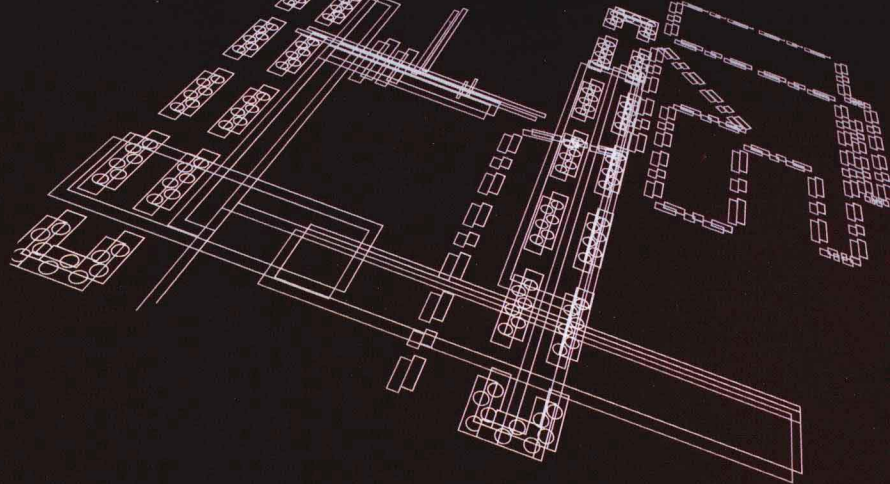




嵌入式 系统基础

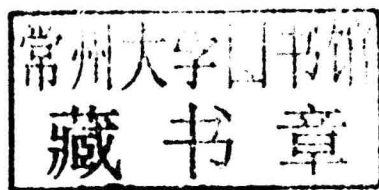
邓力 钟国文 主编

QR SYSTEM

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

嵌入式系统基础

主 编 邓 力 钟国文
副主编 梁广瑞 谭 莉 杨 佳



 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

图书在版编目 (CIP) 数据

嵌入式系统基础/邓力, 钟国文主编. —北京: 北京理工大学出版社, 2013. 3
ISBN 978 - 7 - 5640 - 6732 - 8

I. ①嵌… II. ①邓…②钟… 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 (2012) 第 199799 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (办公室) 68944990 (批销中心) 68911084 (读者服务部)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京慧美印刷有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 16

字 数 / 371 千字

版 次 / 2013 年 3 月第 1 版 2013 年 3 月第 1 次印刷

责任编辑 / 陈莉华

印 数 / 1 ~ 1500 册

责任校对 / 周瑞红

定 价 / 45.00 元

责任印制 / 王美丽

Foreword 前言

Foreword

21 世纪是嵌入式计算机系统时代,又称“后 PC 时代”。嵌入式系统已广泛应用到了信息家电、通信设备、仪器仪表、军事装备、船舶等众多领域中,市场对嵌入式系统人员的需求日趋上升。近年来,市场急需专业化的嵌入式系统开发特别是嵌入式软件开发人才,加强这方面的人才培养尤为重要。

ARM 技术已在许多领域得到广泛的应用,作为 32 位的 RSIC 架构,ARM 芯片不但占据了高端的微控制器市场,同时也逐渐向低端控制器应用领域拓展。在通信产品领域,智能手机大部分的处理器都是采用 ARM 核。

嵌入式技术往往和行业背景结合紧密,由于嵌入式系统技术更新较快,难以找到适用的系列教材。本书以嵌入式系统硬件以及嵌入式实时操作系统为重点,全面介绍嵌入式系统概况、ARM 体系结构、ARM 的指令系统、LPC2000 系列 ARM 的硬件结构、基于源代码开放的 $\mu\text{C}/\text{OS} - \text{II}$ 程序设计、嵌入式系统的设计与开发。本书是为了克服传统教学体系中教与学互相脱离,以嵌入式系统的知识模块与工程实训项目相结合来驱动学生的工程实践能力培养,对每个知识点采用了“项目开发实例”的方式来编写,并通过 Proteus 软件的电路仿真完成项目的电路设计。

