DAB 수신기용 TCM 디코더의 설계

김덕현*, 김건, 박소라, 정영호, 오길남
광주대학교 컴퓨터전자통신공학부*, 한국전자통신연구원 무선방송연구소 DAB팀

A Design of the TCM Decoder for DAB Receiver

Duckhyun Kim*, Geon Kim, Sora Park, Youngho Chung, Kilnam Oh
Kwangju University*, DAB System Team of ETRI

Abstract

The Trellis Coded Modulation(TCM) allows the considerable achievements of coding gains compare with conventional multi-level modulation without compromising bandwidth efficiency. In this paper, we are presented a design of the parallel Viterbi decoder for 16-QAM TCM decoder with large constraint length (K=9), which can be applicable for the Digital Audio Broadcasting(DAB) receiver. As a mid-term result, a parallel Branch Metric Calculator (BMC) can compute 16 BMs within 3 clocks and a parallel 16 Add-Compare-Selects (ACS) unit can compute in a single clock. And also, two 256 Path Metric Memories (PMM) 32 Trace Back(TB) memories are specially designed with shuffle exchange switches for 16 parallel accesses. As a VHDL simulation, we can find the correctness of proposed model, which can be operated 16 μS per symbol. Now, we are performing the hardware reduction for realtime operation and FPGA implementation.

I. 서론

디지털 기술은 정보, 통신 등의 영역에서 비약적인 발전을 가져왔으나 방송분야는 기술의 안정성으로 인해 다른 분야보다 디지털화가 늦은 편이다. 그러나 기술적 파급효과가 지대하므로, 선진 각 국에서는 방송의 디지털화를 서두르고 있다. 이미 영국과 미국에서는 각각 DVB (Digital Video Broadcasting)와 ATSC (Advanced Television System Committee) 규격에 의한 DTV(Digital TeleVision)방송을 실시하고 있으며, 국내의 경우도 DTV 실험 방송을 시작하였다. DAB의 경우, 유럽 및 캐나다에서는 Eureka-147 방식으로 방송을 실시하고 있으며, 미국이나 우리나라 등은 방식개발을 서두르고 있는 실정이다[1].

한국전자통신연구원(Electronics Telecommunication Research Institute : ETRI)에서는 채널 코딩으로 Reed Solomon(RS)-길쌈 연접부호 방식과 RS-TCM 연접부호 방식의 In-Band DAB 시스템을 개발 중에 있다. 특히 In-Band DAB 시스템의 경우 가용 대역폭이 한정되어 있으므로, 대역폭의 증가 없이 전송률을 높이기 위해서는 TCM의 선택이 요구된다[2].

본 논문에서는 Du와 Vucetic[3]이 제안한 K=9, 16-QAM TCM 구조를 채택한 TCOFDM (Trellis Coded Orthogonal Frequency Division Multiplexing) DAB 시스템에서 TCM 복호를 수행하기 위한 병렬 비터비 복호기를 구조를 제안하고 그 구현 가능성에 대해 논의한다. II 장에서는 TCM 복호기의 상세 구조를 논의하고, III 장에서는 이러한 복호기의 VHDL시뮬레이션 및 논리 합성된 결과를 논의하며, IV장에서 결론을 맺는다.

II. TCM 복호기 구조

TCM 복호기는 종래 길쌈부호의 비터비 복호와 달리 복호 시 많은 계산을 필요로 한다. 왜냐하면 각 상태에서 분기될 수 있는 가지(Branch)의 수가 입력 비트와 상태수의 승수배에 비례하기 때문이다. 만일 TCM을 적용 할 때 구속장 K가 크면 직렬 비터비 복호기(Serial Viterbi Decoder)로서는 실시간 수행이 어려워지며, 병렬로 ACS 연산을 수행할 수 있도록 설계되어야 한다. 이 경우 병렬 ACS 동작에 필요한 경로척도(Path Metric) 메모리의 관리의 어려움이 따르게 된다[4]. 본 논문에서는 실시간 동작을 위해 16 개의 ACS 연산장치 병렬로 사용하고 16 개의 자료를 동시에 액세스 할 수 있는 PM 메모리를 설계하여 128 회의 ACS 동작으로 1 개의 심벌을 복호 가능한 TCM 복호기 구조를 제안한다. 그림 1은 제안된 TCM 복호기의 블록도를 나타낸다.

