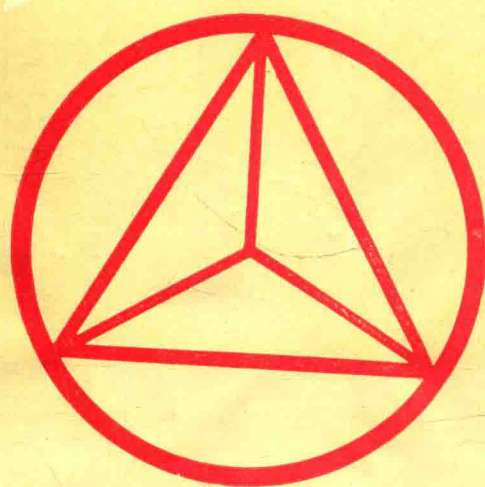


# 晶峰器件应用

一位微处理器应用实例选

(第二届一位机应用技术交流会论文选之一)



15

专 辑

上海元件五厂

## 代 前 言

《晶峰器件应用》自创刊以来,至今已出了十五期。作为一个厂办的技术性刊物,尽管我们在编辑水平,稿件质量,刊登内容等诸方面,都存在某些不足,但还是得到国内各行各业,厂、所、院校的热情帮助和支持,特别是广大《晶峰》器件用户单位有关技术人员的关怀与帮助,使这个“刊物”保持它介绍《晶峰》器件、推广《晶峰》器件以及交流《晶峰》器件应用经验的特色。近年来,我们这个器件应用刊物,曾不断收到不少读者的来信来稿,提了许多宝贵意见,使我们的刊物得以改进。

在这一期中,我们选辑了一批 5G1400 一位微处理机的论文,其中有机床电气、自动控制和技术革新中的各种应用总结,这些总结和论文中,充分显示了一位微处理机的灵活、简便的特点和在工业控制中的广泛应用前景。有些文章还对一位微机的某些可能的不足和应用中可能出现的问题作了建设性的改进,并提出了相应的措施。其中大多数对读者会有一定的参考价值。

在这之前,我厂《晶峰器件》编辑部曾先后编辑出版过二次论文汇编和专辑,还出版过介绍一位微处理机的芯片专文,这些介绍和文章曾多次再版,受到读者的欢迎,对一位机的组装和应用起到一定的作用。鉴于这次资料的一贯脱销,因此决定再一次编辑出版第三个专辑,这就是这次第十五期。

目前,我厂从国外引进了一部分先进的大规模集成电路制造设备,结合国内攻关设备建成了上海市第一条大规模集成电路引导线,使我厂在大规模集成电路生产和开发能力上有较大的提高,我厂 14500 系列一位微处理机电路无论在质量、性能、价格上均可与引进同类芯片相匹敌,不仅如此,我厂还专门生产了一位微机模块,使用户试用(或应用) 5G14500 系列更方便,目前已接受用户订货。

在这里,我们深切希望广大《晶峰器件应用》的读者对我们的刊物进一步提出宝贵的意见,帮助我们改进,并欢迎用户单位有关人员来稿,把我们的用户园地搞得更加生动活泼,发挥它技术交流的良好作用。限于我们的编辑水平,这一期中的错误与不足在所难免,祈望读者谅解。

王国定

1985.4.2.

## 5GM—14500 I 型一位微机系统

一位微机系统是一种可程序的逻辑控制装置，它以我厂生产的5G14500一位工业控制单元（ICU）为核心，用CC4000B系列CMOS器件作为辅助电路，具有功耗低、抗干扰性能好、便于本厂配套等优点。

本系统适用于机械、冶金、纺织、食品加工、化工等行业自动控制中的程序控制设备。

板上备有键盘编程器以及各种内容的光电显示，对于初学者作练习，或者用于高等院校及中等专科学校进行教学极为理想。

### 主要技术指标：

1. ICU具有16条指令。

2. 存储器容量：

EPROM:  $2K \times 8$ ，可存放1024条指令。

RAM:  $256 \times 8$ ，可存放128条指令。

3. 暂存器:  $256 \times 1$  单元。

4. I/C寻址范围：共11位地址码。

板内有输入/输出各16点。不加译码器，I/O可扩展到64点。

5. 键盘编程。

编好的程序可通过键盘送入RAM中，存储器地址和内容由LED显示。

6. 输入/输出：

输入有13个开关，输出端有14个LED显示。可供程序的验证和练习。

7. 电源电压: 5VDC、 $\pm 10\%$ 。

如关闭EPROM和LED电源，整个系统工作电流小于10mA。

8. 时钟频率：从单步到100KHz。

上 海 元 件 五 厂

## 生 产 经 营 总 纲

### 确立经营为中心 讲求实效为重点

产

品

方

向

坚决贯彻大规模集成电路为主，大规模集成电路和中、小规模集成电路结合的方针。大力开发各类大规模集成电路，努力生产优质线性电路、MOS电路和低噪声器件，加速发展CMOS模拟电路，认真搞好高速和超高速集成电路，以形成一个综合性大规模集成电路开发和中试生产基地。

#### 《 经 营 管 理 百 字 方 针 》

经

营

思

想

|       |       |
|-------|-------|
| 立足长远， | 着眼当前。 |
| 打好基础， | 快上水平。 |
| 开拓市场， | 加速更新。 |
| 巩固老户， | 发展新户。 |
| 胸怀全局， | 面向世界。 |

经

营

方

针

|       |         |
|-------|---------|
| 质量可靠， | 确保经久耐用。 |
| 性能优良， | 达到国际标准。 |
| 品种齐全， | 形成系列配套。 |
| 价格低廉， | 实行薄利多销。 |
| 应用广泛， | 渗透各行各业。 |
| 服务周到， | 满足用户需要。 |

