

Design and Implementation of Real-time Video Processing and Transferring System Based on TMS320C6678

CHEN Hong-zhou⁺, XU Tong-lei, CHEN Dong-cheng

Jiangsu Automation Research Institute, Lianyungang, Jiangsu, 222006, China

Abstract: In order to resolve the problem that processing and transferring video in real time, using the TI TMS320C6678 high-performance multi-core DSP and its high-speed SRIO and PCIe interfaces, a real-time video processing and transferring system is designed and realized. Firstly, the hardware design of a video processing board is designed. Introduce the video capturing, pre-processing and how to process video in real time. And then introduce the data processing, data transferring via SRIO between chips and data transferring between board and host computer via PCIe. Especially introduce the realizing step and key points. Lastly, experiments using the designed board are implemented. The result shows that the transfer delay between FPGA and DSP is less than 240ns, which can meet the real-time quality of video processing system. The designed board can process and transfer video in real time and robustly.

Keywords: TMS320C6678; PCIe; SRIO; real-time processing; data transfer

1. Introduction

As digital signal processor (DSP) has the features of controllability, fast calculation, and various interfaces and so on, signal processing boards based on DSP have been applied in more and more fields such as communication, computer, image processing etc. Recently, along with the continuous improvement of performance, the processor's processing capability is getting stronger and stronger. And also the interfaces of DSP are developed from simple interfaces to complex interfaces, from low-speed parallel interfaces to high-speed serial interfaces, so data transfer speed has been greatly improved. The improvement of interfaces provides support for application of the high performance multi-core DSP[1].

TMS320C6678 has various data communication interfaces, such as Peripheral Component Interconnect Express (PCIe), Serial RapidIO (SRIO), HyperLink etc. Those interfaces can satisfy the requirement of large data flow in communication and video processing field. The SRIO bus is an open standard, which has high-performance, good stability, high speed, low cost, and less pin counts. It provides a data transfer solution which is high bandwidth, low delay, and high reliability. SRIO supports point-to-point or one point to multiple points communication mode, and also support DMA, message passing and multiple topological structures. It is a packet-switched interconnect intended primarily as an intrasystem interface for chip-to-chip and board-to-board communications at gigabyte-per-second performance levels. PCIe bus is the third general serial I/O connection, which has the features like low-pin-count, high-reliability, and high-speed data transfer. It has been widely used in computer, mobile equipment, and embedded system[2].

For the C6678 DSP which has high performance, the implementation of its peripheral interface is also the embodiment of system performance, which may become the bottleneck of performance improvement. According to this requirement, a real-time video processing and transferring system has been designed in this paper. Based on TI's multi-core C6678 DSP, making full use of the high-speed communication interfaces like SRIO and PCIe, video data capturing, real-time video processing and transmission are realized. The system transfers video data using SRIO from FPGA to DSP in real time. Through PCIe bus, a large amount

⁺ Corresponding author: CHEN Hong-zhou Tel.: +86-18036678734
E-mail address: cchzz2001@163.com

of video data is transmitted to the host computer in real time and then displayed and stored in real time. In this paper, the overall architecture of the video processing and transferring system is firstly introduced, and then the in-depth study of data transmission is focused on. After that, the software design and implementation of the system are introduced in detail. Finally, the performance of the designed system is verified by the actual video capturing and transmission.

2. Overall Design of System

To realize the function of video capture, processing, transfer and display, a real-time video processing and transferring system based on TMS320C6678 is designed. The block diagram of the system is shown in Figure 1. The system consists of a video processing board and a commercial computer. The video processing board is inserted into the PCIe expansion slot of the computer, and unified power supply is provided. When the computer is powered on, the DSP of the video processing board loads the program from the SPI flash, and then starts running. The video signal is captured by decoder and send to FPGA to preprocess, and the results of preprocessing are transferred to DSP which realizes video processing algorithm via SRIO. Then the video data and the processed data are transferred to the computer in real time via PCIe interface. The computer displays and stores the data at last, and also it can interact with DSP through PCIe bus[3][4].

