

Cet événement est scientifique et financièrement supporté par :



ECOLE CONJOINTE GT IQ, GT2L et EU/ME

Approches Quantiques pour une nouvelle recherche opérationnelle !?

Montpellier du 2-5 Novembre 2021

Your first steps into IBM Quantum Computing

IBM Client Center Montpellier

Léna Pérennès | lena.perennes@ibm.com

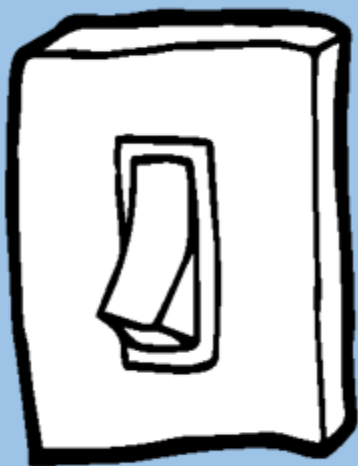
Jean-Michel Torres | torresjm@fr.ibm.com

Nov 2nd 2021

Part 1

**Guided tour of the IBM Quantum devices,
and Quantum « Hello World! »**

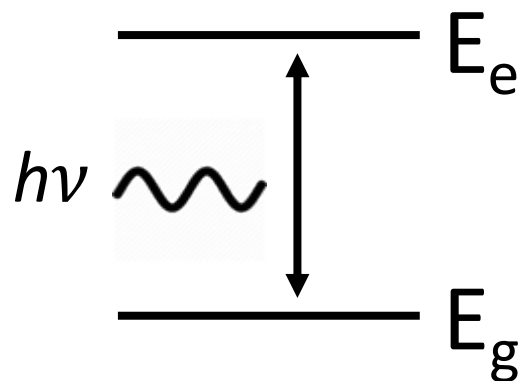
0



1

classical bit

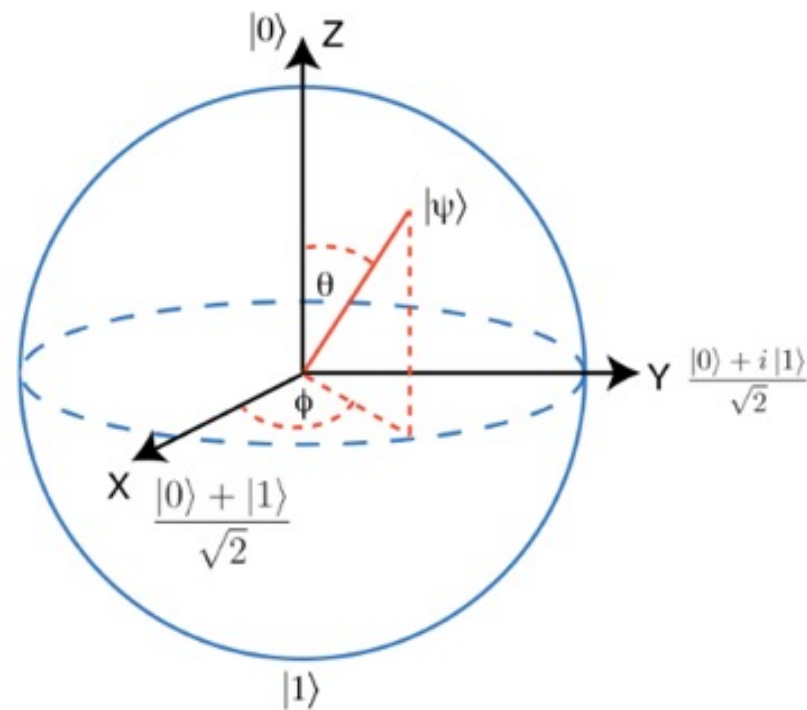
qubit : quantum bit



$$|e\rangle \sim |1\rangle$$







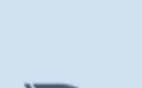
$$|g\rangle \sim |0\rangle$$

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










The Bloch sphere

Controlling a qubit

NOT	
Buffer	
AND	
NAND	
OR	
NOR	
XOR	

« PAULI » Operators

rotation around x axis		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	qc.x(qr[n])		$\begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$
rotation around y axis		$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	qc.y(qr[n])		$\begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$
rotation around z axis		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	qc.z(qr[n])		$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$
Identity		$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	qc.id(qr[n])		

superposition

(X+Z)

