

# 微处理机应用基础

天津科学技术出版社

# 微处理器应用基础

梁震源 黄书基 编

天津科学技术出版社

## 内 容 提 要

本书是一本学习微处理机基础知识的参考书。书中收编并介绍了Motorola公司6800系列的主要产品。全书共分六章，着重从原理、编程和接口三个方面系统地阐述微处理机应用的基本知识。内容深入浅出，联系实际，可供从事微处理机应用的工程技术人员、科学研究人员自学时参考，亦可作为大学有关专业教材使用。

### 微处理机应用基础

梁展翼、戴书基 编

责任编辑：宋淑萍

天津科学技术出版社出版

天津市赤峰道124号

天津市晒图厂印刷

新华书店天津发行所发行

开本 787×1092 毫米 1/32 印张 10.625 字数 241,000

一九八四年十月第一版

一九八四年十月第一次印刷

印数：1—38,000

书号：15212·139 定价：1.80 元

## 编者的话

大规模集成电路技术的发展，微处理机的出现，促进了计算机在许多领域和部门的普及。由于微处理机体积小，使用灵活，价格低廉，除了被用来组装成廉价的微型计算机之外，还被广泛用于过程控制和智能化仪器仪表中。因此，有关微处理机的知识不仅对从事计算机专业者，而且对广大科技人员也十分必要。

本书的初稿是为南开大学计算机应用专业、电子学专业和物理学专业开设的《微处理机应用基础》课编写的教材，并且为校外举办的微处理机应用训练班多次采用。

微处理机产品繁多，发展迅速，我们认为只有具备微处理机应用的基础知识才能适应迅速发展的形势。因此本书没有罗列许多产品，而是从国内常用的产品之一——6800微处理机入手，从应用的角度，在有限的篇幅当中从原理、编程和接口三个方面介绍微处理机应用的基础知识。尽管涉及的仅仅是6800，但仍可作为进一步学习和应用各种微处理机的入门。

本书是在李乐天副教授关心指导下编写的。杨兴武、吴大征、郁鑫昌、王计会、余金森同志参加了初稿的编写工作，王计会同志还对本书进行了认真的校对，在此特致以谢意。由于编者水平所限，谬误之处请读者指正。

编者

1983年8月

# 目 录

## 第一章 微处理机的基本原理

|                       |      |
|-----------------------|------|
| §1.1 微处理机中的数和代码 ..... | (1)  |
| 一、二进制数 .....          | (1)  |
| 二、十六进制数 .....         | (2)  |
| 三、二十进制数 .....         | (4)  |
| 四、ASCII码 .....        | (4)  |
| §1.2 二进制数的机器运算 .....  | (7)  |
| 一、不带符号数的运算 .....      | (7)  |
| 二、带符号数的运算 .....       | (11) |
| 三、逻辑运算 .....          | (15) |
| §1.3 微型计算机 .....      | (19) |
| 一、微处理机与微型计算机 .....    | (19) |
| 二、计算机字与字长 .....       | (21) |
| 三、指令与程序 .....         | (22) |
| §1.4 基本的微处理机 .....    | (23) |
| 一、微处理机的结构 .....       | (25) |
| 二、存储器 .....           | (27) |
| 三、指令的取出-执行序列 .....    | (30) |
| 四、程序实例 .....          | (31) |
| §1.5 程序的执行 .....      | (34) |
| 一、LDA指令的执行 .....      | (36) |
| 二、ADD指令的执行 .....      | (43) |
| 三、STA指令的执行 .....      | (45) |
| 四、WAI指令的执行 .....      | (50) |
| §1.6 寻址方式 .....       | (51) |

|                 |      |
|-----------------|------|
| 一、隐含(固有)寻址..... | (51) |
| 二、立即寻址.....     | (52) |
| 三、直接寻址.....     | (54) |

## 第二章 6800微处理机

|                         |      |
|-------------------------|------|
| §2.1 6800的结构与编程模型 ..... | (56) |
| 一、6800的结构.....          | (56) |
| 二、6800的编程模型.....        | (56) |
| §2.2 6800新的寻址方式.....    | (61) |
| 一、扩展寻址.....             | (61) |
| 二、变址寻址.....             | (62) |
| 三、相对寻址.....             | (64) |
| §2.3 6800指令系统.....      | (70) |
| 一、算术逻辑指令.....           | (71) |
| 二、数据处理指令.....           | (78) |
| 三、数据测试指令.....           | (83) |
| 四、转移和跳转指令.....          | (84) |
| 五、变址寄存器和堆栈指示器指令.....    | (91) |
| 六、条件码寄存器指令.....         | (93) |
| 七、6800指令表.....          | (94) |

