Forest Chemicals Revew
www.forestchemicalsreview.com
ISSN: 1520-0191
July-August 2021 Page No. 545-556
Article History: Received: 12 May 2021 Revised: 25 June 2021 Accepted: 22 July 2021 Publication: 31 August 2021

Research on Embedded Machine Vision Inspection System Based on FPGA

Xin Nie[1*], Chengcheng Lou[2], Ruolan Yin[2]
[1]Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430205, China.
[2]School of Computer Science & Engineering, Wuhan Institute of Technology, Wuhan 430205, China.
*Corresponding Author.

Abstract:

With the improvement of automation level, detection systems and real-time processing systems based on machine vision appear more and more frequently in people's lives, and are widely used in public security traffic, safety monitoring, mechanical tests, industrial detection and other fields. On the basis of analyzing the structure and working mode of FPGA (Field Programmable Gate Array), this paper points out the advantages of using FPGA for image acquisition and real-time processing to solve the real-time problem of the system. An embedded machine vision scheme based on FPGA with processor core is designed, and the hardware platform of embedded machine vision detection system is built. On this basis, the system has completed the basic functions including image acquisition, ARM-FPGA data interaction and data output. Effectively solve many shortcomings of traditional PC-based machine vision detection system; It has been verified that the system is efficient and reliable, and can replace the traditional PC-based machine vision inspection system in some industrial occasions, and continuously and stably measure and control products.

Keywords: *SoC, FPGA, Embedded system, Machine vision inspection.*

I. INTRODUCTION

The application of machine vision inspection system in the field of automatic production is an important way to realize the intelligence, high efficiency and high precision of industrial equipment[1]. However, the traditional PC-based machine vision inspection system is not modular in structure, poor in portability and inconvenient in installation, especially it is difficult to communicate with industrial field devices. Therefore, there is an urgent need for an integrated machine vision component that is more suitable for industrial needs, instead of the traditional PC-based machine vision inspection system.

Using ASIC (Application Specific Integrated Circuit) to realize vision processing algorithm can solve the contradiction between performance, size and power consumption of vision system, and it is an effective solution for high-performance embedded vision system. However, ASIC has some shortcomings, such as long development cycle and inconvenient modification.

As a high-performance programmable logic device, FPGA (Field Programmable Gate Array) can easily modify its internal logic functions by programming, thus realizing high-speed hardware calculation and parallel operation, which is a more convenient solution for high-performance embedded vision system[2]. With the continuous progress of technology, the integration of FPGA is getting higher and higher, the design scale that can be realized is getting larger and larger, and the power consumption is getting lower and lower[3-4]. Therefore, embedded vision system based on FPGA will be an important development direction of computer vision system.

In this paper, aiming at the problems of large size and poor real-time performance of the existing machine vision inspection system based on hardware and software co-design, an embedded machine vision inspection system solution based on ARM+FPGA is built, the image processing operation is realized by FPGA, and an FPGA connected domain analysis algorithm for target location is proposed. Compared with the traditional machine vision inspection system, the embedded vision system based on FPGA built in this paper has the advantages of small size, high resource utilization and small delay, and can be widely used in target location using machine vision.

II. SYSTEM ARCHITECTURE

Embedded machine vision measurement and control system integrates CCD camera, image processor and communication control interface, embeds real-time operating system kernel and image processing algorithm, and is an intelligent image acquisition and processing unit. Through analysis, parameter information is configured by configuration software, relevant functional modules are called, data are collected and processed, results are fed back, and after being connected with field industrial control equipment, measurement and control tasks can be conveniently completed. The overall structure of embedded machine vision measurement and control system is shown in Figure 1.

