

# CATV 하향 스트림 적용 시스템에서 동기 검출 방안 및 FPGA 설계

정 지 원\*

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

Ji-Won Jung\*

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 두가지로 분류되는데, 첫째로 다양한 부호화 방식과 변조방식 그리고 심볼 단위 및 비트 단위의 처리로 인해 많은 클럭수를 요구하는데 본 논문에서는 read/write 메모리를 이용하여 필요한 클럭수를 줄였다. 둘째로는 평치링 부호화된 TCM 복호기에 평치링 패턴에 정확한 동기를 얻지 못하면 프레임 동기 심볼인 UW(Unique sync-Word)를 획득하지 못한다. 따라서 본 논문에서는 평치링 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.

ABSTRACT

Cable modems typically are implemented by a forward error correction(FEC) scheme. The ITU-T Recommendation J-38 Annex B specifies using 64- and 256- quadrature amplitude modulation (QAM) and extended RS coding scheme. In implementing the cable modem, there are some problems to fabricate and fitting on FPGA chip. First, many clocks are needed in implementing cable modem because of different code rate and different modulation types. To reduce the number of clocks, we use the two memories, which are different clock speed for reading and writing data. Second, this system lost the bit-synchronization and frame-synchronization in decoder, the system recognize that all data is error. This paper solves the problems by using simple 5-stage registers and unique sync-word. Based on solutions for about problems, the cable modem is fabricated on FPGA chip name as Vertex II pro xc2vp30-5 by Xilinx, and we confirmed the effectiveness of the results.

**Keywords :** FPGA, FSM, QAM, TCM, Sync-word detection

---

\* 한국해양대학교 전자공학과 교수 (jwjung@hhu.ac.kr)

접수일자 : 2011년 11월 07일, 수정일자 : 2011년 11월 29일, 심사완료일자 : 2011년 12월 03일

## I. 서론

방송을 위한 통신시스템이나 데이터 전송을 위한 통신시스템에서는 주로 연결부호가 사용된다. 연결부호는 두 가지 이상의 부호를 연결해 사용하여 높은 부호이득을 얻을 수 있어서 각광을 받고 있다. 연결 부호화 방식은 주로 RS 부호와 convolutional 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. 현재 사용되고 있는 케이블망의 표준인 ITU-T J38은 유럽형 CATV 에 관한 표준이며, Series J는 텔레비전과 음향 그리고 그 외의 멀티미디어 신호 전송에 관한 것으로 J38은 그 중 디지털 CATV 시스템에 관한 것이다.

본 논문은 케이블 망에서 적용되고 있고 채널 부호화 방식의 알고리즘을 분석 및 Visual C++ 언어를 이용하여 시뮬레이션을 하여 성능을 분석하였으며, FPGA 구현시 문제점 제시 및 해결방안을 제시 하였다. 구현상의 문제점으로는 첫째로 케이블망의 오류 정정 방식의 부호화율에 따른 클럭이 많이 필요 하는데 본 논문에서는 메모리를 이용하여 필요한 클럭수로 줄였으며, 둘째로는 채널 복호화기에서 동기를 잃어버리면 복호 데이터가 모두 오류로 처리되는데, 본 논문에서는 몇 개의 레지스터를 이용하여 동기를 맞추었다. 셋째로, 복호단의 동기는 비트 단위의 동기인 반면 프레임간의 동기를 또한 맞추어야한다. 방송 데이터 통신인 경우 데이터 프레임의 맨 앞 또는 맨 뒤에 UW를 사용한다. 이는 수신단에서 데이터 프레임 동기를 맞추기 위해 사용된다. 동기를 맞추는 것은 매우 중요하다. 복호단에서 복호되어진 데이터들을 원래의 데이터 프레임으로 구성을 해야되는데, 이때 데이터 프레임의 동기를 맞추지 못한다면 원래의 데이터 프레임이 구성되지 않아 올바른 방송을 볼 수 없게 된다. 따라서 본 논문에서는 이러한 UW를 맞추어 구현하였다.

이러한 알고리즘 분석 및 구현상의 문제점 해

결을 토대로 본 논문에서는 ITU-T J38 annex B의 하향 스트림 채널 부호화 시스템을 VHDL 언어를 사용하여 FPGA 칩에 직접 구현하였다.

