

32 位

嵌入式系统硬件 设计与调试

张 崧 编著

 机械工业出版社
CHINA MACHINE PRESS



32 位嵌入式系统硬件设计与调试

张 崧 编著



机械工业出版社

本书全面、系统地讲解了 32 位嵌入式系统中各组成部分的原理和相关设计。书中对包括 ARM、PowerPC 在内的 RISC 处理器, 常用嵌入式操作系统, 以及嵌入式系统中常用的接口电路(如 SDRAM、DDR 内存、各类 Flash、硬盘、液晶、网络接口等)设计做了详细介绍。本书还结合硬件设计, 给出了大量用于硬件调试的代码和驱动程序的编写方法。

本书适合于从事嵌入式系统硬件开发及嵌入式系统驱动软件编写的工程师。

图书在版编目(CIP)数据

32 位嵌入式系统硬件设计与调试 / 张嵩编著. —北京: 机械工业出版社, 2005.5
ISBN 7-111-16417-2

I. 3… II. 张… III. ①微型计算机—系统设计
②微型计算机—调试 IV. TP360.21

中国版本图书馆 CIP 数据核字 (2005) 第 028524 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)
责任编辑: 吉玲 (E-mail: jiling@mail.machineinfo.gov.cn)
责任印制: 石冉
三河市宏达印刷有限公司印刷·新华书店北京发行所发行
2005 年 7 月第 1 版第 1 次印刷
787mm×1092mm 1/16·19.5 印张·482 千字
0001-4000 册
定价: 31.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换
本社购书热线电话 (010) 68326294

[Http://www.machineinfo.gov.cn/book/](http://www.machineinfo.gov.cn/book/)

封面无防伪标均为盗版

前 言

笔者从事嵌入式系统开发多年，却很少看到比较实用的有关嵌入式系统的参考书。当然这也不是全盘否定市面上所有的嵌入式开发的书籍，毕竟笔者仔细读过的也不多。不过笔者感觉，很多嵌入式开发的书的内容比较偏“理论”，而轻“实用”。举个例子，现在市面上有很多 ARM 的嵌入式开发丛书，每一本书都必然会把 ARM 的每一条汇编指令讲一遍。有的 ARM 书籍甚至用大量篇幅向读者讲解如何用汇编指令完成各种功能函数。而笔者从 ARM 的硬件设计到软件调试，到操作系统的驱动程序编写，再到应用程序编写，从来没有用过一条汇编指令。我们都知道，汇编语言移植性几乎没有，可读性差，编写调试困难，现在连 8 位单片机都用 C 语言编写程序了（笔者也开发过多个基于 51 系列单片机的系统，也是用 C 语言编写代码），在 32 位系统下几乎没有使用汇编的机会！书中大量出现这些没有实用价值的内容，只能说是教条化的结果。可能很多读者在读完这些内容后，已经被书中出现的大量术语、寻址模式和状态模式搞得头晕目眩了。

针对这种情况，笔者决定写一本从实际应用出发，全面介绍和理解嵌入式系统的书。笔者也是想起到一个抛砖引玉的作用，希望看到以后市面上出现更多更好的嵌入式开发书籍，为提供国内嵌入式开发水平起一点作用。

当然，笔者学识浅薄，书中难免出错，希望大家不吝赐教。

本书主要适合于哪些读者？

嵌入式系统硬件开发初级工程师

你们可以从本书中系统、全面地了解一个嵌入式系统的组成和设计。

嵌入式系统硬件开发中级工程师

本书针对嵌入式系统常用的接口，从规范到信号定义到波形图，都提供了详细的介绍，并给出了大量的参考设计和调试程序。本书中有关操作系统和驱动程序的内容也可以让你们对嵌入式软件有更深入的理解。

嵌入式系统驱动软件编写工程师

本书给出了嵌入式系统常用接口在嵌入式 Linux 操作系统下的驱动程序示例，而且本书提供的很多硬件知识也可以加深你们对嵌入式系统硬件的理解。

本书包含哪些内容？

本书从实用角度出发，详细介绍了嵌入式系统常用的器件，常用的接口规范的硬件设计参考原理图、硬件调试程序和方法以及驱动程序。本书硬件设计多以三星的 ARM 芯片 S3C2410 为例，操作系统以 Linux 操作系统为例。

