(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2023년09월06일
(11) 등록번호 10-2574250
(24) 등록일자 2023년08월30일

(51) 국제특허분류(Int. Cl.)
H03M 13/11 (2006.01) *H03M 13/00* (2017.01)
H03M 13/15 (2015.01) *H03M 7/00* (2006.01)
(52) CPC특허분류
H03M 13/1105 (2013.01)
H03M 13/1515 (2013.01)
(21) 출원번호 10-2021-0104372
(22) 출원일자 2021년08월09일
심사청구일자 2021년08월09일
(65) 공개번호 10-2023-0022510
(43) 공개일자 2023년02월16일
(56) 선행기술조사문헌
KR1020210023674 A
JP2021071966 A
WO2018148260 A1

(73) 특허권자
서울대학교산학협력단
서울특별시 관악구 관악로 1 (신림동)
전남대학교산학협력단
광주광역시 북구 용봉로 77 (용봉동)
(뒷면에 계속)
(72) 발명자
박성준
서울특별시 서초구 잠원동 신반포로 33길 66, 101-606
정재호
경기도 성남시 분당구 양현로 272, 310동 1301호
(뒷면에 계속)
(74) 대리인
특허법인비엘티

전체 청구항 수 : 총 25 항 심사관 : 조춘근

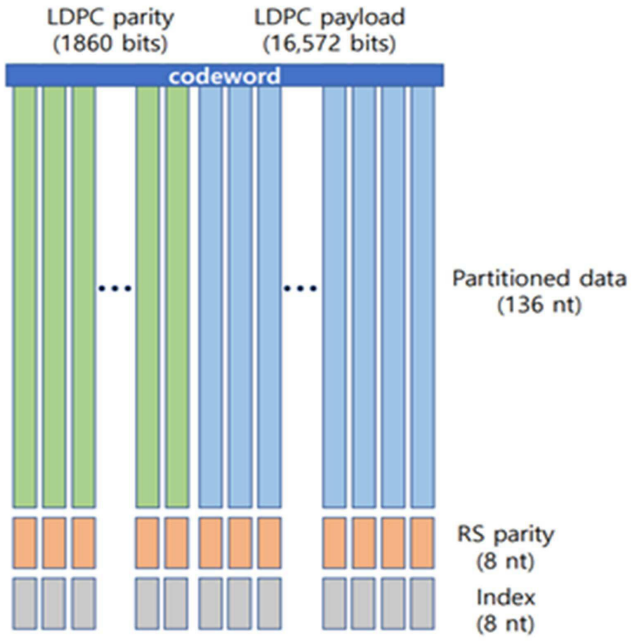
(54) 발명의 명칭 **저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법, 프로그램 및 장치**

(57) 요 약

저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법, 프로그램 및 장치가 제공된다. 상기 방법은, 특정 데이터를 기 설정된 제1 길이의 제1 이진수 벡터로 변환하는 단계, 상기 제1 이진수 벡터를 기 설정된 제2 길이의 제2 이진수 벡터로 나누되, 기 설정된 제1 개수로 나누는 단계, 상기 제1 개수의 제2 이진수 벡터를

(뒷면에 계속)

대 표 도 - 도3



세로 방향으로 정렬하는 단계, 기 설정된 제3 길이의 패리티를 이용하여 가로 방향으로 LDPC(low density parity check) 부호화를 수행하는 단계, 상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 기 설정된 제4 길이의 주소 값을 지정하는 단계, 상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2 비트(bit) 당 하나의 DNA로 치환하는 단계 및 상기 DNA로 치환되어 상기 제5 길이의 절반 길이인 상기 제2 개수의 염기 배열 각각에 대해서 천공(puncturing)을 수행하는 단계를 포함한다.

본 발명은 "삼성전자 미래기술육성센터의 지원을 받아 수행된 연구"이다.

(52) CPC특허분류

H03M 13/6362 (2013.01)

H03M 7/001 (2013.01)

(73) 특허권자

울산대학교 산학협력단

울산광역시 남구 대학로 93(무거동)

홍익대학교 산학협력단

서울특별시 마포구 와우산로 94 (상수동)

(72) 발명자

노종선

서울특별시 강남구 광평로10길 15 상록수아파트 109동 406호

박호성

광주광역시 광산구 수완로33번길 76, 106-101

김성환

서울특별시 광진구 아차산로66길 70, 102-302

노 알버트

서울시 용산구 이촌로 65가길 51, 한가람아파트 216동 1005호

명 세 서

청구범위

청구항 1

장치에 의해 수행되는 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법에 있어서,

특정 데이터를 기 설정된 제1 길이(bit)의 제1 이진수 벡터로 변환하는 단계;

상기 제1 이진수 벡터를 기 설정된 제1 개수의 기 설정된 제2 길이(bit)의 제2 이진수 벡터가 되도록 나누는 단계;

상기 제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬하는 단계;

기 설정된 제3 길이(bit)의 패리티를 이용하여 상기 제2 이진수 벡터를 가로 방향으로 LDPC 부호화를 수행하는 단계;

상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 기 설정된 제4 길이(bit)의 주소 값을 지정하는 단계;

상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이(bit)의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2 비트 당 아데닌(A, Adenine), 구아닌(G,Guanine), 시토신(C, Cytosine) 및 티민(T, Thymine) 중 하나로 치환하는 단계; 및

상기 치환을 통해 상기 제5 길이의 절반 길이(nt)로 형성되는 염기 배열에 대해서 천공(puncturing)을 수행하는 단계;를 포함하는, 방법.

청구항 2

제1 항에 있어서,

상기 제1 이진수 벡터를 상기 제1 개수의 상기 제2 이진수 벡터가 되도록 나누기 전에,

난수를 이용한 XOR 연산을 통해 상기 제1 이진수 벡터를 랜덤화하는 단계;를 더 포함하는, 방법.

청구항 3

제1 항에 있어서,

상기 주소 값 지정 단계는,

부호화를 위해 필요한 주소 값의 개수를 산출하는 단계;

상기 산출된 주소 값 각각에 대해 RS(Reed-Solomon) 부호를 적용하여 부호화하는 단계; 및

상기 부호화된 주소 값 각각을 상기 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 순서대로 지정해 주는 단계;를 포함하는, 방법.

청구항 4

제3 항에 있어서,

상기 주소 값의 개수를 산출하는 단계는,

상기 주소 값의 비트 수에 기초한 전체 경우의 수 중에서 상기 염기 배열 내의 특정한 위치의 연속된 두 개의 염기가 서로 다른 경우의 수에 상응하는 개수를 상기 주소 값의 개수로 산출하는, 방법.

청구항 5

제3 항에 있어서,

상기 주소 값은 짝수 패리티를 포함하는, 방법.

청구항 6

제1 항에 있어서,

상기 치환 단계는,

상기 2 비트에 포함된 0 및 1의 개수에 따라 상기 2 비트를 상기 아데닌(A, Adenine), 상기 구아닌(G, Guanine), 상기 시토신(C, Cytosine) 및 상기 티민(T, Thymine) 중 하나로 치환하는, 방법.

청구항 7

제6 항에 있어서,

상기 천공 수행 단계는,

상기 제2 개수의 염기 배열 중에서, 상기 구아닌(G, Guanine) 및 상기 시토신(C, Cytosine)의 비율이 기준 비율과의 차이가 기 설정된 값보다 큰 염기 배열과 동일한 염기가 기 설정된 횟수 이상 반복되는 염기 배열에 대해서 천공을 수행하는 것인, 방법.

청구항 8

제1 항에 있어서,

상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 단계(여기서 N은 자연수); 및

상기 복호화된 결과에 기초하여, 상기 제1 개수의 제2 이진수 벡터를 상기 지정된 주소 값의 순서에 따라 상기 제1 길이의 제1 이진수 벡터로 통합하는 단계;를 더 포함하고,

상기 N개의 염기 배열은 서로 다르거나 적어도 두개가 동일하고,

상기 N개의 염기 배열이 서로 다른 경우에 상기 N은 상기 제2 개수보다 작거나 동일하고,

상기 N개의 염기 배열이 적어도 두개가 동일한 경우에 상기 N은 상기 제2 개수보다 큰, 방법.

청구항 9

제8 항에 있어서,

상기 제1 LDPC 복호화 및 제2 LDPC 복호화 수행 단계는,

상기 추출된 N개의 염기 배열 각각의 주소 값에 대한 RS 복호화 및 짝수 패리티 복호화를 수행하는 단계;

상기 복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑하는 단계;

상기 그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환하는 단계;

상기 주소 값에 염기 배열의 존재 여부에 따라 상기 치환된 이진수 벡터 각각에 대해 로그 우도 비 (LLR, Log-Likelihood Ratio)를 산출하는 단계;

상기 산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬하는 단계;

상기 세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행하는 단계;

상기 제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행하는 단계; 및

상기 제2 LDPC 복호화된 결과에 기초하여, 최종적으로 복호화에 성공하였는지 여부를 판단하는 단계;를 포함하는, 방법.

청구항 10

제9 항에 있어서,

상기 제1 LDPC 복호화 수행 단계는,

각 주소 값에 대응하는 염기 배열 각각에 대해, 제1 LDPC 복호화 수행 전과 후의 바뀐 비트 수를 계산하는 단계; 및

상기 바뀐 비트 수가 기 설정된 개수 이상인 염기 배열에 대해 상기 로그 우도 비를 0으로 초기화하는 단계;를 포함하고,

상기 제2 LDPC 복호화 수행 단계는,

상기 제1 LDPC 복호화에 실패한 로그 우도 비를 가로방향으로 제2 LDPC 복호화를 수행하고,

상기 최종 복호화의 성공 여부를 판단하는 단계는,

상기 제1 LDPC 복호화된 모든 염기 배열과, 상기 제1 LDPC 복호화 및 상기 제2 LDPC 복호화된 모든 염기 배열에 대한 패리티 검사 행렬을 통해 최종적으로 복호화에 성공하였는지 여부를 판단하는, 방법.

청구항 11

제10 항에 있어서,

상기 최종적으로 복호화에 실패한 것으로 판단된 경우,

상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추가로 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 과정을 반복하는, 방법.

