

병렬처리를 이용한 HEVC 디코더의 화면간 예측 보간 필터 하드웨어 구조

최승환*, 배종우*
 *명지대학교 정보통신공학과
 e-mail:choish159@naver.com
 jongwoo.bae@gmail.com

A Interpolation Hardware Architecture for HEVC Inter-Prediction Decoder Using Parallel Process

Seung-Hwan Choi*, Jong-Woo Bae*

*Dept of Information and Communications Engineering, Myongji University

요 약

본 논문에서는 HEVC 디코더에서 화면간 예측의 보간 필터에 대한 하드웨어 구조를 제시하고, 설계 및 분석결과를 통해 연구 결론을 도출하는 것이 목적이다.

제안하는 하드웨어 구조는 보간 필터의 각 필터 간의 유사성을 확인하고 빠르게 데이터를 처리하기 위한 병렬처리 방법을 제시한다. 또한 레지스터를 통한 데이터를 재사용하는 방식을 이용하여 외부 메모리와 불필요한 연결을 줄여 성능을 향상시켰다.

1. 서론

HEVC는 ITU-T의 VCEG(Video Coding Experts Group)와 ISO/IEC의 MPEG(Moving Picture Experts Group)이 협력하여 JCT-VC(Joint Collaborative Team on Video Coding)이라는 팀이 표준화를 진행하였다. HEVC의 목적은 이전 비디오 코덱의 표준인 H.264/AVC에 비해 50% 정도의 비트 전송률의 감소와 UHD 해상도의 지원을 목표로 하였다.

2013년 1월 차세대 비디오 코덱인 HEVC(High Efficiency Video Codec)가 표준안으로 승인되었다. HEVC는 향후 3D TV 방송, IPTV, 감시카메라, 스마트폰, 디지털 캠코더 등 여러 응용 분야에서 고품질 비디오 서비스에 빠른 속도로 적용될 가능성이 매우 높은 중요한 비디오 코덱 표준이다.

HEVC의 화면간 예측에서는 H.264/AVC에 비해 예측 부호화 블록의 종류가 다양해졌으며, 움직임 벡터의 예측 성능을 향상시키기 위해 더 많은 참조 블록을 사용하고, 움직임 벡터의 정보량을 줄이기 위해 새로운 움직임 벡터 부호화 기술인 Merge 모드를 사용한다.

본 논문에서는 움직임 예측을 향상시키기 위해 사용하는 보간 필터에 대하여 필터 간의 유사성을 이용한 병렬처리와 레지스터를 이용한 데이터 재사용 기법을 사용하여 성능을 향상시키는 보간 필터 하드웨어 구조를 제안한다.

2. 보간 필터 알고리즘

HEVC의 휘도 보간 필터는 그림 1과 같이 대문자로 쓰인 정수 화소를 이용하여 소문자로 쓰인 부화소의 값을

구하는 필터로 구성된다. 휘도 보간 필터는 화소의 위치에 따라 7-tap 필터와 8-tap 필터 두 가지 종류의 필터를 사용하고, 또한 부화소의 위치에 따라 표 1과 같이 3가지 종류의 필터 계수를 가진다.

A _{1,1}				A _{0,1}	a _{0,1}	b _{0,1}	c _{0,1}	A _{1,1}				A _{2,1}
A _{1,0}				A _{0,0}	a _{0,0}	b _{0,0}	c _{0,0}	A _{1,0}				A _{2,0}
	d _{1,0}			d _{0,0}	e _{0,0}	f _{0,0}	g _{0,0}	d _{1,0}				d _{2,0}
		h _{1,0}		h _{0,0}	i _{0,0}	j _{0,0}	k _{0,0}	h _{1,0}				h _{2,0}
			n _{1,0}	n _{0,0}	p _{0,0}	q _{0,0}	r _{0,0}	n _{1,0}				n _{2,0}
A _{1,1}				A _{0,1}	a _{0,1}	b _{0,1}	c _{0,1}	A _{1,1}				A _{2,1}
A _{1,2}				A _{0,2}	a _{0,2}	b _{0,2}	c _{0,2}	A _{1,2}				A _{2,2}

