

<<Xilinx All Programmable Zynq-7000 SoC设计指南>>

图书基本信息

书名：<<Xilinx All Programmable Zynq-7000 SoC设计指南>>

13位ISBN编号：9787302322221

10位ISBN编号：7302322228

出版时间：2013-5

出版时间：何宾 清华大学出版社 (2013-05出版)

作者：何宾

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：<http://www.tushu007.com>

<<Xilinx All Programma>>

前言

<<Xilinx All Programma>>

内容概要

<<Xilinx All Programma>>

作者简介

何宾，长期从事数字系统EDA方面教学与科研工作。

在全国进行大学生电子设计竞赛极力推进FPGA专题方面的培训工作，在EDA教学与科研应用方面积累了丰富的经验。

已出版相关图书《EDA原理及Verilog实现》、《EDA原理及VHDL实现》、《基于AXI4的可编程SOC系统设计》、《Xilinx FPGA设计权威指南》等10本深受读者喜欢的Xilinx FPGA图书。

<<Xilinx All Programma>>

书籍目录

第1篇Zynq—7000体系结构 第1章可编程SoC设计导论19 1.1可编程SoC系统设计基础19 1.1.1软核及硬核处理器19 1.1.2可编程SoC技术的发展20 1.1.3可编程SoC系统技术特点21 1.1.4可编程SoC设计流程21 1.1.5可编程SoC开发工具23 1.2XilinxZynq平台导论28 1.2.1XilinxZynq平台功能28 1.2.2处理系统PS特性30 1.2.3可编程逻辑PL特性35 1.2.4互联特性及描述37 1.2.5Zynq信号、接口和引脚39 1.3Zynq平台设计方法学46 1.3.1使用PL实现软件算法的优势46 1.3.2设计PL加速器47 1.3.3PL加速限制47 1.3.4降低功耗48 1.3.5实时减负49 1.3.6可重配置计算49 第2章AMBA协议规范50 2.1AMBA规范导论50 2.2AMBAAPB规范51 2.2.1AMBAAPB写传输51 2.2.2AMBAAPB读传输52 2.2.3AMBAAPB错误响应53 2.2.4操作状态54 2.2.5AMBA3APB信号54 2.3AMBAAHB规范56 2.3.1AMBAAHB结构56 2.3.2AMBAAHB操作58 2.3.3AMBAAHB传输类型60 2.3.4AMBAAHB猝发操作61 2.3.5AMBAAHB传输控制信号64 2.3.6AMBAAHB地址译码66 2.3.7AMBAAHB从设备传输响应67 2.3.8AMBAAHB数据总线70 2.3.9AMBAAHB传输仲裁71 2.3.10AMBAAHB分割传输76 2.3.11AMBAAHB复位79 2.3.12关于AHB数据总线的位宽79 2.3.13AMBAAHB接口设备80 2.4AMBAAXI4规范11 2.4.1AMBAAXI4概述错误！未定义书签。 2.4.2AMBAAXI4功能11 2.4.3AMBAAXI4互联结构20 2.4.4AXI4—Lite功能22 2.4.5AXI4—Stream功能23 第2篇Zynq—7000体系结构 第3章Zynq—7000应用处理单元27 3.1应用处理单元27 3.1.1基本功能27 3.1.2系统级视图29 3.2Cortex—A9处理器30 3.2.1中央处理器30 3.2.2L1高速缓存33 3.2.3存储器管理单元34 3.2.4接口37 3.2.5NEON38 3.2.6性能监视单元39 3.3侦听控制单元39 3.3.1地址过滤39 3.3.2SCU主设备端口40 3.4L2高速缓存40 3.4.1互斥L2—L1高速缓存配置42 3.4.2高速缓存替换策略43 3.4.3高速缓存锁定43 3.4.4使能 / 禁止L2高速缓存控制器44 3.4.5RAM访问延迟控制45 3.4.6保存缓冲区操作45 3.4.7在Cortex—A9和L2控制器之间的优化46 