

Quantum Computing Algorithm

코딩을 위한 기술 현황과 기본 개념 소개

곽재영 j0210003@naver.com

< 세션을 들어가면서 >

양자 컴퓨팅 (Quantum Computing)

코딩은 어떤 것인지?

어떻게 시작할 것인지?

목차

1. 현황 소개
2. 기초 이론
3. 코딩 환경
4. 맺음말

Quantum Computing 현황 1. 동향 - IQQ

- UN has designated 2025 as the **International Year of Quantum** Science & Technology.

<https://quantum2025.org/>

What Is IQQ?



Recognizing the importance of quantum science and the need for wider awareness of its past and future impact, dozens of national scientific societies gathered together to support marking 100 years of quantum mechanics with a U.N.-declared international year.

On June 7, 2024, the United Nations proclaimed 2025 as the International Year of Quantum Science and Technology (IQQ). According to the proclamation, this year-long, worldwide initiative will “be observed through activities at all levels aimed at increasing public awareness of the importance of quantum science and applications.”

Anyone, anywhere can participate in IQQ by helping others to learn more about quantum on this centennial occasion, participating in or organizing an IQQ event, or simply taking the time to learn more about quantum science and technology.

Get Involved →



- 1900 : Black-body Radiation - Max Planck (Quantized Energy)
- 1913 : Bohr Model - Why Quantized
- 1925 : Matrix Mechanics - Werner Heisenberg
- 1926 : Schrödinger equation

Quantum Computing 현황 1. 동향 - Bitcoin

- Bitcoin Will Adapt To The Technological Advances Of Quantum Computing.

<https://www.forbes.com/sites/digital-assets/2025/01/17/bitcoin-will-adapt-to-quantum-computing/>



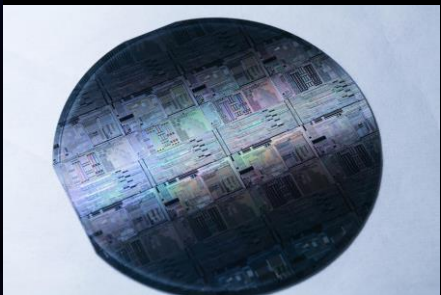
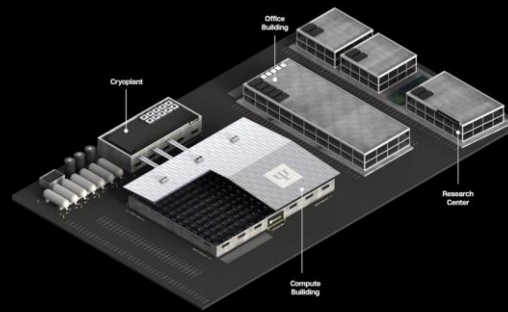
- Bitcoin 취약점
 - Signatures
 - Hash Functions (SHA-256)
- Bitcoin 대처
 - 양자 대응 보안으로 이동

Quantum Computing 현황 1. 동향 - Investment

- PsiQuantum Receives \$940 Million (AUD) From Australian Government.

<https://thequantuminsider.com/2024/04/29/psiquantum-receives-940-million-aud-from-australian-government/>

May Now Be World's Highest Funded Independent Quantum Firm



- 호주 정부 투자 : 한화 8400억 이상
- 불확실한 청사진 (~2027)
 - 크기, 사양, 용도, 운영방안
- Dr Shadbolt says
 - 첫 양자 컴퓨터는 결승 지점 반대편인 출발선에 있을 것입니다
 - 암호 해독이나 경로 최적화에 사용되지 않을 것입니다

Quantum Computing 현황 2. 하드웨어 – Qbit 제조사 1

- Top 18 Quantum Computer Companies [2025 Updated]

<https://www.spinquanta.com/news-detail/quantum-computer-manufacturers>

순위	회사	특징
1	IBM	USA, Superconducting, Hardware, Software, Ecosystem
2	SPINQ	China, Superconducting + NMR (Nuclear Magnetic Resonance)
3	Google Quantum	USA, Superconducting, Bristlecone(2017) → Sycamore(2019) → Willow(2024)
4	Microsoft	USA, Azure Quantum platform (IonQ and Quantinuum), Topological Qubits
5	Intel	USA, Spin Qubits using Silicon, Chips only
6	Honeywell Quantum Solutions	USA, Trapped-Ion Qubits, Quantinuum with Cambridge Quantum Computing

Quantum Computing 현황 2. 하드웨어 – Qbit 제조사 2

- Top 18 Quantum Computer Companies [2025 Updated]

<https://www.spinquanta.com/news-detail/quantum-computer-manufacturers>

순위	회사	특징
7	Quantinuum	USA/UK, Honeywell Quantum Solutions (HW) + Cambridge Quantum Computing (CQC) (SW)
8	IonQ	USA, Trapped-Ion Qubits, System Access through Clouds
9	D-Wave Systems	Canada, Quantum Annealing Technology, D-Wave One to Lockheed Martin
10	Xanadu	Canada, Photonics Qubits, Open-source Software Library through Clouds
11	Quantum Circuits	USA, Superconducting, Dual-rail Cavity Qubits → correct first, then scale
12	PsiQuantum	USA*, Photonics Qubits, Chips, Cabinets, and Superconducting Photon Detectors

Quantum Computing 현황 2. 하드웨어 – 주요 항목

- Qbit 제조 방식은 다양하며, 각자 장단점이 있음
- Coherence Times and Qubit Connectivity
- Error Correction
- Interface with Classical Hardware
- Software Development Environment

Quantum Computing 현황 1. 동향 – 주가 등락

2025.01.08.

- Quantum computing stocks take a hit as Nvidia CEO predicts long road ahead
- <https://www.reuters.com/technology/quantum-computing-stocks-take-hit-nvidia-ceo-predicts-long-road-ahead-2025-01-08/>



2025.01.16.

- NVIDIA's Quantum Day at GTC 2025 to Highlight Latest Advances, Future Potential of Quantum Computing
- <https://thequantuminsider.com/2025/01/15/nvidias-quantum-day-at-gtc-2025-to-highlight-latest-advances-future-potential-of-quantum-computing/>



• 불확실성

• 기대감

Quantum Computing 현황 3. 소프트웨어 – 알고리즘

• Quantum Computing을 적용하기 가장 좋은 분야와 이유는? *Feat. ChatGPT*

카테고리	Use Cases	적합한 이유
최적화 문제 (Optimization)	물류 네트워크 최적화, 금융 포트폴리오 최적화, 교통 경로 최적화	병렬 계산 능력
제약 및 신약 개발 (Pharmaceutical R&D)	분자 시뮬레이션, 약물-표적 상호작용 분석, 신약 개발 시간 단축	분자 수준에서 복잡한 상호작용을 정확히 모델링
암호학 및 사이버 보안 (Cryptography)	기존 암호 체계의 해독, 양자 암호화 기술 개발	RSA와 같은 기존 암호화 알고리즘을 빠르게 해독
기계 학습 및 인공지능 (Machine Learning & AI)	이상 탐지, 데이터 분류, 딥러닝 모델의 학습 가속화	복잡한 다변수 계산을 동시에 수행
재료 과학 (Material Science)	새로운 재료 설계, 나노 기술 개발, 에너지 저장 장치 연구	원자 및 분자 수준에서의 상호작용을 시뮬레이션
기후 모델링 및 날씨 예측 (Weather Prediction)	기후 변화 예측, 극단적 기상 현상 시뮬레이션	대규모 데이터 세트와 복잡한 상호작용을 빠르게 분석
금융 분석 (Finance)	리스크 분석, 시장 시뮬레이션, 금융 상품 가격 책정	복잡한 상호작용을 모델링하고 분석

Quantum Computing 현황 3. 소프트웨어 – 알고리즘 ('23.02)

- **EQUALITY CONSORTIUM** SELECTED BY THE EU'S HORIZON EUROPE PROGRAM TO DEVELOP QUANTUM ALGORITHMS FOR INDUSTRIAL APPLICATIONS.

