

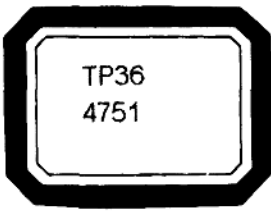
中等专业学校教材

微处理器及接口

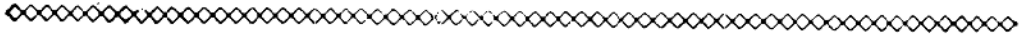
郑州电力学校 杨振 主编



177049



中等专业学校教材



微处理器及接口

郑州电力学校 杨振 主编

水利电力出版社

(京)新登字115号

内 容 提 要

本书依据电力类中专《微处理器及接口》教学大纲编写。内容分八章,前六章围绕Z80微机介绍微机原理和应用,后两章分别介绍MCS-51单片机与可编程序控制器。

本书从中专教学水平出发,注意精选内容,突出重点,使读者能在较短时间掌握微机基本原理,为在过程控制领域应用微机打好基础。

各章均有不同类型的习题,供加深理解概念和编程练习之用。

本书除作为电力类中专(非计算机专业)教材外,还可供初学微机的工程技术人员参考。



中等专业学校教材
微 处 理 器 及 接 口
郑州电力学校 杨振 主编

*

水利电力出版社出版

(北京三里河路6号)

新华书店北京发行所发行·各地新华书店经售

水利电力出版社印刷厂印刷

*

787×1092毫米 16开本 11.75印张 264千字
1993年6月第一版 1993年6月北京第一次印刷

印数00001—13590册

ISBN 7-120-01753-5/TM·464

定价5.50元

前 言

在电力生产和调度自动化中，微机的应用日趋广泛。微机在过程控制中的应用，要求使用者有较深的软件和硬件知识。

1985年，电力类中专计算机课程组会议建议对普通专业（发电、热动等）开设微机课，受到了领导的重视和支持，将微机课列入教学计划。在这前后，各兄弟学校相继开了课。1987年课程组会议在交流各校经验的基础上，制订了教学大纲，并以《微处理器及接口》作为课程名称。

本书在编写过程中，受到了课程组和兄弟学校的大力支持。1989年8月课程组会议讨论了编写提纲，1990年3月初审了教材初稿。在教材出版之际，谨向兄弟学校的老师们致以衷心的感谢。

本书由郑州电力学校罗杰兴（编写第二、三章）和杨振编写，由杨振任主编，由重庆电力学校何宗琦主审。

由于编者水平所限，书中的缺点和错误在所难免，欢迎同志们批评指正。

编 者

1992年6月

目 录

前 言

第一章 微机基础知识	1
第一节 概述	1
第二节 数制与码制	2
第三节 微机组成与工作原理	7
第四节 Z80微处理器	16
第二章 指令系统	23
第一节 寻址方式	23
第二节 传送指令	31
第三节 算术逻辑指令	41
第四节 移位和循环指令	44
第五节 转移指令	48
第六节 子程序调用和返回指令	50
第三章 汇编语言程序设计	54
第一节 概述	54
第二节 伪指令	56
第三节 汇编语言源程序的手工汇编	57
第四节 程序设计初步	59
第四章 半导体存储器	74
第一节 概述	74
第二节 静态RAM和动态RAM	74
第三节 EPROM和EEPROM	76
第四节 半导体存储器与CPU的连接	77
第五节 CPU读写内存的时序	78
第五章 输入输出原理	82
第一节 接口编址方式	82
第二节 Z80的输入输出指令	83
第三节 CPU与外设数据传送的方式	84
第四节 中断	88
第六章 常用接口	97
第一节 Z80-CTC	97
第二节 Z80-PIO	103
第三节 串行接口简单介绍	111
第四节 数/模与模/数转换芯片	112
第七章 单片微型计算机	120

