

# HEVC CABAC 복호기의 문맥 모델러 설계

## Hardware Implementation of Context Modeler in HEVC CABAC Decoder

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

### Abstract

HEVC (high efficiency video coding) exploits CABAC (context-based adaptive binary arithmetic coding) for entropy coding, where a context model estimates the probability for each syntax element. In this paper, a context modeler was designed and implemented for CABAC decoding. lookup table was used to reduce computation and to increase speed. 12 simulations for HEVC standard test sequences and encoder configurations were performed, and the context modeler was verified to perform correction operations. The designed context modeler was synthesized in 0.18 um technology. Maximum frequency, maximum throughput, and gate count are 200 MHz, 200 Mbin/s, and 29,268 gates, respectively.

### 요 약

HEVC(high efficiency video coding)의 엔트로피 코딩 방식인 CABAC(context-based adaptive binary arithmetic coding)에서는 각 구문 요소의 발생 확률을 추정하는 문맥 모델이 사용된다. 본 논문에서는 CABAC 복호화에 필요한 문맥 모델러를 설계하고 이를 구현하였다. 초기화에 필요한 연산 숫자를 줄이고 속도를 높이기 위해 참조 테이블을 사용하였으며, HEVC의 표준 테스트 영상 및 표준 부호기 구성에 대해 12가지의 시뮬레이션을 수행하여 모두 성공적으로 동작하는 것을 확인하였다. 설계된 문맥 모델러를 0.18 um에서 합성하였을 때의 최대 동작 주파수, 최대 처리율 및 게이트 수는 각각 200 MHz, 200 Mbin/s, 29,268 게이트이다.

*Key words: HEVC, CABAC, Decoder, Context Modeler, Lookup Table*

### 1. 서론

HEVC(high efficiency video coding)[1]-[8]는 현재 가장 널리 사용되는 동영상 압축 국제 표준이다. HEVC는 기존의 동영상 압축 국제 표준에 비해 압축 효율이 매우 높은 대신에 하드웨어 구현이 어려운데, CABAC(context-based adaptive binary arithmetic coding)[9]-[14]과 같이 복잡한 기법을 많이 채용했기 때문이다.

문맥 모델러(context modeler)[11][12]는 문맥에 따른 구문 요소(syntax element)가 발생하는 확률을 추정하는 블록으로, CABAC에서 매우 중요한 역할을 담당하는 모듈의 하나이다. CABAC 부호화에서의 문맥 모델러[11][12]와는 다르게, CABAC 복호화에서는 문맥 모델을 각 슬라이스마다 초기화 시키는 과정에서 많은 연산량이 필요하며 속도 저하도 발생하게 된다. 본 논문에서는

\* 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 Sep. 18, 2017; revised Sep. 26, 2017; accepted Sep. 28, 2017.

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.

이러한 문제점을 해결하기 위해서 참조 테이블(lookup table: LUT)을 사용한 CABAC 복호기용 문맥 모델러를 설계, 구현 및 검증하였다.

## II. 문맥 모델러의 설계

### 1. 전체 아키텍처

HEVC CABAC 복호기는 그림 1처럼 문맥 선택기(context selector), 문맥 모델러, 이진 산술 복호기(binary arithmetic decoder)[14], 역 이진화기(de-binarizer)[13]로 구성되어 있다. 문맥 모델러는 구문 요소의 발생 확률을 추정하기 위해 각 구문 요소의 ID와 정보, 복호화된 빈(bin), 주변 블록 정보 등을 받아 확률 모델링을 진행한다.

본 논문에서 설계한 문맥 모델러는 그림 2와 같이 크게 네 부분으로 이루어져 있다. 전체 동작을 관리 및 제어하는 제어기(controller), 문맥 모델을 초기화시켜 주는 초기화기(initializer), 문맥 모델

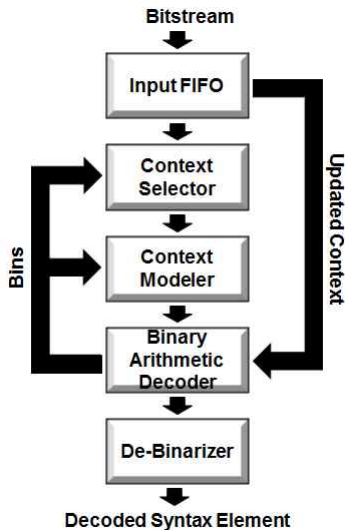


Fig. 1. CABAC decoder in HEVC  
그림 1. HEVC에서의 CABAC 복호기

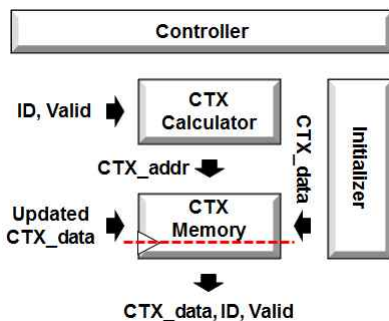


Fig. 2. Context modeler in HEVC CABAC decoder  
그림 2. HEVC CABAC 복호기의 문맥 모델러

을 계산하는 문맥 모델 계산기(CTX calculator), 마지막으로 각 문맥 모델을 저장하는 문맥 모델 메모리(CTX memory)이다.

