FPGA-Based High Definition Image Processing System

Xinxin He and Linbo Tang(✉)

Beijing Key Laboratory of Embedded Real-Time Information Processing Technology, Beijing Institute of Technology, Beijing, China
hxx9418@163.com, tanglinbo@bit.edu.cn

Abstract. With the continuous development of mobile communication and the Internet, people's requirements for the speed and quality of digital image processing are increasingly improved. A large amount of high-speed, parallel video stream data needs to be processed in real time, especially in video image processing. The author presents a method of high-definition image transmission and processing system based on FPGA. The system uses FPGA as the main controller, consisting of front-end HDMI video receiving module, image fast median filtering processing module, image ping pong Storage module and HD-SDI video display module through hardware description language programming, effectively realizing real-time video capture, transmission and display. It has been verified that the processing and transmission of digital video signals in this system is stable and reliable. The system also has a series of advantages such as low power consumption, low cost, flexible design, and easy expansion and has been applied in practical engineering.

Keywords: FPGA · HDMI · SDI · Advanced median filtering

1 Introduction

Digital image processing technology has developed rapidly in contemporary society and plays an irreplaceable role and has been widely used in aerospace, communications, medical and industrial production. With the continuous development of mobile communication and the Internet, people's requirements for the speed and quality of digital image processing are increasingly improved. A large amount of high-speed, parallel video stream data needs to be processed in real time, especially in video image processing, and FPGA can exert its unique advantages.

As the amount of image data grows, the computation of algorithm that is implemented in FPGA becomes increasingly large. The traditional method of increasing the clock frequency is difficult to solve the system power consumption problem and the cache speed problem, which are caused by the large computation. Because of the structural characteristics and design rules of FPGA in image processing, there are many common design ideas and techniques can be used: serial-to-parallel conversion, pipeline operation, Ping-Pong operation. These design techniques increases not only the efficiency of code execution, but also the speed of code calculations and the stability of the design. In this paper, these design ideas have been all applied in the system [1].

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 208–219, 2019.
https://doi.org/10.1007/978-3-030-19156-6_20

(1) Serial and parallel conversion. Spatial parallelism based on image processing divides the image into several parts, and the same algorithm is used for each part at the same time. The calculation time of all data processing is long, and processing at the same time saves a lot of calculation time, but the storage resources required for spatial parallel processing are large. Changing serial data to parallel data processing is done by increasing the consumption of logic resources in exchange for processing speed, which can reduce processing time and improve design performance. Conversely, changing parallel data to serial can save logic resources [2].
(2) Ping Pong Cache. The Ping-Pong cache caches data into two different spaces, buffers data in one space, and reads data in another space, usually used in the processing of high-speed data streams. The most important thing to achieve Ping-Pong cache is the mutual cooperation of the two spaces to achieve seamless switching.
(3) Pipeline operation. Time parallelism is a pipeline idea that divides image processing into different modules, each of which runs separately and is irrelevant in terms of computation time. When processing video stream image data, the data is pipelined to the FPGA and the image processing through a multi-stage pipeline greatly increases data throughput.

Due to the characteristics of video images, the characteristics of image processing algorithms and the processing techniques of FPGA, FPGA become the fastest and best performing hardware platform for real-time video image processing [3].

In this paper, an FPGA-based high-definition digital image transmission and processing system is designed, which can convert the HD video with resolution of 1080P60 received by the front-end HDMI receiving module. It can also use DDR3 to Ping-Pong the image data and display the processed video in real time on the terminal device supporting the SDI interface. At the same time, in order to meet the image quality and speed requirements, a fast median filtering process is added to the system to ensure the image clarity and real-time processing. The system has the advantages of low cost, flexibility, and ease of expansion.

2 System Structure Design

The high-definition video transmission system proposed in this paper uses XILINX's A7 series FPGA as the core device. First, the video is sent to the FPGA through the HDMI interface for decoding, and 24-bit RGB image data and the horizontal and field synchronizing signal are extracted therefrom. Then, the fast median filtering algorithm is used to denoise the video signal, and the SRAM is used for the Ping-Pong buffer. Finally, the SDI data stream is generated and displayed through the SDI output.

