

論文2003-40SP-5-11

Radix-4 트렐리스 병렬구조 및 역방향 상태천이의 제어에 의한 역추적 비터비 디코더

(Radix-4 Trellis Parallel Architecture and Trace Back Viterbi Decoder with Backward State Transition Control)

鄭 且 根 *

(Cha-Keon Cheong)

요 약

본 논문에서는 2-step 트렐리스를 하나로 통합한 Radix-4 트렐리스 병렬구조 및 역방향 상태천이의 연속적인 제어에 의한 역추적 비터비 디코더를 구현하고, 이를 초고속 무선 랜에 응용한 결과를 제시한다. Radix-4 트렐리스 병렬구조의 비터비 디코더는 throughput을 개선함과 동시에 구조가 간단하고 지연시간 및 회로의 overhead가 적은 이점이 있다. 이 특성을 기반으로, 본 논문에서는 Radix-4 트렐리스 병렬구조의 구현을 위한 가지 메트릭의 계산과 ACS의 구성, 역방향 상태천이의 연속적인 제어에 의한 역추적 복호 등으로 구성된 새로운 비터비 디코더를 제안한다. 본 제안방법의 적용으로 펀처링의 결과로 인한 가변 부호율의 복호를 통합된 하나의 디코더로 대응할 수 있으며, 부호율의 변화에 따라 별도의 부가회로나 주변제어 회로를 요구하지 않는 특성을 갖는다. 또한, 본 논문에서 제안한 역방향 상태천이의 제어에 의한 역추적 복호는 메모리 제어를 위한 별도의 회로를 추가함이 없이 ACS 사이클 타임에 정확히 동기되어 순서적인 복호를 수행할 수 있게 한다. 제안방법의 유용성을 검증하기 위해, 초고속 무선 랜 규격인 IEEE 802.11a PHY 계층의 채널부호 및 복호에 적용하고, HDL 언어로 구현한 회로의 시뮬레이션 결과를 제시한다.

Abstract

This paper describes an implementation of radix-4 trellis parallel architecture and backward state transition control trace back Viterbi decoder, and presents the application results to high speed wireless LAN. The radix-4 parallelized architecture Viterbi decoder can not only improve the throughput with simple structure, but also have small processing delay time and overhead circuit compared to M-step trellis architecture one. Based on these features, this paper addresses a novel Viterbi decoder which is composed of branch metric computation, architecture of ACS and trace back decoding by sequential control of backward state transition for the implementation of radix-4 trellis parallelized structure. With the proposed architecture, the decoding of variable code rate due to puncturing the base code can easily be implemented by the unified Viterbi decoder. Moreover, any additional circuit and/or peripheral control logic are not required in the proposed decoder architecture. The trace back decoding scheme with backward state transition control can carry out the sequential decoding according to ACS cycle clock without additional circuit for survivor memory control. In order to evaluate the usefulness, the proposed method is applied to channel CODEC of the IEEE 802.11a high speed wireless LAN, and HDL coding simulation results are presented.

Keyword : radix-4 트렐리스 병렬구조, 비터비 디코더, 역방향 상태천이, 역추적 복호, 가변 부호율, 초고속 무선 랜

* 正會員, 湖西大學校 電氣情報通信工學部
(Hoseo Univ., The School of Electrical Engineering)

接受日字:2003年8月18日, 수정완료일:2003年9月17日

I. 서론

통신기술과 반도체 기술의 획기적인 진전에 따라, 통신속도의 고속화와 통신모듈의 SOC(System on Chip)화 및 저소비 전력화가 급속히 진전되고 있다. 또한, 통신 대역폭의 높은 사용밀도로 인해 통신오류의 가능성이 증대하고 있어, 보다 강인한 채널 오류의 검출과 정정을 위한 많은 연구가 진행되어 왔다^{1, 2)}.

비터비 알고리즘(Viterbi Algorithm: VA)^{3, 4)} 채널 잡음이 포함된 수신 데이터로부터, 부호화기 상태의 모든 경로를 탐색한 후 부호어와 유사성(likelihood)이 적은 경로를 제거하고 가장 높은 유사성을 갖는 경로의 상태를 선택함으로써 송신정보를 검출하는 최우추정(Maximum Likelihood: ML) 복호 알고리즘이다. 따라서, VA에 의한 디코더는 최적의 경로탐색을 위한 반복 과정과 고유의 제한구조로 인해, 상태수가 증가함에 따라 계산량과 지연시간이 증가하게 되고, 이는 초고속 통신 및 SOC화의 구현을 제한하는 핵심 요소의 하나가 되고 있다.

이로 인해, VA에 관한 주요 연구는

- 단위시간당 처리되는 데이터 비트수(이후, throughput 이라 함)의 향상
- SOC화를 위한 칩 면적의 소형화 및 소비전력의 저감을 위한 구조의 간략화
- 다양한 통신속도에 대응할 수 있는 시스템 구성의 높은 유연성

등을 구현할 수 있는 구조의 개발에 집중되고 있다. VA에 의한 복호 레이트는 제한구조를 갖는 Add-Compare-Select (ACS)의 반복연산과 최적경로의 탐색에 요구되는 연속적인 생존 메모리(Survivor Memory: SM)의 제어에 의해 제한된다. 특히, ACS부는 비터비 디코더의 구현에서 데이터 처리의 고속화 및 VLSI 칩의 효율적인 공간사용(area-efficient)의 가장 큰 장애물로 작용하므로, 이를 개선하기 위한 연구가 활발히 진행되어 왔다⁷⁻¹⁷⁾.

Throughput의 향상과 area-efficient 구조의 비터비 디코더를 구현하기 위한 가장 일반적인 방법은 시스템 배열 구성에 기반한 병렬 또는 파이프라인 구조의 디코더를 구성하는 것이다. 즉, Fettweis와 Meyr¹¹⁾ 등은 M-스텝(step) 트렐리스의 대수적 선형 병렬구조의 ACS와 시스템 배열구조를 제안하고, 데이터 처리의

고속화 및 VLSI 칩의 효율적인 비터비 디코더 구현이 되도록 하고 있다. Gulak과 Shwedyk¹²⁾ 등은 de Bruijn 형상의 트렐리스를 갖는 상태들의 병렬 구현은 Shuffle-Exchange 그래프로 정의되는 상호연결 망이 되는 것을 확인하고, VLSI 그리드 모델을 기반으로 wiring area를 줄이기 위한 SE 그래프 링 구조의 최적 배치방법을 제안했다. 또한, Gulak 및 Kailath⁷⁾ 등은 de Bruijn 그래프 구조를 갖는 트렐리스에 대한 3종류의 국부적으로 연결되는 프로세서 배열 방법을 제안하고, 각 ACS간의 데이터 교환이 인접한 것들만의 통신이 되도록 제한함으로써 wiring area를 더욱더 줄일 수 있는 방법을 기술하고 있다. C. B. Shung et al¹⁰⁾ 등은 비터비 알고리즘의 area-efficient 구조를 위해, 여러개의 트렐리스 상태에 대한 ACS를 하나의 ACS로 공유시켜 처리하는 방법을 제안하고 있다. 트렐리스도의 N 상태에 대한 각각의 ACS를 P개의 ACS로 분할하고, 스케줄링 및 매핑을 위한 체계적인 기법을 제안하고, 순서적 순환망에 의한 경로 매트릭의 저장으로 경로 매트릭의 정렬을 위한 기법을 소개하고 있다. 또한, P. J. Black 및 T. Meng^{18, 13, 17)} 등은 2-step 트렐리스 radix-4 병렬 ACS 구조와 radix-16 트렐리스의 역추적에 의한 복호방법을 기술하고, 이를 구축장 $K=6$, 부호율 $R=1/2$ 인 길쌈부호기의 복호에 적용함으로써 140Mbps의 비터비 디코더 구현이 가능함을 보이고 있다. 이들의 연구결과로부터 ACS부의 고속화 및 area-efficient를 위한 이론적인 근거는 어느정도 확립된 것으로 생각할 수 있다.

채널의 다양한 통신속도에 대응하는 일반적인 방법으로 Rate Compatible Punctured Code (RCPC)¹²⁶⁾가 널리 사용되고 있다. RCPC는 전송하기 전의 기본 부호(base code)로부터 펄처링 패턴에 따라 부호어를 제거해서 전송 부호율을 조절하는 것으로, 기본 부호의 디코더 구조를 변경함이 없이 가변 부호율의 복호를 수행할 수 있다. VA에 의한 복호에서는 제거된 부호어의 정보가 최적 경로 결정에 영향을 미치지 않도록 해야 된다. 이를 위한 방법으로 누락된 부호어에 대응하는 가지 매트릭을 상수 또는 제로값으로 하거나 펄처링된 위치에 제로값의 부호어를 채워넣는 방법이 주로 사용된다. 이 경우, 매트릭의 실제 연산에서 일정한 제약은 받게 되고, 복호깊이를 부호율에 따라 가변적으로 제어해야 하는 문제가 있다.

