

Face-Recognition Hardware Implementation Based On SOPC

Cheng He, Chunchun Sui, Jing Chen, Cao Xixin

(School of Software and Microelectronic, Peking University, Beijing, China)

Email: cxx@ss.pku.edu.cn

Abstract— As the rapid development of computer science and pattern recognition, visual technology is widely applied in industry and daily life. This paper provides a face-recognition hardware implementation based on SOPC, in which custom peripherals, manufacture-provided IP and NiosII processor are connected by Avalon Memory-Mapped Interface. Real time data is transported through Avalon Streaming Interface between custom peripherals without the intervening of CPU. Two kinds of hardware acceleration are used to improve performance, C2H and Custom instructions, which can accelerate the speed of face-recognition up to about 50 and 90 times, respectively.

Key Word— Face-recognition, SOPC, NiosII, Hardware acceleration

1. Introduction

70% of information people receive is through vision, as visual signals contain more information than that of audio ones with a series of advantages like precision, visibility, vividness, concretion, high efficiency and wide application, and so on. Therefore, visual technology is widely applied to such areas as radiated television, communication, entertainment and computer, and is playing a more and more important role. Therewith, processing technique of visual signals has become the focus of present researches, with image collection as its first step.

At present, image processing based on embedded system is frequently applied to pattern recognition of images, including that of bar code, biological characteristics, animated objects in intelligent transportation, handwriting cursive recognition, and so on. Image recognition technology is an extension of human visual cognition and the base of artificial intelligence. Model recognition technology and its products on the basis of PC, with relatively fast recognition speed and high accuracy is already mature. As model recognition system

based on embedded system is restricted by arithmetic speed and memory resources, mature recognition algorithm can not be transplanted into embedded environment, which reduces the speed and accuracy of embedded face recognition system.

The existing embedded model recognition system usually employs the framework of DSP and ARM, which is of high cost and long development cycle. This paper suggests a hardware implementation of image acquisition system based on SOPC platform, which takes the advantage SOPC technology to shorten development cycle and custom peripherals to realize image acquisition and display as peripherals are connected through Avalon Streaming Interface, which releases processor to the greatest extent. Besides, this image acquisition platform can be employed in model recognition system like embedded face recognition system.

2. An introduction to SOPC technology and its development platform

2.1 SOPC technology

SOPC (System-on-a-Programmable-Chip) technology was first raised by ALTERA in 2000 and its corresponding developing software QuartusII and NiosII IDE were promoted at the same time. SOPC is SOC based on FPGA solution. Compared with SOC of ASIC, its system and developing technology are of unique characteristics.

On the one hand, it's a system on chip (SOC), that is, main logical functions of the whole system are completed by a single chip; on the other hand, it's a programmable system, with flexible design that is expandable and upgradeable, and programmable functions of software and hardware on system. That is to say,

SOPC is composed of all the advantages of SOC, PLD and FPGA.

SOPC, a combination of PLD and ASIC technologies, compared with the expensive ASIC, it has the advantage of flexible application of hard core or soft core CPU, DSP, memory, peripheral I/O and programmable logical SOPC chip, low cost and short developing cycle, therefore, is known as “the future of semiconductor industry”.

2.2 An introduction to Avalon bus:

There are two types of Altera Avalon bus: Avalon Memory-Mapped Interface and Avalon Streaming Interface. The former is used to connect Memory-Mapped peripherals and the system interconnect fabric, with similar functionality of AMBA bus while the latter is based on two peripherals to define a standard, flexible, and modular protocol for data transfers from a source interface to a sink interface, with the similar functionality of DMA.

2.3 An introduction to developing board

This system employs the Altera DE2 for development, which consists of the following resources:

FPGA

- Cyclone II EP2C35F672C6 FPGA

I/O Devices

- Video Out (VGA 10-bit DAC)
- 18 toggle switches
- 4 debounced pushbutton switches
- 18 red LEDs, 9 green LEDs

