

論文2000-37SD-2-10

# 멀티미디어 통신용 병렬 아키텍처 고속 비터비 복호기 설계 (Implementation of a Parallel Viterbi Decoder for High Speed Multimedia Communications)

李丙哲\*, 鮮于明勳\*  
(Byung Chul Lee and Myung Hoon Sunwoo)

## 요 약

비터비 복호기는 직렬 복호 방식과 병렬 복호 방식 2 가지로 분류할 수 있다. 병렬 비터비 복호기는 직렬 비터비 복호기에 비해 보다 높은 데이터율을 얻을 수 있다. 본 논문에서는 고속 멀티미디어 통신을 위한 병렬 비터비 복호기 구조를 설계하고 구현한다. 설계한 비터비 복호기는 고속 동작을 위해 64개의 PE(Processing Element)를 사용해 한 클럭에 처리가 가능하도록 하였다. 또한 파이프라인 스테이지를 갖는 시스톨릭 어레이 구조의 TB(Traceback) 블록을 설계하였다. 본 논문에서 설계한 비터비 복호기는 puncturing을 통해 부호율 1/2, 2/3, 3/4, 5/6, 7/8을 지원한다. Verilog 모델을 구현하였고 0.6  $\mu\text{m}$  Samsung KG75000 SOG 셀 라이브러리를 이용하여 논리합성을 수행하였다. 구현된 비터비 복호기는 약 100,400 게이트이며 동작 속도는 worst case에서 70MHz로 기존 상용 칩들보다 빠르다.

## Abstract

The Viterbi decoders can be classified into serial Viterbi decoders and parallel Viterbi decoders. Parallel Viterbi decoders can handle higher data rates than serial Viterbi decoders. This paper designs and implements a fully parallel Viterbi decoder for high speed multimedia communications. For high speed operations, the ACS (Add-Compare-Select) module consisting of 64 PEs (Processing Elements) can compute one stage in a clock. In addition, the systolic array structure with 32 pipeline stages is developed for the TB (traceback) module. The implemented Viterbi decoder can support code rates 1/2, 2/3, 3/4, 5/6 and 7/8 using punctured codes. We have developed Verilog HDL models and performed logic synthesis. The 0.6  $\mu\text{m}$  SAMSUNG KG75000 SOG cell library has been used. The implemented Viterbi decoder has about 100,400 gates, and is running at 70 MHz in the worst case simulation.

## I. 서 론

이동 통신이나 위성통신 시스템에서는 채널에서 발생하는 랜덤 오류를 정정하기 위해 랜덤 오류 정정 능력

이 우수한 콘볼루션 부호기와 비터비 복호기를 사용한 다<sup>[1]</sup>. 현재 CDMA 방식의 셀룰라 이동 통신이나 유럽의 GSM 등과 같은 음성 통신에서 사용되고 있는 비터비 복호기는 데이터 전송속도가 수십 Kbps에 불과하며 단말기의 전력 소모를 적게 하고 부피를 줄이기 위해 직렬 복호 방식을 주로 채택하고 있다. 그러나 LMDS(Local Multipoint Distribution Systems)나 MCNS(Multimedia Cable Network System)등의 표준안을 만족하는 CATV나 VOD 등

\* 正會員, 亞州大學校 電子工學府

(School of Electronical Eng., Ajou Univ.)

接受日字:1998年11月30日, 수정완료일:2000年2月1日

의 멀티미디어 통신 서비스를 위해서는 수 십 Mbps 의 데이터 전송속도가 필요하다.

기존의 음성 통신에 사용되는 직렬 방식 비터비 복호기로는 이러한 전송 속도가 높은 서비스를 지원할 수 없기 때문에 병렬 복호 방식으로 비터비 복호기를 설계해야 한다. 기존의 비터비 복호기 칩으로는 25MHz의 동작 속도를 갖는 QUALCOMM사의 Q1900<sup>[9]</sup>, 37 MHz인 MOTOROLA사의 MC92300<sup>[10]</sup>, 60 MHz를 지원하는 LSI LOGIC사의 L64709<sup>[11]</sup> 등이 개발되었다. LMDS 표준안에서 규정한 데이터율이 50Mbps 이상이므로 기존 칩들은 지원하지가 어렵기 때문에 고속의 처리속도를 갖는 비터비 복호기 설계가 요구된다.

