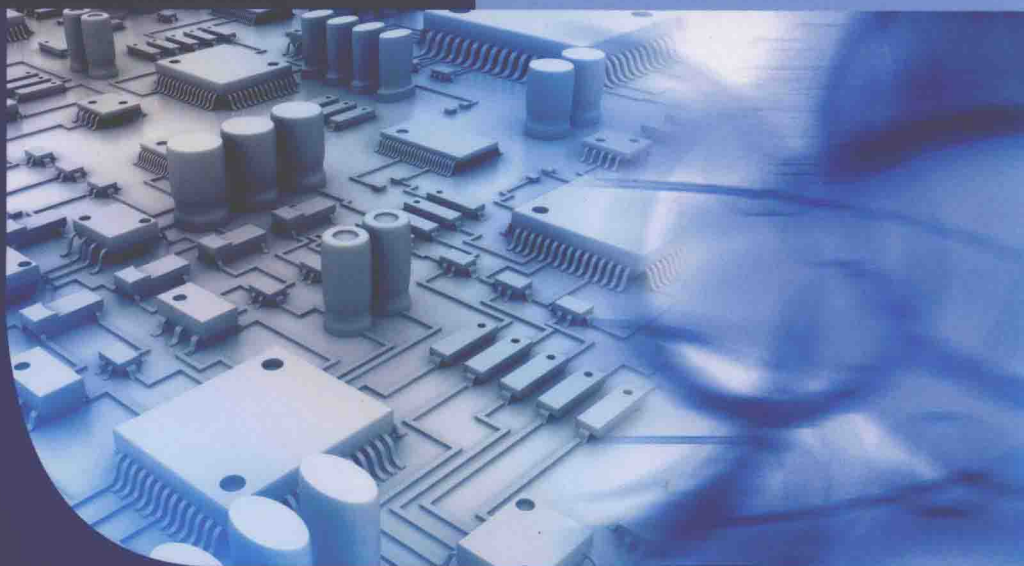


普通高等教育“十三五”规划教材

嵌入式系统设计

© 张永辉 主编

EMBEDDED SYSTEM ANALYSIS



免费电子课件



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十三五”规划教材

嵌入式系统设计

张永辉 主编

杨永钦 易家傅 副主编



机械工业出版社

本书从基础的理论知识到实际的应用开发,详细地介绍了 ARM Cortex-M0 处理器的内核体系结构与特性,包括总线架构、编程模型、存储器模型、异常模型、电源管理、系统控制模块、嵌套向量中断控制器和系统节拍定时器等核内外设,结合 CMSIS 对内部函数和外设的操作进行了详解。以恩智浦半导体(NXP)公司的 ARM Cortex-M0 处理器的 LPC1114 微控制器为例,详细介绍了 LPC1100 系列的内核结构与高级外设的工作原理与应用开发方法,以及基于 CMSIS 接口标准的软件设计方法。

本书的 LPC1114 最小应用系统板和开发板电路设计和仿真是基于 Proteus 仿真软件平台完成的,读者可以在没有硬件的情况下进行仿真调试,也可以利用本书所提供的 Proteus 项目文件自行制板、购买元器件和焊板,完成 LPC1114 最小应用系统板和开发板的制作,进一步制作扩展电路。

本书可作为普通高校电子信息专业、通信专业、仪器专业、自动化专业的本科生及研究生的入门教材,也可供从事嵌入式系统设计的研发人员参考。

本书配有免费电子课件,欢迎选用本书作教材的老师登录 www.cmpedu.com 注册下载,或发邮件到 jincamp@163.com 索取。

图书在版编目(CIP)数据

嵌入式系统设计 / 张永辉主编. —北京:机械工业出版社, 2018.10
普通高等教育“十三五”规划教材
ISBN 978-7-111-60943-8

I. ①嵌… II. ①张… III. ①微型计算机—系统设计
—高等学校—教材 IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2018)第 215154 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:吉玲 责任编辑:吉玲 刘丽敏

责任校对:王明欣 封面设计:张静

责任印制:李昂

河北宝昌佳彩印刷有限公司印刷

2019 年 1 月第 1 版第 1 次印刷

184mm×260mm·18 印张·445 千字

标准书号:ISBN 978-7-111-60943-8

定价:45.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:010-88379833

机工官网:www.cmpbook.com

读者购书热线:010-88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

前 言

随着计算机技术、网络技术和微电子技术的深入发展,嵌入式系统的应用无处不在。ARM 是目前公认的业界领先的 32 位嵌入式 RISC (精简指令计算机) 微处理器,并且全世界有 200 多家领先的半导体厂商获得 ARM 授权,组成了一个牢靠的产业联盟,每秒可生产 90 个芯片,基于 ARM 体系结构的微处理器占领了 32 位嵌入式系统领域的大部分份额。

