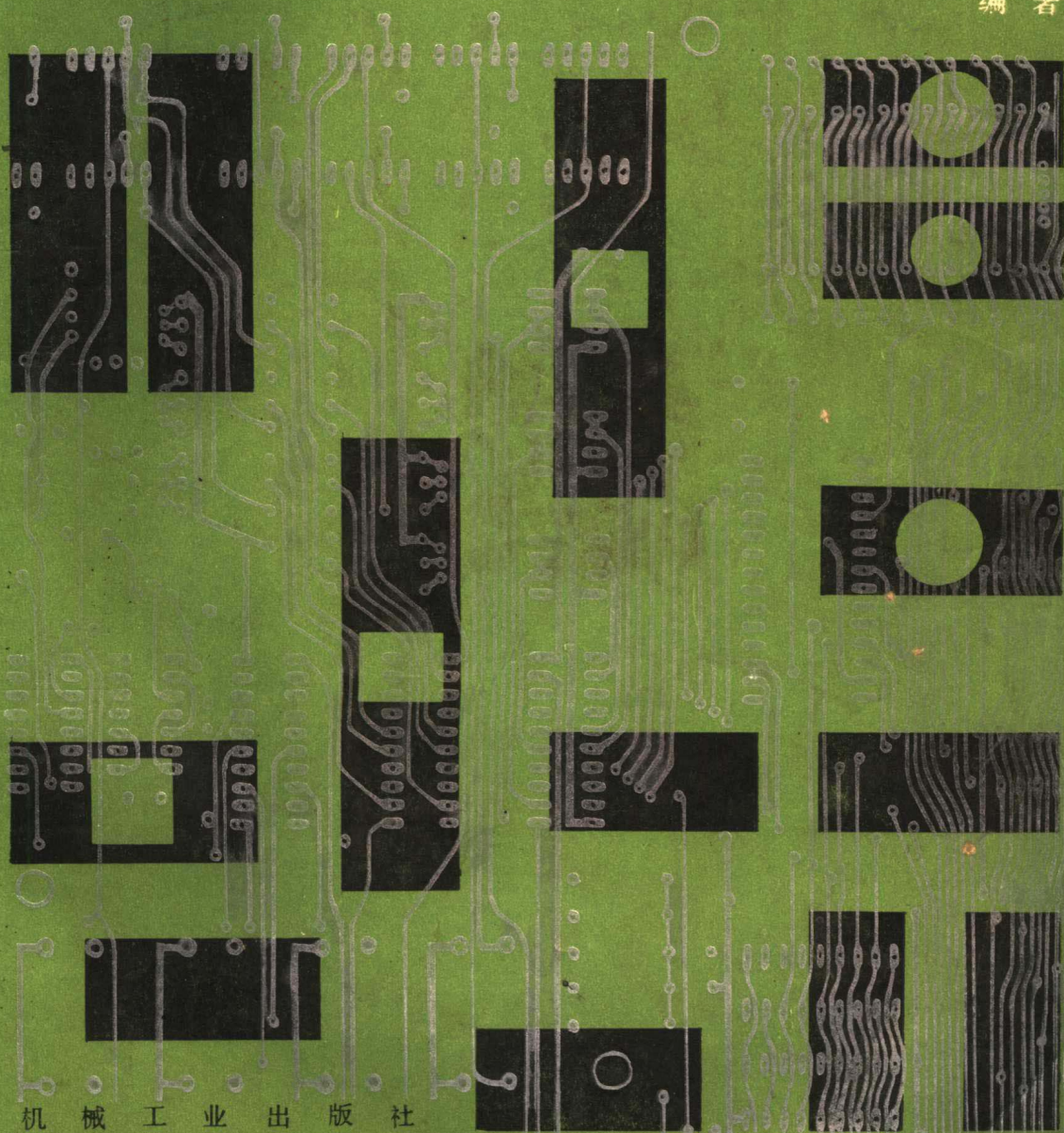


Z80 单板计算机

清华大学

郑学坚 朱善君 严继昌

编著



机械工业出版社

Z 80 单 板 计 算 机

清华大学 郑学坚 朱善君 严继昌 编著



机械工业出版社

内 容 简 介

本书是一本完整的微型计算机培训教材。包括：微型计算机的基础知识，原理，习题及实验。

本书适当介绍 Z 80 单板计算机的硬件线路，着重阐述其指令系统、程序设计、中断方式及接口技术。各章配有相当数量的例题，并附必要的 Z 80 单板计算机技术资料。

本书可作为在职工程技术人员初学微型计算机的自学读物，也可供有关培训班及大专院校非计算机专业教材使用。

Z 80 单板计算机

清华大学 郑学坚 朱善君 严继昌 编著

*

机械工业出版社出版(北京阜成门外百万庄南街一号)
(北京市书刊出版业营业许可证出字第 117 号)

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 $787 \times 1092 \frac{1}{16}$ · 印张 $13 \frac{1}{2}$ · 插页 1 · 字数 332 千字

1985 年 6 月北京第一版 · 1985 年 6 月北京第一次印刷

印数 00,001—22,500 · 定价 3.00 元

**

统一书号：15033 · 6010

前 言

微型计算机及单板计算机在国外已得到广泛的应用,并已深入到各行各业、办公室和家庭中去。在我国也已逐渐受到各方面的重视。但是,由于种种原因,现在的科技人员大多缺乏这方面的知识。用微型机进行实时控制更是感到困难。这主要是由于以前国内计算机数量较少,价格较高的原因。