Memory

- 8-MB SDRAM, 512-KB SRAM, 4-MB Flash
- SD memory card slot

2.4 developing tools

QuartusII 8.0, SOPC Builder, NIOSII IDE

3. Hardware design of face recognition system

3.1 system diagram

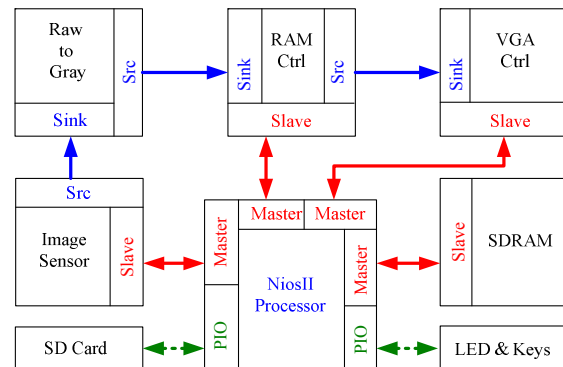


Figure 1. system diagram of hardware

The whole hardware platform is composed of camera controller, format conversion, RAM, VGA and SD Card controller modules. As all the modules between camera and display which can do necessary format conversion and caching are connected by Avalon Stream Interface which contains read-write and effective signals, real-time display of image can be realized with no consumption of CPU MIPS. Meanwhile, all the modules and Nios are linked by Avalon Memory-Mapped Interface, through which the Nios can configure peripherals, receive data for processing and send image to display. The blue arrows in the diagram stand for Streaming Interface, red ones Memory-Mapped Interface, and green ones Programmable IO.

3.2 Raw to Gray modular

Camera adopts Micron MT9M011 Image Sensor, Active Pixels 1,280H x 1,024V, Color Filter Array RGB Bayer Pattern, Frame Rate:

SXGA (1280 x 1024) Programmable up to 15 fps
VGA (640 x 480) Programmable up to 60 fps
CIF (352 x 288) Programmable up to 150 fps

Image Sensor which outputs Bayer Pattern by row, should convert into RGB format first for Nios processing and VGA display.

Suppose one point's output value $V(x, y)$, then its RGB component is listed as follows:

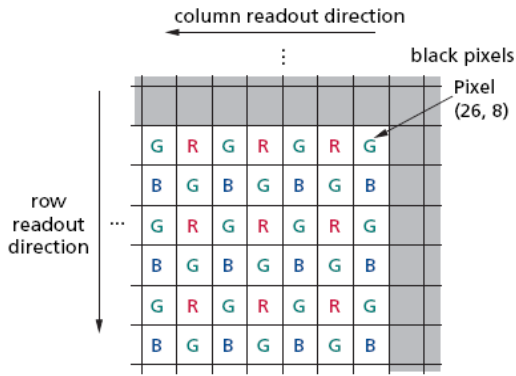


Figure 2. data sequence of image sensor

3.3 SRAM Control module

As image acquisition system needs to realize real-time display of Image Sensor's image capture, while frame rate of Image Sensor's image output (related to image size and time of exposure, up to 60 frame per second) is different from the refresh rate of VGA display (fixed at 60 frame per second), a one-image-large buffer which can be realized by SRAM on DE2 developing board is in need as cache. As read and write do not operate at the same time, pingpong operation is the best solution.

The Pixel Clock provided by Image Sensor for output data synchronization is 25M. As there are line/frame signals, the Stream Interface between Raw2RGB and SRAM Ctrl can not receive effective data every cycle, so the frequency of writing signal of Input FIFO on SRAM Ctrl modular is less than 25M. When the refresh rate of VGA is 60 Hz, Pixel Clock is 25M. Also due to the existence of line/frame signal, the frequency of reading signal of Output FIFO is less than 25M. Therefore, so long as SRAM works at the frequency of more than 50M, pingpong operation can be completed.

At the time of image collection and display, Nios will read an image for processing and output one for display. SRAM can be designed as "T" framework, so that when Nios does not need to read or write Image Buffer, SRAM Ctrl

can do so according to pingpong operation; when Nios needs to read and write, pingpong operation continues to work but produces no SRAM command signals which will be controlled by Nios.

Table 1 formulas used in color recovery

VGA coordinate	X odd Y even	X even Y even	X odd Y odd	X even Y odd
VGA color	R	G	G	B
Reconstruct R	$V(x,y)$	$V(x-1,y)$	$V(x,y-1)$	$V(x-1,y-1)$
Reconstruct G	$V(x-1, y)/2 + V(x,y-1)/2$	$V(x,y)/2 + V(x-1,y-1)/2$	$V(x,y)/2 + V(x-1,y-1)/2$	$V(x-1,y)/2 + V(x,y-1)/2$
Reconstruct B	$V(x-1, y-1)$	$V(x,y-1)$	$V(x-1,y)$	$V(x,y)$

