

PowerPC 架构下 USB3.0 的 IP 核软硬件协作验证

邓佳伟, 张梅娟, 王 琪, 杨楚玮, 应凌楷

(中国电子科技集团第五十八研究所 CPU 研究室, 江苏 无锡 214062)

摘要: 随着 USB(Universal Serial Bus)设备传输速率要求的不断提高,传统基于 USB2.0 控制器设计的 PowerPC 架构处理器已经不能满足高速率、大容量传输的需求。提出了一种基于 PowerPC 架构的 USB3.0 的处理器解决方案。通过对比 USB3.0 和 USB2.0,USB3.0 拥有更高的传输速率,同时具备更强的抗干扰能力。通过和高性能 PowerPC 处理器协同工作,显著提高了 USB 控制器的传输能力,充分利用了 PowerPC 计算资源。阐述了一种基于 PowerPC 架构下 USB3.0 的 IP 核(Intellectual Property Core)方案设计。最后结合软硬件协作验证方法,验证方案可行性,为后续芯片设计和验证奠定了一定的技术基础。

关键词: PowerPC; USB3.0; Linux; 软硬件协作验证; IP 核

中图分类号: TN401

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.212329

中文引用格式: 邓佳伟,张梅娟,王琪,等. PowerPC 架构下 USB3.0 的 IP 核软硬件协作验证[J]. 电子技术应用, 2022, 48(5): 37-41.

英文引用格式: Deng Jiawei, Zhang Meijuan, Wang Qi, et al. Hardware/software co-verification of USB3.0 IP core under PowerPC architecture[J]. Application of Electronic Technique, 2022, 48(5): 37-41.

Hardware/software co-verification of USB3.0 IP core under PowerPC architecture

Deng Jiawei, Zhang Meijuan, Wang Qi, Yang Chuwei, Ying Lingkai

(CPU Research Office, The 58th Research Institute, CETC, Wuxi 214062, China)

Abstract: With the increasing requirement of transmission rate of USB, the traditional PowerPC processor based on the USB2.0 can not meet the requirements of high speed and large-capacity transmission. The paper presents a solution of USB 3.0 base on PowerPC processor architecture. By comparing USB3.0 and USB2.0, USB3.0 has higher transmission rate. By cooperating with the high performance PowerPC processor, the transfer capability of the USB controller is improved significantly, and the PowerPC computing resources are fully utilized. This paper describes a programme of USB3.0 IP core based on the PowerPC architecture. Finally the method of hardware/software cooperative verification is used to verify the feasibility of the design, which lays a technical foundation of subsequent chip design and verification.

Key words: PowerPC; universal serial bus 3.0; Linux; hardware/software co-design technology; IP Core

0 引言

传统 USB2.0 协议最大带宽 480 Mb/s, USB2.0 已经不能满足用户的需求, USB3.0 作为新一代高速接口,不但兼容 USB2.0 协议,而且在低速、全速、高速的三种传输模式基础上增加了一种超高速模式。USB3.0 工作单元拥有两个数据传输通道,一对通道用于传输速率可高达 5 Gb/s 的超高速数据传输,同时还有一对通道用于支持 USB2.0 的数据通信。

PowerPC 架构是一种精简指令集(RISC)架构,其原始的设计源自 IBM 的 POWER RISC 架构。PowerPC 架构处理器有着广泛的应用,主要集中在服务器 CPU 和嵌入式 CPU 市场。PowerPC 在嵌入式处理器上表现非常优异,不仅具备高性能、低功耗以及较低散热量等特点,而且资源丰富,能够满足各类需求。

芯片验证主要为了验证研制芯片的功能性,验证其各功能设计正确。验证的方法有很多,主要有基于事件的验证、基于周期的模拟验证、基于事务的验证、软硬件协同验证、验证仿真器验证和形式验证等。基于软硬件系统验证^[1-2]是将软硬件同时集成并加以验证,验证人员直接在芯片设计过程中参与,在硬件设计平台上运行软件。协同验证中验证平台设计采用真实的输入激励来模拟,减少了验证平台和芯片差异性^[3],但由于验证平台一般的模拟速度不够快等问题,系统验证提供不了足够高的性能,特别在实时操作系统上运行各种应用。

