

논문 2015-52-5-16

HEVC 구문요소에 적응적인 파이프라인-병렬 CABAC 복호화기 설계

(A Design of Pipelined-parallel CABAC Decoder
Adaptive to HEVC Syntax Elements)

배 봉 희*, 공 진 흥**

(Bong-Hee Bae and Jin-Hyeung Kong[Ⓞ])

요 약

본 연구에서는 다양한 HEVC 구문요소들을 적응적으로 파이프라인 및 병렬 처리할 수 있는 CABAC 복호화기 아키텍처를 설계 및 구현하였다. CABAC는 높은 압축률을 제공하지만, 구문요소 단위 순차적 복호화와 문맥간 강한 데이터 종속성, 빈 단위 복호화 과정 때문에 고성능 복호화 처리를 어렵게 한다. CABAC의 복호화 처리 성능을 높이기 위하여 연속된 flag 타입의 구문요소에 대해서는 다음에 복호될 구문요소들을 선행 연산하여 적응적으로 파이프라인 처리하였고, 멀티빈으로 구성된 구문요소는 최대 3개 빈까지 병렬 처리하는 고성능 구조를 설계하였다. 또한 이진산술복호기를 가속화하기 위해 문맥모델 업데이트와 재정규화를 선행 병렬 연산하고, 복호화 결과값에 따라 선택해서, 이진산술복호기의 임계 지연시간을 개선하였다. 제안하는 HEVC CABAC 아키텍처는 최대 1.01bins/cycle의 처리 성능으로 기존 구조대비 약 2배의 가속화 성능을 갖는다. 65nm ASIC 합성 결과 224M bins/sec.의 복호화 성능을 보이며, QFHD영상의 실시간 처리를 가능하게 하였다.

Abstract

This paper describes a design and implementation of CABAC decoder, which would handle HEVC syntax elements in adaptively pipelined-parallel computation manner. Even though CABAC offers the high compression rate, it is limited in decoding performance due to context-based sequential computation, and strong data dependency between context models, as well as decoding procedure bin by bin. In order to enhance the decoding computation of HEVC CABAC, the flag-type syntax elements are adaptively pipelined by precomputing consecutive flag-type ones; and multi-bin syntax elements are decoded by processing bins in parallel up to three. Further, in order to accelerate Binary Arithmetic Decoder by reducing the critical path delay, the update and renormalization of context modeling are precomputed parallel for the cases of LPS as well as MPS, and then the context modeling renewal is selected by the precedent decoding result. It is simulated that the new HEVC CABAC architecture could achieve the max. performance of 1.01 bins/cycle, which is two times faster with respect to the conventional approach. In ASIC design with 65nm library, the CABAC architecture would handle 224 Mb/sec, which could decode QFHD HEVC video data in real time.

Keywords: HEVC CABAC decoder, Adaptively pipelined-parallel manner, Flag-type syntax elements, Sequential computation, Strong data dependency, LPS and MPS, Update and renormalization, Context modeling, Real time decoding

* 정회원, ** 평생회원, 광운대학교 컴퓨터공학과

(Department of Computer Engineering, Kwangwoon University)

Ⓞ Corresponding Author(E-mail: kongjh@kw.ac.kr)

※ 이 논문은 2013년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

※ This work was supported by a grant from the IT R&D program of MOTIE/KEIT

[10048285, Improving Industrial Infrastructure through embedded system research and development]

Received ; March 24, 2015

Revised ; April 15, 2015

Accepted ; April 22, 2015

I. 서 론

UHD 영상 미디어의 발전으로 대용량 동영상 데이터를 효율적으로 전송하고 서비스하기 위한 HEVC(High Efficiency Video Coding) 표준안^[1]이 완성되었다. HEVC는 과거 H.264/AVC^[2]와 비교해서 압축율을 약 2배 개선^[3]하였으나, 복잡도와 연산량이 크게 증가하여 4/8K-UHD급 영상의 실시간 복호화가 어려운 한계를 보이고 있다. 따라서 UHD급 HEVC 압축영상의 실시간 복호화 문제를 해결하기 위한 고성능 HW 복호화기의 설계 및 구현이 필요하다. HEVC 복호화 기능 모듈의 처리시간을 분석한 결과^[4]에 따르면, 엔트로피 복호화 모듈이 움직임 보상 모듈 다음으로 많은 연산 처리 시간을 요구하는 것으로 확인되었다. 따라서 실시간 HEVC HW 복호화기의 구현을 위해서는 엔트로피 CABAC(Context-based Adaptive Binary Arithmetic Coding) 복호화를 위한 고성능 실시간 설계 및 구현이 요구되고 있다.

HEVC 및 H.264/AVC 엔트로피 복호화의 핵심 모듈인 CABAC는 문맥기반의 이진 발생 확률 정보를 통해 비트스트림을 복호화하여 구문요소의 값을 출력하는 기능 모듈이다. CABAC는 현재 처리 결과를 바탕으로 다음 연산 과정이 순차적으로 결정되는 문맥기반(context-based) 복호화 과정이다. 따라서 입력 데이터 비트 단위로 순차적으로 복호화되는 제약을 갖게 되며, 처리 결과가 다음 입력의 복호화에 반영되는 문맥기반 데이터 종속성을 보이고 있다. 또한 CABAC 구문요소는 flag 타입에서부터 멀티빈까지 가변적으로 구성되기 때문에 비트 단위 복호화 연산 알고리즘을 바탕으로 처리된다. 이와 같은 CABAC 복호화 과정의 연산 오버헤드를 효율적으로 해결하기 위한 고성능 CABAC 복호화 HW 아키텍처 연구^[5~11]들이 활발하게 제안되고 있다.

