

# HEVC CABAC 복호화기의 이진 산술 복호화기 설계

## Hardware Implementation of Binary Arithmetic Decoder in HEVC CABAC Decoder

김 소 현\*, 김 두 환\*, 이 성 수\*  
Sohyun Kim\*, Doohwan Kim\*, Seongsoo Lee\*

### Abstract

HEVC CABAC binary arithmetic decoder operates in three decoding modes i.e. regular, bypass, and termination modes, where their decoding operations and time differ a lot. Furthermore, when renormalization occurs, its internal feedback loop induces large delay. In this paper, a binary arithmetic decoder was designed to solve this problem. In advance, it checks all range values with possible renormalization. When renormalization occurs, it immediately updates range value and finishes all calculation in a cycle. When implemented in 0.18 um process technology, its maximum operating frequency and gate counts are 215 MHz and 5,423 gates, respectively.

### 요 약

HEVC CABAC의 이진 산술 복호화기는 정규, 우회, 종료의 세 가지 복호화 모드에 따라 동작하고 각 모드에 따라 복호화 동작과 시간에 많은 차이가 있다. 또한 재정규화를 진행하게 되면 내부에서 피드백 루프가 발생하여 지연 시간이 길어지게 된다. 본 논문에서는 이를 해결하기 위해 재정규화가 일어날 수 있는 모든 range 값의 범위를 미리 체크하여 정규화가 일어나면 바로 range 값을 업데이트하고 모든 계산을 한 사이클 안에 수행할 수 있도록 설계하였다. 0.18 um 공정에서 구현된 이진 산술 복호화기의 최대 동작 속도는 215 MHz이며 크기는 5,423 게이트이다.

*Key words: HEVC, CABAC, Decoder, Binary Arithmetic Decoder, BAD*

### 1. 서론

CABAC(Context-based Adaptive Binary Arithmetic Coding)[1-5]은 HEVC (High Efficiency Video Coding)[6][7]에서 쓰이는 엔트로피 코딩 기법으로 다른 엔트로피 코딩 기법에 비해 압축률이 높다. 하지만 이전까지 복호화 된 심볼들의 정보를 이용하는 부분에서 데이터의 의

존성이 크게 발생하고 피드백 루프가 길기 때문에 동작 속도가 느려지게 되며, 이 때문에 파이프라이닝을 적용하는 데에 어려움이 있다. HEVC에서 많은 연산량을 차지하는 코어 변환 (Core Transform)[8-13], 움직임 추정 (Motion Estimation)[14] 등은 하드웨어를 병렬로 사용하여 쉽게 속도를 높일 수 있지만, CABAC의 경우에는 순차 작업만 가능하기 때문에 속도를 높이

\* School of Electronic Engineering, Soongsil University

★ Corresponding author (e-mail: sslee@ssu.ac.kr, tel: 02-820-0692)

※ Acknowledgment

"This research was supported by Industrial Core Technology Development Program (10052009) funded by the Ministry of Trade, Industry & Energy, Korea."

Manuscript received Dec. 07, 2016; accepted Dec. 27, 2016

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.

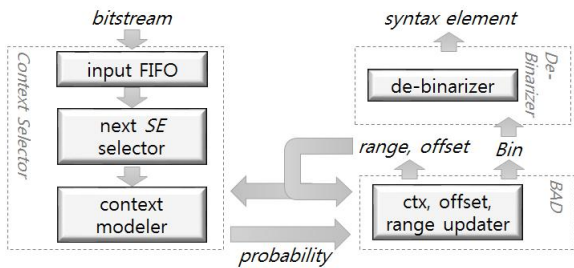


Fig. 1. Block diagram of HEVC CABAC decoder  
그림 1. HEVC CABAC 복호화기의 블록도

기 위해서는 파이프라이닝 기법 정도만 적용이 가능하다. 본 논문에서는 HEVC CABAC 복호화기를 설계할 때 파이프라이닝을 적용하기에 적합하도록 이진 산술 복호화기(Binary Arithmetic Decoder)의 구조를 제안하고 이를 구현하였다.