The entire structure framework of the system is shown in the Fig. 1.

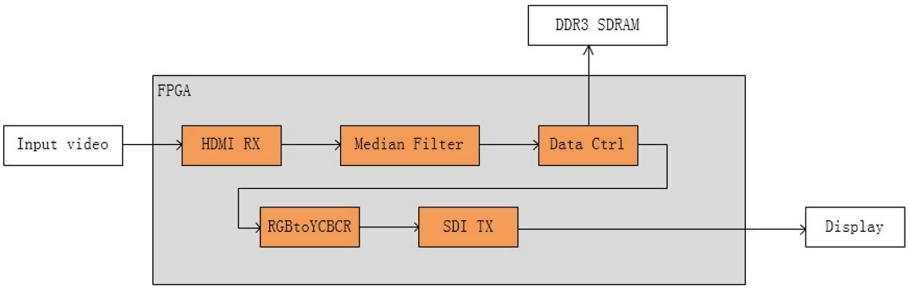

Fig. 1. The entire structure framework of the system

The entire design of the system can be divided into the following modules:

(1) HDMI receiving module: This module will extract valid 24-bit RGB image signal and horizontal and field synchronizing signal from the received HDMI format video;
(2) Median filter module: Image preprocessing module, which can effectively remove noise and smooth the image. Using the fast median filtering algorithm can remove the noise without degrading the edges of the image, better maintain the clarity of the image, and meet the real-time requirements;
(3) Storage module: In the real-time video processing system, since the video processing code execution process is variable speed, it is necessary to provide a certain buffer circuit between the input and output of the image.
(4) SDI output module: This module is mainly divided into two parts: image format conversion and SDI data stream generation. Since the SDI interface has its specific output standard, the RGB format needs to be converted to the YCbCr format, and the SDI data stream output is generated according to the interface protocol.

3 Core Module Introduction

3.1 HDMI Receiving Module

High Definition Multimedia Interface (HDMI) is a dedicated digital video/audio interface technology suitable for image transmission, which can transmit audio and video signals at the same time. The maximum data transmission speed of HDMI is 2.25 GB/s and a 1080p video requirement is less than 0.5 GB/s. Therefore, HDMI can be used as a receiving interface for HD video [4].

HDMI uses the Time Minimized Differential Signal (TMDS) transmission technology. TMDS, also known as a transiently modulated differential signal, is a differential signaling mechanism that uses a differential transmission. When HDMI is encoded by TMDS technology, the video signal is divided into R, G, and B data signals and two horizontal and field control signals H and V. These five signals are transmitted in four channels. Each of R, G, and B uses one channel. The horizontal and field control signal is transmitted in the B signal channel, and the audio signal is transmitted

in the R and G signal channels. There is also a separate channel for transmitting clock data. One TMDS channel can transmit 10 bits of data per clock cycle.

Therefore, it is necessary to decode the video transmitted through the HDMI interface to generate a data stream in the VGA format. The timing diagram of the video decoding module is illustrated in the Fig. 2.

