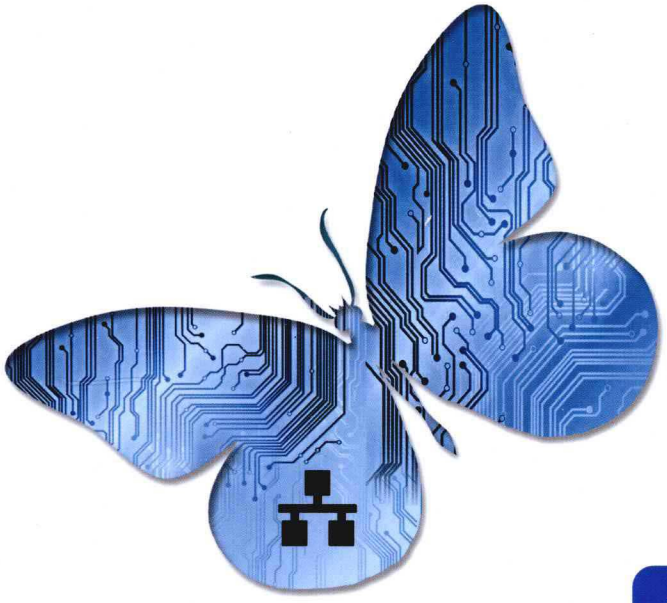
首本从系统视角讲解STM32开发规范和编码规范的书，学习嵌入式系统开发必读。
全方位讲解在STM32F107开发板上移植FreeRTOS和LwIP的全过程。
配套极具性价比的硬件开发平台，并提供完整工程文件和源代码，极具可操作性。

HZ BOOKS 华章科技

单片机与嵌入式



STM32

STM32嵌入式系统开发实战指南

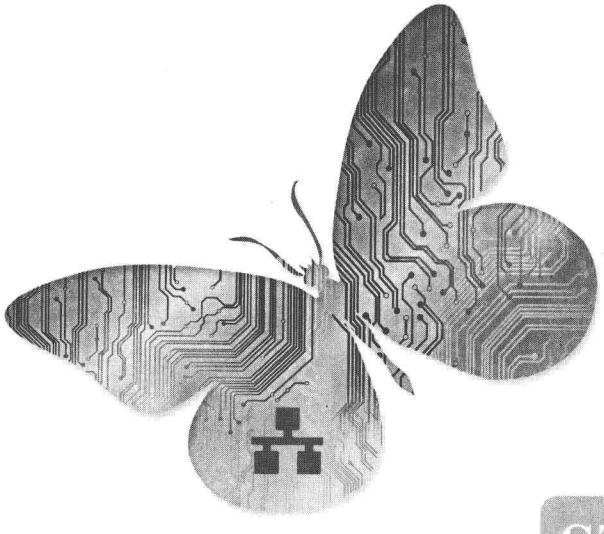
FreeRTC 联合移植

李志明 檀永 徐石明 等编著

附光盘

机械工业出版社
China Machine Press

单片机与嵌入式



STM32

STM32嵌入式系统开发实战指南

FreeRTOS操作系统的合移植

李志明 檀永 徐石明 丁孝华 桑林 编著

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

STM32嵌入式系统开发实战指南：FreeRTOS与LwIP联合移植 / 李志明等编著. —北京：机械工业出版社，2013.4

ISBN 978-7-111-41716-3

Ⅰ. S… Ⅱ. 李… Ⅲ. 实时操作系统 Ⅳ. TP316.2

中国版本图书馆CIP数据核字（2013）第042854号

本书以一款轻量级嵌入式实时操作系统为样本，阐述了嵌入式实时操作系统任务管理、时间管理、资源共享、内存管理等机制，介绍了内核及TCP/IP的移植和具体使用方法。为了避免枯燥的理论阐述，本书辅以适量的例程帮助大家学习。此外，本书还简要阐述了硬件平台设计、项目开展的一般步骤和注意事项。

本书适合已熟悉STM32的操作、掌握基于STM32官方驱动库的前后台模式应用软件开发的读者或初级嵌入式软件开发工程师阅读。

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：秦 健

北京市荣盛彩色印刷有限公司印刷

2013年5月第1版第1次印刷

186mm × 240mm · 21印张

标准书号：ISBN 978-7-111-41716-3

ISBN 978-7-89433-833-4（光盘）

定 价：69.00元（附光盘）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991 88361066 投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

