

Design and Simulation of Image Preprocessor Based on FPGA

Yuan Qinghui^{1, a}, Nie Xiujun^{1, b} and Feng Guofang^{2, c}

¹ Binzhou Polytechnic, Binzhou, Shandong Province, 256603

² Sixth Primary Schools in Bincheng, Binzhou, Shandong Province, 256603

^asdyqh1979@126.com, ^b443183538@qq.com, ^c48908892@qq.com

Keywords: FPGA, Image processing, Algorithm

Abstract: With the continuous development of image processing technology, and with the development and application of ASIC, DSP and FPGA devices, image processing has acquired a number of new technology, and FPGA based image processing technology has become the development trend in the field of image processing. This paper gives a brief introduction of the digital image, and gives the FPGA design scheme of the algorithm of median filter, edge sharpening and gray level histogram, and also gives the design principle diagram and simulation results. The experiment results showed that in the image processing system, it is feasible to realize the fast algorithm through the hardware of the software, which can well achieve the established objectives and meet the requirements of real-time.

Introduction

Image processing is generally divided into two types, one for digital image processing, the other is an analog image processing.

The object of the analog image processing is analog image, the method generally used with optical and photographic. Analog image processing advantage is started early, after a long development time, which theory has been very mature, and the processing speed is very fast; The disadvantage is that the design of the professional optical device is relatively large, and the processing device is huge, the data is not easy to store, Because of these shortcomings, its further development is limited^[1].

Digital image processing firstly use camera or camera and other digital equipment for image acquisition and then use the special image processing equipment for image processing to obtain the required information in the image, compared with analog image processing, digital image processing is more flexible, and storage is also convenient.

Usually based on the image processing algorithm of data, complexity and other characteristics, we can divide it into low, intermediate, and advanced image processing.

In this paper low-level image processing algorithms is implemented in FPGA, such as mean filter, median filter algorithm.

(1) Design of median filtering algorithm for 3 * 3 square windows

The filtering method first sorts the gray values of the three elements, and gets the maximum, median and minimum values of each row, and then sorts the three maximum, three median and three minimum values. as long as we have drawn the three maximum values of the minimum, median and three minimum values of the maximum value, by the previous theoretical analysis, the median value of the nine pixel gray value must be one of the three values of. We just have to carry on a three value comparison and get these three values, Finally we compare these three values and get these three values, that is, the median value for

us.

Three value comparison unit interface module 1 respectively.

