

무선 랜의 인터리버 메모리 구조 설계에 대한 연구

*길민수, 김태기, 정차근
호서대학교 전기정보통신공학부

e-mail : *ackil@nate.com, taeghi@naver.com, cheong@office.hoseo.ac.kr*

Study of Interleaver Memory Architecture Design on Wireless LAN

*Min-Su Kil, Tae-Ghi Kim, Cha-Keun Cheong
Information Control Engineering
Hoseo University

Abstract

본 논문에서는 IEEE 802.11a 무선 랜에서 burst error에 대한 에러 정정 코드로 사용되는 블록 인터리버의 설계방법을 제안한다.

블록 인터리버 메모리는 읽기·쓰기의 주소가 다르기 때문에 주소생성을 하기 위한 회로가 복잡해진다. 본 논문에서 제안하는 방법은 블록 인터리버의 설계에서 사용되는 16x18 크기의 메모리를 세분화하여 데이터를 읽어 들일 때 쓰이는 룬이나 복잡한 로직을 제거하거나 메모리 선택기를 추가하여 보다 간소화된 주소 생성 모듈을 설계하여 로직의 효율을 높인다.

I. 서론

무선LAN은 기존 유선LAN에 비해서 데이터 전송률은 떨어지지만, 이동성 및 휴대성, 간편성 등의 이점을 가질 수 있기 때문에 응용분야가 확장되고 있다. 사무실 구조변경시의 수고를 대폭 줄일 수 있는 등, 편리성이 높은 무선LAN, 도입을 검토하는 기업도 증가하고 있다. 지금까지 주류였던 IEEE802.11b 규격에 이어, 1999년 9월 6~54Mbps의 전송속도를 제공하는 OFDM 방식의 IEEE802.11a 무선LAN 표준안으로 확정되었다. IEEE802.11a는 무선주파수로 미국, 유럽, 일본 등에서

광대역 무선 전송을 위하여 허가 없이 사용할 수 있는 대역으로 확정된 5GHz대를 사용하는 무선LAN 규격이다. 동시에 표준화하였던 2.4GHz대를 사용하는 IEEE802.11b의 최대 11Mbps에 비해 약 5배의 전송속도를 얻을 수 있다.[2][3] 고속 무선 LAN의 표준안으로 IEEE 802.11a가 확정되고 이 표준안에 따르는 고속 무선 LAN을 사용하여 공중망과 연동하여 광대역 무선 서비스를 제공하는 계획이 발표됨에 따라 만든 국제 표준화 기구와 국내·내외 기업 및 연구소에서 이에 대한 연구 및 개발이 활발히 수행되었다.[2]

IEEE 802.11a에서 규정한 블록인터리버는 두 단계의 순환 치환을 사용하여 데이터의 순서를 뒤섞어 놓는다. 인터리빙은 메모리가 필수적이며 복잡한 구조의 입출력 주소 생성 및 제어 모듈이 포함된다. 본 논문에서는 기존의 메모리 구조를 세분화하여 간소화된 주소 생성 및 제어 구조를 제안한다. 본 논문은 II장에서 IEEE 802.11a의 인터리버와 permutation을 기술하고 III장에서는 제안되는 메모리 구조와 permutation을 만족하는 입·출력 주소의 생성 및 제어 방법의 특징을 알아본다. 그리고 마지막으로 IV장에서는 결론으로 구성되어 있다.

II. IEEE 802.11a Interleaver

IEEE 802.11a 무선 랜에서 사용되는 인터리버의 규

격은 N_{CBPS} 와 N_{BPSC} 는 Data Rate에 의존하는 파라미터로서 N_{CBPS} 와 N_{BPSC} 의 서로 다른 4쌍에 의해서 총 4가지로 분류된다. 구체적으로 이러한 인터리버는 첫 번째 서로 인접한 코딩된 데이터들을 인접하지 않은 부반송파에 매핑하고, 두 번째 심볼 매핑 정상도에서의 비트 위치를 바꾸는 두 단계의 순열 치환을 이용하여 데이터의 순서를 바꾸어 놓는다. 다음 표1은 IEEE 802.11a에서 규정한 Rate에 의존하는 파라미터를 정리한 것이다.

