

터보 복호를 위한 슬라이딩 윈도우 방식을 적용한 MAP 구조에 관한 연구

정회원 최 광 석*

A Study of MAP Architecture Adopting the Sliding Window Method for Turbo Decoding

Goang-seog Choi* *Regular Member*

요 약

터보 복호를 위한 MAP(Maximum A Posteriori) 알고리즘을 슬라이딩 윈도우 방식으로 설계하고 구현하였다. 먼저, 구현하는데 필요한 논쟁점들 즉, 슬라이딩 윈도우의 길이 및 상태 메트릭의 정규화 방식 등이 논의되고 그들의 최적 값을 모의실험을 통하여 얻었다. 또, 가지 메트릭 계산기를 포함한 복호기의 모든 구성요소의 회로가 제출되었다. 제안된 MAP구조는 버퍼 제어를 간단히 하여 슬라이딩 윈도우 크기 즉, 부-프레임 길이의 변경에 따라 쉽게 재설계 될 수 있다.

Key Words : Turbo code, MAP, SOVA, Sliding Window

ABSTRACT

The MAP algorithm is designed and implemented through the sliding window method for turbo decoding. First, the implementation issues, which are the length of the sliding window and the normalization method of state metrics are reviewed, and their optimal values are obtained by the simulation. All component schemes of the decoder including the branch metric evaluator are also presented. The proposed MAP architecture can be easily redesigned according to the size of sliding window, that is, sub-frame length because of its simplicity on buffer control.

I. 서 론

1993년에 발표된 터보 부호는 전력이 제한된 시스템에서 낮은 수신 전력으로도 샤논(Shannon)의 이론적 한계치에 근접하는 높은 성능을 나타내는 것으로 알려졌다^[1]. 터보 부호를 복호 하는 알고리즘에는 MAP 알고리즘과 SOVA (Soft-Output Viterbi Algorithm) 등이 있는데, 일반적으로 MAP 알고리즘이 SOVA보다 더 우수한 성능을 나타낸다고 알려져 있다. 한 예로 1024 비트 프레임, 8회

반복, 10^{-4} 의 BER (Bit Error Rate), 32 상태, AWGN (Additive White Gaussian Noise) 채널에서 MAP 알고리즘이 SOVA보다 0.7dB정도 부호이득이 있는 것으로 알려졌다^[2]. MAP은 각 복호 비트의 APP (A Posteriori Probability)를 최적화 하는 알고리즘으로, 구현의 복잡성으로 인해 알고리즘을 간략화 하는 연구가 많이 이루어졌다^[3,4,5]. 간략화 된 알고리즘을 바탕으로 MAP 구조 및 구현에 관한 논문들이 발표되었다. 입력과 출력 부호 심벌의 MAP 확률을 개선시키는 형태의 SISO (Soft In Soft

* 조선대학교 정보통신공학부 (gschoigs@chosun.ac.kr)

논문번호 : KICS2007-01-031, 접수일자 : 2007년 1월 26일, 최종논문접수일자 : 2007년 5월 3일

Out)^[6]과 몇 가지의 부호 상태와 부호율에 대해 복호 할 수 있는 연속(Continuous) MAP 복호기^[7] 등이 발표되었고, 또한 버퍼 메모리(Buffer memory)를 감소시키면서 MAP 복호기를 구현하는 직관적인 방법도 제시되었다^[8]. 또, 90년대 말에 발표된 논문^[9]에서는 고속 동작에 적합한 전속(Full-speed) 구조와 하드웨어 크기를 대폭 줄인 메모리(Memory) 구조를 제안하기도 했다.

본 논문에서는 터보 복호기를 위한 슬라이딩 방식 MAP 모듈의 구조를 제안한다. 제안된 구조는 이동 통신에 응용된다는 특성을 감안해서 저 전력으로 설계되고, 반복 횟수에 따라 성능이 좌우되는 터보 부호의 특성을 감안하여 고속으로 동작할 수 있게끔 설계되었다.

