

IEEE 802.11a Wireless Lan CODEC 칩 설계

변남현, 조영규, 정차근
 호서대학교 전기정보통신공학부 정보제어전공

IEEE 802.11a Wireless Lan CODEC Chip Design

Nam-Hyun Byun, Young-Kyu Cho, Cha-Keon Cheong^o
 The School of Electrical Engineering, Hoseo University

요 약

본 논문에서는 IEEE 802.11a 무선 LAN 용 CODEC 회로를 설계하고, VHDL 코딩과 FPGA에 의한 회로설계 검증에 관해 기술한다. IEEE 802.11a WLAN CODEC의 구조는 크게 데이터 보호를 위한 스크램블러/디스크램블러, 채널 에러에 대한 정보보호를 위한 Convolutional 부호기와 Viterbi 복호기로 구성된 채널 코덱, 그리고 연접에러를 랜덤 에러로 변화시키는 인터리버/디인터리버로 구성된다. 본 논문에서는, 이와 같은 CODEC의 각 부분을 하드웨어로 구현하기 위한 새로운 회로구성을 제안하고, 그 성능을 VHDL 코딩에 의한 시뮬레이션과 FPGA에 의한 하드웨어 검증 결과를 제시한다.

I 서 론

무선 LAN은 다양한 정보와 자원을 공유할 수 있게 하는 LAN의 장점과, 공간상의 제한을 받지 않는 자유로운 이동성을 제공하고, 신뢰성 있는 데이터 전송뿐 아니라 유연성과 설치의 용이성을 장점으로 하고 있다. [2]

최근 OFDM 기술의 적용으로 6-54Mbps까지의 다양한 데이터 전송속도를 갖는 IEEE 802.11a 무선 LAN 표준안이 확정되어 광대역 무선 데이터 초고속 전송 실현의 가능성이 구체화되고 있다. 이로 인해, 미국을 비롯한 선진 각국의 주요 칩 메이커를 중심으로 IEEE 802.11a WLAN 칩 개발이 진행되고 있으며 실용화를 위한 연구가 활발히 이루어지고 있다.

본 논문에서는 IEEE 802.11a 무선 LAN에서 규격화된 CODEC 회로의 각 블록에 관한 새로운 하드웨어 구조를 설계하고, 이를 VHDL 코딩 및 FPGA 구현으로 그 동작을 확인한다. 본 논문의 구성은 다음과 같다. 먼저, II절에서 IEEE 802.11a WLAN의 전체 구성과 본 논문에서 구현하는 CODEC의 구조에 대해서 간략히 기술한다. 다음으로 III절에서는 CODEC용 회로의 동작과 구현회로의 구조에 관해서 상세하고, 구체적인 회로구성과 VHDL 코딩 및 FPGA 검증에 따른 모의 실험 결과를 IV절에서 제시하고, V절에서 결론을 기술한다.

II. IEEE 802.11a WLAN 전체구성과 코덱

1. IEEE 802.11a 무선 LAN의 전체구성

그림 1은 802.11a WLAN 시스템의 전체구성을 블럭 다이어그램으로 간략히 나타낸 것이다.

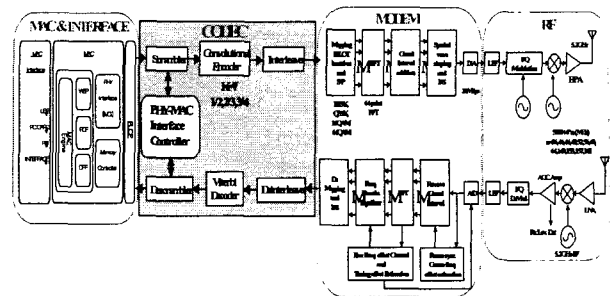


그림 1. 무선 랜의 전체 구성

PC나 주변 Network로부터 송신되는 데이터는 MAC에서 제어되어 IEEE 802.11a 모뎀의 CODEC부로 입력된다. MAC 계층은 상/하위 계층(LLC와 PLCP PHY 레이어)간의 인터페이스로서, 기본적인 데이터 액세스와 동기화, 데이터 로밍, 전력 제어, 사용자의 등록 및 관리 암호화 등을 처리한다.

PHY 계층의 CODEC부에서는 MAC과 PHY 계층간의 인터페이스를 제공하는 PLCP, 스크램블링에 의한 데이터의 보호, 채널부호화 및 인터리버/디인터리버 블록으로 구성된다.

