

MPEG-4 CODEC 용 디블로킹 필터 회로 설계

김승호, 조정순
한국의국어대학교 전자정보공학부

Design of a Deblocking Filter Circuit for MPEG-4 CODEC

Seungho Kim, Kyeongsoon Cho
School of Electronics and Information Engineering
Hankuk University of Foreign Studies

E-mail : acad@hufs.ac.kr

Abstract

본 논문에서 기술하고 있는 디블로킹 필터는 ISO/IEC 14496-2 의 디블로킹 필터링 알고리즘[1][2]을 기반으로 한다. 한 개의 레지스터 뱅크를 이용한 효율적인 데이터 스케줄링을 통해 면적과 전력 측면에서 디블로킹 필터를 사용함으로써 생기는 오버헤드를 최소화시켰으며, CIF 급 영상을 27MHz 동작주파수에서 실시간으로 처리할 수 있도록 설계 하였다. 0.25 μ m Standard Cell Library 로 합성한 결과 총 9800 게이트로 구성 되었으며, 외부 메모리의 도움 없이 동작 시키기 위해 4.4KByte 의 버퍼가 사용되었다.

I. 서론

MPEG-4 와 같은 많은 동영상 압축 표준들은 DCT (discrete cosine transform)와 같은 8x8 픽셀 블록 기반의 변환을 사용한다. 블록 기반의 변환 방법은 양자화 과정과 결합되어 공간적 중복성을 제거하여, 영상에 큰 손상을 주지 않고, 압축률을 효과적으로 증가 시킨다. 그러나 이러한 블록 기반의 변환 방법은 8x8 픽셀 블록 단위로 이루어 지기 때문에 블록간의 경계 부분에 블로킹 현상이 발생 일어나게 된다. 그리고 이 현상은 압축률이 높아 질수록 더욱 심화 된다. MPEG-4 표준에서는 블로킹 현상을 제거하기 위해 디블로킹 필터를 사용

하는 것을 권고하고 있다. 그림 1 은 MPEG-4 복원 과정을 나타낸다. 디블로킹 필터는 영상 복원 과정 중 후처리(Post processing) 과정에 포함 되어 화질을 개선하는 역할을 수행한다.

본 논문은 ISO/IEC 14496-2 의 디블로킹 필터링 알고리즘을 이용한 디블로킹 필터 설계에 대한 내용을 담고 있다. 기존의 디블로킹 필터의 구조[4]는 필터링을 위한 데이터와 출력을 위한 데이터를 저장하는 레지스터 뱅크를 각각 따로 두었지만, 본 논문에서 제안한 구조는 한 개의 레지스터 뱅크만을 사용하기 때문에 회로 크기와 연산 시간을 줄였다. 또한 외부 메모리의 도움 없이 2-D 디블로킹 필터링이 가능하도록 구성하였다. 2 절에서는 적용된 디블로킹 필터 알고리즘에 대해서 설명하고, 3 절에서는 설계된 디블로킹 필터의 구성과 1-D 디블로킹 모듈의 구조를 나타내었다. 그리고 4 절과 5 절에서는 실험 결과와 결론을 제시하였다.

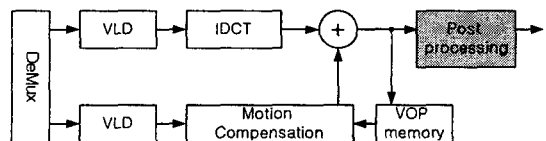


그림 1 Video decoder for MPEG-4

II. 디블로킹 필터 알고리즘

MPEG-4 에서 표준화된 디블로킹 알고리즘[3]은 블록 경계부분에 존재하는 픽셀들의 상태에 따라 두개의 분리된 필터링 모드를 가진다. 각 모드에서 필터링은 수평, 수직 방향에 대한 일 차원 필터링을 통해 이루어지며, 그림 2 에서와 같이 수직 경계에 대한 필터링을 수직 경계 필터링이라하고, 수평 경계에 대한 필터링을 수평 경계 필터링이라 한다. 수평 경계 필터링을 수행한 이후 수직 경계 필터링을 수행하는데, 만약 수평 경계 필터링에 의해서 픽셀 데이터가 변경되었다면, 변경된 데이터로 수직 경계 필터링을 수행하여야 한다.