编 者

目 录

前言

第 1 章 概述	1
1.1 嵌入式系统概述	1
1.1.1 嵌入式系统的历史	1
1.1.2 嵌入式系统的定义	2
1.1.3 嵌入式系统的特点	3
1.1.4 嵌入式系统的市场	4
1.2 32 位嵌入式系统介绍	4
1.3 本书内容介绍	5
第 2 章 32 位嵌入式处理器	6
2.1 RISC 的介绍	6
2.1.1 RISC 的历史	6
2.1.2 RISC 的特点	7
2.1.3 RISC 和 CISC 的比较	7
2.1.4 RISC 在嵌入式系统中的优势	8
2.2 PowerPC 的介绍	8
2.2.1 PowerPC 的历史	8
2.2.2 PowerPC 的特点	9
2.2.3 PowerPC 产品简介	10
2.3 68K/ColdFire	12
2.3.1 68K/ColdFire 的历史	12
2.3.2 ColdFire 系列处理器的特点	13
2.4 MIPS 处理器	15
2.5 ARM 处理器介绍	17
2.5.1 ARM 的历史	17
2.5.2 ARM 内核的特点	18
2.5.3 ARM 内核系列的介绍	19
第 3 章 嵌入式操作系统	32
3.1 操作系统的作用和分类	32
3.1.1 操作系统的功能	32
3.1.2 通用操作系统 (General Purpose Operating System)	33
3.1.3 实时操作系统 (Real Time Operating System)	36
3.1.4 实时操作系统与通用操作系统的比较	37
3.2 常见嵌入式操作系统的介绍	40

3.2.1	标准 Linux 系统.....	40
3.2.2	μ CLinux 介绍.....	44
3.2.3	RTLinux 介绍.....	47
3.2.4	微软公司的 Windows CE.....	48
3.2.5	VxWorks.....	51
3.3	嵌入式操作系统特点总结.....	57
第 4 章	动态内存 (DRAM)	59
4.1	早期动态内存简介.....	59
4.2	SDRAM (Synchronous DRAM) 原理介绍.....	60
4.2.1	SDRAM 简介.....	60
4.2.2	SDRAM 的启动和初始化.....	62
4.2.3	SDRAM 的模式寄存器 (MR).....	63
4.2.4	SDRAM 命令解析.....	64
4.2.5	SDRAM 读过程.....	66
4.2.6	SDRAM 写过程.....	68
4.2.7	SDRAM 读写效率.....	69
4.3	SDRAM 的电路设计.....	70
4.3.1	SDRAM 的硬件设计.....	70
4.3.2	SDRAM 的 PCB 设计.....	71
4.4	SDRAM 的硬件调试和软件测试.....	73
4.5	DDR (Double Data Rate) SDRAM.....	76
4.5.1	DDR 内存结构简介.....	77
4.5.2	DDR 内存的技术要点.....	79
4.5.3	DDR 内存读过程.....	82
4.5.4	DDR 内存写过程.....	84
4.5.5	DDR II 内存介绍.....	85
4.5.6	DDR 内存发展趋势.....	87
4.6	RDRAM (Rambus DRAM).....	88
4.6.1	RDRAM 介绍.....	89
4.6.2	RDRAM 技术和 DDR 内存技术特点比较.....	93
第 5 章	串口及控制台调试技术	96
5.1	串口技术介绍.....	96
5.1.1	串口接口信号.....	96
5.1.2	串口原理.....	97
5.1.3	串口电平.....	97
5.1.4	流量控制.....	98
5.1.5	串口参数设置.....	98
5.2	串口硬件设计.....	99
5.2.1	电平转换电路.....	99

5.2.2	串口芯片	100
5.3	串口的设置和控制台技术介绍	102
5.3.1	串口设置	102
5.3.2	串口功能函数	103
5.3.3	串口控制台调试技术	106
第 6 章	Flash	110
6.1	Flash 技术介绍	110
6.1.1	Flash 介绍	110
6.1.2	Flash 技术原理	111
6.1.3	NOR Flash 市场介绍	113
6.2	NOR Flash 的硬件设计和调试	114
6.2.1	处理器的总线操作	114
6.2.2	NOR Flash 的电路设计	116
6.2.3	NOR Flash 的识别	117
6.2.4	NOR Flash 的擦除	119
6.2.5	NOR Flash 的编程	121
6.3	NOR Flash 的软件编写	121
6.3.1	NOR Flash 的刻录程序介绍	121
6.3.2	NOR Flash 在 Linux 系统下的使用	126
第 7 章	NAND Flash	129
7.1	NAND Flash 的介绍	129
7.1.1	各种 Flash 技术简介	129
7.1.2	NAND Flash 和 NOR Flash 的比较	131
7.1.3	常见的存储卡标准	133
7.2	NAND Flash 的接口规范	138
7.2.1	NAND Flash 的接口信号	138
7.2.2	NAND Flash 的地址结构	139
7.2.3	NAND Flash 的命令	140
7.3	NAND Flash 电路设计	144
7.3.1	可热插拔的 NAND Flash 和 2410 处理器的接口设计	144
7.3.2	NAND Flash 和普通处理器的接口设计	146
7.3.3	NAND Flash 软件调试	146
7.4	NAND Flash 驱动程序的编写	150
7.4.1	NAND Flash 文件系统简介	151
7.4.2	SSFDC 软件规范	152
7.4.3	Linux 操作系统下 NAND Flash 驱动程序的编写	153
第 8 章	硬盘及 FAT 文件系统	164
8.1	硬盘接口设计	164
8.1.1	硬盘接口信号	164

