



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0103910
(43) 공개일자 2013년09월25일

(51) 국제특허분류(Int. Cl.)

H03M 13/11 (2006.01)

(21) 출원번호 10-2012-0024917

(22) 출원일자 2012년03월12일

심사청구일자 없음

(71) 출원인

에스케이하이닉스 주식회사

경기도 이천시 부발읍 경충대로 2091

고려대학교 산학협력단

서울 성북구 안암동5가 1

(72) 발명자

박종선

서울시 서대문구 홍제동 남양아파트 103-305

윤지환

서울시 강북구 삼양동 776-125번지 101호

(74) 대리인

특허법인 이노

전체 청구항 수 : 총 9 항

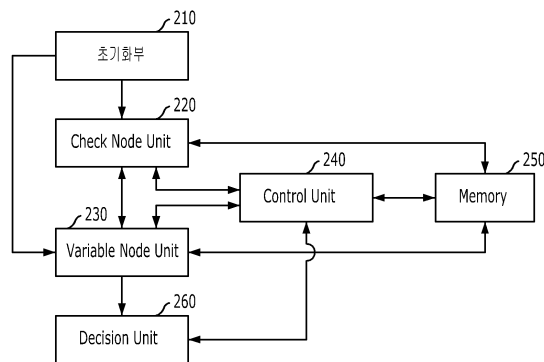
(54) 발명의 명칭 저밀도 패리티 검사 부호용 장치

(57) 요약

본 발명은 체크 노드의 연산 수행 결과를 복수의 블록 열에 대한 가변 노드 연산에서 공유함으로써 지연 시간을 절감할 수 있고, 메모리 읽기 동작의 횟수를 경감할 수 있는 저밀도 패리티 검사 부호용 장치를 제공한다.

본원의 제1 발명에 따른 저밀도 패리티 검사 부호용 장치는, 체크 노드 유닛, 가변 노드 유닛, 제어 유닛 및 메모리를 포함하는 저밀도 패리티 검사 부호용 장치에 있어서, 상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 하나의 뱅크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 제어 유닛은 상기 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트시켜 상기 서로 다른 어드레스를 상호 일치시킨다.

대표도 - 도2



특허청구의 범위

청구항 1

체크 노드 유닛, 가변 노드 유닛, 제어 유닛 및 메모리를 포함하는 저밀도 패리티 검사 부호용 장치에 있어서, 상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 하나의 뱅크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 제어 유닛은 상기 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 상기 서로 다른 어드레스를 상호 일치시키는 것을 특징으로 하는 저밀도 패리티 검사 부호용 장치.

청구항 2

저밀도 패리티 검사 부호에 의해 부호화된 신호가 전송됨에 따라, 상기 저밀도 패리티 검사 부호를 위한 검사 행렬에 포함된 가변 노드에 대하여 초기화부;

각 체크 노드에 연결된 모든 다른 가변 노드로부터의 정보를 바탕으로 가변 노드의 신뢰를 갱신하는 체크 노드 유닛;

상기 가변 노드의 연산을 수행하는 가변 노드 유닛;

상기 가변 노드 유닛이 연산한 결과값의 부호를 이용하여 상기 결과값의 부호에 따라 이진으로 판정하는 판정 유닛; 및

상기 가변 노드 유닛이 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 상기 가변 노드의 연산을 동시에 수행할 때, 상기 메모리 내 하나의 뱅크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 상기 서로 다른 어드레스를 상호 일치시키는 제어 유닛

을 포함하는 저밀도 패리티 검사 부호용 장치.

청구항 3

제1항 또는 제2항에 있어서, 상기 제어 유닛은,

상기 가변 노드 유닛과 상기 메모리를 연결하는 네트워크; 및

상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 뱅크 내에서 서로 다른 어드레스의 데이터를 동시에 읽으면, 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 서로 다른 어드레스를 상호 일치시키는 서브 컨트롤러

를 포함하는 저밀도 패리티 검사 부호용 장치.

청구항 4

제1항 또는 제2항에 있어서, 상기 제어 유닛은,

상기 제어 유닛은, 상기 가상의 복수의 블록 열에 관하여 상기 메모리 내 나머지 뱅크의 어드레스를 상기 소정 간격 쉬프트 시키는 것을 특징으로 하는 저밀도 패리티 검사 부호용 장치.