# 晶 峰 器 件 应 用

## 第 十 五 期

1985年7月出版

### 目 录

#### 代 前 言

- (一)一位微处理机系统设计的不严密性、由此造成的后果以及改进措施 ..... 张勉令 ( 1 )
- (二)有关一位微处理机问题的问答 ..... 董伯明 ( 5 )
- (三)一位机的前途 ..... 郝鸿安 ( 8 )
- (四)MIC400微型工业控制器在长春第一汽车制造厂铸造分厂的应用  
..... 机械部自动化所 杨昌焜 ( 10 )
- (五)用于数控机床的一位微处理机  
..... 上海第二机床厂 姜继明 邓时鸣 ( 28 )
- (六)中频感应加热的一位机控制  
..... 兵器部第六二研究所 陈志强 ( 39 )
- (七)一位机在自来水厂的应用  
..... 四川皮革厂 王 铀 陈 燕 ( 45 )
- (八)SXZ440自动锁线机一位微机控制系统  
..... 上海市电气自动化研究所 郑德彰 周耀雄 ( 47 )
- (九)一位微机组成的遥信系统  
..... 上海铁道学院 员春欣 ( 53 )
- (十)WGK—130一位机在UX02型曲柄加工组合机床上的应用  
..... 广州南洋电器厂研究所 刘奕华 广州自行车厂 郑惠明 ( 56 )
- (十一)一位机为核心的自动给药控制仪在临床上的试验  
..... 上海第二医学院 蒋学鹤 王载礼 金正均 ( 60 )
- (十二)一位微机GK4在真空控制系统中的应用  
..... 机械工业部第二设计研究院 林道止 顾建华 汪杭明 ( 63 )

- (十三)一位微处理器在功率因数自动补偿器中的应用  
..... 四川重庆第一棉织厂 谢公勇 代燕芳(68)
- (十四)ES16/14500一位微处理机在精密半自动车床上的应用  
..... 北京市机电研究院机床室 王文生 唐 苏(87)
- (十五)一位微机控制自行车桥管专机  
..... 上海自行车三厂 王子龙(88)
- (十六)一位微处理机WAIT—STEP结构程序及其实现方法  
..... 湖北汽车工业学院 汪世文(101)
- (十七)LTS—82系列一位微处理机在“空调”控制系统中的应用  
——节能优化  
.....核工业部728工程研究设计院 密龙虎(109)
- (十八)一位机系统程序计数器(PC)的应用技术  
..... 杭州大学物理系数字技术研究组 吴训威(115)
- (十九)锅炉燃烧的一位机自动控制  
..... 上海海运学院防污染研究室 金起农(120)
- (二十)一位微机在核电站化容补给水程控试验系统中的应用  
.....核工业部728工程研究设计院 李惠馨(123)
- (二十一)一位微处理机在回收合成氨弛放气中氢气的工艺装置上的应用  
..... 吴淞化肥厂 王昌隆(128)
- (二十二)一位微处理机控制的西湖牌全自动捆扎机  
..... 浙江省邮电机械厂 杜顺庆(133)

# 一位微处理机系统设计的不严密性、由此造成的 后果以及改进措施

张勉令

本文试图说明一位微处理机由于输入输出部份逻辑设计的不严密性,以致引起哪些不正常现象,产生什么后果。目的是与读者共同探讨改进措施。

之所以要提出这个问题,是因为涉及到运算结果的输出。如大家所知,计算机无非是用来产生运算结果并把它输出。如果输出产生误操作,即使运算结果再精确,最后仍然出错。前功尽弃。

首先阐述,按常理,完整的逻辑设计应该怎样?

一位微机输入输出器件大致可分为两类:输入选择器IS和输出锁存器OL。每片IS有八个输入 $X_0 \sim X_7$ 。当选中时,将 $X_0 \sim X_7$ 上的外来信息输入至双向数据总线D(以下简称D)。IS专将外来信息输至D。而OL有两种使用方法:一仅仅作为输出通道,用符号OL<sub>①</sub>表示;二兼有中间结果暂存功能的输出通道,用符号OL<sub>②</sub>表示。前者只有写功能,即将运算结果经D写入OL内的锁存器中;后者除具写功能外,还能将原写入的内容再送回D。就是说,后者还具有似IS的读功能。因此OL<sub>②</sub>既是输出器件,又兼作了输入器件。

现在,讨论IS、OL在上述三种使用方法中完整的逻辑设计。

IS、OL的片选条件有三项。

第一项:指令。输入指令选中输入器件,输出指令选中输出器件。

第二项:时间。一位微机每个指令周期

分为 $X_1 = "1"$ 、 $X_1 = "0"$ 前后两部份:前为取指周期,ICU自存贮器取指令操作码;后为执行周期,进行输入、运算、输出等操作。由此可见,选中IS进行输入、选中OL进行输入输出的时间是在后半周期,即 $X_1 = "1"$ 时。

第三项:I/O(输入/输出)地址。输入地址表示ICU自哪片IS上哪个输入端( $X_0 \sim X_7$ )、或是哪片OL的哪个锁存器输入信息;输出地址则表示将结果寄存器RR中的运算结果输出到哪片OL的哪个锁存器。

无论是IS还是OL,都设有三位地址线。IS输入地址线CBA,OL I/O地址则是 $A_2 A_1 A_0$ 。由它们来指定该片选中时,自 $X_0 \sim X_7$ 或 $Q_0 \sim Q_7$ 中哪一个单元出入信息。系统中将所有IS的CBA与所有OL的 $A_2 A_1 A_0$ 按位并联后依次连至存贮器I/O地址低三位。余下的I/O地址自第四位开始直至最高位的译码信号,也用作片选条件之一。

I/O地址的译码,用于IS的则是输入地址的译码 $DC_i$ ,用于OL<sub>①</sub>是输出地址译码 $DC_o$ ,用于OL<sub>②</sub>则是I/O地址的译码 $DC_{I/O}$ 。为了确保入出数据时,每次只是在一个点上进行,因此 $DC_i$ 可等于 $DC_o$ ,但不得等于 $DC_{I/O}$ 。换言之,IS输入地址也可作为OL<sub>①</sub>的输出地址,但这两者不可等同于OL<sub>②</sub>的I/O地址。

根据三述三项条件,依次看看IS、OL<sub>①</sub>、OL<sub>②</sub>严密的逻辑设计

先讨论IS。IS无片选端,用失效端(三

态控制) DIS 作片选。当选甲时, 处于个失效状态; 不选中, 失效, 数据输出端 Z 呈高阻, 为别片选中创造条件。其真值表见图(一)

| C | B | A | 禁止<br>I <sub>nb</sub> | 三态控制<br>DIS | 输出<br>Z        |
|---|---|---|-----------------------|-------------|----------------|
| 0 | 0 | 0 | 0                     | 0           | X <sub>0</sub> |
| 0 | 0 | 1 | 0                     | 0           | X <sub>1</sub> |
| 0 | 1 | 0 | 0                     | 0           | X <sub>2</sub> |
| 0 | 1 | 1 | 0                     | 0           | X <sub>3</sub> |
| 1 | 0 | 0 | 0                     | 0           | X <sub>4</sub> |
| 1 | 0 | 1 | 0                     | 0           | X <sub>5</sub> |
| 1 | 1 | 0 | 0                     | 0           | X <sub>6</sub> |
| 1 | 1 | 1 | 0                     | 0           | X <sub>7</sub> |
| φ | φ | φ | 1                     | 0           | 0              |
| φ | φ | φ | φ                     | 1           | 高阻抗            |

