电子计算机应用系列教材

汇编语言程序设计

陆忠华　刘镜年
沈邦济　汤星辉　庄牧彬 编著

科学出版社

电子计算机应用系列教材

汇编语言程序设计

陆忠华　刘镜年
沈邦济　汤星辉　庄牧彬　编著

科学出版社

1993

(京)新登字092号

内 容 简 介

本书是电子计算机应用系列教材之一. 全书共十一章. 前三章以 Z80计算机为背景介绍了汇编语言程序设计的初步知识,后八章以 IBM PC 机为背景,深入地讨论了汇编语言程序设计的基本方法、DOS 和 BIOS 功能调用、磁盘文件管理、综合应用程序设计等.

本书内容由浅入深,从实用出发,理论联系实际,突出编程开发应用,附有大量程序实例和习题,可供高等院校非计算机专业作软件方面的教科书,也可作为教材供有关部门对广大非计算机专业的科技人员进行培训.

电子计算机应用系列教材

汇编语言程序设计

陆忠华 刘镜年
沈昌济 汤星辉 庄牧彬 编著

责任编辑 杨家福

科学出版社出版
北京东黄城根北街16号
邮政编码:100717

北京市华星计算机公司激光照排
天津市蓟县燎原印刷厂 印刷
新华书店北京发行所发行 各地新地书店经售

*

1993年6月第 一 版 开本787×1092 1/16
1993年6月第一次印刷 印张:22¾
印数:1—4500 字数:522 000

ISBN 7-03-001353-0/TP·88

定价:17.00元

电子计算机应用系列教材主持、组织编著单位

主持编著单位：

国务院电子信息系统推广应用办公室

组织编著单位：

广东、广西、上海、山东、山西、天津、云南、内蒙古、
四川、辽宁、北京、江苏、甘肃、宁夏、江西、安徽、
河北、河南、贵州、浙江、湖北、湖南、黑龙江、福建、
新疆、广州、大连、宁波、西安、沈阳、武汉、青岛、
重庆、哈尔滨、南京等35省、市、自治区、计划单列市

电子振兴
计算机领导小组办公室
科技工作

电子计算机应用系列教材联合编审委员会名单

（以姓氏笔画为序）

主编审委员：

王长胤* 苏世生 何守才 陈有祺 陈莘萌* 邹海明* 郑天健
殷志鹤 童 颋 赖翔飞 （有"*"者为常务主编）

常务编审委员：

于占涛 王一良 冯锡祺 刘大昕 朱维华 陈火旺 陈洪陶 余 俊
李 祥 苏锦祥 佟震亚 张广华 张少润 张吉生 张志浩 张建荣
钟伯刚 胡秉光 高树森 徐洁盘 曹大铸 谢玉光 谢育先 韩兆轩
韩培尧 董继润 程慧霞

编审委员：

王升亮 王伦津 王树人 王振宇 王继青 王翰虎 毛培法 叶以丰
冯鉴生 刘开瑛 刘尚威 刘国靖 刘晓融 刘德镇 孙令举 孙其梅
孙耕田 朱泳岭 许震宇 何文兴 陈凤枝 陈兴业 陈启泉 陈时锦
邱玉辉 吴宇尧 吴意生 李克洪 李迪义 李忠民 迟忠先 沈林兴
肖金声 苏松基 杨润生 闵福德 张志弘 张银明 张 勤 张福源
张翼鹏 郑玉林 郑 重 郑桂林 孟昭光 林俊伯 林钧海 周俊林
赵振玉 赵惠溥 姚卿达 段银田 钟维明 袁玉馨 唐肖光 唐楷全
徐国平 徐拾义 康继昌 高登芳 黄友谦 黄 侃 程锦松 楼朝城
潘正运 潘庆荣

秘书组：

秘 书 长：胡茂生
副秘书长：何兴能 林茂荃 易 勤 黄雄才

序

当代新技术革命的蓬勃发展，带来社会生产力新的飞跃，引起整个社会的巨大变革。电子计算机技术是新技术革命中最活跃的核心技术，在工农业生产、流通领域、国防建设和科学研究方面得到越来越广泛的应用。

