2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia

Efficient Implementation of Mean Formula for Image Processing using FPGA Device

Aqwam Rosadi Kardian and Sunny Arief Sudiro
STMIK Jakarta STI&K
Jakarta, Indonesia
{aqwam,sunny}@jak-stik.ac.id

Sarifuddin Madenda
Universitas Gunadarma
Jakarta, Indonesia
sarif@staff.gunadarma.ac.id

***Abstract*—Image processing needs a process based on mathematical formula applied to image data. In software it is easy to do that and accessing data from memory. The other hand for hardware implementation with a lot of constraint. This article propose an implementation of optimum mean formula in FGPA Device. For mean calculation only need one addition component (in one accumulator) and one division using shift right register, for 8x8 image size need 64 clock cycle to finish the mean calculation, comparing other approach that need more than 512 addition component.**

Keywords—mean, histogram, FPGA, counter, image

I. INTRODUCTION

Digital image processing is a proccess for manipulating and analyzing image with computer. Image processing can be divided into two kinds of activities:

1. Increasing the quality of image for easy interpretation by human eye, this activity known as image enhancement
2. Processing information in the image for object identification otomatically

Second application has tight correlation with pattern recognition which usually recognizing the object by extracting important information from the image (image features). Some image processing uses basic statistic formula such as : histogram, mean, variance, etc. In embedded environment those formula must be implemented in hardware base, like FPGA for example. A research for histogram calculation for 8x8 image need 256 addition, 320 clock cycle and 256 clock for interface with memory 2KB. [1]

Other research for histogram and mean calculation base on FPGA device is proposed in [2]. This article proposed a method that needs 512 addition operations, 64 cycles, 1 division (using shift right register), no need for memory but using 256 register 8 bit for 8x8 image size.

Another research for histogram calculation is proposed in [3]. This approach use RAM subcel and FSM for controlling, this approach use 850 logic element of FPGA and can work at more than 100 MHz of frequency clocks and able to process 260 frames per seconds.

Another approach is called sum and difference histograms (SDHs) for a dense texture analysis algorithm. The implementation of this approach into a reconfigurable architecture needs 73,021 logic elements, required memory is 737,152 bits, and the number of embedded multipliers is 480 [4].

This article proposes a more efficient method for mean value calculation of the image intensity and its implementation on a FPGA. First section describes the state of the art of the same research. Second section is about basic statistic formula for image processing. Third section is proposed algorithm in Matlab. Fourth section is about the efficient algorithm implementation base on FPGA device. Finally we conclude this article.

II. BASIC IMAGE PROCCESSING FORMULA

A. Image Histogram

Image histogram is a diagram that draw frequency of the appearance every intensity value from the whole image pixel element. [5][6]. The higher value of histogram show the number of pixels with that intensity value is high and vice versa. Histogram can show the brightness and contrast of the image. A lot of use histogram in image texture analysis because of the simplicity of the algorithm. Histogram of an image with NxM pixels can be calculated mathematicaly as :

$$H(i) = \sum_{n=0}^{N} \sum_{m=0}^{M} k, \qquad (1)$$

where $k=1$ if $f(n,m)=i$ and $k=0$ if $f(n,m)\neq i$, while $f(n,m)$ is intensity value of the pixel at coordinate (n,m) in the image. Fig 1. shows some histogram diagram from several image with their own texture. The images are obtain from http://www.freeimages.com/search/texture

B. Mean Formula

From the histogram value we can calculate the 'mean' value from the image. Mathematical formula for 'mean' (μ) is : [5]

$$\mu = \sum_{i=0}^{L} i.p(i), \quad where \quad p(i) = \frac{H(i)}{NM} \qquad (2)$$

In this formula i is grey level of pixel intensity in the image and $p(i)$ is the probability of occurrence i. $L=255$ is the higher value of grey level i. This formula will produce an average brightness of objects.

