

고화질 텔레비전용 이차원 필터 설계

(Two dimensional Filter Design for HDTV)

朴柱成*, 尹炳愚**, 諸映昊***, 梁眞榮**, 朴鍾哲****, 沈英錫*****

(Ju Sung Park, Byung Woo Yoon, Yeong Ho Je, Jin Young Yang,
Jong Chul Park and Young Serk Shim)

要約

HD-MAC 수상기 인터플레이션용 이차원 필터를 1.0 μm CMOS 표준셀을 이용하여 설계하고 검증하였다. 인터플레이션 기본 알고리즘은 FMH(FIR-Median Hybrid) 필터를 이용하였다. FMH 필터에서 가장 복잡한 부분인 메디안 필터를 중앙값을 찾는 알고리즘인 Hadian-Sobel 알고리즘을 개조함으로써 비교적 간단한 로직으로 구현할 수 있었다. 설계된 필터는 배치배선후의 최악의 지연조건하에서 휘도 및 색차신호의 전송되지 않은 픽셀을 27MHz 클럭에서 실시간으로 인터플레이션 하는 것을 확인했다.

Abstract

Two dimensional interpolation filter for HD-MAC is designed with 1.0 μm CMOS standard cells and verified by logic simulation. The interpolator uses FMH(FIR-Median Hybrid) filter. The median filter, which is the most complicated part of FMH filter, is simply implemented by modifying Hadian-Sobel algorithm. The filter generates accurately the missed pixel data of luminance and chrominance with the worst case simulation condition at 27MHz clock rates.

1. 서론

*正會員, 釜山大學校 電子工學科
(Dept. Elec. Eng., Pusan Nat'l Univ.)

**正會員, 韓國電子通信研究所
(ETRI)

***正會員, 金星일렉트론 仲央研究所
(Gold Star Electron)

****正會員, 金星社 LCD 研究所
(Gold Star LCD Research Center)

*****正會員, 生産技術研究院 電子情報시스템 센터
(KAITECH)

接受日字 : 1994年 1月 21日

1960년대 말 부터 일본 NHK에서 HDTV(High Definition TV)에 대한 연구를 시작한 이후 여러가지 방식의 HDTV가 출현하였다. HDTV 방식은 일본의 MUSE(Multiple Sub-nyquist Sampling Encoding) 방식과 미국의 완전 디지털 방식, 유럽의 HD-MAC(High Definition Multiplexed Analog Components) 방식으로 나눌 수 있다.

전송 채널의 용량한계로 인하여 HDTV 방송국에서는 샘플링된 영상 데이터의 일부만 전송하므로 수 상기는 전송된 데이터와 각종 콘트롤 신호를 이용하

여 빠진 데이터를 복원한다. 일반적으로 영상신호 인터폴레이션을 위해서는 이차원 FIR (Finite Impulse Response) 필터나 메디안 필터가 널리 이용된다. FIR 필터는 계산량이 많기 때문에 많은 로직소자가 필요하며 가는 선이나 날카로운 경계를 잘 복원하지 못한다. [1] [2] 메디안 필터는 회로구성이 간단하며 잡음에 강하고 또한 가는 선을 잘 복원할 수 있는 장점이 있기 때문에 영상신호 인터플레이터로 많이 사용되고 있다. [3] 두 필터의 장점을 고루 갖춘 FMH (FIR-Median Hybrid) 필터가 영상신호의 인터폴레이션에 사용될 수 있음이 알려졌다. [4] [5] 이 필터방식을 이용하여 HDTV용 인터플레이션 필터를 만들었을 경우 필터 특성도 양호하고 비교적 간단한 하드웨어로 구현할 수 있다고 발표된 바 있으나 [6], 구체적인 구현방법에 대해서는 알려진 바가 없다. 메디안 필터는 알고리즘의 복잡성 때문에 하드웨어적인 구현보다는 소프트웨어 방식으로 처리되었다. [7] 소프트웨어적인 방법으로 처리할 경우는 처리시간이 오래 걸리기 때문에 HDTV 영상신호와 같은 수십 MHz 급의 신호를 실시간으로 처리하는데는 많은 문제점을 안고있다. HDTV 신호를 실시간으로 처리하기 위해서 일차원 FIR 필터를 만들어 이차원 어레이를 구성하여 사용되었다. [8] [9] 본 연구에서는 여러 숫자중에서 크기 순서를 가장 적은 비교로 결정할 수 있는 Hadian-Sobel 알고리즘을 개선하여 HD-MAC 수상에 사용될 수 있는 FMH방식의 이차원 인터플레이터를 설계하고 검증하였다. 본 연구에서 고안된 메디안 필터 구현 방법은 다른 방식의 HDTV 인터플레이터에도 응용될 수 있을 것이다.

II. HD-MAC 신호 및 신호처리 방법

HDTV 영상신호는 휘도신호(luminance)와 색차신호(chrominance)로 구성되며 색차신호의 대역폭은 휘도신호의 반이다. 휘도신호의 인터폴레이션 개념을 색차신호에도 그대로 적용할 수 있기 때문에 본 논문에서는 편의상 휘도신호의 인터폴레이션 방법만 다룬다. 인간의 눈은 수평주파수나 수직주파수에 대해서 민감한 반면 대각선 주파수 성분에 대해서는 그렇지 않다. 이러한 점을 이용하여 그림 1과 같은 형태의 샘플링을 하면 수직·수평주파수 성분에는 손상을 주지 않으면서 대각선 주파수 성분을 줄일 수 있어서 대역폭을 줄일 수 있다. [10] 그림 1에서 x 표시된 것은 전송되지 않은 픽셀이고, 숫자(1,2,3,4)는 해당픽셀이 전송되는 필드번호이며 이와같은 샘플링 방식을 quincunx 샘플링이라고 한다.

