

通过 ISO9001 国际质量体系认证

DK—PC80 PC40

MF—PC24

可编程序控制器 使用手册



® 无锡中远工业自动化有限公司

无锡中远自动化研究所

目 录

第一篇 PC80 系列小型可编程序控制器使用手册

第一章 概述

一、系统描述	1
二、系统构成	1
三、PC80 可编程序控制器的技术指标及功能	2
四、PC80 可编程序控制器的特点	3
五、系统设计流程	4

第二章 PC80 型可编程序控制器

一、整体结构	6
二、微处理器(CPU)单元	7
三、输入输出系列模块	7
四、各模块型号说明	9
五、输入、输出单元继电器号分配	15
六、主机安装	16
七、外部连接	17
八、使用操作顺序	18
九、故障判断	18
十、电池更换	18

第二篇 PC80 指令系统及编程指导

第一章 继电器号及内部数据存贮器的分配

一、输入继电器	19
二、输出继电器	19
三、辅助继电器	19
四、特殊功能继电器	20
五、继电器号的分配	20

第二章 指令系统

一、指令汇总表	22
二、基本逻辑指令的使作	23
三、定时器及计数器指令	28
四、移位寄存器指令和步控制器指令	30
五、锁存器指令	34

第三章 编程指导	
一、基本梯形图的编程	36
二、基本电路应用实例	39
三、步进过程控制应用举例	48

第三篇 PRG—001 编程器使用手册

第一章 编程准备	
一、概述	51
二、PRG—001 编程器键和显示灯	51
三、PRG—001 编程器的功能	54

第二章 编程操作指导	
一、程序区及数据区的清除	55
二、程序步的调出与程序的显示	56
三、程序写入	56
四、程序更改	57
五、程序插入	58
六、程序删除	59
七、程序检索	59
八、程序从 RAM 写入 EPROM	60

第三章 监控功能	
一、输入、输出状态的监视	61
二、定时器的现时值监视	61
三、计数器的现时值监视	62
四、强制 ON/OFF	62

第四篇 PC40 小型可编程序控制器使用手册	63
-------------------------------------	-----------

第五篇 MF—PC24 超小型可编程序控制器使用手册	70
---	-----------

附表 1 程序的检验及错误类别	76
------------------------------	-----------

第一篇 PC80 系列小型可编程序控制器 使用手册

· 本手册向用户提供 PC80 可编程序控制器及该系列输入/输出模块的技术参数及安装指导。

· 用户在安装或使用本装置之前,请详读本手册。

第一章 概 述

一、系统描述

可编程序控制器 (PROGRAMMABLE CONTROLLER 简称 PC), 自七十年代问世以来, 不断更新换代, 发展极为迅速。七十年代末期, 国外许多厂家将微机技术引入 PC, 产生了新一代功能更强的 PC, 并向着大、中、小型系列化方向发展。有的 PC 机增加了数据通讯功能, 甚至还可以由各种层次的 PC 机, 或与通用微机构成分布式的控制系统。

从功能和规模上区分 PC80 系列可编程序控制器属于小型 PC 档次。系参考国外各厂家同类型产品性能后, 自行开发的国产化系列产品。它可以完成开关量和逻辑运算和控制、有计时、计数、移位、步控及在线监视等功能。对来自按钮、限位开关、传感器的输入信号进行检测, 根据输入信号状态而产生相适应的输出信号, 用以驱动电动机控制器、接触器、电磁阀、以及步进电机、指示灯等各类负载。

可编程序控制器使用灵活、方便。用户无需了解复杂的计算机软件, 经过短期训练就可自行编程。它采用通常使用的梯形逻辑图或布尔方程进行编程。用户若变更机械加工零件、自动流水线流程, 或更改旧的工艺流程、按新的工艺流程来加工时, 只须改变用户程序, 而不必进行硬接线的变动。这便是 PC 机和继电器控制所不同点之一。

PC80 系列可编程序控制器的设计、工艺优良, 使它具有较强的抗干扰性能, 对使用环境无特殊要求, 工作稳定可靠, 无故障试验 (MTBF) 已超过了 5 万小时, 在设计时, 充分考虑了性能价格比、选件的灵活性可维护性。用户可根据不同的需要任意选择和组合。因此要控制器具有较高的性能价格比和合理的结构形式。它适用于各种自动化设备、生产自动线。是电磁继电器控制盘理想的替代产品。并具有一定的数控功能。

二、系统构成

PC80 系列可编程序控制器目前主要有 3 种规格: 1、40 点, 2、80 点, 3、112 点, 它们的组成部份相似, 由以下四部分组成:

- 微处理器单位 CPU
- 编程器 PRG—001
- 输入/输出单位
- 电源单元

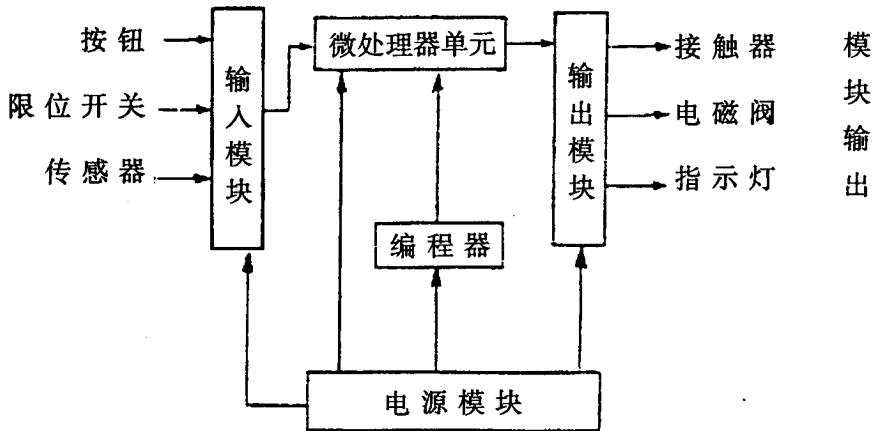


图 1—1 系统方框图

可编程序控制器能完成梯形逻辑图的逻辑控制功能。这种梯形逻辑图是包含计数器、计时器、数据锁存器、步控制器、步进马达控制器等等。并且在每个逻辑行里使用了多少个串联或并联的元件，原则上是没有限制。用户程序采用本控制器专门提供的编程语言编制，并由编程器送入。可长期保存在 RAM 中，或固化在 EPROM 中。

编程器的功能是把用户程序输入到控制器中。并同时能完成修改、编辑等任务，编程器为手持式的，通过一根扁平电缆和控制器相连接。对于编程语言的各条指令、数据等都标明在编程器的键盘上，采用不同颜色的键加以区别，通过数码管、发光二极管、指明程序的步号、指令符号及数据，关于编程器具体使用方法请参阅第三篇介绍。

输入/输出：它们用于完成控制器外部、内部信号之间的转换。输入模块将来自现场的状态信号（AC 或 DC、也可能是模拟量）转换成控制器可以处理的逻辑信号；输出模块将控制器发出的控制信号加以变换，用于驱动各种不同性质的输出负载。输入 0、1 号卡还可连接数码拨盘，用户可以通过拨盘开关来设置定时器的常数，而无须通过编程设定。它可以接 4 只拨盘，组成 2 组不同常数。

PC80 配置 2000 系列输入输出模块。每块板为 16 个 I/O 点，或 12 个 I/O 点。

三、PC80 可编程序控制器的技术指标及功能

· 技术指标

工作温度： $0^{\circ}\text{C} \sim +45^{\circ}\text{C}$

工作湿度：小于 85%（不结露）

存放温度： $-10^{\circ}\text{C} \sim 70^{\circ}\text{C}$

抗干扰：符合 IEC—801 标准、GB—6833 标准

MTBF：大于 50000 小时

交流电源： $220\text{V}/110\text{V}(-15\%: +10\%)$

耐地泄漏：不大于 3.5mA

主机芯片：8031

存储器类型：CMOSRAM（带电池保护）和 EPROM 或 EEPROM

程序步数：0 ~ 1000（或扩充到 2000 步）

工作方式;存贮程序、循环扫描方式

处理速度:小于 25ms/1k 字

输入/输出继电器:最大 112 个

辅助继电器:528 个,其中最具有掉电保护能力的为 256 个(包括内部继电器、定时器、计数器、移位寄存器、步控制器及锁存器,使用时由用户定义)

定时器:0.1 秒 ~ 1270 秒,设有三个时标:0.1 秒、1 秒、10 秒。定时时间可扩展。并具有 2 组 2 位数(十进制)的外部预置功能。定时器可以在 528 个辅助继电器内任意设定,定时器数量不限。

计数器;计数 1 ~ 127 次,可串联使用,具有 2 组 2 位数(十进制)的外部预置数功能。计数可以在 528 个辅助继电器内任意设定。计数器数量不限。

指令系统:

逻辑运算:AND、AND NOT、OR、OR NOT、(A)、(O)

输入输出:LD、LD NOT、OUT

定时:VAL、TIM

计数:VAL、CNT

移位寄存器:SR;8 位为一组,可串联使用

步控制器:SC;8 位为一组,可串联使用

主控制:MCS、MCR

锁存器:SET、RST

消除:CLR

编辑功能:

程序调出、程序更改、程序插入、程序删除、程序检索。

EPROM 写入。本机具有 EPROM 写入功能,不需另外附加任何设备。

监控功能:可监视输入、输出及内部继电器的现时状态及监视定时器、计数器现时值。

强制 ON/OFF,可通过编程器直接操作某一输出继电器的状态。

程序的检验及错误指示。

四、PC80 可编程序控制器的特点:

易于编程

以梯形逻辑图、布尔方程、逻辑图为基础,采用指令表语言编程,简单易学,它是一种计算机的专门语言、面向一般用户。所以 PC 的编程人员无须了解计算机复杂的程序语言,也可以极快地掌握、编制控制程序。

功能强

可编程序控制器内部具备了逻辑控制、定时器、计数器(定时器、计数器具有外接预置数码拨盘功能)、锁存器、主控功能、步控功能、移位功能等。特别步控功能在顺序控制中提供了极大方便。为了适应控制技术发展 PC80 又开发了步进马达控制、A/D, D/A, 通讯等功能并向简易数控配套发展。

编程灵活性大

在任何一条梯形逻辑行里,可用多少个串联和并联的触点不受限制。每对触点在程序里使用的次数不限。断电器状态带有记忆装置。这是继电器控制盘所不及的。各功能器(定时、计数、锁存、步控、移位)的位置及数量不受限制。

系列产品

PC80 可编程控制器的箱体可以从 24 点到 112 点,均为模块式插板结构,每块插板分别为 16 点或 8 点,供用户自选组合输入、输出。为了适应不同工业现场需要,备有交、直流输入、晶体管、继电器、固态继电器输出,为了进一步提高功能配备了 A/D, D/A 及步进马达控制器等智能化模块。

运行过程的监视

控制器在运行时,可通过和其相连的编程器的键盘和显示灯的状态来监视控制器的输入输出及内部状态、计数器、定时器、步控制器等的运行状况,实现在线监视。该功能在调试和判断故障位置时极有用。通过与输入输出指令相配合,它可以迅速测定故障在输入模块、输出模块,还是程序错误。

方便固化手段

在程序调试完以后,往往需要将程序固化、以防意外。PC80 在主机板上均有固化装置,可以直接固化程序,并检验固化是否正确,固化只需主机与编程器配合,不需要其他任何外部设备。

五、系统设计流程

系统设计过程可以用流程图表示(图 1—2)此流程图供用户设计系统时参考,对于首次使用可编程控制器的用户来说,则更需要他细考虑流程中的每一步骤。在设计过程中请注意以下几个问题。

1、首先对系统的工艺过程要熟悉,列出工艺流程、工艺流程各个状态图以及状态变换的条件。

2、根据工艺过程要求确定使用输入输出点的数量,并确定各个输入开关、输出负载的位置。

3、根据输入开关、负载的性能、工艺要求,选择输入输出板的种类,根据 I/O 的数据,确定使用 I/O 板的数量,一般讲 I/O 最好保留 10 ~ 20% 余量,便于调整。并适当调整 I/O 的位置。

4、PC 的输入、输出均有自己的特性,请注意它与输入开关、输出负载的匹配。特别是使用电子元件时应注意 PC 的带负荷能力、输入阻抗、极性等等。由于输入开关、输出负载的种类很多、技术参数各异、则需要选择合适的输入开关、输出负载与 PC 机相连。如果选择不合适将影响整个系统工作。

5、系统安装之前,对用户程序进行模拟调试。

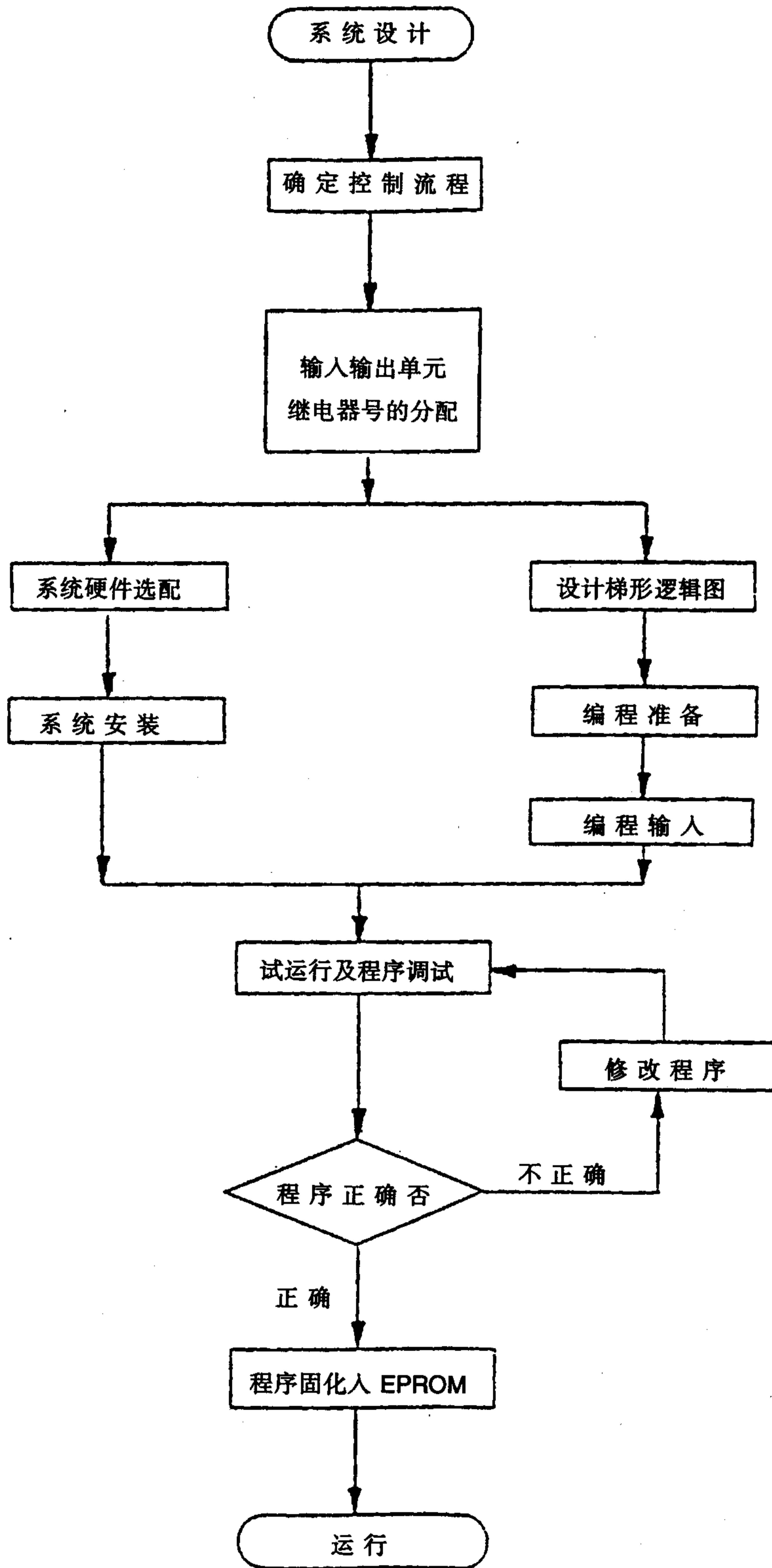


图 1—2 系统设计流程图

第二章 PC80 型可编程序控制器

一、整体结构

其外型见图 1—3, 本机分为基本型和扩展型, 基本型可带 80 点 I/O; 扩展型可带 112 点 I/O, 它们的位置分配见表 1—1, 基本型到 I/OV 槽为止, 总计共需七块模块 (电源一块、处理器一块、I/O 五块), 它们之间都采用接触可靠的镀金针状插头座和系统母线相连, 16 线的扁平电缆插头座通过电缆和 PRG—001 编程器相连。对输入输出单元的安装位置采用由用户自己确定的方法, 任何考虑程序问题。由于 I/O I 槽要兼作外预置数上可随意安装 (除了 I/O I 槽以外), 不必考虑程序问题。由于 I/O I 槽要兼作外预置数计时、计数器, 故该槽只能插入输入模块或专门的预置数模块。I/O 通道的继电器号分配则是从左至右依资助进行的, 见表 1—1。

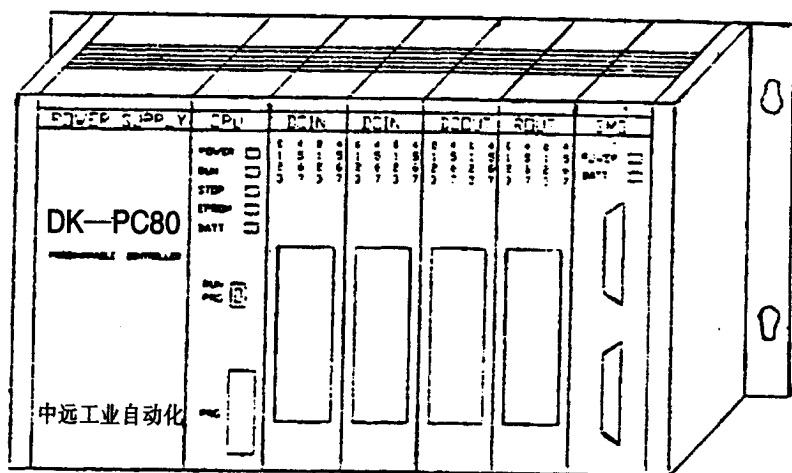


图 1—3 PC80 的外形

表 1—1

输入、输出单元继电器号分配表

		I/O I	I/O II	I/O III	I/O IV	I/O V	I/O VI	I/O VII
POWER	CPU	0.0	2.0	4.0	6.0	8.0	10.0	12.0
		?	?	?	?	?	?	?
		0.7	2.7	4.7	6.7	8.7	10.7	12.7
		1.0	3.0	5.0	7.0	9.0	11.0	13.0
		?	?	?	?	?	?	?
		1.7	3.7	5.7	7.7	9.7	11.7	13.7

例如某一输入模块插入 I/OI 槽内, 则其输入继电器号为 0.0 ~ 1.7 计 16 个输入点。某一输出模块输入 I/OII 槽内, 则输出继电器号为 4.0 ~ 5.7 计 16 点输出点。

二、微处理器(CPU)单元

微处理器单元能支持 112 个输入、输出点。其程序存储器采用可充电的镍镉电池作后备电源的 SRAM 或 EPROM。除了支持 112 个输入输出点外, 该单元内部还设有 528 个辅助继电器。(其中 256 个辅助继电器具有停电保持功能)供用户选用。它和其他输入输出模块一样, 采用引线针状插头座作为接插件和系统母板相连, 16 线的扁平电缆插头座通过电缆和 PRG—001 编程器相连, 所以它的安装及维修均十分简单。

图 1—4 为 PC80CPU 内部元件布置图。

* 1、D13—2764A EPROM 插座, 当用户需将程序写入 EPROM 时, 将空白 2764A 插入 D13 插座、执行 EPROM 写入操作程序, 可将用户程序保存进 EPROM, 从而可省去 EPROM 写入器。若将写有用户程序的 EPROM 2764A 插入 D13 插座, 当执行运行时, PC 将 EPROM 中程序送入 RAM 中的程序被清除, 执行新的用户程序。

* 2、S1—控制器状态开关(工作方式选择)。向上为运行状态, 向下为编程状态。

* 3、后备电池为 3.6V 镍镉电池, 向 SRAM 存储器供电, 它可以充电, 经常运行可编程控制器可不必要更换电池。更换电池须经过专门培训。

* 4、XSI—编程器连接插座。在面板上有一个小盖子, 打开盖子就可以看到该插座。它通过专用电缆与编程连接。

三、输入输出系列模块

PC80—2000I/O 系列模块具有如下一些特点:

每个模块有 16 个 I/O 点(或 12 个点)

单独插入式的模块结构

光电耦合的隔离方式或继电器的隔离方式

发光二极管作为状态指示

PC80—2000 输入系列模块设计目的是把来自按钮、限位开关或其它传感器的 AC 或 DC 信号变换为可编程序控制器可以处理的低压 DC 逻辑信号。

其输出系列模块把可编程序控制器的逻辑信号变换为用以驱动输出负载的 AC 或 DC 信号。

PC80—2000I/O 系列模块安装极为方便, 只需把模块插入机架上连接母板上的 I/O 槽内即可。I/O 模块在连接母板上的插槽位置没有硬性规定, 可随意选择, 只是一旦选定槽位, 就具有一定的地址号, 因此在编程时, 需注意各模块所在位置被赋予的地址号。

PC80—2000I/O 系列每个模块提供 16 个 I/O 点, 所以系统的设计有很大灵活性, 独立的模块型式使 I/O 点数可按 16 的增量向上扩展。因此备件费用和系统的扩展费用都比较节省, 同时也给维修带来方便。

PC80—2000I/O 系列模块, 采用光耦合器件作为隔离措施。为了避免外部电压的瞬变冲击, 设有内部保护从而有效地提高了整个系统的抗干扰能力。为了清晰地显示输入、输出模块的状态, 每个模块在其每一个输入输出点上设有一个 LED 状态指示灯来表示该点的工作状态。有输入信号, 绿色 LED 发光。有输出信号, 红色 LED 发光。

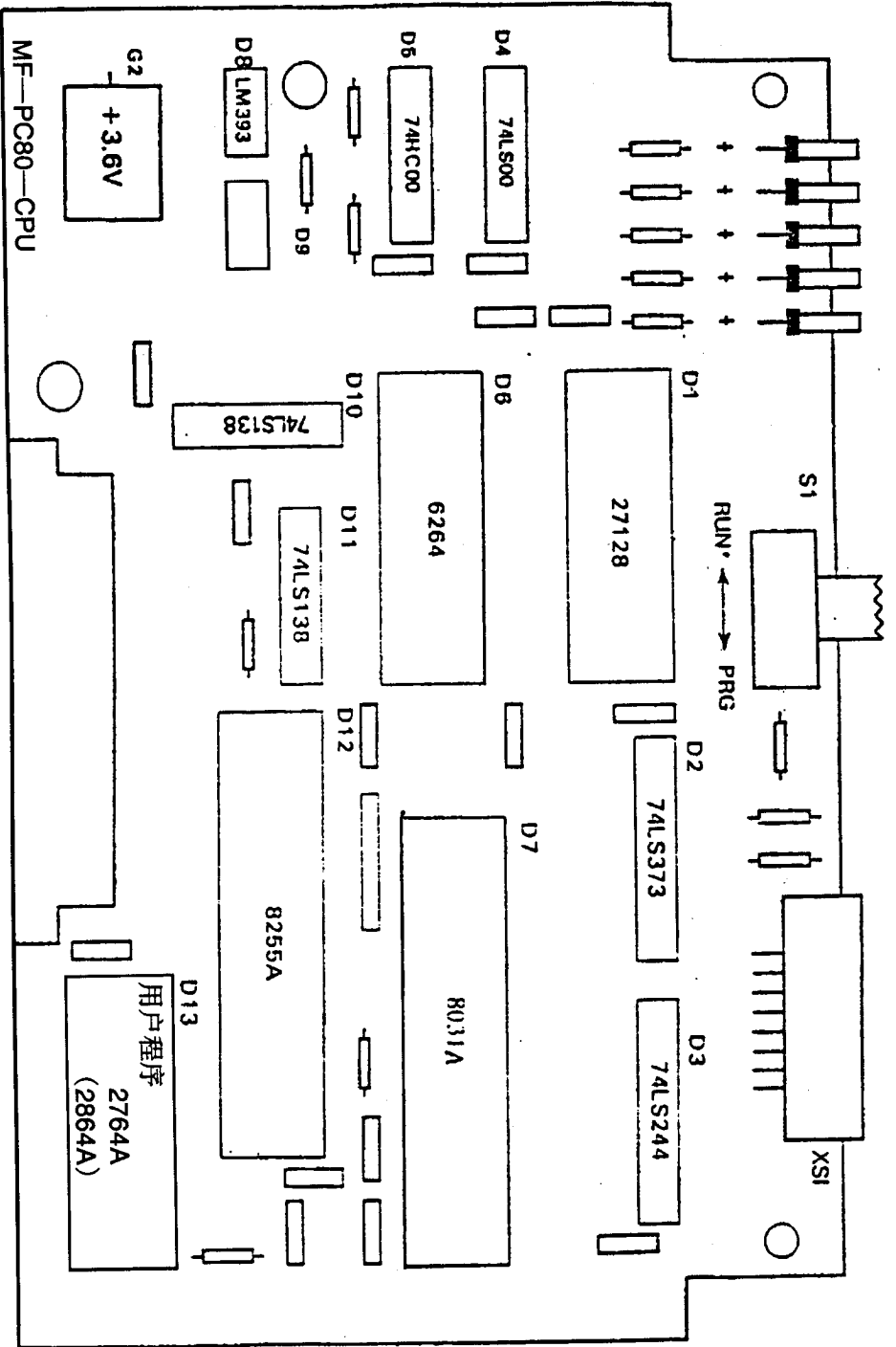


图 1—4 CPU 板内部元件布置图

用户使用时可以根据自己的需要选用有关 I/O 模块,与可编程序控制器连接的周连器件是多种多样,性能各有差异,故用户使用时应考虑 I/O 模块的特性,选用合适的周边器件。

0.0	C+
0.1	1.0
0.2	1.1
0.3	1.2
0.4	1.3
0.5	1.4
0.6	1.5
0.7	1.6
C-	1.7

四、各模块型号说明

本公司目前可以提供下列 I/O 模块,见表 1—2。若用户有特殊需要可设计专门模块,每块 I/O 模块由一个 18 点的端子板供外界接线。

例:某一 I/O 板插入紧靠微处理器(CPU)板,其各点的含义如图 1—5。它的使用方法各种模块有所不同,具体见后面说明,中间 16 点表示输入输出继电器的编号。具体编号见下一节。

表 1—2 PC80 系列简表

PC80—ACIN	AC 输入	12 点
DCIN	DC 输入	16 点
ROUT	继电器触点输出	12 点或 16 点
DCOUT	晶体管直流输出	16 点或 8 点
TCOUT	双向可控硅交流输出	12 点
DCIN/ROUT	混合输入、输出 专用模块	16 点 (8/8)

1、交流输入特性、输入信号分为二组，COM₁，COM₂ 分别为输入信号的公共线。

型 号	PC80—ACIN1	PC80—ACIN2
电 压	100V ~ 130V	200V ~ 240V
电压种类	AC	AC
输入阻抗	22K	44K
输入 信号	ON 通	外部输入点闭合，工作指示灯亮
	OFF 断	外部输入点断开，工作指示暗
隔离方式	光 电 耦 合	
电 路 图		
外部接线		

2、直流输入特性，输入信号为一组，C+ 为 +24V，供 PC80—DCIN1 作公共母线。
C- 为地线，供 PC80—DCIN 作公共母线。

型 号		PC80—DCIN1 (高有效)	PC80—DCIN (低有效)
电 压		24V	(+10%; -15%)
电压种类		DC (直流)	
输入阻抗		2K	
输入信号	ON 通	外部输入点闭合，工作指示灯亮	
	OFF 断	外部输入点断开，工作指示灯暗	
隔离方式		光 电 耦 合	
电 路 图			
外部接线			

接近开关输出选择电流型，若接近开关输出电流为流出型的应采用高有效即 DCIN1，若为流入型则采用低有效即 DCIN。

3、继电器输出特性，输出信号为二组，C1、C2 分别为二组输出信号的共用母线。
ROUT1 为 12 点输出，ROUT 为 16 点输出模块。

型 号		PC80—ROUT1	PC80—ROUT
电 压		12V ~ 220VAC	12V ~ 24VDC
电压种类		AC/DC	
电 流		2A (感性负载)	
母线最大电流		8A/Common	
输出 信号	ON 通	输出接通，工作指示灯亮	
	OFF 断	输出断开，工作指示灯暗	
隔离方式		电 磁 继 电 器	
电 路 图			
外部接线			

