

# IBM Q를 이용한 양자 컴퓨팅 개념의 구현 및 분석

윤진호, 문봉교  
동국대학교 컴퓨터공학과

## Implementation and Analysis of Quantum Computing Concepts on IBM Q

Jinho Yoon, Bong-kyo Moon

Dept. of Computer Science and Engineering, Dongguk University

e-mail: [prlstk@naver.com](mailto:prlstk@naver.com), [bkmoon@dgu.edu](mailto:bkmoon@dgu.edu)

### 요약

본 논문에서는 IBM사의 Q를 이용하여 몇 가지 양자 컴퓨팅 개념을 구현해보고 검증한다. Superdense coding과 Quantum teleportation, Bell's Inequality를 python 기반의 코드로 구현하고 실제 ibmqx4 양자 컴퓨터로 실행한 결과, Superdense coding은 약 85%의 정확도, Quantum teleportation은 96.7%의 정확도를 보이고 Bell's Inequality가 성립하지 않는 것을 확인하였다.

### I. 서론

1990년대 이후로 고전적 컴퓨터에서 사용되던 알고리즘에 비해서 엄청난 효율을 보여줄 것으로 기대될 양자 컴퓨팅의 개념이 제안되면서 양자 역학의 원리를 따르는 양자 컴퓨터가 많은 관심을 받기 시작하였다.

이에 IBM에서는 Q라는 이름으로 양자 컴퓨터를 공개하고, 클라우드를 통하여 전 세계에서 사용할 수 있도록 공개하였다. 그러나 양자 컴퓨터에서 실제로 양자 컴퓨팅이 가능한지는 아직 검증이 부족한 상황이다. 본 논문에서는 IBM Q를 이용하여 세 가지의 양자 컴퓨팅 개념을 python 기반으로 구현하고, 그 결과를 분석한다.

### II. IBM Q

IBM Q는 2016년에 IBM에서 공개한 양자 컴퓨터로, 클라우드 환경에서 사용자에게 양자 컴퓨팅 환경을 제공하고 있다. IBM에서 공개하고 있는 양자 컴퓨터에는 ibmqx2 (5큐비트), ibmqx4(5큐비트), ibmqx5(16큐비트)가 있다. IBM에서는 GUI 기반의 양자 컴퓨팅 도구인 Composer와 SDK인 QISKit를 도구로 제공하여 양자 컴퓨터 프로그램을 작성하여 시뮬레이터 상에서 구동해보거나 실제의 양자 컴퓨터에서 실행할 수 있도록 해준다.

본 논문에서는 ibmqx4 양자 컴퓨터를 이용한다. ibmqx4 컴퓨터는 fixed-frequency superconducting transmon qubit에 바탕을 두고, 다섯 개의 큐비트에 양자 정보를 저장한다.

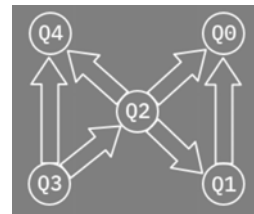


그림 1. ibmqx4의 양자 프로세서 다이어그램

큐비트는 0과 1의 상태를 동시에 갖는 단위로서,  $|\alpha\rangle$ 로 표시한다. 프로그램을 작성하기 위해서 다음과 같은 게이트가 사용되는데, 하나의 큐비트에 적용되는 게이트로 Hadamard gate는  $\{[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}], [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]\}$  연산을 수행하고, X gate는  $\{[0,1], [1,0]\}$  연산을 수행하고, Z gate는  $\{[1,0], [0,-1]\}$  연산을 수행한다. 두 개의 큐비트에 적용되는 게이트로 CNOT gate가 있는데, 이 게이트는  $\{[1,0,0,0], [0,1,0,0], [0,0,0,1], [0,0,1,0]\}$ 의 행렬 연산을 수행한다.

ibmqx4에는 오류가 존재하는데, Gate error는 단일 큐비트에 적용하는 연산 수행 시에 발생하는 오류이고, MultiQubit Gate error의 경우에는 여러 개의 큐비트에 적용하는 연산 수행 시에 발생하는 오류이다.

또, Readout error의 경우는 측정 시에 발생하는 오류이다. 또한 IBM Q에서는 단일 큐비트의 Z 기저 측정만 가능하다는 제약 사항이 있다.

