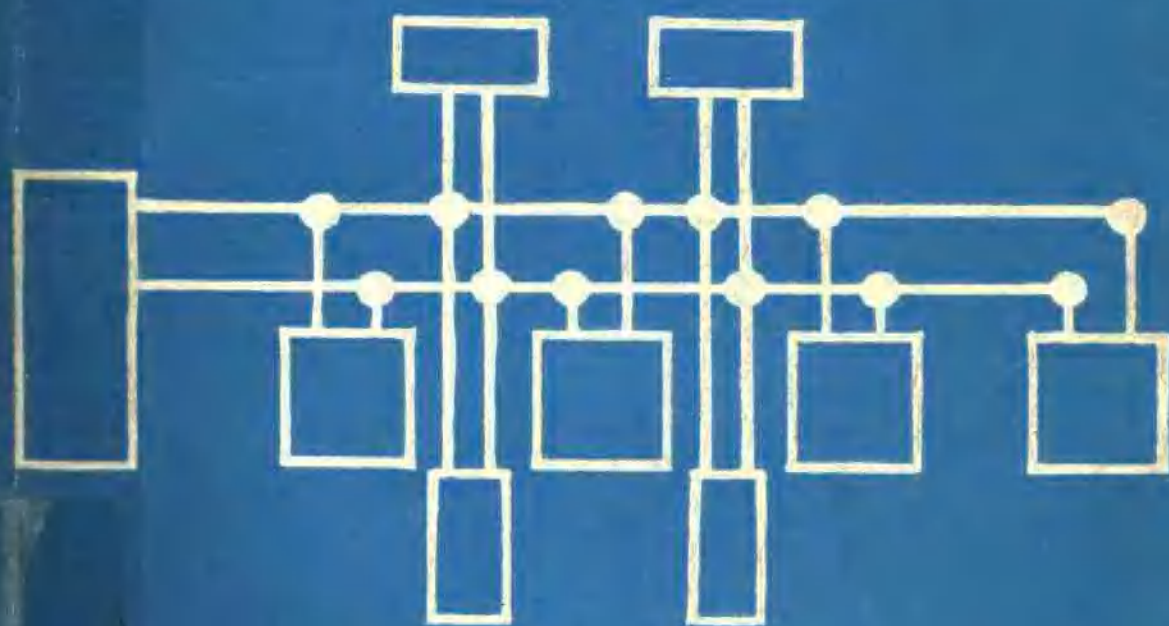


微计算机 应用基础

叶其寿 编著
陈玉彬



武汉工业大学出版社

微计算机应用基础

叶其寿 编著
陈玉彬

武汉工业大学出版社

内 容 简 介

本书系统而简明地介绍了目前广泛流行的 Z80 微处理机及 STD 工业控制机, 6502 微处理机及 APPLE II 微计算机, Intel 8088 微处理机及 IBM-PC 微计算机, MCS-51 单片机, A/D, D/A 转换等结构原理及应用技术。从工程实用观点出发介绍了应用系统设计方法及微机应用实例。

本书适合于各种不同专业需要熟悉微计算机及其应用技术的科技人员参考, 也可作为大专院校有关专业的教学参考书。

微计算机应用基础

叶其寿 班玉彬 编著

武汉工业大学出版社出版发行

新华书店湖北发行所经销

七〇九所印刷厂印刷

开本: 787×1092 毫米¹/₁₆ 印张: 17.5 字数: 406 千字

1990 年 6 月第一版

1990 年 8 月第一次印刷

印数: 3,000

ISBN 7-5629-0357-3/TP·0006

定价: 5.00 元

前 言

微计算机是大规模集成电路和计算技术相结合的产物。随着 LSI 技术的发展, 微计算机性能不断提高, 应用日益广泛, 已深入到工业、农业、交通、国防、文教及日常生活中各个领域。各个不同专业领域的科技人员都需要熟悉和应用微计算机。

