

非计算机专业用

电子计算机原理 与程序设计



赠书 155
之

哈尔滨工业大学

非计算机专业用

电子计算机原理与程序设计

电子计算机第一教研室 朱志莹 李 生等编

T/P305/8

林希良

贈書之

哈尔滨工业大学



0686035

内 容 提 要

本书由计算机的硬件和软件两部分组成，硬件部分包括逻辑设计基础和计算机的基本工作原理，软件部分包括程序设计基础（汇编）和 BASIC 语言。

本书讲述了一般电子计算机原理及微型机结构，硬件原理限于框图一级。本书力求做到软硬件结合。每部分后面都附有一定数量的习题。

本书可做理工院校非计算机专业教材或教学参考书，也可供有关科技人员参考。

非计算机专业用

电子计算机原理与程序设计

*

哈尔滨工业大学教材科出版
哈尔滨工业大学教材科发售
哈尔滨工业大学印刷厂印刷

*

1789×1092 1/16 印张13,125字数28,000
1981年8月第一版 1981年10月第一次印刷 印量1—3000
校内使用 校内书号0118 工本费×××

前 言

近年来,电子计算机已推广应用于科学技术及国民经济的各个领域,普及计算机知识已成为一项迫切任务。目前,国内各高等院校以及中等专业学校已纷纷将电子计算机列为各专业的必修课及选修课程。一个亟待解决的问题是提供一本可供普及推广的适应非计算机专业需要的教材。本书就是为解决这个问题的一次尝试。

非计算机专业所需要的知识是电子计算机的应用而不是研究制造。针对这种要求,本书内容具有下面一些特点:(1)对计算机硬件作一般性介绍,只涉及框图一级,但叙述上力求清晰明确,使读者对计算机的操作过程及信息的流动路线有具体而明确的概念。(2)对于缺少数字电路及逻辑设计知识的读者,另辟有逻辑电路及逻辑部件两章,以适应设计接口电路、通道电路及外围电路的需要。(3)详细具体地介绍程序设计方法,除介绍典型的指令系统外,还列举大量程序实例,读者熟悉这一章内容后,即可着手设计应用程序。(4)为适应科学计算及数据处理的需要,着重介绍了 BASIC 语言,这是一种易于掌握而内容又较丰富的语言,适用范围很广,特别适合初学者的需要。

本书结合讲述举有大量实例,每章都附有习题。

本书力求条理清晰,语言简练,易学易懂,在内容上尚有增删余地。一个突出的特点是:不拘泥于哪一种具体的机器,指令系统具有典型性。在掌握本书内容的基础上,可进一步对某一部分作深入的学习和提高。可以说,对于初学计算机的读者来讲,这是一本较为适宜的教材。

当然,由于初次尝试再加业务水平所限,本书在内容上难免还有不少缺点和不当之处,恳请读者和教学工作者不吝指教,以便再版时予以修订完善。

本书由朱志莹、李生统编,朱志莹侧重于硬件部分,李生侧重于软件部分。参加本书编写工作的有:王继德(第二章、第三章),高明(第四章),张惠蕴(第六章),徐淑华(第十章),朱志莹(第五章、第七章、第八章),李生(第一章、第九章、第十一章)。本书经吴纯园、李仲荣二位副教授审阅。

最后,特向参与和支持本书编写和出版的有关同志致以衷心的感谢。

编 者

1981年6月2日

目 录

第一章 绪论

§ 1.1 电子计算机的简单介绍	1
一、电子计算机的特点	1
二、电子计算机的分类	1
三、计算机使用的工作流程	2
四、计算机系统的构成	2
§ 1.2 硬件的框图及工作过程概述	3
一、硬件框图	3
二、硬件的工作过程	5
§ 1.3 各种软件的作用及相互之间的关系	6
一、各种软件的作用	6
二、系统软件之间的相互关系	7
§ 1.4 计算机的应用	7
一、科学计算	7
二、数据处理	8
三、过程控制	8
§ 1.5 计算机的历史及发展动向	8
一、计算机的历史	8
二、计算机的发展动向	9
§ 1.6 微型计算机的出现	10
一、微型计算机的特点	10
二、微处理器与微型计算机	10
三、微型计算机的发展	11

