Shor’s algorithm

https://youtu.be/2fJQk0UkBpc

정보컴퓨터공학과 송경주

Shor’s algorithm

- 1994년 Peter Shor가 개발한 정수의 소인수를 찾는 양자 알고리즘
- 양자컴퓨터 성능을 사용하여 큰 수의 소인수 분해를 효율적으로 수행 (공개키 암호에 위협이 됨)
- 핵심 연산으로 1. QFT을 통한 양자 주기 찾기 (양자 알고리즘), 2. 찾은 주기를 활용한 소인수 찾기(고전 알고리즘)
- 실직적으로 유효한 숫자를 인수분해하기 위해 많은 큐비트가 필요

Shor’s algorithm

1. 소수 p, q의 곱으로 이루어진 큰 수 N 선택 (N 이 소수인지 확인)

2. N 보다 작은 임의의 난수 a $(1 < a < N - 1)$를 선택한 후 a 와 N이 서로소(relation)인지 확인: gcd(a, N) (서로소: 두 수의 최대 공약수가 1)

3. 주기적인 패턴 찾기($a \bmod \mathrm{N}$의 차수를 찾음, a의 차수는 $a^r \equiv 1$ (mod N)을 만족하는 가장 작은 양의정수 r(주기)) → QFT 를 활용하여 a의 주기 r을 효율적으로 찾음

4. (1) r가 짝수일 때 : $a^{r/2} = -1$ mod N 조건을 만족할 경우의 a는 N의 소인수를 찾을 수 있는 힌트가 됨

   (2) r가 홀수일 때: 2번으로 돌아가 다시 a 선택하여 과정 반복

Shor's algorithm

1. 소수 p, q의 곱으로 이루어진 큰 수 N 선택 (N 이 소수인지 확인)
2. N 보다 작은 임의의 난수 a ($1 < a < N - 1$)를 선택한 후 a 와 N이 서로소(relation)인지 확인: gcd(a, N) (서로소: 두 수의 최대 공약수가 1)
3. 주기적인 패턴 찾기(a mod N의 차수를 찾음, a의 차수는 $a^r \equiv 1$ (mod N)을 만족하는 가장 작은 양의정수 r(주기)) → QFT 를 활용하여 a의 주기 r을 효율적으로 찾음 (양자 컴퓨팅의 핵심연산) : quantum order-finding algorithm
4. (1) r가 짝수일 때 : $a^{r/2} = -1 \bmod N$ 조건을 만족할 경우의 a는 N의 소인수를 찾을 수 있는 힌트가 됨

   (2) r가 홀수일 때: 2번으로 돌아가 다시 a 선택하여 과정 반복

   다항시간 내에 수행 가능

Shor’s algorithm

- QFT (Quantum Fourier Transform)
  - 주기성 찾기에 사용됨

[주기 찾기]

