

H.264/AVC 부호화기용 움직임 보상의 연구

김원삼* · 손승일*

*한신대학교 정보통신학과

A Study on Motion Compensation for H.264/AVC Encoder

Won-sam Kim* · Seung-il Sonh*

*Department of Information and Communication, Hanshin University

E-mail : ryuna80@gmail.com, saisonh@hs.ac.kr

요 약

여러 동영상 부호화 방식에서 영상프레임을 분할해서 이전에 부호화된 프레임으로부터 움직임을 추정하여 현재의 블록을 예측하는 움직임 보상을 사용하고 있다. 움직임 보상에 사용되는 화소정밀도가 높을수록 보다 좋은 성능을 갖지만 연산량은 증가하게 된다. 본 논문에서는 1/4 화소 정밀도를 지원하는 H.264/AVC 부호화기에 적합한 움직임 보상기를 연구하였다. 전치 배열과 휘도 6-tap 필터 3개를 사용하여 높은 하드웨어 이용률을 갖게하였다. VHDL을 사용하여 Xilinx ISE툴을 사용하여 합성하고, 보드 수준에서 PCI인터페이스를 사용하여 검증하였다. 본 논문에서 제안하는 움직임 보상기는 실시간 처리를 요구하는 분야에 응용 가능할 것으로 예견된다.

키워드

H.264/AVC, 인터 예측, 움직임 보상, FPGA, HDL

I. 서 론

H.264/AVC 기술은 ISO/IEC 의 MPEG과 ITU-T의 VCEG 두 그룹이 공동 연구기관 JVT를 창설하여 새롭게 제안한 동영상 압축에 관한 국제 표준이다[1].

H.264/AVC는 우수한 화질을 얻기 위해 이전 기술과는 다른 4x4 블록 단위의 정수변환 부호화, 가변 블록 움직임 보상, 다양한 방향성을 고려한 인트라 모드 부호화 기법, 다중 참조 영상 부호화 기법, 그리고 1/4 화소 단위 움직임 벡터 추정 기법을 사용하여 MPEG-4 보다 2배 이상 향상된 동영상 압축률과 지상파/위성 DMB 등에 사용으로 많은 사람들의 관심과 국제 비디오 압축 표준으로 인정받고 있다. 이 기술은 위에서 나열한 기법을 근간으로 기존의 H.263이나 MPEG-4 SP(Simple Profile)에 비해 동일한 화질에서 압축률이 최대 1.5~2 배 가량 더 좋다는 것이 실험을 통해 밝혀졌으나 연산량 또한 증가하여 복잡도면에서는 MPEG-4 SP 부호화기보다 약 14배 높은 것으로 보고되었다[2].

압축 효율을 높이기 위한 부호화 기술들은 기술 자체의 복잡도는 고려하지 않고 오직 효율이라는 측면에 중점을 두고 개발되었기 때문에

H.264/AVC의 복잡도가 큰 인터 예측 부호화에 효율적인 하드웨어 구현이 필요하다. 따라서, 본 논문에서는 1/4 화소 정밀도를 갖는 H.264/AVC 부호화기에 적합한 움직임 보상기를 연구하였다.

II. 움직임 보상 알고리즘

H.264/AVC는 움직임 예측을 위해 화상을 4배 확대하게 되는데, 이때 확대에 의한 경계점이 화질에 나쁜 영향을 미치므로 이를 제거하기 위해 보간을 수행해야 한다. 4배 확대에 의한 보간 방법은 보다 효율적인 움직임 벡터의 연산을 가능하게 한다[1][3].

움직임 보상 예측을 할 때 정수화소 이하의 화소 정밀도 신호는 참조픽처의 화소 값들 간의 보간을 통해 생성한다. 그림 1은 보간 후 신호와 보간 전의 정수화소 신호와의 위치관계를 나타낸다. 여기서 알파벳 대문자는 보간 전의 정수화소 신호를 나타내고 알파벳 소문자는 1/2화소 정밀도, 1/4화소 정밀도의 신호를 의미한다[1][4].

