
SAO의 성능개선을 위한 저면적 하드웨어 설계

최지수* · 류광기**

Area Efficient Hardware Design for Performance Improvement of SAO

Jisoo Choi* · Kwangki Ryoo**

이 논문은 교육과학부와 한국연구재단의 지역혁신인력양성사업 및 지식경제부 출연금으로 수행한 ETRI SW-SoC 융합 R&BD 센터와의 공동 연구의 결과임

요 약

본 논문에서는 고성능 HEVC 복호기 설계를 위해 SAO(Sample Adaptive Offset)의 수행시간 단축과 연산량, 하드웨어 면적 감소를 위한 하드웨어 구조를 제안한다. 제안하는 SAO 하드웨어 구조는 8x8 CU(Coding Unit)를 처리하는 연산기를 구성하여 하드웨어 면적을 최소화하고, 내부레지스터를 이용하여 64x64 CU의 처리를 지원한다. 또한 기존 SAO의 top-down 블록분할 구조 대신 bottom-up 블록분할 구조로 설계하여 연산시간 및 연산량을 최소화한다. 제안한 SAO 하드웨어를 TSMC 0.18 μ m CMOS 표준 셀 라이브러리 이용해 합성한 결과 게이트 수는 30.7k개의 로직 게이트로 구현되며 최대동작주파수는 250MHz이다. 제안한 SAO 하드웨어 구조는 하나의 매크로 블록을 복호화하는데 64사이클이 소요된다.

ABSTRACT

In this paper, for HEVC decoding, an SAO hardware design with less processing time and reduced area is proposed. The proposed SAO hardware architecture introduces the design processing 8x8 CU to reduce the hardware area and uses internal registers to support 64x64 CU processing. Instead of previous top-down block partitioning, it uses bottom-up block partitioning to minimize the amount of calculation and processing time. As a result of synthesizing the proposed architecture with TSMC 0.18 μ m library, the gate area is 30.7k and the maximum frequency is 250MHz. The proposed SAO hardware architecture can process the decode of a macroblock in 64 cycles.

키워드

HEVC, 적응적 샘플 오프셋, 비디오 코딩, 루프내필터

Key word

HEVC, Sample Adaptive Offset, Video Coding, In-loop filter

* 준회원 : 한밭대학교 정보통신공학과

** 종신회원 : 한밭대학교 정보통신공학과(교신저자, kkryoo@hanbat.ac.kr)

접수일자 : 2012. 12. 07

심사완료일자 : 2013. 01. 02

Open Access <http://dx.doi.org/10.6109/jkiice.2013.17.2.391>

©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.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

최근 HDTV의 대중화에 따라 사용자들이 고화질, 고 해상도의 영상에 익숙해지면서 HDTV 이후의 차세대 영상기기에 대한 관심이 증가하고 있다. 차세대 영상기 기들은 1920x1080 해상도인 Full-HD급을 넘어서 HDTV 의 4배 이상 해상도를 갖는 UHD(Ultra High Definition; 7680x4320)급의 초고해상도 영상을 지원하도록 개발되 고 있다. 이러한 개발 방향에 맞추어 최근 비디오 압축 기술 표준을 주도하는 ISO/IEC MPEG(Moving Picture Experts Group)과 ITU-T VCEG(Video Coding Experts Group)은 초고해상도, 초고화질의 영상에 대한 압축 기 술의 필요성을 공감하여 공동으로 차세대 비디오 코덱 기술인 HEVC (High Efficiency Video Coding)를 표준화 하고 있다. HEVC는 초고해상도, 초고화질 영상을 지원 하기 위해 기존 비디오 코덱 기술인 H.264/AVC 기술 표 준에 비해 2배 이상의 압축 효율을 목표로 하고 있으며, 넓은 범위의 해상도와 무손실 코덱, 그리고 스케일러블 비디오 코딩 등을 고려하고 있다[1].