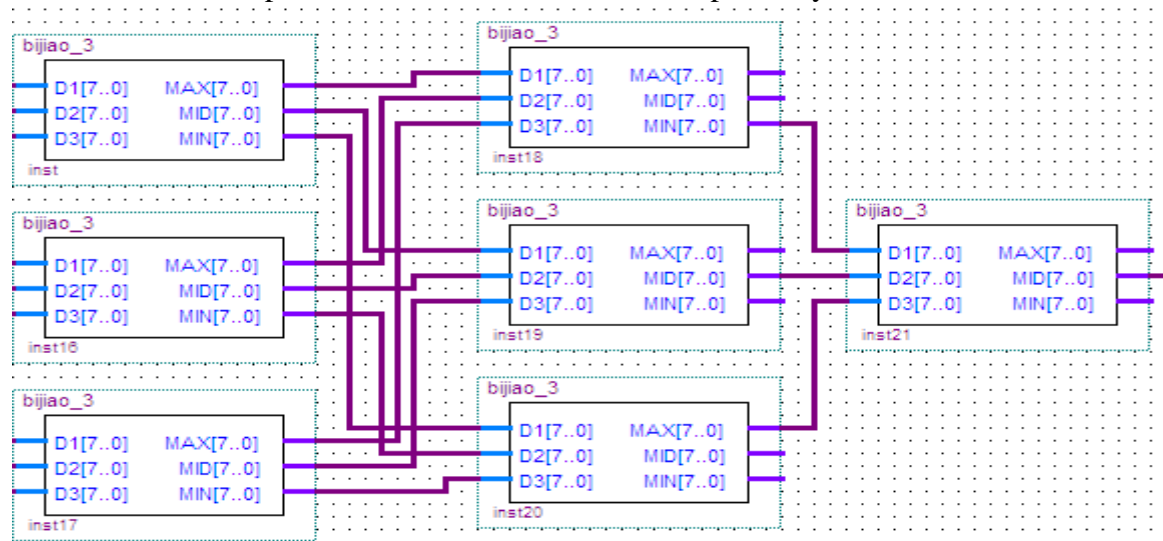


Fig. 1 The diagram of 3×3 square window median filter algorithm

From figure 6, 3 * 3 square window median filtering algorithm need to pass 7 times comparison of three values , and if it is designed synchronous computation, each of the three values requires the CLK clock cycle. Due to the parallel execution in the external, the horizontal direction is equivalent to pass 3 times comparison of three values, so the number of clock cycles required is 3 x 3=9. But this will cause a delay of 9 CLK clock cycles, namely, within the nine clock cycle, the output state of the MID port is uncertain. So in order to avoid this delay, by using asynchronous clock, we can achieve the results of the input data and get median value.

(2) Simulation of 3 * 3 square window median filtering algorithm

In the simulation, we use the QuartusII function simulation module to observe whether the design of the algorithm can achieve the function required. The simulation results are shown in figure 2.

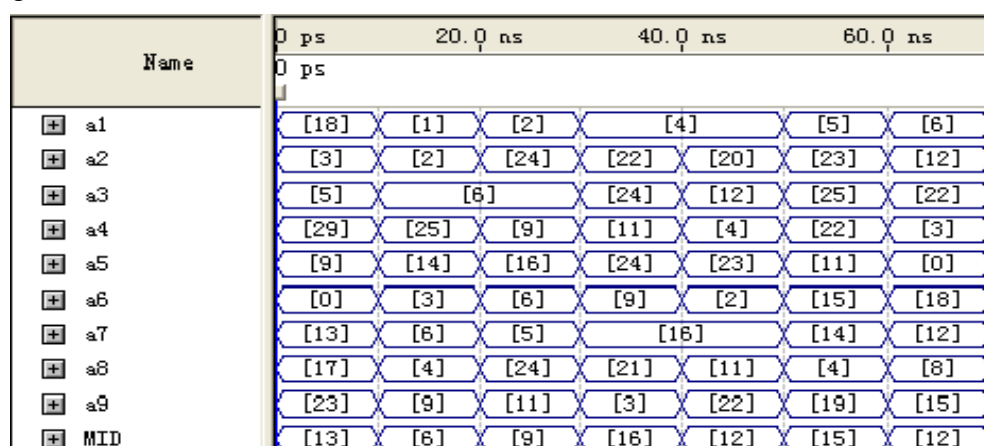


Fig. 2 The simulation result of 3×3 square window median filter algorithm

In order to achieve the purpose of simulation, pixel elements are artificially customized data. By the above results we can know that the output result is consistent with our original design, and it also avoids the data delay.

Edge sharpening

This design uses 3 * 3 size of the Prewitt operator, Prewitt operator seek the difference after it has seek average for image gray, so it can suppress noise^[2].

If we want to carry out the 3 x 3 window operator edge detection, it is necessary to firstly generate a 3 x 3 window, but in the generation window, not every pixel is effective.

Pixel capable of forming an effective window is elements in the solid line box, that is, to remove the elements of the first and last row and the first column and last column elements, so we can know, the size of the image resolution we set is 320 x 240, and the image resolution obtained by the algorithm is 318 * 238. In order to form a 3 * 3 window and Parallel output element, there is designed according to the following structure. As shown in figure 3.