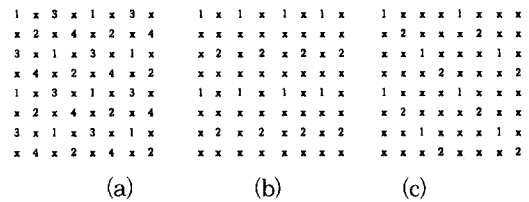


그림 1. HD-MAC 방식의 휘도신호 샘플링 방법
(a) 80msec 모드 (b) 40msec 모드 (c) 20msec 모드

Fig. 1. Subsampling patterns of luminance signals for HD-MAC. (a) 80msec mode, (b) 40msec mode, (c) 20msec mode

1. HD-MAC 영상신호

HD-MAC 방식에서는 화면을 16×16 픽셀 단위의 블록으로 나누는데, 각 블록의 움직임 정도에 따라서 그림 1과 같이 정지모드(80msec 모드), 저속모드(40msec 모드), 이동모드(20msec 모드)로 샘플링한다. HD-MAC 방식은 필드 주파수가 50Hz이고 정지모드는 한번 촬영된 화면을 4개의 필드에 걸쳐 전송하므로 전송되는 데이터들이 80msec마다 새로운 데이터로 리프레쉬되므로 이 모드를 80msec 모드라고도 한다. 저속모드와 이동모드의 경우에는 각각 40msec와 20msec 마다 전송되는 각 픽셀이 새로운 값으로 갱신된다. 저속모드인 경우는 홀수 필드의 데이터를 두 필드에 걸쳐 나누어 전송하고, 짝수 필드에 대해서는 홀수 필드에 대하여 얼마만큼 움직였는가를 나타내는 움직임 벡터만 전송된다. 수상기에서 용이하게 인터폴레이션하기 위해서 sub-sampled 데이터와 함께 각 블록의 모드, 필드 넘버, 라인 넘버, 움직임 벡터 등 각종 제어신호(DATV: Digitally Assisted Television)가 전송된다.

2. 인터폴레이션 방식

영상신호 인터폴레이션을 위해서는 이차원 필터를 구성해야 한다. 필터 마스크의 크기가 커짐에 따라서 화질은 좋아지지만 VLSI화 측면에서는 상당한 부담이 되어 HDTV 수상기에는 적합하지 않다. 본 연구에서는 하드웨어 부담이 적으면서 비교적 좋은 화질을 가지는 raster 형의 이차원 필터를 설계하였다. [6] 이 필터는 세개의 라인 정보로부터 한 라인을 인터폴레이션하기 때문에 필터 마스크가 큰 이차원 필터보다 처리해야 할 라인수가 작기 때문에 비교적 간단하게 구현할 수 있으며, 캐스캐이드 인터폴레이션 방식에 사용된다. [11] 그림 1과 같이 서브샘플링되어 전송된

각 모드의 휘도신호를 80msec 모드에서는 4개의 필드 데이터를 모아서 전송되지 않은 픽셀을 인터폴레이션한다. 또한 40msec 모드에서는 2개의 필드 데이터를 모아서 홀수 필드의 데이터를 인터폴레이션 한다. 40msec 모드의 짝수 필드는 DATV신호를 통하여 전송된 움직임벡터를 이용하여 홀수필드로 부터 만들어 낸다. 20msec 모드에 해당하는 블록은 화면의 움직임이 빠르기 때문에 매 필드의 데이터를 이용하여 인터폴레이션을 한다.

80msec 모드와 40msec 모드에서는 이차원 인터폴레이터를 이용하면 모든점이 데이터를 갖게되지만 20msec 모드에서는 이차원 필터를 통과시키면 인접한 두 픽셀에서 한 픽셀만 데이터를 가지게 된다. 따라서 이차원 인터폴레이션을 한 후에 일차원 인터폴레이션을 이용하여 나머지 데이터를 인터폴레이션 하여야 모든 픽셀값을 얻을 수 있다. 본 연구에서는 raster 형태로 인터폴레이션하기 때문에 인터폴레이션할 라인과 상·하 두 라인이 필요하다. 그림 2는 휘도 신호에 대한 인터폴레이션 예이다. 여기서 각 픽셀의 표시에서 첫번째 숫자는 블록내에서 라인을, 두번째 숫자는 그 라인에서 픽셀의 위치를 나타낸다. 예를들면 "32"는 어떤 블록의 세번째 라인의 2번째 픽셀을 의미한다. 그리고 실선화살표는 인터폴레이션을 할 라인이고 점선은 인터폴레이션에 필요한 데이터를 제공하는 라인이다. 인터폴레이션 되어야 할 픽셀의 위치는 ⊗로 표시되어 있다.

