

논문 2007-44SD-12-15

효율적인 H.264/AVC 엔트로피 복호기 설계

(An Efficient H.264/AVC Entropy Decoder Design)

문 전 학*, 이 성 수**

(Jeonhak Moon and Seongsoo Lee)

요 약

본 논문에서는 메모리 공정이 필요 없고 내장 프로세서를 사용하지 않는 H.264/AVC 엔트로피 복호기를 제안한다. 기존에 발표된 H.264/AVC 엔트로피 복호기의 경우 상당수의 연구가 내부의 ROM 또는 RAM이 필요하기 때문에 일반적인 디지털 로직 공정에서 구현이 어렵다. 또한 상당수의 연구가 비트열 처리를 위하여 내장 프로세서를 사용하기 때문에 면적이 크고 전력 소모가 많은 단점을 가지고 있다. 본 논문에서는 내장 프로세서를 사용하지 않는 H.264/AVC Hardwired 엔트로피 복호기를 제안함으로써 데이터 처리 속도를 증가 시키고 전력 소모를 줄인다. 또한 CAVLC 복호기에서 복호 시에 이용되는 룩업 테이블 및 저장 공간을 최적화하고 내장 메모리를 사용하지 않는 구조를 제안함으로써, 기존 연구에 비해 하드웨어 크기를 줄이고 ROM 또는 RAM이 지원되지 않는 디지털 로직 제조 공정에서도 쉽게 구현이 가능하다. 설계된 엔트로피 복호기는 H.264/AVC 비디오 복호기의 일부로 내장되어 전체 시스템에서 동작하는 것을 검증하였다. TSMC 90nm 공정으로 합성한 결과 최대동작주파수는 125MHz이며, QCIF, CIF, QVGA 영상을 지원할 뿐만 아니라 nC 레지스터 등 약간의 수정을 통해서 VGA 영상도 지원이 가능하다.

Abstract

This paper proposes a H.264/AVC entropy decoder without embedded processor nor memory fabrication process. Many researches on H.264/AVC entropy decoders require ROM or RAM fabrication process, which is difficult to be implemented in general digital logic fabrication process. Furthermore, many researches require embedded processors for bitstream manipulation, which increases area and power consumption. This paper proposes hardwired H.264/AVC entropy decoder without embedded processor, which improves data processing speed and reduces power consumption. Furthermore, its CAVLC decoder optimizes lookup table and internal buffer without embedded memory, which reduces hardware size and can be implemented in general digital logic fabrication process without ROM or RAM fabrication process. Designed entropy decoder was embedded in H.264/AVC video decoder, and it was verified to operate correctly in the system. Synthesized in TSMC 90nm fabrication process, its maximum operation frequency is 125MHz. It supports QCIF, CIF, and QVGA image format. Under slight modification of nC register and other blocks, it also support VGA image format.

Keywords : H.264/AVC, Entropy Decoder, Exp-Golomb, CAVLC, Variable Length Coding

I. 서 론

ITU-T와 ISO/IEC가 공동 제정한 동영상 압축 표준 H.264/AVC는 다양한 네트워크 환경에 쉽게 부응할 수 있는 유연성과 동영상의 부호화 효율성 측면에서

MPEG-2, MPEG-4 (Part 2) 등 기존 기술표준들에 비해 많은 진보가 있었다^[1]. 그러나 높은 압축 성능과 고화질의 장점을 가지는 반면, 부호기 및 복호기의 복잡도가 훨씬 증가하는 단점이 있다. 또한 요즘 이동통신 기기 및 휴대용 멀티미디어 기기의 다기능화 요구로 인한 내장 프로세서의 과부하 및 배터리 소모가 많은 것도 문제이다.

