천공 부호를 지원하는 Viterbi 복호기의 면적 효율적인 생존자 경로 계산기 설계

정회원 김 식*, 황 선 영*

Design of an Area-Efficient Survivor Path Unit for Viterbi Decoder Supporting Punctured Codes

Sik Kim*, Sun-young Hwang* *Regular Members*

요 약

천공 부호를 지원하는 비터비 복호기는 하드웨어 복잡도를 유지하는 선에서 부호율을 효율적으로 높일 수 있지만 충분한 BER 성능을 얻기 위해 복호 지연 시간이 길어지고 생존자 메모리의 크기가 늘어나는 단점이 있다.

본 논문은 비터비 복호기의 메모리 소요량을 줄이는 파이프라인화된 순방향 추적기를 포함하는 생존자 경로 계산기를 제안한다. 제안된 생존자 경로 계산기는 역추적에 필요한 초기 복호 지연을 없애고, 경로 계산을 위한 순방향 추적 과정을 가속함으로써 생존자 메모리의 사용량을 감소시킨다. 실험 결과, 제안된 비터비 복호기의 생존자 계산기는 기존의 혼성 생존자 경로 계산기에 비해 약 16% 면적이 감소함을 확인하였다.

Key Words : Viterbi 복호기, 면적 최적화, 생존자 경로 계산기

ABSTRACT

Punctured convolutional codes increase transmission efficiency without increasing hardware complexity. However, Viterbi decoder supporting punctured codes requires long decoding length and large survivor memory to achieve sufficiently low bit error rate (BER), when compared to the Viterbi decoder for a rate 1/2 convolutional code.

This paper presents novel architecture adopting a pipelined trace-forward unit reducing survivor memory requirements in the Viterbi decoder. The proposed survivor path architecture reduces the memory requirements by removing the initial decoding delay needed to perform trace-back operation and by accelerating the trace-forward process to identify the survivor path in the Viterbi decoder. Experimental results show that the area of survivor path unit has been reduced by 16% compared to that of conventional hybrid survivor path unit.

Ⅰ. 서 론

오류 정정이 가능한 채널 부호화 시스템은 90년대 이후 다양한 응용분야에서 연구되어 실용화되었다. 디지털 DBS나 HDTV 같은 응용 분야의 채널 변복조 기법으로 전력 효율을 높이기 위한 TCM (Trellis Coded Modulation) 기법이 채용되었다. TCM은 변조기법과 오류 정정 부호로써의 길쌈부호를 결합한 것으로 대역폭의 변화 없이 기존 변복조 방식보다 전력 효율상 이득을 얻을 수 있으며[1] 복조기(Demodulator)와 Viterbi 복호기로 구성된다. Viterbi 복호기는 연집오류 정정 능력이 뛰어난 RS

*서강대학교 전자공학과 CAD & Computer System 연구실 (hwang@ccs.sogang.ac.kr)
논문번호 : 030259-0618, 접수일자 : 2003년 6월 17일
※본 연구는 대학 IT 연구센타 육성 지원 사업의 연구 결과로 수행되었습니다.

복호기와 함께 순방향 오류 정정 (Forward Error Correction, FEC) 복호기로 널리 사용되었으며[1], IS-95와 같은 이동 통신의 채널 부호화에도 길쌈 부호와 Viterbi 복호기가 채택되었다. 현재 서비스 사업 초기 단계인 IMT-2000의 경우에도 32Kbps 이하의 저속 음성 통신에는 길쌈 부호(Convolution codes)를 채택하고 있다[2].

Viterbi 복호기는 최대 유사 복호(Maximum likelihood decoding) 방식으로[3], 송신단과 수신단 사이에 발생한 차이를 나타내는 척도로 수신된 신호와 예상 가능한 모든 경우의 차이를 나타내는 가지 메트릭을 계산하고 해당 길쌈 코드가 생성할 수 있는 모든 상태에 대하여 이 차이를 누산하는 경로 메트릭을 생성하여 일정 시간이 경과한 후 가장 작은 차이를 가지는 생존자 경로 (Survivor path)를 계산하고 복호 출력을 결정한다. Viterbi 복호기는 매 복호 단계에서 모든 상태에 관한 생존자 경로 및 복호 정보를 기억하고 이를 갱신, 복호 출력을 생성하므로 메모리 사용량과 접근 빈도가 높은 구조적 특성을 나타낸다. 또한, 복호 성능의 향상을 위해서 제한 길이 (Constraint length)를 증가시켜 입력간의 상관관계를 높여주는 경우 제한 길이의 증가에 따라 회로 복잡도와 메모리 사용량이 지수적으로 증가하게 되며, 신호 전송 효율을 증가시키기 위한 천공 코드(Punctured code)를 적용하는 경우 생존자 경로 정의에 필요한 최소시간이 천공하지 않은 경우의 두 배 이상에 해당하여 메모리 사용량이 더욱 증가하는 문제점이 있으므로 생존자 경로 계산 방식의 최적화를 통한 메모리 면적 감소가 필수적이다[4].

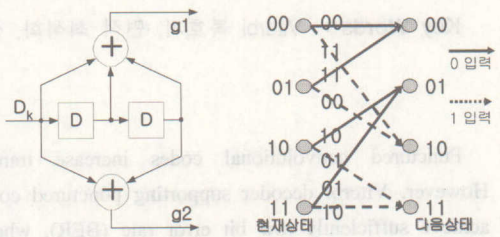
본 논문에서는 Viterbi 복호기의 생존자 경로 계산기 구조와 메모리 요구량, 접근빈도에 대한 분석을 토대로 면적 효율적인 Viterbi 복호기 구조를 제안한다. 제안된 Viterbi 복호기는 파이프라인화된 순방향 추적기를 사용한 혼성 생존자 경로 계산기를 적용하여 생존자 경로 계산시 초기 복호 지연을 제거하고 복호에 필요한 순방향 추적 과정을 가속함으로써 생존자 경로 메모리의 요구량과 단위 시간당 필요한 메모리 접근 빈도를 낮춤으로써 생존자 경로 계산기의 면적을 효율적으로 감소시킨다.

