



全国高等学校计算机教育研究会教材建设立项项目

微机原理与接口技术

——嵌入式系统描述

姚琳 万亚东 汪红兵 编著

清华大学出版社



微机原理与接口技术

——嵌入式系统描述

姚琳 万亚东 汪红兵 编著



清华大学出版社
北京

内 容 简 介

本书内容全面、重点明确、表述简洁,注重将微机接口控制器的基本原理和实际操作相结合,突出软硬件设计中的计算思维模式。全书共 12 章,内容包括微机原理及基本概念、Cortex-M3 处理器体系结构、ARM 汇编、嵌入式系统开发基础、GPIO 控制器、NVIC 及 EXTI 中断控制器、定时器、USART 总线、IIC 总线、SPI 总线、ADC 以及低功耗控制,并配套基于 STM32L15x 系列的实验教程。

本书适合作为非计算机专业微机原理及接口技术的教材,也可作为计算机类嵌入式系统课程的参考教材。

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

微机原理与接口技术:嵌入式系统描述/姚琳,万亚东,汪红兵编著. —北京:清华大学出版社,2019
ISBN 978-7-302-52859-3

I. ①微… II. ①姚… ②万… ③汪… III. ①微型计算机—理论 ②微型计算机—接口技术
IV. ①TP36

中国版本图书馆 CIP 数据核字(2019)第 082374 号

责任编辑:谢 琛

封面设计:常雪影

责任校对:焦丽丽

责任印制:沈 露

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>,010-62795954

印 装 者:三河市君旺印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:25.5 字 数:589 千字

版 次:2019 年 8 月第 1 版 印 次:2019 年 8 月第 1 次印刷

定 价:69.00 元

产品编号:083063-01

前言

随着智能制造、物联网、大数据技术的推进和应用,以及新工科建设的需求,数据的采集和感知成为这些技术应用不可或缺的重要环节,各种物联网大赛、创新创业大赛都对软硬件系统设计能力提出了很高的要求,需要学生具有数据感知、处理、传输和分析的综合能力;此外,随着计算思维在计算机基础教学方面的不断推进,思维能力培养已成为教育教学界的共识,计算机硬件系统结构中包含大量计算思维的知识点,如 RISC、CISC、哈佛体系结构、Cache 分层存储、中断处理及优化机制、流水线、串行并行总线技术等,是计算思维培养非常有效的一门课程。微机原理与接口技术是非计算机专业计算机硬件教育的重要课程,本教材以嵌入式系统为对象,对微机的基本原理、ARM 微处理器的接口技术进行梳理,结合大量实验培养学生计算机硬件素养和计算思维能力,提高学生在计算机软硬件系统设计、调试和创新方面的能力,适用于本科非计算机专业学生。

本书选用 Cortex-M3 处理器内核的 STM32L152 系列低功耗微控制器对 ARM 嵌入式系统的体系结构进行讲述,教材以计算机硬件体系涉及的计算思维为主线,第 1 章阐述微型计算机的基本概念、内部架构和嵌入式系统概念;第 2 章以 ARM Cortex-M3 的处理器工作模式、流水、中断等为案例具体阐述硬件设计的方法;第 3 章介绍汇编指令编码、寻址技术并对启动代码进行了分析;第 4 章简述了嵌入式开发流程及 C 语言基础;第 5~11 章对常用外围控制器 GPIO、EXTI、Timer、USART、IIC、SPI、ADC 的一般性工作原理、STM32L1 系列处理器的具体实现和特色、寄存器级别和库函数级别两个层次的程序设计方法进行了详细阐述;第 12 章对低功耗设计进行了介绍。教材内容兼顾嵌入式处理器及外围控制器原理讲解和应用程序设计,让读者理解 Cortex-M3 处理器的特性,各种控制器的工作原理及使用方法,理解嵌入式处理器架构。

本教材目标定位为软硬件协同设计思维,而不仅仅是会使用 and 开发嵌入式系统,结合实验设计,让学生必须理解 ARM 架构、外围控制器的工作原理和设计思路,能够进行应用系统设计。

