

重点大学计算机教材

HZ BOOKS
华章教育

教育部-英特尔精品课程, “教育部高等学校特色专业建设点” 资助项目 (TS2120)

嵌入式系统 原理与设计

吴国伟 徐子川 姚琳 编著
郭禾 主审

大连理工大学



机械工业出版社
China Machine Press

重点大学计算机教材

嵌入式系统 原理与设计

吴国伟 徐子川 姚琳 编著
郭禾 主审



机械工业出版社
China Machine Press

本书深入而全面地论述了嵌入式系统的开发与设计方法。主要内容包括：嵌入式系统设计流程、硬件设计（基础电路、存储设备、接口及 I/O 设备）、软件设计（结构与方法、异常和中断）、定时器和定时器服务、内存管理、软件设计的常见问题等。

本书重点突出，层次分明，注重知识的系统性和先进性；注重理论与实践的联系，以实例应用来阐述相关知识模块，有利于培养工程应用能力。另外，本书还配有完整的电子课件，便于教学。

本书可作为普通高等院校电子信息工程、通信工程、软件工程、电气工程、自动化等专业的高年级本科生与研究生教材，也可作为相关工程技术人员的参考用书。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

嵌入式系统原理与设计/吴国伟，徐子川，姚琳编著. —北京：机械工业出版社，2009.10
（重点大学计算机教材）

ISBN 978-7-111-28228-0

I. 嵌… II. ①吴… ②徐… ③姚… III. 微型计算机-系统设计 IV. TP360.21

中国版本图书馆 CIP 数据核字（2009）第 160585 号

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：盛东亮

北京诚信伟业印刷有限公司印刷

2010 年 9 月第 1 版第 1 次印刷

184mm × 260mm · 19.25 印张

标准书号：ISBN 978-7-111-28228-0

定价：35.00 元

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

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzsj@hzbook.com

前 言

嵌入式技术是在自动化和计算机技术基础上发展而来，是计算机发展的一个分支。嵌入式系统的应用无处不在，尤其体现于信息家电、汽车电子、消费类电子、航空航天等应用领域。

伴随着嵌入式技术的发展，国内外各高校陆续开设了“嵌入式系统原理与设计”的相关课程，介绍微控制器与微处理器的结构、原理及应用。然而，现有嵌入式教材，有的偏重于结构、原理的阐述，比较抽象，学生难于掌握和应用；有的侧重阐述某一个具体的微控制器或微处理器应用方法，缺少嵌入式系统的一般知识与原理，读者难以举一反三。据此，编者精心组织编写了本书，以利于教学和应用。

内容安排上，全书共分12章，分别介绍了嵌入式系统的基本概念、嵌入式系统的整体设计方法、嵌入式系统的组成结构、嵌入式系统的硬件设计和嵌入式系统的软件设计等。在嵌入式系统硬件设计部分，重点阐述了硬件的整体设计、存储设备、接口和I/O设备等。在嵌入式系统软件设计部分，重点阐述了嵌入式操作系统的软件设计、中断和异常的处理、定时服务、软件设计中的普遍问题等。

叙述方法上，各章首先介绍了相应知识模块的基本原理，然后通过实例设计来阐述相应的实践应用，这种叙述方法最大限度地使读者易于掌握嵌入式原理及其应用。此外，本书的案例均是作者多年科研实践的项目总结，具有很强的工程实用性。

本书的读者对象为高年级本科生、研究生和从事嵌入式系统开发设计的工程技术人员。

本书由多位教师合力完成，吴国伟撰写了第1、2、8、9章，徐子川撰写了第3、4、10、11章，姚琳撰写了第5、6、7、12章；郭禾教授对全书进行了审读。

本书编写过程中参考并引用了大量国内外相关书籍，不再一一列举，对作者表示感谢。本书的出版得到教育部特色专业建设的支持，以及Intel教育部精品课程建设和大连理工大学精品课程建设的支持。感谢大连理工大学软件学院领导对本书出版的关怀和支持；同时，本书的出版也得到了机械工业出版社的大力支持，对华章分社编辑们的辛苦工作表示感谢。

嵌入式是一门蓬勃发展的技术，各种新器件、新应用不断涌现；由于作者学疏才浅，书中错误在所难免，敬请读者批评指正。作者的联系方式为E-mail: wgw dut@dlut.edu.cn。