“φ”任意状态  
图(一) IS真值表

IS 为输入器件, 只有输入指令才可能选中它。一位微处理器十六条指令, 其中输入指令占九条。它们是: LD、LDC、AND、ANDC、OR、ORC、XNOR、IEN、OEN。只要是这九条中任一条, 即 LD+LDC+AND+ANDC+OR+ORC+XNOR+IEN+OEN 为“1”, 都会选中 IS。

第二项条件, 时间。如前所述, 应为执行周期  $\overline{X}_1 = "1"$  时。

第三项, I/O 地址。前面说过, 此项条件现应为输入地址的译码信号 DC<sub>I</sub>。

综上所述, 选中 IS, 使之处于不失效, 即 DIS = “0” 的条件应为:

$$DIS = (LD + LDC + AND + ANDC + OR + ORC + XNOR + IEN + OEN) \cdot \overline{X}_1 \cdot DC_I \quad (1)$$

式中第一项为九条输入指令的或信号。而 ICU 不提供 LD、LDC 等指令的译码。诚然, 可将指令线 I<sub>3</sub>、I<sub>2</sub>、I<sub>1</sub>、I<sub>0</sub> 通过译码来

获得。如 LD 操作码为 0001, 代数式为  $LD = \overline{I}_3 \overline{I}_2 \overline{I}_1 I_0$ , 仅 LD 一个, 就需用三个反相器、一个四输入端与门。无疑, 这样逻辑势必复杂。

可见, IS 逻辑严密的设计不是不可以, 而是过于庞杂。看来, 简化是必要的。

简化的方法如下:

将(1)式第一项增下列五条指令: NO-PO、NOPF、JMP、RTN、SKZ。即将九条输入指令扩为除输出指令 STO、STOC 外的所有十四条指令。得下式:

$$LD + LDC + AND + ANDC + OR + ORC + XNOR + IEN + OEN + NOPO + NOPF + JMP + RTN + SKZ = STO + STOC \quad (2)$$

STO、STOC 是一位微处理器仅有的两条输出指令。只有执行输出指令, 才有可能输出写脉冲 W, 除此之外所有指令均不会有写脉冲输出。这里仅从指令范围而言, 不考虑时间, 可得:

$$STO + STOC = W \quad (3)$$

因此(1)式可简化为:

$$DIS = \overline{W} \cdot \overline{X}_1 \cdot DC_I \quad (4)$$

根据以上分析, (4)式已从九条输入指令扩展至十四条指令, 除输出指令 STO、STOC 外的所有指令。这样, 不仅执行输入指令, 而且空操作、转移、返回、零跳指令同样也从 IS 输入信息。就是说, 非输入指令 IS 也在输入, 以致使 D 载送的常为输入信息。而且常常不是输至 ICU 的真正想得到的输入数据。不过, 必须指出, 这只是造成 D 上信息混乱。由于 ICU 内部逻辑十分严密, 所有这些非真正的输入信息绝不可能进入 ICU 内部运算器或寄存器。

下面讨论 OL。OL 的逻辑设计比较复杂, 除了片选 CE 外, 还要考虑写读控制 W/R、写禁止 W.D。其真值表①、②、③见图(二)。其中真值表①表示在什么情况下复位、保持、选中。只有选中时才见真值表②、③。

真值表① 复位片选特性

| 输入端   |        |        | 输出端    |
|-------|--------|--------|--------|
| ReSet | CE     | 其它输入端  | Q0~Q7  |
| 1     | $\phi$ | $\phi$ | 0      |
| 0     | 0      | $\phi$ | 保持原有状态 |
| 0     | 1      | 见表2.3  |        |

• 当CE = 0, D端总为高阻抗

真值表②

地址选通特性

| A2 | A1 | A0 | Q <sub>n</sub> |
|----|----|----|----------------|
| 0  | 0  | 0  | Q <sub>0</sub> |
| 0  | 0  | 1  | Q <sub>1</sub> |
| 0  | 1  | 0  | Q <sub>2</sub> |
| 0  | 1  | 1  | Q <sub>3</sub> |
| 1  | 0  | 0  | Q <sub>4</sub> |
| 1  | 0  | 1  | Q <sub>5</sub> |
| 1  | 1  | 0  | Q <sub>6</sub> |
| 1  | 1  | 1  | Q <sub>7</sub> |

真值表③ 读写特性

| W/ $\bar{R}$ | W·D    | 输出状态                      |
|--------------|--------|---------------------------|
| 1            | 0      | 写状态 Q <sub>n</sub> = D    |
| 1            | 1      | 写禁止 Q <sub>n</sub> 保持原有状态 |
| 0            | $\phi$ | 读状态 D = Q <sub>n</sub>    |

图(二) OL真值表

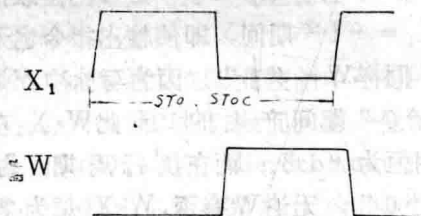
先说OL<sub>①</sub>。

OL<sub>①</sub> 仅作为系统的输出通道, 它只具有写功能, 接受来自D上的信息写入内部锁存器。因此只作为输出器件。按上述片选三条件, CE的代数式如下所示:

$$CE = (STO + STOC) \cdot \bar{X}_1 \cdot DC_0 \quad (5)$$

(5)式中, STO + STOC表示指令范围; 两条输出指令中任一条, 均可选中OL。 $\bar{X}_1$ 与DC<sub>0</sub>如前所述。

(5)式还可简化。当允许输出时, 执行输出指令, 在ICU Write端产生写脉冲W。W在X<sub>1</sub>的下降沿产生, 至上升沿结束, 脉宽为X<sub>1</sub>周期的二分之一。见图(三)。写脉冲用来禁止数据输入, 并用来打开OL, 以便将D上RR内容写入到某一指定的锁存器Q中去。



