Hardware Architecture of High Speed HEQ for Image Enhancement

Kibum Suh

Abstract: ***In this paper, the hardware architecture of high speed HEQ, which can process full HD image and can be incorporated in small size FPGA, is proposed. In order to verify the efficiency of the proposed architecture, a reference C was constructed and compared with other algorithms. The proposed architecture has similar performance compared with previous existing algorithm and can process image enhancement in high speed. Synthesized FPGA logic has minimum 6.25ns period and can process full HD image sequence at 60frame/sec. The SRAM used is 10752 bits and the number of slice LUT is 119 out of 4420, which is the number of slice LUT for XC6SL9. Unlike the retinex algorithm, which requires pay royalties, the designed modules can be used without royalties, and can be easily ported to small FPGAs. The developed module can be used as a component of a video camera in a DVR system or a surveillance system.***

Index Terms: ***image enhancement, Retinex , iridix , small size FPGA, high resolution image***

I. INTRODUCTION

Today's law enforcement departments, government and private business owners benefit from state-of-the-art surveillance camera products that can provide incredible image quality with valuable information. In order to improve image quality at high resolution, high-speed processing is required. The most widely used image enhancement algorithms are Arm's iridix algorithm [1], NASA's Retinex algorithm [2,3], and histogram equalization algorithm [4,5,6]. ALTERA uses Cyclone FPGA and implements the iridix algorithm using APTINA camera sensor to enhance the dynamic range of the pixel, thus realizing the camera system with the same performance as the human eye. Quick logic also sells reference designs using iridix algorithms. The iridix method is a private resource of ARM. ARM sells ISP solutions (Mali-C32, C52, C71) with iridix algorithm named MALI Camera[1]. For this solution, source code is not open, but the reference board and the binary code are provided. When commercializing it, the license fee must be paid. Nowadays, several papers have been proposed for porting image enhancement algorithms to low cost FPGAs, and there are Trends to implement by avoiding ARM iridix algorithm[7,8,9]. Also, methods for image enhancement method using multi-GPU are proposed[10,11]. In this paper, we compare the performance of existing image enhancement algorithms, develop a hardware structure suitable for small size FPGAs, and develop low cost circuits suitable for full HD images.

The Comparison of Image Enhancement Algorithms

The Reinex algorithm is based on the fact that NASA's Land and McCann have a logarithmic relationship between the brightness of the image and the visually perceived sensation and that the brightness of the image is given by the product of the actual brightness. It is a method to increase the contrast of the image by reducing the component of the illumination in the image based on the fact and showing only the component of the reflection.

$$I(x,y)=L(x,y) \times R(x,y) \qquad (1)$$

where $I(x,y)$ is observed image.
$R(x,y)$is percieved reflectance.
$L(x,y)$ is percieved illumination.

In this case,

$$\mathrm{Log}\{R(x,y)\} = \log\{I(x,y)\} - \log\{L(x,y)\} \qquad (2)$$

It is possible to obtain a better image by using the image only as the reflection component as the output. Therefore, in order to obtain the illumination component, only the image component at a large scale which does not depend on the detail of the image is viewed (low frequency), and the illumination component can be obtained from the image obtained by Gaussian blurring the original image with an appropriate scale. In the case of applying the multi-scale, the reflection component in the Retinex algorithm is

$$St[i]=\{\log(src[i])-\log((gaussian\ fileter(s)*src)[i]) \qquad (3)$$

As shown in figure 1, the histogram equalization algorithm obtains the cumulative distribution function (CDF) of the luminance signal and adjusts the luminance component so that the CDF is linear[12]. Consider the example below. The histogram before smoothing is densely distributed between 100 and 200. After smoothing, small values of histogram are distributed densely and large values are distributed widely, and CDF becomes linear.

In this paper, we compared the retinex algorithm of NASA with the HEQ method using own model, selected HEQ method, and proposed HEQ hardware architecture which can be ported to a small FPGA.

