

# High Performance Implementation of Image Scaling Processor in FPGA Using Bilinear Interpolation

S.Hariprasath<sup>1</sup>, M.Santhi<sup>2</sup>,

Department of Electronics and Communication Engineering  
<sup>1,2</sup>Saranathan College of Engineering,  
 Trichy, Tamil Nadu, India

## Abstract

Image scaling is an imperative task in most of the machine vision applications like pattern recognition, pattern understanding. The Region of Interest (ROI) in the image must be resized before features are extracted. At different resolutions the region to be processed is described in order to evaluate the content present in the image under interest. In this paper, memory efficient and delay efficient image scaling algorithm is implemented. The proposed method consists of four modules namely sharpening filter, clamp filter, edge detector and bilinear interpolator. The bilinear interpolation algorithm is chosen as scale up or scale down algorithm due to its simplicity, high quality output and less operational cycles consumption in hardware compared other well-known methods like cubic interpolation, nearest neighbourhood interpolation. Using Verilog HDL, the RTL description of the algorithm is developed and the functional output is verified using Modelsim10.4a. Using Xilinx 9.2i IDE, the RTL implementation is verified. The target FPGA chosen is Spartan3 XC3S400PG208 board. For the input images of size 64\*64, 8 bit gray scale image, the operating frequency of the design is found to be equal to 215.64Mhz. The area occupied for the proposed algorithm is less compared exiting implementations.

**Key words** - image processing, scaling, FPGA, Verilog, PSNR.

## I. INTRODUCTION

Image scaling is a significant process for resizing an image. Digital display devices such as liquid crystal display (LCD) or plasma display panel (PDP), digital camera, surveillance camera, digital video recorders, digital photo frame, mobile phone, tablet PC become increasingly popular [1,2] nowadays. Hence the demand and significance of image scaling are very high.

There are various scaling methods which can be generally classified as polynomial based methods and non-polynomial based methods [1].

Hence in this work bilinear interpolation algorithm is chosen for implementation.

Pei-Yin Chen, Chi-Yuan Lien [3] have proposed a method for implementing edge oriented image scaling processor. In their implementation, the architecture designed has seven stages. A processing rate of about 200 MHz is achieved by using TSMC 0.18- $\mu$ m technology.

Francisco Cardells-Tormo and Jordi Arnabat-Benedicto [4] have implemented a digital friendly architecture of two dimensional separable convolutional algorithm for image scaling purpose.

In the architecture proposed by Lun Chen [5], low cost implementation is taken as design constraint. H. Kim, Y. Cha, and S. Kim [6] have designed curvature based interpolation method zooming digital images. J.W.Han, J.H.Ki [7] have proposed a novel bilateral filter for image zooming.

For hardware implementation polynomial based methods are generally preferred over Non-polynomial based methods due to less memory requirement and simple mathematical modeling. Some of the polynomial based methods are nearest neighborhood method, bilinear interpolation, bi-cubic interpolation. Among these algorithms [8, 9], the nearest neighborhood method is the simplest and cost effective in terms of logic cells consumption in hardware implementation. But the scaled output images may consist of blocking and aliasing artifacts.

Shih-Lun Chen [10] have proposed an adaptive scaling procedure to implement zooming of image pixels.

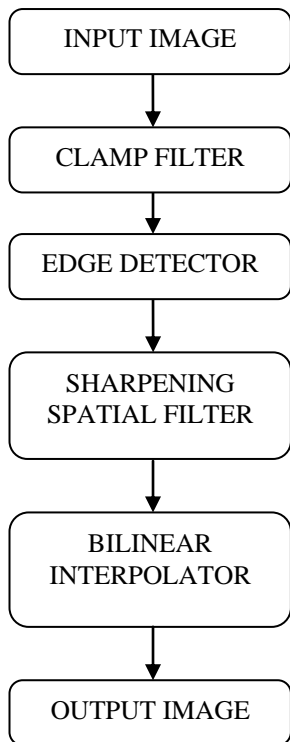
In Bilinear interpolation, the target pixel is obtained by performing linear interpolation in both horizontal and vertical directions. By interpolating the image in this manner causes the edges of scaled image to become blurred and aliased. In order to smoothen these effects, clamp filters and sharpening spatial filters as pre filters along with an edge detector to enhance the edges are used. For real time applications, low complex image processing algorithms are needed for FPGA implementation. This algorithm demands low computing resources and less memory storage per pixel. The choice of clamp parameter C and

sharpening value S are limited by the above requirements.

