Design and Implementation of HD Video Encoding System Based on HDMI

Wanjia Gao[1], Baohai Sun[2], Qingqing Peng[3], Huixin Zhang[1*]

[1]National Defense Key Laboratory for Electronic Measurement Technology, North University of China, Taiyuan, China;
[2]Tianjin 712 Communication and Broadcasting Group Co. LTD (TCB), Tianjin, China;
[3]North Automatic Control Technology Insititute,Taiyuan, China

Abstract. Focus on the problem that TI's dedicated video processor TMS320DM8148 cannot directly collect video data from HDMI interface, this paper presents a video coding system based on H.264.The system used DSP+FPGA architecture, FPGA is responsible for collecting video data of HDMI interface and caching. And then send to DM8148 through GPMC, DM8148 completes video encoding and decoding through the interaction of internal modules. The results show that video is displayed clearly and smoothly, without distortion or error. It successfully realizes the collection of the video data of HDMI interface obtained with DM8148.

1 Introduction

In recent years, digital image technology has been widely used in security, medical treatment, education, and monitoring, which has led to a great development and attention. With the continuous improvement of video quality, the storage speed and the bandwidth growth speed cannot meet its requirements. So the video encoding technology is becoming increasingly important. With the continuous popularization and application of advanced video compression coding standards, such as H.264, HD video is rapidly entering into people's daily life. HDMI HD video interface not only has the features of higher bandwidth and higher resolution, but also integrates video transmission and audio transmission, which greatly simplifying the cable connection settings.

The DSP of TMS320DM8148 of TI is a special chip with rich video image processing module and low power consumption. It can achieve the processing of 1 channel 1080P/60fps or 4 channels 720P/60fps, and the speed is fast. However, the video data information transmitted by HDMI HD video interface cannot be directly collected.

In view of this, this paper designs and implements a high definition video coding system based on HDMI interface.

2 Overall system design

The overall system adopts FPGA+DSP architecture, and the overall scheme design structure is shown in Fig 1. FPGA sends control signals and clock synchronization signals to the forward HDMI interface to control the beginning and end of video information collection. The received high definition video information is cached in the internal FIFO of FPGA, then be sent to the FIFO of GPMC though the data logic control unit. Then the information is sent to DSP by GPMC data bus. The DSP of the system is the TMS320DM8148 of TI scheme. After receiving video information, the DSP will encode and decode video according to the program control process. Finally, the video data is output by the HD interface, which is displayed by the special display.

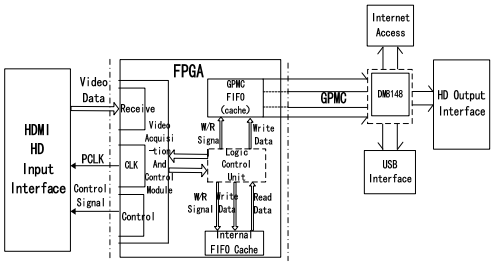


Fig. 1. General block diagram of the system structure.

3 HD interface design

3.1. High definition interface hardware design

High definition interface design includes video data input interface and video data output interface. The output interface is interlinked with DSP, and the input interface can be interlinked with FPGA or DSP. Because the DSP integrates the high definition control module, it is easier to handle the data collection and interface controlling than the FPGA. However, the internal integration of DSP is high, and when the utilization rate is high, it will have a certain effect on the internal running speed. Therefore, when the internal resource utilization rate of DSP is high, FPGA can be used to

* Corresponding author: author@e-mail.org

collect data and reduce the DSP burden. When the usage rate is low, it can be collected directly by DSP.

3.1.1 High definition input interface design

The system uses ADV7619 and IP4776CZ38 as the video decoding and receiving chip. ADV7619 is a high quality, double-input, single-output(2:1) multiplex HDMI receiver with a maximum clock frequency of 170MHz. And it supports all mandatory 3D extension formats that specified in the (HDMI®) 1.4a specification, named sYCC601, Adobe RGB, Adobe YCC 601, xvYCC color range. IP4776CZ38 can provide high quality ESD protection for TMDS signals. It can also perform a level conversion to protect HDMI's DDC channel. It is used to prevent the system from collapsing or sending errors due to excessive static electricity. Fig 2 shows the high definition interface.

