Research on the High-Speed Image Acquisition and Storage Technology based on TMS320C6748

Qiang Wu[1, a], Tao Song[1, b], Xuwen Li[2, c]

[1] College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing, 100124, China

[2] College of Life Science and Bioengineering, Beijing University of Technology, Beijing, 100124, China

[a]email: wuqiang@bjut.edu.cn, [b]email:st192@sina.com, [c]email:lixuwen@bjut.edu.cn

Keywords: High-Speed Image; uPP; SATA; DSP/BIOS

Abstract. For the huge amount of image data, high transfer rate, high real-time requirements of the imaging system, this paper proposes a high-speed image data acquisition and storage system based on TMS320C6748 DSP, with the use of uPP interfaces to capture images from the FPGA in real time, using GPIO and UART to implement synchronous work between FPGA and DSP, using external DDR2 memory to design Ping-Pong buffering mechanism to ensure real-time hard disk storage of huge image data. Experimental results show that the designed system can achieve image data acquisition and storage both stably and rightly in the high-speed data transmission.

Introduction

With the rapid development of digital signal processing technology, data acquisition and storage systems, especially for high-speed image in aviation, aerospace and other fields is more and more widely used[1]. The Airborne image acquisition and storage device is a core component of the visual system of the aircraft, and is responsible for the image acquisition and storage of the imager. Airborne infrared, radar and other imaging devices transfer high-speed images with high resolution, and make high real-time, accuracy, stability demands. Embedded system with high real-time, high integration, low cost and other advantages plays an important role in the development of high-speed data acquisition and storage. Embedded image processing systems often use FPGA+DSP hardware architecture model[2][3], FPGA for the image acquisition and pre-processing, DSP for further processing, so that the communication performance between FPGA and DSP is especially important and to choose a stable and reliable interface technology for data transmission is very important.

TI's TMS320C6748[6] belongs to C6000 high-performance DSP[4], clocked at up to 456MHz, with a strong floating-point computing ability up to 3648/2746MIPS/MFLOPS. TMS320C6748 has rich peripheral resources, with uPP. It is the first integrated SATA TI device, provides support for the DDR2. The Universal Parallel Port (uPP) peripheral is a multichannel, high-speed parallel interface with dedicated data lines and minimal control signals. It is designed to interface cleanly with high-speed ADCs or DACs, FPGAs or other uPP devices to achieve high-speed digital data transfer. For the high performance requirements of image acquisition and storage systems, this paper proposes an uPP interface-based embedded high-speed image acquisition and storage system, using its internal integrated uPP and SATA provides a high-speed and reliable channel for the massive image data storage, in TI's TMS320C6748 processor platform.

System Structure

In the entire image acquisition and storage system, front-end FPGA is responsible for collecting high-speed LVDS images and complement pretreatment, then images is sent to the DSP for further processing and storage. The FPGA is the Xilinx Virtex-5, and the DSP is TMS320C6748. Image resolution is 320*256*16bit, and 320*258*16bit with parameters and fault-tolerant data added by FPGA. The frame rate reaches up to 150f/s. System structure is shown in Fig.1. DSP connects with

FPGA through uPP receiving the images, through UART and GPIO to achieve the synchronization during the transmission of the image and command signal. DSP use internal integrated SATA II controller to connect directly to the SATA interface of SSD, to write the image data into the hard disk. The DSP connects an external DDR2 memory as a data buffer. DDR2 SDRAM choose MT47H16M16-37E, with capacity 4Meg*16bits*4banks = 256Mb and bus speed 533MHz.

