



# Implementation of Morphological Image Processing on FPGAs

BHAVANA SHARMA<sup>1</sup>, V.K. PANDEY<sup>2</sup>, AYUB KHAN<sup>3</sup>

Assistant Professor, Dept. of ECE, G L Bajaj Institute of Technology & Management, Greater Noida, India<sup>1</sup>

Professor (HOD), Dept. of ECE, G L Bajaj Institute of Technology & Management, Greater Noida, India<sup>2</sup>

Associate Professor, Dept. of ECE, Anand Engineering College, Agra, India<sup>3</sup>

**ABSTRACT:** The concept of Image Processing is totally related to real time work, which is done by FPGA. Mathematical morphology is a well known image and signal processing technique. However, most morphological tools such as MATLAB are not suited for strong real-time constraints. Application of FPGA Coprocessors as a means of delivering hardware IP to software and system engineers is presented. The hardware and software architecture of FPGA coprocessors is described in detail. This paper gives the algorithm and implementation of morphological image processing using median filter on FPGA. The design is implemented on a Xilinx xc3s500e-4fg320 FPGA chip.

**Keywords:** Morphology, MATLAB, FPGAs, Image Processing, XC3S500E-4FG320

## I. INTRODUCTION

FPGA coprocessors are ubiquitous. They can be used with standard DSP processors to offload computationally intensive tasks, or to provide digital signal processing capabilities to a general purpose microprocessor. They can be used with processors embedded within FPGAs as hard or soft logic, or with off-FPGA processors.

It is widely recognized that FPGAs are very efficient for the implementation of many computationally complex digital signal processing algorithms. In comparison with programmable DSP processors, they can deliver a lower-cost and lower-power solution for a variety of algorithms. FPGAs, however, do not offer the same flexibility and ease of design as DSP processors. FPGA coprocessors are blocks of hardware IP that can easily be integrated into a processor-based system in order to offload some of the most computationally intensive tasks. A combination of standardized hardware interfaces, design automation tools to assemble a system, and a standardized software API forms the concept of FPGA coprocessors. The design automation tools and software API make it possible for system and software engineers to make use of hardware IP with a minimum of actual FPGA design. The standardized interfaces provide orthogonally. If an IP designer conforms to the standards, an IP block can be used as a coprocessor with any of the supported processors. In a similar way, once the necessary interface hardware and software drivers have been created, all FPGA coprocessor IP can be used with that processor.

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. Implementing such applications on a general purpose computer can be easier, but not very time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances in the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative. Implementing complex computation tasks on hardware and by exploiting parallelism and pipelining in algorithms yield significant reduction in execution times. There are two types of technologies available for hardware design. Full custom hardware design also called as Application Specific Integrated Circuits (ASIC) and semi custom hardware device, which are programmable devices like Digital signal processors (DSPs) and Field Programmable Gate Arrays (FPGA's). Full custom ASIC design offers highest performance, but the complexity and the cost associated with the design is very high. The ASIC design cannot be changed and the design time is also

Very high. ASIC designs are used in high volume commercial applications. FPGA coprocessors by definition implement computationally complex tasks, mainly in the field of digital signal processing. Candidate functions include FIR filters, FFT processors, and error correction and detection.

During design fabrication the presence of a single error renders the chip useless. DSPs are a class of hardware devices that fall somewhere between an ASIC and a PC in terms of the performance and the design complexity. DSPs are specialized microprocessors, typically programmed in C, or with assembly code for improved performance. It is well suited to extremely complex math intensive tasks such as image processing. Knowledge of hardware design is still required, but the learning curve is much lower than other design choices. Applications of coprocessors are varied, but include software-defined radio as well as error correction and detection in base-stations for mobile communications. Hardware Architecture In order to be used as an FPGA coprocessor, a hardware IP block needs to use defined standard interfaces. The design automation tools can take this IP block and connect it to ancillary functions such as FIFOs, DMA controllers and bus interfaces to create a system. Altera has chosen to use a defined subset of the Avalon<sup>1,2</sup> interface for the data input and output ports of an IP block and a simple Avalon<sup>3</sup> slave interface for control and status.

Two hardware architectures have been defined. The first, known as “Type F” (see ) uses DMA controllers in the FPGA to move data between the processor’s memory and the coprocessor, and is particularly suited to systems where the processor’s main memory is accessible directly from the FPGA. This is typically true when the processor itself is included in the FPGA.