党的十一届三中全会以来，我国计算机应用事业的发展是相当迅速的。到目前为止，全国装机量已突破三十万台，十六位以下微型计算机开始形成产业和市场规模，全国从事计算机科研、开发、生产、应用、经营、服务和教学的科技人员已达十多万人，与1980年相比，增长了近八倍。他们在工业、农业、商业、城建、金融、科技、文教、卫生、公安等广阔的领域中积极开发应用计算机技术，取得了优异的成绩，创造了显著的经济效益和社会效益，为开拓计算机应用的新局面作出了重要贡献。实践证明，人才是计算机开发应用的中心环节。我们必须把计算机应用人才的开发与培养放在计算机应用事业的首位，要坚持不懈地抓住人才培养这个关键。

从目前来看，我国计算机应用人才队伍虽然有了很大的发展，但是这支队伍的数量和质量还远不适应计算机应用事业发展的客观需要，复合型人才的培养与教育还没有走上规范化、制度化轨道，教材建设仍显薄弱，培训质量不高。因此，在国务院电子信息系统推广应用办公室领导、支持下，全国三十五个省、市、自治区、计划单列市计算机应用主管部门共同组织118所大学和科研单位的400多位专家、教授编写了全国第一部《电子计算机应用人才培训大纲》以及与之配套使用的电子计算机应用系列教材，在人才培训和开发方面做了一件很有意义的工作，对实现培训工作规范化、制度化将起到很好的推动作用。

《电子计算机应用人才培训大纲》和电子计算机应用系列教材贯穿了从应用出发、为应用服务，大力培养高质量、多层次、复合型 应用人才这样一条主线。大纲总结了近几年各地计算机技术培训正反两方面的经验，提出了计算机应用人才的层次结构、不同层次人才的素质要求和培养途径，制定了一套必须遵循的层次化培训办学规范，编制了适应办学规范的“课程教学大纲”。这部大纲为各地方、各部门、各单位制定人才培养规划和工作计划提供了原则依据，为科技人员、管理人员以及其他人员学习计算机技术指出了努力方向和步骤，为社会提供了考核计算机应用人才的客观尺度。“电子计算机应用系列教材”是培训大纲在教学内容上的展开与体现，是我国目前规模最大的一套计算机应用教材。教材的体系为树型结构，模块化与系统性、连贯性、完整性相兼容，教学内容注重实用性、工程性、科学性，并具有简明清晰、通俗易懂、方便教学、易于自学等特点，是一套很好的系列教材。

这部大纲和系列教材的诞生是各方面团结合作、群策群力的结果，它的公开出版和发行，对计算机应用人才的培训工作将起到积极的推动作用。希望全国各地区、各部门、各单位广泛运用这套系列教材，发挥它应有的作用，并在实践中检验、修改、补充和完善它。

通过培训教材的建设，把培训工作与贯彻国家既定的成人教育、函授教育、电视教育

和科技人员继续工程教育等制度相结合，逐步把计算机应用人才的培训工作引向规范化、制度化轨道，为培养和造就大批高素质、多层次、复合型计算机应用人才而努力奋斗，更好地推动计算机应用事业向深度和广度发展.

李祥林

1988年10月17日

前　言

汇编语言是电子数字计算机的基本语言之一，是最接近机器语言的符号语言．因此，本课程是学习计算机专业的必修课程．汇编语言程序设计广泛应用于过程控制的接口设计，也是软件开发的手段之一．因此，本课程也是计算机应用课程的基础．

本书在介绍了冯·诺依曼的程序存储思想、电子数字计算机基本工作原理及汇编语言基本概念等预备知识基础上，以8位微处理器芯片 Z80为先导，介绍了 Z80微处理器及其指令系统，汇编语言设计初步；以16位微处理器8086/8088为重点，介绍了8086/8088微处理器及其指令系统、MCS-86宏汇编语言、8086/8088汇编语言程序设计基本方法及实用汇编语言程序实例；最后结合 IBM PC 微机介绍了 IBM PC DOS 和 BIOS 的功能调用及 IBM PC 综合应用程序设计．汇编语言程序设计是一门实践性很强的课程，因此本书编写了若干实例，各章之后均附有习题，并强调上机实践，以培养学员的软件开发能力．

本书共分十一章．第一、二、三、七章由武汉钢铁学院陆忠华编写．第四、五章由武汉测绘科技大学沈邦济编写．第六、八、九章由武汉测绘科技大学刘镜年编写．第十、十一章由西北工业大学汤星辉和庄牧彬共同编写．本书全稿由陆忠华统编．山东大学郑玉林主审全稿，并提出了许多指导性意见．特在此对他表示诚挚感谢．

