



高等学校计算机规划教材

汇编语言实用教程

——基于Win32编程

■ 秦 莲 姬伟峰 主编

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

高等学校计算机规划教材

汇编语言实用教程

——基于 Win32 编程

秦 莲 姬伟峰 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统介绍基于 80x86 的汇编语言程序设计方法和实用技术。主要内容包括：80x86 指令系统、寻址方式、宏指令与伪指令、汇编语言格式与程序结构、分支程序设计、循环程序设计、子程序设计、宏汇编技术、系统功能调用与使用方法、I/O 程序设计方法与中断程序设计。详细介绍 Win32 汇编编程技术，以及 Win32 汇编语言与 C/C++ 语言混合编程方法，使读者对 Win32 汇编编程有初步认识。本书给出了大量实例，并按知识点设计了 21 个实验，各章均附有习题，并免费提供教学课件、程序源代码、习题参考答案。

本书可作为高等学校相关专业本科生汇编语言课程的教科书，以及相关领域工程技术人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目(CIP)数据

汇编语言实用教程：基于 Win32 编程 / 秦莲，姬伟峰主编. —北京：电子工业出版社，2010.8

高等学校计算机规划教材

ISBN 978-7-121-11322-2

I. ①汇… II. ①秦…②姬… III. ①汇编语言—程序设计—高等学校—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字(2010)第 131903 号

策划编辑：史鹏举

责任编辑：史鹏举 特约编辑：张慧

印 刷：北京丰源印刷厂

装 订：涿州市桃园装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：17.5 字数：448 千字

印 次：2010 年 8 月第 1 次印刷

定 价：29.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前 言

汇编语言是用户直接访问计算机系统最快、最有效的一种编程语言。使用汇编语言编写程序能够充分发挥计算机硬件系统的功能，具有占用存储空间少、运行速度快，代码质量高等优点。那些需要对计算机硬件进行控制，或者对运行时间和效率有要求的系统软件及应用软件，通常都是用汇编语言编写的。因此，汇编语言程序设计始终是高校计算机及相关学科的主干课程之一。

面对微型计算机技术的迅猛发展和操作系统的更新换代，本书从便于教学和注重实际应用出发，在内容编排上既兼顾了以传统的 Intel 8086/8088 为代表的 16 位汇编程序设计，同时也介绍了以 80386 为代表的 32 位汇编程序设计、Win32 汇编技术，以及 32 位汇编与 C/C++ 混合编程方法。

本书深入讨论各种实际应用问题和解决问题的方法，给出了大量实例，并按知识点设计了相关实验内容。各章均附有习题，便于学生课后练习。此外，考虑到目前大部分应用系统都是基于 Windows 系统之上的实际情况，传统的基于 DOS 的应用平台已不能满足现实应用的需要。因此，在内容安排上详细介绍了 Win32 汇编编程技术，以及 Win32 汇编语言与 C/C++ 语言的混合编程方法，使读者能够对 Win32 汇编编程有一个初步的认识。

本书共 9 章，内容安排如下。

第 1 章：介绍汇编语言程序的基础知识，包括微型计算机系统的组成与发展概况、数据信息的表示及运算方法。

第 2 章：讨论 Intel 8086 的内部结构和外部引脚功能，对其内部寄存器组的设置与功能进行了详细说明。针对 8086/8088 微处理器，详细介绍了指令系统和寻址方式，对各类指令的指令格式、指令功能和使用方法进行重点阐述。

第 3 章：系统介绍 8086 汇编语言的语句格式、汇编程序结构及上机调试方法。阐述了常用伪指令、宏指令的使用方法。

第 4 章：针对汇编语言的特点，详细介绍汇编语言程序的设计方法，重点讨论顺序结构、分支结构、循环结构、子程序及模块化程序设计技术，并结合大量实例对汇编语言程序的编程方法进行详尽的说明。

第 5 章：阐述 I/O 程序和中断程序的设计方法，介绍 I/O 的基本概念、I/O 控制方式、I/O 指令，叙述中断的概念及其工作过程，列举计算机系统中若干常用中断及其功能调用方法，详细介绍 I/O 程序的编程设计方法，并给出典型 I/O 程序的应用实例。