Ⅱ. 길쌈 부호기와 Viterbi 복호기 구조

1. 길쌈 부호기

길쌈 부호화 기법은 구현이 간단하고 상대적으로 높은 부호 이득을 얻는 장점으로 순방향 오류 정정 복호 기법에 널리 채택되어 왔다[5]. 길쌈 부호는 현재 입력과 저장된 과거 입력의 modulo-2 가산을 통해 전송되는 데이터간의 상관관계를 높여줌으로써, 수신단에서 오류를 정정하는 기법을 사용한다.

길쌈 부호기는 m개의 입력을 이용 n개의 출력을 생성하는 방법으로 부호율(Code Rate)은 m/n으로 정의된다. 부호화에 사용되는 과거 입력의 개수가 많을수록 상관관계가 높아지는 특성으로 길쌈 부호에 사용되는 데이터 길이를 제한길이(Constraint Length) K로 정의하고 성능 척도로 사용된다. 그림 1은 부호율 1/2, 제한길이 K = 3인 길쌈 부호기를 나타낸다. 생성 다항식(Generator Polynomial)은 $g1 = 111_2$, $g2 = 101_2$로 modulo-2 가산기 입력 위치를 나타낸다. 그림 1 (a) 에 부호기 구조도를 그림 1 (b)에 해당 부호기의 trellis 도를 각각 표시하였다. Trellis 도는 부호기의 상태도를 시간 축 진행에 따른 함수형식으로 표현한 것으로 각 가지(Branch)에 표시된 값은 상태 천이시 출력값을 의미한다.



(a) (b)

그림 1. 부호율 1/2, 제한길이 K = 3 길쌈 부호기. (a) 부호기 구조, (b) Trellis 도.

2. Viterbi 복호기

Viterbi 복호 알고리듬은 길쌈 부호에 대한 최대 유사 복호(Maximum Likelihood Decoding)를 수행하며[1][3], 송신단의 trellis와 수신 신호간에 발생한 오차를 나타내는 척도로 가지 매트릭(Branch Metric)과 경로 매트릭(Path Metric)을 사용한다. 가지 매트릭은 trellis도의 발생 가능한 모든 출력과 수신 신호간의 차이를 나타내며 가우시안 채널 모델을 사용할 경우 Euclidian distance로 구해진다. 경로 매트릭은 trellis 도의 시간축을 따라 각 상태별로 입력되는 두 개의 경로 매트릭에 가지 매트릭

을 합산하여 최소값을 갖는 쪽을 생존자 경로(Survivor Path)로 설정하고, 그 값을 경로 매트릭으로 갱신한다. 충분한 시간 후에 생존자 경로를 역추적할 경우 다수개의 생존자 경로는 하나로 통합되며 이 경로값이 복호 결과로 사용된다.

길쌈 부호의 낮은 부호율을 개선하기 위한 방법으로 천공 부호 기법이 도입되었다[6]. 천공 부호는 생성된 길쌈 부호에서 주기적으로 비트를 삭제하는 방식으로, 손실된 정보에 의해복호 길이가 길어지는 단점이 존재하나, 복호기 설계 과정에서 Viterbi 복호기의 구조에 변화를 주지 않고 전단에 전용 처리 회로를 부가함으로써 쉽게 길쌈 부호기의 부호율을 높일 수 있는 장점이 있다. 천공 부호는 생성된 2n 비트에서 m 비트를 미리 정의된 삭제도(Deleting Map)에 의해제거하는 방식으로, 주어진 제한 길이와 부호율에 대해 천공 부호에 의한 오류를 최소화하는 삭제도가 제안되어 있다[6]. 복호시에는 삭제 위치에 더미 비트를 삽입하여 입력의 개수를 복원한 뒤 일반적인 Viterbi 복호기와 같은 방식으로 복호한다. 일반적으로 복호 길이(Decoding Length)는 제한 길이의 다섯배 정도면 충분한 것으로 알려져 있으나[7], 천공 부호를 사용하여 정보가 일정 부분 소실된 경우 제한길이의 열 배 이상의 복호 길이를 필요로 하므로[8] 시스템이 요구하는 복호 성능과 하드웨어 복잡도 등을 고려하여 면적을 최적화하는 설계가 필요하다.

그림 2는 천공 부호를 지원하는Viterbi 복호기의 하드웨어 구조를 나타낸다. Viterbi 복호기는 수신 신호과 가능한 모든 입력에 대한 차이인 가지 매트릭을 계산하는 BM 계산기와 trellis 상의 모든 상태에 대해 경로 매트릭을 생성하는 ACS (Add Compare Select) 계산기, 생존자 경로를 계산하고 최대 유사 복호를 수행하는 생존자 경로 계산기로 구성되며, 천공 부호를 지원하기 위해 입력 부호 블록의 정확한 천공 부위를 찾아주는 동기화기와 동기 위치를 기준으로 삭제된 곳에 더미 입력을 삽입하고 수신 신호 블록을 복원하는 역 천공기(Depuncturing Unit)가 복호기 전단에 추가된다. 그림 2의 Ri 는 3비트로 양자화된 두 개의 수신 입력을 나타내며 입력 clock은 외부 신호 수신을 위한 clk1과 내부 처리를 위한 clk2로 구분된다. DR은 천공 부호를 복원한 입력을 의미하며 Sync1, Sync2는 회로의 동기 설정을 위한 신호이다. 경로 매트릭은 동기감지를 위한 입력으로 사용된다.