现在,经过最近几年对各种微型计算机的引进和开发,人们已有机会亲手接触微型计算机。各行业的工程技术人员普遍的愿望是:能在较短的时间内基本掌握单板计算机的工作原理、系统接法、程序设计,以便能够亲手使用一台单板计算机去解决各自所关心的问题。

在北京科技协作中心及清华大学自动化系联合举办的几期培训班上试用了本书作为讲义,反应良好。为了满足更多科技人员的需要,我们将其修订出版。

本书的写作原则是:尽量删除初学者不必深究的某些基础知识,着重阐述其指令系统、程序设计、中断方式及接口技术。再配以Z80单板计算机的实际操作和实验,边学边试,大概在三至四周的时间内,就可学完整个课程,从而初步掌握Z80单板计算机应用的基本技能。

本书的内容共分九章。第一章为基础知识。简要地讲述后面各章必须用到的逻辑代数和某些电路知识。第二、三章为对Z80单板计算机的硬件作适当的介绍。第四章讨论各种指令类型的含义。第五章介绍程序设计。第六章介绍Z80中断方式。第七、八章介绍CTC、PIO的工作原理、接线及编程方法。第九章介绍了键盘、显示器、A/D及D/A转换器的原理、接线及应用。

各章配有相当数量的例题,每章后面附有练习题。书末还有习题答案,以便自学者自我考查之用。为了配合实际操作本书附有实验指示书。

本书由王汝范同志审校。在修订过程中得到了各方面关心计算机技术教育的人士肯定和鼓励,在此致以衷心的感谢。由于我们水平有限,时间仓促,错漏在所难免。恳请读者不吝指正。

编者 郑学坚 朱善君 严继昌

1984年5月于北京清华园

目 录

前 言

第一章 计算机入门基础知识	1
§ 1-1 数制	1
§ 1-2 布尔代数	2
§ 1-3 逻辑门电路	6
§ 1-4 加减算术电路	8
§ 1-5 记忆电路	11
§ 1-6 译码电路	13
§ 1-7 总线联结技术	14
习题	15
第二章 中央处理器	16
§ 2-1 微型计算机结构及指令执行过程	16
§ 2-2 Z80-CPU内部结构	19
§ 2-3 Z80-CPU外部引线	22
习题	24
第三章 单板计算机 (TP-801) 简介	25
§ 3-1 TP-801 单板计算机功能及技术指标	25
§ 3-2 TP-801 单板计算机的电路原理框图	27
§ 3-3 存贮器的联接及存贮空间地址	29
§ 3-4 输入输出接口联接及接口地址	31
习题	34
第四章 Z 80指令系统	35
§ 4-1 Z 80指令助记符	35
§ 4-2 传送类指令	41
§ 4-3 数据操作类指令	47
§ 4-4 程序控制类指令	52
§ 4-5 CPU控制类指令	56
习题	58
第五章 Z 80程序设计	59
§ 5-1 程序设计过程	59
§ 5-2 简单程序	60
§ 5-3 分支程序	63
§ 5-4 循环程序	67
§ 5-5 子程序	70
§ 5-6 查表程序	73
习题	74

第六章 输入输出与中断技术	76
§ 6-1 输入输出概念	76
§ 6-2 输入输出传送方式	78
§ 6-3 中断的概念	83
§ 6-4 CPU响应中断的条件及响应过程	84
§ 6-5 中断源寻找及优先权排队	86
§ 6-6 Z80系统的中断方式	90
习题	94
第七章 Z80-CTC 芯片	95
§ 7-1 Z80-CTC结构和工作原理	95
§ 7-2 Z80-CTC引脚说明	97
§ 7-3 Z80-CTC控制字规约	99
§ 7-4 Z80-CTC初始化程序设计	102
§ 7-5 Z80-CTC应用实例——电子时钟程序设计	104
习题	109
第八章 Z80-PIO 芯片	110
§ 8-1 Z80-PIO的结构	110
§ 8-2 Z80-PIO引脚说明	111
§ 8-3 Z80-PIO控制字规约	113
§ 8-4 Z80-PIO初始化程序设计	117
§ 8-5 Z80-PIO应用实例——打印机接口	120
习题	123
第九章 微型计算机的接口技术	124
§ 9-1 微型机与开关的接口	124
§ 9-2 微型机与LED显示器的接口	125
§ 9-3 微型机与键盘的接口	128
§ 9-4 微型机与A/D转换器和D/A转换器的接口	133
习题	144
附 录	145
附录1 TP801单板计算机键盘操作说明	145
附录2 Z80指令系统	160
附录3 习题答案	189
附录4 实验指示书	202

第一章 计算机入门基础知识

对于计算机本体来说，不管何种型号，不管大小如何，其基本结构是一样的，如图 1-1 所示。

这原因很简单，就是因为它必须能完成数的输入输出、运算、存贮等基本功能。

我们以算盘算题为例，若要计算下述题目

$$13 \times 4 + 72 \div 12 = (52 + 6) = 58$$

我们需要用算盘作运算工具，其次需要用纸和笔来记原始数据、中间结果、最后结果以及运算步骤（先乘除后加减）。人们在算题时：（1）大脑指挥笔写原始数据在纸上，记运算步骤在纸上。（2）然后按运算步骤用算盘计算 13×4 得中间结果 52，记在纸上；算 $72 \div 12$ 得中间结果 6，记在纸上；取出中间结果相加得最后结果。（3）再用嘴告诉人们。

