ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 15-02-18

j.inst.Korean.electr.electron.eng.Vol.19,No.2,254~259,June 2015
http://dx.doi.org/10.7471/ikeee.2015.19.2.254

HEVC CABAC 문맥 모델러의 하드웨어 구현

Hardware Implementation of HEVC CABAC Context Modeler

김 두 환*, 문 전 학*, 이 성 수*★

Doohwan Kim*, Jeonhak Moon*, Seongsoo Lee*★

Abstract

CABAC is a context-based adaptive binary arithmetic coding method. It increases the encoding efficiency by updating the probability based on the information of the previously coded symbols. Context modeler is a core block of CABAC, which designs a probability model according to the symbol considering statistical correlations. In this paper, an efficient hardware architecture of CABAC context modeler is proposed. The proposed context modeler was designed in Verilog HDL and it was implemented in 0.18 um technology. Its gate count is 29,832 gates including memory. Its operating speed and throughput are 200 MHz and 200 Mbin/s, respectively.

요 약

CABAC은 문맥 기반 적응적 이진 산술 부호화 방식으로, 이전까지 부호화 된 심볼들의 정보를 이용하여 확률을 업데이트하여 부호화 효율을 높이는 기법이다. 문맥 모델러는 통계적 상관성을 고려하여 심볼에 따라 확률 모델을 설계하는 CABAC의 핵심 블록으로서, 본 논문에서는 문맥 모델러의 효율적인 하드웨어 아키텍처를 제안한다. Verilog HDL로 기술되어 0.18 um 공정으로 설계된 문맥 모델러는 메모리를 포함하여 29,832개의 게이트로 이루어져 있으며, 최대 동작속도는 200 MHz, 최대 처리율은 200 Mbin/s이다.

Key words : HEVC, CABAC, context modeler, hardware, implementation

Ⅰ. 서론

최근 디지털 정보는 매년 급격하게 증가하고 있다. 이러한 정보 증가의 원인은 정보의 형태가 예전의 텍스트 위주의 정보에서 음성, 영상 데이터 위주로 변화되고 있기 때문이다. 영상을 포함한 데이터들이 등장하면서 이를 처리하기 위한 다수의 영상 압축 표준들이 제정되었다. 가장 최근 제정된 영상 압축 표준인 HEVC [1]-[4]는 가장 뛰어난 압축율을 보이고 있지만 기본적으로 많은 연산량이 요구된다.

본 논문에서는 HEVC 연산 중에서 엔트로피 코딩을 담당하는 CABAC의 문맥 모델러의 연산량을 줄이

* School of Electronic Engineering, Soongsil University

★ Corresponding author: sslee@ssu.ac.kr, 02-820-0692

※ Acknowledgment

"This research was supported by the System IC Commercialization R&BD Program (10049498) funded by the Ministry of Trade, industry & Energy."

Manuscript received Jun. 1, 2015; revised Jun. 22, 2015 ; accepted Jun. 23. 2015

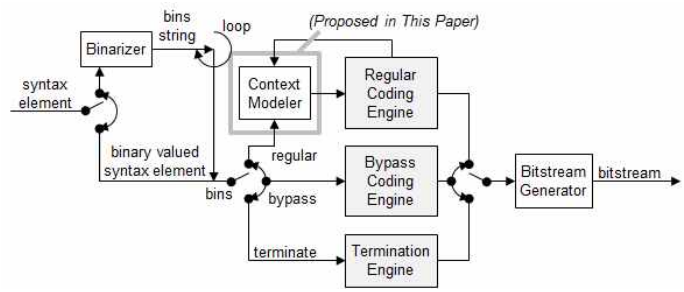

Fig. 1. Block diagram of CABAC.
그림 1. CABAC의 전체 블록도.

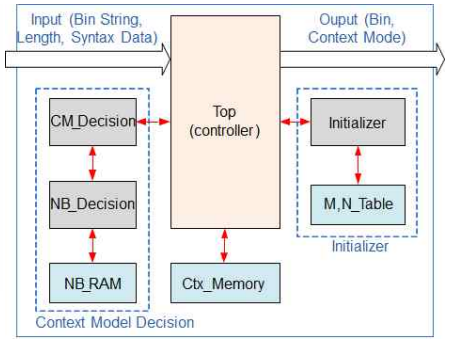

Fig. 2. Architecture of the proposed context modeler.
그림 2. 제안하는 문맥 모델러의 아키텍쳐.

고 보다 효율적으로 하드웨어를 구현하기 위한 아키텍처에 대해 제안한다.

