

12 Mbps 무선 LAN 비터비 디코더 설계 및 구현

최창호*, 정해원*, 이찬구*, 임명섭**

*한국전자통신연구원

**전북대학교 전자,정보공학부

Implementation of 12 Mbps Viterbi Decoder for Wireless LAN

Changho Choi, *Haewon Jung, *Changoo Lee, **Myoungseob Lim

* Electronic and Telecommunications Research Institute

**Division of Electronic & Information Engineering Chonbuk National UNIV.

chhchoi@etri.re.kr

요 약

본 논문은 IEEE 802.11a에 의해 규정되어진 데이터율 12Mbps, 부호화율 1/2, 구속장이 7인 무선LAN용 비터비 디코더를 설계하고 구현한다. 구현에 앞서 각 구속장에 따른 전달함수를 구하여 각 구속장 별 first event 에러 확률과 비트 에러 확률을 구한다. 4bit 연성판정을 위해 입력 심볼을 16단계로 양자화 하였으며 역 추적을 위한 방식으로 메모리를 사용하는 대신 새로운 알고리즘을 적용한 레지스터 교환방식을 사용함으로써 majority voting을 가능하도록 하였다. 고속의 데이터를 처리하기 위해 병렬구조를 갖는 설계를 FPGA 칩을 사용하여 구현하였고 AWGN 환경 하에서 성능검증을 하였다.

1. 서론

오류정정을 위한 비터비 복호기에서 부호화된 정보 열을 복구하는 메모리 관리 방식으로 메모리를 사용한 역추적방식과 레지스터교환 방식이 잘 알려져 있다[1],[2]. 비터비 복호기의 출력단에서 최종 복호된 정보 열을 얻기 위해 본 논문에서는 기존의 접근방식과 다른 레지스터 교환방식을 사용하는 새로운 역추적 알고리즘을 제시한다.

2. 무선 LAN용 비터비 디코더

무선 LAN 서비스의 경우 정보 데이터 열은 원하는 전송 데이터 율에 적합하도록 부호화율 1/2, 2/3, 또는 3/4의 길쌈부호화기를 사용하여 부호화 된다[3]. 그림 1은 구속장이 7인 경우의 길쌈부호기에 의해 생성된 격자도를 나타낸다.

그림 1에서 각각의 상태는 $(000000)_2$ 부터 $(111111)_2$ 까지 총 64개의 상태가 있음을 알 수 있다. 그림 2는 그림 1에 나타난 64개의 상태를 2로 나눈 32개조의 나비구조 중 한 조를 일반적으로 나타낸 것이다. 여기서 X는 현 상태의 최

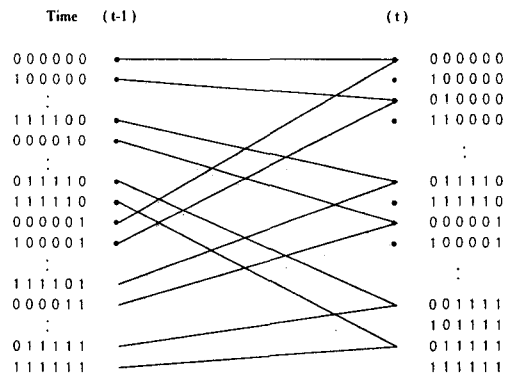


그림 1. 구속장이 7인 길쌈부호기의 격자도

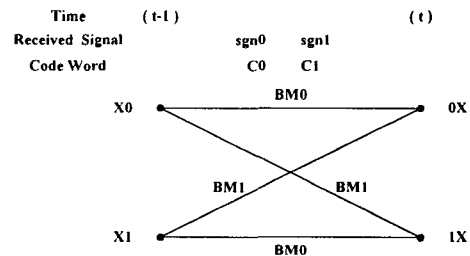


그림 2. 나비구조

상위비트를 제외한 5비트를 나타내며 상태의 변화 시 변하지 않는 부분이다. 비터비 복호기는

크게 BM(Branch Metric)부, ACS(Add Compare Select)부, TB(Trace Back)부로 구성된다. BM부는 전송된 심볼과 각각의 상태(구속장이 7인 경우 64상태)에서의 부호어와의 거리값을 계산한다. ACS부는 상태값을 갱신하고 TB부로 선택정보를 전송한다. TB부는 생존 경로에 대한 정보를 저장한 TBM을 갱신하고 최종 복호된 데이터를 출력한다.

3. 각 구속장별 에러확률

AWGN 채널에서 연성판정의 경우 비터비 복호의 에러확률은 다음과 같이 쓸 수 있다[4].

