

---

# 고성능 H.264 인코더를 위한 CABAC 하드웨어 설계

명제진\* · 류광기\*\*

The Hardware Design of CABAC for High Performance H.264 Encoder

Jejin Myoung\* · Kwangki Ryoo\*\*

---

본 논문은 2011년도 한밭대학교 교내학술연구비의 지원결과로 CAD Tool은 IDEC의 지원을 받았음

---

## 요 약

본 논문에서는 공통 연산기(Common Operation Unit)를 이용한 CABAC의 이진 산술 부호화기를 제안한다. 제안한 공통 연산기는 모드에 상관없이 하나의 공통 연산기를 이용하여 산술 부호화 및 재정규화를 수행하는 이진 산술 부호화기의 하드웨어 구조를 단순하게 구현할 수 있다. 제안하는 CABAC의 이진 산술 부호화기는 Context RAM, Context Updater, Common Operation Unit, Bit-Gen으로 구성되며 매 클럭당 하나의 심볼이 부호화될 수 있는 4단 파이프라인으로 구성하였다. 제안한 CABAC의 이진 산술 부호화기는 기존 CABAC의 이진 산술 부호화기와 비교하여 게이트 수는 최대 47% 감소하였고, 동작 주파수는 최대 19% 성능이 향상됨을 확인하였다.

## ABSTRACT

This paper proposes a binary arithmetic encoder of CABAC using a Common Operation Unit including the three modes. The binary arithmetic encoder performing arithmetic encoding and renormalizer can be simply implemented into a hardware architecture since the COU is used regardless of the modes. The proposed binary arithmetic encoder of CABAC includes Context RAM, Context Updater, Common Operation Unit and Bit-Gen. The architecture consists of 4-stage pipeline operating one symbol for each clock cycle. The area of proposed binary arithmetic encoder of CABAC is reduced up to 47%, the performance of proposed binary arithmetic encoder of CABAC is 19% higher than the previous architecture.

## 키워드

H.264/AVC, CABAC, 엔트로피 코딩, 비디오 코딩, 이진 산술 코딩

## Key word

H.264/AVC, CABAC, Entropy Coding, Video Coding, Binary Arithmetic Coding

---

\* 준희원 : 한밭대학교 정보통신공학과

접수일자 : 2011. 12. 24

\*\* 종신희원 : 한밭대학교 정보통신공학과(교신저자, kkryoo@hanbat.ac.kr)

심사완료일자 : 2012. 02. 25

**Open Access** <http://dx.doi.org/10.6109/jkiice.2012.16.4.771>

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## I. 서 론

H.264/AVC는 ISO/IEC 동영상 전문가 그룹과 ITU-T 비디오 코딩 전문가 그룹에 의해 개발되어 표준화된 최신 비디오 코딩이다. H.264/AVC는 인트라 예측 부호화, 1/4 단위의 가변적 블록 단위의 움직임 보상 및 예측, 엔트로피 부호화, 그리고 블록 경계의 왜곡을 억제하는 디블로킹 필터 등을 이용하여 기존의 비디오 부호화 표준인 MPEG-2 보다 약 2배, MPEG-4 보다 약 1.5배의 압축률이 향상된다[1,2]. H.264/AVC는 CABAC (Context Adaptive Binary Arithmetic Coding)와 CAVLC (Context Adaptive Variable Length Coding) 엔트로피 부호화를 사용한다. CABAC은 메인 프로파일 이상에서 사용되며 CAVLC와 비교하여 약 9%~14%의 비트 압축률을 나타내지만 연산의 복잡도가 증가하는 단점이 있다[3]. CABAC은 이진화의 확률값에 대해서 산술 연산을 수행하며 순차적인 구조로 인해 데이터의 의존도가 높고 확률값의 범위에 따라 재정규화 과정을 여러 번 수행한다.

본 논문에서는 H.264/AVC 엔트로피 코딩중의 하나인 CABAC 이진 산술 부호화기의 면적 감소 및 고성능을 위한 공통 연산기 구조를 제안한다. 제안한 공통 연산기 구조는 모드에 따라 다르게 동작하는 산술 부호화기 및 재정규화기를 하나의 공통 연산기로 처리하여 이진 산술 부호화기의 면적 및 복잡도가 감소한다. 또한 제안한 CABAC의 이진 산술 부호화기는 4단파이프라인을 적용하여 매 클럭당 한 심벌이 부호화를 수행한다.

