

高等学校试用教材

严志华 李樟云 编

微处理器及其应用

高等教育出版社

025524

681

H/1

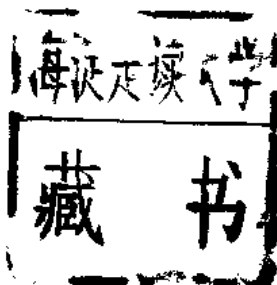
TP368.1
YZH/1



高等学校试用教材

微 处 理 器 及 其 应 用

严志华 编
李樟云



025527

0026925

高等教育出版社

JK16/3
高等学校试用教材

微 处 理 器 及 其 应 用

严志华 编
李樟云

高等教育出版社出版
新华书店北京发行所发行
二二〇七工厂印刷

*

开本787×1092 1/16. 印张 19.75 字数450,000
1986年8月第1版 1986年8月 第1次印刷
印数00,001—4,700
书号13010.01118 定价 3.79 元

序 言

一、微处理器的发展历史

微处理器(Microprocessor 简称 μP)是一片或几片大规模集成电路所组成的中央处理单元,或者说它是微型化的中央处理器。它自1971年首先由Intel公司发明以来,已经有了飞速的发展,对科学技术产生了巨大的促进作用。表0-1列出了四个不同发展阶段中的主要微处理器型号、制造厂家和主要工艺指标。

表 0-1 微处理器的发展概况

年 代	型 号	时 钟 频 率	生 产 厂 家
第一代微处理器 1971—1972 (PMOS)	4004	750kHz	Intel
	4040	750kHz	Intel
	8008	2MHz	Intel
第二代微处理器 1973—1975 (NMOS)	8080 / 8080A	2MHz	Intel
	8085	3MHz	Intel
	6800	1MHz	Motorola
	Z 80	4MHz	Zilog
	6502	1—3 MHz	MOS Technology
第三代微处理器 1976—1979 (十六位HMOS)	8086 / 8088	8MHz	Intel
	6809	4—8 MHz	Motorola
	Z 8000	4MHz	Zilog
	TMS 9900	2—3 MHz	Texas Instruments
	68000	10MHz	Motorola
第四代微处理器 1980— (三十二位)	iAPX 432	(三芯片) 8 MHz	Intel
	68020		Motorola

表中,前面三代产品已十分成熟,价格也急剧下降,正在被广泛地应用。第四代产品——32位微处理器正处于定型和完善阶段,利用这些产品所组成的微型计算机已达到中、小型计算机的能力。另外,为了克服微处理器利用MOS器件而致使工作速率偏低的缺点,一种高速肖特基TTL工艺的位片(Bit-slice)式微处理器(如AMD 2900系列)业已得到了广泛的使用。

二、微处理器的应用概况

随着大规模集成电路技术的高速发展,微处理器及其支持器件的生产量越来越大,价格越来越低,从而使它们的应用越来越广。到目前为止,它已应用于从日常生活到复杂的仪器仪表、工业控制及计算机本身等各个领域之中。最近几年,微处理器已在我国初步推广使用。据有关资料统计,八十年代初的应用情况如下:

过程控制和仪器仪表 26.4%

科学计算	22%
教学和培训	12.4%
企业管理	11.2%
资料检索	4.3%
汉字信息处理	1.6%
其它	22.1%

从以上统计数字可以看出，微处理器在工业控制和仪表测量中的应用居首位，这是因为微处理器的应用带来了以下几点明显的好处：

1. 节省了硬件，降低了成本。各种硬件逻辑在速率允许的条件下，均可以由微处理器系统及其软件代替。根据国外有关学者统计，如果一个实际系统所用的集成电路数量在50个门以上，就可以考虑采用微处理器系统去代替它们，从而降低系统的成本。图0-1表示了机器功能复杂程度和其制造成本之间的关系。从中可以看出，对于功能复杂的机器，特别宜于采用微处理器。因为建立了基本微处理器系统之后，就可以通过不断增加和改进软件，以及扩展少量硬件来完成各种更加复杂的功能。

