CMOS 카메라 이미지 센서용 ISP 구현

손승일* · 이동훈*

An Implementation of ISP for CMOS Image Sensor

Seung il Sonh* · Dong hoon Lee*

이 논문은 2007년도 한신대학교 연구비 지원에 의해 수행되었음.

요 약

CMOS 이미지 센서로부터 수신한 베이어 입력 스트림을 디스플레이 장치로 보기위해서는 영상 신호처리를 수행해야 한다. 즉, 이러한 영상 신호처리를 수행한 부분을 ISP(Image Signal Processor)라 한다. ISP 처리를 통해서 실제 원본 이미지를 볼 수 있다. ISP는 감마교정, 인터폴레이션, 공간적 변환, 이미지 효과, 이미지 스케일, AWB, AE, AF 등과 같은 기능을 수행한다. 본 논문에서는 CMOS 카메라 이미지 센서용 ISP를 모델링하여 최적화 알고리즘을 소프트웨어 검증을 통해 도출하였으며, VHDL 언어를 이용하여 설계하고 ModelSim6.0a 시뮬레이터를 이용하여 검증하였다. 또한 보드 레벨의 검증을 위해 PCI 인터페이스를 이용하여 설계한 ISP 모듈을 자일링스 XCV-1000e에 다운로드하여 결과를 확인하였다.

ABSTRACT

In order to display Bayer input stream received from CMOS image sensor to the display device, image signal processing must be performed. That is, the hardware performing the image signal processing for Bayer data is called ISP(Image Signal Processor). We can see real image through ISP processing. ISP executes functionalities for gamma correction, interpolation, color space conversion, image effect, image scale, AWB, AE and AF. In this paper, we obtained the optimum algorithm through software verification of ISP module for CMOS camera image sensor and described using VHDL and verified in ModelSim6.0a simulator. Also we downloaded into Xilinx XCV-1000e for the designed ISP module and completed the board level verification using PCI interface.

키워드

ISP, Interpolation, AWB, Scaler, VHDL, ModelSim

Ⅰ. 서 론

CMOS 이미지 센서는 CCD 이미지 센서보다 칩 제작 과정에서 공정비용이 저렴하고 대량생산이 용이한 장점을 갖기 때문에 최근 CMOS 이미지 센서에 대한 관심이 증대되고 있다. CMOS 이미지 센서로부터 입력 받은 영상 이미지에 대해 이미지 신호처리를 수행하여만 출력 화면을 통해 올바른 이미지를 볼 수 있게 된다[1][2].

* 한신대학교 정보통신학과

접수일자 : 2006. 3. 29

이러한 이미지 신호처리를 수행하는 프로세서를 ISP(Image Signal Processor)라 부른다. ISP는 감마교정, 컬러 인터폴레이션, 컬러정정 & 컬러 공간적 변환, 시각적 효과를 주기위해 세피아, 반전, 스케치 효과를 주어 영상을 변환하는 이미지 효과부, 영상의 확대 및 축소를 수행하는 이미지 스케일부, 자동 노출을 제어하는 AE (Automatic Exposure), 화이트 밸런스를 조절하는 AWB (Automatic White Balance), 초점을 조절하는 AF (Automatic Focus)부 등으로 구성되게 된다.

멀티미디어 관련 업체에서는 CMOS 센서에 ISP 프로세서 기능을 탑재하여 화소별로 VGA, 1.3M, 2M, 3M 급을 엠텍비젼, 픽셀 플럭스, 코아로직 등 주요 카메라 및 동영상 관련 업체에서 양산하여 카메라 휴대폰, 멀티미디어 영상 제품군에 내장하여 판매하고 있다[3][4]. 그러나 현재까지 발표된 논문은 ISP의 전반적으로 구현과 성능 평가에 대한 내용이 아니라 부분 블록의 성능 평가와 구현에 대해 다루고 있다[5-8].

특히 본 논문에서는 인터폴레이션, 이미지 스케일 및 AWB의 하드웨어 설계에 대해 자세히 다루고자 한다.

