

H.264/AVC Main Profile 의 실시간 복호화를 위한 CABAC 복호기의 설계

Design of a CABAC Decoder for Real-Time Decoding of H.264/AVC Main Profile

이용석, 박인철

한국과학기술원 전자전산학과 전기 및 전자공학 전공
yslee@ics.kaist.ac.kr, icpark@eekaist.ac.kr

Abstract – H.264/AVC (MPEG-4 Part 10) is an emerging video compression standard to be used in digital broadcasting and storage system. The main advantage of H.264/AVC is high compression ratio achieved by exploiting Context-Based Adaptive Binary Arithmetic Coding (CABAC). However, since the decoding procedure of CABAC is very sequential in nature, it is difficult to exploit parallelism in algorithmic level. To meet the real-time constraints with a given operating frequency, the number of cycles to decode a bin should be minimized. This paper presents efficient hardware architecture to decode CABAC codewords with the performance targeted at 1xHD real-time decoding.

Keywords: video compression, BAC, CABAC, H.264, MPEG-4

1 서문

H.264/AVC (MPEG-4 Part 10) [1]는 MPEG-2 비디오를 대체할 수 있는 영상 압축 표준이다. H.264/AVC 의 가장 큰 장점은 MPEG-2 와 비교하여 비슷한 화질을 얻으면서도 훨씬 향상된 압축률을 보인다는 것이다. 이러한 압축률은 주로 CABAC (Context-Based Adaptive Binary Arithmetic Coding)이라는 부호화 방식에 기인한다.[2]

CABAC 은 기존의 AC (Arithmetic Coding) [3]에 더하여 MPS (Most Probable Symbol)이 나타날 확률을 컨텍스트에 따라 변화하도록 한 것이며 그림 1-(a)와 그림 1-(b)는 각각 CABAC 의 부호화 및 복호화 순서를 나타내고 있다.

H.264/AVC 로 부호화된 영상을 복원하는 입장에서의 문제점은 소프트웨어 방식으로는 CABAC 의 복호화가 너무 높은 처리 속도를 요구한다는 것이다. BAC 의 특성 상, 한 개의 bin (binarization 과정을 거쳤을 때 심벌에 대응되는 bin string 을 이루는 하나의 비트)에

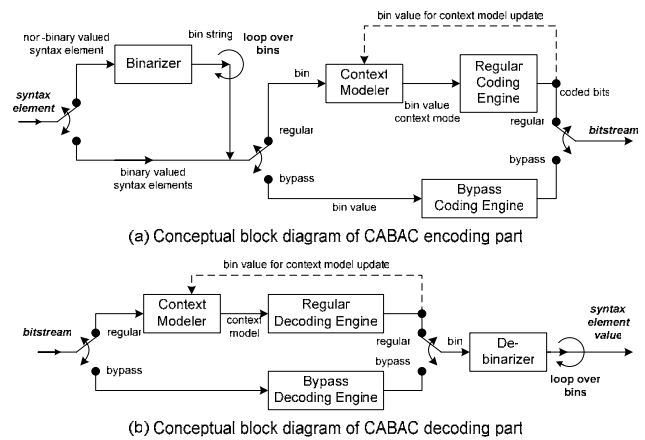


그림 1 H.264/AVC 의 CABAC 부호화 및 복호화 블록도

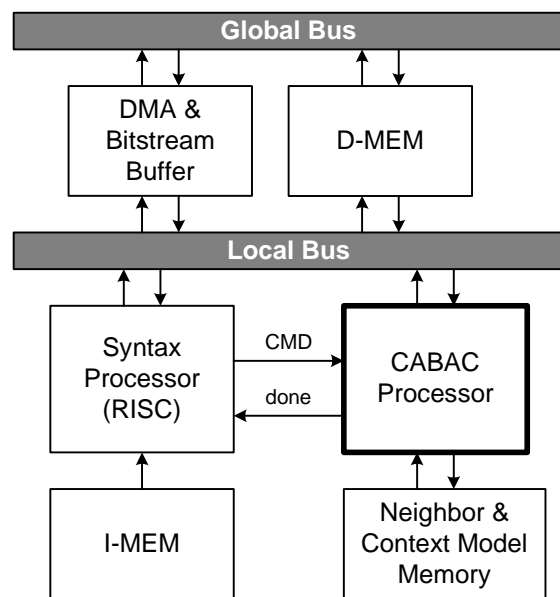


그림 2 전체 시스템의 개요

대한 복호화가 완료되어야만 다음 bin 에 대한 복호화가 이루어질 수 있으며, 이는 CABAC 복호화의 알고리즘 수준에서 어떠한 방식으로든지 병렬처리를 어렵게 하는 요인이 된다.