청구항 12

제9 항에 있어서,

상기 로그 우도 비 (LLR, Log-Likelihood Ratio)는 아래 수학식을 이용하여 산출되는, 방법.

$$LLR = \log \frac{p(k_0, k_1 \mid 0)}{p(k_0, k_1 \mid 1)} = (k_0 - k_1) \log \frac{1-\epsilon}{\epsilon}$$

여기서, K_0은 같은 주소 값을 가지는 염기 배열들의 같은 위치에서의 0의 개수, K_1은 같은 주소 값을 가지는 염기 배열들의 같은 위치에서의 1의 개수, ϵ는 기 설정된 파라미터 값

청구항 13

컴퓨터인 하드웨어와 결합되어, 제1 항 내지 제12 항 중 어느 한 항의 방법을 실행하기 위해 컴퓨터 판독 가능한 기록 매체에 저장된 프로그램.

청구항 14

저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 장치에 있어서,

통신부;

상기 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화를 위한 적어도 하나의 프로세스를 저장하고 있는 메모리; 및

상기 프로세스에 따라 동작하는 프로세서;를 포함하고,

상기 프로세서는, 상기 프로세스를 기반으로,

특정 데이터를 기 설정된 제1 길이(bit)의 제1 이진수 벡터로 변환하고,

상기 제1 이진수 벡터를 기 설정된 제1 개수의 기 설정된 제2 길이(bit)의 제2 이진수 벡터가 되도록 나누고,

상기 제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬하고,

기 설정된 제3 길이(bit)의 패리티를 이용하여 상기 제2 이진수 벡터를 가로 방향으로 LDPC 부호화를 수행하고,

상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각

에 기 설정된 제4 길이(bit)의 주소 값을 지정하고,

상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이(bit)의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2 비트 당 아데닌(A, Adenine), 구아닌(G,Guanine), 시토신(C, Cytosine) 및 티민(T, Thymine) 중 하나로 치환하고,

상기 치환을 통해 상기 제5 길이의 절반 길이(nt)로 형성되는 염기 배열에 대해서 천공(puncturing)을 수행하는, 장치.

청구항 15

제14 항에 있어서,

상기 프로세서는,

상기 제1 이진수 벡터를 상기 제1 개수의 상기 제2 이진수 벡터가 되도록 나누기 전에,

난수를 이용한 XOR 연산을 통해 상기 제1 이진수 벡터를 랜덤화하는, 장치.

청구항 16

제14 항에 있어서,

상기 프로세서는,

상기 주소 값 지정 시,

부호화를 위해 필요한 주소 값의 개수를 산출하고,

상기 산출된 주소 값 각각에 대해 RS(Reed-Solomon) 부호를 적용하여 부호화하고,

상기 부호화된 주소 값 각각을 상기 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 순서대로 지정해 주는, 장치.

청구항 17

제16 항에 있어서,

상기 프로세서는,

상기 주소 값의 개수를 산출 시,

상기 주소 값의 비트 수에 기초한 전체 경우의 수 중에서 상기 염기 배열 내의 특정한 위치의 연속된 두 개의 염기가 서로 다른 경우의 수에 상응하는 개수를 상기 주소 값의 개수로 산출하는, 장치.

청구항 18

제16 항에 있어서,

상기 주소 값은 짝수 패리티를 포함하는, 장치.

청구항 19

제14 항에 있어서,

상기 프로세서는,

상기 치환 시,

상기 2 비트에 포함된 0 및 1의 개수에 따라 상기 2 비트를 상기 아데닌(A, Adenine), 상기 구아닌(G, Guanine), 상기 시토신(C, Cytosine) 및 상기 티민(T, Thymine) 중 하나로 치환하는, 장치.

청구항 20

제19 항에 있어서,

상기 프로세서는,

상기 천공 수행 시,

상기 제2 개수의 염기 배열 중에서, 상기 구아닌(G, Guanine) 및 상기 시토신(C, Cytosine)의 비율이 기준 비율과의 차이가 기 설정된 값보다 큰 염기 배열과 동일한 염기가 기 설정된 횟수 이상 반복되는 염기 배열에 대해서 천공을 수행하는 것인, 장치.

청구항 21

제14 항에 있어서,

상기 프로세서는,

상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하고(여기서 N은 자연수),

상기 복호화된 결과에 기초하여, 상기 제1 개수의 제2 이진수 벡터를 상기 지정된 주소 값의 순서에 따라 상기 제1 길이의 제1 이진수 벡터로 통합하고,

상기 N개의 염기 배열은 서로 다르거나 적어도 두개가 동일하고,

상기 N개의 염기 배열이 서로 다른 경우에 상기 N은 상기 제2 개수보다 작거나 동일하고,

상기 N개의 염기 배열이 적어도 두개가 동일한 경우에 상기 N은 상기 제2 개수보다 큰, 장치.

청구항 22

제21 항에 있어서,

상기 프로세서는,

상기 제1 LDPC 복호화 및 제2 LDPC 복호화 수행 시,

상기 추출된 N개의 염기 배열 각각의 주소 값에 대한 RS 복호화 및 짝수 패리티 복호화를 수행하고,

상기 복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑하고,

상기 그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환하고,

상기 주소 값에 염기 배열의 존재 여부에 따라 상기 치환된 이진수 벡터 각각에 대해 로그 우도 비 (LLR, Log-Likelihood Ratio)를 산출하고,

상기 산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬하고,

상기 세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행하고,

상기 제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행하고,

상기 제2 LDPC 복호화된 결과에 기초하여, 최종적으로 복호화에 성공하였는지 여부를 판단하는, 장치.

청구항 23

제22 항에 있어서,

상기 프로세서는,

상기 제1 LDPC 복호화 수행 시,

각 주소 값에 대응하는 염기 배열 각각에 대해, 제1 LDPC 복호화 수행 전과 후의 바뀐 비트 수를 계산하고,

상기 바뀐 비트 수가 기 설정된 개수 이상인 염기 배열에 대해 상기 로그 우도 비를 0으로 초기화하고,

상기 제2 LDPC 복호화 수행 시,

상기 제1 LDPC 복호화에 실패한 로그 우도 비를 가로방향으로 제2 LDPC 복호화를 수행하는 것으로 결정하고,

상기 최종 복호화의 성공 여부를 판단 시,

상기 제1 LDPC 복호화된 모든 염기 배열과, 상기 제1 LDPC 복호화 및 상기 제2 LDPC 복호화된 모든 염기 배열에 대한 패리티 검사 행렬을 통해 최종적으로 복호화에 성공하였는지 여부를 판단하는, 장치.

청구항 24

제23 항에 있어서,

상기 프로세서는,

상기 최종적으로 복호화에 실패한 것으로 판단된 경우,

상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추가로 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 과정을 반복하는, 장치.

청구항 25

제22 항에 있어서,

상기 로그 우도 비 (LLR, Log-Likelihood Ratio)는 아래 수학식을 이용하여 산출되는, 장치.

$$LLR = \log \frac{p(k_0, k_1 \mid 0)}{p(k_0, k_1 \mid 1)} = (k_0 - k_1) \log \frac{1-\epsilon}{\epsilon}$$

여기서, K_0은 같은 주소 값을 가지는 염기 배열들의 같은 위치에서의 0의 개수, K_1은 같은 주소 값을 가지는 염기 배열들의 같은 위치에서의 1의 개수, ϵ는 기 설정된 파라미터 값

발명의 설명

기 술 분 야

[0001] 본 발명은 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법, 프로그램 및 장치에 관한 것이다.

배 경 기 술

[0002] 디지털 데이터는 실시간으로 기하급수적으로 증가하고 있다. 2025년에는 175 제타바이트의 데이터가 생성된다고 추측되고 있다. 이렇게 증가하는 데이터를 처리하고 저장하기 위해서 새로운 저장 장치의 필요성이 야기되고 있다.

[0003] 새로운 저장 장치는 높은 저장 밀도, 긴 저장 시간, 낮은 에너지 비용을 필요로 한다. 하지만 기존의 저장 장치인 자기 테이프, 하드 디스크 드라이브, SDD는 새로운 저장 장치에 적합하지 않다. 이에 따라 DNA가 새로운 저장 장치의 매개체로 떠오르고 있다. 우리는 이것을 DNA 저장 장치라고 부른다. 이는 데이터를 A, C, G, T 네 가지 염기로 치환하여 저장하는 장치이다.

[0004] DNA 저장 장치는 현재 매우 활발히 연구가 이루어지고 있는 분야 중 하나이다. 이러한 DNA 저장 장치에는 3 가지 서로 다른 오류가 존재한다: 치환 오류 (substitution error), 삽입 오류 (insertion error), 삭제 오류 (deletion error).

[0005] 치환 오류는 말 그대로 본래 전송되어야 하는 염기 대신 나머지 3 가지 다른 염기 중 하나로 바뀌어서 전송되는 오류이다. 삽입 오류는 본래 전송되어야 하는 염기들 사이에 임의의 염기가 추가로 발생하여 전송되는 오류이고, 삭제 오류는 본래 전송되어야 하는 염기가 사라져서 전송되는 오류이다.

[0006] 이 때, 치환 오류는 길이가 그대로이고, 하나의 염기에서만 오류가 생기지만, 삽입/삭제 오류가 생길 경우, DNA의 길이가 달라지고, 전송되는 염기의 위치가 하나씩 밀리는 경우가 발생한다. 따라서 치환 오류에 비해 삽입/삭제 오류가 DNA 저장 장치에서 데이터를 저장하고 읽는데에 매우 치명적인 오류로 작용한다. 이러한 오류들로부터 DNA 저장 장치를 보호하고, 완벽하게 데이터를 저장하고 읽기 위해서는 DNA 저장 장치에 오류 정정 부호를

필수적으로 적용해야 한다.

[0007] 또한 DNA 저장 장치의 오류율은 DNA의 생화학적 구조에 영향을 많이 받는다. DNA를 합성하고, 이를 저장하고 읽는 과정에서, 가장 큰 두 가지 제한이 있다. DNA 내에서 G, C의 비율이 50%에서 멀 때 오류가 많이 생기고, 같은 염기서열이 연속적으로 여러 개 반복되면 오류가 많이 생긴다고 알려져 있다. 따라서 이러한 두 가지의 생화학적 특성을 만족시키는 것이 DNA 저장 장치에서는 오류를 줄이는데 도움을 줄 수 있다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 등록특허공보 제10-1529360호, 2015.06.10.

발명의 내용

해결하려는 과제

[0009] 본 발명이 해결하고자 하는 과제는 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법, 프로그램 및 장치를 제공하는 것이다.