본 논문에서는 이러한 문제점을 해결하기 위해 프로세서를 사용하지 않는 엔트로피 복호기를 제안한다. H.264/AVC 엔트로피 복호기는 NAL (Network Abstraction Layer) 헤더와 RBSP (Raw Byte

* 학생회원, ** 평생회원, 숭실대학교 정보통신전자공학부

(School of Electronic Engineering, Soongsil University)

* 본 논문은 숭실대학교 교내 연구비 지원으로 이루어졌음

접수일자: 2007년9월17일, 수정완료일: 2007년11월21일

Sequence Payload)로 구성된 비트열 데이터를 받아서 영상 복원에 필요한 데이터를 얻는다. 그 중에서 높은 압축 효율을 요구하는 DCT계수의 복호화는 CAVLC (Context-Adaptive Variable Length Coding) 방식을 사용하며, 그 외 구문요소의 복호화는 Exp-Golomb (Exponential Golomb) 방식을 사용한다. 기존의 엔트로피 복호기는 CAVLC 복호기만 하드웨어로 구현하고, Exp-Golomb 복호화와 남은 비트열 데이터의 복호화는 프로세서에서 처리한다. 그러나 본 논문에서는 위의 모든 것을 프로세서 없이 처리하는 Hardwired 엔트로피 복호기를 제안한다. Hardwired 엔트로피 복호기는 휴대용 멀티미디어 기기 등에서 프로세서의 부하를 줄여주어 데이터 처리 속도를 증가시키며 전력 소모를 줄일 수 있다. 또한 엔트로피 복호기가 프로세서를 대신 하여 H.264/AVC 복호기 전체를 제어함으로써 시스템의 복잡도 및 면적이 줄어들고 다른 IP 블록과의 연계를 통한 설계 및 디버깅이 쉬어진다.

특히 CAVLC 복호기는 복호화 순서에 따라 달리 선택되는 룩업 테이블과 복호된 이전 블록의 정보를 다음 블록 복호화에 참조하기 위한 저장 공간의 확보로 인해 로직 크기를 매우 증가시킨다. 본 논문에서는 룩업 테이블 및 저장 공간의 최적화 기법을 사용하여 하드웨어 크기를 최적화 하는 방법도 제시한다.

II. 엔트로피 복호화

엔트로피 복호기는 비트열을 입력받아 구문요소 (Syntax Element)의 종류에 따라 복호화 방법을 결정하여 해당하는 복호기에 데이터를 보내준다. 표 1을 보면 대부분의 구문요소는 Exp-Golomb 복호화를 하며, 영상 정보 데이터는 CAVLC 복호화를 한다.

1. Exp-Golomb 복호화

Exp-Golomb 부호는 표 2의 코드워드(Codeword)처럼 규칙적이다. 처음 '1'이 나올 때까지의 '0'의 개수(M)와 '1' 뒷부분에 대한 정보(INFO)를 얻어낸 뒤 식(1)을 이용하여 code_num 값을 구한다. 그리고 표 2와 같이 code_num 값에서 ue(unsigned exp-golomb code), se(signed exp-golomb code), me(mapped exp-golomb code), te(truncated exp-golomb code)의 4가지 방법에 따라 구문요소 값을 구한다. 같은 code_num 값이라도 구하는 방법에 따라 구문요소 값은 달라진다.

표 1. 구문요소에 대한 부호화 방식

Table 1. Coding method for syntax element.

Syntax Element	Description	Coding
SPS/PPS	Headers and parameters	Exp-Golomb Coding
MB type	Prediction method for each coded MB	
Coded Block Pattern	Indicates which blocks within a MB contain coded coefficients	
Quantization Parameter(QP)	Transmitted as a delta value from the previous value of QP	
Reference frame index	Identify reference frame(s) for inter prediction	
Motion Vector(MV)	Transmitted as a difference from predicted MV	CAVLC Coding
Residual data	Coefficient data for each 4x4 or 2x2 block	

표 2. Exp-Golomb 복호화 예제

Table 2. Exp-Golomb decoding example.

