1.4 Quantum algorithms

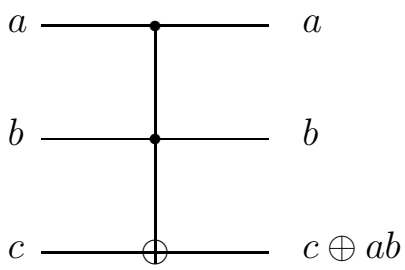
1.4.1 Classical computations on a quantum computer

We can simulate a classical logic circuit using a quantum circuit. ($\because$ classical world $\subset$ quantum world)

Q-circuit cannot be used to directly simulate classical circuit, because unitary Q logic gates are inherently **reversible**, whereas many classical logic gates are inherently **irreversible**. *eg)* NAND gate

$\forall$ Classical circuit can be replaced by an equivalent circuit containing only reversible elements, by making use of a reversible gate like *Toffoli gate*.

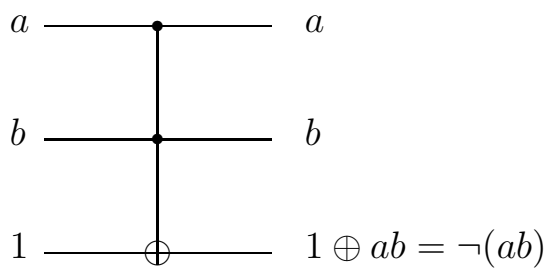
Inputs			Outputs		
a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0



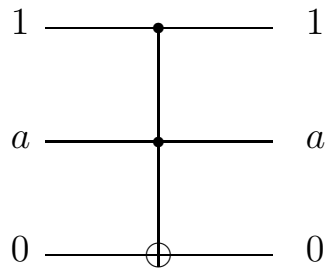
- Toffoli gate has 3 input bits (2 control bits & 1 target bit) and 3 output bits.

$$T(a, b, c) = (a, b, c \oplus ab)$$

- Toffoli gate is reversible.
- NAND gate can be simulated by the Toffoli gate.



- FANOUT gate can be simulated by the Toffoli gate. FANOUT = T (1, a, 0) = (1, a, a)



- $\forall$ classical gate can be simulated using NAND & FANOUT gates.
- Toffoli gate has been described as a classical gate,
  but can also be implemented as a Q logic gate.
  can be represented as a 8x8 unitary mat U.
  is a legitimate Q gate
- The Q-Toffoli gate can be used to simulate irreversible classical logic gates.
- Q computers are capable of performing any computation which a classical computer may do and has ability to efficiently simulate a non-deterministic classical computer.

1.4.2 Quantum parallelism

Q parallelism is a fundamental feature of many Q algorithms.

Q parallelism allows Q computers to evaluate a function f(x) for many different values of x simultaneously.

Suppose $f(x) : \{0, 1\} \rightarrow \{0, 1\}$ is a functioin with a 1-bit domain & range.

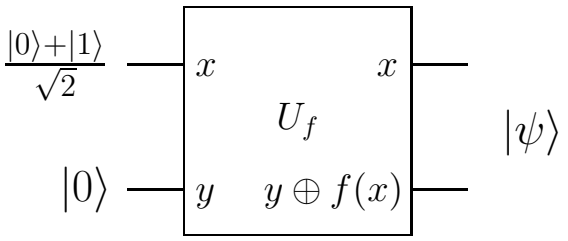
Two qubit Q-computerr computing f(x) : A convenient way of computing $f(x)$ on a Q computer

$$|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$$

where, x is data register and y is target register. If $y = 0$, then the final state of the second qubit is just the value $f(x)$.

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

where, U_f is unitary transformation.



eg) Q-cct evaluating $f(0)$ & $f(1)$ simultaneously.

U_f is the Q-circuit which takes input like $|x, y\rangle$ to $|x, y \oplus f(x)\rangle$

If data register is prepared in the superposition $f(0) = (|0\rangle + |1\rangle)/\sqrt{2}$, which can be created with a Hadamard gate acting on $|0\rangle$

$$\begin{aligned} U_f\left[H(|0\rangle), |0\rangle\right] &= U_f\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}, |0\rangle\right] \\ &= \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}} \end{aligned}$$

It is almost as if we have evaluated f(x) for two values of x simultaneously. A single f(x) circuit evaluate f(x) for multiple values of x simultaneously by exploiting the ability of a Q computer to be in superposition of different states.

This procedure can be easily generalized to ftns on arbitrary # of bits, by using Hadamard transform : n Hadamard gates acting in parallel on n qubits.

