International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.04,Issue.24,
July-2015,
Pages:4538-4543

www.ijsetr.com

Design and Implementation of Quality Optimized Image Scaling Processor using VLSI Technology

M. NIKHITHA[1], K.S.R. MURTHY[2]

[1]PG Scholar, Dept of ECE, MVSR Engineering College, Osmania University, Telangana, India,
Email: nithareddy29@gmail.com.

[2]Professor, Dept of ECE, MVSR Engineering College, Osmania University, Telangana, India,
Email: ksrmurthy48@yahoo.co.in.

Abstract: In this brief, a low-complexity, low-memory requirement, and quality algorithm is proposed for VLSI implementation of an image scaling processor. The proposed image scaling algorithm consists of a sharpening spatial filter, a clamp filter, and a bilinear 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. Furthermore, two T-model or inversed T-model filters are combined into a combined filter which requires only a one-line-buffer memory. Moreover, a reconfigurable calculation unit is invented for decreasing the hardware cost of the combined filter. Moreover, the computing resource and hardware cost of the bilinear interpolator can be efficiently reduced by an algebraic manipulation and hardware sharing techniques. The VLSI architecture in this work can achieve 197MB memory, 4.004 delay, and 0.030MW power consumption. Compared with previous low-complexity techniques, this work reduces gate counts and requires only a one-line-buffer memory.

Keywords: Sharpening Spatial Filter, Clamp Filter, and Bilinear Interpolation, Image Scaling Processor.

I. INTRODUCTION

Image scaling has been widely applied in the fields of digital imaging and mainly on Electronic based imaging devices. Image scaling is the process of scaling down the high – quality frames or pictures to fit small size LCD panel of electronic displays as digital PDA's are growing fast. Scaling algorithm can be classified into two types as polynomial-based and non-polynomial-based. Nearest neighbor algorithm is the uncomplicated polynomial algorithm, but resultant images are with full of aliasing artifacts. Bilinear interpolation algorithm and Bi-cubic algorithm are the other polynomial based methods widely used to target the pixels. For the past decade many non-polynomial high performance methods have been proposed [4]. The techniques like bilateral filter, interpolation, and autoregressive model. These methods are used to boost the image quality by reducing the artifacts. These Image scaling algorithms are very complex to implement in VLSI. Thus, for fast, real time applications, less complexity based algorithms are necessary. Area pixel model Winscale method is previously proposed for less complexity methods. Adding of sharpening spatial and clamp filers [3] effectively improves the image quality with bilinear interpolation algorithm. By these cost of the hardware and memory also reduced.The design and implementation of image processing algorithms in VLSI is an expanding area of research. The complexity of VLSI design is the main obstacle that blocks the widespread use of it in real time image processors. In this work a high quality algorithm with low complexity and low memory is used. To reduce the memory requirement and computation cost filter combining, hardware sharing and reconfigurable techniques had been used in the scaling algorithm. Due to computational efficiency and qualitative stability bilinear interpolation algorithm is selected by trading off complexity and quality. Because of its low complexity and simple architecture bilinear interpolation [10] is efficient for VLSI implementation. The coding can be synthesized using Xilinx ISE Design Suite.

II. LITERATURE REVIEW

In this section some related works are discussed below. A brief introduction to blocks in image scaling processor is given below. Many different VLSI architectures have been proposed for image scaling over the literature survey [8]-[11]-[5]. For bilinear interpolation technique they can be well characterized with respect to zooming image and less delay as well as suitability for logic optimization and synthesis.

A. Comparison of different interpolations

1. VLSI Architectures for Image Interpolation

Image interpolation is a method of estimating the values at unknown points using the known data points. This procedure is used in expanding and contrasting digital images. In this survey, different types of interpolation algorithm and their hardware architecture have been analyzed and compared. They are bilinear, winscale, bi-cubic, linear convolution, extended linear, piecewise linear, adaptive bilinear, first order polynomial, and edge enhanced interpolation architectures. The algorithms are implemented for different types of field programmable gate array (FPGA) and/or by different types of complementary metal oxide semiconductor (CMOS) technologies like TSMC 0.18 and TSMC 0.13. These interpolation algorithms are compared based on different types of optimization such as gate count, frequency, power, and memory buffer. The goal of this work is to analyze the different very large scale integration (VLSI) parameters like area, speed, and power of various implementations for image interpolation. From the survey followed by analysis, it is observed that the performance of hardware architecture of image interpolation can be improved by minimising number of line buffer memory and removing superfluous arithmetic elements on generating weighting coefficient.

