

高等院校计算机教育系列教材

0110 01 101 10
010 001 010 011 0
0101100 01 01 010
011 01 01 010 0
010 0 010 010 0
100 0 00 101 100 0 00
1001101010 1001101
010 0101 010 010 0101
0101100 01 010110



ARM 体系 结构与编程

葛超 王嘉伟 陈磊 编著

- 知识点新，突出实践教学，强化能力培养
- 理论知识+感性认识+动手实践，完美结合
- 内容简明扼要，突出知识要点
- 以实用为宗旨，实例丰富，用实例引导读者模仿学习

赠送
电子课件



清华大学出版社

00010
0000 0 10
0001 01
00010 00
00000 010
000 0 00 101
0001101010
010 0101 010
01
00010 10
00010101
010101010
011 01 01
01 010 00
010 0 010

高等院校计算机教育系列教材

ARM 体系结构与编程

葛 超 王嘉伟 陈 磊 编著

清华大学出版社
北 京

内 容 简 介

ARM 处理器是一种 16/32 位的高性能、低成本、低功耗的嵌入式 RISC 微处理器,由 ARM 公司设计,然后授权各半导体厂商生产,它目前已经成为应用最为广泛的嵌入式处理器。

本书分为 14 章,对 ARM 处理器的体系结构、指令系统以及嵌入式系统移植、设备驱动程序等做了比较全面的讲解。其中包括 ARM 系列处理器介绍、ARM 编程模型、嵌入式开发编译工具的使用,并在此基础上介绍了一些典型的基于 ARM 系统的嵌入式应用系统程序设计。通过阅读本书,可以使读者掌握开发基于 ARM 的应用系统的各方面知识。

对于 ARM 指令系统,本书给出了详细的介绍,希望该部分能作为编写 ARM 汇编程序的开发人员的参考资料,提高开发人员的工作效率。本书既可以用作学习 ARM 技术的培训材料,也可以供嵌入式系统开发人员作为参考资料手册。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

ARM 体系结构与编程/葛超,王嘉伟,陈磊编著.--北京:清华大学出版社,2012

(高等院校计算机教育系列教材)

ISBN 978-7-302-30270-4

I. ①A… II. ①葛… ②王… ③陈… III. ①微处理器—计算机体系结构—高等学校—教材
②微处理器—程序设计—高等学校—教材 IV. ①TP332

中国版本图书馆 CIP 数据核字(2012)第 234416 号

责任编辑:汤涌涛 桑任松

封面设计:刘孝琼

责任校对:周剑云

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62791865

印 刷 者:清华大学印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:26 字 数:632 千字

版 次:2012 年 12 月第 1 版 印 次:2012 年 12 月第 1 次印刷

印 数:1~4000

定 价:46.00 元

产品编号:046545-01

前 言

1. 为何编写本书

ARM(Advanced RISC Machines Ltd.)公司自 1990 年 11 月正式成立以来,在 32RISC (Reduced Instruction Set Computer, 精简指令集计算机)CPU 开发领域中不断取得突破。ARM 作为 IP(Intelligence Property, 知识产权)供应商,依靠转让许可,由合作伙伴公司来生成各具特色的芯片。其设计的芯核具有功耗低、成本低等显著优点,获得众多的半导体厂家和整机厂商的大力支持。世界上几乎所有的主要半导体厂商都生产基于 ARM 体系结构的通用芯片,或在其专用芯片中嵌入 ARM 的相关技术。

现在,嵌入式技术无处不在,而 ARM 几乎成为嵌入式技术的代名词。目前 ARM 芯片广泛应用于无线产品、PDA、GPS、网络、消费电子产品、STB 及智能卡。鉴于 ARM 微处理器在国内的逐步推广应用,很多大学已开设了基于 ARM 的 32 位微处理器的相关课程,包括计算机组成原理、计算机体系结构、计算机操作系统,甚至包括电子技术。尽管已经具备这些嵌入式系统的基础知识,但是多数大学毕业生还是不清楚到底如何开发嵌入式系统。编写本书的目的,就是阐述嵌入式体系结构的组成部分及其编程方式,从概念上和实践上说明嵌入式系统的设计开发过程。这本书可以帮助具备计算机基础知识的开发者迅速进入嵌入式系统的开发领域。希望本书能够帮助读者更好地理解嵌入式体系结构,并且参与到嵌入式系统的开发中来。

2. 本书内容特色

(1) 内容新颖、知识全面

全书以嵌入式系统的基础知识开始,逐渐引导读者了解嵌入式系统和嵌入式处理器的基本结构及嵌入式系统的设计方法,详细讲述了嵌入式系统开发的相关知识,重点介绍了基于嵌入式 Linux 的设备驱动程序的开发,以及嵌入式操作系统移植等方面的内容,做到了知识面的全覆盖。