Video processing board is the key instrument of the system. The board has the core chip which is TMS320C6678. This chip has eight processing cores, and each processor can work at 1.25GHz. The SRIO of the chip is connected to SRIO of FPGA, and can receive video data from FPGA. The GPIO of the chip receives interrupt signal generated by FPGA. The chip realizes the real-time transfer with host computer using PCIe interface[5]. The 512MB externally mounted DDR3 SDRAM is used as data cache unit. The program of DSP is loaded from SPI flash. And the video decoder chip TVP5150 is controlled by DSP via IIC bus. The FPGA of the board is XC5VLX110T Virtex-5 of Xilinx Company. It is used to capture video data which output by the video decoder chip, and packet the data according to the SRIO protocol and then send to DSP. After a row data is sent, the FPGA generates an interrupt to DSP's GPIO. Then the DSP can start to process the received video data between video blanking period[6] and transmit the video data and processed results to the host computer to display and store.

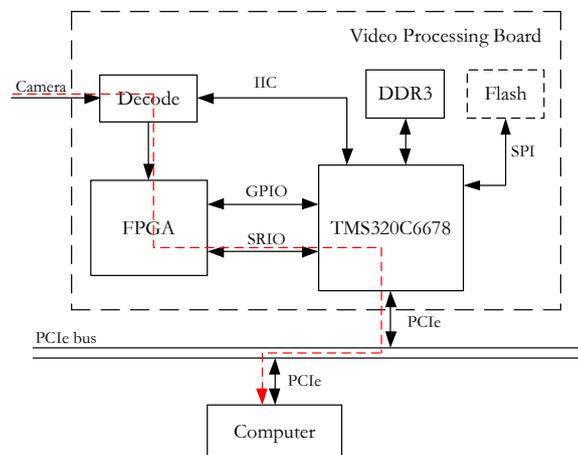


Fig. 1: Block diagram of system

3. Designs of High Speed Communication Interfaces

To insure communication delay is as low as possible, high speed communication interfaces such as SRIO and PCIE of TMS320C6678 DSP are utilized to realize the real-time processing in the system.

3.1. Realization of SRIO Based on FPGA

Video processing system based on DSP and FPGA usually reads data from RAM of FPGA via DSP's EMIF. This method of data transfer at least has one frame delay. As the master device, the C6678 DSP reads data from FPGA via EMIF which is an asynchronous interface, and the delay is more than the time of capturing one frame. So, this method will affect the performance of systems requiring high real-time performance.

In our system, the FPGA is the master device. It directly writes data to memory of DSP using SRIO interface. When one row of data is transferred, the FPGA generates an interrupt to DSP, and then DSP reads data immediately and processes the data in parallel. The delay of data transfer is the time of capturing one row. This can satisfy the requirement of real-time transferring.

SRIO running in X4 mode is used to transfer data between FPGA and C6678 DSP[7]. The transmission rate of each channel can reach 2.5 Gbps, and transmission rate of four channels can reach 10 Gbps. Except the 8B/10B coding and packing overhead during transmission, the data transmission efficiency can reach 75% that is 7.5 Gbps. Video data is transferred row by row to realize strong real-time transmission technology. As SRIO is a packet-switched interconnect, in our system the SRIO is initiated for a SWRITE RapidIO transaction. The input video of the system is standard PAL video, and the resolution is 720*576. The data is transferred row by row, and one row data transaction will span three packets, because SRIO specifies data packets with payloads up to 256 bytes. Each packet has 240 bytes data. Once a row data was put to the buffer of FPGA, SRIO can read data from the buffer and send to DSP at the same time. The experiment result shows that the delay of one row data transmission is less than 240ns, which can meet the real-time quality of video processing system. Figure 2 shows the sequence chart of FPGA sending data to DSP via SRIO. The real data which is captured by Chipscope when sending a row data is shown in the figure. As you can see from the figure, a row of data is sent in three SRIO packets.



Fig. 2: Sequence chart of FPGA sending data to DSP via SRIO

3.2. Design of PCIe Interface

TMS320C6678 DSP integrates multiple external interfaces, which can meet various needs of users. PCIe, as one of the high-speed communication interfaces, provides a high-speed transmission channel between C6678 and external devices, and also provides an interconnection interface between DSP and other PCIe devices. The PCIe module of C6678 has the advantages of less pin, high reliability and maximum raw speed of 5.0 Gbps per lane per direction. It supports dual operation modes: Root Complex(RC) and End Point(EP), and supports single channel or two channels. It is compatible with the first and the second generations of PCIe standard^[8]. In our system, we use a PCIe bidirectional link interface with one lane. The video processing board works in EP mode, and the host computer works in RC mode.