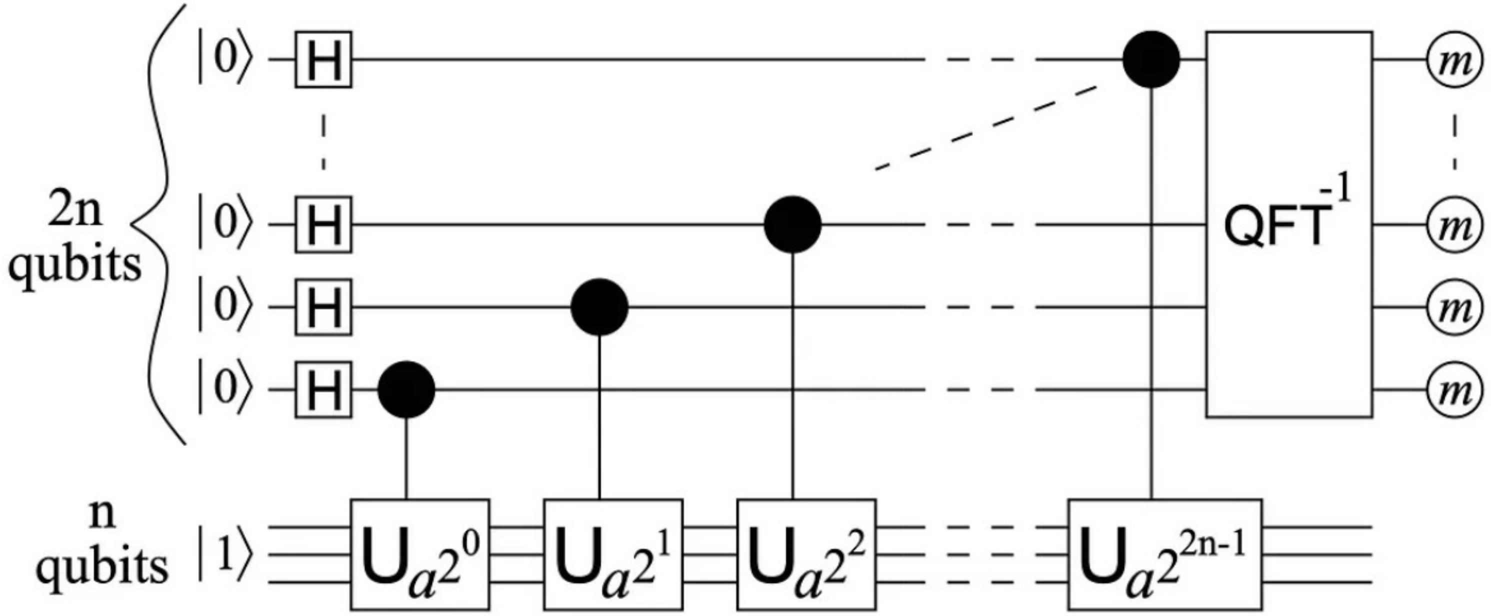
1. a^r mod N에 대해 모든 r을 중첩상태의 큐비트로 표현하여 모든 값을 동시에 나타냄

2. 중첩 상태의 r로 인해 a^r mod N 을 동시에 계산할 수 있음 → 결과를 ancilla 큐비트에 저장

3. <u>결과가 저장된 ancilla 큐비트에 QFT를 적용을 통해 결과의 양자 상태를 주기적인 구조로 변환하여 주기 r에 대한 정보를 찾음</u>

4. 찾은 정보를 통해 실제 주기 r 유추 (연속 분수 알고리즘?)

Shor’s algorithm



$U_{a^{2^k}}$: a^{2^k}을 N으로 나눈 나머지 → $a^{2^k} mod\ N$

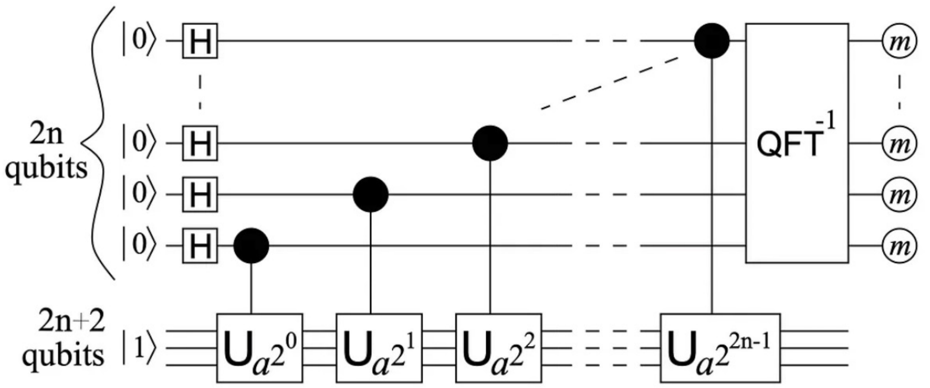
Quantum order-finding algorithm

- 큐비트: 3n개의 큐비트가 필요
- U: 특정 a, N에 대해서만 동작하므로 일반화 할 수 없음 (하드코딩)

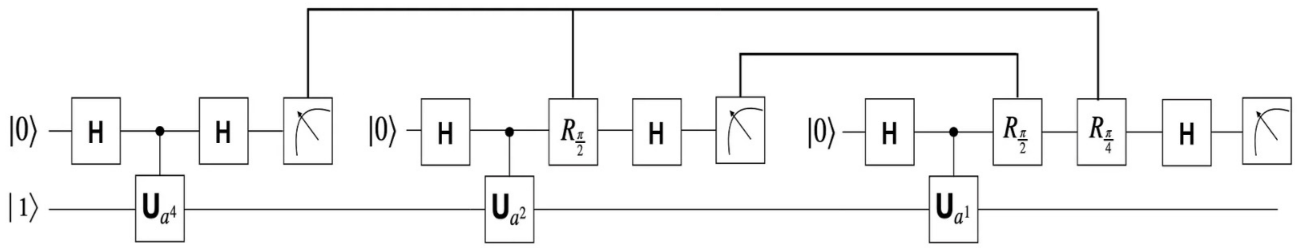
→ 일반적인 방식(해당 방식)에서 큐비트 수를 줄인 방식이 연구되고 있음

