Your first steps into IBM Quantum Computing

IBM Client Center Montpellier
Jean-Michel Torres | torresjm@fr.ibm.com
Nov 16nd 2021
Part 1

Guided tour of the IBM Quantum devices,

and Quantum « Hello World! »
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

« PAULI » Operators

<table>
<thead>
<tr>
<th>Operation</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>rotation around x axis</td>
<td>$\begin{pmatrix} 0 &amp; 1 \ 1 &amp; 0 \end{pmatrix}$ qc. x(qr[n])</td>
</tr>
<tr>
<td>rotation around y axis</td>
<td>$\begin{pmatrix} 0 &amp; -i \ i &amp; 0 \end{pmatrix}$ qc. y(qr[n])</td>
</tr>
<tr>
<td>rotation around z axis</td>
<td>$\begin{pmatrix} 1 &amp; 0 \ 0 &amp; -1 \end{pmatrix}$ qc. z(qr[n])</td>
</tr>
<tr>
<td>Identity</td>
<td>$\begin{pmatrix} 1 &amp; 0 \ 0 &amp; 1 \end{pmatrix}$ qc. id(qr[n])</td>
</tr>
</tbody>
</table>

superposition (X+Z) Hadamard gate

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

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

Bloch Sphere

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

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

© 2021 IBM Corporation
quantum operators:

H operator (Hadamard)

\[ |0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \]

creates equal superposition of states \( |0\rangle \) and \( |1\rangle \)

Control-Not operation

controller \[\rightarrow\] target qubit state is flipped if and only if the control qubit is in state \( |1\rangle \)

creates quantum entanglement of two qubits
Hello World!

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

![Quantum Circuit Diagram]

\[ |00\rangle \quad \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \]

\[ \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \]
Hello World! example

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

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 q0):
$$\left( \frac{1}{\sqrt{2}} \begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{array} \right) \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right) = \frac{1}{\sqrt{2}} \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right)$$
Stage 2: CNOT(0,1)
$$\left( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right) \times \frac{1}{\sqrt{2}} \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right) = \frac{1}{\sqrt{2}} \left( \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right)$$
$$= \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

This produces the « Bell-State »
Quantum Circuit
Demo : Bell state on a quantum machine

print('Hello World!')
using qiskit library to run quantum program with Python.
Programing

In [1]:
```python
from qiskit import QuantumCircuit, Aer, execute
backend = Aer.get_backend('qasm_simulator')
qc = QuantumCircuit(2,2)
qc.h(0)
qc.cx(0,1)
qc.measure([0,1],[0,1])
d = execute(qc,backend).result().get_counts()
print(d)
```

```python
{'00': 491, '11': 533}
```
Historic Quantum Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Year</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deutsch</td>
<td>1985</td>
<td>$2 \rightarrow 1$</td>
</tr>
<tr>
<td>Bernstein-Vazirani</td>
<td>1992</td>
<td>$N \rightarrow 1$</td>
</tr>
<tr>
<td>Deutsch-Josza</td>
<td>1992</td>
<td>$2^{N-1} + 1 \rightarrow 1$</td>
</tr>
<tr>
<td>Shor</td>
<td>1994</td>
<td>$e^n \rightarrow (n^2 \log n)(\log \log n))$</td>
</tr>
<tr>
<td>Grover</td>
<td>1996</td>
<td>$N \rightarrow \sqrt{N}$</td>
</tr>
</tbody>
</table>

More and new ones on [quantumalgorithmzoo.org/](quantumalgorithmzoo.org/)
Deutsch & Deutsch-Josza

\[ x |0\rangle \xrightarrow{H} |\Psi_1\rangle \]

\[ y |0\rangle \xrightarrow{X H} |\Psi_2\rangle \]

\[ \xrightarrow{U_f} 0 \oplus f(x) \]

\[ \xrightarrow{H} |\Psi_3\rangle |\Psi_4\rangle \]

\[ \xrightarrow{Z} \]
Bersntein-Vazirani