Article History: Received: 12 May 2021 Revised: 25 June 2021 Accepted: 22 July 2021 Publication: 31 August 2021

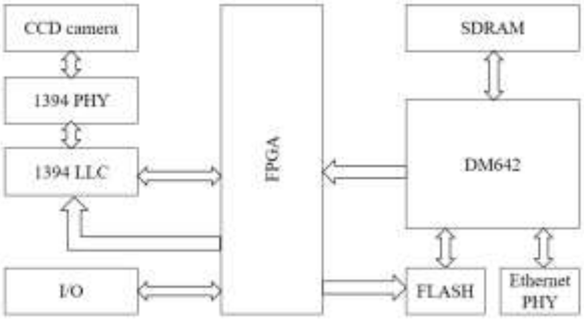

Fig 1: Architecture of embedded machine vision monitoring system

It consists of acquisition module, processing module, external communication module and other peripheral devices. Image signals are collected by CCD camera and sent to FPGA via 1394 bus. FPGA converts the image data and stores it in SDRAM. DSP calls the image processing program in FLASH to process the image data in SDRAM in real time, and sends control signals to field devices through digital IO according to the processing results. The whole processing flow can be remotely monitored in real time through Ethernet.

III. DESIGN OF SYSTEM HARDWARE PLATFORM

3.1 System Hardware Module Division

The whole hardware of the embedded machine vision inspection system proposed in this paper is divided into three independent modules: visual image acquisition, conversion and interface module, visual image acquisition, processing and control module, and motor drive module. The relationship block diagram is shown in Figure 2.

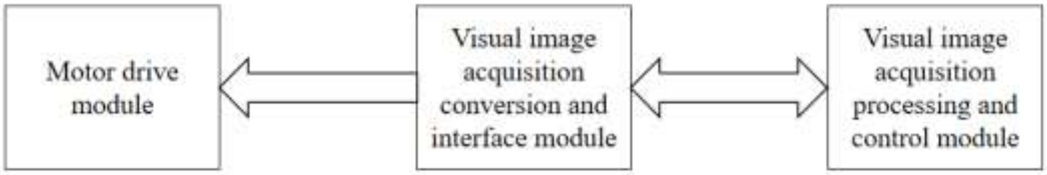
Motor drive module
Visual image acquisition conversion and interface module
Visual image acquisition processing and control module

Fig 2: Hardware module partition

3.2 Design of Control Interface Based on Avalon-MM Protocol

Avalon-MM protocol uses address and enable signal to read and write data, which is divided into master interface and slave interface[5-6]. The main interface is used to access the external storage controller by IP verification, that is, the IP core writes out data to the outside; the slave interface is used to monitor and control the IP core attributes, that is, to write data to the IP core from the outside.

In all IP cores in FPGA hardware engineering, the Slave ports used are clk, chipselect, address, read, readdata, write and writedata. In addition, Slave interface signals include waiting period signal, pipeline signal, burst signal, flow control signal, tri-state signal and other signals. If necessary, please refer to Avalon bus interface manual, which is not listed here. When designing Slave interface, data transmission can be completed by assigning logical values according to the use requirements of each signal. The basic interface signals provided in Avalon-Slave interface are shown in Table I.

TABLE I. Slave interface signal

Signal type	**Signal width**	**Direction**	**Function and use description**
clk	1	In	Avalon slave port synchronous clock, all signals must be synchronized with clk, and asynchronous peripherals can ignore clk signal.
chipselect	1	In	Avalon chip select signal from port. Avalon ignores all other signals from the port when the chipselect signal is invalid.
address	1~32	In	The address line connecting Avalon switching architecture and slave port specifies an address offset of one word from the peripheral address space.
read	1	In	Read the request signal from the port. When the slave port does not output data, it is not necessary to use this signal. If this signal is used, the readdata or Data signal must also be used.
readdata	2^n（2<n<11)	out	When reading transmission, it is output to the data line of Avalon switching architecture. This signal is not needed when no data is output from the port.
write	1	In	Write the request signal from the port. This signal is not needed when the slave port does not receive data from the master port. If this signal is used, the writedata or data signal must be used, and the writebyteenable signal cannot be used.
writedata	2^n（2<n<11)	In	Data line from Avalon switching architecture when writing transmission. This signal is not needed when no data is received from the port. If this signal is used, the write or writebyteenable signal must be used, but the data signal cannot be used.