본 논문에서는 고속 데이터 처리가 가능하도록 병렬 아키텍처를 적용해 비터비 복호기를 설계하였다. 기존 병렬 구조를 사용한 비터비 복호기에 비해 ACS 블록을 여러단의 파이프라인으로 구성하였고 생존자 경로를 구하기 위한 블록의 임계경로를 줄이기 위해 PE 하나에 사용되는 MUX의 수를 2개로 제한하였다. 구현한 콘볼루션 부호기와 비터비 복호기는 구속장(K) 7, 부호화율 1/2이고 2/3, 3/4, 5/6, 7/8의 다양한 부호화율과 3-bit 연판정(soft decision)을 사용해 2 dB의 추가적인 부호 이득(coding gain)을 얻도록 하였다. 콘볼루션 부호기의 생성 다항식으로 (171,133)<sub>8</sub>을 사용하였고 puncturing 기법을 적용해 출력 데이터의 부호율을 조절할 수 있도록 하였다<sup>[8]</sup>. ACS(Add-Compare-Select) 연산을 병렬로 수행하기 위한 방법으로 전체 상태 수  $2^{K-1}$ 에 해당하는 64개의 PE(Processing Element) 들을 병렬로 연결해 구현하였다. 생존자 경로 메모리 블록은 파이프라인 기법을 적용한 시스톨릭 어레이(systolic array) 아키텍처를 사용하여 한 클럭 주기동안 하나의 데이터가 복호되어 고속 데이터 처리가 가능하다. 유일한 생존자 경로를 구하기 위한 아키텍처로는 복호 속도가 가장 빠른 레지스터 교환(Register Exchange) 방식<sup>[2-4]</sup>을 사용하였다.

Top-down 설계 방식에 따라 Verilog HDL (Hardware Description Language)을 사용해 기능 모델과 아키텍처 모델을 구현하였고 SYNOPSIS™ CAD 툴을 이용하여 논리합성을 수행하였다. CADENCE™ CAD 툴을 이용하여 기능 검증 시뮬레이션과 타이밍 시뮬레이션을 수행하였으며 worst

case 시뮬레이션 주파수는 70 MHz이다. 설계된 비터비 복호기의 총 게이트 수는 약 10만 개이다. 구현된 비터비 복호기는 기존의 상용 칩들과 동일한 사양을 지원하며 보다 고속의 동작 속도를 갖는다.

본 논문은 다음과 같이 구성된다. 2장에서 길쌈 부호기와 비터비 복호기 아키텍처에 대해 기술하고 3장에서 구현 방법과 성능 평가에 대해 기술하며 끝으로 4장에서 결론을 맺는다.

## II. 고속 병렬처리 방식 비터비 복호기 아키텍처

이 장에서는 길쌈 부호기(convolutional encoder)와 비터비 복호기의 사양과 블록별 아키텍처에 대해 설명한다. 본 논문에서 설계한 비터비 복호기는 DAVIC(Digital Audio Visual Council) 표준안의 LMDS 시스템에 적용할 수 있다. LMDS 표준안의 길쌈 부호기와 비터비 복호기 사양은 구속장 7과 부호화율 1/2에 punctured 모드를 통해 5가지 부호율을 지원하도록 하고 있으며 본 논문에서는 이와 동일한 사양의 비터비 복호기를 설계하였다.

### 1. 길쌈 부호기

길쌈 부호기의 아키텍처는 그림 1과 같다. 길쌈 부호기는  $k$  비트의 입력 데이터를 받아들여  $n$  비트의 출력을 생성하는 부호기로 현재 입력 데이터와  $(K-1)$  비트의 과거 데이터를 사용해 부호어를 생성해 낸다. 구속장의 길이는 길쌈 부호화를 수행할 때 사용된 데이터의 길이를 나타내는 것으로 많은 입력 데이터로 콘볼루션을 취하면 출력 결과가 입력 데이터와 높은 상관 관계를 갖게 되므로 구속장의 길이가 클수록 높은 부호 이득을 얻을 수 있다.

