

논문 2005-42SD-9-9

# 3세대 이동통신에 적합한 슬라이딩 윈도우 로그 맵 터보 디코더 설계

(Turbo Decoder Design with Sliding Window Log Map for 3G W-CDMA)

박 태 근\*, 김 기 환\*\*

(Tae Geun Park and Ki Hwan Kim)

## 요 약

로그 맵 복호 알고리즘 기반의 터보 디코더는 뛰어난 복호 성능에도 불구하고, 반복적 연산으로 인한 인터리버 크기에 비해 하는 많은 양의 메모리와 높은 하드웨어 복잡도가 단점으로 지적된다. 이에 본 논문에서는 이전 연구 결과를 바탕으로 많은 양의 메모리의 절감과 하드웨어 복잡도를 감소시킨 3G W-CDMA 시스템에 적합한 터보 디코더를 설계하였다. 하드웨어 복잡도와 복호 성능간의 균형을 고려하여 수신정보, 사전정보, 상태 메트릭을 각각 5비트, 6비트 그리고 7비트로 할당하였고, 로그 맵 복호 알고리즘의 주연산인 MAX\* 연산 중 계산도가 큰 오류 보정 함수를 근사화한 조합회로로 구성하여 하드웨어 부담을 감소하였으며, 윈도우 블록의 길이가 32인 슬라이딩 윈도우 기법을 적용하였다. 본 논문에서 제안한 터보 디코더는 0.35 $\mu$ m Hynix CMOS technology로 합성한 합성 결과로부터 Eb/No가 1dB, 인터리버 크기가 2000, 5번의 반복 복호에서 10<sup>-6</sup> 이하의 비트 오류율을 달성하였으며, 이때 최고 9Mbps의 복호 성능을 발휘한다.

## Abstract

The Turbo decoders based on Log-MAP decoding algorithm inherently requires large amount of memory and intensive complexity of hardware due to iterative decoding, despite of excellent decoding efficiency. To decrease the large amount of memory and reduce hardware complexity, the result of previous research. And this paper design the Turbo decoder applicable to the 3G W-CDMA systems. Through the result of previous research, we decided 5-bits for the received data, 6-bits for a priori information, and 7-bits for the quantization state metrics. The error correction term for MAX\* operation which is the main function of Log-MAP decoding algorithm is implemented with very small hardware overhead. The proposed Turbo decoder is synthesized in 0.35 $\mu$ m Hynix CMOS technology. The synthesized result for the Turbo decoder shows that it supports a maximum 9Mbps data rate, and a BER of 10<sup>-6</sup> is achieved(Eb/No=1.0dB, 5 iterations, and the interleaver size  $\geq$  2000).

**Keywords :** Turbo Codes, Turbo Decoder, W-CDMA, Log-MAP

## I. 서 론

3세대 이동통신 시스템에서는 음성, 문자 그리고 고속의 멀티미디어 데이터의 신뢰성 있는 전송이 요구되

는데, 이러한 고속 데이터의 신뢰성을 높이기 위해서는 오류에 강한 채널 코딩 방식이 필요하다. 채널 코딩 방식에 대한 많은 연구와 기술의 발달로 인하여 세계적으로 여러 방식이 제안되고 있는데, 그 중에서도 3GPP(3rd Generation Partnership Project)에서 저속 음성 통신용으로 컨볼루션 코드(Convolution Codes)와 고속 데이터 전송용으로 터보 코드(Turbo Codes)가 채택되었다<sup>[1]</sup>.

1993년 C.Berrou 등은 Bahl 등에 의해 제안된 BCJR 알고리즘을 개량하여 인터리버를 경유하는 반복 복호를

\* 정회원, 가톨릭대학교 정보통신전자공학부  
(School of Information, Communications and Electronics Engineering, Catholic University)

\*\* 정회원, LG전자 이동통신기술연구소  
(Mobile Communication Technology Research Lab. LG Electronics Inc.)

접수일자: 2005년3월3일, 수정완료일: 2005년8월16일

통한 Shannon의 채널 용량 한계에 근접하는 성능의 터보 코드를 제안하였다<sup>[2][3]</sup>. 초기 터보 코드는 연산기의 복잡성과 복호 시간의 지연으로 인해 실시간 처리에 어려움이 따랐으나 많은 사람들에 의하여 지속적으로 연구되어 왔고, 저전력, 고성능 그리고 실시간 처리가 가능한 터보 디코더를 구현하기 위한 연구가 진행되고 있다<sup>[4]</sup>.

터보 복호 알고리즘은 크게 MAP(Maximum a Posteriori) 기반의 복호 알고리즘과 Viterbi 알고리즘에 기반한 SOVA(Soft Output Viterbi Algorithm) 복호 알고리즘으로 분류된다. MAP 복호 알고리즘은 이론적인 최적 알고리즘으로 SOVA 알고리즘에 비해 약 2배의 좋은 성능을 보이나 다수의 곱셈과 지수 연산으로 인하여 약 4배의 복잡한 구조와 많은 메모리가 필요한 단점이 있다<sup>[5]</sup>.

