

1000BASE-T의 4조 PAM-5 신호 상에서 동작하는 비터비 디코더의 구현

정재우 · 정해*

Implementation of a Viterbi decoder operated in 4 Dimensional PAM-5 Signal of 1000Base-T

Jae-Woo Jung · Hae Chung*

Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi 730-701, Korea

요 약

LAN 방식은 국내의 초고속 인터넷 서비스에서 가장 널리 사용되며, UHD TV와 같은 고속의 서비스를 제공하기 위하여 100 메가급에서 1 기가급 인터넷으로 빠르게 전환되고 있다. 1000BASE-T 물리계층은 1 Gbps의 전송속도를 달성하기 위해, 4조의 UTP상에서 각 조당 125 MHz의 PAM-5신호로 데이터를 전송한다. 채널 상에서 발생한 오류를 정정하기 위하여 송신측에서는 컨벌루션 부호와 PAM-5신호를 결합한 TCM을 사용하고, 수신측에서는 비터비 복호기를 사용한다. 본 논문에서는 1000BASE-T의 수신측에서 최소 125 MHz 속도와 최대 2조까지 오류 정정 능력을 제공하는 비터비 디코더를 구현한다. 그리고 구현한 디코더를 논리분석기로 이용해서 동작속도와 오류 정정 능력을 검증한다.

ABSTRACT

The LAN method is the most widely used in domestic high-speed internet access and rapidly moving to 1 Gbps Ethernet from 100 Mbps one to provide high-speed services such as UHD TV. The 1000BASE-T PHY with 4 pairs UTP transmits a PAM-5 signal at the 125 MHz clock per each pair to achieve 1 Gbps rate. In order to correct errors over the channel, the transmitter uses a TCM which is combined the convolutional encoder and PAM-5, and the receiver uses the Viterbi decoder. In this paper, we implement a Viterbi decoder which can correct two pair errors and operate at the least 125 MHz clock speed. Finally, we will verify the error correction function and the operating speed of the implemented decoder with a logic analyzer.

키워드 : 1000Base-T, 컨벌루션 부호, 비터비 디코더, 오류정정, 격자 부호화 코드

Key word : 1000Base-T, Convolutional code, Viterbi decoder, Error correction, TCM

접수일자 : 2014. 05. 06 심사완료일자 : 2014. 05. 27 게재확정일자 : 2014. 06. 10

* **Corresponding Author** Hae Chung (E-mail: hchung@kumoh.ac.kr, Tel: +82-54-478-7484)

Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, 730-701, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2014.18.7.1579>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

방송통신위원회의 2013년도 국내 초고속 인터넷 서비스 현황 발표에 의하면 LAN 방식의 가입자가 36.3%로서 xDSL 방식을 누르고 가장 큰 비중을 차지하고 있다. LAN 방식은 올해부터 100 메가급에서 1 기가급 인터넷으로 전환되고 있으며, FTTH (Fiber-to-the-Home)와 더불어 가장 높은 성장세를 보이는 이유는 UHD (Ultra High Definition) TV와 같은 고속의 서비스를 실시간으로 제공하는데 가장 적합한 방식이기 때문이다. 현재는 기가 인터넷의 적용지역은 약 17% 수준이고, 미래창조과학부는 2017년까지 기가 인터넷을 90% 이상 공급할 계획을 세우고 있다[1].

기가비트 이더넷이 4조 (4 pairs)의 UTP (Unshielded Twisted Pair cable) 상에서 각각 125 MHz의 PAM-5 신호로 동작할 때의 규격을 1000BASE-T라고 한다[2]. PAM-5 신호는 신호레벨간의 간격이 좁아 전송 중 노이즈의 영향에 민감하다. 따라서 정보의 신뢰성 보장과 채널 용량 한계를 극복하기 위해 송신측에서 TCM (Trellis Coded Modulation)을 적용하여 데이터를 전송한다[3,4]. 그리고 수신기는 채널코딩 된 신호를 복호화하기 위해 DFE (Decision Feed-back Equalizer), NEXT (Near End Cross Talk) canceller, echo canceller, 비터비 디코더[5-7] 등의 디지털 신호처리가 필요하다.

최근 비터비 디코더의 연구는 여러 무선 통신 환경과 대용량 저장 매체에서 초고속, 저전력으로 동작하는 것에 목적을 두고 있다. 비터비 디코더는 ACS (Adder-Compare-Select)와 TBM (Trace-Back Memory)의 구현이 주로 다루어지는데 ACS의 경우 연산 구조나 연산 알고리즘의 연구를 통해 성능을 개선하고 있고[8-10], 대용량 저장 매체의 경우는 ACS를 아날로그방식으로 바꿔서 성능을 개선하지만 복잡한 코딩이 필요한 통신 환경에서는 사용하기가 어렵다[11]. TBM의 경우는 생존경로와 역추적 방법을 연구하여 성능을 개선하거나 사용하는 메모리를 변경하여 성능을 개선하려고 있다[12-15]. 최근 국내에서는 1000Base-T를 가지고 대부분의 주거지역에 포설된 2조의 UTP를 계속 사용하여 500 Mbps급의 전송속도로 만드는 연구가 진행되어 유선환경에서의 비터비 디코더 연구가 필요해지고 있다[16,17].