2. Interpolation Methods

This chapter briefs a review of different image interpolation algorithms such as winscale, bi-cubic, linear, polynomial convolution, bilinear, and adaptive scaling algorithms. Various algorithms are discussed by their hardware characteristics and visual quality. The hardware characteristics such as area utilization and speed and power consumptions are mainly focused upon. The quality of the interpolation algorithms can be expressed in dB (decibel) with peak signal-to-noise ratio (PSNR) of scaled image and the original image.

3. Digital Image Scaling Algorithm

Andreadis and Amantiadis [6] proposed an image interpolation algorithm for both grayscale and colour images of any resolution in any scaling factor .This algorithm uses a mask of maximum four pixels and calculates the final luminosity of each pixel combining two factors such as the percentage of area that mask covers from each source pixel and the difference in luminosity between the source pixels. This interpolation operates on linear area domain and uses continuous area filtering. It can perform both upscale and downscale processes for fast real-time implementation. This method is implemented on Quartus II FPGA with the operating frequency of 55 MHz. The hardware architecture of this interpolation uses 20 additions and 13 multiplications. The root mean square error (RMSE) of this method is very less than the RMSE of other interpolation scheme like nearest neighbour, bilinear, and winscale.

- Line buffer
- Register bank
- Sharpening filter
- Clamp filter
- Reconfigurable calculation unit
- Multiplier adder unit
- Bilinear interpolator

4. An Edge-Oriented Image Scaling Algorithm

An edge-oriented area-pixel scaling algorithm has been presented by Chen et al [6]. This algorithm aims to achieve low cost; the area-pixel scaling technique is implemented with low-complexity VLSI architecture in this design. A simple edge catching technique is adopted to preserve the image edge features effectively so as to achieve better image quality. This scaling processor can support floating-point magnification factor and preserve the edge features efficiently by taking into account the local characteristic that existed in those available source pixels around the target pixel. Furthermore, it handles streaming data directly and requires only small amount of memory like one-line buffer rather than a full frame buffer. The FPGA implantation of this method is obtained by Xilinx Virtex-II XC2VP50. The FPGA implementation utilizes only about 581 CLBs but the bi-cubic algorithm utilizes about 890 CLBs. Thus, the seven stage pipelined VLSI architecture of this image scaling processor contains 10.4 K gate counts and yields a processing rate of about 200 MHz by using TSMC 0.18 μm technology, but the bi-cubic architecture utilizes about 29 K gates. The quality comparison shows that this method has higher PSNR as 38.16 than the other methods such as nearest neighbour, bilinear, bi-cubic, area-pixel scaling, and winscale algorithms. The PSNR is obtained for the test image (crowd) for image reduction from the size of 600 × 600 to 512 × 512.

III. PROPOSED SCALING ALGORITHM

Fig1 shows the block diagram of the proposed scaling algorithm. The sharpening spatial and clamp filters act as pre- filters to reduce blurring and aliasing artifacts produced by the bilinear interpolation. The input pixels of the original images are filtered by the spatial filter to remove associated noise and enhance the edges.

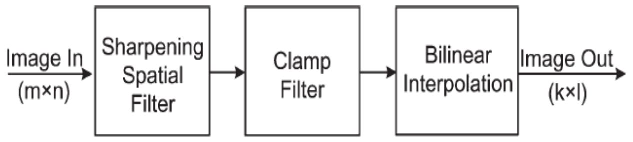


Fig.1. shows the block diagram of the proposed scaling algorithm for image zooming.

