

Hardware Implantation Of Morphological Image Filtering Based On 2d Convolution

Shaik Thaseen
M.Tech student, Dept of ECE,
Vaagdevi Institute of Technology & Science,
Proddatur, Kadapa (DT), AP-516361

C. Md. Aslam,
HOD, Dept. of ECE,
Vaagdevi Institute of Technology & Science
Proddatur, Kadapa (DT), AP-516361

ABSTRACT: *Field Programmable Gate Array (FPGA) technology has become a viable target for the implementation of real time algorithms suited to video 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. Among those algorithms, linear morphological filtering based on a 2D convolution; filters represent a basic set of image operations for a number of applications. In this work, an implementation of linear and morphological image filtering using a FPGA Spartan 3 EDK with educational purposes is presented. 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. After performing the required filtering/processing the result will be transferred back to PC. In PC both the results will be validated.*

Keywords— *Field Programmable Gate Array (FPGA), Image processing algorithms, MATLAB simulation.*

I. INTRODUCTION

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? 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.

In this paper, linear morphological filtering - based image processing system is developed on a Xilinx Spartan3 Field Programmable Gate Array (FPGA) device using an embedded development kit (EDK) from Xilinx... This paper is organized as follows: Section II briefly reviews Back Ground Subtraction method. Section III discusses the design flow. Section IV covers different architecture for video surveillance co-processor and compares their performance. Section V is the conclusion part.

II. 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. 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. Morphological dilation is defined as the set union of the objects A obtained after the translation of the original image for each coordinate pixel b in the structuring element B.

i) Dilation Operation:

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 cell will take care of it. Row contains Data banks which is equal to the dimensions (only width) of the image. Row cell 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_cel 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_cel 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.

ii) Erosion Operation:

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_cel will take care of it. Row contains Data banks which is equal to the dimensions (only width) of the image. Row cel 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_cel 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 cel 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.

III. DESIGN FLOW

To build an embedded system on Xilinx FPGAs, the embedded development kit (EDK) is used to complete the reconfigurable design. Figure 1 shows the design flow.

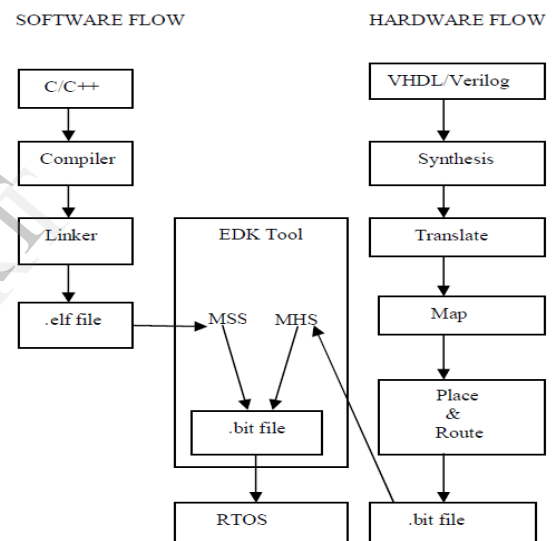


Fig 1: Design flow

Unlike the design flow in the traditional software design using C/C++ language or hardware design using hardware description languages, the EDK enables the integration of both hardware and software components of an embedded system. For the hardware side, the design entry from VHDL/Verilog is first synthesized into a gate-level netlist, and then translated into the primitives, mapped on the specific device resources such as Look-up tables, flip-flops, and block memories. The location and interconnections of these device resources are then placed and routed to meet with the timing Constraints. A downloadable .bit file is created for the whole hardware platform.

The software side follows the standard embedded software flow to compile the source codes into an executable and linkable file (ELF) format. Meanwhile, a microprocessor software specification (MSS) file and a microprocessor hardware specification (MHS) file are used to define software structure and hardware connection of the system. The EDK uses these files to control the design flow and eventually merge the system into a single downloadable file. The whole design runs on a real-time operating system (RTOS).