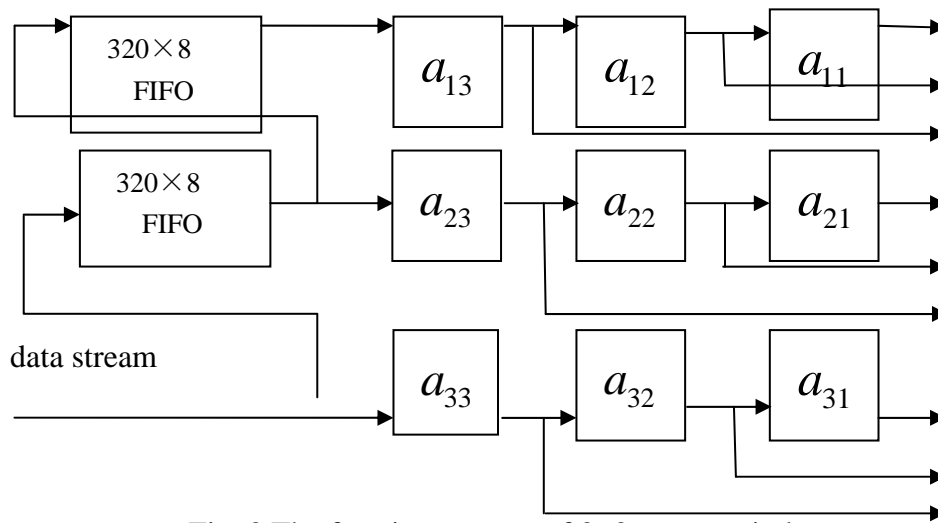


Fig. 3 The forming process of 3x3 square window

After the camera is initialized, it will enter the FPGA driven by pixel clock PCLK and along with the line start signal VREF and field start signal VSYNC. This need to design a module to count the VREF, and when VREF=3, 3 * 3 window is formed, and when the VREF=240, window is unable to form after pulse 323 PCLK, namely a field of data has been processed. Waiting for the VSYNC signal, the signal as the reset signal of the module and ready to a new field of data processing. The VSYNC signal waiting for is used as the reset signal of the module, and is ready to process the new data.

Realization of histogram equalization

Resolution for gray level 0 to 255 is 240 * 320 image, the cumulative distribution curve of the gray value can be represented by the following formula:

$$S_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{240 \times 320} = \sum_{j=0}^k P_r(r_j) \quad 0 \leq r_j \leq 1, k = 0, 1, 2, \dots, 255$$

Among them, the K is a pixel gray value, for the gray value, the number of pixels for the K. n_j is the number of pixels whose gray value is k.

According to the distribution function based on calculation, the corresponding relationship of gray value between the input image and the output image is established, and then the transformed gray level is restored to the original range.

To achieve histogram equalization, that need to complete three processes of the histogram, namely, statistics, cumulative and mapping. So, we need to design each module.^[3]

(1) Histogram statistics module

The histogram statistics module is used to count the number of various gray level pixels in the image, due to the input pixel data stream of FPGA, in order to keep the data in real time, the module with two double port RAM ping-pong operation, which a statistical operation, another is behind the operation.

As the pixel data is continuously input into the FPGA, in order to keep the data in real time, the module uses two dual port RAM to do ping-pong operation, that is, one of them does statistical operation, the other is the following operations. In order to achieve the purpose of counting each gray value, The address of the dual port RAM is represented by the gray level of 0 to 255, and each unit of the ram must be able to accommodate a image data, that is, each storage unit must be greater than or equal to $320 \times 240 = 76800$, so the corresponding data width is:

$17(2^{17} = 131072 > 76800, 2^{16} = 65536 < 76800)$, That is, the size of the RAM block that needs to be constructed is $256 * 17\text{bit}$.

Fig. 13 The double port RAM and space distribution of the histogram

The process description of the accumulation module is as follows: When the gray data after the arrival of the data as the dual port RAM address, the address data in the data will be removed, and then add a, then the data into the original address. When the gray data comes, this data is the address of the dual port RAM. Firstly we take the data out of the address, then the data plus one, finally put the data into the original address.