II. 전체 아키텍처

CABAC 인코더는 그림 1과 같이 이진화기, 문맥 모델러, 이진 산술 부호화부로 구성되어 있다 [5]-[7]. 여기서 문맥 모델러는 이진화부를 통해 이진화된 빈을 입력으로 하여 심볼의 확률을 모델링한다. 확률 추정을 위해 빈 (bin) 외에도 구문 요소 (syntax element)의 정보와 주변 블록들의 정보가 필요하기 때문에, 이를 저장 및 처리하기 위해 효율적인 하드웨어의 설계가 요구된다. 본 논문에서는 적절히 파이프라인으로 동작하며 구문요수를 효율적으로 저장하여 메모리의 용량을 감소시킨 저면적 문맥 모델러의 하드웨어를 제안한다.

제안하는 문맥 모델러는 그림 2와 같이 제어부와 초기화부, 문맥 모델을 저장하는 메모리, 문맥 모델 결정부로 이루어져 있다. 입력으로는 이진화기로부터 각 구문요수들의 ID와 이진화된 빈 스트링, 빈 스트링의 길이, 그리고 구문요수의 정보와 슬라이스에 대한 정보를 받으며, 이진 산술 부호화기로부터 업데이트 된 문맥 모델을 받는다. 출력으로는 하나의 빈과 그 빈의 문맥 모델, 그 빈의 인코딩 모드를 내보낸다.

각각의 구문요수마다 다른 정보들을 필요로 하며 발생하는 빈의 길이 또한 다르다. 따라서 본 논문에서는 구문요수의 데이터와 빈 스트링을 묶어 3가지의 포맷으로 나누어 각각의 필요한 정보와 빈을 할당하여 비트수를 줄였다. 제어부는 ID를 확인하여 이에 맞는 포맷으로 정보를 인식하며, 이를 통해 입력의 총 비트 수를 절반 이하로 줄일 수 있다.

III. 제어부

제어부는 문맥 모델러의 동작을 제어한다. 그림 3과 같이 초기화부와 문맥 모델 결정부의 enable 신호를 통해 두 모듈의 동작을 제어하고 메모리의 읽고 쓰기 동작을 제어한다. 유한 상태 머신을 이용하여 설계하였으며 그림 4는 제어부의 유한 상태 머신을 보여준다.

- START : FIFO 메모리에 유효한 값이 들어올 때까지 대기하는 상태로 초기 상태이다. 메모리로부터 읽어오는 입력이 유효할 때 초기화 신호가 들어오면 INIT 상태로 아닐 경우 RUN 상태로 전이한다.
- INIT : 초기화 상태로 초기화 동작이 종료될 때까지 초기화부를 동작시키며 메모리의 쓰기 동작을 제어한다. 초기화가 종료되면 START 상태로 전이한다.
- RUN : 구문요수를 처리하는 상태로 문맥 모델 결정부를 동작시키며 각 구문요수의 ID를 읽어 메모리를 제어한다. 동작 중 초기화 신호가 들어오면 초기화 상태로 전이하며 중단 신호가 들어오면 중단 상태로 전이한다.
- STOP : 중단 상태로 비트 스트림의 양을 컨트롤하기 위한 출력 단의 메모리가 포화되어 더 이상 데이터를 내보낼 수 없을 경우 빈의 처리를 중단하며, 중단 신호가 0으로 떨어지면 다시 동작 상태로 전이한다.

IV. 초기화부

초기화부는 새로운 슬라이스를 부호화할 때마다 실행되며 문맥 모델을 슬라이스의 qp값과 타입에 따라 문맥 모델을 초기화시켜준다.

초기화 모듈은 곱셈 연산이 두 번 들어가기 때문에 연산량이 가장 많은 모듈이다. 본 논문에서는 연산량을 줄이기 위해 m과 n값을 미리 계산하여 Look Up Table (LUT)로 만들면 곱셈 연산을 한번으로 줄일 수

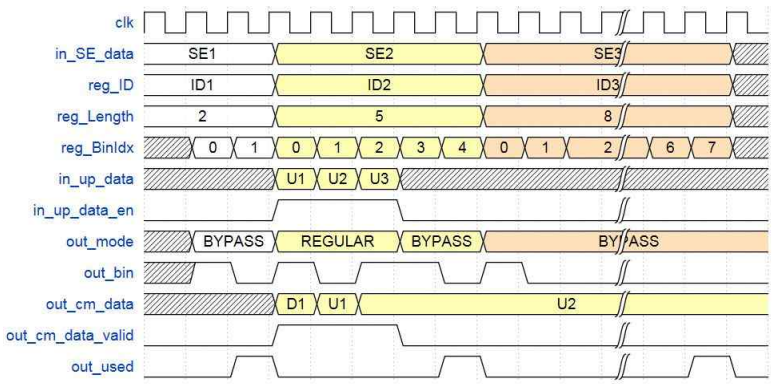

Fig. 3. Timing diagram of the context modeler.
그림 3. 문맥 모델러의 타이밍도.