### 2. 제어기

제어기는 전체 모듈을 관리 및 제어하는 모듈이다. Enable 신호를 통해 각 모듈의 동작을 관리하고 메모리의 읽기/쓰기 동작을 제어한다. 이 모듈은 초기화 신호와 종료 신호를 입력으로 받아 그에 따라 상태를 변화시키며 동작한다. 제어기는 READY, RUN, INIT의 세 상태를 가지며, 이들 상태는 그림 3과 같이 변화한다.

READY 상태는 초기 상태로서, 유효한 값이 들어올 때까지 대기하는 상태이다. 초기화 신호가 들어와 INIT 상태로 바뀔 때까지 해당 상태를 유지한다. 이후 종료 신호가 입력될 때까지 READY 상태로 바뀌지 않는다.

INIT 상태는 초기화 상태로 초기화 동작이 전부 끝날 때까지 초기화기를 동작시킨다. 동시에 초기화를 위해 문맥 모델 메모리의 쓰기 동작을 제어한다. 초기화가 끝나 초기화 종료 신호가 발생하면 RUN 상태로 바뀐다.

RUN 상태는 초기화가 끝난 후 구문요소를 처리하고 문맥 모델 계산기를 동작시키는 단계이다. 모든 구문 요소를 처리하여 종료 신호가 발생할 때까지 해당 상태를 유지한다.

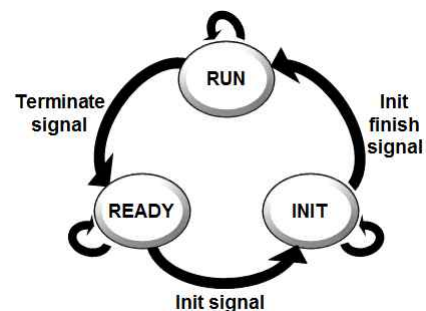


Fig. 3. State diagram of controller  
그림 3. 제어기의 상태도

```
//m, n LUT (address, slice_type)
initValue = LUT(address, slice_type)

slopeIdx = initValue >> 4
offsetIdx = initValue & 15

m = slopeIdx * 5 - 45
n = (offsetIdx << 3) - 16
```

Fig. 4. Initialization and LUT  
그림 4. 초기화 연산과 LUT

### 3. 초기화기

초기화기는 새로운 슬라이스가 처리될 때마다 실행되는 모듈로서, 문맥 모델을 해당 슬라이스의 QP값과 타입에 따라 초기화시켜 주는 모듈이다.

초기화기는 곱셈 연산이 두 번 들어가므로 모듈 중 연산량이 가장 많다. 이와 같은 문제를 개선하기 위해 초기화기는 m과 n을 미리 계산하여 그림 4와 같은 초기화 연산 수식과 LUT로 설계하였다. 이 경우 곱셈 연산이 두 번에서 한 번으로 감소하여 면적을 줄이고 속도를 높일 수 있다.

초기화기에서는 한 사이클에 하나의 확률 값을 처리하고, 한 슬라이스 종류에 최대 154개의 init\_value가 존재하므로 초기화에는 총 156cycle이 소요된다.

### 4. 문맥 모델 계산기 및 문맥 모델 메모리

문맥 모델 계산기는 그림 5와 같이 이웃 블록 데이터(NB data calculator) 계산기와 문맥 주소 계산기(CTX address calculator)로 구성되어 있으며, 문맥 모델 메모리는 이웃 블록의 데이터를 저장한다.

대부분의 구문 요소에서는 그 자신의 데이터만으로도 문맥 모델을 결정하는 것이 가능하다. 그러나 몇몇 구문 요소에서는 해당 bin이 속한 블록과 이웃 블록의 데이터를 전부 고려하여 문맥 모델을 결정해야 한다. 그러므로 문맥 모델을 결정하기 이전에 이웃 블록의 정보들을 처리해야 할 필요가 있다. 이에 해당하는 구문 요소는 대표적으로 cu\_skip\_flag, split\_cu\_flag, residual coding data 등이 있다. 문맥 모델 계산기는 각 구문 요소의 ID, bin, 데이터 깊이(depth)등 이웃 블록 데이터 처리에 필요한 파라미터, 이웃 블록 데이터를 입력으로 하는 LUT로 설계되었다.