## II. CATV 하향 스트림 시스템 모델

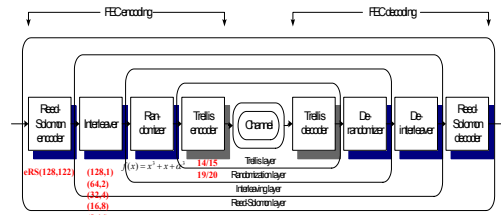


그림 1. CATV 하향 스트림 시스템 모델  
Fig. 1 System Model of CATV Down-Link Stream

그림 1은 ITU-T\_J83E\_하향표준에 제시되어 있는 채널코딩 시스템 모델이다. 부호화기는 eRS(128,122), convolutional interleaver, randomizer, TCM encoder의 순서로 되어있고, 복호화기는 그 역순이다. 다음 표1은 구현하고자 하는 시스템의 파라미터를 나타낸 것이다.

표1. 시스템 파라미터  
Table 1. System Parameters

방식	규격
eRS 부호	(128,122,3)
TCM	64-QAM : 14/15 256-QAM : 19/20
Interleaver	128x1
Randomizer	

본 논문에서 구현한 시스템은 부호화율 122/128, 에리정정 능력 t=3인 확장된 RS 부호를 사용하였고, 128x1의 사이즈를 가진 convolutional interleaver, randomizer, 그리고 두 가지 모드를 사용하는 TCM을 사용하였다.

### 2.1 확장된 RS (extended Reed-Solomon) 부호

본 논문에서 사용되어진 7 비트의 RS code는 (128,122)다. 보통의 RS code는  $2^n - 1$ 의 형태로 출력이 나오나 이는 마지막 한 심볼을 추가하는

확장된 RS 부호를 이용하여 1 심볼 오류를 더 정정 할 수 있다.

확장된 RS 부호는 부호화기의 출력으로 나오는 첫 번째 심볼부터 마지막 심볼까지 모두 더하는 부호 다항식을 식 (1)이라 하면, 식 (2), 식 (3)을 통해 구할 수 있다.

$$c(x) = m_{121}x^{126} + m_{120}x^{125} + m_{119}x^{124} + \dots + r_4x^4 + r_3x^3 + r_2x^2 + r_1x + r_0 \quad (1)$$

$$c_- = c(\alpha^6) \quad (2)$$

$$\begin{aligned} \hat{c} &= xc(x) + c_- \\ &= m_{121}x^{127} + m_{120}x^{126} + \dots + m_1x^7 + m_0x^6 \\ &\quad + r_4x^5 + r_3x^4 + r_2x^3 + r_1x^2 + r_0x + c_- \end{aligned} \quad (3)$$

### 2.2 TCM(Trellis Coded Modulation)

그림 2는 64-QAM에서의 TCM 블록도 이다. Randomizer 출력으로 나온 7 비트의 심볼들은 TCM 입력에 맞게 ordering 되어 비트 단위로 TCM 입력으로 들어가게 된다. 64-QAM의 경우 6비트씩 입력으로 들어가게 되는데 LSB 두 비트는 90도 invariant를 가지는 차등 부호화기를 거쳐 부호화율 4/5의 컨볼루션 인코더를 통과하여 맵핑되어진다. 256-QAM의 경우 코딩이 되지 않는 비트가 6비트가 되고, 나머지는 64-QAM과 같은 구조를 가진다.[2]

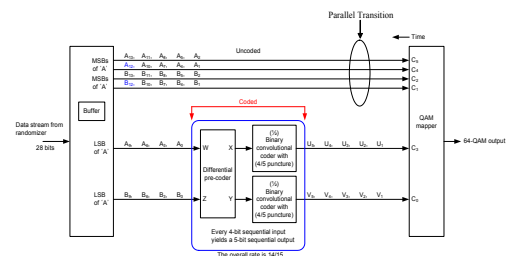


그림 2. 64QAM TCM 블록도  
Fig. 2 Blockdiagram of 64QAM TCM

#### 1) Differential Code

Differential code는 phase ambiguity를 방지하기 위해 사용된다. CATV 하향 스트림 시스템에서는 90°의 위상 틀어짐을 방지한다. 식 (4)는 그림 3의 differential encoder를 나타낸다. j는 시점을 나타낸다.

$$\begin{aligned} X_j &= W_j + X_{j-1} + Z_j(X_{j-1} + Y_{j-1}), \\ Y_j &= Z_j + W_j + Y_{j-1} + Z_j(X_{j-1} + Y_{j-1}) \end{aligned} \quad (4)$$

식 (4)에 의해 coding 된 데이터는 식 (5)에 의해 decoding 된다.

$$\begin{aligned} W_j &= X_j + X_{j-1} + (X_{j-1} + Y_{j-1})(X_j + Y_j) \\ Z_j &= X_j + Y_j + X_{j-1} + Y_{j-1} \end{aligned} \quad (5)$$

#### 2) Convolutional Encoder

그림 3은 convolutional encoder의 구조이다.

