



水泥中技电视录像教材

微机在水泥生产中的应用

孙光澍 编

湖南省建筑材料工业局

70128
MJC

水泥中技电视录像教材（试用本）

微机在水泥生产中的应用

（内部发行）

湖南省建材工业局组织编写

望城县丰华印刷厂承印

开本：16开 印张：15.1 字数：349.5千字

国家建材局教材办公室

(87) 材人开材便字第101号

各省、区、市建材局(公司)或建材主管部门:

为适应水泥行业岗位培训的需要,湖南省建材工业局组织编写和摄制一套培训中级技术工人的电视教材和电视录像教学片作为水泥企业的看火工、粉磨工、机修工、电工、化验工和生产管理干部岗位培训专业技术知识的规范标准。各级建材主管部门可因时、因地制宜、组织教学和考核。

此套教材和教学片定于今年十一月陆续出版发行,你局如需此教材和教学片,请直接与湖南省建材工业局人事处联系。

地址:湖南省长沙市人民路164号

国家建材工业局教材办公室(印)

一九九〇年元月十日

湖南省建筑材料工业局文件

(89)湘建材人字第057号



关于继续发行《水泥

中技电视教材和录像教学片》的通知

各地(州)市建材工业局(公司)、各建材企、事业单位:

为适应水泥行业岗位培训的需要,国家建材工业局教材办公室(87)材人开材便字第101号函委托,我局组织编写和摄制一套培训中级技术工人的电视教材和录像教学片,作为水泥企业看火工、粉磨工、机修工、电工、化验工、生产管理干部岗位培训专业技术知识的规范标准和劳动工资部门“考工定级”的培训资料。

此套教材和教学片包括《水泥生产基本知识》、《煅烧工艺与设备》、《粉磨工艺与设备》、《机械安装与维修》、《电工基础》、《水泥生产检验与控制》、《微机在水泥生产中的应用》和《环境保护与除尘》等。今年四月,我局商请国家建材局人才开发司同意,对教材、录像带进行了进一步修改和审定,并确定继续由湖南省建材干部中专学校发行。各建材企事业单位可按岗位培训为主、业余自学为主的原则,组织教材订购和及时进行教学考核。

一九八九年十二月二十一日

抄报:国家建材工业局、省经委、省教委。

抄送:各省、区、市建材局(公司)或建材主管部门、湖南省劳动厅、省成人教育局。

前 言

微型计算机虽然历史不长，但其发展速度却十分惊人，应用范围愈来愈广。近几年来，水泥生产控制中也在逐步推广微机应用，并取得显著的成效。如生料配料的质量控制，磨机负荷控制，预加水成球自动控制，机电设备的程序控制，过程参数的巡回检测，立窑闭门煅烧等。在微机应用推广过程中，最主要的问题是如何培养一批具有微机初步知识的操作人员和维修人员，如何使有关管理人员和技术人员具有相应的微机应用知识。

目前，大多数中小型水泥企业缺乏这类人材，因而不敢引用微机这种现代技术，或者虽然引用了微机，而又对付不了调试、维修问题。鉴于这种需要，我们编写了本书，供初中以上文化程度的有关人员学习。

本书选用水泥行业较流行的Z80微机做范例，作为学习微机的入门。在内容编排方面，力求通俗易懂，简明扼要，由浅入深，强调应用。对于几乎不具备电工、电子理论的人员，只要经过短期学习，就能达到初步应用微机的水平。

计算机科学是一门实验科学，“纸上谈兵”并没有实际意义，只有通过边学边做（实验），才能真正掌握微机的基本知识和应用技术。有条件的地方，希望能配备一台Z80单板机，提供给学员实验，这将会起到事半功倍之效果。

