

QC-LDPC Code 설계 및 거리에 따른 물리 Qubit 수를 Surface Code 와 비교

김홍준, 강유진, 국성연, 허준*

고려대학교*

hongjunkim98@korea.ac.kr, yujin20@korea.ac.kr, viviangood@korea.ac.kr, *junheo@korea.ac.kr

Designing QC-LDPC Code and Comparing Physical Qubit Number Used by Distance with Surface Code

Hongjun Kim, Yujin Kang, Sungyeon Kook, Jun Heo*
Korea Univ., Korea Univ.

요약

본 논문은 distance-3 논리 Qubit 를 Quasi-Cyclic(QC) LDPC Code 로 설계하며, IBM Q로 구현한다. 또, 높은 거리의 논리 Qubit 를 설계하는 H_x , H_z 를 도출하여, Surface Code 보다 향상된 거리 대비 물리 Qubit 수를 확인한다.

I. 서 론

본 논문은 양자 오류 정정(Quantum Error Correction)의 목표인 높은 성능을 가진 논리 Qubit를 개발하기 위한 연구를 진행한다. 인코딩 방법 중 Surface Code는 강한 보호성과 안정적인 디코딩 능력으로 주목받지만, 낮은 인코딩 효율과 확장성의 한계를 가진다. 본 논문은 대안으로 q-LDPC Code의 일종인 QC-LDPC Code를 활용한 인코딩 기법을 제안하며, IBM Q 시뮬레이터로 Code를 동작 시킨다. 나아가, 거리의 증가에 따른 물리 Qubit 수 증가율을 Surface Code와 비교 및 분석한다.

II. 본 론

A. Distance-3 논리 Qubit 회로 설계 및 실험

다음 4 단계를 통해 논리 Qubit 를 설계할 수 있다:
(1) p 크기의 순환 행렬을 Shift Table 활용하여
 $H_X H_Z^T = 0$ 만족하는 H_X, H_Z 행렬 구하기, **(2)** H_X, H_Z 행렬
 을 바탕으로 특정 X,Z-Ancilla 와 CNOT 연산한 Data
 Qubit 세트 구하기, **(3)** 스케줄링을 통한 Ancilla 연산
 순서 결정하기, **(4)** 특정 Data Qubit 에 오류 삽입 후,
 Ancilla 측정값 변화를 통해 디코더 사용하여 오류 감
 지 및 정정하기.

(1) H_X, H_Z 행렬 구하기

H_X, H_Z 가 $H_X H_Z^T = 0$ 을 만족하기 위해서는 이론적으로는 CSS 조건인 (Eq 1.3)을 만족해야 한다. [1]

$$H_X = [A, B] \quad (\text{Eq 1.1})$$

$$H_Z = [C, D] \quad (\text{Eq 1.2})$$

$$H_X H_Z^T = AC^T + BD^T = 0 \pmod{2} \quad (\text{Eq 1.3})$$

(A, B, C, D 는 Quasi-Cyclic 행렬)

이때, A, B, C, D 내부의 각 블록에 $m \times n$ 크기의 Shift Table에 의해 결정된 $p \times p$ 크기의 순환행렬이 위치한다. 그러나, 1개의 논리 Qubit를 만들기 위해서는

(Eq 2)가 만족되어야 한다.[2]

$$k = n - \text{rank}(H_X) - \text{rank}(H_Z) \quad (\text{Eq 2})$$

(Eq 2)를 만족하면서 블록의 순환성을 유지할 수 없기 때문에, 순환성이 깨지되 CSS 조건을 만족하도록 H_X , H_z 를 (Eq 3) 같이 구하였다.

이때, $p = 7, m = 2, n = 2$ 로 설정했다.

[Eq 3.1) H_X 행렬, Eq 3.2) H_Z 행렬]

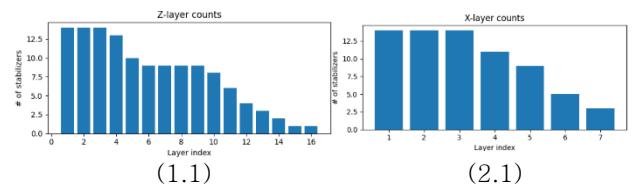
$n = 28$ 이며, 해당 행렬은 $\text{rank}(H_X) = 14$, $\text{rank}(H_Z) = 13$ 으로 $k = 1$ 이 만족함을 알 수 있다.

(2) X,Z-Ancilla 와 연산할 Data Qubit 구하기

H_X, H_Z 의 정보를 토대로 각각의 X,Z-Ancilla가 어떤 Data Qubit들과 CNOT 연산을 수행하는지 구한다. 이를 테면, (Eq 3.1)의 1행의 정보를 토대로, X-Ancilla 한 개는 Data Qubit q0, q21, q24, q27와 CNOT 연산을 수행함을 알 수 있다. 나아가, H_X, H_Z 정보를 토대로 Data Qubit은 총 28개, X-Ancilla는 14개, Z-Ancilla는 14개가 사용됨을 확인할 수 있다.

(3) 스케줄링을 통한 Ancilla 연산 순서 결정

한 time 에 여러 안정자를 측정하지만, 그 때 같은 Data Qubit 가 동시에 측정되는 걸 방지해야 한다. 따라서 스케줄링을 통해 연산 순서를 결정하고, Hopcroft-Karp 알고리즘을 적용하여 순서를 결정한다.



[Figure(1.1)Z-Layer Counts, (1.2)X-Layer Counts]

시뮬레이션 결과, [Figure 1]과 같이 Z-Ancilla 는 16 번, X-Ancilla 는 7 번의 time 에 걸쳐 측정되었다.

