



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년11월06일
(11) 등록번호 10-1198433
(24) 등록일자 2012년10월31일

(51) 국제특허분류(Int. Cl.)

H03M 13/41 (2006.01)

(21) 출원번호 10-2011-0096548

(22) 출원일자 2011년09월23일

심사청구일자 2011년09월23일

(56) 선행기술조사문헌

KR100680024 B1

JP10145243 A

JP11312984 A

KR1020090071455 A

(73) 특허권자

인하대학교 산학협력단

인천광역시 남구 인하로 100, 인하대학교 (용현동)

(72) 발명자

강진구

서울특별시 서초구 잠원동 잠원한신아파트 2동 505호

서준협

전라남도 영광군 군남면 남양길1길 46

(74) 대리인

특허법인무한

전체 청구항 수 : 총 10 항

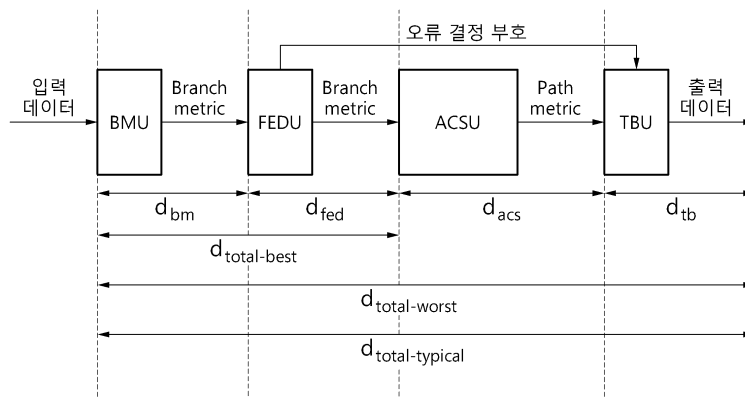
심사관 : 장석환

(54) 발명의 명칭 오류 결정 블록을 이용한 비터비 복호화 방법 및 장치

(57) 요약

오류 결정 블록을 이용한 비터비 복호화 방법 및 장치가 개시된다. 비터비 복호화 방법은, 입력 데이터와 길쌈 부호의 상태도(state diagram)을 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(Branch Metric)을 계산하고, 구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정할 수 있다. 그리고, 오류 발생 여부에 기초하여 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행하거나 바이패스할 수 있다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 MEST 2011-0018394

부처명 지식경제부

연구사업명 중점연구소

연구과제명 그린 생체모방형 임플란터블 전자칩 및 U-생체정보처리 플랫폼 융합기술 개발

주관기관 인하대학교 산학협력단

연구기간 2011.05.01 ~ 2012.04.30

특허청구의 범위

청구항 1

구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정하는 단계; 및

상기 오류 발생 여부에 기초하여 상기 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행할지 여부를 결정하는 단계

를 포함하는 비터비 복호화 방법.

청구항 2

제1항에 있어서,

상기 비교 선택 연산 및 역추적 연산을 수행할지 여부를 결정하는 단계는,

상기 비교 선택 연산 및 상기 역추적 연산을 바이패스(bypass)하는 것으로 결정함에 따라, 상기 입력 데이터에 기초하여 출력 데이터를 결정하는 것을 특징으로 하는 비터비 복호화 방법.

청구항 3

제1항에 있어서,

상기 오류 발생 여부를 결정하는 단계는,

상기 입력 데이터의 길이와 상기 구속장의 길이에 기초하여 상기 데이터 입력 상태를 락(Lock) 상태 및 언락(UnLock) 상태로 식별하는 것을 특징으로 하는 비터비 복호화 방법.

청구항 4

제1항에 있어서,

입력 데이터와 상태도(state diagram)을 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(Branch Metric)을 계산하는 단계

를 더 포함하고,

상기 오류 발생 여부를 결정하는 단계는,

상기 데이터 입력 상태가 락 상태로 식별됨에 따라, 상기 브랜치 메트릭을 이용하여 상기 입력 데이터의 오류 발생 여부를 나타내는 오류 결정 부호(Error Decision Code)를 생성하는 것을 특징으로 하는 비터비 복호화 방법.

청구항 5

제1항에 있어서,

상기 오류 발생 여부를 결정하는 단계는,

상기 데이터 입력 상태가 락 상태로 식별된 상태에서, 상기 입력 데이터에 오류가 발생함에 따라 상기 데이터 입력 상태를 언락 상태로 변경하는 것을 특징으로 하는 비터비 복호화 방법.

청구항 6

제1항에 있어서,

상기 비교 선택 연산 및 역추적 연산을 수행할지 여부를 결정하는 단계는,

상기 데이터 입력 상태가 언락 상태로 식별됨에 따라, 상기 입력 데이터에 대해 비교 선택 연산을 수행하는 단계; 및

상기 입력 데이터에 대해 역추적(Trace Back: TB) 연산을 수행하는 단계를 포함하는 비터비 복호화 방법.

청구항 7

구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정하는 오류 결정부; 및

상기 오류 발생 여부에 기초하여 상기 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행할지 여부를 결정하는 연산 결정부

를 포함하는 비터비 복호화 장치.

청구항 8

제7항에 있어서,

상기 연산 결정부는,

상기 비교 선택 연산 및 상기 역추적 연산을 바이패스(bypass)하는 것으로 결정함에 따라, 상기 입력 데이터에 기초하여 출력 데이터를 결정하는 것을 특징으로 하는 비터비 복호화 장치.

