Design of Video Image Low-Pass Filtering Process Based on FPGA

Xiaojing Zhang and Hua Wang
Beijing Polytechnic, Communication University of China
Email: dky_zxj@sina.com; 1014690221@qq.com

Abstract. A real-time video low-pass filter processing system is designed and implemented in this paper through VHDL language programming, applying Altera Cyclone III series FPGA as the main control chip. The changes in display quality of video signals of different bandwidths can be observed on the monitor by changing the bandwidths of the input video signals through low-pass filtering.

1. Introduction
The real-time video processing has been widely applied in many fields, such as video telephony, video conferences, monitoring, communications, and proved to be one of the research hotspots in the field of image processing. Field-Programmable Gate Arrays (FRGA) allows many image processing functions to be integrated on an FPGA chip, which has the distinct advantages of good flexible programmability, powerful parallel processing. In order to observe the display quality variation of different bandwidth video signals, this paper designs and implements a video image low-pass filter processing system based on FPGA. As reducing the bandwidths of the video signals only changes the horizontal resolutions of the video images, and the vertical resolutions of the video images are determined by the scanning parameters of the TV system and basically independent of the variation of the signal bandwidths, this paper only designs horizontal filters, while the design of 2-dimensional filters is not required.

2. Overall System Design
The hardware platform of this system is mainly consistent of the following parts: the FPGA model EP3C55F484 used for real-time low-pass filtering of video, A/D chip converting the incoming analog video signals into digital ones to input FPGA, and D/A chip converting the processed digital video signals for ADV7171 to analog video which is displayed on the monitor at last. Besides, the FPGA connects SRAM cells of two 16bit data lines, used to store video data. The system board is shown in Figure1.
2.1. Signal Acquisition

The camera is used to obtain optical images at the front end of the system and its outputs are PAL composite video signals, completing the transition from the optical signals to the analog electrical signals. The analog electrical signals are then converted to digital signals by the TVP5150 video decoding chip. The FPGA is used to configure for TVP5150 and the output format is selected as ITU-R BT.565, that is the YCrCb signals of the 4:2:2 sampling format with a clock frequency of 27 MHz.

2.2. Field Decoding

The complete PAL-system ITU-R BT656 data of one frame is divided into odd and even fields, the 23-311 rows are effective data of the even field, the 366-624 rows are effective data of the odd field, and the rest are the field control signals or invalid data. The first 288 bytes of each row are row control signals, with the first 4 bytes being EAV (Active Video End) signals and the last 4 bytes being SAV (Active Video Start) signals.

Both the EAV signals and the SAV signals have a 3-byte preamble ‘FF, 00, 00’, and the last 1 byte is ‘XY’. The parity flag is ‘F’, the vertical blanking flag is ‘V’, the horizontal blanking flag is ‘H’, and the protection bits are ‘P3, P2, P1, P0’. The extractions and detections of ‘F’, ‘V’ and ‘H’ are performed by continuously determining of ‘FF, 00, 00’ and ‘XY’. If the numerical value of ‘H’ is 0, the signal should be SAV signal, and then judge the numerical value of ‘F’, if numerical value of ‘F’ is determined as 0, the current field is an odd field, and another case is that numerical value of ‘F’ is 1, then it is an even field. When the numerical value of ‘F’ changes from 1 to 0, it is the frame header of a frame. The next step is judging the numerical value of ‘V’, if it can be judged as 0, it is the field forward. Since we only process the valid data, the row is proved to be the valid data, only when the frame header is detected and the numerical values of ‘H’ and ‘V’ are 0.

2.3. Frame Buffer

This paper uses two blocks of SRAM for ping-pong storage to realize the conversion between processing and display, with storing in units of one frame of data. When the current frame is written to SRAM1, the previous frame stored in SRAM2 is being read. The ping-pong switch is only performed to read the data of SRAM1, after the frame of SRAM1 is stored.

The valid data in one frame are 720 × 576 pixels, and the write address pointer is located at (0, 78) at the beginning. ‘FF, 00’ in the SAV of the valid data of the first row will be written on (0, 78). After the frame header is detected and the numerical values of ‘H’ and ‘V’ are judged as 0. The next ‘00’ and ‘XY’ will be written on (0, 79), before the subsequent first valid pixel will be written on (0, 80). During the storage, if the numerical value of ‘F’ of the SAV is found to change from 0 to 1, which means that the odd field ends and the incoming data will be the even field data, the write address pointer will jump to (1, 78), and the coming stage is the storage of even field. When the
numerical value of ‘F’ changes from 1 to 0, which is the new frame header, the write address pointer will reposition to (0, 78), then cycle back and forth.

