ISSN: 2664-9640

Moving Target Detection Method Based on FPGA

Wensheng Li [a], Qingsong Lin [b], Yiming Liu [c] and Shuai Zhang [d]

Henan University of Science and Technology, School of Information Engineering, Luoyang 471000, China;

[a]18437952315@163.com, [b]lqs6501@163.com, [c]383474461@qq.com, [d]zs1289879@163.com

Abstract

In recent years, with the continuous development and progress of computer technology and image processing technology, highly practical moving target detection technology has become a hot topic in major universities and research institutions. At present, the widely used X86 processor-based moving target detection system can no longer meet today's society's pursuit of low-carbon and environmental protection. Therefore, this paper proposes a low-power FPGA-based moving target detection system. Starting from the FPGA implementation of the moving target detection algorithm and the FPGA implementation of the image acquisition and display function, the paper designs and implements the FPGA-based image acquisition, image grayscale, image filtering and inter-frame difference method and other functions. Each function has been tested and analyzed.

Keywords

FPGA; Moving target detection; Morphological image processing; Inter-frame difference method.

1. Introduction

Vision is one of the important ways for people to perceive the external world. About 80% of external information is obtained through the eyes [1]. The human visual system has very rich functions, including image acquisition, information transmission, and processing of image information in the brain. It also processes external information with high efficiency, high speed, and large processing volume. The realization of such a complete vision system has always been the highest pursuit in the field of vision technology.

Computer vision [2] is such a research field that uses how to use computers combined with image sensors to build a set of simulated biological vision systems. Moving target detection [3] technology is a middle-to-high level algorithm in computer vision systems, which has high research value and challenge. Moving target detection, in general, is to find the moving target contained in the image collected by the camera. Realizing the detection and tracking of moving targets is mainly to acquire and recognize the characteristic data and movement trajectories of moving targets. Target detection and related technologies have been widely used in many fields such as agriculture, industry, medical care, transportation, military, and public security.

At present, most moving target detection is developed based on the X86 processor. Although these systems have good performance, they are often at a disadvantage compared to embedded systems in terms of power consumption. Therefore, the study of a low-power embedded moving target detection system has important meaning.

2. FPGA-based image acquisition and display system

The FPGA chip used in this article is EP4CE10 of the Intel Cyclone IV series, and the Verilog HDL code of each function module is written using Quartus II 13.1 software. The system flow chart is shown in Figure. 1. It can be seen from Figure 1 that the system first uses the image sensor OV7725 to collect image data, and then the image is grayed and filtered, and then the inter-frame difference method is used to detect the moving target in the continuous image. And finally mark the moving target in real time on the VGA display [4].

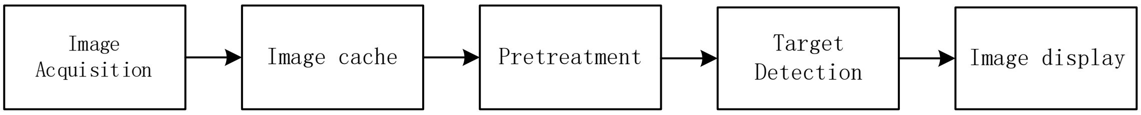


Figure. 1 System workflow

2.1. Camera driver module

2.1.1. Camera I2C control bus introduction

The board selected in this article has reserved the interface of the CMOS camera module, and the interface follows the interface definition of all official CMOS modules. The camera model uses OV7725 Sensor as the image sensor. The official OV7725 document points out that OV7725 should use the SCCB interface to modify the registers. The SCCB protocol is a simplified form of the I2C protocol we often use [5]. The I2C bus mainly consists of a data line SDA and a clock SCL to form a serial bus; each circuit and module has a unique address. The registers of OV7725 can be configured to read and write through the I2C bus. The complete data consists of 9 bits, the upper 8 bits are the data that needs to be transmitted, and the minimum is different according to the environment where the device is located. Figure 2 shows the timing diagram of the I2C bus.

