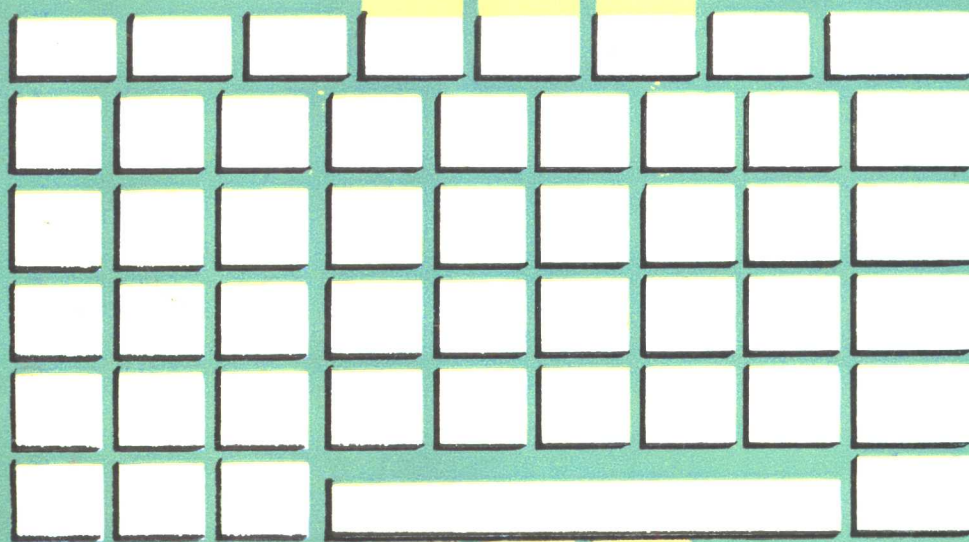


微处理器与微计算机在通信中的应用丛书

3

微处理机在彩色电视 技术中的应用

刘忠恩 李鹏飞 编著



国防工业出版社

微处理器与微计算机在通信中的应用丛书③

微处理机在彩色电视 技术中的应用

刘忠恩 李鹏飞 编著

国防工业出版社

内 容 简 介

本书是一本介绍微处理机在电视技术中应用的专业性读物。主要包括微处理机的基本原理、微处理机在彩色电视接收机、彩色录象机、彩色电视中心中的具体工程应用及其接口。同时,还介绍了多工文字广播和可视数据等新技术。

本书是中等专业性读物,适宜工作于电视技术行业的工程技术人员及广播电视专业的大学生阅读;对于无线电工程技术人员、大学生以及无线电爱好者也可作为参考用书。

微处理器与微计算机在通信中的应用丛书③

微处理机在彩色电视技术中的应用

刘忠恩 李鹏飞 编著

*

国防工业出版社出版

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

787×1092¹/₁₆ 印张7¹/₂ 169千字

1987年11月第一版 1987年11月第一次印刷 印数: 00,001—03,600册

统一书号: 15034·3250 定价: 1.65元

前 言

随着新技术革命的到来及人类向信息社会的过渡，电子计算机尤其是微处理机的广泛应用是这个时代的重要标志。编写本书的目的，就是介绍微处理机在彩色电视技术领域中的广泛应用的具体情况。

全书共包括七章。在扼要地介绍了微处理机的基本工作原理以后，重点地讲解了微处理机在彩色电视接收机、彩色电视中心、彩色录象机等领域中的具体应用及其接口。每种应用都从基本工作原理讲起，力求简明、概括、扼要。本书还用两章专门介绍了多工文字广播、可视数据等新技术。

本书第一章至第六章由刘忠恩同志编写，第七章由李鹏飞同志编写。全书由全子一同志审阅过。在本书的编写过程中蒋君湘同志、魏启福同志协助做了大量工作，在此表示衷心感谢。

由于时间仓促，编者水平有限，书中错误和不妥之处一定不少，欢迎读者批评指正。

目 录

第一章 微处理机的基本原理	1
§ 1-1 数制与编码	1
一、二进制数制	1
二、其他数制系统	2
三、二进制编码	3
四、计算机字的组成	4
§ 1-2 微处理器	5
一、微处理器的作用与组成	5
二、定时与控制部分	5
三、寄存器	6
四、算术逻辑单元 (ALU)	7
§ 1-3 存储器	8
一、存储器基本工作原理	8
二、随机存取存储器	9
三、只读存储器	10
§ 1-4 微处理机的工作原理	11
一、微处理机的基本结构	11
二、微处理机的总线系统	12
三、读出与写入过程	13
四、执行一个程序	14
五、接口与中断	16
六、微处理机在电视技术中应用的特点	18
第二章 微处理机在彩色电视接收机中的应用	19
§ 2-1 彩色电视接收机的基本原理	19
一、彩色电视接收机的组成	19
二、电视频道的划分	20
§ 2-2 用微处理机控制彩色电视接收机	21
一、问题的提出	21
二、控制系统的组成	22
§ 2-3 自动调谐系统	23
一、系统的作用与组成	23
二、相位锁定环路	24
三、频率锁定环路	24
四、自动调谐回路举例	25
§ 2-4 显示、模拟控制与预置程序	26
一、显示系统	26
二、模拟控制	27
三、程序的预置	28
§ 2-5 遥控系统	28
一、遥控系统的作用及要求	28
二、遥控系统的组成	29

