Firmware Design of the USB3.0 Data Transfer Interface

Jie Liu^{1,a}, Weihua Chen^{2,b}, Jietao Diao^{1,c} and Zhaolin Sun^{1,d}

¹School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China

²Meteorological and Hydrological Center of Nanjing Military Region, Nanjing 210016, China

 $\label{eq:com} \ensuremath{^{a}1031815609@qq.com}, \ensuremath{^{b}chenweihua@163.com}, \ensuremath{^{c}jietaodiao@nudt.edu.cn}, \ensuremath{^{d}sunzhaolin@nudt.edu.cn}, \ensuremath{^{d}sunzhaolin@nu$

n

Keywords: firmware design, USB3.0, transfer speed, EZ-USB FX3.

Abstract. USB (Universal Serial Bus) as a computer peripheral bus standards, is a flexible and efficient bus interface. The new version of USB3.0 theoretical transfer speed is up to 5.0Gbit/s, which greatly increases the data transfer bandwidth while retaining its inherent advantages. It's very suitable for high-speed data acquisition field. But in the actual use of that, data transfer speed is limited that USB devices can't meet the actual requirements. In this paper, we designed the firmware of the controller to give full play to the characteristics of the chip by using GPIF II digital transmission mode and establishing an automatic DMA channel, thereby improving the actual data transfer rate. Test results show that the real effective transfer rate is up to 350 Mbyte/s, greatly reducing the transfer time.

1. Introduction

With the rapid development of electronic technology and computer technology, the rate of the data acquisition module is increasingly high and data transmission capacity requirements are also rising. The current data collection system is mainly using the PCI-E interface to transmit the collected data to the computer for data processing and storage in real time. But there are some shortcomings of high costs, complex operation and poor portability in this way. These problems can be solved with the new generation of USB3.0 data transfer interface. The theoretical transfer speeds of USB3.0 is up to 5.0Gbit/s, which greatly improves the data transfer bandwidth with inherent advantages of USB1.0, USB1.1 and USB2.0.It's very suitable for high speed data acquisition.

Many chip manufacturers have produced chips that support USB3.0 protocol since the protocol proposed. EZ-USB FX3 (CYUSB3014) produced by the company called cypress is currently the most widely used chip for USB devices. The chip provides powerful interface design patterns, including the I/O mode, Slave FIFO mode and GPIF II data transmission mode. Designers can complete the high-speed data transmission for data acquisition devices by GPIF II mode.

The firmware of the USB3.0 chip is the key of affecting the actual transfer rate of USB3.0 interfaces. In this paper, we introduced the firmware design method of the USB3.0 data transfer interface and designed the firmware through the use of GPIF II data transmission mode and automatic DMA data transmission channel to improve the chip resource utilization, thereby increasing the actual transfer rate.

2. Architecture of the Hardware

The experimental hardware uses the Super-Speed Explorer Kit with EZ-USB FX3. EZ-USB FX3 is Cypress's USB3.0 peripheral controller with high-bandwidth that provides integrated and flexible features. As is shown in Fig. 1, FX3 integrates the USB 3.0 and USB 2.0 physical layers (PHYs) along with a 32-bit ARM 926EJ-S microprocessor for powerful data processing and for building custom USB Super-Speed applications[1].

To provide high-bandwidth access to USB 3.0 data, FX3 contains a hardware unit called General Programmable Interface, Generation 2 (GPIF II). GPIF II is an enhanced version of the GPIF in

FX2LP, Cypress's USB 2.0 product. GPIF II provides easy and glueless connectivity to popular interfaces such as asynchronous SRAM and asynchronous and synchronous address and data multiplexed interfaces[2]. FX3 implements a DMA-centric architecture that enables direct 375-MBps data transfer from GPIF II to the USB interface without CPU intervention.



Fig. 1 FX3 block diagram

An integrated USB 2.0 USB On-The-Go (OTG) controller enables applications in which FX3 may serve dual high-speed roles; for example, EZ-USB FX3 may function as a High-Speed On-The-Go (HS-OTG) host to USB Mass Storage Class (MSC) devices and HID-class devices. FX3 contains 512 KB or 256 KB of on-chip SRAM for code and data. EZ-USB FX3 also provides interfaces to connect to serial peripherals such as UART, SPI, I2C, and I2S[1-3]. FX3 comes with application development tools. The software development kit (SDK) provides application examples for accelerating the time to market.FX3 complies with the USB 3.0 v1.0 specification and is also backward compatible with USB 2.0. It also complies with the Battery Charging Specification v1.1 and USB 2.0 OTG Specification v2.0.