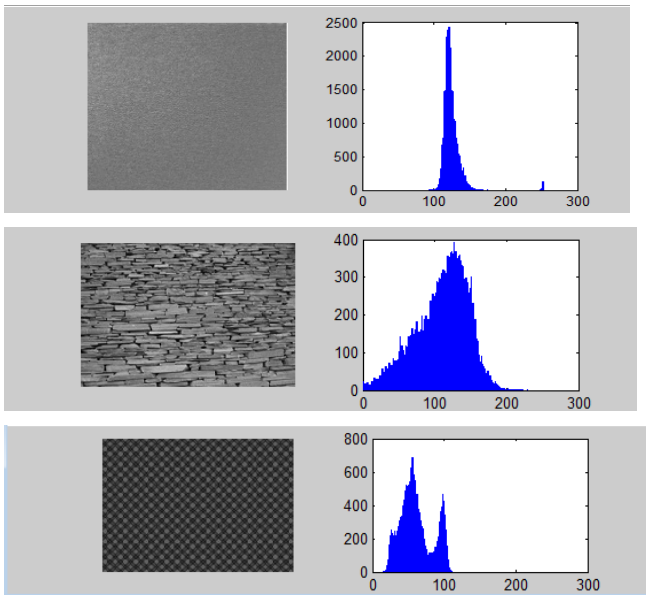


Fig. 1. Example of a histogram diagram comming from several image.

III. PROPOSED AGORITHM

Mean is the first-order statistical analysis methods used for segmentation and feature extraction processes an image. Mean represents the mean value of the intensity of the entire pixel in an image. Referring to the equations (1) and (2), where the mean value of the image intensity is calculated, we can derive the equation 3 that is proposed to be implemented on the FPGA.

$$\mu = \sum_{i=0}^{255} i.P(i) = \frac{1}{NxM} \sum_{n=0}^{N} \sum_{m=0}^{M} f(n,m) \qquad (3)$$

The psudo code of mean calculation based on equation (2) is showed in Psudo Code-1. There are three steps of calculation: the first one is histogram $H(i)$, the second one is probability $p(i)$ and the last one is mean value μ. The algorithm of the mean formula in the equation (3) is presented in Pseudo Code-2. In this algorithm, there is just one step of calculation that has the same complexity as histogram $H(i)$ calculation. It means that the proposed algorithm can reduce two steps of calculation (2 x 256 clock cycles) and some number of arithmetic operations e.g. addition, multiplication and division.

Pseudo code 1. Mean formula (1st version in equations 2)

```
(1)  I1 = % data %
(2)  h=zeros(256,1);
(3)  [N,M]=size(I1);
(4)  for i=1:N                  % Histogram H(i) calculation%
(5)     for j = 1 : M
(6)           h(I1(i,j)+1) = h(I1(i,j)+1)+1;
(7)     end
(8)  end
(9)  N1=(N*M);
(10) for i = 1 : 256            % Probability p(i) calculation%
(11)    p(i) = h(i)/N1;
(12) end
(13) Mean=0;
(14) for i = 1 : 256                  % Mean μ calculation%
(15)    Mean = Mean + p(i) * (i-1);
(16) end
(17) Mean
```

Pseudo code 2. Mean formula (revised version in *equation 3*)

```
(1) [M,N,L]=size(Im); % Im is the data
(2) htsum=0;
(3) for i=1:M;                  % Mean μ calculation%
(4)   for j=1:N;
(5)       htsum=htsum+Im(i,j);
(6)   end;
(7) end;
(8) optimalmean=htsum/(M*N).
```

Im=[6 6 6 6 10 10 8 8; 6 6 48 48 10 10 8 8;
6 6 48 48 10 10 8 8; 6 6 48 48 13 13 10 10
6 6 6 52 13 13 10 10; 5 5 5 52 13 13 10 10
5 5 5 52 4 4 12 12; 5 5 5 52 4 4 12 12].