[0010] 본 발명이 해결하고자 하는 과제들은 이상에서 언급된 과제로 제한되지 않으며, 언급되지 않은 또 다른 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0011] 상술한 과제를 해결하기 위한 본 발명의 일 면에 따른 장치에 의해 수행되는 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 방법은, 특정 데이터를 기 설정된 제1 길이의 제1 이진수 벡터로 변환하는 단계, 상기 제1 이진수 벡터를 기 설정된 제2 길이의 제2 이진수 벡터로 나누되, 기 설정된 제1 개수로 나누는 단계, 상기 제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬하는 단계, 기 설정된 제3 길이의 패리티를 이용하여 가로 방향으로 LDPC(low density parity check) 부호화를 수행하는 단계, 상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 기 설정된 제4 길이의 주소 값을 지정하는 단계, 상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2 비트(bit) 당 하나의 DNA로 치환하는 단계 및 상기 DNA로 치환되어 상기 제5 길이의 절반 길이인 상기 제2 개수의 염기 배열 각각에 대해서 천공(puncturing)을 수행하는 단계를 포함한다.

[0012] 본 발명에서, 상기 방법은, 상기 제1 이진수 벡터를 상기 제1 개수의 상기 제2 이진수 벡터로 나누기 전에, 난수를 이용한 XOR 연산을 통해 상기 긴 이진수 벡터를 랜덤화하는 단계;를 더 포함할 수 있다.

[0013] 본 발명에서, 상기 주소 값 지정 단계는, 부호화를 위해 필요한 주소 값의 개수를 산출하는 단계, 상기 산출된 주소 값 각각에 대해 RS(Reed-Solomon) 부호를 적용하여 부호화하는 단계 및 상기 부호화된 주소 값 각각을 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각에 순서대로 지정해주는 단계를 포함할 수 있다.

[0014] 본 발명에서, 상기 주소 값의 개수를 산출하는 단계는, 상기 주소 값의 비트 수에 기초한 전체 경우의 수 중에서 특정 위치의 복수의 비트가 서로 다른 염기인 경우의 수에 상응하는 개수를 상기 주소 값의 개수로 산출할 수 있다.

[0015] 본 발명에서, 상기 주소 값은 짝수 패리티를 포함할 수 있다.

[0016] 본 발명에서, 상기 치환 단계는, 상기 2 비트는 0 및 1의 개수에 따라 아데닌(A, Adenine), 구아닌(G, Guanine), 시토신(C, Cytosine) 및 티민(T, Thymine) 중 하나로 치환될 수 있다.

[0017] 본 발명에서, 상기 천공 수행 단계는, 상기 제2 개수의 염기 배열 중에서, 구아닌(G, Guanine) 및 시토신(C, Cytosine)의 비율이 기준 비율과의 차이가 기 설정된 값보다 큰 염기 배열과 동일한 염기가 기 설정된 횟수 이상 반복되는 염기 배열에 대해서 천공을 수행하는 것일 수 있다.

[0018] 본 발명에서, 상기 방법은, 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 단계(여기서 N은 자연수) 및 상기 복호화된 결과에 기초하여, 상기 제1 개수

의 제2 이진수 벡터를 상기 지정된 주소 값의 순서에 따라 상기 제1 길이의 제1 이진수 벡터로 통합하는 단계를 더 포함하고, 상기 N개의 염기 배열은 서로 다르거나 적어도 두개가 동일하고, 상기 N개의 염기 배열이 서로 다른 경우에 상기 N은 상기 제2 개수보다 작거나 동일하고, 상기 N개의 염기 배열이 적어도 두개가 동일한 경우에 상기 N은 상기 제2 개수보다 클 수 있다.

[0019] 본 발명에서, 상기 제1 LDPC 복호화 및 제2 LDPC 복호화 수행 단계는, 상기 추출된 N개의 염기 배열 각각의 주소 값에 대한 RS 복호화 및 짝수 패리티 복호화를 수행하는 단계, 상기 복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑하는 단계, 상기 그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환하는 단계, 상기 주소 값에 염기 배열의 존재 여부에 따라 상기 치환된 이진수 벡터 각각에 대해 로그 우도 비(LLR, Log-Likelihood Ratio)를 산출하는 단계, 상기 산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬하는 단계, 상기 세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행하는 단계, 상기 제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행하는 단계 및 상기 제2 LDPC 복호화된 결과에 기초하여, 상기 최종 복호화의 성공 여부를 판단하는 단계를 포함할 수 있다.

[0020] 본 발명에서, 상기 제1 LDPC 복호화 수행 단계는, 각 주소 값에 대응하는 염기 배열 각각에 대해, 제1 LDPC 복호화 수행 전과 후의 바뀐 비트 수를 계산하는 단계 및 상기 바뀐 비트 수가 기 설정된 개수 이상인 염기 배열에 대해 상기 로그 우도 비를 0으로 초기화하는 단계를 포함하고, 상기 제2 LDPC 복호화 수행 단계는, 상기 제1 LDPC 복호화에 실패한 가로방향 LPDC 부호어에 대해서만 제2 LDPC 복호화를 수행하는 것으로 결정하고, 상기 최종 복호화의 성공 여부를 판단하는 단계는, 상기 제1 LDPC 복호화된 모든 염기 배열과, 상기 제1 LDPC 복호화 및 상기 제2 LDPC 복호화된 모든 염기 배열에 대한 패리티 검사 행렬을 통해 최종 복호화의 성공 여부를 판단할 수 있다.

[0021] 본 발명에서, 상기 최종 복호화가 실패한 것으로 판단된 경우, 상기 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추가로 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 과정을 반복할 수 있다.

[0022] 본 발명에서, 상기 로그 우도 비 (LLR, Log-Likelihood Ratio)는 아래 수학식을 이용하여 산출될 수 있다.

[0023]
$$LLR = \log \frac{p(k_0, k_1 \mid 0)}{p(k_0, k_1 \mid 1)} = (k_0 - k_1) \log \frac{1-\epsilon}{\epsilon}$$

[0024] 여기서, K_0은 같은 클러스터 내에서 같은 위치에서의 0의 개수, K_1은 같은 클러스터 내에서 같은 위치에서의 1의 개수, ϵ는 기 설정된 파라미터 값

[0025] 상술한 과제를 해결하기 위한 본 발명의 다른 면에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 장치는, 통신부, 상기 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화를 위한 적어도 하나의 프로세스를 저장하고 있는 메모리 및 상기 프로세스에 따라 동작하는 프로세서;를 포함하고, 상기 프로세서는, 상기 프로세스를 기반으로, 특정 데이터를 기 설정된 제1 길이의 제1 이진수 벡터로 변환하고, 상기 제1 이진수 벡터를 기 설정된 제2 길이의 제2 이진수 벡터로 나누되, 기 설정된 제1 개수로 나누고, 상기 제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬하고, 기 설정된 제3 길이의 패리티를 이용하여 가로 방향으로 LDPC(low density parity check) 부호화를 수행하고, 상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 기 설정된 제4 길이의 주소 값을 지정하고, 상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2비트(bit) 당 하나의 DNA로 치환하고, 상기 DNA로 치환되어 상기 제5 길이의 절반 길이인 상기 제2 개수의 염기 배열 각각에 대해서 천공(puncturing)을 수행한다.

[0026] 본 발명에서, 상기 프로세서는, 상기 제1 이진수 벡터를 상기 제1 개수의 상기 제2 이진수 벡터로 나누기 전에, 난수를 이용한 XOR 연산을 통해 상기 긴 이진수 벡터를 랜덤화할 수 있다.

[0027] 본 발명에서, 상기 프로세서는, 상기 주소 값 지정 시, 부호화를 위해 필요한 주소 값의 개수를 산출하고, 상기 산출된 주소 값 각각에 대해 RS(Reed-Solomon) 부호를 적용하여 부호화하고, 상기 부호화된 주소 값 각각을 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각에 순서대로 지정해줄 수 있다.

[0028] 본 발명에서, 상기 프로세서는, 상기 주소 값의 개수를 산출 시, 상기 주소 값의 비트 수에 기초한 전체 경우의 수 중에서 특정 위치의 복수의 비트가 서로 다른 염기인 경우의 수에 상응하는 개수를 상기 주소 값의 개수로

산출할 수 있다.

[0029] 본 발명에서, 상기 주소 값은 짝수 패리티를 포함할 수 있다.

[0030] 본 발명에서, 상기 프로세서는, 상기 치환 시, 상기 2 비트에 포함된 0 및 1의 개수에 따라 아데닌(A, Adenine), 구아닌(G, Guanine), 시토신(C, Cytosine) 및 티민(T, Thymine) 중 하나로 치환할 수 있다.

[0031] 본 발명에서, 상기 프로세서는, 상기 천공 수행 시, 상기 제2 개수의 염기 배열 중에서, 구아닌(G, Guanine) 및 시토신(C, Cytosine)의 비율이 기준 비율과의 차이가 기 설정된 값보다 큰 염기 배열과 동일한 염기가 기 설정된 횟수 이상 반복되는 염기 배열에 대해서 천공을 수행할 수 있다.

[0032] 본 발명에서, 상기 프로세서는, 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하고(여기서 N은 자연수), 상기 복호화된 결과에 기초하여, 상기 제1 개수의 제2 이진수 벡터를 상기 지정된 주소 값의 순서에 따라 상기 제1 길이의 제1 이진수 벡터로 통합하고, 상기 N개의 염기 배열은 서로 다르거나 적어도 두개가 동일하고, 상기 N개의 염기 배열이 서로 다른 경우에 상기 N은 상기 제2 개수보다 작거나 동일하고, 상기 N개의 염기 배열이 적어도 두개가 동일한 경우에 상기 N은 상기 제2 개수보다 클 수 있다.

[0033] 본 발명에서, 상기 프로세서는, 상기 제1 LDPC 복호화 및 제2 LDPC 복호화 수행 시, 상기 추출된 N개의 염기 배열 각각의 주소 값에 대한 RS 복호화 및 짝수 패리티 복호화를 수행하고, 상기 복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑하고, 상기 그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환하고, 상기 주소 값에 염기 배열의 존재 여부에 따라 상기 치환된 이진수 벡터 각각에 대해 로그 우도 비(LLR, Log-Likelihood Ratio)를 산출하고, 상기 산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬하고, 상기 세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행하고, 상기 제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행하고, 상기 제2 LDPC 복호화된 결과에 기초하여, 상기 최종 복호화의 성공 여부를 판단할 수 있다.

