DVD 시스템에서 사용되는 변조 코드에 대한 간소화된 디코더

준회원 김 형 석*, 정회원 이 주 현**, 이 재 진***

Simplified Decoder of the Modulation Code for DVD System

Hyoung seok Kim*, Joohyun Lee**, Jaejin Lee*** ***Regular Members***

요 약

현재 DVD 시스템의 채널에서 디코더는 16비트 입력을 받아 8비트 데이터 심볼을 복원한다. 이러한 DVD용 변조 코드는 주 테이블(main table)과 부 테이블(sub table)로 이루어진 EFMplus 코드를 사용한다. 본 논문에서는 디코더 구현시 필요한 코드 테이블의 크기를 줄이기 위해 EFMplus 코드를 3개의 그룹으로 나눈 후 각 코드의 가지 수를 줄이는 방법을 제안하고 이 코드 테이블을 이용하여 디코더를 구현하였다. 이 방식은 기존의 테이블을 이용하여 디코딩할 때 필요한 코드 수 1376개를 750개로 약 46%정도 감소시켜 EFMplus 코드를 구현할 때 필요한 ROM의 크기를 약 2배 정도 줄였다.

Key Words : EFMPlus code, DVD, Optical recoding system

ABSTRACT

Currently, the decoder receives a 16 bit channel input and restores an 8 bit data symbol in the DVD system. Such modulation code of DVD is the EFMplus code, and it is composed of a main table and a sub table. For reducing the size of the code table, this paper divided the code table into 3 groups and we implemented the decoder using this new code table. After all, this method enables us to reduce the size of ROM as reducing the total number of code from 1376 to 750.

Ⅰ. 서 론

DVD(digital versatile disc)는 일반적으로 CD compact disc)에 비해 7배 이상의 저장 능력을 갖는 광 기록 매체이다. 이러한 광 기록 장치에서는 기록 밀도 증가로 인한 인접 심벌간의 간섭 현상의 심화와 DC 영역에 존재하는 미디어 잡음으로 인하여 검출 성능이 저하된다. 따라서, 고밀도 광 기록 장치에서는 채널에서 발생되는 오류를 줄이기 위해 변조코드를 사용하는데 이중에 '1'과 '1'사이의 '0'의 개수를 최소 d개, 최대 k개로 제한하는 런-길이 제한(run-length limited, RLL) 코드를 사용한다. 이러한 변조 코드 방법은 천이 간격을 제한함으로써 인접 심벌간의 간섭을 감소시키고 기록 밀도의 향상을 가져올 수 있다. DVD에서 사용되는 EFMplus 변조 코드에서는 (2, 10) RLL 조건을 사용하였고, 8비트의 입력에 대해 16비트를 출력함으로써 8/17의 코드 율을 갖는 EFM코드 보다 8/16의 향상된 코드 율을 나타낸다. 이러한 코드 율은 EFMplus코드가 EFM코드에 비하여 6~7% 정도의 저장 능력

* 동국대학교 전자공학과 통신 및 정보 저장 연구실 (qqjdaos@naver.com),
** 동국대학교 전자공학과 통신 및 정보 저장 연구실 (xmas@dongguk.edu),
*** 동국대학교 전자공학과 통신 및 정보 저장 연구실 (zlee@dgu.ac.kr)
논문번호 : KICS2005-01-035, 접수일자 : 2005년 1월 24일

을 향상시킨다[1].

이러한 (2, 10) RLL 조건을 만족하는 EFMplus 코드는 각각 4개의 상태(state)로 구성된 주 테이블(main table)과 부 테이블(sub table)로 이루어져 있으며, 다음 상태(next state) 값에 의해 연속되는 코드워드가 디코딩될 상태를 지정한다. 주 테입블의 코드와 부 테이블의 코드 중 1개를 선택하기 위해서 연속적인 런 디지털 합(running digital sum, RDS)을 구하여서 작은 값을 가지는 코드를 선택한다. 이렇게 변조된 코드는 DC 성분을 억압함으로써 미디어 노이즈의 영향을 감소시키고 검출 성능을 향상시킬 수 있다[2].

