

Explore a new era of IT with
Quantum Computing at IBM

IBM Client Center Montpellier

Q



Quantum computing principles
Definition of a quantum bit
Controlling a qubit
Working with many qubits



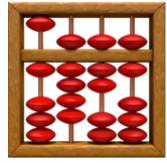
Explore a new era of IT with
Quantum Computing at IBM

IBM Client Center Montpellier

Q



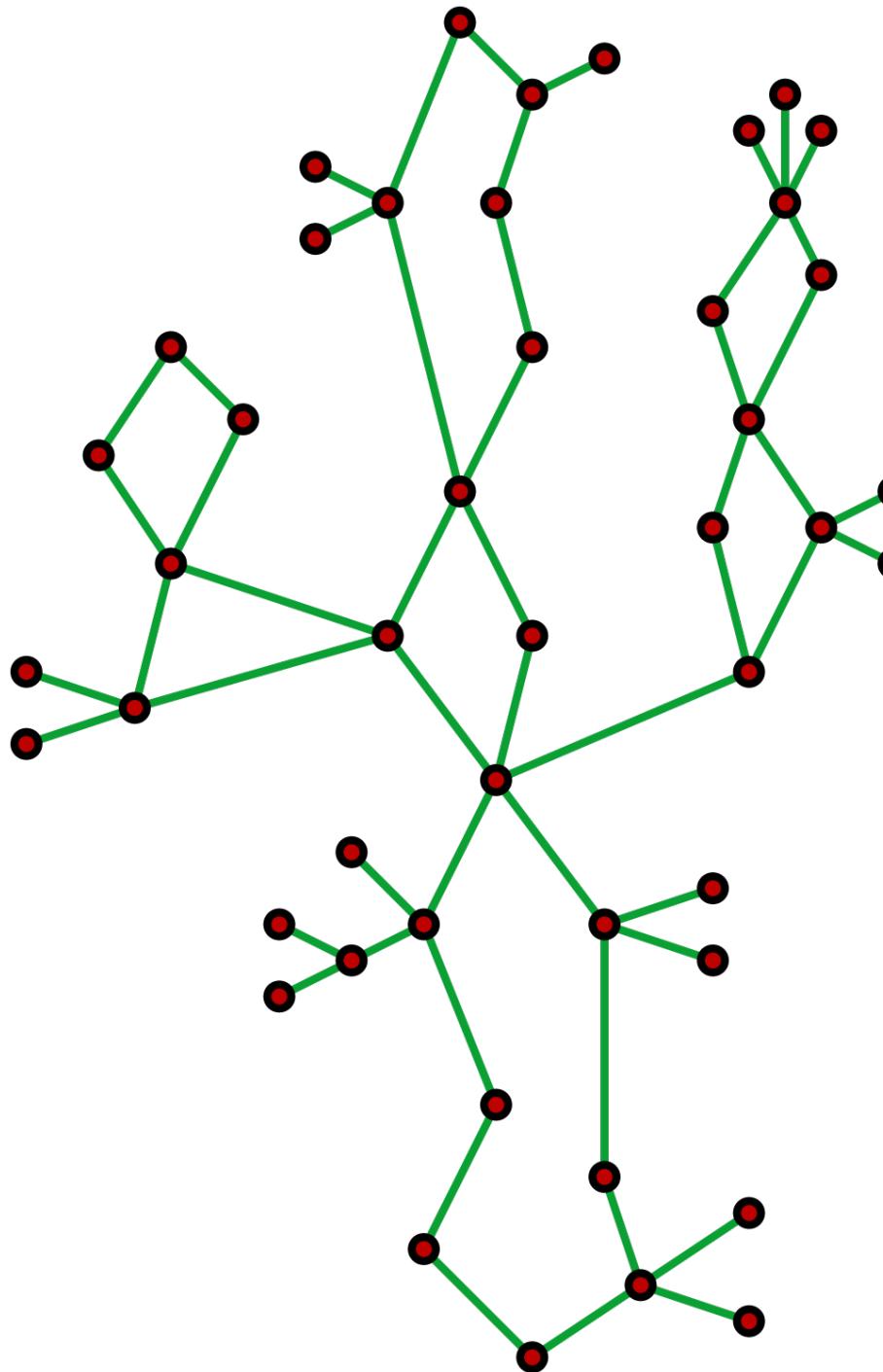
Quantum computing principles



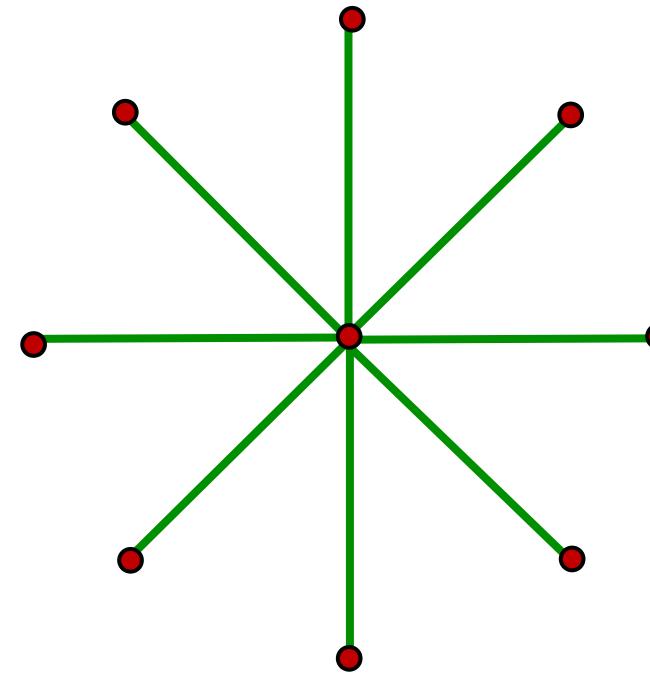
Turing machine and other computational models

a graph

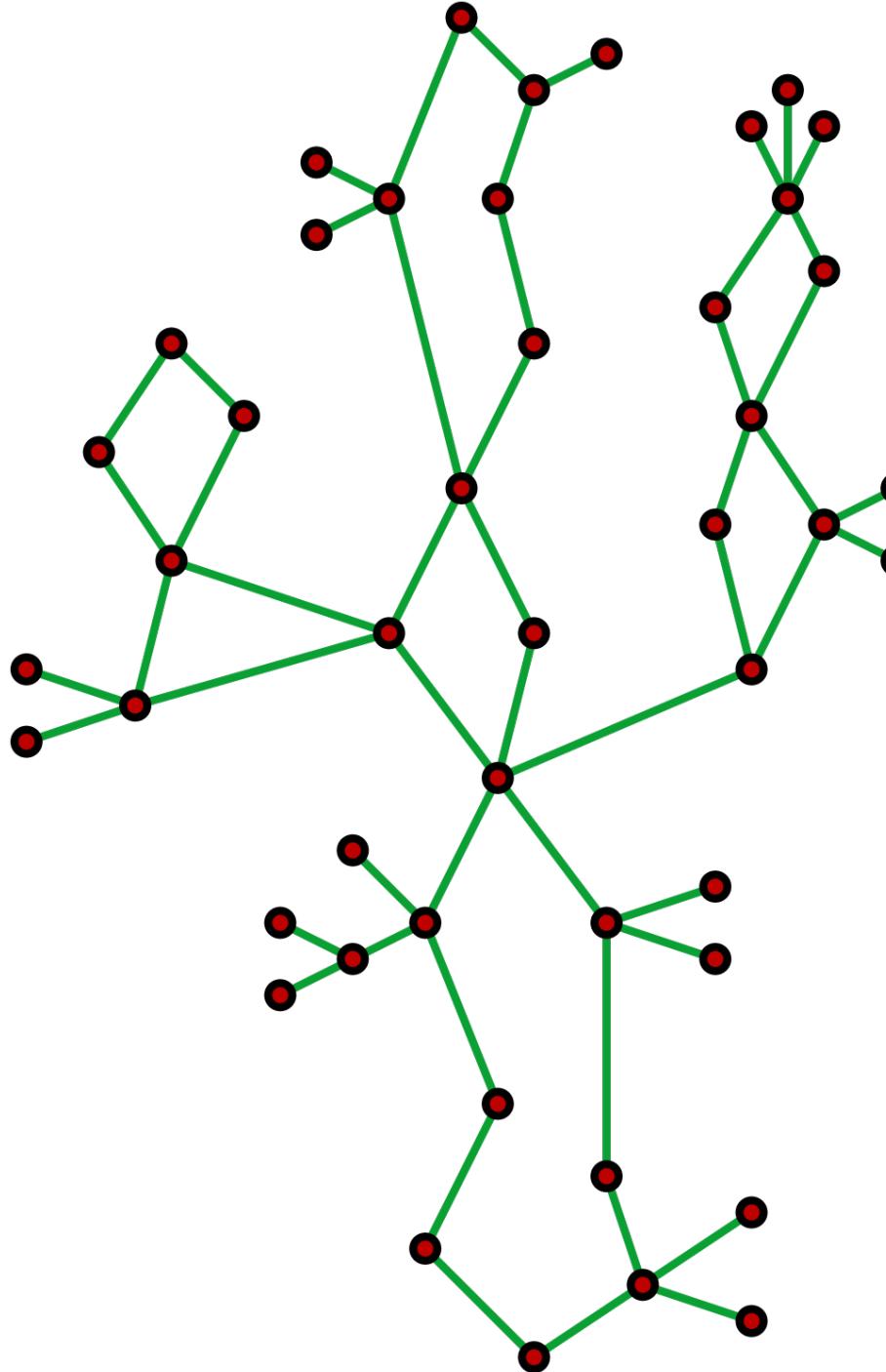
How can we
build a planar
graph of order n
with minimal
diameter (and
what is that
value) ?



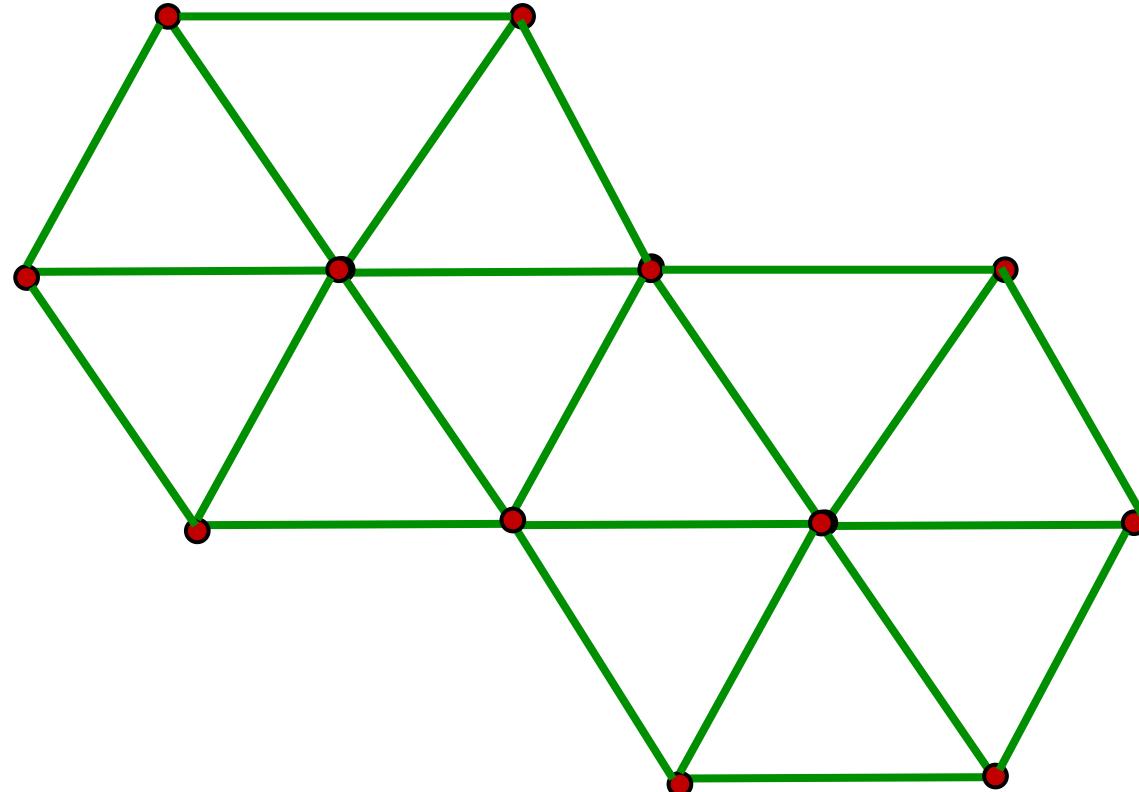
2



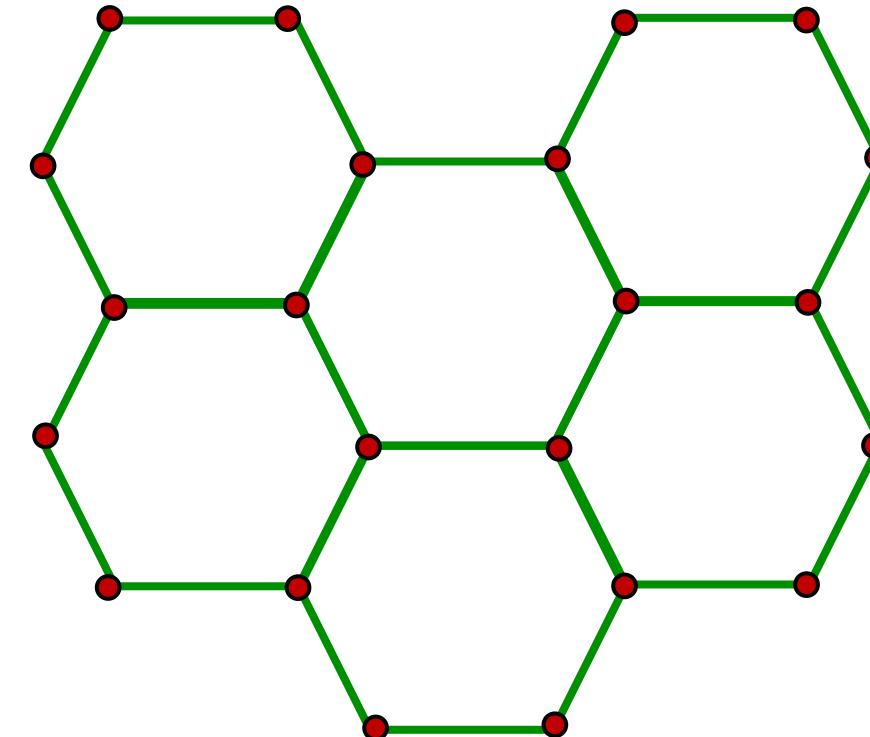
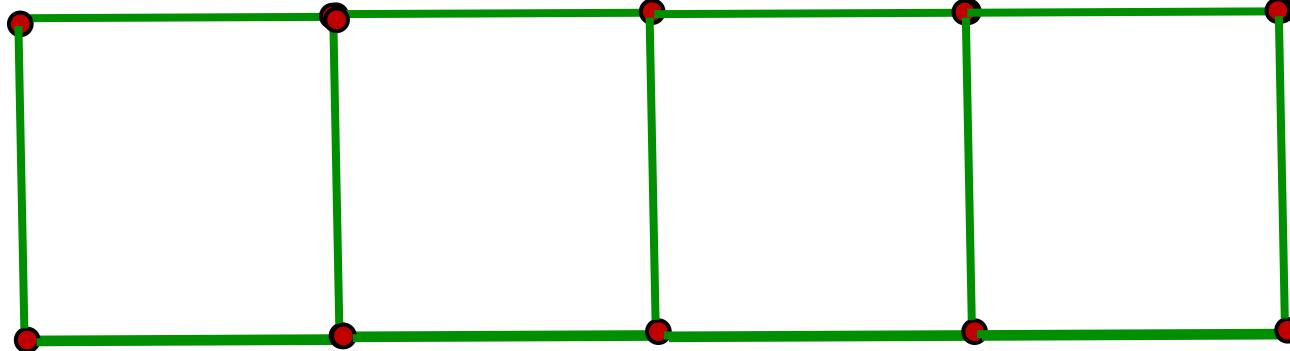
How can we build
a planar graph of
*order n , all
vertices having
max order 6, with
minimal diameter*
(and what is that
value) ?



