

HEVC 부호기를 위한 효율적인 SAO의 저면적 하드웨어 설계

조현표 · 류광기*

Low Area Hardware Design of Efficient SAO for HEVC Encoder

Hyunpyo Cho · Kwangki Ryoo*

Department of Information and Communication Engineering, Hanbat National University, Daejeon, Korea

요 약

본 논문에서는 HEVC(High Efficiency Video Coding) 부호기를 위한 효율적인 SAO(Sample Adaptive Offset)의 저면적 하드웨어 구조를 제안한다. SAO는 HEVC 영상 압축 표준에서 채택된 새로운 루프 내 필터 기술로서 최적의 오프셋 값들을 화소 단위로 적용하여 영역 내 평균 화소 왜곡을 감소시킨다. 하지만 표준 SAO는 화소 단위 연산을 수행하기 때문에 초고해상도 영상을 처리하기 위해서 많은 연산시간과 연산량을 요구한다. 제안하는 SAO 하드웨어 구조는 SAO의 연산시간을 감소시키기 위해서 한번에 4개의 입력 화소들을 병렬적으로 처리하며, 2단계 파이프라인 구조를 갖는다. 또한 하드웨어 면적을 최소화하기 위해서 휘도 성분과 색차 성분에 대해 단일 구조를 가지며, 하드웨어에 적합한 연산기 및 공통 연산기를 사용한다. 제안하는 SAO 하드웨어 구조는 Verilog HDL로 설계하였으며, TSMC 0.13 μ m CMOS 표준 셀 라이브러리로 합성한 결과 약 190k개의 게이트로 구현되었다. 제안하는 SAO 하드웨어 구조는 200MHz의 동작주파수에서 4K UHD@60fps 영상의 실시간 처리가 가능하며, 최대 250MHz까지 동작 가능하다.

ABSTRACT

This paper proposes a hardware architecture for an efficient SAO(Sample Adaptive Offset) with low area for HEVC(High Efficiency Video Coding) encoder. SAO is a newly adopted technique in HEVC as part of the in-loop filter. SAO reduces mean sample distortion by adding offsets to reconstructed samples. The existing SAO requires a great deal of computational and processing time for UHD(Ultra High Definition) video due to sample by sample processing. To reduce SAO processing time, the proposed SAO hardware architecture processes four samples simultaneously, and is implemented with a 2-step pipelined architecture. In addition, to reduce hardware area, it has a single architecture for both luma and chroma components and also uses optimized and common operators. The proposed SAO hardware architecture is designed using Verilog HDL(Hardware Description Language), and has a total of 190k gates in TSMC 0.13 μ m CMOS standard cell library. At 200MHz, it can support 4K UHD video encoding at 60fps in real time, but operates at a maximum of 250MHz.

키워드 : HEVC, SAO, 루프 내 필터, VLSI

Key word : HEVC, SAO, In-loop filter, VLSI

접수일자 : 2014. 11. 15 심사완료일자 : 2014. 12. 15 게재확정일자 : 2014. 12. 30

* **Corresponding Author** Kwangki Ryoo(E-mail:kkryoo@gmail.com, Tel:+82-42-821-1710)

Department of Information and Communication Engineering, Hanbat National University, Daejeon 305-791, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2015.19.1.169>

print ISSN: 2234-4772 online ISSN: 2288-4165

©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 영상 및 통신 기술이 빠르게 발전함에 따라 고 해상도 영상을 지원하는 HDTV와 멀티미디어 기기의 대중화가 이루어졌고, 고해상도 영상 서비스에 대한 사용자들의 관심과 수요가 증가하였다. 차세대 영상 기기들의 개발 방향은 1920x1080 해상도인 FHD(Full High Definition)급을 넘어서 4배 이상의 해상도인 4K, 8K UHD(Ultra High Definition)급의 초고해상도 영상으로 발전하고 있다.

영상의 해상도가 증가하면서 새로운 영상 압축 표준 기술의 필요성이 대두되었고, 국제 표준화 기관인 ITU-T의 VCEG(Video Coding Experts Group) 그룹과 ISO/IEC의 MPEG(Moving Picture Experts Group) 그룹은 공동으로 JCT-VC(Joint Colaborative Team on Video Coding)를 결성하여 차세대 영상 압축 표준 기술인 HEVC를 개발하였다.

HEVC는 초고해상도 영상과 병렬 처리 구조를 주요 이슈로 개발되었으며, 2013년 4월에 국제 표준화로 제정되었다. HEVC는 저해상도 영상부터 초고해상도 영상까지 다양한 해상도의 영상 압축을 지원하고, 부호화 효율을 개선하기 위해 다양한 영상 압축 기술들을 포함하고 있다. HEVC는 이전 영상 압축 표준인 H.264/AVC와 비교하여 약 50% 이상 개선된 부호화 효율을 갖는다[1].