To decrease the blurring and aliasing artifacts caused by the bilinear interpolation, the sharpening spatial filter and clamp filter are added as prefilters. Initially, the input pixels of the images existing from the beginning are passed through a filter by the sharpening spatial filter to extend the edges and get rid of associated noise. Next, the filtered pixels are passed through a filter again by the clamp filter to smooth the noise and interpolate the missing samples along the edge direction. Coming to the end, the pixels are again filtered by the prefilters and then moved to the bilinear interpolation to perform up-1 downscaling. To protect the calculating resource and memory buffer, these prefilters are made understandable and joined to form a combined filter. Unwanted discontinuous edges and boundaries are filtered by clamp filter. To conserve computing resource and memory buffer, these two filters are simplified and combined into a combined filter.

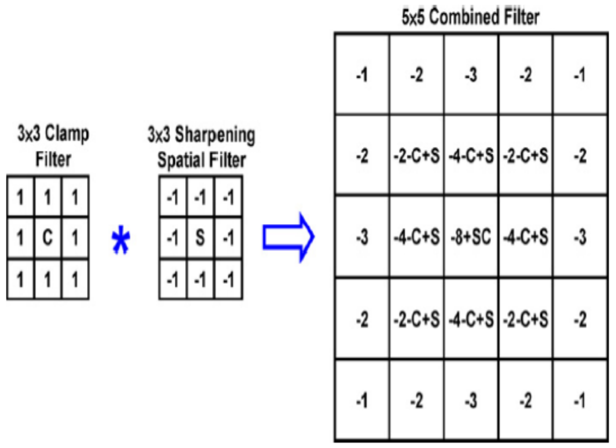
A. Clamp Filters and sharpening filter

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, 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. The clamp filter is used to reduce aliasing artifacts and smooth the unwanted discontinuous edges of the boundary regions. 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. For example, a 6 × 6 convolution filter demands at least a five-line-buffer memory and 36 arithmetic units, which is much more than the two-line-buffer memory and nine arithmetic units of a 3 × 3 convolution filter. In our previous work, each of the sharpening spatial and clamp filters was realized by a 2-D 3 ×3 convolution kernel as shown in Fig. 3.3. It demands at least a four-line-buffer memory for two 3 × 3 convolution filters.

For example, if the image width is 1920 pixels, 4 × 1920 × 8 bits of data should be buffered in memory as input for processing. 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. As shown in Fig. 2(c), the 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. In the proposed scaling algorithm, both the T-model and inversed T-model filters are used to improve the quality of the images simultaneously. The T-model or inversed T-model filter is simplified from the 3 × 3 convolution filter of the previous work, which not only efficiently reduces the complexity of the convolution filter but also greatly decreases the memory requirement from two to one line buffer for each convolution filter. The T-model and the inversed T-model provide the low-complexity and low memory- requirement convolution kernels for the sharpening spatial and clamp filters to integrate the VLSI chip of the proposed low-cost image scaling processor.

B. Combined Filter

The sharpening spatial filter and clamp filter can be represented by convolution kernels [4]. Convolution kernel is the matrix of weights. The image quality can be increased if a large sized convolution kernel is used. But this increases the hardware cost and memory requirement. For example, the use of a 3×3 convolution sharpening spatial filter and 3×3 convolution clamp filter produces a 5×5 combined filter is shown in Figure



3x3 Clamp Filter
1 1 1
1 C 1
1 1 1
*
3x3 Sharpening Spatial Filter
-1 -1 -1
-1 S -1
-1 -1 -1
5x5 Combined Filter
-1 -2 -3 -2 -1
-2 -2-C+S -4-C+S -2-C+S -2
-3 -4-C+S -8+SC -4-C+S -3
-2 -2-C+S -4-C+S -2-C+S -2
-1 -2 -3 -2 -1

Fig.2.Combination of clamp and sharpening filter.