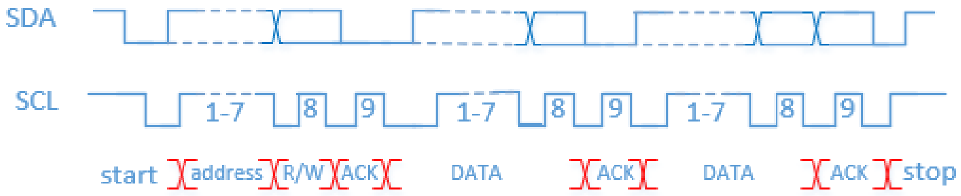


Figure. 2 I2C bus timing diagram

As can be seen from Figure. 2, the I2C bus itself has three types of signals:

(1) Start signal: During the period when the clock line SCL remains high, the level on the data line SDA is pulled low. This is defined as the start signal of the I2C bus, which marks the beginning of a data transmission.

(2) Stop signal: During the period when the clock line SCL maintains a high level, the data line SDA is released, causing SDA to return to a high level, which is called the stop signal of the I2C bus, which marks the termination of a data transmission.

(3) Response signal: All data on the I2C bus is transmitted in 8-bit bytes. Each time a byte is sent, the data line is released during the clock pulse 9, and the receiver feeds back a response signal.

2.1.2. Realization of camera data collection function

The video output port of the development board used in the design is in the RGB565 [6] format, so the OV7725 needs to be configured in the RGB565 format. It can be seen from the output timing diagram of the camera in Figure. 3 that OV7725 outputs one line of image data when the HREF signal is high, and the output data is valid at the rising edge of PCLK, so that each CLK output of OV7725 is 8bit, but RGB565 shows every One pixel is 16bit, so in order to meet the pixel format of RGB565, the pixels of each image are output in stages, the first Byte output is

R4-R0 and G5-G3, and the second Byte output is G2~G0 and B4-B0. Splicing the front and back 2 bytes is 16Bit RGB565 data.

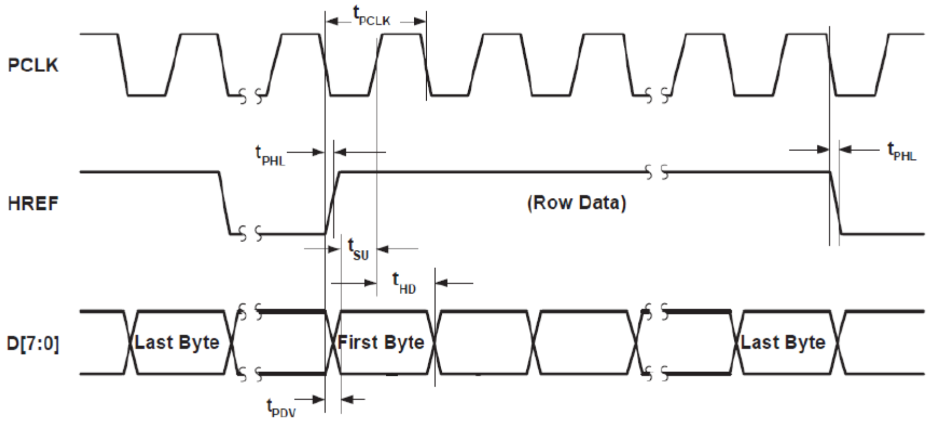


Figure. 3 Camera output timing diagram

According to the output characteristics of the OV7725 and the control principle of the I2C bus, the RTL diagram finally obtained through the Verilog HDL code is shown in Figure. 4. Picture (a) is the camera driver module of I2C configuration, and picture (b) is the image capture module.

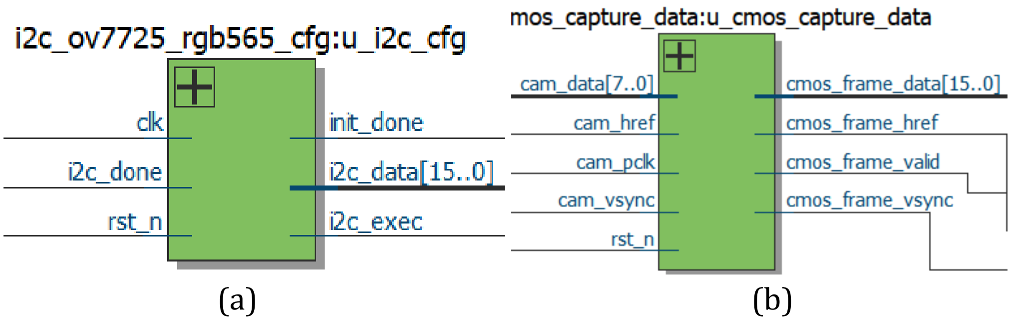


Figure. 4 RTL diagram of camera driver module

2.2. SDRAM read and write control module

The image data collected by the camera needs to be cached in SDRAM to prepare for the call of other modules. Therefore, it is very important to realize the image data cache and read work, which plays a role in connecting the whole system [7].