第一节 概述	122
第二节 8051单片机的结构	123
第三节 8051指令系统	130
第四节 单片机扩展	136
第五节 单片机的开发	137
第八章 可编程序控制器 (PC)	140
第一节 概述	140
第二节 PC的组成	141
第三节 PC的编程	142
第四节 PC产品	143
附录	151

第一章 微机基础知识

本章介绍微机发展概况、数制和码制、微机的组成与工作原理、微处理器的结构，这些内容是学习微机的基础知识。

第一节 概 述

随着计算机科学和电子技术的发展，人们用大规模集成电路制成了微处理器。以微处理器为核心的计算机，称为微型计算机，简称微机。

微机具有许多优点：体积小、价格便宜、耗电少、工作可靠、对环境要求低、维护简便，因而深受用户欢迎，在各行业中都获得了广泛的应用。

在电力部门，微机除用于电力系统计算和企业管理外，还用于过程控制。例如用微机实现发电厂生产过程的自动控制以及电网调度自动化，可以提高供电可靠性，降低生产成本，提高电能质量，减轻运行人员的劳动量。

第一个微处理器是美国Intel公司于1971年推出的，字长是4位。以后的发展非常迅速，至今已经历了4代，工艺不断改进，功能不断提高。

在发展微处理器的同时，还发展了单片微型计算机，简称单片机。单片机是将计算机的主要组成部分（中央处理器、存储器、输入/输出接口等）制作在一个芯片上。表1-1列

表 1-1 微处理器和单片机的发展

代	年 份	典 型 产 品	字 长 (位)	基本指令执行时间 (μ s)
1	1971~1973	Intel 4004	4	10
2	1973~1977	Intel 8080	8	2
		Motorola 6800	8	2
		Zilog 80	8	2
		Intel 8048(单片机)	8	
3	1978~1980	Intel 8086	16	<1
		Zilog 8000	16	
		Motorola 68000	16	
		Intel 8051(单片机)	8	
4	1981~今	iAPX 432	32	<0.125
		Intel 8096(单片机)	16	

出微处理器和单片机的发展概况。

本书的前六章介绍Z80微机的基本原理，后两章分别介绍单片机与可编程序控制器（简称PC）。PC是专门用于过程控制的微机，在现代化的发电厂中用得很多。通过本课程的学习，为读者今后在过程控制方面应用微机打下初步基础。

在学习中，除了注意理解基本概念和原理外，还应作好习题和实验。

第二节 数制与码制

数制是人们利用符号来计数的科学方法。在微机应用中常用二进制、十进制和十六进制。本节讨论常用进位制及带符号数的表示方法。

一、常用进位制及其相互关系

1. 常用进位制

在十进制中，用0、1、2、3、4、5、6、7、8、9等数码来表示数。一种进位制所用的数码个数称为它的进位基数，十进制的进位基数是十。

把各位数码乘以对应位的权，然后相加，即可得到数的值。十进制各位的权是以10为底的幂，整数从最低位起的权依次是 10^0 、 10^1 、 10^2 、 10^3 ……。例如

$$367 = 3 \times 10^2 + 6 \times 10^1 + 7 \times 10^0$$

由于十进制所用数码个数太多，难用电器件表示，故在计算机内部一律采用二进制。

二进制只用0和1两个数码，进位基数是二。我们只讨论整数，从最低位起，二进制的权依次是 2^0 、 2^1 、 2^2 、 2^3 ……。为了与十进制数相区别，在二进制数后面加写符号B。例如

$$110111\text{B} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 55$$

由于二进制数的位数太多，书写不便，在编写程序时，常用十六进制。

十六进制用0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F代表从零到十五的数。进位基数是十六，从最低位开始的权是 16^0 、 16^1 、 16^2 ……。以字尾H作为十六进制数的特征，例如

$$2\text{C5H} = 2 \times 16^2 + 12 \times 16^1 + 5 \times 16^0 \quad (\text{C为十二的数码}) = 709$$