3. Firmware Design

The firmware is loaded by other devices to the USB controller to achieve the data transfer function of the interface after the USB controller chip is powered up. EZ-USB FX3 firmware design doesn't require too much attention to the underlying hardware settings for designers, but simply calls the appropriate library functions, which reduces the difficulty of the firmware development .As is shown in Fig. 2,EZ-USB FX3 firmware design is mainly related to the GPIF II interface design and DMA channels design. Control information, status information, and data information could exchange with the hardware through the general programmable interface called GPIF II.GPIF II interface allows super-speed data transfer between the interface chip and FPGA, due to the high frequency of the GPIF II interface. At the heart of the FX3 is a sophisticated, distributed DMA controller that is capable of moving data at 800-MBps that allows high performance data transfers between memories and peripherals without CPU intervention.



Fig. 2 Block diagram of the firmware design

3.1 GPIF II Interface

EZ-USB FX3 integrates a high-performance interface named GPIF II, which enables functionality similar to but more advanced than the FX2LP GPIF and Slave FIFO interfaces. GPIF II is a programmable state machine that provides the flexibility to design a variety of interfaces to outside entities. The GPIF II interface may function either as a master or a slave in industry-standard or proprietary interfaces and it enables interface frequencies to 100M Hz. GPIF II can implement both parallel and serial high-bandwidth interfaces, that the number of bits of the data bus is flexible to choose between 8 bit, 16 bit and 32 bit. It supports 14 configurable control pins(strobes, enable, GPIO)when a 32-bit data bus is used and 16 configurable control pins when a 8-bit,16-bit,or 24-bit data bus is used. All control pins can be input, output or bidirectional.



Fig. 3 Signals of the GPIF II interface

As shown in Fig. 3, the design of the GPIF II interface is configured as a 32-bit synchronous FIFO, and works in slave mode, with an external clock signal PCLK as a working clock of the state machine. External devices judge that the FIFO buffer is empty or full by detecting the flag signal FLAGA and FLAGB. In terms of external devices, the USB3.0 device controller acts like the storage area, which could freely read from or write to it.

The GPIF II interface is configured by creating a GPIF II state machine, and 8 KB of memory space is allocated to store the GPIF II state machine definition. Each state is defined by 32 bytes in (SRAM) memory. These 32 bytes define the properties of a state and the trigger conditions that can cause state (or I/O) transitions. Each state has two transitions out of it. The transition out of a state is determined based on transition conditions. Transitions are caused by both external and internal triggers. Each state is programmed to perform certain actions. The transition conditions are checked on each GPIF II clock edge or after a programmable number of clock cycles.

3.2 FX3 DAM Subsystem

EZ-USB FX3 device architecture includes a DMA fabric that is used to route data between various peripheral interfaces and/or the system memory of the device.



Fig. 4 Block diagram of FX3 DMA subsystem

The Advanced Microcontroller Bus Architecture - Advanced High Performance Bus (AMBA AHB) interconnect forms the central nervous system of FX3[3]. Fig. 4 shows how the CPU accesses the system memory using the System AHB. All peripheral DMA paths connect to the DMA AHB. Bridges between the System bus and the DMA bus are essential in routing the DMA traffic through the system memory. The width of a peripheral connection to the AHB determines its throughput. The peripheral core implements the actual logic of the peripheral (I2C, GPIF, and USB).

The FX3 DMA subsystem runs on an internal DMA bus clock, dma_bus_clk_i, that is divided down from the CPU clock. DMA descriptors are DMA instructions in a set of registers allocated in the FX3 RAM. A DMA descriptor holds information about the address and size of the DMA buffer as well as pointers to the next DMA Descriptor. These pointers create DMA descriptor chains. Descriptors enable the synchronization between sockets. DMA buffers are data buffers allocated in the system memory used for DMA. They can be of any size within the memory region and byte aligned. However, if the ARM data cache is enabled, it requires that the full buffer must be 32-byte aligned and of a size that is a multiple of 32 bytes. A socket is the unidirectional virtual port (gateway) used by a peripheral (IP) block to transfer data to/from the system SRAM. An FX3 DMA capable peripheral has multiple sockets in the DMA adapter. The number of sockets and their properties depend on the specific DMA adapter to the peripheral. Each peripheral block (IP block) in the device can support a predefined number of sockets which is the maximum number of independent data flows that can be done through that IP at a given point of time.