第 6 章：讨论 Intel 80386 CPU 体系结构与 32 位汇编，介绍 32 位 CPU 的系统结构、寄存器组、工作方式、寻址方式及扩展指令。

第 7 章：介绍汇编语言与 C/C++ 混合编程技术，以及高级语言与汇编语言的混合编程方式和实际应用，并给出了设计实例。

第 8 章：讨论基于 Win32 平台，如何利用简单实用、功能强大的可视化开发环境，借助

API 编写 32 位 Windows 汇编程序，既可以充分利用 Windows 的高级特性，又可以得到短小精悍的可执行文件。

第 9 章：根据全书重要知识点，系统安排了本课程的实验内容。

本书配有电子课件、程序源代码、习题参考答案，需要者可从华信教育资源网 <http://www.hxedu.com.cn> 免费注册下载。

本书由秦莲、姬伟峰主编，参加本书编写工作的还有殷肖川、张月玲、孙鹏、赵雪岩、李胜先、陈桂茸、郭捷、李玉、文玉峰、闫小梅等。

由于编者水平所限，书中难免存在错误和不妥之处，敬请广大读者批评指正。

咨询、意见和建议可反馈至本书责任编辑邮箱：shipj@phei.com.cn

编 者

目 录

第 1 章 基础知识	(1)
1.1 微型计算机概述	(1)
1.2 微型计算机系统组成	(3)
1.3 程序设计语言	(7)
1.4 微型计算机运算基础	(9)
习题 1	(17)
第 2 章 微处理器与指令系统	(19)
2.1 8086 微处理器	(19)
2.2 8086 寻址方式	(24)
2.3 8086 指令系统	(28)
习题 2	(47)
第 3 章 8086 汇编语言	(50)
3.1 汇编语言源程序格式	(50)
3.2 伪指令语句	(51)
3.3 汇编语言程序的调试与运行	(65)
习题 3	(72)
第 4 章 汇编程序设计	(74)
4.1 程序设计方法	(74)
4.2 顺序程序设计	(74)
4.3 分支程序设计	(76)
4.4 循环程序设计	(84)
4.5 子程序设计	(92)
4.6 模块化程序设计	(103)
习题 4	(105)
第 5 章 输入/输出与中断	(107)
5.1 I/O 概述	(107)
5.2 简单 I/O 程序举例	(111)
5.3 中断系统	(113)
5.4 中断服务程序设计	(117)
5.5 DOS 与 BIOS 功能调用	(121)
5.6 驻留中断服务程序设计	(139)
习题 5	(141)

第 6 章 32 位 80x86 汇编	(143)
6.1 80386 系统结构	(143)
6.2 80x86 CPU 的工作方式	(147)
6.3 32 位扩展指令	(151)
6.4 32 位程序设计	(157)
习题 6	(171)
第 7 章 汇编语言与 C/C++ 混合编程	(172)
7.1 混合编程	(172)
7.2 C/C++ 的嵌入式汇编	(172)
7.3 C/C++ 调用汇编模块	(183)
习题 7	(189)
第 8 章 Win32 程序设计	(191)
8.1 汇编语言 Win32 程序框架	(191)
8.2 资源文件的使用	(197)
8.3 Win32 程序设计实例	(201)
习题 8	(206)
第 9 章 课程实验	(207)
9.1 顺序程序实验	(207)
9.2 分支程序实验	(210)
9.3 循环程序实验	(215)
9.4 子程序实验	(221)
9.5 模块化程序实验	(231)
9.6 输入/输出与中断程序实验	(236)
9.7 32 位 80x86 汇编语言程序设计	(245)
9.8 汇编与 C/C++ 混合编程实验	(247)
9.9 Win32 汇编程序实验	(251)
附录 A ASCII 码表	(261)
附录 B 常用指令及其执行周期数	(262)
附录 C DEBUG 命令表	(273)
参考文献	(274)

第1章 基础知识

微型计算机系统分为硬件和软件两大部分，硬件(Hardware)是微型计算机工作的物理实体，微型计算机系统硬件结构由微处理器、存储器、I/O 接口等部分组成，了解它们的内部结构和工作方式将有助于学习后续内容；软件(Software)是为了运行、管理和维护微型计算机而编写的各种程序的集合。汇编语言以助记符形式表示微型计算机指令，每条指令对应着微型计算机硬件的一个具体操作，利用汇编语言编写的程序与微型计算机硬件密切相关，程序员可直接对处理器内的寄存器、主存储器的存储单元、外设的端口等进行操作，从而能够有效地控制微型计算机的各个部件。本章重点介绍学习汇编语言程序需要掌握的基础知识，内容包括：微型计算机产生与发展；微型计算机系统组成；程序设计语言；原码、反码、补码的定义与运算规则；编码的使用与运算规则；逻辑与、或、非、异或运算规则等。

1.1 微型计算机概述

自从 1946 年世界上第一台数字电子计算机 ENIAC(Electronic Numerical Integrator And Calculator)诞生以来，计算机的发展突飞猛进，短短几十年中，已经历了电子管计算机、晶体管计算机、集成电路计算机、大规模及超大规模集成电路计算机的发展历程，目前智能型计算机正在加快研制中。作为计算机的一个重要分支，微型计算机诞生于 20 世纪 70 年代初。微型计算机(Microcomputer)与其他大、中、小型计算机的区别主要在于，其中央处理器 CPU(Central Processing Unit)采用了大规模、超大规模集成电路技术，而其他类型计算机的 CPU 则由相当多的分立元件电路或集成电路组成。