(4) 오류 삽입 후, 디코더 활용하여 오류 정정

우선, 오류 없이 안정화 작업 수행한 다음 모든 Ancilla 값을 측정한다. 그 후, 한 개의 Data Qubit 에 오류를 추가한 뒤 안정화 작업을 수행하여 Ancilla 값을 다시 측정한다. 이 때, Ancilla 측정 값이 달라져 있을 것이며, 두 값을 비교하여 디코더를 활용하여 오류를 탐지 및 정정한다.

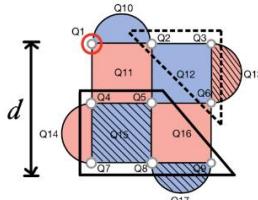
예시로 Data Qubit q3에 X 오류를 삽입하였으며, 이 때 7, 10, 11 번 째 Z-Ancilla 측정 값에 변화를 감지하였다. 추후에는, 오류 정보 없이 디코더를 활용하여 Ancilla 값을 토대로 오류의 위치를 감지할 수 있다.

B. 거리 증가에 따른 필요한 물리 Qubit 수 비교

(Surface Code VS QC-LDPC Code)

(B-1) Surface Code

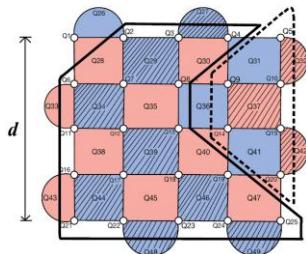
우선, Surface Code 중 거리 3의 Corner Rotated(CR) 구조는 아래와 같은 모양을 가진다.



[Figure 2] CR distance-3($d=3$) 논리 Qubit[3]

[Figure 2]의 논리 Qubit은 Data Qubit 9개, Ancilla 8개를 사용하여 총 17개의 물리 Qubit으로 구성된다.

나아가, 거리 5의 CR Surface Code는 [Figure 3]의 구조를 가진다.



[Figure 3] CR distance-5($d=5$) 논리 Qubit[3]

[Figure 3]의 논리 Qubit은 Data Qubit 25개, Ancilla 24개를 사용하여 총 49개의 물리 Qubit으로 구성된다.

(B-2) QC-LDPC Code

QC-LDPC Code는 Surface Code와 달리 특정 구조가 있는 게 아니며, 따라서 각 거리에 따라 사용될 수 있는 물리 Qubit 수의 편차가 있다. 우선, 거리 3인 QC-LDPC Code는 [A] 실험에서 보였듯 Data Qubit 28개, Ancilla 14개(X,Z-Ancilla 중복 사용)를 사용하여 총 42개의 물리 Qubit를 사용한 걸 확인하였다. 나아가, 거리가 5일 때 Data Qubit 42개, Ancilla 21개

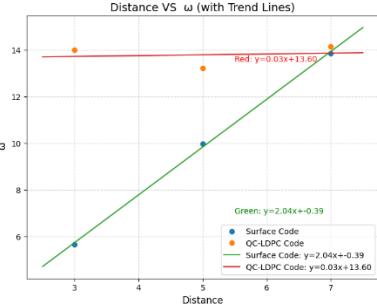
로 총 63개의 물리 Qubit을 사용하여 인코딩에 성공하였으며($p = 7, m = 3, n = 3$), Data Qubit 66개, Ancilla 33개로 총 99개의 물리 Qubit을 사용해 거리 6의 Code를 설계할 수 있었다($p = 9, m = 3, n = 3$).

III 결 론

(Eq 4)와 같이 ω 를 정의한다면,

$$\omega = \frac{\text{사용된 물리 Qubit 수}}{\text{distance}} \quad (\text{Eq 4})$$

거리에 따른 ω 값을 [Figure 4]와 같이 표현 가능하다..



[Figure 4] Distance VS ω

[Figure 4]를 통해 거리의 증가에 따라 QC-LDPC와는 달리 Surface Code에서는 ω 값 크게 증가함을 확인할 수 있다. 이는, QC-LDPC는 거리 증가에 따라 사용되는 물리 Qubit 수가 선형적으로 증가하고, Surface Code는 더 가파른 폭으로 증가함을 의미한다.

즉, QC-LDPC Code가 Surface Code의 거리 증가에 따른 인코딩 효율 저하라는 한계점을 보완하는 대안임을 확인할 수 있고, 추후에는 QC-LDPC에 다양한 물리 오류 파라미터를 삽입하여 Surface Code와 비교하였을 때 Logical Error Rate의 변화가 상대적으로 어떠한 지로 연구를 확장할 수 있다.

ACKNOWLEDGEMENT

이 논문은 2025년도 정보(과학기술정보통신부)의 재원으로 한국연구재단 양자기술연구개발선도(양자컴퓨팅)사업(RS-2024-00431853, 50%), 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. RS-2023-00225385, NISQ 환경에서 저부하, 고효율 양자 오류 경감 기술 개발 및 응용) (50%)의 지원을 받아 수행된 연구임

참 고 문 현

- [1] Mondal, Arijit, and Keshab K. Parhi. "Quantum circuits for stabilizer error correcting codes: A tutorial." *IEEE Circuits and Systems Magazine* 24.1 (2024): 33–51.
- [2] D. Gottesman, "Stabilizer Codes and Quantum Error Correction," Phd Thesis, Caltech (1997)
- [3] Kim, Hong Jun., "A Study on Designing 5x5 Surface Code and Analysis on Logical Error Rate based on Asymmetric Error Environment" *한국통신학회 학제학술대회* (2025)