스플라인 보간법을 이용한 스캔 변환기

A Scan Converter Using Spline Interpolation

이범근*, 권영민*, 정연모*

Beom-Geun Lee, Young-Min Kwon, Yunmo Chung

Abstract

The purpose of format conversion is to convert a wide range of personal computer video formats into a target format. Circuits for the conversion have been developed by means of interpolation techniques, such as zero-order interpolation, bilinear interpolation, and bisigmoidal interpolation. This paper proposes a scan converter using cubic splines. The converter was modeled in VHDL, simulated on Max+PlusII, and implemented with an FPGA chip. The circuit gives much better conversion performance than a scan converter with zero-order or linear interpolation techniques according to simulation results and implementation.

* 경희대학교 전자공학과

1. 서 론

PC에서 출력되는 비디오 신호 모드는 VGA, SVGA, XVGA등과 같이 다양한데 비해서 프로젝션 모니터를 포함한 TV 모니터는 위의 모드들 중에서 하나 또는 NTSC, PAL 모드로 제한이 됨에 따라 PC의 영상을 TV 모니터로 출력하는데 문제가 발생한다. 따라서 이와 같이 PC에서 나올 수 있는 다양한 비디오 신호 포맷을 변환시켜주는 스캔 변환기에 대한 연구와 개발이 요구되고 있다. 즉, PC로부터 입력되는 다양한 종류의 비디오 신호를 원하는 디스플레이 장치에 맞는 포맷으로 변환시키는 것이 필요하며, 이와 같은 기능을 수행하기 위한 것이 스캔 변환기(scan converter)이다[2,4].

지금까지의 스캔 변환기는 입력과 출력 비디오 신호 포맷간의 차이에서 생기는 여분의 신호를 일정한 간격으로 빼는 방식을 사용하고 있다. 여기서는 변환된 비디오 신호는 계단 모양의 층이 생기는 문제점을 가지고 있다. 이러한 문제를 개선하기 위해서 다양한 보간법을 적용한 스캔 변환기에 대한 연구가 활발하게 이루어지고 있다.

기존의 영차 보간법(zero order interpolation)이나 선형 보간법(linear interpolation)을 이용한 스캔 변환기는 윤곽선에서의 원하지 않는 계단형 일그러짐 현상과 윤곽선의 흐려짐 현상이 나타난다. 본 논문에서는 두 포인트간의 값들의 연관성을 보다 더 정확히 계산할 수 있는 스플라인 보간법(spline interpolation)을 적용하여 위 문제를 최소화할 수 있는 변환 기법을 제안하였다.

이 기법은 VHDL을 사용하여 모델링하였으며 Max-Plus II상에서 시뮬레이션하였다. 또한 FPGA 칩을 사용하여 구현하였다.

2. 보간법

스캔 변환 과정에서 포맷의 불일치로 인한 새로운 픽셀의 생성이 필요할 경우 주변의 픽셀들을 참조하여 새로운 픽셀을 만드는 과정에서 보간법을 이용한다. 스캔 변환과정에서 사용되는 보간법으로는 영차 보간법, 선형 보간법, 양선형 보간법(bilinear interpolation) 등과 본 논문에서 구현하고자 하는 스플라인 보간법에 대해서 구체적으로 설명하기로 한다.

2.1 영차 보간법

출력 픽셀이 생성될 위치에서 가장 가까운 원시 픽셀을 할당하는 방법이다. 이 방법은 간단하기 때문에 처리속도 및 하드웨어 구현이 쉽지만 가장 가까운 픽셀을 할당함으로써 원래의 화상이 크게 변하는 결과를 가져올 수 있다. 새로운 픽셀값이 계산될 수 없기 때문에 모든 출력에 대응하는 픽셀값들은 입력 픽셀에서 찾을 수밖에 없다. 따라서 출력 픽셀은 픽셀 단위들만큼의 오류를 가질 수 있다. 따라서 톱니 모양으로 알려진 뭉툭함(blockiness)이 출력에 나타날 수 있다.

2.2 선형 보간법

선형 보간법은 단순히 두 점에 의하여 이루어지는 직선을 이용하여 그 사이 값들을 얻어내는 방식이다. 임의의 두 점을 지나는 직선의 방정식을 사용하여 보간된 픽셀값을 구한다. 선형 보간법은 적용 가능 범위내의 값들이 선형적으로 변하는 경우에만 좋은 결과를 얻을 수 있으며 그 외의 대부분의 경우는 왜곡된 결과 값을 가져오기 쉽다. 하지만 구현이 쉬운 관계로 인하여 여러 하드웨어에서 보간값을 구하고자 할 때 많이 사용된다.

2.3 스플라인 보간법

3차 스플라인 보간법에서는 이웃하는 데이터 사이에 3차 다항식이 사용된다. 변하고자 하는 기준점을 일정간격이 떨어진 좌우 점의 값과 기울기를 이용하여 3차 다항식을 구하는 방식이다. 이 다항식은 4개의 미정 계수가 있으므로 4

가지 조건을 필요로 한다. 4개의 미정 계수들 중 2개는 이 다항식이 두 점을 반드시 통과해야 한다는 조건에서 구할 수 있고 나머지 2개는 1차 미분, 2차 미분한 값이 주어진 두 점에서 연속해야 한다는 사실로부터 구할 수 있다.

2.3.1 하드웨어 적용을 위한 3차 스플라인 보간법의 간략화

$a \le x \le b$ 구간에서 $f(x)$가 정의되었을 때 이를 $g(x)$로 근사화 한다고 할 때,

$a = x_0 < x_1 \cdots < x_n = b$

라고 가정하고 다음 두 가지 조건을 고려한다.

[조건 1]

$$\begin{aligned} g(x_0) &= f(x_0) = f_0, \\ g(x_1) &= f(x_1) = f_1, \\ &\vdots \\ g(x_n) &= f(x_n) = f_n \end{aligned} \quad \cdots\cdots(1)$$

여기서, $g(x)$는 $a \le x \le b$ 구간에서 여러 번 미분 가능해야 한다.

