Resource Efficient Image Scaling Processor Using VLSI Technology

[I]J.Jeba Priya, [II]M.Annalakshmi
[I]Final Year, M.E. – VLSI DESIGN, [II]Asso. Professor, ECE Dept.
[I,II]Sethu Institute of Technology, Kariapatti, India

Abstract

A low-cost, low-memory-requirement, high quality and high-performance VLSI architecture of the image scalingprocessor is *proposed in this project. The filter combining, hardware sharing, and reconfigurable techniques had been used to reduce hardware cost. The proposed image scaling algorithm consists of a sharpening spatial filter, a clamp filter, bilinear and a nearest neighborhood interpolation. To reduce the blurring and aliasing artifacts produced by the bilinear interpolation, the sharpening spatial and clamp filters are added as pre filters. To minimize the memory buffers and computing resources for the proposed image processor design, a T-model and inversed T-model convolution kernels are created for realizing the sharpening spatial and clamp filters. Compared with previous low-complexity techniques, this work reduces gate counts when compared with existing method and requires only a one-line-buffer memory. This proposed system is designed using verilog HDL, simulated using Modelsim Software and synthesized using Xilinx Project Navigator.*

Keywords

Bilinear interpolation, clamp filter, sharpening spatial filter, reconfigurable calculation unit (RCU).

I. Introduction

Image scaling has been widely applied in the fields of digital imaging devices such as digital cameras, digital video recorders, digital photo frame, high-definition television, mobile phone, tablet PC, etc. An obvious application of image scaling is to scale down the high-quality pictures or video frames to fit the minisize liquid crystal display panel of the mobile phone or tablet PC. As the graphic and video applications of mobile handset devices grow up, the demand and significance of image scaling are more and more outstanding. The image scaling algorithms can be separated into polynomial-based and non-polynomial-based methods. The simplest polynomial-based method is a nearest neighbor algorithm. It has the benefit of low complexity, but the scaled images are full of blocking and aliasing artifacts. The most widely used scaling method is the bilinear interpolation algorithm [1], by which the target pixel can be obtained by using the linear interpolation model in both of the horizontal and vertical directions. Another popular polynomial-based method is the bicubic interpolation algorithm [15], which uses an extended cubic model to acquire the target pixel by a 2-D regular grid. In recent years, many high-quality non-polynomial-based methods [2]–[4] have been proposed. These novel methods greatly improve image quality by some efficient techniques, such as curvature interpolation [2], bilateral filter [3], and autoregressive model [4]. The methods mentioned earlier efficiently enhance the image quality as well as reduce the artifacts of the blocking, aliasing, and blurring effects. However, these high-quality image scaling algorithms have the characteristics of high complexity and high memory requirement, which is not easy to be realized by VLSI technique. Thus, for real-time applications, low-complexity image processing algorithms are necessary for VLSI implementation [5–9]. To achieve the demand of real-time image scaling applications, some previous studies [10–15] have proposed low complexity methods for VLSI implementation. Kim et al. proposed the area-pixel model Winscale [10], and Lin et al.realized an efficient VLSI design [11]. Chen et al. [12] also proposed an area-pixel-based scalar design advanced by an edge-oriented technique. Lin et al. [13, 14] presented a low cost VLSI scalar design based on the bicubic scaling algorithm. In our previous work [15], an adaptive real-time, low-cost, and high-quality image scalar was proposed. It successfully improves the image quality by adding sharpening spatial and clamp filters as pre filters [5] with an adaptive technique based on the bilinear interpolation algorithm. Although the hardware cost and memory requirement had been efficiently reduced, the demand of memory still costs four line buffers. Hence, a low-cost and low-memory-requirement image scalar design is proposed in this brief.

II. Proposed Method

A. Block Diagram

The block diagram of the proposed scaling algorithm is shown Fig 1.It consists of a sharpening spatial filter, a clamp filter, and a bilinear interpolation. The sharpening spatial and clamp filters [6] serve as prefilters [5] to reduce blurring and aliasing artifacts produced by the bilinear interpolation.