기존 ISP의 주요 모듈 기능에서 영상 인터폴레이션 수행은 3x3라인 보간을 이용하여 처리하였다[5]. 영상의 확대 및 축소를 지원하는 이미지 스케일 모듈은 화면 출력 크기를 VGA, QVGA, QQVGA 중에서 지원하였다[6]. AWB는 R, G, B 평균값과 Y의 평균값을 이용하여 AWB를 구한다. 프레임 단위의 AWB 처리에서 계수 값을 구하기 위해 2개 프레임의 연산량을 필요로 하였다[8]. 본 논문에서는 하드웨어 칩 면적과 레지스터 수를 줄일 수 있는 알고리즘을 적용하였다. ISP 주요 모듈 기능에서 칩의 면적을 줄이기 위해 인터폴레이션 수행시 2x2라인을 사용하여 구현하였다. 또한 AWB의 계수를 구하기 위해 YCbCr의 변환 데이터 포맷에서 Cb, Cr의 평균값을 구하여 1개 프레임의 연산량으로 줄일 수 있도록 하였다. 마지막으로 이미지 스케일에 보간 알고리즘에서 양선형과 비교하여 연산량 대비 이미지 화질을 개선할 수 있는 이미지 스케일을 처리하여 적용하였다. 이러한 ISP 처리기능을 수행하여 하드웨어 설계시 저전력, 저비용을 고려한 설계를 진행하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 ISP 전체 블록도의 개요를 설명하고 Ⅲ장에서는 ISP 주요 모듈의 기능 처리 방법에 대해 알아보고 Ⅳ장에서는 제안한 ISP 알고리즘을 적용한 하드웨어 설계를 진행한다. Ⅴ장에서는 설계한 ISP 주요 모듈의 기능블록의 성능평가를 진행하고 Ⅵ장에서는 설계한 ISP 모듈의 데이터를 검증 하기위해 PCI 인터페이스를 이용한 보드 레벨의 검증을 진행한 후 Ⅶ장에서 결론을 맺는다.

Ⅱ. ISP 전체 블록도

다음 그림 1은 본 논문에서 구현한 ISP의 전체 블록도를 보여주고 있다.

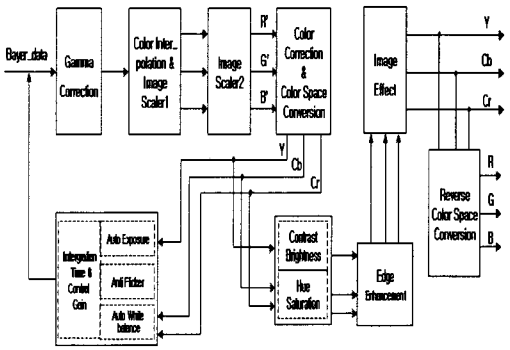

그림 1. ISP 전체 블록도
Fig. 1 Block diagram for the Designed ISP

ISP 블록도 처리 순서는 다음과 같다. 감마교정 블록, 스케일링을 포함하는 컬러 인터폴레이션 블록, 컬러교정&컬러 공간적 변환 블록, 3 Auto Focus(Auto Exposure, Auto White Balance, Auto Focus) 블록, Contrast/Brightness 블록, Hue/Saturation 블록, 엣지 향상 블록(Edge Enhancement), 이미지 효과 블록, 역 컬러 공간 변환(Reverse Color Space Conversion) 블록으로 그림 1에서와 같이 ISP를 처리 절차 순서로 진행한다[1][2].

Ⅲ. ISP 주요 모듈의 기능 처리방법

3.1 2x2라인 인터폴레이션 처리

베이어 입력 스트림은 행 단위의 데이터로 입력을 받는다. 그림 2은 1개의 라인 FIFO를 사용하고 4개의 Z^{-1} 전송 버퍼를 사용하여 라인 단위의 픽셀 보간을 계산하여 새로운 픽셀 R, G, B값을 구하게 된다. 그림 2에서 사용하는 라인 FIFO를 한 개만을 사용하며, 전송 버퍼의

수를 줄이면서 영상의 효과적인 처리를 수행 할 수 있다. 다음 그림 2는 2x2라인 인터폴레이션 처리방법을 나타내고 있다[2][5].

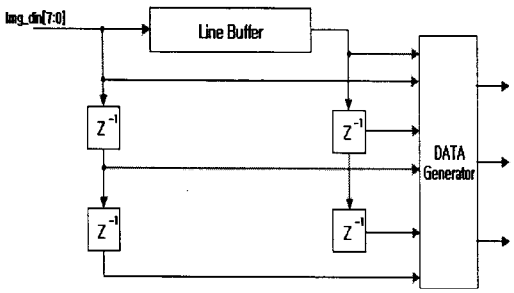

그림 2. 2x2 라인 인터폴레이션 처리
Fig. 2 2x2 line interpolation processing

3.2 이미지 스케일 처리

다음 그림 3은 원본 이미지와 새로운 이미지의 크기를 카운트 하기위한 Resize 카운트 블록, 원본 이미지 픽셀의 램덤 액세스를 위한 한 라인에 대한 이중 포트 RAM을 사용하여 쓰고, 읽기를 효과적으로 사용할 수 있다.

