

■ PHILIPS 大学计划指定教材

Modeling
Solutions

ARM

与嵌入式技术

■ 孙红波 等编著 张 林 徐光峰 主审

Design
Consulting

IP
Solutions

Development
Tools and
Systems

Support
Services



<http://www.phei.com.cn>



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

PHILIPS 大学计划指定教材

ARM 与嵌入式技术

孙红波 陶 品 李 莉 刘 瑾 等编著

张 林 徐光峰 主审

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书是学习 ARM 基础知识、嵌入式技术的入门性教材。

全书内容分三部分：第一部分是 ARM 基础内容，介绍 ARM 的基本概念、基本结构、基本寻址方式和基本指令，通过 PHILIPS 公司的 LPC2200 系列芯片对 ARM 微控制器进行详细介绍。第二部分是编程技术，配合 LPC2292 实验系统给出若干实验，如实现串行通信、A/D 转换、LCD 驱动、键盘扩展、定时器、GUI 和 USB 驱动等。第三部分主要介绍嵌入式 Linux 的基础知识和嵌入式 Linux 开发环境的配置，uClinux 操作系统相关的知识，如任务管理、任务间通信、内存与 I/O 管理，以及中断处理机制等。

全书侧重基本概念，内容丰富，语言流畅，通俗易懂。本书备有相应习题和机考系统。本书在编写中，尽量减少对其他专业的依托。它既可作为各高等院校有关嵌入式理工科大学本科生的教材，也可供从事 ARM 及嵌入式 Linux 的工程人员学习和参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

ARM 与嵌入式技术/孙红波等编著. —北京：电子工业出版社，2006.3

ISBN 7-121-02368-7

I. A… II. 孙… III. ①微处理器，ARM②Linux 操作系统 IV. ①TP332②TP316.89

中国版本图书馆 CIP 数据核字 (2006) 第 018437 号

责任编辑：孙延真

印 刷：北京牛山世兴印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：17 字数：442 千字

印 次：2006 年 3 月第 1 次印刷

印 数：5000 册 定价：29.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：(010) 68279077。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

时间进入 21 世纪，人们关于处理器结构的争论不再幼稚，这又一次体现了时事造英雄。在第二代无线通信技术的带领下，ARM 技术得到了充分的应用。可编程门阵列（FPGA）的出现，使半导体设计技术从硬核时代进入软核时代。半导体行业设计产业门槛迅速降低，IC 设计进入了基于知识产权（IP）库的设计阶段，硬件设计采用软件编程实现，集成电路设计产业不再是少数几个行业巨头的专利。芯片设计公司利用高效的 IP 内核（Intellectual Property Core），根据需要，推出专用芯片产品。

一个专用系统的设计包括硬件和软件两部分。硬件可以通过控制器与其他电路组合，或者利用 VHDL 调用 IP 内核，编译下载到 FPGA 等可编程器件中；软件编程可以用汇编语言或 C 语言等编写。生活在后 PC 时代的人们，越来越多的产品需要网络功能。需要操作更简单，需要功能更强大，需要开发周期短，需要代码容错性好等，满足这些开发需求的捷径就是使用有源码可以裁减、稳定运行的操作系统。操作系统是一个可以调用的高效率程序集，它能够提供操作过程的协议或准则，但是在 8 位控制器上运行操作系统难以做到，只有真正地掌握嵌入式操作系统才能随心所欲。

本书包括了当前硬件、软件方面的两个流行趋势，一个是 32 位微控制器 ARM，一个是嵌入式操作系统 Linux。Linux 提供了强大的操作系统功能和丰富的应用软件。开发者和学习者可以从 Internet 上下载 Linux 源代码和许多 Linux 的应用程序，并根据需要对源代码进行修改，减少程序的编写和调试负担。

本书作为 PHILIPS 公司与深圳市英蓓特信息技术有限公司大学计划的组成部分，在编写上注重基本概念和基本方法，内容丰富，语言流畅。在编写中，尽量减少对其他专业的依托和非基础性内容。希望本书能够作为理工科高等院校嵌入式系统的教材，也希望为从事 ARM 及嵌入式 Linux 开发的工程人员起到学习和参考作用。

本书由孙红波、陶品、李莉、刘瑾、刘小春等共同编写，张林、徐光峰主审，林惠民、周轩、余跃、史利坤、任海英、郭乃豪、梁颖、章蔚、李晓莉等为本书提供了部分实例和素材。

感谢 PHILIPS 公司、英蓓特公司的大力协作，感谢电子工业出版社的大力支持。

由于本书作者能力有限，编写中的用词偏差、写作错误等方面问题在所难免，恳请读者给予批评指正。有关 PHILIPS 大学合作计划的申请可在英蓓特公司网站（www.embedinfo.com）查询或与 E-mail: P_E_PLAN@163.com 联系。

