

이중 채널 파이프라인 구조의 H.264용 고성능 보간 연산기 설계

Design of High Performance Dual Channel Pipelined Interpolators for H.264 Decoder

Chanho Lee*

이찬호*

Abstract

The motion compensation is the most time-consuming and complex unit in the H.264 decoder. The performance of the motion compensation is determined by the calculation of pixel interpolation. The quarter-pixel interpolation is achieved using 6-tap horizontal or vertical FIR filters for luminance data and bilinear FIR filters for chroma data. We propose the architecture for interpolation of luminance and chroma data in H.264 decoders. It is composed of dual-channel pipelined processing elements and can interpolate integer-, half- and quarter-pixel data. The number of the processing cycles is different depending on the position. The processing elements are composed of adders and shifters to reduce the complexity while the accuracy of the pixel data are maintained. We design interpolators for luminance and chroma data using Verilog-HDL and verify the function and performance by implementing using an FPGA.

요약

비디오 압축 코덱으로 널리 이용되는 H.264 표준의 움직임 보상기는 디코더에서 가장 복잡하고 연산시간이 많이 소모되는 유닛이다. 이러한 움직임 보상기의 성능을 결정하는 연산기가 보간 연산기(interpolator)이다. 1/4 보간 연산을 위해 휘도 픽셀은 6 탭 FIR 필터 연산이, 색차 픽셀은 2 탭 FIR 필터 연산이 필요하다. 본 논문에서는 이러한 복잡한 연산을 효과적으로 수행하는 고성능 보간 연산기 구조를 제안한다. 제안하는 구조는 이중 채널과 파이프라인 방식의 연산기로 구성되고 정수, 1/2, 1/4 보간 연산을 모두 수행할 수 있다. 연산기는 복잡도를 줄이기 위해 덧셈기와 쉬프터만으로 구성되면서도 반올림 오차가 전파되지 않도록 하여 연산결과의 정확도를 유지할 수 있다. 또한 보간 연산기의 구조는 연산기의 수를 조절하여 성능과 면적을 조절할 수 있다. 제안된 구조에 따라 휘도 및 색차 데이터를 위한 보간 연산기를 각각 Verilog-HDL을 이용하여 설계하여 동작과 성능을 검증하였다.

Key words : H.264, motion compensation, interpolation, decoder

1. 서론

H.264는 ISO/IEC와 ITU-T의 JVT(Joint Video Team)에 의해 개발된 동영상 압축 표준으로

MPEG-4 part 10 AVC라고도 불린다[1]. H.264의 압축률은 MPEG-4 ASP(Advanced Simple Profile)보다 25-40% 정도 더 좋다[2]. H.264 표준은 가변 블록 크기와 1/4 픽셀 보간, 다중 참조 프레임, 그리고 CAVLC (Context-based Adaptive Variable Length Coding)와 CABAC(Context-based Adaptive Binary Arithmetic Coding)와 같은 컨텍스트 기반의 적응적 엔트로피 코딩 등의 새로운 기능을 포함하는데 이들은 높은 압축률과 개선된 화질을 제공한다[1]. 그러나 연산 알고리즘이 복잡하여 매우 많은 연산을 필요로

* 崇實大學校 情報通信電子工學部

(School of Electronic Engineering, Soongsil University)

※ 본 논문은 교육과학기술부의 재원으로 한국연구재단의 기초연구사업(2009-0072664)의 지원을 받아 수행되었으며 IDEC의 SW 지원을 받았습니다.

한다. H.264 표준의 복호 연산의 경우 MPEG-4에 비해 연산 복잡도가 2배 정도 증가하였다. H.264 복호 소프트웨어를 이용한 실행 프로파일 분석 결과에 의하면 움직임 보상기(motion compensation)는 전체 복호 시간 중에서 최대 55%h까지 차지한다. 따라서 움직임 보상기의 구조를 최적화하는 것이 복호기 성능에 매우 큰 영향을 미친다는 것을 알 수 있다[3,4].