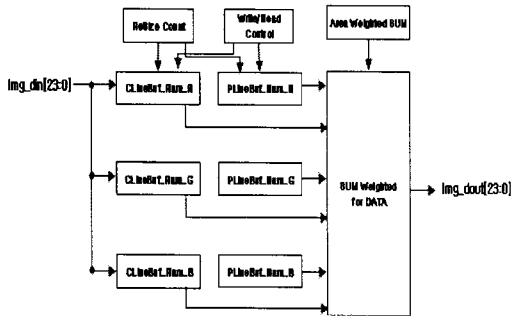

그림 3. 이미지 스케일 블록
Fig. 3 Image Scale Block

RAM을 액세스하기 위한 Write/Read Control블록, 원본 이미지 픽셀 수를 최대 0~4개 사용 할 수 있으며, 각 원본 이미지에 분할 영역의 가중치 합을 계산하기 위한 Area Weighted SUM블록, 최종 데이터의 정렬 가중치 합을 통해 새로운 이미지를 발생하는 가중치 합 데이터 블록으로 구성되어 있다[8][9].

3.3 AWB 처리

AWB를 수행하기위해 특정 색상 Cb, Cr의 픽셀값에 대한 임계치 범위를 체크하여야 한다. 그림 4에서 점선에 포함하는 블록이 Cb, Cr의 허용 임계치를 계산하기위한 블록이다. Cb, Cr의 값은 경계 데이터 체크 블록에서 허용 임계치에 들어오면 AWB를 수행하지 않고, 허용 임계치를 벗어난 경우 AWB를 수행한다. Y의 값은 AWB에 거의 사용하지 않으므로 Y_Line_FIFO블록을 통해 원하는 결과 시점에 데이터를 출력한다. 또한 다중 영역 분할 방법을 사용하여 각 분할 영역에 가중치를 두고 있다. 분할 영역에 가중치가 적용된 값으로 히스토그램 방법을 사용하여 다분할 AWB 프로세서 블록을 통해 Cb, Cr의 값이 허용범위에 들어오도록 데이터 처리를 수행한다[8][9].

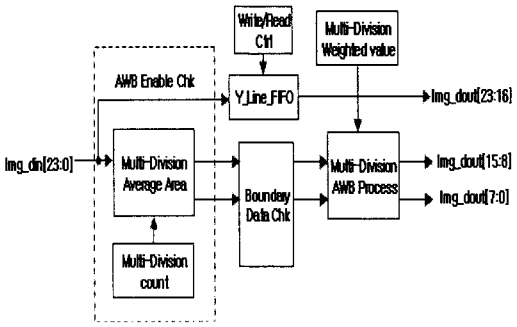

그림 4. 다중분할 영역 AWB 처리블록
Fig. 4 Multi-division Area AWB Processing Block

Ⅳ. 제안한 알고리즘 설계

4.1 인터폴레이션 블록

블록 그림 5에서는 2x2라인 인접한 픽셀에 대한 보간을 수행하기 위해 이전 라인 데이터와 현재 라인 데이터를 사용한다[10].

이전 데이터에 대한 버퍼링을 고려해서 라인버퍼 1개와 필요한 4개의 레지스터를 두고 각 라인별 처리를 수행한다. 행렬 라인에 대한 처리는 짝수행(짝수, 홀수열)과 홀수행(짝수, 홀수열)으로 구분하여 이를 위해 행 카운트와 열 카운트를 사용한다.

그리고 각 행렬 연산을 수행 한다. 그림 5에서는 2x2 라인 인터폴레이션 수행을 위한 세부 연산 블록도이고 연산에 필요한 라인버퍼 1개, 4개의 레지스터, 행렬 카운트 MUX, 레지스터 선택 MUX, 나눗셈 수행을 위한 오른쪽 쉬프트, 덧셈기 등으로 구성되어있다[5].