本书内容由浅入深,从基础知识讲起,使初学者对 ARM 体系有总体的了解,然后通过具体的编程实例来进行嵌入式设备驱动开发。

(2) 层次分明,学习轻松

本书结合作者多年讲授嵌入式课程及参与科研项目的经验,借鉴了多位同行教授的提示,从嵌入式系统的基本知识到系统的开发流程,最后是实例编程及嵌入式操作系统的移植等内容,全面介绍了嵌入式系统的结构和编程方面的知识,内容翔实、层次分明,为嵌入式系统的开发者提供了很好的参考。

(3) 通俗易懂,针对性强

本书适合刚刚踏入嵌入式领域的初学者,同时也适合想要进一步掌握嵌入式系统开发高级技巧的读者。本书并没有运用过多的专业术语,即使有也都做了具体的名词解释,并且采用通俗易懂的文字、清晰形象的图片,便于读者理解和阅读,从而帮助读者快速掌握

嵌入式体系结构的基本知识及系统开发的基本方法。

3. 适合的读者群

本书的主要目的是向社会推广与嵌入式系统有关的知识和技术，内容编排是由浅入深的，适合不同程度的读者。入门的读者可以很快地掌握一些常用的技术并积累开发经验；专业读者则可以从对某一嵌入式设备的驱动程序开发中，掌握相应的开发技术和技巧。

本书既可以作为高等院校的教科书使用，又可以作为 ARM 技术人员的培训资料使用，也可以供嵌入式系统开发人员作为参考资料手册。

编者

目 录

第 1 章 嵌入式系统基础 1	2.2.7 ARM 状态下的寄存器结构..... 38
1.1 嵌入式系统简介..... 2	2.2.8 ARM 异常中断..... 42
1.1.1 嵌入式系统的概念..... 2	2.2.9 ARM 处理器的应用选型..... 46
1.1.2 嵌入式系统的特点..... 4	2.3 ARM 指令集..... 47
1.1.3 嵌入式系统的分类..... 4	2.3.1 ARM 指令的一般格式..... 47
1.2 嵌入式系统的研究现状和发展趋势..... 6	2.3.2 ARM 存储器访问指令..... 49
1.2.1 研究现状..... 6	2.3.3 跳转指令..... 54
1.2.2 发展趋势..... 7	2.3.4 数据处理指令..... 55
1.3 典型嵌入式操作系统..... 8	2.3.5 程序状态寄存器(PSR) 传输指令..... 61
1.3.1 嵌入式操作系统的特点..... 8	2.3.6 协处理器指令..... 62
1.3.2 几种典型的嵌入式操作系统..... 8	2.3.7 ARM 杂项指令..... 64
1.3.3 嵌入式系统与 PC 机..... 10	2.3.8 ARM 伪指令..... 67
1.3.4 嵌入式系统与单片机..... 11	2.4 Thumb 指令集..... 69
1.3.5 嵌入式 Linux 系统的特点..... 11	2.5 课后练习..... 70
1.4 课后练习..... 12	第 3 章 嵌入式系统的设计方法 73
第 2 章 嵌入式处理器介绍 13	3.1 嵌入式系统设计概述..... 74
2.1 嵌入式处理器概述..... 14	3.1.1 嵌入式系统的总体结构..... 74
2.1.1 嵌入式处理器简介..... 14	3.1.2 嵌入式系统设计内容..... 76
2.1.2 嵌入式处理器的特点..... 14	3.1.3 嵌入式系统设计的特点..... 76
2.1.3 嵌入式处理器的分类..... 15	3.1.4 嵌入式系统设计方法的分类... 77
2.1.4 嵌入式处理器的体系结构..... 17	3.2 嵌入式系统设计流程..... 79
2.1.5 嵌入式处理器的存储 体系结构..... 24	3.3 课后练习..... 82
2.1.6 典型嵌入式处理器..... 26	第 4 章 ARM 编程模型的工作原理 85
2.2 ARM 系列处理器..... 28	4.1 系统设计概述..... 86
2.2.1 ARM 简介..... 29	4.1.1 嵌入式系统开发基础..... 86
2.2.2 ARM 技术的应用领域..... 29	4.1.2 嵌入式系统选型原则..... 87
2.2.3 ARM 处理器系列..... 29	4.2 ARM920T 简介..... 88
2.2.4 ARM 处理器的工作状态..... 35	4.3 S3C2410X 开发板..... 89
2.2.5 ARM 处理器运行模式..... 36	4.3.1 S3C2410 处理器的特点..... 91
2.2.6 ARM 处理器的存储格式..... 37	4.3.2 ARM 片上总线..... 95