第二章 数制、码制和编码

§ 2.1 二进制数	12
§ 2.2 二进制数与十进制数的转换	14
§ 2.3 二进制定点数和码制	16
一、原码	16
二、反码	16
三、补码	17
§ 2.4 二进制浮点数	18
§ 2.5 编码	19
§ 2.6 二十进制编码 (BCD)	20
§ 2.7 其他编码	21

第三章 逻辑代数与基本逻辑电路

§ 3.1 逻辑代数	24
§ 3.2 典型逻辑门	26
§ 3.3 卡诺图化简方法	27
§ 3.4 逻辑表达式的实现	31
§ 3.5 半加器、全加器和加法器	32
一、半加器	32
二、全加器	32
三、加法器	33
四、溢出检测	33

第四章 触发器与基本逻辑部件

§ 4.1 触发器	35
一、RS 触发器	35
二、D 触发器	36
三、JK 触发器	37
四、触发器小结	38
§ 4.2 基本逻辑部件	38
一、寄存器	38
二、计数器	40
三、译码器	42
四、节拍脉冲发生器	43
五、基本逻辑部件小结	43

第五章 运算方法和运算器

§ 5.1 加法电路	46
一、并行加法电路	46
二、串行加法电路	48
§ 5.2 定点加减法运算	49
§ 5.3 定点乘除法运算	50
一、原码乘法运算	50
二、原码除法运算	52
§ 5.4 浮点加减运算	55
一、对阶	55
二、求和	55
三、规格化	56
四、舍入	56
§ 5.5 逻辑运算	57

第六章 存储器

§ 6.1 存储器的功用和种类	59
-----------------	----

§ 6.2 磁心存贮器	60
一、磁心存贮二进制代码的原理	60
二、磁心体的组成	61
三、磁心的存取方法	61
§ 6.3 磁心存贮器的组成和操作流程	65
一、地址部件	65
二、读写部件	65
三、时序控制部件	66
四、读写操作流程	66
§ 6.4 半导体存贮器	67
一、静态双极型存贮器	67
二、静态 MOS 存贮器	69
§ 6.5 只读存贮器	69
第七章 控制器	
§ 7.1 控制器的功能及组成	71
§ 7.2 指令系统	72
一、指令格式	72
二、指令的类型	73
三、寻址方式	74
§ 7.3 指令部件	77
§ 7.4 定时部件	77
一、机器周期的设置	77
二、节拍电位和工作脉冲	78
三、启停控制	79
§ 7.5 操作控制部件	80
一、什么是操作控制部件?	80
二、指令操作时间表	80
三、微操作控制信号的综合	82
§ 7.6 微程序操作控制	83
第八章 外部设备及中断	
§ 8.1 外部设备的功用和类型	85
§ 8.2 程序中断	85
§ 8.3 输入输出接口	87
§ 8.4 直接存贮器存取	90
§ 8.5 纸带输入机	91
§ 8.6 行式打印机	92
第九章 微型计算机结构	
§ 9.1 微型计算机结构及其工作过程	94

一、微型计算机结构	94
二、工作过程	95
§ 9.2 CPU与寄存器	95
一、Z80 的 CPU 结构	95
二、Z80 的电路引线	98
§ 9.3 内存与总线	99
一、半导体存储器	99
二、总线	101
§ 9.4 输入输出及其接口	102
一、输入输出控制方式	102
二、I/O 接口	104

第十章 程序设计

§10.1 程序设计基本概念	106
一、程序设计语言	106
二、程序流程图	106
三、程序分类	107
§10.2 指令系统	108
一、寻址方式	108
二、指令系统	111
三、伪指令	115
§10.3 汇编语言程序格式	117
§10.4 算术运算程序	118
一、多倍精度加法	118
二、十进制加法	119
三、八位二进制乘法	119
四、八位二进制除法	120
§10.5 数据处理程序	121
一、ASCII 字符加偶校验位	121
二、字符串匹配	121
三、ASCII 代码化为十进制(BCD)代码	122
四、BCD代码化为二进制数	123
五、在表内增加一项	123
六、有序表内容检索	123
七、8 位分类	125
八、数据块搬家	125
§10.6 子程序调用	126
一、十六进制变换为 ASCII 代码	126
二、字符串长度检测	127