First-event 에러확률;

$$P(e) < Q\left(\sqrt{\frac{2dr E_b}{N_0}}\right) e^{dr E_b / N_0} T(D) \Big|_{D=e^{-r E_b / N_0}} \quad (1)$$

비트 에러확률;

$$P_u(e) < \frac{1}{k} Q\left(\sqrt{\frac{2dr E_b}{N_0}}\right) e^{dr E_b / N_0} \frac{\partial T(D, M)}{\partial N} \Big|_{N=1, D=e^{-r E_b / N_0}} \quad (2)$$

윗 식에 전달함수 T(D)를 적용하여 각 전달함수에 따른 first-event 에러확률과 비트 에러확률을 나타내면 그림 3, 4와 같다.

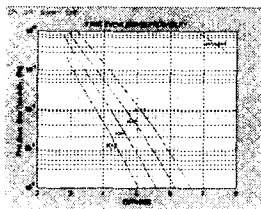


그림 3. First event 에러확률

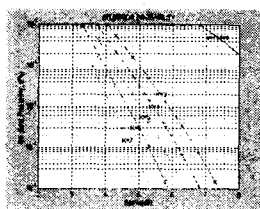


그림 4. 비트에러 확률

4. 비터비 복호기의 세부설계

4.1 BM부

BM부는 가지값 계산을 용이하게 하기 위해 2의 보수(2's complement)형태로 전송된 심볼을 부호절대값(signed magnitude)형태로 변환한 심볼과 상태천이에 따라 발생하는 각 상태에서의 부호어와의 거리를 구하는 부분이다. 각 상태에서의 부호어는 모두 4가지로써 이는 병렬구조 적용 시 모든 상태에서 가지값을 구할 필요가 없이

4가지 경우만을 계산함으로써 하드웨어를 절약할 수 있다. 각 상태에서의 가지값을 구하기 위해 SED(Squared Euclidean Distance)식이 사용되는데 이는 곱셈기가 필요하게 되어 많은 양의 하드웨어가 요구된다. 따라서 본 연구에서는 SED와 등가인 아래의 식을 이용하여 위의 문제점을 해결할 수 있었다[5].

$$BM = \sum_{i=0}^3 \begin{cases} |sgn_i| & sgn_i \text{ 와 } C_i \text{의 부호가 같은 경우} \\ 0 & sgn_i \text{ 와 } C_i \text{의 부호가 다른 경우} \end{cases} \quad (3)$$

sgn : 4bit로 양자화 된 수신 심볼
C : 각 상태에서의 부호어

이러한 방식의 장점은 가지값이 수신된 심볼의 절대값 자체로부터 구해지므로 SED를 직접 계산하는 것보다 계산시간 및 하드웨어 크기가 줄어든다.

4.2 ACS부

ACS부에서는 BM부로부터 수신된 가지값과 누적된 이전 상태값을 이용하여 새로운 상태값을 갱신하고 저장하며 다음 ACS 동작수행을 위해 갱신된 상태값을 전송하고, 또한 갱신된 상태값의 비교 선택 시 발생하는 정보를 TB부로 전달하는 동작을 수행한다. ACS의 기본블록은 4개의 덧셈기와 2개의 비교선택기, 2개의 truncation logic, 상태값 저장을 위한 2조의 레지스터로 구성되어 나비구조의 형태를 지닌다. 즉, 두 개의 상태값과 각 가지값을 더한 뒤 작은 값으로 상태값을 갱신한다. 상태값 (SM ; State Metric)을 구하는 식은 다음과 같다.

$$\begin{aligned} SM0X &= \min(SMX0+BM0, SMX1+BM1) \\ SM1X &= \min(SMX0+BM1, SMX1+BM0) \end{aligned} \quad (4)$$

기존의 메모리를 사용하는 경우 ACS부에서 최소 상태값을 구하는 과정이 필요하게 되는데 이럴 경우 64상태의 상태값을 모두 비교하여야 하는 번거로움이 있다. 또한 입력되는 심볼이 12Mbps로, 약 83nsec에서 모든 과정을 수행해야 한다. 따라서 83nsec 안에 64개의 상태값을 모두 비교하기에는 시간적인 제약이 있다. 본 연구에서 제안한 방식의 경우 이러한 최소 상태값 결정과정이 필요하지 않는 장점을 가지게 되는데 이러한 알고리즘에 대해서는 TB부에서 자세히 설명하기로 한다.