本书利用学生学过单片机课程的条件,将 LPC2000 系列首先当成高级单片机来使用,使学生对嵌入式知识的学习有更好的连贯性;将 $\mu\text{C}/\text{OS} - \text{II}$ 的内核源代码作为教学案例,讲解嵌入式操作系统内核的实现机制和原理,同时将前面 LPC2000 系列无操作系统的项目通过 $\mu\text{C}/\text{OS}$ 操作系统的任务来实现,使学生明白操作系统的特点。

本书由唐秋玲主审,由邓力、钟国文担任主编,梁广瑞、谭莉、杨佳担任副主编。其中,前言、第 1 章、第 2 章、第 3 章的 3.1~3.4 节、3.5 节中的 3.5.2、第 4 章、第 5 章的 5.1~5.4 节、5.11~5.13 节、第 6 章这几部分的内容由邓力编写,第 5 章的 5.6、5.7 节的内容由钟国文编写,第 5 章的 5.8~5.10 节的内容由梁广瑞编写,第 5 章的 5.5 节内容由谭莉编写,第 3 章 3.5 节的 3.5.1 内容由杨佳编写。全书由邓力、杨佳统稿。本书得到了广州市风标信息技术有限公司、广西火炬高技术发展公司苏鸿高级工程师的支持,在此表示感谢!

由于编者经验不足,本书难免有不足之处,恳请各位读者批评指正。

编 者

第1章 嵌入式系统的概况	1
1.1 嵌入式系统的定义和组成	1
1.2 嵌入式微处理器体系结构	3
1.2.1 ARM 体系的硬件架构	3
1.2.2 冯·诺依曼体系结构和哈佛体系结构	6
1.2.3 RISC 体系结构	7
1.2.4 流水线技术	8
1.3 嵌入式操作系统	9
1.3.1 嵌入式操作系统的特点	9
1.3.2 嵌入式操作系统的选择	9
1.3.3 实时操作系统	10
1.3.4 目前市场上流行的嵌入式操作系统	11
习题	12
第2章 ADS 项目开发及 AXD 调试	13
2.1 ADS1.2 集成开发环境的安装及使用	13
2.2 ADS 项目及文件的建立	15
2.3 在工程中添加源文件	18
2.4 对工程进行编译和连接	19
2.5 AXD 调试器的使用	21
习题	23
第3章 ARM 微处理器的指令系统	24
3.1 ARM 微处理器的指令集概述	24
3.2 ARM 指令的寻址方式	24
3.2.1 立即寻址	24
3.2.2 寄存器寻址	24
3.2.3 寄存器间接寻址	25
3.2.4 基址变址寻址	25
3.2.5 多寄存器寻址	25
3.2.6 相对寻址	26
3.2.7 堆栈寻址	26

3.2.8 块拷贝寻址	26
3.3 ARM 指令集	27
3.3.1 跳转指令	27
3.3.2 数据处理指令	28
3.3.3 乘法指令与乘加指令	33
3.3.4 程序状态寄存器访问指令	35
3.3.5 加载/存储指令	35
3.3.6 批量数据加载/存储指令	37
3.3.7 数据交换指令	38
3.3.8 移位指令(操作)	39
3.3.9 协处理器指令	40
3.3.10 异常产生指令	42
3.3.11 ARM 伪指令	42
3.4 ADS1.2 集成开发环境汇编语言项目实训	53
3.4.1 ARM 指令的立即寻址	53
3.4.2 ARM 指令的寄存器寻址	54
3.4.3 ARM 的寄存器偏移寻址	55
3.4.4 ARM 的寄存器间接寻址	57
3.4.5 验证 ARM 的基址变址寻址方式	59
3.4.6 验证 ARM 的堆栈寻址方式	61
3.4.7 汇编加法运算	63
3.5 ARM 的 C 语言程序设计	63
3.5.1 C 语言的基础知识	63
3.5.2 ADS1.2 的 C 程序设计	105
小结	105
习题	105
第 4 章 Proteus 软件简介及应用	106
4.1 Proteus 电路原理的设计	106
4.2 Proteus 使用过程中的一些常见问题	112
第 5 章 LPC2106 嵌入式微处理器硬件结构	116
5.1 LPC2000 系列简介	116
5.2 通用输入/输出端口 GPIO 相关寄存器描述	118
5.3 GPIO 相关控制寄存器	119
5.4 GPIO 项目实例	121
5.4.1 LED 灯闪烁 Proteus 电路设计	121
5.4.2 LPC2106 ADS 项目的建立	122
5.4.3 LED 灯闪烁流程图、ADS 项目主程序及 Proteus 电路仿真	145

