

효율적인 8진 트리 검색 방식의 허프만 복호화기 구현

학생회원 우 광 희*, 김 구 용*, 정회원 한 현 수*, 차 형 태*

Efficient Huffman decoder using octal tree search algorithm

Kwanghee Woo*, Gooyong Kim* *Student Members*

Hernsoo Han*, Hyungtai Cha* *Regular Members*

요 약

여러 가지 압축알고리즘 중에서 허프만 부호화는 데이터의 통계적인 중복성을 제거하기 위한 방법으로 많이 사용되어왔다. 이제까지 허프만 테이블의 특성과 코드워드의 패턴에 따라 효율적으로 복호화할 수 있는 알고리즘이 많이 제안되었다. 본 논문에서는 8진 트리 검색 방법을 이용한 허프만 복호화 알고리즘을 제안하고 효율적인 하드웨어 구현 방법을 제안한다. 하드웨어의 메모리와 게이트 면적을 소형화하였고 빠른 검색을 위하여 알고리즘을 최적화하여 MPEG 오디오 복호화기 시스템에 적용할 수 있게 하였다.

ABSTRACT

Huffman coding which has been used in many data compression algorithms is popular data compression technique to reduce statistical redundancy. It has been proposed that Huffman decoding algorithm can decode efficiently using characteristics of Huffman tables and patterns of Huffman codewords. We propose a new Huffman decoding algorithm which used octal tree search technique, and present efficient hardware implementation method. This algorithm has a small logic area and memory space, and is optimized for high speed decoding. Proposed Huffman decoding algorithm can be applied for many multimedia systems such as MPEG audio decoder.

I. 서 론

허프만 부호화는 데이터의 통계적인 중복성 제거를 이용한 압축방법으로 가장 일반적으로 사용되고 있는 알고리즘이다. 멀티미디어 응용 시스템인 MPEG 오디오 알고리즘에도 다양한 형식의 데이터들을 여러 가지 알고리즘을 적용하여 최대한 단순화시키고 통계적 중복성을 제거하기 위해 허프만 부호화를 거쳐 압축된 비트열을 전송한다^{[1][2][3]}. 보통 한 개의 코드북만을 사용하는 기존의 방식들과는 달리 MPEG 오디오 시스템은 전 대역 오디오 신호를 처리하기 위해 많은 허프만 코드북을 사용한다. 복호화기의 실시간 처리를 위하여 빠른 허프만 복호화 알고리즘을 적용하여야 하고, 하드웨어를

소형화하기 위해 적은 면적, 적은 메모리를 사용하는 알고리즘이 요구되어진다.

본 논문의 구성은 II장에서 기존의 허프만 복호화기 알고리즘을 소개하고, III장에서 8진 트리 검색 방식의 허프만 복호화기 알고리즘을 제안하였다. IV장에서는 III장에서 제안된 허프만 복호화 알고리즘을 하드웨어로 구현하기 적합하게 최적화하여 MPEG 오디오 복호화기에 적용하였으며, V장에서 실험 및 결과로 제안된 알고리즘을 검증하였다. 마지막으로 VI장에서 결론으로 끝을 맺는다.

II. 기존의 허프만 복호화기 알고리즘

효율적인 허프만 복호화기를 구현하기 위해 지금

* 숭실대학교 정보통신전자공학부
논문번호: 00337-0828, 접수일자: 2000년 8월 28일

까지 다양한 방법들이 제안되었다. 이러한 방법들은 크게 2진 트리 검색 알고리즘(the binary tree search algorithm)^{[5][6]}과 비트 패러렐 알고리즘(the bit parallel algorithm)^{[7][8][9][10]}의 두 가지 방법으로 분류할 수 있다.

1. 2진 트리 검색 알고리즘

트리 검색 알고리즘은 허프만 복호화기의 가장 일반적인 방법으로 사용된다. 표 1에 간단한 허프만 테이블을 나타내었다.

