International Journal of Combined Research & Development (IJCRD)
eISSN:2321-225X; pISSN:2321-2241 Volume: 2; Issue: 5; May -2014

FPGA IMPLEMENTATION OF RECONFIGURABLE PROCESSOR TO PERFORM MULTIPLE OPERATIONS OF BINARY IMAGE PROCESSING

Vijay.A
PG student , M.Tech(VLSI & ES),Dept. of ECE.
AMCEC, Bangalore
Email: vijay.a@rediffmail.com

Meena Deshpande
Assistant Professor ,Dept. of ECE.
AMCEC, Bangalore
Email: meenapandit26@gmail.com

ABSTRACT: In this paper a reconfigurable processor to perform multiple operations of binary image processing is proposed. The processor's architecture consists of reconfigurable binary processing module and multiplexer circuits. The reconfigurable binary processing module consists of many binary computational units. Each binary computational unit has unique operation implemented with related algorithm which can be used for different applications. Some examples of binary computational units are like Median filtering with sort optimization, Dilation and Erosion without using structuring element, Edge detection with sobel edge algorithm, etc. This processor is designed in Verilog language and implemented on Vertex-4 Field Programmable Gate Array.

Keywords

Binary image processing, Reconfigurable, Binary computational unit(BCU), Field Programmable Gate Array(FPGA) etc.

1. INTRODUCTION

Binary image processing is extremely useful in various areas, such as object recognition, tracking, motion detection and machine intelligence, image analysis and understanding, video processing, computer vision and identification and authentication systems. Binary image processing has been commonly implemented using processors such as CPU or DSP. However, it is inefficient and difficult to use such processors for binary image processing.

High speed implementation of binary image processing operations can be efficiently realized by using chips specialized for binary image processing. Application specific chips have been used for various applications but each chip was used for specific applications. The major drawback of application specific chip is lack of flexibility. The reconfigurable technique can bridge the gap between application-specific integrated circuits and flexibility. Some of the chips are made of analog circuit and some are made up of an analog part and a digital part. When compared with the digital part, the analog part shows low robustness, accuracy, and scalability although it has a small area and low power consumption.

This paper presents a processor that consists of a reconfigurable binary processing module, including reconfigurable binary compute units and output control logic, input and output image control units, and peripheral circuits. The reconfigurable binary processing module includes many binary computational units and converters. Each binary computational unit has unique operation implemented with related algorithm which can be used for different application. Some of the BCU are like Median filtering with sort optimization, Dilation and Erosion without using structuring element, Edge detection with sobel edge algorithm. The BCU has the characteristics of high flexibility, efficiency, and performance. Basic mathematical morphology operations and complicated algorithms can easily be implemented on it. The processor has the merit of high speed, simple structure, and wide application range.

2. ARCHITECTURE

The architecture of the proposed processor is shown in Fig-1. The processor consists of a reconfigurable binary processing module consisting of binary compute units and multiplexer circuits.

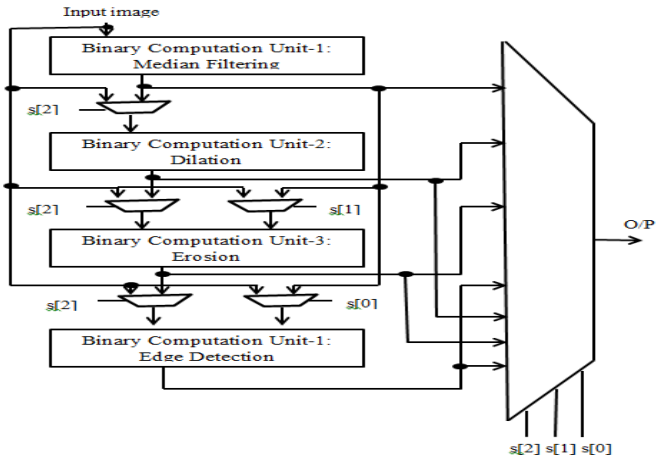

Fig 1. Architecture of reconfigurable binary processing module

2.1 Reconfigurable binary processing module

The reconfigurable binary processing module contains four binary compute units. The input image , which is a converted image text using mat lab software, is given as inputs to all the binary computational units. Based on the command given, following operations take place as shown in Table-1. For the command 000, median filtering is done to remove noises(salt and pepper noise). Similarly for the command 001,010 and 011, dilation, erosion and edge detection. For 100 command median filtering output is given as input to dilation unit and dilation output is performed. For 101 command median filtering output is given as input to erosion unit and erosion output is performed. For 110 command , median filtering output is given as input to edge detection unit and edge detection is performed. Based on the applications we can perform the operations.

