

高等院校计算机教材

高等教育自学考试教材

# 微型计算机 基本原理 与应用

王克义  
宋新波 编著  
罗建英

北京大学出版社

高等院校计算机教材  
高等教育自学考试教材

# 微型计算机基本原理与应用

王克义 宋新波 罗建英 编著

北京 大学 出版社  
北 京

## 内 容 简 介

本书为高等院校计算机教材,并被列为北京市高等教育自学考试指定教材。本书着重讲述微型计算机的基本结构和工作原理;指令系统和汇编语言的基础知识,汇编语言程序设计方法;中断系统和输入/输出接口技术,主要 I/O 接口部件的工作原理和使用方法,高性能微处理器及相关技术等。

本书由主讲教师严格按照教学大纲编写,内容精练,层次清楚,实用性强。本书可作为大学理工科各专业计算机课程教材,也可作为自学考试和成人教育以及各类职业学校的教材。

### 图书在版编目(CIP)数据

微型计算机基本原理与应用/王克义等编著. —北京:北京大学出版社,1997.11  
ISBN 7-301-03511-X

I. 微… II. 王… III. 微型计算机-基本知识 IV. TP36

书 名 : 微型计算机基本原理与应用

著作责任者: 王克义 宋新波 罗建英

责任编辑: 杨锡林

标准书号: ISBN 7-301-03511-X/TP·371

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

电 话: 出版部 62752015 发行部 62559712 编辑部 62752032

排 印 者: 北京飞达印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787×1029 16开本 20.375印张 510千字

1997年10月第一版 1997年10月第一次印刷

定 价: 28.00元

# 前 言

“微型计算机原理与应用”是高校理工科大学生的一门重要的计算机技术基础课程。学习本课程对于掌握现代微型计算机技术及学习后续有关计算机课程(如汇编语言程序设计、计算机体系结构、操作系统等)均具有重要意义。本书是该课程使用的基本教材。

目前,随着大规模和超大规模集成电路技术的飞速发展,现代微型计算机技术的发展真可谓日新月异,一日千里。386,486,Pentium,MMX…,新技术、新产品层出不穷,让人目不暇接。在这种情况下,一个突出的问题摆在了我们面前,即在这种新技术、新产品迅速更新换代,技术知识的有效生存周期较短的形势下,学校里的计算机技术基础课程教学究竟如何进行?教学内容如何选取?教学环节如何组织?怎样在教学中反映和吸收当前最新技术?

对这一问题,目前各学校的看法和做法不尽相同。编者则比较赞同下述观点:即学校的计算机教学确应努力反映计算机科学技术发展的新概念和新技术,但作为入门和专业技术基础课程,还应着重从基本概念、基本分析方法和基本能力的培养入手,恰当选择那些带有共性和有利于学生能力培养的教学内容,实施有效的教学环节,从而达到预期的教学目标。对于“微机原理与应用”课程来说,首先就应使学生较好地理解和掌握微型计算机的基本组成结构、基本工作原理及典型应用,确实掌握微型计算机到底是怎样构成的,它们又是如何工作的。在此基础上,再学习和掌握微机发展的最新成果和技术。这样做较符合人们的认识规律和知识学习过程。本书在内容的选取和结构安排上力图体现这一思想。

全书内容共分十一章,从整体上可划分为四个知识单元:① 微型计算机的基本结构及工作原理(第一,二,五,六,十章);② 汇编指令基础及汇编语言程序设计入门(第三,四章);③ I/O接口技术(第七,八,九章);④ 高性能微处理器及相关技术(第十一章)。

学习本教材的预备知识为数字电路及逻辑设计基础知识。

本教材可供60—70学时的课堂教学使用,有些章节的内容可根据不同的教学要求进行适当取舍。每章后面列出的思考题与习题,主要供理解和复习本章基本内容而用,习题解析及补充、提高性习题将由北京大学出版社另行编辑出版。

另外,鉴于本课程是技术性、实践性较强的课程,因此应有相应的实验及上机环节。教师可根据具体实验设备及上机条件,安排相应的接口实验及汇编程序上机内容。

本书由北京大学计算机科学技术系王克义,中国科学院计算技术研究所宋新波,美国德克萨斯州奥斯汀大学计算机科学系在读博士生罗建英三位编著者合作完成。王克义担任主编,并编写第一,七,八,九章,宋新波编写第二,三,四,五,十章,罗建英编写第六,十一章。

本书是在作者近几年给北京大学计算机系本科生、北京大学理学试验班教学实践的基础上编写而成的,并参考和吸收了国外较新同类教科书及国内兄弟院校优秀教材的有关内容,在此,特向有关作者一并致谢。