本书从工程实用观点出发, 系统而简明地介绍了目前广泛流行的 Z80 微处理机及 STD 工业控制机, 6502 微处理机及 Apple II 微计算机, Intel8088 微处理机及 IBM-PC 微计算机, MCS-51 单片机, A/D, D/A 转换及接口技术。本书重点介绍了在工业过程控制、智能仪器仪表、机电一体化等领域获得愈来愈广泛应用的 STD 总线工业控制机和 MCS-51 单片机的性能、特点及硬件与软件结合的应用设计技术, 并介绍了各种微机应用实例。

本书共分九章, 叶其寿编写第一, 二, 三, 四章, 陈玉彬编写第五, 六, 七, 八章, 两人共同编写了第九章。

由于编者水平有限, 加之时间仓促, 书中难免有错误和不妥之处, 敬请读者批评指正。

编 者 1988.11

目 录

第一章 微计算机概况和基础知识

1.1 微计算机的发展概况	(1)
1.2 计算机中的数制和编码	(1)
1.2.1 二进制数	(1)
1.2.2 十六进制数	(2)
1.2.3 不同数制间的相互转换	(2)
1.2.4 二进制数的运算	(3)
1.2.5 带符号的二进制数	(4)
1.2.6 补码的运算	(4)
1.2.7 二进制编码	(5)
1.3 数字逻辑电路	(5)
1.4 计算机的基本组成	(9)

第二章 Z80 微处理器

2.1 Z80 CPU 的结构	(11)
2.1.1 寄存器组	(11)
2.1.2 算术逻辑运算部件(ALU)	(13)
2.1.3 指令译码器和 CPU 控制	(14)
2.1.4 总线缓冲器	(14)
2.1.5 Z80 引线功能简述	(14)
2.1.6 Z80 CPU 的定时	(15)
2.2 Z80 指令系统	(18)
2.2.1 数据传送指令类	(18)
2.2.2 运算指令类	(21)
2.2.3 比较和查找指令类	(22)
2.2.4 循环和移位指令类	(23)
2.2.5 位操作指令类	(24)
2.2.6 转移、调用和返回指令类	(25)
2.2.7 输入与输出指令类	(27)
2.2.8 通用运算和 CPU 控制指令类	(28)
2.3 Z80 寻址方式	(30)
2.4 Z80 中断系统	(31)

2.4.1 非屏蔽中断	(31)
2.4.2 可屏蔽中断	(32)
2.5 Z80 汇编语言程序设计	(34)
2.5.1 汇编语言的指令类型和语句格式	(35)
2.5.2 简单程序设计	(37)
2.5.3 分支程序设计	(37)
2.5.4 循环程序设计	(38)
2.5.5 查表程序设计	(40)
2.5.6 子程序设计	(44)
2.6 Z80 常用的接口芯片	(45)
2.6.1 Z80 CTC	(45)
2.6.2 Z80 PIO	(51)

第三章 STD BUS 工业控制微机概述

3.1 STD BUS 的基本特点	(58)
3.1.1 总线的基本概念	(58)
3.1.2 STD 总线的特点	(58)
3.2 STD 总线规范	(59)
3.2.1 STD 总线的信号定义和引脚分配	(60)
3.2.2 STD 总线的时序规范	(65)
3.2.3 STD 总线的电气特性规范	(67)
3.2.4 STD 总线的机械特性规范	(68)
3.2.5 STD 总线的演变	(70)
3.2.6 STD 总线模板的编号规定	(71)
3.3 STD 总线母板	(71)
3.3.1 母板的作用和系统对母板的要求	(71)
3.3.2 母板的类型	(72)
3.3.3 改善母板性能的措施	(72)
3.4 半导体存储器 and STD 总线的存储器扩展方法	(73)
3.4.1 可擦除可编程只读存储器 (EPROM)	(73)
3.4.2 静态读写存储器(SRAM)	(75)
3.4.3 存储容量的扩展信号(MEMEX)和段选信号(SEGMENT)的使用	(76)
3.5 中断控制	(79)
3.5.1 串行优先级中断控制	(79)
3.5.2 并行优先级中断控制	(80)
3.6 STD 总线功能模板的设计特点	(81)

