

Heavy Hexagon 구조에서 표면 코드 배치 방법의 비교와 분석

국성연, 정유신, 허준*
고려대학교

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

Comparison and Analysis of Surface Code Mapping Methods on Heavy Hexagon Structure

Sungyeon Kook, Youshin Chung, Jun Heo*
Korea Univ.

요 약

Heavy Hexagon 구조는 제한된 연결성을 가진 양자 컴퓨터에서 안정성과 효율성을 높이기 위해 설계된 레이아웃이다. 본 연구에서는 표면 코드를 Heavy Hexagon 구조에 배치하는 두 가지 방법을 비교하고, 플래그 큐비트와 SWAP 게이트 사용 여부에 따른 자원 사용량과 실행 시간을 분석하였다. 분석 결과를 통해 각 방법의 장단점을 평가하고, 효율적인 배치 설계를 위한 방향성을 제시하였다.

I. 서 론

양자 오류 정정 부호는 양자 컴퓨터가 외부 환경과의 상호작용으로 발생하는 오류를 해결하기 위한 부호이다. 표면 코드(Surface Code)는 높은 안정성과 오류 허용도를 제공하는 양자 오류 정정 부호이다.

양자 컴퓨터 하드웨어는 큐비트 간 연결성에 제한이 있으며, 이는 SWAP 게이트 사용을 증가시켜 회로 실행 시간을 늘리고 오류율을 높이는 원인이 된다. Heavy Hexagon 구조는 큐비트 간 연결성을 3 이하로 제한하여 하드웨어 안정성을 높이고, 편향된 오류 모델에서 효율적인 오류 정정을 하기 위해 제안된 레이아웃이다[2]. 이 구조에서 표면 코드를 배치할 때, 배치 방법에 따라 자원 사용량과 실행 시간의 효율성이 달라질 수 있다.

본 논문은 표면 코드를 두 가지 방법으로 Heavy Hexagon 구조에 배치하고 각 방법의 자원과 실행 시간을 비교한다. 첫번째 배치는 SWAP 게이트를 사용하지 않고 6 개의 플래그 큐비트를 사용하는 방법이며[1,3], 두번째 배치는 SWAP 게이트를 사용한 방법이다. 이를 통해 효율적인 배치 설계를 위한 방안을 제시하고자 한다[2].

II. 본 론

A. 플래그 큐비트

플래그 큐비트(Flat qubit)는 제한된 연결성 환경에서 데이터 큐비트(Data qubit) 또는 안정자 큐비트(Stabilizer qubit)와 상호작용하여, 직접 연결되지 않은 큐비트 간의 연산을 간접적으로 지원한다. 이를 통해 SWAP 게이트 사용을 최소화하며, 게이트 오류를 줄이고 실행 시간을 단축시킨다.

또한, 다중 큐비트 오류로 인한 오류 전파를 방지하고, 고중량 오류(high-weight error)를 사전에 감지하여 정정함으로써 회로의 신뢰성을 높인다[4-6].

B. 배치 1) SWAP 게이트 미사용 방법



그림 1. (왼쪽) Heavy Hexagon 구조에서 SC-17의 배치 1 과 (오른쪽) Z error syndrome 추출 회로

SWAP 게이트를 사용하지 않고 플래그 큐비트를 활용하여 SC-17 을 배치한 방법이다[1]. 그림 1 의 파란색 원은 데이터 큐비트, 초록색 원은 Z 안정자 큐비트, 주황색 원은 X 안정자 큐비트이다. 레이아웃에서 연결성이 3 인 큐비트의 오른쪽 큐비트에 데이터 큐비트를 배치하여 SWAP 게이트로 인한 추가적인 실행 시간과 게이트 오류를 방지하고 회로 효율성을 높였다. 또한 데이터 큐비트 4 개가 1 개의 안정자를 가운데에 둔 마름모 형태를 가져, 표면 코드 구현의 확장에 용이하다[1-3].

그림 1 의 오른쪽 회로는 Z 신드롬 추출 회로이다. 하나의 안정자에 6 개의 플래그 큐비트가 사용되었다. Z 신드롬 추출에는 12 개의 CNOT 게이트와 4 개의 CZ 게이트가, X 신드롬 추출에는 16 개의 CNOT 게이트가 사용된다. CNOT 과 CZ 는 하다마드 게이트를 통해 상호 변환할 수 있으므로 두 게이트의 depth 를 동일하다고 근사하면, 단일 신드롬 추출 회로의 depth 는 9 이다.

C. 배치 2) SWAP 게이트 사용 방법

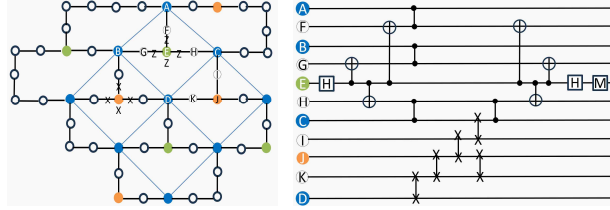


그림 2. (왼쪽) Heavy Hexagon 구조에서 SC-17의 배치 2와 (오른쪽) Z error syndrome 추출 회로

플래그 큐비트와 SWAP 게이트를 사용하여 SC-17을 배치한 방법이다. 그림 2의 왼쪽 그림은 배치 1 상태에서 모든 큐비트를 왼쪽으로 1 칸 이동시킨 배치로, 모든 큐비트가 연결성이 3인 큐비트에 위치하게 된다. 이에 따라 안정자 큐비트를 기준으로 아래쪽에 위치한 데이터 큐비트를 안정화하기 위해 SWAP이 필요하다. 안정자 큐비트 아래쪽 데이터 큐비트와 SWAP하는 데이터 큐비트는 안정자 큐비트의 왼쪽 혹은 오른쪽 큐비트가 가능하며 그림 2는 오른쪽 큐비트를 선택했다.

