HEVC 복호기의 연산 복잡도 감소를 위한 화면내 예측 하드웨어 구조 설계

정홍균* · 류광기**

An Intra Prediction Hardware Architecture Design for Computational Complexity Reduction of HEVC Decoder

Hongkyun Jung* · 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 HEVC 복호기내 화면내 예측의 연산 복잡도를 감소시키기 위해 공유 연산기, 공통 연산기, 고속 smoothing 결정 알고리즘, 고속 필터계수 생성 알고리즘을 적용한 하드웨어 구조를 제안한다. 공유 연산기는 공통 수식을 공유하여 smoothing 과정의 연산 중복성을 제거하고, DC모드의 평균값을 미리 계산하여 수행 사이클 수를 감소시킨다. 공통 연산기는 모든 예측모드의 예측픽셀 생성과 필터링 과정을 하나의 연산기로 처리하기 때문에 연산기의 개수를 감소시킨다. 고속 smoothing 결정 알고리즘은 비트 비교기만을 사용하고, 고속 필터계수 생성 알고리즘은 곱셈연산 대신 LUT를 사용하여 연산 개수, 하드웨어 면적과 처리 시간을 감소시킨다. 또한 제안하는 구조는 2개의 공유 연산기와 8개의 공통 연산기를 사용하여 병렬처리함으로써 화면내 예측의 수행 사이클 수를 감소시킨다. 제안하는 구조를 TSMC 0.13um CMOS 공정 라이브러리를 이용하여 합성한 결과 게이트 수는 40.5k, 최대 동작 주파수는 164MHz이다. HEVC 참조 소프트웨어 HM 7.1에서 추출한 데이터를 이용하여 성능을 측정한 결과 제안하는 구조의 수행 사이클 수가 기존 구조 대비 93.7% 감소하였다.

ABSTRACT

In this paper, an intra prediction hardware architecture is proposed to reduce computational complexity of intra prediction in HEVC decoder. The architecture uses shared operation units and common operation units and adopts a fast smoothing decision algorithm and a fast algorithm to generate coefficients of a filter. The shared operation unit shares adders processing common equations to remove the computational redundancy. The unit computes an average value in DC mode for reducing the number of execution cycles in DC mode. In order to reduce operation units, the common operation unit uses one operation unit generating predicted pixels and filtered pixels in all prediction modes. In order to reduce processing time and operators, the decision algorithm uses only bit-comparators and the fast algorithm uses LUT instead of multiplication operators. The proposed architecture using four shared operation units and eight common operation units which can reduce execution cycles of intra prediction. The architecture is synthesized using TSMC 0.13um CMOS technology. The gate count and the maximum operating frequency are 40.5k and 164MHz, respectively. As the result of measuring the performance of the proposed architecture using the extracted data from HM 7.1, the execution cycle of the architecture is about 93.7% less than the previous design.

키워드

HEVC, 화면내 예측, 공유 연산기, 공통 연산기, 고속 알고리즘

Key word

HEVC, Intra prediction, shared operation unit, common operation unit, fast algorithm

* 준회원 : 한밭대학교 정보통신공학과

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

접수일자 : 2013. 01. 11

심사완료일자 : 2013. 01. 28

Open Access http://dx.doi.org/10.6109/jkiice.2013.17.5.1203

Ⅰ. 서 론

HEVC(High Efficiency Video Coding)은 JCT-VC(Joint Collaborative Team on Video Coding)에서 개발 중인 새로운 국제 비디오 부호화 표준이다. HEVC는 기존 비디오 부호화 표준인 H.264/AVC 대비 2배의 압축 성능과 저복잡도를 목표로 새로운 기술들을 채택하였다[1].

새로운 기술들 중 화면내 예측은 현재 프레임과 가장 유사한 예측 프레임을 생성하기 위해 4x4 PU(Prediction Unit)부터 64x64 PU까지 다양한 크기의 PU에 대한 예측을 수행하고, 35개의 예측모드를 갖는다. 또한 참조픽셀 smoothing 과정과 예측픽셀의 필터링 과정을 채택하여 예측 오차를 감소시킨다. HEVC의 화면내 예측은 많은 예측모드 수와 다양한 크기의 PU를 지원하고 참조픽셀 smoothing 과정과 예측픽셀의 필터링 과정을 수행하기 때문에 하드웨어 구현시 기존 H.264/ AVC에 비해 연산량과 연산 복잡도가 많이 증가한다[1-5].

Li[6]는 유연한 참조픽셀 선택 기술과 DC 모드와 방향성 모드를 처리하는 4x4 화면내 예측 하드웨어 구조를 제안했다. Li[6]의 구조는 17개의 레지스터들과 유연한 참조픽셀 선택 기술을 사용하여 메모리 자원과 processing latency를 감소시켰고, 모든 방향성 모드의 예측픽셀을 하나의 연산기로 처리한다. Li[6]의 구조는 하나의 예측픽셀을 생성하는 데 24 사이클이 소요되고, 4x4 PU만을 처리하고, DC 모드와 방향성 예측모드에 대한 연산기가 각각 존재하고, working draft 1[7]에 기초한 구조이기 때문에 Planar 모드, 참조픽셀 smoothing 과정과 예측픽셀 필터링 과정을 지원하지 않는 단점이 있다.