5.4.4	GPIO 项目二: 1602 液晶 ADS 项目程序代码及 Proteus 电路仿真	148
5.4.5	GPIO 项目二的扩展	154
5.5	向量中断控制器	155
5.5.1	寄存器描述	155
5.5.2	向量中断控制器项目	158
5.6	定时器	161
5.6.1	寄存器描述	161
5.6.2	定时器项目	165
5.7	UART	168
5.7.1	UART 寄存器描述	168
5.7.2	串口通信项目代码编写、编译及 Proteus 电路仿真	173
5.8	脉宽调制(PWM)	176
5.8.1	特性	177
5.8.2	描述	177
5.8.3	PWM 功能寄存器设置流程	184
5.8.4	PWM 项目实例	184
5.9	模数转换器(ADC)	186
5.9.1	特性	187
5.9.2	描述	187
5.9.3	管脚描述	187
5.9.4	寄存器描述	188
5.9.5	A/D 项目实例	193
5.10	实时时钟(RTC)	197
5.10.1	特性	197
5.10.2	描述	197
5.10.3	RTC 寄存器描述	197
5.10.4	RTC 应用实例	204
5.11	SPI 接口	208
5.11.1	SPI 简介	208
5.11.2	SPI 描述	208
5.11.3	SPI 寄存器描述	210
5.11.4	SPI 应用示例	213
5.12	I ² C 接口	215
5.12.1	I ² C 简介	215
5.12.2	I ² C 描述	216
5.12.3	寄存器描述	217
5.12.4	I ² C 应用示例	220
5.13	看门狗	227
5.13.1	看门狗简介	227

5.13.2 看门狗内部结构	227
5.13.3 看门狗寄存器描述	228
5.13.4 使用示例	229
习题	231
第6章 基于 μC/OS 的程序开发	232
6.1 了解 μ C/OS-II 内核的任务管理	232
6.2 μ C/OS-II 在 LPC2106 平台的移植	238
6.2.1 编写与编译器相关的数据类型及与 ARM 处理器相关的代码(OS_CPU.H 的移植)	238
6.2.2 用 C 语言编写 6 个操作系统相关的函数(OS_CPU_C.C 的移植)	240
6.2.3 用汇编语言编写 4 个与处理器相关的函数(OS_CPU.ASM 的移植)	241
6.3 基于 μ C/OS-II 的 LCD 显示项目的开发	243
习题	246
参考文献	247

第 1 章

嵌入式系统的概况

1.1 嵌入式系统的定义和组成

(一) 嵌入式系统的应用领域和定义

嵌入式技术无处不在,各种使用嵌入式技术的电子产品有 MP3、PDA、手机、智能玩具、网络家电、智能家电、车载电子设备等。

在工业和服务领域中,大量嵌入式技术也已经应用于工业控制、数控机床、智能工具、工业机器人、服务机器人等各个行业,正在逐渐改变着传统的工业生产和服务方式。

(二) 嵌入式系统的定义

1. IEEE 给出的定义

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”(原文为 devices used to control, monitor, or assist the operation of equipment, machinery or plants)。

2. 国内普遍认同的定义

嵌入式系统是以应用为中心,以计算机技术为基础,软、硬件可裁剪,适应应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

可以从以下几个方面来理解国内对嵌入式系统的定义:

(1) 嵌入式系统是面向用户、面向产品、面向应用的,它必须与具体应用相结合才能具有生命力。

(2) 嵌入式系统必须能够根据应用需求对软硬件进行裁剪,满足应用系统的功能、可靠性、成本、体积的要求。

(三) 嵌入式系统的几个重要特征

1. 系统内核小

由于嵌入式系统一般是应用于小型电子装置的,系统资源相对有限,所以内核较之传统的操作系统要小得多。

2. 专用性强

嵌入式系统的个性化很强,其中的软件系统和硬件的结合非常紧密,一般要针对硬件进行系统的移植。

即使在同一品牌、同一系列的产品中也需要根据系统硬件的变化和增减不断进行修改。

3. 系统精简

嵌入式系统一般没有系统软件和应用软件的明显区分,不要求其功能设计及实现上过于复杂,这样一方面利于控制系统成本,同时也利于实现系统安全。

4. 高实时性

这是嵌入式软件的基本要求,而且软件要求固化存储,以提高速度。软件代码要求高质量和高可靠性、实时性。

5. 嵌入式系统的开发

嵌入式系统开发需要开发工具和开发环境(交叉开发环境)。

(四) 嵌入式技术的发展趋势

嵌入式技术将成为“后 PC 时代”的主宰:

(1) 嵌入式技术成为当前微电子技术 with 计算机技术中的一个重要分支。

(2) 使计算机的分类从以前的巨型机、大型机、小型机、微机之分变为了通用计算机和嵌入式系统之分。

(3) 嵌入式的应用更是涉及金融、航天、电信、网络、信息家电、医疗、工业控制、军事等各个领域,以致一些学者断言嵌入式技术将成为“后 PC 时代”的主宰。

(五) 嵌入式处理器的分类

1. 嵌入式 MPU

嵌入式 MPU(Micro - Processor Unit)就是将整个计算机系统的主要硬件集成到一块芯片中,芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、Watchdog、I/O、串行口等各种必要功能和外设。

(1) 嵌入式 MPU 的特点。

1) 其功能和微处理器基本一样,是具有 32 位以上的处理器,具有较高的性能。

2) 具有体积小、功耗少、成本低、可靠性高的特点。有的可提供工业级应用。

(2) 流行的嵌入式 MPU。

1) 通用系列:8051、MCS251、MCS - 96/196/296 等。

2) 半通用系列:支持 I²C、CAN BUS 及众多专用 MCU 和兼容系列。

2. 嵌入式 DSP

嵌入式 DSP(Digital Signal Processing)是专门用于信号处理方面的处理器,其在系统结构和指令算法方面进行了特殊设计,具有很高的编译效率和指令执行速度。

应用领域:数字滤波、频谱分析、FFT。

流行的嵌入式 DSP:TMS320C2000 系列(TI)、MCS - 296(Intel)等。

3. 嵌入式 SoC

SoC(System on Chip,片上系统)是采用 ASIC(Application Specific Integrated Circuits)设计方法学中的新技术,以嵌入式系统为核心,以 IP 复用技术为基础,集软、硬件于一体,并追求产品系统最大包容的集成芯片。它是指在一个芯片上实现信号采集、转换、存储、处理和 I/O 等功能,包含嵌入软件及整个系统的全部内容。SoC 的设计基础是 IP(Intellectual Property)复用技术。为了加快 SoC 芯片设计的速度,人们将已有的 IC 电路以模块的形式在 SoC 芯片设计

中调用,从而简化芯片的设计,缩短设计时间,提高设计效率。这些可以被重复使用的 IC 模块就叫作 IP 模块(或者叫作系统宏单元、芯核、虚拟器件)。IP 模块是一种预先设计好,已经过验证,具有某种确定功能的集成电路、器件或部件。它具有 3 种不同形式:软 IP 核(soft IP core)、固 IP 核(firm IP core)和硬 IP 核(hard IP core)。嵌入式 SoC 是追求产品系统最大包容的集成器件。

(1) SoC 具有以下几方面的特点。

- 1) 电量小:由于 SoC 产品多采用内部信号的传输,可以大幅降低功耗。
- 2) 体积小:数颗 IC 整合为一颗 SoC 后,可有效缩小电路板上占用的面积。
- 3) 系统功能丰富:在相同的内部空间内,SoC 可整合更多的功能元件和组件。
- 4) 性效比高:随着芯片内部信号传递距离的缩短,信号的传输效率将提升,而使产品性能有所提高。