4.3.3 S3C2410 的处理器中断	97	第 6 章 Linux 交叉编译环境	133
4.3.4 S3C2410 处理器片上资源的 定义和使用	103	6.1 交叉编译环境简介	134
4.3.5 编程参考软件包 2410TEST... ..	106	6.1.1 交叉编译环境概念模型	134
4.4 课后练习	109	6.1.2 目标板与宿主机之间的 连接	135
第 5 章 BootLoader	111	6.1.3 文件传输方式	137
5.1 BootLoader 介绍	112	6.1.4 网络文件系统	138
5.1.1 BootLoader 的基本概念	113	6.2 交叉编译工具简介	138
5.1.2 BootLoader 所支持的 CPU 和 嵌入式体系结构	114	6.2.1 gcc 编译器	138
5.1.3 BootLoader 的安装媒介	115	6.2.2 Binutils 工具包	140
5.1.4 BootLoader 的概念扩展	115	6.2.3 GNU make	140
5.1.5 ARM BootLoader 的共性	115	6.2.4 glibc 库	142
5.1.6 BootLoader 移植的必要性	117	6.2.5 gdb 调试工具	142
5.1.7 BootLoader 的烧录和存储	117	6.2.6 交叉编译环境的建立	144
5.1.8 BootLoader 与主机间文件 传输所用的通信设备及协议 ..	117	6.3 主机开发环境配置	148
5.1.9 BootLoader 的通用执行 流程	117	6.3.1 配置主机服务	148
5.2 常用的嵌入式 Linux BootLoader	118	6.3.2 串口控制台工具	150
5.2.1 U-Boot	118	6.3.3 配置 DHCP 服务	152
5.2.2 VIVI	119	6.3.4 配置 TFTP 服务	154
5.2.3 Blob	120	6.3.5 配置 NFS 服务	156
5.2.4 RedBoot	120	6.4 课后练习	158
5.2.5 ARMBoot	120	第 7 章 嵌入式开发及调试	159
5.2.6 DIY	120	7.1 嵌入式系统的开发环境	160
5.3 U-Boot 基础	121	7.1.1 开发环境简介	160
5.3.1 U-Boot 源代码的目录结构 ..	121	7.1.2 ADS 集成开发环境的使用 ..	161
5.3.2 U-Boot 支持的主要功能	123	7.2 嵌入式系统调试方法	174
5.3.3 U-Boot 命令介绍及 环境变量	123	7.2.1 基于主机的调试	176
5.4 U-Boot 的启动过程	124	7.2.2 远程调试器与调试内核	176
5.5 U-Boot 的编译过程	128	7.2.3 在线仿真 ICE	177
5.6 U-Boot 移植的关键技术	131	7.2.4 BDM	178
5.7 课后练习	132	7.2.5 软件仿真器	179
		7.3 ROM 仿真器	179
		7.4 JTAG 接口	180
		7.5 课后练习	181
		第 8 章 简单设备驱动程序	183
		8.1 设备驱动概述	184

8.1.1	基本概念	185	10.1.3	音频文件的格式	252
8.1.2	数据结构	187	10.1.4	WAV 文件格式剖析	253
8.1.3	文件结构	191	10.2	基于 IIS 接口的音频系统	255
8.2	设备驱动基础	192	10.2.1	IIS 接口总线控制原理	255
8.2.1	设备驱动程序结构	194	10.2.2	音频接口设计	257
8.2.2	设备驱动程序接口	195	10.3	音频设备程序的实现	263
8.3	设备驱动模块化编程	196	10.3.1	音频设备编程接口	263
8.3.1	模块化驱动程序概述	197	10.3.2	音频设备文件	266
8.3.2	设备驱动模块化编程的 一般框架	200	10.3.3	音频设备编程设计	268
8.4	PCI 总线	201	10.4	课后练习	274
8.4.1	PCI 设备驱动概述	201	第 11 章	USB 设备驱动程序开发	275
8.4.2	PCI 驱动程序的编程实现	203	11.1	USB 驱动程序简介	276
8.5	课后练习	209	11.1.1	USB 总线概述	276
第 9 章	网络设备驱动程序开发	211	11.1.2	USB 设备的基础构成	277
9.1	网络设备驱动程序简介	212	11.1.3	USB 中的描述符	279
9.1.1	网络设备概述	212	11.1.4	USB 的数据传输	284
9.1.2	重要数据结构 ——struct device	214	11.1.5	USB 文件系统(usbfs)	284
9.2	以太网控制器	218	11.1.6	Libusb 介绍	284
9.3	网络设备的初始化	220	11.2	Linux 下 USB 系统文件节点	286
9.3.1	模块初始化模式分析	221	11.3	USB 主机驱动结构	287
9.3.2	启动初始化模式分析	226	11.4	USB 驱动数据结构	296
9.4	网络设备的打开和关闭	230	11.4.1	USB 驱动的整体构架	296
9.5	数据包的传输和接收	232	11.4.2	USB 驱动中主要的 数据结构	297
9.5.1	Socket 缓冲区及相关操作	233	11.4.3	驱动程序相关函数	298
9.5.2	数据包的传输	235	11.5	USB 主机驱动在 S3C2410X 平台的实现	304
9.5.3	数据包的接收	236	11.5.1	S3C2410X 简介	304
9.6	网络设备驱动程序实例	238	11.5.2	USB 主机控制器	305
9.7	TCP 编程实例	244	11.5.3	USB 驱动程序的移植	305
9.8	课后练习	247	11.6	课后练习	306
第 10 章	音频设备驱动程序开发	249	第 12 章	μC/OS-II 在 ARM 平台的 移植	307
10.1	音频信号基础	250	12.1	实时操作系统简介	308
10.1.1	音频信号	250	12.1.1	实时操作系统的基本概念	308
10.1.2	模拟音频的数字化过程	250			