따라서 본 논문에서는 4x4 PU의 수행 사이클 수를 감소시키고, 화면내 예측의 연산량 및 연산 개수를 감소시키는 화면내 예측 하드웨어 구조를 제안한다. 제안하는 구조는 smoothing 계산 수식의 공통 수식을 공유하는 연산 구조와 DC 모드의 평균값을 계산하는 구조를 적용한 SSDAOU (Smoothing Shared/ DC Average Operation Unit)를 사용하여 연산량, 연산 개수, DC 모드의 수행 사이클 수를 감소시키고, PGCOU(Pixel Generation Common Operation Unit)를 사용하여 모든 예측모드의 예측픽셀 생성과 예측픽셀 필터링 과정을 하나의 연산기만으로 처리하여 연산기의 개수를 감소시킨다. 또한 산술 연산기 대신 비트 비교기만을 사용한 고속 smoothing 결정 알고리즘, LUT 기반 고속 필터 계수 결정방법을 적용하여 하드웨어 면적 및 계산 지연 시간을 감소시킨다.

본 논문의 구성은 다음과 같다. 2장에서는 HEVC의 화면내 예측에 대해 기술하고, 3장에서는 제안하는 화면내 예측 하드웨어 구조에 대해 기술한다. 4장에서는 제안하는 구조의 검증 및 성능 비교에 대해 기술하고, 5장에서는 본 연구의 결론을 도출한다.

Ⅱ. HEVC의 화면내 예측

HEVC의 화면내 예측은 4x4 PU부터 64x64 PU까지 다양한 크기의 PU를 처리하고, 그림 1과 같이 각 PU별로 35개의 예측모드를 갖는다. 모드 0은 Planar 모드, 모드 1은 DC 모드, 모드 2에서 34까지는 Angular 모드이다. 모드 35는 chroma 모드에서만 사용하는 IntraFromLuma모드이다[2-4].

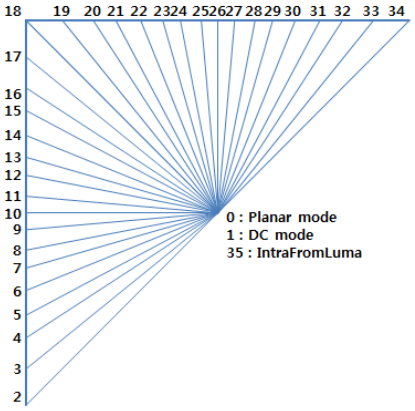

그림 1. 화면내 예측모드의 방향성
Fig. 1 Intra Prediction mode Directions

HEVC의 화면내 예측 복호 과정은 먼저 예측모드를 복호하고 참조픽셀 substitution 과정을 수행한 후 예측모드와 PU 크기에 따라 참조픽셀 smoothing 과정을 수행한다. 보간된 참조픽셀을 이용하여 예측모드에 대한 예측픽셀을 생성한 후 예측픽셀 필터링 과정을 수행한다[2-4].

2.1. 참조픽셀 smoothing 과정

참조픽셀 smoothing 과정은 예측 오차를 감소시키기 위해 현재 참조픽셀과 좌우 참조픽셀을 이용하여 평활화(smoothing)를 수행하는 과정으로써 PU 크기와 예측 모드에 의해 수행 여부가 결정된다. smoothing 과정의 수행 여부는 Draft 6[4]에 정의된 식 1과 표 1에 의해 결정된다. 식 1은 현재 예측모드와 horizontal 모드(10)의 차이와 현재 모드와 vertical 모드(26)의 차이 중 작은 값을 계산하는 식으로써 Mode는 현재 예측모드를, Min()은 최소값을 구하는 함수를, Abs()는 절대값을 구하는 함수를 의미한다. 식 1에서 계산된 결과 값과 표 1에 정의된 경계값을 비교하여 경계 값보다 큰 경우에만 참조픽셀의 smoothing 과정이 수행된다.

$$\min Dist = Min(Abs(Mode-26), Abs(Mode-10)) \quad (1)$$

표 1. PU 크기별 필터링 경계 값
Table. 1 Filtering Threshold Values for PU Size

PU 크기	4x4	8x8	16x16	32x32	64x64
경계 값	10	7	1	0	10

식 2는 현재 PU의 위쪽에 위치한 참조픽셀들의 평활화를 위해 3탭 필터를 적용시킨 수식이다. pF는 평활화된 참조픽셀, p는 평활화되지 않은 참조픽셀, y는 참조픽셀의 열 위치를 의미하고, 현재 참조픽셀의 계수는 1/2이고, 좌우 픽셀의 계수는 1/4이다.

$$pF[-1,y] = (p[-1,y+1] + 2*p[-1,y] + p[-1,y-1] + 2) \gg 2 \text{ for } y = nS*2-2..0 \quad (2)$$

2.2. 예측픽셀 생성 및 필터링 수식

화면내 예측은 각 예측모드에 해당하는 예측픽셀 생성 수식을 적용하여 예측픽셀을 생성하고, DC 모드에서는 PU의 첫 번째 행과 첫 번째 열에, Horizontal 모드에서는 첫 번째 행에, Vertical 모드에서는 첫 번째 열에 보간 필터를 적용하여 예측픽셀을 필터링한다. Draft 6[4]에 정의된 식 3부터 식 15까지는 예측모드에 해당하는 예측픽셀 생성 수식과 필터링 수식으로써, DCVal은 참조픽셀들의 평균값을, p는 참조픽셀을, pS는 예측픽셀을, nS는 PU의 크기를, x는 행의 위치를, y는 열의 위치를 의미한다. 식 3은 DC 모드에서 참조픽셀들의 평균값을 계산하는 수식이고, 식 4부터 7까지는 DC 모드에서 예측픽셀을 생성하는 수식이다. DC 모드에서 첫 번째 행의 예측픽셀 값들은 식 4부터 6까지와 같이 DCVal과 현재 PU의 좌측과 상단에 위치한 참조픽셀에 보간 필터를 적용하여 생성되고 나머지 예측픽셀 값들은 식 7과 같이 현재 PU의 참조픽셀들의 평균을 계산한 결과값이 된다.