	Q0	Q1	Q2	Q3	Q4	
Gate error( $10^{-3}$ )	0.60	1.03	0.94	1.72	1.63	
Readout error( $10^{-2}$ )	6.50	6.60	3.00	4.80	5.80	
MultiQubit Gate error ( $10^{-2}$ )	CX1_0	CX2_0	CX2_1	CX2_4	CX3_2	CX3_4
	2.51	2.21	2.87	4.75	5.65	4.32

표 1. ibmqx4의 오류율

IBM Q에서 동작하는 양자 컴퓨팅 프로그램을 작성하는 데 사용한 방법은 다음과 같다.

1. IBM Q 계정 등록
2. python 환경에서 QISKit SDK 설치
3. QISKit과 ibmqx4 연결
4. python 기반의 OpenQASM 코드 작성
5. 작성한 프로그램 실행 및 결과 출력

### III. 컴퓨팅 개념 구현

#### A. Superdense coding

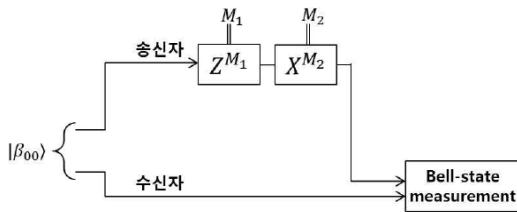


그림 2. Superdense coding의 구성도

Superdense coding은 1992년에 Bennett와 Weisner가 처음 제안한 양자 컴퓨팅 개념으로, 하나의 큐비트를 이용하여 두 개의 고전 비트를 전송하는 프로토콜이다. ibmqx4에서는 벨 상태로 부호화시키는 과정에서 Hadamard gate와 CNOT gate가 사용된다.

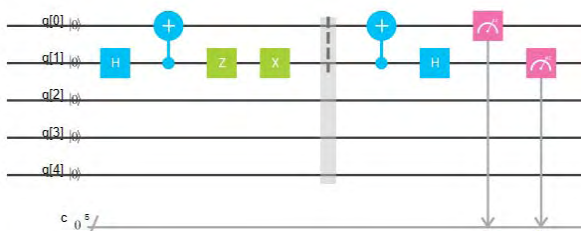


그림 3. IBM Q에서의 Superdense coding 예시

이론적인 Superdense coding에서는 고전 비트 00을 전송할 때에는 00, 01을 전송할 때에는 01, 10을 전송할 때에는 10, 11을 전송할 때에는 11의 결과가 출력되어야 한다. 그림 3에서는 송신자가 11의 고전 정보를 전송하였다.

Hadamard q[1]
CNOT q[1], q[0]
(optional) Z q[1], X q[1]
measure q[0], q[1]

표 2. Superdense coding pseudo code

작성한 프로그램은 Hadamard와 CNOT을 이용하여 벨 상태를 만들고, Z와 X를 통하여 보낼 큐비트의 값을 결정한다. 최종적으로 보내진 두 큐비트를 측정하게 되어 있다.

#### B. Quantum teleportation

Charles Bennett에 의해 만들어진 용어인 Quantum teleportation은 양자 정보가 어느 한 위치에서 다른 위치로 전송되는 것을 말한다. 다음 그림은 Quantum teleportation을 구현하기 위한 구성도이다.

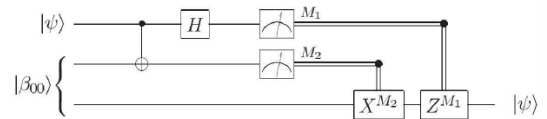


그림 4. Quantum teleportation 구성도

그림 4에서 벨 상태를 형성해주기 위하여 CNOT gate와 Hadamard gate가 사용되었다. 또, Hadamard gate를 통해 전송될 양자 상태가 설정된다. IBM Q에서는 현재 프로그램 제어문의 사용이 불가능하므로 수신자의 큐비트에 적용될 연산자의 선택이 불가능하다. 따라서, 본 논문에서는 연산 후에 측정하는 기법인 deferred measurement를 이용한다.

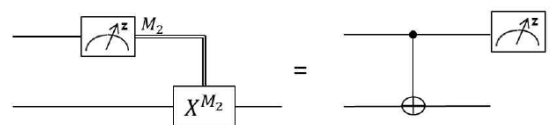


그림 5. Deferred measurement

그림 5의 deferred measurement를 통하여 고전 채널 없는 정보 전송이 가능해지고, 결과에 따라 X 연

산 과정과 Z 연산 과정을 대체할 수 있다.