청구항 5

제3항에 있어서,

상기 메모리 내 나머지 뱅크에서의 어드레스 간에는 충돌이 없는 것을 특징으로 하는 저밀도 패리티 검사 부호용 장치.

청구항 6

체크 노드 유닛, 가변 노드 유닛, 제어 유닛 및 메모리를 포함하는 저밀도 패리티 검사 부호용 장치에 있어서,

상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 복수 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 제어 유닛은 상기 복수 बैं크 중 어느 하나의 बैं크에 대하여 서로 다른 어드레스 중 어느 하나의 어드레스를 소정 간격 쉬프트시켜 상호 일치시키는 것을 특징으로 하는 저밀도 패리티 검사 부호용 장치.

청구항 7

저밀도 패리티 검사 부호에 의해 부호화된 신호가 전송됨에 따라, 상기 저밀도 패리티 검사 부호를 위한 검사 행렬에 포함된 가변 노드에 대하여 초기화부;

각 체크 노드에 연결된 모든 다른 가변 노드로부터의 정보를 바탕으로 가변 노드의 신뢰를 갱신하는 체크 노드 유닛;

상기 가변 노드의 연산을 수행하는 가변 노드 유닛;

상기 가변 노드 유닛이 연산한 결과값의 부호를 이용하여 상기 결과값의 부호에 따라 이진으로 판정하는 판정 유닛; 및

상기 가변 노드 유닛이 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 복수 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 복수 बैं크 중 어느 하나의 बैं크에 대하여 서로 다른 어드레스 중 어느 하나의 어드레스를 소정 간격 쉬프트시켜 상호 일치시키는 제어 유닛

을 포함하는 저밀도 패리티 검사 부호용 장치.

청구항 8

제6항 또는 제7항에 있어서,

상기 제어 유닛은,

상기 가변 노드 유닛과 상기 메모리를 연결하는 네트워크;

상기 복수 बैं크 중 상기 어느 하나의 बैं크를 제외한 나머지 बैं크에 대하여 상기 쉬프트 후 불일치하는 어드레스 차이를 보상하기 위한 레지스터 어레이; 및

상기 레지스터 어레이의 갱신되는 데이터를 가변 노드 유닛에 공급하는 서브 컨트롤러

를 포함하는 저밀도 패리티 검사 부호용 장치.

청구항 9

제8항에 있어서,

상기 서브 컨트롤러는, 상기 메모리 내 해당 어드레스의 데이터를 읽어들이어 상기 레지스터 어레이 내 최후 데이터를 갱신하고, 상기 레지스터 어레이 내 최선 데이터를 상기 가변 노드 유닛으로 출력하는 것을 특징으로 하는 저밀도 패리티 검사 부호용 장치.

명세서

기술분야

[0001] 본 발명은 패리티 검사 부호에 관한 것으로, 더욱 상세하게는 저밀도 패리티 검사(Low Density Parity Check, 이하, 'LDPC'라 한다) 부호용 장치에 관한 것이다.

배경기술

[0002] 최근, LDPC 부호는 쉐넌 리미트(Shannon limit)에 가까운 강력한 에러 보정 능력을 가지는 오류 정정 부호로서 차세대 무선 통신 기술에서 폭 넓게 상용화될 것으로 각광받고 있다. 즉, DVB, WLAN, WiMAX등의 통신 표준뿐 아니라 4세대 통신 표준인 LTE에서도 주된 화두의 하나로 포함되어 논의되고 있다.

[0003] 일반적으로, LDPC 부호는 '(N, K) LDPC 부호'와 같은 방식으로 표기하는데 여기서 K는 통신 모듈의 송신부에서

부호화하는 비트들의 수이고, N 은 부호화의 결과로 생성된 블록의 비트 수이다. 부호화의 결과로 생성된 N 비트는 부호화하는 K 비트를 그대로 포함하면서 $(N-K)$ 비트가 추가적으로 덧붙여진 것이다. 여기서 추가적으로 덧붙여진 $(N-K)$ 비트는 K 비트에 오류가 발생하였을 경우, 이를 정정하는데 결정적인 역할을 한다.