三、无序表的检索.....	128
§10.7 输入输出程序.....	130
一、并行输入输出接口 (PIO)	130
二、工作方式选择.....	131
三、输入/输出程序举例	132
§10.8 中断处理程序.....	134
一、键盘中断.....	134
二、打印中断.....	135
第十一章 BASIC 语言	
§11.1 概述.....	137
一、BASIC语言的引进.....	137
二、BASIC程序的构成.....	137
三、BASIC程序的运行.....	138
§11.2 基本表达方式.....	139
一、基本符号.....	139
二、数的表示.....	140
三、表达式.....	140
§11.3 赋值语句和结束语句.....	141
一、赋值语句.....	142
二、结束语句.....	142
§11.4 输入语句和输出语句.....	144
一、输入语句.....	144
二、输出语句.....	145
§11.5 数据语句和读语句.....	149
§11.6 转向语句和条件转向语句.....	151
一、转向语句.....	151
二、条件转向语句.....	152
§11.7 循环语句.....	158
§11.8 开关语句和转子 (程序) 语句.....	161
一、开关语句.....	161
二、转子 (程序) 语句.....	162
§11.9 函数语句.....	165
一、正弦函数 $\sin x$	165
二、余弦函数 $\cos x$	165
三、正切函数 $\operatorname{tg} x$	166
四、反正切函数 $\operatorname{arc} \operatorname{tg} x$	166
五、指数函数 e^x	166
六、自然对数函数 $\ln x$	166

七、平方根函数 $\sqrt{x}$	166
八、绝对值函数 $ x $	166
九、符号函数	166
十、取整函数	167
十一、取小函数	167
十二、最大值函数	167
十三、最小值函数	167
十四、“与”函数	167
十五、“或”函数	167
十六、“异或”函数	167
十七、随机整数函数	168
十八、随机小数函数	168
十九、自定义函数	170
§11.10 数组及矩阵运算	172
一、数组	172
二、矩阵运算	178
§11.11 输出格式	182
一、TAB 输出	182
二、SPC 输出	183
三、USING 输出	183
§11.12 扩充 BASIC——文件的建立与使用	186
一、字符串的表达方式	187
二、字符串函数	188
三、文件命令和文件语句	189
四、扩充 BASIC 程序编制举例	190
§11.13 BASIC 程序的建立、修改及运行	194
一、常用命令	194
二、新程序的建立	196
三、修改程序	196
四、程序的运行	198

第一章 绪 论

§ 1.1 电子计算机的简单介绍

一、电子计算机的特点

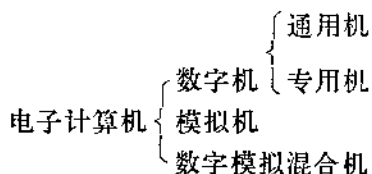
由于计数和计算的需要，人类在很早以前就发明了算盘、计算尺、手摇计算机等计算工具。但随着生产和科学技术的发展，大量的计算工作已经不是上述计算工具所能胜任的了。伴随着电子技术的发展，二十世纪四十年代出现了一种新型的计算工具——电子计算机。

电子计算机之所以能够完成如此复杂的计算任务，原因在于它是一种能够自动地、高速度地、精确地进行大量计算工作的电子设备。它可以按照人们事先编好的程序自动地进行计算，而不需要人们直接参与计算过程；它的计算速度可以达到每秒几百万次，几千万次，甚至几亿次；它的精确度可以达到十几位的十进制数字。这些都是过去任何计算工具所望尘莫及的。

电子计算机的强大计算功能是由它的结构特点所决定的。它的基本元件采用了在速度上远远超过机械元件的电子元件，在结构上采用了“程序存贮”和“程序控制”的方法。程序是指人们事先编好的解题顺序，它是由一系列的指令所组成，而指令则是人们事先规定的指示机器完成某种操作的命令。所谓“程序存贮”，是指在解题前将编好的程序按照一定的地址顺序存入主存贮器。解题时再由主存贮器把事先存好的程序中的指令按一定顺序一条一条地取出来，由它来控制机器的解题过程，这就是所谓的“程序控制”。实际上，电子计算机的解题过程就是程序的执行过程。

电子计算机不仅能够进行大量的数字运算，而且还能够进行逻辑分析和逻辑判断。所以它不仅可以部分地减轻和代替人的繁琐的体力劳动，还可以部分地减轻和代替人的脑力劳动。电子计算机的出现和发展，开辟了生产和科学技术发展的一个新纪元。

