

# 응용물리연구실심화실습3 (PHY3075)

2024년 12월 1주차

응용물리학과 2022006971 이민성

## 번역

- 슈퍼덴스 코딩

**슈퍼덴스 코딩**은 텔레포테이션과 상반되는 목적을 달성하는 프로토콜입니다. 텔레포테이션이 두 개의 고전적인 비트를 전송하기 위해 한 개의 큐비트를 사용하는 반면, 슈퍼덴스 코딩은 한 개의 큐비트를 이용해 두 개의 고전적인 비트를 전송할 수 있습니다. 이 과정에서는 한 개의 e-bit(얽힌 비트)를 공유하는 대가가 필요합니다.

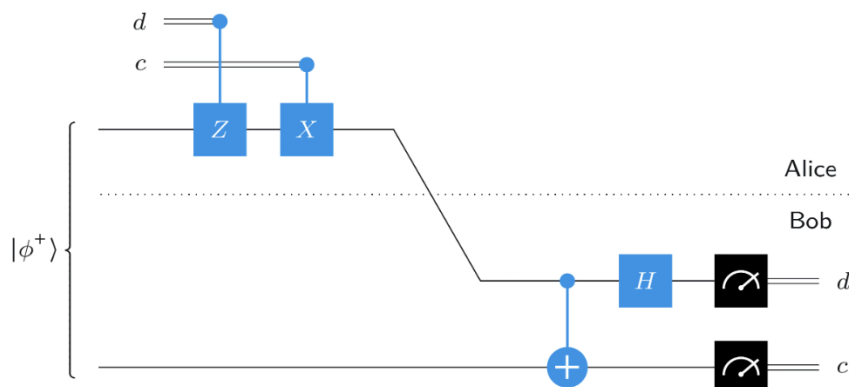
자세히 설명하자면, 슈퍼덴스 코딩에서 송신자(Alice)와 수신자(Bob)는 한 개의 e-bit의 얽힘을 공유합니다. 즉, Alice는 큐비트 A를 가지고 있고, Bob은 큐비트 B를 가지고 있으며, 이 두 큐비트는 상태  $|\phi^+\rangle$ 에 있습니다. Alice는 Bob에게 두 개의 고전적인 비트(c와 d)를 전송하려고 하고, 이 두 비트는 Alice가 하나의 큐비트를 Bob에게 보내는 방법을 통해 전송됩니다.

슈퍼덴스 코딩은 텔레포테이션이 달성하는 목표보다는 덜 흥미롭게 보일 수 있습니다. 큐비트를 전송하는 것은 향후에도 고전적인 비트를 전송하는 것보다 훨씬 어려울 가능성이 크기 때문에, 하나의 큐비트를 보내는 대신 두 개의 고전적인 비트를 전송하는 것은 그리 유용하지 않다고 생각될 수 있습니다. 그러나 슈퍼덴스 코딩이 흥미롭지 않다는 의미는 절대로 아닙니다.

슈퍼덴스 코딩이 흥미로운 이유 중 하나는 그것이 얽힘의 구체적이고도 (정보 이론의 관점에서) 눈에 띄는 사용을 보여주기 때문입니다. 양자 정보 이론에서 유명한 정리인 홀레보의 정리(Holevo's theorem)는, 공유된 얽힌 상태 없이 단일 큐비트를 전송해서는 한 개 이상의 고전적인 비트를 전송할 수 없다고 암시합니다. (홀레보의 정리는 이보다 더 일반적인 정리로, 그 정확한 진술은 기술적이지만, 이것이 바로 슈퍼덴스 코딩을 통해 얽힘을 활용함으로써 큐비트를 보내는 것만으로 고전적인 정보의 전달 용량을 두 배로 늘릴 수 있게 됨을 설명합니다.)

## 프로토콜

다음 양자 회로 다이어그램은 슈퍼덴스 코딩 프로토콜을 설명합니다:



간단히 말해서, Alice가 수행하는 작업은 다음과 같습니다:

- 만약  $d=1$ 이면, Alice는 자신의 큐비트 A에 Z게이트를 적용합니다. (만약  $d=0$ 이면, 적용하지 않습니다.)

