



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0174676
(43) 공개일자 2023년12월28일

(51) 국제특허분류(Int. Cl.)
G06N 10/20 (2022.01) G06N 10/60 (2022.01)
G06N 10/70 (2022.01) B82Y 10/00 (2017.01)
(52) CPC특허분류
G06N 10/70 (2022.08)
G06N 10/20 (2022.01)
(21) 출원번호 10-2022-0113870
(22) 출원일자 2022년09월08일
심사청구일자 2022년09월08일
(30) 우선권주장
1020220075912 2022년06월21일 대한민국(KR)

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암
동5가)
(72) 발명자
허준
서울특별시 강남구 광평로10길 6, 206동 103호(일
원동, 한솔마을아파트)
이종현
서울특별시 동대문구 회기로12길 47-9, 101호(회
기동)
(뒷면에 계속)
(74) 대리인
이대호, 박건홍

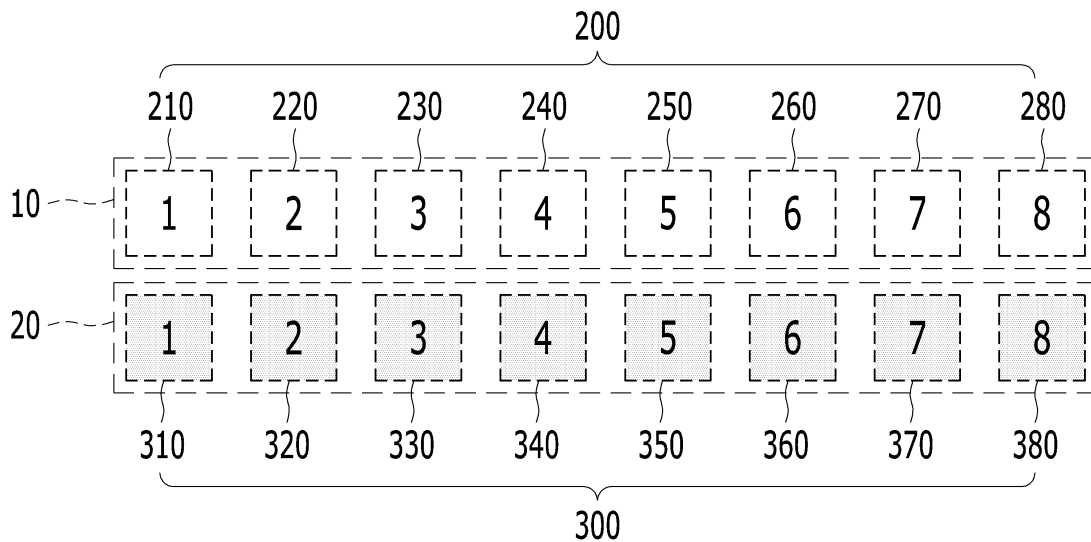
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 양자 논리적 큐비트의 논리적 CNOT 연산 방법

(57) 요약

본 개시의 일 실시예에 따라, 양자 컴퓨팅 장치에 의해 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 방법으로서, 복수의 논리적 데이터 큐비트들(logical data qubits)을 가로열인 제 1 열로 배치하는 단계; 상기 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 상기 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 상기 제 1 열과 상이한 가로열인 제 2 열로 배치하는 단계; 및 상기 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행하는 단계;를 포함할 수 있다.

대표도 - 도3



(52) CPC특허분류

G06N 10/60 (2022.01)

B82Y 10/00 (2013.01)

(72) 발명자

강유진

서울특별시 성북구 지봉로21길 22, 401호(보문동2가)

하진영

서울특별시 성북구 고려대로12길 13-3, 302호(안암동2가)

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	1/2
과제수행기관명	한국전자통신연구원
연구기간	2022.01.01 ~ 2022.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711152525
과제번호	2021-0-01810-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	양자세계를 연결하는 초신뢰 양자인터넷 요소 기술 개발
기 여 율	1/2
과제수행기관명	고려대학교산학협력단
연구기간	2022.01.01 ~ 2022.12.31

명세서

청구범위

청구항 1

양자 컴퓨팅 장치에 의해 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 방법으로서,

복수의 논리적 데이터 큐비트들(logical data qubits)을 가로열인 제 1 열로 배치하는 단계;

상기 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 상기 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 상기 제 1 열과 상이한 가로열인 제 2 열로 배치하는 단계; 및

상기 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행하는 단계;

를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 제 1 논리적 데이터 큐비트는 제어 큐비트(control qubit)이고, 그리고

상기 제 2 논리적 데이터 큐비트는 타겟 큐비트(target qubit)인,

방법.

청구항 3

제 1 항에 있어서,

상기 복수의 논리적 엔실라 큐비트들은 각각 중간의 논리적 큐비트(intermediate logical qubit)를 포함하는, 방법.

청구항 4

제 2 항에 있어서,

상기 복수의 논리적 데이터 큐비트들 중 상기 제 1 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT를 수행하는 단계는,

상기 제 1 논리적 데이터 큐비트와 상기 제 2 논리적 데이터 큐비트 사이에 다른 논리적 데이터 큐비트가 존재하지 않는 경우, 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 1 병합 큐비트를 생성하는 단계;

상기 제 2 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트를 병합하여 제 2 병합 큐비트를 생성하는 단계;

상기 제 1 병합 큐비트를 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 엔실라 큐비트로 분할(splitting)하는 단계;

분할된 상기 제 1 논리적 엔실라 큐비트와 상기 제 2 병합 큐비트를 병합하여 제 3 병합 큐비트를 생성하는 단

계; 및

상기 제 3 병합 큐비트를 상기 제 2 논리적 데이터 큐비트, 상기 제 1 논리적 엔실라 큐비트 및 상기 제 2 논리적 엔실라 큐비트로 분할하고, 그리고 상기 제 1 논리적 엔실라 큐비트 및 상기 제 2 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행하는 단계;

를 포함하는,

방법.

청구항 5

제 2 항에 있어서,

상기 복수의 논리적 데이터 큐비트들 중 상기 제 1 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT를 수행하는 단계는,

상기 제 1 논리적 데이터 큐비트와 상기 제 2 논리적 데이터 큐비트 사이에 적어도 하나의 제 3 논리적 데이터 큐비트가 존재하는 경우, 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 4 병합 큐비트를 생성하는 단계;

상기 제 2 논리적 데이터 큐비트, 상기 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트 및 상기 적어도 하나의 제 3 논리적 데이터 큐비트에 대응되는 적어도 하나의 제 3 논리적 엔실라 큐비트를 병합하여 제 5 병합 큐비트를 생성하는 단계;

상기 제 4 병합 큐비트를 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 엔실라 큐비트로 분할(splitting)하는 단계;

분할된 상기 제 1 논리적 엔실라 큐비트와 상기 제 5 병합 큐비트를 병합하여 제 6 병합 큐비트를 생성하는 단계; 및

상기 제 6 병합 큐비트를 상기 제 2 논리적 데이터 큐비트, 상기 제 1 논리적 엔실라 큐비트, 상기 제 2 논리적 엔실라 큐비트 및 상기 적어도 하나의 제 3 논리적 엔실라 큐비트로 분할하고, 그리고 상기 제 1 논리적 엔실라 큐비트, 상기 제 2 논리적 엔실라 큐비트 및 상기 적어도 하나의 제 3 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행하는 단계;

를 포함하는,

방법.

청구항 6

제 1 항에 있어서,

상기 복수의 논리적 데이터 큐비트들을 가로열인 상기 제 1 열로 배치하는 단계는,

상기 복수의 논리적 데이터 큐비트들을 수행 회로에서 관여되는 큐비트 연산의 개수에 기초하여 상기 제 1 열로 배치하는 단계;

를 포함하는,

방법.

청구항 7

제 6 항에 있어서,

상기 복수의 논리적 데이터 큐비트들을 상기 수행 회로에서 관여되는 상기 큐비트 연산의 개수에 기초하여 상기

제 1 열로 배치하는 단계는,

상기 큐비트 연산의 개수에 기초하여 1순위로 결정된 제 4 논리적 데이터 큐비트를 상기 제 1 열의 중심부에 배치하는 단계;

상기 큐비트 연산의 개수에 기초하여 2순위로 결정된 제 5 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 4 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계; 및

상기 큐비트 연산의 개수에 따라 3순위로 결정된 제 6 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 4 논리적 데이터 큐비트의 타측에 대응되도록 배치하는 단계;

를 포함하는,

방법.

청구항 8

제 7 항에 있어서,

상기 큐비트 연산의 개수에 따라 4순위로 결정된 제 7 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 6 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계; 및

상기 큐비트 연산의 개수에 따라 5순위로 결정된 제 8 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 5 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계;

를 더 포함하고,

상기 제 5 논리적 데이터 큐비트의 일측은 상기 제 4 논리적 데이터 큐비트의 일측에 대응되지 않는 부분이고, 그리고

상기 제 6 논리적 데이터 큐비트의 일측은 상기 제 4 논리적 데이터 큐비트의 타측에 대응되지 않는 부분인,

방법.

청구항 9

제 1 항에 있어서,

복수의 논리적 CNOT들을 수행하는 경우, 스케줄링(scheduling) 함수를 포함하는 라우팅(routing) 알고리즘을 이용하여 상기 복수의 논리적 CNOT들의 연산의 순서를 결정하는 단계;

를 더 포함하는,

방법.

청구항 10

양자 컴퓨팅 장치에 의해 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 방법으로서,

복수의 논리적 데이터 큐비트들(logical data qubits)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열로 배치하는 단계;

상기 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 상기 적어도 하나의 제 3 열과 상이한 가로열인 적어도 하나의 제 4 열로 배치하는 단계;

복수의 추가 논리적 엔실라 큐비트들을 상기 적어도 하나의 제 3 열 및 상기 적어도 하나의 제 4 열의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열로 배치하는 단계; 및

상기 복수의 논리적 데이터 큐비트들 중 제 9 논리적 데이터 큐비트 및 제 10 논리적 데이터 큐비트에 대한 논

리적 CNOT(controlled NOT)를 수행하는 단계;
를 포함하는,
방법.

발명의 설명

기술 분야

[0001] 본 개시 내용은 양자 오류정정 부호에 관한 것으로, 보다 구체적으로, 양자 논리적 큐비트의 논리적 CNOT 연산 방법에 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] 인코딩된 논리적 큐비트를 배치하는 방식은 해당 큐비트를 활용하여 특정 양자 회로를 수행 시 활용되는 시간 및 큐비트의 수에 대한 리소스에 영향을 줄 수 있다. 따라서, 특정 양자 회로를 수행 시 소요되는 시간 및 큐비트의 수에 대한 리소스는 인코딩된 논리적 큐비트를 효율적으로 배치하여 줄일 필요가 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 대한민국 공개특허 제10-2019-0070243호(2019.06.20. 공개)

발명의 내용

해결하려는 과제

[0008] 본 개시는 전술한 배경기술에 대응하여 안출된 것으로, 양자 논리적 큐비트의 논리적 CNOT 연산 방법을 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 몇몇 실시예에 따라, 양자 컴퓨팅 장치에 의해 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 방법으로서, 복수의 논리적 데이터 큐비트들(logical data qubits)을 가로열인 제 1 열로 배치하는 단계; 상기 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라

큐비트들(logical ancilla qubits)을 상기 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 상기 제 1 열과 상이한 가로열인 제 2 열로 배치하는 단계; 및 상기 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행하는 단계;를 포함할 수 있다.