二、电子计算机的分类



从工作原理上来看，电子计算机可以分成电子数字计算机、电子模拟计算机和电子数字模拟混合计算机等三大类。数字机的输入和输出都是不连续的数字量，而模拟机的输入和输出则是连续的模拟量，如温度、流量、电压等等。将这两类机器结合起来，互为补充，就形成了这第三类电子计算机——数字模拟混合计算机。从性能上来看，数字机又可以有通用机和专用机之分。通用机是用来解决一般的数学问题和逻辑问题的，专用机则是用来解决某一专门问题的。例如用于实时控制的计算机就是一种专用机。

我们这里要讲到的，主要是电子数字计算机。对于数字机来说，从机器规模上来分，还可以分成大型机、中型机、小型机和微型机等四种。区分大小的标准，有的是按技术指标来分，有的是按售价来分，如下表所列。

表 1—1 按技术指标和售价分类表

划分标准 数据 机器类型	按技术指标分	按售价分
大型机	1000万次/秒以上	100万美元以上
中型机	100万次~1000万次/秒	15万~100万美元
小型机	50万次~100万次/秒	1万~15万美元
微型机	50万次/秒以下	1万美元以下

这种分法并不是一成不变的，例如目前许多微型机的技术指标就远远超过了过去的小型机的技术指标。

三、计算机使用的工作流程

客观事物变化过程→数学模型→近似计算公式→编制计算程序→计算机的自动计算（计算程序及原始数据的输入、计算、结果的输出）→分析结果。

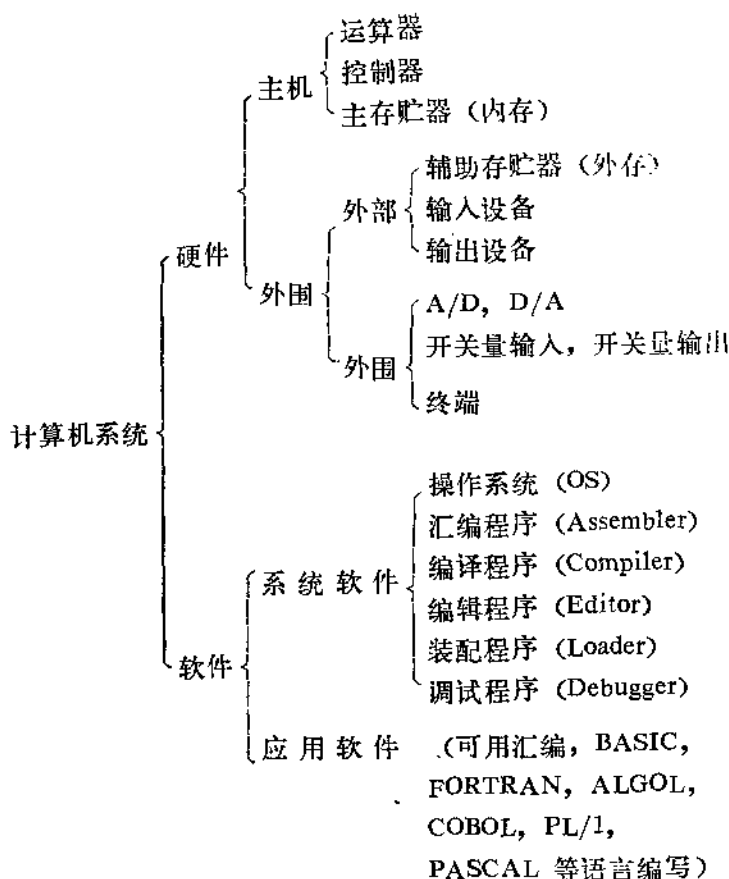
数字机本身只能进行简单的加、减、乘、除等算术运算（有的机器甚至连乘除运算也不能直接进行）和“与”、“或”、“非”等逻辑运算。要想用数字机处理一个问题，首先需将这个问题转换成相应的数学问题，也就是建立数学模型，然后再将这个数学模型用适当的方法转换成一系列简单的算术运算和逻辑运算，这就是所说的计算方法。确定了算法之后，再用适当的语言编制计算程序，将这种编制好的程序连同原始数据一起送入计算机，进行自动计算，最后分析计算结果。

