선형 보간을 이용한 컬러 버스트 동기화 기법

최종필
한국산업기술대학교
jpchoi@kpu.ac.kr

Color Burst Synchronization Technique Using Linear Interpolation

Jongpil Choi
Dept. of Computer Eng., Korea Polytechnic University

요 약

아날로그 NTSC 비디오 디코더 신호를 디코딩하여 디지털화된 컬러 값을 얻기 위해서는 컬러 버스트 신호를 동기화 해야 한다. 이 버스트 신호를 이용하여 Y, I, Q의 값을 분리하기 때문이다. 아날로그 디코더의 경우에는 내부에 버스트 신호와 동기화 한 클럭을 PLL이나 DLL 등을 이용하여 발생시켜서 I의 위치를 알아낸다. 비디오 신호 해독을 위한 전용의 PLL을 위해 아날로그 방식의 VLSI 설계를 하는 것은 많은 노력이 들어갈 뿐만 아니라 특정 Fab에 종속되어 전체 칩의 이식성을 떨어뜨리게 된다. 본 논문에서는 아날로그 PLL이 없이도 디지털 입력데이터의 산술 연산을 통해서 컬러 버스트 동기화를 검출하는 방법을 제안한다.

1. 서 론

아날로그 NTSC 비디오 디코더 신호를 디코딩하여 디지털화된 컬러 값을 얻기 위해서는 컬러 버스트 신호와 동기화되어서 일치하는 신호를 만들어야 한다[1]. 이 동기화된 버스트 신호를 이용하여 Y, I, Q의 값을 분리하기 때문이다. TV 신호는 Y의 값에 I와 Q값이 번갈아 가며 나타나기 때문에 버스트 구간에서 I의 위치를 반드시 알아내야 한다. 아날로그 디코더의 경우에는 내부에 버스트 신호와 동기화 한 클럭을 PLL이나 DLL 등을 이용하여 발생시켜서 I의 위치를 알아낸다[2]. 비디오 신호 해독을 위한 전용의 PLL을 위해 아날로그 방식의 VLSI 설계를 하는 것은 많은 노력이 들어갈 뿐만 아니라 특정 Fab에 종속이 되어 전체 칩의 Fab 이식성이 좋지 않게 된다. 본 논문에서는 아날로그 PLL을 사용하지 않고 디지털 입력 데이터의 산술 연산을 통해서 컬러 버스트 동기화를 검출하는 방법을 제안한다. 2절에서는 사용된 알고리즘을 소개하며, 3절에서는 설계와 구현 방법을 언급하며 4절에서 결론을 맺는다.

2. 알고리즘

디지털 방식의 비디오 디코더는 정해진 주파수로 이미 샘플링된 데이터를 이용하기 때문에 I의 정확한 위치를 알아내지 못하며, 정확하지 않은 I의 값을 이용하면 올바른 컬러 신호를 얻을 수 없다. 동기화되지 않은 $4f_{sc}$의 클럭을 이용하여 아날로그 비디오 입력 신호를 샘플링하면 컬러 버스트 구간에서 한 사이클당 4개의 샘플이 얻어지는데 이 샘플들이 사인파의 어느 위치에 해당될지는 알 수 없다.

이 샘플들 사이에서 원래 버스트 신호의 최대 정점의 위치를 알아내야 한다. 그림 1은 샘플값이 얻어지는 경우를 구분한 것이다. 먼저 컬러 버스트 구간 상에서 최대인 디지털 값을 구한다. 최대값의 위치를 dBurC라고 하고, 최대값 바로 이전의 샘플값을 dBurP, 바로 다음의 샘플값을 dBurN이라고 하자. (A)는 최대값의 바로 이전 샘플값과 바로 다음 샘플값이 같을 때를 뜻한다. 이 경우에는 dBurC의 위치가 사인파의 최고 정점이 되며 버스트 사인파의 위상 파악이 완료가 되었다고 할 수 있다. 따라서 I축의 위치는 33° 떨어진 위치가 된다. 샘플간의 간격이 90° 이지만 계산을 위해 이 값을 1로 정규화 시킨 값을 dIPos라 하면 dIPos는 (1)과 같다.

dIPos = 0.3667 ---(1)

(B)는 dBurC와 dBurP의 값이 같은 경우로, 버스트 사인파의 위치는 dBurC와 dBurP의 정 중간에 위치하게 되며 I축의 위치는 (2)이 된다.

dIPos = -0.5 + 0.3667 ---(2)

(C)는 dBurC와 dBurN의 값이 같은 경우로, 버스트 사인파의 위치는 dBurC와 dBurN의 중간 위치가 되며, I축의 위치는 (3)이 된다.

dIPos = 0.5 + 0.3667 --- (3)

(D)와 (E)는 그 밖의 경우로 (D)는 실제 사인파의 위치가 (A)와 (C)의 중간에 해당하는 경우를 나타내고, (E)는

실제 사인파의 위치가 (A)와 (B)의 중간에 해당하는 경우를 나타낸다. 즉 정점의 위치 Smax는 $-45° < Smax < 45°$ 이면서 $Smax \neq 0$ 인 구간이며 (4)의 식을 사용할 수 있다.

dIPos = divA / divB / 2 + 0.3667 ---(4)

dIPos를 추정하는 이 근사식은 경계 조건인 (A)의 경우에는 divA = 0이 되어 (1)식과 같고, (B)나 (C)의 경우에는 divA = divB가 되어 (2)나 (3)식이 된다. 근사식에 따른 오차는 그림 2와 같다.