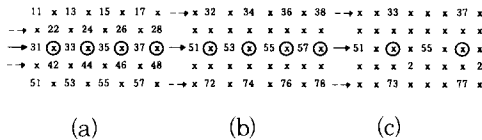


그림 2. 휘도신호 각 모드의 대표적인 인터폴레이션 방법 (a) 80msec 모드 홀수 라인, (b) 40msec 모드 4n+1 라인, (c) 20msec 모드 4n+1 라인

Fig. 2. Representative interpolation method of each luminance mode. (a) 80m-sec mode odd line, (b) 40msec mode 4n+1 line, (c) 20msec mode 4n+1 line

Ⅲ. 인터폴레이션 방법 및 메디안 알고리즘

1. 인터폴레이션 방법

메디안 필터와 같은 비선형 필터를 정확한 주파수

특성분석을 바탕으로 설계하는 것이 매우 어렵다. 따라서 비선형 필터인 경우 특수한 패턴, 즉 날카로운 경계나 가는 선의 복원능력 등으로 그 성능을 평가한다. 몇 종의 간단한 메디안 필터의 특성을 살펴보면 다음과 같다. 그림 3의 (a)와 같은 수직3점 메디안필터의 메디안 값은

$$Y_{MED3} = \text{MEDIAN}[X(0,-1), X(-1,0), X(0,1)] \quad (1)$$

와 같이 정의되는데, 여기서 X(-1,0)은 인터폴레이션할 픽셀과 같은 라인에 있으면서 한 픽셀 좌측에 있는 값이다. 이와같은 표시 방법은 다른 곳에서도 적용한다. 이 알고리즘을 이용하면 수직선은 잘 재생하나 가는 수평선은 잘 재생하지 못한다. 또한 이것을 90° 회전시킨 수평3점 메디안필터에서는 수평선은 잘 재생하지만 가는 수직선은 잘 재생하지 못한다.

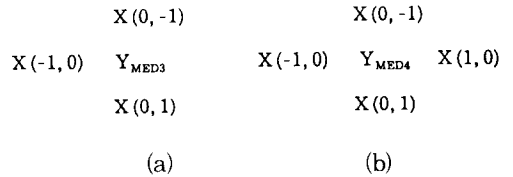


그림 3. 간단한 메디안 필터 (a) 수직3점 메디안필터 (b) 4점 메디안 필터

Fig. 3. Simple median filter. (a) Vertical 3 points median filter, (b) 4 points median filter

그림 3의 (b)와 같은 4점 메디안필터의 메디안 값은

$$Y_{MED4} = \text{MEDIAN} [X(0, -1), X(-1, 0), X(1, 0), X(0, 1)] \quad (2)$$

와 같이 정의되는데 여기서 메디안 값은 두 중앙값의 평균값으로 취할 수 밖에 없다. 이 필터는 대체로 경계선들과 경사들을 잘 복원한다. 그러나 그림 4의 (a)와 (b)에서 ⊗로 표시된 픽셀을 인터폴레이션할 때에는 검은색과 흰색의 중간값으로 인터폴레이션하므로 한 픽셀폭의 가는 선을 잘 복원하지 못한다.

4점 메디안필터의 기능을 보완할 수 있는 방법으로 FIR 필터를 이용하여 제5의 숫자를 만들어 5점 메디안 필터를 만들면 날카로운 경계나 수직·수평의 가는 선을 잘 복원할 수 있다. [6] 이러한 인터폴레이터 출력은 식 (3)과 같이 나타낼 수 있다.

$$Y_{FMH} = \text{MEDIAN}[X(0,-1), X(-1,0), X(1,0), X(0,1), Y_{FR}] \quad (3)$$

여기서 Y_{FIR} 은 인터플레이션할 픽셀 주위의 8점 데이터로부터 만들어 지는데 식 (4)와 같이 표시된다.

$$Y_{FIR} = [X(0,-1) + X(-1,0) + X(0,1) + X(1,0)] / 2 \quad (4)$$

$$- [X(-2,-1) + X(-2,1) + X(2,1) + X(2,-1)] / 4$$

위와같은 마스크와 계수를 이용하여 그림 4의 검은선을 인터플레이션하면 세개의 검은점과 두개의 흰점 데이터 중에서 중간값을 찾는 경우가 되어 검은점으로 인터플레이션 되므로 한 픽셀폭의 선을 잘 복원할 수 있음을 알 수 있다. 실제 이미지 두 프레임을 quincunx smampling한 영상에 대하여 이 필터와 5x7 FIR 필터의 성능을 컴퓨터 시뮬레이션을 통하여 서로 비교하여 보았다. 각 필터의 PSNR을 비교해본 결과 5x7 FIR 필터는 35.2 ~ 38.1dB 이었으며 본 연구에서 사용한 인터플레이션 방법은 35.3 ~ 36.2dB이었다. 소프트웨어적으로 처리한 두 방식의 결과를 실제 화면에 display한 결과 화질의 차이를 느낄 수 없었다. 두 결과가 큰 차이를 보이지 않기 때문에 하드웨어 구현이 용이한 FMH 방식을 채택하였다.

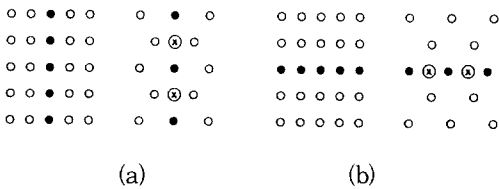


그림 4. One pixel-width의 수직·수평의 원래영상과 quincunx sampling 패턴

Fig. 4. One pixel-width horizontal and vertical lines and their quincunx sampling patterns.

