

On-the-fly 인터리빙 방식의 터보코덱의 아키텍처 설계

이 성 규[†] · 송 낙 운^{††} · 계 영 철^{††}

요 약

본 논문에서는 IMT-2000용 터보코덱의 개선된 아키텍처를 제안하였다. 이는 on-the-fly 인터리빙 쉬프트 레지스터를 이용하여 외부 RAM을 사용하지 않는 인터리버에 의한 부호기와 필요한 메모리의 양을 줄이기 위한 복호기로 구성되었다. 제안된 구조는 C/VHDL 언어를 이용하여 시뮬레이션을 수행하여 반복횟수, 인터리버 블록크기, 부호율에 따른 비트오류율 성능이 이전 데이터와 비교적 일치함을 확인하였다.

Architecture Design of Turbo Codec using on-the-fly interleaving

Sunggyu Lee[†] · Nagun Song^{††} · Yongchul Kay^{††}

ABSTRACT

In this paper, an improved architecture of turbo codec for IMT-2000 is proposed. The encoder consists of an interleaver using an on-the-fly type address generator and a modified shift register instead of an external RAM, and the decoder uses a decreased number of RAM. The proposed architecture is simulated with C/VHDL languages, where BER (bit-error-rate) performances are generally in agreement with previous data by varying interation numbers, interleaver block sizes and code rates.

키워드 : 터보 코덱(Turbo Codec), 온더플라이 인터리빙(On-the-fly Interleaving), 에러정정(Error Correction)

1. 서 론

최근 몇 년간 무선 이동통신 서비스가 널리 상용화되면서 고속의 보다 나은 품질을 제공하는 통신시스템의 개발이 이루어지고 있다. 이 경우 제한된 대역폭에서 고품질의 서비스를 제공하기 위하여서는, 성능 개선을 위한 강력한 채널코딩 기법이 필요하게 되며, 이를 위한 한 방법은 블록부호 또는 길쌈부호를 연결하는 것이다. 이런 부호들은 대개 시간이 오래 걸리는 복호 알고리즘을 이용해야만 하는데, 이 경우 연결부호들은 부호 및 복호화하는 과정을 보다 간단한 단계로 나누어 처리할 수 있는 장점이 있게 되며, 터보부호는 이전에 제안된 직렬 연결부호를 이용한 방법보다 성능과 구현측면에서 더 강력한 성능을 보이며, 3세대 이동통신인 IMT-2000의 관련 표준인 CDMA-2000과 WCDMA에 32kbps 이상의 고속 데이터통신을 위한 순방향 오류정정부호로 채택이 되었다[1, 2].

터보부호에 적합한 연판정 값을 이용하여 복호하는 알고리즘은 MAP(Maximum A Posteriori) 알고리즘계열과 Viterbi 알고리즘계열이 있다[3-5]. 1974년에 제안되어 BCJR 알

고리즘[3]으로 알려져 있는 MAP 알고리즘 계열(MAP, Log-MAP, Max-Log-MAP)이 있으며, Hagenauer 등에 의해 제안된 SOVA(Soft Output Viterbi Algorithm)[6]가 있다. 이는 MAP 계열보다 성능은 다소 떨어지지만, 하드웨어 구현이 비교적 용이한 것으로 알려져 있다[7].

터보부호기는 조직형 재귀구조(systematic recursive structure)를 갖으며, 일반적으로 직병렬 연결한 두개 이상의 부호기(예, 길쌈부호기)로 구성되어 있으며 인터리버를 통해 순서가 바뀐 뒤에 천공(puncturing)을 통해 원하는 부호율로 부호화시킨 후 직-병렬 변환기를 통해 송신된다. 이에 사용되는 인터리버는 랜덤, 블록, 나선(helical), GF(Galois field), Mother 등이 있으며[8], 일반적으로 블록크기가 큰 경우에는 랜덤 인터리버가, 블록크기가 작은 경우에는 블록 인터리버가 성능이 좋은 편이나, 이중 주소화에 의한 최적화 구조에 관한 연구가 있다[9]. 이의 구현은 모든 어드레스를 저장하는 ROM 테이블 방법, on-the-fly 방법[10-12], 길쌈부호기를 이용한 방법 등이 있다. 다음으로, 트렐리스 종료는 성능의 차이가 인터리버와 관련이 높기 때문에 코덱에서 채택한 인터리버의 종류에 따라 결정되어야 한다[13]. 채널을 통해 전송된 부호어는 복호기로 공급이 되는데, 이는 병-직렬 변환기를 통해 역천공(depuncturing)되어 디인터리버와 연판정을 위한 구성부호기에 대응되는 복호

[†] 정 회 원 : 삼성전자

^{††} 정 회 원 : 홍익대학교 전자전기공학과 교수

논문접수 : 2002년 9월 17일, 심사완료 : 2003년 2월 21일

단의 상호연결로 이루어져 있으며 수신된 시퀀스의 복호는 연판정을 위한 사후 확률값을 계산하는 알고리즘을 통해 반복복호함으로써 이루어진다[14]. 아울러 알고리즘과 연관된 아키텍처 및 성능에 관한 다양한 연구가 있다[15-19].