실제로 CABAC의 구문요소가 순차적으로 비트 단위로 복호화되는 한계를 해결하기 위해서 H.264/AVC에서는 복호화될 다음 구문요소를 예측하여 파이프라인^[5] 및 병렬^[6~7]처리하는 CABAC HW 아키텍처들이 개발되었다. 그러나 HEVC 구문요소의 종류가 H.264/AVC와 비교해서 약 2.3배 다양하게 증가하였으며 복호화 알고리즘 또한 매우 복잡해서, HEVC CABAC에 적합한 새로운 예측 알고리즘이 필요하다. HEVC 구문요소

는 다수의 flag 타입 구문요소들로 구성되어 있으며 실제의 비디오 부호화 스트림에서 높은 발생 빈도를 보이고 있다. 본 연구는 HEVC의 연속적 flag 타입 구문요소에 대한 문맥모델러(context modeler) 및 이진산술복호기(binary arithmetic decoder; BAD)의 2단 파이프라인-병렬 아키텍처를 설계하였다. 연속 2개의 flag 구문요소는 3병렬-2단 파이프라인 문맥모델러 및 이진산술복호기를 통해서 2사이클 이내에 처리된다. 이후 연속된 3번째 flag 구문요소는 선행의 연속 2개 flag 구문요소에 대한 2번째 사이클에서 동시에 4병렬 문맥모델링을 처리한 다음 4가지 결과 중에서 선행 2 연속 flag 구문요소의 복호화 결과 값으로 하나를 선택하여 2단계 이진산술복호기 연산이 3번째 사이클에서 가능하게 한다. 이와 같은 아키텍처를 통해서 2~3개 및 그 이상의 연속 flag 구문요소 복호화를 위한 2단 파이프라인 연산 처리가 구문요소에 적응적으로 stall없이 1 bin/cycle의 성능을 유지할 수 있도록 하였다.

CABAC의 이진산술복호기는 재정규화(renormalization) 및 문맥모델 메모리 갱신(context model update) 과정 때문에 선행(precedent) 및 다음(next) 입력 빈 간의 순차적 데이터 종속성을 갖고 있다. 문맥모델 갱신에 의한 순차적 데이터 종속성을 해결하기 위해서 이진산술복호기 연산이 끝나면 갱신된 문맥모델을 다음 이진산술복호기 연산에 바로 전달하는(context model forwarding) 파이프라인 아키텍처^[8]가 제안되었다. 그러나 이진산술복호기 연산 후에 다음 이진산술복호기 연산에 필요한 range와 offset 정보를 재정규화시키는 과정 때문에 입력 빈들에 대한 병렬 처리가 제한된다. 한 번에 최대 16개의 멀티빈까지 병렬 복호화 처리를 위한 아키텍처^[9]가 개발되었으나, 실제로 병렬 복호화되는 멀티빈의 평균 개수가 2.27에 불과한 문제점을 보이고 있다. 본 연구에서는 1개 이상의 빈으로 구성된 구문요소를 복호화할 때 최대 3개의 멀티빈을 병렬 처리하는 고성능 아키텍처를 설계하였다. 멀티빈 복호화를 위한 문맥모델러의 3병렬화는 연속된 flag 타입 구문요소를 위한 4병렬 문맥모델러 구조를 이용해서 연산 처리할 수 있다. 하지만 이진산술복호기는 LPS(Least Probable Symbol)와 MPS(Most Probable Symbol)에 대한 재정규화 및 문맥모델 업데이트에 따른 데이터 종속성 때문에 병렬 처리가 불가능하다. 이를 위해서 재정규화 및 업데이트를 이진산술 연산 처리

에 대해서 선행 병렬 연산하여, 이진산술 복호화 임계 지연 시간을 감소시키는 연산 과정을 통해서 고속화하였다.

본 논문의 II장에서는 HEVC CABAC 기존 연구들의 문제점을 분석하고, 고성능 HW 설계를 위한 파이프라인-병렬 가속화 방안을 기술한다. III장에서는 HEVC 구문요소에 적응적인 파이프라인-병렬 CABAC 아키텍처의 구조와 동작에 대해 설명한다. 실험 결과를 통해 제안된 고성능 HEVC CABAC 복호화 HW 아키텍처의 기능 및 성능을 IV장에서 검증한다. 마지막 V장에서는 본 연구의 결론을 맺는다.

II. HEVC CABAC 가속화 방안

1. CABAC 복호화 알고리즘

HEVC CABAC는 이진 발생 확률 정보를 통해 비트 스트림을 복호화하여 구문요소의 값을 출력하는 엔트로피 디코더의 핵심 모듈로 그림 1과 같은 데이터패스를 가진다. 실제로 CABAC 데이터패스에서 이진산술복호기의 결과에 따라서 문맥모델러(context modeler)의 메모리를 갱신하는 피드백(feedback) 데이터패스가 CABAC 연산 처리 과정을 고속화할 때 다음의 세 가지 한계점을 발생시킨다.

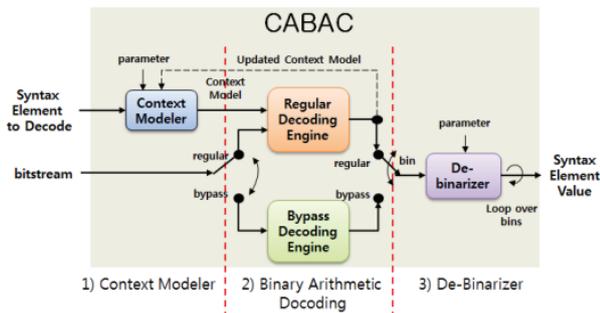


그림 1. CABAC 복호화의 데이터패스
Fig. 1. Datapath of CABAC decoder.