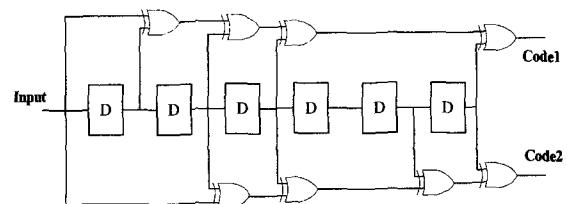


그림 1. 길쌈 부호기  
Fig. 1. The convolutional encoder.

표 1에 구속장에 따른 최대 자유도 (dfree)를 나타

내었으며 이 때 가장 최적인 생성 다항식을 나타내었다. 표 1에서와 같이 길쌈 부호기 설계시 non-systematic 방법이 systematic 방법보다 자유도가 큰 것을 알 수 있다. 본 논문에서는 non-systematic 길쌈 부호기를 설계하였으며 구속장 7과 부호율 1/2을 사용하였다.

생성 다항식으로는 표 1에 나타낸 최적 값을 사용하였으며 식 1에 나타내었다. 또한 이 때의 부호 이득은 3.98 dB이다.

$$g_1(x) = 1111001_{(2)} = 1 + x + x^2 + x^3 + x^6$$

$$g_2(x) = 1011011_{(2)} = 1 + x^2 + x^3 + x^5 + x^6$$
(1)

표 1. 부호율 1/2인 콘볼루션 코드의 최대 자유도와 최적의 생성 다항식

Table 1. The maximum  $d_{free}$  and the best generator polynomials of rate 1/2 convolutional codes.

구속장(K)	Systematic matrix $d_{free}$	Nonsystematic matrix $d_{free}$	생성다항식	
			$g_{(1)}$	$g_{(2)}$
3	4	5	7	5
4	4	6	17	15
5	5	7	35	23
6	5	8	57	65
7	6	10	133	171

길쌈 부호기에서 출력된 데이터의 전송률은 부호율이 1/2이므로 입력 데이터에 비해 2 배 높다. 이 경우 전송 데이터의 정보 전송률이 낮기 때문에 이를 높이기 위해 일정한 주기에 따라 특정 위치의 데이터를 지운 후 전송하는 puncturing 기법을 사용한다. 삭제된 데이터는 전송되지 않으며 수신부에서는 삭제된 위치에 0을 삽입해 원래의 부호율을 복원해낸다. 부호율 1/2 인 코드를 사용하여 생성되는 2/3, 3/4, 5/6, 7/8 등의 다양한 부호율과 그에 따른 삭제 패턴(deleting pattern)은 이전의 많은 연구들을 통해 최적 값이 구해져 있다<sup>[8]</sup>. 부호율 1/2 인 길쌈 부호기에 7/8 로 punctured 코딩을 사용하면 정보 전송률은 높아지는 대신 약 2 dB 가량 부호 이득이 저하되나 부호율이 7/8인 길쌈 부호기를 사용하는 것에 비해 복호기의 하드웨어 부담 증가가 적은 장점을 갖는다.

2. 비터비 복호기 아키텍처

비터비 복호기의 전체 아키텍처는 그림 2와 같다. 비터비 알고리즘은 수신된 부호어를 격자도(trellis diagram)와 비교하여 해밍 거리(Hamming distance)가 최소가 되는 경로를 찾아내는 알고리즘이며 이를 하드웨어로 구현한 것이 비터비 복호기이다. 비터비 복호기는 수신된 부호어와 부호기의 쉬프트 레지스터 상태에 따라 발생하는 부호어와의 차이를 계산하는 BMC(Branch Metric Calculation) 블록, BMC로부터 발생된 BM(Branch Metric) 값에 현재의 상태 값을 더한 뒤 두 값을 비교해 작은 값을 선택하는 ACS 블록, 마지막으로 ACS로부터 선택된 최소거리를 갖는 경로들을 저장하고 복호해 내는 생존자 경로 메모리 블록의 3 블록으로 크게 구성된다.