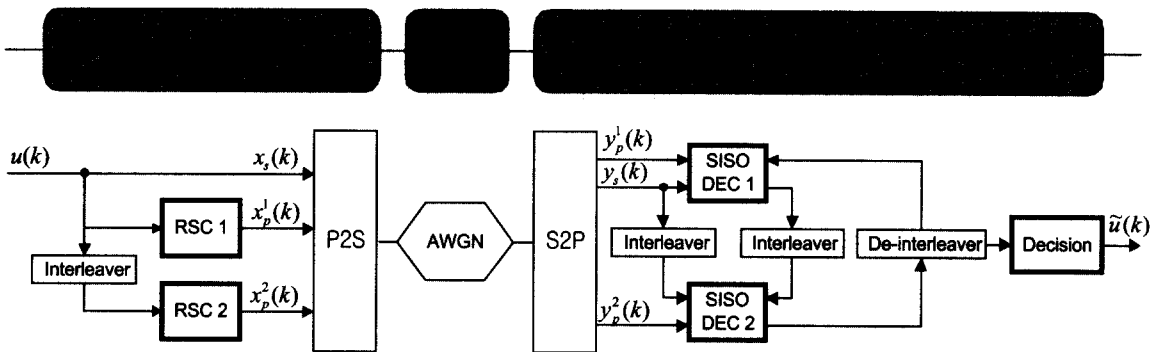
터보부호의 성능을 향상시키기 위해 랜덤 인터리버를 사용하는 경우 인터리버의 주소를 저장해야 하며 각 전송에 따른 ROM의 증가와 이에 따른 제어블록의 증가 및 신호 지연 등 하드웨어적인 복잡도를 증가시키는 요인이 된다. 이를 개선하기 위하여 본 논문에서는, SOVA 방식에 비해 성능이 우수한 Log-Map을 채택하고, on-the-fly 방식[15-18]에 의한 인터리버의 부호기 아키텍처를 채택하여 다양한 전송률에 따라 모든 인터리빙 주소값을 ROM 테이블에 저장하지 않고, 계산에 의해 만들어 내어 이에 따른 메모리 관련 하드웨어감소와 랜덤성유지가 가능하도록 하였으며, 메모리 사용을 줄이기 위한 구성복호기의 아키텍처를 제안하고 C/VHDL 언어를 이용하여 시뮬레이션하였다. 본 논문의 구성은 2장에서는 부호기와 복호기의 아키텍처를 제안하였으며, 3장에서는 시뮬레이션을, 마지막으로 4장에서는 결론을 기술하였다.

2. 아키텍처 설계

간략화된 터보 코덱의 아키텍처는 (그림 1)과 같다.

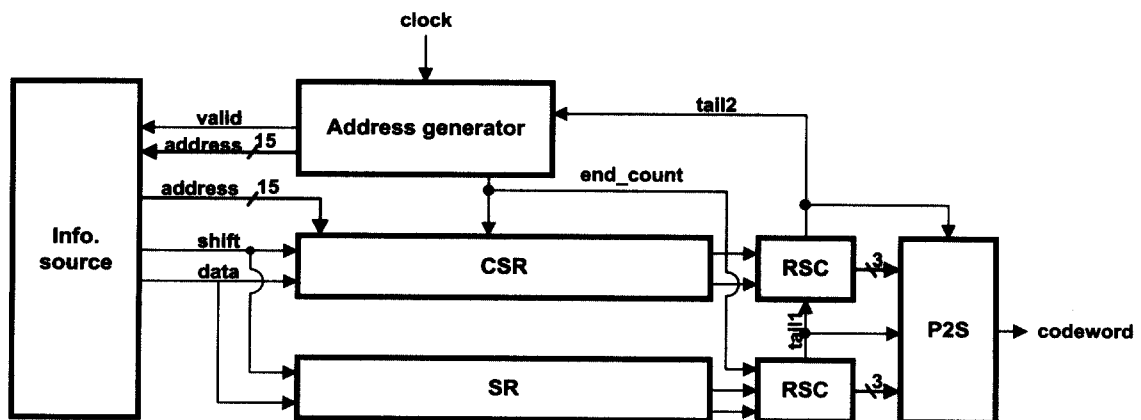
부호기는 원래의 데이터와 인터리버를 통해 순서가 바뀐 데이터가 순차적으로 입력되므로, 외부 RAM을 이용하지 않고 쉬프트 레지스터를 이용한 아키텍처를 제안하였다. 이를 통해 RAM 액세스 타임과 원칩화를 통한 속도의 향상을 얻을 수 있다. 제안된 부호기와 인터리버를 결합하기 위하여, 제어비트로 다른 레지스터의 내용을 입력할수 있도록 설계한 쉬프트 레지스터의 사용을 제안하여, 부호기의 동작이 연속적으로 이루어지게 하였다.

한편 복호기의 기반이 되는 MAP 알고리즘은 최종 복호를 하기 위해 많은 데이터를 필요로 하므로 이를 위해 많은 양의 메모리가 필요하게 되는데 쉬프트 레지스터를 이용하여 꼭 필요한 값만을 저장하고 필요한 값들이 순차적으로 해당 블록에 공급되는 아키텍처를 제안하여 이를 통해 외부 메모리의 사용과 액세스타임을 줄여 전체 복호 지연시간을 줄일수 있게 하였다.



