

사양변수를 이용한 비터비 복호기의 자동설계

(Automated Design of Viterbi Decoder using Specification Parameters)

孔明皙*, 裴晟日*, 金在錫**

(Myoungseok Kong, Sungil Bae, and Jaeseok Kim)

요 약

본 논문에서는 이동 통신 시스템에서 많이 사용되는 다양한 사양의 비터비 복호기를 자동으로 생성할 수 있는 가변적 비터비 복호기의 설계방법을 제안한다. 여기서 제안하는 가변적 비터비 복호기는 구속장, 부호율, 생성다항식 등의 길쌈부호기 사양, 프레임당의 비트 수, 전송 속도 등의 데이터 전송 사양, 그리고 복호기의 성능을 위한 연판적 비트수 등을 매개 변수화(parameterization)하여, 사용자가 제공하는 사양변수에 맞는 비터비 복호기를 최적으로 자동 생성하도록 설계되었다. 이를 위해 C 언어로 설계된 사용자 인터페이스 환경모듈을 구현하였고, 또한 VHDL 언어와 generic 변수를 활용한 비터비 복호기의 기능 블록 모듈이 계층 구조적으로 설계되었다. 설계된 가변적 비터비 복호기의 검증을 위해, IS-95 CDMA 시스템의 규격에 맞는 비터비 복호기를 자동 생성하여 기존의 설계된 내용과 비교 검증하였다. 제안된 방식은 앞으로 사양이 조금씩 바뀔 때마다 비터비 복호기를 새로이 설계할 필요없이, 변경된 사양만 제공함으로써 매우 빠른 시간 내에 변경된 하드웨어 설계를 얻을 수 있는 새로운 설계방식이라 할 수 있다.

Abstract

In this paper, we proposed a design method of parameterized viterbi decoder, which automatically synthesizes the diverse viterbi decoders used in the digital mobile communication systems. It is designed to synthesize a viterbi decoder specified by user-provided parameters. Those parameters are constraint length, code rate and generator polynomials of the convolutional encoder, data rate and bits/frame of the data transmission, and soft decision bits of viterbi decoder. For the design of the parameterized viterbi decoder, we designed a user interface module using C-language, and a viterbi decoder module in a hierarchical structure using VHDL language and its generic statement. For the verification of the parameterized viterbi decoder, we compared our synthesized viterbi decoder with the conventional viterbi decoder which is designed for the IS-95 CDMA system. The proposed design method of the viterbi decoder will be a new method to obtain a required viterbi decoder in a very short time only by supplying the design parameters.

I. 서 론

디지털 이동 통신 시스템에서는 데이터 전송시 비선

형 감쇄, 잠음간섭 현상, 페이딩(fading)등의 채널장애로 인해 발생하는 전송 데이터 오류를 감쇄시킬 필요가 있다. 따라서, 대부분의 디지털 이동 통신 시스템은 채널코딩 방식으로, 메모리 소자를 이용하여 이전의 저장된 정보가 현재의 데이터에 일정한 규칙성으로 영향을 미치는 구조로 되어 있어 블록코딩보다 에러정정 효율이 우수한 길쌈부호방식을 사용하고 있다^{[1][2]}.

길쌈부호기의 사양은 생성다항식(Generator poly-

* 學生會員, ** 正會員, 延世大學校 電子工學科

(VLSI&CAD Lab., Dept. of Electronic Eng., Yonsei University)

接受日字:1998年9月24日, 수정완료일:1998年12月8日

nomial), 구속장(Constraint length), 부호율(Code rate)로 결정된다. 생성다항식은 구속장의 길이에 따라 채널 에러 정정능력을 나타내는 최대 자유 거리(Maximum free distance)를 나타낼 수 있도록 결정되어 있다. 구속장은 구속장의 길이가 길수록 채널 에러 정정 능력이 우수하지만 복호기의 하드웨어 부담이 커지기 때문에 이동 통신 시스템에서는 보통 7, 9 정도의 구속장을 사용하고 있다. 그리고 부호율은 입력된 정보비트 k 에 대해 부호화된 출력 비트 n 의 비율을 나타내며 $R=k/n$ 으로 표현한다. 부호율이 $1/2$ 이상의 사용은 복호기의 하드웨어 구조가 복잡해지기 때문에 펄쳐드 길쌈부호기(Punctured convolutional encoder)를 사용하여 $R=1/2$ 의 부호율로 구현하고 있다^[3].

위와 같은 길쌈부호에 대한 최적 복호 알고리즘으로, 수신된 부호열들의 여러 경로를 탐색한 후 유사성(Likelihood)이 가장 높은 경로의 부호열을 선택함으로써 원하는 데이터를 복호하는 비터비 알고리즘이 일반적으로 널리 사용된다^{[4] [5]}. 이 알고리즘을 적용한 비터비 복호기는 복호기의 성능향상을 위하여 연판정(Soft decision)을 사용하는 경우에 수신된 심볼과의 유클리디언 거리(Euclidean distance)를 계산하는 가지값계산부, 최소 상태를 구하는 ACS(Add-Compare-Select)부, 갱신된 상태값을 저장하는 상태값저장부, 그리고 ACS부에서 전달되는 경로 선택 정보를 이용하여 역추적방법으로 데이터를 복호하는 역추적부의 기능 블록들로 구성되어 있다. 일반적인 비터비 복호기의 분류는 ACS의 수에 따라 완전 병렬 처리 복호기^{[6] [7]}와 부분 병렬처리 복호기^{[8] [9]}로 나눌 수 있다. 전자인 경우에는 각 상태마다 ACS를 할당하여 빠른 데이터 복호 처리를 수행할 수 있으나 하드웨어 비용이 커진다는 부담이 있고, 후자인 경우는 상태수보다 적은 제한된 ACS를 공유하여 복호를 수행하기 때문에 하드웨어를 효율적으로 사용할 수 있다.

현재 길쌈부호기를 사용하는 대부분의 디지털 통신 시스템은 비터비 복호기를 사용하고 있기 때문에 여러 통신 시스템들에 대응되는 다양한 사양의 복호기를 필요로 한다. 기존의 설계 방식에서는 비터비 복호기를 특정 시스템 사양에 맞게 설계하기 때문에 시스템의 사양이 변화할 때 이에 맞는 비터비 복호기의 설계 변경이 불가피하게 된다. 따라서 이러한 문제를 해결하기 위해 비터비 복호기의 구조에 영향을 미치는 사양

변수를 추출한 후, 이러한 변수의 입력에 따라 사용자가 원하는 복호기를 자동 생성할 수 있는 설계방법과 비터비 구조에 대한 연구가 필요하다.

