Design of a High Definition Video Communication System in Real-time Network

Liu Yunfeng
University of Chinese Academy of Sciences
Beijing, China
Institute of Optics and Electronics, CAS
Chengdu, China
e-mail: liuyun_ww@163.com

Peng Xianrong, Jin Zheng
Institute of Optics and Electronics
Chinese Academy of Sciences
Chengdu, China
e-mail: jinzhang27@126.com

***Abstract*—This paper presents the design of a high definition video communication system with real-time performance in the network. In order to process and transmit the high definition video, the design combines FPGA and SOC, follows the H.264 video coding compression. The FPGA realizes the high–speed video acquisition via Cameralink interface, and convert the raw image to ITU-R BT.1120 stream. The SOC contains CPU and Codec, it compresses the BT.1120 stream to H.264 steam and transmit the stream in the network. The low-delay rate control algorithm can limit the bitrate in a very low level. The result shows that the system reduces bandwidth and lowers the latency, the design is adaptable to the real time environments.**

Keywords-FPGA; H.264; ITU-R BT.1120; Low latency;

I. INTRODUCTION

In recent years, there is an urgent demand of the high resolution image processing in the aerospace, opto-electronic detection and other fields [1], such as Full HD (High Definition) video. The HD video (1920 pixels x 1080 pixels 30 frame per second) has huge data, and is difficult to be processed on the traditional image processing platform. So the video compression is needed. H.264 as the new generation video coding standard, has greatly improved the image quality and bitrate control. This paper uses the FPGA and SOC (System On Chip) with hardware and software combination, designs and implements a network video communication system on the H.264 coding algorithm.

II. SYSTEM HARDWARE

The system contains the CCD camera, FPGA, SOC and some peripheral hardware. The camera transmits the images by the CameraLink interface. The FPGA is Spartan-6 [2], it processes the digital image, converts the image to ITU-R BT.1120 video stream [3], and then, transmits the stream to the SOC. The SOC is MG3500 [4], it contains a ARM9 CPU and a H.264 codec, the codec encodes the video to H.264 stream in real time, the CPU transmit the stream in the network at last. Fig.1 shows the system architecture.

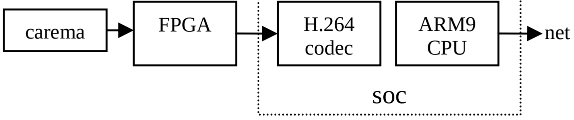

Figure 1. The hareware framework

III. FPGA DESIGN

A. Video acquistion

High definition digital video is transmitted from camera to FPGA via CameraLink interface, and then buffered in the DDR2 SDRAM. Spartan-6 has the specialized MCB core for the DDR access, DDR2 SDRAM can read and write data conveniently. In order to balance the speed of the image data input and output, one frame must be buffered at least; the ping-pong operation is also used to ensure the image integrity. So the system design use DDR2/800MHz as the frame buffer for the high speed image data access, and open two frame buffers for ping-pong operation.

The CCD pixels are preceded in the optical path by a color filter array (CFA) in a Bayer mosaic pattern, in order to get real color images, the raw image should be processed in the pipeline including demosaicing [5], auto white balance and auto gain control [6].

After that, the Bayer pattern image is converted to RGB image.

B. BT.1120 conversion

After all the image processing above, the digital image should be converted to BT.1120 stream.

The BT.1120 is the recommendation of digital interfaces for HDTV studio signals. It complies with the characteristics described in Recommendation ITU-R BT.709 [7]. BT.709 contains some HDTV studio standards to cover a wide range of applications including Common Image Format (CIF) system which has 1125 total lines and 1080 active lines. The standards contain opto-electronic conversion, picture characteristics, picture scanning characteristics, signal format and analogue representation. BT.1120 also follows these standards, and then contains more such as bit-serial data format and transmission format.

BT.1120 video interface supports 10-bit and 8-bit video data transmission, the video data format can be RGB 4:4:4 or YCrCb 4:2:2. First, the color space should be converted from RGB to YCbCr. The raw digital RGB color has 3 components including Red, Green and Blue. Each component has a value between 0 and 255, corresponding to 8-bit quantization. In accordance with BT.1120 recommendation, the value should be normalized as in (1)

Published by Atlantis Press, Paris, France.