作 者

2005 年 12 月

目 录

第 1 章 ARM 处理器概述	(1)
1.1 ARM 技术的发展	(1)
1.2 RISC 微处理器的体系结构及其特点	(2)
1.3 ARM 微处理器的指令系统	(3)
1.4 ARM 处理器系列	(3)
1.4.1 ARM7 微处理器系列	(4)
1.4.2 ARM9 微处理器系列	(4)
1.4.3 ARM9E 微处理器系列	(4)
1.4.4 ARM10E 微处理器系列	(5)
1.4.5 SecurCore 微处理器系列	(5)
1.4.6 Xscale 处理器	(6)
1.5 ARM 微处理器的应用系统开发	(6)
习题 1	(6)
第 2 章 ARM 体系结构	(7)
2.1 ARM7TDMI 程序员模型	(7)
2.1.1 内核概述	(7)
2.1.2 译码器与控制逻辑	(8)
2.1.3 运算器	(9)
2.1.4 内核接口信号	(9)
2.2 工作状态与工作模式	(13)
2.2.1 工作状态	(13)
2.2.2 ARM 的工作模式	(14)
2.3 ARM 和 Thumb 状态的寄存器组织	(15)
2.3.1 通用寄存器	(15)
2.3.2 Thumb 状态下的寄存器组织	(16)
2.3.3 程序计数器	(17)
2.3.4 程序状态寄存器	(17)
2.3.5 连接寄存器	(19)
2.3.6 堆栈指针寄存器	(19)
2.4 存储器组织结构	(20)
2.4.1 数据类型和存储数据格式	(20)
2.4.2 数据总线类型	(21)
2.4.3 ARM7TDMI 总线访问周期	(24)
2.4.4 指令预取与代码自修改	(26)
2.4.5 AMBA 接口 (Advanced Microcontroller Bus Architecture)	(27)

2.5	异常 (Exceptions)	(27)
2.5.1	异常类型及其处理过程	(27)
2.5.2	异常优先级 (Exception Priorities) 和异常向量 (Exception Vectors)	(29)
2.5.3	应用程序中的异常响应	(29)
2.5.4	异常返回	(30)
2.6	复位	(31)
2.7	协处理器接口	(31)
2.7.1	协处理器概述	(31)
2.7.2	ARM7TDMI 与协处理器的协同关系	(32)
2.7.3	协处理器使用中的若干问题	(32)
习题 2		(33)
第 3 章	ARM 指令的特点和寻址方式	(34)
3.1	ARM 指令特点	(34)
3.2	ARM 指令的寻址方式	(35)
习题 3		(40)
第 4 章	ARM 指令系统概述	(42)
4.1	分支指令	(44)
4.2	ARM 的数据处理指令	(45)
4.2.1	算术运算类指令	(45)
4.2.2	逻辑运算类指令	(47)
4.2.3	比较类指令	(49)
4.2.4	测试类指令	(49)
4.2.5	传送类指令	(50)
4.2.6	乘法类指令	(51)
4.3	异常处理指令	(53)
4.4	ARM 的存储器加载/存储指令	(54)
4.4.1	单寄存器加载	(55)
4.4.2	单寄存器存储	(57)
4.4.3	多寄存器加载和多寄存器存储	(58)
4.4.4	寄存器交换	(59)
4.5	协处理器指令	(60)
4.6	ARM 状态寄存器访问指令	(62)
习题 4		(63)
第 5 章	ARM 汇编语言程序设计	(64)
5.1	ARM 汇编语言伪指令	(64)
5.1.1	符号定义伪指令	(64)
5.1.2	空间分配伪指令	(66)
5.1.3	汇编控制伪指令	(68)
5.1.4	格式控制伪指令	(70)
5.1.5	ARM 伪指令	(71)