四、计算机系统的构成

现代的计算机系统包括计算机硬件和计算机软件两大部分。所谓硬件是指构成计算机的具体的电子——机械设备。软件是指那些不能直接接触到的存放在内存贮器或磁带、磁盘等外存贮器上的存贮程序。这些存贮程序有的是预先装入系统内以充分发挥系统功能的，叫做系统软件或系统程序，这些程序包括有操作系统、汇编程序、编译程序、编辑程序、装配程序和调试程序等等。还有一些是用汇编语言或者是用 BASIC、FORTRAN、ALGOL、COBOL、PL/1、PASCAL 等高级语言写成的用户程序，即应用软件。

硬件包括主机和外围两大部分，这部分放到后面去具体介绍。

计算机的硬件和软件之间是互相配合的。不配软件的硬件通常称之为“裸机”。可以形象化地将电子计算机比喻成一个人，硬件好比一个人的躯体，软件则相当于人的灵魂（包括思维和语言）。可见做为电子计算机的硬件是离不开软件的，没有软件的硬件就相当于人没有灵魂，既不能理解向他提出的问题，也不能完成交付给他的任务。同样，只有灵魂而没有躯体的人也是不存在的，因为软件必须依附于硬件而存在，必须借



助于硬件来实现其功能。可见，软件和硬件是不可分离的。

§ 1.2 硬件的框图及工作过程概述

一、硬件框图

电子数字计算机是直接对数字进行运算或加工的一种工具，为了完成运算或加工任务，计算机必须具备下述几个部分。

1. 主机

① 运算器 (ALU——Arithmetic Logic Unit)

进行算术运算和逻辑运算的装置，由完成加法运算的加法器 (Adder)，存放操作数和运算结果的寄存器 (Register) 和累加器 (Accumulator) 等部分组成。

② 控制器 (Control Unit)

全机的指挥中心，依据不同的指令产生不同的操作命令，指挥整机有条不紊的进行工作。它由给出指令地址的指令计数器 (PC——Program Counter)，暂时寄存指令的指令寄存器 (IR——Instruction Register) 和将指令翻译成操作命令的指令译码器 (ID——Instruction Decoder) 及其他有关部分组成。

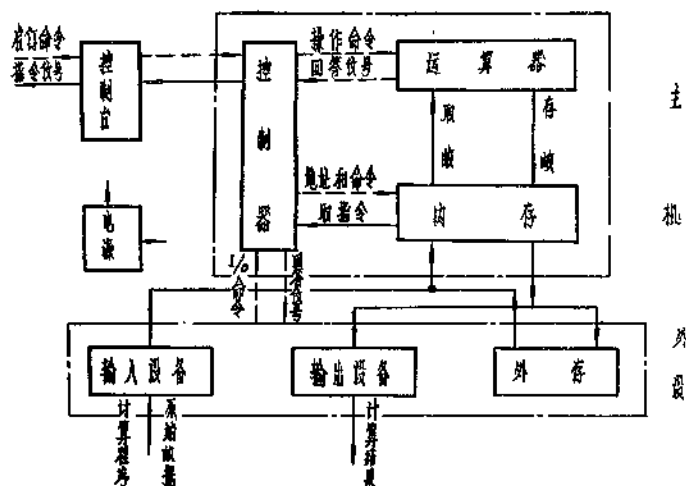


图 1-1 硬件的简单框图

③ 主存储器 (Main Memory)

又称内存贮器，包含有许多存储单元，用以存放大量的二进制代码，这些代码包括指令码与数据码，它可以直接给控制器提供指令或者给运算器提供操作数。基本存储单元有的采用磁芯元件（磁芯存储器），有的采用半导体元件（半导体存储器）。

运算器和控制器有时又统称为中央处理单元（CPU——Central Processing Unit）。

2. 外设

① 输入设备 (Input device)

用以将计算程序和原始数据转换成电信号，并在控制器的指挥下按一定地址顺序送入内存。

目前国内通常采用穿孔纸带做为输入形式，也有的采用卡片做为输入形式。输入设备通常有光电输入机、电容式纸带输入机、卡片输入机等等。国外已大量采用通过输入键盘将输入代码记录在软磁盘上，这种输入方式灵活方便。

② 输出设备 (Output device)

用以将计算结果在控制器的指挥下，依照人们能够识别的形式由内存送往机外。

输出的形式有穿孔、打字、绘图和屏幕显示等等。常用的输出设备有穿孔输出机、行式打印机、x-y 绘图仪及屏幕显示设备等等。

控制台打字机和显示终端既可以输入信息，也可以输出信息。

③ 辅助存储器 (External Storage)