From that, Each pixel gray data have these three processes, that is, taking a number, adding 1 and writing the number, and when these three processes are carried out, address remains unchanged. So we design a special module whose the driving clock is 4 times of the pixel clock, Thus, a PLL (phase-locked loop) added in front of the module is used to spread spectrum. The module with a state machine to complete is divided into four states, respectively, waiting, reading, plus 1, writing.

When a pixel comes, firstly we Take the data out of the Pixel value corresponding address and add a, then it is sent to the address, until the statistics of a picture has completed.

The simulation diagram of statistical module is shown in Figure4.

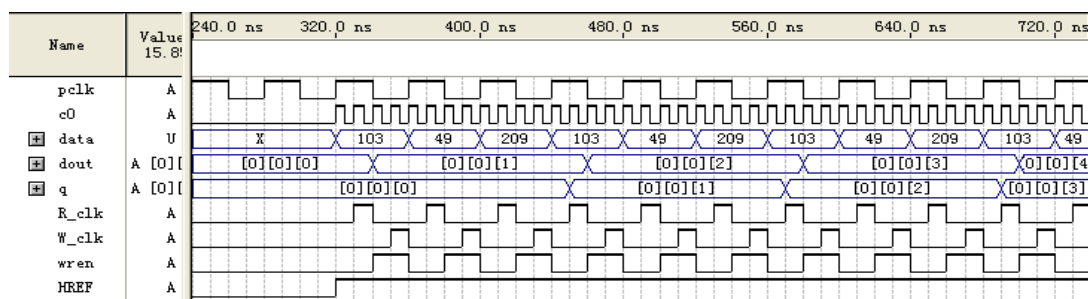


Fig. 4The simulation result of the statistics module

From the figure we can be seen, Only when the HREF signal is high, the clock signal is valid, i.e. pixel data was sent to the statistics module. In the beginning, 4 frequency clock of Pixel clock is not generated because the PLL module will calculate PCLK of clock cycles. When VSYNC signal comes, after 17 row periods, the first valid line signal can come, this

fully meets the time used by the PLL module to output the 4 PCLK clock. Within each PCLK period, the cumulative module have done four actions, that is, waiting, reading, accumulating, writing. The four state of the state machine can be read from the graph. The input data is artificially set, and it can be seen by taking out the numbers that the statistical results are correct, which proves that the module meets the requirements of our design.

(2) Histogram mapping module

Statistical module has been drawn to the number of pixels contained in each gray value, in order to get the mapping pixel value of a pixel value, it is needed to get the mapping table. The module realization process is as follows: When the gray histogram statistics are completed, it will send out the information to inform the system which achieve mapping table. First the 0 to 255 were sent to the statistical table address end, then the data is read out, and the data is added to the last gray value data, when all the gray scale calculation is completed, the mapping table can be drawn. Finally, the corresponding data are extracted from the mapping table according to the image pixel gray level data. the data is multiplied by the number of gray level (256) and then divided by the total number of pixel data, in this way, the data obtained is that the gray value corresponds to the gray value.

Mapping statistical module and mapping module schematic diagram 5

Schematic diagram contains two dual port RAM, respectively for the statistical table and mapping table, the two RAM are controlled by a finite state machine which is generated by an onboard phase-locked loop, the external clock is a pixel clock PCLK, and after spreading it get C0 and c1. The two RAM are working in turn, namely, first statistics, then mapping, which is completed by CSYNC control of the frame signal