Shor’s algorithm

Quantum order-finding algorithm 변형

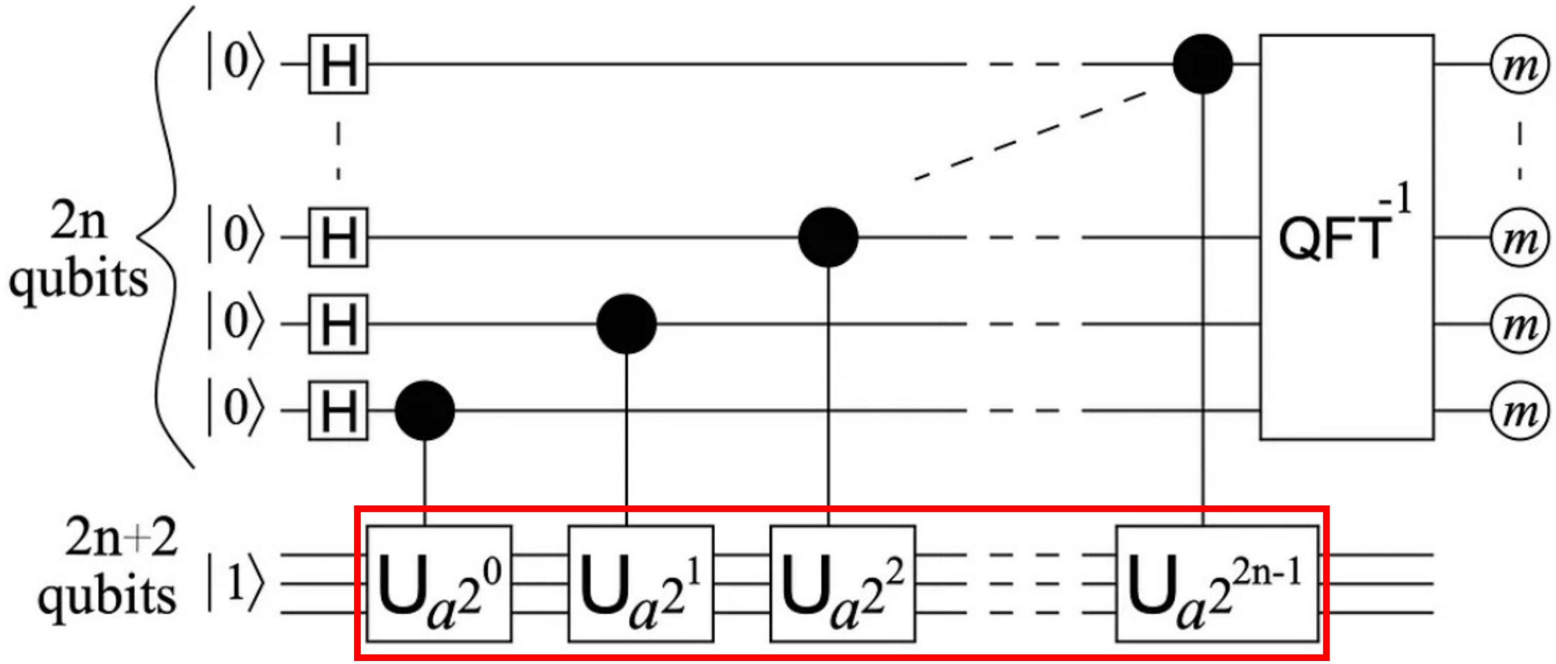


$4n+2$ order-finding algorithm



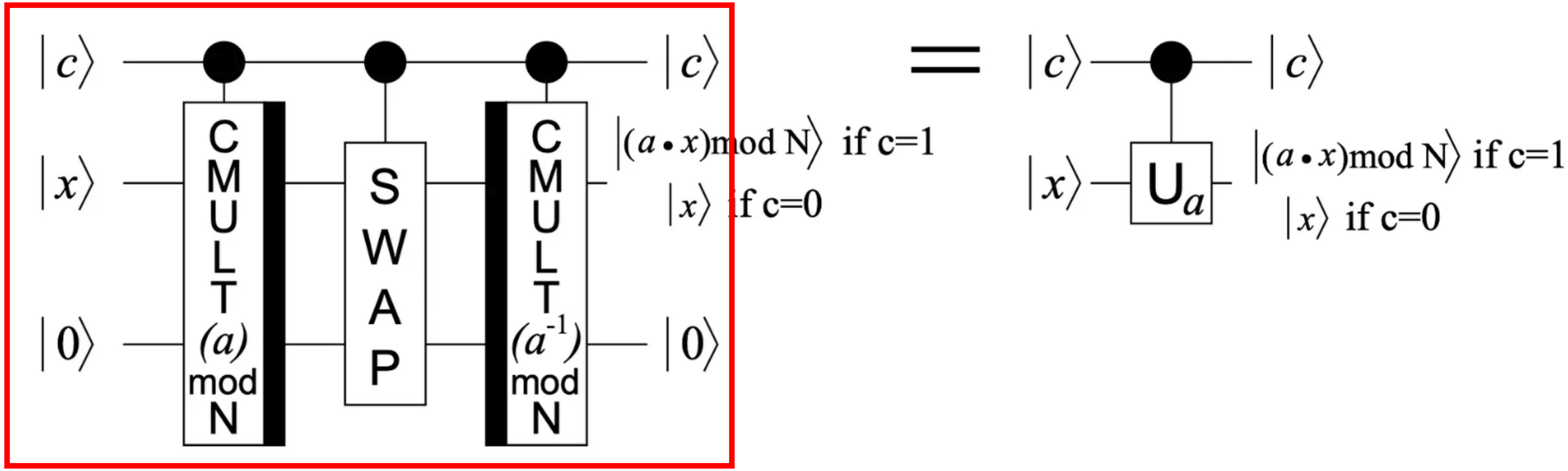
$2n+3$ order-finding algorithm