Turbo codec structure

(그림 1) 터보 코덱의 전체 아키텍처



(그림 2) 제안된 부호기

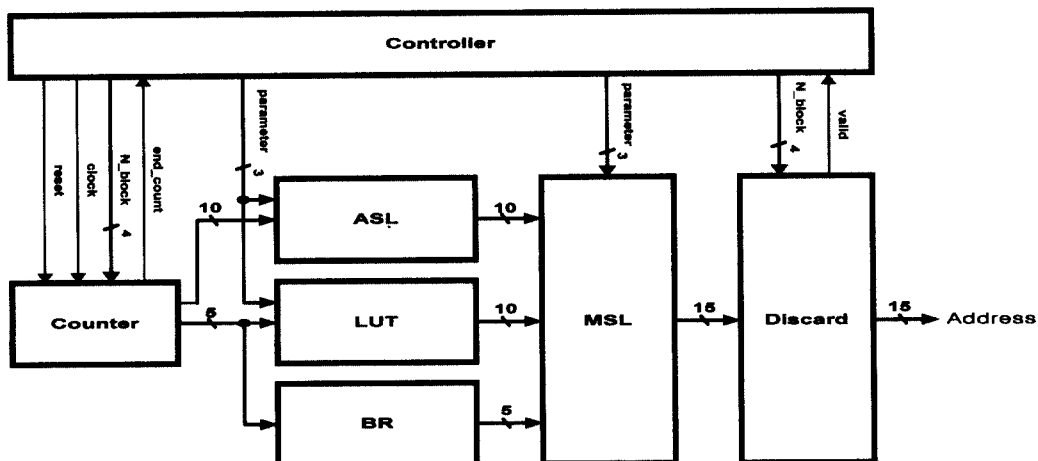
2.1 부호기의 설계

일반적으로 부호기의 입력 비트들은 RAM에 저장되어 인터리빙을 거친 후, 원래의 순서대로 비트들이 저장된 RAM과 인터리버를 통해 순서가 바뀌어 저장된 RAM에서 데이터를 순차적으로 읽어 구성 부호기에 입력된다[16]. 본 논문에서 제안된 부호기 아키텍처는 외부 RAM을 이용하지 않고, 쉬프트 레지스터를 이용하여 구성하였으며 RAM을 읽기(read), 쓰기(write), 수정(modify)하는 액세스 시간을 줄여 고속 동작에 더 적합하도록 하였다. 다음 그림에 이의 구조를 보였다.

여기서 CDMA2000 표준을 기반으로 1/2, 1/3, 1/4의 부호율로 부호화할 수 있으며, 인터리버의 블록크기는 378, 570, 762, 1146, 1530, 2298, 3066, 4602, 6138, 9210, 12282, 20730비트 중 전송률에 맞추어 선택할 수 있도록 하였다[21]. 또한 CDMA-2000에 채택된 주소생성기를 참고하여, 이에 다양한 전송률에 따라 모든 인터리빙 주소 값을 계산에 의해 주소를 만들어내는 on-the-fly 방식으로 모든 주소들을 저장하지 않고 계산에 의해 만들어내는 아키텍처를 설계하였으며, 이를 위한 주소 발생순서와 관련 인터리버표와 파라미터표를 구하였다. 여기에서 부호기로 들어오는 모든 데이터는 버퍼에 저장하고, 인터리버 블록크기만큼의 버퍼가 차면 인터리빙하여 두 번째 구성 부호기에 입력되는 버퍼에 값을 저장하였다. 인터리빙이 되면 순차적으로 구성부호기에 데이터가 공급이 되고, 버퍼에 있는 데이터가 모두 처리되면 첫 번째 구성 부호기의 트렐리스 종료를 위한 테일비트가 추가된 뒤, 두 번째 구성 부호기를 위한 테일비트가 추가된다. 모든 데이터는 천공기에 의해 정해진 부호율로 천공되어 데이터를 출력하도록 하였다. 이러한 방식에 의하여 주소 생성기를 다음과 같이 설계하였다. (그림 3)의 아키텍처는 전송률이 바뀔 때마다 참조 테이블, 덧셈기, 곱셈기, 비교기, 카운터 등을 이용하여 인터리빙 주소 값을 생성한다.

카운터블록은 제어기로부터 reset, clock, N_block, end_count를 입력으로 받고 ASL(Add-Select-LSBs) 블록에 카운터 값의 상위 n비트를, LUT(LookUp Table) 블록과 BR (Bit Reverse) 블록에 카운터 값의 하위 5비트 값을 보낸다. 인터리버 파라미터 n값은 N_block 값으로부터 얻을 수 있으며, 인터리버표에서처럼 4에서 10까지의 값을 가질 수 있다. 작은 블록 크기에서 쓰지 않는 비트들은 버려진다. 카운터는 입력클록이 rising edge일 때 출력값을 1씩 증가시키며, 그 값이 파라미터에 의해 정해진 값에 도달하면 end_count 신호를 통해 제어기블록에 알린다. 다음으로 ASL 블록은 제어기로부터 공급받은 파라미터 값을 참조하여 카운터로부터 전달받은 10비트 중 필요한 비트만 추출하여 그 값에 1을 더해 임시값을 만든다. 다시 임시값 중 파라미터값과 같은 길이의 비트값을 MSL 블록에 전달한다. LUT 블록은 파라미터표의 값을 카운터로부터 전달된 값을 입력으로 하여 파라미터 값을 참조하여 해당하는 값을 MSL 블록에 전달한다. BR 블록은 카운터로부터 전달된 값을 비트 반전하여 MSL 블록에 전달한다. MSL(Multiply-Select-LSBs) 블록은 ASL 블록과 LUT 블록으로부터 받은 10 비트 길이의 두 값을 곱한 결과값에 파라미터 값을 참조하여 n개의 최하위 비트들 값을 만든다. 그리고, BR 블록으로부터 전달 받은 5비트를 최상위 비트로 하여 곱해진 결과값과 연결하여 discard 블록으로 전달한다. Discard 블록은 N_block값과 MSL 블록으로부터 전달받은 값을 비교하여 같거나 크면 값을 버리고, 작으면 올바른 주소값으로 출력한다. 출력되는 주소값이 올바른지는 valid 신호를 통해 제어기에 전달된다.