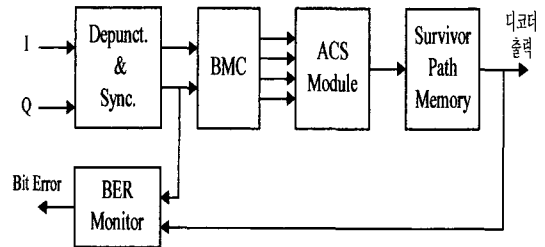


그림 2. 비터비 복호기 아키텍처  
Fig. 2. The Viterbi decoder architecture.

Depuncturing 블록에서는 puncturing 블록을 통해 삭제된 데이터 위치에 0을 삽입해주어 길쌈 부호기의 부호율 1/2과 같아지게 한다. Depuncturing을 통해 원래의 부호율로 변환된 데이터는 BMC 블록에 입력된다. 수신된 부호어와 송신한 부호어간의 차를 알기 위해서 수신단에서는 송신단의 생성 다항식을 알고 있어야 하며 이것을 비터비 복호기의 BMC에서 담당한다. 복호기의 입력은 I, Q 채널 각각 3 비트로 연관정된 데이터로 약 2dB 정도 추가적인 부호 이득을 얻게 되며 이에 따른 하드웨어는 거의 증가하지 않는다. 부호율이 1/2일 경우 4가지 부호어가 존재하므로 BMC 블록을 통해 4가지 경로 값들이 출력된다.

ACS 블록에서는 BMC에서 계산된 경로 값들을 비교해 각 상태의 경로로 입력되는 값들 중 작은 값을 선택하여 출력하는 기능을 하며 64개의 PE들로 구성된다. 하나의 상태값을 계산할 수 있는 PE를 병렬로 64개 사용하면 한 사이클 당 하나의 스테이지를 계산할 수 있으며 이 때 64 비트가 생존자 경로 메모리 블록에 저장된다. 생존자 경로 메모리는 ACS에서 연

산된 결과 값들을 저장하였다가 복호 깊이에 해당하는 시간이 지나면 역추적 알고리즘에 따라 원래의 데이터를 복호한다. 일정한 시간이 지난 후 유일한 생존자 경로가 나타나게 되는데 이것을 복호 깊이라 한다. 보통 7/8로 puncturing을 할 경우 복호 깊이는 96 이상이 되어야 BER 성능이 저하되지 않는다. BER 측정 블럭은 복호된 데이터를 송신부와 동일한 부호기를 사용해 다시 부호화 한 뒤 수신단에 입력된 데이터와 비교해 다른 값일 경우 내부 계수기를 1씩 증가시켜 채널 상태를 감시한다.

3. Puncturing 회로

길쌈 부호기에서 출력된 데이터의 전송률은 부호율이 1/2이므로 입력 데이터에 비해 2배 높다. 이 경우 전송 데이터의 정보 전송률이 낮기 때문에 이를 높이기 위해 일정한 주기에 따라 특정 비트의 데이터를 지워버리는 puncturing 기법을 사용한다. 부호율 7/8로 punctured 코딩을 하면 약 2dB 가량 부호 이득이 저하되지만 부호율 7/8인 길쌈 부호기와 비터비 복호기를 설계하는 것에 비해 하드웨어 부담 증가가 적은 장점을 갖는다. 설계된 puncturing 블럭은 병렬로 데이터를 입력받아 내부에서 직렬로 변환해 처리한 후 다시 병렬로 바꾸어 출력하는 방식을 사용하였다.

그림 3에 부호율 2/3인 punctured 코드의 생성 예를 나타내었다. 입력 데이터 속도는 10 Mbps라고 가정한다. 시간 t0, t1에서 데이터 D0, D1이 길쌈 부호기에 입력되면 4개의 부호어 C00, C01, C10, C11이 출력되는데 puncturing 블럭에서는 이 중 한 비트를 일정한 삭제 패턴(deleting map)에 의해 삭제한다.

