

# 강화된 스케일링 정밀도와 자체 파라미터 계산 기능을 가진 실시간 이미지 크기 조절기

김기현\*·류광기\*

\*한밭대학교 정보통신전문대학원 정보통신공학과

## A Real time Image Resizer with Enhanced Scaling Precision and Self Parameter Calculation

Kihyun Kim\* · Kwangki Ryoo\*

\*Graduate School of Information and Communication, Hanbat National University

E-mail : kkkh89@hotmail.com, kkryoo@hanbat.ac.kr

### 요 약

이미지 크기 조절기는 영상을 화면에 표시하기 위한 디스플레이 장치나 원하는 크기의 영상을 처리할 필요가 있는 영상처리블록 등에서 사용하게 되는 IP이다. 제안한 이미지 크기 조절기는 기존의 이미지 크기 조절기에서 사용하는 프레임 메모리가 아닌 라인 메모리를 사용하고 기본 유닛에 대한 기존 연산 과정에  $2^m$ 배를 확대해주고 다시 시프트하여  $2^m$ 배를 축소시키는 방법을 추가함으로써, 하드웨어 구현이 용이하고 적은 자원을 소모하며 뛰어난 정밀도를 가진 이미지 크기 조절기를 보인다. 또한 기존의 이미지 크기 조절기들과 다르게 크기 조절을 위한 내부 파라미터를 자동으로 자체 계산하는 직렬 계산기를 내장하여 IP의 효율성을 증가시켰다. 본 논문에서는 이미지 크기 조절기를 Verilog HDL로 설계하였고 검증용 하기 위하여 이미지 센서와 LCD를 이용하는 어플리케이션 IC에 적용되어 Xilinx Vertex-4 XC4LX80 FPGA로 기능 및 타이밍 검증을 마쳤다 또한 TSMC 0.18um 공정을 이용하여 ASIC으로도 구현하였다.

### ABSTRACT

An image scaler is a IP used in a image processing block of display devices to adjust image size. Proposed image scaler adopts line memories instead of a conventional method using a frame memory. This method reduced hardware resources and enhanced data precision by using shift operations that number is multiplied by  $2^m$  and divided again at final stage for scaling. Also image scaler increased efficiency of IP by using serial divider to calculate parameters by itself. Parameters used in image scaling is automatically produced by it. Suggested methods are designed by Verilog HDL and implemented with Xilinx Vertex-4 XC4LX80 and ASIC using TSMC 0.18um process.

### 키워드

Scaler, Bilinear, Line memory, Small chip size, Serial divider

### 1. 서 론

최근 스마트폰의 보급이 빠르게 진행되고 Apple 사의 iPad와 같은 Tablet PC의 발전으로 영상 분야에서 큰 해상도의 영상을 화질 저하 현상 없이 입력 영상 보다 작은 크기의 디스플레이

장치에 표현하는 것이 중점화가 되고 있다 여기서 중요한 부분 중의 하나는 입력된 영상을 얼마만큼 정밀하게 조절하는 가에 있다[1].

본 논문에서는 프레임 메모리가 아닌 라인 메모리를 사용하고 기본 유닛에 대한 기존 연산 과정에  $2^m$ 배를 확대해주고 다시 시프트 하여  $2^m$ 배

를 축소시키는 방법을 추가함으로써 하드웨어 구현이 용이하고 적은 자원을 소모하며 뛰어난 정밀도를 가진 이미지 크기 조절기를 보인다 또한 기존의 이미지 크기 조절기들과 다르게 크기 조절을 위한 내부 파라미터를 자동으로 자체 계산하는 직렬 제산기를 내장하여 IP의 효율성을 증가시켰다.

## II. 본 론

### A. 이미지 크기 조절기의 구조

구현한 이미지 크기 조절기는 YCbCr 422 형식의 영상을 실시간으로 두 개의 가로 픽셀과 한 개의 세로 픽셀을 처리하도록 설계하였다

그림 1은 구현한 이미지 크기 조절기의 구조를 나타낸다. 이미지 크기 조절기의 동작 원리는 이미지 센서로부터 영상이 입력되면 파라미터 조절기에서 외부로부터 입력 받은 파라미터를 내부 파라미터에 맞게 만들어 Resize 모듈에 보낸다.