第四章 6502 微处理器及 APPLE II 微计算机

4.1	6502 微处理器的内部结构	(83)
4.2	6502 的时序	(85)
4.3	6502 的寻址方式	(86)
4.4	6502 的指令系统	(88)
4.4.1	传送指令	(88)
4.4.2	算术逻辑运算指令	(91)
4.4.3	置标志位指令	(94)
4.4.4	比较指令	(95)
4.4.5	移位指令	(96)
4.4.6	堆栈操作指令	(98)
4.4.7	转移指令	(99)
4.5	6502 汇编语言程序设计	(100)
4.6	APPLE II 微计算机简介	(102)
4.7	APPLE II PLUS 监控程序和小汇编的操作	(107)

第五章 Intel 8088 微处理机与 IBM-PC 微机系统

5.1	8088 微处理机结构	(109)
5.2	存储器组织	(111)
5.3	最小组态和最大组态	(113)
5.4	8088 引线定义	(116)
5.5	指令格式	(117)
5.6	8088 寻址方式	(118)
5.7	8088 指令系统	(119)
5.8	8088 时序	(126)
5.9	IBM PC 系统结构与原理	(129)
5.10	IBM PC 磁盘操作系统简介	(132)

第六章 MCS-51 单片机结构与原理

6.1	单片机概述	(137)
6.2	MCS-51 单片机结构	(138)
6.2.1	运算器	(138)
6.2.2	控制器	(138)
6.2.3	存储器	(139)
6.2.4	输入输出	(140)
6.3	存储器结构与配置	(142)
6.3.1	程序存储器	(143)

6.3.2 内部数据存储器	(143)
6.3.3 专用功能寄存器	(144)
6.3.4 外部数据存储器	(144)
6.4 寻址方式	(145)
6.5 定时器/计数器	(146)
6.5.1 方式 0	(147)
6.5.2 方式 1	(148)
6.5.3 方式 2	(148)
6.5.4 方式 3	(149)
6.6 串行接口	(149)
6.6.1 方式 0	(150)
6.6.2 方式 1	(152)
6.6.3 方式 2 和 3	(153)
6.6.4 多处理机通讯	(154)
6.6.5 波特率	(155)
6.7 中断系统	(155)
6.8 MCS-51 单片机系统外围扩展设计	(159)
6.8.1 程序存储器扩展设计	(159)
6.8.2 数据存储器扩展设计	(162)
6.8.3 输入/输出扩展设计	(163)

第七章 MCS-51 指令系统及应用程序设计

7.1 数据传送类指令	(166)
7.2 算术操作类指令	(170)
7.3 逻辑操作类指令	(172)
7.4 控制转移类指令	(174)
7.5 布尔变量操作类指令	(176)
7.6 MCS-51 单片机应用程序设计	(178)
7.6.1 基数转换子程序 BINBCD	(178)
7.6.2 多精度无符号数加法子程序	(179)
7.6.3 多精度无符号数减法子程序	(180)
7.6.4 双字节整数乘法子程序	(180)
7.6.5 整数除法子程序	(182)
7.6.6 256 种分支转移程序	(184)
7.6.7 大于 256 个分支转移程序	(185)
7.6.8 查表程序	(186)
7.6.9 定时器应用编程	(189)
7.6.10 数据排序子程序	(190)

7.6.11 布尔处理机逻辑操作程序	(191)
7.7 新型 16 位单片机 MCS-96 简介	(191)
7.7.1 MCS-96 结构	(194)
7.7.2 存储器空间	(195)
7.7.3 中断系统	(197)
7.7.4 A/D 转换器和脉宽调制器	(197)
7.7.5 MCS-96 指令系统	(199)

