Implementation of 2D Non-linear Morphological Image Processing on FPGA Based Architecture

K.Sambashivudu, Md.Javeed, R.Kiran

Abstract- *Image processing requires high computational power and the ability to experiment with algorithms. Recently, reconfigurable hardware devices in the form of field programmable gate arrays (FPGAs) have been proposed as a way of obtaining high performance at an economical price. FPGA technology has become a viable target for the implementation of real time algorithms suited to video image processing applications. Morphing is a Technique used to transfer from one image to another. However, most morphological tools such MATLAB are not suited for strong real-time constraints. The unique architecture of the FPGA has allowed the technology to be used in many applications encompassing all aspects of video image processing. Among those algorithms, linear filtering based on a 2D convolution, and non - linear 2D morphological filters, represent a basic set of image operations for a number of applications. This paper reports on the design and realization of an FPGA based image processing for implementation of morphological image filtering using a FPGA NexysII, Xilinx Spartan 3E, with educational purposes. The system is connected to a USB port of a personal computer, which in that way form a powerful and low-cost design. The FPGA technologies offer basic digital blocks with flexible interconnections to achieve high speed digital hardware realization. The FPGA consists of a system of logic blocks, such as look up tables, gates, or flip-flops and some amount of memory. The image will be transferred from PC to FPGA board using UART serial communication/JTAG cable. After performing the required filtering/processing the result will be transferred back to computer. In PC both the results will be validated. A comparison between results obtained from MATLAB simulations and the described FPGA-based implementation is presented.*

Keywords: *Morphology, Image processing algorithms, Field Programmable Gate Array (FPGA), filtering, Simulation.*

I. INTRODUCTION

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image enhancement and object detection. Image processing algorithms are conventionally implemented in DSP processors and some special purpose processors. In recent days ARM processors are being used for implementing Image processing algorithms. However all these implementation styles are limited by throughput which becomes very critical parameter for several image processing applications.

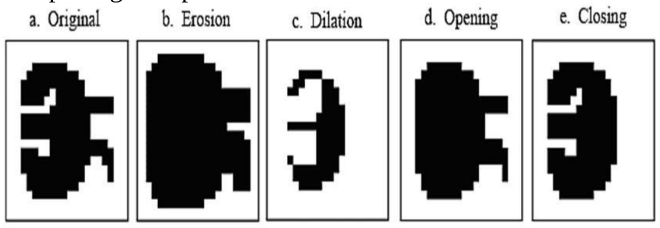
Manuscript published on 30 November 2013.
*Correspondence Author(s)
K.Sambashivudu, M.Tech(VLSI SYSTEM DESIGN), Ganapathy Engineering College, Warangal, India.
MD. Javeed, M. Tech (VLSI SYSTEM DESIGN), Ganapathy Engineering College, Warangal,India.
R.Kiran, Assistant Professor, Department of ECE, Ganapathy Engineering College, Warangal, India.

Field Programmable Gate Arrays (FPGAs) are part of Current reconfigurable computing technology, which in some ways represent an ideal alternative for image and video processing. FPGAs generally consist of a system of logic blocks, such as look up tables, gates, or flip-flops, just to mention a few, and some amount of memory, all wired together using a vast array of interconnects. All of the logic in an FPGA can be rewired, or reconfigured, with a different design, according to the designer needs. The FPGA technologies offer basic digital blocks with flexible interconnections to achieve high speed digital hardware realization. In video and picture processing requirements for security and multimedia applications we require different linear filtering algorithms based on a 2D convolution and non-linear 2D morphological filters. In this research such filters will beimplemented in FPGA platform using VHDL. The image will be transferred from computer to FPGA board using UART serial communication/JTAG cable. After performing the required filtering/processing the result will be transferred back to PC. In PC both the results will be validated. The HDL implementation uses basic blocks registers, adder s, multipliers, control logic, UART transmitter and receiver etc. The project aims to setup an image processing platform on FPGA hardware. The results shall demonstrate the performed operation on FPGA and validate the implemented architecture.

