微处理机普及教育丛书

M6800 单板微型机
原理与应用

湖南科学技术出版社

微处理机普及教育丛书

M6800单板微型机

原理与应用

冯　昭　逢　编

湖南科学技术出版社

内 容 提 要

本书从基础知识、硬件、软件和应用四个方面，深入浅出地、全面介绍用途广泛的M6800单板微型计算机的原理与应用知识。基础知识的内容包括数制及其转换，码制、基本数字逻辑电路和部件等；硬件则主要是针对单板微型机而言，包括各种存贮器、6800MPU以及几种配套芯片的结构分析和引脚功能、接口技术和输入输出等，第八章介绍了MEK6800D$_2$单板机；软件包括基本概念及汇编语言程序设计，分枝、循环、宏指令、子程序等，并对6800指令系统作了详细介绍。除了各章给出大量实例之外，全书最后一章专门给出了五个完整的实例，旨在帮助读者提高实际设计和应用的能力。

本书适合作为微机培训班的教材及自学用，特别是供厂矿企业的工程技术人员及大、中学生阅读参考。

M6800单板微型机原理与应用

冯昭途编

责任编辑：夏可军

湖南科学技术出版社出版

（长沙市展览馆路14号）

湖南省新华书店发行　湖南省新华印刷二厂印刷

1985年1月第1版第1次印刷

开本：850×1168毫米 1/32　印张：11.75　插页：2　字数：308,000

印数：1——28,000

统一书号：15204·135　　定价：2.30元

前　言

自1971年第一块微处理器INTEL4004问世以来，微型计算机的发展速度极为迅速。特别是由于其价格/性能比日益大幅度下降，使它的应用很快在各个领域中得以推广。它的性能和集成度，几乎每两年增加一倍，而价格却更便宜。目前，造价只有十来美元的单片微型机，性能却超过了六十年代初耗资数十万美元的晶体管计算机。

国内目前已经大量引进的微型机系列主要有三种型号：INTEL8080、M6800、Z—80。本书介绍M6800单板机级水平的原理和应用。各不同型号的微处理器各有优点，而M6800在国内推广却有以下四个方面的优势：

(1) M6800微处理器及其系列支持芯片，是世界上公认的设计得较为成功的微处理器系列产品之一。其原理和操作技术比较容易学习和掌握。

(2) 国内有关的研究所和工厂已经掌握了M6800系列芯片的制造工艺和模块板的开发。其中不少已批量投入生产。M6800系列微型机国内仿制的型号为DJS—060系列。这是M6800相对其他型号在国内推广应用的一个主要优势。

(3) 电子工业部已确定M68000为国内十六位机的选型。M68000和M6800之间软件和硬件的兼容性给八位机M6800的选用带来新的吸引力。

(4) M6800系列具有完整的硬件支持芯片、模块板、开发系统及丰富的支持软件。这方面的主要问题是：国内M6800硬、软

件资料的翻译、整理和推广较之其他型号要差，这是M6800研制和应用推广部门需要认真考虑的。

本书主要内容为M6800微处理器原理、指令系统、各种半导体存贮器原理和芯片、各类接口芯片和其他支持芯片、汇编语言——机器语言级的程序设计方法。并以MEK6800D_2单板机系统为例介绍M6800微型机硬件系统的构成和软件系统的功能。还举了一些在D_2机系统或其他M6800微机系统中很快可以实现的应用实例。

学习微型机必须强调实践，学习本书的读者最好能有一台MEK6800D_2单板机，或者有以M6800微处理器为核心的微型机系统。边学习、边实践，方能取得比较满意的结果。

郭德纨副教授、胡嘉勋同志审阅了全书，邵晶同志整理了全部书稿并设计描绘了部分插图，电子工业部47所潘方荣工程师和沈阳辽宁精密仪器厂杨博强工程师为本书提供了不少资料和建议，在此一并表示感谢。

作　者

1984.1

目　录

·第一章·

数的表示方法和信息编码

任何计算机中完成各种运算所需的信息无非是以下两类：数据类信息和命令类信息。例如：

$45+72=?$

45和72就是需要进行运算的数据信息，而“+”则是执行何种运算的命令信息。在数字计算机中，数据信息和命令信息都是用数字来表示的。通常人们所熟悉的数字是十进计数制，其特点是逢十进一。它的出现显然是由于人类有十个手指头的缘故。实际上，数字可以不是十进制的，在日常生活中就存在许多各种不是十进制的数字。例如：