이제 주소생성기 이외의 블록을 본다. SR 블록은 보통의 비동기 쉬프트 레지스터로 정보원으로부터 쉬프트 신호가 들어오면 레지스터의 내용을 우측으로 이동하고 최좌측 레지스터에 데이터 값을 쓴다. CSR 블록은 controlled SR로써 on-the-fly 방식으로 구현하기 위해 만들어진 특수한 쉬



(그림 3) 주소 생성기 블록

프트 레지스터이다. 이 블록은 SR 블록의 구조에서 각각의 레지스터에 D플립플롭이 추가된 형태를 가진다. 정보원 블록으로부터 주소값이 전달되면 이 CSR 블록은 쉬프트 레지스터의 최좌측 레지스터에 값을 쓰는 것이 아니고, 주소값에 해당하는 레지스터의 D플립플롭에 값을 저장한다. 쉬프트 레지스터의 내용은 하나씩 우로 쉬프트된다. 카운터가 종료되면 주소 생성기의 end_count 신호가 CSR 블록에 전달되어, D플립플롭의 내용이 모두 쉬프트 레지스터에 기록이 된다.

SR 블록과 CSR 블록의 최우측으로부터 출력되는 비트는 동일한 RSC(Recursive Systematic Code) 블록에 전달이 되고, 부호화가 진행된다.

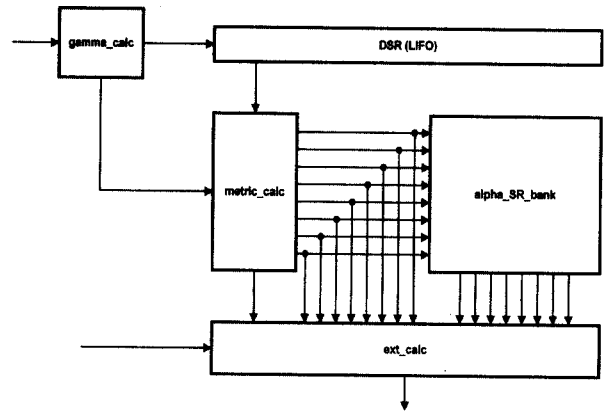
P2S 블록은 병-직렬변환기 역할과 천공기의 역할을 한다. 병-직렬 변환기는 복호기에 데이터를 전달하기 위해 병렬로 처리된 데이터를 직렬로 바꾸는 동작을 하고, 천공기는 원하는 부호율을 맞추기 위해 특정한 패턴에 따라 생성된 부호심볼들을 삭제한다.

터보부호기에서 트렐리스 종료는 각각의 부호기에 따른 테일비트를 따로 전송하는 방법을 사용한다. 주소생성기에서 end_count 신호가 전달되면 전체 부호어 블록에서 데이터비트 구간은 끝나고 테일비트 구간이 시작되었다는 것을 나타낸다. 인터리버를 거치지 않은 RSC 구성 부호기가 먼저 트렐리스 종료를 위한 테일비트가 생성되고, 그 뒤에 다른 구성부호기의 테일비트가 만들어진다. 테일비트 구간이 끝나면, 주소생성기는 다시 리셋되어 0부터 카운트가 다시 시작된다.

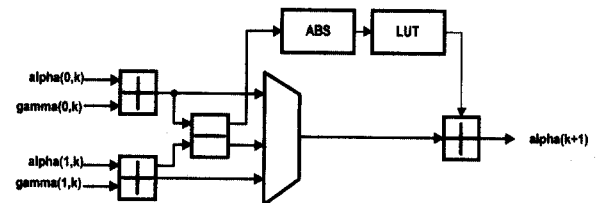
2.2 복호기의 설계

터보복호기는 구성복호기, 인터리버, 디인터리버로 이루어져 있다. 구성복호기는 연판정 매트릭 값을 입력으로 받고 연판정 매트릭 값을 출력으로 내보내며, 하나의 구성복호기는 다른 구성복호기의 출력과 가지 매트릭 값을 전달 받는다. 복호기는 채널을 통과한 데이터를 입력으로 받아들이며, 역천공기에 의해 각각의 구성부호기에 데이터를 분배한다. 부호기에서 천공된 데이터는 모두 0으로 바뀌어 입력된다. 구성복호기는 MAP 알고리즘에 의해 외부 정보값을 출력하고, 이 정보는 다음 구성복호기에 인터리빙 또는 디인터리빙되어 입력된다. 원하는 횟수만큼 반복한 후 각 구성복호기로부터 출력되는 외부 정보값과 정보 비트의 매트릭 값을 더하여 양수이면 1, 음수이면 0으로 경판정하여 최종 복호한다. 다음은 제안된 구성복호기의 구조이다.

우선 매트릭계산 블록의 설계를 한다. 알파 매트릭 값과 베타 매트릭 값을 계산하는 과정은 공급되는 가지 매트릭 값의 순서가 순방향 또는 역방향인 것을 빼면 동일한 구조를 갖는다. (그림 5)는 Log-MAP 알고리즘을 이용했을 경우 알파 매트릭 값을 예를 들어 설계하였다[20].



