ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Real Time Implementation of an Image Scaling Processor Using VLSI Technique

Fathima Abdul Azeez[*1], M AnandKumar[2]

[*1] PG Scholar,Department of Electronics and Communication Engineering, Maharaja Institute of Technology,Coimbatore,Tamilnadu, India

[2] Assistant Professor,Department of Electronics and Communication Engineering,, Maharaja Institute of Technology,Coimbatore,Tamilnadu, India

hifathy123@gmail.com

Abstract

Image scaling is a very important technique and has been widely used in many image processing applications. In applications where the scaling process must be performed at the display rather than at the CPU OR GPU, dedicated hardware implementation is necessary.Low-complexity image processing algorithms are necessary for VLSI implementation of real time applications. The image scaling algorithm of the proposed system consists of a sharpening spatial filter, a clamp filter, and a bilinear interpolation. Images are captured in real time by an image sensor and send to the FPGA along with the scaling parameter. Serial connectivity is provided to the FPGA and the scaled images are displayed on the PC. The filter combining, hardware sharing techniques of the bilinear interpolator and reconfigurable techniques has been used to reduce hardware costs.

Keywords: Sharpening filter, Very large scale integration (VLSI),Clamp filter, Image sensor, Bilinear interpolation, Reconfigurable calculation unit ,FPGA.

Introduction

Image scaling is the process of resizing a digital image. 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, The graphic and video applications of mobile handset require high quality pictures to be scaled down so as to fit in the display panel of the mobile phone and tablet PC. In the image scaling process, image quality should be preserved as much as possible. For real time applications, the scaling process is done at end user equipment. A dedicated hardware for image scaling averts performance degradation of CPU and/or graphics processing unit (GPU).This paper involves a real time implementation of the image scaling processor.

Related Work

S. L. Chen [2] proposed a scaling algorithm for the implementation of 2-D image scalar. The algorithm consists of a bilinear interpolation, a clamp filter, and a sharpening spatial filter.. An adaptive technology is used to enhance the effects of clamp and sharpening spatial filters. The clamp filter and sharpening filter are both combined into a 5 × 5 combined convolution filter. The edge oriented technique [3] is a seven stage VLSI architecture which adopts an edge catching technique.It makes use of an area pixel [4] image scaling algorithm. Shin L.Chen [5] proposed a low-complexity, low-memory-requirement, and high-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.

The Real Time System

The VLSI architecture of the real time image scaling processor consists of a camera interfacing module, an image scaling module, a controller and a UART module. Block diagram of the system is shown in Fig 1.Image sensors capture the images in real time and send it to the FPGA.A UART module is designed in the FPGA to provide serial connectivity to the PC to display the scaled images .

ISSN: 2277-9655
Impact Factor: 1.852

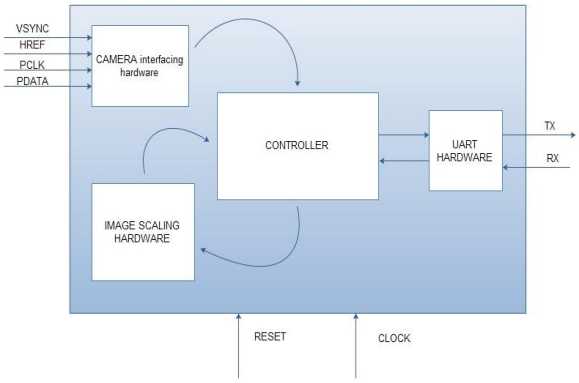

Fig 1: Real time image scaling processor

Camera Interfacing Module

The camera interfacing module on the FPGA on receiving a command starts capturing the pixels. The camera interfacing module waits for VSYNC from the image sensor which indicates the start of a frame, HREF which indicates start of a line , PCLK which is generated for every pixel .Data is transferred through the ports. The pixels are shifted into the register bank of the image scaling module .Fig 2 shows the timing of the various signals of the camera interfacing module .