청구항 9

제7항에 있어서,

입력 데이터와 상태도(State Diagram)를 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(Branch Metric)을 브랜치 메트릭 계산부

를 더 포함하고,

상기 오류 결정부는,

상기 데이터 입력 상태가 락 상태로 식별됨에 따라, 상기 브랜치 메트릭을 이용하여 상기 입력 데이터의 오류 발생 여부를 나타내는 오류 결정 부호(Error Decision Code)를 생성하는 것을 특징으로 하는 비터비 복호화 장치.

청구항 10

제7항에 있어서,

상기 오류 결정부는,

상기 데이터 입력 상태가 락 상태로 식별된 상태에서, 상기 입력 데이터에 오류가 발생함에 따라 상기 데이터 입력 상태를 언락 상태로 변경하는 것을 특징으로 하는 비터비 복호화 장치.

명세서

기술 분야

[0001] 본 발명은 비터비 복호화 방법 및 장치에 관한 것으로, 길쌈 부호(Convolution Code)를 사용하여 수신한 데이터를 복호화하는 기술에 관한 것이다.

배경 기술

[0002] 디지털 통신 시스템에서 데이터 정보를 전송할 때 다중 경로, 간섭, 잡음 등으로 인한 오류가 발생한다. 이때, 오류 발생이 적을수록 디지털 통신 시스템의 신뢰도가 높아진다.

[0003] 일반적으로, 비터비 복호화(Viterbi Decoding)는 3단계의 연산 과정으로 이루어진다. 1단계는 브랜치 메트릭(Branch Metric: BM)을 계산하는 과정이고, 2단계는 계산된 브랜치 메트릭들을 비교하여 어느 하나의 브랜치를 선택(Add Compare Select: ACS)하는 과정이고, 3단계는 역추적(Trace Back: TB) 계산하는 과정이다.

- [0004] 한국공개특허 제10-2007-0048947호(공개일 2007년 05월 10일)에는 입력된 데이터에 대해 가산 및 비교 동작을 수행하고, 선택 동작을 수행하여 데이터를 비터비 디코딩하는 기술이 개시되어 있다. 그러나, 종래의 비터비 복호화(Viterbi Decoding) 장치는, 3단계 과정을 통해 랜덤 에러(Random error)에 우수한 성능을 가지나, 시스템의 복잡도가 높아져 전송 속도가 느리며, 랜덤 에러를 찾는 데 소모되는 시스템의 자원이 증가한다.
- [0005] 특히, 역추적 계산 시, 입력 데이터에 대한 복호화 연산이 모두 종료될 때까지 오류 가능성이 가장 적은 경로(Path)를 항상 메모리에 저장하고 있어야 하므로, 메모리가 구속장의 길이(Constraint Length: K)에 비례하여 점유된다. 이로 인해, 오류 정정 시간이 길어져서 비터비 복호화 장치의 성능이 저하된다.
- [0006] 따라서, 비터비 복호화를 고속으로 수행할 수 있는 방안이 요구된다.

발명의 내용

해결하려는 과제

- [0007] 본 발명은 오류 정정 블록 및 복잡한 3단계 연산 과정을 제어하여 비터비 복호화를 고속으로 수행할 수 있는 방법 및 장치를 제공하고자 한다.

과제의 해결 수단

- [0008] 본 발명의 일실시예에 따른 비터비 복호화 방법은, 구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정하는 단계, 및 상기 오류 발생 여부에 기초하여 상기 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행할지 여부를 결정하는 단계를 포함할 수 있다.
- [0009] 또한, 상기 비교 선택 연산 및 역추적 연산을 수행할지 여부를 결정하는 단계는, 상기 비교 선택 연산 및 상기 역추적 연산을 바이패스(bypass)하는 것으로 결정함에 따라, 상기 상태도에 기초하여 입력 데이터에 대응하는 식별 데이터를 출력 데이터로 결정할 수 있다.
- [0010] 또한, 상기 오류 발생 여부를 결정하는 단계는, 상기 입력 데이터의 길이와 상기 구속장의 길이에 기초하여 상기 데이터 입력 상태를 락(Lock) 상태 및 언락(UnLock) 상태로 식별할 수 있다.
- [0011] 또한, 입력 데이터와 상태도(state diagram)를 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(Branch Metric)을 계산하는 단계를 더 포함할 수 있다. 그러면, 상기 오류 발생 여부를 결정하는 단계는, 상기 데이터 입력 상태가 락 상태로 식별됨에 따라, 상기 브랜치 메트릭을 이용하여 상기 입력 데이터의 오류 발생 여부를 나타내는 오류 결정 부호(Error Decision Code)를 생성할 수 있다.
- [0012] 또한, 상기 오류 발생 여부를 결정하는 단계는, 상기 데이터 입력 상태가 락 상태로 식별된 상태에서, 상기 입력 데이터에 오류가 발생함에 따라 상기 데이터 입력 상태를 언락 상태로 변경할 수 있다.
- [0013] 또한, 상기 비교 선택 연산 및 역추적 연산을 수행할지 여부를 결정하는 단계는, 상기 데이터 입력 상태가 언락 상태로 식별됨에 따라, 상기 입력 데이터에 대해 비교 선택 연산을 수행하는 단계, 및 상기 입력 데이터에 대해 역추적(Trace Back: TB) 연산을 수행하는 단계를 포함할 수 있다.
- [0014] 본 발명의 일실시예에 따른 비터비 복호화 장치는, 입력 데이터와 상태도(State Diagram)를 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(Branch Metric)을 브랜치 메트릭 계산부, 구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 상기 입력 데이터의 오류 발생 여부를 결정하는 오류 결정부, 및 상기 오류 발생 여부에 기초하여 상기 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행할지 여부를 결정하는 연산 결정부를 포함할 수 있다.