过去的老秤：十六两为一斤，是十六进位计数制，逢十六进一；

钟表的计数：六十秒为一分，六十分为一小时，是六十进位制，逢六十进一；

一打为十二个：是十二进位制，逢十二进一；

二个为一双：是二进位制，逢二进一。

这些不同的进位计数制，称为**数制**。每种数制的基本特征就是**底数**(或称为**基数**)。十进制数制的底数为十，八进制数制的底数为八，二进制数制的底数为二。在电子数字计算机中，常用十六进制、十进制、八进制和二进制数。下面我们就来讨论一下计算机中常用数制及其转换的方法。

§1.1 数制及其转换方法

1.11 位权的概念

以十进制数为例：

5 5 5 =？

②①⓪

这个三位的十进制数，虽然每一位的数字都是5，但代表的数值却是大不相同的。

第⓪位数5是个位的5，其数值就是五；

第①位数5是十位，其数值为五十；

第②位数5是百位，其数值为五百。

即一个数中每位数字的**位权**是不同的。同样是一个数字5，在不同的**位**上，其数值的"**权利**"是大不相同的。

每位数的**位权**由数制底数的n(位数）次幂来确定。对于十进制数来说，其底数为十、最低位，即第 0 位的位权为$10^0=1$；第1位的位权为$10^1=10$；第2位的位权为$10^2=100$；……；第 n 位的位权为10^n。

推而广之，对于B进制数来说，底数为B。其最低位，即第 0 位的位权为$B^0=1$；第1位的位权为$B^1=B$；第 2 位的位权为B^2；……；第n位的位权为B^n。

任意进制的N 位数的总值，等于其每位数字分别乘以它们所在位的位权的积之和。

例如：B进制的N位数

$$a_{N-1}a_{N-2}a_{N-3}\cdots a_3a_2a_1a_0$$

这个数的总值Σ为

$$\Sigma=a_{N-1}\times B^{N-1}+a_{N-2}\times B^{N-2}+\cdots$$
$$+a_2\times B^2+a_1\times B^1+a_0\times B^0$$

为了区分所用的数制，一般在数的右下角标明其底数。例如：

$(4096)_{10}$表示为十进制数；$(7341)_8$表示为八进制数；$(1011)_2$表示为二进制数。

由于人们对十进制数比较熟悉，而对一个其他进制数的大小，则很难判别。计算机中又常用二进制、八进制、十六进制。为了将人们熟悉的十进制数输入计算机，必须转换成二进制数，而经过计算机运算和处理后的二进制数又要转换成十进制数才能被人们识别。因此，了解数制之间的转换是必要的。

1.12 二进制数及其转换

二进制数是最简单的数制，只包含两个数码，即0和1。由于二进制的一系列优点，现代电子计算机均采用二进制来表示数字和各种信息。其原因是：

(1) 具有两种物理状态的器件非常之多，很容易用来表示0，1两个状态。例如：电平的高、低；磁化方向的不同；电子元件的导通和截止等等。而且由于只用二个状态表示数字，工作稳定可靠，抗干扰能力强。

(2) 二进制数的算术运算较为简单。

加法，只有三种情况：

$0+0=0$；$0+1=1$；$1+1=10$。

乘法，也只有三种情况：

$0\times0=0$；$1\times0=0$；$1\times1=1$。

这样的运算规则比十进制要简单得多，使计算机中实现算术运算的逻辑电路大为简化。

(3) 更重要的一点是：由于逻辑运算中也只有二种基本状态："是"与"非"。用"1"表示"肯定"，用"0"表示"否定"。这样，算术运算和逻辑运算所用的元件和线路便统一起来了。使得计算机中许多问题的处理也大为简化。

当然，还有一些次要的原因。可以这样说，没有二进制，就没有现代的电子计算机。

1、二进制数转换成十进制数

一个如下示的八位的二进制整数，其总数值为多少？

1 0 1 1 0 1 0 1

按前述位权的概念，我们可以列出二进制整数各位的位权。如表1—1示。

表1.1　　二进制整数各位位权表

位　　数	7	6	5	4	3	2	1	0
位　　权	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
十进制值	128	64	32	16	8	4	2	1

