Implementation of Digital Image Morphological Algorithm on FPGA using Hardware Description Languages

R. Arunmozhi
Assistant Professor
Department of Electrical Engineering
Annamalai University

G. Mohan, PhD.
Professor
Department of Electrical Engineering
Annamalai University

ABSTRACT

In this work, an implementation of linear filtering and morphological image operation using a EDK 11.1 FPGA Spartan 3E is implemented in Field Programmable Gate Array this technology has become a viable target for the implementation of real time algorithms suited to video image processing and image processing applications. The unique architecture of the FPGA has allowed the technology to be used in many applications encompassing all aspects of video image processing and image processing. Among those algorithms, dilation, erosion and linear filtering, represent a basic set of image operations for a number of applications. the system is connected to a USB port of a personal computer, which in that way form a powerful and low-cost design station. A comparison between different HDL language and the described FPGA-based implementation is presented. Image processing algorithms are conventionally implemented in DSP, ARM processors and some special purpose processors. However all these implementation styles are limited by throughput which becomes very critical parameter for several image processing applications. The FPGA technologies offer basic digital blocks with flexible interconnections to achieve high speed digital hardware realization. The image will be transferred from computer to FPGA board using JTAG cable. After performing the required filtering process the result will be transferred back to computer.

Keywords — Field Programmable Gate Array (FPGA), Image processing algorithms, MATLAB simulation, Embedded Development Kit, Advanced RISC Machine.

1. INTRODUCTION

1.1 Field Programmable Gate Array

The FPGA provides large arrays of programmable logic resources that can be programmed an unlimited number of times. The Configurable Logic Blocks (CLBs) inside each FPGA chip can be used to implement sequential or combinational logic. The CLBs of the Xilinx 4000 family [10] chip contain two flip-flops and two function generators. The functionality of the CLB is established by programming Static Random Access Memory (SRAM) bits within the CLB that control multiplexers for signal routing or look-up tables to define the function generators. More complicated sequential or combinational logic functions can be implemented by programming the interconnections and logic of the CLBs in the array. Input/Output Blocks (IOBs), allow connection to the pins of the chip. Spartan-3 EDK Development Board utilizes Xilinx Spartan-3 XC3S200-4TQ144 device.

In video and image processing requirements for security and multimedia applications we require different morphological operation. In this work such operation will be implemented in FPGA platform using Handel C language. The image will be transferred from computer to FPGA board using JTAG cable. After performing the required process the result will be transferred back to computer. Both Matlab and FPGA results will be validated. The implementation uses basic blocks registers, adders, multipliers, control logic, UART transmitter and receiver etc. The work aims to setup an image processing platform on FPGA hardware. The results shall demonstrate the performed operation on FPGA, and validate the implemented architecture.

1.2 FPGA using Handel-C

FPGAs have traditionally been configured by hardware engineers using a Hardware Design Language (HDL). The two principal languages used are Verilog HDL (Verilog) and Very High Speed Integrated Circuits (VHSIC) HDL (VHDL) which allows designers to design at various levels of abstraction. Verilog and VHDL are specialized design techniques that are not immediately accessible to software engineers, who have often been trained using imperative programming languages. Consequently, over the last few years there have been several attempts at translating Algorithmic oriented programming languages directly into hardware descriptions.

Given the importance of digital image processing and the significance of their implementations on hardware to achieve better performance, this work addresses implementation of image processing algorithms like on dilation, erosion and linear filtering on FPGA using Handel-C language. Also novel architectures for the above mentioned image processing algorithms have been proposed. The dilation, erosion and linear filtering algorithm implementation is compared at simulation and synthesis levels with other hardware descriptive languages. The Handel-C implementation is compared with VHDL and Verilog [2].

2. MORPHOLOGICAL OPERATIONS

Mathematical morphology is a geometric approach to non linear image processing that was developed as a powerful tool for shape analysis in binary and gray scale 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.

2.1 Binary Erosion and Dilation

Erosion and Dilation are related to convolution but are more for logical decision-making than numeric calculation. Like convolution, binary morphological operators such as erosion and dilation combine a local neighborhood of pixels with a pixel mask to achieve the result. The output pixel, 0, is set to either a hit (1) or a miss (0) based on the logical AND relationship.