[0011] 대안적으로, 상기 제 1 논리적 데이터 큐비트는 제어 큐비트(control qubit)이고, 그리고 상기 제 2 논리적 데이터 큐비트는 타겟 큐비트(target qubit)일 수 있다.

[0012] 대안적으로, 상기 복수의 논리적 엔실라 큐비트들은 각각 중간의 논리적 큐비트(intermediate logical qubit)를 포함할 수 있다.

[0013] 대안적으로, 상기 복수의 논리적 데이터 큐비트들 중 상기 제 1 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT를 수행하는 단계는, 상기 제 1 논리적 데이터 큐비트와 상기 제 2 논리적 데이터 큐비트 사이에 다른 논리적 데이터 큐비트가 존재하지 않는 경우, 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 1 병합 큐비트를 생성하는 단계; 상기 제 2 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트를 병합하여 제 2 병합 큐비트를 생성하는 단계; 상기 제 1 병합 큐비트를 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 엔실라 큐비트로 분할(splitting)하는 단계; 분할된 상기 제 1 논리적 엔실라 큐비트와 상기 제 2 병합 큐비트를 병합하여 제 3 병합 큐비트를 생성하는 단계; 및 상기 제 3 병합 큐비트를 상기 제 2 논리적 데이터 큐비트, 상기 제 1 논리적 엔실라 큐비트 및 상기 제 2 논리적 엔실라 큐비트로 분할하고, 그리고 상기 제 1 논리적 엔실라 큐비트 및 상기 제 2 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행하는 단계;를 포함할 수 있다.

[0014] 대안적으로, 상기 복수의 논리적 데이터 큐비트들 중 상기 제 1 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT를 수행하는 단계는, 상기 제 1 논리적 데이터 큐비트와 상기 제 2 논리적 데이터 큐비트 사이에 적어도 하나의 제 3 논리적 데이터 큐비트가 존재하는 경우, 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 4 병합 큐비트를 생성하는 단계; 상기 제 2 논리적 데이터 큐비트, 상기 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트 및 상기 적어도 하나의 제 3 논리적 데이터 큐비트에 대응되는 적어도 하나의 제 3 논리적 엔실라 큐비트를 병합하여 제 5 병합 큐비트를 생성하는 단계; 상기 제 4 병합 큐비트를 상기 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 엔실라 큐비트로 분할(splitting)하는 단계; 분할된 상기 제 1 논리적 엔실라 큐비트와 상기 제 5 병합 큐비트를 병합하여 제 6 병합 큐비트를 생성하는 단계; 및 상기 제 6 병합 큐비트를 상기 제 2 논리적 데이터 큐비트, 상기 제 1 논리적 엔실라 큐비트, 상기 제 2 논리적 엔실라 큐비트 및 상기 적어도 하나의 제 3 논리적 엔실라 큐비트로 분할하고, 그리고 초기화 시키는 트리밍(trimming)을 수행하는 단계;를 포함할 수 있다.

[0015] 대안적으로, 상기 복수의 논리적 데이터 큐비트들을 가로열인 상기 제 1 열로 배치하는 단계는, 상기 복수의 논리적 데이터 큐비트들을 수행 회로에서 관여되는 큐비트 연산의 개수에 기초하여 상기 제 1 열로 배치하는 단계;를 포함할 수 있다.

[0016] 대안적으로, 상기 복수의 논리적 데이터 큐비트들을 상기 수행 회로에서 관여되는 상기 큐비트 연산의 개수에 기초하여 상기 제 1 열로 배치하는 단계는, 상기 큐비트 연산의 개수에 기초하여 1순위로 결정된 제 4 논리적 데이터 큐비트를 상기 제 1 열의 중심부에 배치하는 단계; 상기 큐비트 연산의 개수에 기초하여 2순위로 결정된 제 5 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 4 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계; 및 상기 큐비트 연산의 개수에 따라 3순위로 결정된 제 6 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 4 논리적 데이터 큐비트의 타측에 대응되도록 배치하는 단계;를 포함할 수 있다.

[0017] 대안적으로, 상기 큐비트 연산의 개수에 따라 4순위로 결정된 제 7 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 6 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계; 및 상기 큐비트 연산의 개수에 따라 5순위로 결정된 제 8 논리적 데이터 큐비트를 상기 제 1 열 상에서 상기 제 5 논리적 데이터 큐비트의 일측에 대응되도록 배치하는 단계;를 더 포함하고, 상기 제 5 논리적 데이터 큐비트의 일측은 상기 제 4 논리적 데이터 큐비트의 일측에 대응되지 않는 부분이고, 그리고 상기 제 6 논리적 데이터 큐비트의 일측은 상기 제 4 논리적 데이터 큐비트의 타측에 대응되지 않는 부분일 수 있다.

[0018] 대안적으로, 복수의 논리적 CNOT들을 수행하는 경우, 스케줄링(scheduling) 함수를 포함하는 라우팅(routing) 알고리즘을 이용하여 상기 복수의 논리적 CNOT들의 연산의 순서를 결정하는 단계;를 더 포함할 수 있다.

[0019] 전술한 바와 같은 과제를 해결하기 위한 본 개시의 몇몇 다른 실시예에 따라, 양자 컴퓨팅 장치에 의해 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 방법으로서, 복수의 논리적 데이터 큐비트들(logical data qubits)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열로 배치하는 단계; 상기 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 상기 적어도 하나의 제 3 열과 상이한 가로열인 적어도 하나의 제 4 열로 배치하는 단계; 복수의 추가 논리적 엔실라 큐비트들을 상기 적어도 하나의 제 3 열 및 상기 적어도 하나의 제 4 열의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열로 배치하는 단계; 및 상기 복수의 논리적 데이터 큐비트들 중 제 9 논리적 데이터 큐비트 및 제 10 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행하는 단계;를 포함할 수 있다.

발명의 효과

[0020] 본 개시는 양자 논리적 큐비트의 논리적 CNOT 연산 방법을 통해 인접하지 않은 논리적 큐비트들 간에 위치 이동 없이 논리적 CNOT을 수행할 수 있다.

[0021] 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

[0022] 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 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법을 나타낸 도면이다.

도 2는 본 개시의 몇몇 실시예에 따른 양자 논리적 큐비트의 논리적 CNOT 연산을 제공하기 위한 양자 컴퓨팅 장치의 블록 구성도이다.

도 3은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산을 위한 레이아웃(layout)을 나타낸 도면이다.

도 4는 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 과정을 나타낸 도면이다.

도 5는 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 과정을 나타낸 도면이다.

도 6은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 복수의 논리적 데이터 큐비트들을 배치하는 방법을 나타낸 도면이다.

도 7은 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산을 위한 레이아웃(layout)을 나타낸 도면이다.

도 8은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법을 나타내는 도면이다.

도 9는 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법을 나타내는 도면이다.

도 10 및 도 11은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법과 종래의 기술들을 각 회로에 입력하여 출력된 결과를 나타내는 도면이다.

도 12는 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0023] 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.
- [0024] 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.
- [0025] 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.
- [0026] 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.
- [0027] 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.
- [0028] 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 안된다.
- [0029] 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.
- [0030] 본 개시내용에서의 제 1, 제 2, 또는 제 3 과 같이 제 N 으로 표현되는 용어들은 적어도 하나의 엔티티들을 구분하기 위해 사용된다. 예를 들어, 제 1 과 제 2로 표현된 엔티티들은 서로 동일하거나 또는 상이할 수 있다.
- [0031] 도 1은 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법을 나타낸 도면이다. 도 1(a) 내지 도 1(e)에 기재된 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법은 순차적으로 수행될 수 있다.
- [0032] 도 1(a)을 참조하면, 종래의 양자 컴퓨팅 장치는 Lattice Surgery에서 컨트롤 큐비트(control qubit)(1)과 타겟 큐비트(target qubit)(2)를 대각선으로 배치하고, 중간 논리적 큐비트(intermediate logical qubit)(3)를 추가로 배치할 수 있다. 여기서, 컨트롤 큐비트(1)는 $|1\rangle$, 타겟 큐비트(2)는 $|0\rangle$, 중간 논리적 큐비트(3)는 $|0\rangle$ 일 수 있다.

- [0033] 도 1(b)을 참조하면, 종래의 양자 컴퓨팅 장치는 타겟 큐비트(2)와 중간 논리 큐비트(3)를 병합(merging)하여 병합 큐비트(4)를 생성할 수 있다. 여기서, 병합은 Z-boundary merging을 포함할 수 있다. 여기서, 병합 큐비트(4)는 " $|0\rangle = |+\rangle + |-\rangle$ " 일 수 있다.
- [0034] 도 1(c)을 참조하면, 종래의 양자 컴퓨팅 장치는 병합 큐비트(4)를 타겟 큐비트(2)와 중간 논리 큐비트(3)로 분할(splitting)할 수 있다. 여기서, 분할은 Z-boundary splitting을 포함할 수 있다. 여기서, 컨트롤 큐비트(1)는 " $|1\rangle = |+\rangle - |-\rangle$ " 이고, 타겟 큐비트(2) 및 중간 논리 큐비트(3)는 " $|++\rangle + |--\rangle$ " 일 수 있다.
- [0035] 도 1(d)을 참조하면, 종래의 양자 컴퓨팅 장치는 컨트롤 큐비트(1)와 중간 논리 큐비트(3)를 병합하여 병합 큐비트(5)를 생성할 수 있다. 여기서 병합은 X-boundary splitting을 포함할 수 있다. 여기서, 컨트롤 큐비트(1) 및 타겟 큐비트(2)는 " $|++\rangle + |--\rangle - |+-\rangle - |-+\rangle = |11\rangle$ " 일 수 있다.
- [0036] 도 1(e)을 참조하면, 종래의 양자 컴퓨팅 장치는 병합 큐비트(5)를 분할하고, 중간 논리 큐비트(3)를 비활성화 시키는 트리밍(trimming)을 수행할 수 있다. 여기서, 컨트롤 큐비트(1) 및 타겟 큐비트(2)는 " $|++\rangle + |--\rangle - |+-\rangle - |-+\rangle = |11\rangle$ " 일 수 있다.
- [0037] 도 1(a) 내지 도 1(e)을 참조하면, 컨트롤 큐비트(1)가 $|1\rangle$, 타겟 큐비트(2)가 $|0\rangle$ 인 경우, 논리적 CNOT의 결과는 " $|11\rangle$ " 일 수 있다. 여기서, 타겟 큐비트(2)는 플립(flip)을 통해 " $|0\rangle$ "에서 " $|1\rangle$ "로 변화될 수 있다. 따라서, 논리적 큐비트들 간의 논리적 CNOT은 컨트롤 비트가 "0"인 경우 타겟 비트를 그대로 출력하고, 컨트롤 비트가 "1"인 경우 타겟 비트를 플립(flip)(예를 들어, "0"을 "1"로 변경, "1"을 "0"으로 변경 등)하는 CNOT 연산에 대응될 수 있다. 예를 들어, CNOT 연산은 컨트롤 비트가 "1"이고, 타겟 비트가 "0"인 경우, 타겟 비트를 "1"로 출력할 수 있다.
- [0038] 도 1을 참조하여 상술한 바와 같이, 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법은 컨트롤 큐비트와 타겟 큐비트가 서로 인접하고, 대각선으로 배치되어야 했다. 따라서, 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법은 컨트롤 큐비트와 타겟 큐비트가 인접해 있지 않은 경우, 즉 원거리에 배치되어 있는 경우에 서로의 위치를 바꿔주는 SWAP gate를 이용해야만 했다. 따라서, 종래의 논리적 큐비트들 간의 논리적 CNOT를 수행하는 방법은 SWAP gate를 이용하여 컨트롤 큐비트와 타겟 큐비트가 대각선으로 배치되도록 위치를 바꿔주는 과정으로 인하여 연산 시간이 오래 걸리고, 많은 리소스(resource)가 필요하여 비효율적일 수 있다.
- [0039] Z-boundary merging, Z-boundary splitting, X-boundary splitting, 트리밍(trimming), 논리적 CNOT 등의 Lattice Surgery와 관련된 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.
- [0040] 도 2는 본 개시의 몇몇 실시예에 따른 양자 논리적 큐비트의 논리적 CNOT 연산을 제공하기 위한 양자 컴퓨팅 장치의 블록 구성도이다.
- [0041] 도 2에 도시된 양자 컴퓨팅 장치의 구성은 간략화 하여 나타낸 예시일 뿐이다. 본 개시의 일 실시예에서 양자 컴퓨팅 장치는 양자 논리적 큐비트의 논리적 CNOT 연산 방법을 제공하기 위한 다른 구성들이 포함될 수 있고, 개시된 구성들 중 일부만이 컴퓨팅 장치를 구성할 수도 있다.
- [0042] 도 2를 참조하면, 양자 컴퓨팅 장치(100)는 양자 논리적 큐비트의 논리적 CNOT 연산을 수행할 수 있다.
- [0043] 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.