5) 节省成本:理论上,IP 模块的出现可以减少研发成本,缩短研发时间,可适度节省成本。

(2) 流行的 SoC:Smart XA (Philips)。

4. 嵌入式 SoPC

SoPC(System-on-a-Programmable-Chip),即可编程片上系统。用可编程逻辑技术把整个系统放到一块硅片上,称作 SoPC。可编程片上系统是一种特殊的嵌入式系统:首先它是片上系统(SoC),即由单个芯片完成整个系统的主要逻辑功能;其次,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的功能。SoPC 结合了 SoC 和 PLD、FPGA 各自的优点,一般具备以下基本特征:

- (1) 至少包含一个嵌入式处理器内核。
- (2) 具有小容量片内高速 RAM 资源。
- (3) 丰富的 IP Core 资源可供选择。
- (4) 足够的片上可编程逻辑资源。
- (5) 处理器调试接口和 FPGA 编程接口。
- (6) 可能包含部分可编程模拟电路。
- (7) 单芯片、低功耗、微封装。

1.2 嵌入式微处理器体系结构

1.2.1 ARM 体系的硬件架构

ARM 是 Advanced RISC Machines 的缩写,它是一家微处理器行业的知名企业,该企业设计了大量高性能、廉价、耗能低的 RISC (精简指令集计算机)处理器。公司的特点是只设计芯片,而不生产。它提供 ARM 技术知识产权(IP)核,将技术授权给世界上许多著名的半导体、软件和 OEM 厂商,并提供服务。有 ARM7/ARM9 等多个版本。除了一些 UNIX 图形工作站外,大多数 ARM 核心的处理器都使用在嵌入领域。

ARM,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通称,还可以认为是一种技术的名字。到目前为止,基于 ARM 技术的微处理器应用占据了 32 位嵌入式微处理

器约75%以上的市场份额。全球80%的GSM/3G手机、99%的CDMA手机以及绝大多数PDA产品均采用ARM体系的嵌入式处理器。

ARM微处理器的应用领域及特点如下。

(1) ARM处理器市场覆盖率最高、发展趋势广阔。

基于ARM技术的32位微处理器,市场的占有率目前已达到80%。

绝大多数IC制造商都推出了自己的ARM结构芯片。我国的中兴集成电路、大唐电信、中芯国际和上海华虹,以及国外的一些公司如德州仪器、意法半导体、Philips、Intel、Samsung等都推出了自己设计的基于ARM核的处理器。

1) 应用一: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。

2) 应用二: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。

3) 应用三:网络设备。

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进行了优化,并获得广泛支持,也对DSP的应用领域提出了挑战。

4) 应用四: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。

5) 应用五: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。手机中的32位SIM智能卡也采用了ARM技术。

(2) ARM处理器的特点。

1) 体积小、功耗低、成本低、性能高。

2) 支持Thumb(16位)/ARM(32位)双指令集,能很好地兼容8位/16位器件。

3) 大量使用寄存器,指令执行速度更快。

4) 大多数数据操作都在寄存器中完成。

5) 寻址方式灵活简单,执行效率高。

6) 指令长度固定。

(一) 通用寄存器和程序计数器

ARM微处理器支持7种运行模式,分别为:

(1) 用户模式(usr):ARM处理器正常的程序执行状态。

(2) 快速中断模式(fiq):用于高速数据传输或通道管理。

(3) 外部中断模式(irq):用于通用的中断处理。

(4) 管理模式(svc):操作系统使用的保护模式。

(5) 数据访问终止模式(abt):当数据或指令预取终止时进入该模式,用于虚拟存储及存储保护。

(6) 系统模式(sys):运行具有特权的操作系统任务。

(7) 未定义指令中止模式(und):当未定义指令执行时进入该模式,可用于支持硬件协处理器的软件仿真。

ARM 体系结构的存储器格式有以下两种:

(1) 大端格式:字数据的高字节存储在低地址中,字数据的低字节存放在高地址中。

(2) 小端格式:与大端存储格式相反,高地址存放数据的高字节,低地址存放数据的低字节。

ARM 处理器共有 37 个寄存器,其中包括:

(1) 31 个通用寄存器,包括程序计数器(PC)在内,都是 32 位寄存器。

(2) 6 个状态寄存器,都是 32 位寄存器,但目前只使用了其中 12 位。

通用寄存器可以分为 3 类:未备份寄存器(R0 ~ R7)、备份寄存器(R8 ~ R14)和程序计数器 PC(R15)。对于每一个未备份寄存器来说,在所有的处理器模式下指的都是同一个物理寄存器。对应备份寄存器 R8 ~ R12 来说,每个寄存器对应两个不同的物理寄存器,这使得中断处理非常简单。例如,仅使用 R8 ~ R14 寄存器时,FIQ 处理程序可以不必执行保存和恢复中断现场的指令,从而使中断处理过程非常迅速。对于备份寄存器 R13 和 R14 来说,每个寄存器对应 6 个不同的物理寄存器,其中的一个是用户模式和系统模式共用的,另外的 5 个对应于其他 5 种处理器模式。

(二) ARM 程序状态寄存器

在所有处理器模式下都可以访问当前的程序状态寄存器 CPSR。CPSR 包含条件码标志、中断禁止位、当前处理器模式以及其他状态和控制信息。每种异常模式都有一个程序状态保存寄存器 SPSR。当异常出现时,SPSR 用于保存 CPSR 的状态。

CPSR 和 SPSR 的格式如表 1-1 所示。

表 1-1 CPSR 和 SPSR 的格式

31	30	29	28	27	26 ~ 8	7	6	5	4	3	2	1	0
N	Z	C	V	Q	DNM (RAZ)	I	F	T	M	M	M	M	M

(1) 条件码标志。

N,Z,C,V 大多数指令可以检测这些条件码标志以决定程序指令如何执行。

(2) 控制位。

最低 8 位 I、F、T 和 M 位用作控制位,当异常出现时可改变控制位。当处理器在特权模式下时也可以由软件改变。

中断禁止位:I 置“1”则禁止 IRQ 中断;F 置“1”则禁止 FIQ 中断。

T 位:T=0 指示 ARM 执行;T=1 指示 Thumb 执行。在这些体系结构系统中,可自由地使用能在 ARM 和 Thumb 状态之间切换的指令。

模式位:M0、M1、M2、M3 和 M4(M[4:0])是模式位,这些位决定处理器的工作模式,如表 1-2 所示。

表 1-2 ARM 工作模式 M[4:0]

M[4:0]	模式	可访问的寄存器
0b10000	用户	PC, R14 ~ R0, CPSR
0b10001	FIQ	PC, R14_fiq ~ R8_fiq, R7 ~ R0, CPSR, SPSR_fiq
0b10010	IRQ	PC, R14_irq ~ R8_irq, R12 ~ R0, CPSR, SPSR_irq
0b10011	管理	PC, R14_svc ~ R8_svc, R12 ~ R0, CPSR, SPSR_svc
0b10111	中止	PC, R14_abt ~ R8_abt, R12 ~ R0, CPSR, SPSR_abt
0b11011	未定义	PC, R14_und ~ R8_und, R12 ~ R0, CPSR, SPSR_und
0b11111	系统	PC, R14 ~ R0, CPSR

(3) 其他位。

程序状态寄存器的其他位保留,用作以后的扩展。

1.2.2 冯·诺依曼体系结构和哈佛体系结构

(一) 冯·诺依曼体系结构

冯·诺依曼体系结构模型如图 1-1 所示。

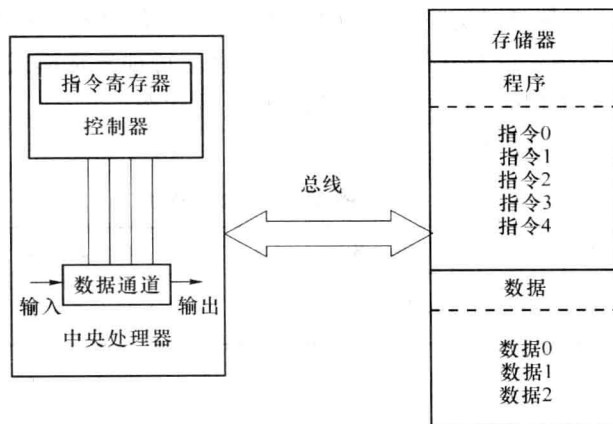


图 1-1 冯·诺依曼体系结构模型

计算系统由一个中央处理单元(CPU)和一个存储器组成。存储器拥有数据和指令,并且可以根据所给的地址对它进行读或写。因此程序指令和数据的宽度相同,如 Intel 8086、ARM7、MIPS 处理器等。

