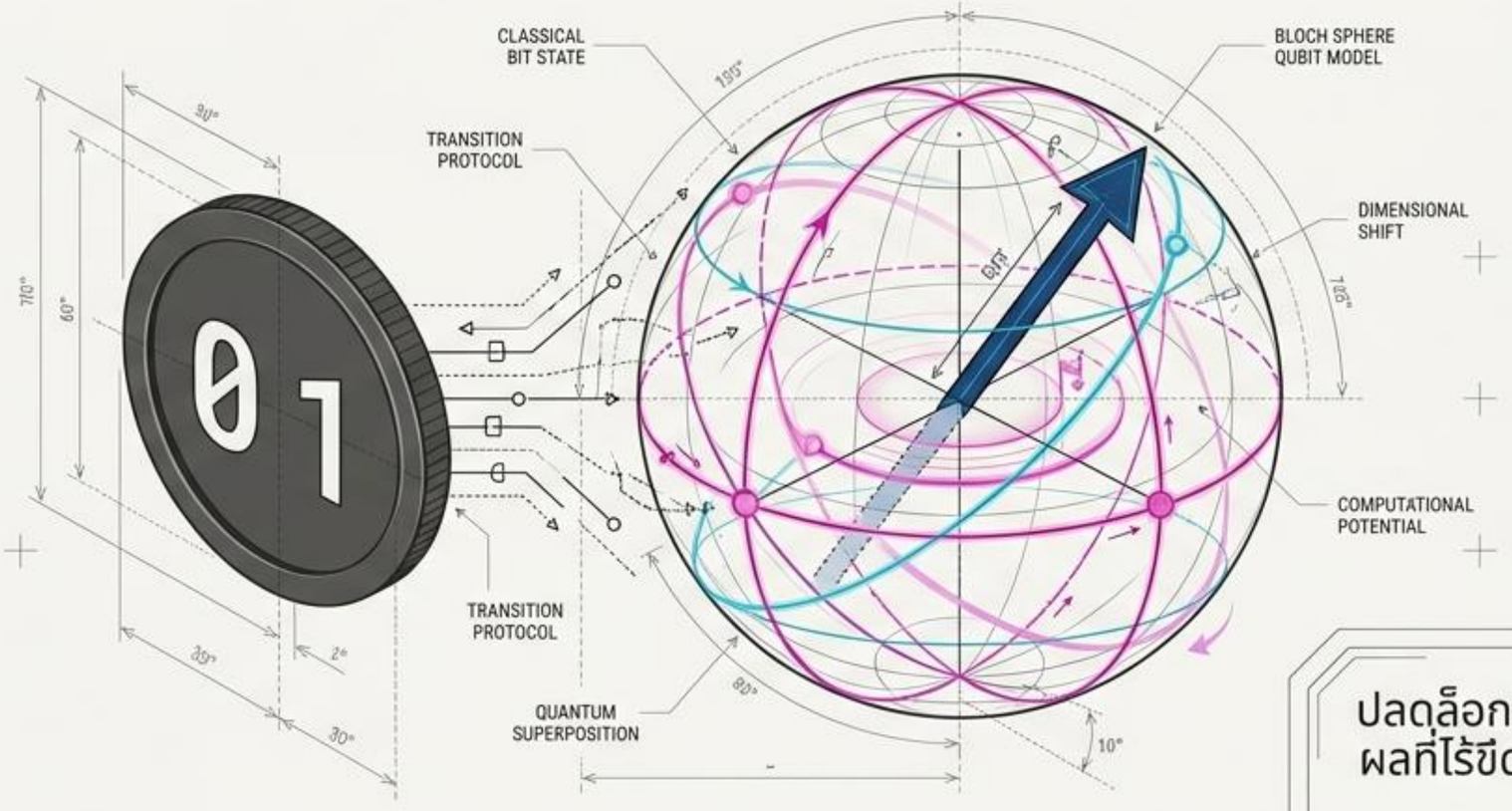
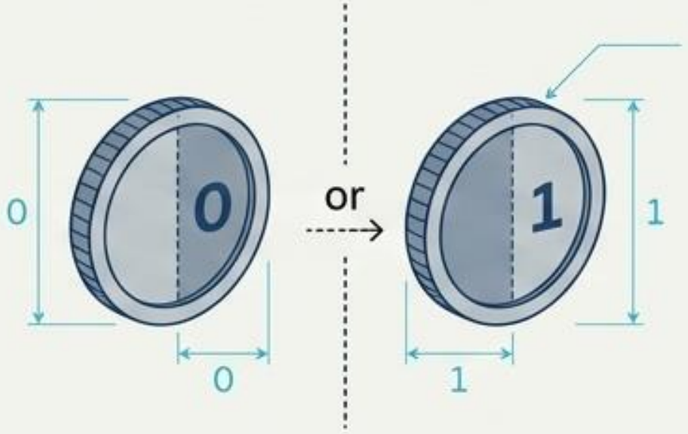
สถาปัตยกรรมแห่งความเป็นจริงใหม่

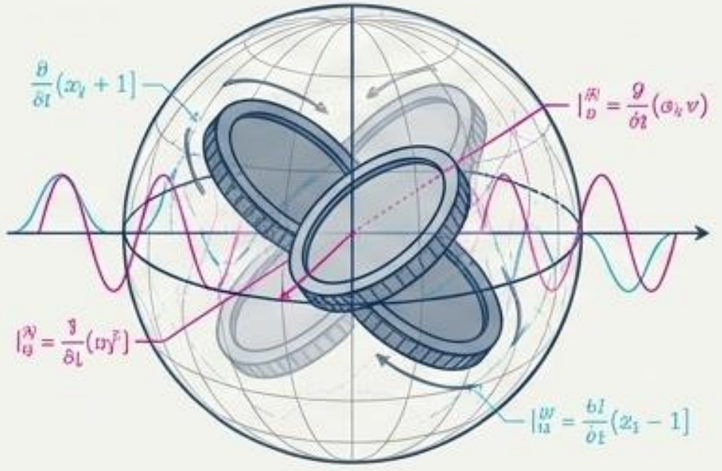
พิมพ์เขียวควอนตัม: จากอนุภาคพื้นฐานสู่อัลกอริทึมที่เปลี่ยนโลก