ARM 公司的微控制器 Cortex 系列为嵌入式市场提供了一个三管齐下的解决方案。Cortex-A 系列提供高性能应用处理器,Cortex-R 系列专门针对实时、深度嵌入式解决方案,Cortex-M 系列专注于低成本嵌入式微控制器。ARM Cortex-M 是一系列可向上兼容的高能效、易于使用的处理器,可以帮助开发人员满足将来的嵌入式应用的需要,以更低的成本提供更多功能、不断增加连接、改善代码重用和提高能效。Cortex-M0 作为 Cortex-M 系列的一款基于 ARMv6M 架构的 RISC 处理器,集 8 位单片机的价格和 32 位 ARM 处理器的性能于一身,为目前能耗最低的最小的 ARM 处理器之一。

NXP 公司的 LPC1100 系列处理器是全球首个基于 Cortex-M0 内核的微控制器系列,时钟速度可达 50MHz,每秒 4500 多万条指令,可以实现更低的功耗和更长的电池寿命,是电池供电的消费电子设备、智能仪表、电机控制等的理想之选。LPC1100 系列丰富,不仅有基本 LPC1100 系列,还有 LPC1100L 更低功耗系列、LPC11U00 USB 系列、LPC11C00 CAN 总线系列等多种选择。LPC1100 系列得到广泛的工具厂商的支持,如 Keil (ARM 旗下公司)、IAR Systems、Hitex Development Tools、Embedded Artists 和 Labcenter Electronics 等公司。

本书主要以 LPC1100 系列微控制器为硬件平台,详细介绍 Cortex-M0 微控制器的原理与开发技术,基于 Proteus 仿真软件平台设计最小应用系统板和具有在板仿真器的口袋开发板以及相关例程。

第 1 章简要介绍嵌入式系统的定义、分类、历史以及发展趋势,主要介绍嵌入式操作系统的基本知识和对 ARM Cortex-M 处理器的支持情况,并对 NXP 公司 ARM Cortex-M0 系列处理器及集成开发环境和仿真软件平台进行简要介绍。

第 2 章主要介绍 ARM Cortex-M0 处理器的结构与特性,内核体系结构的介绍包括总线架构、编程模型、存储器模型、异常模型、电源管理和系统控制模块、嵌套向量中断控制器、系统节拍定时器等核内外设,结合 CMSIS 对内部函数和外设的操作进行讲解。

第 3 章主要介绍 LPC1100 系列处理器的基础部分,包括其基本结构、存储器管理、时钟与 PLL、引脚描述与 I/O 配置、GPIO、中断和串行线调试接口,在此基础上讲解 LPC1100 最小系统的设计方法,并给出一个 LPC1114 开发板的设计实例。

第 4 章主要介绍 LPC1100 系列处理器外设的结构及工作原理,包括通用定时器/计数器、看门狗定时器、通用异步收发器 UART、SSP 同步串行端口控制器、I²C 总线接口、A-D 转换器和电源管理单元,并对部分外设进行示例操作。

第 5 章主要介绍基于 CMSIS 接口标准的 ARM Cortex-M0 软件设计,包括 Cortex 微控制器软件接口标准 CMSIS 的基本构架以及如何在 Keil MDK 软件中使用 CMSIS。通过本章的学

习，读者可以掌握基于 CMSIS 接口标准的函数的使用。

本书由张永辉组织完成编写任务，团队成员杨永钦、易家傅、陈敏、潘学松、冯尔理和王萍等多位同事参与了编写工作。刘一鸣、贾烁、王容、谢宇威、张帅岩、何超等硕士研究生对本书的编写进行了资料整理和开发板调试等工作；本书的出版得到了清华大学丁天怀教授的鼎力支持；编写过程中得到了 NXP 公司王朋朋女士、辛华峰先生、梁平先生以及广州风标公司匡载华先生的大力支持和帮助，在此表示衷心的感谢！

ARM Cortex-M0 的功能强大，LPC1100 微控制器的型号丰富，但限于篇幅，本书只对部分内容进行介绍。由于编者的水平有限，书中的错误在所难免，恳请读者批评指正。