$$E_R = R/255, E_G = G/255, E_R = R/255 \tag{1}$$

YCbCr is not an absolute color space; rather, it's a way of encoding RGB information. It contains the luminance component(Y), blue-difference component (B-Y) and red-difference component (R-Y). In accordance with BT.1120 recommendation, the YCbCr is derived from RGB according to the following transform:

$$E_Y = 0.2126E_{\mathrm{R}} + 0.7152E_G + 0.0722E_B \tag{2}$$

$$E_{CB} = (E_B - E_Y)\frac{0.5}{0.9278} = \frac{-0.2126\,E_R - 0.7152\,E_G + 0.9278\,E_B}{1.8556} \tag{3}$$

$$E_{CR} = (E_R - E_Y)\frac{0.5}{0.7874} = \frac{0.7874\,E_R - 0.7152\,E_G - 0.0722\,E_B}{1.5748} \tag{4}$$

After the transform, the signal should be quantized. In BT.1120 recommendation, as 8-bit quantization, the Y component has 220 steps, the value 0 is mapped to 16; the Cr and Cb components both have 255 steps and 0 is mapped to the median, 128. So the quantized signal is derived as follows:

$$D_Y = \mathrm{int}(219E_Y + 16.5) \tag{5}$$

$$D_{CB} = \mathrm{int}(224E_{CB} + 128.5) \tag{6}$$

$$D_{CR} = \mathrm{int}(224E_{CR} + 128.5) \tag{7}$$

D_Y, D_{CB}, D_{CR} respectively represents the quantized digital YCbCr signal. For FPGA to achieve, fixed-point decimal arithmetic can be multiplied by a factor into integer arithmetic. All the values in (5)~(7) are multiplied by 1024, the new transform is (8)~(10):

$$1024D_Y = 187R + 629G + 63B + 16896 \tag{8}$$

$$1024D_{CB} = \text{-}103R \text{-} 347G + 450B + 131584 \tag{9}$$

$$1024D_{CR} = 450R - 409G - 41B + 131584 \tag{10}$$

After the color space converted, the signal should be sampled. ITU-R BT.1120 describes the YCbCr signals in 4:2:2 sampling rate. This system uses the 8-bit color space, so that each pixel has 16 bit data composed of a chrominance(C) component and a luminance(Y) component. In accordance with the bit-parallel interface in BT.1120, Cb and Y are transmitted in odd numbers of pixel clock; Cr and Y are transmitted in even numbers. The sequence is represented as:

(Cb_1 Y_1) (Cr_1 Y_2) (Cb_3 Y_3) (Cr_3 Y_4)…

The FPGA can convert the image from RGB to YCbCr in parallel processing rapidly, the conversion can be completed in two timing cycles while nine DSP multipliers are used. The data path is shown as Fig. 2.

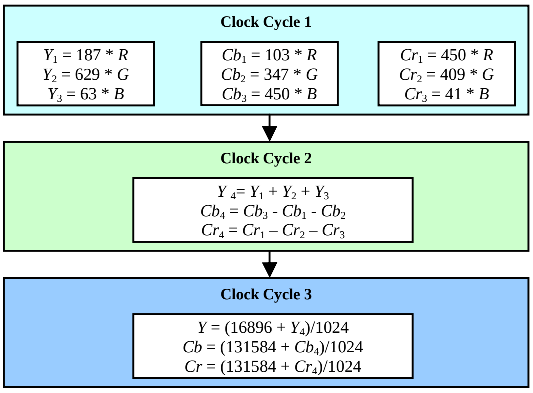

Figure 2. YCbCr conversion.

ITU-R BT.1120 digital interface contains only the video data signal without a separate control signal. it use the timing reference codes to identify the line and frame. There are two timing reference codes embedded into video data stream, one at the beginning of each video data block(start of active video, SAV) and the other at the end of each video data block(end of active video, EAV). These codes are contiguous with the video data, and continue during the field/frame blanking interval.

Each code consists of a four-word sequence. In the 8-bit implementation, the bit assignment of the word is given in Table I. The first three words are fixed preamble and the fourth word carries the information that defines field identification (F), frame blanking period(V), and line blanking period (H).

TABLE I. BIT ASSIGNMENT FOR VIDEO TIMING REFERNCE CODES

Word	Bit number							
	7	6	5	4	3	2	1	0
First	1	1	1	1	1	1	1	1
Second	0	0	0	0	0	0	0	0
Third	0	0	0	0	0	0	0	0
fourth	1	F	V	H	P3	P2	P1	P0

The bits F and V change state synchronously with EAV at the beginning of the digital line. The value of protection bits, P0 to P3, depends on the F, V and H as shown in Table 15. The arrangement permits one-bit errors to be corrected and two-bit errors to be detected at the receiver, as shown in Table II.

