

Qiskit 개발자 자격 시험

Inho Choi

Qiskit Advocate

- Lecture 1: 게이트와 양자 회로
- Lecture 2: 양자 회로의 측정과 OpenQasm
- Lecture 3: 양자 백엔드에 양자회로 실행하기
- Lecture 4: 양자 회로 및 회로의 정보와 실행결과를 해석하기
- Lecture 5: 유용한 기능들

Lecture 2: 양자 회로의 측정과 OpenQasm

1. 측정과 비단일 연산자
2. 레지스터
3. OpenQasm

측정과 비단일 연산자

Non-unitary operator

- 측정
- 큐비트의 초기화
- 고전적 조건부 연산자



```
QuantumCircuit.measure(qubit,cbit)
```



Classical bit



```
QuantumCircuit.measure_all
```

measure_all과 measure의 차이

- **measure**: 측정을 위한 큐비트와 고전비트 지정이 필요
- **measure_all**: 지정된 측정 고정 비트가 없다면 새로운 고전 비트를 생성함
또한 측정 전에 모든 큐비트에 **배리어**를 생성함

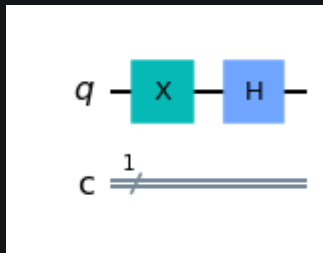
큐비트의 초기화

- 지정한 큐비트의 상태를 $|0\rangle$ 으로 만들어줍니다

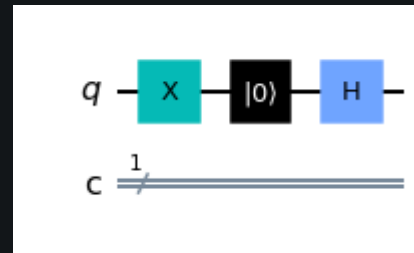
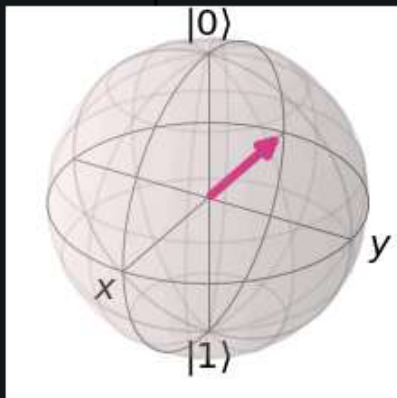


```
QuantumCircuit.reset(qubit)
```

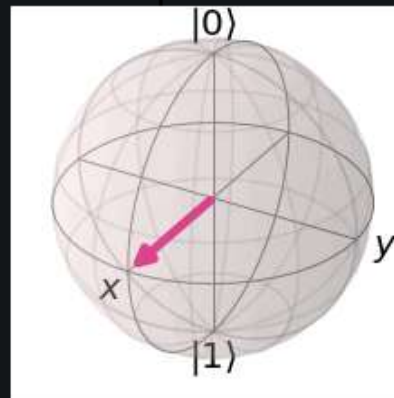

큐비트의 초기화



qubit 0



qubit 0





```
QuantumCircuit.Gate(qubit).c_if(classical bit, value)
```

- 양자 회로에서 지정한 고전적 비트가 `value`의 상태일때 게이트를 해당 큐비트에 실행합니다.

레지스터



```
QuantumRegister(number of qubits, name=optional)
```

- 만약 name을 지정해주지 않는다면 기본 이름이 주어집니다.



```
ClassicalRegister(number of classical bits, name=optional)
```

- 만약 name을 지정해주지 않는다면 기본 이름이 주어집니다.