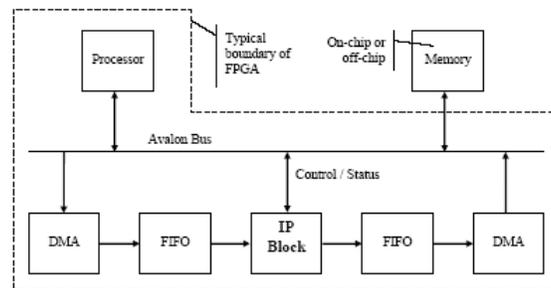


Fig.1 FPGA Coprocessor Architecture with DMA Controller in FPGA (Type F)

The second architecture, known as “Type E” (Figure 2) uses an off-FPGA DMA controller, typically built into a standard processor chip, and is particularly suited to systems where the main memory is not directly accessible from the FPGA. Both architectures make use of the same IP block, but different ancillary functions. These ancillary functions are shared between all IP blocks and so need only be designed once for all coprocessors. The same design automation tool is used to assemble the systems, and both architectures present the same software API to the user.

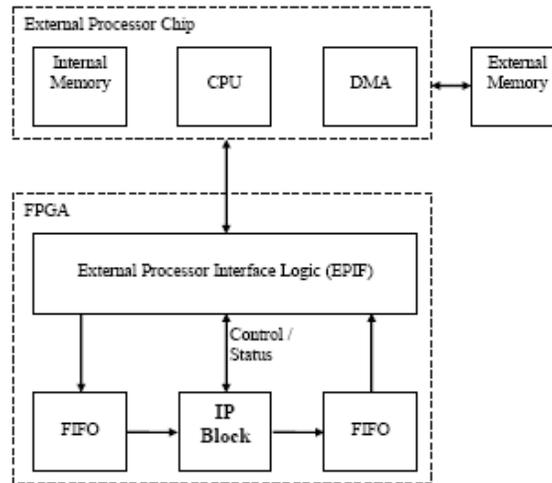


Fig. 2 FPGA Coprocessor Architecture with Off-FPGA DMA Controller (Type E)

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. 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 median filter, morphological, convolution and smoothing operation and edge detection on FPGA using VHDL language. Also novel architectures for the above mentioned image processing algorithms have been proposed. Gray-level images are very common in image processing. These types of images use eight bits to code each pixel value, which results in 256 different possible shades of grey, ranging from 0 (black value) to 255 (white value). Latest generations FPGAs compute more than 160 billion multiplication and accumulation (MAC) operations per second.

## II. IMAGE PROCESSING ALGORITHM

This section discusses the theory of most commonly used image processing algorithms like (1) Filtering, (2) Morphological operations

### A. Median Filtering

A median filter is a non-linear digital filter which is able to preserve sharp signal changes and is very effective in removing impulse noise (or salt and pepper noise). An impulse noise has a gray level with higher or lower value that is different from the neighborhood point. Linear filters don't have ability to remove this type of noise without affecting the distinguishing characteristics of the signal. Median filters have remarkable advantages over linear filters for this particular type of noise. Therefore median filter is very widely used in digital signal and image/video processing applications. A standard median operation is implemented by sliding a window of odd size (e.g. 3x3 window) over an image. At each window position the sampled values of signal or image are sorted, and the median value of the samples replaces the sample in the center of the window as shown in Figure 3.

