Introduction to Quantum Computing with Qiskit Workshop

3 February 2024

Speaker | Volintine Ander

UNIVERSITI TEKNOLOGI PETRONAS

KanouQuantum
UTP Quantum Computing Coalition

ELECTRICAL & ELECTRONIC ENGINEERING STUDENT SOCIETY

- What is Quantum Computing?
- What is a qubit?
- What we can and can't do (yet)
- Why quantum computing

What is Quantum Computing?

Quantum computing is a way of using quantum mechanical phenomena to perform computations.

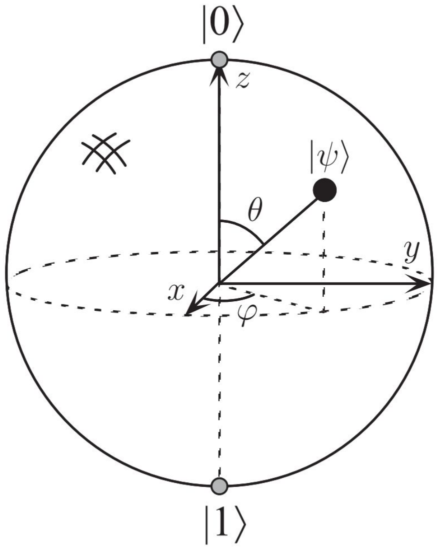
We take advantage of *superposition*, *interference*, and *entanglement*.

What is a qubit?

A **qubit** (also spelled q-bit or qbit) stands for *quantum bit*.

It is the most basic unit of computation in quantum computing.

A qubit is different from a classical bit (or cbit) in that it can either be 0, 1, or a superposition of both.



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

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

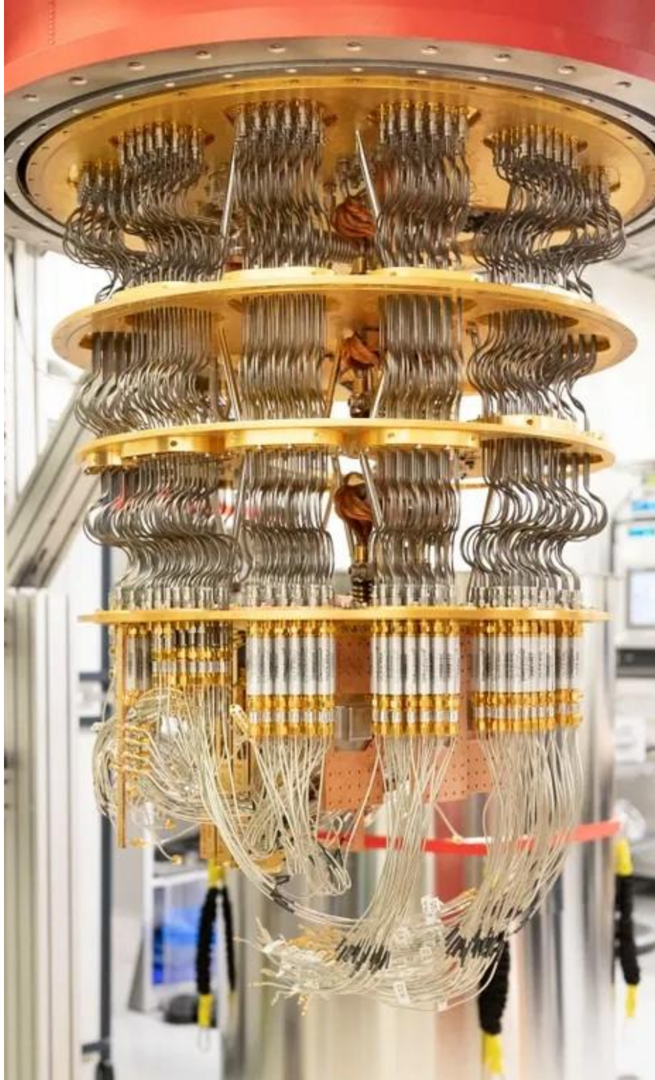
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle.$$

What we can do and can’t do (yet)

Right now: Quantum computing has no practical use.

In the near future: Quantum communications *might* have some limited practical use, more than actual quantum computers.