张永辉

于海南大学

目 录

前 言

第 1 章 嵌入式系统概述	1
1.1 嵌入式系统	1
1.1.1 嵌入式系统的定义	1
1.1.2 嵌入式系统的历史	1
1.1.3 嵌入式系统的发展趋势	2
1.2 嵌入式处理器	3
1.3 嵌入式操作系统	5
1.3.1 嵌入式操作系统简介	5
1.3.2 嵌入式实时操作系统的特点	6
1.3.3 常用的嵌入式操作系统	6
1.4 ARM Cortex 系列嵌入式处理器	10
1.5 ARM Cortex-M0/M0+处理器	12
1.5.1 Cortex-M0/M0+处理器简介	12
1.5.2 Cortex-M0/M0+处理器的特性	13
1.6 NXP 公司 Cortex-M0/M0+系列处理器	13
1.6.1 LPC800 系列	14
1.6.2 LPC1100/L 系列	16
1.6.3 LPC1200 系列	19
1.7 ARM Cortex-M 处理器开发工具	21
1.7.1 集成开发环境	21
1.7.2 Proteus Design Suite 仿真平台	27
1.7.3 嵌入式操作系统支持	29
习题	30
第 2 章 ARM Cortex-M0 内核体系结构	31
2.1 处理器结构与特性	31
2.2 总线架构	33
2.3 编程模型	34
2.3.1 操作模式和状态	34
2.3.2 堆栈	34
2.3.3 内核寄存器	35
2.3.4 内部函数	38
2.4 存储器模型	39
2.4.1 存储区、类型和属性	39

2.4.2	存储器访问秩序	40
2.4.3	存储器访问的行为	41
2.4.4	存储器的字节存储顺序	41
2.4.5	数据类型	42
2.5	异常模型	42
2.5.1	异常状态	42
2.5.2	异常类型	42
2.5.3	向量表	44
2.5.4	异常优先级	44
2.5.5	异常的进入和返回	44
2.5.6	中断输入及挂起行为	46
2.5.7	故障处理	48
2.5.8	异常处理程序	48
2.6	电源管理	50
2.6.1	进入睡眠模式	50
2.6.2	从睡眠模式唤醒	50
2.6.3	电源管理编程提示	51
2.7	核内外设	51
2.7.1	系统控制模块	51
2.7.2	嵌套向量中断控制器	56
2.7.3	系统节拍定时器	59
习题		62
第3章	LPC1100 系列处理器基础	63
3.1	LPC1100 系列处理器基本结构	63
3.2	存储器管理	64
3.2.1	LPC1100 系列处理器存储器映射	64
3.2.2	异常向量表及其重映射	65
3.2.3	Boot ROM	66
3.2.4	Flash 存储器访问	67
3.3	系统控制模块	67
3.3.1	系统控制模块概述	67
3.3.2	引脚描述	68
3.3.3	系统控制模块寄存器	68
3.3.4	复位	84
3.3.5	掉电检测	84
3.3.6	代码安全与调试——代码安全保护寄存器	84
3.4	时钟系统	86
3.4.1	时钟系统结构	86
3.4.2	振荡器	86

3.4.3	多路选择输出外部时钟	87
3.4.4	PLL 工作原理与使用	88
3.5	引脚描述与 I/O 口配置	92
3.5.1	引脚描述	92
3.5.2	I/O 口的引脚模式	97
3.5.3	I/O 口的配置	97
3.5.4	I/O 配置示例	102
3.6	GPIO 口的结构及功能	104
3.6.1	GPIO 口的结构特点	104
3.6.2	GPIO 口的配置	104
3.6.3	GPIO 口中断	107
3.6.4	GPIO 应用示例	109
3.7	中断源及 NVIC 相关寄存器	110
3.7.1	中断源	110
3.7.2	NVIC 相关寄存器	113
3.8	串行线调试	115
3.8.1	串行线调试概述	115
3.8.2	串行线调试接口	116
3.8.3	SWD 调试接口设置	116
3.9	LPC1100 最小系统和开发板	117
3.9.1	LPC1100 最小系统	117
3.9.2	LPC1114 开发板	122
3.9.3	ISP 程序下载	124
	习题	126
第 4 章	LPC1100 系列处理器外设	127
4.1	定时器/计数器	127
4.1.1	定时器/计数器概述	127
4.1.2	定时器/计数器寄存器	129
4.1.3	定时器中断设置	137
4.1.4	定时器操作示例	139
4.2	通用异步收发器	142
4.2.1	UART 概述	142
4.2.2	UART 接口引脚与配置	144
4.2.3	UART 寄存器	145
4.2.4	RS-485/EIA-485 模式的操作	164
4.2.5	UART 中断	165
4.2.6	UART 接口电路设计	168
4.2.7	UART 程序设计示例	170
4.3	I ² C 总线接口	176

