

Design and Implementation of Efficient Histogram Equalization in FPGA

M.C Prabavathi ¹, Padmaja Jain ², Sachin S Munji ³

¹ M.Tech student, BNM Institute of technology,

² Asst. prof. BNM Institute of technology,

³ R&D engineer, SaiTektronics Pvt. Ltd.

Abstract - The complexity of any picture is principally relying on the light control exhibit in the photo and is in an indirect way identified with the nature of a picture. This is on the grounds that if any picture has low light power then it shows up as the dull picture and a high power picture gives off an impression of being a brilliant picture. In the two cases, it is hard to make sense of the articles display in that picture. Along these lines, to make awful quality picture to great quality picture separate renovation procedures are ordinarily utilized. However, the principle issue of those strategies is the figuring of Threshold value. In this project, we propose and execute FPGA design for differentiate improvement utilizing adaptive threshold. The execution of the proposed design is superior to existing strategies as far as both equipment usages and yield picture precision with no noise using Gaussian filter.

Key Words: contrast, light power, picture, adaptive threshold, FPGA, Gaussian filter.

1. INTRODUCTION

Histogram Equalization (HE) is an alternate form to represent an image. It can be range from '0' to 'L-1' where L is range of pixel value for gray image it is 256. Histogram is a simple plot of number of pixel with brightness value of pixel. It simply counts the pixel intensity value and pixel probability. HE has two approaches global (GHE) and local or Adaptive histogram equalization (AHE). In GHE the histogram of entire image is considered. It is nothing but simple HE obtains of image. Hence it requires more hardware implementation with poor image quality. We are designing an efficient Histogram Equalization that has many applications in automotive applications.

2. PROPOSED ENHANCEMENT ARCHITECTURE

The proposed architecture for contrast enhancement is shown in the Figure 1. The histogram is calculated from the input image. Depending upon the pattern of histogram the decision making block will decide that the contrast level of the input image i.e. high contrast or low contrast. This will be used by linear stretching block to stretch the contrast to acceptable level. Then the enhanced image pixel is mapped to a certain boundary by mapping block.

2.1. Preprocessing block

In this block the input image is fed, the color image is converted by gray to simplify the hardware requirements and finally resized to [256*256].

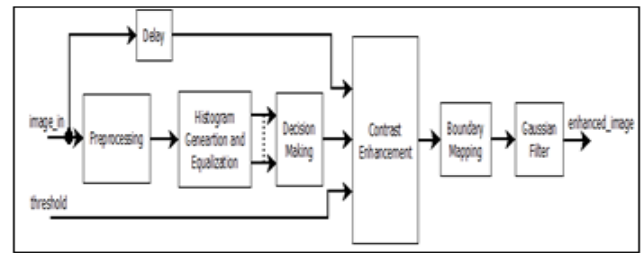


Fig 1: Proposed Model for contrast enhancement

2.2. Histogram Generation and Equalization

The histogram will give the frequency of occurrence of each pixel. This will help the system to understand the intensity level of the input image. To implement this block we use basic comparator and counter as shown in the Figure 2. In normal process we have to check the histogram of all pixel intensity range (i.e. 0 to 255). But this requires a large amount of hardware resources. To reduce the resource utilizations without affecting the output accuracy we consider only the pixel intensity range from first and last value up to 20 offset. Because most of the bad quality image histogram is higher in this range.

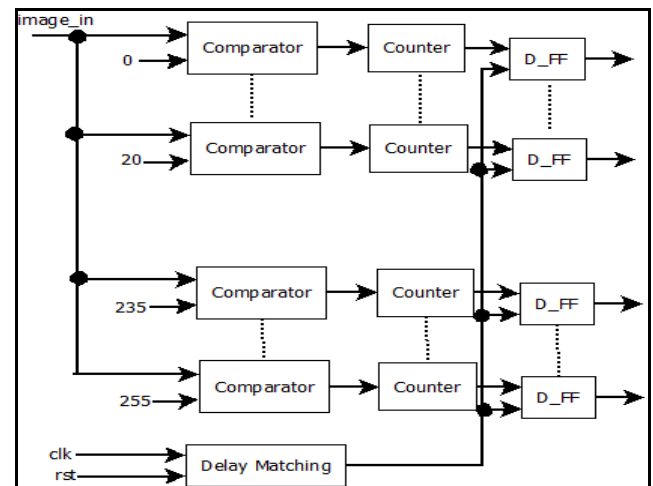


Fig 2: Proposed Histogram generation and equalization

2.3. Decision Making

Depending upon histogram values, decision is made for the contrast of the input image. If the contrast is shifted '0' value then the image having low contrast. Similarly, if the contrast is shifted towards '255' then it's having high contrast. The contrast decision is shown in Figure 3.