로그 맵(Log-MAP) 복호 알고리즘은 MAP 복호 알고리즘을 로그 영역으로 치환함으로써 복호 과정의 연산 복잡도가 감소하였지만, 여전히 인터리버 크기에 비례하는 많은 메모리를 필요로 한다. 슬라이딩 윈도우(Sliding Window) 기법은 로그 맵 복호 알고리즘의 단점인 메모리 필요량을 감소시키기 위하여 제안된 방법으로, 입력 프레임을 여러 개의 윈도우 블록으로 나누어 각각의 윈도우 블록 단위로 복호하고, 더미 연산 과정을 포함하여 로그 맵 복호 알고리즘과 비슷한 성능을 나타낸다<sup>[6][7]</sup>.

터보 디코더 설계에 관한 연구는 터보 코드 고유의 특징인 반복 복호에 있어서의 성능 최적화 그리고 복호 알고리즘의 효율적인 하드웨어 구조를 초점으로 터보 복호 알고리즘을 TMS320C55x 프로세서에 구현하는 연구<sup>[8]</sup>와 필요 메모리를 줄이기 위해 상태 매트릭 difference-storing 기법을 도입한 연구<sup>[9]</sup> 그리고 lookahead 변환 기법의 고속 MAP 디코더에 관한 연구<sup>[10]</sup>가 진행되어 오고 있다.

본 논문에서는 이전 연구<sup>[11]</sup>인 터보 디코더의 설계에 대한 다양한 설계 이슈들을 모의실험을 통하여 성능 및 복잡도에 미치는 영향을 분석하여 3세대 이동통신에 적합하고, 슬라이딩 윈도우 기법을 도입하여 필요 메모리 양을 감소시킨 로그 맵 터보 디코더의 구조를 제안한다.

본 논문의 제 II장에서는 터보 인코더와 디코더의 구조와 복호과정, 대표적인 복호 알고리즘을 소개한다. 제 III장에서는 메모리 사용량과 복잡도를 낮춘 슬라이딩 윈도우 로그 맵 터보 디코더의 구조를 제안하고, 제 IV장에서는 모의실험을 통하여 제안한 슬라이딩 윈도우

로그 맵 터보 디코더의 구현을 보이며, 다른 터보 디코더와의 성능을 비교 분석한다. 마지막으로 제 V장에서는 본 논문의 결론을 제시한다.

## II. 터보 코드

### 2.1 터보 인코더와 디코더

일반적으로 병렬 연결 컨볼루션 코드(Parallel Concatenated Convolutional Codes: PCCC)를 터보 코드라고 하며, 터보 인코더는 프레임 단위로 입력되는 정보 비트 열로부터 두 개 이상의 병렬 연결된 구성 부호기를 통하여 패리티 비트 열을 생성한다. 터보 인코더는 구성 부호로 순환 조직형 컨볼루션 코드(Recursive Systematic Convolutional Codes: RSC)를 사용하며, 패리티 출력에 천공 기법을 사용하여 필요한 부호율을 생성한다<sup>[12]</sup>.

그림 1은 부호율이 1/3인 3GPP 표준 터보 인코더의 구조이다. 입력  $X_k$ 는 구성 부호기에 의해서 부호화 되어 패리티 비트  $Z_k$ 를 생성하고, 입력  $X_k$ 가 인터리버를 통하여  $X'_k$ 로 재배열 한 후 구성 부호기에 의해 부호화 되어 패리티 정보  $Z'_k$ 를 출력한다<sup>[1]</sup>.

터보 디코더는 그림 2와 같이 터보 인코더의 구성 부호기에 대응하는 구조로써 두 개의 SISO(Soft-In Soft-Out) 디코더로 구성된다. 또한 수신된 정보의 순서를 바꾸는 인터리버와 디인터리버 그리고 경판정을 통하여 출력을 결정하는 경판정 블록으로 구성되어, 두 SISO 디코더가 서로 데이터를 교환함으로써 반복 복호가 가능한 재귀 구조를 갖는다<sup>[13]</sup>.

부호율이 1/3인 3GPP 표준 터보 인코더를 이용하여 부호화 되고 전송된 정보(systematic) 비트와 두 개의 패리티(parity) 비트는 터보 디코더의 입력으로 사용된

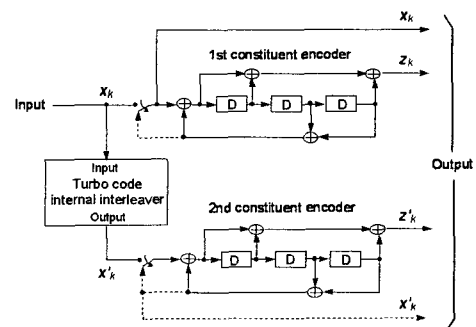


그림 1. 3GPP 표준 터보 인코더  
Fig. 1. Turbo Encoder of 3GPP.

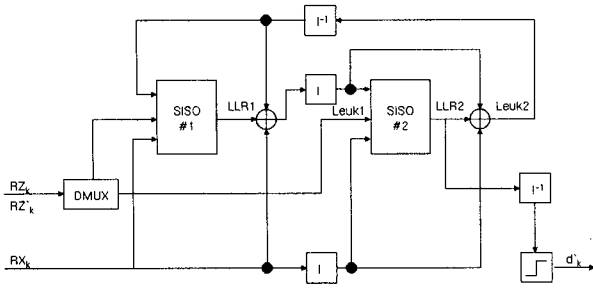


그림 2. 로그 맵 터보 디코더  
Fig. 2. Turbo decoder with log map.