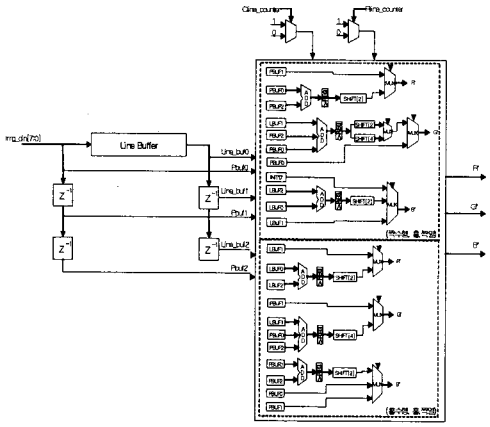

그림 5. 2x2 Line 인터폴레이션 블록
Fig. 5 2x2 Line Interpolation Block

4.2 이미지 스케일 블록

필터 윈도우 계수 6개의 좌표 값은 라인 단위 스캔 스트림 데이터 입력으로 이동하면서 6개의 좌표 값을 구한다.

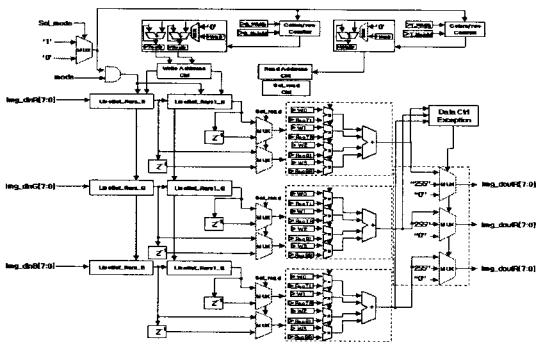

그림 6. 이미지 스케일 처리 블록
Fig. 6 Image Scale Processing Block

그림 6에서 원본 이미지에서 최대 0~4개의 픽셀을 사용하고 분할영역에 대한 영역별 가중치를 계산하여 적용한다.

또한 이중 포트 라인버퍼 RAM을 사용하여 쓰고/읽기의 액세스를 효과적으로 할 수 있게 하였다. 데이터 처리는 두 개의 라인을 사용하여 0~4개의 원본 이미지 픽셀 값과 가중치를 계산하여 새로운 이미지 스케일 데이터를 처리한다. 또한 연속 스트림 데이터를 처리 하기위해 핑퐁 RAM 액세스 방법을 사용하였다. 데이터와 가중치 계산을 위해 곱셈기 4개와 가산기 3개를 사용하여 출력 결과를 얻도록 설계하였다.

4.3 AWB 블록

그림 7은 각 AWB_en 신호가 활성화되면 영역별로 분할 한 후 평균값을 구한다. 처음 센서를 통하여 입력된 데이터는 일련의 ISP 처리 과정을 거친 후 AWB 수행에서는 YCbCr로 변환된 데이터를 사용하여 처리한다. 분할 영역의 평균값은 덧셈기, 나눗셈 연산기를 사용하여 처리한다. 또한 영역별 가중치를 할당한다. 다음 Cb, Cr의 픽셀 값이 임계치 안에 들어올 수 있도록 이전에 계산한 영역별 가중치와 평균을 이용한다. 히스토그램 방법은 Cb, Cr의 목표치 값으로 이동할 수 있게 처리한다. 히스토그램 방법에 사용하는 연산기는 비교적 간단한 곱셈기, 나눗셈 기를 이용하여 처리한다[12].

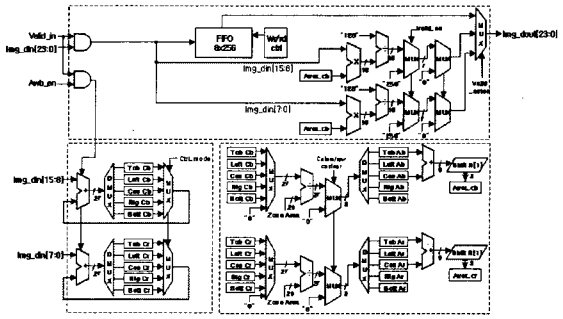

그림 7. AWB 처리 블록
Fig. 7 AWB Processing Block

V. 하드웨어 성능평가

5.1 2x2라인 인터폴레이션 성능

표 1은 회로의 합성 결과의 정보를 표로 나타내었다. 전체 게이트 사용량에서 제안한 IP와 3x3라인의 IP의 사용량을 비교하였다. 게이트 사용량은 왼쪽에서부터 플립-플롭의 사용량 2x2라인 처리의 1.5배 이상 연산량이 요소 되었고, 나머지 슬라이스 점유율, 전체 게이트 사용량에서도 3x3라인은 2x2라인 처리 연산량보다 1.5배 이상의 게이트 수를 차지하는 것을 알 수 있다.