2. 具有更大的灵活性。在原来的硬件逻辑系统中，当系统建成后，如想改变它，就得重建。这将耗费大量人力和资金。但在以微处理器为基础的系统里，人们可以通过改进它的软件来完成新的任务，大大提高了系统的灵活性。

3. 低功耗、小体积，也提高了系统的可靠性。这是由于微处理器及其支持器件均为MOS工艺制造的大规模集成电路(LSI)，从而使系统功耗小，元器件和焊接点数量也大大减少。

4. 使系统向着“人工智能”的方向发展。人们可以利用编制各种诊断、调试软件，使系统能够自动地发现错误，报告故障部位和原因，甚至自动地排除故障。

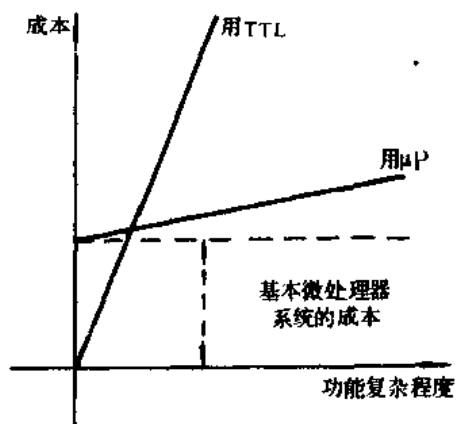


图 0-1 系统的功能和成本的关系

三、本书的编写原则和说明

1. 本书是为非计算机专业人员学习微处理器而编写的一本教材。内容选择是为使事先没有计算机方面的基础知识的读者，通过学习本书能够基本具备将微处理器应用到其所从事的相应领域的能力。

2. 力争根据教学的一般规律编写，以便读者掌握微处理器及其系统设计中的普遍性规律，如微处理器的硬件模型、汇编语言编程、微处理器系统的组建等等。

3. 结合我国目前的实际情况，以Z80微处理器为主要讨论对象。由于Z80是8080A的发展，它包括了8080A的全部指令，故在本书讨论Z80时，经常地附带把8080A的相应问题加以联系和比较，使读者在学会Z80微处理器及其应用的同时，也能对8080A/8085微处理器有

所掌握。

4. 加强“系统”的概念。本书把微处理器等LSI当作组建系统的器件，着重讨论其外部特性，不去过份细致地研究芯片内部结构，以免使本书变成产品手册的综合。正如上面已经讲过的，我们的目的是希望读者通过本书的学习，能够具有一定的应用微处理器等LSI器件来组建自己的微处理器系统的能力。为此，在第一、二、三章介绍了基本准备知识、微处理器原理、汇编语言编程之后，便在第四、五章中讨论一个微处理器系统的组建和调试。然后，再逐一介绍系统中所用的必要器件：存储器、中断控制、并串行接口、定时器等。为了加强读者对系统的认识，我们还引入了若干微处理器小系统和各种编程实例。

5. 加强了对十六位微处理器的基本介绍。考虑到目前国内尚无较详细介绍十六位微处理器的书籍，我们加强了本书第十二章的内容，对目前国内应用得较多的8086、68000、Z8000三种十六位微处理器芯片作了基本介绍，以便读者在学完本书之后，借助于本书附件（指令表），能够初步应用它们。另外，在书后的附录中还引入了我们自己设计并经过试验的一个MC 68000简易单板机和一个8086的简易单板机实例，以供读者进一步自学和动手实践。

6. 在附录中给出了本书的习题集。读者可以根据自己的需要和学习进度选做，并在Z80单板机上验证通过，以巩固和加深学过的知识。

7. 为了便于使用，我们将常用的八位和十六位微处理器的指令表单独汇编成册，定名为“常用微处理器指令表集”作为本书的附件。

8. 我们参照一九八四年十月在九华山召开的综合大学无线电专业计算机教学大纲讨论会上制订的“计算机原理”课程教学大纲，对本书内容做了部份修改和补充。由于时间紧迫，加之水平有限，书中一定存在不妥之处，欢迎读者批评指正。