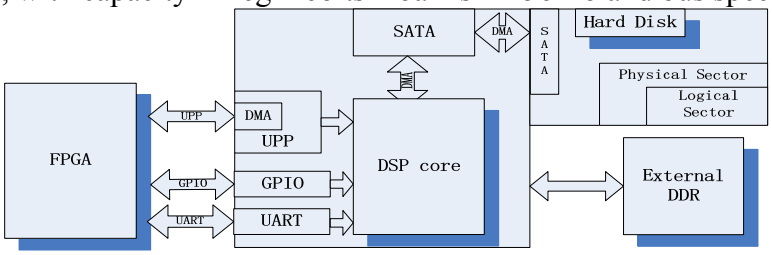


Fig.1 System structure

Image Data Transfer Mechanism and Synchronization Mechanism Design

uPP includes two separate two-way channels(A and B) with dedicated data lines. Each channel is 8-16bits width. The transmission speed is up to 75MHZ. Protocol is simple, only 2-4 configurable control pins per channel. It includes an internal DMA controller which includes two channels to maximize throughput and minimize CPU overhead, not taking up CPU EDMA resources. UPP transmission rate is much faster than serial interface as SPI, UART; compared with HPI uPP has more simple agreement and much faster speed. In this design, uPP is configured as interrupt mode, the clock falling edge, uPP clock provided by the FPGA. Channel A is used configured as a 16-bit wide, no data package, single-signal rate. It works in window mode which is set to 258*640 bytes. DMA Channel I services channel A, as shown in Fig.2. When receiving data, uPP interrupt counts, compared with GPIO interrupt to verify the correctness of the data frames.

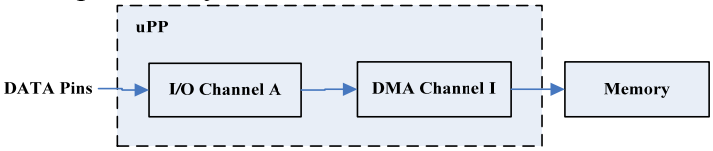


Fig.2 Single-channel mode data stream

The data amount is huge, and the system executes acquisition and storage simultaneously, so it designs Ping-Pong double buffer alternating storage mode, using an external DDR2 memory. It stores the received data into DDR2 through the DMA bus of uPP and moves data from the DDR2 to the external hard disk through DMA bus of SATA.

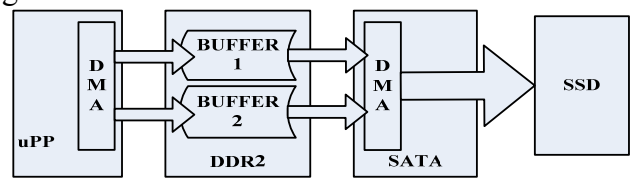


Fig.3 Double buffer structure diagram

According to the image size, it defines two mutually non-conflicting memory spaces on the DDR2: Buffer I and II, and pointers bufPing and bufPong, and it operates the pointers to complete the data input and output. When the transmission starts, input data is stored in Buffer I, and the address pointer is incremented by frame size. When Buffer I is filled with data, SATA's DMA moves data from Buffer I to the hard disk for storage, and the new uPP receiving buffer is set Buffer II. Data reception and storage operate simultaneously. Receiving buffer address changes each 50 frames. Since uPP and SATA occupy the two buffers simultaneously and separately, it is necessary to set up two data flow flags, marking current respective buffer to prevent data store errors.

In order to ensure the integrity of each frame, it uses GPIO synchronization at each frame transmission beginning. GPIO principle is simple. It is flexible to configure. The hardware circuit and driver software are easily accessible. The DSP connects to FPGA through a GPIO in edge-triggered interrupt mode. FPGA first generates a GPIO rising edge to trigger interrupt on DSP

before each frame sending to the DSP, to achieve frame synchronization. Since the actual image data come from multiple data sources of infrared, radar imager, etc., the DSP needs to distinguish attribute of each frame. In addition, DSP requires reporting the DSP status and other information to the FPGA. UART protocol is simple and easy to implement. This paper designs instruction agreement, and transfers the instructions via UART to achieve synchronization of image attributes and processor status information between DSP and FPGA.