图(三)

可见, 将指令与时间两项一起考虑, 可得。

$$(STO + STOC) \cdot \bar{X}_1 = W \quad (6)$$

因此(5)式可简化为;

$$CE = W \cdot DC_0$$

根据真值表③, 在写操作时, W/ $\bar{R}$ 必须为“1”, W·D为“0”。上面已经分析过, 写脉冲是在X<sub>1</sub> = “0”时发生, 即W = “1”时, X<sub>1</sub> = “0”。显而易见, W/ $\bar{R}$ 、W·D即是W与X<sub>1</sub>。

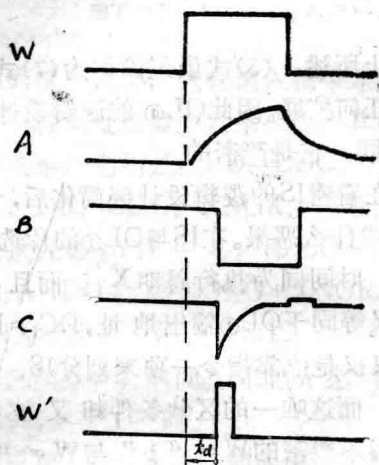
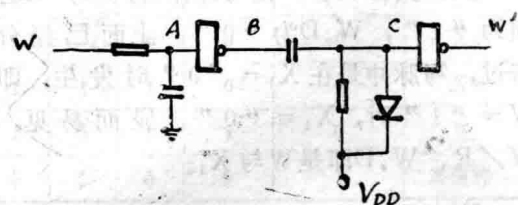
$$W/\bar{R} = W \quad (8)$$

$$W \cdot D = X_1 \quad (9)$$

综上所述, (5)式虽经简化为(7)式, 但并没作任何扩展。因此OL<sub>①</sub>的逻辑设计与IS OL<sub>②</sub>不同, 它是严密的。

现在看看IS的逻辑设计经简化后, 会给OL<sub>①</sub>造成什么恶果。在IS与OL<sub>①</sub>的片选诸项条件中, 时间同为执行周期 $\bar{X}_1$ ; 而且IS输入地址又等同于OL<sub>①</sub>输出地址, DC<sub>1</sub> = DC<sub>0</sub>; 余下的仅仅是依靠指令一项来划分IS、OL<sub>①</sub>的片选。而这唯一的区分条件却又被扩展了, 被很不严密的 $\bar{W}$  = “1”与W = “1”所替代。另外, 存储器自接收地址信息到数据输出存在着一定的延迟时间, 在此延迟时间内, 存储器提供的指令操作码、地址码全不稳定。正由于这些原因, 造成D上的数据混乱, 常载送不该载送的输入信息, 甚至执行输出指令时, 不是将RR的内容输出至OL的锁存单元中去, 而是将杂乱无章的不该输入的数据直接输出至OL, 造成出错。由此可见, 正常的运算结果不得输出, 一切努力全付之东流。这种不正常现象非去除不可。

那末,如何消除呢?正确方法是:输出指令的执行周期一开始,先关闭IS的输入,使它们的数据输出端Z全呈高阻,然后将RR的内容输至D,待稳定后再写入OL<sub>①</sub>中去。这里必须有时间上的先后。显然不能采用W同一上升沿既使IS关闭又同时打开OL<sub>①</sub>,应该利用W的上升沿先使IS的Z呈高阻,间隔一定时间后,再打开OL<sub>①</sub>,从而正确无误地将D上的RR内容写入指定的OL<sub>①</sub>锁存单元中。对W信号处理的线路与波形图见图(四)。



(四)

对W信号处理,前为积分电路,后部份为微分。二极管为嵌位二极管,保护反相器的输入端。关键是:W'比W的上升沿延迟了 $t_d$ , $t_d$ 必须大于IS的延迟,必须大于存贮器的最大存取时间。同时,W'的脉宽必须大于OL的延迟时间。还须指出:W经反相后的 $\bar{W}$ 用于(4)式,(7)式(8)式两式中的W应用W'。

现讨论OL<sub>②</sub>。OL<sub>②</sub>是兼有中间结果暂存功能的输出通道,它除了作为输出器件具

有OL<sub>①</sub>的写功能外,还要将原输出的中间结果再经D送回ICU继续参加运算。因此还具有读功能,兼作输入器件。简言之,OL<sub>②</sub>既能接受输出数据又能提供输入信息。兼有IS、OL<sub>①</sub>的功能。所不同的是IS输入的是外来信息,OL<sub>②</sub>输入的是原输出的暂存在它所含的锁存器Q中的信息。

正因为OL<sub>②</sub>输入输出指令都须涉及,它的CE应为:

$$\begin{aligned} CE &= [(LD + LDC + AND + ANDC + OR + ORC + XNOR + IEN + OEN) + (STO + STOC)] \cdot \bar{X}_1 \cdot DCI/O \\ &= [(LD + LDC + AND + ANDC + OR + ORC + XNOR + IEN + OEN) \bar{X}_1 + (STO + STOC) \bar{X}_1] \cdot DCI/O \end{aligned}$$

以上两式,均是严密的逻辑设计,显然太复杂。根据(2)、(3)式和(6)式,可简化为:

$$CE = (\bar{W} \cdot \bar{X}_1 + W) \cdot DCI/O$$

将上式括号内添加上 $\bar{W} \cdot X_1$ ,可得:

$$\begin{aligned} CE &= [(\bar{W} \cdot \bar{X}_1 + \bar{W} \cdot X_1) + W] \cdot DCI/O \\ &= [(\bar{W}(\bar{X}_1 + X_1) + W)] \cdot DCI/O \\ &= (\bar{W} + W) \cdot DCI/O \\ &= DCI/O \end{aligned} \quad (10)$$

添加项 $\bar{W} \cdot X_1$ 是这样的:输入指令无写脉冲,W="0",即 $\bar{W}$ ="1";在取指令周期 $X_1$ ="1"期间,即使输出指令也无写脉冲,同样 $\bar{W}$ ="1"。因为写脉冲W是在 $X_1$ ="0"期间产生的。因此 $\bar{W} \cdot X_1$ 在取指周期恒为"1";而在执行周期,由于 $X_1$ ="0",无论W有否, $\bar{W} \cdot X_1$ 总为零。简言之,添加上 $\bar{W} \cdot X_1$ ,在取指周期等于逻辑加上"1",在执行周期逻辑加上"0"。