- [0044] 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.
- [0045] 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(Read-Only 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. 제 1 노드(100)는 인터넷(internet) 상에서 상기 메모리(130)의 저장 기능을 수행하는 웹 스토리지(web storage)와 관련되어 동작할 수도 있다. 전술한 메모리에 대한 기재는 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.
- [0047] 본 개시의 몇몇 실시예에 따르면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들(logical data qubits)을 가로열인 제 1 열로 배치할 수 있다. 복수의 논리적 데이터 큐비트들은 양자 알고리즘 또는 단일 변환의 대상이 되는 양자 회로에서 지정된 대로 동작하는 큐비트들일 수 있다.
- [0048] 프로세서(110)는 복수의 논리적 데이터 큐비트들을 수행 회로에서 관여되는 큐비트 연산의 개수에 기초하여 제 1 열로 배치할 수 있다.
- [0049] 예를 들어, 프로세서(110)는 큐비트 연산의 개수에 기초하여 1순위로 결정된 제 4 논리적 데이터 큐비트(a)를 제 1 열의 중심부에 배치할 수 있다. 프로세서(110)는 큐비트 연산의 개수에 기초하여 2순위로 결정된 제 5 논리적 데이터 큐비트(c)를 제 1 열 상에서 제 4 논리적 데이터 큐비트(a)의 일측에 대응되도록 배치할 수 있다. 프로세서(110)는 큐비트 연산의 개수에 따라 3순위로 결정된 제 6 논리적 데이터 큐비트(b)를 제 1 열 상에서 제 4 논리적 데이터 큐비트(a)의 타측에 대응되도록 배치할 수 있다. 프로세서(110)는 큐비트 연산의 개수에 따라 4순위로 결정된 제 7 논리적 데이터 큐비트(d)를 제 1 열 상에서 제 6 논리적 데이터 큐비트(b)의 일측에 대응되도록 배치할 수 있다. 프로세서(110)는 큐비트 연산의 개수에 따라 5순위로 결정된 제 8 논리적 데이터 큐비트(e)를 제 1 열 상에서 상기 제 5 논리적 데이터 큐비트(c)의 일측에 대응되도록 배치할 수 있다. 제 5 논리적 데이터 큐비트(c)의 일측은 제 4 논리적 데이터 큐비트(a)의 일측에 대응되지 않는 부분일 수 있다. 제 6 논리적 데이터 큐비트(b)의 일측은 제 4 논리적 데이터 큐비트(a)의 타측에 대응되지 않는 부분일 수 있다. 이에 대한 자세한 설명은 도 6을 참조하여 후술하도록 한다.
- [0051] 프로세서(110)는 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 제 1 열과 상이한 가로열인 제 2 열로 배치할 수 있다.
- [0052] 복수의 논리적 엔실라 큐비트들은 계산에서 특정 목표를 달성하기 위해 추가되는 큐비트들일 수 있다. 예를 들어, 복수의 논리적 엔실라 큐비트들은 양자 게이트에서 연산을 위해 배치되는 큐비트들일 수 있다.
- [0053] 복수의 논리적 엔실라 큐비트들은 각각 중간의 논리적 큐비트(intermediate logical qubit)를 포함할 수 있다. 중간의 논리적 큐비트는 복수의 논리적 데이터 큐비트들에 포함되는 제어 큐비트(control qubit)와 타겟 큐비트(target qubit) 간에 연산 시에 사용되는 큐비트일 수 있다. 제어 큐비트는 CNOT 게이트에서 논리적 큐비트를 인코딩하는데 사용되는 적어도 하나의 물리적 큐비트의 집합일 수 있다. 제어 큐비트는 CNOT 게이트에서 타겟 큐비트를 제어하는 큐비트일 수 있다. 예를 들어, 제어 큐비트는 CNOT 게이트에서 $|1\rangle$ 인 경우, 타겟 큐비트를 플립(예를 들어, 타겟 큐비트가 “ $|0\rangle$ ”인 경우, “ $|1\rangle$ ”로 변화)시킬 수 있다. 타겟 큐비트는 CNOT 게이트에서 제어 큐비트에 의해 변화되는 큐비트일 수 있다.
- [0055] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행할 수 있다.

- [0056] 제 1 논리적 데이터 큐비트는 제어 큐비트(control qubit)일 수 있다. 그리고 제 2 논리적 데이터 큐비트는 타겟 큐비트(target qubit)일 수 있다.
- [0057] 일 실시예에 따르면, 프로세서(110)는 제 1 논리적 데이터 큐비트와 제 2 논리적 데이터 큐비트 사이에 다른 논리적 데이터 큐비트가 존재하지 않는 경우, 제 1 논리적 데이터 큐비트와 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 1 병합 큐비트를 생성할 수 있다.
- [0058] 프로세서(110)는 제 2 논리적 데이터 큐비트 및 상기 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트를 병합하여 제 2 병합 큐비트를 생성할 수 있다.
- [0059] 프로세서(110)는 제 1 병합 큐비트를 제 1 논리적 데이터 큐비트와 제 1 논리적 엔실라 큐비트로 분할(splitting)할 수 있다.
- [0060] 프로세서(110)는 분할된 제 1 논리적 엔실라 큐비트와 제 2 병합 큐비트를 병합하여 제 3 병합 큐비트를 생성할 수 있다.
- [0061] 프로세서(110)는 제 3 병합 큐비트를 제 2 논리적 데이터 큐비트, 제 1 논리적 엔실라 큐비트 및 제 2 논리적 엔실라 큐비트로 분할하고, 그리고 제 1 논리적 엔실라 큐비트 및 제 2 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행할 수 있다. 이에 대한 자세한 설명은 도 4를 통해 후술하도록 한다.
- [0062] 다른 일 실시예에 따르면, 프로세서(110)는 제 1 논리적 데이터 큐비트와 제 2 논리적 데이터 큐비트 사이에 적어도 하나의 제 3 논리적 데이터 큐비트가 존재하는 경우, 제 1 논리적 데이터 큐비트와 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 4 병합 큐비트를 생성할 수 있다.
- [0063] 프로세서(110)는 제 2 논리적 데이터 큐비트, 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 데이터 큐비트에 대응되는 적어도 하나의 제 3 논리적 엔실라 큐비트를 병합하여 제 5 병합 큐비트를 생성할 수 있다.
- [0064] 프로세서(110)는 제 4 병합 큐비트를 제 1 논리적 데이터 큐비트와 제 1 논리적 엔실라 큐비트로 분할(splitting)할 수 있다.
- [0065] 프로세서(110)는 분할된 제 1 논리적 엔실라 큐비트와 제 5 병합 큐비트를 병합하여 제 6 병합 큐비트를 생성할 수 있다.
- [0066] 프로세서(110)는 제 6 병합 큐비트를 제 2 논리적 데이터 큐비트, 제 1 논리적 엔실라 큐비트, 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 엔실라 큐비트로 분할하고, 그리고 제 1 논리적 엔실라 큐비트, 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행할 수 있다. 이에 대한 자세한 설명은 도 5를 통해 후술하도록 한다.
- [0068] 프로세서(110)는 복수의 논리적 CNOT들을 수행하는 경우, 스케줄링(scheduling) 함수를 포함하는 라우팅(routing) 알고리즘을 이용하여 복수의 논리적 CNOT들의 연산의 순서를 결정할 수 있다.
- [0069] 라우팅 알고리즘은 동시에 수행될 수 없는 복수의 논리적 CNOT이 존재하는 경우, 스케줄링 함수를 호출하여 복수의 논리적 CNOT의 연산의 순서를 결정하는 알고리즘일 수 있다.
- [0070] 스케줄링 함수는 사전 결정된 우선 순위에 기초하여 연산의 순서를 결정하는 함수일 수 있다. 사전 결정된 우선 순위는 예상 연산 시간, 연산에 필요한 리소스 등에 기초하여 낮은 순 또는 높은 순으로 결정될 수 있다.
- [0071] 따라서, 프로세서(110)는 동시에 수행될 수 없는 복수의 논리적 CNOT이 존재하는 경우, 라우팅 알고리즘을 이용하여 복수의 논리적 CNOT의 연산의 순서를 결정할 수 있다. 프로세서(110)는 동시에 수행될 수 있는 복수의 논리적 CNOT을 동시에 수행할 수 있다.
- [0073] 본 개시의 몇몇 다른 실시예에 따르면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들(logical data qubits)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열로 배치할 수 있다.
- [0074] 프로세서(110)는 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical

ancilla qubits)을 적어도 하나의 제 3 열과 상이한 가로열인 적어도 하나의 제 4 열로 배치할 수 있다.