又称外存贮器，它是内存的后备，工作速度比内存低，但存储容量比内存大，它不直接和运算器打交道，通常用来存放那些在解题过程中暂时还用不到的程序和数。

常用的外存有磁鼓、磁带、磁盘。

做为工业控制机来说，因为输入量和输出量都是模拟量，因而除了配有必备的外部设备之外，还应配有一些相应的外围设备，这些设备通常有模/数 (A/D——Analog/Digital) 转换器和数/模 (D/A) 转换器，开关量输入装置和开关量输出装置等等。

此外,电子计算机还应配备有控制台(Console)和电源(Power)。

一台电子计算机结构十分复杂,但实际上它所包含的电路通常只有下述几种:

- ① 逻辑门——与门、或门、非门,实际上经常采用它们的复合形式;
- ② 寄存电路——触发器;
- ③ 辅助电路——脉冲发生器、整形器、放大器及延迟线等等。

现代计算机已经采用了中大规模集成电路,往往是将上百或上千个上述简单电路集成在一块芯片上,从而使硬件的外部联结越来越简单了。

二、硬件的工作过程

前面已经讲过,计算机是在控制器的指挥下自动地进行工作的,而控制器指挥机器工作的依据是计算程序——有序指令串。

指令是指示机器完成某种操作的一种命令。指令一般由操作码(OP—code)和操作数(Operand)两部分组成。操作码部分决定指令的操作性质,包括“+”、“-”、“×”、“÷”等算术操作和“与”、“或”、“非”等逻辑操作。操作数部分给定了参加操作运算的数码,操作数可能是两个(如算术运算和逻辑运算),也可能是一个(如移位操作),但指令中通常只给出一个。对于大多数机器来说,如果执行一条指令需要两个操作数,则另一个已经存放在运算器的累加器当中。一般情况下,指令的操作数部分并不是直接给出参加操作运算的数码,而是给出该数码所在的内存单元或寄存器的编号,所以这部分通常称之为操作地址(Address)或称之为地址码(见图1—2)。

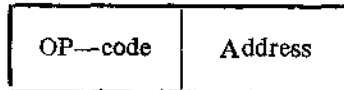


图 1—2 指令格式

下面以计算 $f = ax^2 + bx + c$ 为例来说明硬件的简单工作过程。这里假定 a 、 b 、 c 和 x 皆为已知数。

第一步:输入,即将计算程序和原始数据一起,通过输入设备依次送入内存当中去。这里原始数据为 a 、 b 、 c 和 x ,计算程序则需在输入之前事先由程序员或用户编好。对于上述题目,我们可以编制成如下计算程序。

先将题目做如下变化:

$$\begin{aligned} f &= ax^2 + bx + c \\ &= (ax + b)x + c \end{aligned}$$

最后编出的程序如表 1—2 所示。

这里 A ——累加器

(A)—— A 中存放的数,即 A 的内容,同样,(8)为内存8号单元的内容等等。

第二步:运算。

- ① 启动,操作员通过控制台发出启动命令给控制器,将机器启动;
- ② 取指令,控制器先按控制台所给出的起始地址(0号单元)从内存“0”单元取出第一条指令 $\rightarrow 8$;
- ③ 取操作数,控制器再按这条指令所给出的操作地址(8号单元),第二次访问

表 1—2

 $f = ax^2 + bx + c$ 的计算程序表

指令或数码地址	操作码	地址码	操作内容	说明
0	$\rightarrow$	8	$(8) \rightarrow A$	$a \rightarrow A$
1	$\times$	11	$(A) \cdot (11) \rightarrow A$	$ax \rightarrow A$
2	$+$	9	$(A) + (9) \rightarrow A$	$ax + b \rightarrow A$
3	$\times$	11	$(A) \cdot (11) \rightarrow A$	$(ax + b)x \rightarrow A$
4	$+$	10	$(A) + (10) \rightarrow A$	$(ax + b)x + c \rightarrow A$
5	$\leftarrow$	12	$(A) \rightarrow 12$	$(ax + b)x + c \rightarrow 12$
6	打印			
7	停机			
8	a			
9	b			
10	c			
11	x			
12				工 作 单 元