本书由湖南省建材技校孙光澍编写，由湖南大学电子工程系谭扬林副教授审稿。

由于编者水平有限，书中错误和不妥之处，在所难免，殷切希望读者给予批评指正。

编 者

1988年7月

目 录

1	微机中的数字逻辑器件.....	(1)
1.1	微机的算术基础.....	(1)
1.1.1	数的进位制及其转换.....	(1)
1.1.2	二进制数的运算.....	(3)
1.1.3	二进制编码.....	(4)
1.2	逻辑电路实验辅助部件.....	(8)
1.2.1	必要的工具和仪器.....	(8)
1.2.2	稳压电源.....	(9)
1.2.3	编码开关部件的制作.....	(10)
1.3	门电路.....	(10)
1.3.1	反相器.....	(10)
1.3.2	与门和或门.....	(12)
1.3.3	三态门.....	(16)
1.4	触发器电路.....	(17)
1.4.1	触发器的工作原理.....	(17)
1.4.2	触发器的类型.....	(20)
1.4.3	触发器的应用.....	(25)
1.5	组合逻辑电路.....	(29)
1.5.1	二进制译码器.....	(29)
1.5.2	二——十进制七段显示译码器.....	(31)
1.5.3	编码器.....	(34)
1.5.4	数据选择器.....	(36)
1.6	存贮器.....	(37)
1.6.1	存贮器的功能.....	(37)
1.6.2	读写存贮器RAM.....	(37)
1.6.3	只读存贮器ROM.....	(38)
1.6.4	一个实际的RAM例子.....	(40)
2	微型计算机与微处理器.....	(44)
2.1	微型计算机的基本结构.....	(44)
2.1.1	微型计算机的发展概况.....	(44)
2.1.2	电子计算机的基本结构.....	(45)
2.1.3	微型计算机的典型结构.....	(46)
2.2	Z80 微处理器.....	(48)

2 · 2 · 1	Z80 微处理器的内部结构	(48)
2 · 2 · 2	Z80 微处理器引脚及其功能	(50)
3	Z80指令系统与汇编语言	(54)
3 · 1	Z80的寻址方式	(54)
3 · 2	Z80指令系统	(57)
3 · 2 · 1	数据传送指令	(58)
3 · 2 · 2	数据块传送和搜索指令	(62)
3 · 2 · 3	算术和逻辑指令	(64)
3 · 2 · 4	循环与位移指令	(66)
3 · 2 · 5	位操作指令	(67)
3 · 2 · 6	转移、调用与返回指令	(67)
3 · 2 · 7	输入与输出指令	(70)
3 · 2 · 8	CPU控制指令	(71)
3 · 3	汇编语言	(71)
3 · 3 · 1	机器语言与汇编语言	(71)
3 · 3 · 2	Z80汇编语言结构	(73)
3 · 3 · 3	伪指令	(75)
3 · 3 · 4	宏指令	(76)
3 · 4	汇编语言程序设计	(77)
3 · 4 · 1	用汇编语言编程的基本过程	(77)
3 · 4 · 2	简单程序	(78)
3 · 4 · 3	分支程序	(79)
3 · 4 · 4	循环程序	(81)
3 · 4 · 5	子程序	(83)
4	输入输出方式与接口芯片	(88)
4 · 1	输入输出方式	(88)
4 · 1 · 1	Z80输入输出操作	(88)
4 · 1 · 2	输入输出方式	(89)
4 · 2	Z80中断系统	(94)
4 · 2 · 1	Z80中断概述	(94)
4 · 2 · 2	Z80中断方式	(96)
4 · 2 · 3	中断优先权	(101)
4 · 3	可编程并行接口芯片PIO	(103)
4 · 3 · 1	PIO的主要功能和内部结构	(103)
4 · 3 · 2	PIO的硬件连接	(105)
4 · 3 · 3	PIO的程序编制	(108)
4 · 3 · 4	PIO应用举例	(109)
4 · 4	计数/定时芯片CTC	(112)