W/R、W、D同OL<sub>②</sub>,如(8)(9)两式所示,不再赘述。

综上所述,由于 $\bar{W} \cdot X_1$ 在取指周期恒为"1",OL<sub>②</sub>逻辑设计比IS更不严密,前者不仅输入指令自九条扩至十四条,而且时

间也自执行周期外又占用了取指周期,扩展到整个指令周期。片选条件指令、时间、I/O地址三项中只剩下最后一项。这样,不论十六条指令中任何一条,也不论取指周期还是执行周期,只要I/O地址符合,OL<sub>2</sub>便选中。至于是输出指令还是除输出指令外的所有指令,得视W有无来定。由于W只有输出指令的执行周期方呈“1”,其它均为“0”,因此OL<sub>2</sub>同样也常处于读状态,也即输入状态。D也常载送输入数据。

由于OL<sub>2</sub>的DC<sub>I/O</sub>不得等于IS的DC<sub>I</sub>和OL<sub>1</sub>的DC<sub>O</sub>,因此选中OL<sub>2</sub>时,IS、OL<sub>1</sub>肯定不选中。

由于OL<sub>2</sub>功能相当于RAM,平时只能常处于读状态,若处于写状态,则要破坏锁

存器的内容。

由于OL<sub>2</sub>也常处于输入状态,D也常载送不是真正输入的输入数据,因此,当执行输出指令STO或STOC,将信息输至OL<sub>2</sub>时,只有当RR内容确实输至D上,并且确实稳定了,方可将它写入OL<sub>2</sub>中。从指令时序波形图可知,RR内容送至D是在X<sub>1</sub>下降沿,尚须间隔一段时间待此信息稳定后,再写入OL<sub>2</sub>中。可见同样要将W信号处理一下,经延迟后的W'才可能作为W/R信号。

从以上分析不难得出结论:鉴于IS、OL<sub>2</sub>逻辑设计存在着不严密性,凡是用于OL的W的上升沿都要作适当的延迟,且保持一定的脉宽。

## 有关一位微处理机问题的问答

董伯明

在多次举办一位微处理机学习班以及接待过程中,经常遇到很多基本而又概念性较强的有关问题,如不明确,则会带来模糊概念,现将这些问题解答如下,以供参考,如有不同的见解,亦欢迎交流商榷。

1.一位机究竟是微处理器还是微处理机?为何称一位机

已构成系统能在工业控制中运行的一位机应当称为微处理机。我们称为一位微机系统OBMS(One Bit Microcomputer System);OBMS中主要系列芯片为14500工业控制单元ICU,14516程序计数器PC,14512输入选择器IS,14599输出锁存器OL。而ICU被称为OBMS中的微处理器。这是要加以区别的。

因为在OBMS中,数据总线DB(Data Bus)只有一条在其ICU的内部结构中,结果寄存器亦只是一位的,每一条指令只能完成

二进制的一位数值的处理,故而称为一位机。

2.一位机主要用途究竟是什么?

一位机主要作为工业控制器广泛地应用于各种工业控制系统。它可以实现顺序控制,条件控制,循环控制,定时,计数,脉冲输入控制等逻辑控制,又具有简单的算术运算和数值比较的功能,故而目前已在机械制造,冶金,电力,化工,煤炭,三废处理,建材,纺织,轻工等各个应用领域发挥作用。具有极强的生命力,在我国四化建设和技术改造事业中具有广阔的前景。

3.如何构成OBMS中的暂存器?

暂存器的主要功能是暂时存放系统运行过程中的中间结果,数据,以便在需要处理的时刻再取出来加以运算。从目前看,在OBMS中构成暂存器的方式有三种:

①利用14599本身就具有输出锁存的功能,加以一定的控制逻辑,完成暂存器的功

能, 因为14599的每一个输出端都有锁存器, 而联接数据总线的Data端又具有双向传输的功能, 故而中间运算结果可以写到其某一个输出端暂存起来, 一俟需要时, 又可通过读操作将暂存的信息通过Data端取回ICU进行处理。不过一般利用了该片14599作为暂存器的话, 其所有的输出端都作为暂存器用了, 一般不再当输出用, 这由系统控制逻辑而定。

②通过将系统的输入选择器14512的某一个输入端任意地和输出选择器14599的某一个输出端用导线连接起来, 就方便地形成了系统的暂存器, 然而此时该输入端就不能再作为系统的输入端使用了, 不然, 输入信息与暂存信息要产生混乱。不过输出端仍可作为系统的输出端加以利用。该形式的暂存器是通过输出指令将中间信息暂存于14599的形成暂存器的那个输出端, 需要时, 再通过输入指令从形成暂存器的那个14512输入端取回处理, 这种暂存器的特点是构成方便, 任意, 使用方便, 缺点是占用了输入端。

③利用随机存取存储器RAM构成系统暂存器, 因为RAM本身就能随机存取, 而OBMS中的每一个中间信息都只是一位, 故而一片任何结构的RAM, 都可形成非常多的暂存器, 既方便, 又灵活, 且价格便宜, 而且系统的暂存器越多的话, 就越能发挥软件的技巧, 在新推出的OBMS中, 越来越多地利用RAM来构成系统的暂存器了。

4.一位机是三总线还是四总线? 系统中的I/O地址线是怎么回事?