본 논문에서는 1000Base-T의 물리 계층에서 동작하

는 비터비 디코더를 구현하였다. 1000Base-T에서는 1Gbps의 전송속도를 제공하기 위해 PCS (Physical Coding Sublayer)와 GMII (Gigabit Media Independent Interface)간에 8 bit의 데이터가 125MHz의 속도로 처리된다. 그러므로 비터비 디코더 역시 최소 125 MHz 이상의 속도로 동작하고 신호가 8 ns의 간격을 두고 처리되어야 한다. 그리고 1000Base-T의 표준안인 802.3ab에서 제시한 BER (Bit Error Rate)이 10^{-10} 이하를 만족하기 위해서는 채널 상의 오류를 정정해야 한다[2]. 본 논문의 목표는 125 MHz의 속도로 동작하고, 적은 역추적깊이 (Trace-back depth)로 1조 뿐만 아니라 2조 오류까지 정정하는 비터비 디코더를 FPGA (Field Programmable Gate Array)에서 HDL (Hardware Description Language)을 이용하여 구현하는 것이다.

본 논문의 구성은 II절에서 송신측에서 채널상의 오류를 정정하기 위해 사용된 기법을 살펴보고, III절에서 비터비 디코더의 구현 방법을 설명한다. IV절에서는 구현한 디코더의 성능을 검증하고, 마지막 V절에서 결론을 맺는다.

II. 1000Base-T에서의 부호기

1000Base-T에서는 채널상의 오류를 정정하기 위해 TCM을 이용하여 데이터를 생성한다. TCM에는 컨벌루션 부호기가 사용된다. 이 부호기는 n 비트의 입력을 받아 k 비트의 심벌로 부호화하는 부호기이다. 블록 부호와 다르게 기억소자가 있어서 현재의 입력뿐만 아니라 과거의 입력과 함께 현재의 출력 값을 이용해서 상태와 코드를 생성한다. 이때 하나의 입력이 다음 심벌의 형성에 영향을 미치는 횟수를 K 로 표시하며, 이것을 구속장 (constraint length)이라고 부른다. 구속장은 컨벌루션 부호기의 성능과 복호시, 비터비 디코더의 복잡도를 결정하는 중요한 파라미터이다[18]. 구속장은 $K=m+1$ 로 m 은 기억소자의 수이다. 규격에서 제시된 1000Base-T의 컨벌루션 부호기 회로는 그림 1과 같은 구조이다.

1000Base-T에서는 3개의 기억소자를 사용하므로 제시하는 TCM의 구속장은 4가 된다. PCS 송신기는 상위 계층에서 내려오는 8 bit 데이터인 TXD[7:0]와 LFSR (Linear Feedback Shift Register)로부터 생성된 8 bit 데

이터 $Sc_n[7:0]$ 을 혼화하여 $Sdn[7:0]$ 을 생성한다. 컨벌루션 부호기는 혼화기 출력의 상위 2 bit $Sdn[7:6]$ 와 기억 소자 $CS[2:0]$ 을 이용하여 $Sdn[8]$ 을 생성한다.

이렇게 생성된 $Sdn[8]$ 과 $Sdn[7]$, $Sdn[6]$ 을 이용해서 송신할 심벌을 표 1과 같이 D0~D7의 그룹 중 하나의 그룹을 결정한다. 그리고 나머지 $Sdn[5:0]$ 6 bit의 값을 보고 규격에 제시된 Look-up Table에 명시된 심벌들로 대응하여 PAM-5신호를 만든다. 이때, 심벌에 대해서 각각 $\{+1, -1\}$ 을 X그룹, $\{+2, 0, -2\}$ 를 Y그룹으로 구분하여 각 조에 대한 심벌 값들을 표 1과 같이 표현한다[2].

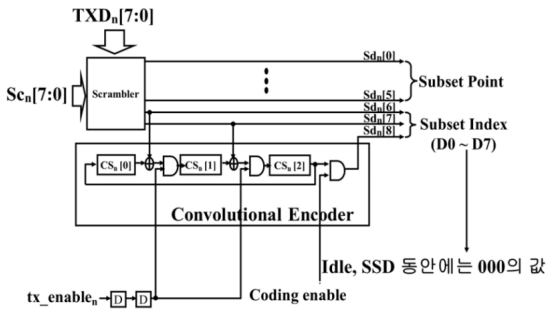


그림 1. 1000Base-T에서 컨벌루션 부호기의 구조
Fig. 1 Architecture of convolutional encoder in 1000Base-T

표 1. 1000Base-T에서 그룹과 심벌 형식
Table. 1 Group and symbol format in 1000Base-T

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	XYYY	54 codes	YXXX	24 codes
D6	XYXY	36 codes	YXYX	36 codes
D7	XYXX	24 codes	YXYY	54 codes

1000Base-T의 TCM은 PAM-5신호를 이용하기 때문에 4조의 선로에서 만들 수 있는 심벌의 수는 $5^4 = 625$ 개이다. 그리고 8 bit의 신호를 전송하는데 필요한 서로 다른 심벌의 수는 $2^8 = 256$ 개이지만, TCM을 이용하여

8개의 그룹을 만들고 6 bit로 각 그룹마다 $2^6 = 64$ 개 심벌을 만들어 두 배로 늘어난 512 개의 심벌을 사용한다. TCM은 그룹 내 심벌들 간의 유클리디안 거리 (Euclidean distance)를 최소 4 이상이 되도록 배치하고 심벌들 간의 거리를 넓힘으로써 잡음에 대한 면역성을 높인다.