IV. MORPHOLOGICAL FILTERING CO-PROCESSOR

There are different ways to include processors inside Xilinx FPGA for System-on-a-Chip (SoC): PowerPC hard processor core, or Xilinx MicroBlaze soft processor core, or user-defined soft processor core in VHDL/Verilog. In this work, The 32-bit MicroBlaze processor is chosen because of the flexibility. The user can tailor the processor with or without advance features, based on the budget of hardware. The advance features include memory management unit, floating processing unit, hardware multiplier, hardware divider, instruction and data cache links etc. The architecture overview of the system is shown in Figure 2. It can be seen that there are two different buses (i.e., processor local bus (PLB) and fast simplex link (FSLbus) used in the system [5-6]. PLB follows IBM core connect bus architecture, which supports high bandwidth master and slave devices, provides up to 128-bit data bus, up to 64-bit address bus and centralized bus Arbitration. It is a type of shared bus. Besides the access overhead, PLB potentially has the risk of hardware/software incoherent due to bus arbitration. On the other hand, FSL supports point-to-point unidirectional communication. A pair of FSL buses (from processor to peripheral and from peripheral to processor) can form a dedicated high speed bus without arbitration mechanism. Xilinx provides C and assembly language support for easy access. Therefore, most of peripherals are connected to the processor through PLB; the DWT coprocessor is connected through FSL instead.

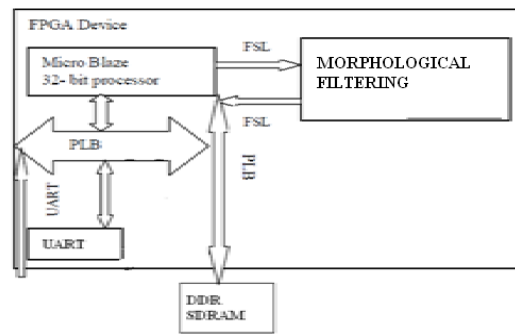


Fig 2: System Overview

The current system offers several methods for distributing the data. These methods are a UART, and VGA, and Ethernet controllers. The UART is used for providing an interface to a host computer, allowing user interaction with the system and facilitating data transfer. The VGA core produces a standalone real-time display. The Ethernet connection allows a convenient way to export the data for use and analysis on other systems. In our work, to validate the DWT coprocessor, an image data stream is formed using VISUAL BASIC, then transmitted from the host computer to FPGA board through UART port.

V. EXPERIMENTAL RESULTS

Experiments are performed on gray level images to verify the proposed method. These images are represented by 8 bits/pixel and size is 128 x 128. Image used for experiments are shown in below figure