HEVC는 이전 영상 압축 표준들과 마찬가지로 블록 기반 부호화와 양자화 기술을 사용하면서, 고정된 8x8 매크로 블록 단위를 사용하여 변환 부호화를 수행하는 H.264/AVC와 다르게 4x4부터 32x32까지 다양한 크기의 블록 단위를 사용하여 변환 부호화를 수행한다. 하지만 변환 부호화의 블록 단위가 클수록 변환 계수들을 양자화하는 과정에서 발생하는 에러로 인하여 영상의 왜곡 현상은 더욱 강하게 발생한다[2]. 따라서 HEVC는 양자화 과정에서 발생하는 영상의 왜곡 현상을 감소시키기 위한 기술을 필요로 하였고, 루프 내 필터 기술로서 기존 DBF(DeBlocking Filter)와 함께 새로운 기술인 SAO를 채택하였다.

SAO는 영역 기반 필터로서 영역 내 화소 간 특성을 기반으로 최적의 오프셋 값을 구한 후, 화소 단위로 오프셋 값을 적용하여 영역 내 평균 화소 왜곡을 감소시키는 기술이다[3].

하지만 화소 단위로 연산을 수행하기 때문에 초고해상도 영상을 처리하기 위해서 많은 연산시간과 연산량을 요구한다[4].

본 논문에서는 SAO의 연산시간을 감소시키기 위해 4개 화소의 병렬처리와 2단계 파이프라인 구조를 사용하고, 하드웨어 면적을 최소화하기 위해 휘도와 색차 성분에 대한 단일 구조와 하드웨어에 적합한 연산기 및 공통 연산기를 사용한 효율적인 SAO 하드웨어 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 HEVC의 표준 SAO 기술에 대해 기술하고, 3장에서는 제안하는 SAO 하드웨어 구조를 기술한다. 4장에서는 하드웨어 구현 결과 및 성능 비교에 대해 기술하고, 5장에서 본 연구의 결론을 도출한다.

II. Sample Adaptive Offset

SAO는 영역 기반 필터로 DBF가 처리된 후 수행된다[5]. SAO의 기본 개념은 필터링을 수행하는 영역 내 화소들을 분류하고, 분류된 정보를 기반으로 최적의 오프셋 값을 구한 후 복원 화소에 오프셋 값을 적용함으로써 영역 내 평균 화소 왜곡을 감소시키는 것이다.

SAO는 영역 내 화소들을 분류하고 오프셋 값을 구하기 위해 두 가지 타입인 에지 오프셋과 밴드 오프셋을 사용한다[6]. 에지 오프셋은 영역 내 화소 간의 에지 특성을 고려하여 복원 영상의 왜곡을 효과적으로 감소시키기 위한 방법이다. 밴드 오프셋은 영역 내 화소 간의 유사한 밝기 값 특성을 고려하여 복원 영상의 왜곡을 효과적으로 감소시키기 위한 방법이다.

SAO는 원본 영상과 복원 영상 간의 왜곡을 감소시켜 영상의 주관적 화질 및 객관적 화질을 향상시킬 뿐만 아니라 부호화 효율도 향상시킨다.

2.1. 에지 오프셋

에지 오프셋은 영역 내 화소들의 에지 특성을 고려하여 화소들을 분류하기 위해서 1차원적인 방향성을 나타내는 4개의 패턴 클래스를 사용한다[7]. 그림 1은 에지 오프셋의 4가지 패턴 클래스를 나타낸다.

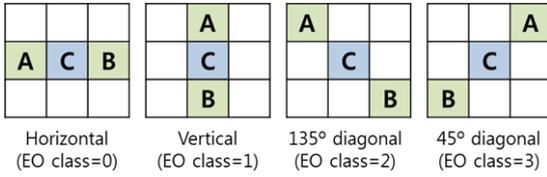


그림 1. 에지 오프셋의 4가지 패턴 클래스
Fig. 1 Four pattern classes for Edge Offset

그림 1의 각 클래스에 표시된 ‘C’는 현재 분류할 화소를 나타내고, ‘A’와 ‘B’는 각 방향성에 따른 주변 화소들을 나타낸다. 에지 오프셋은 그림 1과 같은 수평, 수직 클래스와 2개의 대각 클래스에 대한 주변 화소 정보를 얻고, 현재 분류할 화소와 2개의 주변 화소 정보를 표 1에 나타난 화소 분류 조건식에 따라 하위 카테고리 로 분류한다.

