

CATV 하향 스트림 적용 시스템에서 동기 검출 방안 및 FPGA 설계

김민혁*, 박태두*, 김남수*, 김철승*, 정지원*

*한국해양대학교 전파공학과

e-mail:squru@hotmail.com

FPGA Design and Sync-Word Detection of CATV Down-Link Stream Transmission System

Min-Hyuk Kim*, Tae-Doo Park*, Nam-Soo Kim*, Chul-Seung Kim*, Ji-Won Jung*

*Dept of Radio Communication Engineering, Korea Maritime University

요 약

본 논문은 ITU-T 권고안 J-38 부록 B에 명시된 전송방식의 분석 및 시뮬레이션을 토대로 성능을 분석 하였으며 FPGA 구현시 야기되는 문제점을 나타내고, 해결방안을 제시하였다. 구현상의 문제점으로는 크게 두 가지로 분류되는데, 첫째로 다양한 부호화 방식과 변조방식 그리고 심볼 단위 및 비트 단위의 처리로 인해 많은 클럭수를 요구하는데 본 논문에서는 읽기/쓰기 메모리를 이용하여 필요한 클럭수를 줄였다. 둘째로는 평처링 부호화된 TCM 복호기에 평처링 패턴에 정확한 동기를 얻지 못하면 프레임 동기 심볼인 UW(Unique sync-Word)를 획득하지 못하여 모든 데이터가 에러 처리되기 때문에 본 논문에서는 평처링 패턴과 UW 심볼의 동기를 맞추는 알고리즘을 제시하였다. 이러한 알고리즘 분석 및 구현상의 문제점 해결을 토대로 본 논문에서는 ITU-T J38 annex B의 하향 스트림 채널 부호화 시스템을 VHDL 언어를 사용하여 FPGA 칩에 직접 구현하였다.

1. 서론

방송을 위한 통신시스템이나 데이터 전송을 위한 통신 시스템에서는 주로 연결부호가 사용된다. 연결부호는 두 가지 이상의 부호를 연결해 사용하여 높은 부호이득을 얻을 수 있어서 각광을 받고 있다. 연결 부호화 방식은 주로 리드-솔로몬(Reed-Solomon : RS) 부호와 컨벌루션 부호화 방식이 결합되는 방식인데, 이는 고차 변조 방식에서의 성능이 미비하여, 고차 변조 방식과 결합한 트렐리스 부호화 방식(TCM, Trellis Coded Modulation)이 주로 적용되고 있으며[1], 이 또한 케이블 방송에서 적용되고 있다. 본 논문은 케이블 망에서 적용되고 있는 채널 부호화 방식의 알고리즘을 분석 및 Visual C++ 언어를 이용하여 시뮬레이션을 하여 성능을 분석 하였으며, FPGA 구현시 문제점 제시 및 해결방안을 제시 하였다.

구현상의 문제점으로는 첫째로 케이블망의 오류 정정 방식의 부호화율에 따른 많은 수의 클럭이 필요로 되어진다. 하지만 많은 수의 클럭을 사용할 경우 구현시 클럭 간의 동기가 어긋나게 되고, 또한 발진 회로가 많이 사용되어 구현에 많은 어려움이 있다. 이에 본 논문에서는 두 개의 메모리를 이용하여 기본으로 제공되는 클럭 주파수를 분주 시켜 두 개의 클럭만으로 사용하여 시스템을 구현하였다. 둘째로는 송신단에서 첨가된 프레임 동기 심볼(UW)을 수신단에서 찾아야 한다. 프레임 동기 심볼을 찾지 못한다면 심볼 단위로 처리되는 데이터가 모두 어긋나

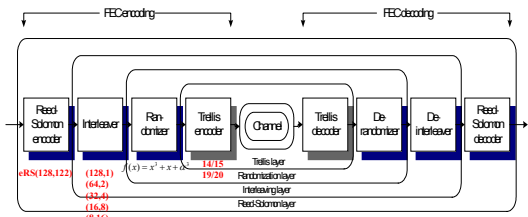
게 되어 대부분의 데이터는 오류로 인식되게 된다. 기존의 많이 쓰이는 프레임 동기 심볼 검출 방식은 비트 단위로 수신단에서 알고 있는 프레임 동기 심볼과 비교하여 일치되는 지점을 찾는 것이다. 하지만 이 방법은 많은 연산량과 그에 따른 속도 지연을 가져 올 수 있고, 구현시 가장 큰 문제가 되는 비트 단위로 연산하기 위한 클럭 역시 필요 되어진다. 이를 개선하는 방안으로 본 논문에서는 비트단위가 아닌 심볼 단위로 프레임 동기를 찾아내는 방법을 사용하였다. 이 방법은 복호단 앞에 5개의 심볼 레지스터를 사용하는 방법으로 비트단위의 클럭의 사용 없이도 쉽게 구현할 수 있다.