## 第三章 堆栈与中断

|                   |       |
|-------------------|-------|
| §3.1 堆栈 .....     | (103) |
| 一、串联堆栈 .....      | (103) |
| 二、存储器堆栈 .....     | (106) |
| §3.2 子程序.....     | (111) |
| 一、子程序与主程序 .....   | (111) |
| 二、子程序的多次调用 .....  | (113) |
| 三、子程序嵌套 .....     | (115) |
| §3.3 输入/输出操作..... | (117) |
| 一、输出操作.....       | (119) |
| 二、输入操作.....       | (120) |

|                          |              |
|--------------------------|--------------|
| 三、输入/输出程序 .....          | (121)        |
| <b>§3.4 中断 .....</b>     | <b>(123)</b> |
| 一、复位 Reset .....         | (124)        |
| 二、不可屏蔽中断 NMI .....       | (127)        |
| 三、中断请求 IRQ .....         | (129)        |
| 四、软件中断 SWI .....         | (131)        |
| 五、几条与中断有关的指令 .....       | (132)        |
| <b>第四章 程序设计入门</b>        |              |
| <b>§4.1 基本运算程序 .....</b> | <b>(136)</b> |
| 一、多字节数的加减法运算 .....       | (136)        |
| 二、BCD 数的加减法运算 .....      | (140)        |
| 三、BCD 数与二进制数的互换 .....    | (145)        |
| 四、乘除法运算 .....            | (155)        |
| <b>§4.2 数据处理程序 .....</b> | <b>(163)</b> |
| 一、一组数求和 .....            | (163)        |
| 二、两组数求积 .....            | (166)        |
| 三、复制一个表 .....            | (168)        |
| 四、查表 .....               | (171)        |
| 五、求最大值和最小值 .....         | (172)        |
| 六、排序 .....               | (174)        |
| <b>第五章 微处理机接口技术</b>      |              |
| <b>§5.1 接口电路基础 .....</b> | <b>(178)</b> |
| 一、逻辑门电路和译码电路 .....       | (178)        |
| 二、电路的相互联接 .....          | (183)        |
| 三、MPU 操作的定时 .....        | (190)        |
| 四、6800 微处理机的接口引线 .....   | (193)        |
| <b>§5.2 存储器 .....</b>    | <b>(195)</b> |
| 一、随机读写存储器 (RAM) .....    | (195)        |
| 二、只读存储器 (ROM) .....      | (198)        |
| 三、集成电路随机存储器的结构 .....     | (200)        |
| 四、存储器的寻址 .....           | (203)        |

|                                                                    |       |
|--------------------------------------------------------------------|-------|
| §5.3 发光二极管 (LED) 显示器 .....                                         | (204) |
| 一、发光二极管显示器 .....                                                   | (205) |
| 二、八段显示器的显示控制方式 .....                                               | (207) |
| 三、多显示器的控制电路 .....                                                  | (215) |
| 四、点阵式显示器的控制电路 .....                                                | (215) |
| §5.4 键盘 .....                                                      | (217) |
| 一、有效键入信号 .....                                                     | (217) |
| 二、键入信号的识别 .....                                                    | (219) |
| 三、通过执行程序读键盘 .....                                                  | (222) |
| 四、键盘编码器 .....                                                      | (227) |
| §5.5 可编程并行接口器件 .....                                               | (229) |
| 一、6820 PIA的结构 .....                                                | (230) |
| 二、MPU读写PIA的操作 .....                                                | (236) |
| 三、通过PIA与外设通信 .....                                                 | (241) |
| §5.6 6800MPU与外设的异步串行接口 .....                                       | (245) |
| 一、串行传送方式 .....                                                     | (245) |
| 二、6850异步串行接口适配器 (ACIA) .....                                       | (246) |
| 三、通过ACIA对电传打字机进行输入输出 .....                                         | (253) |
| §5.7 同步串行数据适配器 (SSDA) .....                                        |       |
| 一、6852 SSDA的接口线 .....                                              | (255) |
| 二、6852 SSDA的寄存器结构与寻址 .....                                         | (258) |
| 三、状态寄存器SR .....                                                    | (260) |
| 四、控制寄存器 (CR <sub>1</sub> 、CR <sub>2</sub> 、CR <sub>3</sub> ) ..... | (262) |
| 五、6852 SSDA的中断控制 .....                                             | (265) |
| 六、6852 SSDA中的总清 .....                                              | (265) |
| 七、6852 SSDA的初始化 .....                                              | (265) |
| §5.8 阴极射线管 (CRT) 字符显示器 .....                                       |       |
| 一、CRT字符显示器的工作原理 .....                                              | (267) |
| 二、CRT控制器6845 (CRTC) .....                                          | (274) |