II. 구현 시 고려사항

터보 복호를 위한 MAP 모듈의 구조를 제안하기 위해서 먼저 결정해야할 사항들을 논의한다.

2.1 (Log-) MAP 알고리즘

지금까지 발표된 논문들^[4,5,6,8]을 근거로 해서 구현하고자하는 AWGN 채널 상에서의 (Log-) MAP 알고리즘을 다음과 같이 정리한다.

$$D_{k,i,m} = -(z_k + L_c x_k) d_k - L_c y_k c_k \quad (1)$$

$$A_{k,m} = \min^* (A_{k-1,b(0,m)} + D_{k-1,0,b(0,m)}, A_{k-1,b(1,m)} + D_{k-1,1,b(1,m)}) \quad (2)$$

$$B_{k,m} = \min^* (B_{k+1,f(0,m)} + D_{k,0,f(0,m)}, B_{k+1,f(1,m)} + D_{k,1,f(1,m)}) \quad (3)$$

$$L_k = \min^* (A_{k,m} + D_{k,0,f(0,m)} + B_{k+1,f(0,m)} | \forall m) - \min^* (A_{k,m} + D_{k,1,f(1,m)} + B_{k+1,f(1,m)} | \forall m) \quad (4)$$

$$\hat{e}_k = 0 \text{ if } L_k \geq 0, \quad 1 \text{ otherwise} \quad (5)$$

$$\min^* (A, B) = \min(A, B) - \log_e(1 + e^{-|A-B|}) \quad (6)$$

여기서, 시간 k , 상태 m , 정보 비트 d_k , 그리고 상응하는 패리티 비트 c_k 에서 상기 식들의 변수들은 다음과 같은 의미를 가진다.

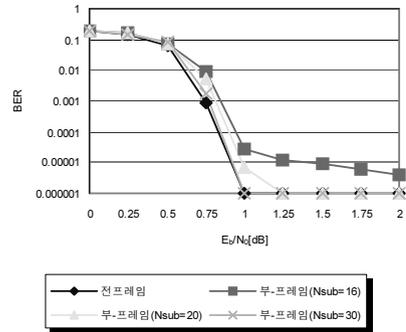
- 1) $i(= 0, 1)$ 는 정보 값, N 은 전체 프레임 길이,

x_k 와 y_k 는 정보 및 패리티 수신심벌, 그리고 $b(d_k=i, m)$ 와 $f(d_k=i, m)$ 는 역방향 및 순방향 상태를 각각 의미한다.

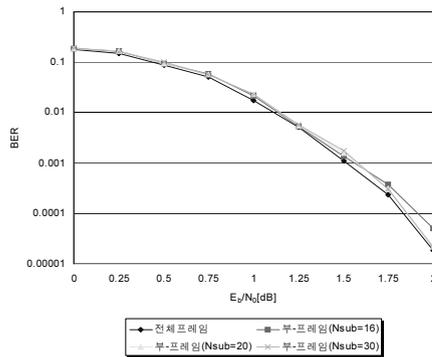
- 2) L_c 는 채널신뢰도(Channel reliability)이고, AWGN 채널에서 $-2/\sigma^2$ (σ^2 는 잡음분산)이다.
- 3) z_k 는 로그영역에서 입력 APriP (A Priori Probability) 비이다.
- 4) $D_{k,i,m}$, $A_{k,m}$ 와 $B_{k,m}$ 는 로그영역에서 각각 가지, 순방향 및 역방향 상태 메트릭(State metric)이다 (초기값 : $A_{0,0}=0$, $A_{0,m}=-\infty$, $B_{N,0}=0$, $B_{N,m}=-\infty$ ($m \neq 0$)).
- 5) L_k 는 로그 영역에서 각 복호 비트의 LLR (Log-Likelihood Ratio)이며, $\forall m$ 은 모든 상태를 의미하고, \hat{e}_k 는 추측된 비트이다.