What is certain: The number of qubits will increase to enable QEC, everything else after that is pure speculation.



Why quantum computing?

Rhetorical question: Why machine learning? Why anything?

In the 1980s AI was very primitive and was not taken seriously by the general public.

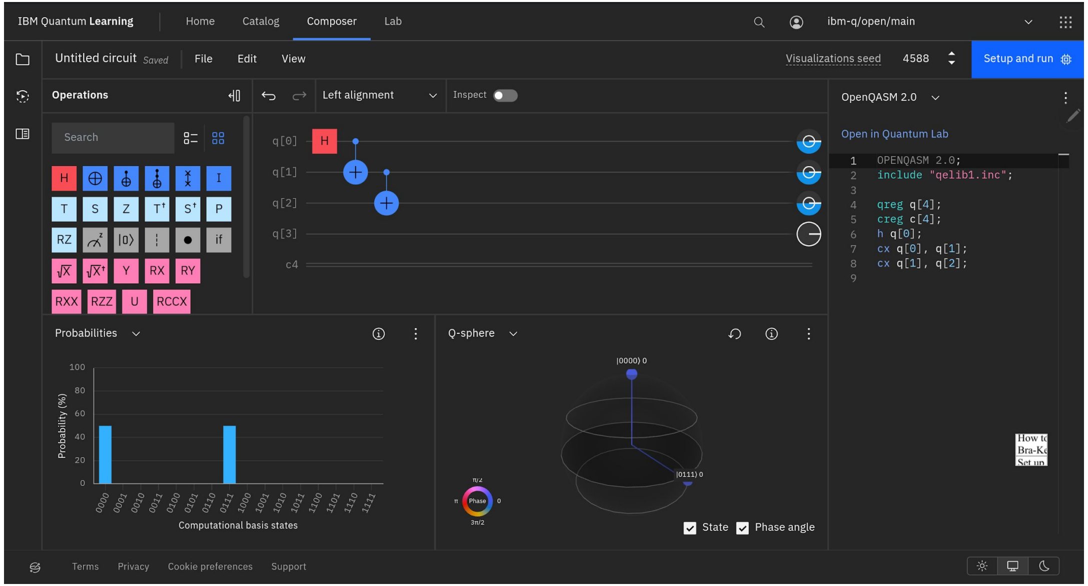
As important algorithms such as stochastic gradient descent, automatic differentiation, transformers, etc were discovered along with more computing power, today, AI is an **industry worth billions**.

Exercise

Explore quantum gates and circuits with IBM Quantum Composer.

Go to quantum.ibm.com/composer

Follow the guide and have a look around.

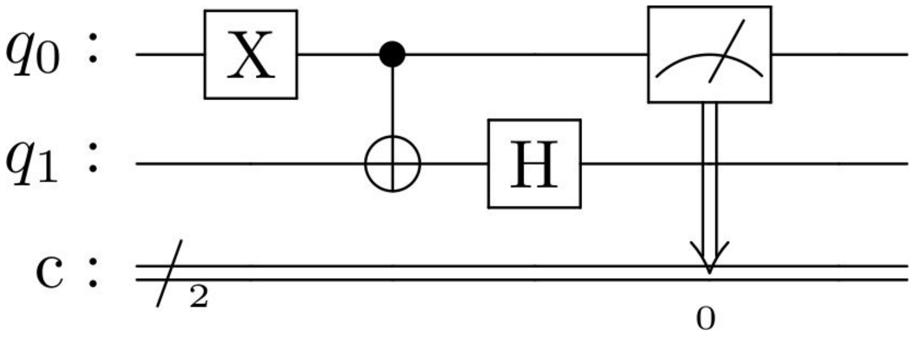


How to read a quantum circuit

Quantum circuits help us model quantum computation. They illustrate what we do to qubits and classical bits.

A quantum circuit will have:

- Qubits and Cbits (Represented as parallel lines)
- Gates



Gates

The basic gates are either unary (takes one input qubit) or binary (takes two input qubits).

