

微计算机原理及其应用

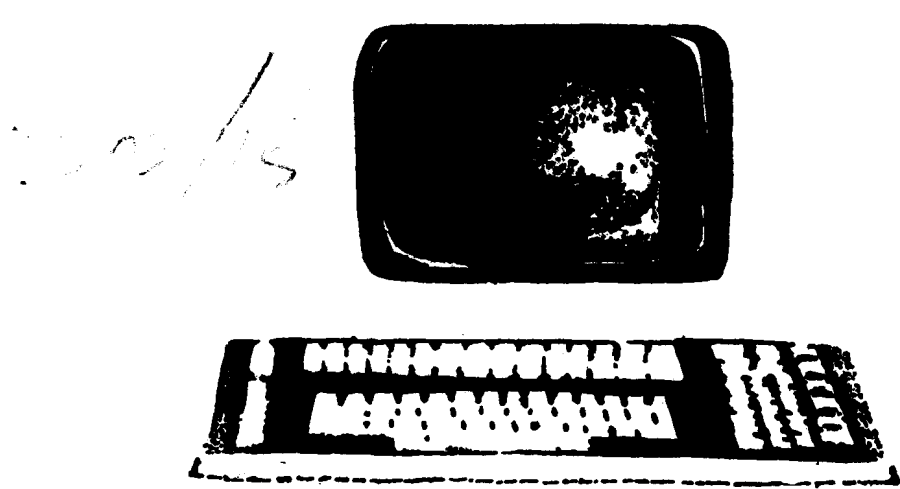
wei ji suan
ji yuan li
ji qi ying
yong

王长胤
鄢定明 编
刘大凯

湖北科学技术出版社

微计算机原理及其应用

王长胤 鄢定明 刘大凯 编



湖北科学出版社

内 容 提 要

本书以 Z-80 系列微计算机芯片功能及其应用为主线,着重介绍微处理机、接口电路及汇编语言程序设计的原理和应用,特别注重了实时控制和各种非数值计算方面的应用,并附有大量经过考验的汇编程序。

本书可作大专院校教材,尤其适于有初步数字电路知识的工程技术人员自学使用。

微计算机原理及其应用

王长胤 鄢定明 刘大凯 编

湖北科学技术出版社出版 湖北省新华书店发行
湖北省新华印刷厂印刷

787×1092 毫米 16 开本 30.75 印张 1 插页 779,000 字
1984 年 4 月第 1 版 1984 年 11 月第 4 次印刷
印数: 292001—412,000

统一书号: 15304·12 定价: 4.75 元

编写说明

近几年来,微计算机以人们料想不到的速度迅猛发展,它正在人们生活和生产的各个领域中获得愈来愈多的应用,成为科学技术现代化不可缺少的重要工具。一个学习计算机的热潮正在广泛兴起。人们迫切希望运用计算机的威力解决各自工作中碰到的技术问题,因而迫切需要一本有关计算机原理及其应用的书籍。为了适应这种形势的需要,湖北省暨武汉市通信学会组织我们编写了本书。

本书在湖北省暨武汉市通信学会两期微计算机学习班讲义的基础上修改而成。它可以作为大专院校的教材,也可以作为工程技术人员学习和应用微型计算机的主要参考书。

阅读本书的读者应具有数字电路的基本知识,对计算机的数制码制也应有所了解。

本书直接从CPU讲起,以便用较大的篇幅深入介绍计算机的有关原理和应用的知识。

本书放弃了把当代各种典型的微处理器逐一介绍的方案,而选择当前流行的Z-80作为样机深入地加以介绍,力求讲深讲透,选材尽可能完整、详尽、实用和系统。有些内容已经超过一般教材的深度,例如在汇编语言程序设计中,增加了浮点运算程序、函数运算程序;在接口部分,详尽地剖析了监控程序并把监控程序文本也附在书末等等。

本书深入浅出,注重实用。每一部分都列举了大量的实例,介绍应用的方法。开头的指令系统讲得很细,以便帮助读者尽快入门。接口部分没有详述大规模集成电路芯片内部的结构,而偏重于掌握这些电路的实际用法,特别强调了软件怎样通过硬件起作用,这对设计接口电路将是很有用的。书中引用的许多程序段和接口电路,一部分取材于国内外资料、手册和教科书,一部分则是直接从我们近年来教学、科研的成果中提取的。这些程序和电路均经过实际考验,可以作为读者应用计算机的借鉴。

书中的第一、二、三、四、九章由空军雷达学院王长胤副教授编写,第五、六、七、八章由华中工学院刘大凯同志编写,第十、十一、十二章由武汉邮电科学研究院鄢定明同志编写。王长胤副教授和华中工学院王飞龙副教授审阅了全书。

在湖北科学技术出版社的大力支持下,在省通信学会李蔚茹同志以及计算机专业委员会唐敏雄、李儒流、普大宽等同志直接领导下,在雷达学院计算机教研室和描图室、华中工学院信息传输与处理教研室、武汉邮电研究院计算机应用研究室的大力支持下,本书才能迅速与读者见面,在此一并表示诚挚的谢意。