본 논문에서는 그림 2 와 같은 시스템에서 신택스 프로세서를 보조하는 코프로세서로 동작하며 150MHz 의 동작 주파수에서 1xHD 영상을 실시간으로 복호화 할 수 있는 CABAC 복호기를 제안한다.

2 CABAC 의 분석

한 개의 bin 을 복호화할 때에는 그림 3 과 같은 순서를 따라야 한다. CS (Context Selection, 컨텍스트 선택) 단계에서는 현재의 SE 종류와 bin 의 인덱스에 기반하여 ctxIdx (컨텍스트 인덱스)를 계산한다. ctxIdx 는 0 에서 398 까지의 값을 가질 수 있으며 CL (Context Loading) 단계에서 399 개의 pStateIdx (확률 상태 인덱스) 및 valMPS (MPS 의 값) 쌍 중에 한 쌍을 고르기 위해 사용된다. pStateIdx 와 valMPS 의 쌍을 통상적으로 컨텍스트 모델이라 칭한다. 이렇게 얻어진 컨텍스트 모델과 오프셋 및 범위값이 BAC 의 입력으로 들어간다. BAC 단계에서는 한 비트의 심벌, 즉 0 또는 1 이 출력으로 나오며, 이를 받아들인 BM (Binarization matching) 단계에서 현재까지 모아진 bin string 이 유효한지의 여부를 판단하여 유효한 bin string 인 경우에는 현재 SE 에 대한 복호화를 종료하고 그렇지 않을 경우에는 CS 단계로 돌아가 다음 bin 의 복호화를 시작한다.

한 번 bin 이 복호화 될 때마다 해당 컨텍스트 인덱스에 대한 컨텍스트 모델의 값이 변화할 수가 있으며 따라서 BAC 단계가 완료된 후에는 이 새로운 값으로 컨텍스트 모델 표를 갱신하는 단계 (CU)가 뒤따라야 한다.

여기에서 CS, BAC, 그리고 BM 은 논리 연산에 의해 이루어지며, CL 과 CU 는 메모리 접근에 의해 이루어진다.

3 복호화 방식

CABAC 복호화의 성능을 높이기 위해서 요구되는 것은 첫 번째 bin 을 복호화하는데 걸리는 사이클 수를 최소화하는 것과 전체적으로 한 bin 을 복호화하는데 걸리는 평균 사이클 수를 최소화하는 것이다. 이 절에서는 전자와 후자를 분리하여 기술하도록 한다.

