



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0062867
(43) 공개일자 2025년05월08일

(51) 국제특허분류(Int. Cl.)	(71) 출원인
<i>G06N 10/70</i> (2022.01) <i>G06N 10/20</i> (2022.01)	고려대학교 산학협력단
<i>G06N 10/40</i> (2022.01) <i>B82Y 10/00</i> (2017.01)	서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(52) CPC특허분류	(72) 발명자
<i>G06N 10/70</i> (2025.05)	강유진
<i>G06N 10/20</i> (2025.05)	서울특별시 성북구 지봉로21길 22, 401호(보문동2가)
(21) 출원번호 10-2023-0148279	이종현
(22) 출원일자 2023년10월31일	경기도 성남시 분당구 불정로 397, 307동 404호(서현동, 효자촌임광아파트)
심사청구일자 2023년10월31일	(뒷면에 계속)
	(74) 대리인
	이대호, 박건홍

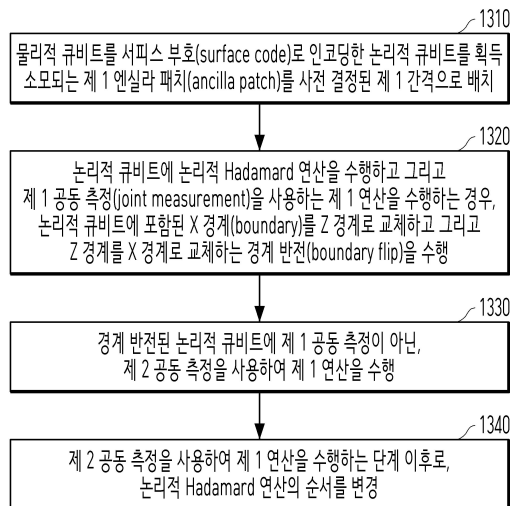
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **경계 반전과 후속 측정의 변형을 사용한 논리적 Hadamard 연산 구현 방법 및 장치**

(57) 요약

본 개시의 일 실시예에 따라, 양자 컴퓨팅 장치에 의해 수행되는 논리적 연산의 방법으로서, 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 단계; 상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 단계; 상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계; 및 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;를 포함할 수 있다.

대표도 - 도13



(52) CPC특허분류

G06N 10/40 (2025.05)

B82Y 10/00 (2013.01)

(72) 발명자

하진영

서울특별시 성북구 고려대로12길 13-3, 302호(안암동2가)

허준

서울특별시 강남구 광평로10길 6, 206동 103호(일원동, 한솔마을아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711152872
과제번호	2020-0-00014-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	한국전자통신연구원연구개발지원
연구과제명	결함허용 논리양자큐빗 컴퓨팅 환경을 제공하는 양자운영체제 원천기술 개발
기 여 율	40/100
과제수행기관명	한국전자통신연구원
연구기간	2023.01.01 ~ 2023.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	2710008721
과제번호	2021-0-01810-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성
연구과제명	양자세계를 연결하는 초신뢰 양자인터넷 요소 기술 개발
기 여 율	30/100
과제수행기관명	고려대학교산학협력단
연구기간	2024.01.01 ~ 2024.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711196351
과제번호	RS-2023-00242396
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	미래유망융합기술파이오니어
연구과제명	가입자 광통신망 적용을 위한 양자암호 기반 도청방지 기술
기 여 율	30/100
과제수행기관명	고려대학교산학협력단
연구기간	2023.04.01 ~ 2023.12.31

명세서

청구범위

청구항 1

양자 컴퓨팅 장치에 의해 수행되는 논리적 연산의 방법으로서,

물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 단계;

상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 단계;

상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계; 및

상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;

를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 X 경계는 상기 논리적 큐비트에 포함된 X 스테빌라이저(stabilizer)로 구성된 경계이고, 그리고

상기 Z 경계는 상기 논리적 큐비트에 포함된 Z 스테빌라이저로 구성된 경계인,

방법.

청구항 3

제 1 항에 있어서,

상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는:

상기 제 1 연산이 논리적 CNOT인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZZ 측정(joint ZZ measurement)이 아닌, 상기 제 2 공동 측정인 공동 XZ 측정(joint XZ measurement)을 수행하는 단계;

를 포함하는,

방법.

청구항 4

제 1 항에 있어서,

상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는:

상기 제 1 연산이 패치 확장(patch extension)인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동

측정인 공동 XX 측정(joint XX measurement)이 아닌, 상기 제 2 공동 측정인 공동 ZX 측정(joint ZX measurement)을 수행하는 단계;

를 포함하는,

방법.

청구항 5

제 1 항에 있어서,

상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는:

상기 제 1 연산이 논리적 S 연산인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZY 측정(joint ZY measurement)이 아닌, 상기 제 2 공동 측정인 공동 XY 측정(joint XY measurement)을 수행하는 단계;

를 포함하는,

방법.

청구항 6

제 1 항에 있어서,

상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는:

상기 제 1 연산이 논리적 T 연산인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZZ 및 ZY 측정(joint ZZ & ZY measurement)이 아닌, 상기 제 2 공동 측정인 공동 ZX 및 XY 측정(joint XY measurement)을 수행하는 단계;

를 포함하는,

방법.

청구항 7

제 1 항에 있어서,

변경된 상기 논리적 Hadamard 연산의 순서에 후속하여 제 3 공동 측정을 사용하는 제 2 연산이 존재하는 경우, 상기 제 1 연산이 수행된 논리적 큐비트에 상기 경계 반전을 수행하는 단계;

상기 경계 반전 및 상기 제 1 연산이 수행된 논리적 큐비트에 상기 제 3 공동 측정이 아닌, 제 4 공동 측정을 사용하여 상기 제 2 연산을 수행하는 단계; 및

상기 제 4 공동 측정을 사용하여 상기 제 2 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;

를 더 포함하는,

방법.

청구항 8

제 1 항에 있어서,

변경된 상기 논리적 Hadamard 연산의 순서에 후속하여 단일 측정(single measurement)이 존재하는 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저(basis)를 변경하는 단계;

를 더 포함하는,

방법.

청구항 9

제 8 항에 있어서,

상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저를 변경하는 단계는:

상기 단일 측정에서 측정하는 기저가 X 기저인 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 측정하는 기저를 Z 기저로 변경하는 단계;

를 포함하는,

방법.

청구항 10

제 8 항에 있어서,

상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저를 변경하는 단계는:

상기 단일 측정에서 측정하는 기저가 Z 기저인 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 측정하는 기저를 X 기저로 변경하는 단계;

를 포함하는,

방법.

청구항 11

컴퓨터 프로그램을 포함하는 비일시적 컴퓨터 판독 가능 매체로서, 상기 컴퓨터 프로그램은 양자 컴퓨팅 장치의 프로세서로 하여금 논리적 연산을 위한 방법을 수행하도록 하며, 상기 방법은:

물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 단계;

상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 단계;

상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계; 및

상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;

를 포함하는,

컴퓨터 프로그램을 포함하는 비일시적 컴퓨터 판독 가능 매체.

청구항 12

논리적 연산을 위한 양자 컴퓨팅 장치에 있어서,

프로세서; 및

메모리;

를 포함하고,

상기 프로세서는:

물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 동작;

상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 동작;

상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 동작; 및

상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 동작;

을 수행하는,

컴퓨팅 장치.

발명의 설명

기술 분야

[0001] 본 개시는 논리적 연산 기술에 관한 것으로, 보다 구체적으로, 경계 반전과 후속 측정의 변형을 사용한 논리적 Hadamard 연산 구현 방법 및 장치에 관한 것이다.

배경 기술

[0002] 양자 컴퓨팅 장치는 양자 얽힘(quantum entanglement), 양자 중첩(quantum superposition) 등의 양자역학과 관련된 현상을 이용하여 데이터를 처리하는 컴퓨팅 장치일 수 있다. 양자 얽힘은 둘 이상의 상태가 양자적으로 서로 연결되어 있어, 각각의 상태로 따로 다룰 수 없는 상태를 의미할 수 있다. 양자 중첩은 양자 상태를 측정하기 전에 측정에 의한 여러 결과 상태가 확률적으로 동시에 존재하는 것을 의미할 수 있다. 양자 컴퓨팅 장치는 양자역학과 관련된 현상을 이용하여 데이터를 처리하기 위한 정보의 기본 단위로 큐비트(qubit)를 사용할 수 있다.

[0003] 큐비트는 양자 중첩 상태를 이용하여 여러 비트에 대응되는 값을 동시에 표현이 가능할 수 있다. 예를 들어, 큐비트는 '20%의 확률로 0, 80%의 확률로 1'과 같이 각각의 값들을 확률로 표현할 수 있다. 큐비트는 관측될 때 양자 중첩 상태가 풀리면서 하나의 상태로 결정될 수 있다.

[0004] 양자 정보에 대한 노이즈 또는 오류(error)로 인한 양자 컴퓨팅 및 양자 정보 통신의 구현의 어려움을 극복하기 위해서, 양자 오류정정 부호(quantum error correcting code)로서 서피스 부호(surface code) 및 연결기반 부호(concatenated code)가 연구되고 있다.

[0005] 서피스 부호는 양자 오류 정정 부호가 효율적으로 동작하기 위해서 물리적 큐비트에 요구되는 오류율에 대한 임계 값(threshold)이 연결기반 부호보다 높으며, 향후 양자 오류 정정 부호로 활용될 수 있다.

[0006] 물리적 큐비트를 서피스 부호로 인코딩한 양자 컴퓨터 아키텍처에서 공동 측정(joint measurement)을 사용하여 논리적 연산을 구현하는 방법이 제안되고 있다. 그리고 공동 측정(joint measurement)을 사용하여 논리적 연산을 효율적으로 구현하기 위한 연구가 지속되고 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 미국 공개특허 16-980592(2021.12.02. 공개)

발명의 내용

해결하려는 과제

[0008] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, 경계 반전과 후속 측정의 변형을 사용한 논리적 Hadamard 연산 구현 방법 및 장치를 제공하고자 한다.

[0009] 본 개시의 기술적 과제들은 이상에서 언급한 기술적 과제로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0010] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 일 실시예에 따라, 양자 컴퓨팅 장치에 의해 수행되는 논리적 연산의 방법으로서, 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 단계; 상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 단계; 상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계; 및 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;를 포함할 수 있다.

[0011] 대안적으로, 상기 X 경계는 상기 논리적 큐비트에 포함된 X 스테빌라이저(stabilizer)로 구성된 경계이고, 그리고 상기 Z 경계는 상기 논리적 큐비트에 포함된 Z 스테빌라이저로 구성된 경계일 수 있다.

[0012] 대안적으로, 상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는: 상기 제 1 연산이 논리적 CNOT인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZZ 측정(joint ZZ measurement)이 아닌, 상기 제 2 공동 측정인 공동 XZ 측정(joint XZ measurement)을 수행하는 단계;를 포함할 수 있다.

[0013] 대안적으로, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는: 상기 제 1 연산이 패치 확장(patch extension)인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 XX 측정(joint XX measurement)이 아닌, 상기 제 2 공동 측정인 공동 ZX 측정(joint ZX measurement)을 수행하는 단계;를 포함할 수 있다.

[0014] 대안적으로, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는: 상기 제 1 연산이 논리적 S 연산인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZY 측정(joint ZY measurement)이 아닌, 상기 제 2 공동 측정인 공동 XY 측정(joint XY measurement)을 수행하는 단계;를 포함할 수 있다.

[0015] 대안적으로, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계는: 상기 제 1 연산이 논리적 T 연산인 경우, 상기 경계 반전된 상기 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZZ 및 ZY 측정(joint ZZ & ZY measurement)이 아닌, 상기 제 2 공동 측정인 공동 ZX 및 XY 측정(joint XY measurement)을 수행하는 단계;를 포함할 수 있다.

[0016] 대안적으로, 변경된 상기 논리적 Hadamard 연산의 순서에 후속하여 제 3 공동 측정을 사용하는 제 2 연산이 존재하는 경우, 상기 제 1 연산이 수행된 논리적 큐비트에 상기 경계 반전을 수행하는 단계; 상기 경계 반전 및 상기 제 1 연산이 수행된 논리적 큐비트에 상기 제 3 공동 측정이 아닌, 제 4 공동 측정을 사용하여 상기 제 2 연산을 수행하는 단계; 및 상기 제 4 공동 측정을 사용하여 상기 제 2 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;를 더 포함할 수 있다.

[0017] 대안적으로, 변경된 상기 논리적 Hadamard 연산의 순서에 후속하여 단일 측정(single measurement)이 존재하는 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저(basis)를 변경하는 단계;를 더 포함할 수 있다.

[0018] 대안적으로, 상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저를 변경하는 단계는: 상기 단일 측정에서 측정하는 기저가 X 기저인 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 측정하는 기저를 Z 기저로 변경하는 단계;를 포함할 수 있다.

[0019] 대안적으로, 상기 단일 측정에서 상기 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저를 변경하는 단계는: 상기 단일 측정에서 측정하는 기저가 Z 기저인 경우, 상기 논리적 Hadamard 연산과 상기 단일 측정을 병합함으로써, 상기 단일 측정에서 측정하는 기저를 X 기저로 변경하는 단계;를 포함할 수 있다.

[0020] 본 개시의 일 실시예에 따라, 컴퓨터 프로그램을 포함하는 비일시적 컴퓨터 판독 가능 매체로서, 상기 컴퓨터 프로그램은 양자 컴퓨팅 장치의 프로세서로 하여금 논리적 연산을 위한 방법을 수행하도록 하며, 상기 방법은: 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 단계; 상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 단계; 상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계; 및 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 단계;를 포함할 수 있다.

[0021] 본 개시의 일 실시예에 따라, 논리적 연산을 위한 양자 컴퓨팅 장치에 있어서, 프로세서; 및 메모리;를 포함하고, 상기 프로세서는: 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득하는 동작; 상기 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 상기 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행하는 동작; 상기 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 동작; 및 상기 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 상기 논리적 Hadamard 연산의 순서를 변경하는 동작;을 수행할 수 있다.

발명의 효과

[0022] 본 개시는 경계 반전과 후속 측정의 변형을 사용함으로써, 적은 시간 및 공간 비용으로 논리적 Hadamard 연산을 수행할 수 있다.

[0023] 본 개시에서 얻을 수 있는 효과는 이상에서 언급한 효과로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 개시가 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0024] 다양한 양상들이 이제 도면들을 참조로 기재되며, 여기서 유사한 참조 번호들은 총괄적으로 유사한 구성요소들을 지칭하는데 이용된다. 이하의 실시예에서, 설명 목적을 위해, 다수의 특정 세부사항들이 하나 이상의 양상들의 총체적 이해를 제공하기 위해 제시된다. 그러나, 그러한 양상(들)이 이러한 구체적인 세부사항들 없이 실시될 수 있음은 명백할 것이다.

도 1은 본 개시의 일 실시예에 따른 논리적 연산을 제공하기 위한 양자 컴퓨팅 장치의 블록 구성도이다.

도 2는 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 논리적 연산의 방법을 나타낸 도면이다.

도 3은 본 개시의 일 실시예에 따른 서피스 부호(Surface code)로 인코딩된 논리적 큐비트 및 이에 대응되는 패치를 나타낸 도면이다.

도 4는 본 개시의 일 실시예에 따른 X 스테빌라이저(stabilizer) 및 Z 스테빌라이저의 패리티 측정 회로와 실행 순서를 나타낸 도면이다.

도 5는 본 개시의 일 실시예에 따른 joint ZY measurement 과정 및 4-qubit twist defect의 패리티 측정 회로를 나타낸 도면이다.

도 6은 본 개시의 일 실시예에 따른 joint XY measurement 과정 및 4-qubit twist defect의 패리티 측정 회로를 나타낸 도면이다.

도 7은 본 개시의 일 실시예에 따른 4-qubit twist defect에서 2개의 measurement 큐비트를 활용한 패리티 측

정 회로를 나타낸 도면이다.

도 8은 본 개시의 일 실시예에 따른 joint ZZ & ZY measurement 과정을 나타낸 도면이다.

도 9는 본 개시의 일 실시예에 따른 논리적 S 연산 수행 과정을 나타낸 도면이다.

도 10은 본 개시의 일 실시예에 따른 논리적 S 연산 수행 시, 스테빌라이저의 변화를 나타낸 도면이다.

도 11은 본 개시의 일 실시예에 따른 논리적 SH 연산 수행 과정을 나타낸 도면이다.

도 12는 본 개시의 일 실시예에 따른 논리적 T 연산 수행 과정을 나타낸 도면이다.

도 13은 본 개시의 일 실시예에 따른 경계 반전과 후속 측정의 변형을 사용한 논리적 Hadamard 연산 구현 방법을 나타낸 도면이다.

도 14는 본 개시의 일 실시예에서 경계 반전에 따른 물리적 CNOT 실행 순서를 나타낸 도면이다.

도 15a는 본 개시의 일 실시예에서 경계 반전 이전에 스테빌라이저, 논리적 X, Z를 나타낸 도면이다.

도 15b는 본 개시의 일 실시예에서 경계 반전을 수행 중인 스테빌라이저, 논리적 X, Z를 나타낸 도면이다.

도 15c는 본 개시의 일 실시예에서 경계 반전을 수행한 이후의 스테빌라이저, 논리적 X, Z를 나타낸 도면이다.

도 15d는 본 개시의 일 실시예에서 경계 반전을 수행한 이후, 새로 활성화된 스테빌라이저를 곱하여 새로운 논리적 연산을 정의하는 과정을 나타낸 도면이다.