Hadamard gate

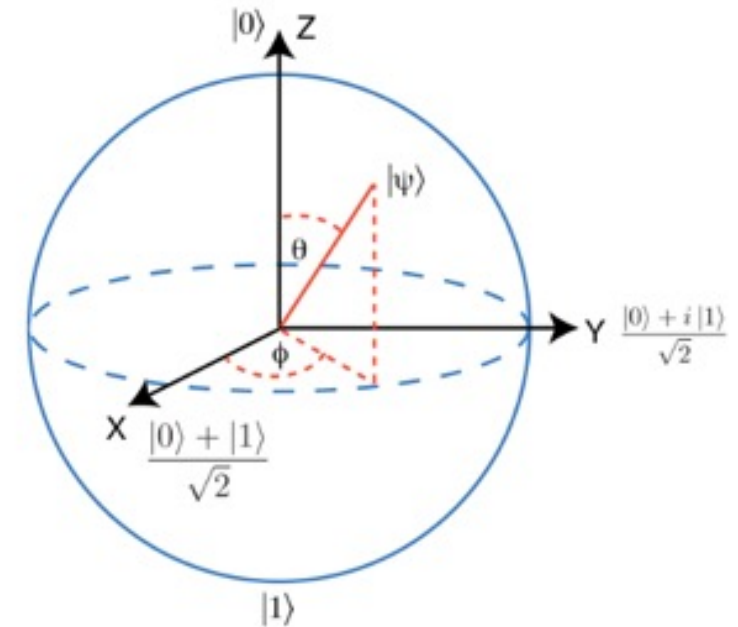


$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ qc.h(qr[n])}$$

More operators are available from qiskit (S, T, swap, cswap, ccx, cz, ...)

Bloch Sphere

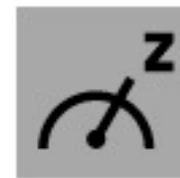
$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$$






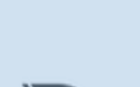


CNOT : flips target qubit according to control qubit state.

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

measurement measures quantum state in quantum register into classical register (0/1)



NOT	
Buffer	
AND	
NAND	
OR	
NOR	
XOR	

**classical
operators**

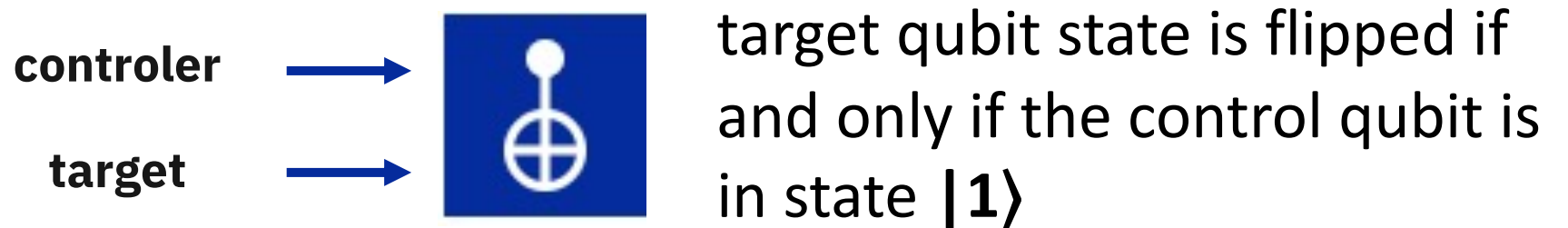
quantum operators :

H operator (Hadamard)

