

基于 SOPC 的 USB 设备控制器设计^①

The Design of USB Device Controller Based on SOPC

戴迎璐 (浙江万里学院 基础学院 浙江宁波 315100)

摘要: 可编程片上系统将处理器、存储器、I/O 接口等系统设计必需的功能模块集成到一个可编程逻辑器件上。其优点是具有灵活的设计方式、可裁减、可扩充、可升级,并具备软件硬件在系统可编程的功能。详细给出了基于 Altera 公司的 SOPC 器件完成 USB 设备控制器的设计方法和过程。首先是对 USB 控制器 IP 核功能模块的划分,然后是分模块进行具体的设计。重点讲述了串行接口引擎控制状态机和端点控制器的设计。最后,在实现 USB 设备控制器的基础上,对整个设计进行仿真、验证和综合。

关键词: 通用串行总线 设备控制器 可编程片上系统 串行接口引擎 端点控制器

1 引言

目前 USB 设备端控制器的实现主要有两种方式:一是全集成芯片,如专用的微处理器(MCU)内嵌符合 USB 规范的 SIE(Serial Interface Engine),此种方式在单芯片内集成 USB 设备控制核和基于该核的单片机,既可以节省芯片资源、也可以方便特定单片机设计。虽然降低了设计成本,但制作专用集成电路(ASIC)的周期长,也不利于产品的升级。二是部分集成芯片,如分立的通用微处理器 + SIE。将 USB 设备控制器与专用单片机组合,虽然开发起来周期短,但集成度低。由于需要专用的 USB 设备控制器,而且 USB 设备控制器作为一个“黑匣子”调试起来也比较麻烦。总而言之,两种方法各有优缺点。

随着电路规模越来越大,片上系统(SoC)已经成为 IC 设计的发展趋势,相应地也有了更加灵活的片上可编程系统(SOPC)。SOPC 将处理器、存储器、I/O 接口等系统设计必需的功能模块集成到一个可编程逻辑器件上,构建成一个可编程的片上系统。具有灵活的设计方式,可裁减、可扩充、可升级,并具备软件硬件在系统可编程的功能。采用 SOPC 进行 USB 控制器的设计可以兼顾以上两种设计方法的长处,即不需要自己设计或采用专用的单片机,又能提高设计的集成度。同时也能兼顾开发周期和开发成本。

本文在介绍 SOPC 设计流程的基础上,详细介绍

了基于 Altera 公司的 SOPC 器件设计 USB 设备控制器的方法。首先是对 USB 控制器 IP 核功能模块的划分,然后是分模块进行具体的设计。文中重点讲述了串行接口引擎 SIE 控制状态机和端点控制器的设计。并在此基础上进行了仿真证明。最后给出了最终实现 USB 控制器的技术指标。

2 SOPC 设计流程

通常来说,EDA 厂商及第三方提供的 IP 核都是通用的,但进行 SOPC 设计时,用户可能有特殊需求,这时必须使用用户自定义逻辑来实现所要的功能。SOPC Builder 集成在 Altera 公司的 EDA 工具 QuartusII 中。用户可以通过 SOPC Builder 提供的图形用户界面从 Altera 公司提供的 IP 库中选取一些组件,如 NiosII, DMA, SRAM, FLASH 等等,根据实际需要设置这些 IP 的配置参数。用户还可以自己编写 HDL 代码模块作为用户自定义逻辑(或称为 IP 核)方便地添加到 SOPC Builder 中,设计流程如图 1 所示。

3 USB 设备控制器的实现

3.1 设备控制器简介

USB 是一种自成体系的通信协议与体系结构,USB

① 基金项目:宁波市自然科学基金项目(2005A620001)