첫 번째 단계로 정수화소 신호로부터 6-tap FIR 필터를 사용하여 1/2화소 정밀도 예측신호를

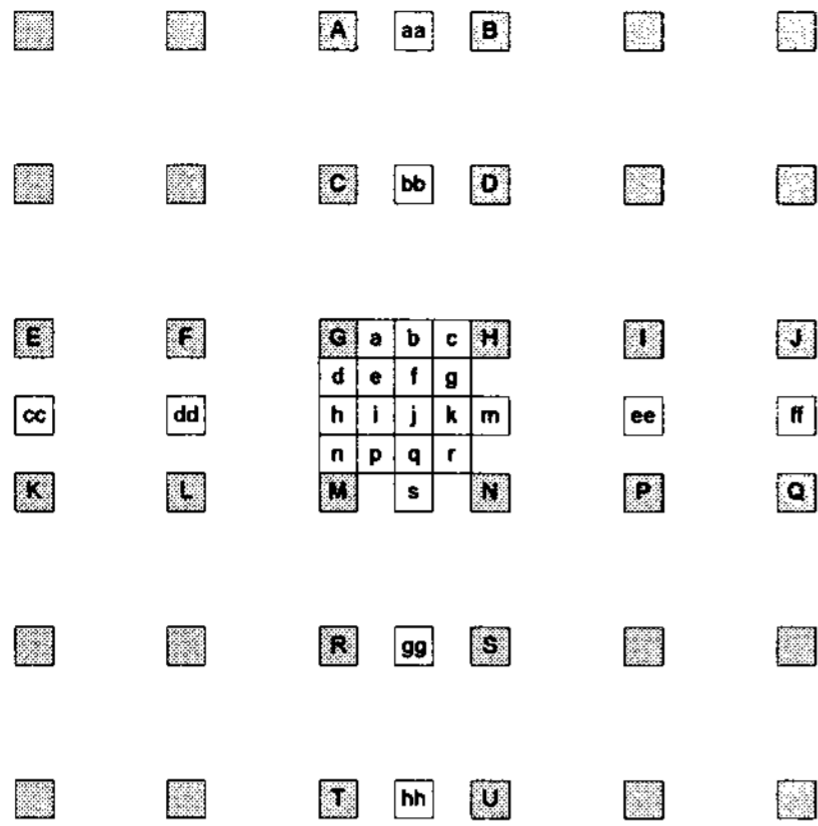


그림 1. 보간을 위한 샘플의 위치

생성하고, 그 다음 과정으로 2-tap 평균치 필터에 의해 1/4화소정밀도 예측신호를 생성한다.

1. 1/2 화소정밀도 예측신호의 생성

수평방향으로 두 개의 정수화소 사이에 위치한 1/2화소 신호는 수평방향으로 6-tap 필터를 생성한다. 예를 들어, 그림 1의 화소 E, F, G, H, I, J에 대해 수평방향 6-tap 필터링을 적용하면 다음 식과 같다.

$$b_1 = (E - 5F + 20G + 20H - 5I + J) \quad (\text{식 1})$$

$$b = \text{Clip1}((b_1 + 16) \gg 5) \quad (\text{식 2})$$

두 개의 정수화소 신호의 수직방향으로 중간 위치인 1/2화소신호는 수직방향으로 6-tap 필터링을 행하여 생성한다.

예를 들어, 그림 1의 화소 h는 정수화소 A, C, G, M, R, T에 수직방향 6-tap 필터를 적용한 것이다. 필터링식은 다음과 같다.

$$h_1 = (A - 5C + 20G + 20M - 5R + T) \quad (\text{식 3})$$

$$h = \text{Clip1}((h_1 + 16) \gg 5) \quad (\text{식 4})$$

2. 1/4 화소 정밀도 예측신호의 생성

앞서 설명한 것처럼 1/2화소정밀도 신호의 값을 생성한 후, 평균치필터를 사용하여 1/4화소정밀도 신호를 생성한다.

화소 a, c, i, k는 주변에 인접한 정수화소 신호 또는 1/2화소 신호의 수평방향 평균치 필터를 사용하여 생성한다. 예를 들어 화소 a, f, r은 다음 식에 의해 계산된다.

$$a = (G + b + 1) \gg 1 \quad (\text{식 5})$$

$$f = (b + j + 1) \gg 1 \quad (\text{식 6})$$

$$r = (m + s + 1) \gg 1 \quad (\text{식 7})$$

3. 색차의 예측 보간 신호의 생성

색차신호의 예측보간 신호는 그림과 같이 정수화소를 1/8화소정밀도로 선형 보간하여 생성하며 다음 식으로 표현된다.