(그림 4) 제안된 복호기



(그림 5) 매트릭 계산 블록

알파 매트릭값과 가지 매트릭값을 더하여 얻은 결과들을 비교기를 통해 큰 값을 선택한다. 그와 동시에 그 결과들끼리 빼주어 절대 값을 얻고, 그 값을 미리 저장한 테이블을 이용하여 보정 값을 얻는다. 앞에서 비교기를 통해 얻은 값과 보정 값을 더하여 다음 알파 매트릭 값을 얻는다.

여기서 metric_calc 블록은 전절에서 설명한 recursion 블록이며, DSR 블록은 쉬프트되는 방향을 정할 수 있는 쉬프트 레지스터이며, LIFO(Last Input First OUT) 버퍼를 이용할 수도 있다. 구성 복호기로 입력되는 값이 gamma_calc 블록에 의해 가지 매트릭이 계산된 다음 DSR 블록에 우로 쉬프트되며 입력이 된다. 그와 동시에 metric_calc 블록에서는 알파 매트릭 값을 alpha_SR_bank 블록에 저장한다. alpha_SR_bank 블록은 여러개의 쉬프트 레지스터로 이루어져 있으며, 각각의 레지스터는 매트릭 값의 워드길기와 같은 크기를 갖는다. 알파 매트릭 값이 모두 계산이 되면, DSR 블록에 이미 저장된 가지 매트릭 값을 좌로 쉬프트하여 metric_calc 블록에 다시 입력되는데, 이 때는 결과값을 저장하지 않고 바로 ext_calc 블록에 보내어 alpha_SR_bank 블록에서 출력되는 알파 매트릭 값과 다른 구성복호기로부터 받는 값을 가지고 추가정보를 계산한다.

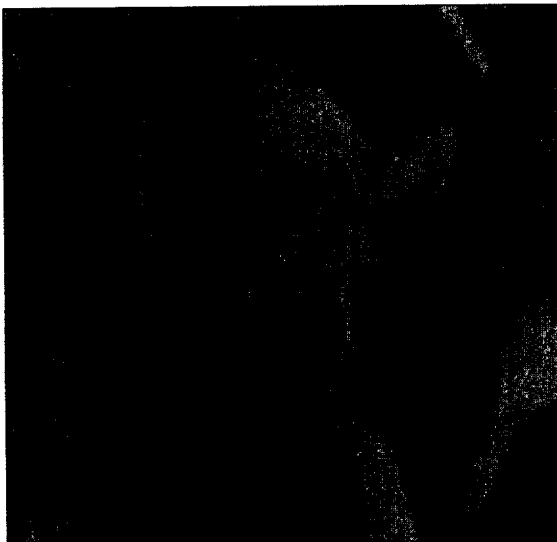
제안된 아키텍처는 매트릭 계산을 위한 블록을 하나로 줄였고, 쉬프트 레지스터를 이용하여 매트릭 계산시 필요한 값들을 순차적으로 공급할 수 있게 하였다. 또, 베타 매트릭값을 저장하지 않고, 다른 구성 복호기에 필요한 추가 정보값만을 저장하였다. 이 저장된 데이터는 인터리빙 또는 디인터리빙되어 다른 구성복호기에 전달된다.

3. 시뮬레이션 및 결과검토

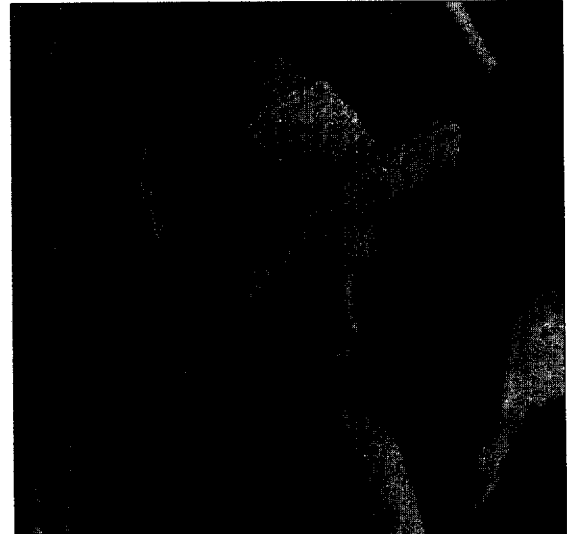
본 논문에서는 Log-Map 알고리즘 검증에 C언어를 사용하였으며, 먼저 채널부호화 시뮬레이션에 필요한 BPSK로 변조했을 경우의 AWGN 채널을 모델링하였으며, 복호시에는 Log-MAP 알고리즘과 최대값을 빼는 방법으로 정규화하였다. 여기에서 임의의 데이터를 송신하여 수신기에서 받은 데이터와 송신기에서 보낸 데이터를 비교하여 비트오류율을 구하였다. 제안된 아키텍처는 IMT-2000 관련 표준인 CDMA-2000에 제시된 터보부호기의 구성을 바탕으로 IDEC C-632 라이브러리를 사용하며, C, VHDL언어를 이용하여 시뮬레이션하였다. 아키텍처의 검증은 VHDL로 functional 시뮬레이션을 수행하여 다시 C로 변환하여 이미지정

정의 정상적인 동작을 확인하였다.