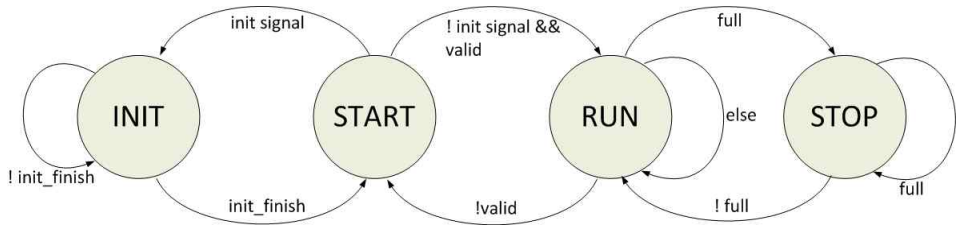

Fig. 4. Finite state machine in the controller.
그림 4. 제어부의 유한 상태 머신

있어 면적을 크게 줄일 수 있다. 그림 5는 초기화 연산 수식과 m,n LUT를 보여준다. 그림 6은 초기화부의 파이프라이닝을 보여주며, 초기화부는 적절한 파이프라이닝을 통해 동작 속도를 향상시킬 수 있다.

초기화부는 병렬 처리를 통해 속도 향상을 시킬 수 있으나 초기화가 자주 발생하지 않기 때문에 병렬처리를 위해 증가하는 하드웨어 크기에 비해 그 효율이 떨어지기 때문에 한 번에 하나의 확률 값만 처리하도록 설계하였다. 하나의 슬라이스 타입에 최대 154개의 init_value가 존재하기에 초기화에 걸리는 cycle은 156 cycle이 소요된다.

V. 문맥 모델 결정부

문맥 모델 결정부는 문맥 모델 결정 모듈과 이웃 블록 데이터 처리부, 이웃 블록의 데이터를 저장하는 Single port RAM 2개로 이루어져 있다. 대부분의 구문요수는 구문요수의 데이터만으로 문맥 모델을 결정할 수 있으나, split_cu_flag나 cu_skip_flag, regidual coding data들은 해당 빈이 속한 블록과 이웃 블록의 데이터를 통해 문맥 모델을 결정한다. 여기에서 split_cu_flag나 cu_skip_flag 등의 용어는 HEVC의 표준에서 제공하는 구문요수들의 명칭이다. 그렇기 때문에 문맥 모델의 결정에 앞서 이웃 블록의 정보가 처리 되어야 한다. 그림 7은 이러한 문맥 모델 결정부의 파이프라이닝을 보여준다. 문맥 모델 결정 모듈의 경우 구문요수의 ID와 Depth 등의 정보와 이웃 블록 값의 LUT로 설계 가능하다. 또한 last_sig_coeff_x(y)_prefix와 coeff_abs_level_greater_flag1(2)의 경우 몇 개의 register와 LUT만으로 이웃 블록의 데이터를 구할 수 있으나, split_cu_flag와 cu_skip_flag, coded_sub_block_flag, sif_coeff_flag의 경우 주변 블록에 저장된 데이터를 통해 이웃 값을 계산하기 때문에 이를 저장할 메모리의 효율적인 설계가 요구된다.

1. split_cu_flag & cu_skip_flag

split_cu_flag와 cu_skip_flag의 이웃 값을 계산하기 위해서는 그림 8과 같이 slice 단위로 저장된 위와 왼쪽의 CTU의 인접한 cu의 depth, skip 정보를 필요로 한다. slice 단위의 data를 모두 저장할 경우 메모리의 크기가 불필요하게 커지기 때문에 이를 줄이기 위해 Row RAM과 Column register를 이용하여 하드웨어

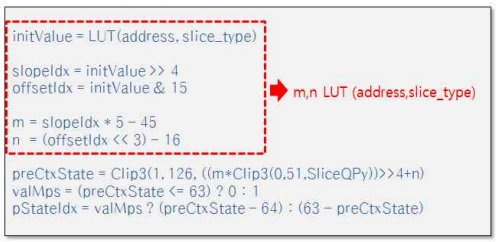

Fig. 5. Initialization operation and m,n LUT.
그림 5. 초기화 연산과 m,n LUT.