Where S and C are the sharpening and clamp filter parameters. This requires four line buffer memory and twenty five arithmetic units. To reduce the complexity the 3×3 convolution kernel can be replaced by a cross model which cuts down four parameters. For further improvement a T-model or inverse T-model convolution kernel can be used to design the filters. Then, two line buffers are required to store input data or intermediate values of filtering. The filter combining technique can be used to decrease the memory requirement to one line buffer and computation cost can also be decreased. The 3×3, cross model and T-model convolution kernels are shown in Figure. 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 for each T-model or inversed T-model filter. Thus, to be able to reduce more computing resource and memory requirement, sharpening spatial and clamp filters, which are formed by the T-model or inversed T-model, should be combined together into a combined filter.

A T-model sharpening spatial filter and a T-model clamp filter have been replaced by a combined T-model filter as shown in (3). The combined inversed T-model filter can be produced in the same way. In the new architecture of the combined filter, the two T-model or inversed T-model filters are combined into one combined T-model or inversed T-model filter. By this filter-combination technique, the demand of memory can be efficiently decreased from two to one line buffer, which greatly reduces memory access requirements for software systems or hardware memory costs for VLSI implementation. In Fig.5, the stages 1 and 2 show the computational scheduling of a T-model combined and an inverse Tmodel filter. The T-model or inversed T-model filter consists of one multiplier–adder, three reconfigurable calculation units, three adders (+), three subtracters (−), and three shifters. The values of the ten source pixels can be obtained from the register bank as mentioned earlier. Bilinear interpolation is an image-restoring algorithm, which linearly interpolates four nearest-neighbour pixels of an unrestored image to obtain the pixel of a restored image. The principle of the bilinear interpolation algorithm is that it executes linear interpolation in one direction, and then repeating the same function in the other direction. Fig.4 depicts a block which includes 8 input pixels of the original image. 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.

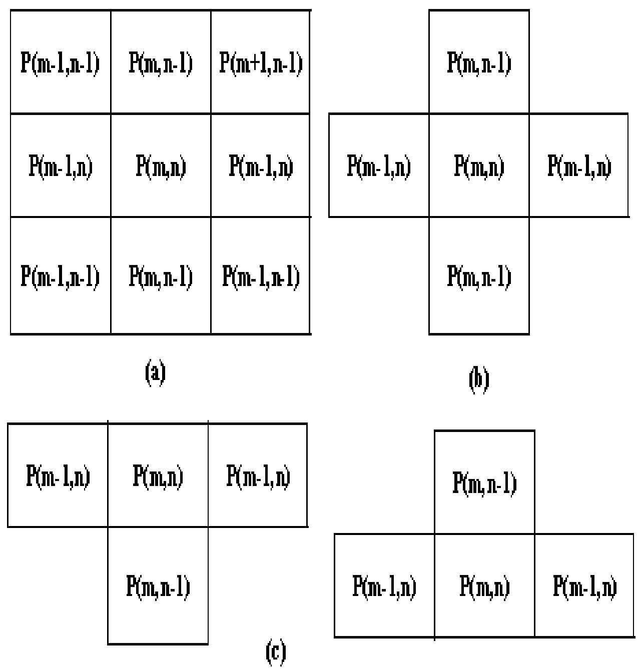


Fig.3.weigts of the convolution kernels (a) 3x3 convolution kernel (b) cross model convolution kernel (c) T-model and inverse T-model.

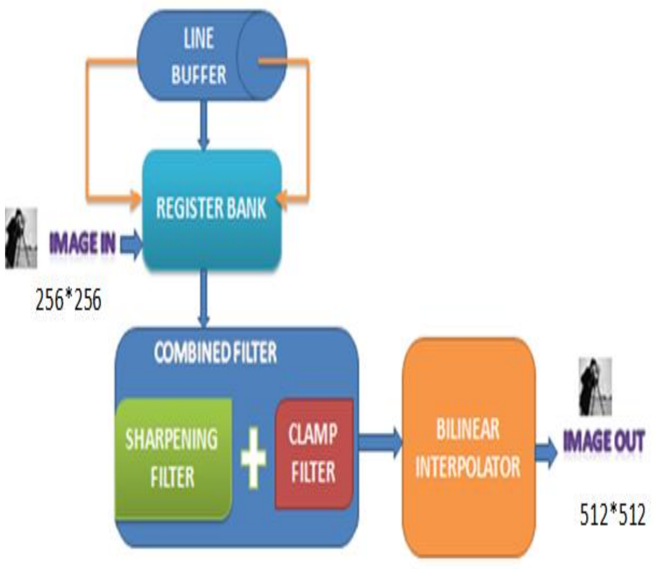


Fig.4. Block diagram of the VLSI architecture for proposed real-time image scaling processor.