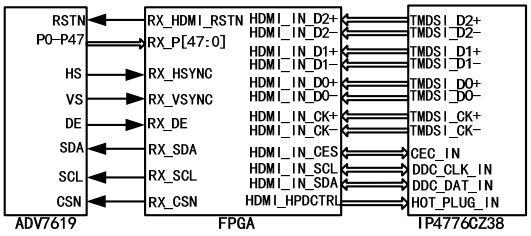


Fig. 2. High definition interface schematic.

3.1.2 High definition output interface design

The HD output interface is TPD12S016. It is a multimedia interface conversion chip that integrates low capacitance direction protection diode, and it contains a conversion buffer that automatically induces the I2C level. TPD12S016 supplies the HDMI power line with the 55mA limited stream 5V. The CT_PHD pin manages the 5V output and hot plug control, which is independent of the LS_OE pin, enabling the activation detection to run between the start of the HD link SDA, SCL, and CEC lines are pulled to VCCA on side A. The I2C control signal SCL and SDA can drive load capacitance higher than 1.4 version of HDMI specification, which can reach 750pF. When the HDMI mechanical interface is inserted into the socket, it may cause error detection. However, the PHD_B port integrated burr pulse filter can solve this problem. The reverse current protection function is integrated internally to prevent the reverse current from burning the chip during a power outage. Features include: Compatible with HDMI test without other external devices; Hold 1.4 version HDMI data transmission rate; Four differential pairs for 8-channel transmission channels are used to prevent electrostatic discharge by using low differential capacitors; 5V output pin integrated chip load switch, current is limited to 50mA; Long HDMI cables use single trigger circuit driver, in line with the HDMI specification on the integration of pull-up resistors and drop-down resistors, also provide reverse drive protection in mechanical interface, this is the direction of automatic induction I2C level converter. Fig 3 shows the high definition output schematic diagram.

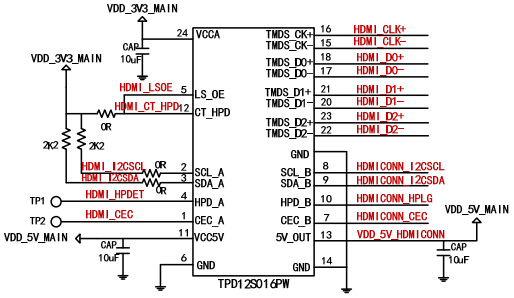
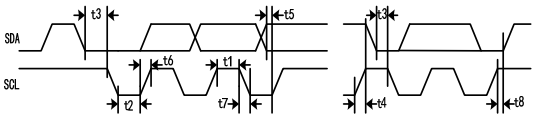


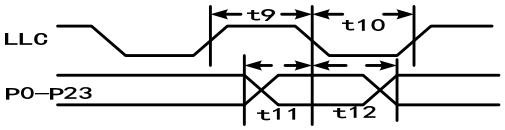
Fig. 3. High definition output interface schematic diagram.

3.2. High definition interface collection timing

The data collection of HD interface is mainly control of ADV7619. Fig 4 is the ADV7619 sequence diagram. And (a) is I2C sequence diagram, it includes eight parts: t1-t8. (b) is pixel port and SDR output control sequence diagram. It consists of two parts: clock output and data, control output. The details are shown in Table 1



（a）I2C Port timing.



（b）Pixel port and SDR control output timing..

Fig. 4. Sequence diagram.

Table 1. Details of sequence diagram.

Parameter		Time
I2C Port	SCL Min Pulse Width High	t1
	SCL Min Pulse Width Low	t2
	Start Condition Holding Time	t3
	Start Condition Setting Time	t4
	SDA Building Time	t5
	SCL and SDA Rising Time	t6
	SCL and SDA Falling Time	t7
	Stop Condition Building Time	t8
CLK Output	LLC Space Ratio	t9; t10
Data and Control Output	Data Output Conversion Time.	t11
		t12

In Table 1, t11 is the time of effective data end to negative clock edge, t12 is negative clock edge to effective data begin time.

Fig 5 is the I2C control sequence diagram, which controls the beginning and end of data exchange. Combining the data reading and writing sequence of Fig 6, it can be seen that the state of the stop bit is always changing. It is needless for stop bit when writing data, but is needed when reading.