§ 2-6 系统软件	31
一、对系统软件的要求	31
二、系统软件的基本工作原理	31
§ 2-7 总体介绍	35
一、总体介绍	35
二、微处理机在其他方面的应用	35
第三章 微处理机在彩色电视中心的应用	38
§ 3-1 彩色电视摄像机与视频通道	38
一、彩色摄像机系统的组成	38
二、彩色摄像机的质量指标	39
三、摄像机的视频通道	40
§ 3-2 微处理机控制的彩色电视摄像机	41
一、使用微处理机进行控制的特点	41
二、基本控制原理	42
三、白平衡及重合的调整	43
四、空间误差校正技术	46
§ 3-3 由微处理机控制的数据采集系统	46
一、数据采集的作用	46
二、系统工作原理	47
§ 3-4 微处理机控制的电视发射机	48
一、概述	48
二、数据采集	49
三、传感器	49
四、故障中断与主程序	50
第四章 微处理机在磁带录象机中的应用	52
§ 4-1 磁带录象机的基本工作原理	52
一、磁带录象机的特点	52
二、磁带录象机的组成	53
三、视频信号记录与重放原理	54
§ 4-2 用微处理机进行时间码误差校正	56
一、时间码及其读法	56
二、时间码误差的产生	57
三、时间码误差的校正	57
§ 4-3 由微处理机控制的变速放象系统	59
一、问题的提出	59
二、包络法	60
三、校正方法	61
四、微处理机控制系统	63
§ 4-4 微处理机在编辑系统中的应用	64
一、用于编辑系统的微处理机	64
二、主处理器与录象机控制接口	65
三、编辑系统	66
第五章 多工文字广播系统	68
§ 5-1 基本概念	68
一、基本概念	68
二、总体系统的组成	69
§ 5-2 多工文字广播系统的应用	70

§ 5-3 发送系统	72
一、系统工作原理	72
二、基本组成格式	73
三、各种制式简介	74
§ 5-4 接收设备	77
一、解码器工作原理	77
二、接收设备的组成	78
§ 5-5 多工文字广播的发展状况	79
第六章 可视数据	81
§ 6-1 概述	81
§ 6-2 可视数据系统的组成	81
一、可视数据系统的特点	81
二、可视数据系统的组成	82
三、终端的组成	83
§ 6-3 通过有线电视系统的可视数据	84
一、基本工作原理	84
二、可视数据系统的应用	85
§ 6-4 通过电话线路的可视数据	86
一、问题的提出	86
二、调制解调器的基本原理	87
三、类型及接口	88
四、同步与非同步传输	89
五、通过电话线路系统的可视数据	89
第七章 微计算机-电视显示系统	91
§ 7-1 家用微计算机发展概况	91
§ 7-2 微电脑-电视显示的组成原理及主要部件图	92
一、光栅扫描式显示方法	92
二、系统方框图及其功用	93
§ 7-3 显示存储器与荧光屏的对应方法	94
一、存储器直接对应法	94
二、面向字符显示的存储器与荧光屏的对应方法	95
三、活动图形与屏幕的对应关系	99
§ 7-4 显示存储中信息数据的修改	104
一、微处理机随机接入视频存储器的接口法	105
二、时间轮换显示 RAM 的接口方法	106
三、直接存储器访问 (DMA) 接口方法	107
§ 7-5 同步发生器	108
§ 7-6 软件简要的介绍	112

第一章 微处理机的基本原理

微处理机是利用大规模集成电路原理组成的可编程序的逻辑设备。由于它体积小、造价低、功能多及灵活性大等特点,因而被广泛地应用于各个领域。通常,微处理器、存储器以及输入输出设备连接起来就形成一台能够执行一定功能的系统设备,称为微型计算机或微处理机系统。为简便起见,本书将微处理机系统统称为微处理机。

本章不是系统地阐述微处理机的具体工作过程,而是以结论方式讲述其基本工作原理及其在电视技术领域中的应用的特点。如欲进一步学习微处理机方面的理论,可以参考其他相应方面的书籍。

§ 1-1 数制与编码

二进制数及编码是所有微处理机的基本语言,其他数制系统如十进制、八进制等也经常用到。因此,掌握这些数制与编码,对深入学习微处理机是十分重要的。

一、二进制数制

在二进制数制系统中,仅仅有两个符号或数字值,即0和1,通常称为比特(bit)。二进制数制系统是位置标量系统,即同一个数字所处的位置不同,其所表示的值也是不同的。例如,在二进制数1 1 1 1 · 1 1中,各位1所表示的值如下所示:

1 1 1 1 · 1 1 ——二进制

$2^3 2^2 2^1 2^0 \cdot 2^{-1} 2^{-2}$ ——权值

我们把各位1所表示的值称为各位的权,它以2的不同方幂表示。将二进制数中的各位乘以它的权值,再将这些乘积相加就得到该数等效的十进制数值。仍以1 1 1 1 · 1 1为例,

$$\begin{aligned} 1\ 1\ 1\ 1 \cdot 1\ 1_2 &= 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &\quad + 1 \times 2^{-2} = 15.75_{10} \end{aligned}$$

再如

$$\begin{aligned} 1\ 0\ 1\ 1 \cdot 1\ 0_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} \\ &\quad + 0 \times 2^{-2} = 11.5_{10} \end{aligned}$$

二进制系统里最大的数值是1,当对二进制计数时,首先从0数到1,到1以后再进行计数,就从1回到0,但这时高位比特将进位为1。

在二进制系统中,用 N 比特数字可以计数 2^N 个不同的数值,其中包括0。例如二个比特的数字,我们可以计数0 0、0 1、1 0和1 1、4个不同的数值。同样4个比特的数字,可以从0 0 0 0数到1 1 1 1,整个为 $2^4=16$ 个不同的数值。在 N 比特数字中,等效于十进制最大的数值是 2^N-1 。如4比特数字,最大的二进制数值是1 1 1 1,它等效于十进制的数值是 $2^4-1=15$ 。

微处理机是直接和二进制数字打交道的。一组二进制数字,并不都是用来表示一定的数值量,而经常表示非数字信息的各种类型的码。概括起来,一组二进制数字可以

表示:

- ① 实际的数值数字。
- ② 在存储单元中的地址。
- ③ 一组指令码。
- ④ 表示字母、字符的一组码。
- ⑤ 表示机器内部状态或外部仪器状态的一组码。

二、其他数制系统

其他数制系统,除了我们熟悉的十进制外,经常用到的还有八进制和十六进制系统。

八进制数字系统共有 8 个数字,即 0、1、2、3、4、5、6 和 7。它也是位置标量系统,即处于不同位置的数字,它所表示的数值也是不同的。将八进制数值转换为十进制的方法如下:

$$372_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 = 250_{10}$$

在八进制系统中,最大的数字是 7,计数从 0 开始数到 7,再继续计数则又重新从 0 开始,但同时临近高位进位。其计数顺序为 65、66、67、70、276、277、300 等等。

如果有 N 个八进制数字,则计数可从 0 一直到 $8^N - 1$,共有 8^N 个不同的数值。例如,三位八进制数字其计数从 0 0 0 到 7 7 7,共有 $8^3 = 512_{10}$ 个不同的八进制数值。

十六进制系统共有 16 个不同的数字符号,即从 0 到 9,另外再加上 A、B、C、D、E 和 F。它的计数从 0 一直到 F,到 F 后下一个计数又从 0 开始,同时向临近高位进一位,其计数顺序是 38、39、3A、3B、3C、3D、3E、3F、40、6BD、6BE、6BF、6C0 等等。

将十六进制数字转换为十进制数字的方法如下:

$$356_{16} = 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0 = 854_{10}$$

$$2AF_{16} = 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 = 687_{10}$$

八进制与十六进制系统常常用于表示多比特的二进制数字,这时既简便又提高了效率。表一给出了二进制、十进制、八进制、十六进制之间的相互对应关系。至于各种数制系统的算术运算以及它们之间的相互转换,我们就不具体讨论了。

表 一

十 进 制	八 进 制	十 六 进 制	二 进 制
0	000	00	0000
1	001	01	0001
2	002	02	0010
3	003	03	0011
4	004	04	0100
5	005	05	0101
6	006	06	0110
7	007	07	0111
8	010	08	1000
9	011	09	1001
10	012	0A	1010
11	013	0B	1011
12	014	0C	1100
13	015	0D	1101
14	016	0E	1110
15	017	0F	1111
16	020	10	10000

三、二进制编码

二进制编码是利用特殊的一组符号表示数字、字母、字符及其他信息，本章将重点介绍经常用到的二-十进制码（BCD 码）和字母数字码（ASCII 码）。

1. 二-十进制码

数字系统在内部操作过程中总是使用二进制数字，而它的外部世界则用人们习惯的十进制系统，这就意味着在这两个系统间的转换是要经常发生的。通常使用二-十进制码，即 BCD 码进行二者之间的转换，这种码具有二进制和十进制两种数制的某些特征。

所谓 BCD 码，就是将每一个十进制数值的数字分别用它等效的二进制来表示。因为十进制最大的数字是 9，因此用 4 比特数字表示即可。例如十进制数值 874，它所对应的 BCD 码为：