一般微机系统都称为三总线结构, 即控制总线CB(Control Buo), 地址总线AB(Address Buo), 数据总线DB(Data Buo)。一位机当然亦具有这三条总线, 然而因平时所画的系统结构示意图往往不画控制总线与地址总线, 而系统输入、输出端口的地址总线总是画在图中, 再加上OBMS的指令结构由二部分组成, 其后部分就是系统输入, 输出端口的地址码, 因而往往使人们引起

概念的混淆, 将输入、输出端口的地址总线误认为是系统的地址总线。因此, 为强调区别, 说明这个问题, 就提出了四总线概念, 加了一条输入、输出地址总线I/O AB。该总线上传递的信息仅仅表示系统输入、输出端口的地址码。而地址总线上传输的信息所指出的是存储器的地址码。这二条总线是一定要加以区别的,

5.请再说明一下交叉存贮和混合存贮的概念, 交叉存贮和混合存贮可能是OBMS中的一个特殊的概念, 是由OBMS中硬件与OBMS的指令格式而决定的, 大家已经清楚OBMS的指令前四位一定是操作码OPC, 而后四位以至更多位就是系统输入、输出端口的地址码, 并且OBMS的系统逻辑允许一条指令由二个节拍中形成, 先取出OPC, 再取出地址码, 然后拼成一条指令。如果OBMS的存储器是每单元字长四位, 那么就只能是二个单元内容形成一条指令, 前一个单元存放OPC, 后一个单元存输入、输出端口地址码, 这样就构成了交叉存贮的结构。然而这种结构的OBMS其输入、输出端口地址码最多只能 $2^4 = 16$ , 显得少了些。

如果OBMS的存储器是每单元字长八位, 那么可以在一个单元存放一条指令, 其前四位是OPC, 后四位是输入、输出端口地址码, 这样的实际效果和交叉存贮一样, 为了使系统能适应各种复杂工业控制系统, 必须扩展系统的输入、输出端口, 于是仍使二个八位单元的内容形成一条指令, 第一单元的高四位仍为OPC, 低四位是输入、输出端口的高四位地址码, 第二个单元存放的内容是输入、输出端口的低八位地址码, 这样, 当二个单元拼成一条指令时, 该指令对输入、输出端口的寻址能力达到了 $2^{10} = 1024$ , 是足够的了, 我们就称这种存放指令的形式为混合存贮。

6.一位机的程序结构是循环结构的道理。

大家从资料上都已看到, 在OBMS中典

型的程序结构是三种。IF~THON; IF~THEN~ELSE; WHILE。然而从微机宏观角度讲,这三种结构应当统归属于分枝程序结构,都是通过某一个条件的判断,决定程序的走向。在OBMS中,这三种程序结构又都可由循环的形式出现,这由OBMS中的三个特点决定的:

①系统存储器EPROM在刷新后,内容均为1,在写入新的程序后,多余的存储单元内容仍保持为1。

②在OBMS指令中,四位全1形成的OPC是NOPF,意味着空操作,执行这条指令,结果寄存器RR的内容不变,不影响指令的结果。

③OBMS的程序计数器PC复位时全零,指向系统存储器的第一个单元,当计数到最大值后,会自动地复零,指向系统存储器的第一个单元。

于是,当OBMS的存储器中写入新的程序以后,启动系统,PC指向存储器的第一单元,如该单元是程序,则就开始执行,如是空单元(全1),就空操作,直至PC自动进到存放程序的单元开始执行,执行到程序结束,如PC仍未计数到最大值,就仍执行空操作,直到计数到最大值一个循环结束PC自动复零,又指向存储器的第一个单元,开始了第二次循环。这就是OBMS的程序是循环结构的道理。(与计算机中程序的循环结构三要素有区别)。

7.为何有的OBMS是16条指令,而有的是20条指令,甚至更多的指令?

指令操作码真正的本质亦是一组二进制的数而已。在OBMS中,ICU的指令寄存器输入端是四位,可以形成 $2^4=16$ 种不同的二进制数的组合,因而OBMS的基本指令有16条。然而,大家已经知道OBMS中的16条指令中,有二条NOPO, NOPF是空操作,还有如JMP, RTN亦不改变RR的内容,故而可以利用这几条指令,与系统中其它的时序组合译码后,完成系统中一些特殊的功能,

故可将这些新的时序组合称为新的指令,因此,各单位在研制OBMS时,根据不同的功能要求组成了一些新的指令,则就出现了各种不同OBMS具有不一样的指令数,不过,一些基本的指令都是一样的,不会改变的。

8.实现真正JMP功能为何一定要加地址锁存器?不加行不行?

在真正实现JMP功能时,实质上是将JMP指令的后面的二进制数值作为欲转移的地址码,在JMP指令时同时发出的一个正脉冲的控制之下,强行置入系统的计数器PC,使PC跳到欲转移的那个地址码。然而,JMP的这个同时发出的正脉冲宽度为一个指令周期,该脉冲电平如一直加在PC的置数允许控制端,则下一条指令出现时,该指令的二进制数同样会误进入到PC,于是就产生了指令与地址的混乱,通常称为“竞争”,达不到JMP的功能。因此一定要在PC之前加一个地址锁存器,该地址锁存器实质上是D触发器,PC的每一位预置输入端都对应加一只D触发器。所有的D触发器的CP端都连好后同PC的PE端连接,于是当执行JMP指令时,其后边的转移地址码在JMP正脉冲的控制之下,通过各个D触发器再进入到PC预置端,改变PC的计数值为转移地址,而D触发器是靠脉冲边沿起作用的,在脉冲电平时间是不起作用的,故而起到了锁定的作用,不会产生“竞争”现象,真正达到JMP的功能。

9.什么叫程序代真?

程序代真的概念是在产生符号指令后诞生的,大家知道计算机的硬件只认识二进制的代码,而符号语言,高级语言都是为了方便人们的书写程序才产生的。因此,为使计算机能执行程序,必须将符号语言,高级语言所编写成的程序译成计算机能执行的二进制代码,这一个翻译的过程就称之为程序代真。在OBMS中只有符号语言,然而在OBMS的程序输入中有二种形式,一种是用开关,一种是用键盘,用开关的二个状态来输入程序必

须是二进制代码，而用键盘输入程序则是16进制代码，故而应当视系统的不同而决定将符号程序代真为二进制或十六进制的代码。

10. 为何加到14599的W控制脉冲要加单稳延迟，而加到14512的W控制脉冲不要？

在OBMS中的信息是通过一位数据总线传递的，OBMS的基本16条指令可以分成三大类，即控制型指令，输入型指令，输出型指令，输入型指令将信息通过DB输入到ICU的RR，而输出型指令将RR的信息通过DB输出到系统的输出端口，信息都要通过DB；当然我们希望系统将信息输出到输出端口的过程之中不会遇到干扰的信息，当然也不希望在输出过程之中有输入信号混到DB上去。

这就需要在OBMS中加上系统控制逻辑，来控制输入选择器14512和输出锁存器14599的工作。在控制逻辑中有一个控制信息就是写脉冲控制信号W，来控制14512的Dis端与14599的W/R端，当W信号为高电平时，封住14512，使系统不能输入信号，而此时只要 $WD = 0$  14599为写状态，就可以将系统的信号输出到输出端口，为了保证输出的数据是稳定的信号，故当系统在输出指令的控制下产生写脉冲W时，先封住14512，再延迟一段时间后去打通14599，于是就将稳定的信息通过DB传输到系统输出端口。所以写脉冲W通过延迟后再加到14599的W/R端，增加了系统工作的稳定性。

