논문 2007-44SD-7-7

실시간 HD급 영상 처리를 위한 H.264/AVC CAVLC 부호화기의 하드웨어 구조 설계

(VLSI Design of H.264/AVC CAVLC encoder for HDTV Application)

우 정 욱*, 이 원 재*, 김 재 석**

(Junguk Woo, Wonjae Lee, and Jaeseok Kim)

요 약

본 논문에서는 실시간 HD급 영상(1920x1080@30fps) 처리를 위한 효율적인 CAVLC (Context-based Adaptive Variable Length Code) 부호화기의 하드웨어 구조를 제안한다. 기존에 제안되었던 CAVLC 하드웨어 구조들은 CAVLC 부호화를 위해 필요한 4x4 블록내의 정보들을 구하기 위해서 16개의 계수들을 모두 탐색하면서 zigzag scanning을 하였다. 그러나 zigzag 방향으로 정렬 된 계수들 중 '0'이 아닌 마지막 계수 이후에 존재하는 '0'의 열은 CAVLC 부호화를 하는데 있어 불필요한 계수들이다. 본 논문에서는 이러한 불필요한 연산을 줄이기 위해서 계수 위치 탐색 기법과 레벨 순차 정렬 기법을 제안한다. 제안된 구조를 적용하여 실험한 결과, 하나의 매크로블록을 처리하는 평균 클럭 수(Cycles/MB)는 기존 방식보다 약 23%가 줄었다. 제안된 CAVLC 하드웨어 구조는 Verilog HDL을 사용하여 하드웨어로 설계 및 검증되었다. 0.18um 표준 셀 라이브러리로 합성한 결과 16.3k 게이트를 가졌고, HD급(1920X1080@30fps) 영상을 기준으로 했을 경우 81MHz에서 동작할 수 있음을 확인하였다.

Abstract

In this paper, we propose an efficient hardware architecture for H.264/AVC CAVLC (Context-based Adaptive Variable Length Coding) encoding. Previous CAVLC architectures search all of the coefficients to find statistic characteristics in a block. However, it is unnecessary information that zero coefficients following the last position of a non-zero coefficient when CAVLC encodes residual coefficients. In order to reduce this unnecessary operation, we propose two techniques ,which detect the first and last position of non-zero coefficients and arrange non-zero coefficients sequentially. By adopting these two techniques, the required processing time was reduced about 23% compared with previous architecture. It was designed in a hardware description language and total logic gate count is 16.3k using 0.18um standard cell library. Simulation results show that our design is capable of real-time processing for 1920x1088 30fps videos at 81MHz.

Keywords: H.264/AVC, Entropy coding, CAVLC(Context-based Adaptive Variable Length Code)

I. 서 론

H.264/AVC는 ITU-T 와 ISO/IEC가 함께 표준화 작업을 진행시켜 얻어낸 새로운 비디오 압축 표준 방식으로 기존의 동영상 압축 방식과는 달리 높은 압축 효율과 다양한 전송 환경에서도 에러에 강한 특성을 지닌 최신의 영상 압축 표준화 방식이다[1]. H.264/AVC 영상 압축 표준 방식이 위와 같은 높은 부호화 효율을 갖는 것은 다양한 크기의 움직임 보상 및 1/4 화소 단위의 움직임 추정, 인트라 추정, 정수 기반 DCT 변환, 디블럭킹 필터와 엔트로피 부호화 같은 기술을 적용했기 때문이다. 기존의 H.263이나 MPEG-1/2/4 와는 달리 이러한 새로운 기술을 채택함으로써 압축 효율은 2배 이

* 학생회원, ** 정회원, 연세대학교 전기전자공학과
(Dept. Electrical and Electronic Eng. Yonsei Univ.)
※ 본 연구는 정보 통신부 및 정보통신연구진흥원 대학 IT연구센터 육성·지원사업 및 2006년도 교육인적자원부 BK21 사업의 일환인 연세대학교 전기전자공학부 TMS 사업단의 지원을 받아 연구되었고, CAD Tool은 IDEC으로부터 지원 받았음
접수일자: 2007년1월11일, 수정완료일: 2007년6월27일

상 향상되었지만, 처리해야 할 연산량은 증가했고 복잡도는 최대 16배 이상 증가했다. 따라서 PVR(Personal Video Recoder)과 같이 실시간에서 HD급을 처리할 수 있어야 하는 시스템을 설계하기 위해서 하드웨어 기반의 구조 설계가 요구되고 있다[2].

본 논문에서는 엔트로피 부호화 방식 중 양자화 된 변환계수를 부호화하는 CAVLC 부호화기의 하드웨어 구조를 제안한다. H.264/AVC의 CAVLC는 MPEG-4와 같은 기존의 영상시스템과는 달리 인접 심볼들 간의 상관관계 특성을 이용한 부호화 방식을 채택하여 부호화 효율을 향상시켰다. 하지만 이로 인해 하나의 4x4 블록의 zigzag scanning이 끝나고 나서야 블록내의 심볼들의 통계학적인 특성들을 알 수 있기 때문에 각 블록은 순차적으로 처리되어야 하는 문제점이 발생했다. 또한 5단계 CAVLC 부호화 과정으로 인해 CAVLC 부호화 수행시간은 증가하였다. 이와 같은 문제점을 해결하여 하드웨어 성능을 향상시키기 위한 다양한 CAVLC 하드웨어 구조가 기존에 제안되었다[7-8]. 하지만 기존에 제안되었던 방식은 공통적으로 CAVLC 5단계 부호화를 하기 위해서 필요한 정보 값들을 수집하기 위해 블록내의 모든 계수들을 탐색하는 zigzag scanning 단계에 의해 부호화 처리 속도가 제한되는 문제점이 있다.