표 2는 회로의 합성 후 최적화된 게이트 단위의 합성 결과 정보를 표로 나타내었다. 인터폴레이션 블록의 회로 합성한 결과를 입력도착 시간, 출력 요구시간, 조합회로 지연시간을 비교한 정보를 표로 나타내었다.

표 1. 3x3라인과 2x2라인의 게이트 수 비교
Table 1. Comparison for Gate Counts between 3x3 and 2x2 line

구분 / 방법	Logic Utilzation (Flip Flops)	Logic Distribution (Slices 점유율)	Total Equivalent (Gate count)
3x3라인	144	317	70,942
본 논문의 2x2라인	86	175	35,721

표 2. 3x3라인과 2x2라인의 합성 결과
Table 2. Synthesized Result for 3x3 and 2x2 Line

구분 / 방법	입력도착 시간 (Input arrival time)	출력요구 시간 (Output required time)	조합회로 지연 (combinational path delay)
3x3라인	14.401ns	39.585ns	32.481ns
본 논문의 2x2라인	13.535ns	31.488ns	No Path found

5.2 이미지 스케일 성능 평가

다음 표 3 은 이미지 스케일의 각 처리 알고리즘별 필요한 연산 컴포넌트 수를 비교한 결과를 보여주고 있다.

가장 좋은 영상의 화질을 보이는 Bicubic은 메모리 액세스에서 최대 16번 읽어 연산을 수행하여 하드웨어 구현 시 가장 많은 연산량을 필요로 한다. 다른 세 가지 알고리즘 처리와 비교하였을 때 영역 가중치, 가중치 합 계산에 필요한 가산기, 곱셈기 수가 가장 많이 소요되었다. 본 논문은 칩의 저전력, 저비용을 고려하여 연산량이 비슷한 제안한 알고리즘을 사용하여 양선형의 처리에 필요한 연산 컴포넌트수와 큰 차이를 가지지 않으면서 영상의 화질의 개선할 수 있는 하드웨어를 설계하였다[6][15].

표 3. 각 알고리즘별 컴포넌트 수
Table 3. The number of components for each algorithm

알고리즘 / 구분	Nearest Neighbor	Bilinear	본 논문의 알고리즘	Bicubic
픽 셀 이동좌표	2Add	2Add	6Add	2Add
가중치 요소	0Mult 0Add	3Mult 2Add	6Mult 2Add	16Mult 36Add
가중치의 합 계산	0Mult 0Add	4Mult 3Add	4Mult 3Add	16Mult 15Add
메모리 액세스	1Read 1Wirte	4Read 1Write	4Read 1Write	16Read 1Write

5.3 이미지 스케일 데이터 분석

다음은 영상의 이미지를 부분 확대하여 영상의 일그러짐 정도와 영상의 부드러운 이미지의 차이를 비교하였다. 이미지는 Lena 표준영상을 사용하였다.

그림 8의 (a)는 Lena 원본영상 256x256의 크기에서 Lena의 특징적인 눈을 확대하여 다양한 스케일 알고리즘으로 처리한 데이터들을 비교하였다. 그림 (b)는 가장 이웃한 화소 픽셀 알고리즘(Nearest Interpolation)을 적용하였다. 이웃한 픽셀 값을 바로 복사하여 사용하기 때문에 확대한 이미지에서 경계선 처리 문제에 대한 일그러짐 현상이 보였고 엣지는 전체적으로 잘 나타났다. 그림 (c)는 3차 회선 보간법(Bicupic Interpolation)을 적용하여 비교 대상 알고리즘에서 가장 좋은 영상의 화질과 엣지를 나타내었다. 그림(d)에서는 양선형 보간법(Bilnear Interpolation)을 사용하였다. 양선형 보간법에서는 영상을 확대하여 보았을 때 부드러운 이미지 영상으로 나타나지만 영상의 엣지(Edge) 검출한 이미지에 대해서는 엣지 처리가 잘 나타나지 않았다.