그림 3 은 디블로킹 필터링 알고리즘을 플로우 차트로 나타낸 것이다[3]. 먼저 인근 픽셀들간의 차이의 합(F(v))을 구한 후에 Th2 값에 따라 default 모드로 동작할 것인지 smooth 모드로 동작할 것인지를 결정한다. Default 모드에서는 블록간의 경계에 위치한 픽셀(V₄, V₅) 값을 필터링된 픽셀(V'₄, V'₅) 값으로 변경한다, Smooth 모드는 낮은 DC offset 값으로 인해 블로킹 현상이 심화된 경우에 적용되는데, 8 개의 픽셀(V₁~V₈) 데이터가 필터링된 픽셀(V'₁~V'₈) 값으로 변경된다. 그러나 필터링 연산 이후 데이터 변경은 항상 이루어 지는 것이 아니라 각 모드별로 QP(quantization parameter) 값에 따른 조건이 만족될 경우에만 변경이 이루어 진다.

III. 디블로킹 필터 구조

설계한 디블로킹 필터는 그림 4 와 같이 수평, 수직 경계 필터링을 위한 1-D 디블로킹 필터 모듈 2 개와 외부 메모리의 도움 없이 IDCT 출력을 필터링하기 위한 두개의 버퍼로 구성되어 있다. 필터링된 결과물은 최종적으로 메모리에 저장되거나 디스플레이 장치에 전달된다.

3.1 전체 구조

MPEG-4 복원 과정중 블로킹 현상을 제거하기 위한 필터링 연산은 IDCT 연산 이후에 수행된다. MPEG-4 복원과정은 매크로 블록 단위로 이루어지기 때문에 IDCT 의 출력은 그림 5 와 같은 순서로 나오게 된다. 필터링 연산은 블록 경계 부분을 중심으로 10 개의 픽셀 데이터(V₀~V₉)를 이용하여 수행 되기 때문에(그림 2

참고) 두 번의 1 차원 필터링을 하기 위해서는 일정한 크기의 버퍼가 필요하다.

Buffer1 은 IDCT 출력을 입력으로 하여 수평방향 필터링을 위해 사용되고, Buffer2 는 수평방향 필터링된 픽셀 데이터를 다시 수직 방향으로 필터링 하기 위해서 사용된다. 필터링을 수행하기 위해서는 인접한 두개의 블록(8x8)이 확보되어야 하므로 두개의 매크로 블록(16x16) 을 저장할 512Byte 의 저장 공간이 필요하다. 그리고, 그림 6 에서와 같이 블록 A, B 사이의 경계 부분에서 디블로킹 필터링을 하기위해 5 라인 크기(5x320=1600Byte)의 데이터를 저장하고 있어야 한다. 4:2:0 포맷의 컬러 이미지의 경우 chrominance 데이터를 포함하기 위하여 5 라인 크기(5x160=800Byte)의 데이터 저장도 필요하다.