그러나, 인코딩된 코드워드는 디코딩 과정에서 16비트의 입력에 대해 8비트의 코드워드와 다음 상태를 나타내는 2비트를 출력해야 하므로 ROM을 이용한 코드워드의 1대 1 대응 방식을 이용할 경우 32Kbyte의 ROM을 필요로 한다. 이러한 대용량의 ROM은 하드웨어 구현시 크기를 증가시키는 문제점이 발생하게 된다. 최근에 개선된 디코딩 방법은 2개의 디코딩 코드 테이블을 구성하여 1개의 디코딩 코드 테이블은 다음 상태를 나타내는 코드 1비트를 붙이고 다른 디코딩 코드 테이블은 다음 상태를 나타내는 코드 1비트 없이 디코딩 할 수 있게 구성되어있다. 이렇게 개선된 디코딩 코드 테이블을 이용하면 2^{17}과 2^{16} 가지 수를 가지는 두개의 ROM을 구성할 수 있다[3].

그러나, 이러한 방법도 구현시 크기의 문제가 발생하기 때문에 본 논문에서는 이 문제점을 개선하기 위하여 2장에서 설명하게 될 새로운 디코더 코드 테이블을 구성하였고 베릴로그(verilog) 언어를 이용한 RTL(register transfer level) 레벨의 시뮬레이션을 통해 확인하였다.

Ⅱ. 새로운 디코딩 테이블 구성

EFMplus 변조 코드의 인코더는 표 1과 같이 입력 데이터 8비트에 16비트의 코드워드를 출력하고, 이 때 (d, k)조건을 만족시키고 DC성분을 억압할 수 있는 코드 구성이 될 수 있도록 다음 코드워드가 선택되어질 상태를 나타내는 2비트를 출력하게 된다. 코드 테이블이 표 1에서와 같이 4개의 상태로 구성되어 있기 때문에 다음 상태는 1, 2, 3, 4 중 하나의 상태를 나타낸다. 디코더 구성에 있어 표 1에서 구성된 코드의 디코딩 테이블을 이용하여 입력 데이터 16비트에 대해 8비트의 코드워드를 출력하게 되는데, 이때 이전 입력 데이터에 의해 생성된 다음 상태 정보와 연속되는 입력 코드워드를 동시에 만족하는 코드워드를 역 코드 테이블에서 찾아 출력하게 된다. 여기서, 입력 코드워드의 다음 상태 정보를 찾는 방법은 다음과 같다.

- 코드 워드 끝단의 '0'의 개수 : 0~1개
  → 다음 코드 워드의 위치 : 상태 1
  (=현재 코드워드의 다음 상태 정보)

- 코드 워드 끝단의 '0'의 개수 : 2~5개
  - 다음 입력 코드워드: $c_k \in \{x_{15}(MSB), x_{14}, \cdots, x_0\}$
    · $x_{15} + x_4 = 0$
    → 다음 코드 워드의 위치 : 상태 2
    · $x_{15} + x_4 = 1$
    → 다음 코드 워드의 위치 : 상태 3

- 코드 워드 끝단의 '0'의 개수 : 6~9개
  → 다음 코드 워드의 위치 : 상태 4

이와 같은 방법을 갖는 기존의 코드 테이블은 상태 정보와 입력정보를 동시에 알아야 하기 때문에 상태를 알려주는 2비트를 코드 테이블에서 코드 워드의 앞 단에 추가 해야만 한다. 하지만 이러한 방식으로 디코딩 테이블을 구성하게 되면 표 2과 같이 총 18비트의 입력 데이터가 필요하게 되고, 그 결과 디코딩 테이블은 2^{18}의 크기를 갖게 된다. 이러한 2^{18}의 디코딩 테이블을 구현하기 위해서는

표 1. EFMplus 코드의 테이블 일부(주 코드 테이블)