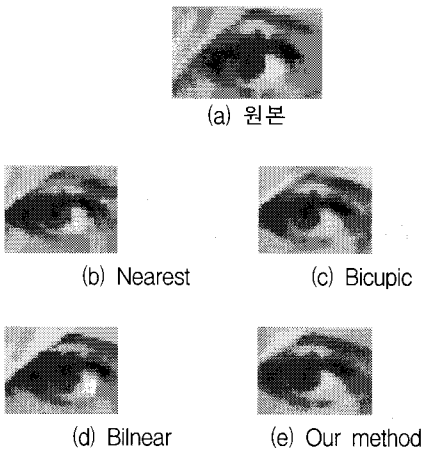

그림 8. 이미지 스케일 업과 엣지 검출 비교
Fig. 8 Comparison of Image Scale-up and Edge Detection

그림(e)는 제안한 알고리즘을 사용하였을 경우이다. 양선형 보간법의 부드러운 이미지 영상과 가장 인접한 픽셀 보간법의 엣지 특징을 잘 나타내어져 두 가지의 장점을 가진 알고리즘으로 양선형과 비교하여 연산량은 비슷하지만 영상의 이미지는 더 좋은 화질의 결과를 보였다.

5.4 제안한 ISP 기능 비교

다음 표 4는 기존 ISP 프로세서와 제안한 ISP 프로세서의 각 기능블록의 성능을 비교하여 표로 보여주고 있다.

표4. ISP의 성능 비교표
Table 4. Performance Comparison for ISP

처리 / 기능 블록	기존 프로세서	제안한 프로세서	비 고
감마교정	선형	선형	
컬 러 인터폴레이션	3x3라인	2x2라인	제안한 블록은 스케일을 고려하였음
이미지 스케일	VGA, QVGA, QQVGA	VGA, QVGA, QQVGA, CIF, QCIF	MPEG4, H.264, 모바일에 사용한 이미지 크기 지원
이미지 이펙트	Edge강조, 세피아, 반전	쿨, 바이올릿, 보사시, 반전, 세피아, 엠보싱	
AWB	라인단위의 처리 평균값, 최대값	히스토 그램 방법	
프레임 처리	30프레임(초당)	30프레임이상(초당)	

인터폴레이션에서는 이미지 스케일을 고려하면서 2x2라인을 사용하여 두 가지 동시에 처리하면서 칩의 면적을 줄이는 효과적인 설계를 진행하였다. 또한 이미지 스케일 처리 할 때 이미지 스케일은 추가로 CIF, QCIF를 지원하여 모바일 응용 스케일 출력 포맷등 다양한 스케일 처리를 지원하였다. 마지막 AWB 처리는 기존의 AWB의 처리에서 프레임 단위의 계수를 구하는 연산 소요 시간을 줄여 하드웨어 설계시 연산량을 줄이는 효과를 적용하여 설계 할 수 있다[11].

VI. PCI 인터페이스를 이용한 데이터 검증

6.1 PCI인터페이스 사용한 검증

다음은 PCI 인터페이스를 이용한 기능블록의 데이터를 FPGA 칩에 다운로드한 후 사용자 어플리케이션으로 각 기능을 동작 수행하여 데이터 검증을 진행한다. 이전 검증에서는 카메라 센서의 ISP 프로세서를 설계하여 각 기능 블록의 검증을 위해 윈도우 환경에서 Xilinx 7.1i툴을 이용하여 VHDL 을 사용한 알고리즘을 기술하고 이를 Model_SIM 6.0을 이용하여 각 기능 불록의 데이터 검증을 진행하였다. 최종적으로 데이터 검증을 진행하기 위해 PC환경과 타켓 XCV-1000e 라이브러리 칩을 사용한 FPGA 보드 상에서 ISP기능 수행 TOP모듈의 최종 (.bit) 파일을 생성하여 이를 FPGA 타켓 칩에 다운로드하고 PC환경에서 소프트웨어를 통한 데이터를 전송하고 이를 FPGA에 전송하여 데이터를 ISP프로세서에서 처리한 후 전송 받은 데이터를 PC에서 결과를 보여주는 보드 레벨의 검증을 수행하였다[15]. 다음 그림 9는 PC와 타켓 보드와의 데이터 전송 중재를 위한 인터페이스 데이터 블록을 보여주고 있다.

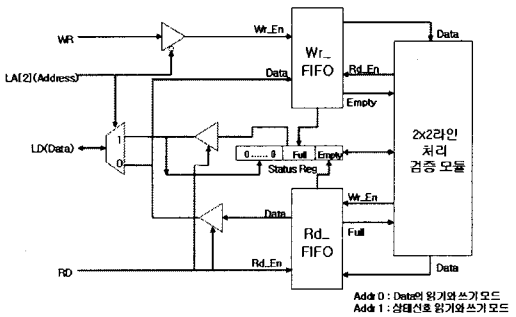

그림 9. PCI9054와 ISP간의 인터페이스 블록
Fig. 9 Interface Block between PCI9054 and ISP

다음 그림 10은 보드 레벨의 검증을 위한 테스트 환경 보여주고 있다.

