

# 基于 FPGA 的 USB 主控芯片 软硬件协同设计与验证 \*

梁瑞彬, 李晓潮, 郭东辉  
(厦门大学 电子工程系, 福建 厦门 361005)

**摘要:**设计了一款采用 PowerPC 架构的 USB1.1 主机控制器芯片, 并对该芯片进行软硬件协同验证。通过内嵌 PowerPC 和 USB 主机 IP 核的 FPGA 系统, 辅以外围收发器电路、驱动、应用程序和文件系统, 完成了对 U-Disk 和 HID 两类典型 USB 应用的测试, 验证结果表明该 USB 主机芯片设计可以符合 USB 技术规范, 并能和其他厂家的设备兼容。

**关键词:** USB 主机, SoC, 软硬件协同验证

中图分类号: TP334.4

文献标识码: A

## The hardware-software Co-design and verification for USB host SoC based on FPGA

LIANG Rui-bin, LI Xiao-chao, GUO Dong-hui

(Department of Electronic Engineering, Xiamen University, Xiamen Fujian 361005, China)

**Abstract:** We design a USB1.1 host controller SoC based on PowerPC architecture and make hardware-software co-verification. The SoC prototype is implemented with embedded PowerPC and the USB host IP core in FPGA, supplemented by an external USB transceiver and software application systems. We complete two typical USB applications testing, eg. U-Disk and HID. The results show that the USB host SoC meets the USB specifications and is compatible with other vendors' products.

**Keywords:** USB host, System on Chip, hardware-software co-verification

### 1 引言

近年来, USB 接口技术在消费性电子、通讯等领域得到了越来越广泛的应用<sup>[1]</sup>。USB 的主从架构决

定了设备之间必须通过主机才能进行数据交互<sup>[2]</sup>, 而目前扮演主机角色的通常是 PC 机, 这给 USB 设备之间的数据交互带来了诸多的不便。集成了 USB 主机功能 IP 的 SoC(片上系统)可以很好地解决这个问题。目前, 国外厂商已经推出了许多具有这种功能

\*基金项目: 厦门市科技计划项目(3502Z20093002), 福建省科技厅科技重点项目(2009H043)

的芯片，例如 CY7C67300、LPC175x 系列、LC87F1J2A 等。

国内与 USB 技术相关的研发主要集中在 USB 应用和系统设计上<sup>[3-5]</sup>。在 SoC 设计上也基本集中于 USB 设备控制芯片的研发<sup>[6-8]</sup>。对 USB 主机 SoC 设计研究还主要处于 IP 核的设计和单独验证，例如，文献[9]中完成了 USB1.1 主机控制器中串行接口引擎 (HSIE) 模块的设计；文献[10]中解决了 USB1.1 主机控制器 IP 核模块划分和实现等问题；文献[11]设计出了一种具备 USB2.0 主机控制器功能的 IP 核。文献[12]完成了具备全速 USB 主机功能的 USB2.0 OTG IP 核的研究和设计。然而，仅仅进行 IP 核的单独验证是不够的，还必须经过芯片级的验证和软硬件协同的验证<sup>[13]</sup>。其中，IP 核的单独验证和芯片级的验证只是针对系统的硬件，由于 USB 协议复杂，它不仅需要硬件支撑，还必须有大量的软件配合才能实现完整功能，因此，其软硬件协同验证是验证 SoC 设计正确与否的关键步骤。

对 USB 主机 SoC 的软硬件协同验证的实现方案有很多种，例如文献[14]采用 Seamless 的 CVE 仿真软件实现了对 USB 主机的软硬件协同验证。为了验证我们基于 PowerPC 架构的 USB1.1 主机 SoC 设计方案，本文基于 FPGA 实现了该 SoC 设计，并辅以必要的软硬件和调试工具，对其进行软硬件的协同验证。本文内容划分如下：第一节介绍该 SoC 的架构及验证环境，第二节阐述验证过程中的硬件实现，第三节介绍软件部分的设计，第四节进行实际的应用测试，最后一节对本文进行总结。