다. 먼저 첫 번째의 SISO 디코더를 통하여 연관정 된 결과는 인터리버를 통하여 재배열되고 다시 다른 SISO 디코더를 통하여 연관정을 반복함으로써 지속적으로 오류를 수정하고, 최종적으로 경관정을 수행한다.

2.2 로그 맵 복호 알고리즘

MAP 복호 알고리즘은 Viterbi 알고리즘에 비해 복잡하지만, 경관정 출력의 성능은 Viterbi 알고리즘의 성능과 유사하여 터보 코드에 이용되고 있다. MAP 복호 알고리즘의 단점인 알고리즘의 복잡성을 감소시키기 위하여 1995년 P.Robertson 등은 재귀 연산을 로그 영역으로 치환하여 적은 성능 저하로 MAP 복호 알고리즘의 연산 복잡도를 상당히 감소시키는 로그 맵 복호 알고리즘을 제안하였다<sup>[14]</sup>.

로그 맵 복호 알고리즘은 식 (1)과 같은 E 함수를 이용하여 MAP 복호 알고리즘을 간략화 할 수 있다.

$$xEy = \ln(e^x + e^y) = x + \ln(1 + e^{-x+y})$$

$$= y + \ln(1 + e^{-y+x}) = \max(x, y) + \ln(1 + e^{-|y-x|})$$
(1)

식 (1)의 E 함수를 이용하여 가능도비 (Log-Likelihood Ratio: LLR)를 나타내면 다음과 같다.

$$L(d_k) = E_m(A_k^1(m)B_k^1(m)) - E_m(A_k^0(m)B_k^0(m))$$
(2)

여기서, 순방향 상태 메트릭(Forward State Metrics)  $A_k^i(m)$ 와 역방향 상태 메트릭(Backward State Metrics)  $B_k^i(m)$  그리고 가지 메트릭(Branch Metrics)  $D_k^i(R_k, m)$ 은 각각 식 (3), 식 (4) 그리고 식 (5)와 같이 나타낼 수 있다.

$$A_k^i(m) = D_i(R_k, m) + E_{m=0}^{2N-1} A_{k-1}^i(S_b^j(m))$$
(3)

$$B_k^i(m) = E_{m=0}^{2N-1} B_{k+1}^i(S_f^j(m)) + D_j(R_{k+1}, S_f^j(m))$$
(4)

$$D_i(R_k, m) = \frac{2}{\sigma^2} (x_k^i + y_k Y_k^i(m))$$
(5)

$S_b^j(m)$ 은 입력이  $j$ 일 때 다음 상태가  $m$ 이 되는 상태,  $S_f^i(m)$ 은 입력이  $i$ 이고 현재 상태가  $m$ 일 때의 다음 상태,  $\sigma^2$ 은 AWGN(Additive White Gaussian Noise) 채널에 대한 잡음의 분산 그리고  $Y^j(m)$ 은 입력이  $j$ 이고 상태가  $m$ 일 때의 디코더 출력을 의미한다.

2.3. 슬라이딩 윈도우

슬라이딩 윈도우 기법은 로그 맵 복호 알고리즘에서 가지 메트릭과 상태 메트릭을 인터리버의 크기, 트렐리스 상태(Trellis State)의 수 그리고 많은 양의 메모리가 필요한 단점을 보완하기 위하여 입력 프레임을 크기가 제한된 윈도우 블록으로 나누고 각 윈도우 블록 크기만큼 순방향 상태 메트릭, 역방향 상태 메트릭 그리고 가능도비를 연산함으로써 필요 메모리를 감소시킬 수 있는 기법이다.

슬라이딩 윈도우 기법을 역방향 상태 메트릭에 적용할 경우 각 윈도우 블록에서 역방향 상태 메트릭의 초기 값을 구하는 과정에서 손실의 발생으로 성능이 저하될 수 있기 때문에 역추적 과정을 설정하여 역방향 상태 메트릭의 손실을 줄일 수 있다.

그림 3.(a)는 MAP 복호 알고리즘의 데이터 의존(dependency)을 나타낸다. 수평 축은 알고리즘의 복호 시간을 나타내고 수직 축은 인터리버의 크기  $N$ 이다. 모든 복호 동작은 각각의 수직선 상에서 실행한다.

그림 3.(b)는 슬라이딩 윈도우 기법을 적용한 동작을 나타내는데 입력 프레임을 윈도우 블록  $L$ 로 나누어 짧은 점선의 더미 역방향 상태 메트릭 연산, 긴 점선의 유

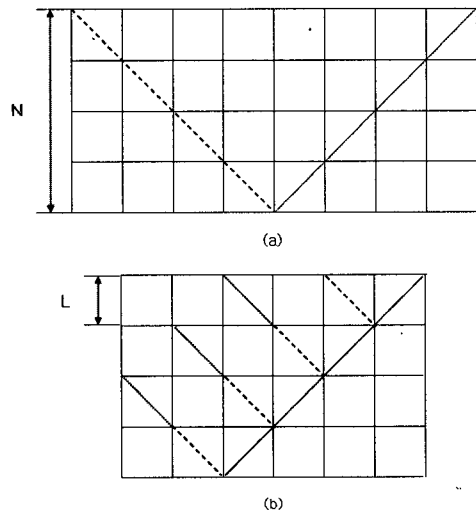


그림 3. 일반적인 방법과 슬라이딩 윈도우  
Fig. 3. General method versus sliding window.

