

# USB 设备控制器的硬件实现与研究

沈小磊, 张晓彤, 李占才

(北京科技大学信息工程学院, 北京 100083)

**摘要:** 在研究通用串行总线(USB)协议的基础上, 建立了数据采样的数学模型, 并基于该数学模型提出了数字锁相环的一种简单可靠的实现方法。在给定的 USB 主机和设备周期差异范围内, 该 USB 设备能够正确传输。在保证带宽利用率的基础上, 设计了缓冲区的实现方案, 并将该 USB 设备控制器应用在基于龙芯的 SoC 芯片上, 用 FPGA 进行了验证。

**关键词:** 通用串行总线(USB); 数字锁相环; 双端口 RAM

## Hardware Implementation and Study of USB Device Controller

SHEN Xiao-lei, ZHANG Xiao-tong, LI Zhan-cai

(Information and Engineering School, University of Science and Technology Beijing, Beijing 100083)

**【Abstract】** On the basis of USB transport protocol research, this paper establishes the mathematical model of data sampling and presents a design method of digital phase locked logic(DPLL) which is simple and reliable. In the given cycle difference range of USB host and device, the USB device controller transmits accurately. On the basis of band utility, this paper proposes the buffer implementation scheme. The USB device controller is embedded in the SoC chip based on Godson CPU and is validated by Field Programmable Gate Array(FPGA).

**【Key words】** Universal Serial Bus(USB); Digital Phase Locked Logic(DPLL); dual-port RAM

通用串行总线(Universal Serial Bus, USB)是一种快速灵活的总线接口, 它的传输速度是标准串口的数十倍, 而且可以支持多达 127 个外设同时级联<sup>[1]</sup>。USB总线上传输的只有数据信号, 没有时钟信号, 数据的接收端要通过锁相环来产生本地采样时钟, USB1.1 中数据的传输速率不高(12 Mb/s), 传统的锁相环的实现方法输入时钟和输出时钟频率差异较大( $F_{in}=M \times F_{out}$ ,  $M=2^n$ ,  $n>3$ ), 实现也稍显复杂<sup>[2]</sup>。本文遵循简单、可靠和易于扩展的原则, 综合考虑带宽利用率和缓冲区大小的关系, 开发出了符合USB1.1 协议的设备控制器IP。

### 1 USB 设备控制器的功能结构

图 1 为本文所述 USB 设备控制器的功能结构图。

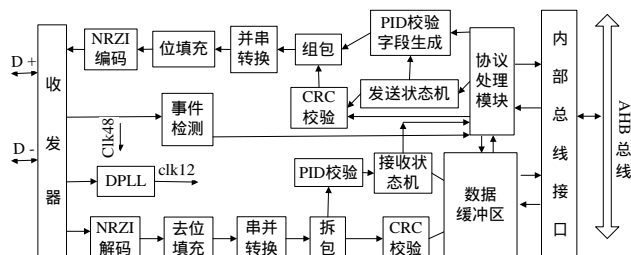


图 1 USB 设备控制器功能结构

在接收数据时, 首先对包标识符(PID)进行解析, 如果是标记包, 则用 5 位CRC进行校验, 如果是数据包, 则用 16 位CRC校验, 如果校验错误, 则丢弃该信息包<sup>[1]</sup>, 校验正确后将标记包送入USB协议处理模块, 针对不同的PID进行相应的处理; 如果是握手包, 由于只包含PID字段, 因此只进行PID校验, 将解析后的PID送入USB协议处理模块。在发送数据时, 如果是数据包, 首先从缓存控制器模块得到所需数据, 然后生成 16 位CRC校验码和相应数据包的PID, 组包后进行

发送; 如果是握手包, 则直接生成相应握手包(ACK, NAK, STALL)的PID, 经过并串转换、位填充和NRZI编码后, 送到收发器PDIUSB11A<sup>[3]</sup>发送。