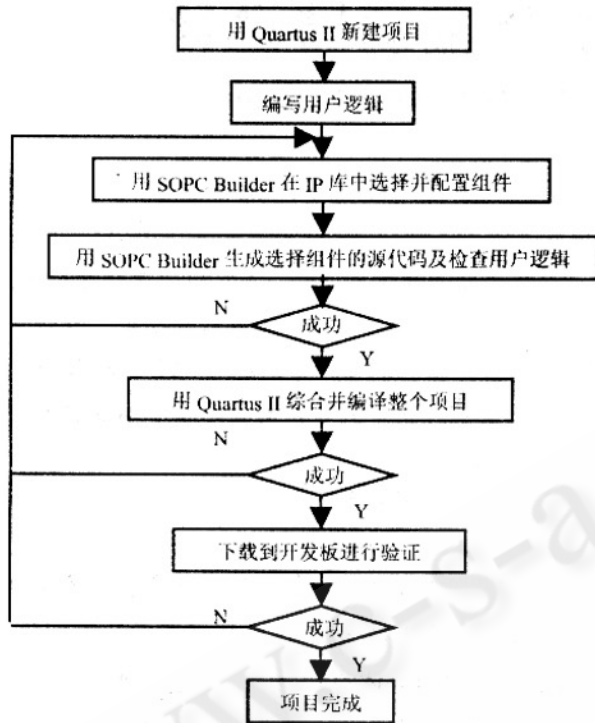


图 1 SOPC 设计流程
Fig. 1 SOPC Design Procedure

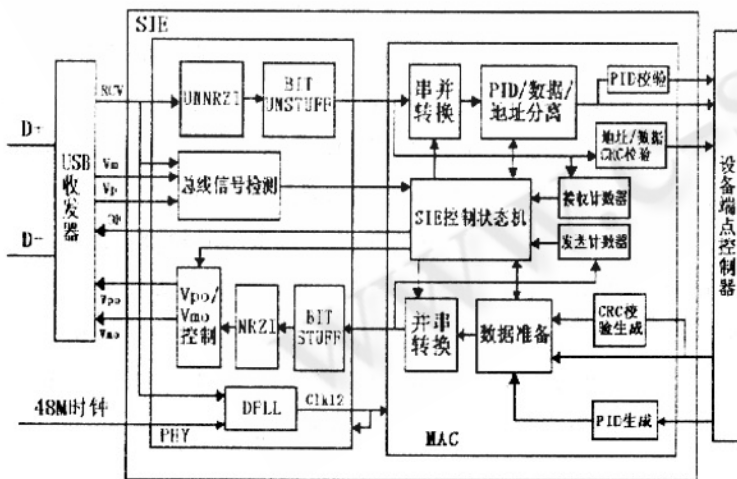


图 2 USB 设备控制器模块划分示意图
Fig. 2 Divide Diagram of the USB Device Controller

设备控制器处于主机(或者集线器)与一般设备之间,

负责将 USB 电缆差分信号串并转换且摒除 USB 格式后向一般设备发送,同时可以将一般设备的数据按 USB 格式发送给主机。USB 设备控制主要分四部分,即电压调整器、收发器 (transceiver)、锁相环 (Phase Locked Loop, 简称 PLL) 和串行接口引擎 (Serial Interface Engine, 简称 SIE)。电压调整器将外部电压或自供电压调整成稳定的 3.3 伏。收发器直接与 USB 电缆相连,自带上拉电阻,协调转换两条差分数据线双向输出。存储器管理负责调用存储器,自带的存储器一般较小,只是用于暂时缓冲数据。SIE 即串行接口引擎,是整个 USB 设备控制器的核心,用来屏蔽数据 USB 格式,完成串并转换。

3.2 USB IP 核模块划分及详细设计

3.2.1 模块划分

根据对 USB 设备架构和设备控制器的功能分析,本文对 USB 设备控制器进行模块划分,如图 2 所示,并将详细讨论 USB 设备控制器具体模块的设计。

本文将 USB 设备控制器主要分为两大部分:串行接口引擎 (SIE) 和端点控制器 (Endpoint Controller)。SIE 主要提供设备控制器与 USB 收发器的接口以及把检测无误的信息流传送给后续端点控制器模块。端点控制器完成端点缓冲区的管理以及与主机的通信工作。串行接口引擎又可分为两大部分:物理接口 (PHY) 和媒体接口 (MAC)。由于篇幅关系,以下重点分析 SIE 控制状态机和端点控制器的设计。

3.2.2 SIE 控制状态机模块的设计

SIE 控制状态机的功能是控制 SIE 接收和发送数据的过程,为其它 SIE 模块和端点控制器提供控制信号。它由接收状态机和发送状态机组成。

(1) 接收状态机

接收状态机用于 USB 包的接收过程,状态机中的状态和状态转换条件是基于对 USB 包格式的分析而提出的,如图 3 所示。

如果总线检测模块检测到包开始的同步序列 SYNC,状态机进入 PID_REV 状态。此时,SIE 开始接收 USB 包的 PID 字段,接收计数器开始对接收到的数据比特进行计数。PID/SOF/数据/地址分离模块根据计数值从串并转换模块提取

PID,并将前四个比特存入寄存器 pid_rx。计数器计数值 Rx_cnt 等于 8 时,PID 字段接收完毕,PID 校验模块对接收到的 PID 进行校验,接收状态机根据寄存器 pid_rx 的值选择进入下一个状态。