도 16은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 CNOT 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

도 17은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 S 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

도 18은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 T 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

도 19는 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 단일 측정(single measurement)인 경우 수행되는 측정을 나타낸 도면이다.

도 20은 본 개시의 내용의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 일반적인 개략도이다.

발명을 실시하기 위한 구체적인 내용

[0025] 다양한 실시예들이 이제 도면을 참조하여 설명된다. 본 명세서에서, 다양한 설명들이 본 개시의 이해를 제공하기 위해서 제시된다. 그러나, 이러한 실시예들은 이러한 구체적인 설명 없이도 실행될 수 있음이 명백하다.

[0026] 본 명세서에서 사용되는 용어 "컴포넌트", "모듈", "시스템" 등은 컴퓨터-관련 엔티티, 하드웨어, 펌웨어, 소프트웨어, 소프트웨어 및 하드웨어의 조합, 또는 소프트웨어의 실행을 지칭한다. 예를 들어, 컴포넌트는 프로세서 상에서 실행되는 처리과정(procedure), 프로세서, 객체, 실행 스레드, 프로그램, 및/또는 컴퓨터일 수 있지만, 이들로 제한되는 것은 아니다. 예를 들어, 컴퓨팅 장치에서 실행되는 애플리케이션 및 컴퓨팅 장치 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트는 프로세서 및/또는 실행 스레드 내에 상주할 수 있다. 일 컴포넌트는 하나의 컴퓨터 내에 로컬화 될 수 있다. 일 컴포넌트는 2개 이상의 컴퓨터들 사이에 분배될 수 있다. 또한, 이러한 컴포넌트들은 그 내부에 저장된 다양한 데이터 구조들을 갖는 다양한 컴퓨터 판독가능한 매체로부터 실행할 수 있다. 컴포넌트들은 예를 들어 하나 이상의 데이터 패킷들을 갖는 신호(예를 들면, 로컬 시스템, 분산 시스템에서 다른 컴포넌트와 상호작용하는 하나의 컴포넌트로부터의 데이터 및/또는 신호를 통해 다른 시스템과 인터넷과 같은 네트워크를 통해 전송되는 데이터)에 따라 로컬 및/또는 원격 처리들을 통해 통신할 수 있다.

[0027] 더불어, 용어 "또는"은 배타적 "또는"이 아니라 내포적 "또는"을 의미하는 것으로 의도된다. 즉, 달리 특정되지 않거나 문맥상 명확하지 않은 경우에, "X는 A 또는 B를 이용한다"는 자연적인 내포적 치환 중 하나를 의미하는 것으로 의도된다. 즉, X가 A를 이용하거나; X가 B를 이용하거나; 또는 X가 A 및 B 모두를 이용하는 경우, "X는 A 또는 B를 이용한다"가 이들 경우들 어느 것으로도 적용될 수 있다. 또한, 본 명세서에 사용된 "및/또는"이라는 용어는 열거된 관련 아이템들 중 하나 이상의 아이템의 가능한 모든 조합을 지칭하고 포함하는 것으로 이해되어야 한다.

- [0028] 또한, "포함한다" 및/또는 "포함하는"이라는 용어는, 해당 특징 및/또는 구성요소가 존재함을 의미하는 것으로 이해되어야 한다. 다만, "포함한다" 및/또는 "포함하는"이라는 용어는, 하나 이상의 다른 특징, 구성요소 및/또는 이들의 그룹의 존재 또는 추가를 배제하지 않는 것으로 이해되어야 한다. 또한, 달리 특정되지 않거나 단수 형태를 지시하는 것으로 문맥상 명확하지 않은 경우에, 본 명세서와 청구범위에서 단수는 일반적으로 "하나 또는 그 이상"을 의미하는 것으로 해석되어야 한다.
- [0029] 그리고, "A 또는 B 중 적어도 하나"이라는 용어는, "A만을 포함하는 경우", "B 만을 포함하는 경우", "A와 B의 구성으로 조합된 경우"를 의미하는 것으로 해석되어야 한다.
- [0030] 당업자들은 추가적으로 여기서 개시된 실시예들과 관련되어 설명된 다양한 예시적 논리적 블록들, 구성들, 모듈들, 회로들, 수단들, 로직들, 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 양쪽 모두의 조합들로 구현될 수 있음을 인식해야 한다. 하드웨어 및 소프트웨어의 상호교환성을 명백하게 예시하기 위해, 다양한 예시적 컴포넌트들, 블록들, 구성들, 수단들, 로직들, 모듈들, 회로들, 및 단계들은 그들의 기능성 측면에서 일반적으로 위에서 설명되었다. 그러한 기능성이 하드웨어로 또는 소프트웨어로서 구현되는지 여부는 전반적인 시스템에 부과된 특정 어플리케이션(application) 및 설계 제한들에 달려 있다. 숙련된 기술자들은 각각의 특정 어플리케이션들을 위해 다양한 방법으로 설명된 기능성을 구현할 수 있다. 다만, 그러한 구현의 결정들이 본 개시내용의 영역을 벗어나게 하는 것으로 해석되어서는 안 된다.
- [0031] 제시된 실시예들에 대한 설명은 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이다. 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예 들로 한정되는 것이 아니다. 본 발명은 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.
- [0032] 본 개시에서 제 1, 제 2, 또는 제 3 과 같이 제 N으로 표현되는 용어들은 적어도 하나의 엔티티들을 구분하기 위해 사용된다. 예를 들어, 제 1 과 제 2로 표현된 엔티티들은 서로 동일하거나 또는 상이할 수 있다.
- [0033] 그리고, 본 개시에서 "A, B 등"과 같이 "~ 등"이라는 용어는, "A만을 포함하는 경우", "B 만을 포함하는 경우", "A와 B의 구성으로 조합된 경우"를 의미하는 것으로 해석되어야 한다.
- [0035] 도 1은 본 개시의 일 실시예에 따른 논리적 연산을 제공하기 위한 양자 컴퓨팅 장치의 블록 구성도이다.
- [0036] 도 1에 도시된 양자 컴퓨팅 장치의 구성은 간략화 하여 나타낸 예시일 뿐이다. 본 개시의 일 실시예에서 양자 컴퓨팅 장치는 논리적 연산을 제공하기 위한 다른 구성들이 포함될 수 있고, 개시된 구성들 중 일부만이 양자 컴퓨팅 장치를 구성할 수도 있다.
- [0037] 도 1을 참조하면, 양자 컴퓨팅 장치(100)는 공동 측정을 사용하여 논리적 연산을 수행할 수 있다.
- [0038] 양자 컴퓨팅 장치(100)는 프로세서(110), 메모리(130), 네트워크부(150)를 포함할 수 있다. 프로세서(110)는 하나 이상의 코어로 구성될 수 있으며, 컴퓨팅 장치의 중앙 처리 장치(CPU: central processing unit), 범용 그래픽 처리 장치 (GPGPU: general purpose graphics processing unit), 텐서 처리 장치(TPU: tensor processing unit) 등의 데이터 분석, 양자 처리를 위한 프로세서를 포함할 수 있다. 프로세서(110)는 통상적으로 양자 컴퓨팅 장치(100)의 전반적인 동작을 제어할 수 있다. 예를 들어, 프로세서(110)는 양자 컴퓨팅 장치(100)에 포함된 구성요소들을 통해 입력 또는 출력되는 신호, 데이터 정보 등을 처리할 수 있다. 다른 예를 들어, 프로세서(110)는 메모리(130)에 저장된 컴퓨터 프로그램을 판독하여 본 개시의 일 실시예에 따른 양자 처리를 수행할 수 있다. 프로세서(110)는 양자 중첩의 지수적인 정보 표현, 양자 얽힘을 이용한 병렬 연산과 같은 양자 역학적인 물리현상을 활용하여 계산을 수행할 수 있다.
- [0039] 본 개시의 몇몇 실시예에 따르면, 메모리(130)는 프로세서(110)가 생성하거나 결정한 임의의 형태의 정보 및 네트워크부(150)가 수신한 임의의 형태의 정보를 저장할 수 있다.
- [0040] 본 개시의 몇몇 실시예에 따르면, 메모리(130)는 플래시 메모리 타입(flash memory type), 하드디스크 타입(hard disk type), 멀티미디어 카드 마이크로 타입(multimedia card micro type), 카드 타입의 메모리(예를 들어 SD 또는 XD 메모리 등), 램(Random Access Memory, RAM), SRAM(Static Random Access Memory), 롬(ReadOnly Memory, ROM), EEPROM(Electrically Erasable Programmable Read-Only Memory), PROM(Programmable Read-Only Memory), 자기 메모리, 자기 디스크, 광디스크 중 적어도 하나의 타입의 저장매체를 포함할 수 있다.

양자 컴퓨팅 장치(100)는 인터넷(internet) 상에서 상기 메모리(130)의 저장 기능을 수행하는 웹 스토리지(web storage)와 관련되어 동작할 수도 있다. 전송할 메모리에 대한 기제는 예시일 뿐, 본 개시는 이에 제한되지 않는다. 본 개시의 몇몇 실시예에 따른 네트워크부(150)는 임의의 형태의 데이터 및 신호 등을 송수신할 수 있는 임의의 유무선 통신 네트워크를 포함할 수 있다. 본 명세서에서 설명된 기술들은 위에서 언급된 네트워크들뿐만 아니라, 다른 네트워크들에서도 사용될 수 있다.

[0042] 도 2는 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 논리적 연산의 방법을 나타낸 도면이다.

[0043] 도 2에 도시되는 단계들은 예시적인 단계들이다. 따라서, 본 개시 내용의 사상 범위를 벗어나지 않는 한도에서도 2의 단계들 중 일부가 생략되거나 추가적인 단계들이 존재할 수 있다는 점 또한 당업자에게 명백할 것이다. 또한, 도 2에 기재된 구성들(예를 들어, 양자 컴퓨팅 장치(100) 등)에 관한 구체적인 내용은 앞서 도 1을 통해 설명한 내용으로 대체될 수 있다. 도 2 내지 도 19에 도시되는 단계들, 수행 과정들은 예를 들어, 양자 컴퓨팅 장치(100)에 의해 수행될 수 있다.

[0044] 본 개시에서 lattice surgery 및 공동 측정(joint measurement)과 관련된 구체적인 방법론들, 정의에 대해서는 논문(Horsman, C., Fowler, A.G., Devitt, S., Van Meter, R.: Surface code quantum computing by lattice surgery. New Journal of Physics 14(12), 123011 (2012))에 기재되어 있으며, 상기 논문은 본 특허 출원에 참조로 통합된다.

[0045] 본 개시에서 물리적 큐비트를 서피스 코드로 인코딩된 논리적 큐비트는 적어도 하나의 양자 게이트(예를 들어, 논리적 단위의 Universal gate set 등)를 통해 사용자가 원하는 임의의 양자 회로를 Fault-tolerant 하게(즉, 오류를 정정하면서) 수행할 수 있다. Universal gate set은 Clifford 연산인 Pauli, CNOT, Hadamard(H), Phase(S) 연산과 Non-Clifford 연산인 Phase(T) 연산을 포함할 수 있다.

[0046] 본 개시에서 양자 회로는 적어도 하나의 큐비트, 적어도 하나의 큐비트에 작용하는 적어도 하나의 양자 게이트 및/또는 측정(measurement)을 포함할 수 있다.

[0047] 본 개시에서 큐비트(예를 들어, 물리적 큐비트, 논리적 큐비트 등)는 양자역학과 관련된 현상을 이용하여 데이터를 처리하기 위한 정보의 기본 단위일 수 있다. 큐비트는 양자 중첩 상태를 이용하여 여러 비트에 대응되는 값을 동시에 표현할 수 있다. 예를 들어, 큐비트는 '20%의 확률로 0, 80%의 확률로 1'과 같이 각각의 값들을 확률로 표현할 수 있다. 큐비트는 측정(관측)될 때 양자 중첩(얽힘) 상태가 풀리면서 하나의 상태로 결정될 수 있다. 예를 들어, 큐비트는 $|0\rangle$ 또는 $|1\rangle$ 의 상태를 나타내는 기저 상태에서 시작하여, 양자 게이트에 의해 상태가 변화될 수 있다. 그리고, 큐비트는 측정(관측)될 때 $|0\rangle$ 또는 $|1\rangle$ 의 상태로 결정될 수 있다. 큐비트는 2차원 벡터로 표현될 수 있다.

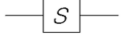
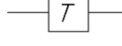
[0048] 본 개시에서 양자 컴퓨팅 장치(100)는 다양한 방법(예를 들어, 양자 게이트를 사용 등)을 이용하여 큐비트를 초기화 하거나 또는 큐비트를 특정 상태로 준비시킬 수 있다.

[0049] 본 개시에서 적어도 하나의 양자 게이트는 양자 컴퓨터에서 큐비트에 적용되는 연산을 나타내는 요소일 수 있다. 적어도 하나의 양자 게이트는 큐비트의 양자 상태를 변환할 수 있다.

[0050] 적어도 하나의 양자 게이트는 복소수 벡터들에 대한 행렬 곱셈을 통해 연산이 이루어질 수 있다. 적어도 하나의 양자 게이트는 Hadamard 게이트, Pauli 게이트, Phase shift 게이트(예를 들어, S 게이트, T 게이트 등), CNOT 게이트 등을 포함할 수 있다.

[0051] Hadamard 게이트는 $|0\rangle$ 또는 $|1\rangle$ 의 상태를 띄고 있는 큐비트의 상태를 $|0\rangle$ 과 $|1\rangle$ 이 동시에 존재하는 중첩 상태로 만들어주는 게이트일 수 있다. 양자 회로에서 Hadamard 게이트의 논리 기호는  일 수 있다. Hadamard 게이트의 행렬식은 $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ 일 수 있다. Hadamard 연산(H)은 X 기저(basis)를 Z 기저로 바꿔주는 연산일 수 있다. 또한, Hadamard 연산(H)은 Z 기저(basis)를 X 기저로 바꿔주는 연산일 수 있다.

[0052] Phase shift 게이트는 큐비트의 위상(phase)을 변경해주는 게이트일 수 있다. Phase shift 게이트는 S 게이트, T 게이트 등을 포함할 수 있다.

- [0053] S 게이트는 큐비트에 90도의 위상 변화를 주는 연산을 수행하는 게이트일 수 있다. 양자 회로에서 S 게이트의 논리 기호는  일 수 있다. S 게이트의 행렬식은 $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ 일 수 있다. i 는 허수 단위일 수 있다.
- [0054] $S^\dagger$ 게이트는 S 게이트에서 행렬의 행과 열을 서로 바꾼 게이트일 수 있다. 양자 회로에서 게이트의 논리 기호는  일 수 있다.
- [0055] T 게이트는 큐비트에 45도의 위상 변화를 주는 연산을 수행하는 게이트일 수 있다. 양자 회로에서 T 게이트의 논리 기호는  일 수 있다. T 게이트의 행렬식은 $\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$ 일 수 있다. i 는 허수 단위일 수 있다. $e^{i\pi/4}$ 는 오일러의 공식(Euler's formula)에 따라 계산되는 복소수로, 복소수 평면 상에서 45도 각도에 해당하는 복소수일 수 있다.
- [0056] CNOT 게이트는 2개의 큐비트를 얽힘 상태로 만드는 게이트일 수 있다.
- [0057] CNOT 게이트는 첫 번째 큐비트(제어 큐비트(control qubit))가 $|1\rangle$ 인 경우, 두 번째 큐비트(타겟 큐비트(target qubit))에 NOT 게이트 연산을 수행하는 게이트일 수 있다. 양자 회로에서 CNOT 게이트의 논리 기호는  일 수 있다. ●는 제어 큐비트, ⊕는 타겟 큐비트일 수 있다. CNOT 게이트의 행렬식은 $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ 일 수 있다.
- [0058] 측정은 큐비트의 상태를 기저 상태 중 하나로 분리하고, 분리된 상태를 통해 측정 결과를 획득하는 과정일 수 있다. 예를 들어, 프로세서(110)는 양자 회로에서 X 또는 Z 기저 상태에서 큐비트의 상태에 대한 측정을 수행함으로써, $|0\rangle$ 또는 $|1\rangle$ 의 상태를 획득할 수 있다. 프로세서(110)는 측정된 큐비트의 상태가 $|0\rangle$ 의 상태인 경우, 고전 비트로 해당 상태를 0으로 결정할 수 있다. 프로세서(110)는 측정된 큐비트의 상태가 $|1\rangle$ 의 상태인 경우, 고전 비트로 해당 상태를 1로 결정할 수 있다. 양자 회로에서 측정의 기호는  일 수 있다.
- [0059] 본 개시에서 기호 d 는 서피스 부호의 거리(distance)를 의미할 수 있다.
- [0060] 본 개시에서 $|0\rangle$ 는 0 상태에 있는 물리적 큐비트일 수 있다.
- [0061] 본 개시에서 $|+\rangle$ 는 + 상태에 있는 물리적 큐비트일 수 있다.
- [0062] 본 개시에서 $|0\rangle_L$ 는 0 상태에 있는 논리적 큐비트일 수 있다.
- [0063] 본 개시에서 $|+\rangle_L$ 는 + 상태에 있는 논리적 큐비트일 수 있다.
- [0064] 본 개시에서 $|Y\rangle_L$ 는 S 연산에 사용되는 매직 상태(magic state)일 수 있다. 예를 들어, $|Y\rangle_L$ 는 $(|0\rangle + i|1\rangle)/\sqrt{2}$ 일 수 있다.
- [0065] 본 개시에서 $|A\rangle_L$ 는 T 연산에 사용되는 매직 상태일 수 있다. 예를 들어, $|A\rangle_L$ 는 $(|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ 일 수 있다.
- [0066] 본 개시에서 시간 비용(time cost)은 측정 오류를 보정하기 위해 연산에 소용되는 서피스 부호 사이클(surface code cycle)의 수를 의미할 수 있다.
- [0067] 본 개시에서 공간 비용(space cost)은 연산 수행에 소요되는 물리적 큐비트의 수를 의미할 수 있다.
- [0068] 본 개시에서 공간-시간 비용(space-time cost)은 공간 비용과 시간 비용의 곱을 의미할 수 있다.
- [0069] 본 개시의 일 실시예에 따르면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 정보가 저장 및 연산되는 데이터 패

치(data patch)와 논리적 연산 과정에서 소모되는 제 1 엔실라 패치(ancilla patch)를 사전 결정된 제 1 간격으로 배치할 수 있다(210).