4.3 TB부

TB부에서는 ACS부에서 결정된 생존경로에 대한 정보를 저장하며 저장과 동시에 이전 TBM (Trace Back Memory)에 저장된 각 상태의 정보값을 갱신하게 되고 일정길이(구속장의 4-5배)를 처리 후 최종 복호 데이터를 출력한다. TB의 기본블록은 2개의 multiplexer와 생존경로 저장을 위한 2개의 레지스터로 구성된다. 본 연구에서는 depth를 48로 하였으며 역 추적과정을 간편화하고 비터비 디코더의 종단에서 majority voting이 가능한 새로운 역추적 알고리즘을 적용한다. 구속장이 7일 경우 각 상태는 6비트(000000, 100000, ..., 111111)로서 표현이 가능하고 짝수상태(000000, 010000, ..., 011111)와 홀수상태(100000, 110000, ..., 111111)로 나눌 수 있다. 임의의 시간 t-1에서 t로의 천이 시 모든 상태로 입력되는 경로는 두 가지가 존재하는데 짝수상태로 천이 되는 경우는 TBX0와 TBX1에서 입력이 0인 경우에 TB0X로 천이 되는 경우이고 홀수상태로 경우는 TBX0와 TBX1에서 입력이 1인 경우에 TB1X로 천이 되는 경우이다. 예를 들어 상태 000000에서 입력으로 0이 인가되면 다음상태로의 천이는 짝수상태인 000000이고, 입력으로 1이 인가되면 다음상태로의 천이는 홀수상태인 100000이다. 이와 같은 성질을 이용하면 TBM 갱신과정을 다음과 같이 표현할 수 있다.

$$\begin{aligned}
 TB0X &= \begin{cases} [TBX0(2:48), 0] & \text{for sel } 0 = 0 \\ [TBX1(2:48), 0] & \text{for sel } 0 = 1 \end{cases} \\
 TB1X &= \begin{cases} [TBX0(2:48), 1] & \text{for sel } 1 = 0 \\ [TBX1(2:48), 1] & \text{for sel } 1 = 1 \end{cases} \\
 \text{sel} &: \text{ACS로부터의 선택신호}
 \end{aligned}
 \tag{5}$$

이러한 과정을 매 단계마다 수행하고 일정길이의 depth를 거치면서 각 상태의 입력정보를 가진 64개의 TBM은 복호 과정을 수행하게 된다. 이 과정에서 일정 길이의 TB과정이 지나면 각 상태의 입력정보를 저장한 TBM은 일정한 값 (0 또는 1)으로 수렴하게 되는데 이는 48단계를 거치는동안 ACS부에서 매 단계마다 높은 확률을 가진 경로만을 선택하게 되고 TB부에서는 높은 확률을 가지는 경로에 대한 정보만을 선택하고 계속해서 갱신과정을 거치므로 매 단계를 거치는동안 레지스터의 뒷부분에 저장되는 모든 TBM은 동일한 값을 가지게 된다[6]. 따라서 모든 상태에서 최소 상태값을 찾을 필요가 없게되

어 역추적 과정이 간편화 되었으며 비터비 디코더의 종단에서 majority voting이 가능한 장점을 가진다.

5. 하드웨어 모의 실험

5.1 AWGN 생성

구현된 비터비 디코더의 성능검증을 위해 중앙극한정리(central limit theorem)을 적용하여 AWGN을 생성하였다. 먼저 균일(uniform)분포를 가진 8개의 랜덤변수를 생성하고 가우시안 분포를 가진 AWGN을 생성하기 위해 더한 후 정규화와 양자화 과정을 거친다. 그림 4는 잡음발생기의 블록도를 나타낸다.



그림 5. AWGN 생성블록

또한 다양한 SNR에 맞는 잡음을 생성하기 위해 구성되었으며 2개의 FPGA 칩 중 첫 번째에 길쌘부호기, BM부, ACS부와 같이 위치한다.

5.2 구현된 비터비 복호기의 성능검증

하드웨어 모의실험을 위해 PCB상에 2개의 FPGA를 사용하여 구현을 하였으며 첫 번째의 칩에는 길쌘부호기, AWGN 생성블록, BM부, ACS부를, 두 번째 칩에는 TB부와 에러율 표시