효 역방향 상태 메트릭 연산 그리고 실선의 순방향 상태 메트릭 연산을 각 단계별로 동시에 진행하여 데이터 처리량을 높일 수 있다<sup>[15]</sup>

### III. 제안한 구조

일반적인 로그 맵 터보 디코더는 구현에 있어 반복적인 복호로 인한 복호속도의 저하 및 입력 프레임 길이의 가지 메트릭과 상태 메트릭을 저장할 수 있는 필요 메모리 용량등이 큰 부담을 갖기 때문에 로그 맵 복호 알고리즘을 그대로 적용하는데 어려움이 따른다. 이에 슬라이딩 윈도우 기법을 적용하여 3세대 이동통신에 적합한 로그 맵 터보 디코더의 구조를 제안한다.

#### 3.1 전체구조

본 논문에서 제안한 로그 맵 터보 디코더는 기존의 로그 맵 터보 디코더의 기본 구조를 바탕으로 그림 4와 같이 휴대용 기기에 적합한 구조를 가지기 위하여 하드웨어 자원의 부담이 큰 SISO 디코더를 한 개만 사용하였고, 고정 소수점 비트 할당을 통하여 하드웨어 복잡도를 줄였으며, 슬라이딩 윈도우 기법을 도입하고 메모리 분할 저장 방식을 적용하여 일련의 정보 열로 이루어진 입력 프레임 길이의 가지 메트릭 저장에 필요한 메모리 용량을 제거 하였고, 상태 메트릭의 필요 메모리 용량을 최소로 하였다. 또한 로그 맵 기반의 터보 디코더에서 제일 이용도가 큰 비교 선택기 내의 오류 보정 연산을 조합회로로 대체하여 하드웨어 복잡도 및 연산시간을 감소하였다.

제안한 슬라이딩 윈도우 로그 맵 터보 디코더는 다음과 같이 동작한다. 우선 일련의 정보 열로 이루어진 수신된 입력 프레임을 인터리버 크기만큼 받아 Pre Module에 4부분으로 분할하여 저장한다. 수신된 순서에 따른 정보 비트와 패리티 비트를 SISO 디코더에 입

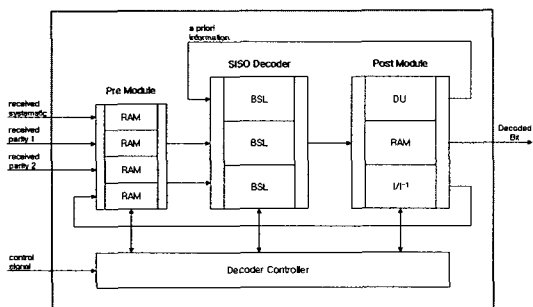


그림 4. 제안한 슬라이딩 윈도우 로그 맵 터보 디코더  
Fig. 4. Proposed turbo decoder with sliding window.

력하여 연성 출력을 얻어내고, 이는 Post Module에서 인터리빙 되어, 다시 SISO 디코더에 정보 비트, 패리티 비트와 함께 입력으로 사용된다. 이때 정보 비트는 수신된 정보 비트 순서 열에 대하여 인터리빙된 순서 열을 갖는다. 다시 SISO 디코더로부터 출력된 연성 출력은 일정 횟수만큼 반복 복호를 수행하여, SISO 디코더로부터의 연성 출력이 최종 경관정 과정을 거쳐 복호화된 정보를 출력한다.

#### 3.2 SISO

그림 5의 SISO 디코더는 일련의 정보 열로 이루어진 입력 프레임으로부터 슬라이딩 윈도우 블록 크기로 나뉘어 동시에 세 가지의 수신정보와 사전정보를 입력으로 사용한다.

가지 메트릭 저장에 필요한 필요 메모리 용량과 복호 시간을 단축하기 위하여 다음 절의 간단한 가지 메트릭 연산기(Branch Metric Unit: BMU)의 구조를 갖는다. 슬라이딩 윈도우 블록 크기의 수신정보와 사전정보를 이용하여 그림 6에 보여진 순서로 복호 연산을 수행한다.

순방향 메트릭 연산기인 SMU(State Metric Unit)는

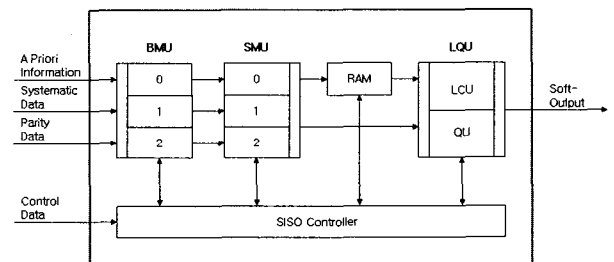


그림 5. SISO 디코더의 구조  
Fig. 5. Structure of SISO decoder.