- 구문요소 단위 순차적 복호화(sequential decoding procedure/syntax element): CABAC를 통해 복호화되는 구문요소는 현재 복호화된 결과에 따라서 다음에 복호화될 구문요소를 결정하기 때문에, 구문요소가 순차적으로 복호화되는 문제를 갖는다.
- 강한 데이터 종속성(strong data dependency): CABAC의 이진산술복호기는 문맥모델 메모리로 부

터 읽어온 문맥모델을 이진산술복호화 연산 처리해서 그 결과(MPS 또는 LPS)에 따라 문맥모델을 업데이트한다. 이러한 메모리 갱신 과정으로 인하여 연속된 이진산술복호화 연산에서 선행 처리 결과를 다음 처리 과정에 반영시키는 데이터 종속성이 발생한다.

- 빈 단위 복호화 과정(decoding procedure/bin): CABAC의 이진산술복호기는 빈 단위로 range 및 offset 정보를 재정규화하고 업데이트하는 순차적 연산 처리를 수행하기 때문에 빈 단위의 복호화 과정을 거치게 된다.

2. HEVC CABAC 파이프라인-병렬 가속화 알고리즘

가. 연속적 Flag 타입 구문요소들의 파이프라인-병렬 처리
H.264/AVC의 CABAC에서는 다음에 복호화될 구문요소를 예측하여 파이프라인 처리하는 구조^[5]와 이진산술복호화 결과에 따라 업데이트된 문맥모델을 캐시 메모리에 저장한 뒤 다음 이진산술복호화 동작에 바로 전달(forwarding)하는 파이프라인 구조^[8] 등이 제안되었다. 그러나 HEVC는 CABAC를 통해 복호화되는 구문요소의 종류가 기존 H.264/AVC 대비 약 2.3배 증가하였으며, 높은 발생 빈도를 보이는 다수의 flag 타입 구문요소를 포함하고 있다. HEVC flag 타입의 구문요소(single bins)는 복호화 결과값이 항상 '0' 또는 '1'이기 때문에 다음에 복호화될 가능성이 있는 2개 구문요소를 예측할 수 있다.

그림 2와 같이 분기를 발생하는 첫 번째 flag 타입의 구문요소 값이 출력될 때 두 번째 구문요소의 결과값이 선택될 수 있다. 첫 번째 구문요소 복호화 결과값 '0'과 '1'에 대해서 두 번째 구문요소 복호화 과정 2가

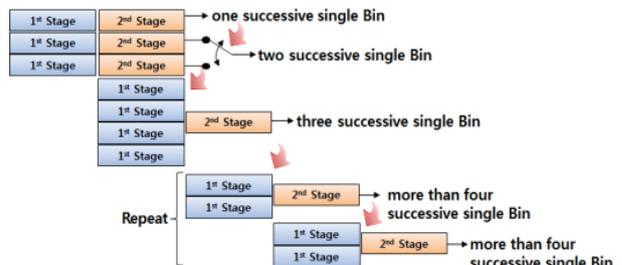


그림 2. 연속적 flag 타입 구문요소들의 파이프라인-병렬 연산
Fig. 2. Pipelined-parallel computations of successive flag type syntax elements.

지 경우를 병렬 연산한다. 세 번째 구문요소를 위한 문맥모델링은 현재 복호화중인 첫 번째와 두 번째 구문요소 2개에서 가능한 4가지 {"00", "01", "10", "11"}의 flag 조합을 위한 문맥모델링들이 병렬 처리된다. 네 번째 이후의 구문요소부터는 이전 구문요소에서 분기된 2가지('0', '1')의 구문요소 복호화를 위한 문맥모델링을 병렬 처리한 후 하나를 선택하는 파이프라인 과정이 반복된다.

나. 멀티빈 구문요소의 파이프라인-병렬 처리

기존 연구에서는 다음에 복호화될 구문요소를 예측하여 2/16 병렬 처리하는 구조^[6-7]들을 H.264/AVC를 위하여 개발하였지만, HEVC에 적합하게 병렬 처리 취소를 방지할 수 있는 새로운 알고리즘 및 구조 설계를 요구한다.

CABAC의 비트 단위 복호화 연산 처리를 가속시키기 위해 1개 이상의 멀티빈으로 구성된 구문요소를 복호화할 때 최대 3개의 빈 단위 병렬 연산을 파이프라인 처리한다. 멀티빈 구문요소의 경우 복호화되는 빈 사이에 구문요소 정보가 동일하고 빈 인덱스 또한 예측 가능하기 때문에 문맥모델러의 병렬 처리가 가능하다. 그림 3과 같이 1~3번째 빈을 복호화하기 위한 병렬 문맥모델러의 수행이 끝나고 나면, 4~6번째 빈을 위한 문맥모델러가 파이프라인-병렬 동작된다. 예를 들어 5개의 빈으로 구성된 멀티빈을 복호화될 경우 두 번째 복호화된 4~6번째 빈의 결과에서 선행 2개만 사용되고 마지막 빈의 결과는 버려지고 파이프라인 stall이 발생된다.

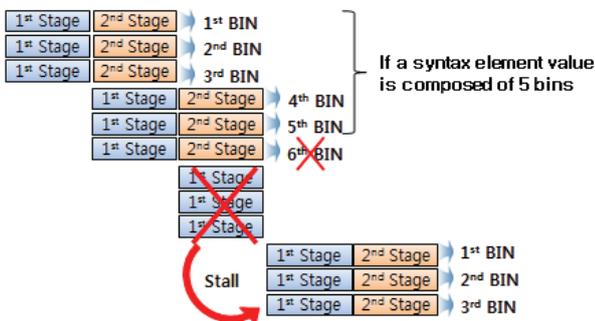


그림 3. 3 멀티빈 파이프라인-병렬 연산
Fig. 3. Pipelined-parallel computation for 3 multi-bins.