The output pixel P(k,l) can be calculated by the operations of the linear interpolation in both x- and y-directions with the four nearest neighbor pixels. The target pixel P(k,l) can be calculated by

$$P_{(K,1)} = P'(m,n) + dx\{P'(m,n) - [P'(m,n) + dy(P'(m,n) - p(m,n+1)]\} \tag{1}$$

The stages 3, 4, 5, and 6 in Fig.5 show the four-stage pipelined architecture of the bilinear interpolation. The two-stages of pipelined multipliers are used to shorten the delay path of the bilinear interpolator. The input values of P'(m, n) and P'(m,n+1) are obtained from the combined filter and symmetrical circuit. The bilinear interpolator circuit results in P(k,l).

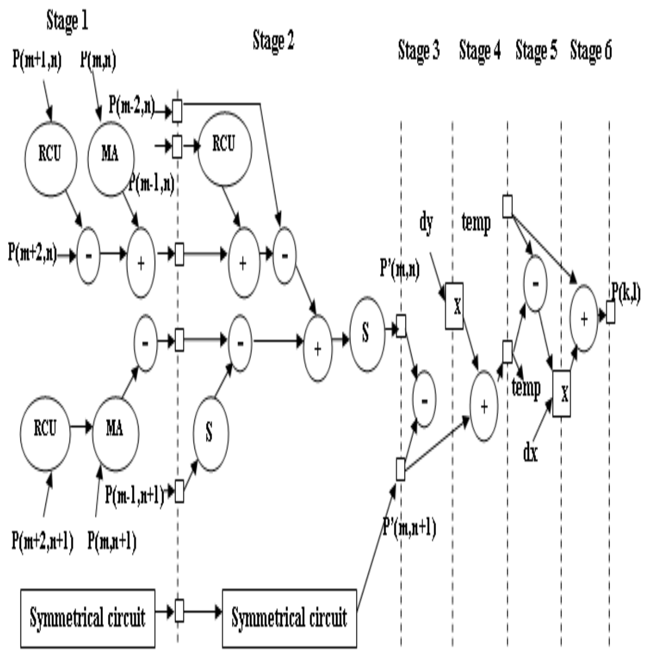


Fig.5. Six-stage pipelined architecture of the combined filter and bilinear interpolator.

The original equation of bilinear interpolation is presented and the simplifying procedures of bilinear interpolation can be described. Since the function of dy × (P(m,n+1) − P(m,n)) + P(m,n) appears twice in, one of the two calculations for this algebraic function can be reduced by the characteristic of the executing direction in bilinear interpolation, the values of dy for all pixels that are selected on the vertical axis of n row equal to n + 1 row, and only the values of dx must be changed with the position of x. The result of the function "[P(m,n) + dy × (P(m,n+1) − P(m,n))]" can be replaced by the previous result of "[P(m+1,n) + dy × (P(m+1,n+1) − P(m+1i,n))]" as shown in (4). The simplifying procedures successfully reduce the computing resource from eight multiply, four subtract, and three add operations to two multiply, two subtract, and two add operations. where P(m,n), P(m+1,n), P(m,n+1), 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. By (2), we can easily find that the computing resources of the bilinear interpolation cost eight multiply, four subtract, and three addition operations. 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.

