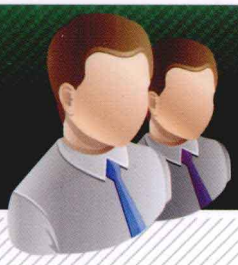
卓越工程师培养计划

■嵌入式系统■

http://www.phei.com.cn

陈志旺　等 编著



STM32

嵌入式微控制器快速上手

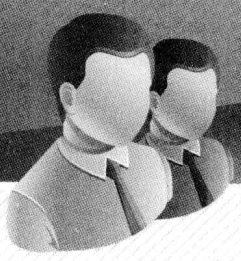
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

卓越工程师培养计划

■ 嵌入式系统 ■

http://www.phei.com.cn

陈志旺　等 编著



STM32 嵌入式微控制器快速上手

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

前　言

嵌入式系统属于一个交叉学科，它涵盖了微电子技术、电子信息技术、计算机软件和硬件等多项技术领域的应用，覆盖面广。目前嵌入式系统发展很快，很多软/硬件技术出现时间不长，掌握这些新技术的人相对较少。很多高校专业划分过细，难以跟上市场变化的步伐，与实际工程项目脱节严重，仍沿用应试教育的教学方式，理论知识讲授过多，动手环节薄弱，理论联系实际能力较差，学生参与社会实践较少，不了解社会需求。嵌入式教学需要相应的嵌入式开发板和软件，需要有经验的人进行开发流程指导，这在目前的高校中是很难实现的。针对上述问题，我们组织多年在一线授课的教师根据教学经验编写了此书。

1. 对教师的建议

把握“三个统一学单片机”的教学理论。

1）一般与特殊的统一　目前电类课程中关于微机方面的课程很多，如《计算机应用基础》、《微机原理》、《单片机原理及应用》、《嵌入式系统原理及应用》、《可编程控制器PLC》等，这些课程间内容既有重复又有联系，因为以奔腾芯片为核心的PC、以8086为核心的第一代PC、51单片机、以ARM为核心的嵌入式系统、PLC都有相同的遗传基因（微机原理）。因此，任课老师都应熟悉上述课程，在课堂讲授中将其融会贯通起来，深入挖掘课程的共性，即微机的基本原理，然后引导学生在学习中侧重其差异，这样不仅可以提高学习效率，还可以启发学生思考。

2）硬件与软件的统一　通过多年的教学实践我们发现，“软件通过控制字寄存器控制硬件”是学生理解的难点，其原因可能是平时同学操作PC直接通过Windows界面达到驱动硬件的目的，没有用软件驱动过底层硬件。因此在介绍嵌入式系统时，最好从硬件控制寄存器程序讲起，让学生对底层原理有个认识。对于STM32，本书在第5章也揭示了控制寄存器和库函数的相关性。另外，在参考文献［2］中，我们将汇编语言依赖硬件的特性深入挖掘，课堂效果较好，推荐采用。

3）内部结构与外部引脚的统一　将内部结构、外部引脚、系统功能、指令集统一起来，只有这样才能做到“庖丁解牛，游刃有余”。

2. 对学生的建议

1）重视实践　工程师解决问题的能力只有从实践中才能获得。从实践经验中归纳出共性的知识，然后再将这些知识重新应用到新的实践中去，这也是当今的大学生要在未来的实际工作中所必须采取的学习和工作方法。只有做到了这一点，才能真正实践以工作为导向的理念：实践、归纳、总结和再实践。学好嵌入式系统，实践必不可少，一定要选一块和微处理器型号对应的开发板，创建一个良好的平台和环境，边实践边学习，尽量弄清其内在原理。硬件开发板的价格不必太高，最好能有自己动手的空间。深入理解STM32的硬件最小系统，对I/O口、串行通信、键盘、LED、LCD、SPI、I^2C、PWM、A/D（包括一些传感器）、D/A等实验逐个实践，逐步理解，再动手制做一个实际的小系统，底层硬件基础就有