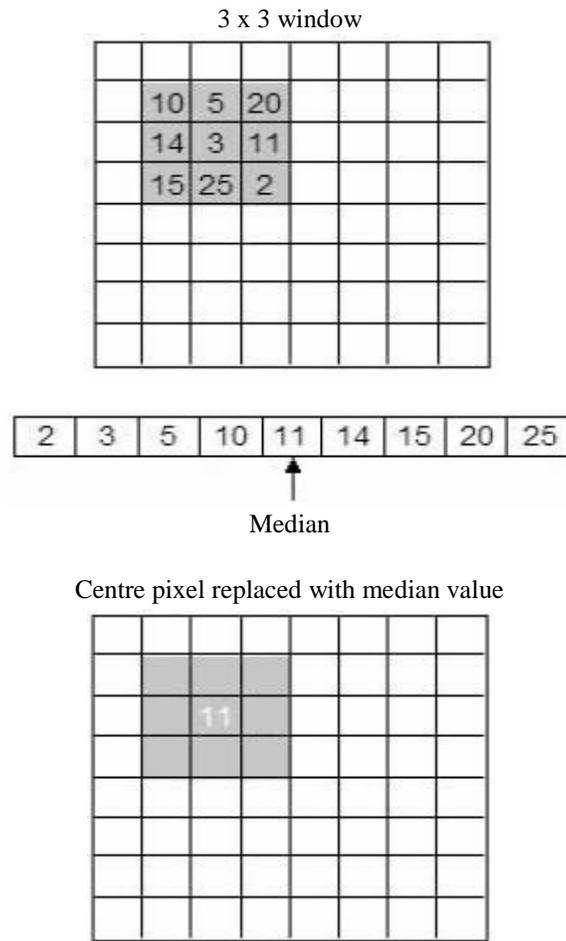


Fig.3 Median Filter

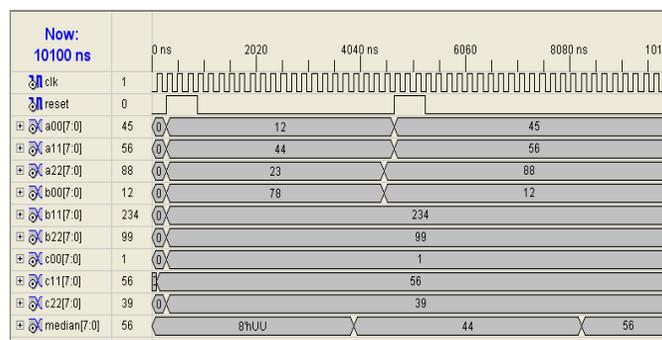


Fig.4 Simulation result of median Filter

**B. Morphological Operation**

The term morphological image processing refers to a class of algorithms that transforms the geometric structure of an image. Morphology can be used on binary and gray scale images, and is useful in many areas of image processing, such as skeletonization, edge detection, restoration and texture analysis.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the *structuring element* used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image.

*Rules for Dilation and Erosion:*

*Dilation*-The value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

*Erosion*-The value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.

The grayscale erosion is performed by minimum filter, whereas the dilation is performed by maximum filter. In a 3 x 3 minimum filter, the centre pixel is replaced by a minimum value of the pixels in the window. In a maximum filter, the centre pixel is replaced by a maximum value of the pixels in the window. The implementation of minimum and maximum filters is similar to the median filters implementation.

Morphological Dilation of a grayscale image

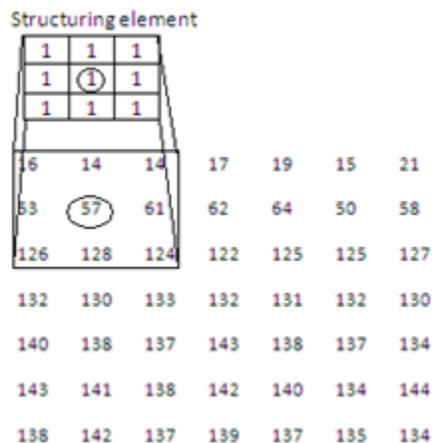


Fig.5 Input image

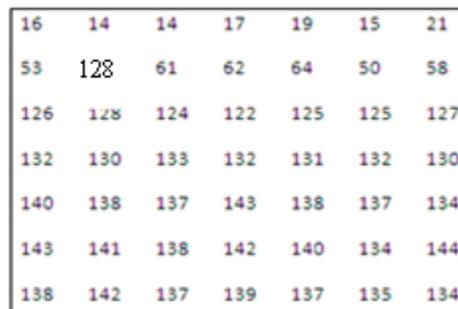


Fig.6 Output image

**III. ALGORITHM**

Step I: First the numbers are sorted vertically i.e. sort elements of each column in the ascending order.

Step II: Numbers are sorted horizontally i.e. sort elements of each row in the ascending order.

Step III: Sort the cross diagonal elements and pick up the middle element as the median element of the window. Minimum is the first and maximum is the last element in the window of the nine elements.

The block schematic of sort-3 is shown in Figure 3. 3 x 3 window pixels values are sorted and outputs H[7:0], M[7:0], L[7:0] corresponding to maximum, median and minimum values respectively.

