The Design of USB Interface for Large-format Plotter

Wang Quan, Wan Bo, Zhang Wengang, Tian Yumin; Research Institute of Peripherals, Xidian University; Xi'an 710071, China; qwang@xidian.edu.cn, wanbo@xidian.edu.cn, wgzhang@xidian.edu.cn, ymtian@xidian.edu.cn

Abstract

Plotter interface system is responsible for receiving commands and datum from host computer and then transferring them to print-control-system. On the basis of protocol of USB and principle of large-format plotter, a newly-designed plotter interface system, which takes high-speed USB chip, CY7C68013, as its core, is recommended in this paper. The structure of CY7C68013 is introduced, and how to work of plotter interface system and how to transfer datum through USB is explained in particular in the paper.

1. Introduction

The traditional plotters gain printing data from PCI bus. Nowadays, the requirement for printing speed and the amount of printing data has been grown significantly. The drawbacks of PCI are that it does not support hot swap, inconvenience of dismounting, PCI is greatly restricted in the practical application.

Hence, the need for advanced bus is urgent. USB is marked for its fast transmission speed, good Anti-jamming, supporting hot swap, easy to mount, set up and extendable, more and more application adopt it as the first choice. The transmission speed for USB 2.0 protocol is up to 480Mbit/s. It can be incorporated into the circumstances of huge amount of data transferring as well. It is suffice to support the speed requirement for the plotter. Consequently, we come up with a plan for the plotter interface system based on USB 2.0 on the basis of our understanding to the principle of plotter and USB.

2. Overall designs

2.1 Function introduction and the requirement to interface system

The premier function of plotter interface system is to provide the function of data transferring and command retransmitting between the host and printing control unit, using proper interface to gain data from the host and transmit them to its subordinating printing control unit, obtaining user's commands from the host and transmitting them to print-control-system, gaining control to the working mode its subordinating printing control unit, as well as ensuring the plotter functions well.

Currently, the speed of transmission between print-controlsystem and interface system is 1MHz. We firstly define the theoretical reading speeding from interface system as 2MHz, so the transmitting speed from the host to interface system then to printing control unit is lower bounded by 2MHz. We utilize USB between interface system and the host. Since the transmitting speed in USB1.1 protocol is merely 1.5Mbps, it does not satisfy our needs, we have to adopt the chips that support USB2.0 protocol to happy our needs. There should be a FIFO interface provided by interface system for the operation of printing control unit.

2.2 system overall design

Our system uses a CY7C68013-128AC which is distributed by CYPRESS. It supports USB2.0 protocol, the rate of transfer reaches 480Mbit/s. considering the buffer of this chip is fairly limited (the maximum buffer size in CY7C68013 is 4k). Some problems arise when the host waits for too long if the host tried to transmit large blocks of data, so we directly put a larger FIFO as buffer between USB chip and print-control-system. Finally, we use the IDT's IDT7206L15, 16KB, standard reading period is 35ns; CY7C68013 works in FIFO mode in our design, we also uses another External dominate chip to corporate between USB chip and IDT7206. We choose the MAX7000 CPLD EPM7128 from ALTERA as our choice. it integrates 2500 Logic gates, 84 pins. The structure of interface system is shown in figure 1:



Figure 1: System Architecture

2.3 system analysis

The protocol between the host and interface system is USB2.0. Maximum transmitting speed is 480Mbit/s (that is 60MByte/s); CY7C68013 works in FIFO mode, controlled by CPLD to transmit data fromCY7C68013 to IDT7206. Theoretically, it takes 100ns to read a byte from CY7C68013, namely 10MByte/s. The average Read and write cycle of IDT7206L15 is 35ns. Hence, the transmitting speed of interface system satisfies the speed requirement of print-control-system in theory.

3 Hardware designs

3.1 Introduction of CY7C68013

CY7C68013 is the series of EZUSB_FX2 control chip developed by CYPRESS, it perfectly support USB2.0. The developer could know nothing about the detail

Protocol and simply operate on endpoint of CY7C68013. This SIE integrated in this chip can accomplish data transmitting automatically. The developers can just need to write to endpoint buffer if they want to send data to endpoint. SIE is able to "USB-lize" these data according to USB protocol and transmits to the

host. When the host tries to send data to CY7C68013, SIE is capable of transmitting the data to the corresponding endpoint buffer. CY7C68013 supports "full-speed" and "high- speed" mode, the maximum transmitting speed is 480Mb/s. SIE is compatible to USB1.1. The structure of CY7C68013 is shown in figure 2:



Figure 2: System Architecture of CY7C68013

CY7C68013 integrates an enhanced 8051 internally and provides a standard data bus. It can connect to Data read-write chips such as memory. R/W frequency is 12MHz.

CY7C68013 can work in the following two modes:

1. GPIF mode: GPIF has 9 bit Address lines, 16 bit data lines, 6 input signal lines and 6 output signal lines, the developers can make these signals satisfy special needs by programming.

2. FIFO mode: CY7C68013 is similar to FIFO under this mode; endpoint buffer is the storage space for this FIFO. It provides signals such as "empty", "full", and "half-full" etc. In addition, it provides the other two address lines to choose currently operated endpoint. External dominate chip just send the address of the required endpoint and read/write to endpoint buffer according to "empty", "full", "half-full" signals. Since the interface of FIFO is easily set up and connects, the speed meets the requirement, CY7C68013 works in FIFO mode in our design.