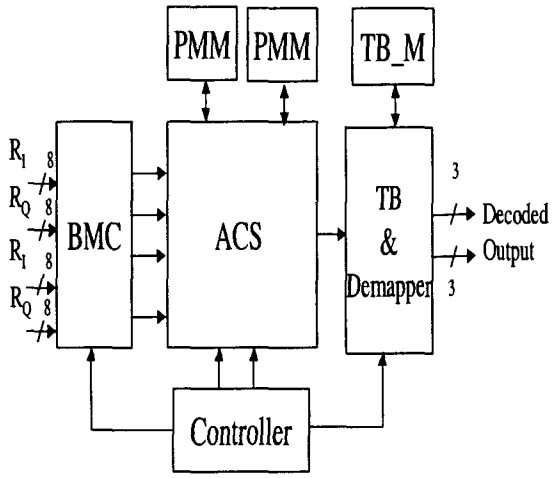

그림 1. TCM 디코더 블록도
Fig. 1. Block diagram of TCM decoder.

그림 1의 TCM 복호기는 가지척도계산기(Branch Metric Calculator), 가산비교선택기(Add-Compare-Selector), 역추적(Trace Back) 및 디매퍼의 세 가지 블록으로 구성된다. BMC 블록은 QAM 신호공간상에서 유클리드 거리(Euclidean Distance : ED)를 계산한다. 본 논문에서는 계산의 간략화를 위해 ED를 이용하는 대신 제곱 유클리드 거리 (Squared Euclidean Distance : SED) 방식을 적용하였으며 II-1절에서 그 구조를 상세히 논의할 것이다. 또한 16개의 데이터를 동시에 비교할 수 있는 가산비교선택기(Add-Compare-Selector)를 가지며 이의 구조는 II-2절에서 논의할 것이다. 병렬 ACS를 위한 16 개의 자료를 동시에 읽고 쓸 수 있는 이중 버퍼링 구조의 PM 메모리를 가지며 그 구조는 II-3절에서 논의 할 것이다. 역추적부 역시 별도의 병렬 메모리를 채용하여 구성하였으며 이는 II-4절에서 논의한다. 기타 정규화 및 다음 상태 어드레스 발생기의 구조는 II-5절에서 상세히 논의할 것이다.

II-1 BMC 블록

BMC 블록에서는 수신된 신호와 각 신호점 사이의 유클리드 거리(ED)를 계산한다. 그러나 제곱근 연산의 어려움으로 인해 이를 룩업 테이블로 실현할 경우 많은 하드웨어를 소모하게 된다. 예를 들면 Qualcomm 사의 Q1900 칩의 경우 16-PSK의 TCM 모드로 동작될 때 섹터 구분과 BMC 계산을 위한 2 x 4 KByte의 외부 ROM을 필요로 한다[5]. ACS 동작에서 BM은 상대비교를 위한 척도로서 사용되어지므로 하드웨어 구현의 간략화를 위해 제곱 유클리드 거리(SED)를 적용하면 식(1)과 같이 표현된다.

$$\begin{aligned} SED &= (R_I - C_{i,j})^2 + (R_Q - C_{i,j})^2 \\ &= R_I^2 - 2R_I C_{i,j} + C_{i,j}^2 \\ &\quad + R_Q^2 - 2R_Q C_{i,j} + C_{i,j}^2 \end{aligned} \tag{1}$$