由于我们学识水平有限,书中难免有错误和不妥之处,恳请读者批评指正。

编者

1984年1月

目 录

第一章 概论	(1)
第一节 引言	(1)
第二节 微处理机与微计算机	(9)
第二章 Z-80 中央处理机 (CPU)	(18)
第一节 概述	(18)
第二节 Z-80的组成及各部分的作用	(21)
第三节 CPU的定时信号	(31)
第三章 Z-80 指令系统	(40)
第一节 概述	(40)
第二节 数据传送指令	(44)
第三节 数据操作指令	(68)
第四节 程序控制指令	(92)
第五节 CPU 控制和位操作指令	(100)
第四章 存贮器	(111)
第一节 概述	(111)
第二节 静态随机存取存贮器 (RAM)	(112)
第三节 动态随机存取存贮器 (RAM)	(118)
第四节 只读存贮器 (ROM)	(124)
第五节 存贮器连接	(128)
第五章 汇编语言程序设计方法	(138)
第一节 Z-80 汇编程序的约定	(138)
第二节 汇编语言程序设计方法	(143)
第六章 算术运算程序设计	(181)
第一节 定点数算术运算程序设计	(181)
第二节 浮点数算术运算程序设计	(210)
第七章 基本函数程序设计	(232)
第一节 算法分析	(232)
第二节 三角函数程序设计	(233)
第三节 平方根程序设计	(242)
第四节 指数函数程序设计	(248)
第五节 对数函数程序设计	(255)
第八章 非数值操作的程序设计	(262)
第一节 代码转换	(262)
第二节 线性表的分类	(272)

第三节 查表技术	(277)
第九章 可编程接口芯片	(285)
第一节 概说	(285)
第二节 Z—80 CTC 接口芯片	(286)
第三节 Z—80 PIO 接口芯片	(298)
第四节 Z—80 SIO 接口芯片	(315)
第十章 外部设备及其管理	(339)
第一节 Z—80 单板机的组成	(339)
第二节 七段发光管显示器	(341)
第三节 键盘	(347)
第四节 EPROM 的写入设备	(352)
第五节 音频盒式磁带机	(356)
第十一章 单板计算机的监控程序	(374)
第一节 监控程序的作用	(374)
第二节 监控程序的结构	(378)
第三节 监控程序的主要内容	(383)
第四节 监控程序的应用与修改	(402)
第十二章 单板计算机的应用	(404)
第一节 TTL 集成电路测试器	(404)
第二节 用 PIO 完成一个工业顺序控制	(409)
第三节 测频仪的自动数据记录	(414)
第四节 游戏程序	(422)
第五节 数据采集和处理系统	(427)
附录一 监控程序表	(438)
附录二 ASCII 码表	(475)
附录三 Z—80 指令表	(475)
附 图 Z—80 单板计算机线路原理图	(485)

第一章 概 论

第一节 引 言

一、电子数字计算机的发展

电子数字计算机简称电子计算机，诞生在本世纪的四十年代。电子计算机的出现是本世纪的重大科学技术成就之一，它有力地推动了各门科学技术的发展，它的应用已深入到科学文化、工农业生产、国防建设甚至于家庭厨房，成为科学研究、工农业生产和社会生活所不可缺少的重要设备。电子计算机的应用程度成了衡量一个国家现代化的重要尺度。

三十多年来，电子计算机更新四代，大体上是按所使用的器件来分代的。第一代时间约在1946年至1957年，其主要特点是使用电子管作为逻辑元件，存贮器用延迟线或磁鼓，软件主要是机器语言，符号语言开始使用。例如1946年出现的第一台计算机ENIAC，使用了1880个电子管，占地150平方米，重30吨，耗电150瓩，价值40万美元，加法运算速度5000次/秒，与今天的微型计算机相比不可同日而语了。但是，它却奠定了电子计算机的技术基础。如采用二进制数进行运算和控制，建立了程序设计的概念等。第二代时间约在1958年至1964年，其主要特点是用晶体管取代了电子管作为逻辑元件，使用了磁芯存贮器，软件方面出现了高级程序设计语言，如ALGOL、FORTRAN，还提出了操作系统。这一代计算机除进行科学计算之外，在数据处理方面得到了广泛的应用，而且开始应用于过程控制。第二代计算机与第一代计算机相比在性能上与可靠性上都提高了一个数量级。第三代时间约在1965年至1972年，这一代计算机的主要特点是用中小规模集成电路取代了晶体管，存贮器仍使用磁芯。由于采用了集成电路，使计算机体积更小，耗电更省，可靠性更高了。在软件上，操作系统得到了进一步发展及普及，使计算机的使用更方便了。除大型机外，这一时期还生产了小型机和超小型机，机型多样化了，应用方面也遍布科学计算、数据处理及工业控制各种领域。第三代计算机在存贮容量、运算速度和可靠性等方面比第二代计算机又提高了一个数量级。第四代从1972年到现在，主要特点是大规模集成电路取代了中小规模集成电路作为逻辑部件，主存贮器也由大规模集成电路取代了磁芯存贮器，这样就有可能使计算机的主机装在一块硅片上，成为微处理机。在软件方面出现了与硬件相结合的趋势。