$$DCVal = \left(\sum_{x'=0}^{nS-1} p[x',-1] + \sum_{y'}^{nS-1} p[-1,y'] + nS \right) \gg (k+1) \quad (3)$$
$$, with\ k = Log_2(nS)$$

$$pS[0,0] = (1*p[-1,0] + 2*DCVal + 1*p[0,-1] + 2) \gg 2 \quad (4)$$

$$pS[x,0] = (1*p[x,-1] + 3*DCVal + 2) \gg 2 \quad (5)$$
$$, with\ x = 1..nS-1$$

$$pS[0,y] = (1*p[-1,y] + 3*DCVal + 2) \gg 2 \quad (6)$$
$$, with\ y = 1..nS-1$$

$$pS[x,y] = DCVal, with\ x,y = 1..nS-1 \quad (7)$$

식 8은 Planar 모드에서 예측픽셀을 생성하는 수식으로 예측픽셀의 행에 위치한 참조픽셀, 열에 위치한 참조픽셀, 현재 PU의 좌측 하단 가장자리에 위치한 참조픽셀의 아래에 위치한 픽셀, 현재 PU의 상단 우측 가장자리에 위치한 참조픽셀의 우측에 위치한 픽셀에 보간 필터를 적용하여 예측픽셀을 생성한다. Horizontal 모드에서는 첫 번째 행을 제외한 각 행의 예측픽셀 값은 식 9와 같이 현재 PU의 좌측에 위치한 각 행의 참조픽셀 값이 되고 첫 번째 행의 예측픽셀 값은 식 10과 같이 첫 번째 행의 참조픽셀과 현재 PU의 상단에 위치한 참조픽셀들에 보간 필터를 적용하여 생성된다.

$$pS[x,y] = ((nS-1-x)*p[-1,y] + (x+1)*p[nS,-1] + (nS-1-y)*p[x,-1] + (y+1)*p[-1,nS] + nS) \gg (k+1) \quad (8)$$
$$with\ x,y = 0..nS-1\ where\ k = Log_2(nS)$$

$$pS[x,y] = p[-1,y], with\ x = 0..nS-1, y = 1..nS-1 \quad (9)$$

$$pS[x,y] = Clip1(p[-1,y] + ((p[x,-1] - p[-1,-1]) \gg 1)) \text{ for } x = 0..nS-1, y = 0 \quad (10)$$

Vertical 모드에서는 식 11과 식 12와 같이 첫 번째 열을 제외한 각 열의 예측픽셀 값이 현재 PU의 상단에 위치한 각 열의 참조픽셀 값이 되고 첫 번째 열의 예측픽셀 값은 첫 번째 열의 참조픽셀과 현재 PU의 좌측에 위치한 참조픽셀들에 보간 필터를 적용하여 생성된다.

$$pS[x,y] = p[x,-1], with\, x = 1..nS-1, y = 0..nS-1 \quad (11)$$

$$pS[x,y] = Clip1(p[x,-1] + ((p[-1,y] - p[-1,-1]) \gg 1)) \text{ for } x = 0, y = 0..nS-1 \quad (12)$$

식 13부터 15까지는 Horizontal과 Vertical 모드를 제외한 Angular 모드에서 예측픽셀을 생성하기 위해 사용하는 수식이다. 식 13은 필터계수인 iFact를 생성하는 수식으로써 예측모드의 방향성을 나타내는 intraPredAngle과 y에 따라 iFact가 결정된다. 식 14는 참조픽셀의 위치를 나타내는 iIdx를 생성하는 수식으로써 식 13과 마찬가지로 inttraPredAngle에 따라 iIdx가 결정된다. 식 15는 예측픽셀을 생성하는 수식으로써, 예측모드의 방향성에 의해 선택된 2개의 참조픽셀에 보간 필터를 적용하여 예측픽셀을 생성한다.

$$iFact = ((y+1)*intraPredAngle)\&\&31 \quad (13)$$

$$iIdx = ((y+1)*intraPredAngle) \gg 5 \quad (14)$$

$$pS[x,y] = ((32 - iFact)*(r[x+iIdx+1] + iFact*r[x+iIdx+2] + 16) \gg 5 \quad (15)$$

Ⅲ. 제안하는 화면내 예측 하드웨어 구조

제안하는 화면내 예측 하드웨어 구조는 연산량 및 연산 개수를 감소시키기 위해 비트 비교연산만을 사용한 고속 smoothing 결정 알고리즘, 참조픽셀의 공통 수식을 공유하고 DC 모드의 평균값을 계산하는 SSDAOU, LUT 기반 고속 필터계수 생성 알고리즘, 모든 예측픽셀을 생성하는 PGCOU를 사용한다. 그림 2는 제안하는 화면내 예측 구조를 나타낸다.

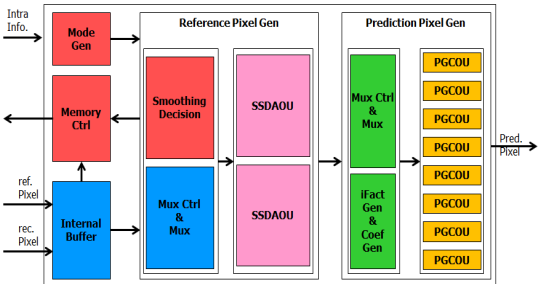

그림 2. 제안하는 하드웨어 구조
Fig. 2 Proposed Hardware Architecture

제안하는 구조는 참조픽셀이 저장되어 있는 외부메모리를 제어하는 메모리 제어기, 현재 PU의 예측모드를 생성하는 모드 생성기, 참조픽셀의 smoothing 과정을 수행하는 참조픽셀 생성기, 복호된 예측모드에 따라 예측픽셀을 생성하는 예측픽셀 생성기로 구성된다. 참조픽셀 생성기는 고속 smoothing 결정기, 예측모드와 PU 크기에 따라 참조픽셀을 선택하는 멀티플렉서, 제어기, 2개의 SSDAOU로 구성되고, 예측픽셀 생성기는 고속 필터계수생성기, PU 크기와 예측모드에 따라 참조픽셀과 필터계수를 선택하는 멀티플렉서, 8개의 PGCOU로 구성된다.