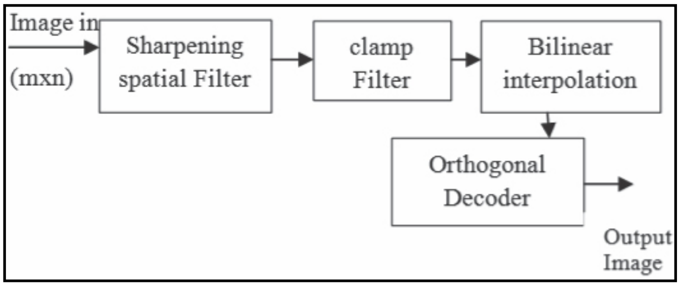


Fig.1: Block diagram of the proposed scaling algorithm

First, the input pixels of the original images are filtered by the sharpening spatial filter to enhance the edges and remove associated noise. Second, the filtered pixels are filtered again by the clamp filter to smooth unwanted discontinuous edges of the boundary regions. Finally, the pixels filtered by both of the sharpening spatial and clamp filters are passed to the bilinear interpolation for up-downscaling. To conserve computing resource and memory buffer, these two filters are simplified and combined into a combined filter. Finally blurred interpolated output can be removed by orthogonal decoder.

B. Spatial and Clamp Filter

The sharpening spatial filter, a kind of high-pass filter, is used to reduce blurring artifacts and defined by a kernel to increase the intensity of a center pixel relative to its neighboring pixels. The clamp filter [6], a kind of low-pass filter, is a 2-D Gaussian spatial domain filter and composed of a convolution kernel array. It usually contains a single positive value at the center and is completely surrounded by ones [15]. The clamp filter is used to reduce aliasing artifacts and smooth the unwanted discontinuous edges of the boundary regions.

C. Convolution Kernels

The sharpening spatial and clamp filters can be represented by convolution kernels A larger size of convolution kernel will produce higher quality of images. However, a larger size of convolution filter will also demand more memory and hardware cost. In previous work [15], each of the sharpening spatial and clamp filters was realized by a 2-D 3 × 3 convolution kernel as shown in Fig. 2(a).

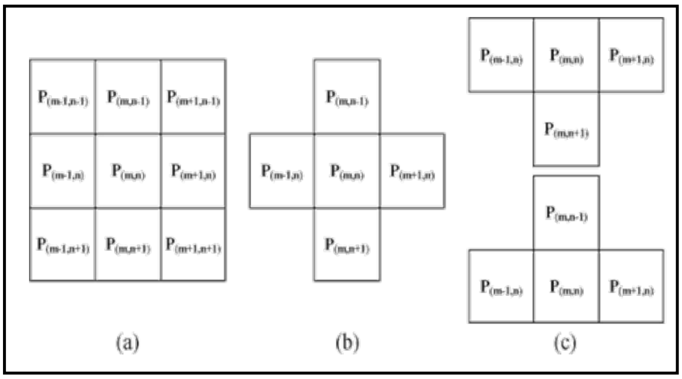


Fig.2. Weights of the convolution kernels (a) 3 × 3 convolution kernel (b) Cross-model convolution kernel (c)
T-model and inversed T-model convolution kernels

The sharpening spatial and clamp filters can be represented by convolution kernels A larger size of convolution kernel will produce higher quality of images. However, a larger size of convolution filter will also demand more memory and hardware cost. In previous work [15], each of the sharpening spatial and clamp filters was realized by a 2-D 3 × 3 convolution kernel as shown in Fig. 2(a). It demands at least a four-line-buffer memory for two 3 × 3 convolution filters. To reduce the complexity of the 3 × 3 convolution kernel, a cross-model formed is used to replace the 3 × 3 convolution kernel, as shown in Fig. 2(b). It successfully cuts down on four of nine parameters in the 3 × 3 convolution kernel. Furthermore, to decrease more complexity and memory requirement of the cross-model convolution kernel, T-model and inversed T-model convolution kernels are proposed for realizing the sharpening spatial and clamp filters. In T-model convolution kernel is composed of the lower four parameters of the cross-model, and the inversed T-model convolution kernel is composed of the upper four parameters.

