



普通高校“十二五”规划教材

嵌入式Linux系统原理

——基于ARM Cortex-A8处理器

温淑鸿 主编

田沛 朱兵 宋金宝 苗方 编著



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS



普通高校“十二五”规划教材

嵌入式 Linux 系统原理

——基于 ARM Cortex - A8 处理器

温淑鸿 主编

田沛 朱兵 宋金宝 苗方 编著

北京航空航天大学出版社

内 容 简 介

本书的编写不求全面详尽,但力求简单实用、容易上手,能够在短时间内掌握在 ARM Cortex-A8 处理器上移植 Linux 操作、开发设备驱动的基本过程。书中简单介绍了嵌入式系统的发展史和嵌入式系统的基本特征,内容包括:嵌入式系统的设计方法和测试方法;嵌入式系统的硬件基础;常见的嵌入式操作系统;嵌入式 Linux 开发环境;嵌入式 Linux 操作系统;Linux 设备驱动统一模型;总线、设备和驱动等。

本书适合作为高等学校电子、电信类专业的教材,也可作为 Linux 驱动开发人员及自学嵌入式开发人员的参考用书。

图书在版编目(CIP)数据

嵌入式 Linux 系统原理:基于 ARM Cortex-A8 处理器 /
温淑鸿主编. — 北京:北京航空航天大学出版社,
2014.6

ISBN 978-7-5124-1431-0

I. ①嵌… II. ①温… III. ①Linux 操作系统—程序
设计 IV. ①TP316.89

中国版本图书馆 CIP 数据核字(2014)第 053379 号

版权所有,侵权必究。

嵌入式 Linux 系统原理 ——基于 ARM Cortex-A8 处理器

温淑鸿 主编

田 沛 朱 兵 宋金宝 苗 方 编著
责任编辑 潘晓丽 刘秀清 张雯佳

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316524

涿州市新华印刷有限公司印装 各地书店经销

*

开本:710×1 000 1/16 印张:27.5 字数:586 千字

2014 年 6 月第 1 版 2014 年 6 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-1431-0 定价:59.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024

前 言

本书的适用对象

本书适合作为高等学校电子、电信类专业的教材,也可作为 Linux 驱动开发人员及自学嵌入式开发人员的参考用书;还可作为高等学校学生和技术开发人员学习在 ARM 处理器上移植 Linux 操作系统、开发设备驱动的参考书籍。

本书的特点

本书的编写不求全面详尽,但求简单实用、容易上手,且能够有助于在短时间内掌握在 ARM 处理器上移植 Linux 操作系统及开发设备驱动的基本过程。举例来说,在介绍 Linux shell 命令时,只介绍 bash 最常用的命令,对 shell、bash 的很多不常用的命令、常用命令的不常用的功能选项基本上不介绍。又比如:设备驱动一章, Linux 设备按功能可以分为字符设备、块设备、网络设备,本书仅介绍层次关系较为简单的字符设备和网络设备,没有介绍块设备。在介绍 Linux 设备驱动统一模型时,则详细地介绍了 bus_type、device 和 device_driver 三者的注册过程,并首先介绍最为简单最为常用的 platform_bus_type(即 CPU 存储器总线,处理器片上的周边设备和片外通过存储器总线扩展的设备都是通过存储器总线和 CPU 连接的),然后介绍了相对简单的 I²C 总线 i2c_bus_type、SPI 总线 spi_bus_type。platform_bus_type、i2c_bus_type、spi_bus_type 体现了设备和处理器的连接,而普通用户更关注的是设备的功能, Linux 内核把设备按功能分成若干类,本书只介绍特别简单的实时时钟 RTC 类的驱动。