The rest of the paper is organized as follows. The proposed system model is explained in Section II. The parameters chosen for achieving low area implementation are explained in Section III and simulation and synthesis results are explained in Section IV. Finally the conclusion is presented in Section V.

**II. METHODOLOGY**

The flow diagram of our proposed scaling algorithm is shown in figure 1.



**Figure.1 Design Flow diagram**

The structure consists of a sharpening spatial filter, a clamp filter and bilinear interpolator. The clamp filter is a generalized weighted smoothing filter which is used for smoothening uncontrollable condition known as an aliasing effect. This is followed by a sharpening spatial filter (high pass filter) to reduce the blurring artifacts.

The clamp and sharpening spatial filters are used as pre-filters. The clamp is used for smoothening and to reduce the aliasing artifacts. The clamp filter is implemented in hardware by specifying it as a mask. The typical mask of clamp filter is given in equation (1).

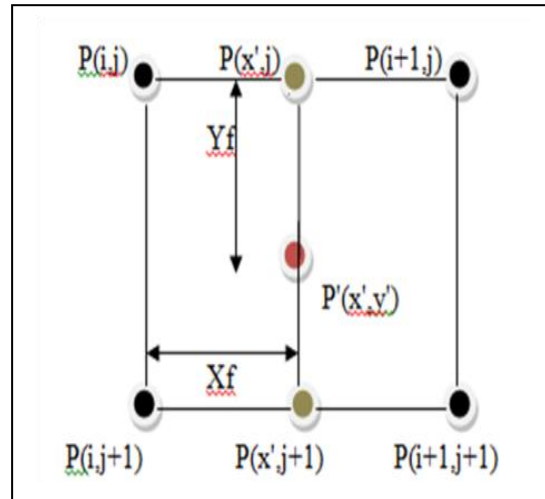
$$K_c = \begin{pmatrix} 1 & 1 & 1 \\ 1 & c & 1 \\ 1 & 1 & 1 \end{pmatrix} \dots\dots\dots (1)$$

The sharpening spatial filter is used to enhance the edges, remove background noise and to reduce the

blurring effects. The mask of the sharpening filter in spatial domain is given in equation (2).

$$K_c = \begin{pmatrix} -1 & -1 & -1 \\ -1 & s & -1 \\ -1 & -1 & -1 \end{pmatrix} \dots\dots\dots(2)$$

**The basic model of bilinear interpolation is shown in figure 2.**



**Figure.2 Basic bilinear interpolation scheme**

$$P(x', j) = (1 - X_f) \times P(i, j) + X_f \times P(i + 1, j)$$

$$P(x', j + 1) = (1 - X_f) \times P(i, j + 1) + X_f \times P(i + 1, j + 1)$$

$$P(x', y') = [P(x', j)] \times (1 - Y_f) + P(x', j + 1) \times Y_f \dots (3)$$

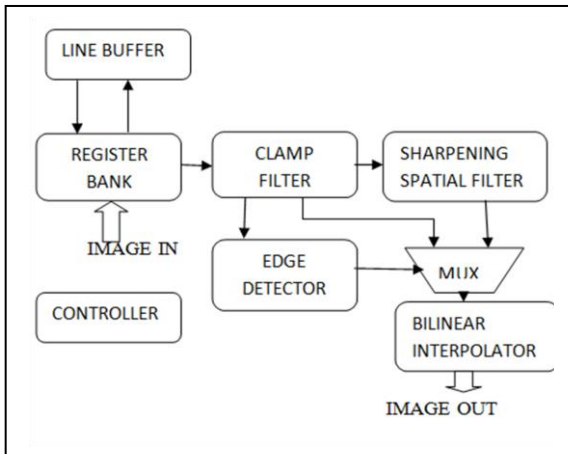
In equation (3),  $P(i, j)$  represents the source pixel,  $P(x', y')$  represents the scaled pixel,  $X_f$  and  $Y_f$  are the scaling factor along horizontal and vertical direction respectively. As shown in equations (3) the interpolation operation can be realized by means of Multipliers and a block of adder/subtractor modules. In this proposed work signed adder/subtractor and a pipelined MAC unit is used to implement the bilinear interpolation algorithm.

