高等学校试用教材

化学专业计算机基础

杨致葳 王祥云 编

原子能出版社

内 容 简 介

本书结合一般化学及应用化学中常遇到的数学问题讲解计算机语言和算法并给出有关程序和例题。

其内容包括计算机基本硬件知识、FORTRAN 语言、化学中常用数值分析方法基础、BASIC 语言及其应用等。附录中收入了Z80 微型计算机指令表、FORTRAN Ⅳ及FORTRAN77 语句表和基本 BASIC语句表等。

本书为高等学校应用化学和其他化学专业本科学生学习计算机与算法语言的教科书，也可供上述领域中从事科研工作的人员参考。

高等学校试用教材

化学专业计算机基础

杨致葳　王祥云　编

陈敏伯　审

原子能出版社出版

（北京2108信箱）

国防科工委印刷厂印刷

新华书店总店科技发行所发行·新华书店经售

☆

开本787×1092 1/16 ·印张15.25·字数368千字

1988年2月北京第一版·1988年2月北京第一次印刷

印数 1—2500

统一书号：15175·886　定价：2.55元

ISBN 7-5022-0031-2/O·5

前　言

本书是为高等学校应用化学和其他化学专业本科学生学习计算机与算法语言编写的教科书。

本书是在北京大学应用化学专业计算机课程讲义基础上编写的。该书的特点是结合一般化学及应用化学中常遇到的数学问题讲解语言和算法并给出有关程序和例题。

全书分四部分：

第一部分为计算机基本硬件知识。其中以Z 80微型计算机为典型进行讲解。该部分的叙述力求通俗、精炼，易于化学专业学生学习。

第二部分为FORTRAN语言。目前我国广泛应用的是FORTRAN Ⅳ文本，它不仅能在大、中、小型计算机上使用，而且也能在八位以上微型计算机系统上使用。近年来在FORTRAN Ⅳ文本的基础上又发展了 FORTRAN 77 文本，后者在功能上有较大的扩充，但 FORTRAN 77必需配置在16位以上的计算机上。用 FORTRAN Ⅳ编写的程序可在配有 FORTRAN 77 的计算机上运行，反之则不可以。本章在讲解FORTRAN语言时基本上采用了FORTRAN Ⅳ文本，其中例题都在Z 80微型计算机系统上调试过，同时对 FORTRAN 77扩充的主要语句及功能也作了简单叙述。在附录中列有FORTRAN Ⅳ及FORTRAN 77语句表。

第三部分为化学中常用的数值方法基础，包括用行式打印机绘图、线性方程组的求解、非线性方程和非线性方程组的求解、插值法、数值积分和常微分方程的数值解法等内容共16个通用子程序。对于每一个子程序，除列出其功能和形式参数外，还比较详细地叙述所用的计算方法并举例说明该子程序在化学中的应用。本章所有程序都是用FORTRAN 77编写的，并且在HONEYWELL DPS 8/52机上调试过。

第四部分为 BASIC 语言及应用。由于 BASIC 语言有很大的实用价值，许多先进大型化学仪器的控制程序都是用 BASIC 语言编制的，仪器中的计算机单独使用时多数都能用 BASIC语言算题，因此本书安排了这部分内容。鉴于有前面 FORTRAN 语言的学习基础，这部分内容的编写方法是，列出 BASIC 语言的所有语句，进行简单解释，重点结合应用讲解程序。

鉴于目前各校化学专业的计算机课学时不统一，我们将有些内容标上* 号，如学时不够，这部分内容可以不讲，留给学生自学。

本书中的程序是从教学和应用角度编写的，不一定是该问题的最佳解法或最完善的程序。

本书第一、二、四章由杨致葳编写，第三章由王祥云编写。由于作者学识水平有限，书中错误在所难免，敬请指正。

本书由北京大学陈敏伯同志审稿，夏松江同志审阅了第一章。

本书曾经审稿会审定。参加审稿会的有北京大学江林根、复旦大学赵魁东、清华大学吴华武、华东地质学院丁维德、四川大学卢集政等同志，他们对本书提出了许多宝贵意见。另外，北京大学技术物理系和计算中心的许多同志对本书的编写也曾给于大力支持。作者在此一并致谢。