Input	State 1	State 2	State 3	State 4
	Codeword (Next state)	Codeword (Next state)	Codeword (Next state)	Codeword (Next state)
23	0010001000100000 (2)	0010001000100000 (2)	1000000001001000 (2)	1000000001001000 (2)
24	0010000100010000 (2)	0010000100010000 (2)	1000000001001000 (3)	1000000001001000 (3)
32	0001000000000100 (2)	0001000000000100 (2)	1000100100000010 (1)	1000100100000010 (1)
33	0001000000000100 (3)	0001000000000100 (3)	1000100010000001 (1)	1000100010000001 (1)
45	0000001000000001 (1)	0100010001000000 (4)	1000001001000010 (1)	0100010001000000 (4)

표 2. 기존의 EFMplus 코드에 대한 디코딩 테이블의 일부

Input codeword	Output code (decimal value)
101000100100000000	00000110 (6)
111000100100000000	00000110 (6)
001000010010000010	00101001 (9)
101000010010000010	00101001 (9)
011000001000100000	00101010 (10)
101000001000100000	00101010 (10)

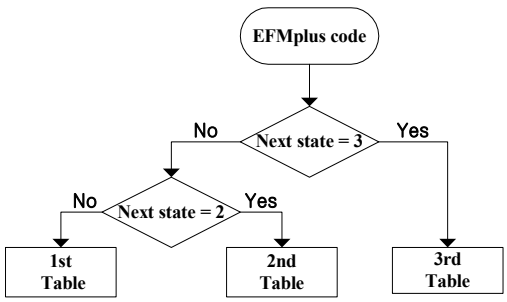

그림 1. 새로운 코드 테이블 구성.

32Kbyte의 ROM이 필요하게 되므로 새로운 디코딩 방법을 제안하여 ROM의 크기를 감소시킨다. 본 논문에서 제안된 디코딩 방법에 대한 전체적인 구성도는 그림 1과 같이 구성 되어있으며, 디코딩 테이블은 크게 표 3와 같이 3개의 코드 그룹으로 나눌 수 있다. 첫 번째 코드 그룹은 현재의 입력 코드워드가 연속되는 코드워드의 상태를 보지 않고 디코딩 할 수 있는 코드워드이다. 예를 들어, 표 1에서 상태 1에서의 입력 데이터 45에 대한 코드워드는 전체 코드 테이블에서 동일한 출력 코드워드가 존재하지 않고 입력 데이터 23에 대한 코드워드는 상태 1과 2에서만 존재하는 코드이기 때문에 디코딩 과정에서 별도로 연속되는 코드워드를 참조하지 않고 8비트의 디코딩된 코드워드를 출력할 수 있다. 이와 같은 코드워드의 개수가 전체 코드 테이블에서 336개가 존재한다. 두 번째 코드 그룹은 다음 상태가 2가 되는 코드워드이며, 첫 번째 그룹을 제외한 나머지 코드워드 중에서 조건을 만족하는 코드워드 207를 선택하였다. 마지막으로 세 번째 그룹은 현재의 입력 코드워드와 다음 상태가 3인 코드워드 중에서 첫 번째 코드 그룹을 제외한 나머지 코드워드 207를 선택하였다. 이렇게 3개의 그룹으로 나눌 수 있는 것은 EFMplus 코드 테이블에서 다음 상태가 1과 4인 코드워드의 경우 동일한 코드워드가 존재하더라도 하나의 입력 데이터만 가지고 있기 때문에 가능하다.