- [0075] 프로세서(110)는 복수의 추가 논리적 엔실라 큐비트들을 적어도 하나의 제 3 열 및 상기 적어도 하나의 제 4 열의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열로 배치할 수 있다.
- [0076] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 9 논리적 데이터 큐비트 및 제 10 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행할 수 있다. 이에 대한 자세한 설명은 도 7을 참조하여 후술하도록 한다.
- [0078] 도 3은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산을 위한 레이아웃(layout)을 나타낸 도면이다.
- [0079] 도 3을 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들(200)(예를 들어, 제 1 논리적 데이터 큐비트(210), 제 2 논리적 데이터 큐비트(220), 제 3 논리적 데이터 큐비트(230), 제 4 논리적 데이터 큐비트(240), 제 5 논리적 데이터 큐비트(250), 제 6 논리적 데이터 큐비트(260), 제 7 논리적 데이터 큐비트(270), 제 8 논리적 데이터 큐비트(280) 등)을 가로열인 제 1 열(10)로 배치할 수 있다.
- [0080] 또한, 프로세서(110)는 복수의 논리적 데이터 큐비트들(200)에 각각 대응되는 복수의 논리적 엔실라 큐비트들(300)(예를 들어, 제 1 논리적 엔실라 큐비트(310), 제 2 논리적 엔실라 큐비트(320), 제 3 논리적 엔실라 큐비트(330), 제 4 논리적 엔실라 큐비트(340), 제 5 논리적 엔실라 큐비트(350), 제 6 논리적 엔실라 큐비트(360), 제 7 논리적 엔실라 큐비트(370), 제 8 논리적 엔실라 큐비트(380) 등)을 가로열인 제 2 열(20)로 배치할 수 있다. 예를 들어, 프로세서(110)는 제 1 논리적 데이터 큐비트(210)의 하부에 제 1 논리적 엔실라 큐비트(310)를 배치할 수 있다. 도 3을 참조하여 상술한 바와 같은 레이아웃은 row type arch(r-arch)일 수 있다. 따라서, 프로세서(110)는 복수의 논리적 데이터 큐비트들(logical data qubits)을 가로열인 제 1 열로 배치하고, 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 제 1 열과 상이한 가로열인 제 2 열로 배치함으로써, row type arch로 배치할 수 있다.
- [0082] 도 4는 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 과정을 나타낸 도면이다.
- [0083] 도 4(a)를 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 제 1 논리적 데이터 큐비트(410)와 제 2 논리적 데이터 큐비트(420)를 가로열인 제 1 열에 배치하고, 제 1 논리적 엔실라 큐비트(510)와 제 2 논리적 엔실라 큐비트(520)를 가로열인 제 2 열에 배치할 수 있다.
- [0084] 도 4(b)를 참조하면, 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트(410)와 제 2 논리적 데이터 큐비트(420)에 대한 논리적 CNOT를 수행하는 경우, 제 1 논리적 데이터 큐비트(410)와 제 1 논리적 엔실라 큐비트(510)를 병합(merging)하여 제 1 병합 큐비트(610)를 생성할 수 있다. 프로세서(110)는 제 2 논리적 데이터 큐비트(420)와 제 2 논리적 엔실라 큐비트(520)를 병합하여 제 2 병합 큐비트(620)를 생성할 수 있다. 여기서, 병합은 X-boundary merging을 포함할 수 있다. 제 1 논리적 데이터 큐비트(410)는 제어 큐비트(control qubit)이고, 그리고 제 2 논리적 데이터 큐비트(420)는 타겟 큐비트(target qubit)일 수 있다.
- [0085] 도 4(c)를 참조하면, 프로세서(110)는 제 1 병합 큐비트(610)를 제 1 논리적 데이터 큐비트(410)와 제 1 논리적 엔실라 큐비트(510)로 분할(splitting)할 수 있다. 여기서 분할은 X-boundary splitting을 포함할 수 있다.
- [0086] 도 4(d)를 참조하면, 프로세서(110)는 분할된 제 1 논리적 엔실라 큐비트(510)와 제 2 병합 큐비트(620)를 병합하여 제 3 병합 큐비트(630)를 생성할 수 있다. 여기서, 병합은 Z-boundary merging을 포함할 수 있다.
- [0087] 도 4(e)를 참조하면, 프로세서(110)는 제 3 병합 큐비트(630)를 제 2 논리적 데이터 큐비트(420), 제 1 논리적 엔실라 큐비트(510), 제 2 논리적 엔실라 큐비트(520)로 분할할 수 있다. 프로세서(110)는 제 1 논리적 엔실라 큐비트(510), 제 2 논리적 엔실라 큐비트(520)를 초기화(initialize) 할 수 있다.

- [0089] 도 5는 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산 과정을 나타낸 도면이다.
- [0090] 도 5(a)를 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 제 1 논리적 데이터 큐비트(710), 제 2 논리적 데이터 큐비트(740), 제 3 논리적 데이터 큐비트(720), 제 4 논리적 데이터 큐비트(730)를 가로열인 제 1 열에 배치하고, 제 1 논리적 엔실라 큐비트(810), 제 2 논리적 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820), 제 4 논리적 엔실라 큐비트(830)를 가로열인 제 2 열에 배치할 수 있다.
- [0091] 도 5(b)를 참조하면, 제 1 논리적 데이터 큐비트(710)와 제 2 논리적 데이터 큐비트(740)에 다른 논리적 데이터 큐비트(제 3 논리적 데이터 큐비트(720) 및 제 4 논리적 데이터 큐비트(730))가 존재할 수 있다. 즉, 프로세서(110)는 인접한 논리적 데이터 큐비트들 간에 논리적 CNOT를 수행하는 것이 아닌, 원거리에 배치된 논리적 데이터 큐비트들 간에 논리적 CNOT를 수행할 수 있다.
- [0092] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트(710)와 제 2 논리적 데이터 큐비트(740)에 대한 논리적 CNOT를 수행하는 경우, 제 1 논리적 데이터 큐비트(710)와 제 1 논리적 엔실라 큐비트(810)를 병합(merging)하여 제 1 병합 큐비트(910)를 생성할 수 있다. 프로세서(110)는 제 2 논리적 데이터 큐비트(740), 제 2 논리적 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820) 및 제 4 논리적 엔실라 큐비트(830)를 병합하여 제 2 병합 큐비트(920)를 생성할 수 있다.
- [0093] 구체적으로, 프로세서(110)는 제 1 논리적 데이터 큐비트(710)와 제 1 논리적 엔실라 큐비트(810)를 X-boundary merging할 수 있다. 프로세서(110)는 제 2 논리적 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820) 및 제 4 논리적 엔실라 큐비트(830)를 Z-boundary merging하고, 그리고 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820) 및 제 4 논리적 엔실라 큐비트(830)를 Z-boundary merging한 큐비트와 제 2 논리적 데이터 큐비트(740)를 X-boundary merging할 수 있다. 여기서, 제 1 논리적 데이터 큐비트(710)는 제어 큐비트(control qubit)이고, 그리고 제 2 논리적 데이터 큐비트(740)는 타겟 큐비트(target qubit)일 수 있다.
- [0094] 도 5(c)를 참조하면, 프로세서(110)는 제 1 병합 큐비트(910)를 제 1 논리적 데이터 큐비트(710)와 제 1 논리적 엔실라 큐비트(810)로 분할(splitting)할 수 있다. 여기서 분할은 X-boundary splitting을 포함할 수 있다.
- [0095] 도 5(d)를 참조하면, 프로세서(110)는 제 1 논리적 엔실라 큐비트(810)와 제 2 병합 큐비트(920)를 병합하여 제 3 병합 큐비트(930)를 생성할 수 있다. 여기서, 병합은 Z-boundary merging을 포함할 수 있다.
- [0096] 도 5(e)를 참조하면, 제 3 병합 큐비트(930)를 제 2 논리적 데이터 큐비트(740), 제 1 논리적 엔실라 큐비트(810), 제 2 논리적 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820) 및 제 4 논리적 엔실라 큐비트(830)로 분할할 수 있다.
- [0097] 프로세서(110)는 제 1 논리적 엔실라 큐비트(810), 제 2 논리적 엔실라 큐비트(840), 제 3 논리적 엔실라 큐비트(820) 및 제 4 논리적 엔실라 큐비트(830)를 초기화(initialize) 할 수 있다.
- [0098] 프로세서(110)는 도 5를 참조하여 상술한 바와 같이, 논리적 큐비트의 레이아웃(layout)을 통해 lattice surgery 방식에 기반하여 논리적 CNOT를 수행할 수 있다. 프로세서(110)는 논리적 CNOT를 수행하는 논리적 큐비트들 사이에 다른 논리적 큐비트가 배치되어 있는 경우에도 SWAP 연산 추가 없이 CNOT를 수행할 수 있다. 따라서, 프로세서(110)는 추가적인 SWAP 연산이 필요하지 않으므로, 양자 오류정정부호가 양자 알고리즘을 수행하는 경우, 수행시간, 물리적 큐비트 수 및 수행시간의 곱인 양자 부피 리소스를 절감할 수 있다.
- [0100] 도 6은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치에서 복수의 논리적 데이터 큐비트들을 배치하는 방법을 나타낸 도면이다.
- [0101] 도 6(a)를 참조하면, 논리적 데이터 큐비트인 a의 2개 큐비트 연산자의 수가 5이고, 논리적 데이터 큐비트인 b의 2개의 큐비트 연산자의 수가 3이고, 논리적 데이터 큐비트인 c의 2개의 큐비트의 연산자의 수가 4이고, 논리적 데이터 큐비트인 d 및 e의 2개의 큐비트의 연산자의 수가 2임을 알 수 있다.
- [0102] 도 6(b)를 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들을 수행 회로에서 관여되는 큐비트 연산의 개수에 기초하여 제 1 열로 배치할 수 있다.
- [0103] 예를 들어, 프로세서(110)는 큐비트 연산의 개수에 기초하여 1순위로 결정된 a를 제 1 열의 중심부에 배치할 수

있다.