<그림 1> HEVC의 휘도 보간 화소

부화소의 위치	필터 계수
1/4	-1, 4, -10, 58, 17, -5, 1
1/2	-1, 4, -11, 40, 40, -11, 4, -1
3/4	1, -5, 17, 58, -10, 4, -1

<표 1> 부화소 위치별 휘도 보간 필터 계수

부화소의 위치	부화소
1/4	a, d, e, f, g
1/2	b, h, i, j, k
3/4	c, n, p, q, r

<표 2> 부화소별 위치

1/4와 3/4의 위치에 자리한 화소들은 7-tap 필터를 사용하고 1/2의 위치에 자리한 화소들은 8-tap 필터를 사용한다. 표 2와 같이 1/4의 위치의 화소들은 a, d, e, f, g의 화소들이고 1/2의 위치의 화소들은 b, h, i, j, k 화소들이고 3/4의 위치의 화소들은 c, n, p, q, r 화소들이다. 각각의 화소들은 위치의 따라 필터 계수를 가져와 사용을 한다. 보간 과정은 보간 필터에 정수 화소를 이용하여 부화소 a, b, c, d, h, n을 먼저 생성을 해주게 된다. 앞서 생성된 a, b, c의 부화소를 이용하여 e, f, g, i, j, k, p, q, r의 부화소를 생성한다.

HEVC의 색채 보간 필터는 그림 2와 같이 대문자로 쓰인 정수 화소를 이용하여 소문자로 쓰인 부화소의 값을 구하는 필터로 구성된다. 색채 보간 필터는 4-tap 필터 한 종류만 사용을 하고 부화소의 위치에 따라 표 3과 같이 7 가지 종류의 필터 계수를 가진다.

	ha _{0,-1}	hb _{0,-1}	hc _{0,-1}	hd _{0,-1}	he _{0,-1}	hf _{0,-1}	hg _{0,-1}	hh _{0,-1}	
ah _{-1,0}	B _{0,0}	ab _{0,0}	ac _{0,0}	ad _{0,0}	ae _{0,0}	af _{0,0}	ag _{0,0}	ah _{0,0}	B _{1,0}
bh _{-1,0}	ba _{0,0}	bb _{0,0}	bc _{0,0}	bd _{0,0}	be _{0,0}	bf _{0,0}	bg _{0,0}	bh _{0,0}	ba _{1,0}
ch _{-1,0}	ca _{0,0}	cb _{0,0}	cc _{0,0}	cd _{0,0}	ce _{0,0}	cf _{0,0}	cg _{0,0}	ch _{0,0}	ca _{1,0}
dh _{-1,0}	da _{0,0}	db _{0,0}	dc _{0,0}	dd _{0,0}	de _{0,0}	df _{0,0}	dg _{0,0}	dh _{0,0}	da _{1,0}
eh _{-1,0}	ea _{0,0}	eb _{0,0}	ec _{0,0}	ed _{0,0}	ee _{0,0}	ef _{0,0}	eg _{0,0}	eh _{0,0}	ea _{1,0}
fh _{-1,0}	fa _{0,0}	fb _{0,0}	fc _{0,0}	fd _{0,0}	fe _{0,0}	ff _{0,0}	fg _{0,0}	fh _{0,0}	fa _{1,0}
gh _{-1,0}	ga _{0,0}	gb _{0,0}	gc _{0,0}	gd _{0,0}	ge _{0,0}	gf _{0,0}	gg _{0,0}	gh _{0,0}	ga _{1,0}
hh _{-1,0}	ha _{0,0}	hb _{0,0}	hc _{0,0}	hd _{0,0}	he _{0,0}	hf _{0,0}	hg _{0,0}	hh _{0,0}	ha _{1,0}
	B _{0,1}	ab _{0,1}	ac _{0,1}	ad _{0,1}	ae _{0,1}	af _{0,1}	ag _{0,1}	ah _{0,1}	B _{1,1}

<그림 2> HEVC의 색채 보간 화소

부화소 위치	필터 계수
1/8	-2, 58, 10, -2
1/4	-4, 54, 16, -2
3/8	-6, 46, 28, -4
1/2	-4, 36, 36, -4
5/8	-4, 28, 46, -6
3/4	-2, 16, 54, -4
7/8	-2, 10, 58, -2

<표 3> 부화소 위치별 색채 보간 필터 계수

색채 보간 필터도 휘도 보간 필터와 동일하게 대문자로 쓰여진 정수 화소를 이용하여 ab, ac, ad, ae, af, ag, ah와 ba, ca, da, ea, fa, ga, ha의 부화소를 생성한 후 앞서 생성된 부화소를 이용하여 bX, cX, dX, eX, fX, gX, hX의 부화소를 생성한다.