새롭게 채택된 HEVC의 루프내 필터링 기술인 SAO 은 부호화 과정을 통해 발생하는 원본 영상과 복원 영 상 간의 왜곡을 화소단위의 offset을 통해 보상하여, 주 관적 화질 및 부호화 효율을 향상시킨다. SAO는 화소 단위로 연산을 수행하기 때문에 최대 블록 크기인 64x64를 처리하기 위해서는 많은 연산량과 연산시간 이 필요하다. 또한 기존 SAO의 블록분할 구조는 top-down으로 최적의 SAO 블록분할과 SAO type을 결정하 기 위해 0-depth부터 3-depth까지 4번을 반복하여 수행 한다.

따라서 본 논문에서는 SAO의 높은 연산량과 연산시 간을 감소시키고, 하드웨어 설계 시 면적을 최소화하는 하드웨어 구조를 제안한다. 제안하는 하드웨어 구조는 8x8 연산기로 구성되며, 내부 레지스터를 이용하여 64x64 연산에 필요한 화소를 저장하는 구조로 구현하였 다. 또한 제안하는 블록분할 구조는 bottom-up방식으로 최하위 depth를 먼저 연산하여 상위 depth에서 하위 depth의 연산결과를 사용하는 구조로써 연산시간 및 하 드웨어 면적을 최소화 한다.

본 논문의 구성은 다음과 같다. 2장에서는 SAO 기술 을 기술하며, 3장에서는 제안하는 SAO 하드웨어 구조를 기술한다. 4장에서는 제안하는 구조의 검증 및 성능 비

교에 대해 기술하고, 5장에서는 본 연구의 결론을 도출 한다.

II. SAO

SAO 기술은 부호화 과정을 통해 발생하는 원본 영상 과 복원 영상 간의 차이를 이용하여 복원 영상의 왜곡을 감소시키는 방법으로, 주관적 화질과 부호화 효율을 향 상시킨다. 특히 왜곡의 정도에 따라 적응적으로 offset을 적용함으로써, 영상의 특성에 따라 부분적으로 다르게 발생하는 왜곡을 효율적으로 최소화할 수 있다. 식 (1)과 같이 부호화기가 N 개의 화소에 대해 원본 화소(orgn)와 복원 화소(recn) 간의 오차를 평균적으로 최소화하는 offset을 계산하고, 이를 복호화기에 전송하여 식 (2)와 같이 복원 화소의 왜곡을 보상할 수 있다[2].

$$offset = \frac{\sum_{n=0}^{N-1} (org_n - rec_n)}{N} \quad (1)$$

$$rec'_n = rec_n + offset \text{ where } n = 0, 1, \dots, N-1 \quad (2)$$

SAO 기술은 원본 영상과 복원 영상 간의 왜곡을 직접 계산하고 복원하기 때문에 주관적 화질뿐만 아니라 객 관적 화질(PSNR)을 향상시킨다. 또한 왜곡이 보상된 영 상을 참조 영상으로 활용함으로써 부호화 효율을 향상 시킬 수 있다.

2.1. SAO 블록분할

일반적으로 양자화 과정을 통해 발생하는 왜곡은 예 측의 정확도에 따라 서로 다르게 나타난다. 따라서 SAO 는 왜곡의 정도에 따라 블록을 분할하고, 다양한 SAO type을 적용하는 블록 단위의 적응적 구조를 갖는다[3].

SAO 블록분할은 그림 1에서와 같이, quadtree 기반의 CU 분할 방법과 동일한 방식으로 LCU(Largest CU) 블록 을 최대 4-depth까지의 하위 블록으로 분할할 수 있으며, 분할된 각 하위 블록에는 서로 다른 SAO type이 적용될 수 있다. 부호화기는 울-왜곡 최적화 방법을 사용하여 부호화 효율 측면에서 최적인 SAO 블록분할 여부 및 SAO type을 결정하고, 이에 해당하는 정보를 복호화기 에 전송한다. 따라서 블록의 분할 여부를 표현하는 정보

(sao_split_flag) 및 분할된 각 하위 블록에 적용된 SAO type을 표현하는 정보(sao_type_idx)가 추가적으로 전송된다.