本书适用于工科非计算机专业微机接口技术、嵌入式系统课程,也可作为计算机专业嵌入式开发课程的教材。

作者
2019 年 4 月

目 录

第 1 章 微型计算机与嵌入式系统概论	1
1.1 微型计算机概述	1
1.1.1 微型计算机系统的组成	1
1.1.2 微处理器的发展	3
1.2 微型计算机的基本原理	7
1.2.1 冯·诺依曼体系结构	7
1.2.2 微机的总线	8
1.2.3 哈佛体系结构	14
1.2.4 微处理器的内部结构	15
1.2.5 I/O 接口技术	19
1.2.6 存储器	20
1.2.7 程序的执行过程	24
1.3 嵌入式系统概述	25
1.4 嵌入式系统架构	27
1.5 嵌入式系统的典型应用	28
1.6 典型嵌入式开源硬件和软件系统	31
1.6.1 开源硬件平台	31
1.6.2 嵌入式开源操作系统	33
第 2 章 Cortex-M3 微处理器的体系结构	35
2.1 ARM 微处理器系列介绍	35
2.2 ARM Cortex-M3 体系结构	37
2.2.1 总体架构	37
2.2.2 操作模式	39
2.2.3 寄存器	40
2.2.4 总线	44
2.2.5 存储器	45
2.2.6 中断	50

2.3	STM32L152RET6 微处理器介绍	50
2.4	STM32L152RET6 微处理器的系统结构	51
2.5	STM32L152RET6 微处理器的引脚说明	54
2.6	STM32L152RET6 微处理器的复位和时钟控制	57
2.7	STM32L152RET6 微处理器的存储映射	59
第 3 章	Cortex-M3 处理器的指令系统	63
3.1	Cortex-M3 处理器的指令系统概述	63
3.1.1	指令系统基本概念	63
3.1.2	指令格式	65
3.1.3	寻址方式	67
3.1.4	数据传送指令	68
3.1.5	存储器访问指令	69
3.1.6	算术运算指令	74
3.1.7	逻辑运算指令	77
3.1.8	移位和循环指令	78
3.1.9	比较指令	79
3.1.10	分支控制指令	80
3.1.11	其他指令	81
3.2	ARM 汇编器中的伪指令	83
3.2.1	Thumb 伪指令	83
3.2.2	符号定义伪指令	84
3.2.3	数据定义伪指令	85
3.2.4	汇编控制伪指令	86
3.2.5	其他常用的伪指令	87
3.3	汇编语言的程序结构	88
第 4 章	开发板硬件系统及开发环境	92
4.1	最小系统设计	92
4.2	开发板电路原理图	93
4.2.1	电源	93
4.2.2	复位和启动电路	95
4.2.3	时钟	95
4.2.4	调试接口	96
4.2.5	按键	97
4.2.6	LED 灯	97
4.2.7	显示屏	98
4.2.8	扩展 I/O 口	98

4.3	软件开发环境	99
4.3.1	嵌入式软件开发流程	99
4.3.2	程序开发库 CMSIS	101
4.3.3	STM32L52 嵌入式程序开发预备知识	103
第 5 章	通用输入输出	112
5.1	GPIO 原理	112
5.1.1	GPIO 功能	112
5.1.2	I/O 模式配置	113
5.2	GPIO 寄存器	115
5.3	GPIO 操作函数库	120
5.4	GPIO 实例	128
5.4.1	GPIO 寄存器基本操作	128
5.4.2	GPIO LED 灯控制	130
5.4.3	GPIO 按键输入	131
第 6 章	异常和中断处理技术	133
6.1	中断的基本概念	133
6.2	中断向量表	134
6.3	中断的执行过程	138
6.3.1	中断响应基本流程	138
6.3.2	中断优化技术	141
6.3.3	系统异常	142
6.4	嵌套向量中断控制器 NVIC	144
6.4.1	STM32L152 NVIC	144
6.4.2	NVIC 寄存器	144
6.4.3	系统异常处理	149
6.4.4	全局中断管理	150
6.4.5	NVIC 库函数	151
6.5	外部中断/事件控制器 EXTI	157
6.6	寄存器说明	158
6.7	EXTI 函数库	162
6.8	中断案例	164
第 7 章	定时器	168
7.1	定时器原理概述	168
7.2	内部定时器 SysTick	170
7.2.1	SysTick 寄存器	170