1.1.1 微型计算机的发展与应用

微处理器的发展决定着微型计算机的更新换代，微处理器的集成度几乎每 2 年增加一倍，产品每 2~4 年更新换代一次，现已进入第 7 代。各代的划分通常以 MPU 的字长和速度为主要依据。

第一代 4 位微处理器以 Intel 公司的 4004 为代表，这种微处理器虽然简单，运算能力不强，速度不高，但这种微处理器的问世标志着计算机的发展进入了一个新纪元；第二代 8 位微处理器的典型产品有 Intel 公司的 Intel 8008/8080，Motorola 公司的 MC 6800，Zilog 公司的 Z80 等；第三代 16 位微处理器的典型产品有 Intel 公司的 Intel 8086/8088，Motorola 公司的 MC 68000/68010，Zilog 公司的 Z8000 等；第四代 32 位微处理器的典型产品有 Intel 公司的 Intel 80386/80486，Motorola 公司的 MC 68020/68030 等；第五代 64 位微处理器则以 Intel 公司先后推出的 Pentium/Pentium Pro，AMD 公司的 K6/K6/K7，Cgrix 公司的 6X86/MediaGX/6X86M，IBM、Apple、Motorola 三大公司共同开发的 Power PC 等为代表产品；第六代微处理器内含创新的超线程技术，是每秒执行 30 亿个运算周期的 CPU，代表产品为 Pentium 4 Xeno；第七代代表产品 Pentium D 是首个双核处理器芯片，揭开了 x86 微处理器多核时代。目前，多核微处理器发展迅猛。

随着微型计算机的发展,其应用范围不断扩大,归纳起来,应用主要包含以下 5 个领域。

(1) 数据处理领域

在科学研究、工程设计和经济管理领域,存在大量复杂的数学计算问题,利用微型计算机进行计算可快速得到较理想的结果。

(2) 过程控制领域

在农业、国防、交通等领域,利用微型计算机对生产和试验过程进行自动实时监测、控制和管理,可提高效率、提高质量、降低成本和缩短周期。

(3) 信息管理领域

采用目前迅猛发展的计算机网络技术,可实现信息管理的自动化和办公自动化,常用于财务管理、人事档案管理、情报资料管理等领域。

(4) 辅助设计与仿真领域

为了提高自动化水平,目前普遍借助计算机进行设计,即计算机辅助设计(Computer Aided Design),还有计算机辅助测试(Computer Aided Test)、计算机辅助制造(Computer Aided Manufacture)、计算机集成制造系统(Computer Aided and Manufacture System)、计算机辅助教学(Computer Aided Instruction)等。

(5) 人工智能领域

用微型计算机系统来模拟人类某些智能行为,包括声音、图像、文字等的模式识别,自然语言理解等。

1.1.2 微型计算机主要性能指标

衡量微型计算机系统的主要性能指标与其他大、中、小型计算机系统的衡量指标基本相同,主要有字长、存储器容量、运算速度、外设扩展及软件配置等指标。

(1) 字长

字长是计算机内部一次可以处理的二进制数码的位数。一般一台计算机的字长取决于它的 CPU 内部通用寄存器的位数、内部存储器的位数,以及算术逻辑运算单元(ALU)的位数和数据总线的宽度。字长越长,一个字所能表示的数据精度就越高,在完成同样精度的运算时,其运算速度就越快。例如,8086 的数据总线为 16 位,字长为 16,称为 16 位机;80486 的数据总线为 32 位,字长为 32,称为 32 位机。

(2) 存储器容量

存储器容量是衡量计算机存储二进制信息量大小的一个重要指标,每 8 bit 二进制数表示一字节。微型计算机一般以字节(B)为单位表示存储器容量,通常将 $2^{10} \text{B} = 1024 \text{B}$ 简称为 1 KB;1024 KB 简称为 1 MB;1024 MB 简称为 1 GB;1024 GB 简称为 1 TB。

存储器容量包括内存容量和外存容量。内存容量又分为最大容量和实际装机容量。最大容量由 CPU 的地址总线的位数决定,如 8080 CPU 的地址总线为 16 位,其内存最大容量为 64 KB;8086 CPU 的地址总线为 20 位,其内存最大容量为 1 MB;80286 CPU 的地址总线为 24 位,其内存最大容量为 16 MB;80386/80486 CPU 的地址总线为 32 位,其内存最大容量为 4 GB。

(3) 运算速度

计算机的运算速度一般用每秒钟所能执行的指令条数表示。常用计算方法是:直接给出 CPU 的主频和每条指令执行所需的时钟周期,主频越高执行速度越快,主频一般以 MHz 为

单位。例如, Intel 8086 的主频为 4.77 MHz; Intel 80286 的主频为 6 MHz; Intel 80386 的主频为 16 MHz; Intel 80486 的主频为 50 MHz; Pentium 4 的主频为 2 GHz; 双核 Pentium D 的主频为 3.8 GHz。