$$\text{predPartLXc}[x_c, y_c] = ((8 - x\text{Frac}_c)(8 - y\text{Frac}_c)*A + x\text{Frac}_c(8 - y\text{Frac}_c)*B + (8 - x\text{Frac}_c)*y\text{Frac}_c*C + x\text{Frac}_c*y\text{Frac}_c*D + 32) \gg 6 \quad (\text{식 8})$$

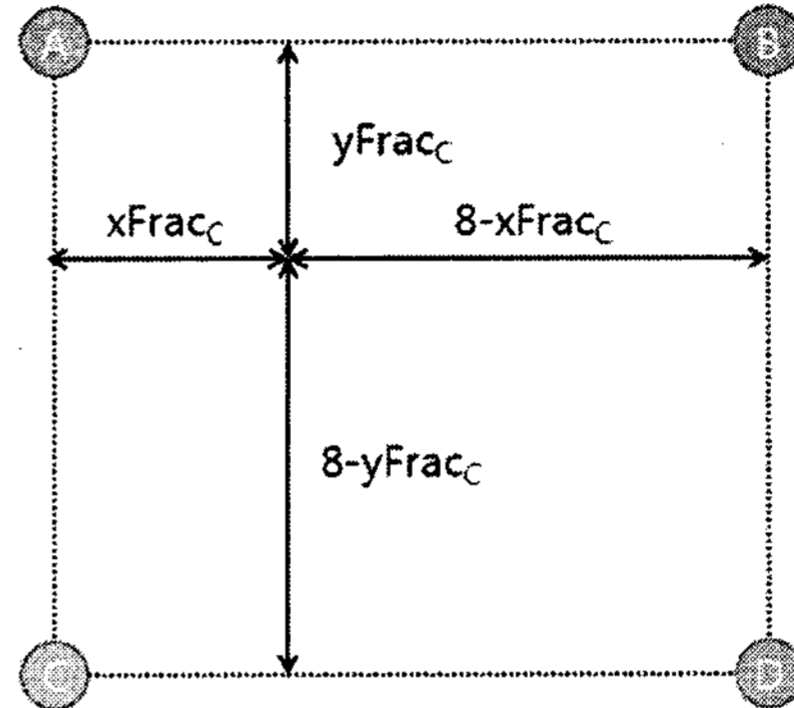


그림 4. 색차 보간을 위한 샘플의 위치

III. 제안하는 구조

그림 3은 움직임 보상을 수행할 때 부호화기와 복호화기의 보상 범위의 차이를 나타낸다.

