

嵌入式系统项目实践技术

ARM嵌入式系统 开发与实践

崔更申 孙安青 编著



中国电力出版社

www.cepp.com.cn

嵌入式系统项目实践技术

ARM嵌入式系统 开发与实践

崔更申 孙安青 编著



中国电力出版社

www.cepp.com.cn

内 容 提 要

本书从理论与实践相结合的角度,全面系统地介绍了 ARM 硬件体系结构、嵌入式设计方法、ARM 指令系统、ARM 程序设计、KEIL 开发环境、基于 LPC2378 的具体开发应用。全书共 15 章,主要内容包
括嵌入式系统基础、ARM 体系结构、ARM 指令寻址方式、ARM 指令系统、ARM 程序设计、KEIL 开发环境、
LPC2378 核心系统、外设扩展原理及应用实例、基于 LPC2124 的具有日历功能的密码锁实例等。

本书理论与实践相结合,具有很强的引导性和实用性,讲解循序渐进,便于读者理解和掌握所学
的知识。非常适合作为计算机结构或计算机逻辑设计的启蒙教材,也适合作为计算机科学或计算机工程专
业的教材,还可供从事相关领域开发的工程技术人员参考。

图书在版编目 (CIP) 数据

ARM 嵌入式系统开发与实践 / 崔更申, 孙安青编著. —北京: 中国电力出版社, 2008
(嵌入式系统项目实践技术)
ISBN 978-7-5083-7236-5

I. A… II. ①崔…②孙… III. 微处理器, ARM IV. TP332

中国版本图书馆 CIP 数据核字 (2008) 第 126354 号

责任编辑: 夏华香
责任校对: 崔燕菊
责任印制: 郭华清

书 名: ARM 嵌入式系统开发与实践
编 著: 崔更申 孙安青
出版发行: 中国电力出版社
地址: 北京市三里河路 6 号 邮政编码: 100044
电话: (010) 68362602 传真: (010) 68316497
印 刷: 航远印刷有限公司
开本尺寸: 185mm × 260mm 印 张: 20 字 数: 473 千字
书 号: ISBN 978-7-5083-7236-5
版 次: 2008 年 10 月北京第 1 版
印 次: 2008 年 10 月第 1 次印刷
印 数: 0001—3000 册
定 价: 32.00 元

敬 告 读 者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

1985年4月26日, ARM的第一批样片送到英国剑桥的Acorn公司。这些样片是在美国加利福尼亚州圣何塞的VLSI Technology公司制造的。几个小时后, 它们开始运行程序。在20世纪80年代后期, ARM悄悄地发展支持Acorn公司的台式计算机产品。该产品成为英国教育界计算机的基础。20世纪90年代, 在ARM公司的精心经营下, ARM步入世界舞台, 在高性能、低功耗和低价格的嵌入式应用领域确立了市场的领先地位。优越的市场地位增加了ARM的资源, 加速了基于ARM产品的开发。

最近10年ARM开发的突出成果包括:

- 开发了称为Thumb的新型压缩指令格式, 这种格式用在小型系统中可降低成本和功耗;
- ARM9、ARM10和StrongARM系列处理器的开发, 显著地提高了ARM的性能;
- 软件开发和调试环境更好;
- 基于ARM处理器核的嵌入式应用领域更为广阔。

现代SoC和处理器设计的大多数原理都在ARM系列处理器设计中得到了应用, 并且ARM也开创了一些新的概念。基础的3级流水线核固有的简单性能使其成为实际处理器设计的优秀教学范例。同时, 嵌入在复杂系统芯片中基于ARM核的调试系统代表着当今技术的前沿。

本书的结构安排

本书从理论与实践相结合的角度, 全面系统地介绍了ARM硬件体系结构、嵌入式设计方法、ARM指令系统、ARM程序设计、KEIL开发环境、基于LPC2378的具体开发应用。全书共分15章, 章节内容安排如下。

第1章: 嵌入式系统基础

第2章: ARM体系结构

第3章: ARM指令寻址方式

第4章: ARM指令系统

第5章: ARM程序设计

第6章: KEIL开发环境

第7章: LPC2378核心系统

第8~14章: 外设扩展原理及应用实例

第15章: 基于LPC2124的具有日历功能的密码锁实例

本书的特色

本书内容安排与嵌入式学习过程一致, 先基础, 后深入, 且将理论与实践相结合, 具有很强的引导性和实用性。

本书的读者对象

本书不适合作为计算机结构或计算机逻辑设计的启蒙教材，其针对的读者应在相应领域具有相当于计算机科学或计算机工程专业大学本科二年级的水平。学习本书不需要事先熟悉 ARM 处理器。