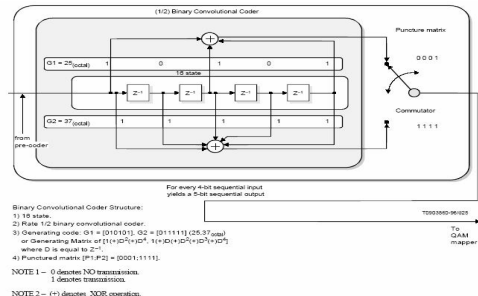


그림 3. 컨볼루션 인코더의 구조  
Fig. 3 The Structure of Convolutional Encoder

(2,1,5) convolutional encoder의 부호화율은 4/5이다. 이는 그림 3에서 puncture matrix에 의해 만들어진다.

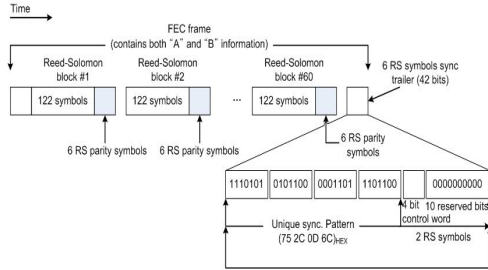
### III. 구현상의 문제점 및 해결 방안

#### 3.1 프레임 동기 알고리즘

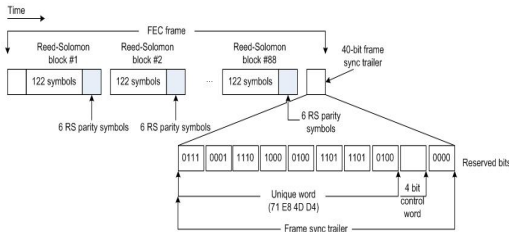
그림 4는 CATV 하향 스트림의 프레임 구조이며, 각 변조 방식 별 프레임 구조는 틀리고 그림

4(a)는 64-QAM 모드 일 때, 그림 4(b)는 256-QAM 모드 일 때의 프레임 구조이다.

5는 RS 심볼을 TCM 심볼로 만들기 위한 데이터 오더링을 나타낸다.



(a) 64-QAM모드의 데이터 구조

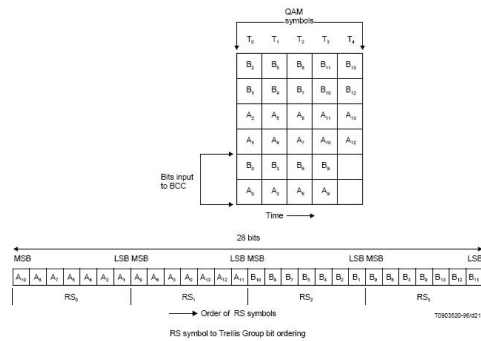


(b) 256-QAM모드의 데이터 구조

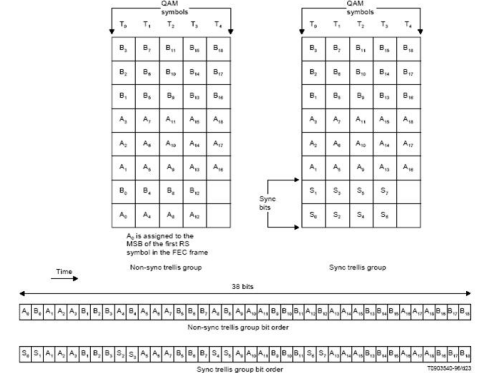
그림 4. CATV 하향 스트림의 데이터 구조  
Fig. 4 Data Structure of CATV Down-Link Stream

64-QAM의 경우 한 프레임의 데이터는 60개의 RS 블록과 42 비트의 sync trailer로 구성되어 있다. 42 비트의 sync trailer는 4개의 RS 심볼과 4비트의 interleaver control word, 10 reserved bit로 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.