12.1.2 几种主要的实时操作系统 ...	311	第 14 章 图形用户界面实例	367
12.1.3 实时操作系统的主要性能 ...	312	14.1 图形用户界面	368
12.2 μ C/OS-II 操作系统	317	14.1.1 GUI 简介	368
12.2.1 μ C/OS-II 操作系统概述	317	14.1.2 GUI 特征	368
12.2.2 μ C/OS-II 系统内核	321	14.1.3 GUI 架构	369
12.3 eCos 简介及移植介绍分析	333	14.2 嵌入式 Linux 图形用户界面简介 ...	370
12.3.1 eCos 的体系结构及 可配置性	333	14.2.1 Micro Windows	371
12.3.2 eCos 内核的移植	335	14.2.2 OpenGUI	371
12.4 课后练习	337	14.2.3 Qt/Embedded	372
第 13 章 Linux 在 ARM 平台的移植	339	14.2.4 MiniGUI	372
13.1 Linux 概述	340	14.3 Qt/Embedded 嵌入式图形开发	373
13.1.1 Linux 操作系统的 产生及发展	340	14.3.1 Qt/Embedded 简介	373
13.1.2 Linux 操作系统的 特点和组成	342	14.3.2 Qt/Embedded 信号和 插槽机制	378
13.2 Linux 内核结构	344	14.3.3 Qt/Embedded 实现技术 分析	382
13.2.1 Linux 内核的主要模块	344	14.3.4 Qt/Embedded 图形引擎 实现基础	382
13.2.2 Linux 的文件系统基础	346	14.3.5 Qt/Embedded 事件 驱动基础	383
13.2.3 文件系统的挂载	349	14.3.6 Qt/Embedded 应用程序的 开发流程	384
13.3 Linux 操作系统移植及 根文件系统	350	14.3.7 Qt/Embedded 移植与应用 ...	384
13.3.1 Linux 内核移植	350	14.3.8 Qt/Embedded 窗口部件	387
13.3.2 Linux 根文件系统	361	14.4 课后练习	390
13.4 课后练习	364	习题答案	393

第1章

嵌入式系统基础

嵌入式系统是指用于执行独立功能的专用计算机系统。本章将详细介绍嵌入式系统的基础知识，包括嵌入式系统的特点、分类、应用领域以及发展情况等。

1.1 嵌入式系统简介

嵌入式系统是指用于执行独立功能的专用计算机系统，主要由软件和硬件组成。硬件方面主要包括微处理器、定时器、微控制器、存储器、传感器等一系列微电子芯片与器件，软件方面主要包括嵌入在存储器中的微型操作系统、控制应用软件等，实现诸如实时控制、监视、管理、移动计算、数据处理等各种自动化处理任务。嵌入式系统以应用为中心，以微电子技术、控制技术、计算机技术和通信技术为基础，强调硬件软件协同性与整合性，软件与硬件可裁剪，以满足系统的功能、成本、体积和功耗等要求。

1.1.1 嵌入式系统的概念

“嵌入”是指将一物“置于”另一物中，“嵌入式系统”的全称应为“嵌入式计算机系统”，嵌入式系统可以理解为“一个成为其他产品构成成分的、为特殊目的而个性化设计的计算机软、硬件的组合”。

目前，对嵌入式系统的定义多种多样。下面给出几种比较合理的定义。

- 根据 IEEE(国际电气和电子工程师协会)的定义：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”(the devices used to control, monitor, or assist the operation of equipment, machinery or plants)。
- 从技术的角度定义：嵌入式系统是以应用为中心、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。
- 从系统的角度定义：嵌入式系统是由功能复杂的硬件和软件紧密耦合在一起的计算机系统。术语“嵌入式”反映了这些系统通常是更大系统中的一个完整的部分。被嵌入的系统中可以共存多个嵌入式系统。