2. 不同进位制间的相互关系

在上面介绍二进制和十六进制时，已说明了将二进制数和十六进制数转换为十进制数的方法。下面讨论将十进制数转换为二进制或十六进制数的方法，以及二进制和十六进制数的相互转换关系。

(1) 十进制数转换为二进制数

十进制整数转换为二进制数可用除二取余法。用2去除十进制数，记下商和余数，再用2去除上次的商，记下商和余数……，直到商等于零。将所得余数逐位（从低位起）排列，即为对应的二进制数。

【例 1-1】 将26转换为二进制数。

解：用除二取余法，列出计算表格，如表1-2。

表 1-2 除二取余法计算表

计算次序	被除数	除数	商	余数	排列顺序
0	26	2	13	0	↑
1	13	2	6	1	
2	6	2	3	0	
3	3	2	1	1	
4	1	2	0	1	

从低位开始，将余数排列得11010，即为26所转换成的二进制数。

(2) 十进制数转换为十六进制数

十进制整数转换为十六进制数可用除十六取余法。

【例 1-2】 将301转换为十六进制数。

解：计算过程见表1-3。

表 1-3 除十六取余法计算表

计算次序	被除数	除数	商	余数	排列顺序
0	301	16	18	D(即十三)	↑
1	18	16	1	2	
2	1	16	0	1	

12D为301所转换成的十六进制数。

(3) 二进制和十六进制数的相互转换

由于 $2^4 = 16^1$ ，4位二进制数恰与1位十六进制数相对应，其关系如表1-4所列。

表 1-4 十六进制与二进制数的对应关系

十六进制	0	1	2	3	4	5	6	7
二进制	0000	0001	0010	0011	0100	0101	0110	0111
十六进制	8	9	A	B	C	D	E	F
二进制	1000	1001	1010	1011	1100	1101	1110	1111

【例 1-3】 将100101101B转换为十六进制数。

解：所给二进制数共有9位，在高位补0，可分3组（每组有4位），再按表1-4转

换为3位十六进制数, 即

$$100101101B=0001\ 0010\ 1101\ B=12DH$$

【例 1-4】 将5FH转换为二进制数。

解: $5FH=0101\ 1111B=1011111B$

【例 1-5】 将255转换为二进制数。

解: 此题数字较大, 先转换为十六进制数, 再转换为二进制数较方便, 即

$$255=FFH\ (\text{用除十六取余法})=1111\ 1111B$$

二、计算机中数的表示方法

在计算机内部, 数据以二进制形式存放在寄存器中, 并通过运算器进行运算处理。寄存器和运算器都有固定的位数, 以8位微机为例, 它的数据寄存器和运算器都是8位的器件。计算机中的数都有一定的位数, 下面的讨论是针对8位微机进行的。

1. 无符号数

当用8位寄存器存放无符号数时, 最小的数是00000000, 最大的数是11111111, 表示数的范围是0~255, 共256个数。

如果在最大数11111111上再加1, 得到的和为100000000, 共9位, 低8位为00000000仍可保存在8位寄存器中, 进位1(其权为 $2^8=256$)则用另外的器件来保存。从寄存器本身看, 存在着加满(到256)归零的现象。

