

# H.264 움직임 예측을 위한 Luma와 Chroma 부화소 보간기 설계

이 선 영<sup>†</sup> · 조 경 순<sup>††</sup>

요 약

본 논문은 H.264 움직임 예측을 위해 휘도 성분과 색차 성분의 부화소를 생성하는 효율적인 부화소 보간기 회로 설계에 대해 기술한다. 제안된 구조를 기반으로 한 회로는 보간 연산을 위해 입력 데이터를 버퍼링하지 않고 수평, 수직, 대각선의 부화소 보간을 병렬로 처리한다. 휘도 성분에 대한 1/2 화소, 1/4 화소 보간과 색차 성분에 대한 1/8 화소 보간을 동시에 처리하여 회로 성능을 더욱 개선하였다. 회로 크기를 줄이기 위해 본 논문에서는 병렬로 보간 연산을 처리하는데 필요한 모든 중간 데이터를 레지스터 대신 내부 SRAM에 저장하였다. 제안된 구조를 레지스터 전달 수준의 회로로 기술하였고, FPGA 보드에서 동작을 검증하였다. 또한 구현된 회로를 130nm CMOS 표준 셀 라이브러리를 이용하여 게이트 수준의 회로로 합성하였다. 합성된 회로의 크기는 20,674 게이트이고 최대 동작 주파수는 244MHz이다. 회로에 사용된 SPSRAM의 전체 크기는 3,232 비트이다. 구현된 회로는 논리 게이트와 SRAM을 포함하여 다른 논문에서 제안한 회로에 비해 크기가 작고 성능도 우수하다.

키워드 : 움직임 예측, H.264, 보간, 부화소, 회로 구조

## Design of Luma and Chroma Sub-pixel Interpolator for H.264 Motion Estimation

Seonyoung Lee<sup>†</sup> · Kyeongsoon Cho<sup>††</sup>

### ABSTRACT

This paper describes an efficient design of the interpolation circuit to generate the luma and chroma sub-pixels for H.264 motion estimation. The circuit based on the proposed architecture does not require any input data buffering and processes the horizontal, vertical and diagonal sub-pixel interpolations in parallel. The performance of the circuit is further improved by simultaneously processing the 1/2-pixel and 1/4-pixel interpolations for luma components and the 1/8-pixel interpolations for chroma components. In order to reduce the circuit size, we store the intermediate data required to process all the interpolations in parallel in the internal SRAM's instead of registers. We described the proposed circuit at register transfer level and verified its operation on FPGA board. We also synthesized the gate-level circuit using 130nm CMOS standard cell library. It consists of 20,674 gates and has the maximum operating frequency of 244MHz. The total number of SPSRAM bits used in our circuit is 3,232. The size of our circuit (including logic gates and SRAM's) is smaller than others and the performance is still comparable to them.

Keywords : Motion Estimation, H.264, Interpolation, Sub-pixel, Circuit Architecture

### 1. 서 론

DMB (digital multimedia broadcasting)와 IPTV(internet protocol television) 등에서 사용되는 비디오 압축 표준인 H.264는 ISO/IEC MPEG과 ITU-T VCEG에서 공동 구성한 JVT (Joint Video Team)에 의해 표준화되었다[1].

H.264의 가장 중요한 목표 중의 하나는 더 좋은 압축 효율을 달성하는 것이다. 실제로 H.264는 기존 표준들보다 동일한 영상 화질에서 50% 이상의 비트율 감소 효과를 얻을 수 있다[2]. 이러한 개선 효과를 달성함에 있어서 정수 화소와 부화소 단위의 움직임 예측이 기여한 바가 가장 크다. H.264는 휘도 (luminance) 성분에 대한 1/2 화소 보간을 수행하기 위해 6탭의 FIR (Finite Impulse Response) 필터를 사용하고 1/4 화소 보간을 수행하기 위해 양선형 필터를 사용한다. 색차 (chrominance) 성분에 대해서는 1/8 화소 보간을 위해 선형 필터를 사용한다. 정수 화소로부터 보간된

※ 이 논문은 2011년도 한국외국어대학교 교내학술연구비의 지원에 의한 것임.