5.1.6 Thumb 伪指令	(73)
5.2 ARM 汇编语言语句格式	(73)
5.2.1 ARM 汇编语言中的符号	(74)
5.2.2 汇编语言程序中的表达式和运算符	(76)
5.3 ARM 语言程序结构	(77)
5.4 简单的 ARM 程序设计	(78)
5.4.1 简单的 ARM 程序	(78)
5.4.2 子程序调用程序设计	(79)
5.4.3 分支程序设计	(79)
5.4.4 查表与散转程序设计	(80)
5.4.5 数据串拷贝程序的设计	(81)
习题 5	(83)
第 6 章 ARM-Thumb 交互工作	(84)
6.1 交互工作原理	(84)
6.1.1 交互工作的必要性	(84)
6.1.2 交互工作的切换指令	(84)
6.1.3 与状态切换有关的伪指令	(85)
6.2 交互程序	(85)
6.2.1 简单的交互程序设计的例子	(85)
6.2.2 ARM 指令头的例子	(86)
6.2.3 交互子程序调用	(86)
6.3 ARM v5T 扩展	(88)
6.4 交互子程序和 Veneer	(89)
6.4.1 交互编译选项	(89)
6.4.2 Veneer	(90)
习题 6	(90)
第 7 章 LPC2200 系列原理与功能	(92)
7.1 概述	(92)
7.1.1 特性	(92)
7.1.2 结构	(93)
7.2 管脚分布及信号描述	(95)
7.3 存储系统	(101)
7.3.1 存储器映射	(101)
7.3.2 存储器加速模块 (MAM)	(105)
7.3.3 外部存储器控制器 (EMC)	(107)
7.4 时钟控制电路	(110)
7.4.1 晶体振荡器	(110)
7.4.2 PLL 及相关寄存器	(110)
7.4.3 VPB 分频器及相关寄存器	(113)
7.5 复位和功率控制	(113)

7.5.1	复位	(113)
7.5.2	功率控制	(114)
7.6	中断	(116)
7.6.1	向量中断控制器 VIC	(116)
7.6.2	外部中断	(121)
7.7	其他片上外围模块	(122)
习题 7		(127)
第 8 章	LPC2200 系列开发平台	(128)
8.1	概述	(128)
8.2	硬件电路的设计及其注意事项	(129)
8.2.1	电源、晶振及复位信号的处理	(129)
8.2.2	JTAG 接口电路的正确设计	(130)
8.2.3	模式管脚的连接方法	(131)
8.2.4	存储器扩展	(131)
8.2.5	UART 扩展	(134)
8.2.6	CAN 总线扩展	(135)
8.2.7	USB 接口	(135)
8.2.8	硬件系统设计需要考虑的问题	(136)
习题 8		(136)
第 9 章	集成开发环境 Embest IDE	(137)
9.1	Embest IDE 概述	(137)
9.1.1	Embest IDE 开发环境	(137)
9.1.2	Embest IDE 安装	(138)
9.1.3	Embest IDE 主框架窗口	(139)
9.2	Embest IDE 的编辑	(139)
9.3	工程管理	(140)
9.3.1	工程管理的操作	(141)
9.3.2	工程配置	(142)
9.4	编译、汇编和链接	(144)
9.4.1	GNU Tools for ARM 选项	(144)
9.4.2	ARM Build Tools 选项	(147)
9.4.3	工程文件夹的编译设置	(147)
9.4.4	工程编译、汇编和链接	(148)
9.5	工程调试	(148)
9.5.1	调试设置	(149)
9.5.2	调试	(151)
9.5.3	Embest IDE 附带工具介绍	(154)
9.6	GNU 编译环境下程序的编写	(155)
9.6.1	移植 SDT 的汇编程序	(155)
9.6.2	链接脚本文件	(157)

习题 9	(159)
第 10 章 ARM 高级语言程序设计基础	(160)
10.1 C 语言与汇编的混合编程	(160)
10.1.1 ARM 过程调用标准 ATPCS	(160)
10.1.2 内嵌汇编器	(162)
10.1.3 C 语言和 ARM 汇编语言间相互调用	(166)
10.2 基本 I/O 程序 (含启动代码)	(168)
10.2.1 启动代码	(168)
10.2.2 基本 I/O 主程序	(173)
10.3 SPI 通信程序	(175)
10.4 串口通信程序	(176)
10.5 A/D 程序	(177)
10.6 中断程序	(178)
习题 10	(179)
第 11 章 嵌入式 Linux	(180)
11.1 嵌入式操作系统概述	(180)
11.1.1 操作系统的作用	(180)
11.1.2 操作系统的主要任务	(181)
11.1.3 嵌入式操作系统的特点	(181)
11.1.4 典型的嵌入式实时操作系统	(184)
11.2 嵌入式 Linux 操作系统内核及其特点	(186)
11.2.1 GNU 及自由软件开发模式	(186)
11.2.2 Linux 内核的引入	(187)
11.2.3 Linux 内核的特点	(187)
11.2.4 标准 Linux 内核的变种	(188)
11.3 Linux 与 uClinux	(190)
11.3.1 uClinux 名称释义	(191)
11.3.2 MMU 工作机制	(191)
11.3.3 uClinux 与普通 Linux 系统的异同	(192)
11.3.4 uClinux 的发展前景	(193)
11.4 构建一个嵌入式 Linux 系统	(194)
11.4.1 Linux 内核与 Linux 系统	(194)
11.4.2 嵌入式 Linux 系统的特点	(195)
11.4.3 嵌入式系统的开发	(195)
11.4.4 开发一个嵌入式 Linux 系统	(197)
习题 11	(198)
第 12 章 嵌入式 Linux 内核移植和定制	(199)
12.1 Linux 内核代码的基本结构	(199)
12.2 Linux 内核的配置与编译	(200)
12.2.1 内核配置	(201)