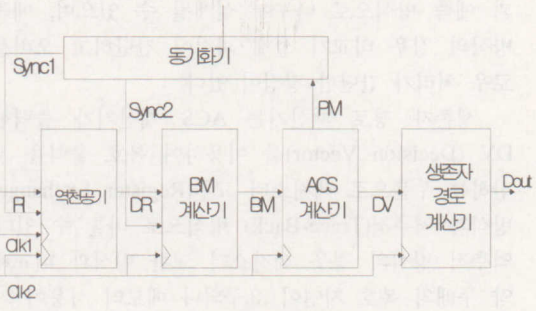


그림 2. 천공 부호를 지원하는 Viterbi 복호기 하드웨어 구조도.

동기화기는 천공 부호를 사용하는 복호기에서 필수적인 요소로써 동기 감지 회로는 오류 최소화를 위해 동기 신호를 빠르게 감지하고, 채널의 상태에 무관하게 동작할 수 있는 특징을 가져야 한다. 주로 오류 발생시 Viterbi 복호기의 동작 특성에 기초하여 동기 감지기를 구성하며, Comatlas 사에서 제안한 최소 경로 매트릭 증가분과 초소 가지 매트릭을 비교하는 법, Sony 사에서 제안한 오버플로우 방지를 위한 정규화(normalization) 빈도를 관찰하는 법 등이 보고되었다[9][10].

그림 2의 BM 계산기는 부호율 1/2 인 길쌈부호의 경우00, 01, 10, 11 과 수신 신호의 오차를 계산한다. 잡음이 존재하는 채널의 상태를 감안하여 수신된 신호는 이진 신호로 변환하지 않고, 신호 강도나 위치를 세분화하여 두 비트 이상 양자화된 입력으로 구성한다. 해당 입력 DR이 천공된 위치의 신호일 경우에는 수신 신호와 기준 값의 차이를 0으로 만들어준다.

ACS 계산기의 경로 매트릭 크기는 오버플로우 발생 빈도와 연산기 면적에 영향을 미친다. 오버플로우 발생시 기존의 경로 매트릭 간에 존재하는 크기 차이를 선형적으로 유지하면서 매핑하는 것이 불가능하며 일부 LSB 정보가 소실되어 성능 저하의 원인이 되므로 경로 매트릭의 크기를 크게 하면 오버플로우 발생 빈도가 떨어지고 복호 정정 능력을 이론치에 가깝게 끌어올릴 수 있으나, 비트 수 증가에 따라 하드웨어의 크기가 커지고 가산기와 비교기의 지연 증가로 인해 속도 저하가 일어날 수 있다. ACS 계산기는 피드백 구조로 인해 파이프라인이나 병렬화가 힘들며, 특정한 오버플로우 처리 방식에 대해 경로 메트릭의 크기를 변화시키면서 오차율을 측정하여 최적의 크기를 선택하는 방법을 사용한다. 오버플로우 처리기는 감지 후 처리 방식

과 예측 방식으로 나누어 설계될 수 있으며, 예측 방식의 경우 비교기 설계 과정이 간단하고 오버플로우 처리가 간단한 장점이 있다.

생존자 경로 계산기는 ACS 계산기가 출력한 DV (Decision Vector)를 이용하여 복호 출력을 계산하는 부분으로 레지스터 교환(Register Exchange) 방식과 역추적(Trace-Back) 방식으로 나눌 수 있다. 역추적 방식의 경우 레지스터 교환 방식과 비교해 약 두배의 복호 지연이 요구되나 메모리 사용이 가능하고 신호 천이 빈도가 낮아 실제 구현 단계에서 널리 사용되고 있다. 자세한 생존자 경로 계산기 구조는 3장에서 다루도록 한다.

Ⅲ. 생존자 경로 계산기 구조

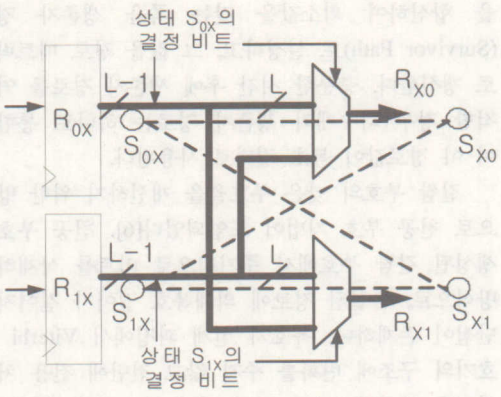
1. 기존 생존자 경로 계산기

일반적인 생존자 경로 계산기는 레지스터 교환 방법과 역추적 방법으로 구현 가능한 것으로 알려져 있다. 본 절에서는 각 생존자 경로 계산기에 대한 기억 소자의 소요량과 복호 방법을 살펴본다.

1.1. 레지스터 교환 방식

레지스터 교환 방식은 trellis 도를 통해 제공되는 배선 구조와 동일하게 레지스터 네트워크를 형성하여 시간 진행과 같은 방향으로 생존자 경로를 찾는 구조로, 소요되는 기억 소자의 수가 일반적인 역추적 방식의 1/2 ~ 1/4만큼을 필요로 하는 장점을 가지고 있으나, 매 시간 주기마다 모든 기억 소자의 내용을 갱신하는 동작 특성으로 인해 메모리를 이용한 설계가 불가능하고 전력 소모가 큰 단점이 있어 제한길이 K가 큰 응용 분야에서는 쓰이지 않고 있다[11].

그림 3에 레지스터 교환 방식의 생존자 경로 계산기 구조와 레지스터 배선 구조를 제시하였다. 레지스터 교환 방식을 사용할 경우 레지스터의 배선 구조는 ACS 계산기의 배선 구조와 동일하며, ACS 계산기의 선택 신호에 따라 해당 레지스터 내용을 갱신하게 된다. 레지스터의 크기는 복호 길이와 동일하며 MSB는 갱신과정에서 출력으로 사용되고 LSB에는 해당 상태의 결정 비트(Decision bit)이 입력되어 다음 상태에서 사용된다. 복호 길이에 해당하는 시간 진행 이후 레지스터의 MSB는 신뢰성 높은 복호 출력을 의미하며 모든 상태에서 그 값이 같아진다.