本书的内容组织

第 1 章 简单介绍嵌入式系统的发展史和嵌入式系统的基本特征。

第 2 章 简单介绍嵌入式系统的设计方法和测试方法。

第 3 章 介绍嵌入式系统的硬件基础,着重介绍 ARM 处理器结构、指令集、汇编语言格式,最后介绍了存储器:内存、Nor flash 和 Nand flash。

第 4 章 介绍常见的嵌入式操作系统。

第 5 章 介绍嵌入式 Linux 开发环境,包括 shell 命令、makefile、编译器、调试器、交叉编译器、交叉调试器。

第 6、7、8 章 分别介绍嵌入式 Linux 操作系统的三个部分:Bootloader、内核、根文件系统。Bootloader 只介绍了 u-Boot。

第 9 章 介绍 Linux 设备驱动统一模型:总线、设备和驱动。总线最先注册进内核,然后是设备和驱动注册进内核。设备注册时在总线上寻找匹配的驱动,驱动注册时在总线上寻找匹配的设备。例如 i2c_adapter 注册时寻找连接在该 i2c_adapter 上的 i2c_client 从设备并注册,i2c_client 注册时会寻找匹配的驱动,i2c_driver 注册时寻找它能驱动的设备。spi_master 注册时寻找连接在该 spi_master 上的 spi_device 从设备并注册,spi_device 注册时会寻找匹配的驱动,spi_driver 注册时寻找它能驱动的设备。

本书编者温淑鸿、田沛、朱兵、宋金宝、苗方,都是中国传媒大学信息工程学院的教师,长期从事“嵌入式计算机系统”课程的讲授和嵌入式系统相关的科研工作,积累了丰富的教学经验。

虽然编者对本书进行了认真的检查、校对,但由于知识有限、时间仓促,书中肯定有不少错误,恳请读者批评指正。

本书在编写过程中得到了北京精仪达盛科技有限公司的支持,在此特别表示感谢,感谢北京精仪达盛科技有限公司的毕才术、高晓慧,他们提供了 EL-ARM-860 实验箱及配套 TECHV-OMAP3530 核心板,为本书的完成提供了良好的测试环境。

编 者

2014 年 5 月

于中国传媒大学信息工程学院

说明:

本书配有教学课件,有需要的教师,请联系 emsbook@gmail.com 索取。



录

第 1 章 嵌入式系统概述	1
1.1 嵌入式系统的发展史	2
1.1.1 嵌入式系统发展的 4 个阶段	2
1.1.2 嵌入式系统的发展趋势	2
1.2 嵌入式系统的定义、组成与特点	3
1.2.1 嵌入式系统的定义	3
1.2.2 嵌入式系统的组成	3
1.2.3 嵌入式系统的特征	3
第 2 章 嵌入式系统的设计方法	5
2.1 传统的软硬件分开设计方法	5
2.2 软硬件协同设计方法	6
2.3 嵌入式系统的测试技术	7
2.3.1 嵌入式软件的测试方法	7
2.3.2 嵌入式系统的测试策略	8
2.3.3 嵌入式软件的测试工具	9
2.4 小 结	10
第 3 章 嵌入式系统硬件基础	11
3.1 嵌入式处理器	11
3.1.1 嵌入式处理器的基本特征	11
3.1.2 嵌入式处理器的分类	11
3.1.3 典型嵌入式处理器	12
3.1.4 嵌入式处理器的发展趋势	14

3.1.5	嵌入式处理器的选型方法	15
3.2	ARM 处理器	16
3.2.1	ARM 处理器系列	16
3.2.2	Cortex A8 处理器	27
3.2.3	ARM 公司的 NEON 技术	28
3.2.4	ARM+协处理器架构	30
3.2.5	ARM 处理器的结构	32
3.2.6	ARM 的指令集	43
3.2.7	ARM 汇编程序设计基础	51
3.3	存储器	73
3.3.1	SDRAM	73
3.3.2	DDR	74
3.3.3	DDR2	74
3.3.4	DDR3	75
3.3.5	Nor Flash	78
3.3.6	Nand Flash	80
3.3.7	Nor Flash 与 Nand Flash 的区别	81
第 4 章	嵌入式操作系统	82
4.1	嵌入式操作系统的特征	82
4.2	嵌入式实时操作系统	83
4.2.1	嵌入式实时操作系统的概念	83
4.2.2	嵌入式实时操作系统的发展过程	84
4.2.3	实时操作系统的几个评价指标	85
4.3	嵌入式软实时操作系统	86
4.4	嵌入式操作系统的种类	87
4.4.1	嵌入式 Linux	87
4.4.2	VxWorks 操作系统	88
4.4.3	Windows CE 操作系统	89
4.4.4	μ C/OS 操作系统	91
4.4.5	Palm Os 操作系统	93
第 5 章	嵌入式 Linux 系统开发环境	94
5.1	bash 常用 Shell 命令	94
5.1.1	常用 Shell 命令	95
5.1.2	git 版本控制	106