전송 속도 (Mbps)	변 조	Coding rate(R)	N_{BPSC}	N_{CBPS}	N_{DBPS}
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	64-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

- N_{BPSC} : Coded bits per subcarrier
 - N_{CBPS} : Coded bits per OFDM symbol
 - N_{DBPS} : Data bits per OFDM symbol

표 3. Rate dependent parameters

802.11a에서 사용되는 인터리버는 변조방식에 따라 4가지 종류가 있으며 2단계의 permutation 과정을 거친다. 첫 단계는 인접한 비트가 인접하지 않는 부반송파에 실리도록 하며, 두 번째 단계는 데이터 심볼 매핑의 constellation상에서 비트 위치를 바꾸어 주는 과정이다.

다음 식 (1), (2)는 두 단계 permutation 과정을 나타낸다.[1]

■ First permutation

$$i = (N_{CBPS}/16)(k \bmod 16) + \text{floor}(k/16) \quad (1)$$

$$k = 0, 1, \dots, N_{CBPS} - 1$$

■ Second permutation

$$j = s \times \text{floor}(1/s) + (i + N_{CBPS} - \text{floor}(16 \times i / N_{CBPS})) \bmod s \quad (2)$$

$$i = 0, 1, \dots, N_{CBPS} - 1 \quad s = \max(N_{BPSC}/2, 1)$$

2단계의 permutation 과정에서 변조방식이 n-PSK 변조방식의 경우는 1단계의 permutation의 결과에 아무런 영향을 미치지 않게 되며 16-QAM과 64-QAM 변조방식의 경우는 인터리빙의 수행 효과가 나타난다. 그림 1은 16-QAM에서의 인터리빙 과정을 나타낸 것이다.

이와 같은 인터리빙의 과정을 수행하려면 메모리가 반드시 요구되며, 메모리에 읽기 주소와 쓰기 주소를

생성 모듈과 관련된 제어 모듈 설계해야만 한다. IEEE 802.11a 인터리버는 보통 쓰기 주소 생성이 간단하며 1,2단계 permutation 수행결과가 동일한 n-PSK 변조방식은 읽기 주소 생성이 간단하다. 하지만 n-QAM 변조방식은 읽기 주소 생성과정이 복잡하게 된다.

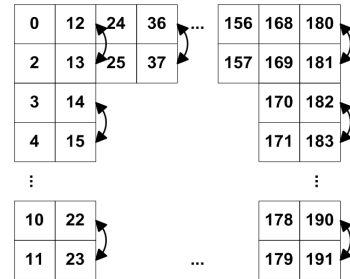


그림 1. 16-QAM Interleaving

III. 메모리 구조와 주소생성 및 제어

1. 메모리 구조

그림 1에서 16-QAM 변조방식에서의 2단계 permutation 과정은 1단계 permutation의 일정 위치의 값을 교환함을 알 수 있다. n-QAM 변조방식에서 두 단계의 permutation 결과의 차이는 식 (2)에서 식 (1)을 뺀 결과와 같다. 또한 이 차이는 각각의 변조에서 일정한 패턴을 가지게 된다. 그림 2는 각각의 변조에서 식 (2)과 식 (1)의 차이의 패턴을 나타낸 것이다.

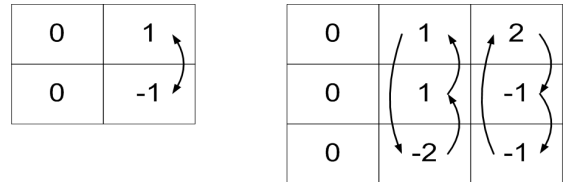


그림 2. 16-QAM(좌)과 64-QAM(우)의 차이 패턴

위의 두 단계 인터리빙의 패턴은 16xK(K=3, 6, 12, 18) 크기의 메모리에서는 각각의 변조의 인터리빙에 맞는 읽기 주소가 저장된 메모리를 구성하여 출력하거나 또는 최초 인터리빙의 메모리에 정보를 뒤섞어 순차적으로 출력을 하는 설계구조이다. 이는 설계가 복잡하며 칩 구현에서 많은 로직과 메모리를 요구하게 된다. 이를 보완하기 위해 메모리 구조를 16x1 크기의 메모리로 세분화하여 18개조된 병렬 메모리 구조를 구성한다. 그림 3은 18개조의 병렬 16x1 메모리를 나타낸다.