本方案主要使用的是软硬件协同验证,通过 Xilinx 的 VCU118 的验证硬件平台, Linux4.1.8 操作系统系统软件平台,协同配合高效能地完成了方案设计,证实了方案的可行性。

1 系统架构

本方案中验证平台 FPGA 包含一个 PowperPC 处理器 IP 逻辑,一个 USB3.0 IP 逻辑。各逻辑单元的功能如下。

PowerPC 处理器 IP 逻辑单元:基于 PowerPC 指令集的一款处理器内核。其主频支持最高达 1.2 GHz。

USB 控制器 IP 逻辑单元:USB 控制器逻辑内部包含 USB3.0 和 USB2.0 两部分。其中 USB3.0 控制器逻辑支持最高可达 5 Gb/s 的传输速率,USB2.0 控制器逻辑支持最高到 480 Mb/s 的传输速率。IP 核内包括了 AHB (Advanced High Performance Bus)总线接口、USB 控制器、发送接收缓存和 USB3.0 和 USB2.0 的物理层 PHY(Port Physical Layer)。

图 1 为整体 IP 逻辑方案设计图。

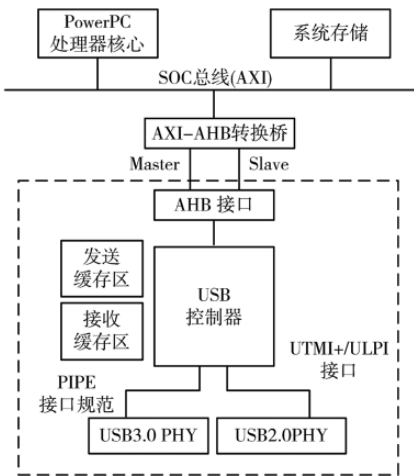


图 1 整体 IP 逻辑方案设计图

1.1 逻辑单元接口设计

本方案的接口主要有处理器和 USB 控制器之间的接口、USB 控制器和物理层 PHY 的接口。

(1)USB3.0 控制器和 PowerPC 接口

PowerPC 处理器 IP 逻辑和 USB 控制器逻辑通过 AXI (Advanced Extensible Interface)/AHB 总线协议连接。

PowerPC 处理器 IP 逻辑通过 AXI 总线连接到 AXI-AHB 转换桥,桥和 USB3.0 控制器的 AHB 接口连接。AHB 总线通过 Master 和 Slave 两路数据通道连接到 USB3.0 的控制器上,Master 总线用作 DMA 通道,提供给 CPU 和 USB 设备的数据通信,Slave 总线用作读写 USB 控制和状态寄存器和通过 SRAM 进行调试^[4]。

(2)USB 控制器逻辑和 PHY 的标准接口

USB3.0 控制器逻辑和物理层 PHY 通过 PIPE3(Physical Interface for PCI Expressc)。USB2.0 控制器逻辑通过 UTMI+(USB2.0 Transceiver Macrocell Interface)/ULPI (UTMI+Low Pin Interface)标准接口连接。

1.2 USB IP 设计

USB3.0 设备控制器 IP 设计如图 2 所示,主要分为六部分:USB 通信功能单元、时钟单元、复位单元、同步单元、应用接口单元和 PHY 单元。各单元功能如表 1 所示。USB 设备控制器时钟单元模块负责内部时钟、复位单位负责复位的管理。应用接口单元负责控制器功能模块和系统之间的通信。USB 通信功能单元主要负责完成 USB 的高速和超高速协议分析^[5]。同步单元的作用是应用接口单元与 USB 通信功能模块的工作时钟在不同的时钟域,通过同步单元进行时钟同步,PHY 单元负责和硬件通信。

表 1 USB IP 内部单元

USB 内部单元	功能
时钟单元	负责内部时钟、复位单位负责复位的管理
应用接口单元	负责控制器功能模块和系统之间的通信
通信功能单元	负责完成 USB 的高速和超高速协议分析
同步单元	同步单元进行时钟同步
PHY 单元	负责和硬件通信