H.264 복호기의 움직임 보상기는 참조 프레임의 블록을 이용하여 현재 프레임의 블록을 예측한다. 구조를 보면 움직임 벡터 예측기(motion vector predictor), 주소 발생기와 참조 픽셀 수집기, 그리고 보간 연산기(interpolator)로 구성된다. 주소 생성기는 움직임 벡터를 이용하여 외부 메모리의 참조 픽셀 데이터의 주소를 계산하여 데이터를 읽어올 수 있게 한다. 움직임 보상기의 성능은 외부 메모리에 접근하여 데이터를 읽어오는 사이클 수에 영향을 받으므로 이를 줄이는 것이 중요하다. 휘도 데이터의 경우 보간 연산기는 4x4 서브 블록을 예측하기 위해 9x9 참조 픽셀이 필요하고 움직임 벡터는 1/4 픽셀 위치까지 표현한다. 1/2 픽셀이나 1/4 픽셀 위치의 데이터는 정수 위치의 픽셀 데이터를 이용하여 보간 연산을 통해 계산한다. 즉, 1/2 픽셀 위치의 데이터는 가로나 세로의 6개의 정수 위치 픽셀 데이터를 6탭 FIR 필터를 이용하여 계산하고 1/4 픽셀 위치 데이터는 1/2 픽셀 값과 정수 픽셀값을 이용하여 반올림한 평균값인 클립(Clip) 연산을 통해 얻는다. 색차 데이터는 네 개의 정수 픽셀값을 이용하여 1/8 픽셀값을 바이리니어(bilinear) 필터를 통해 계산한다. 보간 연산은 픽셀의 위치에 따라 최대 7번의 6 탭 필터 연산과 한번의 클립 연산이 필요하므로 데이터를 공급하는 유닛을 포함한 보간 연산기는 움직임 보상기의 성능을 결정한다. 따라서 움직임 보상기의 성능을 개선하기 위해서는 메모리 접근 사이클 수와 보간 연산기의 연산 사이클을 줄이는 것이 중요하다.

본 논문에서는 H.264 복호기의 움직임 보상기를 위한 효율적인 보간 연산기 구조를 제안한다. 제안한 구조는 이중 채널 파이프라인 방식의 연산기(PE: Processing Element)와 클립 유닛을 포함한다. 연산기는 6 탭 FIR 필터와 바이리니어 FIR 필터의 동작을 하는데 곱셈기 없이 덧셈기와 쉬프트만으로 구성된다. 또한 연산기의 추가만으로 확장 가능하도록 하여 필요한 성능에 맞추어 보간 연산기 구성이 가능하다. 제안한 구조는 휘도 및 색차 데이터의 보간 연산에 적용 가능하다. 제안된 구조에 따라 휘도 및 색차 데이터를 위한 연산기를 설계하고 이를 이용하여 휘도 및 색차 보간 연산기를 각각 구성하였으며 동작과 성능을 검증하였다.

II. 보간 연산 방법

1/2 픽셀 위치의 휘도 데이터는 정수 위치의 픽셀이나 다른 1/2 위치의 픽셀을 이용하여 계수가 [1, -5, 20, 20, -5, 1]인 6 탭 FIR 필터 연산을 통해 구한다. 1/4 위치의 픽셀은 정수 위치 또는 1/2 위치의 픽셀 데이터를 이용하여 클립 함수 연산을 통해 계산한다. 그림 1은 정수와 1/2, 그리고 1/4 위치의 휘도 픽셀 데이터 성분을 보여준다. 그림에서 대문자는 정수 위치를 나타내고 a, h, j, m, s와 aa 등의 이중 소문자는 1/2 위치의 픽셀을 나타낸다. 나머지는 1/4 위치의 픽셀이다. 각 픽셀값을 계산하는 방법은 그 위치에 따라 조금씩 차이가 있다. 6 탭 FIR 필터와 클립 연산기 하나로 구성된 보간 연산기의 경우, j 픽셀을 계산하기 위해서는 8 사이클이 필요하고 1 사이클에 연산 결과를 얻기 위해서는 7개의 6 탭 FIR 필터가 필요하다. 이 경우 면적이 증가하고 동시에 입력값을 공급하기 위한 주변 회로가 복잡해지며 최대 동작 주파수는 감소한다.

