2001년 한국정보처리학회 춘계 학술발표논문집 제8권 제1호

효율적인 자유거리를 갖는 인터리빙 아키텍처 설계

이성우*, 백승재*, 정근열*, 박진수*
*청주대학교 전자공학과
e-mail:swlee77@chongju.ac.kr

The Architecture Design of Interleaving with Effectual Free Distance

Sung-Woo Lee*, Seung-Jae Baek*
Keun-Yeol Jeong*, Jin-Soo Park*
*Dept of Electronic Engineering, Chong-Ju University

요약

인터리빙은 부호화된 메시지를 전송채널을 통하여 전송하기 전에 이루어지는 시간 다이버시티 기능으로 전송채널에서 일어나는 전송메시지에 대한 연집 오류를 시간적으로 확산시켜 산란오류로 분포시키는 기능을 수행한다. 따라서 복호기에서는 산란오류에 대하여 오류정정을 하게 되어 전송 데이터의 신호품질을 향상시킨다. 본 논문에서는 부·복호기에서 인터리빙·디인터리빙을 수행하는 블록, 대각, 랜덤 인터리버 설계방법을 제시하고 블록, 제안된 블록, 랜덤 인터리버·디인터리버를 VHDL언어로 설계 및 검증한다.

1. 서론

인터리빙은 페이딩 채널에서 발생하는 연집 오류에 대비하기 위하여 사용하는 시간다이버시티의 형태로 간단하고 효율적인 방법이다. 다이버시티의 기법[1,2]으로는 크게 주파수 다이버시티, 공간 다이버시티, 시간 다이버시티 등 3가지 방법으로 분류한다.

시간 다이버시티는 일정한 간격으로 동일 정보를 여러 번 송신하는 기법과 인터리빙 기법을 이용하여 암호화하는 방법이 있다. 인터리빙의 기본적인 원리는 부호어를 분산시켜 비트와 비트를 서로 독립적인 페이딩이 되도록 하는 것이다. 이 경우 연집 오류는 다수의 부호어에 속하는 다수의 비트군에 영향을 미친다. 부호화된 메시지를 전송하기 전 인터리빙을 하고, 메시지를 수신한 후에 역 인터리빙을 행함으로써 채널에 연집오류가 발생할 경우 이것을 시간적으로 확산시킴으로써 연집오류를 랜덤 오류의 형태로 바꾸어 오류정정이 가능하다. 그러므로 오류는 메시지 전체에 영향을 주게 되지만 영향을 받지 않은 비트로부터 복원할 수 있으며, 메모리를 사용하여 구현할 수 있다. 따라서 이와 같은 인터리버는 통신 시스템에서 아주 중요하고 유용하게 널리 쓰이고 있으며 특히 무선 페이딩 채널하에서 디지털 신호를 전송할 때 품질향상을 가져올 수 있다.

본 논문은 현재 많이 쓰이고 있는 블록, Mother, 랜덤 인터리버 설계방법을 제시하고 VHDL통해 각각의 인터리버·디인터리버를 설계하고 설계된 인터리버를 시뮬레이션 및 검증을 한다.

2. 인터리버의 종류

2-1. 블록 인터리버 (Block Interleaver)

입력 비트의 계열을 N×M 배열이라고 한다면, 정보 비트의 계열을 블록 인터리버 내에 행으로 데이터를 쓰고(표1) 배열이 완전히 채워지면 심볼은 한번에 한 열씩 변조기로 보내지고(표2) 채널을 통하여 전송된다.[3] 수신된 후에 그것의 역동작인 디인터리버(De-interleaver)를 실행하는데, 수신기는 복조기로부터 심볼을 수신하여 디인터리버를 하고

* 청주대학교 이공대학 전.정.반공학부
※ 본 연구는 과학기술부·한국과학재단 지정 청주대학교 정보통신연구센터의 지원에 의한 것입니다.

복호기로 전송된다. 심볼은 열로 디인터리버 배열안으로 메모리로 들어가고 행으로 전송된다.

1s	6	11	16	21
2	7	12	17	22
3	8	13	18	23
4	9	14	19	24
5	10	15	20	25

표 1. 인터리버 내에 쓸 때

1s	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

표 2. 인터리버를 읽을 때

블록 인터리버는 인터리빙의 종류 중 가장 구성하기 쉽고 간단한 방식이며 일반적으로 이러한 블록 인터리버는 간단하기는 하지만 부호화된 정보에서 발생하는 에러의 형태가 행렬의 크기를 넘어서게 되면 에러를 랜덤한 형태로 변환시킬 수 없게 되는 단점이 있다.