本书主要由崔更申、孙安青编写，参与编写的有卜波涛、朱聪、王建华、潘振西、莫敬文、覃学松等，他们在资料搜集、整理方面做了大量工作；另外，孙洪波老师提供了大量的帮助，深圳英蓓特技术有限公司提供了 KEIL 开发环境的支持，深圳北天星技术有限公司提供 LPC2378 实验板及技术方面的支持，在此一并向他们表示感谢！

由于时间仓促，加之作者的水平有限，书中难免存在一些不足之处，欢迎广大读者批评指正。

作者

2008 年 9 月

目 录

前 言

第 1 章 嵌入式系统基础	1
1.1 ARM 处理器概述	1
1.1.1 ARM 技术的发展	1
1.1.2 RISC 微处理器的体系结构及其特点	2
1.1.3 ARM 微处理器的指令系统	3
1.1.4 ARM 微处理器系列	4
1.1.5 ARM 微处理器的应用系统开发	6
1.2 嵌入式系统的基本概念	6
1.2.1 嵌入式系统概念	6
1.2.2 嵌入式系统的组成	6
1.2.3 嵌入式系统的开发流程	10
1.2.4 嵌入式软件的开发环境	12
第 2 章 ARM 体系结构	15
2.1 ARM7TDMI 程序员模型	15
2.1.1 内核概述	15
2.1.2 译码与控制逻辑	16
2.1.3 运算器	17
2.1.4 内核接口信号	17
2.1.5 ARM 芯片的功能结构	21
2.2 工作状态与工作模式	22
2.2.1 ARM 的工作状态	22
2.2.2 ARM 的工作模式	23
2.3 ARM 和 Thumb 状态的寄存器组织	23
2.3.1 通用寄存器	23
2.3.2 Thumb 状态下的寄存器组织	25
2.3.3 程序计数器	25
2.3.4 程序状态寄存器	26
2.3.5 连接寄存器	28
2.3.6 堆栈指针寄存器	28
2.4 存储器组织结构	28
2.4.1 数据类型和存储数据格式	28
2.4.2 数据总线类型	30
2.4.3 ARM7TDMI 总线访问周期	32
2.4.4 指令预取与代码自修改	34
2.4.5 AMBA 接口	34
2.5 异常	34

2.5.1	异常类型及其处理过程	35
2.5.2	异常优先级 (Exception Priority) 和异常向量 (Exception Vector)	36
2.5.3	应用程序中的异常响应	37
2.5.4	异常返回	38
2.6	协处理器接口	38
2.6.1	协处理器概述	39
2.6.2	ARM7TDMI 与协处理器的协同关系	39
2.6.3	协处理器使用中的若干问题	39
第 3 章	ARM 指令的特点和寻址方式	41
3.1	ARM 指令特点	41
3.2	ARM 指令的寻址方式	43
第 4 章	ARM 指令系统概述	49
4.1	ARM 指令集	49
4.2	分支指令	51
4.3	ARM 的数据处理指令	52
4.4	异常处理指令	60
4.5	ARM 的存储器加载/存储指令	61
4.5.1	单寄存器加载	62
4.5.2	单寄存器存储	64
4.5.3	多寄存器加载和多寄存器存储	65
4.5.4	寄存器交换	66
4.6	协处理器指令	67
4.7	ARM 状态寄存器访问指令	69
第 5 章	ARM 程序设计	71
5.1	ARM 汇编语言伪指令	71
5.1.1	符号定义伪指令	71
5.1.2	空间分配伪指令	73
5.1.3	汇编控制伪指令	75
5.1.4	格式控制伪指令	77
5.1.5	ARM 伪指令	78
5.1.6	Thumb 伪指令	80
5.2	ARM 汇编语言语句格式	80
5.2.1	ARM 汇编语言的符号	80
5.2.2	汇编语言程序中的表达式和运算符	82
5.3	ARM 语言程序结构	84
5.4	简单的 ARM 程序设计	85
5.4.1	简单的 ARM 程序	85
5.4.2	子程序调用程序设计	86
5.4.3	分支程序设计	86
5.4.4	查表与散转程序设计	87
5.4.5	数据串拷贝程序的设计	88
5.5	ARM-Thumb 交互工作	89
5.5.1	交互工作原理	89
5.5.2	交互程序	91