본 논문에서는 기 개발된 M-스텝 트렐리스를 하나

의 트렐리스로 통합한 M-스텝 비터비 프로세서^[11] 기반으로, 2-스텝 트렐리스를 하나로 통합한 radix-4 트렐리스 병렬구조의 비터비 디코더^[13] 구현하기 위한 새로운 구조를 제안하고, 이를 초고속 무선 랜의^[2] 채널 코덱에 적용한 결과를 제시한다. 기존의 M-스텝 트렐리스 병렬구조는 ACS의 고속 연산과 구현을 위한 체계적인 방법을 제시하고 있으나, throughput의 개선에 비해 시스템 구성의 복잡성이 크게 증가하는 문제가 있다. 또한, radix-4 트렐리스 병렬구조의 VA는 일반적인 M-스텝 트렐리스 구조의 VA에 비해 구조가 간단하고 지연시간 및 회로의 overhead가 적은 이점을 갖는다^[13]. 전체 비터비 디코더의 throughput을 개선시키기 위해서는 ACS의 고속화뿐만 아니라, 생존 메모리의 제어에 의한 복호과정도 중요하다. 이를 위해, [13]에서는 radix-4 트렐리스 구조에서 radix-2 구조의 역추적을 수행한 후, radix-16의 역추적을 수행하는 2 단계의 역추적 복호 구성을 제안하고 있으나, 트렐리스 상태수가 증가하게 되면 역추적 알고리즘이 방대해지는 문제가 있다. 따라서, 본 논문에서는 이를 해결하기 위한 방법으로 역방향 상태천이의 연속적인 제어로 상태수의 증가나 사이클 타임에 관계없이 단일 역추적 알고리즘으로 자동 복호하는 새로운 기법을 제안한다.

본 논문에서 제안하는 비터비 디코더의 구성은 높은 throughput의 달성이 가능하다. 즉, 부호율 1/2의 기본 부호율 및 가변 부호율을 위한 RCPC 코더의 복호를 동일 구조의 복호기로 처리할 수 있다. 또한, 비트의 결정비트를 하나의 ACS 사이클 타임에서 동시처리 가능한 역추적 복호와 메모리 제어를 위한 별도의 주변회로를 요구하지 않는 특징이 있다. 본 제안 기법의 유용성을 검증하기 위해, 초고속 무선 랜 규격인 IEEE 802.11a PHY 계층의 채널부호 및 복호에 적용한다. IEEE 802.11a PHY 계층의 부호율은 1/2, 2/3 및 3/4의 3종류이고, 최대 데이터 전송율 54Mbps를 달성하는 구속장 $K=7$ 의 길쌈부호기가 규격화되어 있다. HDL 언어로 구현한 회로설계 시뮬레이션을 수행하고, 그 결과를 제시한다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 본 논문의 제안 알고리즘의 설명에 필요한 내용을 중심으로 VA를 간략히 기술한다. 본 논문에서의 제안 방법은 3절에서 기술된다. M-step 트렐리스 구조의 비터비 디코더의 핵심적인 내용을 간단히 언급한 후, radix-4 트렐리스 비터비 디코더의 구조, 가지 메트릭의 계산,

ACS 처리 블록의 구성과 역방향 상태천이의 순서적 제어에 의한 역추적 기법 등을 상세히 기술한다. 다음으로, 제안방법의 구체적인 시스템에의 응용 예로 초고속 무선 랜의 규격과 실험조건 및 시뮬레이션 결과 등을 4절에서 제시하고, 결론을 5절에서 기술한다.

II. 비터비 알고리즘

비터비 디코딩은 채널잡음이 포함된 수신 데이터로부터, 부호화기의 모든 상태에 대해 유사성이 가장 높은 상태천이를 추정해서 데이터를 복호하는 방법이다. 이와 같은 VA에 의한 복호과정은 트렐리스도를 사용해서 간결하게 기술할 수 있다. 즉, VA는 주어진 트렐리스도의 각 상태에서 최소의 가중치를 갖는 경로를 탐색하고, 이를 송신 데이터와 유사성이 가장 높은 상태 시퀀스에 해당하는 생존경로로 결정하고 복호하는 것이다^[5,6].

채널 잡음이 부가된 수신 데이터 열 \mathbf{r}_k 로부터 부호기의 입력정보 u_k 를 최적복호하는 VA는 송신 데이터 \mathbf{v}_k 와의 유사성에 따라 다음과 같이 계산된 메트릭값이 최소인 경로를 선택한다.

$$\lambda_{j,k} = \|\mathbf{r}_k - \mathbf{v}_k\|_{(i,j)} \quad (1)$$

$\lambda_{j,k}$ 는 사이클 타임 k 의 i 번째 상태 $s_{i,k}$ 에서 사이클 타임 $k+1$ 의 j 번째 상태 $s_{j,k+1}$ 로 천이되는 경우의 가지 메트릭 또는 상태천이 메트릭으로, 이는 상태 $s_{j,k+1}$ 에 입력되는 i 번째 가지의 메트릭에 해당된다. 비터비 디코더에서 가지 메트릭 $\lambda_{j,k}$ 의 연산을 수행하는 부분을 BMU(Branch Metric Unit)라고 하고, 실제 계산에서는 유클리드(Euclidean) 거리나 해밍(Hamming) 거리 등이 연판정(soft decision) 또는 경판정(hard decision)의 적용에 따라 구분되어 사용된다.

입력의 복호 사이클 시간에서 특정 상태에 이르기까지의 생존경로를 따라 누적된 가지 메트릭을 경로 메트릭(path metric) 또는 상태 메트릭(state metric)이라 한다. <그림 1>은 구속장 $K=3$ 부호율 $R=1/2$ 의 트렐리스도를 나타낸 것이다. 이 그림에서 사이클 타임 $k+1$ 의 j 번째 경로 메트릭을 $\gamma_{j,k+1}$ 이라 하면, 사이클 타임 k 의 i 번째 상태와 i 번째 상태에서부터 천이되는 경우, $\gamma_{j,k+1}$ 는

$$\gamma_{i,k+1} = \min \{ \gamma_{t,k} + \lambda_{\mu,k}, \gamma_{i,k} + \lambda_{\mu,k} \} \quad (2)$$

$(i, j, t \in \{0, 1, 2, 3\})$

와 같이 사이클 타임 k 에서의 경로 메트릭과 이 상태에 이르게 하는 가지 메트릭의 합이 최소가 되는 경로 및 가지 메트릭으로 주어진다. 매 사이클 시간마다 각 상태에 도달하는 모든 경로 중에서 최소의 경로 메트릭이 되게 하는 경로를 생존경로라 한다. 일반적으로 각 상태에 대한 최적경로의 탐색은 식 (2)와 같은 ACS 순환에 따라 매 사이클 타임의 각 상태에 대한 경로 메트릭의 계산으로 동적 프로그래밍의 방법을 사용해서 수행된다. 이와 같은 ACS를 위한 데이터 처리과정은 본질적으로 비선형이고 데이터에 좌우되는 성질을 갖는다. 각 상태에 대해 경로 메트릭의 연산과 생존경로의 결정 및 이에 관한 정보를 출력하는 부분을 ACS 부라 한다.

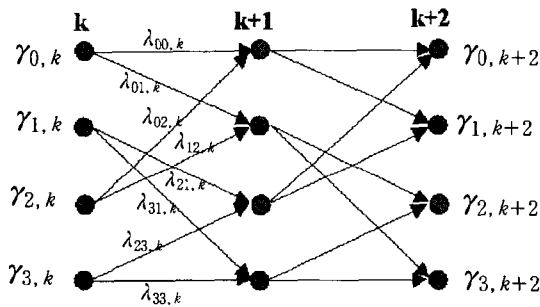


그림 1. $N=4$ 인 트렐리스도
Fig. 1. The trellis diagram for $N=4$.

VA에 의한 복호는 생존경로에 관한 정보, 즉 결정비트를 별도의 메모리에 저장하고, 매 ACS 사이클마다 메모리에 저장되는 값을 교환하거나 역추적해서 최소 메트릭의 생존경로를 추정함으로써 이루어진다. 결정비트 저장에 사용되는 메모리를 생존 메모리라 하고, SM 제어의 방법은 생존경로의 저장에 사용되는 메모리의 구성방법에 따라 레지스터 교환 (Register Exchange: RE) 방식과 역추적 (Trace-Back: TB) 방식으로 분류된다^[19-21]. RE 방식은 레지스터를 사용해서 각 상태의 생존경로에 관한 정보를 저장하고, 각 복호 사이클마다 저장된 전체 레지스터의 정보를 갱신해서 복호하는 방법으로 복호 지연시간, 메모리 사이즈, regularity 등의 파라미터들이 복호에 영향을 크게 미치는 시스템에 주로 사용되고 있다. RE 방식은 SM 제어 구조가 비교적

간단해서 짧은 구속장을 갖는 부호기의 복호에 적용되고 있으나, 복호과정의 매 사이클마다, 전체 레지스터의 값을 교환해야 하므로 광대역의 메모리 액세스와 메모리 제어를 위한 주변회로가 복잡해지고, 이는 소비전력의 증가와 VLSI 구현시 칩 사이즈를 증가시키는 문제가 있다.