3.1 첫 Bin 의 복호화의 최적화

내부 메모리를 사용하여 메모리의 읽기 및 쓰기가 1 사이클에 이루어질 수 있다고 가정할 때, 2 절에서

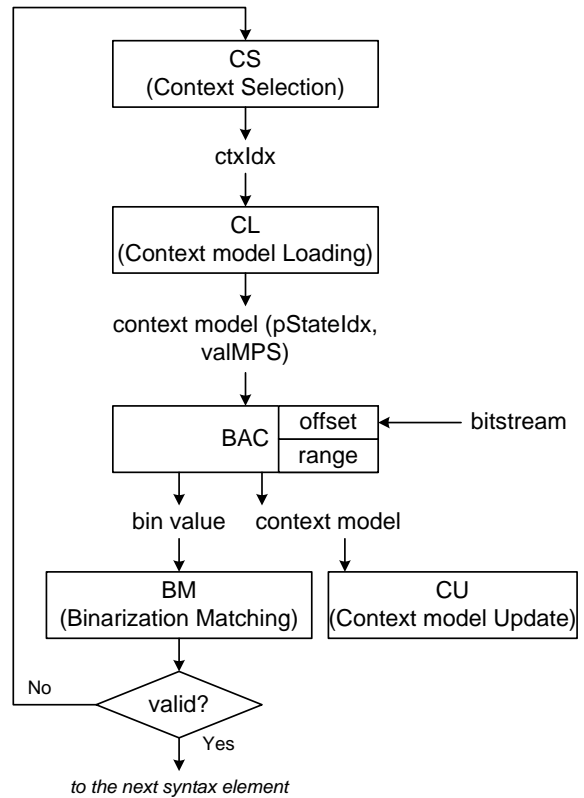


그림 3 Bin 한 개에 대한 CABAC 복호화의 순서도

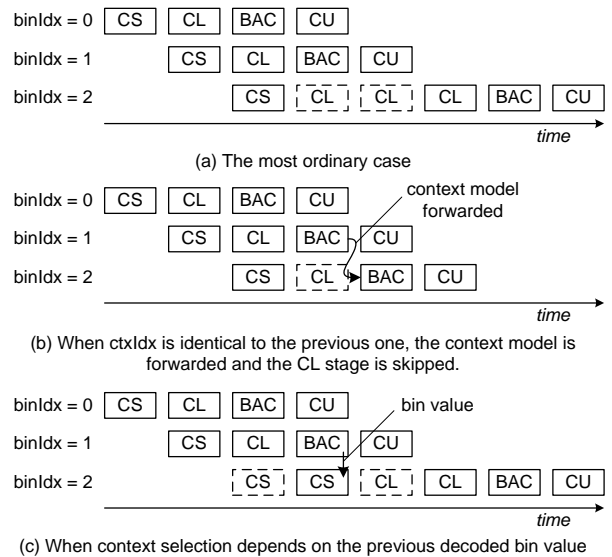


그림 4 CABAC 복호화의 파이프라이닝

나열한 각 단계들은 모두 1 사이클에 이루어질 수 있다. 소프트웨어로 구현할 경우 가장 걸림돌이 되는 부분은 CS 단계인데 이는 SE 종류에 따라 상당히 많은 조건들을 동시에, 또는 순차적으로 검사하며 그 조건들을 생성하기 위해서는 인접한 매크로 블록들의 정보를 메모리로부터 읽어와야 하기 때문이다. 하드웨어의

경우 다중 조건에 따른 판단은 모두 논리 회로를 사용하여 동시에 수행할 수 있으므로 문제가 되지 않으나, 인접 매크로 블록의 정보를 읽어오는 것은 여전히 수행속도를 느리게 하는 원인으로 남는다. 따라서 본 설계에서는 한 개의 매크로 블록을 복호화하기 시작할 때, 인접 매크로 블록의 정보를 한꺼번에 읽어서 레지스터에 저장해놓는 방식을 사용함으로써 CS 를 수행할 때마다 메모리 접근을 하지 않아도 되게 하였다.

또한 BAC 과 BM 은 그림 3 에서와 달리 동시에 이루어질 수가 있는데 그 이유는 BM 이 현재의 상태에서 받아들일 수 있는 입력은 0 또는 1 밖에 없고 그 중에 어떤 것을 받아들였을 때 유효한 bin string 이 이루어지는지를 이미 알 수 있기 때문이다.

3.2 평균 복호화 시간의 최소화

한 SE 가 여러 개의 bin 으로 이루어져 있을 경우, 각각의 bin 에 대한 복호화를 파이프라인으로 구현할 수 있다. 파이프라인이 가능한 것은 대부분의 경우 현재 bin 의 컨텍스트 인덱스와 다음 bin 의 컨텍스트 인덱스가 서로 독립적이기 때문이다. 그림 4-(a)는 가장 일반적인 경우의 파이프라인을 묘사하고 있다. 이전 bin 의 컨텍스트 인덱스가 현재 bin 의 컨텍스트 인덱스와 동일할 경우에는 이전 BAC 단에서 나온 새로운 컨텍스트 모델이 현재 BAC 단의 입력으로 포워딩 된다 (그림 4-(b)). 소수의 경우 현재의 컨텍스트 인덱스가 이전에 복호화된 bin 값에 의존할 때가 있는데 이러한 경우의 파이프라인은 그림 4-(c)와 같이 이루어진다.