2.2 슬라이딩 윈도우의 크기

복호 비트의 LLR 계산은 양방향의 상태 메트릭 뿐만 아니라 가지 메트릭(Branch metric)에 의존하



(a) 5,000 비트/프레임



(b) 320 비트/프레임

그림 1. 전체 프레임과 몇 가지 부-프레임의 BER 성능

기 때문에 복호지연과 많은 양의 메모리를 필요로 하는데, 이는 전적으로 인터리버(Interleaver)의 크기, 즉 입력 프레임 길이 N에 의존한다. 복호 지연과 메모리 양을 줄이기 위해 터보 부호에서 인터리버의 크기를 줄이면 BER의 열화를 일으키게 된다^[10]. 그런 이유로 프레임 길이를 줄이는 대신에 하나의 전체 프레임을 여러 개의 작은 프레임으로 나누어서 복호 알고리즘을 적용하여 허용할만한 성능 열화만을 갖는 슬라이딩 윈도우 방식이 이미 제시되었다. 본 논문에서 터보 복호에 슬라이딩 윈도우 방식을 적용하기 위해서 윈도우의 크기 즉, 부프레임(Sub-frame)의 크기 N_{sub} 에 따른 시스템 성능을 모의실험 하였다. 여기서 터보부호는 3세대 이동통신 3GPP의 표준으로 사용되는 규격을 사용하였다. 그림 1은 AWGN 채널 상에서 전체 프레임이 각각 5,000비트와 320비트이고 8회 반복인 경우 전-프레임과 몇 가지 부-프레임에 대해 모의실험 하여 얻은 BER 결과들이다. 각각 5×10^6 과 4.8×10^6 의 정보 비트들을 가지고 모의실험 하였다. 그림 1(a)과 1(b)에서 알 수 있듯이, N_{sub} 가 20인 경우에는 전체 프레임과 성능이 비슷하고, N_{sub} 가 30인 경우에는 거의 같다. 그러나 본 논문에서는 하드웨어의 복잡도(Complexity)를 감안하여 N_{sub} 의 값을 16으로 정하여 구조를 제안한다.

2.3 상태 메트릭의 정규화

복호 과정의 상태 메트릭 연산과정에서 발생하는 오버플로(Overflow)를 피하기 위해 반드시 정규화를 해야 한다. 종래의 직접적인 방식은 매 단계마다 전체 상태 메트릭들 중에서 가장 작은 메트릭을 찾아내서 모든 메트릭에서 그 값을 감산하는 것이다. 그러나 이 방법은 2^{K-1} 개의 뺄셈기와 $2^{K-1}-1$ 개의 비교기를 더 필요로 하는 고 비용을 요구하며 복호 지연을 일으키는 요인이 된다. 여기서, K는 터보 부호 시스템의 구속장(Constraint length)이다.

본 논문에서 채택한 방식은 모든 메트릭들 사이의 가능한 차 값은 상한 값을 가진다는 사실을 적용하는 것이다^[11,12]. 그리하여, 상태 메트릭들이 이진 보수의 정수로 표시되고, 연산이 2M 모듈러 연산이 행해질 때 자동적으로 정규화가 이루어진다. 여기서, M은 상태 메트릭들 사이에서 가장 큰 차이를 가진 랜덤 변수이다. 그리고 이진 보수 모듈러 연산이 ACS(Add-Compare-Selector)에 사용되었을 때, M은 최대와 최소 메트릭사이의 가능한 최대 차 값보다는 커야한다. M은 다음과 같이 얻어진다.

$$M > d_{free} \cdot J_{max} \tag{7}$$

여기서, d_{free} 는 부호의 자유거리 (free distance)이고, J_{max} 는 복조 채널 심벌의 최대 절대 값을 의미한다. 전체 프레임이 5,000비트이고 6 비트의 복조 심벌인 경우를 모의실험 한 결과, 가능한 최대 차 값이 300을 넘지 않는다는 사실을 확인하여, M을 512로 결정했다. 주어진 M을 가지고 상태 메트릭의 비트 길이 n은 다음과 같이 구할 수 있다.

$$n = 1 + \lceil \log_2 M \rceil \tag{8}$$

그래서 본 논문에서 구현한 시스템의 상태 메트릭의 비트 길이를 10으로 결정한다.