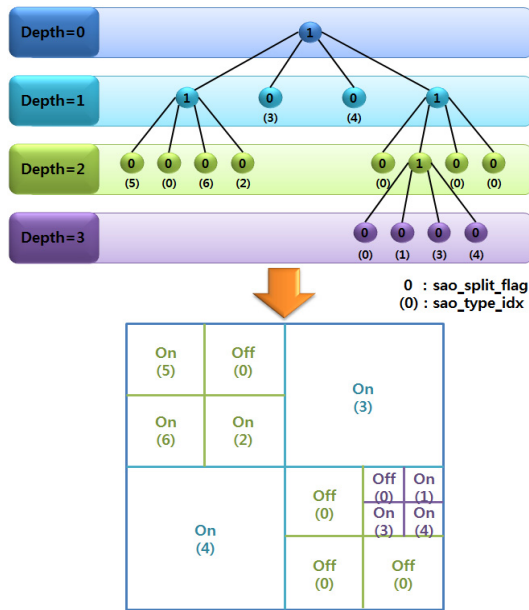


그림 1. SAO 분할구조 및 SAO type 적용
Fig. 1 Application of SAO Partitioning Architecture and SAO type

2.2. SAO type

분할된 각 하위 블록에 적용 가능한 SAO type은 표 1과 같다. SAO type은 SAO를 적용하지 않는 방법과 EO(Edge Offset) 및 BO(Band Offset)으로 SAO를 적용하는 방법으로 구분된다.

표 1. SAO type 정보
Table. 1 SAO type Information

sao_type_idx	SAO type to be used	Number of categories
0	None	0
1	1-D 0-degree pattern edge offset	4
2	1-D 90-degree pattern edge offset	4
3	1-D 135-degree pattern edge offset	4
4	1-D 45-degree pattern edge offset	4
5	4 consecutive band offset	4

EO 및 BO은 해당 화소의 카테고리를 구분하기 위한 방법으로 EO은 0-degree, 90-degree, 135-degree, 45-degree 총 4가지 방향으로 구성되며, BO은 화소가 속할 수 있는 값을 일정한 크기의 32개 구간으로 나누고 윗-왜곡측면에서 최적의 연속된 4개의 오프셋을 결정한다. EO은 그림 2같이 현재 위치 화소 값(C)과 이웃한 두 개의 화소 값의 관계를 이용하는 방법이다. 4개의 패턴에 따라 해당 화소가 속하는 카테고리를 분류하고, 분류된 카테고리 별로 서로 다른 offset을 적용한다. 예를 들어, 해당 블록에 0-degree pattern의 EO가 적용되면, 해당 블록에 속한 모든 화소는 0-degree pattern으로 표 2와 같이 5개의 카테고리로 분류되고, 카테고리 0을 제외한 각각의 카테고리 단위별로 식 (1)이 적용되어 각각의 offset이 결정된다[4].

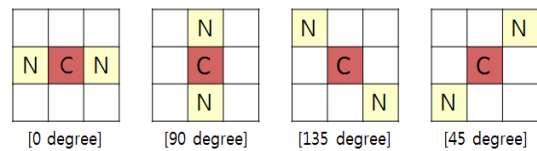


그림 2. EO의 4개 패턴
Fig. 2 The four patterns of EO

표 2. EO의 카테고리
Table. 2 Category of EO

Category	Condition
1	C < 2 neighboring pixels
2	C < 1 neighbor & C = 1 neighbor
3	C > 1 neighbor & C = 1 neighbor
4	C > 2 neighboring pixels
0	None of the above

BO는 화소 밝기 값을 이용한 방법으로 전체 화소 값을 32개의 구간으로 나눈 후, 각각의 밴드에 해당하는 offset을 결정한다. 이 중에서 윗-왜곡측면에서 최적인 연속된 4개의 offset을 결정하여 그 위치를 가리키는 band_position과 사용된 4개의 offset을 전송한다. 그림 3은 BO 밴드에서 선택된 band_position과 연속된 4개의 오프셋의 예시를 보여준다.