이와 같은 문제점을 해결하여 HD급 영상을 실시간으로 처리할 수 있는 CAVLC 부호화기를 설계하기 위해서, 본 논문에서는 계수 위치 탐색 기법과 레벨 순차 정렬 기법을 제안하여 블록내의 계수들 중 부호화를 하는데 있어 불필요한 계수들을 탐색하는 시간을 줄임으로써 전체 수행 시간을 줄이는 방법을 제시한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 CAVLC의 개요 및 기존 하드웨어 구조를 소개하고 새로운 하드웨어 구조의 필요성을 언급한다. Ⅲ장에서는 불필요한 계수탐색 시간을 줄여, 부호화 수행시간을 줄일 수 있는 새로운 기법을 제시한다. Ⅳ장에서는 기존에 제안된 구조와 비교 분석을 하고, 마지막으로 Ⅴ장에서는 결론을 맺는다.

Ⅱ. CAVLC

2.1. CAVLC 개요

H.264/AVC에서 사용하는 엔트로피 부호화 방식은 크게 가변 길이 부호화(Variable Length Coding)방식과 이진 산술 부호화(Binary Arithmetic Coding)방식으로 나누어진다. 가변 길이 부호화 방식에는 양자화 된 변환계수들을 제외한 모든 구문 요소(syntax elements)

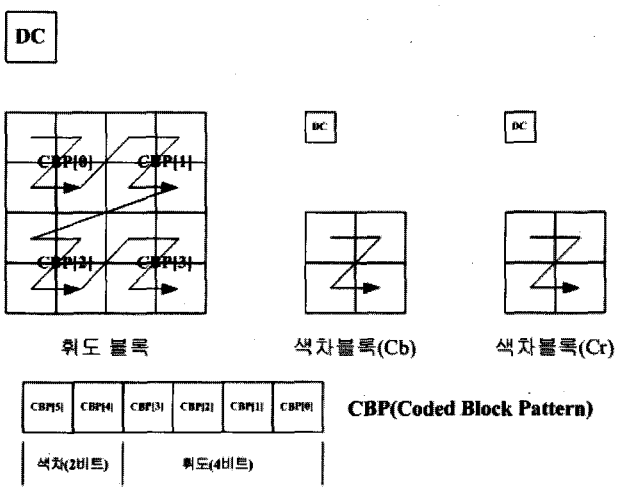


그림 1. CAVLC 부호화 순서
Fig. 1. CAVLC encoding order.

들을 부호화하기 위한 Exp-Golomb 코드 부호화 방식과 높은 부호화 효율이 요구되는 양자화 된 변환 계수들을 부호화하기 위해서 사용되는 CAVLC 부호화 방식이 있다[3].

CAVLC는 프레임 내에 매크로블록 단위로 처리하며, 매크로블록 내에서는 매크로블록 유형과 6비트 CBP(Coded Block Pattern) 정보에 따라서 부호화 순서가 달라진다. CAVLC는 그림 1과 같은 방향으로 부호화 되며, 만약 매크로 블록의 유형이 Intra 16x16 모드이면 먼저4x4 DC 계수들을 부호화 하고, 나머지 16개의 4x4 AC 계수들을 부호화 한다. Intra 16x16 모드가 아닐 경우에는 16개의 4x4 블록을 부호화 한다. 이때각 블록에 해당하는 CBP 값이 0이 아닌 블록에 대해서만 선택적으로 부호화 한다. 휘도 블록에 대한 부호화가 끝나면, 상위 2비트 CBP 값에 따라서 8x8 색차 Cb, Cr 블록의 부호화 순서가 결정된다. 만약 상위 2비트가 '10' 값을 가지게 되면 Cb, Cr 블록의 2x2 DC 계수들을 먼저 부호화 하고 이후에 나머지 AC 계수들을 부호화 한다. '01'의 값을 가지면 2x2 DC 계수들만 부호화를 하고, 나머지 AC 계수들을 부호화 하지 않는다. 나머지 경우에는 색차 블록을 부호화 하지 않는다. CAVLC는 이처럼 CBP 값에 따라 매크로블록내의 블록들을 선택적으로 부호화함으로써 압축 성능을 더욱 더 향상시켰다.

CAVLC는 다음과 같은 특성의 장점을 살려 zigzag 방향으로 스캔 되어진 변환 계수를 부호화한다[4].

1. zigzag scanning 이후에 많은 '0'의 열이 존재한다.
2. zigzag scanning 이후에 가장 많이 존재하는 '0'이 아닌 계수 값은 ±1이다.

3. 인접한 블록내의 계수의 총 수는 상관관계가 있다. 따라서 블록내의 계수에 대해서 부호화 할 때는 인접한 블록의 계수의 수에 영향을 받는다.
4. zigzag 방향으로 재 정렬된 계수들은 주로 DC 성분으로 갈수록 값이 크고, AC 성분으로 갈수록 값이 작아진다. 그러므로 각각의 계수에 대한 크기를 부호화 할 때 각 계수들은 이전에 부호화 된 계수 값들의 영향을 받는다.

CAVLC 알고리즘은 4x4 블록 내의 양자화 된 변환 계수 값들을 zigzag 방향으로 역순으로 부호화 한다. 이는 DCT 변환 이후에 저주파수 대역으로 갈수록 계수 값들이 몰리고, 고주파수 대역으로 갈수록 0에 가까운 계수 값이 많아지기 때문에 실제 통계적 특성에 적응해 나가기 위해서이다. zigzag 방향으로 계수를 탐색하여 부호화를 위한 정보 값들을 수집하면 다음의 5가지 순서로 부호화 한다[5].

1. 블록 내의 '0' 이 아닌 계수의 총 수(Total Coefficients)와 계수 ±1(Trailing ones)의 개수를 하나의 부호 길이로 부호화한다. 이때 위쪽과 왼쪽 블록내의 Total Coefficients의 수의 평균값에 따라 룩업 테이블이 선택된다.
2. 계수 ±1의 부호(Trailing signs)를 부호화 한다. +1 이면 0으로, -1이면 1로 부호화 한다.
3. Trailing ones를 제외한 '0'이 아닌 계수 값들(Levels)을 부호화 한다. 이때 각 계수들은 바로 이전에 부호화 된 계수 값들의 영향을 받는다.
4. Zigzag 방향으로 정렬 된 계수 중에서 '0'이 아닌 마지막 계수 이전에 존재하는 모든 '0'의 수(Total Zeros)를 부호화 한다.
5. 각각의 계수 사이에 존재하는 '0'의 열을 부호화 한다.