R의 크기 = 복호 길이 (L)

X : 상태표시에 필요한 공통 비트들

그림 3. 레지스터 교환 방식의 생존자경로 계산기.

1.2. 역추적 방식

역추적 방식은 C. Radar 에 의해 제안되었으며 [11], 복호 길이에 해당하는 시간 진행 후에 임의의 상태 혹은 최소 경로 매트릭을 갖는 상태에서 출발하여 생존자 경로를 시간의 역순으로 탐색하는 방법으로, 모든 생존자 경로가 하나의 경로로 병합되는 특성을 이용해 복호를 진행한다. 그림 4에 역추적 방식 생존자 경로 계산기 구조와 복호 진행과정을 제시하였다. 그림 4 (a)는 복호과정에 필요한 역추적 회로의 구조도를 나타낸다. 생존자 경로의 병합을 위한 역추적과정과 복호에 필요한 읽기 포인터들과 해당 상태의 결정비트들을 저장하는 메모리로 구성되며, 구현 방식에 따라 필요한 메모리 뱅크의 수는 변화한다. 그림 4 (b)는 시간 진행에 따른 역추적 방법을 trellis 도에서 나타내었다. 그림 4 (b)의 시간 축은 두 방향으로 구성되어 있으며, 위쪽 시간 축은 ACS 출력 DV(Decision Vector)의 저장 과정을, 아래 쪽의 시간 축은 역추적 레지스터와 MUX를 이용한 복호에 필요한 시간을 의미한다. 굵은 화살표로 표시된 경로는 레지스터와 MUX에 의해 선택된 이전 상태를 포인터 형식으로 나타낸 것이며, 선택된 상태에 의한 복호 진행을 나타낸다. 가는 화살표의 경우는 실제 복호 과정에서 선택되지 않은 가상의 생존자 경로를 의미하며 이 경우에도 복호 진행에 따라 모든 경로들이 하나의 경로로 수렴되고 있음을 알 수 있다.

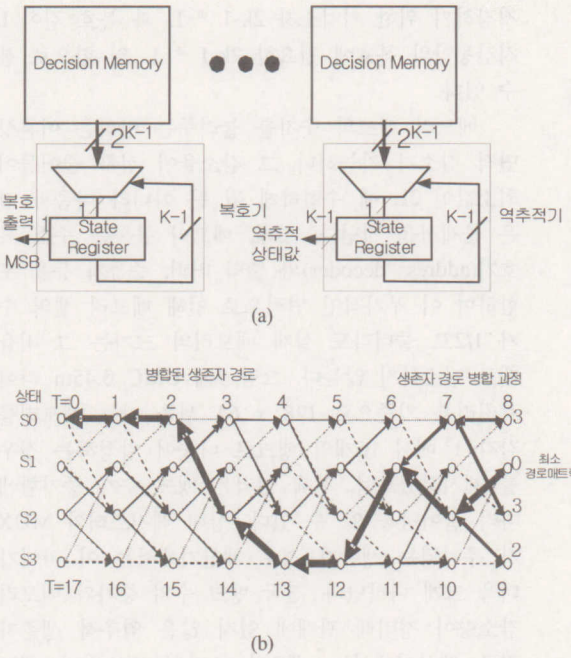


그림 4. 역추적 방식 생존자 경로 계산기. (a) 역추적 생존자 경로 계산기 구조, (b) Trellis 상에서 역추적 방식에 의한 복호 진행 과정.

생존자 경로 계산은 이론적으로 무한 시간 동안 진행되므로 제한된 메모리를 재사용하기 위한 읽기 및 쓰기 방법 구성에 의해 다양한 형태의 구현이 가능하며, 복호 지연의 크기와 읽기 및 쓰기 속도 차이, 포인터의 개수에 따라 k-pointer even, k-pointer odd, one-pointer 방식으로 나누어 필요한 메모리 크기와 메모리 개수를 계산할 수 있다 [12][13][14]. 이 때 생존자 경로 계산기의 메모리 동작은 다음 세 가지의 읽기 및 쓰기 동작으로 구분된다.

WD (Write Data) - ACS 계산기에서 발생하는 DV의 저장과정

TBR (Trace-Back Read) - 생존자 경로 수렴에 필요한 역추적 과정.

DR (Decode Read) - 수렴된 생존자 경로에서 복호 출력을 읽는 과정.

메모리 읽기 과정은 MUX와 레지스터로 구성된 포인터 개념으로 설명할 수 있으며 포인터 개수에 따라 메모리 크기와 뱅크 수가 달라진다. 참고문헌 [13]에서 제시한 k-pointer even 방식의 경우 역추적과 복호에 필요한 읽기 포인터의 개수를 k로 설정하고 읽기 및 쓰기를 분리하여 진행한다. 포인터 수를 3 으로 설정하고 길이가 L/2 인 메모리 뱅크를 사용할 경우 전체 메모리 사용량은 3L 에 해당하며 역추적 방식의 생존자 경로 계산기 가운데 가장 제어가 간단한 경우이다. 복호와 쓰기 과정을 한 메모리 뱅크에서 동시에 진행하는 k-pointer odd 방식의 경우[14] 포인터를 3으로 설정하는 경우 전체 메모리 크기는 2.5L 로 감소하나 시간 진행에 따라 메모리의 읽기, 쓰기 방향이 반복적으로 변하게 되므로 제어 장치의 설계가 k-pointer even 방식에 비해복잡한 단점이 있다. 쓰기 및 읽기의 속도가 1 대 3으로 서로 다른 경우인 1-pointer 방식의 경우[12] 전체 소모량은 2L 로 기타 방식들에 비해 가장 적으나 메모리 접근(access) 속도가 다르므로 전체 시스템의 동작 속도가 메모리 읽기 속도의 1/3으로 제한되는 단점이 있어 완전 병렬 구조에서 사용이 힘들며, 다중 프로세서 및 분산 메모리 환경에서의 소프트웨어 구현에 적합한 것으로 알려져 있다.