5.5.3	交互子程序调用	92
5.6	ARM 汇编语言与 C 语言混合编程	94
5.6.1	ARM 过程调用标准 ATPCS	94
5.6.2	内嵌汇编器	95
5.6.3	C 语言和 ARM 汇编语言间的相互调用	100
第 6 章	Keil 开发环境的使用	103
6.1	Keil 开发工具介绍	103
6.1.1	μ Vision3 集成开发环境	103
6.1.2	操作模式	103
6.1.3	菜单命令、工具栏、快捷键	104
6.2	Keil 开发功能的使用	107
6.2.1	创建应用程序	107
6.2.2	工程目标和文件组	114
6.3	使用 EnLink 仿真 LPC23xx 控制器	115
6.3.1	开发环境	115
6.3.2	创建一个工程	115
第 7 章	LPC23XX 系统核心应用	120
7.1	系统结构与引脚分布	120
7.1.1	PLC23xx 系统结构图	120
7.1.2	PLC2378 引脚分布	122
7.2	核心部分组成	130
7.2.1	电源电路	130
7.2.2	复位	131
7.2.3	系统时钟	131
7.2.4	锁相环 (PLL)	132
7.2.5	存储映射	139
7.2.6	向量中断管理	146
7.2.7	外部中断	155
7.2.8	IRQ 中断的设计实例	157
7.2.9	IRQ 中断过程分析	157
第 8 章	GPIO 原理与应用实例	158
8.1	概述	158
8.2	特性	158
8.3	应用	158
8.4	引脚描述	158
8.5	寄存器描述	159
8.6	GPIO 使用注意事项	160
8.7	GPIO 应用实例	161
8.7.1	循环灯实例	161
8.7.2	按键计数实例	162
8.7.3	简易数字计算器实例	164
第 9 章	定时器 0/定时器 1 原理与应用实例	175
9.1	概述	175

9.2	特性	175
9.3	引脚描述	176
9.4	结构	176
9.5	寄存器描述	177
9.6	功能框图	182
9.7	使用示例	183
9.8	定时器 0/定时器 1 应用实例	184
9.8.1	“嘀嘀”报警产生实例	184
9.8.2	MIDI 音乐发生器实例	186
9.8.3	数字钟实例	191
第 10 章	脉宽调制 (PWM) 原理及应用实例	197
10.1	概述	197
10.2	特性	197
10.2.1	单边沿控制的 PWM 输出规则	199
10.2.2	双边沿控制的 PWM 输出规则	200
10.3	管脚描述	200
10.4	寄存器描述	200
10.5	PWM 基本操作方法	205
10.6	PWM 应用实例	205
10.6.1	PWM 产生警笛音实例	205
10.6.2	PWM 式 D/A 转换实例	208
第 11 章	SPI 接口原理及应用实例	211
11.1	概述	211
11.2	特性	211
11.3	管脚描述	211
11.4	SPI 结构	212
11.5	SPI 描述	212
11.5.1	SPI 总线描述	212
11.5.2	SPI 数据传输	212
11.6	SPI 外设描述	213
11.7	寄存器描述	214
11.8	基本操作	216
11.8.1	主机操作	216
11.8.2	从机操作	217
11.9	应用实例	217
11.9.1	串/并转换实例	217
11.9.2	16 × 16 点阵 LED 屏显示实例	219
第 12 章	I²C 接口原理及应用实例	227
12.1	I ² C 简介	227
12.1.1	控制字节	228
12.1.2	写操作	228
12.1.3	读操作	228
12.1.4	结束语	229
12.2	概述	229

12.3	特性	229
12.4	引脚描述	229
12.5	结构	230
12.5.1	串行时钟发生器	231
12.5.2	时序和控制	231
12.5.3	控制寄存器	231
12.5.4	状态译码器和状态寄存器	231
12.6	I ² C 描述	231
12.6.1	总线配置	231
12.6.2	I ² C 操作模式	232
12.6.3	主 I ² C 模式	233
12.6.4	从 I ² C 模式	237
12.6.5	两种特殊状态	241
12.6.6	仲裁和同步逻辑	241
12.7	寄存器描述	242
12.8	基本操作	245
12.9	初始化举例	246
12.10	应用实例	246
12.10.1	AT24C02 的读写实例	246
第 13 章	A/D 转换器	255
13.1	特性	255
13.2	引脚描述	255
13.3	寄存器描述	256
13.4	基本操作	257
13.5	使用示例	257
13.6	应用实例	258
第 14 章	UART0 接口	264
14.1	特性	264
14.2	引脚描述	264
14.3	结构	264
14.4	寄存器描述	265
14.5	使用示例	270
14.6	应用实例	271
第 15 章	基于 LPC2124 的具有日历功能的密码锁实例	274
15.1	概述	274
15.1.1	系统功能	274
15.1.2	系统功能的实现	274
15.2	硬件电路的设计	275
15.2.1	电源电路	275
15.2.2	LPC2124 的最小系统	275
15.2.3	键盘接口电路	277
15.2.4	LCD 液晶显示电路	278
15.2.5	实时日历时钟接口电路	281
15.2.6	提示音功放电路	283