본 논문은 위에서 제시된 문제점을 해결하기 위해 두가지 방향으로 연구를 수행하였다. 우선 비터비 복호기의 설계에 영향을 주는 사양변수들을 정의한 후, 이 변수가 VHDL 언어로 설계된 비터비 복호기의 설계변수 입력으로 주어지도록 하는 C-언어를 사용한 사용자 인터페이스부를 구현하였다. 그리고 입력된 설계변수에 따라 복호기가 자동 생성될 수 있도록, 설계변수에 따른 비터비 복호기 구조의 규칙적이고 반복적인 특성을 이용하여 VHDL 언어의 설계기법으로 구현하였다. 그림 1은 본 논문에서 제안하는 사용자 제공 사양변수를 입력받아 원하는 비터비 복호기가 자동적으로 생성되는 개념도를 보여준다.

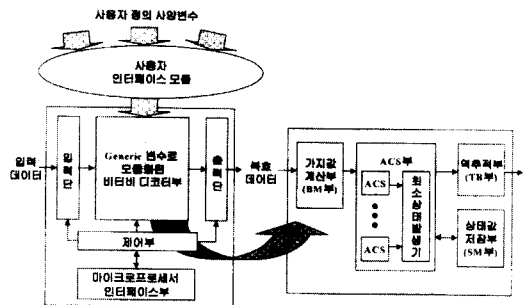


그림 1. 비터비 복호기 자동 생성기의 전체 개념도
Fig. 1. Conceptual diagram for automatic generation of viterbi decoder.

본 논문의 구성에서 II장에서는 사용자가 요구하는 사양변수를 입력으로 받는 C-언어로 구현된 사용자 인터페이스 모듈을 설명한다. III장에서는 사양변수에 따른 비터비 복호기의 계층구조 영향에 대한 분석과 이를 통해 VHDL 설계방법을 사용한 가변적 비터비 복호기의 설계 구조를 보여주고, IV장에서는 IS-95 CDMA 시스템의 특정 사양변수에 따라 자동 합성된 결과와 이의 시뮬레이션 검증을 살펴본다. 그리고 마지막으로 V장에서 결론을 제시한다.

II. 사용자 인터페이스 모듈의 설계

가변적 비터비 복호기의 구현을 위해, 우선 설계에 영향을 주는 사양변수를 정의하는 작업이 필요하다. 이를 위해 현재 사용중인 디지털 통신 시스템의 길쌈

부호기 사양과 데이터 전송속도를 살펴보면 표 1과 같다.

표 1. 디지털 통신 시스템별 길쌈부호기와 데이터 전송 속도

Table 1. Convolutional encoder and data rate of diverse digital communication system.

시스템 \ 사양	Constraint Length	Code Rate(R)	Information Data Rate(bps)
GSM (TDMA)	5	1/2	2.4K, 9.6K
IS-54 (TDMA)	6	1/6	8K
IS-95 (CDMA)	9	1/2(F) 1/3(R)	1.2K, 2.4K, 4.8K, 9.6K
PCS (CDMA)	9	1/2(F) 1/3(R)	1.2K, 2.4K, 4.8K, 9.6K
WLL (CDMA)	7, 9	1/2	8K, 16K, 32K, 64K, 80K, 144K
IMT-2000 (CDG 사양)	9	1/2, 1/3(F) 1/2, 1/4(R)	· 1.228M: 9.6(0.8K)~153.6K · 3.686M: 9.6K(0.8K)~460.8K · 7.372M: 9.6K~614.4K · 14.745M: 9.6K~1.228M

표 1을 보면 각 통신 시스템마다 길쌈부호기와 시스템의 데이터 전송속도의 사양이 다양함을 확인할 수 있다. 길쌈부호기는 구속장, 부호율, 생성다항식의 3가지 변수들에 의해 사양이 결정되기 때문에 이 변수값들에 따라 여러 사양의 길쌈부호기가 존재하게 된다. 이 3가지 변수들은 각각 비터비 복호기의 설계에 영향을 주게 된다. 구속장은 상태의 비트수나 역추적 깊이(Traceback depth)등에 영향을 주게 된다. 부호율은 연판정인 경우 유클리디언 거리를 계산하거나 상태 천이등의 결정에 영향을 주게 된다. 그리고 생성다항식은 송신 측의 길쌈부호기에 사용되는 생성다항식과 동일하게 사용하며, 이 값은 유클리디언 거리의 계산시에 필요하게 된다. 이와 같이 길쌈부호기를 결정하는 3가지 변수들은 각각 비터비 복호기의 사양을 결정하는 중요한 요소가 된다.

비터비 복호기는 ACS의 수에 따라 복호 처리 속도가 결정된다. 복호 처리 속도는 일반적으로 ACS의 수에 따라 비례하게 되며, 빠른 처리 속도를 원하는 경우 각 상태마다 ACS를 할당하여 완전 병렬 처리 방식으로 복호를 수행할 수 있고, 그렇지 않은 경우에

는 원하는 처리 속도에 맞게 ACS를 두는 부분 병렬 처리 방식을 사용하게 된다. 복호 처리 속도는 사용중인 시스템의 데이터 전송속도에 따라 결정되므로 이 값은 비터비 복호기의 ACS 수를 결정하는 사양변수가 된다. 또한, 시스템의 데이터 전송사양인 전송속도 이외에도 복호를 수행할 데이터의 개수를 나타내는, 부호화 이전의 데이터 비트수는 복호기의 제어부에 영향을 미치는 사양변수가 된다.

마지막으로 비터비 복호기의 설계에 영향을 미치는 사양변수로, 복호기의 성능을 결정 하는 연판정 비트 수가 있다. 보통 8레벨(3비트)의 연판정인 경우 경판정(Hard decision)에 의한 방법보다 약 2dB의 성능 향상이 있어 대부분의 디지털 통신 시스템에서 3, 4비트의 연판정을 주로 사용하고 있다^[10]. 연판정을 사용하는 경우에 비트수에 따라 유클리디언 거리 계산이나 상태값(State metrics)의 비트수를 결정하게 된다.