### 2 数字锁相环的实现

在 USB1.1 协议中规定全速传输为 12 Mb/s, 即串行数据以 12 Mb/s 的速度在主机和设备之间传输, 因此在数据的接收端可以采用 48 MHz 的时钟采样串行数据, 当主机时钟和本地时钟(48 MHz)四分频严格一致的情况下每一位数据可以被采样 4 次, 由此, 可以将每一位数据分成 4 个部分, 即 Start, Mid1, Mid2, End。

图 2 中 clk48 为本地 48 MHz 采样时钟, 其周期为  $T_r$  (20.83 ns)。

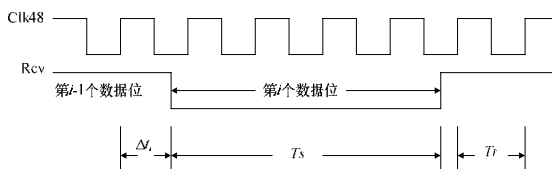


图 2 周期关系

rcv为串行输入数据。  $t_i$  为第  $i$  个数据到前一个时钟上升沿的相位差异。  $T_s$  为数据的发送周期。可以得到第  $i$  个数据被 clk48 采样的次数为

$$S_i = \text{INT}[(t_i + T_s) / T_r] \quad (1)$$

$$T = T_s - 4T_r \quad (2)$$

**基金项目:** 北京市科技计划基金资助重大项目(京科技发[2002]188号)

**作者简介:** 沈小磊(1982 - ), 男, 硕士研究生, 主研方向: 计算机系统结构, 集成电路设计; 张晓彤、李占才, 副教授

**收稿日期:** 2007-01-21 **E-mail:** sincerely\_0986@sohu.com



利用率和端点的应用背景得到的。通过以下的公式可以得到各种传输方式可达到的最大带宽和缓冲区空间的关系。

$$BW=EP*INT(BPF/(EP+OH))/125 \quad (7)$$

其中,  $BPF$  表示在一个时间片内可传输的字节数, 对应 USB1.1 其值为:  $12\ 000/8=1\ 500$ ;  $BW$  表示 USB 的实际传输率;  $EP$  表示端点的数据分组的大小;  $OH$  表示传输时的协议开销。

利用式(7)可得到在各种传输方式下 USB 的带宽利用率如表 1 所示。

表 1 包长度和带宽利用率关系

传输方式	协议开销(OH) /B	数据净长度(EP) /B	实际最大带宽 /(Mb·s <sup>-1</sup> )	带宽利用率 /(%)
控制传输	46	16	3.072	25.6
		32	4.864	40.53
		64	6.656	55.47
批量传输	13	16	6.528	54.4
		32	8.448	70.4
		64	9.728	81.07
中断传输	13	8	4.544	37.87
		16	6.528	54.4
		32	8.448	70.4
实时传输	9	64	10.24	85.33
		128	10.24	85.33
		512	8.192	68.27

由表 1 可知, 对于控制传输数据, 净长度为 64 B 时可以达到最大带宽利用率, 但是考虑到控制传输只发生在数据传输的开始阶段, 因此, 在本文中控制传输数据净长度采用 32 B; 对于批量传输数据, 净长度为 64 B 时可以达到最大带宽利用率; 由于中断传输用在小规模的数据传输上, 数据传输量不大, 中断传输的数据净长度采用 8 B; 对于实时传输数据净长度为 64 B 和 128 B 时都可以达到最大带宽利用率, 为了节省缓存, 实时传输的数据净长度在本文中采用 64 B。

由于 USB 规范中规定缓冲区必须有足够的空间为全速设备容纳一个时间片所能传输数据量 2 倍的容量。这样 USB 在处理一个数据分组的同时, 可以接收下一个数据分组。由于每种传输方式都有 IN 和 OUT 两个过程, 因此控制传输需要  $32 \times 2 = 128$  B 的缓冲区, 同理, 批量传输和实时传输各需要 256 B 的缓冲区; 中断传输需要 32 B 的缓冲区。本文中缓冲区的设计采用双端口 RAM, 一个端口为 8 位, 负责接收 8 位输入数据, 一个端口为 32 位, 用于将数据发送到 AHB 总线。缓冲区大小为 1 024 B, 除去上述 4 种传输方式需要的 672 B 的空间, 剩余的 352 B 的空间用于存放 USB 设备控制器的设备描述符。