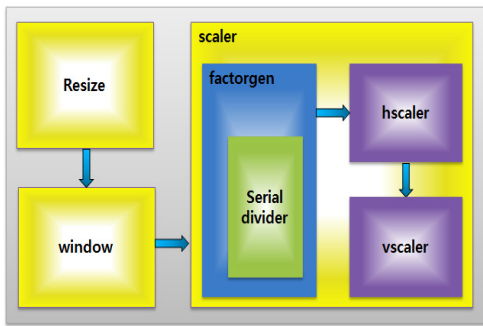


그림 1. 이미지 크기 조절기의 구조

입력 받은 내부 파라미터 값에 맞게 Resize 모듈에서 원하는 크기에 가장 근접한  $1/2^n$ 의 크기로 화면을 줄인다. 그 다음 Resize 모듈에서 크기가 줄어든 화면 데이터와 파라미터 조절에서 받은 내부 파라미터를 입력 받아 Scaler 모듈에서 최종 목표 크기로 세부 조절하는 방식이다

### B. Bilinear interpolation

Interpolation이란 이미지의 크기를 조절하거나 이미지의 형태를 변형할 때 새롭게 생기는 픽셀들을 설정하는 방법이다[2].

Interpolation의 종류로는 Nearest neighbor, Bilinear, Bicubic 등의 방식들이 있으며 본 논문에서는 그림 2와 같은 Bilinear 알고리즘을 적용하여 좌표에서 주위 픽셀들의 값을 사용하여 가까운 픽셀에 가중치를 주어 그 픽셀을 새로운 픽셀로 만들어 사용했다.

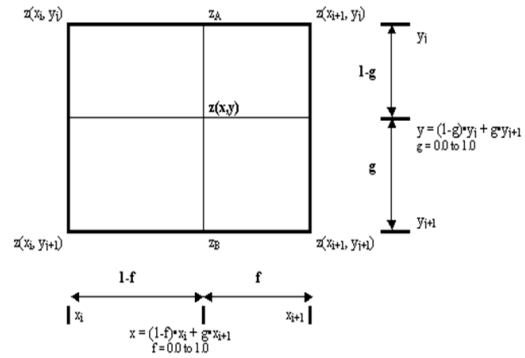


그림 2. Bilinear interpolation

## III. 강화된 스케일링

### A. 일반적인 알고리즘

일반적으로 기하학적인 영상 처리에 있어서 기본 유닛인 픽셀 단위(a)을 계산하는 알고리즘은 픽셀 단위(a)을 실제 픽셀/목표 픽셀로 연산하며, 연산 후에는 각 픽셀이 입력됨에 따라 픽셀 유닛에 해당하는 영역에 이르게 되면 Bilinear interpolation 알고리즘 방법에 의해 주변의 실제 픽셀들로부터 목표 픽셀의 값을 계산하는 방식으로 이루어진다.

식 1은 기존 이미지의 모든 픽셀 수를 구할 때 사용하는 연산이고, 식 2는 픽셀 유닛을 구할 시에 필요한 변수인 k를 구하는 연산으로써, 앞에서 구한 k와 Total을 이용하여 픽셀 유닛을 구한다

$$\text{Total} = \text{original}_{\text{width}} \cdot \text{resolution} \quad (1)$$

$$k = 2^n / \text{target}_{\text{width}} \quad (2)$$

$$\text{pixel unit}(a) = (\text{Total} \cdot k) / 2^n \quad (3)$$

식 4는 기존 이미지와 목표 이미지간의 픽셀 격차를 구하는 연산이고, 식 5는 픽셀들의 격차로 생긴 손실의 정도를 구하는 연산이다

$$\text{diff} = \text{Total} - (\text{target}_{\text{width}} \cdot \text{pixel unit}) \quad (4)$$

$$\text{loss}_{\text{pixel}} = \text{diff} / \text{resolution} \quad (5)$$

표 1은 식 1~5에서 나타낸 연산 식들을 이용하여 기존 이미지의 크기가 2048이고 목표 이미지의 크기가 1311이며 resolution이 16,  $2^n$ 이  $2^{15}$ 일 때의 픽셀 유닛 값과 diff, loss 값을 구한 표이다.

loss pixel이 82가 나온 것을 토대로 일반적인 알고리즘은 기존 이미지와 목표 이미지의 픽셀간 격차가 크므로 인하여 손실되는 픽셀이 생기는 것을 알 수 있다.