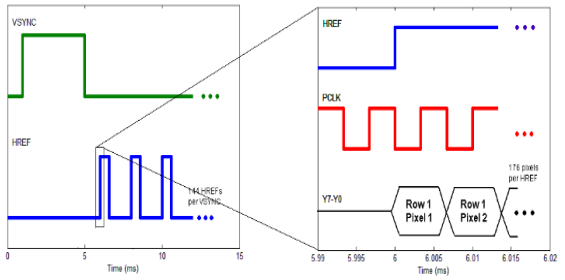

Fig 2: Timings of camera signals

Image Scaling Module

The image scaling hardware consists of a register bank with one line buffer, a combined filter and a bilinear interpolator. Block diagram of the scaling algorithm is shown in Fig 3.The combined filter consists of a sharpening spatial and clamp filters which serve as prefilters. The sharpening filter is a high pass filter to enhance the edges as well as to remove the noise. The kernel of the sharpening filter contains a single positive value at its center and completely surrounded by negative values. The clamp filter is a low pass filter used to smooth the unwanted edges and to reduce the aliasing effects. The kernel array contains a single positive value at its center surrounded by ones. The pixels filtered by both of the sharpening spatial and filters are passed to the bilinear interpolation for up-down scaling.

Fig 3: Block diagram of scaling algorithm

Combined Filter

The sharpening spatial and clamp filters are realized by the T-model and inverse T-model convolution kernels instead of 3×3 convolution kernels as shown in Fig 4. The sharpening filter and clamp filters is combined together into a combined filter as to decrease the demand of memory.

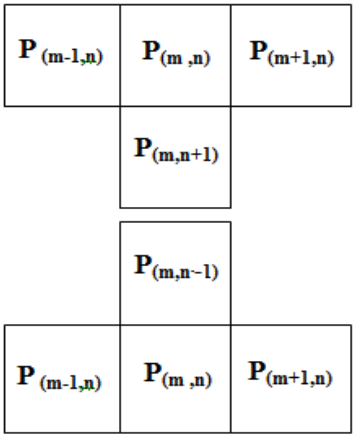

Fig 4: T-model and inversed T-model convolution kernels

$$P'_{(m,n)} = P^*_{(m,n)} \begin{bmatrix} -1 & S & -1 \\ & -1 & \end{bmatrix} / (S-3) * \begin{bmatrix} 1 & C & 1 \\ & 1 & \end{bmatrix} / (C+3) \quad (1)$$

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

$$\sim P^*_{(m,n)} \begin{bmatrix} -1 & S-C & SC-2 & S-C & -1 \\ & -S & S-C-1 & -S & \\ & & -1 & & \end{bmatrix} / [(S-3) \times (C+3)] \quad (2)$$

S and C are the sharp and clamp parameters and $P'_{(m,n)}$ is the filtered result the target pixel $P_{(m,n)}$ by the combined filter.

Bilinear Interpolation

Bilinear interpolation is a low complexity algorithm [1] which can be realized by VLSI technique. Bilinear interpolation is an operation that determines the intensity from the weighted average of the four closest pixels to the specified input coordinates, and then assigns that value to the output coordinates. It performs linear interpolation first in one direction, and then again in the other direction.

ISSN: 2277-9655
Impact Factor: 1.852

$$P_{(k,l)} = (1 - dx) \times (1 - dy) \times P_{(m,n)} + dx \times (1 - dy) \times P_{(m+1,n)} + (1 - dx) \times dy \times P_{(m,n+1)} + dx \times dy \times P_{(m+1,\, n+1)} \quad (3)$$

The dx and dy are zooming ratios of the horizontal and vertical directions , both of them which can be set by users. The bilinear interpolator can directly obtain two input pixels $P'_{(m,n)}$ and $P'_{(m,n+1)}$ from two combined prefilters without any additional line-buffer memory.