12.2.2 应用程序配置	(203)
12.3 嵌入式 Linux 的引导过程	(204)
12.4 内核移植的一般步骤	(207)
12.5 根文件系统的分析与构造	(210)
12.5.1 根文件系统的组成	(210)
12.5.2 文件系统的选择	(210)
12.5.3 根文件系统的生成	(211)
习题 12	(211)
第 13 章 嵌入式 Linux 应用程序的开发	(212)
13.1 概述	(212)
13.2 嵌入式应用程序的开发过程	(212)
13.3 交叉编译环境的建立	(213)
13.3.1 Cygwin 开发环境	(214)
13.3.2 交叉编译工具 gcc	(215)
13.4 多程序的编译工具 make	(218)
13.5 交叉调试工具 gdb	(221)
13.5.1 gdb 的使用	(221)
13.5.2 gdb 的基本命令	(222)
13.5.3 gdb 的工作过程	(223)
13.5.4 关于 core 文件	(224)
13.5.5 调试运行中的进程	(225)
习题 13	(225)
第 14 章 嵌入式 Linux 驱动程序的开发	(226)
14.1 概述	(226)
14.2 将设备驱动程序纳入到文件系统中	(228)
14.3 设备驱动程序的编程框架及函数调用接口	(229)
14.3.1 设备驱动程序开发的特性与共性	(229)
14.3.2 程序框架及解释	(229)
14.4 设备驱动程序的实例	(236)
14.5 设计自己的驱动程序	(244)
14.5.1 设备的功能	(244)
14.5.2 设备的实现	(245)
14.5.3 其他驱动程序相关技术	(255)
14.5.4 动态加载设备驱动程序	(257)
习题 14	(258)
参考文献	(259)

第 1 章 ARM 处理器概述

由于信息技术和网络技术的高速发展，嵌入式产品广泛地渗透到日常生活、科学研究和军事技术的领域。从家里的洗衣机、电冰箱，到交通工具汽车，以及办公室里的远程会议系统，都在使用着嵌入式产品。

嵌入式系统通常包括硬件和操作系统两部分，硬件是构成软件的基本运行环境。嵌入式操作系统一般采用了微内核结构，内核只提供基本的功能，比如：任务的调度、任务之间的通信与同步、内存管理、时钟管理等。对于应用组件，用户可以根据自己的需要选用如网络功能、文件系统、GUI 系统等。

嵌入式操作系统大多数支持多任务。任务的调度采用抢占式调度、不可抢占式调度和时间片轮转调度。目前多数嵌入式操作系统对不同优先级的任务采用基于优先级的抢占式调度法，对相同优先级的任务则采用时间片轮转调度法。

嵌入式处理器通常将 CPU、少量的 RAM、Flash 和其他接口封装在一片集成电路中，嵌入式处理器可以分为嵌入式微处理器（MPU）、嵌入式微控制器（MCU）、嵌入式 DSP（Digital Signal Processor）。

大多数嵌入式系统处理器中采用的是实存储器管理策略，而不使用虚拟内存管理技术，程序中使用的地址都是物理地址，直接将地址送到地址总线上输出，开发人员需要参与系统的内存管理。

1.1 ARM 技术的发展

20 世纪 80 年代，半导体行业产业链刚刚出现分工，设计企业与生产企业分离，其流行做法是创办没有生产线的公司，如 Cirrus Logic。

ARM（Advanced RISC Machines）是一家坐落在英国剑桥的电子公司，成立于 1990 年 11 月，由苹果电脑、Acom 电脑集团和 VLSI Technology 共同组建。