指令的执行周期 T 包括以下几项:

- (1) 取指令(Instruction Fetch): TF 。
- (2) 指令译码(Instruction Decode): TD 。
- (3) 执行指令(Instruction Execute): TE 。
- (4) 存储(Storage): TS 。

每条指令的执行周期: $T = TF + TD + TE + TS$ 。

冯·诺依曼体系结构的特点如下:

- (1) 数据与指令都存储在同一个存储区中,取指令与取数据利用同一数据总线。
- (2) 被早期大多数计算机所采用。
- (3) 结构简单,但速度较慢,因为取指令与取数据不能同时进行。

ARM7 使用冯·诺依曼体系结构。

(二) 哈佛体系结构

哈佛机:为数据和程序提供了各自独立的存储器。

程序计数器只指向程序存储器而不指向数据存储器,这样做的后果是很难在哈佛机上编写出一个自修改的程序。

独立的程序存储器和数据存储器为数字信号处理提供了较高的性能。

指令和数据可以有不同的数据宽度,具有较高的效率,如摩托罗拉公司的 MC68 系列、Zilog 公司的 Z8 系列、ARM10 系列等。

哈佛体系结构模型如图 1-2 所示。

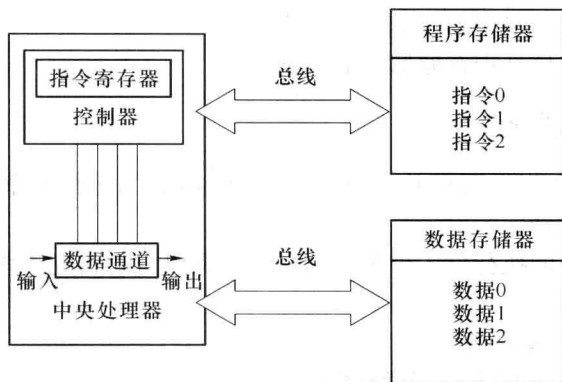


图 1-2 哈佛体系结构模型

哈佛体系结构的特点如下:

- (1) 程序存储器与数据存储器分开。
- (2) 提供了较大的存储器带宽,各自有自己的总线。
- (3) 适合于数字信号处理。
- (4) 大多数 DSP 都是哈佛结构。
- (5) 取指令和取数据在同一周期进行,提高速度。

改进的哈佛体系结构分成 3 个存储区:程序、数据、程序和数据共用。ARM9 使用哈佛体系结构。

1.2.3 RISC 体系结构

CISC(Complex Instruction Set Computer,复杂指令集计算机)具有以下特点:

- (1) 具有大量的指令和寻址方式。
- (2) 8/2 原则:80% 的程序只使用 20% 的指令。
- (3) 大多数程序只使用少量的指令就能够运行。
- (4) CISC CPU 包含有丰富的单元电路,因而功能强、面积大、功耗大。

RISC(Reduced Instruction Set Computer,精简指令集计算机)具有以下特点:

- (1) 在通道中只包含最有用的指令,只提供简单的操作。
- (2) 确保数据通道快速执行每一条指令。
- (3) LOAD-STORE 结构——处理器只处理寄存器中的数据,LOAD-STORE 指令用来完成数据在寄存器和外部存储器之间的传送。
- (4) 使 CPU 硬件结构设计变得更为简单,RISC CPU 包含较少的单元电路,因而面积小、功耗低。

CISC 与 RISC 的主要区别:

(1) 寄存器。

1) RISC 指令集:拥有更多的通用寄存器,每个可以存放数据和地址,寄存器为所有的数据操作提供快速的存储访问。

2) CISC 指令集:多用于特定目的的专用寄存器。

(2) LOAD-STORE 结构。

1) RISC 结构:CPU 仅处理寄存器中的数据,采用独立的、专用的 LOAD-STORE 指令来完成数据在寄存器和外存之间的传送。访存较费时,处理和存储分开,可以反复地使用保存在寄存器中的数据,而避免多次访问外存。

