

# 1000Base-T에서 동작하는 Viterbi Decoder 구현

정재우\* · 정해\*

\*금오공과대학교 전자공학과

## Implementation of a Viterbi Decoder Operated in the 1000Base-T

Jae-woo Jung\* · Hae Chung\*

\*Department of Electronic Engineering, Kumoh National Institute of Technology

E-mail : hchung@kumoh.ac.kr

본 연구는 미래창조과학부 정보통신산업진흥원의 IT명품인재양성사업 (NIPA-2013-H0203-13-1001)의 연구결과로 수행되었음

### 요 약

UDTV와 같이 고품질의 애플리케이션이 등장함에 따라 고속 고용량의 통신서비스가 요구되고 있다. 이를 위해 통신시스템은 데이터 처리 속도를 높이고 다양한 에러 정정기법을 사용한다. 본 논문에서는 UTP 케이블을 이용하여 1 Gbps를 전송하는 1000BASE-T에 적용되는 비터비 디코더를 구현한다. 1000BASE-T에서는 각 pair 당 125 MHz의 PAM-5로 변조된 신호가 전송되므로 이 디코더는 최소 125 MHz 이상의 속도로 동작해야 한다. 이를 위하여 파이프라인과 병렬처리를 사용하여 디코더를 FPGA에서 구현하고 로직분석기를 이용해서 125 MHz로 동작함을 확인한다. 최종적으로 비터비 디코더가 부가된 임의의 에러에 대하여 원래의 데이터를 복구하는 것도 보여준다.

### ABSTRACT

As appearance of high-quality service such as UDTV application, high-speed and high-capacity communication services are required. For this, communication systems increase the data processing speed and use various error correction techniques. In this paper, we implement the Viterbi decoder applied in 1000BASE-T with 4 pairs UTP cable. The minimum operating speed of the Viterbi decoder should be more than 125 MHz because 125 MHz PAM-5 signal is transmitted on each pair of cables in 1000BASE-T. To do this, we implement the decoder by using the pipeline and parallel processing and verify the operation with 125 MHz by using a logic analyzer. Finally, we will show that the decoder recovers the original data for the added random error data.

### 키워드

Convolutional code, Viterbi decoder, 1000Base-T, Error correction, Ethernet

### I. 서 론

1000Base-T는 통신환경이 실시간 대용량의 멀티미디어 정보 전송이 필요해지면서 Ethernet의 한 갈래로 개발되었다. 이미 설치되어 있는 전송망을 그대로 사용할 수 있는 장점이 있어 고속 통신 근거리 통신망에 널리 사용되고 있다 [1]. 1000Base-T는 UTP (Unshielded Twisted Pair) cable내의 4쌍의 선을 이용하여 5레벨의 PAM5신호를 전송한다. 각 선으로 5레벨의 신호가 125 MHz로 전송되므로 4쌍의 선로의 총 전송률은  $2 \text{ bit} \times 4 \times 125 \text{ MHz} = 1,000 \text{ Mbps}$ 가 된다. 신호전송 특성이 좋지 않은 UTP선을 이용하여 신호를

전송하므로 송수신기에는 채널 인코딩과 디코딩, DFE (Decision Feed-back Equalizer), NEXT (Near End Cross Talk) canceller, echo canceller 등의 디지털 신호처리가 필요하다 [1].

본 논문에서는 1000Base-T 물리 계층의 비터비 디코더를 FPGA (Field Programmable Gate Array)에서 VHDL (Very high speed integrated circuit Hardware Description Language)을 이용하여 설계 및 합성해서 125 MHz의 속도로 동작 하도록 구현하는 것을 목표로 하였다. 1000Base-T규격에서 송신측이 사용하는 인코딩은 TCM (Trellis Coded modulation)을 사용한다. TCM에서 채널 부호화는 블록 부호나 콘볼루션 부호 모두 사용할 수 있는

나 1000Base-T에서는 콘볼루션 부호를 사용한다.

본 논문의 구성은 II절에서 송신측에서 채널상의 에러를 정정하기 위해 사용된 기법을 살펴보고, III절에서 비터비 디코더의 기본적인 구조를 설명한다. IV절에서는 비터비 디코더의 구현 방법을 설명하고, V절에서 구현된 비터비 디코더의 성능을 검증하고, 마지막 VI절에서 결론을 맺는다.