† 정 회 원 : 전자부품연구원 융합신호SoC연구센터 선임연구원

†† 정 회 원 : 한국외국어대학교 전자공학과 정교수

논문접수: 2011년 3월 29일

수정일: 1차 2011년 5월 17일

심사완료: 2011년 5월 18일

부화소를 이용하는 부화소 움직임 예측은 정수 화소 움직임 예측만을 수행하는 MPEG-2와 비교하여 영상 화질을 평균 4 dB 정도 개선시킨다. 그러나 부화소 움직임 예측은 H.264 비디오 압축 과정에서 전체 계산량의 25%, 메모리 접근의 50%를 점유한다[3].

부화소 보간 회로 구조에 대한 다양한 연구들이 진행되었다. 논문 [4,5]는 보간 회로의 처리 속도를 높이기 위해 알고리즘을 단순화시켰지만 화질을 떨어뜨리는 문제가 있다. 논문 [6]은 구현된 회로가 복잡하여 HDTV (High-definition Television)와 같은 고화질 영상에 대해 고속 처리가 어려운 문제가 있다. 또한 이전에 제안된 보간 회로들은 FIR 연산을 위해 일반적으로 필터 탭 수만큼의 입력 데이터를 회로 내부에 저장한다. 이를 위해 추가적인 저장 시간이 필요하고 메모리 접근 방법이 복잡해진다. 본 논문에서 제안된 구조는 입력되는 데이터를 내부 레지스터에 저장하는 동작을 제거하고 수평, 수직, 대각선 방향 부화소 보간 동작을 병렬로 처리하여 고성능으로 동작하도록 하였다. 제안된 회로는 휘도 성분에 대한 1/2 화소, 1/4 화소 보간과 색차 성분에 대한 1/8 화소 보간을 동시에 처리하여 성능을 더욱 개선하였다. 본 논문에서는 모든 부화소 보간을 병렬로 처리하기 위해 요구되는 중간 데이터를 저장하는데 필요한 내부 SRAM (static random access memory)을 효율적으로 재사용하여 회로 크기를 줄였다. 제안된 부화소 보간 회로는 Verilog HDL (hardware description language)을 이용하여 RTL (register transfer level)로 기술되었으며 동작을 Xilinx Virtex4 LX60 FPGA를 사용하여 검증하였다. RTL 회로는 130nm 표준 셀 라이브러리를 이용하여 게이트 수준 회로로 합성하였고 그 결과 다른 연구들과 비교하여 좋은 성능을 보임을 확인하였다. 본 논문은 다음과 같이 구성된다. 2장에서는 휘도 성분과 색차 성분에 대한 부화소 보간 알고리즘에 대해 설명하고 3장에서는 본 논문에서 제안한 1/2, 1/4, 1/8 부화소 보간 회로의 구조에 대해 설명한다. 4장은 제안한 구조에 대한 구현 결과에 대해 기술하고 5장에서 결론으로 본 논문을 마무리한다.

## 2. 부화소 보간 알고리즘

### 2.1 휘도 성분 보간

움직임 예측에서 1/2 화소 보간과 1/4 화소 보간은 정수 화소 움직임 예측만을 수행하는 것에 비해 좋은 성능 개선 효과를 얻게 하므로, H.264에서는 1/2 화소와 1/4 화소 단위의 움직임 벡터 정확도를 지원한다.

(그림 1)은 정수 화소들과 보간된 1/2 화소들을 설명하고 있다. 각 1/2 화소는 1차원 6탭 FIR 필터를 이용하여 인접한 6개 정수 화소로부터 보간된다. FIR 필터 계수들은 1/32, -5/32, 20/32, 20/32, -5/32, 1/32이다. 1/2 화소 보간 연산식의 일부를 식 (1)에 나타내었다. A~U는 정수 화소들이고 나머지는 보간된 1/2 화소들이다.

수평 방향 1/2 화소 (aa, bb, b, s, gg, hh)들은 동일한 수

평 방향에 있는 6개 정수 화소들로부터 보간된다. 유사한 방법으로 수직 방향 1/2 화소 (cc, dd, h, m, ee, ff)들은 수직 방향에 있는 6개 정수 화소들로부터 보간된다. 대각선 방향 1/2 화소 j는 수평방향으로 cc, dd, h, m, ee, ff를 이용하거나 또는 수직방향으로 aa, bb, b, s, gg, hh를 이용하여 보간할 수 있다.