<https://www.capgemini.com/news/press-releases/equality-consortium-selected-by-the-eus-horizon-europe-program-to-develop-quantum-algorithms-for-industrial-applications/>

Members

- Airbus
- Capgemini
- Da Vinci Labs
- Fraunhofer Institute for Electronic Nano Systems
- German Aerospace Center
- INRIA
- Leiden University
- Pasqal

Selected 8 Use Cases

- 항공기 공기역학 airfoil aerodynamics
- 배터리 설계 battery design
- 유체역학 fluid dynamics
- 우주 임무 최적화 space mission optimization
- 재료 설계 materials design
- 다분야 최적화 multidisciplinary optimization
- 우주 데이터 분석 space data analysis
- 연료 전지 설계 fuel cell design

* **EQUALITY** stands for Efficient 'QUAntum ALgorithms for IndusTrY'

Quantum Computing 현황 4. 상용화와 불확실성 1

- Australia is making a billion-dollar bet on a 'useful' quantum computer. So what are we buying?
- <https://www.abc.net.au/news/science/2024-10-04/psi-quantum-quantum-computer-investment-billion-dollars/104394996>



- PsiQuantum은 기존 컴퓨터로는 해결할 수 없는 문제를 해결할 수 있는 세계 최초의 유용한 양자 컴퓨터가 될 것이라고 밝혔습니다.

Simon Devitt

퀀텀 물리학자

University of
Technology Sydney

- 상업적 유용성에 필요한 것과 PsiQuantum이 해당 기계에 대해 공개적으로 말한 매우 불분명한 '사양' 사이에는 엄청난 차이가 있습니다

- PsiQuantum은 자사 컴퓨터에 백만 개의 "큐비트"가 있다고 밝혔지만, 정확히 무슨 뜻인지는 복잡합니다. 양자 컴퓨팅 성능을 결정하는 것은 큐비트 수를 세는 것만으로는 충분하지 않습니다.

Quantum Computing 현황 4. 상용화와 불확실성 2

- Australia is making a billion-dollar bet on a 'useful' quantum computer. So what are we buying?
- <https://www.abc.net.au/news/science/2024-10-04/psi-quantum-quantum-computer-investment-billion-dollars/104394996>

Howard Wiseman
Griffith University

- 그리피스 대학의 양자 물리학 센터 소장인 하워드 와이즈먼은 PsiQuantum과 협력하고 있으며, Qubit은 "1,000에서 10,000"까지 가능할 것이라고 말했습니다.

Stephen Bartlett
Simon Devitt
퀀텀 물리학자
University of
Technology Sydney

- 시드니 대학의 양자 물리학자이자 새로운 양자 산업 허브인 Quantum Australia의 이사인 스티븐 바틀릿은 훨씬 더 높은 수치를 제시했습니다.

- UTS의 사이먼 데빗은 그 숫자(Qubit)가 "수십 개"에 달할 수 있다고 추측했습니다.

 PsiQuantum

- 샤드볼트 박사는 ABC에 "우리는 첫 번째 Machine에 약 200개의 논리적 큐비트를 탑재할 계획입니다."라고 말했습니다.

Quantum Computing 현황 4. 상용화와 불확실성 3

- Australia is making a billion-dollar bet on a 'useful' quantum computer. So what are we buying?
- <https://www.abc.net.au/news/science/2024-10-04/psi-quantum-quantum-computer-investment-billion-dollars/104394996>



- "첫 번째 Machine은 양자 컴퓨팅의 결승선이 아니라 출발선입니다."
- 이 Machine은 암호를 해독하지 않을 것입니다. "1세대 Machine에서 의미 있는 암호 해독을 기대하지 않습니다."
- 또한 항구나 운송 경로와 같은 최적화 문제를 해결하지도 않을 것입니다. "최적화 유형의 애플리케이션의 경우, 나중에 더 큰 Machine이 필요합니다."
- 기계 학습에는 절대 사용되지 않을 것입니다. 이러한 알고리즘에 대한 깊이와 성숙도가 없습니다.
- 새드볼트 박사는 브리즈번 컴퓨터가 주로 계산 화학, 즉 원자와 분자가 주변 환경과 어떻게 상호 작용하는지 모델링하는 데 사용될 것으로 예상합니다. "우리가 해결하고자 하는 문제는 기본적으로 응집 물질 시스템과 소분자 화학입니다."

Quantum Computing 기초 이론 1. Quantum Computing – History

1925년

- Quantum Mechanics 탄생 – 하이젠베르크 Matrix Mechanics

1926년

- Quantum Mechanics 탄생 – 슈뢰딩거(Erwin Schrödinger) 파동 역학(Wave Mechanics)

1927년

- 불확정성 원리 – 위치 vs 운동량 (하이젠베르크)

1935년

- Einstein–Podolsky–Rosen paradox – Entanglement (양자얽힘) 시초

1981년

- 리처드 파인만(Richard Feynman) MIT 컨퍼런스 "양자역학 시스템 고전 컴퓨터로 시뮬레이션 불가 " 주장

1985년

- 데이비드 도이치(David Deutsch) - Deutsch Algorithm 제안으로 Quantum Computer 가능성 시초

1992년

- 데이비드 도이치(David Deutsch) 리차드 조사(Richard Jozsa) - Deutsch-Jozsa Algorithm

1994년

- 피터 쇼어(Peter Shor)가 쇼어 알고리즘을 개발 → 양자 컴퓨터가 큰 수의 소인수 분해를 빠르게 수행

2016년

- IBM 5 Qubit 양자컴퓨터 사용 클라우드를 통한 서비스 제공 가능성 보임

2019년

- Google 개발 양자컴퓨터 Sycamore를 통하여 양자 우월성 달성 발표 - 200초 vs 기존 1만년

Quantum Computing 기초 이론 1. Quantum Mechanics – 슈뢰딩거 이론

- Schrödinger equation
- https://en.wikipedia.org/wiki/Schr%C3%B6dinger_equation

$$i\hbar \frac{d}{dt} |\Psi\rangle = \hat{H} |\Psi\rangle$$



$$i\hbar \frac{\partial}{\partial t} \psi(\mathbf{r}, t) = \hat{H} \psi(\mathbf{r}, t)$$

$\psi(\mathbf{r}, t)$

wave function,

$\hat{H}$ Hamiltonian

$\hbar$

Planck constant,

$\hat{H}$

operator.