序　言

嵌入式系统起源于微型机时代，伴随着网络、通信、多媒体技术的发展，嵌入式系统的应用已经深入社会生活的各个领域，业已成为继个人计算机和互联网之后，信息技术领域技术的新热点。

测量与控制的需求一直伴随着人类社会的进步，在人类进入现代化工业社会后，人们对这种需求有了更高的要求和期待。从以蒸汽机的发明为标志的工业革命开始，人们对控制有了全新的认识。最初，人们运用机械原理实现复杂程度不一的机械结构来满足对工程控制的需求，随着电子技术的发展及其展现的优点，人们又逐渐采用模拟、数字或混合式的电子控制方式来完成既定的测量控制需求。直到20世纪70年代，新型的微型计算机凭借其体积小、功耗低、结构简单、可靠性高、使用方便、性能价格比高等一系列优点，及其表现出的智能化水平迅速获得了控制专业人士的青睐。例如，将微型计算机经适当的机械加固和电气改造，并配置相应外围接口电路，即可实现各种复杂的工程应用，如工况监测、实时控制等。在微型计算机诞生后相当长的时期，这种将微型计算机嵌入一个对象体系中，实现对象体系的智能化控制得到了广泛的工程应用。

但微型计算机的设计是以商业应用为初衷的，微型计算机的体积、价位和可靠性都无法满足广大对象系统的嵌入式应用要求。半导体厂商逐渐意识到嵌入式系统的潜在市场需求，并陆续推出了各具特色的微型化控制芯片。1976年，Intel公司推出了MCS-48单片机，开创了将微处理机系统的各种CPU以外的资源集成到CPU硅片上的时代。1980年，Intel公司对MCS-48单片机进行了全面完善，推出了8位MCS-51单片机，并获得巨大成功，奠定了嵌入式系统的单片机应用模式。1984年，Intel公司又推出了16位8096系列并将其称为嵌入式微控制器，“嵌入式”一词第一次在微处理机领域出现，这标志嵌入式应用的兴起与快速发展的时代来临。

随着半导体制造工艺的进步，为了满足高速、实时信号处理的市场需求，数字信号处理芯片（DSP）及可编程逻辑器件（PLD、CPLD、FPGA）等高速嵌入式处理器件应运而生。DSP是将模拟信号转换成数字信号以后进行高速实时处理的专业处理器，在其诞生的最初只能完成既定的逻辑算术运算，但其处理速度已远远超越当时的微控制器。随着集成电路技术的发展，DSP处理能

力得到了不断提升，当前基于DSP的工程应用主要着眼于算法设计和实现，已广泛应用于数字通信、测量控制、图像处理等领域。可编程逻辑器件的发展始于20世纪70年代，在经历了40多年的发展后，已形成了以现场可编程门阵列器件为代表的特色各异的信号处理器件。FPGA通常比DSP拥有更快的运行速度，可以实现复杂的高速逻辑运算，其具有设计灵活、高集成度、高速、高可靠性、开发周期短、前期投资风险小等优点，在芯片内部可实现板级电路的功能，能有效提高设计的效率和可靠性。此外，随着FPGA芯片供应商和第三方公司对应用开发支持的进一步完善，芯片的设计周期和成本得以进一步压缩。

自嵌入式处理器诞生后的40多年历史中，面向各种不同应用领域、特色各异的嵌入式处理器件不断出现、升级和换代，广泛应用于人类的社会生活，也极大地改善了人类的生活。特别在20世纪90年代后，在网络、通信、多媒体技术应用需求的驱动下，嵌入式处理器经历了高度的发展和工程应用。随着32位微处理器和32位微控制器成本大幅下降，它们已经逐渐取代了传统8位、16位微控制器。与此同时，由于面向嵌入式系统的高端应用的工作速度快，外部设备资源丰富，加上应用本身的复杂性、可靠性要求等，软件开发的复杂性逐渐提升，如网络通信设备、电力二次设备、手持终端、多媒体设备、机顶盒等。这促使嵌入式软件开发工程师渴望摆脱繁重的底层软件开发，将更多的精力集中于应用层面的开发。嵌入式软件开发工程师希望能够提供一种类似于微型计算机操作系统的平台，只需根据实际采用的硬件平台做少量的移植工作即可完成嵌入式系统资源的管理和任务调度，在此基础上将主要精力集中于任务级代码的实现。