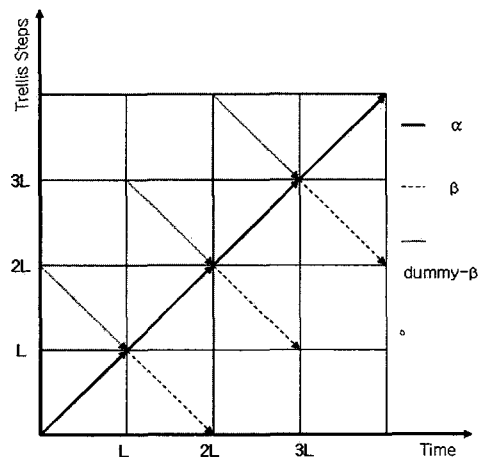


그림 6. 제안한 구조에 적용된 복호 순서  
Fig. 6. Decoding flow of sliding window.

입력 프레임에 대해 가지 메트릭 연산값을 이용하여 순차적 연산을 수행하고, 슬라이딩 윈도우 블록 크기의 메모리에 연산결과를 저장한다. 역방향 메트릭 연산기인 SMU1,2는 둘로 나뉘는데 첫 번째의 SMU1은 슬라이딩 윈도우 블록에서 역순으로 더미 역방향 상태 메트릭을 계산하여 두 번째의 SMU2 로 해당 슬라이딩 윈도우 블록의 최종 연산값을 넘겨준다. 두 번째의 SMU2는 유효한 역방향 상태 메트릭 값을 연산하여 가능도비 계산을 위한 가능도비 연산기(Log-Likelihood Ratio Unit: LCU)로 입력된다.

가능도비 연산기에서는 해당 슬라이딩 윈도우 블록에 대해 SMU0로부터 계산되어 메모리에 저장된 순방향 상태 메트릭, 두 개의 SMU1,2로부터 계산된 역방향 상태 메트릭, 그리고 BMU로부터 계산된 가지 메트릭을 입력으로 가능도비 값을 계산하여 연성 출력을 발생한다.

### 3.3 BMU

가지 메트릭 연산은 식 (5) 처럼 곱과 합으로 표현되지만, 이 공식은 그림 7과 같은 3GPP 규격의 트렐리스 상태도로부터 쉽게 구현될 수 있다.

식 (5)에서 채널 신뢰도  $\frac{2}{\sigma^2}$  값은 채널 예측기에 의해 터보 디코더 입력전에 수신정보에 적용되므로, 가지 메트릭 연산은 입력 비트와 그 입력에 대한 상태천이에 의해 부호가 결정된다. 따라서 모든 가지 메트릭은 출력 '1' 과 '0' 을 '+1' 과 '-1' 로 변조시키는 경우에 식 (6) 과 같이 연산할 수 있다.

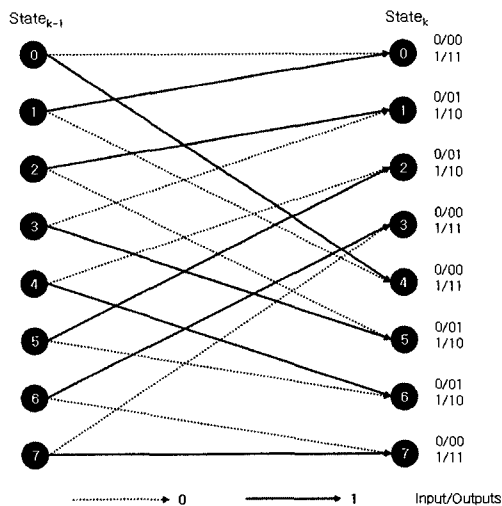


그림 7. 3GPP 터보 코드의 트렐리스 상태도  
Fig. 7. Trellis state diagram of 3GPP turbo codes.

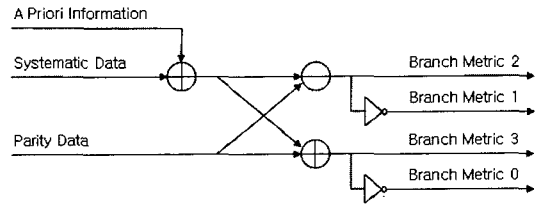


그림 8. 가지 메트릭 연산기  
Fig. 8. Branch metrics unit.

$$\begin{aligned} \text{BranchMetric0} &= - \frac{API + \text{Systematic} + \text{Parity}}{2} \\ \text{BranchMetric1} &= - \frac{API + \text{Systematic} - \text{Parity}}{2} \\ \text{BranchMetric2} &= \frac{API + \text{Systematic} - \text{Parity}}{2} \\ \text{BranchMetric3} &= \frac{API + \text{Systematic} + \text{Parity}}{2} \end{aligned} \quad (6)$$

이전 연구에서의 고정소수점 비트할당 결과를 참조하여 수신정보 입력은 5 비트, 사전정보 입력은 6 비트로 하였으며 연산된 가지 메트릭 값은 6 비트로 그림 8과 같이 설계하였다.

### 3.4 SMU

상태 메트릭 연산은 식 (3) 과 식 (4) 를 이용하여 트렐리스 상태 수 만큼 동시에 수행하여야 한다. 이때 상태 메트릭 연산을 위한 공통된 기본 연산기능인 가산 비교 선택 정규화기(Add-Compare-Select-Normalize: ACSN)을 설계하여야 한다.

그림 9는 본 논문에서 설계한 ACSN 이다.