이와 같이 여러 시스템 사양에 대응되는 가변적 비터비 복호기의 설계를 위해 비터비 복호기의 설계에 영향을 미치는 사양변수를 정의하면 다음과 같이 6개로 정리할 수 있다.

- CL = 구속장(constraint length)
- CR = 부호율(code rate)
- GP = 생성다항식(generator polynomial)
- AP = ACS 쌍의 수(ACS pair)
- SN = 프레임당 비트 수(bits/frame)
- SD = 연판정 비트 수(soft decision bits)

여기서, CL, CR, GP는 길쌈부호기의 사양변수이며 AP, SN은 데이터 전송 사양변수를 나타낸다. 그리고 SD는 비터비 복호기의 성능 향상을 위한 사양변수를 나타낸다. 여기서 데이터 전송속도에 따라 결정되는 AP는 한 쌍의 ACS를 기본 단위로 하여 2의 자승으로 개수를 정하였다.

C-프로그램을 이용한 사용자 인터페이스 모듈은 사용자의 편리성을 제공하기 위해 이와 같은 6개의 복호기 사양변수를 입력으로 받아 비터비 복호기 설계변수의 입력값으로 주어지게 된다. 사용자 인터페이스 모듈의 구성은 사용자가 요구하는 사양변수가 가변적 비터비 복호기의 설계변수 입력값으로 대치되도록, 설계 변수 앞에다 특수문자를 두어 프로그램이 특수문자를 인식하여 사양 변수값으로 대치하는 기능을 수행하도록 되어있다. C-프로그램의 구성 흐름도는 그림 2와 같다.

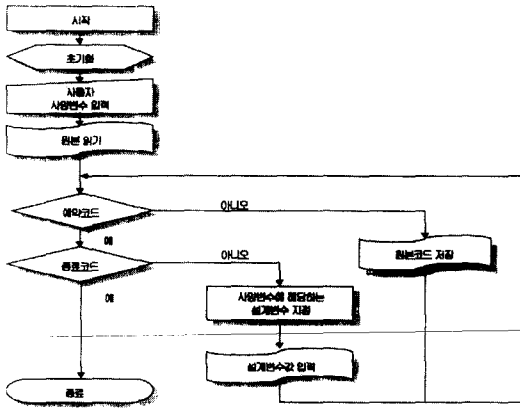


그림 2. C-프로그램의 구성 흐름도
Fig. 2. Flow chart of C-program.

가변적 비터비 복호기는 VHDL의 설계기법 중 generic 문을 이용하여 구현하였기 때문에 VHDL 코드의 generic 문 입력변수에 설계변수 값을 대치함으로써 가능하게 된다. 따라서, 이와 같은 C-프로그램을 이용한 사용자 인터페이스부를 통해 사용자는 사양변수의 입력만으로 원하는 비터비 복호기의 VHDL의 코드를 얻을 수 있다.

III. 가변적 비터비 복호기 모듈의 설계

가변적 비터비 복호기가 입력된 설계변수에 따라 자동적으로 생성되기 위해서는 VHDL 설계기법에 대한 연구가 필요하다. 가변적 VHDL 모델링 기법은 VHDL 설계기법 중 generic 문과 generate 문을 사용하여 구현될 수 있다. VHDL 언어에서 generic 문은 계층적 구조에서 상위 모듈의 설계변수가 하위 모듈로 전달하는 기능을 수행할 수 있다. 그리고 설계변수 전달시에 설계변수의 조작이 필요한 곳에는 function을 사용하여 변수의 조작이 가능하게 된다. 또한 VHDL 언어의 generate 문은 전달받은 설계변수에 따라 규칙적이고 반복적인 모듈의 생성에 사용된다.

이와 같은 VHDL 설계기법을 이용하여 가변적 비터비 복호기를 설계하기 위해서는 우선 비터비 복호기의 내부 구성 블록들을 계층 구조화하는 작업이 필요하게 된다. 이 작업을 통해 II장에서 정의된 6개의 복호기 사양변수에 대한 복호기 내부의 영향도를 살펴볼 수 있다. 그림 3은 정의된 6개의 사양변수에 따른 계

층 구조화된 비터비 복호기의 내부 블록 영향도를 보여준다.

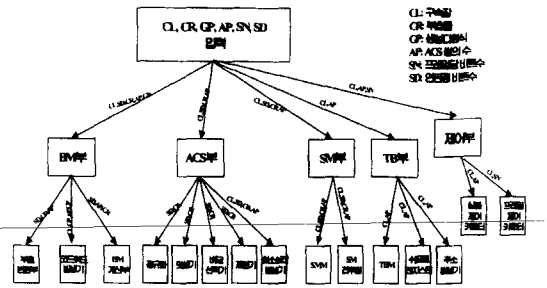


그림 3. 정의된 사양 변수에 따른 비터비 복호기 계층 구조의 영향도
Fig. 3. Effect diagram of specification parameters for hierarchical structure of viterbi decoder.

그림 3을 보면 정의된 6개의 사양변수에 따른 비터비 복호기의 내부 구성 블록인 가지값계산부(BM부), ACS부, 상태저장부(SM부), 역추적부(TB부) 및 제어부의 영향도와 각 내부 구성 블록의 하위블록에 대한 영향도를 살펴볼 수 있다. 이처럼 상위 블록에 영향을 주는 설계 변수는 상위 블록의 내부 구성 블록에 전달되어 규칙적이고 반복적인 형태로 영향을 미치게 된다. 이와 같은 설계변수의 전달은 VHDL 언어의 generic 기법을 이용하여 구현되고, 전달받은 변수들을 이용하여 내부 블록들은 규칙적이고 반복적인 구조로 생성되게 된다.

1. 가변적 BM부의 설계

BM(Branch Metric)부는 연판정인 경우 수신된 부호어와 길쌈부호기에서 발생하는 코드워드간의 유클리디언 거리인 가지값을 계산하는 기능을 수행한다. 가지값의 계산은 하드웨어 부담을 줄이기 위해 절대값 계산방법을 사용하여 구현하였다. BM부는 2의 보수로 수신되는 연판정 부호의 부호 비트를 반전시키는 부호 변환부, 길쌈 부호기의 부호화 방식과 동일한 코드워드 발생기, 그리고 부호변환된 값과 코드워드 발생기의 출력값인 코드워드를 이용하여 가지값을 계산하는 BM계산부로 구성되어 있다. 그림 4는 이와 같은 기능을 수행하는 BM_SUB 모듈들로 구성된 가변적 BM부의 구조를 보여준다.

