Lattice Surgery 기반 Checkerboard 아키텍처의 양자 회로 스케줄링

국성연, 허준*
고려대학교

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

Quantum Circuit Scheduling of Checkerboard Architecture Based on Latice Surgery

Kook Sung Yeon, Heo Jun*
Korea Univ.

요 약

본 논문은 Lattice Surgery 기반 surface code 회로를 효율적으로 실행하기 위한 checkerboard 아키텍처에 특화된 스케줄링 최적화 모델을 제안한다. checkerboard 구조의 제약을 반영하여, CNOT 연산의 수행 시간을 인접성 및 SWAP 횟수에 따라 정의하고, 자원 충돌 제약을 수식으로 정의하였다. QODG 기반 레벨 할당을 통해 전체 연산 시작 시점을 결정하고, 회로의 총 실행 시간을 최소화하는 문제로 정의하였다.

Ⅰ. 서 론

Lattice Surgery 기반의 Surface Code 는 양자 연산을 이웃 큐비트 간의 병합 및 분할 연산만으로 구현할 수 있어, 제한된 연결성을 가지는 2 차원 양자 하드웨어에서 효과적이다. 이는 Braiding 기반 연산 방식에 비해 연산 경로가 단순하고 자원 소모가 적다.

Lattice Surgery 연산을 효과적으로 수행하기 위해 논리적 데이터 큐비트와 논리적 안실라 큐비트를 surface code 격자에 어떻게 배치할 것인지를 정의하는 아키텍처 설계는 자원 효율성과 latency 측면에서 핵심적이다[1, 2]. 아키텍처가 정의되면, 양자 회로를 해당 구조 내에서 배치하고 실행할 방법을 결정하는 스케줄링 프로세스가 요구된다. 연산 순서를 조정하고 물리적 자원 충돌을 회피함으로써, 회로 실행 시간을 단축하고 ancilla 자원을 효율적으로 재사용할 수 있다.[3]

본 논문에서는 [1]에서 제안된 checkerboard 아키텍처를 대상으로 양자 회로를 실행하기 위한 스케줄링 수식 모델을 제시한다. 연산 가능성에 따라 달라지는 조건부 연산 수행 시간 모델을 포함하여, checkerboard 아키텍처의 스케줄링 제약 조건을 수식으로 정의하였다.

Ⅱ. 본론

1. Checkerboard 구조

Checkerboard 아키텍처는 이러한 요구를 반영하여 제안된 구조로, 모든 논리적 데이터 큐비트를 격자의 대각선 위치에 고정 배치하고, 이들 사이의 타일을 논리적 안실라 큐비트로 활용하는 방식이다. 그림 1 은 checkerboard 아키텍처의 큐비트 배치 예시를 나타낸다. 분홍색 타일은 논리적 데이터 큐비트를, 회색 타일은 논리적 안실라 큐비트를 의미한다. 각 논리 큐비트는 해당 행과 열이 모두 짝수 또는 모두 홀수인 위치에 배치되며, 이를 통해 90° elbow-shaped 구조의 CNOT 연산이 사전 정의된 ancilla 를 경유하여 수행되도록 구성된다.[1]

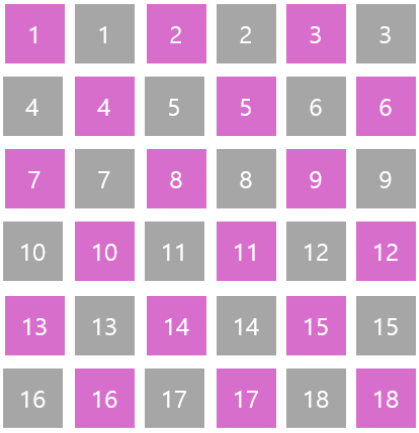

그림 1. Checkerboard 아키텍처 상 logical qubit 배치

Checkerboard 아키텍처의 큐비트 효율성은 1/2 이며, 모든 CNOT 연산은 이미 90° elbow-shaped 형태로 ancilla 경유 경로가 형성되어 있다. 인접한 데이터 큐비트 간의 CNOT 연산에 필요한 surface code cycle 은 $3d$ 이다. 인접하지 않은 데이터 큐비트 간 연산을 수행하기 위해서는 논리 큐비트의 위치를 이동시키는 SWAP 연산이 필요하다. 단일 SWAP 연산의

비용은 $9d$이며, 스케줄링 과정에서 연산 수행 시간 T_g을 정의하기 위해 각 연산 수행 전 필요한 SWAP 횟수를 사전에 평가하고 연산 시간에 포함해야 한다.

2. Checkerboard 구조의 양자 회로 스케줄링

1 단계: QODG 구성

입력 양자 회로로부터 모든 CNOT 연산을 추출하고, 동일한 큐비트를 공유하는 연산 쌍에 대해 간선을 부여하여 Quantum Operation Dependency Graph(QODG)를 구성한다. 각 노드는 단일 CNOT 연산을 나타내며, 위상 정렬을 통해 수행 순서를 정의할 수 있다.

2 단계: 연산 레벨 할당

QODG 에 대해 ASAP(As Soon As Possible) 또는 ALAP(As Late As Possible) 전략을 적용하여 각 연산 g 에 레벨 l_g을 할당한다. 레벨은 연산 간 데이터 의존성을 보존하면서 병렬 실행 가능한 최소 시점을 나타낸다. 추후 자원 제약을 고려한 연산 시작 시간은 레벨에 기반하여 결정된다.