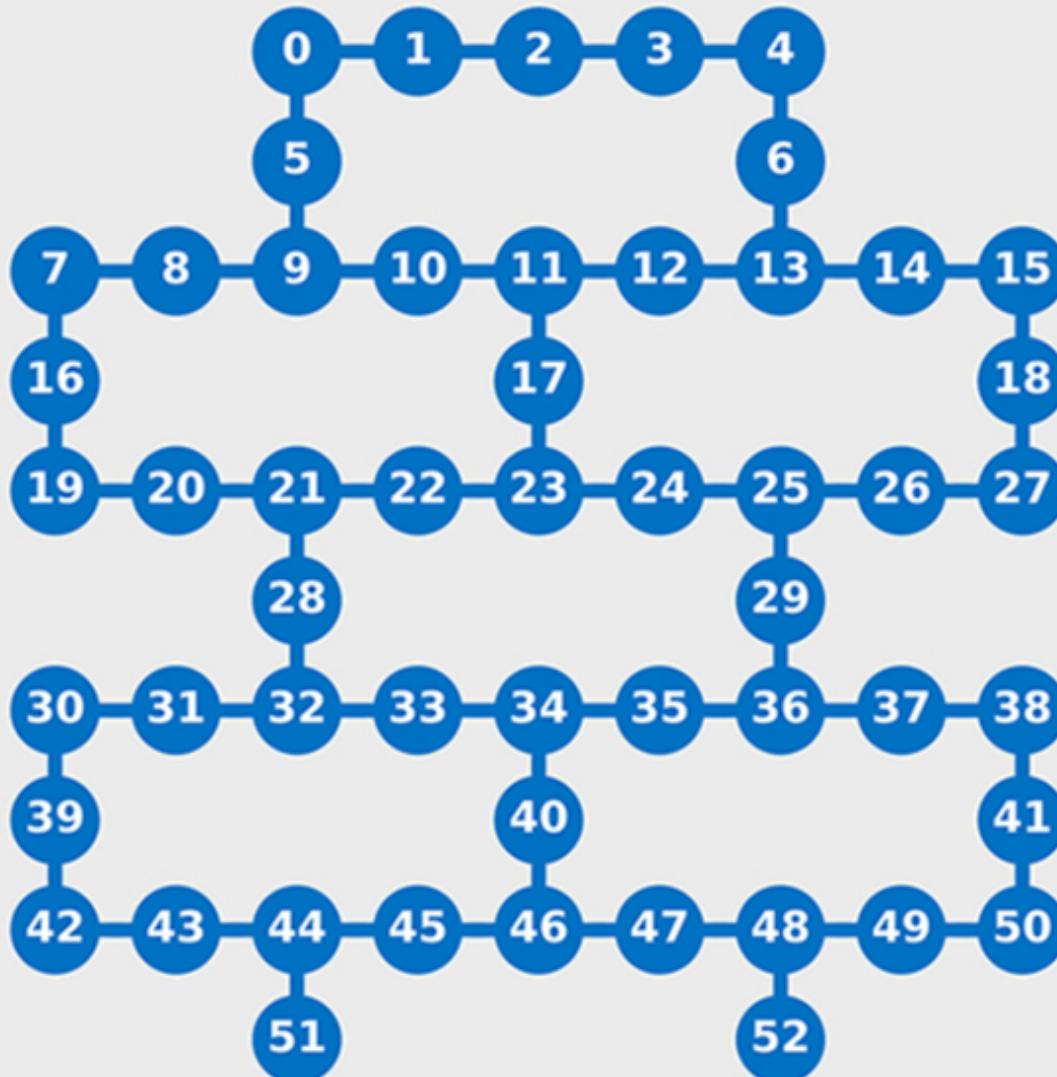
How can we build
a planar graph of
*order n , all
vertices having
max order 6, with
minimal diameter
(and what is that
value)?*



How can we build
a planar graph of
*order n , all
vertices having
max order 3, with
minimal diameter
(and what is that
value)?*



IBM Q System 53 qubit ("Rochester")

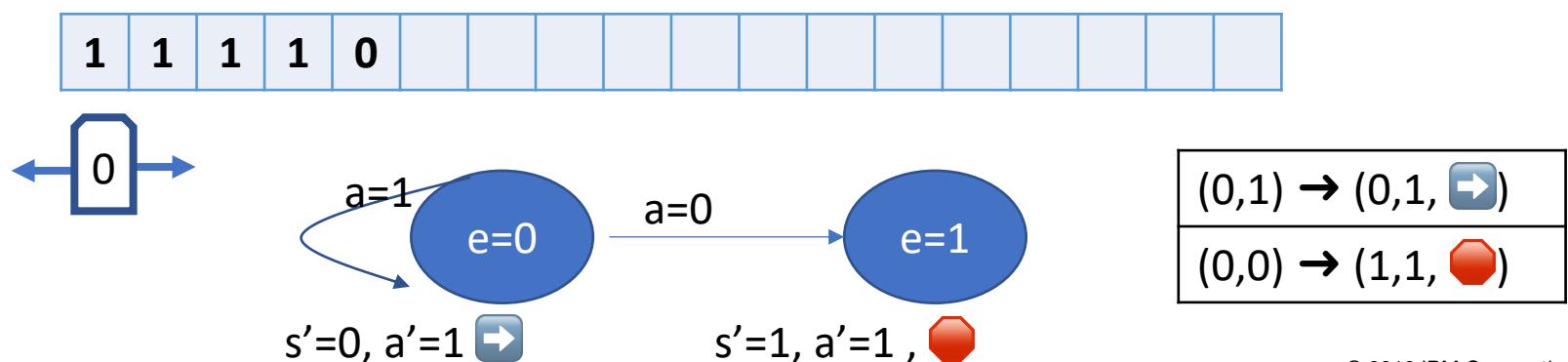


Turing Machine



- infinite tape strip
- mobile read/write head
- set of states s_0, s_1, s_2, \dots
- alphabet of symbols a_0, a_1, a_2, \dots
- transition table, list of :
(current state; symbol) \rightarrow (new state, new symbol, move: $\rightarrow/\leftarrow/\text{stop}$)

For example the following machine increments a number n by one unit ($n \rightarrow n+1$) :

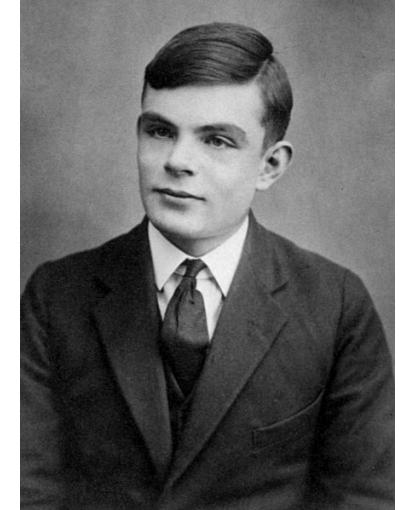


Church-Turing thesis

Church-Turing thesis : **any computation can be simulated by a Turing Machine**

JUL
17

1936 : Alan Turing states there exists Universal Turing Machines (can simulate any Turing Machine)

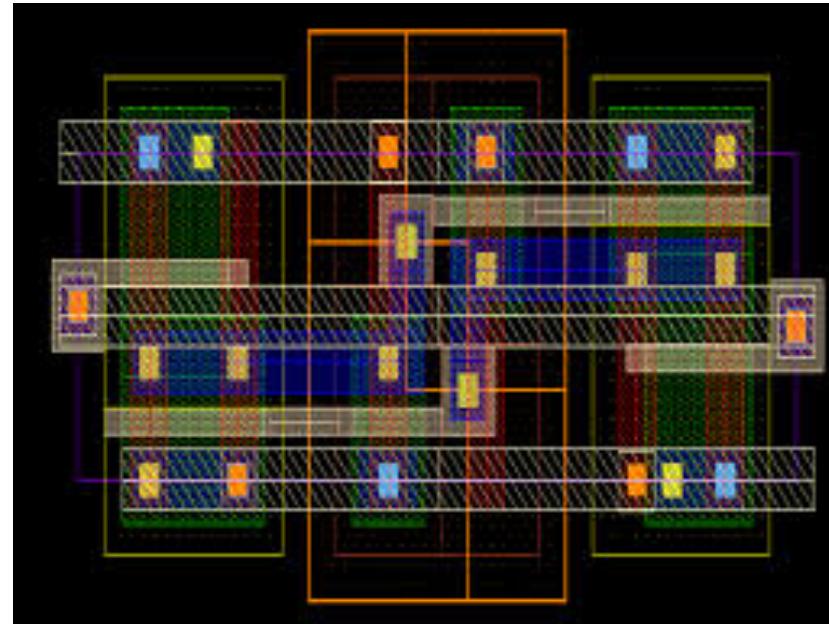


JUL
17

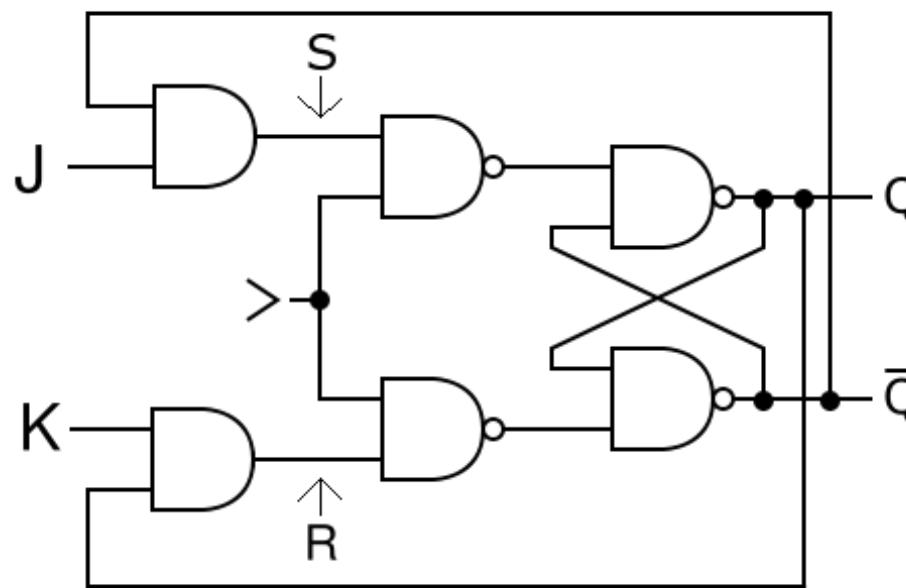
2002 : Yurii Rogozin proves an Universal Turing Machine can be build using only 4 states, 9 symbols and 24 transitions.

but... computation time are not practical, and the tape can be very long...

circuit model



Boolean logic



← this is a bit

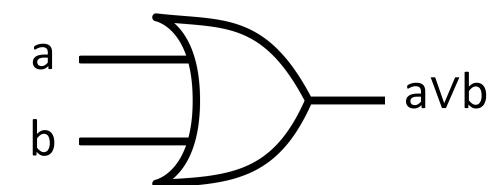
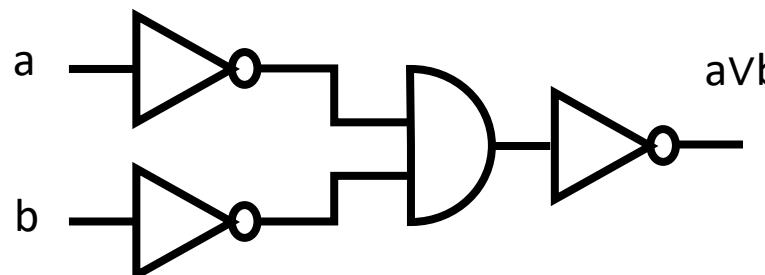


universality

{NOT, AND} : is a universal gate set

Morgan theorem : OR gate can be build from AND and NOT :

$$a \vee b = \neg(\neg a \wedge \neg b)$$



a	b	$\neg a$	$\neg b$	$\neg a \wedge \neg b$	$\neg(\neg a \wedge \neg b)$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1



{NOT, AND} : is a universal gate set

Any logical function can be calculated with NOT and AND gates (and OR) :

let $f : n+1 \text{ bits} \rightarrow \{0,1\}$

we can define : $f_0(X_1, X_2, \dots, X_n) = f(\mathbf{0}, X_1, X_2, \dots, X_n)$

and.

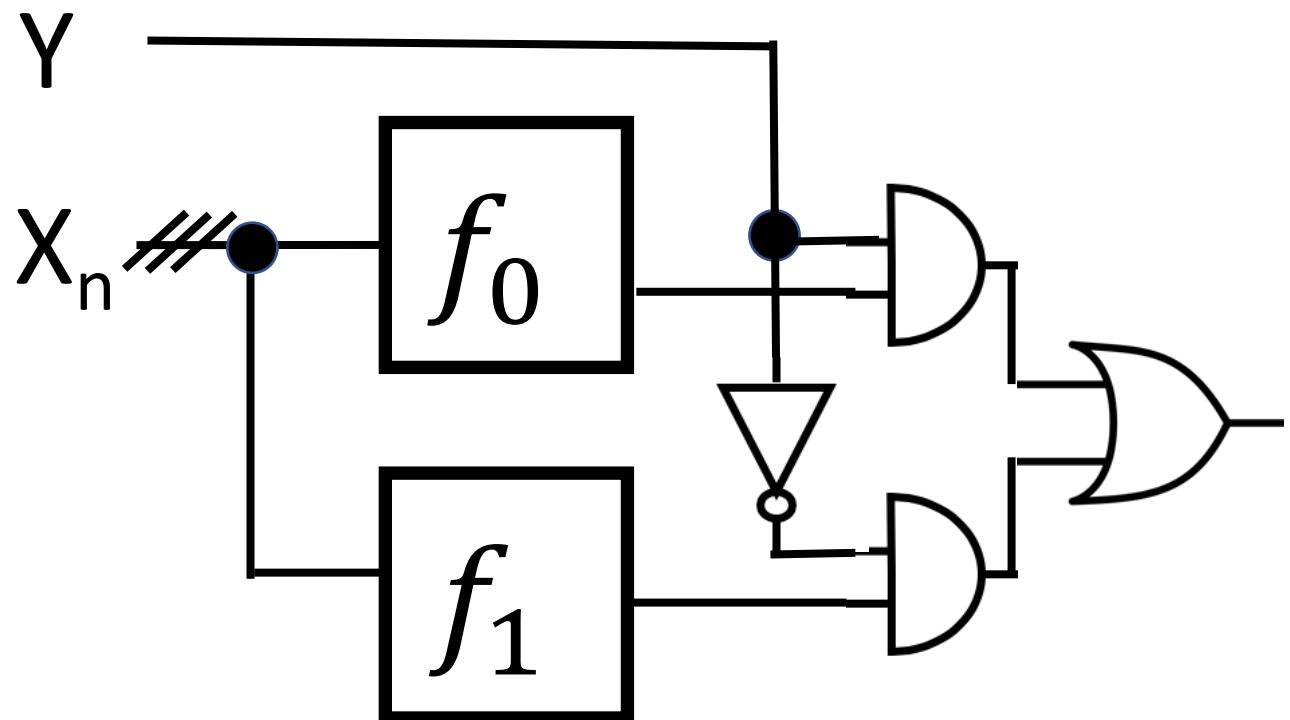
$f_1(X_1, X_2, \dots, X_n) = f(\mathbf{1}, X_1, X_2, \dots, X_n)$

if f_0 & f_1 can be calculated with AND and NOT, then f can be as well :

$$f(Y, X_1, X_2, \dots, X_n) = (\neg Y \wedge f_0) \vee (Y \wedge f_1)$$

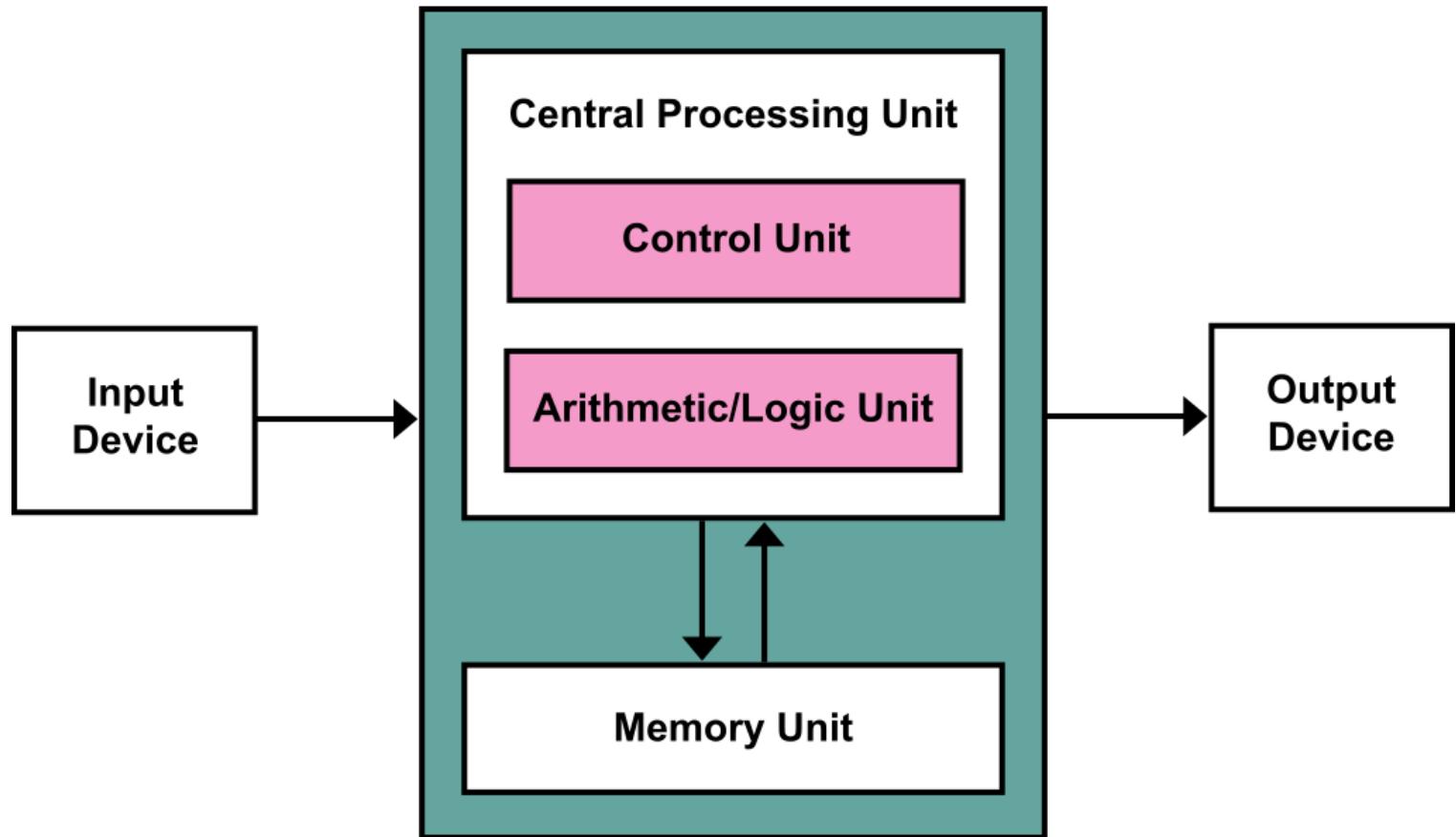
2^n

But, the gate count increases exponentially (size and power consumption as well)

