TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.6, June 2014, pp. 4387 ~ 4392
DOI: 10.11591/telkomnika.v12i6.5472

A Network Disk Device Based on Web Accessing

QunFang Yuan[1], Wenxia DI[2], YaoLi Wang*[3]
[1,3]College of Information Engineering, Taiyuan University of Technology
Taiyuan, 030024, China, Ph./Fax: +0086-3516014052/3516014052
[2]Taiyuan Normal University
Taiyuan, 030012, China, Ph. /Fax: +0086-3514885311/3514885311
*Corresponding author, e-mail: willingwan@gmail.com

Abstract

The future personal and home demand of big data helped fuel an explosion of development in network disk device market. Compared to a traditional USB hard drive, it separated the user from the hard disk, enjoyed the advantage of accessing from anywhere for unspecific user and easy operation and easy expansion. The paper established the remote access to hard drives via browser with Xilinx's XUPV5 LX110T FPGA development board. The system mounted an ATA host controller IP core and connected it to the network with RJ45. By porting PetaLinux operating system to the MicroBlaze embedded processor of FPGA, an independent FAT32 file system module was added, the users could establish the remote access to hard drives with specific CGI programs through the WEB browser. It was tested that Personal/Home Network Disk can meet the standards of basic file operation.

Keywords: *network disk, FPGA, FAT file system, CGI*

1. Introduction

In the age of big data [1], an embedded network storage device based on FPGA has been designed and developed in this paper to meet the pressing demand of accessing remote file resources through WEB. USB hard drive can only provide file resource information to its holder, it does not satisfy the random accessing demand of non-specific users. With the continuous improvement of high performance network technology and the gradual rising of network storage mode [2], it is a storage mode that uploads files to server host which can be accessed by users through a client anywhere through the internet. A personal/home network disk based on FPGA is designed in this paper, and the system overall architecture is shown in Figure 1 (the shaded part is the main work completed in this paper).

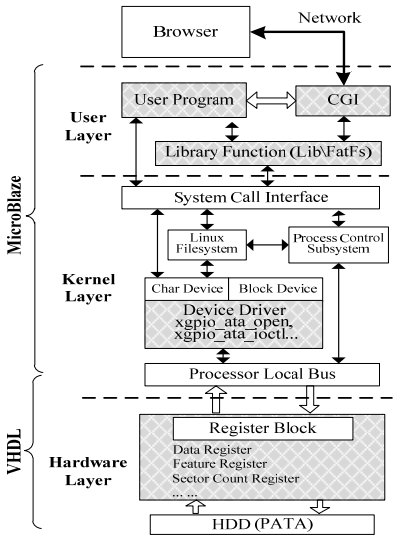

Figure 1. System Architecture

Received December 21, 2013; Revised February 9, 2014; Accepted February 22, 2014

The system is divided into hardware layer, kernel layer and user layer; user layer is the interactive operation interface of the system, hardware layer achieves the various operations on underlying hard disk of the system, kernel layer is responsible for the transmission of driver parameters and data between user layer and hardware layer. IP core of ATA host controller is designed and mounted by using XUPV5 LX110T FPGA development board; B/S mode is adopted by client. As client is a standard browser, users can complete the remote operation on the hard disk by calling relevant CGI (Common Gateway Interface) through WEB browser on a PC connected to the internet. Personal/Home Network Disk keeps working all day long, easy and convenient, and the user storage capacity can be expanded through mounting multiple hard disks. The system can be further developed and perfected in the future according to the general method proposed in the paper.

2. System Design

2.1. Introduction of Platform

The hardware platform structure of the system is shown in Figure 2, the system adopts Xilinx XUPV5 LX110T FPGA development board, embedded with modules such as MicroBlaze soft core, Ethernet controller, FLASH, DDR and so on, achieving inter-communication between modules through PLB bus. The low-level data storage device uses the ATA hard disk with IDE interface. By connecting the extended interface of V5 development board to FPGA chips, client achieves the remote access to the hard disk through accessing network.

