CASL 汇编语言程序设计

周人杰 王培德 编著

ZHE JIANG UNIVERSITY PRESS • ZHE JIANG UNIVERSITY PRESS

浙江大学出版社
ZHE JIANG UNIVERSITY PRESS

CASL
汇编语言
程序设计

周人杰　王培德　编著

浙江大学出版社

(浙)新登字 10 号

内 容 简 介

本书紧扣中国计算机应用软件人员水平考试大纲，在介绍和应用 CASL 汇编语言的基础上论述程序设计方法。全书共分六章，分别是计算机硬件基础、汇编语言、程序设计基础、基本程序设计、子程序设计和水平考试例题选解。

本书可作为各级软件人员提高业务水平的参考书和参加水平考试的辅导材料，也可作为高等院校汇编语言课程的教学参考书。

CASL 汇编语言程序设计

周人杰 王培德 编著
责任编辑 任 洁

浙江大学出版社出版
浙江大学出版社电脑室排版
浙江省新华书店发行
富阳何云印刷厂印刷

开本 787×1092 1/32 印张 7.75 字数 194 千字
1991 年 9 月第 1 版 1991 年 9 月第 1 次印刷
印数 0001—3000
ISBN 7—308—00854—1/TP·060 定价：4.95 元

前　言

计算机技术的迅速发展，特别是微处理器和微型计算机的广泛应用，使得计算机已普及到当代社会生活和生产的所有领域，成为人们不可缺少的重要工具。目前，尽管计算机的应用开发已主要使用高级语言，可是汇编语言在系统软件、实时控制、过程控制、智能仪器、数据传输等领域中仍占有重要地位。因此，计算机应用开发人员必须掌握汇编语言程序设计的原理和方法。

CASL 是 1987 年日本计算机应用软件人员全国统考委员会定义的一种抽象的汇编语言。它的硬件基础是抽象的 COMET 模型机。1990 年中国计算机应用软件人员水平考试委员会规定采用 CASL 作为程序员级水平考试选考和高级程序员级水平考试必考的程序设计语言。CASL 从当代计算机汇编语言中精炼出共同的最基本部分，具有语句较少、功能单一、程序格式较为规范等特点。既方便易学，又有利于锻炼程序设计技巧。掌握了 CASL 汇编语言程序设计的方法，可以为学习和应用各种 16 位或 32 位高档微机的汇编语言程序设计打下坚实的基础。

本书紧扣中国计算机应用软件人员水平考试大纲，在介绍和应用 CASL 汇编语言的基础上论述程序设计方法，并根据 1990 年两期 CASL 学习班的讲稿修改编写而成。在编写过

程中力求由浅入深、循序渐进、多举实例，便于自学。可作为各级软件人员提高业务水平的参考书和参加水平考试的辅导材料。也可作为高等院校汇编语言课程的教学参考书。

作　者

1990 年 5 月

目　录

第一章　计算机硬件基础

电子计算机是20世纪的一项重大发明，它是科学技术和生产发展的产物，反过来又大大地促进了科技和生产的发展。电子计算机的出现被认为是第三次产业革命，促使人类进入了信息社会，信息已成为当今社会三大资源之一。

电子计算机是一种对数据进行加工处理转换为信息的机器。计算机对输入的数据进行处理，输出用户所需要的信息。计算机为了处理信息，必须具有信息的输入、贮存、处理和输出等基本功能，以及对这些基本功能的控制功能。用户为了应用计算机处理特定的问题，必须编写程序。所谓程序，是根据计算机处理特定问题的操作步骤而编制的一系列指令的集合。把程序事先存放在计算机内的存贮器中，然后启动程序，计算机便按照程序中编排好的次序自动顺序执行各条指令，不需要人工干预就能完成整个处理过程。

第一节　计算机的基本组成

电子计算机的发展已有40余年历史，经历了四代的变化，目前正在研制第五代计算机。到目前为止，各种计算机的结构都基于冯·诺依曼(*John Von Neumann*)型计算机的基本组成。如图1-1所示，由运算器、控制器、存贮器、输入设备和输出设备五大部分组成。

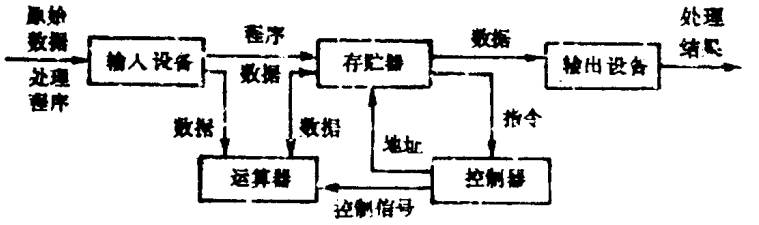

图 1-1

一、运算器

运算器是计算机加工处理数据的核心，由一个加法器和若干个寄存器组成，它从存贮器或输入设备取得数据存放在寄存器中，在控制器送来的控制信号控制下完成各种算术运算和逻辑运算，然后把运算结果送入存贮器或输出设备。在一般的计算机中，各种算术运算都转换成加法，由加法器实现。