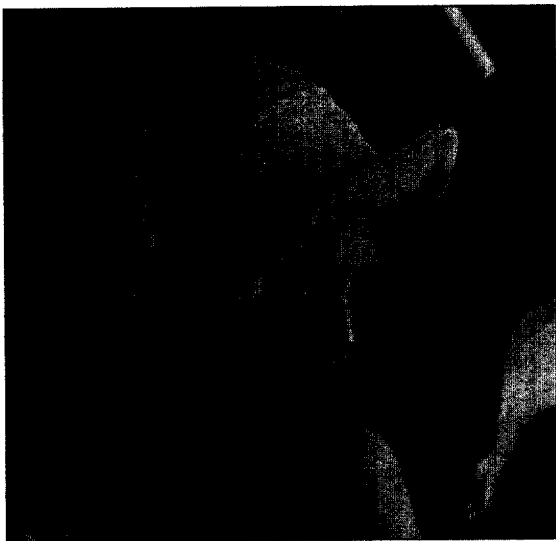
(그림 6)은 512×512 크기의 Lena 이미지(블록크기 ; 378 비트)를 이용하여 1.8dB의 신호대 잡음비에서 매 반복복호마다 결과를 출력하였다. 반복횟수가 증가할수록 오류가 정정되는 것을 확인할 수 있다. (그림 7)은 부호율 1/2일 경우, 블록크기의 변화에 따른 비트오류율 커브이다. 블록크기가 커질수록 더 좋은 성능을 내는 것을 알 수 있다. (그림 8)은 블록크기가 378일 경우, 부호율에 따른 비트오류율 커브이다. 부호율이 낮을수록 더 좋은 성능을 내는 것을 알 수 있다. 이를 통하여 기존논문[1, 2, 8]에서의 결과와 유사한 성능을 확인할수 있었으며, 일부 성능이 기존논문에 비해 약간 저하되는 부분은 양자화시 오차로 기인한 것으로 사료된다.