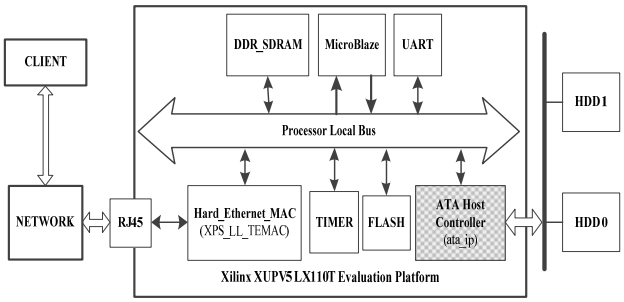

Figure 2. Hardware Platform Structure

In the modules above, the shaded module needs to be developed by users, while other parts are provided by XUPV5 LX110T FPGA development board

2.2. ATA Host Controller

This paper takes Seagate ST340014A hard disk (40GB) with IDE interface as a data storage device, completes the design of ATA host controller based on ATA/ATAPI-6 protocol [3] and packaged it into an IP core for integrating into different bus architecture of ASIC or SoC systems easily [4]. The structure of the ATA host controller is shown in Figure 3.

ATA host controller includes Reset, PIO Control Module, MDMA Control Module, Ultra DMA Control Module, receive and send buffer, which are achieved with VHDL and simulated and tested to be correct. The reset module resets other modules under initialization or abnormal conditions; PIO Module controls the read and write hard disk operations of PIO mode; MDMA Module realizes the read and write operations on multiword DMA mode of the hard disk; Ultra DMA Control Module is for the control of the whole process of achieving hard disk access in the way of Ultra DMA; receive and send buffer module is used to control data buffer [5].

Besides designing strict sequential logic and state machine, the key to implementing ATA host controller also includes realizing the read/write operation on the register assigned by protocol. The transmission of command and command parameters and the transmission of device state information are completed through the read/write operation on register. The data transmission of registers in ATA protocol accords with PIO transmission mode, the register address is gated by setting CS1-, CS0- and DA [2:0], DIOR-/DIOW- signal controls read and

write; Meanwhile, ATA host controller and hard disk puts the data to be transmitted on data bus and data on the bus are read by controller or hard disk according to data transmission direction. The definitions of registers [6] in ATA/ATAPI-6 protocol are listed in Table 1.

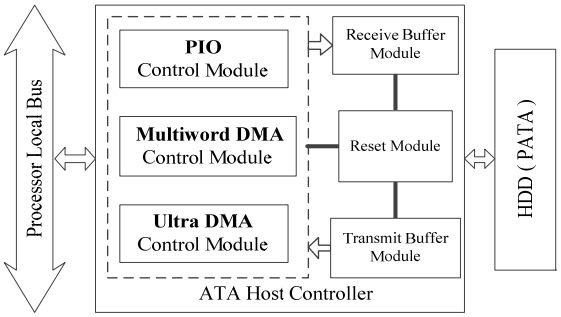

Figure 3. ATA Host Controller Structure

First, the ATA host controller module has been packaged into user-customized IP core and connected to the MicroBlaze processor with PLB bus; then connecting the kernel driver to hard disk by packaging XGpio functions [7], the hard disk driver is defined in Table 2.

Table 1. Register Address and Name

Address CS1-	CS0-	DA	Register Name Read Operation	Write Operation
1	0	0	Data Register	
1	0	1	Error Register	Feature Register
1	0	2	Sector Count Register	
1	0	3	LBA Low Register [7:0]	
1	0	4	LBA Mid Register [15:8]	
1	0	5	LBA High Register [23:16]	
1	0	6	Device Register / LBA [27:24]	
1	0	7	Status Register	Command Register
0	1	6	Alternate Status Register	Device Control Register
1	1	*	Data Port	

Table 2. Definition of Hard Disk Driver

ID	Function Name	Description
1	xgpio_ata_init	Initialization
2	xgpio_ata_ioctl	Control Device
3	xgpio_ata_open	Open Device
4	xgpio_ata_fops	Device File Operation
5	xgpio_ata_dev	Register Device
6	xgpio_ata_exit	Cancel Register
7	xgpio_ata_read	Read Data
8	xgpio_ata_write	Write Data

2.3. FAT File System

File system is a bridge between the operating system and the underlying hardware, which has been widely used in embedded systems and other fields. U disk, MP3, MP4 and other digital device almost all adopt the Microsoft FAT 32 file system [8], but the Microsoft Corporation haven't released the source code of the FAT 32 file system, so this paper used an open-source project FatFS [9], which is independent from the underlying DISK I/O layer, not depend on the hardware architecture and has good compatibility with the Windows FAT file system.

FAT file system can be divided into three levels [10]: demand, logical and physical. The demand layer supports for user's basic file operation; the logical layer implements a mapping

between FAT file system and the physical layer; the physical layer implements interfaces to physical storage media, including read or write sectors, hard reset operation. The physical layer has achieved five interfaces: disk_initialize (initialize the disk drive), disk_status (get disk statue), disk_read(read sector), disk_write(write sector), disk_ioctl(control equipment associated characteristics). Using these five interfaces, the demand and logic of the FAT file system can be realized.