由于编者水平有限，编写时间仓促，书中难免有缺点和错误，欢迎读者批评指正．

目　录

第一章　汇编语言程序设计基础知识

1.1　冯·诺依曼的程序存储思想

从1946年第一台电子计算机问世以来，随着电子技术的发展，电子计算机的面貌日新月异．但有趣的是，直到目前为止，绝大多数电子计算机都是以美国普林斯顿大学的冯·诺依曼于1945年提出的一份研制报告中所阐述的计算机体系结构设计思想为基本依据的．冯·依诺曼设计思想的要点，简单说来，就是“程序存储”的思想．为了更好地了解这一点，我们不妨来看一看这一思想产生的历史背景．

第一台电子计算机ENIAC是由埃克特和莫克利于1946年在美国宾夕法尼亚大学研制成功的．此机重30t，占地165m^2，采用18000个电子管，用20个10位的十进制电子管加法器来进行加法运算，运转时耗电140kW．但由于运算器采用电子管元件，故运算速度每秒可达5000次，这在当时是史无前例的．

在ENIAC机中，程序编排是通过非常复杂的外部硬接线逻辑电路对电子管运算器的控制来实现的．编程时需要先对6000个拨盘开关进行手工定位，再用转接插线将选定的各个控制部分连接起来．数据预置在存储器中．操作命令则直接通过硬接线电路对运算器进行控制，而不是像数据那样预置在存储器中．因此，严格说来，ENIAC只是一台数字逻辑控制计算器，而不是一台数字电子计算机．

冯·诺依曼曾担任研制ENIAC的顾问，他在对ENIAC机的系统结构进行了深入研究后发现，如对拨盘开关定位及连接转插线等动作实际上也不过是一些数字信息．这些信息完全可以像数据一样存放在存储器中．这一观点也就是“程序存储”的思想．

冯·诺依曼在他的研制报告中提出了以下论点：

(1)电子计算机不应采用十进制，而应采用二进制，因采用二进制便于实现逻辑运算，且运算器的电路结构更为有效．

(2)操作命令也是一种信息，不妨碍采用二进制代码来表示．

(3)程序和数据可以用完全相同的形式存放在存储器中．

(4)程序本身也可包含数据，即程序中的每一条指令可分成操作码与操作数二部分，前者给出操作内容，后者给出数据所在存储单元地址或直接给出数据．

这些观点即所谓“程序存储”思想．在同一报告中，冯·诺依曼还提出其它一些建议，但影响最大的还是“程序存储”的观点．

在1946年的冯·诺依曼的第二份报告中，他进一步将其设计思想具体落实到IAS机的设计方案上，另外他还具体指导了该机的研制工作．该机于1952年在普林斯顿大学正式投入运转．它是一台二进制计算机，字长40位，内存容量1K，运算速度每秒2万次．该机的资料流传很广，对计算机的发展影响甚大．

1.2 电子计算机的基本工作原理

1.2.1 电子计算机的基本组成及其工作过程

冯·诺依曼型电子计算机的典型系统结构框图如图 1.1 所示.

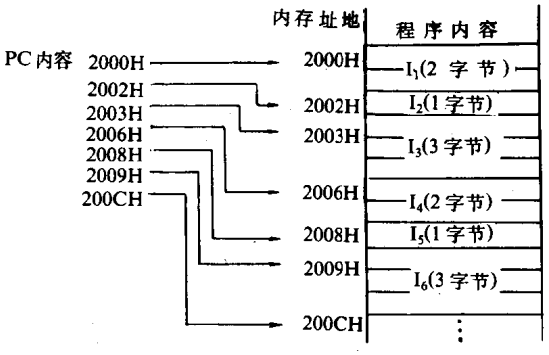

图 1.1 电子计算机系统结构框图

图中运算器用来进行各种算术运算和逻辑运算. 存储器用来存放程序指令及原始数据. 控制器用来指挥和控制整个运算过程,以使之按指令一条一条协调进行. 输入设备用来输入指令代码及原始数据. 输出设备用来显示或打印计算结果,二者均通过接口与其它功能部件进行信息交换.

运算器与控制器通常合称为中央处理器,简称 CPU. 在微型计算机中,由于大规模半导体集成技术的发展,整个中央处理器电路可集成在一个芯片上,它称为微处理器,简称 MPU,或按习惯仍称为 CPU.