3. 제안하는 구조

본 논문에서 제안하는 구조는 휘도 보간 필터와 색채 보간 필터에 병렬구조와 데이터 재사용을 위해 레지스터의 배치한 하드웨어 구조를 제안한다. 위의 두 가지 방법에 대해 밑에서는 수식을 통해 제안하는 두 가지 방법에 대해 설명을 하고자 한다.

아래에 보이는 수식 (1)~(15)는 휘도 보간 필터의 수식이다. 휘도 보간 필터의 수식을 살펴보면 (1)~(3)까지에 이용되는 정수 화소들이 겹침을 알 수 있고, 이후의 수식에서 3개씩 묶어서 확인을 해보게 되면 위와 같이 정수 화소 또는 부화소가 겹치게 들어가는 것을 알 수 있다. 또한 각각 3개씩 묶인 수식들은 앞서 설명한 부화소의 위치에 따라 각기 다른 필터에서 연산을 할 수 있다. 이를 통해 각각의 필터가 동시에 같은 입력을 넣어준다면 동시에 병렬처리가 가능함을 알 수 있다. 또한 (1)~(3) 수식을 통해 계산한 부화소 a, b, c는 (7)~(15) 수식의 입력으로 사용이 된다. 이를 통해 데이터의 재사용도 가능함을 알 수 있다.

$$a_{0,0} = (-A_{-3,0} + 4*A_{-2,0} - 10*A_{-1,0} + 58*A_{0,0} + 17*A_{1,0} - 5*A_{2,0} + A_{3,0}) \gg shift1 \quad (1)$$

$$b_{0,0} = (-A_{-3,0} + 4*A_{-2,0} - 11*A_{-1,0} + 40*A_{0,0} + 40*A_{1,0} - 11*A_{2,0} + 4*A_{3,0} - A_{4,0}) \gg shift1 \quad (2)$$

$$c_{0,0} = (A_{-2,0} - 5*A_{-1,0} + 17*A_{0,0} + 58*A_{1,0} - 10*A_{2,0} + 4*A_{3,0} - A_{4,0}) \gg shift1 \quad (3)$$

$$d_{0,0} = (-A_{0,-3} + 4*A_{0,-2} - 10*A_{0,-1} + 58*A_{0,0} + 17*A_{0,1} - 5*A_{0,2} + A_{0,3}) \gg shift1 \quad (4)$$

$$h_{0,0} = (-A_{0,-3} + 4*A_{0,-2} - 11*A_{0,-1} + 40*A_{0,0} + 40*A_{0,1} - 11*A_{0,2} + 4*A_{0,3} - A_{0,4}) \gg shift1 \quad (5)$$

$$n_{0,0} = (A_{0,-2} - 5*A_{0,-1} + 17*A_{0,0} + 58*A_{0,1} - 10*A_{0,2} + 4*A_{0,3} - A_{0,4}) \gg shift1 \quad (6)$$

$$e_{0,0} = (-a_{0,-3} + 4*a_{0,-2} - 10*a_{0,-1} + 58*a_{0,0} + 17*a_{0,1} - 5*a_{0,2} + a_{0,3}) \gg shift2 \quad (7)$$

$$i_{0,0} = (-a_{0,-3} + 4*a_{0,-2} - 11*a_{0,-1} + 40*a_{0,0} + 40*a_{0,1} - 11*a_{0,2} + 4*a_{0,3} - a_{0,4}) \gg shift2 \quad (8)$$

$$p_{0,0} = (a_{0,-2} - 5*a_{0,-1} + 17*a_{0,0} + 58*a_{0,1} - 10*a_{0,2} + 4*a_{0,3} - a_{0,4}) \gg shift2 \quad (9)$$