编　者

1986年 6 月于北京大学

目　　录

第一章　微型计算机基本原理及汇编语言简介

一、微型计算机的发展及在化学中的应用

电子计算机是当代最激动人心的科学成就之一，它的发展不仅对国民经济有着重大的推动作用，而且对人民生活也产生着极深刻的影响。

当前计算机正朝着巨型化与微型化方面发展。微型计算机，人们又习惯地称其为“电脑”，自1971年问世以来，得到迅猛的发展。微型计算机的发展是以大规模集成电路的发展为基础的。六十年代后期，大规模集成电路的研究取得很大进展，使得有可能把计算机中的运算控制器即微处理器做在一块小硅片上。1971年美国Intel公司第一次成功地在一块硅片上制成了字长为4位的微处理器，后来很快又制成了字长为8位的微处理器。由于微处理器价格便宜，配上接口及存贮器后具有完整的计算机功能，可大量用于仪器设备的控制，从而使仪器设备的自动化产生了巨大的变革。七十年代末到八十年代初又生产了字长为16位及32位的微处理器。微处理器配上各种外围设备及软件，就形成微型计算机系统，能进行小规模科学计算。这样，微型计算机便可广泛应用于生产、科研、教学以及人民生活的各个领域了。

在化学领域中，人们早已认识到计算机对化学科学发展的重要作用。1971年在美国召开了“计算机在化学研究及教育中应用”的首次国际会议，后来在1973年和1976年又相继召开了第二次和第三次国际会议，最近一次是1987年在中国召开的第八次国际会议。在这些会议上广泛交流了计算机在化学各领域中的应用，并证明了计算机应用于化学从而大大推动了化学学科的发展。

计算机在化学中的应用大致可归纳为三方面。第一方面是进行各种运算与数据处理，例如对实验中的数据进行拟合，找出经验公式等。利用计算机进行运算和处理数据可大大提高处理的速度和准确性。第二方面是将微型计算机应用于各种化学仪器，使其成为智能仪器，使仪器的分析过程高度自动化。目前较先进的大型化学仪器如液相色谱、气相色谱、红外光谱、核磁共振仪等都已用微型计算机进行控制。第三方面是利用计算机建立化学数据库、进行化学教学和情报检索等。基于上述原因，作为一个化学工作者，掌握最基本的计算机和算法语言方面的知识是很重要的。

二、微型计算机系统、微型计算机和微处理器

微型计算机系统和大型计算机系统的基本结构相同。这里以Z80微型计算机为典型实例来叙述微型计算机的原理。图1.1表明了构成微型计算机系统的基本部件。

微型计算机系统由硬件和软件组成。硬件由微型计算机、输入、输出外围设备及电源组成。软件由各种高级语言编译程序、汇编语言编译程序及操作系统程序等组成。

微型计算机具有完整的计算机功能，它包括微处理器、系统总线、内存贮器、输入电路和输出电路等部件。

微处理器也叫中央处理器(Central Processing Unit)，简称CPU。它是计算机的运算控制部分，它包括运算器、寄存器、控制器。由于微处理器是计算机的核心，我们将在第四节中对其进行专门讨论。

