新编高等院校计算机科学与技术规划教材

嵌入式开发技术

原理与实践

郑 伟 编 著

QIANRUSHI KAIFA JISHU YUANLI YU SHIJIAN

北京邮电大学出版社
www.buptpress.com

新编高等院校计算机科学与技术规划教材

嵌入式开发技术原理与实践

郑 伟 编 著

北京邮电大学出版社
·北京·

内 容 简 介

本书作为嵌入式实验课的教材，着眼于嵌入式系统开发基本原理和实际开发过程的讲解，内容涉及嵌入式系统的发展、组成、特点、开发流程，ARM 系列微处理器的基本原理和嵌入式 Linux 开发的关键技术，着重介绍了 ARM 微处理器的体系结构和产品系列、嵌入式 Linux 内核体系结构、嵌入式 Linux 的剪裁以及嵌入式 Linux 开发环境，实际选取基于 Motorola i. MX 系列嵌入式处理器 MC9328MX1/MXL 的应用开发系统，详细讲述了 ADS 开发板的使用和基于嵌入式 Linux 操作系统的开发过程，包括嵌入式 Linux 应用程序编写、驱动程序编写，详细介绍了从交叉编译环境的建立、内核编译、根文件系统的生成、Bootloader 映像的烧写、内核映像和文件系统的烧写到系统的运行及调试等嵌入式系统软件开发的详细过程。本书给出了二十个实验，详细描述了具体的开发步骤，使读者在学习基本的嵌入式开发原理的同时，能从具体的实验中得到切实的实践和体验，掌握嵌入式开发的关键技术。

本书可作为 Motorola i. MX 系列嵌入式处理器应用开发系统开发的参考手册、嵌入式系统开发技术的教学用书，以及嵌入式系统应用设计人员的参考用书。

图书在版编目(CIP)数据

嵌入式开发技术原理与实践/郑伟编著. --北京：北京邮电大学出版社，2010.8

ISBN 978-7-5635-2357-3

Ⅰ. ①嵌… Ⅱ. ①郑… Ⅲ. ①微处理器—系统设计 Ⅳ. ①TP332

中国版本图书馆 CIP 数据核字(2010)第 158061 号

书　　名：嵌入式开发技术原理与实践

编 著 者：郑　伟

责任编辑：陈岚岚

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路 10 号(邮编：100876)

发 行 部：电话：010-62282185　传真：010-62283578

E-mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：北京忠信诚胶印厂

开　　本：787 mm×1 092 mm　1/16

印　　张：9.75

字　　数：241 千字

印　　数：1—2 000 册

版　　次：2010 年 8 月第 1 版　2010 年 8 月第 1 次印刷

ISBN 978-7-5635-2357-3　　定　价：18.00 元

前 言

本书是北京邮电大学研究生必修课“微机与嵌入式技术实验”课程的实验指导书，是北京邮电大学研究生教育研究项目“研究生实验教学模式的研究与实践”的重要成果之一。

本书不但讲解了嵌入式系统开发的原理，更多的是对实际应用开发系统的实践。本书选取了基于 Motorola i. MX 系列嵌入式处理器的 ADS 开发板，基于此开发系统详细讲述了二十个实验的具体操作，并给出了程序示例，使读者在学习基本的嵌入式开发原理的同时，能从具体的实验中得到切实的实践和体验，掌握嵌入式开发关键技术。

全书共分 6 章，各章节内容安排如下。

第 1 章介绍嵌入式系统概念、特点、发展历程，着重讲述嵌入式系统的组成和嵌入式系统开发关键技术，包括开发流程、交叉编译、链接、调试以及系统测试。

第 2 章介绍嵌入式微处理器中的 ARM 微处理器的特点、体系结构，重点对 ARM 微处理器的编程模型、指令系统进行介绍并对其产品系列进行总结，使读者对 ARM 微处理器技术的发展有一个总的了解。

第 3 章重点介绍嵌入式 Linux 操作系统，包括嵌入式 Linux 内核体系结构、嵌入式 Linux 的剪裁以及嵌入式 Linux 开发环境。

第 4 章详细介绍基于 Motorola i. MX 系列嵌入式处理器的 ADS 开发板及集成开发环境，包括 ADS 开发板的组成、配置、对外接口以及集成开发环境的使用。

第 5 章介绍基于 Motorola i. MX 系列嵌入式处理器的应用开发系统，详细讲述 ADS 开发板的使用和基于嵌入式 Linux 操作系统的开发过程，包括嵌入式 Linux 应用程序编写、驱动程序编写，详细介绍了从交叉编译环境的建立、内核编译、根文件系统的生成、Bootloader 映像的烧写、内核映像和文件系统的烧写到系统的运行及调试等嵌入式系统软件开发的详细过程。