The FPGA implementation of the SDRAM read/write control function has different configuration methods for different memories. The memory used in this system is a more advanced DDR3 memory, power consumption is less than 1W, but DDR3 controller programming is difficult, so this design uses Intel to provide users with DDR memory control IP core, you can do not understand the DDR control It is also very convenient to read and write the DDR processor through the DDR controller in the case of the mode and the read and write timing of the DDR.

DDR3 controller consists of 3 parts: user interface block (User interface Block), memory control module (Memory Controller) and DDR3 physical interface (Physical Layer). Developers only need to develop the user's logic design and interface with the user interface of the DDR controller to read and write DDR3 data. Figure. 5 shows the RTL diagram of the SDRAM read-write control module.

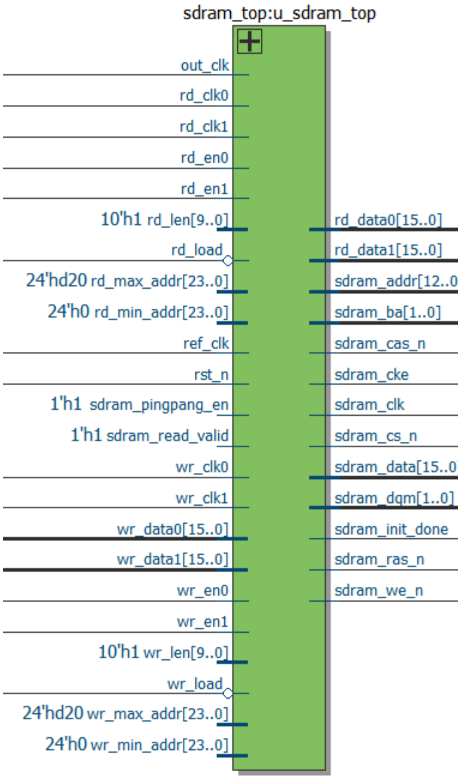


Figure. 5 SDRAM read and write control RTL diagram

2.3. VGA driver module

This article uses a VGA monitor for image display. The VGA monitor has a specific timing chart and must follow the VGA corresponding timing to obtain the target image. The important principle of VGA image display is scanning, and the basic element of scanning is a line, and then a frame is composed of multiple lines. Figure. 6 shows the time sequence of displaying one line. Among them, "Active"Video is the effective pixel of one line of video, most of which are resolved. In the rate clock, Top/Left Border and Bottom/Right Border are both 0. "Blanking" is the synchronization time of one line, and "Blanking" time plus Active"Video time is the time of one line. "Blanking" is divided into "Front Porch", "Sync", and "Back Porch" three segments.

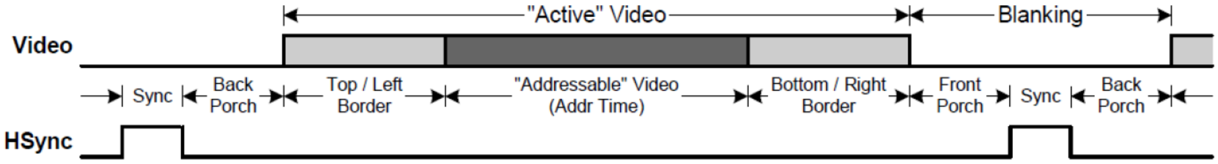


Figure. 6 VGA line synchronization timing

The image display resolution of this design is 640×480, the field frequency is 60Hz, and the line frequency is 31.5kHz. Under this parameter, the values of each parameter for scanning one line according to industry standards are: Tc-96, Td-48, Ta-640, TF -16, so the line period is 96+48+640+16-800, which is the prime factor. The values of trace 1 are: Tc-2, Td-33, Ta-480, TF-10, so the field period is 2+33+480-10-525, and the unit is pixel.

According to the 640X480 resolution protocol and the timing requirements of the VGA display, a control module of the VGA interface is designed. Figure 7 is the RTL diagram of the module, where pixel_data [15 0] is the input of the video signal, and vga_hs, vga_rgb [15 0], vga_vs are the VGA row height, VGA video data signal, and VGA column height respectively.