嵌入式系统最初的应用是基于单片机的，大多数以可编程控制器的形式出现，具有监测、控制、工况指示等功能，在工业和军事应用领域最为普遍。由于受限于系统资源，通常没有获得操作系统的支持，只能通过汇编语言对系统进行直接控制，所以只能使用8位的单片机芯片来运行一些单线程的程序。20世纪80年代，随着I/O接口、串行接口以及RAM、ROM等外设部件集成在芯片微处理器中，芯片制造商推出了面向I/O设计的微控制器，一经推出便迅速得到了市场和专业人士的认可。与此同时，业界出现了功能较为简单的“操作系统”雏形，使得开发周期和效率得到了一定的提升。直到20世纪90年代，在分布控制、柔性制造、网络通信和信息家电等巨大需求的驱动下，嵌入式系统进一步飞速发展，而面向实时信号处理的嵌入式处理器也向高速度、高精度、低功耗的方向发展。随着硬件实时性要求的提高，嵌入式系统的软件规模不断扩大，复杂程度也逐渐提高，实时多任务操作系统（Real-Time Operating System，RTOS）诞生，并开始成为嵌入式系统的主流，例如WindRiver公司的VxWorks、QNX系统软件公司的QNX、嵌入式Linux、微软公司的Windows CE、美国系统集成公司的pSOS、Micrium公司的uC/OS、FreeRTOS、RT-thread等。

实时操作系统是一种能够在既定时间内完成系统功能，对外部和内部事件在同步或者异步时间内做出及时响应，并控制所有实时任务协调一致运行的系统。在实时操作系统中，操作的正确

性不仅依赖于逻辑设计的正确程度，而且与系统任务级时序密切相关。因而，实时系统具有及时响应和高可靠性等主要特点。通常，实时操作系统有硬实时和软实时之分，硬实时要求在规定的时间内必须完成既定的操作，否则有可能造成灾难性后果；软实时则只要按照任务的优先级，尽可能快地完成操作即可，广泛应用于消费类电子产品，这类操作系统对事件响应没有严格要求，较为关注事件响应的正确性和用户体验的舒适度，如产品的按键操作应不致使用户产生“反应迟钝”的感受。当然，上述这种划分仅具有概念上的意义，实时性对不同应用有着不同的界定，而且大多数嵌入式系统允许软硬两种实时性同时存在，对系统产生关键影响的事件一般具有严格的时限要求，是硬实时的，另外一些事件的时限要求则是软实时的，诸如人机交互。也有业内人士根据系统对任务的响应时间将实时操作系统分为弱实时系统、一般实时系统和强实时系统。无论如何划分，其本质是对系统事件响应时间的一种要求和评判。嵌入式软件开发工程师在项目规划时需根据工程实际需求，在硬件配置、实时操作系统选择和项目成本之间进行一定的统筹和折中，结合应用层代码开发策略和技巧，完成系统的实时性要求。

大多数嵌入式实时操作系统采用了微内核结构，即内核只提供诸如资源管理、内存管理和任务调度等基本功能，而网络通信功能、文件系统、GUI 系统等应用组件均驻留于用户进程（任务）中，或以函数调用的方式工作。用户可以根据实际工程的需求适当裁剪，选用相应的组件。嵌入式操作系统与通用操作系统类似，具有管理系统资源、物理层抽象的功能，能够提供库函数、驱动程序、开发工具集等。但与通用操作系统相比，嵌入式操作系统在系统实时性、硬件依赖性、软件固化性以及应用专用性等方面，具有更加鲜明的特点。目前，嵌入式系统的主流趋势是 32 位嵌入式微处理器或微控制器与实时多任务操作系统的结合，嵌入式系统往往指的是包含这种资源的系统。

需要指出的是，本书中涉及的“微处理器”和“微控制器”虽无本质上的区别，但从其组成要素及设计特色来讲，可以总结为如下几点：

❑ 硬件结构

微处理器是一个单芯片的处理器，而微控制器则在一块集成电路芯片中集成了处理器和其他片内外设，功能相对健全，基本构成了一个完整的微型计算机系统。片内外设通常包括 RAM、ROM、串行接口、并行接口、计时器和中断管理等。微控制器具有体积小、功耗低、结构简单、可靠性高、使用方便等特点，应用领域遍及国防、工业、汽车、医疗设备和消费电子等几乎所有的行业和领域。此外，作为面向控制的设备，为了保证系统的实时性，微控制器必须拥有强大的中断功能，能够及时响应外界的激励，快速执行上下文切换，因此微控制器在片上集成了所有处理中断必需的电路。