발명의 효과

- [0015] 본 발명의 일실시예에 따르면, 각 스테이트(state)에서 입력된 데이터의 오류 발생 여부에 기초하여 오류 정정 블록 및 복잡한 3단계 연산 과정을 제어함에 따라, 비터비 복호화를 고속으로 수행할 수 있다.

도면의 간단한 설명

- [0016] 도 1은 본 발명의 일실시예에 따른 비터비 복호화 장치의 개략적인 구성을 도시한 블록도이다.

도 2는 본 발명의 일실시예에 따른 오류 결정을 이용하여 고속으로 비터비 복호화를 수행하는 방법을 설명하기 위해 제공되는 흐름도이다.

도 3은 본 발명의 일실시예에 따른 비터비 복호화 장치의 구성을 도시한 블록도이다.

도 4는 본 발명의 일실시예에 따른 입력 데이터의 오류 발생 여부에 따른 비터비 복호화 과정을 도시한 트렐리스 다이어그램(Trellis Diagram)이다.

도 5는 본 발명의 일실시예에 따른 비터비 복호화 장치의 오류 결정부의 세부 구성을 도시한 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0017] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0018] 도 1은 본 발명의 일실시예에 따른 비터비 복호화 장치의 개략적인 구성을 도시한 블록도이다.

[0019] 도 1에 따르면, 비터비 복호화는, 브랜치 메트릭을 계산하는 과정(BM), 고속으로 오류를 결정하는 과정(Fast Error Decision: FED), 비교 선택하는 과정(ACS), 및 역추적하는 과정(TB)으로 이루어질 수 있다.

[0020] 이때, 비터비 복호화 장치는, 입력 데이터의 오류 발생 여부를 결정하는 (FED)는 브랜치 메트릭을 계산하는 과정(BM) 바로 다음에 연속하여 수행할 수 있다. 다시 말해, 기존의 비터비 복호화에서 입력 데이터의 오류 발생을 결정하는 과정(FED)이 추가될 수 있다.

[0021] 그리고, 비터비 복호화 장치는, 비터비 복호화 장치로 입력된 데이터가 비터비 복호화를 통해 출력될 때까지 고정된 지연 시간을 아래의 수학식 1 및 수학식 2를 이용하여 계산할 수 있다.

수학식 1

$$d_{total_worst} = d_{bm} + d_{fed} + d_{acs} + d_{tb}$$

[0023] 수학식 1에서, d_{total_worst} 은 비터비 복호화 장치에서 발생한 가장 긴 지연 시간을 의미하고, d_{bm} 은 브랜치 메트릭을 계산하는 과정에서 발생하는 지연 시간, d_{fed} 는 입력 데이터의 오류 발생 여부를 결정하는 과정에서 발생하는 지연 시간, d_{acs} 는 비교 및 선택 과정에서 발생하는 지연 시간, d_{tb} 는 역추적 과정에서 발생하는 지연 시간이다.

[0024] 이때, d_{fed} 는 d_{bm} , d_{acs} , d_{tb} 보다 매우 작기 때문에 수학식 1에서 d_{fed} 는 거의 무시될 수 있다. 다시 말해, 수학식 1의 가장 긴 지연 시간은 $d_{total_worst} = d_{bm} + d_{acs} + d_{tb}$ 표현될 수도 있다. 이처럼, 본 발명의 일실시예에 따른 비터비 복호화 장치는 입력 데이터의 오류를 결정하는 과정을 추가하더라도, 입력 데이터를 복호화하면서 발생한 가장 긴 지연 시간 d_{total_worst} 은 기존의 비터비 복호화 장치에서 입력 데이터를 복호화하면서 발생한 지연 시간 $d_{total_typical}$ 과 동일하거나 유사할 수 있다.

[0025] 그리고, 비터비 복호화 장치에서 발생한 가장 짧은 지연 시간 d_{total_best} 은 아래의 수학식 2와 같이 표현될 수 있다.

수학식 2

$$d_{total_best} = d_{bm} + d_{fed}$$

[0027] 수학식 2에 따르면, 비터비 복호화 장치는, 브랜치 메트릭을 연산하는 과정 및 입력 데이터의 오류 발생을 결정하는 구성을 통해 입력 데이터를 복호화할 수도 있다. 이에 따라, 가장 짧은 지연 시간은 브랜치 메트릭을 계산하는 과정에서 발생하는 지연 시간 d_{bm} 과 입력 데이터의 오류 발생 여부를 결정하는 과정에서 발생하는 지연 시간 d_{fed} 의 합으로 계산될 수 있다.

[0028] 이때, 100개의 입력 데이터 중에서 1개의 입력 데이터에만 오류가 발생한 경우, 비터비 복호화 장치는, 아래의 수학적식 3을 이용하여 지연 시간 단축으로 인해 얻을 수 있는 이득 $d_{advance}$ 을 계산할 수 있다.