기존의 메모리를 세분화하는 기법은 메모리 선택기가 추가되지만 단순화 구조를 갖으며, 읽기·쓰기 주소 생성이 간소화된다.

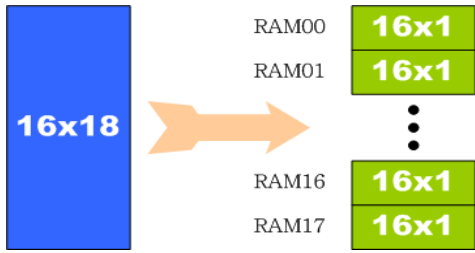


그림 3. 18개조의 병렬 16x1 메모리 구조

2. 쓰기모드의 메모리 선택기와 메모리 주소

인터리빙의 위한 데이터의 메모리 기록 순서는 그림 3의 RAM₀₀의 0번지부터 아래(메모리 선택)-우(메모리 쓰기 주소) 지그재그 스캔으로 된다. n-PSK 변조방식에서 지그재그 쓰기에 변화가 없지만 n-QAM 변조방식에서는 2단계 permutation으로 인해 그림 2의 패턴과 같은 변화를 갖는다. 그러므로 패턴 변화에 맞는 메모리 선택기 설계가 필요하다.

n-QAM 변조 방식에 그림 2의 패턴의 을 세분화된 메모리 구조 전체에 적용하고 각 조의 메모리의 주소를 s로 나눈 나머지의 값($m=Address \text{ mod } s$)을 이용하면 16-QAM 변조방식의 경우 $m=0$ 일 때는 변화가 없으나 $m=1$ 일 때는 교환패턴이 발생하며 64-QAM 변조방식의 경우 $m=0$ 일 때는 변화가 없으나 $m=1, 2$ 일 때는 교환 패턴이 발생함을 알 수가 있다. 또한 패턴 과정을 확장하면 16-QAM에서 2개조의 메모리 선택이, 64-QAM에서 3개조의 메모리 선택이 로테이션하는 것을 알 수 있다.

2.1. 16-QAM 변조방식에서 메모리 선택기

16-QAM 변조방식에서는 18개조의 메모리 중 12개조를 사용하며 크기는 192 (16x1x12)이다.

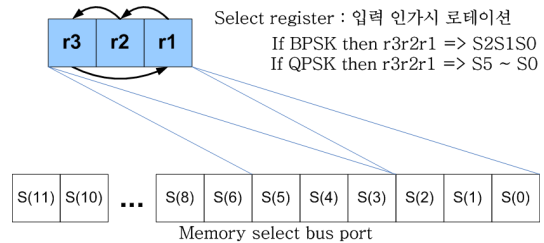
우선 메모리의 선택은 Active High로 선택 디플트를 '0'로 설계한다. 12개조 메모리 중 2개조를 한 쌍으로 하여 $m=0$ 인 경우 LSB가 하위 메모리 선택인 선택 값이 저장된 레지스터를 '01'으로 초기화 하고 데이터 입력 시마다 로테이션 쉬프트를 한다. 초기 상태에는 선택 레지스터의 값을 RAM_n과 RAM_{n+1}에 인가하고 선택 레지스터의 상태가 초기 상태와 동일하게 되면 선택 레지스터의 값을 RAM_{n+2}와 RAM_{n+3}에 인가하며 일련의 과정을 선택 레지스터의 값이 RAM₁₀과 RAM₁₁에 인가 될 때까지 반복한다. $m=1$ 인 경우 LSB가 하위 메모리 선택인 선택 값이 저장된 레지스터를 '10'으로 초기화 하고 $m=0$ 인 경우일 때의 과정을 반복한다.