为便于讨论, 把满数(256)与一个数的差称为该数的补数。例如243是13的补数。

在二进制中, 求补数的运算可用求反加1法完成, 即

$$X' = \overline{X} + 1\ (X' \text{为} X \text{的补数})$$

证明: 设 $X = x_7x_6x_5x_4x_3x_2x_1x_0$,

则
$$X' = 100000000 - x_7x_6x_5x_4x_3x_2x_1x_0$$

注意
$$100000000 = 11111111 + 1$$

且
$$1 - x_i = \overline{x_i}$$

所以
$$X' = \overline{x_7x_6x_5x_4x_3x_2x_1x_0} + 1$$

【例 1-6】 求00001101(13)的补数。

解: $X' = \overline{00001101} + 1 = 11110011\ (243)$

2. 带符号数

对带符号数, 规定以最高位表示正负号, 正数的最高位为0, 负数的最高位为1。

当用8位寄存器存放正数时, 最大正数为01111111(即+127)。负数有多种表示方法, 下面介绍两种。

(1) 原码

将最高位作为符号位, 其余位表示数的绝对值, 这种方法称为原码表示法。例如

$$00001101 \text{表示} +1101$$

$$10001101 \text{表示} -1101$$

用8位原码表示数时, 最大正数是01111111, 最小负数是11111111, 数的范围是+127~-127。

原码表示法很直观，易于理解。但在微机中采用的却是补码表示法，这是因为补码能将减法化为加法，从而简化了运算器的结构。

(2) 补码

用补码表示数时，正数和原码表示一样，区别在于负数。负数用绝对值的补数来表示，称为补码表示法。

补码表示法能够将减法运算化为加法。例如 $13-5$ ，先化作 $13+(-5)$ ，再将 (-5) 用绝对值 (5) 的补数表示，该补数为 251 ，将 $13+251$ 得和为 264 。 264 的 2^8 部分作为进位从 8 位寄存器中消失了，在寄存器中只剩下 8 ，恰是 $13-5$ 的结果。

这很象校正钟表的操作，钟表的满数是 12 ，到 12 就归 0 （ 12 好象进位）。如果现在是 6 点，有一个钟表却指在 8 点上，这时有两种拨针方法

$$8 - 2 = 6 \text{ (倒拨)}$$

$$8 + 10 = 6 \text{ (+12) (顺拨)}$$

在顺拨时，减 2 的运算用加 2 的补数 10 代替了，由于 12 是满数，它自动从钟表上消失了，剩下的数 6 恰是正确的时间。

【例 1-7】 用 8 位补码计算 $13-5$ 。

解： $13 = 1101\text{B} = 00001101\text{B}$

$5 = 101\text{B} = 00000101\text{B}$

$-5 = -101\text{B}$

-5 的补码表示为 $\overline{0000\ 0101} + 1 = 1111\ 1011$

$$13 - 5 = 0000\ 1101\text{B} + 1111\ 1011\text{B} = 1\ 0000\ 1000\text{B}$$

在 8 位寄存器中保留的是 $0000\ 1000$ ，即 8 。

用 8 位补码表示数时， $00000000 \sim 01111111$ 代表正数（最高位 0 代表正号）， $11111111 \sim 10000000$ 代表负数（最高位为 1 ）。因 11111111 是 1 的补数，所以 (-1) 的补码表示是 11111111 ，同理 10000000 表示 -128 。 8 位补码所表示的数的范围是 $-128 \sim +127$ 。

用补码表示负数时，其绝对值要用求反加 1 法求出。

【例 1-8】 用 8 位补码计算 $5-13$ 。

解：用补码表示时

$$5 = 0000\ 0101\text{B}$$

$$(-13) = \overline{0000\ 1101} + 1 = 1111\ 0011\text{B}$$

相加结果为 $1111\ 1000\text{B}$ （负数），其绝对值为

$$\overline{1111\ 1000} + 1 = 0000\ 1000$$

所以结果为 -8 。

三、数和字符的编码

1. BCD码

计算机中普遍采用二进制进行计算，但人们却习惯于用十进制数，为了便于转换，用二进制数来对十进制数编码，这就是BCD码。

由于 4 位二进制数可表示 $0000 \sim 1111$ 共十六种状态，而十进制数只有 $0 \sim 9$ 共十种数

码，所以用4位二进制数对十进制数编码时，多余六种状态。通常将1010~1111六种状态舍弃，用0000~1001依次作为0~9的编码。表1-5列出了这种编码关系。

表 1-5 BCD 编 码 表

十进制数	0	1	2	3	4	5	6	7	8	9
BCD码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