计算机中各部分作用与上述比喻：算盘就是运算器，纸就是存贮器，手和笔就是输入设备，嘴就是输出设备。而大脑就是控制器。

在计算机中，基本上是有两种信息在流动。一种是数据流（用双线带箭头表示），即原始数据、中间结果以及用数码表示的命令代码。一种是控制信息流（图中用单线带箭头表示）。

为了更好地理解这些基本框框（单元）以及信息流在电路上是怎样实现的，我们先离开计算机，学一点数字电路知识。

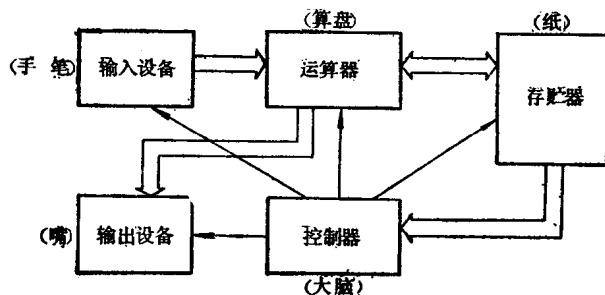


图 1-1 计算机的基本结构

§ 1-1 数 制

数制是人们利用符号来计数的科学方法。数制可以有很多种。但在计算机设计与使用中，经常是十进制、二进制、八进制和十六进制。

数制的基与权：数制所使用的数码的个数称为基；数制每一位所具有的值称为权。

十进制 十进制的基为“十”。即它所使用的数码为 0、1、2、3、4、5、6、7、8、9，共有十个。十进制各位的权是以 10 为底的幂，如下面这个数：

5	2	3	7	9	1
10^5	10^4	10^3	10^2	10^1	10^0
十万	万	千	百	十	一

其各位的权为个、十、百、千、万、十万，即以十为底的 0 幂、1 幂、2 幂…等。故有时为了说话简便，而称各位为 0 权、1 权、2 权、3 权等。

二进制 二进制的基为“二”。即它所使用的数码为 0、1 共两个。二进制各位的权是以 2 为底的幂，如下面这个数：

$$\begin{array}{ccccccc} 1 & 1 & 0 & 1 & 1 & 1 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \text{三十二} & \text{十六} & \text{八} & \text{四} & \text{二} & \text{一} \end{array}$$

其各位的权为一、二、四、八、十六、三十二…，即权是以 2 为底的 0 幂、1 幂、2 幂、3 幂等。

八进制 八进制的基为“八”，即其数码共有八个：0、1、2、3、4、5、6、7。八进制的权为以 8 为底的幂： 8^0 、 8^1 、 8^2 …等。

十六进制 十六进制的基为“十六”，即其数码共有十六个：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F。十六进制的权是以 16 为底的幂： 16^0 、 16^1 、 16^2 ，…。

在微处理器中，这些数制都是会用到的。但在本书后面的内容中，二进制和十六进制更为常用。希初学者注意。

那么在计算机（或微处理器）技术中为什么要用二进制呢？

原因是电路通常只有两种稳态：导通与阻塞，饱和与截止，高电位与低电位等。具有两个稳态的电路称为二值电路。而用二值电路来计数时，只能代表二个数码：0 和 1。如以 1 代表高电位，则 0 代表低电位。所以采用二进制，就可以利用电路进行计数工作。而用电路来组成计算机，则有运算迅速，线路简便，成本低廉等优点。

那么十六进制在计算机技术中为什么也应用得较广泛呢？

原因是用十六进制可以简化二进制数的书写。二进制数想要转换成十六进制数是很方便的。二进制的数只要每四位一分，即可一一对应写出十六进制数。如下列一些等值的数：

$$\begin{aligned} 1\ 0\ 0\ 0_{(2)} &= 8_{(16)} \text{ (即八)} \\ 1\ 1\ 1\ 1_{(2)} &= F_{(16)} \text{ (即十五)} \\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0_{(2)} &= 3A_{(16)} \text{ (即五十八)} \end{aligned}$$

可以看出，十六进制可以把代表同一数量的符号写得短些，所以它更易于记忆。尤其是当二进制数位很多时，更可看到十六进制的优点了。如：

$$1010\ 1101\ 1000\ 0101_{(2)} = AD85_{(16)}$$

显然记 $AD85_{(16)}$ 要比记十六位的二进制数容易得多了。

上面书写的意义：在数字后面加 (2) 或 (16) 是指二进制和十六进制。同理如写 (8) 和 (10)，则表示是八进制和十进制。书写各种数制的数字，也可以用英文字母来表示其数制。用英文字母代表数制时，所用符号如下：

$$\begin{array}{ll} B \text{ —— 二进制} & H \text{ —— 十六进制} \\ D \text{ —— 十进制} & O \text{ —— 八进制} \end{array}$$