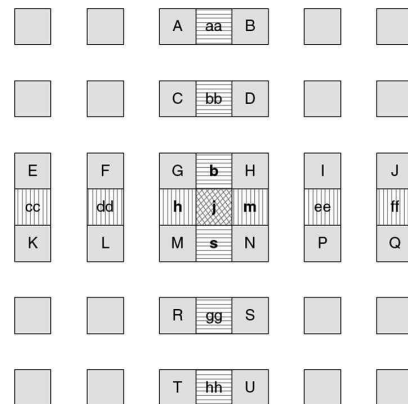
1/4 화소는 (그림 2)에 나타난 것과 같이 1차원 양선형 필터를 이용하여 두 인접한 화소 (정수 화소 또는 1/2 화소)들로부터 보간된다. 이 그림에서 정수 화소들은 G, H, M, N 이다. 1/2 화소들은 b, h, j, m, s이다. 보간된 1/4 화소들은 (a, c, i, k), (d, f, n, q), (e, g, p, r)이다. 1/4 화소 보간 연산식의 일부는 식 (2)에 나타내었다.

$$\begin{aligned}
 b &= \text{round}\left(\frac{E - 5 \cdot F + 20 \cdot G + 20 \cdot H - 5 \cdot I + J}{32}\right) \\
 h &= \text{round}\left(\frac{A - 5 \cdot C + 20 \cdot G + 20 \cdot M - 5 \cdot R + T}{32}\right) \\
 j &= \text{round}\left(\frac{cc - 5 \cdot dd + 20 \cdot h + 20 \cdot m - 5 \cdot ee + ff}{32}\right)
 \end{aligned} \tag{1}$$

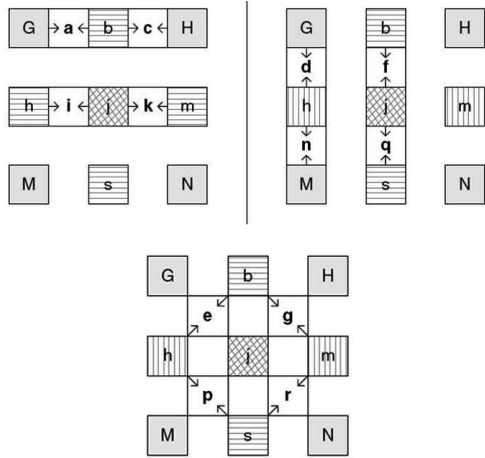
$$\begin{aligned}
 a &= \text{round}\left(\frac{G + b}{2}\right) \\
 d &= \text{round}\left(\frac{G + h}{2}\right) \\
 e &= \text{round}\left(\frac{b + h}{2}\right)
 \end{aligned} \tag{2}$$

### 2.2 색차 성분 보간

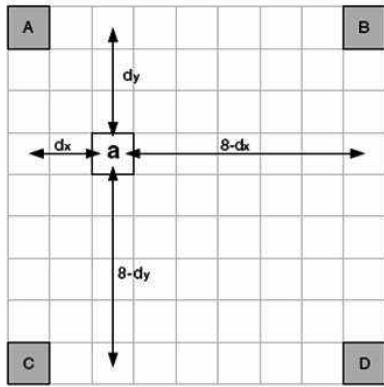
색차 성분들에 대한 보간 화소들은 (그림 3)과 같이 선형 보간을 이용하여 정수 화소 사이에서 1/8 화소 간격으로 생성된다. 각 부화소 'a'는 이웃한 정수 화소들 A, B, C, D의 선형 조합으로 얻어진다. 1/8 화소 보간 연산식은 식 (3)과 같이 정리할 수 있고 dx와 dy는 0~8사이의 정수이다.