[0034] 본 발명에서, 상기 프로세서는, 상기 제1 LDPC 복호화 수행 시, 각 주소 값에 대응하는 염기 배열 각각에 대해, 제1 LDPC 복호화 수행 전과 후의 바뀐 비트 수를 계산하고, 상기 바뀐 비트 수가 기 설정된 개수 이상인 염기 배열에 대해 상기 로그 우도 비를 0으로 초기화하고, 상기 제2 LDPC 복호화 수행 시, 상기 제1 LDPC 복호화에 실패한 가로방향 LPDC 부호어에 대해서만 제2 LDPC 복호화를 수행하는 것으로 결정하고, 상기 최종 복호화의 성공 여부를 판단 시, 상기 제1 LDPC 복호화된 모든 염기 배열과, 상기 제1 LDPC 복호화 및 상기 제2 LDPC 복호화된 모든 염기 배열에 대한 패리티 검사 행렬을 통해 최종 복호화의 성공 여부를 판단할 수 있다.

[0035] 본 발명에서, 상기 프로세서는, 상기 최종 복호화가 실패한 것으로 판단된 경우, 상기 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추가로 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 과정을 반복할 수 있다.

[0036] 본 발명에서, 상기 로그 우도 비 (LLR, Log-Likelihood Ratio)는 아래 수학식을 이용하여 산출될 수 있다.

[0037]
$$LLR = \log \frac{p(k_0, k_1 \mid 0)}{p(k_0, k_1 \mid 1)} = (k_0 - k_1) \log \frac{1-\epsilon}{\epsilon}$$

[0038] 여기서, K_0은 같은 클러스터 내에서 같은 위치에서의 0의 개수, K_1은 같은 클러스터 내에서 같은 위치에서의 1의 개수, ϵ는 기 설정된 파라미터 값

[0039] 이 외에도, 본 발명을 구현하기 위한 다른 방법, 다른 시스템 및 상기 방법을 실행하기 위한 컴퓨터 프로그램을 기록하는 컴퓨터 판독 가능한 기록 매체가 더 제공될 수 있다.

발명의 효과

[0040] 본 발명에 따르면, DNA 저장 장치에 존재하는 3 가지의 오류를 유발하는 생화학적 특정들을 가지고 있는 염기 배열들을 제외하고, 최종 생성된 염기 배열들의 G와 C의 비율과, 연속된 염기의 개수를 확인하여, 기준 조건을 충족하지 않는 염기 배열을들 제외할 수 있다. 이에 따라, 데이터 복호화 시 오류 발생 가능성을 낮추고 보다 정확도 높은 데이터 복원이 가능하다.

[0041] 본 발명의 효과들은 이상에서 언급된 효과로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로

부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0042] 도 1은 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 장치의 개략적인 블록도이다.

도 2는 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 방법의 순서도이다.

도 3은 본 발명에 따른 DNA 데이터 부호화를 설명하기 위한 도면이다.

도 4는 본 발명에 따른 주소 값을 설명하기 위한 도면이다.

도 5는 본 발명에 따른 2 비트를 염기로 치환하는 것을 설명하기 위한 도면이다.

도 6은 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 복호화 방법의 순서도이다.

도 7은 도 6의 단계 S210의 구체적인 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0043] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 제한되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야의 통상의 기술자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0044] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다. 명세서 전체에 걸쳐 동일한 도면 부호는 동일한 구성 요소를 지칭하며, "및/또는"은 언급된 구성요소들의 각각 및 하나 이상의 모든 조합을 포함한다. 비록 "제1", "제2" 등이 다양한 구성요소들을 서술하기 위해서 사용되나, 이들 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 구성요소를 다른 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 구성요소는 본 발명의 기술적 사상 내에서 제2 구성요소일 수도 있음은 물론이다.

[0045] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술분야의 통상의 기술자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.

[0046] 공간적으로 상대적인 용어인 "아래(below)", "아래(beneath)", "하부(lower)", "위(above)", "상부(upper)" 등은 도면에 도시되어 있는 바와 같이 하나의 구성요소와 다른 구성요소들과의 상관관계를 용이하게 기술하기 위해 사용될 수 있다. 공간적으로 상대적인 용어는 도면에 도시되어 있는 방향에 더하여 사용시 또는 동작시 구성요소들의 서로 다른 방향을 포함하는 용어로 이해되어야 한다. 예를 들어, 도면에 도시되어 있는 구성요소를 뒤집을 경우, 다른 구성요소의 "아래(below)"또는 "아래(beneath)"로 기술된 구성요소는 다른 구성요소의 "위(above)"에 놓여질 수 있다. 따라서, 예시적인 용어인 "아래"는 아래와 위의 방향을 모두 포함할 수 있다. 구성요소는 다른 방향으로도 배향될 수 있으며, 이에 따라 공간적으로 상대적인 용어들은 배향에 따라 해석될 수 있다.

[0047] 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다.

[0048] 설명에 앞서 본 명세서에서 사용하는 용어의 의미를 간략히 설명한다. 그렇지만 용어의 설명은 본 명세서의 이해를 돕기 위한 것이므로, 명시적으로 본 발명을 한정하는 사항으로 기재하지 않은 경우에 본 발명의 기술적 사상을 한정하는 의미로 사용하는 것이 아님을 주의해야 한다.

[0049] 본 명세서에서 '장치'는 연산처리를 수행하여 사용자에게 결과를 제공할 수 있는 다양한 장치들이 모두 포함된다. 예를 들어, 장치는 컴퓨터 및 이동 단말기 형태가 될 수 있다. 상기 컴퓨터는 클라이언트로부터 요청을 수신하여 정보처리를 수행하는 서버 형태가 될 수 있다. 또한, 컴퓨터에는 시퀀싱을 수행하는 시퀀싱 장치가 해당

될 수 있다. 상기 이동 단말기는 휴대폰, 스마트 폰(smart phone), PDA(personal digital assistants), PMP(portable multimedia player), 네비게이션, 노트북 PC, 슬레이트 PC(slate PC), 태블릿 PC(tablet PC), 울트라북(ultrabook), 웨어러블 디바이스(wearable device, 예를 들어, 워치형 단말기 (smartwatch), 글래스형 단말기 (smart glass), HMD(head mounted display)) 등이 포함될 수 있다.

[0050] 본 명세서에서 '올리고(oligo)' 또는 '올리고 리드(oligo read)'는 특정한 염기(아데닌, 구아닌, 시토신 또는 티민)를 포함하는 복수개의 뉴클레오타이드(nucleotide) 단위체가 합성된 중합체를 의미한다.

[0051] 본 명세서에서 '시퀀스(sequence)'는 특정한 올리고 리드를 순서대로 판독(시퀀싱)하여 읽어낸 염기 서열을 의미한다. 본 명세서에서 '시퀀스'와 '올리고 리드'는 혼용되어 사용될 수 있다. 예를 들어, '스티치된 시퀀스'는 구체적으로 '스티치된 올리고 리드의 시퀀스'를 의미할 수 있다.

[0052] 도 1은 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 장치의 개략적인 블록도이다.

[0053] 도 2는 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 방법의 순서도이다.

[0054] 도 3은 본 발명에 따른 DNA 데이터 부호화를 설명하기 위한 도면이다.

[0055] 도 4는 본 발명에 따른 주소 값을 설명하기 위한 도면이다.

[0056] 도 5는 본 발명에 따른 2 비트를 염기로 치환하는 것을 설명하기 위한 도면이다.

[0057] 도 6은 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 복호화 방법의 순서도이다.

[0058] 도 7은 도 6의 단계 S210의 구체적인 방법의 순서도이다.

[0059] 이하에서 도 1을 참조하여, 본 발명에 따른 저밀도 패리티 체크 부호를 이용한 DNA 데이터 부호화 및 복호화 장치(10)에 대해서 설명하도록 한다.

[0060] 본 발명에 따른 장치(10)는 외부 장치(미도시)로부터 특정 데이터 파일이 입력되면 하나의 긴 이진수 벡터로 표현한 후 네 가지의 염기로 치환하여 부호화하고 이를 저장할 수 있다.

[0061] 본 발명에 따른 장치(10)는 외부 장치(미도시)로부터 부호화된 특정 데이터에 대한 복원이 요청되면 염기 배열로 저장된 데이터를 다시 이진수 벡터로 복호화하여 입력된 데이터의 형태로 다시 복호화하여 제공할 수 있다.

[0062] 이때, 장치(10)는 데이터를 부호화하고 복호화함에 있어서, 저밀도 패리티 체크 부호(LDPC, low density parity check)를 이용하여 복원된 데이터의 정확도를 높일 수 있다.

[0063] 본 발명에 따른 장치(10)는 데이터를 DNA로 치환하여 저장하는 DNA 저장 장치일 수 있다.

[0064] 이러한 장치(10)는 연산처리를 수행하여 사용자에게 결과를 제공할 수 있는 다양한 장치들이 모두 포함될 수 있다.

[0065] 여기서, 장치(10)는 컴퓨터의 형태가 될 수 있다. 보다 상세하게는, 상기 컴퓨터는 연산처리를 수행하여 사용자에게 결과를 제공할 수 있는 다양한 장치들이 모두 포함될 수 있다.

[0066] 예를 들어, 컴퓨터는 데스크 탑 PC, 노트북(Note Book) 뿐만 아니라 스마트폰(Smart phone), 태블릿 PC, 셀룰러 폰(Cellular phone), 피씨에스폰(PCS phone; Personal Communication Service phone), 동기식/비동기식 IMT-2000(International Mobile Telecommunication-2000)의 이동 단말기, 팜 PC(Palm Personal Computer), 개인용 디지털 보조기(PDA; Personal Digital Assistant) 등도 해당될 수 있다. 또한, 헤드마운트 디스플레이(Head Mounted Display; HMD) 장치가 컴퓨팅 기능을 포함하는 경우, HMD장치가 컴퓨터가 될 수 있다.

[0067] 또한, 컴퓨터는 클라이언트로부터 요청을 수신하여 정보처리를 수행하는 서버가 해당될 수 있다.

[0068] 그리고, 장치(10)는 통신부(12), 메모리(14) 및 프로세서(16)를 포함할 수 있다. 여기서, 장치(10)는 도 1에 도시된 구성요소보다 더 적은 수의 구성요소나 더 많은 구성요소를 포함할 수 있다.

[0069] 통신부(12)는 장치(10)와 외부 장치(미도시) 사이, 장치(10)와 외부 서버(미도시) 사이 또는 장치(10)와 통신망(미도시) 사이의 무선 통신을 가능하게 하는 하나 이상의 모듈을 포함할 수 있다.