例：11 (B) 为三，11 (O) 为九，11 (D) 为十一，11 (H) 为十七。

§ 1-2 布 尔 代 数

布尔代数亦称开关代数或逻辑代数。它是一种因果关系代数

布尔代数也和一般代数一样，可以写成下面这样的表达式：

$$Y = f(A, B, C, D, \dots)$$

但它有两个特点。

第一，其中的变量 A 、 B 、 C 、 D 等均只有两种可能的数值：0 或 1。

所以如用于开关电路则开关是变量（原因），0 代表打开（断路）或低电位，1 代表合上（通路）或高电位。灯亮否是函数（结果），0 表示不亮，1 表示亮。

如用于逻辑推理上，考虑某几个工作的方式和完成任务的关系，则工作方式是原因，0 代表错误，1 代表正确。完成任务与否是结果，0 表示没完成，1 表示完成。

第二，函数 f 只有三种基本运算方式：“或”运算“与”运算，以及“反”（或称“非”）运算。

下面分别讲这三种运算及它们的运算规律。

一、“或”运算

$$Y = A + B$$

或的逻辑运算，可用两个开关并联控制灯的电路来说明。即两个开关有一个合上（为 1）灯就亮（为 1）。

这里式子 $Y = A + B$ 中的符号“+”并不是算术运算中的加，而是代表“或”的逻辑关系。

由于 A 和 B 只有取 1 或 0 的两种值（开和关），所以可以将其各种可能及结果（灯亮或灭）写出如下：

$$\begin{array}{l} Y = 0 + 0 = 0 \\ Y = 0 + 1 = 1 \\ Y = 1 + 0 = 1 \\ Y = 1 + 1 = 1 \end{array} \left. \vphantom{\begin{array}{l} Y = 0 + 0 = 0 \\ Y = 0 + 1 = 1 \\ Y = 1 + 0 = 1 \\ Y = 1 + 1 = 1 \end{array}} \right\} Y = 1$$

这四个式子与一般的算术加法类同，只有第四式不符合加法规律。这是因为 Y 也只有两种取值可能，即 0（不亮）或 1（灯亮）。

上面四个式子可归纳成一句话“有一个 1，即为 1”来记忆。

这个结论也可推广至多变量（ A 、 B 、 C 、 D …等）的布尔代数函数。

$$Y = A + B + C + D + \dots$$

各变量全为 0 时， Y 才为 0。否则，有一个为 1，则 Y 就为 1。

这概念用到多输入的或门电路中，则意味着只要其中一个输入为 1，则其输出必为 1。或者说只有全部输入均为 0 时，输出才为 0。

二、“与”运算

$$Y = A \times B$$

或写成：

$$Y = A \cdot B$$

与的逻辑运算可用开关电路来说明：就是两个开关串起来控制一个灯。只有两个开关都合上，灯才可能亮。

根据 A 和 B 的可能取值 0 或 1，可以列出下列各种可能的运算结果：

$$\begin{array}{l} Y = 0 \times 0 = 0 \\ Y = 1 \times 0 = 0 \\ Y = 0 \times 1 = 0 \\ Y = 1 \times 1 = 1 \end{array} \left. \vphantom{\begin{array}{l} Y = 0 \times 0 = 0 \\ Y = 1 \times 0 = 0 \\ Y = 0 \times 1 = 0 \\ Y = 1 \times 1 = 1 \end{array}} \right\} Y = 0$$

$$Y = 1 \times 1 = 1 \quad Y = 1$$

这种运算结果也可归纳成一句话：“有一个0，即为0”。

同样，这个结论也可推广到多变量的与函数：各变量均为1，结果才会为1，有一个为0，结果必为0。

写成表达式如下：

$$Y = A \times B \times C \times D \times \dots$$

这个概念用到多输入与门电路中，则意味着只要其中一个输入为0，则门电路的输出必为0。或者说，只有全部输入均为1时，输出才为1。

三、“反”运算

$$Y = \bar{A}$$

“反”的逻辑运算也可以用开关电路来说明：即开关并接在灯上。当开关合上时灯中电流被开关短路。灯反而不亮。

因为布尔代数的变量 A 或 B 等只能取0和1的两个值。所以 A 取值为0时， $\bar{A}$ 必取值为1，反之 A 取值为1时， $\bar{A}$ 必为0。

四、布尔代数的基本运算规律

1. 恒等式

$$\begin{array}{lll} A \cdot 0 = 0 & A \cdot 1 = A & A \cdot A = A \\ A + 0 = A & A + 1 = 1 & A + A = A \\ A + \bar{A} = 1 & A \cdot \bar{A} = 0 & \overline{\bar{A}} = A \end{array}$$

2. 运算规律

与普通代数一样，布尔代数也有交换律、结合律、分配律。而且它们与普通代数的规律完全相同。

交换律 $A \cdot B = B \cdot A$

$$A + B = B + A$$

结合律 $(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$

$$(A + B) + C = A + (B + C) = A + B + C$$

分配律 $A(B + C) = AB + AC$

$$(A + B)(C + D) = AC + AD + BC + BD$$

我们利用这些运算规律和恒等式，就可以化简很多逻辑关系式。

例1 $A + AB = A(1 + B) = A$

$$A + \bar{A}B = A + AB + \bar{A}B = A + (A + \bar{A})B = A + B$$

例2 如果原设计继电器线路如图1-2 a。现用逻辑关系，化简线路。

首先我们可以把图1-2 a中

触点（如同开关）和灯的关系用布尔代数表达出来。

$$Y = (A + \bar{A}B) \cdot \bar{B}$$

其中 A 与 $\bar{A}$ 是同一继电器的常开与常闭触点。一般我们把常开触点定为变量 A 、 B 。则常闭触点相应的为 $\bar{A}$ 、 $\bar{B}$ 。