了。各个硬件模块驱动程序的编写是嵌入式系统的必备基础。在编程中，主要针对 main. c 和 stm32f10x_it. c 两个文件，其他定义硬件控制寄存器的文件 stm32f10x_map. h、硬件初始化结构体的文件 stm32f10x_xxx. h 和 stm32f10x_xxx. c 也要仔细研读，将软件和硬件联系起来学习，会加深对硬件的理解。把书上的例程亲手输入到计算机中进行实践是很有必要的，一定要多上机操作。程序是抽象的，有时程序能看懂，但自己却不一定能编出来；而有时虽然程序没看懂，但若经常着手去编，就会非常熟悉该程序应该怎样去处理。要把在书中看到的有意义的例子举一反三；经常回顾自己以前写过的程序，并尝试重写，把自己学到的新知识运用进去，温故而知新；当程序写到一半时却发现自己用的方法烦琐，不要马上停手，应尽快将余下的部分粗略地完成，以保证设计的完整性，然后分析自己的错误并重新设计和编写；遇到问题时，不要马上向其他人求救，以免养成依赖性，要学会自己去解决问题。看到一个编程要求后，首先要在头脑中有一个大体的轮廓，独立构思，不要看参考提示，只有这样才可以达到真正的训练目的，才会逐步把思路培养出来；保存好写过的所有的程序——那是最好的积累之一。学习的过程中要经过思考加工，去粗取精，抓本质和精华，仔细研读并尝试修改别人的例程代码，真正将新知识应用到实践中，理解后融入到自己的知识体系中。本书每个知识点后都对应一个开发板实例，推荐以项目开发的方式进行学习，将学习成果变成自己的作品。

2）重视官方文档 学习时，应关注两个比较重要的文档：《STM32F103 × × × 参考手册》，《STM32 固件库使用手册》。

阅读《STM32F103 × × ×参考手册》时，不需要全部阅读，但是前几章必须重点阅读，包括存储器和总线架构，电源控制，备份寄存器，复位和时钟控制，通用和复用功能 I/O，中断和时间等。后续章节讲述的是具体的功能模块设计，如果要用到哪个模块，就可以去阅读相应的模块。

阅读《STM32 固件库使用手册》的主要目的是为了简化编程。STM32 提供了一个非常好的固件函数库，只需调用即可。阅读《STM32 固件库使用手册》时，前面几章也是必须阅读的。比如，第 1 章文档和库规范中的命名规则，编码规则，这些都是需要注意的。第 2 章是最关键的，描述了固件库的架构，以及如何使用固件库等。有了这些基础，就可以借助固件库写出自己的代码了。第 4 章以后的章节都是描述某个模块有什么函数，每个函数如何使用等，可以根据需要来阅读。建议对 GPIO 库函数、中断部分库函数、复位和时钟设置的库函数等内容要重点阅读。

无论何时，官方手册都是最好的老师和帮手，千万不要因为它的枯燥乏味而将其束之高阁。使用任何外设前，都必须仔细看参考手册和使用手册。

3）重视交流 在网上建个交流平台，或者开博，或者社区交流，把自己解决的问题和经验与他人分享，这样通常会让读者个人的研究更具可行性和更深入地了解；构建或参与技术圈子，一个好的圈子通常会给读者带来钻研和共同提高的激情，或者说是一个良性的竞争环境。

3. 本书特点

在讲述具体内容时，本书各章节均以我们开发的 STM32 开发板为硬件教具，每章均会提供一个设计任务实例，以任务为驱动，通过“学中做、做中学”，即 DIY（Do It Yourself）和 LBD（Learning By Doing）的方式，介绍和讲解所需要用到的新知识、新技能，按照认识

的规律学习和掌握基于 ARM Cortex－M3 内核的 STM32 嵌入式微控制器技术及其应用编程，尽量避免纯理论性描述带给读者的枯燥感。有别于数据手册式的教材，本书并没有面面俱到地谈及 Cortex－M3 的技术细节，各个章节也没有繁冗的寄存器说明（参见 ST 公司网页上的数据手册或本书配套资料），而且涉及微机原理的相关知识都在每章“本章前导知识”中给出（可阅读参考文献［2］），每章的例程只给出关键代码，这样做的目的旨在突出重点。本书写作过程中，也注意了软、硬件的结合，将 STM32 的内部结构、控制寄存器和库函数对应结构体，还有功能特点和初始化设置结构体结合起来，揭示软、硬件之间的联系，使读者能够对 STM32 应用“快速上手”。本书不仅是教给大家 STM32 基础知识，更重要的是介绍嵌入式系统的学习方法，启发读者的创意思维。

本书受全国教育科学规划 2010 年度教育部重点课题（课题批准号：GKA103004）的子课题（立项编号：GKA10105）资助。第 1、2、7、8、9、10、11、12 章及附录由燕山大学陈志旺编写，第 3 章由燕山大学刘宝华编写，第 4 章由燕山大学王荣彦编写，第 5 章由燕山大学程淑红编写，第 6 章由燕山大学吕宏诗编写。全书由陈志旺统稿。参加本书编写的还有李萌、薛佳伟、王敬、吴晨、李晓、李坤、赵春媛、刘砚、赵晓娟和李江艳。书中引用了一些网上文献，无法一一注明出处，在此向原作者表示感谢！