- [0070] 사전 결정된 제 1 간격은 특정 값으로 한정되지 않으며, 상황에 따라 다양하게 설정될 수 있다.
- [0071] 데이터 패치는 제 1 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 제 1 논리적 큐비트에 대응될 수 있다.
- [0072] 제 1 엔실라 패치는 제 2 물리적 큐비트를 서피스 부호로 인코딩한 제 2 논리적 큐비트에 대응될 수 있다.
- [0073] 제 1 논리적 큐비트 및 제 2 논리적 큐비트 각각은 논리적 정보가 저장되는 데이터 큐비트, 데이터 큐비트의 일측에 형성되는 X 스테빌라이저(stabilizer), 데이터 큐비트의 타측에 형성되는 Z 스테빌라이저(stabilizer) 등을 포함할 수 있다.
- [0074] 스테빌라이저는 양자 비트의 상태를 설명하는 일련의 양자 연산자(에르미트 연산자)의 그룹일 수 있다. 양자 연산자의 그룹은 양자 상태 공간에서 양자 비트의 상태를 나타내는데 사용될 수 있다.
- [0075] X 스테빌라이저는 X 연산자를 포함하는 양자 연산자 그룹을 의미할 수 있다. X 연산자는 큐비트의 상태를 X축을 중심으로 회전시키는 연산자일 수 있다. X 연산자는 큐비트의 상태를 반전시키는 연산을 수행할 수 있다.
- [0076] Z 스테빌라이저는 Z 연산자를 포함하는 양자 연산자 그룹을 의미할 수 있다. Z 연산자는 큐비트의 상태를 Z축으로 회전시키는 연산자일 수 있다. Z 연산자는 큐비트의 상태의 부호를 변경하는 연산을 수행할 수 있다.
- [0077] X 스테빌라이저 및 Z 스테빌라이저는 각각 측정(measurement) 큐비트를 포함할 수 있다.
- [0078] 프로세서(110)는 데이터 패치와 제 1 엔실라 패치 사이에 제 1 논리적 엔실라 큐비트를 배치할 수 있다(220).
- [0079] 프로세서(110)는 제 1 엔실라 패치의 일측과 사전 결정된 거리를 갖도록 제 2 논리적 엔실라 큐비트를 배치할 수 있다(230). 사전 결정된 거리는 특정 값으로 한정되지 않으며, 상황에 따라 다양하게 설정될 수 있다.
- [0080] 본 개시에서 논리적 엔실라 큐비트(예를 들어, 제 1 논리적 엔실라 큐비트, 제 2 논리적 엔실라 큐비트, 제 3 논리적 엔실라 큐비트 등)는 계산에서 특정 목표를 달성하기 위해 추가되는 큐비트일 수 있다. 예를 들어, 논리적 엔실라 큐비트는 논리적 연산을 위해 배치되는 큐비트일 수 있다.
- [0081] 프로세서(110)는 데이터 패치와 제 1 엔실라 패치를 이용하여 공동 측정(joint measurement)을 수행할 수 있다(240).
- [0082] 상술한 바와 같이, 공동 측정(joint measurement)과 관련된 구체적인 방법론들, 정의에 대해서는 논문(Horsman, C., Fowler, A.G., Devitt, S., Van Meter, R.: Surface code quantum computing by lattice surgery. New Journal of Physics 14(12), 123011 (2012))에 기재되어 있으며, 상기 논문은 본 특허 출원에 참조로 통합된다.
- [0083] 일 실시예에서, 프로세서(110)는 데이터 패치와 제 1 엔실라 패치에 공동 ZY 측정(joint ZY measurement)을 수행함으로써, 데이터 패치에 논리적 S 연산을 수행할 수 있다. 예를 들어, 프로세서(110)는 제 1 엔실라 패치를 0 상태로 초기화할 수 있다. 프로세서(110)는 제 1 논리적 엔실라 큐비트를 + 상태로 초기화할 수 있다. 프로세서(110)는 제 2 논리적 엔실라 큐비트를 0 상태로 초기화할 수 있다. 프로세서(110)는 데이터 패치와 제 1 엔실라 패치 사이의 스테빌라이저를 이용하여 데이터 패치와 제 1 엔실라 패치에 공동 ZY 측정(joint ZY measurement)을 수행할 수 있다.
- [0084] 일 실시예에서, 프로세서(110)는 데이터 패치와 제 1 엔실라 패치에 공동 XY 측정(joint XY measurement)을 수행함으로써, 데이터 패치에 논리적 SH 연산을 수행할 수 있다. 예를 들어, 프로세서(110)는 제 1 엔실라 패치를 0 상태로 초기화할 수 있다. 프로세서(110)는 제 2 논리적 엔실라 큐비트를 0 상태로 초기화할 수 있다. 프로세서(110)는 데이터 패치와 제 1 엔실라 패치 사이의 스테빌라이저를 이용하여 데이터 패치와 제 1 엔실라 패치에 공동 XY 측정(joint XY measurement)을 수행할 수 있다.
- [0085] 일 실시예에서, 프로세서(110)는 제 1 엔실라 패치와 중복되는 않는 위치에 데이터 패치(data patch)와 사전 결정된 제 2 간격으로 제 2 엔실라 패치를 배치할 수 있다.
- [0086] 프로세서(110)는 데이터 패치와 제 2 엔실라 패치 사이에 제 3 논리적 엔실라 큐비트를 배치할 수 있다.
- [0087] 프로세서(110)는 데이터 패치, 제 1 엔실라 패치 및 제 2 엔실라 패치에 공동 ZZ & ZY 측정(joint ZZ & ZY

measurement)을 수행함으로써, 데이터 패치에 논리적 T 연산을 수행할 수 있다.

- [0088] 예를 들어, 프로세서(110)는 제 1 엔실라 패치를 0 상태로 초기화할 수 있다. 프로세서(110)는 제 2 엔실라 패치를 사전 결정된 매직 상태(magic state)로 준비할 수 있다. 사전 결정된 매직 상태는 $|A\rangle_L$ 일 수 있다. 프로세서(110)는 제 1 논리적 엔실라 큐비트 및 상기 제 3 논리적 엔실라 큐비트를 + 상태로 초기화할 수 있다. 프로세서(110)는 제 2 논리적 엔실라 큐비트를 0 상태로 초기화할 수 있다. 프로세서(110)는 '데이터 패치와 제 1 엔실라 패치 사이의 스테빌라이저' 및 '데이터 패치와 제 2 엔실라 패치 사이의 스테빌라이저'를 이용하여, '데이터 패치, 제 1 엔실라 패치 및 제 2 엔실라 패치'에 공동 ZZ & ZY 측정(joint ZZ & ZY measurement)을 수행할 수 있다.
- [0089] 이하에서는 앞서 도 2를 참조하여 상술한 구성, 단계 등에 대한 구체적인 설명을 도 3 내지 도 12를 참조하여 후술하도록 한다.
- [0090] 도 3은 본 개시의 일 실시예에 따른 서피스 부호(Surface code)로 인코딩된 논리적 큐비트 및 이에 대응되는 패치를 나타낸 도면이다.
- [0091] 도 3을 참조하면, 임의의 상태인 제 1 논리적 큐비트(310) 및 제 2 논리적 큐비트(330)에서 데이터 큐비트(311)는 논리적 정보가 저장될 수 있다. 제 1 논리적 엔실라 큐비트(350) 및 제 2 논리적 엔실라 큐비트(360)는 lattice surgery 과정에서 소모되는 큐비트일 수 있다. 하나의 데이터 큐비트(311)의 주변에는 X 스테빌라이저(312)와 Z 스테빌라이저(313)가 존재할 수 있다. 각 스테빌라이저에는 측정(measurement) 큐비트가 하나씩 포함될 수 있고, 도 3에는 생략되었다. 도 3은 거리(distance)가 3인 논리적 큐비트로, 논리적 X 또는 Z 연산자가 3개의 큐비트에 XXX 연산 또는 ZZZ 연산으로 정의되고, 굵은 실선으로 표기되어있다. 데이터 패치(320)는 제 1 논리적 큐비트(310)를 패치(patch) 형태로 나타낸 것일 수 있다. 제 1 엔실라 패치(340)는 제 2 논리적 큐비트를 패치 형태로 나타낸 것일 수 있다. 데이터 패치(320)는 사용자가 원하는 정보를 저장 및/또는 연산 중 적어도 하나를 수행할 수 있다. 제 1 엔실라 패치(340)는 논리적 연산 과정에서 소모될 수 있다.
- [0092] 도 4는 본 개시의 일 실시예에 따른 X 스테빌라이저(stabilizer) 및 Z 스테빌라이저의 패리티 측정 회로와 실행 순서를 나타낸 도면이다.
- [0093] 도 4를 참조하면, X 스테빌라이저(410)에서 회로의 수행 순서는 위에서 아래 순일 수 있다. X 스테빌라이저(410)에서 회로의 수행 순서는 X 스테빌라이저(410) 내에 화살표로 표시된 순서일 수 있다. Z 스테빌라이저(430)에서 회로의 수행 순서는 왼쪽에서 오른쪽 순일 수 있다. Z 스테빌라이저(430)에서 회로의 수행 순서는 Z 스테빌라이저(430) 내에 화살표로 표시된 순서일 수 있다.
- [0094] X 스테빌라이저의 패리티 측정 회로(420)와 Z 스테빌라이저의 패리티 측정 회로(440) 각각은 측정 큐비트를 $|0\rangle$ 로 초기화하는 과정까지 포함하여 8 단계(step)안에 수행이 완료될 수 있다. 회로의 동작이 완료되기까지 걸린 시간을 1 서피스 부호 사이클(surface code cycle)로 정의할 수 있다. 스테빌라이저의 패리티는 측정 큐비트의 Z 기저(basis) 측정 결과에 따라 +1 또는 -1의 값을 가질 수 있다. 양자 컴퓨팅 장치(100)는 측정 결과 +1의 값을 가지는 경우, 0 상태로 결정할 수 있다. 양자 컴퓨팅 장치(100)는 측정 결과 -1의 값을 가지는 경우, 1 상태로 결정할 수 있다.
- [0095] 도 5는 본 개시의 일 실시예에 따른 joint ZY measurement 과정 및 4-qubit twist defect의 패리티 측정 회로를 나타낸 도면이다.
- [0096] 도 5를 참조하면, 양자 컴퓨팅 장치(100)는 임의의 상태인 제 1 논리적 큐비트(511) 및 제 2 논리적 큐비트(512)가 있는 경우, 제 1 논리적 엔실라 큐비트(513)를 $|+\rangle$ 로 초기화할 수 있다. 양자 컴퓨팅 장치(100)는 제 2 논리적 엔실라 큐비트(514)를 $|0\rangle$ 로 초기화할 수 있다.
- [0097] 양자 컴퓨팅 장치(100)는 제 1 논리적 큐비트(511)와 제 2 논리적 큐비트(512) 사이에 있는 스테빌라이저를 커쥔(활성화)으로써, 통합 논리적 큐비트(520)를 생성할 수 있다. 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(520)를 이용하여, 제 1 논리적 큐비트(511)와 제 2 논리적 큐비트(512)에 공동 ZY 측정(joint ZY measurement)을 수행할 수 있다. 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZY 측정 결과를 결정하는 스테빌라이저(521)의 패리티를 모두 곱함으로써, 공동 ZY 측정의 결과를 획득할 수 있다. 스테빌라이저(521)의 패리티는 상술한 실행 순서에 대응되는 패리티 측정 회로(540)를 이용하여 획득될 수 있다. 측정 오류를 보완하기 위해 공동 ZY 측정은 총 1d cycle이 소요될 수 있다. 공동 ZY 측정 결과를 결정하는 스테빌

라이저(521)는 사전 결정되어 있을 수 있다.

- [0098] 4-qubit twist defect(530)은 X와 Z, Y가 혼합된 스테빌라이저일 수 있다. 4-qubit twist defect(530)의 측정 회로의 실행 순서는 4-qubit twist defect(530) 내에 화살표로 표시되어 있다.
- [0099] 4-qubit twist defect(530)의 측정 회로의 실행 순서는 주변 X, Z 스테빌라이저의 순서를 고려하여 사전에 결정되어 있을 수 있다.
- [0100] 패리티 측정 회로(540)는 8 단계(step)안에 완료되므로, 종래의 논문(Litinski, Daniel, and Felix von Oppen. "Lattice surgery with a twist: simplifying clifford gates of surface codes." Quantum 2 (2018): 62.)에서 제안한 5-qubit twist defect의 패리티 측정 회로보다 적은 시간이 걸릴 수 있다.
- [0101] 도 6은 본 개시의 일 실시예에 따른 joint XY measurement 과정 및 4-qubit twist defect의 패리티 측정 회로를 나타낸 도면이다.
- [0102] 양자 컴퓨팅 장치(100)는 임의의 상태인 제 1 논리적 큐비트(611) 및 제 2 논리적 큐비트(612)가 있는 경우, 제 2 논리적 엔실라 큐비트(613)를 $|0\rangle$ 로 초기화할 수 있다.
- [0103] 양자 컴퓨팅 장치(100)는 제 1 논리적 큐비트(611)와 제 2 논리적 큐비트(612) 사이에 있는 스테빌라이저를 커짐(활성화)으로써, 통합 논리적 큐비트(620)를 생성할 수 있다. 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(620)를 이용하여, 제 1 논리적 큐비트(611)와 제 2 논리적 큐비트(612)에 공동 XY 측정(joint XY measurement)을 수행할 수 있다. 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 XY 측정 결과를 결정하는 스테빌라이저(621)의 패리티를 모두 곱함으로써, 공동 XY 측정의 결과를 획득할 수 있다. 스테빌라이저(621)의 패리티는 상술한 실행 순서에 대응되는 패리티 측정 회로(640)를 이용하여 획득될 수 있다. 측정 오류를 보완하기 위해 공동 XY 측정은 총 1d cycle이 소요될 수 있다. 공동 XY 측정 결과를 결정하는 스테빌라이저(621)는 사전 결정되어 있을 수 있다.
- [0104] 4-qubit twist defect(630)은 X와 Z, Y가 혼합된 스테빌라이저일 수 있다. 4-qubit twist defect(630)의 측정 회로의 실행 순서는 4-qubit twist defect(630) 내에 화살표로 표시되어 있다. 4-qubit twist defect(630)의 측정 회로의 실행 순서는 주변 X, Z 스테빌라이저의 순서를 고려하여 사전에 결정되어 있을 수 있다.
- [0105] 패리티 측정 회로(640)는 8 단계(step)안에 완료되므로, 종래의 논문(Litinski, Daniel, and Felix von Oppen. "Lattice surgery with a twist: simplifying clifford gates of surface codes." Quantum 2 (2018): 62.)에서 제안한 5-qubit twist defect의 패리티 측정 회로보다 적은 시간이 걸릴 수 있다.
- [0106] 도 7은 본 개시의 일 실시예에 따른 4-qubit twist defect에서 2개의 측정(measurement) 큐비트를 활용한 패리티 측정 회로를 나타낸 도면이다.
- [0107] 도 7은 도 6의 4-qubit twist defect에서 물리적 CNOT의 거리가 멀 경우, 2개의 측정 큐비트를 활용하여 패리티 측정 회로를 수행하는 방법을 나타낸 것일 수 있다. 패리티 측정 회로(720)에서 상단에 있는 회로는 4-qubit twist defect(710)에서 왼쪽 위에 있는 두 큐비트(X_2 , Y_3)에 대한 회로일 수 있다. 하단에 있는 회로는 4-qubit twist defect(710)에서 하단(큐비트(Y_3))보다 아래에 있는 두 큐비트(X_2 , X_3)에 대한 회로이다. 패리티는 상단에 있는 회로의 측정 큐비트를 통해 획득할 수 있다.
- [0108] 도 8은 본 개시의 일 실시예에 따른 joint ZZ & ZY measurement 과정을 나타낸 도면이다.
- [0109] 공동 ZZ 측정(joint ZZ measurement)과 공동 ZY 측정(joint ZY measurement)은 서로 대체가능한(commutable) 동작이므로 동시에 수행할 수 있다.
- [0110] 양자 컴퓨팅 장치(100)는 임의의 상태인 세 개의 논리적 큐비트(811, 812, 813)를 위에서부터 공동 ZZ & ZY 측정을 할 경우, 논리적 큐비트들 간에 존재하는 스테빌라이저를 커짐(활성화)으로써, 통합 논리적 큐비트(820)를 생성할 수 있다. 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(820)를 이용하여, 세 개의 논리적 큐비트(811, 812, 813)에 공동 ZZ & ZY 측정을 수행할 수 있다.
- [0111] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZZ 측정 결과를 결정하는 스테빌라이저(821)의 패리티를 모두 곱함으로써, 공동 ZZ 측정의 결과를 획득할 수 있다. 공동 ZZ 측정 결과를 결정하는 스테빌라이저(821)는 사전 결정되어 있을 수 있다.