여기서 R_I, R_Q 는 수신된 신호를 나타내고 는 소수점 이하 3 비트로 표현된 8 비트 2진수 $C_{i,j}$는 QAM 코드 워드를 나타낸다. 입력신호로 정하였다. 이때 R_I^2, R_Q^2 항들은 모든 코드 워드 사이에 더해지므로 ACS에 영향을 주지 않는다. 따라서 이를 제거하고 식(1)을 다시 쓰면

$$SED = |C_{i,j}^2 - 2R_I C_{i,j}| + |C_{i,j}^2 - 2R_Q C_{i,j}| \quad (2)$$

이 된다. 여기서 $C_{i,j} \in \{-3,-1,1,3\}$ 이다. 따라서 식(2) 는

$$\begin{aligned} SED &= |1 \mp 2R_I| + |1 \mp 2R_Q| \ \text{for} c_{i,j} \in \{-1,1\} \\ &= |9 \mp 6R_I| + |9 \mp 6R_Q| \ \text{for} c_{i,j} \in \{-3,3\} \end{aligned} \quad (3)$$

로 나타낼 수 있다. 식(3)에서 2와 6의 곱은 미리 알려져 있는 값이므로 곱셈이 제거될 수 있다. 따라서 식(3)은 가산기와 멀티플렉서 만으로 구성할 수 있다. C로 구성된 DAB 시뮬레이터 상에 SED, ABS, ED에 대해 시뮬레이션 한 결과 SED는 ED비해 약 0.2 dB 정도 우수한 것으로 나타났으며, ABS와 ED는 거의 동등한 것으로 나타났다. 이는 제곱을 취함으로서 신호 공간상에서 인접신호 사이의 거리를 늘이는 효과 때문인 것으로 사료된다. 따라서 본 논문에서는 SED를 BM으로 채택하였고 음의 값이 나오는 것을 방지하기 위해 절대값을 취하였다. 이러한 8 BMC 계산블록은 그림으로 나타내면 그림 2와 같다.

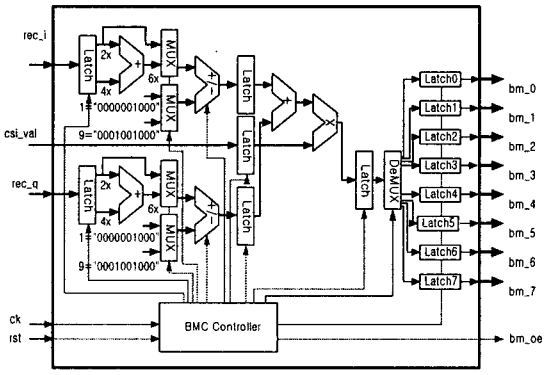

그림 2. BMC 블록도

Fig 2. Block diagram of BMC.

그림 2에서 8 비트 크기의 I, Q 수신 신호 값이 입력되면 3 클럭 후 16개의 BM 값을 동시에 계산한다. 또한 다음 입력 값을 동시에 수신하면 짝수 및 홀수의 BM 값은 동시에 계산할 수 있으며 이는 그림 2의 BMC 블록이 병렬로 사용되며 그림 3은 16-BMC 블록의 논리 합성된 결과를 나타낸다.

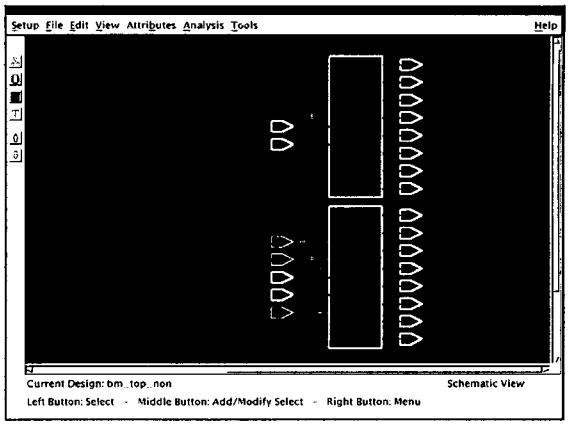

그림 3. 16-BMC 합성결과

Fig. 3. Synthesis result for 16-BMC.