二、控制器

控制器是计算机的控制中心。它从存贮器中读取指令（指定计算机进行某种操作的命令），根据指令的内容向各个部件发出相应的控制信号，保证各部件协调工作完成指定的操作。控制器连续按顺序读取指令，控制各部件一步一步完成各个运算步骤，从而自动完成整个运算过程。

运算器和控制器合在一起，称为中央处理器（*Central Processing Unit*），简称CPU。在现代计算机中，CPU做在一块大规模集成电路芯片中，称为微处理器。

三、输入设备

输入设备是向计算机输入程序和数据的设备。它的基本

功能是把数据和程序中的每一个字符转换成计算机能识别的二进制代码送到计算机中。常用的输入设备有键盘、磁带机、卡片输入机、磁盘机等。

四、输出设备

输出设备的功能是把计算机中的二进制信息转换成用户需要的形式(文字、图形、声音等)输出。常用的输出设备有显示器、打印机、绘图仪。此外，还有纸带穿孔机、磁带机、磁盘机等。

五、存贮器

存贮器的基本功能是保存信息。存贮器由很多单元组成，这些单元称为"存贮单元"。在计算机中，信息以8位二进制代码为一个基本单位，称为1个字节。每个存贮单元存放1个字节。存贮器包含的存贮单元数称为该存贮器的容量。微型计算机存贮器的容量从几十"K"(1K=1024)至几兆字节。为了识别每一个单元，必须对各单元编号，每个单元的编号称为该单元的地址。与存贮器有关的计算机指令，除了指定操作类型(读出、存入)外，还必须指定与操作有关的存贮单元地址。如读取某地址单元中的内容送入运算器中的某个寄存器；把某个寄存器中的内容存入某地址的存贮单元中等等。这种在计算机内部直接与运算器传送信息的存贮器又称为内存贮器或主存贮器，简称内存或主存。CPU和内存合在一起，称为主机。前面提到的磁带机和磁盘机也是存贮信息的设备，因其在主机的外边，称为外存贮器或辅存贮器，简称外存或辅存。外存与运算器没有直接的联系，信息的存取速度较低，但是容量较大，价格较低。通常内存贮器存放当前使用的数据和程序，外存则作为内存的后备，存放当前不用而需要保存备用的信

息，需要使用某部分信息时，便将其送至内存供主机使用。主机从外存读取数据时，外存相当于一个输入设备；主机向外存贮器存放信息时，外存又相当于一个输出设备。

计算机的主机、外存、输入设备和输出设备等具体设备的集合称为计算机硬件系统。系统中除主机以外的其他各种设备统称为“外部设备”，各部件之间传递数据或控制信号的通道，统称为“总线”。图 1-2 为计算机硬件系统的结构示意图。