표 1. 허프만 부호 테이블

Symbol	Codeword	Length	Symbol	Codeword	Length
S0	0	1	S6	11010	5
S1	1000	4	S7	11011	5
S2	1001	4	S8	11100	5
S3	1010	4	S9	11101	5
S4	1011	4	S10	11110	5
S5	1100	4	S11	11111	5

그림 1에 표 1의 허프만 테이블을 2진 트리로 나타내었다.

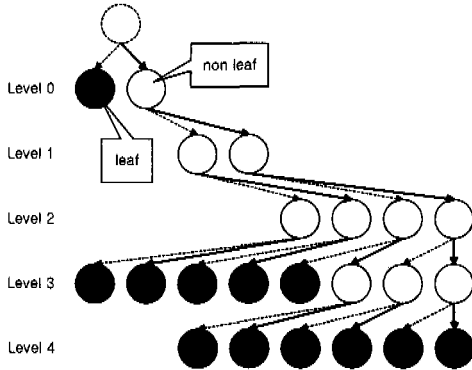


그림 1. 2진 트리 검색 허프만 복호화

그림 1의 2진 트리 검색에서 길이가 가장 긴 S6~S11을 검색하는데 5클럭 사이클이 소요된다. 이 알고리즘은 FSM(finite state machine)으로 구현할 수 있다^[5]. 이때, 각 노드의 수는 하드웨어 크기와 비례하게 되는데, 복호화할 심볼의 수가 많을 경우 각각의 상태를 메모리에 저장하여 구성한다.

이 알고리즘은 입력 비트를 1비트씩 순차적으로 입력받아 가변 길이의 코드워드와 비교하는 방법으로 복호주기가 일정하지 않고, 길이가 긴 코드워드에 대해서는 코드워드 길이 만큼 많은 시간을 필요

로 한다.

2. 비트 패러렐 알고리즘

하드웨어 구현 방법으로 수행속도가 가장 빠른 방법인 비트 패러렐 알고리즘은 코드워드의 일정한 패턴에 따라 로직을 구성하여 정합 시킴으로서 한 클럭 사이클에 하나의 코드워드를 복호화할 수 있다.

입력 비트열에서 허프만 테이블의 공통 비트 패턴이 인식되어지면 메모리 검색을 위한 첫 번째 인자로 사용되고 입력 비트열의 공통 비트 패턴을 뺀 나머지 코드워드는 메모리 검색을 위한 두 번째 인자로 사용되는 방식이다. 이러한 비트 패러렐 복호화 알고리즘은 어떤 비트열이 입력되더라도 복호화 주기를 일정하게 유지 할 수 있다는 장점이 있다.

그러나, 비트 패러렐 알고리즘은 모든 입력 비트열을 로직을 통해 분석해야 하므로 허프만 코드북의 크기가 크면 그에 비례하여 하드웨어의 크기도 커지게 되며 여러 개의 테이블을 사용하는 MPEG 오디오 복호화기와 같은 시스템에서는 로직의 증가에 따라 하드웨어가 크게 복잡해 질 수 있다.

III. 8진 트리 검색방식의 허프만 복호화기

MPEG 오디오 AAC의 허프만 복호화기는 11개의 스펙트럼 데이터와 1개의 스케일팩터 허프만 테이블을 가지고 있다. 전체 심볼의 개수는 1,362개이며, 코드워드는 최대 19비트로 구성되어있다^[1]. 이러한 MPEG 오디오의 허프만 복호화기를 2진 트리 검색 알고리즘으로 구현하면 메모리 검색 시간을 제외하더라도 가장 긴 코드워드의 길이가 19비트이므로 최대 19클럭이 소요되며, 비트 패러렐 알고리즘으로 구현하면 많은 수의 심볼을 로직으로 구성해야 하므로 하드웨어 크기가 커지게 된다.

본 논문에서는 MPEG 오디오와 같은 많은 수의 허프만 테이블을 가진 시스템에서 하드웨어와 메모리의 크기를 고려한 8진 트리 검색 방법을 제안한다. 트리의 Level을 줄여 검색 속도를 높이고, 입력 비트열의 코드워드를 메모리의 주소 인덱스로 사용하여 메모리 검색시간을 없애고 코드워드를 직접 저장하지 않으므로 메모리 효율을 높일 수 있다.