Codeword	code_num	Syntax Element	
		ue(v)	se(v)
1	0	0	0
010	1	1	1
011	2	2	-1
00100	3	3	2
00101	4	4	-2
00110	5	5	3
00111	6	6	-3
...	k	k	$(-1)^{k+1} \text{Ceil}(k/2)$

$$\text{code_num} = 2^M + \text{INFO} - 1 \quad (1)$$

2. CAVLC 복호화

CAVLC 부호화는 DCT계수를 지그재그 스캐닝한 후 5가지의 구문요소(coeff_token, sign of T1s, level, total_zeros, run_before)로 변환한 뒤, 서로 다른 룩업 테이블을 참조하여 코드워드로 부호화한다.

coeff_token: '0'이 아닌 계수의 개수 (TotalCoeff)와 ±1의 개수 (T1s)의 전체 숫자

Sign of T1s: trailing ±1의 부호 (0이면+, 1이면-)

Level: T1s을 제외한 '0'이 아닌 계수들의 값

total_zeros: DC와 마지막 계수 사이 '0'인 계수들의 개수

run_before: 각 '0'이 아닌 계수들 사이에 있는 '0'인 계수들의 개수

복호화 시, 이전에 복호화 된 위쪽과 왼쪽 블록 TotalCoeff의 nC (평균값)에 따라 룩업 테이블을 선택하고, 입력된 코드워드에 맞게 TotalCoeff, T1s 값을 복

호한다. 또한 값(0이면 +, 1이면 -)에 맞게 T1(±1)값을 복호한다. Level 값은 각 조건에 따라 맞게 복호하며, total_zeros, run_before는 룩업 테이블을 이용하여 복호한다. 이렇게 5가지의 구문요소를 얻은 뒤 DCT계수를 결정한다.

III. 제안하는 엔트로피 복호기

본 절에서는 프로세서를 사용하지 않고 비트열을 처리하는 Hardwired 엔트로피 복호기를 제안한다. 복호기는 크게 NAL 복호기, Exp-Golomb 복호기, CAVLC 복호기로 구성되며 NAL 복호기와 Exp-Golomb 복호기를 프로세서에서 처리하지 않고 CAVLC 복호기와 같이 Hardwired로 설계함으로써 프로세서의 부하를 줄여주고 데이터 처리 속도를 증가시키며, 전력 소모를 줄인다. 또한 CAVLC 복호기는 룩업 테이블 및 저장 공간의 최적화 기법을 통해 하드웨어 크기를 줄인다.

1. 엔트로피 복호기

제안하는 엔트로피 복호기는 그림 1과 같이 NAL 복호기, Exp-Golomb 복호기, CAVLC 복호기, Input Buffer, ReOrder 등으로 구성된다. NAL 복호기는 비트열을 처리하며 프로세서 대신 엔트로피 복호기를 제어한다. NAL 복호기는 현재 복호되는 코드워드의 종류를 판단한 후, 영상 데이터에 해당하는 비트열은 CAVLC 복호기로 보내고, 그 외 구문요소에 해당하는 비트열은 Exp-Golomb 복호기로 보낸다. CAVLC 복호기가 얻은 영상 데이터는 ReOrder를 거쳐 IQ/IT 블록으로 전송되며, Exp-Golomb 복호기가 얻은 구문요소는 MC, INTRA, IQ/IT, DF 등 다른 블록으로 전송된다.

엔트로피 복호기가 프로세서 대신 H.264/AVC 복호기 전체를 제어함으로써 시스템의 복잡도 및 면적이 줄

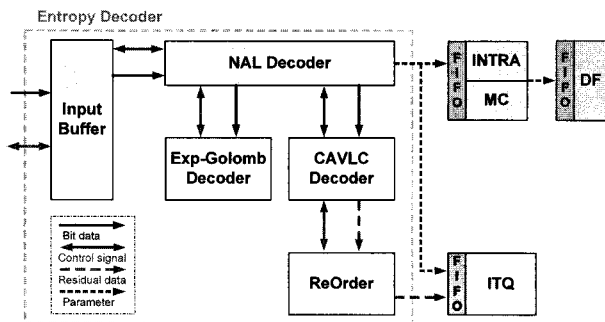


그림 1. 제안하는 엔트로피 복호기의 구조
Fig. 1. Architecture of the proposed entropy decoder.