(그림 1) 1/2 화소 보간



(그림 2) 1/4 화소 보간



(그림 3) 1/8 화소 보간

$$Z_1 = (8 - dx) \cdot (8 - dy) \cdot A + dx \cdot (8 - dy) \cdot B$$

$$Z_2 = (8 - dx) \cdot dy \cdot C + dx \cdot dy \cdot D$$

$$a = \text{round} \left( \frac{Z_1 + Z_2}{64} \right)$$

(3)

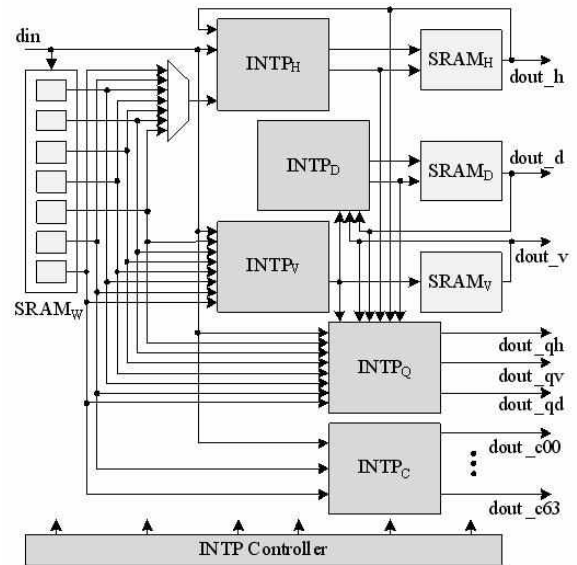
### 3. 제안한 회로 구조

#### 3.1 부화소 보간 회로의 전체 구조 및 동작

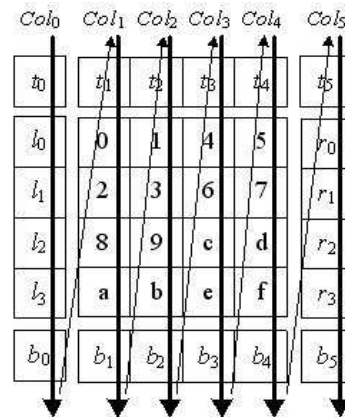
(그림 4)는 본 논문에서 제안한 회로 및 색차에 대한 부화소 보간 회로의 구조를 나타낸 블록 다이어그램이다. 제안한 회로는 1/2 화소 보간기 (INTP<sub>H</sub>, INTP<sub>V</sub>, INTP<sub>D</sub>), 1/4 화소 보간기 (INTP<sub>Q</sub>), 1/8 화소 보간기 (INTP<sub>C</sub>), 1/2 화소들을 저장하는 SRAM (SRAM<sub>H</sub>, SRAM<sub>V</sub>, SRAM<sub>D</sub>), 연산을 처리하기 위한 SRAM (SRAM<sub>W</sub>), 그리고 보간 연산 제어기 (INTP Controller)로 이루어진다. SRAM<sub>H</sub>는 2개의 16x32 비트 단일 포트 SRAM (SPSRAM)으로 구성된다. SRAM<sub>V</sub>는 1개의 32x32 비트 SPSRAM으로 구성된다. SRAM<sub>D</sub>는 2개의 16x32 비트 SPSRAM으로 구성된다. SRAM<sub>W</sub>는 7개의 3x32 비트 SPSRAM으로 구성된다. 제안한 부화소 보간 회

로는 'din'을 통해 4 화소의 32비트 데이터가 입력된다. 입력된 데이터는 중간 버퍼 없이 INTP<sub>H</sub>에서 1/2 화소를 보간하고 데이터의 중복 읽기를 방지하기 위해 SRAM<sub>W</sub>에 데이터가 저장된다. INTP<sub>V</sub>는 SRAM<sub>W</sub>에 저장되어 있던 데이터를 이용하여 보간을 하고 INTP<sub>D</sub>는 INTP<sub>H</sub>와 INTP<sub>V</sub>의 결과를 활용하여 1/2 화소 보간을 처리한다. 1/4 화소나 1/8 화소 보간기는 SRAM<sub>W</sub>의 데이터를 이용하여 보간을 수행한다.