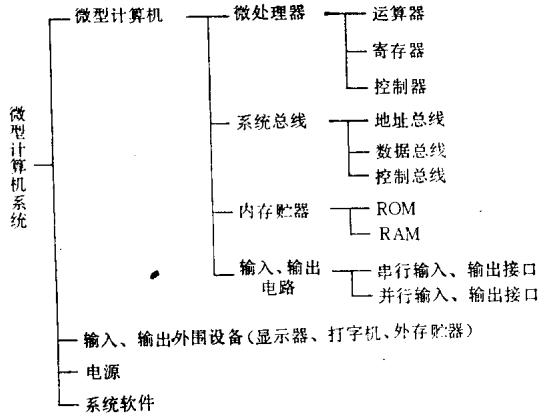

图1.1 微型计算机系统

系统总线是计算机中传输信息的通讯线路，外部信息要通过总线输到计算机中进行运算，计算结果也要通过总线送到外部设备。总线分地址总线、数据总线与控制总线。地址总线是传送地址信息的线路。当算题时，首先要将被计算的题目作为输入信息存到存贮器中，存贮器的每一个单元都有一个地址，用户的信息要存放到哪一个存贮单元中去，在地址总线上就要存放该存贮单元地址的信息。Z80微型计算机地址总线为16位。数据总线是微型计算机输入、输出数据的线路，指令及被解题目中的数据都可做数据信息。Z80微型计算机数据总线为8位。控制总线是传送微型计算机各种控制信息的线路。

内存贮器是按一定的地址存入或取出程序和各种数据的装置。内存贮器由磁芯存贮器或半导体存贮器构成，一般大型计算机采用磁芯式存贮器，微型计算机采用半导体式存贮器。内存贮器按其功能又分随机存取存贮器（RAM）和只读存贮器（ROM）。RAM是用户用来随意存取程序和数据的地方，当电源关掉后，RAM中的信息会消失。ROM在出厂时由设计者存入一定的程序，用户只能将里面的程序调出来使用，不能随意改变其中的程序，当电源关掉后，ROM中的程序是不消失的。

输入、输出电路是将外部设备的信息输入到计算机或将计算机内的信息输给外部设备的接口电路。例如用户要把指令信息通过键盘输入到CPU，通常要通过串联接口电路；把计算机运算的结果送到打字机，通常通过并联接口电路。

输入、输出外围设备包括显示器、打字机、外存贮器等。显示器通常包括键盘输入装置，用户通过它向计算机输入程序、数据等信息。打字机为输出设备，将计算的中间结果和最后结果以数字、字母、符号和图形等形式表示出来。外存贮器有磁鼓、磁带、硬磁盘和软磁盘等。一般微型计算机系统都带有软磁盘，这里我们着重介绍软磁盘。软磁盘按其大小分8英寸盘（直径为8英寸）及5英寸盘（直径为5英寸）。软磁盘的存贮容量分单面单密度、单面双密度、双面单密度及双面双密度等。单面单密度8英寸盘容量为256K字节，双面单密度8英寸盘容量为512K字节，单面单密度5英寸盘容量为80K字节，双面单密度5英寸盘容量为180K字节，一般双密度盘的容量比单密度盘的容量扩大一倍。外部存贮器用来存放大量程序和数据，需要时可随时调到RAM中使用，也可把计算结果及用户程序存入外存贮器。存在RAM中的信息关机后就不存在了，而存在外部存贮器的信息关机后是不消失的。外部存贮器不能直接与CPU交换信息，必须通过内部存贮器。

电源是微型机系统中必有的部件，它的好坏对微型机的质量影响很大，如电源设计不好会使机器发热、经常出故障、不能长时间使用。

三、数据的表示和逻辑运算

（一）数制及其转换

人们在日常生活中最熟悉的数据表示方法是十进制，如1984可很快理解到是一千九百八十四的意思。十进制的表示方法是任意一位数都可取0、1、2、3、4、5、6、7、8、9中的任一个数。从低位到高位是“逢十进一”，因此在十进制中10是十的意思，1984即是

$$1\times10^3+9\times10^2+8\times10^1+4\times10^0$$

的缩写

在计算机中采用的是二进制的数据表示方法，因为计算机中存贮数据的存贮器只能有“1”、“0”两种状态。二进制每位数只能用1和0两种符号，从低位到高位是“逢二进一”，因此在二进制中10就不是十而是二了。