CODEC부에서 처리된 데이터는 모뎀부에서 IFFT에 의한 변조, OFDM에 의한 다중화 처리 등을 거쳐 RF단에 의해 송신된다. 데이터 수신은 RF단으로부터 기저대역으로 천이시킨 후 채널 등화, FFT에 의한 복조 등의 과정을 거쳐 CODEC부에서 처리된다.

2. CODEC 구조

그림 2는 IEEE 802.11a CODEC부의 구조로 PLCP interface 부, 스크램블러/디스크램블러부, Convolutional 부호기 및 Viterbi 복호기부, 인터리버/디인터리버부 등으로 구성된다.

먼저, PLCP 인터페이스부는 PHY 계층에서 데이터 송수신에 적합한 포맷으로 MAC(Medium Access Controller) 계층으로부터 송수신한 데이터를 변환시키는 역할을 수행한다.[3] 다음으로 스크램블러 및 디스크램블러부에서는 데이터보호와 기본적인 security를 제공한다.

또한 채널 에러에 대한 정보보호를 위한 채널 부호화기로서 부호율 1/2인 Convolutional 부호기와 Viterbi 복호기를 사용한다. 인터리버와 디인터리버부는 다중경로 페이딩 채널에서 연속적인 에러의 발생으로 인한 연접오류를 랜덤오류의 형태로 변환시켜, 채널에러에 보다 강한 정보전송이 이루어지도록 하기 위해 사용한다.

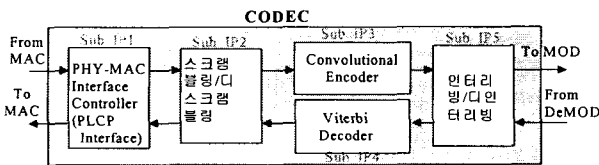


그림 2. PHY 계층의 CODEC부의 구성

III 회로설계

3.1 스크램블러 및 디스크램블러

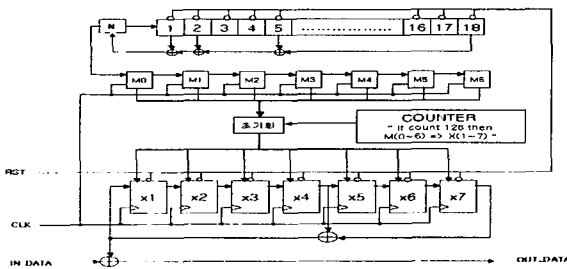


그림 3. 스크램블러 H/W 전체 구조

IEEE 802.11a에서는 스크램블러로 길이 127인 프레임 동기화된 스크램블러를 사용하며 그림 3과 같이 구성된다.

스크램블러는 전송시에 non-zero상태로 초기화되고 127 비트 단위의 반복적인 스크램블링 시퀀스를 생성하여 입력 데이터와 XOR한 뒤, 데이터가 출력된다.

그림 3에서 스크램블러를 위한 생성 다항식은 다음 식 (1)과 같이 규격화 되어있다.

$$S(x) = x^7 + x^4 + 1 \tag{1}$$

데이터 스크램블러는 쉬프트 레지스터를 이용하여 간단히 구성된다. 따라서, 127 비트의 데이터가 쉬프트 된 후 스크램블러의 레지스터 값이 다시 초기값과 같은 상태가 되므로, 새로운 초기값을 필요로 하게 된다. 이를 보완하기 위해서는 PN코드의 사용에 의한 랜덤 데이터의 발생이 필요하다. 랜덤화를 극대화 하기 위해 다음과 같은 PN코드의 생성 다항식 $N(x)$ 를 제안하고 이를 구현했다.

$$N(x) = x^{18} + x^5 + x^2 + x^1 + 1 \tag{2}$$

스크램블러로부터 암호화된 데이터를 본 논문에서는 수신단에서 다시 원래의 데이터로 복원하기 위하여 송신단에서와 같은 구조의 스크램블러를 사용하므로써 원래의 소스 데이터가 얻어진다.

3.2 인터리버 및 디인터리버

인터리빙과 디인터리빙의 과정은 RAM으로 간단히 구현할 수 있다. IEEE 802.11a의 무선모뎀에서는 다양한 전송속도를 지원하기 위한 방법으로 전송률에 따라 다른 크기의 인터리버 사이즈를 갖는 2단의 블록 인터리빙을 수행한다. 전체 인터리버의 사이즈는 하나의 OFDM 심볼에 포함되어 있는 부호화된 비트 수 N_{CBPS} 와 같으며, 변조방식에 좌우된다. 크기가 다른 4종류의 메모리(RAM)를 사용하는 것은 하드웨어가 복잡해지므로, 가장 큰 사이즈의 메모리 (18×16)를 기준으로 설계한다.