5.2	Shell 脚本编程	119
5.2.1	执行 Shell 脚本的方式	120
5.2.2	Shell 特殊字符	122
5.2.3	Shell 变量	127
5.2.4	使用整数	129
5.2.5	分支语句	129
5.2.6	test 测试语句	133
5.2.7	循环控制语句	136
5.3	GNU 编译器 GCC	141
5.3.1	预处理阶段	142
5.3.2	编译阶段	145
5.3.3	汇编阶段	145
5.3.4	链接阶段	145
5.3.5	GCC 的编译选项	146
5.4	GNU 调试器:GDB	147
5.5	GNU Makefile	158
5.5.1	GNU Make	158
5.5.2	Makefile 语法规则	159
5.5.3	Makefile 文件中变量的应用	159
5.6	交叉编译工具链及交叉调试器	162
5.6.1	交叉编译的基础知识	162
5.6.2	交叉编译器的生成步骤	163
5.6.3	远端调试	163
5.7	主机开发环境的搭建	167
第 6 章	嵌入式 Linux 系统的 Bootloader	179
6.1	Bootloader 的基本概念	179
6.2	Bootloader 的种类	179
6.3	u-Boot 的 2 种操作模式	180
6.4	u-Boot 的启动过程	181
6.4.1	Bootloader 的存储位置	181
6.4.2	u-Boot 向 Linux 内核传递参数	182
6.5	u-Boot 的命令解析器	182
6.5.1	旧的、简单的命令行分析器	182
6.5.2	hush Shell	182
6.5.3	命令解析器的具体实现	184

6.6	怎样给 u-Boot 添加命令	184
6.7	u-Boot 的环境变量与常用命令	186
6.7.1	u-Boot 的环境变量	186
6.7.2	u-Boot 的常用命令	187
6.8	u-Boot 引导 Linux 内核参数设置	191
第 7 章	Linux 内核	193
7.1	Linux 内核简介	193
7.1.1	Linux 内核的功能	193
7.1.2	Linux 内核的特点	195
7.2	Linux 内核配置过程	196
7.2.1	Linux 内核配置的菜单生成工具	201
7.2.2	Kconfig 语法	202
7.3	Linux 内核编译	207
7.4	Linux 内核模块	212
7.4.1	外部内核模块的构建	213
7.4.2	Linux 内核模块的调试	217
第 8 章	常用 Linux 文件系统	230
8.1	Linux 文件系统概述	230
8.1.1	根文件系统	233
8.1.2	日志文件系统	236
8.1.3	Ramdisk	237
8.2	ext2 文件系统	237
8.3	ext3 文件系统	238
8.4	Reiserfs 文件系统	239
8.5	JFFS2 文件系统	240
8.6	YAFFS 文件系统	241
8.7	CRAMFS 文件系统	243
8.8	UBIFS 文件系统	244
第 9 章	Linux 设备驱动	245
9.1	设备驱动基本概念	245
9.1.1	Linux 设备驱动的分类	245
9.1.2	Linux 驱动中 I/O 端口和 I/O 内存的访问	247
9.2	字符设备驱动	252