- [0112] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZY 측정 결과를 결정하는 스테빌라이저(822)의 패리티를 모두 곱함으로써, 공동 ZY 측정의 결과를 획득할 수 있다. 공동 ZY 측정 결과를 결정하는 스테빌라이저(822)는 사전 결정되어 있을 수 있다.
- [0113] 측정 오류를 보완하기 위해 반복 수행되는 과정에서 총 1d cycle이 소요될 수 있다.
- [0114] 도 9는 본 개시의 일 실시예에 따른 논리적 S 연산 수행 과정을 나타낸 도면이다.
- [0115] 도 9를 참조하면, 데이터 패치(911)는 논리적 S 연산을 수행하고자 하는 임의의 상태 $|\psi\rangle_L$ 일 수 있다. 제 1 엔실라 패치(912)는 $|0\rangle_L$ 일 수 있다. 또는, 제 1 엔실라 패치(912)는 $|0\rangle_L$ 로 준비될 수 있다(초기화될 수 있다).
- [0116] 양자 컴퓨팅 장치(100)는 데이터 패치(911)에 대응되는 제 1 논리적 큐비트(921)와 제 1 엔실라 패치(912)에 대응되는 제 2 논리적 큐비트(922) 사이에 있는 스테빌라이저를 켜줌(활성화)으로써, 통합 논리적 큐비트(930)를 생성할 수 있다.
- [0117] 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(930)를 이용하여, 제 1 논리적 큐비트(921)와 제 2 논리적 큐비트(922)에 공동 ZY 측정을 수행할 수 있다.
- [0118] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZY 측정 결과를 결정하는 스테빌라이저(931)의 패리티를 모두 곱함으로써, 공동 ZY 측정의 결과를 획득할 수 있다.
- [0119] 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(940)의 제 1 영역(941)에 존재하는 큐비트는 Z 기저로 측정하고, 제 2 영역(942)에 존재하는 큐비트는 X 기저로 측정할 수 있다. 측정이 완료된 후, 제 1 엔실라 패치(912)에 대응되는 제 2 논리적 큐비트(922)는 연산 과정에서 소모되고, 연산 결과에 대응되는 논리적 큐비트(950)만 존재할 수 있다.
- [0120] 상술한 본 개시의 일 실시예에 따른 논리적 S 연산 수행 과정은 총 1d cycle이 소요될 수 있다.
- [0121] 도 10은 본 개시의 일 실시예에 따른 논리적 S 연산 수행 시, 스테빌라이저의 변화를 나타낸 도면이다.
- [0122] 도 10을 참조하면, 양자 컴퓨팅 장치(100)는 임의의 상태인 제 1 논리적 큐비트(1010) 및 제 2 논리적 큐비트(1020)가 있는 경우, 제 1 논리적 엔실라 큐비트(1030)를 $|+\rangle$ 로 초기화할 수 있다. 양자 컴퓨팅 장치(100)는 제 2 논리적 엔실라 큐비트(1040)를 $|0\rangle$ 로 초기화할 수 있다.
- [0123] 양자 컴퓨팅 장치(100)는 제 1 논리적 큐비트(1010)와 제 2 논리적 큐비트(1020) 사이에 있는 스테빌라이저를 켜줌(활성화)으로써, 통합 논리적 큐비트(1050)를 생성할 수 있다. 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1050)를 이용하여, 제 1 논리적 큐비트(1010)와 제 2 논리적 큐비트(1020)에 공동 ZY 측정을 수행할 수 있다.
- [0124] 논리적 S 연산은 X 연산을 Y로 바꾸고, Z 연산을 Z로 바꾸는 특성을 지니며, 도 10에 명시된 스테빌라이저를 새롭게 켜올 때 해당 특성 보이는 것을 확인할 수 있다.
- [0125] 도 11은 본 개시의 일 실시예에 따른 논리적 SH 연산 수행 과정을 나타낸 도면이다.
- [0126] 도 11을 참조하면, 데이터 패치(1111)는 논리적 SH 연산을 수행하고자 하는 임의의 상태 $|\psi\rangle_L$ 일 수 있다. 제 1 엔실라 패치(1112)는 $|0\rangle_L$ 일 수 있다. 또는, 제 1 엔실라 패치(1112)는 $|0\rangle_L$ 로 준비될 수 있다(초기화될 수 있다).
- [0127] 양자 컴퓨팅 장치(100)는 데이터 패치(1111)에 대응되는 제 1 논리적 큐비트(1121)와 제 1 엔실라 패치(1112)에 대응되는 제 2 논리적 큐비트(1122) 사이에 있는 스테빌라이저를 켜줌(활성화)으로써, 통합 논리적 큐비트(1130)를 생성할 수 있다.
- [0128] 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1130)를 이용하여, 제 1 논리적 큐비트(921)와 제 2 논리적 큐비트(922)에 공동 XY 측정을 수행할 수 있다.
- [0129] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 XY 측정 결과를 결정하는 스테빌라이저

(1131)의 패리티를 모두 곱함으로써, 공동 XY 측정의 결과를 획득할 수 있다.

- [0130] 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1140)의 제 1 영역(1141)에 존재하는 큐비트는 Z 기저로 측정하고, 제 2 영역(1142)에 존재하는 큐비트는 X 기저로 측정할 수 있다. 측정이 완료된 후, 제 1 엔실라 패치(1112)에 대응되는 제 2 논리적 큐비트(1122)는 연산 과정에서 소모되고, 연산 결과에 대응되는 논리적 큐비트(1150)만 존재할 수 있다.
- [0131] 상술한 본 개시의 일 실시예에 따른 논리적 SH 연산 수행 과정은 총 1d cycle이 소요될 수 있다.
- [0132] 도 12는 본 개시의 일 실시예에 따른 논리적 T 연산 수행 과정을 나타낸 도면이다.
- [0133] 양자 컴퓨팅 장치(100)는 데이터 패치(1211)는 논리적 T 연산을 수행하고자 하는 임의의 상태 $|\psi\rangle_L$ 일 수 있다. 제 1 엔실라 패치(1212)는 $|0\rangle_L$ 일 수 있다. 또는, 제 1 엔실라 패치(1212)는 $|0\rangle_L$ 로 준비될 수 있다(초기화될 수 있다). 제 2 엔실라 패치(1213)는 $|A\rangle_L$ 로 준비될 수 있다.
- [0134] 양자 컴퓨팅 장치(100)는 '데이터 패치(1211)와 제 1 엔실라 패치(1212) 사이에 존재하는 스테빌라이저' 및 '데이터 패치(1211)와 제 2 엔실라 패치(1213) 사이에 존재하는 스테빌라이저'를 커짐(활성화)으로써, 통합 논리적 큐비트(1230)를 생성할 수 있다. 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1230)를 이용하여, 세 개의 논리적 큐비트(1211, 1212, 1213)에 공동 ZZ & ZY 측정을 수행할 수 있다.
- [0135] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZZ 측정 결과를 결정하는 스테빌라이저(1231)의 패리티를 모두 곱함으로써, 공동 ZZ 측정의 결과를 획득할 수 있다. 공동 ZZ 측정 결과를 결정하는 스테빌라이저(1231)는 사전 결정되어 있을 수 있다.
- [0136] 양자 컴퓨팅 장치(100)는 새롭게 동작하는 스테빌라이저 중에서 공동 ZY 측정 결과를 결정하는 스테빌라이저(1232)의 패리티를 모두 곱함으로써, 공동 ZY 측정의 결과를 획득할 수 있다. 공동 ZY 측정 결과를 결정하는 스테빌라이저(1232)는 사전 결정되어 있을 수 있다.
- [0137] 양자 컴퓨팅 장치(100)는 공동 ZZ 측정 결과, +1 인 경우, $|0\rangle_L$ 인 제 1 엔실라 패치(1212)를 Z 기저로 측정할 수 있다. 따라서, 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1240)의 제 1 영역(1241)에 존재하는 큐비트는 Z 기저로 측정하고, 통합 논리적 큐비트(1240)의 제 2 영역(1242)에 존재하는 큐비트는 X 기저로 측정할 수 있다.
- [0138] 양자 컴퓨팅 장치(100)는 공동 ZZ 측정 결과, -1 인 경우, $|0\rangle_L$ 인 제 1 엔실라 패치(1212)를 X 기저로 측정할 수 있다. 따라서, 양자 컴퓨팅 장치(100)는 통합 논리적 큐비트(1250)의 제 1 영역(1251)에 존재하는 큐비트는 Z 기저로 측정하고, 통합 논리적 큐비트(1250)의 제 2 영역(1252)에 존재하는 큐비트는 X 기저로 측정할 수 있다.
- [0139] 상술한 본 개시의 일 실시예에 따른 논리적 T 연산 수행 과정은 총 1d cycle이 소요될 수 있다.
- [0140] 이하에서는 종래에 제안된 논리적 S 연산 및 T 연산의 구현 방법과 본 개시의 일 실시예에 따른 논리적 S 연산 및 T연산의 구현 방법을 비교한 결과를 후술한다.
- [0141] 아래의 표 1은 종래에 제안된 논리적 S 연산 방법과 본 개시의 일 실시예에 따른 논리적 S 연산의 시간 비용, 공간 비용, 공간-시간 비용을 비교한 표이다.

표 1

Method	Distilled $ Y\rangle_L$	시간 비용	공간 비용 (Data+ ancilla patches)	공간-시간 비용 (논리적 큐비트 당)	Reference
논리적 Hadamard, CNOT, 사용	O	$14d$	$2 \times 1d^2$	$28d^3$	Fowler, 2012
$ Y\rangle_L$ 와 ZZ measurement	O	$1d$	$2 \times 1d^2$	$2d^3$	Haah, 2021
$ 0\rangle_L$ 와 5-qubit defect를 사용하여 ZY measurement	X	$1d$	$2 \times (2d^2 - 2d)$	$4d^3 - 4d^2$	Litinski, 2019
$ 0\rangle_L$ 와 4-qubit defect를 사용하여 ZY measurement	X	$1d$	$2 \times (1d^2 + 2d)$	$2d^3 + 4d^2$	본 개시에서 제안

[0142]

[0143]

표 1을 참조하면, 2012년 Fowler가 제안한 방법(Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. "Surface codes: Towards practical large-scale quantum computation." Physical Review A 86.3 (2012): 032324.)은 논리적 Hadamard, CNOT 연산을 사용하여 논리적 S를 수행하는 경우로, 매직 상태인 $|Y\rangle_L$ 를 요구한다. 논리적 Hadamard 와 CNOT 연산에 많은 비용이 소요되므로 시간 및 공간 비용이 증가한다. 표에서는 고려하지 않았지만 매직 상태인 $|Y\rangle_L$ 를 준비하는 Magic state distillation 과정에도 별도의 비용이 요구된다.

[0144]

2021년 Haah의 연구에 소개된 논리적 S 연산 수행 방법(Haah, Jeongwan, and Matthew B. Hastings. "Measurement sequences for magic state distillation." Quantum 5 (2021): 383.)은 joint ZZ measurement와 매직 상태인 $|Y\rangle_L$ 를 이용하였다. 해당 기법은 적은 시간 및 공간 비용을 요구하지만 이는 매직 상태인 $|Y\rangle_L$ 를 준비하는데 소요되는 비용은 제외한 것이므로, Magic state distillation 과정에도 별도의 비용이 요구된다.

[0145]

2019년 Litinski의 연구에서 제시된 논리적 S 연산 수행 방법(Litinski, Daniel. "A game of surface codes: Large-scale quantum computing with lattice surgery." Quantum 3 (2019): 128.)은 5-qubit twist defect를 사용하여 two-patch wide의 논리적 큐비트에서 joint ZY measurement를 사용하는 방식이다. 해당 기법은 매직 상태인 $|Y\rangle_L$ 를 요구하지 않으나, two-patch wide의 논리적 큐비트가 사용되므로 공간 비용이 증가하였다.

[0146]

아래의 표 2는 종래에 제안된 논리적 T 연산 방법과 본 개시의 일 실시예에 따른 논리적 T 연산의 시간 비용, 공간 비용, 공간-시간 비용을 비교한 표이다.

표 2

Method	Distilled $ Y\rangle_L$	시간 비용	공간 비용 (Data+ ancilla patches)	공간-시간 비용 (논리적 큐비트 당)	Reference
논리적 CNOT과 S 사용	O	$17d$	$3 \times 1d^2$	$51d^3$	Zhou, 2000
$ A\rangle_L$ 와 ZZ measurement 및 S 사용	O	$2d$	$3 \times 1d^2$	$6d^3$	Bravyi, 2016, Fowler, 2018
$ A\rangle_L$ 및 $ 0\rangle_L$ 와 5-qubit defect를 사용하여 ZZ & ZY measurement	X	$1d$	$2 \times (2d^2 - 2d) + 1d^2$	$5d^3 - 4d^2$	Litinski, 2019
$ A\rangle_L$ 및 $ 0\rangle_L$ 와 4-qubit defect를 사용하여 ZZ & ZY measurement	X	$1d$	$3 \times (1d^2 + 2d)$	$3d^3 + 6d^2$	본 개시에서 제안

[0147]

- [0148] 표 2를 참조하면, 2000년 Zhou의 연구에서 제안한 논리적 T 연산 수행 방법(Zhou, Xinlan, Debbie W. Leung, and Isaac L. Chuang. "Methodology for quantum logic gate construction." Physical Review A 62.5 (2000): 052316.)은 논리적 CNOT 및 S 연산을 사용한 teleportation 회로이다. 해당 기법은 논리적 S 연산을 요구하므로 많은 시간 및 공간 비용이 요구된다.
- [0149] 2016년 Bravyi의 연구(Bravyi, Sergey, Graeme Smith, and John A. Smolin. "Trading classical and quantum computational resources." Physical Review X 6.2 (2016): 021043.)와 2018년 Fowler의 연구(Fowler, Austin G., and Craig Gidney. "Low overhead quantum computation using lattice surgery." arXiv preprint arXiv:1808.06709 (2018).)에서 매직 상태인 $|A\rangle_L$ 와 joint ZZ measurement를 이용하여 논리적 T 연산 수행방법을 소개하고 있다. 해당 기법은 논리적 CNOT을 사용한 방법보다는 적은 비용을 사용하였지만, 논리적 S 연산을 사용하기 때문에 2d의 시간 비용이 요구된다.
- [0150] 2019년 Litinski의 연구(Litinski, Daniel. "A game of surface codes: Large-scale quantum computing with lattice surgery." Quantum 3 (2019): 128.)에서 제시된 논리적 T 연산 수행 방법은 5-qubit twist defect를 사용하여 two-patch wide의 논리적 큐비트에서 joint ZZ & ZY measurement를 사용하는 방식이다. 해당 기법은 매직 상태인 $|A\rangle_L$ 를 요구하지 않으나, two-patch wide의 논리적 큐비트가 사용되므로 공간 비용이 증가하였다.
- [0151] 도 1 내지 도 12를 참조하여 상술한 바와 같이, 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 논리적 S와 T 연산을 one-patch wide의 논리적 큐비트만으로 수행이 가능하므로, two-patch wide의 논리적 큐비트가 필요한 기존 방식보다 적은 공간 비용을 요구할 수 있다.
- [0152] 또한, 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 논리적 큐비트의 layout을 one-patch wide의 논리적 큐비트로 구성할 수 있기 때문에 checkerboard-type, row-type 등의 layout에 맞춰서 논리적 큐비트를 배치하기 적합할 수 있다.
- [0153] two-patch wide에서 사용하는 5-qubit twist defect는 기존의 X, Z 스테빌라이저보다 패리티 측정 회로의 시간이 더 길게 소요된다. 따라서 1 surface code cycle에 더 많은 시간이 걸리게 되고 이는 전체 회로의 소요시간을 증가시킨다. 반면 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)에서 사용되는 4-qubit twist defect의 패리티 측정 회로(parity measurement circuit)는 기존의 X, Z 스테빌라이저와 동일한 시간 안에 수행이 가능하므로 1 surface code cycle을 증가시키지 않을 수 있다.
- [0154] 한편, 도 13은 본 개시의 일 실시예에 따른 경계 반전과 후속 측정의 변형을 사용한 논리적 Hadamard 연산 구현 방법을 나타낸 도면이다.
- [0155] 도 13에 도시되는 단계들은 예시적인 단계들이다. 따라서, 본 개시 내용의 사상 범위를 벗어나지 않는 한도에서 도 13의 단계들 중 일부가 생략되거나 추가적인 단계들이 존재할 수 있다는 점 또한 당업자에게 명백할 것이다. 또한, 도 13에 기재된 구성들(예를 들어, 양자 컴퓨팅 장치(100) 등)에 관한 구체적인 내용은 앞서 도 1 내지 도 12를 통해 설명한 내용으로 대체될 수 있다. 도 13에 도시되는 단계들은 예를 들어, 양자 컴퓨팅 장치(100)에 의해 수행될 수 있다.
- [0156] 도 13을 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 물리적 큐비트를 서피스 부호(surface code)로 인코딩한 논리적 큐비트를 획득할 수 있다(1310).
- [0157] 프로세서(110)는 논리적 큐비트에 논리적 Hadamard 연산을 수행하고 그리고 제 1 공동 측정(joint measurement)을 사용하는 제 1 연산을 수행하는 경우, 논리적 큐비트에 포함된 X 경계(boundary)를 Z 경계로 교체하고 그리고 Z 경계를 X 경계로 교체하는 경계 반전(boundary flip)을 수행할 수 있다(1320).
- [0158] X 경계는 논리적 큐비트에 포함된 X 스테빌라이저(stabilizer)로 구성된 경계일 수 있다.
- [0159] Z 경계는 논리적 큐비트에 포함된 Z 스테빌라이저로 구성된 경계일 수 있다.
- [0160] 프로세서(110)는 경계 반전된 논리적 큐비트에 제 1 공동 측정이 아닌, 제 2 공동 측정을 사용하여 제 1 연산을 수행할 수 있다(1330).
- [0161] 프로세서(110)는 논리적 Hadamard 연산 이후에 후속되는 연산의 측정이 공동 XX 측정(joint XX measurement)일 경우, 아래의 수학적 1과 같이 변형할 수 있다.