TB 방식은 일정한 크기에 해당하는 메모리에 각 상태의 생존경로에 관한 정보를 저장한 후, 역추적의 방법으로 ML 경로를 탐색해서 복호하는 것으로, 구속장이 비교적 크고, 높은 코딩 이득이 요구되는 시스템에 주로 사용된다. 또한, TB 방식은 온 칩 레지스터 대신에 RAM과 같은 외부 메모리를 사용한 구현이 가능하며 전력소모가 RE 방식에 비해 적은 이점을 갖는다. 높은 throughput과 칩 구현의 효율을 향상시키기 위해 다양한 방법들이^[19, 20] 연구되어, RE 방법보다 널리 사용되고 있으나, TB 방법은 그 특성상 사용되는 메모리 사이즈가 RE 방식에 비해 방대해지고, 생존경로에 관한 정보의 저장과 역추적 및 복호 등의 과정이 동시에 수행되어야 하므로, 메모리 액세스를 위한 제어가 복잡하게 될 뿐만 아니라 복호 지연시간이 길어지는 근본적인 제약을 동반하고 있다.

III. Radix-4 트렐리스 비터비 디코더

높은 throughput의 달성과 VLSI 구성시 칩 사이즈 및 전력소모를 줄이기 위해서는, 비선형 데이터 종속성의 특성을 갖는 순환구조의 ACS를 변경시키는 것이다. 이 때, VLSI 구현에 적합한 구성은 각 처리 프로세서간 데이터가 지역적으로 분산되어 병렬로 교환되고, 높은 규칙성과 칩 공간의 면적이 적게 되도록 함과 동시에 회로구성의 복잡성이 throughput의 개선속도에 비례하는 성질을 갖도록 하는 것이다^[7]. 이를 위한 대표적인 방법으로 M-스텝 트렐리스를 하나의 트렐리스로 통합해서 경로 메트릭을 선형 대수함수의 형태로 연산하는 것이다.

1. M-스텝 트렐리스 Radix- 2^M 비터비 디코더^[11]

<그림 1>에 나타난 트렐리스도의 경로 메트릭은 Semiring 대수 연산으로 다음과 같은 행렬의 형태로 기술할 수 있다. 사이클 타임 $k+1$ 에서의 모든 상태의 경로 메트릭 Γ_{k+1} 은

$$\Gamma_{k+1} = A_k \otimes \Gamma_k \quad (3)$$

가 된다. 여기서, \otimes 은 덧셈의 연산을 나타내는 기호이고, $\Gamma_k = [\gamma_{0,k}, \gamma_{1,k}, \gamma_{2,k}, \gamma_{3,k}]^T$ (T 는 전치(transpose))의 사이클 타임 k 에서의 모든 상태의 경로 메트릭을 벡터로 나타낸 것이다. 또한, Λ_k 는 사이클 타임 k 에서 $k+1$ 에 이르는 가지 메트릭 $\lambda_{ji,k}$ 로 구성된 $N \times N$ 상태전이 행렬로서,

$$\Lambda_k = \begin{bmatrix} \lambda_{00,k}, & Q, & \lambda_{02,k}, & Q \\ \lambda_{10,k}, & Q, & \lambda_{12,k}, & Q \\ Q, & \lambda_{21,k}, & Q, & \lambda_{23,k} \\ Q, & \lambda_{31,k}, & Q, & \lambda_{33,k} \end{bmatrix} \quad (4)$$

으로 정의된다. 이 때, $Q = (-\infty)$ 또는 (∞) 의 값으로 상태전이에서 서로 연결되지 않는 상태들을 나타낸 것으로 생존경로의 선택에 포함되지 않도록 하는 것을 의미한다. 상태전이 행렬과 경로 메트릭 벡터간의 연산에서 덧셈은 기호 \oplus 로 표시되고, 이는 생존경로를 선택하는 과정을 의미한다.

이상의 인접한 두 트렐리스간의 경로 메트릭은 M-스텝 트렐리스 ACS 순환식으로 간단히 확장시킬 수 있다. 즉, 사이클 타임 $k+M$ 에서의 모든 상태에 대한 경로 메트릭 벡터 Γ_{k+M} 은

$$\begin{aligned} \Gamma_{k+M} &= {}_M\Lambda_k \otimes \Gamma_k \\ &= (\Lambda_{k+M-1} \otimes \Lambda_{k+M-2} \otimes \dots \otimes \Lambda_k) \otimes \Gamma_k \end{aligned} \quad (5)$$

으로 주어지고, ${}_M\Lambda_k$ 은 사이클 타임 k 에서 $k+M$ 까지의 각 상태에 대한 모든 경로를 나타내는 $N \times N$ 상태전이 메트릭의 행렬이다. 이와 같은 M-스텝 트렐리스는 기본적인 1-스텝 트렐리스 M개를 하나의 M-스텝 트렐리스로 결합시킨 것에 다름 아니다.

ACS 순환 연산의 본질적인 비선형 데이터 중속성은, 파이프라인이나 look-ahead 처리와 같은 병렬화 기법의 직접 적용을 어렵게 한다. 이에 반해, M-스텝 ACS 순환식 (5)는 시스틀릭 배열 구조의 사용으로 feedforward 구조인 선형 대수 연산의 적용을 가능케 해, 다양한 ACS 구조를 구현할 수 있게 한다^[6]. 병렬구조 상태를 갖는 비터비 디코더의 반복시간은 하나의 상태에 대한 ACS의 반복시간에 제한되므로, M-스텝 트렐리스 병렬구조의 비터비 디코더는 그 throughput을 M배 증가시킬 수 있다. 이와 같은 M개 트렐리스의 통합에서 모든 상태간의 가지 메트릭 및 경로 메트릭이 1대1 대응관계가 성립하므로 디코더의 성능에는 영향을 미치지 않는다.

그러나, M-스텝 ACS 순환식은 $N \times N$ 상태전이 행렬 Λ_k 의 연산에 요구되는 계산량이 M 값의 증가에 따라 폭발적으로 증가하게 되는 문제가 있다. 또한, M-스텝 트렐리스 병렬구조의 ACS에서 M값에 따라 증가하는 회로의 복잡성과 부가적인 지연시간 및 overhead로 인해, 실제 시스템의 구현에서 throughput이 M배로 개선되지 않는다.

2. 2-스텝 트렐리스 Radix-4 병렬구조 비터비 디코더
2-스텝 트렐리스 radix-4 병렬구조 ACS는 순환식의 산술적 최적화 여지가 많아 실제 디코더 구현에서 area-efficiency를 유지하면서 throughput을 개선시킬 수 있다^[13]. 이는 가지 메트릭의 계산을 위한 연속적인 덧셈을 효율적인 병렬구조로 구현할 수 있고, 하나의 ACS 사이클 타임으로 2비트의 결정비트 발생과 처리가 가능하기 때문이다. 또한, 이 구조는 부호율 $R=1/2, 2/3, 3/4$ 등과 같은 RCPC에 의한 가변 전송율에 효율적인 통합구조의 비터비 디코더를 구현할 수 있다. 예로, <그림 1>의 부호율 $R=1/2$ 인 트렐리스도에서 부호율 $R=2/3$ 를 위한 부호기의 트렐리스도를 나타낸 것이 <그림 2(a)>이다. 이 그림에서 d는 평처링에 의해 제거되는 부호어를 나타내는 것이고, 경로 메트릭의 계산 및 생존경로의 선택에 평처링된 비트 d의 영향을 받지 않도록 해야 된다. <그림 2(b)>는 2-스텝 트렐리스 radix-4 병렬구조의 트렐리스

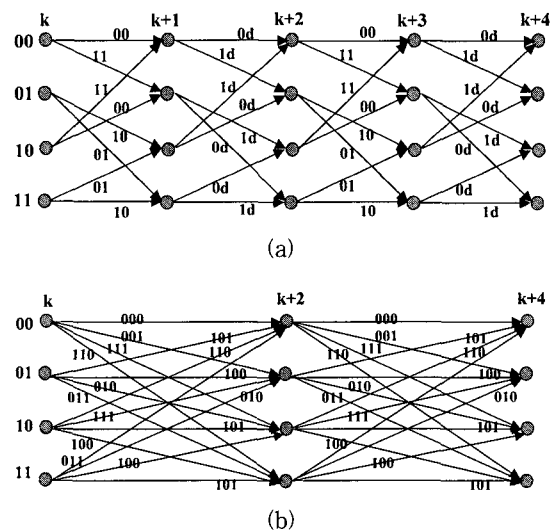


그림 2. 부호율 $R=2/3$ 에 대한 (a) radix-2 트렐리스도와 (b) radix-4 트렐리스도
Fig. 2. (a) Radix-2 trellis diagram and (b) radix-4 trellis diagram for the coding rate $R=2/3$.