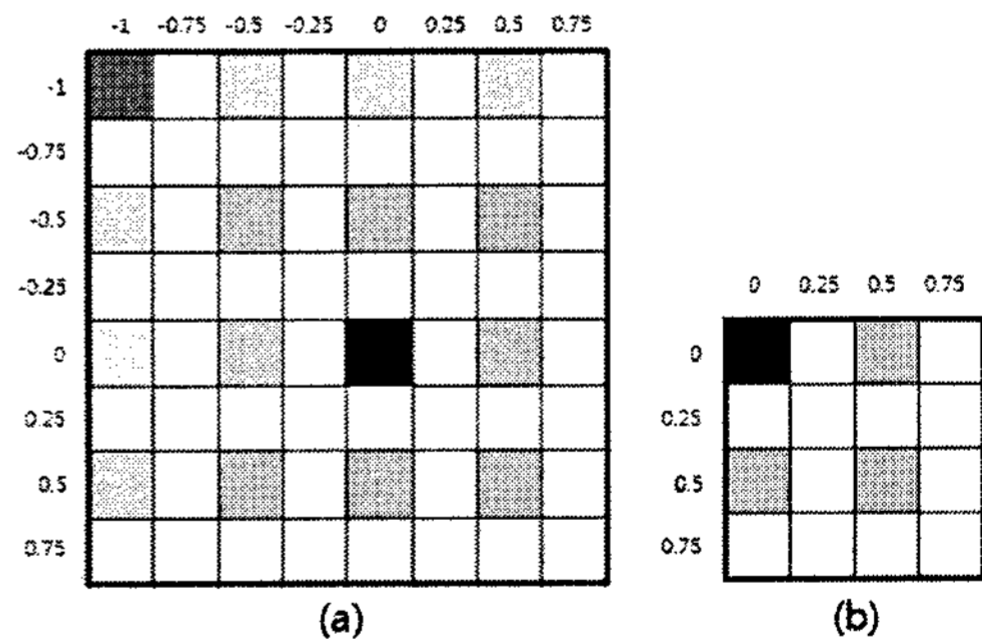


그림 5. 부호화기와 복호화기의 보상 범위

그림 3의 (a)는 부호화기의 움직임 추정 수행 시 정수화소를 기준으로 수직 및 수평 방향으로 -1 ~ 0.75 까지 탐색이 필요하다는 것을 보여주고, 그림 3의 (b)는 복호화기에서 움직임 보상 수행 시 정수화소를 기준으로 0 ~ 0.75 범위내에서 움직임 벡터의 값에 따라 예측 신호를 생성해야

함을 보여준다. 본 논문은 부호화기에 사용되는 움직임 보상의 구현이므로 그림 3의 (a)의 보간 범위를 갖는다.

H.264/AVC의 인터 예측은 최소 블록 크기는 4x4이고 각각의 매크로블록 및 서브매크로블록의 파티션은 여러 개의 4x4블록들로 나누어 질 수 있다. 따라서 가변 블록 움직임 보상을 수행할 때 4x4 블록 단위로 나누어 수행하게 되면 표준을 준수하면서 손쉽게 움직임 보상을 수행 할 수 있다. 부호화기에서는 4x4 블록 당 16x16크기의 보상을 해야하지만 부호화기에서는 그림 4와 같이 20x20 크기의 보상을 수행해야 한다.