Shor's algorithm



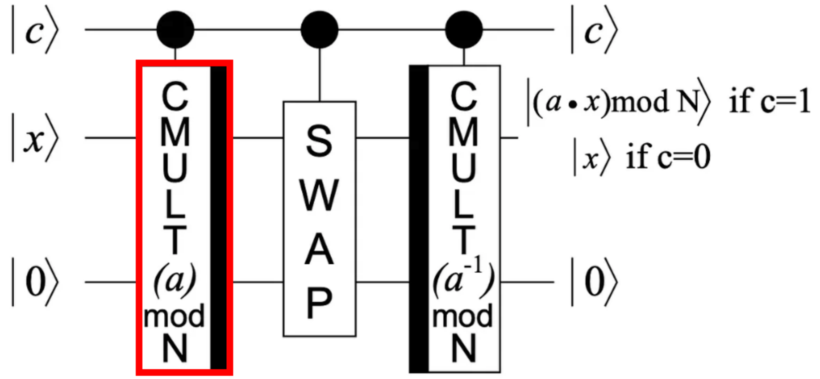
$4n + 2$ order-finding algorithm

Shor’s algorithm

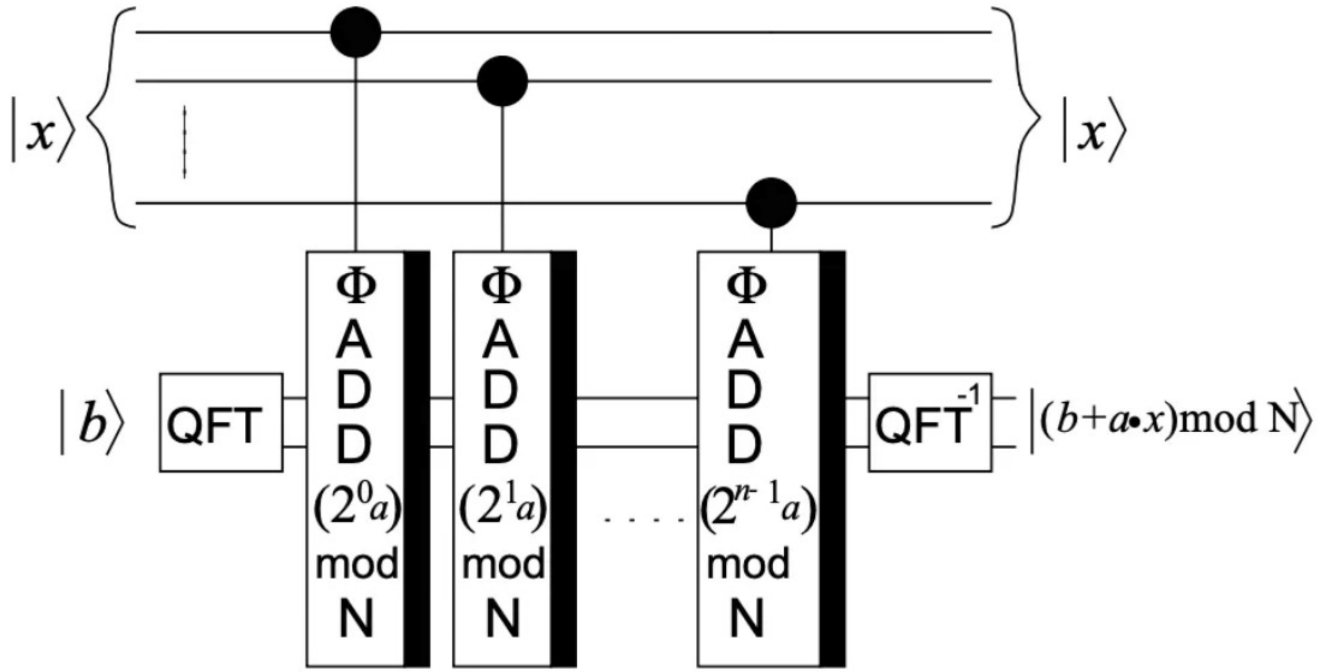


[U(a) gate]: Takes x to ax mod N

Shor's algorithm