4.3.1	I ² C 总线接口描述	176
4.3.2	I ² C 总线配置	178
4.3.3	I ² C 寄存器与功能描述	178
4.3.4	I ² C 接口中断	186
4.3.5	I ² C 操作模式详解	187
4.3.6	I ² C 状态服务程序	199
4.3.7	I ² C 总线接口应用示例	199
4.3.8	I ² C 程序设计	201
4.4	SSP 同步串行端口控制器	211
4.4.1	引脚描述	212
4.4.2	基本配置	213
4.4.3	寄存器描述	213
4.4.4	SPI 帧格式	217
4.4.5	SSI 帧格式	220
4.4.6	Microwire 帧格式	222
4.4.7	SSP 接口中断设置	223
4.4.8	SPI 接口应用示例	225
4.5	A-D 转换器	230
4.5.1	A-D 转换器概述	230
4.5.2	ADC 引脚描述和配置	230
4.5.3	ADC 寄存器	231
4.5.4	基本操作	234
4.5.5	ADC 中断设置	235
4.5.6	ADC 操作与示例	236
4.6	看门狗定时器	239
4.6.1	看门狗定时器概述	239
4.6.2	时钟和功率控制	240
4.6.3	看门狗定时器结构	240
4.6.4	看门狗定时器的配置	241
4.6.5	看门狗定时器中断	243
4.6.6	看门狗定时器应用示例	243
4.7	电源管理单元	246
4.7.1	功率控制	246
4.7.2	功率控制相关寄存器	246
4.7.3	电源管理单元及其相关寄存器	247
4.7.4	节电工作模式的配置	248
4.7.5	三种节电模式的比较	251
4.7.6	功率控制注意事项	251
4.7.7	CMSIS 内在函数	252

习题	252
第 5 章 基于 CMSIS 接口标准的软件设计	253
5.1 CMSIS 标准简介	253
5.2 CMSIS 代码规范	255
5.3 CMSIS 文件结构	258
5.4 CMSIS 例子程序	263
5.5 Startup_LPC11xx.s 启动代码	266
5.5.1 启动代码的作用	266
5.5.2 Startup_LPC11xx.s 启动代码分析	266
习题	270
附录	271
附录 A Cortex-M0/M0+指令集	271
A.1 指令集汇总	271
A.2 内部函数	272
附录 B 缩写	273
附录 C 术语	275
参考文献	278

第1章 嵌入式系统概述

本章主要介绍嵌入式系统的定义、分类、历史以及发展趋势，介绍嵌入式操作系统的基本知识和对 ARM Cortex-M 处理器的支持情况，对 NXP 公司 ARM Cortex-M0 系列处理器进行简要介绍，并列出了常用的 ARM Cortex-M 处理器集成开发环境和仿真软件平台。

1.1 嵌入式系统

1.1.1 嵌入式系统的定义

嵌入式系统 (Embedded System) 是一种“完全嵌入受控器件内部，为特定应用而设计的专用计算机系统”，根据英国电气工程师协会 (U.K. Institution of Electrical Engineer) 的定义，嵌入式系统为控制、监视或辅助设备，机器或用于工厂运作的设备。从中可以看出，嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。目前，国内一个普遍认同的定义是以应用为中心，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。

嵌入式系统是相对通用型计算机来讲的，凡是带有微处理器的专用软硬件系统都可以称为嵌入式系统。区别于可以执行多重任务的通用型计算机，嵌入式系统是为某些特定任务而设计的。有些系统必须满足实时性要求，以确保安全性和可用性；另一些系统则对性能要求很低，甚至不要求性能，以简化硬件、降低成本。

1.1.2 嵌入式系统的历史

MIT 仪器研究室的查尔斯·斯塔克·德雷珀开发的阿波罗制导计算机是现代嵌入式系统的雏形之一。在项目初期，它被看成风险最高的部分，原因是为了减小尺寸和重量，它采用了当时最新研发的单片集成电路。

第一款大批量生产的嵌入式设备是 1961 年发布的民兵 I 导弹内的 D-17 自动制导计算机。当民兵 II 导弹在 1966 年开始生产的时候，D-17 升级成一种新型计算机，其中首次大量使用了集成电路。仅这个项目就将与非门模块的单价从 1000 美元降低至 3 美元，低到可用于商业产品。

由于 20 世纪 60 年代的这些早期应用，不仅嵌入式设备的价格降低了，同时处理能力和功能也获得了巨大的提高。以第一款 4 位单片机英特尔 4004 为例，它是为计算器和其他小型系统设计的，但仍然需要外部存储器和外围芯片。1978 年，美国国家工程制造商协会发布了可编程单片机的“标准”，涵盖了几乎所有以计算机为基础的控制器，如单板机、数控设备，以及基于事件的控制器。