当前计算机正在向四个方面发展：

1. 微型化 由于大规模集成电路技术的发展，现在已经生产出一些单片机，在一块邮票大小的面积上装进了一部计算机(外设除外)，其性能不亚于传统的小型机。微型机的出现及其性能不断提高已经模糊了传统的大中小型机的概念，现在的微型机性能有的超过了过去的小型机或中型机，但体积却小得多且不要求严格的环境条件，价格低廉，大有取代中、小型机之势。微型机的出现开拓了计算机普及的新纪元。

2. 巨型机 当前计算机发展的另一个趋向是制造一些功能极强，运算速度特快的巨型计

算机，以满足尖端科学技术发展的需要。如 STAR—100 巨型计算机，其运算速度为每秒五千万次，存贮容量为 0.5~1.0 兆字，字长为 64 位。正在研制的有运算速度达每秒十亿次甚至百亿次的计算机。

3. 组成计算机网络 计算机网络是计算机发展的又一个重要方面，它以一 台 计算机为中心通过通信线路与多个终端相连成为一个系统，各个终端共享此计算机的硬件 和 软件 资源。另一种形式是多台计算机与多个终端通过通信线路连接起来成为一个系统，它们共享各计算机的硬件及软件资源。随着计算机微型化以及电视系统、激光技术和光导纤维等方面的发展，计算机网络会有更大的发展。

4. 智能机 智能机是在计算机与控制论等研究的基础上发展起来的一门新技术，它 能模拟人的智能，如识别图形、语言和物体等。机器人就是智能机的一种，它现在已被用在生产线上取代了一些人们的劳动。

二、计算机的组成

到今天为止，计算机虽然发展到了第四代，但就其组成而言统统称为冯·诺伊曼 (VON NEUMANN) 型计算机。它们都与 1946 年出现的第一台电子计算机大同小异，都由五大部分组成，这五大部分如图 1—1 所示。

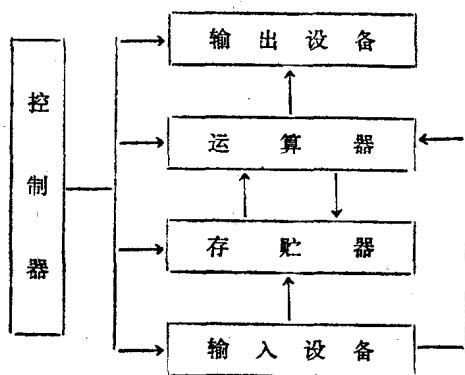


图 1—1 计算机组成框图

这五部分的作用简述如下：

1. 输入设备 是人与计算机进行交往的入口，常用的输入设备有扳键、键盘、光电或磁输入机等。无论哪一种输入设备都以二进制数或代码的形式向计算机输入，当前还不能直接用语言或图形向计算机输入数据、指令等。

2. 存贮器 相当于计算机的仓库，它有很多“房间”，这些“房间”都编上了号，称为地址。它们用来存放输入设备送来的代码或数据以及运算器送来的运算结果等。计算机的存贮器分内存贮器与外存贮器两大类，用作内存贮器的有磁芯存贮器和半导体存贮器，用作外存贮器的有磁带机、磁鼓和磁盘等。

3. 运算器 是计算机对各种信息进行算术运算和逻辑运算的主要部件，由很多逻辑电路组成，它们包括寄存器、加法器、移位器和一些控制电路等。

4. 输出设备 是计算机与人们交往的输出窗口，它把结果或各种信息以数字、字符图形等形式表示出来。常用的输出设备有打印机、凿孔机、数码管、显示器和绘图仪等。

5. 控制器 是计算机的指挥部，控制整个计算机自动地协调一致地工作。控制器由时序电路和逻辑电路组成。它对计算机的控制是通过输出的电压和脉冲信号来实施的。

在上面五大组成部分中，习惯上人们把运算器和控制器看成一个整体，称为中央处理机或中央处理器(CPU)，而把输入设备和输出设备以及外存贮器称为外部设备。有的人更进一步把除中央处理机外，统统称为外部设备，用 $\overline{\text{CPU}}$ 表示，这样 $\overline{\text{CPU}}$ 除前面讲过的外部设备之外，还包括了存贮器。

三、计算机字长

计算机内所有的信息都以二进制的代码形式来表示，这个二进制代码的位数称为计算机的字长。从需要来讲，计算机位数愈多，它能代表的数值就愈大，能表示的数值有效位数也愈多，计算的精度也愈高。但是，字长愈长，用来表示二进制代码的逻辑电路也愈多，使得计算机的结构变得庞大，电路变得复杂，造价变得昂贵。字长位数少，虽然能表示的数值范围小，数的有效位数也少，但计算机的结构简单，造价也低廉。计算机字长一般是根据计算机的性能及完成的任务不同而折衷决定。大型计算机字长一般都是 32 位或大于 32 位，中型机字长是 32 位或 16 位，而小型机几乎都是 16 位。微型计算机字长早期由于半导体制造技术的限制，一般是 4 位，以后是 8 位，现在已经能够造 16 位和 32 位微型机了，但是目前大量使用的微型机其字长还是 8 位。图 1-2 所示为一字长是 8 位的代码， D_0 表示最低位， D_7 表示最高位，每一位的数码只能是“0”或者是“1”。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	1	1	0	0	1