II. MORPHOLOGICAL OPERATIONS

Morphology is a theory and technique for the analysis and processing of geometrical structures, based on set theory and random functions. Morphology deals with shapes of images in Image Processing. The value of the each output pixel is based on a comparison of the corresponding pixel In the input image with its neighborhood pixels. The size and shape of the structural element determines the morphological operation to be done on a particular image. Morphological operations are usually applied to the processing of binary and Grayscale images. Morphological operators are defined as combinations of basic numerical operations taking place over an image A and a small object B, called a structuring element.B can be seen as a probe that scans the image and modifies it according to some specified rule. The shape and size of B, which is typically much smaller than image A, in conjunction with the specified rule, define the characteristics of the performed process. Binary mathematical morphology is based on two basic operators: Dilation, and erosion. Both are defined in terms of the interaction of the original image A to be processed, and the structuring element B. From these two basic operations; other operations like opening and closing of an image are developed. The principle of Dilation operation is the value of the output pixel is the maximum value of all the pixels in the input pixels neighborhood.

In Erosion, the value of the output pixel is the minimum value of all pixels in their input pixel's neighborhood. Opening means erosion followed by dilation operation and closing means dilation followed by erosion operation. The Morphological operations are:



i) Erosion:

Erosion uses the following mask:

1 1 1
1 1 1
1 1 1

This means that every pixel in the neighborhood must be 1 for the output pixel to be 1. Otherwise, the pixel will become 0. No matter what value the neighboring pixels have, if the central pixel is 0 the output pixel is 0. Just a single 0 pixel anywhere within the neighborhood will cause the output pixel to become 0. Erosion can be used to eliminate unwanted white noise pixels from an otherwise black area. The only condition in which a white pixel will remain white in the output image is if all of its neighbors are white. The effect on a binary image is to diminish, or erode, the edges of a white area of pixels. The data from UART receiver is given to module which performs erosion. Pixel information as in gray scale coded format is taken as input. Pixel information must be stored into rows for this row_sel will take care of it. Row contains data banks which is equal to the dimensions (only width) of the image. Row_sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes for row 3 after completing the row 3 it go for the row 1 and the process will continue. Row_sel will select row simultaneously till the image is complete. The data in row 1 columns 1,2 and 3 pixel which ever pixel is minimum will be selected, in the same manner row 2 and row 3 minimum pixel value is selected from these three minimum is given to min4,min4 resultant will be minimum of all three minimum of rows, that resultant is taken as output.

ii) Dilation:

Dilation is the opposite of erosion. Its mask is:

0 0 0
0 0 0
0 0 0

This mask will make white areas grow, or dilate. The same rules that applied to erosion conditions apply to dilation, but the logic is inverted - use the NAND rather than the AND logical operation. Being the opposite of erosion, dilation will allow a black pixel to remain black only if all of its neighbors are black. This operator is useful for removing isolated black pixels from an image. The data from UART receiver is given to module which performs Dilation. Pixel information as in gray scalecoded format. Pixel information must be stored into rows for this row_sel will take care of it. Row contains Data banks which is equal to the dimensions (only width) of the image. Row_sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes to row3 after completing the row 3 it go for the row 1 and the process will continue. Row_sel will select row simultaneously till the image is complete. The data in row 1 columns 1,2 and 3 pixel which pixel is maximum will be selected in the same manner row 2 and row 3 maximum pixel value is selected from these three maximum is given to max4 in max4 resultant will be maximum of all three maximum of rows, that resultant is taken as output. This is forced on to the UART transmitter section.

iii) Opening:

The data from UART receiver is given to module which performs erosion. Pixel information as in gray scale coded format is taken as input. Pixel information must be stored into rows for this row_sel will take care of it. Row contains data banks which is equal to the dimensions (only width) of the image. Row_sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes to row 3 after completing the row 3 it goes to the row 1 and the process will continue. Row_sel will select row simultaneously till the image is complete. The data in row 1 columns 1,2 and 3 pixel which ever pixel is minimum will be selected, in the same manner row 2 and row 3 minimum pixel value is selected from these three minimum is given to min4, min4 resultant will be minimum of all three minimum of rows, that resultant pixel information must be stored into rows for this row_sel will take care of it in dilation process. Row _sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes for row 3 after completing the row 3 it goes to the row 1 and the process will continue. Row _sel will select row simultaneously till the image is complete. The data in row 1 columns 1,2 and 3 pixel which ever pixel is maximum will be selected in the same manner row 2 and row 3 maxim pixel value is selected from these three maximum is given to max4,max 4 resultant will be maximum of all three maximum of rows, that resultant is taken as output. This is forced on to the UART transmitter section.

iv) Closing:

The data from UART receiver is given to Module which performs Dilation. Pixel information as in gray scale coded format. Pixel information must be stored into rows for this row_sel will take care of it. Row contains Data banks which is equal to the dimensions (only width) of the image. Row_sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes to row3 after completing the row 3 it goes to the row 1 and the process will continue. Row _sel will select row simultaneously till the image is complete. The data in row 1columns 1,2 and 3 pixel which ever pixel is maximum will be selected in the same manner row 2 and row 3 maximum pixel value is selected from these three maximum is given to max4 in max 4 resultant will be maximum of all three maximum of rows, that resultant is given to erosion block. Row_sel first select first row and stores information (pixel value) in it after first row is full i.e. it is end of the image of first line then the row_sel select row 2 for filling of the data if row 2 is full then goes to row3 after completing the row 3 it goes to the row 1 and the process will continue. Row _sel will select row simultaneously till the image is complete. The data in row 1 columns 1, 2 and 3 pixel which ever Pixel is minimum will be selected, in the same manner row 2 and row 3 minimum pixel value is selected from these three minimum is given to min4, min4 resultant will be minimum of all three minimum of rows, that resultant is taken as output. This is forced on to the UART transmitter section.

III. DESIGN METHODOLOGIES OF VLSI

VHDL (VHSIC hardware description language) is a high level description language for system and circuit design. The language supports various levels of abstraction. In contrast to regular netlist formats that supports only structural description and a Boolean entry system that supports only dataflow behavior, VHDL supports a wide range of description styles. These include structural descriptions, data flow description and behavioral descriptions. The structural and dataflow descriptions show a concurrent behavior. That is, all statements are executed concurrently, and the order of the statements is not relevant. On the other hand, behavioral descriptions are executed sequentially in processes, procedures and functions in VHDL. The behavioral descriptions resemble high-level programming languages. VHDL allows a mixture of various levels of design entry abstraction. Precision RTL Synthesis Synthesizes will accept all levels of abstraction, and minimize the amount of logic needed, resulting in a final netlist description in the technology of your choice. The Top-Down Design Flow is shown in Figure