Binary 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.

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.

3. DESIGN IMPLEMENTED ON FPGA

The Embedded Development Kit (EDK) is an Integrated Development Environment for designing embedded processing systems. This pre-configured kit includes Xilinx Platform Studio and the Software Development kit, as well as all the documentation and IP that you require for designing Xilinx Platform FPGAs with embedded PowerPC® hard processor cores and/or MicroBlaze soft processor cores.

Processing IP and MicroBlaze Soft Processor Core -Pre-verified IP catalog, including a wide variety of processing peripheral cores for customizing your embedded systems as well as the flexible MicroBlaze 32-bit soft processing core. The MicroBlaze processor offers memory management and FPU configuration options enabling commercial grade RTOS support, unique for a soft processor.

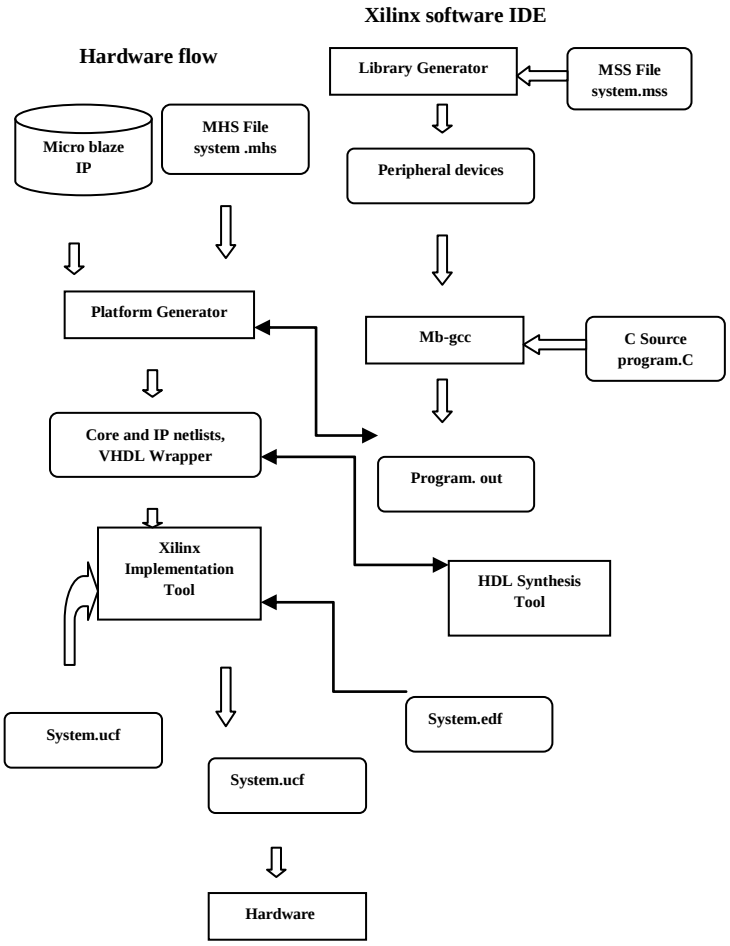

Fig 1 Design flow for MicroBlaz Virtual processor

3.1 Hardware Development

Xilinx FPGA technology allows you to customize the hardware logic in your processor subsystem. Such customization is not possible using standard off-the-shelf microprocessor or controller chips. The term "Hardware platform" describes the flexible, embedded processing subsystem you are creating with Xilinx technology for your application needs. The hardware platform consists of one or more processors and peripherals connected to the processor buses. XPS captures the hardware platform description in the Microprocessor Hardware Specification (MHS) file. The MHS file is the principal source file that maintains the hardware platform description and represents in ASCII text the hardware components of our embedded system. When the hardware platform description is complete, the hardware platform can be exported for use by SDK.

3.2 Software Development

A Board Support Package (BSP) is a collection of software drivers and, optionally, the operating system on which to build by our application. The created software image contains only the portions of the Xilinx library you use in your embedded design. You can create multiple applications to run on the BSP. The hardware platform must be imported into SDK prior to creation of software applications and BSP.