### II. HEVC CABAC 복호화기 아키텍처

CABAC 복호화기는 부호화기가 송신한 데이터인 비트열(bitstream\*)을 분석하고 이를 다시 구문 요소 (syntax element)로 복호화하는 기능을 수행한다. 그림 1은 CABAC 복호화기의 전체적인 아키텍처를 나타낸 것인데, 크게 문맥 선택기 (Context Selector), 이진 산술 복호화기 (Binary Arithmetic Decoder), 역이진화기 (De-Binarizer)의 세 부분으로 이루어져 있다. 문맥 선택기에서는 다음의 syntax element가 무엇일지 예측하고 그에 따른 확률 모델(context model)을 계산한다. 이진 산술 복호화기에서는 context model에 따라 bitstream을 빈(bin)으로 바꾸고 그에 사용되는 값들인 range, offset 등의 값을 업데이트한다. 여기서 업데이트 한 값과 복호화한 bin의 값을 전 단계인 문맥 선택기에서 이용해야하기 때문에 전체적인 지연 시간이 커지게 된다. 마지막으로 역이진화기에서는 bin을 받아 syntax element로 바꾸어 출력한다.

### III. HEVC CABAC 이진 산술 복호화기 아키텍처

\* 본 논문에서 이탤릭체로 표기된 것은 HEVC 표준 내에서 규정하는 공식적인 데이터와 파라미터로서 이들은 CABAC 복호화기 내의 각종 연산 하드웨어에서 사용되는 물리적 신호이다.

일반적으로 산술 코딩은 다른 가변 길이 코딩보다 더 높은 압축 성능을 가지며, 산술 코딩의 일종인 CABAC은 range를 [0, 1]의 범위가 아닌 [0, 510]의 범위를 사용하고 MPS (Most Probable Symbol) 및 LPS (Least Probable Symbol) 확률에 따라 범위를 간격으로 나누는 과정을 근사하기 때문에 복잡한 계산이 아닌 참조 테이블 (lookup table)로 더욱 간단하고 빠른 구현이 가능해졌다.

그림 2는 이진 산술 복호화기의 전체 구조를 나타낸 그림이다. BAD는 복호화 모드(decoding mode)에 따라 각각 정규(regular), 우회(bypass), 종료(termination)의 세 가지 모드로 동작하고 주어진 모드에 따라 결과를 선택하여 출력한다. 복호화기에 입력으로 들어가는 bitstream은 복호화하기 전까지 몇 비트가 사용될 지 알 수 없으므로, 사용한 bitstream의 비트 수를 받아 그만큼 쉬프트하여 이전에 사용한 bit를 다시 사용하는 일이 없도록 설계하였다.

#### 가. 정규 모드

정규 모드는 통상적으로 데이터패스가 가장 긴 모드로 context model을 필요로 하고 LPSrange와 range, offset을 전부 업데이트한다. 업데이트한 range, offset의 값은 내부에서 다음 bin의 복호화에 다시 사용된다. 또한 필요할 경우 재정규화(renormalization)를 진행한다. LPSrange 업데이트는 참조 테이블을 사용하여 그림 3과 같이 설계하였다.

#### 나. 우회 모드

우회 모드는 offset만을 업데이트하고 재정규화를 하지 않기 때문에 다른 모드에 비해 데이터패스가 매우 짧으며 그림 4와 같이 설계하였다.

#### 다. 종료 모드

종료 모드는 context model을 사용하지 않아 LPSrange의 업데이트는 하지 않지만 range, offset을 업데이트하고 재정규화를 진행하며 그림 5와 같이 설계하였다.

#### 라. 재정규화

재정규화는 확률 영역을 일정 크기 이상으로 유지하기 위해 수행하는 것으로, 하나의 bin을 복호화할 때 재정규화 과정은 일어나지 않을 수도 있고, 여러 번 일어날 수도 있다. 이 과정에서 range가 일정 이하의 값을 가지는지 비교하고 2