## 2 USB 主机 SoC 架构及其验证环境

基于 PowerPC 架构的 USB1.1 主机 SoC 的架构如图 1 所示。整个系统工作在两个严格的数字时钟域（总线时钟和 USB 时钟），尤其是 USB 时钟必须严格控制在 48 MHz，因此需要一个时钟管理模块对外部输入的 100 MHz 时钟进行频率合成和锁相，以生成系统需要的时钟频率。PowerPC405 微处理器作

为整个 SoC 的控制核心与处理器局部总线 (PLB) 相连，并提供 JTAG、电源管理、中断控制、片上数据 / 指令存储器 (DOCM/IOCM) 等接口，其上运行着 USB 主机的系统软件，因此需要在其数据 / 指令存储接口上连接两个片上存储单元 (OCM) 用于存储数据和指令。PLB 总线主要为处理器和高速设备提供低延时、高带宽通路。片上外围总线 (OPB) 面向片上外围接口 IP 核，用于连接具有不同总线带宽及时序要求的外设，其总线上可挂载包括 USB 主机控制器、中断管理、定时器和串口等 IP 核。由于 USB 主机 IP 核采用 wishbone 作为内联总线，必须在 wishbone 和 OPB 总线之间进行必要的桥接设计 (IPIF)。PowerPC405 仅提供两个外设中断接口，当外部中断多于两个时，需要添加一个中断管理模块管理外围 IP 核的中断。定时器用于系统必要的定时。串口作为输出 / 输入端口。此外，USB 主机 SoC 要能正常工作还需要相应的外部收发器支持，该收发器提供了 USB 的物理接口并对逻辑信号与物理端信号进行转换。

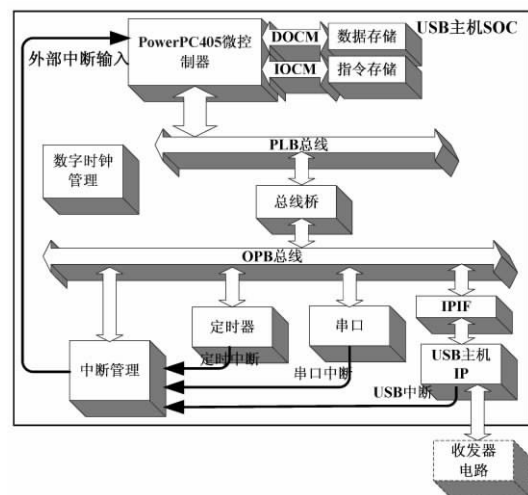


图 1 基于 PowerPC 架构的 USB 主机 SoC

我们构建的协同验证环境如图 2 所示。采用 Xilinx 公司 VIRTEX-II PRO 开发板上的 FPGA 芯片 XC2VP30 来实现该 USB 主机的 SoC。通过开发板的低速数字扩展口连接外部收发器子板。开发板上围绕 FPGA 的其他功能模块包括电源模块，时钟晶振等。PC 机将 USB 主机 SoC 的软硬件比特流文件通

过 JTAG 接口配置到 FPGA, 并通过串口和逻辑分析仪观察该 SoC 的运行情况。以下将对验证系统的软硬件设计做进一步的阐述。

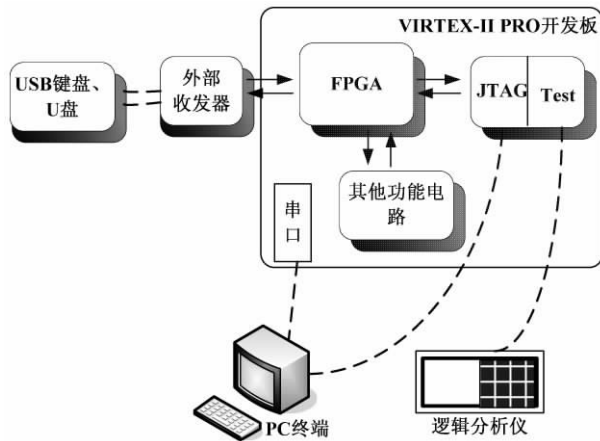


图 2 USB 主机 SoC 的协同验证环境

### 3 验证系统的硬件设计

硬件上的设计包含 FPGA 内部的 SoC 实现和收发器电路设计两个部分。我们在 EDK 开发平台中建立 SoC 系统的硬件，该平台提供了除 USB 主机 IP 核外的其他 IP 核，它们具有统一的接口标准，应用时仅需要进行必要的参数配置和接口连接，因此整个硬件设计的重点在于 USB 主机 IP 核的集成以及收发器电路的设计。