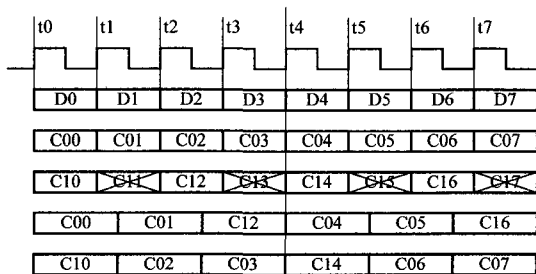


그림 3. 부호율 2/3 Punctured 코딩  
Fig. 3. The code rate 2/3 punctured coding.

이 예제에서는 삭제 패턴이 1110으로 네 번째마다 데이터를 삭제하게 되어 C11이 삭제된다. 삭제된 데이터는 전송되지 않으므로 D0, D1의 2 비트 입력 시

C00, C10, C01의 순서대로 3 비트만이 출력되어 부호율 2/3인 코드가 생성된다. 이 때의 출력 데이터 전송 속도는  $10 \times 3/2$  인 15 Mbps가 된다. 이와 같은 기법을 사용하여 제안한 비터비 복호기에서는 1/2에서 7/8까지 부호율을 변경할 수 있다.

4. ACS 아키텍처

설계된 ACS 블럭은 64개의 PE를 병렬 처리 방식으로 연결하였다. 각 PE에서는 격자도 상의 상태들 중 하나의 상태를 처리하며 ACS 연산을 수행해 각 상태로 입력되는 현재 경로값들과 누적 경로 값들을 더한 값을 비교해 작은 값을 갖는 경로를 선택한다. BMC 블럭에서 3 비트로 입력된 데이터로부터 현재 경로값을 구한다. 한 스테이지에  $2^{K-1}$ 개의 상태가 존재하므로 64개의 PE를 사용하면 매 클럭 사이클마다 격자도 상의 한 스테이지를 처리할 수 있다. 누적 경로 값들을 더하는 과정에서 사용되는 가산기의 입력 비트 수가 8 비트로 제한됨에 따른 오버플로우로 인해 발생하는 연산 에러를 막기 위한 정규화 블럭 및 rescaling 블럭을 설계하였다. 이들 블럭에서는 상태 값들의 오버플로우를 방지하기 위해 사용되는 정규화 블럭에서는 각 상태들의 MSB를 검사해 MSB가 1일 때 미리 정해진 값을 감하도록 설계하였다. PE 블럭과 정규화 블럭은 파이프라인을 통해 연결되어 최대 임계시간을 줄이도록 하였다. 또한 PE 내부에 사용되는 덧셈기를 고속으로 설계하여 동작속도를 높이도록 하였다.

본 논문에서 설계한 방법은 64개의 PE를 병렬로 연결하여 ACS 연산을 수행하므로 연산속도는 향상되지만 PE들 사이의 연결이 매우 복잡해진다. 이러한 문제점을 해결하기 위해 PE 사이의 연결을 인접하는 것들로 제한하는 방법을 사용하여 결선에 소모되는 면적을 줄였다. PE의 구성은 8 비트 가산기 2개, 8비트 비교기와 MUX로 구성된다. 구현한 ACS 블럭의 최대 동작 속도는 PE의 연산속도에 의해 결정된다. PE의 구성에 사용되는 덧셈기는 고속 동작이 가능한 CLA 방식을 사용하였고 덧셈기, 8비트 비교기와 8비트 2x1 MUX를 통과한 지연시간이 최대 임계경로가 되므로 70MHz 이상의 고속 동작이 가능하다.

5. 생존자 경로 메모리

생존자 경로 메모리는 ACS로부터 선택된 경로들로

부터 원래의 데이터를 복호하며 역추적 방식과 레지스터 교환 방식이 주로 사용되고 있다. 역추적 방식은 현재 상태에서부터 이전 상태값을 순차적으로 추적해 가는 방식으로 하드웨어를 적게 사용할 수 있지만 구성장의 길이가 클수록 데이터 복호에 많은 시간이 걸려 높은 전송속도를 얻을 수 없다. 따라서 하드웨어 크기는 증가하지만 고속 동작을 위해서 레지스터 교환 방식의 사용이 필수적이다. 그림 4는 레지스터 교환방식의 PE 구조를 나타낸다.