3.1. 고속 smoothing 결정 알고리즘

제안하는 고속 smoothing 결정 알고리즘은 식 1과 표 1을 이용한 smoothing 결정 알고리즘 대신 산술 연산 없이 예측모드의 비트 값 비교만으로 smoothing 과정을 결정하는 고속 결정 알고리즘이다. 표 2는 각 PU별로 예측모드에 smoothing을 결정하는 식 1과 표 1을 적용한 결과를 나타낸다. 8x8 PU에서 smoothing을 수행하는 예측모드인 0, 2, 18, 34를 6비트로 표현하면 각각 000000, 000010, 010010, 100010이다. 예측모드 2, 18, 34는 하위 4비트는 0010이고, 6비트 110010은 50으로써 예측 모드의 범위에 해당하지 않는 무관조건(don't care)이기 때문에 하위 4비트 0010으로 표현할 수 있다. 따라서 8x8 PU의 경우 예측모드의 6비트가 000000이거나 하위 4비트가 0010일 때 smoothing을 수행한다.

표 2. 예측모드에 대한 smoothing 결정
Table. 2 Smoothing Decision for Each Prediction mode

PU	Smoothing 수행여부	
	수행	미수행
4x4	-	0-34
8x8	0, 2, 18, 34	1, 3-17, 19-33
16x16	0, 2-8, 12-24, 28-34	9-11, 25-27
32x32	0, 2-9, 11-25, 27-34	1, 10, 26
64x64	-	0-34

위와 같은 방법을 적용하여 표 2를 정리하면 표 3과 같이 뺄셈연산과 산술 비교연산 대신 비트 비교연산만으로 smoothing 과정의 수행 여부를 결정할 수 있다. 32x32 PU인 경우 예측모드 6비트가 000001 이거나 하위 4비트가 1010 인 경우에만 smoothing을 수행하지 않는다. 제안하는 결정 알고리즘은 각 비트 값을 비교하는 간단한 조합논리회로로 구현되어 하드웨어 면적과 결정 지연시간을 감소시킨다.

표 3. 제안하는 smoothing 결정 알고리즘
Table. 3 Proposed Smoothing Decision Algorithm

PU	예측모드에 따른 smoothing 실행여부
4x4 64x64	모든 예측모드에서 미실행
8x8	“000000” 또는 하위 4비트가 “0010”인 경우 실행
16x16	“000001” 또는 하위 4비트가 “1001” 또는 “1010” 또는 “1011”인 경우 미실행
32x32	“000001” 또는 하위 4비트가 “1010”인 경우 미실행

3.2. SSDAOU

제안하는 SSDAOU는 smoothing 과정의 연산량 및 연산 개수를 감소시키고 DC 모드에서 발생하는 유휴사이클을 제거하기 위해 DC 모드의 평균값을 계산하고 smoothing 연산에서 발생하는 공통수식을 공유하는 연산기이다. 식 16과 식 17은 이웃한 2개 참조픽셀에 대해 식 2를 적용한 smoothing 수식의 예를 나타낸다.

$$pF1' = (p0 + p1 + 1 + p1 + p2 + 1) \gg 2 \quad (16)$$

$$pF2' = (p1 + p2 + 1 + p2 + p3 + 1) \gg 2 \quad (17)$$

식 16과 식 17에서 보는 바와 같이 인접한 2개의 참조픽셀 평활화 수식은 p1+p2+1 연산 수식을 공통으로 갖고 있기 때문에 평활화된 참조픽셀들을 각각 계산할 경우 공통 수식에 대한 연산이 중복되는 단점이 발생한다. DC 모드인 경우 PU크기에 상관없이 smoothing과정을 수행하지 않기 때문에 smoothing 단계에서 유휴 사이클이 발생하는 단점이 있다.

따라서 제안하는 SSDAOU는 smoothing 단계에서 DC 모드의 유휴 사이클에 DC 모드의 평균값을 미리 계산하여 DC 모드의 수행 사이클 수를 감소시키고, smoothing 연산에서 발생하는 공통 수식을 공유함으로써 연산의 중복성을 제거하여 연산량 및 연산 개수를 감소시킨다. SSDAOU의 구조는 그림 3과 같이 17개의 전사간기, 22개의 반가산기, 18개의 왼쪽 쉬프터, Line Connection으로 구성되고 P0부터 P16은 평활화된 참조픽셀을 의미하고, DC는 참조픽셀의 평균값을 의미한다.

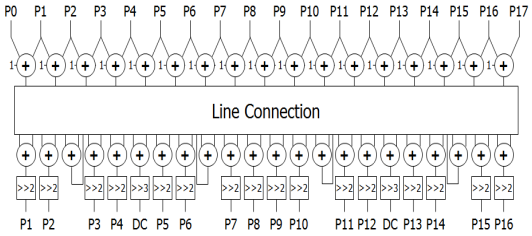

그림 3. SSDAOU 구조
Fig. 3 SSDAOU Architecture

Smoothing 연산 모드에서 SSDAOU의 Line Connection은 그림 4와 같이 17개의 전가산기와 16개의 반가산기를 연결하여 평활화된 참조픽셀을 생성한다. 점선으로 표시된 영역에서 각가의 전가산기는 공통 수식을 계산하는 연산기로써, 2개의 반가산기가 1개의 전가산기를 공유하여 덧셈 연산의 중복성을 제거한다. 그림 5는 PU 크기가 4x4인 경우 DC 평균값 모드에서 SSDAOU의 Line Connection을 나타낸다. 4x4 PU에서 DC 모드의 평균값은 8개의 참조픽셀을 이용하여 4개의 전가산기, 3개의 반가산기, 1개의 왼쪽 쉬프터로 계산한다.