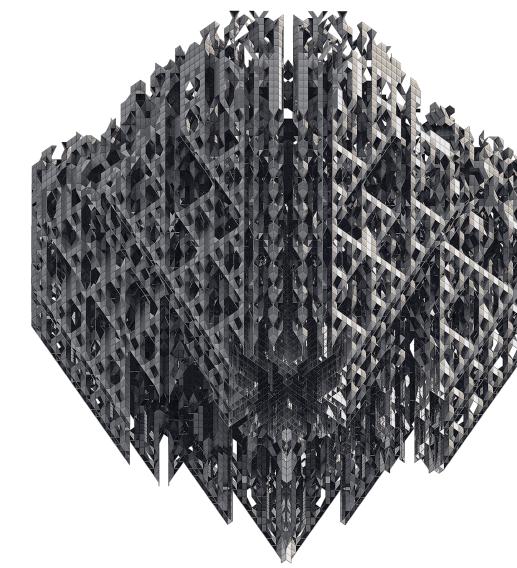
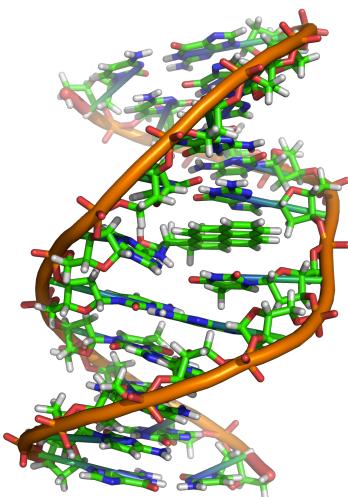
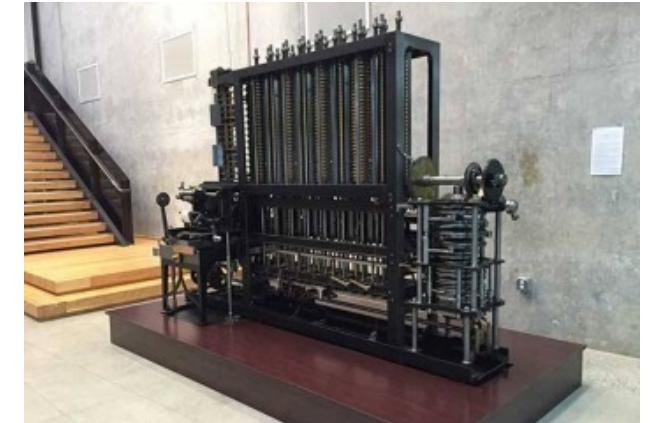
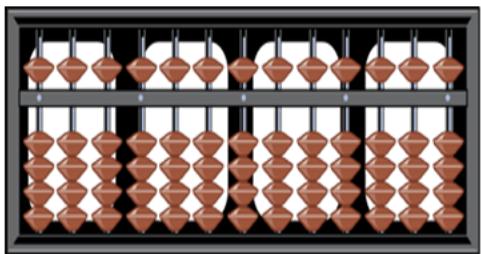


“classical”
computers

Von Neuman architecture (ALU+Memory) can work-around the circuit size problem, but the computation time issue is not solved :



Other computational models



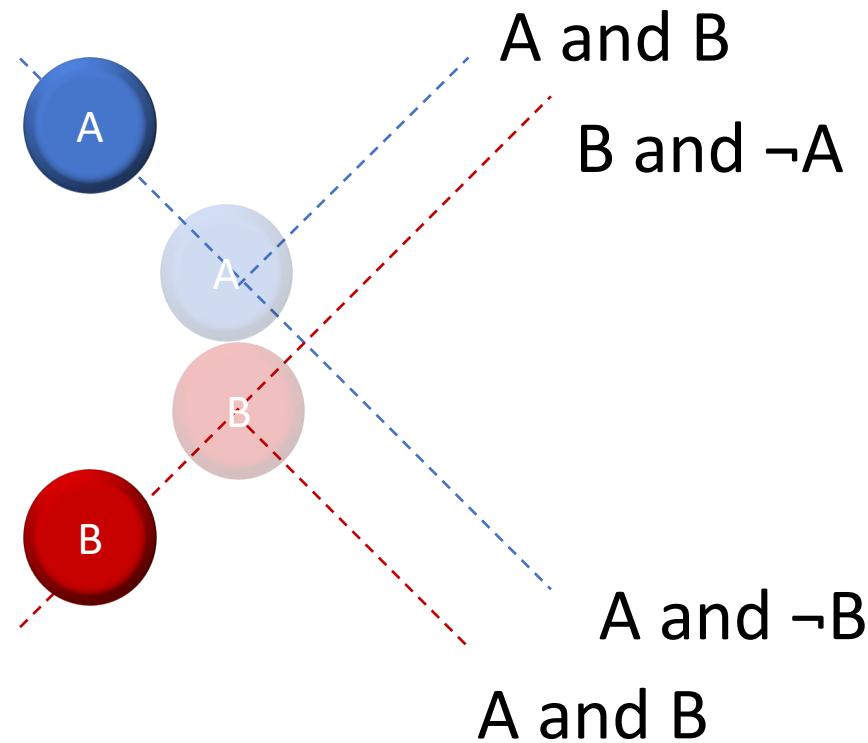
Q



reversible computation

reversible computation

Reversible mechanical computer : « billiard balls Machine » (BBM)
Edward Fredkin & Tommaso Toffoli - 1982



<http://jameslin.name/bball/>

reversible computation

Reversible mechanical computer : « billiard balls Machine » (BBM) Edward Fredkin & Tommaso Toffoli - 1982

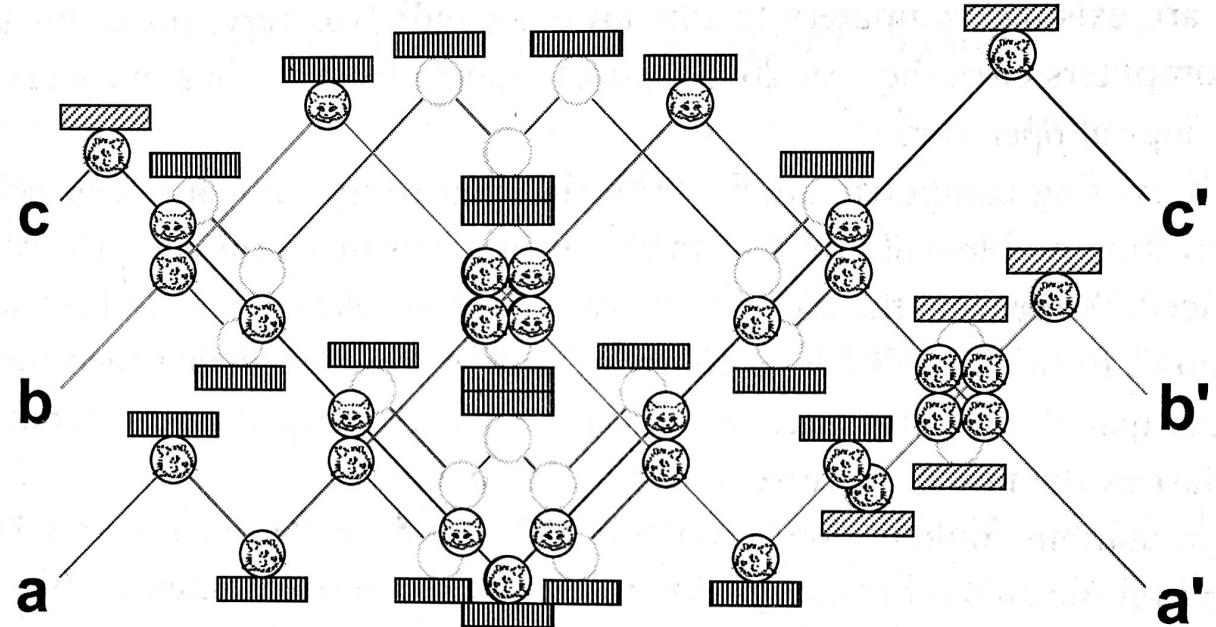


Figure 3.14. A simple billiard ball computer, with three input bits and three output bits, shown entering on the left and leaving on the right, respectively. The presence or absence of a billiard ball indicates a 1 or a 0, respectively. Empty circles illustrate potential paths due to collisions. This particular computer implements the Fredkin classical reversible logic gate, discussed in the text.

Michael Nielsen and Isaac Chuang (2000). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press. ([ISBN 0-521-63503-9](#)).



Computation and energy:

Landauer Principle (first form) : when a bit is erased in a computer, the value of energy given to its environment is at minimum : $kT\log 2$

Landauer Principle (second form) : when a computer erases a bit of information, its environment entropy raises by at least $kT\log 2$:

$kT\log 2$

$3 \cdot 10^{-21} \text{ J}$

Current computers operate at 500 times this limit

Reversible computation (of which Quantum Computing) is not constrained by this limit.

Rolf Landauer. Irreversibility and heat generation in the computing process. IBM Journal of Research and Development, 5:183, 1961.



algorithmic complexity



algorithmic complexity

Searching an element from a non sorted list :

```
list = [u,g,r,e,x,d,t,l,o,p,q,m,h,z,a,v,n,i,k,f,b,c,s,d,w...];
searched = y;
for (i = 0; i < list.length(); i++) {
    if (list[i] == y) {
        found = 1;
        print(y, « found at position », i);
        break;
    }
}

if (found == 0) { print(y, « not found in list »); }

liste = [u,g,r,e,x,d,t,l,o,p,q,m,h,z,a,v,n,i,k,f,b,c,s,d,w...];
searched = y;
l = liste.length();
for (i = 0; i < l; i++) {
    if (liste[i] == y) {
        found = 1;
        print(y, « found at position », i);
        break;
    }
}

if (found == 0) { print(y, « not found in list »); }
```

algorithmic complexity

Information technology complexity : $f(\text{space, time})$

For a given problem P, looking at time complexity is studying how the number of instructions changes to solve this problem as a function of a parameter n which represents the size of this problem.

Let $C(n)$ be that number of instructions, we look for an evaluation of $C(n)$ when $n \rightarrow \infty$

One way to do this is to compare $C(n+1)$ with to $C(n)$

$C(n)$ evolution	Complexity	« big O » notation
$C(n+1) = C(n)$	constant	$\mathcal{O}(1)$
$C(n+1) = C(n) + \varepsilon$ (par exemple $C(n+n) = C(n) + 1$)	Logarithmic	$\mathcal{O}(\log(n))$
$C(n+1) = C(n) + k$	Linear (kn)	$\mathcal{O}(n)$
$C(n+1) = C(n) + n$	Polynomial	$\mathcal{O}(n^2)$
$C(n+1) = C(n) + n^k$	Polynomial	$\mathcal{O}(n^{k+1})$
$C(n+1) = C(n)*2$	Exponential	$\mathcal{O}(2^n)$
$C(n+1) = C(n)*n$	Exponential	$\mathcal{O}(n!)$
...		



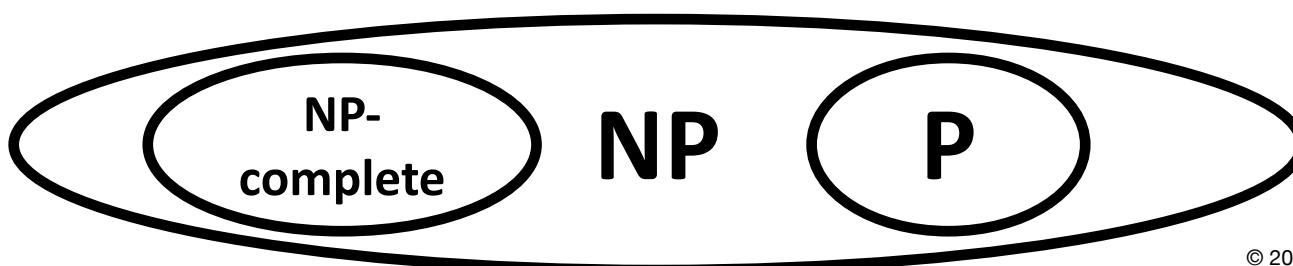
complexity examples

Dealing cards to n players :

Distribution process	\mathcal{O}	example
52 cards deck is distributed, whatever the quantity of players.	$\mathcal{O}(1)$	Fetch 10 first lines of a data list.
Deal one card to player 1, half the number of players, deal one card, half the number of player, deal another card, and so on...	$\mathcal{O}(\log_2(n))$	Search for one value in a sorted list.
Deal one card to each player	$\mathcal{O}(n)$	Search for one value from a non sorted list
Deal n cards to n^{th} player	$\mathcal{O}(n^2)$	Process all pairs of a set.
Deal 1 card to first player, 2 cards to 2 nd player, 4 cards to 3 rd , 8 to 4 th , and so on : to each player deal twice the number of cards compared to previous player.	$\mathcal{O}(2^n)$	When we need to compute on all subsets of a set.

P & NP are two sets of decision problems :

- a problem p can be solved efficiently if there is an algorithm of polynomial order solving p . Then p is in P.
- there are problems for which we know of an exponential algorithm that can solve them, but we do not know if there is a polynomial algorithm to solve them. But we know a polynomial algorithm to verify a solution. Those are NP problems.
- It is not known if $P = NP$.
- We know that there is a class of NP problems that collect all the difficulty of any NP problem. These are called NP-complete.
- for any one of the NP-complete problem :
 - If we can prove no polynomial algorithm can solve it : then $P \neq NP$
 - if one polynomial algorithm is found, then $P = NP$.



« 3SAT » satisfiability problem :

Satisfiability: For a logical expression F using n booleans B_i , we ask if there exists a combination of values for B_i so that $F = \text{True}$?

3SAT has the following expression (for example) :

$$F = (B_0 + B_3 + B_4) * (\neg B_1 + B_5 + \neg B_7) * (B_2 + B_3 + \neg B_4) * \dots * (\neg B_0 + B_1 + B_6)$$

Solving the 3SAT problem requires the evaluation of F for all possible combination of the values of the n booleans.



3SAT

It is a NP-complete problem.

Note that 2SAT is in class P:

$$(F = (B_0 + B_3) * (\neg B_1 + B_2) * (B_2 + B_3) * \dots * (\neg B_0 + B_4))$$

N-complete examples

CLIQUE (graph theory): a CLIQUE in a non oriented graph is a set of vertices that are all connected together. The CLIQUE size is the number of vertices it contains. For a graph G, is there a CLIQUE of size N ?

SUM of SUBSETS (arithmétique) : given a finite set E of integer, and an integer s : is there a subset of E for which the sum of its elements adds up to s ?

MAXCUT (graph theory) Given a graph, with weighted edges, a cut is a subset of vertices. The cut weight is the summation of weight for edges having one end inside the subset and the other outside. A cut is called a maxcut if it has the maximum possible weight of all cuts.