由于笔者水平有限，书中难免存在错误与不妥之处，欢迎读者朋友不吝赐教！来信可与如下邮箱联系：czwaaron@ysu.edu.cn。博客：http://blog.sina.com.cn/gksupermarket，本书的进一步资料会在博客上更新。

编著者

目　录

第1章　嵌入式系统概述

1.1　嵌入式系统简介

1. 嵌入式系统定义

嵌入式系统通常定义为以应用为中心，以计算机技术为基础，软件、硬件可剪裁，对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。它主要由嵌入式微处理器、外围硬件设备、嵌入式操作系统及用户应用软件等部分组成，其分层结构如图 1-1 所示，用于实现对其他设备的控制、监视和管理等功能，嵌入式系统通常被嵌入在主要设备之中。

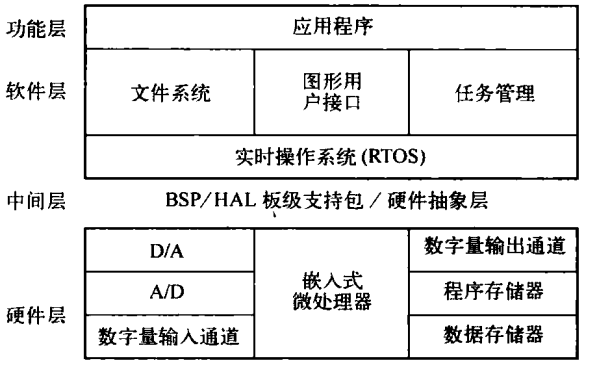


图 1-1　嵌入式系统分层结构

IEEE（国际电气和电子工程师协会）对嵌入式系统定义为：嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置”。

中国微机学会对嵌入式系统定义为：嵌入式系统是以嵌入式应用为目的的计算机系统，可以分为芯片级、板卡级、系统级。芯片级嵌入的是含程序或算法的处理器；板卡级嵌入的是系统中的某个核心模块板；系统级嵌入的是主计算机系统。

国内有学者认为，将一套计算机控制系统嵌入到已具有某种完整的特定功能的（或者将会具备完整功能的）系统内（如各种机械设备），以实现对原有系统的计算机控制，此时将这个新系统称为嵌入式系统。它通常由特定功能模块和计算机控制模块组成，主要由嵌入式微处理器、外围硬件设备、嵌入式操作系统及用户应用软件等部分组成。

从上述定义不难看出，嵌入式系统是一个针对特定的应用而“量身定做”的专用计算机系统，它具有嵌入性、专用性、计算机系统的特点。

施乐公司 Palo Alto 研究中心主任 Mark Weiser 认为：“从长远来看，PC 和计算机工作站将衰落，因为计算机变得无处不在，如在墙里、在手腕上、在手写电脑中（像手写纸一样）等，随用随取、伸手可及”。无处不在的计算机就是嵌入式系统。

2. 嵌入式系统特点

1）通用计算机与嵌入式系统对比 通用计算机与嵌入式系统对比见表 1-1。

表 1-1 通用计算机与嵌入式系统对比

特 征	通用计算机	嵌入式系统
形式和类型	按其体系结构、运算速度和结构规模等因素分为大、中、小型机和微机	形式多样
组成	通用处理器、标准总线和外设，软件和硬件相对独立	面向应用的嵌入式微处理器，总线和外部接口多集成在芯片内部。软件与硬件是紧密集成在一起的
开发方式	开发平台和运行平台都是通用计算机	采用交叉开发方式，开发平台一般是通用计算机，运行平台是嵌入式系统
二次开发性	应用程序可重新编制	一般不能再编程
通用性	通用计算平台	专用系统，用特定设备完成特定任务
资源	较多	跟任务有关，一般较少
程序存储	内存中	ROM
可封装性	看得见的计算机	隐藏于目标系统内部而不被操作者察觉
实时性	不要求实时性	与实际事件的发生频率相比，嵌入式系统能够在可预知的极短时间内对事件或用户的干预做出响应
可靠性	对可靠性要求不高	嵌入式计算机隐藏在系统或设备中，用户很难直接接触、控制，因此一旦工作就要求它可靠运行

从以上对比可以看出，与通用计算机不同，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，这样才能更具有竞争力。嵌入式处理器要根据用户的具体要求，对芯片配置进行裁剪或添加才能达到理想的性能，但同时还受用户订货量的制约，因此不同的处理器面向的用户是不一样的，可能是一般用户、行业用户或单一用户。嵌入式系统和具体用户有机地结合在一起，它的升级换代也是和具体产品同步进行的。嵌入式系统中的软件，一般都固化在 ROM 中，很少以磁盘为载体，所以嵌入式系统的应用软件的生命周期也和嵌入式产品一样长。此外，应用于各行业的嵌入式软件各有其专用化的特点，与通用计算机软件不同，嵌入式系统的软件更强调可继承性和技术衔接性。