3.4.8预取操作47 3.4.9编程模型48 3.5片上存储器48 3.5.1片上存储器结构48 3.5.2片上存储器功能50 3.6APU接口57 3.6.1PL协处理接口57 3.6.2中断接口60 3.7APU内的TrustZone60 3.7.1CPU安全过渡61 3.7.2CP15寄存器访问控制61 3.7.3MMU安全性62 3.7.4L1缓存安全性62 3.7.5安全异常控制63 3.7.6CPU调试TrustZone访问控制63 3.7.7SCU寄存器访问控制63 3.7.8L2缓存中的TrustZone支持63 3.8应用处理单元复位64 3.8.1复位功能64 3.8.2复位后的APU状态65 3.9功耗考虑65 3.9.1待机模式66 3.9.2在L2控制器内的动态时钟门控66 3.10系统地址分配66 3.10.1地址映射66 3.10.2系统总线主设备68 3.10.3I / O外设68 3.10.4SMC存储器69 3.10.5SLCR寄存器69 3.10.6杂项PS寄存器70 3.10.7CPU私有总线寄存器70 3.11中断70 3.11.1中断环境71 3.11.2中断控制器的功能72 3.11.3编程模型76 3.12定时器77 3.12.1CPU私有定时器和看门狗定时器77 3.12.2全局定时器78 3.12.3系统看门狗定时器79 3.12.4三重定时器 / 计数器81 3.12.5I / O信号84 3.13DMA控制器85 3.13.1DMA控制器结构及特性85 3.13.2DMA控制器功能89 3.13.3外部信号99 3.13.4寄存器描述101 3.13.5.用于管理器和命令的指令集参考102 3.13.6编程模型参考103 3.13.7编程限制109 3.13.8DMACIP配置选项111 第4章Zynq—7000可编程逻辑资源112 4.1Zynq—7000可编程逻辑资源特性112 4.2可编程逻辑资源功能114 4.2.1CLB, Slice和LUT114 4.2.2时钟管理114 4.2.3块RAM115 4.2.4数字信号处理—DSPSlice116 4.2.5输入 / 输出117 4.2.6低功耗串行收发器118 4.2.7PCI—E模块119 4.2.8XADC (模拟—数字转换器) 120 4.2.9配置120 第5章系统互连结构122 5.1系统互连功能及特性122 5.1.1数据路径124 5.1.2时钟域125 5.1.3连接性126 5.1.4AXIID127 5.1.5寄存器概述128 5.2服务质量128 5.2.1基本仲裁128 5.2.2高级QoS128 5.2.3DDR端口仲裁129 5.3AXI_HP接口129 5.3.1AXI_HP接口结构及特点129 5.3.2接口数据宽度133 5.3.3交易类型134 5.3.4命令交替和重新排序135 5.3.5性能优化总结135 5.4AXI_ACP接口136 5.5AXI_GP接口137 5.6AXI信号总结137 5.7PL接口选择139 5.7.1使用通用主设备端口的Cortex—A9140 5.7.2通过通用主设备的PSDMA控制器 (DMAC) 141 5.7.3通过高性能接口的PLDMA142 5.7.4通过AXIACP的PLDMA143 5.7.5通过通用AXI从 (GP) 的PLDMA144 第6章系统公共资源特性及功能145 6.1时钟子系统145 6.1.1时钟系统结构及功能145 6.1.2CPU时钟域147 6.1.3时钟编程实例148 6.1.4时钟系统内生成电路结构149 6.2复位子系统153 6.2.1复位系统结构和层次154 6.2.2启动流程155 6.1.3复位的结果156 第7章Zynq调试和测试子系统158 7.1JTAG和DAP子系统158 7.1.1JTAG和DAP系统功能描述160 7.1.2JTAG和DAP系统I / O信号162 7.1.3编程模型163 7.1.4ARMDAP控制器164 7.1.5跟踪端口接口单元TPIU165 7.1.6XilinxTAP控制器165 7.2CoreSight系统结构及功能166 7.2.1CoreSight结构166 7.2.2CoreSight