[0004] 통신의 수신부에서는 송신부에서 전송된 LDPC 부호를 수신한 후, 복호기를 통해 N 비트의 LDPC 부호를 K 비트로 복원하는 과정을 수행하는데, 이 과정에서 핵심 기반이 되는 것이 '이진 패리티 체크 행렬'이라 할 수 있다. 이진 패리티 체크 행렬은 $(N-K)$ 행 N 열의 크기를 가지며, LDPC 복호 과정의 핵심 연산과 직결되는 구조를 포함하고 있다. LDPC 복호과정은 이분 그래프로 표현할 수 있는데, 이 이분 그래프의 두 개 유형의 노드 중 체크 노드는 이진 패리티 체크 행렬의 $(N-K)$ 행에 각각 대응되어 $(N-K)$ 개 존재하며, 또 다른 유형의 노드인 비트 노드는 이진 패리티 체크 행렬의 N 열에 각각 대응하여 N 개 존재한다. 이분 그래프의 에지는 서로 다른 유형의 노드 간에만 연결되고, 이진 패리티 체크 행렬의 m 행 n 열의 성분을 $H(m,n)$ 이라 하면(여기서 $0 \leq m < M$, $0 \leq n < N$), 이 성분이 1일 경우에만 m 행에 대응하는 체크노드와 n 행에 대응하는 가변 노드가 연결된다. 이러한 방식에 의해 이진 패리티 체크 행렬의 모든 성분을 살펴 $(N-K)$ 개의 체크 노드와 N 개의 가변 노드 간의 에지 연결을 완료하면, LDPC 핵심 복호과정이 모두 이분 그래프로 표현되게 된다.

[0005] 또한 완성된 이분 그래프는 LDPC 복호기의 구조와 매우 유사해진다. 체크 노드는 복호기의 체크 노드 프로세서에 대응되고, 가변 노드는 복호기의 가변 노드 프로세서에 대응된다.

[0006] 두 핵심 프로세서는 복호기 내부에서 번갈아 동작하며 연산을 수행한다. 즉, 체크 노드 프로세서가 입력되는 가변 노드에 대하여 연산을 수행하여 체크 노드 연산을 완료하면, 완료된 체크 노드를 바탕으로 가변 노드 프로세서는 입력되는 체크 노드에 대하여 가변 노드 연산을 수행한다. 가변 노드 연산 수행 후에는 그 결과 값을 활용하여 복호를 시도하여 이에 성공하면 연산을 멈추고, 실패하면 다시 순환적으로 다음 번 체크 노드 연산과 이에 이은 가변 노드 연산을 수행하는 것이 LDPC 복호기가 수행하는 연산의 개요라고 할 수 있다.

[0007] 종래 기술에 따르면, LDPC 장치의 하드웨어 구조를 3가지로 구분할 수 있다. 구체적으로, 1) 첫번째, 모든 체크 노드와 가변 노드가 개개의 하드웨어에 대응되는 완전 병렬 구조, 2) 두번째, 단지 하나의 체크 노드 프로세서와 가변 노드 프로세서가 모든 복호 연산을 수행하기 위해 반복적으로 재사용되는 완전 직렬 구조, 그리고, 3) 세번째, 완전 병렬 구조와 완전 직렬 구조를 절충한 부분 병렬 구조이다.

[0008] 일반적으로, 완전 병렬 구조는 하드웨어 구현의 복잡도가 매우 높아 잘 채택되지 않으며, 완전 직렬 구조의 경우, 복호 지연 시간이 너무 길어 잘 채택되지 않는다. 따라서 하드웨어 복잡도와 복호 지연 시간을 절충한 부분 병렬 구조가 많이 채택되고 있는 추세이며 LDPC 블록 부호와 같은 이진 패리티 체크 행렬이 정의되어 부분 병렬 구조를 이용한 구현의 적합도를 높이고 있다.

[0009] 부분 병렬 구조가 하드웨어 복잡도와 복호 지연 시간 양면에서 이득을 가지는 것은 사실이지만, 모든 체크 노드 및 가변 노드의 연산을 동시에 수행하지 않기 때문에 체크 노드 연산의 결과가 임베디드 SRAM에 저장되어 수차례에 걸쳐 반복적으로 수행하는 가변 노드 프로세서 연산을 대비하여야 한다. 이러한 임베디드 SRAM 사용상의 특성으로 인해 추가적인 지연 시간이 발생하고 있다. 하지만 아직까지 임베디드 SRAM 사용으로 인해 추가 지연 시간을 대폭 절감하기 위한 접근법이 아직 효과적으로 제시되지 않고 있다.

