高等学校工程专科教材

微型计算机原理及应用

唐俊杰　高秦生　俞光昀　编著

高等教育出版社

高等学校工程专科教材

微型计算机原理及应用

唐俊杰　高秦生　俞光昀　编著

高等教育出版社

(京)112号

内 容 提 要

本书是根据全国高等工程专科计算机基础课教材编审组审定的编写大纲编写的，并经教材编审组审定、推荐出版的.

本书主要内容：计算机基础知识；MCS－51单片计算机的基本原理和应用；IBM－PC计算机系统简介；MCS－51单片计算机实验．每章配有习题和思考题.

全书在编写中注重理论与实践的结合，精心设计了10个实验题目，使本书从理论到实践构成了一个完整的体系，并为教学提供了方便.

本书可作为高等工程专科学校、职大、夜大计算机课程的教材，亦可供工程技术人员学习参考.

图书在版编目(CIP)数据

微型计算机原理及应用/唐俊杰等编著. -北京：高等教育出版社，1993.11(1998重印)

ISBN 7-04-004352-1

Ⅰ.微… Ⅱ.唐… Ⅲ.微型计算机-高等学校-教材 N TP36

中国版本图书馆CIP数据核字(95)第13192号

*

高等教育出版社出版

新华书店总店北京科技发行所发行

高等教育出版社印刷厂印装

*

开本 787×1092 1/16 印张 25.25 插页 1 字数 620 000

1993年11月第1版 1998年5月第6次印刷

印数 36 941—44 950

定价 19.00元

前　言

本教材是根据全国高等工程专科计算机基础课教材编审组审定的编写大纲编写的，由国家教育委员会高等工程专科基础课及技术基础课教材编审组推荐出版.

本书主要内容:第一部分为计算机基础知识(第一章)；第二部分为MCS－51单片计算机的基本原理与应用知识；第三部分为IBM－PC个人计算机系统简介；第四部分为MCS－51单片机实验．本课程的参考学时为90学时.

全书在编写中力求贯彻在掌握基本知识和基本理论的基础上，加强应用和注重实践的指导思想。本书可供非计算机类专业学生使用，也可作为其它专业学生及从事自动化工作的工程技术人员学习单片机原理及应用技术的参考书.

本书由上海机械专科学校唐俊杰同志编写第一、五、六、七、十一章；南京机械专科学校高秦生同志编写第二、三、四、八、九、十章，附录和绪论；南京化工动力专科学校俞光昀同志、李绮红同志编写实验部分.全书由高秦生同志统稿.

在成书过程中，编著者得到全国高等工程专科计算机基础课教材编审组全体编委的具体指导和热情帮助;也得到全国许多高等专科学校同行专家的鼎力帮助;干敏梁同志、胡汉才同志、王连民同志审阅了初稿，提出了许多宝贵意见；全书由吉林电气化高等专科学校刘天赐同志担任主审、杨国慧同志参加了主审工作，上海纺织专科学校李士允同志为责任编委，在此一并表示衷心的感谢.

由于编著者水平有限，漏误之处在所难免，敬请同行专家和广大读者批评指正。

编　者

1992年7月

绪　论

一

计算机是人类有史以来最伟大的发明之一，它也是迄今为止，文明社会中最有价值的工具之一，是人类智慧的结晶.

人类经过几个世纪的努力，把计算器从中国古老的算盘发展到当代的计算机.现在，电子计算机已经广泛应用到社会生活的各个领域，促进现代文明的进步，推动人类社会的发展.

从宇宙飞船到人造卫星；从天气预报到地震预报；从办公室自动化到生产过程自动化……这一切如果没有计算机，都是无法实现的.

现在，计算机不仅仅是计算的工具，它已经成为高效率的"智能工具";它再也不是少数科学家的宠儿，已经成为广大知识群众手中的工具；它再也不是少数实验室的宝贝，已经在各行各业大显身手.

二

计算机不是某个人突发奇想的产物，也不是社会发展突然提供了某种机遇.它经历了漫长的发展过程，几代人为此付出了毕生的精力.