随着微控制器和微处理器的价格下降，消费品也可以更换掉基于按钮的模拟器件，如分压计和可变电容，采用微处理器读取开关或按钮信号。

到了20世纪70年代末期,存储器、输入/输出部件集成到处理器内,产生了单片机。在采用通用计算机占用的成本太高昂的应用中,单片机取而代之。

嵌入式系统的出现最初是基于单片机的。单片机的出现,使得汽车、家电、工业机器、通信装置以及成千上万种产品可以通过内嵌电子装置来获得更佳的使用性能:更容易使用、更快、更便宜。这些装置已经初步具备了嵌入式的应用特点,但是这时的应用只是使用8位的芯片,执行一些单线程的程序,还谈不上“系统”的概念。

最早的8位单片机是Intel公司的8048,它出现在1976年。Motorola同时推出了68HC05,Zilog公司推出了Z80系列,这些早期的单片机均含有256B的RAM、4KB的ROM、4个8位并口、1个全双工串行口、2个16位定时器。之后在20世纪80年代初,Intel又进一步完善了8048,在它的基础上研制成功了8051,这在单片机的历史上是值得纪念的一页。迄今为止,51系列仍然是成功的单片机芯片系列,在各种产品中有着非常广泛的应用。

从20世纪80年代早期开始,嵌入式系统的程序员开始用商业级的“操作系统”编写嵌入式应用软件,这使得人们可以获取更短的开发周期、更低的开发资金和更高的开发效率,“嵌入式系统”真正出现了。确切地说,这个时候的操作系统是一个实时核,这个实时核包含了许多传统操作系统的特征,包括任务管理、任务间通信、同步与相互排斥、中断支持、内存管理等功能。其中比较著名的有Ready System公司的VRTX、Integrated System Incorporation (ISI)的PSOS和IMG的VxWorks、QNX公司的QNX等。这些嵌入式操作系统都具有嵌入式的典型特点:均采用占先式的调度,响应的时间很短,任务执行的时间可以确定;系统内核很小,具有可裁剪、可扩充和可移植性,可以移植到各种处理器上;较强的实时性和可靠性,适合嵌入式应用。这些嵌入式实时多任务操作系统的出现,使得应用开发人员得以从小范围的开发解放出来,同时也促使嵌入式有了更为广阔的应用空间。

20世纪90年代以后,随着对实时性要求的提高,软件规模不断上升,实时核逐渐发展为实时多任务操作系统(RTOS),并作为一种软件平台逐步成为目前国际嵌入式系统的主流。这时候更多的公司看到了嵌入式系统的广阔发展前景,开始大力发展自己的嵌入式操作系统。除了上面几家老牌公司的产品以外,还出现了Palm OS、WinCE、嵌入式Linux、Lynx、Nucleux,以及国内的Hopen、Delta Os等嵌入式操作系统。

目前智能手机上应用最广泛的嵌入式操作系统是iOS和Andriod。iOS(iPhone OS)是由苹果公司为移动设备所开发的操作系统,支持的设备包括iPhone、iPod touch、iPad、Apple TV。与Android及Windows Phone不同,iOS不支持非苹果的硬件设备。Android是Google公司的一款以Linux为基础的开放源代码操作系统,该平台由操作系统、中间件、用户界面和应用软件组成,号称是首个为移动终端打造的真正开放和完整的移动软件。Windows Phone是微软发布的一款手机操作系统,诺基亚把Windows Phone作为智能手机的主要操作系统,2014年4月27日,微软完成诺基亚手机业务的并购。

随着嵌入式技术的发展前景日益广阔,会有更多的嵌入式操作系统和应用软件出现。

1.1.3 嵌入式系统的发展趋势

经过几十年的发展,嵌入式系统已经在很大程度上改变了人们的生活、工作和娱乐方式,而且这些改变还在加速。嵌入式系统具有无数的种类,每类都具有自己独特的个性。例如,MP3、数码相机与打印机就有很大的不同。汽车中更是具有多个嵌入式系统,使汽车更轻快、

更干净、更容易驾驶。

即使不可见，嵌入式系统也无处不在。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、运输和航天领域。例如，神州飞船和长征火箭中有很多嵌入式系统，导弹的制导系统也是嵌入式系统，高档汽车中也有多达几十个嵌入式系统。在日常生活中，人们使用各种嵌入式系统，但未必知道它们。图 1-1 就是一些比较新的、生活中比较常见的嵌入式系统。事实上，几乎所有带有一点“智能”的家电（全自动洗衣机、电饭煲等）都是嵌入式系统。嵌入式系统广泛的适应能力和多样性，使得视听、工作场所甚至健身设备中到处都有嵌入式系统。