图 1-2 代码形式

(一) 代码内容 在计算机内一个字所表示的内容是多样的，可以用它来表示一个数，也可以用它表示一个信息。概括起来，计算机内一个字的代码可以是数据代码、字符代码、指令代码或地址码。至于一个代码表示的是数还是信息，从形式上是不能分辨的，但是，在计算机内它又是确定的，不能含糊的。读者通过进一步学习，能够在不同的应用场合把它们分辨出来。

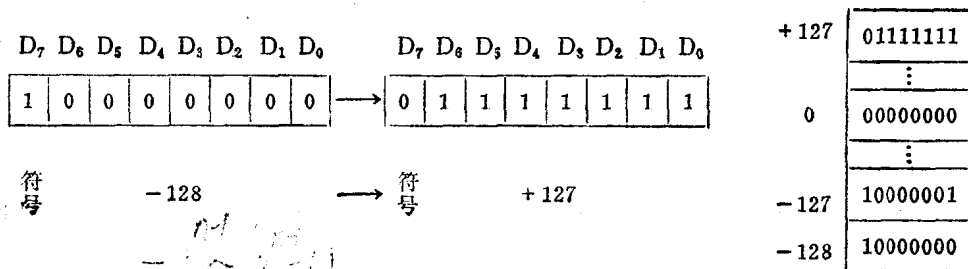
(二) 数据代码 二进制代码用来表示数据时，要解决可表示的数的范围和表示数的正负两个问题。前一个问题由采用定点数或浮点数的数据类型来解决。后一个问题是由原码和补码来解决。

1. 无符号整数：假若某计算机的字长是 8 位，用它来表示无符号整数时，其能表示的数值范围为 0~255。这是一个定点数，小数点定在 D_0 位之后，即：

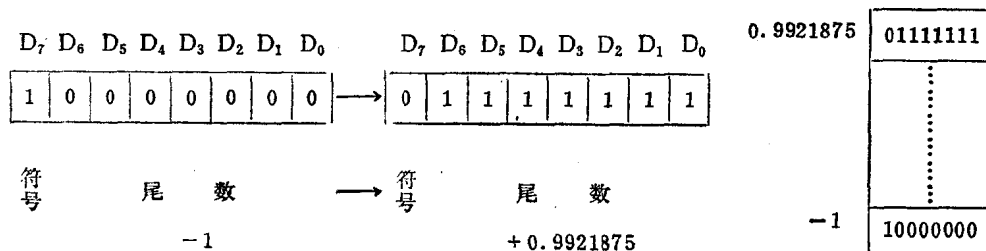
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
0	0	0	0	0	0	0	0	→	1	1	1	1	1	1	1	1	255	
																	255	11111111
																		⋮
																	1	00000001
																	0	00000000

0 → 255
(0~255)

2. 有符号整数：若字长同样为 8 位，用它表示有符号整数时，其采用的是定点补码码制，能表示的数值范围为 $-128 \sim +127$ 。这时 D_7 位代表符号， D_7 为“1”表示负数， D_7 为“0”表示正数，而表示数值的位数只剩下 7 位了，即：



3. 定点小数：若某计算机字长为 8 位，采用定点补码码制表示定点小数，其最高位 D_7 是符号位，小数点定在 D_7 与 D_6 之间，表示数的尾数是 $D_6 \sim D_0$ 7 位，它能表示的数值范围为 $-1 \sim +0.9921875$ ，可以近似认为 $-1 \sim +1$ ，即：



用二进制代码表示定点小数，它表示的数值范围不会超出 $-1 \sim +1$ ，代码位数愈多，只是表示的有效位数愈多，表示的数愈精确而已。象上面讲的字长 8 位时，其能表示的有效位数为 7 位十进制数。若字长为 16 位，其表示的数值范围为 $-1 \sim 0.999969482421875$ ，有效位数为 15 位十进制数。

由上述讨论可以看出，8 位字长能表示的数值太小了，表示无符号整数时最大只能表示 255，不能满足实际需要。如果字长仍是 8 位，而我们要求表示的数值大于 255 时，可以采用两个字或三个字去表示一个数，这种数据我们称为多字节数。图 1—3 所示是一个二字节数，即采用两个字表示一个数。

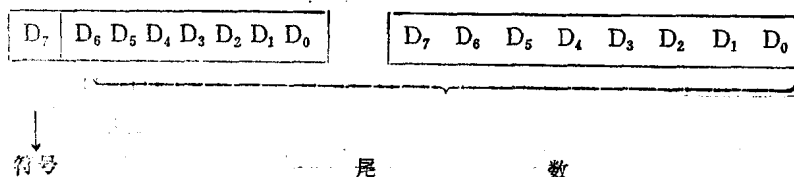


图 1—3 多字节数

$$1101011 = 1 + 2 \times 1 + 4 \times 0 + 8 \times 1 + 2^4 \times 0 + 2^5 \times 1 + 2^6 \times 1$$

$$= 1 + 2 + 8 + 32 + 64 = 107$$

假若用它表示无符号整数，16 位二进制代码能表示的数值范围为 0~65535，若用它表示有符号整数，则尾数有 15 位，能表示的数值范围为 -32768~+32767。显然，它表示的数值范围比 8 位大得多。