III. 비터비 디코더의 구현

전체적인 비터비 디코더는 그림 2와 같이 가지 평가량 (Branch Metric, BM) 블록, 경로 평가량 (Path Metric, PM) 블록, 경로제어 (Path Control, PC) 블록, 역추적 메모리 (Trace-Back Memory, TBM) 블록으로 구성된다. 비터비 알고리즘을 수행하기 위해 BM 블록은 채널을 통해 수신된 데이터들의 여러 경로에 대한 가지 평가량을 계산한 후, 가지 평가량을 가지고 PM 블록에서 여러 경로에 대한 경로 평가량을 계산한다. 그리고 PC 블록에서는 그 중에서 가장 유사성이 높은 경로를 선택하고 TBM 블록으로 선택된 경로의 데이터를 복호하게 된다. 지금부터 각 블록의 구조와 구현방법에 대하여 상세히 설명하고자 한다.

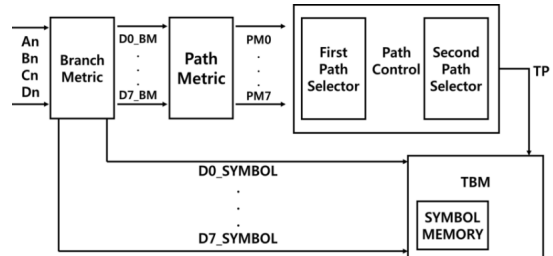


그림 2. 비터비 디코더의 전체 구조
Fig. 2 Overall architecture of Viterbi decoder

3.1. BM 블록

BM 블록에서는 변조된 4조의 신호를 수신하여 모든 경우의 가지 평가량을 계산한다. 이는 채널 상에서 잡음이 부가되어 전송된 심벌을 복호하기 위한 과정의 첫 단계이다. 1000BASE-T의 4조 선로에서 n 번째 클락 시점에 유입되는 각각의 신호를 An, Bn, Cn, Dn 이라 정의한다. AWGN (Additive White Gaussian Noise) 채널

을 통과한 각 채널의 유입된 실제 값을 각각 R_{An} , R_{Bn} , R_{Cn} , R_{Dn} 라 하고, 그에 대한 판단 값을 S_{An} , S_{Bn} , S_{Cn} , S_{Dn} 라 정의할 때, 비트 평가량은 실제 값 (R)과 슬라이스에서 판단된 값 (S)의 차이 값을 사용해서 구한다. D_i ($i = 0, \dots, 7$) 그룹의 가지 평가량(BM_i)는 다음과 같이 얻어진다[19].

$$BM_i = (R_{An} - S_{An})^2 + (R_{Bn} - S_{Bn})^2 + (R_{Cn} - S_{Cn})^2 + (R_{Dn} - S_{Dn})^2 \quad (1)$$

식 (1)의 BM_i 를 계산하기 위해서 각각의 신호 An , Bn , Cn , Dn 에 대한 비트 평가량을 그림 3과 같은 슬라이서를 이용해서 X심벌과 Y심벌을 구분 지어 계산한다. 그 이유는 1000Base-T에서 TCM 신호를 생성할 때, X 그룹과 Y 그룹으로 나누었기 때문이다. X_Slicer는 X 그룹 $\{-1, 1\}$ 의 경우에 대해 판단을 하여 비트 평가량을 생성한다. 그리고 Y_Slicer는 Y 그룹 $\{-2, 0, 1\}$ 의 경우에 대해 판단을 하여 비트 평가량을 생성한다. BM 블록은 D_i 그룹에 맞게 8개가 병렬로 구성되어 있다.

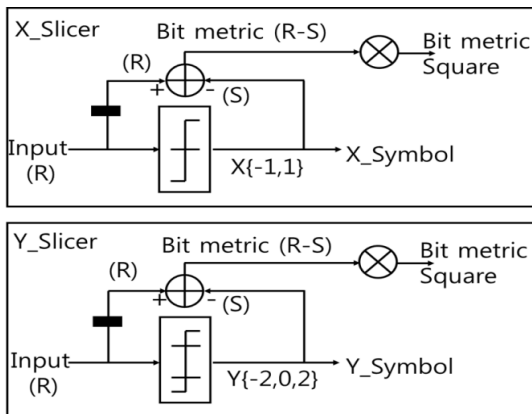


그림 3. 비트 평가량 계산을 위한 슬라이서 구조
Fig. 3 Slicer architecture for bit metric calculation

그림 4는 D0와 D1의 BM 블록도이다. 각 그룹의 BM 블록은 심벌형식에 맞게 슬라이서를 구성해 놓고 비트 평가량을 생성한다. 그리고 구성된 슬라이서에서 나온 비트 평가량을 모두 더하여 X심벌, Y심벌에 대한 두 개의 가지 평가량을 구한다. 마지막으로 CS (Comparison

& Selector) 블록에서 두 개의 가지 평가량 중에서 작은 값을 가진 가지 평가량을 선택하여 각 그룹의 가지 평가량 (16 bit)과 심벌 정보 (12 bit)를 생성한다.

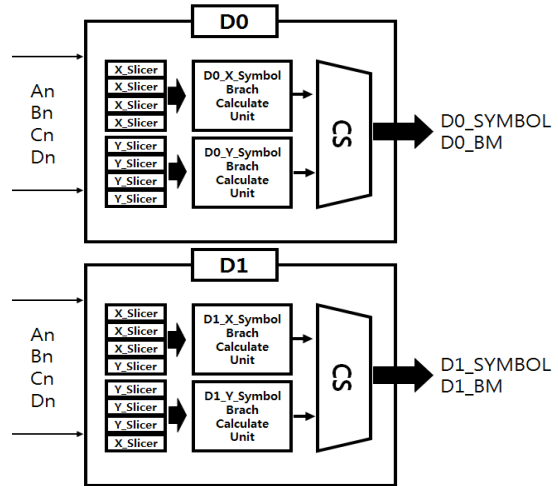


그림 4. D0와 D1의 BM 블록도
Fig. 4 BM block diagram of D0 and D1

예를 들어 D0 그룹의 경우, 4조 선로에서 입력되는 값을 표 1과 같이 X심벌 {XXXX}, Y심벌 {YYYY} 구조에 맞게 구조화된 슬라이서로 가지 값을 구한다. 그리고 D0_X_Symbol과 D0_Y_Symbol의 가지 계산 블록을 통해 두개의 가지 평가량을 계산하고 그 중 작은 값을 선택해서 심벌정보 (D0_SYMBOL)와 가지 평가량(D0_BM)을 생성하여 PM 블록으로 보내준다.