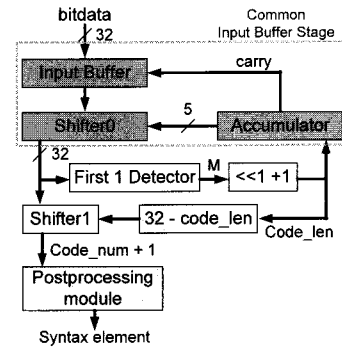


그림 2 제안하는 Exp-Golomb 복호기
Fig. 2. Architecture of the proposed exp-golomb decoder.

어들과 다른 IP 블록과의 연계를 통한 설계 및 디버깅이 쉬워진다. 또한 프로세서 없이 동작하는 단일 H.264/AVC 복호기를 구현할 수 있다. 엔트로피 복호기는 다른 IP 블록의 동작 상태에 영향을 받지 않도록 FIFO를 통해서 데이터를 전달한다. FIFO의 상태가 'Full'이면 다른 IP 블록이 더 이상 데이터를 받아서 저장해 둘 여유가 없다는 것을 의미하므로 엔트로피 복호기는 대기 상태가 된다.

2. Exp-Golomb 복호기

일반적인 복호기^[2]는 비트열을 입력받아 데이터 복호에 필요한 길이만큼 처리한 뒤, 다음 데이터 복호를 위해 그 길이만큼 비트열을 이동시키는 부분을 가진다. NAL 복호기, Exp-Golomb 복호기, CAVLC 복호기도 이러한 입력단을 필요로 한다. 그러나 제안하는 엔트로피 복호기는 입력단을 하나만 설계하여 같이 사용함으로써 불필요한 면적을 줄였다. 제안하는 Exp-Golomb 복호기는 그림 2와 같으며, 입력된 비트열을 code_num으로 복호화해주는 First 1 Detector와 Combination 로직, 그리고 code_num을 구문요소로 복호화 해주는 Postprocessing Module 등으로 구성되어 있다.

3. CAVLC 복호기

제안하는 CAVLC 복호기의 구조는 그림 3과 같다. CAVLC 복호기도 엔트로피 복호기의 입력단을 공동으로 사용함으로써 불필요한 면적을 줄였다. CAVLC 복호기는 coeff_token, T1s, Level, total_zeros, run_before 5개의 구문 요소를 처리하는 복호기와 내부 버퍼 (nC Register, Level FIFO, Run_Before Stack), 제어기 블록으로 이루어져 있다. 특히 coeff_token과 total_zeros 복호 시 참조해야할 룩업 테이블의 크기가 매우 크고 비

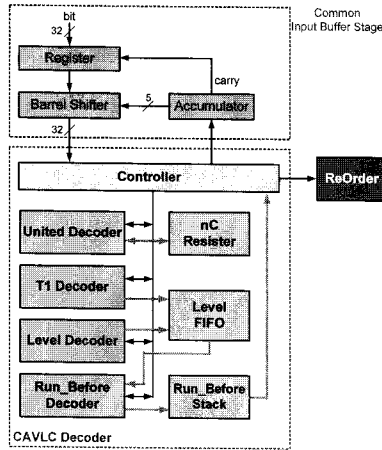


그림 3. 제안하는 CAVLC 복호기
Fig. 3. Architecture of the proposed CAVLC decoder.

슷한 유형의 인덱스가 많기 때문에 두 복호기를 통합하여 설계하였다.

가. 룩업 테이블의 최적화

CAVLC 복호화는 각 단계별로 여러 개의 룩업 테이블을 참조하여 코드워드를 구문요소로 복호한다. 룩업 테이블의 인덱스들은 매우 길고 그 양 또한 방대하다. 이를 로직으로 구현한다면 그 크기 또한 상당해진다. 그래서 테이블의 크기를 줄이고자 다음과 같은 기법들을 사용한다. 코드워드를 처음 '1'이 나올 때까지의 '0'의 개수(Prefix Length)와 그 외 부분(Suffix Info.)으로 나누어서 이를 룩업 테이블의 인덱스로 이용하는 것이다. 표 3은 이를 적용한 예를 보여준다. 또한 coeff_token 룩업 테이블에서 '8<=nC' 범위의 코드워드는 6bit로 고정된 값을 가지며 아래 식과 같은 규칙성을 가지고 있다. 이렇게 간단한 식으로 인덱스를 나타냄으로써 테이블의 크기를 크게 줄일 수 있다.

$$\text{TrailingOnes} = \text{Input Bit}[1:0] \quad (2)$$

표 3. 룩업 테이블의 최적화 방법 ([1]에서 Table 9-5의 일부를 예시)

Table 3. Optimization of lookup table (Part of Table 9-5 in [1] is shown.)