- 만약  $c=1$ 이면, Alice는 자신의 큐비트 A에 X 게이트를 적용합니다. (만약  $c=0$ 이면, 적용하지 않습니다.)

그 후 Alice는 자신의 큐비트 A를 Bob에게 보냅니다.

Bob이 큐비트 A를 받으면, 먼저 controlled-NOT(CNOT) 게이트를 적용하여 A를 제어 큐비트로, B를 타겟 큐비트로 설정합니다. 그 후 Bob은 A에 Hadamard 게이트를 적용합니다. 그런 다음 Bob은 큐비트 B에서  $c$  값을 측정하고, 큐비트 A에서  $d$  값을 측정합니다. 두 측정 모두 표준 기저(basis) 측정 방식으로 수행됩니다.

분석

이 프로토콜의 아이디어는 매우 간단합니다: Alice는 실질적으로 Bob과 공유하고 싶은 Bell 상태를 선택하고, Alice는 Bob에게 자신의 큐비트를 보내며, Bob은 측정을 통해 Alice가 선택한 Bell 상태를 결정합니다.

즉, 그들은 처음에  $|\phi^+\rangle$  상태를 공유하며,  $c$ 와  $d$  비트에 따라 Alice는 이 상태를 그대로 두거나 I, X, Z, 또는 XZ 게이트를 자신의 큐비트 A에 적용하여 다른 Bell 상태로 전환합니다.

$$\begin{aligned}(\mathbb{I} \otimes \mathbb{I})|\phi^+\rangle &= |\phi^+\rangle \\(\mathbb{I} \otimes Z)|\phi^+\rangle &= |\phi^-\rangle \\(\mathbb{I} \otimes X)|\phi^+\rangle &= |\psi^+\rangle \\(\mathbb{I} \otimes XZ)|\phi^+\rangle &= |\psi^-\rangle\end{aligned}$$

Bob의 동작은 네 가지 Bell 상태에 대해 다음과 같은 효과를 줍니다:

$$\begin{aligned}|\phi^+\rangle &\mapsto |00\rangle \\|\phi^-\rangle &\mapsto |01\rangle \\|\psi^+\rangle &\mapsto |10\rangle \\|\psi^-\rangle &\mapsto -|11\rangle\end{aligned}$$

이는 Bob의 동작이 각 상태에 미치는 효과를 하나씩 계산하여 직접 확인할 수 있습니다.

따라서 Bob이 측정을 수행하면, 그는 Alice가 선택한 Bell 상태를 알 수 있습니다. 이 프로토콜이 올바르게 작동하는지 확인하려면 각 경우를 확인하면 됩니다:

- 만약  $cd=00$ 이면, Bob이 큐비트 A를 받을 때 (B, A)의 상태는  $|\phi^+\rangle$ 입니다. Bob은 이 상태를  $|00\rangle$ 로 변환하고,  $cd=00$ 을 얻습니다.
- 만약  $cd=01$ 이면, Bob이 큐비트 A를 받을 때 (B,A)의 상태는  $|\phi^-\rangle$ 입니다. Bob은 이 상태를  $|01\rangle$ 로 변환하고,  $cd=01$ 을 얻습니다.
- 만약  $cd=10$ 이면, Bob이 큐비트 A를 받을 때 (B,A)의 상태는  $|\psi^+\rangle$ 입니다. Bob은 이 상태를  $|10\rangle$ 로 변환하고,  $cd=10$ 을 얻습니다.
- 만약  $cd=11$ 이면, Bob이 큐비트 A를 받을 때 (B,A)의 상태는  $|\psi^-\rangle$ 입니다. Bob은 이 상태를  $-|11\rangle$ 로 변환하고,  $cd=11$ 을 얻습니다. (음의 1 위상 요소는 여기서 영향이 없습니다.)