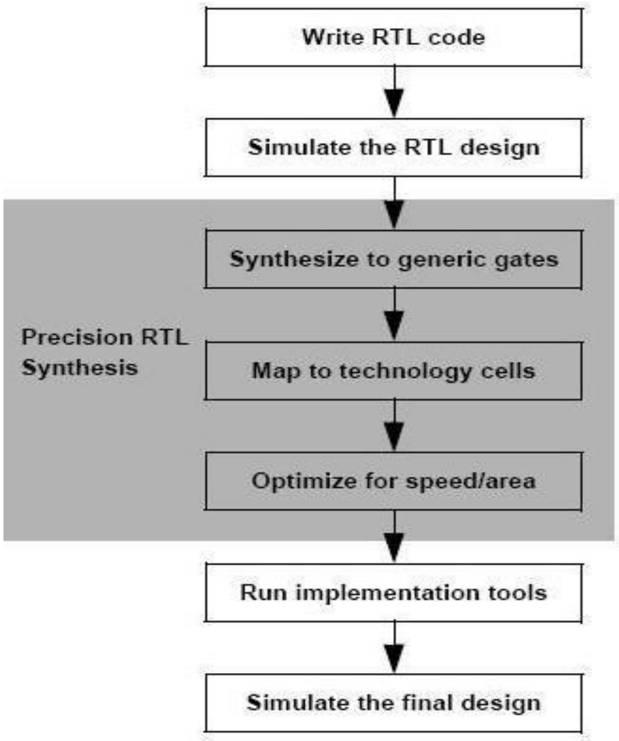


Figure: Top-Down Design Flow

VHDL is a fairly general-purpose language, and it doesn't require a simulator on which to run the code. There are many VHDL compilers, which build executable binaries. It can read and write files on the host computer, so a VHDL program can be written that generates another VHDL program to be incorporated in the design being developed. Because of this general-purpose nature, it is possible to use VHDL to write a test bench that verifies the functionality of the design using files on the host computer to define stimuli, interacts with the user, and compares results with those expected. It is relatively easy for an inexperienced developer to produce code that simulates successfully but that cannot be synthesized into a real device, or is too large to be practical. VHDL is not a case sensitive language. One can design hardware in a VHDL IDE (for FPGA implementation such as Xilinx ISE, Altera Quartus, or Synopsys Synplify) to produce to generate an appropriate test bench for a particular circuit or VHDL code, the inputs have to be defined correctly. For e.g., for clock input, a loop process or an iterative statement is required. The key advantage of VHDL when used for systems design is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires). Another benefit is that VHDL allows the description of a concurrent system (many parts, each with its own sub-behavior, working together at the same time). VHDL is a Dataflow language, unlike procedural computing languages such as BASIC, C, and assembly code, which all run sequentially, one instruction at a time. A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device such as a CPLD or FPGA, and then it is the actual hardware being configured, rather than the VHDL code being "executed" as if on some form of a processor chip. VHDL is frequently used for two different goals: simulation of electronic designs and synthesis of such designs. Synthesis is a process where a VHDL is compiled and mapped into an implementation the RTL schematic of the desired circuit. After that, the generated schematic can be verified using simulation software which shows the waveforms of inputs and outputs of the circuit after generating the appropriate test bench. Technology such as an FPGA or an ASIC. Many FPGA vendors have free (or inexpensive) tools to synthesize VHDL for use with their chips, where ASIC tools are often very expensive. Not all constructs in VHDL are suitable for synthesis. For example, most constructs that explicitly deal with timing such as wait for 10 ns; are not synthesizable despite being valid for simulation. While different synthesis tools have different capabilities, there exists a common synthesizable subset of VHDL that defines what language constructs and idioms map into common hardware for many synthesis tools. A large subset of VHDL cannot be translated into hardware. This subset is known as the non- synthesizable or the simulation-only subset of VHDL and can only be used for prototyping, simulation and debugging. For example, the following code will generate a clock with the frequency of 50 MHz it can, be used to drive a clock input in a design during simulation. It is, however, a simulation only construct and cannot be implemented in hardware.