수학식 1

$$(H \otimes I)^\dagger (X \otimes X) (H \otimes I) = Z \otimes X \rightarrow (X \otimes X) (H \otimes I) = (H \otimes I) (Z \otimes X)$$

프로세서(110)는 논리적 Hadamard 연산 이후에 후속되는 연산의 측정이 공동 ZZ 측정(joint ZZ measurement)일 경우, 아래의 수학식 2와 같이 변형할 수 있다.

수학식 2

$$(H \otimes I)^\dagger (Z \otimes Z) (H \otimes I) = X \otimes Z \rightarrow (Z \otimes Z) (H \otimes I) = (H \otimes I) (X \otimes Z)$$

수학식 1 및 수학식 2에서 화살표는 화살표의 왼쪽의 수식을 화살표의 오른쪽의 수식으로 변화시킬 수 있다는 것을 의미할 수 있다. $\otimes$ 는 tensor product이고, † 는 Conjugate-transpose일 수 있다. 수학식 1 및 수학식 2를 참조하면, Hadamard 연산(H)와 Pauli 연산(X,Z), Identity 연산(I) 간에는 화살표 왼쪽의 관계가 성립하고, 이를 화살표 오른쪽과 같이 변형하면 H 연산 이후 X 연산을 수행하는 것은 Z 연산 이후 H 연산을 수행하는 것과 같음을 알 수 있다.

일 실시예에서, 프로세서(110)는 제 1 연산이 논리적 CNOT인 경우, 경계 반전된 논리적 큐비트에 상기 제 1 공동 측정인 공동 ZZ 측정(joint ZZ measurement)이 아닌, 제 2 공동 측정인 공동 XZ 측정(joint XZ measurement)을 수행할 수 있다.

일 실시예에서, 프로세서(110)는 제 1 연산이 패치 확장(patch extension)인 경우, 경계 반전된 논리적 큐비트에 제 1 공동 측정인 공동 XX 측정(joint XX measurement)이 아닌, 제 2 공동 측정인 공동 ZX 측정(joint ZX measurement)을 수행할 수 있다. 패치 확장은 임의의 ψ (psi) 상태인 하나의 패치가 + 상태인 패치와 공동 ZZ 측정을 하거나 0 상태인 패치와 공동 XX 측정을 하는 것을 의미할 수 있다. 병합된 패치(patch)는 여전히 ψ (psi) 상태이므로 단순히 패치가 늘어난 것과 같은 효과를 가질 수 있다.

일 실시예에서, 프로세서(110)는 제 1 연산이 논리적 S 연산인 경우, 경계 반전된 논리적 큐비트에 제 1 공동 측정인 공동 ZY 측정(joint ZY measurement)이 아닌, 제 2 공동 측정인 공동 XY 측정(joint XY measurement)을 수행할 수 있다.

일 실시예에서, 프로세서(110)는 제 1 연산이 논리적 T 연산인 경우, 경계 반전된 논리적 큐비트에 제 1 공동 측정인 공동 ZZ 및 ZY 측정(joint ZZ & ZY measurement)이 아닌, 제 2 공동 측정인 공동 ZX 및 XY 측정(joint XY measurement)을 수행할 수 있다.

프로세서(110)는 제 2 공동 측정을 사용하여 상기 제 1 연산을 수행하는 단계 이후로, 논리적 Hadamard 연산의 순서를 변경할 수 있다(1340).

일 실시예에서, 프로세서(110)는 변경된 논리적 Hadamard 연산의 순서에 후속하여 측정이 아닌 다른 연산이 존재하는 경우, 앞선 단계(1320, 1330)를 반복하여 수행할 수 있다. 예를 들어, 프로세서(110)는 변경된 논리적 Hadamard 연산의 순서에 후속하여 제 3 공동 측정을 사용하는 제 2 연산이 존재하는 경우, 제 1 연산이 수행된 논리적 큐비트에 경계 반전을 수행할 수 있다. 프로세서(110)는 경계 반전 및 제 1 연산이 수행된 논리적 큐비트에 제 3 공동 측정이 아닌, 제 4 공동 측정을 사용하여 제 2 연산을 수행할 수 있다. 프로세서(110)는 제 4 공동 측정을 사용하여 제 2 연산을 수행하는 단계 이후로, 논리적 Hadamard 연산의 순서를 변경할 수 있다.

일 실시예에서, 프로세서(110)는 변경된 논리적 Hadamard 연산의 순서에 후속하여 단일 측정(single measurement)이 존재하는 경우, 논리적 Hadamard 연산과 단일 측정을 병합함으로써, 단일 측정에서 제 1 연산이 수행된 논리적 큐비트를 측정하는 기저(basis)를 변경할 수 있다.

예를 들어, 프로세서(110)는 단일 측정에서 측정하는 기저가 X 기저인 경우, 논리적 Hadamard 연산과 단일 측정을 병합함으로써, 단일 측정에서 측정하는 기저를 Z 기저로 변경할 수 있다.

다른 예를 들어, 프로세서(110)는 단일 측정에서 측정하는 기저가 Z 기저인 경우, 논리적 Hadamard 연산과 단일

측정을 병합함으로써, 단일 측정에서 측정하는 기저를 X 기저로 변경할 수 있다.

[0175] 이하에서는 앞서 도 13를 참조하여 상술한 구성, 단계 등에 대한 구체적인 설명을 도 14 내지 도 19를 참조하여 후술하도록 한다.

[0176] 도 14는 본 개시의 일 실시예에서 경계 반전에 따른 물리적 CNOT 실행 순서를 나타낸 도면이다.

[0177] 도 14를 참조하면, 'd=3'인 서피스 코드로 인코딩된 하나의 논리적 큐비트(1401)에서 모든 원은 하나의 물리적 큐비트를 의미할 수 있다. X 스테빌라이저(1403)와 Z 스테빌라이저(1404) 내에서 화살표는 스테빌라이저의 패리티 측정 회로(parity measurement circuit)에서 물리적 CNOT 연산이 실행되는 순서를 의미할 수 있다. X 스테빌라이저(1403)는 위에서 아래 순으로 CNOT을 실행하며, Z 스테빌라이저(1404)는 왼쪽에서 오른쪽 순으로 실행할 수 있다. 경계 반전을 수행하기 이전의 노말 사이클(normal cycle)(1405)에서, 논리적 큐비트(logical qubit)의 위아래에는 X 스테빌라이저(1403)로 구성된 X 경계가 존재하고, 좌우에는 Z 스테빌라이저(1404)로 구성된 Z 경계가 존재할 수 있다. 프로세서(110)는 경계 반전을 수행한 경우, 1 서피스 코드 사이클(surface code cycle)(1406)이 끝나면 경계에 있는 스테빌라이저가 동시에 모두 바뀌게 되므로 경계는 90도 회전한 것과 같이 변함으로써, 경계 반전된 논리적 큐비트(1402)를 획득할 수 있다. 마찬가지로 논리적 X(logical X)와 논리적 Z도 90도 회전할 수 있다. 프로세서(110)는 해당 동작을 1d 번 반복하여 측정 오류를 보정할 수 있다. 동작 과정에서, 스테빌라이저는 동시에 바뀌고 거리(distance)와 자유도(degree of freedom)는 동일하게 유지되기 때문에 원래 정보를 훼손하지 않을 수 있다.

[0178] 프로세서(110)는 새로 켜진 스테빌라이저의 패리티 측정(Parity measurement) 결과에 따라 논리적 연산(logical operator)을 보정할 수 있다.

[0179] 도 15a는 본 개시의 일 실시예에서 경계 반전 이전에 스테빌라이저, 논리적 X, Z를 나타낸 도면이다. 도 15b는 본 개시의 일 실시예에서 경계 반전을 수행 중인 스테빌라이저, 논리적 X, Z를 나타낸 도면이다. 도 15c는 본 개시의 일 실시예에서 경계 반전을 수행한 이후의 스테빌라이저, 논리적 X, Z를 나타낸 도면이다. 도 15d는 본 개시의 일 실시예에서 경계 반전을 수행한 이후, 새로 활성화된 스테빌라이저를 곱하여 새로운 논리적 연산을 정의하는 과정을 나타낸 도면이다.

[0180] 도 15a 내지 도 15d를 참조하면, 도 15a의 논리적 큐비트(1510)는 경계 반전 이전에 스테빌라이저, 논리적 X, Z를 나타낸 큐비트일 수 있다. $K_1 \sim K_8$ 은 스테빌라이저이며, Z_L 과 X_L 은 논리적 연산(logical operator)일 수 있다. 논리적 연산에 스테빌라이저를 곱하더라도 기존의 논리적 연산과 동일할 수 있다. 따라서 도 15b의 논리적 큐비트(1520)에서 Z_L 과 X_L 에 K_7 과 K_6 을 곱한 K_7Z_L 과 K_6X_L 로 변형해도 동일할 수 있다. 도 15c의 논리적 큐비트(1530)에서 $M_9 \sim M_{12}$ 는 $K_9 \sim K_{12}$ 의 패리티 측정 결과를 나타낸 것일 수 있다. 프로세서(110)는 도 15d의 논리적 큐비트(1540)에서 K_7Z_L 에 새로 활성화된 스테빌라이저인 K_{10} 을 곱하여 새로운 논리적 연산을 정의할 수 있다. 마찬가지로, 프로세서(110)는 K_6X_L 에 K_{12} 를 곱하여 새로운 논리적 연산을 정의할 수 있다. 따라서 경계 반전 이후 논리적 연산은 $Z'_L = K_{10}K_7Z_L$, $X'_L = K_{12}K_6X_L$ 이고 새로 활성화된 K_{10} 과 K_{12} 의 패리티 측정 결과에 따라 보정이 이뤄질 수 있다. 즉, 프로세서(110)는 새로 활성화된 K_{10} 과 K_{12} 의 패리티 측정 결과에 기초하여, 논리적 연산인 $Z'_L = K_{10}K_7Z_L$ 및 $X'_L = K_{12}K_6X_L$ 에 대한 보정을 수행할 수 있다.

[0181] 상술한 바와 같이, 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 추가적인 공간 비용을 사용하지 않으면서, 1d의 시간 비용만 소요하여 경계의 변형을 수행할 수 있다.

[0182] 도 16은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 CNOT 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

[0183] 도 16을 참조하면, 양자 회로(1610)은 임의의 상태 $|\psi_1\rangle_L$ 에 논리적 Hadamard 연산 수행 후, 논리적 CNOT의 control인 경우를 나타낸 것일 수 있다. 논리적 CNOT 연산은 공동 ZZ 측정(joint ZZ measurement)과 공동 XX 측정(joint XX measurement)으로 분해될 수 있다. 논리적 CNOT의 control 큐비트에 수행될 논리적 Hadamard 연산은 joint 공동 ZZ 측정을 공동 XZ 측정(joint XZ measurement)으로 변화시키면서 논리적 CNOT 이후로 이동할 수 있다. 프로세서(110)는 공동 XZ 측정(joint XZ measurement)을 수행하는 과정(1620)을 진행할 수 있다. 예를 들어, 프로세서(110)는 $|\psi_2\rangle_L$ 와 마주보는 경계(boundary)를 X 경계로 바꾸기 위해 $|\psi_1\rangle_L$ 에 경계 반전

(boundary flip)을 수행할 수 있다. 프로세서(110)는 X와 Z가 혼합된 혼합 스테빌라이저(mixed stabilizer)를 새롭게 꺼줄 수 있다. 프로세서(110)는 검은색 점으로 표기된 스테빌라이저의 패리티를 곱함으로써, 공동 XZ 측정 결과를 획득할 수 있다. X와 Z가 혼합된 혼합 스테빌라이저의 패리티 측정 회로에서 물리적 CNOT의 실행 순서는 큐비트(1630) 내에 화살표 및 숫자로 제시되었다. 해당 순서는 도 14의 X 스테빌라이저(1403) 및 Z 스테빌라이저(1404)에 명시된 순서를 기반으로 주변 스테빌라이저의 물리적 CNOT 실행 순서와 겹치지 않도록 사전에 결정되어 있을 수 있다. target 큐비트에 논리적 Hadamard를 수행한 경우, 공동 XX 측정 대신 공동 XZ 측정을 수행할 수 있다. 공동 XZ 측정을 수행하는 과정은 앞선 설명과 대응되므로 생략한다.

[0184] 도 17은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 S 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

[0185] 도 17을 참조하면, 논리적 S 연산은 $|0\rangle_L$ 로 준비된 엔실라 패치(ancilla patch)와 공동 ZY 측정(joint ZY measurement)을 수행한 것일 수 있다. 양자 회로(1700)에 도시된 바와 같이, 프로세서(110)는 공동 ZY 측정(joint ZY measurement)을 공동 XY 측정(joint XY measurement)으로 변화시키면서, 임의의 $|\psi\rangle_L$ 에 적용될 논리적 Hadamard 연산을 논리적 S 연산 이후로 이동시킬 수 있다.

[0186] 도 18은 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 논리적 T 연산인 경우 수행되는 공동 측정을 수행하는 과정을 나타낸 도면이다.

[0187] 도 18을 참조하면, 논리적 T 연산은 매직 상태(magic state)인 $|A\rangle_L$ 및 $|0\rangle_L$ 로 준비된 엔실라 패치(ancilla patch) 2개와 공동 ZZ 및 ZY 측정(joint ZZ & ZY measurement)을 수행한 것일 수 있다. 양자 회로(1800)에 도시된 바와 같이, 프로세서(110)는 공동 ZZ 측정(joint ZZ measurement)을 공동 ZX 측정(joint ZX measurement)으로 변화시키고, 공동 ZY 측정(joint ZY measurement)을 공동 XY 측정(joint XY measurement)로 변화시키면서, 임의의 $|\psi\rangle_L$ 에 적용될 논리적 Hadamard 연산을 논리적 T 이후로 이동시킬 수 있다. 공동 ZX 측정과 공동 XY 측정은 서로 대체가능한(commutable) 동작이므로 동시에 수행할 수 있다.

[0188] 도 19는 본 개시의 일 실시예에서 논리적 Hadamard 연산에 후속하는 연산이 단일 측정(single measurement)인 경우 수행되는 측정을 나타낸 도면이다.

[0189] 도 19를 참조하면, 양자 회로(1910)에 도시된 바와 같이, 프로세서(110)는 단일 측정에서 측정하는 기저가 Z 기저인 경우, 논리적 Hadamard 연산과 단일 측정을 병합함으로써, 상단일 측정에서 측정하는 기저를 X 기저로 변경할 수 있다.

[0190] 일 실시예에서, 프로세서(110)는 양자 회로(1920)에 도시된 바와 같이, 단일 측정에서 측정하는 기저가 X 기저인 경우, 논리적 Hadamard 연산과 단일 측정을 병합함으로써, 단일 측정에서 측정하는 기저를 Z 기저로 변경할 수 있다.

[0191] 이하에서는 종래에 제안된 논리적 H 연산의 구현 방법과 본 개시의 일 실시예에 따른 논리적 H 연산의 구현 방법을 비교한 결과를 후술한다.

[0192] 아래의 표 3은 종래에 제안된 논리적 H 연산 방법과 본 개시의 일 실시예에 따른 논리적 H 연산의 물리적 H 연산 수행 여부, 시간 비용, 공간 비용, 공간-시간 비용을 비교한 표이다.

표 3

Method	물리적 Hadamard 수행	시간 비용	공간 비용 (논리적 큐비트 당)	공간-시간 비용 (논리적 큐비트 당)	Reference
Merging with surrounding areas	○	$4d$	$1d^2 + 3d$	$4d^2 + 12d^2$	Horsman, 2012
Quantum teleportation	×	$7d$	$2d^2$	$14d^3$	Erhard, 2021
Boundary flip and transformation	×	$1d$	$1d^2$	$1d^3$	본 개시에서 제안

[0193]

[0194] 표 3을 참조하면, 2012년 Horsman이 제안한 방법(Clare Horsman, Austin G Fowler, Simon Devitt, Rodney Van

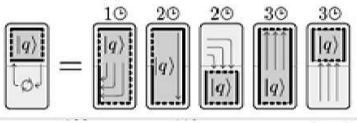
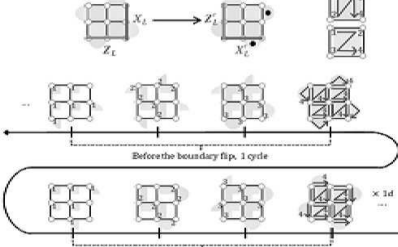
Meter, "Surface code quantum computing by lattice surgery", New Journal of Physics, Dec. 2012)은 물리적 Hadamard를 수행하는 경우로, 주변 영역과 병합하여 로테이션(rotation)을 수행한다. 이때, 물리적 큐비트(physical qubit)의 스왑 연산(swap operation)이 필요하므로 모든 논리적 큐비트(패치)(logical qubit(patch)) 마다 여유 공간이 필요하므로 하나의 논리적 큐비트 당 필요한 물리적 큐비트 수는 $1d^2 + 3d$ 가 된다.