15.2.7	开锁控制电路	284
15.3	软件设计	284
15.3.1	初始化程序设计	285
15.3.2	LCD 液晶显示驱动程序设计	287
15.3.3	键盘驱动程序设计	289
15.3.4	实时日历时钟 (DS1302) 驱动程序设计	291
15.3.5	Timer0 的定时中断服务程序设计	293
15.3.6	主程序设计	296
15.3.7	实例总结	308
	参考文献	309

嵌入式系统基础

1.1 ARM 处理器概述

随着信息技术和网络技术的高速发展，嵌入式产品日益广泛地渗透到日常生活、科学研究和军事技术等领域。从家用洗衣机、电冰箱，到交通工具（如汽车），以及办公室里的远程会议系统，都使用了嵌入式产品。

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三种方式。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）、嵌入式数字信号处理器（Digital Signal Processor, DSP）。

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。

1.1.1 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、Acorn 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 起初进行的是 GPS、音乐播放装置和游戏机等的设计，移动通信行业的发展，给 ARM 带来了巨大的机会。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：如果项目从头立项，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%，这种做法风险低，开发速度也快。正巧，ARM 正在为其设计紧凑、功耗低的 32 位 CPU 微处理器技术寻找市场。TI 采用 ARM 的授权技术，实现了手机从模拟向数字的转变。

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，授权业务发展起来。

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的硬件或者软件，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，世界上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。利用合作关系，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。

1.1.2 RISC 微处理器的体系结构及其特点

传统的计算机结构是复杂指令集计算机 (Complex Instruction Set Computer, CISC) 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% 程序代码使用，这就是 2—8 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：

- (1) 采用固定长度的指令格式。
- (2) 使用单周期指令，便于流水线操作执行。
- (3) 使用大量寄存器，数据处理指令只对寄存器进行操作。
- (4) 采用加载/存储指令批量传输数据，以提高数据的传输效率。

- (5) 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。
- (6) 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。
- (7) 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，但绝不是互不相容，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。

1.1.3 ARM 微处理器的指令系统

ARM 微处理器在较新的体系结构中支持 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 位的代码优势，而且大大节省了系统存储空间。在一般情况下，Thumb 指令与 ARM 指令的时间效率和空间效率关系为：

- (1) Thumb 指令代码所占用的存储空间约为 ARM 代码的 60%~70%。
- (2) Thumb 指令代码使用的指令数比 ARM 系统中的代码多 30%~40%。
- (3) 使用 32 位的存储器，ARM 指令系统比 Thumb 指令系统执行约快 40%。
- (4) 使用 16 位的存储器，Thumb 代码比 ARM 代码执行约快 40%~50%。
- (5) 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。

ARM 微处理器目前包括下面几个系列，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。

- (1) ARM7 系列。
- (2) ARM9 系列。
- (3) ARM9E 系列。
- (4) ARM10E 系列。

(5) SecurCore 系列。

(6) Intel 的 Xscale。

其中, ARM7、ARM9、ARM9E 和 ARM10 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, 由几十个芯片厂家生产, 内部功能配置组合千变万化。下面简要了解各种处理器的特点。

1.1.4 ARM 微处理器系列

1. ARM7 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。其中 ARM7TMDI 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。

- (1) 具有嵌入式 ICE 逻辑, 调试开发方便。
- (2) 极低的功耗, 适合对功耗要求较高的应用, 如便携式产品。
- (3) 能够提供 0.9MIPS/MHz 的 3 级流水线结构。
- (4) 代码密度高并兼容 16 位的 Thumb 指令集。
- (5) 对操作系统的广泛支持, 包括 Windows CE、Linux、Palm OS 等。
- (6) 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。

(7) 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 的基本含义如下。

- (1) T: 支持 16 位压缩指令集 Thumb。
- (2) D: 支持片上 Debug。
- (3) M: 内嵌硬件乘法器 (Multiplier)。
- (4) I: 嵌入式 ICE, 支持片上断点和调试点。