The original equation of bilinear interpolation is presented in(2), and the simplifying procedures of bilinear interpolation can be described from (2)–(4). Since the function of dy × (P(m,n+1) − P(m,n)) + P(m,n) appears twice in (4), one of the two calculations for this algebraic function can be reduced.

$$P_{(x,1)} = \left[(1-dy) X P_{(m+1,n)} + dy\, X\, P_{(m+1,n+1)}\right] X\, dx + \left[(1-dy) X P_{(m,n)} + dy\, X\, P_{(m,n+1)}\right](1-dx) \tag{2}$$

$$P_{(x,1)} = \left[P_{(m+1,n)} + dy X \left(P_{(m+1,n+1)} - P_{(m,n+1)}\right)\right] X\, dx + \left[P_{(m,n)} + dy\, X\, P_{(m,n+1)} - P_{(m,n)}\right](1-dx) \tag{3}$$

$$P_{(x,1)} = \left[P_{(m+1,n)} + dy X \left(P_{(m+1,n+1)} - P_{(m+1,n)}\right)\right] - \left[P_{(m,n)} + dy\, X\, P_{(m,n+1)} - P_{(m,n)}\right] dx + \left[P_{(m,n)} + dy\, X\, P_{(m,n+1)} - P_{(m,n)}\right] \tag{4}$$

By the characteristic of the executing direction in bilinear interpolation, the values of dy for all pixels that are selected on the vertical axis of n row equal to n + 1 row, and only the values of dx must be changed with the position of x. The result of the function ―[P(m,n) + dy × (P(m,n+1) − P(m,n))] can be replaced by the previous result of ―[P(m+1,n) + dy × (P(m+1,n+1) − P(m+1i,n))]‖ as shown in (4). The simplifying procedures successfully reduce the computing resource from eight multiply, four subtract, and three add operations to two multiply, two subtract, and two add operations.

IV. RESULTS

As per the six stage pipeline architecture, the image is initially passed through the combined filter, where the aliasing and blurring effects are removed. After that bilinear interpolation is performed on the filtered output to obtain a scaled image. The input image used for the image scaling process is jpg image. The image matrix is represented by 256x 256 pixels, which totally contains 65536 bytes pixel values.

