

연접 부호를 적용한 단일 논리 큐빗의 공간 비용 감소 기법

강유진, 정유신, 국성연, 허준*

고려대학교

yujin20@korea.ac.kr, jeongys604@korea.ac.kr, viviangood@korea.ac.kr, *junheo@korea.ac.kr

Reducing the space cost of single logical qubit using a concatenated code

Kang Yu Jin, Chung Youshin, Kook Sungyeon, Heo Jun*
Korea Univ.

요약

본 논문은 표면 부호와 [[5,1,3]] 부호를 연접하여 사용하여 표면 부호의 낮은 부효율을 보완하였다. 부호 거리를 3 배 증가시키기 위해 표면 부호만을 사용한 경우 큐빗 개수가 9 배 증가하지만, 제안 방식은 [[5,1,3]] 부호 인코딩 시간에 따른 시간 비용이 증가하였지만 5 배의 논리 큐빗과 논리 연산을 위한 Ancilla 큐빗 1 개를 사용하므로 22.2% 더 적은 큐빗 만으로 구현하였다.

I. 서론

초전도 큐빗, 이온 트랩 큐빗 등을 사용한 양자 프로세서에서는 논리 큐빗을 만들기 위해 표면 부호(Surface code)를 사용하는데, 이는 표면 부호의 임계값(Threshold)이 매우 높을 뿐만 아니라 인접한 4 개의 큐빗 간의 연산만으로 신드롬 측정 및 논리 연산이 가능하다는 장점이 있기 때문이다[1].

그러나 표면 부호는 하나의 정보를 인코딩 하기 위해 많은 물리 큐빗을 요구하므로 부효율이 낮다는 단점이 있다. 예를 들어 부호 거리가 3 이면서 1 개의 정보를 표현하는 논리 큐빗을 제작하려면 최소 17 개의 물리 큐빗이 요구된다[2]. 따라서 qLDPC 와 같은 부호율이 높은 부호에 대한 연구도 진행되고 있으나 해당 부호는 인접 4 개 큐빗이 아닌 그 이상의 연결성이 요구되어 이를 해결하기 위한 연구가 진행되고 있다[3].

한편, Google에서 2025년에 발표된 연구에 따르면, 표면 부호의 부호율을 보완하기 위해 연접 부호를 활용하였다[4]. 연접 부호는 논리 큐빗을 또 인코딩하는 단계를 거쳐 부호 거리를 증가시키는 방법으로 각 단계의 논리 큐빗을 1-level, 2-level 등으로 구분한다[5]. 해당 연구에서는 표면 부호가 아닌 다른 부호들을 활용하여 표면 부호의 거리를 증가시키면서 소요 큐빗을 감소시켰다. 이때 2-level에 활용된 부호들은 인접 4 개 큐빗 이상의 연결성을 요구하나, 표면 부호의 논리 연산 기법인 Lattice surgery 를 활용하여 인접 연산들로 수행할 수 있다. 본 연구는 해당 연구에 제시된 기법을 활용하여 표면 부호와 CSS code 의 일종인 [[5,1,3]] 부호를 연접하여 부호 거리를 증가시키고 공간 비용에 해당하는 소요 큐빗을 감소시킨다. 또한 논리 연산을 통한 2-level 인코딩 과정에 논리 큐빗 구조의 공간 비용을 최적화한다.

II. 본론

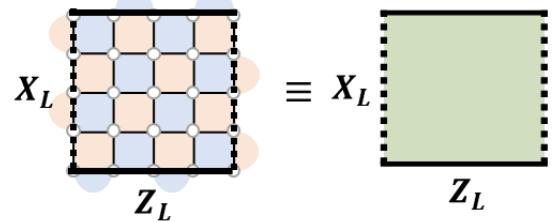


그림 1. 논리 큐빗을 Patch로 표기한 예시 (부호 거리가 5인 경우를 대상으로 하였으나 부호 거리는 다른 값으로 변경 가능)