3.2. PM 블록

PM 블록은 BM 블록에서 계산된 가지 평가량 (BM_i)과 이전 경로 평가량 (PM_i^n)의 합으로 경로 평가량을 계산하고 이 값 중에서 가장 작은 값을 새로운 경로 평가량 (PM_i^n)으로 선택하고 상태를 업데이트한다. 각 상태에서의 입력 심벌과 선택되는 경로는 그림 5와 같다. 만약 상태가 S0의 경우, 입력 경로는 S0, S2, S4, S6가 되고 입력 심벌은 D0, D2, D4, D6가 된다. 그래서 계산되는 경로 평가량은 아래와 같다.

$$PM_0^n = \min(PM_0^n + BM_0, PM_2^n + BM_2, PM_4^n + BM_4, PM_6^n + BM_6) \quad (2)$$

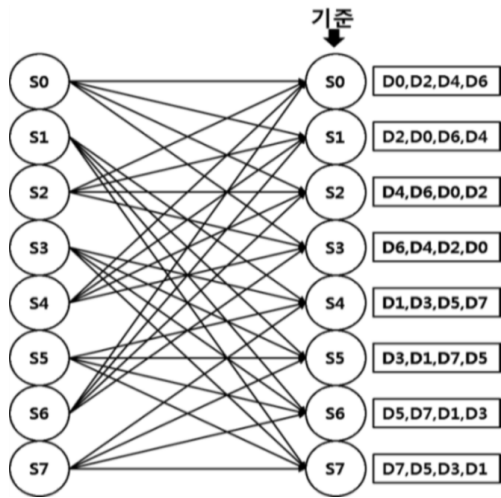


그림 5. 각 상태별 입력 경로와 입력 심벌
Fig. 5 Input path and symbol for each state

이때 사용되는 경로 선택기의 구조는 그림 6과 같다. S0의 경우를 보게 되면, 첫 번째 ACS (Adder-Compare-Select)의 입력에는 이전 경로에서 값이 없고 입력 심벌은 D0, D2, D4, D6가 된다. 첫 번째 ACS는 입력받은 심벌의 가장 작은 가지 평가량을 PM_0^n 로 선택해서 두 번째 ACS단의 입력 값의 PM_0^n 로 사용한다.

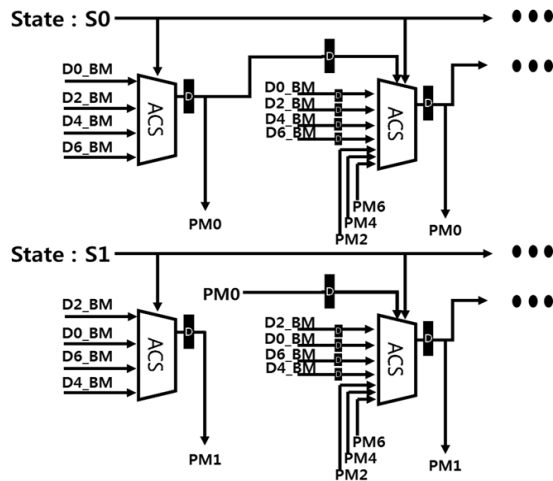


그림 6. PM0와 PM1의 PM 블록도
Fig. 6 PM block diagram of PM0 and PM1

두 번째 ACS단의 입력부터는 이전 경로 값으로 첫 번째 ACS단에서의 S0, S2, S4, S6에서 생성된 $PM_0^n, PM_2^n, PM_4^n, PM_6^n$ 가 되고 입력 심벌은 D0, D2, D4, D6의 BM_0, BM_2, BM_4, BM_6 가 된다. 두 번째 ACS는 입력받은 이전 경로의 PM_i^n 값과 입력 심벌의 BM_i 를 식 (2)와 같이 더한 다음에 가장 작은 값을 선택해서 다음 ACS단에 입력으로 보내준다.

그림 7은 구현한 PM 블록의 RTL (Register Transistor Logic) 회로도를 나타낸다. BM 블록에서 생성된 D_i 의 가지 평가량 (16 bit)을 입력받아 8개의 경로 평가량 (16 bit)을 생성한다. 8개의 상태에 따른 7단의 ACS로 구성되어 있으며, ACS단의 단 수는 역추적깊이 (Trace-back depth)를 나타낸다.