Register Bank

The register bank provides the ten source pixels for the combined filter to produce the target pixels of $P'_{(m,n)}$ and $P'_{(m,n+1)}$. The ten pixels are distributed in two lines of the original image and a one line memory buffer memory is enough to obtain the ten pixels from the two rows. Fig 5 shows the architecture of the register bank that consists of ten shift registers which connects with a one line buffer memory. When the shift command is issued from the controller, a new value of the source pixel is be read into Reg41 ,and each value stored in other registers belonging to row n+1 will be shifted right into the next register or line buffer memory. The Reg40 reads a new value of the pixel from the line buffer memory and each value in other registers belonging to row n will be shifted right into the next register.

The five pixels of row n is available in Reg00, Reg10, Reg20, Reg30, Reg40 and the five pixels of row n+1 is available in Reg01, Reg11, Reg21, Reg31 and Reg41.

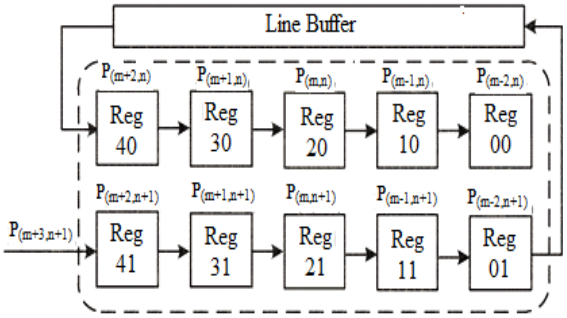

Fig 5: Architecture of register bank

Pipelined Architecture

The combined filter and bilinear interpolator consists of a six stage pipelined architecture as shown in Fig 6. The stages 1 and 2 shows the computational scheduling of a T-model and inverse T-model filter and stages 3, 4, 5 and 6 of that bilinear interpolation. The combined T and inverse T model filters provides two processed pixels $P'_{(m,n)}$ and $P'_{(m,n+1)}$ simultaneously to the next stage. The symmetrical circuit shown in stages 1 and 2 is the inversed T-model combined filter for producing the filtered result of $P'_{(m,n+1)}$. The combined filter consists of three reconfigurable calculation units (RCUs), one multiplier-adder (MA), three adders(+),three subtractors (-),and three shifters(S).

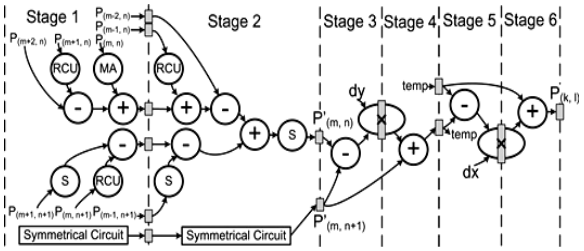

Fig 6: Computational scheduling of the combined filter and bilinear interpolator

Reconfigurable Calculation Unit

The reconfigurable calculation unit is designed to avoid performing multiplications in order to reduce the computing resources which will consume more silicon area. The RCU is designed for producing the calculation functions of (S-C) and (S-C-1) times of the source value which must be implemented with C and S parameters. Fig 7 shows the architecture of the RCU. It consists of four shifters, three multiplexers, three adders and one sign circuit. The multiplexers are turned on according to the value of (S-C) and (S-C-1). Table 1 lists the parameters and computing resource for the RCU. The C and S parameters can be set by users according to the characteristics of the images.

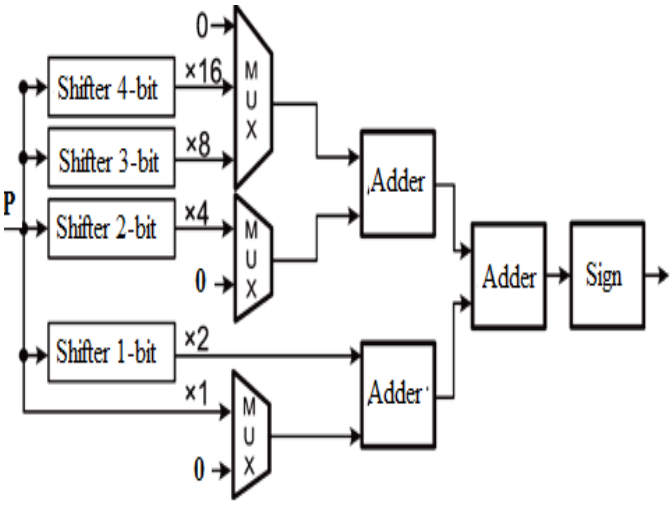

Fig 7: Architecture of RCU

ISSN: 2277-9655
Impact Factor: 1.852

TABLE 1:Parameters and Computing Resource for RCU

Parameters	Values	Computing
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