9.2.1	struct cdev	252
9.2.2	字符设备的注册	254
9.3	Linux 网络设备驱动	265
9.3.1	网络设备接口层	266
9.3.2	一个以太网设备驱动示例	284
9.3.3	小 结	298
9.4	sysfs 文件系统	298
9.4.1	udev	299
9.4.2	sysfs 挂载点	299
9.4.3	sysfs 文件系统的挂载	299
9.4.4	sysfs 文件系统下的目录结构	300
9.4.5	sysfs 的实现机制:kobject	302
9.4.6	sysfs 的文件和目录的创建	309
9.5	Linux 设备驱动基本模型	314
9.5.1	总线(bus_type)	314
9.5.2	device	321
9.5.3	device_driver	336
9.6	platform_bus_type	344
9.6.1	platform_bus_type	344
9.6.2	platform_device	345
9.6.3	platform_driver	358
9.7	I2C 总线及设备驱动	360
9.7.1	I2C 总线工作原理	361
9.7.2	Linux 中 I2C 驱动架构	362
9.7.3	Linux I2C 驱动中的 i2c_bus_type	362
9.7.4	Linux I2C 驱动中的 i2c_adapter	364
9.7.5	i2c_adapter 的注册	367
9.7.6	i2c_client	374
9.7.7	I2C 从设备的驱动注册	385
9.8	SPI 总线及设备驱动	388
9.8.1	SPI 接口协议	388
9.8.2	Linux 驱动模型中的 spi_bus_type	392
9.8.3	spi_master	393
9.8.4	spi_master 的注册	397
9.8.5	spi_device	400
9.8.6	spi_device 的实例化	403

目 录

9.8.7	spi_device 的驱动 spi_driver	407
9.9	Linux 设备驱动的基本思想	408
9.9.1	隔离思想	408
9.9.2	面向对象的思想	410
9.10	实时时钟 RTC 驱动	410
9.10.1	RTC 类的初始化	412
9.10.2	RTC 类设备的注册	413
附录	EL-ARM-860 型嵌入式实验开发系统	418

第 1 章

嵌入式系统概述

嵌入式技术的快速发展,使其成为当今计算机技术和电子技术的一个重要分支。嵌入式系统具有非常广阔的应用领域,是现代计算机技术改造传统产业、提升多领域技术水平的有力工具,可以说嵌入式系统无处不在。其主要应用领域包括智能产品(智能仪表、智能和信息家电)、工业自动化(测控装置、数控机床、数据采集与处理)、办公自动化(通用计算机中的智能接口)、电网安全、电网设备检测、石油化工、商业应用(电子秤,POS机,条码识别机)、安全防范(防火、防盗、防泄漏等报警系统)、网络通信(路由器、网关、手机、PDA、无线传感器网络)、汽车电子与航空航天(汽车防盗报警器、汽车和飞行器黑匣子)以及军事等,如图 1.1 所示。

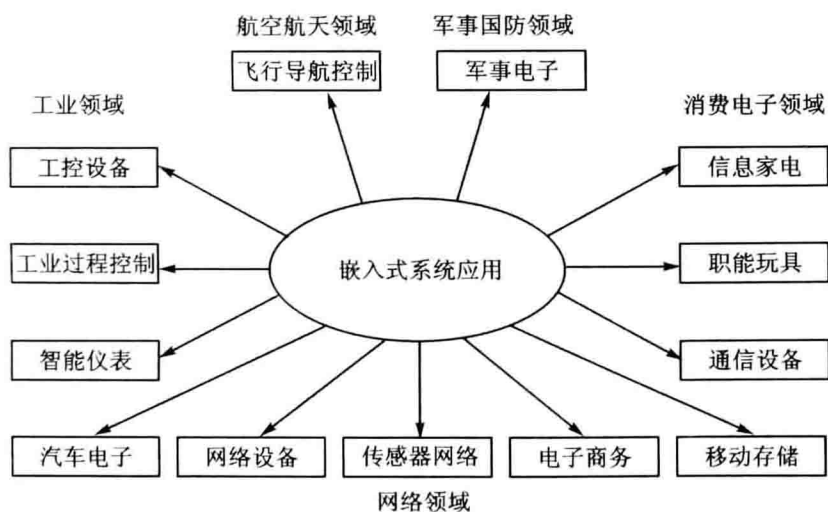


图 1.1 嵌入式系统的应用领域

嵌入式系统在很多产业中得到了广泛的应用并逐步改变着这些产业,包括工业自动化、国防、运输和航空航天领域。神州飞船和长征系列火箭系统中就有很多嵌入式系统;导弹的制导系统也有嵌入式系统;在日常生活中,人们也在使用各种嵌入式系统。事实上,几乎所有带有一点“智能”的家电(全自动洗衣机、电脑、电饭煲等)都使用了嵌入式系统,嵌入式系统具有广泛的适应能力和多样性。