D. Combined Filter

In proposed scaling algorithm, the input image is filtered by a sharpening spatial filter and then filtered by a clamp spatial filter again. Although the sharpening spatial and clamp filters are simplified by T-models and inversed T-models, it still needs two line buffers to store input data or intermediate values T-model or inversed T-model, should be combined together into a combined filter

$$P'_{(m,n)} = \begin{bmatrix} -1 & S-C & SC-2 & S-C & -1 \\ \square & -2 & S-C & -2 & \square \\ \square & \square & -1 & \square & \square \end{bmatrix} \Big/ [(S-3)\times(C+3)]$$

$$P^{-}_{(m,n)}$$

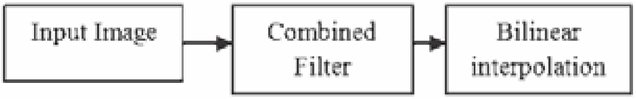


Fig. 3: Block diagram of combined filter

where S and C are the sharp and clamp parameters. is the filtered result of the target pixel$_{)}$ by the combined filter.

E. Bilinear Interpolation

Interpolation is the process of determining the values of a function at positions lying between its samples. It achieves this process by fitting a continuous function through the discrete input samples. This permits input values to be evaluated at arbitrary positions in the input, not just those defined at the sample points. The process of interpolation is one of the fundamental operations in image processing. The image quality highly depends on the used interpolation technique.

In the proposed scaling algorithm, the bilinear interpolation method is selected because of its characteristics with low complexity and high quality. The bilinear interpolation is an operation that performs a linear interpolation first in one direction and, then again, in the other direction. The output pixelcan be calculated by the operations of the linear interpolation in both x- and y-directions with the four nearest neighbor pixels. The target
Pixel can be calculated by

$$P_{(K,1)} = (1-dx)\times(1-dy)\times P_{(m,n)} + dx \times (1-dy) \times P_{(m+1,n)} + (1-dx)\times dy \times P_{(m,n+1)} + dx \times dy \times P_{(m+1,n+1)}$$

Where $P_{(m,n)}$, $P_{(m,n+1)}$, $P_{(m+1,n)}$, and $P_{(m+1,n+1)}$are the four nearest neighbor pixels of the original image and the dx and dy are scale parameters in the horizontal and vertical directions. It costs a considerable chip area to implement a bilinear interpolator with eight multipliers and seven adders. Thus, an algebraic manipulation skill has been used to reduce the computing resources of the bilinear interpolation.

III. VLSI Architecture

The proposed scaling algorithm consists of two combined prefilters and bilinear interpolator. For VLSI implementation, the bilinear interpolator can directly obtain two input pixels from two combined prefilters. It consists of four main blocks: a register bank, a combined filter a bilinear interpolator and controller. The details of each part will be described in the following sections.

A. Register Bank

In this brief, the combined filter is filtering to produce the target pixels of and by using ten source pixels. The register bank is designed with a one-line memory buffer, which is used to provide the ten values for the immediate usage of the combined filter.

Figure shows the architecture of the register bank with a structure of ten shift registers.