In the image processing IP core proposed in this paper, the Slave port is mainly used to receive the signals sent by the applications in HPS, including control signals such as IP core configuration and IP core start and stop. Master port is used to write the processing results of IP core into HPS. In image processing IP core, connected domain feature information obtained by connected domain analysis algorithm is written into HPS, which is further extracted and used by subsequent applications.

3.3 Design of VGA Driver Circuit

Because FPGA outputs 3.3V level and VGA requires 0~0.714V analog signal, in order to design VGA standard interface and complete level conversion, it is necessary to design DAC digital-to-analog conversion function to realize 0~0.714V analog video signal transmission. According to the depth level of RGB, different digital-analog implementation schemes can be adopted, and there are many different design schemes for VGA driving circuit[7]. When designing the driving circuit of VGA, we mainly pay attention to the five signals shown in Table II.

TABLE II. Signal table

Serial number	Number	Name	Describe
1	1	RED	Red primary color signal input
2	2	GREEN	Green primary color signal input
3	3	BLUE	Blue primary color signal input
4	13	HSYNC	Row synchronization signal
5	14	VSYNC	Field synchronization signal

HSYNC and VSYNC are 3.3V digital signals, which can be directly connected with FPGA without considering level conversion in design, while RGB is 0~0.714V analog signal, which is the part to be processed. The VGA interface has been introduced for decades, so the DAC conversion scheme for digital-to-analog conversion has already matured[8-9]. Using special video conversion DAC chip, VGA interface can be designed quickly and the speed of product launch can be accelerated; In addition, the chip has a stable ASIC, which ensures the stability of video transmission quality and has high cost performance.

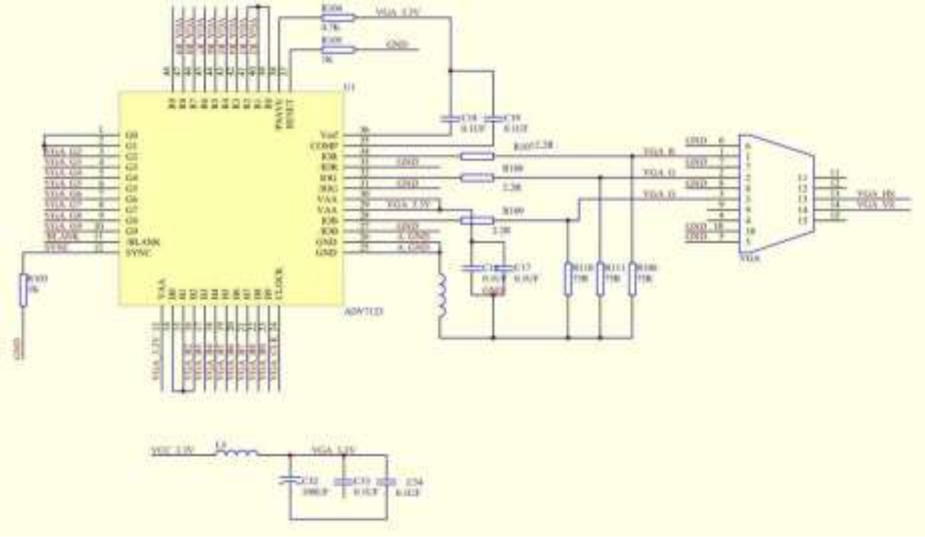

Fig 3: RGB888 VGA drive circuit diagram of ADV7123

The VGA interface conversion circuit based on ADC7123 is shown in fig. 3. Because only RGB888 is used in the design, the method of high-bit output and low-bit mask is adopted in the hardware circuit. When designing the hardware circuit, the layout and power scheme determine the quality of VGA analog signal transmission. In order to ensure the stability of DAC conversion, the ripple of supply voltage is small and the stability of video signal is high, the ADV7123 input power supply is filtered.

3.4 Execution Flow of FPGA Hardware Engineering

The execution flow of image processing IP in FPGA hardware engineering is shown in Figure 4.