표 1. 에지 오프셋의 화소 분류 조건식

Table. 1 Sample classification conditions for Edge Offset

Category	Condition
1	$(C < A) \ \&\& \ (C < B)$
2	$((C < A) \ \&\& \ (C = B)) \ \parallel \ ((C = A) \ \&\& \ (C < B))$
3	$((C > A) \ \&\& \ (C = B)) \ \parallel \ ((C = A) \ \&\& \ (C > B))$
4	$(C > A) \ \&\& \ (C > B)$
0	None of the above

에지 오프셋은 표 1의 조건식을 사용하여 현재 분류할 화소를 4개의 패턴 클래스에 따른 주변 화소 정보와 비교하여 각각의 하위 카테고리로 분류하고, 분류된 정보를 기반으로 각 카테고리에 해당하는 최적의 오프셋 값을 구한다. 즉, 에지 오프셋의 4개 패턴 클래스들은 각각의 하위 카테고리 4개씩을 갖게 되고, 에지 오프셋은 분류된 정보들을 기반으로 총 16개의 오프셋 값을 구한다.

2.2. 밴드 오프셋

밴드 오프셋은 영역 내 화소들의 유사한 밝기 값 특성을 고려하여 화소들을 분류하는 방법이다. 밴드 오프셋은 화소의 밝기 값 범위를 일정한 구간을 갖는 밴드들로 나눈 후 분류할 화소의 밝기 값에 따라 해당하는 밴드로 분류한다.

그림 2는 8비트 심도를 갖는 화소 표현 체계에서 밴드 오프셋의 화소를 분류하기 위한 밴드 구성의 예를 나타낸다.

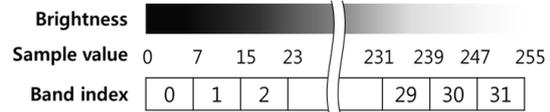


그림 2. 밴드 오프셋의 밴드 구성 예
Fig. 2 The band composition example for Band Offset

밴드 오프셋은 그림2와 같이 화소의 밝기 범위를 32개의 균일한 밴드 구간으로 나누고, 분류할 화소의 밝기 값에 따라 해당하는 밴드로 분류한다. 즉, 분류할 화소의 밝기 값이 21일 경우 3번째 밴드로 화소를 분류한다. 밴드 오프셋은 총 32개의 밴드에 대하여 각각 오프셋 값을 구한다.

2.3. SAO 파라미터

SAO는 에지 오프셋의 16개 오프셋 값들과 밴드 오프셋의 32개 오프셋 값들을 구할 때 각각의 울-왜곡 비용을 계산하여 최소의 울-왜곡 비용을 갖는 오프셋 값들을 최적의 오프셋 값으로 결정한다.

에지 오프셋과 밴드 오프셋에 대하여 모든 48개의 최적 오프셋 값들이 결정되면, 에지 오프셋에서는 각각의 클래스에 대한 울-왜곡 비용을 계산하고, 그 중 최소의 울-왜곡 비용을 갖는 최적의 클래스를 결정한다. 밴드 오프셋에서는 각각의 연속한 4개 밴드들로 이루어진 밴드 그룹에 대한 울-왜곡 비용을 계산하고, 그 중 최소의 울-왜곡 비용을 갖는 최적의 밴드 그룹을 결정한다.

마지막으로 SAO는 현재 영역에 대하여 최적의 SAO 파라미터를 결정하기 위해, 각각 Merge 파라미터와 에지 오프셋, 밴드 오프셋의 울-왜곡 비용을 비교하여 최소의 울-왜곡 비용을 갖는 정보를 최적 SAO 파라미터로 결정하며, SAO 파라미터는 타입 정보, 4개의 오프셋 값, Merge 플래그 정보 등을 포함한다.

III. 제안하는 SAO 하드웨어 구조

제안하는 SAO 하드웨어 구조는 표준 SAO 기술의 화소 단위 연산으로 인한 많은 연산시간을 감소시키기 위

해 4개 화소를 병렬적으로 처리하며, 화소를 분류하고 울-왜곡 최적화를 수행하여 최적의 SAO 파라미터를 생성하는 과정과 최적의 SAO 파라미터를 복원 화소에 적용하여 보상하는 과정을 2단계 파이프라인 구조로 구현하였다. 또한 하드웨어 면적을 최소화하기 위해 영상의 휘도 성분과 색차 성분을 단일 구조로 처리하여 하드웨어 자원을 재사용하도록 구현하였고, 하드웨어에 적합한 연산기 및 공통 연산기를 구현하여 사용하였다.

제안하는 SAO 하드웨어 구조의 전체 블록 다이어그램은 그림 3과 같다.