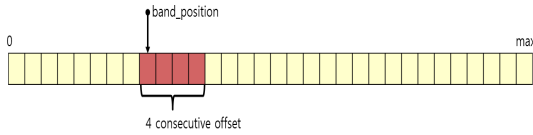


그림 3. Band Offset 밴드
Fig. 3 Band Offset Band

SAO 기술은 율-왜곡 최적화 방법을 사용하여 부호화 효율 측면에서 최적인 SAO 블록분할 여부 및 SAO type 을 결정한다. top-down 방식으로 0-depth의 율-왜곡 비용과 1-depth의 율-왜곡 비용과 비교하여, 최적의 율-왜곡 성능을 보이는 SAO 블록분할 구조 및 SAO type을 결정한다. 1-depth이 결정되면 위와 같은 방법으로 1-depth와 2-depth의 율-왜곡 비용을 비교하게 된다. 총 3-depth까지 수행을 하며 최적의 SAO 블록분할과 SAO type을 결정한다.

이와 같은 수행 방법은 0-depth에서 3-depth까지 4번을 수행해야하며 하드웨어로 구현시 각 depth에 해당하는 연산기를 설계해야 하는 문제점이 발생한다. 이러한 문제점은 HEVC 부호기 및 복호기의 성능에도 큰 영향을 준다.

III. 제안하는 SAO 하드웨어 구조

제안하는 SAO 하드웨어 구조는 디블록킹 필터에서 처리되는 영상의 CU크기를 고려하여 8x8 CU를 처리하는 연산기 구조를 사용하고 64x64 LCU까지 처리 가능하다. 또한, 기존 SAO 기술의 top-down 방식 대신 bottom-up 방식으로 설계하여 SAO 블록분할과 SAO type을 복호화하는 연산시간을 최소화 하였다. 그림 4는 제안하는 bottom-up 방식으로 최하위 depth인 3-depth부터 SAO 연산을 수행하며, 3-depth의 SAO 카테고리 결과들을 2-depth에서 사용하는 방식을 갖는다. 또한 3-depth와 2-depth의 율-왜곡 비용을 비교하여 최적의 SAO type과 SAO 블록분할을 결정한다. 상위 depth 연산에 필요한 화소는 내부 레지스터에 저장하여 블록의 경계 부분만 SAO를 수행하는 구조로 설계하였다.

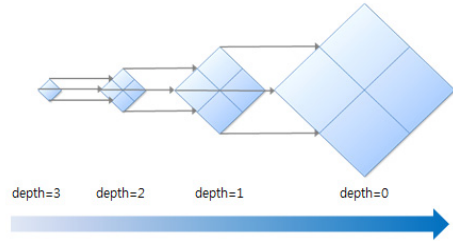


그림 4. 제안하는 bottom-up 구조
Fig. 4 Proposed bottom-up Architecture

그림 5는 제안하는 SAO 하드웨어 구조를 나타낸다. 제안하는 구조는 8x8 크기의 매크로블록 연산을 수행하며 64개의 edge_band_offset으로 구성된다. edge_band_offset은 하나의 픽셀을 처리하며 64x64를 처리하기 위해 64개가 필요하다.