继电器输出由于断电、接通时容易产生火花及干扰，故在继电器触点并联灭弧吸涌电路，由于它采用阻容式，对交流负载有几毫安漏电流，对于特小负载使用时请注意。

4、晶体管输出特性，输出信号为一组，C+接+24V电源正极、C-接24V电源负极。DCOUT为16点输出，DCOUT8为8点输出。

型 号	PC80—DCOUT1	PC80—DCOUT	PC80—DCOUT8
电 压	24V		
电压种类	DC (直流)		
电 流	0.5A	2A	
母线最大电流	8A/Common		
输出 信号	ON 通	输出接通，工作指示灯亮	
	OFF 断	输出断开，工作指示灯暗	
隔离方式	光 电 耦 合		
电 路 图			
外部接线			

晶体管输出及下面双向可控硅输出均属于电子器件，其输出端过载以后容易损坏器件。在调试及使用时请注意。另外，每路输出均有快速保险丝经插座装在印刷板上，更换时插入即可。

5、双向可控硅输出特性，输出信号为二组， C_1 、 C_2 分别为二组的共用母线。

型 号	PC80—TCOUT	
电 压	110V ~ 220V	
电压种类	AC (交流)	
电 流	2A	
母线最大电流	8A/Common	
输出信号	ON 通	输出接通，工作指示灯亮
	OFF 断	输出断开，工作指示灯暗
隔离方式	光 电 耦 合	
电 路 图		
外部接线		

6、混合输入输入特性，继电器输出类型中， $C-$ 为输入信号的公共母线（低有效）， C 为输出继电器的公共线；晶体管输出型中， $C-$ 为输出负载外部电源公共负极， $C+$ 为输出负载外部 DC24V 电源正极；输入信号公共母线“COM”接到 16 点输入模块“ $C-$ ”端子上。

型 号		PC80—DCIN/ROUT	PC80—DCIN/DCOUT
输入 8点	电 源	24V	
	电压种类	DC (直流)	
	阻 抗	2K	
输出 8点	电 压	220V/24V	24V
	电压种类	AC/DC (交流/直流)	DC (直流)
	电 流	2A	2A
	形 式	继电器	晶体管
	隔离方式	电磁、光电耦合	光电耦合
	整板最大输出	8A/Common	
电路原理		本表所列输入，输出均为前面所列输入，输出电路重新组合，原理请见前面各表。	
外部接线			

所有的输出模块，均加快速熔断器，以防外界电器故障损害内部线路，但由于安装在印刷板上，装拆比较困难，希望外接一个熔断器以双重保护。

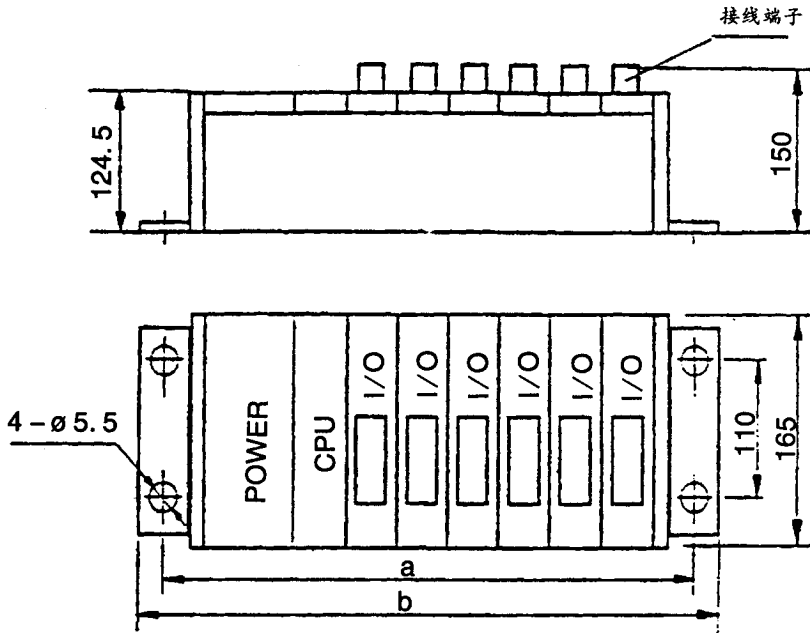
7、步进模块，A/D、D/A 模块特性详见另册。

五、输入、输出单元继电器号分配

本控制器对输入输出单元的安装位置采用由用户自己确定的方法，任何型号的输入输出模块在机架内 I/O 槽上可随意安装，不必考虑顺序问题，但作为外预置数计时、计数用的输入模块。只能插入 I/O 槽内。I/O 通道的继电器号分配则是从左至右依次进行的，如表 1—1。

六、主机安装

主机由电源、微处理器、输入输出模块及机架组成，机架从左到右依次为电源（占用二槽位置）、微处理器、输入输出模块。出厂时电源及微处理器已安装在内，输入输出模块可根据用户对输入输出的设定而选择其位置。机架有二种，即基本型和扩展型。当 I/O ≤ 80 点可采用基本型，超过 80 点（即大于 5 块输入输出模块）选用扩展型，比时可扩充至 112 个 I/O 点，安装尺寸见下图。



	a	b
7 槽机箱	326	342
9 槽机箱	396	412

注：尺寸单位：mm

图 1—6 PC80 外形尺寸

七、外部连续

1、电源的连接

打开 PC 最左边盖板，在下部有三个接线端子，见图 1—7，电源线由接线端子“1,2”引入，端子“3”为保护接地与机壳相连，用户必须可靠地用 1mm² 以上的黄绿线接至大地。电源输入电压标为“AC85V ~ 265V”时，只要将电源为 AC110V ~ 220V 之间的任意电压直接接入“1,2”端子上即可。

2、输入和输出的连接

本机的输入/输出接线端子在面板的外部,并可以拔插,上部有透明有机玻璃小盖板对端子起保护作用,盖板背面可插入一张纸条,作为标志牌用户可写入各端子的 I/O 编号或元件号。引线直接由接线端子引出。

3、输入/输出的走线要和动力线或高压线分开,尽可能远离,避免平行走线。

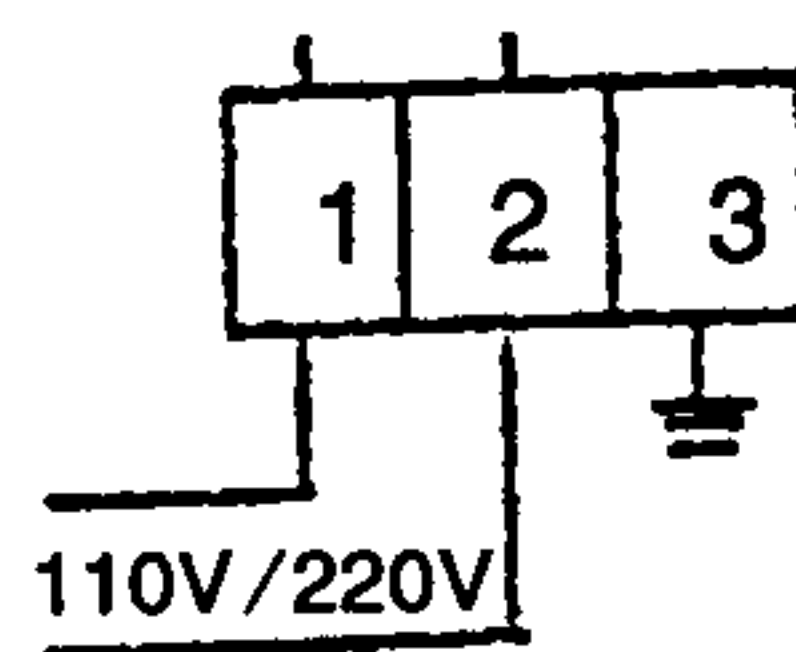


图 1—7 电源接线图

另外,每个 I/O 模块板上的小面板为 PVC 塑料面板,为方便用户,小面板上每个 I/O 点均对应用状态指示灯的指示,可极为方便地了解某被测(被控)的输入/输出点,乃至整个设备的运行状况。这在调试和维修时尤为方便。

为了安全起见,有时需要迅速切断可编程制器的输出负载电源,最好在输出负载电源上增加一个控制继电器由外接按钮开关控制,当按下 A1 输出负载电源与 PC 接通,按下 A2 输出负载电源与 PC 断开,从而停止负载输出,见图 1—8。

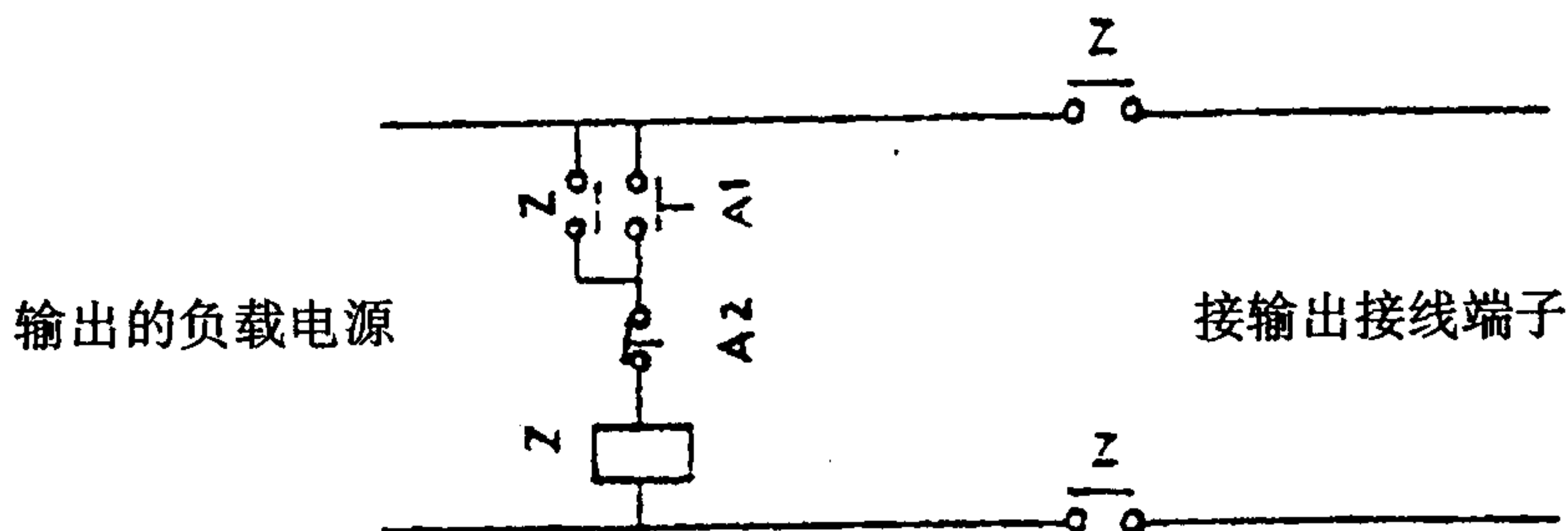


图 1—8 输出负载电源的连接

八、使用操作顺序

1、出厂时为了防止镍镉电池无效的消耗,由跳线座将其断开(见图 1—4,插头在“OFF”一方)。故使用前首先将跳线座接上(拔起插头,插向“ON”一方,电池方能向 RAM 有效地供电。

2、检查主机是否安装牢固,连线是否正确,接地线是否接好。

3、合上电源,CPU 面板上的“POWER”指示灯应点亮。如果状态开关在“PRG”位置,则“STOP”灯应亮,如果状态开关在“PUN”位置,则“PUN”灯应亮。面板上其余灯均不亮。

4、编程器可以插入 CPU 板上的 16 线插座内。插拔时带电、不带电均可,不影响主机

工作。主机运行时可以不插编程器。

5、装态开关置于“PRG”位置可进行编程,如程序已编好,状态开关位置于“RUN”位置,机器即可投入运行。I/O 板上的发光二极管显示输入输出的状态。

6、如欲停止运行,将状态开关位置于“PRG”位置,运行立刻停止。

7、模块的插拔和 EPROM 取放,都要在断电的情况下进行。

九、故障判断

1、输入部份的故障判断可由外中输入信号 ON, OFF 进行检查。状态开关置于“PRG”位置(机器不运行),人为的使某信号为 OFF 时,对应点的输入灯应灭,这说明输入信号是正常的,否则这一点就有故障。

a、在外部输入信号由 ON→OFF 时在输入点(即端子排)上测量,其电压是否有变化,若无变化,则说明外部断线。若输入电压有变化,则问题在输入板的输入部分。电平的参考点为 C+ 或 C-。

b、再检查输入部分,测发光二极管两端电压。若有电压,而指示灯不亮则发光二极管断开,否则光电耦合器前的部分有问题。

2、若输入信号正常,机器仍有故障,则可以编制一个简单程序;

```
LDX. X
```

```
OUTY. Y
```

进行多卡,多路试运行,若此时存在输出灯亮,一般来说微处理器板是正常,否则是不正常。

3、对输出灯不亮的卡,可进行编程强行输出,若输出类亮说明问题在输入板,否则问题在输出板。

4、若输出灯亮,输出端子无信号,则驱动部分有问题,应查熔断器是否断,若断更换熔断器,参考点为 C+ 或 C-。

若此时,存在输出信号,说明 PC 机正常,问题在外部连接。

5、若输入输出正常,而运行不正常,应检查程序是否正确。

十、电池更换

1、电池电压过低时,CPU 板上的“BATT”灯亮后,可在一周内更换电池。否则 RAM 中的信息在停电后将失去。

2、更换电池时,可将 CPU 模块板拔下,在后备电池两端并联外接电池或者电容(10 μ F 以上)。

3、换下旧电池,接上新电池即可,同时取下外接电池或电容器。

4、插好模块,接通电源,“BATT”灯灭,即表示更换完毕。

注意:拔插模块时均须断电操作。

第二篇 PC80 指令系统及 编程指导

本篇详细介绍 PC80 的指令系统（梯形图语言）请用户在编程之前详细阅读本篇有关内容。为了方便用户编程，本篇将介绍基本图型的编程方法及部分灵活应用范例。

第一章 继电器号及内部数据存贮器的分配

PC80 全部指令可分为两类：一类能单独起作用，另一类必须与继电器号组合起作用。这一章首先介绍继电器的种类、编号及功能。按其使用时状况可分为四类即输入继电器、输出继电器、辅助继电器及特殊功能继电器。

所谓继电器实质上是计算机内存内开辟一定的区间，存贮逻辑控制信息，通过逻辑控制不断地对它进行读写。由于计算机内存是二进制，而继电器恰恰也是二进制器件，为了方便和与传统继电器控制相适应，把这些内存称作继电器，内存中每一位相当于一个继电器线圈，当它置“1”时相当于线圈接通，当它置“0”时相当于线圈断开。对该位进行读则可得到继电器触点的状态。

一、输入继电器

输入继电器的作用是保证 PC 接收外部传感器或开关的信号，故输入继电器由外部信号驱动，而不能由程序中的指令来驱动它。PC 内部与输入端相连的输入继电器上有常开触点和常闭触点，每对触点在 PC 程序内可以重复选用，使用次数不限。输入回路的线圈按不同的输入模块分别适用于 DC24V、AC110V、AC220V。

二、输出继电器

输出继电器的作用是将 PC 的信号传给外部负载（电磁阀、指示灯、接触器等等）。它的工作状态是根据程序的执行情况而定。每个输出继电器的线圈不能重复使用，其触点可以在程序中重复使用，并不受次数限制。它的输出形式可根据输出模块不同的型号来选择。它拥有继电器触点输出、双向可控硅输出和晶体管输出等不同的输出形式。

三、辅助继电器

PC 具有许多辅助继电器，这些辅助继电器由 PC 内的程序指令来驱动，从而完成各种动作。它与输出线圈的驱动方式相同，同样每个辅助继电器的线圈不能重复使用，同样辅助继电器具有常开与常闭触点，这些触点可在 PC 内选择使用，使用次数不受限制。但是它不能直接驱动外部负载，只有通过输出继电器才能驱动外部负载。

辅助继电器按其掉电后状态分为两类：

1、一般型；这类继电器线圈被激励，则常开触点通、常闭触点断，当停电时，触点恢复常态。和通常的继电器类同。

2、停电保持型：这类继电器具有记忆功能，当停电时，它保持停电前的状态，以便 PC 恢复工作时重新被使用，它特别适合一些控制对象需要存贮停电前的条件。这是一般继电器很难做到的。

辅助继电器经过定义，可以构成一些专门的寄存器，例如：移位寄存器、步控寄存器、

定时器及计数器。这些专用寄存器可以在辅助继电器内选择,一旦选定不能移作他用。具体使用方法请参见后两章有关部份。

四、特殊功能继电器

本机还向用户提供了几个特殊功能继电器,供用户直接使用。

1、初始化脉冲继电器 79.0。在程序运行的第一个扫描周期产生一个宽度为一扫描周期方波。可用来作初始复位信号。(如图 2—1)

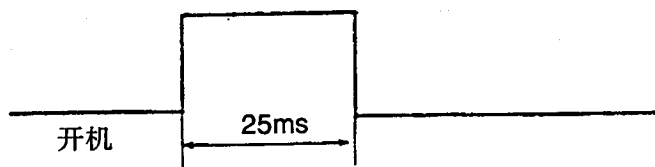


图 2—1 79.0 继电器信号图

2、50 毫秒时钟信号发生器 79.1。该电器提供用户一个周期为 50ms 的计时信号。(如图 2—2)



图 2—2 79.1 继电器信号图

3、全部输出禁止继电器 79.2。该继电器动作后,全部输出被自动切断。因此,若编制的程序使用 79.2 动作,可提供紧急停止动作。

注意;编程时程序不可使 79.0 及 79.1 动作,如 OUT7961、SR79 等命令都是非法的。

五、继电器号的分配

由于可编程控制器具有许多输入、输出、辅助继电器。为了准确地使用这些继电器需要给每个继电器分配一个继电器号,也称地址号。这些继电器的地址号安排采用矩阵式的排列,有下列几个约定。

1、以每 8 个继电器为一组,称之为卡(Card),相当于计算机中一个字节。

2、每一卡里,各个继电器占据的位置称之为位,相当于计算机中一位。

3、按上述 2 个约定,每个卡里有 8 位,分别用 0~7 这 8 个数字表示,称为位地址。位地址表示包含在一个卡内的每个继电器的位地址号。

4、所有继电器按 8 个一组分为若干个卡,这些卡按十进制从 00 开始编号,称为卡地址。它表示每 8 个继电器一组的卡地址号。

5、按上面的规定,所有继电器的完整地址号,采用卡地址和位地址综合表示,如 14.5,前二位表示卡地址 14,后面一位表示位地址 5。总的表示第 14 卡第 5 位的继电器。