본 논문의 구성은 다음과 같다. II장에서는 CABAC 부호화 알고리즘에 대해서 기술하고, III장에서는 제안하는 CABAC 이진 산술 부호화기에 대해 기술하며, IV장에서는 하드웨어 구현 및 성능 분석에 대해 기술한다. 마지막으로 V장에서는 본 연구의 결론을 도출한다.

## II. 본 론

높은 통계적 의존성을 갖는 비디오 신호의 경우 기존에 존재하는 비디오 부호화 기술을 사용하여 효율적인

부호화를 할 수 없으므로 통계적인 특성을 고려한 새로운 엔트로피 부호화 방식인 CABAC를 고안하였다. CABAC 부호화는 각 문맥 요소에 대한 확률 모델을 선택하는 문맥 기반 모델링 방법을 통해 적응적 이진 산술 부호화를 수행하여 부호화 한다[4]. CABAC 부호화기는 그림 1과 같이 이진화기(Binarizer), 문맥 모델러(Context Modeler), 이진 산술 부호화기(Binary Arithmetic Encoder)의 3단계로 구성된다.

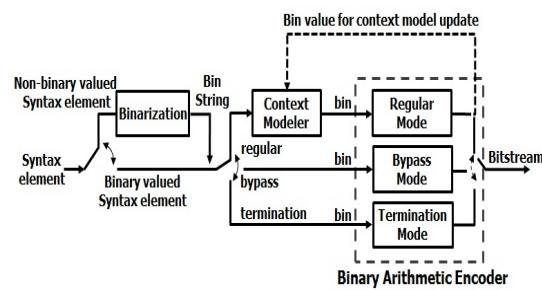


그림 1. CABAC 부호화기의 블록 다이어그램

Fig. 1 Block diagram with a CABAC encoder

이진화기는 이진값이 아닌 신택스 요소(Syntax Element)의 비 이진값을 빈 스트링(Bin String)이라 불리는 이진화된 시퀀스로 처리된다. 이진값을 갖는 신택스 요소가 입력 신호로 주어질 경우에는 이진화기는 바이패스 된다.

선택스 요소의 이진값들은 주어진 발생 확률 값에 따라 세 가지 모드로 산술 부호화 한다. 정규화 모드에서는 실제 산술 부호화 과정을 수행하기 전에 주어진 빈값은 위해 문맥 모델 단계로 들어간다. 문맥 모델 단계에서 현재 빈값에 대응하는 확률 모델은 이전에 부호화된 신택스 요소, 혹은 빈값에 따라 선택된다. 확률 모델이 결정된 후, 결정된 확률 모델과 주어진 확률 모델에 따라 빈값은 정규 부호화 엔진의 입력이 된다. 입력된 두 정보를 이용하여 이진 산술 부호화를 수행하며 확률 모델을 업데이트 한다.

바이패스 모드는 주어진 빈값에 대해 바이패스 부호화 모드를 선택할 때 사용되며 이전에 부호화된 신택스 요소의 확률 모델을 사용하지 않는 부호화 방식이다. 종결 모드는 매 슬라이스의 종결 여부를 판단하는 `end_of_slice_flag`의 신택스 요소의 이진값을 처리한다.

### III. 제안하는 CABAC 이진 산술 부호화기

기존 CABAC의 이진 산술 부호화기는 이진 값의 발생 확률에 따라 표 1과 같이 정규화, 바이패스, 종결 모드로 동작한다. 각 모드마다 Range와 Low를 연산하는 방법이 다르기 때문에 이진 산술 부호화기는 3가지 모드로 분리하여 수행하며 3가지 모드를 각각 하드웨어로 구현하기 때문에 수행 복잡도 및 면적이 증가한다. 또한 이진 산술 부호화기는 연산의 복잡성을 줄이기 위해 곱셈의 연산을 덧셈으로 계산하는 Q-Code, QM, MQ 코더를 사용하지만[5-8], 제안한 구조에서는 쉬프트로만 구현하여 하드웨어 복잡도를 줄였다.