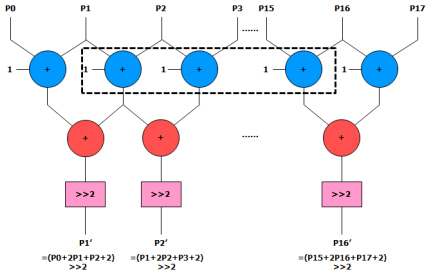

그림 4. Smoothing 모드에서의 SSDAOU 구조
Fig. 4 SSDAOU Architecture in Smoothing Mode

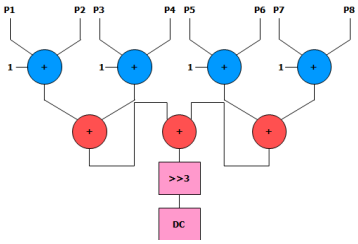

그림 5. DC 평균값 모드에서의 SSDAOU 구조
Fig. 5 SSDAOU Architecture in DC Average Mode

3.3. 간소화된 고속 필터계수 생성 알고리즘

제안하는 고속 필터계수 생성 알고리즘은 식 13을 적용한 곱셈연산 및 논리 AND 연산 대신 LUT만을 사용하여 고속으로 Angular 모드의 필터계수를 생성하는 알고리즘이다. Angular 모드에 해당하는 모든 예측모드와 y에 대해 식 13을 적용하여 계산한 iFact는 그림 6과 같다.

mode \ y	0	1	2	3	4	...	30	31
2	0	0	0	0	0	...	0	0
3	6	12	18	24	30	...	26	0
4	11	22	1	12	23	...	21	0
5	15	30	13	28	11	...	17	0
6	19	6	25	12	31	...	13	0
7	23	14	5	28	19	...	9	0
8	27	22	17	12	7	...	5	0
9	30	28	26	24	22	...	2	0
11	2	4	6	8	10	...	30	0
12	5	10	15	20	25	...	27	0
13	9	18	27	4	13	...	23	0
14	13	26	7	20	1	...	19	0
15	17	2	19	4	21	...	15	0
16	21	10	31	20	9	...	11	0
17	26	20	14	8	2	...	6	0
18	0	0	0	0	0	...	0	0

mode \ y	0	1	2	3	4	...	30	31
19	26	20	14	8	2	...	6	0
20	21	10	31	20	9	...	11	0
21	21	10	31	20	9	...	11	0
22	17	2	19	4	21	...	15	0
23	13	26	7	20	1	...	19	0
24	9	18	27	4	13	...	23	0
25	2	4	6	8	10	...	30	0
27	30	28	26	24	22	...	2	0
28	27	22	17	12	7	...	5	0
29	23	14	5	28	19	...	9	0
30	19	6	25	12	31	...	13	0
31	15	30	13	28	11	...	17	0
32	11	22	1	12	23	...	21	0
33	6	12	18	24	30	...	26	0
34	0	0	0	0	0	...	0	0

그림 6. Angular 모드에 대한 iFact
Fig. 6 iFact for Angular Modes

각 y에 대한 iFact는 y가 0인 경우의 iFact에 y+1를 곱한 후 하위 5비트를 취한 결과이다. 예를 들어 예측모드가 4인 경우 y가 0일 때 iFact는 11이고, y가 1일 때 iFact는 22이다. y가 2일 때는 y가 0인 경우의 iFact와 y+1의 곱의 결과인 33에서 하위 5비트를 취하므로 iFact는 1이 된다. 그림 6의 iFact를 분석하면 예측모드 18을 기준으로 예측 모드에 대한 iFact가 상하대칭임을 알 수 있다.

예측모드에서 2를 뺀 결과인 mode-2 모드를 그림 5에 적용하면 mode-2 모드 16을 기준으로 mode-2 모드 0부터 mode-2 모드 15까지의 iFact와 mode-2 모드 17부터 mode-2 모드 32까지의 iFact는 상하대칭이 되고, 32에서 17부터 32까지의 mode-2 모드를 뺀 결과는 15부터 0까지의 mode-2 모드가 되므로 그림 6은 mode-2 모드 0부터 16까지의 iFact로 간소화된다.

그림 6의 iFact에서 예측모드 2부터 예측모드 9까지의 iFact와 예측모드 18부터 예측모드 11까지의 iFact를 각각 더하면 32가 됨을 알 수 있다. 따라서 32에서 예측모드 2부터 예측모드 9까지의 iFact를 뺀 결과는 예측모드 18부터 예측모드 11까지의 iFact가 되므로 예측모드 2부터 9까지의 iFact로 간소화된다.

그림 6에 mode-2 모드와 간소화 기법을 적용하고 4x4 PU에 필요한 iFact를 정리하면 그림 6을 표 4와 같이 간단한 LUT로 표현할 수 있다. 표 4에서 offset은 8x8 PU, 16x16 PU, 32x32 PU에 필요한 iFact를 생성하는데 사용된다.

