

4:3 데시메이션 알고리즘을 이용한 멀티플라이어리스 레터박스 변환기

한선형, 오승호, 이분기

연세대학교 전자공학과

A multiplierless Letter-box converter using 4:3 decimation algorithm

Sun-hyoung Han, Seung-ho Oh, Moon-key Lee

Dept. of Electronic Eng. Yonsei University

Abstract

This paper proposes a efficient algorithm of letter-box converter using 4:3 decimation algorithm. To display 16:9 wide images on a 4:3 screen, there is need to convert the 16:9 wide images. The letter-box converter is designed with multiplierless architecture. We have modeled the letter-box converter in Verilog-HDL and verified to show little difference between the original image and the converted image.

I. 서론

현재 유럽에서 사용되고있는 텔레비전 방송규격 중 하나인 PALplus는 유럽의 주요 방송국들에 의해 착수된 대규모 프로젝트의 산물로 현존하는 수신기와 호환 가능한 enhanced definition television system (EDTV)을 제공하고자 1990년에 시작되었다. PALplus는 PAL 컬러 코딩 구조 뿐만 아니라 625/50(라인/필드) 구조도 사용하여 '보통'의 4:3 수신기가 16:9 와이드 컬러 영상을 받아들일 수 있게 하는데 여기서 16:9 와이드 컬러 영상(576 라인)을 4:3 수신기(432 라인)가 받아들일 수 있도록 하려면 레터박스(letter-box)를 통과시켜 규격을 맞추어야 한다. 여기서 레터박스(letter-box)는 576 라인을 최종적으로 432 라인으로 된 영상에 손상을 최대한 주지 않으면서 바꾸어 주는 역할을 하여야 한다. 576 라인을 432 라인으로 바꾸기 위해서는 4:3 데시메이션(decimation) 알고리즘이 가장 적합한 형태이다. 본 논문에서는 4:3 데시메이션 알고

리즘을 제안하고 이를 VLSI로 설계하는데 있어 곱셈기나 나눗셈기등을 사용하지않고 오로지 쉬프트와 덧셈기 등을만을 사용한 레터박스 아키텍처를 소개한다.

II. 레터박스(Letter-box) 알고리즘

(1) 4:3 알고리즘

576 라인을 432 라인으로 변환하는 4:3 알고리즘에 대해 살펴보자. 이것은 4 라인을 받아들여 이것을 3 라인으로 데시메이트(decimate)하는 것으로 그림 1과 같다.

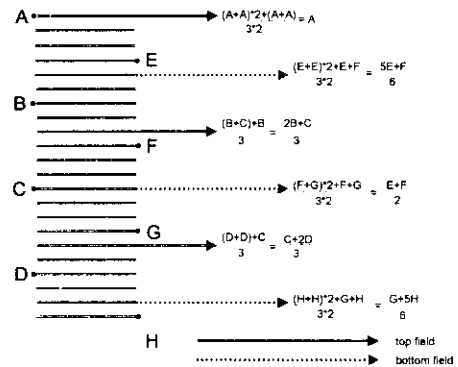


그림 1. 4:3 decimation algorithm

이 알고리즘에서는 톱(top) 필드의 경우 첫 번째 라인을 기준으로 4 개의 간격을 정확히 3등분해서 각 등분한 위치를 나머지 두 라인의 위치로 이용하였고 버텀(bottom) 필드의 경우는 톱 필드의 사이사이에 정확히 들어가도록 계수를 결정하였다.

그림 1의 알고리즘은 최적의 데시메이션(decimation) 이긴 하지만 계수를 구하는 방법에서 4 라인의 간격을 3개로 등분하려면 각 항에 나누기(/3)의 부분이 사용되어야 한다. 이 부분을 VLSI로 구현하는 것은 하드웨어 부담이 많으며 또한 게이트 수가 많이 증가한다는 단점을 가지고 있다.

(2) 5:4 알고리즘

4:3 알고리즘은 576 라인을 432 라인으로 변화 시키는 데에 있어서 최적화(optimize)한 형태이긴 하지만 나누기(/3)를 구현해야 한다는 단점을 가지고 있다. 따라서 이를 보완하기 위한 알고리즘이 5:4 알고리즘이다. 5:4 알고리즘은 5 개의 라인을 4 개의 라인으로 데시메이션(decimate) 하는 것으로 그림2와 같다.

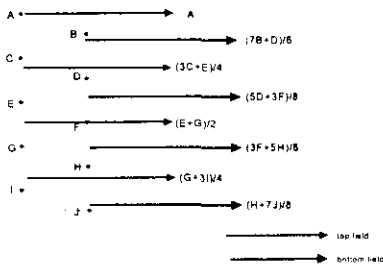


그림 2. 5:4 데시메이션 알고리즘

5:4 데시메이션 알고리즘에서는 톱(top) 필드의 경우 첫 라인을 기준으로 나머지 5개 간격을 정확히 4등분해서 각 등분점을 나머지 세 라인의 위치로 이용하였고 버텀(bottom) 필드의 경우는 톱 필드의 사이사이에 정확히 들어가도록 계수를 결정하였다. 그림2의 알고리즘에서는 톱 필드와 버텀 필드 처리과정에서 나누기(/8)이면 되기 때문에 쉬프트(shift)로 간단하게 구현가능하며 또한 게이트 수를 줄일 수 있다. 그러나 5:4 알고리즘은 576 라인을 432 라인으로 완전히 데시메이션 시키지 못하고 460 라인으로 만들기 때문에 영상이 수직으로 길게 보이는 단점이 있다.