본 논문은 이러한 알고리즘을 이용하여 구현 시 문제점들을 해결 하였고, ITU-T J38 부록 B의 하향 스트림 채널 부호화 시스템을 VHDL 언어를 사용하여 PAR(Place And Route)레벨 까지 모두 타이밍 시뮬레이션 한 뒤FPGA 칩에 직접 구현하였다.

2. CATV 하향 스트림 시스템 모델

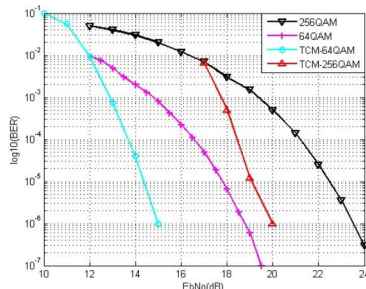
그림 1은 ITU-T_J83E_하향표준에 제시되어 있는 채널 부호화 시스템 모델이다. 부호화기는 확장된 RS(128,122), 컨벌루션 인터리버, 데이터 랜덤화기, TCM 부호기의 순서로 되어있고, 복호화기는 그 역순이다. 본 논문에서 구현한 시스템은 부호화율 122/128, 에러정정 능력 t=3인 확장된 RS 부호를 사용하였고, 128x1의 사이즈

를 가진 컨벌루션 인터리버, 데이터 랜덤화기, 그리고 두 가지 모드를 사용하는 TCM을 사용하였다[1][2][3][4].



(그림 1) CATV 하향 스트림 시스템 모델

그림 2는 비부호화 QAM과 64-QAM과 256-QAM 방식의 TCM 성능곡선이다.



(그림 2) 비부호화 QAM과 TCM의 성능 곡선

시뮬레이션에서 사용한 데이터는 비부호화 QAM과 TCM의 64-QAM과 256-QAM 모두 10⁶ 비트를 사용하였고, 채널 환경은 AWGN을 첨가하였다. 64-QAM과 256-QAM은 BER이 10⁻⁴에서 약 4dB 정도의 성능차가 있고, 비부호화 QAM과 TCM간에는 약 3dB 정도의 성능차가 있음을 알 수 있다.

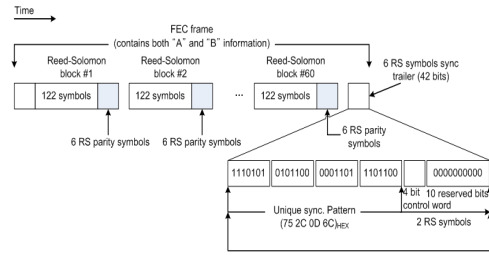
3. 구현상의 문제점 및 해결 방안

1) 프레임 동기 알고리즘

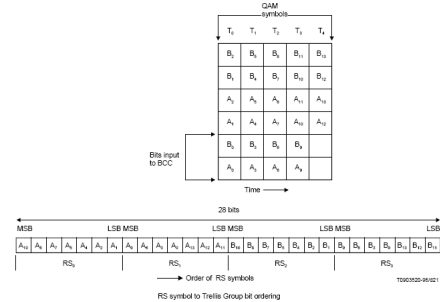
그림 3은 64-QAM 모드 일 때의 CATV 하향 스트림의 프레임 구조이다.

64-QAM의 경우 한 프레임의 데이터는 60개의 RS 블록과 42 비트의 프레임 동기 심볼로 구성되어 있다. 42 비트의 프레임 동기 심볼은 4개의 RS 심볼과 4비트의 인터리버 제어 심볼, 10 예비 데이터로 구성된다. 여기서 4개의 RS 심볼은 UW로 항상 일정하며, 이를 이용해 수신단에서 프레임 동기를 맞추게 된다. TCM 심볼과 RS 심볼을 구성하는 비트 수가 틀리므로 프레임 동기를 맞추지 못하면 수신단에서는 RS 심볼을 제대로 구성할 수 없다. 때문에 프레임 동기를 맞추는 것은 매우 중요하다.