2）三个基本 嵌入式系统通常的定义中有以下 3 个基本要素。

- “嵌入性”的特点：由于是嵌入到对象系统中，必须满足对象系统的环境要求，如物理环境（小型）、电气环境（可靠）、成本（价廉）等要求。
- “专用性”的特点：软、硬件的裁剪性；满足对象要求的最小软、硬件配置等。
- “计算机系统”的特点：嵌入式系统必须是能满足对象系统控制要求的计算机系统。与上两个特点相呼应，这样的计算机必须配置有与对象系统相适应的接口电路。

3）从用户方和开发方的角度看嵌入式系统

- 用户方要求：功能简单；专门完成一个或几个任务；要考虑体积、功耗、价格和开发周期等因素；实时与环境交互，如图 1-2 所示；安全可靠，软、硬件的错误不能使系统崩溃。
- 开发方要求：软件硬件协同并行开发；多种多样的微处理器；实时操作系统的多样性（RTOS）；与台式机相比，可利用系统资源很少；应用支持很少；要求特殊的开发工具；调试很容易；软件、硬件都应很“健壮”。

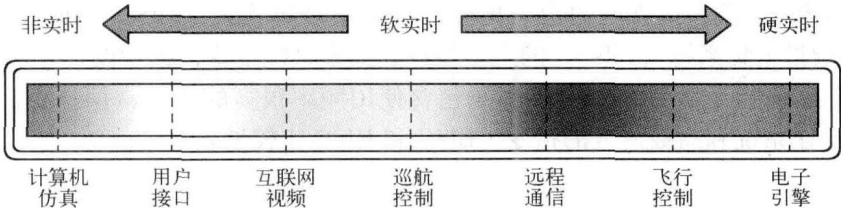


图 1-2　实时频谱图

3. 嵌入式系统分类

1）普林斯顿结构和哈佛结构　普林斯顿结构是由一个中央处理单元（CPU）和单存储空间组成的，即这个存储空间存储了全部的数据和程序，它们内部使用单一的地址总线和数据总线，也称为冯·诺伊曼结构，如图 1-3 所示。这样由于在取指令和取数据时都是通过一条总线分时进行的，所以要根据所给的地址对其进行读/写操作。

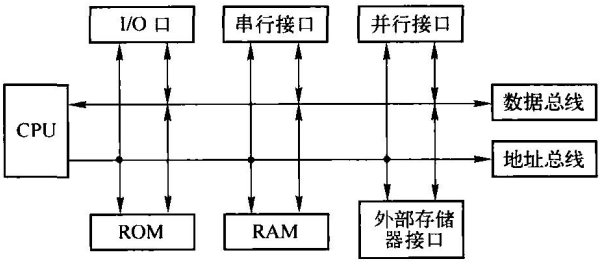


图 1-3　普林斯顿结构示意图

当进行高速运算时，普林斯顿结构计算机不仅不能同时进行取指令和取数据，而且还会造成数据传输通道的瓶颈现象，其工作速度较慢。通常使用的 ARM7 就是属于普林斯顿结构。

哈佛体系结构存储器分为数据和程序两个存储空间，有各自独立的程序总线和数据总线，可以进行独立编址和独立访问，如图 1-4 所示。这样独立的程序存储器和数据存储器为数字处理提供了较高的性能。数据和程序可以并行完成，这使得数据移动更加容易。数据的吞吐量比普林斯顿结构提高了大约 1 倍。

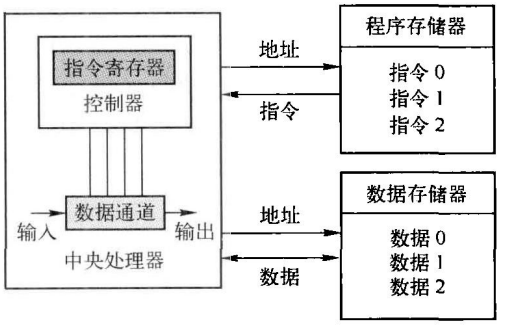


图 1-4　哈佛体系存储系统结构图

目前大部分 DSP 和 ARM9 微处理器都是采用哈佛体系结构。但这种结构的弱点是很难在哈佛机上编写出一个自修改的程序，即写入数据值后，使用这些值作为指令的程序。