2.2 기존 CAVLC 하드웨어 구조

H.264/AVC의 CAVLC 부호화방식은 부호화 되어야 할 데이터의 통계학적인 특성들을 구하기 위해서 4x4 블록내의 양자화 된 변환 계수들을 탐색하는 zigzag scanning을 한다. 그런 다음 인접 블록간의 상관관계 특성을 이용하여 CAVLC 5단계 부호화를 수행한다. 이와 같은 부호화 방식을 채택하여 기존의 영상시스템보다 압축효율은 향상되었지만 처리해야 할 연산량이 증가하여 CAVLC 총 부호화 수행시간과 하드웨어 복잡도가 증가하였다. 이러한 문제점을 해결하기 위해서 Lei[7]는 하드웨어 복잡도를 줄이기 위한 직렬 부호화 구조를 제안하였다. 하지만 이 구조는 블록내의 심볼들의 특성을 구하기 위해서 블록내의 모든 계수들을 탐색하고, CAVLC 5단계 부호화를 직렬 방식으로 처리함으로써 부호화 수행 시간이 증가하게 되는 단점이 있다. 따라서 이 구조는 HD급 영상을 실시간으로 처리하기에는 부적합한 구조이다. Chen[8]은 Dual-buffer를 가지고 두 개의 4x4 블록을 파이프라인 방식으로 처리하기 위한 2단계 파이프라인 기법을 제안하였다. 이 구조는 듀얼 버퍼를 사용하여 하나의 4x4 블록이 첫 번째 버퍼에 저장된 심볼 값들을 이용하여 부호화를 진행하는 동안에 다음 4x4블록은 두 번째 버퍼를 이용하여 zigzag scanning을 하는 구조이다. 파이프라인 기법을 적용하여 처리 성능은 향상시켰지만, 이 구조는 추가적인 버퍼를 필요로 하고 zigzag scanning 단계와 부호화 단계에서 요구되는 최대 클록 수에 의해 하드웨어 성능이 저하되는 문제점이 있다.

2.3 새로운 CAVLC 하드웨어 구조의 필요성

기존에 제안된 구조들은 초기 정보들을 얻기 위해 블록내의 모든 계수들을 탐색하는 zigzag scanning 단계에 의해 하드웨어 성능이 저하된다는 문제점이 있다.

그림 2에 나타난 것처럼 블록내의 정보를 구하기 위한 zigzag scanning 단계에서 zigzag 방향으로 정렬된 계수들 중 '0'이 아닌 마지막 계수 이후에 존재하는 '0'의 열은 CAVLC 부호화를 하는데 있어 불필요한 계수들이다.

그림 3은 CAVLC 부호화를 함에 있어 하나의 프레임 내에 평균적으로 불필요한 계수탐색 비중이 어느 정도 차지하는지 확인하기 위해서 QP (Quantization Parameter)값의 변화에 따른 결과 값들을 그래프로 나타낸 것이다. 영상의 특성에 따라서 약간의 차이는 있으나 그래프에 나타나 있듯이 프레임 내에 불필요한 계

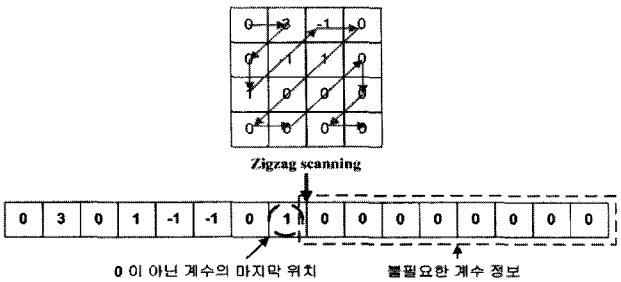
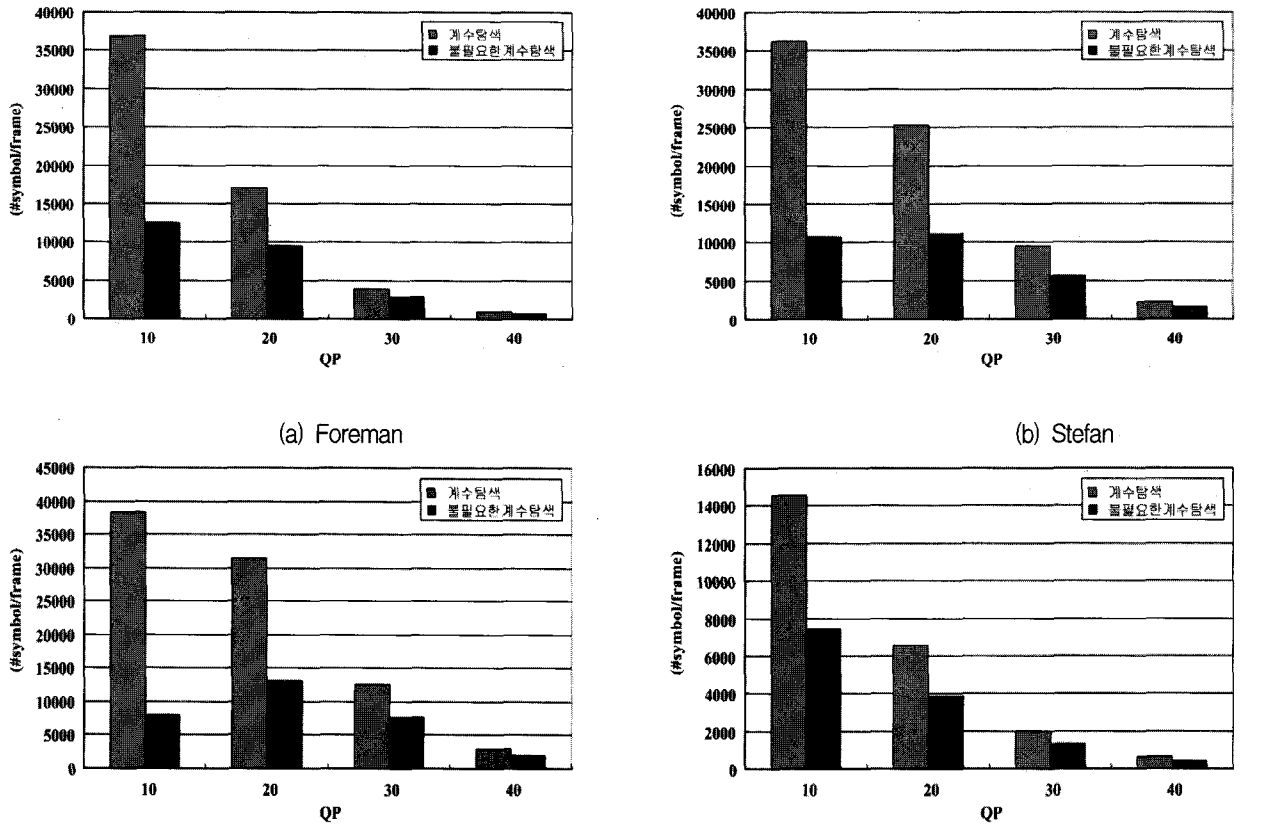