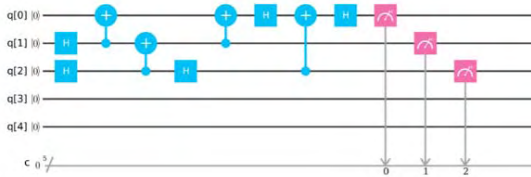


그림 6. IBM Q에서의 Quantum teleportation

```

Hadamard q[1], q[2]
CNOT q[1], q[0]
CNOT q[2], q[1]
Hadamard q[2]
CNOT q[1], q[0]
Hadamard q[0]
CNOT q[2], q[0]
Hadamard q[0]
measure q[0], q[1], q[2]
    
```

표 3. Quantum teleportation pseudo code

작성한 프로그램은 Hadamard와 두 번의 CNOT을 통해 벨 상태를 만들고, Hadamard를 통하여 큐비트를 전송한다. 만약 X연산이 필요하다면 CNOT을 사용하고, Z연산이 필요하다면 Hadamard-CNOT-Hadamard를 거치고 각 큐비트를 측정한다.

### C. Bell's inequality

Bell이 일반화한 이 부등식은 지금까지 양자 역학에서 진행한 실험 전부가 아래의 부등식을 만족하지 않았다.

$$1 + C(\vec{b}, \vec{c}) \geq |C(\vec{a}, \vec{b}) - C(\vec{a}, \vec{c})|$$

위의 부등식의 C값은 두 벡터 사이의 각의 코사인 값을 의미한다. 이 부등식은  $\vec{a}, \vec{b}, \vec{c}$ 가 다음과 같을 때 만족하지 않는 최댓값이 나온다.

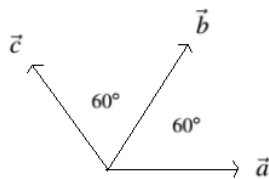


그림 7. 만족하지 않는 최댓값의  $\vec{a}, \vec{b}, \vec{c}$ 의 위치

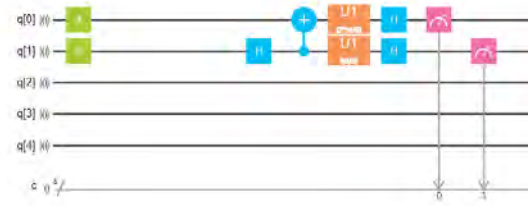


그림 8. IBM Q에서의  $C(\vec{b}, \vec{c})$  측정

```

X q[0], q[1]
Hadamard q[1]
CNOT q[1], q[0]
if q[x] is  $\vec{a}$ , do nothing
else if q[x] is  $\vec{b}$ , multiply  $\{[1,0], [0, e^{i*\pi/3}]\}$ 
else if q[x] is  $\vec{c}$ , multiply  $\{[1,0], [0, e^{i*2\pi/3}]\}$ 
Hadamard q[1], q[0]
measure q[1], q[0]
    
```

표 4. Bell's inequality pseudo code

작성한 프로그램은 우선 두 큐비트의 값을  $|1\rangle$  상태로 만들고 Hadamard와 CNOT을 이용하여 벨 상태를 만든다. 그 다음 큐비트가 어떤 벡터에 대응하는지에 따라서 회전 행렬을 곱한다. Hadamard를 통해서 각각의 큐비트의 값을 교환하고 결과를 측정한다.

## IV. 구현 결과 분석

### A. Superdense coding

송신 \ 수신	00	01	10	11
00	0.918	0.036	0.035	0.011
01	0.105	0.851	0.012	0.032
10	0.121	0.015	0.842	0.022
11	0.020	0.105	0.099	0.776

표 2. Superdense coding 컴퓨팅 결과

각각의 상태에 대해서 1024번을 수행한 결과, 약 85%의 확률로 제대로 된 전송이 이루어졌다. 00에서 11로 갈수록 올바르게 전송되지 않을 확률이 높아짐을 확인할 수 있는데, 이는 00에서 01, 10 또는 11 상태로 만들기 위한 과정에서 Gate error가 일어나 연산이 제대로 수행되지 않았거나, 결과를 출력하는

과정에서 Readout error가 발생하였기 때문이다.