2）CISC 和 RISC 计算机的指令集分为复杂指令集系统（CISC）和精简指令集系统（RISC）。复杂指令集系统（CISC）的主要特点是指令系统丰富，程序设计方便，代码短小，执行性能高。精简指令集系统（RISC）只包含使用频率很高的少量常用指令，再提供一些必要的支持操作系统和高级语言的指令。CISC 和 RISC 比较见表 1-2。

表 1-2 CISC 和 RISC 比较

	CISC	RISC
价格	由硬件完成部分软件功能，硬件复杂性增加，芯片成本高	由软件完成部分硬件功能，软件复杂性增加，芯片成本低
性能	减少代码尺寸，增加指令的执行周期数	使用流水线降低指令的执行周期数，增加代码尺寸
指令集	复杂庞大	简单精简
指令周期	不固定	一个周期
编码长度	编码长度可变，1～15 字节	编码长度固定，通常为 4 个字节
高级语言支持	软件完成	硬件完成
寻址模式	复杂的寻址模式，支持内存到内存寻址	简单的寻址模式，仅允许 LOAD 和 STORE 指令存取内存，其他所有的操作都基于寄存器到寄存器
寄存器数目	寄存器较少	寄存器较多
总线结构	普林斯顿结构	哈佛结构
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
应用实例	MCS－51 系列；Motorala（现为 FressScale）的 MC68HC 系列；Atmel 的 AT89 系列；Philips 的 80C51 系列等	Microchip 的 PIC 系列；Zilog 的 Z86 系列；Atmel 的 AT90S 系列；ARM 公司的 ARM 系列等

CISC 技术的复杂性在于硬件，在于微处理器中控制器部分的设计及实现；RISC 技术的复杂性在于软件，在于编译程序的编写和优化。通常较简单的消费类电子设备，如微波炉、洗衣机等，可以采用 RISC 单片机；较复杂的系统如通信设备、工业控制系统等，应采用 CISC 单片机。

随着微处理器技术的进一步发展，CISC 与 RISC 两种体系结构的界限已不再泾渭分明，在很多系统中有融合的趋势。一方面，RISC 设计正变得越来越复杂，例如，超长指令字的提出让一条 RISC 指令可以包含更多信息，同时完成多条传统指令的功能；ARM 微处理器含有普通 ARM 指令和 Thumb 指令两套指令集，以适应嵌入式系统对低功耗、小存储的要求。另一方面，CISC 也在吸收 RISC 的优点，例如，Pentium Ⅱ以后的微处理器在内部实现时也采用 RISC 架构，把复杂的指令在内部由微码通过执行多条精简指令实现。

3）嵌入式系统内核种类 嵌入式微处理器的基础是通用计算机中的 CPU。在应用中只保留和嵌入式应用紧密相关的功能硬件，去除其他的冗余功能部分，这样可以大幅度减小系统体积和功耗。为了满足嵌入式应用的特殊要求，嵌入式微处理器虽然在功能上和标准微处理器基本是一样的，但在工作温度、抗电磁干扰、可靠性等方面进行了增强。

微控制器又称单片机，一般以某一种微处理器内核为核心，芯片内部集成存储器、I/O 接口等各种必要功能，如图 1-5 所示。微控制器的片上外设资源一般比较丰富，适合于控制，因此称微控制器。广义地讲，微控制器产品的作用就是通过预先编制的程序，接收特定的环境参数或用户操作，按照一定的规则控制电信号的变化，再通过各种转换机制把电信号转换成诸如机械动作、光信号、声音信号、显示图像等形式，从而达到智能化控制的目的。

为适应不同的应用需求，一般一个系列的单片机具有多种衍生产品，每种衍生产品的处理器内核都是一样的，不同的是存储器和外设的配置及封装。这样可以使单片机最大限度地与应用需求相匹配，对不同应用进行量体裁衣，从而减少功耗和成本。与嵌入式微处理器相比，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。微控制器是目前嵌入式系统应用的主流。

数字信号处理器（DSP）对系统结构和指令进行了特殊设计，使其适合于执行 DSP 算法，编译效率较高，指令执行速度也较高。在数字滤波、FFT、谱分析等方面 DSP 算法正在大量进入嵌入式领域，DSP 应用正在从通用单片机中以普通指令实现 DSP 功能，过渡到采用嵌入式 DSP 处理器，DSP 外观如图 1-6 所示。推动嵌入式 DSP 处理器发展的一个因素是嵌入式系统的智能化，如各种带有智能逻辑的消费类产品、生物信息识别终端、带有加/解密算法的键盘、ADSL 接入、实时语音压缩解压缩系统、虚拟现实显示等。这类智能化算法一般运算量较大，特别是向量运算、指针线性寻址等较多，而这正是 DSP 处理器的长处所在。嵌入式 DSP 处理器有两个来源，一是传统 DSP 处理器经过单片化和电磁兼容改造，增加片上外围接口成为嵌入式 DSP 处理器，TI 的 TMS320C2000/C5000 等属于此范畴；二是在通用单片机中增加 DSP 协处理器，如 Intel 的 MCS－296 和 Infineon（原 Siemens）的 TriCore 等属于此范畴。