4. 浮点数：要表示的数很大或者要求有效位数很多时，常采用浮点数表示，浮点数通常都是多字节数。图 1—4 是一个四字节的浮点数的结构。最左边一个字节最高位 D_7 代表尾数的符号，“0”为正，“1”为负， D_6 位代表阶码的符号，“0”为负，“1”为正，阶码符号的规定与尾数不同。 $D_6 \sim D_5$ 位表示阶数；它为有符号的整数类型，后面的三个字节为尾数，它的数据类型为定点小数，小数点定在 D_7' 位之前。浮点数的结构没有统一的规定，在不同的计算机上采用不同的约定方式，图 1—4 的结构只是浮点数结构的一个例子。浮点数表示的数值范围与有效位数都比定点数大。用公式来表示，即：

$$N = 2^J \times S$$

式中 J 是阶码，S 为尾数。

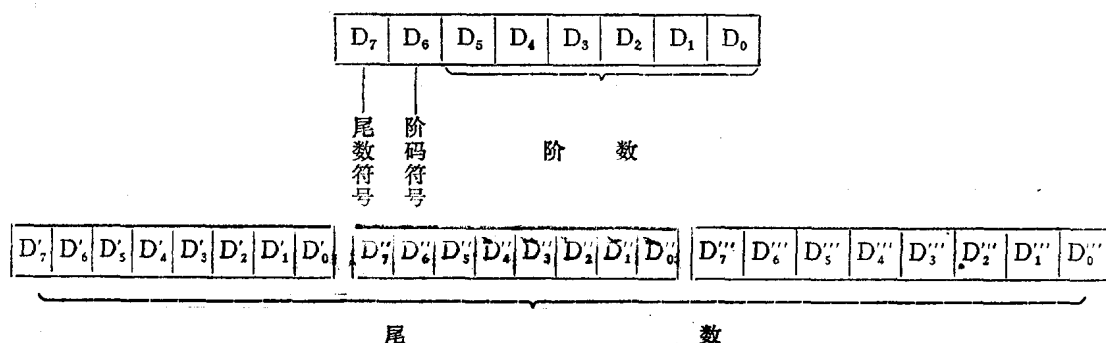


图 1—4 浮点数

(三) 字符代码 二进制代码也可用来表示字符。此时，它仅仅是一种编码的形式。例如计算机内常用的 ASCII (AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE) 码，它是美国国家标准信息交换代码，该代码由 7 位组成，可以表示的状态是 $2^7 = 128$ 种，因而可以代表 128 个字符。这些字符包括 0~9 十个十进制数，26 个英文字母以及一些控制符号，见附录。若计算机的字长是 8 位，那么一个字只能表示一个 ASCII 码其最高位或者空着不用，或者用来作为 ASCII 码的奇偶校验位。所谓奇偶校验位，就是保持代码“1”的个数总和是奇数还是偶数。如果我们采取偶校验，那么奇偶校验位的设置就要使 8 位代码中“1”的个数总和为偶数，如 ASCII 码 G，其代码为 1000111，原来“1”的个数总和为偶数，那么校验位就应填“0”，编码变为 01000111，保持“1”的个数总和为偶数。ASCII 码常用在计算机的输入，输出设备上，比如打印机，它只识别 ASCII 码，向它输出一个 01000111，它就打印一个“G”字，向它输出一个 00110011，它就打印一个“3”字。由于 ASCII 码不是真正的二进制数，所以用 ASCII 码向计算机输入数字时，必须换成真正的二进制数，计算机才能进行运算，运算后的结果再变成 ASCII 码，然后才能输出。在计算机内 ASCII 码与二进制数之间的变换是由程序来完成的。

(四) 代码书写形式 如某计算机字长为 8 位，写一个代码，就要写 8 个“0”或者“1”。这样不仅书写困难，不便阅读，而且容易出错。现在不论二进制代码用来代表数据、字符或者指令，一般都不用二进制的形式书写，而是用八进制或者十六进制的形式书写。因为八进制或十六进制的数与二进制数的互换非常方便，而八进制数或十六进制数的位数比二

进制数要少很多，书写起来就简单了。八进制和十六进制数与二进制数的互换关系见下表。八进制数由数字 0~7 来表示，一位八进制数等效于三位二进制数。十六进制数每一位有十六种状态，即 0~15，10 以上的数用字母 A、B、C、D、E、F 来代表。每一位十六进制数等效于 4 位二进制数。例如某计算机的字长为 8 位，它的某一个字的代码为 01000111，这个代码可以表示一个有符号或无符号的整数，也可以表示定点小数，还可以表示 ASCII 码 G。我们暂不管它所代表的内容是什么，从书写的角度来讲，可以写成三种形式，这三种形式完全是等效的：二进制“01000111”、八进制“107”、十六进制“47”。显然用八进制或十六进制书写就简单多了。但是上面写的 107 与 47，怎样知道前面的是八进制数，而后面的是十六进制数呢？人们在各种进制数的后面加上一个字母，以示区别：在二进制数后面加一字母 B (BINARY)，八进制数后面加一个字母 C (OCTAL)，十六进制数后面加一个字母 H (HEXADECIMAL)。