## 第六章 6809及其它CPU简介



|                        |       |
|------------------------|-------|
| §6•1 6809微处理机 .....    | (283) |
| 一、概述 .....             | (283) |
| 二、6809微处理机的编程模型 .....  | (284) |
| 三、6809微处理机的引线 .....    | (287) |
| 四、6809微处理机的寻址方式 .....  | (297) |
| 五、6809微处理机的指令系统 .....  | (308) |
| 六、6809微处理机指令表 .....    | (311) |
| §6•2 6802微处理机 .....    | (315) |
| 一、6802微处理机的引线 .....    | (315) |
| 二、片内RAM的使用 .....       | (318) |
| 三、外接晶体的联接 .....        | (318) |
| §6•3 6801单片微型计算机 ..... | (319) |
| 一、6801的新指令 .....       | (320) |
| 二、6801的内部结构 .....      | (321) |
| 三、6801的工作方式 .....      | (323) |
| 四、6801的可编程定时器 .....    | (327) |
| 五、6801的串行通信接口 .....    | (329) |
| 六、6800系列的单片微计算机 .....  | (329) |

# 第一章 微处理机的基本原理

## §1.1 微处理机中的数和代码

### 一、二进制数

微处理机中数据和其它信息通常以二进制的形式表示和进行运算处理。

我们对于十进制数非常熟悉。例如，7801.693表示

$$7 \times 10^3 + 8 \times 10^2 + 0 \times 10^1 + 1 \times 10^0 + 6 \times 10^{-1} + 9 \times 10^{-2} + 3 \times 10^{-3}$$

一个任意的十进制数S其形式为 $k_n k_{n-1} k_{n-2} \dots k_1 k_0 . k_{-1} k_{-2} \dots k_{-m}$ ，表示为

$$k_n \times 10^n + k_{n-1} \times 10^{n-1} + \dots + k_1 \times 10^1 + k_0 \times 10^0 + k_{-1} \times 10^{-1} + \dots + k_{-m} \times 10^{-m}$$

或者

$$\sum_{i=-m}^n k_i 10^i$$

其中 $n, m$ 为正整数， $k_i$ 为 $0, 1, 2, \dots, 9$ 这十个数字中的一个，由S决定。

在进位记数制中用到的数字的个数叫基数。十进记数制中数字 $k_i$ 一共有十个，所以基数为10，每位计满十向高位进一，即“逢十进一”。此外，每个数字 $k_i$ 所在的位都对应一个固定值 $10^i$ 叫做该位的“权”，它是基数的某个方次。例如百位的权为 $10^2$ ，个位的权为 $10^0$ ，小数点后第一位的权为 $10^{-1}$ 等等。

上面的式子称为十进制数按权展开的多项式，其中的每一项为某个数字与该位权的乘积，数S就是该多项式的和。

对于二进制数来说，仅采用0和1两个数字，因此基数为2，每位计满二向高位进一，即“逢二进一”。所以二进制数S按权展开的多项式为

$$\sum_{i=-n}^{-m} k_i 2^i$$

其中 $k_i$ 为0或者1，由S决定。 $2^i$ 为 $k_i$ 对应的权。例如

$$\begin{aligned} 101101.11_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \\ &\quad \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 32 + 8 + 4 + 1 + 0.5 + 0.25 \\ &= 45.75_{10} \end{aligned}$$

为了避免混淆，当两种进位制数同时出现时，应有下标表示该数的基数。

由于数字电路具有两种稳定的状态，用0和1表示很方便，所以微处理机就采用二进制。

## 二、十六进制数

除了二进制外，微处理机还采用十六进制 (Hexadecimal)。十六进制的基数为16，每位计满十六向高位进一，即“逢十六进一”。十六进制用到的数字 $k_i$ 是16个，开头的10个和十进制相同，后面的6个借用英文字母A、B、C、D、E、F，见表1-1。

十六进制数S按权展开的多项式为

$$\sum_{i=-n}^{-m} k_i 16^i$$

表1-1 十六进制数

|           |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-----------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 十 进 制 数   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 十 六 进 制 数 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A  | B  | C  | D  | E  | F  |

$16^i$  为十六进制数某位的权。例如

$$\begin{aligned}
 2AD.8_{16} &= 2 \times 16^2 + A \times 16^1 + D \times 16^0 + 8 \times 16^{-1} \\
 &= 2 \times 256 + 10 \times 16 + 13 \times 1 + 8 \times 0.0625 \\
 &= 685.5_{10}
 \end{aligned}$$