IV. SIMULATION RESULTS

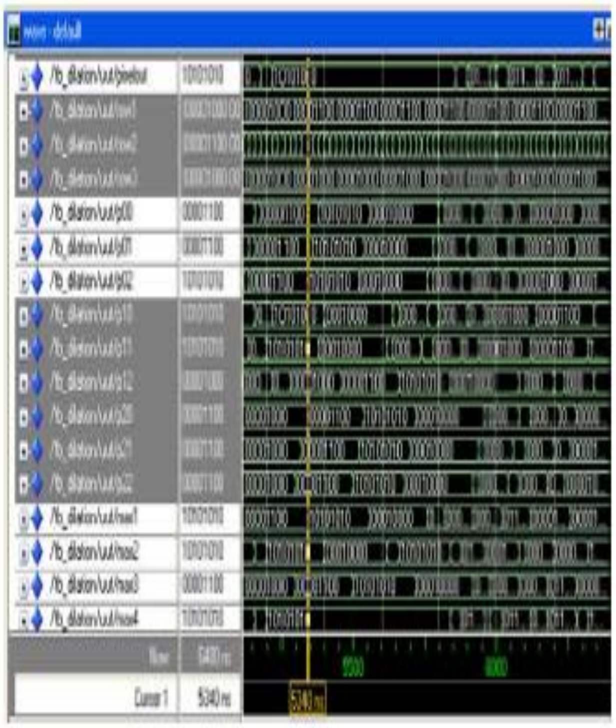
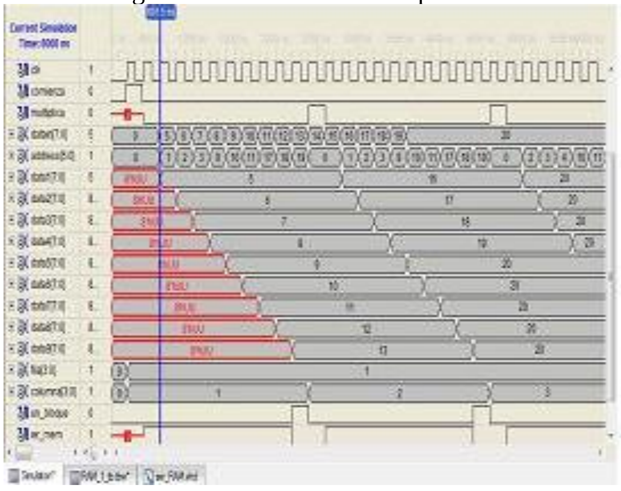


Fig: Simulation results for top module



V. CONCLUSION

The proposed method is inherently parallel, since computations for each pixel of each sequence frame can be done concurrently with no need for communications. This can help in lowering execution times for high-resolution sequences. Moreover, the approach is suitable to be adopted in a layered framework, where, operating at region-level, it can improve detection results allowing to more efficiently tackle the camouflage problem and to distinguish morphological Image by the morphological operator. A low-cost image processing system for real time applications with educational purposes has been presented. The system takes advantages of the available resources in a Nexis II system based on the Xilinx FPGA Spartan 3E.The described FPGA-based real-time image processing system was shown to provide a very good tool for further computer vision applications. At the same time, it is worth-while to mention the educational value of the developed prototype as a laboratory tool in modern digital system courses. It combines hardware and software to achieve accurate as well as a considerably high performance, which can be accounted to the efficient parallel implementation so that the speed is increased. It is well known that there are many different learning styles. Some people learn better by reading books, others through a verbal explanation, while others learn most effectively through application. The goal of this project is to add another tool to the learning style, one focused on a visual learning style. By developing an application to demonstrate some tools of morphological image processing.