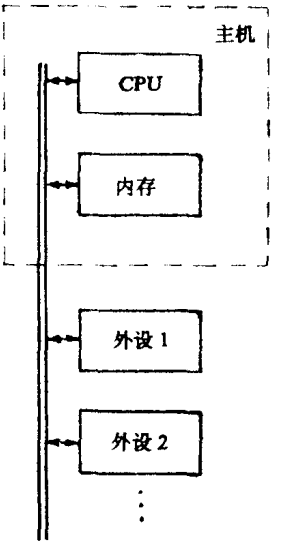

图 1-2

第二节　中央处理器和存贮器的结构

中央处理器(CPU)由运算器和控制器两部分组成。存贮器包括内(主)存贮器和外(辅)存贮器，本节主要介绍内存贮器。

一、运算器的结构

运算器由算术逻辑部件(*Arithmetic Logic Unit*，简称 ALU)和寄存器组等部分组成，如图 1-3 所示。

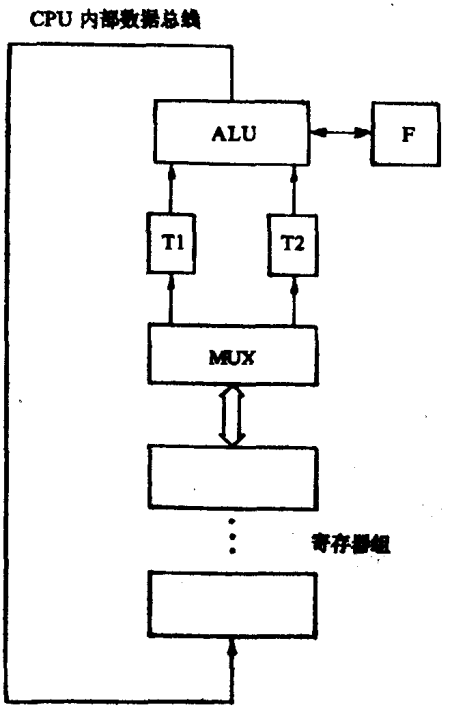

图 1-3

算术逻辑部件是运算器的核心，用来进行算术运算和逻辑运算，也能够进行移位操作。每次运算通常有两个操作数，但数据总线不能同时传送两个数据，必须先后分两次送至 ALU。因为 ALU 本身不能保存信息，所以在 ALU 的两个输入端各配置一个数据暂存器 T1 和 T2，用来存放两个操作数。这两个操作数都送入暂存器后，ALU 立即形成运算结果并向内部数据总线输出，再转送到其它部件。

寄存器组由若干个8位或16位寄存器组成，按其作用可分成通用寄存器和专用寄存器两种。

通用寄存器用来临时存放参加运算的操作数或中间结果，相当于CPU内部的一个小存贮器。由于寄存器在CPU芯片内部，存取数据比内存方便而且速度快。采用多寄存器结构可以大大减少CPU访问内存的次数，以提高运算速度。

专用寄存器是承担专门任务的寄存器。通常有以下几种：

1. 程序计数器PC(*Program Counter*)　它的作用是存放将要取出执行的指令的存贮地址。CPU在执行完当前指令后，根据PC的内容(指令地址)取出下一条指令继续执行。PC又称为指令地址指针，每从PC中取出一次地址，PC就自动计数，指向下一条指令的地址，从而保证程序能自动地顺序执行。PC的内容在机器启动时自动置0，也可以用专门的指令置入指定的地址码。

2. 变址寄存器　它的作用是在变址寻址方式中存放基地址(又称变址基值)。这个基地址可以根据程序的需要用专门的指令置入，基地址加上当前指令中给出的地址偏移量，便形成操作数的实际地址。

3. 堆栈指针SP(*Stack Pointer*)　堆栈是一个按照先进后出存取原则设置的存贮区域。栈底的地址固定，栈顶的地址随数据的存取而浮动。SP存放当前的栈顶地址，也就是说，堆栈指针始终指向栈顶。

4. 标志寄存器F(*Flag*)　标志寄存器的作用是保存并反映CPU执行当前指令后的某些状态，如有无进位、是否溢出、结果是否为0、结果的正负等等。每种状态由标志寄存器的某一位表示，供程序或下一条指令选择操作方向，以及对计算机

的工作状态进行检查之用。

二、控制器的结构

控制器由指令寄存器、指令译码器、时序电路和系统控制信号电路等部件组成，如图 1-4 所示。

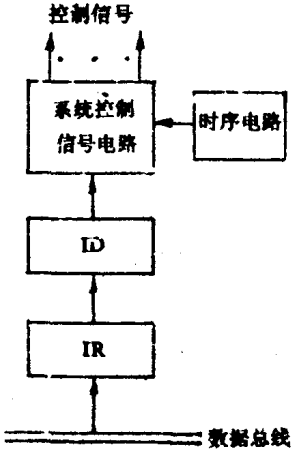

图 1-4

1. 指令寄存器 IR(*Instruction Register*)　指令寄存器用来存放将要执行的指令。存贮器根据 PC 所指示的地址，在控制器发来的读信号控制下，把指令代码从存贮单元中取出，通过数据总线送入指令寄存器，然后送到指令译码器中进行译码。

2. 指令译码器 ID(*Instruction Decoder*)　指令译码器对 IR 中的指令代码进行译码。根据指令代码的内容译成不同的电位组合，这些电位表示当前在 IR 中的指令的操作内容。其中一部分电位信号在 CPU 内部控制寄存器组和 ALU 的操作，另一部分电位信号送至系统控制信号电路。

3. 系统控制信号电路　系统控制信号电路根据 ID 送来

的译码电位信号，在时序信号的作用下，产生对计算机系统各部件的控制信号。使各部件协调地工作，完成指令规定的操作。

4. 时序电路　时序电路通常由晶体振荡器组成的脉冲信号源、启停控制电路、脉冲分配器等组成，是计算机系统的时标系统，用以控制计算机有组织、有规律、有条不紊地工作。

每条指令的执行过程中，各有关部件按一定的时间顺序做各种动作，这些动作必须配合协调，才能完成该指令规定的操作。

计算机工作的时间顺序称为时序，具体地说，是一系列的时间脉冲信号。计算机内部最基本的时间脉冲信号是脉冲信号源产生的时钟脉冲信号，一个时钟脉冲通常称为一个“T”。其频率即为 CPU 的主频，目前一般为几兆至几十兆 Hz，也就是说，计算机系统最基本的时间单位为几百万分之一秒或几千万分之一秒。