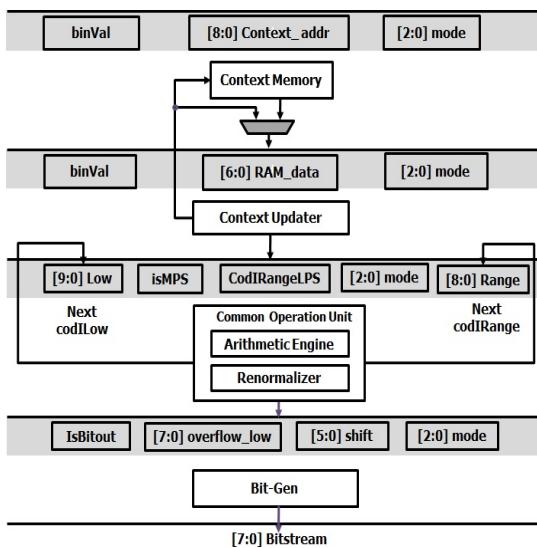


그림 2. 제안하는 CABAC 이진 산술 부호화기 구조  
Fig. 2 Proposed CABAC of binary arithmetic encoder architecture

본 논문에서 제안한 CABAC 이진 산술 부호화기 하드웨어 구조는 그림 2와 같이 Context Memory, Context Updater, 공통 연산기(Common Operation Unit), Bit-Gen으로 구성된다.

표 1. 기존 산술 부호화기 계산식  
Table. 1 Previous formula of Arithmetic Encoder

정규화 모드(Regular Mode)		
	MPS	LPS
Range	Range - RangeLPS	RangeLPS
Low	Low	Low + Range - RangeLPS
바이패스 모드(Bypass Mode)		
	MPS	LPS
Range	Range	Range
Low	Low << 1	(Low << 1) + Range
종결 모드(Termination Mode)		
	MPS	LPS
Range	Range - 2	2
Low	Low	Low + Range - 2

공통 연산기는 산술 부호화기(Arithmetic Engine)와 재정규화기(Renormalizer)로 구성되며, 본 논문에서 제안하는 산술 부호화기는 각 모드에 따라 다른 연산을 수행하는 Range와 Low의 값을 표 2와 표 3과 같이 최적화 한다.

표 2. 제안하는 Low와 Range  
Table. 2 Proposed Low and Range

구분	Low	Range
MPS	Low	Range
LPS	Low + Range	Mode 선택

표 3. 제안하는 MPS와 LPS  
Table. 3 Proposed MPS and LPS

Mode	MPS	LPS
정규화	Range-RangeLPS	RangeLPS
바이패스	0	Range
종결	2	2

최적화를 통해 하나의 공통 연산기를 사용하는 구조를 제안함으로써 연산 복잡도가 감소하고, 하드웨어 구조를 단순하게 구현함으로 저면적, 고성능 산술 부호화 기기를 구현 할 수 있다. 본 논문에서는 3가지 모드에 대해 최적화를 하였으며 그림 3과 같이 3개의 Mux, 2개의 빨리셈기, 1개의 덧셈기, 1개의 쉬프트 연산기를 사용한 공통 연산기 구조로 구현하였다.

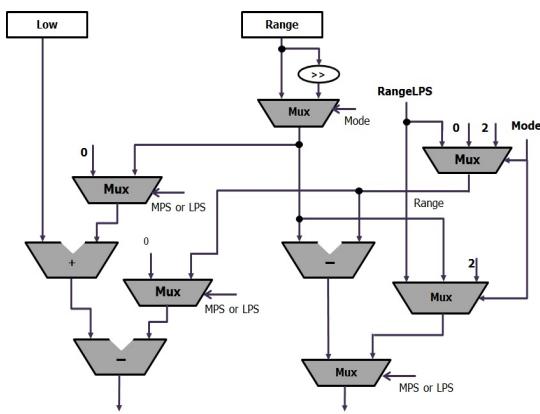


그림 3. 제안하는 산술 부호화 구조  
Fig. 3 Proposed Arithmetic Engine architecture

기존의 재정규화기[9,10] 또한 표 4와 같이 각 모드에 따라 다른 연산을 수행한다. 각 모드마다 Range와 Low를 연산하는 방법이 다르며 두개의 모드로 분리하여 수행한다.