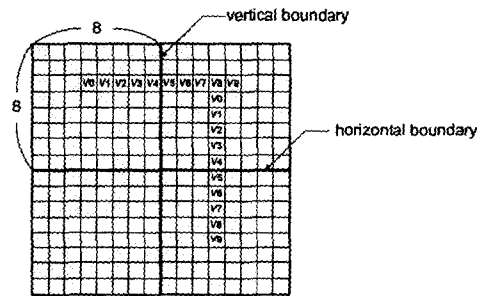


그림 2 Block boundary

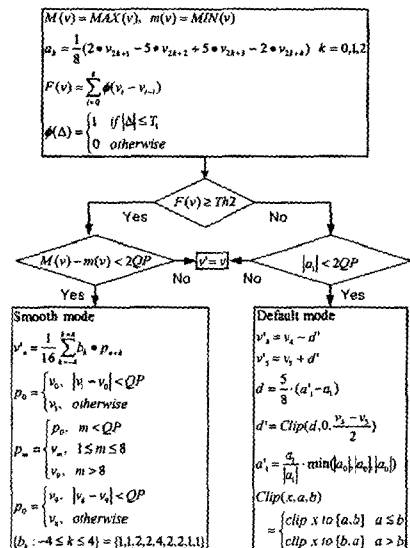


그림 3 Deblocking filtering algorithm

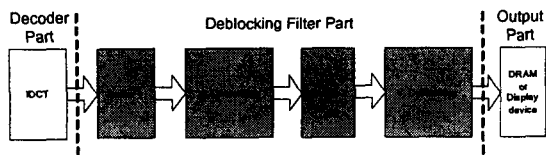


그림 4 2-D 디블로킹 필터 구조

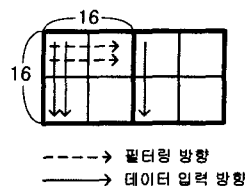


그림 7 수직 방향 필터링

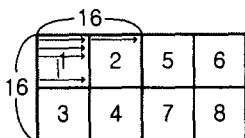


그림 5 IDCT의 출력 순서

수평 방향의 디블로킹 필터링이 끝나면 수직 방향의 필터링을 수행한다. 이때 수평 방향 필터링의 데이터 출력 순서는 그림 7 와 같이 매크로 블록 단위로 세로 방향으로 출력되고, 필터링 연산은 가로 방향으로 수행이 되므로 두 개의 매크로 블록(512Byte)을 저장할 공간이 필요하다. 따라서 CIF 급 320x240 의 해상도를 갖는 영상에 대해서 외부 메모리의 도움 없이 디블로킹 필터링을 수행하기 위해서는 아래와 같이 4.4KByte 의 메모리가 필요하다. 표 1 은 필요한 버퍼의 크기를 표로 정리한 것이다.

표 1 CIF 급 영상 필터링을 위한 메모리 사이즈

| | |
|--------------|--|
| Buffer1 | Luminance: 2,112 Byte (512 + 1,600) Chrominance: 1,312 Byte (512 + 800) |
| Buffer2 | Luminance: 512 Byte Chrominance: 512 Byte |
| Total | 4,448 Byte |

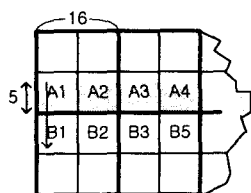


그림 6 수평 방향 필터링을 위해 저장될 픽셀

3.2 1-D 디블로킹 필터 구조

그림 8 은 1-D 디블로킹 필터 모듈의 구조를 나타낸다. Default 모드와 smooth 모드의 필터링은 DM 모듈과 SM 모듈에서 각각 수행된다. 레지스터 뱅크는 10 개의 쉬프트 레지스터로 구성되어 있으며 DM 과 SM 에 필요한 데이터를 전달하는 역할을 한다. 필터링은 10 개의 픽셀 데이터의 상태에 따라 2 가지 모드 중 하나를 선택하여 필터링을 수행하게 된다. 모드 선택은 Mode decision 블록에서 계산된 $F(v)$ 에 의해 이루어진다. 각각의 모드가 선택 되어진 이후에는 QP 값에 따라 필터링된 데이터가 반영된다. 그림 6 은 1-D 디블로킹 필터 모듈의 구성을 나타낸다.

Default 모드에서의 필터링은 DM 모듈에서 이루어 지는데, 필터링을 위해서는 $a(k)$ 값이 필요하다. $a(k)$ 는 4x1 DCT 연산을 통해 이루어지며 본 논문에서는 $a(k)$ 값을 구하는데 필요한 모든 곱셈 연산을 덧셈과 쉬프트 연산으로 구현하였다. Smooth 모드에서의 필터링은 9 개의 필터 계수를 사용한 저역 통과 필터링 연산으로 이루어져 있다(그림 3 참조). 본 논문에서는 이를 그림 9 와 같이 구성 하였다.