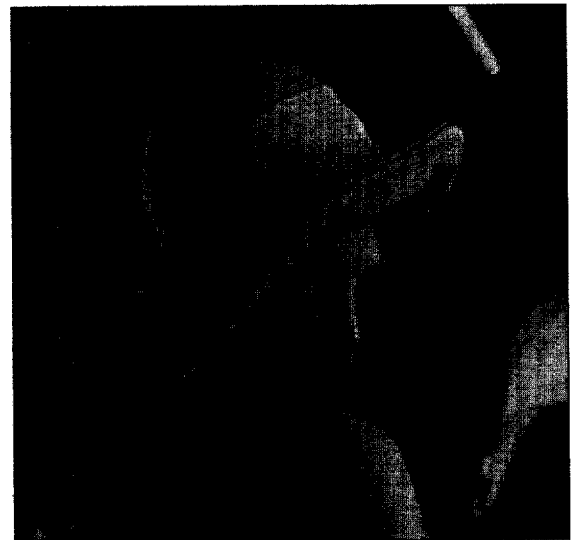
(a) 1회



(b) 2회

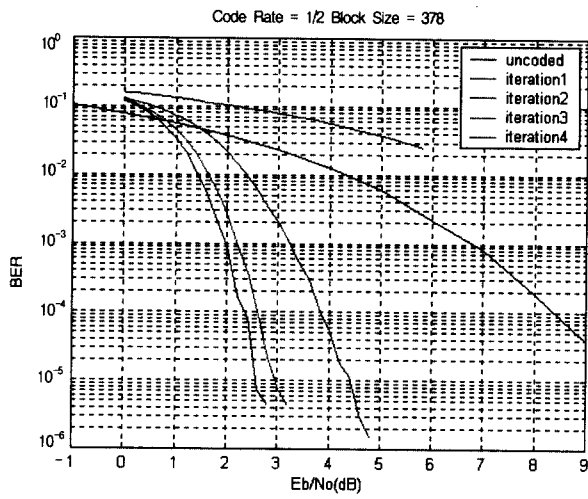


(c) 3회

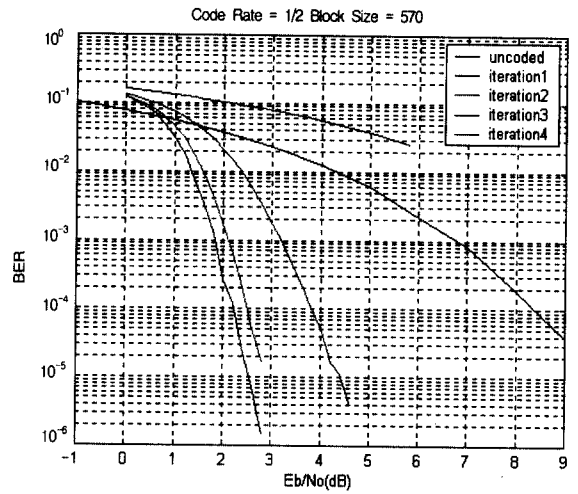


(d) 4회

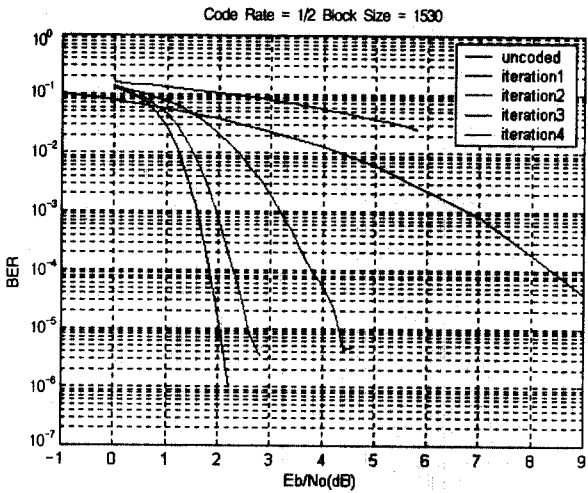
(그림 6) 반복횟수에 따른 시뮬레이션 결과



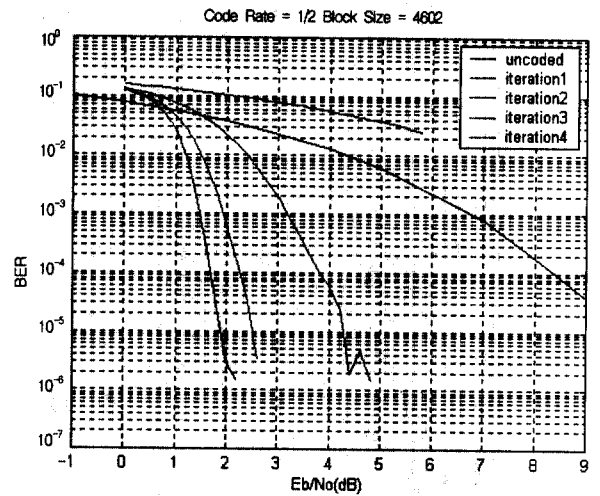
(a) 블록 크기 : 378



(b) 블록 크기 : 570

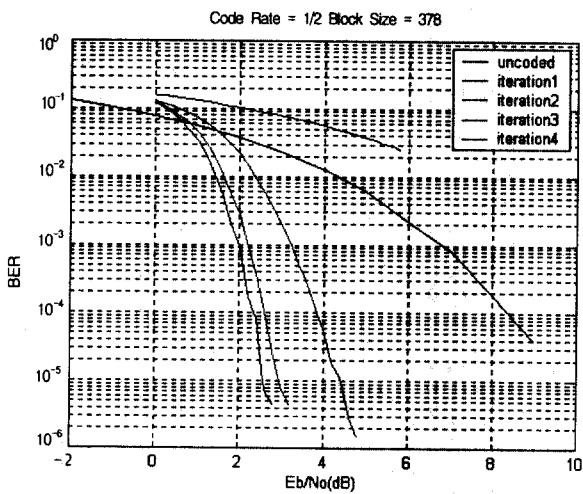


(c) 블록 크기 : 1530

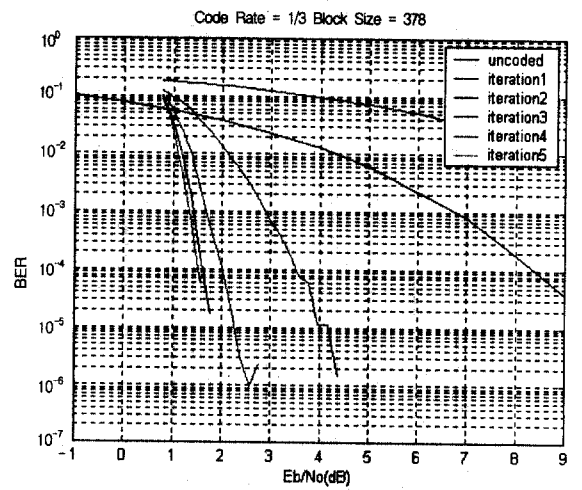


(d) 블록 크기 : 4602

(그림 7) 블록크기에 따른 시뮬레이션 결과(반복횟수가 증가할수록 아래방향으로 위치함)



(a) 부호율 : 1/2



(b) 부호율 : 1/3

(그림 8) 부호율에 따른 시뮬레이션 결과(반복횟수가 증가할수록 아래방향으로 위치함)

4. 결론 및 향후 연구과제

본 논문에서는 IMT-2000 이동통신 시스템 기술의 핵심인 터보부호의 아키텍처를 제안하였다. 이때 BPSK 변조와 AWGN 채널을 가정하였으며, 복호기는 Log-MAP 알고리즘과 최대값을 빼는 방법으로 정규화하였다. 부호기의 경우, on-the-fly 방식을 이용한 인터리버를 통해 다양한 전송률에 맞추어 어드레스를 생성하도록 하였으며, 부호기와 복호기에 데이터가 순차적으로 입력되는 부분에 변형된 쉬프트 레지스터를 이용하여 외부 RAM과 ROM의 사용을 가능한 줄여서 액세스 타임과 IC 사이의 연결로 인한 전체 지연을 줄이고자 하였다.

제안된 구조에 의한 시뮬레이션을 통해 부호율과 인터리버 블록크기에 따른 다양한 부호화이득을 얻을 수 있었는데, 반복횟수, 블록크기, 부호율에 따른 정상적인 동작을 확인할 수 있었으며, 성능이 기존논문에 비해 약간 저하되는 부분은 양자화시 오차로 기인한 것으로 사료된다.

앞으로의 개선 과제는 첫째, 알파와 베타값을 저장할 때 최적의 결과를 얻을 수 있는 양자화 레벨과 스텝을 모의실험을 통해 구하는 것과 둘째, 저장소자 블록감소와 이의 연산계어블록의 증가는 상충하는 면이 있어 이를 체계적으로 분석하여 최적화하는 문제가 있으며, 마지막으로 복호기의 저전력, 소형화설계를 블록별로 또한 전체적으로 분석하여 이를 최적화 설계에 응용하는 문제가 있다.