8.1.2	硬盘的 PIO 工作方式及其硬件设计	165
8.2	硬盘软件设计	167
8.2.1	硬盘寄存器介绍	168
8.2.2	硬盘命令介绍	169
8.2.3	硬盘调试程序	170
8.2.4	Linux 操作系统下硬盘驱动程序编写	174
8.3	硬盘逻辑结构介绍	176
8.4	FAT 文件系统介绍	178
8.4.1	FAT16 分区构造	178
8.4.2	FAT16 文件系统解析	179
8.4.3	FAT32 文件系统介绍	181
8.4.4	FAT32 文件系统 DBR 和 BPB 介绍	182
8.4.5	FAT32 文件系统下读取文件的过程	185
8.4.6	FAT32 文件系统中目录和文件存储	186
8.4.7	在 FAT32 文件系统目录下查找文件的代码	190
第 9 章	液晶 (LCD)	191
9.1	LCD 介绍	191
9.1.1	LCD 原理	191
9.1.2	LCD 种类	192
9.1.3	LCD 参数说明	195
9.2	字符型 LCD 的设计	197
9.2.1	字符型 LCD 的硬件设计	197
9.2.2	字符型 LCD 的测试	199
9.2.3	Linux 操作系统下字符型 LCD 的驱动程序编写	201
9.3	STN 型 LCD 的设计	205
9.3.1	STN 型 LCD 的硬件设计	205
9.3.2	STN 型 LCD 的软件调试	207
9.4	TFT 型 LCD 的设计	211
9.4.1	TFT 型 LCD 的硬件设计	211
9.4.2	TFT 型 LCD 的软件调试	213
9.5	VGA 接口设计	214
9.5.1	VGA 接口介绍	214
9.5.2	VGA 接口硬件设计	216
9.6	LCD 的电源设计	217
9.6.1	单路高压直流电的电路设计	217
9.6.2	高压交流电的设计	217
9.6.3	多路高压电源的设计	218
9.6.4	LED 背光电流源的设计	220
9.7	Linux 操作系统下 LCD 驱动程序 FrameBuffer 和图形引擎的介绍	220

9.7.1	Linux 操作系统下图形驱动程序介绍	221
9.7.2	Linux 操作系统下 FrameBuffer 驱动程序测试程序	222
9.7.3	Linux 操作系统下图形界面系统介绍	224
第 10 章	PCMCIA 接口	229
10.1	PCMCIA 规范	229
10.1.1	PCMCIA 物理规范	229
10.1.2	PCMCIA 内存卡规范	231
10.1.3	PCMCIA I/O 卡规范	236
10.1.4	CF 卡规范	239
10.2	PCMCIA 接口的硬件设计和调试	240
10.2.1	专用芯片实现 PCMCIA 接口	240
10.2.2	PD6710 芯片实现 PCMCIA 接口的调试	242
10.2.3	组合逻辑实现 PCMCIA 接口	247
10.2.4	组合逻辑实现 PCMCIA 接口的调试	249
10.3	PCMCIA 接口的软件结构和驱动程序	250
10.3.1	PCMCIA 接口的软件结构	250
10.3.2	PCMCIA 接口的 CIS 信息和配置寄存器	251
10.3.3	PD6710 芯片实现 PCMCIA 接口 Linux 操作系统下的驱动程序	253
10.3.4	组合逻辑实现 PCMCIA 接口 Linux 操作系统下的驱动程序	255
第 11 章	网络	259
11.1	网络概述	259
11.1.1	网络的历史	259
11.1.2	网络的分层模型	259
11.1.3	中国网络现状	261
11.2	以太网接口设计	262
11.2.1	以太网介绍	262
11.2.2	以太网接口芯片 RTL8019AS 的硬件设计	263
11.2.3	RTL8019AS 芯片的软件调试	264
11.2.4	Linux 操作系统下 RTL8019AS 芯片的驱动程序编写	265
11.3	Modem 接口设计	266
11.3.1	Modem 的硬件设计	267
11.3.2	Modem 的软件设计	267
11.4	GPRS 接口设计	270
11.4.1	GPRS 简介	270
11.4.2	GPRS 接口硬件设计	270
11.4.3	GPRS 的软件设计	271
11.5	WLAN (无线局域网) 接口设计	275
11.5.1	WLAN 简介	275
11.5.2	WLAN 接口设计和调试	277

11.5.3 Linux 操作系统下 WLAN 软件设计介绍	278
第 12 章 BootLoader 编程	279
12.1 BootLoader 介绍	279
12.1.1 BootLoader 简介	279
12.1.2 BootLoader 的功能	279
12.1.3 BootLoader 的结构	280
12.2 ARM 处理器的初始化代码分析	280
12.2.1 ARM 处理器的中断处理和运行模式	280
12.2.2 ARM7 处理器的初始化代码示例及分析	282
12.2.3 ARM9 处理器的初始化代码示例及分析	284
12.3 硬件初始化和操作系统加载代码分析	290
12.3.1 Linux 操作系统启动模式介绍	290
12.3.2 从 NOR Flash 启动 Linux 操作系统的 BootLoader 代码分析	291
12.3.3 从 NAND Flash 启动 Linux 操作系统的 BootLoader 代码分析	293
12.3.4 从 FAT32 硬盘分区上启动 Linux 操作系统的 BootLoader 代码分析	295
附录 名词解释	298

第1章 概述

随着半导体技术的发展,越来越多的设备开始具备“智能”,而嵌入式系统就是各种设备里“智能”的实现手段。今天,嵌入式系统已经融入了我们工作和生活的各个领域,发挥着重大的作用。我们身边大到飞机、汽车,小到遥控器、手机,里面都有嵌入式系统在默默地工作。你可能难以想象,一辆汽车上所包含的大小嵌入系统总和可能超过 50 个!