The results of mean calculation using basic algorithms and optimization algorithms for the same data (Im) are the same, as shown in Figure 2. The values resulting from the evaluation using Matlab will be used for comparison with values that are processed using FPGA components.

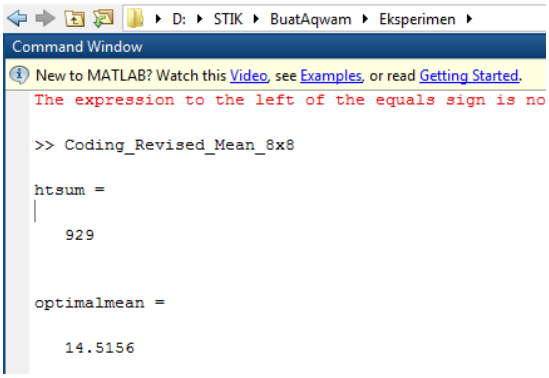


Fig. 2. Results form both algorithm with the same data.

IV. HARDWARE IMPLEMENTATION

A. Component Design

Figure 3 is the first design based on first pseudo code, this design need a selector to select intensity value of pixel from the image between 0 to 255 and send to their own respective accummulator (256 accumulators with 256 addition components). All these accumulator value will be sent to 256 input addition (or another 256 additions) so this approach need 512 additions [2]. And finnally to obtain mean value we divided with the number of pixels wich is 8x8 = 64. We use shift right register 6 bits to do this division operation.

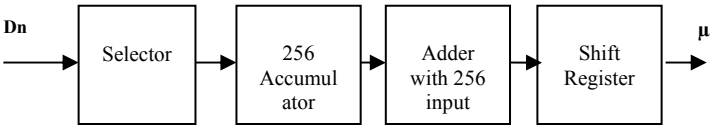


Fig. 3. Component design for pseudo code-1 in [2].

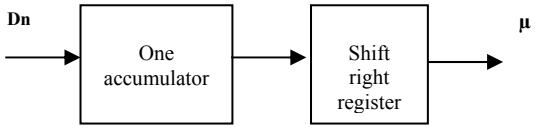


Fig. 4. New Component design for pseudo code-2.

Figure 5 and 6 are the new design based on the design in figure 4, to calculate mean value actually we don't have to calculate histogram, we can directly using one accummulator to hold the total value of pixel element value (htsum) in the 2nd pseudo code. This approach give us the reduction of mathematical operation (component). This design only need one addition in one accummulator and one shift right register, with the same processing time (NxM=64 clock cycles) for 8x8 image size and the same result of mean value. In this design we use counter and buffer to send the final value of mean and htsum after 64 clocks.