[0070] 여기서, 통신망(미도시)은 장치(10), 외부 장치(미도시) 및 외부 서버(미도시) 간의 다양한 정보를 송수신할 수 있다. 통신망은 다양한 형태의 통신망이 이용될 수 있으며, 예컨대, WLAN(Wireless LAN), 와이파이(Wi-Fi), 와

이브로(Wibro), 와이맥스(Wimax), HSDPA(High Speed Downlink Packet Access) 등의 무선 통신방식 또는 이더넷(Ethernet), xDSL(ADSL, VDSL), HFC(Hybrid Fiber Coax), FTTC(Fiber to The Curb), FTTH(Fiber To The Home) 등의 유선 통신방식이 이용될 수 있다.

[0071] 한편, 통신망(미도시)은 상기에 제시된 통신방식에 한정되는 것은 아니며, 상술한 통신방식 이외에도 기타 널리 공지되었거나 향후 개발될 모든 형태의 통신 방식을 포함할 수 있다.

[0072] 통신부(12)는 장치(10)를 하나 이상의 네트워크에 연결하는 하나 이상의 모듈을 포함할 수 있다.

[0073] 메모리(14)는 장치(10)의 다양한 기능을 지원하는 데이터를 저장할 수 있다. 메모리(14)는 장치(10)에서 구동되는 다수의 응용 프로그램(application program 또는 애플리케이션(application)), 장치(10)의 동작을 위한 데이터들, 명령어들을 저장할 수 있다. 이러한 응용 프로그램 중 적어도 일부는, 장치(10)의 기본적인 기능을 위하여 존재할 수 있다. 한편, 응용 프로그램은, 메모리(14)에 저장되고, 장치(10) 상에 설치되어, 프로세서(16)에 의하여 상기 장치(10)의 동작(또는 기능)을 수행하도록 구동될 수 있다.

[0074] 프로세서(16)는 상기 응용 프로그램과 관련된 동작 외에도, 통상적으로 장치(10)의 전반적인 동작을 제어할 수 있다. 프로세서(16)는 위에서 살펴본 구성요소들을 통해 입력 또는 출력되는 신호, 데이터, 정보 등을 처리하거나 메모리(14)에 저장된 응용 프로그램을 구동함으로써, 사용자에게 적절한 정보 또는 기능을 제공 또는 처리할 수 있다.

[0075] 또한, 프로세서(16)는 메모리(14)에 저장된 응용 프로그램을 구동하기 위하여, 도 1과 함께 살펴본 구성요소들 중 적어도 일부를 제어할 수 있다. 나아가, 프로세서(16)는 상기 응용 프로그램의 구동을 위하여, 장치(10)에 포함된 구성요소들 중 적어도 둘 이상을 서로 조합하여 동작 시킬 수 있다.

[0076] 이하에서는 도 2 내지 도 5를 참조하여, 프로세서(16)가 저밀도 패리티 체크 부호를 이용한 DNA 데이터를 부호화하는 방법을 설명하도록 한다. 여기서, 프로세서(16)의 동작은 장치(10)에서 수행 가능할 수 있다.

[0077] 도 2를 참조하면, 프로세서(16)는 특정 데이터를 기 설정된 제1 길이의 제1 이진수 벡터로 변환할 수 있다(S110).

[0078] 다음으로, 프로세서(16)는 상기 제1 이진수 벡터를 기 설정된 제2 길이의 제2 이진수 벡터로 나누되, 기 설정된 제1 개수로 나눌 수 있다(S120).

[0079] 다음으로, 프로세서(16)는 상기 제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬할 수 있다(S130).

[0080] 다음으로, 프로세서(16)는 기 설정된 제3 길이의 패리티를 이용하여 가로 방향으로 LDPC(low density parity check) 부호화를 수행할 수 있다(S140).

[0081] 다음으로, 프로세서(16)는 상기 제1 개수 및 상기 제3 길이를 합한 값에 상응하는 제2 개수의 세로 방향으로 정렬된 제2 이진수 벡터 각각에 기 설정된 제4 길이의 주소 값을 지정할 수 있다(S150).

[0082] 다음으로, 프로세서(16)는 상기 제2 길이 및 상기 제4 길이를 합한 값에 상응하는 제5 길이의 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각을 2 비트(bit) 당 하나의 DNA로 치환할 수 있다(S160).

[0083] 다음으로, 프로세서(16)는 상기 DNA로 치환되어 상기 제5 길이의 절반 길이인 상기 제2 개수의 염기 배열 각각에 대해서 천공(puncturing)을 수행할 수 있다(S170).

[0084] 단계 S110에서, 특정 데이터는 부호화가 요청된 데이터 파일일 수 있다.

[0085] 상술한 바와 같이, 프로세서(16)는 특정 데이터 파일을 제1 길이의 하나의 긴 이진수 벡터로 표현한 후, 단계 S120에서 이를 다시 짧은 길이의 이진수 벡터 여러 개로 나누게 되다.

[0086] 그런데 이때, 부호화 마지막 단계에서 2 비트(bit)의 데이터를 하나의 염기로 치환하는데, 데이터는 위에서 언급한 두 가지 생화학적 특성(DNA 내에서 G, C의 비율이 50%에서 멀지 않고, 같은 염기서열이 연속적으로 여러 개 반복되지 않는 것)을 만족하지 못할 가능성이 크다. 따라서, 본 발명은 단계 S120 전에, 난수를 발생시켜 XOR 연산을 통해 데이터를 랜덤화시킬 수 있다. 보다 구체적으로, 본 발명은 상기 제1 이진수 벡터를 상기 제1 개수의 상기 제2 이진수 벡터로 나누기 전에, 난수를 이용한 XOR 연산을 통해 상기 긴 이진수 벡터를 랜덤화하는 단계를 더 포함할 수 있다.

[0087] 이렇게 랜덤화시킨 후, 프로세서(16)는 단계 S120에서, 제1 길이의 긴 이진수 벡터를 제2 길이($n_{payload}$)의 짧은

이진수 벡터 제1 개수(K_{LDPC} 개)로 나누어 줄 수 있다. 즉, 상기 제1 길이는 제2 길이($n_{payload}$)와 제1 개수(K_{LDPC} 개)를 곱한 값에 상응할 수 있다. 예를 들어, 제2 길이($n_{payload}$)는 272 bits로 설정될 수 있고, 제1 개수(K_{LDPC} 개)는 16,572 개로 설정될 수 있다.

[0088] 그리고 단계 S130에서, 도 3에 도시된 파란색 블록과 같이 각각 길이가 272 bits인 16,572 개의 이진수 벡터를 세로 방향으로 정렬할 수 있다. 이렇게 동일한 길이를 갖는 여러 개의 이진수 벡터를 세로 방향으로 정렬하는 이유는 길이가 너무 길면 DNA를 합성할 때의 비용이 매우 비싸기 때문이다.

[0089] 이후, 프로세서(16)는 세로 방향으로 정렬해 준 데이터를 오류 정정 부호 중 현재 가장 널리 쓰이고, 성능이 좋은 저밀도 패리티 체크 부호(LDPC, low density parity check)를 이용하여 부호화 및 복호화 시의 오류를 보호할 수 있다.

[0090] 이때, LDPC 부호의 부호화는 가로 방향으로 수행할 수 있다. 도 3에 도시된 초록색 블록과 같이, 길이가 제3 길이(M_{LDPC}), 예를 들어, 길이가 1,860 bits인 패리티가 형성되고, 각각의 부호는 가로 방향으로 표현될 수 있다.

[0091] 이에 따라, 도 3에 도시된 바와 같이, 제1 개수에 상응하는 파란색 블록의 개수와 제3 길이에 상응하는 초록색 블록의 개수가 더해진 값에 상응하는 제2 개수(N_{LDPC}= K_{LDPC} +M_{LDPC})의 이진수 벡터가 생성되게 된다. 이때, 제2 개수(N_{LDPC}= K_{LDPC} +M_{LDPC}) 이진수 벡터의 길이는 상기 제2 길이($n_{payload}$), 예를 들어, 272 bits일 수 있다.

[0092] 이때, 상술한 바와 같이 DNA 저장 장치에서는 삽입/삭제 오류가 치환 오류에 비해 치명적이다. 하나의 삽입/삭제 오류가 생겨도 데이터의 위치가 하나씩 밀리기 때문에 전체적으로 복원하고자 하는 데이터가 망가지는 경우가 생길 수 있다. 따라서 본 발명과 같이 데이터의 정렬 방향과 부호화의 방향을 수직으로 함으로써 세로 방향으로 정렬된 데이터에 삽입/삭제 오류가 생겨도, 가로 방향의 부호 입장에서는 하나의 비트 오류로 간주하기 하기 때문에 삽입/삭제 오류에 매우 강력한 방법으로 부호화를 수행할 수 있다.

[0093] 이렇게 LDPC 부호의 부호화를 통해 오류 정정 부호를 데이터에 적용하고 나면, 각 세로 방향의 데이터와 패리티에 대해 순서를 알아야 이후에 복호화를 진행할 수 있기 때문에, 각각의 데이터의 순서대로 주소를 지정해준다. 총 N_{LDPC} 개의 세로 방향 데이터의 주소를 각각 지정해 주기 위해서는, 최소 $\lceil \log_2(N_{LDPC}) \rceil = k_{index} bit$ 의 주소값이 필요하다. 예를 들어, $\lceil \log_2(18.432) \rceil = 15bit$ 의 주소값이 필요하다. 0부터 18,432까지의 수를 이진수로 고치면 도 4와 같이 표현될 수 있다.

[0094] 이때, 최종적으로 2bit를 각각 A, C, G, T의 염기로 치환을 하게하게 되는데, 이러한 경우 연속된 염기의 길이가 길게 나타날 수 있다. 그런데 상술한 바와 같이 DNA를 합성하고, 이를 저장하고 읽는 과정에서, 염기 배열이 연속적으로 여러 개 반복되면 오류가 많이 생길 수 있다. 따라서 상기와 같이 주소 값을 지정하여 이진수를 염기로 치환한다면, 같은 염기가 여러 개 반복될 수 있으므로, 0부터 N_{LDPC}까지의 수를 이진수로 치환하여 주소 값으로 쓰는 것은 오류를 많이 발생시킬 수도 있다.

[0095] 따라서, 본 발명은 단계 S150에서, 프로세서(16)는 부호화를 위해 필요한 주소 값의 개수를 산출하고, 상기 산출된 주소 값 각각에 대해 RS(Reed-Solomon) 부호를 적용하여 부호화하고, 상기 부호화된 주소 값 각각을 상기 세로 방향으로 정렬된 제2 이진수 벡터 각각에 순서대로 지정해줄 수 있다.

[0096] 이때, 상기 주소 값의 개수를 산출함에 있어서, 프로세서(16)는 상기 주소 값의 비트 수에 기초한 전체 경우의 수 중에서 특정 위치의 복수의 비트가 서로 다른 염기인 경우의 수에 상응하는 개수를 상기 주소 값의 개수로 산출할 수 있다.