일반적으로 동기를 맞추기 위해 심볼을 비트 단위로 풀어내어 한 비트씩 UW와 비교하면서 동기를 맞추는 방법을 사용한다. 하지만 이 경우 64-QAM같은 경우 한 TCM 심볼당 6비트이므로 6배 빠른 클럭이 필요하거나 굉장히 큰 메모리 용량이 필요하다. 또한 256-QAM의 경우 8배 빠른 클럭 또는 그에 맞는 메모리가 필요하다. 그림



(a) 64-QAM 데이터 오더링 패턴



(b) 256-QAM 데이터 오더링 패턴

그림 5. 데이터 오더링 패턴  
Fig. 5 Data Ordering Pattern

64-QAM의 경우 28bits의 데이터, 256-QAM의 경우 38bits의 데이터를 TCM 복호후 UW를 찾아 동기를 맞춘 후 뒷단의 RS 심볼단위로 derandomizer, deinterleaver, RS 복호기로 입력되는데 이때 UW를 정확하게 찾지 못하면 TCM 뒷단에서 모두 오류가 발생한다. 따라서 UW의 검출시각 동기 비트의 매칭 정도, 즉, threshold에 따라 프레임 오류율을 그림 6과 같이 구하였다.

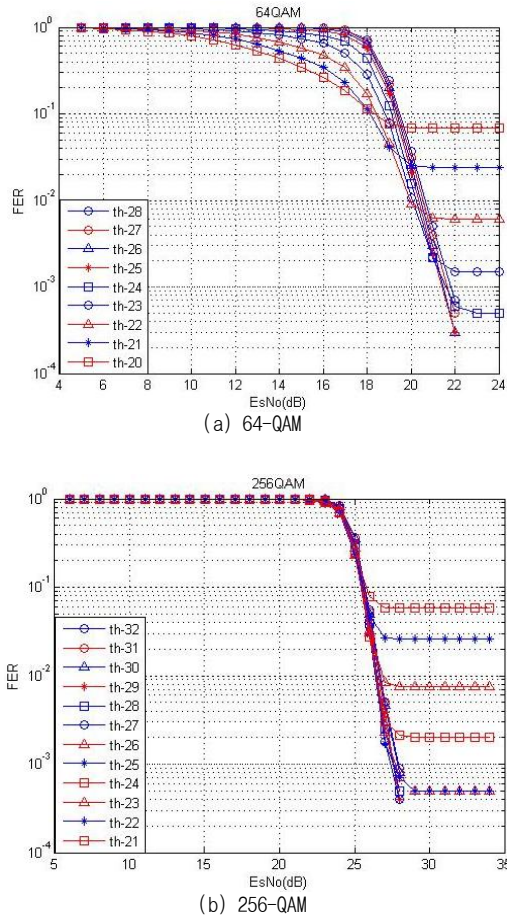


그림 6. Threshold에 따른 프레임 에러율  
Fig. 6 Frame Error Rates for Threshold

구현을 위한 sync-word detection에서 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. sync-word detection에서 threshold는 63-QAM에서는 최소 th=25 이상을, 256-qam에서는 최소 th=27 이상을 지정해야 할 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 페딩에 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 7은 사용한 레지스터와 UW detection의 flow-chart를 나타낸 것이다.

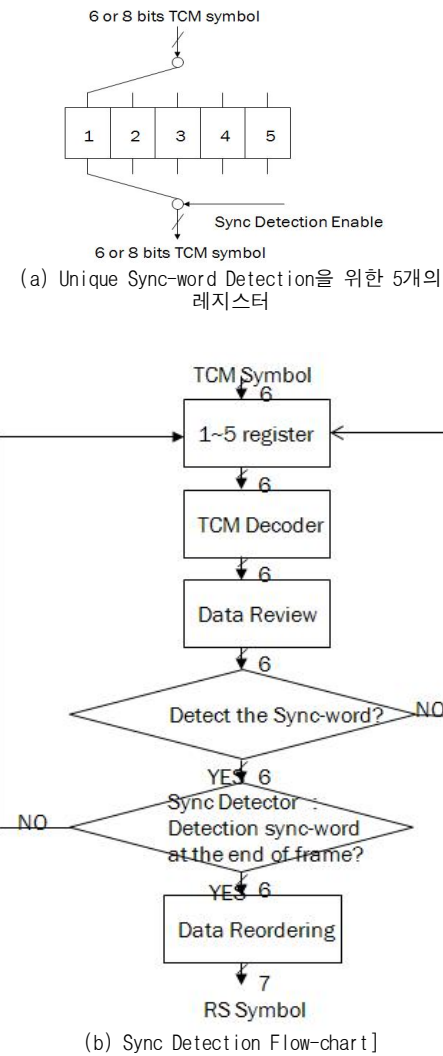


그림 7. Sync Detection을 위한 레지스터와 Flow-chart  
Fig. 7 The Register and Flow Chart for Sync Detection