The PCIe interface of TMS320C6678 has inbound and outbound address translations to realize address mapping between PCIe address and internal bus address. Inbound address translation is used to remap accepted incoming accesses from other PCIe devices to locations within the device's memory map. And outbound address translation can convert an internal bus address to a PCIe address of memory type.

Usually, a host computer reads data from a PCIe device actively using inbound address translator. This method has low speed when reading large amount of data. So, in our system, we use outbound address translator to remap the local bus address to PCIe bus address. When video capturing and video processing

are finished in video processing board, the video data and results of processing are transferred from internal address to the PCIe bus address directly using EDMA and outbound address translator. Then the host computer can get the data rapidly.

The inbound address translator is also used by the host computer to realize the access to DSP. Other PCIe devices can access the internal bus address via Base Address Registers and inbound translation registers. The BARs in EP devices are usually programmed by RC during the PCIe configuration process. When an incoming read/write access matches one of the BARs and the inbound translation registers are set correctly, the incoming address will be translated to internal bus address[8].

4. Design and Realization of Software

To realize the functions of real-time video processing and transferring, the software based on DSP and host computer is designed. The software control flow of real-time video processing system is shown in Figure 3. According to the running environment, the software can be divided into two parts: DSP software and PC software. The former is developed using CCStudio5.5 integrated development environment based on SYS/BIOS, which realizes initialization, video processing, EDMA transfer, data transfer via PCIe and interrupt handling. The latter is running under Windows7 operation system, and is developed using Visual Studio 2008, which realizes the driver of the board, PCIe configuration, video fresh and display, and the controls of board.

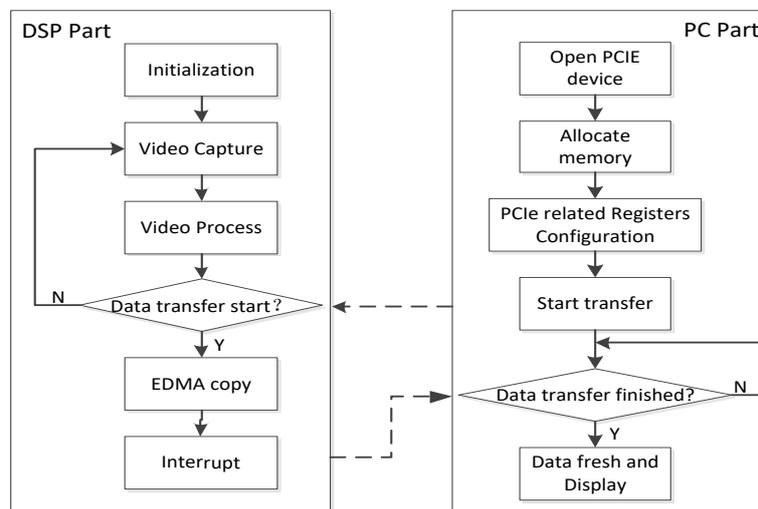


Fig. 3: Software control flow chart

4.1. The Software of DSP

For the captured video data, it is sent to DSP through SRIO. Every time a row of video data is sent, FPGA generates an external interrupt to GPIO of DSP. And when a frame of video data is sent, FPGA also generates another interrupt to another GPIO of DSP. Then DSP can process the data which is sent to DSP memory according to the interrupt numbers.

The DSP software mainly completes the following control processes.

The PCIe initialization should be accomplished at first, including PCIe configuration space setting, PLL setting, running mode setting, and link establish.

Then, the DSP software should get the allocated base address which will be used to configure inbound translation registers. When the computer is powered on, BIOS detects PCIe device and assigns base address to PCIe device according to its configuration.

After that, the inbound translation registers should be configured. The inbound translation registers translate PCIe bus address to local bus address, these registers include: IB_BAR n , IB_START n _HI, IB_START n _LO, IB_OFFSET n , ($n=0\sim3$). The value i of IB_BAR n register associates the match between Region n and configuration register BAR i . The IB_START registers is set to PCIe bus address, and the IB_OFFSET register is set to the start address of the internal address which is allowed to access by PC.

