

<<ARM体系结构与编程>>

图书基本信息

书名：<<ARM体系结构与编程>>

13位ISBN编号：9787121147739

10位ISBN编号：7121147734

出版时间：2012-6

出版时间：电子工业出版社 电子工业出版社 (2012-06出版)

作者：唐振明 编

页数：207

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>

<<ARM体系结构与编程>>

内容概要

目前基于ARM架构的嵌入式CPU在手持移动设备和通信等嵌入式领域占据绝对市场的状况，应重视学习嵌入式体系架构，首选是ARM体系架构。

编者唐振明认为：学习ARM体系架构不仅是概念的罗列、知识点的讲解及理论的贯穿，更重要的是在理解ARM设计理念的同时，要有好的配套实验或跟踪的案例，即强调“理论+实践”的学习方法。

《ARM体系结构与编程》就是围绕这一方法而编写的，主要内容包括ARM处理器概述、ARM体系结构、ARM编程模型、ARM微处理器的指令系统、ARM汇编语言程序设计、Bootloader启动代码分析和ARM RealView MDK集成开发环境，并针对ARM硬件寄存器和RISC指令集做了大量的配套实验及调试方法，最后在理解以上知识点的基础上对Bootloader进行了详细的分析。

《ARM体系结构与编程》适合作为计算机、软件工程、电气自动化及电子信息工程等大专生、本科生或研究生课程的教材。

<<ARM体系结构与编程>>

书籍目录

第1章 ARM处理器概述 1.1 嵌入式处理器简介 1.1.1 嵌入式处理器分类 1.1.2 嵌入式操作系统 1.1.3 嵌入式处理器评价指标 1.2 什么是ARM 1.2.1 ARM的概念 1.2.2 ARM公司发迹史 1.3 ARM体系结构的命名规则 1.4 ARM系列处理器简介 1.4.3 ARM9E系列 1.4.4 ARM10系列 1.4.5 ARM11系列 1.4.6 SecurCore系列 1.4.7 ARM Cortex-A8处理器 1.5 ARM处理器的技术优势及其应用 1.6 本章小结 1.7 课后练习 第2章 ARM体系结构 2.1 ARM体系结构的特点 2.1.1 RISC设计思想的体现 2.2 ARM流水线 2.2.1 ARM流水线的概念与原理 2.2.2 流水线的分类 2.2.3 3级流水线ARM组织 2.2.4 5级流水线ARM组织 2.2.5 6级流水线ARM组织 2.2.6 影响流水线性能的因素 2.3 ARM存储器 2.4 I/O管理 2.5 ARM开发调试方法 2.5.1 指令集模拟器 2.5.2 驻留监控软件 2.5.3 JTAG仿真调试 2.5.4 基于ULINK在线仿真器 2.6 本章小结 2.7 课后练习 第3章 ARM编程模型 3.1 数据类型 3.1.1 ARM的基本数据类型 3.1.2 浮点数据类型 3.1.3 存储器大/小端 3.2 处理器工作模式 3.3 ARM寄存器组织 3.3.1 通用寄存器 3.3.2 程序状态寄存器 3.4 异常中断处理 3.4.1 异常种类 3.4.2 异常优先级 3.4.3 处理器模式和异常 3.4.4 异常响应流程 3.4.5 从异常处理程序中返回 3.5 本章小结 3.6 课后练习 第4章 ARM微处理器的指令系统 4.1 ARM微处理器的指令集概述 4.1.1 ARM微处理器的指令的分类与格式 4.1.2 指令的条件域 4.2 ARM指令集 4.2.1 数据处理指令 4.2.2 移位指令 4.2.3 乘法指令与乘加指令 4.2.4 批量数据加载/存储指令 4.2.5 跳转指令 4.2.6 程序状态寄存器访问指令 4.2.7 加载/存储指令 4.2.8 协处理器指令 4.2.9 异常产生指令 4.3 ARM指令的寻址方式 4.3.1 立即寻址 4.3.2 寄存器寻址 4.3.3 寄存器间接寻址 4.3.4 基址变址寻址 4.3.5 多寄存器寻址 4.3.6 相对寻址 4.3.7 堆栈寻址 4.4 Thumb指令及应用 4.5 本章小结 4.6 课后练习 第5章 ARM汇编语言程序设计 5.1 ARM汇编器所支持的伪操作 5.1.1 符号定义Symbol Definition伪操作 5.1.2 数据定义Data Definition伪操作 5.1.3 汇编控制Assembly Control伪操作 5.1.4 信息报告Reporting伪操作 5.1.5 指令集选择Instruction Set Selection伪操作 5.1.6 其他伪操作 5.2 ARM汇编器所支持的伪指令 5.2.1 ADR伪指令 5.2.2 ADRL伪指令 5.2.3 MOV32伪指令 5.2.4 LDR伪指令 5.3 汇编语言文件格式 5.3.1 ARM汇编语言语句格式 5.3.2 ARM汇编语言中的符号 5.3.3 汇编语言程序中的表达式和运算符 5.3.4 汇编语言预定义寄存器和协处理器 5.3.5 汇编语言的程序结构 5.3.6 汇编语言子程序调用 5.4 ARM汇编语言与C语言混合编程 5.4.1 在C语言中内嵌汇编语言 5.4.2 在C语言中调用汇编语言的函数 5.4.3 在汇编语言中调用C语言的函数 5.5 本章小结 5.6 课后练习 第6章 Bootload启动代码分析 6.1 汇编基础 6.2 启动代码功能模块分解 6.2.1 程序的入口地址 6.2.2 看门狗及中断的禁止 6.2.3 系统时钟初始化 6.2.4 初始化内存控制器 6.2.5 系统堆栈的初始化 6.2.6 建立中断向量表 6.2.7 跳转到C语言入口 6.3 Bootload实验部分 6.3.1 实验环境 6.3.2 实验步骤 6.3.3 实验总结 6.4 Bootload扩展部分—U-Boot分析 6.4.1 Bootloader的引导方式 6.4.2 Bootloader的种类 6.4.3 U-Boot源码结构 6.4.4 U-Boot的编译 6.4.5 U-Boot的移植 6.4.6 添加U-Boot命令 6.4.7 U-Boot的调试 6.4.8 U-Boot与内核的关系 6.4.9 U-Boot的常用命令 6.4.10 U-Boot的环境变量 6.4.11 使用U-Boot 6.5 本章小结 6.6 课后练习 第7章 ARM RealView MDK集成开发环境 7.1 RealView MDK突出特性 7.2 MDK功能介绍 7.2.1 Vision4 IDE 7.2.2 Vision4 IDE主要特性 7.2.3 启动代码配置向导 7.2.4 Vision4设备模拟器 7.2.5 性能分析器 7.2.6 RealView编译器 7.2.7 MicroLib 7.2.8 RealView Real-Time Library RealView RTL实时库 7.2.9 ARM软件开发工具解决方案 7.3 RealView MDK的使用 7.3.1 Vision4的安装 7.3.2 创建Vision工程 7.4 Keil MDK编译器与ULINK2使用 7.4.1 ULINK2概述 7.4.2 ULINK2与MDK的链接使用 7.5 Keil MDK编译器与J-LINK使用 7.5.1 J-LINK概述 7.5.2 J-LINK与MDK的链接使用 7.6 Keil MDK编译器与H-JTAG使用 7.6.1 H-JTAG概述 7.6.2 H-JTAG 调试结构 7.6.3 H-JTAG的安装 7.6.4 H-JTAG配置 7.6.5 MDK的安装与设置 7.6.6 调试 7.7 Keil开发工具链 7.7.1 用UltraEdit查看和编辑程序源文件 7.7.2 Keil MDK生成BIN过程 7.7.3 ARM工具链准备实验 7.7.4 armasm汇编器的使用 7.7.5 armlink链接器的使用 7.7.6 armcc编译器的使用 7.7.7 FromELF实用工具实验 7.8 本章小结 7.9 课后练习 参考文献