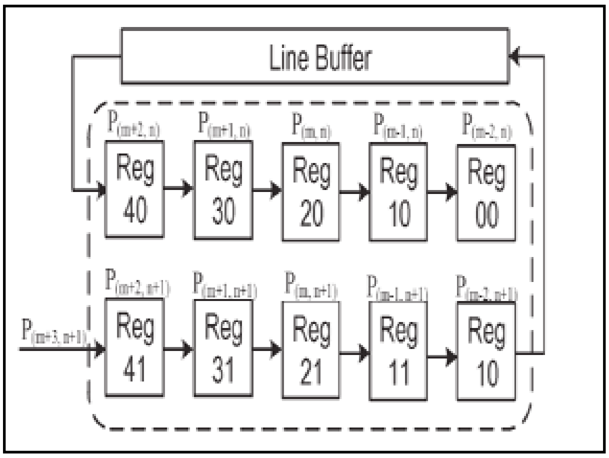
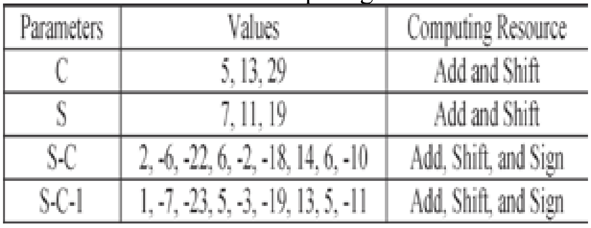


Fig. 4: Architecture of register bank

B. Bilinear Interpolator And Controller

The T-model or inversed T-model filter consists of three reconfigurable calculation units (RCUs), one multiplier–adder (MA), three adders (+), three subtracters (−), and three shifters (S).But the combined filter circuit consist of one MA and three RCUs. The MA can be implemented by a multiplier and an adder. The RCU is designed for producing the calculation functions of (S-C) and (S-C-1) times of the source pixels value, which must be implemented with C and S parameters. The C and S parameters can be set by users according to the characteristics of the images. Table I lists the parameters and computing resource for the RCU.

Table 1 : Parameters and computing resource for RCU



Parameters	Values	Computing Resource
C	5, 13, 29	Add and Shift
S	7, 11, 19	Add and Shift
S-C	2, -6, -22, 6, -2, -18, 14, 6, -10	Add, Shift, and Sign
S-C-1	1, -7, -23, 5, -3, -19, 13, 5, -11	Add, Shift, and Sign

With the selected C and S values listed in Table I, the gain of the clamp or sharp convolution function is {8, 16, 32} or {4, 8, 16}, which can be eliminated by a shifter rather than a divider.

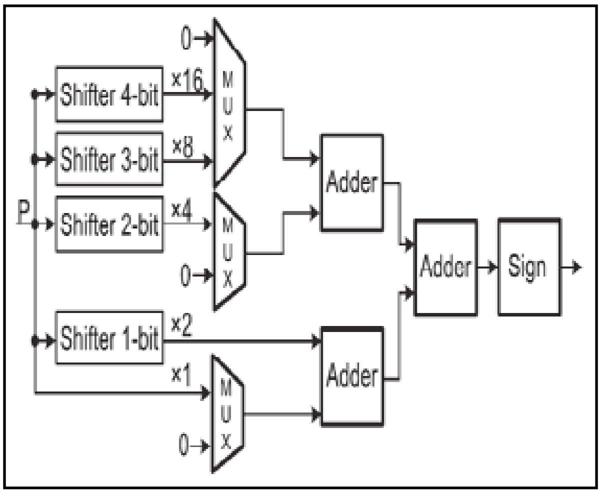


Fig. 5 : Architecture of the RCU

It consists of four shifters, three multiplexers (MUX), three adders, and one sign circuit. By this RCU design, the hardware cost of the combined filters can be efficiently reduced. The controller is implemented by a finite-state-machine circuit. It produces control signals to control the timing and pipeline stages of the register bank, combined filter, and bilinear interpolator.

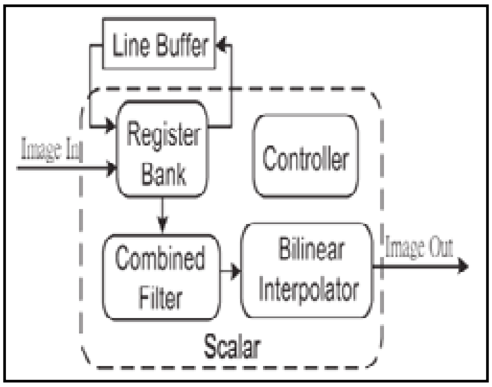


Fig. 6 : Block diagram of scaling processor using VLSI Design

C. Orthogonal Decoder

The block diagram of the Orthogonal Decoder is shown Fig 1.It consists of a cyclic shift register, XOR matrix, and Majority gate and control unit.