## 一位机的前途

郝鸿安

集成电路（IC）正以非常快的速度发展着。特别是在80年以后，发展之快是人们5年前未曾预料到的。就微机用IC来说，自4004/8008起步，沿着8080→8085→8048→8086→80186/286/386前进，后起的有Z80→Z8000→Z80000以及6800→68000→68020。这些产品大都已有CMOS版的产品，都力争向LSI化、向低功耗化，特别是向系统化、成品化发展，功能扩大而使用简化是两个重要的发展方向。面对这种趋势，自1977年MOTA的14500和1978年1月GI的SBA这两种一位机出世，至今基本无进展。究其原因，一是国际上对一位机的兴趣不如多位机浓厚；其二是生产成本一位机与多位机相差甚小。在是否容易掌握方面，以基本的简易系统来看，无疑一位机容易理解掌握，但是稍微复杂化以后就不见得比多位机容易，复杂化将使一位机完全丧失优势。现在有不少四位机并不见得比一位机难很多，

同样成本基础上功能却大得多，使用也灵活且适应性强。实际上国际上占数量最大的正由过去的四位机转向八位机。

现在已到80年代中期，一位机不应再停留在14500这样许多IC片子结构基础上，而应当尽快向单片或少片化方向迈进了。生存在于发展，要以发展、更新来求得生存。其他产品都发展了，不能唯独让一位机停滞，停滞是无生命力的。现在是应当吸取GI公司SBA的经验，尽快把一位机单片化，以便使组装、应用简单化。下面介绍一下SBA作为抛砖引玉，提供借鉴，以便能产生出保持一位机简单易理解的优点在使用上大大简化的新一代一位机IC。

SBA是一种单片一位机用IC，SBA即Sequential Boolean Analyzer（顺序布尔逻辑解析器），通用仪器公司（GI）在1978年1月发表。具有24条指令，30个入出口公用，可以满足大多数小型工业控制用户。

图1是SBA的方框图，图2是引脚排列。采用DIP—40脚封装，用NMOS工艺使用双电源工作（+5V，+12V）。可以外接电容C、电阻R<sub>1</sub>/R<sub>2</sub>构成内时钟振荡，频率为f<sub>CLK</sub>=10~800KHz，当然也可以外加时钟同步工作。其中的问题目前均容易采用CMOS工艺来改进或克服。

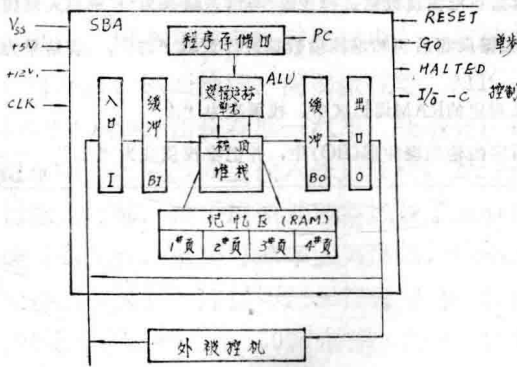


图1 SBA框图结构

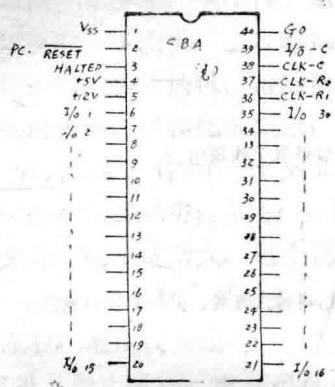


图2 SBA引脚排列

SBA有内存1024字的8Bit程序存储器，另外有一位逻辑运算单元（ALU），有一位16单元堆栈以及30单元为一页、共4页合计120单元的堆栈记忆区域（RAM）。

8 Bit的程序存储器分为高5位（H—5bit）和低3位（L—3bit）。H—5bit用作选择30个I/O地址以及堆栈记忆区地址，L—3bit用作指令码。用8进制表示就是0~377o，30个地址对应36o，指令码用0/37o对应。

若要求更多路I/O用，可以采用I/O多

路转换的方式来扩展，利用I/O—C端来控制I/O。I/O—C=“0”则I/O作为输出；I/O—C=“1”则I/O作为输入用。

Reset是对PC（程序计数器）复位用，当Reset=“0”则PC复为零且置为程序开头地址。程序可以执行再启动指令RST完成同样动作。

GO引脚是“单步/连续”控制。GO=“0”时，完成程序周期就停下来，仅走一步，并且同时产生中止预备信号HALTED输出；GO=“1”则程序连续执行，PC达到最大值后自行返回。

输入使用INPUT指令，写入锁存在30个I/O的之一之中。运算是利用ALU和堆栈，有两种运算型式。一是，输入到锁存器中的1bit或记忆堆栈区现行页中的1bit与堆栈顶单元作输入运算；另一种是，栈顶单元和栈的下一单元作为输入的逻辑运算。堆栈作为工作区域或累加器来使用，运算结果都是置于栈顶单元（相当14500中的RR），再用STORE指令可存放在记忆区现行页之一中（作暂存器用），或用OUTPUT指令置于输出缓冲器BO中。BO中的数据在程序周期结束时再锁入出口（相当14500中14599或14099）中，向外送出。

SBA和14500B相比较来看，前者内有程序存储器、输入/输出（I/O）口以及R/W（读/写）存储器（暂存器）RAM记忆区。所以SBA有利于用单片构成一个系统的突出优点。另外SBA有24种指令，采用堆栈运算方式，因此更容易处理相当复杂的逻辑运算。

另外GI还发表了开发用片子SBA—1，这是外加程序存储器的片子，采用DIP—64脚封装。

尽管SBA也未能获得普遍制造和使用，这也许是因1bit微机的致命弱点。而对我们来说，是如何使一位机走向使用简单化、低成本化，以一位机的优势去占领市场，要扬长避短。