6、输入、输出继电器的空地址号(即该号没有实际的输入、输出模块对应),可以作辅助继电器使用。

7、所有继电器地址号,内部数据存贮器的分配见下表。

表 2—1

位地址 卡地址	0	1	2	3	4	5	6	7	继电器 种 类	
0	0.1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	输入 输出 继电器	
1	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7		
2	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7		
~~~~~										
13	13.0	13.1	13.2	13.3	13.4	13.5	13.6	13.7		
14	14.0	14.1	14.2	14.3	14.4	14.5	14.6	14.7		一 般 辅 助 继 电 器
15	15.0	15.1	15.2	15.3	15.4	15.5	15.6	15.7		
16	16.0	16.1	16.2	16.3	16.4	16.5	16.6	16.7		
~~~~~										
45	45.0	45.1	45.2	45.3	45.4	45.5	45.6	45.7		
46	46.0	46.1	46.2	46.3	46.4	46.5	46.6	46.7		
47	47.0	47.1	47.2	47.3	47.4	47.5	47.6	47.7	停 电 保 持 型 辅 助 继 电 器	
48	48.0	48.1	48.2	48.3	48.4	48.5	48.6	48.7		
49	49.0	49.1	49.2	49.3	49.4	49.5	49.6	49.7		
50	50.0	50.1	50.2	50.3	50.4	50.5	50.6	50.7		
~~~~~										
74	74.0	74.1	74.2	74.3	74.4	74.5	74.6	74.7		
75	75.0	75.1	75.2	75.3	75.4	75.5	75.6	75.7		
76	76.0	76.1	76.2	76.3	76.4	76.5	76.6	76.7		
77	77.0	77.1	77.2	77.3	77.4	77.5	77.6	77.6		
78	78.0	78.1	78.2	78.3	78.4	78.5	78.6	78.7		
79	79.0	79.1	79.2	79.3	79.4	79.5	79.6	79.7	特 殊	

继电器地址号及内部数据存贮器的分配表

8、带停电保持的辅助继电器，掉电后其状态保持，但当控制器开关从 RUN 拨到 PRG 时，所有停电保持继电器复位到初始态（OFF）。



## 第二章 指令系列

PC80 可编程序控制器的指令系统采用指令表语句形式,学习方便,使用简单。

### 一、指令汇总表

表 2—2

序号	指令字	梯形图符号	操作对象	功 能
1	LD		X,Y,M,T,C	常开触点逻辑开始
2	LD NOT		同 上	常闭触点逻辑开始
3	AND		同 上	常开触点逻辑与
4	AND NOT		同 上	常闭触点逻辑与
5	OR		同 上	常开触点逻辑或
6	OR NOT		同 上	常闭触点逻辑或
7	(A)		同 上	中间贮存结果和
8	(O)		同 上	现结果逻辑与 中间贮存结果和 现结果逻辑或
9	OUT		Y,M	输出
10	TIM		T	定时器动作
11	CNT		C	计数器动作
12	VAL		/	定时、计数设定值
13	SC		Y,M	步控制器工作
14	SR		Y,M	移位寄存器工作
15	SET		Y,M,T,C	锁存器置位
16	RST		Y,M,T,C	锁存器复位
17	MCS		/	主控母线开始
18	MCR		/	主控母线结束
19	CLR		X,Y,M,T,C	卡的内容清除
20	END		/	程序结束

X:输入继电器

Y:输出继电器

C:计数器

M:辅助继电器

T:定时器

## 二、基本逻辑指令的使用

### 1、LD、LD NOT、OUT

LD 为“取”指令 LD NOT 为“取反”指令,分别表示电路开始的常开、常闭触点

OUT 为输出指令,表示继电器线圈驱动

这两条指令后均需跟编码。LD、LD NOT 后面跟随某一继电器的触点,这继电器可以是输入继电器、输出继电器、辅助继电器、专门寄存器,也可以是特殊继电器。OUT 后面跟随某一继电器线圈,它可以是输出继电器,也可以是没有定义过的辅助继电器,而且这些继电器线圈在整个程序中只能使用一次。

指 令		备 注
指令符	编码	
LD	1.0	LD 作为电路开始
OUT	20.0	20.0 作辅助继电器
LD NOT	1.0	
OUT	20.1	
LD	20.0	
OUT	8.2	8.2、8.3 为输出继电器
OUT	8.3	多重输出

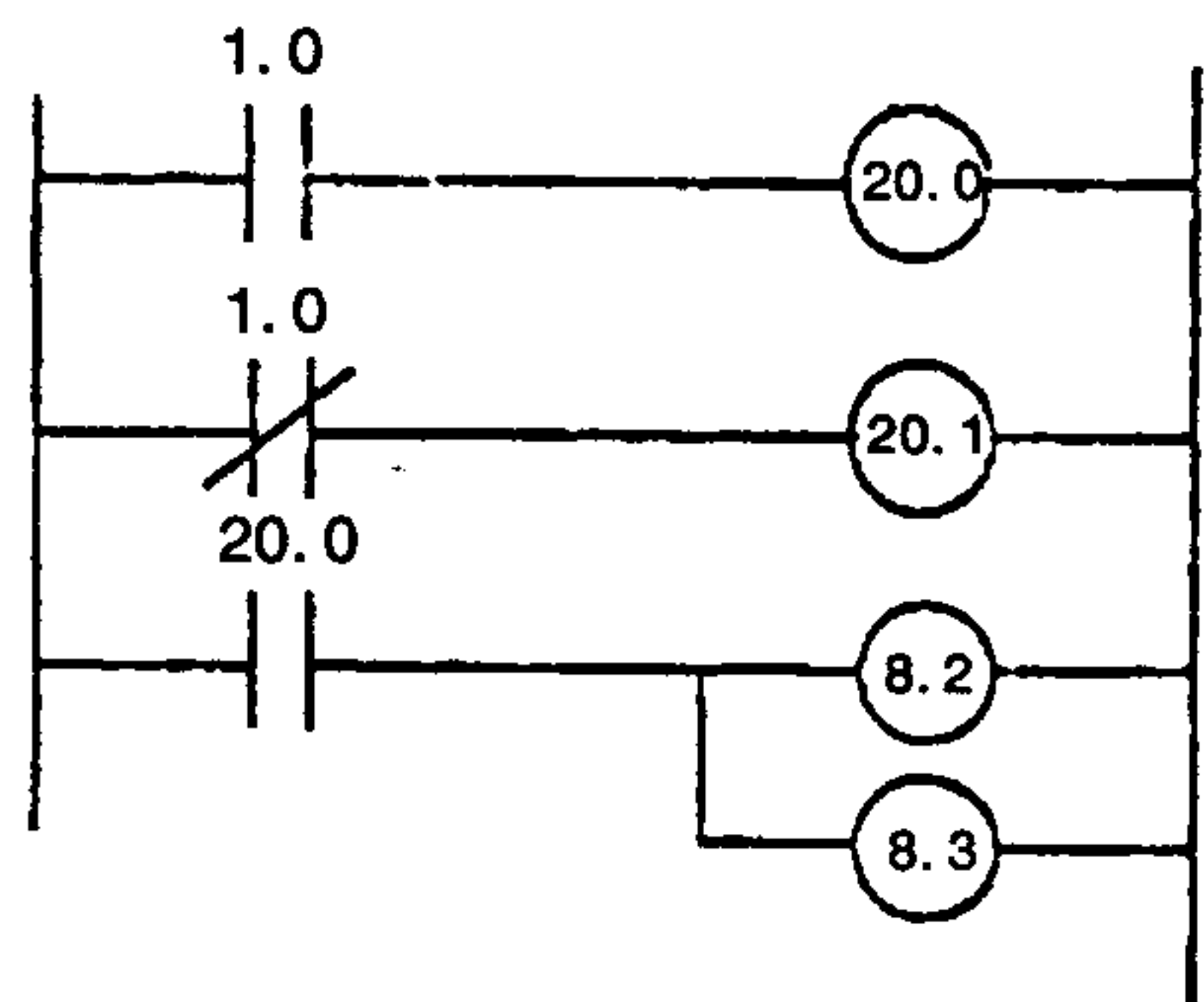


图 2—3

说明: 1. LD 和 LD NOT 指令用来作为电路开始(与左连母线连接)或和(A)、(O)连用时作为分支电路开始。(关于这方面的使用及含义见(A)、(O)指令说明)

2、OUT 输出指令作为对外部输出、内部输出。外部输入不能使用这条指令。需要并联多路输出时可以连续使用。OUT 指令也表示一电路结束,故它一定连接右连母线。

3、如图 2—3 所示, 1.0 为输入继电器, 8.2、8.3 为输出继电器, 20.0、20.1 为辅助继电器。当连接 1.0 继电器的外接开关接通时,也就是输入信号为“1”它的常开触点闭合、常闭点断开。第一电路常开触点 1.0 闭合, 20.0 线圈通电; 第二电路常闭触点 1.0 断开, 20.1 线圈断电; 第三电路由于前面 20.0 线圈已通电, 常开触点 20.0 闭合 8.2、8.3 线圈通电。对 8.2、8.3 的外电路接通, 有输出。当 1.0 开关断开, 第一电路常开触点断开, 20.0 线圈断电, 第二电路常闭触点 1.0 闭合, 20.1 线圈通电, 第三电路由于 20.0 线圈断电, 常开触点 20.0 断开, 8.2、8.3 线圈断电, 对外输出断开。

### 2、AND 和 AND NOT

AND(与指令): 常开触点串联指令。

AND NOT(与反指令): 常闭触点串联指令。

第二条指令均为双字节指令, 是指令符及编码组成, 编码为某一触点的代码, 它可以是输入继电器、输出继电器, 也可以是辅助继电器。

指令		备注
指令符	编码	
LD	2.0	触点串联
AND	2.1	
AND NOT	2.2	
OUT	9.0	连续输出
LD	3.0	
OUT	9.1	
AND	3.1	
OUT	9.2	
AND OUT	3.2	
OUT	9.3	

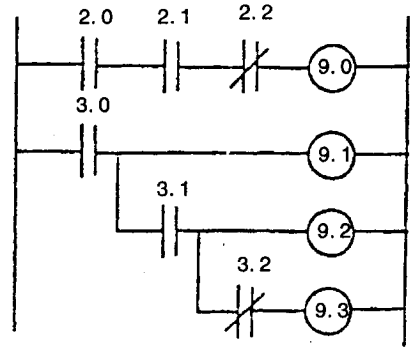
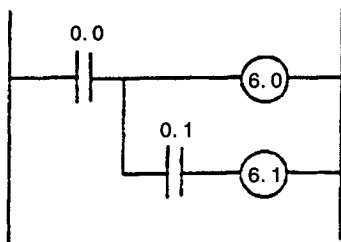


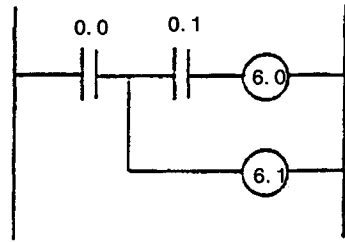
图 2—4

说明:

- 1、AND 和 AND NOT 指令用于触点的串联,串联触点的数量不限。
- 2、连续输出是指 OUT 指令后通过触点再输出其它线圈,而该触点并不是返回母线,但再输出线圈必须放在下面,不能放在上面(见图 2—4、图 2—5、图 2—6)
- 3、如图 2—4 所示,当输入继电器 2.0、2.1 接通,2.2 不动作,则 9.0 支路形成通路,我们认为 9.0 线圈通电,有输出,中间若有一个触点不通则 9.0 线圈断电无输出。



正确  
图 2—5



不正确  
图 2—6

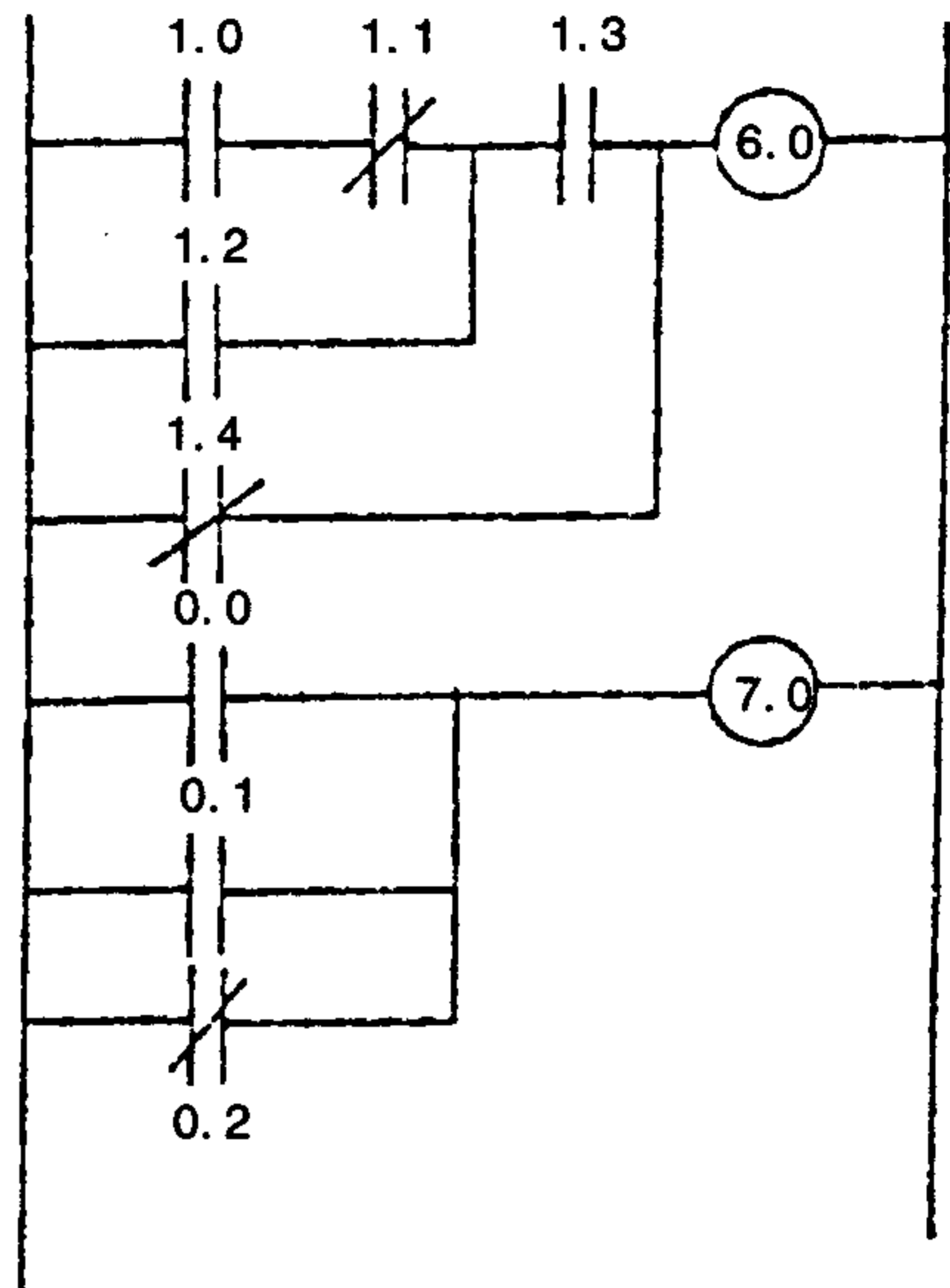
### 3、OR 和 OR NOT

OR(或)指令:常开触点的并联指令。

OR NOT(或反指令);常闭触点并联指令。

第二条指令均为双字节指令,即由指令符及编码组成,编码为某一触点的代码,可以是输入继电器、输出继电器、辅助继电器或特殊继电器。

指令		备注
指令符	编码	
LD	1.0	
AND NOT	1.1	
OR	1.2	
AND	1.3	
OR NOT	1.4	
OUT	6.0	
LD	0.0	
OR	0.1	
OR NOT	0.2	
OUT	7.0	



说明:该指令可以连续使用。次数不限。

4、(A)、(O)

(A)(与块状指令):复杂支路的串联指令。

(O)(或块状指令):复杂支路的并联指令。

这二条指令为单字节指令,只有指令符,没有编码。

图 2—7

指令		备注	
指令符	编码		
LD	2.0	a分支路	
OR	2.1	模块 a、b 串联	
LD	2.2		b分支路
OR	2.3		
(A)		模块 a、b 串联	
LD	2.4	c分支路	
OR	2.5	模块(a,b),c 串联	
(A)			模块(a,b),c 串联
OUT	6.0		
LD	3.1	d分支路	
AND NOT	3.2	模块 d、e 并联	
LD	3.3		e分支路
AND	3.4		
(O)		模块 d、e 并联	
OUT	6.1		

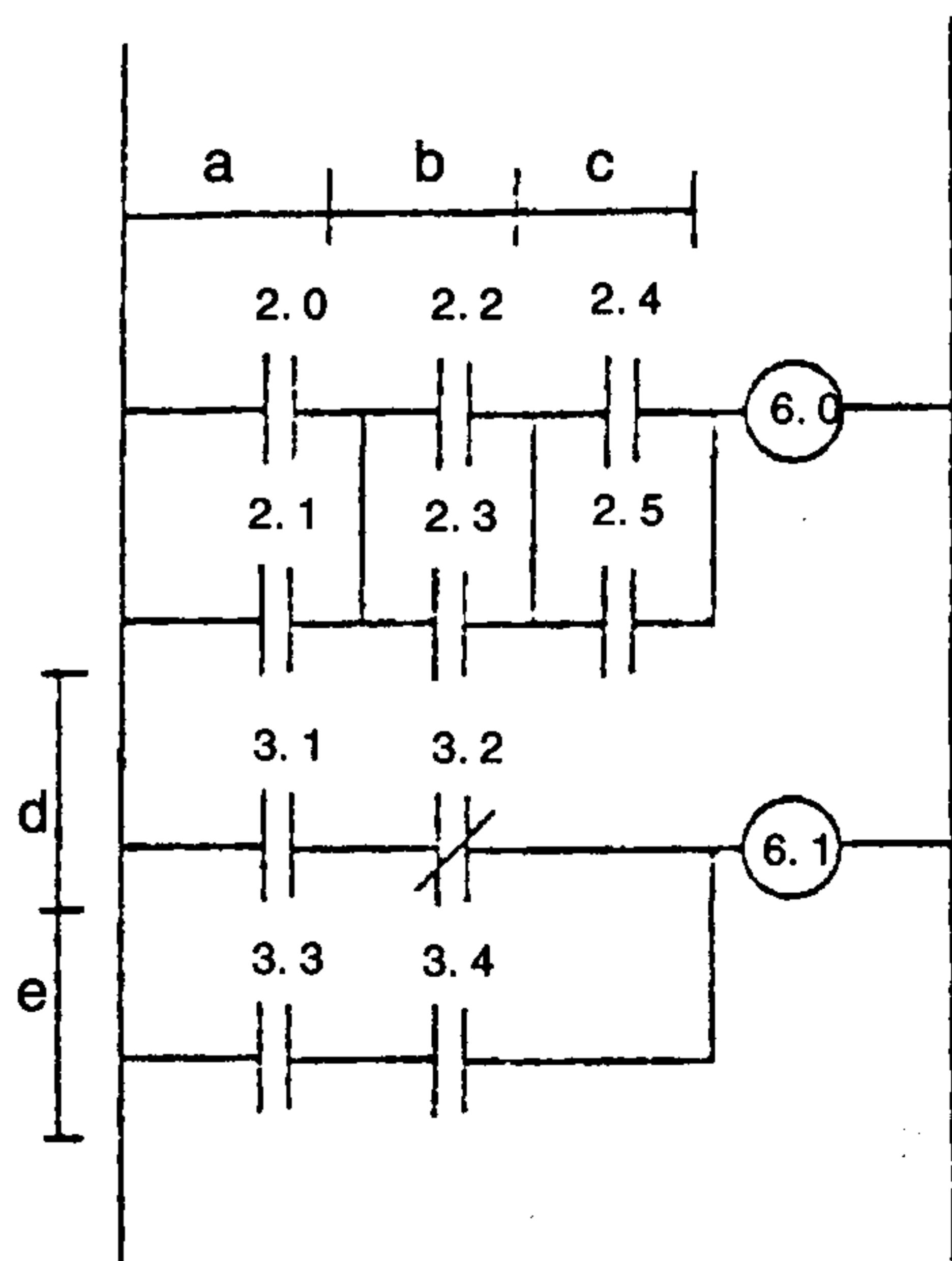


图 2—8

说明: 1、二个或更多的触点并联连接的电路称为并联电路(例 a、b、c 模块)二个或更多的触点串联连接的电路,称为串联电路模块(例 d、e 模块)。对并联电路模块的串联,串联电路模块的并联称为复杂支路。每个电路模块也可称为复杂分支路。

2、并联电路模块串联时,分支电路起始用 LD 或 LD NOT,以(A)结束,分支内编程不变,不允许没有(A)指令,又采用新的分支回路。

3、串联电路模块并联时,分支电路起始用 LD 或 LD NOT,以(O)结束,分支内编程不变,不允许没有(O)指令,又采用新的分支回路。

4、有多个电路模块串并联,可依次使用(A)(O)指令,模块数不受限制。

5、MCS 和 MCR:

MCS:主控母线开始指令(Master Control Set)。

MCR:主控母线复位指令(master control Ret)。

这两条指令为单字节指令,只有指令符,没有编码。而且这两条指令在程序中必须配对出现。

某些复杂的梯形图上要求,通过几个触点后,再控制一部份线路,待该部分线路动作后,再回到主控母线,继续完成下面的动作,这时就需要使用主控指令 MCS 和 MCR。

指 令		备 注
指令符	编码	
LD	1.0	主控母线开始
MCS		
LD	2.0	
OUT	7.0	
LD	2.1	
AND NOT	2.2	
OUT	7.1	
LD	2.3	
LD	2.3	
VAL	0.100	
TIM	40	主控母线结束
MCR		
LD	1.1	
AND	1.2	
OUT	7.2	

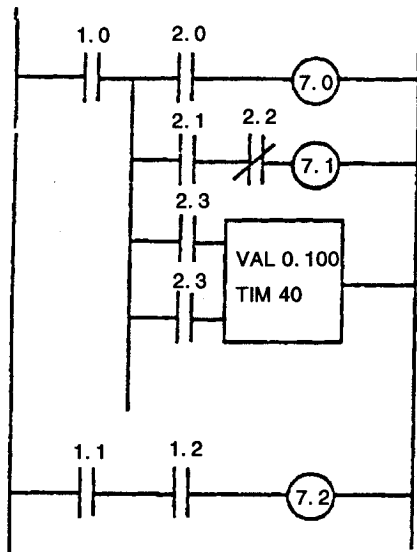


图 2—9

说明: 1、MCS、MCR 必须成对出现,注意编程时不能将 MCR 漏掉,否则将出现 Err8。

2、在梯形图中可以多次出现主控母线,但嵌套只能如图 8 次,超地 8 次将出错(图 2—10)

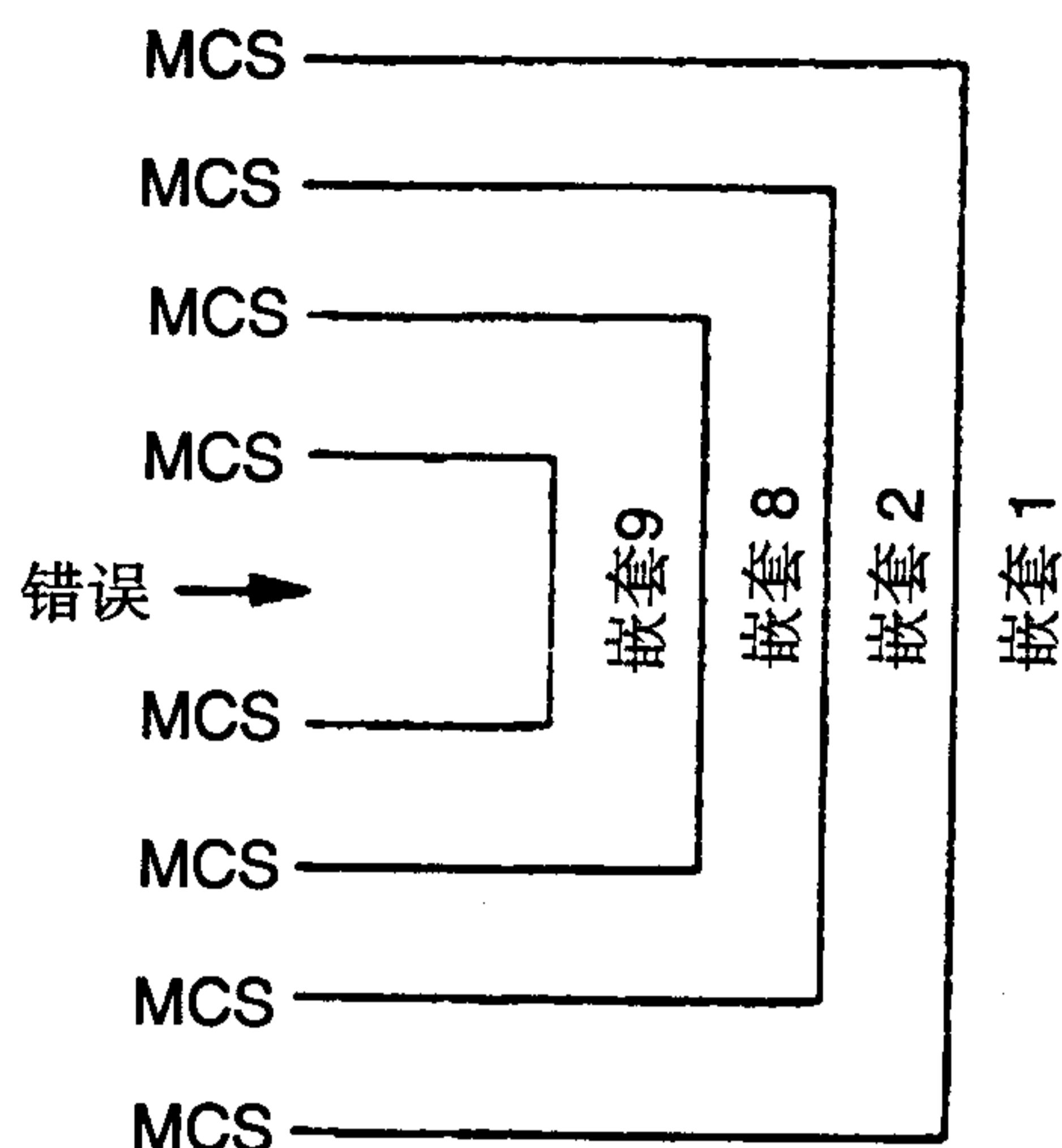


图 2—10

3、当触点 1.0 断开时,不论其它触点状态如何,受母线控制的 7.0、7.1 断开,定时器复位到设定值。

4、主控指令 MCS 之后,任何支路都需要以 LD 或 LD NOT 开头(公共母线移至一个新的母线)

5、用 MCR 指令使得各支路起点回到原来母线上。

6、不要将输出类指令为 OUT、TIM、CNT、SR、SC、CLR、SET、RST 等直接跟在 MCS 指令后,否则容易出错(如图 2—11)

指令		备注
指令符	编码	
LD	2.0	
AND	1.0	
MCS		
OUT	8.0	容易出错,不采用
LD	2.0	正确
MCS		
LD	1.0	
OUT	8.0	