먼저 인터리빙의 첫 번째 단계는 부분송파에 실리는 비트를 최대한 서로 떨어져 위치케 하는 것으로, 다음 식을 사용해서 수행한다.

$$i = (N_{CBPS} / 16)(k \bmod 16) + \text{floor}(k / 16) \tag{3}$$

여기서 $k=0,1,2,\dots,N_{CBPS}-1$ 는 인터리빙 되기 전의 비트열의 순서이고, i 는 인터리빙 후의 비트열의 순서를 나타낸다. 다음으로 Constellation 상에서 인접한 비트의 위치를 바꾸어 주는 인터리빙으로, 다음 식을 사용해서 수행한다.

$$j = s \times \text{floor}(i/s) + (i + N_{CBPS} - \text{floor}(16 \times i / N_{CBPS})) \bmod s \tag{4}$$

여기서는 첫 번째 인터리빙 후의 비트열의 순서이고, j 는 최종적인 비트열의 순서를, $s = \max(N_{BPSC}/2, 1)$ 및 N_{BPSC} 는 하나의 부분송파에 실리는 비트 수를 나타낸다.

디인터리버는 메모리의 주소를 생성하는 과정이 인터리버와 반대이며, 나머지 구조는 인터리버와 동일하다. 메모리 액세스를 위한 주소의 생성은 다음의 두 식으로 이루어진다.

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

$$j = 0, 1, K N_{CBPS} - 1 \tag{5}$$

$$k = 16 \times i - (N_{CBPS} - 1) \times \text{floor}(16 \times i / N_{CBPS})$$

$$i = 0, 1, \dots, N_{CBPS} - 1 \quad (6)$$

그림 4는 본 논문에서 구성한 인터리버의 하드웨어를 블록도로 나타낸 것이다. 각 하나의 WR 카운터와 RE 카운터를 이용해서 메모리에 데이터가 저장될 어드레스를 발생시킨다.

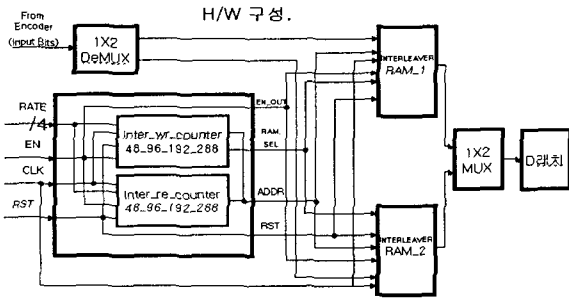


그림 4. 인터리버 H/W 구성

3.3 Convolutional 부호기[4][5]

(1) Convolutional 부호기

IEEE 802.11a의 구속장 K는 7로 정의되고, 부호율 1/2인 Convolutional 부호기를 사용하고, 부호어를 생성하기 위한 생성다항식은 다음 식(7)로 정의된다.

$$g_1 = 133_8 = 1011011_2 = 1 + x^2 + x^3 + x^5 + x^6$$

$$g_2 = 171_8 = 1111001_2 = 1 + x + x^2 + x^3 + x^6 \quad (7)$$

IEEE 802.11a의 Convolutional 부호기에서도 1/2의 기본 부호율로부터 전송율에 따라 2/3와 3/4의 높은 부호율을 얻기 위해 평처링을 기법을 사용한다. 본 논문에서는 평처링 패턴으로부터 부호율에 따른 각각의 하드웨어를 구현할 필요없이 단일화된 하드웨어 구조로써 전송율에 따른 부호율을 생성하도록 회로를 구성했다.

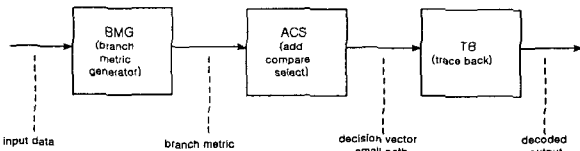


그림 5. Viterbi decoder의 전체 구조

(2) 비터비 복호기

Convolutional 부호기로부터 부호화된 심볼의 복호로 Viterbi 복호기를 사용해서 수행한다. Viterbi 복호기의 구조는 그림 5에 나타낸 것처럼 크게 BM부(branch metric), ACS (Add-Compare-Select)부, TB부(Trace Back)로 구성된다.