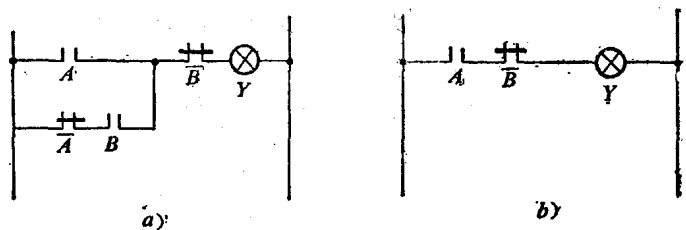


图1-2 用布尔代数化简继电器线路

下面我们来用布尔代数知识简化:

$$\begin{aligned} Y &= (A + \bar{A}B) \cdot \bar{B} \\ &= A\bar{B} + \bar{A}B \cdot \bar{B} \\ &= A\bar{B} + 0 \\ &= A\bar{B} \end{aligned}$$

从而我们可以用图 1-2 中 b 电路, 代替原设计电路 a。电路大大简化了, 但能起到同样的作用。

五、摩根定理

在电路设计时, 人们手边有时没有与门, 而只有或门和反门。或者只有与门和反门没有或门。利用摩根定理, 可以帮你解决元件互换问题。

摩根定理式为:

$$\begin{aligned} \overline{A + B} &= \bar{A} \cdot \bar{B} \\ \overline{A \cdot B} &= \bar{A} + \bar{B} \end{aligned}$$

推广到多变量:

$$\begin{aligned} \overline{A + B + C + \dots} &= \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots \\ \overline{A \cdot B \cdot C \cdot \dots} &= \bar{A} + \bar{B} + \bar{C} + \dots \end{aligned}$$

此定理我们可以用表格法验证。

从表 1-1 中可以看出变量在各种可能取值的情况下, 式 $\overline{A + B}$ 的值始终和 $\bar{A} \cdot \bar{B}$ 相同。所以证明了 $\overline{A + B} = \bar{A} \cdot \bar{B}$ 。同理也证明了 $\overline{A \cdot B} = \bar{A} + \bar{B}$ 。

表 1-1 摩根定理验证

二 变 量	A	B	$\overline{A + B}$	$\bar{A} \cdot \bar{B}$	$\overline{A \cdot B}$	$\bar{A} + \bar{B}$
	0	0	1	1	1	1
	0	1	0	0	1	1
	1	0	0	0	1	1
	1	1	0	0	0	0
变量可能取值			相等		相等	

至于多变量的摩根定理用相同的方法同样可以得到证明。

这个定理可以用一句话, “头上切一刀, 下面变个号” 来记忆使用。

例

$$\begin{aligned} \overline{\bar{A} \cdot \bar{B}} &= \overline{\bar{A} + \bar{B}} = A + B \\ \overline{\bar{A} + \bar{B} + \bar{C}} &= A \cdot B \cdot C \end{aligned}$$

六、真值表及布尔代数式的关系

当人们遇到一个因果问题时, 常常把各种因素全部考虑进去, 然后研究结果。真值表也就是这种考虑问题的方法的一种表格形式。

例如考虑两个一位的二进制数 A 和 B 相加, 其本位的和 S 与向高一位进位 C 的结果如何?

全面考虑两个一位二进制数, 可能出现四种情况: 或 $A = 0, B = 0$; 或 $A = 0, B =$

1, 或 $A = 1, B = 0$, 或 $A = 1, B = 1$ 。(一般 n 个因素可有 2^n 种情况)。这实质是两个一位数(可零可1)的排列。我们可以把它们列入表内。如表 1-2 左边部分所示。然后, 对每一种情况进行分析。当 A 和 B 都为 0 时, S 为 0, 进位 C 也为 0; 当 A 为 0 且 B 为 1 时, S 为 1, 进位 C 为 0; 当 A 为 1 且 B 为 0 时, S 为 1, 进位 C 为 0; 当 A 为 1 且 B 也为 1 时, S 由于是一位数所以得 0, 而有进位 $C = 1$ 。填入表 1-2。如表右部分。

我们称表 1-2 为这个逻辑问题(二进制数相加的运算由于取值只能 1, 0 所以可以转化为逻辑问题)的真值表。

表 1-2

	A	B	S	C
第一项	0	0	0	0
第二项	0	1	1	0
第三项	1	0	1	0
第四项	1	1	0	1

从真值表上我们可以方便地分别写出 S 和 A 、 B 的关系布尔式子, 以及 C 和 A 、 B 的关系布尔式子。

对于 C , 因为只有 A 与 B 都为 1 时, 它才为 1。所以一分析即可知 $C = A \cdot B$ 。

对于 S , 因为在第二项或第三项都可能为 1。而第二项要求 $A = 0$ 与 $B = 1$, 在写式子时要求能使 S 为 1。显然只有写 $\bar{A} \times B = 0 \times 1 = 1$ 。所以第二项布尔式子就是 $\bar{A} \cdot B$ 。对于第三项要求 $A = 1$ 与 $B = 0$, 在写式子时要使 S 为 1, 显然只有写 $A \times \bar{B} = 1 \times 0 = 1$ 。所以第三项布尔式子就是 $A \bar{B}$ 。从而我们可以写出 S 和 A 、 B 的关系式为 $S = \bar{A}B + A\bar{B}$ 。

这种从真值表写出布尔代数式的方法可以用下面两段话来描述。

1. 写布尔代数式先看真值表中结果为 1 的项。有几项就有几个“或”项。
2. 每一项各因素之间是“与”关系。写该项时每个因素都写上, 然后添“反”。至于哪个因素要加“反”(上横线)要看该因素在这项里是否是“0”状态, 是“0”状态则加“反”。否则不加“反”。