II-2 ACS 블록

ACS 블록은 BM값과 현재의 PM값을 더한 결과와 다음 상태의 PM 값을 비교하여 작은 값은 저장하고 그렇지 않으면 생존 경로에서 제거한다. 이때의 상태 값을 역추적 메모리에 저장한다. TCM에서 각 상태에서 분기될 수 있는 가지의 수가 입력 비트 수의 승수 배에 비례하기 때문에 입력 비트 수가 증가하게되면 ACS 수는 지수적으로 증가하게된다. 즉 입력 비트 수를 b 비트, 구속장을 K로 가정하면 각 상태마다 2^b의 가지가 존재하고 전체 $2^{(b+K-1)}$ 의 ACS 연산을 수행하게 된다. 구속장 K=9 이면 한 심볼 당 256*8 = 2048 회의 ACS 연산을 수행하여야 한다. 이를 N 개의 병렬 ACS 구조로 설계할 경우 N 개의 2048/N 번의 ACS 동작과 동시에 N 개의 PM 값을 액세스 할 수 있어야 한다. 본 논문에서는 N=16 으로 채택하여 16 개의 ACS 연산을 병렬로 수행하고 이 과정을 128 회 반복하면 PM 메모리에는 각 상태에서 최소 경로인 값이 저장된다. 이러한 ACS 16블록의 합성 결과는 그림 4와 같다.

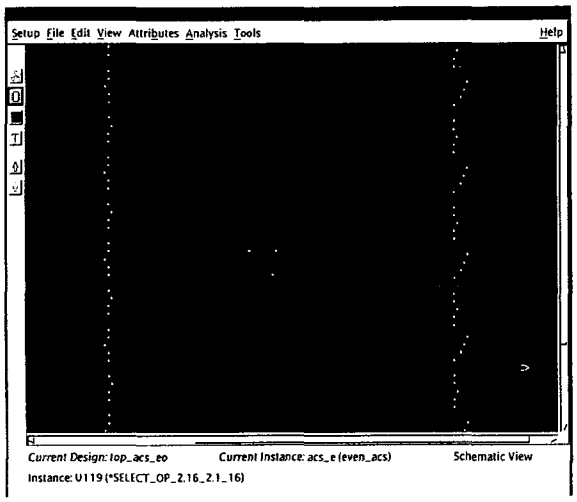

그림 4. ACS-16 합성 결과
Fig 4. Synthesis result of ACS-16.

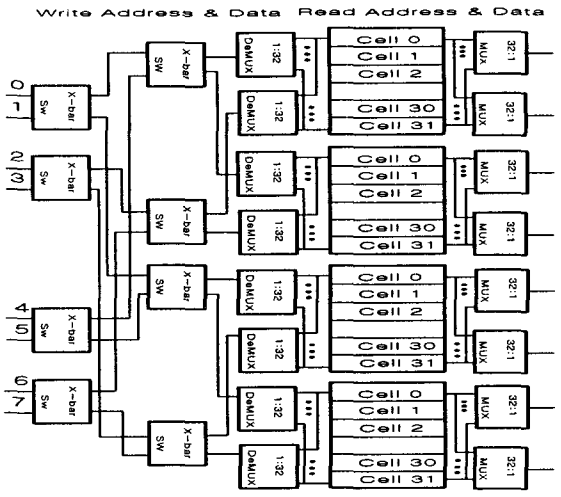

그림 5. PM128 메모리 블록도
Fig 5. Block diagram of PM128 memory.