7.2.2	SysTick 定时器库函数	172
7.2.3	SysTick 定时器应用例程	174
7.3	外围定时器基本概念	175
7.4	基本定时器 TIM6、TIM7	179
7.5	通用定时器 TIM2~TIM4、TIM9~TIM11	181
7.5.1	通用定时器时基单元	182
7.5.2	通用定时器输入捕获和输出比较单元	185
7.5.3	TIMx 的外部触发同步模式	189
7.6	定时器寄存器	191
7.7	外围定时器库函数	197
7.8	定时器应用例程	205
7.8.1	定时器寄存器操作案例	205
7.8.2	基本计时中断示例	206
7.8.3	比较输出示例	208
7.8.4	输入捕获示例	212
7.8.5	PWM 输出和输入示例	215
第 8 章	USART 串口控制器	218
8.1	串行输入输出接口的基本概念	218
8.2	串行通信协议	219
8.2.1	异步串行通信协议	219
8.2.2	同步串行通信协议	220
8.2.3	串行通信基本概念	221
8.3	STM32L152 USART 内部结构与原理	224
8.3.1	发送器	226
8.3.2	接收器	228
8.3.3	校验控制	232
8.3.4	硬件流控制	232
8.3.5	USART 中断请求	233
8.4	USART 寄存器	234
8.5	USART 数据传输配置	240
8.5.1	波特率计算	240
8.5.2	异步双向通信模式配置	241
8.6	USART 帧传输协议	242
8.6.1	串行链路帧格式设计	242
8.6.2	MODBUS 帧格式	246
8.7	USART 函数库	247
8.7.1	寄存器定义	247

8.7.2	USART 库函数	250
8.8	USART 案例	257
8.8.1	串口寄存器操作案例	257
8.8.2	串口配置基本流程	258
8.8.3	PC 串口通信案例	259
8.8.4	状态机多字节数据帧发送和接收案例	261
第 9 章	IIC 总线	267
9.1	IIC 总线概述	267
9.2	I2C 总线的基本操作	268
9.3	STM32L152 I2C 总线控制器	273
9.4	I2C 寄存器描述	275
9.5	I2C 数据通信流程	281
9.5.1	I2C 从模式通信	281
9.5.2	I2C 主模式通信	283
9.5.3	总线通信错误	285
9.5.4	中断请求	285
9.6	函数库	287
9.6.1	I2C 寄存器结构	287
9.6.2	I2C 库函数	288
9.7	I2C 案例	298
9.7.1	I2C 寄存器操作案例	298
9.7.2	I2C 基本配置	299
9.7.3	模拟 I2C 实现	301
9.7.4	串行 Flash 通信	304
9.7.5	ADT7420 温度传感器通信	306
第 10 章	SPI	309
10.1	SPI 总线概述	309
10.2	SPI 总线控制器架构	310
10.2.1	接口信号和连接方式	310
10.2.2	传输模式和时序	313
10.2.3	STM32L15x SPI 总线控制器	315
10.3	SPI 寄存器说明	317
10.4	SPI 通信流程	320
10.4.1	SPI 双工通信模式配置	321
10.4.2	SPI 单工/半双工通信	323
10.5	函数库	326

10.5.1	SPI 寄存器结构	326
10.5.2	SPI 库函数	328
10.6	SPI 案例	333
10.6.1	SPI 寄存器操作案例	333
10.6.2	SPI 函数库案例	334
10.6.3	温度传感器 ADT7320 案例	335
第 11 章	模拟/数字转换	339
11.1	ADC 简介	339
11.2	STM32L152 ADC	344
11.2.1	STM32L152 ADC 功能	346
11.2.2	温度和电压转换	357
11.3	ADC 寄存器	358
11.4	ADC 寄存器结构及 ADC 库函数	365
11.4.1	ADC 寄存器结构	366
11.4.2	ADC 库函数	368
11.5	ADC 案例	377
11.5.1	ADC 寄存器操作案例	377
11.5.2	ADC 库函数操作案例	378
第 12 章	低功耗技术	380
12.1	处理器功耗的构成/类型	380
12.1.1	动态功耗	380
12.1.2	静态功耗	381
12.2	STM32L1 系列处理器低功耗设计	382
12.2.1	STM32 的电源系统	382
12.2.2	动态电压调节管理	383
12.2.3	电源检测	385
12.2.4	低功耗模式	386
12.3	功耗控制寄存器	389
12.4	PWR 寄存器结构及库函数	391
12.4.1	PWR 寄存器结构	391
12.4.2	PWR 库函数	391
12.5	PWR 案例	394
参考文献	397