그림 7 (a)의 5개의 레지스터에 1번부터 순서대로 데이터가 저장된다. 저장된 데이터는 순서대로 TCM decoder로 출력된다. 그림 7 (b)의 flow-chart에서 보면 레지스터에서 출력된 데이터는 TCM decoder에 의해 decoding 되어진다. decoding된 데이터는 sync-word detector에 의해 판별되어지는데, 맨 처음의 데이터부터 시작해서 한 프레임의 데이터가 decoding 될 때까지 sync-word를 찾지 못하면 TCM decoder 전의 레지스터로 신호를 보내게 된다. 레지스터에 sync-word detector에서 신호가 들어오는 순간 레지스터의 출력 순서를 하나 건너뛰게 한다. 예를 들어 sync-word detector에서 신호를 보내는 순간 레지스터의 출력 순서가 3번 레지스터였다면, 다음 클락에서는 4번 레지스터에서 출력이 되지 않고, 4번을 건너뛰고 5번 레지스터에서 출력 후 다시 순서대로 계속 출력을 하는 것이다. 이 방법은 그림 9의 데이터 오더링을 이용한 것으로 오더링 패턴에서 첫번째 TCM 심볼을 찾아 낼 수 있으면 sync-word detection을 할 수 있다는 것에서 감안한 것이다. 오더링 패턴에서 생성되는 TCM 심볼이 5개 이므로 5개의 레지스터를 사용하는 것이다. 이 방법을 사용할 경우 sync-word는 최대 5프레임의 데이터가 첫 detection에 필요하지만 5 프레임이 실제로 처리되는 시간은 극히 짧은 시간이므로 많은 클락과 계산량 없이 detection 할 수 있다.

처음 detection이 성공한 후 detection에 실패할 때까지 최소 35 프레임 동안 정확히 프레임이 끝나는 지점에서 sync-word가 계속 detection되는 것을 확인하고, detection되지 않는다면 detection과정을 처음부터 다시 수행한다. Detection이 성공하면 다시 데이터 오더링을 통해 TCM 심볼을 RS 심볼로 변환시킨 후 출력한다.

### 3.2 제한된 클락을 사용한 시스템 구현 기법

연접 부호의 구현일 경우 사용된 부호화 기법에 따라 각각의 클락이 필요하다. 그림 8은 CATV 하향 스트림 채널 코딩 시스템에서 각 블록마다 필요한 클락을 나타낸 것이다.

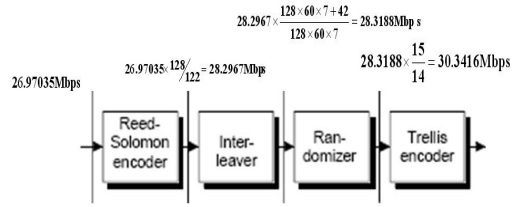
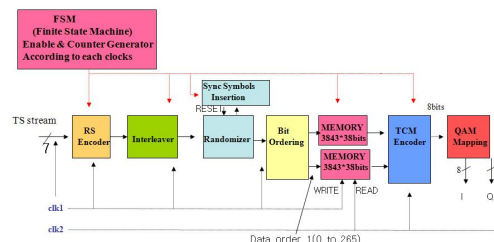


그림 8. 각 블록의 필요한 클락  
Fig. 8 Required Clock in each Block

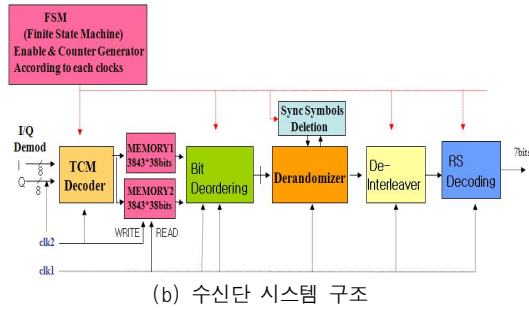
RS 부호와 TCM의 경우 부호화율이 적용이 되기 때문에 입력과 출력이 필요한 클락이 틀려야 된다. 송신단에서는 입력보다 출력의 클락이 빨라야 입출력에 대한 타이밍을 맞출 수 있다. Randomizer의 경우 출력단에서 데이터 프레임이 끝날 때 sync trailer를 첨가 하기 때문에 그에 대해 클락이 틀려진다. 그림 8에는 각 블록에서 출력시 필요한 클락 계산을 볼 수 있다. 수신단의 경우는 송신단과 반대로 입력의 클락이 출력의 클락보다 빨라야 함을 알 수 있다.