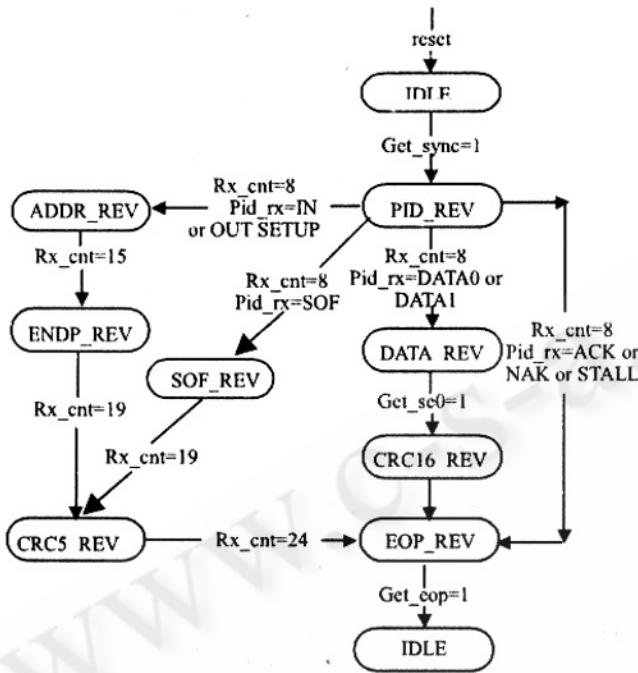


图 3 接收状态机

Fig. 3 Receive State Machine

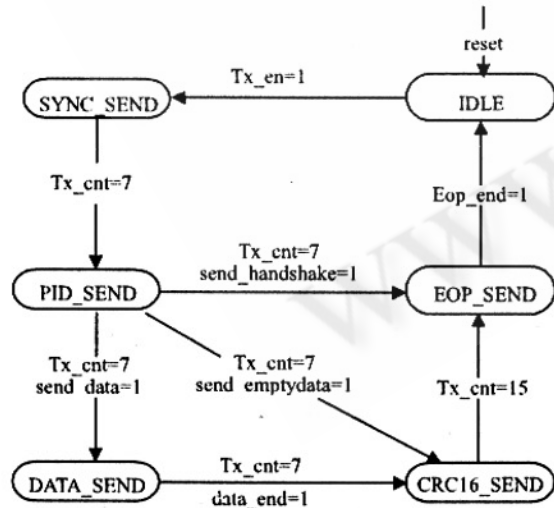


图 4 发送状态机

Fig. 4 Transmit State Machine

(2) 发送状态机

发送状态机用于 USB 包发送过程的控制。根据 USB 协议,USB 设备可能需要发送的包的种类有数据包和握手包。根据这两类包的格式,发送状态机设定了相应的 SYNC_SEND、PID_SEND、DATA_SEND、CRC16_SEND、EOP_SEND 状态,对应于包的每个字段,如图 4 所示。

状态机的启动,通过信号 Tx_en 来控制,Tx_en 由设备端点控制器提供,当其为 1 时,状态机进入 SYNC_SEND 状态,开始发送包的同步序列头 SYNC。计数器 Tx_cnt 的值为 7 时发送结束,进入 PID_SEND 状态,计数器复位。在 PID_SEND 状态时,PID 生成模块生成的 PID 被载入数据准备模块,等待发送。

3.2.3 端点控制器设计

USB 端点是 USB 内有地址标志的实体,它需要对来自主机的数据包做出响应。其结构简图如图 5 所示。

端点控制器中的数据分配器用于根据端点地址译码器的指令选通数据通路,并把数据输送到相应的端点缓冲区之内。对于控制传输,由端点地址比较器决定将数据通过数据分配器送到 EPO 的 FIFO 缓冲器内,端点 0 控制器自动从缓冲区内读取数据来识别标准设备请求。

对于块传输端点 EP2,芯片中设计了两个相同的 64 字节的 FIFO 缓冲器。每个缓冲器都有各自的计数。主机传送来的数据根据状态位由比较器决定送往哪一个缓冲区。中断传输端点 EPI,用于向主机反馈块传输端点缓冲区的状态信息。信息由端口缓冲器状态信息、外部用户控制信息等组成。中断传输每 1ms 时间内由主机轮询一次,在主机收到关于设备块传输端点的状态信息之后,由程序决定发送下一组数据包或者调整程序进程。

4 IP 核的仿真验证

我们使用 QuartusII 软件进行编译和仿真,产生 VCD 文件格式的波形文件。利用其良好的调试功能,对仿真波形进行分析、比较,定位设计中存在的问题。仿真结果表明,设备控制器可按照 USB 传输协议的规定,正确地接收到的分组数据进行解码和识别,并能够及时地回复数据,完成对功能设备的上电、总线枚

(上接第 72 页)

参考文献

- 1 MindShare, Inc, Don Anderson, Dave Dzatko 著, 孟文译, USB 系统体系(第二版)[M], 北京: 中国电力出版社, 2003.
- 2 王成儒、李英伟, USB 2.0 原理与工程开发[M], 北京: 国防工业出版社, 2004.
- 3 刘金凤、杨方, USB 接口设备的开发[J], 农机化研究, 2004, (1): 244 - 245.
- 4 陈源、裘正定, 基于嵌入式系统的 USB 接口的设计[J], 计算机工程与应用, 2004, (7): 110 - 112.
- 5 黄卫华、朱向东等, 一种高速 USB 设备控制器 IP 核的设计与实现[J], 微电子学与计算机, 2005, 22 (5): 106 - 108.
- 6 刘连东、郁滨、罗长远, 基于 NiosII 的嵌入式 USB 主机设计与研究[J], 计算机工程与设计, 2007, 28 (9): 2100 - 2102.