- Particle과 Wave의 기존 이론의 연결
 - Least Action Principles : Particle
 - Ferma's Least Time Principle : Wave
- 물질은 파동성과 입자성을 모두 가진다
 - 입자의 시간에 따른 상태 변화를 기술하며, 그 결과를 통해 입자가 특정 위치에 있을 확률을 예측할 수 있습니다
- 파동 함수에서 r 은 위치를 t 는 시간을 나타내어, 특정 시간에 위치에 존재할 확률을 가지고 물질을 기술
- Werner Heisenberg의 Matrix mechanics와 동일

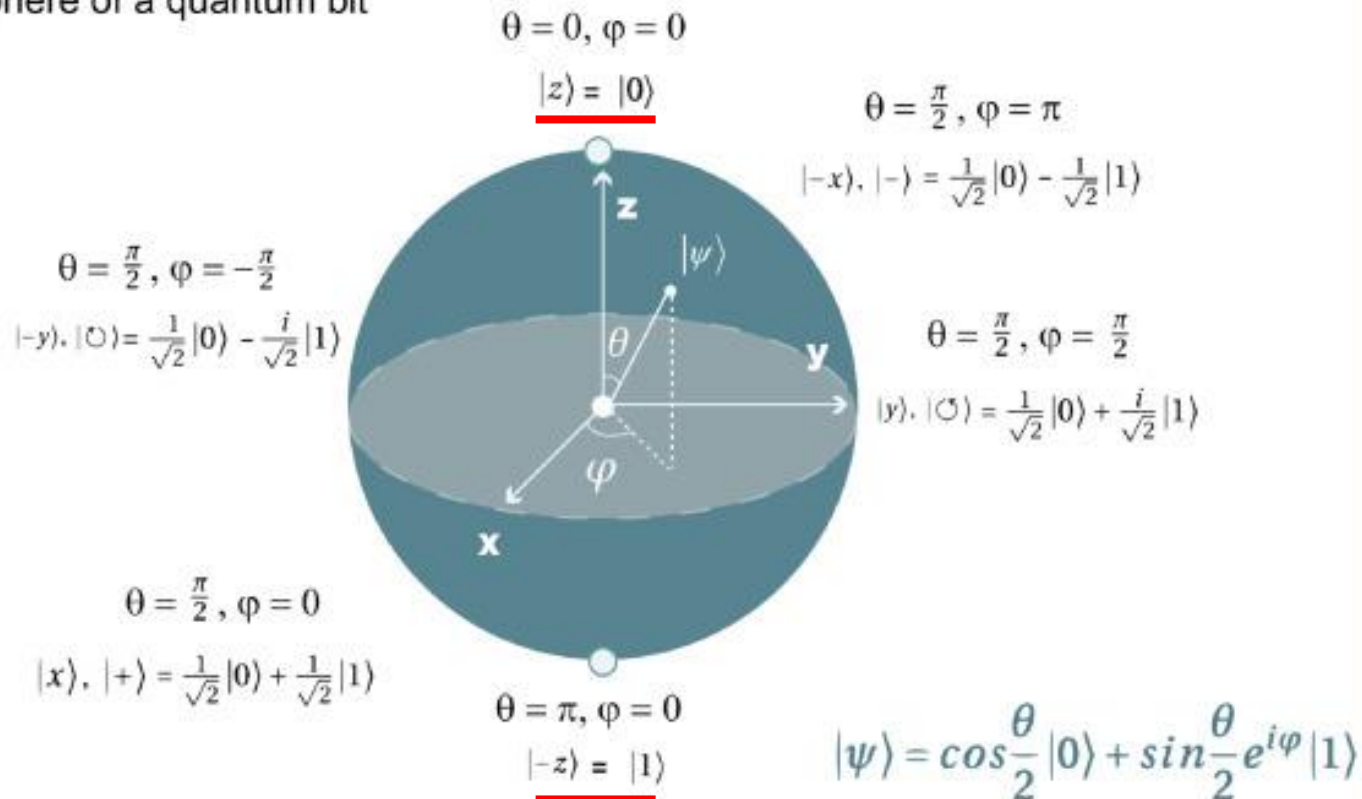
Quantum Computing 기초 이론 1. Quantum Computer – 잘못된 상식

- Qbit는 항상 0과 1을 동시에 가진다 → Superposition 상태만
- Qbit는 0과 1을 동시에 저장한다 → 상태이며, 확률의 분포
- Qbit가 많을 수록 계산 속도가 빨라진다 → Qbit간의 관계성 구현도에 따라 다름
- Qbit는 고전적인 Bit와 전혀 다르다 → 확장으로 보아야 한다
- 양자 얽힘으로 초고속 정보 전달이 가능하다 → 다수의 상관성을 의미하는 개념

Quantum Computing 기초 이론 1. Quantum Computer – Qbit (Qubit)

- Qubit 표현과 Bracket notation (Dirac notation)

Bloch Sphere of a quantum bit



Bra-ket notation

A **bra** is of the form $\langle f|$, it denotes a **linear form** f

A **ket** is of the form $|v\rangle$, it denotes a **vector**, v ,

$$|0\rangle \rightarrow |0\rangle e^{i\alpha} \equiv |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle \rightarrow |1\rangle e^{i\beta} \equiv |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Superposition

$$|\psi\rangle = C_0|0\rangle + C_1|1\rangle$$

$$|C_0|^2 + |C_1|^2 = 1$$

Quantum Computing 기초 이론 1. Quantum Computer – Qbit (Qubit)