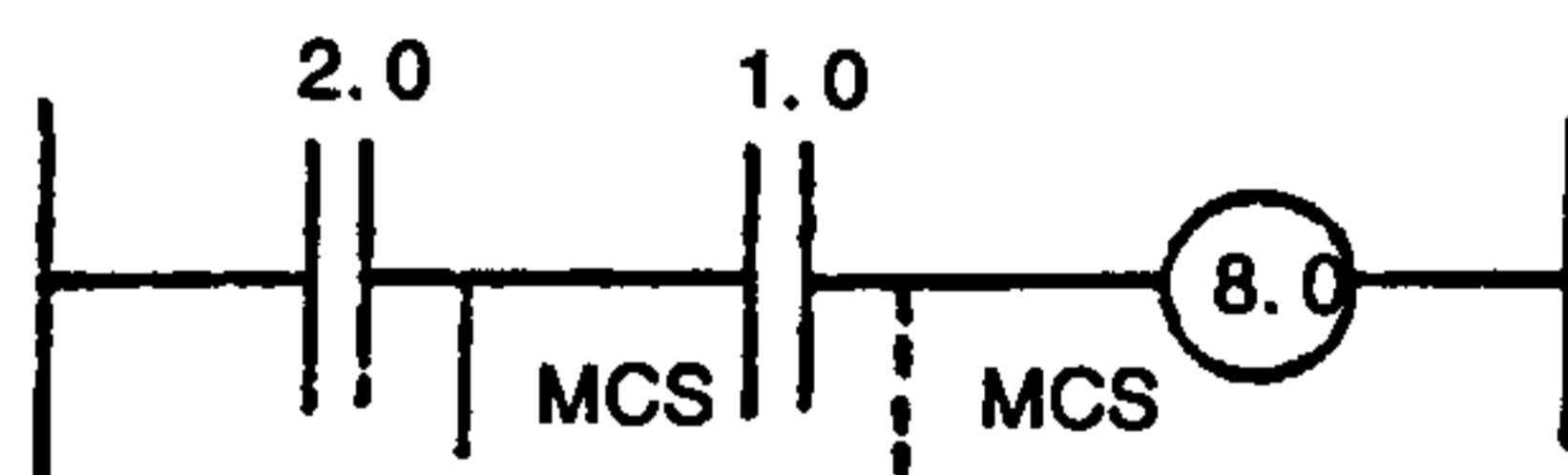


图 2—11

### 6 END

这是一条辅助指令,用户在所有程序输入完以后,输入此指令。控制器在循环执行用户程序时,遇到 END 指令则返回,从头开始新的循环。

另外,END 的使用,可以使用用户分段调试程序,如图 2—12。当用户程序可以分为若干个独立的子块时,在调试第一子块时将 END 指令插入其后,则可单独调试第一块,当第一块调通后,将 END 指令移至第二块后,则能进行第二块程序的调试,直到最后一个程序块调试完。

END 指令的输入方式和其它指令略为不同,按 **SHF** 上档键后,再按 **END NOT** 键,这时 END 指令已输入用户程序区。不必再按动 **WRITE** 键,若按动 **WRITE** 键,则给出错误信息。

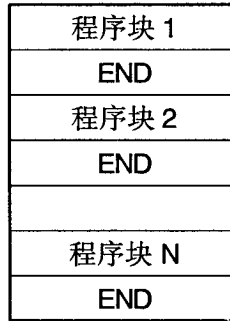


图 2—12

### 三、定时器及计数器指令

#### 1、定时器指令 VAL TIM

定时器的选用可以在辅助继电器区域内任选一卡作为定时器,其卡号作为定时器号,该卡的位地址 7 便作为定时器的输出触点。当定时器定时到达时,该触点接通,此触点可以在 PC 的程序上任意选用,其余 7 个继电器作为中间寄存器,存放计数瞬时值,故不能作为输出使用。

PC80 指令系统提供的定时器,共有两个输入端,其中一个为复位端,另一个为定时器控制端。如图 2—13 所示。

定时器的编程为双语句形式,它由定时值和继电器定时输出构成,VAL 语句指定定时值,TIM 语句指定定时器输出号,定时器号可以是辅助继电器中任意一卡号。

指 令		备 注
指令符	编码	
LD	0.0	0.1 秒时标,定时 10 秒 30 卡作为定时器 30.7 作为定时器输出
LD	0.1	
VAL	0.100	
TIM	30	
LD	30.7	
OUT	4.0	

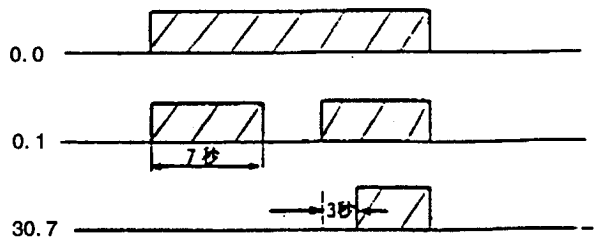
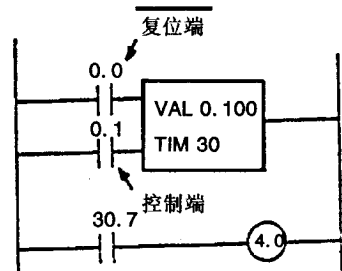


图 2—13

定时复位端触点不接通时,定时器恢复设定的初值,使输出触点断开。控制端触点断开时,定时器停止计时,控制端触点通时,定时器计时,定时器的正常工作,要求控制端、复位端同时接通。

定时器采用三个基准脉冲时标;0.1秒、1秒、10秒。

VAL语句所跟参数,小数点前一位为时标代码。可选用0、1、2三个数中一个,分别代表0.1秒、1秒、10秒3种脉冲时标。小数点后的三位数为设定值的范围,规定为000—127。如果我们想定时50秒,则可有两种选择。一种是:VAL1.050;另一种为VAL2.005。为了得到较高的定时精度,对50秒定时,推荐使用VAL1.050语句,这时定时误差在1秒范围内。定时误差与时标基准有关,时标基准愈大,误差愈大。

对于选用一般的辅助继电器区域作定时器,在电源OFF时,保持计时值,重新给电后,继续计时。因此想在停电后也能记忆以前定时器值,则应选用停电保持继电器。如图2—14,选用48卡作定时器,定时5秒,当计时到3秒时,电源掉电,重新给电,定时器接着计时2秒,接通输出点,如图2—15。此外,本控制器还提供两路两位外接拨码盘预置数的定时器。此时定时器的时间数值由外接十进制二位拨码盘控制,编程时定时值可选用VAL0.000、VAL1.000或VAL2.000,分别表示时标0.1秒、1秒和10秒。定时器号只能选用TIMO或TIM1。而且在第一个I/O槽插入模块,并连续接拨码盘。详见第三章编程指导。

指令		备注
指令符	编码	
LD	0.0	
LD	0.0	
VAL	0.050	
TIM	48	
LD	48.7	
OUT	4.0	

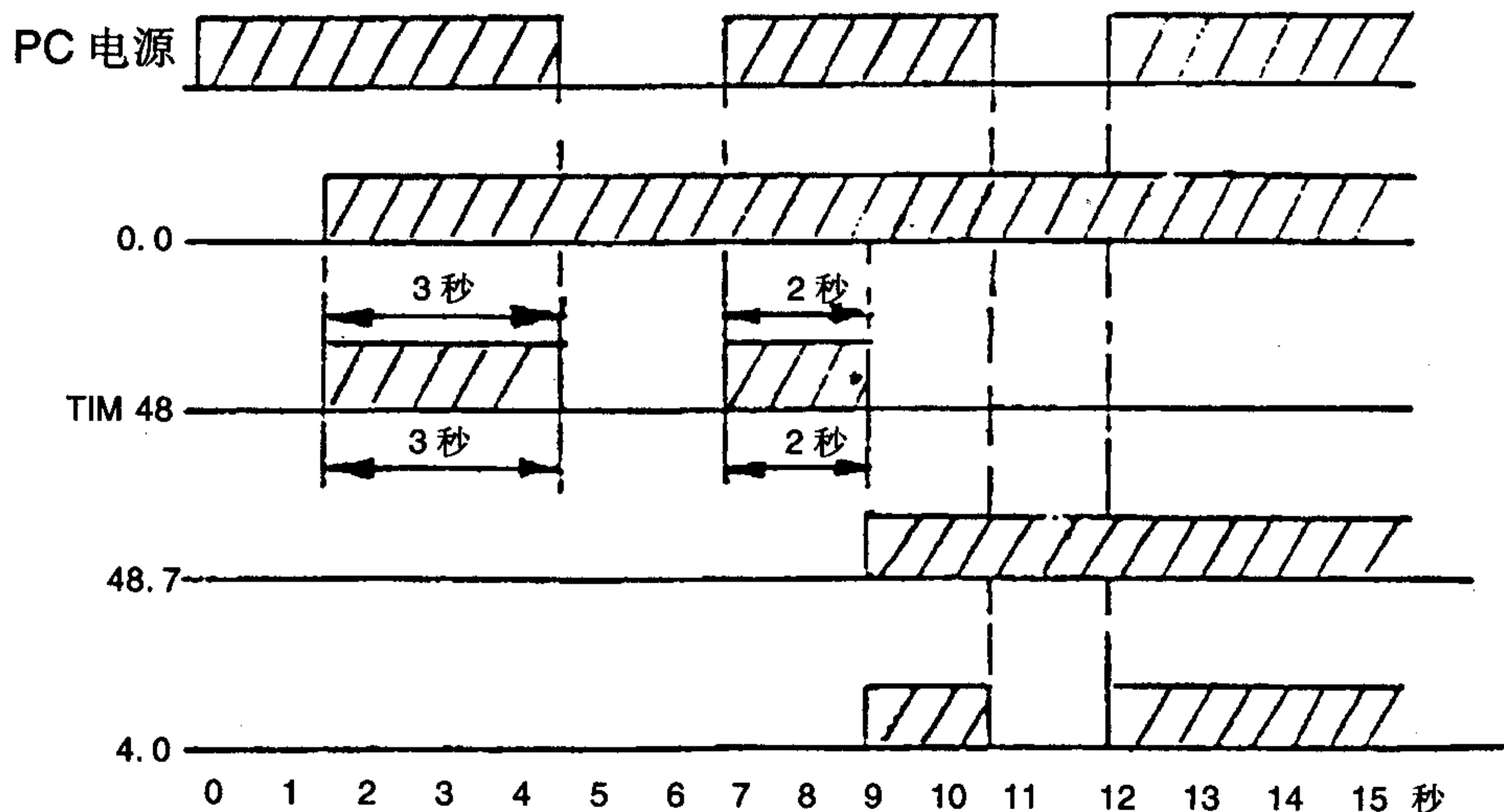
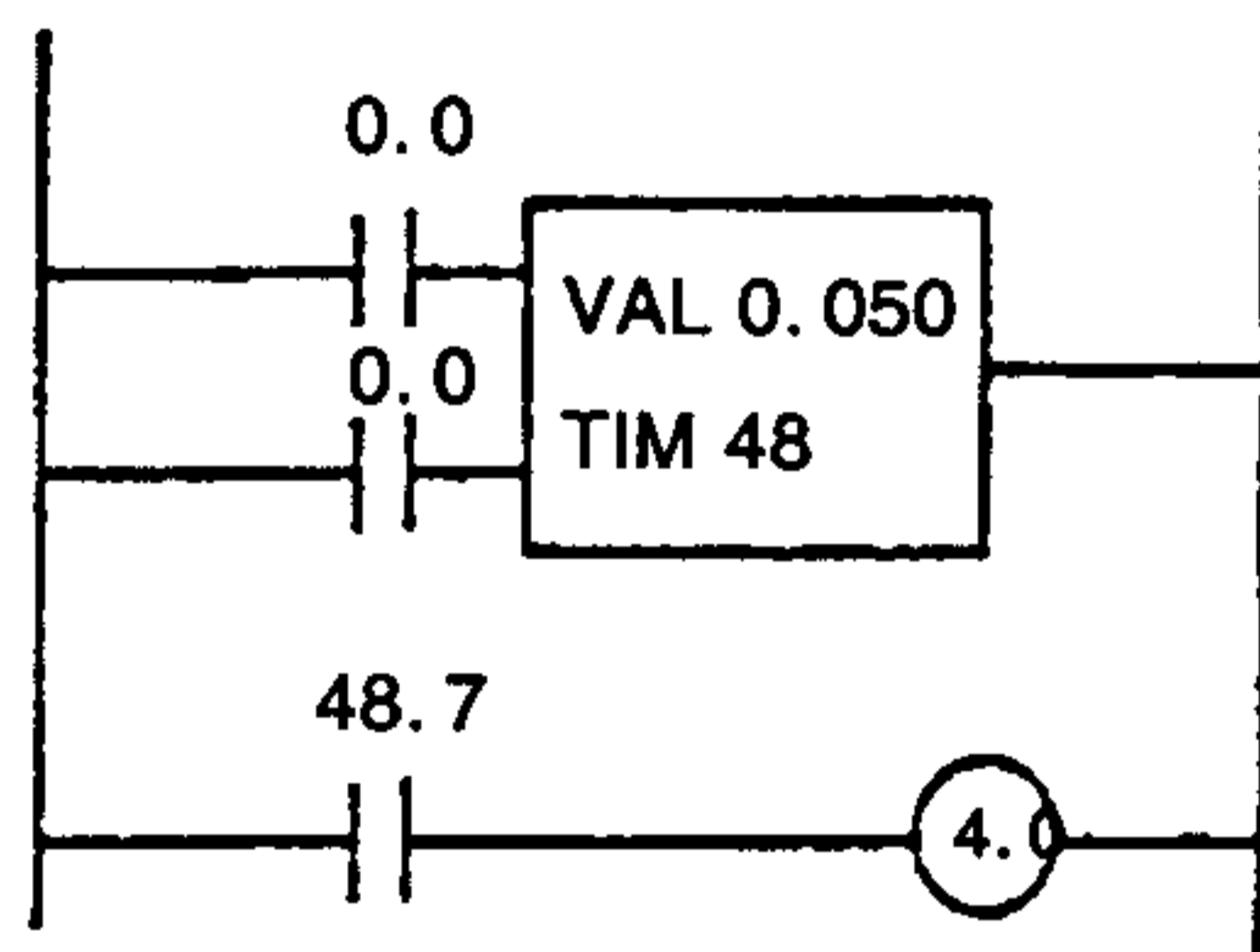


图 2—15



## 2、计数器指令 VAL CNT

计数器的选用也是选定辅助继电器区域的一个卡作为计数器使用。该卡的卡号作计数器号。位地址 7 作为计数器的输出触点此触点可以在 PC 程序中任意选用。其余 7 个继电器作为中间寄存器,存放计数瞬时值,故不能作为输出使用。当计数器计数到达时,该点接通。计数器的编程为双语句形式,它由计数值和继电器计数输出构成。VAL 语句指定计数值,CNT 语句指定计数器输出号,计数器号可以是辅助继电器中任意一卡号。

计数器有二个输入端,一个为计数脉冲输入端,一个为计数器复位端(如图 2—16)。计数脉冲输入端上升沿有效,每来一个脉冲上升沿,计数器计数一次。当复位端无效时(OFF),计数器被复位,重新置为设定值。上例中选用 42 号卡作为计数器。计数值为 40,当计数脉冲累计到 40 时,42.7 接通,继电器 4.1 接通。

计数器的设定范围为 0—127,值为 0.000 ~ 0.127

如果希望 PC 掉电后,计数器仍然保持在掉电前的计数值,则可选用停电保持型辅助继电器区域作计数器。

计数器同样可以选用外接拨码盘即选用 0 号、1 号的输入卡,此时计数器的计数值由外接十进制二位拨码盘控制,编程时定时值用 VAL0.000,计数器号只能选用 CNT0 或 CNT1 而且在第一个 I/O 槽插入输入模块,并连接拨码盘。见第三章编程指导。

指 令		备 注
指令符	编码	
LD	0.0	计数脉冲
LD	0.1	
VAL	0.040	
CNT	42	
LD	42.7	
OUT	4.1	

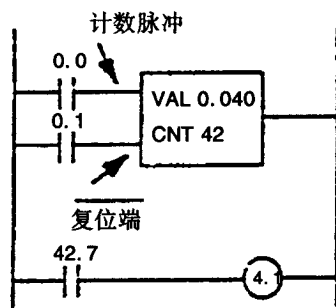
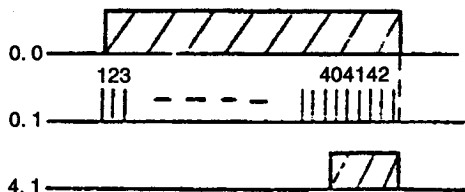


图 2—16



## 四、移位寄存器指令和步控制器指令

它们的清零指令为(CLR)

### 1、移位寄存器及编程指令 SR

移位寄存器是选用辅助继电器区域内的任意一个卡组成,即一个基本的移位寄存器由 8 位组成,它不受数量限制。

移位寄存器结构如图 2—17、图 2—18 所示。移位寄存器有两个输入端:移位数据输入端,移位脉冲输入端,移位脉冲上升沿有效,当脉冲输入端产生一个上升沿时,移位寄存器将此刻数据输入端的数据移入寄存器,同时原来寄存器同的数据向后移一位,原 7 号位的数据丢失。移位寄存器的每一位都可以向外输出。

指令		备注
指令符	编码	
LD	0.0	清零
LD	0.1	
SR	16	
LD	0.2	
CLR	16	

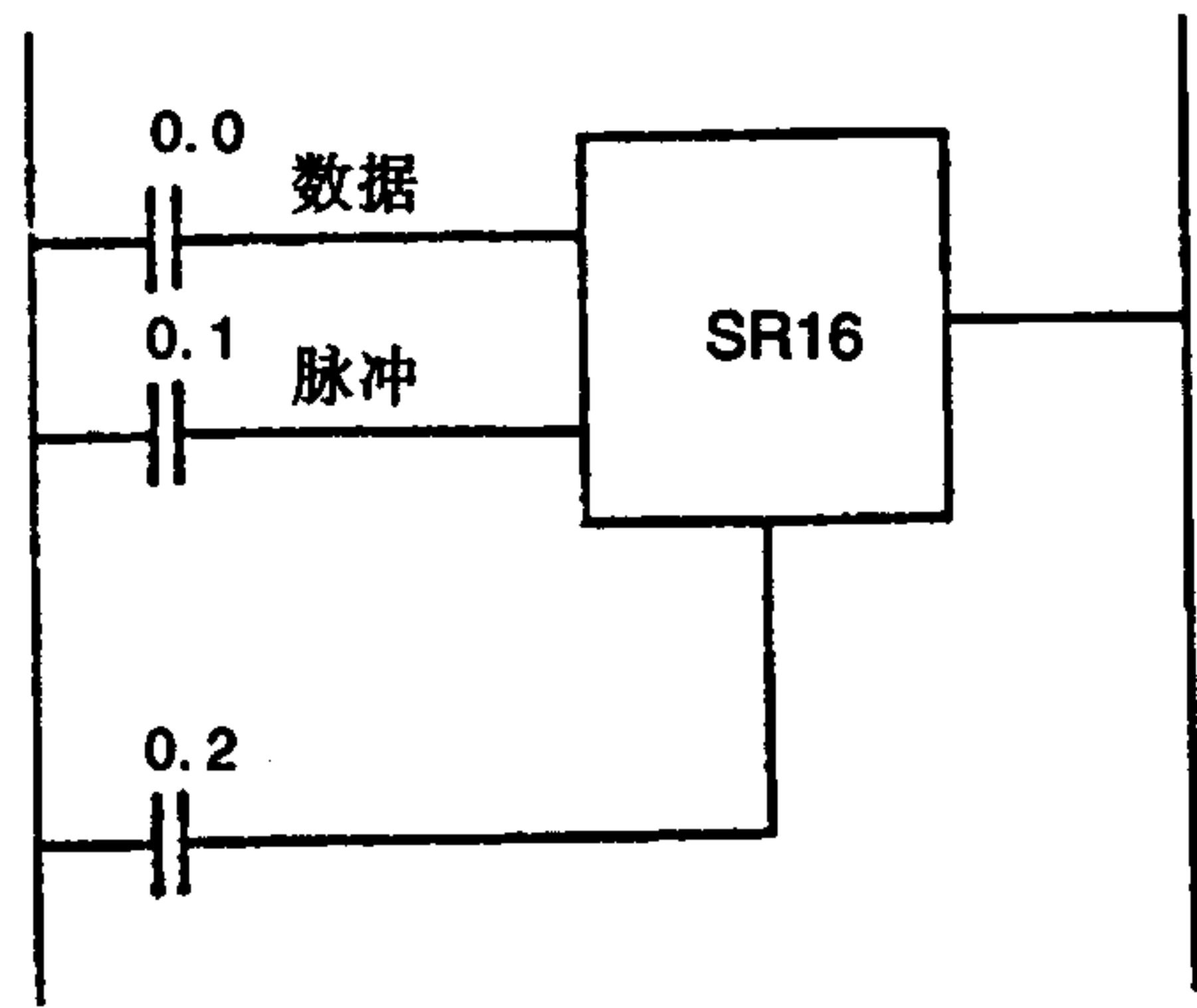


图 2—17

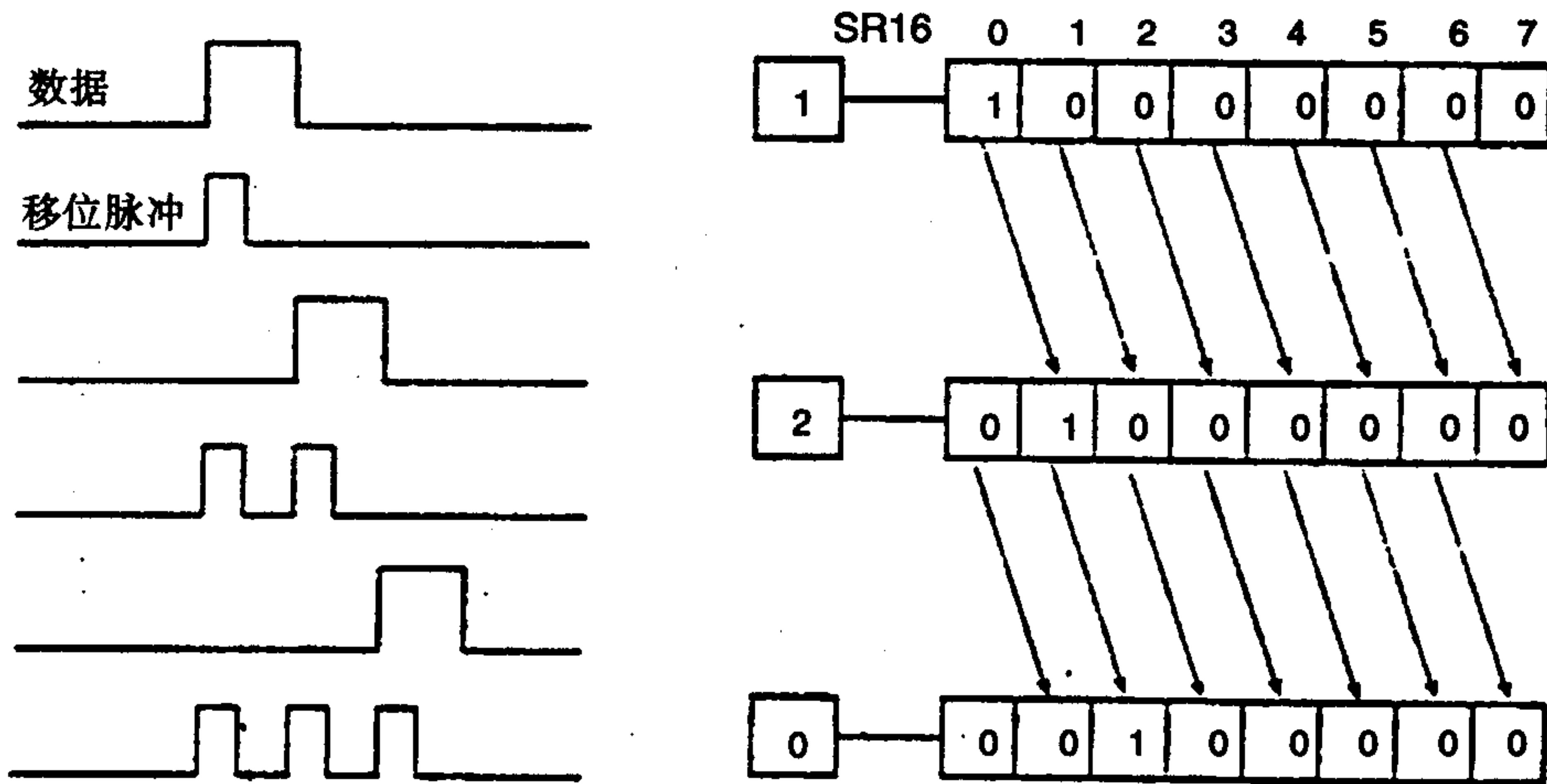


图 2—18

当执行到 CLR16 时 16 卡内移位寄存器全部清零

当一个移位寄存器不够使用时,可以把二个、三个移位寄存器串联起来,为了防止信息丢失,一定要把高位移位寄存器放在上面,低位移位寄存器放在下面,具体的梯形图及指令见第三章编程指导。

## 2、步控制器及编程指令 SC

步控制器的构成也是选用辅助继电器区域的一个卡,组成一个具有 8 个控制步的基本步控制器。该控制器的编号即为选用的卡号。位地址 0~7 表示位。分别代表控制器的第 0~7 步。步控制器具有自保持功能,具各步之间具有互锁功能。步控制器的一般编程过程如图 2—19。图中给出最基本的步控制器前三步的编程过程,整个控制器 8 步的编程如此类似。

指 令		备 注
指令符	编码	
LD	0.0	40 卡清零
SC	40.0	
LD	0.1	
SC	40.1	
LD	0.2	
SC	40.2	
LD	1.0	
CLR	40	

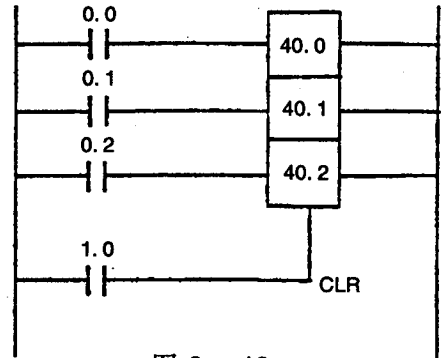


图 2—19

说明：

- 1、步控制器中,输出是自保持的,到下个输出指令或有清除指令为止,连续输出。
- 2、步控制器是后人优先电路,即对应程序中后人的指令,输出时优先。
- 3、步控制器内各步的输出是互锁的,即使有多个输入条件有效,同时只有一步输出。
- 4、当 CLR 有效时,卡 40 的内容全部清零。

表 2—3 将清楚地表明了图 2—19 所示电路的工作状态。

表 2—3

	输入条件				输出条件	备 注
	0.0	0.1	0.2	1.0		
初始	0	0	0	0	无输出	
单 个 输 入	1	0	0	0	40.0	
	0	0	0	0	40.0	自保持
	0	1	0	0	40.1	
	0	0	1	0	40.2	
	0	0	0	1	无输出	清除
多 个 输 入	0	1	1	0	40.2	后人优先
	0	1	0	0	40.1	互锁
	1	1	1	0	40.2	
	1	1	0	0	40.1	
清 除	x	x	x	1	无输出	
	0	0	0	0	无输出	自保持

### 3、CLR 指令：

CLR 指令是清除指令，当 CLR 指令的执行条件满足时，则清除 CLR 指令中操作数所指出的卡地址的内容。这一指令的使用可以使用户方便地在程序运行中，用条件控制指定卡的清除。

辅助继电器运行中的强制 OFF。在运行中使 CLR 执行条件满足，可以对辅助继电器区域进行一个卡的强制 OFF。

输入，输出继电器运行中的强制 OFF。对输入继电器某个卡的输入在运行的某一阶段要求全卡无效，只需在整个用户程序最前面加入一条 CLR 语句。对输出继电器整个卡的输出，如果在运行过程中的某一阶段要求输出无效，只需在用户程序的最后加一条 CLR 指令。

定时器计数器运行中的强制接通。定时器、计数器运行时只有其值为零时，输出触点才接通，采用运行中执行 CLR 指令，强制迫使数值为零，从而触点接通。（对计数器而言，CLR 指令执行后，还需有一个计数脉冲产生，才使输出触点接通）

移位寄存器及步控制器中应用 CLR 指令可对移位寄存器、步控制器清零。

按图 2—19 例中的功能还不能满足步进式控制的要求。在步进式顺序控制系统中要求各工步是一步接一步运行下去，在程序没有进行到最后一步不允许返回到前面工步。在上例中如果执行到 40.1 工步时，由于触点 0.0 接通，40.0 的输入条件满足，则 40.0 有输出。步控制器又返回 40.0 工步，显然不符合步进式控制的要求。为了确保程序正常执行，必须对程序进行改造。图 2—20 表示 8 步步进式顺序控制器的梯形图，与上例比较加二个条件，一是起动条件，二是步进条件。

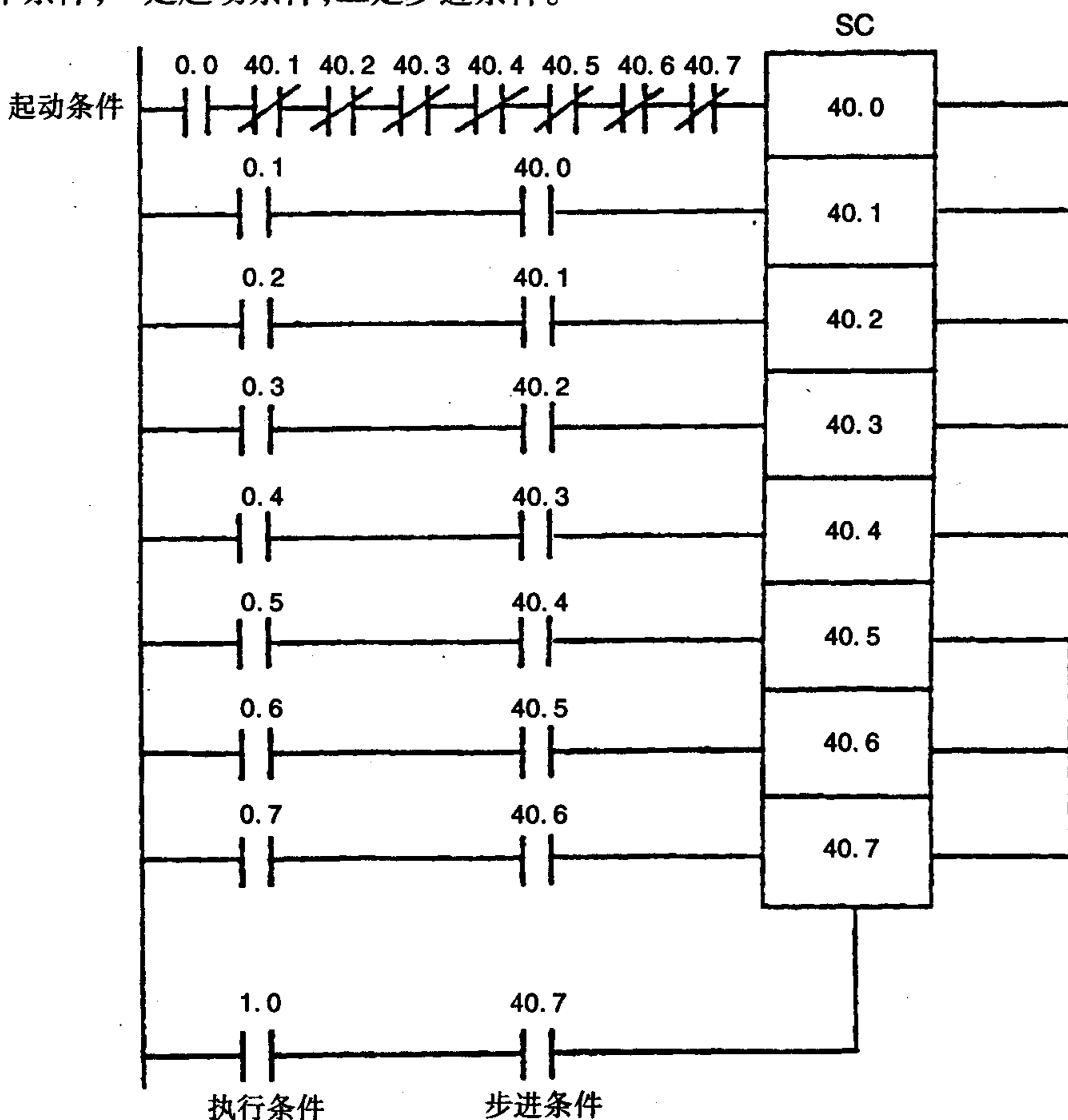


图 20—20 实用步进控制器

如果起动条件没有满足,则 40.0 不可能输出。由于在起动条件中使用了 40.1 ~ 40.7 触点与反(即常闭触点串联),如果步控制停留在某一步,代表该工步的继电器的常闭触点必须断开,起动条件不满足,那末不可能从中间某一步返回到第一步。例如当步控制器执行到 40.3 工步,则 40.3 常闭触点断开,40.0 的起到条件不满足。如果这时 40.0 的执行控制 0.0 触点接通,但不可能中断步控而重新启动。只有当所有工步执行完毕并执行 CLR40 指令,40.0 ~ 40.7 全部清零,起动条件才满足。在执行控制 0.0 触点接通时,SC40.0 有输出,开始执行步控第一步。确保下个步控周期的完整。

在 40.1 到 40.7 及 CLR 前面都增加了 1 个步进条件,每一工步从无输出变到有输出必须在执行完上一工步的基础上执行,同时由于 SC 的后入优先特性结束上一工步。例如要执行 40.4 工步,必须满足它的输入条件即执行控制 0.4 触点及步进条件 40.3 触点接通。而 40.3 触点接通,则必须执行完 40.3 工步。而当执行到其它工步时,40.3 触点不接通,40.4 工步不可能执行。同时 40.3 触点仅仅用在 40.4 工步的输入条件,故执行 40.3 工步以后也只能执行 40.4 工步,不可能转移到其它工步。从而保证步进式顺序控制的要求。如果在步进条件上作一些修改,可以完成步进的转移,用户可以根据自己的需要而作一些修改。

### 五、销存器指令

锁存器置位指令 SET