Trailing Ones	Total Coeff	Before	After	
		Codeword (0<=nC<2)	Prefix Length	Suffix Info.
3	4	0000 11	4	1
3	5	0000 100		00
2	3	0000 101		01
3	6	00000 100	5	00
2	4	00000 101		01
1	3	00000 110		10
0	2	00000 111		11

표 4. 룩업 테이블의 최적화

Table 4. Optimization of lookup tables.

	Codeword 비트수	Suffix Info. 비트수(1)	Prefix Length 비트수(2)	(1)+(2) 비트수
coeff_token	2248	491	800	1291
total_zeros	544	109	576	685

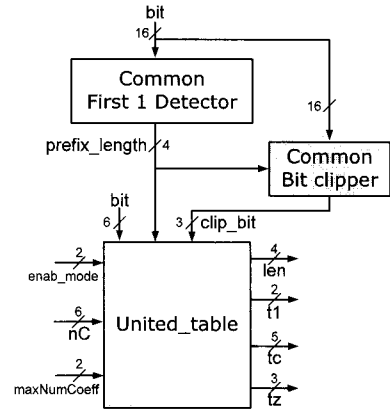


그림 4. 제안하는 통합 복호기의 구조
Fig. 4. Architecture of the proposed united decoder.

$$\text{TotalCoeff} = \text{Input Bit}[5:2] + 1 \quad (3)$$

위의 두 방법을 사용하여 표 4에서 보는 것과 같이 coeff_token과 total_zeros 룩업 테이블을 최적화하였다. 'Suffix Info.'의 비트수만 보면 룩업 테이블의 크기가 크게 줄어들었다. 하지만 로직으로 구현 시 'Prefix Length' 값도 참조해야 하기 때문에 비트수로 계산한 값을 포함한 비트수는 증가하게 된다. 그러나 두 개의 룩업 테이블을 통합하여 하나의 복호기로 설계함으로써 하드웨어의 크기는 크게 감소한다. First 1 Detector와 Bit Clipper를 하나씩만 사용하여 효율성을 높인다. 또한 여러 개의 'Suffix Info.'가 동일한 'Prefix Length' 값을 공유하고 3비트 이내의 값을 중복으로 가지기 때문에 최적화 과정을 거쳐 합성함으로써 로직의 크기가 크게 감소한다. 기존의 연구^[3]에서는 여러 룩업 테이블을 각각 ROM으로 구현하여 메모리의 크기는 줄일 수 있으나, ROM의 개수가 증가함에 따라 ROM 내부 셀보다는 디코더 등 부가적인 면적이 크게 증가하여 하드웨어 면적이 증가하는 단점이 있다. 그림 4는 두 복호기를 통합한 복호기이다.

나. nC 레지스터

CAVLC 알고리즘은 이전에 복호된 정보를 현재 복호에 참조하는 방식을 택하고 있다. 이 방식은 블록간의 유사성을 이용함으로써 높은 압축 효율을 가지나,

참조할 정보를 저장할 공간이 필요하다. 블록의 개수의 개수를 복호화 할 때, 이전에 복호화 된 위쪽과 왼쪽 블록 TotalCoeff의 nC (평균값)가 룩업 테이블의 인덱스로써 사용된다. 참조되는 블록의 정보를 저장하기 위한 공간이 필요하며, CIF (352x288) 영상에서는 9504 byte의 저장 공간이 필요하다^[4]. 그러나 지상파 DMB 표준에서는 ASO (Arbitrary Slice Order)를 지원하지 않아 매크로블록 부호화 순서가 주사선 방식 (Raster Order)이므로 저장된 블록의 정보는 두 번 참조된 이후 더 이상 참조되지 않는다.