#### 3.1 USB 主机 IP 核集成

我们的 USB 主机 IP 核内部包含了 5 个模块（如图 3）。Wishbone 总线模块提供了与处理器的接口通道，内连 USB 协议核，数据接收缓存和发送缓存模块，USB 主机协议核是 USB1.1 主机协议的具体硬件实现，它主要由可寻址寄存器和相应处理逻辑组成，通过配置这些寄存器可以完成 USB 协议规定的四种传输类型（控制、批量、中断和同步传输）、直接控制总线状态、在全速状态下实现每 1 ms 自动发送 SOF 包及帧号计算等功能，接收数据缓存和发送数据缓存用来暂时存储主机端接收到的和待发送的数据；串行接口引擎模块的主要功能是检测收发

器状态，接收和发送总线信号，进行总线数据采样、串/并转换、同步插入、位填充、非归零反相编/解码等操作。

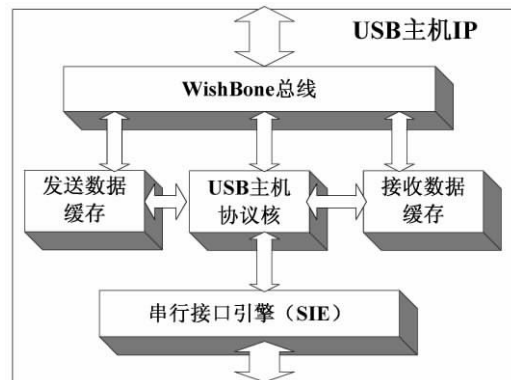


图 3 USB 主机 IP 核模块

为了将 USB 主机 IP 核集成到 SoC 系统中，必须解决 OPB 到 WishBone 总线之间的桥接问题，EDK 工具的 IPIF 服务<sup>[5]</sup>提供两种 OPB 到用户 IP 的桥接方式：寄存器模式和存储器模式。它们的最大区别在于读写方式不同。寄存器模式为每个寄存器单独提供读写使能，存储器模式则为一个连续的地址空间提供一个读写使能。为了满足 WishBone 接口信号时序，我们选用存储器模式桥接，接口信号如图 4。由于 USB 主机 IP 核地址空间为 256 字节，且数据宽度为一个字节，因此选择 IPIF 接口数据、地址总线的低 8 位连接到 WishBone 总线，并采用 Bus2IP\_ArCS 信号作为 USB 主机 IP 核的片选信号。另外扩展了 USB 时钟输入信号 (USB CLK) 和 USB 中断输出信号 (USB IRQ)。

#### 3.2 外部收发器电路的设计

外部收发器电路连接着 USB 主机 IP 核的串行接口引擎 (SIE) 外部逻辑端口，实现总线信号与数字逻辑信号之间的电平转换，满足全速和低速两种传输模式。SIE 外部逻辑端口包括：两对差分线 WireDataIn (VMI, VPI)，WireDataOut (VMO, VPO)，线输出有效信号 (OE) 和传输速率控制信号 (SPD)。这些逻辑端口工作在  $V_{cc}$  为 3.3 V 的 LVTTTL 电平模式下，其输出和输入的高/低电平必

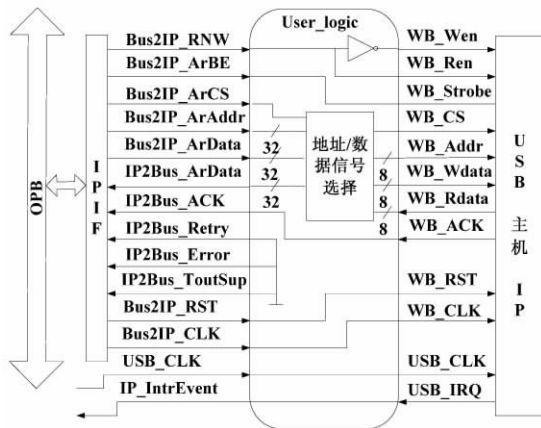


图4 接口桥模块

须满足  $VOH \geq 2.4V$ ,  $VOL \leq 0.4V$ ;  $V_{IH} \geq 2V$ ,  $V_{IL} \leq 0.8V$  的电平匹配。为此,我们选用了FAIRCHILD公司的USB1T11A作为外围收发器芯片,供电电压为3.3V。外部收发器的设计电路图如图5所示。