众所周知，世界半导体产业（包括设计和生产）的中心在美国，在半导体制造和市场方面，英国没有什么优势。ARM 如何在市场中求生存是一个严重的问题。

ARM 起初进行 GPS、音乐播放装置和游戏机等的设计。不得不提的是移动通信行业的发展，给 ARM 带来巨大的机会。20 世纪 90 年代初，美国 TI（德州仪器）公司与诺基亚合作，把模拟蜂窝电话改变为数字蜂窝电话。模拟手机内部采用的是 16 位 CPU 技术，处理能力有限，需要 32 位的 CPU 处理。TI 认为：项目从头立项，再进行研发到投产，成功率只有 25%；如购买技术进行再设计研发成功率能够达到 70%，这样做风险低，开发速度也快。说来也巧，ARM 正在为其设计紧凑、功耗低 32 位 CPU 微处理器的技术寻找市场。TI 采用 ARM 的授权技术，实现了手机从模拟技术向数字技术的转变。

ARM 在市场方面采用非常灵活的政策。例如，诺基亚认为 32 位 CPU 要求的存储指令代码量太大，系统设计不能忍受。ARM 根据这个需求开发了 Thumb 技术：32 位 CPU 核配合 16

位的指令集技术。从 1995 年开始 ARM 的授权业务发展起来了。

ARM 在不利的环境下创造了 chipless 生产模式。ARM 既不生产芯片,也不设计芯片,他们以高效的 IP (Intellectual Property) 内核为产品,出售芯片设计技术,授权给其他半导体公司使用。IP Core 是没有任何物理意义上的硬件或软件,设计公司可以在 ARM 技术的基础上添加自己的设计并推出芯片产品。采用 ARM 技术知识产权 (IP) 核的微处理器,即我们通常所说的 ARM 微处理器,已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场,ARM 技术正在逐步渗入到生活的各个方面。

ARM 将其技术授权给世界上许多著名的半导体、软件和 OEM 厂商,每个厂商得到的都是一套 ARM 相关技术及服务。目前,几十家大的半导体公司都使用 ARM 公司的授权,其中包括 Intel、PHILIPS 等一系列知名公司。利用合伙关系,ARM 很快成为许多全球性 RISC 标准的缔造者。

ARM 是第一款面向低价位市场的 RISC 架构微处理器。目前,ARM 已经成为嵌入式微处理器的代名词。

1.2 RISC 微处理器的体系结构及其特点

传统的计算机结构是 CISC (Complex Instruction Set Computer, 复杂指令集计算机) 结构。处理器在发展过程中,一方面不断引入新的指令,设置一些功能复杂的指令,把一些原来由软件实现的常用功能改用硬件的指令系统实现,以提高计算机执行速度;另一方面,为了保持兼容,一些老的指令又不能废弃,因而计算机的体系结构会越来越复杂。根据统计,CISC 指令集中的指令,大约只有 20% 的指令会在整个 80% 程序代码中反复使用,而其余的 80% 的指令只在 20% 程序代码中使用,这就是所谓的“二八”原理。

1979 年美国加州大学伯克利分校提出了 RISC (Reduced Instruction Set Computer, 精简指令集计算机) 的概念,基本思想是尽量简化计算机指令功能,只保留那些功能简单、能在一个节拍内执行完成的指令,而把较复杂的功能用一段子程序实现。

RISC 思想的精华就是通过简化计算机指令功能,简化计算机指令格式,使指令的平均执行周期减少,同时大量使用通用寄存器,来提高计算机的工作主频,提高程序执行的速度。RISC 结构优先选取使用频率最高的简单指令,避免复杂指令;指令长度固定,减少指令格式和寻址方式的种类;在指令译码上以硬布线逻辑为主,不用或少用微程序。通常,RISC 计算机的速度是同等 CISC 计算机的 3 倍左右。RISC 体系结构应具有如下特点:

- 采用固定长度的指令格式;
- 使用单周期指令,便于流水线操作执行;
- 使用大量寄存器,数据处理指令只对寄存器进行操作;
- 采用加载/存储指令批量传输数据,以提高数据的传输效率;
- 所有的指令都可根据前面的执行结果决定是否被执行,从而提高指令的执行效率;
- 在一条数据处理指令中,同时完成逻辑处理和移位处理两个功能;
- 在循环处理中使用地址的自动增减,提高运行效率。

RISC 架构与 CISC 架构相比具有许多优点,但绝不是泾渭分明的,现代计算机通常外围工作采用 CISC 技术,内部运算采用 RISC 技术。未来的处理器发展方向是 RISC 和 CISC 技术