- [0104] 프로세서(110)는 큐비트 연산의 개수에 기초하여 2순위로 결정된 c를 제 1 열 상에서 a의 일측(예를 들어, a의 좌측)에 대응되도록 배치할 수 있다.
- [0105] 프로세서(110)는 큐비트 연산의 개수에 따라 3순위로 결정된 b를 제 1 열 상에서 a의 타측(예를 들어, a의 우측)에 대응되도록 배치할 수 있다.
- [0106] 프로세서(110)는 복수의 논리적 데이터 큐비트들의 큐비트 연산의 개수가 동일한 경우, 사전 설정된 방식으로 우선순위를 결정할 수 있다. 사전 설정된 방식은 무작위로 결정하는 방식, 사전 설정된 순서(예를 들어, 알파벳 순서, 숫자가 낮은 순서 등)에 따라 결정하는 방식 등을 포함할 수 있다. 예를 들어, 프로세서(110)는 동일한 큐비트 연산의 개수를 가지는 복수의 논리적 데이터 큐비트들 중에서 무작위(random)로 우선순위를 결정할 수 있다.
- [0107] 프로세서(110)는 사전 설정된 방식으로 결정된 우선순위에 따라 4순위로 결정된 d를 제 1 열 상에서 b의 일측(예를 들어, b의 우측)에 대응되도록 배치할 수 있다. b의 일측은 a의 타측(예를 들어, a의 우측)에 대응되지 않는 부분일 수 있다. 즉, b의 일측은 a와 맞닿는 b의 좌측이 아닌 b의 우측일 수 있다.
- [0108] 프로세서(110)는 사전 설정된 방식으로 결정된 우선순위에 따라 5순위로 결정된 e를 제 1 열 상에서 c의 일측(예를 들어, c의 좌측)에 대응되도록 배치할 수 있다. c의 일측은 a의 일측(예를 들어, a의 좌측)에 대응되지 않는 부분일 수 있다. 즉, c의 일측은 a와 맞닿는 c의 우측이 아닌 c의 좌측일 수 있다.
- [0110] 도 7은 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치에서 수행되는 양자 논리적 큐비트의 논리적 CNOT 연산을 위한 레이아웃(layout)을 나타낸 도면이다.
- [0111] 도 7을 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들(1110 내지 1180)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열(31, 32)로 배치할 수 있다.
- [0112] 프로세서(110)는 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(1210 내지 1280)을 적어도 하나의 제 3 열(31, 32)과 상이한 가로열인 적어도 하나의 제 4 열(41, 42)로 배치할 수 있다.
- [0113] 프로세서(110)는 복수의 추가 논리적 엔실라 큐비트들(1310 내지 1380)을 적어도 하나의 제 3 열(31, 32) 및 적어도 하나의 제 4 열(41, 42)의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열(51, 52)로 배치할 수 있다.
- [0114] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 복수(예를 들어, 2개)의 논리적 데이터 큐비트들에 대한 논리적 CNOT(controlled NOT)를 수행할 수 있다. 예를 들어, 프로세서(110)는 제 1 논리적 데이터 큐비트(1110) 및 제 2 논리적 데이터 큐비트(1120)에 대한 논리적 CNOT를 수행할 수 있다.
- [0115] 도 7을 참조하여 상술한 바와 같은 레이아웃은 modified row type arch(modi r-arch)일 수 있다. 따라서, 프로세서(110)는 복수의 논리적 데이터 큐비트들(logical data qubits)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열로 배치하고, 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 적어도 하나의 제 3 열과 상이한 가로열인 적어도 하나의 제 4 열로 배치하며, 복수의 추가 논리적 엔실라 큐비트들을 적어도 하나의 제 3 열 및 적어도 하나의 제 4 열의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열로 배치함으로써, modified row type arch 로 배치할 수 있다.
- [0117] 도 8은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법을 나타내는 도면이다.
- [0118] 도 8을 참조하면, 양자 컴퓨팅 장치(100)의 프로세서(110)는 복수의 논리적 데이터 큐비트들을 가로열인 제 1 열로 배치할 수 있다(S110).
- [0119] 프로세서(110)는 복수의 논리적 데이터 큐비트들을 수행 회로에서 관여되는 큐비트 연산의 개수에 기초하여 제 1 열로 배치할 수 있다.

- [0120] 프로세서(110)는 큐비트 연산의 개수에 기초하여 1순위로 결정된 제 4 논리적 데이터 큐비트를 제 1 열의 중심부에 배치할 수 있다.
- [0121] 프로세서(110)는 큐비트 연산의 개수에 기초하여 2순위로 결정된 제 5 논리적 데이터 큐비트를 제 1 열 상에서 제 4 논리적 데이터 큐비트의 일측에 대응되도록 배치할 수 있다.
- [0122] 프로세서(110)는 큐비트 연산의 개수에 따라 3순위로 결정된 제 6 논리적 데이터 큐비트를 제 1 열 상에서 제 4 논리적 데이터 큐비트의 타측에 대응되도록 배치할 수 있다.
- [0123] 프로세서(110)는 큐비트 연산의 개수에 따라 4순위로 결정된 제 7 논리적 데이터 큐비트를 제 1 열 상에서 제 6 논리적 데이터 큐비트의 일측에 대응되도록 배치할 수 있다.
- [0124] 프로세서(110)는 큐비트 연산의 개수에 따라 5순위로 결정된 제 8 논리적 데이터 큐비트를 제 1 열 상에서 제 5 논리적 데이터 큐비트의 일측에 대응되도록 배치할 수 있다.
- [0125] 제 5 논리적 데이터 큐비트의 일측은 제 4 논리적 데이터 큐비트의 일측에 대응되지 않는 부분이고, 그리고 제 6 논리적 데이터 큐비트의 일측은 제 4 논리적 데이터 큐비트의 타측에 대응되지 않는 부분일 수 있다.
- [0127] 프로세서(110)는 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들을 복수의 논리적 데이터 큐비트들의 각각에 대응되도록 제 1 열과 상이한 가로열인 제 2 열로 배치할 수 있다(S120).
- [0128] 복수의 논리적 엔실라 큐비트들은 각각 중간 논리적 큐비트(intermediate logical qubit)를 포함할 수 있다.
- [0129] 프로세서(110)는 제 1 논리적 데이터 큐비트와 제 2 논리적 데이터 큐비트 사이에 다른 논리적 데이터 큐비트가 존재하지 않는 경우, 제 1 논리적 데이터 큐비트와 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 1 병합 큐비트를 생성할 수 있다.
- [0130] 프로세서(110)는 제 2 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트를 병합하여 제 2 병합 큐비트를 생성할 수 있다.
- [0131] 프로세서(110)는 제 1 병합 큐비트를 제 1 논리적 데이터 큐비트와 제 1 논리적 엔실라 큐비트로 분할(splitting)할 수 있다.
- [0132] 프로세서(110)는 분할된 제 1 논리적 엔실라 큐비트와 제 2 병합 큐비트를 병합하여 제 3 병합 큐비트를 생성할 수 있다.
- [0133] 프로세서(110)는 제 3 병합 큐비트를 제 2 논리적 데이터 큐비트, 제 1 논리적 엔실라 큐비트 및 제 2 논리적 엔실라 큐비트로 분할하고, 그리고 제 1 논리적 엔실라 큐비트 및 제 2 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행할 수 있다.
- [0134] 일 실시예에서, 프로세서(110)는 제 1 논리적 데이터 큐비트와 제 2 논리적 데이터 큐비트 사이에 적어도 하나의 제 3 논리적 데이터 큐비트가 존재하는 경우, 제 1 논리적 데이터 큐비트와 상기 제 1 논리적 데이터 큐비트에 대응되는 제 1 논리적 엔실라 큐비트를 병합(merging)하여 제 4 병합 큐비트를 생성할 수 있다.
- [0135] 프로세서(110)는 제 2 논리적 데이터 큐비트, 제 2 논리적 데이터 큐비트에 대응되는 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 데이터 큐비트에 대응되는 적어도 하나의 제 3 논리적 엔실라 큐비트를 병합하여 제 5 병합 큐비트를 생성할 수 있다.
- [0136] 프로세서(110)는 제 4 병합 큐비트를 제 1 논리적 데이터 큐비트와 제 1 논리적 엔실라 큐비트로 분할(splitting)할 수 있다.
- [0137] 프로세서(110)는 분할된 제 1 논리적 엔실라 큐비트와 제 5 병합 큐비트를 병합하여 제 6 병합 큐비트를 생성할 수 있다.
- [0138] 프로세서(110)는 제 6 병합 큐비트를 제 2 논리적 데이터 큐비트, 제 1 논리적 엔실라 큐비트, 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 엔실라 큐비트로 분할하고, 그리고 제 1 논리적 엔실라 큐비트, 제 2 논리적 엔실라 큐비트 및 적어도 하나의 제 3 논리적 엔실라 큐비트를 비활성화 시키는 트리밍(trimming)을 수행할 수 있다.

- [0140] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 1 논리적 데이터 큐비트 및 제 2 논리적 데이터 큐비트에 대한 논리적 CNOT를 수행할 수 있다(S130).
- [0141] 제 1 논리적 데이터 큐비트는 제어 큐비트(control qubit)이고, 그리고 제 2 논리적 데이터 큐비트는 타겟 큐비트(target qubit)일 수 있다.
- [0142] 프로세서(110)는 복수의 논리적 CNOT들을 수행하는 경우, 스케줄링(scheduling) 함수를 포함하는 라우팅(routing) 알고리즘을 이용하여 복수의 논리적 CNOT들의 연산의 순서를 결정할 수 있다.
- [0144] 도 9는 본 개시의 몇몇 다른 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법을 나타내는 도면이다.
- [0145] 프로세서(110)는 복수의 논리적 데이터 큐비트들(logical data qubits)을 사전 결정된 개수의 논리적 데이터 큐비트들을 포함하는, 가로열인 적어도 하나의 제 3 열로 배치할 수 있다.
- [0146] 프로세서(110)는 복수의 논리적 데이터 큐비트들에 각각 대응되는 복수의 논리적 엔실라 큐비트들(logical ancilla qubits)을 적어도 하나의 제 3 열과 상이한 가로열인 적어도 하나의 제 4 열로 배치할 수 있다.
- [0147] 프로세서(110)는 복수의 추가 논리적 엔실라 큐비트들을 적어도 하나의 제 3 열 및 적어도 하나의 제 4 열의 우측 끝단 또는 좌측 끝단 중 적어도 하나에 대응되도록, 세로열인 적어도 하나의 제 5 열로 배치할 수 있다.
- [0148] 프로세서(110)는 복수의 논리적 데이터 큐비트들 중 제 9 논리적 데이터 큐비트 및 제 10 논리적 데이터 큐비트에 대한 논리적 CNOT(controlled NOT)를 수행할 수 있다.
- [0149] 도 8 및 도 9에 도시되는 단계들은 예시적인 단계들이다. 따라서, 본 개시 내용의 사상 범위를 벗어나지 않는 한도에서 도 8 및 도 9의 단계들 중 일부가 생략되거나 추가적인 단계들이 존재할 수 있다는 점 또한 당업자에게 명백할 것이다. 또한, 도 8 및 도 9에 기재된 구성들(예를 들어, 양자 컴퓨팅 장치(100))에 관한 구체적인 내용은 앞서 도 1 내지 도 7을 통해 설명한 내용으로 대체될 수 있다.
- [0150] 도 1 내지 도 9를 참조하여 상술한 바와 같이, 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치(100)는 논리적 큐비트의 레이아웃(layout)을 통해 lattice surgery 방식에 기반하여 논리적 CNOT를 수행할 수 있다.
- [0151] 양자 컴퓨팅 장치(100)는 논리적 CNOT를 수행하는 논리적 큐비트들 사이에 다른 논리적 큐비트가 배치되어 있는 경우에도 SWAP 연산 추가 없이 CNOT를 수행할 수 있다. 따라서, 양자 컴퓨팅 장치(100)는 추가적인 SWAP 연산이 필요하지 않으므로, 양자 오류정정부호가 양자 알고리즘을 수행하는 경우, 수행시간, 물리적 큐비트 수 및 수행시간의 곱인 양자 부피 리소스를 절감할 수 있다.
- [0152] 양자 컴퓨팅 장치(100)는 스케줄링(scheduling) 함수를 포함하는 라우팅(routing) 알고리즘을 이용하여 복수의 논리적 CNOT들의 연산의 순서를 결정하여 연산의 속도를 높일 수 있다.
- [0154] 도 10 및 도 11은 본 개시의 몇몇 실시예에 따른 양자 컴퓨팅 장치를 통해 양자 논리적 큐비트의 논리적 CNOT 연산을 수행하는 방법과 종래의 기술들을 각 회로에 입력하여 출력된 결과를 나타내는 도면이다.
- [0155] 큐비트 효율성(qubit efficiency)은 예약된 양자 회로에 포함된 논리적 데이터 큐비트 수와 모든 논리적 큐비트 수의 비율일 수 있다.
- [0156] 회로 지연(circuit latency)은 신드롬 측정 주기들(syndrome measurement cycles)에서 양자 회로의 실행 시간일 수 있다. 코드 거리 d 는 회로 지연이 낮으면 줄어들 수 있으며, 이는 더 적은 수의 물리적 큐비트를 소비하게 할 수 있다.
- [0157] 양자 볼륨(quantum volume)은 물리적 큐비트 수에 회로 지연 시간을 곱한 값일 수 있다.
- [0158] $[[7,1,3]]$ 및 $[[15,1,3]]$ 인코딩 회로는 각각 $|Y\rangle_L$ 상태 증류(state distillation) 및 $|A\rangle_L$ 상태 증류에 사용될 수 있다. Shor(15) 회로는 휴리스틱 방식(heuristically)으로 구현될 수 있다. Toffoli 게이트는 Hadamard, CNOT 및 T 게이트로 분해될 수 있다. Adder0-5와 Adder1-8은 두 가지 유형의 가산기(adder)일 수 있다

다. 아치 크기는 각 레이아웃 RXC의 크기이며, 여기서 R과 C는 각각 행과 열의 논리적 데이터 큐비트 수일 수 있다.