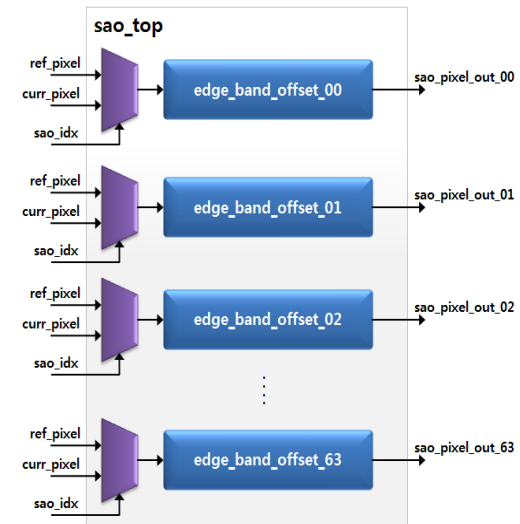


그림 5. SAO 하드웨어 구조
Fig. 5 SAO Hardware Architecture

그림 6은 제안하는 edge_band_offset 하드웨어 구조이며, 총 4개의 edge_offset, band_offset, category_select, offset_adder로 구성된다.

edge_offset은 현재 픽셀과 참조 픽셀을 이용하여 edge에 해당하는 offset을 선택하고 band_offset은 band_position 값과 현재 픽셀을 가지고 band에 해당하는 offset을 선택한다. category_select은 sao_idx에 따라 edge_cate

신호와 band_cate 신호 중 하나를 선택하여 cate_sel 신호를 출력한다. 출력된 cate_sel 신호에 의해 4개의 offset 중 하나의 offset이 선택된다. 선택된 offset 값은 offset_adder에서 현재 픽셀과 덧셈을 수행하여 최종적으로 SAO가 적용된 픽셀 값을 출력한다.

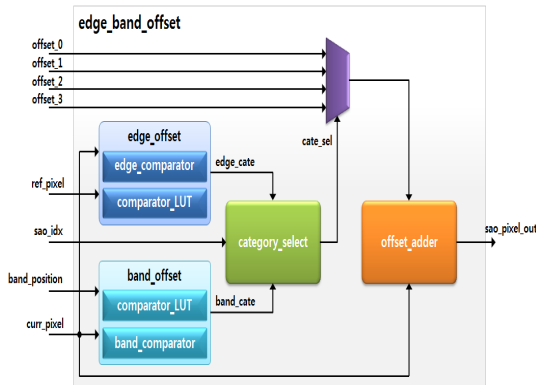


그림 6. edge_band_offset 하드웨어 구조
Fig. 6 edge_band_offset Hardware Architecture

IV. 실험 및 고찰

제안하는 구조의 동작을 검증하기 위해서 HEVC 표준 참조 소프트웨어 HM-8.0[5]에서 SAO를 적용하기 전 영상을 추출한 후 제안한 하드웨어 구조로 SAO를 적용한 영상을 시뮬레이션 한 결과, 그림 7과 같이 참조 소프트웨어에서 추출한 영상과 비교하여 정상적으로 동작함을 확인하였다.



그림 7. 제안하는 구조의 검증 결과
Fig. 7 Verification result of the Proposed Architecture

그림 8은 제안하는 SAO에 대한 시뮬레이션 결과이다. sao_idx 신호값에 따라 입력으로 들어오는 offset 값이 선택되어 sao_pixel_out 신호 값으로 출력된다.

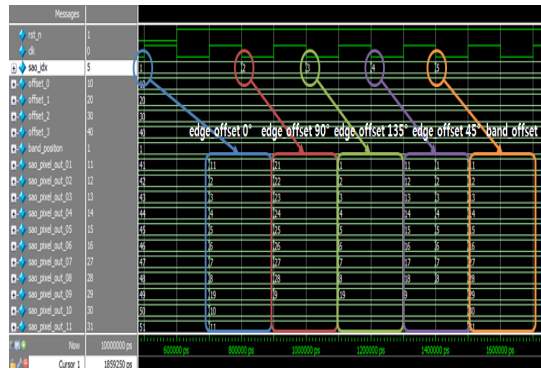


그림 8. SAO 시뮬레이션 결과
Fig. 8 Simulation Result of SAO

제안하는 하드웨어 구조는 Verilog HDL로 설계되었으며, TSMC 0.18um 표준 셀 라이브러리를 사용하여 IDEC에서 지원하는 CAD Tool인 Synopsys사의 Design Compiler로 합성하였다. 표 3은 제안하는 구조의 합성 결과를 나타낸다. 제안하는 구조의 게이트 수는 30.7k이고, 최대 동작 주파수는 250MHz이다.