编 者

2010年7月1日

目 录

前言

第 1 章 嵌入式系统简介 1

1.1 嵌入式系统的基本概念 1

1.1.1 现代计算机发展的历史 1

1.1.2 嵌入式系统的定义和特点 2

1.1.3 嵌入式系统的独立发展 2

1.2 嵌入式系统的架构 3

1.3 嵌入式处理器 4

1.3.1 嵌入式处理器分类 5

1.3.2 典型嵌入式处理器介绍 7

1.3.3 嵌入式处理器选型原则 9

1.4 嵌入式操作系统 10

1.4.1 概述 10

1.4.2 典型的嵌入式操作系统 11

1.5 嵌入式系统的开发方法 13

1.6 嵌入式系统的现状和发展趋势 16

思考题 17

第 2 章 嵌入式系统设计 18

2.1 统一建模语言 UML 19

2.1.1 UML 概述 19

2.1.2 UML 的发展历史 19

2.1.3 UML 基本元素 20

2.2 需求 21

2.2.1 需求信息的获取 21

2.2.2 需求定义 22

2.3 规格说明 25

2.4 体系结构设计 28

2.5 设计硬件构件和软件构件 29

思考题 35

第 3 章 嵌入式系统的硬件设计——

电路 36

3.1 嵌入式硬件基础 36

3.1.1 电路原理图阅读与绘制 37

3.1.2 印制电路板的基础知识 37

3.2 电路设计示例：复位电路设计 40

3.2.1 基本复位电路 40

3.2.2 复位电路设计的注意点 43

3.2.3 复位芯片的选择 43

3.2.4 复位电路的调试 44

思考题 45

第 4 章 嵌入式系统的硬件设计——存储

设备 46

4.1 存储设备概述 46

4.1.1 存储器的基本概念 46

4.1.2 存储器的作用 47

4.1.3 存储设备的分类 47

4.1.4 存储器系统层次结构 47

4.1.5 存储器的性能指标 49

4.1.6 存储器的基本构成 49

4.2 常用存储器 52

4.2.1 IC 卡 52

4.2.2 CF 卡 55

4.2.3 MMC 卡 57

4.2.4 SD 卡 60

4.2.5 TF 卡 63

4.2.6 工业电子盘 63

4.2.7 U 盘 63

4.3 存储器件的选择 64

思考题 66

第 5 章 嵌入式系统的硬件设计——接口

及 I/O 设备 67

5.1 接口 67

5.1.1 概述 67

5.1.2 常用接口 67

5.2 定时器和计数器 80

5.3 A/D 和 D/A 转换器 82

5.3.1 概述 82

5.3.2 A/D 和 D/A 转换原理 82

5.4 键盘 87

5.4.1 概述 88

5.4.2 键盘原理 88

5.5 显示器	90	7.3.4 表达式简化	147
5.5.1 CRT 显示器	90	7.3.5 死代码清除	148
5.5.2 液晶显示器	90	7.3.6 过程内嵌	148
5.5.3 发光二极管显示器	92	7.3.7 循环变换	149
5.5.4 等离子显示器	94	7.3.8 寄存器分配	150
5.5.5 显示器的驱动方式	95	7.3.9 指令选择	153
5.5.6 设计实例: 大屏幕 LED 显示屏的 设计	96	7.3.10 理解并使用编译程序	154
5.6 触摸屏	99	7.3.11 解释程序与适时编译程序	154
5.6.1 概述	99	7.4 程序的质量保证和测试	154
5.6.2 触摸屏分类	99	思考题	156
5.7 I/O 性能	102	第 8 章 嵌入式系统软件设计——嵌入式 操作系统	157
思考题	103	8.1 概述	157
第 6 章 嵌入式系统的软件设计——结构 与方法	104	8.1.1 基本概念	157
6.1 嵌入式系统的软件结构	104	8.1.2 内核	158
6.1.1 概述	104	8.1.3 可裁剪性	160
6.1.2 板级支持包 BSP	109	8.2 嵌入式操作系统的服务	161
6.1.3 轮转结构	111	8.2.1 消息队列、信箱和管道	161
6.1.4 带有中断的轮转结构	113	8.2.2 定时器服务	169
6.1.5 实时操作系统结构	118	8.2.3 事件	174
6.1.6 嵌入式软件结构的选择	119	8.2.4 存储管理	176
6.2 软件开发工具	120	8.2.5 中断服务	178
6.2.1 宿主机与目标机	120	8.3 嵌入式操作系统的开发	182
6.2.2 嵌入式系统的定位器	121	8.3.1 多进程多线程设计	182
6.2.3 调试技术	129	8.3.2 进程间通信	184
6.3 软件的设计方法	133	8.3.3 进程的功率优化策略	186
6.3.1 无嵌入式操作系统	133	8.4 调度器	190
6.3.2 有嵌入式操作系统	134	8.4.1 可调度实体	190
思考题	135	8.4.2 多任务	190
第 7 章 嵌入式系统的软件设计——设计 范型与编程	136	8.4.3 上下文切换	190
7.1 嵌入式系统软件的范型	136	8.4.4 调度算法	191
7.1.1 范型	136	8.4.5 $\mu\text{C}/\text{OS-II}$ 任务调度	192
7.1.2 两种范型	137	8.4.6 Linux 任务调度	193
7.2 编程模型	138	8.5 性能问题	196
7.2.1 数据流图	139	8.6 设计实例: 嵌入式 Linux 中的进程 管理	197
7.2.2 控制/数据流图	140	8.7 完全公平调度算法	203
7.3 程序设计语言及基本编译技术	141	8.7.1 概述	203
7.3.1 语句翻译	143	8.7.2 CFS 原理	203
7.3.2 过程	146	8.8 Linux 在嵌入式系统中实时调度算法 改进	203
7.3.3 数据结构	146	8.8.1 概述	203