<<Xilinx All Programma>>

功能167 第8章Zynq平台的启动和配置172 8.1Zynq平台启动和配置功能172 8.2外部启动要求173
 8.3BootROM175 8.3.1BootROM功能175 8.3.2BootROM头部178 8.3.3启动设备180 8.3.4BootROM多启动和
 启动分区查找184 8.3.5调试状态186 8.3.6BootROM后状态187 8.4器件配置接口189 8.4.1器件配置接口功
 能191 8.4.2器件配置流程193 8.4.3PL配置196 8.4.4寄存器集合197 第9章Zynq平台主要外设模块199
 9.1DDR存储器控制器199 9.1.1DDR存储器控制器接口及功能200 9.1.2AXI存储器端口接口202 9.1.3DDR
 核交易调度器204 9.1.4DDRC仲裁204 9.1.5DDR控制器PHY206 9.1.6DDR初始化和标定206 9.1.7纠错码207
 9.2静态存储器控制器208 9.2.1静态存储器控制器接口及功能209 9.2.2静态存储器控制器和存储器的信号
 连接210 9.3四—SPIFlash控制器212 9.3.1四—SPIFlash控制器功能213 9.3.2四—SPI控制器反馈时钟216 9.3.3
 四—SPIFlash控制器接口216 9.4SD / SDIO外设控制器218 9.4.1SD / SDIO控制器功能219 9.4.2SD / SDIO
 控制器传输协议221 9.4.3SD / SDIO控制器接口信号连接223 9.5通用输入输出控制器225 9.5.1通用输入输
 出GPIO接口及功能226 9.5.2通用输入输出GPIO中断功能227 9.6USB主机、设备和OTG控制器229
 9.6.1USB控制器接口及功能230 9.6.2USB主机操作模式233 9.6.3USB设备操作模式235 9.6.4USBOTG操作
 模式238 9.7吉比特以太网控制器239 9.7.1吉比特以太网控制器接口及功能240 9.7.2吉比特以太网控制器
 接口编程向导242 9.7.3吉比特以太网控制器接口信号连接246 9.8SPI控制器248 9.8.1SPI控制器的接口及功
 能249 9.8.2SPI控制器时钟设置规则251 9.9CAN控制器252 9.9.1CAN控制器接口及功能252 9.9.2CAN控制
 器操作模式255 9.9.3CAN控制器消息保存256 9.9.4CAN控制器接收过滤器256 9.9.5CAN控制器编程模
 型257 9.10UART控制器260 9.10.1UART控制器接口及功能261 9.11I2C控制器264 9.11.1I2C速度控制逻
 辑265 9.11.2I2C控制器的功能和工作模式265 9.12ADC转换器接口268 9.12.1ADC转换器功能269
 9.12.2ADC命令格式269 9.12.3供电传感器报警270 9.13PCI—E接口271 第10章Zynq平台描述规范273
 10.1Zynq平台文件描述规范功能集273 10.2微处理器硬件规范273 10.2.1通用微处理器硬件规范274
 10.2.2AXI系统微处理器硬件规范275 10.2.3Zynq—7000系统微处理器规范实例276 10.3微处理器外设规
 范280 10.3.1微处理器规范框架280 10.3.2总线接口规范283 10.3.3IO接口规范283 10.3.4选项规范284 10.3.5
 参数规范286 10.3.6端口规范293 10.3.7设计考虑295 10.4外设分析命令296 10.5黑盒定义298 10.6微处理器