For example, n = 2

$$\begin{aligned} H(|0\rangle)H(|0\rangle) &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\ &= \frac{|00\rangle + |01\rangle + |10\rangle + |1\rangle}{2} \end{aligned}$$

More generally,

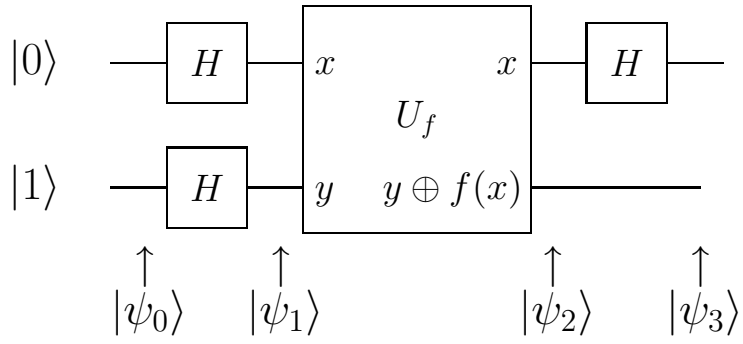
$$H^{\otimes n}|0 \ldots 0\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$$

Hadamard transform produces an equal superposition of all computational basis states and does it extremely efficiently, producing a superposition of 2^n states using just n gates.

Q computation requires the ability to extract information about more than one value of f(x) from superposition states like $\sum_x |x, f(x)\rangle$

1.4.3 Deutsch's algorithm

Deutsch's algorithm combines Q-parallelism with interference.



$$\begin{aligned}
|\psi_0\rangle &= |01\rangle \\
|\psi_1\rangle &= H|\psi_0\rangle \\
&= |H(0), H(1)\rangle \\
&= \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
\end{aligned}$$

Applying U_f to the state $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$

$$\begin{aligned}
U_f|x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} &= |x\rangle\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} \\
&= \begin{cases} |x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(x) = 0 \\ -\ |x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(x) = 1 \end{cases} \\
&= (-1)^{f(x)}|x\rangle\frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
|\psi_2\rangle &= U_f|\psi_1\rangle \\
&= U_f\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\
&= \left[\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\
&= \begin{cases} (-1)^{f(0)}\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0) = f(1) \\ (-1)^{f(0)}\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{if } f(0) \neq f(1) = f(0) \oplus 1 \end{cases}
\end{aligned}$$

$$
\begin{aligned}
&= (-1)^{f(0)} \left[\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}} \right] \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\
|\psi_3\rangle &= (-1)^{f(0)} H \left[\frac{|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle}{\sqrt{2}} \right] \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \\
&= \begin{cases} (-1)^{f(0)}|0\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(0) = f(1) \\ (-1)^{f(0)}|1\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} & \text{if } f(0) = f(1) \neq f(0) \oplus 1 \end{cases} \\
&= (-1)^{f(0)} |f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]
\end{aligned}
$$

Q-circuit can determine a global property of $f(x)$, namely $f(0) \oplus f(1)$, using only one evaluation of $f(x)$.

Possible for two alternatives to interfere with one another to yield some global property of f by using sth like H-gate to recombine the different alternatives.

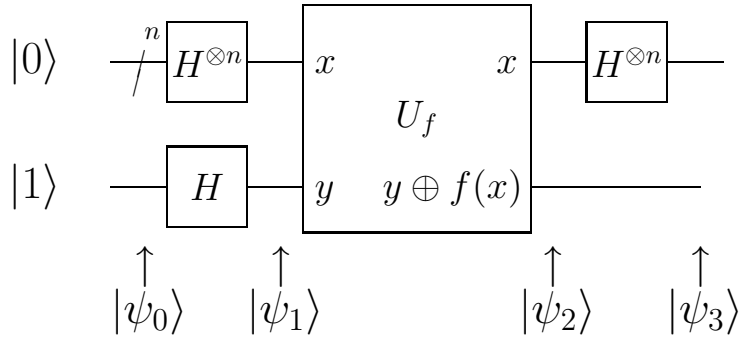
A clever choice of function & final transform allows efficient determination of useful global information about the function.

1.4.4 The Deutsch–Jozsa algorithm

Deutsch's problem

Alice selects a # x from 0 to $2n-1$, & mail it in a letter to Bob.
Bob calculates some ftn f(x) & replies with the result, which is either 0 or 1. Bob has promised to use a function f which is one of two kinds; constant or balanced ftn. Alice's goal is to determine with certainty what ftn Bob has used.

How fast can she succeed? If Alice & Bob were able to exchange qubit, & if B agreed to calculate $f(x)$ using U_f, then A could achieve her goal in just one correspondence with B, using D-J algorithm.



$$
\begin{aligned}
|\psi_0\rangle &= |0\rangle^{\otimes n}|1\rangle \\
|\psi_1\rangle &= \sum_{x\in\{0,1\}^n} \frac{|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
\end{aligned}
$$