内存，从内存“8”单元取出操作数 a 送运算器；

④ 操作运算，将控制器中的指令操作码所形成的操作命令和第二次访问内存所取得的操作数同时送运算器，由运算器完成操作运算 $a \rightarrow A$ 。

在进行操作运算的同时，控制器里保存指令地址的指令计数器 PC 自动“+1”，然后按新的指令地址（1号单元），再重复执行第二步中的②，③，④，即取指令 $\rightarrow$ 取操作数 $\rightarrow$ 操作运算，完成 $ax \rightarrow A$ 的操作；再继续执行 $ax + b \rightarrow A$ ， $(ax + b)x \rightarrow A$ ， $(ax + b)x + c \rightarrow A \dots\dots$ 等操作，最后将运算结果送内存工作单元。

第三步：输出。

取出存放在内存6号单元中的第七条指令，命令输出设备将运算结果打印出来。

第四步：停机。

解题结束。

§ 1.3 各种软件的作用及相互之间的关系

一、各种软件的作用

1. 操作系统 (OS——Operating System)；

用于管理计算机的软件和硬件资源。

2. 编辑程序 (Editor)；

允许程序员通过显示终端交互式地建立和修改程序或数据文件。

3. 编译程序 (Compiler) 和汇编程序 (Assembler)；

编译程序用于将高级语言编写的源程序编译成机器码程序。

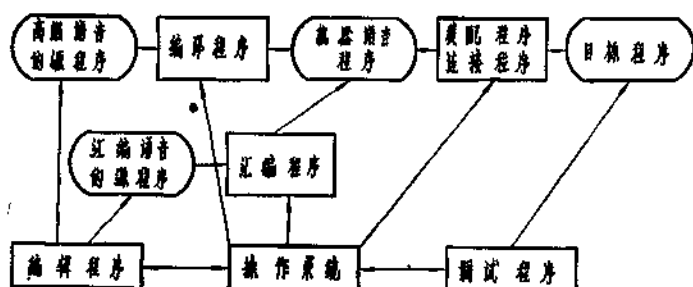


图 1—3 各种软件之间的相互关系

汇编程序用于将汇编语言编写的源程序翻译成机器码程序。

4. 装配程序 (Loader) 和连接程序 (Linker) :

将各种程序分配给以一定的存储空间, 并将目标程序的各部分相互连接起来, 最后将机器语言指令和数据都赋予具体的存储地址, 形成机器可以直接执行的目标程序。

5. 调试程序 (Debugger) :

用于解题程序的故障诊断和排除。

二、系统软件之间的相互关系

作为电子计算机的硬件只能执行符合机器的指令系统规定并且用二进制代码所表示的机器语言程序。机器语言就是用二进制代码所表示的指令, 也就是说, 计算机只能识别用机器语言所编写的程序。但是为了编制程序的方便, 目前人们普遍采用符号语言 (汇编语言)。还有更接近人们日常生活所用的语言和数学公式编制的程序 (高级语言)。然而这种程序计算机是不能直接识别的, 计算机必须将这种程序经过编辑之后再经过一个翻译过程 (高级语言用编译程序做翻译, 汇编语言用汇编程序做翻译) 将其翻译成机器语言程序。这种机器语言程序的主程序和机器语言的子程序通过装配和连接程序, 装配到主存储器里面, 成为计算机可以执行的程序——目标程序。程序的编制若有错误, 还可以通过调试程序检查排除, 然后机器便可以正确地执行了。

§ 1.4 计算机的应用

由于电子计算机能够自动地、高速度地、精确地进行大量算术运算和逻辑运算, 从而使得许多过去无法解决的问题得到了解决, 它不仅能够解决数学问题, 还能够解决逻辑问题, 因而在工业、农业、科学技术和国防等各个领域得到了广泛应用。它的主要应用领域有以下几个方面:

一、科学计算 (数值计算)

现代科学技术和工程设计上经常遇到一些极其复杂的数学问题, 这些问题用落后的计算工具是无法解决的。如求解描述导弹空间运动的微分方程组, 一个计算员用手摇计算机一天24小时不停地工作需要2年。有些计算还有很强的时间性, 如天气预报不超前一定时间就不成其为预报了。用人工方法去解气象方程需要2个星期, 但是这些工作若让一台几十万次/秒的电子计算机去干, 只需要几分钟就足够了。