二 进 制 数	八 进 制 数
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

二 进 制 数	十 六 进 制 数
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

在计算机内也常用二进制代码表示十进制数，自然这个代码不是真正的二进制数，称为 BCD 码。为了区别起见，在 BCD 码后加一个字母 D (DECIMAL)，有时 BCD 码数也可省略不写字母 D。但八进制数与十六进制数后一定要加上字母 C 与 H。下表是各种进制数书写形式与二进制数的对照关系。在微型计算机内大量使用十六进制数的形式来书写各种代码。

各种进制形式	二进制形式
01011101 B	01011101
0 3 6 C	00011110
7 E H	01111110
6 8 D	01101000

01101000
 2 | 550500
 2 | 75250
 3 | 137625
 3 | 45875
 5 | 9775

四、指令

指令就是控制计算机进行各种操作和运算的代码形式的命令。对不同的计算机指令代码的编码规律是不同的，它们都有一套自己的代码指令，这套代码指令就叫做指令系统。指令从形式上看与数字代码完全一样，如果一部计算机采用的字长是8位，指令就是8位二进制代码了（有时可能多于8位）。

（一）指令格式 一条指令的代码是由操作码和地址码两部分组成，操作码部分在指令中的作用是规定计算机进行什么操作，而地址码部分则指明计算机进行操作的数到哪里去取，操作的结果放到哪里去。

（二）指令的地址码 地址码部分可能包括四个部分，即：

操作码	第一操作数地址	第二操作数地址	结果地址	下条指令地址
-----	---------	---------	------	--------

其中第一操作数地址是第一个数的取数地址，第二操作数地址是第二个数的取数地址，将两数运算后的结果，存到结果地址内，最后还指明下一条指令到哪里去取。这是最清楚的指令形式，可惜实际中很难实现，原因之一就是它占用的二进制代码的位数太多。为了克服上述困难，现在的计算机指令中都没有下条指令地址，在设计计算机时已约定，无特殊情况，指令就按顺序执行，即若第一条指令地址为D，第二条指令的地址就是D+1，这就是所谓隐含约定。特殊情况，若要求指令不按顺序执行，就需要向计算机提供下条指令的地址，如何提供，以后再讲。除去指令地址部分，地址码最多包括三部分，按地址码段数的多少来分，常见的指令有三种。下面用OP代表操作码，S代表源地址，D代表终地址。

1. 三地址指令：

OP	S ₁	S ₂	D
----	----------------	----------------	---

这条指令的动作是源地址S₁的内容与源地址S₂的内容进行操作，结果存到终地址D。我们常用地址码符号如S、D加一括号代表该地址内所存放的代码内容，这样上述三地址指令的操作用符号表示，就成为：

$$(S_1)OP(S_2) \rightarrow D$$

三地址指令的优点是操作所要的信息都包括在一条指令内，缺点是能代表的操作和访问的地址太少，效率低，在中小型机里皆不采用。

2. 双地址指令:



双地址指令的操作是源地址 S 的内容与终地址 D 的内容进行操作, 结果存到终地址 D 内。

$$(S) OP (D) \rightarrow D$$

3. 单地址指令:

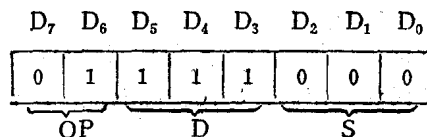


单地址指令指出了源地址, 而用隐含约定的办法规定另一个操作数放在运算器中的累加器 A 内, 那么它的操作就是:

$$(A) OP (S) \rightarrow A$$

与单地址指令比较, 双地址指令对两个数进行操作, 只要一条指令就行了, 单地址指令则要两条指令, 需要指令条数多, 计算的速度就慢, 从这点讲, 双地址指令答题的速度快。但单地址指令因分的段数少, 能表示较多的操作种类和地址, 功能较双地址指令强。这两种指令格式在 Z-80 微处理机内都被选用。在 Z-80 微处理机里地址码 S、D 指示的地址有三类: 运算器中的各寄存器、存储器的不同单元和外围设备的各个不同接口, 也就是说它可以在运算器的寄存器、存储器和外围设备之间进行数据传递和操作运算。