第 1 章 微型计算机与嵌入式系统概论

【导读】 嵌入式系统属于微型计算机范畴,本章首先介绍微型计算机的基本组成,核心部件微处理器的发展历史,然后对微处理器工作原理的一些基本概念:组成架构、总线、输入输出、存储系统等进行了介绍。微处理器应用于微机,属于通用计算机系统;而与应用场景结合,形成专用计算系统,称为嵌入式系统,本章对嵌入式系统的概念、组成和典型应用进行介绍,并列举了目前典型的开源嵌入式开发硬件和软件平台。通过本章学习,建立微机系统的整体构成和微处理器设计的相关计算思维方法,对嵌入式系统的概念及软硬件系统有总体的认识。

1.1 微型计算机概述

微型计算机是针对小型计算机、大型计算机和超级计算机而言的,是根据规模和性能进行计算机分类的。一般来说,微型计算机是一种小型的、相对便宜的、以微处理器作为 CPU 的计算机。这类计算机由印制电路板、微处理器、存储器和输入输出电路组成,占用很少的物理空间。随着集成电路技术的发展,微型计算机在 20 世纪 80 年代开始流行,获得广泛的应用。目前我们使用的个人计算机(台式机、笔记本计算机、平板计算机、智能手机、计算器等)、家用娱乐设施(游戏机、智能电视、智能音箱、电子书阅读器等)以及路由器、交换机等通信设备均是微型计算机系统。

1.1.1 微型计算机系统的组成

一个完整的微型计算机系统由硬件系统和软件系统两部分组成,如图 1-1 所示。

计算机硬件部分包括中央处理器(CPU)、存储器、输入和输出设备。

(1) 微型计算机的中央处理器也称为微处理器(Micro Processor Unit, MPU)。计算机利用 CPU 处理数据,利用存储器存储数据。CPU 是计算机硬件的核心,主要包括运算器和控制器两大部分,控制着整个计算机系统的工作。计算机的性能主要取决于 CPU 的性能。

运算器又称为算术逻辑单元(Arithmetic Logic Unit, ALU),控制器的主要作用是使整个计算机系统能够自动运行。控制器从存储器取出数据,运算器进行算术运算或逻辑运算,