[0010] 도 1은 종래기술에 따른 부분 병렬 구조의 LDPC 복호기의 기반이 되는 이진 패리티 체크 행렬이다.

[0011] 이 행렬은 행이 6개의 블록, 열이 12개의 블록으로 구성되어 있으므로 총 72개의 블록으로 구성되어 있으며, 각 블록은 80행 80열로 구성되어 있다. 따라서 총 480행 960열의 이진 패리티 행렬을 나타낸다. 여기서, 검은색으로 표기된 블록은 영행렬을 의미하고, 나머지 인덱스는 인덱스의 수만큼 오른쪽으로 Cyclic shift된 Identity 행렬을 의미한다.

[0012] 도 1의 QC-LDPC 블록 부호를 기반으로 부분 병렬 복호기를 구현할 때, 복호 연산의 기본 단위는 단위 블록 행렬이라고 할 수 있다. 이때, 복호 연산 과정에서 우선적으로 각 블록 행렬에 대한 체크 노드 프로세서 연산을 완료하면, 단일 블록 행 내부에 있는 행에 대한 체크 노드 프로세서의 연산 수행 결과는 동일한 SRAM 메모리 बैं크의 서로 다른 주소에 기록된다.

[0013] 이와 같은 SRAM 기록 과정은 체크 노드 프로세서의 연산 수행 결과가 여러 열에 대한 가변 노드 프로세서의 연산 수행이 필요하기 때문이다. 예를 들면, 블록 4행에 대한 체크 노드 프로세서의 연산 수행 결과는 가변 노드 프로세서의 연산 수행 중 블록 1열, 3열, 9열, 10열, 16열, 17열에 대해 모두 필요하게 된다. 하지만 일반적인 1포트 혹은 2포트 SRAM 메모리는 읽기 동작을 할 때, 단일 주소로 밖에 접근할 수 없으므로 같은 메모리의 서로

다른 주소에 있는 값을 요구하는 이들 연산을 동시에 수행할 수 없다.

- [0014] 이러한 이유로 기존의 복호 연산은 복수 개의 블록 열에 대한 가변 노드 연산을 시행하고자 할 경우에 동일한 블록 행 내의 서로 다른 단일 행에 대한 체크 노드 프로세서 연산 결과가 필요하다면, 서로 다른 가변 노드 프로세서의 연산이 동시에 수행될 수 없기 때문에 복호 지연 시간이 매우 길어질 수밖에 없다.

발명의 내용

해결하려는 과제

- [0015] 본 발명은 체크 노드의 연산 수행 결과를 복수의 블록 열에 대한 가변 노드 연산에서 공유함으로써 지연 시간을 절감할 수 있는 저밀도 패리티 검사 부호용 장치를 제공한다.
- [0016] 또한, 본 발명은 메모리 읽기 동작의 횟수를 경감할 수 있는 저밀도 패리티 검사 부호용 장치를 제공한다.