锁存器复位指令 RST

这两条指令均为双字节指令,即由指令符及编码组成,编码为某一触点的代码,它可以是输出继电器,也可以是输助继电器。

指 令		备 注
指令符	编码	
LD	1.0	9.0 继电器,接通并锁存
SET	9.0	
LD	1.1	锁存器断开
RST	9.0	

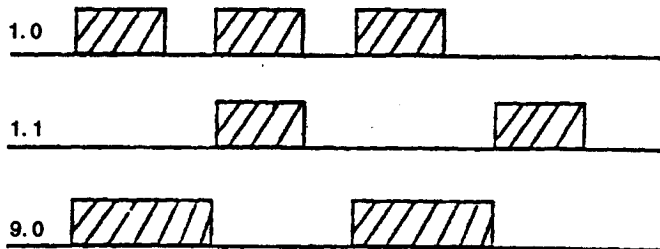
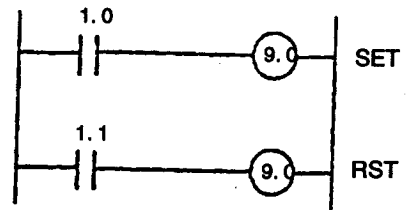


图 2—21

## 说明

锁存器一旦接通后,控制端对它无效,只有 RST 指令才能复位,如果锁存器选在停电保持区域,锁存器接通后,即使电源掉电,锁存器状态仍然保持,当电源重新供给后,锁存器继续接通,直到 RST 指令有效,锁存器才断开。

# 第三章 编程指导

本章着重介绍可编程控制器的编程原则及部份常用线路的编程方法。

## 一、基本梯形图的编程

可编程控制器编程时采用的梯形图,基本上与以往电气原理图有相似之处。

### 1、用 PC 代替继电器控制回路

图 2—22 表示用可编程控制器来代替继电器线路。对 PC 来说仅仅需要和外部输入、输出相连,中间有关内容,可用程序代替,而程序用编程器写入 PC 机。

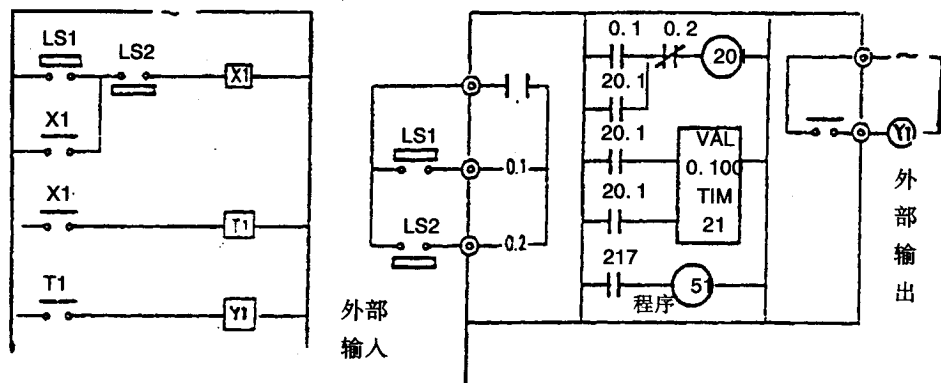


图 2—22

### 2、编程注意事项

- * 可编程控制器内部触点不受数量限制。
- * 图中只要标注触点、线圈编号。连接线不必编号。
- * 电流是从左向右流动,故复杂电路中不可有回路反流,不需要隔离二极管。
- * 继电器回路可以作并行处理,而 PC 机则对写入的程序从第一步到最后一步进行串行处理。处理完最后一步以后,又返回到第一步,就这样周期性反复,完成一个周期的时间为一个扫描时间。

### 3、梯形图的基本法则

触点放在支路左边,输出线圈接在右边总线上,一切触点不能放在输出线圈右边。如图 2—23。输出线圈不能直接接在左边总线上。如需要的话可以通过常闭触点连接输出线圈,如图 2—24。

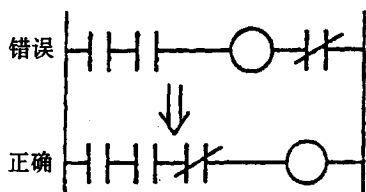


图 2—23

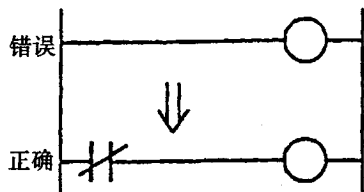


图 2—24

串联触点多的电路排在梯形图上面(对一个支路而言)。如图 2—25。

关联触点我的电路排在梯形图的左面。如图 2—26

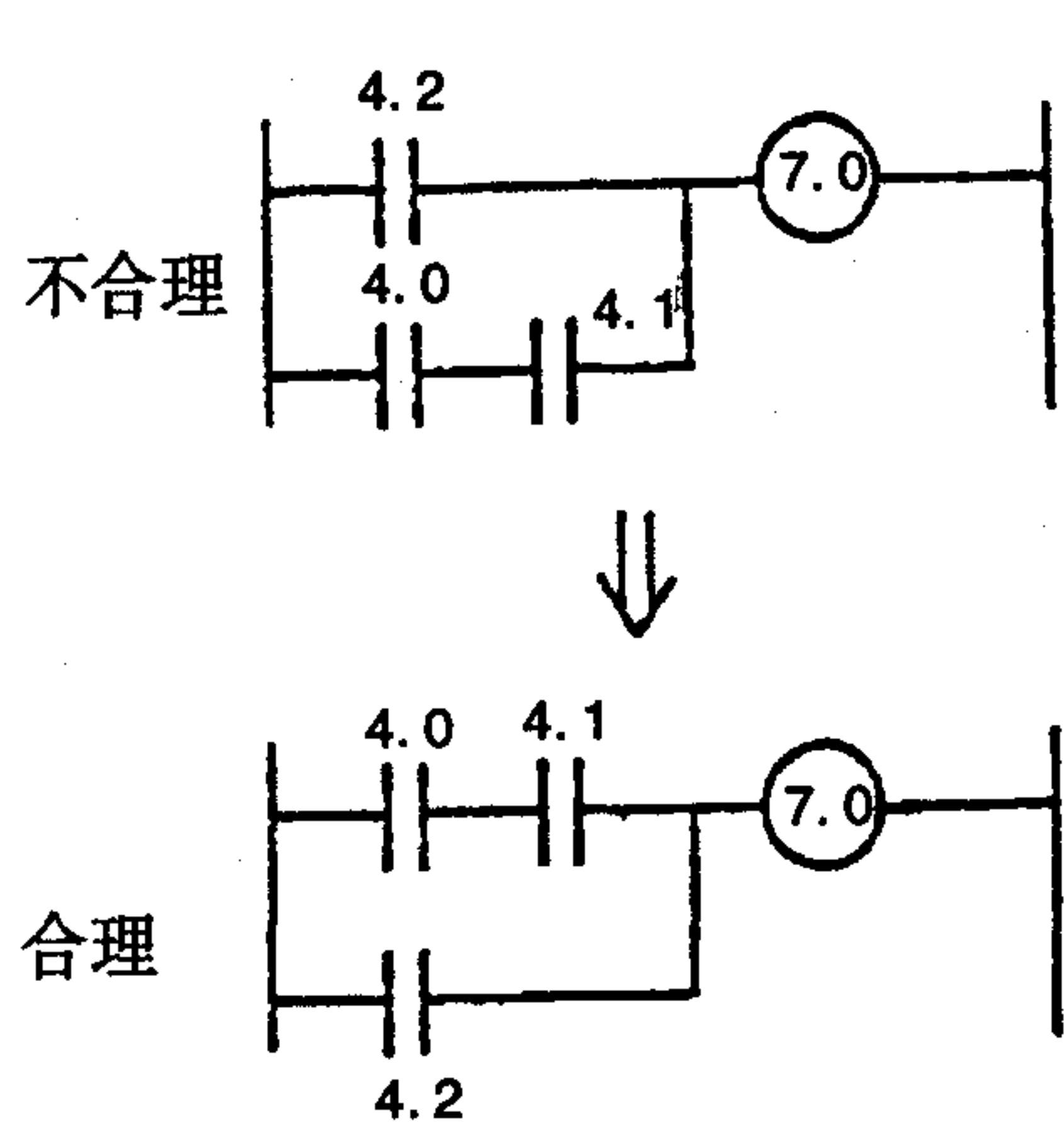


图 2—25

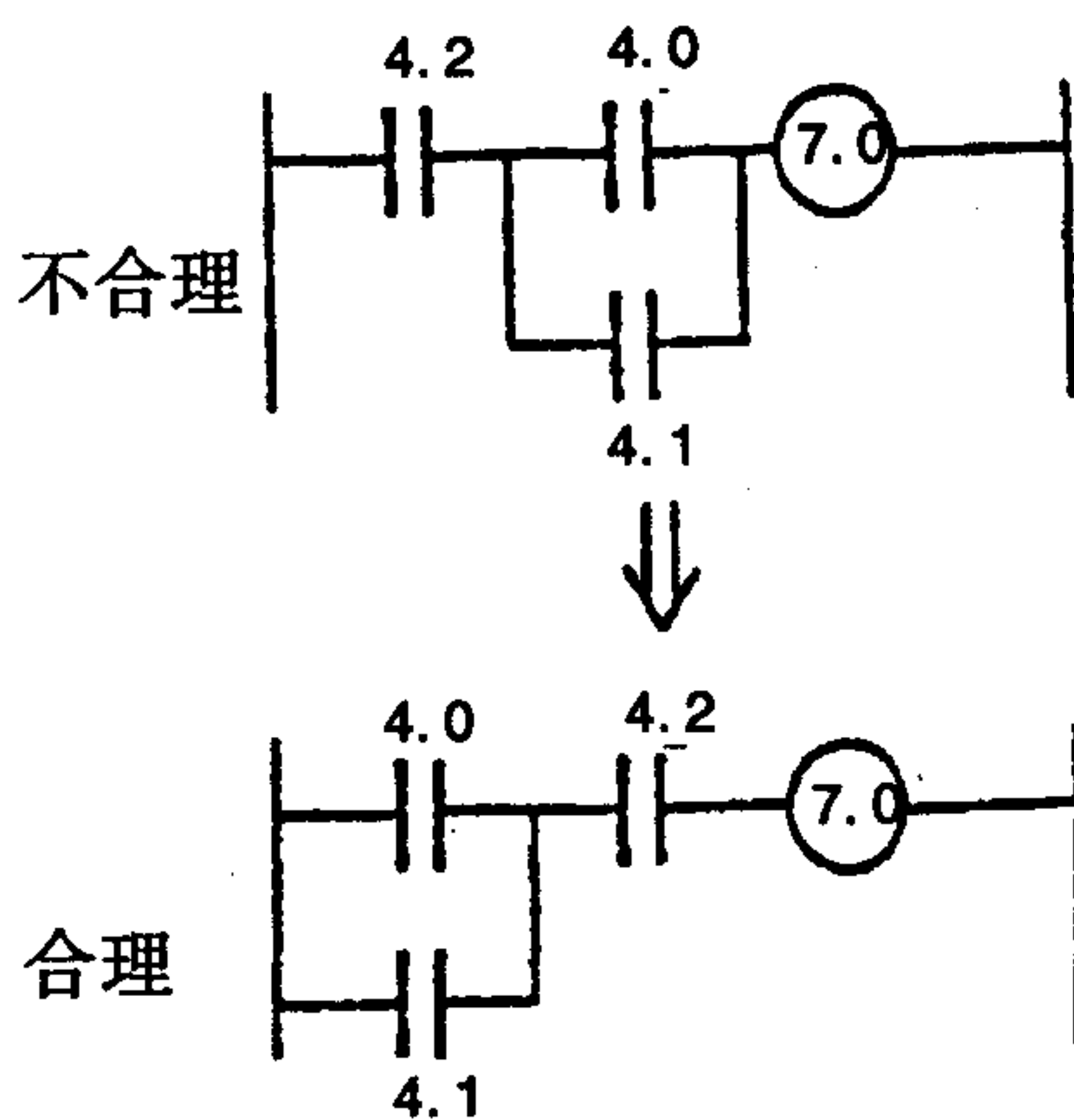


图 2—26

桥式电路必须修改后才能画出梯形图,如图 2—27,图 2—28。修改的方法可以是多种多样的。

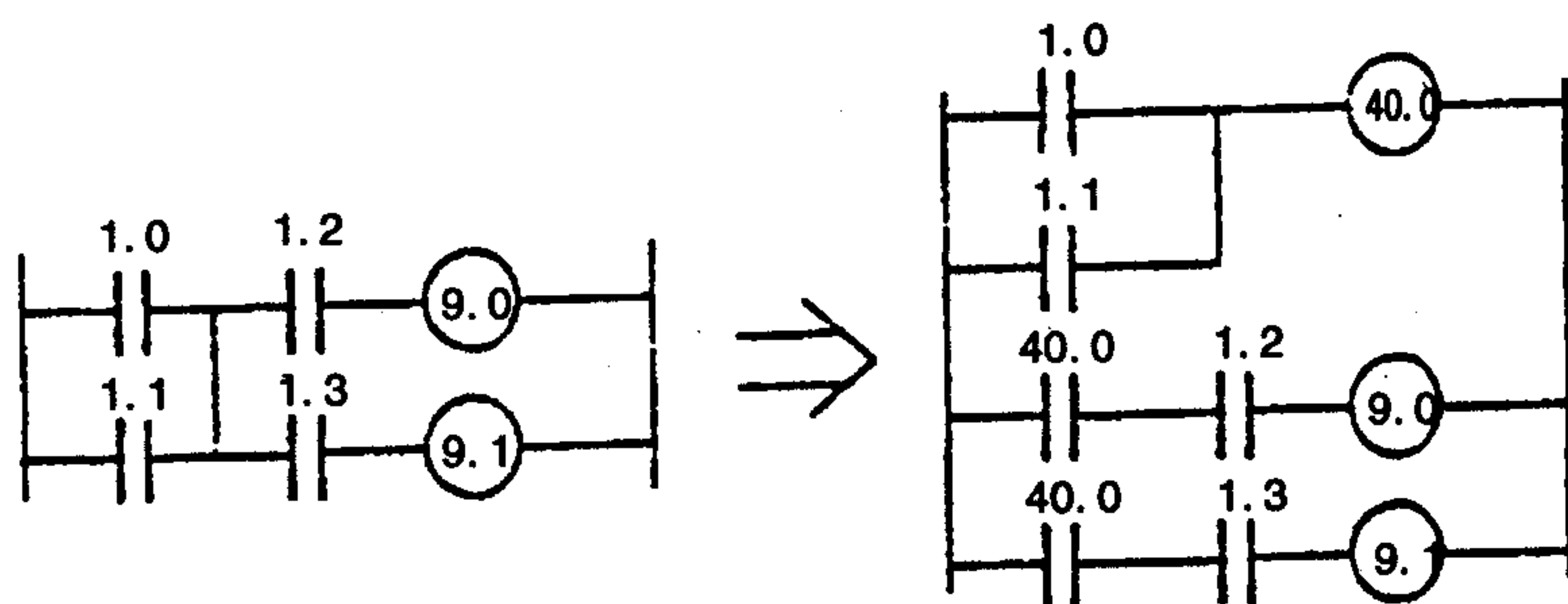


图 2—27

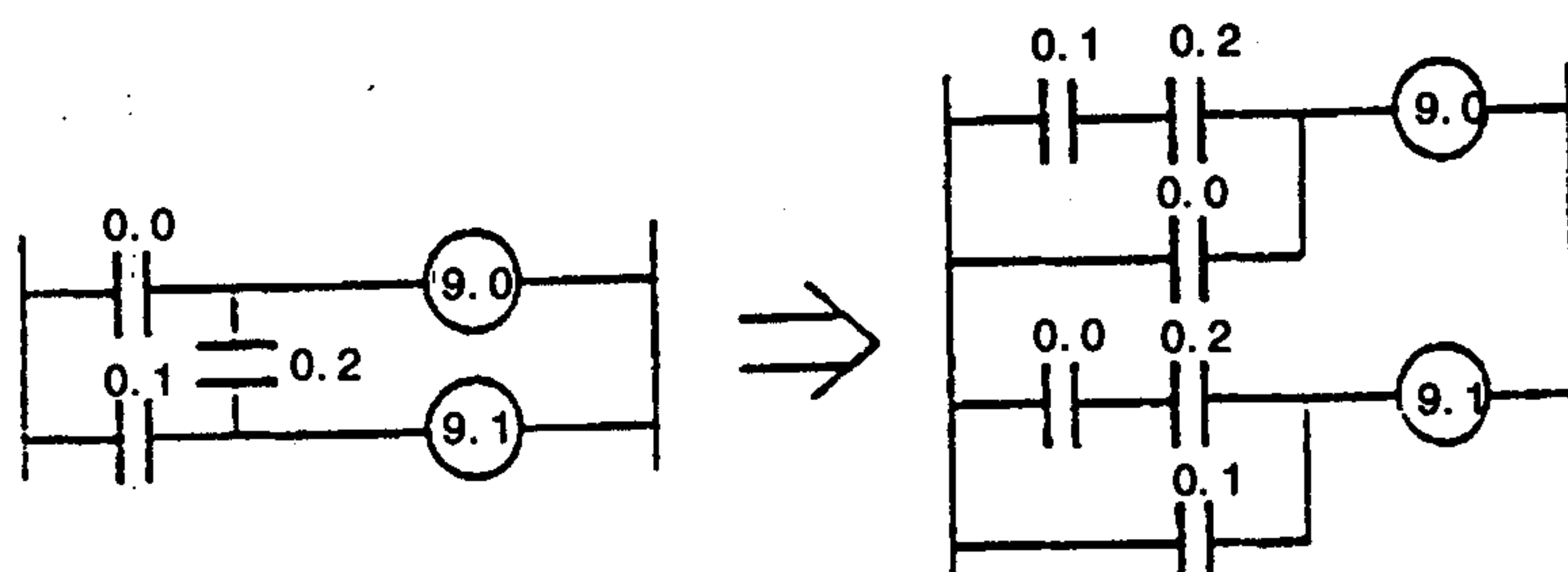


图 2—28

#### 4、PC 程序编制的原则

- (1)编程应从最左边的元素开始向右边进行编制。
- (2)计数器、移位寄存器等有二个或更多的输入支路,程序应按指定顺序编制。

#### 5、如何进行编制

把一个需要编程的支路分成若干个模块(分支)。每一个模块从最左边触点开始(由上而下)向右边进行编程。如图 2—29,表—4



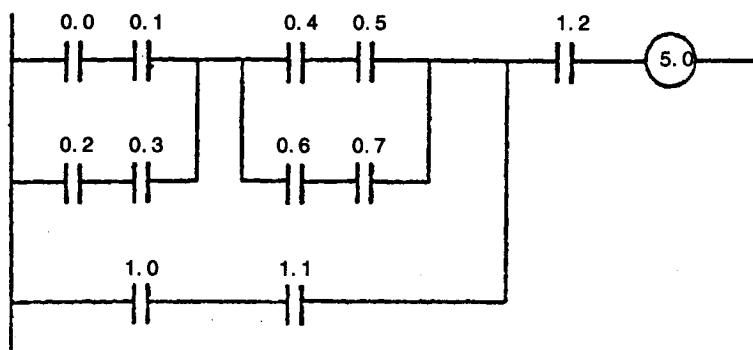
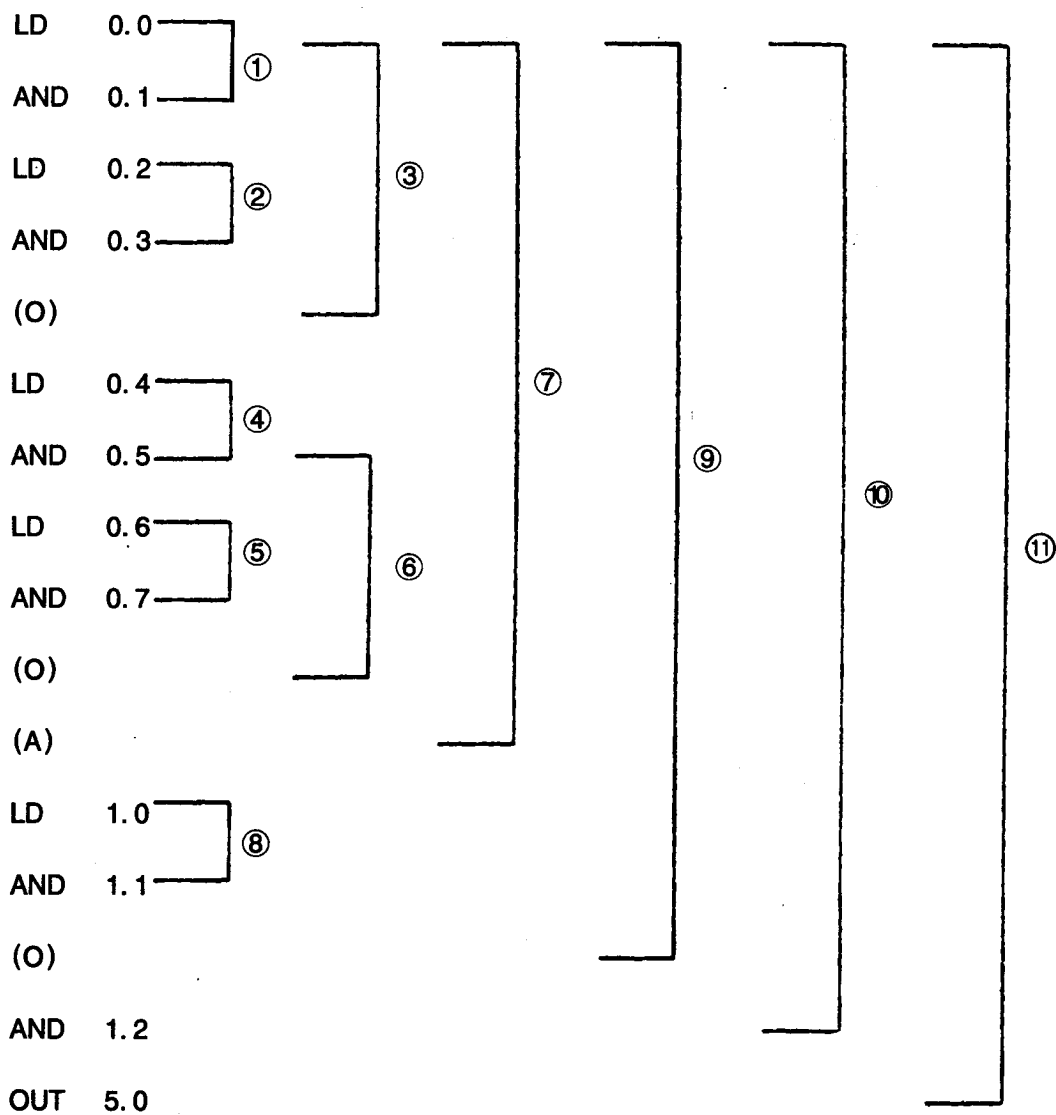
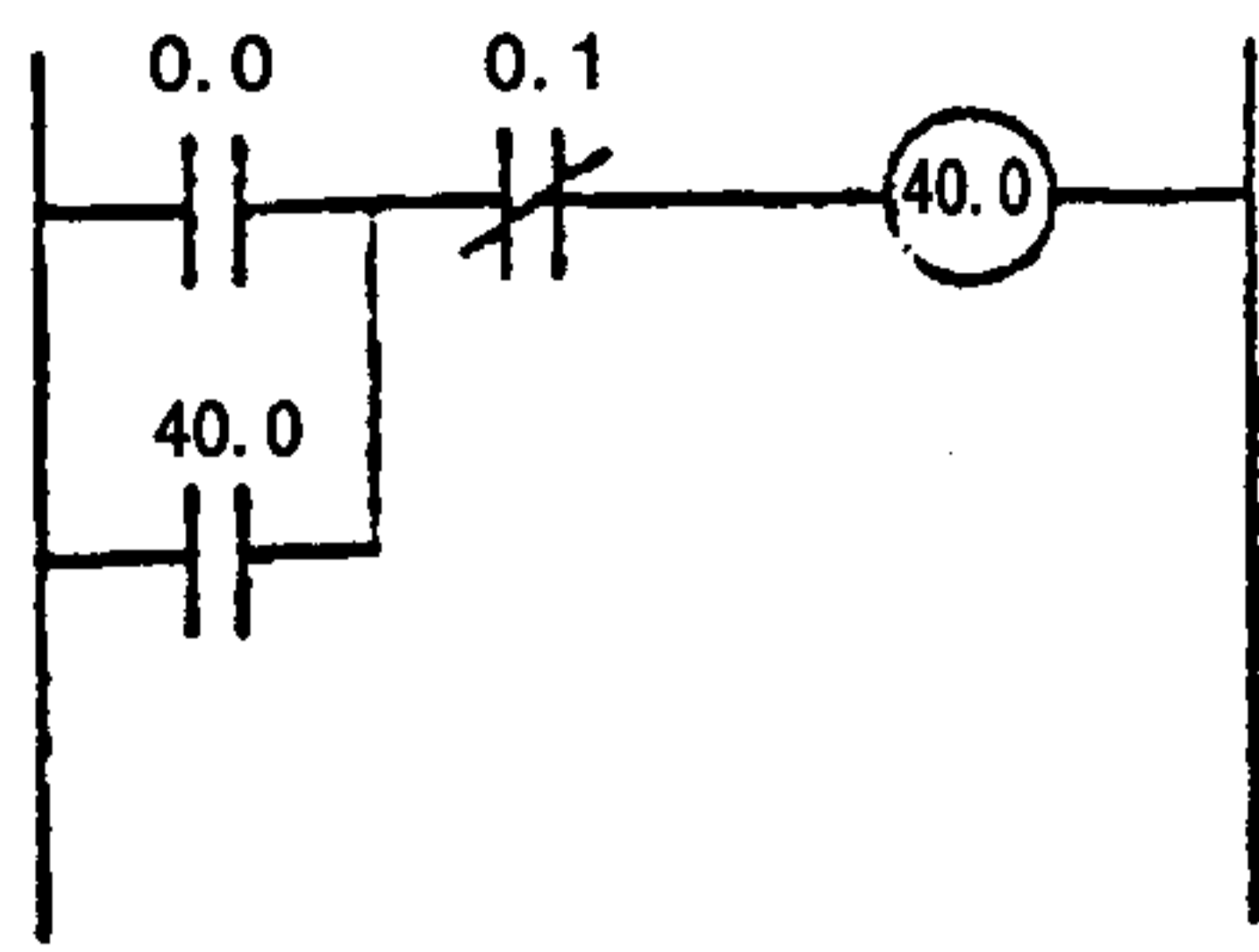
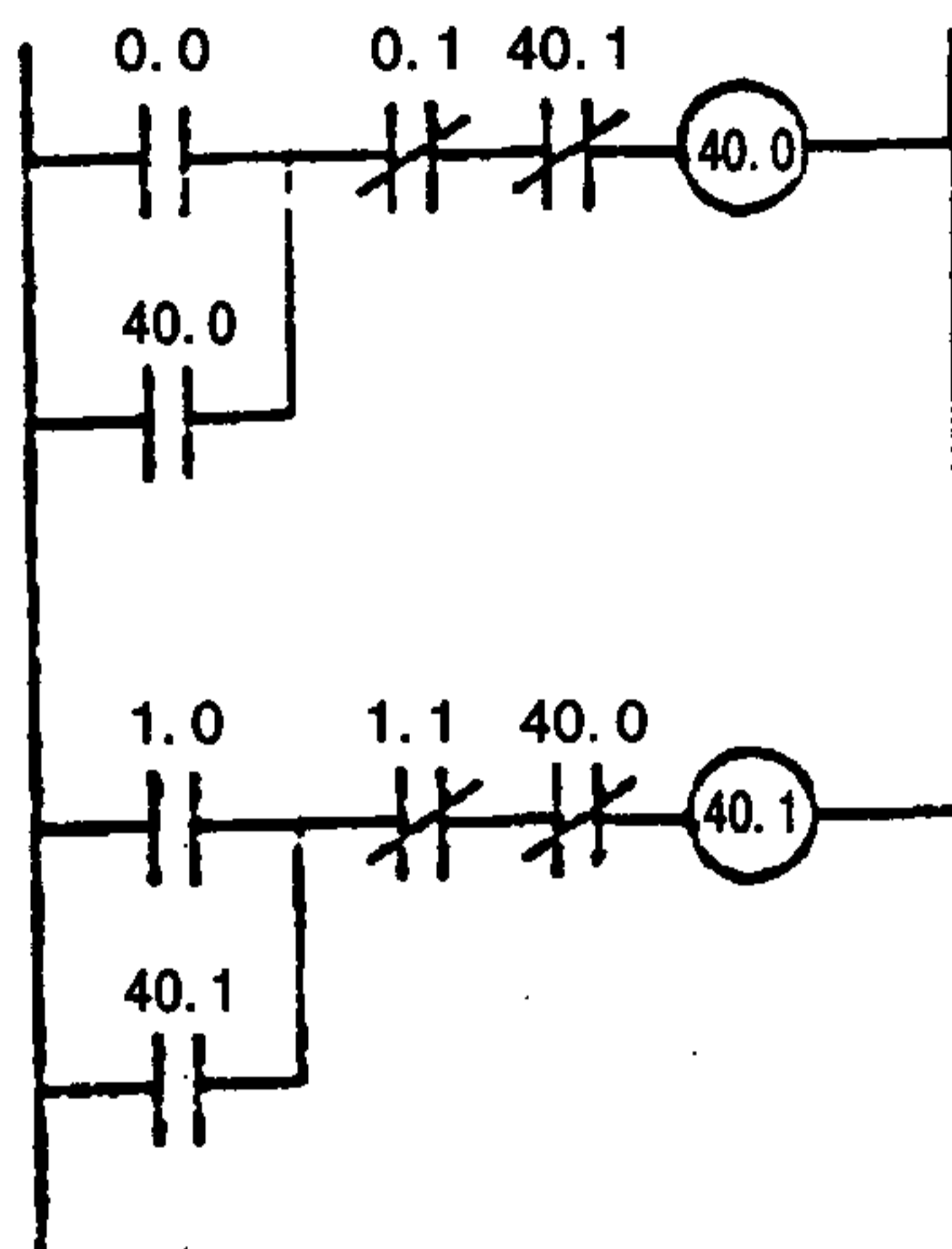
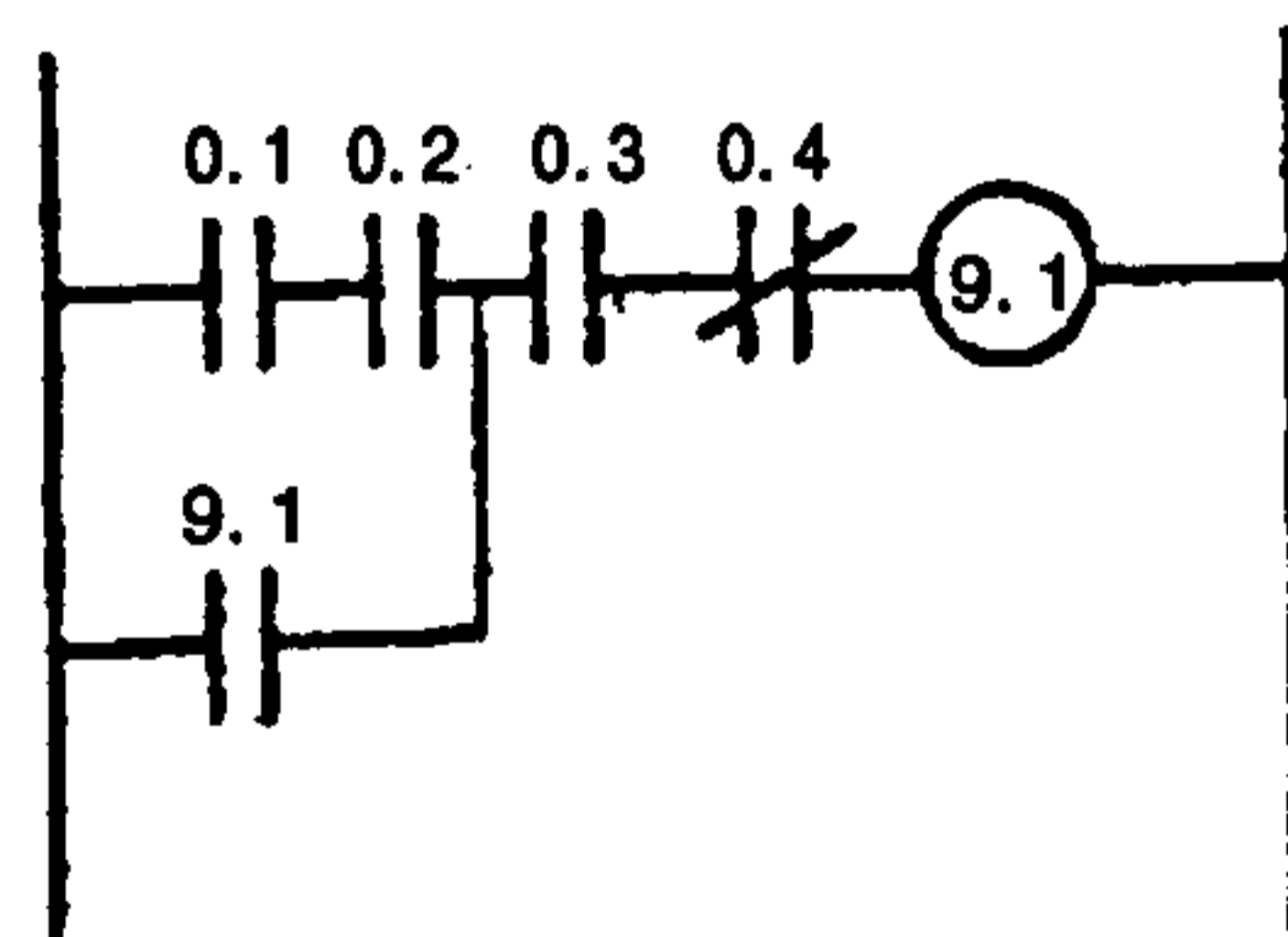


图 2—29

电路	梯 形 图	程序指令码	程序数据	说 明
自 锁		LD OR AND NOT OUT	0.0 40.0 0.1 40.0	开—0.0 关—0.1
互 锁		LD OR AND NOT AND NOT OUT LD OR AND NOT AND NOT OUT	0.0 40.0 0.1 40.1 40.0 1.0 40.1 1.1 40.0 40.0 40.1	开—0.0 关—0.1 40.1 导通,关断 40.0 开—1.0 关—1.1 40.0 导通、关断 40.1
关联 / 串 联电 路 (一)		LD AND OR AND AND NOT OUT	0.1 0.2 9.1 0.3 0.4 9.1	编程的顺序从上到下 从左到右执行 9号卡为输出继电器
	图形见下页	LD AND LD AND	0.1 0.2 0.3 0.4	先编制块 A1,A2,然后 用(O)联接成一体 A。 后编制块 B1,B2,同样 用(O)联接成一体 B。

续表

电路	梯形图	程序指令码	程序数据	说明
并联 / 串联 电路 (二)		(O) LD AND LD AND (O) (A) OUT	1.0 1.1 1.2 1.3  9.0	最后用(A)把 A、B 联接起来       9号卡为输出继电器
复杂 电路		LD AND AND LD AND AND AND (O) LD AND AND (O) OUT	0.0 0.1 0.6 0.2 0.3 0.5 0.4  0.4 0.5 0.6  9.0	如果有一个电路编程很复杂,可以重复使用一些接点改成等效电路,再进行编程就比较清晰易懂。       9号卡为输出继电器
桥式 电路		LD AND OR AND	0.2 0.4 0.0 0.1	桥式电路的编程要先将桥式线路改写成分支电路,如下图,然后再进行编程。

续表

电路	梯形图	程序指令码	程序数据	说明
桥式电路 (续)		<p>OUT 9.0 LD 0.0 AND 0.4 OR 0.2 AND 0.3 OUT 9.1</p>		9号卡为输出继电器
移位寄存器扩展和输出		<p>LD 16.7 LD 0.1 SR 17 LD 0.0 LD 0.1 SR 16 LD 16.5 OUT 9.0 LD 17.7 OUT 9.1</p>	<p>16.7 0.1 0.0 0.1 16.5 17.7</p>	<p>先编高八位后编低八位连接起来成16位,可进一步扩展,高位在前,低位在后。  任意位都可以输出。 9号卡为输出继电器</p>
先输入优先		<p>LD NOT 40.0 AND NOT 40.1 AND NOT 40.2 AND NOT 40.3 AND 0.0 SC 40.0 LD NOT 40.0 AND NOT 40.2 AND NOT 40.3 AND 0.1 SC 40.1</p>	<p>40.0 40.1 40.2 40.3 0.0 40.0 40.2 40.3 0.1 40.0 40.2 40.3 0.1 40.1</p>	

续表

电路	梯形图	程序指令码	程序数据	说明
		LD NOT AND NOT AND NOT AND SC LD NOT AND NOT AND NOT AND SC LD CLR	40.0 40.1 40.3 0.2 40.2 40.0 40.1 40.2 0.3 40.3 0.7 40	
顺序 工作 基本 电路		LD AND NOT AND NOT AND NOT SC LD AND SC LD AND SC LD AND SC LD AND CLR	0.0 40.1 * 1 40.2 * 1 40.3 * 1 40.0 0.1 40.0 * 2 40.1 0.2 40.1 * 2 40.2 0.3 40.2 * 2 40.3 0.7 40.3 * 2 40	* 1. 防止在其他步时, 顺序控制又从40.0步起动。 * 2. 前一步输出作为后一步的输入条件, 确保步进。

续表

电路	梯形图	程序指令码	程序数据	说明
步 进 动 作 扩 展		LD NOT AND NOT AND NOT AND NOT ∴ AND NOT SC LD AND SC ∴ LD AND SC LD AND SC CLR LD AND SC LD AND SC LD AND CLR	0.0 30.1 30.2 30.3 ∴ 31.7 30.0 0.1 30.0 30.1 ∴ 0.7 30.6 30.6 30.7 30.7 30.7 31.0 1.0 30.7 31.0 31.0 31.1 31.0 31.0 31.1 ∴ 1.7 31.6 31.6 31.7 1.1 31.0 31.0 31.1 ∴ 1.7 31.6 31.7 2.0 31.7 31.7 31	先对 30 卡进行步进， 30 卡 8 步完成以后再 对 31 卡开始进行步 控，待 31.0 完成以后， 对 30 卡进行清零。

续表

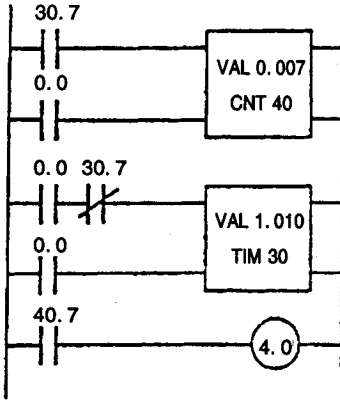
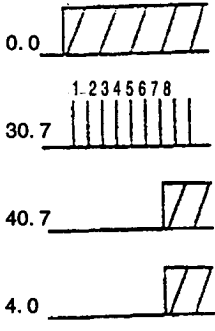
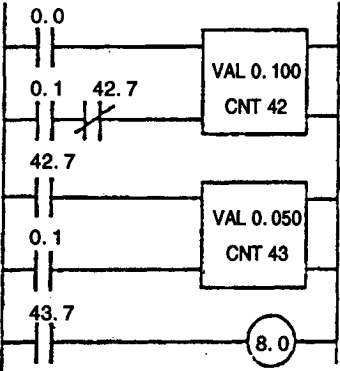
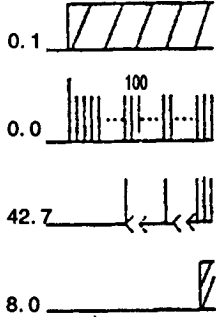
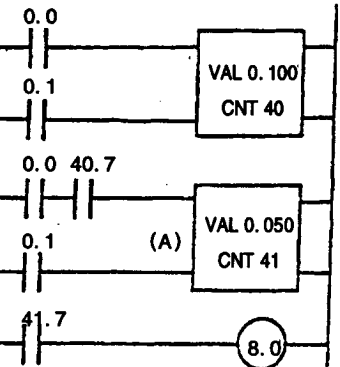
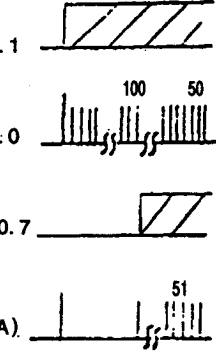
电路	梯形图	程序指令码	程序数据	说明
延 时 接 通 / 断 开		LD 0.0 LD 0.0 VAL 0.100 TIM 20 LD 5.0 AND NOT 0.0 LD 5.0 AND NOT 0.0 VAL 0.050 TIM 21 LD 20.7 OR 5.0 AND NOT 21.7 OUT 5.0	0.0 0.0 0.100 20 5.0 0.0 5.0 0.0 0.050 21 20.7 5.0 21.7 5.0	20号卡延时10秒,作为延时接通。 21号卡延时5秒,作为延时开断。  5号卡为输出继电器
单 脉 冲 电 路		LD 9.0 OR 0.0 AND NOT 20.7 OUT 9.0 LD 9.0 LD 9.0 VAL 0.020 TIM 20 LD 9.0 AND NOT 20.7 OUT 9.1	9.0 0.0 20.7 9.0 9.0 9.0 0.020 20 9.0 20.7 9.1	输出2秒单脉冲。  9号卡为输出继电器

续表

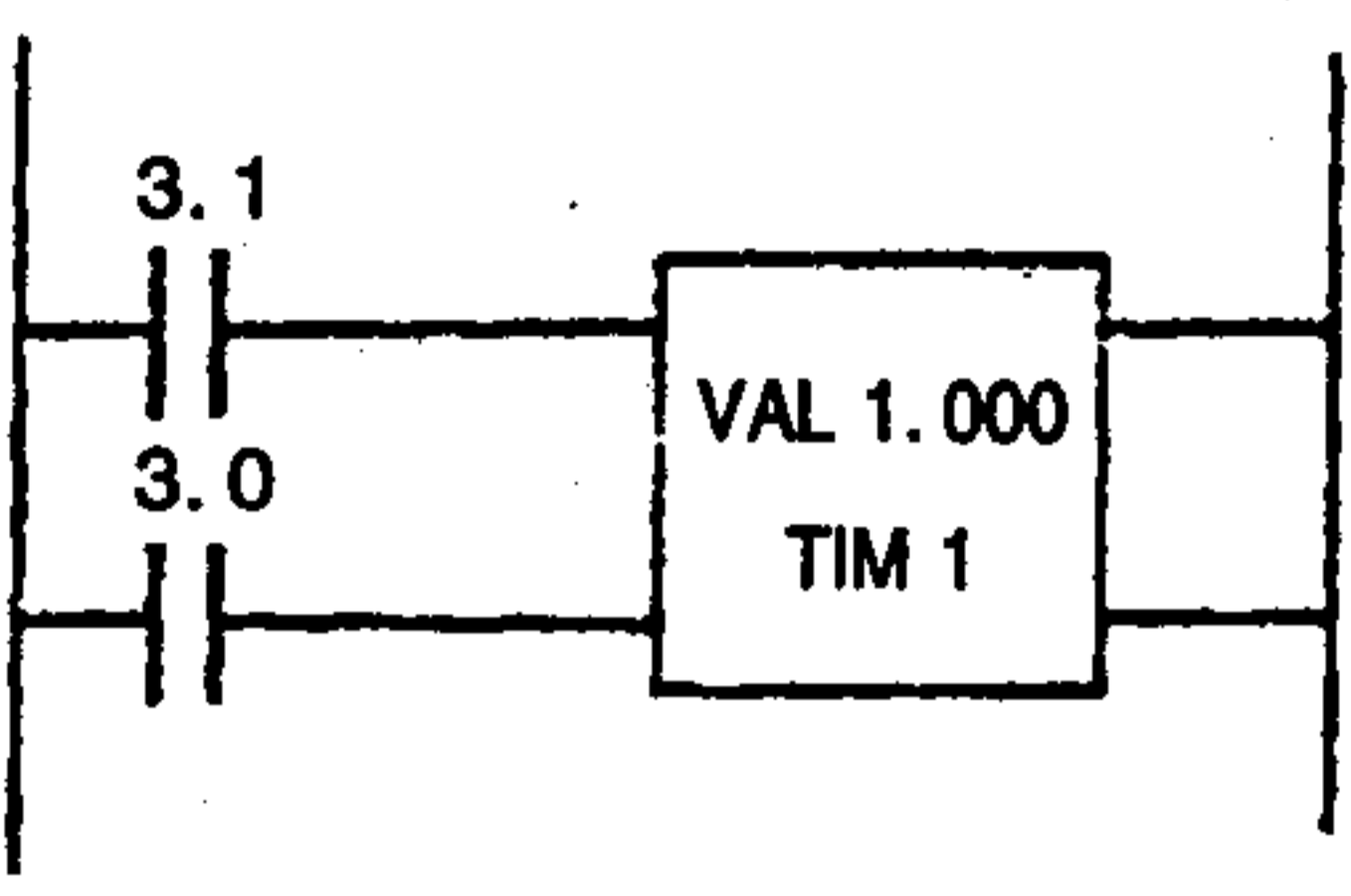
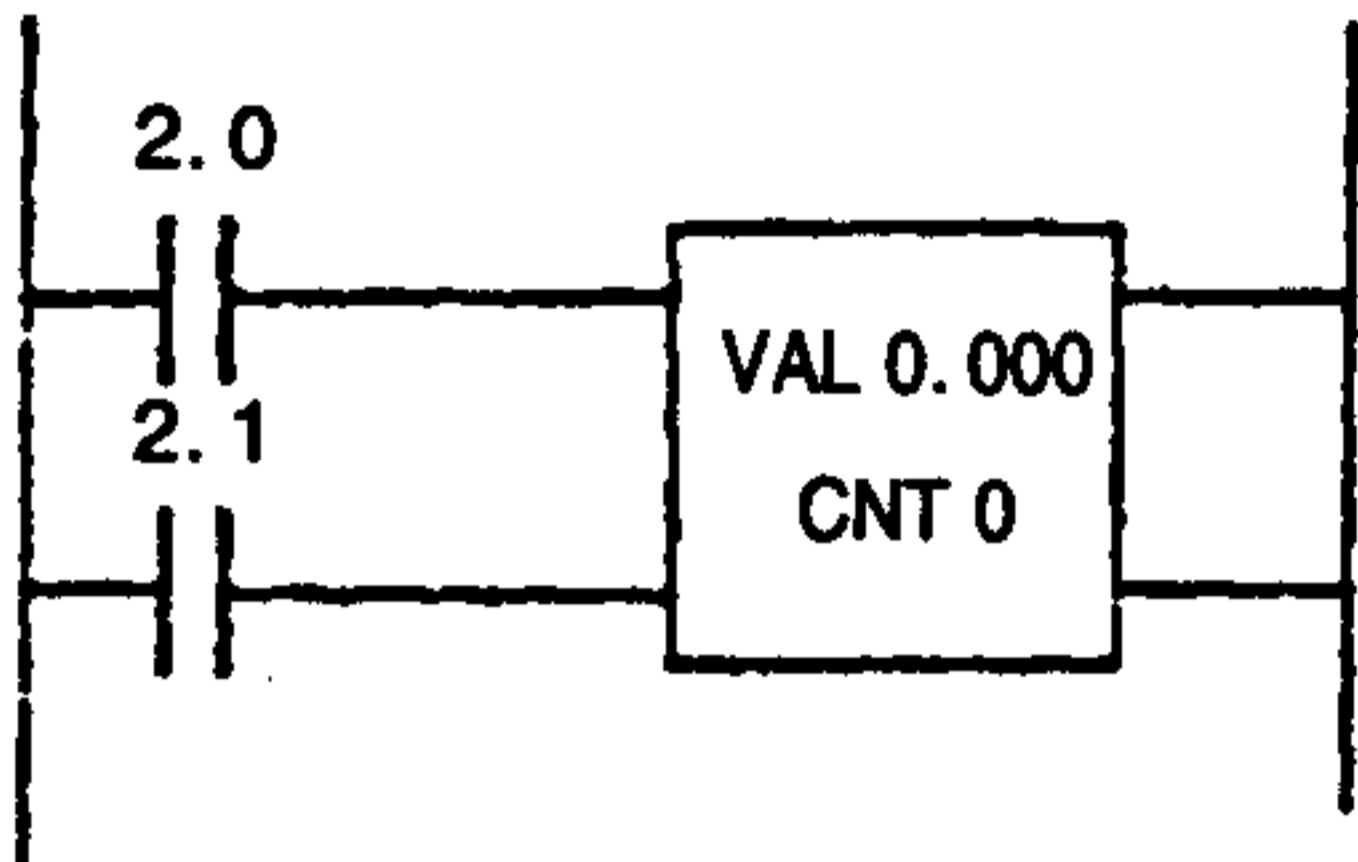
电路	梯形图	程序指令码	程序数据	说明
多谐振荡器(闪烁)		<p>LD 0.0 AND NOT 21.7 LD 0.0 AND NOT 21.7 VAL 0.010 TIM 20 LD 20.7 LD 20.7 VAL 0.030 TIM 21 LD 20.7 VAL 0.030 TIM 21 LD 20.7 OUT 5.0</p>	<p>0.0 21.7 20.7 21.7 21.7 20.7 20.7 5.0 3秒 1秒 5.0</p>	<p>0.0 21.7 20.7 21.7 21.7 20.7 20.7 5.0 3秒 1秒 输出3秒,间隙1秒。 5号卡为输出继电器</p>
定时器扩展 T ₁ + T ₂		<p>LD 0.0 LD 0.0 VAL 0.100 TIM 40 LD 40.7 LD 40.7 VAL 0.070 TIM 41 LD 41.7 OUT 7.0</p>	<p>0.0 40.7 41.7 7.0 10 7 17秒</p>	<p>0.0 40.7 41.7 7.0 10 7 17秒 7号卡为输出继电器</p>
定时器扩展 n × T		<p>LD 30.7 LD 0.0 VAL 0.007</p>		



续表

电路	梯形图	程序指令码	程序数据	说明