다. 이진산술복호화 임계 지연 시간 가속화 방안

이진산술복호기는 그림 4a)와 같이 문맥모델 업데이트에 의한 메모리 read/write를 수행하며 range와

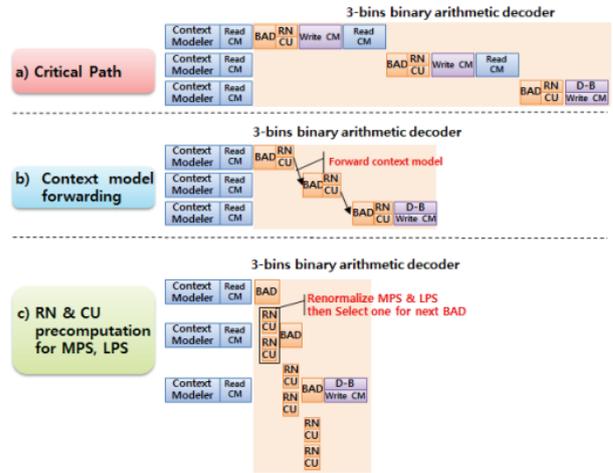


그림 4. 3병렬 이진산술복호기의 가속화 방안
Fig. 4. Acceleration scheme for 3 parallel BAD.

offset 정보에 대한 재정규화 과정을 처리하게 된다. 그 결과 문맥모델러의 병렬 처리에도 불구하고 이진산술복호기의 임계 지연 시간이 과도하게 길어지는 문제가 발생한다. 연속된 이진산술복호화의 임계 지연 시간을 최소화하기 위해서 그림 4b)와 같이 복호화될 연속된 빈들이 동일한 문맥모델을 사용할 경우에 다음 이진산술복호화 연산에 갱신될 문맥모델을 바로 전달(forwarding)하는 구조^[8]를 통해서 지연을 감소시킬 수 있다.

하지만 1단계 파이프라인 동작에서 병렬 처리된 문맥모델러에 맞추어 2단계 이진산술복호기를 가속화하는 것이 요구된다. 그림 4c)와 같이 재정규화와 문맥모델 업데이트 동작을 LPS와 MPS에 대하여 선행 병렬 처리 후 이진산술복호화 결과(MPS or LPS)에 따라서 갱신된 문맥모델, range, offset 정보를 선택하여 2단계 이진산술복호기의 처리 성능을 고속화하였다.

III. HEVC CABAC 고성능 아키텍처

1. 파이프라인-병렬 아키텍처

HEVC CABAC 고성능 아키텍처는 그림 5와 같다. 제안된 HEVC CABAC는 문맥모델러(context modeler; CM), 이진산술복호기(binary arithmetic decoder; BAD) 및 역이진화기 (debinarizer; DB)등 주요 3블럭으로 구분된다. 그림 6은 CABAC 주요 블록들의 2단 파이프라인-병렬 동작을 보인다.

HEVC 구문요소에 적응적인 병렬 처리는 연속적인

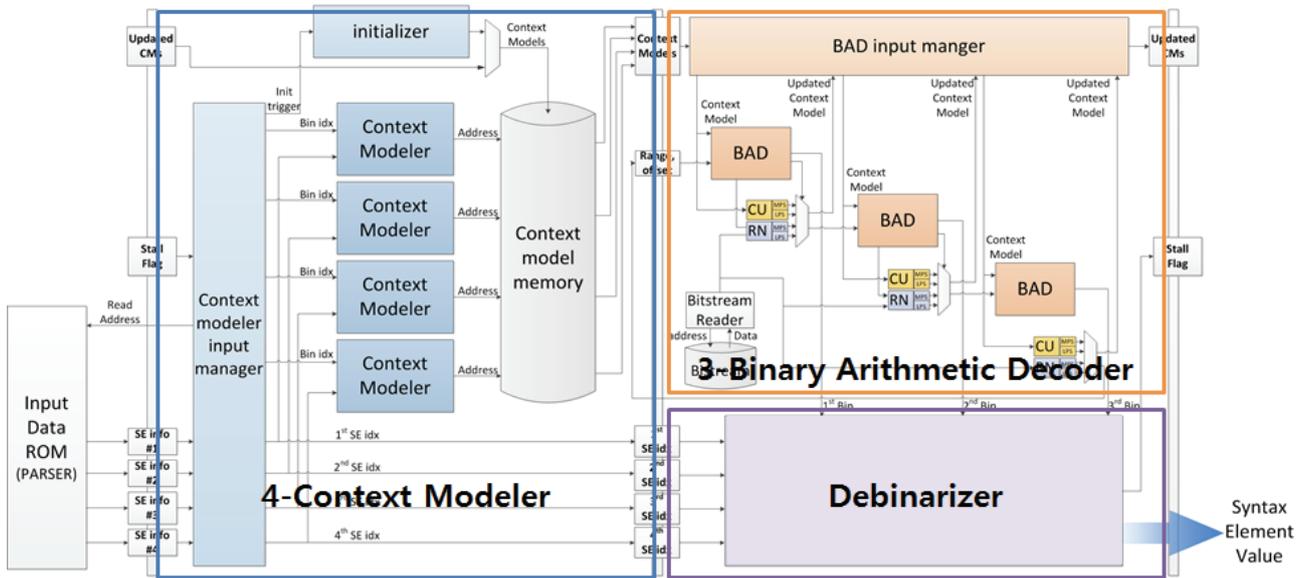


그림 5. HEVC CABAC 고성능 아키텍처
Fig. 5. High performance architecture of HEVC CABAC.