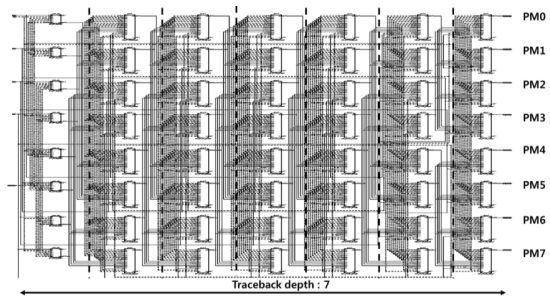


그림 7. PM RTL 회로도
Fig. 7 PM RTL Schematic

일반적으로 비터비 디코더의 역추적깊이는 구축장의 5~10 배의 길이가 필요하다[7]. 그러므로 구축장의 크기가 4인 1000Base-T는 이론상 20 이상의 역추적깊이가 필요하다. 그러나 하드웨어 구현상 역추적깊이가 증가하는 것은 비터비 알고리즘을 수행하는데 긴 시간 지연과 많은 메모리를 요구한다. 실제로 하드웨어를 구현한 한 논문[19]에서는 역추적깊이를 12로 구현하였지만, 본 논문에서는 하드웨어 로직 소모를 최소화하고 전력 소모를 줄이기 위해 역추적깊이를 7로 하였고, 그 대신 2조의 오류를 수정하기 위해 PC 블록에서 2개의 경로 선택기를 사용한다.

3.3. PC 블록

PC 블록에서는 PM 블록에서 생성한 최종단의 출력 PM0~PM7을 가지고 생존 경로를 선택하게 된다. 역추

적깊이를 7로 구성한 비터비 디코더에서 정확한 생존 경로를 선택하기 위해 그림 8과 같이 두 개의 경로 선택기와 추적 포인트 생성기를 가지고 경로를 판단하게 된다. 경로 선택기는 4개의 값을 비교하는 CS 블록 두 개와 2개의 값을 비교하는 CS 블록 하나로 구성되어 있다. 그리고 추적 포인트 생성기는 경로 검사기 (Path Checker)와 MUX로 구성되어 있다.

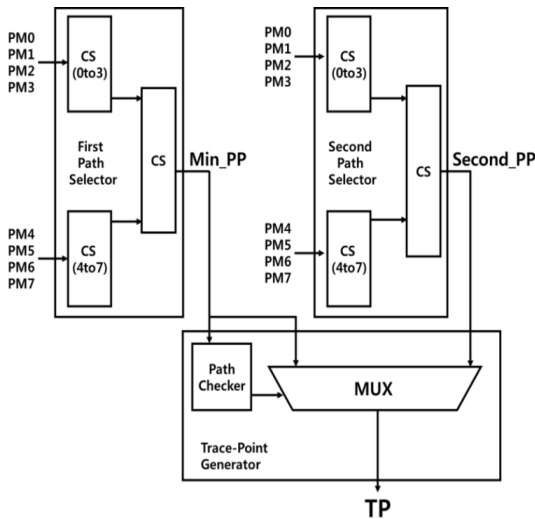


그림 8. 경로 제어 블록도
Fig. 8 Path control block diagram

제 1 및 제 2 경로 선택기는 PM 블록에서 계산된 PM0~PM7값을 입력받아 가장 작은 경로 평가량의 경로인 3 bit의 Min_PP (Path Point)와 두 번째로 작은 경로 평가량의 경로인 3 bit의 Second_PP를 생성한다. 이렇게 생성된 제 1 경로 선택기의 경로 포인트로 역추적 생성기의 경로 검사기에서 상태 천이 가능성을 검사한다. MUX는 제 1 경로 선택기의 경로가 이상이 없다면 제 1 경로 선택기의 경로를 선택해서 3 bit의 추적 포인트 (Trace Point, TP)를 생성한다. 만약 경로 검사기에서 천이 불가능한 경로가 판단되면 MUX는 제 2 경로 선택기의 경로로 대체하여 TP를 생성한다. 생성된 TP는 TBM 블록에서 역추적을 하는데 사용된다.

3.4. TBM 블록

TBM 블록은 그림 9와 같이 BM 블록에서 생성한 심

벌 (12 bit)와 PC 블록에서 생성한 TP (3 bit)를 입력받아 그림 5의 상태를 역으로 추적한다. 그리고 저장해둔 각 그룹의 심벌 (12 bit)의 정보 중에서 생존한 경로에 맞는 심벌 하나를 최종적으로 출력한다. 12 bit의 출력은 상위부터 3 bit마다 An, Bn, Cn, Dn 심벌의 정보를 가지고 있다.

예를 들어, TBM 블록의 출력이 ‘110001101010’라면 An(-2), Bn(1), Cn(-1), Dn(2)으로 복호하면 된다. BM 블록에서 0(000), 1(001), 2(100), -1(101), -2(110)의 3 bit 심벌로 대응시켜 4개의 신호를 12 bit의 심벌 정보로 만들어 주기 때문이다. TBM 블록은 PC 블록의 TP값이 계산될 때까지 BM 블록에서 생성된 모든 그룹의 심벌 값들을 가지고 있어야 한다. 그러므로 역추적 깊이가 깊어지면 TBM 블록의 메모리도 많이 필요하게 된다.

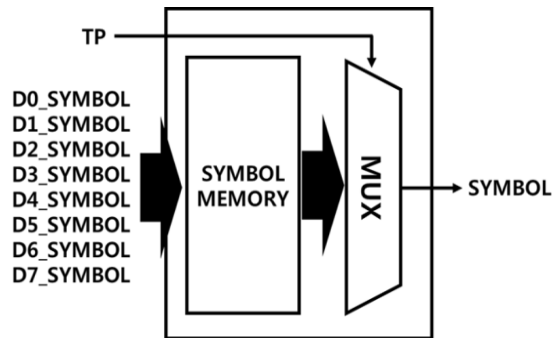


그림 9. 역추적 메모리 블록도
Fig. 9 Trace-back memory block diagram

IV. 구현결과 검증

본 논문에서는 비터비 알고리즘을 HDL로 설계 하였으며, 합성 툴은 Xilinx의 ISE (Integrated Software Environment)를 사용하였다. 지금까지 기술한 내용을 구현하고 검증하는 절차는 다음과 같다.