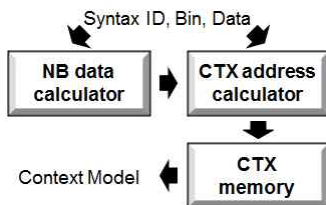


Fig. 5. Determination of context model  
 그림 5. 문맥 모델의 결정

### III. 문맥 모델러의 구현 및 검증

본 논문에서는 HEVC CABAC 복호기의 문맥 모델러를 Verilog HDL로 설계한 후, ModelSim을 이용하여 검증하였다. 설계된 문맥 모델러가 정상적으로 동작하는지 확인하기 위해 표 1과 같이 HEVC 표준 테스트 영상을 클래스 A~D까지 4개의 영상을 사용하였고, 각각에 대해 HEVC 표준 부호기 구성 3가지에 대해 시뮬레이션을 수행하였다. 총 12가지의 시뮬레이션을 수행한 결과를 HEVC 표준 소프트웨어 모델 HM Revision 4919[15]와 비교하여 문맥 모델러가 정상적으로 동작하는 것을 확인하였다.

설계된 하드웨어를 0.18um 공정에서 합성하였을 때의 결과는 표 2와 같으며 게이트 수, 최대 동작 주파수, 최대 처리율은 각각 29,268 게이트, 200 MHz, 200 Mbin/s이다. [2]에 따르면 HEVC에서 4k급 영상을 처리하기 위한 main profile, level 5.1은 화면 크기 4,096×2,160, 60 fps이며 최대 비트레이트는 40 Mbps이다. bps와 bin/s가 동일한 것은 아니며 부호화 조건에 따라 크게 차이 나기 때문에 정확하게 비교할 수는 없지만, 본 논문에서 설계된 문맥 모델러는 최대 처리율이 200 Mbin/s이므로 최대 비트레이트가 40 Mbps인 4k급 60 fps 영상을 쉽게 처리할 수 있다.

### IV. 결론

Table 1. Test sequences

표 1. 테스트 영상

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

Table 2. Synthesize result

표 2. 합성 결과

Fabrication Technology	0.18um
Synthesized Gates	29,268 gates
Maximum Operating Frequency	200 MHz
Maximum Throughput	200 MBin/s

본 논문에서는 HEVC CABAC 복호기를 위한 문맥 모델러를 설계하고 이를 구현 및 검증하였다. 설계된 문맥 모델러는 슬라이스 별 초기화에 소요되는 연산을 줄이기 위해 LUT 방식으로 설계되었으며 4k급 60 fps 영상을 문제없이 처리할 수 있다. 0.18 um 공정으로 합성하였을 때의 최대 동작 주파수, 최대 처리율 및 게이트 수는 각각 200 MHz, 200 Mbin/s, 29,268 게이트이다.

## References

- [1] G. Sullivan, J. Ohm, W. Han, and 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.  
DOI : 10.1109/TCSVT.2012.2221191
- [2] B. Bross, W. Han, J. Ohm, G. Sullivan, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 10," Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.
- [3] S. Jung and S. Lee, "Design of Unified HEVC 4×4 IDCT/IDST Block," *j.inst.Korean.electr.electron.eng.* vol. 19, no. 2, pp. 271-275, 2015.  
DOI : 10.7471/ikeee.2015.19.2.271
- [4] S. Jung and S. Lee, "Design of Unified HEVC/VP9 4×4 Transform Block," *j.inst.Korean.electr.electron.eng.* vol. 19, no. 3, pp. 392-399, 2015.
- [5] J. Lee and S. Lee, "16×16 HEVC Inverse Core Transform Architecture Using Multiplier Reuse," *j.inst.Korean.electr.electron.eng.* vol. 19, no. 3, pp. 378-384, 2015.  
DOI : 10.7471/ikeee.2015.19.3.378
- [6] 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.  
DOI : 10.7471/ikeee.2015.19.4.596
- [7] 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.
- [8] S. Lee, "Optimization Method on the Number of the Processing Elements in the Multi-Stage Motion Estimation Algorithm for High Efficiency Video Coding," *j.inst.Korean.electr.electron.eng.* vol. 21, no. 1, pp. 100-103, 2017.
- [9] 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.  
DOI : 10.7471/ikeee.2014.18.3.356
- [10] 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.  
DOI : 10.7471/ikeee.2014.18.3.356
- [11] 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.  
DOI : 10.7471/ikeee.2015.19.2.254
- [12] 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.  
DOI : 10.7471/ikeee.2015.19.4.583
- [13] 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.
- [14] S. Kim, D. Kim, and S. Lee, "Hardware Implementation of Binary Arithmetic Decoder in HEVC CABAC Decoder," *j.inst.Korean.electr.electron.eng.* vol. 20, no. 4, pp. 435-438, 2016.
- [15] HEVC Software - Revision 4919, [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/branches/HM-dev/](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-dev/)