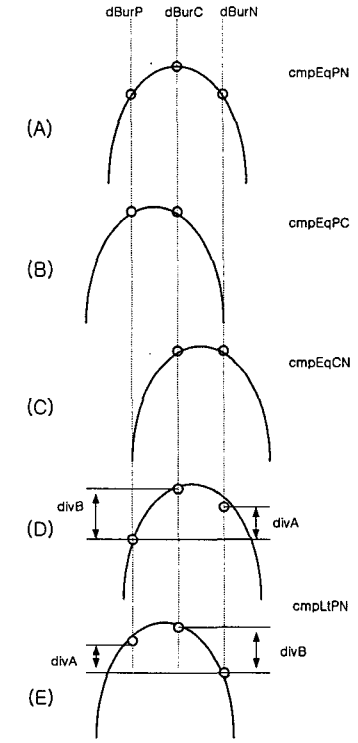

그림 1

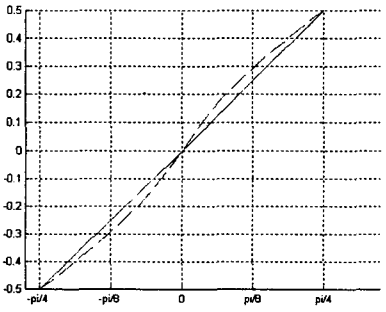

그림 2

선형 보간을 이용한 버스트 동기 검출 방법에 대한 전체 알고리즘을 플로우차트로 표현한 것이 그림 3과 그림 4이다.

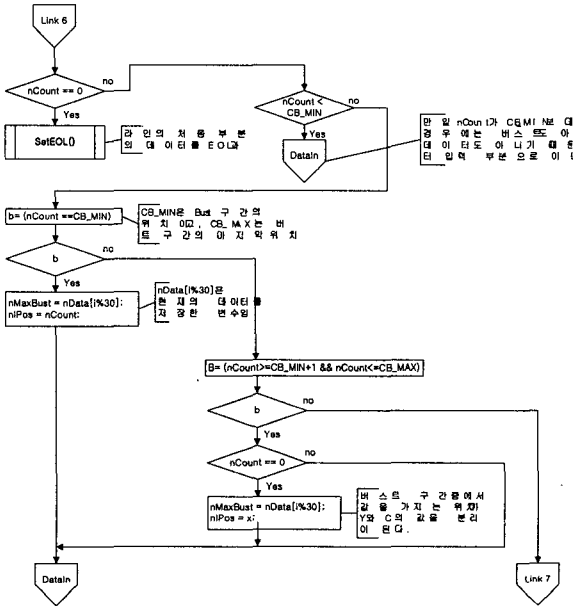

그림 3

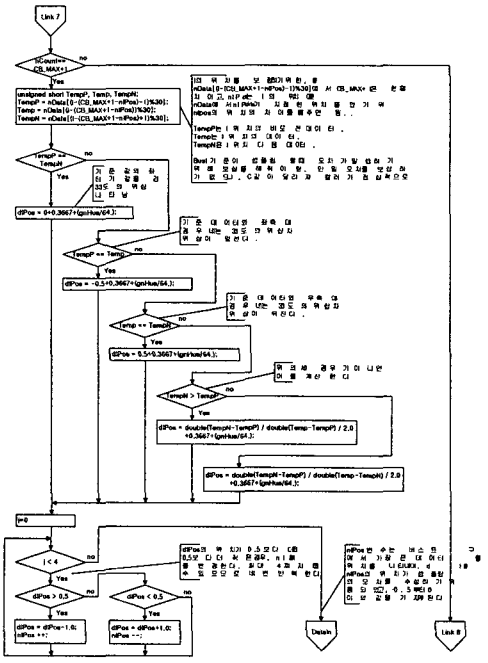

그림 4

그림 3에서는 라인 버퍼에 저장된 샘플 데이터 중에서 컬러 버스트 구간 동안 최대값을 찾기 위한 검색과 비교를 수행한다. 최대값이 발견되면 그림 4에서 앞에서 소개한 알고리즘대로 계산을 수행하여 dIPos 값을 얻는다. 위상값에 변화를 주면 색상(hue)을 조절할 수 있다.