The five interface functions of the physical layer call PetaLinux kernel driver [11] to achieve the connecting between the underlying hard disk and file system, and then realize the various processes of the Fat file system. For example, the operational processes of disk_read, xgpio_ata_ioctl and xgpio_ata_read defined in the previous section achieve read and write function of ATA IP core register, first pass the parameter and data to the kernel; then xgpio_ata_ioctl and xgpio_ata_read are encapsulated in the disk_read function of the application layer, it will meet the demand of the file system logical layer so as to implement the interfaces of the demand layer. All the Fat file system operational processes are packaged into a library, so the users will call the process directly. All file system interface functions are defined in the application layer for CGI to call.

The FatFS file system physical layer interface functions work properly after testing, also meet the demands of the logical layer and demand layer.

2.4. User Interface

The operating system PetaLinux is an industry standard development environment for embedded Linux on Xilinx FPGAs [12], users can easily establish system develop environment with its tools. PetaLinux also integrates part of some commonly used IP cores, it will reduce the workload of porting and programming the driver programs; the source code also includes much scripting which could simplify operations. Compiling the generated FatFS lib and putting it under the directory of PetaLinux /bin, users can call it as an application program.The process is shown in Figure 4.

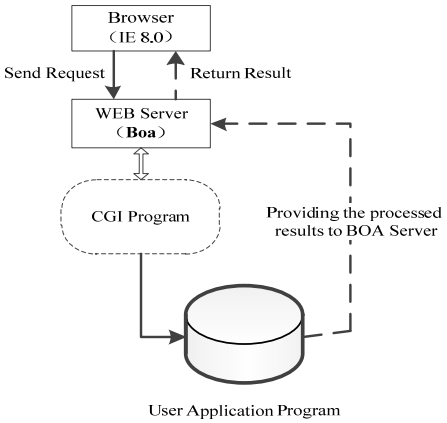

Figure 4. CGI Working Process

The user interface and the system interface are written in C language, they also meet the specification of CGI [13]. Server uses a single-task, high performance, embedded web server Boa [14] which fully support CGI. When porting the PetaLinux, it needs to enable Boa to be supported.

When the user enters the IP address, the user's browser will establish a TCP connection to the Boa server; CGI request will be issued to Boa server in PetaLinux when user's actions have been made on WEB page, Boa server will create a CGI process when it receives the request, the specific environment variables, parameters and standardized input program will be passed to the CGI program. The results will be passed to the Boa server while the CGI program completes processing. The server displays the data with identification information added in HTML Web pages and forms to users [15-17].

For example, the user who clicks on the "Read" button will trigger in the form of HTML pages and issues an HTTP request to the server (CGI request), sends to the address of the ACTION attribute in <FORM> tag.

3. Demonstration of System

The personal/home network disk has the following characteristics:

It offers superior security and privacy, the storage device is visible to user, ensuring the data's privacy and security.

It also allows for the function of plug and play, easy to setup, the system can work immediately when connected to power supply and network cable.

The personal/home network disk can be accessed via browser; the PC which is connected to the network can access the personal/home network disk with the user's browser.

Easy to operate and extend, by adding the number of hard disk and changing the storage device, the personal/home network disk supports user-defined extension of storage capacity.

The personal/home network disk keeps working all day long, which can be accessed or used through Network Disk Management System. The system run smoothly, safe and stable, it could easily go beyond local area network and connect to wide area network. Enter the IP address of the development board in browser: 192.168.0.10, then the connection will be established between PC and FPGA development board. The user operation interface is shown as follows in Figure 5.

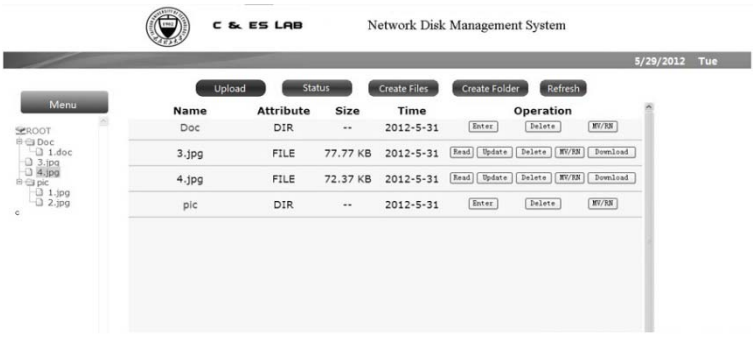

Figure 5. User Operation Interface

When click "Upload" button, the system indicate users to choose the file to be uploaded, then click "Submit" button and execute the upload operation: the data will be written from local PC to remote hard disk.