1.3. 혼성 생존자 경로 계산기

혼성 생존자 경로 계산기는 레지스터 교환 및 역추적 방식을 결합한 구조로 [15], 생존자 경로 병합을 위한 TBR 과정을 레지스터 교환 방식을 응용한 순방향 추적기 구조로 대체함으로써 메모리 뱅크 수와 포인터 개수의 증가 없이 복호기의 메모리 소요량을 감소시킬 수 있다. 그림 5 는 혼성 생존자 경로 계산기의 구조를 나타낸다. 그림 5의 순방향 추적기는 레지스터 교환 방식을 응용하며, 각 상태의 레지스터 크기는 L 이 아닌 K-1 비트로 고정되며 초기값으로 각 상태의 이진 표현이 할당된다. 각 레지스터는 복호 길이에 해당하는 L 시간 동안 새로운 비트의 추가 혹은 쉬프트 연산 없이 ACS 계산기의 결정에 따라 두 개의 입력 상태 중 하나의 값으로 갱신된다. 복호 길이 이후 각 레지스터는 수렴된 하나의 생존자 경로에 해당하는 상태 값을 공통적으로 가지게 되며 이는 TBR 동작과 동일한 결과를 의미한다. 순방향 추적기에 의해 TBR 동작은 생략되며, 복호 길이에 해당하는 L 시간마다 복호 시작에 필요한 생존자 경로의 상태를 역추적 레지스터에 제공한다. 역추적 레지스터는 순차적인 메모리 읽기 과정을 통해 해당 블록의 복호 결과를 추출하게 된다.

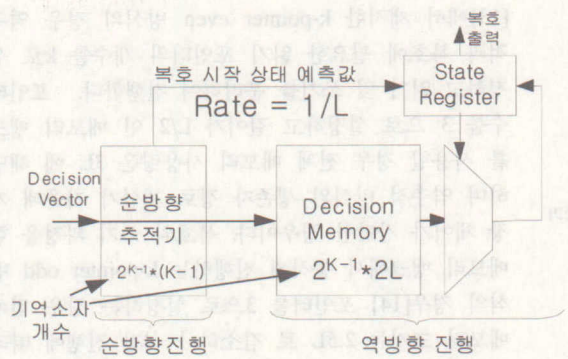


그림 5. 혼성 생존자 경로 계산기 구조.

혼성 생존자 경로 계산기의 메모리는 WD와 DR의 두 동작만을 수행하게 된다. 그림 6은 읽기 및 쓰기 뱅크를 분리한 방식으로 전체 소요 메모리는 3L 에 해당하며, 읽기와 쓰기를 한 뱅크에서 병행하는 형태로 전환할 경우 소요량은 2L로 줄어들 수 있다. 혼성 생존자 경로 계산의 경우 TBR이 생략되어 읽기 및 쓰기의 속도가 동일하므로 1-pointer 방식과 비교는 불필요하다.

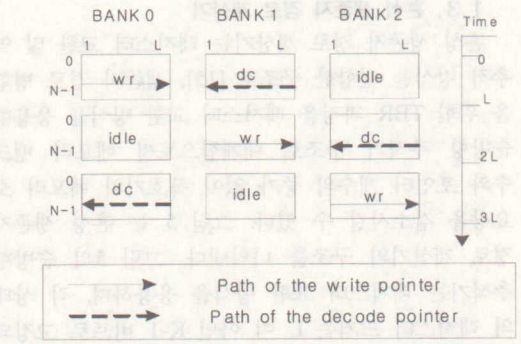


그림 6. 혼성 생존자 경로 계산기의 메모리 구성 및 동작.

2. 파이프라인 구조의 순방향 추적기와 생존자 경로 계산기

2.1. 메모리 구성방식과 면적의 상관 관계

앞 절에서 기술한 기존의 생존자 경로 계산기는 메모리 뱅크 수와 읽기 포인터의 개수를 늘려주는 형태의 파이프라인 기법을 도입하거나, 읽기 및 쓰기 동작의 속도를 다르게 함으로써 메모리 소요량을 줄일 수 있다. 각 방식에서 이론적으로 얻을 수 있는 메모리 크기의 최소값은 2k-1 * 2L 이며, 이는 역추적을 위한 초기 복호 지연 시간 동안 DV를 저장하기 위한 기억소자 2k-1 * L 과 복호 길이 L 시간동안의 복호에 필요한 2k-1 * L 의 합으로 볼 수 있다.

메모리 뱅크의 숫자를 늘려주는 방식은 이론상 면적 감소가 가능하나 그 감소율이 점차 줄어들어 최소값인 2L 에 수렴하게 될 뿐 아니라 다음과 같은 문제점을 지닌다. 첫째, 메모리 블록은 주소 복호기(address decoder)와 출력 버퍼, 증폭기 등을 포함하며 이 부가적인 면적으로 인해 메모리 셀의 수가 1/2로 줄더라도 실제 메모리의 크기는 그 비율대로 감소하지 않는다. 그림 7에 UMC 0.45m 라이브러리를 기준으로 128 x 64 셀을 갖는 메모리를 각각 1 에서 16개의 뱅크로 나누어 합성하는 경우를 제시하였으며, 전체 면적은 뱅크 수가 증가함에 따라 늘어남을 알 수 있다. 상태 레지스터와 MUX가 추가되는 생존자 경로 계산기에서는 이 차이가 더욱 크게 나타난다. 결국 뱅크 수의 증가와 메모리 감소량이 정비례 관계에 있지 않은 역추적 생존자 경로 계산기에서는 메모리 소자의 감소량과 뱅크 수 증가에 따른 면적 증가분을 비교하여 적정한 선에서 뱅크 수를 제한하여야 한다. 둘째, 뱅크 수의 증가는 메모리에 인가되는 주소의 용량 부하를 증가시킨다. 메모리는 전기적 특성 상 뱅크 별로 간격을 확보하여야 하며 복호기의 다른 부분을 메모리 사이에 배치하는 방식을 택해야 하므로[16] 뱅크 수가 늘어나면 배치, 배선에 어려움이 증가하고 주소 공급을 위한 전역 배선의 증가로 용량 부하가 늘어나므로 실제 구현 가능성은 제한된 범위에서 가능하다.