嵌入式系统除了传统的工业控制、电信设施继续发挥它巨大的作用,它还在家用电器、医疗保健设备、数码相机、多媒体播放设备、手机、PDA 等各个领域扮演着越来越重要的角色。嵌入式系统已有 30 多年的发展历史,但是从来没有像现在这样风靡。

1.1 嵌入式系统概述

1.1.1 嵌入式系统的历史

世界上第一个被使用的嵌入式系统可以追溯到 20 世纪 60 年代中期的美国阿波罗登月计划。当时,登月飞船上包含两个阿波罗导航计算机 AGC (Apollo Guidance Computer) 系统(见图 1-1)。每个 AGC 的逻辑规模大概 4 000 门左右,拥有 2.048MHz 的时钟源、4KB 的 RAM、32KB 的 ROM,指令周期为 12ms。AGC 的任务是完成阿波罗飞船的导航控制,它是整个系统中非常重要的部分。由于这是首次采用嵌入式系统,并利用计算机系统来控制飞行。因此,在当时 NASA 将 AGC 系统评定为整个飞船中最不稳定的部分。

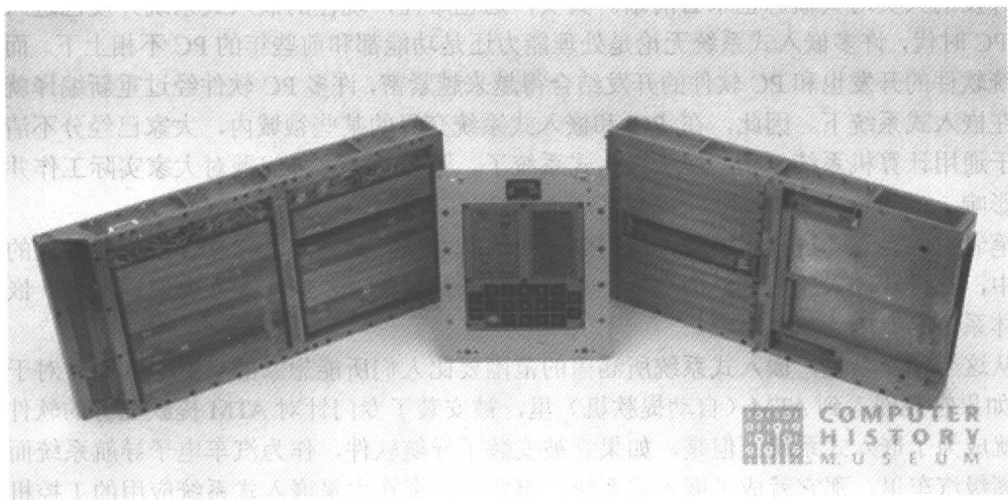


图 1-1 世界上第一个嵌入式系统

那么世界上首次大量使用的嵌入式系统又是什么呢?著名的美国战略核武器民兵

(MinuteMan) 系列导弹的导航系统被认为是首次大量使用嵌入式系统的产品。图 1-2 给出了民兵 III 导弹内嵌入式导航系统的照片。不过也正是因为民兵导弹大胆采用了嵌入式导航系统, 才使得它的弹道精度大大提高, 成为美国至今仍在使用的主要战略核武器之一。