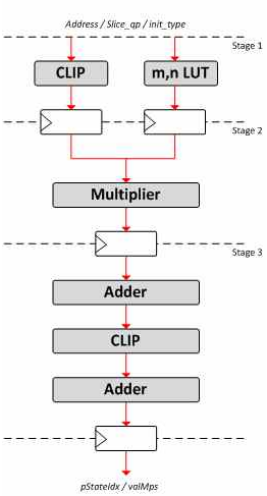

Fig. 6. Pilelining of the initializer.
그림 6. 초기화부의 파이프라이닝.

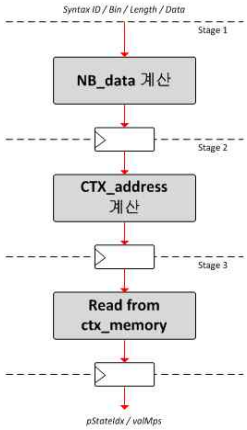

Fig. 7. Pipelining of the context model decider.
그림 7. 문맥 모델 결정부의 파이프라이닝.

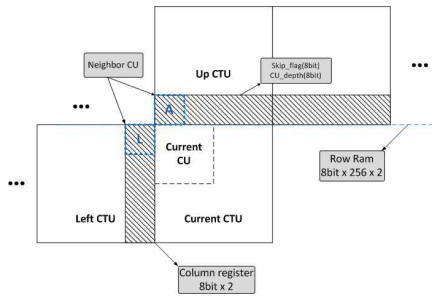

Fig. 8. Row ram and column register.
그림 8. 로우 램과 컬럼 레지스터.

를 설계하였다. Row RAM은 8bit * 256 (4K 기준)의 Single Port RAM으로 설계되었으며, Column register는 8bit의 register로 메모리의 크기를 크게 절약할 수 있다. 저장된 데이터는 현재 CTU의 좌표에

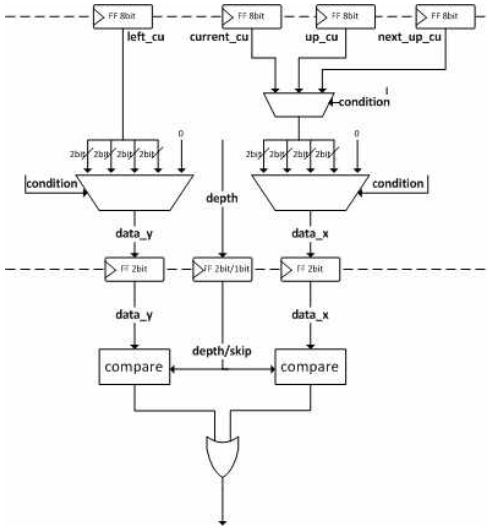

Fig. 9. Split_cu_flag & cu_skip_flag module.
그림 9. Split_cu_flag & cu_skip_flag 모듈.

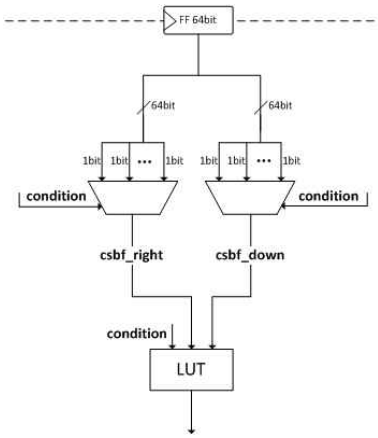

Fig. 10. Coded_sub_block_flag & sig_coeff_flag module.
그림 10. Coded_sub_block_flag & sig_coeff_flag 모듈.

의해 불러져 FlipFlop에 저장되며, 저장된 데이터는 그림 9와 같이 cu의 좌표에 의해 멀티플렉싱된 후 현재 cu의 값과 비교하여 이웃 값을 출력하게 된다.

2. coded_sub_block_flag & sig_coeff_flag

coded_sub_block_flag (csbf)와 sig_coeff_flag (scf)의 경우 cu단위로 데이터를 저장하여 해당 서브블록의 오른쪽과 아래의 서브블록의 csbf 값을 통해 이웃 값을 계산한다. 이를 저장하기 위해 메모리는 최대 cu 사이즈를 기준으로 한 64bit 레지스터로 설계되었다.

저장된 64bit의 데이터는 해당 서브블록의 좌표를 통해 멀티플렉싱 되어 아래와 왼쪽의 csbf 값으로 출력되고 이 값들을 통해 LUT를 거쳐 이웃 값을 계산하게 된다. 그림 10은 csbf와 scf의 모듈을 보여준다.