[조건 2]

$g'(x_0) = k_0$, $g'(x_n) = k_n$ (주어지는 값) …(2)

$k_1, k_2, \cdots, k_{n-1}$ (나중에 계산되어질 값)

여기서 식 (1)과 식 (2)를 만족하는 유일한 다항식 3차 함수 $P_j(x)$는 다음과 같다.

$$P_j(x) = a_{j0} + a_{j1}(x-x_j) + a_{j2}(x-x_j)^2 + a_{j3}(x-x_j)^3 \quad \cdots\cdots(3)$$

여기서 계수는 아래의 식으로 구해진다.

첫째,

$$k_{(j-1)} + 4k_j + k_{(j+1)} = \frac{3}{h} \cdot (f_{(j+1)} - f_{(j-1)}) \cdots(4)$$

$j = 1, \cdots, n-1$ (n : interval의 개수)

둘째,

$$a_{j0} = P(x_j) = f_j \quad \cdots\cdots(5)$$

$$a_{j1} = P'(x_j) = k_j \quad \cdots\cdots(6)$$

$$a_{j2} = \frac{1}{2} \cdot P_j''(x_j) = \frac{3}{h^2} \cdot (f_{j+1} - f_j) - (1/h) \cdot (k_{j+1} + 2k_j) \quad \cdots(7)$$

$$a_{j3} = \frac{1}{6} \cdot P_j'''(x_j) = \frac{2}{h^3} \cdot (f_j - f_{j+1}) - \frac{1}{h^2} \cdot (k_{j+1} + k_j) \quad \cdots(8)$$

위의 식을 간략화한 3차 스플라인 식으로 표시할 경우

(오른쪽 함수)

$$f(R) = Y_0 + K_1 \cdot X_1 + A_{12} \cdot X_1^2 + A_{13} \cdot X_1^3 \quad \cdots(9)$$

(왼쪽 함수)

$$f(L) = Y_0 + K_1(M - X_1) + A_{12}(M - X_1)^2 + A_{13}(M - X_1)^3 \quad \cdots\cdots(10)$$

로 표시된다.

위의 식에서 보는 바와 같이 3차 스플라인 방정식을 전개할 경우 좌우 2개의 함수가 필요하게 된다. 하지만, 연속되고 있는 점의 연속적인 3차 스플라인 방정식은 계수를 한 방향으로만 구할 경우 1개 기준점에 대하여 2개의 함수가 필요치 않고 한 개의 일정 방향 함수만을 연속하여 전체 연결된 함수를 만들어 낼 수 있다.

따라서, 하드웨어의 구성은 식(11)과 같으며, X는 변화하고 있는 수평라인을 나타낸다.

$$f(X) = Y_0 + K_1 \cdot X + A_{12} \cdot X^2 + A_{13} \cdot X^3 \quad \cdots(11)$$

이와 같은 식으로 간략화 할 경우 하드웨어 응용이 한결 간략해진다.

2.3.2 스플라인 보간 수행

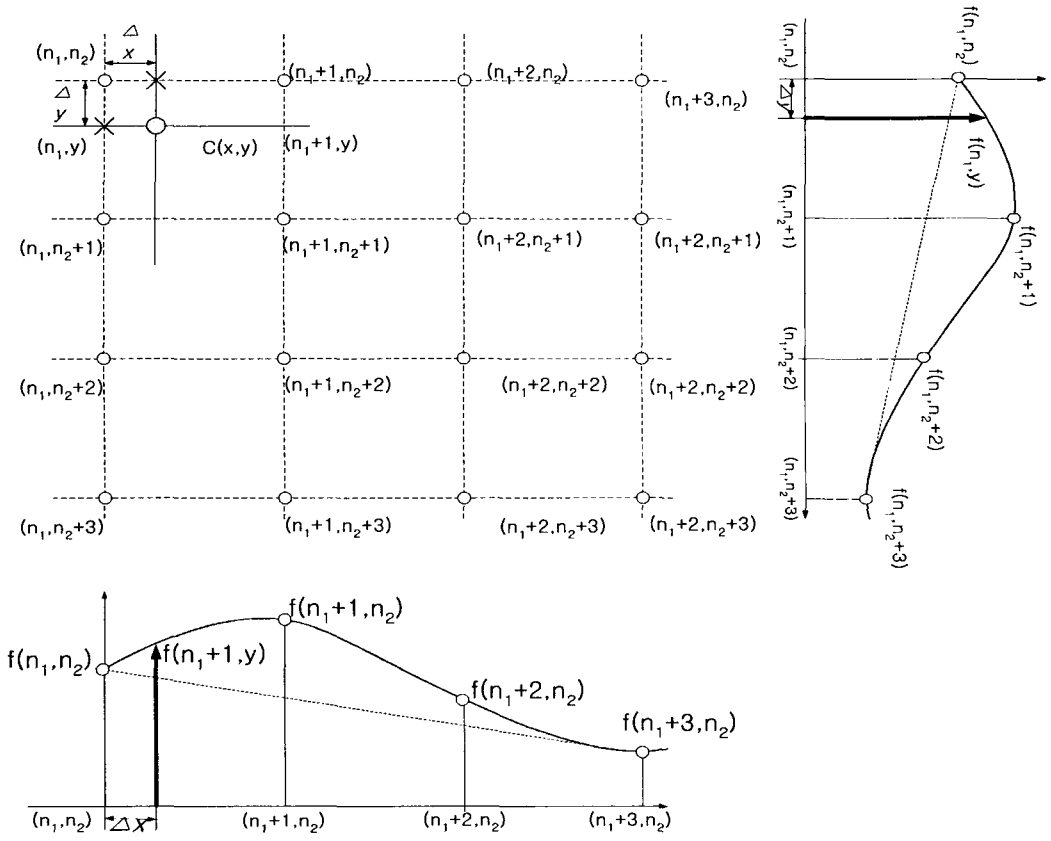
스플라인 보간의 수행은 <그림 1>과 같다. 점 C(x, y)는 목표 비디오 포맷상의 보간이 수행되어야 할 점이다. 이 점을 둘러싼 4점 (n_1, n_2), (n_1+1, n_2), (n_1, n_2+1), (n_1+1, n_2+1)을 포함한 16점에서의 픽셀 값과 위치 데이터 $\triangle x$, $\triangle y$, 그리고 참조되는 두 픽셀간의 스플라인 함수의 계수 a 값들을 계산하여 스플라인 보간이 수행된다.