그림 2는 그림 1의 2진 트리를 변형하여 8진 트리로 구성하였다.

그림 1의 Level 1과 Level 2에서는 입력 비트열이 코드워드와 적중하는 경우가 없고, Level 3에서

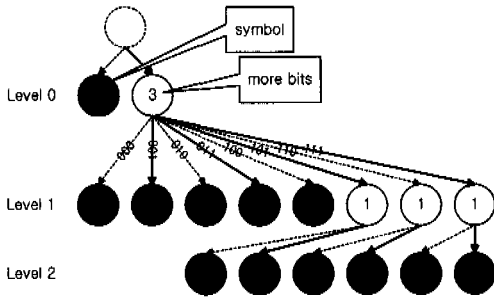


그림 2. 8진 트리 검색 허프만 복호화

부터 코드워드와 적중하는 경우가 발생한다. 그림 2에서는 Level 1과 Level 2를 생략하고 Level 0에서 3비트를 입력받아 다음 Level로 상태를 이동한다. 이때, 가지 노드에서는 다음 레벨로 이동하기 위해 필요한 입력 비트(more bits)를 표시 해두어야 한다. 그림 2의 Level 0과 Level 1에서 '3'과 '1'이 이것을 나타내고 있다.

트리의 각 노드를 메모리를 사용하여 구성하여야 하는데, 각각의 노드가 하나의 메모리 주소를 차지하게 된다. 또한, 레벨과 레벨의 구분을 위한 세그먼트(segment) 주소 기법을 사용하기 위해서 버려지는 공간이 생기는 데, 메모리의 효율적 구성을 위해 그림 3과 같이 트리의 모양을 변형시켜 사용하였다.

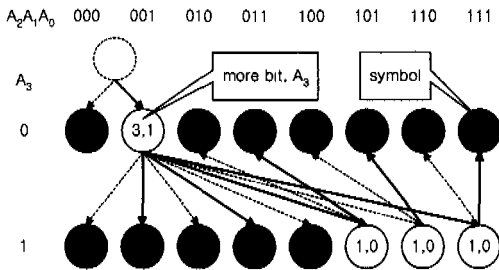


그림 3. 메모리 효율을 위한 8진 트리

그림 3은 그림 2에서 Level 2의 노드들을 Level 0 ($A_3=0$)으로 옮겨 사용하였다. 각 Level은 메모리의 상위주소로 사용되며, 이 주소를 세그먼트(segment)라고 한다. 입력 비트열 3비트는 메모리의 하위주소로 사용되며 오프셋(offset)이라고 한다. 그림 3에서 주소를 형성하는 $A_3A_2A_1A_0$ 에서 하위 3비트($A_2A_1A_0$)는 비트열의 입력으로서 오프셋 주소이고 상위 1비트(A_3)는 Level을 표시하기 위한 세그먼트 주소로 사용되었다. 가지 노드에는 더 필요한 비트 수 외에 다음 Level의 세그먼트 주소가 포함

되어 있어야 한다. 그림 3의 트리를 표 2와 같이 LUT(Look Up Table)로 구성하였다.

표 2. 그림 3의 LUT 구성

addr	table value		addr	table value	
	more bits	symbol/segment		more bits	symbol/segment
0000	0	S0	1000	0	S1
0001	3	1	1001	0	S2
0010	0	S6	1010	0	S3
0011	0	S7	1011	0	S4
0100	0	S8	1100	0	S5
0101	0	S9	1101	1	0
0110	0	S10	1110	1	0
0111	0	S11	1111	1	0

표 2의 LUT에 따라 복호화 알고리즘은 그림 4의 순서도와 같이 요약된다.