과제의 해결 수단

- [0017] 본원의 제1 발명에 따른 저밀도 패리티 검사 부호용 장치는, 체크 노드 유닛, 가변 노드 유닛, 제어 유닛 및 메모리를 포함하는 저밀도 패리티 검사 부호용 장치에 있어서, 상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 하나의 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 제어 유닛은 상기 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 상기 서로 다른 어드레스를 상호 일치시킨다.
- [0018] 또한, 본원의 제2 발명에 따른 저밀도 패리티 검사 부호용 장치는 저밀도 패리티 검사 부호에 의해 부호화된 신호가 전송됨에 따라, 상기 저밀도 패리티 검사 부호를 위한 검사 행렬에 포함된 가변 노드에 대하여 초기화부; 각 체크 노드에 연결된 모든 다른 가변 노드로부터의 정보를 바탕으로 가변 노드의 신뢰를 갱신하는 체크 노드 유닛; 상기 가변 노드의 연산을 수행하는 가변 노드 유닛; 상기 가변 노드 유닛이 연산한 결과값의 부호를 이용하여 상기 결과값의 부호에 따라 이진으로 판정하는 판정 유닛; 및 상기 가변 노드 유닛이 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 상기 가변 노드의 연산을 동시에 수행할 때, 상기 메모리 내 하나의 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 상기 서로 다른 어드레스를 상호 일치시키는 제어 유닛을 포함한다.
- [0019] 또한, 본원의 제3 발명에 따른 저밀도 패리티 검사 부호용 장치는, 체크 노드 유닛, 가변 노드 유닛, 제어 유닛 및 메모리를 포함하는 저밀도 패리티 검사 부호용 장치에 있어서, 상기 가변 노드 유닛이 상기 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 복수 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 제어 유닛은 상기 복수 बैं크 중 어느 하나의 बैं크에 대하여 서로 다른 어드레스 중 어느 하나의 어드레스를 소정 간격 쉬프트 시켜 상호 일치시킨다.
- [0020] 또한, 본원의 제4 발명에 따른 저밀도 패리티 검사 부호용 장치는, 저밀도 패리티 검사 부호에 의해 부호화된 신호가 전송됨에 따라, 상기 저밀도 패리티 검사 부호를 위한 검사 행렬에 포함된 가변 노드에 대하여 초기화부; 각 체크 노드에 연결된 모든 다른 가변 노드로부터의 정보를 바탕으로 가변 노드의 신뢰를 갱신하는 체크 노드 유닛; 상기 가변 노드의 연산을 수행하는 가변 노드 유닛; 상기 가변 노드 유닛이 연산한 결과값의 부호를 이용하여 상기 결과값의 부호에 따라 이진으로 판정하는 판정 유닛; 및 상기 가변 노드 유닛이 메모리 내 체크 노드 결과값을 읽어들이어 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 상기 메모리 내 복수 बैं크에서 서로 다른 어드레스의 데이터를 동시에 읽는 경우, 상기 복수 बैं크 중 어느 하나의 बैं크에 대하여 서로 다른 어드레스 중 어느 하나의 어드레스를 소정 간격 쉬프트 시켜 상호 일치시키는 제어 유닛을 포함한다.

발명의 효과

- [0021] 본 발명의 저밀도 패리티 검사 부호용 장치에 따르면, 체크 노드의 연산 수행 결과를 여러 개의 블록 열에 대한 가변 노드 연산에서 공유할 수 있고, 메모리 읽기 동작의 횟수를 경감할 수 있는 효과가 있다.

도면의 간단한 설명

- [0022] 도 1은 일반적인 QC-LDPC 블록 부호,

도 2는 본 발명의 일실시예에 따른 LDPC 부호용 장치의 블럭 구성도,

도 3(a)은 동일 बैं크 내 서로 다른 주소의 데이터가 충돌하는 상황에 대한 설명도,

도 3(b)은 본 발명의 일실시예에 따른 단일 बैं크 내 어드레스 충돌을 회피하기 위한 어드레스 쉬프트 기법을 설명하기 위한 도면,

도 4(a)는 복수개의 बैं크 내에서 복수의 어드레스들이 충돌하는 상황에 대한 설명도,

도 4(b)는 본 발명의 다른 실시예에 따른 복수 बैं크 내 어드레스 충돌을 회피하기 위한 어드레스 쉬프트 기법을 설명하기 위한 도면,

도 5(a)(b)(c)는 본 발명의 일 실시예에 따른 레지스터 내 데이터 갱신 패턴 설명도, 및

도 6은 도 5(a)(b)(c)를 구현하기 위한 주요 구성에 대한 블럭도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하, 본 발명의 바람직한 실시예(들)에 대하여 첨부도면을 참조하여 상세히 설명한다. 우선 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호로 표기되었음에 유의하여야 한다. 또한, 하기의 설명에서는 많은 특정사항들이 도시되어 있는데, 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐 이러한 특정 사항들 없이도 본 발명이 실시될 수 있음은 이 기술분야에서 통상의 지식을 가진 자에게는 자명하다 할 것이다.

[0024] 도 2는 본 발명의 일실시예에 따른 LDPC 부호용 장치의 블럭 구성도이다.

[0025] 본 발명의 일실시예에 따른 LDPC 부호용 장치는 초기화부(210), 체크 노드 유닛(220), 가변 노드 유닛(230), 제어 유닛(240), 메모리(250), 및 판정 유닛(260)을 포함한다.