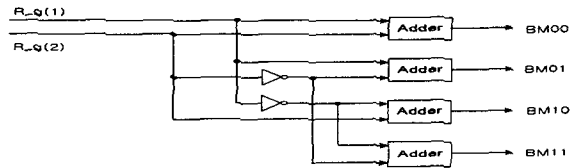


그림 6. offset binary BM 발생기

① BMG(Branch Metric Generator)

BMG 구조는 offset-binary notation을 이용하는 방법을 사용하였다. offset-binary notation 구조를 그림 6에서 나타내고, 이 구조를 이용해 BMG에 들어오는 데이터를 BM00, BM01, BM10, BM11 4개의 BM으로 만들어 이 값을 ACS에 전달한다.

② ACS(Add-Compare-Select)

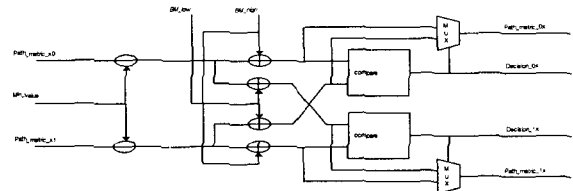


그림 7. ACS 연산기

구속장이 7일 경우의 총 상태수는 64이고, 32조의 버터플라이 구조로 구성한다. 이는 ACS 연산기의 기본 구성에서 기본원리는 상태전이 정보가 나비(butterfly) 구조를 하고 있다는 것을 이용한 것이다.

ACS 연산 모듈은 버터플라이 구조에서 BM값과 이전상태값인 PM값을 더하여, 두 개의 새로운 상태값과 메트릭값을 비교한 다음 작은 값을 선택하여 현재 상태의 상태 메트릭 값으로 결정하는 기능을 수행한다. 또한 ACS연산 모듈에서는 해당 PM과 BM을 더한 후에 비교하여 생존 경로에 대한 정보를 TB부로 전달하는 역할을 한다.

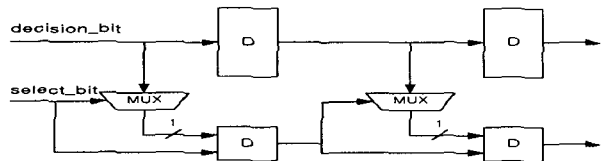


그림 8. 레지스터 교환 방식의 TB 구조

③ TB(Trace Back)

TB 블록은 ACS에서 출력된 결정비트를 원래의 데이터를 복호한다. 복호 방식으로는 역추적 방식과 레지스터 교환방식이 주로 사용되고 있으며, 고속 동작을 위해서 레지스터 교환방식으로 구현하였다.

ACS에서 연산된 64개의 결정비트(decision bit)값들은 디코딩 깊이에 해당하는 수만큼 레지스터에 저장된다. 디코딩 깊이 이전 시간에는 유효하지 않은 데이터가 출력되며 디코딩 깊이가 지난 후에 복호된 데이터가 출력된다.

그림 8은 레지스터교환 방식의 TB 구조를 나타낸 것으로, 레지스터와 MUX들로 구성되며 첫 번째 MUX의 선택비트 입력으로는 0을 인가하도록 설계하였다. 디코딩 깊이 동안은 ACS로부터 연산된 결정비트들이 FIFO(First Input First Output) 형태의 레지스터로 구성된 메모리 블록에 저장된다. 디코딩 깊이에 해당하는 수의 데이터가 입력되면 현재 입력된 결정 비트들을 시작으로 하여 이전 결정 비트들을 역추적하여 원래의 신호를 복원하게 된다.

IV 모의 실험 결과

설계한 CODEC 회로는 VHDL을 이용하여 각 블록별 및 통합 시뮬레이션을 수행하였고, 평처링 패턴에 따른 성능을 분석하였다.

그림 9는 IEEE 802.11a 코덱부 전체에 관한 시뮬레이션 결과를 나타내고 있다. 입력 데이터가 스크램블러를 거쳐 스크램블링되면 입력데이터와는 전혀 다른 형태로 변하여 다음 단인 Convolutional 부호기를 통과한다.

이때, 입력 데이터는 Convolutional 부호기를 통해 부호화가 이루어지며 인터리버를 거쳐서 인터리빙 된 데이터는 송신 단(Mode)으로 전달된다.

다시 수신 단(Demode)으로부터 수신된 데이터는 디인터리버를

거쳐 인터리빙 되기 전의 데이터를 복원한다. 인터리빙 전에는 카운터를 이용하여 0~47 까지의 숫자가 순차적으로 발생되지만 인터리빙 된 후에는 메모리의 주소가 식 (3)과 (4)에 따라 발생되는 것을 확인할 수 있다.