그림 1 은 부호 거리 $d = 5$ 인 경우를 예시로 들어 논리 큐빗을 박스 형태인 Patch로 나타낸 것이다. 왼쪽 그림에서 각 원은 물리 큐빗을 의미하며, 파란색 면은 X 안정자(Stabilizer)를, 주황색 면은 Z 안정자를 의미한다. 원으로 표기되지는 않았지만, 각 면에는 1 개의 신드롬 측정 큐빗이 존재한다. 논리 큐빗의 X boundary에 존재하는 논리 Z 연산자(Z_L)는 굵은 실선으로 표기하였으며, Z boundary에 존재하는 논리 X 연산자(X_L)는 굵은 점선으로 표기하였다. 해당 논리 큐빗은 오른쪽에 있는 연두색 박스로 대체하여 표기한다. 박스에서 실선과 점선은 각각 논리 Z 연산자와 논리 X 연산자를 의미한다. 부호 거리가 d 인 표면 부호의 소요 큐빗 개수는 $2d^2 - 1 \approx 2d^2$ 이다. 따라서 부호 거리가 5인 해당 예시에서는 50개의 큐빗이 사용된다. d 를 $3d$ 로 증가시키기 위해 표면 부호의 논리 Z 연산자와 논리 X 연산자의 길이(가로-세로 사이즈)를 키울 수 있다. 해당 경우에 소요 큐빗 개수는 $2 \cdot (9d^2) = 18d^2$ 로 9 배 증가한다.

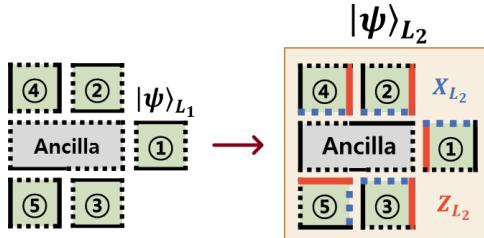


그림 2. 표면 부호와 [[5,1,3]] 부호를 연접하여 생성된 2-level 논리 큐빗

본 논문은 [4]에 제시된 연접 아이디어에서 착안하여, 1-level 논리 큐빗은 부호 거리 d 인 표면 부호로 인코딩하고, 2-level 논리 큐빗은 부호 거리가 3인 [[5,1,3]] 부호로 인코딩하여 연접 부호를 통해 부호 거리를 증가시키는 방식을 제안한다. 그림 2는 연접 부호를 통해 생성된 논리 큐빗을 나타낸 것이며, 연접 부호의 부호 거리는 각 level의 부호 거리를 곱하여 계산되므로 2-level 논리 큐빗의 부호 거리는 $3d$ 이다. 표면 부호로 인코딩된 1번 큐빗이 임의의 $|\psi\rangle_{L_1}$ 상태를 가질 때, ancilla를 포함한 5개의 논리 큐빗 추가로 사용하여 [[5,1,3]] 부호로 한번 더 인코딩 할 수 있다. 이때 생성된 2-level 논리 큐빗은 임의의 $|\psi\rangle_{L_1}$ 와 동일한 정보를 지닌다. 2-level의 논리 연산자는 1-level의 논리 큐빗마다 논리 X 연산을 하거나 논리 Z 연산을 하는 것으로 수행 가능하므로 2-level의 논리 X 연산자와 논리 Z 연산자는 각각 굵은 파란색 점선과 굵은 빨간색 실선으로 표기하였다.

연접 부호의 소요 큐빗 개수는 다음과 같이 계산된다. 먼저, 표면 부호로 인코딩한 논리 큐빗은 5개 사용하였기 때문에 $2 \cdot (5d^2) = 10d^2$ 의 큐빗을 소요한다. 이를 [[5,1,3]] 부호의 안정자로 인코딩하기 위해 $|+\rangle_L$ 상태의 ancilla 논리 큐빗을 추가로 활용하며 이는 2개의 논리 큐빗에 해당하는 크기로 구성된다. Ancilla 논리 큐빗은 $2 \cdot (2d^2) = 4d^2$ 의 큐빗을 소요하므로, 전체 소요 큐빗은 둘을 더한 $14d^2$ 이다. 따라서 제안 방식을 사용하면, 표면 부호의 사이즈를 키워서 $18d^2$ 개의 큐빗을 소모하는 것보다 22.2% 더 적은 큐빗 만으로 부호 거리 3 d 를 지닌 논리 큐빗 1개를 생성할 수 있다.