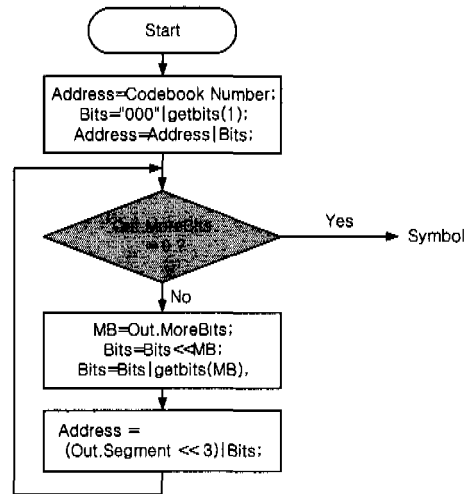


그림 4. 허프만 복호화 과정

먼저 허프만 복호화 모듈이 수행되면 코드북 번호에 따라 미리 결정된 세그먼트 주소 값이 결정되며 비트열로부터 한 비트를 읽어 오프셋 주소로 사용된다. 테이블의 출력이 심볼 값이면 해당 심볼을 출력한다. 그렇지 않으면 LUT에 저장된 세그먼트 주소와 더 필요한 비트 수만큼 읽어들인 비트열과 조합하여 다음 주소가 결정된다. 이렇게 해당 심볼이 출력 될 때까지 이 과정을 반복하여 수행한다.

IV. 허프만 복호화기의 하드웨어 구현

본 장에서는 제안된 알고리즘을 적용하여 MPEG 오디오의 Layer-3과 MPEG-2 오디오의 AAC에 사용되는 허프만 부호화기를 설계하였다. 설계도구는 알고리즘을 하드웨어로 구현하기에 적합한 하드웨어 개발 언어인 VHDL(Very High Speed Integrated Circuit Hardware Description Language)을 사용하여 설계하였고 synopsys를 통해 합성하였다.

MPEG 오디오의 허프만 부호화는 파서부에서 비트열을 분석하여 각 밴드별로 여러 개의 허프만 테이블중 하나를 선택하여 부호화된 비트열을 복호화한다. 하나의 허프만 테이블이 선택되면 세그먼트 주소는 선택된 LUT의 시작주소로 된다.

LUT의 시작 주소가 선택되면 처음 한 비트를 입력받아 코드워드가 일치할 때까지 최대 3비트씩 읽어오고, 읽어올 코드워드가 2비트 이하일 경우 이전의 비트들을 왼쪽쉬프트를 이용하여 전체 3비트로 구성하여 오프셋 주소로 사용한다. 코드워드가 일치하지 않을 경우 LUT 값은 다음 사이클에 더 읽어와야 할 비트수 2비트와 다음 사이클의 세그먼트 주소 8비트를 포함하고 있다.

그림 5에 8진 트리 검색 알고리즘을 구현한 허프만 부호화기의 블록도를 나타내었다. 구현된 허프만 부호화기는 LUT와 각 코드북의 시작 주소만 수정하여 MPEG 오디오 Layer-3과 MPEG-2 오디오 AAC의 부호화기에서 각각 사용할 수 있고, 공용으로 사용할 수도 있다.

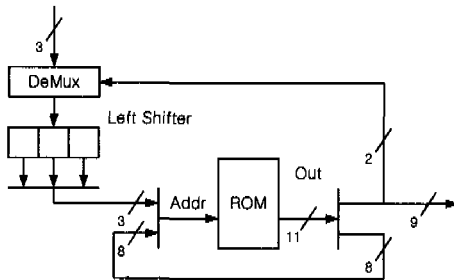


그림 5. 허프만 부호화기의 블록도

그림 6은 코드북 데이터가 저장된 ROM의 주소와 데이터의 형식을 나타내었다.

그림 6 a)의 주소 형식은 LUT 전체 2k word를 표현하기 위한 11비트 주소로서 8비트의 세그먼트 주소와 3비트의 오프셋 주소로 구성되어 있다. 그림 6 b)의 데이터 형식은 비트열이 코드워드와 일치할 경우를 나타낸다. 상위 2비트는 "00"으로 표현되