图 1-1 常见的嵌入式系统实例

物联网技术、工业 4.0、智能手机、智能硬件、可穿戴设备，以及大数据、人工智能等新技术的引领，使得嵌入式系统获得了巨大的发展契机，为嵌入式市场展现了美好的前景，同时也对嵌入式生产厂商提出了新的挑战。ARM 公司不断地完善嵌入式开发的生态系统，开发软件接口标准，使得开发者使用 C 语言等高级语言而不用汇编就可以自由地编写嵌入式应用软件，并且可以利用接口 API 设计自己的嵌入式操作系统，可以利用开源的软硬件设计低成本的开发工具，可以利用 ARM 授权 IP 和 FPGA 进行专用 SoC 设计，这在过去是难以想象的。

1.2 嵌入式处理器

普通个人计算机中的处理器是通用目的的处理器。它们的设计非常丰富，因为这些处理器提供全部的特性和广泛的功能，故可以用于各种应用中。但是通用处理器能源消耗大，产生的热量高，尺寸也大，其复杂性意味着这些处理器的制造成本昂贵。在早期，嵌入式系统通常用通用目的的处理器设计。近年来，随着大量先进的微处理器制造技术的发展，越来越多的嵌入式系统用嵌入式处理器设计，而不是用通用目的的处理器。这些嵌入式处理器是为完成特殊的应用而设计的特殊目的的处理器。

嵌入式处理器是嵌入式系统的核心，是控制、辅助系统运行的硬件单元。其种类范围极其广阔，从最初的 4 位处理器，目前仍在大规模应用的 8 位单片机，处境尴尬的 16 位单片机，到最新的受到广泛青睐的 32 位、64 位嵌入式 CPU，都属于嵌入式处理器。根据性能与工作

方式可以将嵌入式处理器分为以下几个类型：

1. 嵌入式微处理器

嵌入式微处理器 (Embedded Microprocessor Unit, EMPU) 的基础是通用计算机中的 CPU。在应用中, 将微处理器装配在专门设计的电路板上, 只保留和嵌入式应用有关的母板功能, 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、可靠性等方面一般都做了各种增强。和工业控制计算机相比, 嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点, 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, 技术保密性也较差。嵌入式微处理器及其存储器、总线、外设等安装在一块电路板上, 称为单板计算机, 如 STD-BUS、PC104 等。近年来, ARM 公司推出了 ARM9、ARM11 以及后来的 Cortex-A/R 系列嵌入式微处理器, 被广大手机芯片厂商所采用。

嵌入式微处理器主要有 Am186/88、386EX、SC-400、PowerPC、68000、MIPS、ARM9/ARM11/ARM Cortex-A/R 系列等。

2. 嵌入式微控制器

嵌入式微控制器 (Microcontroller Unit, MCU) 又称单片机, 顾名思义, 就是将整个计算机系统集成到一块芯片中。嵌入式微控制器一般以某一种微处理器内核为核心, 芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时器/计数器、看门狗 (WatchDog)、I/O、串行口、脉宽调制输出、A-D、D-A、Flash RAM、EEPROM 等各种必要功能和外设。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和应用需求相匹配, 功能不多不少, 从而减少功耗和成本。和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工业的主流。微控制器的片上外设资源一般比较丰富, 适合于控制, 因此称为微控制器。

嵌入式微控制器目前的品种和数量最多, 比较有代表性的通用系列包括 8051、MCS-96/196/296、MC68HC05/11/12/16、数目众多的 ARM7 和 ARM Cortex-M 系列芯片等。目前 MCU 占嵌入式系统约 70% 的市场份额。

3. 嵌入式 DSP

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 应用正从在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 (Embedded Digital Signal Processor, EDSP)。

嵌入式 DSP 比较有代表性的产品是 Texas Instruments 的 TMS320 系列和 Motorola 的 DSP56000 系列。TMS320 系列处理器包括用于控制的 C2000 系列、移动通信的 C5000 系列, 以及性能更高的 C6000 和 C8000 系列。DSP56000 目前已经发展成为 DSP56000、DSP56100、DSP56200 和 DSP56300 等几个不同系列的处理器。另外, ARM 公司在其 Cortex-M 系列芯片中也集成了 DSP 的特性, 提供免费的 DSP 资源库, 逐渐模糊了 ARM 处理器和 DSP 的界限。