KNAP SACK(combinatory optimization) : given a set of objects for which we know weight and value : what subset of object should I put in the KnapSack with the maximum value under a maximum weight value.

https://fr.wikipedia.org/wiki/Liste_de_probl%C3%A8mes_NP-complets

BQP is the category of algorithms that can be solved efficiently with a quantum algorithm :

Assumed
relation
between
BQP and
other
complexity
classes

PSPACE problems

NP problems

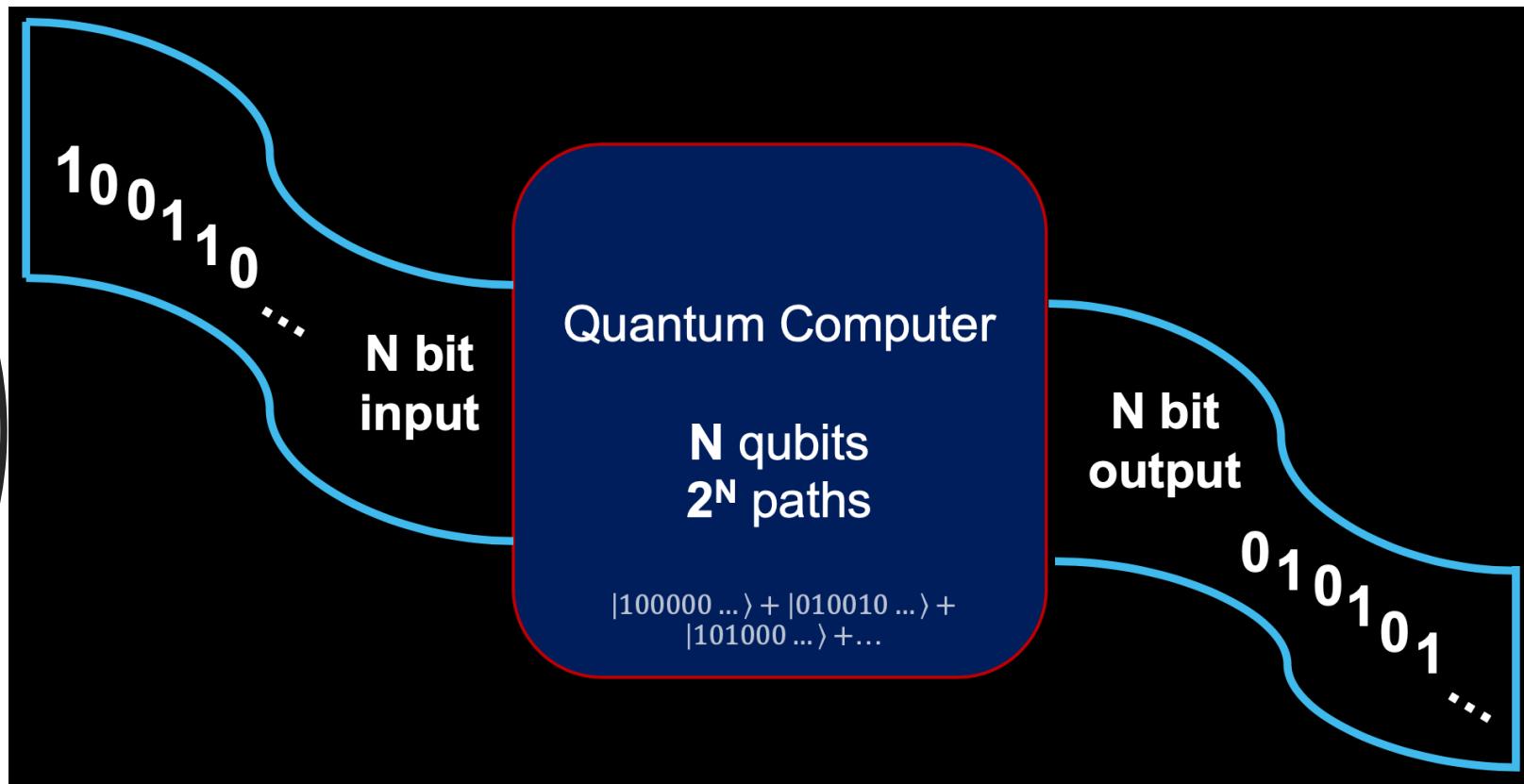
NP complete

P problems

BQP



BQP



Explore a new era of IT with
Quantum Computing at IBM

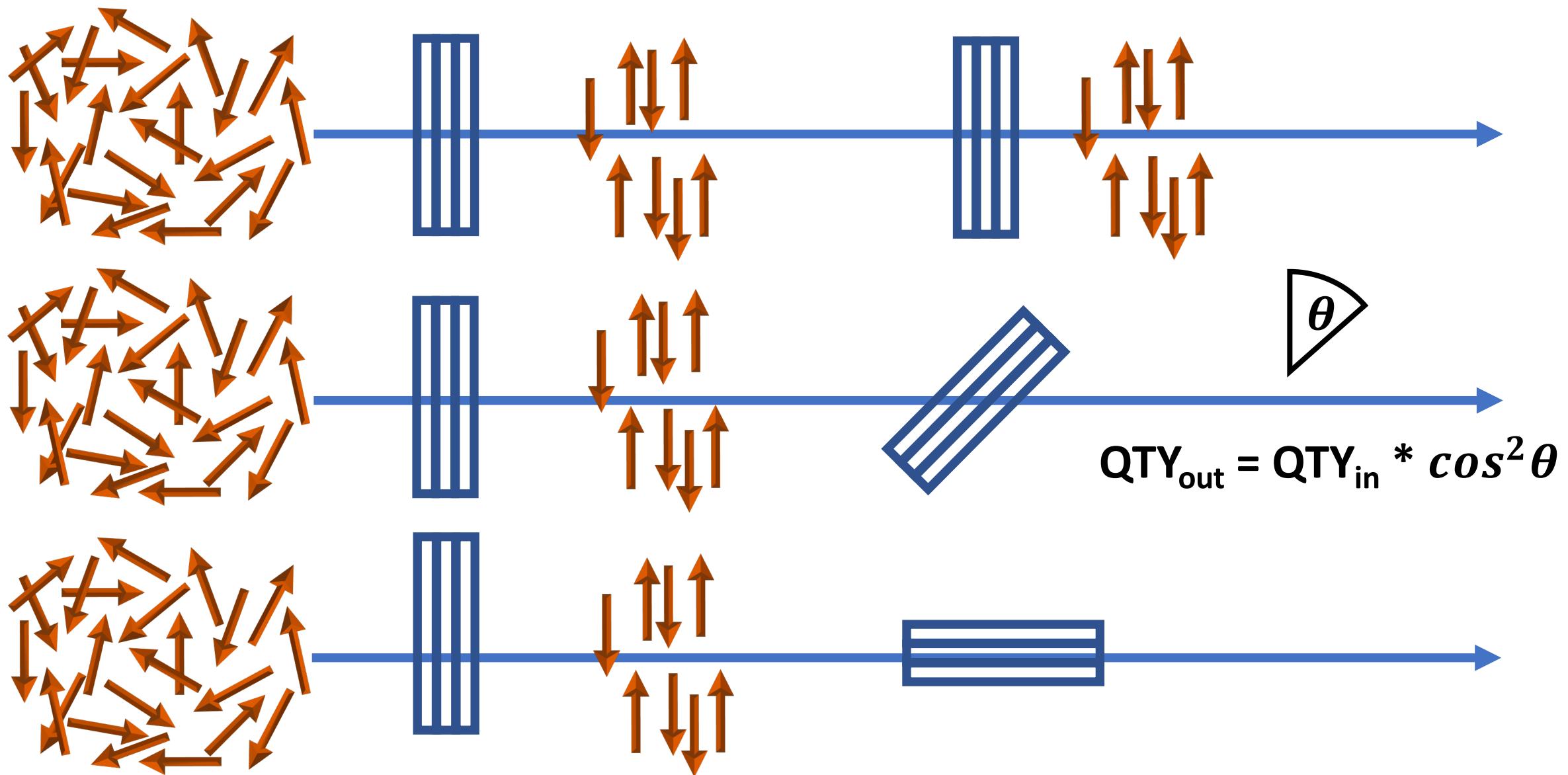
IBM Client Center Montpellier

Q

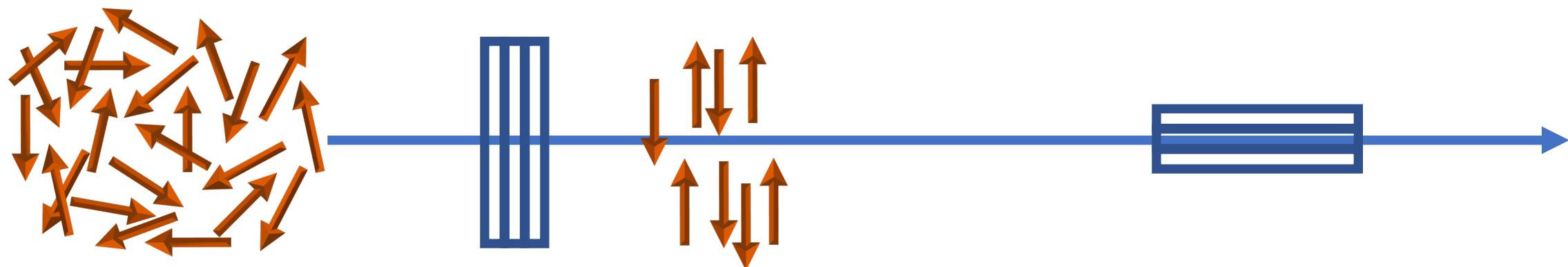


Definition of a quantum bit

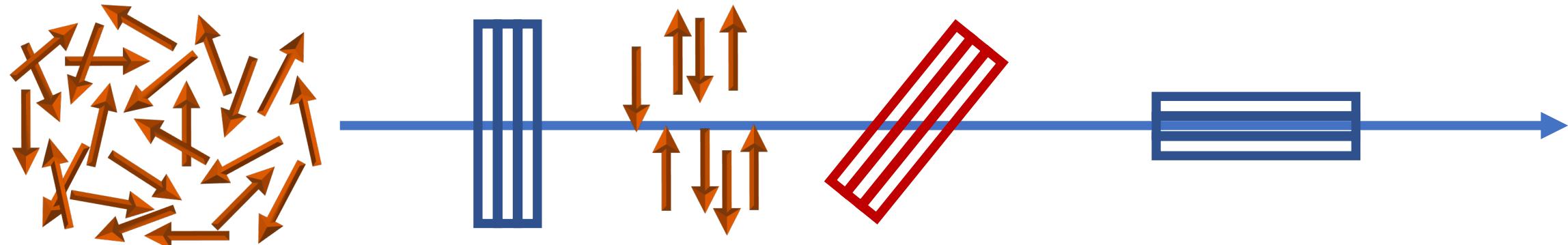
Polarized photons (A)



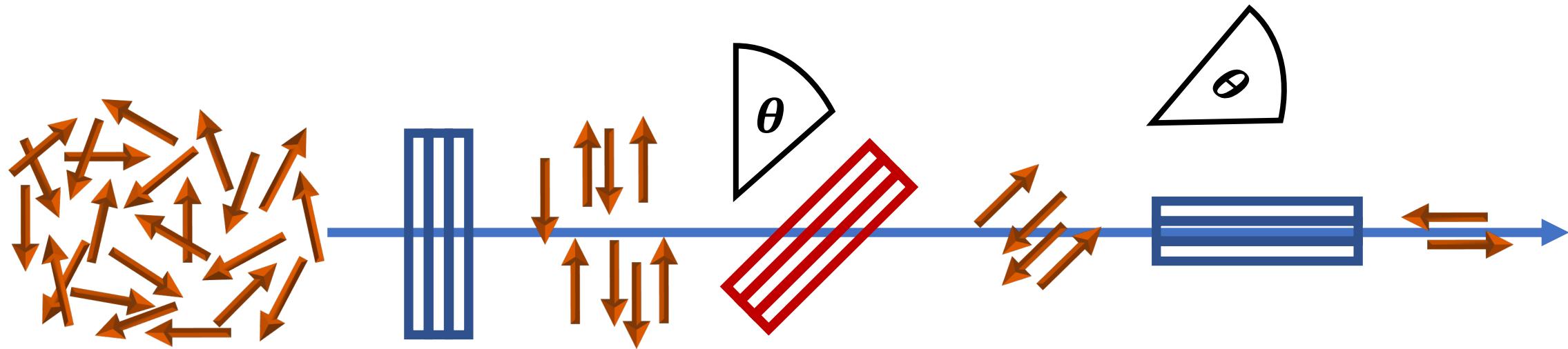
Polarized photons (B)



?

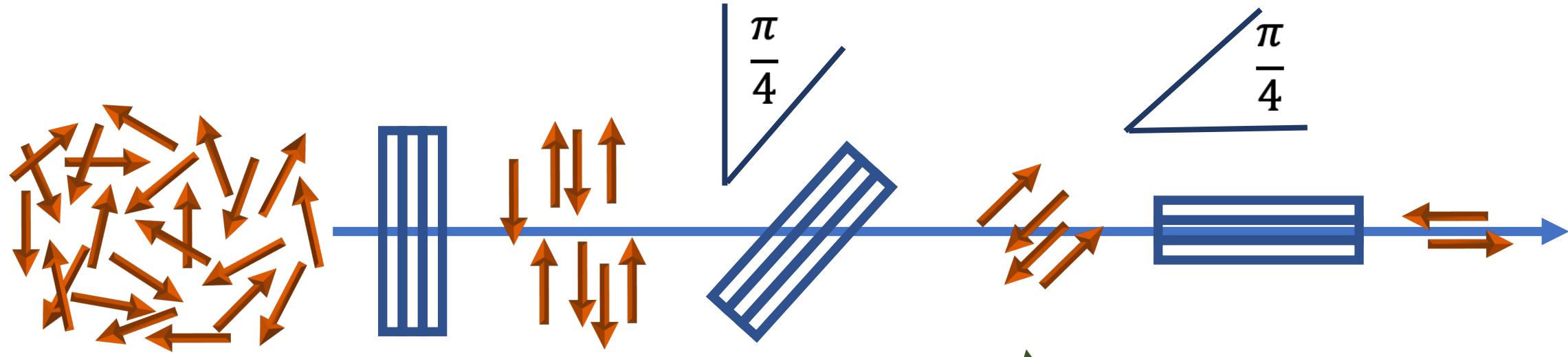


Polarized photons (B)



$$\text{QTY}_{\text{out}} = \text{QTY}_{\text{in}} * \cos^2 \theta$$

Polarized photons (B)



$$\cos^2 \frac{\pi}{4} = \frac{1}{2}$$

100 → 50 → 25

Thought experiment (1/2)

Consider the spin as a magnetic moment which we can imagine as a rotation of the particle around itself (this is only an image)

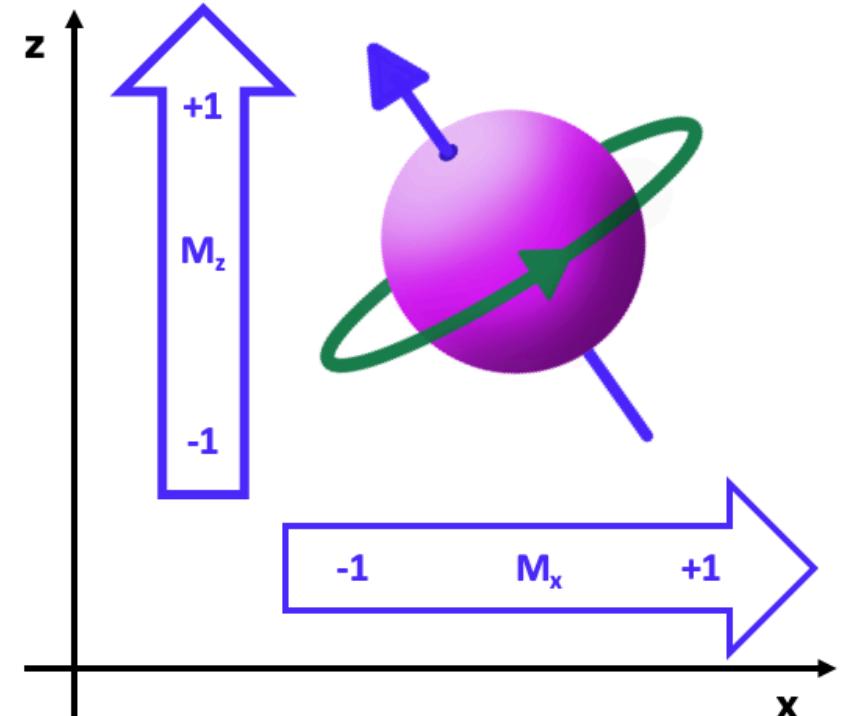
Looking at the two following statements:

- A: measuring the spin along z axis results in +1
- B: measuring the spin along x axis results in +1

I want to study ($A \vee B = 1$)?