해당 트렐리스 상태에서의 상태 메트릭을 연산하기 위하여 해당 트렐리스 상태의 가지 메트릭과 이전 트렐리스 상태의 상태 메트릭을 필요로 한다. 먼저, 현재 가지 메트릭과 이전 상태 메트릭을 각각 더하여 비교 후 큰 값과 이전 연구에서 제안한 오류 보정 함수를 이용하여 오류 보정값을 더하고 해당 트렐리스 상태에 대한 정규화 과정을 거쳐 상태 메트릭을 동시에 연산한다.

그림 10은 3GPP 규격의 터보 코드 트렐리스 상태에 적합하게 설계한 상태 메트릭 연산기의 구조이다.

이전 연구에서의 고정소수점 비트할당 결과를 참조

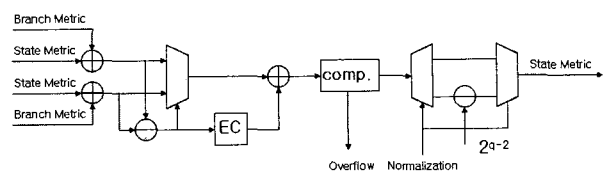


그림 9. 가산 비교 선택 정규화기  
Fig. 9. ACSN(add-compare-select-normalize) unit.

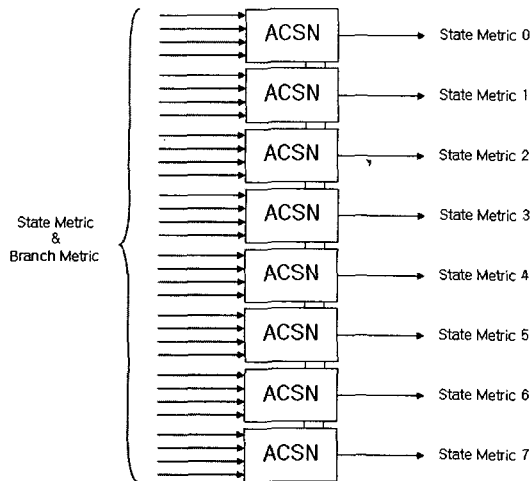


그림 10. 상태 메트릭 연산기  
Fig. 10. State metrics unit.

하여 연산된 상태 메트릭 값을 7 비트로 설계하였다.

### 3.5 LCU

가능도비 연산을 담당하는 LCU는 식 (2)로부터 E 함수에 따른  $MAX^*(A,B)$  연산을 하는 비교 선택기를 이용하여 그림 11과 같이 설계하였다.

비교 선택기는 앞의 가산 비교 선택 정규화기와 비슷한 구조를 갖지만, 덧셈기와 정규화기가 제외된 구조를 갖고, 이전 연구에서 제안한 조합회로로 구성된 오류 보정 함수를 이용한다.

그림 12의 LCU는 이전 연구에서의 로그 맵 복호 알고리즘의 성능분석에 따라  $MAX^*(A,B)$  연산구조를 채택하여 설계하였다. 가능도비 값은 해당 트렐리스 상태

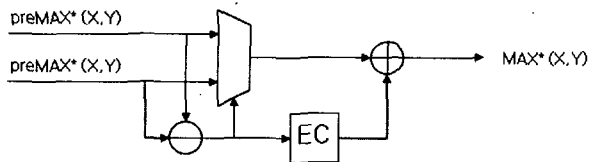


그림 11. 비교 선택기  
Fig. 11. CS(compare-select) unit.

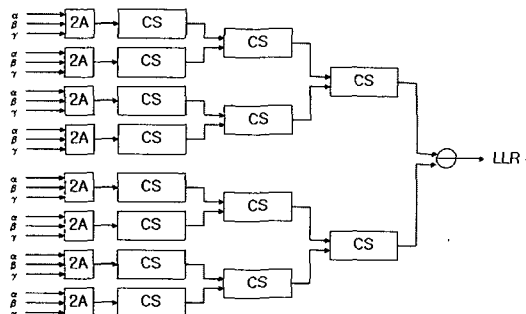


그림 12. 가능도비 연산기  
Fig. 12. Log-likelihood calculation unit.

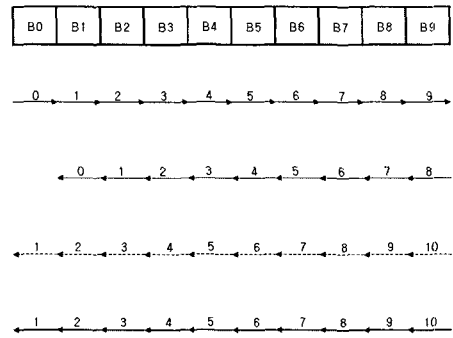


그림 13. 제안한 터보 디코더의 연산  
Fig. 13. Arithmetic of proposed turbo decoder.

에 대한 순방향 메트릭, 역방향 메트릭 그리고 가지 메트릭을 이용하여 동시에 순차적으로 연산한다.

### 3.6 제안한 구조의 복호과정

그림 13은 제안한 슬라이딩 윈도우 로그 맵 터보 디코더에서 SISO 디코더의 가지 메트릭, 상태 메트릭 그리고 가능도비의 연산 순서를 나타낸다. 각 화살표 위의 숫자는 시간이다.