도를 나타낸 것이다. <그림 2(b)>로부터 radix-4 트렐리스 병렬구조는 가지 메트릭의 계산과 생존경로의 선택에 3비트의 부호어를 동시에 고려함으로써, 평처링 비트의 영향을 제거하고 throughput을 2배 개선시킬 수 있다.

<그림 1>의 $N=4$ 인 트렐리스도에 대한 사이클 타임 $k+2$ 에서의 2-스텝 트렐리스 radix-4 구조의 상태들에 대한 경로 메트릭 Γ_{k+2} 는 식 (5)로부터

$$\begin{aligned} \Gamma_{k+2} &= (\Lambda_{k+1} \otimes \Lambda_k) \otimes \Gamma_k \\ &= {}_2\Lambda_k \otimes \Gamma_k \end{aligned} \quad (6)$$

이고 4×4 상태천이 메트릭 행렬 ${}_2\Lambda_k$ 는 식 (4)의 반복 적용으로

$$\begin{aligned} {}_2\Lambda_k &= \begin{bmatrix} \lambda_k^{(00)}, & \lambda_k^{(01)}, & \lambda_k^{(02)}, & \lambda_k^{(03)} \\ \lambda_k^{(10)}, & \lambda_k^{(11)}, & \lambda_k^{(12)}, & \lambda_k^{(13)} \\ \lambda_k^{(20)}, & \lambda_k^{(21)}, & \lambda_k^{(22)}, & \lambda_k^{(23)} \\ \lambda_k^{(30)}, & \lambda_k^{(31)}, & \lambda_k^{(32)}, & \lambda_k^{(33)} \end{bmatrix} \\ &= \begin{bmatrix} \lambda_{00,k+1} \otimes \lambda_{00,k}, & \lambda_{02,k+1} \otimes \lambda_{21,k}, & \lambda_{00,k+1} \otimes \lambda_{02,k}, \\ \lambda_{10,k+1} \otimes \lambda_{00,k}, & \lambda_{12,k+1} \otimes \lambda_{21,k}, & \lambda_{10,k+1} \otimes \lambda_{02,k}, \\ \lambda_{21,k+1} \otimes \lambda_{10,k}, & \lambda_{23,k+1} \otimes \lambda_{31,k}, & \lambda_{21,k+1} \otimes \lambda_{12,k}, \\ \lambda_{31,k+1} \otimes \lambda_{10,k}, & \lambda_{33,k+1} \otimes \lambda_{31,k}, & \lambda_{31,k+1} \otimes \lambda_{12,k}, \end{bmatrix} \\ &\quad \begin{bmatrix} \lambda_{02,k+1} \otimes \lambda_{23,k} \\ \lambda_{12,k+1} \otimes \lambda_{23,k} \\ \lambda_{23,k+1} \otimes \lambda_{33,k} \\ \lambda_{33,k+1} \otimes \lambda_{33,k} \end{bmatrix} \end{aligned} \quad (7)$$

와 같이 전개된다. 식 (7)에서 2-스텝 트렐리스 각 상태간의 천이를 나타내는 가지 메트릭은 1-스텝 트렐리스 가지 메트릭의 산술합(\otimes)으로 주어진다. 따라서, 단순한 트렐리스 통합이 가지 메트릭이나 경로 메트릭의 결과 미치는 영향은 없다. 이와 같은 관계는 임의의 상태수를 갖는 트렐리스도에 대해서도 동일하다.

일반적인 길쌈부호기의 경우, 트렐리스도는 고속 푸리에 변환(FFT)에서와 같이 de Bruijn 형태 또는 Shuffle-Exchange 구조를 갖는다^[6]. 따라서, 구속장 K 인 부호기는 $N=2^{K-1}$ 의 상태와 $N/2$ 에 해당하는 2^{K-2} 조의 나비구조를 갖는다. <그림 3>은 구속장 K 인 일반적인 radix-2 트렐리스의 2-스텝 트렐리스를 하나로 통합한 radix-4 병렬구조의 한쌍의 트렐리스도를 나타낸 것이다. 이 그림에서 γ_k^{pqX} 는 사이클 타임 k , 상태 (pqX) 의 경로 메트릭이고, γ_{k+2}^{Xab} 는 사이클 타임 $k+2$, 상태 (Xab) 의 경로 메트릭을 나타낸다. 또한, X 는 상태천이에서 변화되지 않는 비트들의 조합을 나타낸다. 사이클 타임 $k+2$ 의 각 상태에서 생존경로의

결정에 관한 정보는 사이클 타임 k 에서의 각 상태를 나타내는 비트들 중 MSB 2비트 (p, q) 의 값과 일치하는 것을 볼 수 있다.

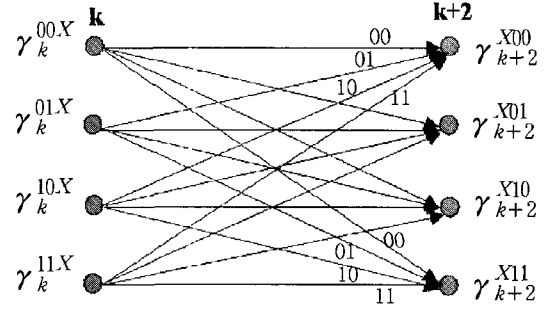


그림 3. Radix-4 트렐리스도

Fig. 3. The radix-4 trellis diagram.

3. 가지 메트릭의 계산

백색 가우스 잡음이 부가(AWGN)된 수신 데이터 \mathbf{r}_k 로부터 부호기의 입력정보를 최적복호하기 위해, radix-2 트렐리스도의 가지 메트릭은 ML 복호로부터 다음과 같은 함수로 정의할 수 있다.

$$\lambda_{j,k} = \max_{\mathbf{v}} \log p(\mathbf{r}_k | \mathbf{v}_k) = \min_{\mathbf{v}} \sum_{(i,j)} |\mathbf{r}_{n,k} - \mathbf{v}_k|_{(i,j)} \quad (8)$$

여기서 $p(\cdot | \cdot)$ 는 조건부 확률이다. 따라서, de Bruijn 그래프 트렐리스의 경우, 유클리드 거리 메트릭에 의한 가지 메트릭은

$$\begin{aligned} \lambda_{j,k} &= \min_{\mathbf{v}} \sum_{n=0}^{N-1} |\mathbf{r}_{n,k} - \mathbf{v}_{n,k}|_{(i,j)}^2 \\ &= \min_{\mathbf{v}} \sum_{n=0}^{N-1} |\mathbf{r}_{n,k}^2 + \mathbf{v}_{n,k}^2 - 2\mathbf{r}_{n,k} \mathbf{v}_{n,k}|_{(i,j)} \end{aligned} \quad (9)$$

이 된다. 한편 각 상태에 입력되는 모든 가지에 대해, $\mathbf{r}_{n,k}^2 + \mathbf{v}_{n,k}^2$ 은 동일하므로 이를 상수로 취급하거나 가지 메트릭의 계산에서 제거해도 최적 상태의 선택에는 아무런 영향을 미치지 않는다. 그러므로 가지 메트릭 $\lambda_{j,k}$ 는 다음과 같이 간략화된 식으로 나타낼 수 있다.

$$\lambda_{j,k} = \max_{\mathbf{v}} \left\{ \sum_{n=0}^{N-1} \mathbf{r}_{n,k} \mathbf{v}_{n,k} \right\}_{(i,j)} \quad (10)$$

BPSK 변조로 가정하고, 부호어 $\mathbf{v}_{n,k}$ 를 다음으로 변환된 값을 가정한다.

$$\begin{aligned} \mathbf{v}_{n,k} = 0 \text{ 이면 } \mathbf{v}_{n,k} &= +1 \\ \mathbf{v}_{n,k} = 1 \text{ 이면 } \mathbf{v}_{n,k} &= -1 \end{aligned} \quad (11)$$

또한, 식 (10)에 의한 가지 메트릭의 계산에서 곱셈의 연산이 이루어지지 않도록 하기 위해, 식 (10)을 다음과 같이 수정해서 사용한다.

$$\lambda_{ji,k} = \max_v \left\{ \sum_{j=0}^1 \frac{1}{2} (r_{n,k} v_{n,k} + \text{abs}(r_{n,k}))_{(i,j)} \right\} \quad (12)$$

식 (12)에 식 (11)로 주어진 부호어 $v_{n,k}$ 의 값을 대입하고 정리하면, 가지 메트릭 $\lambda_{ji,k}$ 는

$$\lambda_{ji,k} = \sum_{j=0}^1 |r_{n,k}| \quad \text{if } \text{sign}(r_{n,k}) = \text{sign}(v_{n,k}) \quad (13)$$

와 같이 간단한 연산으로 구해지는 것을 유도할 수 있다. 한편, radix-2 트렐리스도에서 한쌍의 나비구조의 부호어는 보수관계를 갖는 2종류의 부호어만이 존재한다. 예로 <그림 2(a)>의 사이클 타임 k 의 상태 (00)와 (10), 사이클 타임 $k+1$ 의 상태 (00)와 (01)은 한쌍의 나비구조를 형성한다. 이 때, 생성되는 부호어는 00 및 11이 되는 것을 알 수 있다. 따라서, 한쌍의 나비구조의 트렐리스도에서 상이한 가지 메트릭은 2개이고, 식 (13)과 다른 하나의 가지 메트릭은

$$\lambda_{ji,k} = \sum_{j=0}^1 |r_{n,k}| \quad \text{if } \text{sign}(r_{n,k}) = \text{sign}(\overline{v_{n,k}}) \quad (14)$$

으로 주어진다. 단, $\overline{v_{n,k}}$ 는 $v_{n,k}$ 의 보수(complement)이다.