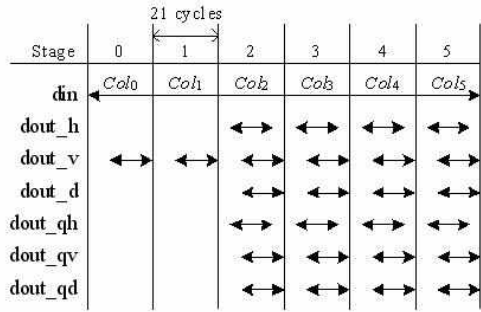
(그림 5)는 1개 매크로블록 (MB)를 보간하기 위해 필요한 36개 4x4 서브 매크로블록 (SMB)를 나타내고 있다. 16개 SMB (0~f)를 보간하기 위해 20개 SMB (*t*<sub>0</sub>~*t*<sub>5</sub>, *l*<sub>0</sub>~*l*<sub>3</sub>, *b*<sub>0</sub>~*b*<sub>5</sub>)가 추가적으로 필요하다. SMB 데이터를 한번 이상 중복해서 읽는 것을 피하기 위해 (그림 5)에 정의된 순서대로 'din'을 통해 보간 회로에 이 데이터들을 입력한다. 'din'을 통해 입력되는 데이터는 *Col*<sub>0</sub>, *Col*<sub>1</sub>, *Col*<sub>2</sub>, *Col*<sub>3</sub>, *Col*<sub>4</sub>, *Col*<sub>5</sub> 순서로 전달되며 SMB 단위로 표현하면 *t*<sub>0</sub>, *t*<sub>1</sub>, *t*<sub>2</sub>, *t*<sub>3</sub>, *t*<sub>4</sub>, *t*<sub>5</sub>, *l*<sub>0</sub>, *l*<sub>1</sub>, *l*<sub>2</sub>, *l*<sub>3</sub>, *b*<sub>0</sub>, *b*<sub>1</sub>, *b*<sub>2</sub>, *b*<sub>3</sub>, *b*<sub>4</sub>, *b*<sub>5</sub>, ... 순서이다.



(그림 4) 제안된 부화소 보간 회로 구조



(그림 5) SMB 데이터 입력 순서



(그림 6) 부화소 보간 동작의 순서

(그림 6)은 1개의 16x16 MB를 보간하는 경우에 대해 스테이지 단위로 보간 회로의 동작을 나타내고 있다. 외부 메모리로부터 'din'을 통해 입력되는 데이터는 SRAM<sub>W</sub>에 저장되고 동시에 INTP<sub>V</sub>는 첫 번째 컬럼 (Col<sub>0</sub>)에 대해 SMB들을 보간한다. INTP<sub>V</sub>의 'dout\_v'으로부터 출력된 보간 결과들은 SRAM<sub>V</sub>에 저장된다. 다른 보간기들이 스테이지 2에서 동작하는 동안 대각선 보간기 INTP<sub>V</sub>는 매 스테이지에서 동작한다.

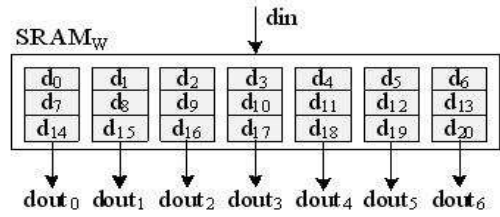
3.2 1/2 및 1/4 부화소 보간 회로의 구조 및 동작

1/2 화소 보간은 6개 정수 화소 데이터들을 요구하기 때문에 2개 SMB (현재와 이전 SMB)가 필요하다. 제한한 회로 구조에서는 외부 메모리로부터 정수 화소 데이터를 가져와서 이 값들을 각 보간기에서 재사용하기 위해 내부 SRAM에 저장한다. 더 자세히 설명하면, 정수 화소 데이터는 수평, 수직, 대각선 방향 부화소 보간 연산을 위해 외부 메모리로부터 가져와서 SRAM<sub>W</sub>에 저장된다. SRAM<sub>W</sub>는 7개 3x32 비트 SPSRAM으로 구성되고 1개 16x16 MB를 보간하기 위해 21개 화소들을 저장한다. SRAM<sub>W</sub>에 입력된 데이터는 7개 3x32 비트 SPSRAM의 각각에 저장되고 그 순서는 (그림 7)에 나타내었다. 1개 정수 화소 데이터는 매 사이클마다 'din'을 통해 입력되고 7개 정수 화소 데이터는 매 사이클 당 'dout<sub>0</sub>'~'dout<sub>6</sub>'을 통해 1/2 화소 수평 방향 보간기 (INTP<sub>H</sub>), 1/2 화소 수직 방향 보간기 (INTP<sub>V</sub>), 1/2 화소 대각선 방향 보간기 (INTP<sub>D</sub>)와 1/4 화소 보간기 (INTP<sub>Q</sub>)에 공급된다.

