaluations to solve the problem [one eval. = querying one qubit]

➤ Can do it with one - with quantum computing

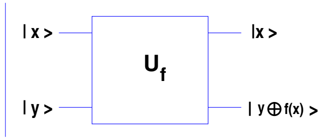
➤ $f : \{0, 1\}^n \rightarrow \{0, 1\}$ would classically need $2^{n-1} + 1$ evals. QC: one

The Deutsch-Jozsa algorithm

➤ First: f is not injective - so cannot tell x from $f(x)$. It is not reversible. Make it reversible with a trick

➤ Define 'Oracle':

$$U_f(|x\rangle|y\rangle) := |x\rangle|y \oplus f(x)\rangle$$



* Note: $\oplus$ == addition mod 2 == XOR

✍1 Show that $U_f \circ U_f = I$ (where: $\circ$ = composition)

➤ From above exercise we see that U_f is now reversible (even though f may not be)

➤ Consider U_f as a function of the 2 qubits x and y

✍2 Show that when $f = f_0$ then U_f is the identity

✍3 Show: when $f = f_1$ then U_f does an XOR on the 2nd qubit

✍D4 When $f = f_x$ then U_f does the CNOT operation:

Case $f = f_x$
Control=x, Target=y

$\|xy\rangle$	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$U_f(\|x\rangle\|y\rangle)$	$\|00\rangle$	$\|01\rangle$	$\|11\rangle$	$\|10\rangle$

✍D5 When $f = f_{\bar{x}}$ then U_f does the operation:

Case $f = f_{\bar{x}}$

$\|xy\rangle$	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$U_f(\|x\rangle\|y\rangle)$	$\|01\rangle$	$\|00\rangle$	$\|10\rangle$	$\|11\rangle$

Note: all second bits are flipped from case f_x above - therefore:

- This is a CNOT operation followed by a NOT (X) on 2nd qubit.

✍D6 Show that for a given f, U_f (a 2 qubit operator) is linear and that it is unitary. What is its matrix representation for each of the 4 functions $f_0, f_1, f_x, f_{\bar{x}}$?

- Deutsch-Jozsa algorithm based on exploiting superposed states
- Take second qubit as $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ and apply oracle.

$$\begin{aligned} U_f|x\rangle|-\rangle &= U_f|x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} \\ &= |x\rangle\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} \\ &= |x\rangle\frac{|f(x)\rangle - |\bar{f}(x)\rangle}{\sqrt{2}} \\ &= (-1)^{f(x)}|x\rangle|-\rangle \end{aligned}$$

- Known as the *phase kick-back trick* – value of the function reflected in phase.

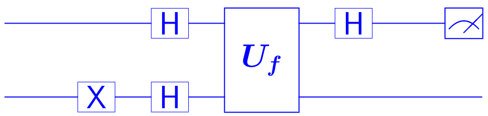
Q: If we observe the first qubit on output: to what operation is the oracle equivalent for $f_0, f_1, f_x, f_{\bar{x}}$?

A:

f_0	f_1	f_x	$f_{\bar{x}}$
I	$-I$	Z	$-Z$

- One more transform: Exploit the relation $HZH = X$. Apply H to x before and after U_f. Let $x = |0\rangle$ (top qubit).

DIAGRAM



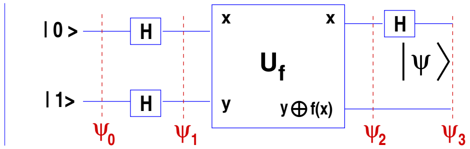
- If f is either f_0 or f_1 we observe $\pm|0\rangle$
- If f is either f_x or $f_{\bar{x}}$ we observe a $\pm|1\rangle$

Done!

- Note: The actual final state has the form (prove it)

$$\psi = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

✍D7 Determine the states $\psi_0, \cdots, \psi_3$ (see figure) after each 'stage'



Partial Solution:

1. $|\psi_0\rangle = |01\rangle$
2. $|\psi_1\rangle = \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$. Write as $|\ x\ \rangle|-\rangle$
3. $|\psi_2\rangle = U_f(|x\rangle, |-\rangle) = (-1)^{f(x)}|x\rangle|-\rangle$
   $= \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|-\rangle$
   If $f(0) = f(1) \rightarrow$ same sign $\psi_2 = \pm|+\rangle|-\rangle$
   Otherwise $\psi_2 = \pm|-\rangle|-\rangle$
4. Apply H to 1st qubit of ψ_2:
   If $f(0) = f(1) \rightarrow \psi_3 = \pm|H+\rangle|-\rangle$ $\boxed{= \pm|0\rangle\ |-\rangle}$
   Otherwise $\psi_3 = \pm|H-\rangle|-\rangle$ $\boxed{= \pm|1\rangle|-\rangle}$

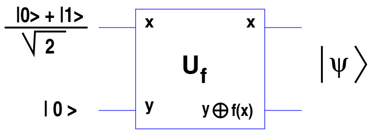
Quantum paralellism

➤ In effect the DJ algorithm is able to evaluate $f(0)$ and $f(1)$ at the same time.