同步单元中包含了多路数据选择器,其根据 USB 工作的速度模式来决定数据流走向^[6-8]。

数据可以分流到 USB2.0 或者 USB3.0 的数据链路上。USB3.0 协议向下兼容 USB2.0 协议。因此 USB3.0 设

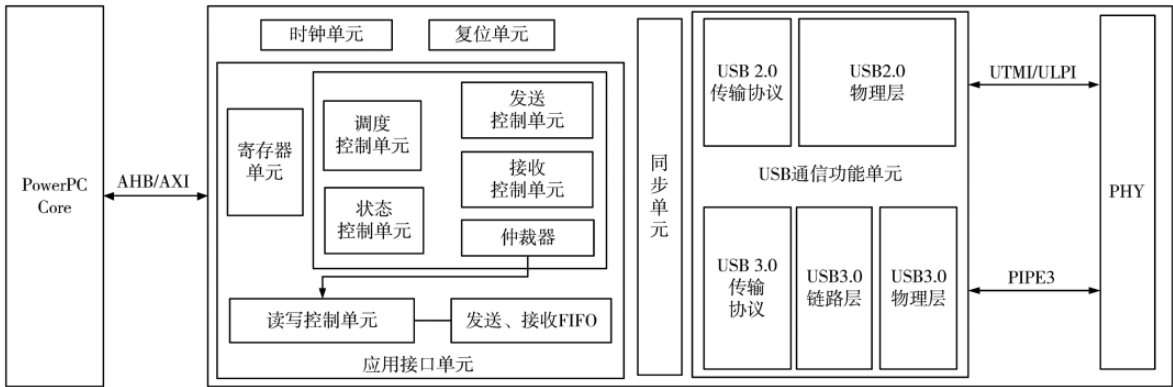


图 2 USB3.0 设备控制器 IP 核设计

备控制器应实现高速和超高速两部分。USB3.0 实现了超高速部分的协议层和链路层^[9]。通过标准的 PIPE3 接口与第三方物理层相连完成超高速功能。USB2.0 实现了全速及以下的传输层协议。通过标准的 UTMI 或者 ULPI 连接到内置 PHY 上。最后通过 PHY 直接和对端设备连接^[10-11]。

2 软件架构

Linux 下 USB 驱动分为 HOST(主机)端和 DEVICE(设备)端,图 3 为 USB 的 Linux 驱动框架。

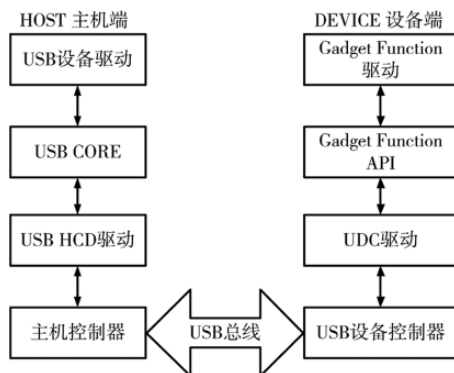


图 3 Linux USB 程序框架图

2.1 主机端驱动

Linux 的 USB 设备驱动分为两部分:主机端驱动和设备端驱动。

主机端驱动包括 USB device driver 驱动、USB Core 驱动、USB HCD 驱动。

(1)USB device driver 驱动:管理 USB 设备和主机通信。

(2)USB Core 驱动:设备驱动程序,提供一个 USB device driver 驱动访问和控制 USB 硬件的接口,不用考虑系统当前使用哪种处理器架构。USB Core 将应用的请求反馈到相关的 HCD 驱动,应用不能直接访问 HCD。

(3)USB HCD 驱动:主机控制器之上运行的是 HCD,是对主机控制器硬件的一个抽象,USB HCD 有多种 USB 接口协议规范。

①UHCI: Intel 提供,通用主机控制接口,支持协议 USB1.0/1.1;

②OHCI: 微软提供,开放主机控制接口,支持协议 USB1.0/1.1;

③EHCI: 增强主机控制接口,支持协议 USB2.0;