Revised Manuscript Received on May 22, 2019

Kibum Suh, Dept. of Railway electrical system, Woosong University, Daejeon, Korea.

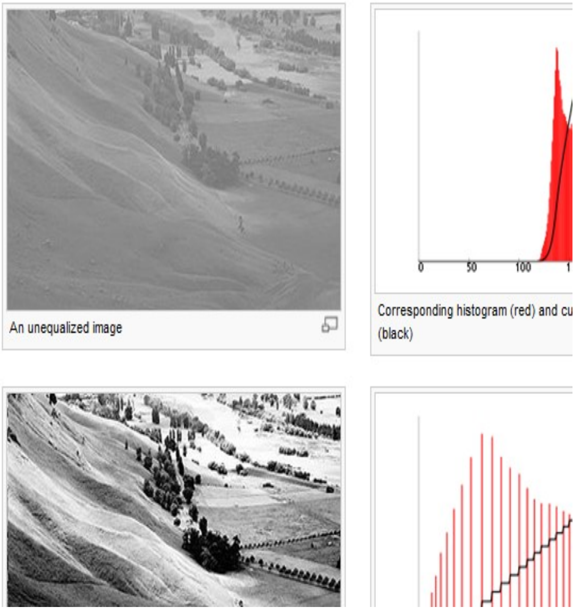


Figure 1. Histogram Equalization[12]

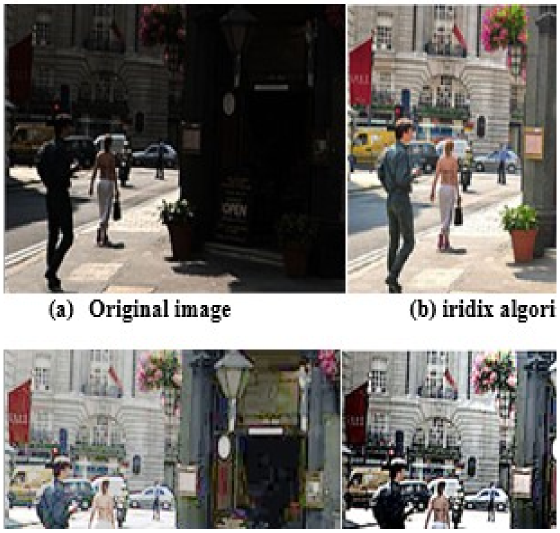

(a) Original image (b) iridix algori

Figure 2.Comparison of histogram smoothing and other algorithms

Figure 2 compares the HEQ method developed in this study with the Retinex[2,3] and iridix algorithms[1]. In the figure, reference images are provided to illustrate the usefulness of the iridix algorithm[1]. Figure 2 shows that the iridix image shows the best results. Of course, this may be the best matching image for the iridix algorithm. To avoid patent of the iridix algorithm, the iridix algorithm was excluded from the selection. Therefore, we selected efficient one of NASA's retinex algorithm and HEQ method, and proposed an efficient hardware structure for porting to a small FPGA.

Therefore, we implemented the retinex algorithm of NASA and experimented with various images, and it is confirmed that HEQ has similar image enhancement effect. Figure 3 shows the results of the case of Retinex and the method of HEQ to identify the parking lot of the night image. From the original image of the parking lot line, the results of the Retinex algorithm and the HEQ algorithm can detect the parking lot line to the same extent.