수학적식 3

$$\begin{aligned} d_{advance} &= 100d_{total_typical} - (d_{total_worst} + 99d_{total_best}) \\ &= 99(d_{acs} + d_{tb}) - 99d_{fed} \\ &\approx 80(d_{acs} + d_{tb}) \end{aligned}$$

[0029]

[0030] 수학적식 3에서, $d_{total_typical}$ 은 기존의 비터비 복호화 장치에서 발생하는 지연 시간으로서, d_{bm} , d_{acs} , d_{tb} 의 합으로 계산될 수 있다. 그러면, 비터비 복호화 장치는, 수학적식 1 및 수학적식 2 그리고 $d_{total_typical}$ 을 이용하여 지연 시간 단축에 따른 이득($d_{advance}$)을 계산할 수 있다. 다시 말해, 수학적식 3에 따르면, 본 발명의 일실시예에 따른 비터비 복호화 장치는 기존의 비터비 복호화 장치보다 ACS 연산 및 TB 연산을 수행하는 소요되는 시간 및 자원(resource)이 절약될 수 있다.

[0031] 이하에서는, 도 2를 참고하여 BM연산, FED 연산, ACS 연산 및 TB 연산을 제어하여 비터비 복호화를 고속으로 수행하는 과정에 대해 설명하기로 한다.

[0032] 도 2는 본 발명의 일실시예에 따른 오류 결정을 이용하여 고속으로 비터비 복호화를 수행하는 방법을 설명하기 위해 제공되는 흐름도이다.

[0033] 먼저, 201 단계에서, 비터비 복호화 장치는, 입력 데이터와 상태도(State Diagram)을 이용하여 현재 시간에 해당하는 각 스테이트(state)의 브랜치 메트릭(branch metric)을 계산할 수 있다. 여기서, 입력 데이터는, 부호화 장치에서 길쌈 부호화 등을 이용하여 부호화된 데이터를 비터비 복호화 장치가 수신한 데이터이다. 그리고, 상태도는 입력 데이터에 따른 각 상태(state)의 변화를 나타내는 다이어그램이다. 예를 들어, 상태도로는, 길쌈 부호기(Convolution Coder)의 상태도가 이용될 수 있다.

[0034] 일례로, 구속장의 길이 $K=3$ 이고, 부호화율(code rate: R)이 $1/2m$ 으로 미리 설정된 경우, 비터비 복호화 장치는, 아래의 수학적식 4를 이용하여 브랜치 메트릭을 계산할 수 있다. 여기서, 구속장은, 출력에 관여하는 총 입력 데이터의 개수를 의미할 수 있다.

수학적식 4

$$bm_k = (r - t_k)^2$$

[0035]

[0036] 수학적식 4에서, bm_k 는 현재 시간 k 에서 각 스테이트의 브랜치 메트릭이고, r 은 입력 데이터, t_k 는 상태도 및 입력 데이터에 기초하여 출력해야 될 데이터이다.

[0037] 수학적식 4에 따르면, 비터비 복호화 장치는, 입력 데이터 k 와 출력해야 될 데이터 t_k 간의 차이를 이용하여 브랜치 메트릭을 계산할 수 있다.

[0038] 이어, 비터비 복호화 장치는 구속장의 길이에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정할 수 있다. 여기서, 데이터 입력 상태는, 락 상태(Lock State) 및 언락 상태(UnLock State)를 포함할 수 있다. 여기서, 언락 상태는, 입력 데이터가 구속장의 길이 이전까지 입력될 때까지의 데이터 입력 상태를 의미하고, 락 상태는, 입력 데이터의 길이가 구속장의 길이가 되는 데이터 입력 상태를 의미한다.

[0039] 다시 말해, 202 단계에서, 입력 데이터의 길이가 구속장의 길이보다 작으면, 209 단계에서, 비터비 복호화 장치는, 데이터 입력 상태를 언락 상태로 결정할 수 있다. 그리고, 202 단계에서, 입력 데이터의 길이가 구속장의 길이보다 작지 않으면, 203 단계에서, 비터비 복호화 장치는, 데이터 입력 상태를 락 상태로 결정할 수 있다.