一个嵌入式系统由 4 个部分组成：嵌入式处理器、嵌入式外围设备、嵌入式操作系统和嵌入式应用软件。嵌入式系统的组成结构如图 1-1 所示。

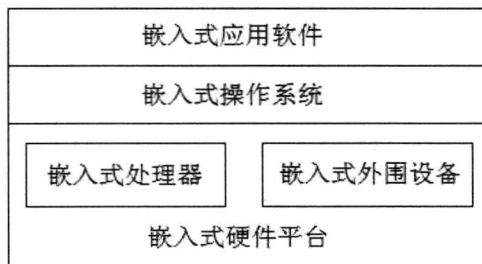


图 1-1 嵌入式系统的组成结构

1. 嵌入式处理器

嵌入式处理器是嵌入式系统的核心部件。嵌入式处理器与通用处理器的最大区别在于嵌入式处理器大多用于特定用户的系统中。它把通用计算机的许多板卡完成的任务集成在

芯片内部，从而有利于嵌入式系统设计趋于小型化，并具有高效率、高可靠性等特征。

大的硬件厂商都会推出自己的嵌入式处理器，因而市面上有 1000 多种嵌入式处理器芯片，其中使用最为广泛的有 ARM、MIPS、PowerPC、MC68000 等。

2. 嵌入式外围设备

嵌入式外围设备是指在一个嵌入式系统中，嵌入式处理器以外的完成存储、通信、调试、显示等辅助功能的其他部件。根据外围设备的功能，可以分为以下几类。

- 存储器：包括静态易失型存储器(RAM/SRAM)、动态存储器(DRAM)和非易失型存储器(Flash)。其中，Flash 以可擦写次数多、存储速度快、容量大及价格低等优点，在嵌入式领域中得到了广泛的应用。
- 接口：应用最为广泛的包括并口、RS-232 串口、IrDA 红外接口、SPI 串行外围设备接口、I²C 接口总线、USB 通用串行总线接口、Ethernet 网口等。
- 人机交互：包括液晶显示器(LCD)、键盘和触摸屏等人机交互设备。

3. 嵌入式操作系统

在大型嵌入式应用系统中，为了使嵌入式开发方便、快捷，需要具备一种稳定、安全的软件模块集合，用来管理存储器分配、中断处理、任务间通信和定时器响应，以及提供多任务处理等，这样的软件模块集合就是嵌入式操作系统。

嵌入式操作系统的引入极大地扩展了嵌入式系统的功能，方便了应用软件的设计，但同时也占用了嵌入式系统的宝贵资源。

早期的嵌入式系统几乎都用于控制，或多或少都有些实时要求，所以以前的嵌入式操作系统实际上是实时操作系统的代名词。近年来，由于手持式计算机和掌上电脑等设备的出现，也有了许多不带实时要求的嵌入式系统。另一方面，由于 CPU 速度的提高，一些原先被认为是实时的反应速度现在已经很普遍了，以前需要在实时操作系统上才能实现的应用，现在已不难在常规的操作系统上实现。在这样的背景下，嵌入式操作系统和实时操作系统就成了不同的概念和名词。

4. 嵌入式应用软件

嵌入式系统的应用软件是针对特定的实际专业领域的，是基于相应的嵌入式硬件平台，并能完成用户预期任务的计算机软件。

用户的任务可能有时间和精度的要求，因此，有些应用软件需要嵌入式操作系统的支持，但在简单的应用场合下不需要专门的操作系统。

嵌入式应用对成本十分敏感，为减少系统的成本，除了精简每个硬件单元的成本外，应尽可能地减少应用软件的资源消耗，尽可能地优化。

应用软件是实现嵌入式系统功能的关键，对嵌入式系统软件和应用软件的要求也与通用计算机软件有所不同。嵌入式软件的特点如下。

(1) 软件要求固化存储。为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器中。

(2) 软件代码要求高质量、高可靠性。半导体技术的发展使处理器速度不断提高，也使存储器容量不断增加；但在大多数应用中，存储空间仍然是宝贵的，还存在实时性的要

求。为此，程序编写和编译工具的质量要高，以减少程序二进制代码的长度，提高执行的速度。

(3) 系统软件的高实时性是基本要求。在多任务嵌入式系统中，对重要性各不相同的任务进行统筹兼顾的合理调度是保证每个任务及时执行的关键，单纯通过提高处理器速度是低效的，甚至是无法完成的。这种任务调度只能由优化编写的系统软件来完成，因此系统软件的高实时性是基本要求。

(4) 多任务实时操作系统成为嵌入式应用软件的必需。随着嵌入式应用的深入和普及，接触到的实际应用环境越来越复杂，嵌入式软件也越来越复杂。支持多任务的实时操作系统成为嵌入式软件必需的系统软件。