[0159] 각 회로에 대한 구체적인 사항은 아래의 표 1에 기재된 내용에 대응될 수 있다.

표 1

Benchmarks	#Qubits	#Gates	#CNOT	#CNOT/#Gates	t-, c- arch size	Modified r- arch size
[[7,1,3]]	7	21	11	52.38%	3X3	2X4
[[15,1,3]]	15	53	34	64.15%	4X4	3X5
Shor15(a=7)	6	231	90	38.96%	3X3	2X3
Adder0-5	16	306	128	41.83%	4X4	4X4
Adder1-8	18	291	129	44.33%	5X5	4X5

[0161] 큐비트 효율성은 아래의 표 2에 기재된 내용에 대응될 수 있다.

표 2

	[[7,1,3]]	[[15,1,3]]	Shor(15)	Adder0-5	Adder1-8
r-arch	7/14	15/30	6/12	16/32	18/36
Modified r-arch	7/20	15/42	6/16	16/56 or 16/48	18/56
c-arch	7/18	15/32	6/18	16/32	18/50
t-arch	7/36	15/64	6/36	16/64	18/100

[0164] 표 2를 참조하면, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)과 종래의 방법(t-arch, c-arch)을 각 회로에서 테스트한 결과, r-arch의 큐비트 효율성은 모든 벤치마크에 대해 1/2이며, c-arch 및 t-arch 보다 각각 최대 16% 및 33% 높게 나타난 것을 알 수 있다. modi r-arch의 효율성 값은 1/2보다 작을 수 있다. modi r-arch는 모든 벤치마크에서 t-arch보다 8~20% 더 나은 효율성을 보여줄 수 있다.

[0165] 도 10을 참조하면, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)과 종래의 방법(t-arch, c-arch)을 각 회로에서 테스트한 결과, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)이 [[7,1,3]], Shor(15), Adder0-5 및 Adder1-8에서 종래의 방법(t-arch, c-arch)보다 회로 지연(circuit latency)이 줄어드는 것을 확인할 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)이 [[7,1,3]], Shor(15), Adder0-5 및 Adder1-8에서 종래의 방법(t-arch, c-arch)보다 양자 회로의 수행 시간이 적게 들어 효율적일 수 있다.

[0166] 도 11을 참조하면, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)과 종래의 방법(t-arch, c-arch)을 각 회로에서 테스트한 결과, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)이 모든 회로에 종래의 방법(t-arch, c-arch)보다 양자 볼륨이 적은 것을 확인할 수 있다. 따라서, 본 개시의 몇몇 실시예에 따른 방법(r-arch, modi r-arch)이 [[7,1,3]], Shor(15), Adder0-5 및 Adder1-8에서 종래의 방법(t-arch, c-arch)보다 적은 리소스를 활용하여 양자 회로를 수행할 수 있어 효율적일 수 있다.

[0168] 도 12는 본 개시의 내용의 실시예들이 구현될 수 있는 예시적인 컴퓨팅 환경에 대한 일반적인 개략도이다.

[0169] 본 개시가 일반적으로 컴퓨팅 장치에 의해 구현될 수 있는 것으로 진술되었지만, 당업자라면 본 개시가 하나 이상의 컴퓨터 상에서 실행될 수 있는 컴퓨터 실행가능 명령어 및/또는 기타 프로그램 모듈들과 결합되어 및/또는 하드웨어와 소프트웨어의 조합으로써 구현될 수 있다는 것을 잘 알 것이다.

[0170] 일반적으로, 프로그램 모듈은 특정의 태스크를 수행하거나 특정의 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조, 기타 등등을 포함한다. 또한, 당업자라면 본 개시의 방법이 단일-프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터는 물론 퍼스널 컴퓨터, 핸드헬드(handheld) 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.

[0171] 본 개시의 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 둘 다에 위치할 수 있다.

[0172] 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. 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.

[0173] 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 데이터 등을 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.

[0174] 컴퓨터(1102)를 포함하는 본 개시의 여러가지 측면들을 구현하는 예시적인 환경(1100)이 나타내어져 있으며, 컴퓨터(1102)는 처리 장치(1104), 시스템 메모리(1106) 및 시스템 버스(1108)를 포함한다. 시스템 버스(1108)는 시스템 메모리(1106)(이에 한정되지 않음)을 비롯한 시스템 컴포넌트들을 처리 장치(1104)에 연결시킨다. 처리 장치(1104)는 다양한 상용 프로세서들 중 임의의 프로세서일 수 있다. 듀얼 프로세서 및 기타 멀티프로세서 아키텍처도 역시 처리 장치(1104)로서 이용될 수 있다.

[0175] 시스템 버스(1108)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(1106)는 판독 전용 메모리(ROM)(1110) 및 랜덤 액세스 메모리(RAM)(1112)를 포함한다. 기본 입/출력 시스템(BIOS)은 ROM, EPROM, EEPROM 등의 비휘발성 메모리(1110)에 저장되며, 이 BIOS는 시동 중과 같은 때에 컴퓨터(1102) 내의 구성요소들 간에 정보를 전송하는 일을 돕는 기본적인 루틴을 포함한다. RAM(1112)은 또한 데이터를 캐싱하기 위한 정적 RAM 등의 고속 RAM을 포함할 수 있다.

[0176] 컴퓨터(1102)는 또한 내장형 하드 디스크 드라이브(HDD)(1114)(예를 들어, EIDE, SATA)-이 내장형 하드 디스크 드라이브(1114)는 또한 적당한 새시(도시 생략) 내에서 외장형 용도로 구성될 수 있음-, 자기 플로피 디스크 드라이브(FDD)(1116)(예를 들어, 이동식 디스켓(1118)으로부터 판독을 하거나 그에 기록을 하기 위한 것임), 및 광 디스크 드라이브(1120)(예를 들어, CD-ROM 디스크(1122)를 판독하거나 DVD 등의 기타 고용량 광 매체로부터 판독을 하거나 그에 기록을 하기 위한 것임)를 포함한다. 하드 디스크 드라이브(1114), 자기 디스크 드라이브(1116) 및 광 디스크 드라이브(1120)는 각각 하드 디스크 드라이브 인터페이스(1124), 자기 디스크 드라이브 인터페이스(1126) 및 광 드라이브 인터페이스(1128)에 의해 시스템 버스(1108)에 연결될 수 있다. 외장형 드라이브 구현을 위한 인터페이스(1124)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술 중 적어도 하나 또는 그 둘 다를 포함한다.

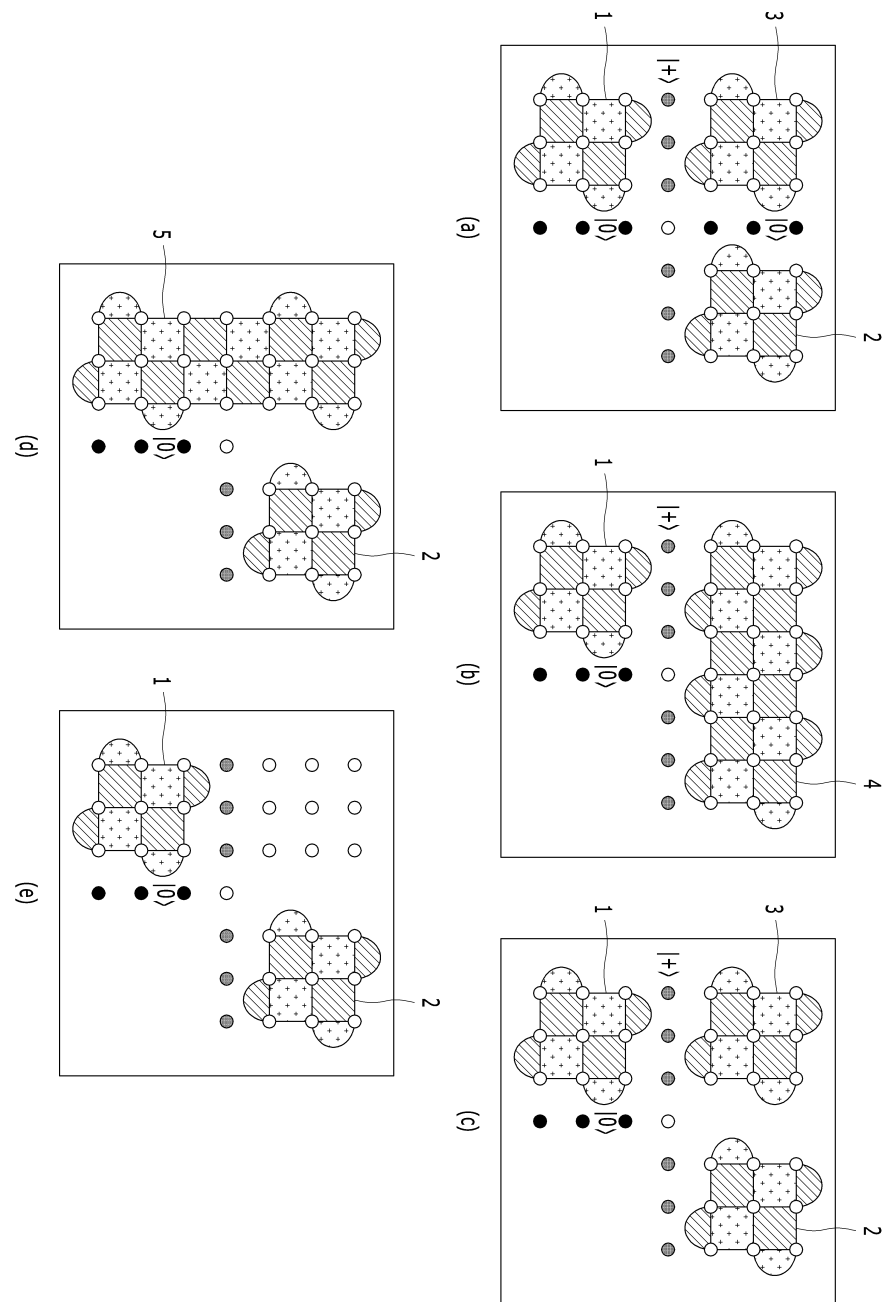
[0177] 이들 드라이브 및 그와 연관된 컴퓨터 판독가능 매체는 데이터, 데이터 구조, 컴퓨터 실행가능 명령어, 기타 등등의 비휘발성 저장을 제공한다. 컴퓨터(1102)의 경우, 드라이브 및 매체는 임의의 데이터를 적당한 디지털 형식으로 저장하는 것에 대응한다. 상기에서의 컴퓨터 판독가능 매체에 대한 설명이 HDD, 이동식 자기 디스크, 및 CD 또는 DVD 등의 이동식 광 매체를 언급하고 있지만, 당업자라면 zip 드라이브(zip drive), 자기 카세트, 플래쉬 메모리 카드, 카트리지, 기타 등등의 컴퓨터에 의해 판독가능한 다른 유형의 매체도 역시 예시적인 운영 환경에서 사용될 수 있으며 또 임의의 이러한 매체가 본 개시의 방법들을 수행하기 위한 컴퓨터 실행가능 명령어를 포함할 수 있다는 것을 잘 알 것이다.