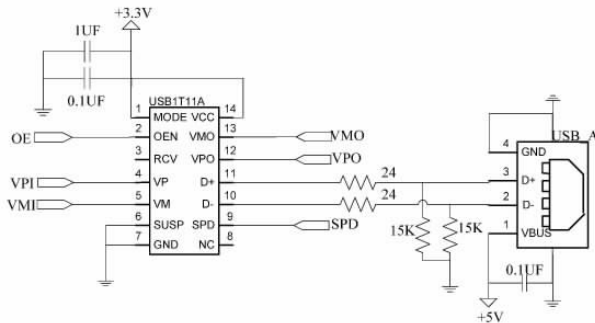


图5 主机收发器电路

电路中在差分总线D+,D-上分别串接一个24Ω电阻用于满足USB协议规定的总线串行阻抗要求。D+,D-线下拉15k电阻用来检测USB设备的插拔动作以及判断全/低速模式。当有设备接时,D+或D-的下拉15k电阻与设备端上拉的1.5k电阻构成回路,通过15k电阻分压使得其中一条数据总线处在高电平,从而通知主机有设备接入,D+为高时是高速设备,D-高时为低速设备,当设备移除后数据总线恢复默认的低电平状态。

## 4 验证系统的软件设计

我们参考USB协议将主机系统的软件分为USB系统软件和设备层软件两个层次,如图6所示,USB系统软件使用主机控制器硬件来管理主机与USB设备的数据传输,设备层软件建立在系统软件之上,描述直接与USB设备进行交互所需要的软件包。每个分层又包含两个小层,以下将结合具体的设备对各层的软件做进一步的阐述。

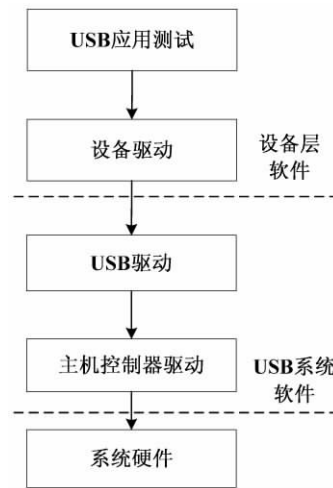


图6 USB主机SoC软件设计框图

### 4.1 USB系统软件

USB系统软件可分为主机控制器驱动和USB驱动两个模块,它们提供了基于不同抽象层级的软件函数。最底层的主机控制器驱动提供了对系统硬件可寻址寄存器的读/写函数,完成USB系统软件与硬件之间的映射。所有的IP核在总线上都分配有独立的地址空间,因此,IP核内可寻址寄存器的地址为IP核的总线基地址加上寄存器的偏移地址。USB驱动层通过调用底层的寄存器读写函数完成特定功能的函数抽象,包括IP核初始化、设备检测以及4种USB传输实现(控制,中断,批量和同步传输)等。IP核初始化函数完成对系统IP核的复位过程和设置系统的中断使能。在寄存器中,主机控制寄存器的最低位用于复位IP核。将其置1并等待10个USB时钟以上(大于200ns),既可完成IP核的复位过程,完成后该位自动清零。中断设置包括主机控制器IP核内的中断使能设置(传输完成中断