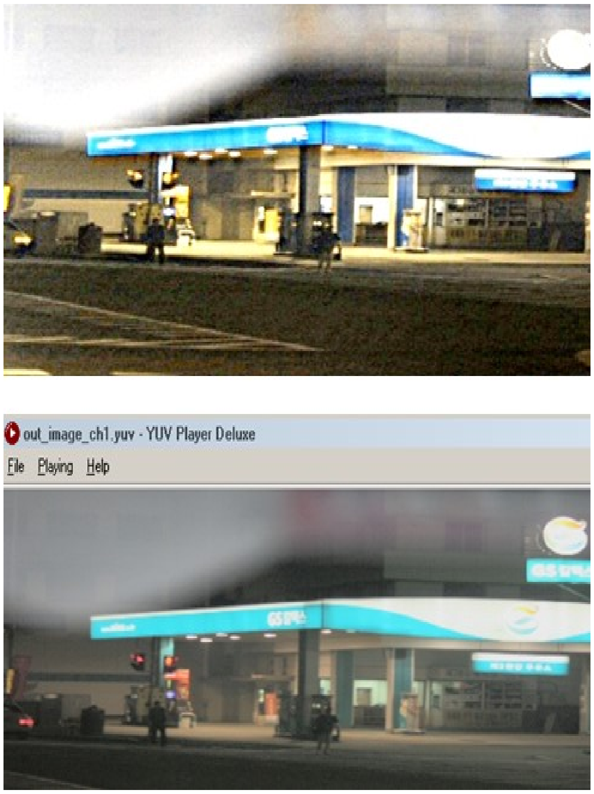
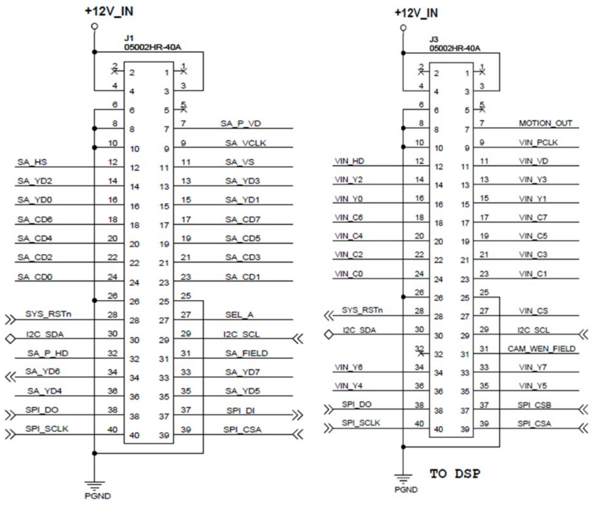


Figure 3.Comparison of Histogram Equalization and Retinex Algorithm
(a) Retinex algorithm results
(b) Histogram Equalization results using our reference C

II. The proposed Hardware architecture

A. input output interface

The input/output interface of the development board is shown in figure 4. In Figure 4 (a), the camera interface is used as the input of the circuit, and the output is sent to the DSP as shown in Figure 4 (b) for H.264 encoding. There are four types of signals output from the sensor. Each name consists of PCLK, LINE_VALID, FRAME_VALID, DATA. FRAME_VALID indicates a period during which one frame of the image sensor is being input from the image sensor when the flag is 'HIGH', and Vertical Blanking interval during the period when the flag is 'LOW'. Likewise, when LINE_VALID is 'HIGH', it means a pixel input period of one line, and when it is 'LOW', it means a horizontal blanking interval. In the interval in which FRAME_VALID is valid, the LINE_VALID signal becomes 'HIGH' by the number of lines, and pixel values are input from the image sensor in synchronization with PCLK in this interval



(a)Camera interface (b)DSP interface

Figure 4. Input/output Interface

B. Histogram equalization hardware architecture

To perform histogram equalization, the probability density function (pdf) and cumulative distribution function (cdf) must be calculated. In the figure 5, the pdf calculation part uses the dual-port SRAM to obtain the pdf value, and the cdf calculation part uses the single-port SRAM to obtain the cdf value. Cdf value is calculated as shown in equation(4), where H and V are the horizontal and vertical pixels of the image.

$$cdf = \left[\frac{acc \times 255}{H \times V} + 0.5 \right] \qquad (4)$$

For the calculation time of PDF and CDF, PDF is obtained in a section where FRAME VALID is 'HIGH' in the Nth frame, and CDF is calculated in a section where FRAME VALID is 'LOW', using the equation (1), as shown in figure 6. The histogram equalized pixel value outputs the pixel value with the calculated cdf result value during the Nth frame stored in the SRAM corresponding to the address of the input pixel value.