计算机作为一种计算的工具,它的雏型，最早可以追溯到中国古老的算盘;可以追溯到17世纪中叶，由法国著名数学家巴斯卡尔(B.Pascal)发明并制造的机械式的加法器；可以追溯到17世纪后半期由德国数学家莱布尼兹(G.W.Leibniz)系统地提出了二进制算术运算法则和由他主持设计制造的通过齿轮传动的机械式计算器；可以追溯到19世纪初由英国数学家巴比吉(C.Babbage)提出并试图制造的机械式"分析机";……;踏着这些计算机先驱者所开拓的道路,到20世纪,计算机进入了迅猛发展的时代.

世界上第一代电子计算机是以在1946年诞生的全真空管化的电子数字计算机ENIAC为标志的.它是美国设计师埃克特(P.Eckert)和莫克利(W.Mauchly)在宾夕法尼亚大学制造成功的.

晶体管的发明为计算机的换代提供了契机.

1958年，IBM公司宣布制成并投入商业化生产的全晶体管化的计算机，开始了第二代计算机(晶体管为逻辑元件)蓬勃发展的时期.这个时期持续到60年代中期(1964年)结束.

集成电路的问世，又很快地被用于计算机的制造.以集成电路取代分立元件，开始了第三代计算机的发展历程.这个阶段是以1964年4月IBM公司宣布IBM360系列计算机问世为起点的.

进入20世纪70年代，微电子技术取得了巨大成就，大规模集成电路和微处理器应运而生.它们给计算机发展注入了新的活力，计算机开始了大规模集成电路时代——即第四代计算机.大规模集成电路微处理器是以1971年美国Intel公司的青年科学家霍夫(M.E.Hoff)发明

第一块微处理器4004为标志的.现在，以各种大规模、超大规模集成电路制成的各种计算机系统已经得到普遍而广泛的使用.其功能之强、体积之小、价格之廉、可靠性之高，是当年的ENIAC所无法比拟的.

现在，人们又在向第五代计算机进军，它将是一代新型计算机，它的目标是向更高智能化的方向发展.

三

近些年来，计算机发展的道路越来越宽广. 一方面大规模集成电路微处理器向着更高集成度方向发展，出现了象Intel 80486这样的32位微处理器芯片；一方面随着专用集成电路ASIC(Applied Specific Integrated Circuit)技术和压缩(或精简)指令集计算机RISC(Reduced Instruction－Set Compater)技术的发展，产生了ASIC型单片计算机.

单片机开拓了计算机发展的另一条康庄大道. 它把构成一个完整计算机的更多的功能部件(中央处理器、存储器、I/O部件、可编程逻辑阵列等)集成在一块芯片上，从而使计算机硬件系统的构成更加简单，使计算机的二次开发更易于实现.这就又为计算机的普及应用开辟了新的道路.

当前，单片机发展的另一特点是，以一块单片机母片为核(例如80C51)在单片内嵌入更多的功能，派生出一系列具有特殊功能的芯片. 根据单片机的这一特点，所以近年来，又称单片机为嵌入式控制器(Embedded Controller).

不仅如此，嵌入的概念还进一步扩展到软件中了，即可以把操作系统软件和执行速度很快的高级语言固化到单片机的ROM内，成为嵌入式软件. 软件和硬件同时嵌入，使单片机真正成为嵌入式微控制器.

同微处理器一样，目前单片机也形成了多品种多系列的繁荣局面.单片机也有4位、8位、16位、32位几个档次的产品.它们各具特点，可以适应不同层次的要求.

正是由于单片机的上述特点，使它特别适合于工业控制领域的应用.因此，当前单片计算机在工业测控领域内的应用呈现出一派繁荣的景象.

当前，在我国以8位单片机的应用最为普及，最有代表性. 因此，本书以8位单片机为主(MCS－51系列)，阐述单片机的基本原理和应用规律，并通过它，使读者进而对微处理器及个人计算机的基本原理及其应用有所了解.