$$f_{0,0} = (-b_{0,-3} + 4*b_{0,-2} - 10*b_{0,-1} + 58*b_{0,0} + 17*b_{0,1} - 5*b_{0,2} + b_{0,3}) \gg shift2 \quad (10)$$

$$j_{0,0} = (-b_{0,-3} + 4*b_{0,-2} - 11*b_{0,-1} + 40*b_{0,0} + 40*b_{0,1} - 11*b_{0,2} + 4*b_{0,3} - b_{0,4}) \gg shift2 \quad (11)$$

$$q_{0,0} = (b_{0,-2} - 5*b_{0,-1} + 17*b_{0,0} + 58*b_{0,1} - 10*b_{0,2} + 4*b_{0,3} - b_{0,4}) \gg shift2 \quad (12)$$

$$g_{0,0} = (-c_{0,-3} + 4*c_{0,-2} - 10*c_{0,-1} + 58*c_{0,0} + 17*c_{0,1} - 5*c_{0,2} + c_{0,3}) \gg shift2 \quad (13)$$

$$k_{0,0} = (-c_{0,-3} + 4*c_{0,-2} - 11*c_{0,-1} + 40*c_{0,0} + 40*c_{0,1} - 11*c_{0,2} + 4*c_{0,3} - c_{0,4}) \gg shift2 \quad (14)$$

$$r_{0,0} = (c_{0,-2} - 5*c_{0,-1} + 17*c_{0,0} + 58*c_{0,1} - 10*c_{0,2} + 4*c_{0,3} - c_{0,4}) \gg shift2 \quad (15)$$

<식 (1)~(15)> HEVC의 휘도 보간 필터의 수식

아래에 보이는 수식 (16)~(36)은 색채 보간 필터의 수식이다. 색채 보간 필터에서는 7개씩 묶어서 수식을 확인해 보면 휘도 보간 필터와 마찬가지로 정수 화소 또는 부화소가 겹치게 들어가는 것을 알 수 있다. 또한 (16)~(22) 수식을 통해 계산한 부화소 ab, ac, ad, ae, af, ag, ah는 (30)~(36) 수식의 입력으로 사용이 된다. 이와 같이 색채 보간 필터에서도 휘도 보간 필터와 마찬가지로 동일한 방법을 이용할 수 있음을 알 수 있다.

$$ab_{0,0} = (-2 * B_{-1,0} + 58 * B_{0,0} + 10 * B_{1,0} - 2 * B_{2,0}) \gg \text{shift1} \quad (16)$$

$$ac_{0,0} = (-4 * B_{-1,0} + 54 * B_{0,0} + 16 * B_{1,0} - 2 * B_{2,0}) \gg \text{shift1} \quad (17)$$

$$ad_{0,0} = (-6 * B_{-1,0} + 46 * B_{0,0} + 28 * B_{1,0} - 4 * B_{2,0}) \gg \text{shift1} \quad (18)$$

$$ae_{0,0} = (-4 * B_{-1,0} + 36 * B_{0,0} + 36 * B_{1,0} - 4 * B_{2,0}) \gg \text{shift1} \quad (19)$$

$$af_{0,0} = (-4 * B_{-1,0} + 28 * B_{0,0} + 46 * B_{1,0} - 6 * B_{2,0}) \gg \text{shift1} \quad (20)$$

$$ag_{0,0} = (-2 * B_{-1,0} + 16 * B_{0,0} + 54 * B_{1,0} - 4 * B_{2,0}) \gg \text{shift1} \quad (21)$$

$$ah_{0,0} = (-2 * B_{-1,0} + 10 * B_{0,0} + 58 * B_{1,0} - 2 * B_{2,0}) \gg \text{shift1} \quad (22)$$

$$ba_{0,0} = (-2 * B_{0,-1} + 58 * B_{0,0} + 10 * B_{0,1} - 2 * B_{0,2}) \gg \text{shift1} \quad (23)$$

$$ca_{0,0} = (-4 * B_{0,-1} + 54 * B_{0,0} + 16 * B_{0,1} - 2 * B_{0,2}) \gg \text{shift1} \quad (24)$$