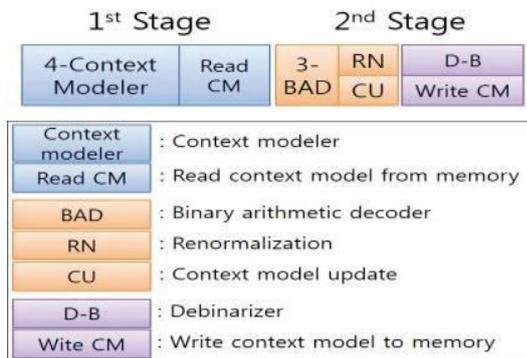


그림 6. HEVC CABAC 복호화기 2단 파이프라인
Fig. 6. 2 stages pipeline of HEVC CABAC decoder.

flag 타입 구문요소와 멀티빈 구문요소로 구분되어 수행된다. 연속적인 flag 타입 구문요소에 대한 가속화 방안으로 제안된 그림 5의 2단 파이프라인 동작에서 1단계 문맥모델러가 그림 2의 파이프라인-병렬 복호화를 위해서 4병렬 처리까지 가능한 구조로 설계되었다. 또한 멀티빈 구문요소를 그림 3과 같이 병렬 처리하기 위해서 1단계의 4병렬 문맥모델러는 최대 3개의 문맥모델을 병렬 처리하고, 출력된 문맥모델을 2단계의 이진산술복호기 블록의 입력 데이터 매니저(BAD input manger)에 전달한다. 만약 연속된 이진산술복호화 연산에서 동일한 문맥정보를 사용할 경우에 이전 이진산술복호화 연산에서 업데이트된 문맥모델을 전달받는다. 반면에 서로 다른 문맥모델을 사용할 경우에는 메모리

로부터 검색된 문맥모델을 사용하여 한번에 최대 3개의 빈까지 병렬복호화한다. 마지막 역이진화기에서는 복호화된 빈과 이진화 테이블과의 비교를 통해 구문요소 값의 구성이 가능한지 확인한다. 현재까지 복호화된 빈 스트링을 통해 구문요소 역 이진화가 가능하다면 구문요소 값(syntax element value)을 출력하고, 다음 구문요소 복호화를 위해 빈 인덱스를 초기화한다.

2. 구성 블록 설계

CABAC 아키텍처(그림 5)의 문맥모델러 블록은 입력 데이터 매니저와 문맥모델러, 문맥모델 메모리 등으로 그림 7과 같이 구성된다. 그림 7a) 문맥모델러 입력 데이터 매니저는 분기 확인 모듈(branch check)에 현재 복호화할 구문요소(In_SeIdx)가 입력되면 멀티빈인지 flag 타입의 구문요소인지 구분한 다음에 그 결과(branch flag)를 FSM에 전달한다. Flag 타입의 구문요소를 복호화할 경우 연속된 횟수에 따라 FSM의 state가 변경되며, 다음 구문요소 예측 모듈(next syntax element finder)을 통하여 현재 구문요소와 함께 병렬 복호화될 구문요소(SeIdx)를 출력한다. 멀티빈으로 구성된 구문요소는 현재 복호화될 구문요소와 함께 빈 인덱스 예측 모듈(bin index predictor)을 통해 3개의 빈 인덱스(BinIdx)를 예측하여 출력한다.

그림 7b) 문맥모델러의 동작은 빈 인덱스(BinIdx)와

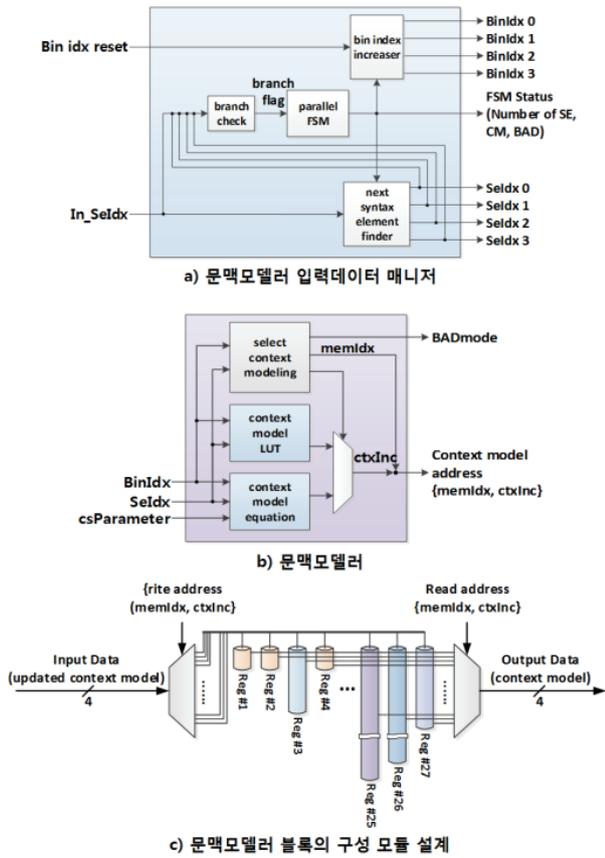


그림 7. 문맥모델러 블록의 구성 모듈 설계
Fig. 7. Component module design in context modeler.

복호화될 구문요소(SelIdx)를 입력받아 문맥모델링 선택 모듈(select context modeling)을 통해 현재 구문요소 복호화에 필요한 메모리 주소 정보(memIdx)와 이진산술복호화 모드(BAD mode)를 출력한다. 동시에 문맥모델 메모리 주소를 구성하는 ctxInc는 복호화될 구문요소(SelIdx)마다 정의된 LUT(context model LUT) 또는 수식(context model equation)을 통하여 계산된다.

마지막 그림 7c) 문맥모델 메모리는 문맥모델러에서 계산된 메모리 주소(memIdx, ctxInc)에서 필요한 문맥모델을 검색한다. memIdx는 현재 구문요소에 필요한 문맥모델 메모리를 선택하고 ctxInc를 통해 메모리에 저장된 문맥모델을 검색 혹은 갱신한다. 한번에 최대 4개의 문맥모델을 출력해야 할 뿐만 아니라 업데이트된 문맥모델을 저장해야 하기 때문에 HEVC 구문요소에 대해서 다중 접근이 가능한 27개 레지스터로 각각 구현되었다.