写出布尔代数式后, 反过来去检查写得对不对。例: 表 1-2 中第一项 $A = 0$ 和 $B = 0$ 代入式 $S = \bar{A}B + A\bar{B}$ 。则 $S = 0 \cdot 0 + 0 \cdot 0 = 0$; 表中第二项 $A = 0$ 和 $B = 1$ 代入式 $S = \bar{A}B + A\bar{B}$ 则 $S = 0 \cdot 1 + 0 \cdot 1 = 1 + 0 = 1$; 依次类推地代入检查。如果四项都对, 则式子 $S = \bar{A}B + A\bar{B}$ 确实代表了真值表中 S 和 A 、 B 之间的逻辑关系。

一般的说来, 用真值表来描述问题, 不仅全面, 而且通过它来写布尔代数手续也很简便。

§ 1-3 逻辑门电路

在上一节中我们学习了布尔代数, 知道了名词“与”、“或”、“非”(即“反”)等的意义。那么在计算机中电路是怎么样的, 其符号是什么?

因为同一种逻辑关系可以用不同电路来实现。在这里为了示意方便, 我们只用最简单的电路来表示出来。另外符号各个国家也不尽相同。我们只是用常用的一种(美国标准符号)符号表示。

1. 或门 $Y = A + B$

见表 1-3。电路中, 当输入 A 和 B 中有一个是高电位, 输出 Y 即为高电位。

表1-3 各种门电路及符号

序号	名称	电路图	符号	逻辑式
1	或 门			$Y = A + B$
2	与 门			$Y = A \cdot B$ ($Y = A \times B$)
3	非 门 (反相器)			$Y = \bar{A}$
4	或非门			$Y = \overline{A + B}$
5	与非门			$Y = \overline{A \cdot B}$
6	异或门 (异门)			$Y = \bar{A}B + A\bar{B}$

2. 与门 $Y = A \cdot B$

见表 1-3。电路中，输入 A 和 B 都是高电位，输出 Y 才可能是高电位。否则，电源流下的电流将被某一输入端的低电位短路掉。

3. 非门（反相器） $Y = \bar{A}$

见表 1-3。电路中当输入 A 端为高电位，三极管导通，输出 Y 为低电位。反之 A 为低电位，三极管截止，输出 Y 为高电位。

4. 或非门 $Y = \overline{A + B}$

见表 1-3。或非门的电路是由或门和非门组合而成的。在或非门的符号中，只是比或门符号多了一个小圆圈来表示“非”门。

5. 与非门 $Y = \overline{A \cdot B}$

见表 1-3。它类似或非门，是由与门和非门组合而成的。

6. 异或门（异门） $Y = \bar{A}B + A\bar{B}$

见表 1-3。异或门常在加法电路中的应用。其逻辑关系是：只有当两个输入 A 、 B 状态不同时（一个是 0，另一个是 1），输出 Y 才为 1。异或门电路的实现无非是“与、或、非”电路组合而成。我们在表 1-3 中只用符号画出。

7. 三态门

上述六种门电路输出都是两种状态，即不是高电位（1 状态）就是低电位（0 状态）。它们的输出和输出之间是不能相接的。例如图 1-3 a，即是一种错误接法。这种错误接法不但 Y 值无法确定，而且容易把电路损坏。

三态门就是为了解决这一问题而设计的。其电路如图 1-3 中 b。

当 E 端为 0 时，两与门输出为 0。两个三极管基极均为低电位，不导通。即输出 Y 既不是高电位，亦不是低电位。呈悬浮状态。

当 E 端为 1 时，如输入 A 为 1，则 T_1 管基极为高电位， T_1 管导通。输出 Y 接近电源正电位，即为高电位“1”。如输入 A 为 0，则 T_2 管基极为高电位， T_2 管导通。输出 Y 接近电源地电位，即为低电位“0”。这里， Y 和 A 状态是一致的，故 $Y = A$ 。

三态门的符号见图 1-3 c。

它的特点是：当 $E = 0$ 时， Y 悬空（无电流输出，亦不可输入电流。即既不是 1 状态也不是 0 状态）。当 $E = 1$ 时，输入和输出同状态（或 0 或 1，决定输入 A ）。

三态门的符号见图 1-3 c。

它的特点是：当 $E = 0$ 时， Y 悬空（无电流输出，亦不可输入电流。即既不是 1 状态也不是 0 状态）。当 $E = 1$ 时，输入和输出同状态（或 0 或 1，决定输入 A ）。