이렇게 분류한 3개 그룹의 코드워드를 비교하면 다음 상태가 2와 3인 경우 동일한 코드워드에서 서로 다른 입력 데이터를 가진다. 그러나, 첫 번째 그룹으로 선택된 코드워드는 다음 상태가 2와 3의 코드들과 같은 데이터 심볼을 나타내지만 동일한 코드는 존재하지 않는다. 이러한 구성은 디코딩할 때 디코더에서 잘못된 코드를 디코딩하지 않도록 해준다. 따라서 다음 상태 2의 코드 그룹과 첫 번째 그룹 코드을 합쳐서 디코더 주 코드 테이블로 만들 수 있다. 다음 상태 3의 코드들로 이루어진 디코더 부 코드 테이블을 구성할 수 있다. 이렇게 구성된 주 테이블은 디코더에 필요한 543개의 코드를 가지고 부 테이블은 207개의 코드를 가진다. 따라서 입력 코드 16비트로 이루어진 2개의 코드 테이블이 필요하기 때문에 2^{17}의 디코딩 테이블을 가지게 된다. 제안된 코드는 표 4와 같이 기존의 디코더 ROM에서 사용되는 코드의 수 262144개를 131072개로 감소시킴으로써 디코더를 구현함에 있어 ROM의 크기를 32Kbyte에서 16Kbyte로 감소 시켰다.

표 4. 기존 방법과 제안된 디코더의 ROM 구성하기 위한 코드 수

	ROM 구성에 대한 코드 수
기존 방법	262144 (32Kbyte)
제안된 디코더	131072 (16Kbyte)

Ⅲ. Verilog HDL 구현

본 논문에서는 여러 가지 디지털 회로의 하드웨어 구현 방법 중 FPGA 소자를 통해 성능 을 검증

표 3. 제안된 디코딩 테이블

Decoded code	First table	Second table	Third table
00000110	1000100100000000	0000000000000000	0010000000100100
00101001	1000010010000010	0000000000000000	0010000100010000
00101010	0000000000000000	1000001000100000	0010001000100000

그림 2. 디코더 전체 입출력 신호

하기 위해서 베릴로그 HDL(Verilog hardware definition language) 언어를 이용하여 RTL(register transfer level) 코드로 구현하였다. 그림 2는 EFMplus 코드의 디코더에 대한 전체 입출력 관계를 나타낸 것이다. 입력 단에 3가지의 신호가 들어오게 되는데 입력 신호(Input signal)는 DVD에서 신호를 읽어오는지를 판별하기 위한 신호를 표시한 것이고, 리셋(Reset)은 초기 값을 설정해 주기 위한 신호이다. 입력 단에 마지막에 들어오는 직렬 입력(Serial input) 신호는 DVD채널을 통해 재생된 신호들이 1비트씩 들어오는 입력을 표시한 것이다. 출력 단에는 디코딩된 신호(Decoded signal)와 디코딩된 코드(Decoded code) 8비트 코드가 출력된다. 디코딩된 신호는 출력되어진 신호가 올바른지 판단하는 신호이고 디코딩된 코드는 앞에서 설명한 디코더 방법에 따른 결과 코드이다.

그림 3은 변형된 디코더의 전체 흐름도와 각 모듈 사의 입출력 관계를 나타내었다. 각 모듈에 대한 특성을 살펴보면 다음과 같다.

3.1 신호 복원부(Signal Restore module)

채널로부터 입력된 데이터는 NRZI(non-return-to-zero inverted) 방식으로 기록되어 있기 때문에 신호 복원부를 이용하여 NRZI로 변환하기 이전의 데이터 형태로 복원해야 한다. 따라서 신호 복원부에서는 그림 4처럼 D-플립플롭(D-FF)을 이용하여 이전 비트를 저장하고 이렇게 저장된 이전 비트를 현재 입력 비트와의 XOR를 통해 NRZI 이전의 데이터로 복원할 수 있다. 즉, 그림 4와 같이 신호 복원부는 3개의 입력을 받는다. 처음 입력은 신호 복원부에서 출력한 신호이고 두 번째 입력은 신호 복원부에서 출력한 신호와 직렬 입력을 XOR한 신호이다. 마지막 입력은 신호 복원부를 제어하는 입력 신호이다. 즉, 신호 복원부의 입력을 선택하는 것은 입력 신호이다. 입력 신호가 '1'을 나타내면 두 번째 입력을 선택하고 입력 신호가 '0'이면 첫 번째 입력을 선택해서 신호를 복원한다. 이렇게 복원된 신호는 직·병렬 변환기를 통하여 EFMPlus 코드의 특성인 16비트 병렬로 신호를 출력하게 된다.