ปลดล็อกศักยภาพการประมวลผลที่ไร้ขีดจำกัดด้วยกลศาสตร์ควอนตัม

ก้าวข้ามขีดจำกัดของทฤษฎีข้อมูลเดิม





บิตคลาสสิก (Classical Bit)

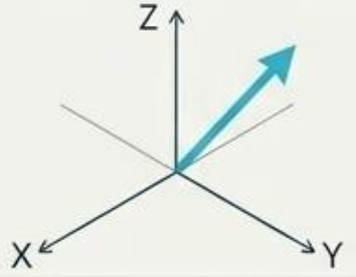
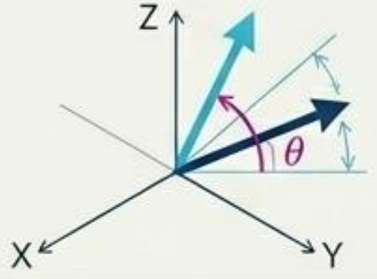
- เป็นค่าใดค่าหนึ่ง (Either 0 or 1)
- ประมวลผลได้เพียง 1 จาก 2^N รูปแบบที่เป็นไปได้ในหนึ่งรอบ

คิวบิต (Quantum Bit / Qubit)

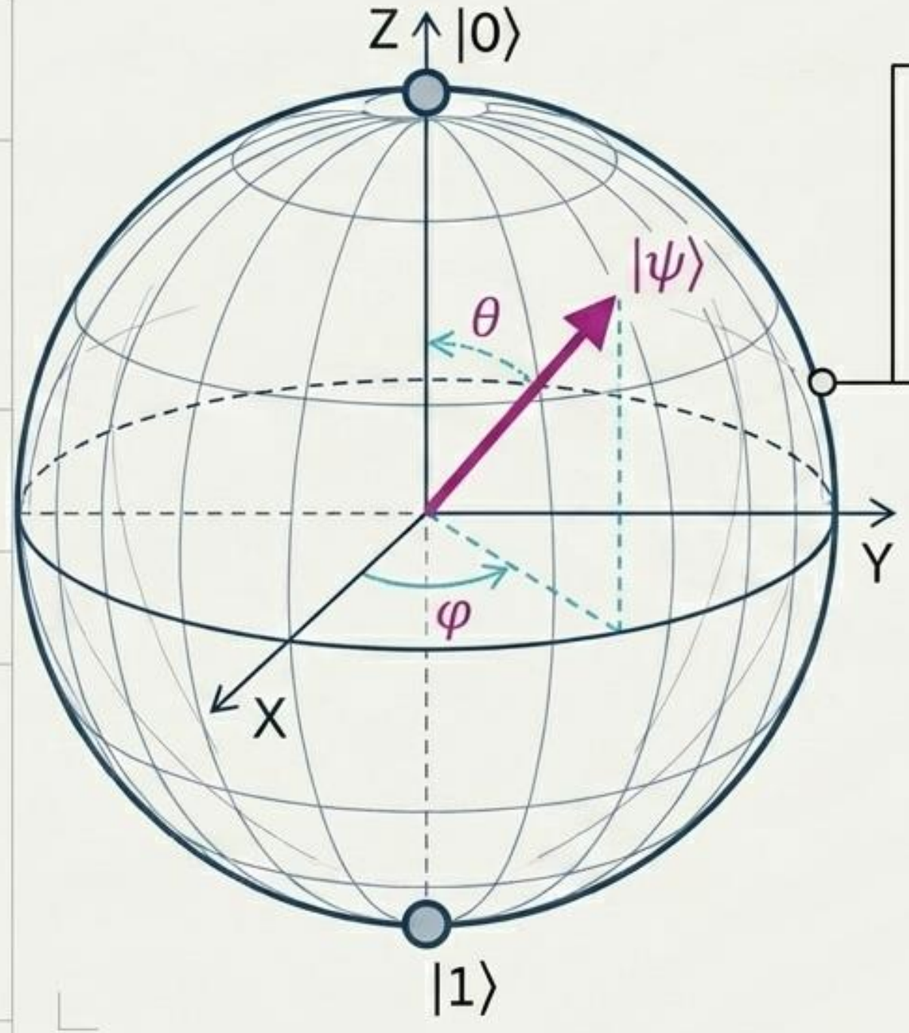
- การทับซ้อนสถานะ (Superposition: Both 0 and 1)
- ประมวลผลได้ทั้งหมด 2^N รูปแบบพร้อมกัน

Superposition allows calculation on many states at the same time.

พจนานุกรมฉบับสถาปนิกควอนตัม

Symbol	Visual Model	Definition
$\|\psi\rangle$		**สถานะควอนตัมบริสุทธิ์** (Pure quantum state). เวกเตอร์คอลัมน์แสดงข้อมูล.
$\langle\varphi\|\psi\rangle$		**ผลคูณภายใน** (Inner product). การวัดความสอดคล้องกันของสองสถานะ.
ρ		**เมทริกซ์ความหนาแน่น** (Density Matrix). ครอบคลุมทั้งสถานะบริสุทธิ์และสถานะผสม อธิบายความเป็นไปได้ทั้งหมดของระบบ.

แผนที่แห่งความน่าจะเป็น: ทรงกลมบล็อค



พิกัดข้อมูล (Coordinates)

สถานะ $|\psi\rangle$ อาศัยอยู่บนพื้นผิวทรงกลม กำหนดโดยมุม θ และ φ.

สถานะพื้นฐาน (Basis States)

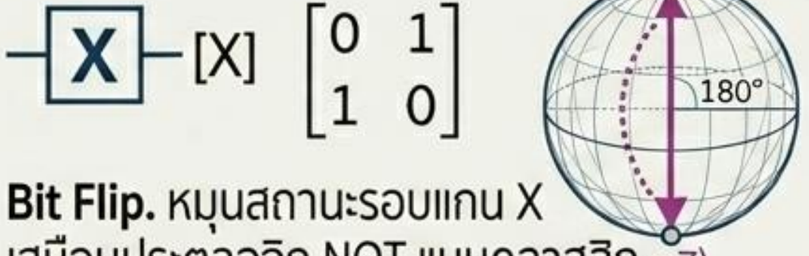
ขั้วเหนือคือ $|0\rangle$, ขั้วใต้คือ $|1\rangle$.
แกน X/Y คือสถานะซ้อนทับ.