표 4. mode-2 모드에 대한 iFact와 offset
Table. 4 iFact and Offset for Mode-2 Modes

Mode-2	iFact				offset
	y=0	y=1	y=2	y=3	
0	0	0	0	0	0
1	6	12	18	24	24
2	11	22	1	12	12
3	15	30	13	28	28
4	19	6	25	12	12
5	23	14	5	28	28
6	27	22	17	12	12
7	30	28	26	24	24

예를 들어 8x8 PU를 처리하기 위해 필요한 y의 비트수는 3이다. y의 하위 2비트로 표 x에서 iFact를 선택하고

y의 최상위 비트로 offset의 덧셈 유무를 결정한다. 즉, 예측모드 3(mode-2 모드 1)인 경우 y가 4일 때 iFact는 y가 0인 경우의 iFact인 6에 offset인 24를 더하여 하위 5비트를 취한 30이 된다. 따라서 표 3의 LUT와 덧셈 연산기를 이용하여 4x4 PU, 8x8 PU, 16x16 PU, 32x32 PU에 필요한 iFcat를 생성할 수 있다. 또한 곱셈 연산 대신 간소화된 LUT를 이용하여 iFact를 생성하기 때문에 결정지연시간을 감소시킨다.

간소화된 표 3과 덧셈 연산을 적용한 고속 필터계수 생성 하드웨어 구조는 그림 7과 같다. 필터 계수 결정 하드웨어 구조는 iFactLUT, 4개의 덧셈기, 레지스터로 구성된다. iFactLUT는 표 3의 iFact를 저장하고 있으며, mode-2가 입력되면 mode-2 모드에 해당하는 4개의 iFact와 offset을 출력한다. 4x4 PU와 8x8 PU, 16x16 PU, 32x32 PU의 첫 번째 4x4 블록을 처리하는 경우 LUT에서 출력되는 4개의 iFact를 출력하고, 8x8 PU, 16x16 PU, 32x32 PU의 두 번째 4x4 블록부터는 레지스터에 저장된 4개의 iFact와 offset을 더하여 각각 y의 위치에 해당하는 4개의 iFact를 출력한다.

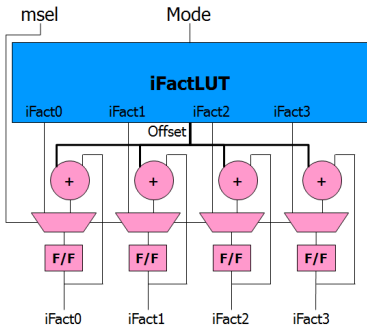

그림 7. 필터 계수 생성 하드웨어 구조
Fig. 7 Filter Coefficient Generation Hardware Architecture

3.4. PGCOU

제안하는 PGCOU는 연산 개수를 감소시키고 모든 예측모드의 예측픽셀을 하나의 연산기로 생성하기 위해 곱셈 연산이 감소된 Angular 모드와 Planar 모드의 예측픽셀 생성 수식과 Angluar 모드와 Planar 모드의 예측픽셀 생성 수식과 DC/ Horizontal/Vertical 모드의 예측픽셀 생성 수식을 적용한 공통 연산기이다. 기존 Planar 모드의 예측픽셀 생성 수식인 식 8에서 x+1과 y+1에 대한 곱셈연산을 정리하면 식 18과 같이 2개의 곱셈연산이 감소된 수식을 생성할 수 있고, 기존 Angular 모드의 예측픽셀 생성 수식인 식 15에서 iFact에 대한 곱셈연산을 정리하면 식 19와 같이 1개의 곱셈연산이 감소된 수식을 생성할 수 있다. 또한 식 18과 식 19는 곱셈연산 뿐만 아니라 감소된 곱셈연산의 계수를 생성하는 뺄셈연산도 감소시킨다. 기존 DC 모드 예측픽셀 생성 수식인 식 5, 식 6는 곱셈연산, 덧셈연산, 쉬프트연산을 사용하기 때문에 식 20과 식 21로 변경할 수 있다. 식 18부터 식 21까지의 뺄셈연산은 2의 보수를 적용하여 덧셈연산으로 계산할 수 있기 때문에 식 4, 식 10, 식 12, 식 18부터 식 21까지는 곱셈연산, 덧셈연산, 쉬프트연산을 처리하는 하나의 연산기로 구현할 수 있다.

$$\begin{aligned} pS[x,y] &= ((nS-1-x)*p[-1,y]+(x+1)*p[nS,-1]+ \\ &\quad (nS-1-y)*p[x,-1]+(y+1)*p[-1,nS] \\ &\quad +nS) \gg (k+1) \\ &= ((((x+1)*(p[nS,-1]-p[-1,y])+nS/2) \gg k) \\ &\quad +P[-1,y]+(((y+1)*(p[-1,nS]-p[x,-1])+ \\ &\quad nS/2) \gg k)+P[x,-1]+1) \gg 1 \end{aligned} \tag{18}$$

$$\begin{aligned} pS[x,y] &= ((32-iFact)*(r[x+iIdx+1] \\ &\quad +iFact*r[x+iIdx+2]+16) \gg 5 \\ &= (coef*(refMain[x+2]-refMain[x+1]) \\ &\quad +16) \gg 5)+refMain[x+1] \end{aligned} \tag{19}$$

$$\begin{aligned} pS[x,0] &= (1*p[x,-1]+3*DCVal+2) \gg 2 \\ &= ((1*(p[x,-1]-DCVal)+2) \gg 2)+DCVal \\ &, with\ x=1..nS-1 \end{aligned} \tag{20}$$

$$\begin{aligned} pS[0,y] &= (1*p[-1,y]+3*DCVal+2) \gg 2 \\ &= ((1*(p[-1,y]-DCVal)+2) \gg 2)+DCVal \\ &, with\ y=1..nS-1 \end{aligned} \tag{21}$$

식 18부터 식 21까지의 수식을 적용한 공통 연산기인 PGCOU는 그림 8과 같이 2개의 곱셈기, 7개의 덧셈기, 3개의 오른쪽 쉬프터로 구성된다. PGCOU는 Planar 모드에서 하나의 예측픽셀을 생성하고, Angular/DC/ Vertical/Horizontal 모드에서 2개의 예측픽셀을 생성한다. Planar 모드를 제외환 모드 예측모드에서는 Planar 픽셀을 생성하는데 필요한 1개의 전가산기와 1개의 오른쪽 쉬프트 연산기를 사용하지 않는다.