【例 1-9】 写出75的BCD码。

解：顺序写出7和5的BCD码，即得75的BCD码为

0111 0101

应该注意的是：只有1位十进制数的BCD码和二进制数相同。若将75转换为二进制数，结果是1001011B，而不是01110101。

BCD码用于一些输入和输出设备以及数字式仪表，多数微机都有BCD码计算电路，能对用BCD码表示的十进制数进行运算。

2. ASCII码

计算机除了处理数据外，还要处理字符，因而必须对字符编码。国际上通用的ASCII

表 1-6 ASCII (美国标准信息交换码) 表

b ₇ b ₆ b ₅ b ₄ (低位)	b ₃ b ₂ b ₁ b ₀ (高位)							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	'	3	C	S	c	s
0100	EOI	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	+	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	Ω	n	~
1111	SI	US	/	?	O	--	o	EEL

码见表1-6所列。

列表时以低4位分行，高3位分列。例如A对应的高位是100，低位是0001，所以它的编码是1000001，用十六进制表示为41H。

ASCII码共有128个字符，每个字符都用7位编码。为了检验传送有无错误，常补充一位 b_7 作为校验位。可以采用奇校验或偶校验，在偶校验中，用 b_7 将8位码中所含1的个数配成偶数。例如A的7位编码中含有2个1，所配 b_7 为0，C的7位编码为1000011，应配 b_7 为1，总共有4个1。如传递中有一位出错，则含1的个数成为奇数，可被发现。

第三节 微机组成与工作原理

一、预备知识

在介绍微机组成之前，先将有关的电子技术知识复习一下。

1. 基本的门电路

与门、或门、非门是三种基本的门电路，其逻辑符号见图1-1。

与门内的&为and的简写，其逻辑关系为：当输入 X_1 与 X_2 均为1时，输出Y为1，否则Y为0。

或门（符号 ≥ 1 ）的逻辑关系为：当输入中有一个或多个1时，输出Y为1，否则Y为0。

非门标有小圆圈，其输出与输入相反。

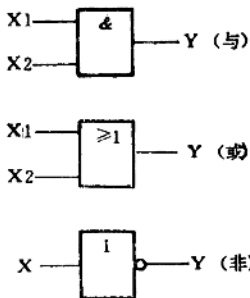


图 1-1 基本门电路

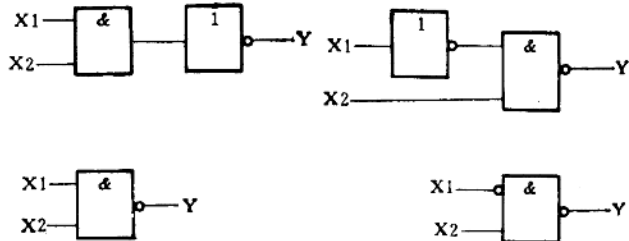


图 1-2 与非门

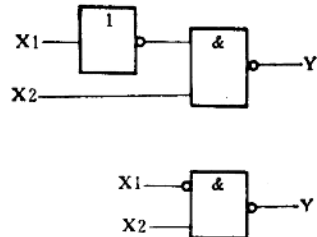


图 1-3 输入反相的与非门

2. 与门和非门的组合

将与门的输出或输入经非门反相，可得不同的逻辑关系。

将与门的输出反相，可得与非门，其逻辑关系是：当输入 X_1 与 X_2 均为1时，输出Y为0，否则Y为1。与非门的逻辑符号见图1-2。

也可将与非门的输入反相，图1-3中将与非门的输入 X_1 反相（用小圆圈表示），其逻辑关系为：当 $X_1 = 0$ 与 $X_2 = 1$ 时， $Y = 0$ ，否则 $Y = 1$ 。

输入和输出经反相连接的与门，其基本逻辑关系仍是与关系，在有些情况下，仍简称为与门。

3. 三态门 (三态缓冲器)