กฎของบอร์น (Born Rule)

ทันทีที่มีการสังเกต (Measurement) สถานะที่ซ้อนทับจะล่มสลาย (Collapse) พุ่งตรงไปยังขั้วใดขั้วหนึ่ง ความจะแป็นขึ้นอยู่กับระยะห่างจากขั้วนั้น.

เครื่องมือของสถาปนิก: ควอนตัมเกตพื้นฐาน

Pauli-X (Bit Flip)



Bit Flip. หมุนสถานะรอบแกน X เสมือนประตูลอจิก NOT แบบคลาสสิก.

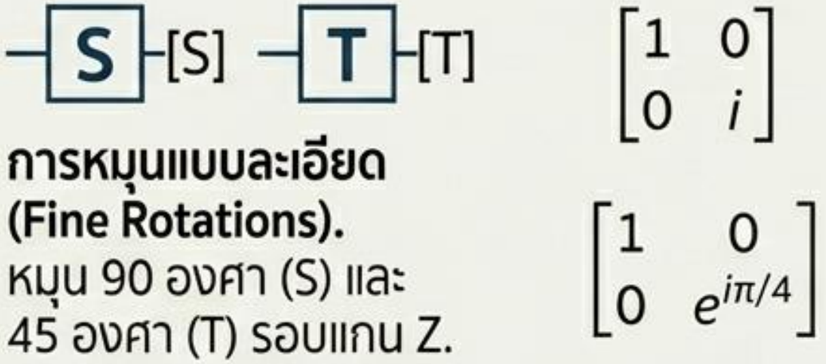
Pauli-Z (Phase Flip)

Phase Flip. หมุนรอบแกน Z เปลี่ยนเฟสของสถานะซ้อนทับ.

Hadamard (Superposition)

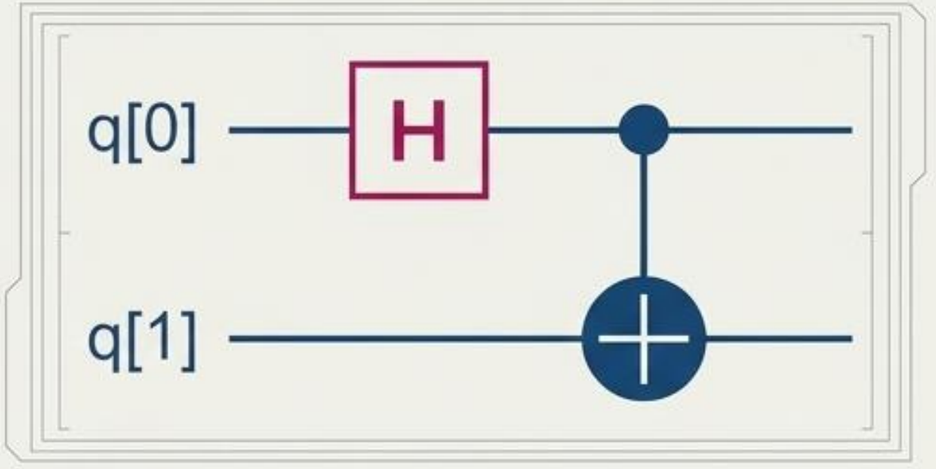
สร้าง Superposition. กุญแจสำคัญที่ทำให้ 1 คิวบิต ประมวลผล 2 สถานะพร้อมกันได้.

Phase & π/8 (Fine Rotations)



การหมุนแบบละเอียด (Fine Rotations). หมุน 90 องศา (S) และ 45 องศา (T) รอบแกน Z.

ความเชื่อมโยงที่ไร้พรมแดน: ความพัวพันเชิงควอนตัม



สถานะที่รวมกันเป็นหนึ่ง (Bipartite States)

เมื่อคิวบิตสองตัวเกิดความพัวพัน (Entangled) สถานะของพวกมันจะไม่สามารถแยกอธิบายอิสระจากกันได้อีกต่อไป.

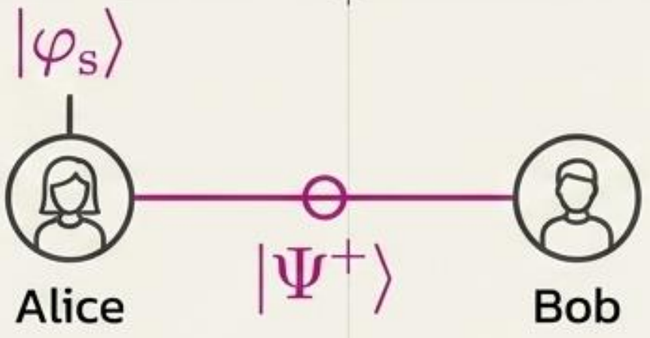
The Master Gate (CNOT)

Controlled-NOT Gate. หากคิวบิตควบคุม (Control) เป็น 1 จะพลิกสถานะของคิวบิตเป้าหมาย (Target)
การประกอบ Hadamard และ CNOT สร้างสถานะ Bell States ที่พัวพันกันสมบูรณ์แบบ.

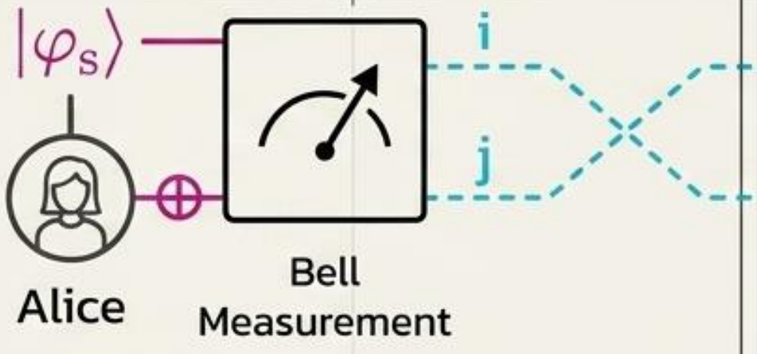
โปรโตคอล 1: การเทเลพอร์ตสถานะควอนตัม

เป้าหมาย: ส่งสถานะควอนตัมที่ไม่ทราบค่า ผ่านบิตคลาสสิก

1. เตรียมพร้อม



2. วัดและทำลาย

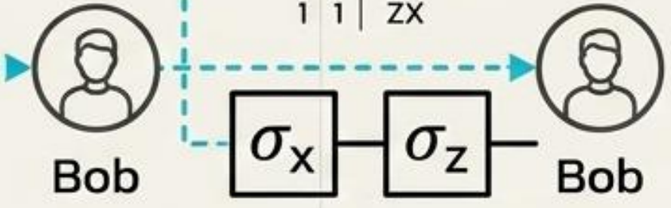


Alice วัดค่าคิวบิต สถานะเดิมถูกทำลาย
ส่งข้อมูลคลาสสิก 2 บิต (i, j) ไปหา Bob.