8.8.2 Linux 实时性改造	205	第 11 章 嵌入式实时系统的内存管理	260
8.8.3 RT-Linux 和 RTAI	206	11.1 概述	260
思考题	207	11.1.1 嵌入式实时系统内存管理的特点	260
第 9 章 嵌入式系统软件设计——异常和中断	208	11.1.2 嵌入式实时系统对内存管理的要求	260
9.1 异常和中断的基本概念	208	11.1.3 内存分配方案	261
9.2 异常的分类和优先级	208	11.1.4 实时系统的内存模式	261
9.2.1 异常的分类	208	11.1.5 内存碎片	262
9.2.2 异常和中断的优先权	209	11.2 常用内存分配算法	264
9.3 可编程中断控制器	210	11.2.1 最先匹配算法	264
9.4 异常和中断的处理	211	11.2.2 伙伴算法	264
9.4.1 异常和中断处理程序的安装	211	11.3 典型嵌入式实时操作系统中内存管理	265
9.4.2 保存处理器状态	211	11.3.1 $\mu\text{C}/\text{OS-II}$ 的内存管理	265
9.4.3 装入和调用异常处理程序	212	11.3.2 VxWorks 的内存管理	266
9.5 ARM 异常中断机制	213	11.3.3 WinCE 的内存管理	267
9.5.1 概述	213	思考题	268
9.5.2 ARM 的异常中断响应过程	214	第 12 章 嵌入式软件设计中的几个问题	269
9.5.3 ARM 异常中断优先级	215	12.1 同步与通信	269
9.5.4 中断控制寄存器	215	12.1.1 资源同步	269
9.6 嵌入式操作系统对中断的支持	224	12.1.2 评估关键段	270
9.6.1 嵌入式 Linux 对异常和中断的处理	224	12.1.3 解决方案	271
9.6.2 $\mu\text{C}/\text{OS-II}$ 异常和中断的处理	228	12.1.4 应用实例: Linux 信号量接口的应用	275
9.6.3 WinCE 异常和中断的处理	230	12.2 死锁及其处理	279
9.7 中断编程实例	231	12.2.1 死锁检测	280
9.7.1 中断编程模式	231	12.2.2 死锁恢复	283
9.7.2 中断编程过程	232	12.2.3 死锁避免和死锁预防	284
思考题	236	12.2.4 应用实例: 银行家算法	285
第 10 章 定时器和定时器服务	237	12.3 优先级倒置	295
10.1 实时时钟和系统时钟	237	12.3.1 资源共享引发的问题	296
10.2 可编程间隔定时器	237	12.3.2 优先级倒置技术	296
10.3 定时器中断服务例程	239	12.3.3 两种优先级倒置方案	296
10.4 软定时器操作和相关问题处理	240	12.4 堆栈溢出	297
10.5 设计实例: S3C2410 的可编程间隔定时器应用	241	思考题	299
10.5.1 实时时钟部件	241	参考文献	300
10.5.2 Timer 部件	249		
10.5.3 看门狗定时器	257		
思考题	259		

第 1 章 嵌入式系统简介

嵌入式系统正在重新塑造人们的生活、工作和娱乐方式。嵌入式系统的应用出现了无数的种类，每类都具有自己独特的个性。例如，当今驾驶的汽车就嵌入了执行附加任务的智能计算机芯片，它使汽车更轻快、更干净、更容易驾驶。电话系统依靠集成在一起的多个硬件和软件将全世界的人们联系起来。即使在私人住宅中，也充满了智能家电和围绕嵌入式系统建造的集成系统，它们都极大地方便和丰富了人们的生活。

现实生活中的嵌入式系统实例无处不在，比如：电视的有线或卫星接收盒、机顶盒、家庭影院系统和电话自动应答机。那么什么是嵌入式系统，嵌入式系统有什么特点以及嵌入式的发展趋势怎么样呢，我们将在本章中详细解释这些问题。

1.1 嵌入式系统的基本概念

要了解嵌入式系统的定义，有必要从现代计算的发展历史了解嵌入式系统的由来，本节从学科建设的角度来探讨。

1.1.1 现代计算机发展的历史

1. 嵌入式系统的产生

电子数字计算机诞生于 1946 年，20 世纪 70 年代，微处理器的出现使计算机进入到现代发展阶段。以微处理器为核心的微型计算机以其小型、价廉、高可靠性特点，迅速占领市场，使计算机应用快速深入到各个领域。基于高速数值计算能力的微型机，表现出的智能化水平引起了控制专业人士的兴趣，要求将微型机嵌入到一个对象体系中，实现对象体系的智能化控制。例如，将微型计算机经电气加固、机械加固，并配置各种外围接口电路，安装到大型舰船中构成自动驾驶仪或轮机状态监测系统。这样，计算机便失去了原来的形态与通用的计算机功能。为了区别于原有的通用计算机系统，把嵌入到对象体系中，实现对象体系智能化控制的计算机，称为嵌入式计算机系统。因此，嵌入式系统诞生于微型机时代，是现代计算机发展的一个重要历史阶段。

2. 现代计算机技术的两大分支

由于嵌入式系统要嵌入到对象体系中，实现的是对象的智能化控制，因此，它有着与通用计算机系统完全不同的技术要求与技术发展方向。通用计算机系统的技术要求是高速、海量的数值计算；技术发展方向是处理速度的无限提升，存储容量的无限扩大。而嵌入式计算机系统的技术要求则是对象的智能化控制能力；技术发展方向是与对象系统密切相关的嵌入性能、控制能力与控制的可靠性。早期，人们勉为其难地将通用计算机系统进行改装，在大型设备中实现嵌入式应用。然而，对于众多的对象系统（如家用电器、仪器仪表、工控单元等），无法嵌入通用计算机系统。况且，嵌入式系统与通用计算机系统的技术发展方向完全不同，因此，通用计算机系统与嵌入式计算机系统必须各自独立地发展，这就形成了现代

计算机技术发展的两大分支。

如果说微型机的出现使计算机进入到现代计算机发展阶段,那么嵌入式计算机系统的诞生,则标志着计算机进入了通用计算机系统与嵌入式计算机系统并存的发展时代,从而导致 20 世纪末的计算机高速发展时期。