在本书的编写过程中,承蒙国家教委高校计算机教学指导委员会副主任、北京大学计算机科学技术系许卓群教授,北京大学计算机科学技术系毛德行教授、余娟芬教授的热情支持和指导,在此谨向他们表示衷心的感谢。

由于编者的水平所限,书中一定存在不少差错和疏漏,诚请广大读者及专家批评指正。

编著者

1997年10月于北京大学

# 目 录

第一章 概论	(1)
第一节 计算机的基本结构	(1)
一、冯·诺伊曼计算机基本结构	(1)
二、计算机的基本组成框图及功能部件简介	(2)
第二节 计算机的工作流程	(4)
一、指令与程序	(4)
二、计算机的基本工作流程	(5)
第三节 计算机系统的组成	(7)
一、硬件与软件	(7)
二、计算机系统的基本组成	(8)
第四节 微型计算机的产生和发展	(9)
一、世界上第一台微型计算机的诞生	(9)
二、微型计算机的发展阶段	(9)
第五节 微型计算机的特点与分类	(11)
一、微型计算机的主要特点	(11)
二、微型计算机的分类	(12)
三、微型计算机的主要技术指标	(12)
第六节 微型计算机的基本结构与系统组成	(13)
一、微型计算机基本结构	(14)
二、微型计算机的系统组成	(15)
思考题与习题	(16)
第二章 微型计算机体系结构	(17)
第一节 CPU 的内部逻辑结构	(17)
一、CPU 的功能	(17)
二、8086/8088 的内部逻辑结构	(17)
第二节 CPU 的外部结构	(22)
一、8086 CPU 的引脚功能	(22)
二、8088 CPU 的引脚功能	(26)
第三节 存储器结构	(26)
一、8086 系统中的存储器结构	(26)
二、8088 系统中的存储器结构	(31)
第四节 系统配置	(32)
一、概述	(32)
二、最小模式系统	(33)
三、最大模式系统	(37)
第五节 CPU 的操作和时序	(43)

一、基本概念 .....	(43)
二、系统的复位与启动操作 .....	(44)
三、最小模式系统中的总线操作 .....	(45)
四、最大模式系统中的总线操作 .....	(47)
思考题与习题 .....	(49)
<b>第三章 指令系统</b> .....	(50)
一、第一节 寻址方式 .....	(50)
一、程序存储器寻址 .....	(51)
二、数据存储器寻址 .....	(51)
三、寻址方式的书写格式 .....	(53)
二、指令编码格式 .....	(53)
一、概述 .....	(53)
二、指令编码 .....	(54)
三、指令编码举例 .....	(57)
三、指令执行时间 .....	(58)
一、概述 .....	(58)
二、有效地址计算 .....	(58)
三、指令执行时间计算 .....	(58)
四、8086/8088 的汇编指令系统 .....	(60)
一、数据传送指令 .....	(60)
二、算术运算指令 .....	(63)
三、逻辑运算和移位指令 .....	(72)
四、串操作指令 .....	(75)
五、转移指令 .....	(77)
六、处理器控制指令 .....	(82)
五、80x86 的指令系统 .....	(84)
一、80286 的指令系统 .....	(84)
二、80386 的指令系统 .....	(85)
三、80486 的指令系统 .....	(87)
思考题与习题 .....	(87)
<b>第四章 汇编语言程序设计</b> .....	(88)
一、第一节 汇编语言程序格式 .....	(88)
一、示例程序 .....	(88)
二、一般概念 .....	(89)
三、指令语句 .....	(94)
四、伪指令语句 .....	(95)
五、结构与记录 .....	(100)
六、宏指令 .....	(101)
七、简化段 .....	(104)
二、第二节 程序设计方法 .....	(105)
一、程序设计的基本步骤 .....	(105)
二、程序的基本结构形式 .....	(105)