표 1. 일반적인 픽셀 유닛의 알고리즘

Variable	Expression of variable	Value
Total	2047 x 16	32752
k	32768 / 1310	25
pixel unit	32752 x 25 / 16	24
diff	32752 - (1310 x 24)	1312
loss pixel	1312 / 16	82

B. 제안하는 알고리즘

설계한 이미지 크기 조절기에서는 제안기가 가진 단점 중 하나인 면적을 많이 차지한다는 점과 기존 픽셀 유닛 연산에서의 기존 이미지와 목표 이미지 간의 픽셀 간 격차가 크다는 점을 고려하였다. 일반적인 변수 k를 구하는 연산 과정에 2<sup>m</sup> 배를 확대시키고 다시 2<sup>m</sup>의 값을 시프트 시켜 축소시키는 과정을 추가하여 픽셀 유닛을 구한다

식 6과 식 7은 제안하는 알고리즘에서의 k를 구하는 연산식과 픽셀 유닛을 구하는 연산식이다

$$k = (2^n \cdot 2^m) / \text{target}_{\text{width}} \quad (6)$$

$$\text{pixel unit}(a) = k / 2^m \quad (7)$$

이러한 방식으로 기존의 픽셀 유닛을 구하는 연산에서 생기는 기존 이미지와 목표 이미지 간의 픽셀 차에 의해 손실 픽셀들이 생성되는 현상을 없애고, 픽셀 단위로 이미지의 크기를 조절할 수 있게 한다.

표 2는 표 1에서와 마찬가지로 기존 이미지가 2048이고 목표 이미지가 1311이며, resolution이 16, 2<sup>n</sup>이 2<sup>15</sup>, 2<sup>m</sup>이 2<sup>2</sup> 일 때의 픽셀 유닛 값과 diff, loss 값을 제안하는 알고리즘으로 구한 표이다. 표 1의 손실 픽셀 값과 표 2의 손실 픽셀 값을 비교해 보면 제안한 알고리즘에서 손실 픽셀 값이 줄어든 것을 알 수 있다

표 2. 제안하는 픽셀 유닛의 알고리즘

Variable	Expression of variable	Value
Total	2047 x 16	32752
k	32768 x 4 / 1310	100
pixel unit	100 / 4	25
diff	32752 - (1310 x 25)	2
loss pixel	2 / 16	0

IV. 자체 파라미터 계산 기능

앞에서 기술한 바 있는 픽셀 유닛은 본 논문에서 구현한 이미지 크기 조절기의 기능을 구현하기 위한 중요한 파라미터로 실제 이를 위해서는

정수화한 계수로 그 값을 구해야한다 기존의 픽셀 유닛을 구하는 연산을 사용하지 않고 제안하는 픽셀 유닛 연산을 사용하여 기존 이미지와 목표 이미지 간의 픽셀 차로 인하여 생기는 오차를 줄였으나 이 과정에서도 오차가 발생하게 된다 이러한 오차를 줄이기 위해서 계산 과정에 사용하는 수치들의 자릿수를 원하는 정밀도가 나올 수 있도록 충분히 설정해야 한다.

일반적으로 쓰이는 제안기 IP들은 하드웨어의 크기가 너무 크다는 단점을 가진다. 게다가 상대적으로 낮은 사용 빈도의 파라미터 설정에 사용되기 때문에 이미지 조절기 IP들은 많은 경우들에 있어서 복잡한 파라미터를 외부 MCU에 제공하여 그 계산을 하도록 구현되어 있다.

본 논문에서는 자체 계산을 어렵게 하는 요인인 제안기의 하드웨어 크기에 비해 적은 면적으로 나누기 연산을 수행하는 직렬 제안기를 설계하여 사용함으로써 해결하였다. 표 3은 설계한 제안기를 TSMC 0.18um 공정에 적용시켜 Synopsys사의 Design Compiler의 Design ware의 제안기와 크기를 비교한 것이다.

표 3. Design ware의 제안기와 설계한 제안기와의 크기 비교

Resource	Design ware divider	Designed divider
Total gates	2568	606

그러나, 직렬 제안기는 연산 시간이 일반 제안기들에 비해 훨씬 길다는 단점을 가지고 있다 이를 피하기 위해 직렬 제안기의 동작이 이루어지는 시점을 그림 3에서 보이는 바와 같이 각 영상이 바뀌는 시기에 넓은 간격이 있다는 점을 고려하여, 화면의 시작을 알리는 동기 신호에 맞추어 줌으로써 직렬 제안기의 긴 동작 시간이 문제가 되지 않도록 하였다. 그와 동시에 IP의 파라미터 세팅이 이루어지는 최소 갱신 시간을 1 프레임 단위로 하여 실시간 처리가 가능하도록 하였다