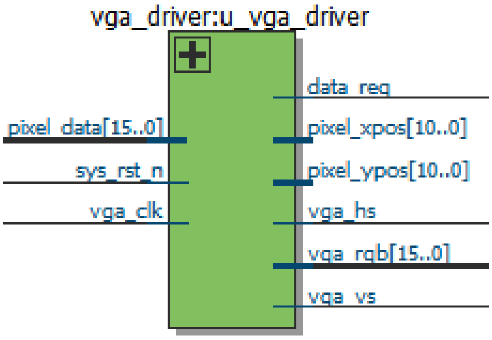


Figure. 7 VGA interface driver RTL diagram

3. FPGA Implementation of Moving Target Detection Technology

The realization of FPGA-based moving target detection technology is a process of implementing multiple image processing algorithms on FPGA. In order to realize the moving target detection algorithm, the image processing algorithms that this article will design and implement include image graying, inter-frame difference method and morphological filtering.

3.1. Color space conversion

The format of the image collected by the OV7725 camera and input to the memory is in RGB format. In order to reduce the calculation amount of the inter-frame difference algorithm, it is necessary to convert the RGB image into a grayscale image. The Y signal in the YCbCr model is a brightness signal, also called a gray signal, and the RGB model [7] and the YCbCr model can be converted mutually by the law of formula (1). Therefore, this article adopts the method of converting RGB to YCbCr, and then extracting the Y channel signal in YCbCr format to realize the gray scale of the image. Since Verilog HDL cannot calculate decimals, the method of multiplying the whole by 256 and shifting the value to the right by 8Bit is adopted here. The conversion formula is shown in (2).

$$\begin{cases} Y = 0.183R + 0.614G + 0.062B \\ Cb = -0.172R - 0.339G + 0.511B + 128 \\ Cr = 0.511R - 0.428G - 0.083B + 128 \end{cases} \tag{1}$$

$$\begin{cases} Y = \left((77R + 150G + 29B) >> 8 \right) \\ Cb = \left((-43R - 85R + 128B) >> 8 \right) + 128 \\ Cr = \left((128R - 107G - 21B) >> 8 \right) + 128 \end{cases} \tag{2}$$

Although the calculation formula is adjusted and the transformation formula is adjusted to formula (2), the process of converting formula (1) to formula (2) is still more complicated to implement through Verilog HDL code. Therefore, it was decided to introduce the FPGA pipeline design method and add a three-stage pipeline. For example, when designing the Y transformation, first calculate all the multiplication operations in the brackets, this is the first clock; then calculate the addition in the brackets, this is the second clock; finally, all the values including 128 are expanded by 256 Double, this is the third clock, so that the process of shifting the Y signal to the right by 8 bits is realized. Since there is a decimal calculation value in the calculation result, the upper eight bits of the calculation value are taken in the FPGA. Figure. 8 is the RTL diagram of the RGB to YCbCr module.

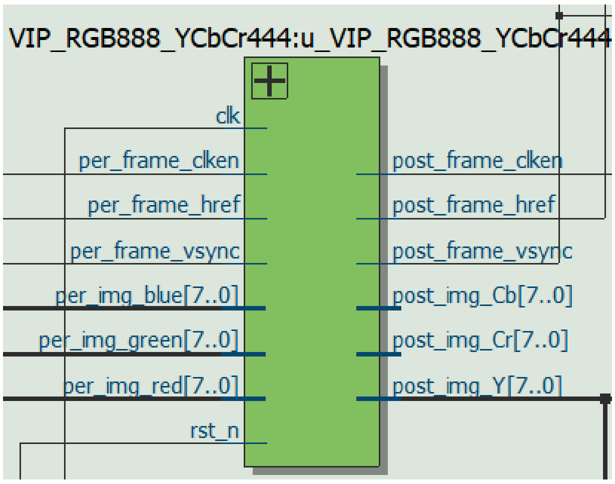


Figure. 8 RTL diagram of RGB to YCbCr module

It can be seen from Figure 2.1 that the main data input signals of the module are the signals of the RGB three components of the image and the display control signal and frame synchronization signal. After processing by this module, the frame synchronization signal and display control signal remain unchanged, but output. The new Y component data post_img_y [7:0] signal, this 8-bit data post_img_y [7:0] is input to other image processing modules as grayscale image data.