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

creates equal superposition of states $|0\rangle$ and $|1\rangle$

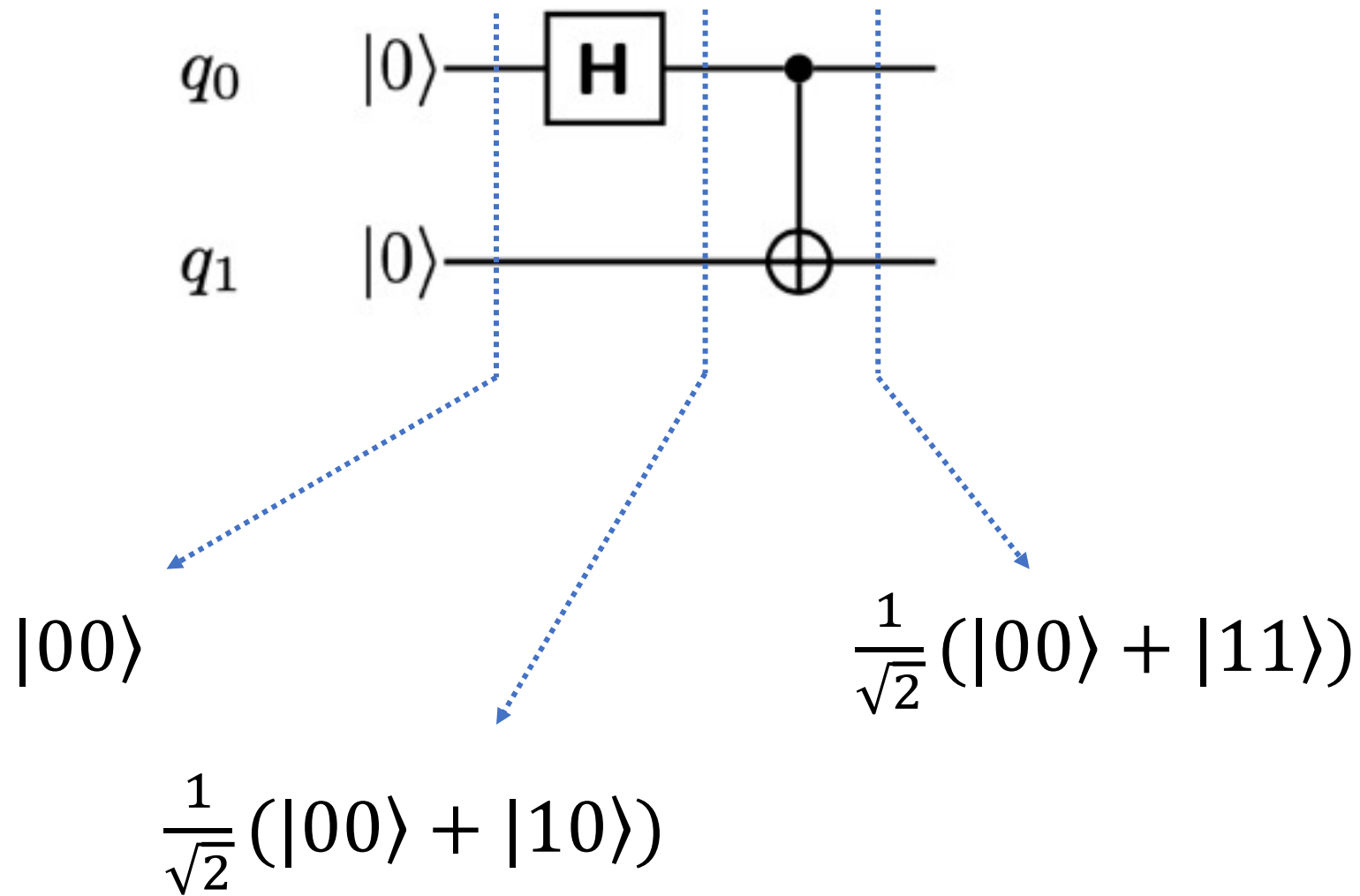
Control-Not operation



creates quantum entanglement of two qubits

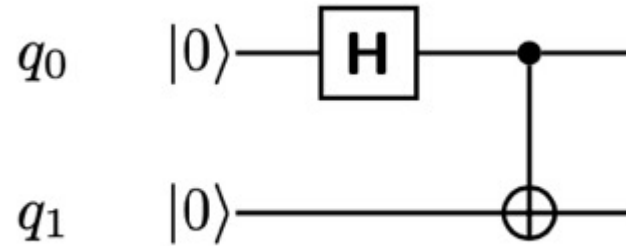
```
print('Hello World!')
```

Hello World!



Hello World! example

Hadamard gate applied to q_0 ,
then Control-Not applied to
 q_1 , controlled by q_0



This produces the
« Bell-State »

With words :

System starts in $|00\rangle$ (both q_0 and q_1 in state $|0\rangle$).

Then q_0 goes through Hadamard and gets into equal superposition of $|0\rangle$ and $|1\rangle$.

After q_0 controls q_1 , the state of q_1 is in a superposition of $|0\rangle$ & $|1\rangle$, (q_1 stays at $|0\rangle$ when q_0 is $|0\rangle$, and q_1 goes $|1\rangle$ when q_0 is $|1\rangle$).

So : both q_0 and q_1 are in $|0\rangle$ (state $|00\rangle$) or both q_0 and q_1 are in $|1\rangle$ (state $|11\rangle$).

Our system is in equal superposition of $|00\rangle$ and $|11\rangle$.

The two qubits are entangled: if you measure one of the qubits, you immediately know the state of the other.

In between :

System starts in $|00\rangle$, then :

$$H|00\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle)$$

Applying CNOT: left part of the sum stays as is, right term goes to $|11\rangle$ resulting state is $\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$.

One can easily prove there are no $\alpha, \beta, \gamma, \delta$ such that:

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

So, the resulting state is not the product of two quantum states, instead this is an entangled state.