4 복호기의 구조

CABAC 복호기는 서문에서 제시한 바와 같이 슬라이스 수준 이하의 모든 SE 들을 복호화한다. 이 과정은 하나의 FSM (유한상태기)에 의하여 수행되며 이 TOP_FSM 은 다시 단일 SE 의 복호화를 제어하는 하위 FSM (CABAC_FSM)을 제어한다. CABAC_FSM 은 그림 5 와 같은 데이터패스를 제어한다.

데이터패스를 이루는 블록들은 크게 2 절에서 나열한 복호화의 단계들을 담당하는 기능 블록들 (CTX_UNIT, BAC_UNIT, BM_UNIT)과 이들에게 적당한 정보를 제공하는 블록들로 구분할 수 있다. 이 중에서 NRB (Neighbor Register Bank)는 인접 매크로 블록 및 현재 매크로 블록의 정보를 저장하는 용도로 사용되며 이 레지스터 집합의 각 부분들은 CS 를 수행하는 CS_UNIT 의 입력으로 들어간다.

상태 레지스터 (status register)는 복호화에 필요한 여러 가지 부가 정보 (화면 크기, CABAC 초기화 인수 등)를

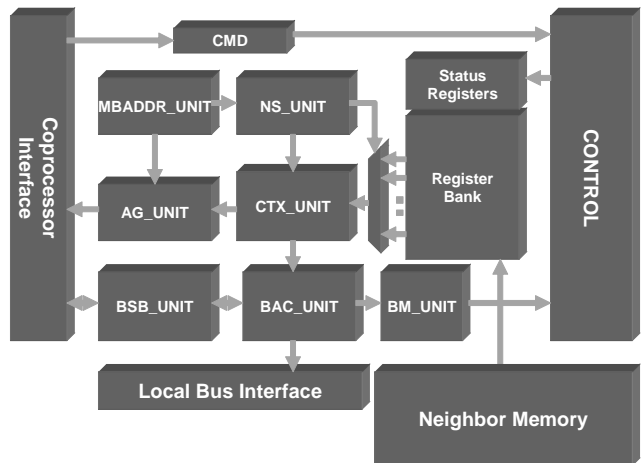


그림 5 CABAC 복호기의 데이터패스 구조

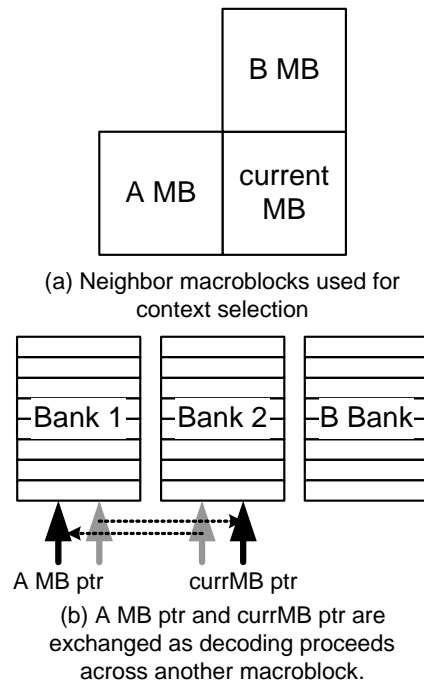


그림 6 인접 매크로블록 및 해당 레지스터 집합

담아두고 있는데 이들은 각 기능 블록에 직접 연결된다. MBADDR_UNIT 은 현재 매크로 블록의 주소를 계산하며 여기서 계산된 주소를 바탕으로 NS_UNIT 은 인접한 매크로 블록 및 서브 블록의 주소를 계산한다.

BAC_UNIT 에서 bin 복호화를 수행할 때 코드 오프셋 값을 갱신하기 위하여 필요한 비트스트림은 비트스트림 버퍼 (BSB_UNIT)를 통하여 공급 받는다. BSB_UNIT 은 내부적으로 32 비트 크기의 버퍼를 가지고 있으며 버퍼가 빌 때마다 버스 인터페이스를 통해 새로운 비트스트림을 읽어오는데 이 경우에는 복호화 과정이 멈추게 된다.

