

《嵌入式系统软硬件协同设计实战指南》

图书基本信息

书名：《嵌入式系统软硬件协同设计实战指南》

13位ISBN编号：9787111411079

10位ISBN编号：7111411072

出版时间：2013-1-26

出版社：机械工业出版社

作者：陆佳华,江舟,马岷

页数：354

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《嵌入式系统软硬件协同设计实战指南》

内容概要

本书由浅入深，由基础知识到实战案例向读者系统阐述了如何利用Zynq平台进行嵌入式系统以及软硬件协同设计的开发。本书分为基础篇与进阶篇两部分，基础篇中介绍了Zynq器件、ZedBoard，并配有简单入门实验，同时针对软件开发人员增设了FPGA硬件加速等内容。在进阶篇中介绍了利用Zynq进行软硬件协同设计，同时对处理器与可编程逻辑接口等技术进行了详细剖析。本书提供了20个详细的设计案例，涵盖了硬件板卡、FPGA逻辑、Linux驱动、Linux操作系统、上层应用、软硬件协同设计等Zynq开发中可能遇到的各个方面的知识，并在最后将前述独立案例整合为4个系统案例。本书重点突出实战，以案例为指导，配合介绍相关参考文档，协助读者尽快掌握在Zynq上进行各项设计的方法。

本书可作为Zynq初学者、软硬件协同设计开发人员的参考用书，亦可作为大专院校嵌入式系统设计、片上系统设计、可编程逻辑器件等相关专业的教师和学生的参考用书。

《嵌入式系统软硬件协同设计实战指南》

作者简介

陆佳华，开源硬件社区Openhw.org资深版主，Xilinx高级应用工程师，现任职于Xilinx全球大学计划部，主要负责Xilinx全球大学相关的参考设计开发，技术推广、支持。陆佳华2006年毕业于西安交通大学并获得硕士学位。2006年加入Xilinx公司后任产品应用工程师，主要负责FPGA上的嵌入式系统设计，以太网设计，内存控制器设计等方向技术支持。陆佳华著有《零存整取-NetFPGA开发指南》一书。

江舟，开源硬件社区Openhw.org的资深版主，浙江大学硕士研究生，现为Xilinx全球大学计划部实习生，主要参与Zynq上的软件开发。江舟是美信DIY大赛的主要技术支持者之一，并担任Xilinx开源硬件大赛的技术支持。

马岷，开源硬件社区Openhw.org的资深版主，浙江大学硕士研究生，现为Xilinx全球大学计划部实习生，主要参与Zynq上的硬件开发。马岷是美信DIY大赛的主要技术支持者之一，并担任Xilinx开源硬件大赛的技术支持。