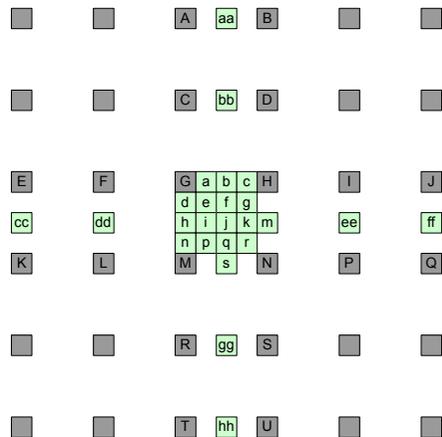


Fig. 1. Integer-, half-, and quarter-position of luminance pixels

그림 1. 정수, 1/2, 1/4 위치의 휘도 픽셀 데이터

색차 데이터의 1/8 위치는 그림 2에 나타난 바와 같이 dx와 dy에 의해 표현된다. A, B, C, D는 정수 위치의 색차 픽셀이고 dx와 dy는 0에서 8 사이의 정수값을 갖는다. 이 때 1/8 위치의 색차 픽셀값은 다음 식에 의해 계산된다.

$$x(xc, yc) = (A(8 - dx)(8 - dy) + Bdx(8 - dy) + C(8 - dx)dy + Dxdy + 32) >> 6 \quad (1)$$

색차 데이터의 1/8 위치 픽셀값은 휘도 데이터와는 달리 하나의 식에 의해 계산되어 더 간단해 보인다. 그러나 필터 계수가 상수가 아닌 변수이므로 곱셈기 없이 구현하는 경우 더 복잡한 구조를 갖는다.

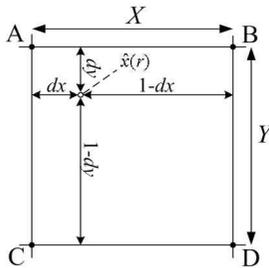


Fig. 2. 1/8-position of chroma pixels
 그림 2. 색차 데이터의 1/8 위치 픽셀

앞서 언급한 바와 같이 휘도와 색차 데이터를 연산하는 방법은 서로 다르지만, 휘도와 색차 데이터를 동시에 계산할 수 있는 보간 연산기 구조를 설계하는 것이 어려운 일은 아니다. 이러한 보간 연산기를 이용한다는 것은 하나의 연산기로 휘도와 색차 데이터를 순차적으로 계산함을 의미하고, 영상 크기가 작고 동작 주파수가 높아 연산 사이클에 여유가 있는 경우가 아니라면 보간 연산기는 많은 FIR 필터를 포함하여 빠른 속도로 결과를 얻어야 한다. 이 경우 하나의 보간 연산기가 두 개의 휘도 및 색차 전용 보간 연산기보다 비효율적일 수 있다. 따라서, 본 논문에서는 휘도와 색차 데이터를 위한 전용 보간 연산기 구조를 제안한다.

III. 보간 연산기 구조

1. 연산기(PE) 구조

앞에서 언급한 바와 같이 휘도 픽셀에 대한 보간 연산의 경우 픽셀의 위치에 따라 연산 방법이 조금씩 차이가 있다. 휘도 보간 연산기의 경우 이를 반영하여 모든 위치의 픽셀에 대한 연산이 가능하도록 하기 위해 연산기 PE-L을 이용한다. 이들은 기본적으로 6 탭 필터로 PE-L 연산기는 그림 1의 E, F, G, H, I, J의 정수 위치 픽셀값을 입력으로 받아 1/2 위치 픽셀값을 그림 3과 같이 덧셈기와 쉬프트만을 이용하여 계산하거나 정수값 G 또는 H를 선택하여 레지스터를 통해 출력한다. j는 aa1, bb1, b1, s1, gg1, hh1에 해당하는 값을 받아 PE-L을 한 번 더 이용하여 계산한다. PE-L에 포함된 버퍼 레지스터는 단순히 필터링

된 값을 한 사이클 지연시키는 역할을 하는 것이 아니고 보간 연산을 위해 필요한 이전 값을 별도로 저장하여 현재의 출력과 이전 값중 필요한 것을 선택하여 내보낼 수 있도록 설계된 데이터 처리기이다.