2-2 Mother 인터리버

Mother 인터리버는 3GPP에서 채택된 터보코드에 사용되는 인터리버로서 한 프레임 내에서 상관관계가 있는 입력정보를 효과적으로 상관 관계가 없는 정보로 변환해주기 때문 해밍무게 분포특성이 다른 인터리버에 비해 우수하다[4]. Mother 인터리버의 구성은 첫 번째, 행 열의 개수를 결정하고 입력 비트열의 개수를 결정한다. 두 번째는 행 내부 단위로 데이터를 섞으며, 세 번째는 행간단위로 데이터를 섞는다.

2-2. 랜덤 인터리버 (Random Interleaver)

입력된 정보 비트를 랜덤하게 재배열하여, 출력으로 내 보내기 위한 정보를 규정된 난수 체계를 이용하여 난수를 발생시킴으로써 랜덤하게 출력한다. 하지만 랜덤 인터리버는 실시간 형태의 인터리버를 구성하기는 어려우며, 단지 랜덤 수열 발생기에 의해 발생하는 수열의 저장을 통해 랜덤 인터리버를 구성하게 된다.

3. 인터리버 설계

3-1. 블록 인터리버 (Block Interleaver) 설계

본 논문은 N×M =256의 메모리크기를 갖는 블록 인터리버를 설계하기 위해 N=8, M=32, 시프트레지스터, MUX를 사용하여 그림1과 같이 설계하였다. 입력으로 들어오는 데이터 시퀀스를 MUX에서 받아서 각각의 노드로 보내어지고 각각의 노드에서는 입력되는 데이터를 분산시키기 위하여 시프트레지스터를 이용하였다. 1번째 노드는 1개의 8비트 시프트레지스터로 구성되어있고 2번째 노드는 2개의 시프트레지스터로 구성되어 있으며 이러한 방법으로 각각의 노드는 1개씩 시프트레지스터가 증가되어 구성된다.[3]

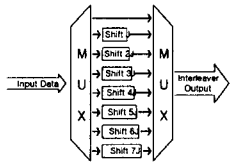

그림1. 인터리버블록도

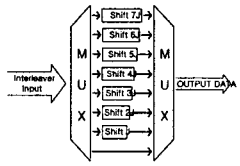

그림2. 디인터리버 블록도

그림 2는 인터리빙된 데이터 시퀀스를 디인터리빙하기위한 디인터리버의 블록도이다. 입력으로 들어오는 인터리버 시퀀스를 MUX에서 받아서 각각의 노드로 보내어지고 각 노드에서는 인터리버에서 분산된 데이터를 재구성하기 위해서 8비트 시프트레지스터 개수가 역으로 구성된다. 마지막으로 최종단에 있는 MUX를 통해 데이터가 디인터리빙 된다.

3-2. Mother인터리버 설계

블록인터리버는 선형적인 법칙에 따라 임의의 주소로부터 다른 주소로 하나의 비트를 보내는 것이다. 이러한 방식의 주요 장점은 단순한 연산에 의해 조건을 부여할 수 있기 때문에 Lookup-table을 필요로 하지 않는다는 것이다. 하지만 인터리버의 크기가 커질수록 성능은 사용자가 원하는 수준까지 이르지 못하는 단점을 갖는다. 따라서 이와 같은 문제점을 해결하기 위해 기존 블록인터리버의 출력 데이터를 다시 한번 변환하는 방법을 사용하였다.

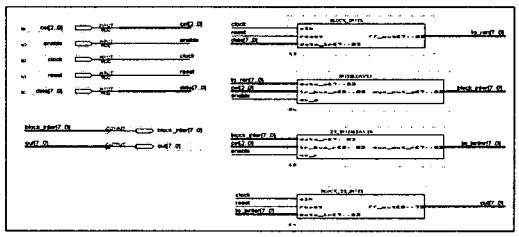

그림 3. Mother 인터리버 회로도

그림 3는 기존의 블록인터리버의 선형적인 데이터 패턴을 비 선형적인 데이터 패턴으로 바꾸기 위

해 제안된 Mother인터리버[4]의 회로도이다. 첫 번째와 마지막 블록은 행으로 데이터를 변환시키는 블록인터리버이고, 추가된 두 번째와 세 번째 블록은 데이터를 열로 변환시키고 복원시키는 블록이다. 추가된 블록의 CELL[2..0]에 임의의 값을 선택하면 선택된 값에 의해 각각 다른 패턴의 메모리 구조로 된 블록으로부터 블록인터리버의 선형적인 데이터를 비선형성적인 데이터로 변환하여 출력할 수 있다.