3. Video Low-Pass Filtering in the FPGA

3.1. The Design of Low-Pass Filter

As the sensitivity of the human eyes to the changes of chromaticity is very low [1], this paper only performs low-pass filtering on the component of luminance signal Y. Since the clock frequency of the input signal is 27MHz and the input data is YCbCr format, presenting the arrangement of a luminance signal and a chrominance signal, the pixel clock is 13.5MHz which means that the sampling frequency of the Y signal is 13.5MHz. The bandwidths of input video signals of the SD signals are generally not more than 6MHz so that this paper chooses to limit the bandwidths of the input video signals to within 3MHz, and the cutoff frequency of the corresponding digital low-pass filter is

\[ f_c = \frac{3}{27} \times 2\pi = \frac{1}{9} \times 2\pi \]

Thus a low-pass filter can be designed as shown in Figure 2.

![Figure 2. Frequency Characteristic of the Filter](image)

3.2. The Low-Pass Filtering in the FPGA

This paper selects to design and implement a 12-order FIR filter, resulting to 13 coefficients. The formula implemented by the FIR filter is

\[ y(n) = \sum_{k=0}^{12} h(k)x(n - k) \]  \hspace{1cm} (1)

\( h(n) \) of the formula above is the coefficient of the filter and \( x(n) \) is the numerical value of the luminance Y in each row of pixels. \( y(n) \) is the component of the luminance signal Y of pixels after filtering.

Filtering with this formula causes the output images to produce rightward translations relative to the input video images. For avoiding the problem of shifting the filtered images, this paper caches the input pixels before filtering them with a non-causal filter.

\[ y(n) = \sum_{k=0}^{12} h(k)x(n - k + 6) \] \hspace{1cm} (2)

As the shown equation above, the primary filtering requires 13 multiplications. Since the designed filter is a linear phase filter, with the coefficients being symmetric, a linear phase structure can be employed to improve the efficiency of calculations. The linear phase structure is shown in Figure 3.
The corresponding calculation formula is

\[ y(n) = \sum_{k=0}^{5} h(k)[x(n - k + 6) + x(n - k - 6)] + h(6)x(n) \]  

(3)

The filtering requires only 7 multiplications at a time.

As the obtained coefficients of the FIR filter are all decimals, this paper shifts the coefficients of the filter to the left by 8 bits for amplification to realize the fractional multiplications. The result is shifted to the right by 8 bits to the original size, after the multiplications and accumulations of the filtering are completed. Thus this paper realizes the low-pass filtering processing of the component of the luminance signal Y.

4. Experimental Result

Through the steps above, the designs of each system module and the reduction processing of bandwidths of the camera acquisition video has been completed. The oscilloscope display of the video images before and after filtering is shown in Figure 4 and Figure 5. The input images are a multi-burst signals, including 0.5M, 1M, 2M, 4M, 4.8M and 5.8MHz signals.

![Figure 3. Linear Phase Structure](image)

After low-pass filtering with a cutoff frequency of 3MHz, signals above 3MHz that are the 4MHz, 4.8MHz and 5.8MHz signals have been filtered out, only leaving 0.5MHz, 1MHz and 2MHz signals shown in Figure 5.

![Figure 4. The Input of Multi-burst Signals](image)
5. Conclusion
This paper designed and implemented low-pass filtering processing of video image signals based on FPGA. As the cutoff frequency of the filter can be different, such as 1MHz, 2MHz, 3MHz, 4MHz etc., which can be modified by the FIR coefficients, it is convenient to observe the changes in the display quality of video signals of different bandwidths. The whole system of this paper was divided into a field decoding module, a frame buffer (ping-pong storage) module and a low-pass filtering module. The experimental results had shown the feasibility of the method.

References
[1] Xiuhua Jiang. 2018. Modern TV Principles [M]. Beijing: Higher Education Press.
[2] Jinming Wang, Shun Zhou. 2010. Digital System Design and VHDL [M]. Beijing: Publishing House of Electronic Industry.
[3] Hujun Zhang. 2011. Implementation of video acquisition and output system based on FPGA [D]. Dalian: Dalian Maritime University.
[4] Binjie Yu. 2008. Design of full color LED synchronous display control system based on FPGA [D]. Shanghai: Shanghai Jiao Tong University.
[5] Zhenyang Wu. 2016. Digital signal processing [M]. Beijing: Higher Education Press.