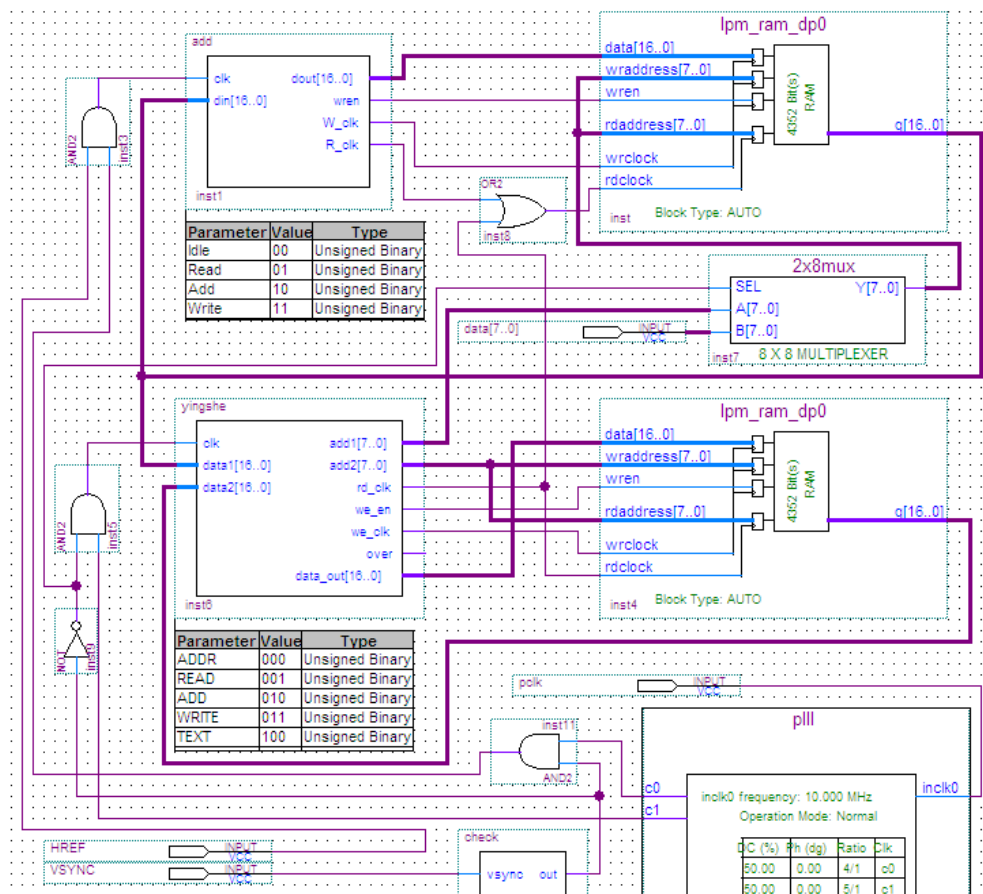


Fig. 5 The designer of the grayscale histogram equalization.

Conclusion

In order to achieve the high processing speed and real-time, in this paper image acquisition and processing of key pre processing algorithm are analyzed, and the underlying hardware processing are adopted to achieve, and a variety of image processing tools are designed by using the logic circuit, including gray-scale image conversion, filtering, edge detection and histogram processing, which provides a very important foundation for advanced visual programming in the future. Through theoretical demonstration and experimental data, it can be concluded that: in the image processing system, it is feasible to realize the fast algorithm through the hardware of the software, which can well achieve the established objectives and meet the requirements of real-time.

Acknowledgments

The research was supported by the science and technology development program of Binzhou Polytechnic. The title of this paper is the development of visual system of Mounter based on FPGA(Project Number: YJKT15)

References

- [1] Niklas Lepisto,Benny Thornberg,Mattias O’Nils.High-Performance FPGA based Camera Architecture for Range Imaging, NORCHIP Conference, Vol.4 (2005), p. 165-168.
- [2] Hu Yueli, Ji Huijie, Wu Pin. Research on Image Filtering Algorithm and Its FPGA implementation. Computer Measurement & Control, Vol.11(2008), p. 1672-1675.
- [3] Woods, R.E and Gonzalez, R.C. Real-Time Digital Image Enhancement. Proc. IEEE. Vol.5 (1981), p. 643-654.