3. 两大分支发展的意义

通用计算机系统与嵌入式计算机系统的专业化分工发展,导致 20 世纪末、21 世纪初计算机技术的飞速发展。计算机专业领域集中精力发展通用计算机系统的软、硬件技术,操作系统则迅速扩张计算机基于高速海量的数据文件处理能力,使通用计算机系统进入到尽善尽美阶段。嵌入式计算机系统的发展与各个领域的应用密切结合,传统电子系统领域的厂家与专业人士,利用起源于计算机领域的嵌入式系统,承担起发展与普及嵌入式系统的历史任务,迅速地将传统的电子系统发展到智能化的现代电子系统时代。因此,现代计算机技术发展的两大分支的意义在于:不仅形成了计算机发展的专业化分工,而且将发展计算机技术的任务扩展到传统的电子系统领域,使计算机成为进入人类社会全面智能化时代的有力工具。

1.1.2 嵌入式系统的定义和特点

通过了解嵌入式系统发展的历史,就能历史地、本质地、普遍适用地定义嵌入式系统。

嵌入式系统的一般定义是:将硬件和软件紧密耦合在一起完成复杂功能的计算机系统。术语“嵌入式”反映了这样的系统通常是更大系统中的一个完整的部分,更大系统内部可以共存多个嵌入式系统。

在大多数情况下,嵌入式系统是系统中的系统,它们没有自身独立的功能。例如,数字机顶盒(Digital Set-Top Box, DST)可以在许多家庭娱乐场所中找到。数字音频/视频解码系统(A/V decoder, A/V 解码器)是 DST 中一个完整的部分,是一个嵌入式系统。A/V 解码器接受单个的多媒体流,并且产生声音和视频帧作为输出。DST 从卫星接收的信号中包含多个流或频道,收到的信号转到传输流解码器, A/V 解码器与传输流解码器连接工作。传输流解码器也是一个嵌入式系统,把收到的多媒体流转到分离的频道上,并且只将所选的频道送给 A/V 解码器。

某些情况下,嵌入式系统有独立的功能。例如,网络路由器就是一个能够提供独立功能的嵌入式系统。它由特殊的通信处理器、内存、许多网络访问接口以及实现包的路由算法的特殊软件或者操作系统组成。也就是说,路由器是一个独立的嵌入式系统,路由包从一个端口到另一个端口,实现程序化的路由算法。

嵌入式系统是不断发展的,我们只有了解嵌入式系统的发展历史,掌握嵌入式系统的发展动向,才能准确地掌握嵌入式系统的定义。

1.1.3 嵌入式系统的独立发展

1. 嵌入式系统独立发展的道路

嵌入式系统虽然起源于微型计算机时代,然而,微型计算机的体积、价位、可靠性都无法满足对系统的嵌入式应用要求,因此,嵌入式系统必须走独立发展道路。这条道路就是芯片化道路。将计算机做在一个芯片上,从而开创了嵌入式系统独立发展的单片机时代。在探索单片机的发展道路时,有过两种模式,即“ Σ 模式”与“创新模式”。“ Σ 模式”本质上

是通用计算机直接芯片化的模式，它将通用计算机系统中的基本单元进行裁剪后集成在一个芯片上，构成单片微型计算机；“创新模式”则完全按嵌入式应用要求设计全新的、满足嵌入式应用要求的体系结构、微处理器、指令系统、总线方式、管理模式等。Intel 公司的 MCS-48、MCS-51 就是按照创新模式发展起来的单片形态的嵌入式系统（单片微型计算机）。MCS-51 是在 MCS-48 探索基础上，进行全面完善的嵌入式系统。历史证明，创新模式是嵌入式系统独立发展的正确道路，MCS-51 的体系结构也因此成为单片嵌入式系统的典型结构体系。

2. 单片机技术的发展史

单片机诞生于 20 世纪 70 年代末，经历了 SCM、MCU、SoC 三大阶段。SCM 即单片微型计算机（Single Chip Microcomputer）阶段，主要是寻求最佳的单片形态嵌入式系统的最佳体系结构。创新模式获得成功，奠定了 SCM 与通用计算机完全不同的发展道路。在开创嵌入式系统独立发展的道路上，Intel 公司功不可没。MCU 即微控制器（Micro Controller Unit）阶段，主要的技术发展方向是：不断扩展满足嵌入式应用时对象系统要求的各种外围电路与接口电路，突显其对象的智能化控制能力。它所涉及的领域都与对象系统相关，因此，发展 MCU 的重任不可避免地落在电气、电子技术厂家身上。在发展 MCU 方面，最著名的厂家当数 Philips 公司。Philips 公司以其在嵌入式应用方面的巨大优势，将 MCS-51 从单片微型计算机迅速发展到了微控制器。

单片机是嵌入式系统的独立发展之路，向 MCU 阶段发展的重要因素就是寻求应用系统在芯片上的最大化解决方案；因此，专用单片机的发展自然形成了 SoC（System on Chip，片上系统）化趋势。随着微电子技术、IC 设计、EDA 工具的发展，基于 SoC 的单片机应用系统设计会有较大的发展。因此，对单片机的理解可以从单片微型计算机、微控制器延伸到片上系统。

1.2 嵌入式系统的架构

虽然嵌入式系统应用具有多样性的特点，但嵌入式系统的组成以及设计理念大体上是相同的。一个典型的嵌入式系统包括一个嵌入式微处理器和一个针对物理环境的接口，物理环境可能是一个化学工厂、一个汽车引擎或者一个键盘。比如，在化学工厂的工艺控制嵌入式应用中可能会存在一些标准 I/O 设备（键盘、打印机等），但是在汽车的燃油喷射控制装置中就不存在标准 I/O。在化学工厂的例子中可以选择多种处理器来作为嵌入式处理器，我们将在 1.3 节详细介绍嵌入式处理器的选型。