三、子程序设计 .....	(106)
第三节 BIOS 和 DOS 中断 .....	(107)
第四节 汇编语言程序设计技巧 .....	(108)
一、I/O 与通信 .....	(108)
二、声音与时钟 .....	(109)
三、图形显示 .....	(111)
四、文件管理 .....	(111)
第五节 高级汇编语言技术 .....	(112)
一、条件汇编 .....	(112)
二、汇编语言与高级语言混合编程 .....	(113)
三、库的使用 .....	(114)
第六节 汇编语言程序的上机过程 .....	(114)
一、编辑程序 .....	(114)
二、汇编程序 .....	(115)
三、连接程序 .....	(115)
四、调试程序 .....	(115)
思考题与习题 .....	(116)
<b>第五章 中断系统</b> .....	(117)
第一节 基本概念 .....	(117)
一、中断 .....	(117)
二、中断处理过程 .....	(118)
三、中断优先级和中断嵌套 .....	(118)
第二节 8086/8088 的中断系统 .....	(119)
一、中断的分类 .....	(119)
二、中断向量表 .....	(119)
三、外部中断 .....	(121)
四、内部中断 .....	(122)
五、中断响应和中断处理过程 .....	(124)
第三节 可编程中断控制器 8259A .....	(125)
一、8259A 的引脚功能 .....	(125)
二、8259A 的内部结构 .....	(126)
三、8259A 的工作方式 .....	(127)
四、8259A 的级联使用 .....	(129)
五、8259A 的控制字及编程使用 .....	(131)
思考题与习题 .....	(135)
<b>第六章 微型计算机存储器</b> .....	(136)
第一节 存储器概述 .....	(136)
一、存储器的分类 .....	(136)
二、存储器的性能指标 .....	(137)
第二节 半导体存储器 .....	(137)
一、可读写存储器 RAM .....	(138)

二、只读存储器 ROM .....	(145)
第三节 存储器与 CPU 的连接 .....	(149)
一、存储器与 CPU 连接时要考虑的问题 .....	(149)
二、存储器中的片选译码 .....	(150)
三、存储器芯片的连接 .....	(152)
第四节 PC 微机的存储器 .....	(154)
一、PC 微机存储器空间分布 .....	(154)
二、PC 微机内存类型及管理 .....	(155)
<del>第五节</del> PC 微机存储器的扩展技术 .....	(156)
一、高速缓冲存储器(Cache) .....	(156)
二、虚拟存储技术 .....	(157)
三、磁盘高速缓存区(磁盘 Cache)和虚拟磁盘技术 .....	(159)
思考题与习题 .....	(160)
<b>第七章 输入/输出接口</b> .....	(161)
第一节 I/O 接口的基本概念 .....	(161)
一、I/O 接口的基本功能 .....	(161)
二、I/O 接口的基本结构 .....	(162)
三、I/O 端口的编址方式 .....	(163)
四、I/O 接口的地址分配 .....	(164)
五、I/O 接口的地址译码及片选信号的产生 .....	(165)
六、I/O 指令 .....	(166)
第二节 I/O 控制方式 .....	(166)
一、程序控制方式 .....	(166)
二、中断控制方式 .....	(167)
三、DMA 方式 .....	(168)
<del>第三节</del> DMA 接口技术 .....	(170)
一、DMA 控制器的基本功能 .....	(170)
二、DMA 控制器的一般结构 .....	(170)
三、DMA 控制器的工作方式 .....	(172)
四、DMA 工作过程 .....	(173)
<del>第四节</del> 可编程 DMA 控制器 8237 .....	(174)
一、8237 的结构 .....	(174)
二、8237 的引脚 .....	(175)
三、8237 的工作方式 .....	(177)
四、8237 的寄存器及有关问题的说明 .....	(179)
五、特殊软件命令 .....	(183)
六、8237 内部寄存器的寻址 .....	(183)
七、8237 的工作周期与操作时序 .....	(184)
八、8237 的编程 .....	(187)
思考题与习题 .....	(188)
<b>第八章 串并行通信及接口电路</b> .....	(190)
√ 第一节 串行通信的基本概念 .....	(190)

一、串行通信的特点 .....	(190)
二、串行通信涉及的常用术语和基本概念 .....	(191)
第二节 串行通信接口标准 .....	(196)
<del>一、RS-232C .....</del>	<del>(196)</del>
<del>二、RS-449 及 RS-423A、RS-422A .....</del>	<del>(202)</del>
第三节 串行接口的基本结构与功能 .....	(202)
一、串行异步接口的基本结构与功能 .....	(203)
二、串行同步接口的基本结构与功能 .....	(204)
√ 第四节 可编程串行通信接口片 8251A .....	(205)
<del>一、USART .....</del>	<del>(205)</del>
二、8251A 的基本功能和工作原理 .....	(206)
三、8251A 对外接口信号 .....	(208)
四、8251A 的编程 .....	(211)
五、8251A 应用举例 .....	(216)
√ 第五节 并行通信及接口电路 .....	(219)
一、并行接口的组成及其与 CPU 和外设的连接 .....	(219)
二、并行接口的数据输入输出过程 .....	(220)
√ 第六节 可编程并行通信接口片 8255A .....	(221)
一、8255A 的性能概要 .....	(221)
二、8255A 芯片引脚分配及引脚信号说明 .....	(221)
三、8255A 内部结构方块图 .....	(222)
√ 四、8255A 的控制字 .....	(223)
√ 五、8255A 的工作方式 .....	(225)
√ 六、8255A 的状态字 .....	(230)
七、8255A 应用举例 .....	(231)
思考题与习题 .....	(234)
第九章 计数/定时技术 .....	(237)
第一节 概述 .....	(237)
√ 第二节 可编程计数器/定时器 8253 .....	(237)
√ 一、8253 的主要功能 .....	(237)
√ 二、8253 的结构框图 .....	(238)
√ 三、8253 的引脚 .....	(240)
四、8253 的工作方式 .....	(240)
五、8253 的初始化编程 .....	(246)
六、8253 的读出操作 .....	(248)
第三节 8253 的应用 .....	(249)
<del>第四节 多功能接口电路 82380 .....</del>	<del>(252)</del>
一、82380 的结构 .....	(252)
二、82380 的 DMA 功能 .....	(253)
三、82380 的中断功能 .....	(254)
四、82380 的定时器功能 .....	(255)
思考题与习题 .....	(255)