1.1.2 嵌入式系统的特点

嵌入式系统与通用型计算机系统相比具有以下特点：

- 嵌入式系统通常是面向特定应用的，嵌入式 CPU 与通用型的 CPU 最大不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中，它通常都具有低功耗、体积小、集成度高等特点，能够把通用 CPU 中许多由板卡完成的任务集成在芯片内部，从而有利于使嵌入式系统设计趋于小型化，移动能力大大增强，与网络的耦合也越来越紧密。
- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一点就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。
- 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，这样才能在具体应用中对处理器的选择更具有竞争力。
- 嵌入式系统和具体应用有机地结合在一起，它的升级换代也是与具体产品同步进行，因此嵌入式系统产品一旦进入市场，就具有较长的生命周期。
- 为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存储在磁盘等载体中。
- 嵌入式系统本身不具备系统自我开发能力，即使设计完成以后用户通常也是不能对其中的程序功能进行修改的，必须有一套开发工具和环境才能进行开发。

1.1.3 嵌入式系统的分类

嵌入式系统有多种分类方法，主要的有按实时性和按软件结构以及复杂程度来分类的方法。

1. 按实时性来分类

嵌入式系统可以分为硬实时(Hard Real-time)系统、软实时(Soft Real-time)系统、严格实时(Firm Real-time)系统和非实时(Non Real-time)系统。

硬实时系统对系统响应时间有严格的要求，如果系统响应时间不能满足，就要引起系统崩溃或致命的错误。软实时系统对系统响应时间有要求，但是如果系统响应时间不能满

足, 不会导致系统出现致命的错误或崩溃。严格实时系统对系统响应时间有严格的要求, 如果系统响应时间不能满足, 就要导致无法接受的低质量服务。非实时系统对系统响应时间没有实时要求。

2. 按软件结构来分类

嵌入式系统可以分为单线程程序(Single-threaded Program)、循环轮询(Polling Loop)系统、有限状态机(Finite State Machine)、前后台系统(Event-Driven System)、实时多任务(Multitasking 或 Multi-thread Program)系统、紧耦合系统(Tightly-coupled System)、松耦合系统(Loosely-coupled System)。

循环轮询系统的优点有: 对于简单的系统而言, 便于编程和理解, 没有中断的机制, 程序运行良好, 不会出现随机的问題; 缺点有: 有限的应用领域(由于不可确定性), 对于大量的 I/O 服务的应用, 不容易实现, 大的程序不便于调试。

有限状态机的优点有: 对于小的系统而言, 便于编程和理解, 可以快速地执行, 只是通过改变输出功能来改变机器的响应; 缺点有: 应用领域有限, 不能保证确定性, 对于大的应用系统, 难于调试。

前后台系统又叫中断驱动系统, 中断服务程序处理异步事件, 这部分可称为前台(Foreground), 或叫中断级。应用程序是一个无限的循环, 循环中调用相应的函数完成相应的操作, 这部分可称为后台(Background), 或叫任务级。

实时多任务系统的优点是: 将复杂的系统分解为相对独立的多个线程, 达到“分而治之”的目的, 从而降低系统的复杂性, 保证系统的实时性, 系统的模块化好, 提高系统的可维护性; 缺点是: 需要采用一些新的软件设计方法, 需要增加功能(如线程间的协调、同步和通信功能), 需要对每一个共享资源互斥, 导致线程间的竞争。

紧耦合系统是多个处理器通过共享内存空间来交换信息。松耦合系统是多个处理器通过通信线路来连接和交换信息。

3. 按复杂程度来分类

(1) 小型嵌入式系统

这些系统是采用一个 8 位或者 16 位的微控制器设计的; 硬件和软件复杂度很小, 需要进行板级设计。它们甚至可以是电池驱动的。当为这些系统开发嵌入式软件时, 主要的编程工具是微控制器或者处理器专用的编辑器、汇编器(Assembler)和交叉汇编器。通常利用 C 语言来开发这些系统。C 程序被编译为汇编程序, 然后将可执行代码存放到系统存储器的适当位置上。为了满足系统连续运行时的功耗限制, 软件必须放置在存储器中。

(2) 中型嵌入式系统

这些系统是采用一个 16 位或者 32 位的微控制器、DSP 或者精简指令集计算机(RISC)设计的; 硬件和软件复杂度都比较大。对于复杂的软件设计, 可以使用如下的编程工具: RTOS、源代码设计工具、模拟器、调试器和集成开发环境(IDE)。软件工具还提供了硬件复杂性的解决方法。汇编器作为编程工具来说用处不大。这些系统还可以运用已有的 ASSP 和 IP(后面将会介绍)来完成各种功能, 例如总线接口、加密、解密、离散余弦变换和逆变换、TCP/IP 协议栈和网络连接功能(ASSP 和 IP 可能还必须用系统软件进行适当的配置, 才能集成到系统总线上)。