书籍目录

Foreword

前言

第一部分 基础篇

第1章 初试ZedBoard/2

1.1 GPIO LED动手玩/2

1.1.1 拷贝SD卡/2

1.1.2 跳线与外设连接/2

1.1.3 演示操作/2

1.2 Linaro Ubuntu动手玩/3

1.2.1 SD卡分区/3

1.2.2 文件拷贝 (FAT/EXT) /6

1.2.3 外设连接/6

1.2.4 可演示的效果/7

第2章 Zynq平台介绍/9

2.1 7系列FPGA简介/9

2.2 Zynq-7000 AP SoC体系简介/12

第3章 ZedBoard开发环境/15

3.1 ZedBoard的板载外设/15

3.1.1 LED/15

3.1.2 按键/16

3.1.3 开关/16

3.1.4 OLED/17

3.1.5 USB接口/18

3.1.6 音频接口/20

3.1.7 VGA接口/21

3.1.8 HDMI接口/22

3.1.9 10/100/1000兆网口/23

3.2 ZedBoard的扩展外设/25

3.2.1 外扩PMod插座/25

3.2.2 外扩FMC插槽/27

3.2.3 外扩AMS插座/28

第4章 开发工具链/29

4.1 可编程逻辑开发工具链/29

4.1.1 PlanAhead/29

4.1.2 Xilinx Platform Studio/31

4.2 软件开发工具链/34

4.2.1 Xilinx Software Development Kit/34

4.2.2 交叉编译工具链/35

4.3 软硬件调试工具/36

4.3.1 ChipScope Pro/36

4.3.2 GDB与GDBserver/38

第5章 Zynq体系结构/40

5.1 应用处理器单元 (APU) /40

5.1.1 ARM Cortex A9处理器/40

5.1.2 侦听控制单元 (SCU) /43

5.1.3 L2高速缓存/44

5.1.4 APU接口/44

- 5.2 通用外设/46
 - 5.2.1 通用IO (GPIO) /46
 - 5.2.2 SPI接口/49
 - 5.2.3 UART接口/51
 - 5.2.4 计时器/54
 - 5.2.5 USB控制器/57
 - 5.2.6 DDR控制器/58
- 5.3 数字逻辑设计/59
 - 5.3.1 可编程逻辑“ 外设 ” (PL) /59
 - 5.3.2 XADC/61
 - 5.3.3 PCIe/62
- 5.4 MIO/EMIO/63
- 第6章 系统级信号/66
 - 6.1 电源管理/66
 - 6.2 Clock信号/67
 - 6.2.1 CPU时钟域/68
 - 6.2.2 DDR时钟域/69
 - 6.2.3 基本的时钟分支结构/69
 - 6.2.4 I/O外设 (IOP) 时钟/70
 - 6.2.5 PL时钟/72
 - 6.2.6 其他时钟/72
 - 6.3 复位系统/73
 - 6.4 JTAG/75
 - 6.5 中断处理/76
- 第7章 Zynq启动与配置/78
 - 7.1 Zynq启动过程简介/78
 - 7.2 外部启动条件/79
 - 7.2.1 电源要求/79
 - 7.2.2 时钟要求/79
 - 7.2.3 复位要求/79
 - 7.2.4 启动引脚设置/80
 - 7.3 BootROM/80
 - 7.3.1 BootROM的作用/80
 - 7.3.2 BootROM的特点/81
 - 7.3.3 BootROM后的状态/82
 - 7.4 FSBL/82
 - 7.5 SSBL/84
 - 7.6 Linux启动过程/84
 - 7.7 Secure Boot/86
- 第8章 面向软件工程师的逻辑设计/87
 - 8.1 FPGA硬件加速原理/87
 - 8.1.1 以空间换时间/87
 - 8.1.2 以存储器换门电路/89
 - 8.1.3 以IP集成换生产力/90
 - 8.2 部分动态可重配置于Zynq/93
- 第9章 ZedBoard入门/95
 - 9.1 UART和GPIO控制/95
 - 9.1.1 UART和GPIO接口/95
 - 9.1.2 硬件设计过程/96

- 9.1.3 软件设计过程/106
- 9.2 硬件/软件调试方法/112
 - 9.2.1 ChipScope IP Core/112
 - 9.2.2 SDK Gdb使用/115
- 9.3 搭建你的单板计算机 (Single Board Computer)/117
 - 9.3.1 搭建系统环境/118
 - 9.3.2 准备工作/118
- 第二部分 进阶篇
- 第10章 基于虚拟平台的Zynq开发/126
 - 10.1 QEMU介绍/126
 - 10.2 编译QEMU源码/126
 - 10.2.1 下载QEMU源码/126
 - 10.2.2 配置QEMU/127
 - 10.2.3 QEMU所依赖的库文件/127
 - 10.2.4 编译QEMU/127
 - 10.3 启动QEMU/127
 - 10.4 QEMU中的嵌入式Linux/128
 - 10.5 商业版虚拟平台/131
- 第11章 PL和PS的接口技术详解/132
 - 11.1 PL和PS的接口/132
 - 11.1.1 AXI接口简介/133
 - 11.1.2 AXI Interconnect/134
 - 11.2 Zynq的内部连接/137
 - 11.2.1 AXI_HP/139
 - 11.2.2 AXI_GP/140
 - 11.2.3 AXI_ACP/140
 - 11.3 PL和存储器系统性能概述/142
 - 11.3.1 接口理论带宽/142
 - 11.3.2 DDR控制器的吞吐率及其效率/143
 - 11.3.3 内部互连吞吐量瓶颈/143
 - 11.3.4 如何选择PL的接口/144
- 第12章 基于Zynq的软硬件协同设计/149
 - 12.1 多核处理器架构简介/149
 - 12.1.1 什么是多核处理器/149
 - 12.1.2 多核处理器发展的动机和优势/150
 - 12.1.3 同构、异构多核架构的优点和挑战/152
 - 12.2 软硬件协同设计方法论/152
 - 12.2.1 什么是软硬件协同设计/152
 - 12.2.2 软硬件协同设计发展的动机和优势/152
 - 12.2.3 软硬件协同设计的基本流程/153
 - 12.2.4 基于Xilinx工具的软硬件协同设计简介/154
 - 12.3 高层次综合/154
 - 12.3.1 高层次综合综述/154
 - 12.3.2 高层次综合发展的动机与优势/155
 - 12.3.3 Xilinx AutoESL工具简介/156
 - 12.4 基于Xilinx Zynq的软硬件协同设计实例/157
 - 12.4.1 功能简介/157
 - 12.4.2 设计流程简介/157
 - 12.4.3 实验结果与验证/165