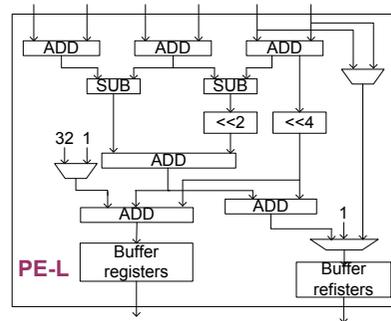


Fig. 3. The proposed structure of PEs for luminance pixels

그림 3. 휘도 픽셀 연산을 위한 연산기 구조

색차 보간 연산의 경우 픽셀 위치에 따른 연산 방법의 차이는 없으나 필터링 계수가 변수이므로 연산기에 곱셈기를 포함하거나 곱셈기가 없는 경우 매우 복잡한 구조를 갖는다. 따라서 색차 보간 연산의 경우 하드웨어 복잡도를 낮추는 것이 중요하다. 그림 4에 나타난 것처럼 덧셈기와 쉬프트만을 사용하고 복잡도를 낮춘 색차 픽셀 연산을 위한 연산기인, PE-C는 그림 2의 정수 픽셀 값인 A, B, C, D 값에 0부터 8 까지의 값을 곱할 수 있다. 식(1)의 A(8-dx)(8-dy)를 계산할 때 첫 번째 PE-C는 정수 위치의 색차 데이터를 입력으로 받아 A(8-dx)를 레지스터를 통해 출력한다. 이 레지스터 출력은 두 번째 PE-C로 입력되어 A(8-dx)(8-dy)를 얻을 수 있다. 따라서 식 (1)의 하나의 항을 계산하기 위해서는 PE-C를 두 번 거쳐야 하고 최종 결과는 연산기 쌍을 네 번 통과해야 얻을 수 있다. PE-C의 버퍼 레지스터도 PE-L과 마찬가지로 반복된 PE 연산에서 필요한 값을 저장하고 선택하는 기능이 포함되어 있다.

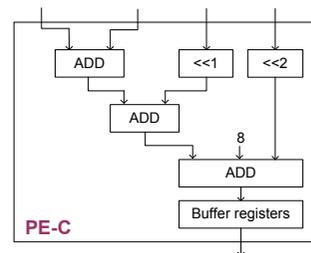


Fig. 4. The proposed structure of PEs for chroma pixels
 그림 4. 색차 픽셀 연산을 위한 연산기 구조

2. 보간 연산기(Interpolator) 구조

그림 5는 휘도 픽셀 데이터를 계산하기 위한 보간 연산기의 제안하는 구조를 보여 준다. 그림 5(a)의 보간 연산기는 두 개의 PE-L 연산기와 하나의 클립 유닛으로 구성된 연산 파이프라인 두 개가 비대칭으로 배치되어 모든 위치의 픽셀을 계산한다. 첫 번째 PE-L은 정규화 전과 후의 1/2 위치 픽셀값을 연산 후 저장하고 두 번째 PE-L은 1/2과 1/4 위치 픽셀의 데이터를 연산 후 저장한다. 저장된 값은 소수점 이하의 값까지 포함하고 마지막 단의 클립 유닛에서 소수점 이하의 값을 제거하여 반올림 오차가 전파되지 않도록 하여 영상 데이터의 정밀도를 최대한 유지하도록 하였다. 두 개의 파이프라인 연산 채널은 그림 1의 1/4 위치 픽셀값인 e, p, g, r을 계산할 때는 함께 연동하여 동작하고 나머지 위치의 픽셀에 대해서는 독립적으로 동작하여 동시에 두 개의 픽셀을 한 번에 계산한다. 따라서 제안된 구조는 픽셀 위치에 따라 한 사이클에 하나 또는 두 개의 픽셀을 처리하거나 11 사이클에 8개의 픽셀을 처리할 수 있다. 그림 5(a)는 11 사이클에 8개의 픽셀을 처리하는 경우가 많이 발생하면 성능이 저하되지만 단순한 구조를 가지고 있어 적은 면적을 차지한다. 고성능이 필요한 경우에는 그림 5(b)와 같이 두 개의 PE-L 연산기와 하나의 클립 유닛으로 구성된 연산 파이프라인 두 개가 대칭으로 배치된 구조를 이용하여 한 사이클에 모든 위치의 픽셀을 하나 또는 두 개씩 연산할 수 있다. 또한 이 구조를 병렬로 배치하면 성능을 두 배로 증가시킬 수 있다. 이 구조는 PE-L 연산기의 수는 같으나 신호 연결과 제어가 복잡해지고 면적이 증가하는 단점이 있다. 따라서 요구하는 성능에 따라 그림 5의 (a) 또는 (b) 구조를 선택하여 사용할 수 있다.