Ⅵ. 구현 및 검증

제안하는 문맥 모델러는 Verilog HDL로 디자인 되었다. HEVC 표준 모델인 HM [3] 을 통해 테스트 벡터를 만들었으며, 이를 입력으로 하여 동작을 검증하였다. 테스트 벡터는 표 1에 나타난 것과 같이 HEVC의 성능 평가에 일반적으로 사용되는 4개의 클래스마다 4개의 인코더 환경 (encoder configuration)을 적용해 총 16개의 테스트 벡터를 사용하여 검증하였다.

Table 1. Test vector.
표 1. 테스트 벡터.

Sequence	Class A	Class B	Class C	Class D
size	2560×1600	1920×1080	832×480	416×240
frame rate	60	24	60	50
frames	50			
encoder configuration	low_delay, low_delay_P, random access, all_intra			

디자인된 문맥 모델러는 0.18um 공정으로 설계되었으며, CAD 툴은 IDEC으로부터 지원받았다. 설계된 하드웨어는 표 2와 같이 메모리를 포함하여 총 29832개의 게이트로 이루어져 있으며, 최대 동작속도는 200MHz, 최대 처리율은 200Mbin/s이다.

Table 2. Synthesis results.
표 2. 합성 결과.

Technology		0.18um
Max Operating Frequency		200MHz
Max Throughput		200MBin/s
Gates	Total	29832 gates
	Single Port RAM	10433 gates
	Logic	19399 gates

Ⅶ. 결론

본 논문에서는 효율적인 메모리 설계로 저면적 하드웨어와 적절한 파이프라이닝과 LUT 설계 등을 이용하여 보다 효율적으로 확률을 모델링하여 연산량이 적은 하드웨어를 설계하기 위한 아키텍처를 제안하였다. 이는 실시간 인코딩이 요구되는 HEVC 영상 압축 시스템에 보다 효율적으로 응용될 수 있다.

Ⅷ. 결론

본 논문에서는 효율적인 메모리 설계로 저면적 하드웨어와 적절한 파이프라이닝과 LUT 설계 등을 이용하여 보다 효율적으로 확률을 모델링하여 연산량이 적은 하드웨어를 설계하기 위한 아키텍처를 제안하였다. 이는 실시간 인코딩이 요구되는 HEVC 영상 압축 시스템에 보다 효율적으로 응용될 수 있다.

References

[1] 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), Jan. 2013.

[2] HEVC software repository HM-11 reference model at https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-11.0-dev/

[3] S. Han, W. Nam, and S. Lee, “Design of Low-Area HEVC Core Transform Architecture”, Journal of IKEEE, vol. 17, no. 2, pp. 119-128, Jun. 2013.

[4] J. Lee and S. Lee, “8×8 HEVC Inverse Core Transform Architecture Using Multiplier Reuse”, Journal of IKEEE, vol. 17, no. 4, pp. 570-578, Dec. 2013.

[5] B. Peng, D. Ding, X. Zhu, and L. Yu, “A hardware CABAC encoder for HEVC”, IEEE International Symposium on Circuits and Systems, pp. 1372 - 1375, 2013.

[6] D. Pham, J. Moon and S. Lee, “Hardware Implementation of HEVC CABAC Binarizer”, Journal of IKEEE, vol. 18, no. 3, pp. 356-361, Sep. 2014.

[7] D. Pham, J. Moon, D. Kim and S. Lee, “Hardware Implementation of HEVC CABAC Binary Arithmetic Encoder”, Journal of IKEEE, vol. 18, no. 4, pp. 630-635, Dec. 2014.

BIOGRAPHY

Doohwan Kim (Student Member)

2015 : BS degree in Electronic Engineering, Soongsil University.
2015~Now : MS candidate in Electronic Engineering, Soongsil University.
<Main Interest> HEVC, Multimedia SoC Design

Jeonhak Moon (Student Member)

2005 : BS degree in Electronic Engineering, Soongsil University.
2008 : MS degree in Electronic Engineering, Soongsil University.
2008~Now : PhD candidate in Electronic Engineering, Soongsil University.
<Main Interest> HEVC, Multimedia SoC Design

Seongsoo Lee (Life Member)

1991 : BS degree in Electronic Engineering, Seoul National University.
1993 : MS degree in Electronic Engineering, Seoul National University.
1998 : PhD degree in Electrical Engineering, Seoul National University.
1998~2000 : Research Associate, University of Tokyo
2000~2002 : Research Professor, Ewha Womans University
2002~Now : Associate Professor in School of Electronic Engineering, Soongsil University
<Main Interest> HEVC, Low-Power SoC Design, Multimedia SoC Design, Battery Management