- [0040] 일례로, 길쌈 부호화된 데이터에 대해 비터비 디코딩을 수행하는 경우, 대기 시간(latency)이 발생하여 길쌈 부호화된 데이터가 비터비 복호화 장치로 입력되는 즉시 복호화된 데이터가 출력되지 않을 수 있다. 이에 따라, 올바른 경로(path)를 결정할 수 있을 만큼의 입력 데이터가 필요하며, 기준 개수로서 기설정될 수 있다. 그러면, 도 4를 참고하면, 비터비 복호화 장치는, 입력 데이터가 기준 개수가 될 때까지 입력 데이터를 수집할 수 있으며, 입력 데이터를 수집하는 동안의 데이터 입력 상태를 언락 상태로 결정할 수 있다. 이때, 데이터를 수집하는 구간은 구속장의 길이인 K 에 비례할 수 있다. 이어, 올바른 경로(path)가 결정되면, 비터비 복호화 장치는, 데이터 입력 상태를 락 상태로 결정할 수 있다.
- [0041] 예를 들어, 구속장의 길이 $K=n$ 으로 미리 설정된 경우, 비터비 복호화 장치는, 입력 데이터가 1부터 $n-1$ 까지 n 개의 입력 데이터가 입력될 때의 데이터 입력 상태를 언락 상태로 결정할 수 있다. 이때, $K=n$ 인 경우, 대기 시간(latency) 또한 구속장의 길이와 동일할 수 있다(latency= n). 그리고, n 번째 입력 데이터가 입력되어 입력 데이터의 길이가 구속장의 길이가 되면, 비터비 복호화 장치는, 데이터 입력 상태를 락 상태로 결정할 수 있다. 이처럼, 데이터 입력 상태가 락 상태가 되면, 비터비 복호화 장치는, 고속으로 오류를 결정하는 과정(FED)을 통해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행하지 않고도 빠르게 입력 데이터를 디코딩하여 출력할 수 있다.
- [0042] 이어, 204 단계에서, 비터비 복호화 장치는, 오류 발생 여부에 기초하여 입력 데이터에 대해 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행할지 여부를 결정할 수 있다.
- [0043] 이때, 비터비 복호화 장치는, 브랜치 메트릭을 이용하여 입력 데이터에 오류가 발생했는지 여부를 결정할 수 있다.
- [0044] 그리고, 비터비 복호화 장치는, 오류 발생 여부에 기초하여 오류 결정 부호(Error Decision Code)를 생성할 수 있다.
- [0045] 일례로, 도 5를 참고하면, 데이터 입력 상태가 락 상태인 경우, 비터비 복호화 장치는, 입력되는 스테이트의 브랜치 메트릭 중 어느 하나라도 "1" 이상이면 오류가 발생한 것으로 결정할 수 있다. 그리고, 입력 데이터에 오류가 발생한 경우, 비터비 복호화 장치는 "1"을 포함하는 오류 결정 부호를 생성할 수 있다. 이처럼, 오류 결정 부호가 "1"을 포함하는 경우, 비터비 복호화 장치는, 비교 선택(ACS) 연산 및 역추적(TB) 연산을 수행하여 비터비 디코딩을 수행할 수 있다.
- [0046] 이때, 입력되는 스테이트의 브랜치 메트릭이 모두 "0"인 경우, 비터비 복호화 장치는, 오류가 발생하지 않은 것으로 결정할 수 있다. 그러면, 입력 데이터에 오류가 발생하지 않은 경우, 비터비 복호화 장치는, "0"을 포함하는 오류 결정 부호를 생성할 수 있다. 이처럼, 입력되는 모든 브랜치 메트릭이 "0"인 경우, 비터비 복호화 장치는 ACS 연산 및 TB 연산을 수행하지 않고도 입력 데이터에 대해 비터비 디코딩을 수행함으로써, 고속으로 디코딩을 수행하면서 올바른 디코딩 결과를 출력할 수 있다. 여기서, 예측 입력값은 트렐리스 스테이트 테이블(trellis state table)에 기초하여 결정될 수 있다.
- [0047] 이때, 205 단계에서, 오류가 발생하지 않은 것으로 결정된 경우, 비터비 복호화 장치는, 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행하지 않고, 바이패스(bypass)하는 것으로 결정할 수 있다.
- [0048] 일례로, 비터비 복호화 장치는, 오류 결정 부호를 확인하고, 확인한 오류 결정 부호가 "1"이면 ACS 연산 및 TB 연산을 바이패스하는 것으로 결정할 수 있다. 그리고, 비터비 복호화 장치는, ACS 연산 및 TB 연산을 바이패스함에 따라, 입력 데이터에 해당하는 식별 데이터를 출력 데이터로 결정할 수 있다.
- [0049] 다시 말해, 입력 데이터에 오류가 존재하지 않으므로, 비터비 복호화 장치는, 입력 데이터를 ACS 연산 및 TB 연산의 수행 없이 바로 출력하여 시스템의 자원, 복호화 시간, 그리고 전송 시간을 절약할 수 있다. 예를 들어, 입력 데이터가 "11 01 01"이고, 오류 결정 부호가 0인 경우, 비터비 복호화 장치는, 상태도에 기초하여 스테이트 00에서 입력 데이터 11에 해당하는 식별 데이터 "1", 스테이트 10에서 입력 데이터 01에 해당하는 식별 데이터 "1", 스테이트 11에서 입력 데이터 01에 해당하는 식별 데이터 "0"을 출력 데이터로 결정할 수 있다. 즉, 출력 데이터는 "1 1 0"가 될 수 있다.
- [0050] 그리고, 204 단계에서, 데이터 입력 상태가 락 상태로 결정된 상태에서, 오류가 발생한 것으로 결정된 경우, 206 단계에서, 비터비 복호화 장치는, 데이터 입력 상태를 락 상태에서 언락 상태로 변경할 수 있다.