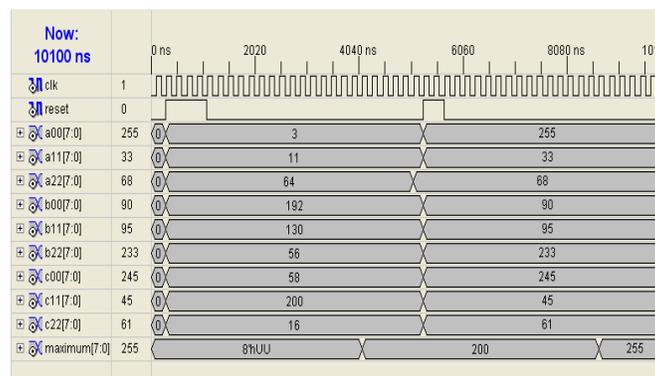


Fig7 Simulation result of Dilation

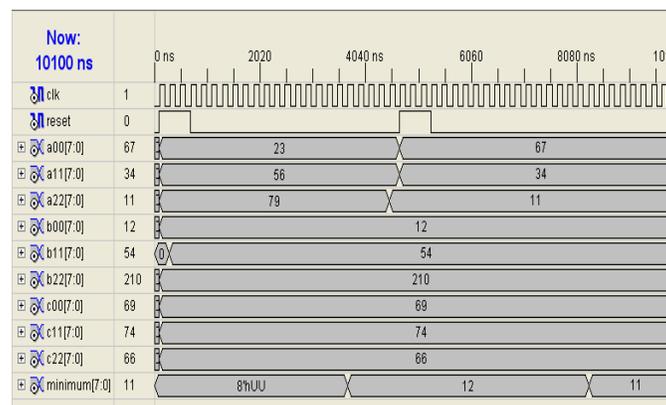


Fig.8 Simulation result of Erosion

**A: Implementation**

The entire implementation of image acquisition, image processing and image retrieval is shown in block diagram of figure 4. In order to reduce complexity of data transactions, RAM is implemented on FPGA. UART is implemented to facilitate data acquisition and communication between PC and FPGA board.

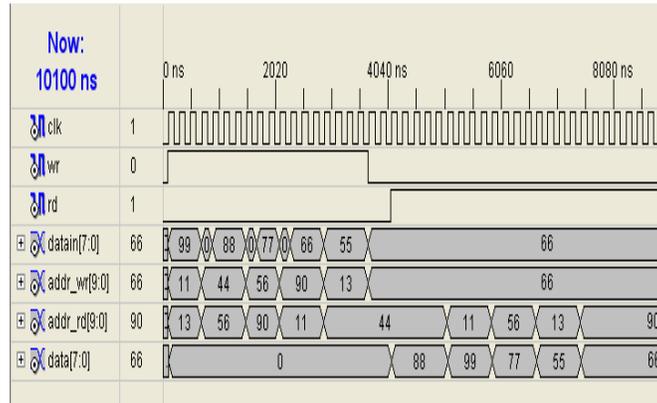


Fig.9 Simulation result of RAM

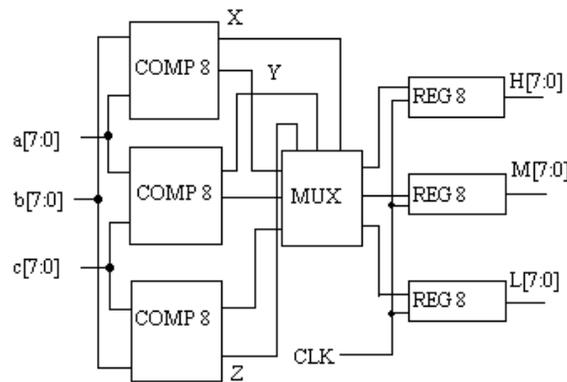


Fig.10 Block schematic of sorter

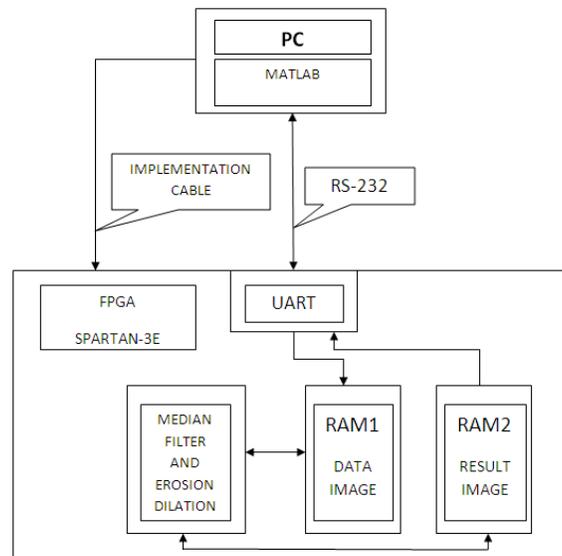


Fig.11 Block diagram of design implemented on FPGA



#### IV. CONCLUSION

The FPGA Coprocessor architecture greatly simplifies the process of offloading computationally intensive functions from a programmable processor into dedicated hardware. This is achieved through a combination of standardized hardware and software interfaces, and the use of design automation tools. A suitably packaged block of IP can be implemented in an FPGA with a minimal requirement for hardware design, and can be accessed through a standardized software API.

#### REFERENCES

- [1] D. Baumann, J. Tinembart. Mathematical Morphology ImageAnalysis 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 Breveglieri, Vincenzo piuri, Digital Median Filters, Journal of VLSI Signal Processing 31, 191–206, 2002
- [5] Atlantic Interface, Functional Spec A-FS-13-3.0, Altera Corporation, June 2002
- [6] [www.xilinx.com](http://www.xilinx.com)
- [7] [www.mathworks.com](http://www.mathworks.com)