<b>第十章 总线</b> .....	(256)
<b>第一节 概述</b> .....	(256)
一、总线的概念.....	(256)
二、总线的分类.....	(256)
三、总线仲裁.....	(257)
<b>第二节 MultiBus 介绍</b> .....	(259)
一、MultiBus 的信号说明.....	(259)
二、MultiBus 的总线操作.....	(263)
<b>第三节 ISA 和 EISA 总线</b> .....	(264)
一、ISA 总线介绍.....	(264)
二、EISA 总线介绍.....	(266)
<b>第四节 VESA(VL-Bus)和 PCI 总线</b> .....	(267)
一、局部总线概述.....	(267)
二、VESA 的 VL-Bus 介绍.....	(267)
三、PCI 总线介绍.....	(268)
<b>思考题与习题</b> .....	(269)
<b>第十一章 高性能微处理器及其相关技术</b> .....	(270)
<b>第一节 80286 微处理器</b> .....	(270)
一、概述.....	(270)
二、80286 的基本结构.....	(270)
<b>第二节 80386 微处理器</b> .....	(273)
一、概述.....	(273)
二、80386 的基本结构.....	(274)
<b>第三节 RISC 技术</b> .....	(277)
<b>第四节 80486 与 Pentium 微处理器</b> .....	(279)
一、80486 微处理器.....	(279)
二、Pentium 微处理器.....	(280)
<b>第五节 Power PC 620 与 Alpha 21164 微处理器</b> .....	(282)
一、Power PC 620 微处理器.....	(282)
二、Alpha 21164 微处理器.....	(283)
<b>第六节 当前最新微处理器及发展趋势</b> .....	(284)
<b>思考题与习题</b> .....	(286)
<b>附录</b> .....	(287)
<b>附录一 8086/8088 的指令系统</b> .....	(287)
<b>附录二 80x86 指令系统</b> .....	(298)
<b>附录三 DOS 功能调用</b> .....	(302)
<b>附录四 BIOS 中断调用</b> .....	(307)
<b>附录五 调试程序 DEBUG 的使用</b> .....	(311)
<b>附录六 ASCII 码字符表</b> .....	(314)
<b>参考文献</b> .....	(315)

# 第一章 概 论

1946年,世界上第一台电子计算机(ENIAC)的诞生,标志着人类文明史上继语言的产生、文字的出现以及印刷术的发明之后,又出现的一个崭新的发展历程。但与以往各项科学技术经历的漫长发展周期不同,电子计算机从它诞生之日起就经历了并不断经历着极为迅速的发展过程。虽然至今才50年的历史,却经历了几代不断提高的发展阶段。尤其是进入本世纪70年代以后,随着微电子学在理论上和制造工艺上的成熟和发展,相继出现了大规模集成电路(LSI)和超大规模集成电路(VLSI),计算机技术除了向高性能巨型机方向发展外,还向微型计算机方向快速发展。各种各样的微处理器及微型计算机先后被研制出来,其性能不断提高,有的已经接近或超过以前中、小型计算机甚至大型计算机的处理能力和性能指标。微型计算机的发展速度大大超过了前几代计算机的发展速度,其更新换代周期之短,性能指标提高之快,是近代科学技术发展史上所罕见的。

为了对微型计算机的有关概念和知识有一个概括的了解,本章首先对一般计算机的结构及工作流程做简要介绍,然后介绍微型计算机的产生和发展,主要特点与分类,以及微型计算机的基本结构与系统组成等方面的内容。

## 第一节 计算机的基本结构

### 一、冯·诺伊曼计算机基本结构

世界上第一台电子计算机(ENIAC)采用电子管作主要构成元件,大大提高了运算速度,达到每秒钟完成加法运算5000次。但它存在一个主要缺陷是不能存储程序。它是由人工设置开关并以插入和拔出导线插头的方式来编制程序的。编程时需要对大约6000多位开关进行仔细的机械定位,并用转插线把选定的各个控制部分互连起来以构成程序序列。这种原始的机械式编程方法显然效率很低。