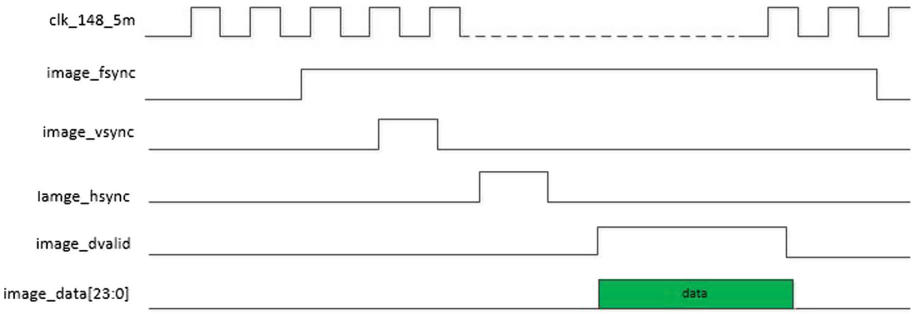

Fig. 2. The timing diagram of the video decoding module

There are three processes in the HDMI transmission process: video data period, data island period, and control period. Each stage transmits different data and has different functions. When HDMI works in the video data period, the three TMDS data channels transmit video data; when HDMI works in the data island period, the TMDS channel transmits audio data and auxiliary data in the form of data packets; when HDMI works during the control period, it transmits the boot information, indicating that no valid data is currently transmitted [5].

3.2 Image Processing Module

Median filtering is a common image preprocessing method that effectively removes noise and smoothes the image. Compared with the mean filter and other linear filters, it can denoise without degrading the edges of the image and better maintain the clarity of the image [6].

Median Filtering Principle

The basic principle of median filtering: the gray value of each pixel is set to the median value of the gray value of all pixels in its neighborhood window [7]. The two-dimensional median filtering is based on a two-dimensional sliding window, which in turn slides from left to right and top to bottom at the pixel points of the image. At each pixel, the gray values in the window are sorted to produce a monotone grayscale sequence with the median value as the output of this pixel:

$$g(x, y) = Med\{f(x - i, y - j)\};\ subject\ to(i, j) \in s \tag{1}$$

where $f\ (x,\ y)$ is the original image, $g\ (x,\ y)$ is the processed image, s is the two-dimensional template [8].

Typical Median Filtering Optimization Algorithm

The core of median filtering is sorting, and the sorting algorithm uses the most traditional bubbling method. A square filter window with N × N pixels is subjected to about $N^2 * (N^2–1)/2$ comparisons every time it is run. The algorithm complexity is o (N^4), and the commonly used 3 × 3 filter window needs to be sorted 36 times. Up to now, there are some mature optimization algorithms at home and abroad.

Quick Sort Method. Among the 9 pixels of the 3 × 3 window, one is set to D, and the remaining 8 pixels are compared with it. The left side is smaller than D, and the left side is larger than D. When the number of the left side m = 4, D is the median. When m < 4, select a number on the right side and repeat the previous step until the sum of the numbers on the left is equal to 4, returning the median value; the same is true when m > 4. For 3 × 3 square windows, 30 comparisons are required [9].

Incomplete Bubble Sorting. Window pixels are sorted by bubble method, and when compared to the middle position, that is, the (N + 1)/2th pixel, stop the sort to obtain the median. For a 3 × 3 square window, 30 comparisons are required [10].

Compared to the 36 binary comparisons of the traditional bubbling method, the two optimization algorithms are reduced by 6 times. However, the above algorithm is not stable enough and the number of comparisons is still too much. The advanced algorithm of this paper overcomes the above shortcomings, that is, the comparison times are the least and the stability is better.

Implementation Methodology of the Advanced Median Filtering Algorithm

The advanced median filtering algorithm has three main steps:

(1) Sort 3 pixels of each row in the 3 × 3 window in ascending order and respectively obtain 3 ordered sequences;
(2) Sort the three ordered sequences in ascending order according to their median sizes;
(3) Compare the minimum value in the max row, the maximum value in the min row, and all the three values in the mid row. And the median of them is the required median.

The sorting process is shown in the Fig. 3.

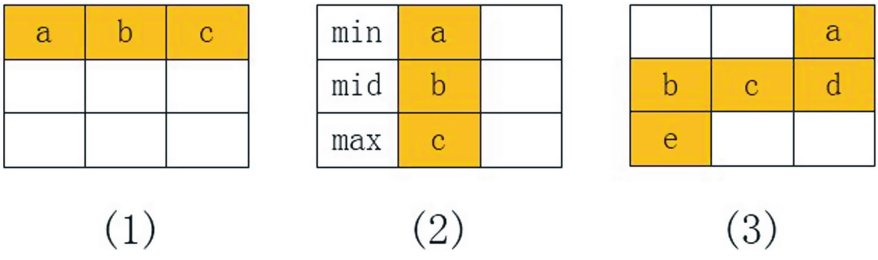

Fig. 3. The sorting process of the advanced median filtering algorithm

Implementing the advanced median filtering algorithm on an FPGA requires three main modules: a 3 × 3 window generation module, a median filter module, and a row and column counting module. The implementation scheme is illustrated in Fig. 4.