The board captures the video as long as the system is power on. If the DSP receives the command to start processing from host computer, it processes the video data sent from FPGA. C6678 has a given PCIe data space which are used for PCIe data transmission, and the address ranges from 0x60000000 to 0x6ffffff, a total of 256M. This memory space is divided to 32 equally-sized translation regions (Regions 0 to 31). These regions can be programmed to be of 1, 2, 4, or 8 MB in size via the OB_SIZE register. When video processing is finished, the DSP software copies the original video data and processed results via EDMA from internal memory to one region of PCIe data space. At this time, the outbound address translator will remap this region to a PCIe address using the value of corresponding configuration registers OB_OFFSET_INDEX n ($n=0-31$) and OB_OFFSET n _HI ($n=0-31$) for this region. The DSP only need to copy data from the internal address of DSP to PCIe data space, and then the data will reach the memory of host computer.

The DSP software configures EDMA related registers to generate an interrupt to host computer when EDMA transfer has been finished. Then the host PC can process the data immediately.

4.2. The Software of PC

The software of host PC includes driver software and application software, driver software is developed by winDriver. The control processes of the application software are as follows.

Firstly, the application opens PCIe device using driver software and then allocates and locks memory for video data and obtains the physical address of the memory.

To realize the data transmission via PCIe bus, the host computer needs to notify the DSP which PCIe address is to be transferred to. After the physical address of host computer which used to store video data is allocated, the application software should write the physical address to the outbound registers. That is, the registers OB_SIZE, OB_OFFSET_INDEX n , OB_OFFSET n _HI should be configured. Because the memory space covered by BAR0 in inbound direction is completely dedicated to accessing the application registers in both RC and EP modes. So the application can configure outbound registers which belong to application registers through BAR0. The application initializes the OB_SIZE register with the size value which should be 0, 1, 2 and 3 corresponding to region sizes of 1MB, 2MB, 4MB and 8MB and the corresponding indexed regions are bits[24:20], bits[25:21], bits[26:22] and bits[27:23] of the physical address respectively. The application software initializes OB_OFFSET_INDEX n register with the lower 32-bit PCIe address. If each region is 1MB, 2MB, 4MB or 8MB, then this field will be used to create bits[31:20], bits[31:21], bits[31:22] or bits[31:23] of the translated address respectively. The application software initializes OB_OFFSET n _HI register and this is a 32-bit field that represents bits[63:32] of translated PCIe space address if using 64-bit addressing. This register is required to be programmed with a Zero value if using 32-bit addressing.

When the host computer is ready, the application software controls the DSP via PCIe bus to start data transferring, and then wait for the data. Once an interrupt is received which means data transmission is finished, the application can get the data from buffer to display and store.

4.3. The Key Points of Software Realization

There are some key points which should be solved in the realization of software. The first one is inbound registers configuration. Usually, the configuration is realized in the software of DSP. Because the DSP can access Base Address Registers and Inbound Translation Registers directly, the DSP software program the registers IB_START n _HI and IB_START n _LO using the value of BAR n which want to match, and the register IB_OFFSET n is programmed with the internal address of DSP which is allowed to be accessed. Then the software of PC can read from or write to the internal memory of DSP through driver of the board.

The second one is outbound registers configuration. As described in Section 4.2, the physical address which allocated by host computer is used to program outbound registers. Suppose *size* is 1M, 2M, 4M, or 8M which is chose according to the actual situation. The physical address may not be aligned to *size*, but the 32 regions of PCIe data space are aligned to *size*. When DSP copies the video data from internal address to PCIe data space using EDMA, suppose *offset* is bits[24:20], bits[25:21], bits[26:22] and bits[27:23] of the physical address according to the *size*, and suppose *len* is the length to be copied. If the *offset* is not equal to zero, then we need to consider whether the *len* exceeds remaining length (that is *size-offset*) of the

corresponding region. If exceed, the direct copy will cause blue screen of computer. At this time, we should deal with it accordingly, such as dividing data into small blocks, or allocating physical address which is aligned to the size. If not, the video data can be copied directly. In addition, the bit 0 of $OB_OFFSET_INDEX_n$ must be set to 1, which enables outbound translation, otherwise, the corresponding region will not be able to read or write.