하나의 PE-L 연산기는 보간 연산에 필요한 연산 기능과 저장 기능을 모두 가지고 있으므로 기본적으로 PE-L 하나와 클립 연산기 하나로도 모든 보간 연산이 가능하다. 따라서 영상 크기가 작아 고성능이 필요 없는 경우에는 하나의 PE-L과 클립 연산기로 보간 연산기를 구성하여 면적을 크게 줄일 수 있다. 또한 두 개의 PE-L 연산기와 클립 연산기로 구성되는 단일 파이프라인 구조도 가능하다. PE-L 연산기의 수를 줄이는 경우 면적은 크게 감소하지만 제어 신호가 복잡해지는 단점이 있다.

4:2:0 영상 포맷의 경우 16 X 16 픽셀의 휘도 데이터에 비해 두 개의 4 X 4 픽셀 데이터가 대응된다. 제안하는 보간 연산기 구조는 하나의 색차 픽셀 데이터를 2 사이클에 걸쳐 처리한다. 즉, 식(1)에 나타난

모든 곱셈과 덧셈/뺄셈을 색차 보간 연산기가 두 번 동작하여 처리한다. 이는 휘도 보간 연산기에 비해 최대 1/4 정도로 성능이 떨어지지만 휘도와 색차 데이터의 수의 차이를 고려하면 오히려 색차 데이터 처리가 더 빠르다. 그림 6은 색차 픽셀 연산기인 PE-C를 이용한 색차 보간 연산기의 구조를 보여준다. 기본 구조는 휘도 보간 연산기와 마찬가지로 이중 채널 파이프라인 구조를 가지고 있고 이 구조를 병렬로 배치하여 보간 연산기 수만큼의 성능 향상을 얻을 수 있다. 반대로 하나의 파이프라인만을 사용하면 성능은 감소하지만 면적을 줄일 수 있다. 색차 데이터는 휘도와는 달리 하나의 식으로 계산되므로 휘도 보간 연산기에 비해 구조가 간단하다. 하이 프로파일의 4:4:4 영상 포맷의 경우에는 색차 보간 연산기 4 - 8 개를 연결하여 사용하면 된다.

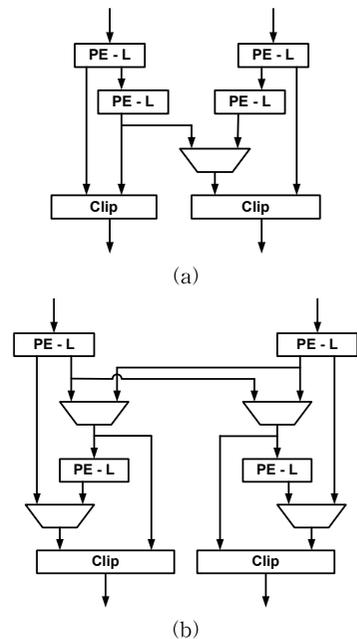


Fig 5. The proposed architecture of luminance interpolator for (a) small area and (b) high performance

그림 5. 제안하는 이중 채널 휘도 보간 연산기 구조 (a) 저면적 구조 (b) 고성능 구조

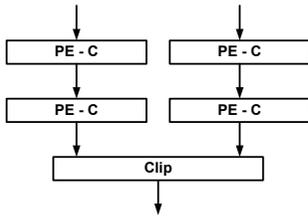
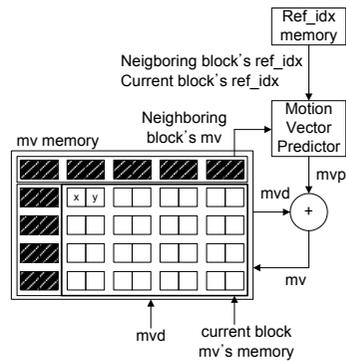


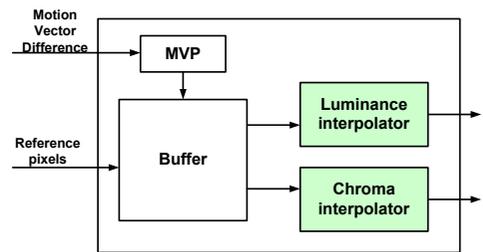
Fig. 6. The proposed architecture of chroma interpolator
 그림 6. 제안하는 이중 채널 색차 보간 연산기 구조