(3) 复杂嵌入式系统

复杂嵌入式系统的软件和硬件都非常复杂，需要可升级的处理器或者可配置的处理器和可编程逻辑阵列。它们用于边缘应用，在这些应用中，需要硬件和软件协同设计，并且都集成到最终的系统中；然而，它们却受到硬件单元所提供的处理速度的限制。为了节约时间并提高运行速度，可以在硬件中实现一定的软件功能，例如加密和解密算法、离散余弦变换和逆变换算法、TCP/IP 协议栈和网络驱动程序功能。系统中某些硬件资源的功能也可以用软件来实现。这些系统的开发工具要么十分昂贵，要么根本就不存在。有时候，必须为这些系统开发编译器或者可重定目标的编译器(可重定目标的编译器就是一种可以根据系统中给定的目标配置进行配置的编译器)。

1.2 嵌入式系统的研究现状和发展趋势

1.2.1 研究现状

嵌入式系统以应用为中心，强调体积和功能的可裁减性，是以完成各种信息处理、控制、监视等功能为目标的专用系统。嵌入式系统最早出现在 20 世纪 60 年代武器控制中，后来用于军事指挥控制和通信系统，在国防、国民经济及社会生活各领域普及应用，主要用于工业控制(包括工业设备/智能仪表/汽车电子)、军事国防(军事电子)、网络(包括网络设备/电子商务)、消费电子(包括信息家电/智能玩具/通信设备/移动存储)，在企业、军队、办公室、实验室以及个人家庭等各种场所发挥着越来越重要的作用。

(1) 军用

各种武器控制(火炮控制/导弹控制/智能炸弹制导引爆装置)、坦克、舰艇、轰炸机等陆海空各种军用电子装备，雷达、电子对抗军事通信装备，野战指挥作战用各种专用设备，在海湾战争到最近伊拉克战争中广泛使用。我国嵌入式计算机最早用于导弹控制。

(2) 家用

我国各种信息家电产品，如数字电视机、机顶盒、数码相机、VCD/DVD、音响设备、可视电话、家庭网络设备、洗衣机、电冰箱、智能玩具等广泛采用微处理器/微控制器及嵌入式软件，EMIT(嵌入式微型网络技术)已用于社区对水、煤气远程抄表和洗衣机的遥控。

(3) 工业用

各种智能测量仪表、数控装置、可编程控制器、控制机、分布式控制系统、现场总线仪表及控制系统、工业机器人、机电一体化机械设备、汽车电子设备等广泛采用微处理器/控制器芯片级/标准总线的模板级及系统嵌入式计算机。

(4) 商用

各类收款机、POS 系统、电子秤、条形码阅读机、商用终端、银行点钞机、IC 卡输入设备、取款机、自动柜员机、自动服务终端、防盗系统、各种银行专业外围设备。

(5) 办公用

复印机、打印机、传真机、扫描仪、激光照排系统、安全监控设备、手机、寻呼机、个人数字助理(PDA)、变频空调设备、通信终端、程控交换机、网络设备、录音录像及电

视会议设备、数字音频广播系统等。

(6) 医用

各种医疗电子仪器,包括 X 光机、超声诊断仪、计算机断层成像系统、心脏起搏器、监护仪、辅助诊断系统、专家系统等。

目前微处理器和微控制器产量从几亿片到十多亿片,远远大于通用的个人计算机数量。目前世界嵌入式系统硬件和软件开发工具市场约 2000 亿美元,嵌入式系统带来的工业年产值达一万亿美元。随着全球信息化的发展,嵌入式系统市场将进一步增长。

我国信息化与全面小康社会建设对嵌入式系统市场提出了巨大的需求,信息家电产品年需求量几亿台,每一类数字化家电产品都有千万台市场需求量,工业控制用嵌入式系统有百十万台套需求量,商用嵌入式系统需求量几百万台。我国已有集成电路及板级产品的大批量生产能力,出口的嵌入式应用产品亦将逐步增长,在全球市场也可占有一席之地。

1.2.2 发展趋势

以信息家电为代表的互联网时代的嵌入式产品,不仅为嵌入式市场展现了美好前景,注入了新的生命,同时也对嵌入式系统技术提出了更高的要求。

(1) 嵌入式应用软件开发需要强大的开发工具和操作系统的支持

随着因特网技术的成熟、带宽的提高,ICP 和 ASP 在网上提供的信息内容日趋丰富、应用项目多种多样。为了满足应用功能的升级,设计师们一方面采用更强大的嵌入式处理器,如 32 位、64 位 RISC 芯片或信号处理器 DSP 增强处理能力,同时还采用实时多任务编程技术和交叉开发工具技术来控制功能复杂性,简化应用程序设计、保障软件质量和缩短开发周期。但目前的嵌入式系统的开发工具显然没有大步跟上,所以我们需要开发工具变得更为强大,操作系统能满足需求功能的要求。