[0026] 초기화부(210)는 가변 노드 각각($v=0, \dots, n-1$)에 대하여 수학식1을 적용하여 초기화를 수행한다.

수학식 1

$$L(q_{cv}) = 2y_v / \sigma^2$$

[0027]

[0028] 여기에서, q_{cv} 는 가변 노드 v 에서 체크 노드 c 로 전달하는 확률 정보를 의미하고, $L(q_{cv})$ 는 가변 노드 v 가 체크 노드 c 로 전달하는 확률이 0인 경우 대한 가변 노드 v 가 체크 노드 c 로 전달하는 확률이 1인 경우의 로그 우도 비율(Log likelihood ratio)를 의미한다. 그리고 σ^2 은 잡음 분산을 의미하며, y_v 는 가변노드로 입력되는 신호를 나타낸다. 즉, 가변 노드 v 가 체크 노드 c 로 전달하는 확률이 0인 경우에 대한 가변 노드 v 가 체크 노드 c 로 전달하는 확률이 1인 경우의 로그 우도 비율을 산출하는 것이다.

[0029] 체크 노드 유닛(220)은 체크 노드($c=0, \dots, m-1$)에 대하여 수학식 2를 이용하여 모든 $L(q_{cv})$ 를 수집하고, 각 체크 노드에 연결된 모든 다른 가변 노드로부터의 정보를 바탕으로 가변 노드의 신뢰를 갱신한다.

수학식 2

$$L(r_{cv}) = \alpha S_{n \setminus N(c) \setminus v}^0 \text{sign}(L(q_{cv})) S_{n \setminus N(c) \setminus v}^{\min} |L(q_{cv})|$$

[0030]

[0031] 가변 노드 유닛(230)은 $v=0, \dots, n-1$ 에 대하여 수학식3을 이용하여 가변노드의 연산을 수행한다.

수학식 3

$$L(q_{cv}) = \sum_{m \in M(v) \setminus c} L(r_{mv}) + 2y_v / \sigma^2$$

[0032]

[0033] 판정 유닛(260)은 가변 노드 유닛(230)이 연산한 수학식4의 결과값의 부호를 이용하여, 결과값의 부호가 음수이면 대응하는 가변 노드를 1로 판정하고, 양수이면 0으로 판정한다.

수학식 4

$$L(Q_v) = \sum_{m \in M(v)} L(r_{mv}) + 2y_v / \sigma^2$$

[0034]

[0035] 그리고, 판정 유닛(260)은 판정한 코드 워드($\hat{c}$)와 패리티 비트 행렬(H)의 행렬 곱 연산을 수행하여 영행렬이면 복호화를 종료하고, 그렇지 않으면 체크 노드 유닛(220)에서의 연산으로 복귀하여 전체 과정을 반복한다.

[0036] 그런데, LDPC 장치가 복수열의 가변 노드 유닛(230)에서의 동시 연산을 수행하고자 하는 경우, 메모리(250) 내 동일 बैं크에 대하여 서로 다른 주소의 데이터가 충돌하는 상황이 발생할 수 있다.

[0037] 도 3(a)은 동일 बैं크 내 서로 다른 주소의 데이터가 충돌하는 상황에 대한 설명도이다.

[0038] 가변 노드 유닛(230)이 메모리(250) 내 체크 노드 결과값을 독출하여 가상의 블록 열C1과 블록 열C6에 대한 가변 노드 연산을 동시에 수행하는 경우, 동일 बैं크R4 내 address25의 데이터와 address0의 데이터를 동시에 독출해야 하나, 동시에 동일 बैं크 내 서로 다른 번지에서 데이터를 독출하는 것은 불가능하다.

[0039] 이때에는, 도 3(b)와 같이, 데이터가 포함된 어드레스를 쉬프트 시켜 처리한다.

[0040] 도 3(b)은 본 발명의 일실시예에 따른 단일 बैं크 내 어드레스 충돌을 회피하기 위한 어드레스 쉬프트 기법을 설명하기 위한 도면이다.