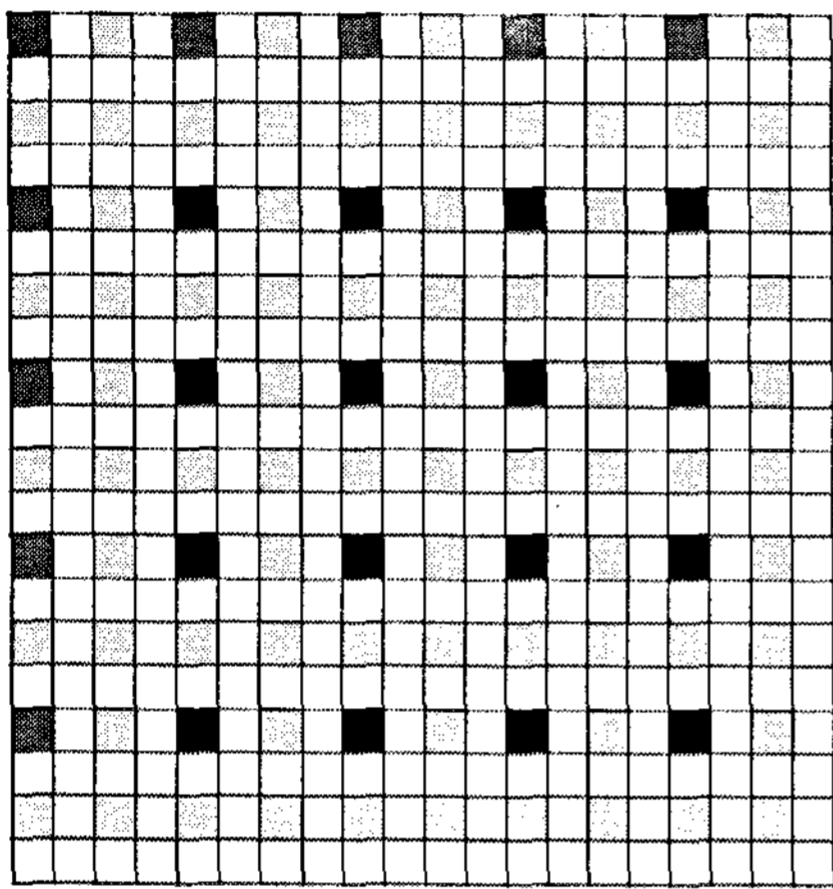


그림 4. 4x4블록 단위의 움직임 보상 범위

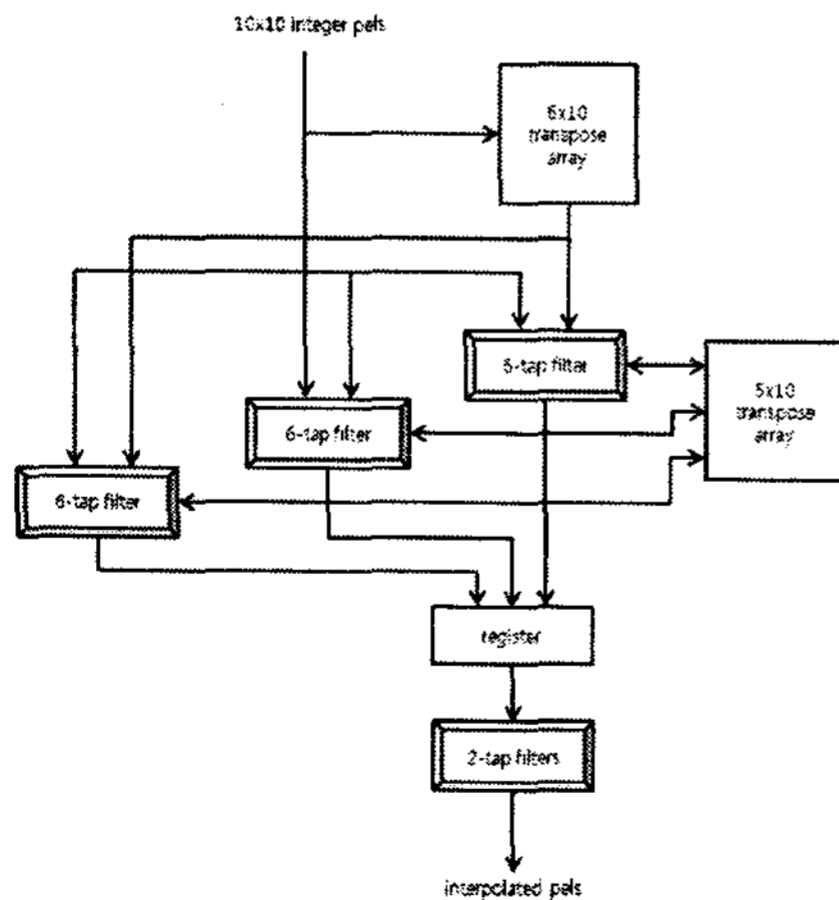


그림 5. 전체 모듈의 블록도

그림 1에서 하나의 정수 화소에 대해 보상에 필요한 인접 정수 화소가 6x6이 필요하기 때문에 그림 4에서 4x4 블록에 대해 보상을 수행하기 위해서는 인접 정수 화소가 10x10이 필요하다. 이것을 입력으로 4x4 블록 단위의 움직임 보상을 구

현한 전체 블록도는 그림 5와 같다.

입력이 들어오기 시작하면 먼저 그림 1의 aa, bb, b, gg, hh 화소들을 6-tap 필터를 통하여 구한다. 이 화소들은 나중에 j에 해당하는 화소를 구하기 위해 입력되어진 모든 행에 대하여 수행되어야 하고 5x10 전치 배열과 레지스터로 전달된다. 여기서, j 화소를 구하기 위해 수직 방향으로 하는 이유는 수평 방향으로 수행하여도 동일한 결과를 얻을 수 있지만 추가적인 전치 배열을 필요로 하기 때문이다.

6-tap 필터는 3개를 사용하기 때문에 하나의 행에 2사이클이 소요되고 전체 10행에 대해서는 20 사이클이 소요된다. 또한 초기 입력된 10x10 화소는 6x10 전치 배열 모듈에 입력되는데 그림 1의 h 화소를 구하기 위해 입력이 전치되어야 하기 때문이고 모듈의 블록도는 그림 6과 같다.