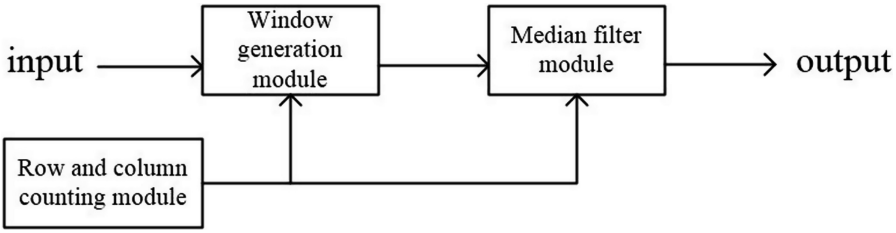

Fig. 4. The implementation scheme of the advanced median filtering algorithm

The function of the 3 × 3 window generation module is to generate a 3 × 3 scan window. When the window passes through the image, a 3 × 3 pixel value is read and passed to the system for subsequent data processing. The window scans from left to right, from top to bottom, until the entire frame is read.

The function of the median filtering module is to process the pixels collected by the 3 × 3 window according to the advanced median filtering algorithm proposed in this paper, and output the processed pixels.

The function of the row and column counting module is to determine whether the center pixel is located at the edge of the image. For the edge portion, the image cannot be covered with a 3 × 3 window, and the filtered output is meaningless. The usual processing method is to output the pixel values of the edge to "0".

Therefore, the system in this paper is designed with the advanced median filtering algorithm, which utilizes parallel and pipeline processing methods, and requires 19 comparisons to complete a median filter. After a latency of 9 clocks, a result is calculated for each clock, which greatly speeds up the calculation of the median filter. The calculation principle is shown in Fig. 5.

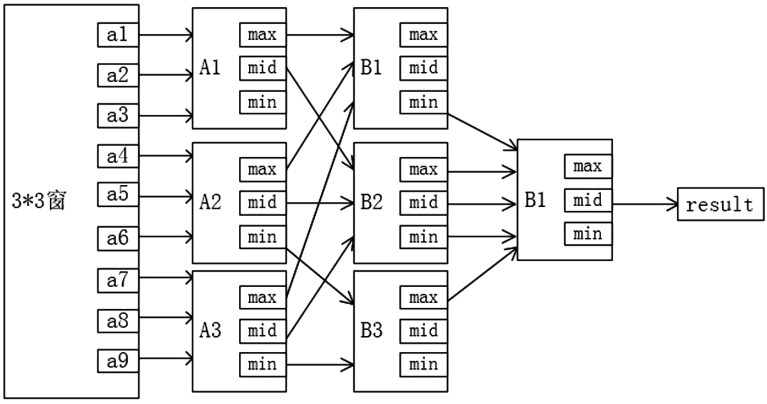

Fig. 5. The calculation principle of the advanced median filtering algorithm

3.3 Storage Module

In a real-time video processing system, since the video processing code execution process is variable, it is necessary to provide a certain buffer circuit between the input and output of the image. The more commonly used cache circuits include: FIFO structure, dual-port RAM structure [11] and Ping-Pong cache structure.

Although the FIFO can perform double-ended reads and writes, it requires that the read and write data must be first-in, first-out, and cannot be accessed arbitrarily. In terms of capacity, dual-port RAM can be made larger than FIFO, but in terms of total capacity, these two structures are based on the SRAM integrated in the controller, so the capacity is relatively limited. For high-definition image processing systems, dual-port RAM and FIFO cannot meet the demand.