Also, the pdf and cdf result values calculated in the (N + 1)th frame are applied to the (N + 2)th frame.

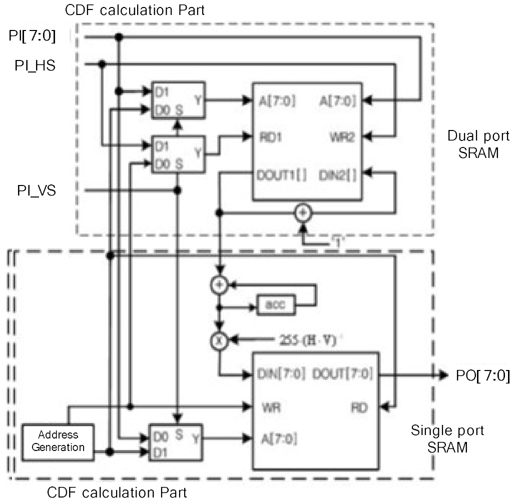


Figure 5. The proposed architecture of Histogram Equalization

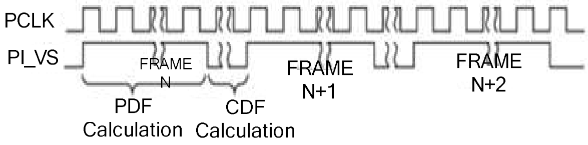


Figure 6. PDF calculation and CDF calculation time

C. Pdf calculation hardware

The pdf is obtained using the dual-port SRAM as shown in figure 7. Since pdf is the pixel frequency and the pixel is valid when PI_VS is 'HIGH', read and write operation is performed during PI_VS is 'High' using dual-port SRAM to obtain pdf value. In the above configuration, the read command is issued one clock before the write command.