- [0051] 이때, 입력 데이터에 오류가 발생하여 데이터 입력 상태가 언락 상태인 경우, 비터비 복호화 장치는, 비교 선택(Add Compare Select: ACS) 연산 및 역추적(Trace Back: TB) 연산을 수행하는 것으로 결정할 수 있다.
- [0052] 그러면, 207 단계에서, 비터비 복호화 장치는, 입력 데이터에 대해 ACS 연산을 수행할 수 있다. 이때, 비터비 복호화 장치는, 각 스테이트(state)의 해밍 디스턴스(hamming distance)를 이용하여 ACS 연산을 수행할 수 있다. 여기서, 해밍 디스턴스는, 같은 비트 수를 갖는 2진 부호 사이에 대응되는 비트값이 일치하지 않는 개수를 의미할 수 있다.
- [0053] 이때, 비터비 복호화 장치는, 현재 시간에 해당하는 각 스테이트에 존재하는 하나 이상의 해밍 디스턴스 중 최소를 선택할 수 있다. 그리고, 비터비 복호화 장치는, 선택한 최소의 해밍 디스턴스에 대응하는 브랜치 메트릭과 이전 시간 상태의 브랜치 메트릭을 누적하여 경로 메트릭(Path Metric, 이하, 경로값이라 칭함)을 각 스테이트 별로 계산할 수 있다.
- [0054] 이때, 비터비 복호화 장치는, ACS연산의 속도를 향상시키기 위해 현재 시간에 해당하는 각 스테이트의 ACS 연산을 병렬화하여 수행할 수 있다.
- [0055] 이어, 208 단계에서, 비터비 복호화 장치는, ACS 연산이 수행됨에 따라 입력 데이터에 대해 TB 연산을 수행할 수 있다. 이때, 비터비 복호화 장치는, ACS 연산을 통해 계산된 각 스테이트의 경로 메트릭을 이용하여 어느 하나의 경로 메트릭을 선택할 수 있다. 예를 들어, 비터비 복호화 장치는, 각 스테이트의 경로 메트릭 중 최소인 경로 메트릭을 선택할 수 있다. 그리고, 비터비 복호화 장치는, 선택된 최소의 경로 메트릭에 해당하는 경로(path)를 기준으로 시간 상태가 0인 처음까지 역추적하여 입력 데이터를 복원할 수 있다. 이어, 비터비 복호화 장치는, 복원된 데이터를 출력 데이터로 출력할 수 있다.
- [0056] 예를 들어, 현재 시간 $t=n$ 인 경우, n 번째 시간 상태에서, 스테이트 1의 경로 메트릭 1이 최소인 경로 메트릭으로 선택된 경우, 비터비 복호화 장치는 경로 메트릭 1에 해당하는 경로 1을 기준으로 $t=n-1$ 부터 $t=0$ 까지 역추적하여 입력 데이터를 복원할 수 있다.
- [0057] 한편, 209 단계에서, 입력 데이터의 길이 및 구속장의 길이에 기초하여 데이터 입력 상태를 언락 상태로 결정한 경우, 비터비 복호화 장치는, 입력 데이터에 대해 ACS 연산 및 TB 연산을 수행할 수 있다.
- [0058] 다시 말해, 언락 상태인 경우, 비터비 복호화 장치는, 데이터 입력 상태를 언락 상태로 변경한 경우와 동일한 방법으로 입력 데이터에 대해 ACS 연산 및 TB 연산을 수행하여 입력 데이터를 복원할 수 있다.
- [0059] 도 3은 본 발명의 일실시예에 따른 비터비 복호화 장치의 구성을 도시한 블록도이다.
- [0060] 도 3에 따르면, 비터비 복호화 장치(300)는, 브랜치 메트릭 계산부(301), 오류 결정부(302), 연산 결정부(303), 비교 선택부(304), 역추적 계산부(305)를 포함할 수 있다.
- [0061] 먼저, 브랜치 메트릭 계산부(301)는 입력 데이터와 상태도(State Diagram)를 이용하여 현재 시간에 해당하는 각 스테이트의 브랜치 메트릭(branch metric)을 계산할 수 있다. 예를 들어, 브랜치 메트릭 계산부(301)는 입력 데이터, 그리고 상태도에 기초하여 현재 시간에 해당하는 각 스테이트에서 출력해야 할 데이터를 이용하여 각 스테이트의 브랜치 메트릭을 계산할 수 있다.
- [0062] 여기서, 상태도는 입력 데이터에 따른 각 상태(state)의 변화를 나타내는 다이어그램이다. 예를 들어, 상태도로는, 길쌈 부호기(Convolution Coder)의 상태도가 이용될 수 있다.
- [0063] 그리고, 오류 결정부(302)는 구속장의 길이(Constraint Length)에 따라 식별된 데이터 입력 상태에 기초하여 입력 데이터의 오류 발생 여부를 결정할 수 있다. 여기서, 데이터 입력 상태는, 언락(UnLock) 상태 및 락(Lock) 상태를 포함할 수 있다.
- [0064] 이때, 오류 결정부(302)는 입력 데이터의 길이와 구속장의 길이를 비교하여 데이터 입력 상태가 언락 상태인지, 락 상태인지 여부를 결정할 수 있다. 예를 들어, 구속장의 길이 $K=3$ 으로 미리 설정된 경우, 오류 결정부(302)는 입력 데이터가 2개일 때까지는 데이터 입력 상태를 언락 상태로 결정하고, 입력 데이터가 3이 되면 데이터 입력 상태를 락 상태로 결정할 수 있다.
- [0065] 그리고, 오류 결정부(302)는 브랜치 메트릭에 기초하여 입력 데이터에 오류가 발생했는지 여부를 결정하고, 결정에 기초하여 오류 결정 부호(Error Decision Code)를 생성할 수 있다. 예를 들어, 오류가 발생한 경우, 오류 결정부(302)는 "1"을 포함하는 오류 결정 부호를 생성할 수 있다. 그리고, 오류가 발생하지 않은 경우, 오류

결정부(302)는 "0"을 포함하는 오류 결정 부호를 생성할 수 있다.

[0066] 이때, 데이터 입력 상태가 락 상태로 식별된 상태에서, 입력 데이터에 오류가 발생한 것으로 결정된 경우, 오류 결정부(302)는 데이터 입력 상태를 락 상태에서 언락 상태로 변경할 수 있다.

[0067] 이어, 연산 결정부(303)는 오류 발생 여부에 기초하여 입력 데이터에 대해 비교 선택(ACS) 연산 및 역추적(TB) 연산을 수행할지 또는 바이패스(Bypass)할지 여부를 결정할 수 있다.

[0068] 이때, 입력 데이터에 오류가 발생한 경우, 연산 결정부(303)는 입력 데이터에 대해 ACS 연산 및 TB 연산을 수행하는 것으로 결정할 수 있다. 그리고, 입력 데이터에 오류가 발생하지 않은 것으로 결정한 경우, 연산 결정부(303)는 입력 데이터에 대해 ACS 연산 및 TB 연산을 수행하지 않는 것으로 결정할 수 있다. 다시 말해, 연산 결정부(303)는 입력 데이터에 대해 ACS 연산 및 TB 연산을 바이패스하고, 상태도에서 입력 데이터에 해당하는 식별 데이터를 바로 출력 데이터로 결정할 수 있다.