8	7	4	——十进制数
1000	0111	0100	——BCD 码

由此可见，每一个十进制数字直接转换为其等效的二进制表示即可。

BCD 码的主要优点是与十进制间相互转换非常简便。在数字机器里，只要输入或者输出是用十进制信息，那么其工作过程就是使用 BCD 码。在微处理机中，输入数值都是通过键盘以十进制方式输入的，而输出又是以十进制进行显示，其中间过程就是使用 BCD 码。

2. 字母数字码

要使计算机更有效用，它就必须能够处理非数字信息。也就是说，它具有识别表示数字、字母和其他符号的能力，这些通称为字母数字码。它一般包括：④26 个小写字母，⑤26 个大写字母，⑥十个数字，⑦大约 25 个特殊符号等。

使用最广泛的是 7 比特的美国信息交换标准码，通常称作为 ASCII 码。绝大部分的小型计算机和微处理机厂家都采用 ASCII 码，其用途是在计算机和外部输入/输出仪器间（例如电传打字机）传输字母数字信息。表二列出的是部分 ASCII 码，同时也给出了等效的八进制和十六进制码。

表 二

字 符	ASCII 码	八 进 制 码	十 六 进 制 码
A	1000001	101	41
B	1000010	102	42
C	1000011	103	43
D	1000100	104	44
E	1000101	105	45
F	1000110	106	46
G	1000111	107	47
H	1001000	110	48
I	1001001	111	49
J	1001010	112	4A
K	1001011	113	4B
L	1001100	114	4C
M	1001101	115	4D
N	1001110	116	4E
O	1001111	117	4F

(续)

字 符	ASCII 码	八 进 制 码	十 六 进 制 码
P	1010000	120	50
Q	1010001	121	51
R	1010010	122	52
S	1010011	123	53
T	1010100	124	54
U	1010101	125	55
V	1010110	126	56
W	1010111	127	57
X	1011000	130	58
Y	1011001	131	59
Z	1011010	132	5A
0	0110000	060	30
1	0110001	061	31
2	0110010	062	32
3	0110011	063	33
4	0110100	064	34
5	0110101	065	35
6	0110110	066	36
7	0110111	067	37
8	0111000	070	38
9	0111001	071	39
.	0101110	056	2E
(	0101000	050	28
+	0101011	053	2B
)	0101001	051	29
=	0111101	075	3D

四、计算机字的组成

在计算机中，一个字是一组二进制数，它一般占一个存储单元。虽然计算机字由若干个二进制数字所组成，但计算机把每个字作为一个单一的单元对待。计算机字是计算机使用的基本信息单元。通常微处理机使用的字长为 8 位。

存储在计算机存储单元里的字包含着不同类型的信息，可以概括为：①二进制数据、②编码数据、③指令。

二进制数据可以是各种参加运算的数据量，例如我们可以将表示室温的数据 0 0 0 1 0 1 0 0 作为一个计算机字存放在某个地址的存储单元中。一个 8 比特的字，它的最大数据字是 1 1 1 1 1 1 1 1 等效于十进制中的 $2^8 - 1 = 255_{10}$ 。

广泛使用的编码数据就是前面提到的 BCD 码和 ASCII 码。许多计算机能够以 BCD 码数字进行算术运算作为其正常功能的一部分，但大部分微处理机都还不能这样做。

值得注意的是，同一组数据字可以表示各种不同的内容，在数据字翻译时必须知道自己使用的是哪种类型的数据字。例如在存储单元中存储的数字是 0 1 0 1 0 1 1 0，它可以表示为二进制的数，相当于十进制的 86，也可以是 BCD 码的 56，同时还可以是 ASCII 码中的字母 V。因此，要进行正确的解释才行。

指令字是计算机在运算过程中所需要的必要信息，它分为二部分。一部分称为操作码，表示计算机要完成的运算，例如加法、减法等。第二部分是操作数地址，它表示在存储单元中操作数的地址。

作为例子，假定操作码为 0 1 0 0 表示相加，地址码为 0 1 0 1 1 0 1 0 0 1 1 1

0 0 1 0, 则整个指令字为

0 1 0 0 | 0 1 0 1 1 0 1 0 0 1 1 1 0 0 1 0

操作码

地址码

将整个指令字写为 16 进制, 则为

4

5 A 72

操作码

地址码

该指令字告诉计算机作下列工作: 从地址 5 A 72 取出储存在那里的数据字, 并将它馈送到算术逻辑单元 (ALU), 与累加器中的数值相加, 然后将其和储存在累加器中。

§ 1-2 微 处 理 器

一、微处理器的作用与组成

各种微处理机尽管在结构上千差万别, 但是它们都有一个公共的部件, 即微处理器。微处理器是微处理机的中心处理单元, 也可以说是它的核心。因为微处理器的能力直接决定了微处理机的能力: 它的工作速度决定了微处理机的最高速度, 它的地址和数据管脚数决定了微处理机的存储能力和字的长短, 它的控制管脚数决定了使用何种类型的输入/输出 (I/O) 外围电路。