(三) 指令的书写形式 在介绍指令格式时, 我们用框图表示指令, 实际上指令是用代码书写的, 例如下面 8 位二进制代码就是 Z-80 微处理机的一条指令, 它是一条双地址指令, 这条指令的操作是将运算器内 B 寄存器(地址 S)的内容传送到 A 寄存器(地址 D), 操作码只有二位, 即“01”。用符号表示, 可以写成:



$$(B) \Rightarrow A$$

1. 机器码: 指令代码可以用二进制代码书写, 也可以用八进制、十六进制代码书写, 象上面那条指令就可以写成: 二进制形式“01111000 B”、八进制形式“170 C”、十六进制形式“78 H”。凡是用二进制、八进制或者十六进制书写的指令代码, 我们称为机器码。机器码是计算机能够识别和执行的唯一代码, 但是用机器码书写指令首先要熟悉指令的编码规律, 这需要专门的训练, 其次用机器码书写指令不形象, 不易识别, 有了错误也不易发现。为了克服这些缺点, 后来发展成用符号来书写指令。

2. 助记符: 是用英文缩写字母表示指令的符号。例如“加法”英文为 ADDITION, 记为 ADD, “减法”英文是 SUBTRACTION, 记为 SUB, 再如数据装入的英文是 LOAD,

记为 LD。下面是三条指令的机器码与助记符形式的对比：

0	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---

 LD A, B (B) $\Rightarrow$ A

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

 ADD A, B (A) + (B) $\Rightarrow$ A

1	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---

 SUB B (A) - (B) $\Rightarrow$ A

助记符与机器码是一一对应的。显然用助记符书写指令较之机器码要形象得多，便于识读，有错也便于发现和修改，因此，助记符形式书写指令被广泛采用。但这种形式的缺点是计算机不能识读，必须将助记符转换成机器码才能输入计算机，这个变换称为代真。现在可以由计算机自己进行这项翻译工作了。

第二节 微处理机与微计算机

一、微处理机与微计算机的定义

(一) 微处理机 MP (MICROPROCESSOR) 微处理机就是传统计算机的 CPU，由于半导体集成技术的发展，将运算器和控制器集成在一小块硅片上，成为一个独立的器件称为微处理器或微处理机。微处理机是半导体技术与计算机技术结合的产物，它的出现是计算机发展过程中的一个里程碑。

(二) 微计算机 MC (MICROCOMPUTER) 以微处理机为中心配上存储器和外设接口电路构成的整体，称为微计算机。如果把微处理机、存储器以及接口电路集成在一块芯片上，就是单片机。微计算机一般不包括外设和软件。

(三) 微计算机系统 MCS (MICRO COMPUTER SYSTEM) 是以微计算机为中心配上外围设备、电源和软件等，能独立工作的完整计算机。微计算机系统根据配上的外设多少，软件功能强弱而分成大系统和小系统。大系统和小系统之间没有明确的定义和界限。

二、微处理机发展历史

微处理机是第四代计算机产品，自 1971 年诞生以来，发展很快。就微处理机本身来讲，几乎是每 2~3 年要换一代。国外生产微处理机的厂商很多，例如英特尔 (INTEL) 公司、莫特洛拉 (MOTOLORA) 公司和泽洛格 (ZILOG) 公司。它们的产品即为 INTEL 系列、MOTOLORA 系列和 ZILOG 系列。下面以英特尔公司的产品为例，列表说明微处理机发展的过程。

生产年代	型 号	字 长	集 成 度 3 mm × 4 mm
1 9 7 1	INTEL 4040	4 位	2000 只管子
1 9 7 4	INTEL 8080	8 位	5000 只管子
1 9 7 6	INTEL 8085	8 位	9000 只管子
1 9 7 8	INTEL 8086	16位	29000 只管子
1 9 8 0	IAPX 43201	32位	10 万只管子

ZILOG 公司是 1976 年由 INTEL 公司分裂出来的一个公司，在 1976 年向市场推出了 Z-80 产品，字长为 8 位，其性能与 INTEL 8080 A 相当。MOTOLORA 公司在 1975 年生产出 M 6800 产品，字长也是 8 位，与 INTEL 8080 A 相当。1978 年后三个公司都生产出了 16 位微处理器，它们的型号分别为 8086、Z-8000、M 68000。

微处理机的改进，主要在下面几个方面：

(一) 字长 小型机字长一般为 16 位，而微处理器字长初期只能做到 4 位和 8 位，因而性能不能与小型机媲美。限制字长不能增多的主要原因有两条：一是制造技术的限制，不能在一块硅片上集成更多的器件；二是封装的限制。当时双列直插式封装只能是 40 条腿或 42 条腿，别的封装就没有测试仪器可以利用。后来由于集成技术不断提高，现在已经能够生产出 16 位、32 位的微处理器，但是封装有的仍用 40 条腿，因而 16 位地址线和 16 位数据线分时并用。

(二) 速度 微型机要与中小型机竞争，就必须设法提高速度。微处理器大都是用 MOS 电路制成，其主要缺点是工作速度较低，初期的微处理器(8008)指令执行时间为 20 μ s。第二代微处理器(8080 A)执行一条指令时间大约在 2 μ s~8 μ s 之间。这已经可以抵得上小型机的速度。第三代微处理器(8086)执行一条指令时间大约是 0.5 μ s~2 μ s，已经快于一般的小型机了。提高速度主要是采用了 NMOS 和 HMOS 电路的结果。