CPU 通过数据总线与存储器及接口电路进行信息交换. 选中某一个指定存储单元或接口单元时,可对该单元进行数据读写操作.

存储器由许多存储单元组成. 每个存储单元有一个编号,这称为地址码. 在微型计算机中,每个存储单元通常可存放 8 位二进制信息. 习惯上将 8 位二进制信息称为一个字节. 计算机在对某一存储单元进行"读"或"写"操作时,首先均应由 CPU 通过地址总线发出与该存储单元对应的地址码,并经地址译码电路(由于地址译码电路可看作存储器的一部份,故图中未单独画出)选中指定存储单元. 只有在选中前提下,才有可能对该单元进行读写操作. 图 1.1 的地址总线的根数决定了 CPU 最多能访问多少个存储单元,即决定系统的存储空间.

为了使输入输出设备不过多地占用总线,也为了避免总线冲突,图 1.1 的输入输出设备通过接口与 CPU 进行信息交换. 同样,CPU 可通过地址总线发出地址码选中接口电路中指定的数据寄存器,以便进行读写操作.

图 1.1 的冯·诺依曼型计算机的工作过程,也就是逐条执行预置在存储器中的指令的过程. 这里,在 CPU 内部保证指令逐条执行的一个关键元件称为程序计数器,简称 PC. 图 1.1 系统的工作过程可简述如下:首先 CPU 将程序计数器 PC 的内容经地址总线送到存储器,按地址码从存储器中读出第一条指令. 由 CPU 中的译码电路分析指令的操作码

部分，确定需要运算器进行何种操作，并确定是否需对存储器及接口进行读写操作，然后由 CPU 控制部件按一定节拍通过控制总线对功能部件发出相应的控制信号时间序列，以完成指令所要求的操作内容. PC 在取出一条指令后即自动更新其内容，指向下一条指令. 若初始时 PC 内容为 N，则开机后 CPU 首先从第 N 存储单元取出指令来执行. 设第一条指令长度为 2 字节，则取出第一条指令后，PC 即自动加 2. 因此执行完第一条指令后，CPU 即自动从 N+2 单元取出第二条指令顺序执行. 同样，取出并执行第二条指令后，PC 即指向第三条指令. 依此类推，顺序执行以下各条指令，直至整个程序执行完毕. 图 1.2 给出程序运行过程中 PC 内容更新情况.

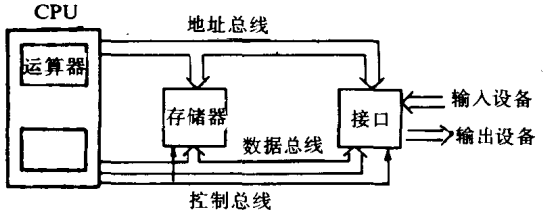

图 1.2　程序运行过程 PC 内容更新情况

1.2.2　计算机面向程序设计的编程模型

对程序设计员来说，最感兴趣的不是电子计算机的硬件结构及组成原理，而是对程序设计有用的有关信息：

(1)CPU 的指令系统.

(2)与编程有关的 CPU 内部寄存器.

(3)计算机的存储空间及分配给用户的范围.

对于在过程控制领域应用的场合，程序设计员还需要了解任务对输入输出接口的要求.

图 1.3 给出典型 8 位微处理器 Z80 的编程模型. Z80CPU 的指令系统及各 CPU 内部寄存器的功能将在本书第二章说明. 由于图 1.3 的通用寄存器及堆栈的作用在电子计算机中具有普遍的意义，因此有必要先对此两者的功能进行简要讨论.

冯·诺依曼在 1845 年的研制报告中除了提出“程序存储”的观点外，还提出了关于采用多级存储结构的建议. 他建议，为了加快数据存取速度，而又不使存储器体积过分庞大，可采用多级存储结构. 目前，大多数通用电子计算机均有通用寄存器、堆栈、主存储器及外存储器等层次结构.

通用寄存器组是运算器的一个组成部分，可用来暂存原始数据及运算的中间结果. 由于各寄存器之间的数据交换速度比寄存器与存储器之间的交换速度快得多，因此设置通用寄存器组有助于提高程序运行速度.