和连接事件中断使能),同时还包括系统中的定时器中断、串口中断以及中断管理单元的使能及优先级设置。设备检测函数通过读取反应 USB 总线连接状态寄存器的值判别设备的插拔,总线连接状态寄存器的低两位反应了总线的 3 种状态(00:不连接,01:低速,10:全速),当有插拔动作发生时,设备检测函数就读取该寄存器并根据所读取的数值设置主机端总线传输速率及极性。USB 的 4 种传输基于 3 种事务传输实现(IN,OUT,SETUP),例如控制传输的实现由一个 SETUP 事务,若干 IN 事务和一个 OUT 事务构成。这 3 个事务都必须由 USB 主机发起,SETUP 事务用于主机向设备发起各种描述符请求,首先将相应的描述符请求包送入发送数据缓存,然后设置设备的地址和端点号,最后设置发送模式并启动,对于全速设备,事务跟随在帧开始包(SOF)之后,低速设备事务直接发送。IN 事务用于接收从设备过来的数据,在设置完设备的地址、端点号和发送模式后启动 IN 事务等待事务传输完成中断,然后读取接收数据缓存中设备发送过来的数据。OUT 事务的工作过程和 SETUP 事务相似,区别在于 OUT 事务所发送的数据量没有固定,当数据量大于发送数据缓存时,必须分多次事务传输。同时在设备端 OUT 事务的端点 BUF(OUTDATA0, OUTDATA1)轮流工作,因此,程序中必须指明当前发送的 OUT 事务的端点 BUF。此外,USB 驱动层还定义了设备描述请求结构体和相应的设备描述符结构体等信息。

## 4.2 设备层软件

设备层的软件描述的是与具体设备相关的软件实体,它分为两个层级:设备驱动和应用测试,设备驱动层提供了针对不同设备应用的接口通道;应用测试则是针对具体设备的应用软件。以 U 盘设备为例,对 U 盘的操作可分为枚举过程和 SCSI 命令操作过程两个部分,因此设备驱动层主要实现 U 盘的枚举和 SCSI 命令。枚举过程通过调用 USB 驱动层的控制传输函数来完成,首先获取 U 盘端点 0 的最

大容量(Max\_Packet\_Size),然后给 U 盘分配一个新地址,接着获取 U 盘全面的设备描述符,根据设备描述符中描述的可用配置数量,获取 U 盘的配置描述符集合(包含接口和端点信息),最后进行 U 盘的配置。U 盘 SCSI 命令操作基于 USB 驱动层的批量传输函数完成,包括 READ(10),WRITE(10),INQUIRY,TEST UNIT READY 等命令。其中,READ(10)、WRITE(10)用于读写 U 盘数据,其它命令用于查询 U 盘的状态信息。为了进一步完成主机对 U 盘的应用测试,我们在应用测试层移植了开源的兼容 FAT12/16/32 的文件系统(FatFs),它提供了调用设备驱动层 SCSI 命令的函数接口,这些函数通过调用 READ(10)、WRITE(10)来完成对 U 盘设备的数据操作。文件系统还向上提供了一系列针对文件操作的应用接口(API),如创建文件夹(f\_mkdir),读目录清单(f\_readdir),打开/创建文件(f\_open),读、写文件(f\_read、f\_write),删除文件/目录(f\_unlink)等。我们通过串口接收定制命令选择相应的文件系统 API 函数完成对 U 盘的文件操作。对 U 盘的应用测试程序设计为首先枚举 U 盘,然后程序进入空闲状态等待接收串口终端的控制命令。当有命令输入时,程序从空闲状态进入数据接收状态。在接收完数据后,程序进入到文件操作状态,完成对所接收数据的译码并执行相应的文件操作函数。最后程序重新进入到空闲状态等待下一次接收串口数据。

## 5 验证结果及分析

我们构建的协同验证环境实物图如图 7 所示,和图 1 匹配。通过连接 U 盘、键盘等测试设备,并通过示波器和串口终端进行数据分析。

图 8(a)为示波器在 SIE 接口上观察到的一个 SETUP 事务的传输过程,和 2.2 节描述的外部接口信号对应,信号从上至下分别为 VMI,VPI,VMO,VPO,OE,SPD。从图中可看出 SPD 为高,表明主机工作在全速状态下,主机每 1 ms 向设备发送一个帧