2. 메디안 필터 알고리즘

n개의 샘플을 크기 순서대로 나열하는 방법으로는 Quick sort, Heap sort, Bubble sort 등의 방법을 생각할 수 있다.^[12] 이 알고리즘들은 주로 많은 샘플을 소프트웨어적으로 그 크기를 결정하는 것이기 때문에 하드웨어를 구현하는데 적합하지 못한 면이 있다. 본 연구에서 필요로하는 것은 5개의 샘플 데이터에서 중앙값을 찾는 메디안 필터이다. 이와같이 비교적 샘플개수가 작을 때 샘플간의 비교 횟수를 최소화할 수 있는 알고리즘이 Hadian과 Sobel에 의하여 고안되었다.^[13] 그들에 의하면 n개의 샘플에서 k번째 크기까지의 순서를 결정하는데는 최대 n-k+(k-1)

$\lceil \log_2(n+2-k) \rceil$ 만큼의 비교만 필요하다고 발표되었다. 여기서 $\lceil \log_2(n+2-k) \rceil$ 는 $\log_2(n+2-k)$ 보다 작은 최대 정수값이다. 따라서 5개의 샘플중에서 중앙값을 알기 위해서는 최대 6번의 비교를 하면된다. 본 연구에서는 샘플간의 비교횟수가 최소라는 장점 때문에 이 알고리즘을 시도 하였다.

Hadian과 Sobel의 알고리즘을 A, B, C, D, E 다섯개의 샘플을 예로하여 설명하면 다음과 같다. 첫 번째로 그림 5의 (a)와 같이 5개의 숫자 중에서 임의의 두쌍(예: A와B, C와D)을 선택하여 각 쌍에서 큰 값 두개(B,D)를 찾은 다음, 두 값 중에서 큰 값(D)을 버린다. 그 다음 그림 5의 (b)와 같이 첫번째 과정에서 가장 큰 값(D)이 있었던 자리에 한번도 비교되지 않았던 수(E)를 넣고 첫번째의 비교결과인 A와 B의 관계는 그대로 이용하면 5번째의 비교로써 상위 두값을 모두 제거할 수 있다. 두 과정을 통하여 세개의 숫자만 남아 있으므로 그림 5의 (c)와 같이 3개의 숫자 중에서 가장 큰 수를 찾으면 원하는 중앙값(C)이 된다.

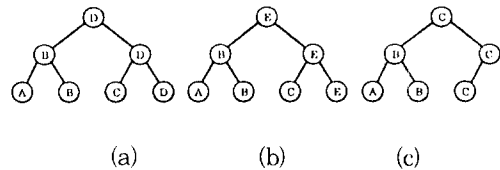


그림 5. Hadian-Sobel 알고리즘을 이용하여 메디안을 찾는 방법 (a) 세번째 비교를 통하여 상위 두 수중의 하나 제거 (b) 다섯번째 비교를 통하여 상위 두 수중의 하나 제거 (c) 여섯번째 비교로써 메디안 결정

Fig. 5. Hadian-Sobel algorithm for finding the median of the five samples. (a) Eliminating one of two biggest numbers from the 3rd comparison (b) Eliminating one of two biggest numbers from the 5th comparison (c) Determination of the median from the 6th comparison.

그림 5를 통하여 메디안 값을 찾기 위해서 6번의 비교가 필요한 것을 알 수 있다. 여섯번의 모든 비교 결과를 고려해서 tree를 그려보면 총 64 경우의 수가 나타난다.^[14] 만약 이 알고리즘을 이용하지 않으면 총 1024 경우의 수를 고려하여 메디안값을 찾아야 하므로 64가지 경우로 줄이는 작업이 용이하지 않다.

Hadian-Sobel 알고리즘을 그대로 구현하면 그림 6과 같은 형태의 로직으로 되는데, 메디안이 결정되기 위해서는 4개의 비교기와 멀티플렉서를 통과하게 되므로 여러클럭이 걸린다. 매 클럭마다 데이터를 실시간으로 처리하기 위해서는 중간 계산 결과를 플립플롭에 저장해야 한다. 로직소자를 연결하는 배선에 의한 지연과 안정성등을 고려하여 비교기와 멀티플렉스를 통과하는데 걸리는 시간이 1 클럭 기간(37nsec)이 된다고 가정하여 중간결과를 저장하는 플립플롭들을 달았을 때 그림 6의 회로를 구현하는데 2,013개의 게이트가 소요되었다. 이 구조에서는 여러 비트로 구성된 데이터들이 여러개의 멀티플렉스와 비교기를 거치므로 중간값을 저장하는데 많은 플립플롭이 들고 실제 배선시에 인터컨넥션에 의한 지연이 많아진다.

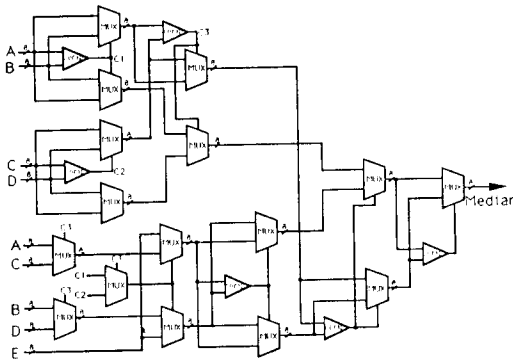


그림 6. Hadian-Sobel 알고리즘을 이용한 메디안 필터 구현

Fig. 6. Median filter implementation based on Hadian-Sobel algorithm.

그림 6의 로직은 Hadian-Sobel 알고리즘 자체에 충실하여 계산시간이 오래걸리게 되어 오히려 전체 게이트 숫자가 늘어났지만, 비교기 갯수를 늘임으로써 메디안을 빨리 찾을 수 있다면 중간처리 결과를 저장할 플립플롭이 필요없게되어 로직갯수를 줄일 수 있을 것이다. 샘플들 간의 크기를 동시에 비교한 결과를 이용해서 어느것이 메디안 이라는 것을 한 클럭안에 결정하면 중간 계산 데이터를 저장하지 않아도 된다. 이 방식을 살펴보면 다음과 같다. Hadian-Sobel 알고리즘에서 A, B, C, D, E 5개의 숫자중에서 A가 메디안이 되는 경우를 정리해 보면 표 1과 같다.