南京大学信息物理系 严志华
南京工学院无线电工程系 李樟云
一九八四年十二月

目 录

序言	1	Evaluation Kit)	78
第一章 准备知识	1	§ 4.6 一个最简单的Z 80微处理器系 统	79
§ 1.1 代码	1	第五章 微处理器系统的设计和调试	88
§ 1.2 二进制运算	5	§ 5.1 应用微处理器系统的机器设计 方法	88
§ 1.3 逻辑系统和门电路	7	§ 5.2 试验样机软件的调试	89
§ 1.4 一些常用的组合逻辑电路	9	§ 5.3 试验样机硬件的调试	98
§ 1.5 位、寄存器、存储器和计数器	12	第六章 半导体存储器	101
§ 1.6 “线”逻辑	13	§ 6.1 作用和分类	101
§ 1.7 母线和三状态逻辑	13	§ 6.2 随机存取存储器(RAM)	102
§ 1.8 键盘和数码显示器	14	§ 6.3 只读存储器	108
第二章 微处理器工作原理	17	§ 6.4 在微处理器系统中存储器的运 行	113
§ 2.1 一个微处理器的典型结构	17	第七章 中断控制	116
§ 2.2 微处理器的运行	26	§ 7.1 中断控制原理	116
§ 2.3 微处理器的软件	28	§ 7.2 Z 80的中断系统	117
§ 2.4 Z 80 CPU的硬件	33	§ 7.3 Z 80系统的中断优先级控制和 中断嵌套	118
§ 2.5 Z 80 CPU的定时	37	§ 7.4 优先级的软件控制	120
§ 2.6 Z 80 CPU的指令系统	41	§ 7.5 可屏蔽中断方式0的控制	121
§ 2.7 Z 80指令应用实例	47	第八章 可编程并行接口	124
第三章 汇编语言程序设计	52	§ 8.1 数据的输入/输出传送	124
§ 3.1 引言	52	§ 8.2 一个并行接口的结构和编程	127
§ 3.2 机器汇编	55	§ 8.3 对Z 80系列的可编程并行接口—— Z 80 PIO的讨论	129
§ 3.3 用汇编语言编程的基本过程	59	§ 8.4 对可编程并行接口——8255的 讨论	135
§ 3.4 编程技巧	62	§ 8.5 应用举例	141
§ 3.5 编程举例	64	第九章 可编程串行接口	149
第四章 微处理器系统	69	§ 9.1 数据的串行传送	149
§ 4.1 引言	69		
§ 4.2 一个微处理器系统的硬件	70		
§ 4.3 最简单的微处理器系统的外围 设备	74		
§ 4.4 微处理器系统的软件	77		
§ 4.5 单板学习机 (Learning Kit 或			

§ 9.2 一个串行接口的结构和运行.....	152	§ 11.4 最简单单板机实例.....	198
§ 9.3 对Z 80系列的可编程串行接口——Z 80 SIO的讨论	155	第十二章 十六位微处理器简介	213
§ 9.4 对可编程串行接口——8251的讨论.....	163	§ 12.1 概述.....	213
§ 9.5 应用举例.....	165	§ 12.2 Intel 8086/8088.....	214
第十章 可编程计数器/定时器	170	§ 12.3 Zilog Z8000.....	230
§ 10.1 Z 80 CTC	170	§ 12.4 Motorola MC 68000.....	236
§ 10.2 Intel 8253.....	175	第十三章 微型计算机系统介绍	246
§ 10.3 可编程计数器/定时器的应用实例.....	179	§ 13.1 TP-801 A 单板机原理.....	246
第十一章 其它八位微处理器简介	185	§ 13.2 监控程序TPBUG-A 分析.....	249
§ 11.1 Intel 8080 A /8085.....	185	§ 13.3 TRS -80微机系统简介.....	258
§ 11.2 Motorola 6800.....	189	§ 13.4 微型机操作系统	262
§ 11.3 MCS 6502.....	194	§ 13.5 微型机系统总线简介	265
		附录 1 习题集	275
		附录 2 MC 68000单板机	283
		附录 3 8086单板机	289