본 논문에서는 이러한 특징을 이용하여 모든 블록에 대해 저장 공간을 할당하지 않고, 같은 열의 블록끼리 저장 공간을 재사용할 수 있는 구조를 제시함으로써 저장 공간을 줄였다. 그림 5에서 보듯이 기본적으로 같은 열의 블록들은 하나의 공통 저장 공간을 사용하고, 그 중에서 특정한 위치의 블록들 (1,5,7,9,13,19,23)만 따로 저장 공간을 사용하는 기법을 사용하였다. 이렇게 저장 공간을 구분하는 이유는 다음과 같다. 현재 복호화 중인 매크로블록에서 블록 1번은 3번과 4번 복호화 시 참조된다. 블록의 복호화 순서 (1~25)에 따라 3번의 복호화 후에 공통 저장 공간에 있던 1번의 값은 3번의 값으로 바뀌게 된다. 그러면 4번 복호화 시 필요한 1번의 값은 사라지게 되어 다음 블록의 복호화가 불가능하다. 이를 방지하기 위해 특정한 위치의 블록들은 따로 저장 공간을 둔다. 또한 모든 매크로블록 마다 따로 저장 공간을 두지 않고 공통 저장 공간을 재사용함으로써 불필요한 중복을 없앤다.

CIF 영상에서는 매크로블록 1개당 8개 (Luma:4, Cb:2, Cr:2)의 저장 공간이 필요하며, 가로 22개의 매크로블록이 있다. 또한 특정한 위치의 블록 7개 (1,5,7,9,13,19,23)의 저장 공간이 필요하며, TotalCoeff 값 (0~16)을 저장하기 위해서는 5bit의 공간이 필요하다. 즉, CIF 영상에서는 5bit 크기의 저장 공간 183개가

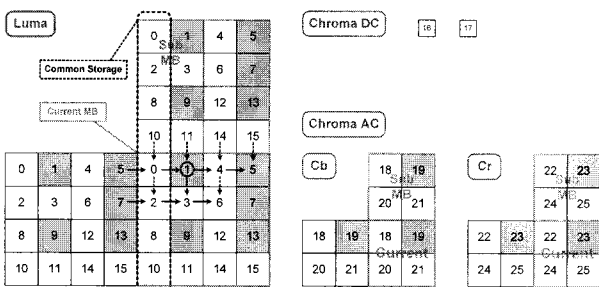


그림 5. nC 레지스터의 원리
Fig. 5. Principle of nC register.

표 5. 참조 값 저장 공간 요구량
Table 5. Reference value storage requirements.

	[3]	소프트웨어 기반[4]	Proposed
CIF(352x288)	320 byte	9054 byte	915 bit
비 고	RAM만 계산 Register 제외	RAM	Register

필요하다. 표 5는 CIF 영상을 CAVLC 복호화 할 때 필요한 참조 값 저장 공간 요구량을 다른 기법들과 비교하여 보여주고 있다. nC 레지스터는 CIF 크기보다 작은 QCIF(176x144), QGVA(320x240) 영상도 지원한다.

IV. 하드웨어 구현

설계된 H.264/AVC 엔트로피 복호기는 영상을 부호화한 비트열을 직접 입력으로 받아 복호화한 데이터를 다른 IP 블록들과 주고받는 것을 통해 검증하였다. 표 6은 IDEC (IC Design Education Center)의 설계 도구 지원을 받아 엔트로피 복호기를 TSMC 90nm 공정으로 108MHz로 합성한 결과이며, 최적화된 룩업 테이블과 nC 레지스터를 ROM과 RAM이 아닌 오직 로직으로만 구현하였다. 복호기의 최대동작주파수는 125MHz 이며, CIF, QCIF, QVGA의 다양한 영상 크기에 대해서 검증하였다. 또한 nC 레지스터 등 약간의 수정을 통해서 VGA(640x480) 영상도 지원 가능하다.