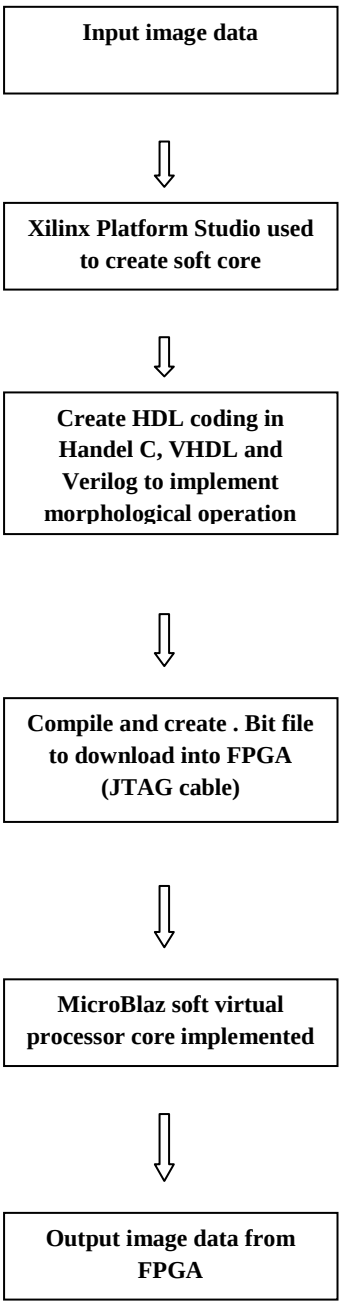

Fig 2 Proposed Block diagram of design implemented on FPGA

4. ALGORITHM

Step I: First the input image is preprocessed by Matlab and converted in to matrix format. (Image size 256 x 256)

Step II: Xilinx Platform Studio is used to implement software core processor.

Step III: HDL Coding is developed in different HDL languages (Handle C, VHDL and Verilog).

Step IV: Compile HDL coding one by one Create .bit file and download into FPGA.

Step V: Output of FPGA is stored in SRAM and post processed by Matlab. (Output image size 256x256)

Step VI: The results are compared with different HDL FPGA outputs.

5. RESULT COMPARISON

The morphological operations involved in the implementation of the described pattern spectrum algorithm, can be performed using Boolean logic in connection with delay lines at pixel level, using an adequate memory organization for the input image and the structuring element [7]. The system used in this work consists of a development platform based on the Xilinx FPGA Spartan 3E[10]. The system includes a high speed USB2 port, 16 Mb of SRAM and ROM memory, input and output ports, and support for embedded processors based on the Xilinx MicroBlaze system. The card connected to a personal computer conform a powerful design station. The operations are performed on a 256X256 input color image is converted into gray image, organized in a linear form as shown in Fig. 3. In each step it is required to evaluate the neighborhood of each pixel according to the specific operation. This decision can be performed by checking if the pixel under test can be induced by a translational displacement of pixels from the original image, while only displacements defined by the structuring element are considered. In other words: the structuring element is seen as a set of translation vectors, given by its pixel coordinates. A pixel in the result image can be obtained by a translational displacement of an original image pixel by the vector forming the structuring element. Accordingly, the erosion and dilation operations are carried out on a set of pixel pairs obtained by vector addition in the neighborhood of some pixel in the input image, and the vector positions in the structuring element, through Boolean operations.

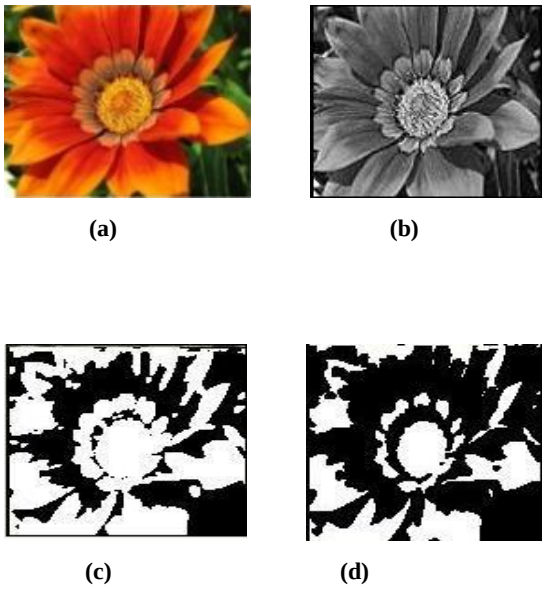

Fig 3 The morphological operations are performed by using Xilinx platform studio EDK 11.1 FPGA Spartan3E on a 256X256 input image (a. input color image, b. gray image, c. dilation image, d. erosion image)

Table 1 shows the comparative study between verilog and Handel-C HDL for various FPGA hardware resources. Table 2 summarizes the comparative design efforts for dilation, erosion and linear filtering using Verilog, VHDL and Handel-C.