6.2 PSD(Program State Dword) 정보

다음 11은 PCI9054 Local 신호의 32비트 입력 데이터에 대한 프로그램 상태 워드에 대한 설명이다. 최상의 바이트[31:24]는 컨트롤 비트(Control bit)의 16개에 대한 상태 디코더를 설정하기 위해 제공해주며, 나머지 4비트 예약어(Reserve)로 사용한다. 최하위 16비트는 정보는 ISP 처리에서 필요한 데이터를 입력 베이어 데이터 8비트와 AWB의 처리를 위한 평균 Cb, Cr값을 소프트웨어로 처리한 값을 입력받아서 ISP의 AWB의 처리 속도를 높이기 위해 다음과 같이 평균값을 PSD정보에 설정하여 사용하였다.

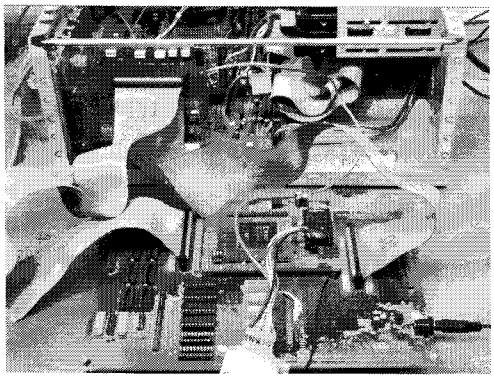

그림 10. PCI 9054와 XCV-1000e 타겟보드 구성 환경
Fig. 10 Environment of Target Board Configuration for PCI9054 AND XCV-1000e

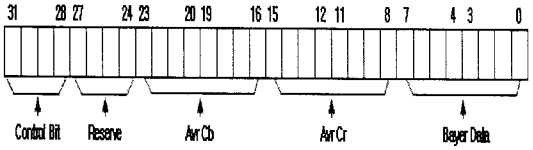

그림 11. PCI인터페이스의 PSD 데이터 정보
Fig. 11 PSD Information of PCI Interrface

6.3 결과 이미지

그림 12, 13에서 최종으로 비주얼 C++프로그램을 이용하여 결과 이미지를 보여주고 있다. 그림 12는 베이어 데이터를 입력 받아 2x2라인 방법을 이용하여 인터폴레이션을 수행한 결과 이미지를 PCI인터페이스를 통해 설계한 ISP 모듈에서 처리하였다.

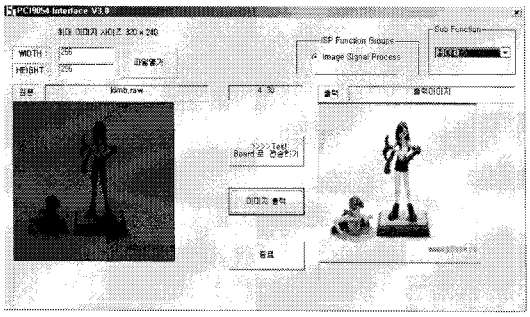

그림 12. 2x2라인을 이용한 인터폴레이션 결과이미지
Fig. 12 Result Image of Interpolation using 2x2 Line

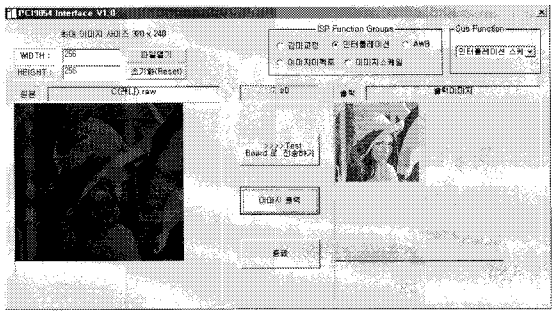

그림 13. 2x2라인 1/2이미지 스케일과 인터폴레이션 처리
Fig. 13 2x2 Line 1/2 Image Scale and Interpolation Processing

다음 13은 입력 데이터에 대한 원본 이미지에 대한 1/2 축소와 동시에 인터폴레이션을 수행한 결과 이미지이다.