With maths :

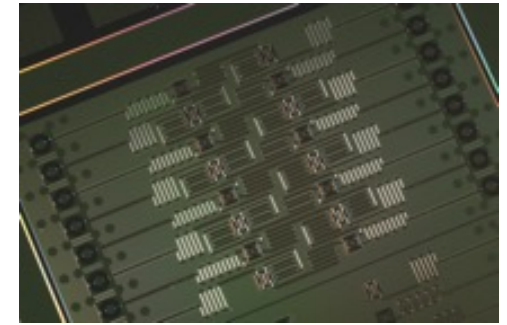
Stage 1 (H on q_0) :

$$(H \otimes I)|00\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Stage 2: CNOT(0,1)

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

Quantum Circuit



Circuits / Untitled circuit *Saved*

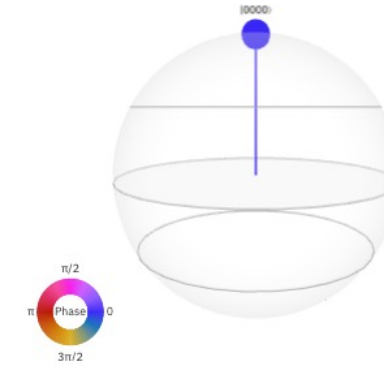
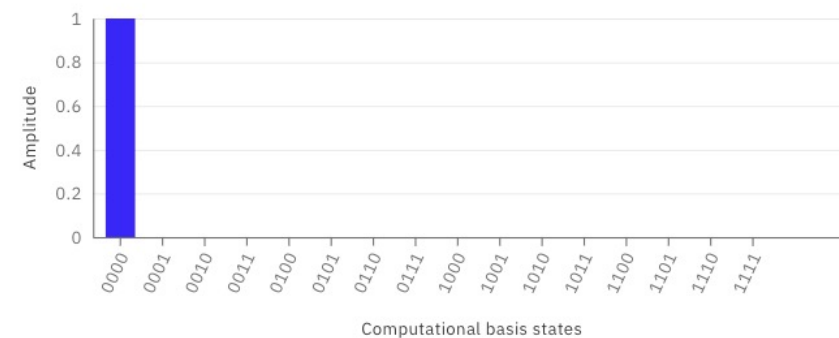
H \oplus \otimes \otimes \otimes \otimes I T S Z T^\dagger S^\dagger P RZ $|0\rangle$ \curvearrowright^z if \vdots \sqrt{X} \sqrt{X}^\dagger Y RX RY U RXX RZZ + Add



Statevector \vee

\textcircled{i} \vdots

Q-sphere \vee




```
print('Hello World!')
```

Demo : Bell state on a quantum machine

Part 2

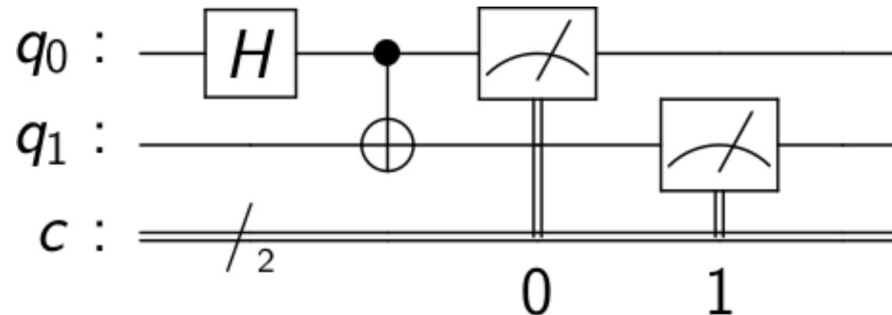
**using qiskit library to run quantum
program with Python.**

```
print('Hello World!')
```

Programing

```
In [1]: 1 from qiskit import QuantumCircuit, Aer, execute           # imports
        2 backend = Aer.get_backend('qasm_simulator')             # select a device for execution
        3
        4 qc = QuantumCircuit(2,2)                                # create a quantum circuit having 2 qubits and 2 cbits
        5
        6 qc.h(0)                                                    # build the circuit by
        7 qc.cx(0,1)                                                # adding operators on qubits
        8
        9 qc.measure([0,1],[0,1])                                  # use measurement gates to retrieve results
       10
       11 d = execute(qc,backend).result().get_counts()            # execute qc on backend and get cumulated results into
       12 print(d)                                                 # a dictionary
```

```
{'00': 491, '11': 533}
```



print('Hello World!')

Historic Quantum Algorithms

Deutsch	1985	$2 \rightarrow 1$
Bernstein-Vazirani	1992	$N \rightarrow 1$
Deutsch-Josza	1992	$2^{N-1} + 1 \rightarrow 1$
Shor	1994	$e^N \rightarrow (\text{Log}N)^3$
Grover	1996	$N \rightarrow \sqrt{N}$

More and new ones on quantumalgorithmzoo.org/