1.1 嵌入式系统的发展史

20世纪60年代末期,随着微电子技术的发展,嵌入式计算机逐步兴起。随着计算机技术、通信技术、电子技术一体化进程不断加快,目前嵌入式技术已成为广大技术人员的研究热点。

1.1.1 嵌入式系统发展的4个阶段

1. 以微控制器为核心的无操作系统阶段

以微控制器为核心的可编程控制器形式的低级嵌入式系统是嵌入式系统发展的第一阶段。它具有监测、伺服、指示设备等功能,应用于专业性很强的工业控制系统中,通常不含操作系统。该阶段的嵌入式系统处于低级阶段,其主要特点是系统结构和功能单一、处理器工作频率低、存储容量较小、使用简单、成本低。

2. 以嵌入式微处理器为基础的简单操作系统阶段

以嵌入式微处理器为基础,以简单操作系统为核心的初级嵌入式系统是嵌入式系统发展的第二阶段。其主要特点是处理器种类多,通用性较弱;系统效率高,成本低;操作系统具有兼容性、扩展性,但用户界面简单。

3. 以嵌入式操作系统为标志的中级嵌入式系统

以嵌入式操作系统为标志的中级嵌入式系统是嵌入式系统发展的第三阶段。其主要特点是嵌入式系统能运行于各种不同嵌入式处理器上,兼容性好;操作系统内核小、效率高,并且可任意裁剪;具有文件和目录管理、多任务功能,支持网络,具有图形窗口以及良好的用户界面;具有大量的应用程序接口,嵌入式应用软件丰富。

4. 以Internet为标志的高级嵌入式系统

以Internet为标志的高级嵌入式系统是嵌入式系统发展的第四阶段。随着网络应用的不断深入,随着信息家电的发展,嵌入式系统的应用必将与Internet有机结合在一起,成为嵌入式系统发展的未来。

1.1.2 嵌入式系统的发展趋势

以信息家电为代表的互联网时代嵌入式产品,不仅为嵌入式市场展现了美好的前景,还注入了新的生命。

1. 支持小型电子设备实现小尺寸、低功耗和低成本

这种特性要求嵌入式产品设计者相应降低处理器的性能,限制内存容量和复用接口芯片。这提高了对嵌入式软件设计技术的要求,使设计者不得不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。这不但要求软件人员有丰富经验,更需要发展先进嵌入式软件技术,如Java、Web和WAP等。

2. 提供精巧的多媒体人机界面

嵌入式设备之所以为亿万用户乐于接受,重要因素之一是它们与使用者之间的亲和力以及自然的人机交互界面,如司机操纵高度自动化的汽车,主要还是通过习惯使用的方向盘、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。

1.2 嵌入式系统的定义、组成与特点

1.2.1 嵌入式系统的定义

嵌入式系统(Embedded System)是嵌入式计算机系统的简称,简单地说,嵌入式系统就是嵌入到目标体系中的专用计算机系统。嵌入性、专用性及计算机系统是嵌入式系统的 3 个基本要素。具体地讲,嵌入式系统是指以应用为中心,以计算机技术为基础,并且软硬件可裁剪,适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。也就是说嵌入式系统是把计算机直接嵌入到应用系统中,它融合了计算机软硬件技术、通信技术和微电子技术,是集成电路发展过程中的一个标志性的成果。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户应用程序 4 个部分组成,用于实现对其他设备的控制、监视或管理等功能。