➤ Assume same context: $f : \{0, 1\} \to \{0, 1\}$. Same oracle U.

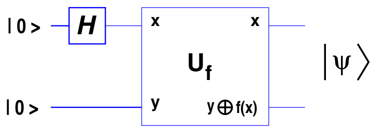
✍08 Consider the circuit to the right. Show that the output is

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle >}{\sqrt{2}}$$



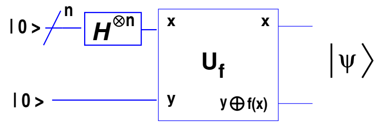
➤ In effect $|\Psi\rangle$ carries information about both $f(0)$ and $f(1)$!

➤ The above circuit is same as:



➤ Generalization to $n + 1$ gates. Function f is now from $\{0, 1\}^n$ to $\{0, 1\}$.

➤ Recall the notation seen earlier: at top we have n qubit at state $|0\rangle$ - each followed by Hadamard.



➤ Output state is now:

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$$

Example: When $n = 2$ – state x input to U_f is

$$x = \frac{1}{2} [|00\rangle + |01\rangle + |10\rangle + |11\rangle]$$

Output: $\frac{1}{2} [|00, f(00)\rangle + |01, f(01)\rangle + |10, f(10)\rangle + |11, f(11)\rangle]$

Cirq codes

Resources:

➤ See https://github.com/quantumlib/cirq

➤ I found a good documentation in https://cirq.readthedocs.io/en/stable/

➤ Also: the the Cirq workshop bootcamp repository (google search it)

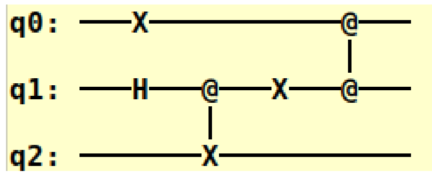
➤ *Cirq* Provides a toolkit (a 'framework') for similating quantum algorithms.

➤ Written in python. Implements all the gates we have seen and more.

➤ The following illustration shows a simple example


```
import cirq
q0 = cirq.NamedQubit("q0")
q1 = cirq.NamedQubit("q1")
q2 = cirq.NamedQubit("q2")
ops = [cirq.X(q0), cirq.H(q1), cirq.CNOT(q1, q2), cirq.X(q1),
  cirq.CZ(q0,q1)]
circuit = cirq.Circuit(*ops)
print(circuit)
```

Output:



A longer example showing many of the gates

```python
import cirq
import numpy as np
q0, q1, q2 = cirq.LineQubit.range(3)
ops = [ cirq.X(q0),
        cirq.Y(q1),
        cirq.Z(q2),
        cirq.CZ(q0,q1),
        cirq.CNOT(q1,q2),
        cirq.H(q0),
        cirq.T(q1),
        cirq.S(q2),
        cirq.CCZ(q0, q1, q2),
        cirq.SWAP(q0, q1),
        cirq.CSWAP(q0, q1, q2),
        cirq.CCX(q0, q1, q2),
        cirq.ISWAP(q0, q1),
        cirq.Rx(0.5 * np.pi)(q0),
        cirq.Ry(.5 * np.pi)(q1),
        cirq.Rz(0.5 * np.pi)(q2),
        (cirq.X**0.5)(q0)]
print(cirq.Circuit(*ops))
print(cirq.unitary(cirq.CNOT))
print(cirq.unitary(cirq.CZ))
```


Output:

```
0: ───X───@───H───────@───×───@───@───iSwap───Rx(0.5π)───X^0.5───
          │           │   │   │   │   │
1: ───Y───@───@───T───@───×───×───@───iSwap───Ry(0.5π)───────────
              │       │       │   │
2: ───Z───────X───S───@───────×───X───Rz(0.5π)───────────────────
[[1.+0.j 0.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [0.+0.j 0.+0.j 0.+0.j 1.+0.j]
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]]
[[ 1.+0.j  0.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  1.+0.j  0.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  1.+0.j  0.+0.j]
 [ 0.+0.j  0.+0.j  0.+0.j -1.+0.j]]
```