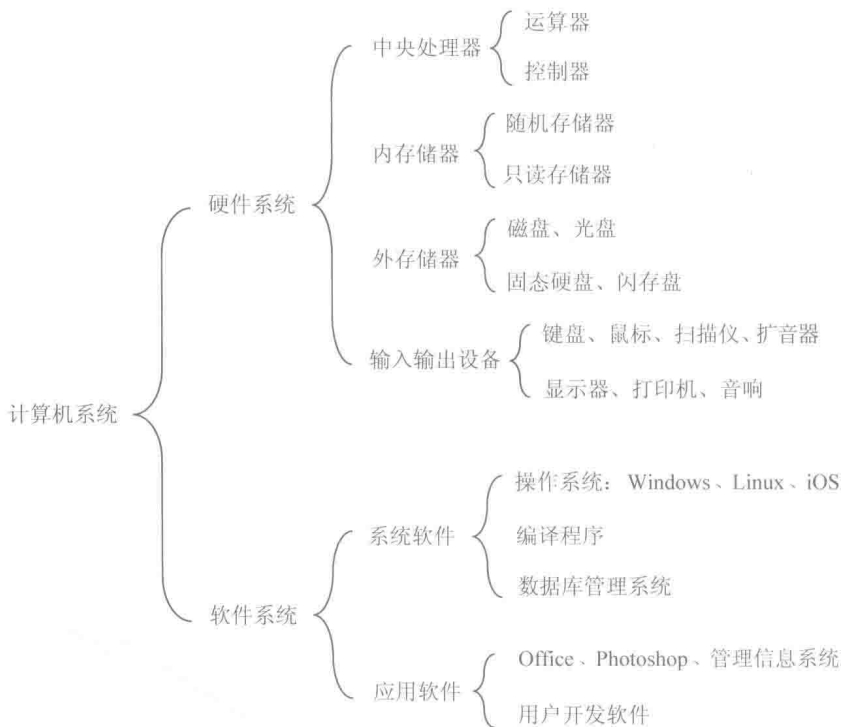


图 1-1 计算机的系统组成

并把处理后的结果送回存储器。执行程序时,控制器从主存中取出相应的指令和数据,然后向其他功能部件发出指令所需的控制信号,完成相应的操作,再从主存中取出下一条指令执行,如此循环,直到程序结束。

(2) 存储器是计算机中的存储部件。存储器分为主存(内存储器)和辅存(外存储器)两大类。在计算机系统中,习惯上把内存、CPU 合称为主机。内存储器分为随机读写存储器(RAM)、只读存储器(ROM)和高速缓冲存储器(Cache)三类。通常生活中的内存一般指的是 RAM。外存储器主要包括硬盘、光盘、U 盘和移动硬盘等。

(3) 输入输出设备主要包括键盘、鼠标、显示器和打印机等。

硬件是组成计算机的基础,软件是计算机的灵魂。计算机的硬件系统上只有安装了软件后,才能发挥其应有的作用,使用不同的软件,计算机可以完成各种不同的工作。微型计算机系统的软件分为两大类,即系统软件和应用软件。

系统软件是指为管理计算机系统的硬件和支持应用软件运行而提供的基本软件,最常用的有操作系统、程序设计语言编译器、数据库管理系统、联网及通信软件等。操作系统是微机最基本、最重要的系统软件,它负责管理计算机系统的各种硬件资源(例如 CPU、内存空间、磁盘空间、外部设备等),并且负责将用户对机器的管理命令转换为机器内部的实际操作。典型的操作系统有 Linux、Mac OS、Windows 7、Windows 10、iOS 等。

应用软件是指除了系统软件以外,利用计算机为解决某类问题而设计的程序的集合,主要包括信息管理软件、辅助设计软件、实时控制软件等。

1.1.2 微处理器的发展

微型计算机系统的核心是微处理器,微处理器的发展直接影响着微型计算机系统的应用。

1.1.2.1 微处理器发展史

1) 早期微处理器

第一款微处理器是美国军方研制的中央空气数据计算机(Center Air Data Computer, CADC),由6颗晶片组成,用于F-14雄猫战机的大气数据测量与控制。1971年,Intel公司发布的4004是世界上第一款商用处理器,主频108kHz,4004和Intel开发出的4001(动态内存DRAM)、4002(只读存储器ROM)、4003(寄存器Register)可架构出一台微型计算机硬件系统。

2) 8位微处理器时期

1972年,Intel公司推出的8008微处理器是第一款8位处理器,主频0.5MHz。1974年,Intel公司推出8080处理器,主频2MHz,16位地址总线、8位数据总线,内部集成7个8位寄存器,支持16位内存,同时也包含了一些输入输出端口。

此时,微处理器的优势已被业界所认同,更多公司开始进入微处理器设计,仙童、AMD、摩托罗拉以及Zilog等公司均开始研发微处理器,摩托罗拉1974年发布了MC6800,工作主频1MHz,1976年Zilog公司发布了Z80,性能比8080更强大。

3) 16位微处理器时期

1978年,Intel公司首次生产出16位的微处理器,命名为8086,同时还生产出与之相配合的数学协处理器8087,这两种芯片使用相互兼容的指令集,即后来PC使用的X86指令集。

1979年,Intel公司推出了8088芯片,主频4.77MHz,地址总线为20位,寻址范围1MB内存。8088内部数据总线都是16位,外部数据总线是8位(8086是16位)。1981年8088芯片首次用于IBM PC中,开创了全新的微机时代。