第八章 A/D, D/A 转换器与微计算机接口

8.1 微计算机 I/O 接口技术	(203)
8.1.1 I/O 口的编址	(204)
8.1.2 CPU 与 I/O 口数据传送控制	(204)
8.2 数据采集系统中 A/D 转换器	(205)
8.2.1 8 位 A/D 转换器(ADC 0808, 0809)及其应用	(206)
8.2.2 12 位 A/D 转换-A/D574A 结构及应用	(211)
8.3 数/模转换器(D/A)及应用	(212)
8.3.1 DAC 0832 结构与性能	(212)
8.3.2 DAC 0832 应用及接口设计	(213)
8.4 可编程 A/D, D/A 模块(ST-732)功能及应用	(214)
8.4.1 结构与原理	(215)
8.4.2 ST-732A/D, D/A 模块程序设计	(216)

第九章 微计算机应用系统

9.1 微计算机应用系统设计方法	(218)
9.1.1 应用系统总体设计	(218)
9.1.2 应用系统中计算机的选择	(219)
9.2 智力竞赛抢答报警指示器	(221)
9.3 自动响铃控制系统	(224)
9.4 APPLE II 的数据采集系统	(229)
9.5 高精度位移测控系统	(234)
9.6 稳流电源监控仪	(237)

附 录

附录 1 Z80 指令表 (按字母顺序排列)	(240)
附录 2 6502 指令表	(252)
附录 3 MCS-51 系列单片指令表	(255)

第一章 微计算机概况和基础知识

1.1 微计算机的发展概况

世界上第一台电子数字计算机(简称 ENIAC)是美国于 1946 年研制成功的。它用了 18800 个电子管, 7500 个继电器和开关, 体积 85m^3 , 占地 170m^2 , 重 30 吨, 每秒运算 5000 次, 它的研制成功解决了一系列科学计算课题。它使弹道计算比人工快 1000 倍, 显示了数字计算机的巨大生命力。

从 1946 年至今 40 年间, 电子计算机更新换代十分迅速。从计算机采用的逻辑电路的元器件来看, 它经历了电子管、晶体管、集成电路、大规模集成电路、超大规模集成电路。从存贮器件来看, 它经过阴极射线管、磁鼓、磁芯、半导体等。大规模集成电路的研制成功推动了计算机工业更加飞速发展, 使计算机开始向巨型和微型两极发展。巨型机具有极强的处理能力和极快的运算速度(每秒达亿次以上)。它的外围设备齐全, 软件十分丰富。在科学研究和经济建设中承担着大型的计算课题。但是由于它的价格昂贵, 占地也相对地大而限制了它的普及应用。

随着大规模集成电路的发展, 从微型机这一极开辟了一条崭新的道路。70 年代初 Intel 公司推出第一代八位微处理器 8008, 它一出现就显示出巨大的生命力。从 1973 年起世界上相继推出第二代微处理器 8080、6800、6502 等八位微处理器。从 1975 年起 Zilog 公司把 8080 和 6800 的优点加以综合, 推出当时功能最强的八位微处理器 Z80, 使微处理器进入了第三代。1976 年 Intel 公司推出 MCS-48 八位单片微机系列, 它将中央处理器, I/O 和少量存贮器集成在一个芯片上, 构成一个小巧而完整的微计算机。1980 年 Intel 公司推出的 MCS-51 单片机是在 MCS-48 的基础上发展的八位单片机, 它的功能更强。随着超大规模集成电路问世, 1978 年后就相继出现各种 16 位微处理器, 如 8086、6800、Z8000, 使微处理器进入了第四代。进入 80 年代, 又发展了象 80286、80386(32 位)以及 MCS-96(1983 年推出的 16 位单片机), 使微计算机的发展进入了一个新的发展时期。由于微处理器的功能越来越强, 而价格又不断下降, 因而使微计算机应用到社会的各个领域, 并进入到家庭。我国微机的开发应用也相当迅速, 国产化的比例也在不断提高, 如紫金 II、长城 0520、TP801、TPSTD 等都是我国微计算机的优选机型。

1.2 计算机中的数制和编码

1.2.1 二进制数

平时人们最广泛使用的数是十进制数, 用 0—9 这十个数来进行计数, 逢十就进位, 构成个十百千万。如三百一十五写作 315。

$$\begin{aligned} 315 &= 3 \times 100 + 1 \times 10 + 5 \\ &= 3 \times 10^2 + 1 \times 10^1 + 5 \times 10^0 \end{aligned}$$