BM부는 CL, SD, CR, AP, GP의 사양 변수에 따라 내부 블록들이 규칙적, 반복적으로 생성된다. 부호 변환부에서 SD는 부호변환할 한 심플당 비트 수, CR

은 한번에 부호변환되는 심볼 개수를 결정한다. 코드 워드발생기에서는 CL, GP, CR이 설계에 영향을 미친다. 여기서 CL은 구속장의 길이로 코드워드 생성을 위한 카운터의 크기를 결정하고 GP는 송신단의 길쌈 부호기와 동일한 값으로 주어지게 된다. CR은 생성다항식의 생성계열로 연결된 이진 가산기의 개수를 결정한다. BM계산부에서는 SD, CR가 영향을 미치며 SD는 비트 수, CR은 각 심볼마다 계산된 유클리디언 거리의 더해질 개수를 결정한다. 그리고 AP는 BM부에서 출력되는 가지값의 개수를 결정하게 된다. 전반적으로 AP의 사양변수는 병렬 처리할 가지값 개수를 나타내므로 BM_SUB 블록들의 개수를 결정하게 된다.

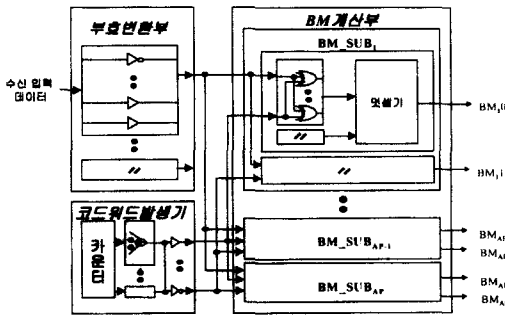


그림 4. 가변적 BM부의 기본구조
Fig. 4. Architecture of the parameterized BM module.

2. 가변적 ACS부의 설계

ACS부는 격자도상에서 각 상태로 상태전이하는 이전 상태의 상태값과 가지값을 이용 하여 상태값이 적은 경로를 선택하는 기능을 수행한다. 또한 선택된 경로의 갱신된 상태값을 SM부에 전달하고 경로 선택 정보와 역추적 초기에 필요한 최소 상태값을 갖는 상태를 TB부에 전달하는 기능을 수행한다. ACS 부에서 ACS 쌍의 수는 송신단의 데이터 전송 속도에 따라 그 수가 결정되며 AP, CL, SD, CR의 사양 변수에 의해 영향을 받는다. 그림 5는 가변적 ACS부의 구조를 보여준다. 여기서 SM_k 는 ACS를 수행할 상태값 k 를 나타내고, S_k 는 상태값 SM_k 를 갖는 상태 k 를 나타낸다.

그림 5에서 1쌍의 ACS는 3단의 파이프라인(Pipeline)구조를 사용하여 클럭 속도를 높일 수 있도록 설계하였다^[11]. 정규화블록은 뱀셈기로 구성되어 있는데, 이 블록은 ACS 처리 후 전 상태값(all state matrices)중 최소상태값으로 다음 심볼 처리시에 입력

되는 상태값을 정규화시켜줌으로써 상태값의 기하급수적인 증가를 막아주는 기능을 수행한다. 덧셈기는 상태값의 품질비트(Quality bit)^[12]인 최상위비트를 제외한 나머지 상태값과 BM부에서 계산된 가지값인 BM0과 BM1을 더하는 기능을 수행한다. 품질비트는 덧셈에 이용되지 않기 때문에 상태값 N-1 비트와 가지값 N-1비트를 더한 N 비트의 덧셈 출력값과 품질비트를 출력 하여 N+1 비트를 출력한다. 제한기는 N 번째 비트 값이 1이면 나머지 N-1 비트의 값을 최고값으로 설정하여, 이 값과 N+1번째 비트인 품질비트를 합하여 N 비트가 출력 되도록 한다. 따라서, 비교 선택기에서의 상태값 비교시에 정확도를 높일 수가 있다. S/P 블록은 경로 선택정보인 역추적 정보를 TB부에 전달할 때 역추적 메모리의 분리 개수만큼 저장 후, 한꺼번에 역추적 메모리에 전달하여 메모리의 접근 횟수를 줄일 수 있도록 하였다. 최소상태발생기는 최소상태값선택기와 최소상태선택기로 구성되어 있고 각각은 정규화를 위한 최소상태값(min_state)과 역추적을 위한 초기 상태 S_{min} 을 결정한다.

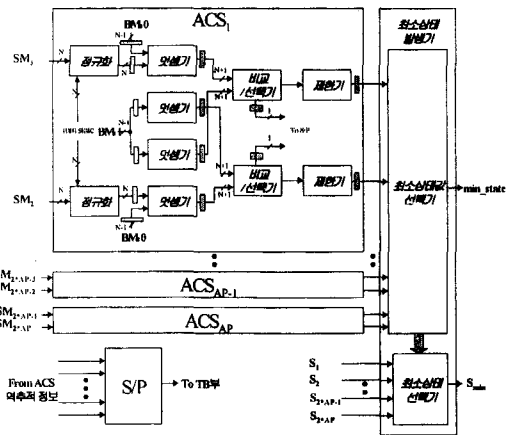


그림 5. 가변적 ACS 부의 구조
Fig. 5. Architecture of the parameterized ACS module.

ACS부는 사양 변수인 AP의 개수에 따라 그림 5의 구조를 갖는 ACS쌍의 수가 생성 된다. SD와 CR은 그림 5의 비트 수 N을 결정하게 되며, CL은 최소 상태발생기의 최소 상태값을 갖는 최소상태를 선택하기 위한 상태들의 비트 수를 결정하게 된다. ACS쌍의 수를 나타내는 AP는 비터비 복호기 전체의 구조에 영향을 미치는 가장 중요한 사양 변수로 이 값에 따라 상태값을 저장하는 SM부와 역추적을 수행하는 TB부

의 메모리 관리에 영향을 미치게 된다.