The Ping-Pong cache allocates data to two cache units simultaneously through a cache switching unit. The write data is buffered to cache unit 1 during the first read and write cycle. At the beginning of the second read/write cycle, the write data is buffered to the cache unit 2 by the cache switch. At the same time, the data of the first read/write cycle of the cache unit 1 is selected by the output switch and read, and then sent to the processor operation unit for processing. In the third read/write cycle, the cached read unit is switched again, so that the cycle is repeated.

Because the speed of RAM reading is much faster than the speed of writing, the Ping-Pong cache can not only meet the real-time requirements of high-speed video streams, but also give the CPU enough time to process data, which solves the synchronization between data modules effectively. Storage module framework is illustrated in Fig. 6.

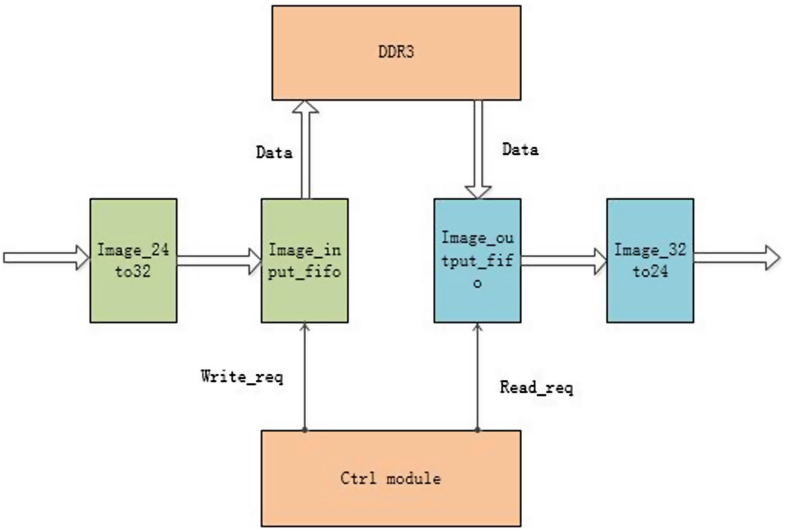

Fig. 6. Storage module framework

3.4 SDI Output

Serial Digital Interface (SDI) is a video standard interface defined by the Society of Motion Picture and Television Engineers (SMPTE) for serial transmission of uncompressed digital video and audio signals over a single coaxial cable. SDI video is divided into standard definition (SD), high definition (HD) and 3G.

The HD-SDI interface has a transfer rate of 1.485/1.4835 Gbps or higher and is compliant with the SMPTE 292M standard. It transmits a 4:2:2 serial uncompressed digital component signal (YCrCb signal) with a transmission rate of 270 Mbps and a maximum transmission distance of 300 m [12].

The SMPTE 292M video standard is an image transmission standard for internal synchronization, which refers to embedding the image horizontal and field synchronizing signal in the image data, and no external synchronizing signal is needed. There are two types of image timing reference codes. One is at the start of the image valid data block (SAV) for the start control of the active image, and the other is at the end of the valid data block (EAV), which is also the starting position of the digital horizontal blanking. Two timing references are used to control the start and end of a line of the image [13]. The 1080P SDI image data format is shown in Fig. 7.

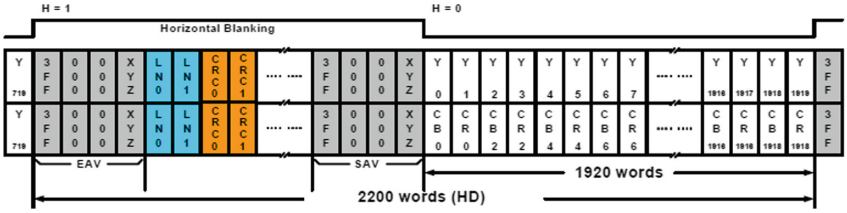

Fig. 7. The 1080P SDI image data format

The effective image data of each line is 1920 words, the other 280 words are horizontal blanking signals. EAV is the end mark, and SAV is the start mark. Through the previous processing, we get only one frame of 1920 * 1080 effective image data. But when performing SDI transmission, image format conversion and standard SDI data stream are required to conform to the transmission protocol.