그림 4는 RS 심볼을 TCM 심볼로 만들기 위한 데이터 구성을 나타낸다. 64-QAM의 경우 28bits의 데이터를 TCM 복호후 UW를 찾아 동기를 맞춘 후 RS 심볼단위로 디랜덤화기, 디인터리버, RS 복호기로 입력되는데 이때 UW를 정확하게 찾지 못하면 TCM 뒷단에서 모두 오류가 발생한다. 따라서 UW의 검출시 각 동기 비트의 매칭 정



(그림 3) 64-QAM모드의 CATV 하향 스트림의 데이터 구조



(그림 4) 64-QAM 데이터 구성 패턴

도, 즉, threshold에 따라 프레임 오류율을 그림 5와 같이 구하였다. 시뮬레이션 시 채널은 AWGN을 사용하였고,

(1)

여기서 T_i 는 데이터 동기 신호 전체의 길이 이고,

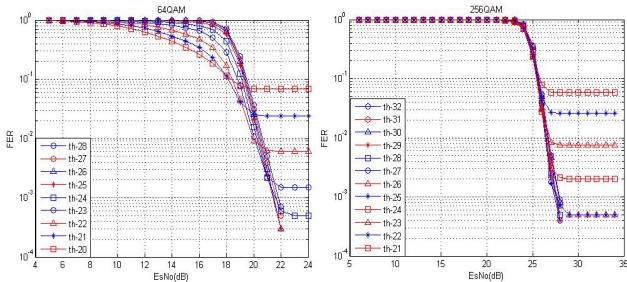
B_i 는 i 번째 동기 신호, A_i 는 i 번째 수신 신호를 나타낸다.

구현을 위한 프레임 동기 심볼 디택션에서 threshold에 따른 프레임 에러율을 고려했을 때 64-QAM에서는 th=24 이하부터 에러 플로어가 발생하며 256-QAM에서는 th=26 이하부터 에러 플로어가 발생함을 알 수 있다. 프레임 동기 심볼 detection에서 threshold는 63-QAM에서는 최소 th=25 이상을, 256-qam에서는 최소 th=27 이상을 지정해야 함을 알 수 있다. 이는 64-QAM의 경우 th=24 이하에서, 256-QAM의 경우 th=26 이하에서 동기 심볼을 제외한 데이터 영역에서 데이터 심볼을 UW로 인식함을 알 수 있다.

TCM 복호기 또한 trellis 상에서 5개의 QAM 심볼이 평치링 되기 때문에 수신 심볼이 평치링 패턴에 대한 동기가 정확하지 않으면 TCM 부호 자체가 오류가 존재해 정확한 동기 심볼을 찾을 수가 없다.

본 논문에서는 구현시 TCM decoder의 이전 단계에서 5개의 레지스터를 사용하였다. 그림 6은 사용한 레지스터를 나타낸 것이다.

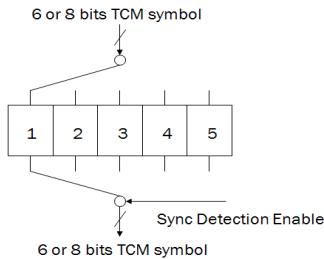
그림 6의 5개의 레지스터에 1번부터 순서대로 데이터가 저장된다. 저장된 데이터는 순서대로 TCM decoder로 출력된다.



(a) 64-QAM (b) 256-QAM
(그림 5) threshold에 따른 프레임 에러율

레지스터에서 출력된 데이터는 TCM 복호기에 의해 복호 되어진 후 동기 심볼 디텍터에 의해 판별되어지는데, 한 프레임이 모두 디텍팅 될 때 까지 데이터 동기 신호를 찾지 못하면 레지스터의 출력 순서를 하나 건너 뛰게 한다. 이러한 방법을 사용할 경우 최대 5 프레임 안에 데이터 구성 패턴에서의 첫 번째 심볼을 찾아 낼 수 있고, 비트 단위의 클럭을 사용하지 않고 데이터 동기 신호를 찾아낼 수 있다.

처음 디텍션이 성공한 후 신뢰를 가지기 위해 최소 3~5 프레임 동안 정확히 프레임이 끝나는 지점에서 프레임 동기 심볼이 계속 찾아지는 것을 확인하고, 디텍션되지 않는다면 위의 과정을 처음부터 다시 수행한다. 디텍션이 성공하면 다시 데이터 재구성을 통해 TCM 심볼을 RS 심볼로 변환시킨 후 출력한다.