1944—1945年间,著名美籍匈牙利数学家冯·诺伊曼(John Von Neumann)应邀参加在美国宾夕法尼亚大学进行的ENIAC计算机研制任务。在研制过程中,他深深地感到ENIAC不能存储程序这一缺陷,并在1945年由他领导的EDVAC(Electronic Discrete Variable Automatic Computer——离散变量自动电子计算机)试制方案中,他作为一位主要倡导者指出:ENIAC的开关定位和转插线连接只不过代表着一些数字信息,它们完全可以像受程序管理的数据一样,存放于主存储器中。这就是最早的“存储程序”概念的产生。EDVAC计算机由运算器、逻辑控制装置、存储器、输入设备和输出设备五个部分组成。它采用了“存储程序”的思想,把数据和程序指令,均用二进制代码的形式存放在存储器中,保证了计算机能按事先存入的程序自动地进行运算。

冯·诺伊曼首先提出的“存储程序”的思想,以及由他首先规定的计算机的基本结构,人们称之为“冯·诺伊曼计算机结构”。归纳其基本内容,主要包括以下几点:

- (1) 计算机应由运算器、控制器、存储器、输入设备和输出设备五个部分组成。

(2) 数据和程序均以二进制代码形式不加区别地存放在存储器中,存放的位置由存储器的地址指定。

(3) 计算机在工作时能够自动地从存储器中取出指令加以执行。

半个世纪以来,随着计算机技术的不断发展和应用领域的不断扩大,相继出现了各种类型的计算机,包括小型计算机、中型计算机、大型计算机、巨型计算机以及微型计算机等等,它们的规模不同,性能和用途各异,但就其基本结构而言,都是冯·诺伊曼计算机结构的延续和发展。

## 二、计算机的基本组成框图及功能部件简介

计算机的基本组成框图如图 1.1 所示。

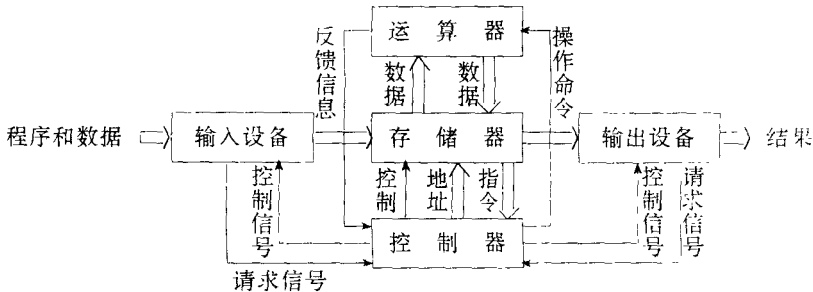


图 1.1 计算机基本组成框图

图 1.1 表明,程序和数据通过输入设备送入到存储器中;程序被启动执行时,控制器输出地址及控制信号,并从相应的存储单元中取出指令送到控制器中进行识别,分析该指令执行什么运算或操作,然后控制器根据指令含义发出操作命令,例如将某存储单元中存放的数据取出并送往运算器进行运算,再把运算结果送回存储器的指定单元中;当指定的运算或操作完成后,将结果通过输出设备送出。

通常将运算器和控制器合称中央处理器 CPU(Central Processing Unit)。CPU 和存储器一起构成计算机的主机部分,而将输入设备和输出设备称为外围设备。在微型计算机中,往往把 CPU 制作在一块大规模集成电路芯片上,称之为微处理器(Microprocessor)。

下面对组成计算机的几个主要功能部件作简要介绍。

### 1. 存储器

存储器是用来存放程序和数据信息的记忆装置。它是组成计算机的重要部件,也是使计算机能够实现“存储程序”功能的基础。

随着科学技术的进步,存储器的存储元件也在不断发展和更新。早期曾用电子管或继电器作存储元件,后来改用水银延迟线等作存储元件。从 1953 年开始使用磁芯(一种用专门的磁性材料制成的小圆环,某一磁化方向代表“0”,相反的磁化方向代表“1”)作存储元件,并使磁芯存储器在一段较长时间内成为计算机内存的主要形式。70 年代以后,大规模集成电路技术飞速发展,半导体存储器逐渐取代了磁芯存储器而成为现代计算机内存的主要形式。目前,由大规模集成电路技术制成的半导体存储器,其集成度及性能指标还在不断提高。