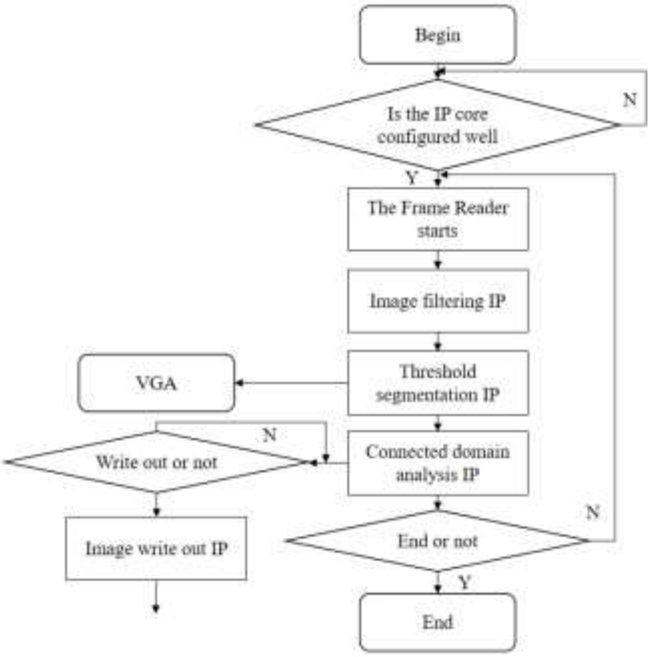

Fig 4: Execution flow of FPGA hardware engineering

First, wait for the user to configure the IP core. If there is an error, the IP core will be reconfigured until the IP core is configured normally. Then the user starts the Frame Reader core to start reading image data; Then, an image filtering IP core, a threshold segmentation IP core and a connected domain analysis IP core are continuously executed, and after the connected domain analysis IP core is executed, a user reads a processing result signal, when the signal is detected, the data writing IP core is executed, and the processing result is written into the HPS kernel space; Finally, judge whether it is finished, if it is not finished, continue to repeat steps 2, 3 and 4.

IV. SYSTEM SOFTWARE DESIGN

4.1 Program Design of VGA Display Module

VGA is mainly controlled by field signal line V and row signal line H, and RGB three lines are data analogy signal lines. The brightness of RGB display is controlled by the voltage. The display mode we use is 640×480, but the actual image size is 360×288, so we take out the 360×288 window as the display in this display mode.

First, the frequency of 640×480 display mode is generated, which is 24.5Mz, and we take it as 25M, and use PL inside FPGA for frequency doubling. In fact, the display size of 640×480 is 800×525, so we use 800×521 in the code implementation because of the slightly higher acquisition frequency, which basically makes up the advanced clock.

Generation of field signal: in line 521, the first two lines are in field invalid state, that is, in the first two lines, V sends a low level, and then it goes to a high level. After writing a picture, the number of lines becomes 0, and the line recording starts again, so the generated signal is a field signal.

Generation of line signal: 96 invalid pixels in 800 pixels, that is, the first 96 pixels H send a low level, then a high level, after writing one line, start writing the second line, and also send a high level after the first 96 pixels H send a low level, so the generated signal is a line signal.

4.2 Verilog HDL Design Flow of SDRAM Controller

Control register configuration, initialization command generation module, refresh request generation module, instruction arbiter, instruction decoding and data path are the internal structures of SDRAM controller. As shown in Figure 5.

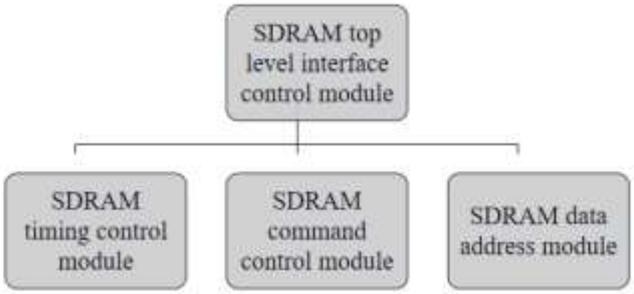
SDRAM top level interface control module
SDRAM timing control module
SDRAM command control module
SDRAM data address module