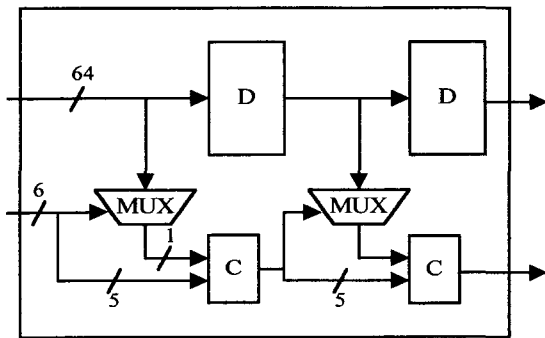


그림 4. 레지스터 교환 방식의 PE 아키텍처  
Fig. 4. The PE structure using the register exchange method.

ACS에서 연산된 64 개의 결정 비트(decision bit) 값들은 복호 깊이 만큼의 레지스터에 저장된다. 복호 깊이가 이전 시간에는 유효하지 않은 데이터가 출력되며 복호 깊이가 지난 후에 올바른 데이터가 출력된다. 그림 5와 같이 레지스터 교환 방식은 레지스터와 MUX 들로 구성되며 첫 번째 MUX의 선택 비트 입력으로는 0을 인가하도록 설계되었다.

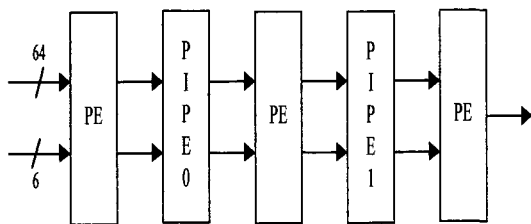


그림 5. Pipeline을 적용한 레지스터 교환 아키텍처  
Fig. 5. The TB structure using the pipelined register exchange method.

ACS로부터 출력된 64 비트의 결정 비트들은 6 비트로 구성된 MUX 선택 비트에 따라 한 비트가 출력된다. 6 비트의 선택 비트 중 LSB를 제외한 5 비

트와 MUX로 출력된 1 비트가 C(combine) 블록을 통해 6 비트로 재구성 된 후 다음 단 MUX의 선택 비트로 입력된다. 유효한 데이터가 출력되기 위해서는 복호 깊이에 해당하는 개수만큼의  $64 \times 1$  MUX 단을 거쳐야 한다. 본 논문에서는 복호 깊이를 96 으로 설계하였으므로 96개의  $64 \times 1$  MUX 지연 시간을 거친 후 데이터가 출력되게 된다. 그 결과, 시간 지연이 커지게 되어 낮은 데이터 출력 속도를 얻게 된다. 따라서 고속 동작을 실현하기 위해 파이프라인 단을 추가함으로써 MUX 단의 임계 경로(critical path)를 원하는 지연 시간으로 제한하였다. 즉 하나의 PE에 사용되는 MUX수를  $64 \times 1$  MUX 2단으로 제한함으로써 지연시간을 줄이도록 하였다. 따라서 생존자 경로를 구하기 위한 최대 지연 시간은 PE 내부의 MUX 2단을 통과하는데 소요되는 시간에 의해 결정된다. 그림 5는 파이프라인 단이 추가된 레지스터 교환 아키텍처를 나타낸다. 이 때의 임계 경로는 파이프라인 스테이지 사이의 지연 시간에 의해 결정된다. 시뮬레이션 결과 worst-case 시뮬레이션에서 70 MHz로 동작함을 확인하였다.

### III. 비터비 복호기 구현 및 성능평가

이 장에서는 비터비 복호기의 설계, 구현 및 성능 평가에 대해 기술한다. Top-down 설계 방식에 따라 Verilog HDL을 이용하여 동작 및 아키텍처 모델을 설계하였으며 각 블록별 최적화를 수행하여 게이트 수를 감소시켰다. CADENCE™ CAD 툴을 이용하여 기능 검증 시뮬레이션과 타이밍 시뮬레이션을 수행하였다. Verilog 모델을 바탕으로 SYNOPSIS™ CAD 툴의 Design Analyzer를 사용하여 논리합성과 최적화를 수행하였다. 논리 합성은  $0.6 \mu\text{m}$  SOG 라이브러리(KG75000)를 이용하여 수행하였다. 구현된 비터비 복호기의 전체 게이트 수는 약 100,400 게이트이며 시뮬레이션 주파수는 worst case에서 70 MHz로 동작한다.