[0097] N_{LDPC}가 18,432이고, k_{index}가 15인 경우를 예로 들어 설명하도록 한다. 편의상 15 bits를 8 nt(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)이라 했을 때, 연속된 염기가 3개를 넘지 않기 위해서는 x_3, x_4, 그리고 x_6, x_7이 다른 염기여야 한다. 이러한 경우의 수는 총 9/16만큼 존재한다. 이를 이진수로 바꿔서 생각하면, 가능한 총 2^{15}의 주소 값 후보군들 중 2^{15}x9/16=18,432개의 주소 값이 산출될 수 있다. 이렇게 산출된 18,432 개의 주소 값을 사용하면 연속된 염기의 수가 최대 3개까지만 존재하는 주소 값들을 구할 수 있다. 이를 통해 연속된 염기의 수를 제한하여 오류가 생길 가능성을 낮출 수 있다.

[0098] 또한 주소 값에 오류가 생기게 되면 순서가 달라지기 때문에, 주소 값에도 오류 정정 부호를 적용할 수 있다.

구체적으로, Reed-Solomon(RS) 부호를 적용하여 부호화할 수 있다. 이때, GF(2^m)에서 단축된 (shortened) RS 부호 $\left(\frac{n_{index}}{m}, \frac{k_{index}}{m}, d_{min}\right)$를 적용하여, 전체 길이 n_{index} bit = $\frac{n_{index}}{m}$ symbol, 주소 값 길이 k_{index} bit = $\frac{k_{index}}{m}$ symbol, 최소 해밍 거리 (minimum hamming distance) d_{min}, $\left(\frac{n_{index}}{m} - \frac{k_{index}}{m}\right)/2$ symbol 오류 정정, $\left(\frac{n_{index}}{m} - \frac{k_{index}}{m}\right)/2$ +1 symbol 오류까지 찾아낼 수 있다.

[0099] 예를 들어, m=4, n_{index}=32, k_{index}=16 인 경우, 실제로 필요한 주소 값은 15 bit 이기 때문에, 16 bit 중 남은 1 bit는 짝수 패리티(parity)를 붙여 주어, 1 bit 오류를 찾아낼 수 있도록 해줄 수 있다. 따라서 부호화한 주소 값은 총 32 bit가 된다. 이렇게 만들어진 주소 값을 순서대로 각각 랜덤화된 데이터에 붙여주어 길이가 제4 길이+제2 길이 (n_{index}+$n_{payload}$) bit 인 벡터를 제2 개수(N_{LDPC}개)만큼 생성한다.

[0100] 그리고 2 bit당 하나의 염기로 치환하면, 최종적으로 길이 $\left(\frac{n_{index}+n_{payload}}{2}\right)$ nt인 염기 배열이 N_{LDPC}개 생성된다. 예를 들어, n_{index}가 32이고, $n_{payload}$가 272인 경우, 152nt의 염기 배열이 N_{LDPC}개 생성될 수 있다.

[0101] 단계 S160에서, 상기 2 비트는 0 및 1의 개수에 따라 아데닌(A, Adenine), 구아닌(G, Guanine), 시토신(C, Cytosine) 및 티민(T, Thymine) 중 하나로 치환될 수 있다. 도 5를 참조하면, 2비트가 00이면 A로 치환되고, 01이면 C로 치환되고, 10이면 G로 치환되고, 11이면 T로 치환될 수 있다.

[0102] 단계 S170에서, 프로세서(16)는 상기 제2 개수의 염기 배열 중에서, 구아닌(G, Guanine) 및 시토신(C, Cytosine)의 비율이 기준 비율과의 차이가 기 설정된 값보다 큰 염기 배열과 동일한 염기가 기 설정된 횟수 이상 반복되는 염기 배열에 대해서 천공을 수행할 수 있다.

[0103] 구체적으로, 50 퍼센트를 기준으로 했을 때 최종 생성된 N_{LDPC} 개의 염기 배열 중에서 G와 C의 비율의 차이가 기 설정된 값, 예를 들어 30보다 크면(즉, G와 C의 비율이 80퍼센트 이상이거나 20퍼센트 이하인 경우), 해당 염기 배열에 대해서 천공을 수행할 수 있다. 또한, 최종 생성된 N_{LDPC} 개의 염기 배열 중에서 특정 염기가 기 설정된 횟수, 예를 들어 4번 이상 반복되는 경우, 해당 염기 배열에 대해서 천공을 수행할 수 있다.

[0104] 최종 생성된 염기 배열들의 G와 C의 비율과, 연속된 염기의 개수를 확인하여, 미리 설정된 기준에서 크게 벗어는 특정 염기 배열들을 찾아내어 제거할 수 있다. 이러한 염기 배열들은 데이터 복원 시 오류율을 높이기 때문에 제거할 필요성이 있다. 또한, 본 발명은 LDPC 부호를 이용하여 부호화를 하기 때문에, 염기 배열 천공(puncturing)을 통해 특정 올리고 시퀀스들을 합성하지 않음으로써 오류를 유발하는 생화학적 특성들을 가지고 있는 염기 배열들을 제외할 수 있다.

[0105] 이하에서는 도 6 및 도 7을 참조하여, 프로세서(16)가 저밀도 패리티 체크 부호를 이용한 DNA 데이터를 복호화하는 방법을 설명하도록 한다. 여기서, 프로세서(16)의 동작은 장치(10)에서 수행 가능할 수 있다.

[0106] 도 6을 참조하면, 프로세서(16)는 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행할 수 있다(S210). 여기서, N은 자연수일 수 있다.

[0107] 상기 N개의 염기 배열은 서로 다르거나 적어도 두개가 동일할 수 있다. 예를 들어, 4개의 염기 배열이 추출됐다고 가정했을 때, 4개의 염기 배열은 AT, AC, TG, CG와 같이 각각 다를 수 있고, 또는 AT, AC, CG, CG와 같이 4개 중 2개의 염기 배열이 중복될 수도 있다.

[0108] 이때, 상기 N개의 염기 배열이 서로 다른 경우에 상기 N은 상기 제2 개수보다 작거나 동일할 수 있다.

[0109] 상기 N개의 염기 배열이 적어도 두개가 동일한 경우에 상기 N은 상기 제2 개수보다 작을수도 있고 클 수도 있으며, 상기 제2 개수와 동일할 수도 있다. 즉, 제2 개수의 염기 배열 중에서 N개의 염기 배열을 추출하는데 이때 특정 염기 배열이 중복되게 추출하는 경우, 중복되는 염기 배열의 개수에 따라 추출된 N개의 염기 배열이 제2 개수의 염기 배열보다 적을 수도 있고 많을 수도 있고 두개가 동일할 수도 있다.

[0110] 도 7을 참조하면, 단계 S210에서, 프로세서(16)는 상기 추출된 N개의 염기 배열 각각의 주소 값에 대한 RS 복호

화 및 짝수 패리티 복호화를 수행할 수 있다(S211).

[0111] 다음으로, 프로세서(16)는 상기 복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑할 수 있다(S212).

[0112] 복호화 과정에서는 저장된 DNA 염기 배열들을 오류 없이 완벽하게 기존의 데이터로 복원해야 한다.

[0113] 그러기 위해서 먼저, 장치(10)에 저장되어 있는 부호화된 DNA 염기 배열들을 무작위로 추출하여 읽어낼 수 있다. 이때, 읽어낸 염기 배열들 중 길이가 $\left(\frac{n_{index}+n_{payload}}{2}\right)$ nt 인 염기 배열들만 남겨 놓는다. 이렇게 무작위로 추출한 같은 길이의 DNA 염기 배열들의 주소 값을 먼저 찾는다. 염기 배열의 앞 n_{index} bit = $\frac{n_{index}}{2}$ nt를 확인하여 RS 부호의 복호화를 수행할 수 있다. 이때, 복호화에 실패하면 특정 염기 배열은 버리고, 그 염기 배열이 N_{LDPC} 개의 주소 값에 해당하는지를 확인할 수 있다. 부호화 과정에서 설명한 방법으로 복호화 과정에서도 정확히 N_{LDPC} 가지의 주소 값을 찾아낼 수 있다. 따라서, N_{LDPC} 개의 주소 값에 해당하지 않는다면, 부호화한 데이터가 아니기 때문에 해당 데이터를 버린다. N_{LDPC} 가지의 주소 값에 해당하면 마지막으로 짝수 패리티를 확인하여, 이를 만족하지 않는다면 해당 데이터를 버린다. 그리고 남은 DNA 염기 배열들을 같은 주소 값끼리 모으고, 빈 주소 값의 데이터는 남겨놓는다.

[0114] 다음으로, 프로세서(16)는 상기 그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환할 수 있다(S213).

[0115] 다음으로, 프로세서(16)는 상기 주소 값에 염기 배열의 존재 여부에 따라 상기 치환된 이진수 벡터 각각에 대해 로그 우도 비 (LLR, Log-Likelihood Ratio)를 산출할 수 있다(S214).

[0116] 다음으로, 프로세서(16)는 상기 산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬할 수 있다(S215).

[0117] 다음으로, 프로세서(16)는 상기 세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행할 수 있다(S216).

[0118] 다음으로, 프로세서(16)는 상기 제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행할 수 있다(S217).

[0119] 다음으로, 프로세서(16)는 상기 제2 LDPC 복호화된 결과에 기초하여, 상기 최종 복호화의 성공 여부를 판단할 수 있다(S218).

[0120] 본 발명은 실시예에 따라, LDPC 부호의 부호의 경판정 복호법(hard-decision decoding) 및 연판정 복호법(soft-decision decoding)을 이용할 수 있다.

[0121] 경판정 복호법을 사용하는 경우에는, 복호화된 주소 값을 통해 같은 주소 값을 가지는 염기 배열들을 모아, 첫 번째 주소 값부터 모인 염기 배열들에서 주소 값을 제거한 후, 도 5에 도시된 표를 이용하여 이진수로 치환할 수 있다. 이때, 주소 값에 모인 염기 배열이 없으면 빈 공간으로 남겨두고, 주소 값에 염기 배열이 하나라도 존재하면, 각 위치마다 0 또는 1 중에 더 많은 값으로 대표 값을 지정하여 모든 주소 값에 대해 하나의 염기 배열만을 선택하도록 할 수 있다.