嵌入式计算机从规模上可以分为三类：微型、小型和大型。微型嵌入式计算机应用在一些小规模系统中，比如机顶盒、遥控器等，在这些应用中，4 位的微控制器往往就可以满足需求。小型嵌入式计算机可以应用 8 位或者 16 位的微控制器或者 PC，经常在图书馆或者工厂的数据采集系统中看到这类应用。大型嵌入式计算机通常是高端的电脑，工厂的管理控制系统往往属于这一类。

嵌入式系统接口可以分为四类：通用串行和并行接口、工业接口、网络接口以及模/数和数/模转换接口。

通用串行接口包括 RS-232、RS-423、RS-422 以及 RS-485 等，并行接口包括打印机接口等，这些将在第 5 章详细介绍。

工业接口用于工业领域，比如 IEEE488（GPIB、HPIB）、SCSI（Small Computer System Interface）、CAMAC（Computer Automatic Measurement And Control）、CAN（Controller Area Network），等等。另外，还有一些新型接口：IrDA（Infrared Data Association）、USB（Universal Serial Bus）以及 IEEE 1394（FireWire or iLink），等等。

嵌入式系统有的时候必须是网络互连的，虽然以上提出的一些技术可用于构建网络（IEEE 1394 等），但是最常用的网络接口是 RJ-45 接口。根据端口的通信速率不同，RJ-45 接口又分为 10Base-T 网 RJ-45 接口和 100Base-TX 网 RJ-45 接口两类。模/数、数/模接口几乎存在于所有的嵌入式设备中，工业控制领域很多设备需要将模拟的连续的信号转换为数字信号，传输给 CPU 处理，最后将数字控制信号转换为相关设备识别的模拟信号，从而完成控制。模数/数模转换的精准程度严重影响着工业控制系统的实时性和精准性。

1.3 嵌入式处理器

市面上有数千种可用的嵌入式微处理器，但是目前还没有一种处于绝对垄断地位的嵌入式微处理器产品。都可以根据嵌入式微处理器的体系结构将它们划分为不同的类别，如表 1-1 所示。

表 1-1 现实中的嵌入式处理器

处理器架构	代表性处理器	制造商
AMD	Au1xxx	Advanced Micro Devices
ARM	ARM7, ARM9	ARM
C16X	C167CS, C165H, C164CI	Infineon
ColdFire	5282, 5272, 5307, 5407	Motorola/Freescale
I960	I960	Vmeto
M32/R	32170, 32180, 32182, 32192	Renesas/Mitsubishi
M Core	MMC2113, MMC2114	Motorola/Freescale
MIPS32	R3K, R4K, 5K, 16	MTI4kx, IDT, MIPS Technologies
NEC	Vr55xx, Vr54xx, Vr41xx	NEC Corporation
PowerPC	82xx, 74xx, 8xx, 7xx, 6xx, 5, 4xx	IBM, Motorola/Freescale
68k	680x0 (68K, 68030, 68040, 68060), 683xx	Motorola/Freescale
SuperH (SH)	SH3 (7702, 7707, 7708, 7709), SH4 (7750)	Hitachi
SHARC	SHARC	Analog Devices, Transtech DSP, Radstone
strongARM	strongARM	Intel
Sparc	UltraSPARPC II	Sun
DSP	TMS320C6xxx	Texas Instrument
TriCore	TriCore1, TriCore2	Infineon

1.3.1 嵌入式处理器分类

根据嵌入式处理器的应用场合及嵌入式处理器发展历史，大致可以将嵌入式处理器分为五类：嵌入式微控制器（MCU）、嵌入式微处理器（MPU）、嵌入式数字信号处理器（DSP）、片上系统（SoC）和可编程片上系统（SOPC）。

1. MCU

MCU（Microcontroller Unit，嵌入式微控制器）一般又称为单片机，即将整个计算机系统集成到一块芯片中。MCU 一般以某种微处理器内核为核心，芯片内部集成 ROM/EPROM、总线、总线逻辑、定时器、I/O 单元、串口、A/D、D/A、Flash 等各种必要功能和外设以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基本都是一样的，不同的是存储器和外设的配置及封装。

与 MPU 相比，MCU 的特点是单片化，体积大大减小，成本和功耗下降，可靠性提高。微控制器的片上资源一般比较丰富，适合于比较简单的控制过程。

2. MPU

MPU（Micro Process Unit，嵌入式微处理器）是由通用计算机中的 CPU 演变而来的，在本质上与通用计算机的 CPU 是一样的，只是在具体的实现细节和功能上有所不同。它装配在专门设计的电路板上，只保留与嵌入式应用紧密相关的功能部件，去除其他冗余功能部件，从而大幅减小系统的体积和功耗。为了满足嵌入式应用的特殊需求，MPU 在工作温度、抗电磁干扰、可靠性方面一般都做了各种增强，在功能上会失去一些在嵌入式领域不常用的功能单元。

和工业控制计算机相比，MPU 具有体积小、质量轻、成本低、可靠性高等优点，但是在电路板上必须包括 ROM、RAM、总线接口、各种外设等器件，从而降低了系统的可靠性，技术保密性也比较差。由于 MPU 及其存储器、总线、外设等安装在一块电路板上，故将其称为单板计算机，如 STD-BUS、PC104 等。