표 4. 기존 재정규화 계산식  
Table. 4 Previous formula of renormalizer

정규화 & 종결 모드	바이패스 모드
<pre> While Range &lt; 256 if (Low &lt; 256)     Putbit(0); else     if (Low &gt;= 512)         Low = Low - 256;         bitsOutstanding++;     else         Low = Low - 512         Putbit(1);     endif endif endwhile </pre>	<pre> Low = Low &lt;&lt; 1;  If (Low &gt;= 1024)     Low = Low - 1024;     Putbit(1); else     if (Low &lt; 512)         Putbit(0);     else         Low = Low - 512;         bitsOutstanding++;     endif Endif </pre>

또한 기존의 재정규화기는 Range와 Low를 재설정하는 과정이 가변적으로 수행된다. Range의 재설정 과정은 최대 7번 반복 수행되며 Range의 재설정에 의해 Low의 값이 결정된다. Low값에 의해 비트 출력이 결정되므로 하드웨어 복잡도 및 면적이 증가한다.

```

if (Low < 256)
    Putbit(0);
else
    if (Low >= 512)
        Low = Low - 256;
        bitsOutstanding++;
    else
        Low = Low - 512;
        Putbit(1);
    endif
endif

```

(1)

본 논문에서 제안한 재정규화기는 각 모드에 따라 다르게 동작하는 재정규화기를 식 (1)과 같이 하나의 계산식을 사용한 공통 연산기 구조로 구현하였으며 그림 3과 같이 2개의 Mux, 4개의 쉬프트 연산기로 구성된 공통 연산기 구조로 구현하여 계산 복잡도를 최소화 하였다.

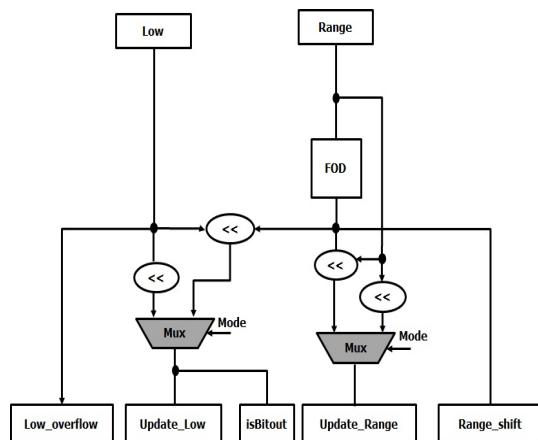


그림 4. 제안하는 재정규화기 구조  
Fig. 4 Proposed renormalizer architecture

또한 Range에 의해 수행되는 재정규화기의 가변적인 연산을 해결하기 위해 첫 번째 ‘1’의 위치를 탐색하는 FOD(First One Detector)를 사용하여 하나의 사이클 내에 계산할 수 있는 구조로 구현하였다. FOD는 Range를 상위, 하위 비트로 나누고 ‘1’의 위치를 탐색하여 왼쪽 쉬프트값을 찾는다. 계산된 왼쪽 쉬프트값은 Low와 Range에 적용하여 다음 산술 연산기의 산술 부호화를 위해 포워딩으로 구현하였다.

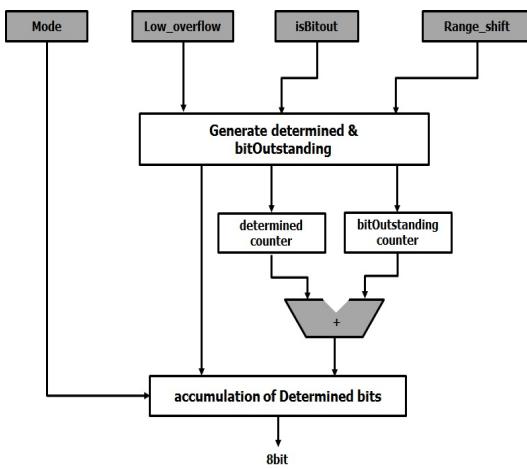


그림 5. 제안하는 Bit-Gen 구조  
Fig. 5 Proposed Bit-Gen architecture

본 논문에서 제안한 Bit-Gen은 그림 4와 같이 비트 출력을 위해 재정규화기로부터 출력된 7 bit Low\_overflow와 1 bit isBitout 그리고 3 bit의 Range\_shift를 입력받는다.