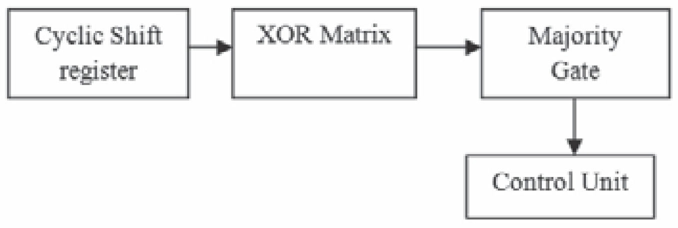


Fig. 7 : Block diagram of orthogonal decoder

Cyclic Shift Register:

In digital circuits, a shift register is a cascade of flip flops, sharing the same clock, in which the output of each flip-flop is connected to the “data” input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the “bit array” stored in it, shifting in the data present at its input and shifting out the last bit in the array, at each transition of the clock input.

There are different types of shift registers are used. Here we use cyclic Shift Register in that, the same values are again and again fed to the shift register until we get the proper output. Input of the shift register in taken from the register bank.

XOR Matrix and Majority Gate:

In XOR Matrix, the values are taken from the shift register. And these values are XOR one by one. And Parity check sum is performed. That is counting the number of zeros and ones. If number of ones is greater than zeros error correction is performed. Otherwise the bits are under gone cyclic shift. These Processes are done in Majority Gate. And it can be done until all the bits are evaluated.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

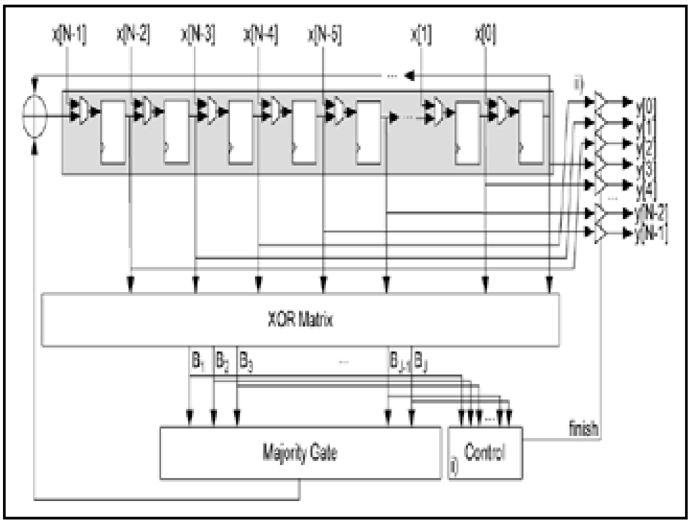


Fig. 8 : Architecture of Orthogonal Decoder

Control Unit:

It consists of Shifregister,Counter,Finite State Machine(FSM)