그림 7은 제안된 보간 연산기를 이용하여 구현된 움직임 보상기의 구조로 움직임 벡터 예측기(MVP), 버퍼, 휘도 및 색차 보간 연산기와 제어기로 구성된다. MVP는 그림 7(a)에 나타난 바와 같이 참조 메모리에서 참조 인덱스 값을 읽어와 내부에 저장된 이웃한 블록의 움직임 벡터를 이용하여 mvp 값을 구하고 이를 현 블록의 mvd 값과 더하여 움직임 벡터를 얻는다. 이 움직임 벡터를 계산하여 보간 연산에 이용될 참조 픽셀 데이터의 주소를 계산한다. 움직임 벡터 계산에 필요한 데이터는 매크로 블록 단위로 전달되므로 이 값들을 미리 계산하여 다음 보간 연산에 필요한 4 X 4 서브 블록의 참조 데이터를 미리 읽어오면 움직임 보상기의 성능을 크게 개선할 수 있다. 버퍼는 보간 연산에 필요한 참조 픽셀 데이터를 읽어와 저장하고 보간 연산기에 공급하는 역할을 한다. 휘도 및 색차 보간은 동시에 진행되므로 버퍼는 필요한 데이터를 끊임없이 읽어와 두 보간 연산기에 교대로 데이터를 공급한다. 일반적으로 색차 보간 시간이 짧기 때문에 버퍼의 데이터 공급은 휘도 보간 연산기에 우선적으로 이루어진다.

제안된 보간 연산기는 휘도 및 색차 데이터를 위한 구조를 서로 다르게 함으로써 구조를 단순화하고 두 보간 연산기가 동시에 동작하므로 처리속도를 높일 수 있다. 움직임 보상기에서 보간 연산이 가장 시간이 오래 걸리는 과정임을 고려할 때 이는 전체 성능 향상에 상당한 기여를 한다. 그림 8은 휘도와 색차 연산이 교대로 일어날 때 연산 파이프라인을 보여준다. 휘도 및 색차 데이터 읽기와 쓰기는 상대 연산기의 연산 동작 구간에서 이루어지고 연산 사이클은 일반적으로 휘도 데이터 보간 연산을 위한 읽기, 쓰기, 연산이 모두 색차 연산에 비해 더 많은 사이클을 소모하므로 휘도 연산에 의해 결정된다.



(a)



(b)

Fig. 7. The architecture of motion compensation unit. (a) MVP (b) MC

그림 7. 보간 연산기를 이용한 움직임 보상기의 구조 (a) MVP 구조 (b) 움직임 보상기 구조

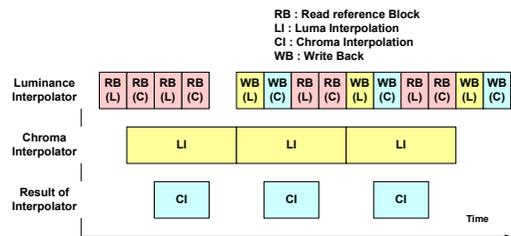


Fig. 8. Pipeline sequence of interpolation process
 그림 8. 휘도 및 색차 보간 연산의 파이프라인 순서

IV. 구현 결과

제안된 구조에 따라 Verilog-HDL을 이용하여 휘도 및 색차 보간 연산기를 설계하고 0.18um CMOS 셀 라이브러리를 이용하여, 비교를 위해 100MHz를 목표로 합성하였다. 표 1은 합성 및 비교 결과를 보여준다. 휘도와 색차 보간 연산기를 모두 포함한 보간 연산기는 12,762 게이트를 차지하는데 이는 기존 결과와 비교하여 40% 정도 감소한 것이다. 최대 동작 주파수는 200MHz로 성능을 2배까지 증가시킬 수 있다.

설계된 결과물은 FPGA에 구현하여 동작을 검증하였고 그림 9에 검증 결과가 나타나 있다.