그림 2. zigzag 방향으로 정렬 된 계수 중 불필요한 계수 정보들

Fig. 2. unnecessary coefficients in zigzag reordered coefficients.



(a) Foreman (b) Stefan

(c) Mobile Calendar (d) Weather

그림 3. 프레임 당 불필요한 계수 탐색 비중

Fig 3. The counts of unnecessary coefficients per frame for QCIF sequences.

수탐색 비중이 큰 부분을 차지하고 있음을 확인 할 수 있다. CAVLC 부호화를 하기 위해 초기 정보 값들을 획득하는 과정에서 이러한 불필요한 계수 탐색으로 인해서 부호화 시간이 증가하여 실시간 HD 영상 처리를 어렵게 하고, 불필요한 파워가 소모된다. 따라서 이러한 불필요한 연산을 줄여 HD급 영상을 실시간으로 처리할 수 있는 새로운 CAVLC 하드웨어 구조가 요구된다.

III. 제안된 CAVLC 부호화기 구조

그림 4는 본 논문에서 제안하는 CAVLC 부호화기의 구조를 보여준다. 전체 부호화기는 그림 4에 나타난 바와 같이 크게 초기화 단계와 부호화 단계로 나뉜다.

초기화 단계는 계수들을 zigzag 방향으로 정렬하는 Zigzag Scan 블록, 인접한 블록간의 상관관계 특성을 이용하기 위한 Gen_nA_nB, Gen_nC 블록 그리고 CAVLC 5단계 부호화를 하기 위해 필요한 초기 정보 값들을 고속으로 획득하는 Information Initialization 블록들로 구성된다.

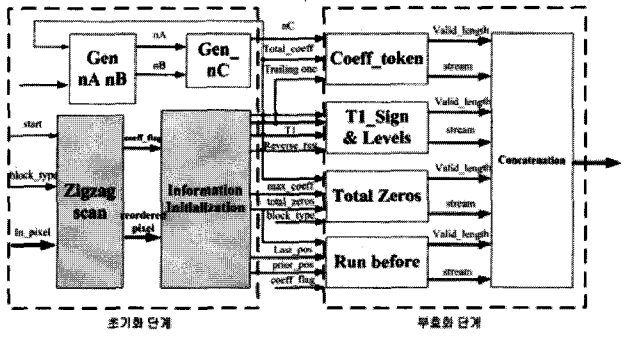


그림 4. 제안하는 CAVLC 부호화기 구조

Fig. 4. The architecture of the proposed CAVLC Encoder.

부호화 단계는 5단계 부호화를 진행하는 각각의 부호화 블록(Coeff_token, T1_Sign & Levels, Total Zeros, Run_before)과 병렬로 처리된 각각의 부호화 블록의 출력 스트림을 CAVLC 5단계 순서에 맞게 조합하는 Concatenation블록으로 구성된다.

본 논문에서는 초기화 단계에서 블록내의 불필요한 계수탐색 시간을 줄임으로써 부호화를 하기 위한 초기 정보 값들을 고속으로 획득하여 CAVLC 부호화 총 수

행시간을 줄일 수 있는 방법을 제시한다.

3.1 초기화 단계 블록

(가) Zigzag scan 블록

그림 4의 Zigzag scan 블록은 기존의 방식처럼 블록내의 정보를 수집하기 위해 블록내의 모든 계수들을 zigzag 방향으로 탐색하는 zigzag scanning을 하지 않고, 블록내의 계수들을 한 클록에 zigzag 방향으로 정렬 한다. 그리고 zigzag 방향으로 정렬된 계수들에 대해서 각 위치에 대응하는 값이 '0'이 아닌 값을 가지면 1, '0'의 값을 가지면 0을 나타내는 16비트 Coeff_status 신호를 생성한다[6]. 그림 5는 zigzag 방향으로 정렬된 계수들로부터 Coeff_status 신호를 생성하는 과정을 보여준다.

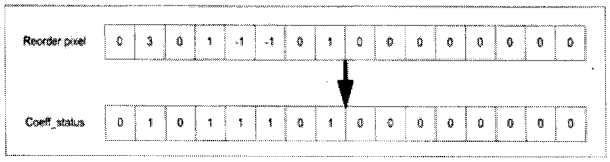


그림 5. Coeff_status 신호 생성 과정
Fig. 5. The generating process of Coeff_status.