식 (7)에서 알 수 있는 바와 같이 radix-4 트렐리스의 각 가지에 대한 메트릭은 기본 radix-2 트렐리스에 대한 가지 메트릭을 연속적으로 적용하면 간단히 구할 수 있다. 즉, 사이클 타임 k 에서의 가지 메트릭을 $\lambda_k(v_{1,k}, v_{0,k})$, 사이클 타임 $k+1$ 에서의 가지 메트릭 $\lambda_{k+1}(v_{1,k+1}, v_{0,k+1})$ 이라 하면, 부호어 (v_1, v_0) 에 따라 4×4 의 16개 가지 메트릭의 계산을 병렬구조로 처리할 수 있다. 즉, 사이클 타임 k 에서 사이클 타임 $k+2$ 까지의 radix-4 트렐리스 각 상태에 입력되는 가지 메트릭은

$$\lambda_k(v_{1,k}, v_{0,k}) \times \lambda_{k+1}(v_{1,k+1}, v_{0,k+1}) \quad (15)$$

의 연산으로 계산된다. 부호율 $R=1/2$ 인 radix-4 트렐리스의 가지 메트릭을 계산하는 병렬구조를^[13] 나타

낸 것이 <그림 4>이다.

평처링에 의한 1/2 이상의 부호율을 갖는 경우, 평처링된 부호어의 메트릭은 가지 메트릭의 계산에 포함되지 않도록 해야 한다. 즉, $R=2/3$ 이나 $R=3/4$ 과 같은 부호율의 경우, 평처링된 부호어를 0으로 취급하고, 이를 식 (15)에 대입함으로써 $R=1/2$ 인 경우의 구조를 변경하지 않고 사용할 수 있다. 예로 $R=2/3$ 의 경우, <그림 2>로부터 가지 메트릭은 $\lambda_k(v_{1,k}, v_{0,k}) \times \lambda_{k+1}(0, v_{0,k+1})$ 의 연산에 의해 8개의 가지 메트릭이 계산되고, 하나의 가지 메트릭이 다른 상태의 메트릭에 2번씩 사용되어 전체 16개의 가지 메트릭이 출력된다.

	$\lambda_{k+1}(00)$	$\lambda_{k+1}(01)$	$\lambda_{k+1}(10)$	$\lambda_{k+1}(11)$
$\lambda_k(00)$	$\lambda_{k+2}^{(00)}$	$\lambda_{k+2}^{(01)}$	$\lambda_{k+2}^{(02)}$	$\lambda_{k+2}^{(03)}$
$\lambda_k(01)$	$\lambda_{k+2}^{(10)}$	$\lambda_{k+2}^{(11)}$	$\lambda_{k+2}^{(12)}$	$\lambda_{k+2}^{(13)}$
$\lambda_k(10)$	$\lambda_{k+2}^{(20)}$	$\lambda_{k+2}^{(21)}$	$\lambda_{k+2}^{(22)}$	$\lambda_{k+2}^{(23)}$
$\lambda_k(11)$	$\lambda_{k+2}^{(30)}$	$\lambda_{k+2}^{(31)}$	$\lambda_{k+2}^{(32)}$	$\lambda_{k+2}^{(33)}$

그림 4. Radix-4 가지 메트릭의 계산을 위한 병렬구조. Fig. 4. The parallel structure for the computation of radix-4 branch metric.

이상과 같이 가지 메트릭을 계산하는 BMU는 2 계층, 즉, 1-스텝 트렐리스의 가지 메트릭을 계산하는 계층과 이들을 통합하여 radix-4 트렐리스의 가지 메트릭을 계산하는 계층으로 구성된다. 따라서, 사이클 타임 k 에서 $k+1$ 의 상태천이의 가지 메트릭과 사이클 타임 $k+1$ 에서 $k+2$ 로 상태천이의 가지 메트릭을 병렬로 먼저 계산하고, 이들 radix-2 가지 메트릭으로부터 radix-4 가지 메트릭을 식 (7)과 같이 계산한다. <그림 5>는 이들의 구성을 블록도로 나타낸 것으로 다양한 채널 부호율에 대한 가지 메트릭을 하나의 통합된 구조로 계산할 수 있게 한다. 이 그림에서 데이터 처리는, 먼저 클럭 주파수 f Hz의 수신 데이터 열은 부호율 R 에 따라 Rf Hz로 변환되고, 직병렬 변환기에 의해 $Rf/2$ Hz의 주차수를 갖는 4개의 데이터 열로 병렬 출력된다.

다음으로 이들 4개의 데이터 심볼로부터 radix-2 구조의 가지 메트릭의 계산과 radix-4 가지 메트릭의 계산이 순서적으로 수행되어 ACSU로 입력되는 것이다.

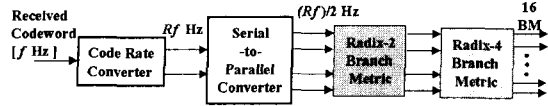


그림 5. 가지 메트릭의 계산을 위한 회로 구성의 블록도
Fig. 5. The block diagram for the implementation of branch metric computation.

4. ACS 처리블록

<그림 6>은 radix-4 트렐리스 병렬구조의 각 상태에 대해, 가지 메트릭과 사이클 타임 k 에서의 경로 메트릭으로부터 사이클 타임 $k+2$ 에서의 경로 메트릭을 계산하고, 4개의 입력 경로 중에서 최적인 생존경로를 선정하는 ACS부의 구성을 나타낸 것이다. radix-2 트렐리스 2개 경로의 생존경로가 한번의 ACS 연산으로 결정되고, 이에 관한 정보가 2비트의 결정정보 d_k^{xab} 으로 출력된다.

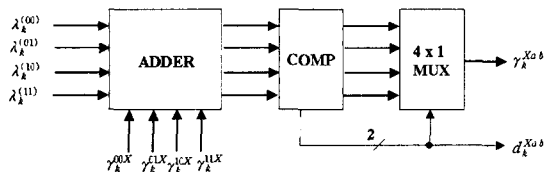


그림 6. Radix-4 트렐리스 병렬구조의 ACS 부
Fig. 6. The radix-4 trellis parallelized architecture for ACS.

ACS의 반복연산으로 발생할 수 있는 경로 메트릭의 Overflow를 방지하기 위해서는 경로 메트릭의 최대값이 일정하게 유지되도록 제어하는 것이 필요하다. 이와 같은 경로 메트릭의 정규화(normalization) 과정을 간단한 구조로 구현하기 위한 많은 연구가 진행되어 왔다^[6, 24, 25]. 본 논문에서는 각 상태의 경로 메트릭간의 차이는 상한으로 제한되는 특성을 사용한다^[24].

$$\Delta \gamma_{MAX} \leq \lambda_{max} \log_2 N \quad (16)$$

여기서 λ_{max} 은 가지 메트릭의 최대값이고, N 은 트렐리스의 전체 상태수이다. 정규화의 기본원리는 임의의 경로 메트릭이 사전에 설정된 기준치 γ_{THR} 이상이

되면, 모든 상태의 경로 메트릭으로부터 일정한 값 $\Delta \gamma$ 를 감산해도 최적경로의 선정에는 아무런 영향을 미치지 않는 성질이다. 즉,

$$\begin{aligned} & \text{if } \gamma_{j,k} \geq \gamma_{THR}, \\ & \text{then } \gamma_{j,k} = \gamma_{j,k} - \Delta \gamma \quad (j=0,1,2,\dots,N-1) \quad (17) \end{aligned}$$

기준치 γ_{THR} 설정의 기본은 모든 상태의 경로 메트릭이 underflow나 overflow가 발생하지 않도록 해야하고, 임의의 경로 메트릭에 대한 기준치 초과 여부의 검출과 감산과정이 간단하고 효율적으로 구성할 수 있어야 한다. 또한, 각 상태 경로 메트릭의 wordlength가 가능한 작은 값을 유지할 수 있도록 해야 한다. 이를 위해, 본 논문에서는 가지 메트릭 및 경로 메트릭이 부호(sign)을 갖지 않는 b 비트 및 p 비트의 사이즈로 한정된 것으로 한다. 이 경우, 가지 메트릭의 최대값

$$\max(\lambda_{j,k}) \max(\lambda_{j,k}) = 2^b - 1 \quad (18)$$

이고, 경로 메트릭의 최대값은 $2^b - 1$ 이 된다. 따라서, 경로 메트릭간 차분의 최대값 $\Delta \gamma_{MAX}$ 은

$$\begin{aligned} & \text{Max. path metric difference value:} \\ & \Delta \gamma_{MAX} \leq 2^{b-1} \quad (19) \end{aligned}$$

으로 주어진다. 이로 인해 임의의 경로 메트릭이 기준치 $\gamma_{THR} = 2^{b-1}$ 을 초과하면, 나머지 모든 상태의 경로 메트릭도 2^{b-2} 을 초과하게 될 것이다. 식 (17)로부터 $\Delta \gamma = 2^{b-2}$ 으로 설정하면, 임의의 경로 메트릭의 MSB가 1에서 0으로 변화되는 순간을 검출해서 모든 경로 메트릭으로부터 $\Delta \gamma$ 의 값을 감산하면 된다. 이 때 감산은 간단한 1비트 우측천이만으로 수행된다.