但在 CPU 内部通用寄存器的数目有一定限制，数目过多将使数据存取速度下降而失去其作为通用寄存器的意义. 补救的办法是在主存储器中划出一定的区域用作所谓堆栈，

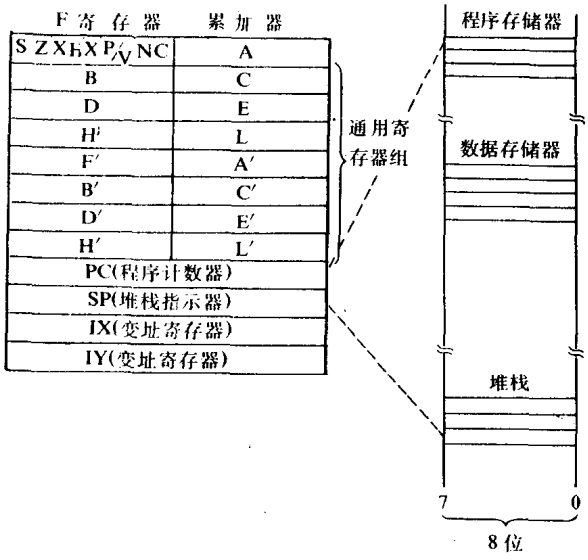

图 1.3 Z80 编程模型图

堆栈可用来保存中间运算结果，以弥补通用寄存器数量的不足. 堆栈的结构示意图如图 1.4 所示. 它的存取速度并不快，但具有按一定规律自动调整存储单元地址的功能，因而省去了通过指令调整存储单元地址的时间，因此总的说来提高了存取速度. 其所以称为堆栈，是由于其数据的存取方式系按照“后进先出”的规律进行的. 这种情况与货物仓库堆栈的情况相仿. 在货物堆栈中，先进栈的木箱放在下面，后进栈的放在上面，取货时，必须将

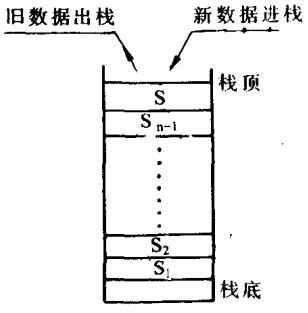

图 1.4 堆栈示意图

后进的木箱先取下来，再取下一层木箱. 同样，在图 1.4 中，数据 $S_1 \sim S_n$ 自下而上顺序进入堆栈. 出栈时的顺序则相反. 当堆栈中没有数据时，栈顶与栈底相重合. 栈底是固定的，而栈顶则是浮动的. 压入数据时，栈顶自动向上浮动，数据出栈时栈顶自动向下浮动. 数据的存取只能在栈顶进行，即位于栈顶与栈底之间的数据 $S_1 \sim S_{n-1}$ 在其上部(后进栈者)数据 S_n 移出之前是不能取出的.

为了实现堆栈顶指针的自动浮动，一般可在 CPU 内部增加一个堆栈指针寄存器 SP. 它能在控制器控制下配合堆栈操作指令自动向上或向下浮动.

堆栈除用来弥补通用寄存器数量不足外，更主要的是可用来保存调用子程序时主程序的断口地址. 这一点将在第二章进行讨论.

1.2.3 程序设计对计算机指令系统的基本要求

下面从程序设计角度讨论一下对 CPU 指令的基本要求.

在图 1.2 中,每执行完一条指令,程序计数器 PC 即自动指向下一条指令,即程序是顺序执行的. 但在实际应用中,仅仅依靠顺序程序是不够的. 实际应用中的程序,通常都要求能依照处理过程中出现的不同状态由机器自动判定对下一步作出不同的处理,这样才能体现电子计算机的"思维与判断"能力. 因此,程序指令执行的次序不一定像图 1.2 那样顺序进行,而可能出现分支,形成分支结构程序. 此外,在解算问题过程中还经常会遇到这样一类问题,它的算法需要多次重复执行相同或相似的操作步骤. 为了满足以上要求,就需要在指令系统中包含程序控制指令.

程序控制指令的主要作用,是根据当前操作结果的某些特征来决定程序下一步的走向,即决定程序计数器 PC 的值. 当程序中遇到程序控制指令且满足程序转换条件时,程序不再按顺序执行下一条指令,而是转到指定地址去执行那里的指令. 操作结果的状态特征通常又被组合在一个称为标志寄存器或 F 寄存器的 CPU 内部寄存器中(参见图 1.3),作为条件转移的依据.