[0195] 2021년 Erhard의 연구(Alexander Erhard, Hendrik Poulsen Nautrup, Michael Meth, Lukas Postler, Roman Stricker, Martin Stadler, Vlad Negnevitsky, Martin Ringbauer, Philipp Schindler, Hans J. Briegel, Rainer Blatt, Nicolai Friis, Thomas Monz, "Entangling logical qubits with lattice surgery", Nature, vol.589, pp.220-224, Jan. 2021)에서는 1개의 데이터 패치(data patch)에 Hadamard 연산을 수행하기 위해 양자 전송(Quantum teleportation)을 활용한다. 따라서 0 상태인 논리적 큐비트 1개를 소모하며, Hadamard가 적용된 데이터는 0 상태였던 논리적 큐비트에 저장된다. Checkerboard-type, row-type 등의 큐비트 레이아웃(Qubit layout)를 고려할 경우, Hadamard 연산은 논리적 큐비트 추가로 사용하기 때문에 라우팅(routing) 시 고려 대상이 된다. 이외에도 데이터를 다시 원래 자리로 되돌리기 위한 시간도 추가된다. 따라서 공간 비용 뿐만 아니라 시간 비용에도 영향을 끼친다.

[0196] 한편, 이하에서는 종래에 제안된 패치의 로테이션 방법과 본 개시의 일 실시예에 따른 경계 반전 방법을 비교한 결과를 후술한다.

[0197] 아래의 표 4는 종래에 제안된 패치의 로테이션 방법과 본 개시의 일 실시예에 따른 경계 반전 방법의 시간 비용, 공간 비용, 공간-시간 비용을 비교한 표이다.

표 4

Method	시간 비용	공간 비용 (논리 큐비트 당)	공간-시간 비용 (논리 큐비트 당)	Reference
Merging with surrounding areas	$3d$	$2d^2$	$6d^3$	 Litinski 2019
Boundary flip and transformation	$1d$	$1d^2$	$1d^3$	 본 발명에서 제안

[0198]

[0199] 표 4를 참조하면, Litinski의 2019년 연구(Litinski, Daniel. "A game of surface codes: Large-scale quantum computing with lattice surgery." Quantum 3 (2019): 128.)는 주변 공간과 병합 및 경계 변형을 반복하기 때문에, 본 개시의 일 실시예에 따른 경계 반전 방법보다 시간 비용이 3배가 더 소요되고, 병합할 추가 공간이 필요하므로 공간 비용이 2배가 더 소요된다.

[0200] 도 13 내지 19를 참조하면 상술한 바와 같이, 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 경계 반전에 후속 측정의 변형(Transformation of the subsequent measurement)을 결합하여 논리적 Hadamard 수행할 수 있다. 해당 방식은 경계 반전에 1d cycle이 소요되며, 논리 큐비트 주변에 추가 공간(extra space)을 준비할 필요가 없다.

[0201] 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 경계 반전과 후속 측정 변형을 통해 종래 방법보다 적은 시간 및 공간 비용으로 H 연산의 수행이 가능할 수 있다.

[0202] 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 논리적 Hadamard를 수행하는 과정에서 물리적 Hadamard를 수행하는 대신 추후 연산의 기저를 변화시킬 수 있다.

- [0203] 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 변화된 기저에 맞춰 논리적 큐비트의 경계를 쉽게 변형할 수 있도록 경계 반전 기법을 사용할 수 있다.
- [0204] 본 개시의 일 실시예에 따른 양자 컴퓨팅 장치(100)는 경계 반전을 사용함으로써, 종래의 방식 보다 적은 시간 내에, 추가 큐비트 소모 없이 논리적 큐비트의 경계를 변형할 수 있다.
- [0205] 도 20은 본 개시의 일 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 간략하고 일반적인 개략도를 도시한다.
- [0206] 본 개시는 일반적으로 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어와 관련하여 전술되었지만, 당업자라면 본 개시가 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로서 구현될 수 있다는 것을 잘 알 것이다.
- [0207] 일반적으로, 본 명세서에서의 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로시저, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 발명의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서-기반 또는 프로그램가능 가전 제품, 기타 등등(이들 각각은 하나 이상의 연관된 장치와 연결되어 동작할 수 있음)을 비롯한 다른 컴퓨터 시스템 구성으로 실시될 수 있다는 것을 잘 알 것이다.
- [0208] 본 발명의 설명된 실시예들은 또한 어떤 태스크들이 통신 네트워크를 통해 연결되어 있는 원격 처리 장치들에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.
- [0209] 컴퓨터는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 일시적(transitory) 및 비일시적(non-transitory) 매체, 이동식 및 비-이동식 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독가능 매체는 컴퓨터 판독가능 저장 매체 및 컴퓨터 판독가능 전송 매체를 포함할 수 있다.
- [0210] 컴퓨터 판독가능 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성 매체, 일시적 및 비-일시적 매체, 이동식 및 비이동식 매체를 포함한다. 컴퓨터 판독가능 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital video disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함하지만, 이에 한정되지 않는다.
- [0211] 컴퓨터 판독가능 전송 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터들을 구현하고 모든 정보 전달 매체를 포함한다. 피변조 데이터 신호라는 용어는 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 제한이 아닌 예로서, 컴퓨터 판독가능 전송 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들 중 임의의 것의 조합도 역시 컴퓨터 판독가능 전송 매체의 범위 안에 포함되는 것으로 한다.
- [0212] 컴퓨터(2002)를 포함하는 본 발명의 여러가지 측면들을 구현하는 예시적인 환경이 나타내어져 있으며, 컴퓨터(2002)는 처리 장치(2004), 시스템 메모리(2006) 및 시스템 버스(2008)를 포함한다. 시스템 버스(2008)는 시스템 메모리(2006)(이에 한정되지 않음)를 비롯한 시스템 컴포넌트들을 처리 장치(2004)에 연결시킨다. 처리 장치(2004)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(2004)로서 이용될 수 있다.
- [0213] 시스템 버스(2008)는 메모리 버스, 주변장치 버스, 및 다양한 상용 버스 아키텍처 중 임의의 것을 사용하는 로컬 버스에 추가적으로 상호 연결될 수 있는 몇 가지 유형의 버스 구조 중 임의의 것일 수 있다. 시스템 메모리(2006)는 판독 전용 메모리(ROM)(2010) 및 랜덤 액세스 메모리(RAM)(2012)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(2010)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(2002) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(2012)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.
- [0214] 컴퓨터(2002)는 또한 내장형 하드 디스크 드라이브(HDD)(2014)(예를 들어, EIDE, SATA)-이 내장형 하드 디스크

드라이브(2014)는 또한 적당한 새시(도시 생략) 내에서 외장형 용도로 구성될 수 있음-, 자기 플로피 디스크 드라이브(FDD)(2016)(예를 들어, 이동식 디스켓(2018)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(2020)(예를 들어, CD-ROM 디스크(2022)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(2014), 자기 디스크 드라이브(2016) 및 광 디스크 드라이브(2020)는 각각 하드 디스크 드라이브 인터페이스(2024), 자기 디스크 드라이브 인터페이스(2026) 및 광 드라이브 인터페이스(2028)에 의해 시스템 버스(2008)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(2024)는 예를 들어, USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘 다를 포함한다.

[0215] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등등의 비휘발성 저장을 제공한다. 컴퓨터(2002)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 저장 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 zip 드라이브(zip drive), 자기 카세트, 플래쉬 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 저장 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 발명의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

[0216] 운영 체제(2030), 하나 이상의 어플리케이션 프로그램(2032), 기타 프로그램 모듈(2034) 및 프로그램 데이터(2036)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(2012)에 저장될 수 있다. 운영 체제, 어플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부분이 또한 RAM(2012)에 캐싱될 수 있다. 본 발명이 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.

[0217] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(2038) 및 마우스(2040) 등의 포인팅 장치를 통해 컴퓨터(2002)에 명령 및 정보를 입력할 수 있다. 기타 입력 장치(도시 생략)로는 마이크, IR 리모콘, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린, 기타 등등이 있을 수 있다. 이들 및 기타 입력 장치가 종종 시스템 버스(2008)에 연결되어 있는 입력 장치 인터페이스(2042)를 통해 처리 장치(2004)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.

[0218] 모니터(2044) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(2046) 등의 인터페이스를 통해 시스템 버스(2008)에 연결된다. 모니터(2044)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.

[0219] 컴퓨터(2002)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(2048) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(2048)는 워크스테이션, 서버 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(2002)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(2050)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(2052) 및/또는 더 큰 네트워크, 예를 들어, 원거리 통신망(WAN)(2054)에의 유선/무선 연결을 포함한다. 이러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 일반적인 것이며, 인트라넷 등의 전사적 컴퓨터 네트워크(enterprise-wide computer network)를 용이하게 해주며, 이들 모두는 전세계 컴퓨터 네트워크, 예를 들어, 인터넷에 연결될 수 있다.

[0220] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(2002)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(2056)를 통해 로컬 네트워크(2052)에 연결된다. 어댑터(2056)는 LAN(2052)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(2052)은 또한 무선 어댑터(2056)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(2002)는 모뎀(2058)을 포함할 수 있거나, WAN(2054) 상의 통신 서버에 연결되거나, 또는 인터넷을 통하는 등, WAN(2054)을 통해 통신을 정하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(2058)은 직렬 포트 인터페이스(2042)를 통해 시스템 버스(2008)에 연결된다. 네트워킹된 환경에서, 컴퓨터(2002)에 대해 설명된 프로그램 모듈들 또는 그의 일부분이 원격 메모리/저장 장치(2050)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.

[0221] 컴퓨터(2002)는 무선 통신으로 배치되어 동작하는 임의의 무선 장치 또는 개체, 예를 들어, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, PDA(portable data assistant), 통신 위성, 무선 검출가능 태그와 연관된 임의

의 장비 또는 장소, 및 전화와 통신을 하는 동작을 한다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술을 포함한다. 따라서, 통신은 종래의 네트워크에서와 같이 미리 정의된 구조이거나 단순히 적어도 2개의 장치 사이의 애드혹 통신(ad hoc communication)일 수 있다.

[0222] Wi-Fi(Wireless Fidelity)는 유선 없이도 인터넷 등으로의 연결을 가능하게 해준다. Wi-Fi는 이러한 장치, 예를 들어, 컴퓨터가 실내에서 및 실외에서, 즉 기지국의 통화권 내의 아무 곳에서나 데이터를 전송 및 수신할 수 있게 해주는 셀 전화와 같은 무선 기술이다. Wi-Fi 네트워크는 안전하고 신뢰성 있으며 고속인 무선 연결을 제공하기 위해 IEEE 802.11(a,b,g, 기타)이라고 하는 무선 기술을 사용한다. 컴퓨터를 서로에, 인터넷에 및 유선 네트워크(IEEE 802.3 또는 이더넷을 사용함)에 연결시키기 위해 Wi-Fi가 사용될 수 있다. Wi-Fi 네트워크는 비인가 2.4 및 5 GHz 무선 대역에서, 예를 들어, 11Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트로 동작하거나, 양 대역(듀얼 대역)을 포함하는 제품에서 동작할 수 있다.

[0223] 본 발명의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 "소프트웨어"로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 어플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 발명의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 어플리케이션에 대하여 다양한 방식으로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 발명의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.

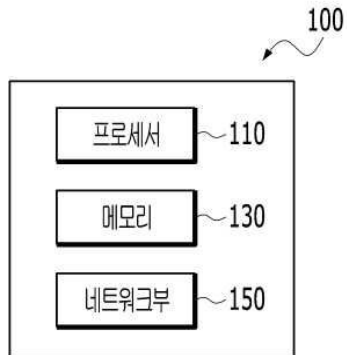
[0224] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조물품(article)으로 구현될 수 있다. 용어 "제조 물품"은 임의의 컴퓨터-관독가능 장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-관독가능 저장 매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트림, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래시 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 용어 "기계-관독가능 매체"는 명령(들) 및/또는 데이터를 저장, 보유, 및/또는 전달할 수 있는 무선 채널 및 다양한 다른 매체를 포함하지만, 이들로 제한되는 것은 아니다.

[0225] 제시된 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조는 예시적인 접근들의 일례임을 이해하도록 한다. 설계 우선순위들에 기반하여, 본 발명의 범위 내에서 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조가 재배열될 수 있다는 것을 이해하도록 한다. 첨부된 방법 청구항들은 샘플 순서로 다양한 단계들의 엘리먼트들을 제공하지만 제시된 특정한 순서 또는 계층 구조에 한정되는 것을 의미하지는 않는다.

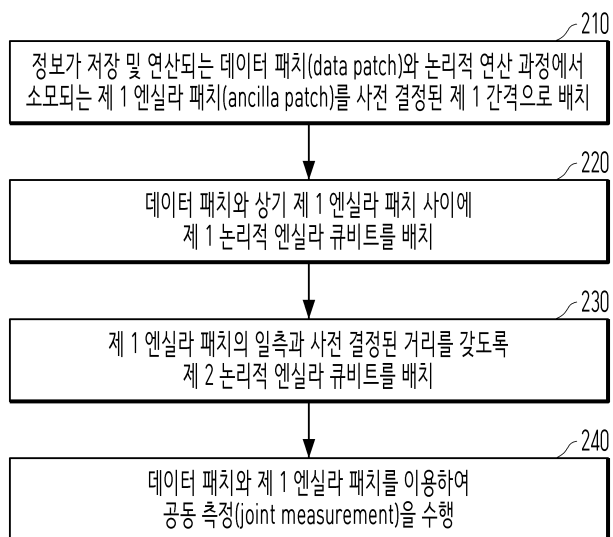
[0226] 제시된 실시예들에 대한 설명은 임의의 본 발명의 기술 분야에서 통상의 지식을 가진 자가 본 발명을 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 발명의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 발명의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 발명은 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의 범위에서 해석되어야 할 것이다.