 软件规范299 10.6.1微处理器软件规范格式299 10.6.2全局参数301 10.6.3实例指定参数301 10.6.4MDD
 / MLD指定参数302 10.6.5OS指定参数302 10.6.6处理器指定参数303 10.7微处理器库定义303 10.7.1库定义
 文件303 10.7.2MLD格式规范304 10.7.3MLD参数描述307 10.7.4设计规则检查309 10.7.5库产生309 10.8微处
 理器驱动定义309 10.8.1驱动定义文件310 10.8.2MDD格式规范310 10.9Xilinx板描述格式312 10.9.1XBD格
 式313 10.9.2属性命令313 10.9.3本地参数命令及子属性314 10.9.4本地端口命令及子属性315 10.9.5使
 用IO_INTERFACE关联IP315 10.9.6AXI系统XBD格式316 第11章高级综合工具HLS320 11.1高级综合工具
 结构320 11.1.1不同的命令对HLS综合结果的影响321 11.1.2从C中提取硬件结构323 11.2高级综合工具调
 度和绑定325 11.2.1高级综合工具调度325 11.2.2高级综合工具绑定326 11.3VivadoHLS工具的优势326
 11.4C代码的关键属性327 11.4.1函数328 11.4.2类型329 11.4.3循环330 11.4.4数组332 11.4.5端口333 11.4.6操
 作符334 11.5HLS内提供的用于时钟测量的术语335 第3篇Zynq—7000设计实战 第12章Zynq基本处理系统的
 的建立和运行338 12.1使用BSB向导生成Zynq基本系统338 12.1.1Zynq硬件系统的生成338 12.1.2生
 成HelloWorld应用工程346 12.1.3运行HelloWorld应用工程350 12.2生成和运行存储器测试工程352 12.2.1
 导入前面的XPS设计到SDK352 12.2.2生成存储器测试工程352 12.2.3运行存储器测试工程355 12.2.4调试存
 储器测试工程356 12.3生成和运行外设测试工程358 12.3.1导入前面的XPS设计到SDK359 12.3.2生成外设
 测试工程359 12.3.3运行外设测试工程364 第13章添加AXIIP到设计366 13.1设计原理366 13.2添加IP到系统
 设计367 13.2.1创建设计工程367 13.2.2添加GPIOIP到设计368 13.2.3添加AXITimerIP到设计370 13.2.4连接
 中断源到PS371 13.2.5通过EMIO将PS的GPIO连接到PL372 13.2.6添加约束到约束文件374 13.3使用SDK设
 计和实现应用工程374 13.3.1导入前面的XPS设计到SDK374 13.3.2生成应用工程374 13.3.3运行应用工
 程382 第14章基于定制IP实现简单嵌入式系统设计383 14.1创建设计工程384 14.2定制GPIOIP核384 14.2.1
 产生GPIOIP模版384 14.2.2基于模板构建完整GPIOIP391 14.3添加和连接AXI外设394 14.4添加约束到用
 户约束文件395 14.5使用SDK设计和实现应用工程396 14.5.1修改模板驱动函数396 14.5.2导入硬件设计
 到SDK工具396 14.5.3生成新应用工程397 14.5.4添加定制IP核软件驱动到设计397 14.5.5导入应用程序398
 14.5.6下载硬件比特流文件到FPGA400 14.5.7运行应用工程400 14.5.8使用XMD分析目标文件401 第15章
 基于定制IP实现复杂嵌入式系统设计402 15.1设计原理402 15.1.1VGAIP核的设计原理402 15.1.2移位寄存