Table 1. Some of the examples of operations.

COMMAND			OPERATIONS(OUTPUTS)
S[2]	S[1]	S[0]	
0	0	0	BCU1
0	0	1	BCU2
0	1	0	BCU3
0	1	1	BCU4
1	0	0	BCU1 + BCU2
1	0	1	BCU1 + BCU3
1	1	0	BCU1 + BCU4

2.2 Binary Computational Unit-1: Median Filtering using Sort Optimization

Median filtering is a nonlinear method used to remove noise such as salt and pepper noise from images. The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighboring pixels.

The formula of median filtering can be expressed as:

$$g(x,y)= \text{med}\,[f(x-i, y-i)]$$

where g(x,y) - filtered image

f(x,y) - original image

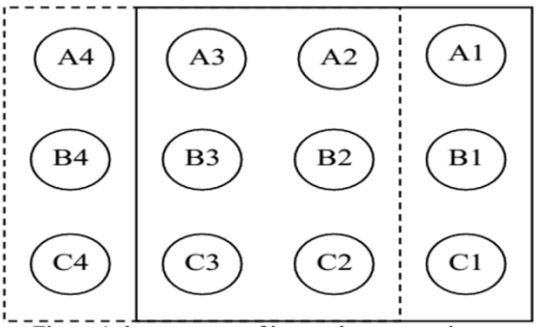

Fig 2. Sequence of image processing data.

First, we must sort the data by column. We need to do 9 comparisons to get the following result, we assume the result is

A1>B1>C1
A2>B2>C2
A3>B3>C3

Secondly, we need to find the maxim data in C1, C2 and C3 by doing 2 comparisons, and find the minimum data in A1, A2 and A3 by doing 2 comparisons, and find the median data in B1, B2 and B3 by doing 3 comparisons .The formula is as following

Amin = min(A1,A2,A3)
Bmed = med(B1,B2,B3)
Cmax = max(C1,C2,C3)

Lastly, we need to find the median data in Amin, Bmed and Cmax by doing 3 comparisons.

Fmed = med（Amin, Bmed, Cmax）

The resulted Fmed is the final result.

In the next circle, the comparison results of （A2,B2,C2） and （A3,B3,C3） have been obtained, we only need to seek the sorting of the updating data A4, B4, C4, and then use previous equations to calculate the required median.

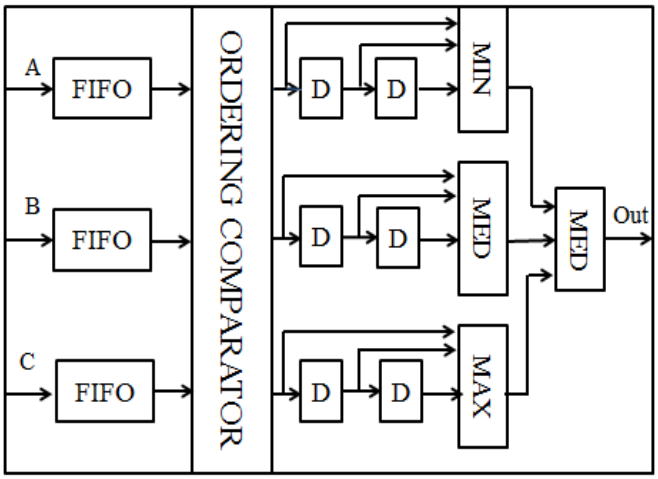

Fig 3. Hardware structure of median filter module.

As shown in Fig-3, the workflow of the median filtering module is that: firstly, three lines of data are wrote into the internal FIFO of FPGA, and when the third FIFO turn into half full

state, the median filtering system begin to work. The data in the 3 FIFO will be sent to the ordering comparator ordinarily for data ordering, and the results will be send to next different comparators. Before the second comparing, the data need 2 circle delay to distinguish the circle order of input data by two D flip-flops. The second comparing results will be sent to the final median comparator to get the final result.