根据存储器和中央处理器的关系,存储器可分为主存储器(简称主存,又称内存)和外存储器(简称外存,又称辅助存储器)。主存储器通常设置在主机内部,是 CPU 可以直接对它进行

读出或写入信息(也称访问)的存储器,用来存放当前正在使用或经常要使用的程序和数据。它的容量较小,速度较快,但价格较高;外存储器设置在主机外部,用来存放相对来说不经常使用的程序和数据,在需要时与内存进行成批交换。外存的特点是存储容量大,价格较低,但存取速度较慢。外存通常由磁表面记录介质构成,如磁盘、磁带等;现在已经出现采用激光技术的光盘作大容量外存储器。需说明的是,从计算机的整体来看,磁盘、磁带等存储器是属于存储系统的一部份;但从主机的角度看,它们又属于外围设备的范畴。

主存储器通常由存储体和有关的逻辑控制电路组成。存储体是由存储元件(如磁心、半导体电路等)组成的一个存储数据信息的阵列。存储体中存放着程序和数据信息,而要对这些信息进行存取,必须通过有关的控制逻辑电路才能实现。存储体被划分为若干个存储单元,每个单元存放一串二进制信息,也称存储单元的内容。为了便于存取,每个存储单元有一个对应的编号,称为存储单元的地址。对于计算机的初学者,需注意的是存储单元的“地址”与“内容”的区别。常有人将此对应地比喻成旅馆的“房间号”与“房间里住的人”,也有一定的道理。当CPU要访问某个存储单元时,必须首先给出地址,送入存储器的地址寄存器(MAR),然后经译码电路选取相应的存储单元。从存储单元读出的信息(内容)先送入存储器的数据寄存器(MDR),再传送给目的部件;写入存储器的信息也要先送至存储器的数据寄存器中,再依据给定的地址把数据写入到相应存储器单元中。

另外,为了对存储器进行读、写操作,控制器除了要给出地址外,还要给出启动读、写操作的控制信号。这些控制信号到底何时发出,要由机器的操作时序决定。

## 2. 运算器

运算器是执行算术运算(加、减、乘、除等)和逻辑运算(“与”、“或”、“非”等)的部件,所以又称之为算术逻辑部件 ALU(Arithmetic Logic Unit)。它除核心部件加法器外,还有一个能在运算开始时提供一个操作数,并在运算结束时存放运算结果的累加寄存器(Accumulator)、通用寄存器组以及有关的输入输出控制逻辑电路组成。功能较强计算机的运算器还具有专门的乘除法部件与浮点运算部件。

## 3. 控制器

控制器是指挥和控制计算机各部件协调工作的功能部件。它从存储器中逐条取出指令,翻译指令代码,并产生各种控制信号以指挥整个计算机有条不紊地工作,一步一步地完成指令序列所规定的任务。同时控制器还要接收运算器、存储器以及输入输出设备的操作状况的反馈信息,以决定下一步的工作任务。所以控制器是整个计算机的操作控制中枢,它依据程序指令决定计算机在什么时间、根据什么条件去做什么工作。

为了让各种操作能按照一定的时间关系有序地进行,计算机内设有一套时序信号,给出时间标志。计算机的各个功能部件按照统一的时钟或节拍信号,一个节拍一个节拍地快速而有秩序地完成各种操作任务。通常将一条指令的整个执行时间定义为一个指令周期(Instruction Cycle);每个指令周期再划分为几个机器周期(Machine Cycle);每个机器周期又分为几个时钟周期。时钟周期是机器操作的最小时间单位,它由机器的主频来决定。

我们将最基本的不可再分的简单操作叫做“微操作”,控制微操作的命令信号叫“微命令”,它是比“指令”更基本、更小的操作命令,例如开启某个控制电位,清除某寄存器或将数据输入某个寄存器等。通常一条指令的执行就是通过一串微命令的执行来实现的。控制器的基本任务就是根据各种指令的要求,综合有关的逻辑条件和时间条件产生相应的微命令。

按照微命令形成方式的不同,控制器的结构可分为两种:组合逻辑控制器和微程序控制器。组合逻辑控制器是早期计算机控制器的组成形式。它是直接由组合逻辑电路产生微操作控制信号,因而其操作速度较快,但相应的控制逻辑电路十分庞杂,给设计、调试和检测都带来不便。这种形式的控制器设计完毕后若想修改,则更为困难。微程序控制器是采用微程序设计技术(Microprogramming)实现的。它是将指令执行所需的微命令编制成微程序,并事先存放在控制存储器(一般为只读存储器)中,修改该只读存储器的内容即可改变计算机的指令系统。它与组合逻辑控制器相比,具有规整性和灵活性的突出优点,但微程序控制器每执行一条指令都要启动控制存储器中的一串微指令(即一段微程序),因此指令的执行速度相对于组合逻辑控制器来说要慢。微程序控制的概念最早由英国剑桥大学的威尔克斯(Wilkes)于1951年提出,并将这种思想用于计算机控制器的设计。它实质上是用软件的方法来产生和组织微操作控制信号,用存储逻辑控制代替组合逻辑控制。