- Qubit 표현과 Bracket notation (Dirac notation) : 1 Qubits

1개 Qubit의 상태

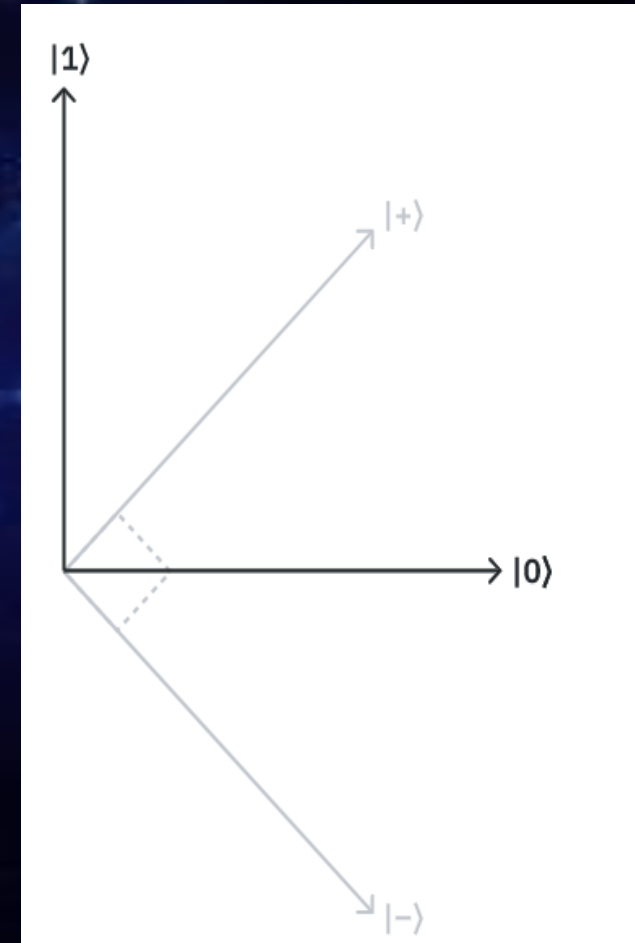
→ 행렬 위부터 순서대로
0, 1 상태 확률

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Superposition & Phase

$$|+\rangle = \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$|-\rangle = \sqrt{\frac{1}{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



Quantum Computing 기초 이론 1. Quantum Computer – Qbit (Qubit)

- Qubit 표현과 Bracket notation (Dirac notation) : 2 Qubits

2개 Qubit의 상태

동시에 표현하는 Notation

→ 행렬 위부터 순서대로 상태 확률

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle = \begin{bmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{bmatrix}$$

$$|+0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle), \quad |\Phi^+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle).$$

'ket plus naught'

'ket phi plus'

Quantum Computing 기초 이론 1. Quantum Computer – Qbit (Qubit)

- Qubit 표현과 Bracket notation (Dirac notation) : 2 Qubits

2개 Qubit이 독립
각 상태 확률과 곱이
4 상태의 확률이 됨

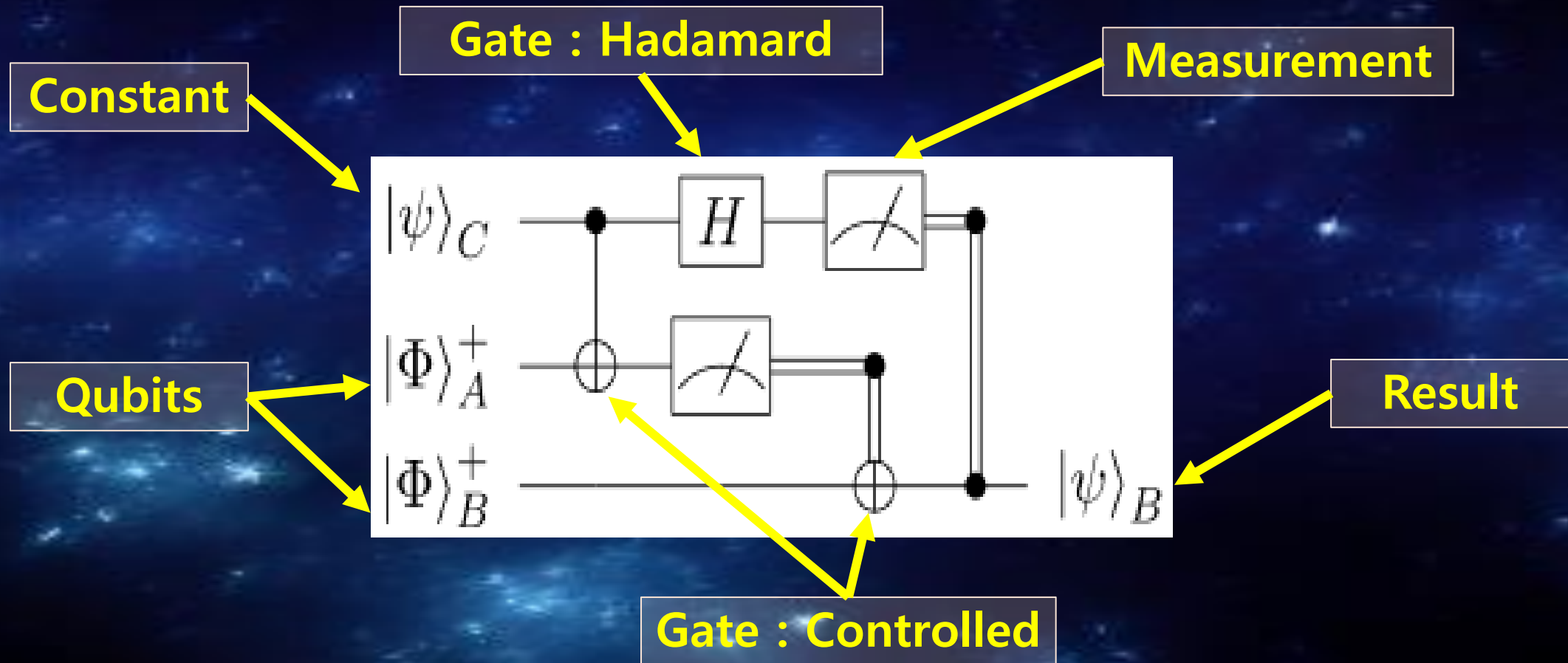
$$|ba\rangle = \begin{bmatrix} b_0a_0 \\ b_0a_1 \\ b_1a_0 \\ b_1a_1 \end{bmatrix}$$

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

얽힘 Entangled

Quantum Computing 기초 이론 1. Quantum Circuit Model

- Quantum Computer를 이루는 Circuit 구성 요소 : Qbits, Q-Gates, Measurements
- https://en.wikipedia.org/wiki/Quantum_circuit



Quantum Computing 기초 이론 1. Quantum Circuit Model

- 참고 : Hadamard Gate
- https://en.wikipedia.org/wiki/Hadamard_transform#Quantum_computing_applications

Superposition

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

Hadamard gate operations [\[edit\]](#)

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

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

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

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

Quantum Computing 기초 이론 1. Quantum Supremacy

- Quantum Supremacy → 양자 우월성
- Quantum Computer가 특정 계산에서 고전적 컴퓨터를 능가하는 것
- 실제적인 예, 혹은 개념적으로 입증된 알고리즘에 적용
- 2019년 Google의 Sycamore가 54개 Qubit으로 일반 컴퓨터가 1만년 걸릴 계산을 200초에 수행함
- 현재 Superposition과 Entanglement 양자역학이 토대가 되고 있음