- 1000Base-T규격에 맞는 4조의 임의의 데이터를 생성한다.
- 생성된 데이터를 SNR이 20 dB인 AWGN 채널에 통과시킨 후, 1조와 2조의 오류를 고의로 발생시켜 테

- 스트 데이터를 생성한다.
- 생성한 테스트 데이터를 가지고 ISim(ISE Simulator)에서 기능 검증을 진행한다.
- 기능 검증 후, 하드웨어 검증을 위해 생성한 데이터를 FPGA의 ROM에 저장한다.
- FPGA의 DCM (Digital Clock Manager)모듈을 이용해서 디코더가 125MHz로 동작하도록 한다.
- 논리분석기를 이용해서 FPGA 하드웨어의 동작을 분석하고 기능 검증 결과와 비교한다.

표 2는 본 논문에서 설계하려는 비터비 디코더의 규격을 표로 나타낸 것이다. 설계된 비터비 디코더는 1000Base-T 시스템에 동작하고 125 Mhz이상의 시스템에서 동작하여야 한다. 입력 심벌은 8 bit 신호를 4조의 선에 동시에 보내는 4D-PAM5이고, 입력 신호의 BER은 20 dB이상으로 한다. 역추적깊이는 12 이하로 2 조의 오류를 정정할 수 있어야 한다.

표 2. 비터비 디코더의 설계 사양
Table. 2 Design specification of Viterbi decoder

Operating System	1000Base-T
Operating Clock	≥ 125 Mhz
input symbol coding	4D-PAM5
trace-back depth	≤ 12 state
BER of input signal	≥ 20dB
error correction function	2 pair error

그림 10은 합성 결과로 최대 동작 주파수가 249.919 MHz로 비터비 디코더의 필요 동작 속도인 125 MHz이상에서 동작이 가능한 것을 확인하였다.

표 3은 1000Base-T규격에 맞게 생성한 1024개의 데이터 중 20개의 원본 데이터이다. 표 2에서의 {96, 48, 0, -48, -96}는 전압레벨 {2, 1, 0, -1, -2,}를 나타내는 8 bit 신호이다. 예를 들어 첫 번째 클락 n=1에서 {0, -96, 0, 96} 값은 송신측에서 {0, -2, 0, 2}의 전압레벨을 4조의 선로 An, Bn, Cn, Dn으로 전송했다고 볼 수 있다.

```

Device utilization summary:
-----
Selected Device : 6s1x150fpg676-3

Slice Logic Utilization:
Number of Slice Registers:      68626 out of 184304  37%
Number of Slice LUTs:          44549 out of 92152   48%
Number used as Logic:          44549 out of 92152   48%

Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 77927
Number with an unused Flip Flop:  9301 out of 77927  11%
Number with an unused LUT:        33378 out of 77927  42%
Number of fully used LUT-FF pairs: 35248 out of 77927  45%
Number of unique control sets:     170

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%
    
```

(a)

```

Timing Summary:
-----
Speed Grade: -3

Minimum period: 4.001ns (Maximum Frequency: 249.919MHz)
Minimum input arrival time before clock: 7.858ns
    
```

(b)

그림 10. 합성 결과 (a) 장치 활용 요약 (b) 타이밍 요약
Fig. 10 Synthesis result (a) device utilization summary (b) timing summary

표 4은 원본 데이터에 오류를 삽입한 테스트 데이터의 일부이다. 송신측의 n=1에서 {0, -2, 0, 2}의 신호가 AWGN 채널을 통과 후, AD컨버터에 의해 8 bit 신호로 변환되어 {-3, -95, 17, 112}의 값이 된 것이다. 테스트 데이터 중에 n=11의 경우, 원래 D2 그룹의 데이터가 전 송되었지만 Cn 조에 오류가 삽입되어 D1 그룹이 되었다. 그리고 n=19의 경우, 원래 D3 그룹의 데이터지만 An과 Cn 조에 오류가 삽입되어 D5 그룹이 되었다. 이렇게 고의로 오류를 삽입한 테스트 데이터를 구현한 비터비 디코더의 입력 값으로 사용해서 ISim에서 기능 검증을 수행하였다.

그림 11의 ‘din_an, din_bn, din_cn, din_dn’은 ISim에서 비터비 디코더에 들어오는 테스트 데이터를 나타낸다. n=1부터 n=10까지 테스트 데이터가 비터비 디코더의 입력 값으로 들어가는 것을 확인 할 수 있다.

표 3. 원본 데이터

Table. 3 Original data

n	An	Bn	Cn	Dn	심벌그룹	
1	0	-96	0	96	YYYY	D0
2	96	96	96	-96	YYYY	D0
3	-96	0	0	0	YYYY	D0
4	0	0	0	0	YYYY	D0
5	-96	-96	96	-96	YYYY	D0
6	96	0	-96	-96	YYYY	D0
7	-96	96	0	0	YYYY	D0
8	0	96	0	96	YYYY	D0
9	96	-96	96	-96	YYYY	D0
10	-96	-96	96	-96	YYYY	D0
11	96	0	48	48	YYXX	D2
12	96	96	96	-48	YYXX	D1
13	-48	96	96	-48	XYYX	D4
14	96	-96	96	96	YYYY	D0
15	48	96	-48	96	XYYX	D6
16	96	-48	96	96	YXYY	D7
17	-96	-96	96	-96	YYYY	D0
18	48	-96	-96	96	XYYX	D5
19	-96	-96	-48	96	YXYY	D3
20	-48	96	96	-48	XYYX	D4