3. สร้างใหม่

Logic Table:

i	j	Gate
0	0	I
0	1	X
1	0	Z
1	1	ZX



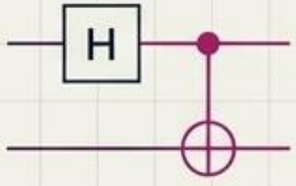
$|\phi\rangle_B$

โปรโตคอล 2: การเข้ารหัสข้อมูลความหนาแน่นสูง

Teleportation: ส่ง 1 คิวบิต โดยใช้ 2 บิตคลาสสิก
Superdense Coding: ส่ง 2 บิตคลาสสิก โดยใช้ 1 คิวบิต

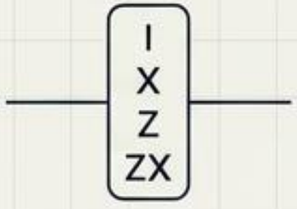
1. Preparation

สร้าง Bell state เริ่มต้น แบ่งคิวบิตให้ Alice และ Bob.



2. Encoding

Alice เข้ารหัสข้อมูลคลาสสิก 2 บิต (00, 01, 10, 11) ลงในคิวบิตตัวเดียว โดยใช้เกต I, X, Z, หรือ ZX.

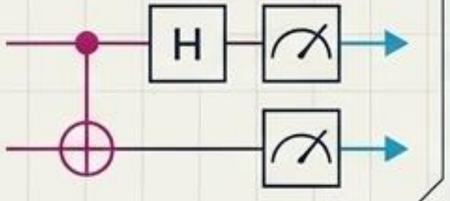


3. Transmission

Alice ส่งคิวบิตของเธอเพียงตัวเดียวไปให้ Bob.

4. Decoding

Bob นำคิวบิตของ Alice มาประมวลผลร่วมกับของเขา (CNOT + H) วัดค่าและได้ข้อมูล 2 บิตกลับคืนมา.

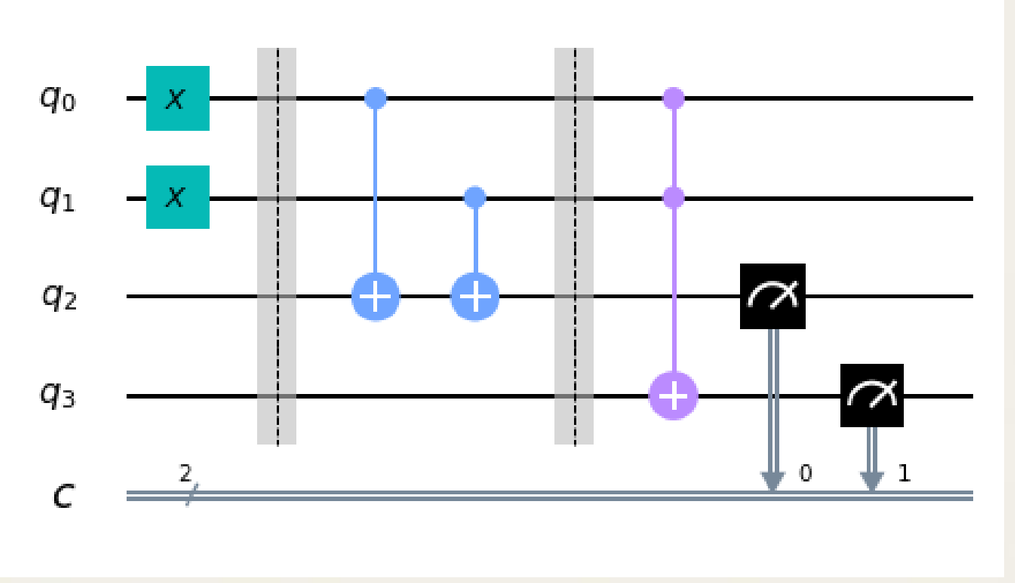


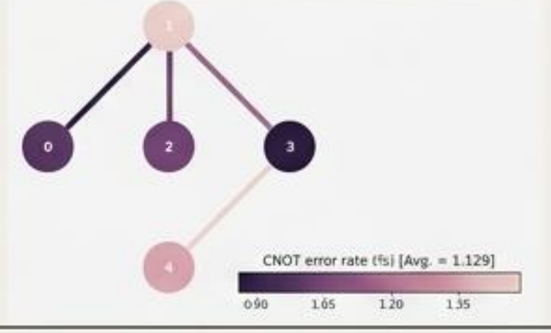
จากทฤษฎีสู่โลกความจริงด้วย Qiskit

ภาษา Qiskit (Python-based framework)

```
qc_ha = QuantumCircuit(4,2)
# encode inputs in qubits 0 and 1
qc_ha.x(0) # For a=0, remove the this line. For a=1, leave it.
qc_ha.x(1) # For b=0, remove the this line. For b=1, leave it.
qc_ha.barrier()
# use cnots to write the XOR of the inputs on qubit 2
qc_ha.cx(0,2)
qc_ha.cx(1,2)
qc_ha.barrier()
# use cnots to write the XOR of the inputs on qubit 2
qc_ha.ccx(0,1,3)
qc_ha.barrier()
# extract outputs
qc_ha.measure(2,0) # extract XOR value
qc_ha.measure(3,0)
```

วงจรควอนตัมลอจิก (Quantum Logic Circuit)