III. 구조와 동작

본 논문에서 제안한 MAP 구조와 동작은 그림 2와 그림 3에 나타난다. 이 동작은 5 단계로 나눌

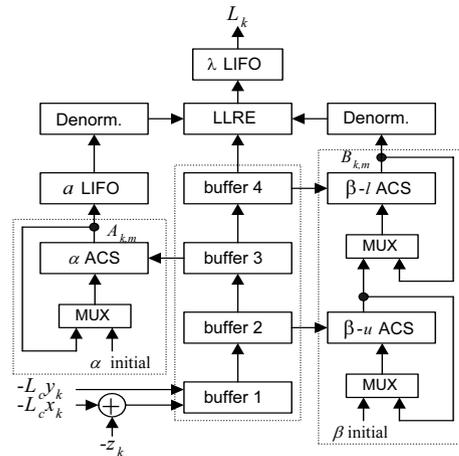


그림 2. 제안된 MAP 구조

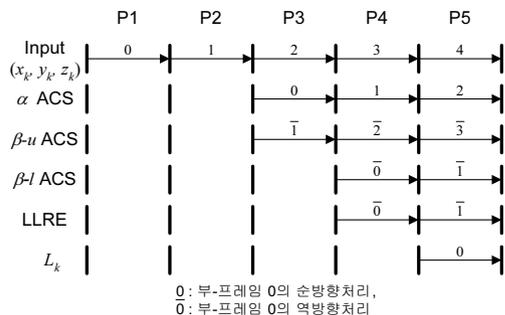


그림 3. 동작 타이밍

수 있는데, 여기서 한 단계는 하나의 부-프레임 입력 심벌들의 처리를 의미한다. 먼저 복조 수신심벌은 부-프레임 단위로 입력된다.

- 1) P1 단계에서 첫 번째 부-프레임 0이 버퍼 1에 순방향이고 직렬로 입력된다.
- 2) P2에서, 부-프레임 0은 버퍼 2에 병렬로 전달되고, 부-프레임 1은 버퍼 1에 직렬 입력된다.
- 3) P3에서, 다시 부-프레임 1과 0은 버퍼 2와 3에 병렬로 전달되어, βu ACS 블록과 a ACS 블록으로 출력되어 역방향 상태 메트릭 $B_{k,m}$ 의 시퀀스를 훈련시키고, 동시에 순방향 상태 메트릭 $A_{k,m}$ 을 계산한다. 계산된 $A_{k,m}$ 은 임시 버퍼 a LIFO(Last In First Out)에 저장된다.
- 4) P4에서, βu ACS는 부-프레임 1의 $B_{k,m}$ 를 훈련시키고, βl ACS는 부-프레임 0의 $B_{k,m}$ 를 계산하며, a LIFO는 부-프레임 0의 $A_{k,m}$ 값을 역방향으로 출력하여, 동시에 LLRE는 부-프레임 0의 복호 비트의 LLR, 즉 L_k 를 계산한다.
- 5) 마지막으로 P5에서, λ LIFO는 역방향으로 입력된 복호 비트의 L_k 값을 순방향으로 출력시킨다.

IV. 구성 요소들

4.1 가지 메트릭 계산기

그림 4는 가지 메트릭 계산기이며, 가지 메트릭 D_k 는 식 (1)에 따라 $-z_k$, $-L_c x_k$ 와 $-L_c y_k$ 를 조합해서 총 4가지 형태의 가지 메트릭을 생성한다. 생성된 가지 메트릭들은 트렐리스(trellis)의 각 상응하는 가지와 사상(mapping)해서 상태 메트릭을 개선하는데 이용된다. 예를 들어 $d_k c_k$ 의 값이 11인 $D_k(d_k c_k=11)$ 는 $-(z_k, +L_c x_k + L_c y_k)$ 로 생성된다.