图7 验证系统实物图

包(SOF),该事务紧跟随在SOF之后,共包含3个包。事务根据包的顺序传输,包按域的顺序传输,而在域中除了同步域和包标志,其它域的传输均遵循先低位后高位的顺序原则进行传输,对于多字节的数据域,按字节顺序传输,每个字节以先低位后高位的顺序传输。包标志域由低4位的包标志加上高4位取反验证码组成。以下将对事务中各个包做进一步分析。

图8(b)(c)(d)分别为该事务三个包的非归零反相编码(NRZI)时序图。NRZI编码原理是当输出0时总线状态取反,当输出1时总线状态保持不

变。因此我们对各包进行解码分析,以令牌包为例,在VMO上取得的令牌包信号的NRZI编码如表1所示,对其进行解码分析后得出该包的5个域:同步域(sync)、包标识域(PID)、地址域、端点域和校验域。可分析结果表面该包为建立请求包,当前设备的地址为0,并采用端点0进行传输。

同理分析得到数据包和握手包的信息如表2。数据包分4个域:同步域、包标识域、数据域和校验域。该数据包中数据描述的是主机获取设备描述符的请求包。其中,十六进制80表示获取标准数据,06表示设备描述符,40表示要获取64字节的数据。握手包分2个域:同步域和标志域。该握手包为应答握手,表明数据包被正确接收。

我们通过串口终端捕捉系统对U盘的应用测试过程,从图9显示的信息可以看出,主机系统在发现一个全速设备后(found Full-Speed device),首先对其进行总线复位(Reset device),然后开始枚举过程(enumerate),在识别设备为存储设备后(Mass storage device),进一步读取文件系统基本信息并识别该设备文件系统类型(FAT32)。我们编写的测试

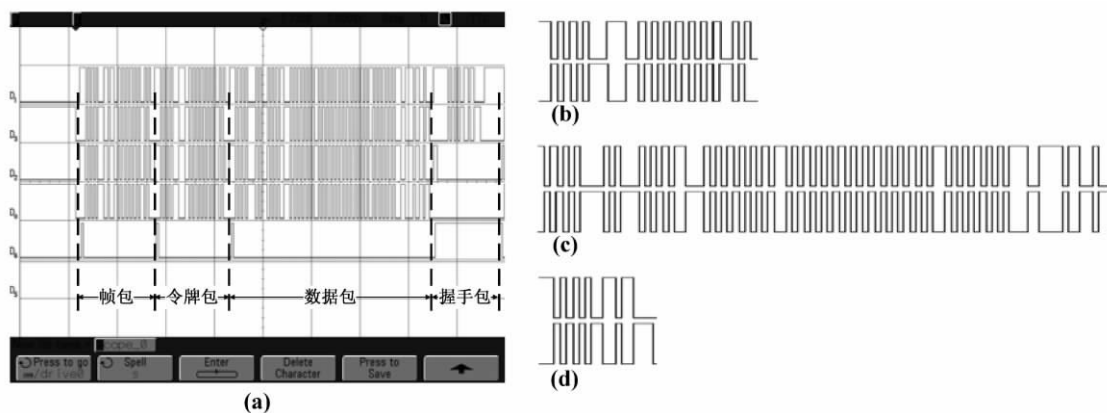


图8 (a)主机IP核端口时序;(b)令牌包;(c)数据包;(d)握手包

表1 令牌包分析

令牌包信息											
NRZI 编码	101010100 01110010 1010101 0101 00101										
解码值	00000001 10110100 0000000 0000 01000										
包信息	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td>Sync (8bits)</td> <td>SETUP (8bits)</td> <td>ADDR (7bits)</td> <td>ENP (4bits)</td> <td>CRC5 (5bits)</td> </tr> <tr> <td>00000001</td> <td>0Xb4</td> <td>0x0</td> <td>0x0</td> <td>0x2</td> </tr> </table>	Sync (8bits)	SETUP (8bits)	ADDR (7bits)	ENP (4bits)	CRC5 (5bits)	00000001	0Xb4	0x0	0x0	0x2
Sync (8bits)	SETUP (8bits)	ADDR (7bits)	ENP (4bits)	CRC5 (5bits)							
00000001	0Xb4	0x0	0x0	0x2							
实际值	00000001 0Xb4 0x0 0x0 0x2										