그림 4의 마지막 부분에서는 색상(hue) 값의 조절로 dIPos 값이 -0.5 ~ 0.5의 범위를 벗어났을 경우에 이를 다시 보정하는 루프를 보인다. 0.5 만큼씩 더하거나 빼면서 인덱스의 값을 조정해주면 된다.

3. 설계와 구현

위의 알고리즘에 대한 구현에 사용되는 데이터 형은 부호 크기 표현법을 사용하였으며, 부호 크기 표현을 사용하여 선형 보간 계산에 사용되는 주요 블럭은 그림 5와 같다.

그림 5의 비교기는 dBurP와 dBurN의 값을 비교하며, 선택 먹스에 의해서 큰 값과 작은 값이 각각 선택된다. 이 값의 차이는 divA가 되며, 정규화 과정을 거치고나서 divB의 값으로 나뉘어 진다.

이 알고리즘을 NTSC 비디오 신호 뿐만 아니라 PAL 비디오 신호에도 적용하였으며, NTSC와 PAL 비디오 신호를 모두 처리하는 컬러 버스트 동기 블럭에 대한 블럭도는 그림 6과 같다.

그림 6에는 7개의 서브 블럭이 보이는데, MAXBUST, DIPOS, HUE, BURST_ADJ, PALDIP, PALPOS, PALADJ로 이루어져 있다. MAXBUST는 컬러 버스트 구간 안에서 최대값을 찾는 기능을 하고, DIPOS는 위의 알고리즘에 의해서 DIPOS를 계산하며, HUE는 색상(hue) 신호 조절을 위한 계산을 담당하고, BURST_ADJ는 그림 4의 마지막 부분에 나와있는 dIPos 값의 범위 조절을 수행한다. 나머지 PALDIP, PALPOS, PALADJ는 PAL 신호에 대해서 NTSC와 유사한 작업을 수행하는 모듈이다.

모든 회로 설계는 Verilog HDL을 이용해서 RTL 수준으로 작성되었으며 Synopsys 논리 합성기를 이용해서 게이트 레벨로 합성되었다. 사용한 라이브러리는 Hyix 0.35u 스탠다스셀이며, 합성된 결과는 19,047개의 게이트 카운트를 얻었다.

5. 결론

본 논문에서는 디지털 입력데이터의 산술 연산을 통해서 컬러 버스트 동기화를 검출하는 방법을 제안한다. 제안하는 방법은 버스트 구간에서 제일 큰 값을 가지는 데이터의 위치를 구하고, 제일 큰 값의 좌, 우에 있는 데이터의 값을 이용하여, I의 근사값을 구한 후, 이것을 이후에 나타나는 데이터에 반영하여 Y, I, Q 값을 얻는다.

본 논문에서 제안한 방법은 아날로그 PLL을 사용하지 않고 간단한 산술 연산 만을 이용함으로 설계가 쉬우며, RTL 수준의 설계를 통해서 다양한 ASIC 라이브러리를 이용해서 게이트 수준의 구현을 얻을 수 있다. 모든 부분이 디지털 회로로 구현되므로 아날로그와 디지털 혼합 모드 설계의 어려운 점을 피할 수 있는 장점이 있다.

제안된 알고리즘을 이용해서 NTSC와 PAL 비디오 신호에 적용되는 설계를 수행하였고 이를 스탠다스셀 라이브러리를 이용해서 구현하여 검증하였다.

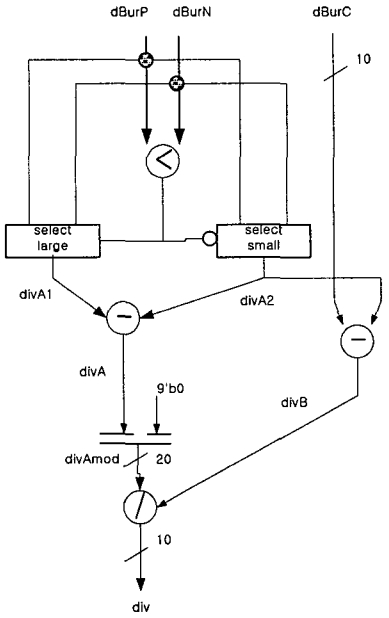

그림 5

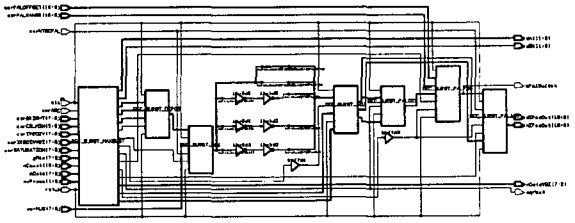

그림 6

참고문헌

[1] Arch Luther, Andrew Inglis, Video Engineering 3rd Ed. McGraw-Hill, 1999

[2] Kazushi Kitagata et al., "A New Video Front-End Processing System", IEEE Tr. on Consumer Electronics, Vol. 47, No. 2, May 2001