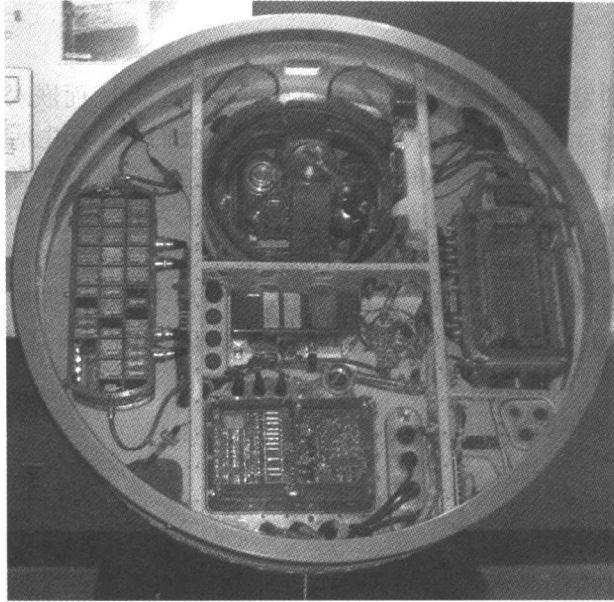


图 1-2 世界上第一个被大量使用的嵌入式系统

1.1.2 嵌入式系统的定义

要准确、清晰地定义嵌入式系统确实是一件比较困难的事情。事实上, 很多从事嵌入式系统开发的人员对其概念也未必清晰。其实, 这也难怪, 现在的嵌入式系统开发已进入所谓的后 PC 时代, 许多嵌入式系统无论是处理能力还是功能都和前几年的 PC 不相上下。而嵌入式系统软件的开发也和 PC 软件的开发结合得越来越紧密, 许多 PC 软件经过重新编译就可以运行在嵌入式系统下。因此, 在 PC 和嵌入式系统交界的某些领域内, 大家已经分不清楚哪些属于通用计算机系统, 哪些属于嵌入式系统了。不过幸好这种问题对大家实际工作并没有太大影响。

笔者也曾经查阅过很多资料, 想得到一个清楚的嵌入式系统的定义。从笔者看过的众多定义中, 觉得从嵌入式系统的历史性、本质性和普遍性来看, 嵌入式系统应定义为: 嵌入到对象体系中的专用计算机系统。

从这个意义上说, 嵌入式系统所涵盖的范围要比人们所能想象的更宽。例如, 对于一台 PC, 如果它被嵌入到 ATM (自动提款机) 里, 被安装了专门针对 ATM 提款业务的软件, 那么它就成为了嵌入式系统; 但是, 如果它被安装了导航软件, 作为汽车电子导航系统而被嵌入到高级汽车里, 那它就成了嵌入式系统。事实上, 存在大量嵌入式系统应用的工控机也是基于 PC 技术实现的。

在嵌入式系统中, 处理器是系统的核心。通常, 人们将嵌入式系统的处理器分为三种: 数字信号处理器 DSP (Digital Signal Processor)、微型控制器 MCU (Micro Control Unit) 和微

型处理器 MPU (Micro Processor Unit)。

DSP 一般用于射频、音频和视频的处理,某些对实时性、计算强度要求较高的场合也使用 DSP。总的说来, DSP 的特点是处理速度快,内部专门针对数字信号处理进行过优化,但 DSP 的外设接口和集成的功能模块一般不多;传统上的 MCU 是指 4 位、8 位和 16 位的嵌入式处理器。这类处理器计算能力往往比较弱,更多的是实现自动控制功能;而 MPU 则是指 32 位的嵌入式处理器,这类处理器计算能力强,往往集成了多种功能模块,拥有多种 I/O 接口,因此逐渐成为市场主流。不过这种划分也未必准确,有很多厂家将自己的 32 位处理器也归入到 MCU 一类。另外,市场上还出现了很多所谓的“双核”处理器,这些处理器往往同时包含一个 DSP 内核和一个 32 位处理器内核。而且,这些“双核”甚至“多核”处理器代表着未来嵌入式处理器的发展方向。比如 Intel 公司的用于网络处理的 IXP 系列处理器内就包含有多达 17 个处理器内核的芯片。而 ST 公司推出的手机芯片除了包含一个主处理器内核外,还包含一个硬件视频/音频编解码处理器模块。在 2004 年 2 月美国的 2004 电子产业高峰会议 (Electronic Summit 2004) 上,从事 64 位 RISC IP 研究的 Tensilica 公司总裁 Chris Rowen 博士说,未来十年,一块芯片中需要上千个处理器芯核,每秒可以完成 10^{12} 次运算。

1.1.3 嵌入式系统的特点

其实,嵌入式系统的定义中就包含了它最本质的三个特点:嵌入性、专用性和计算机体系。嵌入式系统的特点可总结为以下 5 点。

(1) 嵌入式系统是面向特定应用的:嵌入式系统的这个特点,决定了嵌入式处理器的千差万别。在 PC 领域,处理器基本上就只有 Intel 和 AMD 这两种品牌,而且它们大部分的代码和软件还是相互兼容的。Intel 公司的 PC 处理器种类虽然不少,但除了在速度上和性能上有差异之外,功能上并没有什么不同。而在嵌入式处理器领域,有集成了众多通信接口的通信处理器;有集成了音频、视频接口用于摄影摄像的专用处理器;还有低功耗的手机处理器。几乎每一个特定的市场,每一种特定的功能,都会有一种特别的嵌入式处理器与之对应。

(2) 嵌入式系统的软件相对固定:还是因为嵌入式系统的专用性,所有嵌入式系统中所使用的软件往往都是针对特定系统特定功能而专门开发的。用户一般不需要更改嵌入式软件,也不需要自己开发或是安装新的软件,这和 PC 体系是完全不同的。

(3) 嵌入式系统的实时性:首先,我们必须清楚并不是所有的嵌入式系统都具有实时性,而且也不是所有嵌入式系统都应该具备实时性。但是,的确有许多嵌入式系统,尤其是工控领域的嵌入式系统对实时性有着非常高的要求。例如前面所提到的用于核武器的导航系统,我想大家都清楚,如果导航系统对外界响应速度不够及时将会带来什么严重的后果。