Fig 5: SDRAM module design block diagram

The FPGA implementation of SDRAM programming adopts the top-down modular approach. The top-level module only defines the interface, does not design sequential logic or combinational logic, and makes an upward simple package, bonding the three modules downward. SDRAM timing control module will refine the time relationship between request and response from initialization to normal operation and self-refresh, which is actually the design of holding time of each state. According to each state generated in the timing module, the SDRAM command control module will assign values to the interface signals of FPGA and SDRAM, so as to achieve the goal[10]. In the specified time state, the data address module is to read and write the data at the corresponding address of SDRAM.

The timing control module of SDRAM mainly completes the power-on initialization, timing refresh, read/write control and other state transitions of SDRAM. It contains two state machines, one for controlling the power-on initialization state and the other for controlling the state during normal operation. SDRAM command control module completes corresponding SDRAM control command and address output according to different state indications of timing control module. SDRAM data read/write module completes the control of SDRAM data bus according to the status indication of timing control module, and completes the data read/write of SDRAM in this module.

4.3 Write FPGA to HPS

Through FPGA-2-HPS data bus, FPGA writes data into the physical memory space in HPS. At this time, the data is stored in the kernel space of Linux operating system, which cannot be directly accessed by user program. Linux operating system provides character device mode. By writing character device driver, user program can copy the data in kernel space to user space.

Character equipment takes bytes as the minimum saving unit, and can be accessed by data stream. Character device driver mainly includes module initialization function, module clearing function and module custom function. The reference of the module custom function is saved in the data structure of type file operations. In this way, when an application operates a device file, the Linux system will find the file operations registered in the kernel according to the type of the device file and the master device number, thereby calling the custom function referenced in the file operations. The module custom functions implemented by the character device driver mainly include open, read and release. Open and release are to open the device file and release the space of the device file and its application, respectively. The read function calls the kernel function copy to user to copy the data from the applied memory space to the user space. The module clears the memory space requested when registering in the function.

4.4 Control Program Design

The DC motor designed in this system adopts PWM speed control. In the driving circuit, motor running direction signal Dir, motor speed regulation signal PWM and motor braking signal Brake are provided. The motor running direction signal diz and motor braking signal brake are directly controlled by the IO port of NiosII CPU according to the results of trajectory identification; the motor speed control signal PWM is directly generated by the corresponding hardware, but the PWM cycle and duty cycle need to be set in the program.

4.5 Problems Needing Attention in FPGA Software Design and Debugging

In the process of FPGA program development, for the sake of system safety and high efficiency, it is necessary to pay attention to avoid burr. Most feedback circuits, counters and combinational circuits may have a large number of potential glitches. However, some glitches do not have to be avoided. For example, the D input of flip-flop will not be harmful to the system as long as the glitches meet the setup time and hold time of data and do not appear on the rising edge of the clock. When glitch signal is used as control signal, start signal, CLEAR signal of flip-flop, PRESET signal, handshake signal, input signal of latch, clock input signal CLK, etc., serious logic error may occur. Therefore, it is imperative to eliminate burrs, but there are many burrs that can't be found out, so we have to avoid them as much as possible in logic design.

V. PRACTICAL APPLICATION EXPERIMENT

After algorithm verification and system verification, this embedded machine vision inspection system is applied to Dleta parallel robot sorting platform to check whether the system can correctly locate the target position. Verify the accuracy of target positioning of FPGA embedded vision system in actual working environment. The experimental scheme is as follows: the target objects are placed on the conveyor belt at different conveyor belt speeds and different densities, the camera collects images and saves them as pictures, and the self-increasing numbers are used as file names. At the same time, record the target positioning result of FPGA embedded vision system, and use the picture name as the record serial number, so as to correspond to the saved image one by one. On the other hand, PC image processing program is used to process the saved images, and the processing results are compared with those of FPGA embedded system, and the correct number of target positioning in FPGA embedded system is counted. The pixel error of the two positioning results is less than 5 pixels, which is regarded as correct positioning.