SISO 디코더내의 BMU는 3개로 구성되어 SMU와 쌍을 이룬다. 각 윈도우 블록에 대하여 시간이 경과함에 따라 SMU0은 순방향으로 윈도우 블록의 끝까지 계산하여 순방향 상태 메트릭을 저장한다. 동시에 SMU1은 유효 상태 메트릭 연산의 초기값을 제공하기 위한 연산을 진행하고, SMU2는 이전 블록에서 SMU1에 의한 초기값을 이용하여 역방향 상태 메트릭을 출력한다. LCU는 두번째 윈도우 블록의 시간  $k$ 에 대하여 이미 연산되어 저장된 순방향 상태 메트릭과 시간  $k$ 에 계산되는 역방향 상태 메트릭, 가지 메트릭을 이용하여 연성 출력을 발생한다.

## IV. 분석 및 구현

### 4.1 모의실험 환경

본 논문에서 제안한 슬라이딩 윈도우 로그 맵 터보 디코더는 C 모델과 VerilogHDL 모델을 이용하여 모의 실험을 진행하였다. C 언어로 구현된 랜덤 발생기를 이용하여 발생된  $10^7$  크기를 갖는 랜덤 정보들은 3GPP 표준 터보 인코더로부터 부호화되고, 정보 '1'과 '0'을 '+1'과 '-1'로 BPSK 변조하여 평균이 0이고 분산이 No/2인 AWGN 채널을 통하여 정보가 전송된다고 가정한다.

정보는 그림 14와 같이 각각 C 모델과 VerilogHDL

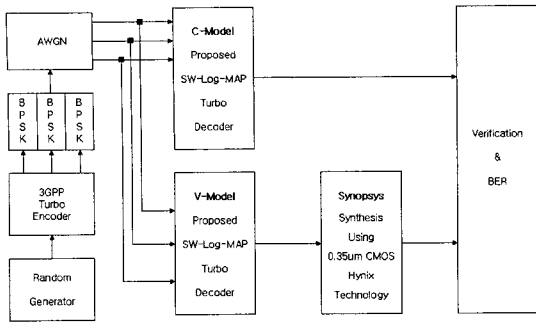


그림 14. 모의실험 환경  
Fig. 14. Simulation scheme.

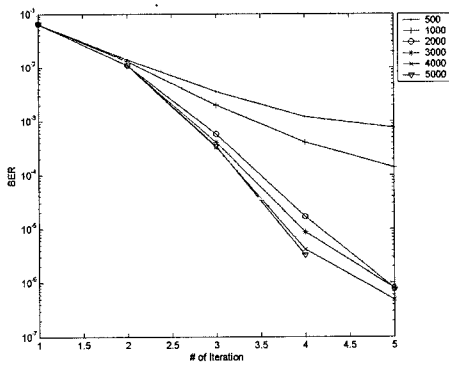


그림 15. 인터리버 및 반복 복호 변화에 따른 BER  
Fig. 15. BER for interleaver size and decoding iteration.

모델로 수신되고, 복호 과정을 거쳐 복호 성능을 비교, 확인하였다. 또한 Synopsys tool을 이용하여 0.35 $\mu$ m CMOS Hynix technology로 합성 하였고, 합성 결과를 ModelSim을 이용하여 검증하였다.

4.2 터보 디코더의 성능비교

비트 오율 성능은 Eb/No를 1dB 로 고정하고, 1번에서 5번까지의 반복 복호에 대하여 인터리버 크기를 변화하면서 평가하였다.

그림 15의 모의실험 결과를 바탕으로 반복 복호 횟수가 적을수록, 인터리버 크기에 상관없이 터보 디코더의 특징인 반복 복호를 통한 비트 오율 성능의 향상을 볼 수 없다. 하지만, 반복 복호 횟수가 많을수록 터보 디코더 내 SISO 디코더 간의 정보 공유 횟수가 많아지고, 인터리버 크기가 커질수록 부호어간의 상관관계가 작아 지므로 복호 성능은 증가한다. 모의실험 결과로부터 인터리버 크기 2000, 5번의 반복 복호에서 10<sup>-6</sup> 이하의 비트 오율을 달성하였다.

10<sup>-6</sup> 이하의 비트 오율 성능을 보이는 인터리버 크기 2000, 반복 복호 5번을 기준으로 3G W-CDMA 시스템을 위한 터보 디코더<sup>[9]</sup>와의 필요 하드웨어 자원 및 메

표 1. 하드웨어 자원 및 복호 성능 비교  
Table 1. Difference of H/W resource and decoding performance.

	[9]		Proposed	
Frequency(Hz)	42M	92M	50M	100M
Throughput(bps)	2.1M	4.6M	4.5M	9M
Gate count	-	40K	37K	52K
Technology	FPGA	0.25 $\mu$ m	0.35 $\mu$ m	
Memory	367K		296K	

모리, 복호 성능을 표 1에 나타내었다.

서로 다른 technology로 인하여 직접적인 비교는 하기 어렵지만, 약 50MHz, 약 100MHz의 동작 주파수를 기준으로 복호 속도는 2배 정도 빠르며, 약 4.5Mbps의 복호 속도에서 합성된 게이트 개수는 비슷하지만, 동작 주파수는 약 절반으로 감소하여 전력 소모를 줄일 수 있다. 또한 20% 정도로 필요 메모리 용량을 감소하여 하드웨어 자원과 전력 소모에 제약을 받는 이동통신 기기에의 적용이 용이하다. 최신 공정 기술을 적용한다면 더 나은 결과를 기대할 수 있을 것이다.

