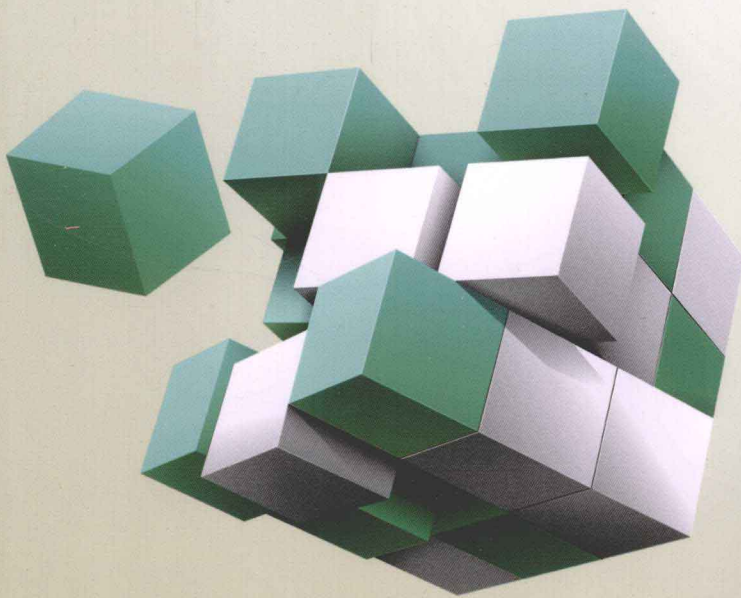


*ARM Architecture and Programming*

# ARM体系结构与编程

- 唐振明 主 编
- 田荣华 杨 强 编 著



中软国际卓越人才培养系列丛书

# ARM 体系结构与编程

唐振明 主编

田荣华 杨 强 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

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。本书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。

本书适合作为计算机、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。

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 体系结构与编程 / 唐振明主编; 田荣华, 杨强编著. —北京: 电子工业出版社, 2012.6

(中软国际卓越人才培养系列丛书)

ISBN 978-7-121-14773-9

I. ①A… II. ①唐… ②田… ③杨… III. ①微处理器, ARM—计算机体系结构—高等职业教育—教材 ②微处理器, ARM—程序设计—高等职业教育—教材 IV. ①TP332

中国版本图书馆 CIP 数据核字(2011)第 205601 号

策划编辑: 程超群

责任编辑: 郝黎明 文字编辑: 裴 杰

印 刷: 北京季蜂印刷有限公司

装 订: 三河市皇庄路通装订厂

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

开 本: 787×1092 1/16 印张: 14 字数: 358 千字

印 次: 2012 年 6 月第 1 次印刷

印 数: 4 000 册 定价: 28.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 序

当我翻阅了“中软国际卓越人才培养系列丛书”后，不禁为这套丛书的立意与创新之处感到欣喜。教育部“卓越工程师教育培养计划”有三个主要特征：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。这套丛书紧紧围绕“卓越计划”的要求展开，以企业人才需求为前提，同时又充分考虑了高校教育的特点，能让企业有效参与高校培养过程，是一套为“卓越计划”量身打造的丛书。

丛书的设计理念紧扣中软国际 ETC 的“5R”理念，即真实的企业环境、真实的项目经理、真实的项目案例、真实的工作压力、真实的就业机会，切实地将企业真实需求展现给读者。丛书中的知识点力求精简、准确、实用，显然是编著者经过反复推敲并精心设计的成果。丛书对企业用之甚少的知识点，都进行了弱化，用较少篇幅讲解，而对于企业关注的知识点，都使用非常详尽的内容进行学习。这样的设计对初学者尤其是在校学生非常必要，能够节省很多学习时间，在有限的时间内学习到企业关注的技能，而不是花费很多精力去钻研并不实用的内容。

丛书非常强调“快速入门”这一法宝，能够对某门技术“快速入门”永远是激发学习兴趣的关键。丛书设计了很多“快速入门”章节，使用详尽丰富的图示以及代码示例，保证读者只要根据丛书的指导进行操作，就能够尽快构建出相关技术的实例。

丛书非常注重实际操作，很多知识点都是从提出问题引出，从而在解决这个问题过程中讲解相关的技能。丛书中没有大篇幅的理论描述，尽力用最通俗、最简练的语言讲解每一个问题，而不是“故作高深”地使用很多新名词。

非常值得一提的是，丛书配备了对应的 PPT 讲义，并将 PPT 讲义显示到了相应章节，这种形式令人耳目一新。首先能起到提纲挈领的作用，帮助读者快速了解每个章节的主要内容，掌握完整的知识体系。另外，这种方式非常适合在高校教学中使用，能够完全与教材同步，方便学生课后复习及课前预习，可以有效提高教学效果。