4. 嵌入式片上系统

随着 EDI 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, 这就是片上系统 (System on Chip, SoC)。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器件以外, 整个嵌入式系统大部分均可集成到一块或几块芯片中去, 应用系统电路板将变得很简洁, 对于减小体积和功耗、提高可靠性非常有利。

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。一个有代表性的产品是 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 SoC, 可用于公众互联网 (如 Internet) 安全方面。

1.3 嵌入式操作系统

1.3.1 嵌入式操作系统简介

在计算机技术发展的初期阶段, 计算机系统中没有操作系统这个概念。为了给用户提供一个与计算机之间进行交互的接口, 同时提高计算机的资源利用率, 便出现了计算机监控程序 (Monitor), 使用户能通过监控程序来使用计算机。随着计算机技术的发展, 计算机系统的硬件、软件资源也愈来愈丰富, 监控程序已不能适应计算机应用的要求, 于是在 20 世纪 60 年代中期监控程序又进一步发展形成了操作系统 (Operating System)。发展到现在, 广泛使用的有 3 种操作系统, 即多道批处理操作系统、分时操作系统以及实时操作系统。

多道批处理操作系统一般用于计算中心较大的计算机系统中。由于它的硬件设备比较全、价格较高, 所以此类系统十分注意 CPU 及其他设备的充分利用, 追求高的吞吐量, 不具备实时性。

分时操作系统的主要目的是让多个计算机用户能共享系统的资源, 能及时地响应和服务于联机用户, 只具有很弱的实时功能, 与真正的实时操作系统有明显的区别。

那么, 什么样的操作系统才能称为实时操作系统呢? IEEE 的实时 UNIX 分委会认为实时操作系统应具备以下几点:

- 1) 异步的事件响应。实时系统为能在系统要求的时间内响应异步的外部事件, 要求有异步 I/O 和中断处理能力。I/O 响应时间常受内存访问、盘访问和处理器总线速度所限制。

- 2) 切换时间和中断延迟时间确定。

- 3) 优先级中断和调度。必须允许用户定义中断优先级和被调度的任务优先级并指定如何服务中断。

- 4) 抢占式调度。为保证响应时间, 实时操作系统必须允许高优先级任务一旦准备好运行马上抢占低优先级任务的执行。

- 5) 内存锁定。必须具有将程序或部分程序锁定在内存的能力, 锁定在内存的程序减少了为获取该程序而访问盘的时间, 从而保证了快速的响应时间。

- 6) 连续文件。应提供存取盘上数据的优化方法, 使得存取数据时查找时间最少。通常要求把数据存储在连续文件上。

7) 同步。提供同步和协调共享数据使用和时间执行的手段。

总的来说,实时操作系统是事件驱动的,能对来自外界的作用和信号在限定的时间范围内做出响应。它强调的是实时性、可靠性和灵活性,与实时应用软件相结合成为有机的整体起着核心作用,由它来管理和协调各项工作,为应用软件提供良好的运行环境及开发环境。

从实时系统的应用特点来看,实时操作系统可以分为两种:一般实时操作系统和嵌入式实时操作系统。一般实时操作系统与嵌入式实时操作系统都是具有实时性的操作系统,它们的主要区别在于应用场合和开发过程。

一般实时操作系统应用于实时处理系统的上位机和实时查询系统等实时性较弱的实时系统,并且提供了开发、调试、运用一致的环境。

嵌入式实时操作系统应用于实时性要求高的实时控制系统,而且应用程序的开发过程是通过交叉开发来完成的,即开发环境与运行环境不一致。嵌入式实时操作系统具有规模小(一般在几千字节到几十千字节内)、可固化使用、实时性强(在毫秒或微秒数量级上)的特点。

1.3.2 嵌入式实时操作系统的特点

1. 高精度计时系统

计时精度是影响实时性的一个重要因素。在实时应用系统中,经常需要精确确定实时地操作某个设备或执行某个任务,或精确地计算一个时间函数。这些不仅依赖于一些硬件提供的时钟精度,也依赖于实时操作系统实现的高精度计时功能。

2. 多级中断机制

一个实时应用系统通常需要处理多种外部信息或事件,但处理的紧迫程度有轻重缓急之分,有的必须立即做出反应,有的则可以延后处理。因此,需要建立多级中断嵌套处理机制,以确保对紧迫程度较高的实时事件进行及时响应和处理。

3. 实时调度机制

实时操作系统不仅要及时响应实时事件中断,同时也要及时调度运行实时任务。但是,处理器调度并不能随心所欲地进行,因为涉及两个进程之间的切换,只能在确保“安全切换”的时间点上进行。实时调度机制包括两个方面:一是在调度策略和算法上保证优先调度实时任务;二是建立更多“安全切换”时间点,保证及时调度实时任务。

