Design of Graphic Processing Unit in Mobiles

J. George Cherian Panappally
Department of Electronics and Communication
Rajagiri School of Engineering and Technology
Cochin, India
Email: jerrygcp@gmail.com

Dhanesh M.S
Department of Electronics and Communication
Rajagiri School of Engineering and Technology
Cochin, India
Email: dhaneshmuralee@gmail.com

Abstract—**This work describes about the designing of a Graphic Processing unit (GPU) that deals with image processing. Graphics processing unit (GPU) has become a standard in high performance computing which is involved in image processing.The design is done using Verilog hardware description language (HDL)which provides an immediate hardware implementationpossibility.GPU designed for image processing on a dedicated hardware is faster and efficient compared to software implemented image processing. The paper focuses on basic image processing techniques such as image zooming,image cropping, brightness manipulation and image rotation.**

Keywords—Graphic processing unit;Image Processing;Verilog HDL;FPGA

I. INTRODUCTION

In recent years, Graphic Processing Unit (GPU) has become more popular as the demand for graphic applications increased. Image Processing requires large computation as it deals with large amount of data pixels. In such cases using dedicated processor for computation purposes dealing with similar image processing applications makes system efficient. While Central Processing Unit (CPU) does different types of tasks, GPU does similar and repeated tasks.

GPU is now commonly used, when dealing with large computation. GPU is used for media applications in mobiles, computers, tablets etc.GPU is designed as a Single Instruction Multiple Data (SIMD) Model. It can execute single instruction on multiple pieces of data pixels with multiple processing elements. Processing Elements are the important modules in the GPU that can perform instructions on different pieces of data. So the GPU can have many number of processing elements each having set of registers and Arithmetic and Processing units (ALU).

Verilog Hardware Description Language (HDL) allows the designer to logically describe the circuit. It also allows to simulate and evaluate the output. The main advantage of using Verilog HDL is that it allows the designer to directly implement the design on to the hardware which improves the speed of execution. Verilog HDL has emerged as the standard Hardware Description Language. The syntax of Verilog HDL is similar to C language which makes it much easier to learn.
So it is more commonly used in industry.Another Hardware Description Language (HDL) that is used is Very High Speed Integrated Circuits HDL (VHDL)[1].

II. PROCESSOR DESIGN

We designed a 64-bit Processor with four stage pipelined execution. This processor could perform basic operations that include add, subtract, multiplication etc. These operations are used for image processing operations such as brightness manipulation, image cropping, image zooming and image rotation. Hardware Organisation of the GPU includes Memory, Program Counter (PC), Processor, Processing Element (P.E) and Local Data Share [2].

The Program Counter points the current instruction to be executed. The memory contains the instructions and the data to be processed. Each instruction contains an opcode and operand. The instructions are fetched, decoded and depending upon the opcode of the instructions, the ALU in the processing elements process the different pieces of data [3]. The common data share shares the data between each other. Then the data is stored back to Data Memory. Figure 1 shows the hardware organisation of the GPU.

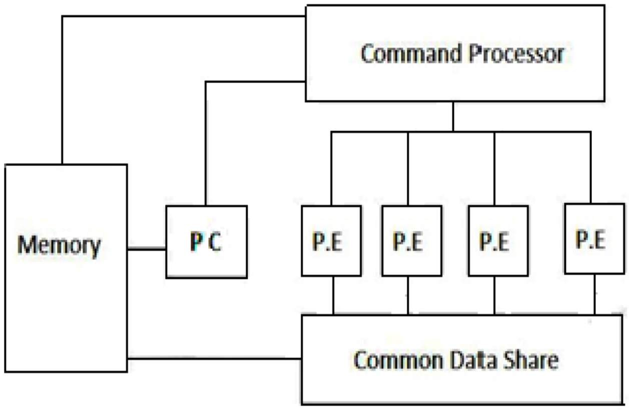

Fig.1.Hardware Organization

The processor design for the GPU is as shown in Fig.2.There is separate instruction and data memory. Program Counter is incremented after the current instruction is fetched. The instructions are decoded. Depending upon the control signals the registers are activated [2]. Multiplexer selects the data in the accumulator depend on the control signals. After the execution of the current instruction the data in the accumulator is stored in the data memory and the next instruction is fetched [3].