## 5 仿真与验证

由于本 USB 设备控制器集成到数字电视 SoC 芯片, 因此该芯片采用龙芯一号处理器。整个系统的验证平台如图 5 所示。将 USB 设备控制器、龙芯一号、RAM 控制器、DMA 控制器、UART 及 AMBA 总线下载到 Xilinx 的 XC2VP70 中, 同时外接 RAM、FLASH 和收发器。PC 将编译好的指令通过串口写入 FLASH 中, 龙芯处理器执行 FLASH 中的指令, 主要包括配置 USB 设备控制器的寄存器, 向数据缓冲区中写入

数据和设备描述符等。收发器接入 PC 主机的 USB 根集线器, 在 PC 上运行设备驱动程序, 对 USB 设备控制器的各种传输方式进行测试。龙芯处理器将接收到的数据存入 RAM 中, 并将 RAM 中的数据通过串口发送到 PC, PC 端即可检验发送的数据是否正确。在此测试平台上对该 USB 设备控制器的 4 种传输方式及支持的标准命令进行了大量的测试, 验证了该 USB 设备控制器的正确性。

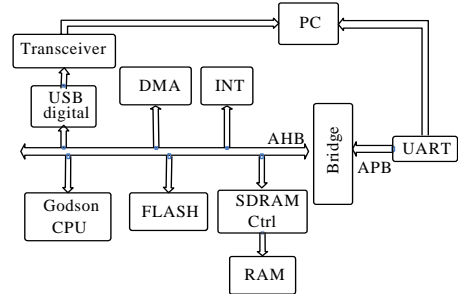


图 5 USB 验证平台

相对传统数字锁相环<sup>[2]</sup>, 本文数字锁相环的综合结果如表 2 所示。将该 USB 设备控制器使用 Quartus 软件综合后下载到 Altera 的 FPGA APEX20KE: EP20K160EFC484-3 后, 相对 USB 设备控制器 IYOYOYO<sup>[4]</sup>, 其规模如表 3 所示。

表 2 锁相环综合结果

	LUT	FLIP FLOPs	Max frequency/MHz
本文锁相环	15	16	389
传统锁相环	87	26	120

表 3 USB 设备控制器综合结果

设备	USB 标准命令	FIFO 大小/B	规模(LE)
USB 设备控制器	支持 9 种 USB 标准命令	1 024	1 558/6 400(24%)
IYOYOYO	USB 标准命令交给 MCU 处理	2 168	1 645/6 400(25%)

## 6 结论

本文实现了一个 USB 设备控制器, 并重点分析了数字锁相环的设计, 有效数据包长度的选择及协议处理模块的实现。该 USB 设备控制器遵循 USB1.1 协议, 随着多媒体应用对数据传输速度的需求越来越高, USB1.1 协议 12 Mb/s 的传输速率必然会成为系统传输的瓶颈, 因此, 向 USB2.0 协议扩展及如何设计可扩展 IP 是本文下一步要研究的内容。

## 参考文献

- [1] Implements Form. Universal Serial Bus Specification1[Z]. (1998-10-18). <http://www.USB.org>.
- [2] 谢程宏. 全数字锁相环的设计[EB/OL]. (2003-04-02). <http://www.eaw.com.cn/>.
- [3] PHILIPS. PDIUSB11A(Universal Serial Bus Transceiver)[Z]. [2006-11-11]. <http://www.alldatasheet.com/>.
- [4] Kouyama T, Nano H, Kon C. The Design of a USB Device Controller IYOYOYO[C]//Proc. of EDA Technofair Design Automation Conference. Beijing: [s. n.], 2003: 573-574.