微处理器的主要作用可以概括如下:

- ① 对微处理机的所有部分提供时钟和控制信号。
- ② 从存储单元中提取指令和数据。
- ③ 向输入/输出设备或从输入/输出设备传送数据。
- ④ 对指令进行解码。
- ⑤ 按着指令的要求, 进行算术和逻辑运算。
- ⑥ 对输入/输出产生的控制信号作出相应动作, 如复位和中断等。

在微处理器中包括执行这些功能的所有逻辑电路。但是必须记住, 微处理器的内部逻辑电路是不能从外部加以控制信号来进行控制的, 它只能以软件的方式加以控制。也就是说只有将程序放到存储器中, 再由微处理机进行运算才能影响微处理器内部的电路。这就使得微处理器变得十分灵活、多用。因为当改变微处理器的工作时, 只简单地改变一下程序即可, 这要比改变硬件容易得多了。

图 1-1 绘出的是微处理器组成示意图。它包括寄存器、控制与定时以及算术逻辑等三个部分。利用地址总线、数据总线和控制总线与微处理机的其他单元相连接。其中地址总线是单向传输的, 而数据总线和控制总线则是双向传输的。

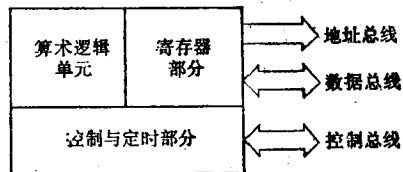


图 1-1 微处理器组成示意图

下面我们对各组成部分, 分别加以扼要的讨论。

二、定时与控制部分

定时与控制部分产生各种控制信号, 使微处理机的各个部分按着程序的要求, 有条不紊地完成各种运算和动作。在这里我们仅简要地介绍微处理器所具有的通用控制总线

信号。

复位 (reset): 所有的微处理器都有此输入。当该输入起作用时, 微处理器绝大部分的内部寄存器均被置位为零。通常程序计数器 (PC) 也被置为零。

读/写 (R/W): 当微处理器进行读出或写入动作时, 将有一控制信号通知微处理机的其他部分。某些微机使用分开的控制线, RD用于读出动作, 而WR用来表示写入动作。

输入/输出请求 (IORQ): 这是微处理器的一个输出信号, 它表示一个输入/输出设备已经连接就绪。

就绪 (ready): 它是微处理器的一个输入端, 在使用速度很慢的存储器或 I/O 设备时用该控制线。当速度低的设备被地址解码电路选择时, 它也马上向微处理器发出一就绪信号。这时, 微处理器中止它所有的内部工作而进入一种称为“等待状态”。直到设备可以传出或接收数据时才消除就绪信号, 使微处理器开始工作。

保持 (Hold): 这是微处理器的一个输入端, 用于直接存储访问。当某一外部设备传送一信号到此输入端时, 微处理器完成正在运算的指令, 然后将它的地址和数据总线漂动, 即与微处理机的总线形成开路状态, 这样就可使外部设备使用地址和数据总线, 直到保持信号消除为止。

保持认可 (HLDA): 这是微处理器产生的一个输出信号。它告诉外部逻辑电路微处理器处于保持状态, 可以使用数据和地址总线。

中断 (INT): 这是微处理器的一个输入端。I/O 设备利用该输入端可以中断微处理器正在执行的程序, 并使微处理器跳转到一个称为中断服务程序上去。当执行完该程序后, 微处理器重新从被中断处执行原来的程序。

中断允许 (INTE): 这是微处理器的一个输出端。它告诉外部设备, 微处理器的内部中断逻辑是否允许中断。如果允许, 则 $INTE = 1$, 如上面提到的微处理器将被中断; 如果不允许, 则 $INTE = 0$, 微处理器对INT输入将不起作用。INTE 的状态是可用软件控制的, 例如程序中可以包括一个使 $INTE = 0$ 的指令, 这样中断的动作就不会发生了。

除此之外, 还有其他一些控制信号, 这里就不一一介绍了。

三、寄存器

寄存器是微处理器的重要组成部分, 最简单的寄存器是一组用于储存二进制信息的存储单元。最广泛使用的寄存器是FF, 而计数器便是FF寄存器的一个具体例子。

在微处理机内部, 最经常发生的动作是二进制信息从一个寄存器转移到另一个寄存器, 寄存器的基本作用就是用来储存各种数据、地址、指令码以及微处理器在工作时的各种状态信息, 有些寄存器用来作为计数器进行计数。下面我们仅重点介绍几种寄存器。