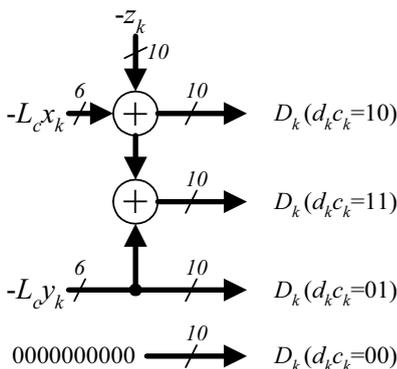


그림 4. 가지 메트릭 계산기

4.2 상태 메트릭 계산기

MAP방식에서 상태 메트릭 $A_{k,m}$ 와 $B_{k,m}$ 의 계산은 그림 5와 같이 도식적으로 나타낼 수 있다. 먼저 $A_{k,m}$ 의 경우를 보자. 가지 메트릭 $D_{k-1,0,b(0,m)}$ 와 $D_{k-1,1,b(1,m)}$ 가 이전 상태 메트릭 $A_{k-1,b(0,m)}$ 와 $A_{k-1,b(1,m)}$ 에 각각 더해져서 비교 선택된다. $B_{k,m}$ 의 경우도 역방향으로 진행된다는 사실을 제외하고는 동일하다.

이러한 일련의 과정을 처리하는 2진 보수 모듈러 정규화를 가진 $A_{k,m}$ 의 ACS는 그림 6에 나타난다. 가지 메트릭과 이전 상태 메트릭이 더해진 2개의 예비 메트릭들, 즉 A_0 와 A_1 이 비교되기 위해서 먼저

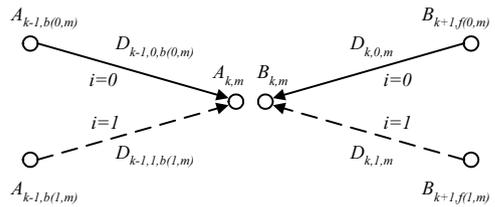


그림 5. $A_{k,m}$ 와 $B_{k,m}$ 도식적 표현

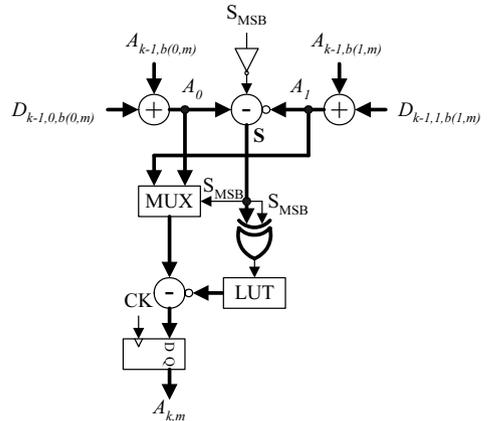


그림 6. $A_{k,m}$ 의 ACS

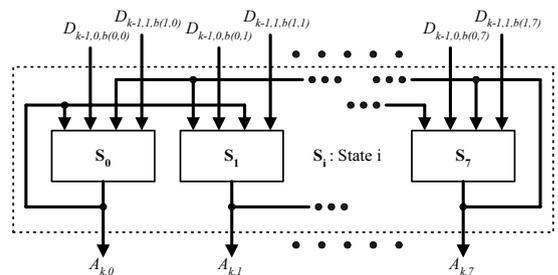


그림 7. 순방향 상태 메트릭 계산기

감산을 한다. 차의 MSB(Most Significant Bit) 즉, S_{MSB} 가 0이면, MUX의 왼쪽, 즉 A_i 이 선택된다. 그리고 LUT(Look Up Table)에서 출력하는 식 (6)의 로그적인 정정 항(logarithmic correction term)과 결합해서 새로운 상태 메트릭이 생성된다. 그리고 상단에 있는 S_{MSB} 의 반전 회로와 LUT 입력 단의 Exclusive OR는 예비 메트릭의 차 값을 항상 양수로 만들어서 LUT의 용량도 반으로 줄이고 절대 값을 만들어야하는 처리과정을 없앤다. 그림 7은 8개 ACS를 포함하는 순방향 상태 메트릭 계산기를 나타내며, 역방향 상태 메트릭 계산기도 같은 방식으로 구성할 수 있다.