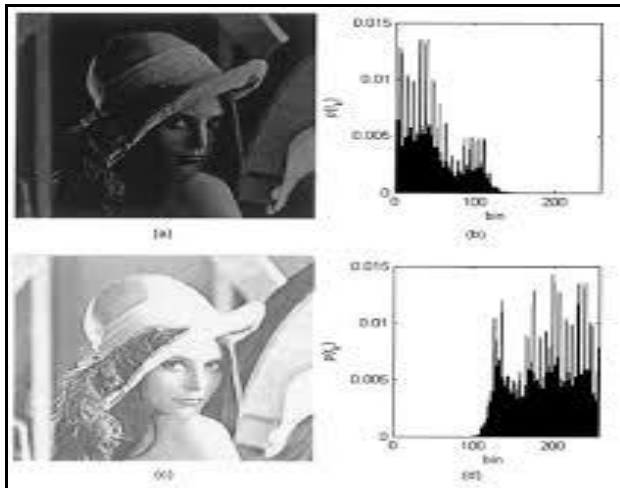


Fig 3: Contrast Decision

2.4 Contrast Enhancement

To enhance the contrast of the image we have dynamically calculated the threshold value using Mean Squared Error (MSE) as shown in equation (1),

$$MSE = \frac{(A_0 - B_0)^2 + (A_1 - B_1)^2 + \dots + (A_N - B_N)^2}{8 * N * N} \dots (1)$$

Since we are not comparing two images hence only equation (2) remains, where 'N' = 256 and 'A' is pixel values.

$$MSE = \frac{(A_0)^2 + (A_1)^2 + (A_2)^2 \dots + (A_N)^2}{8 * N * N} \dots (2)$$

RTL view of Intended threshold calculator is shown in Figure 4. Equation (1) and (2) are used in sensitivity block and the code used for comparator is shown in figure 5 and D- flip flop is used as a latch.

The equation for linear stretching is given by

$$\text{Output image pixel} = \text{input image pixel} \pm \gamma \dots (3)$$

Where, $\gamma \rightarrow$ constant, depend upon the level of enhancement.

2.5 Boundary Mapping

After linear stretchy there are high possibilities of getting pixel mapping block is used to map all pixel values into corresponding boundary for proper display purpose. In the proposed method we consider the gray pixel which can vary from 0 to 255 so the equation for boundary mapping function is given by the equation (4).

$$\text{mapped pixel value} = \begin{cases} \text{input pixel} & ; \text{if } 0 > \text{input pixel} < 255 \\ 255 & ; \text{if } \text{input pixel} > 255 \\ 0 & ; \text{if } \text{input pixel} < 0 \end{cases} \dots (4)$$

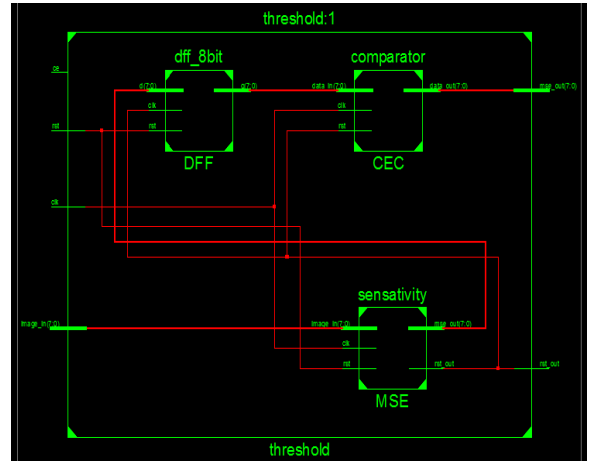


Fig 4: RTL view of Designed threshold.

```

if (rst='1') then
if (clk'event and clk='1') then
data_int:=conv_integer(data_in);
if (data_int<60) then
data_out<=data_in;
elseif (data_int>=60 and data_int<100) then
data_out<=conv_std_logic_vector(100,8)-data_in;
elseif (data_int>=100 and data_int<150) then
data_out<=conv_std_logic_vector(150,8)-data_in;
--data_out<='0' & data_in(7 downto 1);
elseif (data_int>=150 and data_int<200) then
data_out<=conv_std_logic_vector(200,8)-data_in;
--data_out<="00" & data_in(7 downto 2);
elseif (data_int>=200 and data_int<250) then
data_out<=conv_std_logic_vector(250,8)-data_in;
--data_out<="000" & data_in(7 downto 3);
else--if (data_int>=250) then
data_out<=conv_std_logic_vector(255,8)-data_in;
end if;
end if;
else data_out<=(others=>'0');
end if;

```

Fig 5: Designed code used for comparator.