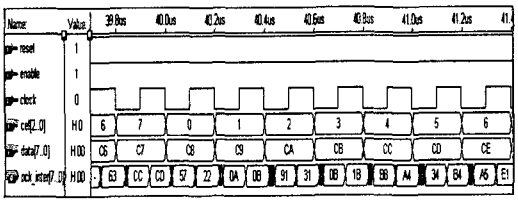

그림 4. Mother 인터리버 시뮬레이션 결과

그림 4는 순차적으로 입력된 원(Source) 데이터가 블록인터리버로 인터리빙하고 그 다음 다시 한번 열로 데이터를 분산시키는 데이터로 출력되는 VHDL 시뮬레이션 결과이다.

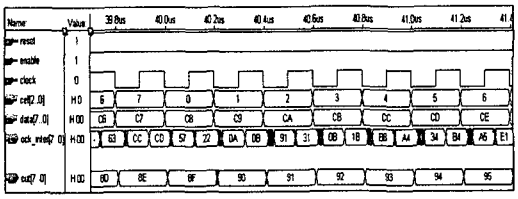

그림 5. Mother 디인터리버 시뮬레이션결과

그림 5는 Mother 블록인터의 반대되는 동작을 수행하는 디인터리버의 출력결과이다. 디인터리버에서는 Mother 인터리버의 데이터를 다시 열로 복원한 다음 블록디인터리버의 입력으로 받아 원(Source) 입력 데이터로 복원하는 기능을 수행한다.

3-3. 랜덤 인터리버(Random Interleaver) 설계

랜덤 인터리버는 부호기의 성능을 연구하는 데 있어서 기준이 되는 인터리버이다. 실시간 형태의 랜덤 인터리버는 구현 불가능하며, 단지 랜덤 수열 발생기에 의해 발생하는 수열의 저장을 통해 랜덤 인터리버[5][6]를 구성하게 된다. 인터리빙과 디인터리빙이 동일한 개체에 의해 수행되어지도록 일단 인터리빙을 위한 배열이 얻어지면 ROM에 저장하여 Lookup-table을 이용하는 방식으로 랜덤 인터리버를 구현한다.

그림 6은 본 논문에서 설계된 랜덤 인터리버의 블록도 이다. 랜덤 발생기의 입력은 시간정보의 초 단위에서 정보를 입력으로 받아 랜덤배열을 얻어 ROM에 저장을 한다. 입력으로 들어오는 순간 시간정보를 받아서 ROM에 있는 랜덤 Lookup-table에 맞는 패턴으로 정보를 랜덤 하게 분산시킨다.

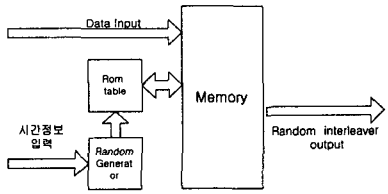

그림 6. 랜덤 인터리버 블록도

그림 7은 랜덤 인터리빙된 데이터 시퀀스를 디인터리빙하기 위한 랜덤 디인터리버의 블록도이다. 입력으로 들어오는 랜덤 인터리버 시퀀스를 랜덤 인터리버의 같은 시간정보에서 서로 반대되는 랜덤 수열이 저장된 Lookup-table을 통해 랜덤 인터리버에서 분산된 데이터가 재구성된다.

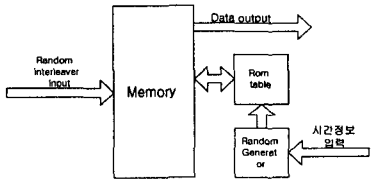

그림 7. 랜덤 디인터리버 블록도

그림 8은 각 블록을 VHDL 언어로 구현한 회로도이다. 인터리버 부분은 3비트 SEL 단자와 8 비트 데이터 입력단자로 구성되어져있다. sel의 입력은 $000_{(2)}$~$111_{(2)}$까지 변화하게 되고, $000_{(2)}$~$111_{(2)}$까지 해당되는 랜덤 패턴이 저장된 Lookup-table에 의해 랜덤 인터리빙 된다. 디인터리버 부분은 인터리버와 반대되는 랜덤한 패턴의 데이터를 Lookup-table에 의해서 원(Source) 데이터 시퀀스로 환원하는 랜덤배열로 구성되어져있다.