4.3 반-정규화 회로

복호 비트의 LLR을 계산하기 위해서는 상태 메트릭의 오버플로우를 해소 해야하는데 그림 8은 순방향 상태 메트릭의 반-정규화 회로이다. 먼저, 오버플로우가 생겼는지를 확인한다. 즉, 모든 순방향 상태 메트릭 중에서 MSB와 2번째 MSB의 값이 동시에 01과 10이 존재하면 오버플로우가 발생한 것이다. 그래서 모든 순방향 상태 메트릭의 MSB를 반전시킴으로써 반-정규화 과정을 행한다. 이 방식은 역방향 상태 메트릭에서도 동일하게 적용된다.

4.4 LLR 계산기

그림 9는 복호 비트 LLR을 구하는 도식적 표현이다. 식 (4)에 의해 복호 비트의 LLR은 정보 0의 로그 APP에서 정보 1의 로그 APP를 감산함으로써 구할 수 있다. 그림 10은 상기에서 설명한 내용을 바탕으로 복호 비트 LLR을 구하는 회로이다. 여기

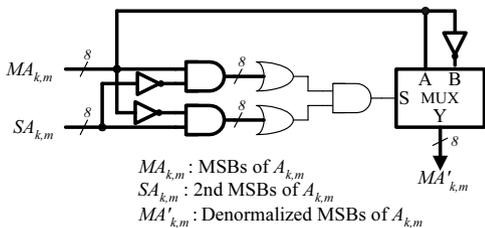


그림 8. $A_{k,m}$ 의 반-정규화

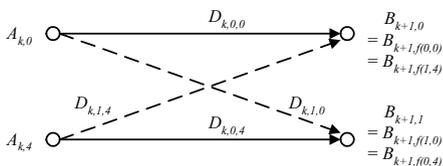


그림 9. 복호 비트 LLR의 도식적 표현

서 CS(Compare-Select)는 상태 메트릭을 구하는 데 사용되어진 ACS 회로에서 앞단의 가산기만 빠진 회로이다. 상태-병렬의 구조이므로 한 클럭 내에서 한 복호 비트의 LLR을 구한다. MAP방식의 터보 복호기에서는 이상의 복호 비트 LLR을 계속해서 개선시키면서 최종적으로 식 (5)과 같은 추측 비트를 결정한다.

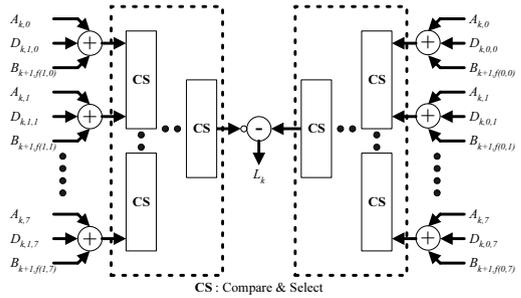


그림 10. 복호 비트의 LLR 계산기

V. 비교 분석 및 실험 결과

5.1 플립-플롭과 가산기의 선택

플립-플롭(Flip-flop)은 시스템 클럭에 연결되는 기본 셀이기 때문에 저 전력 및 고속 응용에서 매우 중요한 구성 요소이다. 4개의 트랜지스터만 클럭에 연결되어 전력을 작게 소모하면서도 비교적 고속으로 동작하는 9T 플립-플롭을 선택하였다.