3. 가변적 SM부의 설계

SM(State Metric)부는 ACS 수행을 위한 상태값의 읽기와 ACS 수행 후 선택된 경로의 갱신된 상태값을 저장하는 기능을 수행한다. SM부에서는 상태수보다 적은 ACS로 구성된 부분 병렬 처리 복호기의 경우 효율적인 메모리 관리가 요구된다. SM부는 ACS 수행시에 상태값의 읽기와 저장을 번갈아 수행하기 위해 동일 구조를 갖는 2개의 SM_SUB 블록들로 구성되어 있다. 그리고 각각의 SM_SUB 블록은 ACS 수행에 필요한 상태값을 선택하기 위한 SM 라우팅 블록과 2^{CL-1} 개 상태값을 저장하는 SM 메모리로 이루어져 있다. 한개의 SM_SUB 블록은 ACS 수행을 위한 상태값의 읽기를, 다른 한 개는 ACS 수행 후에 상태값의 저장을 매 입력 데이터마다 switch 신호에 따라 번갈아 수행한다. 그림 6은 가변적 SM부의 구조가 보여 준다.

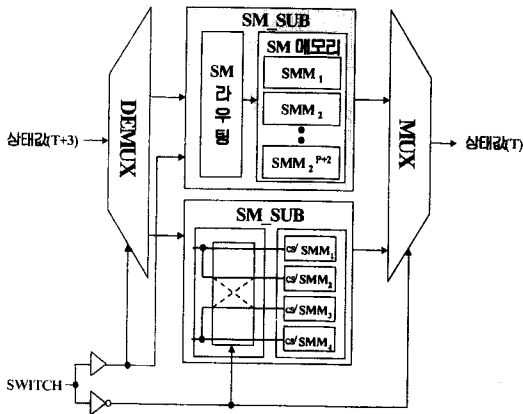


그림 6. 가변적 SM부의 구조
Fig. 6. Architecture of the parameterized SM module.

SM 메모리는 일반적으로 상태값의 위치를 메모리에 고정하여 사용하는 고정 상태배열과 일정한 규칙성으로 메모리 안에서 상태값의 위치를 변화시키는 유동 상태배열로 관리된다. 유동 상태배열을 사용하는 경우 중에서 ACS 수행 후에 메모리 안에서 상태값의 쓰기 위치를 읽기 위치에다 저장하는 방법으로 상태값을 관리할 수 있다. 이 방법은 1개의 SM 메모리만으로 상태값을 효율적으로 관리할 수 있지만, 여러 쌍의 ACS를 사용하는 경우에 메모리 관리가 매우 복잡해지는 문제가 발생한다. 따라서 ACS 쌍의 수에 따라 규칙

적으로 메모리 관리를 할 수 있도록, 고정 상태배열로 2개의 SM 메모리를 사용하여 상태값을 관리 하였다. SM부에 영향을 주는 사양 변수는 CL, SD, CR, AP로 CR과 SD는 상태 값의 비트수, CL은 상태의 비트수를 결정하게 된다. AP는 SM 메모리의 구조와 이에 따르는 라우팅구조에 영향을 미치게 된다. SM 메모리는 AP의 수에 따라 ACS를 수행하기 위해 여러 개의 상태값들을 동시에 메모리에서 읽고 저장해야 한다. 따라서 2-포트(2-port) 메모리를 사용하기 때문에 한 개의 SM 메모리에서 여러 상태값들을 동시에 읽고 저장하기 위해서는 SM 메모리를 여러 개의 하위 메모리로 분리해야 한다. 상태값들의 읽기와 저장을 수행할 때 하나의 하위 메모리에서 두 개의 상태값을 읽거나 저장할 수 없기 때문에 분리된 각각의 하위 메모리에서 하나의 상태값 읽기와 저장을 할 수 있는 상태배열이 필요하다. 구속장이 5, 부호율이 1/2 그리고 1쌍의 ACS를 사용한 경우를 살펴보면, 우선 그림 7(a)와 같이 SM 메모리를 홀수와 짝수 상태배열로 이루어진 2개의 하위 메모리로 나누어 상태값을 관리할 수 있다. 이와 같은 경우 상태값의 읽기는 동시에 수행할 수 있지만 상태값의 동시 저장을 할 수 없음을 알 수 있다. 즉, (0000, 0001)의 두 개의 상태들이 ACS 수행 후 (0000, 1000)의 상태 위치에 저장할 때 상태값의 순차 저장만이 가능하다. 따라서 이와 같은 구조에서는 상태값의 동시 읽기와 동시 저장이 가능하지 않게 된다. 따라서 그림 7(b)에서 볼 수 있는 것처럼 짝수 상태의 메모리와 홀수 상태의 메모리 각각을 두 개로 분리함으로써 상태값의 동시 읽기와 저장이 가능하게 된다. 즉, ACS 수행 후 (0000, 1000) 상태들의 상태값을 동시 저장할 수 있음을 알 수 있다. 그림 7(b)에서 볼 수 있는 방법으로 ACS 쌍의 수에 따라 SM 메모리는 규칙적으로 여러 개의 하위메모리로 분리되고, 각각 분리된 하위 메모리의 상태배열들이 결정된다. 즉, 상태들을 2^{P+1} ($2^P = \text{ACS 쌍의 수}$)로 나누는 나머지 값들이 각각 0, 1, 2, ..., $2^{P+1}-1$ 에 따라 상태들을 분리한 후, 각각을 반으로 분리하게 된다. 따라서 SM 메모리는 이와 같은 상태배열로 총 2^{P+2} 개의 하위 메모리로 구성되게 된다. 1쌍의 ACS를 사용한 경우인 그림 7(b)를 보면 위와 같은 규칙성으로 상태들을 2로 나눈 후, 나머지가 0, 1에 따라 상태들을 분리하고 각각을 반으로 분리 하였음을 알 수 있다.

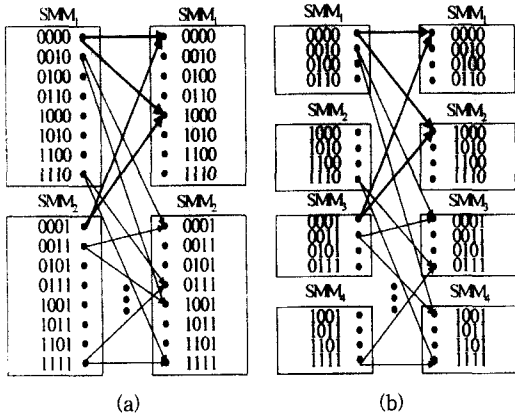


그림 7. SM 메모리 관리(구속장=5, 부호율=1/2, ACS 쌍의 수=1)
 Fig. 7. Management of the SM memory(Constraint Length=5, Code Rate=1/2, ACS pair=1).