디인터리버를 거친 데이터는 Viterbi 복호기를 거쳐 복호화되는 것을 볼 수 있다.

수신 단에서 출력데이터를 디스크램블러를 통하여 원래의 데이터로 복원하면 처음 스크램블링되기 전의 소스 데이터 형태 그대로 나타나는 것을 확인 할 수 있다.

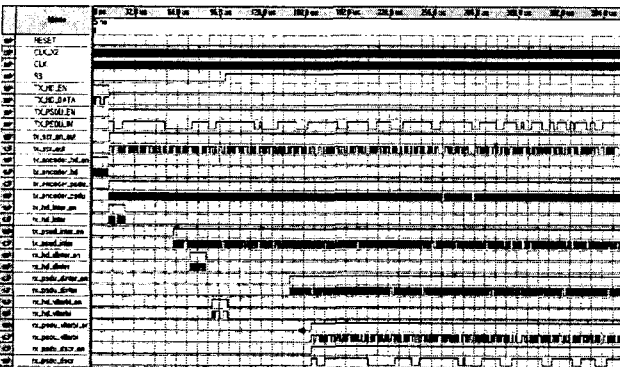


그림 9. 전송속도 6Mbps에 대한 시뮬레이션 결과

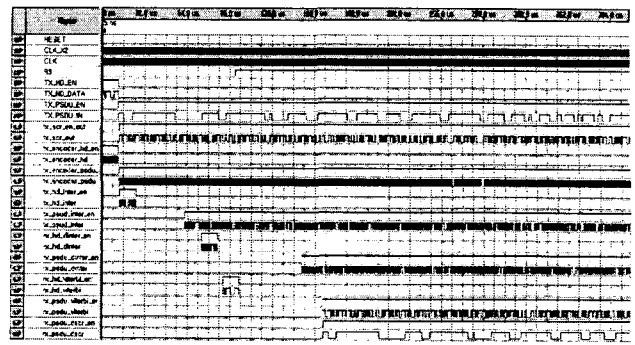


그림 10. 전송속도 24Mbps에 대한 시뮬레이션 결과

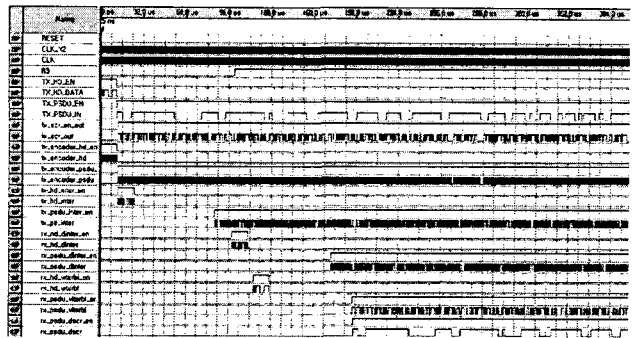


그림 11. 전송속도 54Mbps에 대한 시뮬레이션 결과

V 결론

본 논문에서는 IEEE 802.11a Wireless LAN CODEC 회로 설계를 설계하고 코덱 부 전체 및 각각의 블록에 대한 성능을 분석하였다. FPGA 테스트를 통하여 CODEC부의 블록별 하드웨어 검증을 하였고, CODEC의 각 블록을 통합한 전체 CODEC부와 모델을 하나로 통합을 하여 하드웨어 검증을 수행하여 규격에 따라 정상적인 데이터 송수신이 되는 것을 확인하였다. 또한 모델과 하나로 통합된 칩 개발을 진행하고 있다.

참고 문헌

- [1] IEEE std 802.11a Wireless LAN Medium Access control(MAC) and physical Layer(PHY) Specifications : Higher-Speed Physical Layer Extension in the 5 GHz Band, 1999
- [2] 권희훈 「다중경로 페이딩 채널에서 DS/CDMA 시스템의 성능분석」 충주대학교 산업과학논문집, 제 8호, 2000. 8. Performance Analysis of DS/CDMA System Over Multipath Fading Channel Study
- [3] Richard O. LaMaire, Arvind Krishna, and Pravin Bhagwat, "Wireless LANS and Mobile Networking : Standards and Future Directions", IEEE Commun. Mag, pp. 86-89, August 1996.
- [4] James C, Chen, and M. Jeffery: "Measured performance of 5GHz 802.11a Wireless LAN systems" Atheros Communications. Inc.