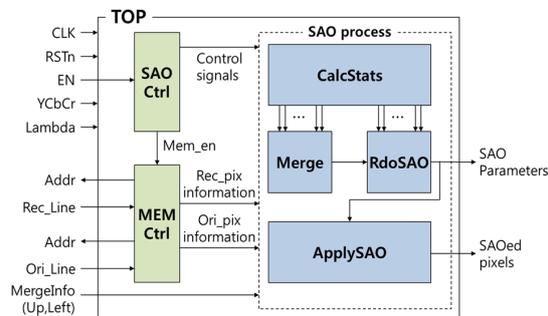


그림 3. 제안하는 SAO 하드웨어 구조
Fig. 3 The proposed SAO hardware architecture

제안하는 SAO 하드웨어 구조는 그림 3과 같이 SAOctrl 모듈에서 SAO의 전체 동작을 제어하는 신호를 생성하여 각각 SAO process 그룹과 MEMCtrl 모듈로 전송한다. MEMCtrl 모듈은 제어신호에 따라 LCU (Largest Coding Unit) 정보를 저장하고 있는 외부 메모리로부터 SAO를 수행할 화소 정보들을 읽고 저장한 후 수행 순서에 따라 필요한 화소 정보들을 SAO process 그룹으로 전송한다.

SAO process 그룹은 실제 SAO를 수행하는 CalcStats, Merge, RdoSAO, ApplySAO 모듈들로 구성된다. SAO process 그룹은 입력 화소들을 에지 오프셋과 밴드 오프셋으로 분류하며, 최적 오프셋 값들과 최적 SAO 파라미터를 결정하여 출력함과 동시에 SAO 파라미터 정보를 이용하여 복원 영상을 보상한다.

3.1. 화소 병렬처리를 위한 메모리 버퍼 구조

제안하는 SAO 하드웨어 구조는 4개 화소를 병렬적으로 처리함으로써 1개 화소 단위로 처리하는 표준

SAO 방식과 비교하여 연산시간을 4배 감소시킨다.

4개의 화소를 에지 오프셋과 밴드 오프셋으로 분류할 때, 밴드 오프셋의 경우 분류할 4개의 화소 정보만 필요하지만 에지 오프셋의 경우 4개의 패턴 클래스에 따른 각각 2개씩의 주변 화소 정보가 추가적으로 필요하다. 따라서 제안하는 SAO 하드웨어 구조는 4개의 화소를 병렬적으로 처리하기 위해 그림 4와 같은 2차원 배열 버퍼 구조를 사용한다.

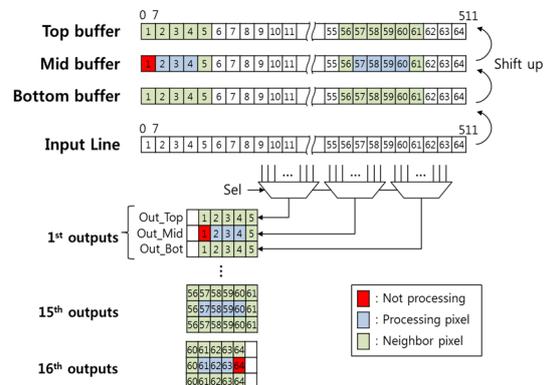


그림 4. 2차원 배열 버퍼 구조
Fig. 4 Two-dimensional array buffer architecture

2차원 배열 버퍼 구조는 그림 4와 같이 LCU(64x64) 정보를 저장하고 있는 외부 메모리로부터 한 사이클 당 64개 화소 정보인 Input Line을 입력받아 Bottom buffer와 Mid buffer, Top buffer로 쉬프트 업을 수행하여 저장한다. 그리고 SAO 수행 순서에 따라 각 버퍼에 저장된 화소 정보를 출력하여 전송한다.

2차원 배열 버퍼 구조는 메모리로부터 LCU의 처음 화소 정보들을 입력받을 때 Bottom buffer와 Mid buffer, Top buffer를 채우기 위해 3 사이클의 딜레이가 발생하지만, 다음 동작 시에는 한 사이클 당 한 줄의 화소 정보를 입력받아 저장할 수 있을 뿐만 아니라 메모리 접근 횟수를 최소화할 수 있고, 제안하는 SAO 하드웨어 구조와 대응되어 한 번에 4개의 복원 화소 정보와 주변 화소 정보를 효율적으로 전송하여 처리할 수 있다.

3.2. 화소 분류 구조

SAO는 입력 화소를 각각 에지 오프셋과 밴드 오프셋으로 분류한 후 분류된 정보를 기반으로 최적 오프셋

값들을 계산한다. 분류된 정보란 화소가 분류될 때 해당하는 카테고리 또는 밴드에 대하여 원본 화소와 복원 화소 간 차이의 누적 값과 분류된 화소 개수의 누적 값을 말한다. 따라서 SAO의 화소 분류 작업은 화소를 분류함과 동시에 각각의 카테고리나 밴드에 분류된 복원 화소와 원본 화소 간 차이 값과 분류된 복원 화소의 개수를 누적하여 저장하여야 한다.