## II. 1000Base-T에서의 변조

### 2.1 TCM

1000Base-T에서는 채널상의 에러를 정정하기 위해 콘볼루션 부호를 이용하여 TCM을 생성한다. 콘볼루션 부호는 블록 부호와 다르게 기억장치가 있다. 그래서 콘볼루션 부호기는 현재의 입력뿐만 아니라 과거의 입력과 함께 현재의 출력값을 이용해서 상태도와 코드를 생성한다. 규격에서 제시된 1000Base-T PCS Transmit의 TCM 회로의 구조는 그림 1과 같다 [2].

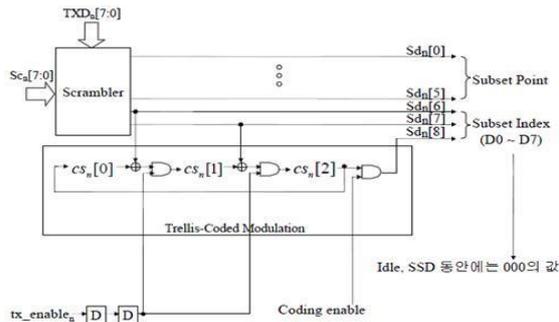


그림 1. 1000Base-T Trellis Coded Modulation

TCM의 상태도에 따라 송신할 심벌을 결정하는 것은 Sdn[8:6] 3 bit에 의해 표 1과 같이 D0~D7의 그룹 중 하나의 그룹을 결정하고, 나머지 5 bit의 값을 보고 규격에 제시된 Look-up Table에 명시된 심벌들로 매칭한다. 이때, 심벌에 대해서 각각 {+1, -1}을 X 그룹, {+2, 0, -2}를 Y그룹으로 구분하여 각 pair에 대한 심벌 값을 표 1과 같이 표현한다 [3].

1000Base-T에서는 PAM5 신호를 이용한다. 그래서 4 쌍의 선에서 만들어 낼 수 있는 심벌의 수는  $5^4 = 625$  개이다. 8 bit의 신호를 전송하는데 필요한 서로 다른 심벌의 수는  $2^8 = 256$  개이지만 그림 1과 같이 TCM을 이용하여 심벌의 개수를 두 배로 늘려 512 개의 심벌을 사용하여 심벌들 간의 거리를 넓힘으로써 잡음에 대한 면역성을 높인다. 이 그룹 내 심벌들 간의 유클리디안 거리는 TCM에 의해 최소 4 이상이 되도록 배치된다.

표 1. 1000Base-T에서 그룹과 심벌 형식

| 그룹 | X심벌  | 심벌 수     | Y심벌  | 심벌 수     |
|----|------|----------|------|----------|
| D0 | XXXX | 16 codes | YYYY | 81 codes |
| D1 | XXXY | 24 codes | YYXX | 54 codes |
| D2 | XXYY | 36 codes | YYXX | 36 codes |
| D3 | XXYX | 24 codes | YYXY | 54 codes |
| D4 | XYYX | 36 codes | YXXY | 36 codes |
| D5 | YYYY | 54 codes | YXXX | 24 codes |
| D6 | XYXY | 36 codes | YXYX | 36 codes |
| D7 | XYXX | 24 codes | YXYX | 54 codes |

## III. 비터비 디코더 구조

비터비 디코더의 구조는 그림 2와 같이 BM (Branch Metric), ACS (Add Compare Select), TBM (Trace-Back Memory) 블록으로 나뉜다. 지금부터 각각의 블록에 대하여 상세한 설명을 하기로 한다.



그림 2. 비터비 디코더의 전체 구조

### 3.1 BM

BM에서는 TCM된 심벌을 수신하여 상태 천이에 따른 가지 평가량을 계산한다. 이는 채널 상에서 잡음이 부가되어 전송된 심벌을 복호하기 위한 과정의 첫 단계이다. AWGN (Additive White Gaussian Noise) 채널의 수신벡터는 각 샘플에 부가된 잡음이 독립적이므로, 조건부 확률은 식 (1)과 같다.