4. Results and Discussion

The firmware test system is shown in Fig. 5.It consists of the excitation source, the USB3.0 controller and the host. The excitation source generates excitation data and then transmits them to the controller by the GPIF II interface. The controller forwards data to the host according to the USB3.0 protocol. We verified the firmware with the correctness of the design mainly through the firmware enumeration test and the data transmission test.



Fig. 5 The architecture of the firmware test system

We tested the firmware enumeration functions by using the host test software named USB Control Center. As shown in Fig. 6, various descriptors sent by the firmware have been successfully accepted by the test software. It indicates that the device controllers comply with the USB3.0 protocol and could be capable of super-speed data transmission.



Fig. 6 Firmware enumeration test

We tested the actual data transmission speed through the software of Streamer provided by Cypress. The configuration in Streamer was in accordance with the rules in the firmware, which has been set to the block transfer mode. As shown in Fig. 7, Streamer shows the test speed. It reached 350 Mbyte/s.

(OxO4B4 - OxOOF1) Cypress USB StreamerExample				T
BULK IN,	10	5384 Bytes,15 M	axBurst, (O -	0x81) 💌
256	Ŧ	Successes		192
64	y	Failures		(
	1500		Stop	
	0x04B4 - BULK IN, 256 64	(0x04B4 - 0x00F1) BULK IN, 10 256 - 1 64 - 1 1500	(0x04B4 - 0x00F1) Cypress USB S BULK IN, 16384 Bytes, 15 M 256 Successes 64 Failures 1500 1500	(0x04B4 - 0x00F1) Cypress USB StreamerExample BULK IN, 16384 Bytes, 15 MaxBurst, (0 - 256 Successes

Fig. 7 Data transfer rate test

5. Conclusion

In this paper, we described the hardware of the EZ-USB FX3 and the main mechanism of the firmware. We designed the firmware of the controller to give full play to the characteristics of the chip by using GPIF II digital transmission mode and establishing an automatic DMA channel, thereby improving the actual data transfer rate. After a lot of practical application and testing, the firmware is stable and the speed can be up to 350Mbyte/s. This design has strong practical value and also has been used in project. Nevertheless, the firmware in this design is still having a big gap between the theoretical value. It is necessary to do further research on this, so that the USB3.0 interface could have a higher actual transfer rate.

References

[1] Cypress Semiconductor, CYUSB3014, EZ-USB® FX3 Super-Speed USB Controller, Document Number 001-52136 Rev. *K.

[2] Cypress Semiconductor, AN65974, Designing with the EZ-USB®FX3TM Slave FIFO Interface, Document No. 001-65974 Rev. *I.

[3] Cypress Semiconductor, AN75705, Getting Started with EZ-USB®FX3TM, Document No. 001-75705 Rev. *C.

[4] Terzopoulos N, Laoudias C, Plessas F, et al. A 5 - Gbps USB3.0 transmitter and receiver linear equalizer[J]. International Journal of Circuit Theory & Applications, 2015.

[5] Rügamer A, Frank Förster, Manuel Stahl, et al. Features and Applications of the Adaptable Flexiband USB3.0 Front-end[J]. Navigationssystem, 2014.

[6] Long B Q, Zeng Z M. High Speed Data Transfer System Based on the USB3.0 Technology[J]. Applied Mechanics & Materials, 2014, 614:381-384.

[7] Nigesh B , Kukal T S, Prakriya S. Modeling, design and automation of 5 Gbps serial link transceiver with jitter cancellation[C]// Communications (NCC), 2013 National Conference on. IEEE, 2013:1 - 5.

[8] Lin M S, Tsai C C, Chang C H, et al. A 5Gb/s low-power PCI express/USB3.0 ready PHY in 40nm CMOS technology with high-jitter immunity[C]// Solid-State Circuits Conference, 2009. A-SSCC 2009. IEEE Asian. IEEE, 2009:177-180.