예측픽셀 생성기는 8개의 PGCOU를 사용하기 때문에 Angular/ DC/Horizontal/Vertical 모드에서 4x4 PU의 예측픽셀을 생성하는데 1 사이클이 소요되고, Planar 모드에서 4x4 PU의 예측픽셀을 생성하는데 2 사이클이 소요된다.

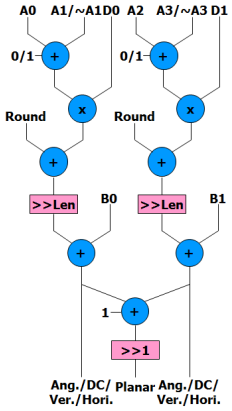

그림 8. PGCOU 구조
Fig. 8 PGCOU Architecture

3.5. 예측모드와 PU크기에 따른 처리 사이클

제안하는 구조는 화면내 예측을 수행하기 위해 참조픽셀 읽기 단계인 RPR(Ref. Pixel Read), 참조픽셀 생성 단계인 RPSDVG(Ref. Pixel Smoothing/DC Value Generation), 예측픽셀 생성단계인 PPG (Predicted Pixel Generation)로 수행된다. 예측모드 복호/참조픽셀 읽기 단계에서는 현재 PU의 예측모드를 복호하고 복호된 예측모드에 따라 필요한 참조픽셀들을 읽는다. 참조픽셀 생성단계에서는 참조픽셀의 평활화를 수행하거나 DC 모드의 평균값을 계산하고, 예측픽셀 생성단계에서는 예측모드에 따라 예측픽셀을 생성한다. 표 5는 PU크기에 따른 각 단계별 처리 사이클 수를 나타낸다.

표 5. PU 크기별 처리 사이클 수
Table. 5 Number of Execution Cycle for PU Size

PU	4x4	8x8	16x16	32x32
RPR	1	2	4	8
RPSDVG	1	1	1	2
PPG*	1/2	4/8	16/32	64/128
총합*	3/4	7/11	21/37	74/138

* : Planar 모드/나머지 모드

제안하는 구조는 외부 메모리로부터 사이클당 8개의 참조픽셀을 읽기 때문에 MGRPR 단계에서 4x4 PU, 8x8 PU, 16x16PU, 32x32 PU에 필요한 참조픽셀을 읽는데 각각 1 사이클, 2 사이클, 4 사이클, 8 사이클이 소요된다. RPSDVG단계에서는 2개의 SSDAOU를 사용하기 때문에 32x32 PU를 처리하는데 2 사이클이 소요되고 나머지 PU를 처리하는데 1 사이클이 소요된다. PPG 단계에서는 모드에 따라 처리 사이클 수가 다르다. Planar 모드인 경우 4x4 PU를 처리하는데 2 사이클이 소요되고, 32x32 PU를 처리하는 데 128 사이클이 소요된다. Planar 모드를 제외한 나머지 예측모드인 경우 4x4 PU를 처리하는데 1 사이클이 소요되고, 32x32 PU를 처리하는데 64 사이클이 소요된다. 따라서 제안하는 구조는 4x4 PU, 8x8 PU, 16x16 PU, 32x32 PU를 처리하는데 소요되는 총 사이클 수는 각각 4 사이클, 11 사이클, 37 사이클, 138 사이클이다.

Ⅳ. 실험 및 고찰

제안하는 하드웨어 구조는 Verilog HDL로 설계되었으며, TSMC 0.13um 표준 셀 라이브러리를 사용하여 IDEC에서 지원하는 CAD Tool인 Synopsys사의 Design Compiler로 합성하였다. 표 6은 제안하는 구조의 합성 결과를 나타낸다.

표 6. 화면내 예측 하드웨어의 합성 결과 및 비교
Table. 6 Synthesis Results and Comparison of Intra Prediction Hardware

	He[8]	Nadeem [9]	Li[6]	Proposed
압축표준	H.264 HP	H.264 BP	HEVC	HEVC
처리 PU(블록)	4x4-16x16	4x4, 16x16	4x4	4x4-32x32
참조픽셀 평활화 /예측픽셀 필터링	미지원	미지원	미지원	지원
공정(nm)	90	180	130	130
동작 주파수(MHz)	175	150	150	164
출력 Pixel 수	4	16	1	8/16*
예측픽셀 생성/ 출력 사이클 수**	4/16/ 64/-	1/-/ 22/-	16/-/ -/-	1/4/ 16/128
게이트 수	29.8k	21k	9k	40.5k

* : Planar 모드/ 나머지 모드, ** : 4x4/8x8/16x16/32x32

제안하는 구조의 게이트 수는 40.5k이고, 최대 동작 주파수는 164MHz이다. H.264/AVC high profile을 지원하는 He[8]의 구조, H.264/AVC baseline profile을 지원하는 Nadeem[9], Li[6]의 구조는 Planar 모드를 제외한 방향성 모드(Angular 모드), DC 모드를 지원하기 때문에 Planar 모드를 제외한 나머지 모드에서의 처리 사이클 수를 기준으로 제안하는 구조의 기존 구조들을 비교하였고, 처리 사이클 수는 예측픽셀 생성/출력 사이클 수를 의미한다. 4x4 PU(블록)을 기준으로 제안하는 구조의 처리 사이클 수는 1 사이클로써, Li[6] 대비 93.7% 감소하였고, 16x16 PU(블록)을 기준으로 제안하는 구조의 처리 사이클 수는 16 사이클로써, Nadeem[9] 대비 27.2% 감소하였다. 제안하는 구조는 4x4 PU부터 32x32 PU까지 처리하고, Planar 모드, 참조픽셀 평활화 과정, 예측픽셀 필터링 과정을 지원하고, 병렬 연산 구조를 사용하였기 때문에 작은 크기의 블록을 지원하고, Planar 모드, 참조픽셀 평활화 과정, 예측픽셀 필터링 과정을 지원하지 않는 Li[6], He[8], Nadeem[9]에 비해 게이트수가 증가하였다.