먼저 각 라인의 4개의 픽셀 값들을 이용한 보간이 이루어지고 각 라인의 첫 구간에서 $\triangle x$에 따라 보간된 값들을 연산하여 수직으로 스플라인 보간이 이루어진다. 수직스플라인 함수 중 첫 구간에서 $\triangle y$를 대입하면 C(x, y)에서의 최종 보간 결과인 픽셀값 $f_c(x, y)$의 값을 구할 수 있다. 이때 다섯 번에 걸친 보간 과정에서 주어지는 계수 값들은 두 픽셀 값의 차이에 따라 각각 다르게 주어지게 된다. 즉, 참조되는 두 픽셀이 나타내는 값의 연관성에 따라 보간은 다르게 적용된다.

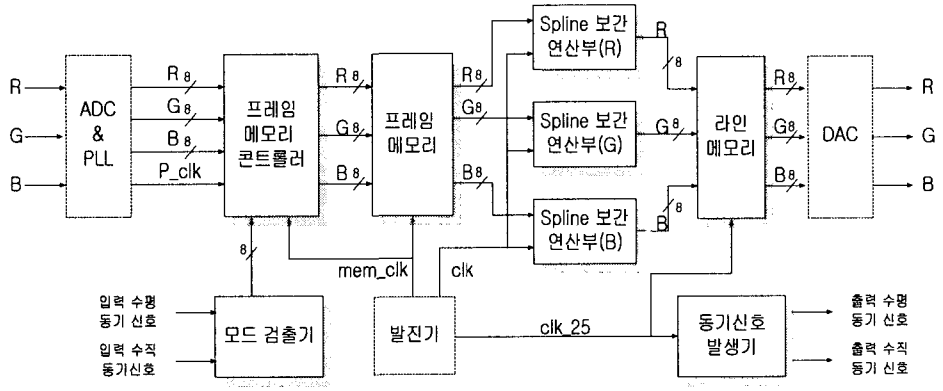
3. 구현 및 결과

스캔 변환기는 <그림 2>와 같이 다양한 비디오 신호의 모드를 검출하는 모드 검출기, 프레임 주파수 제어기(frame rate controller), 프레임을 저장하기 위한 프레임 메모리 및 라인 메모리, 스플라인 보간법을 적용시킨 스플라인 보간 연산부, 수평, 수직 동기신호를 만들어 내는 동기신호 발생기로 구성된다.

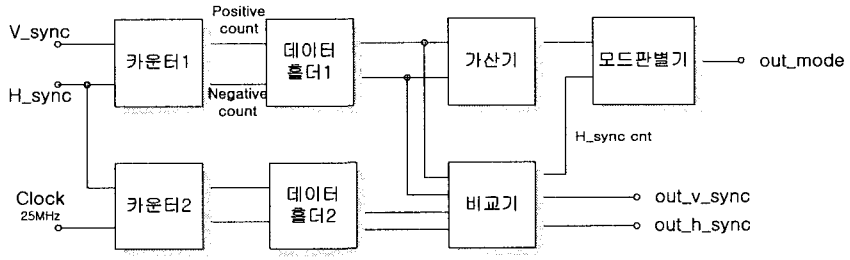
<그림 2>의 전체 블록도에 나타난 각 블록에 대해서는 아래에서 자세히 설명되며 실제 Max-Plus II 상에서 시뮬레이션을 통해서 검증된다.



<그림 1> 스플라인 보간의 예



<그림 2> 스캔변환기 전체 블록도



<그림 3> 모드 검출기

3.1 모드 검출기

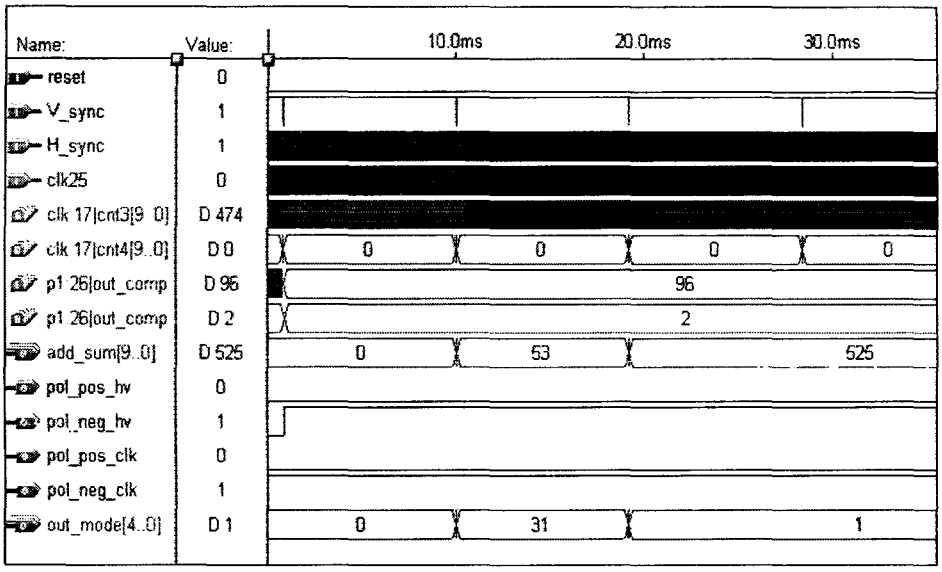
다양한 비디오 모드를 검출하기 위해 수직 동기 신호가 포함하는 수평 동기 신호의 수를 비교하고 여기에 수평 동기 신호의 sync구간의 시간을 비교하여 입력되는 신호의 모드를 검출한다. 그리고, 입력되는 수직 동기 신호와 수평 동기 신호의 극성을 비교하여 스캔 변환기에서 필요로 하는 극성으로 바꾸어준다.