第 6 章详细讲述实验的具体操作，给出具体的开发步骤和程序示例，使读者在学习基本的嵌入式开发原理的同时，能从具体的实验中得到切实的实践和体验，掌握嵌入式开发的关键技术。

本书在编写过程中，研究生苏东明、梁辉、梁瑞参与了部分资料的搜集、翻译和整理工作。温向明教授在北京邮电大学研究生教育研究项目“研究生实验教学模式的研究与实践”中对本教材的编写给予了很多指导性的意见，没有温教授的推动和激励，本书是不可能完成的。还有很多同志为本书的出版提供了很大的帮助，在此一并表示感谢。

由于作者水平有限，错误和不妥之处敬请读者批评指正。

目 录

第1章　嵌入式系统概述

嵌入式系统(Embedded System)是继 IT 网络技术之后，又一个热门的技术发展方向。由于嵌入式系统具有体积小、性能强、功耗低、可靠性高的突出优势，目前已被广泛地应用于军事、国防、消费电子、网络通信、工业控制等领域，具有广阔的市场前景。随着计算机技术与通信技术的发展，嵌入式系统的研究与开发也有着越来越重要的实际意义。

1.1　什么是嵌入式系统

简单地讲，嵌入式系统指操作系统和功能软件集成于计算机硬件系统之中。

广义地讲，凡是带有微处理器的专用软硬件系统都可以称为嵌入式系统。

嵌入式系统一般被定义为：以应用为中心，以计算机技术为基础，软件硬件可裁剪，适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

定义要素包括：

- 专用计算机系统(非 PC 智能电子设备)；
- 以应用为中心；
- 以计算机技术为基础；
- 软件硬件可裁剪；
- 适应应用系统对功能、可靠性、成本、体积、功耗严格要求。

和通用计算机不同，嵌入式系统的硬件和软件都必须高效率地设计，量体裁衣、去除冗余，力争在同样的硅片面积上实现更高的性能；要针对用户的具体需求，对芯片配置进行裁剪和添加，以达到理想的性能。这样才能在具体应用对处理器的选择面前更具有竞争力。

1.2　嵌入式系统的特点

(1) 用于特定的任务、专用性强。

嵌入式系统的开发一定是用于某种特定的任务，专用性很强，其中的软件系统和硬件的结合非常紧密，一般要针对硬件进行系统的移植，而且针对不同的任务，往往需要对系统进行较大更改，程序的编译下载要和系统相结合。这种修改和通用软件的“升级”是完全不同的概念。

(2) 系统内核小。

由于嵌入式系统一般是应用于小型电子装置的，系统资源相对有限，所以内核较之传统

的操作系统要小得多。比如 ENEA 公司的 OSE 分布式系统，内核只有 5 KB，而 Windows 的内核则要大得多。

(3) 极其关注成本。

嵌入式系统一般要求系统设计精简，一般没有系统软件和应用软件的明显区分，不要求其功能设计及实现上过于复杂，这样一方面利于控制系统成本，另一方面也利于实现系统安全。

(4) 大多有功耗的要求。

(5) 有实时的要求。

与通用系统软件代码相比，嵌入式系统的软件代码要求具有更高质量、可靠性和实时性。这是嵌入式软件的基本要求，而且软件要求固化存储，以提高速度。嵌入式系统的应用程序可以没有操作系统，直接在芯片上运行，但多数嵌入式应用要求操作系统具有实时处理能力。在多任务嵌入式系统中，对重要性各不相同的任务进行合理调度是保证每个任务能够及时执行的关键，单纯通过提高处理器速度是低效的，甚至是无法实现的，必须有一个高效的多任务实时操作系统来支撑。为了合理地调度多任务，利用系统资源、系统函数，需选配 RTOS(Real-Time Operating System)开发平台，这样才能保证程序执行的实时性、可靠性，也有助于缩短开发时间，保障软件质量。

(6) 软件要求固化存储。

为了保证系统的高效性和可靠性，嵌入式系统中的软件一般都固化在只读存储器中，而不是以磁盘为载体、随意更换，因此嵌入式系统的应用软件生命周期也和嵌入式产品一样长。

(7) 软件硬件可靠性更高。

软件代码要求高效和高可靠性。在大多数嵌入式应用中，存储空间是很宝贵的，同时还有实时性要求，为此对嵌入式操作系统的质量要求很高，因此要求程序编写和编译工具的质量要高，以减少程序二进制代码长度、提高执行速度。