计算机是由开关电路组成的，它一般只有二个状态，开/关或高电平/低电平，所以电子数字计算机都采用二进制数进行运算、存贮和传输。

二进制数只有 0 和 1 二个数，逢 2 就进位。十进制数和二进制数的对应关系如下：

十进制数	0	1	2	3	4	5	6	7	8	9
二进制数	0	1	10	11	100	101	110	111	1000	1001

二进制数每位叫一个“比特”(bit)，八位叫一个“字节”(byte)，而四位叫“半字节”(Nibble)。

1.2.2 十六进制数

二进制数表示的数，当位数较多时，写起来是很长的，为了读写方便起见，在计算机的输入输出中又常用十六进制数来表示一个数，它共有十六个数码：0~F，逢 16 进位。十进制数、十六进制数、二进制数的对应关系如下：

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
十六进制	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
二进制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

为便于区别，常在一个数后而如后缀来标明，如

101D (D 表示十进制)

101H (H 表示十六进制)

101B (B 表示二进制)

同是 101，但其值却不同，101B 等效于 5D，101H 等效于 257D。

1.2.3 不同数制间的相互转换

1. 二—十转换

通常采用按幂次展开的方法，如

$$\begin{array}{cccc}
 1 & 0 & 1 & 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow \\
 2^3 & 2^2 & 2^1 & 2^0 \\
 1011B = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
 = 8 + 0 + 2 + 1 = 11D
 \end{array}$$

掌握规律后可按方法连加而得：每进一位乘 2，便得出下列数列，然后将等于 1 的位的数值累加起来，如：

$$\begin{array}{ccccccc}
 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 1011011B = 64 + 16 + 8 + 2 + 1 = 91D
 \end{array}$$

二进制数转换成十进制数也可直接计算。如

$$(0.11)_B = 1 \times 2^{-1} + 1 \times 2^{-2} = 0.5 + 0.25 = 0.75_D$$

2. 十一二转换

常用的方法是“除2取余法”，把十进制数连续除以2，其整数商依次写在左边，直至商等于1为止。若商为奇数，则对应写下1，若商为偶数，则对应写下0。所得数列就是该十进制数的二进制数，例如。

$$(229)_D = (11100101)_B$$

$$\leftarrow (+2)$$

1	3	7	14	28	57	114	229	$\leftarrow$ 整数商
1	1	1	0	0	1	0	1	$\leftarrow$ 余数

十进制小数转换为二进制小数，可采用“乘2取整法”。每次乘2后，将积的小数部分依次写在右边，将积的整数对应写在下面，没有整数写下0。所得数列就是该十进制小数的二进制小数。例如

$$(0.375)_D = (0.011)_B$$

$$0.375 \rightarrow 0.75 \rightarrow 1.50 \rightarrow 1.00$$

小数部分($\times 2$) $\rightarrow$	0	1	1	(小数部分)
----------------------------------	---	---	---	--------

3. 二一十六转换

二进制数转换为十六进制数十分方便，只要把二进制数自右至左每四位划一段，然后分别按段进行转换就得十六进制数，而每4位二进制数与十六进制数的对应关系，前面已说明（参阅1.2.2）。这十六个数的对应关系记牢了，那么十六一二转换也就不难。

十进制数与十六进制数的相互转换，可通过二进制数作为过渡，先将十进制数换成二进制数，再换成十六进制数。

以上介绍的转换方法是人工的转换方法，在计算机内，数的转换则有它的特殊方法。

1.2.4 二进制数的运算

二进制数的四则运算跟十进制数的运算类似，只不过因为只有1、0二个数，所以更简单。但要记住“逢2进1”、“退1为2”。

基本运算关系：

$$0+0=0, 1+0=1, 1+1=10 \text{ (逢2进1)}$$

$$0 \times 0=0, 1 \times 0=0, 1 \times 1=1$$

例1 $(1011)_B + (1101)_B = (11000)_B$

$$\begin{array}{r} 1011 \\ + 1101 \\ \hline 11000 \end{array}$$

例2 $(1101)_B - (1011)_B = (10)_B$

$$\begin{array}{r} 1101 \\ - 1011 \\ \hline 0010 \end{array}$$

例3 $(1101)_B \times (1011)_B = (10001111)_B$

$$\begin{array}{r}
 \\
 \\
 \times \\
 \hline
 \\
 \\
 \\
 \\
 \\
 \hline
 10001111
 \end{array}$$