下面表格表示了二进制数与十进制数的对应关系。

表1.1　二进制数与十进制数的对照

整数		小数	
二进制	十进制	二进制	十进制
1	$2^0=1$	0	0
10	$2^1=2$	0.1	$2^{-1}=\frac{1}{2}=0.5$
100	$2^2=4$	0.01	$2^{-2}=\frac{1}{4}=0.25$
1000	$2^3=8$	0.001	$2^{-3}=\frac{1}{8}=0.125$
10000	$2^4=16$	0.0001	$2^{-4}=\frac{1}{16}=0.0625$
100000	$2^5=32$	0.00001	$2^{-5}=\frac{1}{32}=0.03125$
1000000	$2^6=64$	0.000001	$2^{-6}=\frac{1}{64}=0.015625$
10000000	$2^7=128$	0.0000001	$2^{-7}=\frac{1}{128}=0.0078125$
⋮	⋮	⋮	⋮
$\underbrace{10\cdots\cdots0}_{n}$	2^n，n是1后面0的个数	$\underbrace{0.0\cdots\cdots01}_{n}$	2^{-n}，n是1前面0的个数包括小数点前一位整数0

按以上表格可实现二进制数和十进制数间的转换。例如

$$(1111)_2=(1000+100+10+1)_2=(1\times2^3+1\times2^2+1\times2^1+1\times2^0)_{10}$$
$$=(8+4+2+1)_{10}=(15)_{10}$$

以上方法可简单归纳为1、2、4、8记忆方法。二进制数中从后往前数，其第一位如果是

1 相当十进制数的 1，第二位是 1 相当十进制数的 2，第三位是 1 相当十进制数的 4，第四位是 1 相当十进制数的 8。这样上面例题可简化如下：

$$(1111)_2=(8+4+2+1)_{10}=(15)_{10}$$

小数点后的二进制转化为十进制比较麻烦，只好靠查表或计算。例如

$$\begin{aligned}(111.111)_2&=(2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3})_{10}\\&=(4+2+1+0.5+0.25+0.125)_{10}\\&=(7.875)_{10}\end{aligned}$$

将十进制数转化为二进制数的方法，这里采用的是试减法。如

$$(90)_{10}=(?\ )_2$$

将90减掉保持其结果为正的最大的2^n，将剩余数再这样继续减下去，直至为 0，就可得出二进制数。

$$\begin{array}{r}90\\-64\rightarrow 2^6\\\hline 26\\-16\rightarrow 2^4\\\hline 10\\-8\rightarrow 2^3\\\hline 2\\-2\rightarrow 2^1\\\hline 0\end{array}$$

$$\begin{aligned}(90)_{10}&=64+16+8+2\\&=1\times 2^6+0\times 2^5+1\\&\quad\times 2^4+1\times 2^3+0\\&\quad\times 2^2+1\times 2^1+0\\&\quad\times 2^0=(1011010)_2\end{aligned}$$

整数十进制数转换为二进制数时，均可用有限位的二进制整数来表示，但对于小数来说，因为十进制有时会出现循环小数，所以十进制的小数就不一定能用有限位的二进制小数来表示。

表1.2　十进制数、二进制数与十六进制数的对照

十进制数	二进制数	十六进制数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

二进制数写起来很长，在表示指令和数据时都不方便，因此通常采用十六进制的方法来表示。十六进制每位数用0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F十六种符号的任一种表示，由低位向高位“逢十六进一”。表1.2为十进制、二进制、十六进制数的对照表。

每四位二进制数可用一位十六进制数表示，这样二进制数转换为十六进制数就很方便。

$$(11111111)_2=(FF)_{16}$$

反过来十六进制数转换为二进制数，将每位十六进制数写成四位二进制数就可以了。如

$$(A9)_{16}=(10101001)_2$$

十进制与十六进制的转换，可按表 1.2 仿照十进制与二进制的转换方法进行。

（二）二进制数的定点与浮点表示

在计算机中，用二进制表示数的方法有两种。一种是定点法，一种是浮点法。

所谓数的定点表示法，就是在计算机内数的小数点的位置是固定的，一般固定在数的最高位之前，例如0.0101。这样计算机所能表示的数总是小于1。当然，实际问题中数可能大于1，这就需要事先进行必要的加工，选择适当的比例因子，使全部参加运算的数都变为小于1。在有些计算机中，小数点的位置是固定在数的最低位之后，这样就只能表示大于1的数，小于1的数同样也要变换成大于1的数才能参加运算。