표 1을 분석해보면 A와 관련된 비교기의 출력들 중에서 A를 기준으로 A보다 크다는 출력이 2개인 경우와 A보다 작다는 출력이 2개인 경우만 A가 메디안이 됨을 알 수 있다.

표 1. Hadian-Sobel 알고리즘에 근거한 5개의 숫자(A, B, C, D, E) 중에서 A가 메디안이 되는 경우

Table 1. 'A' median case of five numbers (A,B,C,D,E) based on Hadian-Sobel algorithm.

비교기 출력									
A≥B	A≥C	A≥D	A≥E	B≥C	B≥D	B≥E	C≥D	C≥E	D≥E
0	⓪	⓪	1	-	0	1	0	0	-
0	1	⓪	⓪	1	0	-	0	1	-
0	⓪	1	0	-	1	-	0	-	0
0	1	0	1	-	1	-	0	-	-
0	⓪	⓪	1	0	-	1	1	-	0
0	⓪	1	⓪	0	1	-	1	-	1
0	1	1	0	1	-	-	1	0	-
0	0	⓪	1	-	-	1	-	-	-
1	1	0	0	-	-	-	0	0	-
1	0	1	0	-	-	-	1	0	-
1	0	1	0	-	-	-	-	1	0
1	0	0	1	-	-	-	-	1	1

주) 1, 0은 각각 true, false를 -는 해당 비교기의 출력이 필요없음, 〇안의 1, 0은 다른 비교기의 결과로부터 유추한 것을 각각 나타낸다.

그렇다면 A와 관련되지 않는 비교기의 결과는 사용하지 않고 A와 관련된 비교기의 출력중에서 "1"이 두개이고 "0"이 두개인 경우만 모으면 A가 메디안이 되는 경우이다. 표 1에서 A와 관련된 비교기 출력모음에서 동일한 경우를 제외하면 총 6가지 경우로 줄어든다. 나머지 B, C, D, E가 메디안이 되는 경우에 대해서도 같은 방법으로 분석해 본 결과 각각 6가지 경우로 되었다. A≥B, A≥C, A≥D, A≥E 비교기의 출력을 각각 a,b,c,d 라 할때 A가 메디안이 되는 경우는 다음식과 같이 표시된다.

$$(a \oplus b) \cdot (c \oplus d) + abc'd' + a'b'cd = As \tag{5}$$

여기서 As는 다섯개의 샘플 중에서 A를 메디안으로 선택하는데 사용될 신호이다. 다른 데이터가 메디안이 되는 경우에도 동일한 로직 형태가 된다. 이러한 여섯개의 로직표현은 입력에 같은 숫자가 있는 경우에 대해서도 중간값을 찾아낼 수 있다. 어떤 입력이 메디안인지 결정되는데 걸리는 시간은 하나의 비교기 지연시간과 식 (6)이 처리되는 시간이므로 1.0um CMOS 표준셀을 이용하면 한 클럭기간인 37nsec 내에 충분히 처리할 수 있다. 이렇게 되면 중간처리 결과를 저장할 필요가 없어 1003개의 게이트로 메디안 필터를

구성할 수 있다. 따라서 이 방법은 Hadian-Sobel 알고리즘에 충실 하면서 요구되는 스피드를 만족시키는 경우에 비하여 게이트 갯수를 반으로 줄일 수 있다. 계산시간이 문제가 되지 않은 경우에는 그림 6을 그대로 사용할 수 있는데 이 경우는 857개의 게이트만 필요하다. 저속 데이터를 처리하는 경우는 Hadian-Sobel 알고리즘을 그대로 사용하는 것이 게이트 갯수를 줄이지만, 고속 데이터를 처리하는 데는 본 연구에서 제안하는 방법이 효과적임을 알 수 있다.

IV. 설계 및 검증

본 연구에서 설계한 IC는 HD-MAC 수신기의 색차신호와 휘도신호를 인터플레이션 할 수 있다. 그리고 각 신호의 20msec, 40msec, 80msec 모드를 편으로 지정할 수 있는데 지정되지 않은 모드의 데이터가 들어오면 인터플레이션 하지 않고 일정한 지연을 거친 후 그대로 출력된다. HD-MAC 그리드 상에서는 픽셀간의 간격은 주파수 차원에서 54MHz이다. 이런 주파수 정도로 데이터를 실시간으로 처리하는 것은 어려우므로 odd 픽셀과 even 픽셀을 동시에 입력시켜서 신호를 처리하면 IC 내부에서는 27MHz 클럭으로 동작시키고 IC 외부의 최종단에서 54MHz 클럭으로 처리하면 된다. 각 모드를 분석하여 인터플레이션할 라인은 odd, even 픽셀 데이터로 공급하고 상·하 라인은 각각 8비트의 데이터만 공급해도 된다는 것을 알았다. 입력 데이터들은 8비트로 구성되며 병렬로 입력된다. 다른 입력으로는 휘도 또는 색차신호(L/C), 필드 구분(FIELD), 블럭 모드(EMA, EMB), 라인 구분(LINE), 라인 데이터 시작신호(LST)가 있으며 출력으로는 인터플레이션된 8비트의 odd 픽셀값(OOUT)과 even 픽셀값(EOUT), 정상적인 데이터 값이 출력되는 시기를 알려주는 신호(OVD)가 있다. 편으로 지정되지 않은 모드가 들어왔을 때는 인터플레이션을 하지 않고 인터플레이션되는 라인에 해당되는 odd, even 픽셀값을 그대로 출력시키는데 IC내에서 지연되는 시간은 인터플레이션되는 경우와 동일하다. 설계한 IC는 위에서 소개한 각종신호와 몇개의 공급전원과 접지 핀을 포함하여 총 68핀 패키지에 수용될 수 있다.