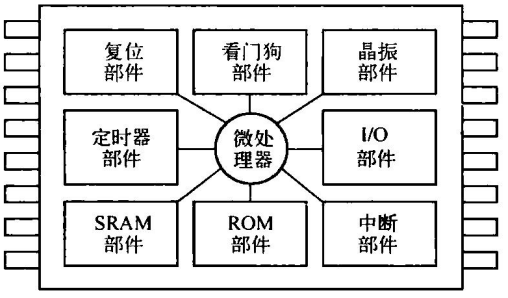


图 1-5　微控制器

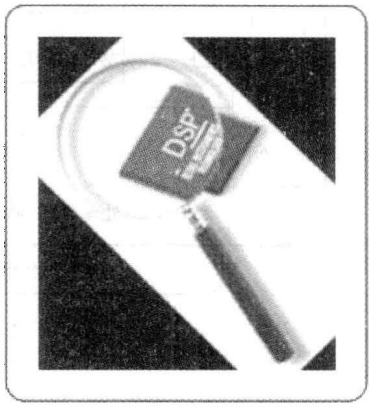


图 1-6　DSP 外观

MPU、MCU 和 DSP 的比较见表 1-3。

表 1-3　MPU、MCU 和 DSP 的比较

	嵌入式微处理器（MPU）	微控制器（MCU）	数字信号处理器（DSP）
定义	由运算器、控制器和寄存器构成的可编程化特殊集成电路	将微处理器、存储器和其他外设接口等集成到一块芯片中	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
优点	对实时多任务有很强的支持能力；具有很强的存储区保护功能；具有可扩展能力	单片化，体积小，功耗和成本低，可靠性高	在信号处理方面有得天独厚的优势
缺点	必须配备 ROM、RAM、总线接口和各种外设接口等	MCU 微控制器，处理速度有限，进行一些复杂的应用很困难	DSP 是运算密集处理器，一般用在快速执行算法，为了追求高执行效率，不适合运行操作系统。核心代码使用汇编
代表	AM186/88、PowerPC	MCS－51、MCS－96、MSP430 等	TI 的 TMS320 系列和 Motorola 的 DSP56000 系列

随着 EDA 的推广和 VLSI 设计的普及化及半导体工艺的迅速发展，在一个硅片上实现一个更为复杂的系统的时代已来临，这就是 System on Chip（SoC）。所谓 SoC 技术，是一种高度集成化、固件化的系统集成技术。使用 SoC 技术设计系统的核心思想，就是针对具体应用，把整个电子应用系统全部集成在一个芯片中，如图 1-7 所示。这些 SoC 芯片是高度集成、没有冗余的，真正体现量体裁衣的特点。SoC 不是各个芯片功能的简单叠加，而是从整个系统的功能和性能出发，用软硬结合的设计和验证方法，利用 IP 复用及深亚微米技术，在一个芯片上实现复杂的功能。各种通用处理器内核将作为 SoC 设计公司的标准库，和许多其他嵌入式系统外设一样，成为 VLSI 设计中一种标准的器件，用标准的 VHDL 等语言描述，存储在器件库中。用户只需定义出其整个应用系统，仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的外部电路或机械部分外，整个嵌入式系统大部分均可集成到一块或几块芯片中去，应用系统 PCB 将变得很简洁。