三态门常常用在总线技术中，起着“信息交通警”的作用。

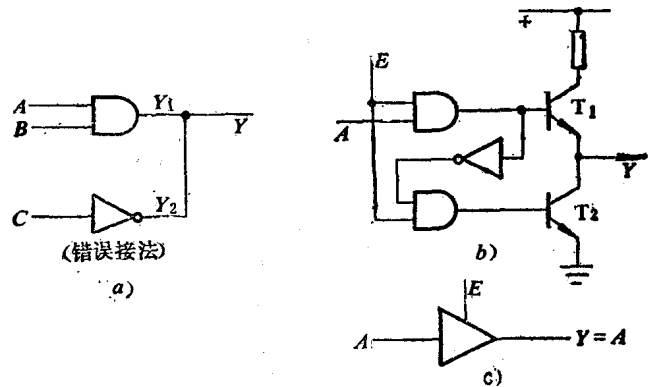


图1-3 三态门的电路及符号

§ 1-4 加 减 算 术 电 路

作为算术的基本运算，众所周知是有加、减、乘、除四种算法。其中乘是累加，除是累

减的演变。所以加减法是基本中的基本。计算机中为了电路简单常常只有加减法器。下面我们来介绍电路是如何实现加减法的。

首先逻辑电路只有高低电平（即电位）两种状态。因此所谓的加法器是基于二进制数上的。

一、加法器

为了研究二进制加法，我们先来找一些逻辑规律。

例如 A 数和 B 数相加。并假设 A 数为 0011 (B)， B 数为 0110 (B) 则：

	1	1	0		进位 C
	0	0	1	1	A 数
+	0	1	1	0	B 数
	1	0	0	1	和数

各位相加（并考虑低位的进位）结果和为 1 0 0 1 (B)。

分析上例可以看出，每一位相加都要考虑三种因素： A_i 、 B_i 、 C_i （ i 代表第 n 位， A_i 即代表 A 数的第 i 位）。这三种因素或 0 或 1 共有 $2^3 = 8$ 种排列。我们把每一位看作是一种因果关系。那么该位的和数 S_i 、以及该位对高位的进位 C_{i+1} ，是 0 或 1 取决于这三种因素的 8 种可能。

列出真值表如表 1-4。

在填写真值表时，我们先填原因部分。 C_i 以 0 和 1 交替填下来， B_i 以两个 0 和两个 1 交替填下来， A_i 以 4 个 0 和 4 个 1 交替填下来。这样填写可保证各种情况都填入不会遗漏。（这实质上是二进制数的顺序排列）然后我们再逐一分析各种情况，把对应的结果，即 S_i 和 C_{i+1} 填进去，以完成真值表的编制。

有了真值表，我们就可以利用 § 1-2 中的方法，写出 S_i 和 C_{i+1} 的布尔代数式子。

$$S_i = \bar{A}_i \bar{B}_i C_i + \bar{A}_i B_i \bar{C}_i + A_i \bar{B}_i \bar{C}_i + A_i B_i C_i$$

$$C_{i+1} = \bar{A}_i B_i C_i + A_i \bar{B}_i C_i + A_i B_i \bar{C}_i + A_i B_i C_i$$

上面两个式子代表了 S_i 和 C_{i+1} 和 A_i 、 B_i 、 C_i 的关系。式子里全是“与、或、非”的关系。显然用逻辑电路实现并不难。但事实上在微处理机中大部分集成电路对使用者来说，只需要知道其输入输出引脚的逻辑关系，而无需详细了解内部的电路。因此在本书中一般只着重介绍“外部特性”。在这儿我们则只给出一位全加器的符号，见图 1-4 a。

上面分析的是一位加法电路。如果二进制数为 n 位，则需要 n 个全加器联起来组成 n 位加法器。见图 1-4 中 b。在加法器中，因为最低一位（ 2^0 位）是没有进位的。所以一般把该位的进位输入 C_0 接地，即 $C_0 = 0$ 就可以了。

二、减法器

实际上，计算机中一般是不另设减法器的，而是利用补码变减法为加法，然后在加法器上运算。

表 1-4 一位加法器真值表

	A_i	B_i	C_i	S_i	C_{i+1}
一	0	0	0	0	0
二	0	0	1	1	0
三	0	1	0	1	0
四	0	1	1	0	1
五	1	0	0	1	0
六	1	0	1	0	1
七	1	1	0	0	1
八	1	1	1	1	1

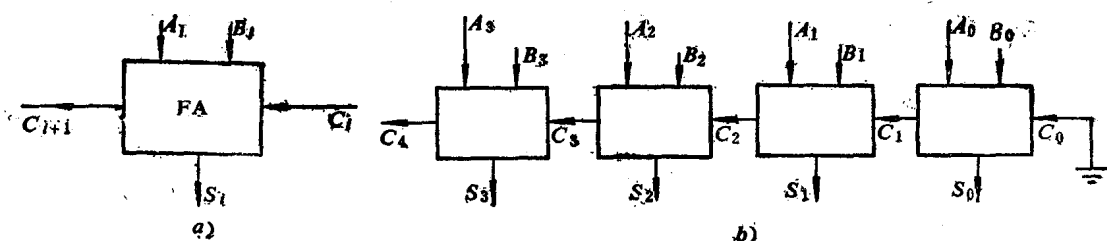


图1-4 全加器及全加器组成的四位加法器

补码与原码之间的关系是：原码加补码正好相互补充，使最高位有进位，后面各位为零。

补码并非只有二进制才有的。在十进制中也有。例，7的补码为3，75的补码为25等。在十六进制中也有。例：A的补码为6，27的补码为D9等。

我们在作十进制减法题 $73 - 15$ 时，利用15的补码为85。使 $73 + 85 = 158$ 。去掉进位即为58。而58正是 $73 - 15$ 的答案。在十进制中求补码并不十分方便，所以很少有人应用这个规律去做减法。