(8) 嵌入式系统开发需要开发工具和环境。

由于嵌入式系统本身不具备自主开发能力，即使设计完成以后，用户通常也不能对其中的程序功能进行修改，必须有一套开发工具和环境才能进行开发。这些工具和环境一般是基于通用计算机上的软硬件设备以及各种逻辑分析仪、混合信号示波器等。

嵌入式系统的开发通常采取“宿主机(host)＋目标机(target)”的模式，宿主机运行仿真或调试软件，目标机运行应用程序，二者通过仿真器或其他通信方式连接。因此在嵌入式系统开发中一套稳定可靠的开发板是非常必要的，特别是在设计的硬件电路没有调试好之前。

图 1-1 所示为采用 JTAG 仿真器的一种开发方式，JTAG 仿真器一端接在主机的并口，另一端接开发板的 JTAG 口。这种方式通常用在硬件调试初期，操作系统还没有跑起来的时候。

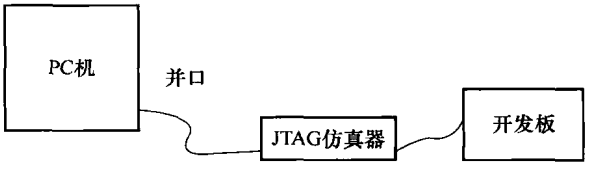

图 1-1 无操作系统时的开发模式

对于本书要讲述的 ADS 开发板来说，在调试初期还可以采用图 1-2 所示的调试方式，这种方式利用 Dragon Ball 处理器的 Bootstrap 模式，将主机和开发板的串口直接相连，利用超级终端发送特定命令即可读写寄存器、运行程序，是一种比较简便的方式。Bootstrap 模式是指：允许用户通过串口(UART)初始化系统和加载程序及数据到系统存储器中，并接收执行命令来运行存储在系统存储器中的程序。如果没有购买仿真器，这种方式也是一种很好的选择。

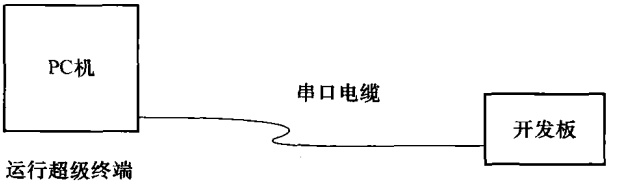

图 1-2 使用 Bootstrap 的开发模式

如果硬件已经调试完毕，正准备调试应用软件，可以采用图 1-3 所示的方式。这种方式下，主机端可以运行 KDevelop＋Insight 集成调试环境软件，通过串口或以太网接口与开发板通信。开发板端运行 Linux 操作系统和 GDB Server 软件，二者通过 GDB 协议调试应用程序，支持单步运行、断点等模式，可以实时查看寄存器、变量的值。当使用以太网接口时，调试速度也是比较快的，不失为一种价廉物美的开发模式。

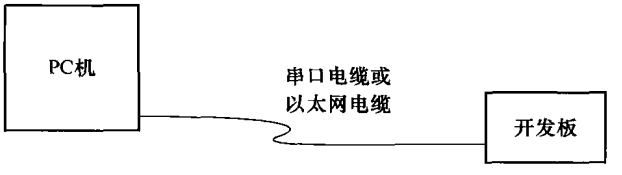

图 1-3 有操作系统时的开发模式

1.3 嵌入式系统的发展

虽然嵌入式系统是近几年才开始真正流行起来的，但嵌入式的概念早已存在。从 20 世纪 70 年代单片机的出现到今天各种嵌入式微处理器、微控制器的广泛应用，嵌入式系统经历了近四十年的历史。纵观嵌入式系统的发展历程，大致经历了以下 4 个阶段。

(1) 无操作系统阶段

嵌入式系统最初的应用是基于单片机的，大多以可编程控制器的形式出现，具有监测、伺服、设备指示等功能，通常应用于各类工业控制和飞机、导弹等武器装备中，一般没有操作系统的支持，只能通过汇编语言对系统进行直接控制，运行结束后再清除内存。这些装置虽然已经初步具备了嵌入式的应用特点，但仅仅只是使用 8 位的 CPU 芯片来执行一些单线程的程序，严格地说还谈不上“系统”的概念。这一阶段嵌入式系统的系统结构和功能相对单一，处理效率较低，存储容量较小，而且几乎没有用户接口。