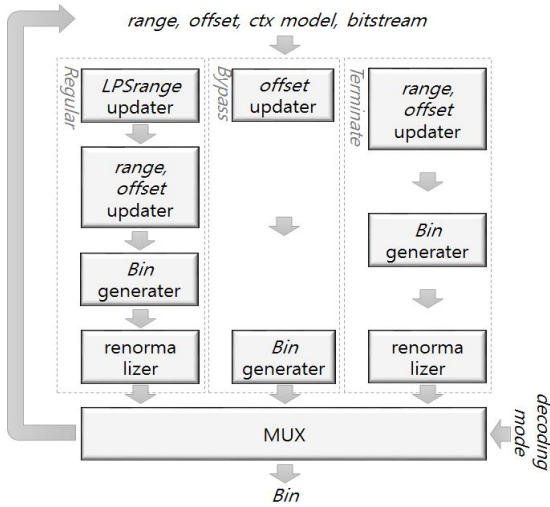


Fig. 2. Block Diagram of Binary Arithmetic Decoder  
 그림 2. 이진 산술 복호화기의 블록도

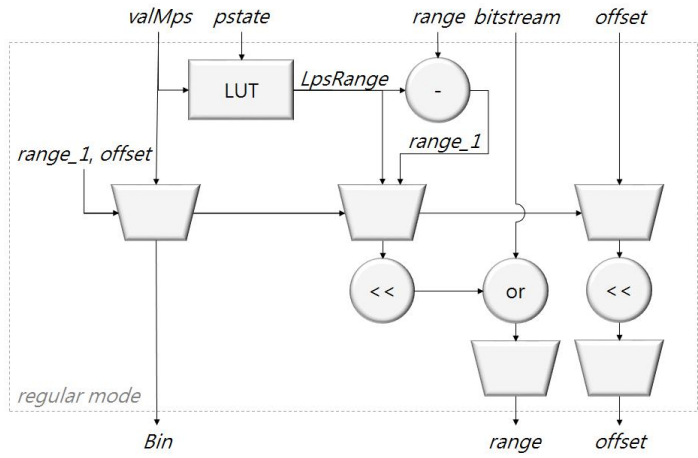


Fig. 3. Block Diagram of Regular mode  
 그림 3. 정규 모드의 블록도

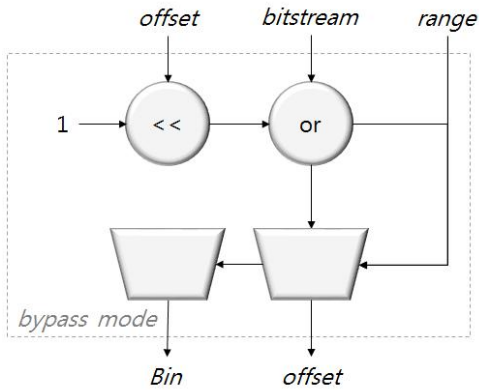


Fig. 4. Block Diagram of Bypass mode  
 그림 4. 우회 모드의 블록도

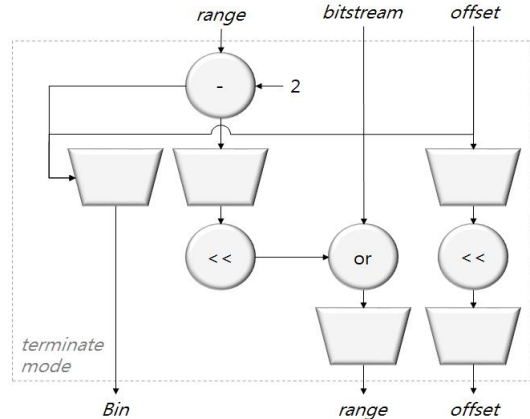


Fig. 5. Block Diagram of Terminate mode  
 그림 5. 종료 모드의 블록도

바로 증가시키는 과정을 반복하면서 *range*를 일정 이상의 값 범위로 유지시킨다.

이는 내부의 피드백 루프와 같으므로 만약 이 과정이 한 번 이상 일어나게 된다면 그만큼 반복해야 하는 횟수도 늘어나 데이터패스가 그 만큼 길어지게 된다. 그래서 이 데이터패스를 줄이고 무조건 한 사이클 안에 실행시킬 수 있도록 하기 위해 재정규화가 일어날 수 있는 모든 *range* 값의 범위를 미리 체크하여 정규화가 일어나면 바로 *range* 값을 업데이트하고 모든 계산을 한 사이클 안에 수행할 수 있도록 설계하였다.

이진 산술 복호화기에서 진행되는 모든 계산 및 과정은 전부 한 사이클에 하나의 *bin*을 디코딩 할 수 있도록 설계되었는데, 이는 파이프라인이 적용된 복호화기를 고려하여 설계한 것이다.