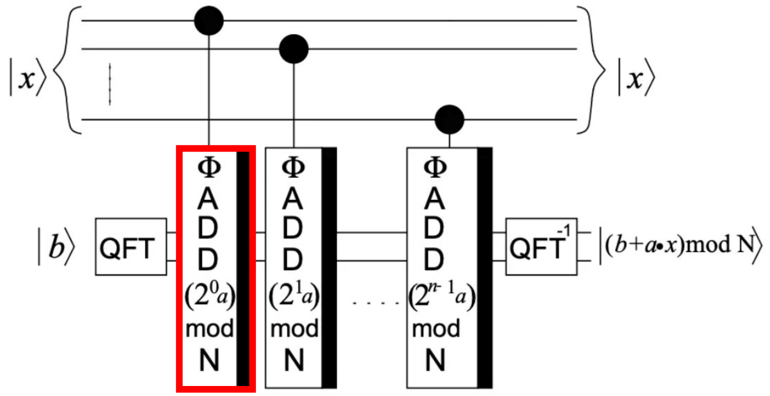


[U(a) gate]: Takes x to ax mod N

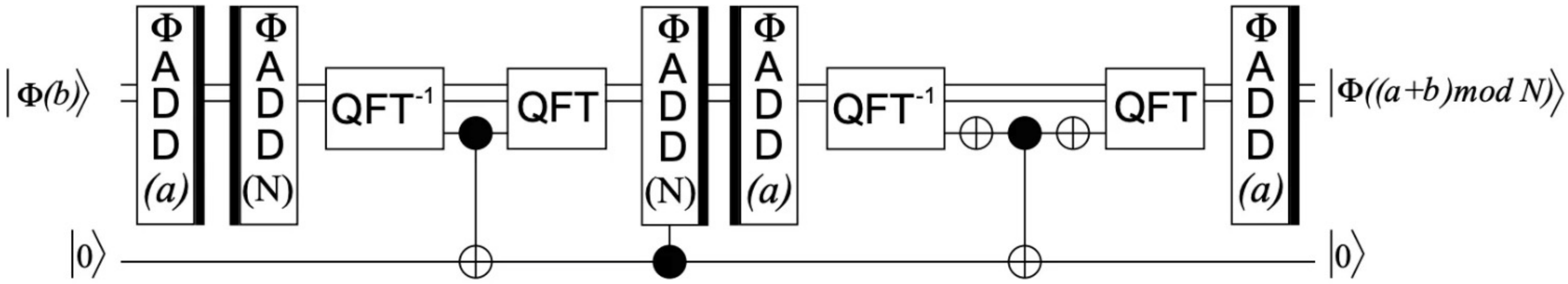


[Multiplier gate] Using the modular adder on each qubit of the x register,
It adds ax to the b register