Quantum Computing 기초 이론 1. Background Algorithms – Deutsch Algorithm

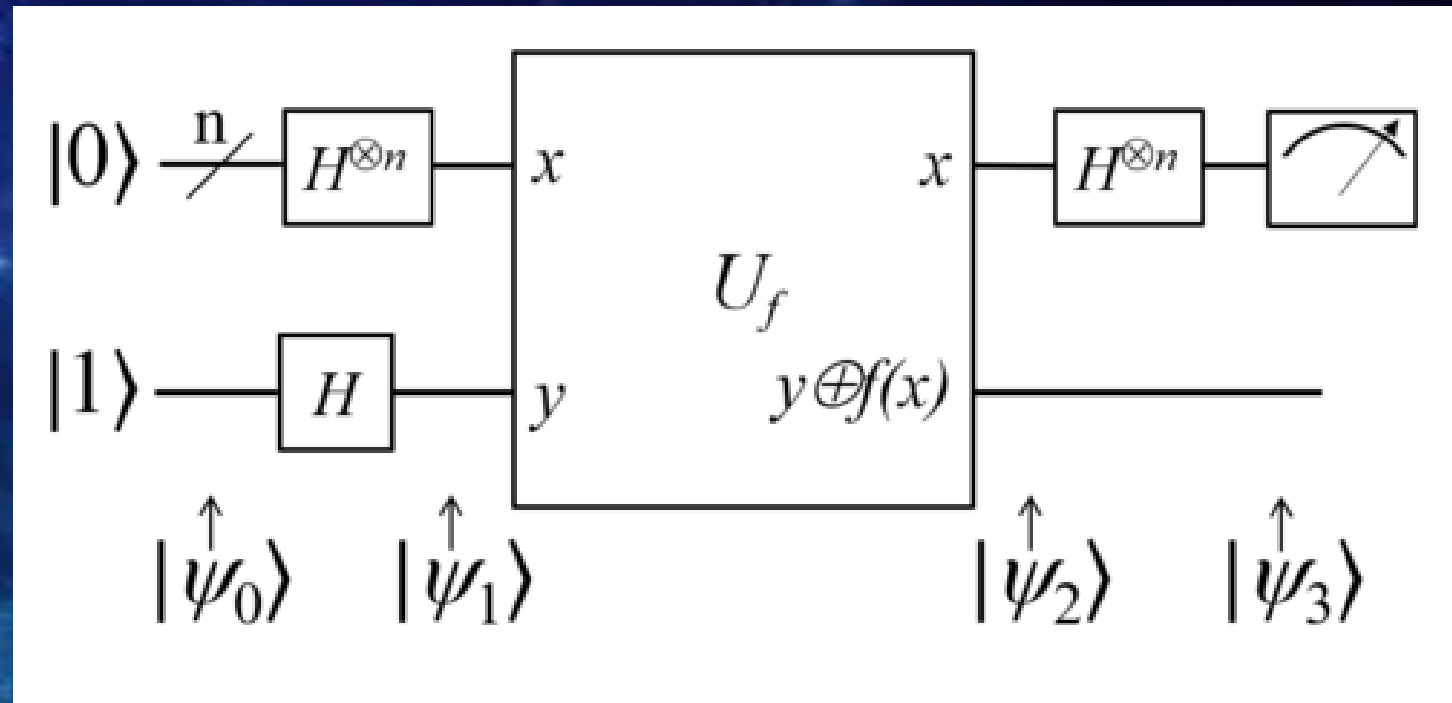
- 다음 함수에서 n 은 입력 bit의 개수를 나타낼 때, 함수가 상수 함수인지 아닌지 알아내는 문제이며, $n=1$ 일 때 Deutsch Algorithm이라 함

https://en.wikipedia.org/wiki/Deutsch%E2%80%93Jozsa_algorithm

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

- Classical Solution**

- Bit 수가 1이므로 함수의 입력 가지 수는 총 $2^{(n-1)+1}$ 개, 즉 2가지 (0,1) 이다
- 따라서 2번 계산하여 결과를 보면 판단이 가능하다



Quantum Computing 기초 이론 1. Background Algorithms – Deutsch-Jozsa Algorithm

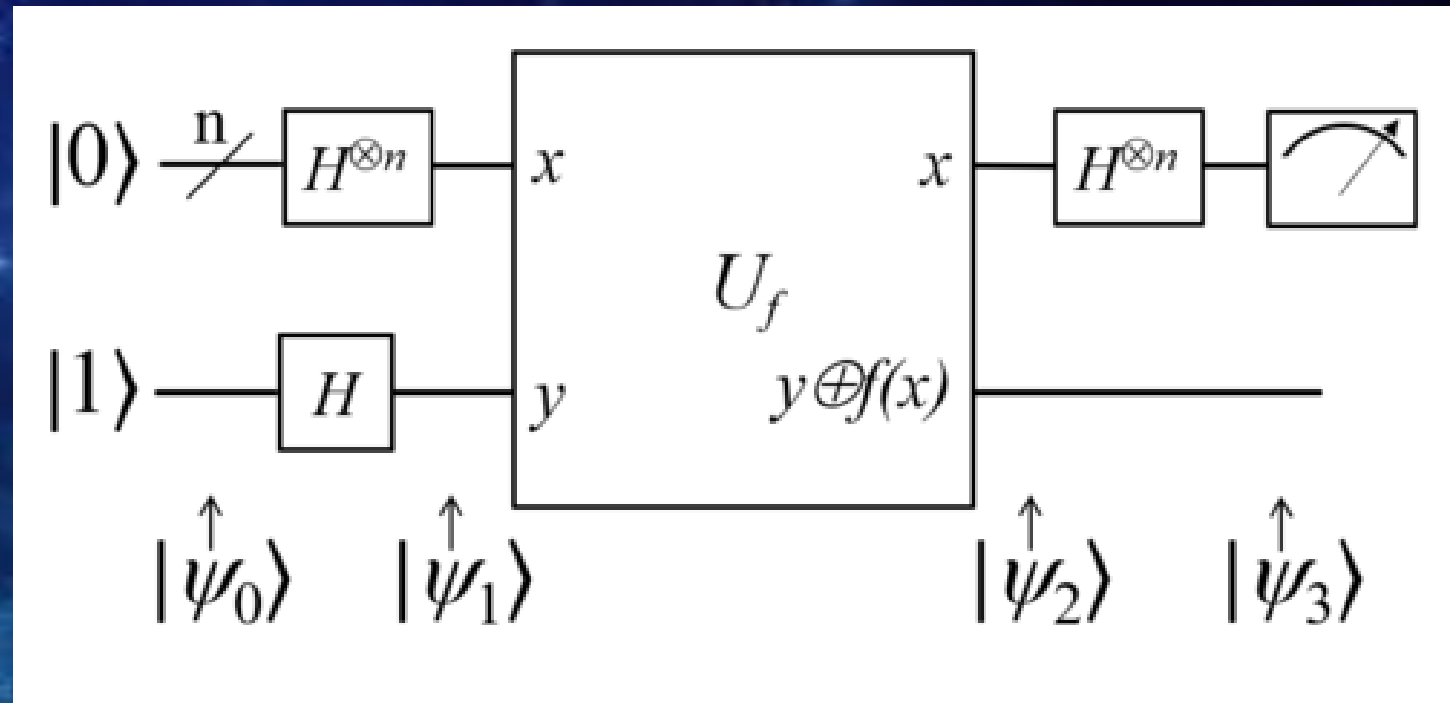
- 다음 함수에서 n 은 입력 bit의 개수를 나타낼 때, 함수가 상수 함수인지 아닌지 알아내는 문제이며, 모든 n 을 Deutsch-Jozsa Algorithm이라 함

https://en.wikipedia.org/wiki/Deutsch%E2%80%93Jozsa_algorithm

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

- Classical Solution**

- Bit 수가 n 이므로 함수의 입력 가지 수는 총 $2^{n-1}+1$ 개임
- 따라서, 2^n 의 지수승 회 계산하여 결과를 보면 판단이 가능하다