그림 8에서 이진산술복호기(regular decoding engine)는 정의된 LUT(LPS range finder)를 통해 문맥정보

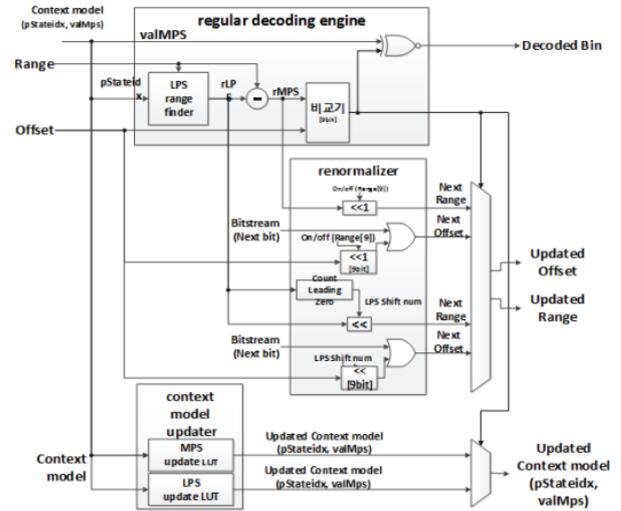


그림 8. 이진산술복호기 설계
Fig. 8. Binary arithmetic decoder design.

(pStateIdx)를 LPS range(rLPS)로 변환하며 출력된 LPS range와 현재 range정보를 통해 MPS range(rMPS)를 출력한다. 다음에 rMPS와 offset의 크기를 비교하여, offset이 rMPS보다 크면 '1'이고 반대의 경우 '0'을 출력하여, MPS와 비교 결과값('1' or '0')을 XOR하여 bin을 복호화한다. 문맥모델 업데이트(context model updater) 동작은 이진산술복호기의 정규 복호화 엔진(regular decoding engine) 연산의 시작과 함께 문맥모델(context model)이 입력되면 정의된 LUT를 통해서 MPS와 LPS 두 가지 모두 업데이트되며, 정규 복호화 결과에 따라서 업데이트된 두 개의 문맥모델(updated context model) 중 하나가 선택 출력된다. 재정규화(renormalizer) 동작은 정규 복호화 동작에서 LPS range(rLPS)가 출력되고 나면 LPS에 대한 재정규화 처리를 시작하며, 이후 MPS range(rMPS)가 출력되면 이어서 MPS range에 대한 재정규화가 병렬 처리된다. 이와 같이 재정규화된 LPS range나 MPS range(rLPS, rMPS)가 정규 복호화 엔진의 비교 결과에 따라서 하나가 선택 출력된다.

IV. 실험 및 고찰

1. HEVC 구문요소 분포도 분석

설계된 CABAC 아키텍처는 HEVC 구문요소에 적응적으로 병렬 및 파이프라인 처리하기 위해서 CABAC를 통해 복호화되는 구문요소의 분포도를 고려하였다.

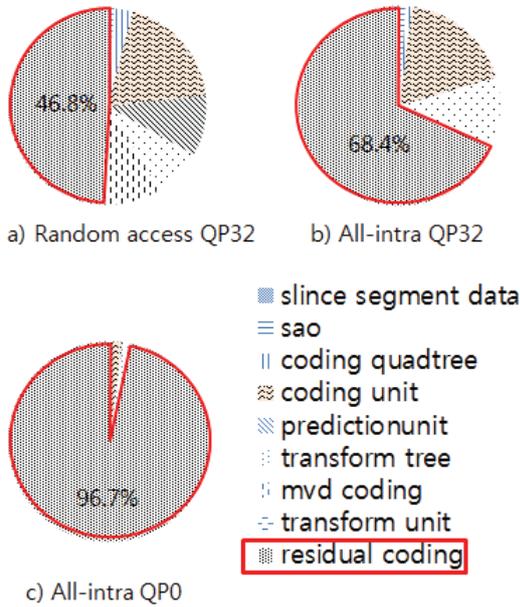


그림 9. <BasketballPass>에서 구문요소 그룹의 분포도 분석
Fig. 9. Distribution analysis of syntax element group in <BasketballPass>.

그림 9와 같이 각기 다른 configuration과 QP에 대한 구문요소의 분포도를 비교 분석한 결과에 따르면 1) 그림 9 a)와 b) QP32에서 All-intra 설정으로 코딩된 영상이 Random access 설정과 비교해서 잔여 화소 구문요소(residual coding)의 비율이 크게 증가한다는 것과, 2) QP를 다른 값(32와 0)으로 적용하여 코딩된 영상의 구문요소 분포도를 비교하였을 때 낮은 QP0의 영상에서 잔여 화소 구문요소가 차지하는 비중이 훨씬

크다는 것을 확인할 수 있었다. 또한 All-intra configuration에서 다른 QP의 그림 9 b)와 c)에서 각각 잔여화소의 빈도수에 대해 빈의 개수(분포도)를 분석한 표 1에서는 3) QP0의 영상이 QP32영상과 비교할 때에 coeff_abs_level_remaining 등 멀티빈 구문요소의 빈의 비중과 숫자가 급격히 증가하는 것을 발견할 수 있다. 결과적으로 4) 멀티빈 구문요소에 대하여 적용적으로 3병렬 파이프라인-병렬 복호화를 처리하는 구조는 멀티빈의 비중 및 개수가 커지는 All-intra configuration과 작은 QP에서 코딩된 데이터의 복호화 과정에서 높은 성능을 제공할 수 있다.

2. 복호화 성능 검증

HEVC의 configuration 설정 및 QP값 변화 등을 반영해서 개발된 파이프라인-병렬 CABAC 복호화 아키텍처의 성능을 검증하고자 한다. 표 2는 동일한 영상 <BasketballDrive>을 다른 3가지 영상 코딩 설정 변수 (All-intra, Low delay, Random access) 그리고 4가지의 QP값(22, 27, 32, 37)으로 부호화된 비디오 스트림에 대해서 복호화 성능들을 비교해서 보인다.