## Qiskit 구현

```
# Required imports

from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit_aer.primitives import Sampler
from qiskit_aer import AerSimulator
from qiskit.visualization import plot_histogram
```

다음은 전송할 비트에 따라 회로를 지정하는 간단한 superdense coding 구현입니다. 먼저 전송할 비트를 지정해 보겠습니다. (비트를 변경하여 제대로 작동하는지 확인해 보세요.)

```
c = "1"
d = "0"
```

이제 이에 맞게 회로를 구축하겠습니다. 여기서는 Qiskit이 기본적인 큐비트 이름을 사용하도록 하겠습니다: 위쪽 큐비트는 `q0`, 아래쪽 큐비트는 `q1`로 지정됩니다.

```
protocol = QuantumCircuit(2)

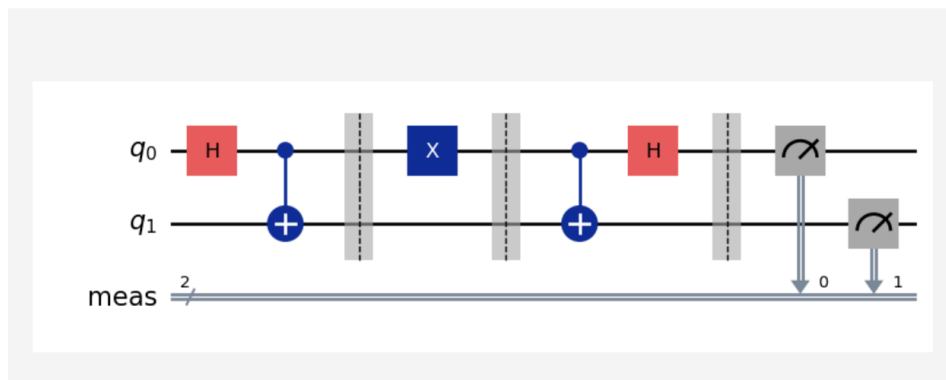
# Prepare ebit used for superdense coding
protocol.h(0)
protocol.cx(0, 1)
protocol.barrier()

# Alice's operations
if d == "1":
    protocol.z(0)
if c == "1":
    protocol.x(0)
protocol.barrier()

# Bob's actions
protocol.cx(0, 1)
protocol.h(0)
protocol.measure_all()

display(protocol.draw())
```

Output:



여기서 새로운 점은 `measure_all` 함수입니다. 이 함수는 모든 큐비트를 측정하고 결과를 하나의 클래식 레지스터에 저장합니다 (따라서 이 경우 두 비트가 됩니다).

Aer 시뮬레이터를 실행하면 예상한 출력이 나옵니다.

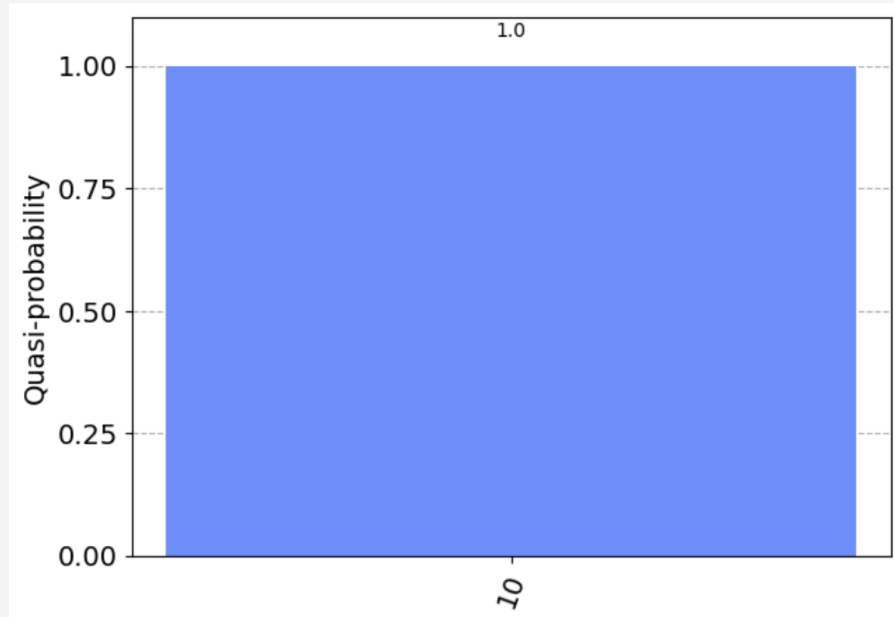
```
result = Sampler().run(protocol).result()
statistics = result.quasi_dists[0].binary_probabilities()

for outcome, frequency in statistics.items():
    print(f"Measured {outcome} with frequency {frequency}")

display(plot_histogram(statistics))
```