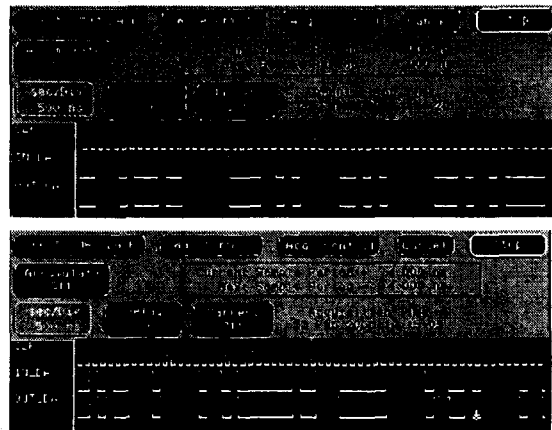


그림 6. Logic Analyzer를 통해본 실시간 모의실험 결과

제어부를 포함시켰다. 그림 6은 PCB상에서 하드웨어 모의실험 한 결과를 Logic Analyzer를 통해 입,출력 파형을 비교한 것으로써 입력신호에 인가된 잡음을 모두 정정했음을 보여주고 있다. 비터비 복호기의 구현을 위해 ALTERA사의 EPF10K100ARC 240-3 칩 두 개를 사용하였으며 길쌈부호기, ACS부, AWGN, 생성블록이 구현된 첫 번째 칩은 73%의 LE(Logic Element) 사용율을, TB부와 에러율 표시제어부를 구현한 두 번째 칩은 57%의 LE 사용율을 보이며 전체적으로 65%, 6521개의 LE가 사용되었다. 그림 7에 비터비 복호기 구현을 위해 제작된 testbed를 나타내었다.

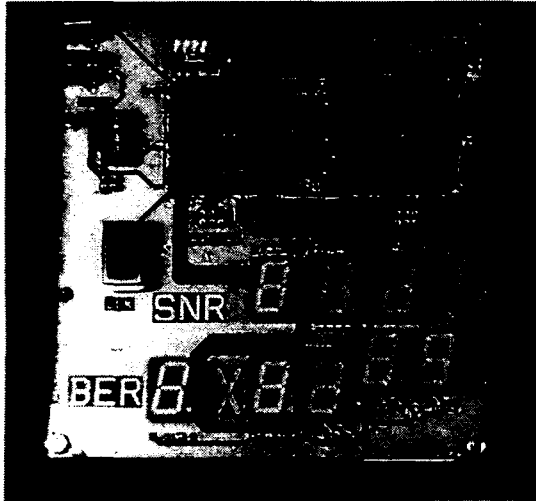


그림 7. 구현을 위해 제작된 PCB 기판

그림 8은 본 연구에서 구현된 비터비 복호기에 대한 모의실험 결과를 보여준다. 오른쪽 곡선은 uncoded BPSK의 에러율을 나타내며 왼쪽의 두 곡선은 기존의 구속장이 7, 부호화율이 1/2, 연성판정의 경우에 대한 비터비 복호기의 에러율과 구현된 비터비 복호기의 에러율을 나타낸다.

5. 결 론

구현된 비터비 복호기에서 12Mbps의 고속 데이터를 처리하기 위해 32조의 병렬구조를 사용하였으며 메모리를 사용한 방식과는 다른 레지스터 교환방식을 적용하였고 이로 인해 복호기의 종단에서 majority voting이 가능하고 하드웨어

를 절약할 수 있었다. AWGN 환경에서 실시간 하드웨어 모의실험을 위해 AWGN 생성블록을 추가하였고 두 개의 FPGA 칩을 사용하여 비터비 복호기를 구현하였다.

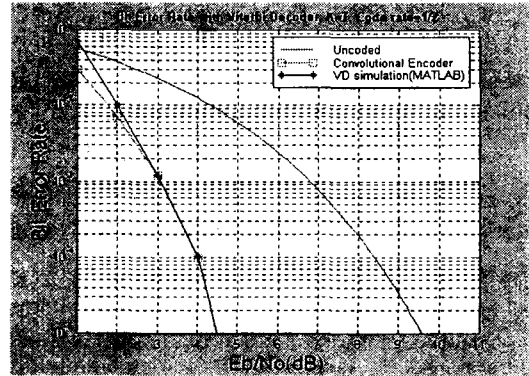


그림 8. 모의실험 결과

참고 문헌

- [1] Gennady Feygin and P.G. Gulak "Architectural Tradeoffs for Survivor Sequence Memory Management in Viterbi Decoders," *IEEE Trans. Commun.*, vol. 41, No.3, March 1993
- [2] Emmanuel BOUTILLON, Nicolas DEMASSIEUX "High Speed Low Power Architecture For Memory Management in a Viterbi Decoder," *Proc. IEEE, Int. Symposium of Circuit and Systems*, vol.4 1996
- [3] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications : High Speed Physical Layer in the 5 GHz Band", IEEE Standard Department. 1999
- [4] Jhong Sam Lee, Leonard E. Miller "CDMA system engineering handbook", Artech house, Boston London, 1998
- [5] 송상섭외 4인, "CDMA용 비터비 복호기의 최적구조 제시 및 FPGA 구현에 관한 연구," ETRI 보고서. 1995
- [6] 임명섭외 2인, "무선 LAN용 OFDM 모델 라이브러리 개발 및 FPGA 구현," ETRI 보고서. 1999