(나) Information Initialization 블록

그림 6은 제안하는 CAVLC 부호화기의 Information Initialization 블록의 세부적인 구조를 보여준다. 이 블록은 CAVLC 5단계 부호화를 하기 위해서 필요한 초기 정보 값들(Total Coefficients, Trailing ones, Total Zeros, Run_before, Levels)을 고속으로 획득하는 블록으로 TotalCoeff_Calu, Totalzero_Calu, Coefficient Detector, Level Arrangement의 4개의 기능 블록으로 구성되어 있다.

TotalCoeff_Calu 블록은 그림 4의 Zigzag scan 블록에서 계산된 Coeff_status 값을 입력받아 '0'이 아닌 계수의 총 수(Total Coefficients)를 한 클록에 구한다.

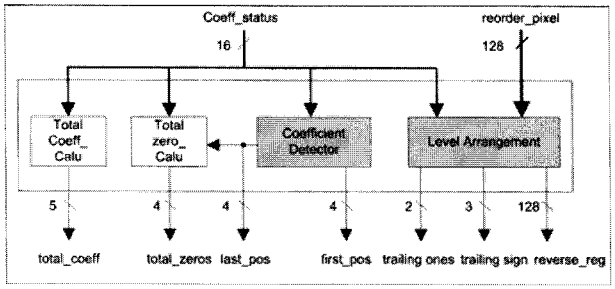


그림 6. Information Initialization 블록
Fig. 6. Information Initialization block of proposed structure.

Totalzero_Calu 블록은 Coefficient Detector 블록에서 구한 zigzag 방향으로 정렬 된 계수 중에서 '0' 이 아닌 계수의 마지막 위치 값을 입력받아 마지막 계수 이전에 존재하는 '0'의 총 수(Total zeros)를 한 클록에 구한다.

Coefficient Detector 블록은 Coeff_status 신호를 가지고 계수 위치 탐색을 사용하여 zigzag 방향으로 정렬 된 계수들 중 '0'이 아닌 계수의 처음 위치와 마지막 위치를 구한다.

Level Arrangement 블록은 Trailing signs와 레벨 값들을 효율적으로 부호화 하기위해서 레벨 순차 정렬 기법을 사용하여 zigzag 방향으로 정렬 된 계수들 중 '0'이 아닌 계수 값들을 레지스터에 순차적으로 저장하는 블록이다. 이 기법을 적용하여 Trailing signs를 한 클록에 부호화하고, Trailing ones를 제외한 레벨 값들을 순차적으로 부호화 한다.

제안된 Information Initialization 블록은 부호화를 위한 초기 정보 값들을 얻기 위해서 한 클록에 하나의 계수를 탐색하여 초기 정보 값들을 레지스터에 저장하는 기존의 방식을 사용하지 않고, 3.2장에서 설명하는 계수 위치 탐색 기법과 레벨 순차 정렬 기법을 사용하여 부호화 수행 시간을 줄인다.

3.2 초기 정보의 고속 획득 기법

(가) 계수 위치 탐색 기법

그림 7은 계수 위치 탐색 기법의 과정을 보여주며, 그림 8은 Information Initialization 블록 내부에 있는 Coefficient Detector 구조를 보여준다. Coefficient

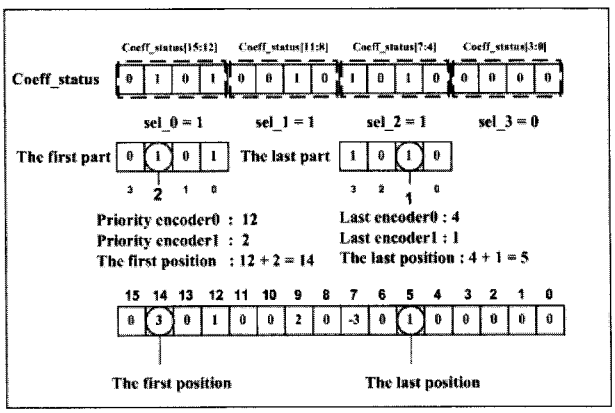


그림 7. 계수 위치 탐색 과정
Fig. 7. The detecting process of the coefficient position.

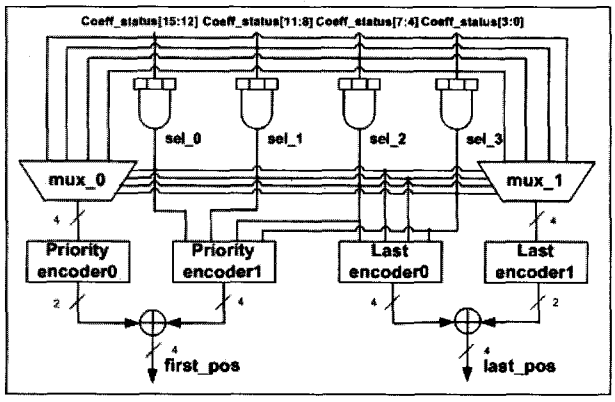


그림 8. Coefficient Detector 구조
Fig. 8. The structure of Coefficient Detector.

Detector 블록은 [9]의 구조를 응용하여 설계한 블록으로 zigzag 방향으로 정렬 된 계수들 중에서 '0'이 아닌 계수의 처음 위치와 마지막 위치를 구하는 블록이다.

계수 위치 탐색 과정은 다음과 같다. 먼저 그림 4의 Zigzag scan 블록에서 계산된 '0'이 아닌 계수들의 존재 유무를 판단하는 Coeff_status 신호를 4파트로 나눈다. 이렇게 나누어진 각각의 파트에 대해 계수들의 존재 여부를 OR 연산을 통해 계산하며, 계수가 존재하면 1을 출력하고 그렇지 않으면 0을 출력한다. 그 결과가 각 파트에 대해 차례대로 sel_0, sel_1, sel_2, sel_3 이다.