此阶段嵌入式设备中没有操作系统，其主要原因是：首先，诸如洗衣机、微波炉、电冰箱这样的设备仅仅需要一个简单的控制程序，以管理数量有限的按钮和指示灯，没有使用操作

系统的必要；其次，硬件资源相对更加有限，不足以支持一个操作系统。由于这种嵌入式系统使用简便、价格低廉，因而曾经在工业控制领域中得到了非常广泛的应用，但已逐渐无法满足现今对执行效率、存储容量都有较高要求的信息家电等领域的需要。

(2) 简单操作系统阶段

20 世纪 80 年代，随着微电子工艺水平的提高，IC 制造商开始把嵌入式应用中所需要的微处理器、I/O 接口、串行接口以及 RAM、ROM 等部件统统集成到一片 VLSI 中，制造面向 I/O 设计的微控制器。与此同时，嵌入式系统的程序员也开始基于一些简单的操作系统开发嵌入式应用软件，大大缩短了开发周期，提高了开发效率。

这一阶段嵌入式系统的主要特点是出现了大量高可靠、低功耗的嵌入式 CPU，各种简单的嵌入式操作系统开始出现并得到迅速发展。此时的嵌入式操作系统虽然还比较简单，但已经初步具备了一定的兼容性和扩展性，内核精巧且效率高，主要用来控制负载以及控制应用程序的运行。

(3) 实时操作系统阶段

20 世纪 90 年代，在分布控制、数字化通信和信息家电等巨大需求的牵引下，嵌入式系统进一步飞速发展，而面向实时信号处理算法的 DSP 产品则向着高速度、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也不断扩大，逐渐形成了实时多任务操作系统，并开始成为嵌入式系统的主流。

这一阶段嵌入式系统的主要特点是操作系统的实时性得到了很大改善，已经能够运行在各种不同类型的微处理器上，具有高度的模块化和扩展性。此时的嵌入式操作系统已经具备了文件和目录管理、设备管理、多任务、网络和图形用户界面等功能，并提供了大量的应用程序接口，从而使得应用软件的开发变得更加简单。

(4) 面向 Internet 阶段

21 世纪无疑是一个网络时代，随着 Internet 的进一步发展，以及 Internet 技术与信息家电、工业控制等技术的结合日益紧密，嵌入式 Internet 将会成为嵌入式技术的真正未来。

1.4 嵌入式系统的组成

嵌入式系统作为一类特殊的计算机系统，一般包括以下 3 个组成部分：硬件设备、嵌入式操作系统和应用软件，如图 1-4 所示。

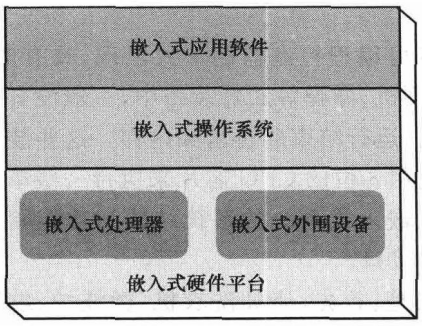

图 1-4 嵌入式系统组成示意图

硬件设备包括嵌入式处理器和嵌入式外围设备。其中的嵌入式处理器是嵌入式系统的核心部分，它与通用处理器最大的区别在于，嵌入式处理器大多工作在为特定用户群所专门设计的系统中，它将通用处理器中许多由板卡完成的任务集成到芯片内部，从而有利于嵌入式系统在设计时趋于小型化，同时还具有很高的性能和可靠性。如今，全世界嵌入式处理器已经超过一千多种，流行的体系结构有三十多个系

统，其中以 ARM、POWER PC、MC68000、MIPS 以及 8051 系列使用最为广泛。嵌入式外围设备是嵌入式系统中用于完成存储、通信、调试、显示灯等辅助功能的其他部件。目前常用的嵌入式外围设备按功能可以分为存储设备（如 RAM、SRAM、Flash 等）、通信设备（如 RS-232 接口、SPI 接口、以太网接口等）和显示设备（如显示屏）三类。图 1-5 为典型的嵌入式系统硬件基本组成示意图。