第一章 准备知识

§ 1.1 代 码

一、自然二进制 (Natural Binary) 码

当我们写一个十进制数157时,这是 $1 \times 10^2 + 5 \times 10^1 + 7 \times 10^0$ 的简化形式,其中 10^0 、 10^1 、 10^2 相应为1、10和100。这是以10为底的逐次幂(个位、十位、百位),10的各次幂的倍乘系数可以是低于底数的任意整数,即0、1、2、3、4、5、6、7、8、9。

二进制数制是一个以2为底数的数字系统,倍乘系数只有0或1两个,以2为底的逐次幂为 2^0 、 2^1 、 2^2 、 2^3 、...或1、2、4、8、16、...等等。对应于一个十进制数,可以用一个二进制数来表示。

如上述的十进制数157可以表示为一个二进制数10011101,它等于

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

或

$$1 \times 128 + 1 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 1 \quad \text{即} 157$$

这一种描绘数的系统是自然二进制码,简称为二进制码,一个长度为n的二进制系统可表示 2^n 个数,这些数的最大值为 $2^n - 1$ 。二进制系统用在数据处理机中特别方便,这是因为在物理现象中可以用电流的有无,光的亮暗,电位的高低,开关的接触与否等等来代表二进制数的1或0。

二、位 (Bit)、字节 (Byte)、字 (Word)

在计算机技术中,广泛使用着一些特殊的名词,如位、字节、字等等,下面对这些名词作一简单的解释。

位 有时也根据其英文发音称为比特。在二进制数的表示中,每一个以2为底的幂称为一位,由于底数为2,故每一位只有0或1两种系数。一个长度为n的数就是一个n位数,如1001为一个4位数,111为一个3位数。数右端的位称为最低有效位,一般称为第0位;数左端的位称为最高有效位,一般称之为第n-1位。

字节 定义每八位二进制数为一个字节,有时也根据其英文发音称为拜特。在微处理器系统中,其基本存储单元几乎都是八位,也就是一个存储单元存放一个字节二进制数。

字 它是计算机中参加运算的基本单位,字的二进制位数称为字长,是计算机的一个重要特征。微处理器中,一般字长为8、16、32、64位,也就是说,一个字可以有1、2、4、8个字节。

三、二—十进制 (BCD) 码

也可以把一个十进制数(如157)的每一个系数以二进制形式表示, 这样就有

$$157 = \underbrace{0001}_1 \underbrace{0101}_5 \underbrace{0111}_7$$

这种数制称为以二进制表示的十进制系统(Binary Coded Decimal 缩写为BCD)简称为二—十进制码, 这种码也用在微处理器中。

四、十六进制 (Hexadecimal) 码

在计算机中, 机器所能真正识别的只能是由0和1组成的二进制表示形式, 但是它们却十分繁琐和极易出错, 因此人们用一位十六进制码来代替四位二进制码。例如用5表示0101, 用9表示1001, 用B表示1011, 用F表示1111等等。这种十六进制码就是以16为底的数制表示法, 它的系数可以为0, 1, 2, ..., 9, A, B, C, D, E, F等十六个数字和字母。表1-1表示了十六进制、十进制和二进制的对应关系。

例如, 一个十六进制码3AC相应的二进制和十进制分别为:

001110101100和940

$$940 = 3 \times 16^2 + 10 \times 16^1 + 12 \times 16^0$$

因为十六进制码是二进制码的简化表示法, 因此, 在微处理器系统的机器语言编程中, 人们总是以它来表示指令和数据。另外, 还有一种八进制码, 它是将二进制码的三位以一个八进制(0—7)数表示, 其情况和十六进制码相类似, 由于目前在微处理器中已很少使用, 故这里不再作详细讨论。

表 1-1 三种数制间的关系

十 进 制 码	二 进 制 码	十 六 进 制 码
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

五、数制的表示法

在微处理器系统的程序编写中, 为了方便, 经常多种数制并用, 为了区别不同的数制, 经常在一个数之后跟着一个字母, 用来注明这一个数是一个什么进制的数。用的字母为B、O(Octal)、D、BCD、H, 分别用来表示该字母前面的数是一个二进制、八进制、十进

制、二十进制、十六进制的数。

六、数制间的转换

由于人们长期习惯于用十进制，因此要知道某一个码的数值大小时，常常要将它转化为十进制的等效值，这个转换十分容易，只要按前面提到的各次幂乘上其相应的系数，然后相加即可得到。例如：

$$10101\text{B} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 21\text{D}$$

$$703\text{O} = 7 \times 8^2 + 0 \times 8^1 + 3 \times 8^0 = 451\text{D}$$

$$\text{D}0\text{E}4\text{H} = 13 \times 16^3 + 0 \times 16^2 + 14 \times 16^1 + 4 \times 16^0 = 53476\text{D}$$

又由于计算机内，存储的信息均是以二进制码的形式存在，故二进制和其它进制之间的转换也十分重要。从十六进制码、BCD码、八进制码的定义中已经可以看出它们和二进制码之间的转换关系。

例如：

$$10101111\text{B} = 257\text{O} = \text{AFH}$$

$$305\text{O} = 11000101\text{B}$$

$$\text{B}4\text{H} = 10110100\text{B}$$

留下的一个复杂问题是：十进制如何向十六进制、八进制、二进制转换。十进制数向这三种数制（也可推广到任意进制）转换的原则是相同的，这里，以十进制向十六进制的转换为例说明如下。

一个十进制数可能包括有整数部分和小数部分，这两部分必须分别作向十六进制的转换。例如有一个十进制数为12983.65625。

整数部分的转换，用连除底数得余数，直至商为零的办法。

$$\begin{array}{r} 16 \overline{) 12983} \\ 16 \overline{) 811} \quad \text{得余数} = 7 \\ 16 \overline{) 50} \quad \text{得余数} = 11 = \text{B} \\ 16 \overline{) 3} \quad \text{得余数} = 2 \\ 0 \quad \text{得余数} = 3 \end{array}$$

$$\text{故 } 12983\text{D} = 32\text{B}7\text{H}$$

小数部分的转换，用连乘底数取其整数，直至小数部分为零的办法。

$$\begin{array}{r} 0.65625 \\ \times \quad 16 \\ \hline 10.5 \quad \text{得整数} = 10 = \text{A} \\ 0.5 \\ \times \quad 16 \\ \hline 8.0 \quad \text{得整数} = 8 \quad \text{小数部分为零} \end{array}$$

故 $0.65625D = 0.A8H$

因此 $12983.65625D = 32B7.A8H$

验证:

$32B7.A8H$

$$= 3 \times 16^3 + 2 \times 16^2 + 11 \times 16^1 + 7 \times 16^0 + 10 \times 16^{-1} + 8 \times 16^{-2}$$

$$= 12288 + 512 + 176 + 7 + 0.625 + 0.03125$$

$$= 12983.65625D$$

必须注意的是, 有时小数部分的转换只能取近似值。

十进制向二进制和八进制的转换同上面原则类似, 读者可以自行练习。

七、ASCII 码

在数据处理和通信传递中, 不但要把数据编为码, 而且也要把字母和某些符号编成码。最常用的包括数、字母和某些符号的码为 ASCII 码 (美国信息交换标准代码 U.S.A. Standard Code for Information Interchange), ASCII 码是一个七位的二进制代码, 表 1-2 列出了七位二进制表示的 ASCII 码。当用一个字节来表示 ASCII 码时, 第七位对 ASCII 码是空着不用的, 在传输信息时, 为了便于检验所传送的代码是否有错, 空着的第七位可用来设置奇偶检验位, 如采用偶数检验, 则每一个包括奇偶检验位的代码中所有“1”的个数应为偶数; 如采用奇数检验, 则每一个包括奇偶检验位的代码中所有“1”的个数应为奇数。