[0122] 연판정 복호법을 사용하는 경우에는, 모든 주소 값에 대해 염기 배열의 각 비트에 대하여 로그 우도 비 (LLR, Log-Likelihood Ratio)를 산출할 수 있다.

[0123] 단계 S214에서, 상기 로그 우도 비 (LLR, Log-Likelihood Ratio)는 아래 [수학식 1]을 이용하여 산출될 수 있다.

수학식 1

[0124]
$$LLR = \log \frac{p(k_0, k_1 \mid 0)}{p(k_0, k_1 \mid 1)} = (k_0 - k_1) \log \frac{1-\epsilon}{\epsilon}$$

[0125] 여기서, K_0은 같은 클러스터 내에서 같은 위치에서의 0의 개수, K_1은 같은 클러스터 내에서 같은 위치에서의 1의 개수, ϵ는 기 설정된 파라미터 값일 수 있다.

[0126] 상술한 바와 같이, 복호화된 주소 값을 통해 같은 주소 값을 가지는 염기 배열들을 그룹핑하고, 첫 번째 주소 값부터 모인 염기 배열들에서 주소 값을 제거한 후, 도 5에 도시된 표를 이용하여 이진수로 치환할 수 있다. 이때, 주소 값에 모인 염기 배열이 없으면 로그 우도 비를 전부 0으로 지정할 수 있다. 주소 값에 염기 배열이 하나라도 존재하면, 치환된 이진수 벡터들에 대해 각 위치에서의 0의 개수와 1의 개수를 세고, 각각을 k_0, k_1이라 한다. 이때, 이전 단계에서 모두 같은 길이의 염기 배열들만 사용했기 때문에, 이진수 벡터의 길이도 모두 같게 된다. k_0, k_1 그리고 실험적인 값 ϵ =0.04을 이용하여 상기 [수학식 1]을 계산한다. 그 결과, 모든 주소 값의 염기 배열의 각 위치마다 로그 우도 비를 구할 수 있다. 로그 우도 비를 주소 값 순서대로 세로 방향으로 배열하고, 가로 방향이 LDPC 부호가 된다. 따라서, 최종적으로 길이 N_{LDPC}인 LDPC 부호 $n_{payload}$가 도출되고, $n_{payload}$ 개의 부호를 오류 없이 완벽히 복호화하면 전체 복호화가 성공한 것이라 할 수 있다.

[0127] 본 발명의 일 실시예에 따라, 길이 N_{LDPC}인 $n_{payload}$개의 부호들에 대해 모두 연판정 복호법을 수행할 수 있다.

[0128] 단계 S216에서, 프로세서(16)는 각 주소 값에 대응하는 염기 배열 각각에 대해, 제1 LDPC 복호화 수행 전과 후의 바뀐 비트 수를 계산하고, 상기 바뀐 비트 수가 기 설정된 개수 이상인 염기 배열에 대해 상기 로그 우도 비를 0으로 초기화할 수 있다.

[0129] 단계 S217에서, 프로세서(16)는 상기 제1 LDPC 복호화에 실패한 가로방향 LPDC 부호어에 대해서만 제2 LDPC 복호화를 수행할 수 있다.

[0130] 단계 S218에서, 프로세서(16)는 상기 제1 LDPC 복호화된 모든 염기 배열과, 상기 제1 LDPC 복호화 및 상기 제2 LDPC 복호화된 모든 염기 배열에 대한 패리티 검사 행렬을 통해 최종 복호화의 성공 여부를 판단할 수 있다.

[0131] 보다 구체적으로, $n_{payload}$개의 부호에 대해 연판정 복호법을 모두 수행한 후, N_{LDPC}개의 각 위치에 대해서 복호화 전후를 비교하여 몇 개의 비트(bit)가 고쳐졌는지 확인할 수 있다. 다시 말해, N_{LDPC}개의 각 위치, 주소 값에 각각 $n_{payload}$ 비트(bit)가 존재하는데, $n_{payload}$ 비트(bit) 중 복호화 후에 바뀐 비트(bit)수를 계산한다. 이렇게 바뀐 비트 수를 계산하는 이유는 바뀐 비트 수가 많다면, 그 염기 배열은 오류가 많았던 배열이라 생각할 수 있고, 오류가 많았다면 전체적으로 연판정 복호화 과정에서 다른 위치들에 안 좋은 영향을 미치는 이진수 벡터일 가능성이 크기 때문이다. 따라서, 각 위치마다 δ 값을 정하여, 최대 $n_{payload}$ 비트(bit) 중 δ 비트(bit) 이상이 바뀌었다면, 복호화에 안 좋은 영향을 미칠 가능성이 크기 때문에 그 위치의 세로로 구성된 이진수 벡터의 로그 우도 비를 0으로 초기화할 수 있다. 정해진 δ 값보다 많은 bit가 바뀐 세로의 이진수 벡터를 모두 초기화시킨 후, 첫 번째 연판정 복호화(제1 LDPC 복호화)에서 복호화에 실패한 부호들만 다시 복호화(제2 LDPC 복호화)를 진행한다. 이때, 최종 복호화의 성공 및 실패 여부는 LDPC 부호의 패리티 검사 행렬을 통해 확인한다. 첫 번째 복호화에서 성공한 부호들은 성공한 값 그대로를 사용한다.

[0132] 또한, 본 발명의 일 실시예에 따라, 프로세서(16)는 제1 LDPC 복호화에서, 같은 주소 값에 대한 복호 전후의 염기 배열의 편집 거리를 구할 수 있다. 여기서, 편집 거리란 두 염기 배열의 유사도를 알려주는 값으로, 치환, 삽입 및 삭제 오류의 개수를 알려줄 수 있다. 따라서, 편집 거리에 비해 해밍 거리가 크다면, 기존의 염기 배열에 삽입/삭제 오류가 존재했음을 추측할 수 있어, 편집 거리를 이용하여 삽입/삭제 오류의 여부를 판단하고 그 위치를 찾아낼 수 있다.

[0133] 다음으로, 프로세서(16)는 상기 복호화된 결과에 기초하여, 상기 제1 개수의 제2 이진수 벡터를 상기 지정된 주소 값의 순서에 따라 상기 제1 길이의 제1 이진수 벡터로 통합할 수 있다(S220).

[0134] 상술한 바와 같이, 제1 LDPC 복호화 및 제1 LDPC 복호화가 모두 완료되었을 때, 부호 하나라도 패리티 검사 행렬을 통과하지 못했다면 복호화가 실패한 것으로 판단될 수 있다.

[0135] 반면에, 제1 LDPC 복호화 및 제1 LDPC 복호화가 모두 완료되었을 때, 부호들이 모두 패리티 검사 행렬을 통과했을 경우에는, 세로 방향의 제2 이진수 벡터들을 하나의 긴 벡터(제1 이진수 벡터)로 통합할 수 있다. 하나의 긴 벡터는 기존의 데이터를 난수와 XOR 시킨 결과 값이기 때문에, 복호화 결과를 다시 같은 난수 배열과 XOR 시켜서 기존의 데이터를 복원할 수 있다. 이때, 복원된 데이터가 기존의 데이터와 일치하면 장치(10)를 이용한 데이

터의 저장 및 복원이 성공한 것으로 판단할 수 있다.

[0136] 한편, 단계 S218에서, 상기 최종 복호화가 실패한 것으로 판단된 경우, 프로세서(16)는 상기 부호화된 상기 제2 개수의 염기 배열 중 N개의 염기 배열을 추가로 추출하여 제1 LDPC 복호화 및 제2 LDPC 복호화를 수행하는 과정을 반복할 수 있다.

[0137] 도 2, 도 6 및 도 7은 단계 S110 내지 단계 S218을 순차적으로 실행하는 것으로 기재하고 있으나, 이는 본 실시예의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 실시예가 속하는 기술분야에서 통상의 지식을 가진 자라면 본 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 2, 도 6 및 도 7에 기재된 순서를 변경하여 실행하거나 단계 S110 내지 단계 S218 중 하나 이상의 단계를 병렬적으로 실행하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이므로, 도 2, 도 6 및 도 7은 시계열적인 순서로 한정되는 것은 아니다.

[0138] 이상에서 전술한 본 발명의 일 실시예에 따른 방법은, 하드웨어인 컴퓨터와 결합되어 실행되기 위해 프로그램(또는 어플리케이션)으로 구현되어 매체에 저장될 수 있다. 여기서, 컴퓨터는 앞에서 설명한 장치(10)일 수 있다.

[0139] 상기 전술한 프로그램은, 상기 컴퓨터가 프로그램을 읽어 들여 프로그램으로 구현된 상기 방법들을 실행시키기 위하여, 상기 컴퓨터의 프로세서(CPU)가 상기 컴퓨터의 장치 인터페이스를 통해 읽힐 수 있는 C, C++, JAVA, 기계어 등의 컴퓨터 언어로 코드화된 코드(Code)를 포함할 수 있다. 이러한 코드는 상기 방법들을 실행하는 필요한 기능들을 정의한 함수 등과 관련된 기능적인 코드(Functional Code)를 포함할 수 있고, 상기 기능들을 상기 컴퓨터의 프로세서가 소정의 절차대로 실행시키는데 필요한 실행 절차 관련 제어 코드를 포함할 수 있다. 또한, 이러한 코드는 상기 기능들을 상기 컴퓨터의 프로세서가 실행시키는데 필요한 추가 정보나 미디어가 상기 컴퓨터의 내부 또는 외부 메모리의 어느 위치(주소 번지)에서 참조되어야 하는지에 대한 메모리 참조관련 코드를 더 포함할 수 있다. 또한, 상기 컴퓨터의 프로세서가 상기 기능들을 실행시키기 위하여 원격(Remote)에 있는 어떠한 다른 컴퓨터나 서버 등과 통신이 필요한 경우, 코드는 상기 컴퓨터의 통신 모듈을 이용하여 원격에 있는 어떠한 다른 컴퓨터나 서버 등과 어떻게 통신해야 하는지, 통신 시 어떠한 정보나 미디어를 송수신해야 하는지 등에 대한 통신 관련 코드를 더 포함할 수 있다.

[0140] 본 발명의 실시예와 관련하여 설명된 방법 또는 알고리즘의 단계들은 하드웨어로 직접 구현되거나, 하드웨어에 의해 실행되는 소프트웨어 모듈로 구현되거나, 또는 이들의 결합에 의해 구현될 수 있다. 소프트웨어 모듈은 RAM(Random Access Memory), ROM(Read Only Memory), EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 플래시 메모리(Flash Memory), 하드 디스크, 착탈형 디스크, CD-ROM, 또는 본 발명이 속하는 기술 분야에서 잘 알려진 임의의 형태의 컴퓨터 판독가능 기록매체에 상주할 수도 있다.