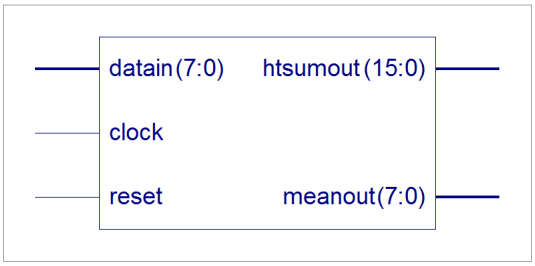


Fig. 5. Entity Diagram of efficient component

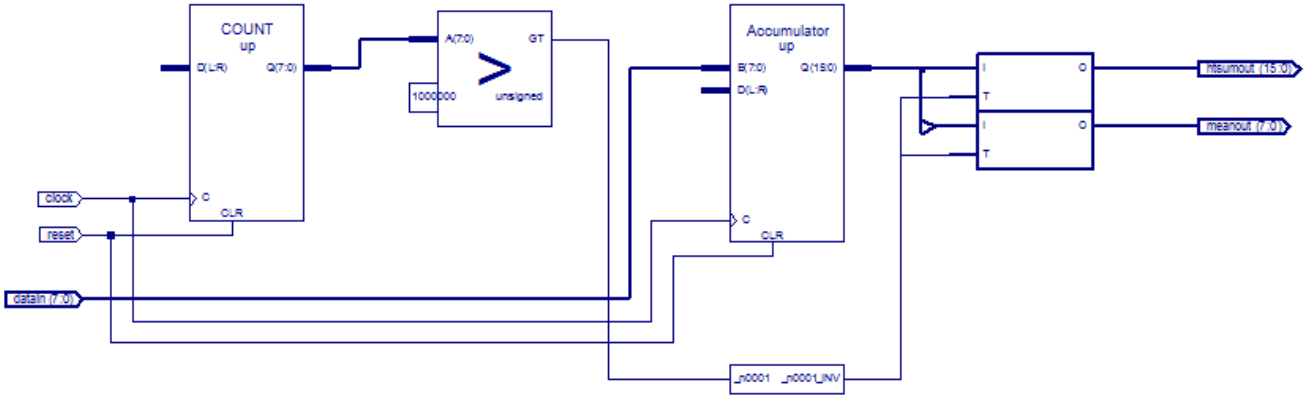


Fig. 6. RTL Schematic Diagram of efficient component.

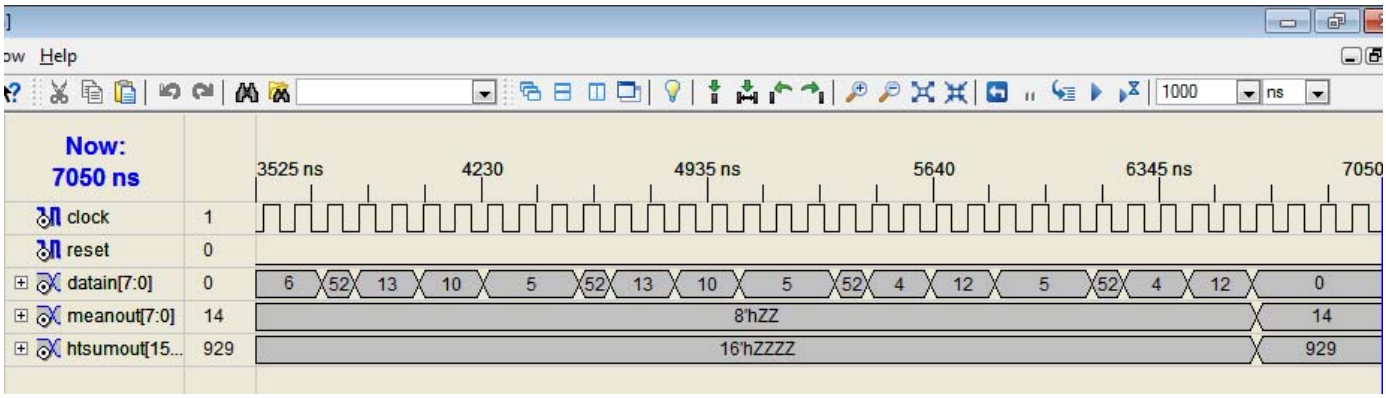
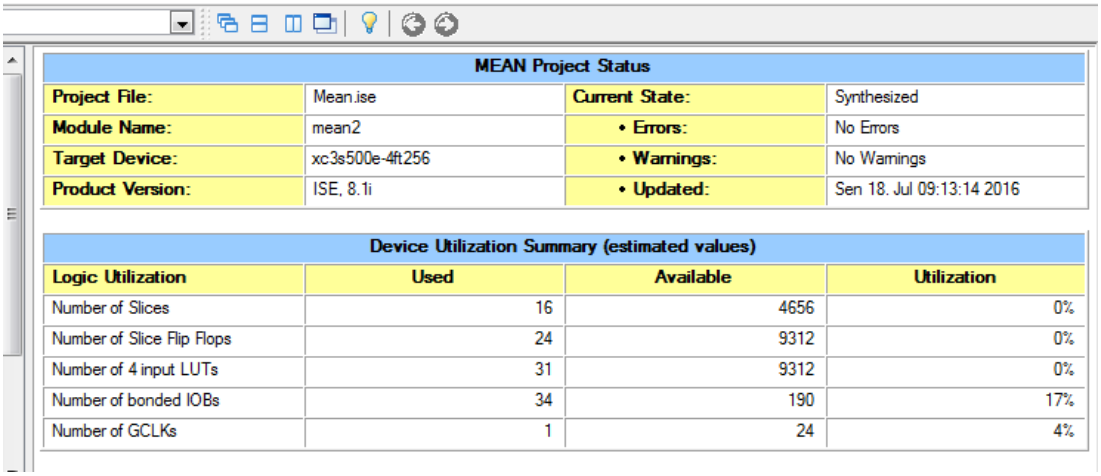