SM부에서 SM 라우팅 블록은 ACS 수행시 상태값을 읽거나 저장하기 위해 해당되는 상태가 저장되어 있는 하위 메모리를 선택하기 위한 구조로 이루어져 있다. 그림 6에서 아래의 SM_SUB 블록은 구속장=5, 부호율=1/2, ACS쌍의 수=1인 경우의 구조를 보여준다. 이 구조에서 SM 라우팅 블록은 그림 6의 예처럼 switch 신호가 '1'인 경우 실선으로, '0'인 경우 점선으로 스위칭되어 하위 메모리의 CS/ 단자를 제어하는 구조로 이루어졌다. 여기서, switch가 '1'인 경우는 상태값의 읽기만을 수행하고 '0'인 경우는 상태값의 쓰기만을 수행한다. 라우팅 구조는 AP의 수에 따라 동일한 구조로 구현된다.

4. 가변적 TB부의 설계

TB(TraceBack)부는 ACS부에서 전달받은 경로 선택 정보인 역추적 정보와 역추적시 초기값으로 사용하는 최소 상태를 이용하여 1-포인트 알고리즘^[13]으로 데이터를 복호하는 기능을 수행한다. 역추적 깊이는 실험적으로 구속장의 5배 이상으로 해야 이상적인 비터비 복호 성능에 비해 거의 차이가 없는 것으로 알려져 있다^[14]. 그래서 본 논문에서는 역추적 깊이를 구속장이 9, 7, 6, 5인 경우 64, 64, 32, 32로 설정하여 사용하였다. 그림 8은 가변적 TB부의 구조를 보여준다.

TB부는 CL, AP의 사양 변수에 의해 영향을 받으며 CL은 상태의 비트 수와 역추적 깊이를 결정하고 AP는 역추적 메모리의 구조와 메모리 관리에 영향을 준다. TB부의 역추적 메모리는 SM 메모리처럼 동일

한 개수의 하위 메모리로 분리되어, ACS부의 S/P블록을 통해 동시에 각 분리된 메모리의 하위 주소부터 동시에 저장되게 된다. S/P 블록은 두 번의 ACS 수행마다 2^{P+2} ($2^P = \text{ACS 쌍의 수}$)개의 역추적 정보를 역추적 메모리에 전달한다. 그림 9의 예에서 ACS부의 S/P 블록을 통해 전달되는 상태인 (0000, 1000), (0001, 1001)의 역추적 정보 4개가 각각 분리된 메모리의 하위 주소부터 동시에 저장되게 된다.

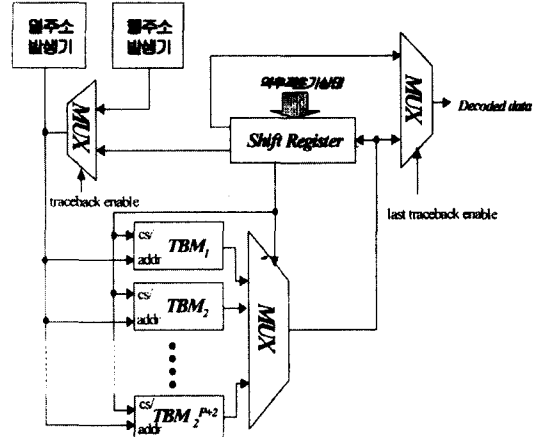


그림 8. 가변적 TB부 구조
 Fig. 8. Architecture of the parameterized TB module.

T	B	M	1
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0

T	B	M	2
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

T	B	M	3
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1

T	B	M	4
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

그림 9. 역추적 메모리의 상태배열(구속장=5, 부호율=1/2, ACS쌍의 수=1)
 Fig. 9. State array of the TB memory(Constraint length=5, Code rate=1/2, ACS pair=1).

TB부는 ACS부에서 전달된 역추적 초기값을 쉬프트 레지스터(Shift register)에 로딩하여 역추적을 수행한다. 프레임 모드에서는 제로 상태에서 시작해 제로 상태로 끝나기 때문에 (역추적깊이+구속장-1) 이후

부터 역추적을 수행하지만, 연속모드에서는 심볼의 ACS 처리 후 곧바로 역추적을 수행한다. 단지 이 모드에서는 안정된 디코딩 길이를 가져야 한다. 그림 8에서 쉬프트 레지스터에 저장된 값의 최상위 비트와 하위 비트(P~0)는 역추적 메모리인 TBM의 CS/ 단자의 입력값으로 메모리를 제어하고 동시에 이 값은 분리된 TBM 출력단에 있는 믹스(mux) 선택값으로 사용된다. 나머지 비트의 값들은 메모리의 주소로 사용된다. 위와 같은 동작을 역추적깊이 만큼 수행후 1비트의 복호데이터를 출력하게 된다. 그림 8에서 TB 부의 출력단에 있는 믹스의 선택값인 last traceback enable 신호는 프레임 모드에서 프레임의 마지막 데이터를 ACS 수행한 후, 연속적으로 데이터를 역추적하기 위한 제어신호이다.

5. 가변적 제어부의 설계

비터비 복호기를 제어하는 제어부는 그림 10과 같은 수행 상태도에 따라 설계된다. 제어부는 크게 심볼 제어카운터와 프레임제어카운터로 구현된다. 심볼제어 카운터는 기본 클럭을 단위로 심볼구간내의 제어를 위해 사용되고, 프레임제어카운터는 심볼단위로 프레임구간내의 제어를 위해 사용된다. 제어부는 CL, AP, SN의 사양 변수에 의해 설계된다. CL은 역추적 깊이를 결정하며, CL과 AP는 각 구간 제어신호를 발생시키는 심볼제어카운터에 영향을 준다. 그리고 CL과 SN은 프레임제어카운터에 영향을 준다.