例 4 $(10001111)_B \div (1011)_B = (1101)_B$

$$\begin{array}{r}
 \\
 \\
 1011 \overline{) 10001111} \\
 \underline{-1011} \\
 \\
 \\
 \\
 \\
 \\
 \hline
 0
 \end{array}$$

1.2.5 带符号的二进制数

跟十进制数相似，二进制数也分正数、负数。计算机做减法运算实际是做加法运算。

如 $45 - 15 = 45 + (-15)$

通常计算机中负数是用补码表示（前面所讲的二进制数叫原码）。

带符号的二进制数最高位定为符号位，0 表示正数，1 表示负数，负数常用补码表示，数的求补方法是先将原码变为反码（即每位是 1 就变为 0，是 0 就变为 1），然后在最低位加 1（叫“取反加 1”）。正数的补码与原码相同。下面以 8 位数的求补为例。

例 $[+18]_{\text{补}} = 00010010_B$

$[-18]_{\text{补}} = 11101101_B + 1 = 11101110_B$

$[+127]_{\text{补}} = 01111111_B$

$[-127]_{\text{补}} = 10000000_B + 1 = 10000001_B$

1.2.6 补码的运算

计算机里二进制数的减法是对其补码进行加法。

例 $79 - 18 = [79]_{\text{补}} + [-18]_{\text{补}} = [61]_{\text{补}}$

$[79]_{\text{补}} = 01001111_B$

$[-18]_{\text{补}} = 11101110_B$

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 11001111
 \end{array}$$

在 8 位微机中，如进位不作特殊处理，就自动丢失。结果 00111101_B 即 $[61]_{\text{补}}$

$18 - 79 = [18]_{\text{补}} + [-79]_{\text{补}} = [-61]_{\text{补}}$

$[18]_{\text{补}} = 00010010_B$

$[-79]_{\text{补}} = 10110001_B$

$$\begin{array}{r}
 00010010 \\
 + 10110001 \\
 \hline
 11000011
 \end{array}$$

结果最高位为 1, 它是负数。

$$[-61]_{\text{补}} = 11000011\text{B}$$

8 位微机表示无符号二进制数的范围 0—255(00000000—11111111)。

而用补码表示时, 最高位为符号位, 表示数的范围是-128—+127。超出这个范围的数进行运算, 计算结果就会溢出。因此在 8 位微机中, 常用多字节运算来扩大数的范围, 提高运算精度。

1.2.7 二进制编码

BCD 码 人们要跟计算机打交道, 只能传送二进制数给计算机, 但人们又习惯用十进制数, 因此创造一种用二进制编码的十进制数, 称为 BCD 码。最常用的 BCD 码是 8421 码, 即每位十进制数的 0—9 用 4 位二进制数来表示, 其对应值如下:

0000	—0
0001	—1
0010	—2
0011	—3
0100	—4
0101	—5
0110	—6
0111	—7
1000	—8
1001	—9

8 位二进制数可表示 2 位 BCD 码, 如

$$(10001001)_{\text{BCD}} = 89\text{D}$$

ASCII 码 它是美国标准信息交换代码, 它用 7 位二进制数来表示数字 0—9、英文字母 A—Z 以及其他符号 (如 +—×÷=?=,.)。7 位二进制数可表示 128 个字符。数字 0—9 对应的 ASCII 码为 30H—39H, 英文字母 A—Z 对应 41H—5AH。

汉字编码 汉字的编码分为汉字输入码和机内编码,“通讯用汉字字符集(基本集)及其交换码标准”按区位编码, 区编号占一个字节, 位编号占一个字节。因此汉字编码必须要二个以上的字节。