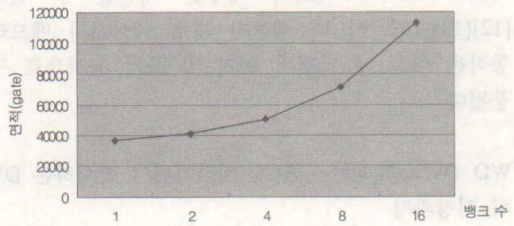
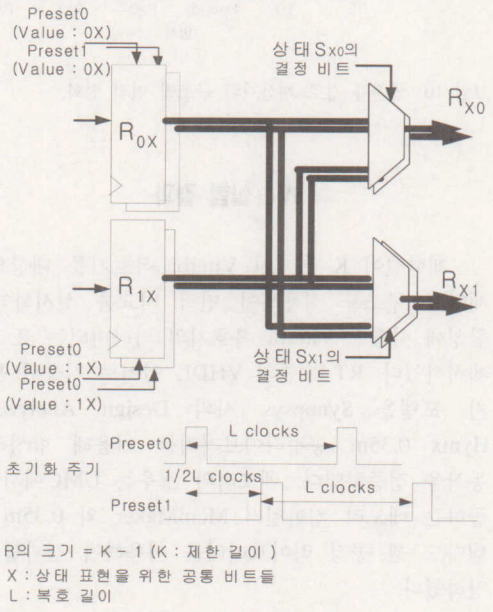


그림 7. 128 x 64 메모리의 구성방식과 면적 비교.

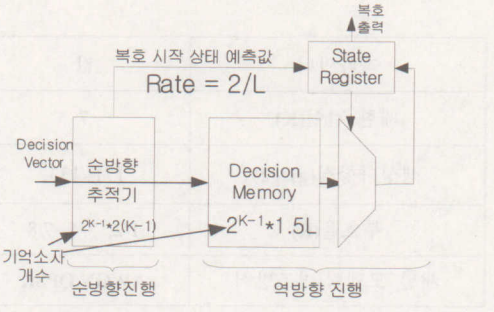
2.2. 파이프라인화된 순방향 추적기 도입

혼성 생존자 경로 계산기는 레지스터 교환 방식의 장점인 순방향 복호 진행을 통해 TBR 동작에 필요한 시간을 없애줌으로써 기존의 방식보다 작은 메모리 크기를 얻을 수 있으나, 역추적 방식에 요구되는 2L x 2K-1 메모리 소요 한계를 극복하지 못

하였다. 이는 TBR 동작을 대체하는 레지스터 교환이 L 시간을 기준으로 주기적으로 일어나기 때문에 메모리 뱅크의 크기가 L로 고정되어 다양한 계산기 구조의 형성이 불가능하기 때문이다. 제안된 생존자 경로 계산기 구조는 이러한 문제점을 바탕으로 레지스터 교환에 의한 TBR 과정을 파이프라인화 함으로써, 메모리 구현의 다양한 가능성을 살리고 초기 복호 지연에 의한 L 크기의 메모리 소요를 감소시킴으로써 역추적 방식의 장점인 저 전력 특징을 유지하는 선에서 메모리 소요를 2L 이하로 감소시키도록 하였다.



(a)

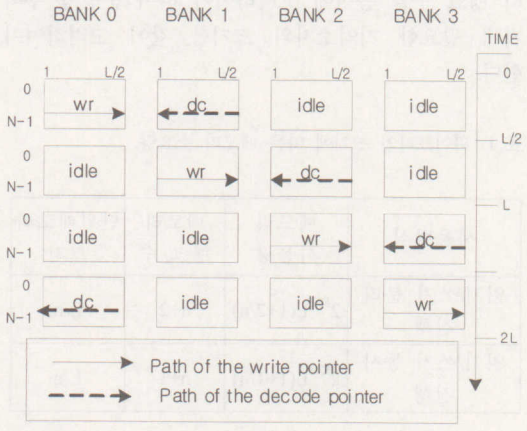


(b)

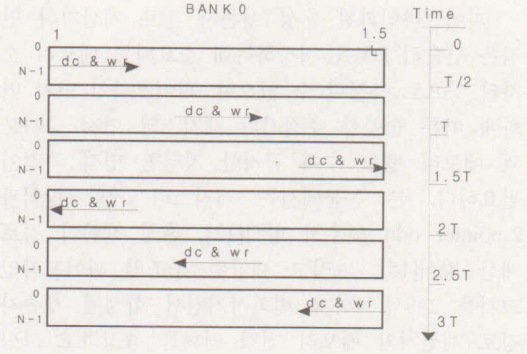
그림 8. 2단 파이프라인화된 생존자 경로 계산기. (a) 파이프라인화된 순방향 추적기, (b) 생존자 경로 계산기 구조.

그림 8은 제안된 생존자 경로 계산기 구조를 보인다. 그림 8 (a)는 2단 파이프라인을 적용한 순방향 추적기를 나타내며 각 상태 레지스터는 L 시간마다 한 번씩 초기화와 결과 출력을 담당하며 전체적인 생존자 상태 출력율은 L/2로 변화한다. 그림 8 (b)는 파이프라인된 순방향 추적기를 사용한 전체 생존자 경로 계산기 구조이다. 혼성 생존자 경로 계산기와 메모리의 크기 및 순방향 추적기 동작 속도에서 차이가 있다.