④XHCI: 支持协议 USB3.0,同时兼容 USB2.0 以下版本。

2.2 设备端驱动

设备端驱动包括:Gadget Function API 驱动、Gadget Function 驱动、UDC 驱动。

(1)Gadget Function API:USB 从设备驱动调用 USB Core 的 API。

(2)Gadget Function 驱动:Function 驱动调用通用的 Gadget Function API。

(3)USB 设备端 UDC 驱动:作为其他 USB 主机控制器外设的 USB 硬件设备上底层硬件控制器的驱动。

3 具体验证方案实现

3.1 验证整体方案

本方案使用的验证平台为 Xilinx 的 VCU118 开发板卡。验证整体方案如图 4 所示。VCU118 板卡上拓展了 JTAG 子板、自研 USB/DDR 子板。VCU118 开发板上拥有多个接口,JTAG 接口用来从上位机下载 FPGA 逻辑文件固化到 FPGA 上。UART 接口用来和上位机电脑进行数据通信,显示系统输出打印。SD 接口用来对接 SD 卡,UBOOT 程序和 Linux 内核程序固化放在 SD 卡中。USB 接口用来拓展 USB 子板,该子板上有 U 盘插槽用来插入 U 盘。DDR 接口用来对接扩展 DDR 内存,系统上电后从 SD 卡中读取 Linux 系统到 DDR 上运行。

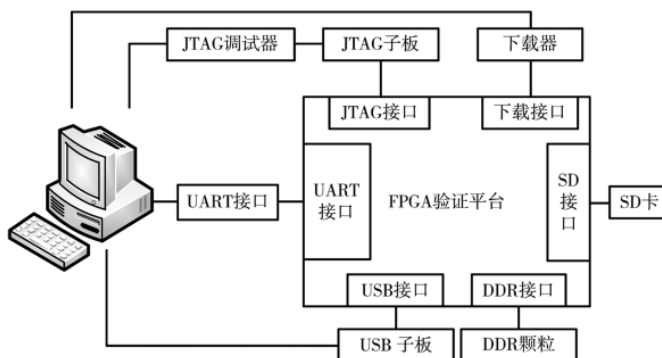


图 4 验证整体方案图

3.2 验证过程

具体的验证过程如下:

- (1)Xilinx 开发工具系统综合生成 SOC 的逻辑文件;
- (2)通过 Xilinx 下载器将逻辑文件通过 JTAG 下载到 FPGA 验证平台;
- (3)热复位验证平台;
- (4)验证平台热复位后,SD 接口将存在 SD 卡中 Linux 的 BOOT 程序和 Linux 内核程序下载到 DDR 存储上;
- (5)程序从 DDR 开始运行;
- (6)等待 Linux 系统完全启动;
- (7)上位机通过 UART 接口和连接验证平台,上位机通过控制台操作 Linux 系统的交互界面;
- (8)控制台通过测试工具,测试 USB2.0 和 USB3.0 的大型存储设备;
- (9)查看能否启动 USB 设备,能否正常读写设备。

3.3 Linux 驱动适配

本方案的 Linux 内核版本为 4.1.5,主要修改的是 HCD 驱动程序,由于验证的是 USB3.0 及其以下版本,因此用的是 XHCI 标准,通过修改 XHCI 对接的 HOST 控制器代

码来实现适配 PowerPC 处理器主控制器^[12]。

在 linux_kernel/deriver/usb/host 下为 xhci 的源代码。其中 usb_xhci_driver 是 xhci 的核心结构体。

```
static struct platform_driver usb_xhci_driver = {
    .probe=xhci_plat_probe,
    .remove=xhci_plat_remove,
    .driver={
        .name="xhci-hcd",
        .pm=DEV_PM_OPS,
        .of_match_table =of_match_ptr (usb_xh-
ci_of_match),
    },
};
```

xhci_plat_init 函数完成了 xhci 驱动的初始化,在其调用的 xhci_init_driver 函数中 xhci_plat_hc_driver 的参数是需要修改的主要结构体。在该结构体下,主要完成的是对 USB 主控制器的抽象。程序主要工作就是适配主控制器到该结构体中^[13-15]。

```
static int __init xhci_plat_init(void)
{
    xhci_init_driver(&xhci_plat_hc_driver, xhci_plat_setup);
    xhci_plat_hc_driver.start = xhci_plat_start;
    return platform_driver_register(&usb_xhci_driver);
}
```