Table 1. The comparison of synthesized results

표 1. 0.18um CMOS 공정을 이용한 합성 및 비교 결과

	[2]	[5]	제안 구조
면적 [게이트 수]	21,506	20,686	12,762
동작주파수 [MHz]	125	100	100 (200)
공정	0.18um	0.18um	0.18um

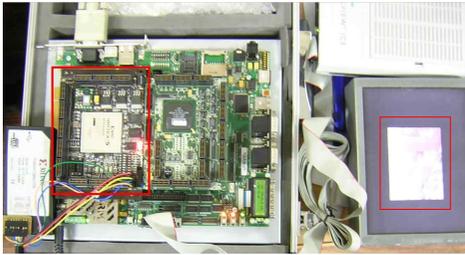


Fig. 9. Verification result of the interpolator using FPGA
 그림 9. FPGA를 이용한 보간 연산기 검증 결과

V. 결론

본 논문에서는 H.264 압축 영상 표준의 복호 과정에서 가장 복잡한 움직임 보상기의 핵심 연산기인 보간 연산기의 하드웨어 구현 복잡도를 줄이면서 성능을 높일 수 있는 구조를 제안하였다. 움직임 보상기의 보간 연산에서 휘도와 색차 연산 방식이 달라 하나의 연산기로 보간 연산을 할 경우 연산기가 매우 복잡해진다. 이를 막기 위해 휘도와 색차 보간 연산기를 분리하여 각 연산기의 구조를 단순화하고 덧셈기와 쉬프트만큼을 이용하여 구성하면서 반올림 오차가 전파되지 않도록 하여 연산 정밀도를 높였다. 또한, 휘도와 색차 연산기가 분리되면서 동시에 보간 연산이 가능해져 성능도 향상시킬 수 있다. 휘도 보간 연산기는 픽셀 위치에 따라 한 사이클에 하나 또는 두 개의 픽셀 연산을 하거나 11 사이클에 8 개의 픽셀 연산을 진행할 수 있다. 색차 보간 연산기는 2 사이클에 하나의 픽셀을 처리하는데 필요시 병렬 또는 순차 연결을 통해 데이터 처리 속도를 증가시킬 수 있다. 제안된 구조에 따라 휘도 및 색차 보간 연산을 위한 연산기를 Verilog-HDL을 이용하여 설계하고 0.18um CMOS 표준 셀 라이브러리를 이용하여 합성하였다. 합성 결과 12,762 게이트를 사용하여 기존 결과에 비해 60%의 면적으로 구현이 가능하며 최대 동작 주파수는 200MHz이다. 설계된 보간 연산기는 FPGA에 구현되어 그 동작을 검증하였다.

참고문헌

[1] Joint Video Team, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, May 2003.
 [2] Chuan-Yung Tsai, Tung-Chien Chen, To-Wei Chen, Liang-Gee Chen, "Bandwidth Optimized Motion Compensation Hardware Design for H.264/AVC HDTV Decoder" Proceedings of 48th Midwest Symposium on Circuits and Systems, Vol. 2, pp. 1199-1202, Aug. 7-10, 2005.
 [3] Wen-Nung Lie, Han-Ching Yeh, Lin, T.C.-I., Chien-Fa Chen, "Hardware-efficient computing architecture for motion compensation interpolation in H.264 video coding", Proceedings of IEEE International Symposium on Circuits and Systems, Vol. 3, pp. 2136-2139, May 23-26, 2005.
 [4] Ronggang Wang, Mo Li, Jintao Li, Yongdong Zhang, "High throughput and low memory access sub-pixel interpolation architecture for H.264/AVC HDTV decoder", IEEE Transactions on Consumer Electronics, Vol. 51, pp. 1006-1013, Aug. 2005.
 [5] Sheng-Zen Wang, Ting-An Lin, Tsu-Ming Liu, Chen-Yi Lee, "A new motion compensation design for H.264/AVC decoder" IEEE International Symposium on Circuits and Systems, Vol. 5, pp. 4558-4561, May 23-26, 2005.

저 자 소 개

이찬호 (정회원)



1987년 : 서울대학교 전자공학과 졸업 (공학사)
 1989년 : 서울대학교 대학원 전자공학과 (공학석사)
 1994년 : University of California Los Angeles, Dept. of Electrical Engr. (공학박사)
 1995년 3월~현재 : 송실대학교 정보통신전자공학부 교수
 <주관심분야> SoC on-chip-network, 3D 그래픽 프로세서 설계, 채널코덱의 구현, SoC 설계방법론, H.264 codec 구현