가산기도 MAP 모듈을 고속 및 저전력으로 구현 하는데 있어서 중요한 구성 요소이다. 최적의 10비트 가산기를 선택하기 위하여 리플 캐리(Ripple carry), 상수 블록 캐리 생략(Constant block carry skip), 캐리 앞섬(Carry look-ahead) 및 캐리 선택(Carry select) 가산기에 대해 모의실험을 하였다. 2.5V의 전원에서 10ns의 주기로 100개의 랜덤 입력을 통하여 표 1과 같은 특성을 얻었고, 전송 지연 및 전력소모를 고려하여 상수 블록 캐리 생략 가산기를 선택하였다.

표 1. 10 비트 가산기들의 특성들

가산기 형태	트랜지스터 수	최대전송 지연(ns)	평균전력 소비(mW)
리플 캐리 가산기	280	0.86	0.15
캐리 생략 가산기	340	0.59	0.17
캐리 앞섬 가산기	428	1.39	0.25
캐리 선택 가산기	632	0.59	0.29

5.2 상태 메트릭의 정규화 방식 분석

본 논문에서 제안한 이진 보수 모듈러 ACS를 평가하기 위해서 종래의 비교-감산을 통한 정규화 방식과 비교하였다. 먼저 순방향 상태 메트릭 계산기를 상태-병렬로 해서 50개의 랜덤 입력을 10ns의 주기로 입력해서 표 2와 같은 결과를 얻었고, 종래 방식에 비해 속도 및 전력소모 면에서 훨씬 유리함을 알 수 있다.

5.3 제안된 구조의 비교 분석

제안된 구조의 메모리량을 평가하기 위해서, 일반적인 전-프레임 MAP 구조와 [9]에서 제안한 메모리 구조에 사용된 메모리량과 비교하였다. 표 3은 3GPP에서 한 프레임이 5,000 비트인 경우의 메모리 요구량을 나타내며, 가지 및 상태 메트릭의 비트-길이는 각각 6 비트, 10 비트이고, 부-프레임의 길이는 16이다. 표 3에서 알 수 있듯이 제안된 MAP 구조는 초기 메모리를 제외하고는 가장 작은 하드웨어 크기를 가지는 메모리 구조와 같다.

복호에 필요한 클럭 수도 시스템 성능을 결정하는데 중요한 요소 중에 하나이다. 특히 터보 부호는 반복 복호로 인한 성능 향상이 기대되므로 가급적 복호 하는데 필요한 클럭 수를 줄이는 것이 중요하다. 표 4는 3GPP에서 한 프레임이 5,000 비트인 경우의 복호에 필요한 클럭 수를 나타낸다. 여기서 메모리 구조에 사용된 RAM의 쓰기와 읽기는 각각 한 클럭이 필요한 것으로 가정한다. 표 4에서와 같이 제안된 구조는 가장 적은 복호 클럭 수를 나타낸다. 이는 제안된 구조에서 메트릭들을 저장하는데 RAM을 사용하는 대신에 플립-플롭과 MUX를 사용해서 한 클럭에 읽고 쓰는 동작이 동시에 이루어지기 때문이다.

VI. 결론

본 논문에서 터보 부호의 복호에 필요한 MAP 모듈을 슬라이딩 윈도우방식으로 구현하는 구조를 제안하였다. 먼저, 저 전력과 고속의 응용을 위해서 상수 블록 캐리 생략 가산기와 9T 플립-플롭을 선택하였고, 전-프레임에 비해 허용할 만한 성능을 지닌 부-프레임의 크기를 모의실험을 통해 알아냈다. 또한 고속 응용에 적합한 상태 메트릭 정규화를 위해서 2진 보수 부호의 모듈러 연산을 채택하고 최적의 상태 메트릭 비트 길이를 얻었다. 선택된 기본 셀 및 부-블록들은 모의실험을 거쳐 성능을 평가하

여 검증하였다.