[0141] 이상, 첨부된 도면을 참조로 하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야의 통상의 기술자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로, 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며, 제한적이 아닌 것으로 이해해야만 한다.

부호의 설명

[0142] 10: 장치

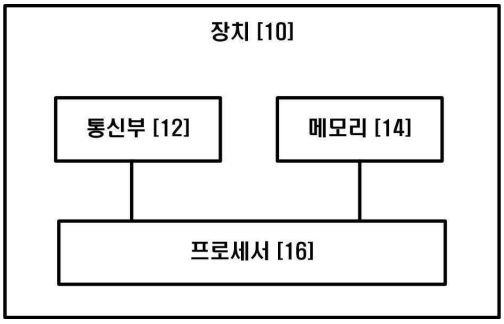
12: 통신부

14: 메모리

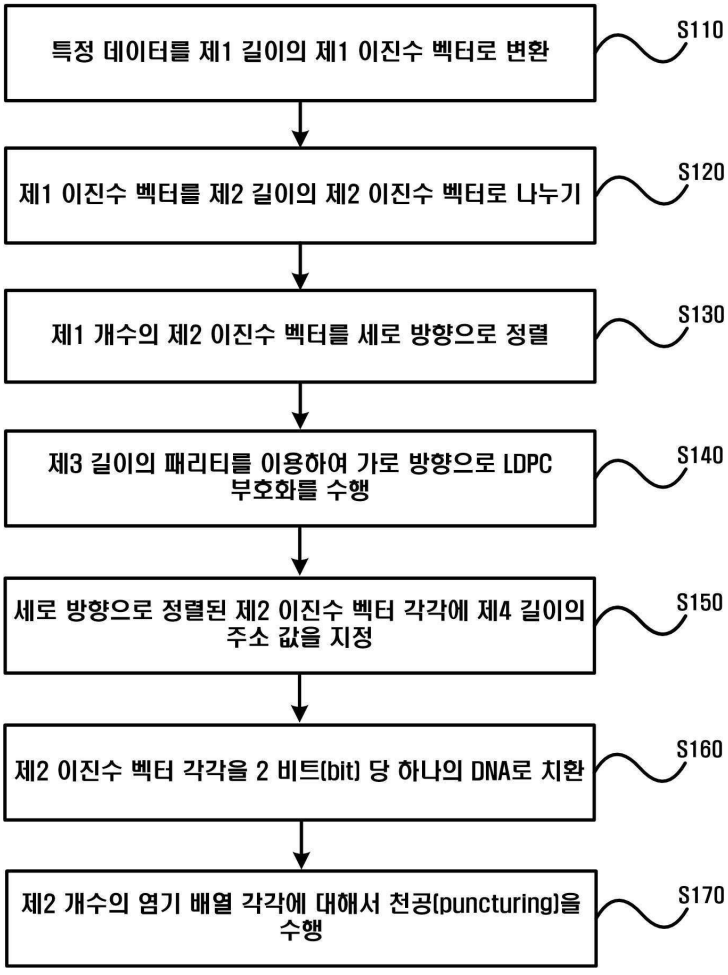
16: 프로세서

도면

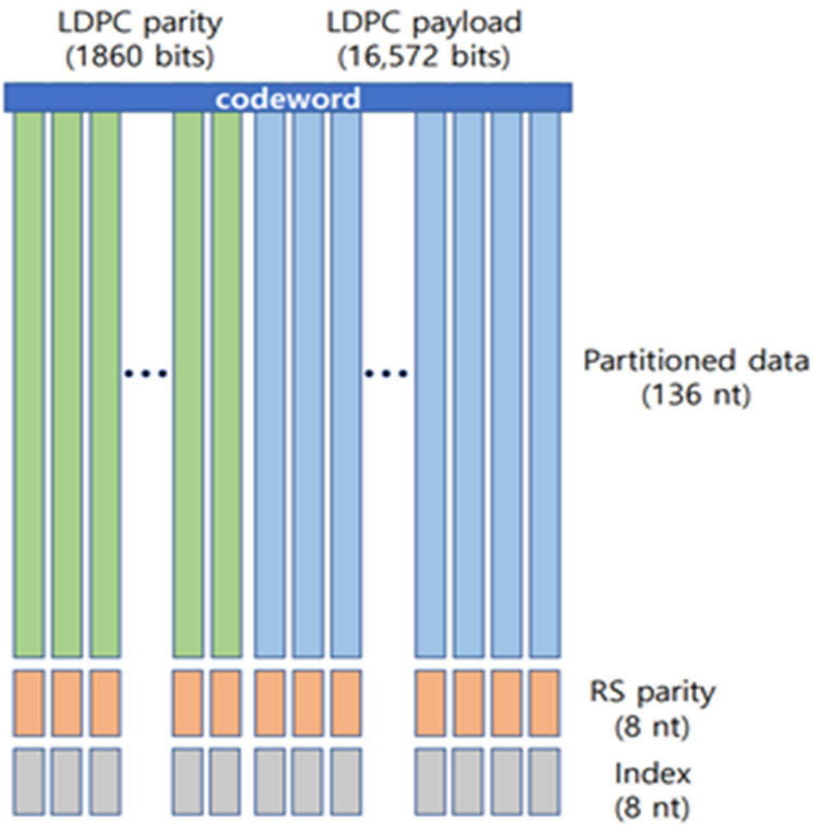
도면1


장치 [10]
통신부 [12]
메모리 [14]
프로세서 [16]

도면2


특정 데이터를 제1 길이의 제1 이진수 벡터로 변환
S110
제1 이진수 벡터를 제2 길이의 제2 이진수 벡터로 나누기
S120
제1 개수의 제2 이진수 벡터를 세로 방향으로 정렬
S130
제3 길이의 패리티를 이용하여 가로 방향으로 LDPC 부호화를 수행
S140
세로 방향으로 정렬된 제2 이진수 벡터 각각에 제4 길이의 주소 값을 지정
S150
제2 이진수 벡터 각각을 2 비트(bit) 당 하나의 DNA로 치환
S160
제2 개수의 염기 배열 각각에 대해서 천공(puncturing)을 수행
S170

도면3



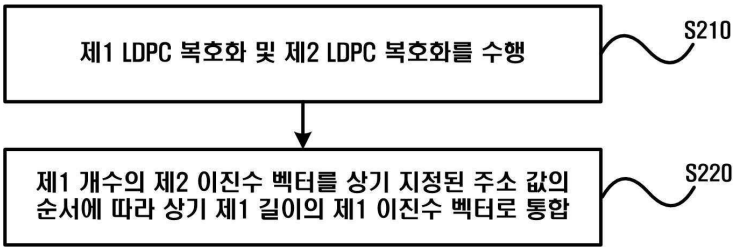
도면4

경우의 수	이진수
0	000000000000000
1	000000000000001
2	000000000000010
3	000000000000011
4	000000000000100
5	000000000000101
⋮	⋮

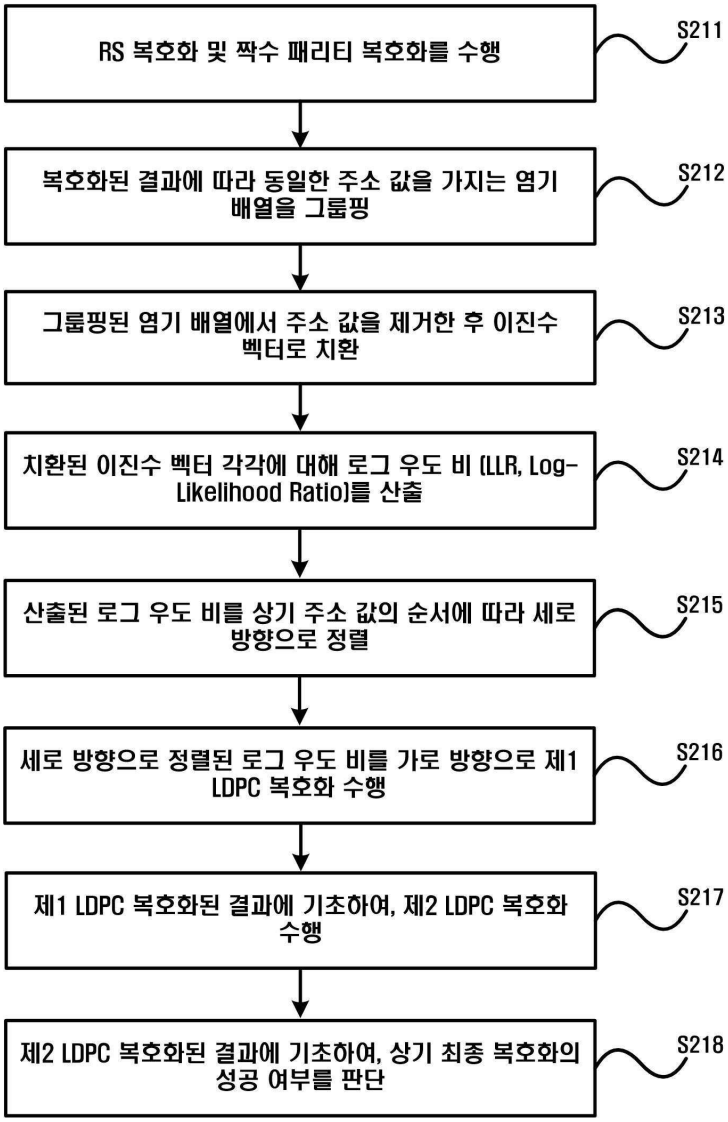
도면5

이진수	염기
00	A
01	C
10	G
11	T

도면6



도면7


RS 복호화 및 짝수 패리티 복호화를 수행
S211
복호화된 결과에 따라 동일한 주소 값을 가지는 염기 배열을 그룹핑
S212
그룹핑된 염기 배열에서 주소 값을 제거한 후 이진수 벡터로 치환
S213
치환된 이진수 벡터 각각에 대해 로그 우도 비 (LLR, Log-Likelihood Ratio)를 산출
S214
산출된 로그 우도 비를 상기 주소 값의 순서에 따라 세로 방향으로 정렬
S215
세로 방향으로 정렬된 로그 우도 비를 가로 방향으로 제1 LDPC 복호화 수행
S216
제1 LDPC 복호화된 결과에 기초하여, 제2 LDPC 복호화 수행
S217
제2 LDPC 복호화된 결과에 기초하여, 상기 최종 복호화의 성공 여부를 판단
S218