❑ 应用领域

微处理器以微型计算机系统应用为设计初衷，适用于在计算机系统中处理信息。这也是微处

理器的优势所在。微控制器通常用于面向控制的应用，其系统设计追求小型化、低功耗，系统设计时尽可能减少元器件数量，适用于那些以极少的元件实现对输入/输出设备进行控制的场合。

❑ 指令集特征

由于应用场合不同，微控制器和微处理器的指令集也有所不同。微处理器的指令集增强了处理功能，使其拥有强大的寻址模式和适于操作大规模数据的指令。另外，微处理器还具有其他特点，如用户程序中无法使用特权指令等。微控制器的指令集适用于输入/输出控制。许多输入/输出的接口是以位为单位寻址的。而微处理器通常不具备这些强大的位操作能力，因为设计者在设计微处理器时，仅考虑以字节或更大的单位来操作数据。

❑ 处理能力

就处理能力而言，微处理器要远高于微控制器的处理水平，微控制器以牺牲处理能力来实现其他片内外设和功能。微控制器受限于片内资源（ROM、RAM），它的指令必须非常精简，大部分指令的长度都短于1个字节。微控制器指令集的基本特点就是具有精简的编码方案。而微处理器庞大的指令集使得其编码要远比微控制器复杂。

本书基于意法半导体公司的32位嵌入式微控制器，以FreeRTOS为嵌入式实时操作系统，讲述了FreeRTOS的移植及工程应用开发的全过程。本书定位于已熟悉STM32寄存器的操作，掌握了基于STM32官方驱动库的前后台模式应用软件开发的读者或初级嵌入式软件开发工程师，并希望在此基础上向读者展示嵌入式操作系统和TCP/IP协议的移植方法和要点。本书第四篇以一个完整工程项目为导引，向读者展示一个基于STM32F107硬件平台的嵌入式实时操作系统移植开发过程。读者在学习完本书内容后，将能够独立进行基于嵌入式实时操作系统和TCP/IP协议的移植和应用系统开发，并能够完成从初级软件开发者到高级开发者的转变。

前　言

自20世纪90年代，鉴于多任务支持、开发便捷、便于维护等特性，同时能够提高系统的稳定性和可靠性，嵌入式实时操作系统（RTOS）逐渐为广大嵌入式从业人员所接受和认可，越来越多的工程师加入使用RTOS的队伍。

与此同时，半导体技术的快速发展及市场需求的多样化对RTOS提出了更高的要求。一方面，新型处理器的大量涌现要求RTOS自身结构的设计应易于移植，以适应不同硬件架构平台的应用。另一方面，人们在使用RTOS进行系统设计的同时，不仅希望得到供应商的技术支持，而且希望获得RTOS的源代码，以便对RTOS做出符合工程实际需求的裁剪，并降低硬件平台的构建成本。如通常裁剪后的内核对ROM、RAM的容量占用量更小，用户可以选择更小容量的存储器以降低成本。为了适应这种市场需求，许多RTOS提供商在出售RTOS时附加了源程序的代码，在众多的RTOS供应商中也不乏免费开放源代码的RTOS。本书以一款轻量级开源RTOS为样本，通过适当的例程阐述了嵌入式实时操作系统任务管理、时间管理、资源共享、内存管理等机制，介绍了RTOS内核及TCP/IP协议栈的移植和具体使用方法。