在微处理机中经常使用六位二进制表示的 ASCII 码, 它的内容包括七位 ASCII 码表 (表 1-2) 中的 2、3、4、5 四列。

表 1-2 七位 ASCII 码代码表

行 \ 列	十六进制码高位	0	1	2	3	4	5	6	7
	十六进制码低位	十六进制码高位	十六进制码低位	十六进制码高位	十六进制码低位	十六进制码高位	十六进制码低位	十六进制码高位	十六进制码低位
	3210位	854位	000	001	010	011	100	101	110
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	^	n	~
F	1111	ST	US	/	?	O	_	o	DEL

注: 第 0、1、2、7 列中特殊控制功能符号的意义。

NUL	空	DC1	设备控制 1
SOH	标题开始	DC2	设备控制 2
STX	正文开始	DC3	设备控制 3
ETX	本文结束	DC4	设备控制 4
EOT	传输结束	NAK	否定
ENQ	询问	SYN	空转同步
ACK	承认	ETB	信息组传输结束
BEL	报警符 (可听见声音)	CAN	作废
BS	退一格	EM	纸尽
HT	横向列表 (穿孔卡片指令)	SUB	减
LF	换行	ESC	换码
VT	垂直制表	FS	文字分隔符
FF	走纸控制	GS	组分隔符
CR	回车	RS	记录分隔符
SO	移位输出	US	单元分隔符
SI	移位输入	SP	空格
DLE	数据键换码	DEL	作废

§ 1.2 二进制运算

微处理器能进行二进制数的加法或减法计算。

一、二进制加法

二进制加法的原理和十进制加法相同，两个二进制数同一位相加可以有下列几种情况：

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

当低位数相加后有进位时：

$$0 + 0 + 1 = 1$$

$$0 + 1 + 1 = 10$$

$$1 + 0 + 1 = 10$$

$$1 + 1 + 1 = 11$$

也就是说两数同一位相加，它们的和大于 1 就有进位。

例 1：

	进位位	第7位		第0位	
		↓		↓	
			0 1 0 0 0 1 1 0 B		7 0 D
+			1 0 0 0 1 1 0 0 B	或	1 4 0 D
			1 1 0 1 0 0 1 0 B		2 1 0 D

例 2:

		1 1 0 0 0 0 1 1 B		1 9 5 D
+		1 0 0 1 1 0 0 1 B	或	1 5 3 D
		1 0 1 0 1 1 1 0 0 B		3 4 8 D

从上述两例可以看出，两个八位二进制数相加，它们的和可以是一个八位二进制数，也可以是一个九位二进制数，这个多出的一位称为进位位。

二、二进制反码

把一个二进制数的每一位取反，得到的数称为原来二进制数的反码，而原来的二进制数称为原码，反码也称为1的补码(1's Complement)。

例：0 1 0 1 0 1 0 1 原码

1 0 1 0 1 0 1 0 反码

三、二进制补码(2's Complement)

一个二进制正数的补码就是它的原码。

一个二进制负数的补码是它的原码取反加1。

例：0 1 0 1 0 1 0 1 原码(85 D)

1 0 1 0 1 0 1 1 补码(-85 D)