四

现代计算机有两个发展空间：一个是计算机科学技术本身的发展，一个是计算机在各行各业的应用. 对于大多数专业的学生来说，学习微型计算机技术的目的在于应用.

就应用的形式而言，基本上有两大类：应用系统的开发和应用软件的编制.当然应用系统的开发就其概念来说，也包含了应用软件的编制，但在这里我们把它局限于构成一个适合某种应用要求的微型计算机应用系统.

微型计算机应用系统的开发和应用软件的开发,两者既有密切的联系，各自又可以独立的发展. 现代，它们已经发展成为两大高技术产业.

微型计算机应用系统的开发又有多种形式，常用的形式有：

• 利用现有系统，例如利用IBM－PC个人计算机，在硬件上加以一定的扩展，构成一

个应用系统;

- 利用具有各种不同功能的模板，组成一个模块化的计算机应用系统;
- 利用微处理器芯片和若干接口芯片，构成一个微机应用系统 ……

从这里我们可以看出，学习微机应用的着眼点应该放到微机系统的构成上.因此，学习的重点是了解微处理器及各种接口芯片的内部结构及性能，掌握它们的外部特性及连接方法以及它们的编程特点等.在学习过程中要避免学习了许多这样那样的计算机芯片，而对如何构成一个微机系统却一无所知.

就软件而言，最重要的是要充分认识软件的重要性. 它不仅在于现代计算机已经无法离开软件而工作，而且在于软件已经发展成为一个巨大的产业. 这是一个广阔的天地. 因此，在学习微型计算机应用技术的时候，认真学好汇编语言程序设计是十分重要的.

为了在计算机应用方面有所成就，也为了提高计算机应用开发的效率，应该注意以下几点:

1. 了解最新的技术进展，及时掌握市场行情，跟上技术发展的步伐.
2. 尽可能利用现代的技术成果，包括软件成果.
3. 与各行各业的应用需要和技术发展相结合.

第一章　微型计算机基础

§1.1　微型机中常用的数制与编码

计算机、简言之，就是自动、高速、连续处理数据的机器．虽然计算机问世以来经历了四代的发展变化，但它的基本原理仍没有改变.迄今为止，计算机是以二进制运算为基础的，与其相应，构成计算机本体的是以二进制为基础的数字逻辑电路．为了书写方便、简化表达式，在计算机中又引进了八进制数、十六进制数等.同时，计算机的数据输入和输出形式必须与人们日常使用的进位制(十进制)相一致.这样在计算机中就存在着多种进位制及由此而引出的有关编码.

1.1.1　微型机中常用的数制及其相互转换

一、进位计数制

将数码按先后位置排列成数位，并按由低到高的进位方式进行计数就称为进位计数制.

每种进位计数制都有两个基本要素，即基数和位权.基数是指计数制中所用到的数码个数．如人们习惯使用的十进制数,它所使用的数码是0,1,…,9等十个，它的基数就是10.也就是说，在一个十进制数中的每一位，只能取0,1,…,9等十个数码之一．另外，对于一种进位制来说，同一个数码在不同的数位时，其值不等．例如十进制数1991.5,在其整数部分有两个1和两个9,它们所表示的数值是不等的.第一个1表示$1\times10^3=1000$;第二个1表示$1\times100^0=1$;第一个9表示$9\times10^2=900$,而第二个9表示$9\times10^1=90$.也就是说，每个数码处在某个数位时,它代表的数值是该数码本身的值乘以与所处的数位相关的一个固定常数(本例中第一个1的固定常数为1000,第二个1的固定常数为1)，这个不同数位的固定常数称为"位权",简称权.它是一个指数，它的底是10,幂是数位的序数减1.因此，十进制数1991.5可以展开成:

$$1991.5=1\times10^3+9\times10^2+9\times10^1+1\times10^0+5\times10^{-1}$$

其一般形式为:

$$S_{10}=\sum_{i=-m}^{n-1}K_i10^i$$

其中K_i为第i位的数码，n为整数的位数，m为小数的位数.