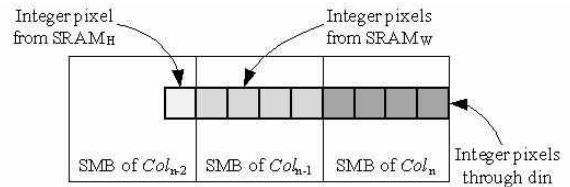
1/2 화소 수평 방향 보간기 INTP<sub>H</sub>는 'din'을 통해 외부 메모리에서 정수 화소 데이터를 입력 받는다. 또한 이전 SMB의 화소 데이터는 입력 데이터로 사용된다. 예를 들어 Col<sub>1</sub>에 대한 SRAM<sub>W</sub>의 SMB는 Col<sub>0</sub>의 수평 방향 보간에 사용된다. (그림 8)에 나타낸 것과 같이 Col<sub>n</sub>에 대한 데이터는 'din'을 통해 입력되고 Col<sub>n-1</sub>와 Col<sub>n-2</sub>에 대한 데이터는 SRAM<sub>W</sub>와 SRAM<sub>H</sub>로부터 제공된다.

(그림 9)는 1/2 화소 수평 방향 보간기 INTP<sub>H</sub>의 블록도이다. 이 회로는 각 사이클마다 9개 화소들을 입력 받아 4개 보간기와 데이터 증채기를 통해 4개의 보간 결과를 생성한다. 수직 방향 보간기 INTP<sub>V</sub>와 대각선 방향 보간기 INTP<sub>D</sub>는 수평 방향 보간기 INTP<sub>H</sub>와 동일한 구조를 갖는다.

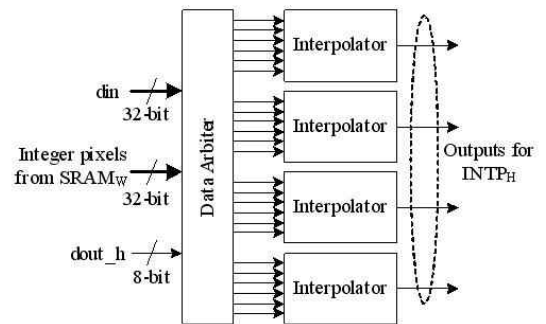
(그림 10)은 1/2 화소 보간기들 (INTP<sub>H</sub>, INTP<sub>V</sub>, INTP<sub>D</sub>)에서 사용되는 FIR 필터의 구조를 나타낸다. 각 보간기마다 4개의 FIR 필터가 필요하므로 총 12개의 필터를 사용한다. 본 논문의 FIR 필터 회로는 곱셈기를 전혀 사용하지 않기 위해 각 필터 계수 (1/32, -5/32, 20/32, 20/32, -5/32, 1/32)를 2의 지수승으로 변환하여 덧셈과 쉬프트 연산만을 사용하였다. 1/4 화소 보간기 (INTP<sub>Q</sub>)는 48개 양선형 필터를 사용한다. 각 양선형 필터는 2개 가산기와 1개 쉬프트 연산기로 구성된다. 양선형 필터 회로에도 역시 곱셈기가 사용되지 않았다.