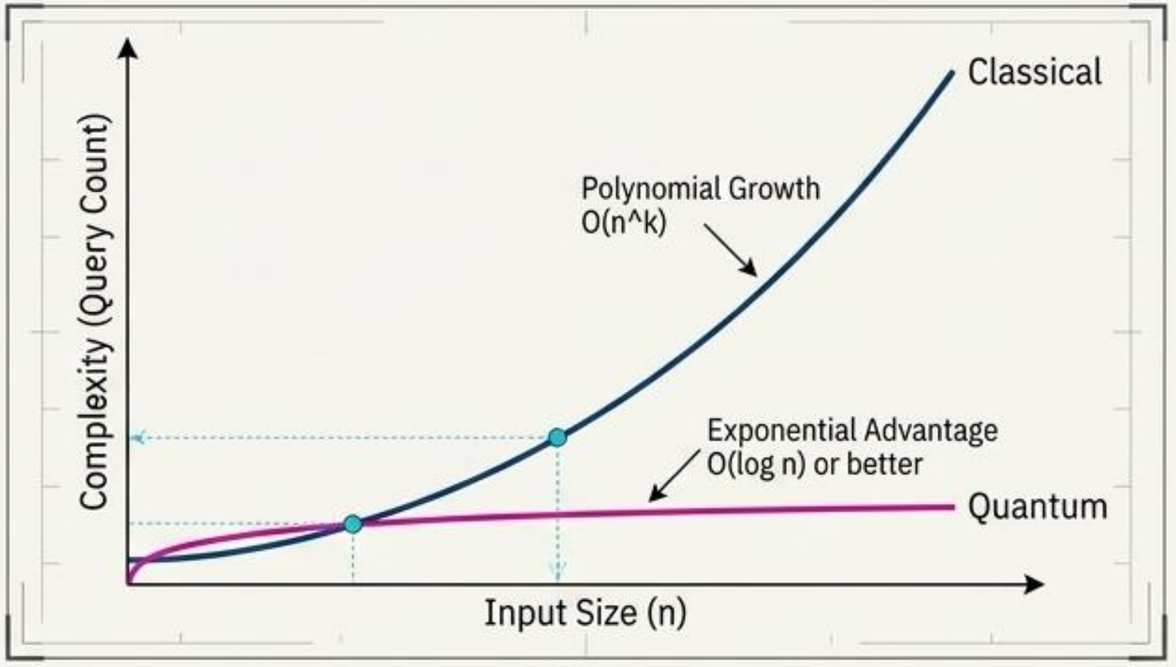




ภูมิสถาปัตย์ฮาร์ดแวร์ (Hardware Topology)

การออกแบบวงจรจริงต้องคำนึงถึง Noise และ Error rate ระหว่างการเชื่อมต่อของคิวบิตทางกายภาพ (Backend transpilation).

จุดพลิกผัน: ทำไมอัลกอริทึมควอนตัมจึงสำคัญ?



วงจรควอนตัมสามารถรันฟังก์ชั่นแบบคลาสสิกได้ทั้งหมด แต่ความได้เปรียบที่แท้จริง (Quantum Advantage) เกิดขึ้นเมื่อเราใช้อัลกอริทึมที่ออกแบบมาเฉพาะ เพื่อลดจำนวนก้าว (Query Complexity) อย่างทวีคูณ.

Deutsch-Jozsa
พิสูจน์ความเหนือกว่า

→

Grover
ค้นหาข้อมูลไร้โครงสร้าง

→

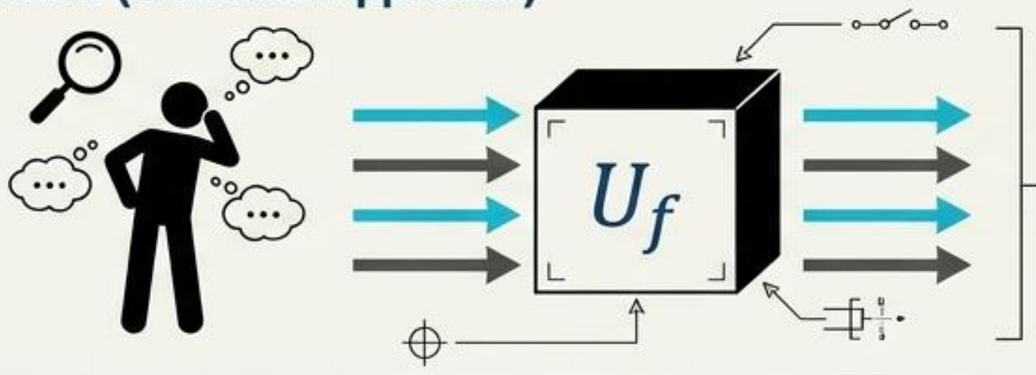
Shor
แยกตัวประกอบและการเข้ารหัส

→

บทพิสูจน์แรก: อัลกอริทึม Deutsch-Jozsa

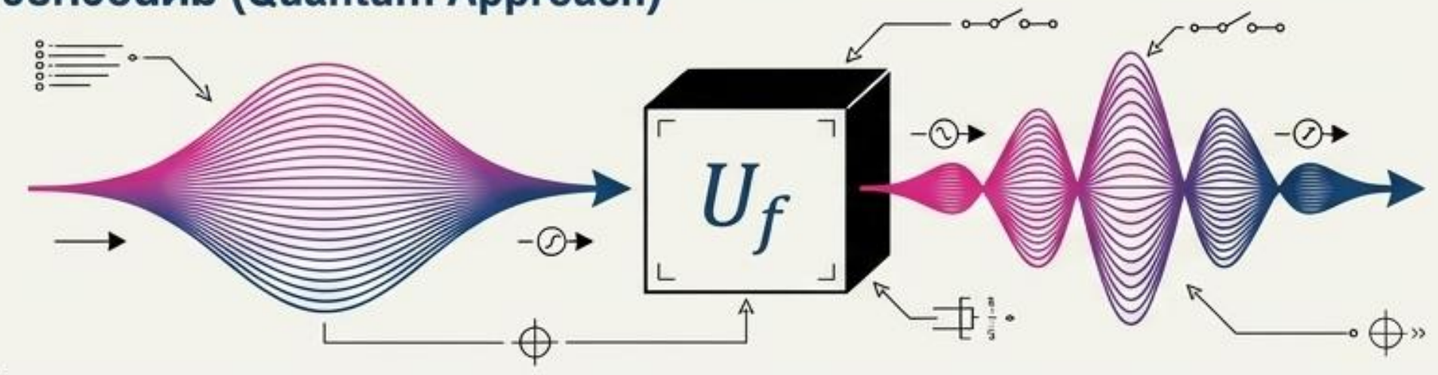
ฟังก์ชั่นในกล่องดำ (Black Box) เป็นแบบใด? แบบคงที่ (Constant: ออก 0 หรือ 1 ล้วน) หรือ แบบสมดุล (Balanced: ออก 0 ครึ่งหนึ่ง 1 ครึ่งหนึ่ง)?