Name	Symbol	Operator	Matrix in computational basis $\{\lvert 0\rangle^2 = \lvert 00\rangle, \lvert 1\rangle^2 = \lvert 01\rangle$ $\lvert 2\rangle^2 = \lvert 10\rangle, \lvert 3\rangle^2 = \lvert 11\rangle\}$
Controlled NOT (C-NOT)	= X	$\Lambda^1(X) = \lvert 0\rangle\langle 0\rvert \otimes \mathbf{1} + \lvert 1\rangle\langle 1\rvert \otimes X$	$\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{pmatrix}$
Controlled phase-multiplication	$M(\alpha)$ = $P(\alpha)$	$\Lambda^1(M(\alpha)) = \lvert 0\rangle\langle 0\rvert \otimes \mathbf{1} + \lvert 1\rangle\langle 1\rvert \otimes e^{i\alpha}\mathbf{1} = P(\alpha) \otimes \mathbf{1}$	$\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&e^{i\alpha}&0 \\ 0&0&0&e^{i\alpha} \end{pmatrix}$

Name	Symbol	Operator	Matrix in basis $\{\lvert 0\rangle, \lvert 1\rangle\}$
Identity		$\mathbf{1}$	$\begin{pmatrix} 1&0 \\ 0&1 \end{pmatrix}$
Phase-factor	$M(\alpha)$	$M(\alpha) := e^{i\alpha}\mathbf{1}$	$\begin{pmatrix} e^{i\alpha}&0 \\ 0&e^{i\alpha} \end{pmatrix}$
Phase-shift	$P(\alpha)$	$P(\alpha) := \lvert 0\rangle\langle 0\rvert + e^{i\alpha}\lvert 1\rangle\langle 1\rvert$	$\begin{pmatrix} 1&0 \\ 0&e^{i\alpha} \end{pmatrix}$
PAULI-X or Q-NOT	X	$X := \sigma_x$	$\begin{pmatrix} 0&1 \\ 1&0 \end{pmatrix}$
PAULI-Y	Y	$Y := \sigma_y$	$\begin{pmatrix} 0&-i \\ i&0 \end{pmatrix}$
PAULI-Z	Z	$Z := \sigma_z$	$\begin{pmatrix} 1&0 \\ 0&-1 \end{pmatrix}$
HADAMARD	H	$H := \frac{\sigma_x+\sigma_z}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1&1 \\ 1&-1 \end{pmatrix}$
Spin-rotation by angle α around $\hat{\mathbf{n}}$	$D_{\hat{\mathbf{n}}}(\alpha)$	$D_{\hat{\mathbf{n}}}(\alpha)$	$\begin{pmatrix} \cos\frac{\alpha}{2} - i\sin\frac{\alpha}{2}n_z & -i\sin\frac{\alpha}{2}(n_x - in_y) \\ -i\sin\frac{\alpha}{2}(n_x + in_y) & \cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2}n_z \end{pmatrix}$

Bra Cat notation

In quantum computing, pretty much everything is described using linear algebra.

Qubits, gates, operations, etc are described with matrices that can get big and tedious.

To simplify the clutter, we use Dirac's *bra-ket* notation.

Bras and kets

When we say *bra* we mean this: $\langle A| \doteq \begin{pmatrix} A_1^* & A_2^* & \cdots & A_N^* \end{pmatrix}$

This expression is pronounced "bra A"

Not this:

Bras and kets

When we say *ket* we mean this:

$$|B\rangle \doteq \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{pmatrix}$$

This expression is pronounced “ket B”

Not *cat*:

Tensor product*

Definition

Given vectors $\mathbf{a} \in \mathbb{H}^{(n)}$ and $\mathbf{b} \in \mathbb{H}^{(m)}$ the tensor product $\otimes$ between the vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{a} \otimes \mathbf{b}$ is a vector in the space $\mathbb{H}^{(n)} \times \mathbb{H}^{(m)} \rightarrow \mathbb{H}^{(n+m)}$

This is defined by the Kronecker product:

$$|x\rangle \otimes |y\rangle = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \otimes \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \cdot \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \\ x_1 \cdot \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} x_0 \cdot y_0 \\ x_0 \cdot y_1 \\ x_1 \cdot y_0 \\ x_1 \cdot y_1 \end{bmatrix}$$

Example

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \cdot \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \\ 2 \cdot \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \\ 8 \\ 10 \end{bmatrix}$$

*This definition is a special case, but enough for all of our intents and purposes. Refer to p. 71 of *Quantum Computation & Quantum Information* by Nielsen & Chuang for further reading.