4 · 4 · 1	CTC结构	(113)
4 · 4 · 2	CTC的硬件连接	(114)
4 · 4 · 3	CTC的工作方式与编程	(116)
4 · 4 · 4	CTC应用举例	(120)
5	微机控制技术	(124)
5 · 1	自动控制简介	(124)
5 · 1 · 1	自动控制系统	(124)
5 · 1 · 2	测量变送元件	(127)
5 · 1 · 3	执行机构	(131)
5 · 1 · 4	工业控制中的抗干扰	(133)
5 · 2	数/模与模/数转换器	(135)
5 · 2 · 1	D/A转换器	(135)
5 · 2 · 2	A/D转换器	(139)
5 · 3	数据采集与数据处理	(142)
5 · 3 · 1	数据采集系统	(142)
5 · 3 · 2	模拟量的输出	(145)
5 · 3 · 3	数字滤波	(146)
5 · 3 · 4	越限报警处理	(148)
5 · 3 · 5	数字PID处理	(150)
5 · 4	顺序控制系统实例	(153)
5 · 5	过程控制系统实例	(165)
6	微机在水泥生产中的应用	(174)
6 · 1	微机配料控制系统	(174)
6 · 1 · 1	微机配料控制系统的组成	(174)
6 · 1 · 2	微机控制原理	(178)
6 · 1 · 3	KLD—1型微机配料控制系统的特点	(185)
6 · 2	预加水成球自动控制	(188)
6 · 2 · 1	预加水成球系统的工艺特点	(188)
6 · 2 · 2	预加水成球自控系统	(189)
6 · 2 · 3	预加水成球微机控制原理	(190)
6 · 2 · 4	推广中应注意的问题	(192)
6 · 3	水泥磨负荷控制	(192)
6 · 3 · 1	系统简介	(192)
6 · 3 · 2	硬件配置	(193)
6 · 3 · 3	软件编制	(194)
6 · 4	用微机控制袋式收尘器清灰	(197)
6 · 4 · 1	微机控制系统	(197)
6 · 4 · 2	外设接口电路	(198)

6 · 4 · 3	系统软件编制	(199)
6 · 5	水泥工业中微机应用的现状与前景	(200)
6 · 5 · 1	微机应用的现状	(200)
6 · 5 · 2	微机应用展望	(202)
附录一	TP801 单板机的操作说明	(204)
附录二	Z80 指令系统的操作说明	(221)
附录三	Z80 指令名称与操作码对照表	(229)

1 微机中的数字逻辑器件

1.1 微机的算术基础

1.1.1 数的进位制及其转换

(1) 二进制

长期来,人们习惯于使用十进制数,在十进制中,每一位数最多有0~9十个基本数码,其进位法则是“逢十进一”,“10”便是十进制记数法的“基数”。

众所周知,一个十进制数如245.36可以写成如下形式:

$$245.36 = 200 + 40 + 5 + 0.3 + 0.06$$

$$= 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$$

所有的十进制数都可以写成以上类似形式,每一数位的数值等于一个数码乘上一个因子,这个因子就称为权, 10^2 、 10^1 、 10^0 、 10^{-1} 、 10^{-2} 分别为各位的权。十进制数是由数码的值和位权来表达的。

十进制虽然方便,然而电子计算机却不能直接利用它去计算,为什么呢?这是因为电器元件通常是处于两个状态之一,即通和断,高电位和低电位,如果把通看成“1”,那末,断就是“0”,要找出具有十个状态的电器元件是困难的,因此,计算机通常用二进制来进行计算。

二进制数中只有两个数码即“1”和“0”,它的进位法则是“逢二进一”,“2”是二进制数的基数,一个二进制数,例如:

$$1010.1111$$

可以表示为

$$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4}) \\ = 8 + 0 + 2 + 0 + 0.5 + 0.25 + 0.125 + 0.0625 = 10.9375$$

那么

$$(1010.1111)_2 = (10.9375)_{10}$$

由上式看出, 2^3 、 2^2 、 2^1 、 2^0 、 2^{-1} 、 2^{-2} 、 2^{-3} 、 2^{-4} 分别是二进制数各位的权。

(2) 十六进制

用二进制表示数,位数多,读写起来比较麻烦。为了弥补这个缺点,在计算机中还采用十六进制数制。

所谓十六进制,就是用十六个不同的数字作基本数码,按“逢十六进一”的计数方法表示数,这十六个数中,“0~9”的书写同十进制,其它六个数码用如下符号表示:

表 1 · 1

十 进 制	十 六 进 制	十 进 制	十 六 进 制
10	A	13	D
11	B	14	E
12	C	15	F

一个十六进制数也可以按权位展开成多项式形式，例如

$$7B3D = 7 \times 16^3 + B \times 16^2 + 3 \times 16^1 + D \times 16^0 = 7 \times 4,096 + 11 \times 256 + 3 \times 16 + 13 \times 1 = 31,549$$

以后为了辨别方便，我们在数字下角加一字母来区分不同进制数，加B的为二进制数，加H表示十六进制数，加D表示十进制数。如(64)_H、(100)_D、(1100100)_B分别表示十六进制数，十进制数和二进制数。

(3) 二进制数和十六进制数之间的转换

由于 $2^4 = 16$ ，一个十六进制数相当于四位二进制数字，0~F的十六进制数可以用表1·2所列的二进制数表示：

表 1 · 2

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

根据这种对应关系，二进制与十六进制之间的转换十分简单。

一长串二进制数可以按四位一组顺序分划，整数部分从右到左，小数部分从左向右分组，末组不足四位可以补0，再按上表对应关系转换成十六进制数。

例 $(1101101110.011)_B = (0011 \ 0110 \ 1110 \cdot 0110)_B = (36E.6)_H$

反之，一个十六进制数可按以上对应关系转换成二进制数：

例 $(B47.0A)_H = (1011 \ 0100 \ 0111 \cdot 0000 \ 1010)_B$

(4) 二进制数和十进制数之间的转换

二进制数转换为十进制数通常采用按权位展开或多项式的方法，这一方法前面已叙述，这里不再介绍。

十进制数转换为二进制数通常采用除2取余法（小数部份是乘2取整法，本书不加

介绍)，这种方法是用2不断地去除被转换的十进制数，取每次除后的余数，组成二进制的各位。

例：(69)_D = (?)_B

2	69
2	34
2	17
2	8
2	4
2	2
2	1
	0

余数	
1	↑ 低位
0	
1	
0	
0	
0	
1	↑ 高位

所得结果为100 0101。
即：(69)_D = (1000101)_B

(5) 十六进制数和十进制数的转换

十六进制数转换为十进制数也是按权位展开，然后相加即可。

十进制数转换为十六进制数类似于转换为二进制数，不同之处只是将2换成16，采用除16取余法。

例：(1192)_D = (?)_H

16	1192
16	74
16	4
	0

余数	
8	↑ 低位
A	
4	↑ 高位

结果为4A8
即：(1192)_D = (4A8)_H

1.1.2 二进制数的运算

二进制数也可以进行加减乘除四则运算，只不过在二进制中，只有0、1两个数码，每个数位非0即1，以2为基数，相邻高位的位置是低位的两倍。现分述如下：

(1) 加法：0+0=0；0+1=1；1+0=1；1+1=10

例 1001.01+0101.01=1110.10

1001.01
+)0101.01
1110.10

(2) 减法：0-0=0；1-0=1；1-1=0；10-1=1

例 $1011.11 - 0101.01 = 0110.10$

$$\begin{array}{r} 1011.11 \\ - 0101.01 \\ \hline 0110.10 \end{array}$$

(3) 乘法: $0 \times 0 = 0; 0 \times 1 = 0; 1 \times 0 = 0; 1 \times 1 = 1$

例 $1011 \times 0101 = 110111$

$$\begin{array}{r} 1011 \\ \times 0101 \\ \hline 1011 \\ 0000 \\ + 1011 \\ \hline 110111 \end{array}$$