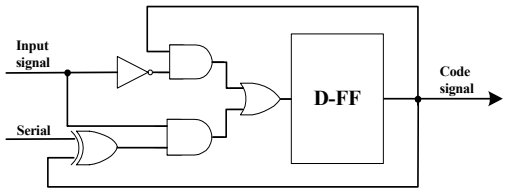

그림 4. 신호 복원부 구성도

3.2 제어부(Control module)

그림 5과 같이 구성된 제어부는 클럭과 입력 신호를 입력으로 받아서 디코더 신호와 카운터 신호를 발생시키는 곳이다. 카운터 신호는 카운터를 사용하여 16클럭마다 신호 '1'을 발생시킨다. 즉, 그림 5의 카운터부에서 1에서 16까지 카운팅한 4비트를 입력으로 첫 번째 조합부(Match 1)에서 1이될 때 마다 신호 '1'를 내보내고 그 밖의 부분에서는 신호 '0'을 출력한다.

또한, 제어부에서는 인코더에서 들어온 동기 코드(synchronization code)와 정보코드를 구분하게 하는

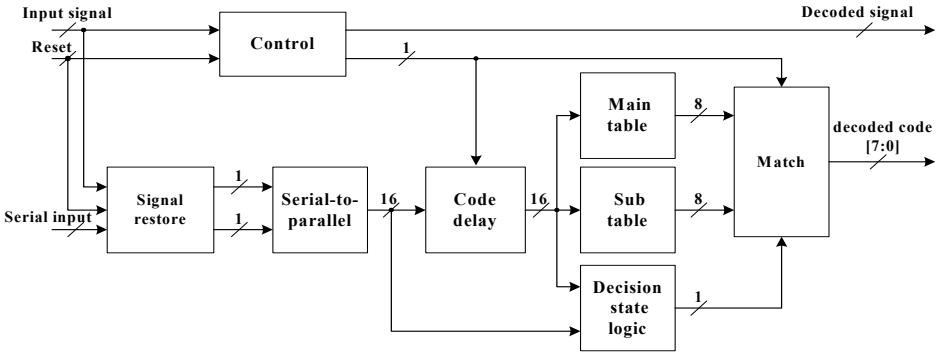

그림 3. 디코더 전체 구성 및 흐름도

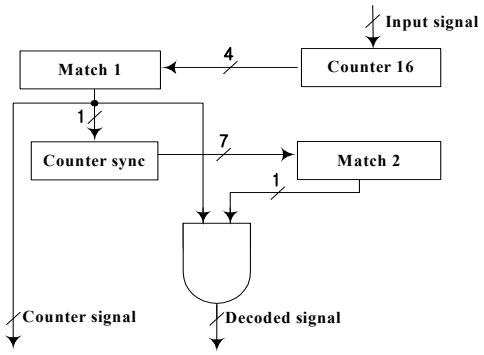

그림 5. 제어부 모듈 구성도

디코더 신호를 출력한다. DVD의 인코더는 앞단에 32 비트의 동기 코드와 16비트 정보 코드 91개를 연속적으로 출력한다. 이 때 동기 코드는 정보가 없기 때문에 제거할 필요가 있다. 첫 번째 조합부에서 신호 '1'을 내보내면 동기 카운터부(Counter sync)에서 카운팅을 하게 된다. 이렇게 카운팅 된 값은 1에서 93까지 7비트를 출력한다. 카운팅 값을 입력으로 받은 두 번째 조합부는 카운팅 값 중 1과 2에서 디코더 신호 '0'을 출력하여 동기 코드임을 나타내고, 나머지 카운팅 값들은 디코더 신호 '1'을 출력하여 정보 코드임을 나타낸다.