由此可知,任何计算机的指令系统至少均应包括数据处理指令及程序控制指令二类,否则其用途将十分有限.

程序控制指令有转移指令及子程序调用指令二大类. 它们的进一步说明将在第二章结合 Z80 指令系统进行.

1.3 机器语言程序

程序是实现某一个算法的指令序列或语句串. 直接用机器指令(二进制或十六进制代码)编写的程序称为机器语言程序.

利用计算机求解实际问题时,首先应确定算法,必要时可将算法用程序流程图表示. 对于较复杂的程序,可先画出程序粗框图,再画出细框图. 对存储单元进行必要分配后,即可用机器指令编写出程序清单.

为了使问题简化,我们假设一台具有简化指令系统的计算机(模型计算机). 它的指令

表 1.1 模型计算机的指令系统

指令名称	操作码	操作数	操作内容	附　　注
取　数	01H	N	A←(N)	
存　数	02H	N	(N)←A	(N)代表地址为 N 的存储单元
加　法	03H	N	A←A + (N)	A 代表 A 寄存器
乘　法	04H	N	A←A * (N)	
比　较	05H	N	A−(N)	只影响标志
转　移	06H	N	有借位则 PC←N	有借位则程序转移
停　机	07H			

系统由表 1.1 给出. CPU 内部通用寄存器只有数据寄存器 A 及标志寄存器 F. 由于存储器容量较小,故存储单元可用 8 位二进制或 2 位十六进制数编码. 存储空间中也不设堆栈区. 由于有关数制与码制方面的基础已在有关课程中介绍过,故此处不再赘述.

下面给出说明机器语言编程步骤的两个例子:

例 1.1 试编制求解 $y = ax^2 + bx + c$ 的程序.

解:(1)算法设计.

令

$$y = ax^2 + bx + c$$
$$= (ax + b)x + c$$

由此可得出以下解题算法,其中每一步均可用给定指令系统中的相应指令实现.

编 号	解 题 算 法
1	取数 a
2	计算 $a * x$
3	计算 $ax+b$
4	计算 $(ax+b) * x$
5	计算 $(ax+b)x+c$
6	存放计算结果 y
7	停 机

(2)存储单元分配.

按照程序存储原理,程序、原始数据及计算结果均应安排相应的存储单元.

给定算法可用 7 条指令来实现. 除停机指令外,其它指令均带操作数,并占用 2 个字节存储单元(每个字节单元可存放一个 8 位二进制信息). 对 a、b、c、x 等 4 个原始数据及计算结果 y 应共安排 5 个存储单元,我们可将程序安排在 00H～0CH 存储单元中. 数据安排在 10H～14H 存储单元中,如下所示:

地址	存储单元
10H	a
11H	b
12H	c
13H	x
14H	y

(3)程序清单.

将各解题算法步骤均用相应的指令实现后,可得出程序如下清单:

存储单元地址	指令	数据	操内容作	说　　明
00H	01	10	A←(10H)	取数 a
02H	04	13	A←A＊(13H)	计算 $a*x$
04H	03	11	A←A＋(11H)	计算 $ax+b$
06H	04	13	A←A＊(13H)	计算$(ax+b)*x$
08H	03	12	A←A＋(12H)	计算 $y=(ax+b)x+c$
0AH	02	14	(14H)←A	存入计算结果 y
0CH	07			
10H	a		原始数据	
11H	b			
12H	c			
13H	x			
14H	y		计算结果	

例 1.2　已知

$$y=\begin{cases}1 & \text{当 } x\geqslant 5\\ 0 & \text{当 } x<5\end{cases}$$

式中 x 为正整数(<255). 试编制由 x 求解 y 的程序.

解:按题意要求,对应不同的 x 值有不同的处理方法,故需采用分支程序. 利用模型计算机的转移指令可构成分支程序.

(1)算法设计.

可画出程序流程图,如图 1.5 所示.

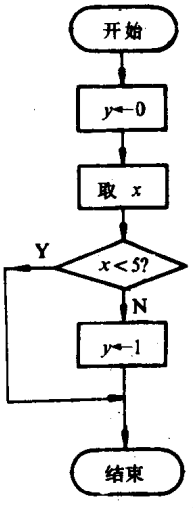

图 1.5

(2)存储单元分配.