그림 5는 제안하는 SAO 하드웨어 구조의 화소 분류 구조를 나타낸다.

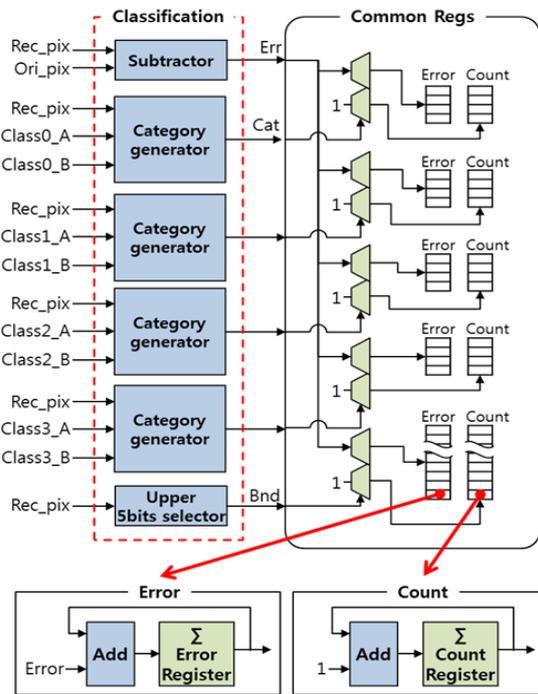


그림 5. 화소 분류 구조
Fig. 5 Sample classification architecture

제안하는 SAO 하드웨어의 화소 분류 구조는 그림 5와 같이 Classification 그룹에서 분류될 정보(Error, Category, Band)를 구하고, Common Regs 그룹에서 분류된 정보(Error, Count)를 해당하는 카테고리 또는 밴드 레지스터에 누적하여 저장한다. 그림 5의 Classification 그룹은 한 화소에 대한 분류 작업을 나타낸 것이며, 제안하는 SAO 하드웨어 구조는 4개의 화소를 병렬적으로 처리하므로 4개의 Classification 그룹이 존재한다. Common Regs 그룹은 분류된 정보(Error,

Count)를 누적하여 저장하기 위해 4개의 Classification 그룹이 공통으로 사용한다.

화소를 분류할 때, 밴드 오프셋의 경우 각 밴드 구간이 일정한 8의 크기를 갖기 때문에 복원 화소 값의 상위 5비트를 이용하여 쉽게 해당하는 밴드로 분류할 수 있다. 하지만 에지 오프셋의 경우 복원 화소를 각 패턴 클래스에 따른 2개의 주변 화소 정보와 표 1의 조건식에 따라 하위 카테고리 분류해야 한다.

에지 오프셋의 카테고리 결정 모듈을 하드웨어로 구현 시 표 1의 조건식들을 조건문(if statement)과 비교 연산자들(>, <, ==)을 사용하여 구현 할 수 있다. 하지만 표 1과 같이 많은 비교 연산자들을 사용하는 것과 각각의 조건문에서 비교 연산자들을 사용하는 것은 불필요한 로직 게이트의 발생을 야기하기 때문에 하드웨어 면적을 증가시킬 수 있다. 따라서 불필요한 로직 게이트의 발생을 방지하고 하드웨어 면적을 감소시키기 위해 그림 6과 표 2에 나타낸 바와 같이 플래그 정보를 생성하고, 생성된 플래그 정보를 이용하여 카테고리를 결정할 수 있다. 그림 6은 Verilog HDL로 작성한 코드이며, 표 1에서 12개의 비교 연산자를 사용하는 것과 비교하여 4개의 비교 연산자만을 사용하여 플래그(LessA, EquA, LessB, EquB) 정보를 생성한다. 생성된 플래그 정보를 기반으로 표 2와 같이 카테고리를 결정할 수 있으며, 이러한 방법은 불필요한 로직 게이트의 발생을 방지하고 보다 적은 수의 비교 연산자를 사용하기 때문에 하드웨어 면적을 감소시킬 수 있다.

```
assign {LessA, EquA} = (C < A) ? 2'b10 : (C == A) ? 2'b01 : 2'b00;
assign {LessB, EquB} = (C < B) ? 2'b10 : (C == B) ? 2'b01 : 2'b00;
```

그림 6. 에지 오프셋의 카테고리 결정을 위한 플래그 생성
Fig. 6 Flag creation for category decision in Edge Offset

표 2. 플래그 정보에 따른 카테고리 결정
Table. 2 Category decision using flag information

LessA	EquA	LessB	EquB	Category
1	0	1	0	1
1	0	0	1	2
0	1	1	0	
0	0	0	1	3
0	1	0	0	
0	0	0	0	4
None of the above				0

3.3. 최적 오프셋 결정 구조

SAO는 영역 내 화소들의 분류가 완료되면 분류된 정보를 기반으로 에지 오프셋과 밴드 오프셋에 대한 최적 오프셋 값들을 결정하고, 현재 영역에 대하여 최적 SAO 파라미터를 결정한다.