3.3 코드 선택부(Code selection module)

그림 6과 같이 구성되어 있는 코드 선택부는 새로운 디코딩된 코드 테이블을 이용하여 디코딩을 하기위해 기존과 다른 하드웨어를 구성하였다. 기존의 선택부는 그림 7과 같이 구성되어 있다. 즉, 기존의 디코딩 방법에 의해서 현재와 과거의 코드를 받아서 상태를 판별하고 이렇게 판별된 상태 2비트와 과거 코드 16비트가 합쳐져 ROM에 18비트 입력으로 들어간다. 이와 같이 구성된 18비트는 ROM에 1대 1 대응되어 그때의 8비트 디코딩된 코드를 출력한다. 그러나, 새로운 선택부는 그림 6에서와 같이 상태 결정부(Decision state logic module)에서 주 테이블 또는 부 테이블 코드를 나타내는 1비트 신호를 조합부(Match module)에 보낸다. 신호의 판별 방법은 2장에서 설명한 새로운 디코딩 테이블의 특성과 디코더 규칙을 이용하였다. 즉, 다음 상태가 3이면 신호를 '0', 나머지의 다음 상태는 신호를 '1'로 보내도록 구성한다.

또한, 조합부에서는 ROM에서 과거의 코드 16비트를 받아서 8비트 출력한 신호를 입력으로 받는다. 신호의 판별 방법에 따라 신호가 '1'이면 주 테이블

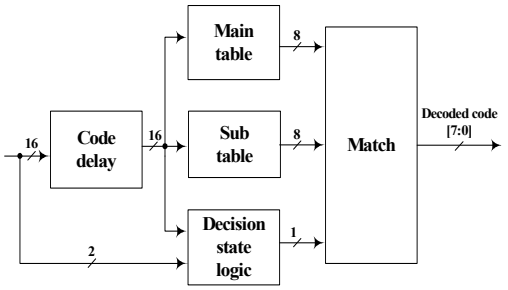

그림 6. 기존 방법의 선택부 구성도

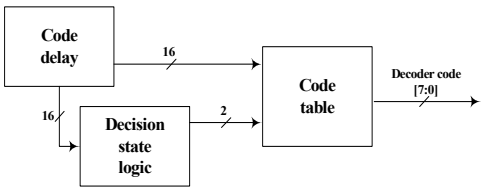

그림 7. 기존 방법의 스펙 선택부 구성도

에서 보낸 코드를 선택하고 '0'이면 부 테이블에서 보낸 코드를 선택한다. 이러한 기능을 하는 조합 부는 ROM의 코드와 상태 결정부에서 오는 신호가 같은 시간 도착해야 정확한 디코딩을 할 수 있다는 단점을 가진다. 조합부에서 이러한 단점을 나타내는 것은 광 기록 장치에서 오는 직렬신호를 병렬로 바꾸어주면서 생기는 시간 지연에서 야기되는 것이다. 이러한 디코딩 오류를 방지하기 위해서 제어 모듈에서 16클럭마다 신호를 주어서 조합부를 정확히 제어해야 한다. 즉, 제어 신호에서 '1'의 신호가 입력되면 그때의 출력 코드를 결정하고 '0'의 신호가 입력되면 그전의 출력 코드를 유지한다.

또한, 그림 6과 같이 새로운 코드로 구성된 조합부는 2개의 ROM을 사용하므로 주 테이블로 구성된 ROM과 부 코드 테이블로 구성된 ROM에서 서로 타이밍 지연이 발생될 수 있다.