目前 MPU 主要有：Intel x86、IBM PowerPC、Motorola 68K/coldfire、MIPS、ARM、Hitachi SuperH 等系列。

3. DSP

DSP（Digital Signal Processor，数字信号处理器）是一种具有特殊结构的嵌入式微处理器。DSP 芯片的内部采用程序和数据分开的哈佛结构，具有专门的硬件乘法器，广泛采用流水线操作，提供特殊的 DSP 指令，可以用来快速地实现各种数字信号处理算法。根据数字信号处理的要求，DSP 芯片一般具有如下的一些主要特点：

- 在一个指令周期内可完成一次乘法和一次加法。
- 程序和数据空间分开，可以同时访问指令和数据。
- 片内具有快速 RAM，通常可通过独立的数据总线在两块内存区域中同时访问。
- 具有低开销或无开销循环及跳转的硬件支持。
- 快速的中断处理和硬件 I/O 支持。
- 具有在单周期内操作的多个硬件地址产生器。
- 可以并行执行多个操作。

- 支持流水线操作, 使取指、译码和执行等操作可以重叠执行。

与通用微处理器相比, DSP 芯片的其他通用功能相对较弱些。DSP 的芯片可以按照以下的三种方式进行分类。

(1) 按基础特性分类

这是根据 DSP 芯片的工作时钟和指令类型来分类的。如果 DSP 芯片在某时钟频率范围内的任何频率上能正常工作, 那么除计算速度有变化外, 没有性能上的下降, 这类 DSP 芯片一般称为静态 DSP 芯片。

如果有两种或两种以上的 DSP 芯片, 它们的指令集和相应的机器代码及引脚结构相互兼容, 则这类 DSP 芯片称为一致性的 DSP 芯片。

(2) 按数据格式分类

这是根据 DSP 芯片工作的数据格式来分类的。数据以定点格式工作的 DSP 芯片称为定点 DSP 芯片。以浮点格式工作的称为浮点 DSP 芯片。不同的浮点 DSP 芯片所采用的浮点格式不完全一样, 有的 DSP 芯片采用自定义的浮点格式, 有的 DSP 芯片则采用 IEEE 的标准浮点格式。

(3) 按用途分类

按照 DSP 芯片的用途来分, 可分为通用型 DSP 芯片和专用型 DSP 芯片。通用型 DSP 芯片适合普通的 DSP 应用, 如 TI 公司的一系列 DSP 芯片。专用型 DSP 芯片是为特定的 DSP 运算而设计, 更适合特殊的运算, 如数字滤波、卷积和 FFT 等。

4. SoC

SoC (System on Chip, 片上系统) 是随着 VLSI 技术的发展, 在一个硅片上包括 CPU、各种外设控制器 (如网络控制器、LCD 控制器) 等多个功能单元, 构成一个完整的复杂硬件系统。嵌入式处理器内核可以作为 SoC 的一种标准库, 与其他嵌入式系统外设一样, 成为 VLSI 设计中的一种标准器件, 用 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。

SoC 分为通用和专用两类, 市场份额比较小。

5. SOPC

SOPC (System on a Programmable Chip, 可编程片上系统) 是以 PLD (可编程逻辑器件) 取代 ASIC (专用集成电路), 更加灵活、高效的 SoC 技术解决方案。SOPC 代表一种新的系统设计技术, 也是一种初级的软硬件协同设计技术。

SOPC 是 Altera 公司提出来的--种 SoC 解决方案。它将处理器、存储器、I/O 口、LVDS、CDR 等系统设计需要的功能模块集成到一个 PLD 器件上, 构建成一个可编程的片上系统。它是可编程系统, 具有灵活的设计方式, 可裁减、可扩充、可升级, 并具备软硬件在系统可编程 (In-System Programming, ISP) 的功能。可编程器件内还具有小容量高速 RAM 资源。由于市场上有丰富的 IP Core 资源可供灵活选择, 用户可以构成各种不同的系统, 如单处理器、多处理器系统。有些可编程器件内还可以包含部分可编程模拟电路。除了系统使用的资源外, 可编程器件内还具有足够的可编程逻辑资源, 用于实现其他的附加逻辑。

SOPC 的基本特征是至少含有一个嵌入式处理器内核, 具有小容量片内高速 RAM 资源, 丰富的 IP Core 资源可供选择, 足够的片上可编程逻辑资源, 处理器调试接口和 FPGA 编程

接口等。

1.3.2 典型嵌入式处理器介绍

随着嵌入式系统不断深入到人们生活中的各个领域，嵌入式处理器得到了前所未有的飞速发展。市面上的嵌入式处理器数量也在迅速上升。目前比较有影响的嵌入式处理器产品有 IBM 公司的 PowerPC、MIPS 公司的 MIPS、Sun 公司的 Sparc 和 ARM 公司的 ARM 系列，下面将详细进行介绍。

1. PowerPC

PowerPC 是早期 Motorola 和 IBM 联合为 Apple 的 MAC 机开发的 CPU 芯片，商标权同时属于 IBM 和 Motorola，并成为他们的主导产品。IBM 主要的 PowerPC 产品有 PowerPC604s（深蓝内部的 CPU）、PowerPC750、PowerPCG3（1.1GHz）。Motorola 主要有 MC 和 MPC 系列。尽管他们产品不一样，但都采用 PowerPC 的内核。这些产品大都用在嵌入式系统中。

Power PC 架构的特点是可伸缩性好，方便灵活。Power PC 处理器品种很多，既有通用的处理器，又有嵌入式控制器和内核，应用范围非常广泛，从高端的工作站、服务器到桌面计算机系统，从消费类电子到大型通信设备，无所不包。