5. 역방향 상태천이의 제어에 의한 역추적 복호

역추적 기법에 의한 최적 생존경로의 결정은 복호값이 L_{min} 이상으로 경로 메모리를 역추적하면, 역추적이 시작되는 시점의 모든 상태는 공통된 하나의 최적상태에 통합될 확률이 매우 높은 이론에 기반을 두고 있다^[18-23]. 즉, 모든 생존경로는 $L \geq L_{min}$ 이상의 경로를 역추적하면, 모든 상태는 하나의 최적 상태에 통합될 확률이 매우 높다. 따라서, 임의의 사이클 타임 k 의 모든 상태로부터 $L \geq L_{min}$ 사이클 타임 동안 역추적하면, 이들 상태는 ML의 의미에서 사이클 타임 $k-L$ 에서 최

적 경로가 되는 하나의 상태에 통합되는 것이다^[22, 23].

ACSU로부터 출력되는 결정비트 d_k^{Xpq} 를 기반으로 역방향의 상태천이를 연속적으로 제어함으로서 간단한 구조의 역추적 복호 시스템을 구현할 수 있다. 이를 위해, 사이클 타임 $k+2$ 의 임의의 상태에서부터 2-step 트렐리스 이전의 사이클 타임 k 의 과거상태로 복귀하는데 필요한 매핑관계를 조사하는 것이 필요하다. <그림 7>은 이를 기술하기 위해 구속장 $K=5$, 부호율 $R=1/2$ 인 길쌈 부호기의 구조를 예로 나타낸 것이다. V_{0k} 및 V_{1k} 는 각각의 생성 다항식으로부터 발생되는 부호어이다. 트렐리스의 각 상태는 부호기의 레지스터에 저장되는 값 $x_{ji,k}$ ($i=0,1,2,3$)로 정해진다. 여기서 $x_{j0,k}$ 를 LSB, $x_{j3,k}$ 를 MSB로 취급한다.

먼저, <그림 7>에서 사이클 타임 k , j 번째 상태 $s_{j,k}$ 는

$$s_{j,k} = (x_{j3,k}, x_{j2,k}, x_{j1,k}, x_{j0,k}) \quad (20)$$

으로 주어지고, 사이클 타임 $k+1$, j 번째 상태 $s_{j,k+1}$ 는 사이클 타임의 진행에 따라 입력정보 u_k 의 입력 및 레지스터 내 저장된 값들의 우측천이로 다음과 같이 전개되는 것을 알 수 있다.

$$\begin{aligned} s_{j,k+1} &= (x_{j3,k+1}, x_{j2,k+1}, x_{j1,k+1}, x_{j0,k+1}) \\ &= (x_{j2,k}, x_{j1,k}, x_{j0,k}, u_k) \end{aligned} \quad (21)$$

동일한 과정으로 사이클 타임 $k+2$, j 번째 상태 $s_{j,k+2}$ 도

$$\begin{aligned} s_{j,k+2} &= (x_{j3,k+2}, x_{j2,k+2}, x_{j1,k+2}, x_{j0,k+2}) \\ &= (x_{j2,k+1}, x_{j1,k+1}, x_{j0,k+1}, u_{k+1}) \\ &= (x_{j1,k}, x_{j0,k}, u_k, u_{k+1}) \end{aligned} \quad (22)$$

이 된다.

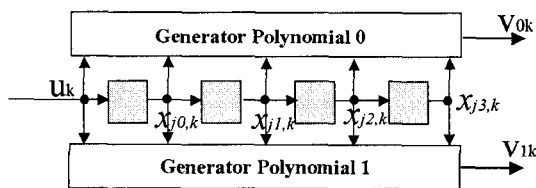


그림 7. $K=5$, $R=1/2$ 인 길쌈부호기
Fig. 7. The convolutional encoder of $K=5, R=1/2$.

<그림 3>의 Radix-4 트렐리스도에서 사이클 타임 k 에서 $k+2$ 로의 천이에 따라, 상태의 천이는

$$(pqX) \rightarrow (Xab) \quad (23)$$

로 변화되면서 (pq) 가 부호기의 레지스터에서 천이되어 빠져나가는 비트들이고, (ab) 는 새로 입력되는 정보 비트들에 해당되는 것을 알 수 있다. 이들 비트들은 식 (20)과 (22)로부터

$$(p, q) = (x_{j3,k}, x_{j2,k}), (a, b) = (u_k, u_{k+1}) \quad (24)$$

의 관계를 갖는다. 또한, <그림 3>으로부터 ACS부에서 생존경로를 나타내는 2 비트의 결정비트는 (p, q) 의 값과 일치하는 것을 알 수 있다. 따라서, j 번째 상태에서의 결정비트 $d_{j,k}^{Xpq}$ 는

$$d_{j,k}^{Xpq} = (d_{j,k}^{Xp}, d_{j,k}^{Xq}) = (x_{j3,k}, x_{j2,k}) \quad (25)$$

가 된다. 사이클 타임 $k+2$ 의 임의의 상태 $s_{j,k+2}$ 에서 사이클 타임 k 의 임의의 상태 $s_{i,k}$ 로의 역추적은

$$s_{i,k} = (d_k^{Xp}, d_k^{Xq}, x_{j3,k+2}, x_{j2,k+2}) \quad (26)$$

의 역방향 상태천이로의 제어로 수행된다.

<그림 8>은 제안된 역방향 상태천이의 연속제어에 의한 역추적 비터비 디코더의 구조를 블록도로 나타낸 것이다. 본 논문에서 제안된 SMU의 구성은 하나의 ACS 사이클 타임에서 출력되는 2비트의 결정비트를 병렬로 저장하는 $2N \times M$ 의 생존 메모리와 $\frac{(M+1)}{2} \times \log_2 N$ 사이즈의 역방향 상태정보를 저장하는 상태 메모리로 구성된다. 이 때, M 은 생존 메모리의 사이즈로 $M=2L-1$, (단, $L \geq L_{\min}$)의 관계를 갖는다. 메모리 번지의 제어를 위해 다소 복잡한 주변회로를 필요로 하거나, 각 ACS 사이클 타임마다 최적 생존 경로 매트릭인 상태를 검출하기 위한 비교회로가 요구되는 기존의 역추적 복호 기법과 달리, 제안방법은 단순한 천이동작만으로 복호가 가능한 특징을 갖는다.

<그림 8>의 역방향 상태천이의 제어에 의한 역추적 복호과정은 다음과 같다.

- (1) 생존 메모리 M 및 상태 메모리 $L = \frac{M+1}{2}$ 의 초기화
- (2) Radix-4 트렐리스 구조의 가지 매트릭 및 경로 매트릭의 계산과 결정비트의 출력

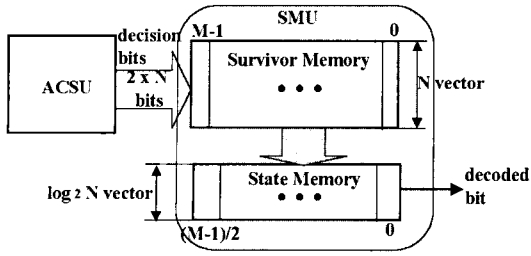


그림 8. 역방향 상태천이의 제어에 의한 역추적 복호의 구성
 Fig. 8. The structure for the implementation of trace back decoding with backward state transition control.