(2) 联网成为必然趋势

为适应嵌入式分布处理结构和应用上网需求,面向 21 世纪的嵌入式系统要求配备标准的一种或多种网络通信接口。针对外部联网要求,嵌入设备必须配有通信接口,相应需要 TCP/IP 协议簇软件支持;由于家用电器相互关联(如防盗报警、灯光能源控制、影视设备和信息终端交换信息)及实验现场仪器的协调工作等要求,新一代嵌入式设备还需具备 IEEE 1394、USB、CAN、Bluetooth 或 IrDA 通信接口,同时也需要提供相应的通信组网协议软件和物理层驱动软件。为了支持应用软件的特定编程模式,如 Web 或无线 Web 编程模式,还需要相应的浏览器,如 HTML、WML 等。

(3) 支持小型电子设备,实现小尺寸、低功耗和低成本

为满足这种特性,要求嵌入式产品设计者相应地降低处理器的性能,限制内存容量和复用接口芯片。这就相应地提高了对嵌入式软件设计技术的要求。例如,选用最佳的编程模型和不断改进算法,采用 Java 编程模式,优化编译器性能。因此,既要软件人员有丰富的经验,更需要发展先进的嵌入式软件技术,如 Java、Web 和 WAP 等。

(4) 提供精巧的多媒体人机界面

嵌入式设备之所以易于为亿万用户所接受,重要因素之一是它们与使用者之间的亲和力,自然的人机交互界面,如司机操纵高度自动化的汽车主要还是通过习惯的方向盘、脚踏板和操纵杆。人们与信息终端交互要求以 GUI 屏幕为中心的多媒体界面。手写文字输

入、语音拨号上网、收发电子邮件以及彩色图形/图像已取得初步成效。

目前一些先进的 PDA 在显示屏幕上已实现汉字写入、短消息语音发布，但离掌式语言同声翻译还有很大的距离。

1.3 典型嵌入式操作系统

信息家电商机引发全球嵌入式操作系统平台大战，全球四大操作系统阵营 WinCE、Palm OS、EPOC 和 Linux 展开规格战，各自拥有软件及硬件合作厂商，正在逐鹿信息家电市场的份额。

微软窗口操作系统拥有在个人电脑上的操作系统占有率的绝对优势，使 WinCE 拥有强大的窗口资源支援。不过 Palm OS 操作系统拥有全球 PDA 产品 70% 的市场占有率；同时获得 3COM、IBM 和索尼等跨国公司的支持。EPOC 是发展自欧洲的操作系统、是由世界上最大的 3 家移动电话厂商诺基亚、爱立信和摩托罗拉所共同开发、整合组成新公司，开发出来的新操作系统；在三大电话厂商的合作下，EPOC 市场潜力很大，且占有率高，但应用功能以手机为主，目前并不开放授权。此外，在三大主流操作系统品牌外，Linux 也将是一股强劲的力量；由于 Linux 开放源代码，也成为各厂商极力发展的操作系统，加上其核心小，潜力更为可观。

1.3.1 嵌入式操作系统的特点

嵌入式操作系统是嵌入到非易失性存储器中的软件，是嵌入式应用软件的开发平台。嵌入式操作系统用户的其他应用程序都建立在嵌入式操作系统之上。嵌入式操作系统是嵌入式系统的灵魂，它使得嵌入式系统的开发效率大大提高，系统开发的总工作量大大减少，并且极大地提高了嵌入式软件的可移植性。为了满足嵌入式系统的要求，嵌入式操作系统必须包含操作系统的一些最基本的功能，用户可以通过 API 函数来使用操作系统。

嵌入式操作系统通常包括与硬件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。

嵌入式操作系统通常应用在实时环境下，因此嵌入式系统的实时性要求嵌入式操作系统也应该具有实时性，因此出现了嵌入式实时操作系统(Real Time Operating System, RTOS)。

嵌入式操作系统具有编码体积小、面向应用、实时性强、可移植性好、可靠性高以及专用性强等特点。

1.3.2 几种典型的嵌入式操作系统

下面简单介绍几种嵌入式操作系统。

1. 嵌入式 Linux

嵌入式 Linux 是一个发展最快、应用最为广泛的操作系统。嵌入式 Linux 本身的种种特性使其成为嵌入式开发中的首选。在进入市场的头两年中，嵌入式 Linux 通过广泛应用