[0069] 예를 들어, 데이터 입력 상태는 락 상태이고, 입력 데이터가 "11 01 01"이고, 오류 결정 코드가 0인 경우, 연산 결정부(303)는 입력 데이터에 대해 ACS 연산 및 TB 연산을 생략하고, 상태도에서 입력 데이터 "11 01 01"에 해당하는 식별 데이터 "1 1 0"를 출력 데이터로 결정할 수 있다. 이때, 오류 결정 코드가 1인 경우, 연산 결정부(303)는 입력 데이터 "11 01 01"에 대해 ACS 연산 및 TB 연산을 수행하는 것으로 결정할 수 있다.

[0070] 그러면, 비교 선택부(304)는 입력 데이터에 대해 ACS 연산을 수행할 수 있다.

[0071] 이때, 비교 선택부(304)는 현재 시간에 해당하는 각 스테이트(state)의 해밍 디스턴스(hamming distance)를 이용하여 ACS 연산을 수행할 수 있다. 여기서, 해밍 디스턴스는, 같은 비트 수를 갖는 2진 부호 사이에 대응되는 비트값이 일치하지 않는 개수를 의미할 수 있다.

[0072] 예를 들어, 비교 선택부(304)는, 현재 시간에 해당하는 각 스테이트에 존재하는 하나 이상의 해밍 디스턴스 중 최소를 선택할 수 있다. 그리고, 비교 선택부(304)는, 선택한 최소의 해밍 디스턴스에 대응하는 브랜치 메트릭과 이전 시간 상태의 브랜치 메트릭을 누적하여 경로 메트릭(Path Metric, 이하, 경로값이라 칭함)을 각 스테이트 별로 계산할 수 있다. 이때, 비교 선택부(304)는, ACS연산의 속도를 향상시키기 위해 현재 시간에 해당하는 각 스테이트의 ACS 연산을 병렬화하여 수행할 수 있다.

[0073] 그러면, 역추적 계산부(305)는, 입력 데이터에 대해 TB 연산을 수행할 수 있다. 이때, 역추적 계산부(305)는, ACS 연산을 통해 계산된 각 스테이트의 경로 메트릭을 이용하여 어느 하나의 경로 메트릭을 선택할 수 있다.

[0074] 예를 들어, 역추적 계산부(305)는, 각 스테이트의 경로 메트릭을 비교하여 최소값에 해당하는 경로 메트릭을 선택할 수 있다. 다시 말해, 역추적 계산부(305)는, 각 스테이트에서 누적된 브랜치 메트릭 중 누적된 값이 가장 작은 브랜치 메트릭을 선택할 수 있다.

[0075] 그리고, 역추적 계산부(305)는, 선택된 경로 메트릭에 해당하는 경로(path)를 기준으로 현재 시간부터 처음 시간까지 상기 경로와 연결된 브랜치들을 역추적할 수 있다. 다시 말해, 역추적 계산부(305)는, 경로 메트릭을 기준으로 연결된 브랜치들을 역추적하여 입력 데이터를 복호화할 수 있다. 그리고, 복호화된 입력 데이터를 출력 데이터로 결정하여 출력할 수 있다.

[0076] 이하에서는, 본 발명의 일실시예에 따른 비터비 복호화 장치가 오류 결정부를 이용하여 비터비 복호화를 고속으로 처리하는 과정과 기존의 비터비 복호화 장치 간의 성능을 아래의 표 1을 참고하여 비교하고자 한다.

표 1

[0077]

			기존의 비터비 복호화	본 발명의 비터비 복호화
Logic utilization	LUTs		365	371
	Slice Registers		1,542	1,597
Total block memory bits			980	981
Maximum Frequency(Mhz)			250	250
Error Rate	1/1000	Output Rate (Mbit/s)	58.7	101.3
	1/100		58.6	72.8
	1/10		57.9	35.1

[0078] 표 1에 따르면, 본 발명의 일실시예에 따른 비터비 복호화 장치는, 오류 결정부를 이용하여 입력 데이터를 복호

화하므로 기존의 비터비 복호화 장치 대비 복호화 속도가 빨라졌음을 확인할 수 있다. 특히, 입력 데이터의 오류 발생 확률이 1% 미만인 경우, 본 발명의 일실시예에 따른 비터비 복호화 장치는, 기존의 비터비 복호화 장치보다 1.5배 빠른 처리 속도를 가질 수 있다.

[0079] 본 발명의 실시 예에 따른 방법들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0080] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

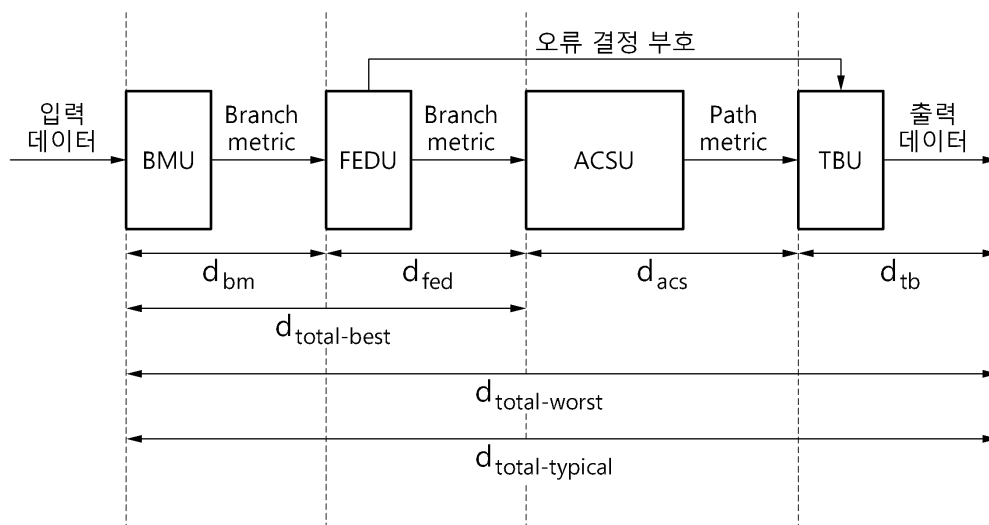
[0081] 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니 되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