#### 4. 输入设备

输入设备的任务是用来输入操作者或其他设备提供的原始信息,并把它转变为计算机能够识别的信息,送到计算机内部进行处理。传统的输入设备有键盘、卡片阅读机、纸带输入机等。新型的输入设备种类很多,如光字符阅读机、光笔、鼠标器、图形输入器、汉字输入设备、视频摄像机等。

#### 5. 输出设备

输出设备的任务是将计算机的处理结果以人或其他设备能够识别和接受的形式(如文字、图像、声音等)输送出来。常用的输出设备有打印机、显示器、绘图仪等。

现在人们常见的各种计算机终端设备,把键盘和显示器配置在一起,它实际上是输入设备(键盘)和输出设备(显示器)的组合。

## 第二节 计算机的工作流程

### 一、指令与程序

指令是用来指挥和控制计算机执行某种操作的命令。通常,一条指令包括两个基本组成部分,即操作码部分和地址码部分。其组成格式如下所示:



其中操作码用来指出操作性质,如加法运算、减法运算、移位操作等;地址码用来表明操作数(即参与运算的数)或运算结果存放的地址。

一台计算机通常有几十种甚至上百种基本指令。我们把一台计算机所能识别和执行的全部指令称为该机的指令系统。指令系统是反映计算机的基本功能及工作效率的重要标志。它是计算机的使用者编制程序的基本依据,也是计算机系统结构设计的出发点。

指令的操作码和地址码均分别用二进制码来表示。它们各自所占的二进制位数决定了指令的操作类型的多少及操作数地址范围的大小。例如,若一个计算机的指令格式中操作码占6位(bit),则该计算机一共可以有64种( $2^6=64$ )不同操作性质的指令。不同的指令对应着不同的二进制操作码。另外,我们已经知道,要从主存中存取操作数,必须先给出地址码,而主存的地址码也是以二进制码表示的。例如,若主存容量为16384个单元,那么至少要用14位二进制

码来表示它的地址( $2^{14}=16384$ )。显然,主存容量越大,为表示它的地址所需要的二进制码位数也越多。也就是说,操作数地址范围越大,则指令中地址码的位数也应越多。

指令的操作码部分和地址码部分的二进制编码排列在一起,就组成了一条指令。从形式上看,它和二进制表示的数据并无区别,但它们的含义和功能是根本不同的。指令的这种二进制表示方法(指令的数码化),使计算机能够把程序像数据一样存放在存储器中。这就是“存储程序”计算机的重要特点。

计算机能够方便地识别和执行存放在存储器中的二进制代码指令。但对于计算机的使用者来说,书写、阅读、记忆以及修改这种表示形式的指令确十分不便,而且是件十分乏味且容易出错的事情。因此,人们通常使用一些助记符来代替它,例如用 ADD 表示加法,用 SUB 表示减法,用 MOV 表示传送等。这样,每条指令有明显的特征,易于理解和使用,也不易出错。

为了让计算机求解一个数学问题,或者做一件复杂的工作,总是先要把解决问题的过程分解为若干步骤,然后用相应的指令序列,按照一定的顺序去控制计算机完成这一工作。这样的指令序列就称为程序。通常把用二进制代码形式组成的指令序列称为机器语言程序,又称目标程序。它是计算机能够直接识别和运行的程序;而把用助记符形式组成的指令序列称为汇编语言程序或符号程序。显然,符号程序比二进制代码程序易读、易写,也便于检查和交流。但是,机器是不能直接识别符号程序的,还必须将其翻译为二进制代码程序,才能被计算机直接识别和执行。

## 二、计算机的基本工作流程

### 1. 模型计算机结构

一个实际的计算机结构,往往比较复杂。用它来说明计算机的基本工作过程,会陷入许多繁琐的细节之中。因此,我们首先从一个经过简化的模型机入手,用以扼要说明计算机是怎样进行工作的。模型计算机的结构如图 1.2 所示。