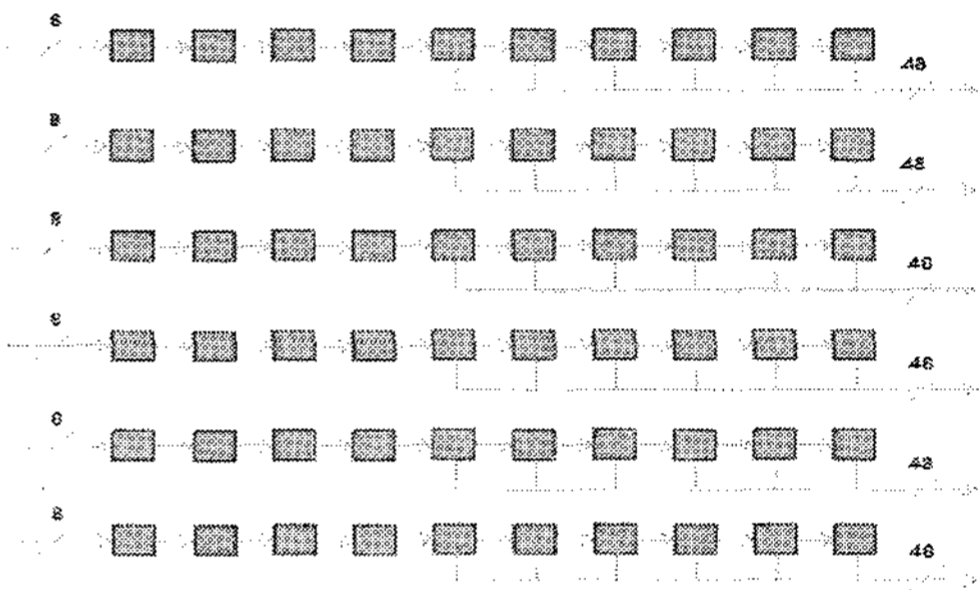


그림 6. 6x10 전치 배열 모듈

6x10 전치 배열 모듈을 통과하면 그림 1의 cc, dd, h, m, ee, ff 화소를 구하기 위해 6-tap 필터로 입력되어 결과가 레지스터에 저장되고 곧바로 5x10 전치 배열에 저장되어 있던 화소들이 j 화소를 구하기 위해 6-tap 필터에 입력되며 그림과 같이 예측 되어져야 할 화소는 다섯 행이므로 각각의 h와 j 화소를 구하기 위해 3개의 6-tap 필터를 사용하면 10 사이클씩 20 사이클이 소요된다.

마지막으로 평균 필터인 2-tap 필터를 사용하여 1/4 화소 정밀도 예측신호를 생성하는데 순차적으로 입력을 받아 6-tap 필터가 수행되는 동시에 2-tap 필터가 수행된다. 그림 4의 보상 범위를 4x4 씩 25 블록으로 나누어 그림 7과 같이 12개의 2-tap 필터를 통하여 한 사이클에 하나의 블록

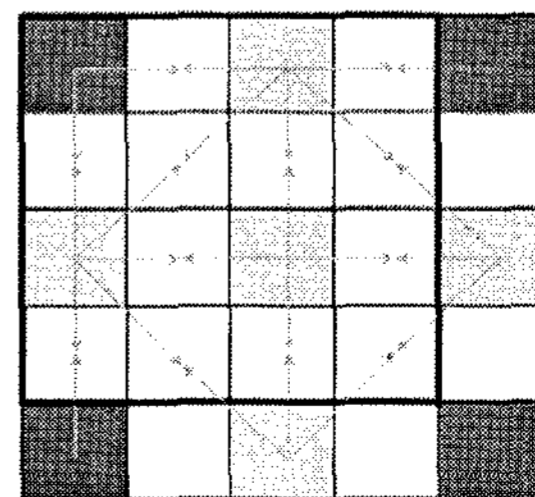


그림 9. 2-tap 필터의 적용

씩을 처리함으로써 25 사이클이 필요하게 된다. 이를 타이밍 차트로 나타낸 것이 그림 8과 같으며 4x4 블록 전체의 보간 수행에 40 사이클이 소요된다.