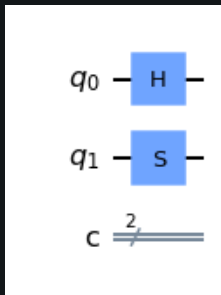
양자회로 만들기

```
QuantumCircuit(registers, name=optional)
```

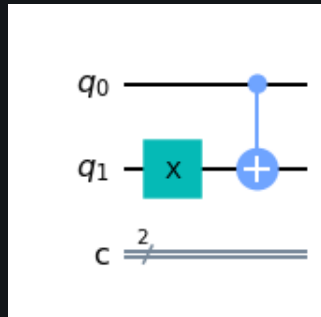
QuantumRegister and ClassicalRegister

compose

여러가지의 양자 회로를 합쳐 하나의 회로로 만들어 줄 수 있습니다.



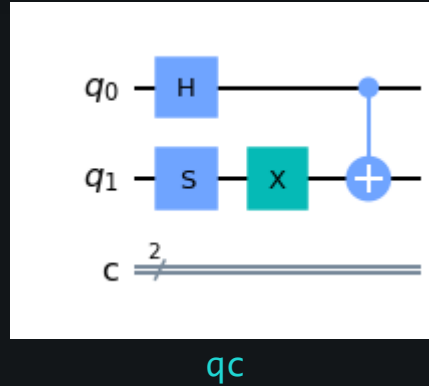
`qc1`



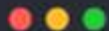
`qc2`

Using "+"

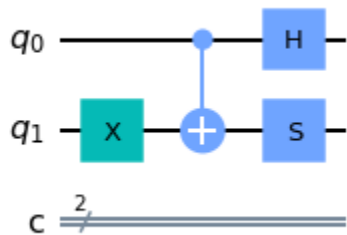
```
qc=qc1 + qc2
```



Using “compose”



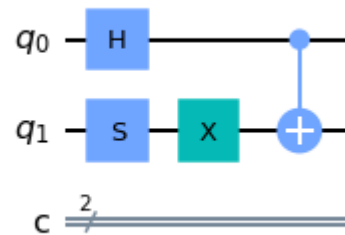
```
qc3=qc2.compose(qc1)
```



qc3



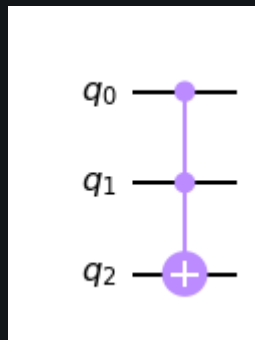
```
qc4=qc1.compose(qc2)
```



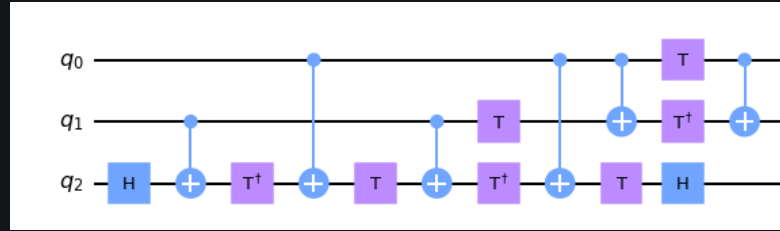
qc4

decompose

양자 회로를 한단계 아래로 분해시켜줍니다.