上式可推广到任何进位计数制中，只要将相应进位计数制的底代替上述的10,并确定K的数码集合.

二、常用数制

1.二进制数

在二进制数中只有0与1两个数码．因此，它的基数为2.二进制数可表示为:

$$S_2=\sum_{i=-m}^{n-1}K_i2^i$$

其中 K_i 可取 0 或 1. 在二进制数的加减运算中逢二进一，借一为二. 例如，二进制数 1011.11 可按权展开为:

$$(1011.11)_2=1\times2^3+0\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}.$$

众所周知，计算机只能直接对二进制数进行传送、存储、运算及加工处理. 在计算机中采用二进制数的主要原因是: 第一、用二进制表示便于物理实现(由于二进制只有0与1两个数码，因此可用任何具有两个不同稳定状态的元器件表示); 第二、二进制算术运算规则简单; 第三、可借助于布尔代数进行化简.

但二进制数不直观，书写不方便.

2. 八进制数

在八进制数中采用 0, 1, …, 7 等八个数码. 八进制数可表示为:

$$S_8=\sum_{i=-m}^{n-1}K_i8^i$$

其中 K_i 可取 0, 1, …, 7 等八个数码之一. 在八进制数加减法中“逢八进一”、“借一当八”.

3. 十六进制数

在十六进制数中采用 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F 等 16 个数码，其中 A ～ F 相应的十进制数为 10 ～ 15. 十六进制数可表示为

$$S_{16}=\sum_{i=-m}^{n-1}K_i16^i$$

其中 K_i 可取 0, 1, 2, …, F 等 16 个数码之一. 在十六进制数加减法中“逢十六进一”、“借一当十六”. 由于微型机中存储单元的位数大多是 4 的整数倍，故采用十六进制数书写程序可以简化数据的表示(相对二进制而言).

在计算机系统中约定，二进制数以 B(B 是 Binary 的缩写)作为后缀; 八进制数以 Q(Octal 的缩写 应为O 为了区别于 0 故写为 Q)作为后缀; 十六进制数以 H(Hexadecimal 的缩写)作为后缀; 十进制数可以 D (Decimal 的缩写)作为后缀也可不加后缀.

表 1－1 给出了上述四种进位制的 0 ～ 16 的数码对照表.

表 1－1 四种进位制的数码对照

十进制	二进制	八进制	十六进制
0	0000B	0Q	0H
1	0001B	1Q	1H
2	0010B	2Q	2H
3	0011B	3Q	3H
4	0100B	4Q	4H
5	0101B	5Q	5H
6	0110B	6Q	6H
7	0111B	7Q	7H
8	1000B	10Q	8H
9	1001B	11Q	9H
10	1010B	12Q	AH
11	1011B	13Q	BH
12	1100B	14Q	CH

续表

十进制	二进制	八进制	十六进制
13	1101B	15Q	DH
14	1110B	16Q	EH
15	1111B	17Q	FH
16	10000B	20 Q	10H

三、常用数制之间的转换

1. 十六进制数与二进制数之间的转换

由于四位二进制数位为一位十六进制数位，所以它们之间的转换相当方便. 如果要将十六进制数转换成二进制数，可将每个十六进制数位化成相应的四位二进制数位(如为纯小数时，小数点左面的 0 仍为一位，不要化成四位). 如要将二进制数转换成十六进制数，则以小数点为基准，向左、右以每四位二进制数位为一组(不够四位者添 0,以补足四位)，然后把每四位二进制数位用相应的一位十六进制数位表示.

例 1－1 试将1A2F.7BH 转换成二进制数.

解: 将每个十六进制数位化成相应的四位二进制数位.

1	A	2	F	.	7	B
0001	1010	0010	1111	.	0111	1011

即1A2F. 7BH＝1101000101111.01111011B

例 1－2 试将1011101011.0110111B 转换成十六进制数.