在嵌入式实时操作系统环境下开发实时应用程序使程序的设计和扩展变得容易,不需要大的改动就可以增加新的功能。通过将应用程序分割成若干独立的任务模块,使应用程序的设计过程大为简化,而且对实时性要求苛刻的事件都得到了快速、可靠地处理。通过有效的系统服务,嵌入式实时操作系统使得系统资源得到了更好的利用。

但是,使用嵌入式实时操作系统还需要额外的 ROM/RAM 开销、2%~5%的 CPU 额外负荷,以及内核的费用。

1.3.3 常用的嵌入式操作系统

1. μ Clinux

嵌入式 Linux 操作系统 μ Clinux 是一个完全符合 GNU/GPL 公约的操作系统,完全开放代码。 μ Clinux 的名字来自于希腊字母“ μ ”和英文大写字母“C”的结合,“ μ ”代表“微小”之意,“C”代表“控制器”,所以从字面上就可以看出它的含义,即“微控制领域中的 Linux”。

系统”。

为了降低硬件成本及运行功耗，很多嵌入式 CPU 没有设计内存管理单元（Memory Management Unit, MMU）功能模块。最初，运行于这类没有 MMU 的 CPU 之上的都是一些很简单的单任务操作系统，或者更简单的控制程序，甚至根本就没有操作系统而直接运行应用程序。在这种情况下，系统无法运行复杂的应用程序，或者效率很低，而且所有的应用程序需要重写，并要求程序员十分了解硬件特性。这些都阻碍了应用于这类 CPU 之上的嵌入式产品开发的进度。

μ Clinux 从 Linux 2.0/2.4 内核派生而来，沿袭了主流 Linux 的绝大部分特性。它专门针对没有 MMU 的 CPU，并且为嵌入式系统做了许多小型化的工作，适用于没有虚拟内存或 MMU 的处理器，如 ARM7TDMI。它通常用于具有很少内存或 Flash 的嵌入式系统。 μ Clinux 是为了支持没有 MMU 的处理器而对标准 Linux 做出的修正。它保留了操作系统的所有特性，为硬件平台更好地运行各种程序提供了保证。在 GNU 通用公共许可证（GNU GPL）的保证下，运行 μ Clinux 操作系统的用户可以使用几乎所有的 Linux API 函数，不会因为缺少 MMU 而受到影响。由于 μ Clinux 在标准的 Linux 基础上进行了适当的裁剪和优化，形成了一个高度优化的、代码紧凑的嵌入式 Linux。虽然它的体积很小，但仍然保留了 Linux 的大多数的优点：稳定、良好的移植性、优秀的网络功能、完备的对各种文件系统的支持以及标准丰富的 API 等。

2. Windows CE

Windows CE 是微软开发的一个开放的、可升级的 32 位嵌入式操作系统，是基于掌上型电脑类的电子设备操作系统，它是精简的 Windows 95。Windows CE 的图形用户界面相当出色，其中 CE 中的 C 代表 Compact、Consumer、Connectivity 和 Companion，E 代表 Electronics。与 Windows 95/98、Windows NT 不同的是，Windows CE 是所有源代码全部由微软自行开发的嵌入式新型操作系统，其操作界面虽来源于 Windows 95/98，但 Windows CE 是基于 Win32 API 重新开发的、新型的信息设备平台。Windows CE 具有模块化、结构化和基于 Win32 应用程序接口以及与处理器无关等特点。Windows CE 不仅继承了传统的 Windows 图形界面，并且在 Windows CE 平台上可以使用 Windows 95/98 上的编程工具（如 Visual Basic、Visual C++ 等）、使用同样的函数、使用同样的界面网格，使绝大多数的应用软件只需简单地修改和移植就可以在 Windows CE 平台上继续使用。

3. VxWorks

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统（RTOS），是嵌入式开发环境的关键组成部分。由于其良好的持续发展能力、高性能的内核以及友好的用户开发环境，在嵌入式实时操作系统领域占据一席之地。VxWorks 以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通信、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18 战斗机以及 B-2 隐形轰炸机和爱国者导弹上，甚至连 1997 年 4 月在火星表面登陆的火星探测器上也使用到了 VxWorks。

4. OSE

OSE 主要是由 ENEA Data AB 下属的 ENEA OSE Systems AB 负责开发和技术服务的，一直以来都充当着实时操作系统以及分布式和容错性应用的先锋。公司建立于 1968 年，由大约 600 名雇员专门从事实时应用的技术支持工作。ENEA OSE Systems AB 是现今市场上一个飞