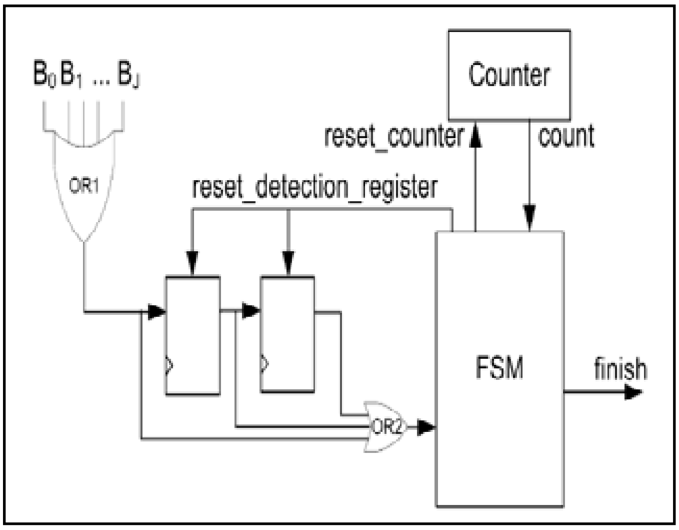


Fig. 9 : Architecture of Control Unit

- The control unit manages the detection process.
- It uses a counter that counts up to three, which distinguishes the first three iterations of the ML decoding.
- In these first three iterations, the control unit evaluates the {Bj} by combining them with the OR1 function.
- This value is fed into a three-stage shift register, which holds the results of the last three cycles.
- In the third cycle, the OR2 gate evaluates the content of the detection register.
- When the result is “0,” the FSM sends out the finish signal indicating that the processed word is error-free.
- In the other case, if the result is “1,” the ML decoding process runs until the end.

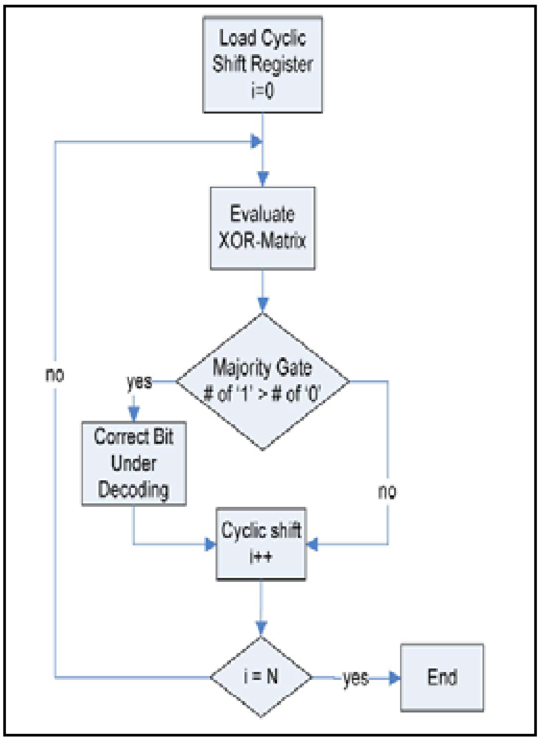

Load Cyclic Shift Register i=0
Evaluate XOR-Matrix
Majority Gate # of '1' > # of '0'
yes
no
no
Correct Bit Under Decoding
Cyclic shift i++
i = N
yes
End