2.2. 64-QAM 변조방식에서 메모리 선택기

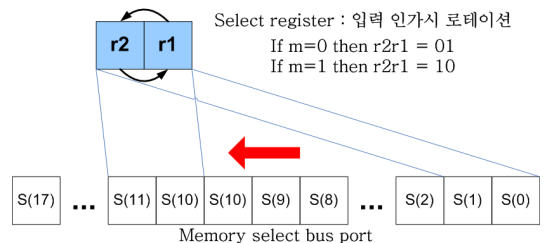
64-QAM 변조방식에서는 메모리 선택기의 설계는 16-QAM 변조방식과 유사하며 메모리 선택 값이 저장

된 선택 레지스터의 크기가 2 bits에서 3 bits로 변경되며 메모리는 18개조의 메모리 모두를 사용하며 크기는 288 (16x1x18)이다.

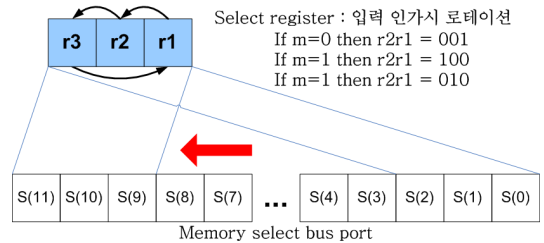
18개조 메모리 중 3개조를 한 쌍으로 하여 $m=0$ 인 경우 LSB가 하위 메모리 선택인 선택 값이 저장된 레지스터를 '001'으로 초기화 하고 데이터 입력 시마다 로테이션 쉬프트를 한다. 초기 상태에는 선택 레지스터의 값을 RAM_n과 RAM_{n+1}에 인가하고 선택 레지스터의 상태가 초기 상태와 동일하게 되면 선택 레지스터의 값을 RAM_{n+2}와 RAM_{n+3}에 인가하며 일련의 과정을 선택 레지스터의 값이 RAM₁₆과 RAM₁₇에 인가 될 때까지 반복한다. $m=1$ 인 경우 LSB가 하위 메모리 선택인 선택 값이 저장된 레지스터를 '100'으로 초기화 하고, $m=2$ 인 경우 LSB가 하위 메모리 선택인 선택 값이 저장된 레지스터를 '010'으로 초기화 하여 $m=0$ 인 경우일 때의 과정을 반복한다.



(a) n-PSK modulation mode



(b) 16-QAM modulation mode



(c) 64-QAM modulation mode

그림 4. 모드별 메모리 선택기 수행 과정

2.3. n-PSK 변조방식에서 메모리 선택기

n-PSK 변조방식에서는 2단계 permutation으로 인해 인터리빙이 변화가 없어 각각 메모리 주소 크기의 선

택 레지스터를 두어 로테이션을 하면 간단하지만 6 bits의 쓸모없는 레지스터가 생성이 된다. 그러므로 n-PSK 변조방식에서 선택 레지스터의 초기 상태는 64-QAM 변조방식의 인터리빙에서 $m=0$ 인 경우의 선택 레지스터 '001'을 사용한다. BPSK 변조방식에서는 선택 레지스터를 로테이션 쉬프팅하면서 메모리 선택 버스에 인가하고 QPSK 변조방식에서는 n-QAM 변조방식에서와 같이 로테이션 쉬프팅 하며 선택 레지스터 값을 RAM₀, RAM₁, RAM₂에 인가하고 선택 레지스터 값이 초기 상태가 되면 RAM₃, RAM₄, RAM₅에 인가한다. 그림 4는 각 모드의 메모리 선택기의 수행 과정을 나타낸 것이다.

2.4. 메모리 주소 생성기

인터리버 메모리에 데이터 기록 순서가 아래-우 지그재그이므로 각 모드 별로 사용하는 n번째 메모리 각조에 데이터 기록이 완료되면 n+1번째 메모리 각조에 데이터를 기록하여 메모리 번지가 15번지가 되면 데이터 기록을 완료하게 된다.