优势的相互融合，在 CISC 计算机中采用 RISC 的设计思想。

1.3 ARM 微处理器的指令系统

ARM 微处理器在一种体系结构中支持 ARM 和 Thumb 两种指令集，其中，ARM 指令长度为 32 位，Thumb 指令长度为 16 位。

为兼容数据总线宽度为 16 位的应用系统，ARM 体系结构除了支持执行效率很高的 32 位 ARM 指令集以外，同时支持 16 位的 Thumb 指令集。Thumb 指令集是 ARM 指令集的一个子集，是一个 ARM 存储压缩的指令系统。Thumb 指令集的指令编码长度为 16 位，与等价的 32 位代码相比较，Thumb 指令集不仅保留 32 代码优势，而且大大地节省了系统存储空间。在一般的情况下，Thumb 指令与 ARM 指令的时间效率和空间效率关系为：

- Thumb 指令代码所占用的存储空间约为 ARM 代码的 60%~70%；
- Thumb 指令代码使用的指令数比 ARM 系统中的代码多 30%~40%；
- 使用 32 位的存储器，ARM 指令系统比 Thumb 指令系统执行约快 40%；
- 使用 16 位的存储器，Thumb 代码比 ARM 代码执行约快 40%~50%；
- 与 ARM 代码相比较，使用 Thumb 代码，存储器功耗约降低 30%。

ARM 指令集和 Thumb 指令集各有其优点，若对系统速度有较高要求，应使用 32 位的存储系统和 ARM 指令集；若对系统的成本及功耗有较高要求，则应使用 16 位的存储系统和 Thumb 指令集。当然，若两者结合使用，充分发挥其各自的优点，会取得更好的效果。

Thumb 指令集是 ARM 指令集的功能子集，所有的 Thumb 指令都有对应的 ARM 指令，而且 Thumb 的编程模型也对应于 ARM 的编程模型。在应用程序的编写过程中，只要遵循调用的规则，Thumb 子程序和 ARM 子程序就可以互相调用。当处理器在执行 ARM 程序段时，称 ARM 处理器处于 ARM 工作状态；当处理器在执行 Thumb 程序段时，称 ARM 处理器处于 Thumb 工作状态。

Thumb 指令集中的数据处理指令的操作数仍然是 32 位，指令地址也为 32 位，但 Thumb 指令集为实现 16 位的指令长度，舍弃了 ARM 指令集的一些特性，如 ARM 指令都是有条件执行的，但大多数的 Thumb 指令是无条件执行的；大多数的 Thumb 数据处理指令的目的寄存器与其中一个源寄存器相同。

1.4 ARM 处理器系列

各厂商基于 ARM 体系结构的处理器，除了具有 ARM 体系结构的共同特点以外，每一个系列的 ARM 微处理器都有各自的特点和应用领域。ARM 微处理器目前包括下面几个系列：

- ARM7 系列；
- ARM9 系列；
- ARM9E 系列；
- ARM10E 系列；
- SecurCore 系列；
- Intel 的 Xscale。

其中, ARM7、ARM9、ARM9E 和 ARM10E 为 4 个通用处理器系列, 每一个系列提供一套相对独特的性能来满足不同应用领域的需求。SecurCore 系列专门为安全要求较高的应用而设计。ARM 微处理器的内核结构多达十几种, 有几十个芯片厂家生产, 内部功能配置组合千变万化。下面, 简要介绍一下各种处理器的特点。

1.4.1 ARM7 微处理器系列

ARM7 系列微处理器包括四种类型的内核: ARM7TDMI、ARM7TDMI-S、ARM720T、ARM7EJ-S。ARM7TDMI 是目前使用最广泛的 32 位嵌入式 RISC 处理器, ARM7 微处理器系列具有如下特点:

- 具有嵌入式 ICE 逻辑, 调试开发方便;
- 极低的功耗, 适合对功耗要求较高的应用, 如便携式产品;
- 能够提供 0.9MIPS/MHz 的三级流水线结构;
- 代码密度高, 并兼容 16 位的 Thumb 指令集;
- 对操作系统的广泛支持, 包括 Windows CE、Linux、Palm OS 等;
- 指令系统与 ARM9 系列、ARM9E 系列和 ARM10E 系列兼容, 便于用户的产品升级换代;
- 主频最高可达 130MIPS, 高速的运算处理能力能胜任绝大多数的复杂应用。