<그림 3>은 스캔변환기로 입력되는 퍼스널 컴퓨터의 다양한 비디오 모드를 검출하는 블록으로 카운터, 데이터 홀더, 가산기, 비교기, 모드 판별기로 구성된다.

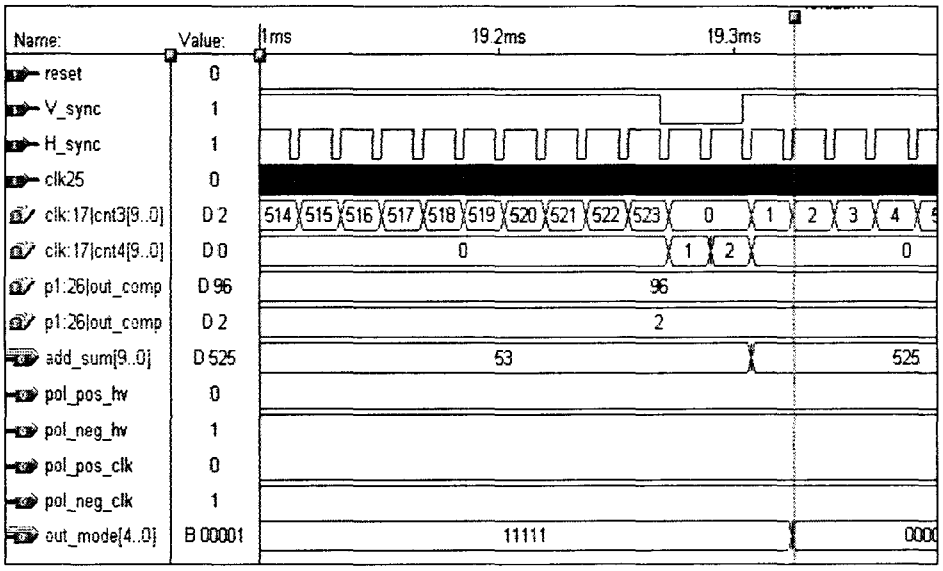
<표 1>은 모드 판별기로 입력되는 두 입력 값인 수직 동기신호의 총 라인수(수평 동기신호의 수)의 합과, 수평 동기 신호의 sync구간의 시간으로 모드를 검출하는데 있어 기준이 되는 표이며, 오차 범위를 고려하여 판별한다.

<표 1> 비디오 모드 검출을 위한 신호 특성

해상도	표준	Frame rate (Hz)	V total (lines)	V_sync 내의 line수	H_sync	
					시간 (μs)	Clock 수 (25MHz)
640x480	IBM	60	525	2	3.813	95.3
	VESA	72	520	3	1.270	31.7
		75	500	3	2.032	50.8
		85	509	3	1.556	38.9
	MAC	60	525	2	3.813	95.3
		67	525	3	2.116	52.9
720x400	IBM	70	449	2	3.813	95.3
800x600	VESA	56	625	2	2.000	50
		60	628	4	3.200	80
		72	666	6	2.400	60
		75	625	3	1.616	40.4
		85	631	3	1.138	28.4
832x624	MAC	75	667	3	1.117	27.9
1024x768	IBM	43.5	817	8	3.920	98
	VESA	60	806	6	2.092	52.3
		70	806	6	1.813	45.3
		75	800	3	1.219	30.4
		85	808	3	1.016	25.4



<그림 4> 모드 검출기의 시뮬레이션 I



<그림 5> 모드 검출기 시뮬레이션 2

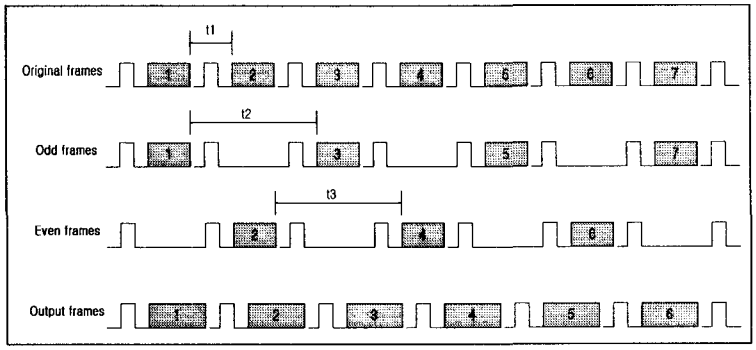
<그림 4>와 <그림 5>는 입력 모드로서 VGA 640×480을 입력하였을 때의 모드 검출 수행 파형을 나타낸다. 그림에서의 수직 동기 신호 한 주기 동안의 라인수 525개, 25Mhz 클럭으로 측정한 수평 동기 신호의 sync 구간 동안의 클럭 수가 96개이고 이를 통해 출력되는 모드 값이 1을 나타내어 <표 1>의 VGA 640×480(IBM)모드를 검출하였음을 알 수 있다. 칩의 내부 reset은 '1'값을 가질 때 전체 칩 내부의 메모리 소자들을 '0'으로 초기화하고, '0'일 때 정상적인 동작을 할 수 있도록 하였음을 볼 수 있다.