VII. 결 론

ISP는 카메라 이미지 센서용 영상 입력 장치로부터 얻은 데이터를 입력 받아서 올바른 영상 데이터를 출력하기 위한 작업을 수행한다. ISP는 인터폴레이션, 이미지 스케일, AWB 기능 블록의 최적화 알고리즘을 도출하고 저전력, 저비용을 고려한 칩의 면적 최소화, 필요한 레지스터 수를 줄임으로써 최종 효과적인 하드웨어 설계를 진행하였다. 또한 하드웨어 성능 분석을 통해 인터폴레이션의 2x2라인 처리는 라인버퍼의 개수를 줄여 칩의 면적을 줄였으며, 이미지 스케일은 양선형 알고리즘의 연산량은 비슷하면서 화질의 개선의 성능을 보였다. 결과적으로 카메라 이미지 센서용 영상 신호 처리 적용 칩에서 제안한 알고리즘이 효과적으로 적용될 수 있을 것으로 사료된다.

참고문헌

[1] Randy Crane, Hewlett-Packard Company. "A Simplified Approach to Image Processing". Prentice Hall PTR

[2] Rafael C. Gonalez, Richard E. Woods. "Digital Image Processing". Addison Wesley.

[3] 엠텍비젼 - http://www.mtekvision.com/
[4] 코아로직 - http://www.corelogic.co.kr/
[5] Yun Ho Jung, Jae Seok Kim, Bong Soo Hur and Moon Gi Kang, "Design of Real-Time Image Enhancement Preprocessor for CMOS Image Sensor", IEEE Trans, Consumer Electronics, Vol 46: No 1, 2. 2000.
[6] 조태경, 홍재인 AMBA기반의 LCD컨트롤러 설계 한국콘텐츠학회 논문지 '04 Vol. 4 No 4.
[7] S. Ramachandran S.Srinivasan "Design and FPGA Implementation of a Video Scaler with on-chip reduced memory utilization". DSD'03. IEEE 2003.
[8] June-Sok Lee You-Young Jung, Byung-Soo Kim, Sung-Jea Ko, "An Advanced Video Camera System with Robust AF, AE, and AWB CONTROL", IEEE, Trans. vol, 47. No 3, 8. 2001.
[9] 정영식, 최영철, 장영조, "ISP를 위한 효율적인 White Balance 알고리즘", 한국기술교육대학교.
[10] 이동훈, 손승일, "CMOS 이미지 센서용 효과적인 인터폴레이션 구현", 해양 정보통신학회 춘계 학술 발표회 논문집, Vol 9 : No 1, (5. 27, 2005), pp353-357
[11] 정형돈, 이동훈, 손승일, "이미지 센서용 효과적인 Effect 구현", 해양 정보통신학회 춘계 학술 발표회 논문집, Vol 9 : No 1, (5. 27, 2005), pp1017-1021.
[12] 이현정, 이동훈, 손승일"CMOS 이미지 센서용 감마 교정 회로 설계", 해양 정보통신학회 춘계 학술 발표회 논문집, Vol 9 : No 1, (5. 27, 2005), pp1008-1012.
[13] 이동훈, 손승일, "CMOS 이미지 센서용 AE의 설계", 한국 통신학회 춘계 학술 발표회 논문집, Vol 31, (6. 30, 2005), pp5C-119.
[14] 정형돈, 이동훈, 손승일, "CMOS 이미지 센서용 효과적인 White Balance 구현", 한국 해양 정보통신학회 추계 학술 발표회 논문집, Vol 9 : No 2, (10. 28, 2005), pp607-610.
[15] 이동훈, 손승일, "CMOS 이미지 센서용 효과적인 이미지 스케일 구현", 한국 해양 정보통신학회 추계 학술 발표회 논문집 Vol 9 : No 2, (10. 28, 2005), pp307-310.
[16] PLX - http://www.plxtech.com/

저자소개

손 승 일(Seung-Il Sonh)

1989년 연세대학교 전자공학과 (학사)
1991년 연세대학교 대학원 전자공학과 (석사)
1998년 연세대학교 대학원 전자공학과(박사)
2002년~현재 한신대학교 정보통신학과 부교수
※관심분야 : ATM 통신 및 보안, ASIC 설계, 영상신호처리칩

이 동 훈(Dong-Hoon Lee)

2006년 한신대학교 컴퓨터정보학과 (석사)
2006년~현재 C&S 테크놀로지 반도체 연구소 연구원
※관심분야 : ASIC 설계, 영상신호처리칩