DSP Software Design and Implementation

The DSP software is implemented based on TI's DSP/BIOS [5] on CCS5.2 which greatly simplifies the implementation and shortens the development cycle. The software is divided into four task modules: uPP module, GPIO module, UART module, hard disk storage module. uPP module is responsible for uPP configuration, GPIO module for frame synchronization and interrupts counting, UART module for UART command transmission, and hard disk storage module for interfaces initialization and read and write operations. As shown in Figure 4. uPP task, GPIO task, UART task are implemented in HWIs, with priority HWI_INT4, HWI_INT5, HWI_INT6 from high to low; task0 is established in TSK with the priority lower than all HWIs, the hard disk storage task is implemented in task0. All the threads are established through DSP/BIOS configuration tool.

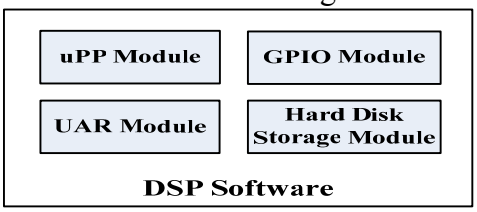
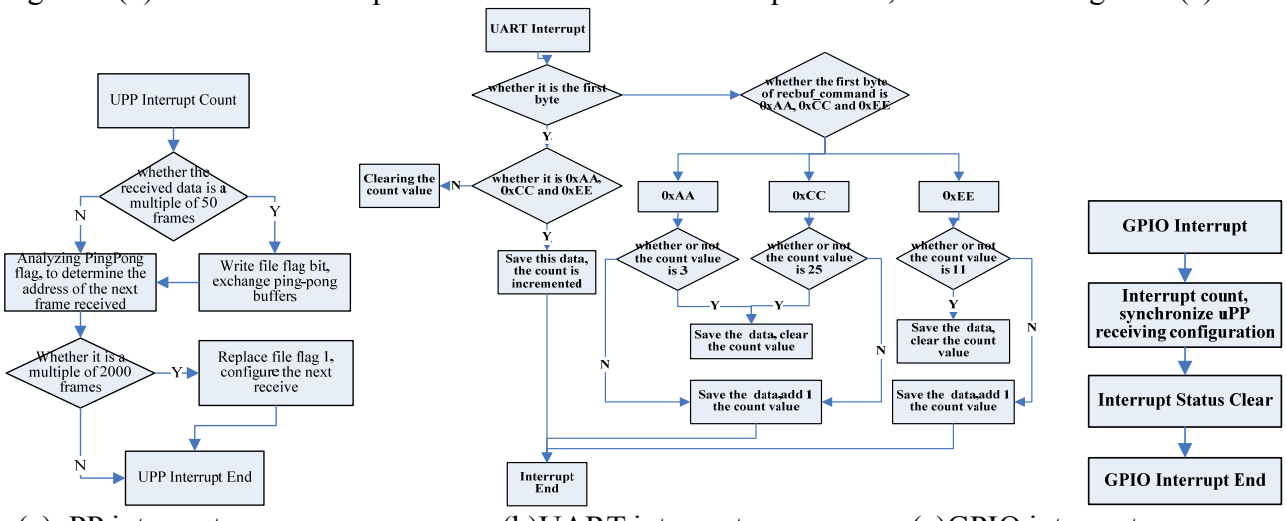


Fig.4 DSP software architecture

Task0 executes uPP, UART, GPIO, and SATA initialization, sends DSP status and handshaking information to the FPGA through the UART, and stores image data according to the actual reception situation. In uPP interrupt service routine uPP interrupts count. Upon receiving 50 frames, Ping-Pong buffer flag Flag_pingpong negates, the two receive buffers exchange; the pre receive buffer is recorded as preFlag_pingpong to help determine the source address to write into hard disk, the new receive buffer is recorded as Pingpong for receiving new images; the flag flag_Sata is set 1, and can be cheched by Task0 to do the write operation. Before the end configure the uPP again. Figure 5 (a) is the uPP interrupt service routine flowchart. The UART interrupt service routine executes UART instructions analyzing, and saving instructions in the instruction buffer. Task0 simultaneously check the instruction buffer to do the corresponding specific tasks, as shown in Figure 5 (b). In GPIO interrupt service routine GPIO interrupts count, as shown in Figure 5 (c).