따라서 랜덤 패턴이 저장된 Lookup-table의 내용에 의해 서 다양한 결과를 얻을 수 있다. 제어 블록은 랜덤 배열을 발생시키는 부분이며 인터리버와

디인터리의 랜덤패턴을 발생시키는 기능을 수행한다.

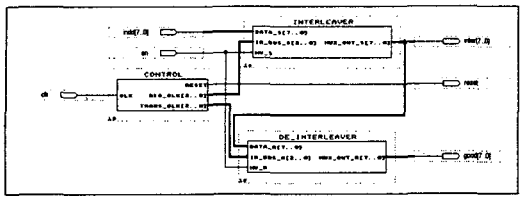

그림 8. 랜덤인터리버 회로도

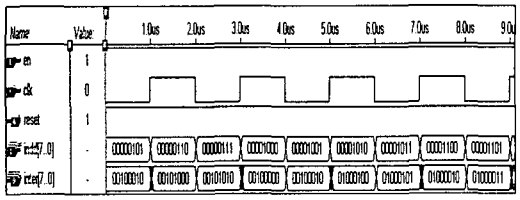

그림 9. 랜덤 인터리버 시뮬레이션 결과

그림 9는 랜덤 인터리버의 출력결과를 보여주고 있다. 그 결과는 랜덤배열에 의해 생성된 3Bit sel의 입력에 따라 랜덤한 패턴으로 데이터가 분산되었음을 보여준다.

그림 9에서 각각 시점이 6.0μs, 7.0μs때 입력되는 indd(7..0) 데이터가 00001011 $_{(2)}$, 00001100 $_{(2)}$경우 랜덤 인터리버의 출력은Lookup-table에 저장된 랜덤패턴인 01000101 $_{(2)}$, 01000010 $_{(2)}$ 랜덤인터리빙 되었음을 알 수 있다.

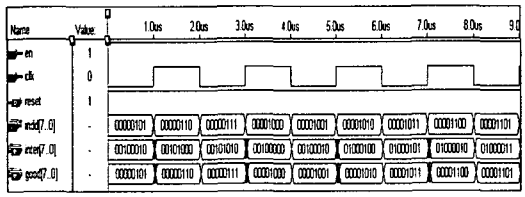

그림 10. 랜덤디인터리버 시뮬레이션 결과

그림 10은 랜덤 인터리빙된 데이터 시퀀스를 원천 데이터 시퀀스로 환원된 디인터리버의 출력결과를 보여주고 있으며, 그 결과는 랜덤한 패턴으로 분산된 데이터를 랜덤 인터리버의 시간정보와 일치시켜 랜덤디인터리버 Lookup-table의 저장 패턴으로 데이터를 복구한다. 그림 8에서 각각 시점이 6.0μs, 7.0μs때 입력되는 indd(7..0) 데이터가 00001011 $_{(2)}$ 00001100 $_{(2)}$일 때 랜덤 디인터리버의 출력은 Lookup-table에 저장된 랜덤패턴인 01000101 $_{(2)}$, 01000010 $_{(2)}$로 랜덤 디인터리빙 되었음을 알 수 있다.

4. 결 론

인터리빙은 연집 오류를 시간적으로 확산시켜 산란오류로 분포시키는 기능을 수행한다. 따라서 복호기에서는 산란오류에 대하여 오류정정을 하게 되어 전송 데이터의 신호품질을 향상시킨다.

본 논문에서는 부·복호기에서 인터리빙·디인터리빙을 수행하는 블록, Mother, 랜덤 인터리버 설계 방법을 제시하고 블록, Mother, 랜덤 인터리버·디인터리버를 VHDL언어로 설계 및 검증하였다. 향후 IMT-2000시스템인 멀티미디어 통신에 적용하여 우수한 통신품질을 제공할 수 있을 것이다.

참고문헌

[1] J.K.Parsons, M.Henze,P.A.Ratliff and M.J.Withers "Diversity techniques for mobile radio reception", Radio&Electron. Eng., 45, pp.357-367 July 1975.

[2] F.Adachi, T.Hattri, K.Hirade and T.Kamata "A periodic switching diversity technique for a digital FM land mobile radio", IEEE Trans. Veh. Tech., vt-27,4, pp.211-219 Nov.1978.

[3] Chris Heegard and Stephen B. Wicker, " Tutbo Coding", KAP(Kluwer Academic Publishers) 1999

[4] 이문호 http://rsiwin.com.ne.kr/docu/imt2000/39.htm

[5] 이문호, 實用디지털通信 -기초와 응용-, 도서출판 영일 1999.

[6] 이문호, 실용정보이론 , 복두출판사 1998.