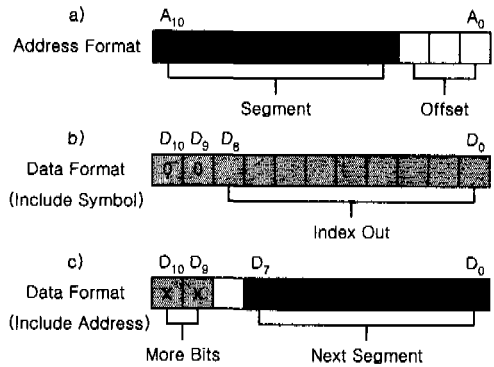


그림 6. 테이블의 주소(a)와 데이터^{b)}의 형식

어있고, 하위 9비트는 해당 심볼 값으로 구성되어 있다. MPEG-2 오디오 AAC의 허프만 코드북은 한 코드북에 최대 289개의 심볼을 가지고 있으므로 이것을 표현하기 위해 9비트로 구성하였다. 그림 6 c)의 데이터 형식은 비트열이 코드워드와 일치하지 않고 입력 비트열이 더 필요할 경우를 나타낸다. 상위 2비트는 다음 클럭 사이클에서 필요한 입력 비트열의 수가 표현되어 있고 하위 8비트는 다음 클럭 사이클의 세그먼트 주소로 구성 되어있다.

V. 실험 및 결과

이 장에서는 제안된 8진 트리 검색 방식의 허프만 부호화 알고리즘을 사용하여 MPEG 오디오 부호화기에 적용하여 검증하였다.

비트열은 MPEG 그룹에서 제공하는 MPEG-2 AAC LC 프로파일로 압축된 1분 이상의 곡을 선택하였다. 그림 7에 순차 검색 방법과 본 논문에서 제안된 8진 트리 검색 방식의 클럭 소요량을 나타내었다. 순차 검색 방식은 MPEG 그룹에서 제공한 VM(Verification Model)코드에서 사용된 방식으로 가장 작은 코드워드부터 시작하여 같은 길이의 코드워드 단위로 메모리와 비교하여 검색하는 방법이다.

실험 결과, 순차 검색 방식은 한 샘플을 부호화하기 위하여 비트열을 읽는 시간과 메모리 검색을 위한 클럭이 평균 17회 소요되었고, 8진 트리 검색 방식은 평균 2.48회 소요되었다. 특히, 최대 300 클럭 사이클까지 소요되는 것을 6클럭 사이클로 줄일 수 있었고, 샘플들의 부호화 주기 편차가 현저히 줄어 쉽게 클럭을 제어 할 수 있다.

Layer 3과 AAC에 사용된 허프만 부호화기의 메모리 용량 및 부호화 수행 속도를 표 3에 나타내었다.

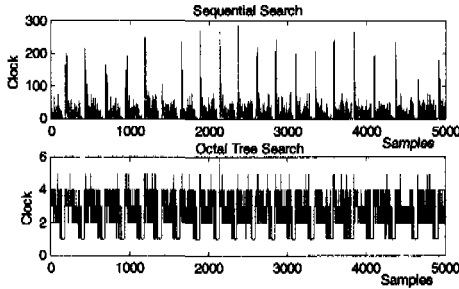


그림 7. 순차 검색 방식과 8진 트리 검색 방식의 비교

표 3. MPEG 오디오 허프만 복호화기의 성능

	Layer 3	AAC
코드북 개수	15개	12개
전체 심볼 수	1,394개	1,362개
코드워드 최대길이	19 bits	19 bits
복호화 속도	최대 8클럭	최대 8클럭
메모리 크기	2.4k word	2k word
하드웨어 크기	400 게이트(공통)	

적은 면적의 하드웨어를 사용하여 메모리의 테이블 값만 바꾸어주면 Layer 3과 AAC의 허프만 복호화기로 사용할 수 있다. 또한, 두 개의 메모리를 함께 사용하여 코드북의 시작 주소 값을 각각 지정하면 Layer 3과 AAC의 공용 허프만 복호화기로도 사용할 수 있다.

V. 결론

본 논문은 많은 수의 허프만 테이블을 가진 시스템에서 적용할 수 있는 허프만 복호화 알고리즘을 제안하였다. 8진 트리 검색 방법을 사용하여 검색 속도를 높였고, 적은 면적의 로직 게이트와 적은 메모리를 사용하였고 MPEG 오디오 복호화기의 허프만 복호화기에 적용하여 검증하였다.