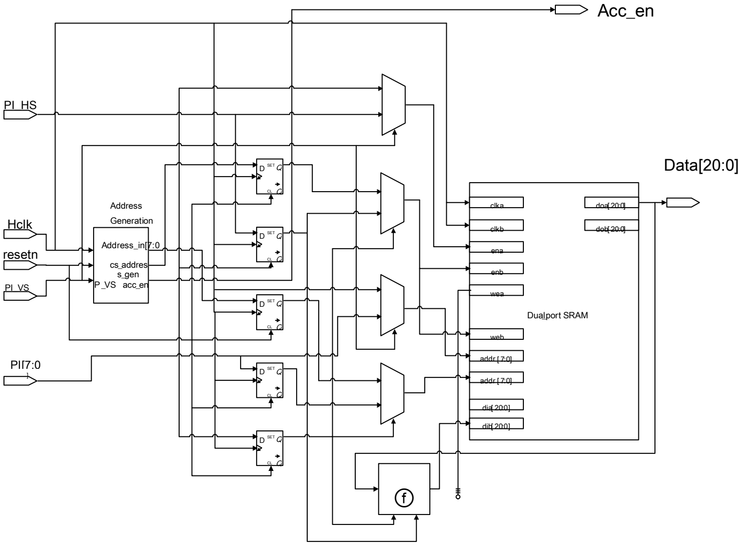


Figure 7. PDF calculation Module

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

D. Cdf calculation hardware

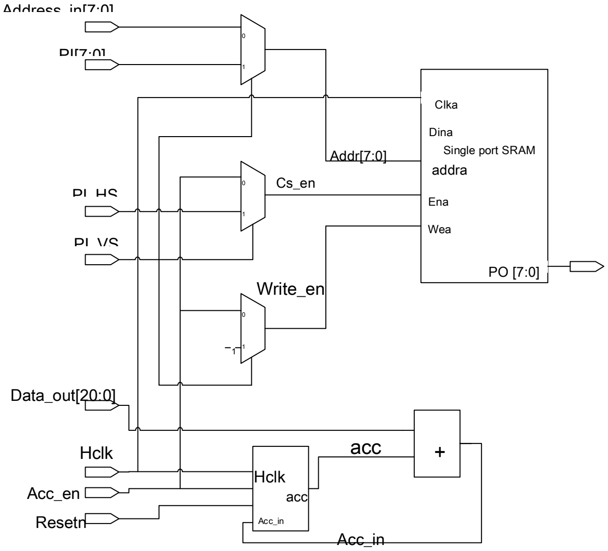


Figure 8. CDF calculation Module

Figure 8 shows the CDF calculation module. It calculates cumulative density function when PI_VS is low. As shown in Figure 9, when write signal PI_VS is 'LOW', cdf value is used as enable signal of single port SRAM. In the timing diagram as shown in figure 9, the addr is sequentially increased from the minimum brightness value (0) to the maximum brightness value (255) of the pixel, and the cdf value calculated by acc is written to the single-port SRAM for each addr.

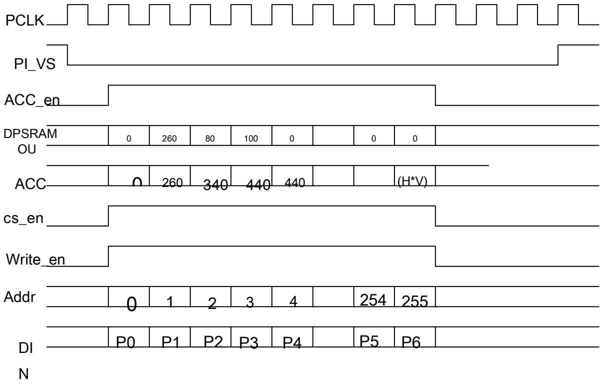


Figure 9. CDF calculation timing Sequence

E. Support format

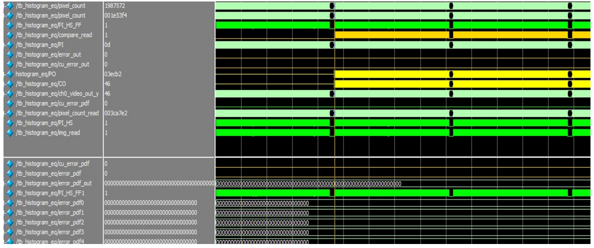
When calculating cdf, we calculate the bit width according to the resolution of the luminance signal, and calculate the cdf calculation result using the ROM table so that the multiplier and divider are not used. Supported formats and bit width need for CDF calculation are shown in Table 1.

Table 1.Supported format and bit width needed for CDF calculation

resolution	Width	Height	Samples	Bit width
720x240	720	240	172800	18
720x288	720	288	207360	18
720x480	720	480	345600	19
960x240	960	240	230400	18
960x288	960	288	276480	19
1280x720	1280	720	921600	20
1920x1080	1920	1080	2073600	21

III. Simulation Results and Discussion

In the case of the simulation, the test sequences for the seven image sizes shown in Table 1 were tested for 300 frames and no errors were generated in the simulation. We checked the error by checking the error of the pdf calculation and the error of the output. For comparison of the errors, Figure 10 shows two error values for the full HD sequence. Error_out represents the error value of the output, and error_pdf represents the error of pdf value at the end of one frame. Figure 10 shows that there is no error in the 4th frame.