解:

0010	1110	1011	.	0110	1110
2	E	B	.	6	E

即 1011101011.0110111B＝2EB. 6EH

2. 二进制数(或十六进制数)转换成十进制数

二进制数(或十六进制数)转换成十进制数时，可采用“按权相加”法，即将二进制数(或十六进制数)按权展开相加，可得到相应的十进制数.

例 1－3 试将101011B 转换成十进制数.

解:

$101011B = 1\times2^5+0\times2^4+1\times2^3+0\times2^2+1\times2^1+1\times2^0 = 43$

例 1－4 试将10001.1101B 转换成十进制数.

解: $10001.1101B = 1\times2^4+0\times2^3+0\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}=17.8125D$

后缀 D 可省略.

3. 十进制数转换成二进制数(或十六进制数)

当将十进制数转换成二进制数(或十六进制数)时，因整数部分的转换与小数部分的转换方法不同，故需分别进行.

整数部分的转换可采用“除 2(或 16)取余”法，即用 2(或 16)不断地去除要转换的十进制数，

直到商为0时为止．将所得各次余数，以最后所得余数为最高数位，最先所得余数为最低数位的顺序排列，即为相应的二进制数(或十六进制数)．

小数部分的转换可采用"乘2(或16)取整"法，即用2(或16)不断地去乘要转换的十进制纯小数，直到满足精度或小数部分等于0为止．将各次所得的乘积整数以最先所得的为最高数位，最后所得的为最低数位的顺序排列，即为相应的二进制数(或十六进制数)．

例1－5 试将115D转换成二进制数．

解：采用除2取余法．

```
2 | 115
  2 | 57   …  余数 = 1(最低数位)
    2 | 28 …  余数 = 1
      2 | 14 …  余数 = 0
        2 | 7 …  余数 = 0
          2 | 3 …  余数 = 1
            2 | 1 …  余数 = 1
              0   …  余数 = 1(最高数位)
```

即 115D＝1110011B

例1－6 试将0.5803D转换成二进制数，且要求有效位数为6位．

解：采用乘2取整法．

```
  0.5803
×      2
----------
  1.1606   … 整数 = 1(最高数位)
  0.1606
×      2
----------
  0.3212   … 整数 = 0
×      2
----------
  0.6424   … 整数 = 0
×      2
----------
  1.2848   … 整数 = 1
  0.2848
×      2
----------
  0.5646   … 整数 = 0
×      2
----------
  1.1392   … 整数 = 1(最低数位)
```

即 0.5803D＝0.100101B

十－十六进制数的互相转换与二－十进制数的，互相转换的方法是相似的．例如，将2446及0.65625转换成相应的十六进制数．

$$
\begin{array}{r|l}
16 & 2446 \\
\hline
16 & 152 \quad \cdots \text{余数} = 14 \quad (\text{最低数位}) \\
\hline
16 & 9 \quad \cdots \text{余数} = 8 \\
\hline
0 & \cdots \text{余数} = 9 \quad (\text{最高数位})
\end{array}
$$

即 2446 = 98EH

$$
\begin{array}{r}
0.65625 \\
\times \quad 16 \\
\hline
10.50000 \quad \cdots \text{整数} = 10 \quad (\text{高数位}) \\
0.50000 \\
\times \quad 16 \\
\hline
8.00000 \quad \cdots \text{整数} = 8 \quad (\text{低数位})
\end{array}
$$

即 0.65625 = 0.A8H

三种常用的数制之间的转换方法可用图 1－1 表示.

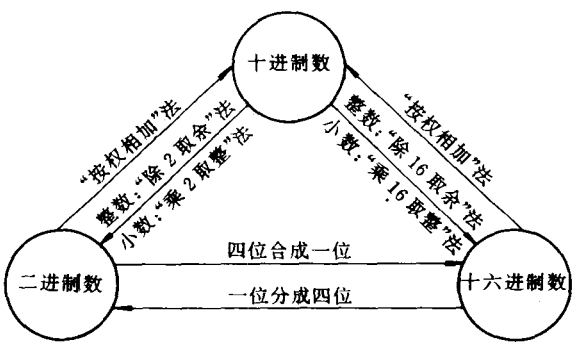

图 1－1　三种常用数制之间的转换

1.1.2　微型机中常用的编码

由于微型机不但要处理数值计算问题，而且还要处理大量非数值计算问题. 因此，除了直接给出二进制数外，不论是十进制数还是英文字母、汉字以及某些专用符号都必须编成二进制代码. 这样它们才能被计算机识别、接收、存储、传送及处理.