**III. IMPLEMENTATION DETAILS**

The scaling filter and sharpening filter are operating in the spatial domain. Hence the source pixel values are stored in a memory with hold time and setup time specifications. In this proposed scheme, a FIFO of size 64 by 8 is implemented as memory for storing and sending values to data path unit.

The filter coefficients are also stored in a 9 by 8 dual port memory designed using Verilog language. A Mask of 3 by 3 is stored in the dual port memory and upon receiving the enable signal from timing unit, the mask values are passed to data path unit. A single MAC unit is used here as processing element (PE). A group of 9 such MAC units form data path unit in this design.

The FPGA implementation is shown in figure 3.



There are some parameters that affect implemented processor performance. Two noticeable parameters that determine the performance of the algorithm are Clamp value (C) and Sharpening value (S) [2, 3, 4]. The blurring effect and clarity vary according to parameter changes. A tradeoff relationship exists between (S), (C) and the data representation chosen for hardware implementation. In this implementation an 8 bit representation is chosen.

#### IV. SIMULATION AND SYNTHESIS RESULTS

The proposed method is simulated using Modelsim10.3b simulator by applying test fixture file in Verilog HDL. The entire design of filters and bilinear interpolator is designed using Verilog HDL.. It is found that the memory unit operates with a maximum frequency of 415.536 MHz. The bilinear interpolator operates with a maximum delay of 21.877ns. The RTL diagram of memory unit and bilinear interpolator are shown in figure 4 and 5 respectively. Figure 6 shows the detailed architectural view of bilinear interpolator.

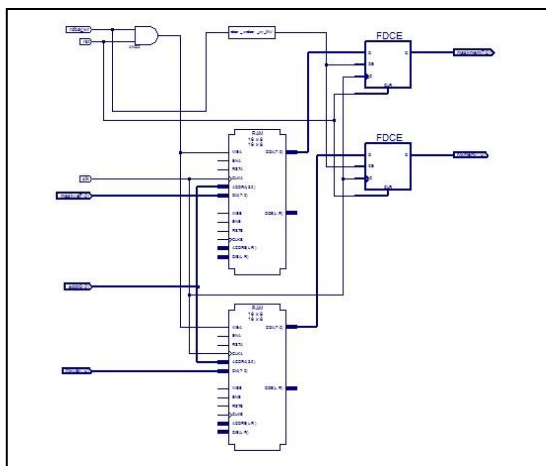


Figure 4. RTL diagram of Memory Unit

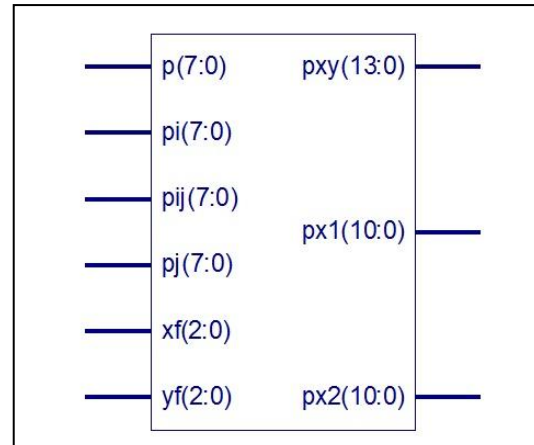


Figure 5. RTL view of bilinear interpolator

The bilinear interpolator is implemented using MAC unit. The signed multiplication is implemented for 8 bits since the input image considered is gray scale image. A total of 5 Adder/Sub tractors are inferred and 6 Multipliers are inferred for one pixel interpolation task. By pipelining of 2 stages and repeating the process for 64 clock cycles scaling of an input image of size 8 by 8 is achieved.

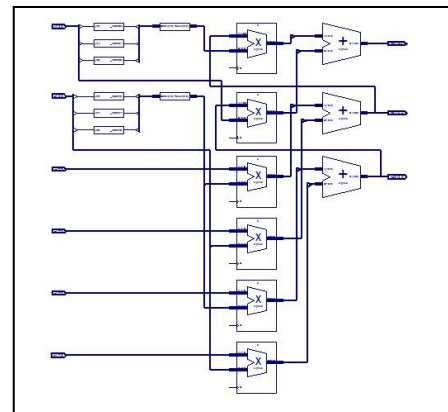


Figure6.Detailed architecture of bilinear interpolator

The simulation result of bilinear interpolator is given in figure 7. The operational speed achieved for the overall design is 215.64 MHz. By reducing the number of multiplications in the case of edge pixels the speed of operation can be further increased by 30 to 40 percentage.