Motorola 的 MPC860 PowerQUICC（Quad Integrated Communications Controller）内部集成了微处理器和一些控制领域的常用外围组件，特别适用于通信产品。MPC860 PowerQUICC 集成了两个处理块。一个处理块是嵌入的 PowerPC 核，另一个是通信处理模块（Communications Processor Module，CPM），通信处理模块支持四个串行通信控制器（Serial Communication Controller，SCC），实际上它有八个串行通道：四个 SCC，两个串行管理控制器（Serial Management Channels，SMC），一个串行外围接口电路（Serial Peripheral Interface，SPI）和一个 I²C（Inter-Integrated Circuit）接口。由于 CPM 分担了嵌入式 PowerPC 核的外围工作任务，这种双处理器体系结构的处理器功耗要低于传统体系结构的处理器。

2. MIPS

MIPS 的意思是“无内部互锁流水级的微处理器”（Microprocessor without Interlocked Piped Stages），其机制是尽量利用软件办法避免流水线中的数据相关问题。它最早是在 80 年代初期由斯坦福大学 Hennessy 教授领导的研究小组研制出来的。MIPS 公司的 R 系列就是在此基础上开发的 RISC 工业产品的微处理器。这些系列产品为很多计算机公司采用构成各种工作站和计算机系统。

新的 64 位 MIPS 处理器是 RM9000x2，从“x2”这个标记判断，它包含了不是一个而是两个均具有集成二级高速缓存的 64 位处理器。RM9000x2 主要针对网络基础设施市场，具有集成的 DDR 内存控制器和超高速的 HyperTransport I/O 链接。处理器、内存和 I/O 均通过分组交叉连接起来的，可实现高性能、全面高速缓存的统一芯片系统。除通过并行处理提高系统性能外，RM9000x2 还通过将超标量与超流水线技术相结合来提高单个处理器的性能。

64 位处理器 MIPS 64 20Kc 的浮点能力强，可以组成不同的系统，从一个处理器的 Octane 工作站到 64 个处理器的 Origin 2000 服务器；这种 CPU 更适合图形工作站使用。

MIPS 最新的 R12000 芯片已经在 SGI 的服务器中得到应用，目前其主频最大可达 400MHz。

在嵌入式方面，MIPS 系列微处理器是目前仅次于 ARM 的用得最多的处理器之一。应用领域涵盖游戏机、路由器、激光打印机、掌上电脑等各个方面。

3. Sparc

Sun 公司以其性能优秀的工作站闻名，这些工作站的处理器的处理器全都是采用 Sun 公司自己研发的 Sparc 芯片。根据 Sun 公司未来的发展规划，在 64 位 UltraSparc 处理器方面，主要有 3 个系列。首先是可扩展式 s 系列，主要用于高性能、易扩展的多处理器系统。目前 UltraSparc IIIs 的频率已经达到 750MHz。UltraSparc IIIi 的频率已经达到 700MHz。最后是嵌入式 e 系列，它为用户提供理想的性价比，嵌入式应用包括瘦客户机、电缆调制解调器和网络接口等。

4. ARM

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统称，还可以认为是对一种技术的称呼。

1991 年 ARM 公司成立于英国剑桥，主要出售芯片设计技术的授权。目前，采用 ARM 技术知识产权 (Intellectual Property, IP) 核的微处理器，即我们通常所说的 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，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，ARM 技术正在逐步渗入我们生活的各个方面。

ARM 公司是专门从事基于 RISC 技术芯片设计开发的公司，作为知识产权供应商，本身不直接从事芯片生产，靠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。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被消费者所接受，更具有竞争力。

到目前为止，ARM 微处理器及技术的应用几乎已经深入到各个领域：

- 工业控制领域：作为 32 位的 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。
- 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，在该领域的地位日益巩固。
- 网络应用：随着宽带技术的推广，采用 ARM 技术的 ADSL 芯片正逐步获得竞争优势。此外，ARM 在语音及视频处理上行了优化，并获得广泛支持，也对 DSP 的应用提出了挑战。
- 消费类电子产品：ARM 技术在目前流行的数字音频播放器、数字机顶盒和游戏机中得到广泛采用。
- 成像和安全产品：现在流行的数码相机和打印机中绝大部分采用 ARM 技术。32 位 SIM 智能卡也采用了 ARM 技术。

常见的 ARM 处理器有：ARM7 系列、ARM9 系列、ARM9E 系列、ARM10E 系列、

SecurCore系列、Intel 的 Xscale、Intel 的 StrongARM。虽然 ARM 处理器种类丰富，但是它们均采用 RISC 架构，且一般具有以下的特点：

- 体积小、低功耗、低成本、高性能；
- 支持 Thumb（16 位）/ARM（32 位）双指令集，能很好地兼容 8 位/16 位器件；
- 使用寄存器，指令执行速度更快；
- 大多数数据操作都在寄存器中完成；
- 寻址方式灵活简单，执行效率高；
- 指令长度固定。

1.3.3 嵌入式处理器选型原则

为嵌入式系统设计选择处理器，需要考虑几个因素。处理器就像汽车世界中的引擎，引擎对车的性能起到了举足轻重的作用，把为了大马力设计的“悍马”引擎放入“大众”的车体中，必然无法正常工作，嵌入式系统也一样，要想系统能高效、低功耗地工作，就要考虑 CPU 在系统中处的地位。我们通过整体设计的复杂度、设计重用、保护、性能、功耗、成本、工具以及中间件的可用性几个方面来考虑处理器的选型。