这套丛书是中软国际多年行业经验的积累和沉淀，也是众多编者智慧与汗水的结晶，一定能在校企合作的道路上发挥积极长远的作用。

国家示范性软件学院建设工作办公室副主任  
北京交通大学软件学院院长



# 前 言

目前主要的嵌入式 CPU 架构有 ARM、Am186/88、SC-400、MIPS、PowerPC，其中基于 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的市场，据国外媒体报道，至 2011 年，英国芯片制造商 ARM Holdings PLC 生产的芯片已经成为全球手机和平板电脑芯片的制造执行标准，ARM 在全球手机芯片的市场份额已经超过 90%，所以学习嵌入式体系架构，ARM 必是首选。

目前类似 ARM 体系结构的书籍已出版了好多，更多的是专注于理念知识的讲解，但笔者认为一本好的技术方面的书籍，特别是作为专业类的技术性教材，要知识体系结构完整、基本概念及知识点清晰，通俗易懂，配套实验丰富。本书注意到这些特点，并提取了企业中 ARM 底层工程师的经验和精华并综合了重点大学一线教师的建议，合力打造了这本能在实际工作中使用的编程技术的书。

本书共分为 7 章，前后关联、层层递进、由简单到复杂、由单一到综合并配有大量的跟踪实验；从理论到实践，从感性到理性，从而使读者学习 ARM 体系结构及 ARM 汇编等不再枯燥无味。

第 1 章 ARM 处理器概述，介绍了 ARM 处理器的一些基本概念、ARM 公司及 ARM 系列的发展史，使读者从感性上认识什么是 ARM。

第 2 章 ARM 体系结构，详细介绍了什么是 ARM 体系结构，并从专业角度深度剖析 ARM 的内部结构和工作方式，包括 Load/Store 体系结构、RISC 的设计思想、ARM 流水线的思想及流水线的组织结构和 ARM 底层开发的调试方法。

第 3 章 ARM 编程模型，对 ARM 微处理器的体系结构、寄存器的组织、处理器的工作状态、运行模式以及处理器异常等内容进行了描述，这些内容也是 ARM 体系结构的基本内容，是系统软硬件设计的基础。希望读者能了解 ARM 微处理器的基本工作原理和一些与程序设计相关的基本技术细节，以便为以后的程序设计打下基础。

第 4 章 ARM 微处理器的指令系统，介绍了 ARM 指令集、Thumb 指令集，以及各类指令对应的寻址方式，通过对本章的阅读，希望读者能够了解 ARM 微处理器所支持的指令集及具体的使用方法。

第 5 章 ARM 汇编语言程序设计，系统地介绍了常见的伪操作，如符号定义伪操作、数据定义伪操作、汇编控制伪操作、信息报告伪操纵宏指令以及其他伪指令，并举例说明了其用法。然后对常见的几个 ARM 伪指令进行了讲解。在本章的最后部分介绍了 ARM 汇编语言与 C 语言混合编程。

第 6 章 Bootload 启动代码分析，主要的任务就是对启动代码的解读，以期读者在理解 ARM 硬件的基础上，对 ARM 的工作模式以及运行的状况有一个深入的理解。

第 7 章 ARM RealView MDK 集成开发环境，介绍了 ARM RealView MDK 集成开发环境的使用，本书大部分实验都是基于这个开发环境的，所以必须要熟练掌握 RealView MDK 的配置、使用及相关功能。并对 MDK 嵌入式开发中比较实用的三种解决方案：Keil MDK 编译器与 ULINK2 使用、Keil MDK 编译器与 J-LINK 使用、Keil MDK 编译器与 H-JTAG 使用做了讲解。

本书所配套的 PPT、课程大纲、代码等内容，可到华信教育资源网（[www.hxedu.com.cn](http://www.hxedu.com.cn)）免费下载。

在编写本书的过程中，得到了中软国际的很多领导、同事、朋友以及电子工业出版社编辑部的程超群的帮助，在此一并表示感谢。由于作者的知识所限，书中难免有不足之处，敬请广大读者海涵，并恳请赐正。