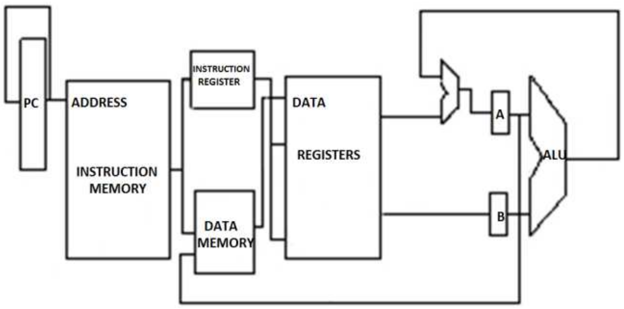

Fig.2.Processor Design

III. IMAGE PROCESSING TECHNIQUE

In image processing a 2D image is processed. In computers and mobiles, image is displayed using RGB color model. RGB image is the superimposing of three colors, red, green and blue. An image contains thousands of pixels. Each pixel contains an amount of red, green and blue color. Image processing uses algorithms to perform operations on the data pixels of the image. The most commonly used operations on images are image zooming, image cropping, image rotation and brightness manipulation. Image processing operations are done on spatial domain where the processing is directly done on data pixels of the image [4].

IV. IMPLEMENTATION OF IMAGE PROCESSING TECHNIQUE

We use Verilog HDL for describing the functionality of the circuit. Verilog HDL does not support bitmap or jpeg images. So, here we should directly transfer the pixel values to the processor for processing the image. The pixel values of the image obtained from the Matlab is transferred to the designed processor through serial Universal Asynchronous Receiver/ Transmitter (UART). After processing the data pixels, they are transmitted back through serial UART. The implementations of image processing are as explained below

A. Brightness Manipulation

In order to change the brightness of the image a constant threshold value is added or subtracted from the original pixel data of the image. This is an image enhancement technique. This technique improves the visibility of the image [4].

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} + \begin{pmatrix} 3 \end{pmatrix} = \begin{pmatrix} 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 \end{pmatrix}$$

Fig.3.Matrix Model of Image Enhancement

Fig.3 shows the matrix model of the brightness manipulation. For brightness manipulation, each pixel value needs to added or subtracted with a threshold value.For adding a threshold value if the pixel value added with threshold value is lesser than 255 then it is accepted else the pixel value is set to 255. Similarly to subtract a threshold value, if the pixel value subtracted with the threshold value is greater than zero it is accepted else the pixel value is set to zero

B. Image Cropping

Cropping an image removes the unwanted or irrelevant outer parts of the image. This can change the aspect ratio of the image. Image cropping can also be used to make the image fit into the frame. In image cropping the intended row or column can be multiplied with zero value. These causes the values in that row or column to be 0.These row or column are then removed from the data pixel. Fig.4 shows the matrix model of image cropping.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 3 & 4 \\ 0 & 6 & 7 & 8 \\ 0 & 9 & 10 & 11 \\ 0 & 14 & 15 & 16 \end{pmatrix}$$

Fig.4.Matrix Model of Image Cropping

C. Image Zooming

Image Zooming is commonly used in mobiles, computers, etc. to obtain the detailed view of the image. By zooming, we can focus on a particular part of the image. If we zoom the image by 50% then what we actually does is magnifying the image 1.5 times. In image zooming we need double the size of the array compared to the original matrix array. Here we use data duplication method where we copy the same value of the data pixel to the neighboring positions. Consider a 3*3 matrix, then to zoom the image we double the new matrix to a 9*9 matrix. Then each value is copied to 3 positions along the row and column. Fig.5 shows the matrix model of image zooming.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{pmatrix}$$