		<p>CNT LD AND NOT LD VAL TIM LD OUT</p>	<p>40 0.0 30.7 0.0 1.010 30 40.7 4.0</p>	 <p>4号卡为输出继电器</p>
<p>计数器 扩展 (积 n1 x n2</p>		<p>LD LD AND NOT VAL CNT LD LD VAL CNT LD OUT</p>	<p>0.0 0.1 42.7 42.7 0.1 0.050 43 43.7 8.0</p>	 <p>8号卡为输出继电器</p>
<p>计数器 扩展和</p>		<p>LD LD VAL CNT LD AND</p>	<p>0.0 0.1 0.100 40 0.0 40.7 0.0 40.7 0.0 51 40.7</p>	

续表

电路	梯形图	程序指令码	程序数据	说明
		LD VAL CNT LD OUT	0.1 0.050 4.1 41.7 8.0	8号卡为输出继电器
外 预 置 定 时		LD LD VAL TIM	3.1 3.0 1.000 1	1秒时标, 占用1号卡, 时间由外部拨盘定(0~99) 0号卡, 1号卡为输入继电器
外 预 置 计 数		LD LD VAL CNT	2.0 2.1 0.000 0	外部定时, 计数只能占用0及1卡。 外部拨盘接线如图2-30 0号卡, 1号卡为输入继电器

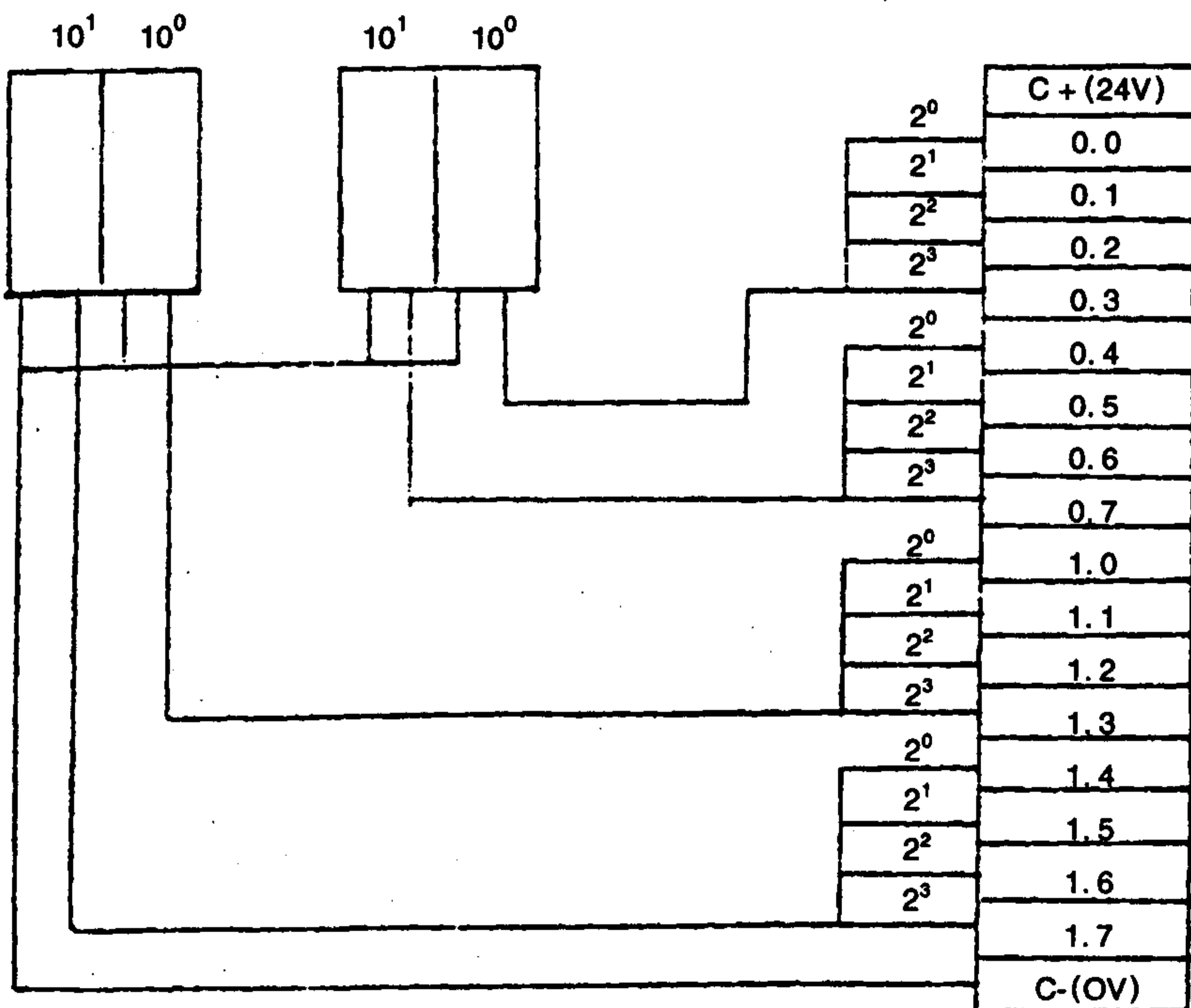


图 2—30 外部预置数的定时器、计数器接线图

### 三、步进过程控制应用举例

顺序控制可以很方便地利用步控制器语句 SC...CLR 来实现，下面举例从状态移位图→梯形图→编程。

#### 例 1

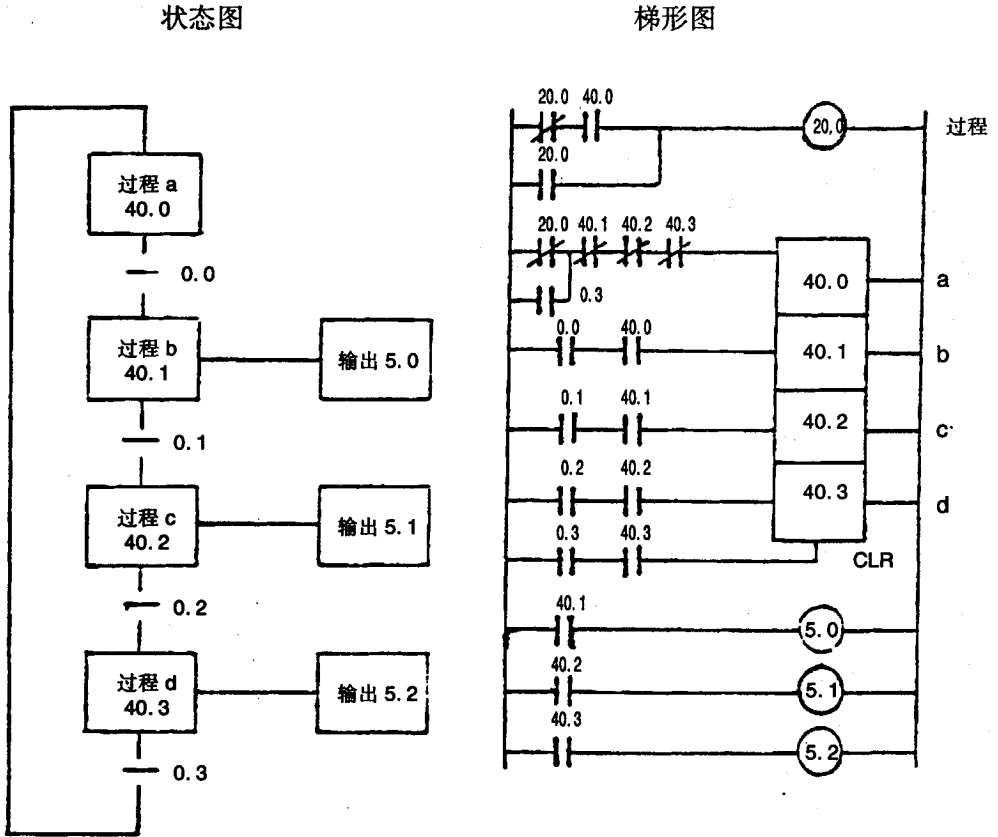


图 2—31

图 2—32

说明：

- 1、操作开始时,过程 a(初始状态)输入系统,即自启动。
- 2、当输入点 0.0 闭合时,控制步序到达 b 过程,输出点 5.0 闭合。
- 3、当输入点 0.1 闭合时,控制步序到达 c 过程,输出点 5.0 打开,输出点 5.1 闭合。
- 4、当输入点 0.2 闭合时,控制步序到达 d 过程,输出点 5.1 打开,输出点 5.2 闭合。
- 5、当输入点 0.3 闭合时,控制步序到达 a 过程,输出点 5.2 打开。
- 6、20.0 支路为开机自动启动,开机时 20.0、40.0 都没有输出故 20.0 支路不通,20.0 常闭触点闭合。40.0 有输出,20.0 支路通,20.0 有输出并自锁,一直到断电或进入编程状态时,20.0 断开。第一次循环结束,第二次循环依靠触点 0.3 接通启动。
- 7、采用步控器以后,由于它本身已经具备自锁、互锁的功能,在设计时可以无需担心因互锁、禁止等问题考虑不周而引起程序的错乱。

例 2

状态图

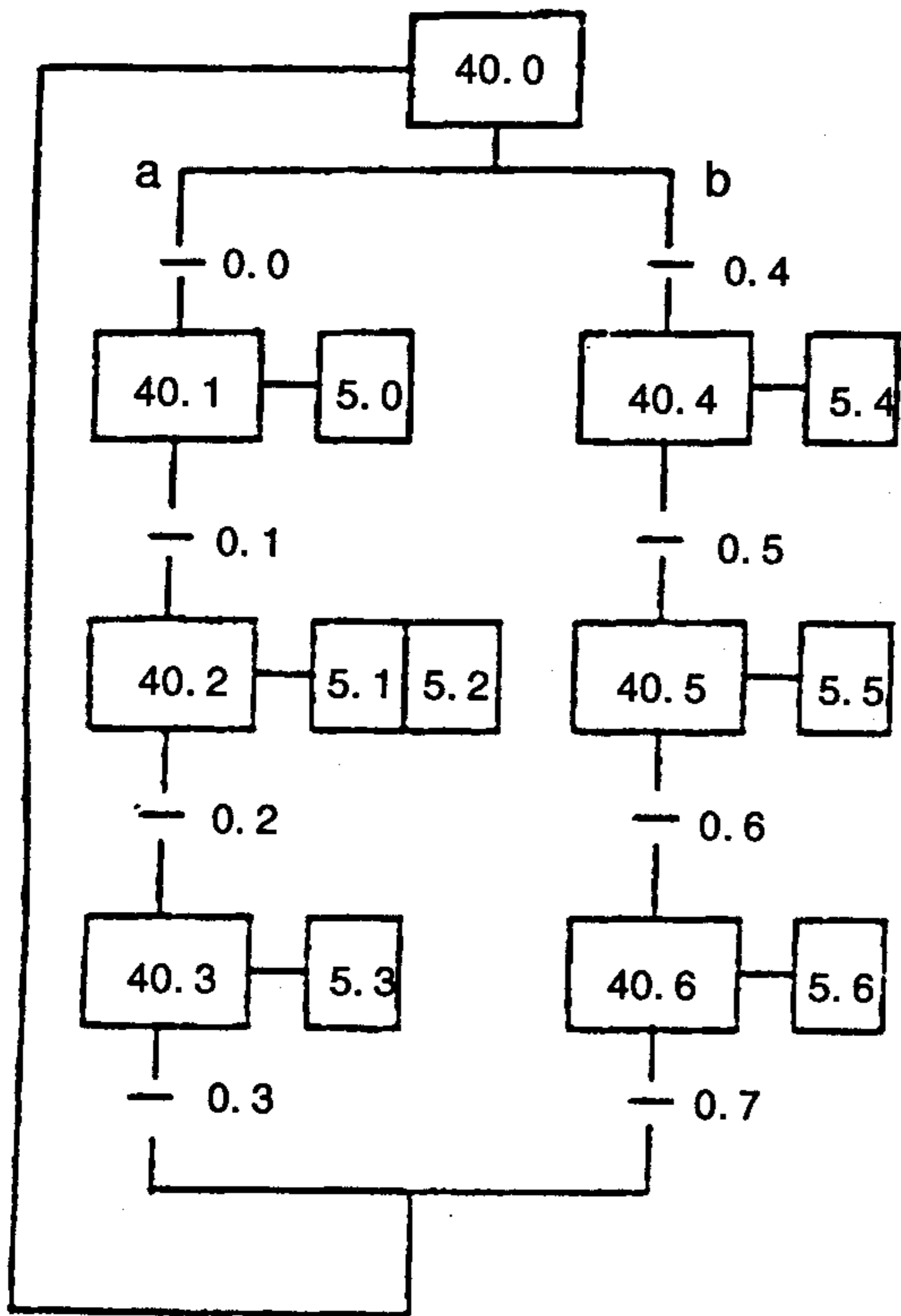


图 2—33

梯形图

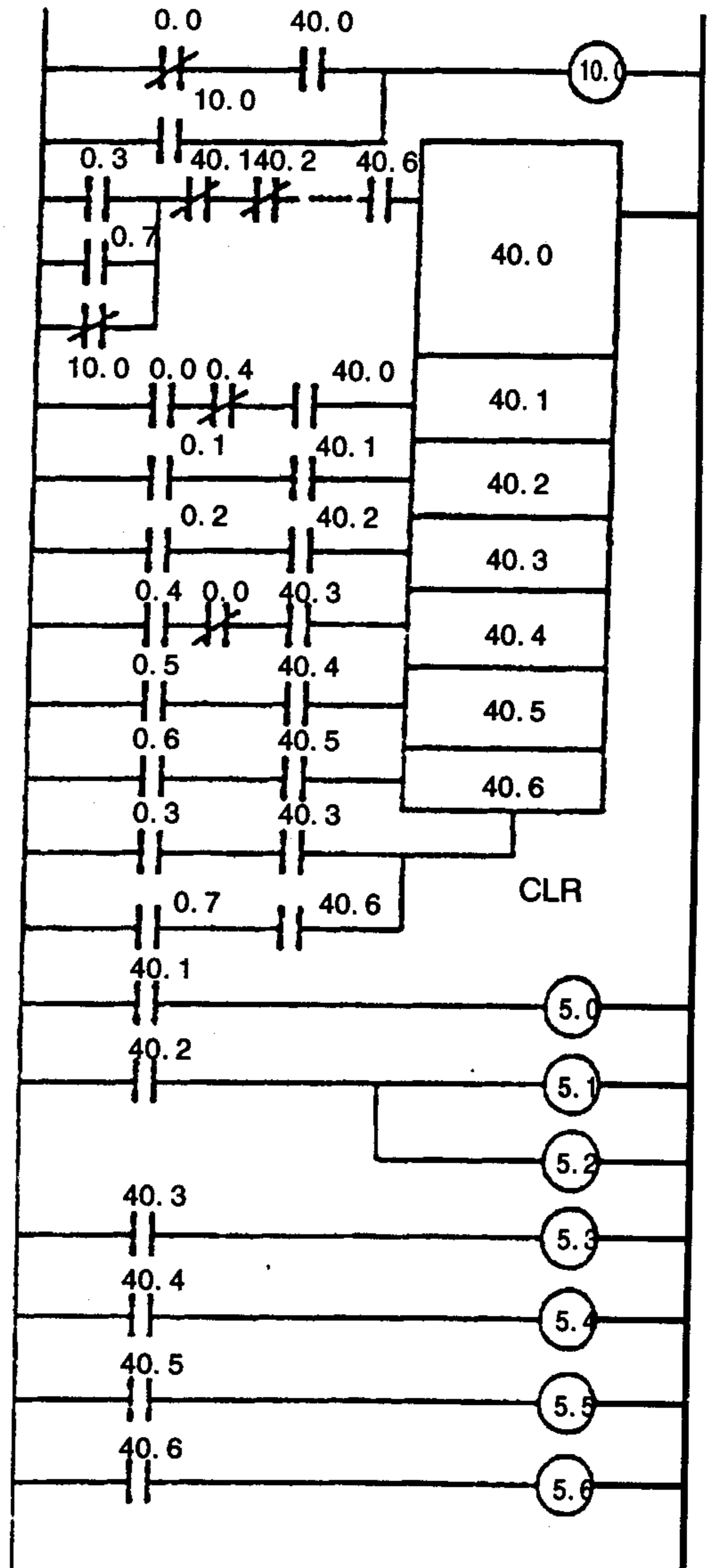


图 2—34

说明:

1、在运行开始后,进入初始状态,即自动起动。

2、当输入点 0.0 闭合 (ON) 时,按流程 a 执行,流程 b 不执行。

当输入点 0.4 闭合 (ON) 时,按流程 b 执行,流程 a 不执行。

执行 a 还是执行 b,取决于输入点 0.0, 0.4 是否接通。二个流程只能选择一个流程若 0.0, 0.4 同时接通,由于程序设计是互锁的,故二个流程都不执行。

3、流程 a: 按 40.1、40.2 和 40.3 的顺序步进,当输入点 0.3 闭合时 (ON) 控制将重新返回 40.0。

4、流程 b: 按 40.4、40.5、40.6 的顺序步进,当输入点 0.7 闭合时 (ON) 控制将重新返回 40.0。

状态图

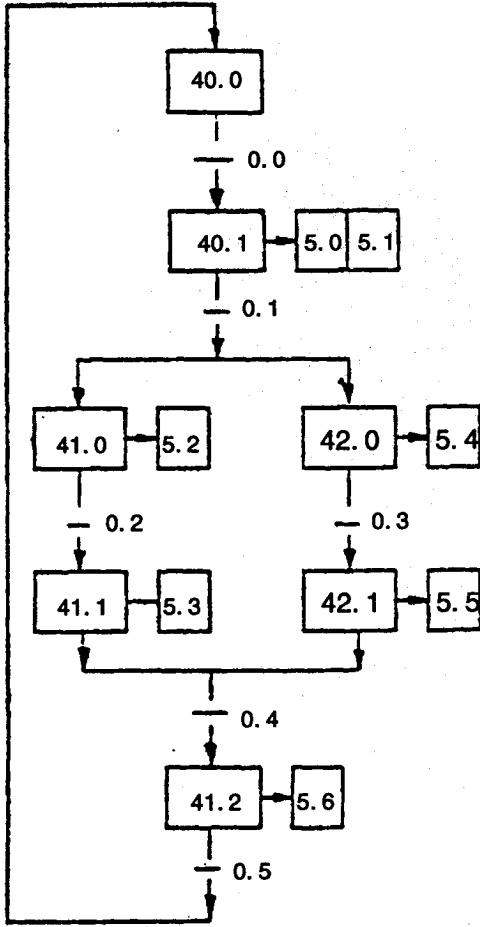


图 2—35

梯形图

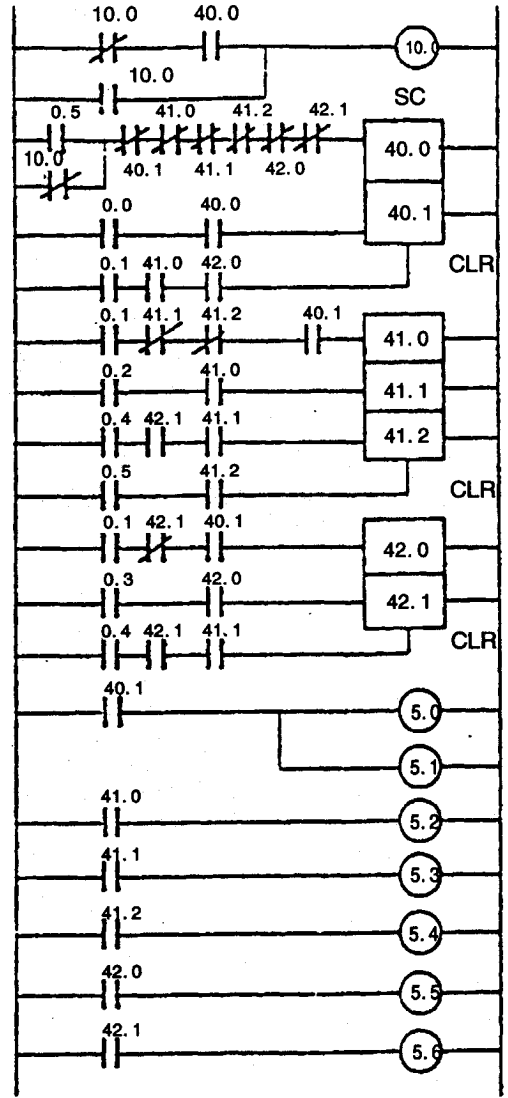


图 2—36

说明:

- 1、运行开始,进入初始状态(40.0通),即自动起动。
- 2、当输入点 0.0 闭合时(ON),控制进入 40.1 输入 5.0、5.1 闭合。
- 3、当输入点 0.1 闭合时(ON),控制同时进入 41.0 及 42.0, 流程 a、流程 b 同时执行。
- 4、流程 a 当输入点 0.2 闭合(ON)时,控制步进入 41.1。
- 5、流程 b 当输入点 0.3 闭合(ON)时,控制步进入 42.1。
- 6、当输入点 0.4 闭合,同时流程 a、b 分别从 41.1 和 42.2, 控制步进入 41.2。
- 7、在 41.2 进程,输入点 0.5 闭合,控制返回 40.0 重新开始一次新的周期。

# 第三篇 PRG—001 编程器使用手册

- 本手册提供了 PRG—001 编程器的操作方法
- 用户在使用前,请仔细阅读本手册
- 有关指令系统及 PC 的使用等详细内容请参见第一篇及第二篇

## 第一章 编程准备

### 一、概述

PRG—001 编程器为小型手续式编程器,在其上部有联接插座,通过扁平电缆和 PC 可编程控制器相联接,主要功能如下:

- * 通过编程器和控制器输入用户程序
- * 在线监视控制器的运行情况
- * 将控制器 RAM 区域里的用户程序写入 EPROM

PRG—001 编程器提供给用户的编程语言为指令表语言,且简便易学,每条语句在编程器上有明确的显示。

### 二、PRG—001 编程器键和显示灯

PRG—001 编程器,包括键盘及显示二大部分,键盘为 24 键的小键盘,用以输入各种语句及命令,显示部分不仅能显示程序步数和卡的号数,而且还具有控制器状态显示,指令显示,监控显示,错误类型显示等功能,其面板布置如图 3—1。

(一)键的分类:

#### 1、功能键

SHF   ↑   ↓   SCH   INS   DEL  
ADDR   CALL   CE   C   WRITE   EPROM   CHE  
CARD   ON   OFF

功能键的定义

(1) SHF 上档键

编程器上绝大数键分成上下两档功能,例如 

(A)
AND

 如果单独按下,则“AND”发光

二级管亮,若先按下 SHF 键,则“SHF”灯亮,再按下 

(A)
AND

 键,此时“DATA”数码管上

显示 [a],也就是 SHF 键的功能是有两种功能的上档功能。它的操作是按下 SHF 键,“SHF”灯亮,然后再按下你所需的键。

(2) **↑** 上翻键

在程序显示时,每按一次 **↑** 编程器显示上一个程序步号的程序内容,同时步号自动减 1,可连续操作,直到“0000”为止。

(3) **↓** 下翻键

在程序显示时,每按一次 **↓** 编程器显示下一程序步号的程序内容,同时程序步号自动加 1,此键可连续操作,直到“1000”为止,再继续按 **↓** 键,显示器不改变显示保持在 1000 步。

(4) **SCH** 检索键

此键可以对某指令或某指令码进行检索。

按下此键,编程器可以对用户程序进行检索,从现在所在程序步号开始,向下进行检索。若检索到第一时则自动停止,在“ADDR”显示检索到程序步号。再按 **SCH**,继续执行对某一指令码进行检索,若检索到程序末尾时,自动从头开始。

(5) **INS** 插入键

该键可以完成对用户程序插入操作,先找到需要插入程序步号,然后键入插入的内容按下 **INS**,即完成程序插入,以下程序步号自动加 1。

(6) **DEL** 删除键

该键可以完成对用户程序中某些指令的删除。首先找到需删除指令的步号,然后按下 DEL 键,则删除该指令,同时以下步号均自动减 1。

(7) **ADDR** 程序步号调用键

按下此键,“ADDR”显示“0000”,此时可输入程序步号。

(8) **CALL** 指令调用键

按下此键,则在“DATA”上显示“ADDR”所标明程序步号的指令数据,该步号的指令码由发光二极管显示。

(9) **CE** 清除键

该键有二功能,在显示指令时,按下此键,清除指令数据,即“DATA”上的数据。在编程器上显示“YOU SURE”字样时,按下此键,则对整个用户程序总的清除。

(10) **C** 总清除键

当要想对用户程序总清除时,可以按此键,当你按下 **C** 键,显示“YOU SURE”字样,此时用户程序仍在,你确认要清除时,再按下 **CE** 键,程序清除,若你按 **C** 键,不想清除,则按下“NOT”键,“YOU SURE”字样清除,程序没有被清除。

(11) **WRITE** 写入键

在编程时,每一条指令输入完毕,按下此键,则将指令写入“RAM”中,同时程序步号自动加 1。

(12) **EPROM** EPROM 写入键

本机主机具有 EPROM 写入功能,通过编程器的 **EPROM** 键来控制主机的 EPROM 写入。

(13) **CHE** 比较键

该键具有将主机中用户 EPROM 与 RAM 中用户程序进行比较功能,若一致,则显示“000”步号的指令,若不一致则显示“ERRd”错误代码。

(14) **CARD** 监控键

该键是在主机运行时使用,可对主机运行状态进行监视。

(15) **ON** 强制接通键

在监控状态时, **ON** 键可以对某一输出继电器强制接通。

(16) **OFF** 强制断开键

在监控状态时, **OFF** 键可以对某一输入继电器强制断开。

## 2. 指令键

LD	AND	OR	NOT	OUT	VAL	TIM	CNT	(A)	(O)
SC	SR	MCS	MCR	CLR	FUN	SET	RST	END	

每个指令键代表一个指令符,各个指令符的作用请见第二篇第二章指令系统介绍。其中 **FUN** 键目前未进行定义,留作开发新的功能时使用。

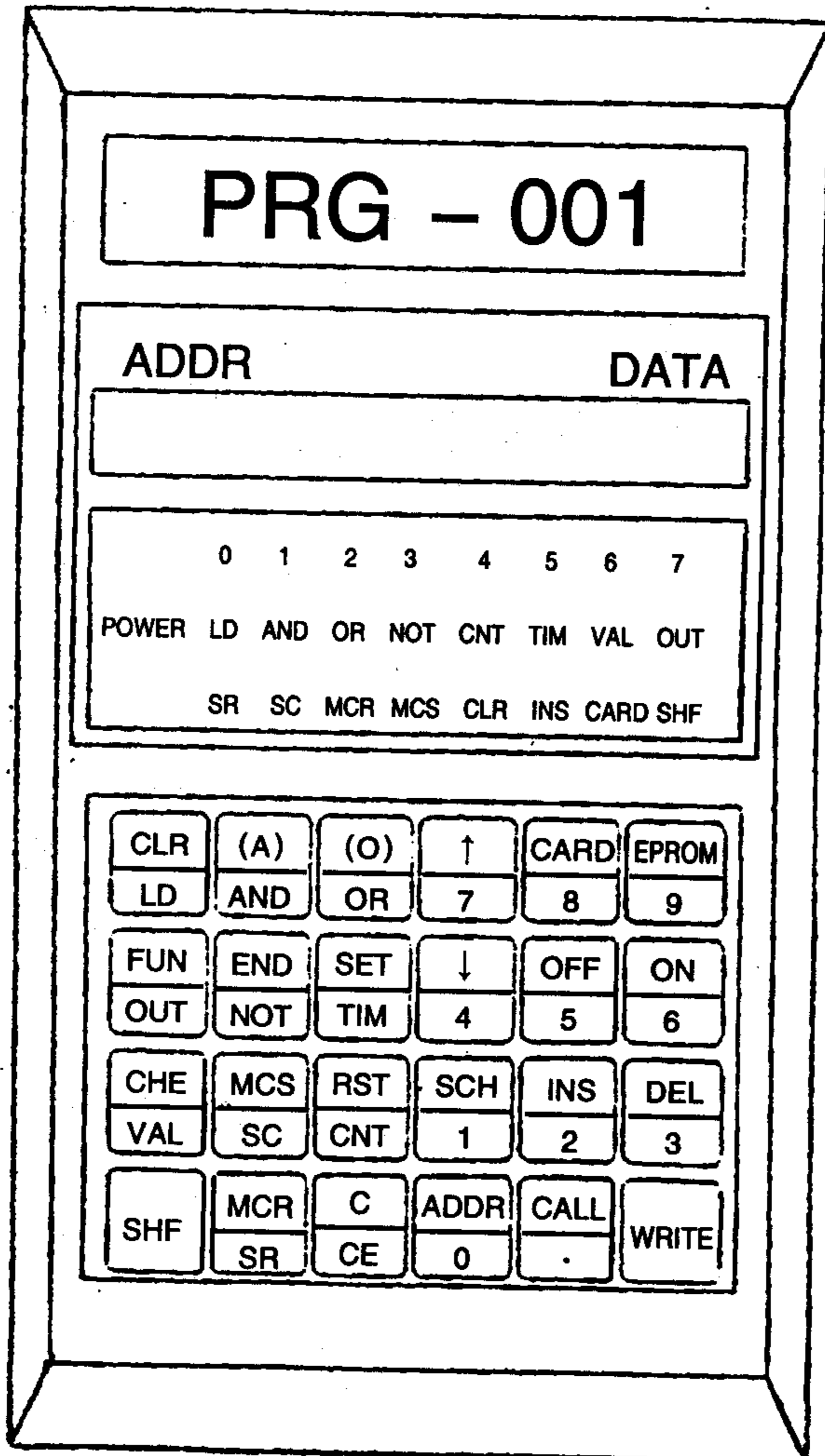


图 3—1 编程器外形图



3、数字键:0、1、2、3、4、5、6、7、8、9。

## (二)显示部分

编程器上显示部分设有 8 个数码管及 17 个发光二极管。

1、ADDR 指示下的四个数码管,用于显示程序地址(即程序步号)及在线运行时定时器、计数器的运行值。

2、DATA 指示下的四个数码管用于显示指令数据、错误类型及监控时的卡号。

3、17 个发光二极管除上排最左边一个作为编程器电源指示外,其余 16 个分上下两排,作为指令码及操作命令的显示。上面一排发光管还可在监控时显示被监视卡的运行状态。

## 三、PRG—001 编程器的功能

当准备编制新程序或检索、修改旧程序及在线监视 PC 控制器运行状态时,必须将 PRG—001 编程器联接到控制器上。

- * PRG—001 可以带电插拔、无须关掉控制器电源。

- * PRG—001 的工作状态由控制器上的状态开关决定,控制器的状态开关有编程“PRG”及运行“RUN”两种位置。

- * 状态开关处于“PRG”状态时,通过编程器可以进行程序存贮区及数据区内容的清除,程序写入、程序修改、检索、显示、插入、删除以及 EPROM 写入等操作。

- * 状态开关处于“RUN”状态时,通过编程器可以完成对控制器运行状态监控,如输入输出点的状态监视,强制接通/断开输出继电器等。

## 第二章 编程操作指导

本章主要介绍用户程序的输入、检索及修改的步骤和方法,大致框图如图 3—2

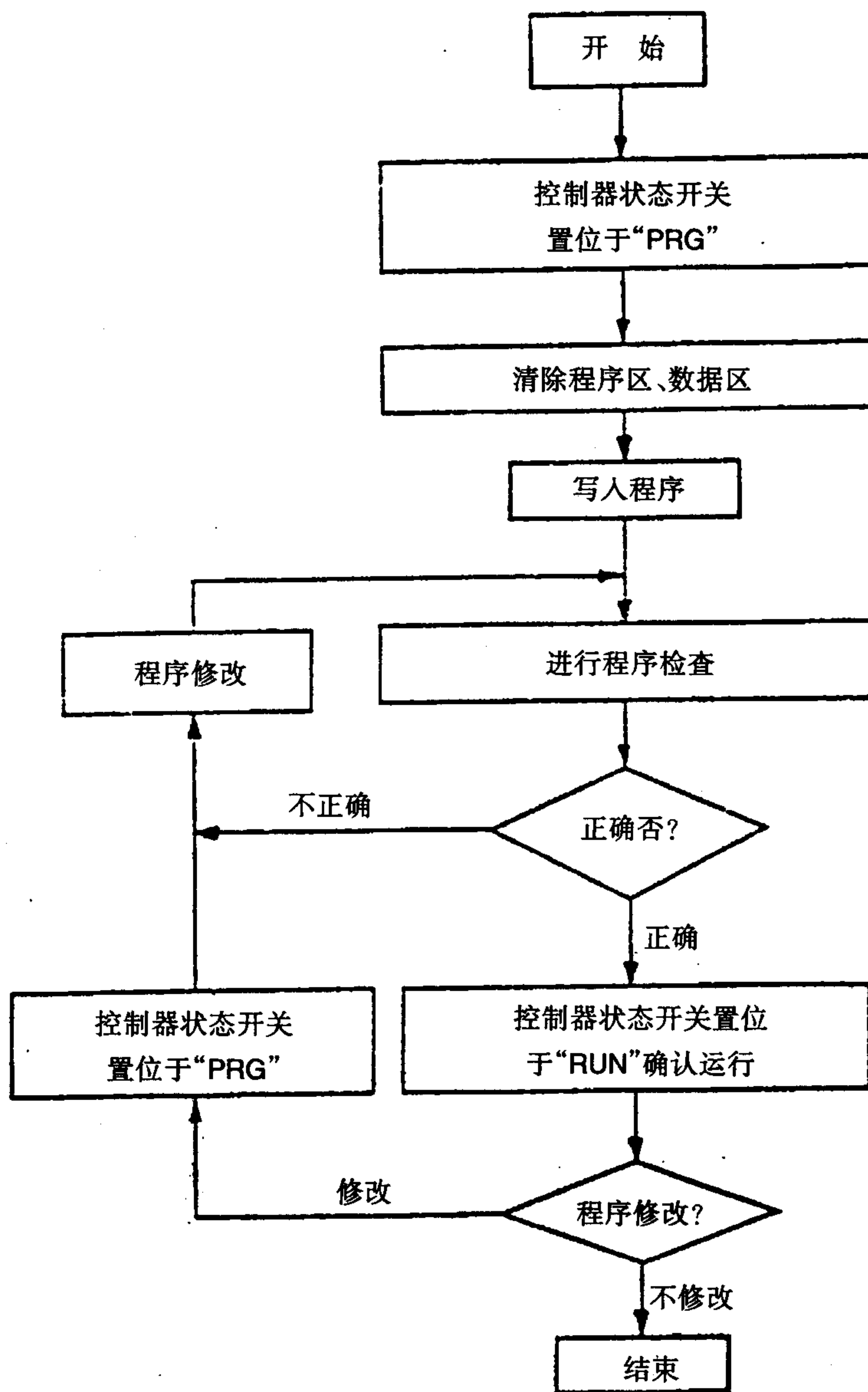


图 3—2

### 一、程序区及数据区的清除

在写一新程序之前,首先执行本操作。它清除程序存贮区里的所有程序及内部的全部数据区(包括带停电记忆的区域)。步骤如下:

- 1、确认状态开关处于“PRG”位置,否则不予执行清除命令。
- 2、按动上档转换键 **SHF**, SHF 灯亮。表示接受上档功能。

3、按动  $\frac{C}{CE}$  键,此时编程器显示“YOU SURE”字样。