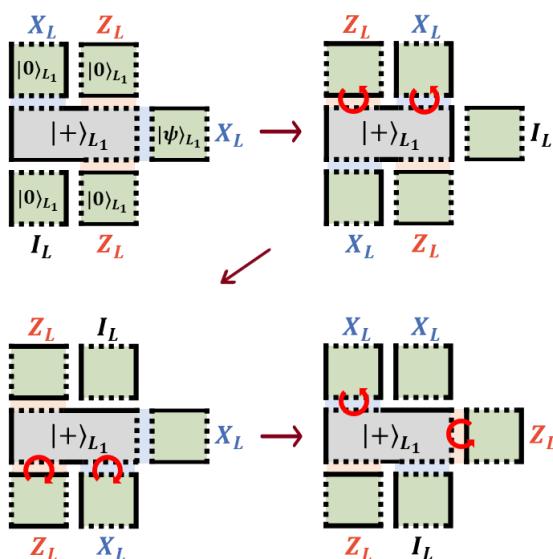


그림 3. 2-level 논리 큐빗의 [[5,1,3]]부호 인코딩 과정

그림 3은 [[5,1,3]] 부호의 안정자로 인코딩하는 과정을 [6]에 제시된 논리 큐빗 구조인 compact block과 lattice surgery 기법을 사용하여 나타낸 것이다.

해당 block은 가장 적은 공간 비용을 사용하며 모든 joint measurement에는 1d cycle이 소요되나, 논리 큐빗의 X boundary를 Z boundary로 변경하는 boundary rotation에는 추가 3d cycle이 소요된다.

각 과정을 살펴보면, 우선 1번 patch에 원하는 정보를 저장하고 2번~5번을 $|0\rangle_L$ 로 준비한다. 이후 [[5,1,3]] 부호 안정자인 $\langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$ 로 사영하는 Joint measurement를 수행한다. 사영 결과는 ancilla patch를 X basis로 측정하여 알 수 있다. 전체 과정은 4번의 joint measurement에 따른 4d cycle과 3번의 rotation에 9d cycle이 소요되어 총 13d cycle이 요구된다.

III. 결론

표면 부호의 낮은 부효율을 보완하기 위해 [[5,1,3]] 부호와 연접하였으며, 2-level 인코딩에 lattice surgery를 통한 joint measurement를 사용하였다. Joint measurement 과정에는 [6]에 제시된 compact block을 활용하여 공간 비용을 최소화하였다. 제안 기법은 부호 거리를 3배 증가하는데 표면 부호의 사이즈를 키우는 방식보다 22.2% 더 적은 큐빗을 사용하였다. 대신, 시간 비용은 13d cycle이 추가로 소요되었다.

추후에는 [6]에 제시된 다른 block 활용하여 공간 비용을 추가하는 대신 시간 비용을 더 줄이거나, [[5,1,3]] 부호가 아닌 다른 부호를 활용하여 부효율을 더 보완하는 연구를 고려할 수 있을 것이다.

ACKNOWLEDGMENT

본 연구 논문은 정부(과학기술정보통신부)의 재원으로 정보통신 기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00014, 결합허용 논리양자큐빗 환경을 제공하는 양자운영체계 원천기술 개발). 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2022-0-00463, 양자인터넷 구현을 위한 유선 양자증계기 개발)

참고 문헌

- [1] Wang, David S., Austin G. Fowler, and Lloyd CL Hollenberg. "Surface code quantum computing with error rates over 1%." *Physical Review A—Atomic, Molecular, and Optical Physics* 83.2 (2011): 020302.
- [2] Tomita, Yu, and Krysta M. Svore. "Low-distance surface codes under realistic quantum noise." *Physical Review A* 90.6 (2014): 062320.
- [3] Panteleev, Pavel, and Gleb Kalachev. "Degenerate quantum LDPC codes with good finite length performance." *Quantum* 5 (2021): 585.
- [4] Gidney, Craig, et al. "Yoked surface codes." *Nature Communications* 16.1 (2025): 4498.
- [5] Devitt, Simon J., William J. Munro, and Kae Nemoto. "Quantum error correction for beginners." *Reports on Progress in Physics* 76.7 (2013): 076001.
- [6] Litinski, Daniel. "A game of surface codes: Large-scale quantum computing with lattice surgery." *Quantum* 3 (2019): 128.