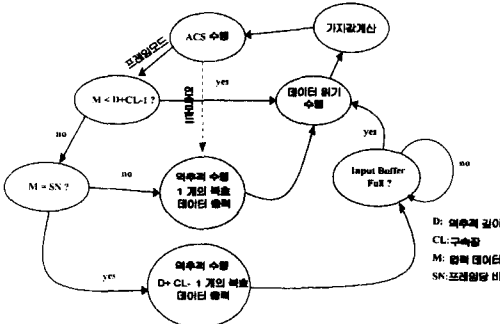


그림 10. 제어부의 수행상태도
Fig. 10. Execution state diagram of the Controller.

제어부는 입력 버퍼에서 한 프레임의 입력 심볼이 저장된 후, 입력단에서 복호단으로의 심볼 전송 준비 완료 신호인 input buffer full 신호를 발생하면 입력 버퍼에서 데이터를 읽어와 복호를 수행한다, 프레임

모드인 경우 ACS단을 수행하는 상태(S0), ACS 수행 후 역추적을 수행하여 1 비트의 복호 데이터를 출력하는 상태(S1) 그리고 ACS 수행 후 역추적과 동시에 프레임 안의 나머지 복호 데이터를 연속적으로 출력하는 제어 상태(S2)로 나뉜다. 반면에 연속 모드에서는 S1 제어 상태로만 복호를 수행한다.

IV. 자동 합성 결과 및 검증

본 가변적 비터비 복호기의 검증을 위해 IS-95 CDMA 시스템^[15]의 순방향 채널에서 필요한 비터비 복호기를 자동 합성하였다. 이 시스템은 트래픽 채널에서 9600, 4800, 2400, 1200bps의 가변적 데이터 전송율로 데이터를 전송하기 때문에 비터비 복호기는 4가지 전송율로 각각 복호를 수행한 후 송신한 전송율을 결정하게 된다. 따라서 1쌍의 ACS를 사용한 경우 4가지 전송율로 복호를 수행하기 위해 필요한 클럭의 수는 표 2와 같다.

표 2. 전송율별 복호 처리 시간
Table 2. Processing time of decoding as data rates.

제어상태 전송율(bps) & 비트수	S0	S1	S2	클럭 수
9600, 192bit	(128+3)x(64+7)	(128+3+64)x120	(128+3+72)x1	32904
4800, 96bit	(128+3)x(64+7)	(128+3+64)x24	(128+3+72)x1	14184
2400, 48bit	(128+3)x47	없음	(128+3+48)x1	6336
1200, 24bit	(128+3)x23	없음	(128+3+24)x1	3168
합계(클럭)	56592			

시스템 클럭은 9.8304 MHz의 CHIPX8을 사용하기 때문에 복호 수행의 전체 시간은 약 5.78ms가 된다. 따라서 입력단으로의 심볼 전송 지연과 출력단에서 외부 인터페이스로의 데이터 전달 시간을 합치더라도 프레임의 단위가 20ms이기 때문에 1쌍의 ACS만으로 충분히 복호가 가능하게 된다. 따라서 이 시스템에서 사용되는 사양변수는 다음과 같이 정의된다.

- 구속장=9
- 부호율=1/2
- 생성다항식=(753, 561)₀
- 연판정 비트 수=4
- 프레임당 비트 수=192, 96, 48, 24

· ACS쌍의 수=1

앞에서 정의된 사양변수를 사용하여 사용자 인터페이스 프로그램을 수행하고 synopsys의 VHDL 합성기로 자동 합성하였다. 그림 11은 합성된 논리회로를 보여주고 있다.

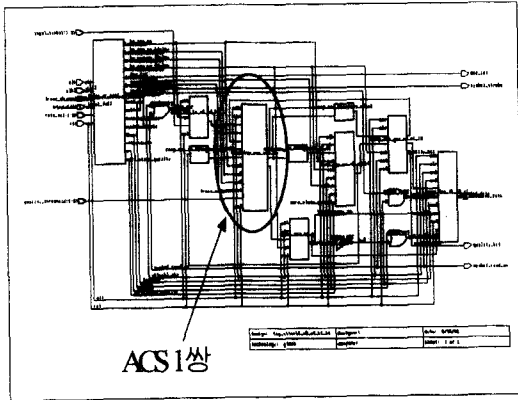


그림 11. 비터비 복호단의 논리합성(1쌍의 ACS)
Fig. 11. Logic Synthesis for Viterbi decoding module(one pair of ACS).

합성된 비터비 복호기 모듈의 전체 논리회로 게이트 수는 약 5600개이고 입력단과 출력단은 별도로 설계하였다. 그리고 자동 합성된 비터비 복호기의 검증을 위해 트래픽 채널에서 프레임모드로 전송율을 9600bps로 하고, 7dB의 가우시안 채널로 시뮬레이션 환경을 설정하였다. 그리고 MATLAB^[16] 프로그램을 사용하여 가우시안 채널환경하에서 192 비트의 한 프레임(20msec) 입력데이터를 추출하여 비터비 복호기의 테스트 벡터로 사용 하였다. 그림 12는 가우시안 채널환경(7dB)에서의 시뮬레이션 검증결과를 보여준다.

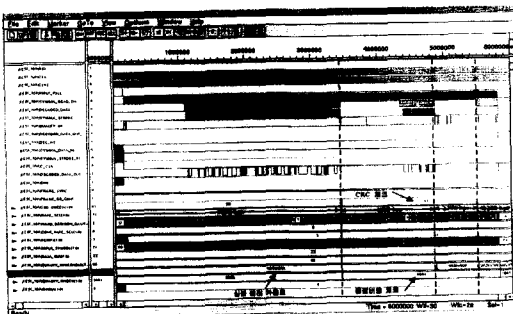


그림 12. 가우시안 채널 환경(7dB)에서의 시뮬레이션 결과
Fig. 12. Simulation result under gaussian channel (7dB).

이 그림은 각각 9600, 4800, 2400, 1200 bps의 4 가지 전송율로 복호를 수행한 결과를 보여준다. 전송율을 9600bps로 처리한 경우의 결과를 살펴보면 심볼 에러 개수가 0 개로 나와 완벽히 데이터를 복호하였음을 확인할 수 있었고, 또한 CRC 체크='1', 품질비트='1'로 나와 전송된 데이터 전송율이 9600bps로 보내졌음을 확인할 수 있었다.