3.2 hardware system designs

Because CY7C68013 works in FIFO mode in our system, EPM7128 acts as External dominate chip and supervise the working of the whole system.IDT7206 provides buffer between CY7C68013 and print-control-system. The workflow of this system is as follow: when the system powered, EPM 7128 sends the corresponding address ADR to select the endpoint of CY7C68013. After IDE sends the data to endpoint buffer from the host, when nEF is high means the endpoint buffer has data, if EMP7128 checks out that IDT7206 is not full, it sends out nRD signal from endpoint buffer to data bus $D0 \sim D7$ and notifies IDT7206 with a writing signal to write the data to IDT7206. It sets nRD and nWR high after the data writing done. Then a data transmitting is accomplished. Print-control-system determines whether to send a read signal by the null signal of IDT7206. The control endpoint of CY7C68013 sends the command received from the host to the next level print-control-system from serial.



Figure 3: Hardware system

4 Firmware designs

4.1 EndPoint installation

Users operate on USB interface by endpoint of CY7C68013, CY7C68013 provides seven endpoints. EP0 is default control endpoint, it can be used to transfer descriptor, command and status without any set up. EP! IN is the input endpoint, it can be set up to interrupt, synchronization, and Mass transport endpoint. The maximum data package is 1024 bytes, they share 4KB endpoint buffer. The plotter just receive data and control command in this system, it doesn't send data to the host, so We just define endpoint 2 as the input endpoint, the transmission mode is mass transport, along with 4 KB buffer.

4.2 Enumeration of devices

The enumeration is defined as: The host first interacts with the devices and gets the device descriptors of the USB devices after the plug-in of the device. If the host sends Interface descriptors orders to the USB devices, the latter should act in response to the command and send back the interface descriptor to the host.

4.2 Functions of firmware

The firmware runs on CY7C68013 and controls the working procedure; it mainly fulfills the following functions:

1 Initialize the register and variable, set the working mode of the chip, set up the number of endpoints and the size of buffer; the chips work in FIFO mode in our system, working frequency is 48MHz with the endpoint 2 as input endpoint, transmitting mode is mass transport, the size of buffer is 4KB.

2 Works in response to the control command from the host; CY7C68013 will send an interrupt signal after receiving the control command, the firmware can determine the type of corresponding command inside interrupted procedures.

5 software designs

5.1 The working principle of software

Software works on the host; it is a special program that implements the hardware control and call from the user. It includes the application and driver. In the first place, the application invokes the API so as to operate on the hardware. WIN32 subsystem will send the commands from user to I/O manager. I/O manager is responsible for integrating those commands to a IRP. IRP is a data structure that is comprised of the hardware related information.

It contains all the parameter of the upper level application. I/O manager will invoke the specific driver that handles the IPR routine according to the main function codes wrapped in IRP package. The detailed hardware operation is implemented in routine.

5.2 Driver designs

USB driver adheres to the WIN32 WDM. It includes the bus driver and function driver. Bus driver is provided by operating system. A bunch of underlying function that calls on the hardware has already defined. The function drivers are fabricated by developers. Different programs are in need to respond to all types of IRP routines. DriverEntry procedure is the entry point for the driver. The driver will invoke this function when it is called in the first place. The main function of DriverEntry is to figure out the specific IRP routine for the I/O manager besides initializing some global variables. The routine will invoke hardware interface function that is provided by bus driver to operate on hardware, for instance, the routine which handles the mass transport invokes UsbBuildInterruptOrBulkTransferRequest function, to deal with endpoint transport, one needs to call on IoBuildDeviceIoControlRequest function.

5.3 Application designs

The user level applications operate on USB devices through a series of Win32 API. It is similar to operate on file in Windows. The user calls CreatFile() function to open devices. ReadFile and WriteFile implement write and read to hardware separately. DeviceIoControl controls the read and write to the haredware; CloseFile is responsible for close the device after the operation is done.

6 Performance and improvement

The interface system can satisfy 1MHz transfer of the next level printing unit when we debug the plotter system. EPM7128 reads data from IDT7206 in 2MHz to simulate the printing unit reading data in 2MHz. This system guarantees the data transmission reliability. Hence we meet our design objection.

For the slave FIFO interface should connect to the control chip, operate on FIFO is somewhat slow, the cure is to utilize GPIF mode, write data directly to IDT7206. Theoretically, the data travel from CY7C68013 to IDT7206 consumes 40ns, the rate of

transport could reach 25MHz. Under this circumstance, EPM7128 is redundant; we can further enhance the transfer speed.

7 Conclusions

In paper we provide a plan to implement plotter based on USB2.0 protocol, we also provide the hardware and software design. We hope it will be help to the readers.

References

- [1] Cypress Semiconductor Corporation. CY7C68013 EZ-USB FX2 USB Microcontroller High-Speed USB Peripheral Controller, Rev[A]. November 25, 2001.
- [2] Compaq, Intel, Philips etc. Universal Serial Bus Specification, Rev [2.0]. April 27, 2000.
- [3] Compaq, Intel, Philips etc. Universal Serial Bus Specification, Rev[1.1].September 23,1998
- [4] Compaq Cypress Semiconductor Corporation. CY7C68013 EZ-USB FX2 USB Microcontroller High-Speed USB Peripheral Controller, Rev[A]. November 25, 2001.
- [5] Wu Tao, Han Wei; "Design and implementation of a data acquisition based on USB2. 0 interface". SPACE ELECTRONIC TECHNOLOGY, 2004.Voll.pp.54-60
- [6] Bian Hai-long, Jia Shao-hua; The Design and Development of USB2.0 Device. Posts & Telecom Press, 2004.

Author Biography

Wang Quan received his BS and MS in Computer Science from Xidian University at 1992 and 1997 respectively .Now he pursues his PhD in Computer Architecture from Xidian University. His work has focused on image process, computer graphics and the design and development of computer peripheral.