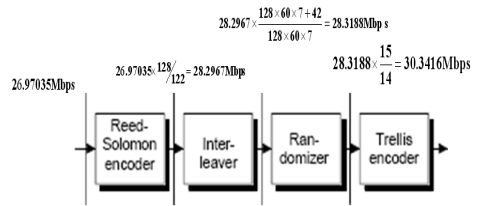


(그림 6) Unique Sync-word Detection을 위한 5개의 레지스터

2) 제한된 클럭을 사용한 시스템 구현 기법

연접 부호의 구현일 경우 사용된 부호화 기법에 따라 각각의 클럭이 필요하다. 그림 7은 CATV 하향 스트림 채널 부호 시스템에서 각 블록마다 필요한 클럭을 나타낸 것이다.

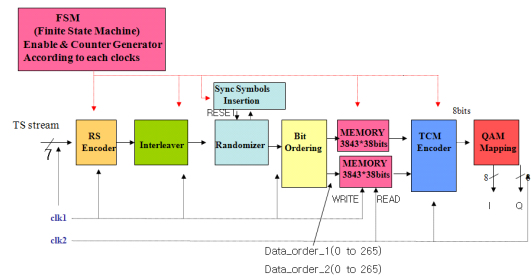
RS 부호와 TCM의 경우 부호화율이 적용이 되기 때문에 입력과 출력에 필요한 클럭이 틀려야 된다. 송신단에서는 입력보다 출력의 클럭이 빨라야 입출력에 대한 타이밍을 맞출 수 있다. 랜덤화기의 경우 출력단에서 데이터 프레임이 끝날 때 프레임 동기 심볼을 첨가하기 때문에 그에 대해 클럭이 틀려진다. 그림 7에는 각 블록에서 출력시 필요한 클럭 계산을 볼 수 있다. 수신단의 경우는 송신단과 반대로 입력의 클럭이 출력의 클럭보다 빨라야 함을 알 수 있다.



(그림 7) 각 블록의 필요한 클럭

실제 시스템 구현시 각 블록에 대한 클럭들을 맞춰서 적용시킬 수 없다. 그 이유는 각 클럭의 동기가 맞아야 하기 때문이다. 각 클럭간의 동기가 맞지 않으면 시스템은 오동작을 일으키게 된다. 기본적으로 제공되는 클럭과의 동기를 맞추는 가장 기본적인 방법은 분주를 이용하는 것이다. 기본 클럭을 분주를 시킬 수 있다면 클럭의 동기는 문제가 되지 않는다. 하지만 그림 6에서 알 수 있듯이 각 블록에서 필요한 클럭은 분주가 불가능하다. 그렇기 때문에 클럭이 제한되어 있으면 시스템을 구현하기 힘들다.

이러한 문제를 해결하기 위해 그림 8과 같이 두 개의 메모리를 사용하였다.



(그림 8) 구현시 시스템 구조

‘clk1’을 시스템의 기본 클럭이라 하면 ‘clk2’는 기본 클럭의 4분주시킨 클럭이다. 메모리의 위치는 RS 심볼과 TCM 심볼이 서로 변환되는 위치이다. RS 부호, 인터리버, 랜덤화기는 RS 심볼을 사용하고, TCM은 TCM 심볼을 사용한다.

메모리 사이즈가 3843x38인 이유는 64-QAM으로 동작 할 때 한 프레임이 데이터 구성 패턴에 정확히 떨어지지 않고, 두 프레임이 될 때 구성 패턴과 일치되기 때문에 메모리 하나 당 64-QAM 데이터 두 프레임을 저장할 수 있도록 하기 위해서이다.

‘clk1’이 ‘clk2’보다 4배 빠르기 때문에 ‘clk2’에 의한 동작이 끝날 때 까지 ‘clk1’이 동작 후 멈추게 하여 메모리의 읽기와 쓰기 동작을 진행 시키면 서로 동기가 일치하는 두 개의 클럭만을 사용하여 시스템을 구현할 수 있다.

4. FPGA 구현 결과

1) 타이밍 시뮬레이션 결과

그림 9는 수신단의 전체적인 타이밍 시뮬레이션이다. RS 심볼 데이터로 동작하는 부분에서 메모리 동작의 상태에 따라 연속적인 동작이 이루어 지지 않음을 볼 수 있고, TCM 심볼 데이터로 동작하는 부분은 데이터가 연