- [0178] 운영 체제(1130), 하나 이상의 애플리케이션 프로그램(1132), 기타 프로그램 모듈(1134) 및 프로그램 데이터(1136)를 비롯한 다수의 프로그램 모듈이 드라이브 및 RAM(1112)에 저장될 수 있다. 운영 체제, 애플리케이션, 모듈 및/또는 데이터의 전부 또는 그 일부가 또한 RAM(1112)에 캐싱될 수 있다. 본 개시가 여러가지 상업적으로 이용가능한 운영 체제 또는 운영 체제들의 조합에서 구현될 수 있다는 것을 잘 알 것이다.
- [0179] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들어, 키보드(1138) 및 마우스(1140) 등의 포인팅 장치를 통해 컴퓨터(1102)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(1108)에 연결되어 있는 입력 장치 인터페이스(1142)를 통해 처리 장치(1104)에 연결되지만, 병렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스, 기타 등등의 기타 인터페이스에 의해 연결될 수 있다.
- [0180] 모니터(1144) 또는 다른 유형의 디스플레이 장치도 역시 비디오 어댑터(1146) 등의 인터페이스를 통해 시스템 버스(1108)에 연결된다. 모니터(1144)에 부가하여, 컴퓨터는 일반적으로 스피커, 프린터, 기타 등등의 기타 주변 출력 장치(도시 생략)를 포함한다.
- [0181] 컴퓨터(1102)는 유선 및/또는 무선 통신을 통한 원격 컴퓨터(들)(1148) 등의 하나 이상의 원격 컴퓨터로의 논리적 연결을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1148)는 워크스테이션, 컴퓨팅 디바이스 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서-기반 오락 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있으며, 일반적으로 컴퓨터(1102)에 대해 기술된 구성요소들 중 다수 또는 그 전부를 포함하지만, 간략함을 위해, 메모리 저장 장치(1150)만이 도시되어 있다. 도시되어 있는 논리적 연결은 근거리 통신망(LAN)(1152) 및/또는 더 큰 네트워크, 예를 들어, 원격 통신망(WAN)(1154)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.
- [0182] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 유선 및/또는 무선 통신 네트워크 인터페이스 또는 어댑터(1156)를 통해 로컬 네트워크(1152)에 연결된다. 어댑터(1156)는 LAN(1152)에의 유선 또는 무선 통신을 용이하게 해줄 수 있으며, 이 LAN(1152)은 또한 무선 어댑터(1156)와 통신하기 위해 그에 설치되어 있는 무선 액세스 포인트를 포함하고 있다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1102)는 모뎀(1158)을 포함할 수 있거나, WAN(1154) 상의 통신 컴퓨팅 디바이스에 연결되거나, 또는 인터넷을 통하는 등, WAN(1154)을 통해 통신을 설정하는 기타 수단을 갖는다. 내장형 또는 외장형 및 유선 또는 무선 장치일 수 있는 모뎀(1158)은 직렬 포트 인터페이스(1142)를 통해 시스템 버스(1108)에 연결된다. 네트워킹된 환경에서, 컴퓨터(1102)에 대해 설명된 프로그램 모듈들 또는 그의 일부가 원격 메모리/저장 장치(1150)에 저장될 수 있다. 도시된 네트워크 연결이 예시적인 것이며 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 잘 알 것이다.
- [0183] 컴퓨터(1102)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 임의의 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.
- [0184] 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 5GHz 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.
- [0185] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 정보 및 신호들이 임의의 다양한 상이한 기술들 및 기법들을 이용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 위의 설명에서 참조될 수 있는 데이터, 지시들, 명령들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 입자들, 광학장들 또는 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.

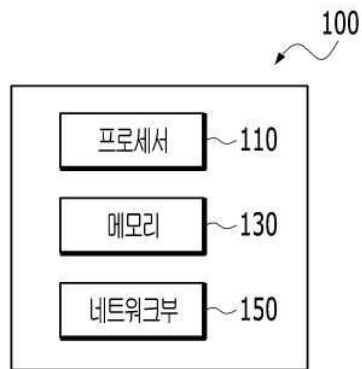
- [0186] 본 개시의 기술 분야에서 통상의 지식을 가진 자는 여기에 개시된 실시예들과 관련하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 프로세서들, 수단들, 회로들 및 알고리즘 단계들이 전자 하드웨어, (편의를 위해, 여기에서 소프트웨어로 지칭되는) 다양한 형태들의 프로그램 또는 설계 코드 또는 이들 모두의 결합에 의해 구현될 수 있다는 것을 이해할 것이다. 하드웨어 및 소프트웨어의 이러한 상호 호환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 이들의 기능과 관련하여 위에서 일반적으로 설명되었다. 이러한 기능이 하드웨어 또는 소프트웨어로서 구현되는지 여부는 특정한 애플리케이션 및 전체 시스템에 대하여 부과되는 설계 제약들에 따라 좌우된다. 본 개시의 기술 분야에서 통상의 지식을 가진 자는 각각의 특정한 애플리케이션에 대하여 다양한 방식으로 설명된 기능을 구현할 수 있으나, 이러한 구현 결정들은 본 개시의 범위를 벗어나는 것으로 해석되어서는 안 될 것이다.
- [0187] 여기서 제시된 다양한 실시예들은 방법, 장치, 또는 표준 프로그래밍 및/또는 엔지니어링 기술을 사용한 제조 물품(article)으로 구현될 수 있다. 용어 제조 물품은 임의의 컴퓨터-판독가능 저장장치로부터 액세스 가능한 컴퓨터 프로그램, 캐리어, 또는 매체(media)를 포함한다. 예를 들어, 컴퓨터-판독가능 저장매체는 자기 저장 장치(예를 들면, 하드 디스크, 플로피 디스크, 자기 스트림, 등), 광학 디스크(예를 들면, CD, DVD, 등), 스마트 카드, 및 플래쉬 메모리 장치(예를 들면, EEPROM, 카드, 스틱, 키 드라이브, 등)를 포함하지만, 이들로 제한되는 것은 아니다. 또한, 여기서 제시되는 다양한 저장 매체는 정보를 저장하기 위한 하나 이상의 장치 및/또는 다른 기계-판독가능한 매체를 포함한다.
- [0188] 제시된 프로세스들에 있는 단계들의 특정한 순서 또는 계층 구조는 예시적인 접근들의 일례임을 이해하도록 한다. 설계 우선순위들에 기반하여, 본 개시의 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.
- [0189] 제시된 실시예들에 대한 설명은 임의의 본 개시의 기술 분야에서 통상의 지식을 가진 자가 본 개시를 이용하거나 또는 실시할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 본 개시의 기술 분야에서 통상의 지식을 가진 자에게 명백할 것이며, 여기에 정의된 일반적인 원리들은 본 개시의 범위를 벗어남이 없이 다른 실시예들에 적용될 수 있다. 그리하여, 본 개시는 여기에 제시된 실시예들로 한정되는 것이 아니라, 여기에 제시된 원리들 및 신규한 특징들과 일관되는 최광의의 범위에서 해석되어야 할 것이다.