Table1.Experimental results with comparison

FPGA Resources	VERILOG	HANDEL-C
Speed (Xilinx Spartan 3)	**NA**	**16 MHz**
Slices	**58%**	**10%**
Slice Registers	**57%**	**3%**
LUTs	**55%**	**5%**
IOB	**9%**	**35%**
GCLKs	**100%**	**50%**
Block RAMs	**-**	**6%**
Equivalent Gate Count	**306,236**	**127, 176**

Table2. Comparison of design efforts for Morphological operations using Handle C, VHDL and Verilog

	HANDEL-C	VHDL	Verilog
Design Time	**46 hrs**	**65hrs**	**56 hrs**
Compilation Time	**50 Sec**	**72 Sec**	**54 Sec**
Program Size (lines)	**1233**	**2232**	**1318**

6. CONCLUSION

In this work, an implementation of linear filtering and morphological image operation using a Xilinx platform studio EDK 11.1 FPGA Spartan 3E have been implemented. It has been observed from the results that Handel-C performance for dilation, erosion and linear filtering in synthesis level is better than VHDL and Verilog implementations. Besides, Handel-C language is introduced as a language for Software Engineers to quickly prototype software concepts in hardware using behavioral model. Our current efforts to efficiently implement image processing algorithms in hardware using high level languages ascertains that hardware knowledge and implementation at Register Transfer Level (RTL) are necessary. However, the class of image processing algorithms considered in our work is limited to basic algorithms and further work on complex image processing algorithms need to be performed.

7. REFERENCES

[1] D. Baumann, J. Tinembart. *Mathematical Morphology Image Analysis on FPGA*, IEEE Int. Conf. on Advances in Intelligent Systems Theory and Applications, 2004.

[2] Daggu Venkateshwar Rao, Shruti Patil, Naveen Anne Babu, V Muthukumar , *Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture Using C-based Hardware Descriptive Languages*, International Journal of Theoretical and Applied Computer Sciences Volume 1 Number 1 (2006) pp. 9–34 (c) GBS Publishers and Distributors (India).

[3] 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.

[4] Luca Breveglieril, Vincenzo piuri, *Digital Median Filters*, Journal of VLSI Signal Processing 31, 191–206, 2002

[5] Komal Vij,et al. "Enhancement of Images Using Histogram Processing Techniques Vol 2" , pp309313, 2009.

[6] M Rama Bai,"A New Approach For Border Extraction Using Morphological Methods", pp3832-3837,2010.

[7] Kevin Loquin,et al. "Convolution Filtering And Mathematical Morphology On An Image: A UnifiedView", pp1-4, 2010.

[8] Przemysław Kupidur,"Semi-automatic method for built-up area intensity survey using morphological granulometry",pp271-277, 2010.

[9] Abhishek Acharya, et al. "FPGA Based Non Uniform Illumination Correction in Image Processing Applications Vol 2", pp349-358, 2009.

[10] www.xilinx.com

8. AUTHOR'S PROFILE

R. Arunmozhi received B.E. in Electrical Engineering from Annamalai University in 2005 and ME in Process Control and Instrumantation from Annamalai University, Annamalai Nagar, Chidambram. He is working as an Assistant Professor in Department of Electrical Engineering, Annamalai University, AnnamalaiNagar,Tamil Nadu. His research interest is in the area of VLSI design and Data communication& Networking.

Dr. G. Mohan received B.Tech. in MIT from Anna University and ME in Power System from Annamalai University, Annamalai Nagar, Chidambram. He is a Professor in Department of Electrical Engineering, Annamalai University, Annamalai Nagar, TamilNadu. His research interest is in the area of Power System Analysis and Simulations.