우선 Low\_overflow와 Range\_shift 값으로부터 결정된 값(determined value)과 bitsOutstanding 값을 분리한다. isBitout이 ‘1’인 경우에는 Low\_overflow와 bitsOutstanding이 존재하며 ‘0’인 경우에는 존재하지 않는다. isBitout이 ‘0’인 경우에 결정된 값을 64 bit 버퍼에 저장한다. ‘1’인 경우에는 카운터를 사용하여 bitsOutstanding을 증가시킨다. 증가된 bitsOutstanding은 다음 결정된 값의 MSB를 이용하여 ‘1’인 경우 “100000...” 형태의 결정된 값으로 변경하며, ‘0’인 경우 “011111...” 형태의 결정된 값으로 변경한다. 결정된 값은 64 bit 버퍼에 저장하며 8 bit 단위로 출력한다.

#### IV. 하드웨어 구현 및 성능 분석

제안한 구조는 H.264/AVC 참조 소프트웨어 JM 10.2에서 추출한 데이터를 이용하여 동작 검증 및 성능을 측정하였다. 표 5는 QP값이 28인 4개의 영상에서 I 프레임에 따른 평균 사이클 수를 비교한 것으로 Osorio[10], Li[11], Choi[12]의 구조에 비해 평균 사이클 수가 6%~44% 감소하였다.

표 5. I 프레임에 따른 평균 사이클 수 (QP=28)  
Table. 5 Average Cycle with I Frame (QP=28)

Sequence	Osorio [10]	Li [11]	Choi [12]	Proposed
Container	73048	73387	53253	51407
Foreman	73112	73430	53127	51252
Mobile	191669	193214	140131	137465
Akiyo	53426	53648	31340	29398

표 6은 제안한 구조의 하드웨어 구현 결과를 다른 구조들과 비교한 결과이다. 제안한 구조의 게이트 수는 4,147이며 Osorio[10], Liu[11], Choi[12], Kim[13] 구조 대비 각각 19%~47% 감소하였다. 제안한 CABAC 이전 산술 부호화 구조는 공통 연산기 구조를 사용하여 최대 222MHz 동작한다.

표 6. 하드웨어 구현 결과  
Table. 6 Implementation result of hardware

	Osorio [10]	Liu [11]	Choi [12]	Kim [13]	Proposed
Process (um)	0.35	0.13	0.18	0.18	0.18
Frequency (MHz)	186	200	NA	135	222
AE	4676	5163	4876	7344	2943
Bit-Gen	2830		2990		1204
Total	7506	5163	7866	7344	4147

## V. 결 론

본 논문에서는 공통 연산기(Common Operation Unit)를 이용한 CABAC의 이진 산술 부호화기를 제안한다. 제안한 공통 연산기는 정규화, 바이패스, 종결 모드에 상관없이 하나의 공통 연산기를 이용하여 산술 부호화 및 재정규화를 수행하며 하드웨어 구조를 단순화하여 면적을 감소시켰다. 또한 가변적인 재정규화 연산을 FOD(First One Detector)를 이용하여 하나의 사이클 내에 재정규화 연산을 수행한다. 제안한 CABAC의 이진 산술 부호화기 구조는 산술 부호화기 및 재정규화를 포함한 Common Operation Unit, Context Updater, Context RAM, Bit-Gen으로 구성되며 매 클럭당 하나의 심볼이 부호화될 수 있는 4단 파이프라인 구조로 구현하였다. 제안한 CABAC 산술 부호화기 구조는 Verilog HDL을 이용하여 설계하였으며, Magnachip 공정의 0.18um 표준 라이브러리를 기준으로 합성한 결과 게이트 수는 약 4.1k이고 기존 구조 대비 47% 감소하였다.