'0'이 아닌 계수가 존재하는 마지막 위치를 구하는 경우에는 먼저 그림 8의 Priority encoder1에서 위에서 구한 4비트 값 중 (sel_0, sel_1, sel_2, sel_3) 최하위에 존재하는 '1'의 위치에 따라 상위비트에서 하위비트 순으로 12, 8, 4, 0의 값을 할당한다. mux_0는 4비트 값 중 (sel_0, sel_1, sel_2, sel_3) 에서 최하위에 존재하는 '1'의 위치에 대응되는 파트를 선택하고, 그림 8의 Priority encoder0에서는 이렇게 선택된 파트 내의 4비트 값 중 최하위에 존재하는 1의 위치에 따라 상위비트에서 하위비트 순으로 3, 2, 1, 0의 값을 할당한다. 이렇게 구해진 두 값을 더한 값이 지그재그 방향으로 정렬 된 계수 중에서 '0'이 아닌 계수가 존재하는 마지막 위치가 되며, '0'이 아닌 계수의 처음 계수 위치는 이와 비슷한 과정을 통해 구할 수 있다.

이와 같이 계수 위치 정보를 이용하여 불필요한 계수 탐색 시간을 줄이고, Total Coefficients와 Total Zeros를 한 클록에 구한다. 또한 Run_before를 부호화 할 때 모든 계수를 탐색 할 필요 없이 '0'이 아닌 계수가 존재하는 처음 위치와 마지막 위치 사이 값들만 부호화 하여 부호화 수행시간을 크게 줄인다.

(나) 레벨 순차 정렬 기법

계수 위치 탐색 기법을 적용하면 Total Coefficients, Total Zeros, Run_before 부호화를 효율적으로 할 수 있다. 하지만 Trailing signs와 Trailing ones를 제외한 레벨 값들을 부호화 하는 과정에서 블록내의 '0'이 아닌 계수들의 존재 유무를 판단하여 그때마다 레벨 값들을 부호화 하게 되면 컨트롤 로직 구성이 복잡해질 뿐만 아니라 불필요한 계수 탐색으로 인해 부호화 수행시간이 증가하게 된다. 이로 인해 CAVLC 부호화기의 성능이 제한되는 것을 피하기 위해서 그림 9와 같이 Zigzag scan 블록에서 계산된 Cf_status 신호를 이용하여 zigzag 방향으로 정렬된 계수들 중 '0'이 아닌 계수 값들을 레지스터에 순차적으로 저장하는 레벨 순차 정렬 기법을 적용한다.

그림 10은 Information Initialization 블록 내부에 있는 Level Arrangement 블록의 세부적인 구조를 보여주

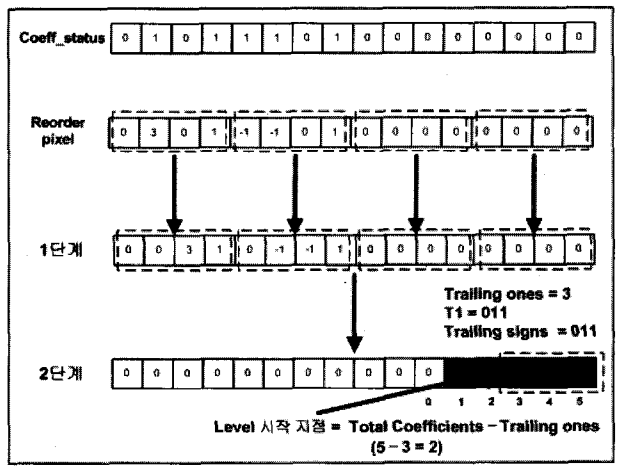


그림 9. 레벨 순차 정렬 과정
Fig. 9. Rearrangement of zigzag scanned coefficients.

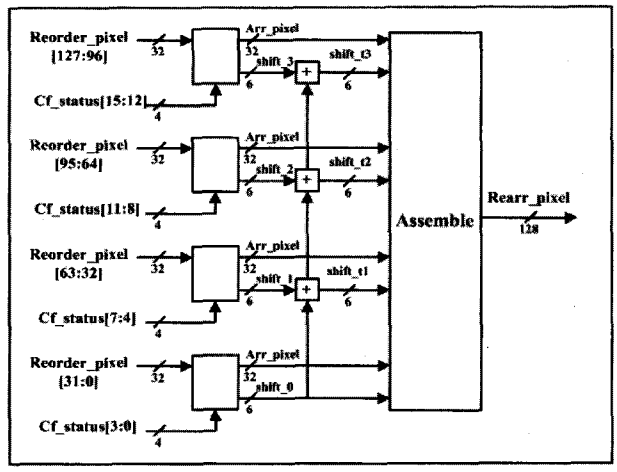


그림 10. Level Arrangement 구조
Fig. 10. The structure of Level Arrangement.

며, 레벨 값들을 레지스터에 순차적으로 저장하는 과정은 다음과 같다.

이 과정은 zigzag 방향으로 정렬된 계수들을 4 파트로 나누고 그에 대응하는 Cf_status 신호의 상태에 따라서 그림 9의 1단계와 같이 각 파트의 '0'이 아닌 계수들을 레지스터에 순차적으로 정렬한다. 그리고 각 파트의 계수 '0'의 개수를 계산한다. 2단계에서는 각 파트에서 계산된 계수 '0'의 계수를 상위 파트 단계로 가면서 누적을 한다. 그런 다음 Assemble 블록에서 각 파트에서 출력된 레지스터에 저장되어 있는 결과 값들을 쉬프트기를 이용하여 적절한 순서로 조합하여 레벨 값들을 순차적으로 정렬한다.