2. ARM9 微处理器系列

ARM9 系列微处理器包含 ARM920T、ARM922T 和 ARM940T 三种类型。ARM9 系列微处理器在高性能和低功耗特性方面提供了最佳的性能, 该处理器具有以下特点。

- (1) 5 级流水线, 指令执行效率更高。
- (2) 提供 1.1MIPS/MHz 的哈佛结构。
- (3) 支持 32 位 ARM 指令集和 16 位 Thumb 指令集。
- (4) 支持 32 位的高速 AMBA 总线接口。
- (5) 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统。
- (6) MPU 支持实时操作系统。
- (7) 支持指令 Cache 和数据 Cache, 具有更高的指令和数据处理能力。

3. ARM9E 微处理器系列

ARM9E 系列微处理器包含 ARM926EJ-S、ARM946E-S 和 ARM966E-S 三种类型。ARM9E 系列微处理器是综合处理器, 它使用单一的处理器的内核, 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。

- (1) 支持 DSP 指令集, 适合于需要高速数字信号处理的场合。
- (2) 5 级整数流水线, 指令执行效率更高。
- (3) 支持 32 位 ARM 指令集和 16 位 Thumb 指令集。
- (4) 支持 32 位的高速 AMBA 总线接口。
- (5) 支持 VFP9 浮点处理协处理器。
- (6) 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统。
- (7) MPU 支持实时操作系统。
- (8) 支持数据 Cache 和指令 Cache, 具有更高的指令和数据处理能力。
- (9) 主频最高可达 300MIPS。

4. ARM10E 微处理器系列

ARM10E 系列微处理器包含 ARM1020E、ARM1022E 和 ARM1026EJ-S 三种类型。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 系列微处理器相比较, 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。

- (1) 支持 DSP 指令集, 适合于需要高速数字信号处理的场合。
- (2) 6 级流水线, 指令执行效率更高。
- (3) 支持 32 位 ARM 指令集和 16 位 Thumb 指令集。
- (4) 支持 32 位的高速 AMBA 总线接口。
- (5) 支持 VFP10 浮点处理协处理器。
- (6) 全性能的 MMU, 支持 Windows CE、Linux、Palm OS 等多种主流嵌入式操作系统。
- (7) 支持数据 Cache 和指令 Cache, 具有更高的指令和数据处理能力。
- (8) 主频最高可达 400MIPS。
- (9) 内嵌并行读/写操作部件。

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 四种类型。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特点:

- (1) 带有灵活的保护单元, 以确保操作系统和应用数据的安全。
- (2) 采用软内核技术, 防止外部对其进行扫描探测。
- (3) 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。

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。

1.1.5 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，简介如下。

(1) 工业控制领域。作为 32 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的优势，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 技术获得了技术优势。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了 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了 ARM 技术。手机中的 32 位 SIM 智能卡也采用了 ARM 技术。

1.2 嵌入式系统的基本概念

在当前数字信息技术和网络技术高速发展的后 PC 时代，嵌入式系统已经广泛渗透到科学研究、工程设计、军事技术等领域以及人们的日常生活。随着国内外嵌入式产品的进一步开发和推广，嵌入式技术和人们的生活结合得越来越紧密。普通的家用电器和通信工具、各种交通工具，以及办公室里的远程会议系统等，都成为可以使用嵌入式技术开发和改造的领域。

在这样的时代背景下，越来越多的技术人员投身于嵌入式系统的设计和开发中。作为一本重视实用性的嵌入式设计教材，本章将为刚开始学习嵌入式系统的人员进行入门级的介绍，内容包括嵌入式开发平台的选择以及嵌入式开发的基本流程等。

1.2.1 嵌入式系统概念

嵌入式系统以应用为中心，以计算机技术为基础，采用可裁减软硬件，适用于对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序四个部分组成，用于实现对其他设备的控制、监视或管理等功能。嵌入式系统最典型的特点是与人们的日常生活紧密相关，任何一个普通人都可以拥有各类运用了嵌入式技术的电子产品，小到 MP3、PDA 等微型数字化设备，大到信息家电、智能电器、车载 GIS。事实上，新型嵌入式设备在数量上已远超过了通用计算机。

1.2.2 嵌入式系统的组成

嵌入式系统是专用计算机应用系统，它具有一般计算机组成的共性，也是由硬件和软件组成。图 1-1 完整地描述了嵌入式系统的软硬件各部分的组成结构。