1.3 数字逻辑电路

电子计算机看起来十分复杂, 但它的内部构造是由许多简单的逻辑电路组成。

逻辑电路分为组合逻辑和时序逻辑, 组合逻辑的输出仅与当前的输入有关, 而时序逻辑

辑的输出不仅与当前的输入有关，而且也跟先前的输入有关。

基本的逻辑门有“与”门、“或”门、“非”门。用这些门电路就可构成各种复杂的逻辑电路。

“与”门(AND) 最简单的与门有二个输入端和一个输出端。设输入为 A 和 B，输出为 C，则与门的表达式 $C=AB$ 。因计算机的电路是二值逻辑电路，它的值只有 0 和 1 (或低电平和高电平) 两个状态。与门的真值表 (即输入与输出的逻辑关系) 如下，与门的逻辑符号如图 1.1

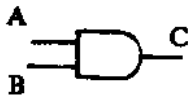


图 1.1

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

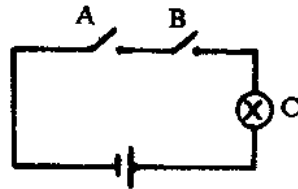


图 1.2

如果用开关和灯泡来比喻 (如图 1.2) 那只有当 A、B 开关都合上，灯才亮。

多输入端与门，只有所有的输入均为 1，输出才为 1。

“或”门(OR) 最简单的“或”门只有二个输入 A 和 B，一个输出 C，所谓或逻辑就是 A、B 二个输入只要有一个是 1 则输出 C 为 1。表达式为 $C=A+B$ ，这是逻辑或运算，不要跟二进制加法混为一谈 (逻辑或运算不进位，而二进制加法“逢 2 进 1”)。或逻辑的真值表如下，逻辑符号如图 1.3。

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1



图 1.3

或门的逻辑关系可用图 1.4 的电路来比喻，二个开关相当于或输入 A、B，灯相当于或输出 C。只要有一个开关合上，灯就亮。

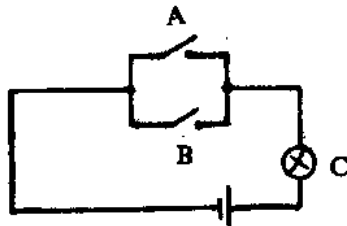


图 1.4

多输入或门，只要有一个或一个以上的输入为 1，则输出就为 1。

“非”门(NOT) “非”门只有一个输入和一个输出。所谓“非”门就是输出状态总是跟输入状态相反，输入 0 输出就为 1，输入 1 输出就为 0。所以非门又叫反相器，逻辑表达式为 $C=\bar{A}$ ，真值表如下，逻辑符号如图 1.5，图 1.6 是用开关灯泡的电路来说明非门，开关合上，灯不亮，开关断开，灯就亮。

A	C
0	1
1	0



图 1.5

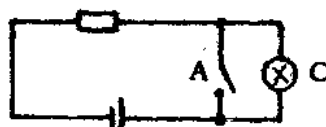


图 1.6

通常在逻辑符号的输出端上的小圆“o”就是表示非（反相）。如与门的输出端加一小圆圈就表示与非门。还有或非门，如图 1.7。

与非门

或非门



图 1.7

“异或”门(XOR) 异或门有二个输入 A 和 B，一个输出 C，只要 A、B 二者的状态不同（相异）则 C 为 1。表达式为 $C = A \oplus B$ 或 $C = A\bar{B} + \bar{A}B$ ，异或门的真值表如下，逻辑符号如图 1.8。