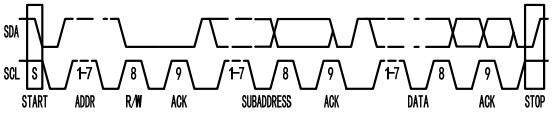


Fig. 5. I2C control timing

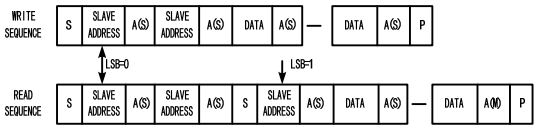


Fig. 6. Data read-write sequence

4 Software coding and decoding

4.1. Software coding design

FPGA collects video information transmitted by HD interface, and caches the information in the internal FIFO. The cached data is transmitted to GPMC FIFO through logical control unit. Then send to DM8148 by GPMC data bus. After receiving video information, the coding and decoding of video is followed by the software process. Finally, it is output by the HD interface and displayed by the display. The interaction between the internal threads involved in the video encoding process is shown in Fig 7.

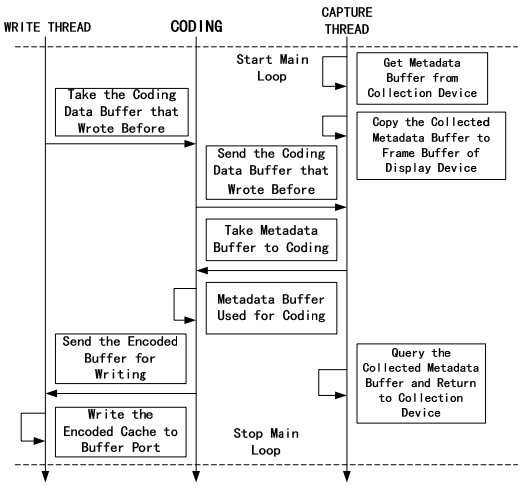


Fig. 7. Video interactive thread.

First, the capture thread takes the video data from the data collection buffer. At the same time, the encoding thread takes out the I/O buffer that has been written to the file from the write thread. Then giving the encoded data buffer to the capture thread to populate the new video frame data. To encode the video frame that is being processed, the encoding thread needs to obtain video source data from the capture thread. At this point, the encoding thread takes the source data cache from the capture thread through FIFO_get () for encoding. After that, the encoding thread calls the internal encoding function on the DSP, encodes the received source video, and stores it in the I/O cache. Setting the collection data cache as read-only to avoid modifying the source data in the encoding process. When the video encoder running on the DSP will code the data in the cache to the I/O cache, the latter will use the FIFO_put () function to send to the write thread. In this way, it ensures that the three threads interact with each other, successfully completing data collection, encoding data, and caching data.

4.2. Video decoding design

The encoded video information cannot be displayed normally without decoding. So when the encoded video data file loading thread is correctly loaded into the buffer, the video file needs to be decoded to ensure its normal display. The thread is mainly divided into four aspects. The process is shown in Fig 8.

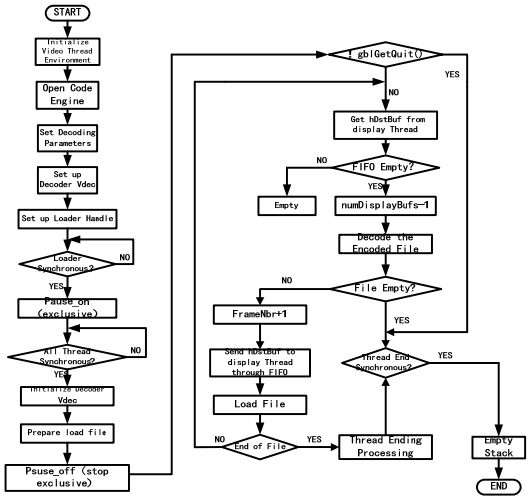


Fig. 8. Decoding process

First, the video thread preparation work. Initialize the video thread environment, and establish the interface API between the decoding thread and the underlying algorithm interface SPI and the high-level application, so that the underlying codec algorithm is transparent to the decoding thread.

Second, run the interrupt service function to store the image data in the appropriate file. When all threads are in sync, initialize the decoder Vdec, then start the interrupt service function to read the image data and store it in the file, using the Pause_off () function to terminate the system resource exclusive.