(4) 外设扩展能力

主要指计算机系统配接各种外部设备的可能性、灵活性和适应性。一台计算机允许配接多少外部设备, 对于系统接口和软件研制都有重大影响。在微型计算机系统中, 外存容量、USB 接口、多媒体设备配置等, 都是外设配置中需要考虑的问题。

(5) 软件配置情况

软件是计算机系统必不可少的重要组成部分, 它的配置是否齐全, 直接关系到计算机性能的好坏和效率的高低。软件主要分为系统软件和应用软件, 包括操作系统(如 DOS、Windows、UNIX、Windows NT 等)、应用软件(如 C++、SQL Server、VB 等)、文字处理软件(如 Office、WPS 等)、图形处理软件(Flash、AutoCAD 等)。

1.2 微型计算机系统组成

微型计算机系统由硬件和软件两大部分组成, 硬件(Hardware)是微机工作的物理实体; 软件(Software)则是为了运行、管理和维护微机而编写的各种程序的集合。

1.2.1 微型计算机硬件系统

1. 结构特点与框图

目前各种微型计算机系统, 无论是简单的个人微型计算机系统, 还是较复杂的超级微型计算机和巨型微型计算机系统, 从硬件体系结构来看, 采用的结构基本上都是计算机的经典结构——冯·诺依曼结构。这种结构的主要特点如下。

(1) 由运算器、控制器、存储器、输入设备和输出设备五大部分组成。

(2) 数据和程序以二进制代码形式不加区别地存放在存储器中, 存放位置由地址指定, 地址码也为二进制码形式。

(3) 控制器是根据存放在存储器中的指令序列即程序来工作的, 并由一个程序计数器控制命令的执行。控制器具有逻辑判断能力, 以计算结果为基础, 选择不同的动作流程。

图 1-1 给出了具有这种结构特点的微型计算机典型硬件组成框图。各组成部分之间通过地址总线(AB)、数据总线(DB)、控制总线(CB)联系在一起, 采用总线结构, 可使微型计算机的系统结构比较方便, 并且具有更大的灵活性和更好的可扩展性、可维修性。

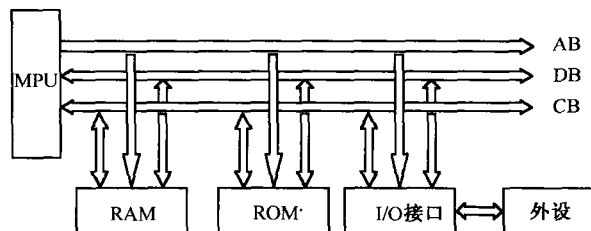


图 1-1 微型计算机典型硬件组成框图

2. 主要组成部分

(1) 微处理器(MPU)

微型计算机的 CPU 芯片通常称为 MPU (Micro Processing Unit)，它是微型计算机的运算和指挥中心。不同型号的微型计算机，其性能的差别首先在于微处理器性能的不同，而微处理器性能又与它的内部结构和硬件配置有关。但是无论哪种微处理器，其内部结构总是相同的，都有控制器、运算器、内部总线及缓存器三大部分，每部分又各由一些基本部件组成。微处理器典型结构如图 1-2 所示。

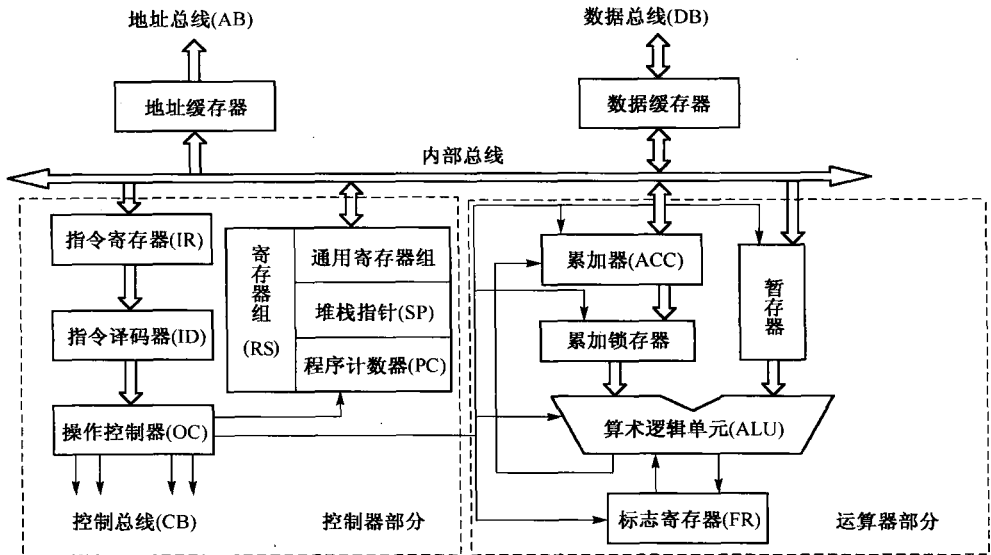


图 1-2 微处理器典型结构

各部件的功能如下。

① 算术逻辑单元 ALU (Arithmetic Logic Unit)。算术逻辑单元 ALU 是运算器的核心。它是以全加器为基础，辅之移位寄存器及相应控制逻辑组合而成的电路，在控制信号的作用下可完成加、减、乘、除四则运算和各种逻辑运算。