浮点表示是在计算机内数的小数点位置是“浮动”的，不是固定的。

任意一个二进制数N都可表示成

$$N=\pm S\times 2^{\pm J}$$

式中S称为尾数，它是一个二进制小数，表示数的有效数字。S前面的符号称为数符，在计算机中通常用尾数前的一位表示，“0”表示该数为正，“1”表示该数为负。式中J称为阶码，说明用S表示数N时小数点移动的位数。J前的符号称为阶符，当阶符为“+”时，将S的小数点向后移动J位即成为数N。阶符为“−”时，将S的小数点向前移动J位即为数N。例如

$$10011.101=+0.10011101\times 2^{+101}$$

在计算机中，浮点数表示的范围比定点数表示的范围大，但浮点数的运算要比定点数的运算复杂。

（三）二进制数的原码、反码、补码

一个数的正、负通常是用“+”、“−”符号来表示，由于计算机不能识别“+”、“−”符号，因此利用一个二进制符号位来表示数的正、负。Z80微型计算机字长为8位，最高位作符号位使用，最高位为0表示该数为正，最高位为1表示该数为负。如

01100111

11100111

前者表示+1100111，后者表示−1100111。

以上表示方法为原码表示，它简单易懂，但最大的缺点是加、减法运算复杂。当两个数相加时，机器必需首先判断两数的符号是否相同，如果相同则进行加法，如果不同则进行减法，在进行减法时，机器首先要比较两数绝对值的大小，然后由大数减去小数，最后还要给结果选择适当的符号。实现上述功能的计算机结构会变得特别复杂，因此实际上计算机中是采用补码表示的。

反码对正数来说和原码一样，对负数来说反码是除符号位外将原码各位数值取反（即0变1，1变0），如

原码　10001011

反码　11110100

补码是为解决用原码表示负数时在运算中出现的矛盾而建立起来的。补码对正数来说和原码一样，对负数来说等于它的反码在最低位上加1。如−7可写成

原码　10000111

反码　11111000

补码　11111001

这种方法也称负数的补码形式表示方法。

两数相减的过程是：被减数用原码表示，减数用补码表示，将两数相加，符号位看作数的一部分进行运算，相加的结果有进位时，表示两数之差是正数，此时应将进位丢掉，便得差值的原码。相加的结果无进位时，表示两数之差是负数，此时应将结果再次取补，便得差值的原码。举例如下：

$9-7=2$

00001001

+11111001←(取－7的补码)

00000010

其结果为＋2

$7-9=-2$

00000111

+11110111←(取－9的补码)

11111110

将结果取补为10000010，得出以上差值的原码为－2。

在计算机中使用这种方法实际上是将减法运算转变为加法运算。

(四) 逻 辑 运 算

逻辑运算已形成一种专门的数学，叫逻辑代数。它是描述逻辑电路的一种数学。而逻辑电路是构成计算机的基本电路。这里讨论计算机中常用的几种逻辑运算。

1. “与”运算

“与”运算用来描述图 1.2 “与”逻辑电路的逻辑关系。

两只开关 A、B 之间的关系称为“与”的关系。A、B 的接通状态记作 1，A、B 的断开状态记作 0。而其结果灯亮记作 1，灯灭记作 0。从图 1.2 可看出该电路的逻辑关系是当 A、B 两开关同时合上时电灯才会亮，一个开关不合或两个开关都不合电灯都不亮。以上逻辑关系可以用逻辑表达式表示如下：