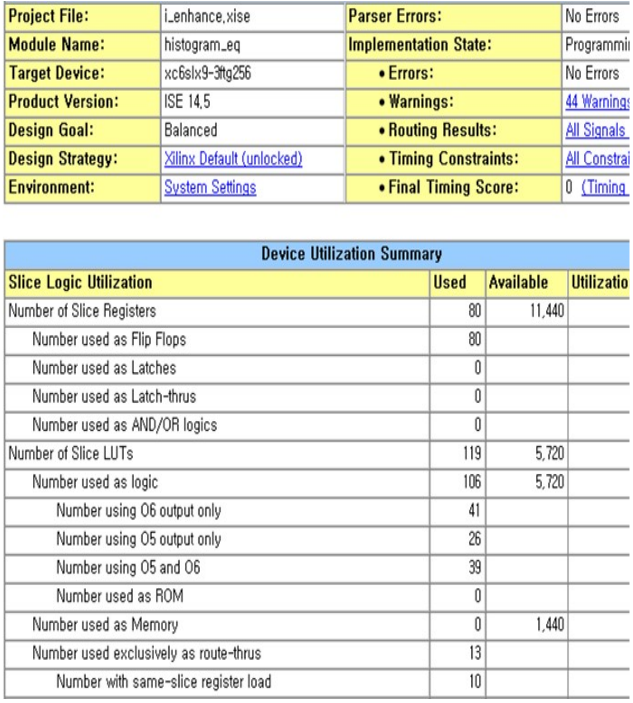
```
/tb_histogram_eq/pixel_count          1987572
/tb_histogram_eq/pixel_count          001e53f4
/tb_histogram_eq/PI_HS_FF             1
/tb_histogram_eq/compare_read         1
/tb_histogram_eq/PI                   0d
/tb_histogram_eq/error_out            0
/tb_histogram_eq/cu_error_out         0
histogram_eq/PO                       03ecb2
/tb_histogram_eq/CO                   46
/tb_histogram_eq/ch0_video_out_y      46
/tb_histogram_eq/cu_error_pdf         0
/tb_histogram_eq/pixel_count_read     003ca7e2
/tb_histogram_eq/PI_HS                1
/tb_histogram_eq/img_read             1

/tb_histogram_eq/cu_error_pdf         0
/tb_histogram_eq/error_pdf            0
/tb_histogram_eq/error_pdf_out        000000000000000000000000000000000000
/tb_histogram_eq/PI_HS_FF1            1
/tb_histogram_eq/error_pdf0           00000000000000000000000000000000
/tb_histogram_eq/error_pdf1           00000000000000000000000000000000
/tb_histogram_eq/error_pdf2           00000000000000000000000000000000
/tb_histogram_eq/error_pdf3           00000000000000000000000000000000
/tb_histogram_eq/error_pdf4           00000000000000000000000000000000
```

Figure 10. Simulation results for full HD sequence

The selected FPGA utilizes the Xilinx® XC6SLX9, the lowest-end class of the Spartan-6 series. Figure 11 shows the board made with the camera. This board performs the image enhancement by the FPGA, and the DSP processes the image with the H.264 codec and transfers it through the LAN. The SRAM used is 10752 bits and the number of slice LUT is 119 out of 4420, which is the number of slice LUT for XC6SL9. Table 2 shows the synthesis result using Xilnix ISE. The result shows that the proposed architecture is designed to use about 2% of the low cost FPGA XC6SL9, and the utilization of memory is less than 10%.

Table 2.The synthesized results of proposed design



Project File:	i_enhance.xise	Parser Errors:	No Errors
Module Name:	histogram_eq	Implementation State:	Programmi
Target Device:	xc6slx9-3ftg256	• Errors:	No Errors
Product Version:	ISE 14.5	• Warnings:	44 Warning
Design Goal:	Balanced	• Routing Results:	All Signals
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constrai
Environment:	System Settings	• Final Timing Score:	0 (Timing

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilizatio
Number of Slice Registers	80	11,440	
Number used as Flip Flops	80		
Number used as Latches	0		
Number used as Latch-thrus	0		
Number used as AND/OR logics	0		
Number of Slice LUTs	119	5,720	
Number used as logic	106	5,720	
Number using O6 output only	41		
Number using O5 output only	26		
Number using O5 and O6	39		
Number used as ROM	0		
Number used as Memory	0	1,440	
Number used exclusively as route-thrus	13		
Number with same-slice register load	10		