วิธีคลาสสิก (Classical Approach)



ต้องสุ่มตรวจสอบถึง (N/2) + 1 ครั้ง
เพื่อยืนยันความถูกต้อง 100%.

วิธีควอนตัม (Quantum Approach)

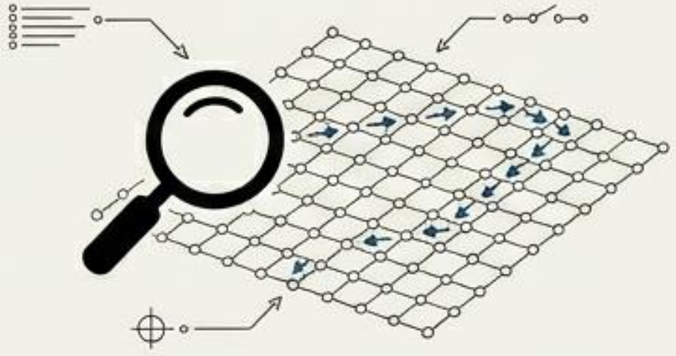


ส่งคลื่น Superposition เข้าไปครั้งเดียว
(1 Query). การแทรกสอดของเฟส
(Phase Interference)
จะเปิดเผยคุณสมบัติของฟังก์ชั่นได้ทันที.

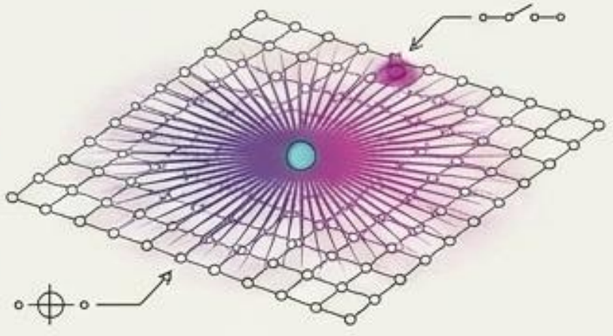
ค้นหาในเขาวงกต: อัลกอริทึมของ Grover

การค้นหาข้อมูลที่ไม่มีโครงสร้าง (Unstructured Search). กำหนดค้นหาเป้าหมาย W จากฐานข้อมูลขนาด N.

Classical Time: O(N)



Quantum Time: $O(\sqrt{N})$

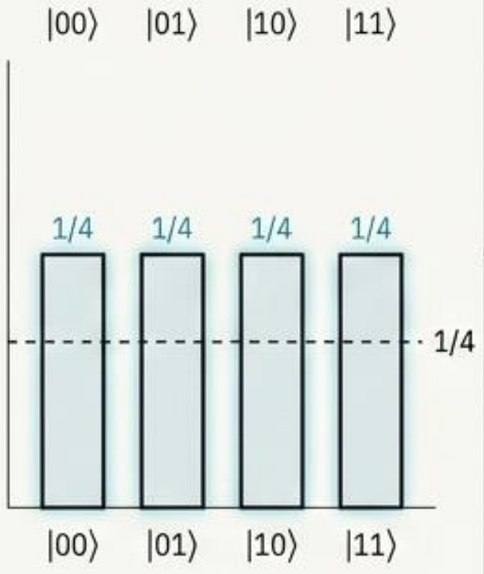


ความเร็วเพิ่มขึ้นแบบก้าวกระโดดเชิงกำลังสอง (Quadratic Speedup).

ใช้อราคิล (Oracle) เพื่อระบุตัวเป้าหมาย
และใช้ การขยายแอมพลิจูด
(Amplitude Amplification)
ดันคำตอบที่ถูกต้องให้โดดเด่นขึ้นมา.

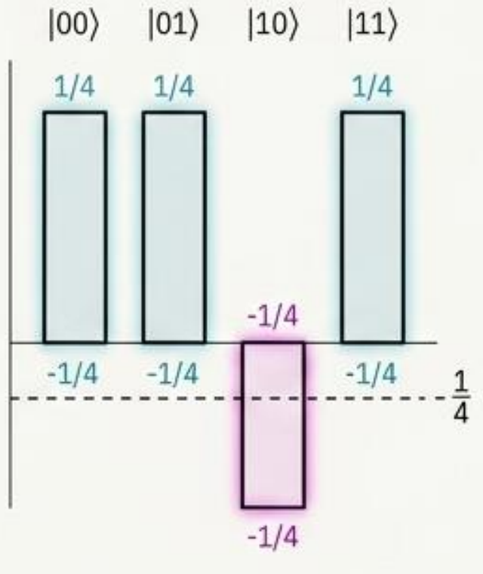
กลไกขับเคลื่อน: การขยายแอมพลิจูด

1. เริ่มต้น



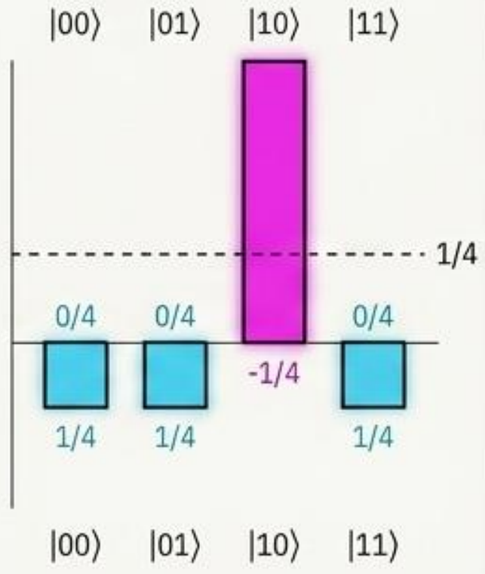
นำคิวบิตเข้าสู่ Superposition ทุกสถานะมีโอกาสถูกพบเท่ากัน.