可见二进制乘法能归结为移位和相加。

(4) 除法: 其商数位非 0 即 1。

例 $11010011 \div 1011 = 10011 \dots \dots \text{余} 10$

$$\begin{array}{r} 10011 \\ 1011 \overline{) 11010011} \\ \underline{-) 1011} \\ 10001 \\ \underline{-) 1011} \\ 1101 \\ \underline{-) 1011} \\ 10 \dots \dots \text{余数} \end{array}$$

表 1·3

可见二进制除法能归结为移位和减法。

1·1·3 二进制编码

计算机不仅要处理二进制数,还要处理诸如 A、B、C、+、-、! 等符号,还要处理负数的运算,要较方便地处理十进制数,为此,我们采用一些二进制编码来解决这些问题。

(1) BCD 码

计算机虽然能方便地处理二进制数,然而人们却对它不习惯,二进制不便于书写、

阅读、辨认,它与十进数的转换也比较麻烦,因此,人们希望一种与十进制转换比较方便又能为计算机接受的数制,这就是目前普遍采用的 BCD 编码,又称二——十进制数。

这种记数编码方式是将每个十进制数码用四个二进制代码表示。四位二进制数有十六种状态,只要去掉其中六个状态,就可以与十进制数码的十个状态一一对应。如表 1·3

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

所示，表中BCD码是其中常用的8421码。

这样，一个十进制数用BCD码表示就比用二进制表示要简单得多。

例 $(97)_D = (10010111)_{BCD}$

反之，也非常简单

例 $(10010010)_{BCD} = (1001 \quad 0010)_{BCD} = (92)_D$

(2) ASCII码

在微机中还经常用到一种编码叫ASCII码，它是英语“美国标准信息交换代码”的缩写，这种编码是用7位二进制码来表示，因为7位二进制码共有128种状态，所以它不仅能表示0~9十个数字，还能表示26个大、小写英文字母，还有运算符号，标点符号和其它特殊符号，如表1•4所示，有些ASCII码，如ASCII码0~32所表示的不是数符，而是计算机操作。从表中，可以很方便地查出某一数符所对应的ASCII码和二进制编码。

例	字符	ASCII码	二进制编码
	A	41H	0100 0001
	※	2AH	0010 1010
	O	30H	0010 0001

表1•4 ASCII（美国标准信息交换码）表

列		0 (3)	1 (3)	2 (3)	3	4	5	6	7 (3)
行	位 ↓ 654→ 3210	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	,	p
1	0001	SOH	DC ₁	!	1	A	Q	a	q
2	0010	STX	DC ₂	"	2	B	R	b	r
3	0011	ETX	DC ₃	#	3	C	S	c	s
4	0100	EOT	DC ₄	\$	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	Ω(1)	n	~
F	1111	SI	US	/	?	O	—(2)	o	DEL

(3) 原码、反码和补码

数学中处理正数和负数问题时，是将数的绝对值前边加上一个符号，“+”表示正数，“-”表示负数。而计算机须将这个符号数字化，便于运算时识别。通常是在数的前面加一位作符号位，0表示正数，1表示负数。如+1101101和-1101101可分别写成：

符号位	数位
0	1101101
1	1101101

正负数可以按规定的方法来表示。可是，被表示的数对于某些计算并不方便、例如，两个正数相减

$$(00000001)_B - (00000010)_B = ?$$

在笔算时总是要把两者颠倒过来相减，再把得到的差冠以负号：

负号

$$-[(00000010)_B - (00000001)_B] = 10000001$$

用计算机进行这种计算，就必须先比较两个数绝对值的大小，然后决定是颠倒过来相减还是直接相减，是否要在结果前面冠以负号等等。这种方法既繁琐又费时间。因此要设法改进一下编码制度而使运算的过程简单些，补码、反码就是为此目的提出来的。