최적 오프셋 및 SAO 파라미터를 결정할 때, SAO는 각각의 울-왜곡 비용을 계산하여 최소의 울-왜곡 비용을 갖는 오프셋 및 파라미터를 결정한다[8].

HEVC의 참조 소프트웨어인 HM-11은 화소가 8비트로 표현되는 영상에서 최적 오프셋 값을 결정할 때 오프셋 값이 0보다 작거나 7보다 클 수 없도록 임계값을 설정하고 있다. 또한 울-왜곡 비용을 계산하기 위해 필요한 변수인 R(Rate)을 에지 오프셋과 밴드 오프셋에서 각각 (Offset+1)과 (Offset+2)로 정의하고 있다[9]. 표 3은 오프셋 0부터 7까지의 울-왜곡 비용을 계산하기 위한 식을 나타낸다.

표 3. 오프셋에 따른 울-왜곡 비용 계산

Table. 3 Rate-Distortion cost calculation for each offset

RDcost = Nh ² - 2Eh + λR		
Offset(h)	RDcost	R
0	0	-
1	(N*1) - (E*2) + (λ*R)	EO : 2 BO : 3
2	(N*4) - (E*4) + (λ*R)	EO : 3 BO : 4
3	(N*9) - (E*6) + (λ*R)	EO : 4 BO : 5
4	(N*16) - (E*8) + (λ*R)	EO : 5 BO : 6
5	(N*25) - (E*10) + (λ*R)	EO : 6 BO : 7
6	(N*36) - (E*12) + (λ*R)	EO : 7 BO : 8
7	(N*49) - (E*14) + (λ*R)	EO : 8 BO : 9

표 3의 RDcost는 울-왜곡 비용을 나타내며, N은 분류된 누적 개수, E는 원본 화소와 복원 화소 간 차이의 누적 값, h는 오프셋 값, λ는 울-왜곡 비용을 계산하기 위한 라그랑지 승수, R은 오프셋을 표현하기 위한 비트 레이트를 나타낸다.

각 오프셋에 따른 울-왜곡 비용을 계산하기 위한 모듈을 하드웨어로 구현 시 표 3과 같이 많은 수의 곱셈기와 덧셈기, 뺄셈기가 필요하다. 하지만 오프셋(h)과 오

프셋에 따른 비트레이트(R)를 상수로 표현할 수 있기 때문에 그림 7과 같이 쉬프트기와 덧셈기만을 사용하여 공통 연산기를 구현할 수 있다.

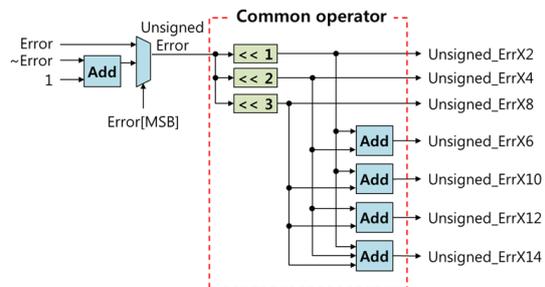


그림 7. 울-왜곡 비용 계산을 위한 공통 연산기
Fig. 7 Common operator for Rate-Distortion cost calculation

그림 7은 공통 연산기를 사용하여 울-왜곡 비용 계산식 중 (2Eh) 부분을 구현한 것이다. 그림 7의 공통 연산기는 원본 화소와 복원 화소 간 차이의 누적 값(Error)을 입력받아 오프셋 0부터 7까지에 대한 (2Eh) 값을 한 번에 생성한다. 울-왜곡 비용 계산식 중 나머지 (Nh²)와 (λR)도 그림 7과 같은 공통 연산기 구조로 구현하였으며, 이러한 구조는 연산기의 사용을 최소화하여 하드웨어 면적을 감소시킬 수 있다.

3.4. 2단계 파이프라인 구조

SAO는 크게 3단계로 화소 분류, 최적 SAO 파라미터 결정, 복원 영상 보상 단계로 나눌 수 있다. SAO는 화소 분류가 끝나면 최적 SAO 파라미터를 결정하고, 결정된 SAO 파라미터 정보를 기반으로 복원 영상을 보상한다. 복원 영상 보상 단계의 경우 결정된 SAO 파라미터 정보만 입력되면 화소 분류와 최적 SAO 파라미터 결정 단계와 독립적으로 수행될 수 있다. 따라서 그림 8과 같은 2단계 파이프라인 구조의 적용이 가능하다.