$$da_{0,0} = (-6 * B_{0,-1} + 46 * B_{0,0} + 28 * B_{0,1} - 4 * B_{0,2}) \gg \text{shift1} \quad (25)$$

$$ea_{0,0} = (-4 * B_{0,-1} + 36 * B_{0,0} + 36 * B_{0,1} - 4 * B_{0,2}) \gg \text{shift1} \quad (26)$$

$$fa_{0,0} = (-4 * B_{0,-1} + 28 * B_{0,0} + 46 * B_{0,1} - 6 * B_{0,2}) \gg \text{shift1} \quad (27)$$

$$ga_{0,0} = (-2 * B_{0,-1} + 16 * B_{0,0} + 54 * B_{0,1} - 4 * B_{0,2}) \gg \text{shift1} \quad (28)$$

$$ha_{0,0} = (-2 * B_{0,-1} + 10 * B_{0,0} + 58 * B_{0,1} - 2 * B_{0,2}) \gg \text{shift1} \quad (29)$$

$$bX_{0,0} = (-2 * aX_{0,-1} + 58 * aX_{0,0} + 10 * aX_{0,1} - 2 * aX_{0,2}) \gg \text{shift2} \quad (30)$$

$$cX_{0,0} = (-4 * aX_{0,-1} + 54 * aX_{0,0} + 16 * aX_{0,1} - 2 * aX_{0,2}) \gg \text{shift2} \quad (31)$$

$$dX_{0,0} = (-6 * aX_{0,-1} + 46 * aX_{0,0} + 28 * aX_{0,1} - 4 * aX_{0,2}) \gg \text{shift2} \quad (32)$$

$$eX_{0,0} = (-4 * aX_{0,-1} + 36 * aX_{0,0} + 36 * aX_{0,1} - 4 * aX_{0,2}) \gg \text{shift2} \quad (33)$$

$$fX_{0,0} = (-4 * aX_{0,-1} + 28 * aX_{0,0} + 46 * aX_{0,1} - 6 * aX_{0,2}) \gg \text{shift2} \quad (34)$$

$$gX_{0,0} = (-2 * aX_{0,-1} + 16 * aX_{0,0} + 54 * aX_{0,1} - 4 * aX_{0,2}) \gg \text{shift2} \quad (35)$$

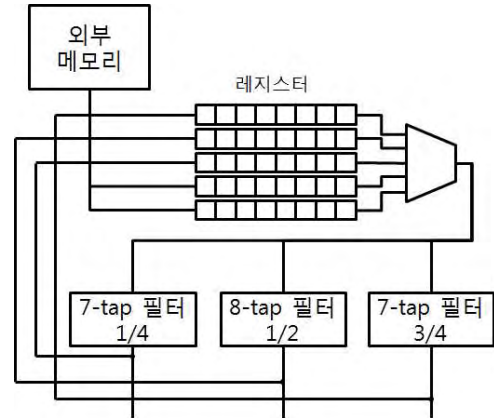
$$hX_{0,0} = (-2 * aX_{0,-1} + 10 * aX_{0,0} + 58 * aX_{0,1} - 2 * aX_{0,2}) \gg \text{shift2} \quad (36)$$

<식 (16)~(36)> HEVC의 색채 보간 필터의 수식

3.1 제안하는 휘도 보간 필터

본 논문에서 제안하는 휘도 보간 필터의 구조는 아래의 그림 3과 같다. 휘도 보간 필터에서는 3개의 필터에 같은 입력을 동시에 받아 병렬처리를 해주는 구조를 가지면서 또한 레지스터를 5개를 배치해두고 2개는 외부 메모리에서 들어오는 정수 화소의 값을 저장하고 나머지 3개는 부화소를 계산하기 위해 사용되는 a, b, c 화소를 각각의 레