TABLE II. PROTECTION BITS FOR SAV/EAV

SAV/EAV bit status			Protection bits			
F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

In order to implement the timing reference codes on FPGA, The Table II should be stored in the ROM with 4-bit width and 8-bit depth. The bits F,V and H are mapped to the ROM address, the protection bits are stored in the corresponding address. When encoding the timing reference codes, the bits are read from the ROM address and then combined with it.

IV. LOW LATENCY RATE CONTRL

The SOC can encode the BT.1120 stream to H.264 data. Under the prerequisite of image quality, this paper designs a low-delay rate control algorithm to reduce the transmission bandwidth and lower the delay.

The rate control algorithm uses data on the size of past frames to estimate the size of future frames. It uses its algorithm to decide if the size of future frames must be changed in order to meet performance requirements such as bitrate and latency.

Rate control algorithms are usually classified as VBR (Variable Bit Rate) and CBR (Constants Bit Rate). CBR generally means maintaining the specified bitrate over time. VBR generally means that the rate control is allowed to generate a bitrate much lower than specified when the image is easy to encode. This is intended to prevent wasting bits, if a lower bitrate can achieve a high quality. This paper uses a VBR algorithm.

Low latency encoding is the most difficult case for the rate control, since it is typically used for network streaming. Network streaming usually requires a low transfer rate due to limited bandwidth, so this produces the rate control’s worst case scenario of low latency plus low transfer rate. Therefore, this section refers to "low latency" encoding but actually means low latency plus low transfer rate.

A. Rate comtrol

On MG3500, the algorithm set the slice QP (quantization parameters) value in order to control the long term bitrate. In AVC encoding, the QP value controls the quantization of all coefficients in the frame [8]. More quantization (a higher QP value) means fewer bits per frame, and less quantization (a lower QP value) means more bits per frame. The QP values range from 0 to 51. For the VBR application, the minimum QP value is set to 20.

The rate control use separate parameters for the bitrate and for the transmission rate. The bitrate is an approximate target for the size of the bitstream. The transfer rate is an absolute upper limit on the size of the bitstream. The transfer rate controls the fluctuations of the bitrate.

If the transfer rate is set much higher than the bitrate, then the bitrate may rise very high for short periods. If the transfer rate is set close to the bitrate, it places a limit on how high the bitrate can increase within a short period.

The transfer rate represents the rate at which data can be transferred to the decoder. In the case of network streaming, the transfer rate should be set to approximately the maximum bandwidth available. The transfer rate should not be set low unless this is required. A lower transfer rate limits the rate control when it is choosing the short term bitrate. This will generally result in a long term bitrate, because the rate control will tend to produce smaller frames in order to avoid exceeding the transfer rate. The risk of frame drops is also increased.

Recommended transfer rates are 1.5 x bitrate for low latency applications.

B. Low latency

The rate control includes explicit setting parameters for the target latency. Technically, this is the amount of data the decoder is expected to buffer before starting playback. In reality, it affects the overall latency between the encoder and the decoder. The rate control’s latency setting does not control the overall latency, which depends on encoder and decoder settings and the network delay. However, the rate control must be set to support this latency or it will occasionally produce frames which are too large, and cannot be transmitted to the decoder when it requires them. The latency setting can only be understood in combination with the transfer rate setting above. The latency controls the number of frames which can be buffered up between the encoder and the decoder, and the transfer rate controls how fast those frames are actually transmit. The two of these together control the size of the frames.

If the latency is low but the transfer rate is very high, only a few frames can be buffered but their size does not matter. Even if they are very large, they can be transmitted before a new frame is encoded and there will always be a small number of buffered frames. But if the transfer rate is low but the latency is very high, many frames can be buffered. A few large frames will be averaged with many smaller frames, and they will not exceed the buffering limit.

Only if both latency and transfer rate are low does the rate control’s behavior change substantially. As latency and transfer rate are reduced, the actual bitrate will fall further below the specified bitrate. The rate control cannot use the full bitrate, or it will risk generating a frame which is too large to transmit to the decoder on time. So the recommended latency for the network is 4 frames(133ms).

the H.264 coding chose *high profile* and *level 4.1* feature, using the IP GOP (group of Pictures) structure [8] with only I-frame and P-frame ,the B-frame is not used to lower the decoder’s latency.

V. Network Tramsmission

As a video network communication system, network transmission is also very important. After encoding, the H.264 stream is transmitted in the network over TCP/IP.

Video communication has high real-time requirements. In order to ensure that the transmission of the image will not obviously delay the system in the unknown network, UDP protocol is used in the transport layer for network communications.