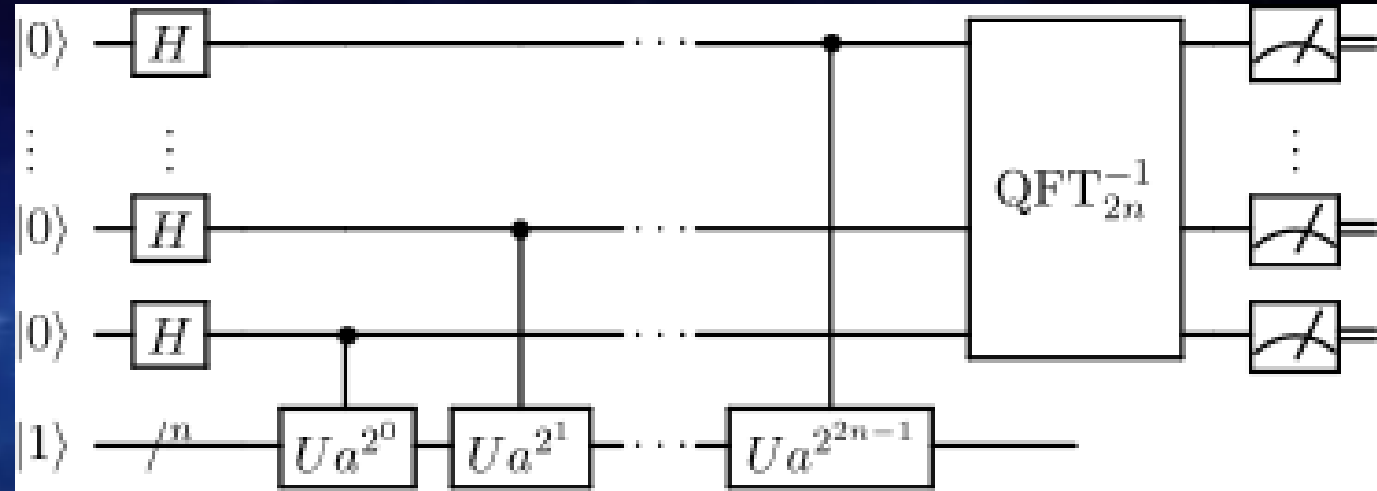


Quantum Computing 기초 이론 1. Background Algorithms – Shor's

- 소인수 분해를 빠르게 처리 가능한 알고리즘 → RSA, Some Key Exchanges
- https://en.wikipedia.org/wiki/Shor%27s_algorithm

• Classical Solution

- 2로 나누기 시작하여 1씩 증가 → 나머지 0일때까지 진행 → 몫을 가지고 다시 수행
- N이라는 수에 대해서, 2의 $\log N$ 지수승의 step들이 필요하며, N 자리수가 커짐에 따라 지수적으로 필요 시간이 증가
- RCA 300-digit → Billions of Years (classical super computer)



Quantum subroutine in Shor's algorithm

- N이라는 수에 대해 $\log N$ 에 비례하는 step들 필요 → Polynomial Time
- RCA 300-digit → 수시간 필요

Quantum Computing 코딩 환경 1. IBM Qiskit using Python 1

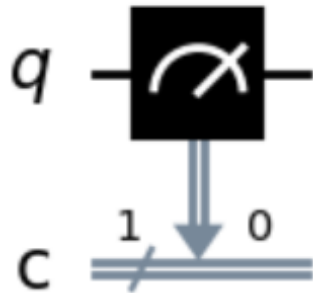
- Library : qiskit

```
from qiskit import QuantumCircuit, IBMQ, execute
from qiskit_aer import AerSimulator
```

```
from qiskit import transpile
from qiskit.visualization import plot_histogram
```

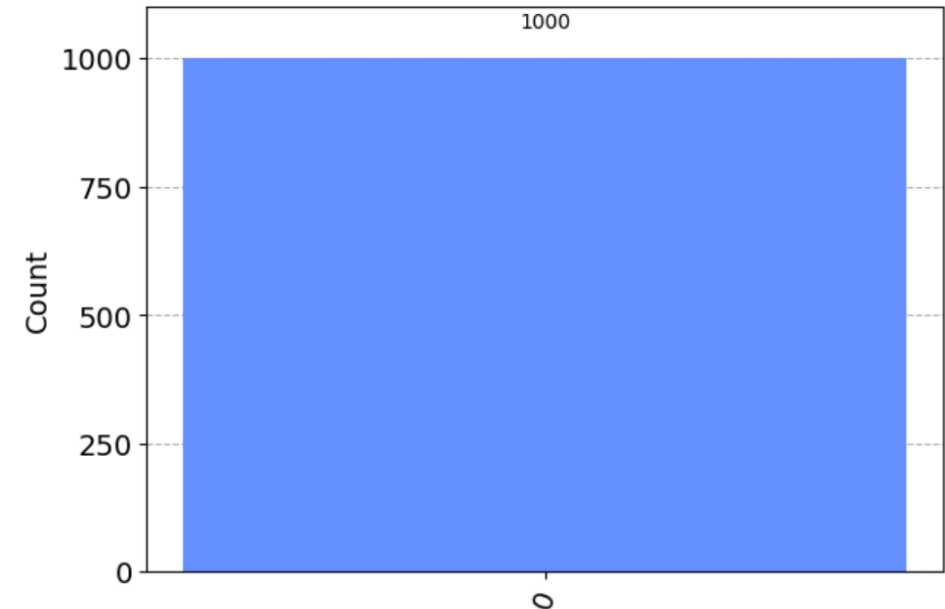
Single qubit gate

```
circ = QuantumCircuit(1,1)
circ.measure(0,0)
circ.draw('mpl')
```



```
backend = AerSimulator() #
qc = transpile(circ, backend) #
job = backend.run(qc, shots = 1000)
result = job.result()
counts = result.get_counts()
print(counts)
plot_histogram(counts)
```