The third one is EDMA transmission. The DSP uses the EDMA copy to realize the data transfer from internal address to PCIe address without occupying CPU time. In the software of DSP, when EDMA reading, the source address is $[0x60000000 + (size*n) + offset]$ ($size$ =the value of OB_SIZE register; n =bits[24:20], bits[25:21], bits[26:22] or bits[27:23] of the physical address according to the $size$), the destination address is the internal address of the DSP. When EDMA writing, the source and destination address is inverse.

5. Experiment

When the hardware design and software realization are finished, we test the process flow and capability of whole system. Figure 4 is the real-time display result on the host computer.

This paper uses X1 PCIe bus, its maximum bandwidth of transfer is 5.0 Gb/s in theory. By actually testing, the results are as follows in table 1. The data transfer speed of the system is nearly 3.0Gb/s which can satisfy the bandwidth requirement in video real-time process system.

Table 1: Results of PCIe transfer speed

Test Item	Test results	
Data length(byte)	1M	2M
Time(ms)	2.775	5.5122
Rate(Gb/s)	2.882	2.902

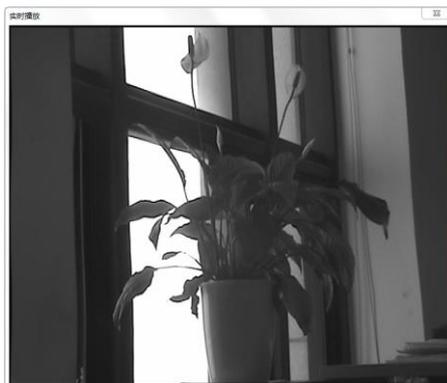


Fig. 4: Result of video real-time player

6. Conclusions

A real-time video processing and transferring system based on TMS320C6678 and FPGA is designed. The system realizes real-time video data transferring from FPGA to DSP using SRIO SWRITE mode. Once DSP receives an interrupt produced by FPGA when one row data was transferred, it processes in real time, and transfers data to host computer via PCIe bus, and the computer displays and stores the data at last. The time cost of data transfer using SRIO interface is less than 240ns, and the rate of data transfer via PCIe interface can reach 3.0Gbit/s. So, the system can realize the real-time video processing and transferring, and can display captured and processed image on screen in real time. The emphasis of the research is the design and realization of data transfer in real time. In the future, we will improve the conventional video processing method and optimize the software on the multi-core DSP.

7. Acknowledgements

This research work was funded by National Key R&D Program of China SQ2018YFB130465.

8. References

- [1] Texas Instruments Incorporated. TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor

(SPRS691E) [Z]. March, 2014. <http://www.ti.com>

- [2] LI Yi-mang, HE Xin, WEI Zhong-hui. Design and Implement of Real-time Image Processing System for IR Warning System Based on Multi-Passage[J]. *Chinese Journal of Liquid Crystals and Displays*, 2013,(1): 110-114
- [3] Qiu Zu-quan. Real-time Processing System for Infrared Video Signal Based on DSP+FPGA[J]. *Laser & Infrared*, 2007, 37(5):440-441.
- [4] Texas Instruments Incorporated. KeyStone Architecture Serial Rapid IO (SRIO) User Guide (SPRUGW1B)[Z]. November, 2012. <http://www.ti.com>
- [5] Texas Instruments Incorporated. KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User Guide (SPRUGS6D) [Z]. September, 2013. <http://www.ti.com>
- [6] JIANG Hong-Xu, LIU Ting-Shan, LI Hui-Yong. A High-efficiency Data Transmission Method Based on SRIO in FPGA+DSP Heterogeneous Video Processing System[J]. *Chinese Journal of Computers*, 2015, 38(6):1119-1130.
- [7] YU Jian, ZHOU Wei-chao, LIU Kun. Interconnection between DSP and FPGA Based on SRIO Protocol[J]. *Semiconductor Optoelectronics*, 2012, 33(6):145-148.
- [8] LE Liang, HU Shanqing, LONG Teng. Drive Program Design of PCIE Interface of TMS320C6678[J]. *Computer Engineering and Application*, 2012, 48(S2):370-373.