<<Xilinx All Programma>>

器IP核的设计原理404 15.2创建设计工程405 15.3定制VGAIP核406 15.3.1产生VGAIP模版406 15.3.2基于模板构建完整VGAIP407 15.4定制移位寄存器IP核412 15.4.1产生shifterIP模版412 15.4.2基于模板构建完整shifterIP413 15.5添加和连接VGAIP核416 15.6添加和连接shifterIP核416 15.7添加约束到用户约束文件417 15.8使用SDK设计和实现应用工程418 15.8.1修改模板驱动函数418 15.8.2导入硬件设计到SDK工具419 15.8.3生成新的应用工程419 15.8.4添加定制IP核软件驱动到设计420 15.8.5编写应用程序421 15.8.6下载硬件比特流文件到FPGA424 15.8.7运行应用工程424 第16章软件和硬件协同调试系统426 16.1复制并打开设计工程426 16.2例化AXIChipscope核426 16.3导入硬件设计到SDK工具429 16.4启动ChipScopePro硬件调试器430 16.5执行H / S验证431 第17章Zynq平台配置和启动的实现433 17.1生成SD卡镜像文件并启动433 17.1.1SD卡接口433 17.1.2复制并打开前面的设计工程434 17.1.3创建第一级启动引导434 17.1.4创建SD卡启动镜像435 17.1.5从SD卡启动引导系统436 17.2生成QSPIFlash镜像并启动437 17.2.1QSPIFlash接口437 17.2.2创建QSPIFlash镜像438 17.2.3从QSPIFlash启动引导系统439 第18章基于ZynqHP从端口的数据传输实现440 18.1设计原理440 18.2创建设计工程441 18.3添加并配置AXICDMA到设计442 18.3.1添加AXICDMAIP和互联到设计442 18.3.2连接AXICDMA到设计443 18.3.3添加端口连接446 18.3.4分配地址空间447 18.4使用SDK设计和实现应用工程448 18.4.1软件应用的实现原理448 18.4.2导入硬件设计到SDK448 18.4.3创建新的软件应用工程449 18.4.4导入应用程序449 18.4.5下载硬件比特流文件到FPGA457 18.4.6运行应用工程457 第19章基于ZynqACP从端口的数据传输实现459 19.1设计原理459 19.2创建设计工程459 19.3配置PS端口460 19.3.1配置PS32位GPAXI主端口460 19.3.2配置PS的ACP从端口460 19.4添加并连接IP到设计461 19.4.1添加IP到设计461 19.4.2总线连接462 19.4.3端口连接463 19.4.4分配地址空间464 19.5使用SDK设计和实现应用工程464 19.5.1导入硬件设计到SDK465 19.5.2创建新的软件应用工程465 19.5.3导入应用程序465 19.5.4下载硬件比特流文件到FPGA469 19.5.5运行应用工程469 第20章XADC在Zynq平台上的应用471 20.1设计原理471 20.2创建设计工程472 20.3添加XADCIP到设计473 20.4添加约束到用户约束文件475 20.4.1Zedboard板上XADC接口475 20.4.2添加约束条件476 20.5使用SDK设计和实现应用工程476 20.5.1导入硬件设计到SDK工具476 20.5.2生成新的应用工程477 20.5.3下载硬件比特流文件到FPGA478 20.5.4运行应用工程478 第21章Ubuntu操作系统在Zynq平台上实现480 21.1Ubuntu操作系统环境搭建480 21.1.1安装虚拟机480 21.1.2在虚拟机上安装Ubuntu12.10485 21.1.3Linux和windows文件传输工具CuteFTP安装使用489 21.1.4Ubuntu相关环境和命令设置490 21.2u—boot原理及实现492 21.2.1u—boot结构492 21.2.2下载u—boot源码493 21.2.3u—boot配置与编译494 21.3内核概述及编译495 21.3.1内核结构495 21.3.2内核编译497 21.4设备树原理及实现497 21.4.1设备树功能497 21.4.2设备树数据格式497 21.4.3设备树的编译499 21.5文件系统原理及实现499 21.6打开设计工程500 21.7使用SDK设计生成软件工程501 21.7.1创建第一级引导启动代码502 21.7.2SD启动镜像的生成503 21.8验证Ubuntu操作系统的运行504 第22章 μ C / OS—III操作系统在Zynq平台上的实现505 22.1 μ c / OS—III操作系统概述506 22.2 μ C / OS—III操作系统环境构建509 22.3创建设计工程509 22.4建立基于 μ c / OS—III操作系统的软件应用工程511 22.4.1导入设计到SDK中511 22.4.2创建新的 μ C / OS—III设计工程511 22.4.3修改编译环境参数513 22.4.4重新编译设计工程514 22.5运行外设测试工程515 22.5.1配置硬件平台515 22.5.2配置运行环境515 22.6相关文件目录功能516 22.6.1App、BSP和Documentation文件夹517 22.6.2 μ C / CPU文件夹517 22.6.3 μ C / CSP文件夹517 22.6.4 μ C / LIB文件夹518 22.6.5 μ C / OS—III文件夹518 22.6.6用于 μ C / OS—III文件夹的XilinxBSP519 22.7基于 μ C / OS—III操作系统的关键工程文件分析520 22.7.1Main () 520 22.7.2AppTaskStart () 521 22.7.3AppTaskCreate () 523 22.7.4AppMutexCreate () 524 22.7.5AppTask1 () 524 22.7.6AppPrint () 525 22.7.7app_cfg.h526 第23章HLS在Zynq嵌入式系统设计中的应用527 23.1设计原理527 23.2基于HLS生成FIR滤波器527 23.2.1设计FIR滤波器528 23.2.2运行仿真和验证功能532 23.2.3设计综合534 23.2.4设计优化536 23.2.5运行CoSimulation537 23.2.6实现ISim软件下的仿真538 23.2.7使用指令创建Pcore540 23.2.8生成Pcore核542 23.3创建处理器系统543 23.3.1建立新的设计工程543 23.3.2修改处理器系统外设参数设置544 23.3.3复制pcore到当前工程544 23.3.4例化并连接生成的Pcore545 23.3.5添加用户约束到用户约束文件547 23.4使用SDK设计和实现应用工程548 23.4.1导入设计到SDK548 23.4.2生成应用工程548 23.4.3导入应用程序549 23.4.4验证硬件设计556 23.4.5运行应用工程556