三态门有三种输出状态, 除了 0 和 1 外, 还有高阻状态, 其逻辑符号见图 1-4。X 为输入, Y 为输出, OE 为输出允许控制端。当 OE = 1 时, 三态门接通, Y = X, 输出为 0 或 1 视 X 而定。当 OE = 0 时, Y 为高阻悬浮状态。三态门相当于电子开关。

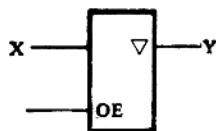


图 1-4 三态门

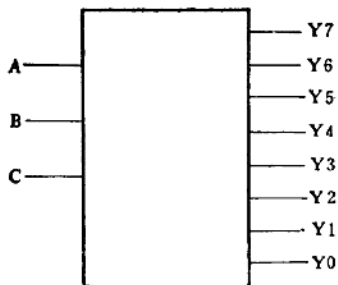


图 1-5 3:8译码器

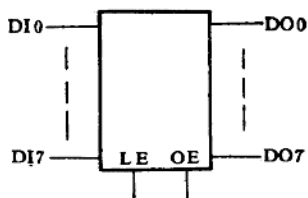


图 1-6 8位寄存器

4. 译码器

译码器能将地址、指令等二进制信息译码。当输入端为 n 位二进制信息时, 则输入状态有 2^n 种。译码器有 2^n 条输出线, 与输入状态对应。由哪一条输出线输出的电平有效, 可知输入是什么状态。

图 1-5 画出一个 3:8 译码器。它有三个输入端, 输入 3 位二进制信息。有八个输出端, 区分八种输出状态。

表 1-7 列出译码器的逻辑关系, 称为真值表。对应一种输入状态, 只有与之相应的一条输出线输出有效电平 (本译码器以 1 为有效)。例如输入 ABC = 001 时, 只有 Y_1 输出为 1。

表 1-7 译码器的真值表

输入			输出							
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

5. 寄存器

寄存器是由触发器组成的, 用来存储二进制信息。

图 1-6 表示一个 8 位寄存器。DI0 ~ DI7 为 8 位数据输入引线, 受信号 LE (装入允许) 控制。当 LE 有效时, 存入 8 位数据。DO0 ~ DO7 为 8 位数据输出引线, 在输出引线上串