1. 설계

본 IC에 데이터가 처음 입력되어 정상적인 값이 출력되는 데 걸리는 시간은 다소 길어도 HDTV의 동작에는 아무런 영향이 없다. 첫번째 값이 출력된 이후에는 27MHz로 계속 정상적인 값이 출력되어야 한

다. 이러한 점을 감안하여 수행해야 할 기능을 여러 단계로 구분하여 각각의 단계에 대하여 한 클럭동안 수행하게 하는 파이프라인 구조를 택했다. 본 IC는 VTI사의 1.0 μm CMOS 표준셀을 이용하여 동사의 COMPASS라는 VLSI 설계툴을 이용하여 설계하였다. 설계된 IC의 내부블럭도는 그림 7과 같다.

데이터 입력부는 각각 5단으로 구성된 4개의 8비트 쉬프트레지스터(IR1 [7:0] ~ IR4 [7:0])와 멀티플렉스(MUX)로 구성되어 있다. 인터플레이션할 라인은 bypass되는 데이터 때문에 odd와 even 픽셀용 쉬프트레지스터가 필요하다. FIR 필터단의 입력을 위해 MUX를 이용하여 각 모드에 따라서 쉬프트레지스터의 정해진 위치의 데이터를 멀티플렉싱 시켜 입력시킨다. 80msec와 40msec 모드에 경우는 3단의 쉬프트레지스터만 있으면 되지만 20msec 모드에서는 5단이 필요하므로 쉬프트레지스터를 5단으로 설계하였다.

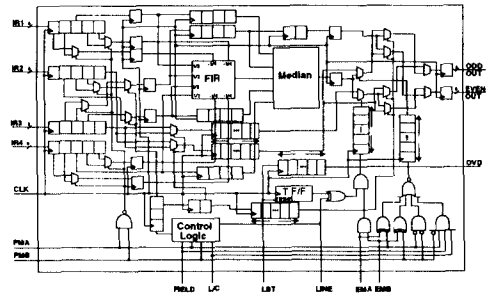


그림 7. 설계된 IC의 내부블럭도

Fig. 7. Internal block diagram of designed IC.

FIR 필터부는 식 (4)와 같은 알고리즘을 수행하는 필터로서 총 8개의 입력데이터를 가진다. FIR 필터의 각종계수들이 2의 멱수 형태이므로 적절하게 쉬프트 라이트하여 2나 4로 나눈 숫자를 만들었으며, 덧셈기 8개와 필요한 로직으로 FIR 필터를 구성했다. 전체 FIR은 4클럭 동안에 수행되도록 설계하였으며, FIR 필터에 사용한 덧셈기는 고속으로 동작을 하여야 하므로 carry-look-ahead 방식으로 설계하였다.

메디안 필터부는 앞장에서 언급한 바와 같이 게이트 수를 줄이기 위해서 10개의 비교기로써 5개 입력값들 간의 크기를 비교하였다. 비교기들의 출력으로부터 특정입력이 중앙값이 되는 경우를 설계한 로직은 그림 8과 같다.

그림 8에서 나타낸 출력은 특정 샘플을 메디안으로 선택하는데 사용된다. 이와같이 만들어진 5개의 선택

신호를 이용하여 5개의 입력값중의 하나를 중앙값으로 선택한다.

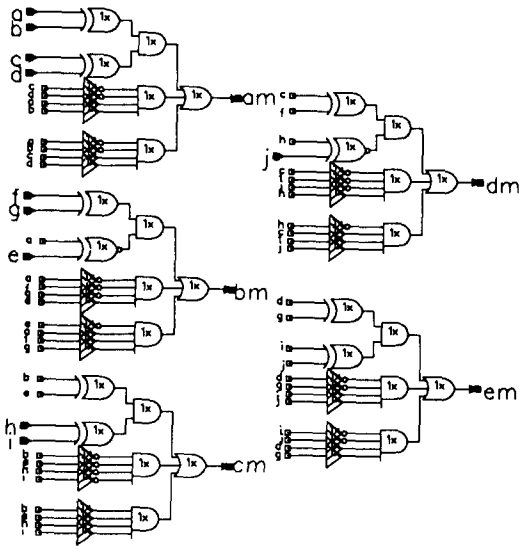


그림 8. 비교기 출력을 이용하여 메디안 값을 찾는 논리회로

Fig. 8. Median determination logic from the comparator outputs.

제어 신호부는 편에 의하여 지정되는 모드신호 (L/C, PMA, PMB)와 외부에서 들어오는 신호 (EMA, EMB, LINE, FIELD, LST) 등으로써 만들어진다. 만들어진 제어신호는 FIR 필터의 입력을 제어하는 MUX에 입력되고, 인터플레이터의 모드와 다른 모드의 데이터가 들어오면 bypass 시키게 된다. 또한 출력 MUX부에 입력되어 적절한 데이터를 출력 시키게 된다. 각종 신호처리가 진행되는 동안 제어신호 및 데이터들은 플립플롭들을 통하여 적절하게 지연시켜 필요한 시기에 사용될 수 있도록 했다. 정상적인 데이터의 입력을 알리는 신호(LST)가 입력된 후 내부처리지시간인 13클럭 후에 실제 데이터가 출력됨을 알리는 신호(OVD)를 만들어서 출력 데이터를 받는 IC의 enable 신호로 사용하게 했다.