제안하는 구조의 동작을 검증하기 위해 HEVC 참조소프트웨어 HM 7.1을 이용하여 표 7과 같은 다양한 영상들을 부호화화여 비트스트림을 생성하였고, 생성된 비트스트림을 HM 7.1[10]로 복호하는 과정에서 화면내 예측 함수의 입출력 데이터를 추출하였다. 추출한 입력 데이터와 SDF 파일을 사용하여 IDEC에서 제공하는 Mentor Graphics사의 Modelsim SE 10.1c로 타이밍 시뮬레이션을 수행하여 제안하는 구조의 동작을 검증하였다. 검증한 결과 제안하는 구조의 출력 데이터와 HM 7.1에서 추출한 데이터와 일치함을 확인하였다.

표 7. 테스트 영상
Table. 7 Test Sequences

영상	해상도	fps	frame 수
BQSquare	416x240	60	600
BasketballDrillText	832x480	50	500
FourPeople	1280x720	60	600
BQTerrace	1920x1080	60	600
Traffic	2560x1600	30	150

V. 결 론

본 논문에서는 HEVC 복호기내 화면내 예측 하드웨어의 수행 사이클 수를 감소시키고, 연산량 및 연산 개수를 감소시키기 위해 공유 연산기, 공통 연산기, 고속 smoothing 결정 알고리즘, 고속 필터계수 생성 알고리즘을 적용한 화면내 예측 하드웨어 구조를 제안한다. 공유 연산기인 SSDAOU는 smoothing 계산 수식의 공통 수식을 공유하여 연산하고, DC모드의 평균값을 계산함으로써 덧셈 연산기의 개수를 감소시키고, DC모드의 수행 사이클 수를 감소시켰다. 공통 연산기인 PGCOU는 모든 예측모드의 예측픽셀 생성 연산과 예측픽셀의 필터링 연산을 하나의 연산기로 수행하기 때문에 연산 개수와 연산기의 개수를 감소시켰다. 고속 smoothing 결정 알고리즘을 비트 비교기만을 사용하고, 고속 필터계수 생성 알고리즘은 LUT와 덧셈 연산만을 사용하여 연산량, 하드웨어 면적, 계산 지연 시간을 감소시켰다.

제안하는 구조를 TSMC 0.13um 공정을 이용하여 합성한 결과 최대 동작 주파수는 164MHz이고, 게이트 수는 40.5k이다. 4x4 PU를 기준으로 제안하는 구조의 처리 사이클 수는 기존 구조 대비 93.7% 감소하였다.

감사의 글

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

참고문헌

[1] G. J. Sullivan, J. R.Ohm, W. J. Han and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.

[2] J. Lainema, F. Bossen, W. J. Han, J. Min and K. Ugur, “Intra Coding of the HEVC Standard,” *IEEE Trans. Circ. and Syst. for Video Tech.*, vol. 22, no. 12, pp.

1792-1801, Dec. 2012.
[3] 최해철, “HEVC 화면내 예측 및 부호화”, 전자공학회지, 제38권, 제8호, pp. 33-39, 2011년 8월.
[4] B. Bross, W. J. Han, G. J. Sullivan, J. R. Ohm and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 6,* JCT-VC, JCTVC-H1003, Feb. 2012.
[5] I. E. Richardson, *The H.264 Advanced Video Compression Standard : Second Edition,* John Wiley & Sons, April 2010.
[6] F. Li. G. Shi and F. Wu, “An efficient VLSI architecture for 4×4 intra prediction in the High Efficiency Video Coding (HEVC) standard,” *18th IEEE Intl. Conf. on Image Processing,* pp. 373-376, Sept. 2011.
[7] B. Bross, W. J. Han, G. J. Sullivan, J. R. Ohm and T. Wiegand, *Working Draft 1 of High Efficiency Video Coding,* JCT-VC, JCTVC- C403, Oct. 2010.
[8] X. He, D. Zhou, J. Zhou and S. Goto, “High Profile Intra Prediction Architecture for UHD H.264 Decoder,” *IPSJ Trans. on Syst. LSI Design Method.,* Vol. 3, pp. 303-313, Aug. 2010.
[9] M. Nadeem, S. Wong and G. Kuzmanov, “An Efficient Hardware Design for Intra-prediction in H.264/AVC Decoder,” *Saudi Intl. Electronics, Comm. and Photonics Conf. 2011,* pp. 1-6, April 2011.
[10] HM 7.1 Reference Software. Available: http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-7.1

저자소개

정홍균(Hongkyun Jung)

2007년 한밭대학교 정보통신공학과 공학사
2009년 한밭대학교 정보통신공학과 공학석사
2009년～현재 한밭대학교 정보통신공학과 박사과정
※ 관심분야 : SoC 플랫폼 설계, 하드웨어/소프트웨어 통합설계, 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 한국전자통신연구원 시스템 IC 설계팀 선임연구원
2010년～2011년 Visiting Scholar at UTD (Univ of Texas at Dallas)
2003년～현재 한밭대학교 정보통신공학과 교수
※관심분야 : SoC 플랫폼 설계 및 검증, 하드웨어/소프트웨어 통합설계 및 통합검증, 멀티미디어 코덱 설계