② 累加器 ACC、累加锁存器和暂存器。累加器 ACC (ACCumulator) 通常简称为累加器 A，它实际上是通用寄存器的一种。

③ 标志寄存器 FR (Flags Register)。用于寄存 ALU 操作结果的某些重要状态或特征，如是否溢出、是否为零、是否为负、是否有进位、是否有偶数个“1”等。

④ 寄存器组 RS (Register Set 或 Registers)。寄存器组 RS 实质上是微处理器的内部 RAM，因受芯片面积和集成度所限，其容量(即寄存器数目)不可能很多。寄存器组可分为专用寄存器和通用寄存器。专用寄存器的作用是固定的。图 1-2 中的堆栈指针 SP、程序计数器 PC、标志寄存器 FR 为专用寄存器。通用寄存器可由程序员规定其用途。

⑤ 堆栈和堆栈指针 SP (Stack Pointer)。堆栈 (Stack) 是一组在寄存器或存储器中开辟的一个特定区域。由内部寄存器构成的堆栈叫硬件堆栈；由软件在存储器区域中开辟的叫软件堆栈。目前绝大多数微型计算机中都是采用软件堆栈。

堆栈指针 (SP) 是用来指示栈顶的寄存器。SP 的初值由程序员设定，一旦设定初值后，便意味着栈底在存储器中的位置已经确定，此后 SP 的内容即栈顶位置便由 MPU 自动管理。

一般说来,对于栈底地址高、栈顶地址低的向下增长型堆栈,将数据压入堆栈时,每压入一字节,SP自动减1,向上浮动而指向新的栈顶;将数据弹出堆栈时,每弹出一字节,SP自动加1,向下浮动而指向新的栈顶。反之,对于栈底地址低、栈顶地址高的向上增长型堆栈,随着数据的压入或弹出,指针PC的浮动方向则正好相反。堆栈主要用于中断处理和过程(子程序)调用/返回,特别为多重中断与多重调用带来很大方便。

⑥ 程序计数器 PC(Program Counter)。PC用于存放下一条要执行指令的地址码,即程序中各条指令所在的地址编号。

⑦ 指令寄存器 IR(Instruction Register)、指令译码器 ID(Instruction Decoder)、操作控制器 OC(Operation Controller)。这三个部件是整个微处理器的指挥控制中心,对协调整个微型计算机有序工作极为重要。MPU根据用户预先编好的程序,依次从存储器中取出各条指令,放在指令寄存器(IR)中,通过指令译码(分析)确定应该进行什么操作,然后通过操作控制器(OC),按确定的时序,向相应的部件发出控制信号。操作控制器(OC)中主要包括节拍脉冲发生器、控制矩阵、时钟脉冲发生器、复位电路和启停电路等控制逻辑。

(2) 存储器

存储器又叫内存或主存,是微型计算机的存储和记忆部件,用以存放数据(包括原始数据、中间结果和最终结果)和程序。微型计算机的内存都是采用半导体存储器。

① 内存单元的地址和内容。内存中存放的数据和程序,从形式上看都是二进制数。内存是由若干内存单元组成的,每一个内存单元中可存放一字节(8位)的二进制信息。内存单元的总数目叫内存容量。微型计算机通过给各个内存单元规定不同地址来管理内存。这样,MPU便能识别不同的内存单元,正确地对它们进行操作。内存单元的地址和内存单元的内容是两个完全不同的概念。图1-3给出了这两个概念的示意图。

② 内存操作。MPU对内存的操作方式有读、写两种。读操作将内存单元的内容送入MPU内部,而写操作是将MPU内部信息传送到内存单元保存起来。显然,写操作的结果改变了被写单元的内容,而读操作则不改变被读单元中原有内容。

③ 内存分类。按工作方式不同,内存可分为两大类:随机存取存储器RAM(Random Access Memory)和只读存储器ROM(Read Only Memory)。RAM可以被MPU随机地读和写,所以又称为读写存储器。这种存储器用于存放用户装入的程序、数据及部分系统信息,当机器断电后,RAM中所存信息消失。ROM中的信息只能被MPU随机读取,而不能由MPU随机写入。

(3) 输入/输出(I/O)设备的接口

I/O设备是微型计算机系统的重要组成部分,微型计算机通过它与外部交换信息,完成实际工作任务。微处理器与指定外设间的信息交换是通过访问与该外设相对应的端口来实现的,如何实现对相关端口的访问,取决于这些端口的编址方式。通常有两种编址方式:存储器映像方式和隔离I/O方式。