With classical reasonning :

1. if I get a result +1 on z : the experiment is over : $A \vee B = 1$.
 - if not, I measure on x : if I get +1 : experiment is over : $A \vee B = 1$
 - else $A \vee B = 0$
2. The other way around (starting with x) : same thing :
 - if I get +1 on x : experiment is over : $B \vee A = 1$.
 - if not, I measure on Z, if I get +1 : experiment is over : $B \vee A = 1$
 - else $B \vee A = 0$

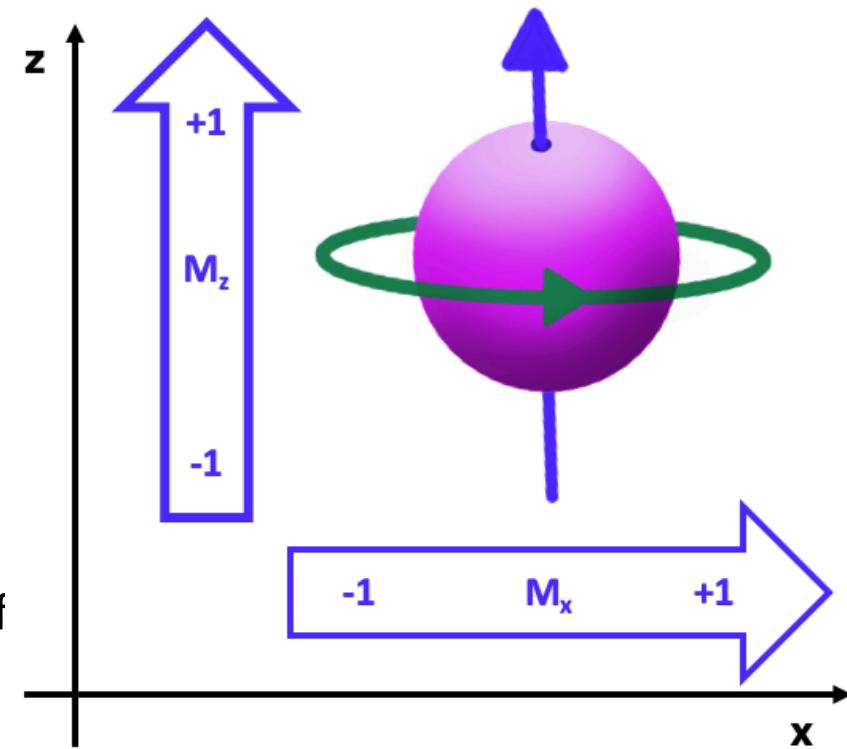


Thought experiment (1/2)

Now with quantum mechanics:

Assume we have prepared the spin on +1 over z.

1. I measure +1 on z : experience is over : AvB=1
2. Now with the same starting condition, let's evaluate ($BvA==1$)?
 - measuring on x : result will be +1 half of the time and -1 the other half. If I found +1 experience is over : $BvA = 1$ (50% chances)
 - If I found -1, now I have to measure against z axis, but now the previous measurement has moved the spin along x, so measuring on z will give -1 in 50% of cases, and +1 in 50% of cases.
 - So at the end $BvA=1$ in 75% of the cases.

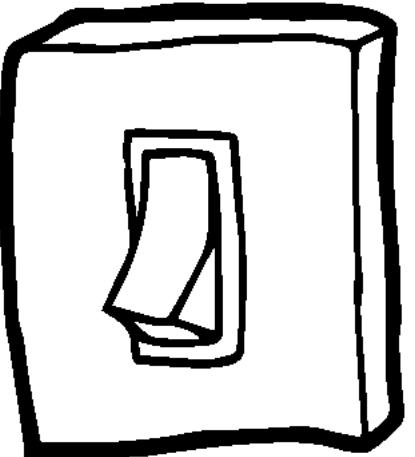


Conclusion : if an interaction has enough energy to perform a measurement of one property of the system, it also has enough energy to modify the system.

In other words : we cannot learn anything from a quantum system without changing it.

bits & qubits

0



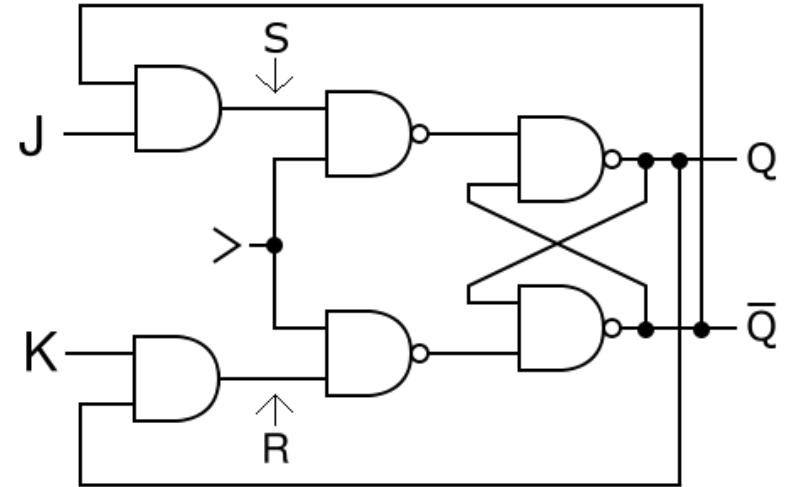
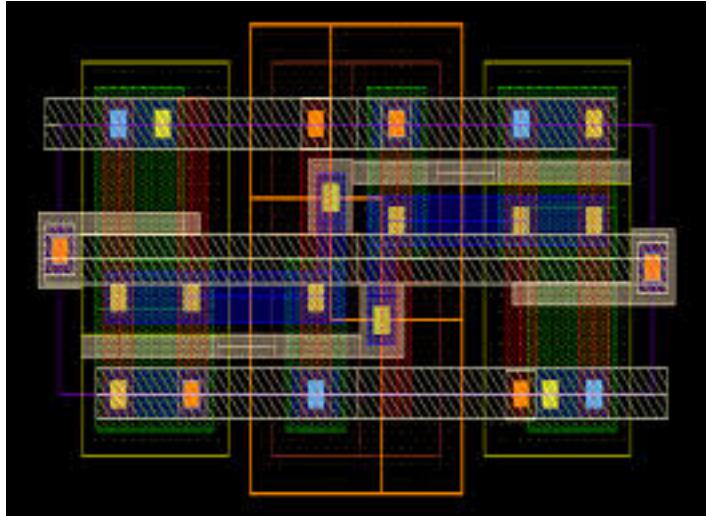
1

0



1

Classical bits properties



- well defined state : 0 or 1, no other value permitted,
- reading its value does not change it,
- information is local.

Classical computers use classical bits ques »

« bits » : (BInary digiT) : smallest piece of information :

True /False, Yes/No, Vcc/0

Usually represented with the values 0 or 1.

Single bits operations : Identity, Not, Set, Unset:

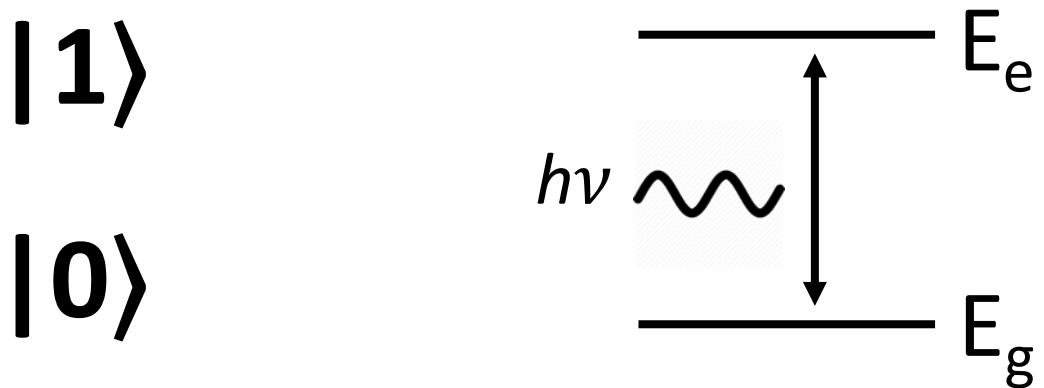
a	$\neg a$	Iden(a)	Set(a)	Unset(a)
0	1	0	1	0
1	0	1	1	0

2 bits operations « Boolean Logic » :

a	b	$a \wedge 0$	$a \wedge b$ [AND]	$a \wedge \neg b$	a	$\neg a \wedge b$	b	$(a \vee b) \wedge \neg(a \wedge b)$ [$a \oplus b$] [XOR]	$\neg(a \vee b)$	$\neg a \wedge \neg b$	$\neg a \wedge \neg b \vee (a \wedge b)$	$\neg b$	$a \vee (\neg a \wedge \neg b)$	$\neg a$	$a \vee b$	$\neg(a \wedge b)$ [NAND]	$a \vee 1$
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Quantum bit : qubit

Similar to a bit, a qubit is the basic unit of information in quantum computing, and can be in one of those two states :

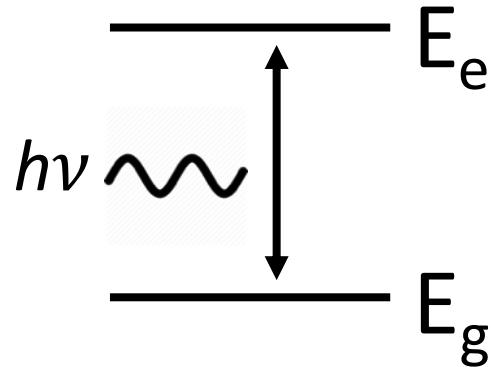


The qubit, however, can also be in a ***superposition state*** : a linear combination of the basic states $|0\rangle$ and $|1\rangle$:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Quantum bit : qubit

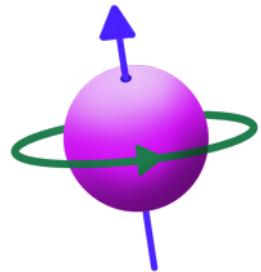
$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$



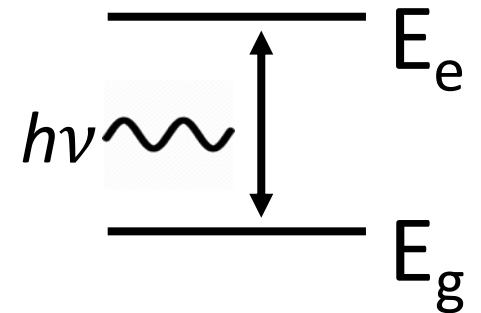
- Can be controlled to a chosen state
- State can be read (sort of)
- $h\nu$ is typically a very small quantity of energy
- Goes back to its ground state after a specific time



Quantum bit : 3 rules



$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

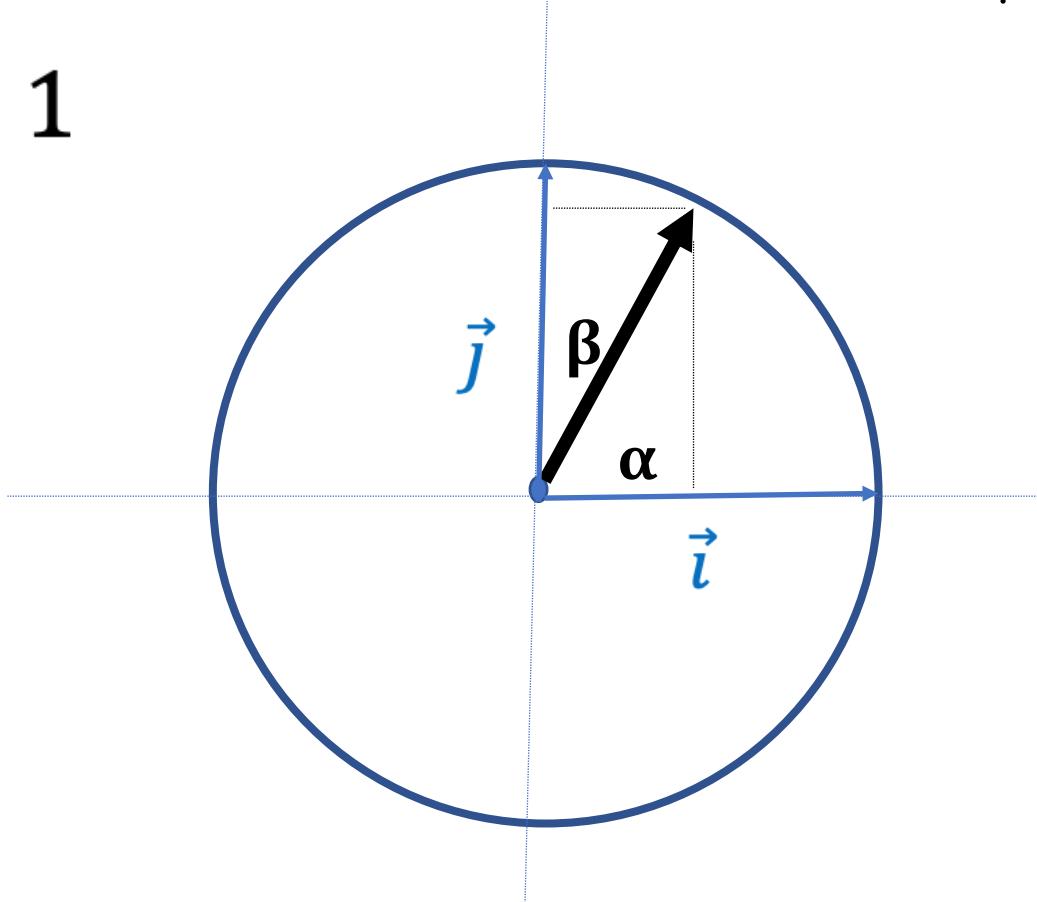


- 1 For any possible state: the measurement can only result in
 $|0\rangle$ or $|1\rangle$
- 2 Probability of measuring $|0\rangle$ is α^2 , probability of measuring $|1\rangle$ is β^2 . (1)
- 3 When the measure is done, the superposition is lost.

(1) α and β are complex numbers the probabilities are $|\alpha|^2 = \alpha\alpha^*$ and $|\beta|^2 = \beta\beta^*$

Quantum bits properties

$$\alpha^2 + \beta^2 = 1$$



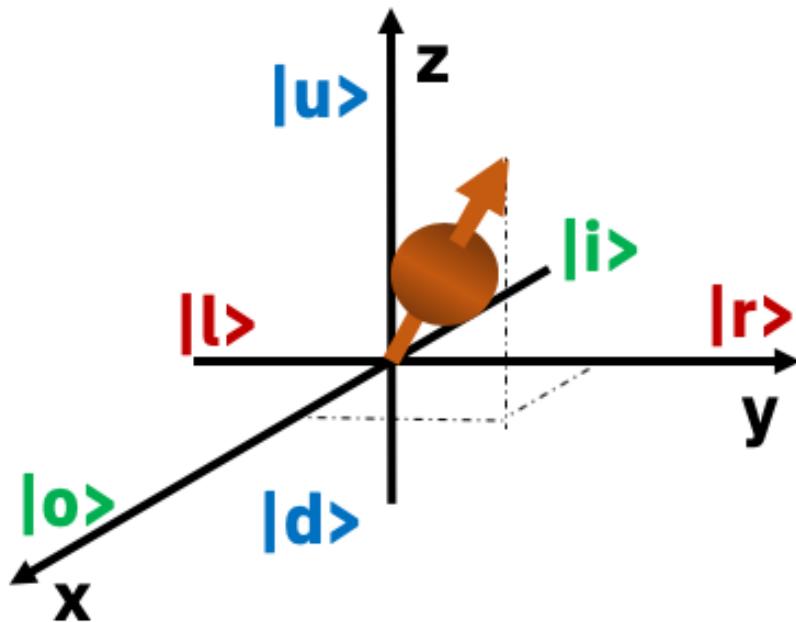
?

This model is not enough.

Let's consider the spin example (magnetic moment), that can be imagined as a particle rotation around itself.

With a measurement apparatus that can be positioned along 3 axes, the following states can be found :

$|up\rangle$, $|down\rangle$, $|left\rangle$, $|right\rangle$, $|in\rangle$, $|out\rangle$, (in short : $|u\rangle$, $|d\rangle$, $|l\rangle$, $|r\rangle$, $|i\rangle$, $|o\rangle$),



This model is not enough (...)

With the z axis as a base, the general state is : $|\psi\rangle = \alpha|u\rangle + \beta|d\rangle$

Then we can write $|r\rangle$ using this base as : $|\mathbf{r}\rangle = \alpha_r|u\rangle + \beta_r|d\rangle$ (with $\alpha_r^2 + \beta_r^2 = 1$),

And, as experience teaches us: $\alpha_r^2 = \beta_r^2 = \frac{1}{2}$

Similarly :

$$|\mathbf{l}\rangle = \alpha_l|u\rangle + \beta_l|d\rangle, \alpha_l^2 = \beta_l^2 = \frac{1}{2}$$

And because $|\mathbf{l}\rangle$ et $|\mathbf{r}\rangle$ are orthogonal : $\langle \mathbf{r} | \mathbf{l} \rangle = 0 : \alpha_r \alpha_l^* + \beta_r \beta_l^* = 0$

This is a valid solution (for $\alpha_r, \beta_r, \alpha_l, \beta_l$) :

$$|\mathbf{l}\rangle = \frac{1}{\sqrt{2}}|u\rangle + \frac{1}{\sqrt{2}}|d\rangle$$

$$|\mathbf{r}\rangle = \frac{1}{\sqrt{2}}|u\rangle - \frac{1}{\sqrt{2}}|d\rangle$$

This model is not enough (...)