<<Xilinx All Programma>>

章节摘录

版权页：插图：如果SCU缺失，它发布一个确认行填充到控制器。

确认和控制器内前面的预测读合并，使得控制器返回数据到L1缓存控制器，比L2缓存命中快。

如果SCU命中，在某个周期后或者当存在资源冲突时，L2自然地终止预测读。

当预测读结束时，L2控制器通知SCU，或者确认或者终止。

3.4.8预取操作 预取操作使得能够事先从存储器取缓存行，用于提高系统性能。

用户通过设置预取控制寄存器的28或者29比特位，使能预取特性。

当使能时，如果来自SCU的从端口接收到一个可缓存的读交易时，在随后的缓存行执行一个缓存查找。

预取控制寄存器的[4：0]比特提供了随后缓存行的地址。

如果发生缺失时，从外部存储器取出缓存行，并且分配到L2缓存。

默认时，预取偏置是5'b00000。

例如，如果S0接收到在地址0x100上的一个可缓存的读，预取0x120地址的缓存行。

预取下一个缓存行可能不产生性能优化。

在一些系统中，超前预取可以达到更好的性能。

通过将预取缓存行的地址设置到缓存行+1+偏置的地址，使得预取偏置能够达到更好的特性。

预取偏置的优化值取决于外部存储器读延迟和L1读发布能力。

预取机制不能跨越4KB的边界。

在控制主端口内，预取访问能使用大量的地址槽。

这可以防止对非预加载访问的服务，它将影响性能。

为了应对这个影响，控制器放弃预取访问。

通过使用预取控制寄存器的24比特位来控制它。

当使能时，如果在控制器主端口内的预取和非预取之间存在资源冲突，则放弃预取访问。

当所放弃的这些预取访问数据从外部存储器返回时，放弃它，并且不把它分配给L2高速缓存。

3.4.9编程模型 下面应用于L2缓存控制器内的寄存器：（1）通过存储器映射的寄存器集合控制缓存控制器。

这些寄存器的存储器区域必须在L1页表内，定义为强顺序或者设备存储器属性。

（2）必须保护所有寄存器的保留位；否则，可能发生设备未定义的行为。

（3）除非在相关的文字中进行了描述，所有寄存器支持读和写访问。

写操作用于更新寄存器的内容，读操作返回寄存器的内容。

（4）在处理所有对寄存器的写操作前，自动执行一个初始的同步操作。

作为一个例子，由下列寄存器操作构成一个典型的缓存控制器启动编程序列。

（1）使用一个读修改写，来设置全局配置。

写到辅助、标记RAM延迟、数据RAM延迟、预取和电源控制寄存器， 关联性和路大小； 用于RAM访问的延迟； 分配策略； 预取和供电能力。

<<Xilinx All Programmable>>

编辑推荐

《EDA工程技术丛书:Xilinx All Programmable Zynq-7000 SoC设计指南》首次论述Zynq-7000 SoC体系结构、程序设计及操作系统移植的方法与实践。

详尽介绍Zynq-7000 SoC的体系结构和相关生态系统，便于读者快速动手实践。

始终围绕软件和硬件协同设计的理念叙述，利于读者彻底掌握Zynq-7000 SoC的设计方法和技巧。

《EDA工程技术丛书:Xilinx All Programmable Zynq-7000 SoC设计指南》的编写得到了Xilinx及其合作伙伴的大力支持，藉其丰富的资源，反映出最新的设计技术水平。

《Xilinx All Programmable Zynq-7000 SoC设计指南》配套提供设计实例的完整设计代码和教学课件资源。

《EDA工程技术丛书:Xilinx All Programmable Zynq-7000 SoC设计指南》内容（含）可编程SoC设计AMBA协议规范Zynq-7000应用处理单元可编程逻辑资源系统互连结构系统公共资源特性及功能Zynq调试和测试系统Zynq平台的启动和配置Zynq平台主要外设模块Zynq平台描述规范高级综合工具HLS14个设计实例附赠内容源代码：第12章～第23章涉及实例的源代码文件教学课件：分成理论和实践两大部分

<<Xilinx All Programma>>

名人推荐

<<Xilinx All Programma>>

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:<http://www.tushu007.com>