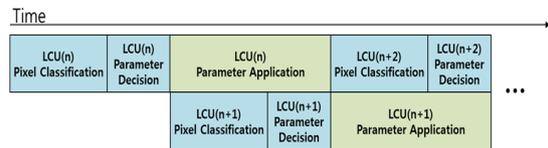


그림 8. SAO의 2단계 파이프라인 구조
Fig. 8 2-step pipeline architecture of SAO

그림 8의 2단계 파이프라인 구조는 현재 LCU 내 화소들을 분류한 후 최적 SAO 파라미터를 결정한다. 그리고 결정된 SAO 파라미터 정보를 기반으로 현재 LCU를 보상합과 동시에 다음 LCU의 화소 분류와 최적 SAO 파라미터 결정을 수행한다.

2단계 파이프라인 구조는 처음 LCU를 처리할 때만 모든 소요 사이클이 필요하지만, 두 번째 LCU부터는 소요 사이클이 1/2로 줄기 때문에 연산시간을 감소시킬 수 있다.

3.5. 단일 SAO 구조

SAO는 휘도 성분과 색차 성분에 대하여 독립적으로 수행되며 비슷한 수행 구조를 갖는다. 따라서 휘도 성분을 위한 SAO 구조는 색차 성분을 위해 재사용될 수 있고, 이러한 단일 구조는 별도의 색차 성분을 위한 SAO 구조가 필요하지 않기 때문에 하드웨어 면적을 크게 감소시킬 수 있다.

IV. 하드웨어 구현 결과 및 성능 비교

본 논문에서 제안하는 SAO 하드웨어 구조는 Verilog HDL로 설계하였으며, Synopsys사의 Design Compiler와 TSMC 0.13 μ m CMOS 표준 셀 라이브러리를 사용하여 합성한 결과 약 190k개의 게이트로 구현되었다. 또한 제안하는 SAO 하드웨어 구조는 Mentor Graphics사의 Modelsim SE 10.1c 시뮬레이터를 사용하여 정상적으로 동작함을 확인하였고, 그림 9는 시뮬레이션 결과 파형을 나타낸다.

그림 9의 입/출력 신호들은 제안하는 SAO 하드웨어 구조의 입/출력 신호들 중 일부분을 나타내며, 입력 신호로는 시스템 클럭과 리셋, 4개의 입력 화소와 주변 화소 정보가 있고, 출력 신호로는 SAO 파라미터 정보와 SAO가 수행된 4개의 출력 화소가 있다.

그림 9에서 SAO 파라미터 정보를 나타내는 신호들은 신호명 앞에 "aply_SAO_Y_" 문자열을 포함하는 신호들이며, 예지 오프셋의 수평 클래스와 4개의 카테고리에 따른 오프셋 값 (2, 1, 0, -2)를 나타내고 있다. 그림 9에서 표시된 입력 화소 Rec_pix1의 경우 화소 값이 124이며, 수평 클래스에 따른 입력 주변 화소 L_pix1과 R_pix1는 각각 116, 118이다.

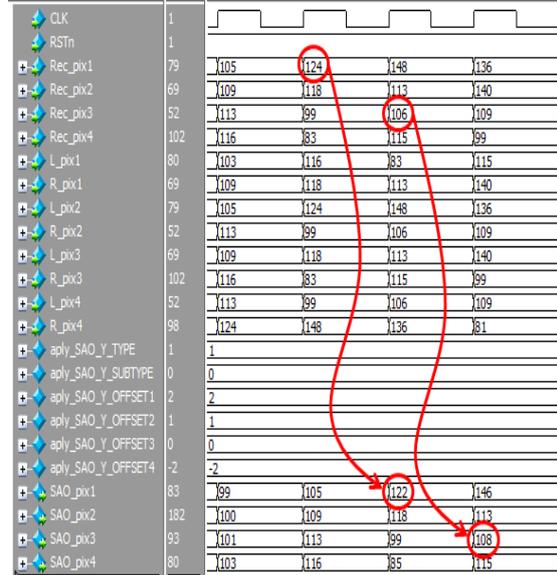


그림 9. 시뮬레이션 파형
Fig. 9 Simulation wave form

Rec_pix1의 값이 주변 화소 값인 L_pix1과 R_pix1보다 크기 때문에 표 1의 화소 분류 조건식에 따라 카테고리 4에 속하게 되고, 카테고리4에 해당하는 오프셋인 (-2)가 적용되어 화소 값 122가 SAO_pix1 신호를 통해 출력된다.

제안하는 SAO 하드웨어 구조는 사이클 당 4개 화소를 처리하므로 휘도 블록(64x64)과 2개의 색차 블록(각 32x32)을 모두 포함하는 하나의 LCU를 처리하는데 약 1600 사이클을 소요한다. 따라서 200MHz의 동작 주파수에서 초당 122,400개의 LCU를 갖는 4K UHD@60fps 영상을 실시간으로 처리할 수 있다.