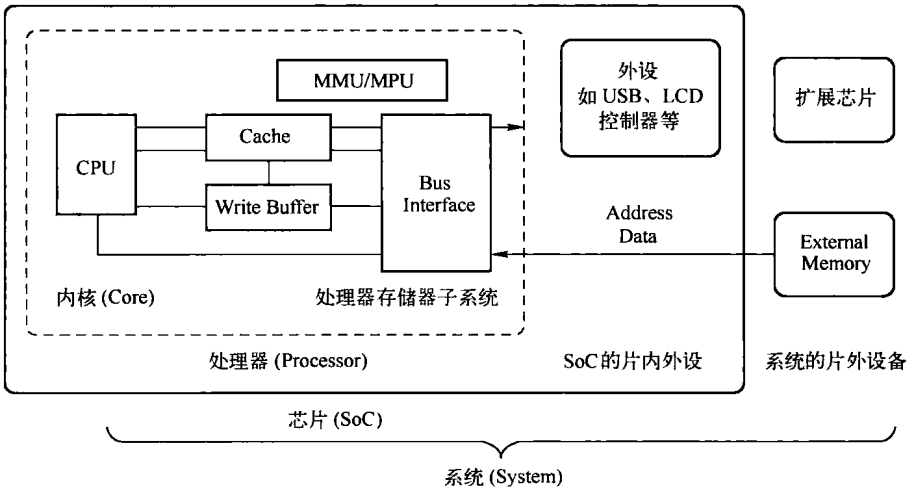


图 1-7　SoC 系统

SoC 具有如下优点。

- 降低耗电量：随着电子产品向小型化、便携化发展，对其省电需求将大幅提升，由于 SoC 产品多采用内部信号的传输，可以大幅降低功耗。
- 减少体积：数颗 IC 整合为一颗 SoC 后，可有效缩小 PCB 上占用的面积，质量轻、体积小。
- 丰富系统功能：随着微电子技术的发展，在相同的内部空间，SoC 可整合更多的功能元件和组件，丰富系统功能。
- 提高速度：随着芯片内部信号传递距离的缩短，信号的传输效率将提升，使产品性能有所提高。
- 节省成本：理论上，IP 模块的出现可以减少研发成本，降低研发时间。不过，在实际应用中，由于芯片结构的复杂性增强，也有可能导致测试成本增加及生产成品率下降。

SoC 可以分为通用和专用两类。通用系列包括 Infineon 的 TriCore、Motorola 的 M－Core、某些 ARM 系列器件、Echelon 和 Motorola 联合研制的 Neuron 芯片等。专用 SoC 一般专用于某个或某类系统中，不为一般用户所知，其代表性的产品是 Philips 的 Smart XA，它将 XA 单片机内核和支持超过 2048 位复杂 RSA 算法的 CCU 单元制作在一块硅片上，形成一个可加载 Java 或 C 语言的专用的 SoC，可用于公众互联网如 Internet 安全方面。

SoC 使应用电子系统的设计技术，从选择厂家提供的定制产品时代进入了用户自行开发设计器件的时代。目前 SoC 的发展重点包括总线结构及互连技术，软、硬件的协同设计技术，IP 可重用技术，低功耗设计技术，可测性设计方法学，超深亚微米实现技术等。

4. 嵌入式系统发展

计算机应用领域的划分如图 1-8 所示。嵌入式系统多属于小型专用型领域。

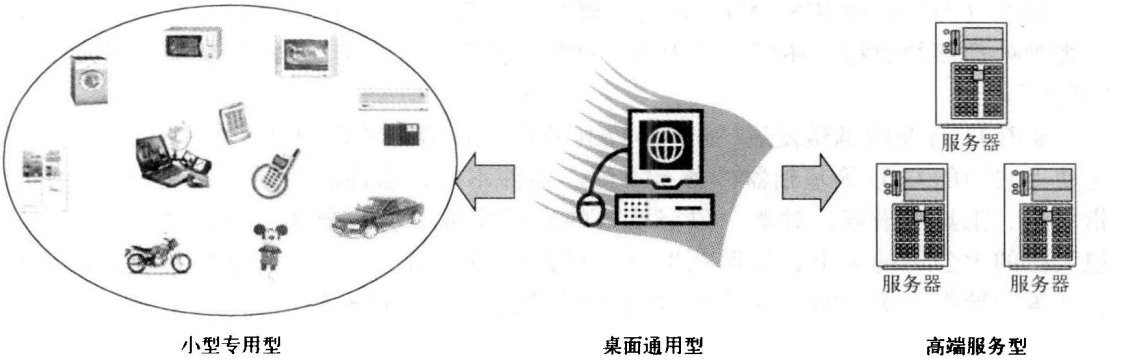


图 1-8　计算机应用领域的划分

嵌入式系统发展主要经历如下 3 个阶段。

（1）20 世纪 70 年代：以嵌入式微处理器为基础的初级嵌入式系统。嵌入式系统最初的应用是基于单片机的。汽车、工业机器、通信装置等成千上万种产品通过内嵌电子装置获得更佳的使用性能。

（2）20 世纪 80 年代：以嵌入式操作系统为标志的中级嵌入式系统。商业嵌入式实时内核包含传统操作系统的特征，开发周期缩短、成本降低、效率提高，促使嵌入式系统有了更为广阔的应用空间。

（3）20 世纪 90 年代：以 Internet 和实时多任务操作系统为标志的高级嵌入式系统。软件规模的不断上升，对实时性要求的提高，使得实时内核逐步发展为实时多任务操作系统，并作为一种软件平台逐步成为国际嵌入式系统的主流。