2. 검증

VTT사의 1.0 μm CMOS 표준셀을 이용하여 설계된 IC를 1차적으로 로직 시뮬레이션을 통하여 기능시험을 하였다. 각 모드별로 250쌍의 임의의 데이터를 순차적으로 주면서 입력데이터의 모드, 필드, 라인 신호등을 바꾸어 가면서 원하는 값이 출력되는가를 조사하였다. 검증의 편의를 위해서 설계된 IC의 시뮬레이터 프로그

램을 만들어서 시뮬레이터의 결과와 COMPASS의 로직 시뮬레이션 결과를 비교하였다. 이와 같은 방법으로 휘도신호와 색차신호의 각 모드 기능을 조사한 결과 정상적인 동작을 하고 있음을 확인 했다.

설계된 IC가 기능적으로 잘 동작되지만 ASIC화 과정에서 발생하는 배치 배선상의 지연시간으로 인하여 오동작하는 경우가 있다. 이 점을 조사하기 위하여 입·출력 패드를 연결해서 배치 배선을 한후 지연시간을 추출하여 기능시험을 하였다. 이때 지연시간은 최악의 조건 (Vdd=4.65V, Vss=0.1V, 동작온도=70°) 하에서 추출된 것이기 때문에 이 시험을 통과하면 ASIC화 했을 때 문제점이 없다. 이러한 지연시간과 기능시험에 사용했던 다양한 테스터 패턴을 이용하여 전 모드에 대하여 시험해본 결과 한 클럭기간(37nsec) 동안에 모든 기능을 수행하는 것을 확인했다.

그림 9은 여러모드 중의 하나인 휘도신호 80msec 모드에 대한 시뮬레이션 결과의 타이밍 다이어그램이다. 시뮬레이션에서 첫째 블럭은 80msec 휘도신호이므로 13클럭 후에 인터플레이션이 된 데이터가 출력되어야 한다. 입력 데이터 및 모든 제어신호는 클럭의 rising edge에 동기되어 들어온다.

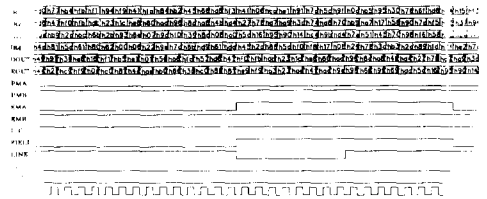


그림 9. 휘도신호 80msec 모드의 시뮬레이션 타이밍 다이어그램

Fig. 9. Simulated result of luminance 80msec.

그림 9에서 LST가 0인 상태에서 IR1~IR4에 각각 16진 데이터 d0,d6,0e,4a가 입력되면 FIR필터와 메디안 필터를 거쳐 13클럭후에 odd출력에는 IR2의 데이터인 16진수 d6이 bypass 되고, even 출력에는 인터플레이션된 값인 16진수 88이 출력됨을 알 수 있다. 인터플레이션된 값은 13번째 클럭이 rising되고 19nsec후에 출력된다. 클럭주기를 36nsec로 가정할 때 17nsec의 여유가 있기 때문에 ASIC화 하여도 정상적으로 동작할 수 있음을 알 수 있다. 또한 DATV 신호 EMA와 EMB가 모두 1 일 때, 즉 bypass 모드의 데이터가 입력될 때 처음 입력데이터가 13클럭 후에 그대로 bypass 됨을 확인 하였다.

V. 결론

HD-MAC 수상기의 캐스케이드 인터플레이터에 사용될 raster형 3×8 FMH(FIR-Median-Hybrid) 필터를 설계하고 검증하였다. HD-MAC 각 모드의 신호처리 방식의 공통점과 차이점을 분석하여 최적의 설계가 이루어지게 하였다. FMH 필터의 사용 알고리즘은 인터플레이션 하려는 픽셀의 주위에 있는 8점의 FIR값과 가장 가까운 위치에 있는 네점의 값으로 부터 중앙값을 찾는 것이다. 이 알고리즘으로 재생한 화면은 5×7 FIR 필터로 인터플레이션한 결과와 큰 차이가 없었다.

다섯개의 샘플값에서 중앙값을 찾는 메디안 필터의 알고리즘은 비교 횟수가 가장적은 Hadian-Sobel 알고리즘을 개조하여 사용하였다. Hadian-Sobel 알고리즘을 그대로 사용하면 여러 클럭기간 동안에 걸쳐 동작이 수행되어야 하므로 중간 계산결과들을 저장해야 할 기억소자들이 많이 사용되므로 하드웨어 부담이 크다. 본 연구에서는 다섯 샘플간의 크기를 비교한 비교기의 출력들의 조합으로부터 어떤 입력이 메디안이 된다는 신호를 만들었다. 이런 식으로 하면 비교기는 많이 필요하지만 한 클럭안에 모든 것이 끝나기 때문에 중간 계산결과를 저장하는 플립플롭이 줄어들어 로직소자의 갯수를 많이 줄일 수 있다.

요구되는 기능이 한 클럭동안에 이루어질 수 없기 때문에 파이프라인 구조를 이용하여 13클럭에 걸쳐 신호처리가 이루어지게 하였다. 로직은 1.0 μm CMOS 표준셀을 이용하여 설계하였다. 설계된 로직을 ASIC 툴을 이용하여 배치배선을 한 후 최악의 지연시간을 추출하여 휘도신호 및 색차신호의 각 모드를 시뮬레이션한 결과 요구되는 클럭기간 37nsec 안에 모든 기능을 수행할 수 있음을 확인했다. 본 연구에서 이용한 개념들은 다른 HDTV방식에도 이용될 수 있으리라 생각된다.