그림 9는 2단 파이프라인을 적용한 혼성 생존자 경로 계산기에서 메모리 구성과 동작을 나타내었다. 그림 9 (a)는k-pointer even 방식과 유사하게 읽기와 쓰기를 구분한 방법으로 전체 메모리 소요량은 2L 이며, 읽기와 쓰기에 필요한 주소 제어기



(a)



(b)

그림 9. 2단 파이프라인 생존자 경로 계산기의 메모리 구성 및 동작. (a) 읽기와 쓰기 분리 (b) 읽기 쓰기 동시 진행.

설계가 간편하다. 그림 9 (b)는 읽기와 쓰기가 하나의 뱅크에서 일어나는 구조를 갖고 있으며 전체 메모리 소요량이 1.5L 로 기존의 역추적 방식 메모리 소요량의 한계를 극복하였다. 다만 이 경우는 k-pointer odd 방식이 갖는 공통적인 문제점, 즉 읽기와 쓰기 방향이 주기적으로 변화하므로 주소 제어기 설계가 even 방식에 비해 복잡하다는 문제점은 그대로 가지고 있다.

표1 에 파이프라인 크기에 따른 메모리 소요량을 일반적인 경우로 확대하고 읽기와 쓰기에 할당되는 메모리 뱅크를 구분하는 경우와 읽기와 쓰기를 하나의 메모리에서 동시에 진행하는 방법으로 구분하여 제시하였다. 표 1의 2K-1 은 ACS 계산기에서 발생하는 DV의 개수를 나타내며 L 는 복호 길이, n 은 파이프라인 스테이지 수를 각각 나타낸다. 전술한 바와 같이 면적은 메모리 사용량과 메모리 뱅크 수를 동시에 고려하여야 하며 순방향 추적기에 필요한 기억소자의 크기를 같이 고려하여야 한다.

표 1. 파이프라인 크기에 따른 메모리 소요량.

사용방식	메모리 사용량	메모리 뱅크 수	단위메모리 크기
읽기/쓰기 분리 진행	$2^{K-1}L(1+2/n)$	n+2	L/n
읽기/쓰기 동시 진행	$2^{K-1}L(1+1/n)$	n+1	L/n

파이프라인화된 혼성 생존자 경로 계산기의 면적은 단순히 DV를 저장하는데 소요되는 메모리 소자의 양으로 측정할 수 없으며, 파이프라인 크기 변화에 따른 순방향 추적기의 레지스터 면적 증가분과 메모리 뱅크 수에 근거한 정밀한 면적 측정이 필요하다. 본 논문에서는 레지스터 교환 방식과 2-pointer odd 역추적 방식[12], 혼성 생존자 경로 계산 방식[15], 그리고 제안된 구조의 파이프라인 크기를 2에서 4까지 변화시키면서 합성된 생존자 경로 계산기의 메모리 면적 변화를 측정하고 그림 10에 결과를 제시하였다. 복호 메모리 소요량의 감소와 순방향 추적기 면적의 증가, 메모리 뱅크 수의 변화에 따라 2 단 파이프라인이 효과적임을 알 수 있으며, 이 때 생존자 경로 계산을 위한 기억소자의 면적은 일반적인 역추적 방식에 비해 37.0%, 파이프라인을 채택하지 않은 혼성 생존자 경로 계산기에 비해 15.9% 감소한다. 따라서 제안된 시스템은 2단 파이프라인을 사용한 혼성 생존자 경로 계산기 구조를 채택하였다.

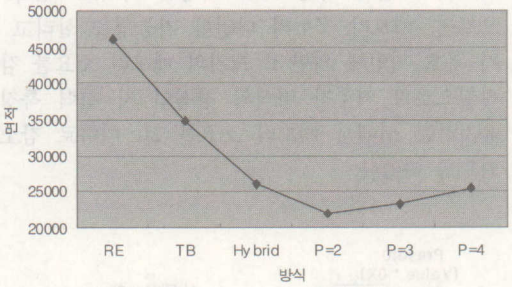


그림 10. 생존자 경로 계산기의 구조별 면적 변화.

Ⅳ. 실험 결과

제한길이 K = 7인 Viterbi 복호기를 대상으로 제안된 구조를 적용하여 면적 비교를 실시하였다. 합성에 사용된 Viterbi 복호기의 파라미터는 표 2에 제시하였다. RT 수준의 VHDL 언어로 기술된 복호기 모델을 Synopsys 사의 Design Analyzer와 Hynix 0.35m 공정라이브러리를 이용해 합성하고 동작을 검증하였다. 메모리의 경우는 UMC에서 제공하는 메모리 컴파일러 MemMaker 와 0.35m 스탠다드 셀 공정 라이브러리를 이용하여 결과를 측정하였다.

표 2. Viterbi 복호기의 설계 파라미터.

파라미터	값
제한 길이(K)	7
생성다항식(g1/g2)	$171_8/133_8$
부호율(R)	1/2, 3/4, 7/8
채널 모델링/변조방식	AWGN/QPSK

표 3에 앞 장에서 기술한 설계 공간 탐색을 바탕으로 복호기의 구조를 2단 파이프라인으로 결정하고 합성을 수행한 경우 기존 구조[15]와 면적 비

교를 제시하였다. 제안된 생존자 경로 계산기는 기존 방식과 비교했을 때 16% 정도 면적 감소가 가능하였으며, Viterbi 복호기에서 생존자 경로 계산기의 면적 비중이 50%를 넘기 때문에 전체적으로도 9%의 면적 감소 효과를 얻을 수 있었다.