2.3 Binary Computational Unit-2&3: Dilation & Erosion

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.

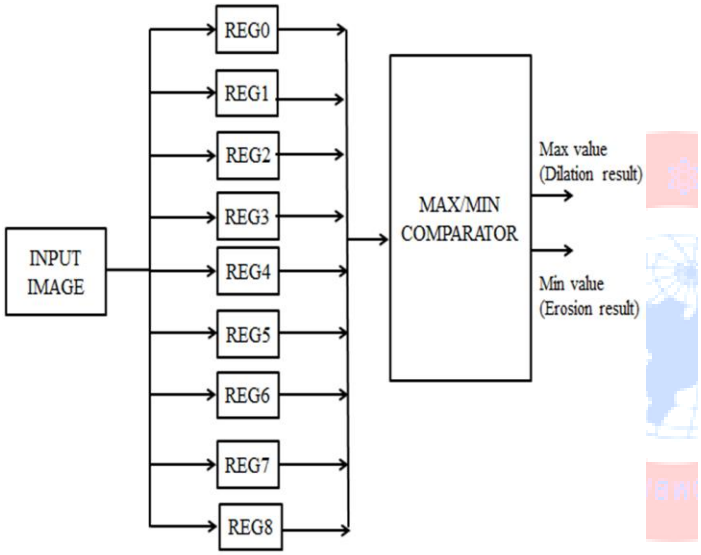

Fig 4. Block diagram of Dilation & Erosion.

From the input image, (3x3) window is selected which contains 9 input pixels are sent to registers as shown in Fig-4 and this is compared in max/min comparator. The max value gives the dilation result and min value gives erosion result. Comparison is done for remaining input pixels in the same manner. Here we are not using structuring element which has to be convolved with input image and then its compared, by doing without using structuring element we can reduce the hardware resources.

2.4 Binary Computational unit-4: Edge detection using Sobel edge algorithm

Edge detection is in the forefront of image processing for object detection. Sobel which is a popular edge detection algorithm is considered. In the edge function, the Sobel method uses the derivative approximation to find edges. Therefore, it returns edges at those points where the gradient of the considered image is maximum. The Sobel operator performs a 2-D spatial gradient measurement on images. It uses a pair of horizontal and vertical gradient matrices whose dimensions are 3×3 for edge detection operations.

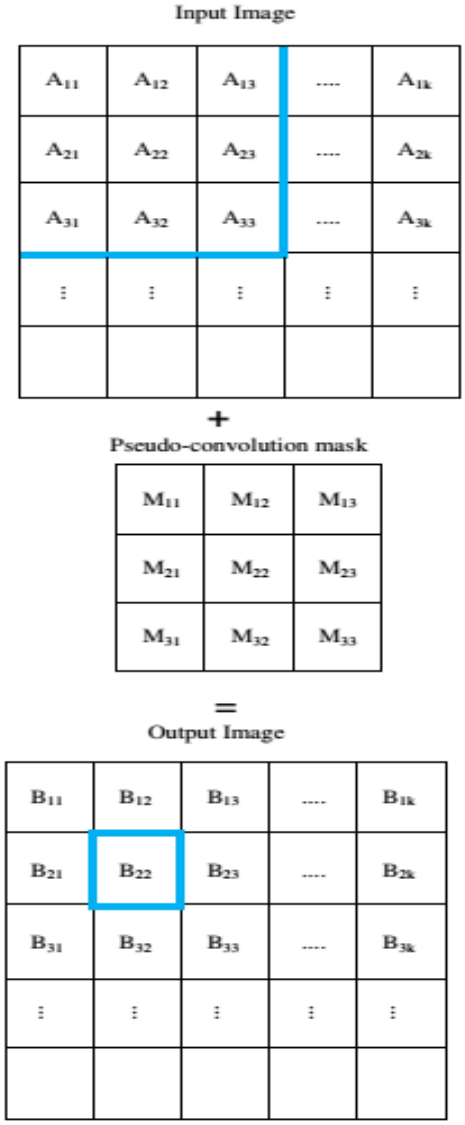

Fig 5. Sobel pseudo convolution mask applied to the image

A convolution mask is used is usually much smaller than the actual image. As a result, the mask is slid over an area of the input image, changes that pixel's value and then shifts one pixel to the right and continues to the right until it reaches the end of a row. It then starts at the beginning of the next row. The example shows the mask being slid over the top left portion of the input image represented by the green outline. The formula shows how a particular pixel in the output image would be calculated. The center of the mask is placed over the pixel you are manipulating in the image. And the I & J values are used to move the file pointer so you can multiply, for example, pixel (A22) by the corresponding mask value (M22). It is important to notice that pixels in the first and last rows, as well as the first and last columns