도면

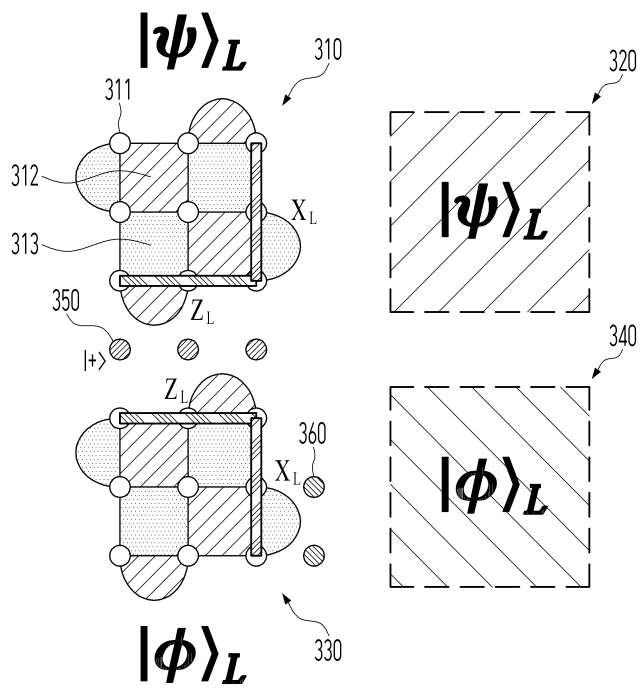
도면1



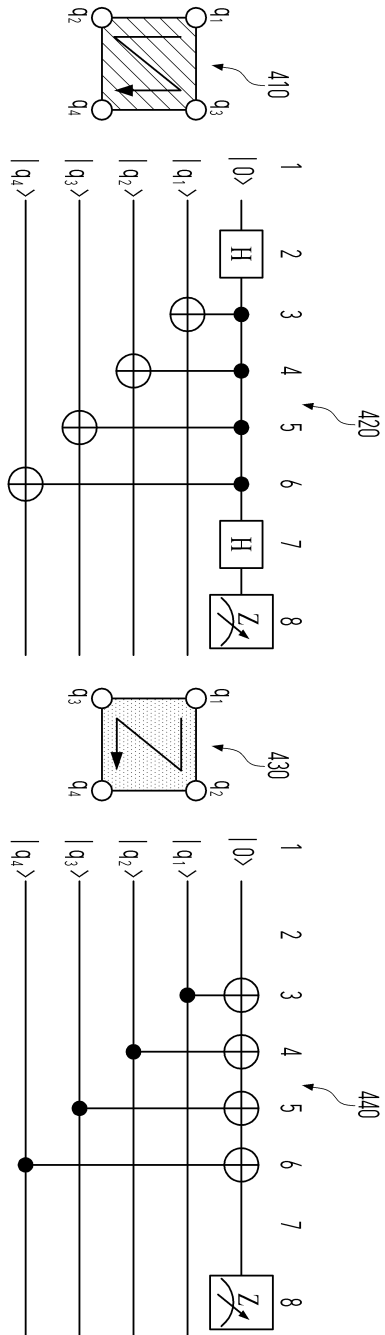
도면2



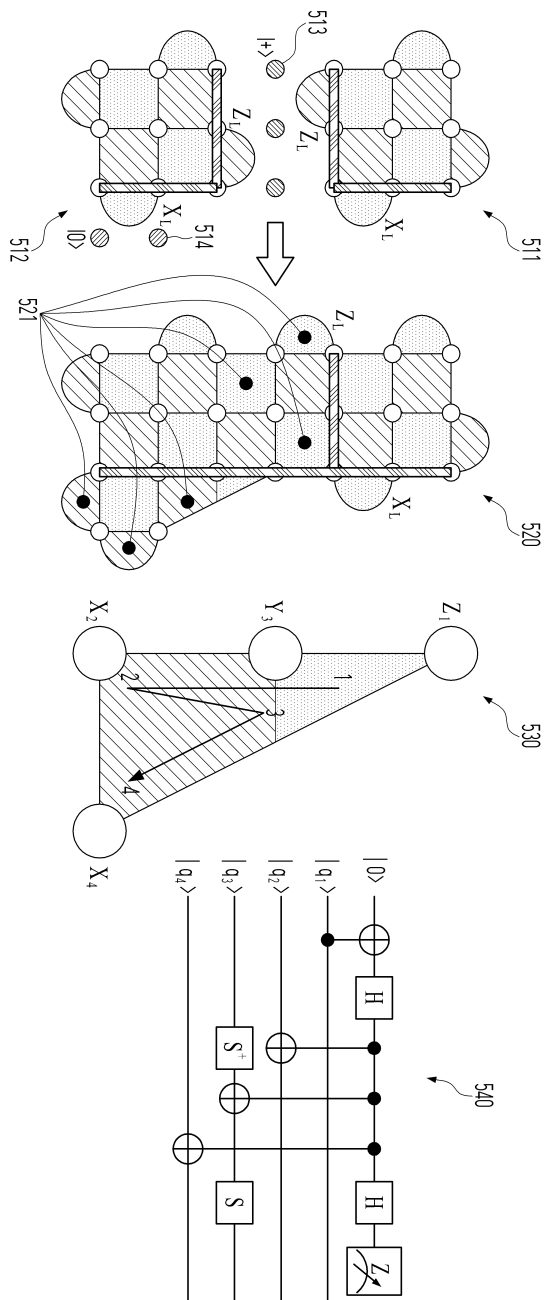
도면3



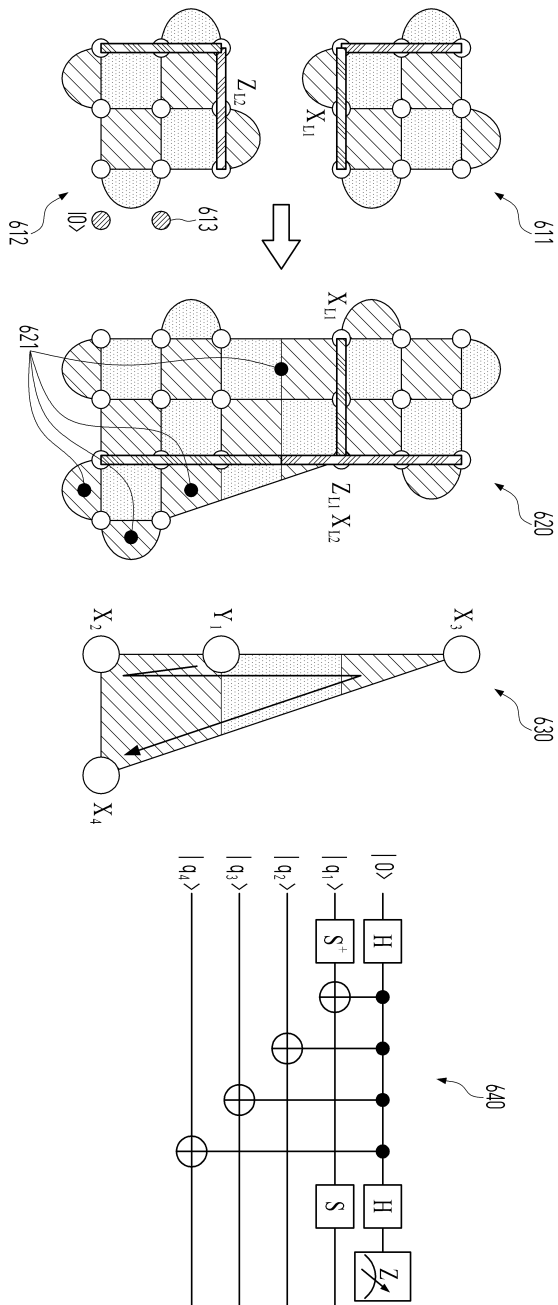
도면4



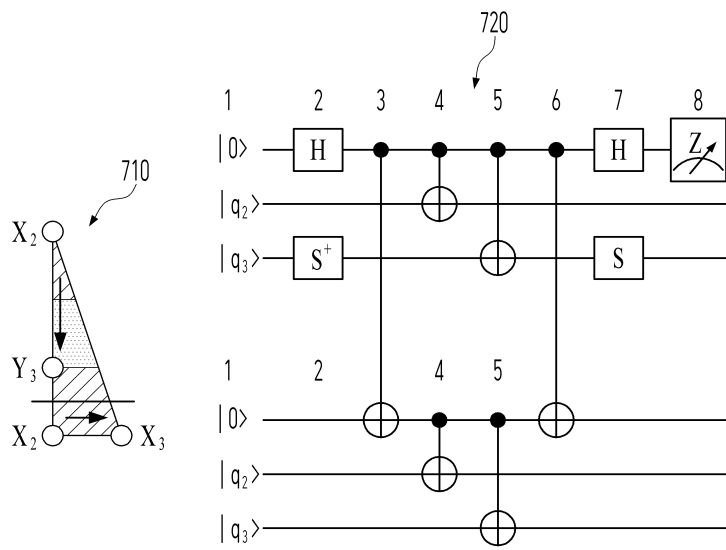
도면5



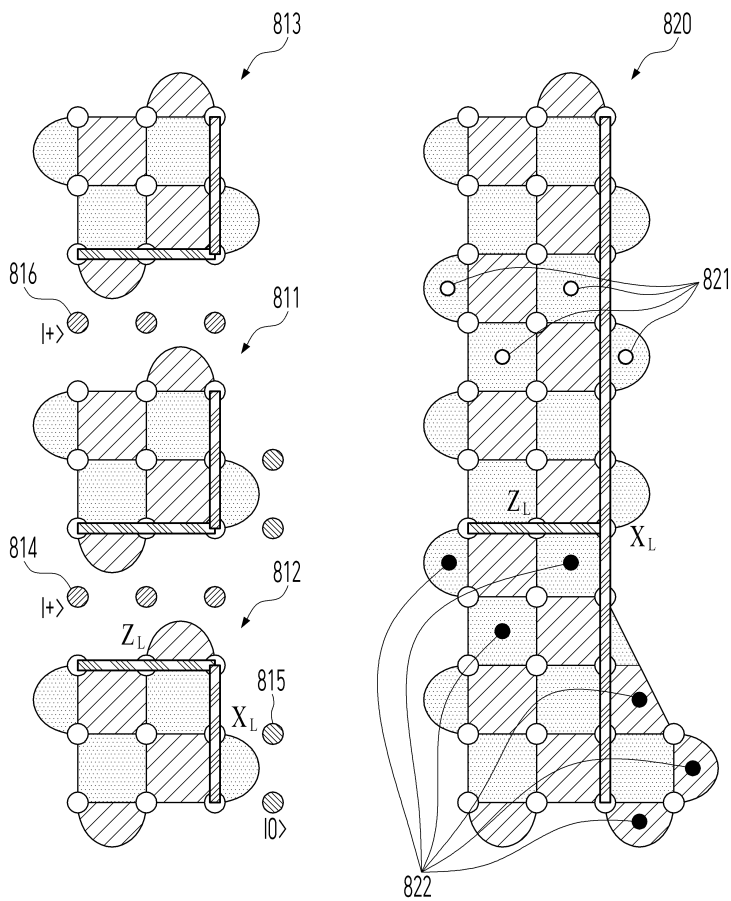
도면6



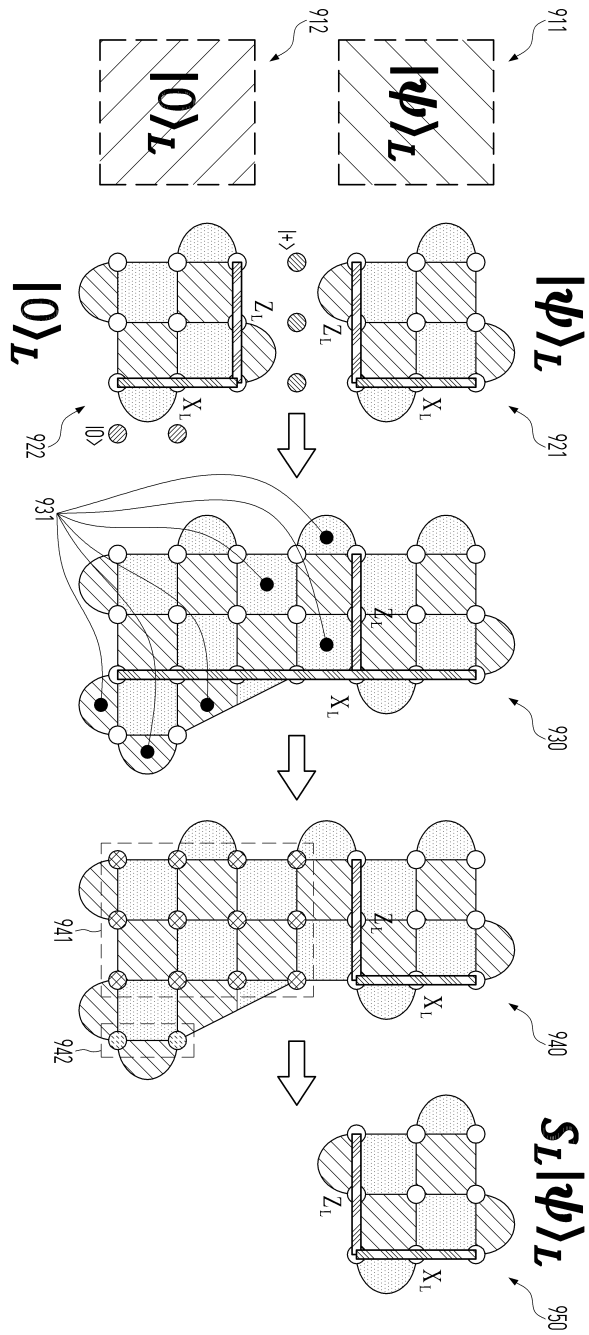
도면7



도면8



도면9

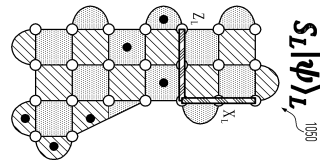
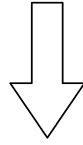
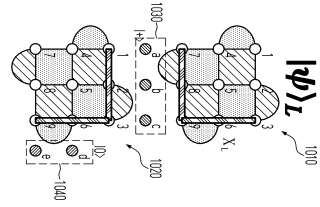


도면10

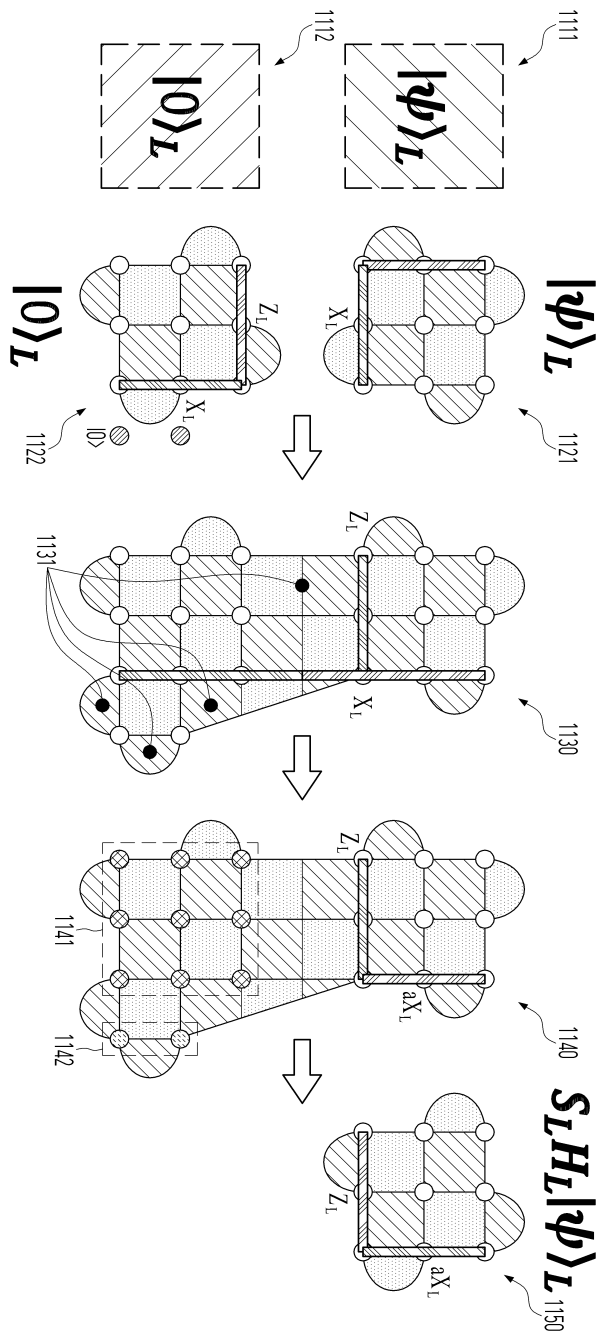
	1	2	3	4	5	6	7	8	9	a	b	c	1	2	3	4	5	6	7	8	9	d	e
K_1		X	X																				
K_2				X	X																		
K_3			X	X																			
K_4								X	X														
K_5		Z	Z		Z	Z																	
K_6				Z	Z		Z	Z															
K_7	Z		Z																				
K_8				Z			Z																
X_U				X		X			X														
Z_U						Z	Z	Z															
K_9									X														
K_{10}										X													
K_{11}											X												
K_{12}												X	X										
K_{13}														X	X								
K_{14}													Z	Z	Z	Z							
K_{15}													Z	Z			Z						
K_{16}															Z			Z					
K_{17}													Z	Z	Z							Z	
K_{18}																						Z	
K_{19}																							Z

	1	2	3	4	5	6	7	8	9	a	b	c	1	2	3	4	5	6	7	8	9	d	e
K_1		X	X		X	X																	
K_2				X	X																		
K_3			X	X																			
K_4, K_5, K_6								X	X														
K_5		Z	Z		Z	Z																	
K_6				Z	Z		Z	Z															
K_7				Z																			
K_8					Z			Z															
Y_L				X		X	Z	Z	XZ				X			X						X	
Z_L						Z	Z	Z	Z														
K_{10}						Z				Z													
K_{11}							Z	Z															
K_{12}																							
K_{13}																							
K_{14}																							
K_{15}																							
K_{16}, K_{17}, K_{18}																							
K_{17}																							
K_{18}																							
K_{19}																							