一、十进制数的编码

在微型机中，十进制数除了转换成二进制数外，还可用二进制数对其进行编码(通常用四位二进制数表示一位十进制数)，使它既具有二进制数的形式又具有十进制数的特点.二－十进制码又称为 BCD 码(Binary－Coded Decimal)，它有 8421 码、5421 码、2421 码以及余3 码等几种编码，其中最常用的是 8421 码. 8421 码与十进制数的对照关系见表 1－2. 每位二进 制数位都有固定的"权"，各数位的权从左到右分别为 2^3、2^2、2^1 及 2^0，即 8, 4, 2, 1. 这与二进制数位的权完全相同，但在 8421 码中绝不允许出现 1010 ～ 1111 这 6 个代码(它们被称为非法码).

由于 BCD 码低位与高位之间是" 逢十进一"，而四位二进制数(即十六进制数)是"逢十六进一". 因此用二进制加法器进行 BCD 码加法运算时，如 BCD 码的各位的和都在 0 ～ 9 之间，则其加法运算规则与二进制加法完全一样；如相加后某一位的和大于 9 或产生进位，则此

表 1－2　8421 码编码表

十进制数	8421 码	十进制数	8421 码
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

位应进行“加 6”修正. 通常在微型机中，都设置有 BCD 码调整电路，机器执行一条十进制调整指令，机器自动地对 BCD 码的和进行修正. 由于 BCD 码向高位借位是“借一当十”，而四位二进制数(相当一位十六进制数)向高位借位是“借一当十六”，因此在进行 BCD 码减法运算时，如某位有借位时，必须在该位进行“减 6”修正.

例 1－7　设有两个BCD 码，x＝01001000(48)，y＝0110 1001(69)，试求 z＝x＋y.

解:

	高位	低位
	进位　1	
	x＝0100	1000
＋)	y＝0110	1001
	中间结果 ＝1011	0001
＋)	修正　0110	0110
	z＝1　0001	0111

由于高位的和等于 1011B(11D)大于 9 且低位有进位，因此高、低位都要进行加 6 修正，修正后的结果为 117D.

例 1－8　设有两个BCD 码　x＝00101000(28)，y＝00011001(19)，试求 z＝x－y.

解：

	高位	低位
	借位　1	
	x＝0010	1000
－)	y＝0001	1001
	中间结果 ＝0000	1111
－)	修正 0000	0110
	z＝0000	1001

二、字符编码

由于微型机需要进行非数值处理(如指令、数据的输入，文字的输入及处理等)，必须对字母、文字及某些专用符号进行编码. 微型机系统的字符编码多采用美国信息交换标准代码—— ASCII 码(American Standand Code for Information Interchange). 它制订于 1963 年，后来国际标准化组织(ISO)和国际电报电话咨询委员会(CCITT)，以其为基础制订了相应的国际标准.

ASCII 码是 7 位代码，共有 128 个字符，如表 1－3 所示. 其中 96 个是图形字符，可在字符印刷或显示设备上打印出来或显示出来，包括数字符号10 个，英文字母大小写共 52 个以及其它字符 34 个；另外 32 个是控制字符，包括传输字符、格式控 制字符、设备控制字符、信息分隔符

和其它控制字符. 这类符号不打印、不显示，但其编码可进行存储，在信息交换中起控制作用.

我国于1980年制订了国家标准 GB1988 — 80，即“信息处理交换用的7位编码字符集”. 其中除了用人民币符号 ¥ 代替美元符号 $ 外，其余代码含义都与 ASCII 码相同.