3.2. Morphological filtering

Noise is inevitably introduced in image acquisition and generation. Image noise refers to unnecessary or redundant interference information in image data, which interferes with our extraction of image information, so filtering processing is essential of. The noise in the gray-scale image presents the shape of white spots. The effect of these noises through median filtering and average filtering is general. This paper adopts the method of morphological filtering to achieve the suppression of the noise in the gray-scale image.

Digital morphology [8] is an image processing tool used to extract useful image components that express and describe the shape of a region from an image. The main idea is to use structural elements to operate the image plan and finally complete the image analysis. In digital morphology image processing, the two algorithms of corrosion and expansion are the most basic, but they are widely used to eliminate noise, segment independent image elements, and connect adjacent elements [6].

Erode

Erode is a process of eliminating boundary points and shrinking the boundary to the inside. The definition of corroding A with structural element B is shown in formula (3). The meaning of this formula is: the set of these points that still intersect with A after B is translated by z steps is the result of B corroding A.

$$A \ominus B = \left\{ z \mid (B)_z \cap A \subseteq \varnothing \right\} \tag{3}$$

Dilate

The function of dilate is to merge all the background points in contact with the object into the object, and the process of expanding the boundary to the outside. In digital morphology, the current pixel is replaced with the maximum value of the defined pixel set. Add pixels to the objects in the image, so that the target object area has more details. Formula (4) is the definition formula of dilation of structure element B to A. This formula is based on the image of B about its origin, and is based on the translation of the image by z. The expansion of B to A is the set of all displacements z, so that at least one element of B and A overlaps.

$$A \oplus B = \left\{ z \mid (B)_z \cap A \neq \varnothing \right\} \tag{4}$$

In digital morphology, the first erosion and then dilate is called the opening operation. The opening operation can smooth the contours of the object, eliminate the small protrusions; while the first dilate and then the erosion is called the closing operation, which is mainly used to eliminate voids. Fill the role of contour breaks. This article selects the open operation for image processing and eliminates the isolated noise in the image.

When using FPGA to implement the corrosion expansion algorithm, the structure element size selected by the Erode and expansion module is a 3*3 matrix, and each pixel of the entire picture is scanned with the structure element, and the structure element and the binary value scanned by the structure element are used. The image performs the "AND" operation, when the two numbers are both 1, the pixel of the output image is 1, otherwise it is 0, which will reduce the binary image; use this structural unit to scan each pixel of the entire image, use the structure element is OR with the binary image scanned by the structure element. If both numbers are 0, the element of the output image is 0, otherwise it is 1, which will enlarge the image. Figure. 9 is a principle of operation of the module of FIG corrosion, and the digital operation is similar.

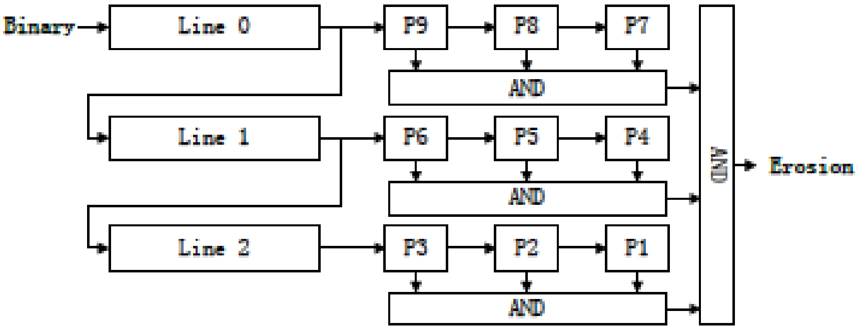


Figure. 9 Schematic diagram of corrosion expansion module

When designing the erode and dilate algorithm, the FPGA pipeline design method is adopted, which is divided into two clocks. In the first clock, first calculate the calculation results of the expansion/corrosion structure unit scanning each row, that is, calculate the three values of P1, P2, and P3; in the second clock, integrate the calculated three rows of values, and finally get Expansion structure element covering the erode or dilate result of this pixel area. According to this design process, the RTL diagram of the erode and dilate module can be obtained.