SBA的指令如下表所示。

表 1

SBA 指 令 表

| 8 进 制 码 | 符 号     | 动 作                                         |
|---------|---------|---------------------------------------------|
| 01~36—0 | ANDIN   | 地址指定的输入数据和堆栈顶数据逻辑“与”                        |
| 01~36—1 | NANDIN  | 地址指定的输入数据反相后, 和堆栈顶数据逻辑“与”                   |
| 01~36—2 | ANDSS   | 地址指定的RAM记忆区数据和堆栈顶数据逻辑“与”                    |
| 01~36—3 | NANDSS  | 地址指定的RAM记忆区数据反相后再和堆栈顶数据逻辑“与”                |
| 01~36—4 | ASPI    | 地址指定RAM记忆区数据和堆栈顶数据进行逻辑“与”, 若结果为“1”则推入堆栈     |
| 01~36—5 | NASPI   | 地址指定RAM记忆区数据反相后再和堆栈顶数据进行逻辑“与”, 若结果为“1”则推入堆栈 |
| 01~36—6 | STORE   | 把堆栈顶数据存入地址指定的RAM记忆区中, 栈顶变为“1”               |
| 01~36—7 | OUTPUT  | 把栈顶数据存入地址指定的输出缓冲器(BO)中, 并把堆栈顶变为“1”          |
| 0 0 0   | RESTART | 再启动                                         |
| 0 0 1   | INVERT  | 堆栈顶数据反相                                     |
| 0 0 2   | PAGE    | 页交换                                         |
| 0 0 3   | HOME    | 返回本页                                        |
| 0 0 4   | PUSH 0  | 把“0”推入堆栈                                    |
| 0 0 5   | PUSH 1  | 把“1”推入堆栈                                    |
| 0 0 6   | PUSH C  | 把栈顶推入堆栈                                     |
| 0 0 7   | POP     | 弹出                                          |
| 3 7 0   | AND     | } 栈顶和次栈之间进行逻辑运算, 结果置于栈顶中                    |
| 3 7 1   | OR      |                                             |
| 3 7 2   | EXOR    |                                             |
| 3 7 3   | COMP    |                                             |
| 3 7 4   | PAND    | } 栈顶和次栈之间进行逻辑运算, 结果推入堆栈, 并且把栈顶变换为逻辑“1”      |
| 3 7 5   | POR     |                                             |
| 3 7 6   | PEXOR   |                                             |
| 3 7 7   | PCOMP   |                                             |

## MIC—400微型工业控制器在长春第一汽车制造厂 铸造分厂的应用

机械部自动化所 杨昌焜

### 1. 概述

#### 1—1 装置的类型和应用

MIC系列微型工业控制器是以CMOS—

位微处理器5G14500为核心的通用化工业控制装置, 分为MIC—100, —200, —300, —400四档, MIC—400是其中容量最大, 功

能齐全的一种。该系列装置均已通过鉴定,并能进行一定批量的生产,该装置除存储器外均采用上海元件五厂的CMOS中、小规模集成电路。MIC—400主要完成开关量的逻辑运算和控制,还具备计时、计数、变址、子程序、数值运算、比较等功能。可用于机械制造、冶金、电力、石油化工、建材、煤炭、纺织、轻工、军工等各种工业生产部门,用来实现多机群控和中、大规模生产自动线、巡回检测、简易数控等。MIC—400于84年5月在长春第一汽车厂铸造分厂铸造车间高压造型线上投入生产应用,至今运行稳定可靠,由于用微处理器代替了原有的硅逻辑元件,系统可障率大为降低,CMOS元件至今只有一片损坏,疵产量由原来的500箱左右,提高到650箱左右,维修工作量亦明显减少,取得了较大的经济效果。该高压造型线上有主机、下主机、落砂机、翻箱机、落箱机、合箱机、推箱机、插箱机、下降机、拉箱机、举箱机、上进箱机、下进箱机、推压铁机、提压铁机、落压铁机、铣浇口机、传送带等,动作比较复杂,共有输入160点,输出90点,计时、计数5路。铸造车间现场条件相当恶劣,如粉尘高、振动大、温度高、电压波动大、电干扰强烈、粉尘在3~4毫克/立方米,夏季控制室的温度达40℃左右,电压波动超过±10%,但MIC—400自投入生产运行,没有出过故障,可靠性、稳定性较高,抗干扰能力强,通过每日两班生产(铸造环境下)的实际运行的考验,已深深得到维修工人和生产工人的欢迎,为微处理器用于大型生产线的控制取得了信誉。

### 1—2 主要技术参数

MIC—400的主要技术参数如下:

- (1)输入点数 128点—512点
- (2)输出点数 128点—512点
- (3)存储器容量  $2^k \times 16$  位 —  $8^k \times 16$  位 (RAM或EPROM)
- (4)硬件计时计数 每板十进制C181四

位,可灵活组合。十进制C180四位,可灵活组合模拟延时四路。时标0.1秒,1秒,6秒,10秒四种,可任意选择。功能板可扩展。

(5)存储器式计时计数 每板64路,可扩展,每路十进制二位,可灵活组合,64路可同时选用。

(6)输入、输出隔离方式:光电隔离。

(7)输出接口方式;

继电器(220V, 3A或380V, 2A)

无触点输出(24V, 1A)

### 2. 整机工作原理

#### 2—1 整机构成及方框图

MIC—400微型工业控制器的方框图如图1所示(存储器式计时、计数等不包括在内)。

MIC—400微型工业控制器由下列三大部份构成,即中央处理部件,外围通道部件及操作控制部件。

中央处理部件包括中央处理单元(CPU<sub>2</sub>)、存储器单元(MEM<sub>1</sub>)、变址、子程序单元(ISU)、及接口单元(IFU)组成。

外围通道部件包括输入单元(BIN),输出单元(BOT)及计时、计数单元(TC—U<sub>1</sub>及TCU<sub>2</sub>),中间变量存储单元等。

操作控制部件包括操作单元(OPU)及键盘编程单元(KPU)等。

中央处理部件各功能板有如下功能:

CPU<sub>2</sub>板上包含有执行运算、控制功能的一位微处理器5G14500,程序地址计数器5G14516及置数电路。节拍脉冲发生器,存储器片选译码器及节拍脉冲奇偶校验电路等。CPU<sub>2</sub>按照程序地址计数器指定的地址依次从存储器中取出指令送到一位微处理器中,并通过一位双向数据总线接收外部数据和向外传送逻辑运算结果。节拍脉冲发生器产生八个节拍对全机工作进行同步控制,有了节拍脉冲发生器使扩展指令及存储器式计时计数,数值运算等复杂功能得以实现。通