V. 결론

본 논문에서 우리는 각 디지털 이동 통신 시스템마다 설계사양이 다른 비터비 복호기를 자동 생성할 수 있는 가변적 비터비 복호기를 제안하였다. 제안된 가변적 비터비 복호기는 설계사양에 영향을 줄 수 있는 설계변수 6개를 매개변수화하여, VHDL 언어의 설계기법을 이용해 원하는 비터비 복호기를 자동 합성하는 구조로 이루어 졌다. 이를 위해 사용자의 입력사양에 따라 원하는 비터비 복호기의 VHDL 코드를 생성시켜주는 사용자 인터페이스 환경을 C-언어로 구현하였다. 그리고 비터비 복호기의 내부 구조를 VHDL 언어의 generic 설계기법을 활용하여 계층 구조적으로 설계함으로써, 설계변수에 따라 원하는 구조가 생성 되도록 하였다. 그리고 설계된 복호기의 검증을 위해 IS-95 CDMA 시스템의 사양변수에 맞는 비터비 복호기를 생성하였고 이를 가우시안 채널 환경에서 시뮬레이션하여 복호가 정상적으로 이루어지는 것을 확인할 수 있었다. 본 논문에서 제안한 가변적 비터비 복호기의 설계방법은 설계변수의 변화로 인한 재설계의 부담없이 변경된 사양변수만을 제공함으로써 빠른 시간내에 원하는 비터비 복호기를 얻게 되는 새로운 방식을 제공하고 있다.

앞으로 제안된 방법을 기반으로 사용자에게 다양한 구조의 비터비 복호기를 제공할 수 있는 DB 구축을 위해 처리속도, 전력 소비등을 고려한 다양한 구조를 갖는 비터비 복호기의 라이브러리 구축과 사용자 선택 조건을 입력으로 받아 이에 적합한 구조의 비터비 복호기를 얻을 수 있도록 사용자 인터페이스부의 고도화, 지능화하는 방향으로 연구가 필요하다.

감사의 글

※ 본 연구는 1997년도 현대전자 (주)와 1996년도 연

세대학교 학술 연구비 일부 지원에 의하여 이루어진 것임.

참 고 문 헌

- [1] J. Anderson, S. Mohan, "Source and Channel Coding: An Algorithmic Approach", Kluwer Academic Pub., Boston, 1991.
- [2] Shu Lin, D. Costello, "Error Control Coding: Fundamentals and Application", Prentice-Hall Inc., New Jersey, pp. 330-331, 1983.
- [3] J. Cain, G. Clark, J. Geist, "Punctured Convolutional Codes of Rate $(n-1)/n$ and Simplified Maximum Likelihood Decoding", IEEE Trans. Inform. Theory, vol. IT-25, no. 1, pp. 97-100, Jan. 1979.
- [4] G. D. Forney, "The Viterbi Algorithm", Proc. of The IEEE, vol. 61, no. 3, pp. 268-278, Mar. 1973.
- [5] J. Omura, "On the Viterbi Decoding Algorithm", IEEE Trans. Inform. Theory, vol. IT-15, no. 1, pp. 177-179, Jan. 1969.
- [6] T. Truong, M. Shin, I. Reed and E. H. Satorius, "A VLSI Design for a TraceBack Viterbi Decoder", IEEE Trans. on Comm., vol. 40, no. 3, pp. 616-624, Mar. 1992.
- [7] C. Chang, K. Yao, "Systolic Array Processing of the Viterbi Algorithm", IEEE Trans. Inform. Theory, vol. IT-35, no. 1, pp. 76-86, Jan. 1989.
- [8] M. B'oo, F. Arguello, J. D. Bruguera, "High-Performance VLSI Architecture for the Viterbi Algorithm", IEEE Trans. on Comm., vol. 45, no. 2, pp. 168-176, Feb. 1997.
- [9] P. Gulak, T. Kailath, "Locally connected VLSI Architectures for the Viterbi Algorithm", IEEE J. Sel. Areas Comm., vol. 6, no. 3, pp. 527-537, Apr. 1988.
- [10] B. Sklar, "Digital Communication", Prentice-Hall Inc., pp. 327-331, 1988.
- [11] 정승훈, 이현주, 정인택, 주유상, 노예철, 박장현, 송상섭, "W-CDMA 방식 PCS의 이동국용 SVD 설계", 대한전자공학회 추계학술논문집, vol. 18, no. 2, pp. 226-229, 1995년 12월
- [12] 차진중, 현진일, 강우, 김재석, 김정주, "CDMA 이동통신 시스템용 비터비 복호기 ASIC 설계 및 구현", 전자통신학회논문집, 제 33권 A편, 제1호, pp. 139~152, 1996년 1월
- [13] Charles M. Radar, "Memory Management in Viterbi Decoder", IEEE Trans. on Comm., vol. COM-29, no. 9, pp. 1399-1401, Sept. 1981.
- [14] J. G. Proakis, "Digital Communications", McGraw-Hill Inc., pp. 483~486, 1995.
- [15] TIA/EIA IS-95 Interim standard, TIA, July 1993.
- [16] J. G. Proakis, M. Salehi, "Contemporary Communication System", PWS Pub., Boston, 1997.

저 자 소 개



孔明 晔(學生會員)

1971년 5월 15일생. 1997년 2월 연세대학교 공과대학 전자공학과 학사. 1997년 3월 ~ 현재 연세대학교 공과대학 전자공학과 석사과정. 주관심 분야는 이동통신 시스템용 Modem ASIC 설계



裴 晟 日(學生會員)

1975년 7월 27일생. 1998년 2월 경북대학교 공과대학 전자공학과 학사. 1998년 3월 ~ 현재 연세대학교 공과대학 전자공학과 석사과정. 주관심 분야는 이동통신 시스템용 Modem ASIC 설계



金 在 錫(正會員)

1955년 10월 1일생. 1977년 2월 연세대학교 공과대학 전자공학과 학사. 1979년 2월 한국 과학원 전기 및 전자공학과 석사. 1988년 8월 Rensselaer Polytechnic Institute 전자공학과 박사. 1979년 2월 ~ 1984년 4월 전자기술 연구소 선임 연구원. 1988년 8월 ~ 1993년 5월 AT&T Bell Lab. MTS(Member of Technical Staff). 1993년 5월 ~ 1996년 2월 한국 전자통신 연구소 VLSI 구조 연구실장. 1996년 2월 ~ 현재 연세대학교 전자공학과 부교수. 주관심분야는 VLSI 설계와 CAD, 고속 DSP 설계, 디지털 통신 및 영상압축용 ASIC 설계