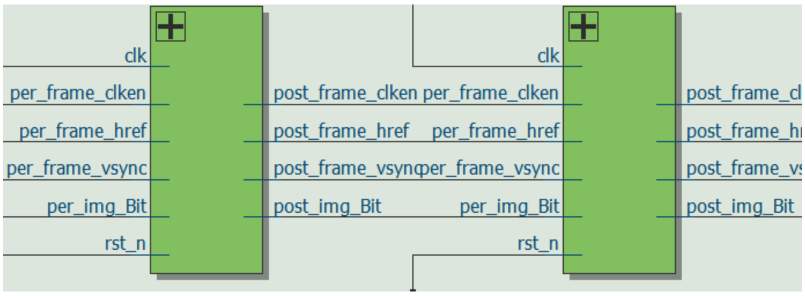


Figure. 10 RTL diagram of corrosion expansion module

3.3. FPGA Realization of Inter-frame Difference Method

The image and video captured by the image sensor are continuous. The inter-frame difference method is an algorithm designed using the continuity of the images collected by the image sensor.

The principle diagram of this algorithm is shown in Figure . If there are moving targets in the scene, there will be obvious grayscale changes in the image pixels between two or three consecutive frames. Therefore, when designing the target detection algorithm, you can perform the difference operation on the image gray between the current frame and the previous frame[9]. By judging whether the absolute value of the gray difference exceeds the set threshold, you can determine whether there is motion in the image. Target, so as to realize the function of moving target detection.

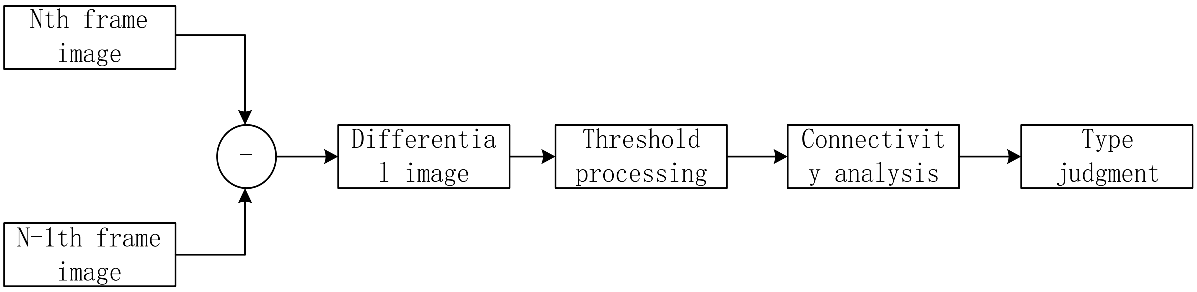


Figure. 11 Inter-frame difference method operation process

The operation process of the two-frame difference method is shown in Figure. 11. In the video sequence, the nth frame and the n-1th frame of the image are the sum, and the gray values of the pixels corresponding to the two frames are $f_n(x,y)$ and $f_{n-1}(x,y)$. Equation (5) takes the gray values of the corresponding pixels of the two frames Subtract and take its absolute value to obtain the difference image D_n.

$$D_n =| f_n(x,y) - f_{n-1}(x,y) | \tag{5}$$

Set the threshold, and binarize the pixels one by one to obtain the binarized image R_n'. Among them, the point with the gray value of 1 is the foreground (moving target) point, and the point with the gray value of 0 is the background point; the connectivity analysis of the image Rn can finally obtain the image R_n containing the complete moving target.

$$R_n'(x,y) = \begin{cases} 1 & D_n(x,y) > T \\ 0 & sele \end{cases} \tag{6}$$

The inter-frame difference method has a simple algorithm principle and a small amount of calculation, which is suitable for hardware implementation [10].

When implementing the inter-frame difference method in FPGA, for two consecutive frames of images, the previous frame of image will be stored in SDRAM as a backup; the latter frame of image will also be cached in SDRAM, and finally the pixels of the two frames of image By subtracting points, you can get the moving target.

When implementing the inter-frame difference method in FPGA, for two consecutive frames of images, the previous frame of image will be stored in SDRAM as a backup; the latter frame of image will also be cached in SDRAM, and finally the pixels of the two frames of image By subtracting points, you can get the moving target.

According to this design idea, the inter-frame difference algorithm is perfectly realized in FPGA. Figure. 12 is the RTL diagram of the inter-frame difference algorithm.