3.4 Verilog 코딩 결과

베릴로그 코딩에 앞서 C언어로 EFMPlus 모델링을 먼저 수행하였다. C언어로 모델링한 프로그램은 EFMPlus의 구조별로 출력값을 비교하여 검증하였다. 이 검증된 C언어 모델링 프로그램을 수행하여 타이밍이 포함되어 있지 않는 입력 및 출력 데이터를 출력한다. 이렇게 주어진 입력에 의해 설계 모듈을 동작시켜 결과를 출력하며, 이 출력 데이터를 C언어 모델링 프로그램에 의해 출력된 데이터와 비교하여 정상적인 동작을 검증하는 방식으로 테스트를 수행하였다. 이렇게 검증된 베릴로그 코딩의 결

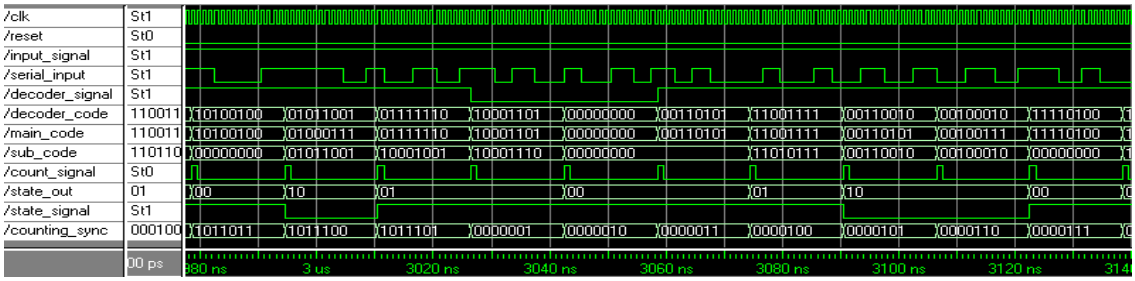

그림 8. EFMPlus 디코더의 타이밍도

과 타이밍도는 그림 8에 나타내었다. 그림 8의 타이밍도는 입력 신호가 '1' 을 나타내고 있고 직렬 입력이 들어온다. 이때, 다음 상태 값이 '10'을 제외한 나머지 상태에서는 '1'값을 출력하여 주 테이블을 선택한다. 그 후, 동기 코드와 정보 코드를 분석하여 디코더 신호는 '0'과 '1'을 나타낸다. 즉, 정확하게 디코딩된 코드는 디코더 신호가 '1'을 나타낼 때 출력된 코드이다.

Ⅳ. 결 론

본 논문에서는 기존의 EFMplus 변조코드에 대한 디코더 칩 구현시 문제점이 되어왔던 ROM 크기의 문제를 해결하기 위하여 새로운 코드 테이블 구성하였다. 새로 구성한 디코딩 테이블은 기존의 코드 테이블의 1367개의 코드 수를 750개로 46% 감소시켰다. 또한, 새로운 코드 테이블은 기존의 EFMplus 코드 테이블 특성을 이용하여 상태를 나타내는 2비트를 제거하고 다음 코드의 상태만을 파악하여 1비트의 신호로 디코딩하는 방식을 채택하였다. 이 방식을 채택함으로 기존의 ROM을 구성하는 코드 2^{18}(=32Kbyte)을 2^{16}(=8Kbyte)의 코드로 구성되는 2개의 ROM을 사용하여 전체 코드워드 개수를 2^{17}(=16Kbyte)로 2배 감소시킬 수 있다.

참 고 문 헌

〔1〕 K. A. S. Immink "EFMPlus : The coding format of the multimedia compact disc", *IEEE Trans. Consumer Electronics*, vol. 3, pp. 491-497, Aug. 1995.

〔2〕 *DVD Specification for Rewritable Disc/Part 1. Physical Specification Version 1.0*, pp. 143, pt. 1. Japan, July, 1997.

〔3〕 P. T. Tran, "Demodulation of DVD codeword using dependency-Sorted tables for duplicate /dependent and unique /non-dependent mapping", U. S. Patent 6, 195, 778, Feb. 27, 2001.

김 형 석 (Hyoung seok Kim) 준회원

2003년 2월 동국대학교 전자공학과 학사 졸업

2003년 9월~현재 동국대학교 전자공학과 석사과정

<관심분야> 변조코드, 검출 기법 연구

이 주 현 (Joohyun Lee) 정회원

한국통신학회 논문지 제29권 8C호 참조

이 재 진 (Jaejin Lee) 정회원

한국통신학회 논문지 제29권 1C호 참조