도면

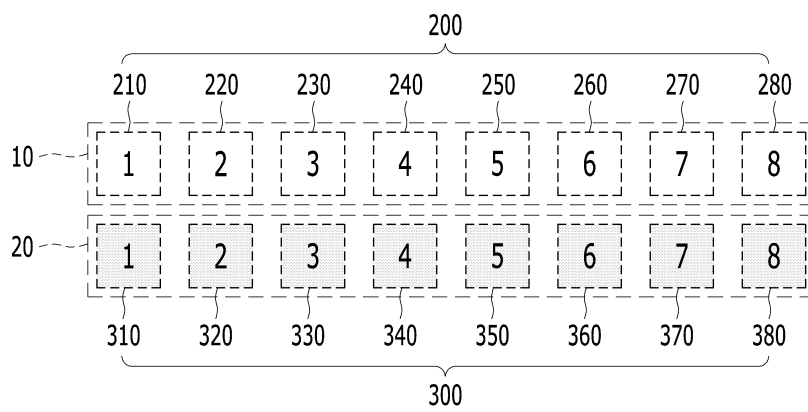
도면1



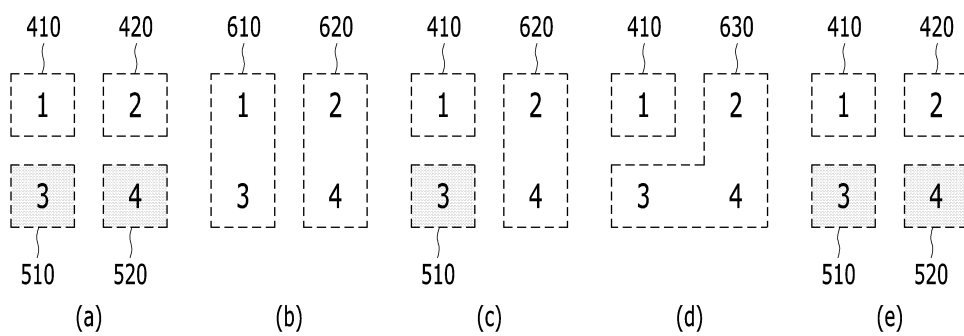
도면2



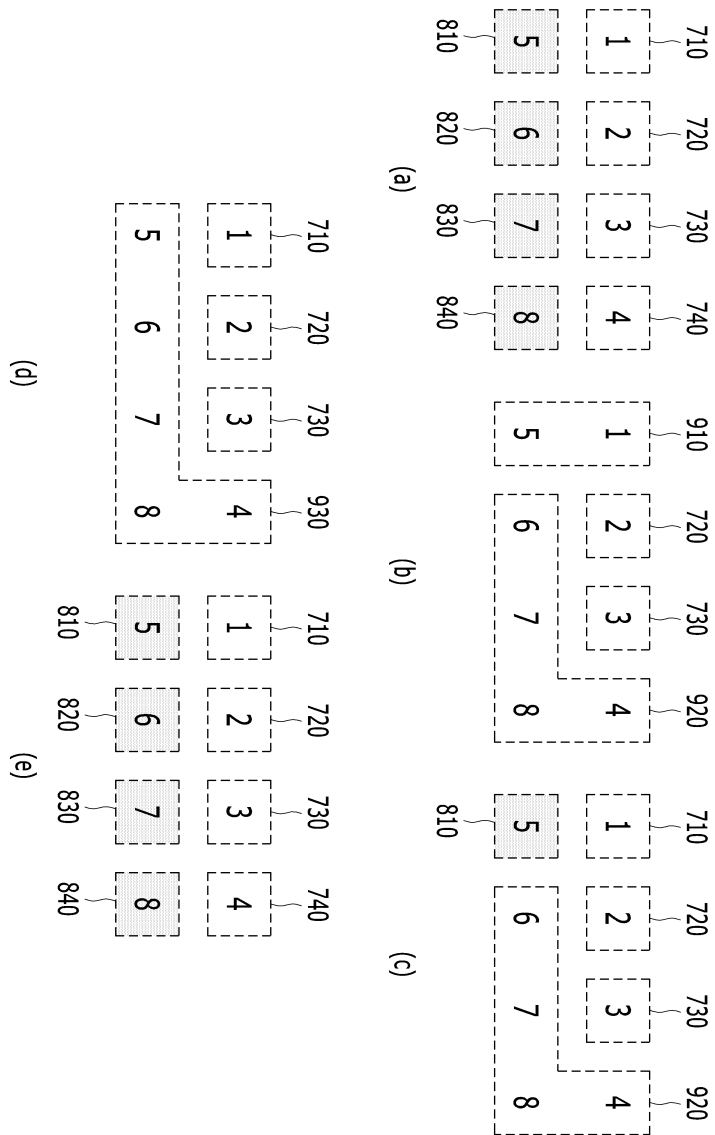
도면3



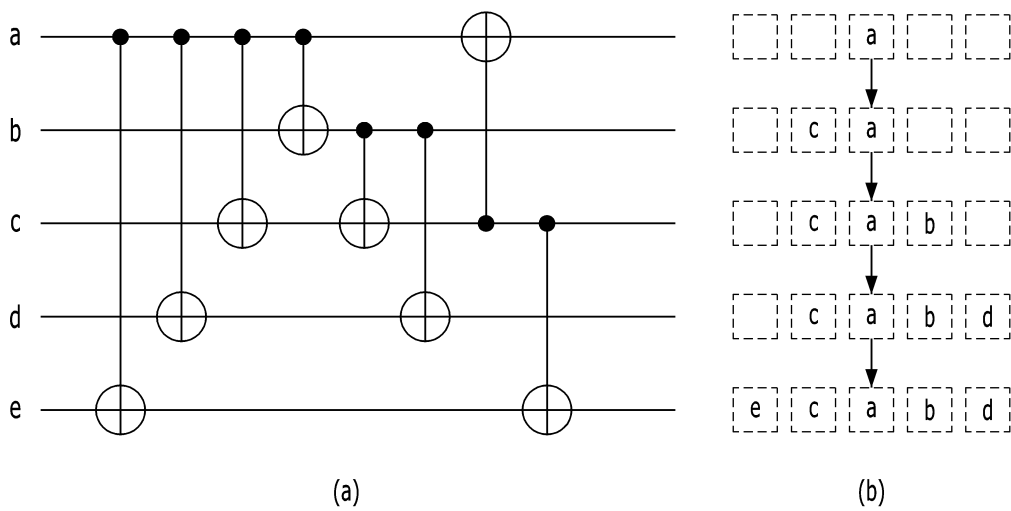
도면4



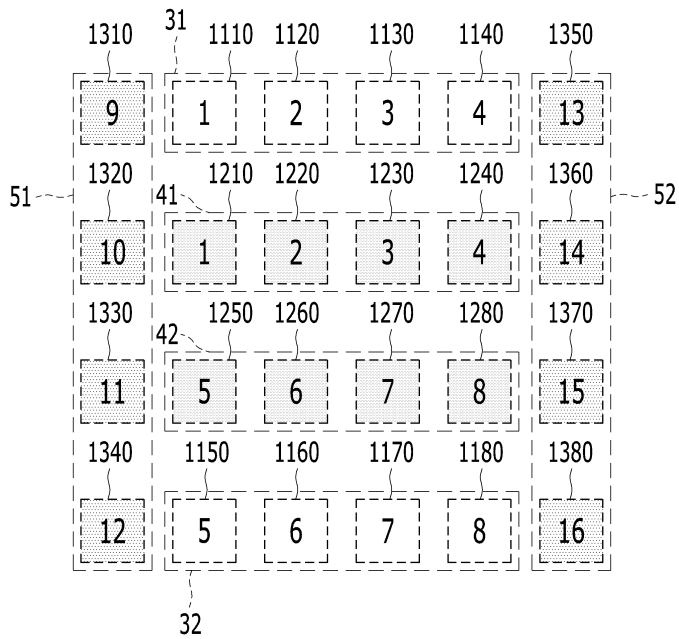
도면5



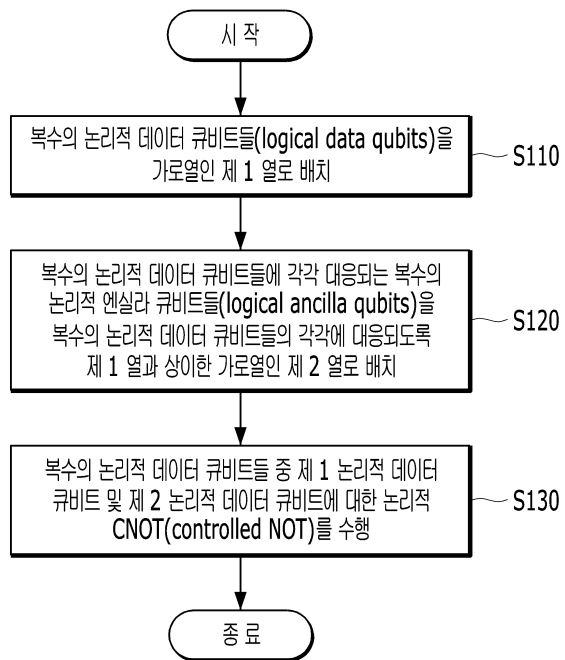
도면6



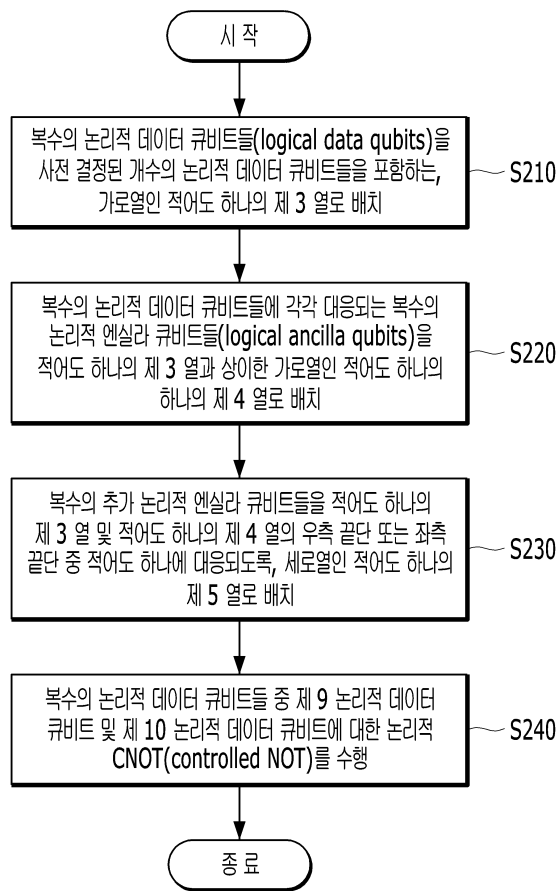
도면7



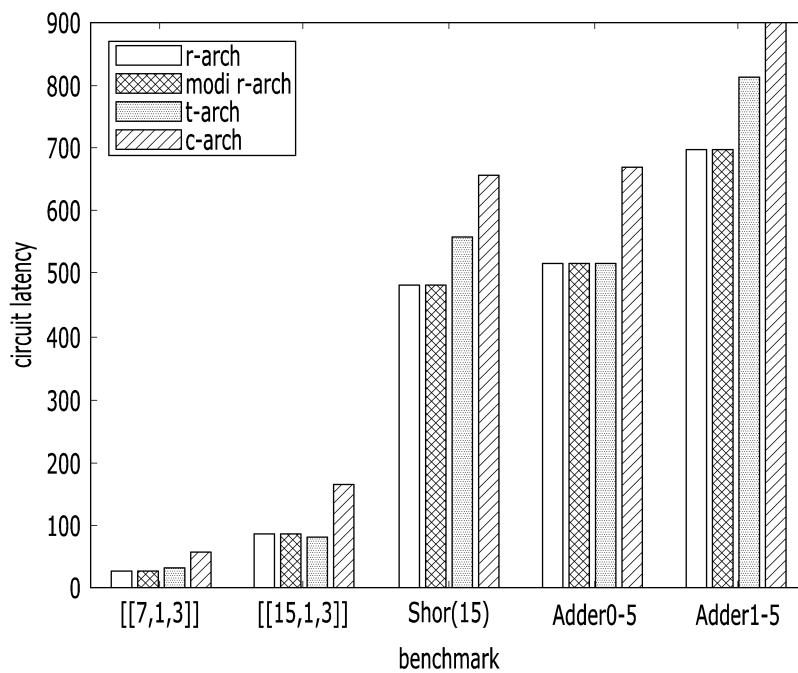
도면8



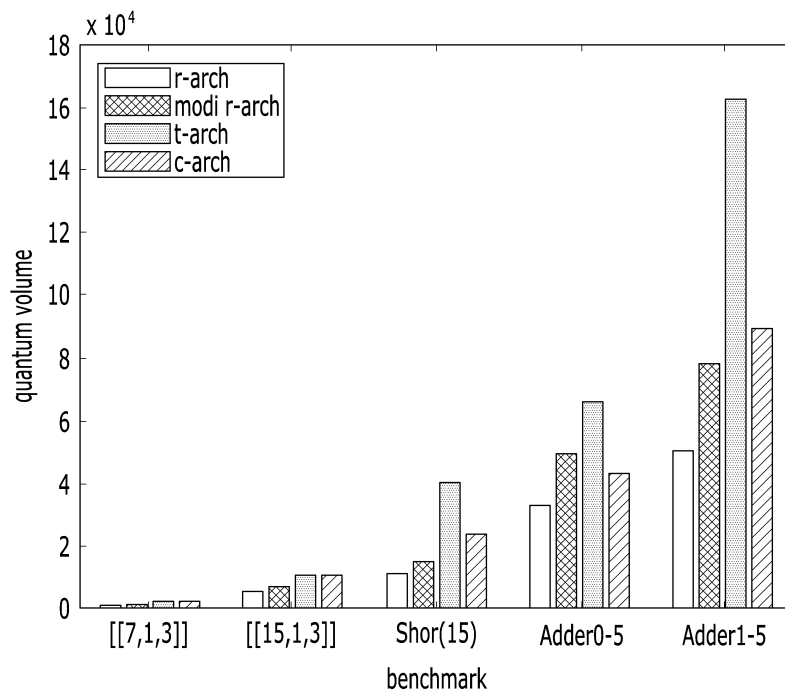
도면9



도면10



도면11



도면12