Third, the main loop, decoding video files. Check if the pipe is empty, and if not, perform the appropriate pipe cleanup operation. Using the Vdec_process () function to perform the decoding operation of the file. If the file is empty, exit the main loop and wait for the end of the other threads. Instead, continue and add 1 to the decoded data frame until all the files have been decoded.

Fourth, finish the document. When all the files have been decoded, to ensure the continuity of the image, the last frame of the image is always displayed on the screen

until a new decoding task arrives or the main thread ends synchronously, emptying the corresponding cache stack space content.

5 System testing and analysis

The HD video encoding system sends the decoded video files to the display thread for final processing, which can be displayed on the displayer. Through the real-time online debugging function of CCS, the video data on the host with the resolution of 1920*1088 and the YUV format is transmitted to DM8148 by using the network port. After coding, the H.264 format video data will be sent back to the upper computer through the network interface. And the coding effect and encoding multiple of DM8148 can be verified. In Fig 9, (a) shows that program initialization, starts to read the configuration file, completes the reading configuration file, and starts to frame the process. The encoding of each frame will be shown, and frames it while passing it back. (b) shows the frame will end when all frames are encoded.

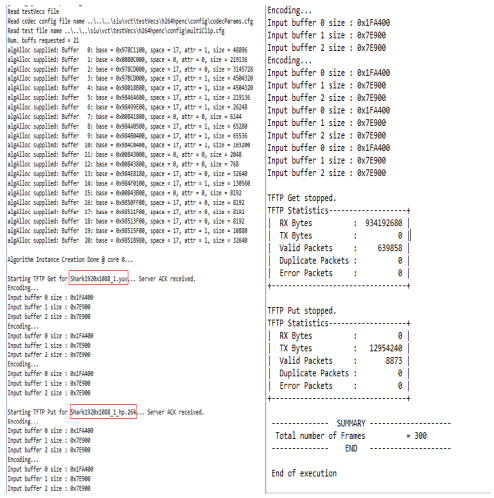


Fig. 9. (a) Program coding beginning (b) Program coding end

Before coding, video information is YUV format video, and video information after coding is hp.264 format video. Compared to video before and after the coding, the coding multiple is about 79 times. The comparison of video screenshots before and after the coding is shown in Fig 10

Fig. 10. Comparison of video screenshots before and after encoding

By comparing to two videos before and after coding, we can find that there is little difference in sharpness, and there is no distortion or disorder in the recognition of human eyes. It indicates that the system works normally and meets the overall design index of the system.

6 Conclusion

In this paper, a high definition video coding system based on HDMI is successfully designed. The system can process the high definition video with a resolution of 1920*1088. And it adopts DSP+FPGA framework. FPGA realized the function of collecting video information, caching data and communicating with DM8148.DSP receives video information from FPGA, encodes and decodes it, and finally uploads to the upper computer, and displays the decoded video information through the displayer. The system uses the coding system of TI scheme. It encodes and decodes the source video information collected by the hardware interface. It successfully solves the video data problem that TMS320DM8148 cannot directly collect the video data from HDMI interface.

Acknowledgement

This paper is supported by the National Natural Science Foundation for Distinguished Young Scholars (NO. 51425505).

References

1. Key Digital Systems; Boscov's Department Store achieves success with 100 HD Displays (2011, 4) HDMI Sources, and custom HDMI Distribution Solutions from Key Digital. In Leisure and Travel Week..
2. Chao Zhang; Li Jun Yun; Jun Sheng Shi; Kun Wang; Zhi Hui Deng (2013), Design and Implementation of the HDMI HD Image Acquisition System Based on FPGA. In Advanced Materials Research.
3. Jiang Xiong; Qing Ming Yi; Min Shi; Ting Ting Li (2013),HDMI Data Transmission System Design. In Applied Mechanics and Materials.
4. J Xiong; Q Yi; M Shi; T Li. Research and Design of Data Transmission System Based on HDMI. (ICCSEE-13 ,2013).
5. S Cho; ST Chung; N Binh; VQ Do (2014), Design and implementation of an embedded multimedia live streaming decoder system. In International Conference on Advanced Technologi.
6. Ming Kui Zheng; Kai Xiong Su; Xiu Zhi Yang (2011), A Scheme of H.264 HD Encoding/Transcoding System Based on ASIC. In Applied Mechanics and Materials.