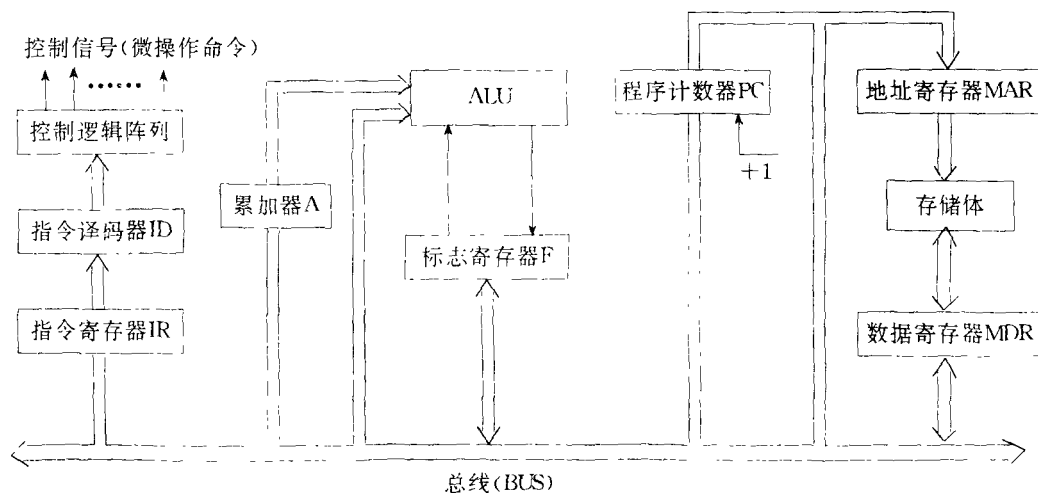


图 1.2 模型计算机结构

图 1.2 的右边为存储器部分,如前所述,它用于存放指令和数据;左边则属于 CPU 部分,它又包括运算器和控制器两个组成部分,实现指令的分析、执行以及数据的运算和处理等功

能;当然,对于一个完整的计算机结构,还应有输入输出设备及接口电路等部分,此处省略未画;图中的总线(BUS)是各部件间传送控制和数据信息的公共通道。

CPU 中有几个最基本的功能部件,对于各种结构形式的计算机来说,都是必不可少的。下面分别予以说明。

#### (1) 程序计数器 PC(Program Counter)

程序计数器也称指令计数器,用来指出计算机要执行的指令所在存储单元的地址,具有自动增量计数的功能。

我们已经知道,程序是由指令序列所组成,指令序列被存放于存储器中,要从存储器中取出指令,必须首先给出指令所在存储单元的地址。当程序被执行时,CPU 总是把 PC 的内容作为地址去访问存储器,从指定的存储单元中取出一条指令并加以译码和执行。与此同时,PC 的内容必须自动地转换成下一条指令的地址,为取出下一条指令做好准备。一般情况下,指令是按顺序一条接一条执行的,指令所在存储单元的地址也是按顺序排列的,所以在这种情况下,每当取出一条指令,PC 就启动增量修改,给出下一条指令的地址,以便使程序顺序往下执行;但是有时会出现指令不是按顺序执行即出现程序“转移”的情况,此时 CPU 就把一个新的地址(即转移地址)送往 PC,下一条指令就按这一新的地址从存储器中取出并加以执行,从而使程序的执行由一个程序段转向另一个程序段。在计算机的指令系统中,专门设有一些转移指令,用来实现程序在特定情况下进行转移。通过后续章节的学习,我们将会实际看到转移指令对于计算机进行逻辑判断和自动重复计算都是很重要的。

#### (2) 指令寄存器 IR(Instruction Register)

它保存着计算机当前正在执行或即将执行的指令。

#### (3) 指令译码器 ID(Instruction Decoder)

它用来对指令进行译码,操作码将被转换成相应的各种微操作命令,地址码被转变为操作数的真正地址。

#### (4) 控制逻辑阵列

由它产生一系列微操作命令信号。当微操作的条件(如指令的操作性质、各功能部件送来的“反馈信息”、工作节拍信号等)满足时,就发出相应的微操作命令,以控制各个部件的微操作。

#### (5) 累加器 A

它是一个在运算前存放操作数而在运算结束时存放运算结果的寄存器。它也用于 CPU 与存储器和 I/O 接口电路间的数据传送。

#### (6) 算术逻辑部件 ALU

它是用来进行算术运算与逻辑运算的部件。

#### (7) 标志寄存器 F

它是用来反映和保存运算的部分结果(例如结果为零、结果为负、进位、溢出等)及 CPU 的某种内部控制状态(例如是否允许中断等)。通常称前者为状态标志,后者为控制标志。

### 2. 指令的执行过程

如前所述,计算机要执行一条指令,先要从存储器中把它取出来,经过译码分析之后,再去执行该指令所规定的操作。所以,归结起来,一条指令的执行过程可以分为三个基本阶段或过程,即取指令、分析指令和执行指令。下面围绕这三个基本阶段来说明计算机执行指令的基本