The experimental results are shown in Table III. From the experimental results, it can be seen that with the increase of conveyor belt speed and target objects, the success rate of system target positioning gradually decreases. The main reason is that FPGA embedded vision system is not robust enough, and there is frame loss phenomenon. Secondly, as the speed of the conveyor belt increases, more noise is generated, and data confusion occurs in FPGA image processing algorithm.

TABLE III. Experimental results of target location

Conveyor belt speed (m/s)	Total number of targets (pieces)	Number of successful positioning (pieces)	Success rate (%)
0.0271	100	93	95
0.0663	100	88	89
0.1057	100	83	85

From the experimental results, it can be seen that the positioning accuracy of the embedded vision system based on FPGA in this subject can meet the requirements of practical industrial application. Of course, the embedded vision system based on FPGA is not good enough in stability, but it has laid a solid foundation for future research and opened up a new route. I believe it can make a further breakthrough under further research.

VI. CONCLUSIONS

In the field of industrial automation, the automation scheme of industrial robot combined with machine vision has been widely used. The traditional machine vision inspection system generally adopts the mode of 'hardware and software cooperation', which has the disadvantages of large volume, high power consumption, poor real-time performance and inflexible application. Based on the field of industrial automation, aiming at the shortcomings of traditional machine vision schemes, this paper designs a FPGA embedded machine vision scheme. Compared with the traditional machine vision inspection system based on PC, this system has a series of advantages such as stable inspection performance, low power consumption and low cost. The FPGA implementation of the system is described in detail in this paper, and the software and hardware work together to form the whole system. Through practical application test, the experimental results show that the system designed in this paper basically achieves the expected goal.

ACKNOWLEDGMENTS

This research was supported by the Hubei Key Laboratory of Intelligent Robot of China (Grant No. HBIRL202009).

REFERENCES

[1] Chen Y, Shu Y, Li X, et al. (2021) Research on detection algorithm of lithium battery surface defects based on embedded machine vision. Journal of Intelligent and Fuzzy Systems, (5): 1-9.

[2] Wen J J, Fan H, Zhu J Z, et al. (2018) Research on Tool Image Preprocessing System Based on FPGA. Microcomputer and Applications, 037 (009): 94-96, 100.

[3] Kang S J, Yang Z H, Yu L L, et al. (2017) Research on DQPSK carrier synchronization based on FPGA. Journal of Information Hiding and Multimedia Signal Processing, 8 (1): 138-147.

[4] Yip W, Zhou Z W, Wang Y W (2017) Research on Data Processing Technology Based on FPGA for Auto Collimating System. Sensors and Microsystems, 036 (002): 46-48, 52.

[5] Kyrkou C, Bouganis C S, Theocharides T, et al. (2017) Embedded Hardware-Efficient Real-Time Classification with Cascade Support Vector Machines. IEEE Transactions on Neural Networks & Learning Systems, 27(1): 99-112.

[6] Long X, Hu S, Hu Y, et al. (2019) An FPGA-Based Ultra-High-Speed Object Detection Algorithm with Multi-Frame Information Fusion. Sensors, 19 (17): 3707.

[7] Zou X F, Li X Y (2019) Research and application of multifunction testing technology based on FPGA dynamic reconfiguration. Modern Manufacturing Engineering, 000 (003): 102-107.

[8] Liang Y, Li K L, Bi F H, et al. (2020) Research on LFMCW Radar Velocity Ranging Optimization System Based on FPGA - ScienceDirect. Procedia Computer Science, 166:187-194.

[9] Liu X, Zhang M, Luo Y, et al. (2017) Machine vision image acquisition hardware system based on FPGA. Agro Food Industry Hi Tech, 28 (1): 3490-3493.

[10] Xu B, Liu L, Wu X (2017) A new method and simulation of image edge detection based on sobel operator and FPGA Design. Boletin Tecnico/Technical Bulletin, 55 (11): 285-292.