입력받은 비트열 메모리의 주소로 사용하므로 메모리 검색을 한번에 할 수 있고, 코드워드를 저장하지 않아 코드북을 저장하기 위한 메모리의 총 비트 수를 절반으로 줄일 수 있었다.

참고 문헌

[1] ISO/IEC 13818-7, *Generic Coding of Moving Pictures and Associated Audio Information - Part 7 : Advanced Audio Coding*, 1997.

[2] ISO/IEC 14496-3, *Information Technology - Coding of Audiovisual Objects - Part 3 : Audio, Subpart 4 : T/F Coding*, 1998.

[3] ISO/IEC 11172-3, *Information Technology - Generic Coding of Moving Pictures and Associated Audio : Part 3 : Audio*, 1992.

[4] ISO/IEC JTC1/SC29/WG11 N2005, *Revised Report on Complexity of MPEG-2 AAC Tools*, Feb. 1998.

[5] Vikram Iyengar, Krishnendu Chakrabarty "An efficient finite-state machine implementation of Huffman decoders," *Information Processing Letters*, V.64 N.6, 271-275, Dec. 1997.

[6] Hong-Chung Chen, Yue-Li Wang, Yu-Feng Lan, "A memory-efficient and fast Huffman decoding algorithm," *Information Processing Letters*, V.69, pp 119-122, 1999.

[7] Seung Bae Choi, Moon Ho Lee "A Fast Huffman Decoder via Pattern Matching," *ISPACS*, pp 134-138, 1994.

[8] Reza Hashemian, "Memory Efficient and High-Speed Search Huffman Coding," *IEEE Transactions on Communications*, V.43 N.10, Oct. 1995.

[9] Reza Hashemian, "Efficient variable - length coding under an assigned maximum code-length constraint," *Proceedings of the IEEE International Symposium on Circuits and Systems - Volume 2*, May. 1996.

[10] Park S, Cho H, Cha JJ, "High speed search and an area efficient Huffman decoder," *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences*, V.E82 -A N.6, pp 1017-20, Jun. 1999.

우 광 회(Kwanghee Woo)

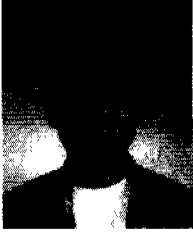
학생회원



1998년 : 숭실대학교
전자공학과 졸업(공학사)
1999년~현재 : 숭실대학교
전자공학과 석사과정
<주관심 분야> 통신 및 신호처리,
MPEG 오디오, 오디오 코
딩, ASIC 설계

김 구 용(Gooyong Kim)

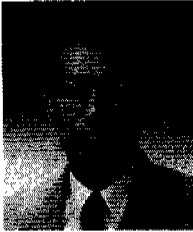
학생회원



2000년 : 숭실대학교
전자공학과 졸업(공학사)
2000년~현재 : 숭실대학교
전자공학과 석사과정
<주관심 분야> 신호처리, MPEG
오디오, ASIC 설계, DSP

한 헌 수(Hernsoo Han)

정회원

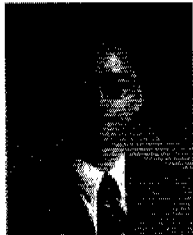


1981년 : 숭실대학교
전자공학과 졸업(공학사)
1983년 : 연세대학교
전자공학과 졸업
(공학석사)
1991년 : University of South-
ern California(공학박사)

1992년~현재 : 숭실대학교 부교수
<주관심 분야> 자동화 시스템, 자료융합, 물체인식

차 형 태(Hyungtai Cha)

정회원



1985년 : 숭실대학교
전자공학과 졸업(공학사)
1988년 : The University of
Pittsburgh(공학석사)
1993년 : The University of
Pittsburgh(공학박사)

1993~1996 : 삼성전자 신호처리 연구소 선임연구원
1996~현재 : 숭실대학교 조교수
<주관심 분야> Audio / Video Coding, Morphology,
신호처리, ASIC 설계