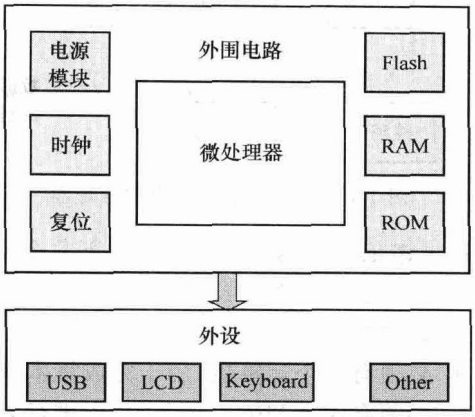

图 1-5　典型的嵌入式系统硬件基本组成示意图

嵌入式操作系统从嵌入式发展的第三阶段起开始引入。嵌入式操作系统不仅具有通用操作系统的一般功能，如向上提供对用户的接口（如图形界面、库函数 API 等），向下提供与硬件设备交互的接口（硬件驱动程序等），管理复杂的系统资源，同时，它还在系统实时性、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。

嵌入式系统的应用软件是针对特定应用领域，基于某一特定的硬件平台，用来达到用户预期目标的专用软件。由于嵌入式系统自身的特点，决定了嵌入式应用软件不仅要满足对准确性、安全性和稳定性等方面的需要，而且还要尽可能进行代码优化，以减少对系统资源的消耗，降低硬件成本。

可以用图 1-6 描述嵌入式系统的软、硬件之间的关系。

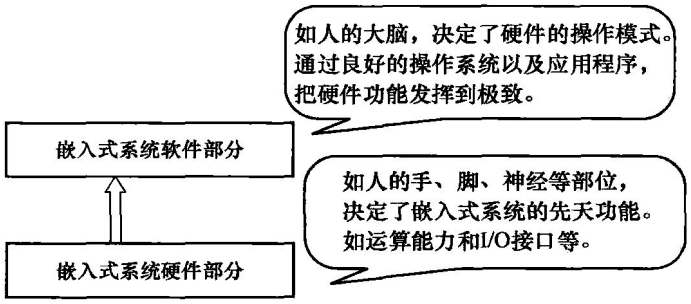

图 1-6　嵌入式系统软、硬件描述

1.4.1　嵌入式处理器

嵌入式系统的核心部件是嵌入式处理器。目前全世界嵌入式处理器的品种总量已经超过一千多种，流行体系结构有三十几个系列，其中 8051 体系占有多半，有 350 多个衍生产品，仅 Philips 就有近一百种。现在几乎每个半导体制造商都生产嵌入式处理器，越来越多的公司有自己的嵌入式处理器设计部门。嵌入式处理器的寻址空间一般从 64 KB 到 16～32 MB，处理速度从 0.1～2 000 MIPS，常用封装从 8～144 个引脚。

在介绍嵌入式处理器的特点和分类之前，首先简单介绍一下相关的基本知识点，包括：冯·诺依曼体系结构与哈佛体系结构、CISC 与 RISC、指令流水线以及存储器系统。

(1) 冯·诺依曼体系结构与哈佛体系结构

1945 年,冯·诺依曼首先提出了“存储程序”的概念和二进制原理,后来,人们把利用这种概念和原理设计的电子计算机系统统称为“冯·诺依曼型结构”计算机。冯·诺依曼结构的处理器使用同一个存储器,经由同一个总线传输,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,因此程序指令和数据的宽度相同,如图 1-7 所示。目前使用冯·诺依曼结构的中央处理器和微控制器有很多,例如,Intel 公司的 8086、ARM 公司的 ARM7、MIPS 公司的 MIPS 处理器都采用了冯·诺依曼结构。

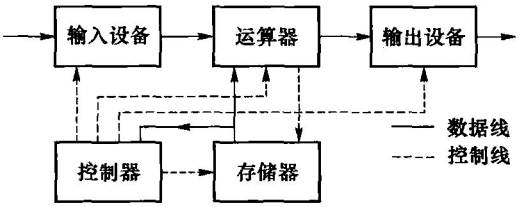

图 1-7 冯·诺依曼结构的处理器

冯·诺依曼结构处理器具有以下几个特点:

- 必须有一个存储器;
- 必须有一个控制器;
- 必须有一个运算器,用于完成算术运算和逻辑运算;
- 必须有输入和输出设备,用于进行人机通信;
- 数据与指令都存储在同一个存储器中。

冯·诺依曼的主要贡献就是提出并实现了“存储程序”的概念。由于指令和数据都是二进制码,指令和操作数的地址又密切相关,因此,当初选择这种结构是自然的。但是,这种指令和数据共享同一总线的结构,使信息流的传输成为限制计算机性能的瓶颈,影响了数据处理速度的提高。

在典型情况下,完成一条指令需要 3 个步骤,即:取指令、指令译码和执行指令。举一个最简单的对存储器进行读写操作的指令,如图 1-8 所示,指令 1 至指令 3 均为存、取数指令,对冯·诺依曼结构处理器,由于取指令和存取数据要从同一个存储空间存取,经由同一总线传输,因而它们无法重叠执行,只有一个完成后再进行下一个。