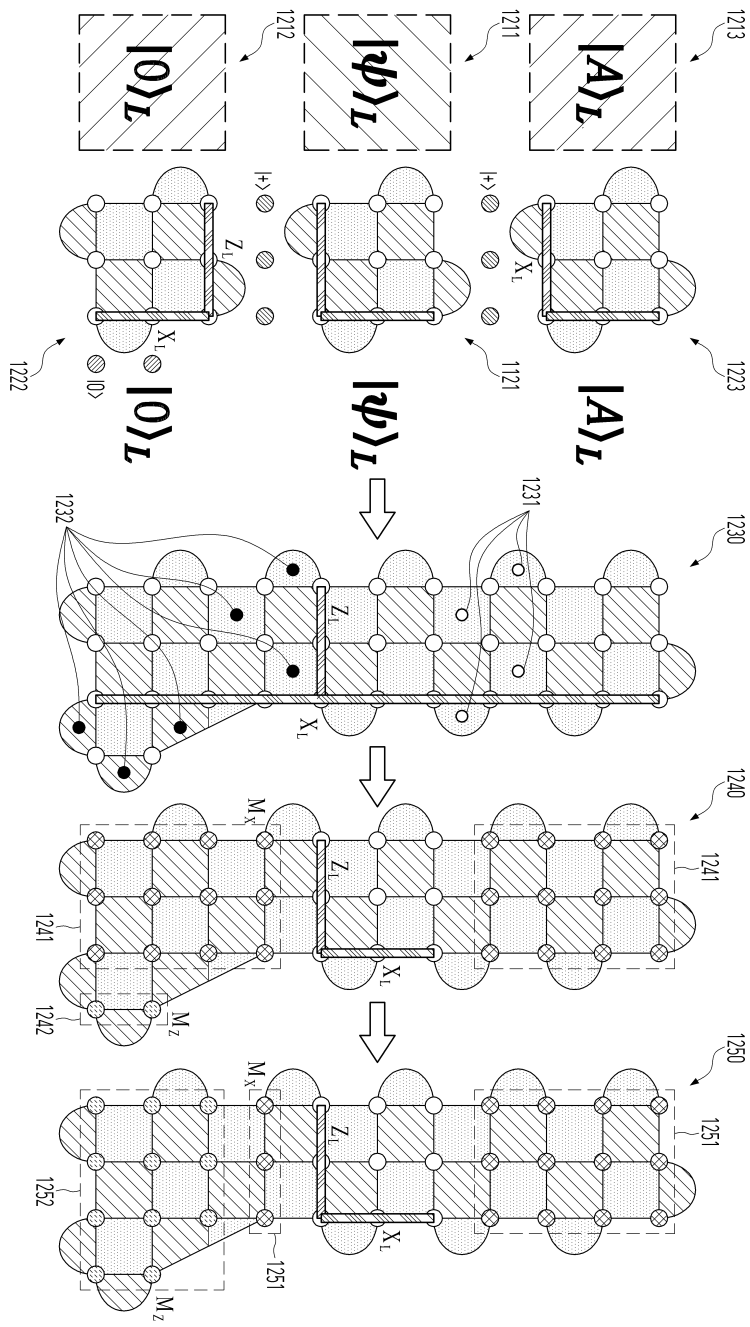
$10\rangle_L$



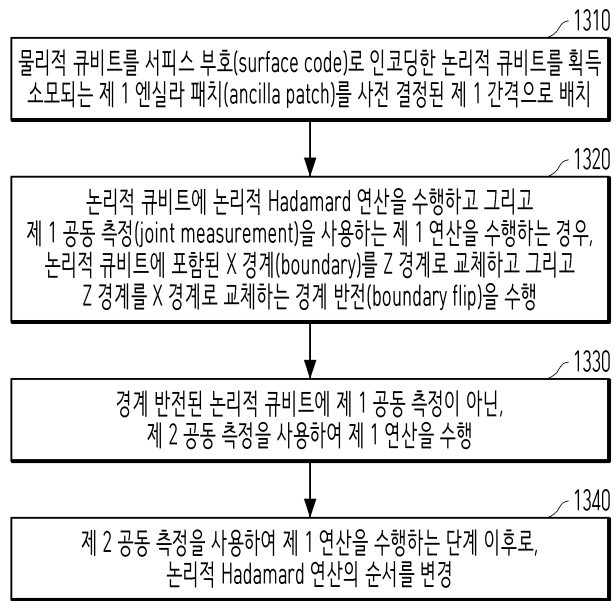
도면11



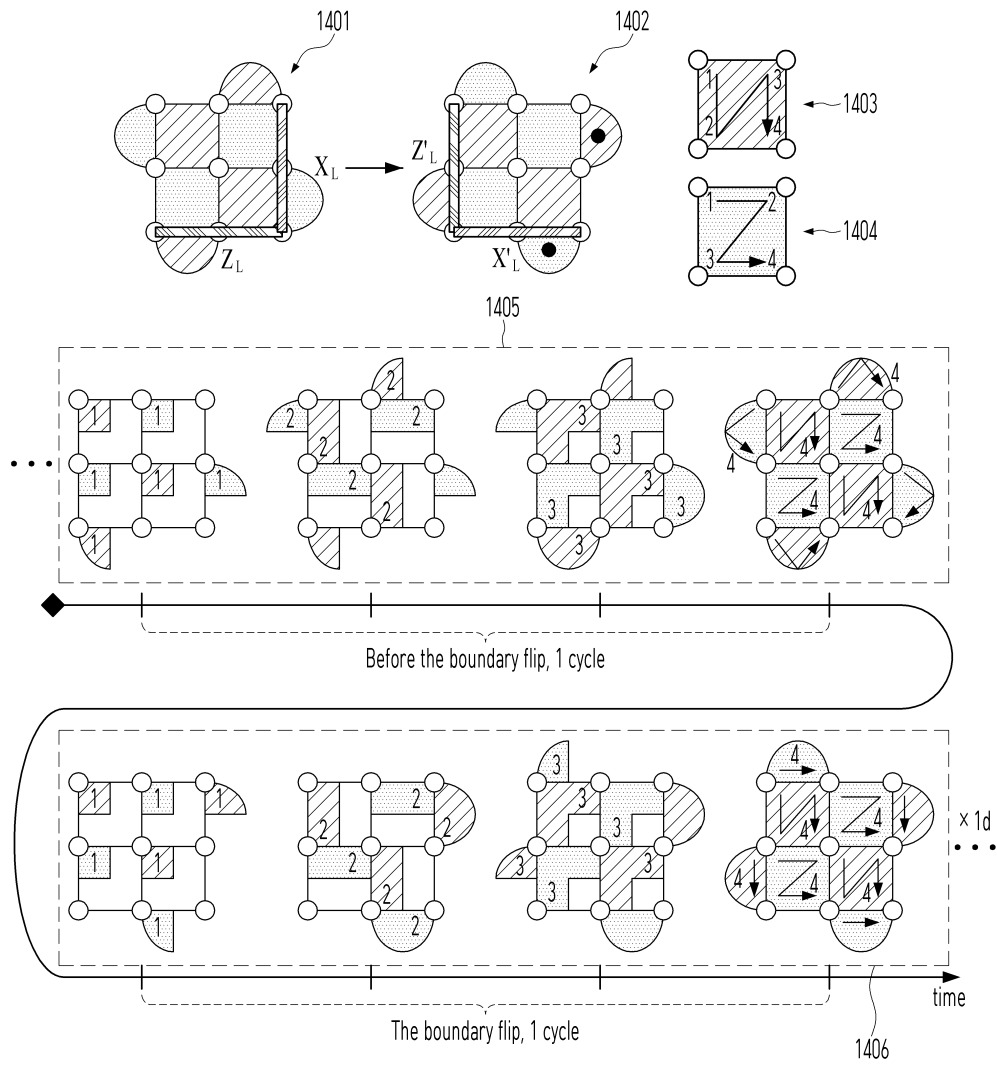
도면12



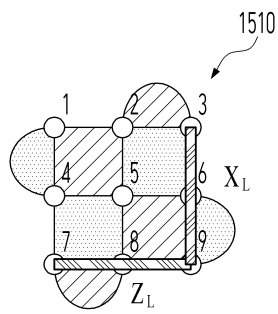
도면13



도면14

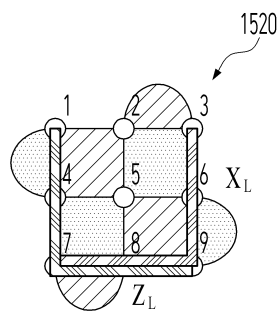


도면 15a



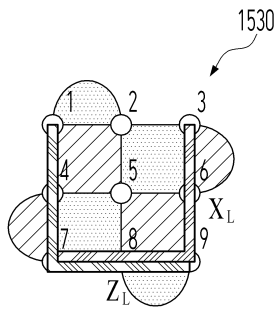
		1	2	3	4	5	6	7	8	9
K_1		X	X		X	X				
K_2						X	X		X	X
K_3			Z	Z		Z	Z			
K_4					Z	Z		Z	Z	
K_5			X	X						
K_6								X	X	
K_7		Z			Z					
K_8							Z			Z
X_L								Z	Z	Z
Z_L				X			X			X

도면 15b



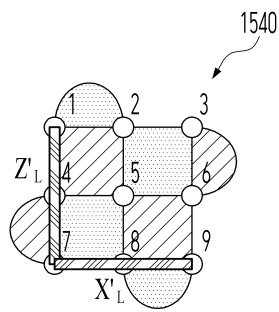
		1	2	3	4	5	6	7	8	9
K_1		X	X		X	X				
K_2						X	X		X	X
K_3			Z	Z		Z	Z			
K_4					Z	Z		Z	Z	
K_5			X	X						
K_6								X	X	
K_7		Z			Z					
K_8							Z			Z
$K_7 Z_L$		Z			Z			Z	Z	Z
$K_6 X_L$				X			X	X	X	X

도면15c



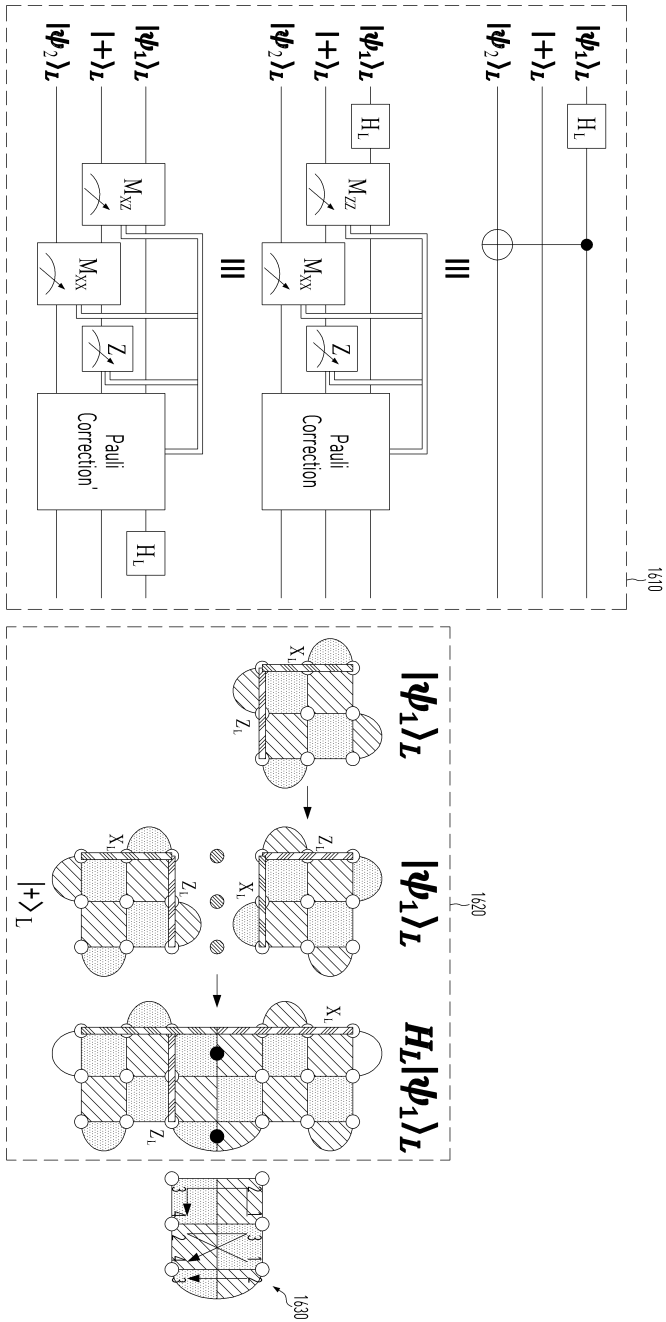
		1	2	3	4	5	6	7	8	9
K_1		X	X		X	X				
K_2						X	X		X	X
K_3			Z	Z		Z	Z			
K_4					Z	Z		Z	Z	
K_9	$(-1)^{M_9}$	Z	Z							
K_{10}	$(-1)^{M_{10}}$								Z	Z
K_{11}	$(-1)^{M_{11}}$				X			X		
K_{12}	$(-1)^{M_{12}}$			X			X			
$K_7 Z_L$		Z			Z			Z	Z	Z
$K_6 X_L$				X			X	X	X	X

도면15d

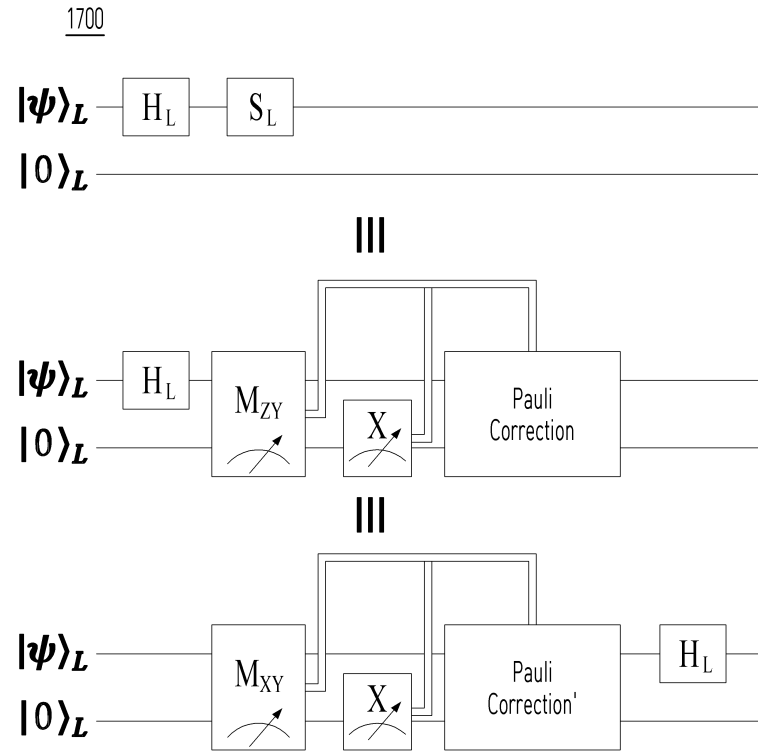


		1	2	3	4	5	6	7	8	9
K_1		X	X		X	X				
K_2						X	X		X	X
K_3			Z	Z		Z	Z			
K_4					Z	Z		Z	Z	
K_9	$(-1)^{M_9}$	Z	Z							
K_{10}	$(-1)^{M_{10}}$								Z	Z
K_{11}	$(-1)^{M_{11}}$				X			X		
K_{12}	$(-1)^{M_{12}}$			X			X			
$Z'_L = K_{10} K_7 Z_L$	$(-1)^{M_{10}}$	Z			Z			Z		
$X'_L = K_{12} K_6 X_L$	$(-1)^{M_{12}}$							X	X	X

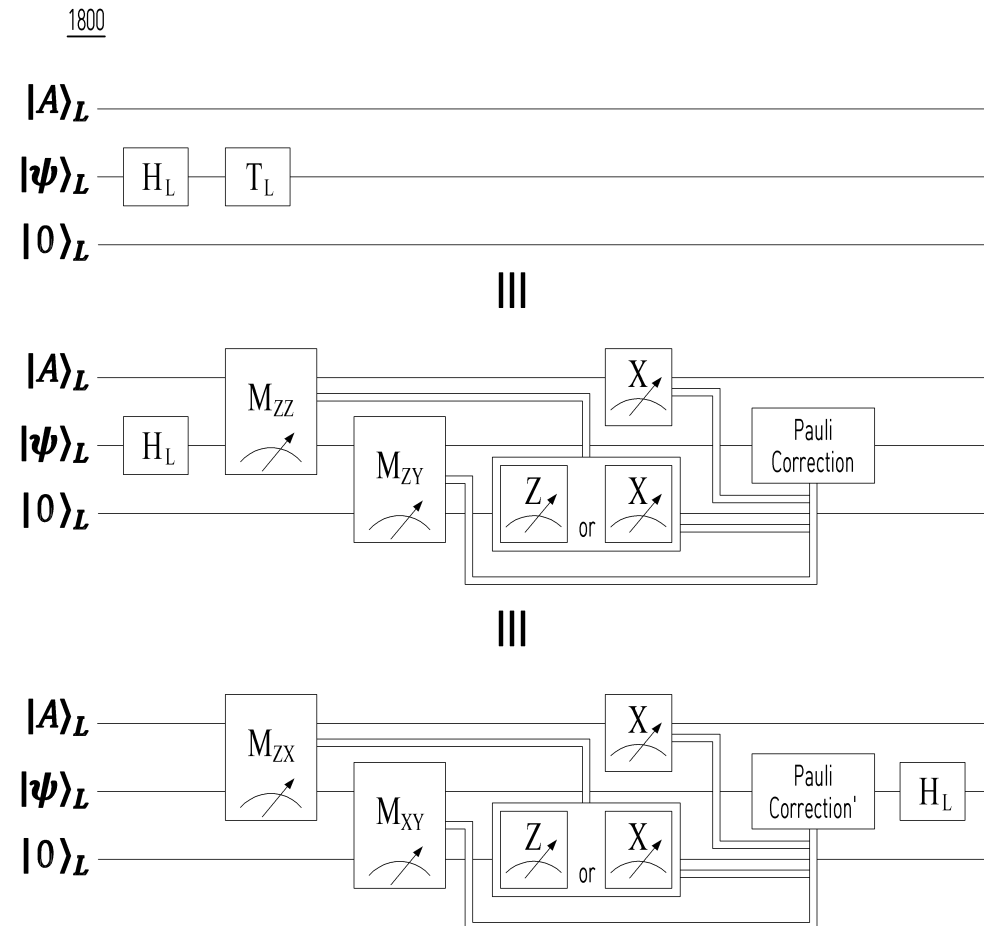
도면16



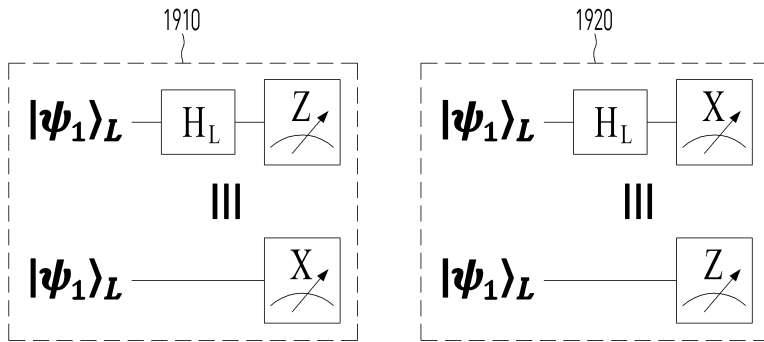
도면17



도면18



도면19



도면20