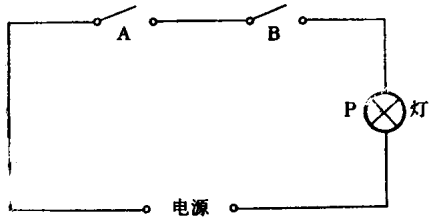

图1.2 “与”逻辑电路

$$A\cdot B=P$$

$$0\cdot 0=0$$

$$0\cdot 1=0$$

$$1\cdot 0=0$$

$$1\cdot 1=1$$

其中 A 和 B 相“与”写成 A·B，P 表示电灯状态。

2. “或”运算

“或”运算用来描述图 1.3 电路的逻辑关系。

从以上电路可看出其逻辑关系是：合上一个开关或合上两个开关电灯都亮，只有当两个开关都开着时电灯才是灭的。这种逻辑关系用“或”逻辑表示如下：

$$A + B = P$$
$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

其中A和B相“或”写成A+B，在逻辑“或”运算中要注意1+1=1的关系。

3. “非”运算

“非”运算用来描述图1.4电路的逻辑关系。

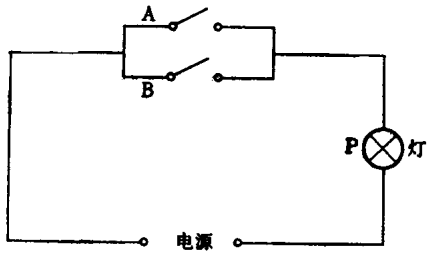

图1.3 “或”逻辑电路

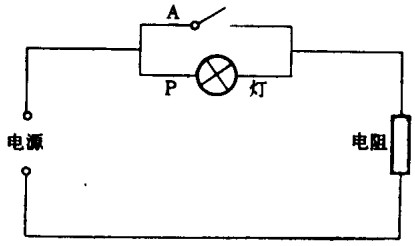

图1.4 “非”逻辑电路

在以上电路中，A开关合上，由于电灯有电阻，电流就从合上的开关中通过，因此电灯不亮；A开关打开电流从电灯通过，电灯亮。因此开关A的状态和P的状态正好相反。这种逻辑关系用“非”运算表示如下：

$$A = \bar{P}$$
$$A = 0 \qquad P = 1$$
$$A = 1 \qquad P = 0$$

这里$\bar{P}$读做“P非”

4. “异或”运算

“异或”运算的逻辑关系是当A、B两变量的状态相同时，其结果为0；当A、B状态不同时，其结果为1。这种逻辑关系可用“异或”运算表示如下：

$$A \oplus B = P$$
$$1 \oplus 1 = 0$$
$$0 \oplus 0 = 0$$
$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$

其中A与B“异或”写成A⊕B。

以上四种逻辑运算在Z80指令表中都有相应的指令。

四、Z80微处理器

无论大型计算机或微型计算机，中央处理器都是关键性的部件。

在微型机中，中央处理器又称微处理器。目前我国应用最广泛的微型计算机是以Z80微

处理器为核心制成的。

(一) Z80微处理器的主要指标

这里通过介绍Z80微处理器的指标，介绍一些计算机常用指标术语的概念。

1. 字长：8位

字长是指微处理器能够同时处理的最大的二进制位数。显然字长愈长，一次处理数据的位数就愈大，计算机精度就愈高，因此字长是计算机的一个重要指标。

下面还会经常提到字节的概念，通常以8位作为一个字节，本机器的字长等于一个字节。

2. 地址总线：16位

前面已谈到存贮单元有唯一确定的地址。在地址总线上存放着存贮器的地址。地址总线的位数愈多，可寻址的能力就愈强。每位地址总线只能存放1或0两种数，16位地址总线共能存放$2^{16}=65536$种存贮器地址，这就限制了存贮器的容量最多为65536单元，1024个存贮单元为1K，65536个存贮单元为64K。

3. 数据总线：8位双向

数据总线的条数一般与字长的位数相同。所谓双向是指输入、输出数据都使用同一数据总线。

4. 主频：Z80A为4MHz，Z80为2.5MHz

主频的倒数为微型计算机的时钟周期。指令的执行时间以此为基本单位来计算。对Z80A来说，时钟周期$T=250$ns。寄存器A中数据与寄存器B中数据相加指令需4个时钟周期完成，即1μs。因此主频决定计算机的工作速度。