(4) 嵌入式系统的特殊要求:嵌入式系统因为被应用到不同的领域,因此不同的领域对嵌入式系统也有不同的要求。比如手持类设备的嵌入式系统就要求系统中所有芯片的功耗都必须足够低;用于汽车的嵌入式系统就要求系统中所有的芯片在低温 (-40°C , 甚至更低) 环境下也能正常工作;用于工业控制的嵌入式系统可能会要求在电磁噪声非常恶劣的环境下系统也能正常工作;用于电信交换设备的嵌入式系统会要求 1 年内系统死机的次数小于 10 次;而用在导弹和卫星上的系统则可能会有更多更苛刻的要求。

(5) 嵌入式系统的“嵌入性”:很多嵌入式系统的用户界面不够友好,甚至没有。这是因为某些嵌入式系统不需要和用户进行数据交互,就能够完成相关的任务。这也和如今强调界

面美观友好的 PC 软件系统不同。

1.1.4 嵌入式系统的市场

今天，嵌入式系统所带来的工业年产值已超过了 1 万亿美元，来自 1997 年美国嵌入式系统大会（Embedded System Conference）的报告指出，未来 5 年仅基于嵌入式计算机系统的全数字电视产品，就将在美国产生一个每年 1 500 亿美元的新市场。美国汽车大王福特公司的高级经理也曾宣称，“福特出售的‘计算能力’已超过了 IBM”，由此可以想见嵌入式计算机工业的规模和广度。

嵌入式处理器的种类从 4 位、8 位、16 位到 32 位、64 位都有；内存寻址空间从几 KB 到上 GB；处理速度从几 MHz 到上 GHz；芯片管脚数目从 8 位到上千位……可以说种类非常繁多。据不完全统计，目前全世界嵌入式处理器的品种总量已经超过 1000 种，流行的体系结构就有 30 多个系列。其中 8051 体系占了多半，生产这种单片机的半导体厂家有 20 多个，共 350 多种衍生产品，仅 Philips 公司就有近 100 种。现在几乎每个半导体制造商都在生产嵌入式处理器，而且越来越多的公司拥有自己的处理器设计部门。

1.2 32 位嵌入式系统介绍

就嵌入式系统的发展情况而言，我们显然已经进入了后 PC 时代。在这个时代，所有的嵌入式系统都面临着两个趋势，即网络化和智能化。著名的工控系统提供商研华公司已经宣布今后将不再生产不带网络接口的工控系统。在可以预见的将来，工厂里每台设备可能都将具有嵌入式处理器系统，它们通过网络相互连接，通过远端的控制系统进行精密地控制。无论是网络化还是智能化，都将使嵌入式系统更加复杂化，并且都将对嵌入式处理器提出更高的要求。

而另一方面，随着半导体工艺的进步，在摩尔定律的作用下，32 位嵌入式处理器的价格已经下降到一个非常低的价格水平。以三星公司的 ARM 芯片 4510 为例，这款处理器速度高达 50MIPS（Mega Instructions Per Second），且带 100M 以太网接口，其售价仅在 3 美元左右。而与此同时，某些速度在几个 MIPS 的 8 位处理器的价格也在 4 美元以上。

在上述两种趋势的共同作用下，32 位嵌入式处理器成为系统主流正在逐渐变为事实。根据世界半导体贸易统计 WSTS 2003 年 12 月进行的“嵌入式设计调查”，52% 的回复者都希望在未来两年内考虑将其系统转向 32 位架构。WSTS 的市场报告还显示，8 位 MCU 市场规模为 36 亿美元，出货量增长，但价值与去年持平。而 32 位市场规模约 25 亿美元，年增长率为 25%。

InStat/MDR 也曾预测在 2001~2006 年期间，32 位微控制器 MCU 的复合年增长率可达 22.6%。而全球 32 位 MCU 市场在 2003 年的成长幅度实际已超过 30%，而在 2004 年预计将达到 38%。据中国有关机构预测分析，今年中国大陆市场 8 位 MCU 的需求量会增长 15%~20% 左右；而 32 位 MCU 的需求量将继续以 100% 的速度增长。

如图 1-3 所示，从 Gartner 公司对几千名亚洲嵌入式开发工程师的调查可以发现，越来越多的 8 位和 16 位嵌入式系统将在最近几年中转向 32 位嵌入式系统。