本书内容及读者对象

全书分为四篇，共13章。第一篇（第1～5章）讲述了ARM处理器的发展沿革及技术特点、基于STM32的硬件平台、开发环境的搭建及在工程应用中的选型要点。同时，结合笔者的工程开发经验，简要介绍了成为一名合格嵌入式软件开发工程师应该具备的工程素养。对于已熟悉ARM硬件知识的读者，可跳过第1～3章的内容。第二篇（第6～8章）简要介绍了嵌入式实时操作系统的基本概念，重点讲述了一个轻量级嵌入式实时操作系统的使用方法，并给出了基于STM32微控制器的移植代码实现。第三篇（第9～11章）结合TCP/IP原理介绍LwIP开源轻量级协议栈的基本原理及常见应用模式，第11章实现了前后台模式下的TCP/IP协议栈移植，并给出了源代码。第四篇（第12～13章）首先介绍了在STM32F107微控制器上移植FreeRTOS和LwIP的全过程，随后介绍了工业通信网关的一般实现方式，作为示例，简要实现了以太网实现通信报文的转发和板载资源的控制。

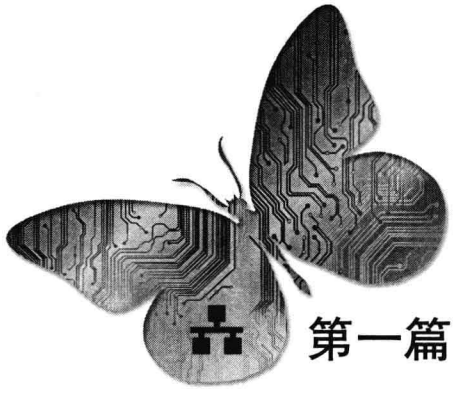
本书以已熟悉STM32寄存器的操作、掌握基于STM32官方驱动库的前后台模式应用软件开发的读者或初级嵌入式软件开发工程师为对象，向读者讲述了嵌入式实时操作系统和TCP/IP协议栈的移植方法和要点，并从系统工程的角度简要介绍了嵌入式工程开发的一般步骤和方法。

致谢

本书的顺利完稿与出版离不开本书的编辑张国强先生的鼓励与支持，他对书稿提出的专业而宝贵的建议使得成书的质量更进一步。笔者对他在书稿审阅和校对过程中付出的辛勤劳动表示衷心的感谢。本书在编写过程中参考了大量书籍和资料，参考文献中未能将其一一列出，在此对书中参考资料的作者一并表示感谢。最后，感谢广大的读者朋友，感谢您花费时间和精力阅读本书，由于水平有限书中难免存在疏漏与错误，诚恳地希望您批评指正。

目　录

第四篇 移植篇

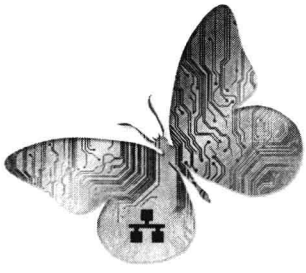


第一篇

平 台 篇

本篇主要介绍基于 STM32 的硬件平台、开发环境的搭建及在工程应用中的选型等要点。

- 第1章　ARM处理器简介
- 第2章　基于STM32F107的开发板
- 第3章　开发环境
- 第4章　编程规范
- 第5章　项目规划



第 1 章
ARM 处理器简介

本章主要讲述 ARM 系列处理器的发展沿革，从处理器技术特点的角度对各系列处理器进行介绍，着重讲述了 Cortex 内核处理器的技术特点。最后，简要介绍了意法半导体公司的以 Cortex-M3 为内核的互联型控制器。

1.1 ARM 内核处理器沿革

ARM（Advanced RISC Machines）是微处理器行业的一家知名企业，1991 成立于英国剑桥，该公司主要出售芯片设计技术的授权。人们将采用 ARM 技术知识产权（IP）核的微处理器称为 ARM 微处理器。ARM 公司利用独特的商业模式在全球范围内拥有极其广泛的合作伙伴。ARM 公司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 ARM 公司提供的一套独一无二的 ARM 相关技术及服务，这些合作伙伴又保证了大量的开发工具和丰富的第三方资源。利用这种合作关系，ARM 公司很快成为许多全球性 RISC 标准的缔造者。

以 ARM 为处理器的应用领域已遍及工业控制、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，基于 ARM 技术的微处理器应用约占据了 32 位 RISC 微处理器 75 % 以上的市场份额。世界上的主流半导体厂商已先后发布了基于 ARMv4、ARMv5、ARMv6、ARMv7 四个架构的 10 多个家族的主流嵌入式微控制器和微处理器产品，其中包括以经典的 ARM7、ARM9、ARM11 为内核的和以最新发布的 Cortex 为内核的种类繁多的微控制器和微处理器，如图 1.1 所示。目前，以 ARM 为内核的处理器大概有上千种，接下来简要介绍 ARM 处理器中的几个重要内核版本。