參 考 文 獻

[1] Neal C. Gallagher, Jr., Gary L. Wise, "A Theoretical Analysis of Properties of Median Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 6 pp 1136-1141, Dec. 1981.

[2] Thomas A. Nodes, Neal C. Gallagher, Jr., "Median filters : Some Modifications and Their Properties," *IEEE*

Trans. Acoustics, Speech, and Signal Processing, vol. ASSP-30, no. 5, pp 739-746, Oct. 1982.

[3] Thomas A. Nodes, Neal C. Gallagher, Jr., "The output Distribution of Median Type Filters," *IEEE Trans. Communications*, vol. COM-32, no. 5, pp 532-541, May 1984.

[4] Ari Nieminen, Pekka Heinonen, and Yrjo Neuvo, "A New class of Detail-Preserving Filters for Image Processing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp 74-90, Jan. 1987.

[5] Pekka Heinonen, Yrjo Neuvo, "FIR-Median Hybrid Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-35, no. 6, pp 832-838, Jun. 1987.

[6] Arto Lehtonen and Markku Renfors, "Nonlinear Interpolation Filtering," *SPIE*, vol. 1360, pp 135-142, Visual Communication and Image Processing '90.

[7] Thomas S. Huang, George J. Yang, and Gregory Y. Tang, "A Fast Two-Dimensional Median Filtering Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-27, no. 1, pp 13-18.

[8] Masahide Abe, Hideki Kokubun, Tahito Aida, Katsuyuki Goto, and Ki-Ichi Kobayashi, "A High-Speed Digital Filter LSI for Video signal Processing," *IEEE Journal of Solid-State Circuits*, vol. sc-32, no. 3, pp 396-402, Jun. 1987.

[9] Christophe Joanblang, Patrice Senn, and Marie-Jean Colaitis, "A 54-MHz CMOS Programmable video signal Processor for HDTV Applications," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp 730-734, Jun. 1990.

[10] F. W. P. Vreeswijk, W. Jonker, J. R. G. M. Leenen, and J. van der Meer, "An HD-MAC Coding System," *Signal*

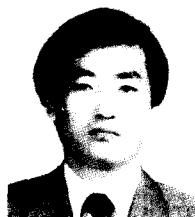
Processing of HDTV, pp 137-143, © Elsevier Science Publishers B. V., North-Holland, 1988.

- [11] 심 영석 외, 위성방송수신용 HDTV 신호처리 기 개발, 상공부(생산기술연구원), 1992. 5
- [12] Donald E. Knuth, *The Art of Computer Programming*, vol. 3, Sorting and Searching, Addison-Wesley,

Reading, Massachusetts, 1973.

- [13] A. Hadian and M. Sobel, Report 121, Univ. of Minnesota, Dept. of Statistics, May 1969. (참고문헌 [12] 의 209 페이지 참조)
- [14] 박 주성의, HDTV 디코더용 이차원 필터 LSI 설계 및 검증에 관한 연구, 연구보고서, 생산기술연구원, 1992. 8

著者紹介



朴柱成(正會員)

1976年 2月 부산대학교 전자공학과 졸업. 1978年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1989年 8月 University of Florida 전기공학과 박사학위 취득. 1978年 3月 ~ 1991年 2月 전자통신연구소 반도체 설계개발실장, 집적회로 연구부 연구위원 역임. 1991年 3月 ~ 현재 부산대학교 전자공학과 교수, 부산대학교 정보통신연구소 전자기술부장. 주관심 분야는 ASIC, DSP 설계, 반도체 소자 모델링 등임.



尹炳憲(正會員)

1987年 2月 부산대학교 전자공학과 졸업. 1989年 2月 부산대학교 전자공학과 석사학위 취득. 1989年 3月 ~ 1990年 3月 부산대학교 전자공학과 조교. 1992年 2月 부산대학교 전자공학과 박사학위 취득. 1993年 5月 ~ 현재 한국전자통신연구소 이동통신연구단 근무. 주관심 분야는 적응신호처리, 음성 코딩, 이동통신, HDTV, ASIC 등임.



諸映昊(正會員)

1992年 2月 부산대학교 전자공학과 졸업(공학사). 1994年 2月 부산대학교 전자공학과 석사학위 취득. 1994年 1月 ~ 현재 금성일렉트론 중앙연구소 연구원. 주관심 분야는 ASIC, MPU, DSP 설계 등임.



梁眞榮(正會員)

1991年 2月 부산대학교 전자공학과 졸업(공학사). 1993年 2月 부산대학교 전자공학과 석사학위 취득. 1993年 3月 ~ 현재 한국전자통신연구소 연구원. 주관심 분야는 디지털 신호처리, HDTV 전송 시스템 등임.

朴鍾哲(正會員)

1985年 2月 숭실대학교 전자공학과 졸업(공학사). 1987年 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1992年 한국과학기술원 전기 및 전자공학과 박사학위 취득. 1992年 ~ 1993年 생산기술연구원 HDTV실. 1994年 ~ 현재 금성사 LCD 연구소 선임 연구원

沈英錫(正會員)

1976年 2月 서울대학교 공과대학 전자공학과 졸업(공학사). 1982年 8月 한국과학기술원 전기 및 전자공학과 박사학위 취득. 1983年 3月 ~ 1990年 3月 경북대학교 전자공학과 부교수. 1990年 4月 ~ 현재 생산기술연구원 전자정보시스템센터 수석연구원