<<ARM体系结构与编程>>

章节摘录

版权页：插图：6.4.1 Bootloader的引导方式 Linux系统是通过Bootloader引导启动的。

加电后，就要执行Bootloader来初始化系统。

系统加电或复位后，所有CPU都会从某个地址开始执行，这是由处理器设计决定的。

例如，X86的复位向量在高地址端，ARM处理器在复位时从地址0x00000000取第一条指令。

嵌入式系统的开发板都要把板上ROM或Flash映射到这个地址。

因此，必须把Bootloader程序存储在相应的Flash位置。

系统加电后，CHU将首先执行它。

主机和目标机之间一般有串口可以连接，Bootloader软件通常会通过串口来输入输出。

例如，输出出错或者执行结果信息到串口终端，从串口终端读取用户控制命令等。

Bootloader启动过程通常是多阶段的，这样既能提供复杂的功能，又有很好的可移植性。

例如，从Flash启动的Bootloader多数是两阶段的启动过程。

从后面U—Boot的内容可以详细分析这个特性。

大多数。

Bootloader都包含2种不同的操作模式：本地加载模式和远程下载模式。

这两种操作模式的差别仅对于开发人员才有意义，也就是不同启动方式的使用。

从最终用户的角度看，Bootloader的作用就是用来加载操作系统，并不存在所谓的本地加载模式与远程下载模式的差别。

因为Bootloader的主要功能是引导操作系统启动，所以我们详细讨论一下各种启动方式的特点。

(1) 网络启动方式 这种方式开发板不需要配置较大的存储介质，与无盘工作站有点类似。

但是使用这种启动方式之前，需要把Bootloader安装到板上的EPROM或者Flash中。

Bootloader通过以太网接口远程下载Linux内核映像或者文件系统。

使用这种方式也有前提条件，就是目标板需有串口、以太网接口或者其他连接方式。

串口一般可以作为控制台，同时可以用来下载内核映像和RAMDISK文件系统。

串口通信传输速率过低，不适合用来挂接NFS文件系统。

所以以太网接口成为通用的互联设备，一般的开发板都可以配置10M以太网接口。

对于PDA等手持设备来说，以太网的RJ—45接口显得大了些，而USB接口，特别是USB的迷你接口，尺寸非常小。

对于开发的嵌入式系统，可以把USB接口虚拟成以太网接口来通信。

这种方式在开发主机和开发板两端都需要驱动程序。

另外，还要在服务器上配置启动相关网络服务。

Bootloader下载文件一般都使用TFTP网络协议，还可以通过DHCP的方式动态配置IP地址。

DHCP / BOOTP服务为Bootloader分配IP地址，配置网络参数，然后才能够支持网络传输功能。

如果Bootloader可以直接设置网络参数，就可以不使用DHCP，TFTP服务为Bootloader客户端提供文件下载功能，把内核映像和其他文件放在 / tftpboot目录下。

这样Bootloader可以通过简单的TFTP协议远程下载内核映像到内存。

大部分引导程序都能够支持网络启动方式。

例如，BIOS的.PXE (Preboot Execution.Environment) 功能就是网络启动方式；U—Boot也支持网络启动功能。

<<ARM体系结构与编程>>

编辑推荐

《ARM体系结构与编程》适合作为计算机、软件工程、电气自动化及电子信息工程等大专生、本科生或研究生课程的教材。

<<ARM体系结构与编程>>

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>