레벨 값들이 레지스터에 순차적으로 저장되면, CAVLC 알고리즘에서 Trailing ones의 최대 개수가 3이라는 특성을 이용하여 Trailing ones를 한 클록에 구하고, Trailing signs를 한 클록에 부호화 한다. 이 과정은 다음과 같다. 먼저 레벨 값들을 순차적으로 저장한 레지스터에서 하위 3개 값들 중 계수 ±1이 존재하는지 확인한다. 만약 계수 ±1이 존재하면 그 수만큼이 Trailing ones가 되고, 여기서 하위 3개 계수 값들에 대해 +1 이면 0의 값을, -1 혹은 나머지 값들의 경우에는 1을 할당한 T1 신호를 생성한다. 이렇게 계산된 3비트 신호(T1)와 Trailing ones는 그림 4의 T1_Sign & Levels 블록에 입력된다. T1_Sign & Levels 블록은 T1 신호 값을 Trailing ones 값만큼 선택하여 Trailing signs를 부호화 한다. Trailing signs 부호화가 끝나면 Trailing ones를 제외한 레벨 값들을 순차적으로 부호화 한다.

이와 같은 레벨 순차 정렬 기법을 적용함으로써 불필요한 계수 탐색 시간을 줄이게 되고, 레벨 값들을 순차적으로 부호화하기 때문에 레벨 값을 부호화하는 컨트롤 로직도 간단하게 구성된다. 또한 Trailing ones를 한 클록에 구하고 Trailing signs를 한 클록에 부호화함으로써 CAVLC 부호화 수행 시간을 줄이게 된다.

Ⅳ. 성능 분석

양자화 된 변환계수들을 부호화하기 위한 CAVLC 부호화기는 크게 초기화 단계와 부호화 단계로 나누어진다. 제안하는 부호화기는 부호화를 하기 위해 필요한 초기 정보 값들을 얻기 위해 4x4 블록 내의 16개의 계수들을 zigzag 방향으로 모두 탐색하지 않는다. 대신에 zigzag 방향으로 한 클록 내에 블록 내의 계수들을 정렬하고 앞장에서 제안한 계수 위치 탐색 기법과 레벨 순차 정렬 기법 두 가지 방식을 이용하여 초기화 정보를 빨리 획득한다. 또한 CAVLC 5단계 부호화를 순차적으로 처리하지 않고 동시에 병렬로 처리하여, 각각의 부호화가 끝나면 이 각 부호화기의 비트 스트림을 적절한 순서로 조합하는 방법을 사용하여 부호화 수행시간을 크게 줄인다.

CAVLC 부호화기는 CBP 정보와 블록 내의 계수 상태에 따라서 성능이 달라지며, 제안된 CAVLC 하드웨어 성능을 확인하기 위해서 기존의 방식[8]과 비교하여 QP(Quantization Parameter) 값들을 변화시키면서 하나의 매크로블록을 처리하는데 소요되는 평균 클록 수(Cycles/MB)를 시뮬레이션 하였다. 제안하는 방식은 [8]의 방식과는 달리 불필요한 계수 탐색 수행시간을 줄이는 기법을 적용하고, 5단계 부호화를 병렬로 처리함으로써 총 수행시간을 줄였으며, 그림 11의 그래프는 다양한 특성을 가진 QCIF 영상을 가지고 시뮬레이션 한 결과를 보여준다. 실험 결과 제안하는 구조가 기존의 구조[8]와 비교했을 때 하나의 매크로블록을 처리하는 평균 클록 수가 약 23%정도 줄었음을 확인하였다.

제안하는 CAVLC 부호화기는 Verilog HDL로 디자인 되었고 0.18um 표준 셀 라이브러리로 합성한 결과 16.3k 게이트를 가짐을 확인하였다. 표 1은 기존에 제안되었던 구조와 하드웨어 성능을 비교 분석한 표이다. 제안한 구조는 블록내의 불필요한 계수 탐색 시간을 줄여 하드웨어 성능을 향상시켰으며, 표에 나타난 바와 같이 기존의 구조보다 성능을 우수함을 확인할 수 있다.

실험 결과를 통해 제안하는 CAVLC 하드웨어 구조가 로직 사이즈 대비 높은 처리량을 가진다는 것을 확인하였으며, 이를 기반으로 제안하는 구조가 실시간 HD급 실시간 영상을 처리할 수 있는 H.264/AVC의 핵심 모듈로 사용하기에 우수한 구조임을 알 수 있다.

표 1. 비교 분석
Table 1. comparison and analysis.

Architecture	Chen[8]	Proposed
Technology	0.18um	0.18um
Gate Count	17635	16385
Clock Frequency	100MHz	81MHz
Target Format	HD1080 30fps	HD1080 30fps