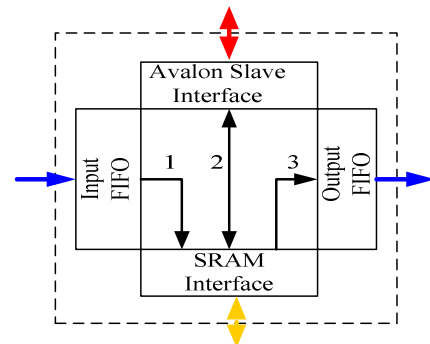


Figure 3. data flow of SRAM Ctrl module

The size of image VGA displays is 640 x 480 which is always completely stored in SRAM. Suppose the size of FIFO is S, after reset, SRAM will be in the state of IDLE. When the data in Input FIFO surpass $S/2$, SRAM Ctrl will jump to the state of Read Input FIFO, and then SRAM Ctrl will read Input FIFO and write data into SRAM, then writing address is increased by one. After repeating $S/2$ times, SRAM Ctrl will jump to IDLE state. When data in Output FIFO is less than $S/2$, SRAM Ctrl will jump to Write Output FIFO state, and then read SRAM and write data into Output FIFO. SRAM reading address is increased by one every time, after repeating $S/2$ times, SRAM Ctrl jumps back to IDLE state. When writes the last pixel of an image, the write point of SRAM rests, so

does the read one while reading. Anytime when Nios visits Buffer, SRAM Ctrl goes into the state of Nios Access Buffer, and then the control signals (read-write, chip selection and enabling) and data will be produced by decoding Avalon Memory-Mapped Interface, at that time, the input of Input FIFO and output of Output FIFO do not stop. In order to keep the correct display of pixel, Reading of Input FIFO and writing of Output FIFO continues, with the exception that data get from the Input FIFO will be abandoned and data write to the Output FIFO are all zeros.

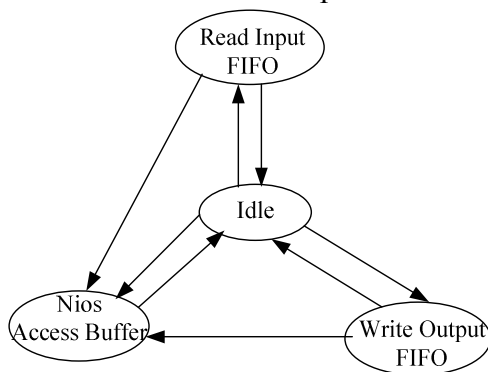


Figure 4. FSM diagram of SRAM Ctrl module

4. Performance and characteristics of system

This system can realize the real-time acquisition and display of image, with 60 frames per second in the VGA format, with no consumption of CPU MIPS. Nios can capture real-time images for processing and display, read images from SD card for display or storing them into SD card. Meanwhile, due to differences in specific algorithms, two means can be adopted to realize the speeding up of hardware, so as to improve performance. One is custom command which realize complex algorithm through hardware, while the other one is C2H (C-to-Hardware) which is a C compiler of Nios that speeds up time-cost functions at the cost of hardware resources. C2H is the easiest way, with another name right-click-acceleration. With the flexibility characteristic of SOPC, the components in this platform can be modified and reused, and some proven IP like FFT and DCT with Avalon interface and be integrated into this platform. The table below is the

performance of this platform when transplant an Open CV face-recognition algorithm. When the picture size of a face is 32 x 24, this platform needs 1.2 seconds to recognize it.

Table 2 performance of acceleration

Performance (seconds/picture)		Picture size		
		32x24	64x48	128x96
Acceleration method	Software (no acceleration)	112	521	1824
	C2H	1.8	8.2	39
	Custom Instructions	1.2	4.9	23

5. Conclusion

C2H and Custom Instructions are two ways of improving program efficiency at the expense of hardware resources, they both need to analyze the algorithm first and find out which functions and arithmetic consume much time and execute much often. But these two ways differ in two points:

1. Different efforts: The former is the easiest way, which can be implemented through the right-click menu, while the latter need to convert the algorithm to hardware description language by oneself.

2. Different efficiency: We can obtain from the result comparing table that the former can speed up about 50 times on the average while the latter 90 times. That's because the former is implemented by C2H compiler to convert key-function to hardware while the latter is done manually, which often has higher efficiency.

References

- [1] Clifford E. Cummings, Simulation and Synthesis Techniques for Asynchronous FIFO Design, Sunburst Design, Inc., SNUG San Jose 2002
- [2] Pan Song,, A practical tutorial on SoPC, Tsinghua university press, 2005
- [3] Zhao Chuan, Research on design methodology of software and

hardware cooperation, Computer engineering and design, 2003, 24

[4] NiosII Processor Reference Book, ALTERA, 2008

[5] Cyclone Device Handbook, ALTERA, 2008