表 2 数据包 / 握手包信息

包信息	数据包信息				握手包信息	
	Sync (8bits)	DATA0 (8bits)	DATA (8bytes)	CRC16 (16bit)	Sync (8bits)	ACK (8bits)
解码值	00000001	11000011	01 60 00 80 00 00 02 00	BB29	00000001	01001011
实际值	00000001	0Xc3	80 06 00 01 00 00 40 00	94DD	00000001	0X4b

程序通过输入控制命令完成对 U 盘的文件操作，getcap 读取 U 盘的总空间和当前的可用空间，crtdir 创建一个文件夹目录，crtfil 创建一个文件，opnfil 打开该文件，setptr 设置读写操作的起始指针，wrfil、rdfil 进行读写操作，完成后 clsfil 关闭该文件。通过 cd ls 显示文件信息可以看到我们所创建的目录及文件的信息。结果表明：对 U 盘应用测试取得很好的效果。

此外在进行 U 盘文件操作测试程序设计时发现一个现象：如果停止对 U 盘的操作一段时间（大于 2 s），再对 U 盘进行批量传输的话就得不到任何数据回应，但在此过程中主机每 1 ms 都向设备发生 SOF 包，表明设备并没有进入挂起状态。针对这一现象我们观察了 PC 机对 U 盘的操作过程，通过 Bus Hound 软件观察到，在空闲状态下 PC 主机大约每 1 s 向 U 盘发送一个 TEST UNIT READY 命令（图 10）。为此我们通过定时器产生一个每 1 s 的定时中

断使程序在 U 盘空闲时每秒发送一次查询命令，解决了这一问题。



图 10 PC 机对 U 盘的定时查询

## 6 结束语

本文通过基于 FPGA 的 USB 主机硬件实现和

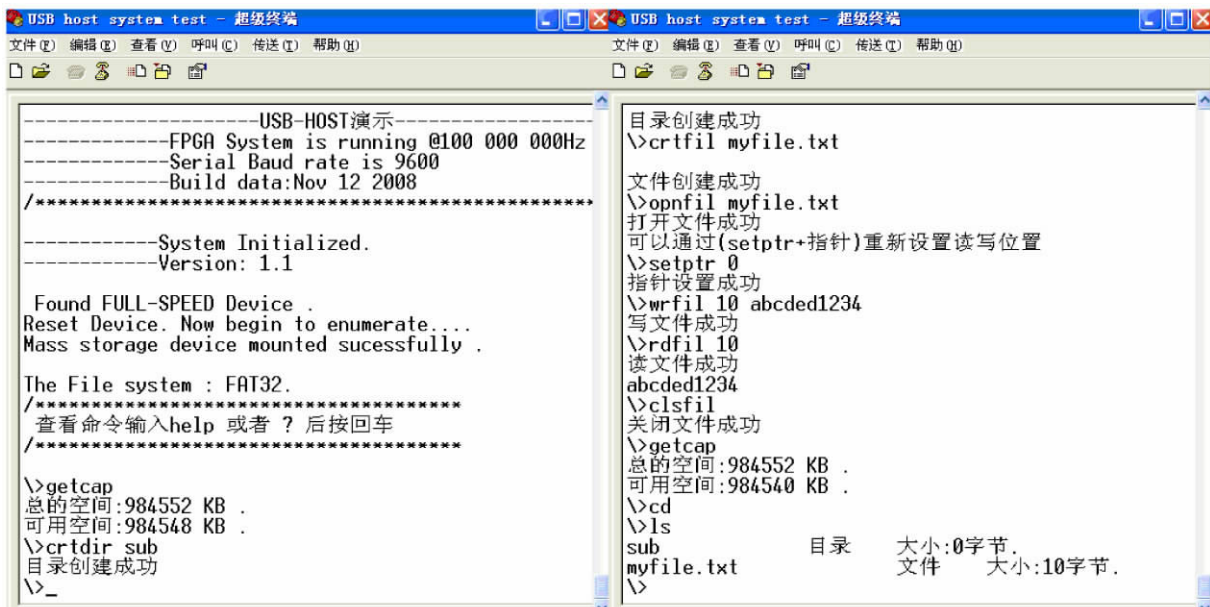


图 9 文件系统演示

驱动程序、HID 和 U-disk 两类典型应用测试程序的软件开发,完成了对 PowerPC 架构的 USB 主机 SoC 设计方案的软硬件协同验证。结果表明该 USB 主机控制器的设计符合 USB 技术规范,并能很好地与 Intel、Cypress 等厂商的 USB 设备兼容。同时,在测试过程中所编写的软件驱动和移植的文件系统等均可以直接应用到 SoC 的后续开发中,降低了后续开发的难度。■