当前嵌入式系统的特点见表 1-4。

表 1-4　当前嵌入式系统的特点

特　点	实例或描述
经济性	要很便宜，让更多的人能买得起
小型化	携带方便（如笔记本、PDA）
可靠性	能够在一般环境条件下或苛刻的环境条件下运行
实时性	能够迅速地完成数据计算或数据传输
智能性	知识推理、模糊查询、识别、感知运动
实用性	使人们用起来更习惯，对人们有更大的使用价值

嵌入式系统未来发展趋势为：支持联网；精简系统内核、算法，设备实现小尺寸、微功耗和低成本；提供精巧的多媒体人机界面。

1.2 ARM 体系结构及微处理器系列

1. ARM 公司简介

ARM（Advanced RISC Machines），既可以认为是一个公司的名字，也可以认为是对一类微处理器的通称，还可以认为是一种技术的名字，是有力的“胳膊”，也是致命的“武器”。

随着 IT 行业的迅猛发展，Intel、摩托罗拉、TI 等上游厂商都有着不同的数字架构，这使得它们的 CPU 等基础器件各有不同。器件不同，软件就不同，而越来越多不同的指令集、工具和语言，对整个数字技术的发展非常不利。全球工业价值链基本就是大包大揽的大公司的天下，像摩托罗拉这样的公司在测试、制造、系统封装，甚至 CPU 设计等领域都是垄断的。直到 20 世纪 80 年代末，产业链开始出现分工，出现一个更上游的厂商来制定标准，而这个标准的统一，是从数字技术的核心 CPU 开始。这个公司就是 ARM 公司。

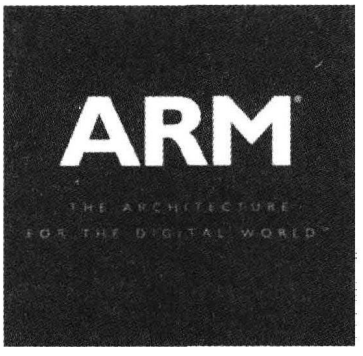


图 1-9 ARM 公司标志

1990 年，英国一位叫 Robin Saxby 的人离开了摩托罗拉与另外 12 名工程师一起开始了创业之旅，于是就有了现在的 ARM 公司。ARM 公司正式成立于 1991 年 11 月，公司标志如图 1-9 所示。ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，作为知识产权（IP）供应商，本身不直接从事芯片生产，而是设计出高效的知识产权内核授权给各半导体公司使用，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，根据各自不同的应用领域，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，如图 1-10 所示。最后由 OEM 客户采用这些芯片来构建基于 ARM 技术的最终应用系统产品。目前，

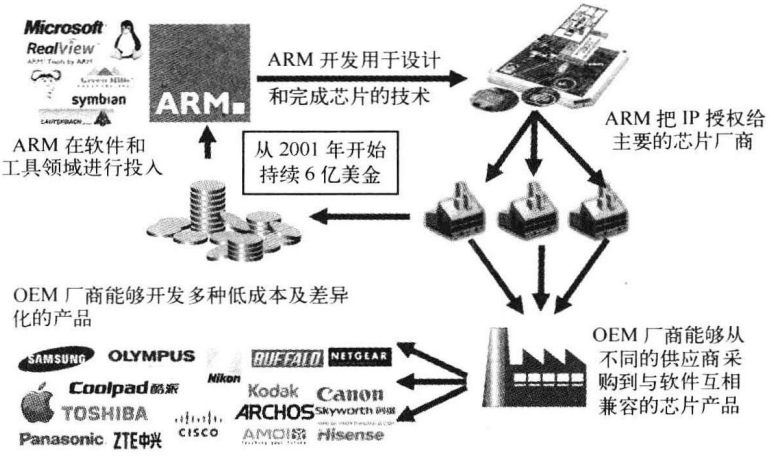


图 1-10 ARM 公司的运作过程

全世界有几十家大的半导体公司都使用 ARM 公司的授权，如图 1-11 所示，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所接受，更具有竞争力。目前，采用 ARM 技术知识产权核的微处理器，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，如图 1-12 所示。基于 ARM 技术的微处理器应用约占 32 位 RISC 微处理器 75% 以上的市场份额，ARM 技术正在逐步渗入到人们生活的各个方面。

图 1-11　ARM 合作伙伴

图 1-12　ARM 产品

2. ARM 体系结构

体系结构定义了指令集（ISA）和基于这一体系结构下处理器的编程模型。基于同样体系结构可以有多种处理器，每个处理器性能不同，所面向的应用也就不同。但每个处理器的实现都要遵循这一体系结构。ARM 体系结构为嵌入式系统发展商提供很高的系统性能，同时保持优异的功耗和面积效率。