실험 결과에서 제안된 CABAC 아키텍처는 1) 최소 0.87 bins/cycle에서 최대 1.01 bins/cycle까지 bins를 복호화할 수 있음을 보인다. 따라서 HEVC 구문요소에 적용적인 파이프라인-병렬 CABAC 구조는 기존의 HEVC CABAC 성능(0.5 bins/cycle) 대비 약 2배의 가속화 성능을 나타내고 있다.

HEVC의 configuration 설정에 대하여 파이프라인-병렬 CABAC 구조는 2) 그림 10a)와 같이 All-intra 모

표 1. All-intra configuration의 잔여 화소 구문요소에서 single 및 multi 빈의 구성
Table 1. Composition of single- & multi-bins in residual syntax elements of All-intra configuration.

	구문 요소	Bins ratio		Bins number	
		QP32	QP0	QP32	QP0
Single-Bin	transform_skip_flag	72.04%	50.20%	10,754,769	165,062,200
	coded_sub_block_flag				
	sig_coeff_flag				
	coeff_abs_level_greater1_flag				
	coeff_abs_level_greater2_flag				
	coeff_sign_flag				
Multi-Bins	last_sig_coeff_x_prefix	27.96%	49.80%	4,174,458	163,715,221
	last_sig_coeff_y_prefix				
	last_sig_coeff_x_suffix				
	last_sig_coeff_y_suffix				
	coeff_abs_level_remaining				
Total		100%	100%	14,929,227	328,777,421

표 2. 복호화 성능 비교 실험 결과

Table 2. Experimental results for decoding performance comparison.

Sequence Info.	Sequence	BasketballDrive (500 frames)											
	Config	All Intra	Low Delay	Random Access	All Intra	Low Delay	Random Access	All Intra	Low Delay	Random Access	All Intra	Low Delay	Random Access
	QP	22			27			32			37		
	Total bins (단위: K)	894,390	257,767	221,904	367,047	85,395	74,914	201,038	39,839	35,504	117,469	20,745	19,238
CABAC 2 cycle direct Architecture (conventional)	Total Cycle (단위: M)	1,789	516	444	734	171	150	402	80	71	235	41	38
	Bin/Cycle	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
CABAC pipelined-parallel Architecture (Proposed)	Total Cycle (단위: M)	886	269	226	382	93	80	218	44	39	132	24	22
	Bin/Cycle	1.01	0.96	0.98	0.96	0.92	0.94	0.92	0.9	0.91	0.89	0.87	0.88

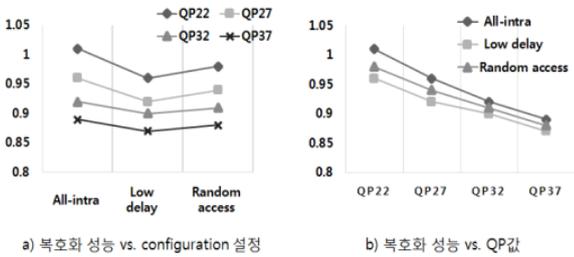


그림 10. Configuration 및 QP에 따른 복호화 성능
Fig. 10. Decoding performances on configuration and QP.

드에서 가장 높은 복호화 성능을 보이며, Low delay에서 가장 낮은 복호화 성능을 나타낸다. 또한 3) 그림 10b)에서는 서로 다른 QP값에 따른 복호화 성능 검증 결과에서는 QP가 낮을수록 가속화 성능(bins/cycle)이 증가하는 것을 확인할 수 있다.

3. 복호화기 기능 검증 및 구현 결과

제안된 CABAC 복호화기 구조는 HW/SW 통합 기능 검증을 위하여 HEVC HM-10.0 SW와 함께 Xilinx Virtex-5 XCTVLX330T에서 HW로 구현되었다. CABAC의 HW/SW 통합 검증의 데이터패스는 그림 11과 같다. 1) 검증을 위해 수정된 HM 레퍼런스 SW 디코더에서 CABAC의 입력 데이터를 PCIe 드라이버를 통해 FPGA에 전달한다. 입력 데이터는 PCIe endpoint를 지나 DMA(Direct Memory Access)를 통해 DDR2 메모리에 저장되고 CABAC IP에서는 메모리에 저장된 입력 데이터를 MIG(Memory Interface Generator)를 통해 읽어온다. 2) CABAC IP HW의 복호화 동작이 끝난 후에는, 복호화된 출력 데이터가 다

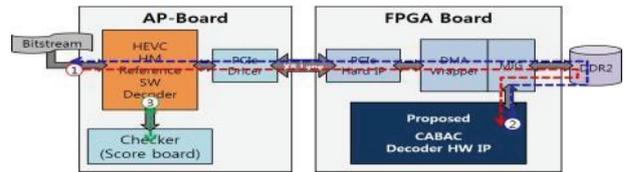


그림 11. HW/SW 통합 검증 시스템
Fig. 11. HW/SW co-verification system.

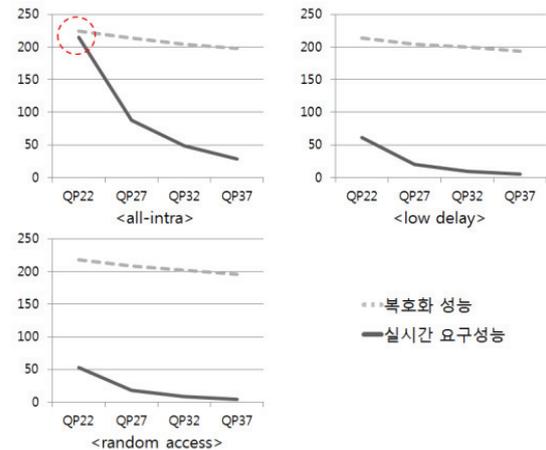


그림 12. 설계 구현된 CABAC의 복호화 성능
Fig. 12. Decoding performances of proposed CABAC design & implementation.