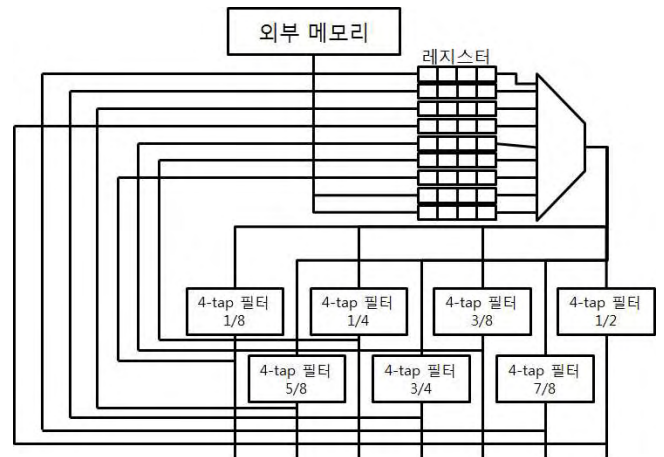
지스터에 저장을 하여 데이터 재사용 기법을 사용한다. 휘도 보간 필터 각 레지스터들은 최대 8개의 데이터를 저장할 수 있도록 한다. 8개의 데이터가 저장 가능한 레지스터들은 각각의 필터의 입력으로 들어가게 된다. 1/4 7-tap 필터는 레지스터에 저장된 앞의 7개의 데이터를 사용하고 3/4 7-tap 필터는 뒤의 7개의 데이터를 사용하고 마지막으로 1/2 8-tap 필터는 저장된 8개의 데이터를 모두 사용한다.



<그림 3> 제안하는 휘도 보간 필터 하드웨어 구조

3.2 제안하는 색채 보간 필터

위에서 제안하는 휘도 보간 필터의 구조와 마찬가지로 7개의 필터를 동시에 병렬처리를 하고 레지스터는 9개를 두고 2개는 외부 메모리에서 정수 화소의 값을 받아오고 나머지 7개는 필터로 계산한 후에 부화소 ab, ac, ad, ae, af, ag, ah를 각각의 레지스터 저장을 한다. 색채 필터의 경우에는 4-tap 필터만 사용하기 때문에 레지스터는 4개의 값을 저장할 수 있도록 하였다. 제안하는 구조는 아래의 그림 4와 같다.



<그림 4> 제안하는 색채 보간 필터 하드웨어 구조

4. 결과

4K(3840X2160)의 해상도에 30fps인 영상을 처리할 때 초당 3840X2160X30+3840X2160X30X0.5 = 373248000의 픽셀을 발생한다. 최악의 경우는 4X4 휘도 블록과 두 개의 양

방향 예측의 2X2 색채 블록이 들어오는 경우이다. 이 경우 휘도 블록은 $4X2 = 8\text{cycles}$, 색채 블록은 $2X2 + 2X2 = 8\text{cycles}$ 이 필요하고 이는 24개의 픽셀을 계산하는데 총 16 cycles이 필요하다. cycle당 1.5개의 픽셀을 계산하므로 동작 주파수는 약 최소 249MHz로 동작해야한다.

5. 결론

본 논문에서는 수식을 통해 필터 간의 데이터 의존성이 없음을 정수 화소를 이용하여 계산된 부화소가 재사용이 가능함을 알고 이를 이용한 병렬처리 구조와 레지스터를 이용한 외부메모리와의 불필요한 데이터 이동을 줄여 처리 속도를 증가시키는 하드웨어 구조를 제안하였다. 현재에도 4K UHD의 해상도를 가지는 TV들이 등장하는 가운데 이후에는 더 높은 해상도를 가지는 미디어들이 등장할 것이다. 본 논문은 이러한 시대의 흐름에 맞춰 앞으로의 비디오 코덱 기술 연구의 좋은 영향일 끼칠 것으로 예상된다.

참고문헌

- [1] B. Bross, W.J. Han, G. J. Sullivan, J. R. Ohm and T.Wiegand, "High efficiency Video Coding(HEVC) text specification draft 10", The Joint Collaborative Team on Video Coding(JCT-VC), JCTVC-L1003_34, 2013
- [2] Z. Guo, D. Zhou, and S. Goto, "An optimized MC interpolation architecture for HEVC," in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, pp. 1117-1120, 2012.
- [3] E. Alshina, "Summary of CE4: interpolation for MC (Chroma)," Document of Joint Collaborative Team on Video Coding, JCTVC-D345, Jan. 2011.
- [4] G. L. Li, C. C. Wang, K. H. Chiang "An efficient motion vector prediction method for avoiding AMVP data dependency for HEVC", IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pp 7363-7366, 2014