指令寄存器 (IR): 当微处理器从存储单元提取一个指令字时, 该指令字便被送到指令寄存器, 然后由指令解码电路决定哪一个指令首先被执行。IR 的大小应与微处理器字的长短相同, 对于一个 8 比特的微处理器, IR 也应为 8 比特长。

程序计数器 (PC): 程序计数器也是一种寄存器, 它控制指令在程序中执行的顺序, 通常按顺序 0、1、2、3、4 等来完成。任何时候, 它总是指示要提取的下一个

指令所在的存储单元。

PC的大小决定于微处理器所能处理的地址比特数目，即它与地址总线具有相同的比特数。使用者不能直接作用于PC，但有些指令却可以改变PC的正常顺序，使微处理器提取正常顺序以外的指令。例如JMP、JMZ就是指令的例子。

储存地址寄存器 (MAR)：它是一个暂存单元，用来暂存微处理器要从存储单元读出或要向存储单元写入的数据地址。例如当执行加法指令时，微处理器将加法指令的操作数地址部分送到MAR，则MAR的内容便被放到地址总线上，这样在下一个时钟周期就可以提取数据了。

累加器 (Accumulator)：是最常用的寄存器之一。在算术和逻辑运算时，它具有双重功能。运算前，它保持一个操作数，运算后，它保存所得的和、差或逻辑运算结果。除此之外它还有别的用途，例如对于那些要送到输出设备的数据它可作为存储寄存器；而对那些从输入设备来的要读入的数据则可作为接收寄存器等。

一般讲，累加器与微处理器具有相同的比特数目。也就是说，一个8比特的微处理器将有一个8比特的累加器。

除上面介绍的以外，还有通用寄存器、变址寄存器、状态寄存器以及堆栈指示器寄存器等等，在这里我们就不一一讨论了。

四、算术逻辑单元 (ALU)

微处理器的算术逻辑单元可以进行各种各样的算术和逻辑运算。这些运算可以包括有二个操作数，一个在累加器，一个在存储器；或者一个在累加器，另一个在微处理器的内部寄存器。某些运算也可能只包含一个操作数，它可以在累加器、寄存器或者存储器中。

图 1-2 给出的是算术逻辑单元工作简化示意图。在方框图中，有进行各种算术、逻辑运算所需要的各种逻辑电路。有二个操作数（通常为 8 比特）作为 ALU 的输入，从状态寄存器来的进位标记也是ALU的一个输入。ALU要执行的动作由微处理器控制部分来的控制信号所决定，控制输入的数目则随微处理器不同而异。

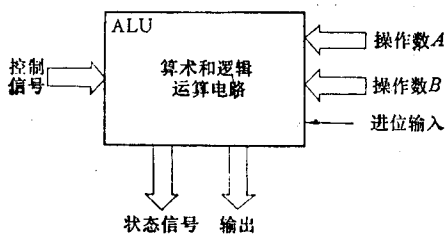


图1-2 算术逻辑单元工作简化示意图

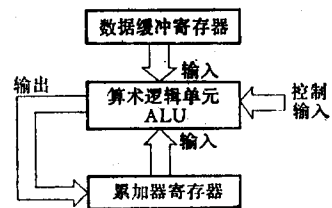


图1-3 双操作数运算示意图

经过运算，ALU产生二组结果，一组是操作数运算的结果，它被馈送到微处理器的相关部分；另一组是状态信号，它被馈送到状态寄存器去置位或清除各种标记比特的状态信号。例如，计算结果恰好为0，则有一信号馈送到状态寄存器中的 z 标记，并将它置于高电平状态等。

操作数 A 和 B 可以从不同的部分来，当二个操作数被ALU运算时，一个操作数来自累加器，而另一个操作数通常则来自从存储器提取的数据字，运算结果一般都送到累加器去，如图 1-3 所示。

当 ALU 运算一个操作数时，这个操作数可为累加器的内容，也可为一个通用寄存器、一个变址寄存器的内容，或者是一个存储器的数据字。这时运算的结果将被馈送到操作数的来源处。当操作数是一个存储器字时，其结果被馈送到数据缓冲寄存器，从那里，微处理器再将它写入到操作数的存储单元地址中去。

单操作数可以进行清除、变为反码、增 1、减 1、移位以及循环等多种工作；而对于双操作数来说，则可以进行加法、减法、比较、逻辑或、逻辑与、异或及十进制 (BCD) 算术运算等多种功能。

上面提及的功能并没有包括乘法、除法、平方等比较复杂的运算。这些运算对于通用的微处理器来说是不能直接实现的，因为它还需要一些特殊的电路。要完成这些运算，可让微处理器进行一系列简单的运算以达到计算的要求。例如，可以通过一系列的移位和加法达到乘法的目的；而除法则通过一系列的移位和减法来实现。

§ 1-3 存 储 器

一、存储器基本工作原理

存储器是用来储存二进制信息的部件。一组二进制信息称为字，它是计算机中信息和数据的基本单位，一个字由多少比特所组成则随使用机器不同而异。一个存储器可以储存一定数目的字。例如通用的存储器，它的储存能力是 256 个字，每个字又由 8 比特组成，通常记作 256×8 。