RGB to YCBCR

YCbCr, a type of color space, is often used for continuous image processing in movies or digital photography systems. Where Y is the brightness, which is the gray level value. “Brightness” is established by RGB input signals by superimposing specific parts of the RGB signal. Cb reflects the difference between the blue portion of the RGB

input signal and the luminance value of the RGB signal. Cr reflects the difference between the red portion of the RGB input signal and the luminance value of the RGB signal [14].

The conversion formula is shown as follows:

$$Y = (0.299\,R + 0.587\,G + 0.114\,B) \tag{2}$$

$$Cr = (0.511\,R - 0.428\,G - 0.083\,B) + 128 \tag{3}$$

$$Cb = (-0.172\,R - 0.399\,G + 0.511\,B) + 128 \tag{4}$$

In the process of RGB to YCb Cr algorithm, since the FPGA cannot perform floating-point operations, it is necessary to first expand the right end of the whole equation by 256 times and then shift the right side by 8 bits, so that it can be processed by multiplication and addition operations suitable for FPGA.

$$Y = [(76\,R + 150\,G + 16\,B) \gg 8] \tag{5}$$

$$Cr = [(-131\,R - 109\,G + 21\,B) \gg 8] + 128 \tag{6}$$

$$Cb = [(44\,R - 102\,G - 131\,B) \gg 8] + 128 \tag{7}$$

Each step of the operation is performed simultaneously and calculated directly in the register, which is the pipeline design that can achieve hardware acceleration. This design technique is the essence of FPGA hardware acceleration, which is a common and very important algorithm implementation idea in FPGA design systems.

Program execution process: the first stage pipeline calculates all multiplications; the second stage pipeline calculates all additions, adding positive and negative separately; the third stage pipeline calculates the final sum, and if the result is negative, it takes 0.

Generate SDI Data Stream

The format of a complete 1080P image is as follows: a total of 1125 lines, each line has 2200 image data. The 1920 words in the 42–1121 line are video data, and the other parts are blanking areas. Lines 1–41 and 1122–1125 are field blanking areas, and 1–280 data per line is a horizontal blanking area. For the field blanking area, we can directly output white. For the horizontal blanking area, we need to meet the format of Fig. 7, adding the start signal of frame, end signal of frame, line count signal and other signals [15]. The structure of one frame image is shown in Fig. 8, where the color portion represents video data.

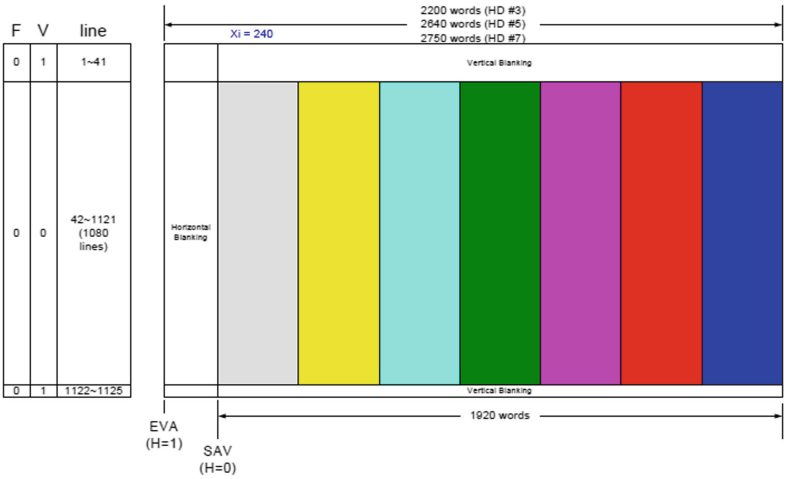

Fig. 8. The structure of one frame image (Color figure online)

4 Practical Application and Testing

This design uses ADLINK's SDI video capture card for image display testing. After testing, the system can send 1080P HD SDI video images of 60 frames per second and display the video correctly on the PC (Fig. 9). It can be seen from the test results Fig. 10 that the image is clear and stable, and the display effect is good.