표 3. 제안된 Viterbi 복호기의 면적 비교

		Hybrid[15]	제안 구조	감소율
면적	생존자경로	26,102	21,929	15.9%
	기타	20,351	20,351	-
	전체	46453	42,280	9.0%

* 단위 면적: 2 input NAND gate

V. 결론 및 추후 과제

본 논문에서는 Viterbi 복호기의 알고리듬 및 구조 수준의 분석을 통해 면적을 최적화하는 생존자 경로 계산기 구조를 제안하였다. 기존의 Viterbi 복호기는 신호 천이 빈도가 낮고 메모리 사용이 용이한 역추적 구조에 기반한 생존자 경로 계산기를 사용함으로써 메모리 사용량의 최저 한계가 복호 길이의 두 배에 이르고 있으며, 제한 길이 K가 크고 천공 부호를 지원하여 복호 길이가 길어지는 경우 생존자 경로 계산기 면적은 전체 복호기 면적의 50%를 넘게 된다. 제안된 Viterbi 복호기는 역추적 방식의 메모리 사용량 최저 한계를 구현하는 혼성 생존자 경로 계산기를 기반으로 순방향 추적기를 파이프라인화 함으로써 소요되는 메모리의 양을 감소시키고, 다양한 메모리 구성 방식을 기반으로 면적을 측정하여 최적의 파이프라인 크기를 설정하였다. 실험 결과 기존의 역추적 방식에 비해 37%, 혼성 생존자 경로 계산기에 비해 16% 정도 생존자 경로 계산기의 면적을 감소시킬 수 있었다.

추후 과제로는 논리 수준 합성 결과를 실제 layout 수준의 회로로 전환하여 알고리듬 수준부터 회로 수준에 이르는 전체적인 단계에서 최적화를 수행할 필요가 있으며, 제한 길이 증가시 지수적으로 복잡도가 증가하는 ACS 계산기의 효율적 설계를 통한 면적 및 전력 감소를 통한 복호기 성능 향상이 필요할 것으로 예상된다.

참 고 문 헌

[1] A. Viterbi, "Convolutional Codes and their Performance in Communication Systems," IEEE Trans. on Commun., Vol. 19, No. 5, pp. 751-772, Oct. 1971.

[2] 이문호, "비동기식 IMT-2000의 채널 부호화," 한국통신학회 학회지, 제 14권, 9호, pp. 170-187, 1996년 9월.

[3] G. Forney, "Convolution Codes II: Maximum Likelihood Decoding," IEEE Trans. on Info. Theory,Vol. IT-25, No. 7, pp. 222-266, July 1974.

[4] S. Kim and S-Y. Hwang, "Area-Efficient VLSI Architecture for the Traceback Viterbi Decoder Supporting Punctured Codes," Electronics Letters, Vol. 32, No. 8, pp. 733-735, April 1996.

[5] A. Netravali and B. Haskell, Digital Picture Representation and Compression, Plenum Press, New York, 1988.

[6] Y. Yasuda, K. Kashiki and Y. Hirata, "High-rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding," IEEE Trans. on Commun., Vol. 32, No. 3, pp. 315-319, March 1984.

[7] T. Ishitaniet al., "A Scarce-State-Transition Viterbi Decoder VLSI for Bit Error Correction," IEEE Journal of Solid State Circuits, Vol. SC-22. No. 4, pp. 575-582, Aug. 1987.

[8] M. Alston and P. Chau, "On the Maximum Path Metric Difference in Viterbi Decoders of Punctured Rate (n-1)/n Codes," in Proc. ISCAS, Vol. 4, pp. 2061-2064, May 1992.

[9] U.S. Patent, Patent Number 5050191, Sept. 17 1991.

[10] U.S. Patent, Patent Number 4802174, June 31 1989.

[11] C. Rader, "Memory Management in a Viterbi Decoder," IEEE Trans. on Commun., Vol. 29, No. 9, pp. 1399-1401, Sept. 1981.

[12] G. Feygin and P. Gulak, "Architectural Tradeoffs for Survivor Sequence Memory

Management in Viterbi Decoders," IEEE Trans. on Commun., Vol. 41, No. 3, pp. 425-429, March 1993.

[13] O. Collins and F. Pollara, "Memory Management in Traceback Viterbi Decoders," The JPL Progress Report, 42-99, Nov. 1989.

[14] H. Bustamante, et al., "Stanford Telecom VLSI Design of a Convolutional Decoder," in Proc. IEEE Conf. Military Commun., Boston, MA, vol. 1, pp. 171-178, Oct. 1989.

[15] P. Black and T. Meng, "Hybrid Survivor Path Architectures for Viterbi Decoders," in Proc. ICASSP-93, Vol. 1, pp. 433-436, April 1993.

[16] D. Dabiri and I. Blake, "An Area Efficient Surviving Path Unit for the Viterbi Algorithm," in Proc. ICC'99, Vol. 1, pp. 300-304, 1999.

황 선 영 (Sun-Young Hwang) 정회원

1976년 2월 : 서울 대학교 전자공학과 졸업
1978년 2월 : 한국 과학원 전기 및 전자공학과 공학석사 취득
1986년 10월 : 미국 Stanford 대학 전자공학 박사학위 취득
1976~1981 : 삼성 반도체 주식회사 연구원, 팀장
1986~1989 : Stanford 대학 Center for Integrated System 연구소 책임 연구원
Fairchild Semiconductor Palo Alto Reaserch Center 기술 자문
1989~1992 : 삼성전자(주) 반도체 기술 자문
2002년 4월~2004년 2월 : 서강대학교 정보통신대학원장
1989년 3월~현재 : 서강대학교 전자공학과 교수
<관심분야> 컴퓨터 시스템, SoC 설계 및 framework 구성, Embedded System 설계 등

김 식 (Sik Kim) 정회원

1994년 2월 : 서강대학교 전자공학과 졸업
1996년 2월 : 서강대학교 전자공학과 석사학위 취득
1996년 : 현대전자 멀티미디어 연구소 연구원
2003년 8월 : 서강대학교 전자공학과 대학원 공학박사 취득
현재 : 삼성전자 반도체 총괄 CAE센터 책임연구원
<관심분야> System level design, Computer Architecture 및 디지털 통신/영상 압축용 VLSI 설계 등