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

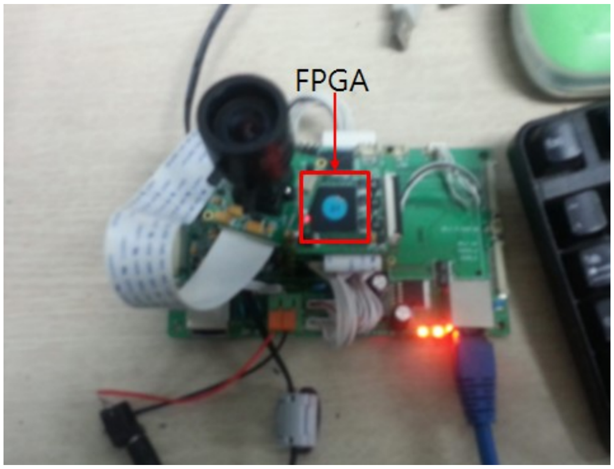
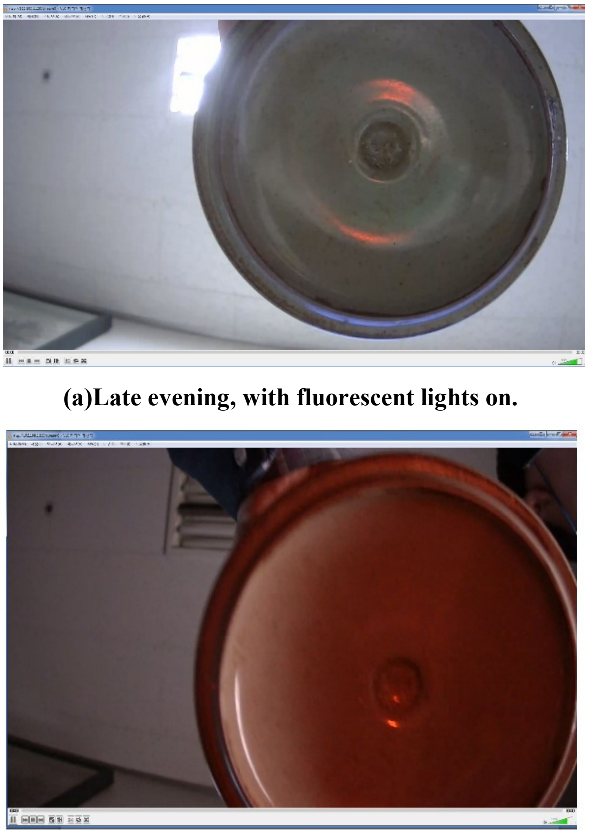


Figure 11. Camera module and FPGA module

In the case of logic synthesis, the synthesis is performed with a period of 8 ns to the VIN_PCLK clock, and the minimum period operates at 6.252 ns, and the operation can be performed at the corresponding period of 13.8 ns at the actual pixel clock frequency 72.5 MHz of image sensor. This corresponds to a period in which full HD image can be processed twice. Figure 12 shows the experimental environment. Experimental environment was experimented by porting image enhancement module to FPGA, experimenting with full HD image input and image enhancement module in FPGA CHIP. In the figure, the output image is processed by the camera with full HD (1920x1080) input and processed through the VLC media player



(a)Late evening, with fluorescent lights on.

(b) Recorded video without light

Figure 12. Experimental Results

Figure 12(a) shows a photograph taken with a fluorescent lamp turned on late. Figure 12 (b) shows the result of processing the image input from the camera when there is no light. Even though the color of the screen changes at this time, it shows a good image.