3.2 프레임 주파수 제어기

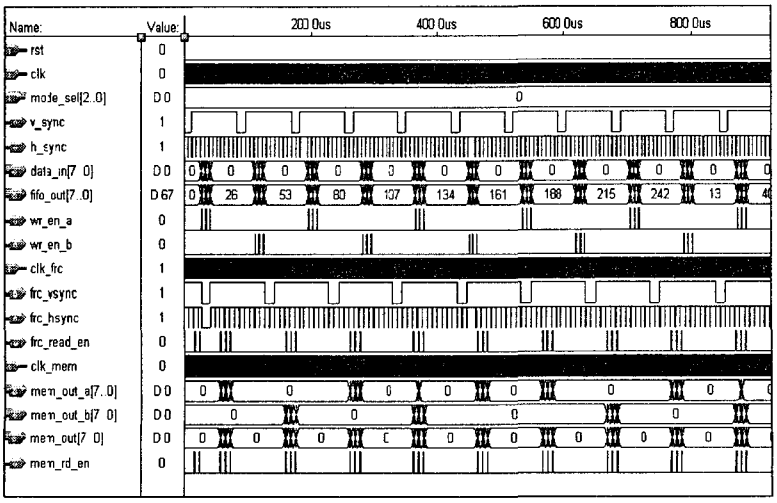
주파수 제어기는 입력되는 비디오 모드에 따라 60, 72, 75 등의 다양한 입력 프레임 주파수를 변환되는 비디오 모드의 일정한 프레임 주파수로 바꾸어준다. 데이터를 프레임 단위로 입력하고, 원하는 프레임 수로 변환하여 출력하도록 제어한다.

<그림 6>에서 보는 바와 같이 프레임 메모리로 입력되는 데이터는 연속적이므로 메모리의 읽기, 쓰기 동작의 시간적 여유를 두기 위해 짝수 번째, 홀수 번째 프레임을 위한 2개의 메모리를 사용한 메모리 인터리빙 기법을 사용하여 설계하였다. 즉, 짝수 번째, 홀수 번째 프레임 메모리에 각각 쓰기 동작이 끝나면 변화될 프레임 주파수에 맞추어 두 메모리로부터 읽기 동작이 수행된다.

<그림 7>은 주파수 제어기에 의한 프레임 주파수 변화 결과 파형으로서 입력으로 수직 동기신호인 v_sync와 수평 동시신호인 h_sync 그리고 픽셀 데이터인 data_in을 받아 FIFO와 프레임 메모리에 데이터를 저장함을 보인다. 그리고 변환하고자 하는 프레임 주파수에 맞는 수직 동기신호와 수평동기 신호인 frc_vsync와 frc_hsync가 만들어짐을 볼 수 있다. 또한 프레임 주파수 변환을 위한 마스크를 통해 프레임 메모리를 출력포트인 mem_out_a와 mem_out_b로 픽셀 데이터를 출력함을 볼 수 있다. 이때 메모리 읽기, 쓰기 동작은 메모리 동작 클럭에 의해 만들어진 제어신호에 의해 이루어짐을 볼 수 있다.



<그림 6> 프레임 주파수 제어기의 타이밍도

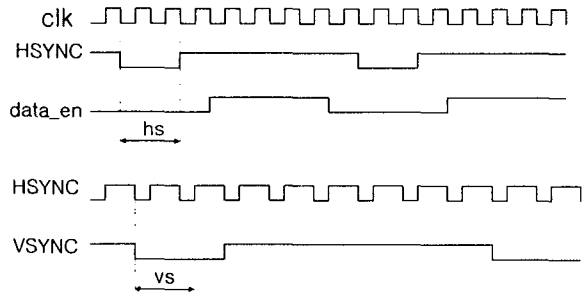


<그림 7> 프레임 주파수 제어기의 시뮬레이션

3.3 동기신호 발생기

동기신호 발생기는 동기 신호 생성에 필요한 데이터들을 가지고 있는 블록과 동기신호를 생성하는 블록으로 구성되며 동기 신호의 특성인 sync의 크기, 영상의 총 픽셀 수, 상하 porch, 좌우 porch등을 고려한 데이터들을 이용하여 수직 및 수평 동기신호를 생성한다.

<그림 8>은 클럭과 동기신호 생성에 필요한 여러 데이터들을 가지고 생성되는 hsync, data_en, vsync의 개념적인 타이밍도이다.