실제 시스템 구현시 각 블록에 대한 클락들을 맞춰서 적용시킬 수 없다. 그 이유는 각 클락의 동기가 맞아야 하기 때문이다. 각 클락간의 동기가 맞지 않으면 시스템은 오동작을 일으키게 된다. 기본적으로 제공되는 클락과의 동기를 맞추는 가장 기본적인 방법은 분주를 이용하는 것이다[5]. 기본 클락을 분주를 시킬 수 있다면 클락의 동기는 문제가 되지 않는다. 하지만 그림 8에서 알 수 있듯이 각 블록에서 필요한 클락은 분주가 불가능하다. 그렇기 때문에 클락이 제한되어 있으면 시스템을 구현하기 힘들다.

이러한 문제를 해결하기 위해 그림 9와 같이 두 개의 메모리를 사용하였다.



(a) 송신단 시스템구조



(b) 수신단 시스템 구조

그림 9. 구현시 송수신 시스템 구조

Fig. 9 Transmitter Structure for Implementation

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

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

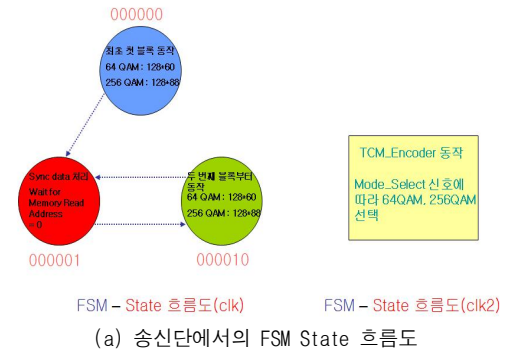
RS 부호, interleaver, randomizer와 같이 ‘clk1’에 의해 동작하는 부분을 part1이라고하고, TCM과 같이 ‘clk2’에 의해 동작하는 부분을 part2라고 하자.

그림 9에서 볼 수 있듯이 우선 part1은 ‘clk1’에 의해 동작을 하고, 첫번째 메모리에 저장이 된다. 첫번째 메모리에 모두 저장이 된 후 두번째 메모리에 저장을 시작할 때 ‘clk2’에 의해 첫번째 메모리에서 읽기 동작이 시작된다. 첫번째 메모리에서 데이터를 읽어내면 part2가 동작한다. ‘clk1’이 두번째 메모리에 모두 저장을 끝마쳐도 ‘clk2’의 첫번째 메모리 동작은 끝나지 않는다. 이 때 ‘clk2’의 첫 번째 메모리 동작이 끝날 때까지 part1은 동작을 하지 않고 part2의 동작이 끝날 때까지 기다린다. ‘clk2’에 의해서 part2가 첫번째 메모리의 동작이 끝나면 두번째 메모리 동작을 시작하고, ‘clk1’에 의해 part1은 다시 첫번째 메모리 동작을 시작한다. 이러한 절차로 part1은 송신단에서는

메모리에 번갈아가며 데이터를 저장하고, 수신단에서는 데이터를 읽어낸다. 마찬가지로 part2는 송신단에서는 읽기 동작을 수신단에서는 데이터를 저장한다. 이러한 방법에 의해 기본 클럭 하나를 이용하여 시스템을 구현할 수 있다.

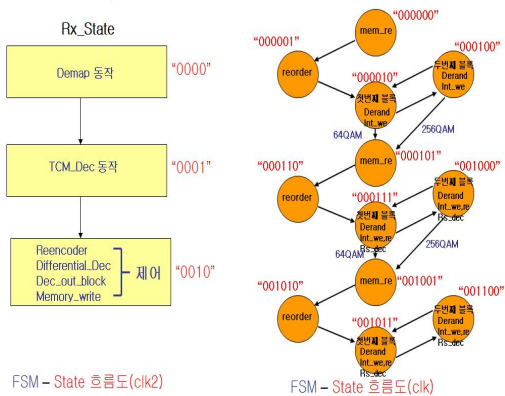
#### IV. FPGA 구현 결과

제 2장과 제 3장에서 구현시 필요한 알고리즘과 시스템 구현시 문제점, 그리고 그 해결 방법에 대하여 알아보았다. 제 4장에서는 이것을 토대로 실제 구현하기 위해 timing simulation을 하였고 이를 FPGA 칩에 구현하였다. 본 논문에서 구현에 사용한 전체적인 컨트롤은 FSM(Finite State Machine)을 사용하였고, 그림 10은 그 흐름도를 나타낸 것이다.