- (3) 결정비트를 생존 메모리에 사이즈 $k = \frac{M}{2} \geq L_{min}$ 이 될 때까지 k 를 증가시키면서 과정 (2)를 반복하면서 저장함. 이 때, 생존 메모리는 1 사이클 타임씩 우측으로 천이시킴.
- (4) $k = L$ 이면,
- 생존 메모리의 천이 및 결정비트의 저장
 - 각 상태에 대한 역방향으로의 추적을 개시함. 역방향 상태추적의 초기값은 다음으로 주어진다.

$$s_{j,k} = (d_{j,k}^{Xa}, d_{j,k}^{Xa}, x_{j3,k}, x_{j2,k}) \quad (27)$$

- ACS가 처리되는 매 사이클 마다 각 상태의 역추적은

$$s_{j,k-L} = (d_{k-L}^{Xab}, x_{j3,k-L+2}, x_{j2,k-L+2}) \quad (28)$$

단 d_{k-L}^{Xab} 는 그 전의 상태값을 제어신호로 사용한 $N \times 1$ MUX에 의해 선정된 결정비트 d_{k-L}^{Xa} 이다.

- (5) $t = 0$ 의 최종단에서, 상태 메모리에 저장된 각 상태값을 비교한다. 이 때, 상태 통합이 발생한 경우에는 상태 메모리의 모든 상태가 동일한 값을 가지게 되고, 상태 통합이 일어나지 않는 경우에는 복호 오류의 발생확률이 높은 것에 대응한다. 이 경우 다수결 결정(majority voting), 즉, 동일한 값을 갖는 다수의 상태값을 통합된 최종 상태로 판단하여 복호한다.

이상의 복호과정으로부터 수신 데이터는 생존 메모리의 사이즈에 대응하는 M 사이클 타임 이후부터 매 ACS 사이클 타임에 동기되어 복호된다. <그림 9(a)>는 역추적 및 복호과정을 수행하는 시스템릭 어레이

구조에 의한 단위 처리블록의 회로구성을 나타낸 것이다. 2-사이클 타임에 해당하는 생존 메모리와 역방향 상태천이의 변화를 저장하기 위한 1-사이클 타임의 상태 메모리로 구성된다. 따라서, 하나의 상태 메모리 변화는 2개의 생존 메모리의 천이에 대응하므로, 최종단에서 생존 메모리와 상태 메모리의 천이시간이 동기화되어 처리된다. 연속적인 역방향 상태천이의 추적은 식 (28)을 사용해서 각 사이클 타임마다 상태값을 제어신호로 사용하고, 단위 처리 블록에서 출력되는 결정비트를 입력 데이터로 사용하는 8×2 MUX에 의해 간단히 처리된다. <그림 9(b)>는 <그림 9(a)>의 단위 처리 블록의 회로구성을 블록도로 나타낸 것으로 이를 생존 메모리 및 역방향 상태천이 제어(Survivor Memory and Backward State Transition Control : SM & BSTC) 블록이라 부르기로 한다.

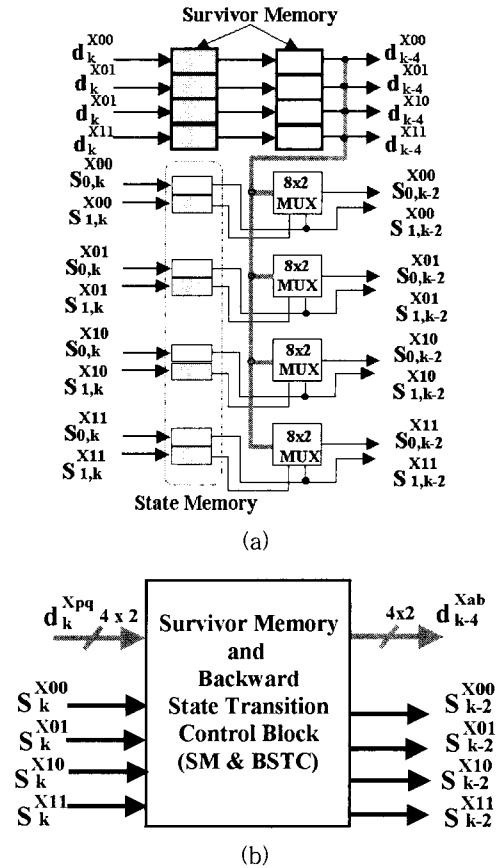


그림 9. 단위처리 블록의 (a) 회로구성과 (b) 블록도
 Fig. 9. (a) The circuit implementation and (b) block representation of a processing unit with systolic arrayarchitecture.

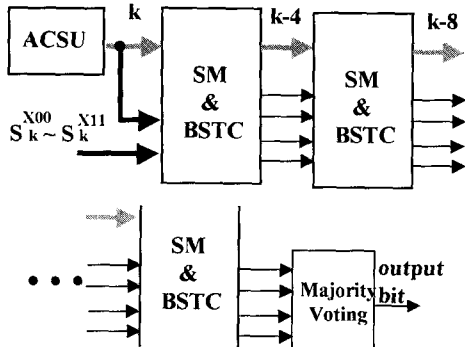


그림 10. 역방향 상태천이의 제어와 복호를 위한 시스템릭 어레이 구성

Fig. 10. The systolic array implementation for the backward state transition control and decoding.

<그림 10>은 <그림 9(b)>의 SM & BSTC 블록을 나열해서 시스템릭 어레이 구조로 역추적 및 복호 과정의 구성을 나타낸 것이다. 시스템릭 배열구조에 의한 SM부의 구현을 나타낸 것으로 최종 출력단의 다수결 결정 블록은 채널잡음이 많은 환경에서 낮은 SNR의 신호를 전송하는 경우, 생존 메모리 사이즈 M 동안 역추적을 해도 하나의 상태로 통합되지 않을 가능성을 보완하기 위한 것이다. 비록 발생확률은 매우 낮은 경우에도 모든 상태가 하나의 상태로 통합되지 않을 경우, 같은 상태값을 갖는 상태가 많은 것보다 정확한 상태라 판단해서 복호하기 위함이다.

IV. 초고속 무선 랜에의 응용

3절에서 기술한 radix-4 트렐리스 병렬구조의 비터비 디코더의 구체적인 성능을 검증하기 위해 초고속 무선 랜 규격인 IEEE 802.11a PHY 계층의 채널 부호화 및 복호에 적용했다. IEEE 802.11a PHY 계층에서는 채널 부호화로 구속장 $K=7$, 생성다항식 $G_0=133_8, G_1=171_8$ 인 길쌈부호기를 사용하고 있다. 다양한 통신속도에 적응적을 대응할 수 있도록 3종류의 부호율 R 과 8종류의 데이터 전송율을 규격으로 정의하고 있다. 데이터 전송율과 부호율의 관계를 <표 1>에 나타낸다.

가지 메트릭은 유클리드 거리 메트릭에 의한 식 (13) 또는 (14)로 계산되고, 수신 부호어는 부호(sign)를 포함한 4비트의 연판정을 사용했다. VHDL 언어에 의한 채널 코덱의 VLSI 구현에서 사용한 각종 메트릭의 wordlength는 다음과 같다.

표 1. IEEE 802.11a PHY 계층의 부호율 R 과 데이터 전송율과의 관계

Table 1. The relationship between data rate and coding rate R of the IEEE 802.11a PHY layer.

Data Rate [Mbps]	Modulation	Coding Rate (R)
6	BPSK	1/2
9	BPSK	3/4
12	QPSK	1/2
18	QPSK	3/4
24	16QAM	1/2
36	16QAM	3/4
48	64QAM	2/3
54	64QAM	3/4

- 수신 부호어 : 4 비트 (부호비트 포함)
- Radix-2 가지 메트릭 : 부호를 갖지 않는 5비트
- Radix-4 가지 메트릭 : 부호를 갖지 않는 5비트
- 경로 메트릭 : 부호를 갖지 않는 8비트

또한, 역방향 상태천이의 제어에 의한 역추적 복호 수행을 위한 복호깊이 L 은 $L=25$ 의 값을 부호율에 관계없이 일정하게 사용했다.

VHDL에 의한 모의실험에서 PN 시퀀스 생성기로부터 발생된 데이터를 길쌈부호기의 입력으로 사용하고, 채널 잡음은 인위적으로 생성시켜 부가되는 것으로 했다. IEEE 802.11a PHY 계층에서 정의된 전체 데이터 전송률 중에서, 본 제안방법의 유효성을 검증하기 위해 3종류의 부호율 1/2, 2/3, 3/4과 이들 각각에 대한 데이터 전송율 24Mbps, 48Mbps, 54Mbps 등의 실험을 수행하고 그 결과를 <그림 11~13>에 나타낸다. 본 실험에서 사용한 시뮬레이션 툴은 ModelSim 5.7이고, 벡터 파일의 입출력 신호는 다음 <표 2>와 같다. 이들 결과

표 2. 벡터파일의 입출력 신호

Table 2. Input/Output signals of the vector file.