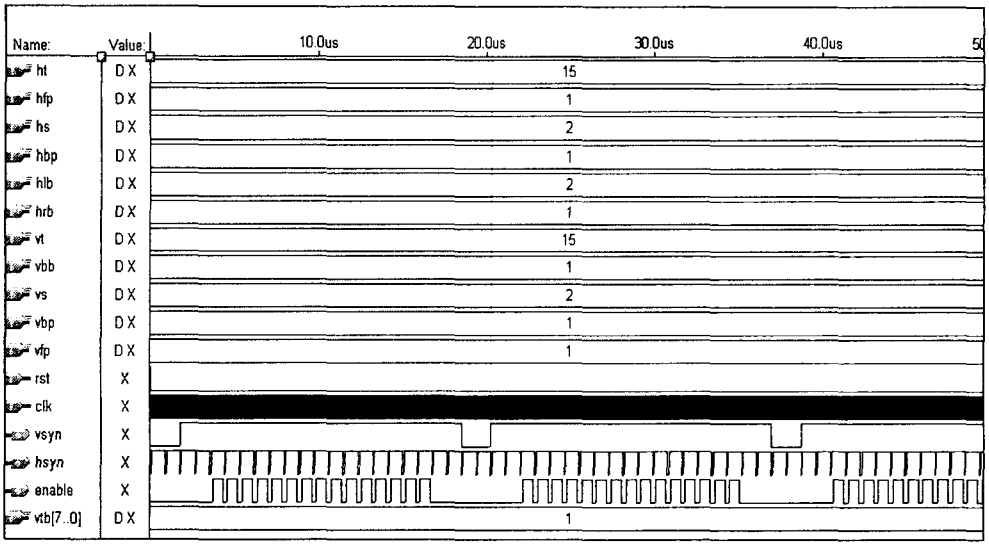


<그림 8> 동기 신호 발생기의 타이밍도

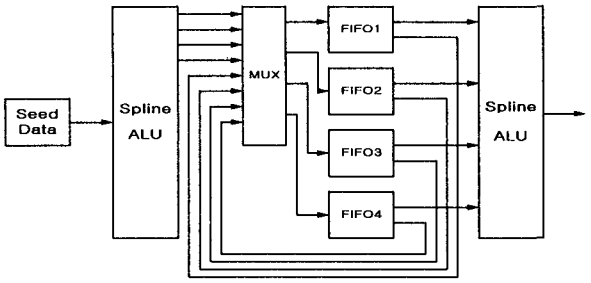
<그림 9>는 클럭과 동기신호 생성에 필요한 데이터들(ht, hfp, hs, hbp, hlb. hrb, vt, hbb, vs, vbp, vfp, vtb)등의 데이터를 실제로 사용 할 경우 수백, 수천개의 클럭을 카운트하여야 하므로 임의의 값을 주어 hsync, data_en, vsync을 생성됨을 보여준다.

3.4 스플라인 보간 연산부

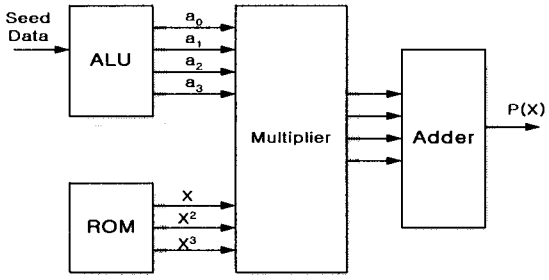
스플라인 보간 연산부는 <그림 10>과 같이 프레임 메모리로부터 순차적으로 각 라인의 데이터를 받아 라인별로 스플라인 보간을 수행하고 그 데이터 값을 각각의 FIFO에 저장한다. 처음 4개의 라인에 보간이 이루어지고 각각의 FIFO에 차례로 저장되면 다섯 번째 라인부터는 FIFO4에 저장이 되며 FIFO1를 제외한 다른 값들은 MUX를 통하여 앞에 있는 FIFO로 전달된다. 즉 여기서 FIFO2, FIFO3, FIFO4에 있는 데이터는 MUX를 통하여 FIFO1, FIFO2, FIFO3에 옮겨진다. FIFO에서 출력된 4개의 라인 데이터는 수직 보간에 이용되며 수직 스플라인 보간 연산부에서 출력되는 값이 최종 값을 나타낸다. 이러한 값들을 보간을 수행하는 두 개의 스플라인 ALU블록으로 구성된다.



<그림 9> 동기 신호 발생기 모의 실험 결과



<그림 10> 스플라인 보간 연산부

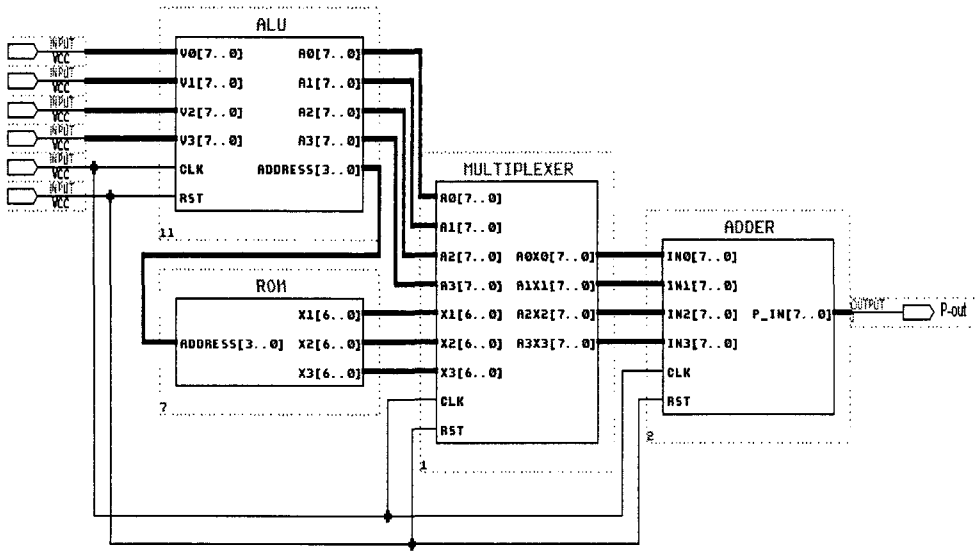


<그림 11> 스플라인 ALU 블록

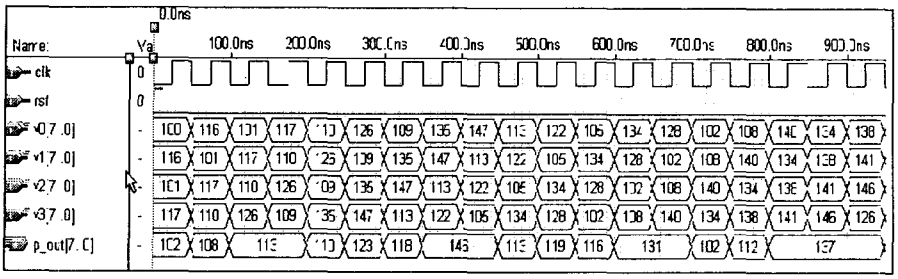
<그림 11>은 스플라인 ALU블록을 자세히 나타내었다. 이 블록의 ALU는 스플라인 함수의 계수 a의 값을 입력되는 시드 데이타(seed data)에 따라 연산하여 출력한다. 그리고 ROM 테이블은 해상도의 차이에서 발생하는 각각 픽셀의 위치 (Δx, Δy)를 계산하여 저장하였다. 수평라인이나 수직라인은 똑같은 비율로 계산되므로 x의 값만을 저장 출력되는데 이 값들은 곱셈기(multiplier)에서 계수와 곱해지고 최종적으로 각각이 대해 더해진 보간된 값이 출력된다.