참 고 문 헌

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo codes," Proc. of the IEEE ICC '93, Geneva, Switzerland, pp. 1064-1070, May, 1993.
- [2] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding ; Turbo codes," IEEE Trans. Commun., Vol.44, No.10, pp.1261-1271, Oct., 1996.
- [3] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimum Decoding of Linear Codes or Minimizing Symbol Error Rate," IEEE Trans. Inform. Theory, Vol.20, pp.284-287, Mar., 1974.
- [4] S. S. Pietrobon and A. S. Babulscu, "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional Codes," IEEE Int. Symp. Inform. Theory and its Applications, pp.1073-1077, Nov., 1994.
- [5] A. J. Viterbi, "Error bounds for convolutional codes and its asymptotically optimum decoding algorithms," IEEE T-IT, Vol.13, No.4, pp.260-260, Apr., 1967.
- [6] J. Hagenauer, P. Hoehner, "A Viterbi algorithm with soft-decision outputs and its applications," IEEE Glo. Telecom, Conf., pp.1680-1689, 1989.
- [7] P. Robertson, E. Villebrun, P. Hoehner, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," IEEE, Int. Conf. on Communications, pp.1009-1013, 1995.
- [8] 진익수 외 3인, "터보부호의 인터리버 분석", ETRI 주간기술동향, Vol.905, 1999.
- [9] J. Hokfelt et al., "Interleaver design for turbo codes based on the performance of iterative decoding," IEEE Int. Conf. Com., pp.93-97, 1999.
- [10] Y. X. Lee et al., "An on-the-fly decoding technique for Reed-Solomon codes," IEEE T-Mag., Vol.32, No.5, pp.3962-3964, Sept., 1996.
- [11] C. Frougny, "On-the-fly algorithm and sequential machines," IEEE Sym. on computer arithmetic, pp. 260-265, July, 1997.
- [12] D. D. Deavours et al., "On-the-fly solution techniques for stochastic petri nets and extensions," IEEE T-SE, Vol.24, No.10, pp.889-902, Oct., 1998.
- [13] J. Hokfelt, O. Edfors and T. Maseng, "A Survey on Trellis Termination Alternatives for Turbo Codes," 1999 IEEE Vehicular Technology Conference, Vol.3, pp.2225-2229, 1999.
- [14] J. Hagenauer, E. Offer and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," IEEE Trans. Inform. Theory, Vol.42, No.2, pp.429-445, Mar., 1996.
- [15] G. Masera, G. Piccinini, M. R. Roch and M. Zamboni, "VLSI Architectures for Turbo codes," IEEE Trans. on VLSI System, Vol.7, No.3, pp.369-379, Sept., 1999.
- [16] H. Suzuki, Z. Wang, K. K. Parhi, "A K=3, 2Mbps Low Power Turbo Decoder for 3rd Generation W-CDMA Systems," IEEE CICC, 3-4, pp.39-42, 2000.
- [17] F. Viglione, G. Masera, G. Piccinini, M. R. Ruch, M. Zamboni, "A 50 Mbits/s Iterative Turbo-decoder," Automation and Test in Europe Conference and Exhibition 2000 Proceedings, pp.176-180, 2000.
- [18] Y. Wu, B. D. Woerner, "The Influence of Quantization and Fixed Point Arithmetic upon the BER Performance of Turbo Codes," IEEE Vehicular Technology Conference, Vol.2, pp.1683-1687, 1999.
- [19] J. Hsu, C-L. Wang, "On Finite-Precision Implementation of A Decoder for Turbo Codes," IEEE, 1999 ISCAS Proceedings of the IEEE International Symposium on Circuits and Systems, Vol.4, pp.423-426, 1999.
- [20] N. Guo, F. Khaleghi, A. Gutierrez, J. Li, M-H. Fong, "Transmission of High Speed Data in CDMA 2000," IEEE, Wireless Communications and Networking Conference, Vol.3, pp.1442-1445, 1999.
- [21] CDMA 2000 ITU-R RTT Candidate Submission, ver.0.18, July, 1998.

이 성 규

e-mail : vichui@hanmail.net

1999년 홍익대학교 전자전기공학부(공학학사)

2001년 홍익대학교 대학원 전자공학과(공학석사)

2001년~2003년 비전텔레콤

2003년~현재 삼성전자

관심분야 : 통신, 단말기 구조 설계

계 영 철

e-mail : yckay@wow.hongik.ac.kr

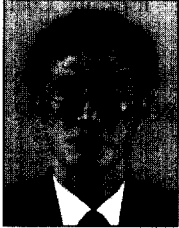
1980년 서울대학교 전자공학과 학사

1982년 한국과학기술원 전기 및 전자공학과 석사

1991년 Univ. of Southern California, Electrical Eng. Ph.D

1991년~현재 홍익대학교 전자전기공학부 부교수

관심분야 : 통신, 신호처리, 음성 및 영상인식, 로봇 비전



송 낙 운

e-mail : snukeh@wow.hongik.ac.kr

1975년 서울대학교 전자공학과 졸업(학사)

1986년 Univ. Texas Austin(Ph.D)

1986년~1989년 금성반도체 근무

1989년~현재 홍익대학교 전자전기공학과
교수

관심분야 : VLSI시스템 자동화 설계