1979年,Zilog发布了其第一款16位处理器Z8000,摩托罗拉发布16位处理器MC68000(16位计算单元,32位数据总线)。

1982年,Intel公司推出80286芯片,它比8086和8088都有了飞跃式的发展,虽然它仍旧是16位结构,时钟频率20MHz。其内部和外部数据总线皆为16位,地址总线24位,可寻址16MB内存。

4) 32位微处理器时期

世界上第一块单片32位微处理器是1982年AT&T(贝尔)实验室的BELLMAC-32A。

1985年,Intel公司发布了80386,首次在X86处理器中实现了32位系统,集成80387数字辅助处理器增强浮点运算能力,首次采用外置的高速缓存(Cache)解决内存速度瓶颈问题。工作频率也从12.5MHz逐步提高到20MHz、25MHz、33MHz,直至最后的40MHz。



图 1-2 Intel 早期 4 位、8 位、16 位处理器

1985 年摩托罗拉推出了 MC68020,增加了 32 位数据和地址总线,在 UNIX 超级微机市场上获得巨大成功,后续又生产了 MC68030(集成了内存管理)、MC68040(集成浮点运算器)。

1986 年 MIPS 推出 R2000 处理器,1988 年推出 R3000 处理器,采用精简指令集(RISC)设计,成为 RISC 微处理器的代表。

1987 年 Sun 公司推出了第一款 32 位的 SPARC 86900 Sunrise 处理器,这款处理器采用 SPARC V7 架构,采用 $0.8\mu\text{m}$ 工艺,主频 16MHz,主要用于 SUN 工作站(Solaris 系统)。

1989 年 Intel 公司发布 80486,支持虚拟存储管理技术,虚拟存储空间 64TB(支持 48 位的有效虚拟地址)。片内集成有浮点运算部件和 8KB 的 Cache,同时也支持外部 Cache。整数处理部件采用精简指令集 RISC 结构,提高了指令的执行速度。此外,80486 微处理器还引进了时钟倍频技术和新的内部总线结构,从而使主频可以超出 100MHz。

之后,Intel 公司陆续发布了 Pentium 系列处理器 Pentium(超标量)、Pentium Pro(动态执行)、Pentium II(MMX 指令集)、Pentium III(SIMD)、Pentium 4、Pentium M(低功耗)、Pentium D(双核)、酷睿 Core 等一系列处理器。目前,酷睿系列(Core M、Core I3、Core I5、Core I7、Core I9)已经经过了 8 代的发展。

1991 年,ARM 发布了自己的第一款 RISC 处理器核 ARM6,1993 年推出 ARM7,一直到目前的 ARM11 和 Cortex,在嵌入式微处理器市场占据了大量份额。

5) 64 位微处理器时期

1991 年,MIPS 推出第一款 64 位商用微处理器 R4000,之后又陆续推出 R8000、R10000 和 R12000 等型号,2007 年,中科院计算所龙芯获得 MIPS 处理器 32 位和 64 位的授权。

1992 年,DEC 发布了 64 位的 Alpha 处理器,主要用于工作站和服务器,后 DEC 将 Alpha 技术出售给康柏公司,最终康柏公司将 Alpha 的技术出售给 Intel 公司。

1995 年,Sun 公司推出了 64 位 UltraSPARC I 微处理器。UltraSPARC I 革新了微处理器的可扩展性和带宽等工业标准,其频率达 143MHz,采用 $0.5\mu\text{m}$ 工艺技术。

2001 年,Intel 公司推出了 IA64 架构的 Itanium 系列处理器,其指令系统与 X86 不兼容,用于服务器市场。

2003 年,AMD 提出了 X86 的 64 位扩展指令集 AMD64,用于服务器市场的 64 位处理器进入到 PC 领域,后来 Intel 公司最终采用了 AMD64。

ARM 于 2011 年发布了 ARMv8 64 位架构,ARMv8 使用了两种执行模式:AArch32 和 AArch64,处理器在运行中可以无缝地在两种模式间切换。这意味着 64 位指令的解码