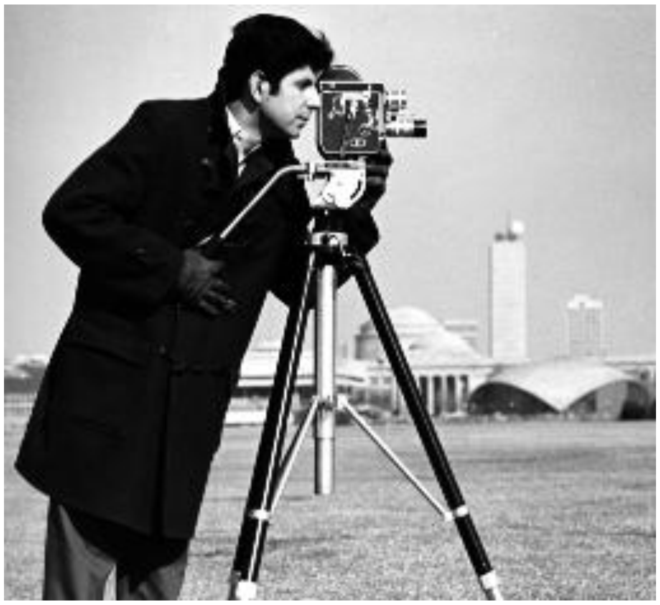
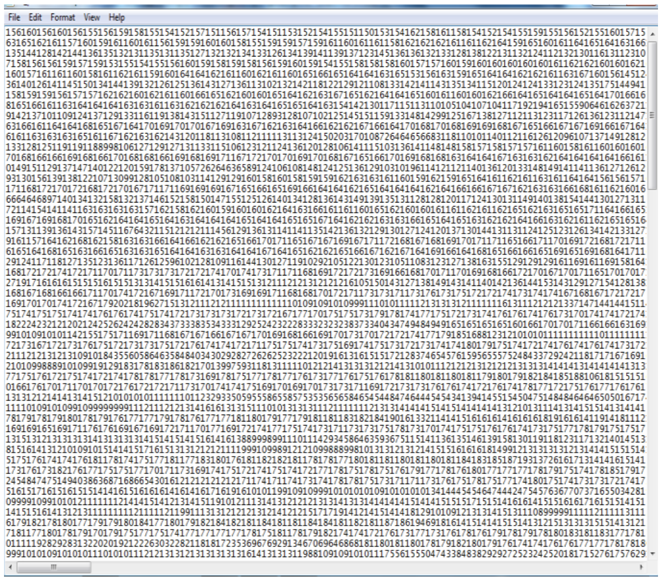
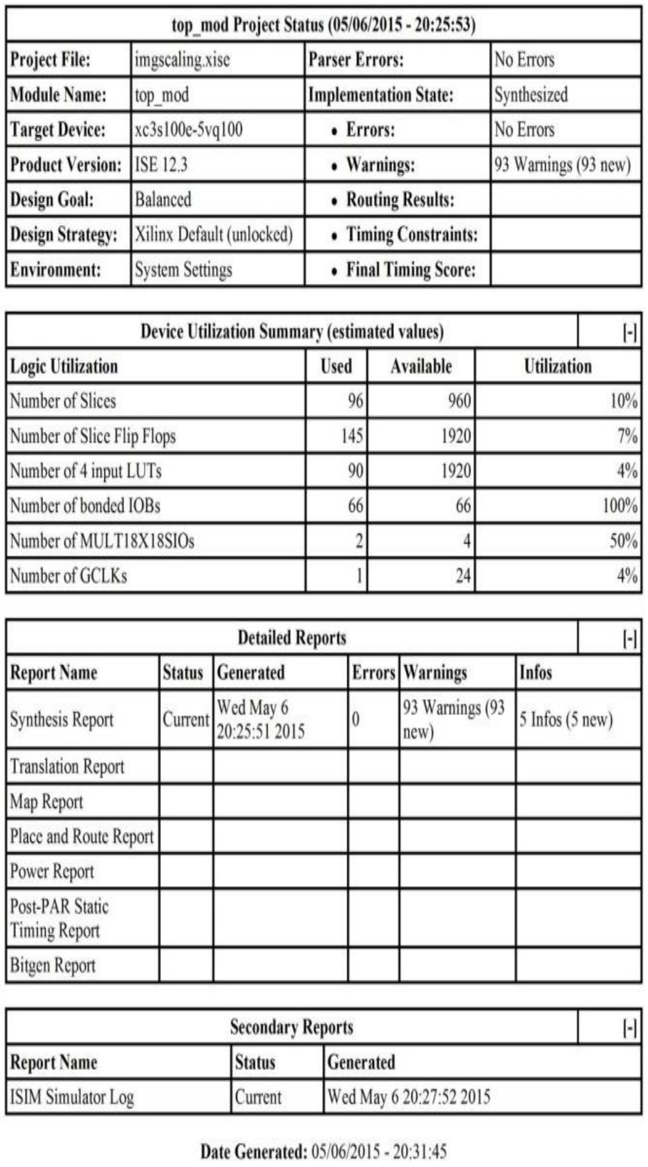


Fig.6. Input Image.



```
[illegible]
```

Fig.7. Input Image pixel file created from matlab (256*256).



top_mod Project Status (05/06/2015 - 20:25:53)			
Project File:	imgscaling.xise	Parser Errors:	No Errors
Module Name:	top_mod	Implementation State:	Synthesized
Target Device:	xc3s100e-5vq100	• Errors:	No Errors
Product Version:	ISE 12.3	• Warnings:	93 Warnings (93 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	96	960	10%
Number of Slice Flip Flops	145	1920	7%
Number of 4 input LUTs	90	1920	4%
Number of bonded IOBs	66	66	100%
Number of MULT18X18SIOs	2	4	50%
Number of GCLKs	1	24	4%

Detailed Reports					[-]
Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Wed May 6 20:25:51 2015	0	93 Warnings (93 new)	5 Infos (5 new)
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports		[-]
Report Name	Status	Generated
ISIM Simulator Log	Current	Wed May 6 20:27:52 2015

Date Generated: 05/06/2015 - 20:31:45

Fig. 8. Design summery report of proposed novel image scaling technique.