可知，十六进制数  $2AD.8_{16}$  的等值十进制数是  $685.5_{10}$ 。

从表1-2看到1位十六进制数刚好对应4位二进制数，因此两者转换起来非常方便。

表1-2 二进制数与十六进制数对照表

|           |      |      |      |      |      |      |      |      |
|-----------|------|------|------|------|------|------|------|------|
| 二 进 制 数   | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| 十 六 进 制 数 | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
| 二 进 制 数   | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 十 六 进 制 数 | 8    | 9    | A    | B    | C    | D    | E    | F    |

例如，十六进制数  $2AD.8_{16}$  把每位换成与它等值的4位二进制数，则成为如下形式：

$$\begin{array}{ccccccc}
 2 & A & D & . & 8 & & \\
 0010 & 1010 & 1101 & . & 1000 & & 
 \end{array}$$

把其中不起作用的0去掉则变成  $1010101101.1_2$ ，它就是  $2AD.8_{16}$  的等值二进制数。

反过来，一个二进制数从小数点起，整数向左，小数向右，

每 4 位作为一组。整数最左边的一组如不足 4 位，用在高位的左边补 0 的办法补齐，小数最右边的一组如不足 4 位，用在低位的右边补 0 的办法补齐。然后每组用与之对应的十六进制数替换。于是得到该二进制数的等值十六进制数。例如，二进制数  $100101110.01_2$  可以表示为

$$000100101110.0100$$

$$1 \quad 2 \quad E \quad 4$$

$12E.4_{16}$  就是  $100101110.01_2$  的等值十六进制数。

十六进制数书写起来简便，和二进制数转换容易。所以在微处理机输入、输出数据时常采用十六进制。

### 三、二十进制数

由于人们习惯于十进制，所以除采用二进制和十六进制外，微处理机有时也采用一种二进制编码的十进制数或称为 BCD(Binary Coded Decimal)数。它是十进制数，不过 0, 1, 2……9 这十个数字用 4 位二进制数的编码来表示，目的是为了适应数字电路表示 0 和 1 方便的特点。从 0000 到 1111 本来有 16 种组合，我们只取 0000 到 1001 表示 0 到 9，从 1010 到 1111 则舍去不用。于是  $384.06_{10}$  就表示成

$$0011 \ 1000 \ 0100.0000 \ 0110_{BCD}$$

为了和纯二进制数区别，在书写时对应一位十进制数的四位二进制数编码写得靠拢一些。一旦记熟 0 到 9 的十个二进制编码，BCD 数使用起来和十进制数一样方便。

表 1-3 是我们已介绍过的几种进位制数的对照表。

### 四、ASCII 码

除了前面介绍的几种进位制数外，微处理机还经常用到一种叫做 ASCII 码的代码，它是美国信息交换标准代码 (American Standard Code For Information Interchange)

**表1-3 几种进位制数对照表**

| 十进制 | 二进制   | 十六进制 | 二 - 十 进 制 |
|-----|-------|------|-----------|
| 0   | 0000  | 0    | 0000      |
| 1   | 0001  | 1    | 0001      |
| 2   | 0010  | 2    | 0010      |
| 3   | 0011  | 3    | 0011      |
| 4   | 0100  | 4    | 0100      |
| 5   | 0101  | 5    | 0101      |
| 6   | 0110  | 6    | 0110      |
| 7   | 0111  | 7    | 0111      |
| 8   | 1000  | 8    | 1000      |
| 9   | 1001  | 9    | 1001      |
| 10  | 1010  | A    | 0001 0000 |
| 11  | 1011  | B    | 0001 0001 |
| 12  | 1100  | C    | 0001 0010 |
| 13  | 1101  | D    | 0001 0011 |
| 14  | 1110  | E    | 0001 0100 |
| 15  | 1111  | F    | 0001 0101 |
| 16  | 10000 | 10   | 0001 0110 |
| 17  | 10001 | 11   | 0001 0111 |
| 18  | 10010 | 12   | 0001 1000 |
| 19  | 10011 | 13   | 0001 1001 |
| 20  | 10100 | 14   | 0010 0000 |
| 21  | 10101 | 15   | 0010 0001 |
| 22  | 10110 | 16   | 0010 0010 |

英文字头的缩写。表1-4列出了7位ASCII码,代表128种不同的符号。这些符号大部分和打字机键盘一致。包括以下几种:

- (1) 26个英文大写字母;
- (2) 26个英文小写字母;
- (3) 10个十进制数字;

表1-4 7位ASCII码

| 列  |                     | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
|----|---------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| 行  | 位<br>4321 765→<br>↓ | 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   |
| 10 | 1010                | LF  | SUB | .   | :   | J   | Z   | j   | z   |
| 11 | 1011                | VT  | ESC | +   | ;   | K   | [   | k   | {   |
| 12 | 1100                | FF  | FS  | .   | <   | L   | \   | l   |     |
| 13 | 1101                | CR  | GS  | -   | =   | M   | ]   | m   | }   |
| 14 | 1110                | SO  | RS  | .   | >   | N   | ^   | n   | ~   |
| 15 | 1111                | SI  | US  | /   | ?   | O   | _   | o   | DEL |

(4) 标点符号及运算符号;

(5) 打字格式及控制符号, 如“空格”、“回车”等;

(6) 其它控制符号如“传输结束”, “铃声”等。

要确定一个数字、字母或其它符号的ASCII码, 可先在表中找到这个符号, 然后将该符号所在的行和列对应的三位和四

位数连起来就得到相应的 ASCII 码。例如，大写字母 V 的 ASCII 码是 1010110，数字 7 的 ASCII 码是 0110111 等等。

ASCII 码在传送时常采用 8 位，其中最高位用做校验位，以便检查代码在传送过程中可能出现的差错。有奇、偶两种校验方式，奇校验规定每个代码中包括校验位在内 8 位数中 1 的个数应当为奇数。按此规定来确定校验位是 0 还是 1。例如大写字母 S，它的 7 位 ASCII 码是 1010011。由于 1 的个数为偶数，为凑成规定的奇数，所以校验位应当取 1。又例如数字 7 的 7 位 ASCII 码为 0110111，其中 1 的个数为 5，已经是奇数了，因此校验位应取 0。

偶校验规定每个代码包括校验位在内的 8 位数中 1 的个数为偶数。校验位取 0 还是取 1，则按这个规定来确定。对于带校验位的 8 位 ASCII 码，可以通过检查其中 1 的个数的奇偶性是否符合规定来发现传送过程中出现的差错。

## §1.2 二进制数的机器运算

微处理机的基本功能是对二进制数进行算术和逻辑运算。多数微处理机算术和逻辑运算是按 8 位二进制数进行的。所以，下面的讨论皆以 8 位二进制数为例。

### 一、不带符号数的运算

#### 1. 加法运算

二进制数的加法运算规则很简单：

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \quad \text{产生进位}$$

$$1 + 1 + 1 = 1 \quad \text{产生进位}$$

例 1. 00110000 加 10110111



按照加法运算规则逐位相加可得到和数。

$$\begin{array}{r}
 \phantom{0}11 \\
 00110000 \\
 +10110111 \\
 \hline
 11100111
 \end{array}
 \begin{array}{l}
 \text{进位} \\
 \text{被加数} \\
 \text{加数} \\
 \text{和}
 \end{array}$$

相当于  $48_{10} + 183_{10} = 231_{10}$

例2. 10110111加01110001

$$\begin{array}{r}
 1111 \quad 111 \\
 10110111 \\
 +01110001 \\
 \hline
 100101000
 \end{array}
 \begin{array}{l}
 \text{进位} \\
 \text{被加数} \\
 \text{加数} \\
 \text{和}
 \end{array}$$

进位 ↑

相当于  $183_{10} + 113_{10} = 296_{10}$

两个8位的二进制数相加和数可能为8位，也可能超过8位，例1和例2代表了这两种情况。由于微处理机只有8位，当和数超过8位时，第9位将会丢失。为此，微处理机另外设置了一个进位位用来保存加法运算时从最高位（第8位）产生的进位。对例1，机器将给出这样的结果 0 11100111，对例2，将给出这样的结果 1 00101000。用方框框住的那一位表示进位位。

## 2. 减法运算

二进制数的减法运算规则也很简单：

$$\begin{array}{l}
 0 - 0 = 0 \\
 1 - 1 = 0 \\
 1 - 0 = 1 \\
 0 - 1 = 1 \quad \text{有借位 1}
 \end{array}$$

例3. 10110111减10001011

按照减法运算规则逐位进行运算可得到差值

$$\begin{array}{r}
 \phantom{1}1 \\
 10110111 \\
 -10001011 \\
 \hline
 00101100
 \end{array}
 \begin{array}{l}
 \text{借位} \\
 \text{被减数} \\
 \text{减数} \\
 \text{差}
 \end{array}$$