1.2.2 嵌入式系统的组成

嵌入式系统既然是一种专用的计算机应用系统,当然也包括硬件和软件两大部分。嵌入式系统的硬件包括电源模块、嵌入式处理器、存储器(程序存储器和数据存储器)、可编程逻辑器件、嵌入式系统周边元器件、各种 I/O 接口、总线以及外部设备。有些应用场合要求具有 USB 接口、I²C 总线接口、SPI 接口、CAN 总线接口、以太网接口以及 A/D、D/A、PWM 等接口。因此,嵌入式系统的硬件要根据实际应用选择或裁剪,以最低成本满足应用系统的要求。嵌入式系统的软件包括设备驱动层、嵌入式操作系统、应用程序接口 API 层以及实际应用程序层。对于简单的嵌入式系统来说,可以没有嵌入式操作系统,仅存在设备驱动程序和应用程序。对于大部分嵌入式系统,由于性能要求越来越高,通常需要嵌入式操作系统。

1.2.3 嵌入式系统的特征

嵌入式系统与以 PC 为代表的通用计算机系统相比,其嵌入性、专用性及计算机系统是嵌入式系统的 3 个基本要素。不同的嵌入式系统其特点会有所差异,其主要特点概括如下:

1. 嵌入式系统是专用的计算机系统

嵌入式系统的硬件、软件均是面向特定应用和任务而设计的,具有很强的专用性和多样性。嵌入式系统提供的功能以及面对的应用和过程都是预知的,相对固定的,而不像通用计算机那样具有很大的随意性。

2. 嵌入式系统必须满足环境的要求

由于嵌入式系统要嵌入到目标系统中,因此它必须满足目标系统的环境要求,如物理环境(集成度高、体积小)、电气环境(可靠性高)、成本低(价廉)、功耗低(能耗少)等高性价比要求,另外还要求它能满足对温度、湿度、压力等自然环境的要求,民用和军用嵌入式系统对自然环境的要求差别很大。

3. 嵌入式系统是集计算机技术与各行业应用于一体的集成系统

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。

4. 嵌入式系统具有较长的生命周期

嵌入式系统和实际应用有机地结合在一起,它的更新换代也是和实际产品一同进行的,因此基于嵌入式系统的产品一旦进入市场,就具有较长的生命周期。

5. 嵌入式系统的软件固化在非易失性存储器中

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Flash等非易失性存储器中,而不是像通用计算机系统那样存储于磁盘等载体中。

6. 多数嵌入式系统具实时性要求

许多嵌入式系统都有实时性要求,需要有对外部事件迅速反应的能力。以前,嵌入式系统几乎是实时系统的代名词,近年来出现了许多不带实时要求的嵌入式系统,所以这两个词的区别才变得显著起来。但是,多数嵌入式系统还是有着不同程度的实时性要求的。

7. 嵌入式系统需专用开发环境和开发工具

嵌入式系统需专用开发环境和开发工具进行设计。

第2章

嵌入式系统的设计方法

2.1 传统的软硬件分开设计方法

传统的嵌入式系统设计方法如图 2.1 所示。该方法将硬件和软件分为两个独立的部分,由硬件设计人员和软件设计人员按照拟定的设计流程分别完成。这种设计方法只能改善硬、软件各自的性能,而有限的设计空间不可能对系统做出较好的性能综合优化。一般来说,每一个应用系统,都存在一个适合于该系统的硬、软件功能的最佳组合,如何从应用系统需求出发,依据一定的指导原则和分配算法对硬、软件功能进行分析及合理的划分,从而使系统的整体性能、能量消耗达到最佳状态,已成为硬、软件协同设计的重要研究内容之一。

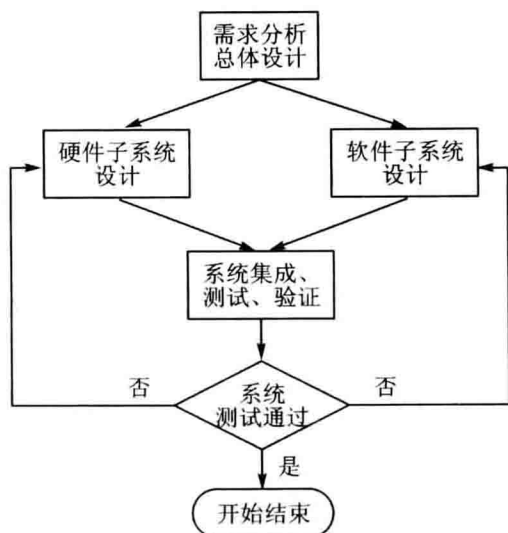


图 2.1 传统嵌入式系统设计方法

传统的嵌入式系统设计过程描述如下:

- ① 要求分析;
- ② 硬、软件分别设计、开发、调试、测试;
- ③ 系统集成;
- ④ 集成测试;
- ⑤ 若系统正确,则结束,否则继续进行;
- ⑥ 若出现错误,则需要对硬、软件分别验证和修改;
- ⑦ 返回③。

显然,这种设计方法对于具体的应用系统而言,是容易满足综合性能指标的最佳解决方案。传统方法虽然也可改进硬、软件性能,但由于这种改进是各自独立进行的,不一定能使系统综合性能达到最佳。传统的嵌入式系统开发采用的是硬件开发

第2章 嵌入式系统的设计方法

与软件开发分离的方式。虽然在系统设计的初始阶段考虑了硬、软件的接口问题,但由于硬、软件分别开发,各自部分的修改和缺陷很容易导致系统集成出现错误。

2.2 软硬件协同设计方法

为避免传统嵌入式系统开发过程中出现的弊端和问题,一种新的开发方法应运而生,即硬、软件协同设计方法。一个典型的硬、软件协同设计过程如图 2.2 所示,它与传统设计相比有 2 个显著的特点:

- ① 描述软硬件使用统一的表示形式;
- ② 软硬件划分可以选择多种方案,直到满足要求。

首先,应用独立于硬、软件功能规格的方法对系统进行描述,采用的方法包括有限态自动机(FSM)、统一化的规格语言(CSP、VHDL)或其他基于图形的表示工具,其作用是对硬、软件统一表示,便于功能的划分和综合。

然后,在此基础上对硬、软件进行划分,即对硬、软件的功能模块进行分配。但是,这种功能分配不是随意的,而是从系统功能要求和限制条件出发,依据算法进行的。完成硬、软件功能划分之后,需要对划分结果做出评估。方法之一是性能评估,另一种方法是对硬、软件综合之后的系统依据指令级评估硬、软件模块,以上过程重复直到系统获得一个满意的硬、软件实现为止。

软硬件协同设计过程可归纳为:

- ① 需求分析;
- ② 软硬件协同设计;
- ③ 软硬件实现;
- ④ 软硬件协同测试和验证。

这种方法的特点是在协同设计(Co-Design)、协同测试(Co-Test)和协同验证(Co-Verification)上,充分考虑了硬、软件的关系,并在设计的每个层次上给以测试验证;尽早发现和解决问题,避免灾难性错误的出现;提高了系统开发效率,降低了开发成本。

需要说明的是,对于许多应用场合的嵌入式系统,并不排除传统的嵌入式系统设计方法,因为这种方法无论是开发经验还是对开发工具的使用都已经广为人们所熟悉,不能一味盲目追求硬、软件协同设计。

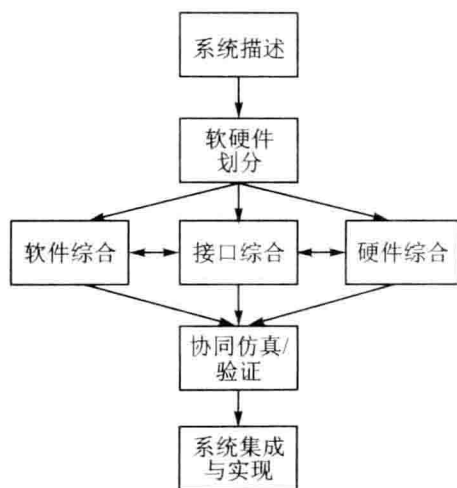


图 2.2 嵌入式系统的软硬件协同设计方法