器是全新设计的,不用兼顾 32 位指令,而处理器依然可以向后兼容。

1.1.2.2 微控制器发展史

微控制器(Micro Controller Unit,MCU)是一种采用超大规模集成电路技术把具有数据处理能力的微处理器 MPU、随机存储器 RAM、只读存储器 ROM、多种 I/O 接口和中断系统、定时器/计数器等功能集成到一块硅片上构成的一个小而完善的微型计算机系统。

早期的微控制器称作单片机(Single Chip Microcomputer,SCM),主要实现单片系统集成,随着嵌入式应用的扩展,系统各种外围接口电路日趋复杂,系统的智能化控制能力凸显,各种电气、控制和电子技术厂家开始进入微处理器的设计和生产,结合 SoC(System on Chip)技术,单片机进入微控制器阶段。

1976 年 Intel 公司推出了第一款单片机 MCS-48。此后,GI 推出了 PIC1x 系列 8 位单片机,摩托罗拉推出了 MC6800 处理器为核心的 M6800 系列单片机,Zilog 推出了 Z8 系列单片机。

1990 年,Intel 公司推出了 MCS-51,MCS-51 采用经典的 8 位单片机的总线结构,包括 8 位数据总线、16 位地址总线、控制总线及具有很多机通信功能的串行通信接口,并授权给其他厂家使用,Intel、Atmel、Philips 和 STC 等推出了一系列 MCS-51 核心的单片机,从此成为 8 位单片机的主流,直到现在还在大量应用。此外,8 位单片机的主流架构还有 Atmel 公司的 AVR 系列,凌阳科技的 SPMC65 系列等。

8 位单片机占据了较大的市场,16 位单片机由于 8 位单片机的市场地位和 32 位单片机的快速发展,相对发展空间较小,典型的芯片包括 Intel 的 MCS-96 系列单片机,TI 的 MSP430 系列,摩托罗拉的 68HC12/16 系列以及 MicoChip 的 PIC24 系列。

随着嵌入式系统应用的爆发,单片机的运算处理性能和外设通信能力等无法满足复杂应用的计算需求,由此催生了 32 位微控制器。32 位微控制器多采用 RISC 指令集,典型的代表有 MIPS、ARM、摩托罗拉的 68K 系列等。目前,微控制器基本已被 ARM 占据,随着物联网应用的发展,芯片厂家的主要在低功耗、低电压、大容量存储和高性能计算上对微控制器进行优化。

1.1.2.3 CISC 和 RISC 设计方法

按照指令系统分类,计算机大致可以分为两类:复杂指令系统计算机(Complex Instruction Set Computer,CISC)和精简指令系统计算机(Reduced Instruction Set Computer,RISC)。CISC 是 CPU 的传统设计模式,其指令系统的特点是指令数目多而复杂,每条指令的长度不尽相等;而 RISC 则是 CPU 的一种新型设计模式,其指令系统的主要特点是指令条数少且简单,指令长度固定。

1) CISC 指令集

计算机的指令系统最初只有很少的一些基本指令,而其他的复杂指令全靠软件编译时通过简单指令的组合来实现。后来,越来越多的复杂指令被加入到了指令系统中,可用硬件实现复杂的运算。但是,一个指令系统的指令条数受到指令操作码位数的限制,如果操作码

为 8 位,那么指令条数最多为 256 条,而指令的宽度则是很难增加的。操作码扩展可以解决这个问题,在指令格式中,操作码后面跟的是地址码,而有些指令是用不到地址码或只用少量位数的地址码的,那么就可以把操作码扩展到地址码的位置,使操作码的位数得以增加。例如,一个指令系统的操作码为 2 位,那么可以有 00、01、10、11 四条不同的指令。现在把 11 作为保留,把操作码扩展到 4 位,那么就可以有 00、01、10、1100、1101、1110、1111 七条指令,其中 1100、1101、1110、1111 这四条指令的地址码部分必须减少两位。为了达到减少地址码这一操作码扩展的先决条件,设计者提出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。

由此,大量的复杂指令、可变的指令长度、多种寻址方式形成了 CISC 指令集。CISC 指令集的复杂性大大增加了译码的难度,早期计算机运算能力差,译码相对时间较短,现代计算机运算速度大大提升,导致译码上所浪费的时间过长,严重影响了计算机性能。