5. 基本指令：158条

指令愈多，计算机的功能愈强。Z80与8080微处理器是兼容的，Z80的158条指令中有78条是8080微处理器指令。

(二) Z80微处理器的内部结构

图1.5为Z80微处理器的结构图。

由上图可看出微处理器由寄存器、运算器、控制器、总线缓冲器、电源等组成。

1. 寄存器

寄存器是微处理器的重要组成部分。它用来存放数据、中间结果、存贮器地址以及标志工作状态的信息等。

Z80微处理器有丰富的寄存器，因此它与同类型微处理器相比功能比较强。Z80微处理器包括18个8位寄存器，4个16位寄存器，2个8位暂存器，总共224位，如图1.6小框表示的寄存器为8位，大框表示的寄存器为16位。

(1)通用寄存器

Z80 CPU设置了两套通用寄存器。通用主寄存器组包括B、C、D、E、H、L 6个8位寄存器。通用辅助寄存器组包括B′、C′、D′、E′、H′、L′ 6个8位寄存器。它们完全是对应的。设置两套通用寄存器，可以在需要保存主寄存器状态时，将主寄存器组的全部信息同时传送给辅助寄存器组，这是Z80 CPU一种很强的功能。通用辅助寄存器组只能暂存数据，不能

直接参加运算，需要参加运算时，要将其数据倒换到主寄存器组。上述通用寄存器可组成寄存器对 BC、DE、HL、B′C′、D′E′、H′L′，每对寄存器中的数据常常用来表示存贮单元的地址。

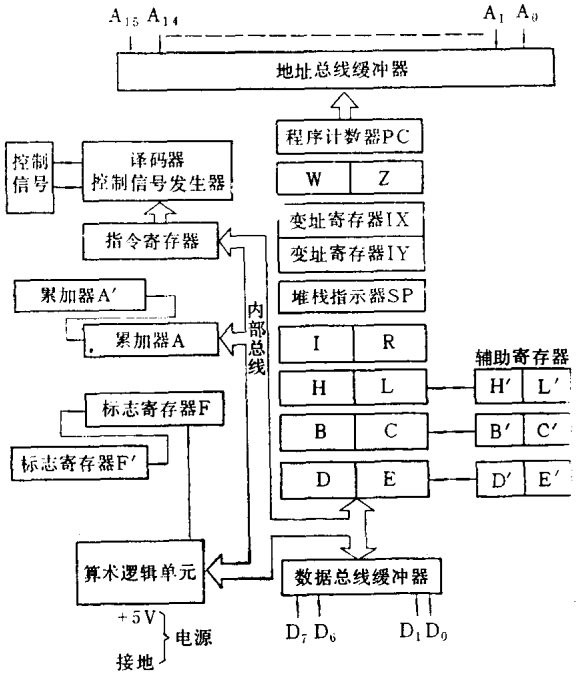

图1.5 Z80微处理器结构图

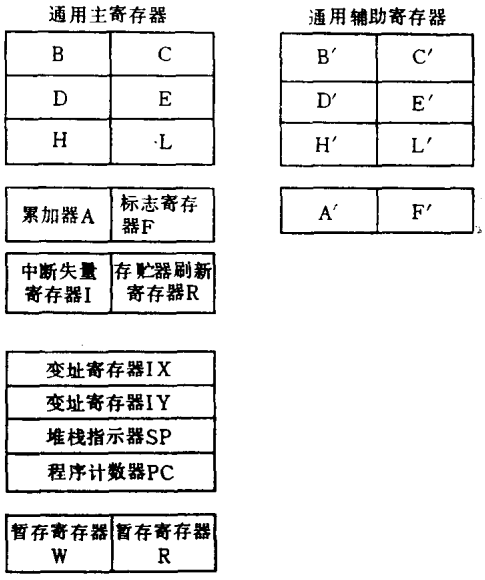

图1.6 Z80寄存器框图

数据存放在通用寄存器中比存放在存贮器中运算速度要快，因寄存器中的数据可直接参加运算，节省了从存贮器调用数据的时间。

（2）累加器与标志寄存器

Z80 CPU 包括两个 8 位累加器 A 与 A′，两个 8 位标志寄存器 F 与 F′。

当计算机进行算术与逻辑运算时，一个数据存放在累加器A中，另一个数据存放在 6 个通用主寄存器中的任一个，然后通过算术逻辑单元进行运算，运算的结果仍存放在累加器 A 中。A′为 A 的替换累加器，当 A 另有用途时，可将 A 中的数据倒换到 A′。