시 DDR2 메모리를 통해서 입력 데이터패스의 역방향으로 HM SW의 레퍼런스 코드에 전달된다. 3) FPGA CABAC로 부터 출력된 데이터를 입력받은 이후에는 HEVC HM SW에서 디코딩을 완료하여 HEVC 복호화 동작을 통합 검증한다.

본 연구의 HEVC CABAC 복호화기 구조는 65nm 라이브러리를 이용한 설계를 통해서 동작 성능을 검증하였다. 설계 결과를 통해서 327K의 logic gate 크기를 갖

고, 222.2Mhz의 최고 동작 주파수에서 성능이 확인되었다. 그림 12는 3종류 configuration 설정 및 4가지 QP 값의 HEVC 부호화 영상에 대해서 복호화 성능을 비교하여 보인다. 빈의 개수가 가장 많은 All-intra configuration 및 QP22의 부호화 QFHD 영상을 실시간으로 복호화하기 위해서는 215Mbins/sec (894,390Kbins/500frame × 30fps × 4(QFHD)) 동작 성능을 필요로 한다. 본연구의 HEVC 구문요소에 적응적인 파이프라인-병렬 CABAC 복호화기는 최대 224Mbins/sec (222.2Mhz × 1.01bins/cycle)의 동작 성능을 갖기 때문에 All-intra 및 QP22의 <BasketballDrive> QFHD영상을 실시간 복호화할 수 있음을 보인다.

V. 결 론

본 논문은 UHD급 영상의 실시간 처리가 가능한 HEVC CABAC 복호화 아키텍처를 제안한다. CABAC는 높은 압축률을 제공하지만, 구문요소 단위 순차적 복호화와 문맥간 강한 데이터 종속성, 빈 단위 복호화 과정 때문에 고성능 복호화 처리를 어렵게 한다. 본 연구에서는 HEVC 잔여 화소 구문요소가 연속적인 flag 타입 구문요소와 멀티빈 구문요소를 많이 포함한다는 HEVC 구문요소 분포도 분석 결과를 반영해서, 연속적 flag 타입 구문요소를 위한 2단 파이프라인 구조와 멀티빈 구문요소를 위한 3병렬 CABAC 복호화기 구조를 설계하였다. 제안된 CABAC 복호화 구조는 멀티빈의 비중 및 개수가 커지는 All-intra configuration 설정과 작은 QP 값에서 높은 성능을 보이는데, 최소 0.87bins/cycle에서 최대 1.01bins/cycle의 처리 성능으로 기존 구조 대비 약 2배의 가속화 성능을 보였다. 65nm 라이브러리 합성 설계 결과에서는 최대 동작 성능 224Mbins/sec를 갖고 QFHD영상을 실시간 처리할 수 있음을 보였다.

REFERENCES

[1] JCT-VC, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)," JCTVC-L1003_v34, Geneva, Switzerland, Jan. 2013

[2] 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, 2003

[3]. Ohm, J. Sullivan, G.J, Schwarz, H, Thiow Keng Tan, Wiegand, T, "Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC)" IEEE Transactions on Circuits and Systems for Video Technology (Volume:22, Issue:12, pp.1669-1684), Dec. 2012.

[4] Bossen, F. Bross, B. Suhring, K. Flynn, D. , "HEVC Complexity and Implementation Analysis", IEEE Transactions on Circuits and Systems for Video Technology (Volume:22 , Issue:12, pp.1685 - 1696), Dec. 2012

[5] W.H. Son and I.C. Park, "Prediction-based Real-time CABAC Decoder High Definition H.264/AVC", Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium, pp.33-36, Seattle, WA, May. 2008.

[6] Jian-Wen Chen, and Youn-Long Lin, "A High-performance Hardwired CABAC Decoder for Ultra-high Resolution Video", IEEE Transactions on Consumer Electronics, (Volume:55 , Issue:3, pp.1614-1622) Aug. 2009.

[7] Yuan-Hsin Liao, Gwo-Long Li, Tian-Sheuan Chang, "A Highly Efficient VLSI Architecture for H.264/AVC Level 5.1 CABAC Decoder", IEEE Transactions on Circuits and Systems for Video Technology, (Volume:22 , Issue:2, pp.272-281), Feb. 2012.

[8] Yongseok Yi, In-Cheol Park, "High-Speed H.264/AVC CABAC Decoding", IEEE Transactions on Circuits and Systems for Video Technology, (Volume:22 , Issue:2, pp.490-494), Apr. 2007.

[9] Zhang, P, Xie, D., Gao, W., "Variable-bin-rate CABAC engine for H.264/AVC high definition real-time decoding." IEEE Transactions on Very Large Scale Integration (VLSI) Systems, (Volume:17 , Issue:3, pp.417-426), Mar. 2009

[10] P. Zhang, "Fast CABAC decoding architecture," Electronics Letter, (Volume:44, Issue:24, pp.1394 - 1395), Nov. 2008.

[11] HS. Kim, KK. Ryoo, "The Hardware Design of a High throughput CABAC Decoder for HEVC", The Korean Institute of Information and Commucation Engineering, (Volume:17 , Issue:3, pp.385-390), Feb. 2013.

 저 자 소 개



배 봉 희(정회원)
 2012년 한림대학교 전자공학과
 학사 졸업.
 2014년 광운대학교 컴퓨터공학과
 석사 졸업.
 <주관심분야 : 영상신호처리,
 VLSI, Embedded System >



공 진 흥(평생회원)
 1980년 서울대학교 전자공학과
 학사 졸업..
 1982년 한국과학기술원 전기 및
 전자공학과 석사 졸업.
 1989년 텍사스주립대학교
 컴퓨터공학과 박사 졸업.
 <주관심분야 : 영상신호처리 SoC설계, VLSI 설
 계, 임베디드시스템 설계>