설계된 비터비 복호기의 성능을 평가하기 위해 통신용 알고리즘 시뮬레이션 툴인 COSSAPT™을 이용해 갈쌈 부호기와 AWGN(Additive White Gaussian Noise) 채널 모델링을 수행하였다. 채널을 통과한 데이터를 시뮬레이션을 통해 이진 파일로 저장하여 에러가 포함된 데이터를 얻은 뒤 이를 Verilog 테스트벡

터로 변환하였다. 변환된 테스트 벡터를 통해 기능검 증 시물레이션과 타이밍 시물레이션을 수행하여 설계 한 모델의 타당성을 검증하였다. CADENCETM 툴 을 사용한 시물레이션 결과는 그림 6에 나타내었다.

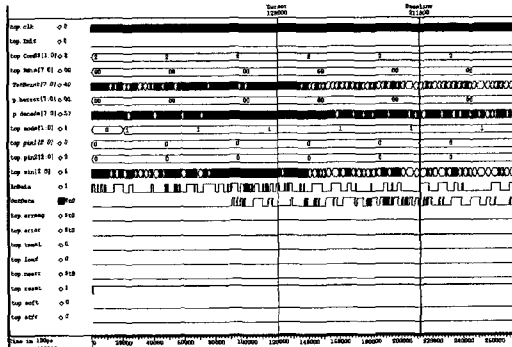


그림 6. 구현한 비터비 복호기의 시물레이션 파형  
Fig. 6. The simulation result of the implemented Viterbi Decoder.

현재 상용화 된 비터비 복호기 칩으로는 QUAL- COMM사의 Q1900, MOTOROLA사의 MC92300, LSI LOGIC사의 L64709 등이 있다. 이 중 동작 속도가 가장 빠른 것은 L64709 칩으로 최대 동작 속도는 60 MHz이다. 본 논문에서 구현한 비터비 복호기의 사양은 구속장 7과 부호화를 1/2이며 puncturing을 통해 2/3, 3/4, 5/6, 7/8의 다양한 부호율을 지원한다. 이는 위에서 언급한 상용 칩의 사양과 거의 동일하다. 그러나 본 논문에서 설계한 복호기는 레지스터 교환 블럭의 파이프라인 단수를 증가시킴으로써 데이터 복호 시간을 줄였다. 설계된 모델의 최대 동작 속도는 70 MHz로 기존 상용화 칩보다 빠르게 동작함을 알 수 있다. 표 2에 기존 칩들과의 비교 결과를 나타내었다.

표 2. 기존 비터비 칩과의 성능 비교  
Table 2. Performance comparisons with existing Viterbi chips.

사양	Q1900	MC92300	L64709	구현한 비터비 복호기
구속장	7	7	7	7
부호율	1/2	1/2	1/2	1/2
동작 주파수	25MHz	37MHz	60MHz	70MHz
Puncture d Mode	2/3,3/4,5/6,7/8	2/3,3/4,5/6,7/8	2/3,3/4,5/6,7/8	2/3,3/4,5/6,7/8
연관정 비트수	3 비트	3 비트	3 비트	3 비트