Fig. 10 : Flow chart for Decoding Alogrithm

IV. Simulation Results

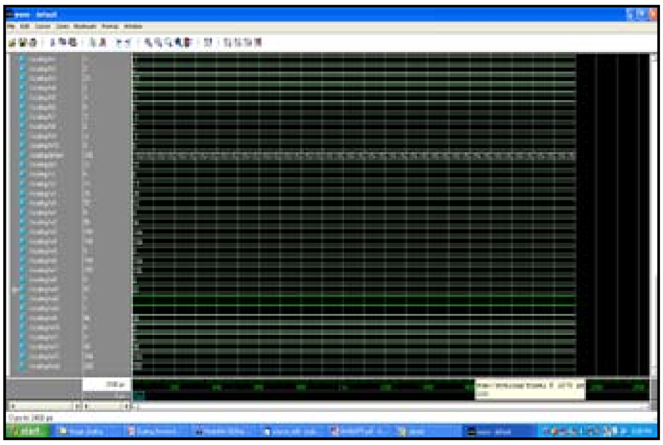


Fig. 11 : Simulation results for scaled image

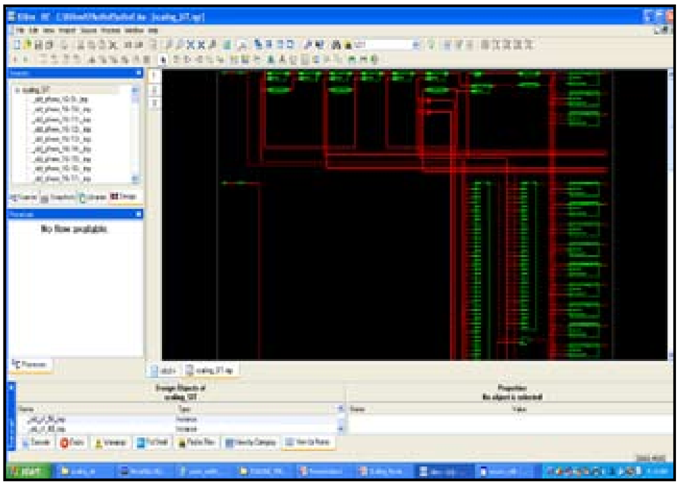


Fig. 12 : Simulation results for RTL Schematic

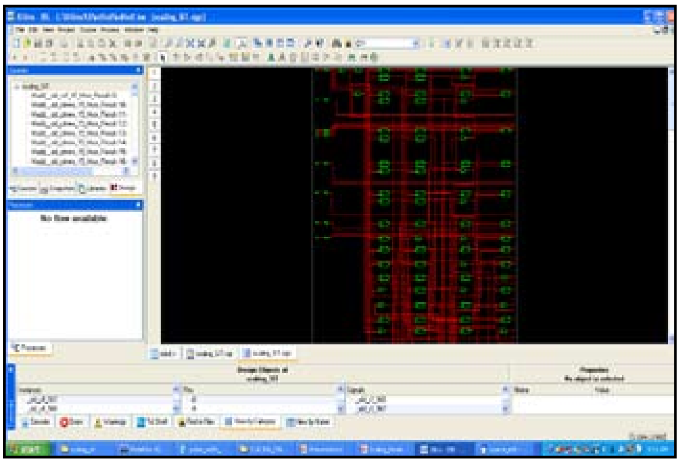


Fig. 13 : Simulation results for Technology Schematic

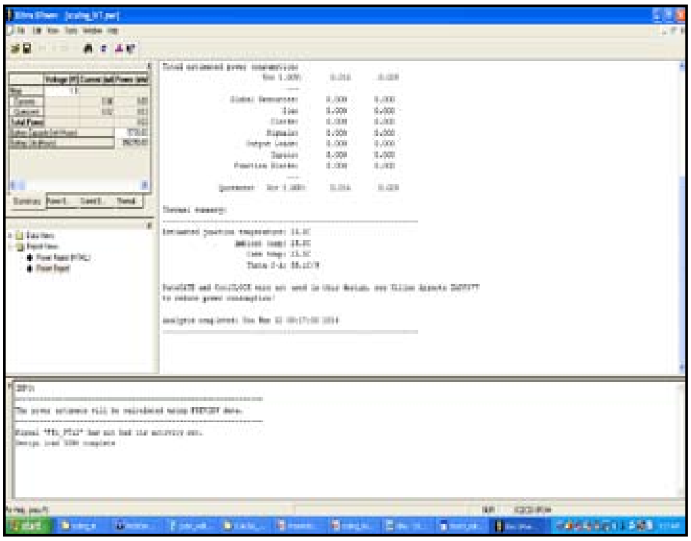
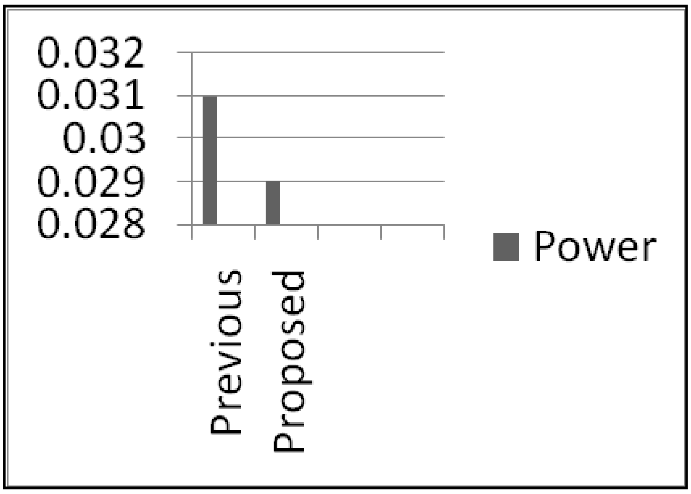