TDMI 的基本含义为:

T 代表支持 16 位压缩指令集 Thumb;

D 代表支持片上 Debug;

M 代表内嵌硬件乘法器 (Multiplier);

I 代表嵌入式 ICE, 支持片上断点和调试点。

1.4.2 ARM9 微处理器系列

ARM9 系列微处理器包含 ARM920T、ARM922T 和 ARM940T 3 种类型。ARM9 系列微处理器在高性能和低功耗特性方面提供最佳的性能, 具有以下特点:

- 五级流水线, 指令执行效率更高;
- 提供 1.1MIPS/MHz 的哈佛结构;
- 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- 支持 32 位的高速 AMBA 总线接口;
- 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- MPU 支持实时操作系统;
- 支持指令 Cache 和数据 Cache, 具有更高的指令和数据处理能力。

1.4.3 ARM9E 微处理器系列

ARM9E 系列微处理器包含 ARM926EJ-S、ARM946E-S 和 ARM966E-S 3 种类型。ARM9E 系列微处理器是综合处理器, 它使用单一的处理器内核, 提供了微控制器、DSP、Java 应用系统的解决方案, 极大地减少了芯片的占用面积和系统的复杂程度。ARM9E 系列微处理器还提供 DSP 处理能力, 很适合于那些需要同时使用 DSP 和微控制器的应用场合。ARM9E 系列微

处理器的主要特点如下:

- 支持 DSP 指令集, 适合于需要高速数字信号处理的场合;
- 5 级整数流水线, 指令执行效率更高;
- 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- 支持 32 位的高速 AMBA 总线接口;
- 支持 VFP9 浮点处理协处理器;
- 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- MPU 支持实时操作系统;
- 支持数据 Cache 和指令 Cache, 具有更高的指令和数据处理能力;
- 主频最高可达 300MIPS。

1.4.4 ARM10E 微处理器系列

ARM10E 系列微处理器包含 ARM1020E、ARM1022E 和 ARM1026EJ-S 3 种类型。ARM10E 系列微处理器具有高性能、低功耗的特点。由于采用了新的体系结构, 与同等的 ARM9 器件相比较, 在同样的时钟频率下, 性能提高了近 50%。同时, ARM10E 系列微处理器采用了两种先进的节能方式, 功耗极低。

ARM10E 系列微处理器的主要特点如下:

- 支持 DSP 指令集, 适合于需要高速数字信号处理的场合;
- 6 级流水线, 指令执行效率更高;
- 支持 32 位 ARM 指令集和 16 位 Thumb 指令集;
- 支持 32 位的高速 AMBA 总线接口;
- 支持 VFP10 浮点处理协处理器;
- 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统;
- 支持数据 Cache 和指令 Cache, 具有更高的指令和数据处理能力;
- 主频最高可达 400MIPS (Million Instructions Per Second);
- 内嵌并行读/写操作部件。

1.4.5 SecurCore 微处理器系列

SecurCore 系列微处理器包含 SecurCore SC100、SecurCore SC110、SecurCore SC200 和 SecurCore SC210 4 种类型。SecurCore 系列微处理器专为安全需要而设计, 提供了完善的 32 位 RISC 技术的安全解决方案。SecurCore 系列微处理器除具有 ARM 体系结构的低功耗、高性能的特点外, 还支持安全的系统解决方案。

SecurCore 系列微处理器除了具有 ARM 体系结构各种主要特点外, 还在系统安全方面具有如下的特点:

- 带有灵活的保护单元, 以确保操作系统和应用数据的安全;
- 采用软内核技术, 防止外部对其进行扫描探测;
- 可集成用户自己的安全特性和其他协处理器。

SecurCore 系列微处理器主要应用于一些对安全性要求较高的应用产品及应用系统, 如电

子商务、电子政务、电子银行业务、网络和认证系统等领域。

1.4.6 Xscale 处理器

Xscale 处理器是基于 ARMv5TE 体系结构的解决方案,是一款性能全、性价比高、功耗低的处理器。它支持 16 位的 Thumb 指令和 DSP 指令集,已使用在数字移动电话、个人数字助理和网络产品等场合。Xscale 处理器是 INTEL 目前主要推广的一款 ARM 微处理器。

1.5 ARM 微处理器的应用系统开发

ARM 微处理器及技术的应用几乎已经深入到各个领域:

(1) 工业控制领域。作为 32 位的 RISC 架构,基于 ARM 核的微控制器芯片不但占据了高端微控制器市场的大部分市场份额,同时也逐渐向低端微控制器应用领域扩展。ARM 微控制器的低功耗、高性价比,向传统的 8 位/16 位微控制器提出了挑战。