IV. CONCLUSION

In this paper, we proposed architecture of ultra-high speed HEQ hardware that can process Full HD video with small FPGA. In the synthesized FPGA, the minimum cycle is 6.252 ns and full HD image can be processed in 60 frames. The SRAM used is 10752 bits and the number of slice LUT is 119 out of 4420, which is the number of slice LUT for XC6SL9. The result of the circuit synthesis is designed to use about 2% of the low cost FPGA XC6SL9, and the utilization of memory is less than 10%. The developed hardware could be included in DVR system or surveillance system with low cost FPGA.

ACKNOWLEDGMENT

This research is based on the support of 2018 Woosong University Academic Research Funding.

REFERENCES

1. MALI-C32(Arm Iridix Technology) Available from: https://www.arm.com/products/silicon-ip-multimedia/image-signal-processor/mali-c32?_ga=2.51959849.1437159166.1553733384-665761424.1553733384
2. Z. Rahman, D. Jobson, and G. A. Woodell, “A multi-scale Retinex for bridging the gap between color images and the human observation of scenes”, IEEE Transactions on Image Processing, 1997 July, 6(7): 965-976
3. Marcelo Bertalmío, Vicent Caselles, Edoardo Provenzi (2009) Issues about Retinex Theory and Contrast Enhancement. IJCV, 83 : 101–119.
4. Ji-Hee Han, Sejung Yang, Byung-Uk Lee, "A Novel 3-D Color Histogram Equalization Method with Uniform 1-D Gray Scale Histogram", IEEE Trans. on Image Processing, 2011 Feb. 20(2): 506-512
5. Yun Ho Jung, Jae Seok Kim, Bong Soo Hur, Moon Gi Kang, “Design of real-time image enhancement preprocessor for CMOS image sensor” , IEEE Transactions on Consumer Electronics, 2000 Feb; 46(1): 68-75. DOI: 10.1109/30.826383
6. Reza, A.M. ,“Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement”, The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology 2004 Jan. 38: 35-42. https://doi.org/10.1023/B:VLSI.0000028532.53893.82
7. Sushant Sadangi, Satyakam Baraha, Darshan Kumar Satpathy, Pradyut Kumar Biswal, “FPGA implementation of spatial filtering techniques for 2D images”, 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017 May; 19-20. DOI: 10.1109/RTEICT.2017.8256791
8. Rahul Shandilya, R K Sharma “FPGA implementation of image enhancement technique for Automatic Vehicles Number Plate detection”, 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017 May : 11-12 India. DOI: 10.1109/ICOEI.2017.8300860
9. Mohamed Awad ; Ahmed Elliethy ; Hussein A. Aly “A Real-Time FPGA Implementation of Visible/Near Infrared Fusion Based Image Enhancement”, 2018 25th IEEE International Conference on Image Processing (ICIP), 2018 Oct: 7-10 DOI: 10.1109/ICIP.2018.8451602
10. Shengnan Xu , Lihui Yang , “Image enhancement algorithm in embedded network video monitoring system”, 2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization, 2012 Oct: 20-21 DOI: 10.1109/ICSSEM.2012.6340873

11. Han Xiao, Yu-Pu Song, Qing-Lei Zhou, "Multi-GPU Accelerated Parallel Algorithm of Wallis Transformation for Image Enhancement", International Journal of Grid and Distributed Computing 2014 April;7(2):99-114
12. Acharya and Ray, Image Processing: Principles and Applications, Wiley-Interscience 2005

AUTHORS PROFILE

Kibum Suh received the BS, MS, PhD degrees in electronics engineering from Hanyang University in Seoul, Korea in 1989, 1991, and 2000. He joined Electronics and Telecommunications Research Institute (ETRI) in Daejeon, Korea. He was engaged in the development of MPEG-4 and H.264 ASIC design, image compression algorithms and VLSI architecture for video codecs. He is currently in the Department of Electronics at Woosong University in Daejeon, Korea. He is currently engaged in research on image processing hardware design, image recognition algorithms, and SOC architecture design.