### 감사의 글

본 연구는 2011도 한밭대학교 교내연구비 지원사업에 의해 진행되었으며, IDEC으로부터 지원된 CAD Tool을 사용하였습니다.

### 참고문헌

- [ 1 ] ITU-T Recommendation H.264 and ISO/IEC 14496 -10, Advanced Video Coding for Generic Audio Visual Service, May 2010.
- [ 2 ] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard and Ajay Luthra, "Overview of the H.264/AVC video coding standard", IEEE Transaction on Circuits and System for Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.
- [ 3 ] J. Ostermann, J. Bormans and P. List, et al., "Video coding with H.264/AVC: tools, performance and complexity", IEEE Circuits and Systems Magazine, vol. 4, no. 1, pp. 7-28, First Quarter 2004.
- [ 4 ] Iain E. G. Richardson, The H.264 Advanced Video Compression Standard second Edition, John Wiley & Sons, August 2010.
- [ 5 ] Osorio. R. R and Bruguera. J. D, "A New Architecture for fast Arithmetic Coding in H.264 Advanced Video Coder", 8th Euromicro Conference on Digital System Design, pp. 298-305, September 2005.
- [ 6 ] I. H. Witten. R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression", Communication of the ACM 30, no. 6, pp. 520-540, June 1987.
- [ 7 ] R. C. Gonzalez, R. E. Woods, "Arithmetic coding", in Digital Image Processing, pp. 444-446, Prentice Hall, 2002.
- [ 8 ] A. Moffat, R. Neal and I. H. Witten, "Arithmetic Coding Revisited", IEEE Data Compression Conference Snowbird Utah, pp. 202-211, March 1995.
- [ 9 ] Nunez-Yanez Y.L., Chouliaras V.A. and Alfonso D., "Hardware assisted rate distortion optimization with embedded CABAC accelerator for the H.264 advanced video codec", IEEE Transactions on Consumer Electronics, vol. 52, no. 2, pp. 590-597, May 2006.
- [10] R. R. Osorio and J. D. Bruguera, "High-Throughput Architecture for H.264/AVC CABAC Compression System", IEEE Trans on Circuits and Systems for Video Technology, vol. 16, no. 11, pp. 1376-1384, November 2006.
- [11] P. S. Liu and J. W. Chen, "A Hardwired Context-based Adaptive Binary Arithmetic Encoder for H.264 Advanced Video coding", IEEE International Symposium on VLSI Design, Automation and Test, VLSI-DAT 2007, pp. 1-4, April 2007.
- [12] 최진하, 오명석, "H.264/AVC의 효율적인 파이프라인 구조를 적용한 CABAC 하드웨어 설계", 대한전자공학회지 논문지 제45권 SD편, 제 7호, pp. 7-9, 2008년 7월.
- [13] 김윤섭, 문전학, 이성수, "구문 요소의 저장 공간을 효과적으로 줄인 H.264/AVC CABAC 부호화기 설계", 대한전자공학회, 전자공학회 논문지-SD, 제47권 SD편, 제4호 pp. 34-40. 2010년 4월.

## 저자소개



명제진(Jejin Myoung)

2007년 한밭대학교  
정보통신공학과 공학사  
2012년 한밭대학교 정보통신전문  
대학원 공학석사

※ 관심분야 : SoC 플랫폼 설계, 하드웨어/소프트웨어  
통합설계, 멀티미디어 코덱 설계



류광기(Kwangki Ryoo)

1986년 한양대학교 공과대학  
전자공학과 공학사  
1988년 한양대학교 대학원  
전자공학과 공학석사

2000년 한양대학교 대학원 전자공학과 공학박사  
1991년 ~ 1994년 육군사관학교 교수부 전자공학과  
전임강사  
2000년 ~ 2002년 한국전자통신연구원 집적회로설계  
연구부 시스템IC설계팀 선임연구원  
2003년 ~ 현재 한밭대학교 정보통신공학과 교수  
※ 관심분야 : SoC 플랫폼 설계 및 검증, 하드웨어/  
소프트웨어 통합설계 및 검증, 멀티미디어코덱설계