4 验证平台具体设计与验证结果

4.1 验证平台

实验验证平台使用 Xilinx 公司的 VCU 118 开发套件平台和自研设计的 USB3.0 拓展子卡验证 USB3.0 的 IP 设计。

图 5 为整个开发平台和拓展子卡的连接方式,图中位置 1 为 JTAG 的位置;位置 2 为自研 USB 插槽;位置 3 的背面是 SD 卡外部存储插槽;位置 4 为 UART 接口位置;位置 5 为下载器调试接口;位置 6 的背板上有 DDR 颗粒。



图 5 实验场景图

当设计调试完成后,将 PowerPC 架构下的 USB3.0 的 IP 逻辑文件烧录到验证平台卡中。将 VCU118 平台

通过拨码切换到 SD 卡启动方式。Linux 相关程序下载到 DDR 中。当系统启动后,通过上位机控制台和 UART 串口访问平台,USB2.0 和 USB3.03.0 的 U 盘插入 USB 子卡,在控制台输入各种命令查看 USB 在位情况。通过格式化 U 盘,写入文件来验证 USB3.0 传输单元设计成功。

4.2 验证结果分析

USB2.0 U 盘插入显示:图 6 中框内就是 USB2.0 的 U 盘。

```
[root@localhost home]# lsusb -t
/: Bus 06.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
/: Bus 05.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 480M
/: Bus 04.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
/: Bus 03.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
|_ Port 3: Dev 17, If 0, Class=Mass Storage, Driver=usb-storage, 480M
|_ Port 3: Dev 17, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|_ Port 6: Dev 2, If 0, Class=Vendor Specific Class, Driver=, 12M
|_ Port 6: Dev 2, If 2, Class=Human Interface Device, Driver=usbhid, 12M
/: Bus 02.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
/: Bus 01.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/6p, 480M
|_ Port 2: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|_ Port 2: Dev 2, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
[root@localhost home]#
```

图 6 USB2.0 枚举

USB3.0U 盘插入显示:图 7 中框内就是 USB3.0 的 U 盘。

```
[root@localhost home]# lsusb -t
/: Bus 06.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
|_ Port 3: Dev 3, If 0, Class=Mass Storage, Driver=usb-storage, 5000M
/: Bus 05.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 480M
/: Bus 04.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
|_ Port 4: Dev 4, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|_ Port 6: Dev 2, If 0, Class=Vendor Specific Class, Driver=, 12M
|_ Port 6: Dev 2, If 2, Class=Human Interface Device, Driver=usbhid, 12M
/: Bus 02.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/4p, 10000M
/: Bus 01.Port 1: Dev 1, Class=root hub, Driver=xhci_hcd/6p, 480M
|_ Port 2: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
|_ Port 2: Dev 2, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
[root@localhost home]#
```

图 7 USB3.0 枚举

当 U 盘插入到插槽后,内核打印如图 8 所示。

通过 Linux 系统给 U 盘制作文件系统,如图 9 所示。

通过测试可以看到,U 盘插入到插槽后,可以在内核的输出打印上看到 U 盘的相关信息,说明 U 盘已经被系统识别。U 盘可以正常地格式化,在 U 盘格式化过程中,会在内核中操作 USB 主控制器,可以成功地在验证平台上运行和格式化 U 盘,说明在 PowerPC 架构下的 USB3.0 已经可以正常工作。

5 结论

随着 USB3.0 设备的广泛应用,大量的高传输、大容量传输场景涌现,为了适应更高传输速率的 USB 设备,各类处理器的 USB 控制器也将升级到 USB3.0 标准。