$$P(r|c) = \prod_{l=0}^{L+m-1} p(r_l|c_l) \quad (1)$$

여기서  $r_l = (r_l^{(1)}, \dots, r_l^{(n)})$ ,  $c_l = (c_l^{(1)}, \dots, c_l^{(n)})$ 는 콘볼루션 부호 격자도의  $l$  번째 가지에 해당하는 길이가  $n$ 인 수신벡터와 부호벡터이다. 그리고  $p(r_l|c_l)$ 를 가지 확률이라 하며, 여기에 로그를 취하면, 식 (2)와 같은 가지 평가량이 된다. 가지 평가량은 각 가지의 비트 평가량의 총 합이다.

$$M(r_l|c_l) = \log(p(r_l|c_l)) \quad (2)$$

$$M(r_l^{(i)}|c_l^{(i)}) = \log(p(r_l^{(i)}|c_l^{(i)}))$$

$$M(r_l|c_l) = \sum_{i=1}^n M(r_l^{(i)}|c_l^{(i)})$$

### 3.2 ACS

ACS는 BM에서 계산된 가지 평가량과 각 상태

천이가 가능한 경로의 평가량을 더하여 각 경로 평가량을 계산한다. 그리고 각 경로 평가량을 비교하여 최소 평가량을 갖는 경로를 선택한다. ACS 블록에서는 각 상태에서 최소 평가량을 갖는 경로에 대한 정보와 최종적으로 생존한 경로에 대한 데이터를 생성한다.

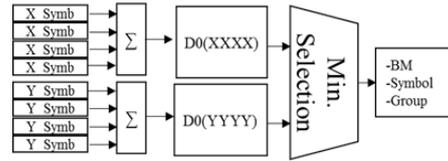


그림 4. D0 경우의 BM 블록도

### 3.3 TBM

TBM은 ACS에서 생성된 선택 경로와 생존 경로를 가지고 경로를 역추적한다. 경로를 추적할수록 모든 상태의 생존경로 값이 공통된 값을 갖기 시작하고 최종적으로 수신한 데이터를 복구한다. 이때 어떤 시점에서의 각 노드는 생존경로를 기억하고 매 시점마다 모든 노드에서의 생존경로를 갱신하여 관리해야 한다. 이는 큰 시간 지연과 많은 메모리를 필요로 한다 [4].

### 4.2 ACS 구현

가지 평가량과 이전 경로 평가량의 합으로 PM을 계산하고 이 값 중에서 가장 작은 값을 새로운 PM으로 선택하고 상태를 업데이트한다. 이때 사용되는 경로 선택기의 구조는 그림 5와 같고 각 상태에서의 입력 심벌과 선택되는 경로는 표 2와 같다. 8개의 경로 선택기에서 선택된 8개의 경로를 다시 비교하여 가장 작은 경로 값을 생존경로로 기록하게 된다.

## IV. 비터비 디코더의 구현

### 4.1 BM 구현

1000BASE-T에서 4 pair의 선로에서  $n$  번째 클락 시점에 유입되는 각각의 신호를  $A_n, B_n, C_n, D_n$ 이라 정의 한다 [2]. BM을 계산하기 위해서 4 pair의 수렴한 DFE의 출력 데이터  $A_n, B_n, C_n, D_n$ 에 대해 그림 3과 같은 슬라이스를 이용해서 각각 X심벌과 Y심벌을 구분 지어 비트 평가량을 계산하고, 그림 4와 같이 각 그룹의 심벌형식에 맞게 비트 평가량을 더하여 가지 평가량을 구하게 된다. 이 때, 가지값을 계산하는데 실제 값(R)과 슬라이스에서 판단된 값(S)의 차이 값이 사용된다. 가지 평가량을 계산하는 공식은 식 3과 같다.

$$BM_i = (R_a - S_a)^2 + (R_b - S_b)^2 + (R_c - S_c)^2 + (R_d - S_d)^2 \quad (3)$$

위 공식의 아래 첨자(a, b, c, d)는 각 신호에 채널을 나타내고  $i$ 는 0~7까지의 그룹을 나타낸다. 1000Base-T에서는 4개의 전송채널이 있으므로 R 값과 S 값이 각각 4개씩 존재한다. BM 블록은 D0~D7까지 8개의 그룹에 대한 모든 데이터 정보를 생성하여 각 그룹에서 X 심벌과 Y 심벌 중에 작은 가지 평가량과 심벌정보, 그룹정보를 ACS 블록으로 보낸다. 그림 4와 같은 BM 블록이 각 그룹에 맞게 8개가 병렬로 구성되어 있다.