The filtered pixels p'(m, n) and p'(m, n+ 1) are given as input to the bilinear interpolator. The fig.8 shows the bilinear interpolation output waveform with new scaled pixels. The scaled output image pixel values displayed using the xilinx software. The output image pixels of size 512 x512 is obtained. The total estimated power consumption of this image scaling algorithm is 0.03W with peak memory usage of about197972kb.

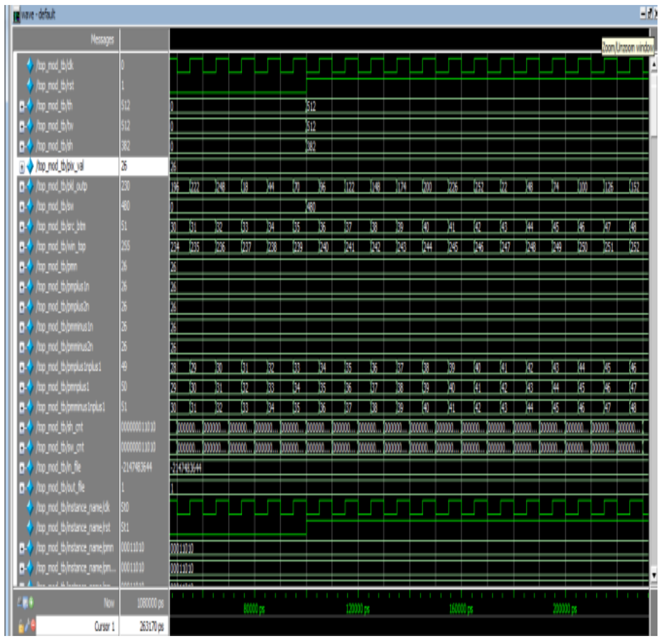


Fig.9.Bilinear interpolation output waveform with scaled Pixel values

V. REFERENCES

[1] A. Amanatiadis, I. Andreadis, and K. Konstantinidis, "Design and implementation of a fuzzy area-based image-scaling technique," IEEE Transaction on Instrumentation and measurement, Vol. 57, No. 8 , pp. 1504-1513, Aug. 2008.

[2] C. C. Huang, P. Y. Chen, and C. H. Ma,(2012) "A novel interpolation chip for real-time multimedia application," IEEE Transaction on Circuits and Systems for Video Technology, Vol. 22, no. 10 , pp. 1512-1525, Oct. 2004.

[3] C. H. Kim, S. M. Seong, J. A. Lee, and L. S. Kim, "Winscale : An image scaling algorithm using an area pixel model," IEEE Trans. Circuits Syst.Video Technol., Vol. 13, No. 6, pp. 549–553, Jun. 2003.

[4] 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.

[5] J. H. Kim, J. W. Han, S. H. Cheon, J. O. Kim and S. J. Ko, "A novel image interpolation method using the bilateral filter," IEEE Trans. Consumer Electronics, Vol. 56, No. 1, pp. 175-181, Feb. 2010.

[6] Kuan-Hung Chen, Yuan-Sun Chu,"A Spurious-Power Suppression Technique for Multimedia/DSP Applications," IEEE Trans. Circuits and Sys. I, vol. 56, no. 1, pp.132-143, Jan.2009.

[7] Luming Liang,"Image Interpolation by Blending Kernels," IEEE Signal Processing Letters, Vol. 15, No. 1, pp. 805–808, Dec.2008.

[8] 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.

[9] Shih-Lun Chen," VLSI Implementation of a Low-Cost High- Quality Image Scaling Processor ," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60, No.1,pp.31-35, Jan. 2013.

[10] 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.

[11] 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. 2009.

Author's Profile:

Ms.M.Nikhitha has completed her B.Tech in ECE Department from RVR Institute of Technology, JNTU Hyderabad. Presently she is pursuing her Masters in VLSI System Design in MVSR Engineering College, Badangpet, Hyderabad, India.

Mr.KSR Murthy, Professor,ECE Dep, has completed MSc, MTech, MIETE Former Director, Computer, communi-cation and Range Systems, DRDL, Hyderabad. Currently working as Prof at MVSR Engineering college, Hyderabad, India.