请读者注意，0 ～ 9 的数字字符其相应的 ASCII 码为 30H ～ 39H；大写英文字母 A ～ F (在16进制数中表示 10 ～ 15)，其相应的 ASCII 码为 41H ～ 46H，这些在数码转换中经常要用到.

表 1－3 ASCII(美国标准信息交换码)表

列	0	1	2	3	4	5	6	7
行	位 654 → ↓ 3210	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	、	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	′	<	L	\	l	|
D	1101	CR	GS	−	=	M	]	m	}
E	1110	SO	RS	.	>	N	Ω(1)	n	~
F	1111	SI	US	/	?	O	−(2)	o	DEL

注：(1)取决于使用这种代码的机器，它的符号可以是弯曲符号，向上箭头，或(－)标记.

(2)取决于使用这种代码的机器，它的符号可以是在下面画线，向下箭头，或心形.

在表中:

NUL	空	DLE	数据链换码
SOH	标题开始	DC1	设备控制1
STX	正文结束	DC2	设备控制2
ETX	本文结束	DC3	设备控制3

EOT	传输结果	DC4	设备控制4
ENQ	询问	NAK	否定
ACK	承认	SYN	空转同步
BEL	报警符(可听见的信号)	ETB	信息组传送结束
BS	退一格	CAN	作废
HT	横向列表(穿孔卡片指令)	EM	纸尽
LF	换行	SUB	减
VT	垂直制表	ESC	换码
FF	走纸控制	FS	文字分隔符
CR	回车	GS	组分隔符
SO	移位输出	RS	记录分隔符
SI	移位输入	US	单元分隔符
SP	空间(空格)	DEL	作废

§1.2 二进制数的运算规则

由于二进制数只有0、1两个数码,因此它的加、减、乘、除等算术运算规则要比十进制数的运算规则简单得多. 这也是计算机中采用二进制数的主要原因之一. 下面讨论二进制数的运算规则.

1.2.1 加法运算规则

二进制数的加法运算规则如下:

$0+0=0$

$0+1=1+0=1$

$1+1=10$ (有进位)

$1+1+1=11$ (有进位)

例1-9 设有两个8位二进制数 $x=11110101$, $y=10110100$, 试求 $z=x+y$

解:

被加数 $x=11110101$

+) 加数 $y=10110100$

和 $z=110101001$

由此可见, 两个多位二进制数相加, 每位有三个数参加运算, 即被加数、加数以及低位的进位, 用二进制的加法规则得到的结果是本位的和以及向高位的进位.

1.2.2 减法运算规则

二进制数的减法运算规则如下:

$0-0=0$

$1-0=1$

$1-1=0$

$0-1=1$　　（有借位）

例 1－10　设两个 8 位二进制数 $x=11000100$，$y=00100101$，试求 $z=x-y$

解：

被减数 $x=11000100$

$-$）减数 $y=00100101$

差 $z=10011111$

与加法类似，两个多位二进制数相减，每位有三个数参加运算，即本位的被减数、减数及低位的借位．其结果为本位的差及向高位的借位．

1.2.3　乘法运算规则

二进制数的乘法规则如下：

$0\times0=0$

$0\times1=0$

$1\times0=0$

$1\times1=1$

不难看出，其乘法规则是十分简单的．除了 $1\times1=1$ 外，其它情况的乘积均为 0．

例 1－11　设有两个4 位二进制数 $x=1101$，$y=1011$，试求 $z=x\times y$

解：二进制的乘法与十进制的类似

被乘数	1101	$x=x_3x_2x_1x_0$
$\times$）乘数	1011	$y=y_3y_2y_1y_0$
	1101	第一次部分积
	1101	第二次部分积
	0000	第三次部分积
	1101	第四次部分积
	10001111	乘积 z