2. พลิกเฟส (Oracle)



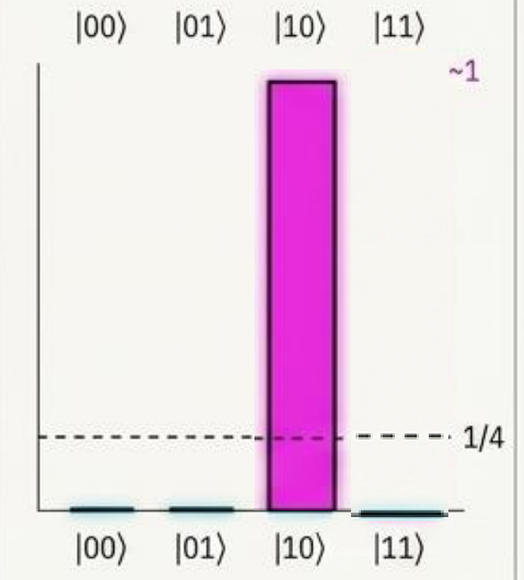
ออราเคิลกลับด้าน (Phase Flip) สถานะเป้าหมาย ทำให้ค่าเฉลี่ยรวมลดลง.

3. พลิกกลับรอบค่าเฉลี่ย



การผกผัน (Inversion) ดันสถานะที่ถูกพลิกเฟสให้พุ่งสูงขึ้น กดสถานะอื่นให้ต่ำลง.

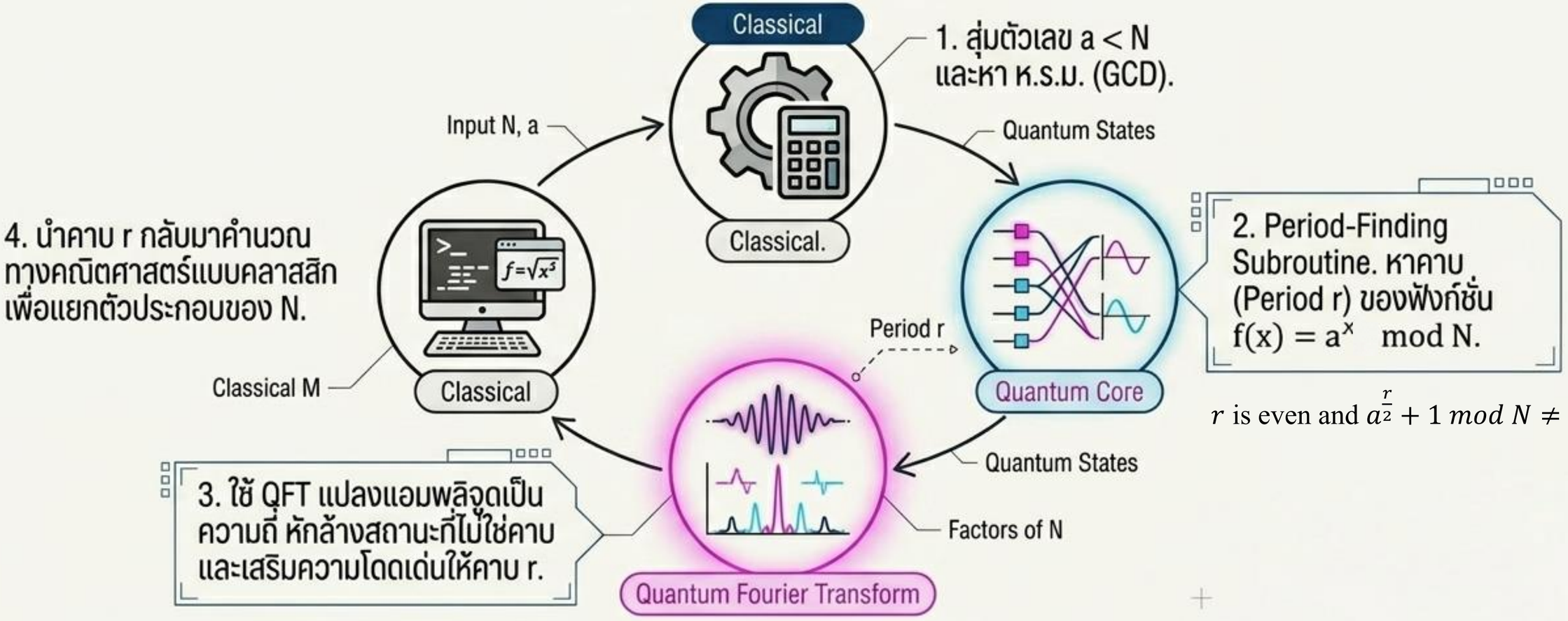
4. วัดค่า (Measurement)



ทำซ้ำรอบ ความน่าจะเป็นของคำตอบที่ถูกจะเข้าใกล้ 1.

ภัยคุกคามการเข้ารหัส: อัลกอริทึมของ Shor

การแยกตัวประกอบจำนวนเต็มขนาดใหญ่ (Factoring large integers) คือรากฐานของการเข้ารหัสปัจจุบัน (RSA). คลาสสิกคอมพิวเตอร์ทำไม่ได้ แต่ควอนตัมทำได้.