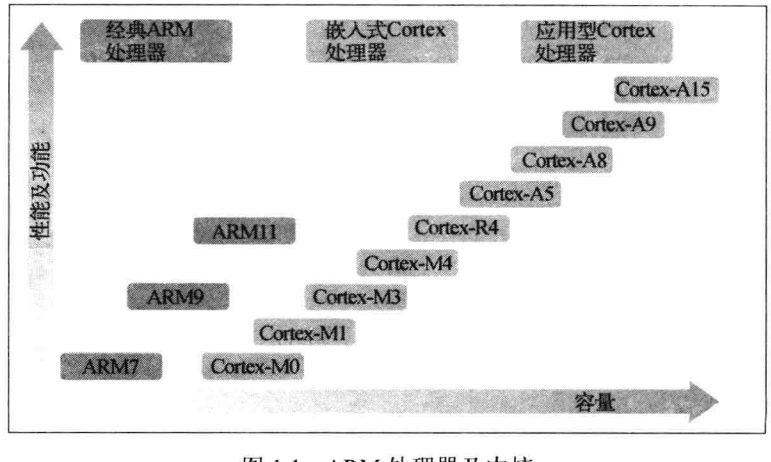


图 1.1 ARM 处理器及内核

提示　为方便起见，若无特别说明，后面内容对处理器和控制器不做详细区分，读者可根据上下文含义理解。对于二者的区别前文有简要的论述，读者可参阅。

1.1.1　传统 ARM 处理器

1. ARM7 处理器

ARM7 处理器采用了 ARMv4T（冯•诺依曼）体系结构。这种体系结构将程序指令存储器和数据存储器合并在一起。主要特点就是程序和数据共用一个存储空间，程序指令存储地址和数据存储地址指向同一个存储器的不同物理位置，采用单一的地址及数据总线，程序指令和数据的宽度相同。这样，处理器在执行指令时，必须先从存储器中取出指令进行译码，再取出操作数执行运算。总体来说，ARM7 体系结构的特点是：三级流水处理、空间统一的程序指令与数据 Cache、平均功耗为 0.6mW/MHz、时钟频率为 66MHz、每条指令平均执行 1.9 个时钟周期等。其中 ARM710、ARM720 和 ARM740 均为内置 Cache 的 ARM 核。ARM7 指令集与 Thumb 指令集扩展组合在一起，可以减少内存容量和系统成本。同时，ARM7 还利用嵌入式 ICE 调试技术简化系统设计，并通过一个 DSP 增强扩展来改进性能。ARM7 体系结构是小型、快速、低能耗、集成式的 RISC 内核结构。该产品主要用于早期的数字蜂窝电话和硬盘驱动器等，主流的 ARM7 内核是 ARM7TDMI、ARM7TDMI-S、ARM7EJ-S、ARM720T。市场上用得最多的 ARM7 处理器有 Samsung 公司的 S3C44B0X 与 S3C4510、Atmel 公司的 AT91FR40162 系列、Cirrus 公司的 EP73xx 系列等。目前，市场上以 ARM7 为处理器的产品大多已升级为 Cortex 内核的处理器。

2. ARM9、ARM9E 处理器

ARM9 处理器采用 ARMv4T（哈佛）体系结构。这种体系结构将程序指令存储和数据存储分开，是一种并行体系结构。主要特点是程序和数据存储在不同的存储空间中，即程序存储器和数据存储器。它们是两个相互独立的存储器，每个存储器独立编址，独立访问。与两个存储器相对应的是体系结构中的 4 套总线：程序的数据总线和地址总线，数据的数据总线和地址总线。这种分离的程序总线和数据总线允许在一个机器周期内同时获取指令字和操作数，从而提高了执行速度，使数据的吞吐量提高了一倍。又由于程序存储器和数据存储器在两个分开的物理空间中，因而取指和执行能完全重叠。ARM9 体系结构的特点是：五级流水处理、分离的 Cache 结构、平均功耗为 0.7mW/MHz、时钟频率为 120~200MHz、每条指令平均执行 1.5 个时钟周期。与 ARM7 处理器系列相似，ARM9 中的 ARM920、ARM940 和 ARM9E 处理器均为内置 Cache 的 CPU 核，性能为 132MIPS（120MHz 时钟频率，3.3V 供电）或 220MIPS（200MHz 时钟频率）。ARM9 处理器同时也配备 Thumb 指令集扩展、调试和 Harvard 总线。在生产工艺相同的情况下，性能是 ARM7TDMI 处理器的两倍之多。ARM9 常用于无线设备、仪器仪表、联网设备、机顶盒设备、高端打印机及数码相机应用中。ARM9E 内核是在 ARM9 内核的基础上增加了紧密耦合存储器 TCM 及 DSP 部分。目前主流的 ARM9 内核是 ARM920T、ARM922T、ARM940。市场上相关的处理器有 Samsung 公司的 S3C2510、Cirrus 公司的 EP93xx 系列等。主流的 ARM9E 内核是