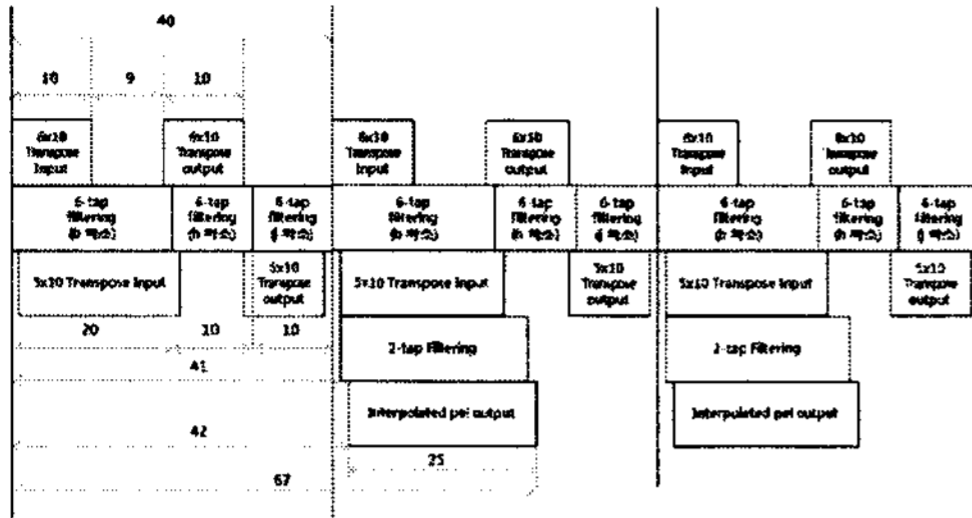


그림 10. 움직임 보상기의 타이밍 구조

IV. 성능 평가

하지만 이전에 제안되어있던 복호화기의 움직임 보상에 관한 구현들과 성능 비교를 하기에는 다소 무리가 있다. 왜냐하면 복호화기에서는 보간해야하는 화소 범위가 다르고 움직임 벡터를 생성해야하며 참조프레임 메모리에 직접 접근해야 하지만 부호화기에서는 움직임 추정기에 종속적이어서 오직 보간만을 수행하기 때문이다. 따라서, 움직임 보상에서 가장 핵심이 되는 6-tap FIR 필터를 얼마나 사용하였는지와 보간화소를 예측하기까지의 최악의 경우 수행 사이클을 비교하면 아래 표 1과 같다.

표 1. 움직임 보상기의 성능 비교

	[5]	[6]	본 논문
구분	디코더	디코더	인코더
보간화소/4x4	16x16	16x16	20x20
사이클/4x4	27	60	40
6-tap 필터	13	2	3

표 1 에서 볼 수 있듯이 [5]의 구조는 4x4 블록을 수행하기 위해 27 사이클이 걸리지만 6-tap 필터가 13개가 사용되었고 수직 필터와 수평 필터가 따로 존재하여 필터의 이용률은 낮다고 할 수 있다. 반면에 [6]의 구조에서 필터의 개수와 수행 사이클의 비율은 일치한다고 할 수 있지만 보간을 수행해야하는 화소수가 144개의 차이를 가지므로 보다 효율적이라고 할 수 있다.

V. 결 론

본 논문은 H.264/AVC 부호화기에서 적용 가능한 움직임 보상에 적합한 구조를 제안한다. 본 논문에서 제안하는 움직임 보상기는 RTL레벨의

VHDL로 구현하고 PCI 인터페이스를 통하여 보드 레벨에서 검증하였다. 전치배열을 사용하여 3개의 휘도 6-tap 필터는 입력이 지속되는 한 쉬지 않고 동작한다. 6-tap 필터에 의해 1/2 화소가 예측 되면 파이프라인 기법을 통하여 다음 주기에 1/4 화소를 2-tap 필터를 통하여 예측하여 출력되어지도록 구현함으로써 높은 하드웨어 이용률을 갖게 되었다. 하나의 매크로블록을 처리하는데 640 사이클이 소요되며 이는 실시간 처리 응용 분야에 적합할 것으로 사료된다. 이 기술을 바탕으로 장래에 움직임 추정 모듈과 결합한 인터 예측 부호화기를 제안하여 성능 효율성을 향상시킬 것이다.

사 사

이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-511-D00249).

참고문헌

- [1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, May 2003, Joint Video Team.
- [2] ISO/IEC JTC1/SC29WG11, "a Computational Complexity Comparison of MPEG4 and JVT Codecs," Doc. M8696, July. 2002.
- [3] 박기현, "코덱의 세계로의 초대," 홍릉과학출판사, 2006
- [4] 카도노 신야, "H.264/AVC 비디오 압축 표준," 홍릉과학출판사, 2005
- [5] H. Tseng, C. Chang and Y. Lin, "A Hardware Accelerator for H.264/AVC Motion Compensation," IEEE Signal Processing System Design and Implementation pp. 214-219, Nov. 2005.
- [6] S. Wang, T. Lin, T. Liu and C. Lee, "A Motion Compensator with Parallel Memory for H.264 Advance Video Coding," IEEE ISCAS Circuit and System, vol. 5, pp. 4558-4561, May 2005.