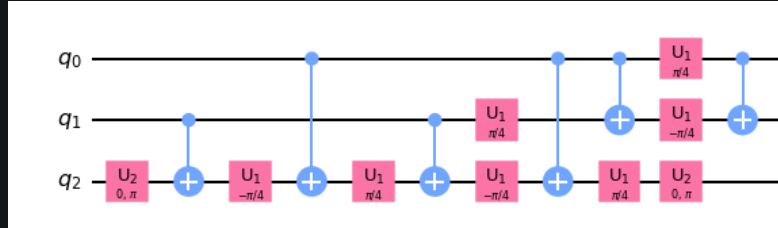


decompose



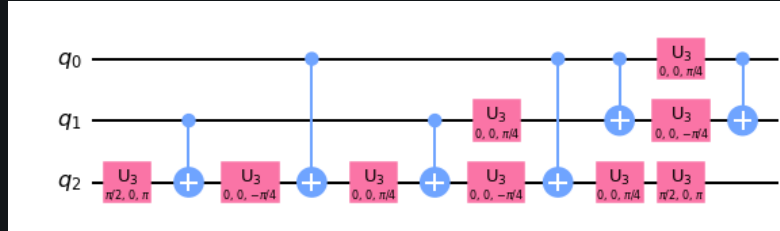
```
qc.decompose()
```

decompose



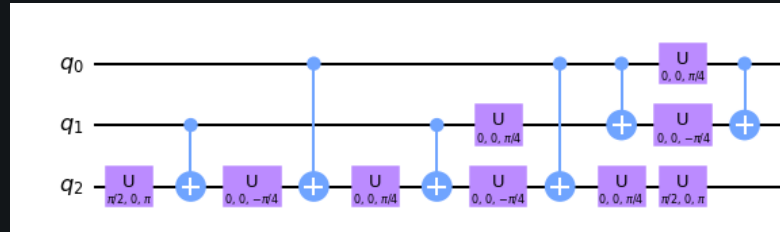
```
qc.decompose().decompose()
```

decompose



```
qc.decompose().decompose().decompose()
```

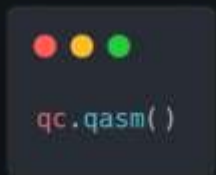
decompose



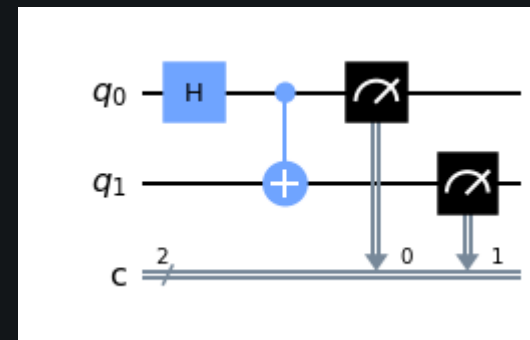
```
qc.decompose().decompose().decompose().decompose()
```

OpenQasm

- Open Quantum Assembly Language
- 양자 하드웨어 친화적인 언어
- Qiskit의 양자 회로를 하드웨어 친화적인 OpenQasm으로 바꿀 수 있음.



```
'OPENQASM 2.0;\ninclude "qelib1.inc";\nqreg q[2];\ncreg c[2];\nh q[0];\ncx q[0],q[1];\nmeasure q[0] -> c[0];\nmeasure q[1] -> c[1];\n'
```

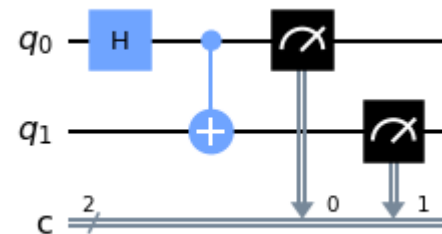


qc



```
qc.qasm(formatted=True)
```

```
OPENQASM 2.0;  
include "qelib1.inc";  
qreg q[2];  
creg c[2];  
h q[0];  
cx q[0],q[1];  
measure q[0] -> c[0];  
measure q[1] -> c[1];
```



qc

Save as "qasm"



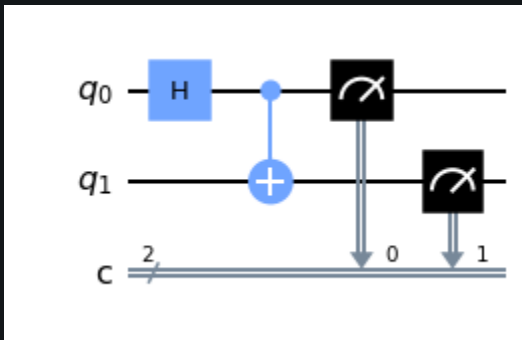
```
qc_qasm = qc.qasm(filename='sample.qasm')
```

양자회로를 qasm파일로 저장



```
qc2 = QuantumCircuit.from_qasm_file("./sample.qasm")
```

Qasm 파일을 불러와 qc2로 지정하기



qc, qc2