제안하는 구조는 edge_offset과 band_offset을 연산하는 부분을 공통으로 사용하고 edge_offset과 band_offset에 해당하는 선택부분만 분리하여 처리함으로써 기존구조에 비해 게이트수와 처리 사이클 수가 감소하였다.

표 3. SAO 하드웨어의 합성 결과 및 비교
Table. 3 Synthesis Results and Comparison of SAO Hardware

	Previous[6]	Proposed	차이값
공정	0.18um	0.18um	-
동작 주파수	250MHz	250MHz	-
LCU 당 처리 사이클 수	128	64	64
게이트(k)	45.4	30.7	14.7

V. 결 론

본 논문에서는 HEVC 루프내 필터인 SAO의 연산시간과 연산량, 하드웨어 면적을 최소화하기 위해 8x8 블록 연산기를 사용하였고, LCU인 64x64을 지원하기 위해 내부 레지스터를 사용하여 8x8 블록 경계의 화소들을 저장하는 방식을 채택하였다. 또한 기존의 SAO 블록분할과 SAO type을 결정하기 위한 top-down 방식 대신 bottom-up 방식으로 설계하여 불필요한 연산 시간을 최소화하였다. 또한 EO와 BO를 하나의 공통 연산기로 연산함으로써 하드웨어 면적을 감소시켰다. 제안하는 구조를 TSMC 0.18um 공정을 이용하여 합성한 결과 최대 동작 주파수는 250MHz이고, 게이트 수는 30.7k이고 하나의 LCU를 처리하는데 64사이클이 소요된다.

참고문헌

[1] 김제우, 박지호, 김용환, 최병호, “HEVC(High Efficiency video Coding) 비디오코덱기술의 응용과 전망,” 방송공학회지, 제 15권, 제 4호, pp. 135-145, 2010.

[2] 양정엽, 원광현, 전병우, “In-loop filtering in HEVC,” 대한전자공학회지, 제 38권, 제 8호, pp. 46-55, 2011.

[3] JCT-VC of ISO/IEC MPEG and ITU-T VCEG, “CE13: Sample Adaptive Offset with LCU Independent Decoding,” JCT-VC, JCTVC-E049, Jan. 2010.

[4] C. M. Fu, C. Y. Chen, Y. W. Huang and S. Lei, “Sample Adaptive Offset for HEVC,” IEEE 13th International Workshop on Multimedia Signal Processing, pp. 17-19, Oct. 2011.

[5] HM 8 Reference Software. Available: https://hevc.hhi.fraunhoferde/svn/svn_HEVCSoftware/tags/HM-8.0

[6] 박승용, 류광기, “고성능 HEVC 복호기를 위한 효과적인 Sample Adaptive offset 하드웨어 설계,” 한국정보처리학회 추계학술발표대회, 제 19권, 제 2호, pp. 645-648, 2012.

저자소개



최지수(Jisoo Choi)

2008년 한밭대학교
정보통신공학과 공학사
2010년~현재 한밭대학교
정보통신공학과 석사과정

※ 관심분야: SoC 플랫폼 설계 및 검증, 영상 코덱 설계



류광기(Kwangki Ryoo)

1986년 한양대학교 공과대학
전자공학과 공학사
1988년 한양대학교 대학원
전자공학과 공학석사

2000년 한양대학교 대학원 전자공학과 공학박사
1991년~1994년 육군사관학교 교수부 전자공학과
전임강사

2000년~2002년 ETRI 시스템IC설계팀 선임연구원
2010년~2011년 Univ of Texas at Dallas 방문교수
2003년~현재 한밭대학교 정보통신공학과 교수

※ 관심분야: SoC 플랫폼 설계 및 검증, 하드웨어/
소프트웨어 통합설계 및 통합검증, 멀티미디어 코덱
설계