Exercise

Go to quantum.ibm.com

Sign up for an IBM account if you haven't already.

- Log into IBM Quantum Lab
- Access the Google Drive link
- Download the notebooks
- Upload to your IBM Lab instances and open

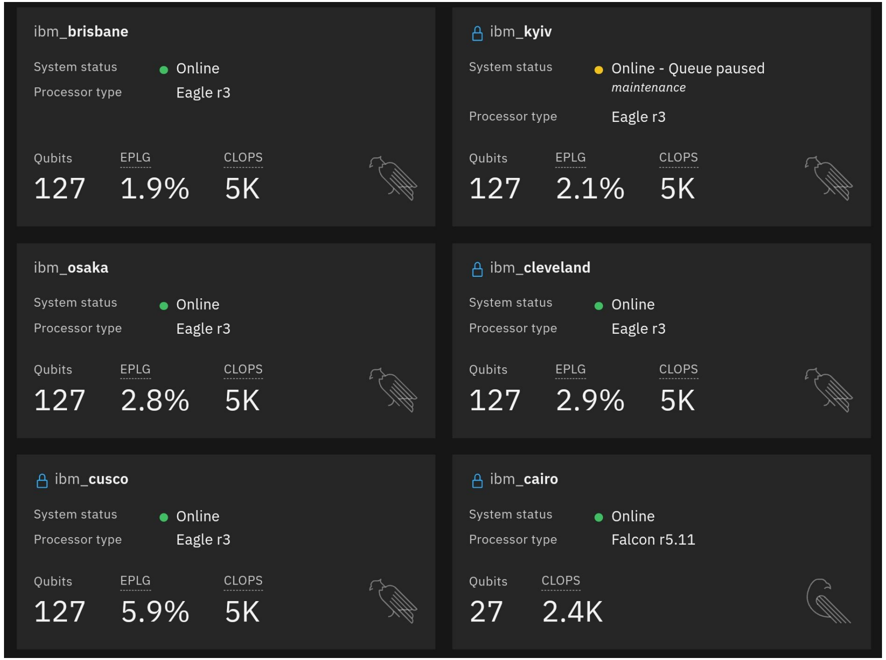
Bah makan dulu

Let’s go get lunch before we continue.

Exercise

Welcome back. Let’s start by doing these:

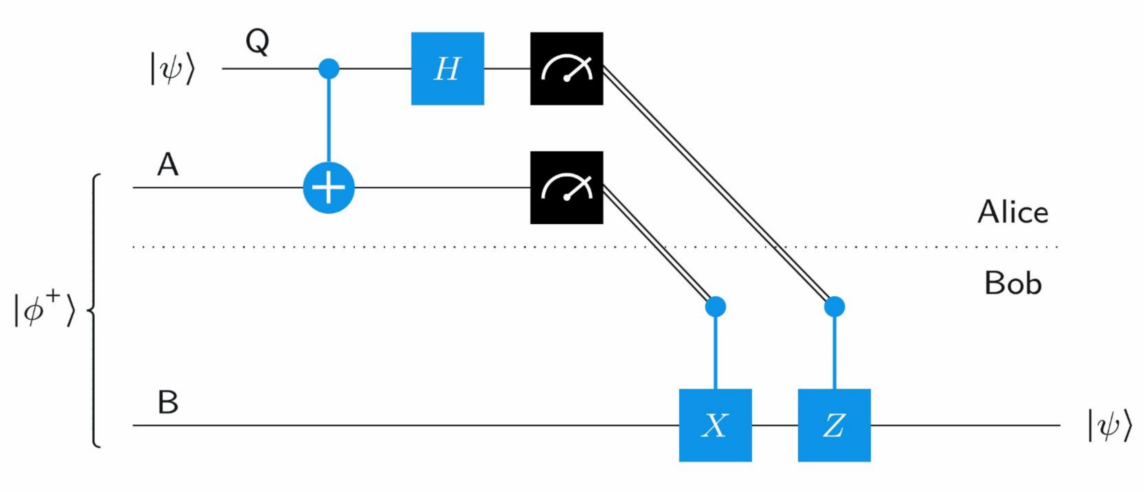
1. **Explore** available IBM quantum computers and simulators
2. **Learn** about quantum and classical registers
3. **Build** circuits using code (try to navigate the API documentation)



Quantum teleportation

Alice (A) wants to send the state |ψ> to Bob (B). She first puts the qubit Q in the desired state, does magic, and then measures A and Q.