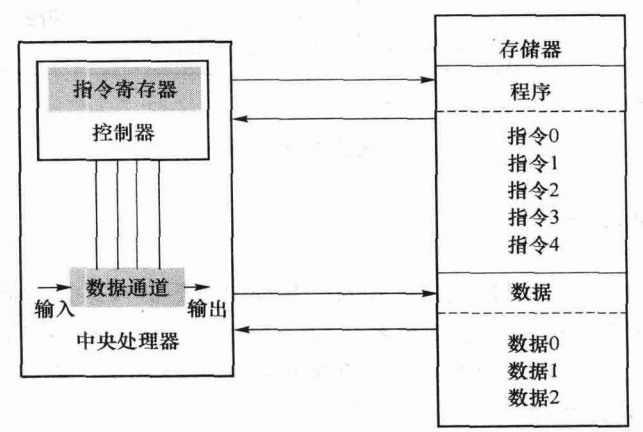

图 1-8 冯·诺依曼体系结构模型

冯·诺依曼体系结构被大多数计算机所采用，例如，ARM7 系列的处理器采用的就是冯·诺依曼体系结构。

与冯·诺依曼体系结构不同，哈佛结构为数据和程序提供各自独立的存储器，程序计数器只指向程序存储器而不指向数据存储器。这样做可以使指令和数据有不同的数据宽度。哈佛体系结构具有如下特点：

- 程序存储器与数据存储器分开；
- 提供了较大的数据存储器带宽；
- 适合于数字信号处理。

图 1-9 为哈佛体系结构模型示意图。哈佛结构的微处理器通常具有较高的执行效率。其程序指令和数据指令是分开组织和存储的，执行时可以预先读取下一条指令。目前使用哈佛结构的中央处理器和微控制器有很多，例如，Microchip 公司的 PIC 系列芯片，Motorola 公司的 MC68 系列，Zilog 公司的 Z8 系列，ATMEL 公司的 AVR 系列和 ARM 公司的 ARM9、ARM10、ARM11 和 Cortex-M3 内核，51 单片机也属于哈佛结构。

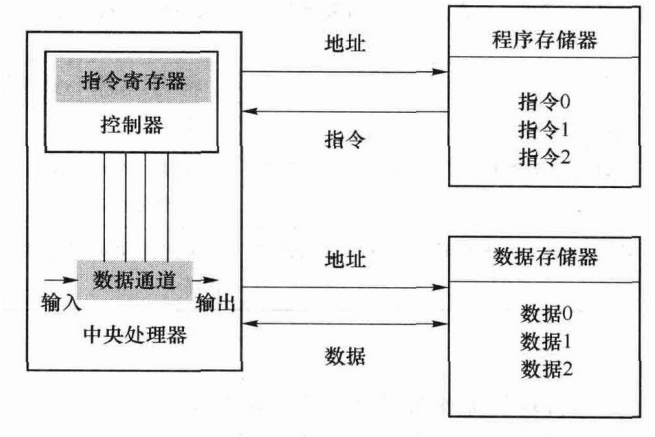

图 1-9　哈佛体系结构模型

(2) 复杂指令集与精简指令集

复杂指令集(Complex Instruction Set Computer，CISC)与精简指令集(Reduced Instruction Set Computer，RISC)是当前 CPU 的两种架构。它们的区别在于不同的 CPU 设计理念和方法。早期的 CPU 全部是 CISC 架构，它的设计目的是要用最少的机器语言指令来完成所需的计算任务。

比如对于乘法运算，在 CISC 架构的 CPU 上，可能需要这样一条指令：

```
MUL ADDRA, ADDRB
```

就可以将 ADDRA 和 ADDRB 中的数相乘并将结果储存在 ADDRA 中。将 ADDRA、ADDRB 中的数据读入寄存器，相乘并将结果写回内存的操作全部依赖于 CPU 中设计的逻辑来实现。这种架构会增加 CPU 结构的复杂性和对 CPU 工艺的要求，但对于编译器的开发十分有利。比如上面的乘法运算，C 程序中的 $a*=b$ 就可以直接编译为一条乘法指令。目前只有 Intel 及其兼容 CPU 还在使用 CISC 架构。

RISC 架构要求软件来指定各个操作步骤。上面的乘法运算如果在 RISC 架构上实现，

将 ADDRA、ADDRB 中的数据读入寄存器，相乘并将结果写回内存的操作都必须由软件来实现，比如：

```
MOV A, ADDRA;
MOV B, ADDRB;
MUL A, B;
STR ADDRA, A;
```