[0041] 도 3(b)에 따르면, 블록 열C1의 address25에 대응하는 블록 열C6의 address0를 address25로 쉬프트 시켜 일치시키고, 블록 열C6의 나머지 address29, 및 address4에 대해서도 동일한 간격으로 쉬프트 시킨다. 즉, 블록 열C6의 address29, 및 address4를 각각 25 만큼 쉬프트 시킨 address54, 및 address29로 변경한다.

[0042] 이를 위하여 본 발명의 일실시예에 따른 제어 유닛(240)은, 가변 노드 유닛(230)과 메모리(250)를 연결하는 네트워크(241)과, 가변 노드 유닛(230)이 메모리(250) 내 체크 노드 결과값을 읽어들이 가상의 복수의 블록 열에 대하여 가변 노드 연산을 동시에 수행할 때, 단일 बैं크 내에서 서로 다른 어드레스의 데이터를 동시에 읽으면, 서로 다른 어드레스 중 어느 하나를 소정 간격 쉬프트 시켜 서로 다른 어드레스를 상호 일치시키는 서브 컨트롤러(242)를 포함할 수 있다.

[0043] 결과적으로, 본 발명에 따른 어드레스 쉬프트 기법을 활용하면, 가변 노드 유닛(230)이 다수의 블록 열에 대하여 동시에 연산을 수행할 수 있다.

[0044] 도 4(a)는 복수개의 बैं크 내에서 복수의 어드레스들이 충돌하는 상황에 대한 설명도이다.

- [0045] 가변 노드 유닛(230)이 메모리(250) 내 체크 노드 결과값을 독출하여 가상의 블록 열C2와 블록 열C5에 대한 가변 노드 연산을 동시에 수행하는 경우, 동일 बैं크 R1 내 address1의 데이터와 address0의 데이터, 동일 बैं크 R2 내 address13의 데이터와 address16의 데이터, 그리고, 동일 बैं크 R5 내 address25의 데이터와 address20의 데이터를 동시에 독출해야 하나, 동시에 동일 बैं크 내 서로 다른 번지에서 데이터를 독출하는 것이 불가능하다.
- [0046] 이때에는, 도 4(B)와 같이, 데이터가 포함된 어드레스를 쉬프트 시키고, 추가적으로 소량의 레지스터를 추가하여 처리한다.
- [0047] 도 4(b)는 본 발명의 다른 실시예에 따른 복수 बैं크 내 어드레스 충돌을 회피하기 위한 어드레스 쉬프트 기법을 설명하기 위한 도면이고, 도 5(a)(b)(c)는 본 발명의 일 실시예에 따른 레지스터 내 데이터 갱신 패턴 설명도이다.
- [0048] 우선, बैं크 R1을 기준으로 하여, 어드레스를 일치시키고, 나머지 어드레스들에 대하여 해당하는 크기 만큼 쉬프트 시킨다.
- [0049] 이때, 어드레스를 쉬프트 시켰으나, 불일치하면서 충돌하는 बैं크R2 및 R5 내 어드레스들의 데이터들은 상호 불일치하는 어드레스의 차에 해당하는 만큼의 레지스터를 마련하여 데이터를 저장한다.
- [0050] 마련된 레지스터를 이용하여 데이터를 공급하는 과정을 구체적으로 설명하면 다음과 같다.
- [0051] (R2, C2')의 어드레스가 12이고, (R2, C5)의 어드레스가 16이므로, 두 위치간의 어드레스 차이는 4이다. 따라서, 도 5에 도시된 바와 같이, 레지스터 내 데이터를 저장하기 위한 필요 공간은 4개가 필요하다. 도 5(a)는 초기에 레지스터 내에 저장되는 데이터 어드레스이고, 도 5(b)는 1 싸이클 후 레지스터 내에 갱신되어 저장되는 데이터 어드레스이며, 도 5(c)는 2 싸이클 후 레지스터 내에 갱신되어 저장되는 데이터 어드레스이다.
- [0052] 도 6은 도 5(a)(b)(c)를 구현하기 위한 주요 구성에 대한 블록도이다.
- [0053] 본 발명의 다른 실시예에 따른 제어 유닛(640)은 가변 노드 유닛(630)과 메모리(650)를 연결하는 네트워크(641), 복수 बैं크 내에서 불일치하는 어드레스 차이를 보상하기 위한 레지스터 어레이(642), 및 레지스터 어레이의 갱신되는 데이터를 가변 노드 유닛에 공급하는 서브 컨트롤러(643)를 포함한다.
- [0054] 서브 컨트롤러(643)는 메모리(650) 내 해당 어드레스의 데이터를 읽어들이 레지스터 어레이 내 최후 데이터를 갱신함과 아울러, 레지스터 어레이 내 최선 데이터를 가변 노드 유닛(630)으로 출력한다.
- [0055] 본 발명에 따르면, LDPC 부호용 장치는 저장된 값들을 여러 번에 걸쳐 다른 순서로 독출할 필요가 있는 병렬 프로세서가 포함된 구조에서 적용 가능하다. 예컨대, 복호화기 등에 적용 가능하다.
- [0056] 이와 같이, 본 발명의 상세한 설명에서는 구체적인 실시예(들)에 관해 설명하였으나, 본 발명의 범주에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시예(들)에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위 뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