표 4. 기존 구조와 제안하는 구조의 비교 결과
Table. 4 Compared result between existing and proposed architecture

Category	Mody[10]	Yang[11]	Proposed
Process	28nm	0.13 μ m	0.13 μ m
Gate Count	300k	218k	190k
Operating Frequency	200MHz for 4K UHD at 60fps	150MHz for FHD at 30fps	200MHz for 4K UHD at 60fps
Maximum Frequency	266MHz	-	250MHz

표 4는 기존의 HEVC 부호기측 SAO 하드웨어 구조와 제안하는 SAO 하드웨어 구조를 비교한 결과를 나타낸다. 제안하는 구조는 표 4에서 나타낸 바와 같이 Mody[10]의 구조와 비교하여 동일한 동작 주파수와 처리 해상도를 갖지만, 하드웨어 면적을 나타내는 게이트 수가 총 30만 게이트에서 약 37%만큼 감소한 19만 게이트로 구현된다. 또한 Yang[11]의 구조와 비교하여 동작 주파수와 처리 해상도가 높을 뿐만 아니라 게이트 수가 총 21.8만 게이트에서 약 13%만큼 감소하였다.

V. 결 론

본 논문에서는 HEVC 부호기를 위한 효율적인 SAO의 저면적 하드웨어 구조를 제안하였다. 제안하는 SAO 하드웨어 구조는 화소 단위로 연산을 수행하는 표준 SAO의 연산시간을 감소시키기 위해 4개의 화소를 병렬적으로 처리하며, 2단계 파이프라인 구조로 구현하였다. 또한 하드웨어 면적을 최소화하기 위해 휘도 성분과 색차 성분에 대하여 단일 SAO 구조로 구현하였고, 하드웨어에 적합한 연산기 및 공통 연산기를 사용하였다.

제안하는 SAO 하드웨어 구조는 TSMC 0.13 μ m CMOS 표준 셀 라이브러리로 합성한 결과 190k개의 게이트로 구현되며, 시뮬레이션 검증을 통하여 정상 동작함을 확인하였다. 제안하는 SAO 하드웨어 구조는 200MHz의 동작 주파수에서 4K UHD@60fps 영상을 실시간으로 처리할 수 있고, 최대 250MHz까지 동작 가능하다. 또한 기존 SAO 하드웨어 구조와 비교하여 하드웨어 면적을 나타내는 게이트 수가 각각 37%와 13%만큼 감소하였다.

감사의 글

본 연구는 교육부와 한국연구재단의 지역혁신인력양성사업(NRF-2012H1B8A2025862)과 미래창조과학부 및 정보통신산업진흥원의 해외인재스카우팅사업(NIPA-HB616-13-1001)의 지원으로 수행되었습니다.

REFERENCES

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [2] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M.-S. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J.-H. Park, "Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709-1720, Dec. 2010.
- [3] C.-M. Fu, E. Alshina, A. Alshin, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, S.-M. Lei, J.-H. Park, and W.-J. Han, "Sample Adaptive Offset in the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755-1764, Dec. 2012.
- [4] J.-S. Choi, and K.-K. Ryoo, "Area Efficient Hardware Design for Performance Improvement of SAO," *J. Korea Inst. Inf. Commun. Eng.*, vol. 17, no. 2, pp. 391-396, Feb. 2013.
- [5] "High-efficiency video coding text specification draft 10," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG, JCTVC-L1003, Jan. 2013.
- [6] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, and W.-J. Han, High Efficiency Video Coding (HEVC) Test Model 11 (HM11) Encoder Description, document JCTVC-M1002, Apr. 2013.
- [7] C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, and S. Lei, CE13: Sample Adaptive Offset with LCU-Independent Decoding, document JCTVC-E049, Mar. 2011.
- [8] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669-1684, Dec. 2012.
- [9] HEVC reference software : HM 11. Available: <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags/HM-11.0>
- [10] M. Mody, H. Garud, S. Nagori, and D. K. Mandal, "High Throughput VLSI Architecture for HEVC SAO Encoding for Ultra HDTV," *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, vol., no., pp. 2620-2623, Jun. 2014.

- [11] S. G. Yang, "Efficient sample adaptive offset algorithm and hardware design for HEVC," M.S. theses, Yonsei University, Seoul, 2012.



조현표(Hyunpyo Cho)

2013년 한밭대학교 공과대학 정보통신공학과 공학사
2013년 ~ 현재 한밭대학교 정보통신전문대학원 정보통신공학과 석사과정
※관심분야 : SoC 플랫폼 설계 및 검증, 멀티미디어 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 공과대학 정보통신공학과 교수
※관심분야 : 공학교육, SoC 플랫폼 설계 및 검증, 멀티미디어 코덱 설계