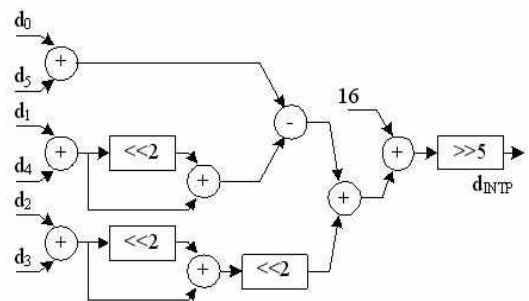
(그림 7) SRAM<sub>W</sub>에서의 화소 데이터 저장 방법



(그림 8) 수평 보간에서의 데이터 입력



(그림 9) 수평 방향 보간기 INTP<sub>H</sub>의 블록도

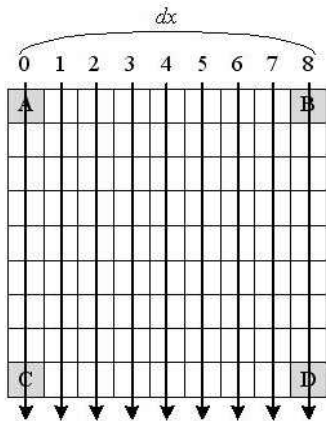


(그림 10) 6탭 FIR 필터의 구조

3.3 1/8 부화소 보간 회로의 구조 및 동작

1/8 화소 보간 회로는 식 (3)을 식 (4)와 같이 변환하여 구현하였다. 동일 열에서  $Z_4$ 의 8개 값들은 (그림 11)과 같이  $dx$  (0~8)의 각 값들이다. 유사하게 같은 행에서  $Z_5$ 의 8개 값들은  $dy$  (0~8)의 각 값들이다.  $Z_3$ 의 모든 값들은 서로 다른 값들로 구성된다. 보간 회로가 매 사이클 마다 64개 1/8 화소들을 보간하는 것을 가정해 보자. 식 (3)을 이용하여 보간한다면, 512 (=8x64)개의 곱셈기와 448 (=7x64)개의 가산기가 필요하다. 이와 달리 식 (4)를 이용하면, 42개의 곱셈기 ( $Z_3$ 에서 28개,  $Z_4$ 에서 7개,  $Z_5$ 에서 7개)와 165개의 가산기 ( $Z_3$ 에서 3개,  $Z_4$ 에서 0개,  $Z_5$ 에서 1개, a에서 3x49+7개)가 필요하다. 식 (3) 대신 식 (4)를 이용하면 약 91.8%의 곱셈기와 약 63.2%의 가산기를 절약할 수 있다. 색차 성분에 대한 1/8 화소 보간기 INTP<sub>C</sub>는 휘도 성분 보간기와 동일한 방법으로 (그림 7)에서와 같이 SRAM<sub>W</sub>와 'din'으로부터 데이터를 입력 받는다. 이 보간기는 1개 4x4 SMB를 보간하는데 16 사이클이 필요하다.

$$\begin{aligned}
 Z_3 &= dx \cdot dy \cdot (A - B - C + D) \\
 Z_4 &= -dx \cdot (A - B) \\
 Z_5 &= -dy \cdot (A - C) \\
 a &= \text{round} (Z_3 \gg 6 + (Z_4 + Z_5) \gg 3 + A) \quad (4)
 \end{aligned}$$



(그림 11) 각각의 dx에 대한 Z 값 분포

4. 구현 결과

제안된 보간 회로는 Verilog HDL을 이용하여 RTL로 설계하였고 Xilinx Virtex4 LX60 FPGA를 사용하여 회로의 동작을 검증하였다. 130nm CMOS 표준 셀 라이브러리와 Synopsys사의 Design Compiler 2007.03을 이용하여 합성된 게이트 수준 회로는 20,674 게이트이고 최대 동작 주파수는 244MHz이다. 본 회로에서 사용된 SPSRAM의 크기는 3,232 비트이다. <표 1>은 4x4 SMB를 보간하는 경우에 다른 연구와 비교한 결과를 나타낸다. 논문 [10]은 처리 속도 면에서는 우수하지만 회로 크기는 크고 휘도 보간만을 지원한

다. 논문 [12]는 처리 속도가 매우 뛰어나지만 회로가 크고 많은 양의 내부 SRAM을 사용한다. 본 논문에서 제안한 회로는 다른 연구 결과와 비교하여 회로 게이트와 SRAM을 포함했을 때 그 크기가 작고 처리 속도도 우수한 것을 알 수 있다. 구현된 회로는 초당 15.3M SMB에 대한 연산이 가능하기 때문에 초당 60장 이상의 1080HD 영상을 처리할 수 있다.

<표 1> 성능 비교

	게이트 수 (K gates)	최대 동작 주파수 (MHz)	#M SMB's/초	지원 가능	SRAM (bits)	공정 (nm)
[7]	23.9	100	10.0	Luma	N.A.	180
[8]	24.3	274	< 2.7	Luma	D: 1.9K	180
[9]	32.5	248	15.5	Luma	0	180
[10]	34.8	270	22.5	Luma	0	130
[11]	46.6	125	4.1	Luma Chroma	0	180
[12]	64.9	250	27.8	Luma Chroma	S: 128K	130
[13]	38.8	250	8.1	Luma Chroma	0	180
제안한 방법	20.7	244	15.3	Luma Chroma	S: 3.2K	130