四、二进制减法

在计算机中，两个数相减，是把减数改为它的补码后，再按二进制加法进行运算。

例 1: 99 D - 87 D = 12 D

99 D 的二进制数 0 1 1 0 0 0 1 1

87 D 的二进制数 0 1 0 1 0 1 1 1

对 87 D 取反加 1 1 0 1 0 1 0 0 1

	0 1 1 0 0 0 1 1
+ 1 0 1 0 1 0 0 1	

1 0 0 0 0 1 1 0 0

↑

进位位

不考虑进位位，00001100 B 为 12 D 确是 99 D - 87 D 的差。

例 2: $87D - 99D = -12D$
 87D 的二进制数 0 1 0 1 0 1 1 1
 99D 的二进制数 0 1 1 0 0 0 1 1
 对 99D 取反加 1 1 0 0 1 1 1 0 1

0 1 0 1 0 1 1 1
 + 1 0 0 1 1 1 0 1

0 1 1 1 1 0 1 0 0

↑

进位位

不考虑进位位 0，这里结果 11110100B 是 $-12D$ 的补码表示，它正是 $87D - 99D$ 的正确结果。如果再考虑进位位，在被减数为正数的情况下，减法结果：进位位为 1，表示结果为正值；进位位为零，表示结果为负值。在下一章中，将进一步讨论实际微处理器中进位位的形成和作用。

§ 1.3 逻辑系统和门电路

一、逻辑系统

任何一个二进制码总是由“1”和“0”组成。在电子技术中，经常用电平或电压的高低来表示二进制码中的“1”和“0”。如果用电压较高的一个值表示逻辑“1”，另一个表示逻辑“0”，则这是一个正逻辑系统，如图 1-1 (a) 所示。如果用电压较低的一个值表示逻辑“1”，而另一个表示逻辑“0”，则这是一个负逻辑系统，如图 1-1 (b) 所示。

由于器件和工作环境的不一致性，加上系统受到外界干扰，或内部相互之间的干扰，因此对逻辑“0”和“1”的电平不可能也并不要求保持在一个严格规定的值上，只要它们在规定电平上下的一定范围之内即可。

在动态系统中，用脉冲的有或无来规定逻辑“1”或“0”，而不考虑电平本身的大小。一个正逻辑系统中用正脉冲来代表“1”，无脉冲时的参考电平为“0”。

逻辑系统中的逻辑运算都是通过门电路来进行的，最基本的门电路为“与”门、“或”门和“非”门，其它的运算也都可以通过这三种门的组合来达到。

二、“与”门

“与”门有两个或两个以上的输入端和一个输出端，它的逻辑关系为：只有当所有的输入端都为“1”时，输出端才为“1”。图 1-2 是“与”门的电路符号，它的逻辑表达式为：