4、如确认要清除,再次按动  $\frac{C}{CE}$  键,则清除完成,并在编程器上显示“CLeared”字样,并等待接受程序步调出命令。

5、如不想清除,不执行第4条而直接按动  $\boxed{NOT}$  键,编程器上显示第0步的内容。

## 二、程序步的调出与程序的显示

当你需要重新写入或修改、显示程序时,本操作是必不可少的一个环节。

第一种状态是在上电及总清除后,编程器显示“PC—80”或 CLeared”时。

1、按动数字键,输入需要调出的程序号。

2、输入的程序号在“ADDR”指示下的四个数码管上显示,输入的数字依次从右向左移动,如输入程序步号不正确,可连续按动  $\boxed{O}$  键,再输入正确的程序步号。

3、正确的程序步号输入后,按动  $\boxed{CALL}$  键,则调出所需步号及该步内容。

4、调出的程序其步号在“ADDR”指示的四位数码管上显示,而其内容通过“DATA”指示的四位数码管及发光二极管显示。

5、如输入程序步号超过程序步号范围(0~1000步)则编程器上显示错误信息“ERR4”。

6、可按动  $\frac{C}{CE}$  键,清除 ERR4,重新输入程序步号。

第二种状态是在程序显示、修改、写入等状态时:

1、按动  $\boxed{SHF}$  键,SHF 灯亮。

2、按  $\boxed{ADDR}$  键,编程器“ADDR”部分显示“0000”,此时输入所需调出的程序步号。

3、重复上述 1~6 的操作。

当程序调出后可以连续地显示上一步的内容。其操作如下:

1、按动  $\boxed{SHF}$  键,SHF 灯亮。

2、按动  $\boxed{\uparrow}$  键,则程序步减 1,显示上一步的内容,连续按动,连续显示,每次程序步减 1。

3、按动  $\boxed{\downarrow}$  键,则程序步加 1,显示下一步的内容,连续按动,连续显示,每次程序步加 1。

## 三、程序写入

程序写入的流程如图 3—3

1、按错键时的修正方法。

a、指令键时

(1)同一键按二次,该指令清除。

(2)由于后人的指令优先,所以在按错的指令之后,再按正确的指令即已修正过来。

b、置数键时

按  $\boxed{CE}$  键后,再置入正确数据。

2、不按动  $\boxed{WRITE}$  键,缓冲区的程序不存贮到控制器的程序存贮区里,因此没必要担心按错键,用上述方法修正即可。若按了  $\boxed{WRITE}$  键,则可以按程序修改方法处理。

### 3、写入程序

写入程序时,首先将梯形图写出梯形图语言,整个程序步号从 0 号开始,每一条指令结束,按下 **WRITE** 后在 ADDR 部分显示下一条指令的步号。

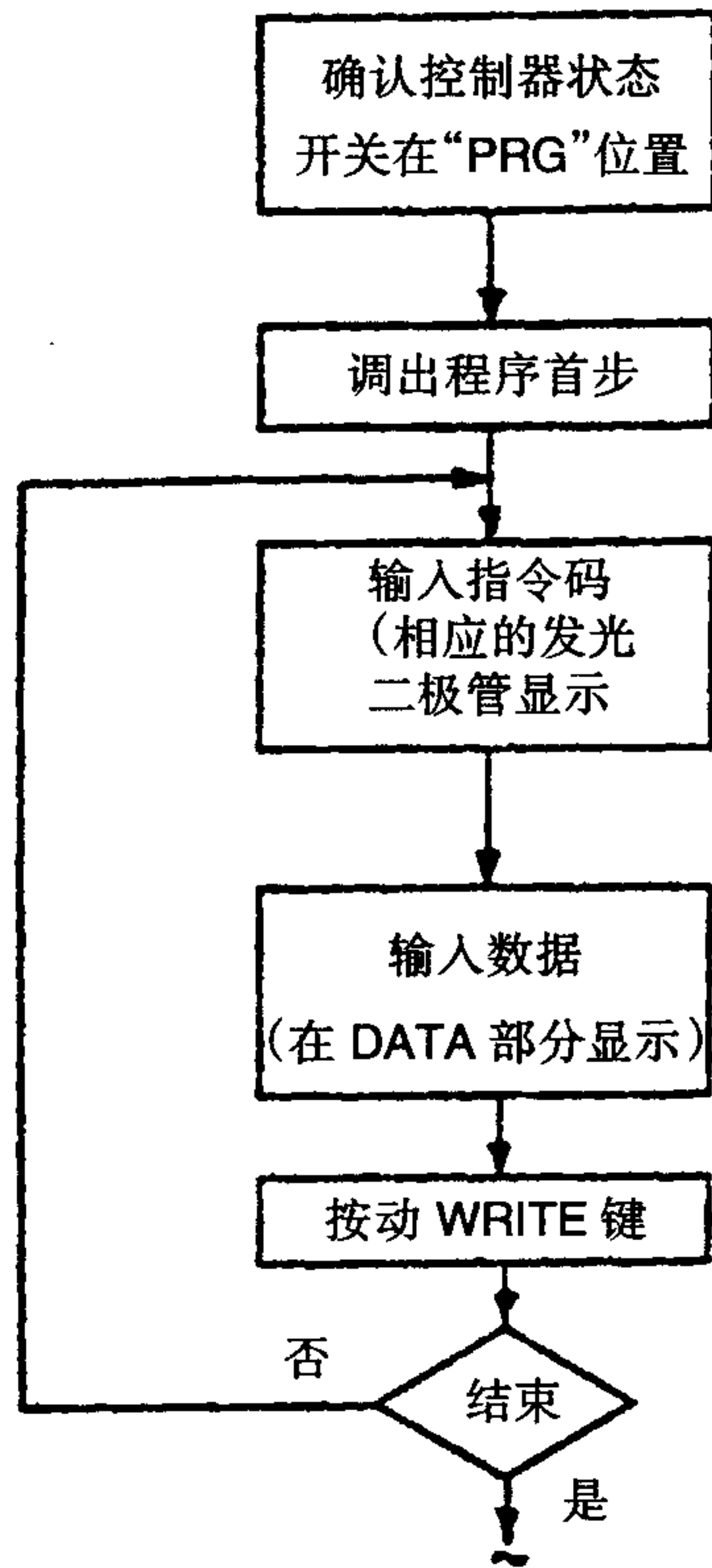
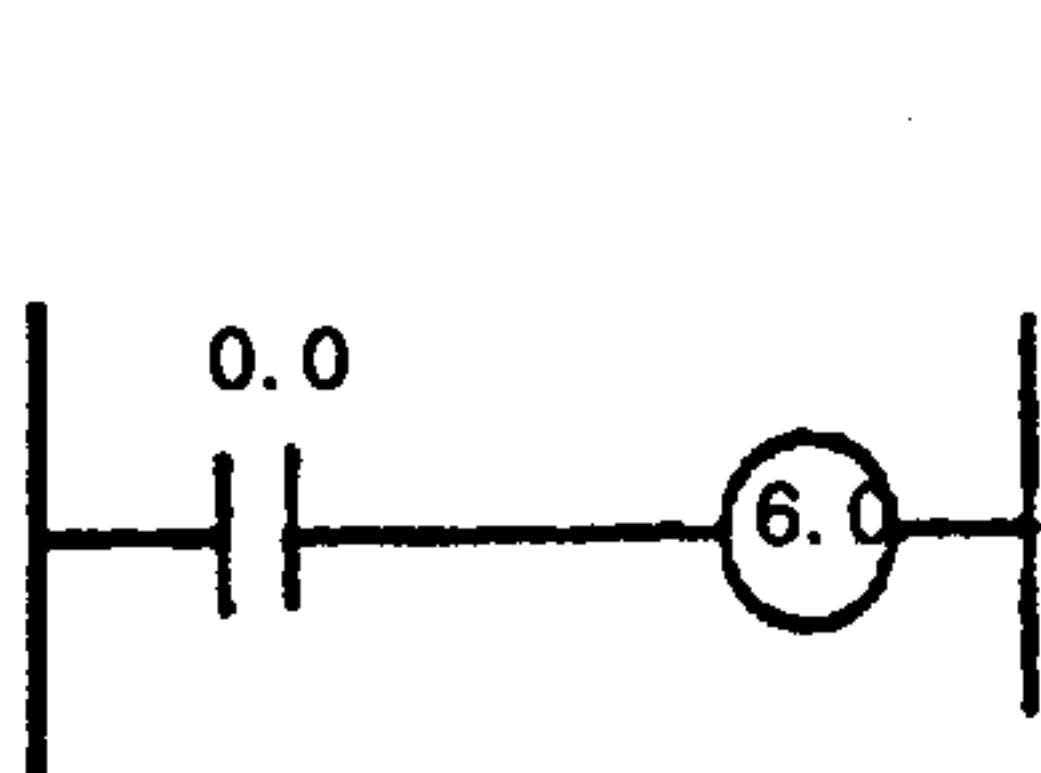


图 3—3 程序写入流程



步号	指令
0	LD 0.0
1	OUT 6.0

图 3—4

#### 键操作

LD	LD 指示灯亮
0	DATA 上显示 0
.	DATE 上显示 0.
0	DATA 上显示 0.0
WRITE	ADDR 显示 0001
OUT	OUT 指示灯亮
6	DATA 上显示 6
.	DATA 上显示 6.
0	DATA 上显示 6.0
WRITE	ADDR 显示 0002

### 四、程序更改

当需要对程序中某语句进行修改时,请按下列步骤进行。

- 1、确认控制器状态开关在“PRG”位置,插入编程器。
- 2、按动 **SHF** 键,使 SHF 灯亮。
- 3、按动 **ADDR** 键。
- 4、输入需要修改语句的步号。

5、按动 **CALL** 键,调出须修改的语句。

6、若是修改指令码,按动需要的指令键,即可修改。若有多余的指令码,则重复按该指令码键一次即可。

7、若是修改数据,按动 **CE** 键,消去原有数据,重新输入你需要的数据。

8、确认修改的语句无误后,按动 **WRITE** 键,则修改后的语句存入控制器内的程序存贮区里。

此时,该语句修改完毕。若需要继续修改,可以按照上述步骤重复操作。也可按动 **↑**、**↓** 键寻找所需语句,然后执行上述步骤的 6~8 即可。

若用户已经使用固化在 EPROM 中用户程序,必须将用户 EPROM 取下,否则的话在运行时仍然执行原程序,并将已经修改的程序又改变到原来的程序内容。

例:原有某语句形式如下:

步号 指令

30 AND NOT 0.2

修改后成为

30 OR 1.2

其操作过程如下:

1、插入编程器,控制状态位置至“PRG”状态。

2、按动 **SHF** 键,使 SHF 灯亮。

3、按动 **ADDR** 键,ADDR 数码管显示 0000。

4、按动 **3**、**0** 键,ADDR 数码管显示 0030。

5、按动 **CALL** 键,显示原程序,即 AND、NOT 灯亮“DATA”部分显示 0.2。

6、按动 **OR** 键,则 AND 灯灭,OR 灯亮。

7、按动 **NOT** 键,则 NOT 灯亮。

8、按动 **CE** 键,消去“DATA”部分的数据。

9、按动 **1**、**.**、**2**、键,“DATA”部分显示 1.2。

此时,“ADDR”部分显示 0030,“DATA”部分显示 1.2

上述修改,确认无误后,进行下一步操作。

10、按动 **WRITE** 键,将程序存入存贮区,编程器上显示下一条指令的步号。

修改结束。

## 五、程序插入

执行一次本操作之后,被插入语句后面的各语句其步号自动加 1。其操作过程与修改语句相似,只是在确认无误以后不按动 **WRITE** 键,而是按动 **SHF**、**INS** 键。操作步骤如下:

1、调出要插入语句的步号。

2、按动 **CE** 键,消去 DATE 部分显示的原内容。

3、输入要插入的语句的内容。

4、确认无误后,按 **SHF** 键,使 SHF 灯亮。

5、按动 **INS** 键,则语句插入到原先的程序之中,同时 ADDR、DATE 显示内容和 INS 发光二极管闪亮一次。连续执行几次本操作,可以连续插入几条语句。

上例中,如要在 30 句之前插入 OR1.2 则步骤如下:

#### 1、调出 30 语句

- a、按动 **SHF** 键,使 SHF 灯亮。
- b、按动 **ADDR** 键,ADDR 数码管显示 0000
- c、按动 **3**、**0** 键,ADDR 数码管显示 0030
- d、按动 **CALL** 键,此时在“DATA”部分显示 0.2,AND、NOT 灯亮。

#### 2、输入需要插入的语句

- a、按动 **OR** 键,使 OR 键亮,由于此时 NOT 灯仍亮,须清除 NOT。
- b、按动 **NOT** 键,使 NOT 灯灭。
- c、按动 **CE** 键,清除“DATA”部分的原数据。
- d、按动 **1** · **2** 键。

此时显示 30 OR 1.2。以上的步骤与语句修改一样。

#### 3、语句的插入

- a、按动 **SHF**,使 SHF 灯亮。
- b、按动 **INS**,使 ADDR、DATA 显示内容和 INS 灯闪亮一次;此时语句已插入,并显示下一条语句;31AND NOT 0.2。原 30 句已变成 31 句,以后依此类推,实现了插入后的自动排列。

### 六、程序删除

执行一次本操作后,被删除语句之后的所有语句步号自动减 1。

#### 1、调出所需删除的语句。

2、按下 **SHF** 键,SHF 灯亮。

3、确认无误后,按下 **DEL** 键,则该显示语句被删除。

4、被删除语句之后的语句依次上移,编程器上显示紧跟被删除语句之后的那条语句。

5、连续几次按动 **DEL** 键,可以连续删除几条相连的语句。和插入相似,实现了删除后的自动排列。

### 七、程序检索

程序的检索分为两种情况。当所需检索的内容已知在程序里存在,一种是检索程序中是否还有相同的语句。另一种是检索程序中是否有相同操作数。

二种检索方法,分别介绍如下:

甲、已知程序中某步号的指令语句,检索是否还存在相同的语句。

#### 检索方法

(1)用程序调出方法调出需查找的某步号的语句。

(2)按动 **SHF** 键,SHF 灯亮。

(3)按动 **SCH** 键,则进行检索,遇有和调出语句相同的情况,则在编程器上显示步号及内容,并暂停检索。SHF 灯不灭。

(4)再次按动 **SCH** 键(SHF 灯不灭),继续寻找,当检索至程序末尾时,自动从头开始。

(5)可以多次按动 **SCH** 键,不断地检索。

(6)如果没有找到相同语句时,编程器上仍显示原先的语句。

乙、已知程序中步号的指令语句,检索是否还存在相同的操作数。

检索方法:

(1)用程序调出法,调出需要查找的某步号的语句,并清除指令码。

(2)按动 **[SHF]** 键,SHF 灯亮。

(3)按动 **[SCH]** 键,则进行检索,遇有和调出语句操作数相同的情况,则在编程器上显示步号及操作数,不显示指令码。SHF 灯保持不灭。

(4)可连续操作 **[SCH]**,继续寻找。当检索到程序末尾时,自动从头开始。

(5)当整个程序区搜索完而没找到相同语句时,编程器上仍显示原先的语句。

## 八、程序从 RAM 写入 EPROM 或 EEPROM

当输入到控制器里的用户程序,经过修改,调试无误后,可以将其从内部 RAM 程序存贮器写入 EPROM 存贮器以作永久保存,固化前,先将空白芯片 2764A 插在 CPU 板上的 D13 位置上,插入时注意芯片缺口方向必须与插座一致,然后将 CPU 板插入槽中,接通电源。

其余操作过程如下:

1、确认控制器状态开关在“PRG”位置上。

2、拔下靠近 CPU 的一块输入模块。

3、按动上档键 **[SHF]**,SHF 灯亮。

4、按动 **[EPROM]** 键,在 DATA 显示部分最右边二位显示“EP”字样,SHF 灯灭。

5、将控制器母板上开关(S2)向上推,CPU 板上 EPROM 灯亮。

6、再次按动 **[EPROM]** 键(此时不需要按动 **[SHF]** 键),这时“ADDR”显示部分最左边有一闪动的“日”字样,表示已进入写入 EPROM 状态。

7、当用户程序全部写完后,数码管显示部分中间显示“00”。

8、将控制器母板上开关 S2 向下推,CPU 板上 EPROM 灯灭。

9、直接按动 **[CHE]** 键,对写入 EPROM 的内容进行正确性检验。

如果程序写入有错,则在编程器 DATA 部分显示“Errd”字样。

如果写入正确,则在 ADDR 部分显示“0000”步号,DATA 部分显示该步内容。此时已退出写 EPROM 操作。

10、如果不按 **[CHE]** 键。而按动 **[CALL]** 键,则跳过 EPROM 检验,直接在编程器上显示第 0 步的语句内容,进入程序写入、修改等操作状态。

请注意本控制器 EPROM 芯片采用 2764A,写入电压为 12.5V。

插入 CPU 板 D13 位置上待写的 EPROM 芯片,必须是高质量的空白 2764A(即新片子或已用紫外线擦除干净的片子),否则此种操作不会成功。

另外要引起注意的是,只有 EPROM 写入成功,才能把控制器开关从“PRG”位置推向“RUN”位置,如果 EPROM 写入不成功,请不要把控制器开关推向“RUN”位置,否则 RAM 里的程序会被该 EPROM 不正确的内容所替代,而造成失误。一旦 EPROM 固化完毕,将拔下的输入模块再插入槽中。

注 1:如果采用 EEPROM2864A 作为用户程序存贮器时,固化操作步骤可以省去上述第“2”、“5”、“8”项步骤。

注 2:在 PC40 上固化程序时,方法同上,但用户程序位于 CPU 板上“2D5”处。