The output of the image scaling module are the interpolated values which are to be mixed with the original image in order to obtain the scaled image. Fig 8 shows simulation results of the image scaling module.

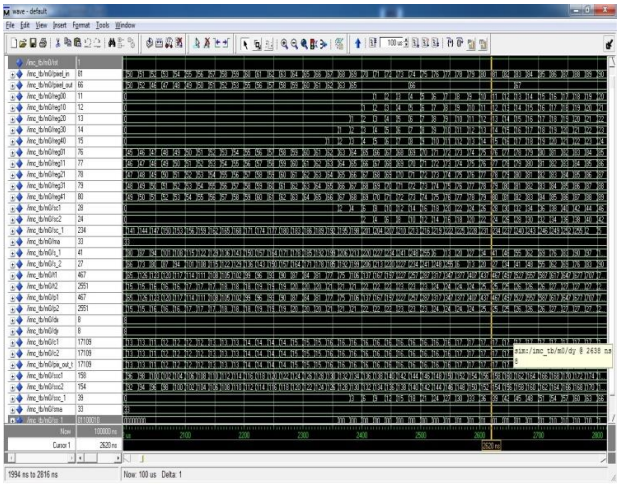

Fig 8: Output of the Image Scaling Module

Controller

The controller is implemented by a finite state machine circuit. It produces control signals and the timing signals for the camera module, pipeline stages of the register bank, combined filter, bilinear interpolator and the UART.

Results and Discussions

The VLSI architecture of the system was designed using VHDL. The system was implemented using ALTERA FPGA Cyclone II, EP2C70F896C6 core. The interpolated values from the image scaling module is sent to the PC and is mixed with the original image in MATLAB to get the scaled images. The real time image scaling processor has been synthesized successfully.

- Total registers -281
- Total pins-23
- Total memory bits - 406784/1152000 (35%)
- Total combinational functions – 587 / 68416 (<1%)
- Dedicated logic registers - 281 / 68416 (<1%)

Conclusion

This work provides a low cost, high quality VLSI architecture for real time image scaling applications. The filter combining, hardware sharing, and reconfigurable techniques had been used to reduce hardware cost. A dedicated hardware can reduce the computing overhead of CPU or GPU. Memory requirement is reduced as it requires only one line buffer. Applications of the image scaling processor includes display side scaling ,remote desktop, screen sharing, sophiscated image rendering applications, sophiscated image editing applications, graphics and video applications of mobile handset devices .

Acknowledgement

I would like to thank Mr. M AnandKumar, Assistant Professor, Maharaja Institute of Technology ,Coimbatore for his support and guidance. I am also thankful to Mr. P Suresh Kumar HOD, ECE Department of Maharaja Institute of Technology, Coimbatore.

References

[1] Andreadis I. and A. Amanatiadis,(1978) Digital image scaling in Proc.IEEE Instrum. Meas. Technol. Conf., May 2005, vol. 3, pp. 2028–2032.

[2] Chen S., H.Huang, and C. H. Luo, (2011), 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.

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

[4] Kim C.H,S. M.Seong, J. A. Lee, and L. S. Kim, (2003) Winscale : An image- scaling algorithm using an area pixel model, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 6, pp. 549–553

[5] Shin L.Chen,(2013),VLSI Implementation of a low cost high quality Image Scaling Processor, IEEE Trans. Circuits & Syst., vol. 60, no.1, pp. 31 - 35