我们可以认为存储器是由一组寄存器所组成，而每个寄存器储存一个字，如图 1-4 所示。每个寄存器的长度是每个字的比特数，寄存器的数目是储存字的数目。通常这样表示一个存储器：储存 N 个字，每个字为 M 比特，则记作 $N \times M$ 存储器。一个存储器通常储存的字为 64、256、512、1024、2048 和 4096。

存储器中的每个寄存器都有二种不同的工作情况：读出和写入。读出是将储存在寄存器里的字取出来，并将它放到待用的一个地方的过程。读出的过程并不改变寄存器的内容。写入是将一个新字放入特定寄存器的过程，写入过程将毁坏原先储存在该寄存器中的那个字。以后我们将会发现，并不是所有的存储器都具有写入的能力。

每一个寄存器都给它一个数目字称为代码，这个数字通常称为是它的地址，因为这个数字直接确定了该寄存器的位置及它所储存的字。地址是一个十分重要的数字，因为我们利用它来选择合适的存储器来进行写入和读出。

图 1-5 是一个典型的存储器的内部结构及工作示意图。该存储器可以存储 64 个字，每个字为 4 比特。这些字的地址范围是从 0 到 63_{10} 。要做到能选择 64 个地址中的一个进行写入或读出，一个二进制地址码必须加到解码器电路。因为 $2^6 = 64$ ，所以解码器需要 6 比特的输入码，每一个地址码相应于特定的寄存器。例如，地址码

$$A_5 A_4 A_3 A_2 A_1 A_0 = 011010$$

因为 $011010 = 26_{10}$ ，解码器输出为 26，则使寄存器 26 参予工作。

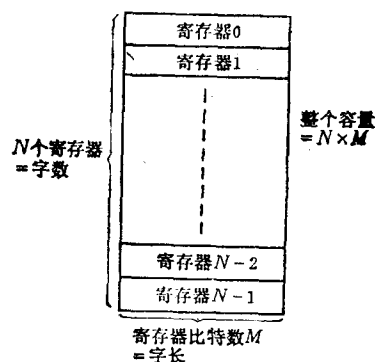


图1-4 存储器组成示意图

存储器都有一个读/写 (R/W) 控制输入。当进行读出时, 使 $R/W = 1$ 、 $CS = 1$ 。

这时输出缓冲器工作, 被选择寄存器的内容在 4 个数据输出线上出现。而 $R/W = 1$ 却使输入缓冲器不工作, 因此在读出过程中, 数据输入并不影响存储单元。

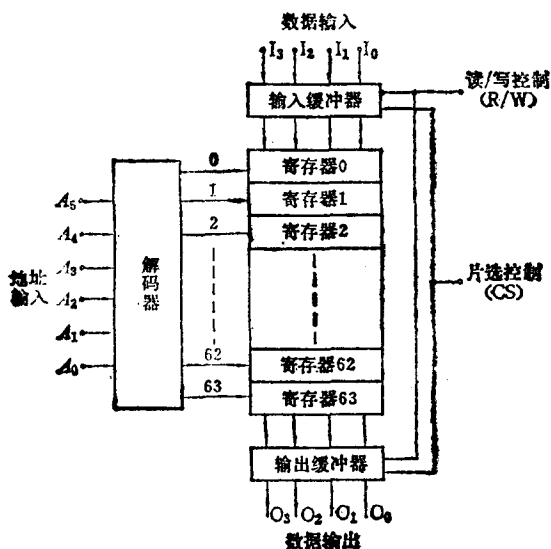


图1-5 存储器内部结构示意图

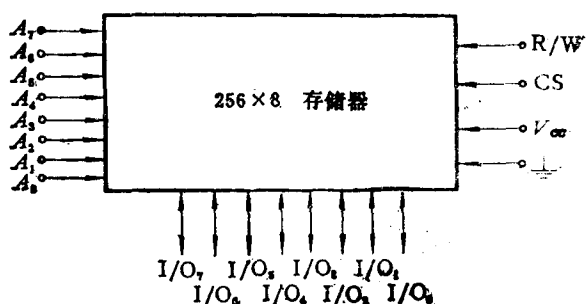


图1-6 256×8 存储器管脚作用分布图

要将一个 4 比特的字写入到选择的寄存器中去, 需使 $R/W = 0$ 和 $CS = 1$ 。这就保证输入缓冲器正常工作, 使 4 比特的字写入到特定的寄存器中, 在 $R/W = 0$ 时, 数据输出与数据总线间处于开路状态。

片选的作用是控制整个存储器处于正常工作状态还是处于中断工作状态。在中断状态时, 即 $CS = 0$, 所有的数据输入和数据输出都被中断, 所以写入和读出的动作均不能发生。这时, 存储器中的内容将不受到影响。