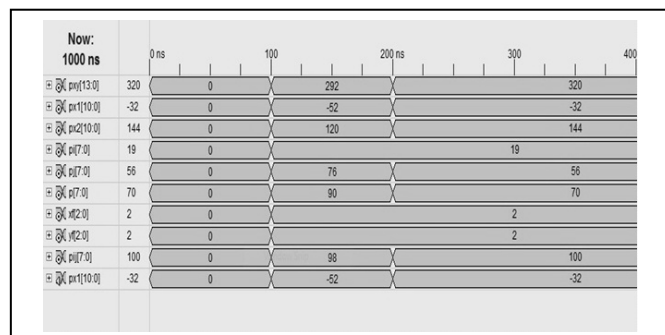


Figure 7.simulation result of bilinear interpolator

The following figure 8 shows the simulation result of memory unit.

Applications”, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 23, No. 9, September 2013

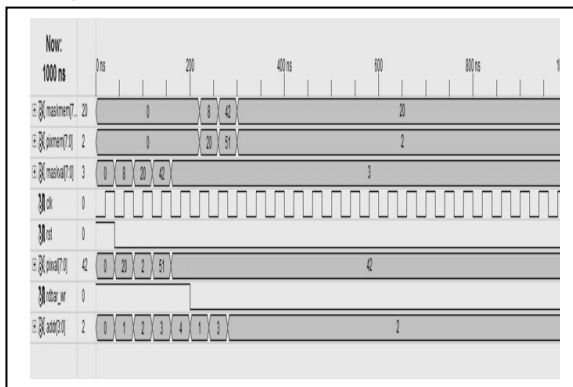


Figure 8. Simulation result of memory unit

## V. CONCLUSION

The present work has been successfully simulated and designed using Verilog HDL. The structure was designed using MAC unit and input values are stored in FIFO memory on Spartan 3 board with xcs3004s-pq208 FPGA with speed grade 5. The board is having 2 in-built block rams which could be used if additional memory structure is required. By incorporating more pipelined stages, the speed of operation can be increased.

## REFERENCES

- [1] Joannis Andreadis, Angelos Amanatiadis, “Digital Image Scaling”, IMTC 2005 - Instrumentation and Measurement Technology Conference, Ottawa, Canada 17-19 May 2005
- [2] Xiangjun Zhang and Xiaolin Wu, “Image Interpolation by Adaptive 2-D Autoregressive Modeling and Soft-Decision Estimation”, IEEE Transactions On Image Processing, Vol. 17, No. 6, June 2008
- [3] Pei-Yin Chen, Chi-Yuan Lien, and Chi-Pin Lu, “VLSI Implementation of an Edge-Oriented Image Scaling Processor”, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 17, No. 9, September 2009
- [4] Francisco Cardells-Tormo and Jordi Arnabat-Benedicto, “Flexible Hardware-Friendly Digital Architecture for 2-D Separable Convolution-Based Scaling”, IEEE Transactions On Circuits And Systems—II: Express Briefs, Vol. 53, No. 7, July 2006
- [5] Lun Chen, “VLSI Implementation of a Low-Cost High-Quality Image Scaling Processor”, IEEE Transactions On Circuits And Systems II Express Briefs, Vol. 60, No. 1, January 2013
- [6] H. Kim, Y. Cha, and S. Kim, “Curvature interpolation method for image zooming,” IEEE Trans. Image Process., vol. 20, no. 7, pp. 1895–1903, Jul. 2011.
- [7] J.W.Han, J.H.Kim, S.H.Cheon, J.O.Kim, and S.J.Ko, “A novel image interpolation method using the bilateral filter,” IEEE Trans. Consum. Electron., vol. 56, no. 1, pp. 175–181, Feb. 2010.
- [8] C. C. Lin, M. H. Sheu, H. K. Chiang, C. Liaw, and Z. C. Wu, “The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing,” in Proc. IEEE Int Conf. Circuits Syst., May 2008, pp. 480–483.
- [9] S. L. Chen, H. Y. Huang, and C. H. Luo, “A low-cost high-quality adaptive scalar for real-time multimedia applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 11, pp. 1600–1611, Nov. 2011.
- [10] Shih-Lun Chen, “VLSI Implementation of an Adaptive Edge-Enhanced Image Scalar for Real-Time Multimedia