(1) 设计的复杂性

设计复杂性是 CPU 选择中非常关键的因素。比如说，设计使用一个状态机来执行来自少量外设的中断，最好使用小的 CPU 或单片机，如 8051 或 Z80。许多系统最初都属于这种类型，如寻呼机，它们存储器引脚少、信号慢，电池消耗很低。

算法及其相互间的交互决定了复杂度，它们也决定了是否需要 RTOS。总的来说，随着应用复杂性的提高，需要高位宽度的处理器。

(2) 设计重用

如今，设计的重用可以大大缩短系统设计的周期，极大程度上减少系统的开发周期，降低成本。随着重用的增加，系统的复杂性也在不断的提高。2000 年的寻呼机，到 2005 年就升级到能播放 MP3 了。现在，8 位 CPU 可能已不足以应付手上的任务。设计包含了多少接口，预示着需要多少处理器能力。在寻呼机例子中，开始的时候只有两个主要接口：用户接口和射频连接。新的设计中，增加了 MP3 播放器，就需要增加存储器接口用来存储和传输数据，增加音频接口用来播放音频数据。现在的系统复杂度比最原始的提高了很多，但如果我们用前瞻性的做法来设计，就能重用以前做过的很多工作。

要确信你的设计还有移植或扩展的空间。今天，你的 8 位设计可能可以做个 MP3 播放器，但是，当设计要被重用，且被移植到机顶盒应用中去时，由于该应用具有很多高带宽的外设，你就可能需要重新建立完整的方案，将它移植到 ARM、MIPS 或是基于 PowePC 的架构上，以应付那些新的约束。

(3) 存储器结构和保护

系统需要自我保护，以防外界的攻击，甚至是来自自身的伤害。这一问题引起了我们对带有存储器单元（MMU）的处理器们的注意。虚拟存储器允许可信的程序访问整个系统，不可信的程序只能访问执行的存储器空间。具有 Internet 连接的 3G 手机就是一个例子。由于错误的程序可能让手机瘫痪，所以必须使用带 MMU 的 CPU。虽然 MMU 不排除系统瘫痪的可能性，但它降低了很难解决的系统失败的可能性。

(4) CPU 性能

CPU 的选择很大程度上影响着整个系统的性能。特别是 CPU 中的高速缓冲寄存器、存储器管理单元、流水线操作系统等。根据系统的需要, 这些特性对于满足系统性能, 都可能是必需的。

(5) 功耗

系统的最后使用, 将决定你的设计功耗。如果设计是电池供电的, CPU 必须尽可能省电。举个例子, 一些 CPU 的工作模式可以是睡眠 (Sleep)、瞌睡 (Doze) 或者休眠 (Snooze)。这些模式允许 CPU 在空闲的时候, 可以挂起操作并通过关闭 CPU 的一些部件来减少功耗。不同的 CPU 在执行相同的任务时, 其功耗是不一样的。

(6) 成本

CPU 的成本由几个方面决定。首先, 是知识产权的成本。然后, 是系统集成的成本, 即用什么工具来设计和实现的系统。最后, CPU 是否可编程芯片, 它适合于在系统中正使用的总线结构吗? CPU 不同的功能/性能, 会导致不同的成本。

(7) 软件问题

RTOS (Real-Time Operating System, 嵌入式实时操作系统) 及中间件的有效性, 可以决定你要做什么样的选择。例如, 设计一个 PDA, 可能想要运行在 Linux 上的中间件。但是, 如果选择支持虚拟存储的操作系统, 意味着将舍去使用不带 MMU 的小型 CPU。

你的设计包含图形界面系统或文件系统吗? 如果是, 就应该选择 RTOS, 而对 RTOS 的选择也将指定所需的 CPU 类型。许多 RTOS 供应商都指定其支持的 CPU 类型, 而把其他的 CPU 排除在外。大多数的 8 位 CPU 都有简单调度器, 适合于使用很少外购代码的小型系统, 而不适合那些需要大量外购代码的系统。外购代码也极大影响着 RTOS 的选择, 决定着可能的 CPU 类型。

系统设计所需的工具对你使用的标准 ANSI C/C++ 编译器有效吗? 如何调试你的系统, 是用硬件/软件协同模拟环境, 还是等到 SoC 系统建立好? 有 JTAG 端口存在吗? 这个端口是用于 CPU 调试, 还是另作其他的串行端口? 选择像 C++ 这样更高级的语言, 或是 UML 设计直接产生代码, 都可能要求更高的总线宽度和时钟频率, 来处理代码的多少和复杂度问题。

总之, 现代的嵌入式系统, 给系统设计师带来了更多的挑战。CPU 的选择不再是件简单的事情。利用一些基准度量, 如总体设计的复杂性、设计重用、保护、性能、功耗、体积、成本、工具及中间件的有效性等, 可以简化决策。

1.4 嵌入式操作系统

1.4.1 概述

许多简单的嵌入式系统 (如单片机系统) 并不需要嵌入式操作系统; 但是, 当系统变得越来越复杂时, 就需要一个嵌入式操作系统来支持, 否则, 应用软件就会变得过于复杂, 使开发难度过大, 安全性和可靠性都难于保证。在多任务嵌入式系统中, 合理的任务调度必不可少, 单纯通过提高处理器速度无法达到目的, 这就要求嵌入式系统的软件必须具有多任务调度能力。现在, 多任务实时操作系统在这一领域的应用越来越多。