D: Dual-port SRAM, S: Single-port SRAM

5. 결론

본 논문은 H.264 부화소 움직임 예측에 적용 가능한 휘도와 색차 성분 부화소 보간기의 구조에 대해 제안하였다. 제안된 구조는 내부 SRAM을 효과적으로 재사용하여 회로의 크기를 줄였다. 또한 반복하여 입력되는 데이터를 줄이고 모든 보간기를 병렬로 동작시킴으로써 성능을 개선하였다. 본 논문에서 제안한 회로는 130nm CMOS 표준 셀 라이브러리로 합성하였을 때, 1/2 휘도 성분 화소, 1/4 휘도 성분 화소, 1/8 색차 성분 화소를 1080HD 영상에 대해 초당 60장 이상 생성할 수 있다.

참고 문헌

[1] ITU-T Recommendation and International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Oct., 2004.  
 [2] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. on Circuits and Systems for Video Technology, Vol.13, No.7, pp.560-576, July, 2003.  
 [3] M. Horowitz, A. Joch, F. Kossentini, and A. Hallapuro,

“H.264/AVC baseline profile decoder complexity analysis,” IEEE Trans. on Circuits and Systems for Video Technology, Vol.13, No.7, pp.704-716, July, 2003.

[4] L.Yang, K.Yu, J.Li, and S.Li, “Prediction-based directional fractional pixel motion estimation for H.264 video coding,” Proceedings of ICASSP, pp.901-904, 2005.

[5] L. Yilong and S. Orintara, “Fractional-pel motion refinement based on hierarchical adjustable dual-parabola model,” IEEE International Symposium on Communications and Information Technology, Vol.2, pp.752-755, Oct., 2004.

[6] T.C. Chen, Y.W. Huang, and L.G.Chen, “Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC,” Proceedings of ICASSP, pp.9-12, May, 2004.

[7] T-C. Chen, Y-W. Huang, and L-G. Chen, “Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC,” IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol.5, pp 749-756, May, 2004.

[8] Y. Song, Z. Liu, S. Goto, and T. Ikenaga, “A VLSI architecture for motion compensation interpolation in H.264/AVC,” International Conference on ASIC, Vol.1, pp.279-282, Oct., 2005.

[9] C. Yang, S. Goto, and T. Ikenaga, “High performance VLSI architecture of fractional motion estimation in H.264 for HDTV,” IEEE International Symposium on Circuits and Systems, pp.2605-2608, May, 2006.

[10] L. Lu, J.V. McCanny, and S. Sezer, “Multi-standard sub-pixel interpolation architecture for video motion estimation,” IEEE International SOC Conference, pp.229-232, Sep., 2008.

[11] C-Y. Tsai, T-C. Chen, T-W. Chen, and L-G. Chen, “Bandwidth optimized motion compensation hardware design for H.264/AVC HDTV decoder,” Midwest Symposium on Circuits and Systems, Vol.2, pp.1199-1202, May, 2005.

[12] M. Alle, J. Biswas, and S.K. Nandy, “High performance VLSI implementation for H.264 Inter/Intra prediction,” International Conference on Consumer Electronics, pp.1-2, July, 2008.

[13] C. Lee and Y. Yu, “Design of a motion compensation unit for H.264 decoder using 2-dimensional circular register files,” International SoC Design Conference, Vol.2, pp.109-112, Nov., 2008.



**이 선 영**

e-mail : drleesy@keti.re.kr  
 1998년 한국외국어대학교 전자공학과(학사)  
 2000년 한국외국어대학교 전자공학과  
 (공학석사)  
 2009년 한국외국어대학교 전자공학과  
 (공학박사)

2001년~2006년 (주)이시티 반도체설계연구소 선임연구원  
 2009년~현 재 전자부품연구원 융합신호SoC연구센터 선임연구원  
 관심분야: SoC 설계 등



**조 경 순**

e-mail : kscho@hufs.ac.kr  
 1982년 서울대학교 전자공학과(학사)  
 1984년 서울대학교 전자공학과(공학석사)  
 1988년 미국 Carnegie Mellon University  
 전기 및 컴퓨터 공학과(공학박사)  
 1988년~1994년 삼성전자(주) 반도체 총괄

선임, 수석 연구원  
 1994년~현 재 한국외국어대학교 전자공학과 조교수, 부교수,  
 정교수  
 관심분야: SoC 설계 등