这种架构可以降低 CPU 的复杂性以及允许在同样的工艺水平下生产出功能更强大的 CPU，但对于编译器的设计有更高的要求。

表 1-1 将 CISC 与 RISC 两种架构做一个对比。

表 1-1 CISC 与 RISC 的对比

类 别	CISC	RISC
指令系统	指令数量很多	较少，通常少于 100
执行时间	有些指令执行时间很长，如整块的存储器内容复制；或将多个寄存器的内容复制到存储器	没有较长执行时间的指令
编码长度	编码长度可变，1～15 B	编码长度固定，通常为 4 B
寻址方式	寻址方式多样	简单寻址
操作	可以对存储器和寄存器进行算术和逻辑操作	只能对寄存器进行算术和逻辑操作，Load/Store 体系结构
编译	难以用优化编译器生成高效的目标代码程序	采用优化编译技术，生成高效的目标代码程序

(3) 指令流水线

影响 CPU 性能的因素包括：流水线技术、超标量执行和缓存。

流水线(Pipeline)技术是指：几个指令可以并行执行。一条指令的执行周期包括：取指令(IF t，Instruction Fetch)、指令译码(ID t，Instruction Decode)、执行指令(IE t，Instruction Execute)以及存储(S t，Storage)，因此每条指令的执行周期 T 可以表示为

$$T = t_{IF} + t_{ID} + t_{IE} + t_S$$

采用流水线技术使几个指令可以并行执行，提高了 CPU 的运行效率，也使内部信息流通畅流动。图 1-10 为采用流水线技术指令并行执行的示意图。

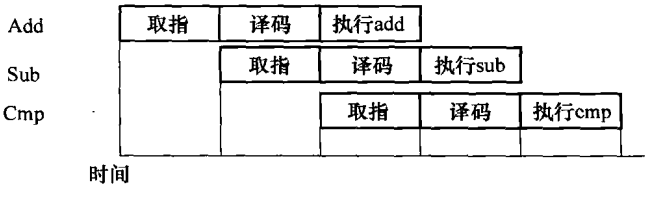

图 1-10 指令流水线技术

除了流水线技术之外，超标量和高速缓存也是影响 CPU 性能的重要因素。其中，超标量(Superscalar)执行是指 CPU 采用多条流水线结构；而高速缓存(Cache)是一种小型、快速

的存储器，它保存部分主存内容的副本，由于微处理器的时钟频率比内存速度提高快得多，通过高速缓存可以提高内存的平均性能。图 1-11 和图 1-12 分别是超标量执行和高速缓存的原理示意图。

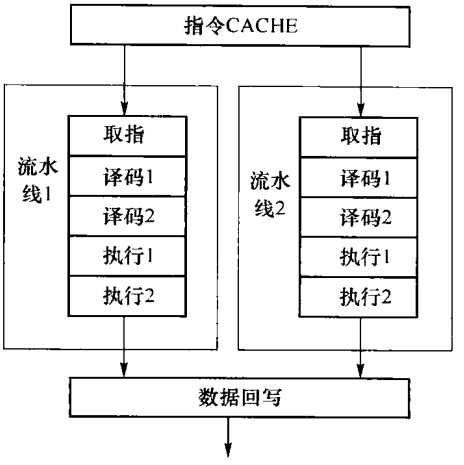

图 1-11　超标量执行示意图

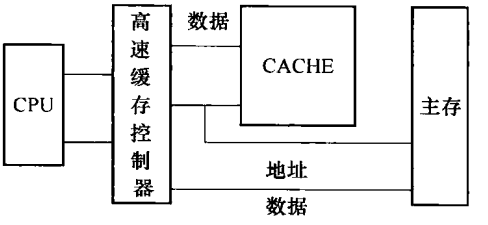

图 1-12　高速缓存工作原理

(4) 存储器系统

图 1-13 表示了按时钟周期分层的存储器系统，时钟周期越长，处理速度越慢；时钟周期越短，处理速度越快。寄存器处理速度最快，位于分层结构图的最上层，而网络存储器、Flash、ROM、磁盘的处理速度相对较慢，位于分层结构图的最下层。