Fig. 14 Simulation results for Power consumption

Table 2 : Performance Comparison

Parameter	Previous	Proposed
Power Consumption	0.031mW	0.029mW



Graph 1 : Performance Analysis

V. Conclusion

In this paper, a novel adaptive scaling algorithm is proposed for developing a low-cost, low-power, and high-quality VLSI scaling circuit for image zooming applications. Bilinear interpolation is selected as an interpolation method due to its low complexity and high quality. A clamp filter and a sharpening spatial filter are added as pre-filters to solve the shortcomings of blurring and aliasing effects caused by the bilinear interpolation. The filter combining, hardware sharing, and reconfigurable techniques had been used to reduce hardware cost. Simulation results demonstrate significantly reduction in power consumption without performance degradation. Furthermore, the idea of image scaling can be implement using several algorithms for achieve more quality.

V. Acknowledgement

References

[1] *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.*

[2] *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.*

[3] *M. Fons, F. Fons, and E. Canto, "Fingerprint image processing acceleration through run-time reconfigurable hardware," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 57, no. 12, pp. 991–995, Dec. 2010.*

[4] *J. W. Han, J. H. Kim, S. H. Cheon, J.O.Kim, and S. J. Ko, "Anovel image interpolation method using the bilateral filter," IEEE Trans. Consum.Electron., vol. 56, no. 1, pp. 175–181, Feb. 2010.*

[5] *P. Y. Chen, C. C. Huang, Y. H. Shiau, and Y. T. Chen, "A VLSI implementation of barrel distortion correction for wide-angle camera images,"IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 56, no. 1, pp. 51–55,Jan. 2009.*

[6] *P. Y. Chen, C. Y. Lien, and C. P. Lu, "VLSI implementation of an edge oriented image scaling processor," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 17, no. 9, pp. 1275–1284, Sep. 2009.*

[7] *X. Zhang and X.Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," IEEE Trans. Image Process.,vol. 17, no. 6, pp. 887–896, Jun. 2008.*

[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] C. C. Lin, M. H. Sheu, H. K. Chiang, W. K. Tsai, and Z.C. Wu, "Real-time FPGA architecture of extended linear convolution for digital image scaling," in Proc. IEEE Int. Conf. Field-Program. Technol., 2008, pp. 3*81–384.*

[10] *C. C. Lin, Z. C. Wu, W. K. Tsai, M. H. Sheu, and H. K. Chiang, "The VLSI design of winscale for digital image scaling," in Proc. IEEE Int. Conf. Intell. Inf. Hiding Multimedia Signal Process, Nov. 2007, pp. 511–514.*

Author's Profile

J. Jeba Priya received her B.E. degree in Electronics and Communication Engineering from the Kamaraj College of Engineering and Technology, Virudhunagar, India, in 2012. Currently doing M.E - VLSI Design in Sethu Institute of Technology, Virudhunagar India. Her research interest includes: low Power VLSI and Image Processing.

Mrs. M.Annalakshmi received her B.E., from Thigarajar College of Engineering in 1998 and completed her M.E from Alagappa Chettiar College of Engineering and Technology in 2005.Presently working as aAssosiate Professor in the Department of ECE at Sethu Institute of Technology, Tamilnadu, India. Her current research areas includes: Image Processing and Signal Processing.

.