$$Y = A \wedge B \wedge \cdots \wedge N$$

式中符号“ $\wedge$ ”表示逻辑“与”（也可用“ $\cdot$ ”、“ $\times$ ”等表示）。

从上述定义出发，一个两输入端“与”门的输入电平和输出电平之间的关系如表 1-3 所示，称这种表为真值表。

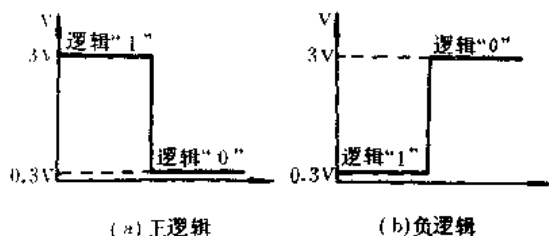


图 1-1 逻辑电平

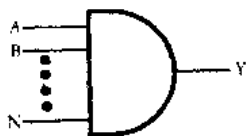


图 1-2 与门符号

表 1-3 两输入端
“与”门真值表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

三、“或”门

“或”门也有两个或两个以上的输入端和一个输出端，它的逻辑关系为：

$$Y = A \vee B \vee \dots \vee N$$

式中符号“ $\vee$ ”表示逻辑“或”（也可用“+”表示）。

根据上述定义，一个两输入端“或”门的输入电平和输出电平之间关系如表1-4所示。

四、“非”门

“非”门只有一个输入端和一个输出端，它的逻辑关系为：输出端与输入端之间处在逻辑反的状态。图1-4是“非”门的电路符号。

“非”门的表达式为：

$$Y = \bar{A}$$

式中A上的符号“—”表示逻辑取反。

“非”门输入端和输出端之间电平关系如表1-5所示。

表 1-4 两输入端或门的真值表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



图 1-3 或门的符号



图 1-4 非门的符号

表 1-5 非门
的真值表

A	Y
0	1
1	0

五、组合逻辑

上面讨论的“与”门、“或”门和“非”门是三种基本的门电路，任何一个数字系统都可以用这三种基本门电路来组合，从而出现了各种各样的组合逻辑电路产品，如“异或”门、“与或非”门及其它更加复杂的组合门电路。

“异或”门是一种常用的简单组合逻辑电路，它具有两个输入端一个输出端，逻辑关系为：当两个输入端中只有一个是“1”，而另一个是“0”时，输出端才为“1”，否则输出为“0”。“异或”门的组合方式有很多种，图1-5是其中的一种，它的逻辑表示式为：

$$Y = A \oplus B$$

式中符号“ $\vee$ ”表示逻辑“异或”(也可用“ $\oplus$ ”表示),它的输入和输出之间的关系如表1-6所示。

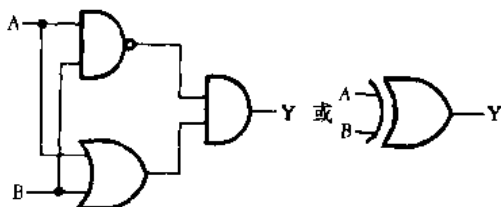


图 1-5 “异或”门

表 1-6 “异或”
门的真值表

A	B	Y
0	0	0
1	0	1
0	1	1
1	1	0

§ 1.4 一些常用的组合逻辑电路

在微处理器系统中,经常要用到一些组合逻辑电路,这里选择较为重要的几种进行介绍。

一、二进制加法器

一台计算机必须完成加、减、乘、除等算术运算,由于乘法实际上是加法运算的重复进行,除法实际上是减法运算的重复进行,而减法又可通过对减数的补码进行加法运算来实现。因此,计算机中的基本运算是加法。

对于两个一位二进制数相加可以用一个“异或”门和一个“与”门来完成(图1-6),图中A、B分别为被加数和加数,D和C为两者之和,其中D为和的本位值,C为进位,它的逻辑关系如表1-7所示,这一电路称为半加器。

对于多位二进制数相加,每一位不但有本位的被加数和加数,而且还有较低位的进位,因此用一个半加器不能完成一位完整的加法运算,用两个半加器和一个“或”门可以组成一个完整的加法器(图1-7),其中 A_n 、 B_n 为本位的被加数和加数, C_{n-1} 为较低位的进位位, S_n 为和的本位值, C_n 为向高位的进位。这一电路称为全加器。在实际的器件中,为了减少门的个数,全加器并不是由两个“异或”门组成,而是用更简单的门电路组成。

二、奇偶校验器/发生器

在数据传输中,为了发现在传输过程中产生的差错,经常采用各种检测方法,奇偶校验是常用方法之一,所谓奇偶校验有两种:

1. 在发送端对每一个字符的所有位相加,总和应为一奇数,若是偶数,则再加一校验位“1”,使包括校验位的总和为奇数;若是奇数,则再加一校验位“0”,结果包括校验位仍为奇数。在接收端检查接收到的包括校验位在内的每个字符时,如果“1”的总和为奇数,说明传输中没有发生奇数个一位码元错误,称这种校验为奇校验,奇校验不能发现偶数个一位码元错误。

2. 使包括校验位在内的每个字符中,“1”的总和为偶数的检测方式称之为偶校验,偶校验不能排除发生偶数个一位码元的错误。