Then for $|i\rangle$ and $|o\rangle$ we have:

$$|i\rangle = \alpha_i |u\rangle + \beta_i |d\rangle, \text{ and } \alpha_i^2 = \beta_i^2 = \frac{1}{2}$$

as well as :

$$|o\rangle = \alpha_o |u\rangle + \beta_o |d\rangle, \alpha_o^2 = \beta_o^2 = \frac{1}{2}$$

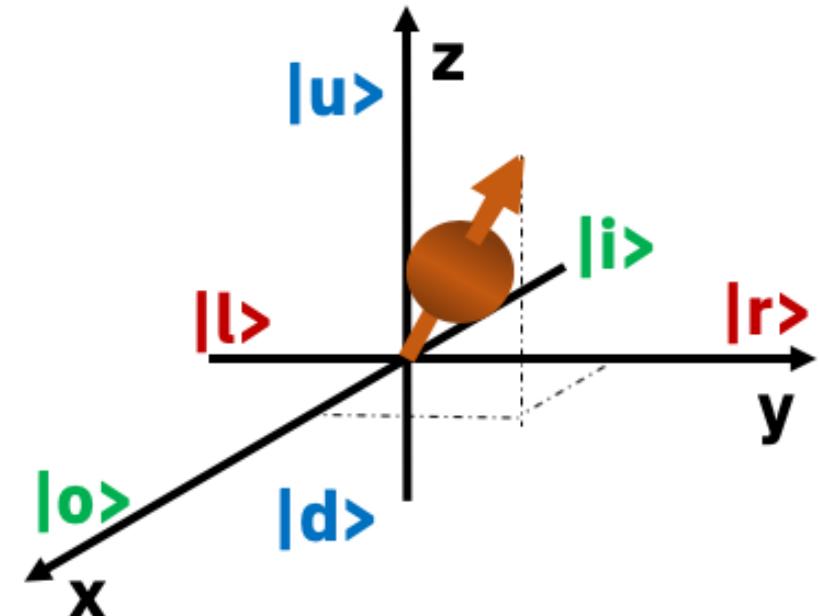
$$\text{with } \langle o|i\rangle = \langle i|o\rangle = 0 : \alpha_i \alpha_o^* + \beta_i \beta_o^* = \alpha_o \alpha_i^* + \beta_o \beta_i^* = 0$$

$$\text{and } \langle o|r\rangle = \langle o|i\rangle = \langle i|r\rangle = \langle i|i\rangle = \frac{1}{2},$$

This solution satisfies the equations above:

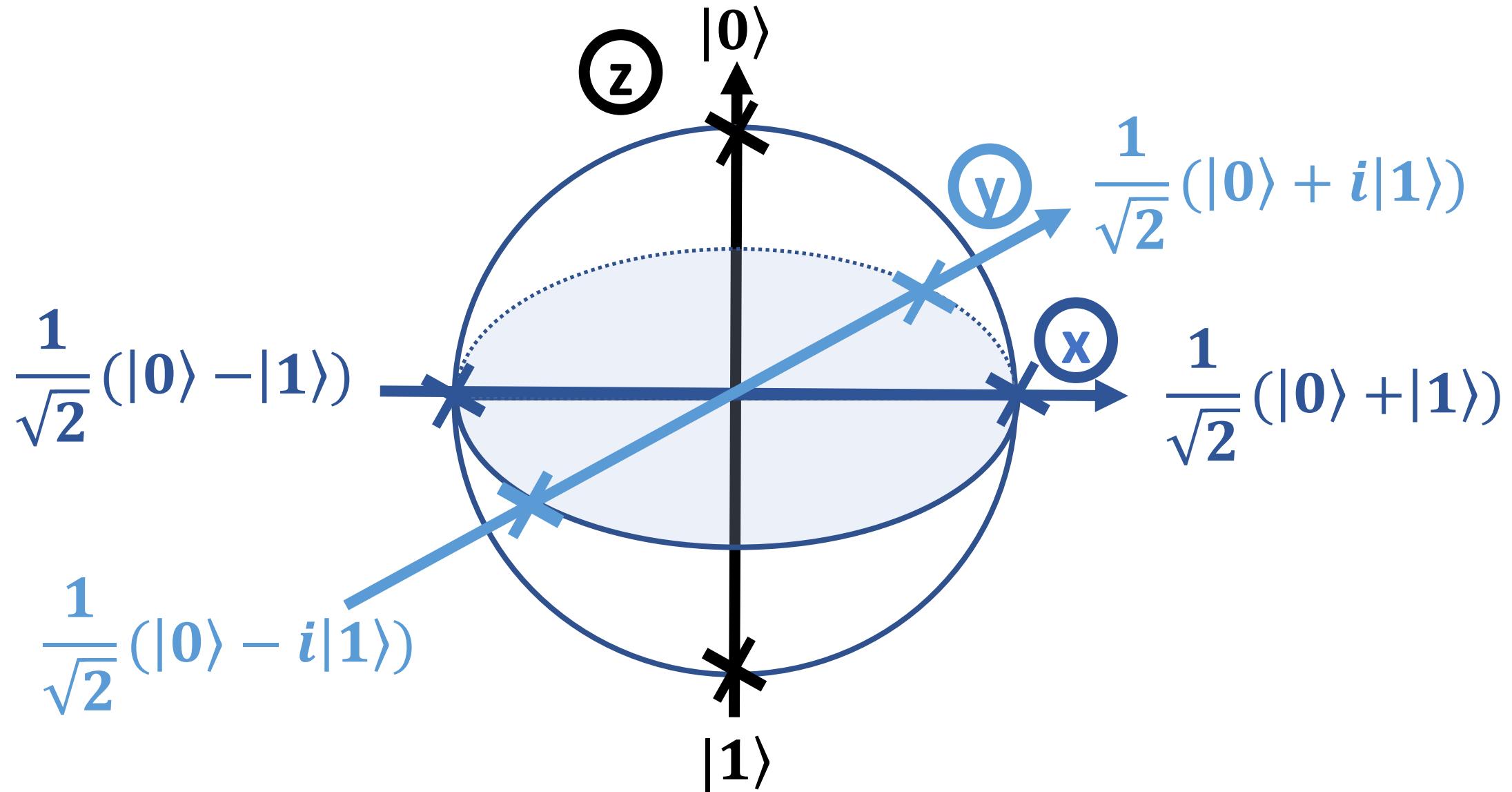
$$|i\rangle = \frac{1}{\sqrt{2}} |u\rangle + \frac{i}{\sqrt{2}} |d\rangle$$

$$|o\rangle = \frac{1}{\sqrt{2}} |u\rangle - \frac{i}{\sqrt{2}} |d\rangle$$



Now the model is usefull for our qubit !

The quantum state can be shown as follows:



Quantum bit : qubit

A **qubit** is a 2 dimensionnal vector space, with a basis $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (\equiv \overrightarrow{OM} = a\vec{i} + b\vec{j})$$

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \quad |\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and :

$$\langle\psi| = (\alpha^* \quad \beta^*)$$

note:

$$\langle\psi|\psi\rangle = |\psi|^2 = \alpha\alpha^* + \beta\beta^*$$

Explore a new era of IT with
Quantum Computing at IBM

IBM Client Center Montpellier

Q



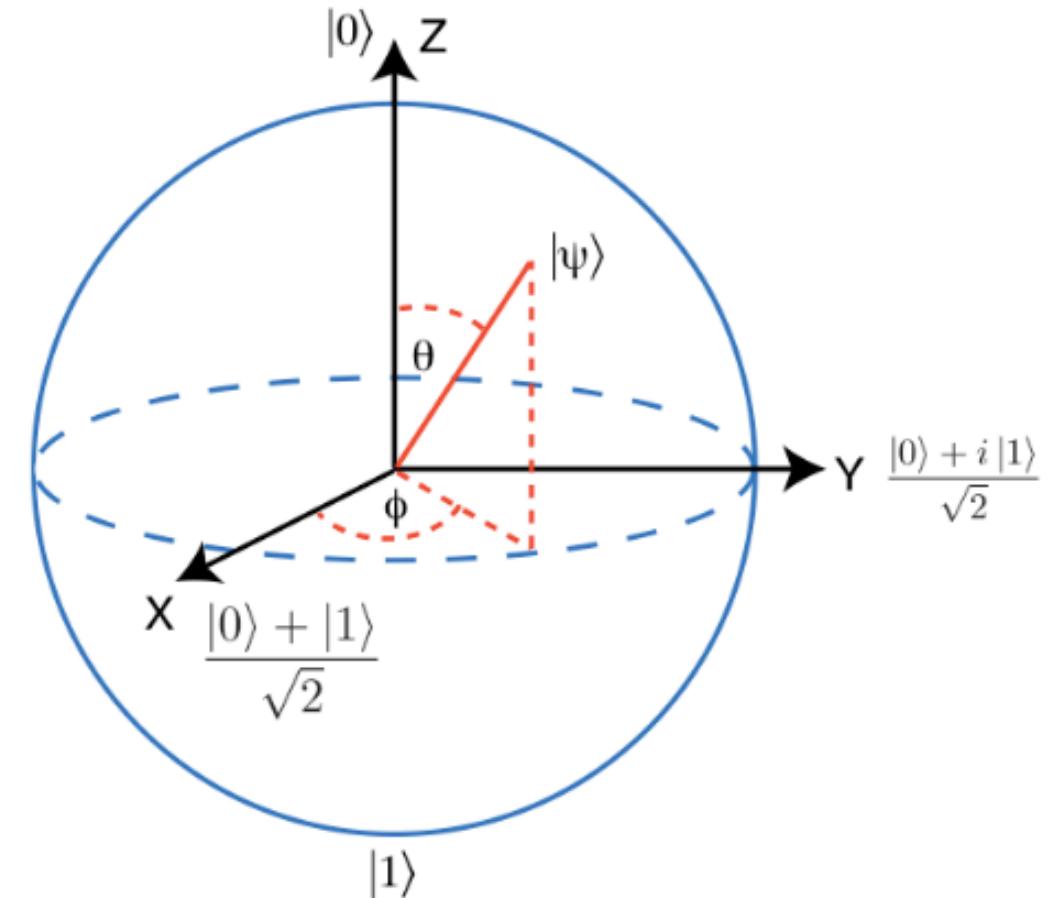
Controlling a qubit



The Bloch Sphere

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$



The Bloch sphere



Controlling a qubit

$$ID|0\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, \quad ID|1\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

$$SET|0\rangle = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \quad SET|1\rangle = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

$$UNSET|0\rangle = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle, \quad UNSET|1\rangle = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

$$NOT|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle, \quad NOT|1\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

Qubit : operators

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \text{identity}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} = \text{NOT because } |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ goes to } |1\rangle, \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ goes to } |0\rangle$$

some operations can't be done :

$$\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ \alpha + \beta \end{pmatrix}$$

but many other are possible! ...

Bit Flip– X gate, NOT

X

- Rotation around the x-axis by π (Bit-Flip ($0 \rightarrow 1$, $1 \rightarrow 0$)

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

$$X|0\rangle = |1\rangle$$

$$X|1\rangle = |0\rangle$$

\leftarrow Your circuit $\rightarrow rx$

Rx

Subroutine params
theta

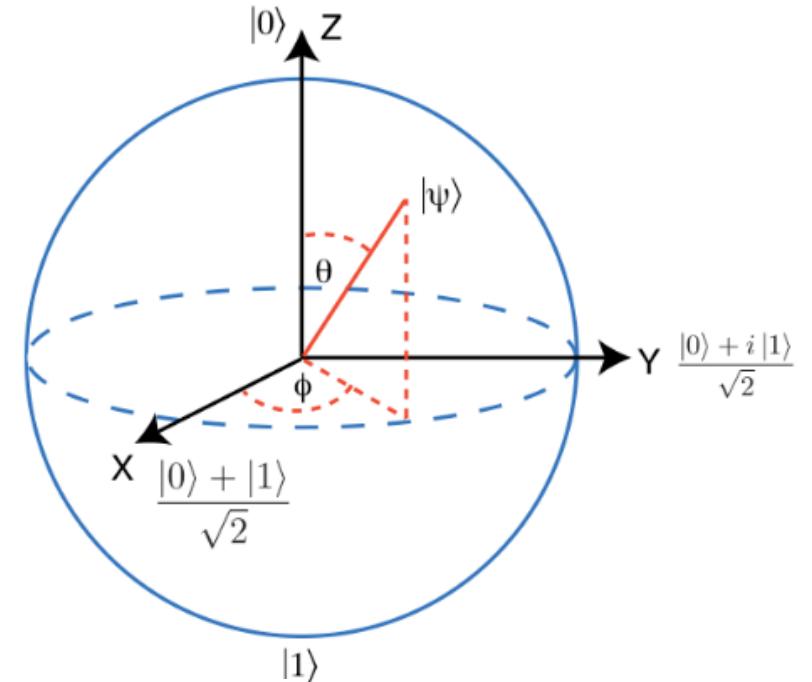
pi/2

Qubits connections

q[0]

a

≡
Rx



The Bloch sphere

Y gate

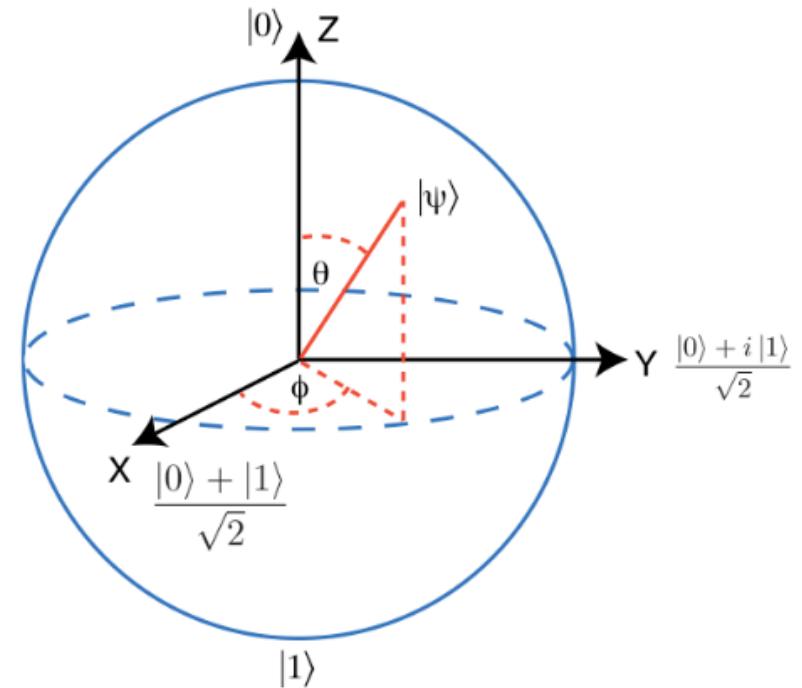
Y

- Rotation around the y-axis by π

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

$$Y|0\rangle = -i|0\rangle$$

$$Y|1\rangle = i|1\rangle$$



The Bloch sphere

Ry

← Your circuit → ry

Subroutine params
theta

pi/3

Qubits connections

q[0] a Ry

≡

Other gates : Z gate

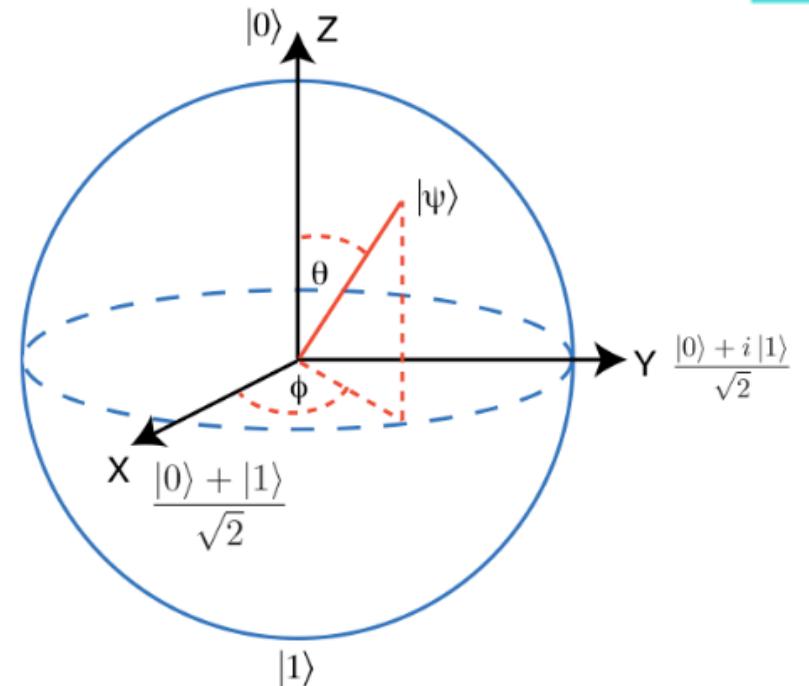
Z

- Rotation around the Z-axis by π

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

$$Z|0\rangle = |0\rangle$$

$$Z|1\rangle = -|1\rangle$$



The Bloch sphere



Summary

Rotation of π around the X, Y, Z axis

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

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

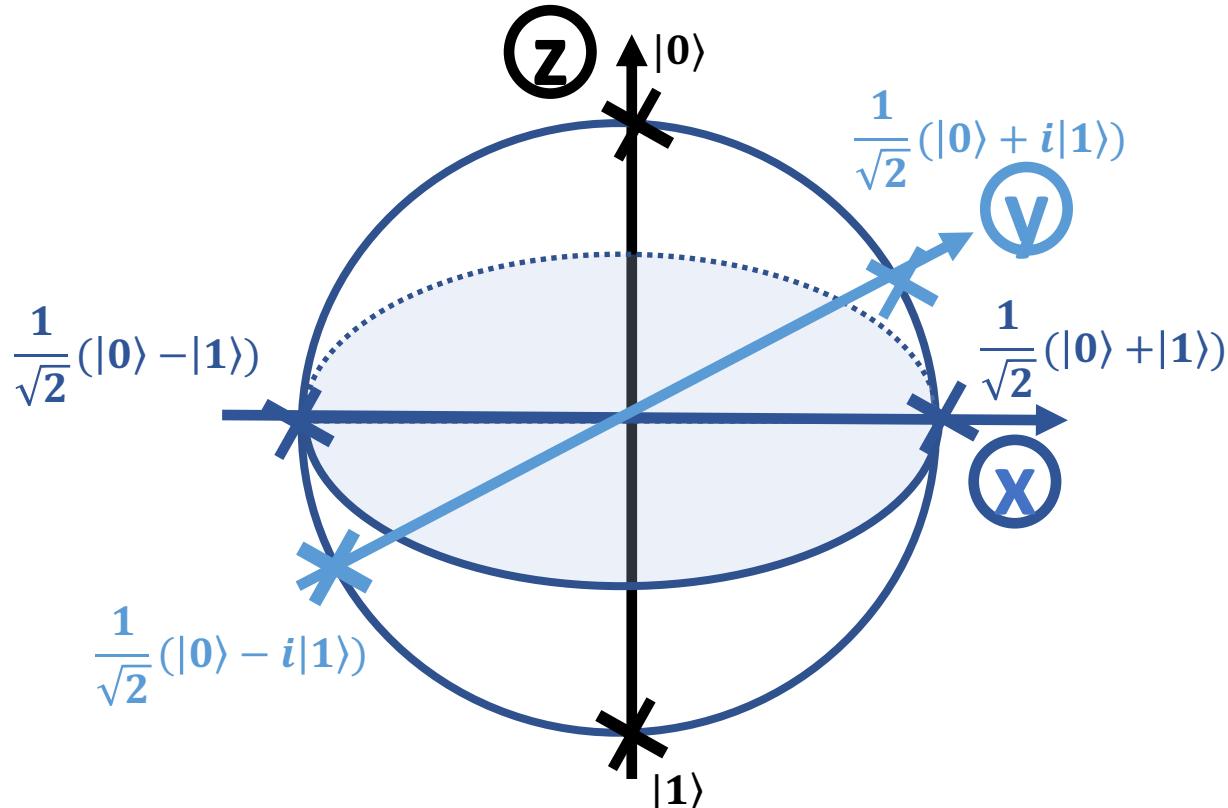
$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Propriétés:

$$X^2 = Y^2 = Z^2 = I$$

$$XY = iZ ; ZX = iY ; YZ = iX$$

$$XY = -YX ; YZ = -ZY ; XZ = -ZX$$



and many more ...

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

$$R_X(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad R_Y(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i\lambda+i\phi} \cos \frac{\theta}{2} \end{pmatrix}$$

$$U_2(\phi, \lambda) = U_3(\frac{\pi}{2}, \phi, \lambda) = \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i\lambda+i\phi} \end{pmatrix}$$

$$U_1(\lambda) = U_3(0, 0, \lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$$

U3 has the effect of rotating a qubit in the initial $|0\rangle$ state to one with an arbitrary superposition and relative phase

From this gate, the Hadamard is done by $H=U2(0,\pi)$

The U1 gate is known as the phase gate and is essentially the same as $Rz(\lambda)$.

Creating Superposition – Hadamard gate

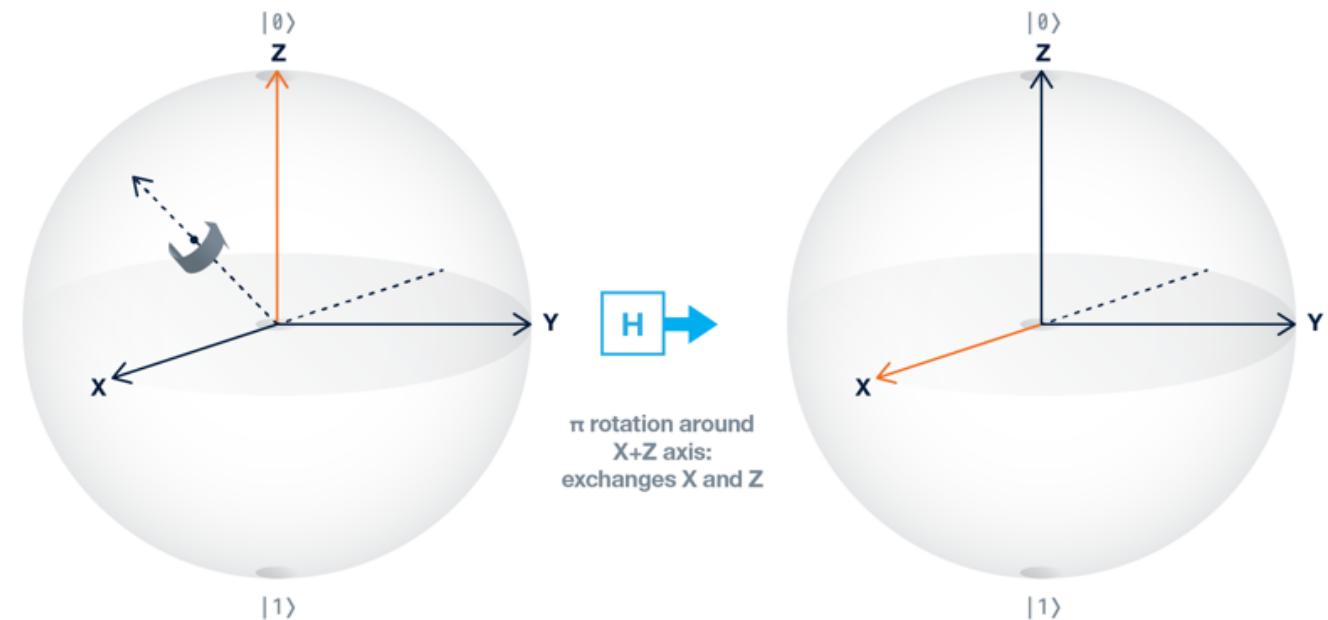
H

Produces equal-weighted combination of the states $|0\rangle$ and $|1\rangle$

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

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

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

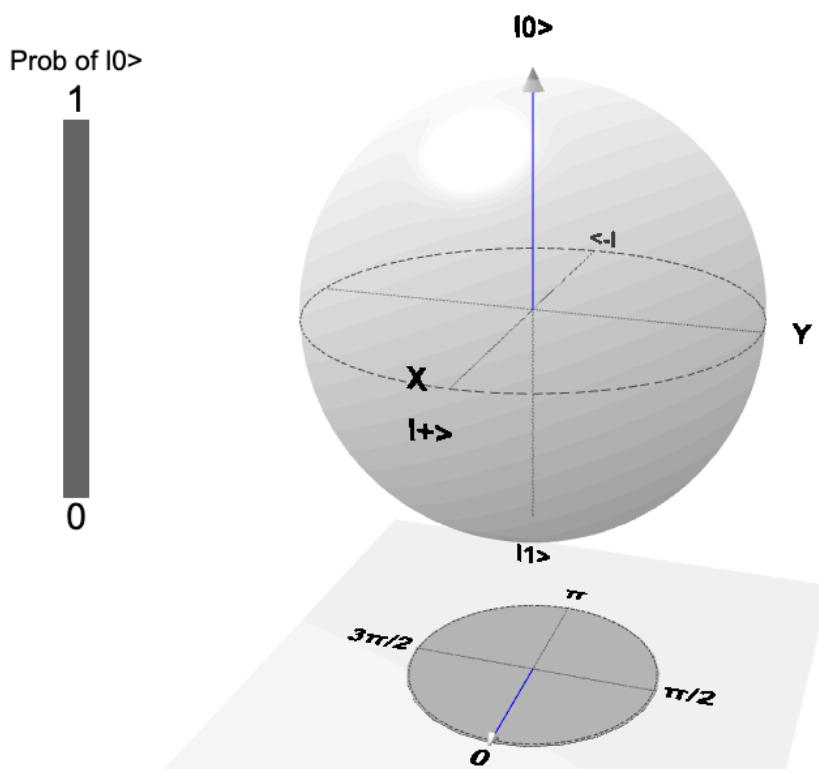


Superposition is one of the two crucial properties giving quantum computers their special ability

Moving around the Bloch Sphere

<https://javafxpert.github.io/grok-bloch/>

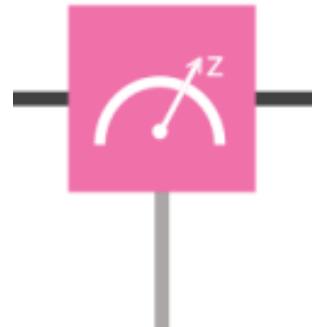
$$|\psi\rangle = \sqrt{1.00}|0\rangle + (\sqrt{0.00})e^{i\theta}|1\rangle$$



X	S
Y	S^\dagger
Z	T
H	T^\dagger
$R_x \frac{\pi}{12}^+$	$R_x \frac{\pi}{12}^-$
$R_y \frac{\pi}{12}^+$	$R_y \frac{\pi}{12}^-$
$R_z \frac{\pi}{12}^+$	$R_z \frac{\pi}{12}^-$
$ 0\rangle$	$ 1\rangle$

Measurement operation

Quantum measurement



$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$

If a qubit in some state $|\psi\rangle$ is measured in the standard basis, the **result 0 is obtained with probability $|\alpha|^2$** , and the **result 1 is obtained with the complementary probability $|\beta|^2$** .

Explore a new era of IT with
Quantum Computing at IBM

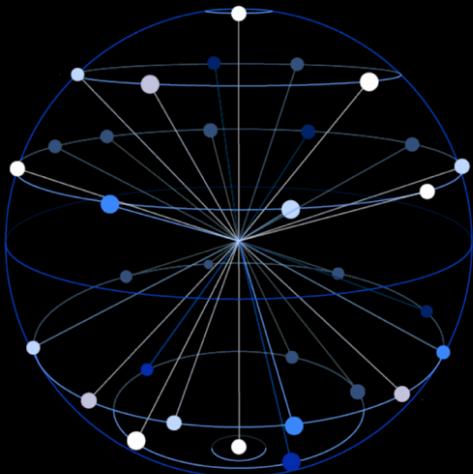
IBM Client Center Montpellier

Q



Working with many qubits

Universal quantum computers leverage quantum mechanical properties of superposition and entanglement to create states that scale exponentially with number of qubits, or quantum bits.

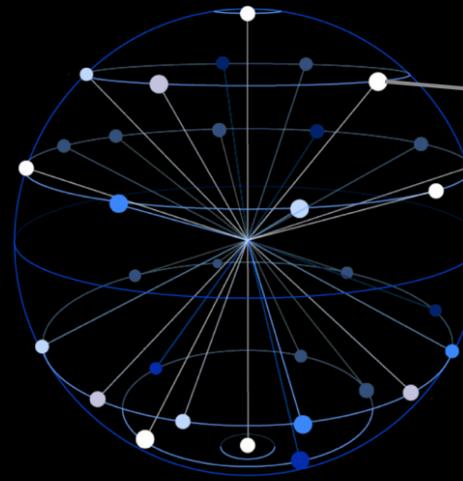


Superposition

A single quantum bit can exist in a superposition of 0 and 1:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

and N qubits allow for a superposition of all possible 2^N combinations.



Entanglement

The states of entangled qubits cannot be described independently of each other.

Two qubits system:

qubit A is defined in its states space : $|\psi_A\rangle = \alpha|0_A\rangle + \beta|1_A\rangle$

qubit B is defined in its states space : $|\psi_B\rangle = \gamma|0_B\rangle + \delta|1_B\rangle$

The two qubits system (A & B), is described by the product of the two single states :

$$|\Psi_{AB}\rangle = |\psi_A\rangle |\psi_B\rangle = (\alpha|0_A\rangle + \beta|1_A\rangle) \times (\gamma|0_B\rangle + \delta|1_B\rangle)$$

$$|\Psi\rangle = \alpha\gamma|0_A\rangle|0_B\rangle + \alpha\delta|0_A\rangle|1_B\rangle + \beta\gamma|1_A\rangle|0_B\rangle + \beta\delta|1_A\rangle|1_B\rangle$$

$$|\Psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

Two qubits system:

$$|\Psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

The two qubit system is defined in a 4 dimension state, with basis vectors: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$

General state is :

$$|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \text{ et } |a|^2 + |b|^2 + |c|^2 + |d|^2 = 1$$

for 3 qubits :

$$|\psi\rangle = a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$$

(with sum of squared components equals 1)

And so on for 4 qubits..., n qubits

Two qubit states calculations (tensor product)

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2 \\ \alpha_1\beta_2 \\ \beta_1\alpha_2 \\ \beta_1\beta_2 \end{pmatrix}$$

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

$$|0\rangle \otimes |1\rangle = |01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle \otimes |0\rangle = |10\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

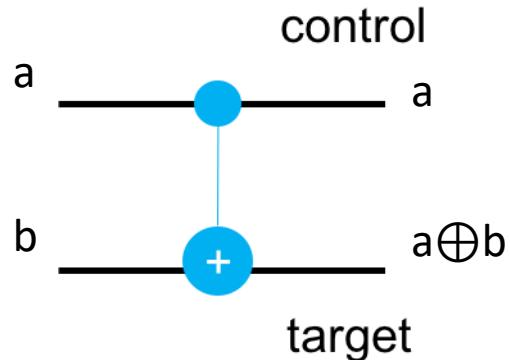
...

And so on:

$$\begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \otimes \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \otimes \begin{pmatrix} \alpha_3 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} \alpha_1\alpha_2\alpha_3 \\ \alpha_1\alpha_2\beta_3 \\ \alpha_1\beta_2\alpha_3 \\ \alpha_1\beta_2\beta_3 \\ \beta_1\alpha_2\alpha_3 \\ \beta_1\alpha_2\beta_3 \\ \beta_1\beta_2\alpha_3 \\ \beta_1\beta_2\beta_3 \end{pmatrix},$$

$$\text{eg: } |101\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Multi-qubits Gates : CNOT gate



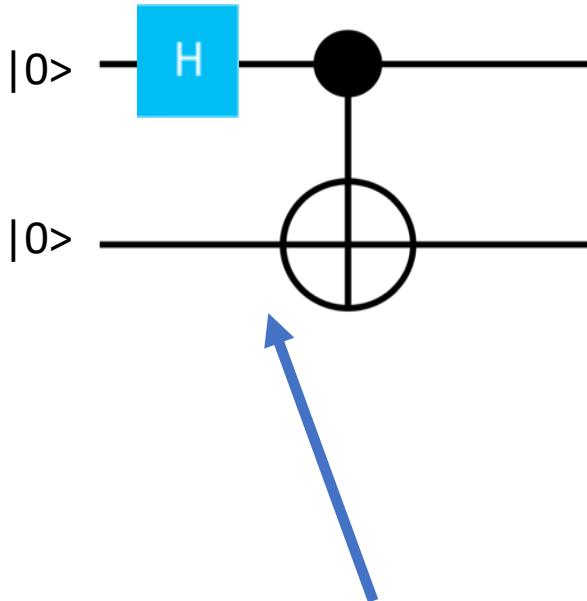
- CNOT = Controlled-NOT Gate
- Inverts a target qubit according to the state of the control qubit

CNOT		
a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

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

“Hello World!”

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

entanglement

$$|Bell\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

If $|Bell\rangle$ could be the product of two quantum states :

$$|\psi\rangle = a|0\rangle + (b + ic)|1\rangle \quad \text{et} \quad |\varphi\rangle = d|0\rangle + (e + if)|1\rangle$$

Then :

$$|\psi\rangle|\varphi\rangle = ad|00\rangle + ae + iaf|01\rangle + (db + idc)|10\rangle + (b + ic)(e + if)|11\rangle$$

one can identify the values along the basis vectors :

$$(1) ad = \frac{1}{\sqrt{2}}$$

$$(2) ae + iaf = 0$$

$$(3) db + idc = 0$$

$$(4) be - cf + i(bf + ce) = \frac{1}{\sqrt{2}}$$

entanglement

$$(1) ad = \frac{1}{\sqrt{2}}$$

$$(2) ae + iaf = 0$$

$$(3) db + idc = 0$$

$$(4) be - cf + i(bf + ce) = \frac{1}{\sqrt{2}}$$

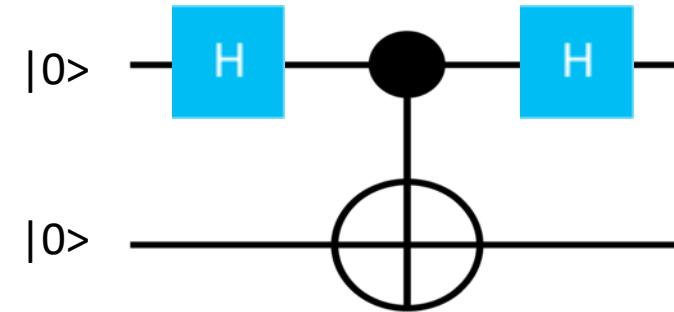
from (1) : ad is not 0, then both a and d are not null, then, from (2) & (3) we get $e = b = 0$ et $f = c = 0$

Then $be - cf = 0$ which does not match with (4)

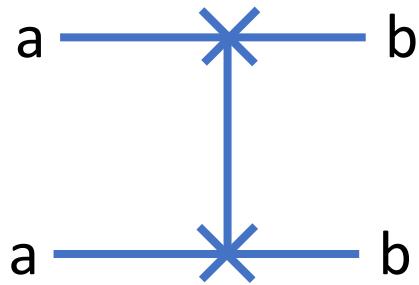
Bell State is not the product of two valid quantum states.