Output:

Measured 10 with frequency 1.0



재미로 추가적인 큐비트를 사용하여 무작위 비트 생성기로서 c와 d를 랜덤하게 선택한 후, superdense coding 프로토콜을 실행하여 이 비트들이 제대로 전송되는지 확인할 수 있습니다.

```
rbg = QuantumRegister(1, "randomizer")
ebit0 = QuantumRegister(1, "A")
ebit1 = QuantumRegister(1, "B")

Alice_c = ClassicalRegister(1, "Alice c")
Alice_d = ClassicalRegister(1, "Alice d")

test = QuantumCircuit(rbg, ebit0, ebit1, Alice_d, Alice_c)

# Initialize the ebit
test.h(ebit0)
test.cx(ebit0, ebit1)
test.barrier()

# Use the 'randomizer' qubit twice to generate Alice's bits c and d.
test.h(rbg)
test.measure(rbg, Alice_c)
test.h(rbg)
test.measure(rbg, Alice_d)
test.barrier()
```

```

# Now the protocol runs, starting with Alice's actions, which depend
# on her bits.
with test.if_test((Alice_d, 1), label="Z"):
    test.z(ebit0)
with test.if_test((Alice_c, 1), label="X"):
    test.x(ebit0)
test.barrier()

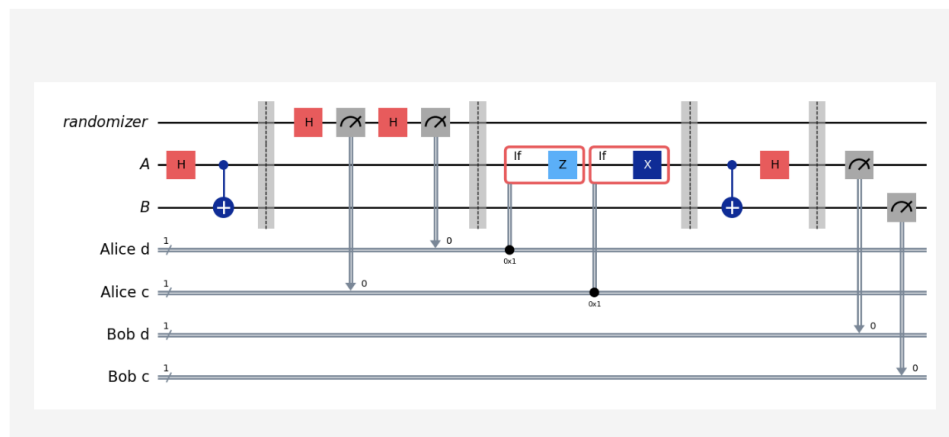
# Bob's actions
test.cx(ebit0, ebit1)
test.h(ebit0)
test.barrier()

Bob_c = ClassicalRegister(1, "Bob c")
Bob_d = ClassicalRegister(1, "Bob d")
test.add_register(Bob_d)
test.add_register(Bob_c)
test.measure(ebit0, Bob_d)
test.measure(ebit1, Bob_c)

display(test.draw())

```

Output:



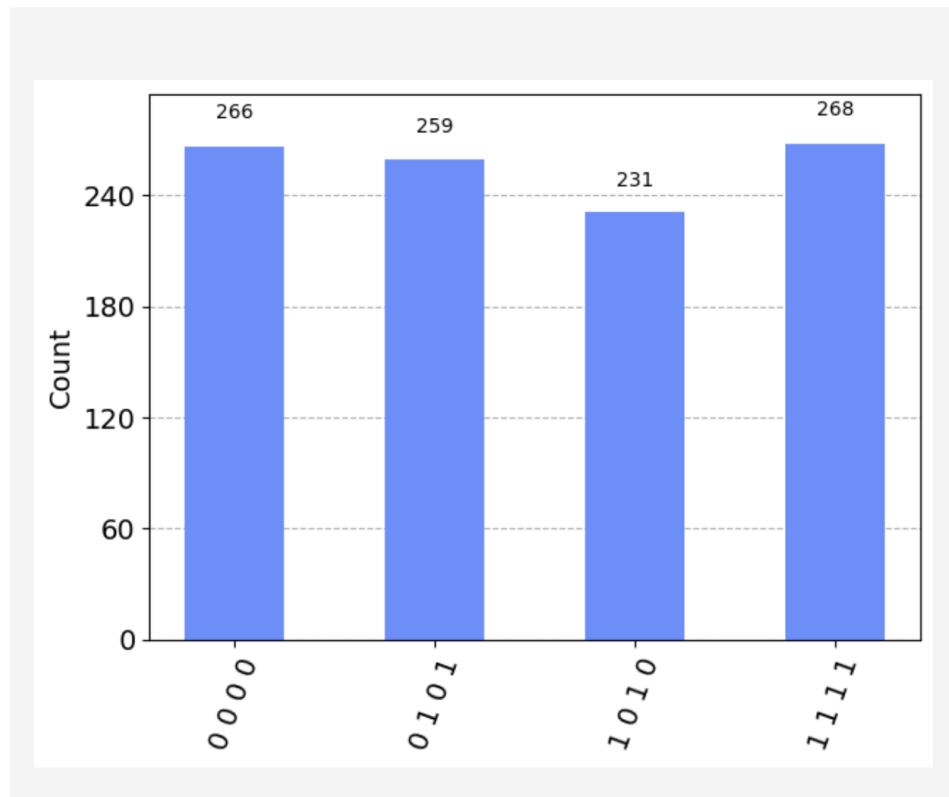
Aer 시뮬레이터를 실행한 결과, Alice와 Bob의 고전적인 비트가 항상 일치하는 것을 확인할 수 있습니다.

```

result = AerSimulator().run(test).result()
statistics = result.get_counts()
display(plot_histogram(statistics))

```

Output:



## 정리

슈퍼덴스 코딩(Superdense Coding)은 하나의 큐비트를 사용하여 두 개의 고전적인 비트를 전송하는 양자 통신 프로토콜입니다. 송신자(Alice)와 수신자(Bob)는 먼저 얽힌 큐비트(에빗)를 공유하고, Alice는 고전적인 비트(c, d)를 전송하려고 합니다.

### 프로토콜:

1. Alice는 자신의 큐비트에 X, Z 게이트를 적용해 두 개의 비트(c, d)에 따라 상태를 변환합니다.
2. Alice는 큐비트를 Bob에게 전송합니다.
3. Bob은 큐비트를 받아 CNOT, Hadamard 게이트를 적용하고, 두 큐비트에서 각각 c와 d 값을 측정합니다.

### 실제 작동 원리:

- Alice는 c와 d 값에 따라 Bell 상태를 변경하고, Bob은 이를 측정하여 두 고전적인 비트(c, d)를 복원합니다.

### Qiskit 구현 예시:

1. 큐비트 두 개를 준비하고, 얽힘 상태를 만들기 위해 Hadamard와 CNOT 게이트를 적용합니다.
2. Alice의 비트에 맞게 X, Z 게이트를 적용합니다.
3. Bob은 큐비트를 받아 CNOT과 Hadamard 게이트로 상태를 측정하고, 결과를 고전적인 비트로 변환합니다.

결과적으로 Alice는 하나의 큐비트를 Bob에게 전송하여 두 개의 고전적인 비트를 성공적으로 전달할 수 있습니다.