입출력 신호명	설명
t_sig_ck_72m	길쌈 부호기의 출력 클럭
t_sig_ck_rate	비터비 디코더 클럭
t_sig_psdu_in	길쌈부호기 입력 데이터 파형
stateXX	SMU 역방향 상태천이 메모리의 최종 상태값 (전체 64 상태 중 XX = 00, 08, 16, 32, 63 만을 표시함)
t_sig_viterbi_out	비터비 디코더 출력

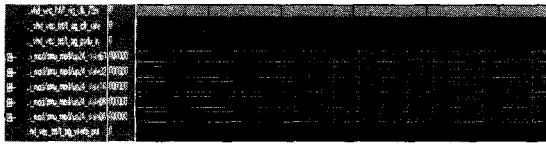


그림 11. 24Mbps, R=1/2에 대한 VHDL 코딩 결과
Fig. 11. The VHDL coding result for 24Mbps, R=1/2.



그림 12. 48Mbps, R=2/3에 대한 VHDL 코딩 결과
Fig. 12. The VHDL coding result for 48Mbps, R=2/3.

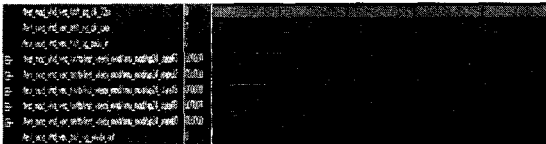


그림 13. 54Mbps, R=3/4에 대한 VHDL 코딩 결과
Fig. 13. The VHDL coding result for 54Mbps, R=3/4.

로부터, 디코더의 출력과형($t_sig_viterbi_out$)은 복호깊이에 해당하는 지연시간 후 길썸부호기의 입력신호($t_sig_psdu_in$)와 일치되어 있음을 볼 수 있다.

V. 결론

본 논문에서는 2-스텝 트렐리스 radix-4 병렬구조와 역방향 상태천이의 제어에 의한 역추적 비터비 디코더를 구현하는 방법을 제안하고, 초고속 무선 랜인 IEEE 802.11a 채널부호 및 복호에 적용했다. 하나의 비터비 구조로서 다양한 부호율의 채널 부호를 취급할 수 있도록 하기 위해, radix-4 트렐리스의 가지 메트릭은 2계층의 병렬구조로 구성해서 사용했다. 또한, 본 논문에서는 역추적 복호를 위한 새로운 생존 메모리 제어 방법으로, 역방향 상태천이의 제어에 의한 역추적 복호방법을 기술했다.

본 제안방법의 비터비 디코더는, throughput을 2배 개선시킬 수 있을 뿐만 아니라 다양한 부호율을 갖는 채널의 복호에 대응할 수 있는 통합구조의 비터비 디코더 구성이 가능하다. 또한, 역추적 복호방법의 개선으로, 생존 메모리의 제어가 간단하고 주변 제어회로가 필요없어 회로구성이 간단하다. 그러나, ACS 회로를

구현함에 있어 회로 규모가 다소 증가될 가능성이 있어, 회로설계의 최적화를 위한 연구가 요구된다. 구체적인 통신 시스템에의 응용으로, IEEE 802.11a 초고속 무선랜의 PHY 계층의 채널 부호 및 복호에 적용하고, HDL에 의한 시뮬레이션 결과를 제시했다. 제안방법에 의한 비터비 디코더는 다양한 채널 부호율 및 고속 데이터 전송율의 통신 시스템에 적합한 구조임을 확인할 수 있었다.

참고 문헌

- [1] D. J. Costello, Jr., H. Hegenaur, H. Imai, and S. B. Wicker, "Applications of error-control coding," IEEE Trans. Information Theory, vol. 44, no. 6, pp. 2531 - 2560, Oct. 1998.
- [2] A. Sabanmaria and F. J. Lopes-Hernandez, Wireless LAN : Standards and application, Artech House, 2001.
- [3] A. J. Viterbi and J. K. Omura, Principles of digital communication and coding, McGraw-Hill, NY, 1979.
- [4] G. G. Forney, "The Viterbi algorithm," Proceedings of the IEEE, vol. 61, no. 3, pp. 268 - 278, March 1973.
- [5] H. Liou, "Implementing the Viterbi algorithm," IEEE Signal Processing Mag., pp. 42-52, Sept. 1995.
- [6] H. Dawid, O. J. Joeressen, and H. Meyr, "Viterbi decoders : High performance algorithms and architectures," Digital Signal Processing for Multimedia Systems edited by K. Parhi and T. Nishitani, Marcel Dekker 1999.
- [7] P. G. Gulak and T. Kailath, "Locally connected VLSI architectures for the Viterbi algorithm," IEEE J. Selected Areas Commun., vol. 6, no. 3, pp. 527 - 537, April 1988.
- [8] P. J. Black and T.H. Meng, "Hybrid survivor path architectures for Viterbi decoders," In Proc. Int. Conference on Acoustics, Speech, and Signal Processing 93 (ICASSP'93), pp. 433 - 436, Nov. 1993.
- [9] T. K. Trung, M. Shih, I. S. Reed, and E. H.

- Satorius, "A VLSI design for a trace-back Viterbi decoder," IEEE Trans. Commun., vol. 40, no. 3, pp. 616 - 624, March 1992.
- [10] C. Shung, H. Lin, R. C. Sypher, P. H. Siegel, and H. K. Thapar, "Area-efficient architecture for the Viterbi algorithm - Part I: Theory," IEEE Trans. Commun., vol. 41, no. 4, pp. 636 - 644, April 1993.
- [11] G. Fettweis and H. Meyr, "High-rate Viterbi processor: A systolic array solution," IEEE J. Selected Areas Commun., vol. 8, no. 8, pp. 1520 - 1534, Oct. 1990.
- [12] P. G. Gulak and E. Shwedyk, "VLSI structures for Viterbi receivers: Part I-General theory and applications," IEEE J. Selected Areas Commun., vol. SAC-4, pp. 142 - 154, Jan. 1986.
- [13] P. J. Black and T. H. Meng, "A 140-Mb/s, 32-state, radix-4 Viterbi decoder," IEEE J. Solid-State Circuits, vol. 27, no. 12, pp. 1877 - 1885, Dec. 1992.
- [14] J. Sparso, H. N. Jorgensen, E. Paaske, S. Pedersen, and T. R. Petersen, "An area-efficient topology for VLSI implementation of Viterbi decoders and other shuffle-exchange type structures," IEEE J. Solid-State Circuits, vol. 26, no. 2, pp. 90 - 97, Feb. 1991.
- [15] H. K. Thapar and H. M. Cioffi, "A block processing method for designing high-speed Viterbi detectors", In Proc. International Conference on Commun. 89 (ICC'89), pp. 1096 - 1100, 1989.
- [16] P. J. Black and T. H. Meng, "A unified approach to the Viterbi algorithm state metric update for shift register processes", In Proc. International Conference on Acoustics, Speech, and Signal Processing 92 (ICASSP'92), pp. V.629 - V.632, 1992.
- [17] H. Burkhardt and L. C. Barbosa, "Contributions to the application of the Viterbi algorithm," IEEE Trans. Information Theory, vol. IT-31, no. 5, pp. 626 - 634, Sept. 1985.
- [18] C. M. Radar, "Memory management in a Viterbi algorithm," IEEE Trans. Commun., vol. 29, pp. 1399 - 1401, Sep. 1981.
- [19] G. Feygin and P. G. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," IEEE Trans. Commun., vol. 41, no. 3, pp. 425 - 429, March 1993.
- [20] R. Cypher and C. B. Shung, "Generalized trace back techniques for survivor memory management in the Viterbi algorithm," In Proc. GLOBECOM, pp. 1318 - 1322, Dec. 1990.
- [21] G. Fettweis, "Algebraic survivor memory management for Viterbi detectors," In Proc. Int. Conference on Commun. 92 (ICC'92), pp. 339 - 343, 1992.
- [22] R. J. McEliece and I. M. Onyszchuk, "Truncation effect in Viterbi decoding," In Proc. Proc. of the IEEE Confer. on Military Commun., (Boston, MA), pp. 29.3.1 - 29.3.3 Oct. 1989.
- [23] I. M. Onyszchuk, "Truncation length for Viterbi decoding," IEEE Trans. Commun., vol. 39, no. 7, pp. 1023 - 1026, July 1991.
- [24] A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," IEEE Trans. Commun., vol. 37, no. 11, pp. 1220 - 1222, Nov. 1989.
- [25] C. Shung et al, "VLSI architectures for metric normalization in the Viterbi algorithm," In Proc. International Conference on Commun. 90 (ICC'90), pp. 1723 - 1728, 1990.

저 자 소 개



鄭 昌 根(正會員)

1982년 2월 : 경북대학교 전자공학과 졸업. 1984년 2월 : 서울대학교 대학원 전기공학과 공학석사. 1993년 2월 : 일본 동경대학 전기공학과 공학박사. 1984년 1월~1997년 8월 : LG종합기술원 책임연구원

1995년 4월~1996년 3월 : 일본 방송통신기구 초빙연구원. 2002년 1월~2002년 4월 : 동경대학 초빙교수. 1997년 9월~현재 : 호서대학교 정보 제어공학과. <주관심분야 : 디지털 신호처리, 디지털 통신, 영상처리 및 부호화, Image Sensor 등.>