cannot be manipulated by a 3x3 mask. This is because when placing the center of the mask over a pixel in the first row (for example), the mask will be outside the image boundaries.
The above Fig-5 illustrates the working of sobel pseudo convolution masks when it is applied to the input given input image. The sobel pseudo mask performs the following operation to calculate the point B22 of the image:
B22 = (A11*M11) + (A12*M12) + (A13*M13) + (A21*M21) +(A22*M22) + (A23*M23) + (A31*M31) + (A32*M32) + (A33*M33)

The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation (call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

This absolute magnitude is the only output required, the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator as shown in Fig-5.

3. RESULTS

Fig(a) input image with noise

Fig(b) input image

Fig (c) median filter output

Fig (d) erosion output

Fig (e) dilation output

Fig (f) edge detection output

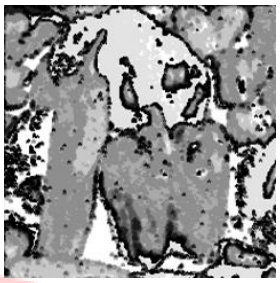

Fig (g) median filter + erosion output

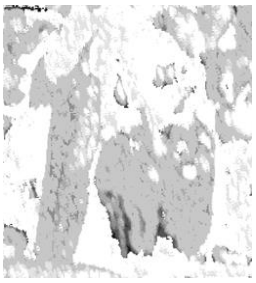

Fig (h) median filter + dilation output

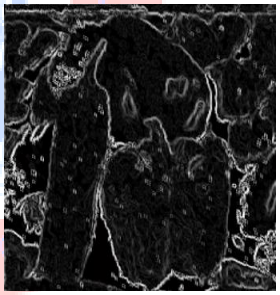

Fig (i) median filter + edge detection output

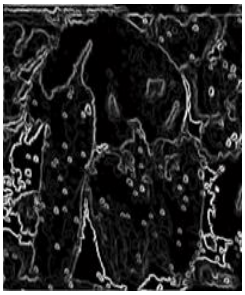

Fig (j) median filter + erosion + edge detection output

The fig (a) shows the image consisting of noise which is type of salt and pepper noise. When this image is passed through median filter , output obtained is shown in fig (c).The fig (b) shows the image with no noise, when this image is passed through erosion, dilation and edge detection unit , output obtained is shown in fig (d), fig (e), fig (f) respectively.

The fig (g) shows the output image obtained from median filter + erosion units. The fig (h) shows the output image obtained from median filter + dilation units. The fig (i) shows the output image obtained from median filter + edge detection units. And also we can get output from three parallel units which is shown in fig (j) , it is obtained fom median filter + erosion + edge detection units.

4. CONCLUSION

In this paper, a reconfigurable binary image processor is proposed to perform multiple operations of binary image processing. The processor consists of a reconfigurable binary processing module and multiplexer circuits. The reconfigurable binary processing module architecture with the characteristics of high efficiency, performance. Basic morphology operation and complicated algorithms can easily be implemented on it because of its simple structure. The processor, featured by simple structure, and wide application range, is suitable for binary image processing, such as object recognition, object tracking and motion detection, computer vision, identification and authentication.

5. REFERENCES

[1] "Sort Optimization Algorithm of Median Filtering Based on FPGA", Yan Lu, Ming Dai, Lei Jiang, 2010 IEEE.

[2] "Sobel Edge Detection Algorithm", Samta Gupta, Susmita Ghosh , February 2013.

[3] "Reconfigurable Processor for Binary Image Processing", Bin Zhang, Kuizhi Mei and Nanning Zheng, May 2013.

[4] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction" Feb 2005.

[5] S. Chien and L. Chen, "Reconfigurable morphological image processing accelerator for video object segmentation" J. Signal Process,2011