**B. Quantum teleportation**

결과	확률	결과	확률
000	0.125	100	0.195
010	0.133	110	0.097
001	0.121	101	0.149
011	0.104	111	0.075

표 3. Quantum teleportation 프로그램 구현 결과

이론적인 Quantum teleportation에서 수신자의 단일 큐비트를 측정할 때 0.5의 확률로 0, 0.5의 확률로 1이 측정된다. ibmqx4에서 1024번의 테스트를 수행하여, 수신자가 0을 측정하거나 1을 측정할 확률은 다음과 같다.

$$P_{Bob=0} = P_{000} + P_{010} + P_{001} + P_{011} = 0.483$$

$$P_{Bob=1} = P_{100} + P_{110} + P_{101} + P_{111} = 0.516$$

이론적인 결과와 비교할 때, 약 3.3%의 오차가 발생하였다. 이러한 오차가 발생한 원인은 Hadamard Gate와 측정 과정에서 Gate error와 Readout error가 발생하였기 때문이다.

**C. Bell's inequality**

Test	00	01	10	11
$C(\vec{a}, \vec{b})$	0.180	0.322	0.391	0.107
$C(\vec{a}, \vec{c})$	0.354	0.165	0.171	0.310
$C(\vec{b}, \vec{c})$	0.173	0.355	0.347	0.125

표 4. 벨 부등식의 확률 측정 결과

8192번 수행한 각각의 측정 결과는 다음과 같다. 최종적으로 얻어진  $C(\vec{a}, \vec{b})$ ,  $C(\vec{a}, \vec{c})$ , 그리고  $C(\vec{b}, \vec{c})$ 의 값은 다음과 같다.

$$C(\vec{a}, \vec{b}) = -0.426,$$

$$C(\vec{a}, \vec{c}) = 0.328,$$

$$C(\vec{b}, \vec{c}) = -0.404,$$

$C(\vec{a}, \vec{b})$ 는 00 또는 11이 나올 확률에서 01 또는 10이 나올 확률을 뺀 값이다.  $C(\vec{a}, \vec{c})$ 와  $C(\vec{b}, \vec{c})$ 에 같은 방법을 적용하여 최종적으로 부등식에 대입해보면 다음과 같다.

$$1 + C(\vec{b}, \vec{c}) \geq |C(\vec{a}, \vec{b}) - C(\vec{a}, \vec{c})| =$$

대입한 값이  $0.596 \geq 0.754$ 이므로 이 식은 모순이다. 따라서, ibmqx4에서 벨의 부등식이 성립하지 않는 것을 확인할 수 있다.

**IV. 결론**

1990년대에 처음 제시되어, 먼 미래의 이야기로 여겨져 왔던 양자 컴퓨팅 프로그램이 실제로 작동하는 수준까지 올라왔다. 본 논문에서는 세 가지의 양자 컴퓨팅 개념을 python 기반의 코드로 구현하고, 구현된 프로그램을 ibmqx4로 실행한 결과, Quantum teleportation과 Bell's inequality에서는 좋은 결과를 보여주었으나, Superdense coding에서는 이론적인 값과 비교하였을 때 많은 차이를 보여주었다. 이것으로 보아 아직 양자 컴퓨터가 실용적으로 쓰이기까지는 시간이 더 필요하다는 것을 알 수 있지만, 기술의 빠른 발전으로 인하여 향후 몇십 년 내에 범용적인 연산을 수행할 수 있는 양자 컴퓨터를 볼 수 있을 것이다.

**V. 참고 문헌**

[1] 하진영, 안병규, 이종현, 신정환, 허준, IBM Q 양자 컴퓨터를 이용한 양자 정보 전송 알고리즘 구현, 한국통신학회 동계종합학술발표회, pp. 25-26, Jan 2018

[2] Diego García-Martín and German Sierra, "Five Experimental Tests on the 5-qubit IBM Quantum Computer", arXiv:1712.05642, pp 1-8, Dec 2017

[3] J. S. Bell, "On the Einstein Podolsky Rosen paradox", Physics Physique Fizika 1, pp 195-200, Nov 1964

[4] Feynman, Richard, "Simulating Physics with Computers", International Journal of Theoretical Physics. vol. 21, pp 467-488, June 1982

[5] Bennett. C, Wiesner. S, "Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states", Physical Review Letters. vol. 69, pp 2881-2884, Nov 1992

[6] IBM Quantum Experience, <http://www.research.ibm.com/quantum>