Shor’s algorithm

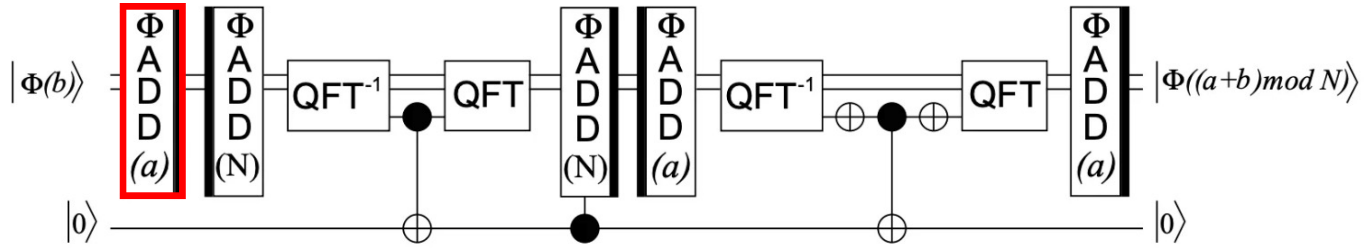


[Multiplier gate] Using the modular adder on each qubit of the x register, It adds ax to the b register

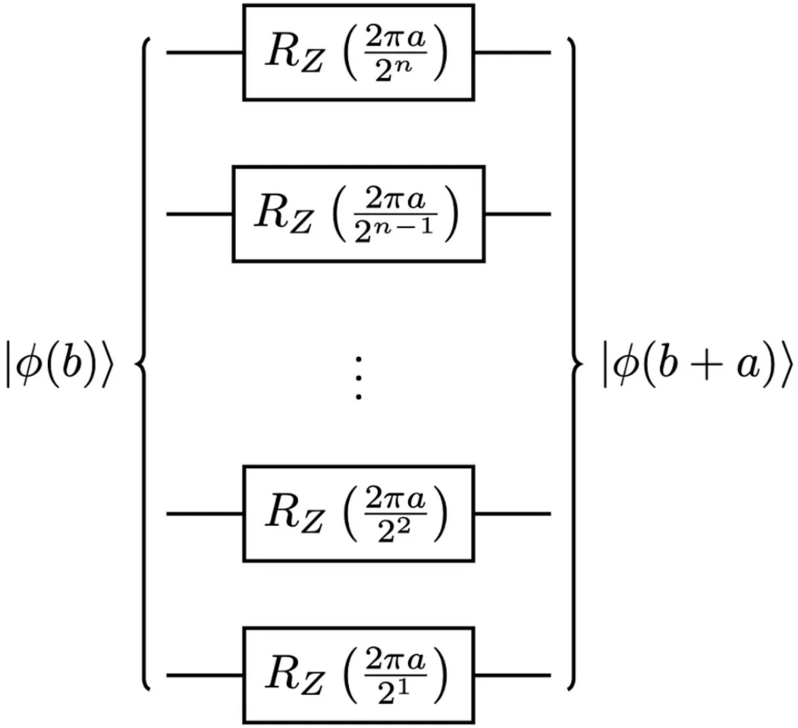


Modular Adder Gate: Using the adder gate and the inverse of the adder gate, it adds a to b modulo N

Shor's algorithm



Modular Adder Gate: Using the adder gate and the inverse of the adder gate, it adds a to b modulo N



Adder Gate: Adds a to the phase of the input

Shor’s algorithm

- QFT (Quantum Fourier Transform)
  - 양자상태 $|x\rangle$를 새로운 양자상태 $|y\rangle$로 변환

$|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle \rightarrow |y\rangle = \sum_{i=0}^{N-1} y_i |i\rangle$:새로운 상태로 변환