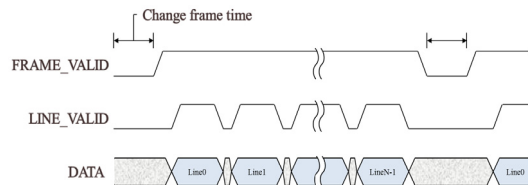


그림 3. 영상의 프레임이 바뀌는 시기

V. 결 론

설계한 IP는 Xilinx Vertex-4 XC4LX80 FPGA 디바이스를 사용하였고, 이미지센서와 LCD를 이

용하는 어플리케이션 IC에 사용하여 기능 및 타이밍 검증은 마쳤다. 그림 4는 구현한 이미지 크기 조절기와 이미지센서와 LCD를 이용하는 어플리케이션 IC에 대한 블록다이어그램이다

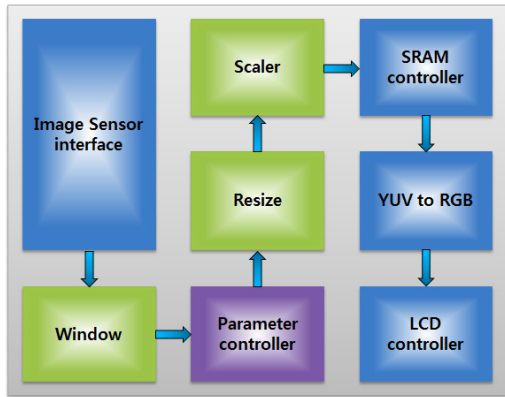


그림 4. 어플리케이션 IC의 블록다이어그램

표 4는 FPGA로 구현한 결과를 나타낸다 Slice 수는 1,285이고 LUT 수는 2,268이다.

표 4. ISP용 SoC의 FPGA 구현 결과

Logic Utilization	Used	Available	Utilization
No of Slices	1,285	35,840	3%
No of Slices Flip Flops	857	71,680	1%
No of Input LUTs	2,268	71,680	3%
No of Bonded IOBs	104	768	13%
No of BRAMs	2	200	1%

또한 TSMC 0.18um공정을 이용하여 ASIC으로도 구현하였다. 표 5는 설계한 이미지 크기 조절기를 ASIC으로 구현한 정보이다. 25MHz의 클럭을 사용하였고, 면적은 3x3mm<sup>2</sup>이다. 총 게이트는 41932를 사용하였다.

표 5. 제안하는 픽셀 유닛의 알고리즘

공정	TSMC 0.18 um CMOS
합성 클럭 주파수	25MHz
P&R 클럭 주파수	25MHz
면적	1.05x1.05mm <sup>2</sup> Core-region of 3x3mm <sup>2</sup>
동작 전압	1.8V Core, 3.3V I/O
기능	Scaler
회로타입	Digital

그림 5는 이미지 크기 조절기를 직접 설계하여 Virtuoso 툴에서 Astro툴로 만든 Layout을 본 것이다.

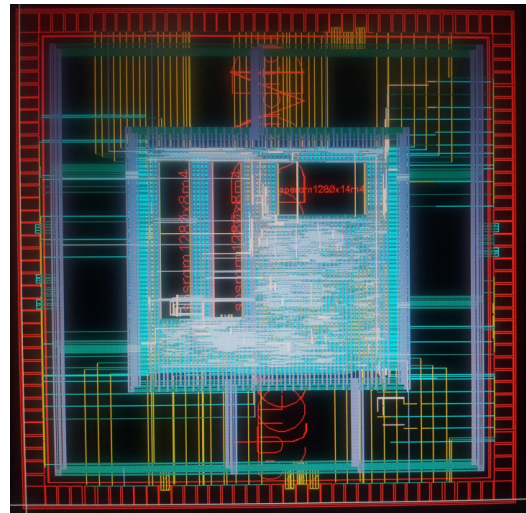


그림 5. 이미지 크기 조절기의 레이아웃

구현한 이미지 크기 조절기는 연속적인 실시간 터치스크린 화면 크기 조절과 같은 응용 분야에 유용하게 사용될 것이다.

### 감사의 글

본 논문은 교육과학부와 한국연구재단의 지역 혁신인력양성사업 및 지식경제부 출연금으로 수행한 ETRI SW-SoC융합R&BD센터와의 공동 연구의 결과입니다.

### 참고문헌

- [1] R. Gonzalez, R. Woods, Digital Image Processing, 2nd Edition, Pearson Prentice Hall, New Jersey, 2002.
- [2] S. Fahmy, C. Bouganis, P. Cheung, and W. Luk, "Realtime hardware acceleration of the trace transform", In Journal of Real-Time Image Processing, vol. 2, pp. 235-248, 2007.