2.6 Gaussian Filter

After mapping the enhanced image into fixed boundary using equation (4), many of high frequency component appears in an image is considered as Noise. To remove that high frequency component from the enhanced image Gaussian filters is used. For good clarity we use 3X3 Gaussian mask, which is given by equation (5):

$$\text{output image pixel} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \dots (5)$$

Where, 'a' to 'i' are the 3x3 sub-matrix of the corresponding image.

III. FPGA IMPLEMENTATIONS

In this chapter we discuss the implementation results on software and hardware, the coding is done on Xilinx 14.5v and given to system generator for verification. For this implementation on hardware we use ATLYS FPGA Board (xc6slx45-2csg324) as shown in Figure 6.

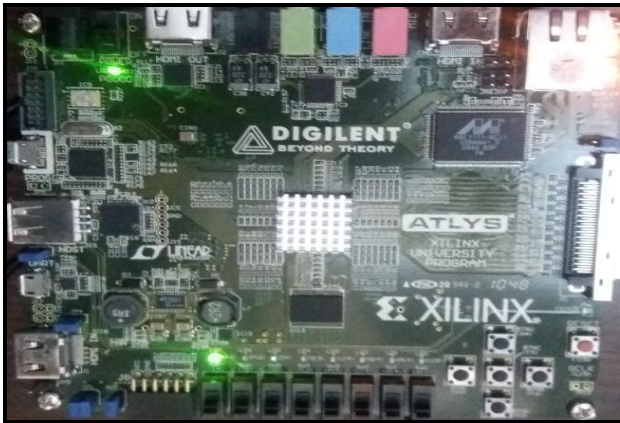


Fig 6: ATLYS FPGA Board

The simulink display (programming) for proposed design is appeared in Figure 7 and Hardware demonstrate is appeared in Figure 8.

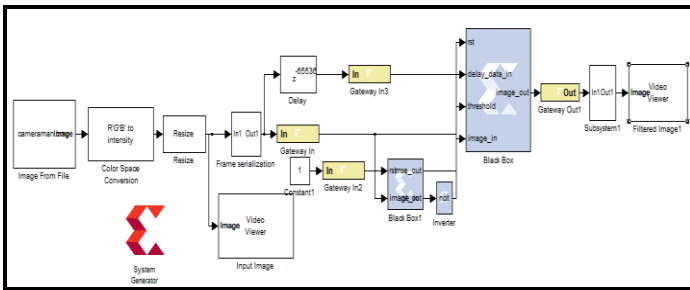


Fig 7: Software Model view in system generator.

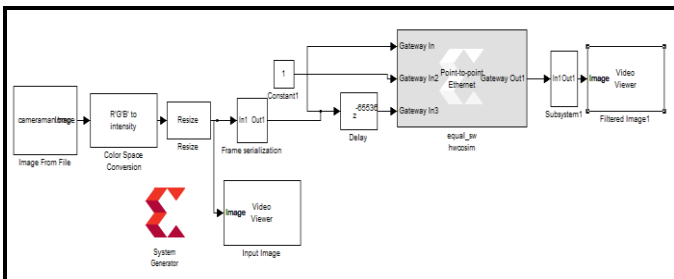


Fig 8: Hardware Model view in system generator.

The Figure 9 shows the output image for a given input dark image, the software results and hardware results are the same and the compilation status is given in figure 10.

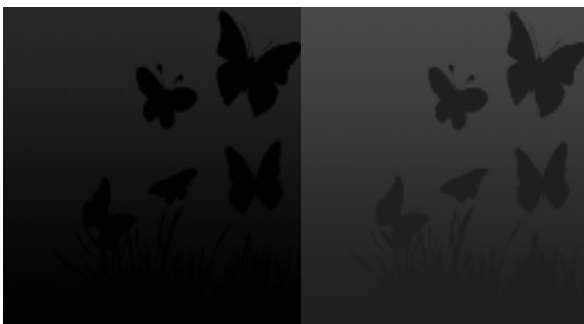


Fig 9: Input Image and output Image.

The proposed technique with existing techniques is specified in the table 1 below and found that proposed method is better than existing in terms of hardware utilization and hence the area is reduced with increased speed.