但是在二进制中却不然。二进制中。原码每位变“反”可得反码。例：10100反码为01011。是很容易办到的。而原码加反码正好差1未溢出（无进位）。即：原码 + 反码 + 1正好相互补充。所以：

$$\text{补码} = \text{反码} + 1$$

也即补码很容易求得。因此二进制中作减法运算时常常把减数先变反（即减数的各位都变反）加1，得补码。然后做加法。舍去最高位的进位，即得差数。

例 求 $Y = F(H) - A(H) = ?$

解 被减数： $A = F(H) = 1111(B)$

减数： $B = A(H) = 1010(B)$

B的反码： $\bar{B} = 0101(B)$

则其补码： $B' = \bar{B} + 1 = 0110(B)$

$$\therefore Y = A + B' = 1111 + 0110 = 10101$$

舍去最前面的进位，结果为0101(B)，即5(H)。就是答案。

上面把减数每位变反，电路上的实现就是每位加个反门。而“加1”工作则是把加法器最低位一个全加器进位端 C_0 的输入变“1”状态即可。

图1-5即为利用三态门、反门和全加器组成的两位“加减法器”。

SUB端为控制端。当SUB为1时， $C_0 = 1$ 起到了“加1”的作用，而且使被减数B各位变反后送入全加器，作减法。当SUB为0状态时， $C_0 = 0$ 且使图1-5中上面的一根控制线（加）为

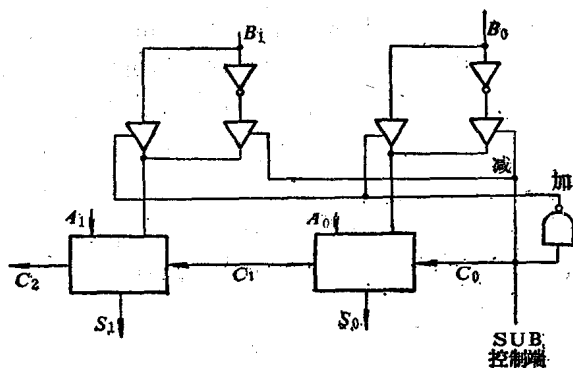


图1-5 两位加减法器

“1”电位，让左边三态门开放， b_1 、 b_0 不变反送入全加器，作加法。

这样我们只要控制SUB端电位，即可使加法器和减法器作加或减的运算。

这种加减法器再加上一些逻辑电路，实际上也就是计算机中的“算盘”——运算器的基本电路原理。

§ 1-5 记 忆 电 路

计算机中记忆电路有各种各样的形式和名称，但其基本作用却都是一样的。它们就是为了记住数或状态而设计的。所以，我们在这儿只举一两个常用电路来理解其工作原理及作用。

一、RS触发器

利用两个“与非门”我们即可组成一个RS触发器，见图1-6。

RS触发器有两个输出端Q和 $\bar{Q}$ 。当S和R没有加信号（低电平信号）时，如Q端先为1状态，则 $\bar{Q}$ 端必为0状态，并通过反馈线到1号元件输入，反向后使Q保持1状态。

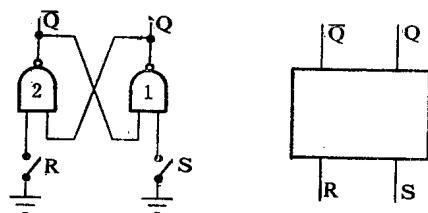


图1-6 RS触发器及符号

RS触发器有两个输入端S和R。如S端接地为0状态，则不管Q为何种状态，S端的低电平信号必使Q为1状态（因为与非门特点是“有一个输入为0，输出就是1”）。从而Q端的1状态馈给2号元件，如R端无信号，则 $\bar{Q}$ 变为0。同样，如S无信号、R端有信号，则 $\bar{Q} = 1$ 而 $Q = 0$ 。

S端可使Q端为1，所以就叫置1端。（SET）

R端可使Q端为0，所以就叫置0端。（RESET）

只要R、S端无新的信号给触发器，触发器将一直保持原状态。即RS触发器有记忆作用。

二、D触发器

RS触发器虽可以记忆状态，但不便多位协调送数。为此人们在RS触发器下面加了四个与非门，组成了D触发器。见图1-7。

图1-3中，D是控制端。cp是时钟脉冲触发端。

平时cp端处在低电位，使元件3、元件4输出为1，无作用于RS触发器。

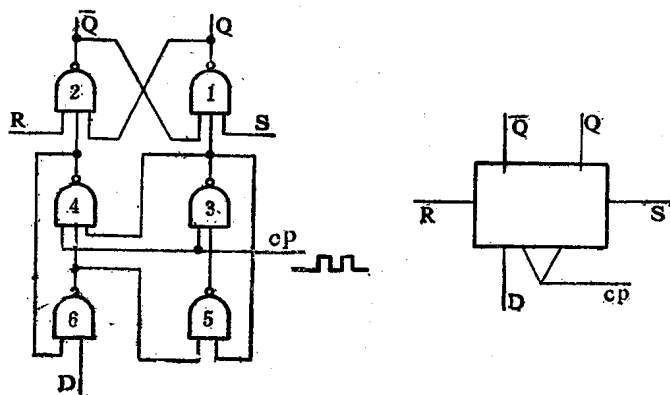


图1-7 D触发器及符号

如 $D = 1$ ，则元件6输入为 $1 \times 1 = 1$ 输出为0，元件6输出又使元件5输出为 $0 \times 1 = 1$ 。

当cp来脉冲变高电位时，由于元件4一端输入是6的输出“0”状态，所以元件4的输出总为1状态不变。而元件3在cp变高电位时，另一端输入是元件5的输出“1”状态，所以输出为 $1 \times 1 = 0$ ，起了S端类似的作用，使Q端为1状态。即置1。