(三) 集成度 原来在一块硅片上只能做出 4 位微处理器，发展到今天已经能在一块硅片上做出 32 位微出机。1977 年出现了单片微计算机，即在一块硅片上不仅能装下处理器，而且连存储器 and 接口电路都包括进去。

现代微型机的性能已经达到了象 PDP—11/70 这样中型机的水平。目前世界上微型机发展很快，应用很广。1980 年美国生产了 1000 万块微处理器，生产了 75 万台微型计算机，在价格上也大幅度下降。六十年代一部小型机售价 100 万美元，现在性能同其差不多的微型计算机售价只有 1000 美元，微处理机的零售价格约 15~30 美元，随着生产技术的改进，其价格每年都在下跌。

三、微计算机系统的组成

图 1—5 是一个微型计算机系统的组成示意图。图中微处理器采用 Z-80，存储器采用半导体存储器 RAM，外存储器采用磁盘。输入设备是键盘，输出设备有显示器和打印机。这个系统由于外围设备不多，人们称它为小系统，比图 1—5 小系统更小的系统还有单板机。单板机就是计算机的所有设备都装在一块印刷电路板上，图 1—6 就是某单板机的安装图，图 1—7 是它的组成方块图。

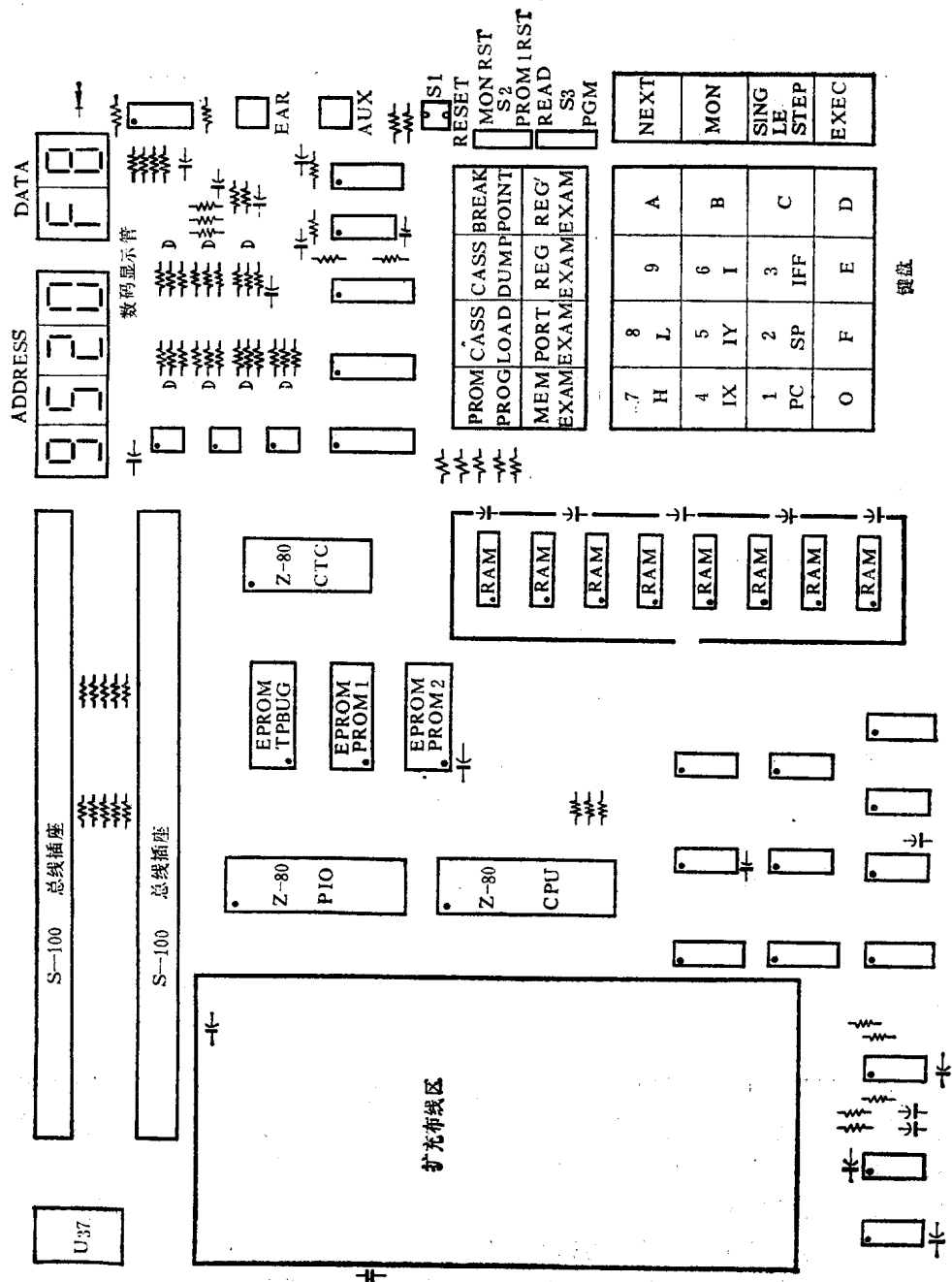


图 1—6 TP 801—Z 80 单板计算机安装图