The query register is now a superposition of all values

$$
\begin{aligned}
|\psi_2\rangle &= U_f|\psi_1\rangle \\
&= \sum_{[x\in\{0,1\}^n]} (-1)^{f(x)} \frac{|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \\
H|x\rangle &= \frac{|0\rangle \pm |1\rangle}{\sqrt{2}} \qquad x = 0\text{or}1 \\
&= \sum_{[z\in\{0,1\}^n]} (-1)^{xz} \frac{|z\rangle}{\sqrt{2}} \\
H^{\otimes n}|x_1,\ldots,x_n\rangle &= \frac{\sum_{z_1,\ldots,z_n}(-1)^{x_1z_1+\ldots+x_nz_n}|z_1,\ldots,z_n\rangle}{\sqrt{2^n}} \\
H^{\otimes n}|x\rangle &= \frac{\sum_z (-1)^{x\cdot z}|z\rangle}{\sqrt{2^n}} \\
|\psi_3\rangle &= H^{\otimes n}|\psi_2\rangle \\
&= \sum_z \sum_x \frac{(-1)^{x\cdot z+f(x)}|z\rangle}{2^n} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]
\end{aligned}
$$

where, $\sum_x (-1)^{f(x)}/2^n$ is the amplitude for the state $|0\rangle^{\otimes n}$

If f is const, $\sum_x (-1)^{f(x)}/2^n$ is ± 1, depending on the const value of $f(x)$.

An observation will yield zeros for all qubit in the query system. ($\because \langle\psi_3|\psi_3\rangle = 1$)

If f is balanced, $\sum_x (-1)^{f(x)}/2^n$ is 0 (+ & - contribution to $\sum_x (-1)^{f(x)}/2^n$ cancel). Alice's measure must yield a result other than 0 on at least one qubit in the query register.

D-J algo contains the seeds for more impressive Q-algorithm.

1.4.5 Quantum algorithms summarized

The Deutsch-Jozsa algorithm suggests that quantum computers may be capable of solving some computational problems much more efficiently than classical computers. Three classes of quantum algorithms which provide an advantage over known classical algorithms

1. Quantum versions of Fourier transformation
2. Quantum search algrithms
3. Quantum simulations

Quantum algorithm based upon the Fourier transform

The descrete Fourier transform is usually described as trasforming a set $x_0, \ldots, x_{N-1}$ of N complex numbers into a set of complex numbers $y_0, \ldots, y_{N-1}$ defined by

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

The Hadamard transformation used in D-J algorithm is an example of this generalized class of F-transformation. Moreover, many of the other important quantum algorithms also involve some type of Fourier trasformation.

eg) Shor's fast algorithms for factoring and discrete logarithm

Imagine that we define a lineartransformation U on n qubits by its actioin on computational basis states $|j\rangle$, where $0 \leq j \leq 2^n - 1$,

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

Moreover, if we write out its action on superpositions,

$$U \sum_{j=0}^{2^n-1} x_j |j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} \left[\sum_{j=0}^{2^n-1} e^{2\pi ijk/2^n} x_j \right] |k\rangle = \sum_{k=0}^{2^n-1} y_k |k\rangle,$$

Deutsch-Jozsa algorithm, Shor's algorithms for factoring and discrete logarithm.
Classical(FFT) : $n2^n$ steps
Quantum computer : n^2 steps

However, that is not exactly the case; the Fourier transformation is being performed on the "hidden" in the collapse amplitude of the quantum state.

Quantum search of given size N Classical : N operations Quantum computer : $\sqrt{N}$ operations

Q simulation simulate a Q system

The power of quantum computation

The power of quantum computer is not yet verified. (a few example exits)

Computational complexity theory : the subject of classifying the difficulty of various computational problems

A *complexity class* : a collection of computational problems, all of which share some common feature with respect to the computational resources needed to solve those problems.

P The class of computational problems that can be solved quickly on a classical computer.

NP The class of problems which have *solutions* which can be quickly checked on a classical computer.

NP-complete An important subclass of **NP** problems

- There are many important problems that known to be **NP**-complete.
- Any given **NP**-complete problem is in some sense 'at least as hard' as all other problems in **NP**. So, an algorithm to solve a specific **NP**-complete problem can be adapted to solve any other problem in **NP**, with small overhead.

P is a subset of **NP**. What is not so clear is whether or not there are problems in**NP** that are not in **P**.

If $\mathbf{P} \neq \mathbf{NP}$, then it will follow that no **NP**-complete problem can be efficiently solved on a classical computer

Factoring is not known to be **NP**-complete, and it can be solved using quantum algorithm through quantum parallelism - quantum computation is expected to solve other **NP** problems.

PSPACE Problems which can be solved using resources which are few in spatial size, but not necessarily in time. **PSPACE** is believed to be strictly larger than both **P** and **NP**

BPP The class of problems that can be solved using randomized algorithms in polynomial time, **BPP** is widely regarded as being the class of problems which should be considered efficiently soluble on a classical computer.

What of quantum complexity classes?

BQP : The class of all computational problems which can be solved efficiently on a quantum computer, where a bounded probability of error is allowed.

Exactly where **BQP** fits with respect to **P**, **NP** and **PSPACE** is as yet unknown.

BQP may be somewhere between **P** and **PSPACE**