{'0': 1000}



Quantum Computing 코딩 환경 1. IBM Qiskit using Python 2

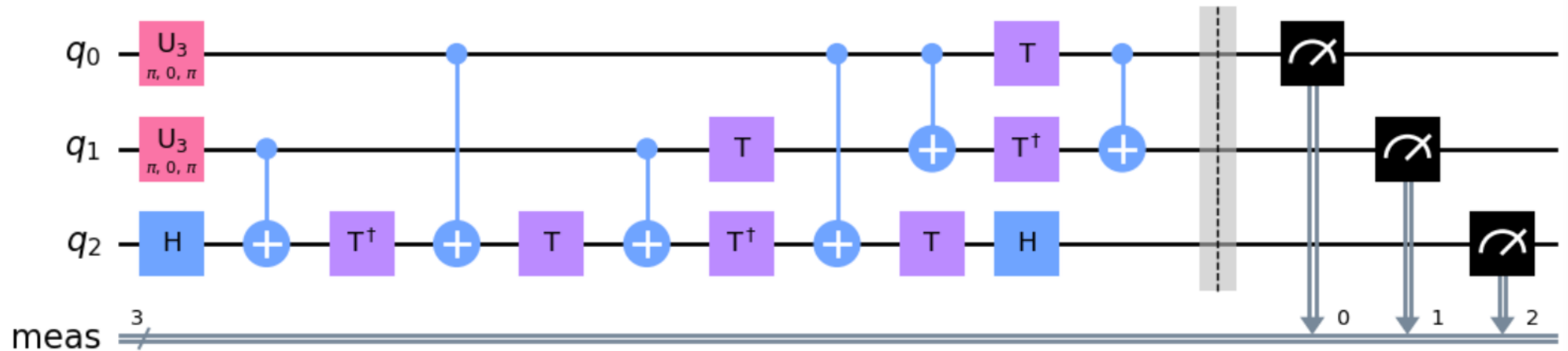
- Toffoli gate : CCNOT gate ("controlled-controlled-not")

https://en.wikipedia.org/wiki/Toffoli_gate

Toffoli gate

```
[38]: circ = QuantumCircuit(3)
      circ.x([0,1])
      circ.ccx(0,1,2)
      circ.measure_all()
      circ.decompose().draw('mpl')
```

[38]:



Quantum Computing 코딩 환경 1. IBM Qiskit using Python 3

- Toffoli gate
- Simulator Result

```
backend = AerSimulator()
qc = transpile(circ, backend)
job = backend.run(qc, shots = 1000)
result = job.result()
counts = result.get_counts()
print(counts)
#plot_histogram(counts)
```

```
{'111': 1000}
```

```
backend = provider.get_backend('ibmq_guadalupe')
circ_transpiled = transpile(qc, backend)
execute_ccx = execute(circ_transpiled, backend, shots= 1000)
```

Quantum Computing 코딩 환경 1. IBM Qiskit using Python 4

- Toffoli gate

→ Quantum Computer Result (2023년)

```
result_ccx = execute_ccx.result()
counts_ccx = result_ccx.get_counts()
print("110->111", counts_ccx)
```

```
110->111 {'001': 24, '010': 19, '011': 88, '100': 3, '101': 31, '110': 28, '111': 807}
```

```
result_ccx = execute_ccx.result()
counts_ccx = result_ccx.get_counts()
print("110->111", counts_ccx)
```

```
110->111 {'000': 2, '001': 26, '010': 19, '011': 76, '100': 3, '101': 39, '110': 34, '111': 801}
```

```
result_ccx = execute_ccx.result()
counts_ccx = result_ccx.get_counts()
print("110->111", counts_ccx)
```

```
110->111 {'000': 3, '001': 27, '010': 18, '011': 113, '100': 3, '101': 39, '110': 37, '111': 760}
```

Quantum Computing 코딩 환경 1. IBM Qiskit using Python 5

- Adder using Quantum Gates

➔ Sub-Block 코딩

```
carry = QuantumCircuit(4)
carry.ccx(1,2,3)
carry.cx(1,2)
carry.ccx(0,2,3)

carry_inv = QuantumCircuit(4)
carry_inv.ccx(0,2,3)
carry_inv.cx(1,2)
carry_inv.ccx(1,2,3)

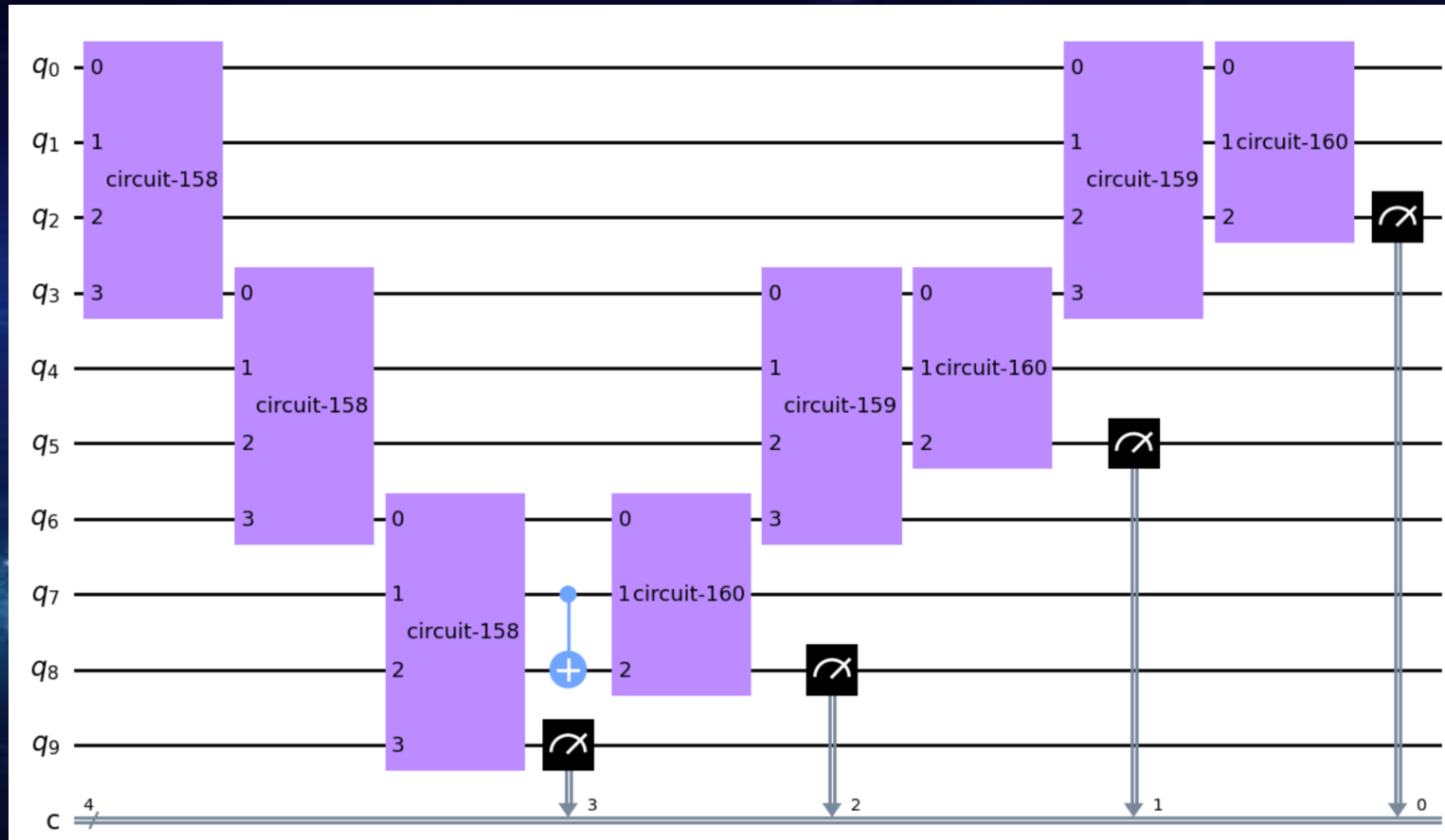
sumc = QuantumCircuit(3)
sumc.cx(1,2)
sumc.cx(0,2)
```

```
adder3 = QuantumCircuit(10,4)
adder3.append(carry,[0,1,2,3])
adder3.append(carry,[3,4,5,6])
adder3.append(carry,[6,7,8,9])
adder3.cx(7,8)
adder3.append(sumc,[6,7,8])
adder3.append(carry_inv,[3,4,5,6])
adder3.append(sumc,[3,4,5])
adder3.append(carry_inv,[0,1,2,3])
adder3.append(sumc,[0,1,2])

adder3.measure(2,0)
adder3.measure(5,1)
adder3.measure(8,2)
adder3.measure(9,3)
adder3.draw('mpl')
```