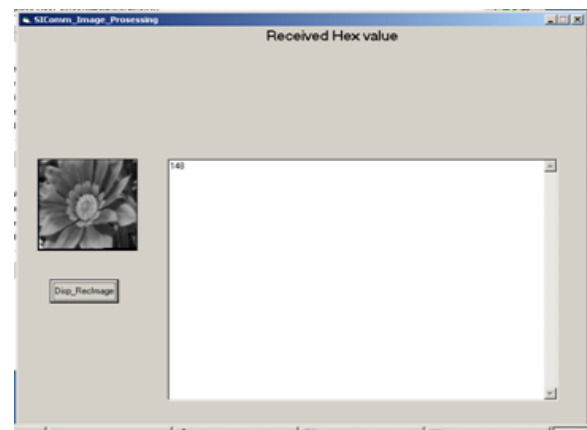


Fig3: Input Image

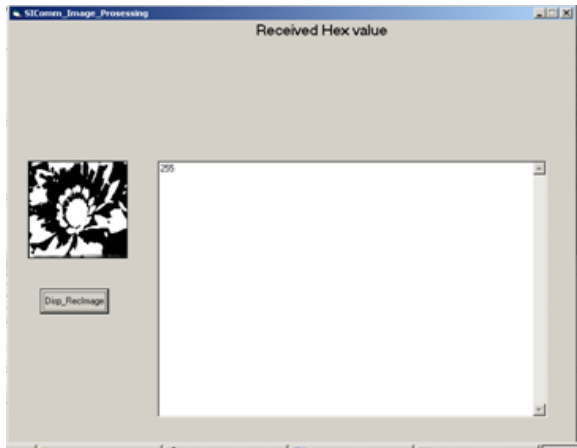


Fig4: Detection of Edges

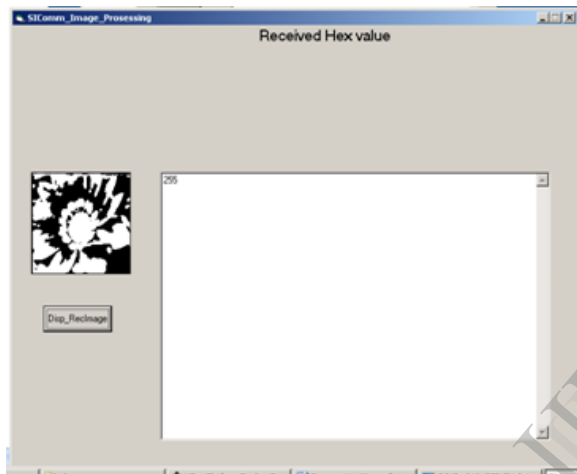


Fig5: Applying Morphological Operator

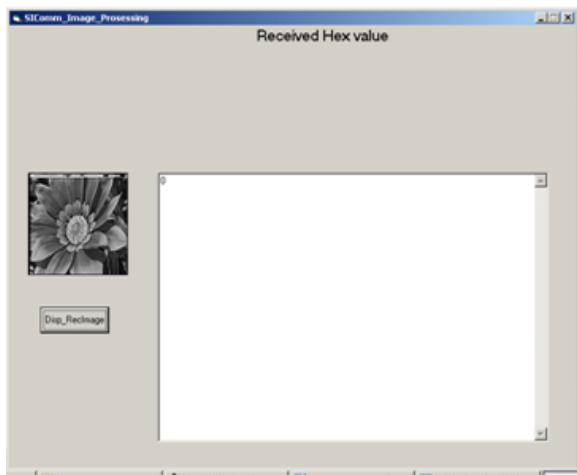


Fig6: Final Output Image

The entropy (E) is defined as Where s is the set of processed coefficients and p (e) is the probability of processed coefficients. By using entropy, number of bits required for compressed image is calculated. An often used global objective quality measure is the mean square error (MSE) defined as

Where, nxm is the number of total pixels. f (i,j) and f(i,j)' are the pixel values in the original and reconstructed image. The peak to peak signal to noise ratio (PSNR in dB) is calculated as follows:

And the synthesis report is below

```

Selected Device : 3s500efg320-4

Number of Slices:           2649 out of 4656 56%
Number of Slice Flip Flops: 3343 out of 9312 35%
Number of 4 input LUTs:    3794 out of 9312 40%
    Number used as logic:   3118
    Number used as Shift registers: 356
    Number used as RAMs:    320
Number of IOs:              83
Number of bonded IOBs:     40 out of 232 17%
    IOB Flip Flops:         55
Number of BRAMs:           7 out of 20 35%
Number of MULT18X18SIOs:   3 out of 20 15%
Number of GCLKs:           7 out of 24 29%
Number of DCMs:            2 out of 4 50%

Timing Summary:
-----
Speed Grade: -4

Minimum period: 12.384ns (Maximum Frequency: 80.749MHz)
Minimum input arrival time before clock: 41.553ns
Maximum output required time after clock: 13.840ns
Maximum combinational path delay: 3.344ns
    
```

Fig. 7 Synthesis report

VI. CONCLUSIONS

In this paper, a morphological image filtering reconfigurable system is designed using the EDK tool. Hardware architectures of Morphological filtering algorithm have been implemented as a coprocessor in an

embedded system. the hardware cost of these architecture is compared for benchmark images. This type of work using EDK can be extended to other applications of embedded system.

REFERENCES

[1] 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.

[2] 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.

[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] 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.

[5] 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.

[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 Aplicación Para la Enseñanza de Procesado de Señal", IEEERITA, Vol. 2, No. 1, 2007.

[8] David Báez-López, David Báez-Villegas, René Alcántara, 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.

[9] Malay Haldar, Anshuman Nayak, Alok Choudhary, Prith 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