표 7은 CAVLC 복호기의 면적을 다른 복호기와 비교한 것이다. 제안하는 복호기는 룩업 테이블과 nC 레지스터를 메모리의 사용 없이 오직 로직으로만 구현하였다. 따라서 ROM 또는 RAM이 지원되지 않는 디지털 로직 제조 공정에서도 쉽게 구현이 가능하다.

표 6. 엔트로피 복호기의 합성 결과
Table 6. Area of entropy decoder.

	Gate Counts (2-NAND equivalent)
엔트로피 복호기	75677
CAVLC 복호기	24517

표 7. CAVLC 복호기의 면적 비교
Table 7. Comparison of CAVLC decoder area.

	[3]	[5]	Proposed
Gate Counts	11700	17202	24517
메모리 요구량	6400 (bits)	5120 (bits)	N/A

V. 결 론

기존의 H.264/AVC 엔트로피 복호기는 CAVLC 복호

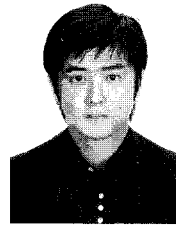
기만 하드웨어로 구현하고, 나머지 비트열 데이터의 처리부분과 Exp-Golomb 복호화는 프로세서에서 처리를 하기 때문에 내장 프로세서의 과부하 및 배터리 소모가 많은 것이 문제였다. 그러나 본 논문에서는 프로세서를 사용하지 않고 전체 복호화를 처리하는 Hardwired 엔트로피 복호기를 제안하였다. 제안한 엔트로피 복호기는 프로세서의 부하를 줄여주어 데이터 처리 속도를 증가 시키며, 전력 소모를 줄일 수 있다. 또한 CAVLC 복호에 이용되는 룩업 테이블 및 저장 공간을 최적화하여 하드웨어 크기를 최적화하였다.

설계한 CAVLC 복호기의 로직 크기는 이전 논문의 구현 사례와 비교할 때 더 작았으며, Hardwired 엔트로피 복호기 역시 프로세서를 사용하지 않았기 때문에 하드웨어 크기 면에서 매우 효율적이다. 또한 단일 H.264/AVC 복호기 설계 시, 설계된 엔트로피 복호기의 별다른 수정 없이 재사용할 수 있어 매우 효율적인 구조이다.

참 고 문 헌

- [1] Joint Video Team, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification", ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, May 2003.
- [2] W. Di, G. Wen, H. Mingzeng and J. Zhenzhou, "An Exp-Golomb Encoder and Decoder Architecture for JVT/AVS", Proceedings of International Conference on ASIC, vol. 2, pp. 910-913, Oct. 2003.
- [3] Dae-Joon Lee and Yong-Jin Jeong, "VLSI architecture design of CAVLC entropy encoder/decoder for H.264/AVC", Journal of Korea Information and Communications Society, vol. 30, no. 5C, pp. 371-381, 2005년 5월.
- [4] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro, "H.264/AVC baseline profile decoder complexity analysis", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no 7, pp. 704-716, Jul. 2003.
- [5] M. Alle, J Biswas and S. Nandy, "High Performance VLSI Architecture Design for H.264 CAVLC Decoder", Proceedings of International Conference on Application-Specific Systems, Architectures, and Processors, pp. 317-322, Sep. 2006.

저 자 소 개



문 전 학(학생회원)
 2005년 숭실대학교 정보통신전자공학부 학사
 2007년 숭실대학교 정보통신공학과 석사 재학
 <주관심분야 : SoC, Video Codec Design, Low Power Design>



이 성 수(평생회원)
 1991년 서울대학교 전자공학과 학사
 1993년 서울대학교 전자공학과 석사
 1998년 서울대학교 전기공학부 박사
 1998년~2000년 University of Tokyo Research Associate
 2000년~2002년 이화여자대학교 정보통신학과 연구교수
 2002년~현재 숭실대학교 정보통신전자공학부 조교수
 <주관심분야 : 바이오칩 설계, H.264 설계, 저전력 설계, SiP 설계>