Quantum Computing 코딩 환경 1. IBM Qiskit using Python 6

- Adder using Quantum Gates
→ 4 bit + 4 bit adder (0001 + 0001 case)



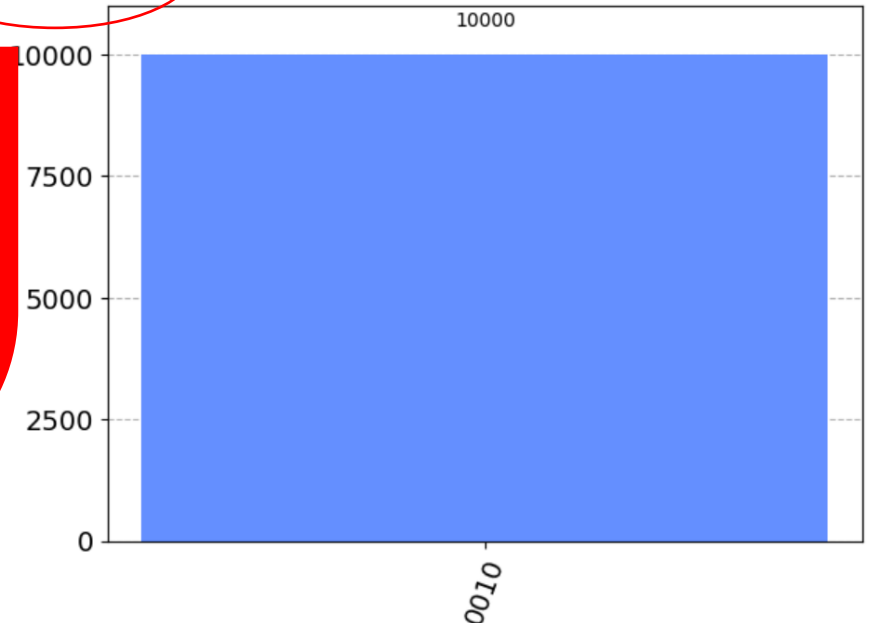
Quantum Computing 코딩 환경 1. IBM Qiskit using Python 7

- Adder using Quantum Gates
→ 4 bit + 4 bit adder (0001 + 0001 case)

```
# 1+1
circ = QuantumCircuit(10,4)
circ.x(1)
circ.x(2)
circ.append(adder3,range(10),range(4))
```

```
{'0010': 10000}
```

```
backend = AerSimulator()
qc = transpile(circ, backend)
job = backend.run(qc, shots = 10000)
result = job.result()
counts = result.get_counts()
print(counts)
plot_histogram(counts,
{'0010': 10000})
```



Quantum Computing 코딩 환경 1. IBM Qiskit using Python 8

- Adder using Quantum Gates
→ Real Quantum Computer Results

```
backend = provider.get_backend('ibmq_guadalupe')
circ_transpiled = transpile(circ, backend)
execute_add11 = execute(circ_transpiled, backend, shots= 10000)
```

```
result_add11 = execute_add11.result()
counts_add11 = result_add11.get_counts()
print("1+1, 0010",counts_add11)
```

```
1+1, 0010 {'0000': 2980, '0001': 926, '0010': 2022, '0011': 761, '0100': 716, '0101': 252, '0110': 540, '0111': 205, '1000': 506, '1001': 166, '1010': 420, '1011': 155, '1100': 156, '1101': 47, '1110': 101, '1111': 47}
```

```
result_add11 = execute_add11.result()
counts_add11 = result_add11.get_counts()
print("1+1, 0010",counts_add11)
```

```
1+1, 0010 {'0000': 2671, '0001': 957, '0010': 1795, '0011': 681, '0100': 973, '0101': 366, '0110': 689, '0111': 284, '1000': 447, '1001': 171, '1010': 354, '1011': 142, '1100': 203, '1101': 64, '1110': 144, '1111': 59}
```

```
result_add11 = execute_add11.result()
counts_add11 = result_add11.get_counts()
print("1+1, 0010",counts_add11)
```

```
1+1, 0010 {'0000': 2501, '0001': 817, '0010': 1735, '0011': 554, '0100': 1497, '0101': 431, '0110': 858, '0111': 218, '1000': 434, '1001': 137, '1010': 253, '1011': 97, '1100': 224, '1101': 55, '1110': 139, '1111': 50}
```

Quantum Computing 맷음말 1. 정리해 본 입문 절차

- Matrix Mechanics와 Schrödinger equation에 대해 정리
- Linear Algebra 복습 (notation에 적응)
- 기본 알고리즘 Study : Deutsch-Jozsa Algorithm, Shor's Algorithm
- 현재 제공되는 알고리즘 코딩 환경 설치
- 기본 알고리즘 코딩 및 실행 with Simulators
- 기본 알고리즘 코딩 및 실행 with Real Cloud Quantum Computers

Quantum Computing 맺음말 2. 사용 가능한 제공되는 환경

- IBM Qiskit

<https://www.ibm.com/quantum/qiskit>

- Microsoft Azure Quantum

<https://azure.microsoft.com/ko-kr/solutions/quantum-computing>

- Google Cirq

<https://quantumai.google/cirq>

- Xanadu PennyLane

<https://www.xanadu.ai/products/pennylane/>

Quantum Computing 맺음말 3. 입문하면서 고려해 볼 점 정리

- 현재 가능한 알고리즘에 대한 이해

- 기존 암호에 대한 무력화 → Quantum Computing 내성 암호 대응
- 다양한 물질 개발 및 최적화 알고리즘 개발 가능

- 향후 가능한 알고리즘에 대한 가능성으로 현재에 대한 보수적인 접근

- Quantum Mechanics의 발전 → 신규 기본 알고리즘 가능성
- 기존 Quantum Mechanics에 대한 상용 구현 진행 중

- 신규 알고리즘을 개발하는 것은 개발 현황과 밀접한 관계

감사합니다!

Thank you!