표 4. 오류가 삽입된 테스트 데이터

Table. 4 Test data with error inserted

n	An	Bn	Cn	Dn	심벌그룹	
1	-3	-95	17	112	YYYY	D0
2	102	111	87	-104	YYYY	D0
3	-90	3	-2	-12	YYYY	D0
4	-13	-7	4	-4	YYYY	D0
5	-91	-100	99	-85	YYYY	D0
6	110	-9	-108	-89	YYYY	D0
7	-103	77	3	-1	YYYY	D0
8	-2	102	-13	91	YYYY	D0
9	102	-99	90	-106	YYYY	D0
10	-102	-91	93	-98	YYYY	D0
11	101	8	83	57	YYXX	D1
12	106	104	94	-34	YYXX	D1
13	-46	94	84	-51	XYYX	D4
14	86	-101	89	99	YYYY	D0
15	28	105	-35	91	XYYX	D6
16	95	-52	83	90	YXYY	D7
17	-114	-103	102	-88	YYYY	D0
18	49	-88	-94	99	XYYX	D5
19	-70	-93	-75	94	XYYX	D5
20	-64	103	105	-44	XYYX	D4

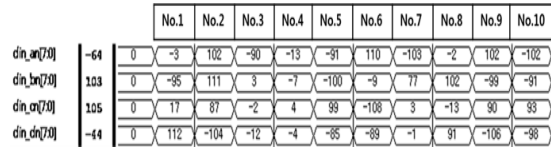


그림 11. ISim에서 입력된 테스트 데이터 파형
Fig. 11 Input test data wave in ISim

그림 12는 기능 검증에서 비터비 디코더의 오류 복구 결과를 보여준다. 첫 번째 입력이 들어가서 비터비 디코더에 의해 복원된 신호가 나오는데 총 74 클럭이 소요 되었다. 1조 오류로 표3에서 n=11의 {101, 8, 83, 57} 으로 들어온 D1 그룹 데이터가 표2에서 n=11 {96, 0, 48, 48}의 원본 D2 그룹 데이터와 같게 복원되었고, 2 조 오류로 표 3에서 n=19 {-70, -93, -75, 94}으로 들어온 D5 그룹 데이터가 표 2에서 n=19 {-96, -96, -48, 96}의 원본 D3 그룹 데이터와 같게 복원된 것을 확인할 수 있다.

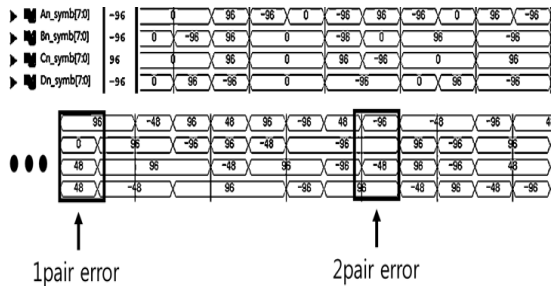


그림 12. ISim에서 1조와 2조에서 발생한 오류를 복구하는 과정
Fig. 12 Error recovery occurred 1 pair and 2 pairs in ISim

마지막으로 그림 13은 구현한 디코더를 하드웨어 검증 을 위해 FPGA에 기능 검증에서와 같은 테스트 데이터를 입력하여 비터비 디코더의 성능을 검증한 것이다. 이때, 사용된 FPGA 모델은 SPARTAN6이고 125 MHz의 클럭을 공급하였다. 이 그림의 ‘An_symb, Bn_symb, Cn_symb, Dn_symb’는 복원된 4 조의 데이터 값을 나타내는 것이다. 하드웨어 검증에서도 기능 검증 결과와 같이 1조 오류와 2조 오류를 원본 데이터로 복원한 것을 확인 하였다. 따라서 구현한 디코더가 1000Base-T에서 보내주는 데이터를 하드웨어에서 올바르게 오류 정정하고 요구되는 속도인 125 MHz에서 정상 작동하는

것을 확인하였다.

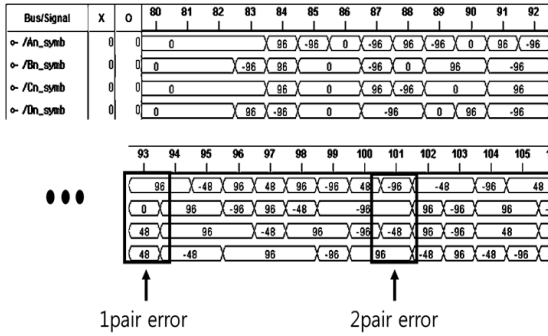


그림 13. 논리분석기를 통해서 확인한 출력 결과 파형
 Fig. 13 Output result wave by logic-analyzer

V. 결론

본 논문에서는 1000BASE-T에 적용되는 TCM을 위한 디코더를 비터비 알고리즘을 이용하여 구현하였다. 이 디코더는 크게 BM 블록, PM 블록, PC 블록 그리고 TBM 블록으로 구성되었다. 구현한 디코더를 ISim을 이용하여 기능 검증하고 FPGA를 이용하여 구현하고 동작을 검증하였다. 1조는 물론 2조에 걸쳐 발생한 오류에 대한 정정 능력을 검증하기 위하여, 임의의 오류를 삽입하여 테스트 데이터를 생성하고 디코더의 입력 값으로 사용하였다. 그리고 논리분석기를 통해 125 MHz로 동작하는 하드웨어의 출력을 확인하였다. 그 결과, 실제 하드웨어에서 정상적으로 목표했던 오류를 정정하고 125 MHz 이상에서 동작하는 것을 확인 할 수 있었다.