REFERENCES

1. K. T. Gribbon, D. G. Bailey and C. T. Johnston, "Design Patterns for Image Processing Algorithm Development on FPGAs", TENCON 2005, pp. 1-6, November 21-24, 2005.
2. A. Castillo, J. Vázquez, J. Ortegón y C. Rodriguez,"Prácticas de laboratorio Para estudiantes de ingeniería con FPGA", IEEE Latin America Transactions, Vol. 6, No.2, pp. 130-136, 2008.
3. Bruce A. Draper, J. Ross Beveridge, A.P. Willem Böhm, Charles Ross, Monica Chaw the, "Accelerated Image Processing on FPGAs", IEEE Transactions on Image Processing, Vol. 12, No. 12. Pp. 1543-1551, 2003.
4. D.G. Bariamis, D.K. Iakovidis, D.E. Maroulis, S. A. Karkanis, "An FPGA-based Architecture for Real Time Image Feature Extraction", Proceedings of the 17th International Conference on Pattern Recognition, August 23-26, Cambridge, UK, 2004.
5. C.T. Johnston, K.T.Gribbon, D.G.Bailey, "Implementing Image Processing Algorithms on FPGAs", Eleventh Electronics New Zealand Conference, Palmerston North, New Zealand, 2004
6. Bob L. Sturm and Jerry D. Gibson, "Signals and Systems Using MATLAB: An Integrated Suite of Applications for Exploring and Teaching Media Signal Processing", 35th ASEE/IEEE Frontiers in Education Conference, pp. 21-25, October 19 – 22, Indianapolis, Indiana, USA, 2005.
7. Javier Vicente, Begoña García, Ibon Ruiz, Amaia Méndez, Oscar Lage,"EasySP: Nueva Aplicacin Para la Enseñanza de Procesado de Se al",IEEE-RITA, Vol. 2,No.1, 2007.
8. A. Ledda and W. Phillips,"Majority Ordering and the Morphological Pattern Spectrum, Conf. Proc. Advanced Concepts for Intelligent Vision Systems, Antwerp, Belgium, 356-363, 2005.
9. Malay Haldar, Anshuman Nayak, Alok Choudhary, Banerjee,A system for synthesizing optimized FPGA hardware from MATLAB",International Conference on Computer Aided Design, pp. 314-319, San Jose, California, 2001.
10. Rafael C. Gonzales, Richard E. Woods, Digital Image Processing using Mat lab, Prentice Hall,2002.
11. Tanvir A. Abbasi, Mohd U. Abbasi,"A Proposed FPGA Based Architecture for Sobel Edge Detection Operator", J. of Active and PassiveElectronic Devices,Vol. 2, pp. 271–277, 2007.
12. David Báez-López, David Báez-Villegas, RenéAlcántara, Juan José Romero, Tomás Escalante, "A package for filter design based on MATLAB", Computer Applications in Engineering Education, Vol. 9, No. 4,pp. 259-264, 2002.
13. R.E. Haralick, K. Shanmugam, I. Dinstein, Textural Features for Image Classification, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, No.6, Nov 1973
14. J.L. Hennesy & D.A. Patterson, Computer Architecture, A Quantitative Approach, Morgan Kaufmann, May 2002
15. D.E. Maroulis, D.K. Iakovidis, S.A. Karkanis, D.A.Karras, CoLD: a versatile detection system for colorectal lesions in endoscopy video frames,Computer Methods and Programs in Biomedicine, vol 70, no. 2,pp. 99-186, Elsevier Science, 2003
16. W. Luk, P. Andreou, A. Derbyshire, F. Dupont-DeDinechin, J. Rice, N. Shirazi and D. Siganos, Field-Programmable Logic: From FPGAs to ComputingParadigm, Springer-Verlag, Berlin, 1998.
17. M. Nibouche, A. Bouridane, D. Crookes, O.Nibouche,An FPGA-based wavelet transforms coprocessor, in Proc. IEEE Int. Conf. Image Processing, pp.194-197, vol.3, 2003.
18. Soohwan Ong and Myung H. Sunwoo, A Morphological Filter Chip Using a Modified Decoding Function, IEEE Transactions on circuit and systems-II: Analog and digital signal processing, vol. 47, no. 9, September 2000.

AUTHOR PROFILE

SAMBASHIVUDU KORUTLA received the M.Tech in VLSI SYSTEM DESIGN from Ganapathy Engineering college, Warangal. His area of interest is VLSI SYSTEM DESIGN.

MOHAMMED JAVEED received the M.Tech in VLSI SYSTEM DESIGN from Ganapathy Engineering college, Warangal. His area of interest is VLSI.

KIRAN RENUKUNTLA is working as an Assistant Professor in the Department of ECE, Ganapathy Engineering College, Warangal. His area of interest is VLSI.