有三态门（图中未画），受信号OE（输出允许）控制。当OE有效时，输出所存的8位数据。当OE无效时，输出为高阻状态。

二、微机的基本组成

微机的基本组成如图1-7所示。

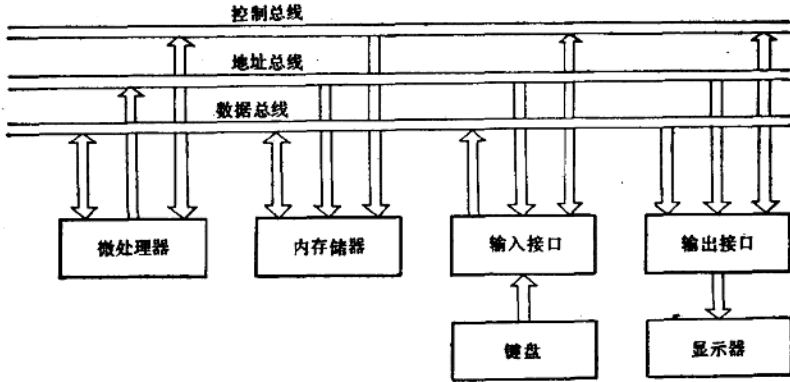


图 1-7 微机基本组成

1. 总线

总线用来传送信息，例如将存储器中的原始数据送到微处理器进行计算，将计算结果送回存储器，都要通过总线。

根据所传送的信息种类，将总线分为三种，即数据总线、地址总线和控制总线。在8位微机中，有8条数据线，16条地址线，还有若干控制信号线。

地址总线将16位地址码传送到存储器，可以选择一个存储单元进行存数或取数操作，具体操作过程将在后面介绍。

控制总线在微处理器和存储器以及外部设备接口之间传送控制信息。例如向存储器存数，要发写控制信息，从存储器取数，要发读控制信息。

向总线输出信息的部件都必须有三态门控制。图1-8表示四个部件和总线的连接情况，A和D是发送部件，其数据经三态门接到总线，B和C为接收部件，在控制信号LE有效时，能从总线接收数据。要将发送部件A的数据传送给接收部件C，应使部件A的三态门接通，使部件A的数据送到总线，再使接收部件C的控制信号LE有效，使C由总线接收数据。

应该注意，当一个发送部件输出数据时，其它发送部件必须处在高阻悬浮状态，不然会使信号混乱。这就是发送部件要经三态门控制的原因。

微处理器、存储器、输入输出接口等部件，只要是向总线发送信息的，都经三态门控制。

2. 内部存储器

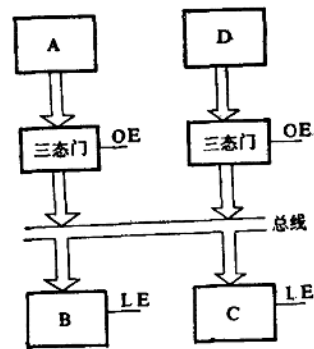


图 1-8 部件与总线的连接

存储器是计算机的重要组成部分，用来存放程序和数据。存储器有内部存储器（内存）和外部存储器（外存）之分，微机以半导体存储器作为内存，以磁盘或磁带机作为外存。内存是主存储器，外存是辅助存储器，外存所存信息必须先调入内存才能为微处理器所应用。从容量和速度来看，内存容量小，但存取速度快，外存容量大，但存取速度慢。一般应用时，只把要用的程序和数据放在内存，暂时不用的放到外存。

内存又分ROM（只读存储器）和RAM（读写存储器）两种类型。ROM信息固定不变，掉电也不会丢失，在计算机中，常用来存放固定的程序，如监控程序、汇编程序等，还用来存放各种表格和常数。RAM能根据需要读出或写入信息，半导体RAM是易失性的存储器，即掉电后所存信息也随之消失。因此，它用来暂时存放输入输出的数据、中间计算结果或断电后不影响大局的程序。

磁盘或磁带也是非易失性的存储器，为了防止RAM在断电时丢失信息，可将信息转存到磁盘或磁带。

半导体RAM又分静态和动态两种。静态RAM用触发器存储0和1，在断电的情况下，其状态可长期维持不变。动态RAM用电容器的充电和放电来表示1和0，由于泄漏电流，只能短时维持信息，使用时必须再生。所谓再生，是指在信息消失前进行重写，其周期一般不超过2ms。动态RAM常用作大容量存储器（以后，当不特别指明时，存储器均指内部存储器）。

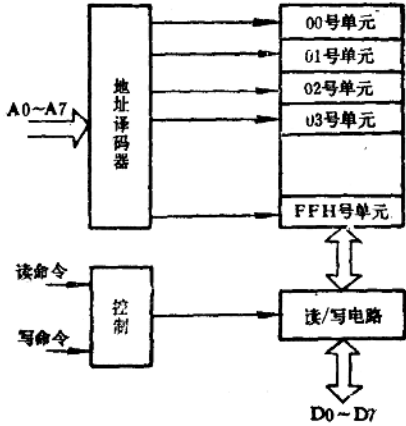


图 1-9 RAM的结构

存储器由大量的存储单元组成，每个单元都有相同的位数（微机多为8位）。为了相互区别，每个单元都有相应的地址编号，称为它的地址。

微处理器从存储器取数也称为读数，向存储器存数也称为写数。无论读或写，都是对其中一个单元进行的，都必须向存储器发送与单元相应的地址码。还要发送控制信号来区分是读还是写。