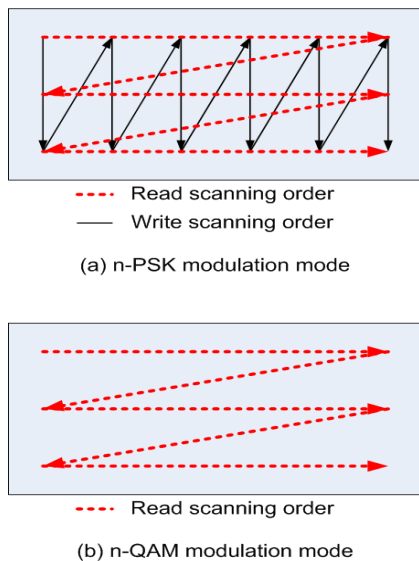


그림 5. 메모리 읽기·쓰기 스캔 순서

3. 읽기모드의 메모리 선택기와 메모리 주소

메모리를 세분화한 구조의 인터리버에서는 쓰기모드에서 n-PSK 변조방식과 n-QAM에 맞는 2단계의 permutation을 수행·결과에 만족하는 위치에 데이터가 기록된다. 그러므로 메모리에 기록된 데이터는 다른 조작 없이 우(메모리 주소)-아래(메모리 선택) 지그재그로 출력 스캔한다. 그림 5은 n-PSK 변조방식에서의 인터리버 메모리 입력스캔 순서와 출력스캔 순서와 n-QAM 변조방식에서의 인터리버 메모리 출력스캔 순서이고 그림 6은 제안된 방법으로 인터리버를 설계하

였을 때의 시뮬레이션 결과로서 잘 동작함을 알 수 있다.

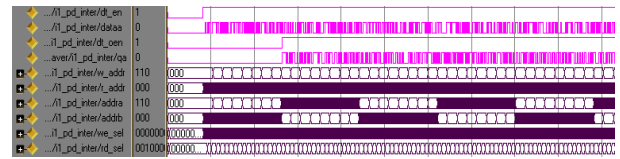


그림 6 인터리버의 시뮬레이션 결과

III. 결론

5GHz대를 사용하며 최대 54Mbps의 속도로 통신하는 무선 LAN 규격인 IEEE 802.11a 무선 랜을 위한 인터리버의 효율적인 메모리 구조와 그에 따른 설계에 대해서 논하였다. 발생 오류 중 burst error에 대한 오류 정정 코드로서 인터리버와 일반 오류에 대한 오류 정정 코드로서 컨볼루션 인코더는 오류 정정 능력 향상에 기여를 한다.

블록 인터리버는 필요에 의하여 메모리를 포함하는 모듈로서 IEEE 802.11a에 사용되는 인터리버는 메모리에 입력된 데이터를 2단계의 permutation 과정을 수행하여 최종 인터리빙된 데이터를 출력하게 된다. 하지만 메모리 읽기·쓰기 주소 생성이 복잡하거나 주소를 가진 다른 메모리가 필요하다는 단점이 있다. 이는 ASIC chip 제작시에 로직의 효율적인 사용을 방해하는 요인인 된다.

이를 해결하기 위한 방법으로 세분화된 16x1 크기 메모리를 사용하는 인터리버의 설계를 제시하였다. 18개조의 세분화된 메모리를 사용하였을 때 메모리 선택기의 설계가 필요하지만 선택기의 크기가 크지 않으며 설계가 용이하였다. 그리고 읽기·쓰기 주소의 생성이 순차적으로 증가하는 구조이기에 설계가 편하고 로직이 간소됨을 알 수 있다. 동일 사양의 FPGA chip 합성에서 인터리버 로직 기준으로 기존의 읽기·쓰기 주소 생성보다 약 3~5%의, 을 주소 저장 메모리 사용보다 3~10%의 로직이 확보되었다.

참고문헌

[1] IEEE Std., "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band, 1999

[2] 김재석, 조용수, 조중휘 공저, "이동통신용 모뎀의 VLSI 설계 -CDMA/OFDM/MC-CDMA 모뎀", 대영사, 2001.

[3] 정지은, 고속 무선LAN, TTA저널 제83호, 2002.