2) CISC 结构:能直接处理存储器中的数据。

1.2.4 流水线技术

流水线技术:即几个指令可以并行执行,可提高 CPU 的运行效率。ARM7 使用 3 级流水线,ARM9 使用 5 级流水线。

3 级流水线,即指令的执行分 3 个阶段:取指令、译码、执行,允许多个操作同时处理,比逐条指令执行要快。

取指令:从存储器中读入指令。

译码:对指令及其用到的寄存器做解码。

执行:从寄存器送出信息,执行移位运算和逻辑运算,并将最后结果写回寄存器。1 条指令需要 3 个时钟周期来完成,流水线使得平均每个时钟周期能完成 1 条指令。

例 1 ADD r1 r2

SUB r3 r2

CMP r1 r3

流水线指令序列如图 1-3 所示。

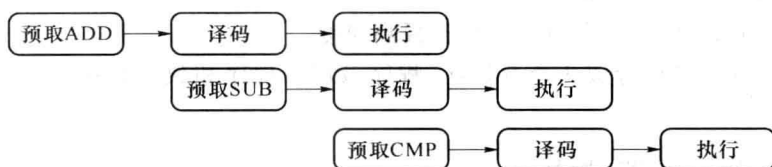


图 1-3 3 级流水线路序指令时序(1)

可以看出,流水线使得每个时钟周期都可以执行一条指令。

例2 (见图1-4)

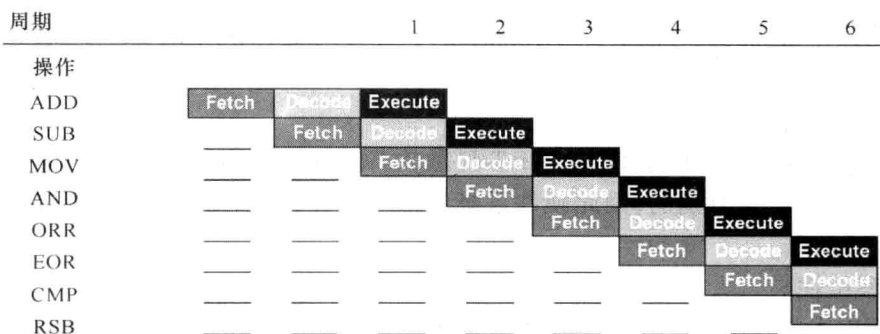


图1-4 3级流水线程序指令时序(2)

该例2中用6个时钟周期执行了6条指令,所有的操作都在寄存器中(单周期执行),指令周期数(CPI)=1。

1.3 嵌入式操作系统

嵌入式操作系统(Embedded Operating System, EOS)是一种用途广泛的系统软件,过去它主要应用于工业控制和国防系统领域。EOS负责嵌入式系统的全部软、硬件资源的分配、任务调度、控制、协调并发活动。它必须体现其所在系统的特征,能够通过装卸某些模块来实现系统所要求的功能。EOS是相对于一般操作系统而言的,它具有一般操作系统最基本的功能,如任务调度、同步机制、中断处理、文件处理等。

1.3.1 嵌入式操作系统的特点

嵌入式操作系统除具备了一般操作系统最基本的任务调度、同步机制、中断处理、文件处理等功能外,还具有以下特点:

- (1) 实时性。
- (2) 支持开放性和可伸缩性的体系结构,具有可裁剪性。
- (3) 提供统一的设备驱动接口。
- (4) 提供操作方便、简单、友好的图形 GUI(图形用户界面)。
- (5) 支持 TCP/IP 协议及其他协议,提供 TCP/UDP/IP/PPP 协议支持及统一的 MAC 访问层接口,提供强大的网络功能。
- (6) 嵌入式操作系统的用户接口通过系统的调用命令向用户程序提供服务。
- (7) 嵌入式系统一旦开始运行就不需要用户过多的干预。
- (8) 嵌入式操作系统和应用软件被固化在嵌入式系统计算机的 ROM 中。
- (9) 具有良好的硬件适应性(可移植性)。

1.3.2 嵌入式操作系统的选择

这些操作系统的出现,使得在嵌入式系统设计时有了很大的选择余地,但选择的操作系统是否恰当对整个系统的开发有着至为关键的影响。总的来说,嵌入式操作系统的选择可以遵