Fig.5. Matrix Model of Image Zooming

D. Image Rotation

Image Rotation is another feature that is used in media applications. In image rotation, the image can be rotated clock-wise or anticlock-wise. Image rotation uses special algorithm that can be used to rotate the image.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \Rightarrow \begin{pmatrix} 4 & 8 & 12 & 16 \\ 3 & 7 & 11 & 15 \\ 2 & 6 & 10 & 14 \\ 1 & 5 & 9 & 13 \end{pmatrix}$$

Fig.6.Matrix Model of Image Rotation

V. RESULTS

The operations where performed on image using Verilog HDL. The operations include brightness manipulation,image zooming, image rotation and image cropping. The processor, designed for image processing was implemented on Spartan3FPGA (XC3S500E-4FG320C).The results of the operations are as shown below.

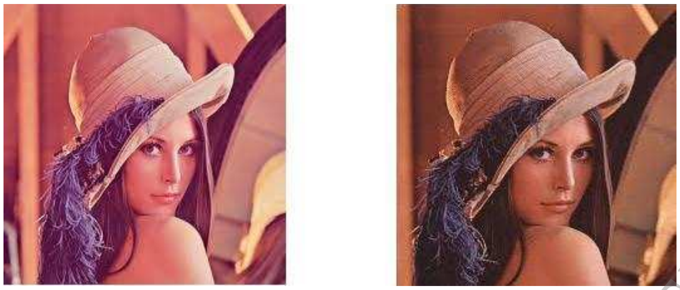

Fig.7. Results for Brightness Manipulation

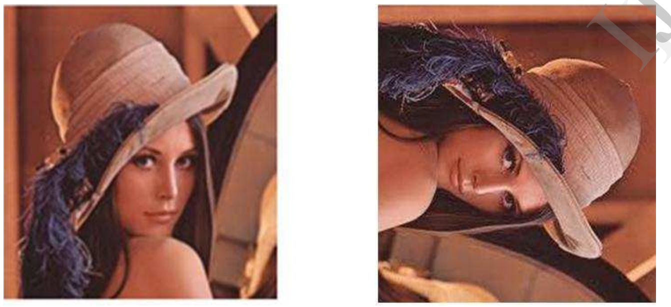

Fig.8.Results for Image Rotation

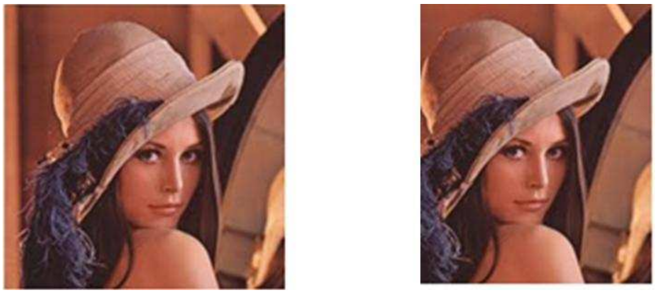

Fig.9.Results for Image Cropping

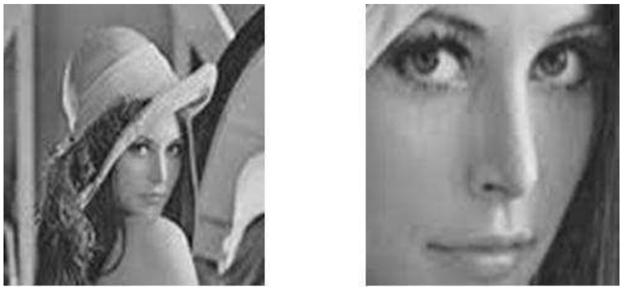

Fig.10. Results of Image Zooming

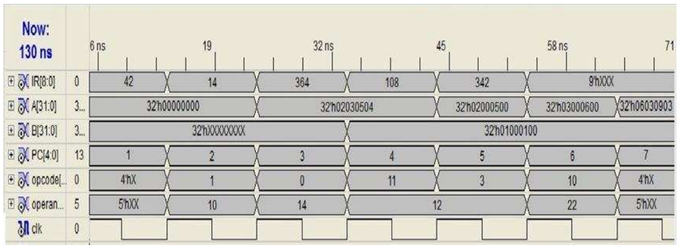

Fig.11. Simulation Results

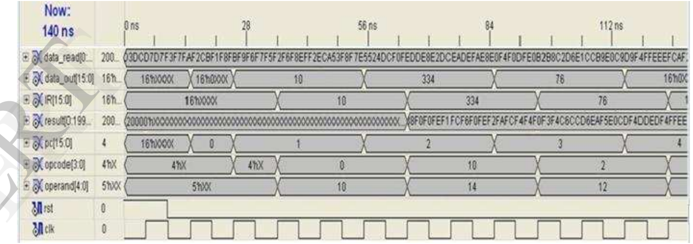
Now:
140 ns
0 ns
28
56 ns
84
112 ns
data_read[0:...
200...
03DCD7D7F3F7FAF2CBF1F8FBF9F6F7F5F2F6F8EFF2ECA53F8F7E5524DCF0FEDDE8E2DCEADEFAE8E0F4F0DFE0B2B8C2D6E1CCB9E0C9D9F4FFEEEFCAF
data_out[15:0]
16'h...
16'hXXXX
16'h0XXX
10
334
76
16'h0X
IR[15:0]
16'h...
16'hXXXX
10
334
76
result[0:199...
200...
20000'hXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8F0F0FEF1FCF6F0FEF2FAFCF4F4F0F3F4C6CCD6EAF5E0CDF4DDEDF4FFEE
pc[15:0]
4
16'hXXXX
0
1
2
3
4
opcode[3:0]
4'hX
4'hX
4'hX
0
10
2
operand[4:0]
5'hXX
5'hXX
10
14
12
rst
0
clk
0

Fig.12. Simulation Results