图 1-9 表示一个简单存储器的结构。它由存储体、地址译码器和读写控制电路组成。

图中，存储体由 256 个存储单元组成，每个单元都是 8 位。一般，称 8 位为一个字节，也可以说，每个单元都是 1 字节。

地址译码器根据 A0~A7 引线来的 8 位地址码选择相应的存储单元。例如要从 03 号单元取数，微处理器必须送来 00000011 的地址码。

读写控制电路用来控制对存储单元的读写操作。读出时，微处理器发来读控制信号（或称读命令），存储单元中的数据由 D0~D7 送到微处理器，D0~D7 为数据引线。写入时，微处理器发来写控制信号（或称写命令），通过 D0~D7 向存储单元存入数据。在无读写控制信号时，D0~D7 为高阻状态（即前面所说的高阻悬浮状态）。

图 1-10 表示微处理器（用 CPU 表示）读写存储器的过程。

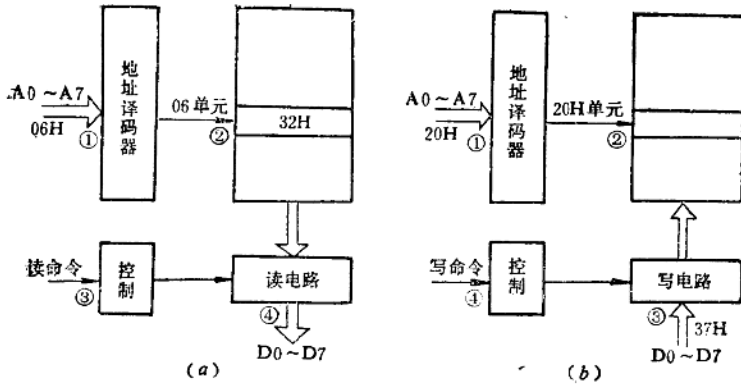


图 1-10 CPU 读写存储器的过程
(a)读; (b)写

读数的操作步骤是:

- ①发送地址码 例如要从06号单元读数, CPU应送出的地址码为00000110, 此码在图中写成06H, 经地址总线送到存储器。
- ②地址译码 译码后, 选中与地址码相应的单元, 图中为06号单元。
- ③发送读命令 通过控制总线, CPU向存储器发送读控制信号。
- ④读数 06号单元中的数据(图中为32H)送到数据总线, 为CPU读取。

图中, 对写数的操作也用顺序号①~④表示, 微处理器要将数据(图中为37H)写入存储单元(图中为20H号单元)时, 在发送写控制信号前, 应将数据送到数据总线上。写控制信号(写命令)一来, 即可将数据存入。写数操作后, 原来所存的数据消失, 代之以新数据。

3. 微处理器

微处理器即微型的中央处理器, 其功能是执行程序。它由运算器、控制器和寄存器等部分组成, 通常作在一块集成电路上, 各部分用内部总线连接。内部总线在芯片内, 为了区别, 也将连接微处理器和存储器以及输入输出接口之间的总线(即前述数据总线、地址总线、控制总线)称为系统总线。图1-11表示微处理器的基本结构。

(1) 运算器

运算器由算术逻辑单元ALU、累加器A和暂存器TMP组成, ALU可进行各种算术运算和逻辑运算, 运算数据来自A和TMP, 运算结果存入A。例如, 加法运算的步骤是:

- ①数据送入ALU 控制器向累加器A和暂存器TMP发送控制信号, 使其向ALU送数。
- ②加法操作 控制器向ALU发送加法操作控制信号, 使之作加法运算, 输出运算结果。
- ③存结果 控制器向A发送控制信号, 将运算结果存入。累加器中原存数据消失。

图1-12表示加法运算过程。

(2) 控制器

控制器由指令寄存器、指令译码器和定时控制电路等部分组成。其功能是根据指令的