- 第13章 Zynq开发实战/166
 - 13.1 用户IP设计/166
 - 13.1.1 用户IPcore介绍/166
 - 13.1.2 用户IPcore设计/167
 - 13.2 嵌入式Linux设备驱动开发/180
 - 13.2.1 设备驱动开发介绍/180
 - 13.2.2 驱动程序的加载与卸载/181
 - 13.2.3 sys文件系统简介/181
 - 13.2.4 PWM模块驱动程序/182
 - 13.2.5 PWM驱动程序编译与测试/184
 - 13.3 构建嵌入式Linux系统/186
 - 13.3.1 搭建系统环境/186
 - 13.3.2 编译u-boot/186
 - 13.3.3 编译内核与设备树/187
 - 13.3.4 制作根文件系统/188
 - 13.3.5 启动嵌入式Linux/192
 - 13.4 HDMI设计/193
 - 13.4.1 HDMI传输原理/193
 - 13.4.2 ADV7511芯片的相关控制信号/195
 - 13.4.3 设计过程/198
 - 13.5 OpenCV移植/203
 - 13.5.1 开发环境准备/203
 - 13.5.2 配置cmake/203
 - 13.5.3 OpenCV编译与安装/205
 - 13.5.4 OpenCV移植与ZedBoard测试/206
 - 13.6 基于OpenCV的树叶识别系统/207
 - 13.6.1 项目总览/208
 - 13.6.2 图像采集/208
 - 13.6.3 预处理/209
 - 13.6.4 特征提取/211
 - 13.6.5 分类决策/216
 - 13.6.6 总结/219
 - 13.7 基于OpenCV的人脸识别系统/220
 - 13.7.1 系统综述/220
 - 13.7.2 基于Haar特征和Adaboost算法的人脸检测/220
 - 13.7.3 系统设计与实现/222
 - 13.7.4 总结/226
 - 13.8 嵌入式Web服务器的移植与搭建/226
 - 13.8.1 嵌入式Web服务器介绍/226
 - 13.8.2 Boa服务器移植与配置/228
 - 13.8.3 Boa服务器部署与测试/230
 - 13.9 嵌入式网络摄像机的移植与搭建/233
 - 13.9.1 嵌入式网络摄像机/233
 - 13.9.2 mjpg-streamer的移植与架设/234
 - 13.10 FreeRTOS实时操作系统的应用/238
 - 13.10.1 FreeRTOS介绍/238
 - 13.10.2 FreeRTOS与ucOS- 的比较/239
 - 13.10.3 FreeRTOS在Zynq上的应用实例与分析/239
 - 13.10.4 基于FreeRTOS的Lwip/250

- 13.11 XADC的使用/250
 - 13.11.1 建立硬件工程/252
 - 13.11.2 软件工程设计/253
 - 13.11.3 程序分析/255
- 13.12 基于Zynq的部分可重配置/256
 - 13.12.1 可重配置系统介绍/256
 - 13.12.2 可重配置的开发流程/257
 - 13.12.3 小结/265
- 13.13 在Zynq上搭建Android简介/265
- 第14章 系统级设计案例/266
 - 14.1 电机控制系统/266
 - 14.1.1 双闭环控制器理论/266
 - 14.1.2 双闭环系统/267
 - 14.1.3 双闭环控制IP核说明/272
 - 14.1.4 硬件实现过程/275
 - 14.1.5 软件实现过程/285
 - 14.1.6 硬件平台测试/286
 - 14.2 智能家居健康平台/287
 - 14.2.1 智能家居健康平台简介/287
 - 14.2.2 EKG AFE模块硬件设计/287
 - 14.2.3 Night EKG Controller IP设计/292
 - 14.2.4 建立可运行Linux的完整系统/295
 - 14.2.5 Night EKG Controller的Linux驱动设计/297
 - 14.2.6 基于Qt的图形用户界面设计/299
 - 14.2.7 在ZedBoard上运行Qt程序/308
 - 14.2.8 实现软件开机自动运行/310
 - 14.3 高性能视频处理系统设计/311
 - 14.3.1 系统架构/312
 - 14.3.2 硬件架构设计/313
 - 14.3.3 软件架构设计/316
 - 14.3.4 利用Vivado HLS实现Sobel滤波硬件/318
 - 14.3.5 使系统在ZedBoard上运行/320
 - 14.4 智能小车系统开发/320
 - 14.4.1 智能小车系统结构/320
 - 14.4.2 运动控制设计/323
 - 14.4.3 Linux系统应用程序设计/326
 - 14.4.4 智能小车平台的后续拓展/333
- 第15章 如何获取资料和帮助/334
 - 15.1 如何获取Xilinx的技术文档/334
 - 15.1.1 DocNav介绍/334
 - 15.1.2 DocNav使用案例/334
 - 15.2 如何找到Zynq开发资料/336
 - 15.2.1 如何获取本书的最新例程/336
 - 15.2.2 如何获取Zynq开发资料/337
 - 15.2.3 如何获取ZedBoard文档与例程/337
 - 15.3 Xilinx网站资源导读/338
 - 15.3.1 序/338
 - 15.3.2 Xilinx软件介绍/338
 - 15.3.3 软件版本和软件更新/340