编 者

# 目 录

第 1 章 ARM 处理器概述 .....	(1)
1.1 嵌入式处理器简介 .....	(1)
1.1.1 嵌入式处理器分类 .....	(1)
1.1.2 嵌入式操作系统 .....	(3)
1.1.3 嵌入式处理器评价指标 .....	(3)
1.2 什么是 ARM .....	(4)
1.2.1 ARM 的概念 .....	(4)
1.2.2 ARM 公司发迹史 .....	(5)
1.3 ARM 体系结构的命名规则 .....	(6)
1.4 ARM 系列处理器简介 .....	(7)
1.4.1 ARM7 系列 .....	(8)
1.4.2 ARM9 系列 .....	(9)
1.4.3 ARM9E 系列 .....	(9)
1.4.4 ARM10 系列 .....	(9)
1.4.5 ARM11 系列 .....	(10)
1.4.6 SecurCore 系列 .....	(10)
1.4.7 ARM Cortex-A8 处理器 .....	(10)
1.5 ARM 处理器的技术优势及其应用 .....	(11)
1.6 本章小结 .....	(12)
1.7 课后练习 .....	(12)
第 2 章 ARM 体系结构 .....	(13)
2.1 ARM 体系结构的特点 .....	(13)
2.1.1 RISC 设计思想的体现 .....	(13)
2.2 ARM 流水线 .....	(15)
2.2.1 ARM 流水线的概念与原理 .....	(15)
2.2.2 流水线的分类 .....	(16)
2.2.3 3 级流水线 ARM 组织 .....	(17)
2.2.4 5 级流水线 ARM 组织 .....	(20)
2.2.5 6 级流水线 ARM 组织 .....	(22)
2.2.6 影响流水线性能的因素 .....	(22)
2.3 ARM 存储器 .....	(23)
2.4 I/O 管理 .....	(24)
2.5 ARM 开发调试方法 .....	(25)
2.5.1 指令集模拟器 .....	(25)
2.5.2 驻留监控软件 .....	(26)

2.5.3	JTAG 仿真调试 .....	(26)
2.5.4	基于 ULINK 在线仿真器 .....	(27)
2.6	本章小结 .....	(27)
2.7	课后练习 .....	(27)
第 3 章	ARM 编程模型 .....	(28)
3.1	数据类型 .....	(28)
3.1.1	ARM 的基本数据类型 .....	(28)
3.1.2	浮点数据类型 .....	(29)
3.1.3	存储器大/小端 .....	(29)
3.2	处理器工作模式 .....	(31)
3.3	ARM 寄存器组织 .....	(32)
3.3.1	通用寄存器 .....	(33)
3.3.2	程序状态寄存器 .....	(36)
3.4	异常中断处理 .....	(39)
3.4.1	异常种类 .....	(39)
3.4.2	异常优先级 .....	(44)
3.4.3	处理器模式和异常 .....	(44)
3.4.4	异常响应流程 .....	(45)
3.4.5	从异常处理程序中返回 .....	(47)
3.5	本章小结 .....	(48)
3.6	课后练习 .....	(49)
第 4 章	ARM 微处理器的指令系统 .....	(50)
4.1	ARM 微处理器的指令集概述 .....	(50)
4.1.1	ARM 微处理器的指令的分类与格式 .....	(50)
4.1.2	指令的条件域 .....	(52)
4.2	ARM 指令集 .....	(53)
4.2.1	数据处理指令 .....	(53)
4.2.2	移位指令 .....	(60)
4.2.3	乘法指令与乘加指令 .....	(63)
4.2.4	批量数据加载/存储指令 .....	(65)
4.2.5	跳转指令 .....	(66)
4.2.6	程序状态寄存器访问指令 .....	(68)
4.2.7	加载/存储指令 .....	(70)
4.2.8	协处理器指令 .....	(74)
4.2.9	异常产生指令 .....	(76)
4.3	ARM 指令的寻址方式 .....	(77)
4.3.1	立即寻址 .....	(77)
4.3.2	寄存器寻址 .....	(78)
4.3.3	寄存器间接寻址 .....	(78)



4.3.4	基址变址寻址	(78)
4.3.5	多寄存器寻址	(79)
4.3.6	相对寻址	(79)
4.3.7	堆栈寻址	(79)
4.4	Thumb 指令及应用	(81)
4.5	本章小结	(82)
4.6	课后练习	(82)
第 5 章	ARM 汇编语言程序设计	(83)
5.1	ARM 汇编器所支持的伪操作	(83)
5.1.1	符号定义 (Symbol Definition) 伪操作	(83)
5.1.2	数据定义 (Data Definition) 伪操作	(87)
5.1.3	汇编控制 (Assembly Control) 伪操作	(91)
5.1.4	信息报告 (Reporting) 伪操作	(95)
5.1.5	指令集选择 (Instruction Set Selection) 伪操作	(96)
5.1.6	其他伪操作	(98)
5.2	ARM 汇编器所支持的伪指令	(107)
5.2.1	ADR 伪指令	(107)
5.2.2	ADRL 伪指令	(108)
5.2.3	MOV32 伪指令	(108)
5.2.4	LDR 伪指令	(109)
5.3	汇编语言文件格式	(110)
5.3.1	ARM 汇编语言语句格式	(110)
5.3.2	ARM 汇编语言中的符号	(111)
5.3.3	汇编语言程序中的表达式和运算符	(112)
5.3.4	汇编语言预定义寄存器和协处理器	(114)
5.3.5	汇编语言的程序结构	(115)
5.3.6	汇编语言子程序调用	(116)
5.4	ARM 汇编语言与 C 语言混合编程	(117)
5.4.1	在 C 语言中内嵌汇编语言	(118)
5.4.2	在 C 语言中调用汇编语言的函数	(119)
5.4.3	在汇编语言中调用 C 语言的函数	(121)
5.5	本章小结	(122)
5.6	课后练习	(123)
第 6 章	Bootload 启动代码分析	(124)
6.1	汇编基础	(124)
6.2	启动代码功能模块分解	(125)
6.2.1	程序的入口地址	(125)
6.2.2	看门狗及中断的禁止	(126)
6.2.3	系统时钟初始化	(127)