V. 결 론

로그 맵 복호 알고리즘은 오류 정정 성능을 높이기 위하여 반복 연산의 과정을 거치기 때문에 인터리버의 크기에 비례하는 많은 메모리의 필요가 치명적인 단점이 된다. 이는 적은 하드웨어 자원과 저전력으로 동작해야 하는 이동통신 기기에의 적용에 큰 장애물로 귀착될 수 있다. 로그 맵 터보 디코더의 설계 시 적은 성능 저하로 하드웨어 복잡도와 필요 메모리 부담을 줄이기 위하여 본 논문에서는 슬라이딩 윈도우 기법을 적용하여 3G W-CDMA 시스템에 적합한 로그 맵 터보 디코더를 설계하였고, 합성하였다.

모의실험과 합성 결과로부터 기존의 터보 디코더에 비하여 성능의 저하 없이 동작 주파수가 약 50% 줄어 들었고, 필요 메모리량은 약 20% 감소되었으며 최고 성능은 약 2 배 정도 되었다. 인터리버 크기 2000 에서 5 번의 반복 복호에서 복호율은 최고 9Mbps 이며, 3GPP 프로토콜의 요구 조건인 10<sup>-6</sup> 이하의 비트 오율을 충족 하였다.

## 감사의 글

저자들은 본 연구를 위하여 설계 환경을 제공하여 준 IDEC(IC Design Education Center)에 감사드립니다.

## 참고 문헌

- [1] 3GPP TS 25.212, 3rd generation partnership project technical specification group radio access network, *multiplexing and channel coding(FDD)*, (Release 4).
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo-codes(1)," in *Proc., IEEE ICC'93*, pp.1064-1070, May 1993.
- [3] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol.IT-20, pp.248-287, Mar. 1974.
- [4] 김수영, 이수인, "터보코드(Turbo Codes) 개발 동향", *주간 기술 동향*, 한국전자통신연구원, 888호, pp.1-12, 1999년.
- [5] S. Barbulescu and S. Piebrobon, "Turbo codes: a tutorial on a new class of powerful error correcting coding schemes, part 2: decoder design and performance," *IEEE Journal of Electrical and Electronics Engineering*, Australia, vol.19, no.3, pp.143-152, Sept. 1999.
- [6] H. Dawid and H. Meyr, "Real-time algorithms and VLSI architectures for soft output MAP convolutional decoding," in *Proc. Personal, Indoor, and Mobile Radio Communications*, PIMRC'95. Wireless: Merging onto the Information Superhighway, vol. 1, pp.193-197, 1995.
- [7] S. Benedetto et al., "Soft-output decoding algorithms in iterative decoding of Turbo codes," *JPL, TDA Progress Report 42-124*, Feb. 1996.
- [8] T. Ngo, I. Verbauwhede, "Turbo codes on the fixed point DSP TMS320C55x," *Signal Processing Systems, 2000 IEEE Workshop on*, pp.255-264, Oct. 2000.
- [9] X. J. Zeng, Z. L. Hong, "Design and implementation of a Turbo decoder for 3G W-CDMA systems," *Consumer Electronics, IEEE Transactions on*, Vol.48, Issue: 2, pp.284-291, May 2002.
- [10] M. El-Assal, M. Bayoumi, "A high speed architecture for MAP decoder," *Signal Processing Systems, 2002 IEEE Workshop on*, pp.69-74, Oct. 2002.
- [11] 김기환, 박태근, "Log-MAP 구조에 대한 터보 디코더의 성능분석," *제17회 신호처리합동학술대회 논문집*, vol.17, no.1, Oct. 2004.
- [12] P. Jung and J. Plechinger, "Performance of rate compatible punctured turbo-codes for mobile radio applications," *Electronics Letters*, vol.33, pp.2102-2103, Dec. 1997.
- [13] Fu-hua Huang, "Evaluation of soft output decoding for Turbo codes," *Virginia Tech*, 1997.
- [14] P. Robertson, E.Villebrun and P.Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. Int. Conf. Communications*, pp.1009-1013, June 1995.
- [15] Z. Wang, Z. Chi and K. K. Parhi, "Area-Efficient High-Speed Decoding Schemes for Turbo Decoders," *IEEE Trans. VLSI Syst.*, vol. 10, no. 6, Dec. 2002.

## 저자 소개



박태근(정회원)

1985년 연세대학교

전자공학 학사.

1988년 Syracuse Univ.

Computer 공학석사.

1993년 Syracuse Univ.

Computer 공학박사.

1991년~1993년 Coherent Research Inc. VLSI 설계 엔지니어.

1994년~1998년 현대전자 System IC 연구소 책임연구원.

1998년~현재 가톨릭대학교 정보통신전자공학부 부교수.

<주관심분야 : VLSI 설계, CAD, 병렬처리>



김기환(정회원)

2003년 가톨릭대학교 정보통신 공학 공학사.

2005년 가톨릭대학교 컴퓨터 공학과 공학석사.

2005년~현재 LG전자 이동통신 기술연구소.

<주관심분야 : ASIC, 차세대 이동통신>