(a)uPP interrupt response (b)UART interrupt response (c)GPIO interrupt response

Fig.5 The main workflow of each thread task

Test results

To test the performance of the designed system, data acquisition and storage are tested under different conditions: different amount of image frames, different frame rates. The IPC with LVDS image transmission function is used as an image source, to output images stably at some rate. The PC playback software is used to unpack results stored in the hard disk for parsing and validation.

Images are sent consecutively by 1000 frames, 2000 frames, ..., 100000 frames, rates of 10f/s, 50 f/s, 100f/s, 150f/s, each frame size is 640*258*16bit. The results are shown in Table 1.

Table 1 The results of storage

frame rate	1000	2000	5000	10000	20000	50000	80000	100000
10f/s	Correct	Correct	Correct	Correct	Correct	Correct	Correct	Correct
50f/s	Correct	Correct	Correct	Correct	Correct	Correct	Correct	Correct
100f/s	Correct	Correct	Correct	Correct	Correct	Correct	Correct	Correct
150f/s	Correct	Correct	Correct	Correct	Correct	Correct	Correct	Correct

Set amount of data written to the hard disk a single time in different sizes of 50 frames, 100 frames, and 500 frames. Record each time when data are written, using the internal DSP precise timing. The results are shown in Table 2.

Table 2 The results of writing time (ms)

Frames	1	2	3	4	5	6	7	8	9	10
50f	88.85	89.21	88.84	88.77	88.80	88.84	88.97	88.80	89.21	88.68
100f	178.25	177.75	177.80	177.99	177.67	176.88	177.71	179.06	178.19	177.5
500f	888.7	888.7	888.6	887.1	889.2	888.0	887.5	888.6	886.9	888.73

Table 1 show that when the amount of data in different cases, the frame rate not the same, the final data frames contents stored in the hard disk are all correct. Calculated from Table 2, the fastest rate of hard disk storage of the system can be achieved 88.8MB/s, and fully meet the requirements in actual using environment. Test results show that the system can achieve image data acquisition and storage both stably and rightly in the high-speed data transmission.

Conclusion

The imagers bring high performance requirements of real-time data acquisition, transmission and storage. This paper designs and implements a high-speed image acquisition and storage system based on TMS320C6748 DSP. It uses its internal uPP, SATA to provide a reliable channel for massive image data storage, improving the transmission efficiency and accuracy, and implement DSP software based on DSP/BIOS. The testing results show that the system is stable and right in high-speed image data acquisition and storage to meet the actual needs.

References

[1] Kewei Qian, Yonghui Xu, Jinsong Liu. Development of Data Acquisition System based on DSP/BIOS [J]. Modern Electronics Technique, 2012，30(25):12-14,18.

[2] Chaohai Li, Chao Luo, Donghui Huang. The design of high-speed transmission system of spectrum receiver based on OMAP-L138[J]. Embedded Technology.2014,40(7):6-8,12.

[3]Wang Zhe. Research and implementation of data acquisition system based on OMAP-L138 (D). Harbin：Harbin Engineering University，2013.

[4] Qingrong Zhang, Xunshwng Ji, Zhenkai Zhang. MS320 C6000 DSP architecture and hardware design principle[M]. Beijing: Beihang University Press.2008:46-61,328-342.

[5] Qicong Peng, Qing Guan. DSP integrated development environment——Principles and Applications of CCS and DSP/BIOS [M]. Beijing：Electronic Industry Press，2004.

[6]Texas Instruments．TMS320C6748 DSP Technical Reference Manual：SPRUH79A，2011