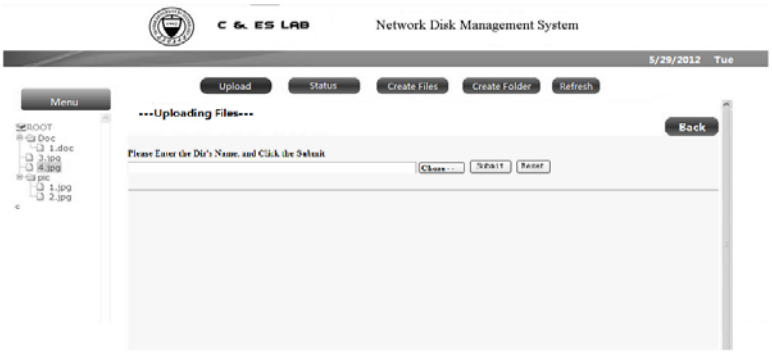

Figure 6. Upload File

4. Conclusion

The paper designed and implemented a personal/home network disk, mounted ATA host controller IP core to the XUPV5 LX110T FPGA development board, ported PetaLinux operating system, added independent FAT32 file system module, designed the interface between user and hard disk, and the feasibility of the program is established through practical system verification. The personal/home network disk can be further developed and perfected in the future according with actual demand.

Acknowledgements

It is a project supported by Shanxi National Natural Science Foundations (No. 2013011015-1).

References

[1] Shi Shuicai. Understanding of big data, practice of big data. *http://wenku.baidu.com/view/1a3828050740be1e650e9a45.html*. 2012.
[2] Deng Yuhui. Mass Storage System Based on Network Attached Disk Array. PhD thesis of Huazhong University of Science & Technology. 2004.
[3] T13 Working Draft. Information Technology-AT Attachment with Packet Interface-6. T13 Technical *Committee*. 2002.
[4] Qu Zuofu. The Research for ATA Device IP Based on FPGA. Master Thesis of Liao Ning Technical University. 2008.
[5] Zhao Zhenting. The Design and FPGA Implementation of Network Hard-disk Storage System*. Master Thesis of Taiyuan University of Technology.* 2012.
[6] Li Xiaojuan. Ultra DMA Write Controller Based on FPGA. *Science Technology and Engineering*. 2007; (15).
[7] Zhang Gang, Zhang Bo, Chang Qing. Design of SoC system. Beijing: National Defense Industry Press. 2012.
[8] Microsoft Cor. Microsoft EFI FAT32 File System Specification. *Microsoft Coporation*. 2000.
[9] ChaN. FatFs Generic FAT File System Module. *http://elm-chan.org/fsw/ff/00index_e. html*. 2011.
[10] Wang Yaoli. Research on Cloud-based Storage Information System. PhD thesis of Taiyuan University of Technology. 2012.
[11] Petalogix. Kernel & Driver Customisation. *http://www.petalogix.com/services/platform-services/kernel-driver- customisation/?searchterm=driver*. 2009.
[12] Xue Huimin, Wu Chuanhua, Lu Houbing, etc. Transplant of PetaLinux (embedded OS) based on MicroBlaze. *Microcomputer Information*. 2011; (8).
[13] Li Lei. Application of CGI Procedure to Realize Development Web Method. *Science Technology and Engineering*. 2009; (06): 1611-1613.
[14] Luo Yi, Zhang Jianwu. The Application and Realization of Embedded Linux WebServer BOA. *Journal of Hangzhou Institute of Electronic Engineering*. 2004; (01): 52-55.
[15] Liu Hui, Liu Yimin. Research and Design of Embedded Web Server. *Digital Communication*. 2011; (4).
[16] Wei Ma, Xiaoyong Li, Yong Shi, Yu Guo. A Virtual Machine Cloning Approach Based on Trusted Computing. *TELKOMNIKA Indonesia Journal of Electrical Engineering*. 2013; 11(11): 6935-6942.
[17] Li Yue-xin, Hong Zong-xiang. Research of Semantic Network Knowledge Representation and Query Algorithm Based on Relational Model. *TELKOMNIKA Indonesia Journal of Electrical Engineering*. 2013; 11(11): 6591-6599.