II-3. PM 메모리

PM 메모리는 ACS연산결과 각 상태의 최소 경로 메트릭을 저장하는 구실을 한다. 이는 ACS 연산수와 같은 메모리 참조가 이루어지므로 K가 클 경우, PM 메모리를 종래의 메모리로 실현할 때 실시간 동작이 어렵다. 본 논문에서는 사용되는 16개의 병렬 ACS 연산을 위해서는 16 번의 메모리 참조가 이루어지고 실시간 동작을 위해서는 16 개의 자료를 동시에 액세스 할 수 있도록 다중입출력단자(Multi- Port RAM)를 가진 구조이어야 한다. 한 상태에서 다음 상태로 천이 할 때 각 각의 어드레스는 병렬로 계산할 수 있다. 특히 제안된 TCM 구조에서 각 각의 상태에서 짝수 상태와 홀수 상태에서 분기되는 어드레스는 전반 128 개와 후반 128 개로 분리된다[2]. 각 상태는 다시 어드레스의 7 번째 MSB값으로 64 개의 블록으로 나누어지고 이는 나중에 스위칭 할 수 있어야 한다. 6번째 어드레스에의 해서는 다시 32 블록으로 분해되므로 전체 128 상태 메모리는 4개의 32 블록으로 분해되어 동시에 8 개의 자료를 액세스 할 수 있다. 따라서 짝수 및 홀수 상태의 각 8 개 값을 동시에 읽거나 쓸 수 있는 메모리의 구조가 가능하고 이를 블록도로 나타내면 그림 5와 같다.

II-4 역추적부와 디매퍼

ACS 연산의 결과, 생존 경로 메트릭 값들은 PM 메모리에 기록되고 그때의 상태 값들은 TB 메모리에 기록된다. 이때 최소 경로 값을 찾아 이 상태에 도달하는 각 각의 상태를 역추적한다. 이러한 역추적의 결과 현재 상태에서 가장 가까운 과거상태를 추정하게 된다. 이를 역추적(Trace Back)이라 한다. 또한 길쌈부호기에서 비터비 디코더의 역추적부는 현재 상태의 비트 값을 추정하게되나 TCM 에서는 현재 상태와 과거 상태 값으로부터 부호기를 사용하여 각 심볼을 역산한다. 이를 디매핑이라 한다. 이를 위해 256 ACS 연산 후 최소 경로 값을 찾아야하며 256 개의 PM 값 중에서 최소 값을 찾는 작업 또한 용이하지 않다. 본 논문에서는 16개의 PM 메모리를 동시에 액세스할 수 있으므로 이 16 개의 경로 값 중에서 국부 최소 값을 찾은 후 다시 16 개의 국부 최소 값을 찾을 수 있는 최소 값 탐색 블록과 TB 메모리를 설계하였다.

설계된 블록은 파이프라인으로 동작하며 24 클록에 최소 값을 찾아 그때의 상태 값을 TB

블록에게 전달한다. 이 상태를 시작 어드레스로 하여 역 추적하면 현 상태에 이르는 과거상태를 추적 가능하다. 이때 역추적 깊이는 시뮬레이션 결과 42 이상이면 성능에 영향을 주지 않는 것으로 나타났다. 본 논문에서는 하드웨어 크기의 제약으로 인해 32로 선택하였으며 5비트 링 카운터로 역추적 어드레스를 구현하였다. 메모리는 256x32x8 = 8 Kbyte 가 요구되며 이는 별도로 PM 메모리와 같은 구조로 합성하였다. 이러한 역추적부의 구조는 그림 6과 같다. 디매퍼 구조는 부호기 구조를 그대로 사용하여 현재 상태와 전 상태로부터 부호화할 수 있으나 이를 간략화하여 그림 6의 디매퍼부와 같이 나타낼 수 있다.

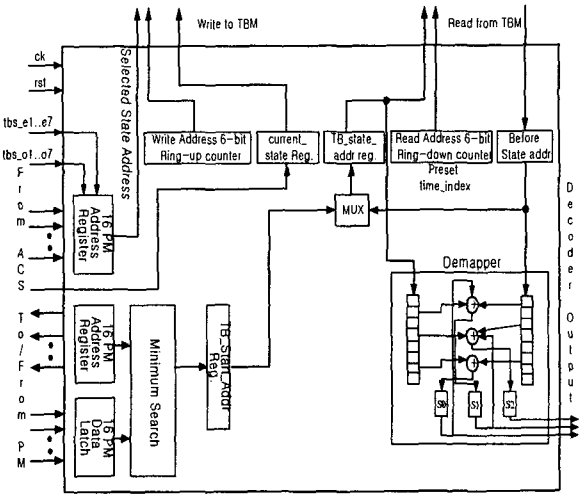

그림 6. 역추적부 및 디매퍼 블록도
Fig 6. Block diagram of Trace Back & Demapper.