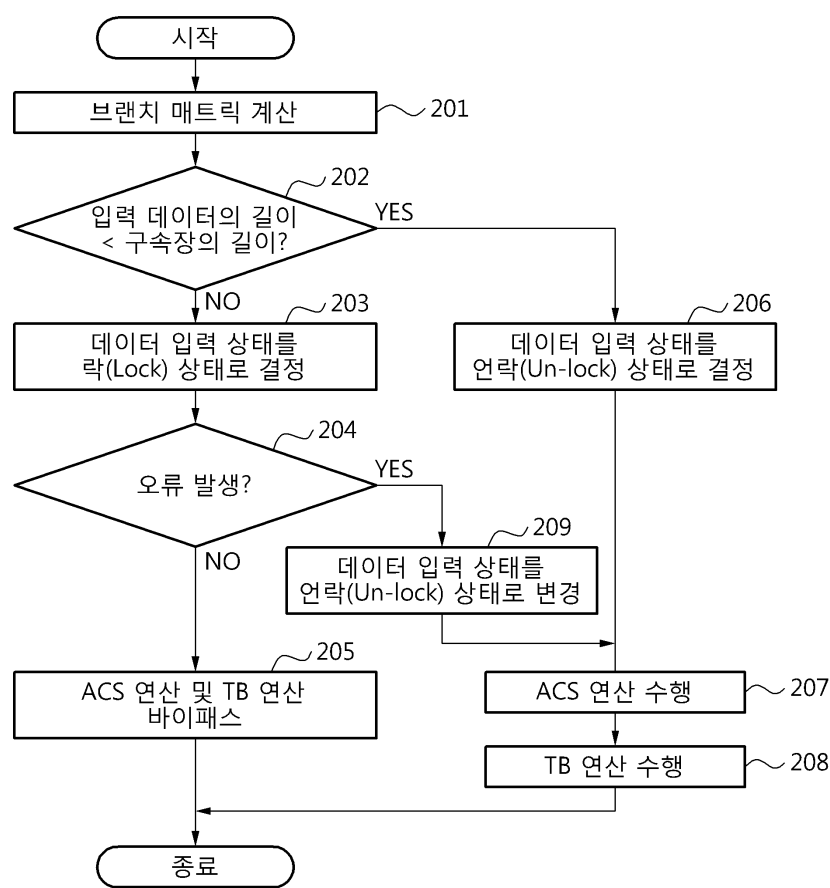
- [0082]
- 300: 비터비 복호화 장치
 - 301: 브랜치 메트릭 계산부
 - 302: 오류 결정부
 - 303: 연산 결정부
 - 304: 비교 선택부
 - 305: 역추적 계산부

도면

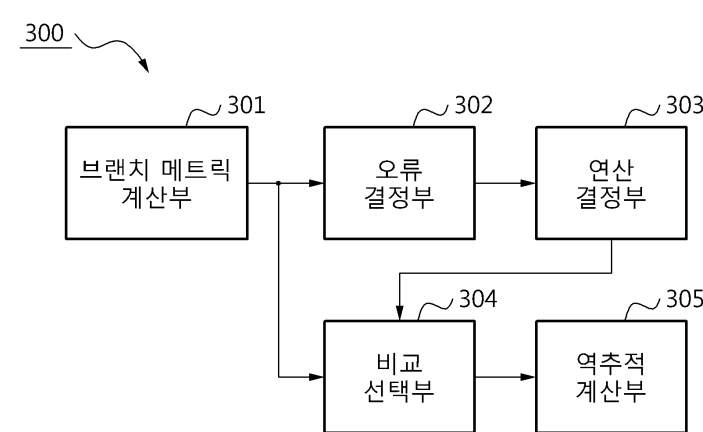
도면1



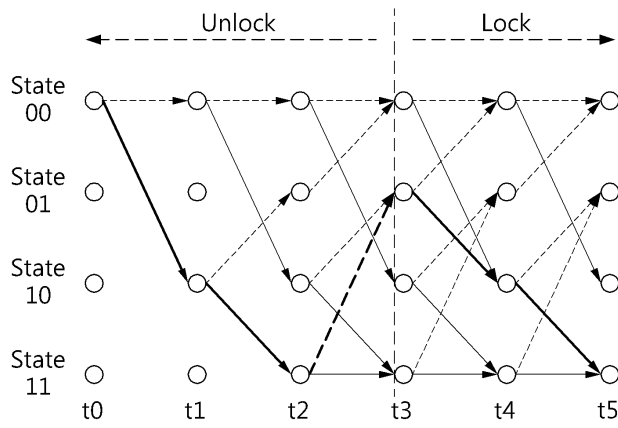
도면2



도면3



도면4



도면5