Table 1 Comparisons with existing techniques

Parameters	Pratik and Nitin et al.[10]	Hanumantharaju et al. [11]	Burak Unal et al.[9]	Proposed Method
Slice Registers	3206	1867	440	1416
Fully Used FF-Pairs	2809	4096	4766	1360
Total	6015	5963	5206	2776

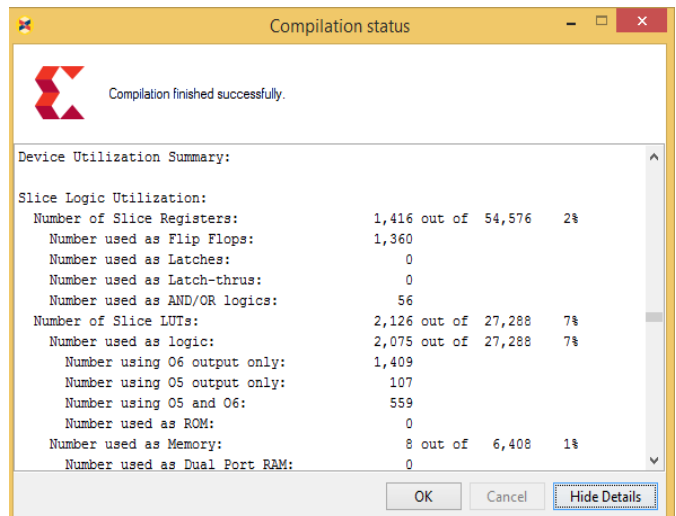


Fig 10: Compilation results

3. CONCLUSION

In this paper, an algorithm that is suitable for implementation on FPGA Spartan 6 board for image enhancement. In VLSI the area is very important when it comes to implementation point of view. The proposed method start with Acquire input image .Then the converting color image to gray image. The Histogram is generated for the gray image. The Decision unit helps to know the contrast of original image. If low contrast then the contrast of an image is increased. Hence the execution of the proposed design is superior to existing strategies as far as both equipment usages and yield picture precision with reduced area which is shown in compilation status and hence the operational speed is increased.

REFERENCES

- [1] Ramyashree B H, Mrs. Vidhya R.et “FPGA IMPLEMENTATION OF CONTRAST STRETCHING FOR IMAGE ENHANCEMENT USING SYSTEM GENERATOR” IEEE: 2325-9418, March 2016
- [2] Vidyasaraswathi H N, Dr. M.C.Hanumantharaju “REVIEW OF VARIOUS HISTOGRAM BASED MEDICAL IMAGE ENHANCEMENT TECHNIQUES”, 2015.

-
- [3] M. Abdullah-Al-Wadud, Md. Hasanul Kabir, M. Ali Akber Dewan, and Oksam Chae, Member, IEEE "A Dynamic Histogram Equalization for Image Contrast Enhancement", 2007 IEEE.
- [4] Shelda Sajeev, Mariusz Bajger and Gobert Lee Medical Device Research Institute "Segmentation of Breast Masses in Local Dense Background using Adaptive Clip Limit-CLAHE, 2015 IEEE
- [5] Phillip David Ferguson ate, Evaluation Of Contrast Limited Adaptive Histogram Equalization (CLAHE) Enhancement on a FPGA, 2008 IEEE
- [6] Anmol Sharma, Jagroop Singh "Image Denoising using Spatial Domain Filters: A Quantitative Study" 2013 6th International Congress on Image and Signal Processing (CISP 2013).
- [7] Naoki Nakajima and Akira Taguchi "A Novel Color Image Processing Scheme in HSI Color Space with Negative Image Processing" 2014 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) December 1-4, 2014.
- [8] Mohammad I. AlAli, Khaldoon M. Mhaidat, and Inad A. Aljarrah "Implementing Image Processing Algorithms in FPGA Hardware" 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT).
- [9] Burak Unal,, Ali Akoglu "Resource Efficient Real-Time Processing of Contrast Limited Adaptive Histogram Equalization".
- [10] Pratik G. Jane and Nitin P. Narkhede, "Image Enhancement Algorithm Implemented on Reconfigurable Hardware", International Journal of Computer Applications, pp. 1-3, 2014.
- [11] H. C. Hanumantharaju, M. Ravishankar, D. R. Rameshbabu and S. Ramchandran, "A Novel FPGA Implementation of Adaptive Color Image Enhancement Based on HSV Color Space", IEEE Conference, pp. 160-163, 2011.
- [12] Sateesh Kumar H C, Sayantam Sarkar, Satish S Bhairannawar, Raja K B and Venugopal K R, "FPGA Implementation of Moving Object and Face Detection using Adaptive threshold", VLSICS, pp. 15-25, 2015.