① 存储器映像方式。这种编址方式将I/O端口和存储器单元一起编址,相当于给每个I/O端口分配一个存储器地址。主要优点如下:

a) 对I/O端口的操作与对存储器的操作完全相同,任何存储器操作指令都可用来操作I/O接口,而不必使用专用的I/O指令。

地址	内容
00000H	10110010
00001H	11000011
00002H	00001001
F0000H	00111110
FFFFFFH	00000000

图1-3 内存单元地址和内容示意图

b) 可以使外设数目或 I/O 寄存器数目几乎不受限制, 而只受总存储容量的限制, 从而增加系统的吞吐率。

c) 使微型计算机系统的读 / 写控制逻辑较简单。

主要缺点如下:

a) 占用了存储器的一部分地址空间, 使可用的内存空间减少。

b) 为了识别一个 I/O 端口, 必须对全部地址线译码, 不但增加了地址译码电路的复杂性, 而且使执行外设寻址的操作时间相对增长。

② 隔离 I/O 方式。这种编址方式将 I/O 端口和存储器做不同的处理分开编址, 即两者的地址空间是互相“隔离”的, I/O 结构不会影响存储器的地址空间。主要优点如下:

a) I/O 端口地址不占用存储器地址空间, 存储器全部地址空间都不受 I/O 寻址的影响。

b) 由于 I/O 地址线较少, 所以 I/O 端口地址译码较简单, 寻址速度较快。

c) 使用专用 I/O 指令和真正的存储器访问指令, 可使程序编写得清晰, 便于理解和检查。

主要缺点如下:

a) 专用 I/O 指令类型少, 远不如存储器访问指令丰富, 使程序设计灵活性差。使用 I/O 指令只能在累加器 A 和 I/O 端口间交换信息, 处理能力不如存储器映像方式强。

b) 要求处理器提供存储器读/写、I/O 端口读/写两组控制信号, 不仅增加了控制逻辑的复杂性, 而且对于引脚线本来就紧张的 MPU 芯片来说, 是一个负担。

(4) 三总线

总线实际上是一组导线, 是各种公共信号线的集合, 用于作为微型计算机中所有各组成部分传输信息共同使用的“公路”。

① 数据总线 DB (Data Bus)。数据总线用来传输数据信息, 是双向总线, MPU 既可通过 DB 从内存或输入设备读入数据, 又可通过 DB 将内部数据送到内存或输出设备。

② 地址总线 AB (Address Bus)。地址总线用于传送 MPU 发出的地址信息, 是单向总线。目的是指明与 MPU 交换信息的内存单元或 I/O 设备。

③ 控制总线 CB (Control Bus)。控制总线用来传送控制信号、时序信号和状态信息等, 其中有的是 MPU 向内存和外设发出的信号, 有的则是内存或外设向 MPU 发出的信息。可见, CB 中每一根线的方向是一定的、单向的, 但作为一个整体则是双向的, 所以在各种结构框图中, 凡涉及控制总线 CB 均以双向线表示。

1.2.2 微型计算机软件系统

软件包括可在微型计算机上运行的各种程序、数据及相关信息。通常可分为系统软件和应用软件。

1. 系统软件

系统软件也称为系统程序, 是完成对整个系统进行调度、管理、监控、维护等功能的软件。系统软件一般包括操作系统、语言处理程序、数据库管理系统、系统服务程序等, 通常由计算机厂家提供, 是用户使用计算机时为产生、准备和执行用户程序所必须的程序。在系统软件中, 最重要的是操作系统, 操作系统负责管理整个系统的软、硬件资源, 向用户提供交互的界面, 为其他程序的运行打下基础。

2. 应用软件

应用软件也称为应用程序，是专业软件公司为解决应用领域实际问题研制开发的程序，或由用户根据各自需要编写的程序。应用软件通常需要系统软件的支持才能在硬件系统上有效运行。计算机系统具有多种多样的应用软件，常用的软件有文本编辑软件、文字处理软件、网页制作软件、财务管理软件、动漫制作软件和系统仿真软件等。

1.3 程序设计语言

程序设计语言通常简称为编程语言，是一组用来定义计算机程序的语法规则。计算机语言能够让程序员准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。根据程序设计语言发展的历程，可将其大致分为4类，即机器语言、汇编语言、高级语言和4GL语言。

1. 机器语言

计算机能够直接识别的是二进制数0和1组成的代码。机器指令(Instruction)就是二进制编码指令，一条机器指令控制计算机完成一个操作。每种处理器都有各自的机器指令集，某处理器支持的所有指令的集合就是该处理器的指令系统(Instruction Set)。指令集及使用它编写程序的规则称为机器语言(Machine Language)。用机器语言形成的程序是计算机唯一能够直接识别并执行的程序，而用其他程序设计语言编写的程序必须经过翻译，变换成机器语言程序，所以，机器语言程序常被称为目标程序(或目的程序)。