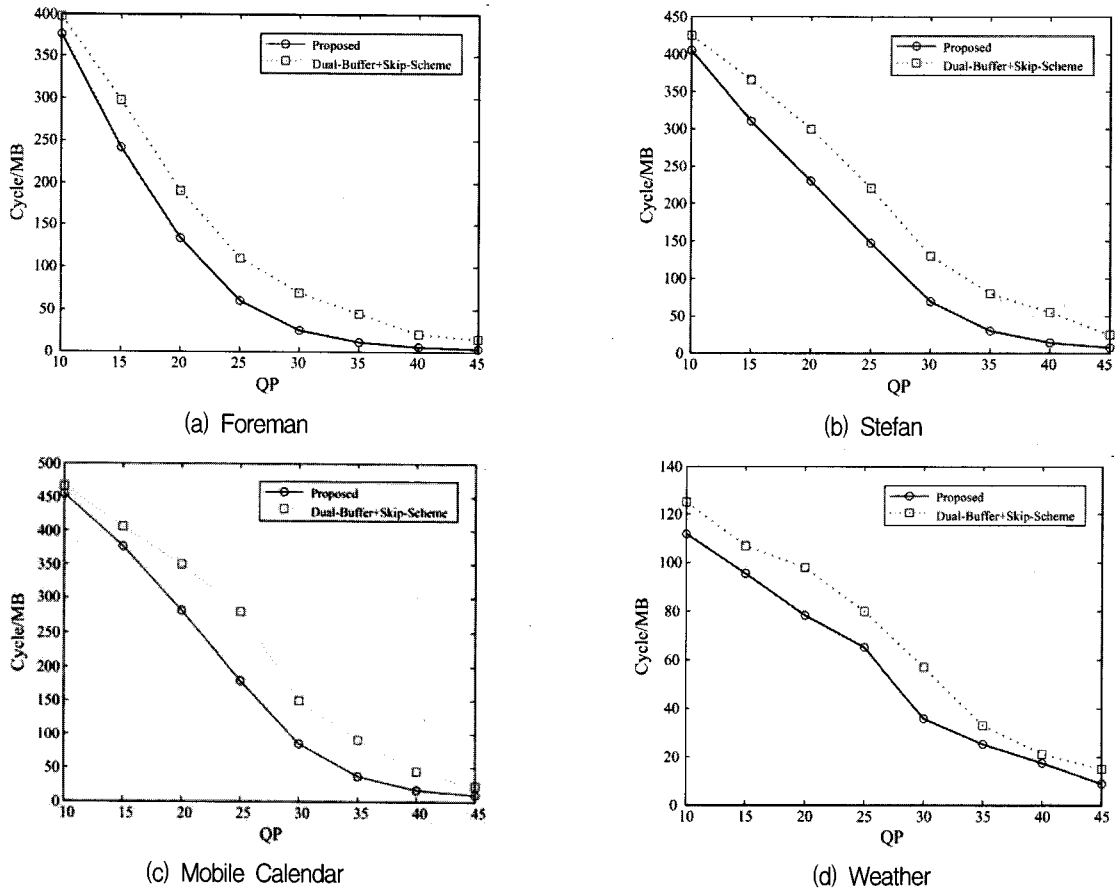


그림 11. 제안하는 CAVLC 부호화기의 성능 비교
Fig. 11. Performance comparison of the proposed CAVLC encoder.

V. 결 론

본 논문에서는 HD급 영상을 실시간으로 처리할 수 있는 H.264/AVC를 위한 효율적인 CAVLC 하드웨어 설계 구조를 제안하였다. 제안된 계수 탐색 기법과 레벨 순차 정렬 기법을 통해 불필요한 계수 탐색 시간을 줄임으로써 총 부호화 수행 시간을 줄였다. 제안된 구조를 적용하여 실험한 결과, 하나의 매크로블록을 처리하는 평균 클럭 수(Cycles/MB)는 기존 방식보다 약 23%가 줄었다. 제안된 하드웨어 구조는 Verilog HDL을 이용하여 설계 및 검증되었고, HD1080@30fps 영상을 81MHz에서 동작시킬 수 있음을 확인하였다.

참 고 문 헌

[1] J. V. 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

[2] M. Zhou, "Evaluation and Simplification of H.26L Baseline Coding Tools," JVT-B030, Jan,2002.

[3] Okubo Sakae, Kadono Shinya, Kikuchi Yoshihiro, Suzuki Teruhiko 저, 정제창 역, "H.264/AVC 비디오 압축 표준", 홍릉 과학 출판사

[4] Richardson, I. E. G. "H.264 and MPEG-4 Video Compression" Video Coding for Next-generation Multimedia, John Wiley & Sons Ltd., Sussex, England, December 2003.

[5] Wiegand, T., and Sullivan, G., "Draft Errata List with Revision-Marked Corrections for H.264/AVC," JVT Document JVT-I050, SanDiego, California, September 2003.

[6] Ihab Amer, Wael Badawy, and Graham Jullien. "TOWARDS MPEG-4 PART 10 SYSTEM ON CHIP : A VLSI PROTOTYPE FOR CONTEXT-BASED ADAPTIVE VARIABLE LENGTH CODING(CAVLC)," Signal Processing Systems, 2004, SIPS IEEE Workshop, Page(s) : 275 - 279.

[7] Yeong-Kang Lai; Chih-Chung Chou; Yu-Chieh Chung; "A Simple and Cost Effective Video

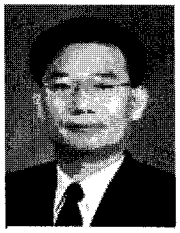
Encoder with Memory-Reducing CAVLC", ISCAS 2005. IEEE International Sysposium on Volume 1, 18-20 Sept. 2003 Page(s):323-326 Vol.1 [2]

[8] Chen T.; Huang Y.; Tsai C.; Hsieh B.; Chen L.; "Architecture Design of Context-Based Adaptive Variable Length Coding for H.264/AVC", IEEE Transactions on Circuit and Systems : Analog and Digital Signal Processing Vol PP, 2006 Page(s) : 832-836.

[9] Wu Di, Gao Wen, Hu Mingzeng, Ji Zhenzhou "An Exp-Golomb encoder and decoder architecture for JVT/AVS" ASIC,2003, Proceedings, 5th International Conference on Volume2, 21-24 Oct, 2003 Page(s) : 910 - 913

저 자 소 개

우 정 욱(학생회원)
2005년 경희대학교 전자공학과 학사 졸업.
2007년 연세대 전기전자공학과 석사 졸업.
2008년~현재 삼성전자 정보통신 총괄 통신연구소 재직
<주관심분야 : 영상처리, SoC 설계>



김 재 석(정회원)
1977년 연세대 전자공학과 학사 졸업.
1979년 KAIST 전기전자공학과 석사 졸업.
1988년 Rensselaer Polytechnic Institute, NY, 박사 졸업.
1993년~1995년 한국전자통신연구원 책임연구원
1996년~현재 연세대학교 전기전자공학과 교수
<주관심분야 : 통신 및 영상 시스템, VLSI 신호처리, 임베디드 S/W 및 SoC 구현>

이 원 재(학생회원)
2001년 연세대 전기전자공학과 학사 졸업.
2003년 연세대 전기전자공학과 석사 졸업.
2003년~현재 연세대 전기전자 공학과 박사 과정.
<주관심분야 : 영상처리, SoC 설계>