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0



图 1.8

从表达式可看出，异或门可由与门、或门和非门组成，如图 1.9。

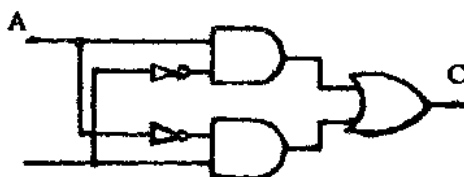


图 1.9

触发器、寄存器、计数器 触发器是有二个稳定状态的时序逻辑单元。我们以常用的 D 型触发器(74LS74)为例来说明它的逻辑功能。它的逻辑符号如图 1.10，它有四个输入端，二个输出端，Q 端常叫 1 端，而 $\bar{Q}$ 叫 0 端，用它代表二进制的一位，Q 端高电平代表 1，低电平代表 0。PR 端为预置输入端，小圆圈表示信号低电平有效。当 PR 为低电平时，把触发器置成 1。CLR 端为复位输入端，当 CLR 输入低电平时，就把触发器置成 0（此时 Q 端为低电平）。D 输入端为数据输入端，CLK 为时钟输入端。如果 PR 和 CLR 保持为高电平，而 CLK 由低变高，触发器变到何种状态取决于 D 端的电平，如 D 为高电平，则触发器为 1 状态，如 D 为低电平，则触发器为 0 状态，跟它的原先状态无关。如 PR 和 CLR 保持高电平，CLK 不产生正跳变，触发器的状态保持不变。所以触发器常用作二进制的寄存器。八位寄存器就需要八个触发器。

当 PR 和 CLR 保持高电平, D 输入端接 $\bar{Q}$ 输出端, 这时 CLK 端输入脉冲信号, 则触发器的状态与原先的状态相反, 原来是“1”就变为“0”, 原来是“0”就变为“1”。如开始是 0, CLK 输入第 1 个脉冲后就变为 1, 第 2 个脉冲后变 0, 第 3 个脉冲后变 1, 这正是一位二进制计数器的逻辑关系, 所以触发器又常用作计数器。一位计数器如图 1.11。

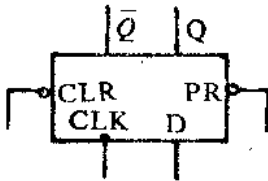


图 1.10

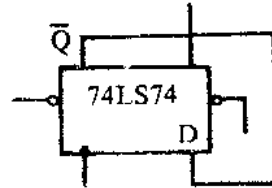


图 1.11

译码器 由一种代码变换（翻译）为另一种代码的逻辑电路叫译码器。计算机中存储器需要有地址译码，控制器需要有操作码的译码，在数码显示器常用到字型七段译码器。下面以 74LS139 为例说明其工作原理。图 1.12 是 74LS139 的逻辑功能图。它将二位二进制代码译成三高一低的四个输出 (Y_0 — Y_3) 所以叫二—四译码器。它的真值表如下。

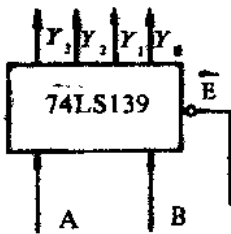


图 1.12

A	B	E	Y_0	Y_1	Y_2	Y_3
×	×	1	1	1	1	1
0	0	0	0	1	1	1
1	0	0	1	0	1	1
0	1	0	1	1	0	1
1	1	0	1	1	1	0

它的内部逻辑结构是由非门和与非门组成。如图 1.13。二个输入 ($2^2=4$) 有四个输出。三个输入的为三—八译码。

三态门 计算机中各部件之间都要相互传输信息，通常一根线需要连接若干设备的输入和输出，这些沟通信息的连线叫总线。如图 1.14 所示。如果输出只有二个状态 0 或 1，那么有的设备输出 0，有的设备输出 1，那就使总线状态无法确定。为解决这一矛盾，而设计一种电路，叫三态驱动器或三态门。它除了有 0 和 1 两个状态外，还有第三个状态，高阻抗输出状态。它的逻辑符号如图 1.15。当 $\bar{E}$ 控制端输入低电平时，它跟普通逻辑电路一样输出电平为 1 或 0 电平。当 $\bar{E}$ 输入高电平时，输出为高阻抗状态，跟输入状态无关，就好像跟总线断开一样。当很多三态驱动器驱动总线时，通过控制 $\bar{E}$ 端，保证在同一时间里只有一个三态门输出信号到总线，其它三态门均处于高阻抗状态。