<그림 12>는 스플라인 ALU블록을 스키메틱으로 나타낸 회로도이다

<그림 13>은 스플라인 보간을 위해 사용한 함수의 계수 값은 입력되는 네 점(a, b, c, d)에 의해 ALU블럭에서 연산하여 출력(p_out)하였다. 또 ALU블럭에서 Δx에 대한 값을 저장한 ROM의 주소(address)를 출력하게 하였다. 이들 값들을 이용하여 곱셈기와 덧셈기(adder)를 수행하면서 보간이 이루어진다.



<그림 12> 스플라인 ALU 블록 회로도



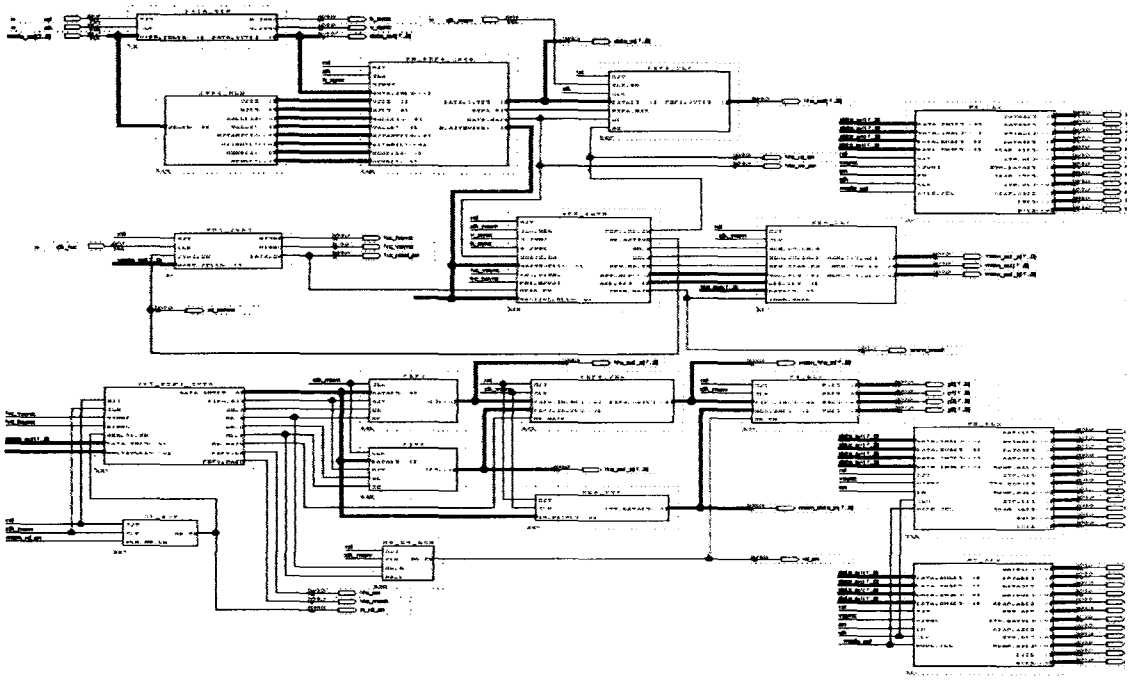
<그림 13> 스플라인 ALU 블록의 시뮬레이션

3.5 스플라인 보간을 통한 스캔 변환기

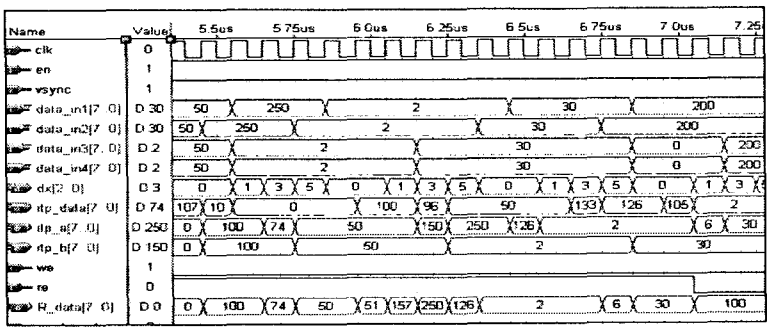
<그림 14>는 포맷 변환기를 스키메틱으로 나타낸 전체 회로도이다

<그림 15>와 <그림 16>은 800×600모드를 640×480모드로 변환시킨 결과로서 프레임 주파수 변환을 통해 입력되는 4픽셀 데이터들을 이용하여 윤곽선을 판단하고 그에 따른 보간을 각각 수행함을 보인다. 그림에서 data_in, data_in2, data_in3, data_in4의 4개의 병렬 데이터와 dx로 나타나 있는 위치 데이터를 입력으로 받는다. 이때 4개의 병렬 데이터는 그림에서 나타낸 것과 같이 경계선의 특성을 가지는 데이터(30,30,2,2)를 예를 들어 실험하였음을 볼 수 있다. 또한 data_in1, data_in2, data_in3, data_in4값을 보간하여 R-data로 출력됨을 보인다.

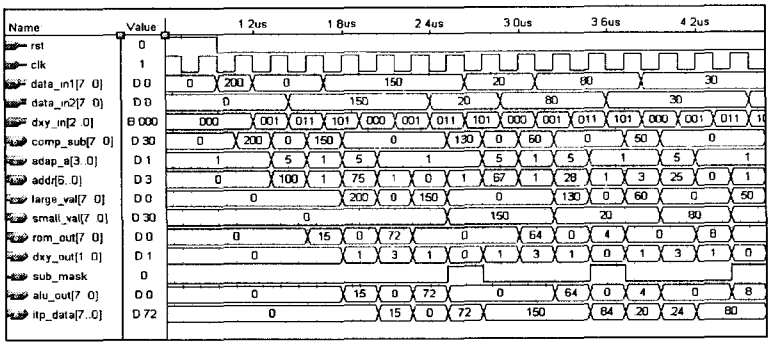
이웃 픽셀들의 연관성을 고려하여 윤곽선을 개선한 스플라인 보간법은 실험 검증을 통해 비교 분석하였다. 800×600화상을 640×480화상으로 변환시키는 실험을 위해 lena화상을 이용하여