6.2.4	初始化内存控制器	(128)
6.2.5	系统堆栈的初始化	(128)
6.2.6	建立中断向量表	(130)
6.2.7	跳转到 C 语言入口	(132)
6.3	Bootload 实验部分	(132)
6.3.1	实验环境	(132)
6.3.2	实验步骤	(132)
6.3.3	实验总结	(133)
6.4	Bootload 扩展部分—U-Boot 分析	(133)
6.4.1	Bootloader 的引导方式	(134)
6.4.2	Bootloader 的种类	(135)
6.4.3	U-Boot 源码结构	(137)
6.4.4	U-Boot 的编译	(138)
6.4.5	U-Boot 的移植	(141)
6.4.6	添加 U-Boot 命令	(143)
6.4.7	U-Boot 的调试	(145)
6.4.8	U-Boot 与内核的关系	(149)
6.4.9	U-Boot 的常用命令	(152)
6.4.10	U-Boot 的环境变量	(155)
6.4.11	使用 U-Boot	(157)
6.5	本章小结	(158)
6.6	课后练习	(158)
第 7 章	ARM RealView MDK 集成开发环境	(159)
7.1	RealView MDK 突出特性	(159)
7.2	MDK 功能介绍	(160)
7.2.1	$\mu$ Vision4 IDE	(160)
7.2.2	$\mu$ Vision4 IDE 主要特性	(161)
7.2.3	启动代码配置向导	(161)
7.2.4	$\mu$ Vision4 设备模拟器	(161)
7.2.5	性能分析器	(162)
7.2.6	RealView 编译器	(162)
7.2.7	MicroLib	(163)
7.2.8	RealView Real-Time Library (RealView RTL)实时库	(163)
7.2.9	ARM 软件开发工具解决方案	(164)
7.3	RealView MDK 的使用	(164)
7.3.1	$\mu$ Vision4 的安装	(164)
7.3.2	创建 $\mu$ Vision 工程	(169)
7.4	Keil MDK 编译器与 ULINK2 使用	(182)
7.4.1	ULINK2 概述	(182)

7.4.2	ULINK2 与 MDK 的链接使用 .....	(183)
7.5	Keil MDK 编译器与 J-LINK 使用 .....	(184)
7.5.1	J-LINK 概述 .....	(184)
7.5.2	J-LINK 与 MDK 的链接使用 .....	(185)
7.6	Keil MDK 编译器与 H-JTAG 使用 .....	(188)
7.6.1	H-JTAG 概述 .....	(188)
7.6.2	H-JTAG 调试结构 .....	(189)
7.6.3	H-JTAG 的安装 .....	(189)
7.6.4	H-JTAG 配置 .....	(192)
7.6.5	MDK 的安装与设置 .....	(194)
7.6.6	调试 .....	(195)
7.7	Keil 开发工具链 .....	(196)
7.7.1	用 UltraEdit 查看和编辑程序源文件 .....	(196)
7.7.2	Keil MDK 生成 BIN 过程 .....	(197)
7.7.3	ARM 工具链准备实验 .....	(200)
7.7.4	armasm 汇编器的使用 .....	(201)
7.7.5	armlink 链接器的使用 .....	(202)
7.7.6	armcc 编译器的使用 .....	(203)
7.7.7	FromELF 实用工具实验 .....	(205)
7.8	本章小结 .....	(206)
7.9	课后练习 .....	(206)
	参考文献 .....	(207)

# 第 1 章 ARM 处理器概述

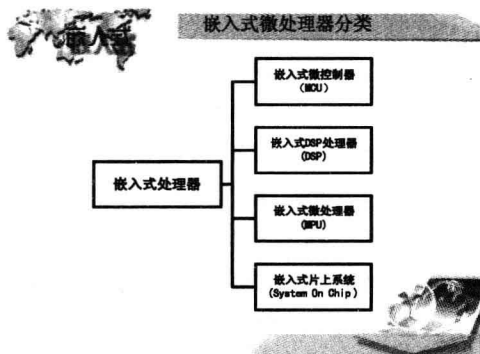
本章介绍 ARM 处理器的一些基本概念、ARM 公司及 ARM 系列的发展史，引导读者进入 ARM 技术的殿堂。

## 1.1 嵌入式处理器简介

嵌入式处理器是嵌入式系统的核心，是控制、辅助系统运行的硬件单元。其范围极其广阔，从最初的 4 位处理器，目前仍在大规模应用的 8 位单片机，到最新的受到广泛青睐的 32 位、64 位嵌入式处理器。