由表1.1位权表，即可计算出上述八位二进制数的总值(十进制表示)。

$$\sum = 1\times128 + 1\times32 + 1\times16 + 1\times4 + 1\times1 = 181$$

一个0.101010的二进制纯小数，其对应的十进制数为多少？

同样，由位权的概念，也可得到表1.2所示的二进制纯小数各位的位权表。

表1.2　　二进制纯小数各位位权表

位　　数	−1	−2	−3	−4	−5	−6	−7
位　　权	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
十进制值	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.0078125

由表1—2，即可求出上述二进制纯小数的十进制数值。

$$\sum = 0.5 + 0.125 + 0.03125 + 0.015625 = 0.671875$$

对于任意的二进制数(包括整数和小数部分)，要将其转换成十进制数，必须分别进行整数部分转换和纯小数部分转换，其结果由两部分组合而成。

2. 十进制数转换成二进制数

十进制整数要转换成二进制数可采用底数连除法。举例说明

如下。

例：将十进制整数19转换成二进制数。

2 | 1 9
2 | 9　　余数为 1 ←最低位 LSB
2 | 4　　余数为 1
2 | 2　　余数为 0
2 | 1　　余数为 0
　 0　　余数为 1 ←最高位MSB

即 $(19)_{10}=(10011)_2$

第一次除2后得的余数为最低位，记为 LSB；最后一次除2后得到的余数为最高位，记为MSB。

底数连除法适用于十进制整数转换成任意进制的整数，只要用该进制的底数连除即行。

十进制纯小数要转换成二进制小数，则可采用**底数连乘法**。这种底数连乘法的原理是："任何两个相等的不同进制的数，它们的整数部分等于整数部分，小数部分等于小数部分"。

以前面由二进制纯小数转换成的十进制纯小数为例，进行底数连乘法。

$0.671875\times2=1.34375$　　整数位溢出 1 ←MSB
$0.34375\times2=0.6875$　　整数位溢出 0
$0.6875\times2=1.375$　　整数位溢出 1
$0.375\times2=0.75$　　整数位溢出 0
$0.75\times2=1.5$　　整数位溢出 1
$0.5\times2=1.0$　　整数位溢出 1 ←LSB

将每次溢出的结果按MSB→LSB顺序记录下来即得结果。

$(0.671875)_{10}=(0.101011)_2$

其结果和转换前的二进制纯小数完全一致。第一次相乘的溢出结果为最高位（MSB），而最后一次相乘的溢出结果为最低位（LSB）。此例中，最后一次相乘后，小数部分为零。但在一般情况下，这种连乘可能是无止境的。可根据精度的要求，选择适当

的位数。

若十进制数包括整数和小数，则必须用底数连除法将整数部分转换成二进制数的整数，再用底数连乘法将小数部分转换成二进制数的小数，两者组合得出完整的结果。

3. 八进制和十六进制

一个十六位的二进制数如下所示：

1011010101111001

这样一长串0和1的序列是很难阅读和书写的，更不易记忆。因此，在计算机中常采用八进制或十六进制来书写二进制数。这是因为八进制和十六进制与二进制之间有些特殊关系。

由于$2^3=8$，一位八进制数正好等于三位二进制数；$2^4=16$，一位十六进制数正好等于四位二进制数。

八进制数有八个数码0～7，进位规则为“逢八进一”。

$(10)_8=(8)_{10}$

要把上述十六位二进制数转换成八进制数，只要把它们从最低位开始三位一分，即可直接写出相应的八进制数，如下所示。

十六位二进制数：1，011，010，101，111，001

相应的八进制数：1　3　2　5　7　1

即，$(1011010101111001)_2=(132571)_8$

用六位八进制数来书写其相应的十六位二进制数也很容易。

用位权的概念亦可很方便地将八进制数转换成十进制数。

$$(132576)_8=1\times8^5+3\times8^4+2\times8^3+5\times8^2+7\times8^2+6\times8^0=(46462)_{10}$$