II-5. 번지 지정 및 정규화

각 상태에서 번지 지정 결과는 현재 상태로부터 다음 상태의 천이는 별도의 번지 발생기로부터 생성된다. 이때의 번지생성은 하나의 상태로부터 독립적으로 계산될 수 있다. 각 각의 번지는 문헌[2]의 패리티 체크 다항식으로부터 생성된다. 짝수 상태의 첫 번째 가지의 번지발

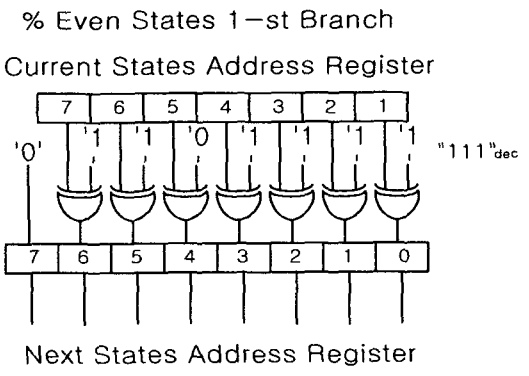

그림 7. 다음 상태 번지 발생기
Fig. 7. Next state address generator.

생기의 구조는 그림 7과 같다. ACS 결과는 PM 값의 증가를 가져다준다. 따라서 이를 정규화하기 위해서는 PM값의 세 MSB를 체크함으로써 이루어진다. PM 값의 세 MSB를 검사결과 전부 1이 되면 다음 ACS에서 넘침(Over flow)이 발생될 가능성이 높으므로 MSB를 음으로 만들어줌으로서 넘침을 회피하게 된다. 이를 정규화라 하고 일정 기간 동안 정규화 횟수를 센다. 만일 동기가 맞지 않으면 일정 기간 동안 정규화 횟수가 이상 증가하게되고 이 경우 시스템은 초기화된다.

III. 시뮬레이션 및 검토

제안된 구조는 Synopsys 툴에서 VHDL로 코딩하여 논리 합성하였다. 테스트 벤치를 작성하여 C 프로그램의 결과와 비교하여 정상적인 동작을 확인할 수 있었다. 논리합성 결과 BMC 블록은 약 8천 게이트가 소요되고 3 클럭에 16개의 BM 값을 동시에 계산할 수 있다. 16 ACS는 약 4만 게이트가 소요되며 단일 클럭에서 16 ACS 동작을 수행한다. PM 메모리는 약 6만 게이트가 소요되고 TB 메모리 또한 48만 게이트가 소요되어 단일 블록으로서는 제일 크다. 이는 메모리를 플립플롭으로 합성한 때문이다. 이 메모리는 단일 클럭에 16 개의 데이터를 동시에 읽거나 쓸 수 있으며 그림 8은 임의 값

을 읽고 쓰는 타이밍을 나타낸다. 이상의 시뮬레이션 결과를 바탕으로 워드 크기를 줄여서 Altera 사의 Max+plus II에서 타이밍 시뮬레이션을 수행하였다. 그림 9는 K=9인 TCM 디코더의 전체 타이밍 중, 첫 번째 심볼 복호의 끝 부분을 나타낸다. 이 그림을 보면 복호 시간은 심볼 당 약 16μS 가 소요됨을 보여주며 이는 실시간 동작(약 2μS)이 어려움을 알 수 있다. 이때 클럭은 50MHz를 가정하였다.