目前世界上具有嵌入式功能特点的处理器已经超过 1000 种，流行的体系结构有 MCU、MPU 等 30 多个系列。鉴于嵌入式系统广阔的发展前景，很多半导体制造商都大规模生产嵌入式处理器，并且自主设计处理器也已经成为了未来嵌入式领域的一大趋势，其中从单片机、DSP 到 FPGA 都有着各式各样的品种，速度越来越快，性能越来越强，价格也越来越低。目前嵌入式处理器的寻址空间可以从 64KB 到 16MB，处理速度最快可以达到 2000 MIPS（每秒百万条指令），封装从 8 个引脚到 144 个引脚不等。

### 1.1.1 嵌入式处理器分类



根据其现状，嵌入式处理器可分为以下几类。

#### (1) 嵌入式微处理器 (Micro Processor Unit, MPU)

嵌入式微处理器是由通用计算机中的 CPU 演变而来的。它的特征是具有 32 位以上的处理器，具有较高的性能，当然其价格也相应较高。但与计算机处理器不同的是，在实际嵌入式应用中，只保留和嵌入式应用紧密相关的功能硬件，去除了其他的冗余功能部分，这样就以最低的功耗和资源实现嵌入式应用的特殊要求。和工业控制计算机相比，嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点。目前主要的嵌入式处理器类型有 Am186/88、386EX、SC-400、Power PC、68000、MIPS 和 ARM/ StrongARM 系列等。其中，ARM/StrongARM 是专为手持设备开发的嵌入式微处理器，属于中档的价位。

嵌入式微处理器是嵌入式系统的核心，目前 32 位嵌入式微处理器是市场的主流。在 32 位嵌入式微处理器市场中，我们可以发现超过 100 家的芯片供应商和近 30 种指令体系结构。

在 1996 年以前, 最成功的嵌入式微处理器是 Motorola 公司的 68000 系列。此外嵌入式微处理器市场还包括其他体系结构, 如 Intel 公司的 I960、Motorola 公司的 Coldfire、Sun 公司的 Sparc, 以及嵌入式 X86 系列平台。当然, 最引人注目的还是 ARM 公司的 ARM 系列、MIPS 公司的 MIPS 系列, 以及 Hitachi 公司的 SuperH 系列(其中 ARM 和 MIPS 都是知识产权公司, 把他们的微处理器 IP 技术授权给半导体厂商, 由他们生产形态各异的微处理器芯片)。

嵌入式微处理器一般具备以下 4 个特点。

① 对实时多任务有很强的支持能力, 能完成多任务并且有较短的中断响应时间, 从而使内部的代码和实时内核的执行时间减少到最低限度。

② 具有功能很强的存储区保护功能, 这是由于嵌入式系统的软件结构已模块化, 而为了避免在软件模块之间出现错误的交叉作用, 需要设计强大的存储区保护功能, 同时也有利于软件诊断。

③ 可扩展的处理器结构, 以便能够最迅速地开发出满足应用的最高性能的嵌入式微处理器。

④ 嵌入式微处理器的功耗很低, 尤其是用于便携式的无线及移动的计算机和通信设备中靠电池供电的嵌入式系统更是如此, 其功耗为 mW 甚至  $\mu$ W 级。

### (2) 嵌入式微控制器 (Micro Controller Unit, MCU)

嵌入式微控制器的典型代表是单片机, 从 20 世纪 70 年代末单片机出现到今天, 虽然已经经过了 30 多年的历史, 但这种 8 位的电子器件目前在嵌入式设备中仍然有着极其广泛的应用。单片机芯片内部集成了 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM、EEPROM 等各种必要的功能和外设。和嵌入式微处理器相比, 嵌入式微控制器的最大特点是单片化, 体积大大减小, 从而使功耗和成本下降、可靠性提高。嵌入式微控制器是目前嵌入式系统工业的主流。嵌入式微控制器的片上外设资源一般比较丰富, 适合于控制, 因此称嵌入式微控制器。

由于嵌入式微控制器低廉的价格、优良的功能, 所以拥有的品种和数量最多, 比较有代表性的是 8051、MCS-251、MCS-96/196/296、P51XA、C166/167、68K 系列, 以及 MCU 8XC930/931、C540、C541, 并且有支持 I2C、CAN-Bus、LCD 及众多专用 MCU 和兼容系列。目前 MCU 占嵌入式系统约 70% 的市场份额。近来 Atmel 出产的 Avr 单片机由于其集成了 FPGA 等器件, 所以具有很高的性价比, 势必将推动单片机获得更高的发展。

### (3) 嵌入式 DSP 处理器 (Embedded Digital Signal Processor, EDSP)

嵌入式 DSP 处理器是专门用于信号处理方面的处理器, 其在系统结构和指令算法方面进行了特殊设计, 具有很高的编译效率和指令的执行速度。在数字滤波、FFT、频谱分析等各种仪器上 DSP 获得了大规模的应用。