机器指令一般由操作码(Opcod)和操作数(Operand)构成。操作码是表明操作性质的代码，操作数则表明参加操作的数或数所在的地址。一条机器指令是一组二进制代码，一个机器语言程序就是一段二进制代码序列。

用机器语言编写程序的最大缺点是难以理解，因而极易出错，也难以发现错误。所以，只是在计算机发展的早期或不得已的情况下，才用机器语言编写程序。现在，除了有时在程序某处需要直接采用机器指令填充外，几乎没有人采用机器语言编写程序了。

2. 汇编语言

为了克服机器语言的缺点，人们采用便于记忆、并能描述指令功能的符号来表达机器指令。表示指令操作码的符号称为指令助记符，简称助记符。助记符一般是表明指令功能的英语单词或其缩写。指令操作数同样也可以用易于记忆的符号表示。

用助记符表示的指令就是汇编格式指令。汇编格式指令及使用它们编写程序的规则就形成汇编语言(Assembly Language)。用汇编语言编写的程序就是汇编语言程序，或称汇编语言源程序。汇编语言是一种符号语言，它用助记符表示操作码，比机器语言容易理解和掌握，也容易调试和维护。但是，汇编语言源程序要翻译成机器语言程序才可以由处理器执行，这个翻译的过程称为“汇编”，完成汇编工作的程序就是汇编程序(Assembler)。

汇编语言的主要优点是可以直接控制计算机硬件部件，可以编写在“时间”和“空间”两方面最有效的程序。这些优点使得汇编语言在程序设计中占有重要的位置，是不可取代的。汇编语言的特点明显，它与处理器密切相关，要求程序员比较熟悉计算机硬件系统、考虑许多细节问题，从而导致编写程序烦琐，调试、维护、交流和移植困难。因此，有时可以采用

高级语言和汇编语言混合编程的办法，互相取长补短，更好地解决实际问题。汇编语言主要应用场合有以下几种。

(1) 程序要有较快的执行时间，或者只能占用较小的存储容量。例如，操作系统的核心程序段，实时控制系统的软件，智能仪器仪表的控制程序等。

(2) 程序与计算机硬件密切相关，程序要直接实时、有效地控制硬件。例如，I/O 接口电路的初始化程序段，外部设备的底层驱动程序等。

(3) 大型软件需要提高性能、优化处理的部分。例如，计算机系统频繁调用的子程序、动态链接库等。

(4) 没有合适的高级语言或只能采用汇编语言的情况。例如，开发最新的处理器程序时，暂时没有支持新指令的编译程序。

(5) 汇编语言还有许多实际应用。例如，分析具体系统尤其是该系统的底层软件、加密/解密软件、分析和防治计算机病毒等。

3. 高级语言

汇编语言虽然较机器语言直观一些，但仍然烦琐难记。于是在 20 世纪 50 年代，人们研制出了高级程序设计语言(High-level Programming Language)。高级语言比较接近于人类自然语言的语法习惯及数学表达形式，它与具体的计算机硬件无关，更容易被广大计算机工作者掌握和使用。利用高级语言，即使一般的计算机用户也可以编写软件，而不必懂得计算机的结构和工作原理。当然，用高级语言编写的源程序不会被机器直接执行，而需经过编译或解释程序的翻译才可变为机器语言程序。

高级语言简单、易学，而汇编语言复杂、难懂，是否还有必要再采用汇编语言呢？这需要比较一下汇编语言和高级语言的特点。汇编语言与处理器密切相关。每种处理器都有自己的指令系统，相应的汇编语言各不相同。所以，汇编语言程序的通用性、可移植性较差。相对来说，高级语言与具体计算机无关，高级语言程序可以在多种计算机上编译执行。汇编语言功能有限，又涉及寄存器、主存单元等硬件细节，所以编写程序比较烦琐，调试起来也比较困难。高级语言提供了强大的功能，采取类似自然语言的语法，所以容易被掌握和应用，也不必关心诸如标志、堆栈等琐碎问题。汇编语言本质上就是机器语言，它可直接地、有效地控制计算机硬件，因而容易产生运行速度快、指令序列小的高效率目标程序。高级语言不易直接控制计算机的各种操作，编译程序产生的目标程序往往比较庞大，程序难以优化，所以运行速度较慢。

4. 4GL 语言

4GL 语言以数据库管理系统所提供的功能为核心，进一步构造了开发高层软件系统的开发环境，如报表生成、多窗口表格设计、菜单生成系统、图形图像处理系统和决策支持系统，为用户提供了一个良好的应用开发环境。4GL 语言兼有过程性和非过程性双重特性，程序员规定条件和相应的动作，即过程性的部分；并且指出想要的结果，即非过程部分。4GL 语言提供了功能强大的非过程化问题定义手段，用户只需告知系统做什么，而无需说明怎么做，因此可大大提高软件生产率。4GL 语言的出现，将语言的抽象层次又提高到一个新的高度，它在更高级抽象的层次上采用不同的语法表示程序结构和数据结构，而不需要规定算法的细节。