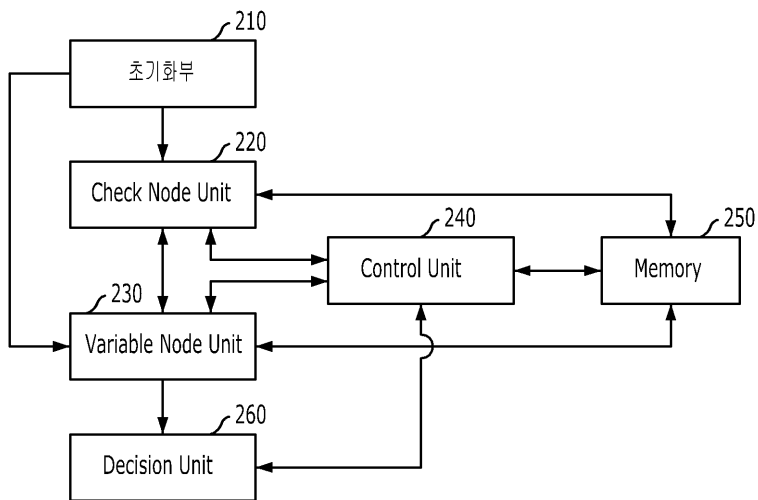
- | | | |
|--------|---------------|---------------|
| [0057] | 210: 초기화부 | 220: 체크 노드 유닛 |
| | 230: 가변 노드 유닛 | 240: 제어 유닛 |
| | 250: 메모리 | 260: 판정 유닛 |

도면

도면1

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
R0		0			2			7	1	4		
R1			1		36	0	29			15	30	
R2	35		13	2		16			30		45	32
R3	3		21	24			4	14				2
R4		25		37	28		0	3				
R5	20	6	25			20			1	15	42	

도면2



도면3a

	C1	C6
R0	0	
R1		29
R2		
R3		4
R4	25	0
R5	6	

도면3b

	C1	C6'
R0	0	
R1		54
R2		
R3		29
R4	25	25
R5	6	

도면4a

	C2	C5
R0		
R1	1	0
R2	13	16
R3	21	
R4		
R5	25	20

도면4b

	C2'	C5
R0		
R1	0	0
R2	12	16
R3	20	
R4		
R5	24	20

도면5a

Position:(R2, C2')

12	13	14	15
----	----	----	----

[C2' supply]

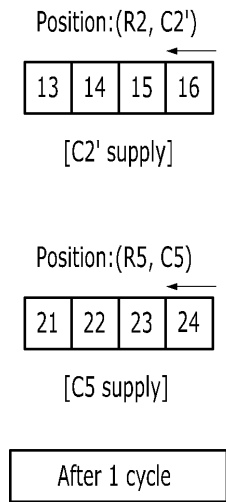
Position:(R5, C5)

20	21	22	23
----	----	----	----

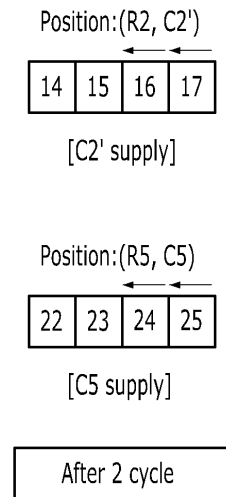
[C5 supply]

Initialization

도면5b



도면5c



도면6