(3) 알고리즘 제안

앞에서도 언급했듯이 576 라인을 432 라인으로 변화 시키는 데에는 4:3 데시메이션(decimation)이 된 영상을 비교적 손실 없

이 받아들이는 데에 가장 적절한 형태이다. 그러나 여기에는 곱셈기와 나눗셈기가 필요하게 되기 때문에 하드웨어로 구현하는 데에 부담이 있다. 이러한 문제를 보완한 알고리즘을 제안하였다. 본 논문에서 제안한 알고리즘은 4간격을 정확히 삼등분해서 구현한 위의 알고리즘을 기반으로 각 라인당 간격을 8등분해서 기존의 4:3 알고리즘의 계수를 최대한 근사화 시켰다. 제안한 알고리즘에서도 첫 라인을 기준으로 하고 있으며 기존의 4:3 알고리즘 계수와 오차가 최소가 되도록 나머지 라인을 배치하였고 버텀(bottom) 필드의 경우는 톱(top) 필드에서 설정된 각 라인의 중간 값과 최대한 오차가 작은 위치에 배치하였다.

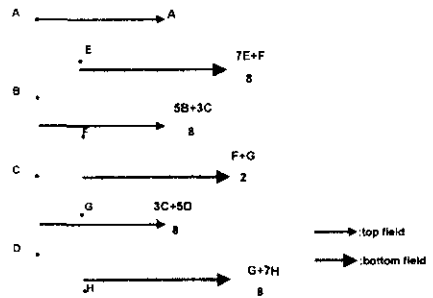


그림 3. 본 논문에서 제안한 4:3 알고리즘

각 화소(pixel)는 보통 8비트로 표시된다. 이때 8비트로는 최대 256가지의 색신호를 표시할 수 있는데 색신호의 빈도수가 0부터 255까지 일정하다고 하면 오차의 기대치는 3.56 즉 1.39% 정도이다.

III. 아키텍처

(1) 기존의 4:3 알고리즘을 구현한 회로

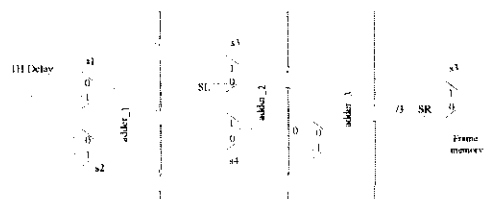


그림 4. 기존의 4:3 알고리즘을 구현한 회로

그림 4는 기존의 4:3 데시메이션 알고리즘을 구현한 회로로서 이 알고리즘은 어떻게 구현하더라도 마지막 단계 나누기 /3 부분이 생기게 된다. 따라서 회로 구현 시 하드웨어가 부담이 많다는 단점이 있다.

(1) 5:4 알고리즘을 구현한 회로

그림 5는 5:4 알고리즘을 구현한 회로로서 이는 가산기와 쉬프트, 그리고 멀티플렉서만을 이용해서 알고리즘을 구현할 수 있다.



그림 5. 5:4 데시메이션 알고리즘을 구현한 회로

여기서 회로 자체는 쉽게 구현할 수 있지만 결과적으로 영상이 세로 방향으로 길어지는 문제점이 있다.

(3) 본 논문에서 제안한 알고리즘 구현

본 논문에서 제안한 알고리즘은 4 라인을 정확하게 똑같이 등분하여 3 라인으로 만들지는 않지만 나누기 /3의 부분이 필요 없기 때문에 멀티플렉서와 쉬프트 등만으로 하드웨어로 구현하기가 용이하며 또한 영상이 수직으로 길어지는 문제점을 해결할 수 있다.

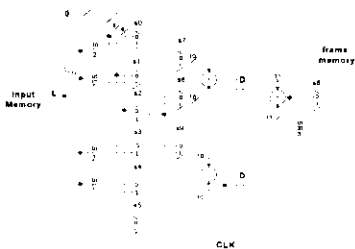


그림 6. 본 논문에서 제안한 레터박스 아키텍처(Letter-box Architecture)

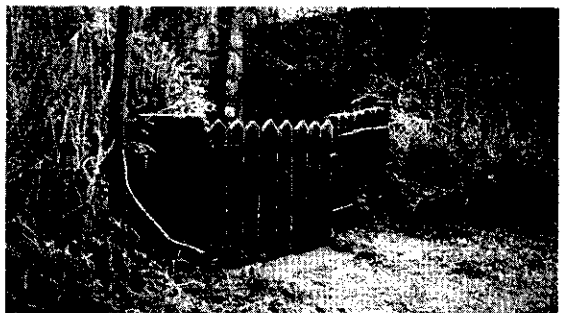
그림 6은 본 논문에서 제안한 알고리즘을

구현한 아키텍처로서 이는 10개의 컨트롤 비트 값으로 컨트롤 되며 이 값은 시스템 초기화 단계에서 롬의 데이터를 통해 할당된다. 이 구조는 이 값만 변경시키면 여러 알고리즘을 동시에 구현할 수 있다.