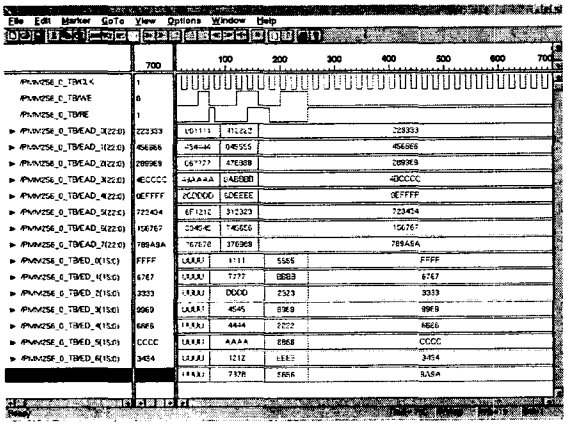

그림 8. PM 메모리 블록 타이밍도
Fig. 8. Timing diagram for PM memory.

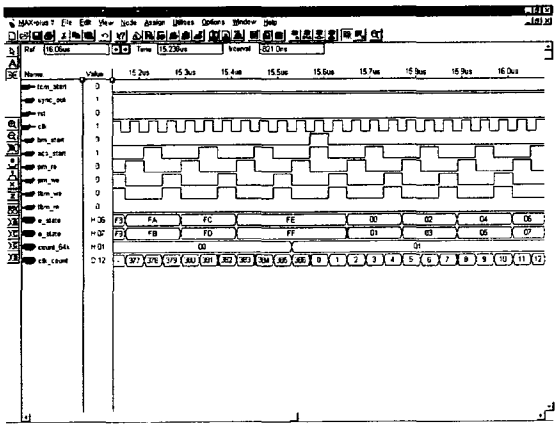

그림 9. K=9인 TCM 복호기 타이밍도
Fig. 9. Timing diagram for TCM decoder (K=9).

IV. 결론

본 논문에서는 DAB 시스템에 적용하기 위한 16-QAM TCM 복호기의 병렬 비터비 디코더 구조에 대해 논의하였다. 제안된 구조는 간략한 BMC 구조를 지니며, 16 개의 자료에 대해 동시에 ACS 연산을 수행 할 수 있다. 또한 16 개의 데이터를 동시에 읽고 쓸 수 있는 다중입출력 단자를 가진 PM 와 TB 메모리를 적용하였다. Synopsys 툴을 이용한 VHDL 모델링 및 시뮬레이션 결과 정상적인 동작을 확인하였다. 그림 9에서 살펴본 바와 같이 50 MHz 클럭을 가정할 때, K=9 인 16-QAM TCM 복호기는 한 심볼 복호에 약 16 μS의 시간이 소요되어 DAB에서 요구되는 2 μS 이내의 실시간 동작이 어렵고 하드웨어 크기 또한 FPGA 내에 집적하기 어렵다. 따라서 구속장 K를 작게 함으로써 하드웨어를 줄이며 실시간 동작을 가능케 하는 타이밍 시뮬레이션을 수행 중에 있다.

V. 참고문헌

[1] 박재홍, 오길남, "디지털 방송 표준화 현황 및 방식 개발," 전자공학회지, pp. 25 ~ 35, 제26권, 제6호, 1999년 6월.

[2] L.H.C. Lee, *Convolutional coding : Fundamentals and applications*, Artech House Inc., Norwood, MA. 1997.

[3] J. Du, B. Vucetic, "New 16-QAM trellis codes for fading channels," IEEE Electronics Letters, Vol. 27, No. 12, June 1991.

[4] 지현순, 박동선, 송상섭, "다중의 ACS 모듈을 갖는 병렬 비터비 알고리즘의 메모리 관리 방법", 한국통신학회 논문지, pp. 2077 ~ 2089, Vol. 21, No. 8, 1996.

[5] Qualcomm Inc., " Q1900 Viterbi/trellis decoder ", Forward error correction products data book, pp. 1-46 ~ 1-49, Aug. 1998.