r is even and $a^{\frac{r}{2}} + 1 \; mod \; N \neq 0$

บทสรุปภาพรวมแห่งอนาคต

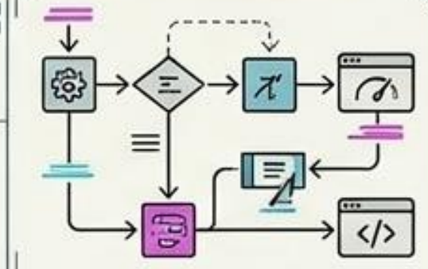
Pillar 1

Qubits (คิวบิต)
- ทลายขีดจำกัดหน่วยความจำ
ด้วย Superposition

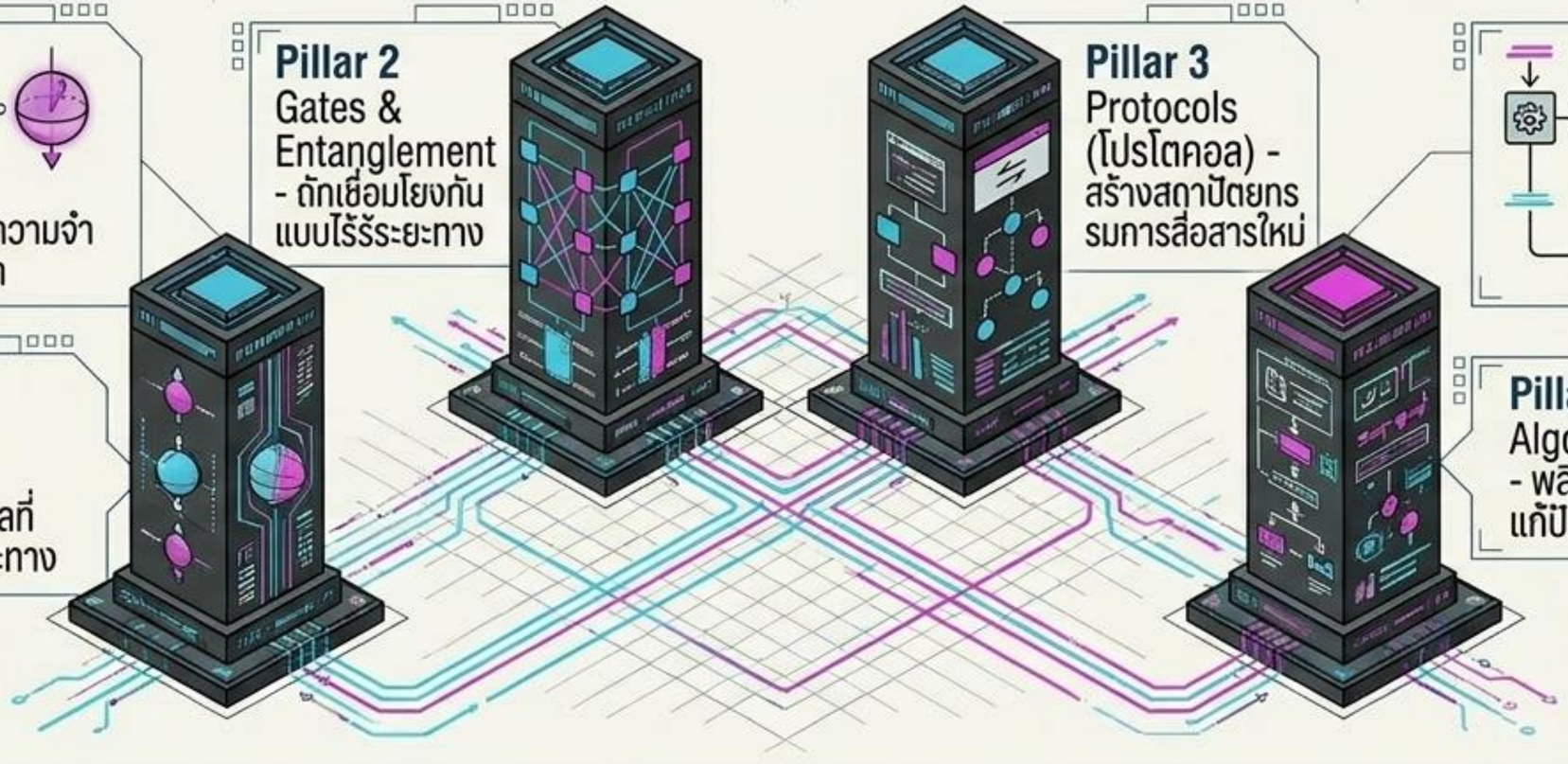
Pillar 2
Gates &
Entanglement -
ถักทอสายใยประมวลผลที่
เชื่อมโยงกันแบบไร้ระยะทาง

Pillar 2
Gates &
Entanglement
- ถักเชื่อมโยงกัน
แบบไร้ระยะทาง

Pillar 3
Protocols
(โปรโตคอล) -
สร้างสถาปัตยกร
รมการสื่อสารใหม่



Pillar 4
Algorithms (อัลกอริทึม)
- พลิกโฉมความเร็วในการ
แก้ปัญหาเชิงกำลัง



ควอนตัมคอมพิวติ้งไม่ใช่เพียงการอัพเกรดความเร็วจากคอมพิวเตอร์คลาสสิก
แต่คือ **กระบวนทัศน์ใหม่** ของมนุษยชาติในการคำนวณและทำความเข้าใจจักรวาล.

Exercise I: Basic Quantum Computing

- Required:
  - Download source codes at https://shorturl.at/yGmXQ for "**Lab-1.ipynb**" and https://shorturl.at/Qa3bV for "**Lab-2.ipynb**" and then upload both files into Google Colab.
- Exercises:
  - Lab-1: Operations on single qubit and multiple qubits gates by IBM Quantum.
  - Lab-2: Quantum circuits by IBM Quantum.

Exercise II: Quantum Algorithms

- Required:
  - Download source codes at https://shorturl.at/6TTYz for "**Lab_Deutsch-Jozsa.ipynb**" and https://shorturl.at/6XEHC for "**Grover's Algorithm.ipynb**" and then upload both files into Google Colab.
- Exercises:
  - Lab_Deutsch-Jozsa : Oracles and the Deutsch-Jozsa algorithm by IBM Quantum
  - Explore Grover's Algorithm to search for the marked quantum state $|010\rangle$.

References & Resources

- Vogel, Manuel. (2011). Quantum Computation and Quantum Information, by M.A. Nielsen and I.L. Chuang. Contemporary Physics - CONTEMP PHYS. 52. 604-605. 10.1080/00107514.2011.587535.
- N. S. Yanofsky and M. A. Mannucci, Quantum Computing for Computer Scientists. Cambridge: Cambridge University Press, 2008.
- Hiu Yung Wong. "Introduction to Quantum Computing: From a Layperson to a Programmer in 30 Steps" *Faculty Research, Scholarly, and Creative Activity* (2022). https://doi.org/10.1007/978-3-030-98339-0
- IBM Quantum https://quantum.cloud.ibm.com/
- Lab answer: https://shorturl.at/8NIsS

Download this slide here ->