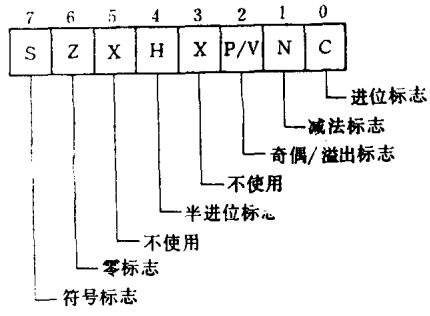

图1.7 标志寄存器

标志寄存器可给出运算结果的状态，程序可根据标志出的状态决定是继续还是转移。标志寄存器共 8 位，如图 1.7 所示。

标志寄存器从右数第一位为 0 位，最后一位为 7 位，标志寄存器各位功能如下：

0 位是进位标志。当进行加法运算时，结果有进位则 0 位置 1，当进行减法运算时，结果有借位，0 位也置 1。

1 位为减法标志。如果进行的运算是减法则此标志位置 1。在二进制与十进制变换中，前面一条指令是加法还是减法，计算机的处理不一样，因此需要此标志位给出信号，CPU

才能进行不同的处理。

2 位为奇偶/溢出位。这是一个具有双重用途的标志位。当进行逻辑运算时，它指出累加器A中运算结果的奇偶性。如果二进制逻辑运算结果中 1 的个数为偶数，则该标志位置1。例如：

$$\begin{array}{r} 10110101 \\ \text{AND } 10010101 \\ \hline 10010101 \end{array}$$

上面是两个数进行逻辑“与”，在结果中 1 的个数是偶数，则该标志位置 1。如果在逻辑运算的结果中，1 的个数为奇数，则该标志位置 0。

2 位的另一种用途是溢出标志。溢出是表示两个带符号的数在计算时出了错误。当计算出现错误时，该标志位置1，否则置0。累加器最高位为符号位，符号位为 0 表示正数，符号位为 1 表示负数。除符号位外表示数的范围只剩下 7 位了，7 位二进制能表示的最大数是 127，当两个数相加的结果超过此范围时，计算就会出错。例如：

$$\begin{array}{rr} +113 & 01110001 \\ +)\ +120 & +)\ 01111000 \\ \hline 233 & 11101001 \end{array}$$

符号位为 1，说明结果为负数，两个正数相加，结果出现负数，当然是运算中出现了错误，此时该标志位置 1。

3 位为空位，不表示什么状态。

4 位为半进位标志。在运算时从 3 位向 4 位有进位或借位，该标志位置 1，否则置 0。

5 位为空位。

6 位为零标志。当运算结果为 0 时，该标志位置 1，否则置 0。

7 位为符号位。当运算结果为负数，该标志位置 1，否则置 0。

在标志寄存器中除 1 位与 4 位是为了二进制与十进制变换服务的，3 位与 5 位是空位外，其他位都说明运算结果的状态。运算结果存在累加器A中，因此可以说这些位是按累加器A的状态来置位的。

（3）中断矢量寄存器

Z80 CPU 设置了 8 位中断矢量寄存器。

什么是中断？

微型计算机在进行运算的同时，还要去“照应”外部设备，如控制台、打字机等，近代的微型计算机是采取中断方式来完成这一任务的。外部设备的工作速度一般较慢，CPU 用不着停下自己的运算工作，等待外部设备输入数据。当外部设备已准备好数据要向计算机输入时，外部设备发出请求中断信号，CPU 就停止自己正在进行的运算工作，响应中断信号，转入接受控制台输入数据的中断服务子程序。当这一工作完毕后，CPU 又回到原来的运算工作。这就是计算机的中断概念。

中断矢量寄存器寄存着中断服务子程序的高 8 位地址，而中断服务子程序的低 8 位地址由请求中断的设备提供。

（4）存贮器刷新寄存器

存贮器刷新寄存器是 8 位寄存器。当存贮器采用半导体动态存贮器时，存贮器内的信息