Bob does some more magic on his qubit (B) based on Alice's measurement result, which results in B being in the exact state |ψ>.



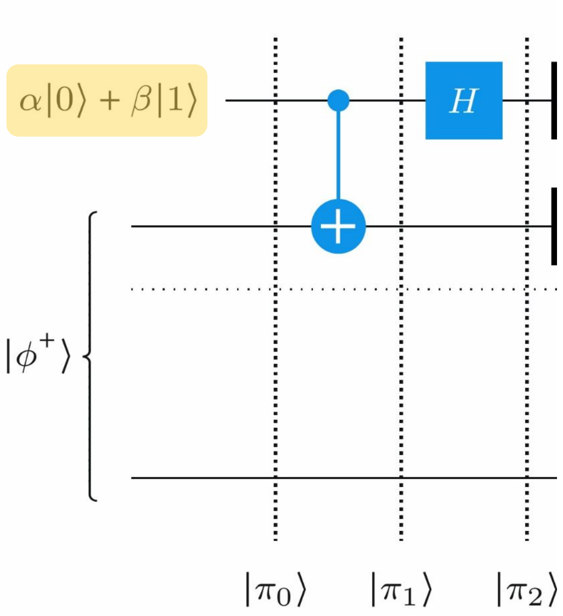
What exactly is the magic?

Alice executes this bit of the circuit:
The teleported state is in orange.

$$|\pi_0\rangle = |\phi^+\rangle \otimes (\alpha|0\rangle + \beta|1\rangle) = \frac{\alpha|000\rangle + \alpha|110\rangle + \beta|001\rangle + \beta|111\rangle}{\sqrt{2}}.$$

$$|\pi_1\rangle = \frac{\alpha|000\rangle + \alpha|110\rangle + \beta|011\rangle + \beta|101\rangle}{\sqrt{2}}.$$

$$\begin{aligned}|\pi_2\rangle &= \frac{\alpha|00\rangle|+\rangle + \alpha|11\rangle|+\rangle + \beta|01\rangle|-\rangle + \beta|10\rangle|-\rangle}{\sqrt{2}} \\ &= \frac{\alpha|000\rangle + \alpha|001\rangle + \alpha|110\rangle + \alpha|111\rangle + \beta|010\rangle - \beta|011\rangle + \beta|100\rangle - \beta|101\rangle}{2}.\end{aligned}$$



$$= \frac{1}{2}(\alpha|0\rangle + \beta|1\rangle)|00\rangle + \frac{1}{2}(\alpha|0\rangle - \beta|1\rangle)|01\rangle + \frac{1}{2}(\alpha|1\rangle + \beta|0\rangle)|10\rangle + \frac{1}{2}(\alpha|1\rangle - \beta|0\rangle)|11\rangle.$$

Continued...

Based on Alice’s measurements, the cbits *ab* can be any of these:
00, **01**, **10**, **11**
These are respectively the rightmost state in each of the term in the sum:

$$= \quad \frac{1}{2}(\alpha|0\rangle + \beta|1\rangle)|00\rangle + \frac{1}{2}(\alpha|0\rangle - \beta|1\rangle)|01\rangle + \frac{1}{2}(\alpha|1\rangle + \beta|0\rangle)|10\rangle + \frac{1}{2}(\alpha|1\rangle - \beta|0\rangle)|11\rangle.$$

The leftmost state in each of the term above are the possible states qubit B can be in. Notice that at 00, B is already in the state |ψ>, whereas the rest of the states only need either X, Z, or ZX to be in the state |ψ>.

X and Z gates conditioned on the cbits *ab* will ensure B is exactly in the same state Q that Alice wanted to teleport.

Exercise

Let's go beyond the theory and implement the algorithm in Qiskit.

Continue onwards to the Quantum Teleportation section.