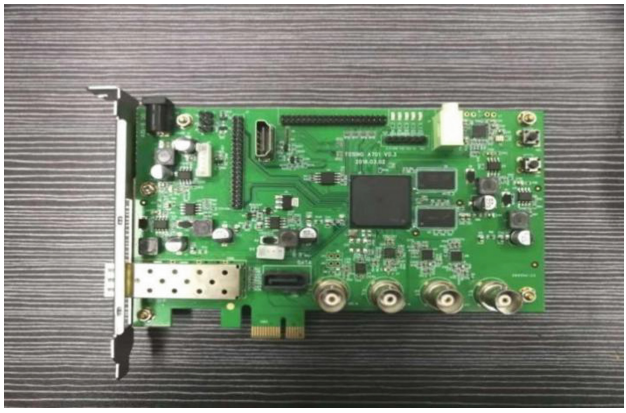

Fig. 9. Experiment environment

Fig. 10. Result for image processing system

5 Conclusion

In this paper, an FPGA-based high-definition digital image transmission and processing system is designed and implemented. The system realizes the conversion of HDMI to HD-SDI, and the advanced median filtering and DDR buffer are added, which not only improves the processing speed and quality of image, but also provides a foundation for implementing image processing in FPGA later. Through the experiment verification, the system has a series of advantages, such as low power consumption, low cost, flexible design, and easy expansion, which can output clear and stable processed video. Currently, the system has been applied to practical engineering.

References

1. Sun, J.: Research on high speed image compression technology based on parallel pipeline structure. Doctor Thesis, Graduate School of the Chinese Academy of Sciences, Beijing, China (2015)
2. Hu, X.: Research of hardware design for video compression algorithm based on FPGA. Master Thesis, Xidian University, Xi'an, Shaanxi, China (2013)
3. Yu, J.: FPGA realization of video compression-related problems. Comput. Knowl. Technol. **5**(24), 6771–6773 (2009)
4. Pan, L.: Design of image processing system for HDMI's video stream based on FPGA. Res. Explor. Lab. **34**(10), 77–78 (2015)
5. Xiao, J.: Design of video data codec based on HDMI. Electron. Des. Eng. **24**(13), 190–193 (2016)
6. Xu, Y.: Real-time image capturing and processing of seam and pool during robotic welding process. Ind. Robot: Int. J. **39**(5), 513–523 (2012)
7. Xu, F.: Feature extraction of acoustic emission signals based on median filter-singular value decomposition and empirical mode decomposition. Chin. J. Sci. Instrum. **32**(11), 2712–2719 (2011)
8. Shen, X.: Research of the advanced median filtering algorithm based on FPGA. Microelectron. Comput. **31**(1), 21–24 (2014)

9. Hou, F.: Image median filter algorithm and FPGA implementation. Microcomput. Inf. **27**(1), 69–71 (2011)
10. Huang, H.: Road recognition and tracking for intelligent vehicle based on SOPC. Chin. J. Sci. Instrum. **33**(2), 321–326 (2012)
11. Zhang, S.: Design and implementation of HDMI high-definition video editing based on Leonardo DaVinci. Microelectron. Comput. **34**(6), 54–57 (2017)
12. Interface for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 Level of Recommendation ITURBT.601 (ITU-RBT.656) (2010)
13. Television 10bit 4:2:2 component and 4FSC composite digital signal serial interface (SMPTE-259M) (2011)
14. Gonzalez, R., Woods, R., Stevenl, E.D.: Digital Image Processing Using Matlab, 2nd edn. Tsinghua University Press, Beijing (2013)
15. Liu, L.: FPGA**-**based SD**-**SDI transmission system design. Electron. Des. Eng. **25**(23), 94–97 (2017)