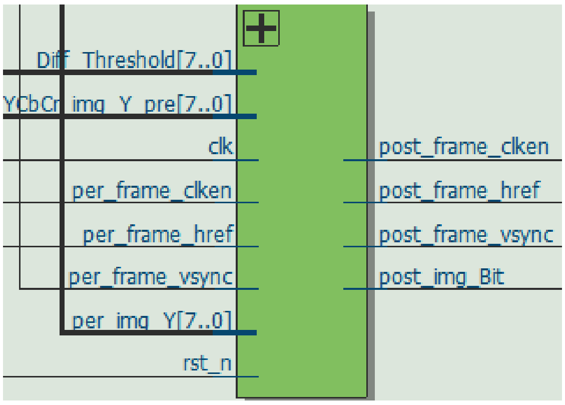
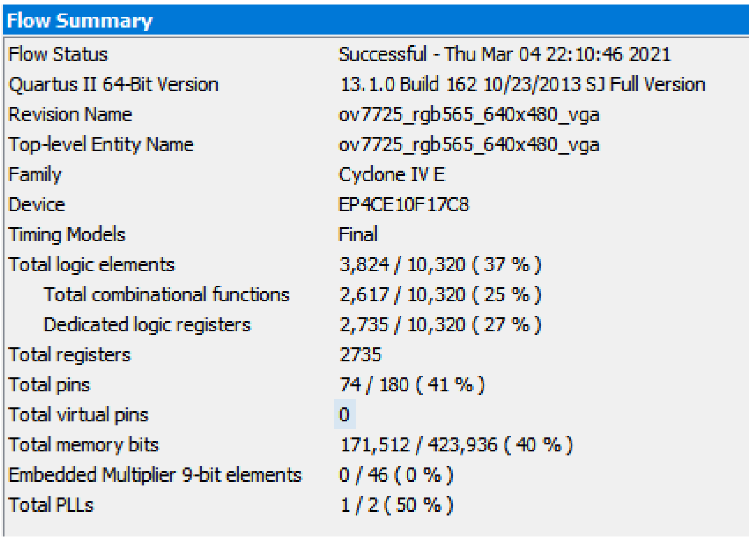


Figure. 12 Inter-frame difference method RTL diagram

In the figure, per_img_y[7..0] is the gray signal of the previous frame, Diff_Threshold[7..0] is the threshold signal, YCbCr_img_Y_pre[7..0] is the gray signal of the current frame, the threshold setting in this article It is 8'h14.

4. System performance test and analysis

After the design of all modules is completed, the overall design module is synthesized through the Quartusll simulation software. As shown in Figure 12, the system resource consumption is as follows: 3,824 logic resources are used, the utilization rate is 37%, and 2735 registers are used. A phase-locked loop is installed. This means that abundant resources can also be used to improve and expand the system.



Flow Summary

Flow Status	Successful - Thu Mar 04 22:10:46 2021
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Version
Revision Name	ov7725_rgb565_640x480_vga
Top-level Entity Name	ov7725_rgb565_640x480_vga
Family	Cyclone IV E
Device	EP4CE10F17C8
Timing Models	Final
Total logic elements	3,824 / 10,320 (37 %)
Total combinational functions	2,617 / 10,320 (25 %)
Dedicated logic registers	2,735 / 10,320 (27 %)
Total registers	2735
Total pins	74 / 180 (41 %)
Total virtual pins	0
Total memory bits	171,512 / 423,936 (40 %)
Embedded Multiplier 9-bit elements	0 / 46 (0 %)
Total PLLs	1 / 2 (50 %)

Figure. 13 System resource utilization

After the bit file with the correct function is downloaded to the development board, the system function can be tested and analyzed. Shown in Figure 13 are the test results of the various modules of the system. The experimental background is set as a smooth plane, and the camera is used to record the process of a black box moving to a fixed beige pen to detect whether the system can capture the moving target.