그림 2의 회로는 Z 신드롬 추출 회로이다. 안정자 큐비트(E)와 플래그 큐비트 1개(F, G, H)로 간접 연결된 데이터 큐비트 3개(A, B, C)를 안정화한 뒤, 아래쪽의 데이터 큐비트(D)가 오른쪽 데이터 큐비트에 위치할 때까지 4번 SWAP한다. 2번째 SWAP에서 데이터 큐비트(D)와 X 안정자(J)를 SWAP했기 때문에, X 안정자(J)를 원래 위치로 되돌려주기 위한 SWAP을 따로 진행한다. 안정자의 위치는 변하면 안 되기 때문에, 이는 모든 안정자에 대해 진행되어야 한다.

배치 2 방법에서는 하나의 안정자에 5개의 플래그 큐비트가 사용된다. SWAP 게이트는 3개의 CNOT 게이트와 동일하게 나타낼 수 있으므로 신드롬 추출에 사용된 CNOT 게이트의 개수는 25개이며, depth는 19이다.

D. 배치 방법에 따른 자원 및 시간 비교

	배치 1	배치 2
Flag qubit의 개수	6	5
CNOT 게이트의 개수	16	25
Depth (CNOT 게이트 기준)	9	19

표 1. 배치 방법에 따른 표면 코드의 단일 신드롬 추출 회로 자원 비교

배치 1 방법은 SWAP 게이트를 사용하지 않고 플래그 큐비트를 활용하였다. SWAP 게이트를 사용하지 않아 추가적인 게이트로 인한 오류 발생 가능성이 줄고, SWAP을 사용하는 회로에 비해 실행 시간이 짧다. 플래그 큐비트의 개수는 배치 2 방법에 비해 1개가 많아 하드웨어 자원소모가 상대적으로 많았다. 또한, 배치 2 방법에 비해 복잡한 플래그 큐비트 배치로 인해 설계가 복잡해질 수 있다.

배치 2 방법은 SWAP 게이트를 사용하여 데이터 큐비트 간의 직접적인 위치 교환이 있다. 플래그 큐비트의 수를 줄여 하드웨어 자원소모가 적다는 장점이 있으며, 큐비트의 배치가 연결성이 3인 위치에 배치하는 단순한 방법이어서 회로 설계가 용이하다. 반면 SWAP 게이트 사용으로 인해 전체 게이트 수와 회로의 depth가 증가하게 된다.

배치 2 방법은 표면 코드와 같이 안정자가 4개의 데이터 큐비트를 안정화하는 것이 아닌, 1개의 안정자가 3개 이하의 데이터 큐비트를 안정화하는 [7,1,3] 코드와

같은 오류 정정 부호에는 SWAP 게이트를 사용하지 않을 수 있다.

III. 결론

본 논문에서는 Heavy Hexagon 구조에서 표면 코드의 두 가지 배치 방법을 비교하였다. 배치 1 방법은 SWAP 게이트를 사용하지 않고 플래그 큐비트를 활용하여 회로의 depth를 줄이고 실행 시간을 단축하는 방법이다. 배치 2 방법은 SWAP 게이트를 사용하는 방법으로, 플래그 큐비트 수를 줄여 하드웨어 자원 소모를 줄였지만, 전체 게이트 수와 depth가 증가했다. 논문에서 제시한 배치 방법은 [7,1,3] 코드 등 다른 양자 오류 정정 부호에 대해서도 확장해 볼 수 있을 것이다. 또한 하드웨어 자원과 Depth 사이의 상충관계를 고려하여 향후 연구에서는 플래그 큐비트와 SWAP 게이트의 최적 조합을 통해 자원 사용과 실행 시간의 균형을 맞추는 방향으로 연구를 진행할 수 있을 것이다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학 ICT 연구센터(ITRC)의 지원(IITP-2025-RS-2021-II211810, 20%)과 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No. 2020-0-00014, 결합허용 논리양자큐비트 환경을 제공하는 양자운영체제 원천기술 개발, 40%)과 2024년도 정보(과학기술정보통신부)의 재원으로 한국연구재단 양자기술연구개발선도(양자컴퓨팅)사업의 지원을 받아 수행된 연구임(RS-2024-00431853, 40%)

참고 문헌

- [1] Kim, Younghun, Jeongsoo Kang, and Younghun Kwon. "Design of quantum error correcting code for biased error on heavy-hexagon structure." *Quantum Information Processing* 22.6 (2023): 230..
- [2] Eickbusch, Alec, et al. "Demonstrating dynamic surface codes." *arXiv preprint arXiv:2412.14360* (2024).
- [3] Kim, Hansol, et al. "Implementation of Magic State Injection within Heavy-Hexagon Architecture." *arXiv preprint arXiv:2412.15751* (2024).
- [4] R. Chao and B. W. Reichardt, "Flag fault-tolerant error correction for any stabilizer code," *PRX Quantum*, vol. 1, no. 1, p. 010302, 2020.
- [5] C. Chamberland et al., "Topological and subsystem codes on low-degree graphs with flag qubits," *Physical Review X*, vol. 10, no. 1, p. 011022, 2020.
- [6] R. Chao and B. W. Reichardt, "Flag fault-tolerant error correction with arbitrary distance codes," *Quantum*, vol. 2, p. 53, 2018.