通常, 一个存储器就是一个集成块, 为了减少组件的管脚, 都采用公共的输入/输出 (I/O) 管脚, 利用 R/W 控制管脚的作用。在读出过程中, I/O 管脚作为数据输出; 在写入过程中, 则又作为数据输入。图 1-6 给出的是 256×8 存储器各管脚作用图。

二、随机存取存储器

随机存取存储器也叫作 RAM, 信息以字为单元存储在存储器中, 而且可以被直接地选择。在这种方式中, 选择任一个字单元所需要的时间几乎相等。利用二进制地址码来完成选择的功能, 这样就可以进行写入或读出操作了。

RAM 是用作暂时存储程序及数据用的。例如在调测彩色电视接收机的过程中, 我们可以将选某一电视台的数据信号送入, 或写入一程序, 使接收机在某一特定的时间转换到某一电视台的节目或关闭电视机。这些数据都是通过使用者由小键盘首先送入到 RAM 里, 然后由微处理器进行处理。由此可见, 在微处理机的运算过程中, 许多 RAM 中的内容将会连续的发生变化。为了不致降低运算速度, 希望 RAM 的写入与读出时间越短越好。

RAM 的最大缺点是它的易失性。就是说, 当电源从这块集成电路中断时, RAM 储

存的所有信息都将被遗失。正因为如此,有些RAM采用备用低压电池,一旦出现电源中断事故,其存储的信息也不致遗失。这一点作为工程应用是十分重要的,因为一般都是将程序和数据写入RAM中。在电源中断的情况下,备用电池通常可以维持工作达三个月之久。

RAM存储器具有静止型和动态型二种类型。静态RAM主要是由FF寄存器及必要的电路所组成。每一个比特需要一个寄存器,辅助电路比较复杂。存储一个比特的信息有时需要多达10个晶体管,这就限制了它的储存容量,通常每片容量为4096比特。它的优点是只要电源不中断,它所存储的信息就将永远保持着。

动态RAM不在FF中存储二进制数据,而是将0和1以向小电容充电的方式存储起来。这些电容不需要特别制造,而用MOS晶体管的门电容即可。采用这种技术。只用三个晶体管就可存储一个比特,因此可以使存储容量大大提高。目前,16K动态RAM已是很普通的了。

动态RAM中的电容总是在不断地放电,因此要保持原来的数据,就要周期性的对电容进行充电,称为刷新存储器,这就要求比较复杂的外部电路。同静态RAM一样,如果电源突然中断,存储的信息也将遗失。

三、只读存储器

只读存储器又称作ROM,它是被当作固定信息源来使用的,即仅从存储器中读出数据。图1-7给出的是 $1K \times 8$ 只读存储器符号表示图。

从图中可以看出,该ROM可以存储1024个8比特的字,因此通常称为 1024×8 ROM,更进一步简化为 $1K \times 8$ ROM。因为要存储1024个不同的字,所以需要10个地址输入;字的长度为8,故有8根输出线。片选是用来控制存储器输出的。当 $\overline{CS} = 0$ 时,被选择的数据在输出端出现;当 $\overline{CS} = 1$ 时,输出呈现开路,无输出。

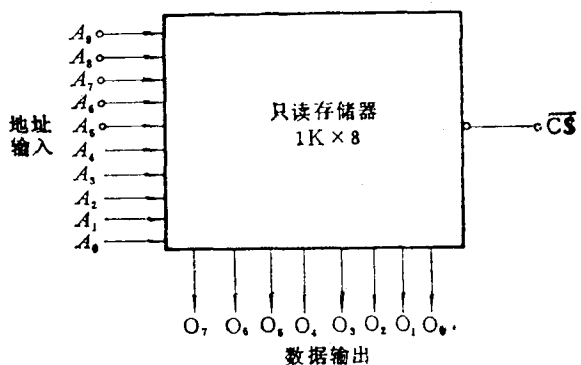


图1-7 只读存储器表示方法

ROM没有数据输入和R/W控制,因为写入过程并不是ROM正常工作的一部分。某些ROM有特殊的输入端将数据写入ROM,但在符号中一般都不表示出来。

ROM主要用来存储那些固定不变的信息,例如常数表、数据转换表以及固定不变的计算程序等。在后面几章的具体应用中,我们要使用各种各样的ROM。例如电视波道数据转换表、字符发生器等都是采用专用的ROM所制成。

ROM中的信息一般是在制造或者安装机器的过程中写入的。这些信息即使在电源中断后也不会消失,因此,ROM又称作非易失性存储器。

按着将信息或程序写入到存储器的方式,ROM又可以分为以下几种类型。

1. 制造厂编程的只读存储器:这种只读存储器由厂家采用掩模的方式,按照预定的用途形成合乎需要的模型。由于掩模工艺是很昂贵的,所以这种类型的ROM只有在