This means we can't know anything of one component of the entangled system, only the whole system can be studied.

trying to understand a bit more...



Multi-qubits Gates : SWAP gate

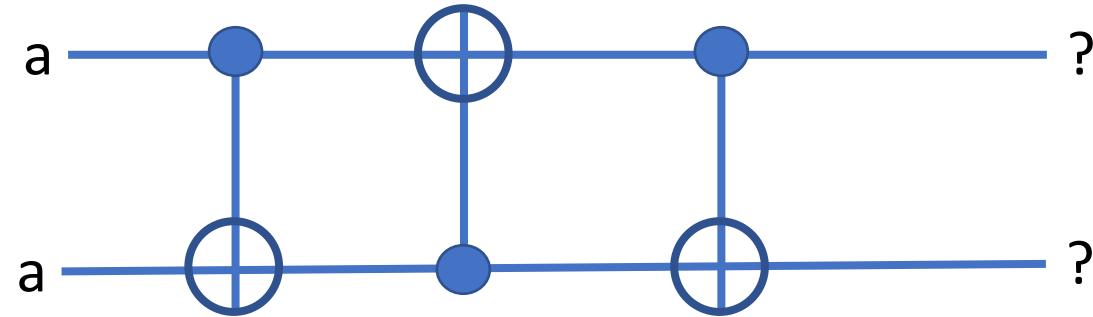


- Swappe qubit a state with qubit b state

SWAP			
a	b	a	b
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	1

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

What is this circuit result ?



SWAPPING is not CLONING !

**Non cloning theorem : a quantum state cannot be copied
(Wootters & Zurech, 1982)**

Assume there exists an unitary operator such that for any state $|\Psi\rangle$:

$$(a) U(|\Psi\rangle \otimes |\Phi\rangle) = |\Psi\rangle \otimes |\Psi\rangle$$

(this would copy the state $|\Psi\rangle$ onto $|\Phi\rangle$). Let α and $|\Omega\rangle$ such that $|\Omega\rangle = \alpha|\Psi\rangle$, let's write again (a) with $|\Omega\rangle$ this time :

$$U(|\Omega\rangle \otimes |\Phi\rangle) = |\Omega\rangle \otimes |\Omega\rangle$$

Replacing $|\Omega\rangle$ with $\alpha|\Psi\rangle$:

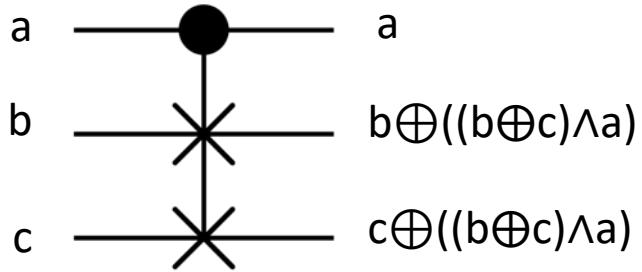
$$U(|\Omega\rangle \otimes |\Phi\rangle) = \alpha|\Psi\rangle \otimes \alpha|\Psi\rangle = \alpha^2|\Psi\rangle \otimes |\Psi\rangle$$

We can also write: $U(|\Omega\rangle \otimes |\Phi\rangle) = U(\alpha|\Psi\rangle \otimes |\Phi\rangle)$

And with linearity of U : $U(|\Omega\rangle \otimes |\Phi\rangle) = \alpha U(|\Psi\rangle \otimes |\Phi\rangle) = \alpha|\Psi\rangle \otimes |\Psi\rangle$

And we find that $\alpha^2 = \alpha$, which is not generally true.

Multi-qubits Gates : CSWAP (Fredkin gate)



- CSWAP = Controlled-Swap Gate
- Swaps two target qubits according to the state of the control qubit

CSWAP					
a	b	c	(b \oplus c) \wedge a	b \oplus ((b \oplus c) \wedge a)	c \oplus ((b \oplus c) \wedge a)
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	0	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	0	1	1

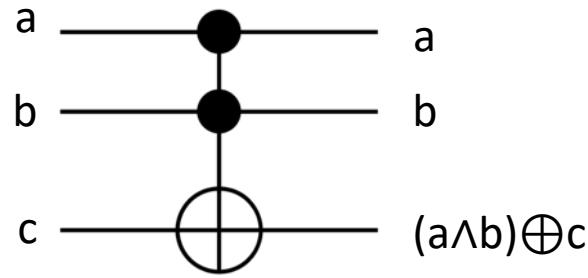
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Basis: |000>, |001>, |010>, |011>, |100>, |101>, |110>, |111>

Note:

- If $c==0$: $c\oplus((b\oplus c)\wedge a) = a \wedge b$
- If $c==1$: $b\oplus((b\oplus c)\wedge a) = a \vee b$
- If $b==0 \& c==1$: $c\oplus((b\oplus c)\wedge a) = \neg a$

Multi-qubits Gates : CCNOT (Toffoli gate)



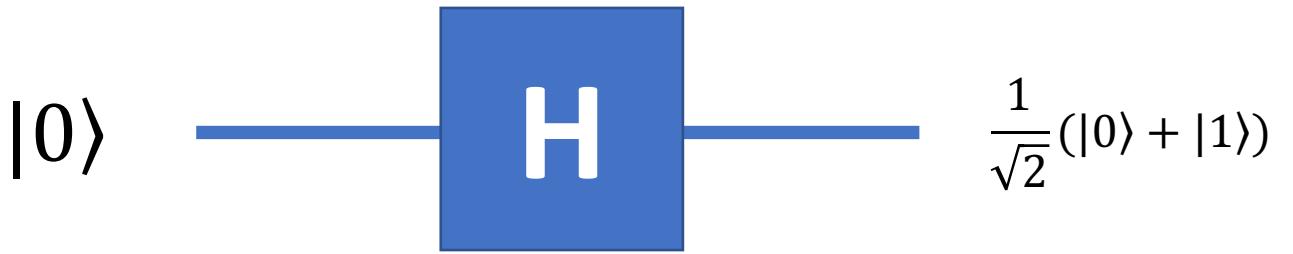
- CCNOT = Control-Control-NOT Gate
- Inverts a target qubit according to the state of the two control qubits

CCNOT			
a	b	c	$(a \wedge b) \oplus c$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

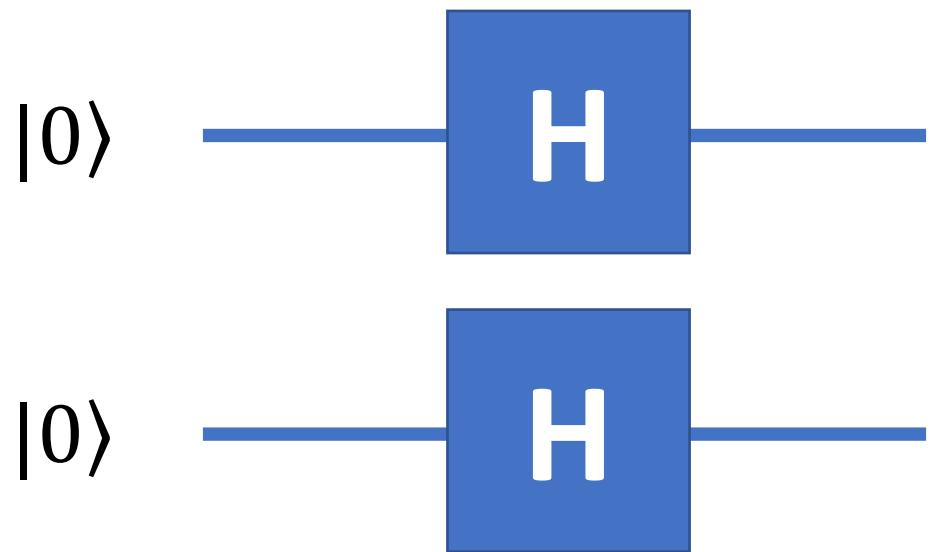
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Basis: $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$

$\mathbb{H}^{\otimes n}$

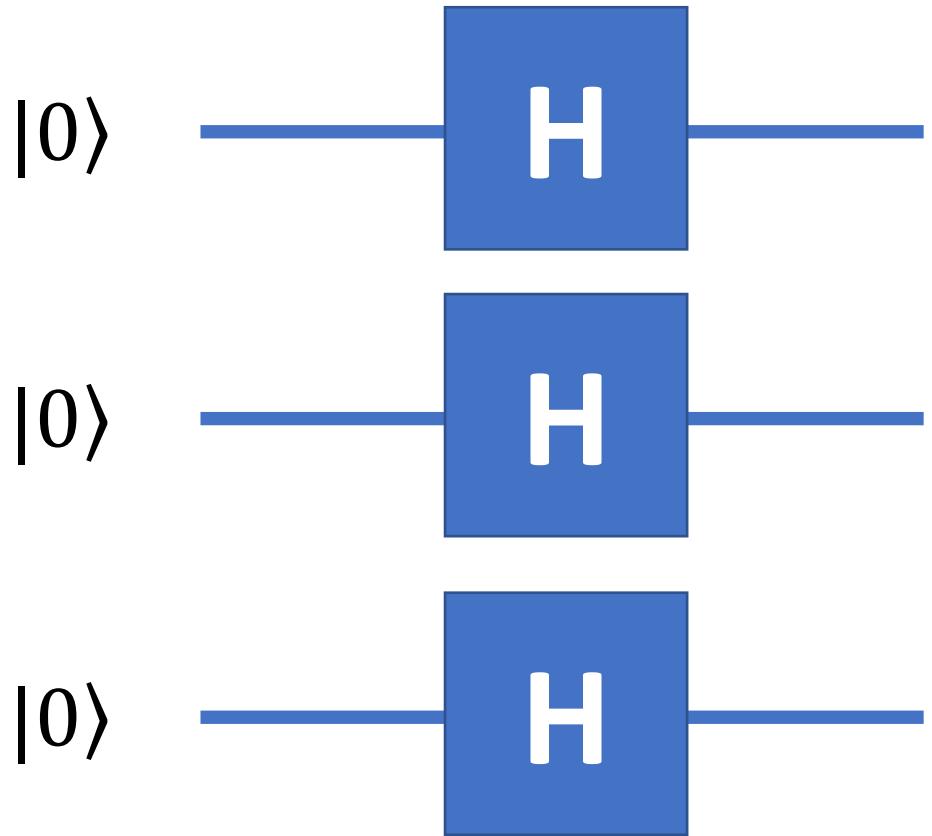


$\mathbb{H}^{\otimes n}$



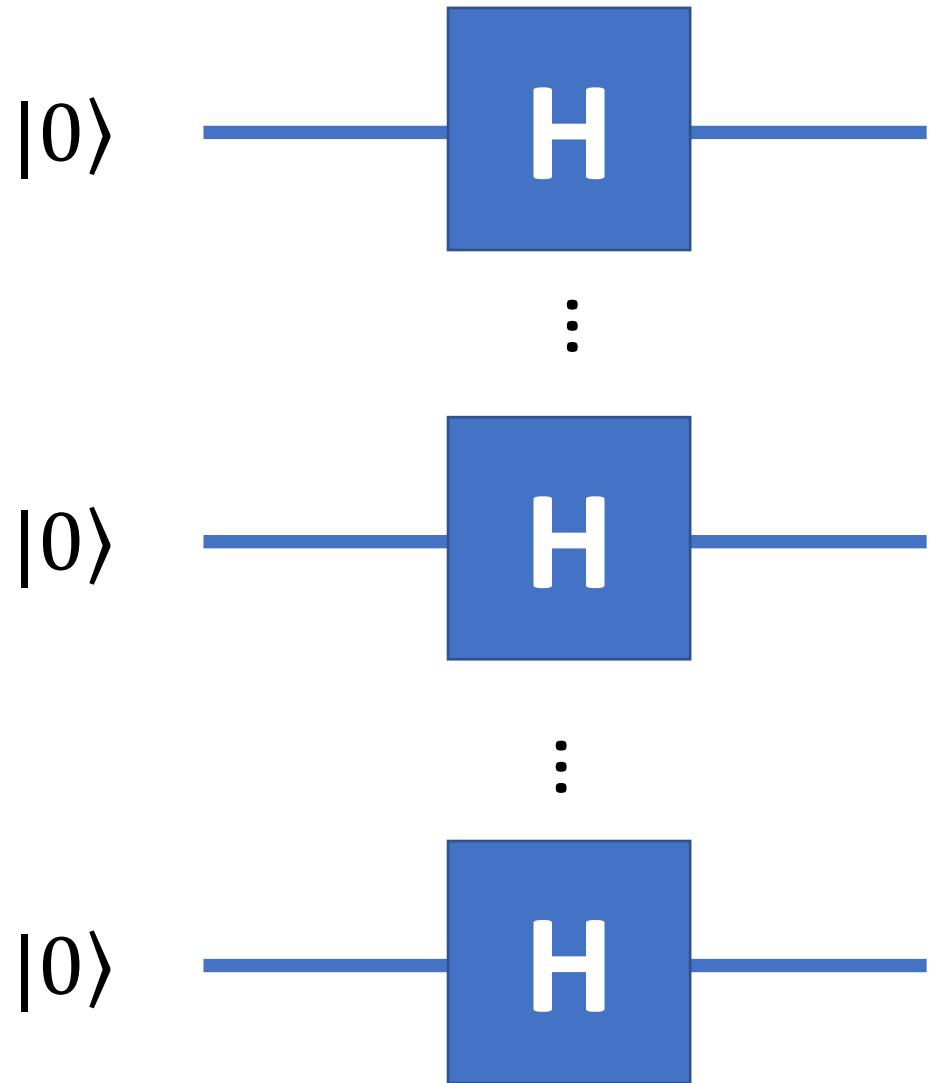
$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$\mathbb{H}^{\otimes n}$



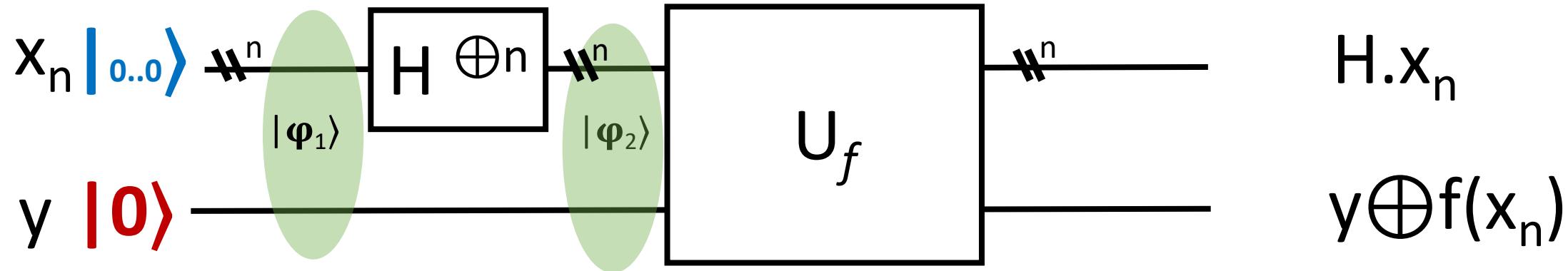
$$\frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$$

\mathbb{H}^n



$$\begin{aligned} & \frac{1}{\sqrt{2^n}}(|0\dots0\rangle + |0\dots1\rangle + \dots \\ & + |1\dots10\rangle + |1\dots11\rangle) \end{aligned}$$

“Parallelism”



$$|\Phi_1\rangle = |0..0,0\rangle$$

$$|\Phi_2\rangle = \frac{1}{\sqrt{2^n}}(|0..0,0\rangle + \dots + |1..1,0\rangle)$$

Output state $|\Phi_3\rangle$ contains all results of f applied to all possible values for x_n , namely : $f(0..0)$, $f(0..1)$, ... and $f(1..1)$, while U_f has been calculated only once.

Doing something usefull with that is another story... and that is all about quantum algorithms...

Explore a new era of IT with
Quantum Computing at IBM

IBM Client Center Montpellier

Q



Exercises

What is \mathcal{O} for a naive bubble sort algorithm ?

a) Calculate $H \times H$ with matrices products

b) Calculate $H \times H$ using : $H = X + Z$ and Pauli identities

Create a circuit of 4 qubits (a,b,c,d):

swap b with d if $a==0$

swap c with d if $a == 1$

leave a unchanged at the end

Create a circuit of 4 qubits (a,b,c,d):

swap c with d if ($a==1$ and $b==1$), you may use an additionnal qubit e (starts and finishes at $|0\rangle$)