컨텍스트 선택을 할 때 참조하는 인접 매크로 블록은 대부분의 경우 그림 6-(a)와 같이 현재 매크로 블록의 좌측 (A)과 상단 (B)의 매크로 블록이 된다. 레지스터 집합에는 이 인접 매크로 블록들의 정보가 저장되며 다음 매크로 블록으로 복호화가 진행할 때에는 그림 6-(b)와 같이 A 블록을 가리키는 인덱스가 바로 이전의 현재 매크로 블록에 해당하는 레지스터 집합을 가리키게 된다. 또한 B 블록에 해당하는 레지스터 집합에는 메모리 (Neighbor Memory)에서 읽어온 정보를 저장한다.

5 성능 분석

표 1은 H.264/AVC Main Profile로 압축된 HD (1920x1088) 영상에 대한 통계자료이다.

표 1 HD 영상에 대한 통계

| | |
|-------------------------------------|------------------------------|
| Profile/Level | Main Profile, Level 5.0 |
| Frame Sequence | 1920x1088 25 Hz, 25 frames |
| Bitrate | 37.64 Mbps \approx 40 Mbps |
| Bits used for CABAC codeword | 36 Mb (96 %) |
| # syntax elements | 40,000,000 |
| # bins | 47,000,000 |
| Average # of bins per SE | 1.2 |
| Skipped Macroblocks | 116,188 out of 204,000 (57%) |
| Average # of SE for non-skipped MBs | 300 |

표 1에서 보듯이 SE 한 개당 복호화되어 나오는 bin의 개수가 1.2개에 지나지 않기 때문에 성능을 향상하기 위하여 가장 중요한 것은 역시 첫 번째 bin에 대한 복호화 시간을 줄이는 것임을 알 수 있다.

앞서 제안한 파이프라인에 의한 복호화에서는 n 개의 bin을 복호화할 때 평균적으로 $3 + n$ 사이클이 소요된다. 그러므로 SE 한 개당 평균 4.2 사이클이 소요됨을 알 수 있다. 여기에 추가적으로 매크로 블록 한 개를 시작할 때마다 인접 매크로 블록의 정보를 읽어오고 이전 매크로 블록의 정보를 메모리에 저장하는 시간이 25 사이클 소요된다. 따라서 매크로 블록 한 개당 평균 $25 + 4.2 \times 300 = 1285$ 사이클이 소요되며, 표 1에 근거하여 전체 매크로 블록 중 약 절반이 스킵된다고 가정했을 경우 1초 분량의 프레임 시퀀스를 복호화 하기 위해서는 총 13,107,000 사이클이 소요된다. 이는 150 MHz의 주파수로 동작시켰을 때 충분히 실시간으로 복호화를 수행할 수 있는 연산량이다.

6 결론

본 논문에서는 150 MHz의 동작 주파수로 H.264/AVC로 압축된 1xHD 영상을 실시간으로 복호화할 수 있는

CABAC 복호기를 제안했다. 실시간 복호화가 가능하도록 하기 위하여 (1) CABAC 복호화에 필요한 각 단계를 1 사이클에 수행되도록 하였고 (2) BAC와 BM을 동시에 수행하도록 하여 bin 당 소요 사이클 수를 줄였으며, (3) 연속되는 bin을 복호화할 때 파이프라인을 도입하여 bin 당 평균 처리 시간을 최소화하였다.

이러한 실시간 복호기는 고성능을 요구하는 차세대 디지털 텔레비전 등에 효과적으로 응용될 수 있다.

Acknowledgment

본 연구는 MICROS 센터를 통하여 한국과학재단, SystemIC 2010 과제와 반도체설계교육센터 (IDEC)을 통하여 정보통신부와 산업자원부에 의해 지원을 받아 이루어졌습니다.

참고 문헌

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC), 2002.
- [2] D. Marpe et al., "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.13, No.7, July 2003.
- [3] G. G. Langdon Jr., "An Introduction to Arithmetic Coding," IBM Journal of Research and Development, Vol. 28, No.2, March 1984.