DSP 的理论算法在 20 世纪 70 年代就已经出现, 但是由于当时专门的 DSP 处理器还未出现, 所以这种理论算法只能通过 MPU 等由分立元件实现。MPU 较低的处理速度无法满足 DSP 的算法要求, 其应用领域仅仅局限于一些尖端的高科技领域。随着大规模集成电路技术的发展, 1982 年世界上诞生了首枚 DSP 芯片, 其运算速度比 MPU 快了几十倍, 在语音合成和编解码器中得到了广泛应用。至 20 世纪 80 年代中期, 随着 CMOS 技术的进步与发展, 第二代基于 CMOS 工艺的 DSP 芯片应运而生, 其存储容量和运算速度都得到了成倍提高, 成为语音处理、图像硬件处理技术的基础。到 20 世纪 80 年代后期, DSP 的运算速度进一步提高, 应用领域也从上述范围扩大到了通信和计算机方面。20 世纪 90 年代后, DSP 发展到了第五代产品, 其集成度更高, 使用范围也更加广阔。

目前最为广泛应用的是 TI 的 TMS320C2000/C5000 系列及 DM 系列。另外, 如 Intel 的

MCS-296 和 Siemens 的 TriCore 也有各自的应用范围。

#### (4) 嵌入式片上系统 (System on Chip, SoC)

嵌入式片上系统追求产品系统最大包容的集成器件,是目前嵌入式应用领域的热门话题之一。嵌入式片上系统最大的特点是成功实现了软硬件的无缝结合,直接在处理器片内嵌入操作系统的代码模块。而且嵌入式片上系统具有极高的综合性,在一个硅片内部运用 VHDL 等硬件描述语言,实现了一个复杂的系统。用户不需要再像传统的系统设计一样,绘制庞大复杂的电路板,一点点地连接焊制,只需要使用精确的语言,综合时序设计并直接在器件库中调用各种通用处理器的标准,然后通过仿真之后就可以直接交付芯片厂商进行生产。由于绝大部分系统构件都是在系统内部,整个系统就特别简洁,不仅减小了系统的体积和功耗,而且提高了系统的可靠性和设计生产效率。

由于嵌入式片上系统往往是专用的,所以大部分都不为用户所知,比较典型的嵌入式片上系统是 Philips 的 Smart XA。少数通用系列如 Siemens 的 TriCore、Motorola 的 M-Core、某些 ARM 系列器件,以及 Echelon 和 Motorola 联合研制的 Neuron 芯片等。

预计不久的将来,一些大的芯片公司将通过推出成熟的、能占领多数市场的嵌入式片上系统芯片,一举击退竞争者。嵌入式片上系统芯片也将在声音、图像、影视、网络及系统逻辑等应用领域中发挥重要作用。

### 1.1.2 嵌入式操作系统

另外,从软件方面,主要可以依据操作系统的类型进行划分。目前嵌入式系统的软件主要有三种操作系统:多道批量处理操作系统,实时操作系统和分时操作系统。其中,实时操作系统又分为两类:硬实时操作系统和软实时操作系统。

多道批量处理操作系统一般用于计算中心比较大的计算机系统中。由于其硬件设施比较全、价格高,所以此操作系统十分注意 CPU 及其他设备的充分利用,追求高的吞吐量,不具备实时性。

实时系统是为执行特定功能而设计的,可以严格地按时序执行功能。其最大的特征就是程序的执行具有确定性。在实时操作系统中,如果系统在指定的时间内未能实现某个确定的任务,会导致系统的全面失败,则系统被称为硬实时操作系统。而在软实时操作系统中,虽然响应时间同样重要,但是超时却不会导致致命错误。一个硬实时操作系统往往在硬件上需要添加专门用于时间和优先级管理的控制芯片,而软实时操作系统则主要在软件方面通过编程实现时限的管理。例如,Windows CE 就是一个多任务分时操作系统,而 Ucos-II 则是典型的实时操作系统。

### 1.1.3 嵌入式处理器评价指标



#### 嵌入式处理器评价指标

- > 功耗: 为减小功耗,嵌入式微处理器设计了三种运行模式。
- > 代码存储密度: RISC指令集计算机由于要求指令编码长度固定,往往需要更多的指令来完成,从而增加了代码长度。
- > 集成度: 嵌入式微处理器一般都为专用市场设计的,需要较高的集成度。
- > 多媒体加速: 在传统的微处理器指令集的基础上增加 JPEG 和 MPEG 解压缩的高散余弦变换指令。



(1) 功耗。一般的嵌入式微处理器都有三种运行模式：运行模式 (operational)、待机模式 (standby or power down)、停机模式 (and clock-off)。功耗的评测指标是 MIPS/W。