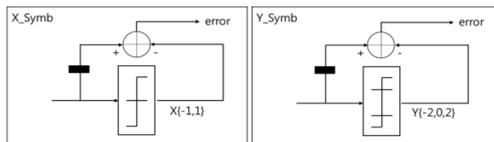


그림 3. 슬라이싱

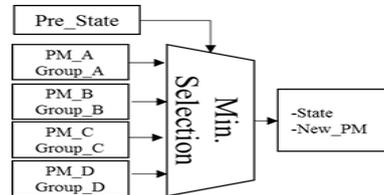


그림 5. 경로 선택기의 구조

표 2. 각 상태의 심벌과 전송경로

| Pre_State : 000(S0) |           | Pre_State : 001(S4) |           |
|---------------------|-----------|---------------------|-----------|
| Even Group          | New State | Even Group          | New State |
| 000(D0)             | 000(S0)   | 000(D0)             | 010(S2)   |
| 010(D2)             | 100(S1)   | 010(D2)             | 110(S3)   |
| 001(D4)             | 010(S2)   | 001(D4)             | 000(S0)   |
| 011(D6)             | 110(S3)   | 011(D6)             | 001(S4)   |

| Pre_State : 100(S1) |           | Pre_State : 101(S5) |           |
|---------------------|-----------|---------------------|-----------|
| Odd Group           | New State | Odd Group           | New State |
| 100(D1)             | 001(S4)   | 100(D1)             | 011(S6)   |
| 110(D2)             | 101(S5)   | 110(D2)             | 111(S7)   |
| 101(D3)             | 011(S6)   | 101(D3)             | 001(S4)   |
| 111(D4)             | 111(S7)   | 111(D4)             | 101(S5)   |

| Pre_State : 010(S2) |           | Pre_State : 011(S6) |           |
|---------------------|-----------|---------------------|-----------|
| Even Group          | New State | Even Group          | New State |
| 000(D0)             | 100(S1)   | 000(D0)             | 110(S3)   |
| 010(D2)             | 000(S0)   | 010(D2)             | 010(S2)   |
| 001(D4)             | 110(S3)   | 001(D4)             | 100(S1)   |
| 011(D6)             | 010(S2)   | 011(D6)             | 000(S0)   |

| Pre_State : 110(S3) |           | Pre_State : 111(S7) |           |
|---------------------|-----------|---------------------|-----------|
| Odd Group           | New State | Odd Group           | New State |
| 100(D1)             | 101(S5)   | 100(D1)             | 111(S7)   |
| 110(D2)             | 001(S4)   | 110(D2)             | 011(S6)   |
| 101(D3)             | 111(S7)   | 101(D3)             | 101(S5)   |
| 111(D4)             | 011(S6)   | 111(D4)             | 001(S4)   |

### 4.3 TBM 구현

TBM에서는 BM에서 생성한 심벌 (12 bit)와 ACS에서 얻어진 경로정보 (3 bit), 생존경로 (3 bit)를 입력으로 받아 경로에 맞는 심벌을 최종적으로 Viterbi\_output (12 bit)로 출력한다.

### V. FPGA 합성 및 실험 결과

본 논문에서는 비터비 알고리즘을 VHDL로 설계 하였으며, 합성툴은 Xilinx의 ISE를 사용하였다. 표 3은 합성결과로 설계된 비터비 디코더가 125 MHz이상에서 동작 가능한 결과를 확인했다. 그리고 ISE의 ISIM을 이용하여 임의의 입력 데이터에 천이 불가능한 1 pair error를 발생시켜 구현한 비터비 디코더의 기능을 검증하고 그림 6과 같은 실제 FPGA에 125 MHz로 동작시켜 로직분석기로 천이 불가능한 1 pair error가 발생된 데이터를 원래 데이터로 복원한 출력을 확인하였다.