ARM926EJ-S、ARM946E-S、ARM966E-S 等。市场上早期的 PDA 多采用此类处理器。

3. ARM10E 处理器

ARM10E 处理器采用 ARMv5T 体系结构，特点是：六级流水处理、采用指令与数据分离的 Cache 结构、平均功耗为 1000mW/MHz、时钟频率为 300MHz、每条指令平均执行 1.2 个时钟周期等。ARM10TDMI 与所有 ARM 核在二进制级代码中兼容，内带高速 32×16 MAC，预留 DSP 协处理器接口。其中的 VFP10（向量浮点单元）为七级流水处理体系结构。ARM10E 中的 ARM1020T 处理器由 ARMl0TDMI、32KB 指令、数据 Cache 及 MMU 部分构成。ARM10E 时钟频率高达 300MHz，指令 Cache 和数据 Cache 分别为 32KB，数据宽度为 64 位，支持多种商用操作系统，主要用于无线设备、数字消费品、成像设备、工业控制、通信和信息系统等领域。主流的 ARM10 内核是 ARM1020E、ARM1022E、ARM1026EJ-S 等。

4. SecurCore 处理器

SecurCore 系列处理器提供了基于高性能的 32 位 RISC 技术的安全解决方案。主要特点是：体积小、功耗低、代码密度大和性能高等。另外，最为特别的就是 SecurCore 系列处理器提供了安全解决方案的支持。SecurCore 系列处理器采用软内核技术，以提供最大限度的灵活性，以及防止外部对其进行扫描探测，提供面向智能卡的和低成本的存储保护单元 MPU，可以灵活地集成用户自己的安全特性和其他协处理器。

5. StrongARM 处理器

StrongARM 处理器采用 ARMv4T 的五级流水处理体系结构，主要有 SA110、SA1100、SA1110 这 3 个版本。另外 Intel 公司基于 ARMv5TE 体系结构的 Xscale PXA27x 系列处理器，与 StrongARM 相比增加了 I/D Cache，并且加入了部分 DSP 功能，更适合于移动多媒体应用。早期市场上的大部分智能手机的核心处理器就是 Xscale 系列处理器。

6. ARM11 处理器

ARM11 处理器采用 ARMv6 体系结构，可以在使用 130nm 代工厂技术、小至 2.2mm^2 芯片面积和低至 0.24mW/MHz 的前提下达到高达 500MHz 的性能表现。ARM11 处理器系列以众多消费产品市场为目标，推出了许多新技术，包括针对媒体处理的 SIMD、用以提高安全性能的 TrustZone 技术、智能能源管理（IEM），以及需要非常高的、可升级的超过 2600 Dhrystone 2.1 MIPS 性能的系统多处理技术。主要的 ARM11 处理器有 ARM1136JF-S、ARM1156T2F-S、ARM1176JZF-S、ARM11 MCORE 等多种。

1.1.2 Cortex 内核处理器

ARM 公司将经典处理器 ARM11 以后的产品命名为 Cortex，分成 A、R 和 M 三个系列，旨在为各种不同的市场提供服务。Cortex 系列属于 ARMv7 体系结构，这是 ARM 公司最新的指令集体系结构。ARMv7 体系结构定义了三大分工明确的系列：A 系列面向尖端的基于虚拟内存的操作系统和用户应用；R 系列针对实时系统；M 系列针对微控制器。由于应用领域不同，基于 ARMv7 体系结构的 Cortex 处理器系列所采用的技术也不相同，基于 ARMv7A 的称为 Cortex-A 系列，基