本方案集合 PowerPC 处理器和 USB3.0 的 IP 核,通过 Xilinx 的 VCU118 开发验证平台,验证了 PowerPC 架构下 USB3.0 的可行性,有效弥补了原 USB2.0 控制器速率低、容量小、抗干扰能力弱等缺点,验证方式采用了先进的软硬件协作验证方式,为下一步芯片设计和验证提供了有效技术支撑。

[illegible]

图 8 USB 设备枚举内核打印

```
root@localhost home]# mkfs.ext2 /dev/sdb
e2fs 1.45.4 (23-Sep-2019)
Found a dos partition table in /dev/sdb
Proceed anyway? (y,N) y
/dev/sdb is apparently in use by the system; will not make a filesystem here
root@localhost home]# mkfs.ext2 /dev/sdb
e2fs 1.45.4 (23-Sep-2019)
Found a dos partition table in /dev/sdb
Proceed anyway? (y,N) y
Creating filesystem with 519808 1k blocks and 130048 inodes
Filesystem UUID: 9d7e9e81-0f0a-477e-a4be-bd96851df785
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

[root@localhost home]# mkdir /mnt/usb -p
[root@localhost home]# cd /mnt/usb/
[root@localhost usb]# mount /dev/sdb /mnt/usb/
[root@localhost usb]# ls
[root@localhost usb]# touch a.txt
```

图 9 USB 设备操作

参考文献

- [1] 梁双杰. SoC 软硬件协同设计系统级验证方法研究[D].

西安:西安电子科技大学,2014.

- [2] 雷淑岚, 吴会祥, 李文学. 基于 PowerPC 架构的波束指向算法的优化[J]. 电子技术应用, 2021, 47(3): 79-82, 90.
- [3] 程翼胜. SoC 芯片 FPGA 原型的软硬件协同验证[J]. 单片机与嵌入式系统应用, 2017, 17(11): 7-10, 13.
- [4] 郭鼎盛. USB2.0 控制器的 IP 核验证[D]. 哈尔滨: 哈尔滨工业大学, 2012.
- [5] 黄凯, 林威, 蒋进松, 等. 低功耗可配置的 USB3.0 设备控制器 IP 核设计[J]. 计算机工程, 2015, 41(12): 1-8.
- [6] 刘云晶, 刘梦影. 一种 32 位 MCU 的 FPGA 验证平台[J]. 电子与封装, 2020, 20(1): 41-47.
- [7] Universal serial Bus 3.0 specification revision 1.0[S]. USA: USB Organization, 2008, 11.
- [8] PHY interface for the PCI express and USB 3.0 architectures. Version 3.0[S]. USA: Intel, 2009, 4.
- [9] WANG J Y. The architecture and design of USB3.0 device controller[Z]. USA, 2010: 102-119.
- [10] JAIN S, PRASANNA G. Using CRC hard blocks in virtex-5 FPGAs: Xcell journal[Z]. USA, 2007: 45-58.
- [11] 夏宇闻. Verilog 数字系统设计教程[M]. 2 版. 北京: 北京航空航天大学出版社, 2008: 20-135.
- [12] 吴从中, 彭乐, 王亚君, 等. USB3.0 设备控制器 IP 核 OUT 端点测试平台的研究与实现[J]. 电子测量与仪器学报, 2012, 26(7): 646-651.
- [13] 章玮. 原型验证过程中的 ASIC 到 FPGA 的代码转换[J]. 今日电子, 2006(7): 56-59.
- [14] 唐宇, 廖小雨, 骆少明, 等. 基于 CPU 和 DDR 芯片的 SiP 封装可靠性研究[J]. 电子元件与材料, 2015, 34(4): 79-83.
- [15] 江敏. 基于 ARM 嵌入式系统的 Boot Loader 启动分析与移植[J]. 信息通信, 2018(7): 98-99.

(收稿日期:2021-11-15)

作者简介：

邓佳伟(1987-),男,硕士,工程师,主要研究方向:嵌入式软件开发、系统级验证。

张梅娟(1981-),女,硕士,高级工程师,主要研究方向:PowerPC 架构、IP 核验证、系统级验证。

王琪(1992-),男,学士,工程师,主要研究方向:嵌入式软件开发、系统级验证。



扫码下载电子文档

版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所