표 2. 순방향 상태 메트릭 계산기들의 특성들

정규화 방식	최대 전송지연	평균전력 소비(mW)
종래의 비교와 감산 방식	5.01	28.5
이진 보수 모듈러 방식	1.38	8.7

표 3. 메모리 요구량

	전-프레임	메모리 구조	제안된 구조
초기 메모리	-	-	$16 \times 10 \times 2 = 320$
버퍼 메모리	$5,000 \times 6 \times 4 = 120,000$	$(16 \times 10 \times 2) \times 3 = 960$	$(16 \times 10 \times 2) \times 3 = 960$
상태 메모리	$5000 \times 10 \times 8 = 400,000$	-	-
α LIFO	-	$16 \times 10 \times 8 = 1,280$	$16 \times 10 \times 8 = 1,280$
λ LIFO	-	$16 \times 10 = 160$	$16 \times 10 = 160$
Total	520,000 bits	2,400 bits	2,720 bits

표 4. 클럭 요구량

	전-프레임	메모리 구조	제안된 구조
초기 메모리	-	-	$16 \times 1 = 16$
버퍼 메모리	$5,000 \times 2 = 10,000$	$(16 \times 3 + 5000) \times 2 = 10,096$	$(16 \times 3) + 5000 = 5,048$
상태 메모리	$5,000 \times 2 = 10,000$	-	-
α LIFO	-	$16 \times 2 = 32$	$16 \times 1 = 16$
λ LIFO	-	$16 \times 2 = 32$	$16 \times 1 = 16$
Total	20,000clocks	10,160clocks	5,096clocks

참고 문헌

- [1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes(1)", *Proc. of ICC*, pp. 1064-1070, May 1993.
- [2] P. Robertson, E. Villebrun and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", *Proc. of ICC*, pp1009-1013, 1995.
- [3] L.R. Bahl, J.Cocke, F.Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. On Information Theory*, pp.284-287, Mar. 1974.

[4] S. Pietrobon and S. Barbuлесcu, "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional Codes", *Int. Symp. On Inform. Theory*, pp. 1073-1077, Nov. 1994.

[5] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes", *TDA Progress Report 42-124*, pp.63-87, Feb. 1996.

[6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A Soft-Input Soft-Output Maximum A Posteriori(MAP) Module to Decode Parallel and Serial Concatenated Code", *TDA Progress Report 42-127*, pp. 1-20, Nov. 1996.

[7] S. Pietrobon, "Implementation and performance of a turbo/MAP decoder", *Int. Journal of Satellite Comm. vol. 16*, pp.23-26, Jan. 1998.

[8] A.J. Viterbi, "An Intuitive Justification and a Simplified Implementation of the MAP Decoder for Convolutional Codes", *IEEE Journal on Sel. Areas in Comm.*, vol. 16, no.2, pp.260-264, Feb. 1998.

[9] Masera, G. Piccinini, M. R. Roch, and M. Zamboni, "VLSI Architectures for Turbo Codes", *IEEE Trans. On VLSI System*, vol.7, no.3, pp.369-79, Sep. 1999

[10] F. Acikel, "Implementation Issues for High Turbo Codes on BPSK/QPSK Channels", *IEEE Globecom*, pp.148-152, 1999.

[11] P. Hekstra, "An alternative to metric rescaling in Viterbi decoder", *IEEE Trans. Comm.*, vol.37, pp. 1220-1222, Nov. 1989.

[12] I. Onyszchuk, K. M. Cheung, and O. Collins, "Quantization Loss in Convolutional Decoding", *IEEE Trans. Comm.*, vol.41, pp.261-65, Feb. 1993.

최 광 석 (Goang-seog Choi)

정회원



1987년 2월 부산대학교 전자
공학과 졸업

1989년 2월 부산대학교 전자
공학과 석사

2002년 2월 고려대학교 전자
공학과 박사

1989년~2006년 : 삼성전자 DM
연구소 수석연구원

2006년 3월~현재 조선대학교 정보통신공학부 전임
강사

<관심분야> 통신 및 디지털 미디어 ASIC/SOC 설계