计算机执行一条指令的时间称为指令周期，包括取指令周期和指令执行周期。由于各指令的操作内容不同，其指令周期也不同。计算机执行一条指令总是要分若干步来完成，每一步称为一个节拍，每个节拍有一个对应的节拍电位，由脉冲分配器产生，故脉冲分配器又称节拍发生器。

三、存贮器的结构

存贮器由存贮体、存贮器地址寄存器(MAR)、地址译码器和存贮器数据寄存器(MDR)等部件组成，如图 1-5 所示。

1. 存贮体

存贮体是由几万至几百万个存贮单元组成的一个存贮矩阵。任何一个存贮单元，只有在地址译码器送来的行选信号和

列选信号都是高电位时，才能在读或写控制信号的作用下进行读或写操作。

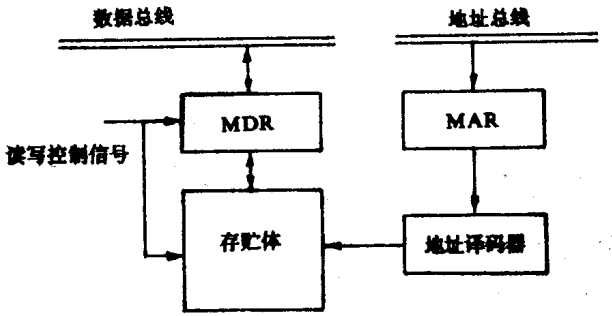

图 1-5

为了说明方便，以 16 个存贮单元的存贮体为例。16 个存贮单元的组成如图 1-6 所示，为 4×4 存贮矩阵。16 个存贮单元的地址由 4 位二进制数来表示。这 4 位地址信号(数据)中，前 2 位为行地址，后 2 位为列地址。行地址和列地址通过地址译码器分别产生 4 个行选信号和 4 个列选信号。设地址为 0110，则 01 行选信号和 10 列选信号为高电位，在 16 个存贮单元中，只有 0110 号存贮单元的行选和列选信号都为高电位，称为被选中的存贮单元。

2. 存贮器地址寄存器 MAR(*Memory Address Register*)

存贮器地址寄存器用来存放将要访问的存贮单元地址。该地址由地址总线送来，在取指令周期，是 PC 的内容；在指令执行周期，是操作数或结果的实际存放地址。

3. 地址译码器

地址译码器把 MAR 中的地址信号分别译码为一组行选

信号和一组列选信号。给定某一个地址，必定有一个行选信号和一个列选信号为高电位。通过这两个高电位信号，选中该地址所对应的存贮单元进行读或写操作。

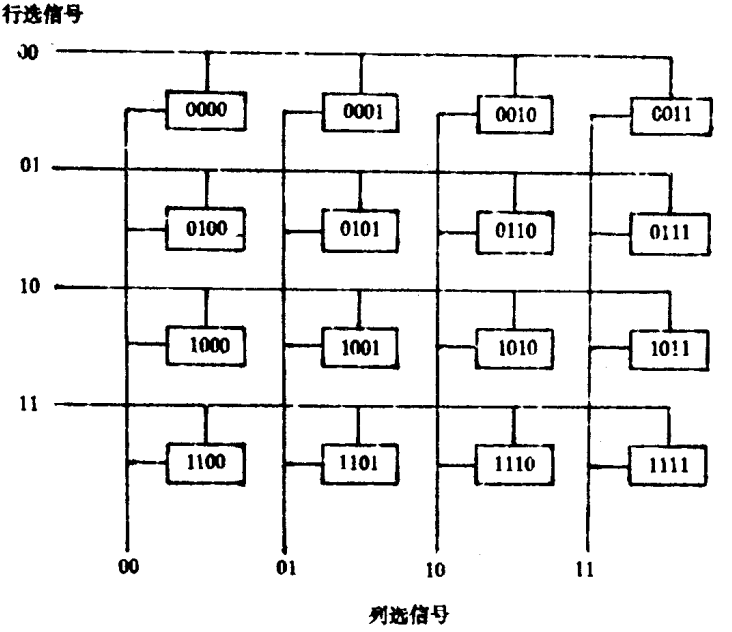

图 1-6

1. 存贮器数据寄存器 MDR(*Memory Data Register*)

存贮器进行读出操作时，MDR 用来存放从被选中的存贮单元中读出的指令或数据，并通过数据总线传送到系统其它有关部件。