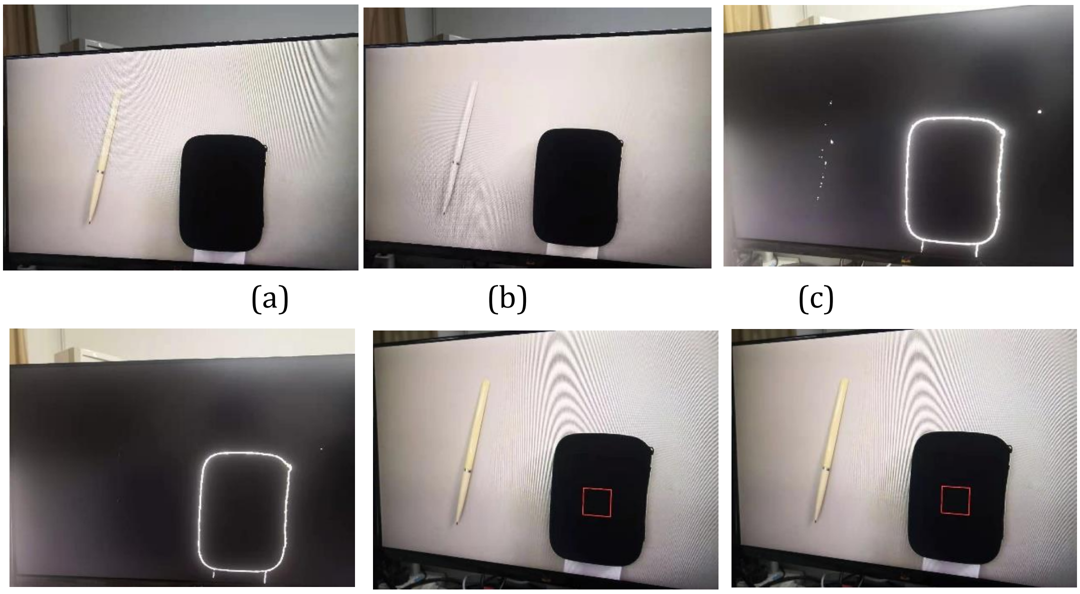


Figure. 14 Function test diagram of each functional module

In Figure 13, Figure (a) is an image with a moving target captured by the image sensor; Figure (b) is a grayscale processed image of Figure (a). After processing, the ratio of yellow becomes white obviously; Figure (c) The moving image obtained after the inter-frame difference algorithm; Picture (d) is the picture after the morphological processing of the picture (c). It can be clearly seen that the isolated points of the picture (c) are eliminated, reducing Interference; Figures (e) and (f) are the target detection effect diagrams, and the slow-moving black box can be well detected. Through many experimental tests and analyses, the performance of the system is summarized as follows:

Table 1 Performance summary

Name	FPGA	X86
Power consumption	1.344W	>10W
Processing time	0.5S	2.5S

In Quartus II, you can see that the FPGA's on-chip power consumption is only 1.344w, which meets the needs of low power consumption.

5. Summary

When designing a moving target detection system, first analyze the principles and functions of each module, and then complete the Verilog code of each module on the Quartus II platform, and configure the module functions to FAGA. By debugging the system functions, The problem was found and solved, and finally a moving target detection system with low power consumption was realized on FPGA.

References

[1]. Han Jiuqiang. Machine Vision Technology and Application [M]. Higher Education Press. 2009, p. 1-6.

[2]. Zhang Yujin. Computer Vision Course [M]. People's Posts and Telecommunications Press, 2011, p.15-16

[3]. Qu Zhong, An Shiquan. Moving target detection and tracking in video sequence[M]. Science Press, 2017, p.17-19

[4]. Le Niea , Limin Maob, Zhenghang Zhao. Design and Implementation of Visual System Based on FPGA[J]. 2019 2nd International Conference on Mechanical Engineering, Industrial Materials and Industrial Electronics.Wuhan, 2019, p. 439-439

[5]. Lu Qingsong, Xu Yishen. Design of Target Recognition and Tracking System Based on FPGA [J]. Modern Electronic Technology, 2018, 41 (18), p.12 - 16.

[6]. Information on: http://www.cnblogs.com/oomusou/archive/2008/10/24/de2_70_rgb16.html

[7]. Translated by Gonzalez (RC.), Woods (RE.), Ruan Qiuqi, Ruan Yuzhi, etc. Digital Graph R Processing (Third Edition) [M], Beijing: Electronic Industry Press, 2011.6, p. 254- 257

[8]. Cai Limei, Wang Lijuan. Digital Image Processing[M]. Xuzhou: China University of Mining and Technology Press, 2014.8, p.175-187

[9]. Baha N. A real-time stereo vision based on obstacle detection[A]. Emerging Trends and Advanced Technologies for Computational Intelligence[C]. San Diego: Springer, 2016, p. 347-364

[10]. Yu Ge, Jia Qian, Li Yuanshuang et al. Moving target detection and tracking system based on DM642[J]. Modern Scientific Instruments. 2011, 4., p. 2:30-32