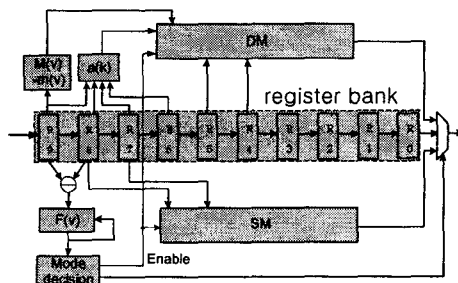


그림 8 1-D deblocking filter block diagram

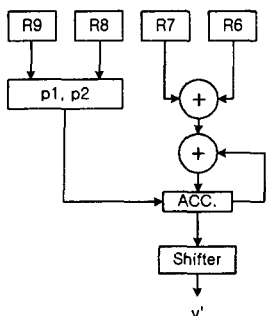


그림 9 SM module 구조

IV. 실험 결과

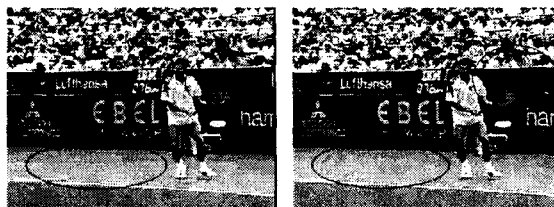
본 논문에서는 Verilog HDL 을 사용하여 회로 설계를 하고 Cadence 사의 NC-Verilog 로 시뮬레이션 하였다. 그리고, MoMuSys MPEG-4 프로그램을 이용하여 결과를 검증하였다. ㈜삼성전자의 0.25 μ m Standard Cell Library 와 Synopsys 사의 Design Compiler 를 사용하여 버퍼에 사용된 메모리를 제외하고 합성한 결과 9800 게이트로 구성된 논리수준 회로를 구현하였으며, 27MHz 의 동작 주파수에서 동작이 가능함을 확인하였다. 표 2 는 1-D 디블로킹 필터 모듈에 대한 게이트 사이즈와 지연시간을 나타내고 있으며, [4]에서 제안한 구조에 비해 약 1000 게이트 정도 적다. 표 3 과 그림 10 은 QP 값 변화에 따른 디블로킹 필터의 효과를 실험한 것이다.

표 2 1-D deblocking filter module 합성결과 비교

| | Gate count | Max. clock rate |
|----------|------------|-----------------|
| Proposal | 3896 | 125MHz |
| [4] | 5213 | 100MHz |

표 3 QP 값에 따른 디블로킹 필터의 효과

| QP | PSNR (before) | PSNR(after) |
|----|---------------|-------------|
| 6 | 35.378 | 35.416 |
| 8 | 33.046 | 33.098 |
| 14 | 28.720 | 28.790 |
| 20 | 26.079 | 26.147 |



(a) 필터링 전 (b) 필터링 후

그림 10 디블로킹 필터링의 효과(QP=6)

V. 결론

본 논문에서는 ISO/IEC 14496-2 의 디블로킹 필터링 알고리즘을 기반으로 하여 CIF 급 320x240 해상도의 이미지를 처리 할 수 있는 디블로킹 필터를 27MHz 동작 주파수에서 실시간으로 처리할 수 있도록 설계한 결과를 기술하였다. IDCT 의 출력을 받아 외부 메모리의 도움 없이 필터링이 가능하며, 한 개의 레지스터 뱅크를 이용한 효율적인 스케줄링을 통해 크기와 연산 시간을 최소화 시켰다. 본 논문에서 기술한 디블로킹 필터는 MPEG-4 CODEC SOC 에서 후 처리 모듈로 사용될 예정이다.

Acknowledgement

본 연구는 ㈜ECT 사의 지원에 의해서 이루어졌음.

참고문헌

- [1] ISO/IEC 14496-2:1999, "Information technology-generic coding of audio-visual objects".
- [2] W. Li, J. R. Ohm, "MPEG-4 video verification model version 18.0", ISO/IEC JTC1/ SC29/WG11 N3908, January 2001.
- [3] S. D. Kim, J. Yi, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, Feb. 1999.
- [4] H. C. Fang, T. C. Wang, L. G. Chen, "Real time deblocking filter for MPEG-4 system," *IEEE Circuits and Systems, APCCAS '02. 2002 Asia-Pacific Conference*, vol. 1, pp. 541-544, Oct. 2002.