표 3. 합성 결과

| Target Device : SPARTAN6(xc6slx150) |                    |     |
|-------------------------------------|--------------------|-----|
| Timing Report                       |                    |     |
| Maximum Frequency:                  | 160.205MHz         |     |
| Design Summary                      |                    |     |
| Slice Logic Utilization             |                    |     |
| Number of Slice Registers:          | 7345 out of 184304 | 3%  |
| Number of Slice LUTs:               | 2721 out of 92152  | 2%  |
| Number used as Logic:               | 2721 out of 92152  | 2%  |
| Slice Logic Distribution            |                    |     |
| Number of LUT Flip Flop pairs used: | 8590               |     |
| Number with an unused Flip Flop:    | 1245 out of 8590   | 14% |
| Number with an unused LUT:          | 5869 out of 8590   | 68% |
| Number of fully used LUT-FF pairs:  | 1476 out of 8590   | 17% |
| Number of unique control sets:      | 10                 |     |
| IO Utilization                      |                    |     |
| Number of IOs:                      | 46                 |     |
| Number of bonded IOBs:              | 34 out of 498      | 6%  |
| Specific Feature Utilization        |                    |     |
| Number of BUFG/BUFGCTRLs:           | 1 out of 16        | 6%  |
| Number of DSP48A1s:                 | 64 out of 180      | 35% |



그림 6. 사용된 FPGA

그림7, 그림8는 로직분석기에서 비터비의 입력 din\_an, din\_bn, din\_cn, din\_dn와 출력 An\_symb, Bn\_symb, Cn\_symb, Dn\_symb을 확인한 샘플이다. 그림7, 그림8에서의 {96, 48, 0, -48, -96}는 {2, 1, 0, -1, -2}를 나타내는 전압레벨로써 8 bit 신호이다. 그림 7의 X커서가 있는 부분을 보면 D2 심벌 {96, 0, 48, 48}을 송신했지만 수신측에서 D1 심벌 {96, 0, 80, 48}을 받아 1 pair error가 발생한 데이터이고 그림 8의 O커서가 있는 부분을 보면 원본 데이터인 D2 심벌 {96, 0, 48, 48}로 복원한 데이터이다. 따라서 구현된 디코더가 올바르게 에러정정을 하는 것을 알 수 있다.

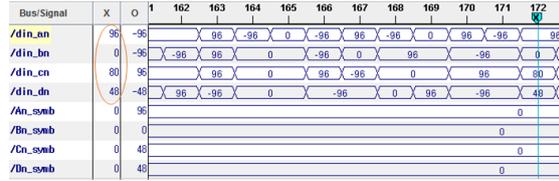


그림 7. 1 pair error 발생된 임의의 입력 데이터

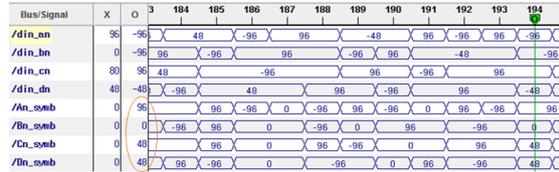


그림 8. 비터비 디코더에 의해 복원된 데이터

### VI. 결론

본 논문에서는 4 pairs UTP 상에서 동작하는 1000BASE-T에 적용되는 수신부의 에러정정 방식인 콘볼루션 코드를 위한 디코더를 비터비 알고리즘을 이용하여 구현하였다. 이 디코더는 크게 BM, ACS, 그리고 TBM 블록으로 구성되며, 이를 FPGA를 이용하여 구현하였으며 125 MHz로 동작하도록 설계하였다. 최종적으로 구현된 디코더의 동작을 확인하기 위하여 임의의 에러를 추가하였고, 에러를 정정하는 것을 확인할 수 있었다. 1000BASE-T는 캠퍼스, 빌딩, 그리고 다양한 사업장에서 고속의 LAN을 구축하는데 적용될 뿐만 아니라, 추후에 UDTV와 같은 고속의 서비스를 수용하기 위하여 현존하는 100 Mbps의 아파트 LAN을 밀어내고 가입자 망의 중추적인 역할을 할 것으로 보인다. 본 논문의 연구 결과는 해당 칩을 개발하는 업체나 이와 유사한 통신 칩을 개발하는 업체에 도움이 될 것으로 판단된다.

### 참고문헌

- [1] 이형록, “1000Base-T를 위한 고속 Viterbi Decoder 설계 및 VLSI 구현”, 서울대학교 대학원 석사학위 논문, 2000년.
- [2] IEEE Std 802.3 Section 3, “Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications,” Approved 26 Dec. 2008.
- [3] G. Ungerboeck, “Trellis-Coded Modulation with Redundant Signal Sets“, *IEEE communication mag*, vol.25, no. 2, pp.5-21, Feb. 1987.
- [4] J. K. Omura, “On the Viterbi Decoding Algorithm”, *IEEE trans. Inform. Theory*, Vol. 15, pp.177-179, Jan. 1969.