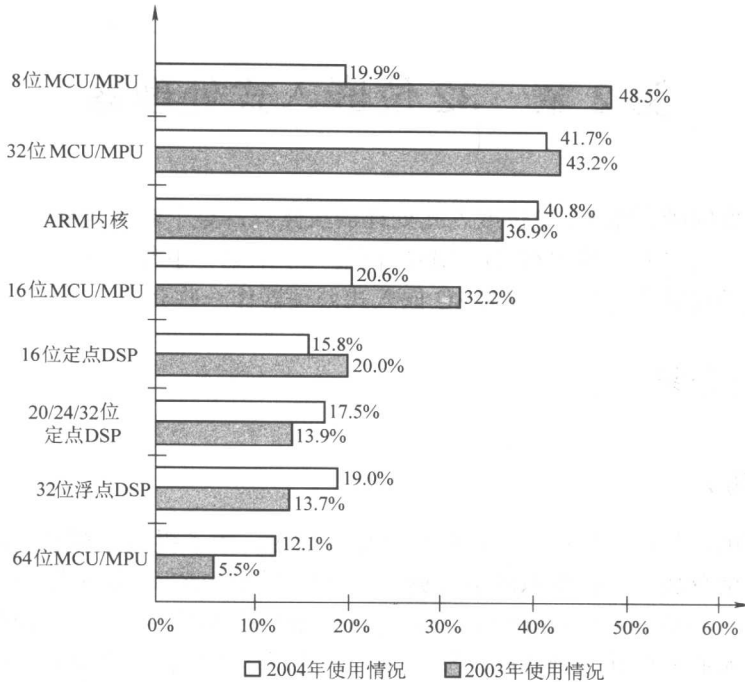


图 1-3 亚洲（不包括日本）使用嵌入式处理器的情况（来源于 Gartner 公司数据）

1.3 本书内容介绍

本书共分 12 章，我们将在第 2 章介绍各种嵌入式处理器的特点，RSIC 和 CSIC 处理器的特点，以及当前常用处理器的简要情况；在第 3 章中，我们会对嵌入式操作系统作一个简单介绍，内容包括实时操作系统的定义和特点以及常用嵌入式操作系统介绍，包括 VxWorks、Linux、 μ CLinux、Windows CE 等；在第 4 章中，我们会重点介绍 SDRAM 的原理、接口信号和电路设计，同时也会对 DDR、RDRAM 技术作一个介绍；在第 5 章中，我们会详细地讲解串口原理、硬件设计，以及如何将串口作为嵌入式系统的调试控制台来实现与嵌入式系统的人机交互；在第 6 章中，我们将会对 Flash 的技术原理和特点作一个介绍，同时会给出 NOR Flash 的硬件设计、软件调试，以及可以向 NOR Flash 编程的软件；在第 7 章中，我们会对大容量存储设备 NAND Flash 作详细介绍，包括基于 NAND Flash 技术的各类存储卡，以及 NAND Flash 的存储结构、硬件设计、驱动设计等；在第 8 章中，我们将对嵌入式系统如何使用硬盘作一个全面的介绍：从硬件设计到软件调试到驱动编写以及 FAT32 文件系统；在第 9 章中，我们将全面介绍各种 LCD 的原理和特点，以及它们的硬件设计和软件调试；在第 10 章中，我们将对 PCMCIA 接口规范、PCMCIA 接口芯片，以及 PCMCIA 接口逻辑实现作全面介绍；在第 11 章中，我们将会和大家探讨嵌入式系统中的网络接口设计，我们会提供 Modem、以太网、GPRS 到无线以太网的上网手段的接口实现到软件编写；最后，我们将在第 12 章，对嵌入式系统的 BootLoader 作一个介绍，并给出几个从 Flash、NAND Flash 和硬盘启动 Linux 操作系统的 BootLoader 示例。

第 2 章 32 位嵌入式处理器

嵌入式处理器的种类繁多,32 位嵌入式处理器虽然只占整个嵌入式处理器的很小一部分,但种类仍然不少。这些处理器大都采用 RISC 构架(本章第一小节会分析其中的原因),因此本章将就一些应用比较广泛的 32 位 RISC 嵌入式处理器作一些介绍。

2.1 RISC 的介绍

2.1.1 RISC 的历史

早期的计算机部件相当昂贵,主频低运算速度很慢。为了提高运算速度,人们不得不将越来越多的复杂指令加入到指令系统中以提高计算机的处理效率,这就逐渐形成了复杂指令集计算机(CISC——Complex Instruction Set Computer)的体系。为了在有限的指令长度内实现更多的指令,人们又设计了操作码扩展。然后,为了达到操作码扩展的先决条件——减少地址码,设计师们又动足了脑筋,发明了各种各样的寻址方式,如基址寻址、相对寻址等,以最大限度地压缩地址码长度,为操作码留出空间。拿典型的采用 CISC 体系的 x86 系列 CPU 来说,从 8086 到现在的 P4,每出一代新的 CPU,就会有自己新的指令,而为了兼容原来的软件,旧的 CPU 的指令集又必须保留,这就使指令解码系统越来越复杂,并且已经到了欲罢不能的地步。

1975 年,IBM 公司的设计师 John Cocke 研究了当时的 IBM370 CISC 系统,发现其中占总指令数仅 20%的简单指令却在程序调用指令数中占到了 80%,而占总指令数 80%的复杂指令却只有 20%的机会被用到。由此,他提出了精简指令集计算机(RISC)的概念。