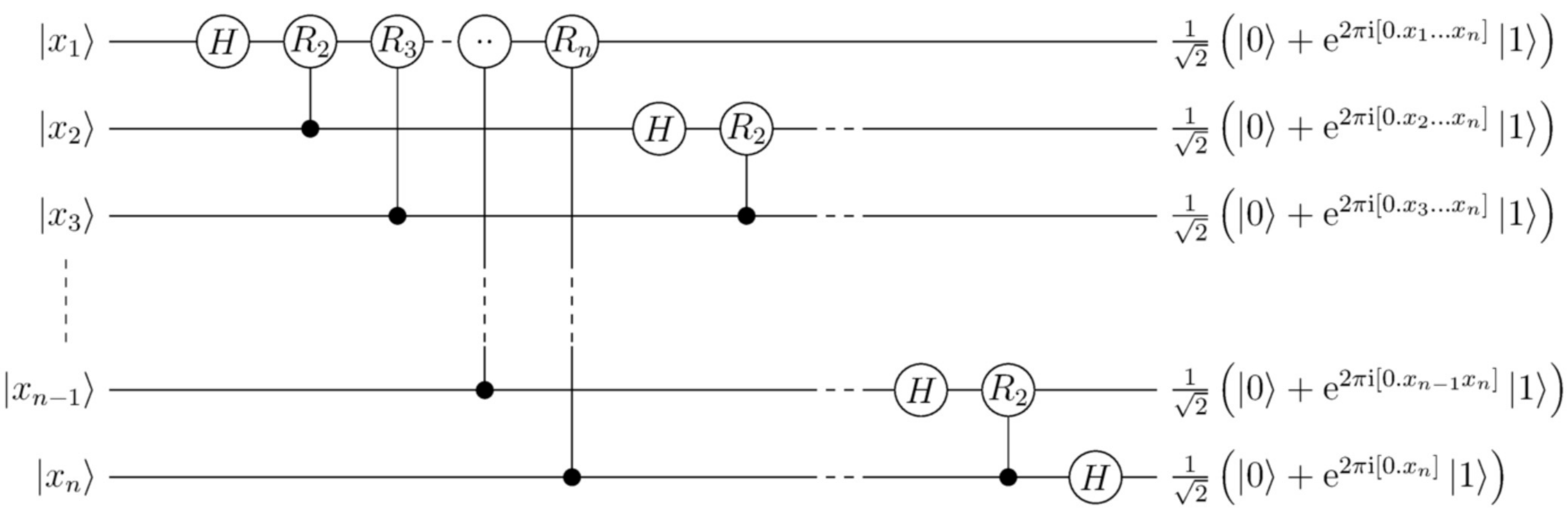
$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{nk}, \quad k = 0, 1, 2, \ldots, N-1,$$

$\omega_N = e^{\frac{2\pi i}{N}}$ and ω_N^n: N-th root of unity

$$\text{QFT} : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle$$

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

phase gate $R_k = \begin{pmatrix} 1 & 0 \\ 0 & \boxed{e^{i2\pi/2^k}} \end{pmatrix}$ 큐비트 상태가 1일 때만 위상에 영향을 줌



Shor's algorithm

- QFT (Quantum Fourier Transform)
  - 양자상태 $|x\rangle$를 새로운 양자상태 $|y\rangle$로 변환

$|x\rangle = \sum_{i=0}^{N-1} x_i|i\rangle \rightarrow |y\rangle = \sum_{i=0}^{N-1} y_i|i\rangle$:새로운 상태로 변환

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{nk}, \quad k = 0, 1, 2, \ldots, N-1,$$