#### IV. 결 론

본 논문은 병렬처리 방식과 파이프라인 아키텍처를 적용한 비터비 복호기의 아키텍처와 이의 설계 및 구현을 제시하였다. 기존 상용 비터비 칩들의 사양을 비교 분석하여 멀티미디어 서비스를 지원하기 위해 적합한 사양을 결정하였다. 데이터 처리 속도를 높이기 위해 PE 들을 병렬로 연결한 ACS 아키텍처와 파이프 라인 기법을 적용한 레지스터 교환 방식으로 생존자 경로 메모리 블럭을 구성하였다. 그 결과 worst case 시물레이션을 통해 70 MHz로 동작함을 확인하였으며 이는 기존 칩들보다 동작속도 면에서 우수하다. 또한 punctured mode를 통해 1/2, 2/3, 3/4, 5/6, 7/8의 다양한 부호율을 지원하며 3 비트 연관정을 통해 2 dB의 추가적인 부호 이득을 얻는다. 설계된 비터비 복호기는 LMDS, 유선 CATV망을 이용한 VOD 서비스, DBS(Direct Broadcast Satellite)를 통한 디지털 TV 수신기 등에 다양하게 사용될 수 있다. 앞으로 설계한 모듈의 아키텍처 개선과 회로의 최적화를 통해 게이트 수를 줄이는 연구를 수행할 것이다.

#### 참 고 문 헌

[1] Stephen B. Wicker, Error Control Systems for Digital Communication and Storage, Prentice-Hall, 1995.

[2] Gennady Feygin and P. G. Gulak, "Architectural Tradeoffs for Survivor Sequence Memory Management in Viterbi Decoders," *IEEE Trans. on Communications*, vol. 41, no. 3, March 1993, pp. 425-429.

[3] Montse Boo, Francisco Arguello, Javier D. Bruguera, Ramon Doallo, and Emilio L. Zapata, "High-Performance VLSI Architecture for the Viterbi Algorithm," *IEEE Trans. on Communications*, vol. 45, no. 2, Feb. 1997, pp. 168-176.

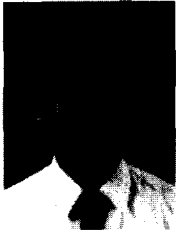
[4] Emmanuel Boutilion, Nicolas Demassieux, "High Speed Low Power Architecture for Memory Management in a Viterbi Decoder," in *Proc. International Symposium on Circuits and Systems (ISCAS'96)*, Atlanta, USA,

- May 1996, pp. 284-287.
- [5] C. Bernard Shung, Horng-Dar Lin, Robert Cypher, Paul H. Siegel, and Hemant K. Thapar, "Area-Efficient Architectures for the Viterbi Algorithm-Part I: Theory," *IEEE Trans. on Communications*, pp. 636-6433, vol. 41, April, 1993.
- [6] Shuji Kubota, Shuzo Kato, and Tsunehachi Ishitani, "Novel Viterbi Decoder VLSI Implementation and its Performance," *IEEE Trans. on Communications*, pp. 1170-1178, vol. 41, No. 8, Aug., 1993.
- [7] Horng-Dar Lin, David G. Messerschmitt, "Parallel Viterbi Decoding Methods for Uncontrollable and Controllable Sources," *IEEE Trans. on Communications*, vol. 41, no. 1, Jan. 1993, pp. 62-69.
- [8] David Haccoun, Guy Begin, "High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding," *IEEE Trans. on Communications*, vol. 37, no. 11, Nov. 1989, pp. 1113-1124.
- [9] Qualcomm Inc., *Forward Error Correction Products Data Book*, Q1900.
- [10] Motorola Inc., *Motorola Semiconductor Technical Data*, MC92300.
- [11] LSI Logic Inc., *LSI Logic Technical Data*, L64709.

---

 저 자 소 개
 

---



李秉徹(正會員)

1997. 2 아주대학교 전자공학 학사.  
 1999. 2 아주대학교 전자공학 석사.  
 1999. 3~현재 기인텔레콤(주) 연구소  
 주관심분야는 통신 및 신호처리  
 용 ASIC 설계

鮮于明勳(正會員)

1980. 2 서강대학교 전자공학 학사. 1982. 2 한국과학기술원 전자공학 석사. 1982. 3~1985. 8 한국전자통신연구소(ETRI) 연구원. 1985. 9~1990. 8 Univ. of Texas at Austin 전자공학 박사. 1990. 8~1992. 8 Motorola, DSP Chip Division(미국). 1992. 8~1996. 10 아주대학교 전기전자공학부 조교수. 1996. 10~현재 아주대학교 전기전자공학부 부교수. 주관심분야는 VLSI 및 Parallel Architecture, 통신, 영상 및 신호처리용 ASIC 설계