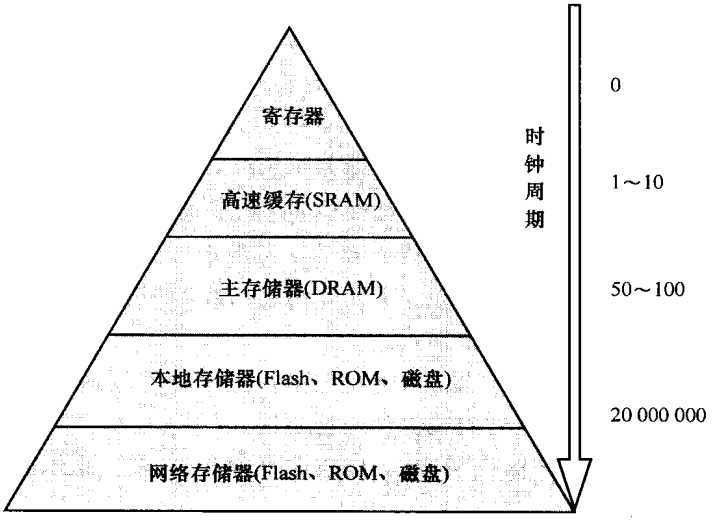

图 1-13　存储器系统分层结构示意图

RAM(随机存取存储器)是英文“Random Access Memory”的缩写，即通常所说的内存，其内容是可读可写的，计算机在正常工作时，存放在其他介质(硬盘、软盘、光盘等)上面的程序都要调到 RAM 中才能运行，内存越大，一次性可以从其他介质上调用的数据越多。内存

如果不够大的话，程序运行时，如果需要从其他介质上调用新的数据，这样，原来调到内存中的数据有部分就要被改写成新数据。因此，内存中的数据，在需要时是可以随机改写的。此外，内存芯片如果没有电源供应的话，那么其中的内容将会全部丢失。而 ROM(只读存储器，Read Only Memory)芯片即使没有电源供应，其中的 BIOS 程序也不会消失。

随机存取存储器又分静态随机存储器(SRAM)和动态随机存储器(DRAM)，SRAM 与 DRAM 相比较而言，SRAM 比 DRAM 快；SRAM 比 DRAM 耗电多；DRAM 存储密度比 SRAM 高得多；DRAM 需要周期性刷新。

EPROM(Erasable Programmable Read Only Memory)，即“可擦除可编程只读存储器”。它是一种可重写的存储器芯片，并且其内容在掉电的时候也不会丢失，是非易失性的。它通过 EPROM 编程器进行编程，EPROM 编程器能够提供比正常工作电压更高的电压对 EPROM 编程。一旦经过编程，EPROM 只有在强紫外线的照射下才能够进行擦除。为了进行擦除，EPROM 的陶瓷封装上具有一个小的石英窗口，这个石英窗口一般情况下使用不透明的膜覆盖，当擦除时将这个膜揭掉，然后放置在强紫外线下大约 20 分钟。主要的 IC 有 27××系列和 27C××系列。

EEPROM(Electrically Erasable Programmable Read Only Memory)，即“电可擦除可编程只读存储器”，并且其内容在掉电的时候也不会丢失。在平常情况下，EEPROM 与 EPROM 一样是只读的，需要写入时，在指定的引脚加上一个高电压即可写入或擦除，而且其擦除的速度非常快，通常 EEPROM 芯片又分为串行 EEPROM 和并行 EEPROM 两种，串行 EEPROM 在读写时数据的输入/输出是通过 2 线、3 线、4 线或 SPI 总线等接口方式进行的，而并行 EEPROM 的数据输入/输出则是通过并行总线进行的。主要的 IC 有 28×× 系列。

由于 ROM 不易更改的特性让更新资料变得相当麻烦，因此就有了 Flash Memory(闪速存储器)的发展，Flash Memory 具有 ROM 不需电力维持资料的好处，又可以在需要的时候任意更改资料，不过单价也比普通的 ROM 要高。相对传统的 EPROM 芯片，这种芯片可以用电气的方法快速地擦写。由于快擦写存储器不需要存储电容器，故其集成度更高，制造成本低于 DRAM。它使用方便，既具有 SRAM 读写的灵活性和较快的访问速度，又具有 ROM 在断电后可不丢失信息的特点，所以闪存技术发展十分迅速。

嵌入式处理器是嵌入式系统的核心。下面介绍嵌入式处理器的特点和分类。

嵌入式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，可以避免在软件模块之间出现错误的交叉作用，同时也有利于软件诊断。

③ 可扩展的处理器结构，以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 嵌入式处理器必须功耗很低，尤其是用于便携式的无线及移动的计算和通信设备中靠电池供电的嵌入式系统更是如此，有些功耗只有毫瓦甚至微瓦级。

嵌入式处理器分为 4 类，可用图 1-14 表示。