A few commands to loot at:

- `cirq.X(q0)` : gate X at q0.
- `cirq.LineQubit.range(p)`: create a line of qubits .. or
- `cirq.GridQubit.range(p,q)` create a grid of qubits ..
- `print(cirq.Circuit(*ops))` prints circuit

Quantum Fourier Transform

- QFT is at the core of the Shor algorithm
- Main idea of QFT: Exploit product decomposition. Recall:

DFT

$x = [x_0, x_1, \cdots, x_{N-1}]^T$ is transformed to y with:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2i\pi jk/N}$$

Therefore:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2i\pi jk/N} |k\rangle \qquad (*)$$

- Suppose that $N = 2^n$. Write any k in its binary representation:

$$k = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n 2^0 = \sum_{l=1}^{n} k_l 2^{n-l}$$

Drop the scaling term $\frac{1}{\sqrt{N}}$ in (*) and set that $N = 2^n$. Then:

$$\begin{aligned}
\sum_{k=0}^{2^n-1} e^{2i\pi jk/2^n} |k\rangle &= \sum_{k=0}^{2^n-1} e^{2i\pi j \sum_{l=1}^{n} k_l 2^{-l}} |k_1 \dots k_n\rangle \\
&= \sum_{k_1=0}^{1} \sum_{k_2=0}^{1} \cdots \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2i\pi j k_l 2^{-l}} |k_l\rangle \\
&= \bigotimes_{l=1}^{n} \left[\sum_{k_l=0}^{1} e^{2i\pi j k_l 2^{-l}} |k_l\rangle \right] \\
&= \bigotimes_{l=1}^{n} \left[|0\rangle + e^{2i\pi j 2^{-l}} |1\rangle \right]
\end{aligned}$$

➤ Write $j = \sum_{m=1}^{n} j_m 2^{n-m}$. Since $e^{2i\pi \times integer} = 1$ then

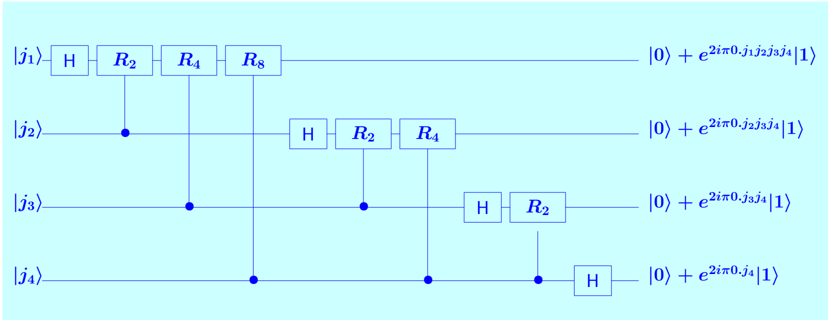
$$\begin{aligned} e^{2i\pi j 2^{-l}} &= e^{2i\pi \sum_{m=1}^{n} j_m 2^{n-m} 2^{-l}} = e^{2i\pi \sum_{m=1}^{n} j_m 2^{n-l-m}} \\ &= e^{2i\pi \sum_{m=n-l+1}^{n} j_m 2^{n-l-m}} \\ &= e^{2i\pi 0.j_{n-l+1} j_{n-l+2} \cdots j_n} \end{aligned}$$

➤ In the end:

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2i\pi jk/2^n} |k\rangle = \frac{\left(|0\rangle + e^{2i\pi 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2i\pi 0.j_{n-1}j_n}|1\rangle\right)\cdots\left(|0\rangle + e^{2i\pi 0.j_1 j_2 \dots j_n}|1\rangle\right)}{2^{n/2}}$$

Let $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix}$

➤ Here is a diagram for a 4-qubit QFT



➤ $O(n^2)$ gates needed for $N = 2^n$ -transform.

➤ Classically: need $O(N \log(N)) = n \times 2^n$ operations.

Concluding notes

L. K. Glover

On the future of QC:

Will quantum computers ever grow into their software? How long will it take them to blossom into the powerful calculating engines that theory predicts they could be? I would not dare to guess, but I advise all would-be forecasters to remember these words, from a discussion of the Electronic Numerical Integrator and Calculator (ENIAC) in the March 1949 issue of Popular Mechanics: Where a calculator on the ENIAC is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and weigh only 1.5 tons.