VI.CONCLUSION

In this paper, Graphic Processing Unit was designed to perform image processing operations. These operations are commonly used in mobiles and computers. Theuse of HDLs in image processing enables an immediate hardware implementation possibility. By using a separate hardware for image processing,the performance in processing was improved. The designed GPU can also be modified to perform different other image operations. Thus we can utilize thisGPU for any operations dealing with images and videos.

ACKNOWLEDGMENT

The Authors wish to thank Mr. Anoop Thomas, Asst. Professor, Mr. Jaison Jacob, HOD, Electronics and Communication of Rajagiri School of Engineering and Technology for their valuable suggestions and contributions.

REFERENCES

[1] N. Thammasan, and P. Chongstitvatana, "Design of a GPU-styled Softcore on Field Programmable Gate Array," Int. Joint Conf. on Computer Science and Software Engineering (JCSSE), 30 May - 1 June 2012, pp. 142-146.

[2] Chongstitvatana, P., "Putting General Purpose into a GPU-style Softcore," Int. Conf. on Embedded Systems and Intelligent Technology, Jan 13-15, 2013, Thailand.

[3] Bin Zhang, Kuizhi Mei and Nanning Zheng, "Reconfigurable Processor for BinaryImage Processing,"IEEE Transactions on Circuits and Systems for video Technology, Vol. 23, no. 5, May 2013.

[4] Wang, A. Sun, Y. Li, H. Liu, “Programmable GPUs: New General Computing Resources Available for Desktop Grids,” Int. Conf. Grid and Cooperative Computing, Oct. 2006, pp. 46-49.

[5] Luliana Chiuchisan, Marius Cerlinca, Alin-Dan Potorac, Adrian Graur,“Image Enhancement Methods Approach using Verilog hardware Description Language,” 11th International Conference on Development And Application Systems, Suceava, Romania, May 17-19, 2012.