(2) 无线通信领域。目前已有超过 85% 的无线通信设备采用了 ARM 技术,ARM 以其高性能和低成本,在该领域的地位日益巩固。

(3) 网络应用。随着宽带技术的推广,ADSL 芯片采用 ARM 技术获得技术优势。ARM 在语音及视频处理上获得广泛支持,对传统的 DSP 应用提出挑战。

(4) 消费类电子产品。目前流行的数字音频播放器、数字机顶盒和游戏机中广泛采用 ARM 技术。

(5) 成像设备和安全产品。现在流行的数码相机和打印机中绝大部分采用 ARM 技术。手机中的 32 位 SIM 智能卡也采用了 ARM 技术。

习题 1

- 1.1 简述 ARM 处理器的发展过程。
- 1.2 简述 RISC 处理器的结构特点。
- 1.3 概述 RAM 指令集与 Thumb 指令集的优缺点。

第2章 ARM 体系结构

本章将介绍 ARM7TDMI 程序员模型、工作模式与工作状态、ARM 和 Thumb 状态的寄存器组织、存储器组织结构、异常及协处理器接口等一些基本概念。本章还要讲述 ARM 的编程基础，如 ARM 微处理器的基本工作原理、与程序设计相关的基本技术细节等。

2.1 ARM7TDMI 程序员模型

ARM7TDMI 是目前应用最广泛的 ARM 内核。它由 74209 个晶体管组成，通常采用 $0.35\mu\text{m}$ 制作工艺，最高时钟为 66MHz，最快速度为 60MIPS，耗电 87mW。ARM7TDMI 处理器是采用 32 位地址寻址，支持片内调试 (JTAG)，支持断点调试 (EMBEDDED ICE)，具有乘法运算器和三级指令流水线的 32 位整数 RISC 处理器。

掌握 ARM7TDMI 的工作性能和使用方法，使用者首先要从程序员编程的角度了解 ARM7TDMI 内核。

2.1.1 内核概述

ARM7TDMI 内核的基本结构如图 2.1 所示。ARM7TDMI 内核由扫描、指令译码和控制逻辑以及 ALU 组成。

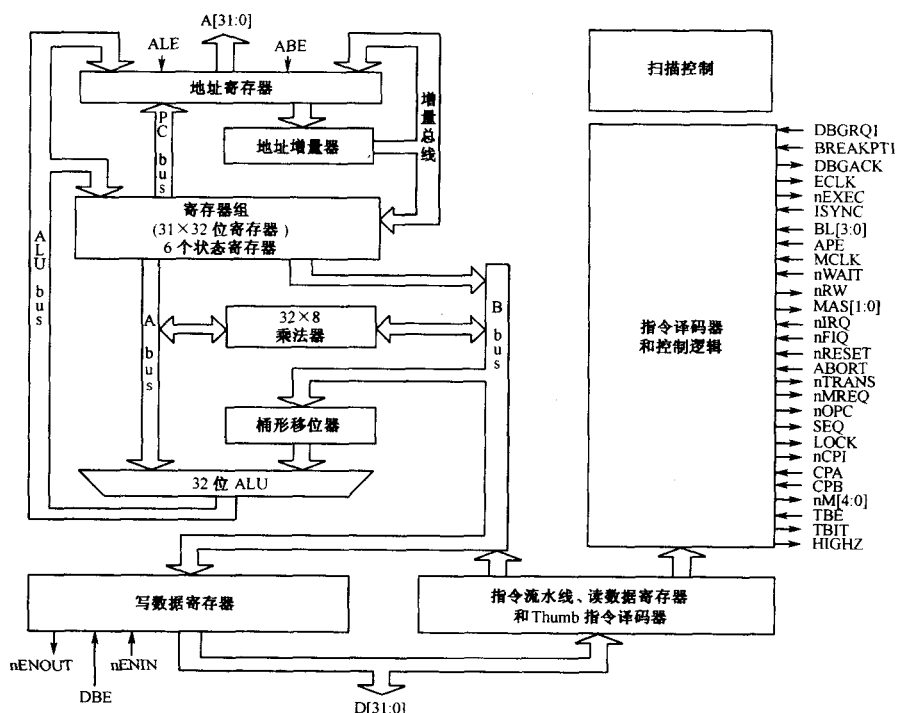


图 2.1 ARM7TDMI 内核的基本结构