3 단계: 연산 수행 가능 여부 판단

각 CNOT 연산 $g = CNOT(q_c, q_t)$에 대해 현재 큐비트 배치 상에서 인접한 ancilla tile 을 통해 직접 수행이 가능한지를 판별한다. 인접한 경우에는 해당 ancilla 를 사용하여 연산을 즉시 수행하며, 인접하지 않은 경우에는 SWAP 연산을 통해 두 큐비트를 이동시킨 후 연산을 수행한다.

4 단계: 연산 수행 시간 계산

연산 g의 수행 시간 T_g은 수행 방식에 따라 다음과 같이 정의된다.

$$T_g = \begin{cases} 3d, & if\ adj(q_c, q_t) = True \\ 3d + 9d \cdot s, & otherwise \end{cases} \quad (1)$$

인접 큐비트 간에는 사전에 형성된 90° elbow-shaped ancilla 경로를 통해 CNOT 을 수행한다. $adj(q_c, q_t)$ 는 큐비트 쌍이 ancilla 를 통해 직접 연결 가능한지 유무를 나타낸다. 인접하지 않은 큐비트 간에는 s 번의 추가 SWAP 을 통해 정보를 교환한 뒤 CNOT 을 수행한다.

5 단계: 자원 충돌 제약 하의 시간 배정

각 연산 g의 시작 시점 S_g은 세 가지의 제약 조건을 만족해야 한다.

$$S_g \geq l_g \cdot T_{cycle} \quad (2)$$

$$S_{g_i} + T_{g_i} \leq S_{g_j}\ or\ S_{g_j} + T_{g_j} \leq S_{g_i} \quad (3)$$

$$q \in g_i \cap g_j \rightarrow S_{g_i} + T_{g_i} \leq S_{g_j}\ or\ S_{g_j} + T_{g_j} \leq S_{g_i} \quad (4)$$

식 (2)는 연산의 수행 순서를 보장하기 위한 조건으로, T_{cycle}는 한 레벨의 최소 보장 시간이다. 시작시간 S_g가 연산 레벨에 비례하는 최소 시간 이상에서 시작되어야 함을 나타낸다. 식 (3)은 동일한 ancilla 타일을 사용하는 두 연산이 동시에 실행되지 않도록 하는 조건으로, 한 연산이 끝난 후에만 다른 연산이 시작될 수 있도록 한다. 식 (4)는 동일한 논리적 큐비트를 두 연산에서 동시에 수행할 수 없도록 하는 조건으로, 연산 간 시간 구간이 겹치지 않도록 한다.

6 단계: 전체 실행 시간 계산 및 최적화

모든 연산 $g \in V$에 대해 시작 시점 S_g과 수행 시간 T_g이 결정되면, 전체 회로의 실행 시간 Δ은 식 (5)와 같다.

$$\Delta = \max_{g \in V}(S_g + T_g) \quad (5)$$

식 (5)는 모든 연산의 종료 시점 중 가장 늦은 시간을 나타내며, 회로 전체의 실행 시간이다. 스케줄링의 최종 목적은 이 Δ값을 최소화하는 것이다. 다양한 휴리스틱 알고리즘 또는 정수 선형 계획 기반 기법과 연계하여 실제 스케줄링 구현으로 확장될 수 있다.

Ⅲ. 결론

본 논문에서는 checkerboard 아키텍처의 물리적 제약을 반영하여, QODG 기반 양자 회로에 대해 구조 인식형 스케줄링 모델을 제안하였다. 제안한 모델은 각 연산의 수행 가능 여부를 ancilla 타일의 위치에 따라 판별하고, SWAP 이 포함된 조건부 연산 시간 정의와 자원 충돌 방지 조건을 수식적으로 통합하였다. 이를 통해 [3]에서 제안된 Ecmas 모델이 전제한 고정 시간 기반 스케줄링을 일반화하고, 아키텍처 제약을 현실적으로 반영한 실행 일정을 구성할 수 있음을 보였다.

본 모델은 연산 간의 논리적 의존성뿐 아니라, 아키텍처 구조로부터 유도되는 공간적 제약을 함께 고려함으로써 실제 하드웨어에 보다 적합한 실행 스케줄을 생성할 수 있다. 향후 tile-based, row-type 등의 다른 아키텍처에도 adjacency 조건과 ancilla 경로 정의만을 조정함으로써 본 모델을 확장 적용할 수 있을 것으로 기대된다.

ACKNOWLEDGMENT

본 연구 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00014, 결함허용 논리양자큐빗 환경을 제공하는 양자운영체제 원천기술 개발). 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학 ICT 연구센터(ITRC)의 지원을 받아 수행된 연구임(RS-2021-II211810)

참 고 문 헌

[1] Lao, Lingling, et al. "Mapping of lattice surgery-based quantum circuits on surface code architectures." Quantum Science and Technology 4.1 (2018): 015005.

[2] Lee, Jonghyun, et al. "Lattice surgery-based Surface Code architecture using remote logical CNOT operation." Quantum Information Processing 21.6 (2022): 217.

[3] Zhu, Mingzheng, et al. "Ecmas: Efficient circuit mapping and scheduling for surface code." 2024 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). IEEE, 2024.