2) RISC 指令集

1975 年,IBM 对 IBM 370 CISC 系统的研究发现,发现其中仅占总指令数 20% 的简单指令却在程序调用中占据了 80%,而占指令数 80% 的复杂指令却只有 20% 的机会被调用到,即符合经济学中的 80/20 法则,由此提出了 RISC 的概念。20 世纪 80 年代末开始,各家公司的 RISC CPU 大量出现,其中典型代表为 MIPS 和 ARM。

RISC 体系结构的基本思想:针对 CISC 指令系统指令种类太多、指令格式不规范、寻址方式太多的缺点,通过减少指令种类、规范指令格式、简化寻址方式,方便处理器内部的并行处理,提高处理器内部器件的使用效率,从而大幅度地提高处理器的性能。

RISC 的目标决不是简单地缩减指令系统,而是使处理器的结构更简单、更合理,具有更高的性能和执行效率,同时降低处理器的开发成本。由于 RISC 指令系统仅包含最常用的简单指令,因此,RISC 技术可以通过硬件优化设计,把时钟频率提得很高,从而实现整个系统的高性能。同时,RISC 技术在 CPU 芯片上设置大量寄存器,用来把常用的数据保存在这些寄存器中,大大减少对存储器的访问,用高速的寄存器访问取代低速的存储器访问,从而提高系统整体性能。

RISC 的典型特征包括:

(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 4 字节),并且在字边界上对齐,字段位置(特别是操作码的位置)固定。

(2) 寻址方式简化:几乎所有指令都使用寄存器寻址方式,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。

(3) 大量利用寄存器间操作:RISC 指令集中大多数操作都是寄存器到寄存器的操作,只有取数指令、存数指令访问存储器。

(4) 简化处理器结构:使用 RISC 指令集,可以大大简化处理器中的控制器和其他功能单元的设计,不必使用大量专用寄存器,特别是允许以硬连线方式来实现指令操作,以期更快的执行速度,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,从而能够快速地直接执行指令。

(5) 加强处理器的并行能力: RISC 指令集非常适合于采用流水线、超流水线和超标量技术,从而实现指令级并行操作,提高处理器的性能。目前常用的处理器的内部并行操作技术,基本上都是基于 RISC 体系结构而逐步发展和走向成熟的。

(6) RISC 技术的复杂性在于它的优化编译程序,因此软件系统开发时间比 CISC 机器要长。

RISC 与 CISC 的主要特征对比如表 1-1 所示。

表 1-1 RISC 与 CISC 的主要特征对比

比较项目	CISC	RISC
指令系统	复杂、庞大	简单、精简
指令数目	一般大于 200	一般小于 100
指令格式	一般大于 4 种	一般小于 4 种
寻址方式	一般大于 4 种	一般小于 4 种
指令字长	不固定	定长
指令执行时间	慢	快
程序代码长度	短	长

1.2 微型计算机的基本原理

1.2.1 冯·诺依曼体系结构

现代计算机基本沿用冯·诺依曼体系结构,其基本设计思想包括以下三点。

1) 计算机系统的组成

运算器、存储器(主存)、控制器、输入设备和输出设备五大部件组成一个完整的计算机系统,如图 1-3 所示。

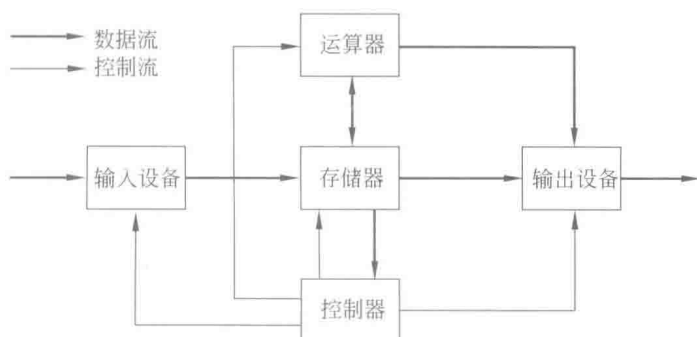


图 1-3 冯·诺依曼体系结构五大功能部件