原码：在数值前加一位符号位，用0表示正数，1表示负数。如：

表1·5

十进制数	符号位	数值	十进制数	符号位	数值
+7	0	0000111	+21	0	0010101
-7	1	0000111	-21	1	0010101

计算机采用原码进行运算，还是很困难的，至少可以看出同一绝对值的正负两数相加不为零。

反码：在反码表示法中，正数的反码与原码相同，负数的反码是数值部份按位取反（即0变1，1变0），符号位为1。如：

表1·6

十进制数	符号位	数值	十进制数	符号位	数值
+7	0	0000111	+21	0	0010101
-7	1	1111000	-21	1	1101010

由原码变为反码是相当方便的。

补码：正数的补码与原码相同，负数的补码是取其反码，然后在最后一位加1。

如：

$[-7]_{\text{原}} = 10000111$

$[-7]_{\text{反}} = 01111000$

$[-7]_{\text{补}} = 01111000 + 1 = 01111001$

下面给出一组8位数的原码、反码和补码。

表 1 · 7

十进制字	原 码 表 示	反 码 表 示	补 码 表 示
+127	0111,1111	0111,1111	0111,1111
+126	0111,1110	0111,1110	0111,1110
⋮	⋮	⋮	⋮
+2	0000,0010	0000,0010	0000,0010
+1	0000,0001	0000,0001	0000,0001
+0	0000,0000	0000,0000	0000,0000
-0	1000,0000	0111,1111	1000,0000
-1	1000,0001	0111,1110	0111,1111
-2	1000,0010	0111,1101	0111,1110
⋮	⋮	⋮	⋮
-126	1111,1110	0000,0001	0000,0010
-127	1111,1111	0000,0000	0000,0001
-128			1000,0000

引入补码的好处是可以将减法化成加法运算，只不过将减数变为补码形式。如：

$(2)_{\text{D}} - (1)_{\text{D}} = (2)_{\text{D}} + (-1)_{\text{D}} = (0000,0010)_{\text{B}} + (1111,1111)_{\text{补}}$

$$\begin{array}{r} 0000,0010 \\ + 1111,1111 \\ \hline 10000,0001 \\ \text{丢失} \end{array}$$

则上式 = 0000,0001

又如：

$(1)_{\text{D}} - (2)_{\text{D}}$

因为 -2 的补码为 1111,1110，所以 1 - 2 为：

$$\begin{array}{r} 0000,0001 \\ + 1111,1110 \\ \hline 1111,1110 \end{array}$$

其结果符号位为1，数值部份再求补(1111,1111)原 = (1000,0000)反 = (1000,0001)补

所以上式 = 1000,0001，即 -1

1.2 逻辑电路实验辅助部件

1.2.1 必要的工具和仪器

学习微机的捷径就是自己动手实验，有时虽然还不懂其中的奥秘，但是你实验作成功了，也就懂得一些应用了。本章的内容主要是通过实验让你形象地理解逻辑电路的原理和功能，为此需要准备些简单的工具、测试仪器和逻辑器件。

(1) 必要的工具

首先是电烙铁，选择烙铁头较细的，耗电量约20W的较好，焊接时不能使元件受热过长，也不能产生虚焊，虚焊往往是机器发生故障的原因。

其次要准备偏口钳、尖嘴钳、镊子、十字型螺丝刀和普通螺丝刀，在拆卸集成电路（简称片子）时，要很小心地使用小型普通螺丝刀。见图1.1。

要把各种实验元件组合起来还必须有一块实验板，这里介绍一种通用电路实验板。俗称面包板，其特点是不用焊接，装配容易，适用于实验，在实验板中，有一部份是五孔内部连接，另一部份是五组或七组内部连接。

(2) 简单测试仪器

要准备好一块万用表，能测电阻、电压和电流，为了检验发光二极管引脚的极性，万用表的电源电压最好在3伏以上。

有条件最好还准备一台示波器，要以能测高频的为好。

在进行逻辑电路实验时，还要准备一台逻辑检验器，它能判别片子引脚是高电平（约5V左右），还是低电平（0V），这种逻辑检验器还可以自己动手制作。

(3) 逻辑检验器的制作

按图1.2装配在实验板上，其中 G_1 可选用3DG6， G_2 可选用3AX1，LED是发光二