1.4 微型计算机运算基础

1.4.1 进位计数制

常用进位计数制如下：

二进制 $R = 2$ $D_i = 0, 1$

八进制 $R = 8$ $D_i = 0, 1, 2, 3, 4, 5, 6, 7$

十进制 $R = 10$ $D_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

十六进制 $R = 16$ $D_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$

其中， R 表示基数， D_i 为基本符号，可见，二进制数有 2 个基本数字，逢二进位；八进制有 8 个基本数字，逢八进位；十进制数有 10 个基本数字，逢十进位；十六进制数有 16 个基本数字，6 个基本字母，逢十六进位。通常，在二进制数尾部加 B 作标识，十进制数尾部加 D 作标识或省略，八进制加 O 作标识，十六进制加 H 作标识。

【例 1-1】 分别写出 63 和 232 对应的的二进制数、八进制数、十六进制数。

$$63 = 00111111B = 77O = 3FH$$

$$232 = 11101000B = 350O = E8H$$

1.4.2 进位计数制的相互转换

由于微型计算机内部是由一系列逻辑电路组成的，因此在微型计算机内部的运算与存储都是采用二进制数，而二进制数对于人们日常的书写、记忆都是很很不方便的。虽然十进制数是人们常用的数制，但计算机又不能直接使用，所以需要各种数制进行转换。

1. 二进制数与十进制数的相互转换

(1) 十进制数转换为二进制数

整数部分：将要转换的十进制数的整数部分用 2 除，取其整数部分，直至商为零，将其整数部分按由下向上的顺序排列。小数部分：将要转换的十进制数的小数部分用乘 2 取整法，取其整数部分，直至小数部分为 0，将整数部分按由上向下的顺序排列，考虑可能有转换误差存在，可按要求保留位数，即得二进制数。

【例 1-2】 将十进制数 $X = 107.125D$ 转换为二进制数。

X 整数部分=107D，其转换步骤如下：

$$107/2=53 \quad (D_0=1)$$

$$53/2=26 \quad (D_1=1)$$

$$26/2=13 \quad (D_2=0)$$

$$13/2=6 \quad (D_3=1)$$

$$6/2=3 \quad (D_4=0)$$

$$3/2=1 \quad (D_5=1)$$

$$1/2=0 \quad (D_6=1)$$



X 小数部分 = 0.125D，其转换步骤如下：

$$0.125 \times 2 = 0.25 \quad (D_0=0)$$

$$0.25 \times 2 = 0.5 \quad (D1=0)$$

$$0.5 \times 2 = 1 \quad (D2=1)$$

$$X = 1101011.001B$$

(2) 二进制数转换为十进制数

其转换规则是将要转换的二进制数按权展开计算。

【例 1-3】 将二进制数 $X=10011001.10101B$ 转换为十进制数。

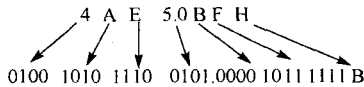
$$10011001.10101B = 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3} + 1 \times 2^{-5} = 153.65625D$$

2. 二进制数与十六进制数的相互转换

(1) 十六进制数转换为二进制数

用 4 位二进制数取代每 1 位十六进制数，即得对应的十六进制数。

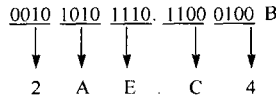
【例 1-4】 将十六进制数 $4AE5.0BFH$ 转换为二进制数。



(2) 二进制数转换为十六进制数

首先从小数点开始分别向左和向右把整数和小数部分每 4 位一组。若整数最高位的一组不足 4 位，则在其左边补 0；若小数最低位一组不足 4 位，则在其右边补 0。然后用十六进制数符取代每组的 4 位二进制数，即得对应的十六进制数。

【例 1-5】 将二进制数 $1010101110.110001B$ 转换为十六进制数。



1.4.3 二进制编码

由于微型计算机只能识别二进制数，输入的数字、字母、符号等要转化成特定的二进制码来表示，这就是二进制编码。

1. 二进制编码的十进制数(二-十进制码或 BCD 码)

在微型计算机中的十进制数用二进制编码表示，比较常用的是 8421 BCD 编码。8421 BCD 编码有 10 个不同的数字符号，逢“十”进位，即为十进制数；同时，每位十进制数用 4 位二进制编码表示，因此，称为二进制编码的十进制数，即二-十进制码或 BCD(Binary Coded Decimal)码。十进制数(0~9)与 8421 BCD 编码的对应关系如表 1-1 所示。

表 1-1 十进制数与 8421 BCD 编码

十进制数	8421 BCD 编码	十进制数	8421 BCD 编码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001