RISC 体系结构的基本思路是,抓住 CISC 指令系统指令种类繁多、指令格式不规范、寻址方式复杂的缺点,通过减少指令种类、规范指令格式和简化寻址方式,以方便处理器内部的并行处理,提高 VLSI 器件的使用效率,从而大幅度地提高处理器的性能。

RISC 概念的提出引起了计算机体系构架的一次影响深远的革命。经过 20 多年的发展,RISC 已经取得了巨大的成功,比较常见的 RISC 处理器有 DEC/Compaq Alpha、MIPS、HP PA-RISC 系列、SUN Sparc 系列、IBM/Motorola PowerPC, ARM 系列和 Motorola Cold Fire 系列。

RISC 的发展甚至也深深影响了 CISC 的发展,当今 CISC 在设计时也采用了 RISC 的思想,即在 CISC 内核里面放一个 RISC 内核和一个微代码解码器;然后将 CISC 指令通过解码器解码成 RISC 指令交给 RISC 内核执行。也正是因为这个原因,有的专家指出:传统的 RISC 与 CISC 之争已经过时,现在所有的 CISC 处理器都是 RISC 处理器,都可通过转换硬件实现与 x86 指令集兼容。而现有的 x86 等 CISC 处理器之所以不转为 RISC 结构,也只是因为需要兼容现有的大量软件。

值得注意的是,现在又有一种新的计算机体系被提出来,那就是显式并行指令计算机

(EPIC——Explicitly Parallel Instruction Computer)，它是 Intel 和惠普两家公司为 Itanium 处理器所创立的新型指令系统。EPIC 结合了 RISC 和 CISC 的优点，以一个超长指令为大的执行单位，长指令又被分割成较短的小指令，EPIC 处理器在运行中可并行执行这些小指令，使得处理效率大大提高。在技术上 EPIC 比 CISC 和 RISC 更为优越，但它的缺点在于，由于指令体系是全新的，包括操作系统、驱动程序和各类应用软件都必须从零开始开发，应用软件缺乏将是 Itanium 平台推广的最大障碍。

2.1.2 RISC 的特点

为了简化指令解释模块，RISC 的指令格式非常严格。尽管有些指令的执行周期超过一个时钟周期，但 RISC 为了简化设计并提高效率而引入了流水线 (pipeline) 的概念，从而使几乎所有指令都能在一个时钟周期内完成。类似汽车生产线那样，RISC 把一个相对复杂的指令拆分成更小的指令，这些小指令放在流水线上等待执行，每一个时钟周期，都有好几条指令同时执行，只不过不同的指令处于不同的执行阶段而已。同样是为了简化处理器结构，RISC 只提供了很少的寻址方式，而大量采用寄存器作为操作单元。这使得 RISC 的寄存器比 CISC 要多得多，但这也大大提高了 RISC 的运行效率，因为如果频繁访问慢速的内存显然会降低 CPU 的速度。使用 RISC 指令集，可以大大简化处理器控制器以及其他功能单元的设计，特别是允许以硬件线路来实现指令操作，而不必像 CISC 处理器那样使用微程序来实现指令操作。因此，RISC 处理器不必像 CISC 处理器那样设置微程序控制存储器，就能够快速地直接执行指令。RISC 的特点可以总结如下。

- (1) 指令种类少，指令格式规范。RISC 指令集通常只使用一种或少数几种格式。指令长度单一（一般 4B），并且在字边界上对齐。字段位置特别是操作码的位置是固定的。
- (2) 利用流水线和超标量技术让处理器在一个时钟周期可以完成一条或多条指令。
- (3) 寻址方式简化。几乎所有指令都使用寄存器寻址方式，寻址方式总数一般不超过 5 个。其他更为复杂的寻址方式，如间接寻址等则由软件利用简单的寻址方式来合成。
- (4) 大量利用寄存器间操作。RISC 指令集中的大多数操作都是寄存器到寄存器操作，只以简单的 Load 和 Store 操作访问内存。因此，每条指令中访问的内存地址不会超过 1 个，访问内存的操作不会与算术操作混在一起。

2.1.3 RISC 和 CISC 的比较

首先，RISC 的运行速度更快。因为 RISC 的特性适于大量采用流水线和超标量技术，这使得同样时钟频率和技术手段下，RISC 的指令执行速度往往是 CISC 的 2~4 倍。其次，因为 RISC 结构简单，所占用的芯片面积较 CISC 要小得多，这有利于在芯片上集成更多的功能和模块。最后，RISC 的简单结构也使 RISC 的设计周期比 CISC 短，这使得 RISC 往往可以最及时地利用最新的技术和工艺以达到更高的性能。表 2-1 是 RISC 和 CISC 的特点比较。

表 2-1 CISC 和 RISC 特点比较

	CISC	RISC
1	复杂指令的执行需要更多的时钟周期	简单指令只需一个时钟周期
2	所有指令都可访问内存	只有 loads/stores 指令可访问内存
3	无流水线或流水线程度较低	流水线结构