FSM - State 흐름도 (clk)      FSM - State 흐름도 (clk2)

(a) 송신단에서의 FSM State 흐름도



FSM - State 흐름도 (clk2)      FSM - State 흐름도 (clk)

(b) 수신단에서의 FSM State 흐름도

그림 10. 송수신단의 FSM State 흐름도

Fig. 10 FSM Flow of Transmitter and Receiver

그림 10 (a)는 송신단에서의 FSM state에 대한 것이다. 'clk'사용에서 상태 값이 "000000" 일 때 최초 첫 데이터 프레임을 처리하고, 첫 프레임이 처리된 후 sync trailer 삽입을 위해 "000001"로 바뀐다. Sync trailer를 삽입하면 "000010"으로 상태 값이 바뀌면서 두 번째 프레임이 처리되며, 이 상태에서 메모리의 데이터를 TCM으로 출력하게 된다. 두 번째 프레임 처리 후 다시 상태 값이 "000001"로 되며 sync trailer를 삽입한다. 이후 "000010"과 "000001"의 상태 값이 계속 반복된다. 'clk2'사용에서는 메모리에서 연속적으로 데이터가 출력되기 때문에 상태 값의 변화가 없다.

그림 10(b)는 수신단의 FSM state에 대한 것으로, 'clk2' 사용시에는 상태 값 "0000"에서 TCM 복호기 이전의 sync detection을 위한 레지스터 사용 컨트롤 신호가 처리된다. "0001"에서 TCM 복호기의 컨트롤 신호가 처리되며, TCM 복호기의 출력이 나오면 "0010" 으로 상태 값이 변하여 'clk2'를 사용하는 나머지 component에 대한 컨트롤 신호를 처리하고, 이 때 데이터는 연속적으로 처리되므로 상태값 반복은 필요하지 않다. 'clk'에서는 메모리 출력과 데이터 오더링 후 상태 값 "000010"과 "000100"을 이용하여 derandomizer를 동작 시키고, deinterleaver에서 메모리의 데이터 저장을 실행한다. 그 후 다시 메모리 출력과 데이터 오더링 후 상태 값 "000111"과 "001000"을 이용하여 derandomizer와 deinterleaver 메모리 데이터 저장을 실행한다. 이 때, 메모리가 다 채워지는 시점이 있기 때문에 그 시점부터는 deinterleaver 출력을 같이 하게 된다. 그 후 "001001"에서부터 "001100"까지의 상태 값이 반복되면서 'clk'의 복호단이 실행된다.

그림 10에서 송신단에 비해 수신단의 상태 변화가 복잡한 이유는 UW detection과 deinterleaver에 의해 상태의 가지 수가 많아 졌기 때문이다.

제 2장에서 알아본 알고리즘 각각을 VHDL을 이용하여 타이밍 시뮬레이션을 했고, 제 3장에서의 구현상의 문제점에 대한 해결책까지 VHDL을 이용하여 시뮬레이션 하였다. 이렇게 CATV 전체 시스템에 대하여 시뮬레이션 한 후 FPGA칩을 이용하여 구현하였다. 시스템 구현시 사용한 시뮬레이션 툴은 Xilinx 사의 modelsim을 이용하였다.

다음 표 2은 구현시 FPGA 칩사용에 관한 내용이다.

표 2. FPGA 구현시 사용된 게이트 정보  
Table 2. Gate Information for FPGA Implementation

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Notes
Number of Slice Flip Flops:	6,468	27,392	23%	
Number of 4 input LUTs:	15,667	27,392	57%	
<b>Logic Distribution:</b>				
Number of occupied Slices:	10,650	13,696	77%	
Number of Slices containing only related logic:	10,650	10,650	100%	
Number of Slices containing unrelated logic:	0	10,650	0%	
<b>Total Number 4 input LUTs:</b>	<b>16,909</b>	<b>27,392</b>	<b>61%</b>	
Number used as logic:	15,667			
Number used as a route-thru:	1,242			
Number of bonded IOBs:	146	416	35%	
Number of PPC4Bs:	0	2	0%	
Number of Block RAMs:	46	136	33%	
Number of GCLKs:	2	16	12%	
Number of GTs:	0	8	0%	
Number of GT10s:	0	0	0%	

구현 시 총 사용된 슬라이스 수는 10,650개이고, 전체의 77%를 사용하였다. 메모리 또한 총 33%를 사용하여 충분한 구현 능력을 보여주고 있다.