Fig. 7. Behavioral simulation result for proposed component



MEAN Project Status			
Project File:	Mean.ise	Current State:	Synthesized
Module Name:	mean2	• Errors:	No Errors
Target Device:	xc3s500e-4ft256	• Warnings:	No Warnings
Product Version:	ISE, 8.1i	• Updated:	Sen 18. Jul 09:13:14 2016

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	16	4656	0%
Number of Slice Flip Flops	24	9312	0%
Number of 4 input LUTs	31	9312	0%
Number of bonded IOBs	34	190	17%
Number of GCLKs	1	24	4%

Fig. 8. Design summary of the component including counter and buffer.

B. Simulation

Figure 7 show the simulation result based on behavioral simulation. This figure shows the result for the same data which is 14 (mean) and 929 (htsum), at the end of 64 clock cycles, comparing to the Matlab evaluation result wich is 14,5 the difference is 0,5 or an error of 0,005 %. The speed of process is paralel with the comming of data input (data in). From figure 8 we can see the design summary of the componen that provide the information of logic element accupation in FPGA device.

V. CONCLUSSION

The efficient implementation of mean formula into hardware base compponent using FPGA device has been proposed. This component use one addition and one shif right register with 64 clock cycles to calculate mean value for 8x8 image size. This design (with counter and buffer) needs 16 slices of flip-flops and 31 of 4 input LUTs. The difference (as an error) of mean value commparing to Matlab result is 0,5 or 0,005 % due to floating point problem in FPGA.

ACKNOWLEDGMENT

Thanks to Yayasan Pendidikan Gunadarma and Yayasan Ilmu Komputer for supporting this research as one part of research dissertation in Doctoral Program of Gunadarma University, Jakarta-Indonesia.

REFERENCES

[1] Nitin Sachdeva and Tarun Sachdeva, " An FPGA Based Real-time Histogram Equalization Circuit for Image Enhancement", International Journal of Electronics and Communication Technology (IJECT) Vol 1 Issue 1 December 2010.

[2] Atit Pertiwi, "Optimalisasi Metode Implementasi Algoritma Histogram, Mean Dan Variansi Ke Dalam IC-FPGA Untuk Aplikasi Analisis Tekstur Citra Real Time", Disertasi, Teknologi Informasi, Universitas Gunadarma, 2015.

[3] Luca Maggiani, Claudio Salvadori, Matteo Petracca, Paolo Pagano, Roberto Saletti, " Reconfigurable Archictecture for Computing Histograms in Real-Time Tailored to FPGA-based Smart Camera", IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istambul Turkey, June 2014.

[4] M.-A. Ibarra-Manzano, D.-L. Almanza-Ojeda and J.-M. López-Hernández, "Design and Optimization of Real-Time Texture Analysis using Sum and Difference Histograms Implemented on an FPGA", 2010 Electronics, Robotics and Automotive Mechanics Conference, Mexico, 2010.

[5] Wendy L. Martinez and Angel R. Martinez, "Computational Statistics Handbook with MATLAB", Chapman & Hall/CRC, USA, 2002.

[6] R.E. Woods , S.L. Eddins , dan R.C. Gonzales, "Digital Image Processing Using MATLAB . Pearson Educatio, 2005.