(2) 代码存储密度。传统的 CISC 指令集计算机具有较好的代码存储密度。而 RISC 指令集计算机由于要求指令编码长度固定，虽然可以简化和加速指令译码过程，但为了实现与 CISC 指令集计算机相同的作业，往往需要更多的指令来完成，从而增加了代码长度。如 Hitachi 的 SuperH 体系结构采用了定长的 16 位指令，对每条指令按 16 位的格式存储。ARM 则采用 16 位扩展的 Thumb 指令集，片内的逻辑译码器将其等价为 32 位的 ARM 指令而实时解码。而 MIPS 则采用 MIPS16 方法来解决这个问题 (影响代码密度的另外一个主要因素是所采用的 C 编译器。ANSI C 是当前嵌入式领域的标准编程语言，随着嵌入式微处理器性能的提高，面向对象的语言也将被采用并会逐渐成为主流，一些编译器供应商已经开始着手解决代码密度问题)。

(3) 集成度。嵌入式微处理器一般都为专用市场设计的，需要较高的集成度。但把所有的外围设备都集成到一个芯片上也不是一种好的解决方案。这是因为高集成度使芯片变得复杂，芯片引脚变密，增加了系统设计和测试的复杂性。因此，集成外围设备时必须要考虑简化系统设计，并缩短整个系统的开发周期。

(4) 多媒体加速。为实现多媒体加速功能，嵌入式微处理器的设计者在传统的微处理器指令集的基础上增加了 JPEG 和 MPEG 解压缩的离散余弦变换指令。还有一些半导体厂商针对智能手机和移动通信市场的需求，将 RISC 微处理器和 DSP 集成在一个芯片上，如 TI 的 OMAP。



本节主要学习了嵌入式处理器的概念及嵌入式处理器的分类，针对嵌入式系统的核心嵌入式微处理器的特点及应用价值做了详细介绍。最后通过功耗、代码存储密度、集成度和多媒体加速等指标对嵌入式处理器的性能方面进行了参考评价。

## 1.2 什么是 ARM

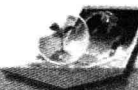
### 1.2.1 ARM 的概念



ARM 的概念

- ARM (Advanced RISC Machines)：是一种 32 位微处理器体系结构。
- ARM 不是单片机，准确来讲 ARM 是一种处理器的 IP 核：ARM 公司是专门从事基于 RISC 技术的芯片设计开发的公司。

www.51net.com



ARM (Advanced RISC Machines) 是一种 32 位微处理器体系结构，当前被广泛应用于消费电子、无线通信、工业控制和手持设备等领域。

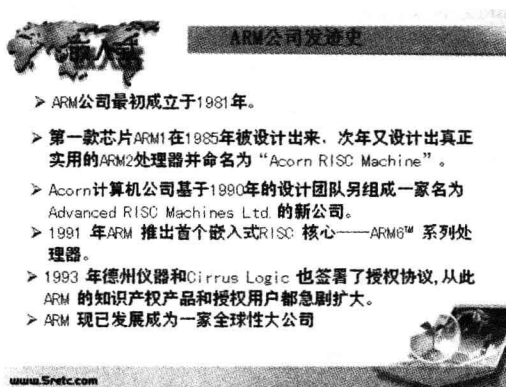
ARM 公司是专门从事基于 RISC 技术的芯片设计开发的公司。作为知识产权供应商,ARM 本身并不直接从事芯片生产,而是转让设计许可,由合作公司生产各具特色的芯片。世界各大半导体生产商从 ARM 公司购买其设计的 ARM 微处理器核,根据各自的不同应用领域,加入适当的外围电路,从而形成自己的 ARM 微处理器芯片进入市场。目前,全世界有几十家大的半导体公司都使用 ARM 公司的授权,因此既使得 ARM 技术获得更多的第三方工具、制造、软件的支持,又使整个系统成本降低,使产品更容易进入市场并被消费者接受,更具有竞争力。ARM 知识产权授权用户非常众多,其中包括世界顶级的半导体和系统公司。全球 20 家最大的半导体厂商中有 19 家是 ARM 的用户。这些合作伙伴通过使用 ARM 低价高效的 IP 核技术,研制生产微处理器、外围设备和系统芯片。迄今这些厂商共发售了超过 10 亿个 ARM 微处理器内核。

ARM 不是单片机,准确来讲 ARM 是一种处理器的 IP 核。ARM 公司开发出处理器结构后,开始向其他芯片厂商授权制造,芯片厂商可以根据自己的需要进行结构与功能的调整,因此实际中使用的 ARM 处理器有很多种类,主要有三星、飞利浦、ATMEL、Intel 制造的几大类,功能与使用上均不相同。ARM 处理器核还可以嵌入到其他专用芯片中作为中央处理单元使用,例如,飞利浦的 MP3 解码芯片就是采用 ARM7 核心的。ARM 系列处理器很少集成片上硬件资源,更接近今天的处理器范畴,基本不认为是单片机。在软件设计方便,ARM 和单片机也有很多不同,其中最大的不同是 ARM 引入了操作系统的概念,有了操作系统的支持,其主要具备以下几个优势:

- (1) 基于操作系统的支持可以轻松地写出多任务的复杂应用程序;
- (2) 完善的文件系统的支持;
- (3) 完善的网络协议支持;
- (4) 很多开源软件的支持。

当然,除了带来的优势以外,同样也带来了一些开发上的难度,例如,移植操作系统需要专业人员进行,开发人员需要熟悉操作系统,庞大的操作系统带来很大的系统开销等问题。

### 1.2.2 ARM 公司发展史



ARM 的设计是 Acorn 计算机公司 (Acorn Computers Limited) 于 1983 年开始的发展计划。后来,ARM 成为微处理器行业的一家知名企业。

ARM 公司最初成立于 1981 年,Acorn 计算机公司于 1983 年开始的发展计划的团队由 Roger Wilson 和 Steve Furber 带领,最初与英国广播公司合作为英国教育界设计小型机,着手开发一



种新架构，类似进阶的 MOS Technology 6502 处理器。当时采用的是美国的 6502 芯片。取得成功，他们开始设计自己的芯片，受当时美国加州大学伯克利分校提出的 RISC 思想的影响，他们设计的芯片也采用 RISC 体系结构，并命名为“Acorn RISC Machine”。此后，第一款芯片 ARM1 在 1985 年被设计出来，次年又设计出真正实用的 ARM2 处理器，ARM2 是具有 32 位数据总线和 24 位地址总线，并带有 16 个寄存器，可以说是当时最简化的 32 位微处理器，上面共有 30000 个晶体管（相较于 Motorola 六年后的 68000 其包含了 70000 个）。如此精简的结构使 ARM2 具有优异的低功耗特性，还能发挥比 Intel 80286 更好的效能，而且能获得比普通单片机更复杂的效果。后继的处理器 ARM3 更备有 4KB 的快取，使它能发挥更佳的性能。

20 世纪 80 年代末，苹果计算机开始与 Acorn 计算机公司合作开发新版的 ARM 核心，由于这个项目非常重要，所以 Acorn 计算机公司基于 1990 年的设计团队另组成一家名为 Advanced RISC Machines Ltd. 的新公司。这是苹果计算机、Acorn 计算机公司和 VLSI Technology 的合资企业，此后 Advanced RISC Machine 公司成为 ARM 公司的注册商标，使得 ARM 有时候反而称为 Advanced RISC Machine 而不是 Acorn RISC Machine，并专门从事 ARM 系列微处理器的开发。此后 ARM 公司设计了大量高性能、廉价、耗能低的 RISC 处理器、相关技术及软件。其技术具有性能高、成本低和功耗低等特点。适用于多种领域，如信息家电、消费电子、广告多媒体机和智能控制应用等。

Acorn 曾推出世界首个商用单芯片 RISC 处理器，而苹果计算机当时希望将 RISC 技术应用于自身系统，ARM 的微处理器新标准因此应运而生。ARM 成功地研制了首个低成本 RISC 架构，迅速在市场上崭露头角。与此同时 RISC 结构的竞争对手都着眼于提高性能，发展高端工作站处理器的 RISC 结构。1991 年 ARM 推出首个嵌入式 RISC 核心——ARM6™系列处理器后不久，VLSI 率先获得授权，一年后夏普和 GEC Plessey 也成为授权用户。1993 年德州仪器和 Cirrus Logic 也签署了授权协议，从此 ARM 的知识产权产品和授权用户都急剧扩大。1993 年 Nippon Investment and Finance (NIF) 成为 ARM 股东后，ARM 开始向全球拓展，分别在亚洲、美国和欧洲设立了办事处。1998 年 4 月 ARM 在伦敦证券交易所和纳斯达克交易所上市。

ARM 现已发展成为一家全球性大公司，在 3 大洲 8 个国家设有分支机构，雇员超过 720 万人。公司在英国布莱克本、剑桥和舍菲尔德，法国的 Sophia Antipolis，以及美国加州 Walnut Creek 和德州奥斯丁设有研发中心，并在法国、德国、日本、韩国、中国台湾、以色列、英国和美国建立了销售、行政和技术支持办事处。ARM 于 2002 年 7 月在中国上海成立分公司。

小结



本节主要学习了 ARM 的概念及 ARM 作为一种处理器的 IP 核在应用方面的优势，并对 ARM 公司的成长历程做了详细阐述。

### 1.3 ARM 体系结构的命名规则

每个 ARM 处理器都有一个特定的指令集架构，而一个特定的指令集架构又可以通过多种处理器实现。特定的指令集架构随着嵌入式市场的发展而发展。由于所有产品均采用一个通用的软件体系，所以相同的软件可在所有产品中运行（理论上如此）。

ARM 产品通常以 ARM{x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{-S} 的形式出现。表 1.1 显示了 ARM 的命名规则中这些后缀的具体含义。