이 회로에서 2x2048 입력 메모리(input memory)를 사용하여 각 라인을 교대로 받아들이고 PALplus 모드가 아니면 컨트롤 핀을 바이패스 상태로 세팅(setting) 한다. PALplus 모드가 되면 두 개의 라인을 처리하는 것으로 처음 한 라인이 채워질 때까지는 출력을 내보내지 않고 하이 임피던스(high impedance)로 유지한다. 다음으로 처리할 두 번째 라인이 들어오기 시작하면 메모리 안에 있는 첫 번째 라인과 연산을 시작하게 된다. 이때 두 화소(pixel) 사이의 연산이 끝나면 그 결과는 출력 메모리(output memory)에 차례대로 한 라인씩 저장하게 된다. 이 회로는 가산기(adder)와 가산기 사이에 래치(latch)를 첨가시켜 동기를 맞추게 된다. 입력 메모리(input memory)와 출력 메모리(output memory)의 크기는 라인 당 화소 수가 PALplus 모드의 경우 최대 1152개까지 처리할 수 있어야 하기 때문에 8바이트 x 2048를 사용하였다.

IV. 검증

본 논문에서 제안한 4:3 데시메이션 알고리즘을 영상크기가 1600 x 900인 영상을 C언어를 이용하여 검증하였다. 그림 7은 이를 처리한 결과로 비교해보면 본 논문의 알고리즘을 통해 데시메이션 시킨 영상이 원영상과 눈으로 식별할 만한 차이가 보이지 않는 것을 알 수 있다.



(a) 원 영상



(b) 제안한 4:3 테시메이션 방식으로 처리한 영상

그림 7. 영상의 시물레이션

이로써 앞서 계산한 오차(1.39%)는 원 영상의 25%를 줄여야 하는 테시메이션 처리에 크게 영향을 주지는 않는다는 사실을 확인하였다.

IV. 결론

PALplus 모드에서 letter-box를 사용하는 최대 목적은 576 라인을 432 라인으로 최대한 원 영상을 훼손하지 않고 변환하는 것이다. 이러한 취지에서 생각해 볼 때 테시메이션 방법은 576:432, 즉 4:3 알고리즘이 가장 적합하다. 본 논문에서는 가장 적합한 테시메이션 알고리즘을 사용하여 가산기와 쉬프터, 멀티플렉서만으로 회로를 간단하게 구현하였다. 기존의 4:3 테시메이션 알고리즘은 테시메이션 성능은 우수하지만 회로가 복잡하고 5:4 테시메이션 알고리즘은 회로는 간단하지만 테시메이션 처리를 한 영상이 원 영상보다 세로축으로 길어지게 된다. 하지만 본 논문에서 제안한 알고리즘은 4:3의 테시메이션을 충실히 수행하며 회로도 간단하게 구현할 수 있다. 이 구조는 Verilog-HDL로 모델링하여 회로를 검증하였으며 C 언어를 이용하여 여러 테시메이션 알고리즘을 검증하였다. 본 논문에서 제안한 레터박스 아키텍처의 응용분야로는 PALplus 모드를 처리하는 부호화기, DVD 등에 사용할 수 있을 것이다.

참고문헌

- [1] K. Jack, Video Demystified : A Handbook for the Digital Engineer, HighText, 1996
- [2] "Encoding Parameters of Digital Television for Studios", CCIR Recommendation 601-2.
- [3] S. Oh, H. Choi, S. Kwon and M. Lee, "A Fully Programmable Systolic Pipelined Digital Video Encoder for NTSC/PAL/PALplus compatibility on a 4:3 Screen", IEEE Transactions on Consumer Electronics, Vol. 43, No. 3, pp. 965-971
- [4] Philip E. Mattison, "Practical Digital Video with Programming Examples in C", John Wiley & sons, Inc., 1994
- [5] P. Liuha, P. Pohjala, P. Vanni and J. Nieminen "Implementation of PALplus Decoder with Programmable Video Signal Processor". IEEE Trans. Circuits and Syst. Video Technol., vol. 1, pp. 429-435, October, 1995.
- [6] J. Jeong and B. Jeon, "A Multiplierless Letter-Box Converter for Displaying 16:9 Images on a 4:3 Screen", IEEE Trans. Circuits and Syst. Video Technol., vol. 5, NO. 4, pp. 363-366, August, 1995.