### 参考文献

- [1] 余瑞琰. USB 市场应用发展趋势[J]. 电子产品世界,2006,5. <http://www.eepw.com.cn/article/12719.htm>.
- [2] Universal Serial Bus Specification Revision 1.1[S]. <http://www.usb.org>, 1998.
- [3] 胡天林,王磊,林春等. 新型紫外分光光度计的 USB 采集卡设计[J]. 厦门大学学报,2009,48(3) 365-368.
- [4] 刘小军,刘栋斌. 基于 EZ-USB 的 CCD 信号采集系统[J]. 计算机工程,2009,35(14) 203-205.
- [5] 于晶荣,滕召胜,孙传奇等. 基于 USB 主机技术的三相多功能电能表设计[J]. 湖南大学学报,2007,34(2) 51-55.
- [6] 沈小磊,张晓彤,李占才. USB 设备控制器的硬件实现与研究[J]. 计算机工程,2007,33(24) 247-249.
- [7] 付华杰,刘丽君,张遂南. 高速 USB IP 核的设计与开发[J]. 微电子学与计算机,2008,25(7) 127-129.
- [8] 戴迎珺. 基于 SoPC 的 USB 设备控制器 IP 核设计[J]. 计算机工程与设计,2008,29(20) 5234-5236.
- [9] 武甲东,梅张雄等. USB 主机控制器端串行接口引擎 IP 的实现[J]. 微电子学与计算机,2005,22(2) 35-38.
- [10] 张美林,章军. USB1.1 主机控制器 IP 核的设计和实现[J]. 计算机工程与应用,2007,43(22) 109-112.
- [11] 胡锦,胡立琴,陈训亮. 一种嵌入式 USB2.0 主机控制器 IP 核的研究与设计[J]. 微电子学与计算机,2009,1:133-136.
- [12] 廖文兵. USB2\_0 OTG IP 核全速主机控制器研究与设计[D]. 电子科技大学,2008.
- [13] Chen Wenwei, Zhang Jinyi, Li Jiao, Ren Xiaojun and Liu Jiwei, "Study On a Mixed Verification Strategy for IP- Based SoC Design", IEEE High Density Microsystem Design and Packaging and Component Failure Analysis, Pages:1-4, June 2005.
- [14] 李栋,李正卫. 基于 SoC 的 USB 主设备的软硬件协同验证[J]. 中国集成电路,2007,2:53-57.
- [15] OPB IPIF (v3.01c) DataSheet [EB/OL]. [http://www.xilinx.com/support/documentation/ipbusinterfacei-o\\_coreconnect\\_opb-ipifarchitect.htm](http://www.xilinx.com/support/documentation/ipbusinterfacei-o_coreconnect_opb-ipifarchitect.htm).

### 作者简介

梁瑞彬, 在读硕士研究生, 研究方向为嵌入式系统设计与应用;  
李晓潮, 副教授, 研究领域: 集成电路、嵌入式系统、扩谱通信和多媒体;  
郭东辉, 教授, 博士生导师, 研究领域: 集成电路设计, 人工智能, 网络通信等。

上接第 35 页

- design rule, Version 1.4.1, 2008
- [3]. ICC user guide, 2008.09 version
- [4]. ICC command reference, 2008.09 version
- [5]. PrimeRail user guide, 2008.09 version
- [6]. 《数字集成电路——设计透视(第 2 版)》, Jan M.Rabaey, Anantha Chandrakasan, Borivoje Nikolic,

清华大学出版社, 北京, 2004

### 作者简介

曾宏, 高级芯片物理设计工程师, 芯原微电子(上海)有限公司。