$\omega_N = e^{\frac{2\pi i}{N}}$ and ω_N^n: N-th root of unity

$$\mathrm{QFT} : |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle$$

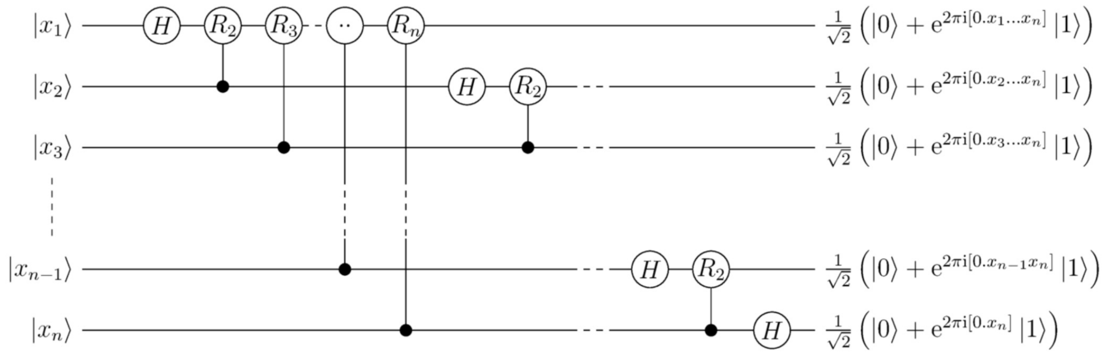
  - Basis state: $|x\rangle = |x_1, x_2 \ldots x_1\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \ldots \otimes |x_n\rangle$
  - Hadamard gate: $H|x_i\rangle = \left(\frac{1}{\sqrt{2}}\right)\left(|0\rangle + e^{2\pi i x_i 2^{-1}}|1\rangle\right)$

  - $\mathrm{QFT}(|x\rangle) = \frac{1}{\sqrt{N}} \bigotimes_{i=1}^{n} \left(|0\rangle + \omega_N^{x 2^{n-j}}|1\rangle\right)$: Quantum Fourier Transform을 텐서곱으로 표현 가능

$$\mathrm{QFT}(|x_1 x_2 \ldots x_n\rangle) = \frac{1}{\sqrt{N}} \left(|0\rangle + e^{2\pi i\,[0.x_n]}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i\,[0.x_{n-1}x_n]}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + e^{2\pi i\,[0.x_1 x_2 \ldots x_n]}|1\rangle\right)$$

*아래의 양자회로에서 해당 상태를 얻기 위해서 SWAP 게이트 사용

Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ phase gate $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$



Shor's algorithm

- QFT (Quantum Fourier Transform)
- Ex) Quantum Fourier transform on three qubit, F_8 with $n = 3, N = 8 = 2^3$

$$\mathrm{QFT}: |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle \longrightarrow \mathrm{QFT}: |x\rangle \mapsto \frac{1}{\sqrt{8}} \sum_{k=0}^{7} \omega^{xk} |k\rangle$$

- $\omega = \omega_8$: 8 root of unity → $\omega^8 = (e^{\frac{i2\pi}{8}})^8 = 1$

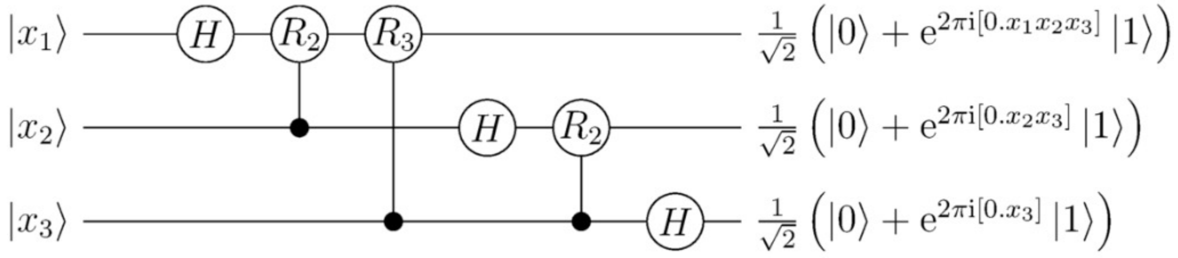
$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

$$\downarrow$$

$$F_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\ 1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega & \omega^4 & \omega^7 & \omega^2 & \omega^5 \\ 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\ 1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega & \omega^6 & \omega^3 \\ 1 & \omega^6 & \omega^4 & \omega^2 & 1 & \omega^6 & \omega^4 & \omega^2 \\ 1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega \end{bmatrix}$$

3-qubit에 대한 푸리에 변환 행렬 표현

$$\mathrm{QFT}(|x_1, x_2, x_3\rangle) = \frac{1}{\sqrt{8}} \left(|0\rangle + e^{2\pi i\,[0.x_3]}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i\,[0.x_2x_3]}|1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i\,[0.x_1x_2x_3]}|1\rangle\right)$$



Q & A