这种算法很简单，从乘数的最低位y_0开始．如 $y_0=1$，则将被乘数写下；若 $y_0=0$，则将全 0 写下．再对高位 y_1 进行运算，其运算规则与 y_0 相同．因 y_1 与 y_0 的“权”不一样，y_1 的权是 y_0 的 2 倍，故写下被乘数(或全 0)的位置要向左移 1 位．如此逐位运算，直到乘数各位都运算完为止．然后把各次的部分积累加即得乘积．因为实现这种算法所需的硬件开销大，所以计算机不采用这种算法求积，而是采用硬件开销不大的算法求积．

硬件开销不大的算法有多种，下面仅介绍一种常用的“部分积右移”的乘法算法．

设被乘数 $x=x_{n-1}x_{n-2}\cdots x_0$，乘数 $y=y_{m-1}y_{m-2}\cdots y_0$，“部分积右移”的算法如下：

(1)开始部分积为 0；

(2)检查乘数最低位的状态．如为 1，则将部分积加被乘数得新部分积；如为 0，则新部分积即为原部分积；

(3)新部分积及乘数各右移 1 位(它们的最低位从右边移出，不再参加运算)；

(4)重复(2)及(3)的步骤，直到乘数最高位运算完毕."部分积右移"的乘法算法的流程如图 1－2 所示.

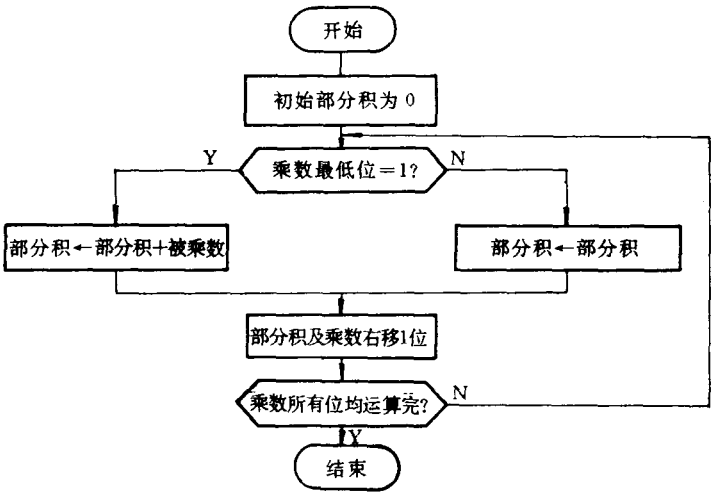

图 1－2 "部分积右移"的乘法算法流程图

部分积右移的乘法算法的硬件实现：首先将部分积寄存器清 0，将乘数送入乘数寄存器中，将被乘数送入被乘数寄存器中，若乘数最低位为 1 时，则将部分积寄存器中内容和被乘数寄存器的内容通过全加器相加，和送回部分积寄存器；若乘数最低位为 0 时，则不作加法，故部分积寄存器内容保持不变，然后将部分积和乘数一起右移一位．在硬件电路中设置一计数器，将乘数的位数(长度)送入其中，每运算一位此计数器减 1，当此计数器减为 0 时，停止操作，此时部分积寄存器的内容为乘积的高位，乘数寄存器的内容为乘积的低位.下面仍以上例中的数据来说明.

部分积寄存器	乘数寄存器	说　明
0000	1 0 1 1	初始部分积为 0，检查乘数最低位状态，
+) 1101		$y_0=1$ 部分积＋被乘数
1101		得第一次部分积
0110	1 1 0 1	部分积及乘数右移 1位，检查 y_0
+) 1101		$y_0=1$，部分积 ＋ 被乘数
10011	1	得第二次部分积
01001	1 1 1 0	部分积及乘数右移 1 位，检查 y_0
+)00000		$y_0=0$，部分积 ＋0
01001	1 1	得第三次部分积
00100	1 1 1 1	部分积及乘数右移 1 位，检查 y_0
+)01101		$y_0=1$，部分积 ＋ 被乘数
10001	1 1 1	得第四次部分积
1000	1 1 1 1	部分积及乘数右移1位，运算完毕得乘积

由此可见，这两种算法所得的结果相等。