15.3.4 软件教程/341

15.3.5 硬件资料/343

15.3.6 参考资源/343

15.3.7 问题解决/344

附录A Xilinx开发套件版本14.1到14.3的主要升级变化/346

参考资料/353

章节摘录

版权页：插图：第11章PL和PS的接口技术详解 Zynq作为首款将高性能ARM Cortex A系列处理器与高性能FPGA在单芯片内紧密结合的产品，与其他独立ARM Cortex A9与Xilinx FPGA在单板上相比，其可以具有如下优点：设计成本降低；设计整体功耗降低；设计体积减小；设计风险降低；设计更灵活。为了实现这些优点，xilinx在设计Zynq时不仅要不同工艺特征的处理器和FPGA融合在一个芯片上并保证其良品率，更要设计高效的片内高性能处理器与FPGA之间互联通路。如果ARM Cortex A9与FPGA之间数据交互成了瓶颈，那么处理器与FPGA的性能优势都不能发挥出来，其他的优势也就无从谈起了。因此，如何设计高效的PL与PS数据交互通路是zynq芯片设计的重中之重，也是产品成败的关键之一。本章我们将主要介绍PL和PS的连接，揭示PL和PS之间的接口技术细节。11.1 PL和PS的接口在第5章中，我们已经初步介绍了Zynq器件上的系统级信号，其中也包括了PL和PS的接口信号，PL与PS的接口主要有两种类型：口功能接口。包括AXI、EMIO、中断、DMA流控制、时钟和调试接口。在FPGA开发人员设计PL模块的时候，就可以使用这些接口的信号，从而和PS进行数据交互。不同的信号接口用途不一样，应按需求使用。口配置接口。包括PCAP、SEU、配置状态信号和Program / Done / Init信号接口。这些信号的接口连接到PL内配置模块的固定逻辑上，给PS提供对PL的控制能力。对于硬件设计而言接触到的主要是AXI、EMIO、DMA、PCAP等接口，这些接口在第5章中我们已经都提过了。EMIO是Zynq提供的一种对MI0进行扩展的机制。其实就是将MI0放不下的PS外设接口，在PL上连接到外部引脚。当然，PL里的逻辑也能访问到这些“连接”，这些“连接”也可以访问PL的逻辑。如果将PL内的逻辑模块看作是系统外部的设备，也就是不需要通过总线和PS通信的设备，那么我们可以考虑使用EMIO接口和这种逻辑模块通信。PCAP是配置模块接口，在后续实验中我们也会用到。

《嵌入式系统软硬件协同设计实战指南》

编辑推荐

《嵌入式系统软硬件协同设计实战指南:基于Xilinx Zynq》可作为Zynq初学者、软硬件协同设计开发人员的参考用书，亦可作为大专院校嵌入式系统设计、片上系统设计、可编程逻辑器件等相关专业的教师和学生参考用书。

《嵌入式系统软硬件协同设计实战指南》

精彩短评

- 1、Xilinx 首款带有ARM处理器硬核的片子，国内的普及做的还是很快的。据说这是全球第一本介绍Zynq的图书。作为新手入门还是非常合适的，这本书里介绍了很多具体的实例，可以快速的上手Zynq系列的芯片。Xilinx 资深工程师编写，不可多得的权威之作。
- 2、帮同事买的，没说不好。
- 3、内容还可以，就是每一项都说个皮毛，看了还是没法学会啊
- 4、书上全是错误！！！不如不看
- 5、书刚开始看，里面都粗粗略过，不详细
- 6、看了前面部分，第十四章的系统工程就没看了。
了解下SOC还不错
- 7、深入浅出 从软件着手了解整个嵌入式系统 很对我的胃口
- 8、Joshua的神书，必须给好评！
- 9、咋评价人数不足...project都是从xilinx开发文档里找的，不过多了很多解释，对于提高姿势水平还是很吼的。
- 10、不知道能不能把里面的实验讲清楚！

《嵌入式系统软硬件协同设计实战指南》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com