1000BASE-T는 캠퍼스, 빌딩, 그리고 다양한 사업장에서 고속의 근거리 망을 구축하는데 사용될 뿐만 아니라, UHD TV와 같은 고속의 서비스를 수용하기 위하여 올해부터 100 메가급의 LAN방식을 몰아내고 가입자 망의 중추적인 대안으로 자리 매김할 것으로 보인다. 본 논문에서 사용된 기법은 해당 칩을 개발하는 업체나 유사한 통신 칩을 설계하는데 도움이 될 것으로 판단된다.

감사의 글

본 연구는 금오공과대학교 교수연구년제에 의하여 연구된 실적물입니다.

REFERENCES

- [1] Ministry of Science, ICT and Future Planning, Accelerating the Deployment of Giga-Internet. [Internet Press release]. 2013. 5. 3. Available: http://www.msip.go.kr/www/brd/m_211/view.do?seq=89
- [2] IEEE Std. 802.3 Section 3, Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) access method and physical layer specifications, IEEE, Piscataway, N.J., 2008.
- [3] G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets," in *IEEE communication magazine*, vol. 25, no. 2, pp. 5-21, Feb. 1987.
- [4] G. Ungerboeck, "Channel coding with multilevel/ phase signals," *IEEE transactions on information theory*, vol. IT-28, no. 1, pp. 55-67. January 1982.
- [5] G. D. Forney, Jr., "The Viterbi Algorithm," in *Proceedings of the IEEE*, vol. 61, pp. 268-278, March 1973.
- [6] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE transactions on information theory*, vol. 15, pp. 177-179, Jan. 1969.
- [7] T. K. Moon, *Error Correction Coding*, Wiley, pp. 482, 2005.
- [8] Rami A. Abdallah, Naresh R. Shanbhag, D. C. Hu, W. Wei, W. C. Lin, H. Lin, "Error-resilient low-power Viterbi decoder architectures," *IEEE transactions on signal processing*, vol. 57, pp. 4906-4917, 2009.
- [9] W. S. Yu, "Cost-efficient implementation of a parallel processing soft-decision Viterbi decoder with a multi-stage pipelined ACS," M.S. Thesis, Sejong University, Korea, 2010.
- [10] T. Y. Kelvin, and Lai, "A high-speed low-power pipelined viterbi decoder: Breaking the ACS-bottleneck," *In Green Circuits and Systems(ICGCS) 2010 International Conference on*, Shanghai, pp. 334-337, 2010.

- [11] B. S. Seo, J. M. Kim, and H. S. Kim, "Parallel structure of Viterbi decoder for high performance of PRML signal," *Journal of The Korean Institute of Electrical Engineers*, vol. 58, no. 4, pp. 623-626, 2009.
- [12] Y. C. Tang, D. C. Hu, W. Wei, W. C. Lin, and H. Lin, "A memory-efficient architecture for low latency Viterbi decoders," *In VLSI Design, Automation and Test, 2009. VLSI-DAT'09. International Symposium on. IEEE, Hsinchu*, pp. 335-338, 2009.
- [13] S. J. Kim, and K. S. Cho, "Design of high-performance Viterbi decoder Circuit by Efficient Management of Path Metric Data" *Journal of IEIE*, vol. 47, no. 7, pp. 44-51, 2010.
- [14] J. H. Seo, J. C. Seo, and J. K. Kang, "High-Speed Viterbi Decoder Algorithms by Additional Error-Decision Circuitry" *In IEIE 2011 SoC Conference*, Korea, pp. 280-284, 2011.
- [15] Y. Chu, Y. S. Su, B. S. Lin, P. H. Cheng, and S. J. Chen, "A memoryless Viterbi decoder for LTE systems," *The 1st IEEE Global Conference on Consumer Electronics 2012*, Tokyo, pp. 643-644, 2012.
- [16] S. C. Lee, *A high-speed data transmission systems and methods using a cable*, KR Patent 10-1193646, to SKT, the Korean Intellectual Property Office. 2012.
- [17] H. J. Park et al, *500 Mbps transmission device using 2p line*, KR Patent 10-1244009, to KT, the Korean Intellectual Property Office. 2013.
- [18] Bernard Sklar, *Digital communication: fundamentals and application*, Prentice Hall, pp. 314-336, 1988.
- [19] H. R. Lee, "A Design of Viterbi Decoder for 1000Base-T," M.S. Thesis, Seoul National University, Korea, 2000.



정재우(Jae-Woo Jung)

2012년 2월 금오공과대학교 전자공학부 공학사
 2012년 7월 ~ 현재, 금오공과대학교 전자공학과 대학원 석사과정
 ※ 관심분야 : PON, 정보이론, 통신이론, 신호처리



정해(Hae Chung)

1987년 2월 한양대학교 전자통신공학과 공학사
 1991년 2월 한국과학기술원 전기및전자공학과 공학석사
 1996년 2월 한국과학기술원 전기및전자공학과 공학박사
 1993년 11월 ~ 1998년 7월 LG정보통신 (현 엘지전자) 선임연구원
 1998년 8월 ~ 현재 금오공과대학교 전자공학부 교수
 2004년 1월 ~ 2005년 1월 UTD 방문교수
 ※ 관심분야 : 가입자 액세스 네트워크, 스마트 디바이스, ICT융합