#### IV. 구현 결과 및 결론

본 논문에서 설계한 이진 산술 복호화기를 검증하기 위해 HEVC 표준 소프트웨어인 HM11 [15]을 사용하여 표 1의 12개 테스트 영상에 대해 시뮬레이션을 수행하였으며, 모두 문제가 없이 동작하였다. 설계된 하드웨어를 0.18um 공정에서 합성해본 결과는 표 2와 같으며 게이트 수, 최대 동작 속도, 최대 처리율은 각각 5,423 게이트, 215 MHz, 215 Mbin/s이다. HEVC의 다른 연산과 달리 CABAC은 부호화 조건에 따라 생성되는 비트열의 양이 크게 달라지기 때문에 최대 처리율만 가지고는 어느 정도의 영상 크기를 처리할 수 있다고 보장할 수 없으나 실용적인 범위에서는 Class A 영상을 충분히 처리할 수 있다.

Table 1. Test sequences

표 1. 테스트 영상

Sequence	Class A	Class B	Class C	Class D
Size	2560x	1920x	832x	416x
	1600	1080	480	240
Frame Rate	60	24	60	50
Frames	30			
Encoder Configuration	low_delay, random access, all_intra			

Table 2. Synthesize result

표 2. 합성 결과

Process Technology	0.18um
Max Operating Frequency	215MHz
Max Throughput	215MBin/s
Gates	5,423 gates

본 논문에서 제안한 CABAC 이진 산술 복호화기는 크기가 작고 속도가 빨라서 다양한 HEVC 복호화기에 응용될 수 있으며 특히 파이프라인 구조로 구현할 때 유용하게 사용될 수 있다.

## References

- [1] D. Pham, J. Moon, and S. Lee, "Hardware Implementation of HEVC CABAC Binarizer," *j.inst.Korean.electr.electron.eng*, vol. 18, no. 3, pp. 356-361, 2014.
- [2] D. Pham, J. Moon, D. Kim, and S. Lee, "Hardware Implementation of HEVC CABAC Binary Arithmetic Encoder," *j.inst.Korean.electr.electron.eng*, vol. 18, no. 4, pp. 630-635, 2014.
- [3] D. Kim, J. Moon, and S. Lee, "Hardware Implementation of HEVC CABAC Context Modeler," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 2, pp. 254-259, 2015.
- [4] D. Kim, J. Moon, and S. Lee, "Design of HEVC CABAC Encoder with Parallel Processing of Bypass Bins," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 4, pp. 583-589, 2015.
- [5] D. Kim, S. Kim, and S. Lee, "Hardware Implantation of De-Binarizer in HEVC CABAC Decoder," *j.inst.Korean.electr.electron.eng*, vol. 20, no. 3, pp. 326-329, 2016.
- [6] G. Sullivan, J. Ohm, W. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.* vol. 22, no. 15, pp. 1649-1668, 2012.
- [7] B. Bross, W. Han, J. Ohm, G. Sullivan, and T. Wiegand, "JCTVC-L1003\_v34: High efficiency video coding (HEVC) text specification draft 10," Joint Collaborative Team on Video Coding (JCT-VC), 2013.
- [8] S. Han, W. Nam, and S. Lee, "Design of Low-Area HEVC Core Transform Architecture," *j.inst.Korean.electr.electron.eng*, vol. 17, no. 2, pp. 119-128, 2013.
- [9] J. Lee and S. Lee, "8x8 HEVC Inverse Core Transform Architecture Using Multiplier Reuse," *j.inst.Korean.electr.electron.eng*, vol. 17, no. 4, pp. 570-578, 2013.
- [10] S. Jung and S. Lee, "Design of Unified HEVC 4x4 IDCT/IDST Block," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 2, pp. 271-275, 2015.
- [11] S. Jung and S. Lee, "Design of Unified HEVC/VP9 4x4 Transform Block," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 3, pp. 392-399, 2015.
- [12] J. Lee and S. Lee, "16x16 HEVC Inverse Core Transform Architecture Using Multiplier Reuse," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 3, pp. 378-384, 2015.
- [13] S. Jung and S. Lee, "Design of Unified Inverse Transformer for HEVC and VP9," *j.inst.Korean.electr.electron.eng*, vol. 19, no. 4, pp. 596-602, 2015.
- [14] A. Hur, T. Park, and S. Lee, "Design of HEVC Motion Estimation Engine with Search Window Data Reuse and Early Termination," *j.inst.Korean.electr.electron.eng*, vol. 20, no. 3, pp. 273-278, 2016.
- [15] HEVC software repository HM-11 reference model, [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVC\\_Software/branches/HM-11.0-dev/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVC_Software/branches/HM-11.0-dev/)