<그림 14> 스캔변환기 전체 블록도

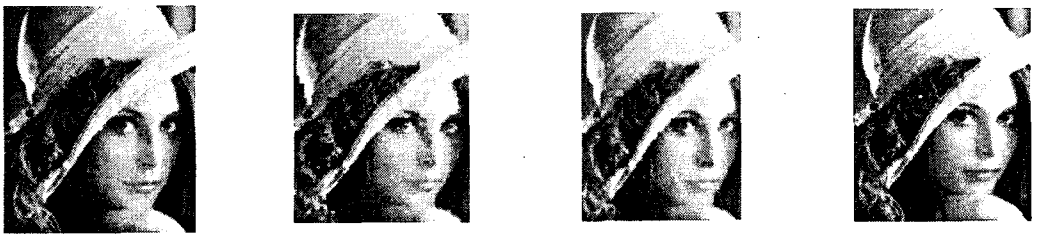


<그림 15> 스플라인 보간을 통한 스캔변환 시뮬레이션 결과 1



<그림 16> 스플라인 보간을 통한 스캔변환 시뮬레이션 결과 2

실험한 결과를 <그림 17>에서 보여주고 있다. <그림 17>의 (a)는 800×600의 원시 화상을 나타내고, <그림 17>의 (b)는 영차 보간일 때 의 보간 결과를 나타낸 것으로 참조되는 두 점들 중 가까운 값을 취함으로 인한 흐려짐 현상을 보인다. <그림 17>의 (c)는 선형보간법을 이용할 때의 보간 결과를 보인 것으로 심한 계단형 일그러짐 현상을 보인다. <그림 17>의 (d)는 스플라인 보간 결과를 나타낸 것으로 두 경우에 비해 시각적 거의 원래 화상에 가깝게 변환되는 것을 볼 수 있다.



(a) 원영상 (b) 영차 보간법을 이용한 영상 (c) 선형 보간법을 이용한 영상 (d) 스플라인 보간법을 이용한 영상

<그림 17> 보드상의 시뮬레이션 결과

4. 결 론

본 논문은 스플라인 보간법을 이용한 스캔변환기를 설계하는 것을 제안하였다. 출력 화상은 입력화상의 각각 픽셀에 대한 연관성을 판단하여 스플라인 보간을 수행하도록 하여 윤곽선 부분의 시각적인 화질 개선을 모의실험을 통해 확인 할 수 있었으며 스플라인 보간법을 이용하며 이웃하는 16 개의 픽셀 값을 참조하여 그에 해당하는 값이 출력되고 픽셀의 위치값을 ROM 테이블을 미리 저장하여 참조하도록 함으로써 기존의 화질보다 개선되고 시스템의 연산속도 또한 개선되도록 제안하였다

또한 기존의 영차 보간이나 선형 보간의 경우 보간 과정이나 설계과정이 단순하며 수행시간이 비교적 짧지만 원래 화상과 너무 차이가 나며 일그러짐이 심함을 알 수 있다. 그러나 스플라인 보간법을 이용한 스캔 변환기의 시뮬레이션인 경우 비교적 복잡한 연산과정과 설계의 어려움은 있으나 거의 원래 화상과 비슷한 수준의 화상을 출력하며 해상도 변화에 있어서 항상 문제가 되는 일그러짐 현상도 나타나지 않음을 알 수 있었다.

본 논문에서 제시한 회로는 VHDL로 모델링 하였으며 Max+Plus II 상에서 시뮬레이션하여 FPGA로 구현하였다.

참고 문헌

[1] C.G. Oh, J.G. Kim, C.W. Hong, S.P. Hong, Y. Chung, "A real time digital convergence system using interpolation," *IEEE Tran-saction on Consumer Electronics*, vol. 42, No. 3, pp. 689-695, August 1996.

[2] 이재준, 홍주선, 정연모, "A Scan Converter For Projection Monitors," CMEW'98, pp.33-38, Mar. 1998

[3] Ohsawa, et. al., "A high-resolution rear projection TV for home-use," *IEEE Transaction on Consumer Electronics*, Vol. 35, No. 3, pp. 325-333, August 1989.

[4] M. Shiomi, et. al., "A fully digital convertgence system forprojection TV," *IEEE Transaction on Consumer Electronics*, Vol. 36, No. 3, pp. 445-453, August 1990.

[5] M. Lindahl, et. al., "Digital convergence system with high perforance but simplified implementation," *Digest of Technical Papers. ICCE., Conference on, Consumer Electronics*, pp. 222-223, 1997.

[6] Y. Shimizu el. al., "Development Of A PC-NTSC Scan Converter System LSI," *IEEE Transactions on Consumer Electronics*, Vol. 42, No. 3, pp.681-688, Aug. 1996.

[7] Genesis Microchip Inc. Preliminary Data Sheet-gmFC1 DAT-0005-A

[8] L. C. Barrett, *Advanced Engineering Mathematics*, pp. 336-337, McGraw Hill, 1955.

저자소개

이범근
1995년 청주대학교 전자공학과 학사
1997년 청주대학교 전자공학과 석사
1997년 ~ 현재 경희대학교 전자공학과 박사과정
관심분야 : HDTV, CAD, 블루투스, Internet Applications

권영민
1999년 경희대학교 전자공학과 학사
1999년 ~ 현재 경희대학교 전자공학과 석사과정
관심분야 : MPEG, HDTV

정연모
1980년 경북대학교 학사
1982년 KAIST 석사
1991년 미국 미시간주립대학교 박사
1992년 ~ 현재 경희대학교 전자공학과 조교수/부교수
관심분야 : 병렬처리 시뮬레이션, CAD, VLSI, Internet Applications.