Considering the complexity of the data packets out of order in the transmission network, this will bring some error in real-time decoding to affect the image quality, the system has completed the RTP (Real-time Transport Protocol) protocol [9]. RTP is an application layer protocol, which also use UDP in the transport layer, not a connection-oriented TCP.

In the data packets to be transported, there is a RTP header that contains some important information such as the data sequence numbers and timestamps, Timestamp is the description of the packet time synchronization information, it helps the critical data be restored to the correct chronological order, which requires that the sender timestamps increase continuously and monotonically. So that at the receiving end, as long as a certain data cached, the video data can be sorted and restored in accordance with the normal sequence.

According to the size of the receive buffer, using the RTP transmission will bring a little certain delay, but a certain improvement in image quality at the same time, which doesn't influence the real-time.

RTP and UDP are point-to-point transmission, in order to meet some one-to-many transmission requirements, the system also implements RTSP (the Real Time Streaming Protocol) protocol stack with the Client (decoder) / Server (encoder) model, it's a text-based protocol for client and server to establish and negotiate the real-time streaming.

RTSP is also an application layer protocol, which is located above RTP. This protocol doesn't transmit the data, it only control the states of the stream. The system still uses the RTP/UDP to transmit the stream at the transport layer.

VI. Experiments

The design described in this paper is successfully used in a opto-electronic system, Fig.3 shows the bitrate curve in the corresponding experiments.

We set the video bitrate of 6000kb/s, the long term rate is stable. In the static scene, the bitrate is about 5000kb/s, especially in some low complexity scene, the rate is below 4000kb/s; in the motion scene in a short while, the rate increases to 6000kb/s.

The whole latency is about 300ms, when the video displays in the remote decoder.

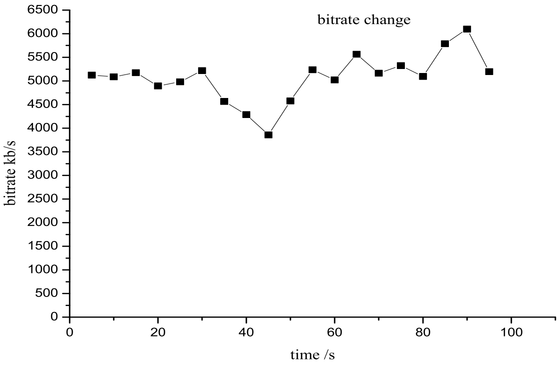

Figure 3. The Bitrate changes.

VII. Conclusion

The paper designs a real-time full HD video communication system which combined FPGA and SOC, and then introduces in detail the design of each module including BT.1120 Stream conversion, low latency rate control and network transmission. The result shows that the system has the following advantages. One is high real time, the H.264 codec can encode the video at high speed with the low latency rate control algorithm. The other one, the FPGA have rich resources inside, and can achieve the required functions on different demands.

References

[1] W. Chen, P. Chen, W. Lee and C. Huang, “Design and Implementation of a Real Time Video Surveillance System with Wireless Sensor Networks,” IEEE Vehicular Technology Conference, May 2008, pp. 218-222,.

[2] Xilinx Corporation. “Spartan-6 FPGA Memory Controller User Guide,” Aug. 2009.

[3] ITU-R. “Digital interfaces for HDTV studio signals, Recommendation ITU-R BT.1120-5,” 2004.

[4] Liu Yunfeng, Guo Xiaoli, Peng Xianrong. “A H.264 Encoding/Decoding System Design based on ASIC”, Technical Acoustics, vol.30, No.4, Aug. 2011, pp. 255-258

[5] GUO Jian-ya and XU Zhi-yong, “A Bayer CFA demosaicing method suitable for real-time hardware implementation,” Electronic Instrumentation Customer, vol.18, No.5, May. 2011, pp. 67-70

[6] D. Nikitenko, M. Wirth, “Applicability of White-Balancing Algorithms to Restoring Faded Colour Slides: An Empirical Evaluation,” Journal of Multimedia, Vol. 3, No. 5, Dec. 2008, pp. 9-18.

[7] ITU-R, “Parameter values for the HDTV standards for production and international programme exchange. Recommendation ITU-R BT.709-5,” Apr. 2002

[8] Said Benierbah and Mohammed Khamadja, "A New Technique for Quality Scalable Video Coding With H.264," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No.11, Nov. 2005, pp. 1332-1340.

[9] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund and D. Singer, "RTP Payload Format for H.264 Video," IETF , RFC3984, Feb. 2005.