同样，用底数连除法和底数连乘法也可很方便地将十进制整数或纯小数转换成相应的八进制整数和纯小数。此时的底数是8。

在应用微型计算机时，更常用十六进制数编写机器语言程序。因此，我们对十六进制数作较为详细的介绍。

十六进制数是用十六个数码分别来表示其十六种状态。现有的十进制只有十个数码0～9，还需指定六个数码来表示从十到十五这六个状态。

通常指定大写英文字母A、B、C、D、E、F分别表示十到十五这六个数码。表1—3列出了从0～34的十进制数与对应的二进制、八进制、十六进制数的对照表。

表1—3　　十、二、八、十六进制数的对照表

十进制	二进制	八进制	十六进制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17
24	11000	30	18
25	11001	31	19
26	11010	32	1A
27	11011	33	1B
28	11100	34	1C
29	11101	35	1D
30	11110	36	1E
31	11111	37	1F
32	100000	40	20
33	100001	41	21
34	100010	42	22

用字母表示数字，对于初学者来说可能很不习惯。但只要经过手工编程训练，很快就会熟悉的。

仍以上述十六位二进制数为例，将其转换成十六进制数。只需由最低位开始，四位一分，由对照表立即可写出相应的四位十六进制数，如下所示。

十六位二进制数：1011，0101，0111，1001

四位十六进制数：B　5　7　9

即　$(1011010101111001)_2=(B579)_{16}$

可见，用十六进制来书写二进制数，比八进制更便于阅读和书写。

在八位微处理器中，一次处理八位二进制数。称八位二进制数为一个字节。这样，可以用二位十六进制数来书写一个字节。需要很熟练地背下四位二进制数和一位十六进制数之间的对应关系。

十六进制数转换成二进制数，只要每位十六进制数直接写成与其相对应的四位二进制数即可。例如

四位十六进制数：A　3　F　4

↓　↓　↓　↓

1010 0011 1111 0100

十六进制数转换成十进制数，只要记住位权的概念，是一点也不困难的。

例如：将$(B5EC)_{16}$转换成十进制数。

$$(B5EC)_{16}=11\times16^3+5\times16^2+14\times16^1+12\times16^0$$
$$=(46572)_{10}$$

而十进制数转换成十六进制数，亦是用底数连除法转换整数部分，用底数连乘法转换小数部分，组合而成。例如：

将十进制数 37318.7654 转换成十六进制数（取小数点后四位）。

先用底数连除法转换其中整数部分：

16 | 3 7 3 1 8

16 | 2 3 3 2　　余数为　6 ←LSB

16 | 1 4 5　　余数为 $(12)_{10}=(C)_{16}$

16 | 9　　余数为　1

0　　余数为　9 ←MSB

即　$(37318)_{10}=(91C6)_{16}$

再用底数连乘法转换其中纯小数部分：

$0.7654\times16=12.2464$　溢出$(12)_{10}=(C)_{16}$←MSB

$0.2464\times16=3.9424$　溢出　3

$0.9424\times16=15.0784$　溢出$(15)_{10}=(F)_{16}$

$0.0784\times16=1.2544$　溢出　1　←LSB

即　$(0.7654)_{10}\doteq(0.C3F1)_{16}$

组合即得　$(37318.7654)_{10}\doteq(91C6.C3F1)_{16}$

§1.2　符号表示法及码制

1.21　数的符号表示法——原码表示法

十进制数正负数的表示方法就是在数的前面加“+”号或“-”号，“+”号一般可省略不记。二进制数也可用正负号来表示正负。如：

+1011；-1101。

但在计算机中用正负号表示正负很不便于运算。通常用二进制数的最高位的“0”或“1”来表示正负。一般都规定“0”表示正数、“1”表示负数。对于八位二进制数来说，最高位是符号位，因此，只有低七位才是数据。

如　01011011　　表示$+(91)_{10}$。

11110101　　表示$-(117)_{10}$。

这样一来，数据和符号全都数码化了，符号位和数据位都是数码。最高位的数码表示后面数据的正负符号，后面（右边）才是数据的绝对值。这种数及符号的表示方法称为**原码表示法**。

对于八位二进制数来说，其可表示的范围从$+(127)_{10}$～$-(127)_{10}$，其中0有两种形式：

+0=00000000。　-0=10000000。

原码表示法在计算机进行加减运算时很麻烦。两数相加，必须先判断两个数的符号是否一致，相同则进行加法，否则就要做减法。做减法时，还必须比较两个数的绝对值的大小，绝对值大