## V. 결론

본 논문에서는 ITU-T J38 CATV 하향 스트림 채널 코딩 시스템을 구현하여 각 블록에 대한 알고리즘의 성능 분석과 timing simulation을 보였고, 이러한 연결 부호 시스템을 구현할 때의 문제점과 이를 극복할 수 있는 구현 기법을 제시하였다.

ITU-T J38 CATV 하향 스트림 채널 코딩 시스템은 RS부호, interleaver, randomizer, TCM(64-QAM, 256-QAM)으로 구성되어 있다. RS 부호는 기존의 알고리즘에서 extended 심볼을 추가하여 에러 정정 능력을 2에서 3으로 향상 시킨 extended RS 부호를 사용하였다. Interleaver는 convolutional interleaver를 사용하였는데, 구현시 많은 레지스터를 사용하게 되면 계산량이 많아지고 하드웨어의 처리에 부하가 많이 걸리는 문제가 있다. 이를 극복하기 위해 메모리를 사용하여 레지스터 사용과 같은 효과를 나타내었다. 또한 데이터의 DC 신호 방지를 위해 randomizer를 사용하였고, 64-QAM, 256-QAM의



두가지 모드를 사용하는 TCM을 구현하였다.

이러한 시스템에서 수신단에서의 데이터 프레임 동기를 맞추기 위해 5개의 레지스터를 TCM decoder 전단에 사용하여, 이를 이용해 프레임 동기를 맞추었다. 이 방법을 사용함으로써 심볼을 비트 단위로 처리하지 않아 불필요한 클럭 사용을 방지할 수 있었다. 또한, 이렇게 각 블록마다 다른 클럭이 필요한 시스템을 구현하기 위해 심볼간의 변환이 있는 곳에 두 개의 메모리를 사용함으로써 하나의 기본 클럭만을 이용한 구현을 하였다.

구현 시 전체 컨트롤은 FSM(Finite State Machine)을 사용하였고, 구현에 사용된 FPGA칩은 Xilinx사의 'Vertex II pro xc2vp30-5'를 사용하여 구현하였다. 구현 결과 칩 용량에 충분한 슬라이스를 사용하여 무리가 없었고, 또한 디바이스에서 제공되는 클럭에 대해 계산량 초과가 생기지 않은 성공적인 구현을 하였다.

**참 고 문 헌**

[1] H. Wymeersch, H. Steendam and M. Moeneclaey, "Error Performance Analysis of a Concatenated Coding Scheme with 64/256-QAM Trellis Coded Modulation for the North American Cable Modem Standard", ISIT Cambridge, MA, USA, Aug., 1998.

[2] G. Ungerboeck, "Trellis Coded Modulation with Redundant Signal Sets, Part I : Introduction" , IEEE Communications Magazine, Vol. 25, No. 2, pp. 5-11, February 1987.

[3] Carden, Frank, and Brian Kopp, "A Quantized Euclidean soft Decision Maximum Likelihood Sequence Decoder of TCM", IEEE Military Communications Conference, Vol. 2, pp. 279-682, October 1988.

[4] Eun-A Choi, Ji-Won Jung, Nae-Soo Kim, Young-Wan Kim, and Deock-Gil Oh,

"A Simplified Decoding Algorithm Using Symbol Transformation for Turbo Pragmatic Trellis-Coded Modulation", 「ETRI Journal」, 제27권, 제2호, pp.223-226, ETRI, 2005.

[5] 김도석, 정훈주, 이용환 " 직렬방식의 주변장치 통합 인터페이스 설계", 한국정보전자통신기술학회 4권 1호, pp.68-75, 2011.

**저자약력**

**정 지원(Ji-Won Jung)**



**정희원**

1989년 2월 :성균관대학교 전자공학과(공학사)  
 1991년 2월 :성균관대학교 전자공학과(공학석사)  
 1995년 2월 :성균관대학교 정보공학과(공학박사)  
 1991년 1월~1992년 2월 : LG 정보통신연구소 연구원  
 1995년 9월~1996년 8월 : 한국통신 위성통신연구실 선임연구원  
 1997년 3월~1998년 12월 : 한국전자통신연구원 초빙 연구원  
 1996년 9월 ~ 현재 : 한국해양대학교 전파공학과 교수  
 2001년 8월 ~ 2002년 8월 : 캐나다 NSERC Fellowship (Communication Research Center 근무)

<관심분야> 위성통신, 이동통신, 변.복조기술, 채널코딩, FPGA 기술 등