## 第三章 监控功能

当控制器在运行过程中,通过编程器可以对控制器的运行状态进行监视。

### 一、输入、输出状态的监视

此种监视功能不仅显示控制器实际输入输出单元的内部映象的状态,而且也显示控制器内部继电器的状态,使得用户可以随时了解整个控制器各继电器的运行状态。监视对象以卡为单位,操作一次,监视一卡,卡的状态在编程器第一排发光二极管上显示。发光二极管上面的数字 0~7 位依次对应于一个卡里的 0~7 位。操作步骤如下:

1、在控制器处于运行状态时,即状态开关位于“RUN”一侧。

2、按动 **CARD** 键(不必按动上档键 **SHF**),CARD 灯亮。

3、按动数字键,输入需监视单元的卡号,在“DATA”显示部分最右边显示输入卡号。如输入错了可以连续按动正确的卡号,显示部分自动消去错误卡号。

4、按动 **SHF**、**CALL** 键,则第一排发光管上显示“DATA”部分所显示卡号的继电器运行状态,灯亮表示该位继电器接通。

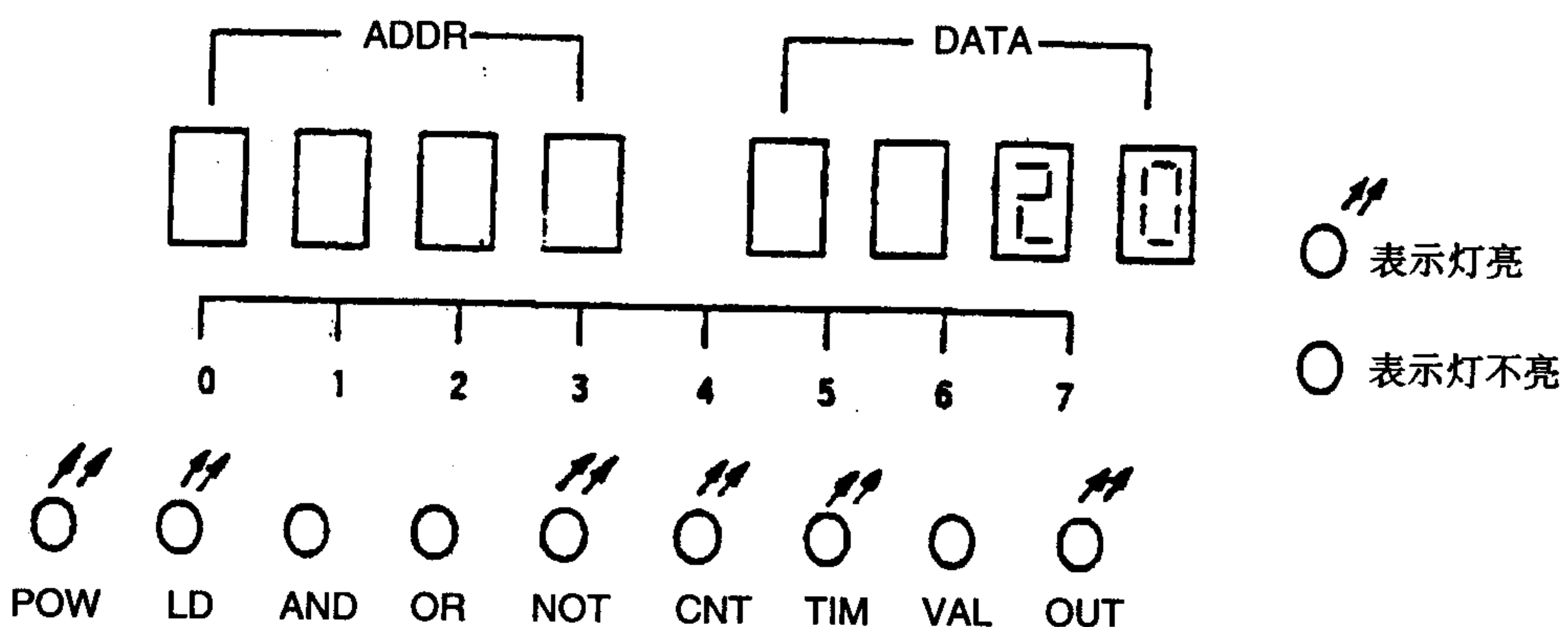
操作例子:监视 20 卡的继电器运行状态。

按动 **CARD**、**2**、**0**、**SHF**、**CALL**

若此时 20.0~20.7 继电器的运行状态和监视显示状态如下:

继电器号	20.0	20.1	20.2	20.3	20.4	20.5	20.6	20.7
状态	通	断	断	通	通	通	断	通

此时编程器上显示:



移位寄存器、步控器因为以卡为单位,因而用此操作也可以监视移位寄存器、步控器的运行状态。

### 二、定时器的现时值监视

在运行过程中如要监视定时器运行状态,可以通过本操作的执行达到目的。

命令输入后,被监视的定时器号显示在“DATA”部分的最右二位,而运行的现时值显示在“ADDR”部分,显示的数值为尚存计时值,每过一个计时单位,显示值减 1,当定时器定时到达,在 ADDR 部分最左边显示“1”字样。

1、控制器在运行状态,即控制器状态开关在“RUN”一侧。

2、按动 **CARD** 键,CARD 灯亮(不必按动 **SHF** 键)。



3、输入要监视定时器的卡号,卡号在 DATA 部分显示。

4、按下 **TIM** 键, TIM 灯亮。

5、按动 **SHF**、**CALL** 键, 则定时器的现时运行值在 ADDR 部分显示出来。

6、如输入卡号超出范围, 则显示“Er”字样, 此时按动 **CARD** 键, 重新开始即可。

### 三、计数器的现时值监视

计数器监视的操作方法及显示形式均与定时器类同, 仅在操作第 4 步时不是按动 **TIM** 键, 而是按动 **CNT** 键。

操作命令输入后, 被监视的计数器号在“DATA”部分最右边二位显示, 计数器的现时值显示于“ADDR”部分, 每来一个计数脉冲, 显示值减 1, 当减为 0 时, “ADDR”部分最左一位显示“1”, 表明计数器计数已到。

1、控制器处于运行状态。

2、按动 **CARD** 键, CARD 灯亮。

3、输入要监视计数器的卡号, 卡号在 DATA 部分显示出来。

4、按下 **CNT** 键, CNT 灯亮。

5、按动 **SHF**、**CALL**, 则计数器的现时运行值在 ADDR 部分显示出来。

6、如输入卡号超出范围, 则显示“Er”字样, 这时按动 **CARD** 键重新开始即可。

### 四、强制 ON/OFF

采用强制 ON/OFF 操作, 不同外部输入信号控制, 而用编程器即可强制操作继电器的 ON、OFF。

如图 3—5 电路里, 采用强制 ON、OFF 对 4.0 进行操作, 操作过程如下:

1、先输入如图用户程序, 然后将控制器处于运行状态, 即状态开关在“RUN”一侧。

2、按动 **CARD** 键, CARD 灯亮。

3、按数字键 **4**、**.**、**0**

4、按 **SHF**、**CALL** 键, 调出 4.0 的状态。

该状态显示在“ADDR”部分的最低位。

5、按动 **ON** 键, 该位显示“1”, 表示 4.0 通。

按动 **OFF** 键, 该位显示“0”, 表示 4.0 断开。

强制 ON/OFF 是瞬间性进行的, 对于上述的图示中, 由于它本身是自锁电路, 故强制 ON 以后可以保持, 强制 ON 以后, 只有当 0.1 接通或强制 OFF 时, 4.0 才断开。当 0.0 接通, 0.1 断开时, 此时 4.0 处于接通状态, 此时采用强制 OFF, 最长只能维持 1 个扫描周期, 由于每一次扫描周期读入的 0.0、0.1 状态又使 4.0 接通。对于图 3—6 所示电路, 对 4.0 进行强制 ON、OFF 操作, 都只能最多维持一个扫描周期, 瞬间性地进行 ON 或 OFF。

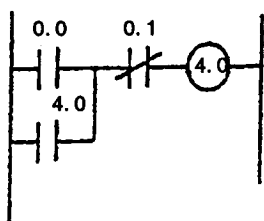


图 3—5

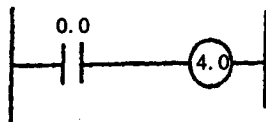


图 3—6

# 第四篇 PC40 小型编程控制器 使用手册

PC40 可编程序控制器与 DK—PC80 可编程序控制器的使用方法基本相同,系统性能和指令功能与 DK—PC80 也相同,编程器与 DK—PC80 通用。用户在编程过程中可参见 PC80 的编程指导,所不同之处作如下补充:

## 一、简介:

PC40 是在 DK—PC80 基础上研制开发的新产品,它不但可实现对开关量的逻辑运算和控制,并具有计数、定时、移位、逻辑与或和步控功能,而且在结构上作了较大的改进,性价比得到进一步提高,该系统主要由电源装置、母板、CPU 处理器功能模块、输入/输出功能模块、PRG—001 编程器等五部分组成。输入/输出点数从 16 点到 72 点任意选配。

## 二、外形尺寸及安装

图 4—1 示出了 PC40 的机箱安装尺寸图,机箱自左到右为电源、CPU 模块、输入输出模块,其中电源、CPU 模块为固定位置,输入/输出模块可根据用户自定。

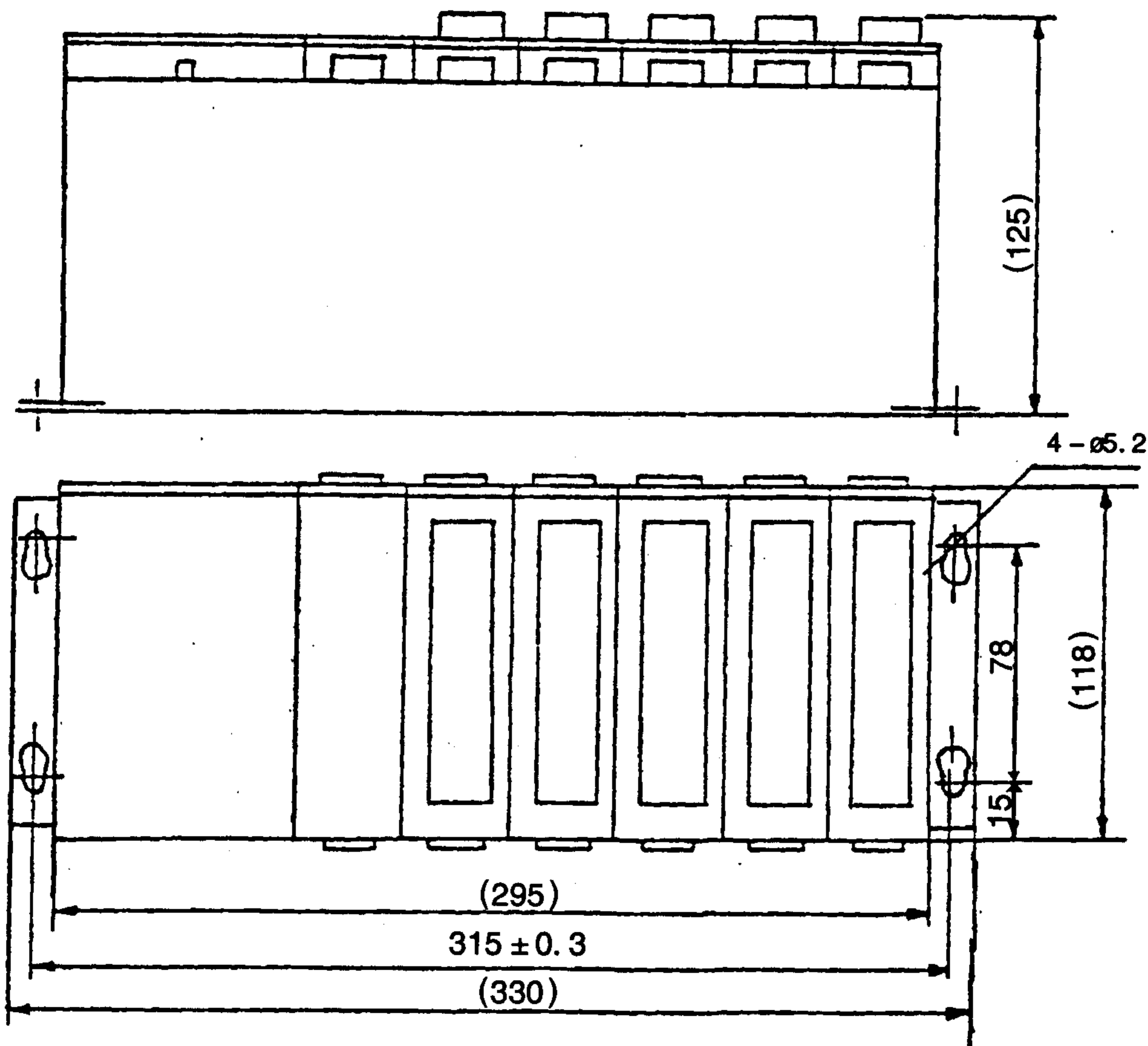


图 4—1 PC40 安装尺寸图

### 三、CPU 用户程序位置示意图

CPU 模块是 PC40 的控制核心，它通过 56 线专用插头座和系统母板相连，16 线带状电缆和 PRG—001 相连，该模块上备有 EPROM2764A 写入插座 (2D5)，使用户程序的读写变得极为方便，用户程序 (EPROM2764A) 单元在 CPU 模块的相对位置见图 4—2 所示。

(用户也可特殊订货, 选用 EEPROM2864A 芯片。)

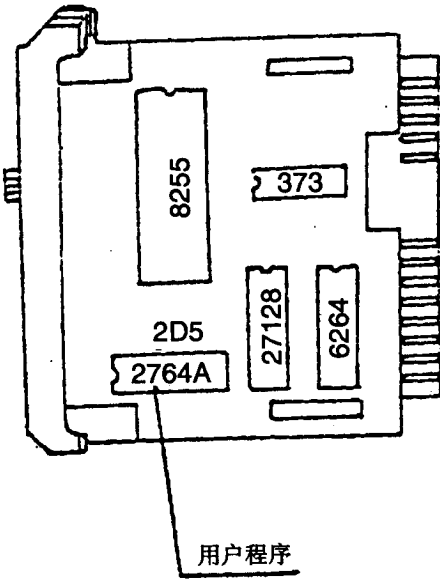


图 4—2 CPU 模块及用户程序位置示意图

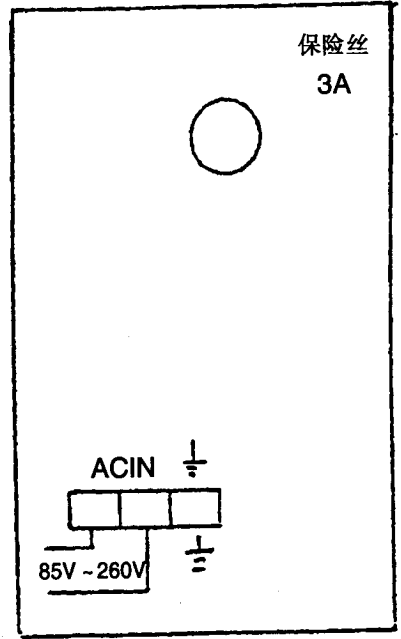


图 4—3 电源装置接线示意图

### 四、电源装置接线参见示意图 4—3

### 五、输入/输出功能模块系列

PC40 输入输出功能模块安装极为方便，只需将其插入机箱内母板上的 I/O 槽口中，I/O 模块插入槽口位置没有硬性规定，只是槽位与地址号有对应的关系，因此希望用户在编程中注意模块的槽位所赋予的地址号。图 4—4 给出了 PC40 插槽和卡口的对应关系

		I/O I	I/O II	I/O III	I/O IV	I/O V
电    源	CPU	COM1	COM1	COM1	COM1	COM1
		0.0 ↓ 0.7	2.0 ↓ 2.7	4.0 ↓ 4.7	6.0 ↓ 6.7	8.0 ↓ 8.7
		1.0 ↓ 1.7	3.0 ↓ 3.7	5.0 ↓ 5.7	7.0 ↓ 7.7	9.0 ↓ 9.7
		COM2	COM2	COM2	COM2	COM2

注:1:当用 BCD 拨码开关时,则 I/O I 槽供拨码开关专用,否则作输入模块使用。图中 COM1、COM2 因模块不同而含义不同,详见后页说明。

2:当采用 8 点输入或输出模块时,地址号为相应槽的前八点。

3:当输入或输出触点动作时,其所在模块而面板上相应的输入(绿色)或输出(红色)指示灯亮,反之则灭。

目前本公司可提供的 I/O 系列模块见下表:

编 号	名 称	点数
PC40—DCIN8	8 点输入	8
PC40—DCIN16	16 点输入	16
PC40—ROUT	继电器输出	8
PC—40—DCOUT	晶体管直流输出	8

1、直流 8 点输入模块(低有效):COM1(C+)为 +24V,COM2(C-)为 24V 地。

型 号		PC40—DCIN8
电 压		DC:24V(+10%、-15%)内部提供
输 入 阻 抗		2K
输入 信号	ON(通)	外部输入点闭合,工作指示灯(绿灯)亮
	OFF(断)	外部输入点断开,工作指示灯灭
隔离方式		光电隔离
电 路 图		
外 部 接 线		

2. 直流 16 点输入模块(低有效):COM1(C+)为 +24V,COM2(C-)为 24V 地。

型 号	PC40—DCIN16	
电 压	DC:24V(+10%、-15%)内部提供	
输 入 阻 抗	2K	
输入 信号	ON(通)	外部输入点闭合,工作指示灯(绿灯)亮
	OFF(断)	外部输入点断开,工作指示灯灭
隔离方式	光电隔离	
电 路 图		
外 部 接 线		

3. 晶体管 8 点直流输出模块: 8 个输出信号为一组 COM1(C+)接 +24V, COM2(C-)接 24V 地。

型 号		PC40-DCOUT
电 压		DC: 12V ~ 24V 外部提供
电 流		2A
母线最大电流		8A
输出 信号	ON(通)	输出接通工作指示灯(红灯)亮
	OFF(断)	输出断开,工作指示灯灭
隔离方式		光电隔离
电 路 图		
外 部 接 线		

3. 继电器 8 点输出模块:4 个输出信号为一组,1~3 和 4~7 共二组,COM1(C1)和 COM2(C2)分别为二组输出信号的公共母线。

型 号		PC40—ROUT
电 压		DC: 12V ~ 24V    AC: 12V ~ 220V    外部提供
电 流		2A
母线最大电流		8A
输出 信号	ON(通)	输出接通工作指示灯(红灯)亮
	OFF(断)	输出断开,工作指示灯灭
隔离方式		电磁继电器隔离
电 路 图		
外 部 接 线		



# 第五篇 超小型可编程序控制器 使用手册

MF—PC24 可编程序控制器与 DK—PC80 可编程序控制器的使用方法基本相同，系统功能与指令也基本相同，编程器与 DK—PC80 通用，用户的安装及编程过程可参见 DK—PC80 可编程序控制器使用手册。本手册仅就部分不同之处作如下补充：

## 1、简介：

MF—PC24 是在 DK—PC80 基础上研制开发的一种超小型新产品，它不但具有开关量的逻辑运算和控制，还具有计数、定时、移位和步控功能，而且在结构上作了较大的改进，使外形尺寸及外观更趋精致，安装使用更为灵活，特别是性能价格比得到进一步提高。

该系统主要由电源、CPU、I/O 三部分组成，电源输入电压可在 AC85V ~ 265V 之间任意变化，用户程序可随时随地进行修改写入、无需固化器。输入为 16 路低电平有效、无需外接电源，并可向外提供 DC24V 左右 300mA 的电源以供其它输入元件使用，输出为 8 路，每路均为继电器，触点容量为 AC5V ~ 220V/1A、DC5V ~ 24V/2A。

2、外形尺寸及安装：参见图 1。

3、地址分布及接口形式：

图 2 为 PC24 对外连接部分。

由图 2 可知，其对外连接由上下、左右四部分组成，现分别叙述：

3.1 上部为输入端子，地址从 2.0 ~ 3.3，共 12 路，COM - 为公共端，所有输入均为直流 24V 低电平有效，亦即任何一路与 COM - 连通，即为该路有输入信号，每路均有 LED 指示。输入部分接口形式见图 3。

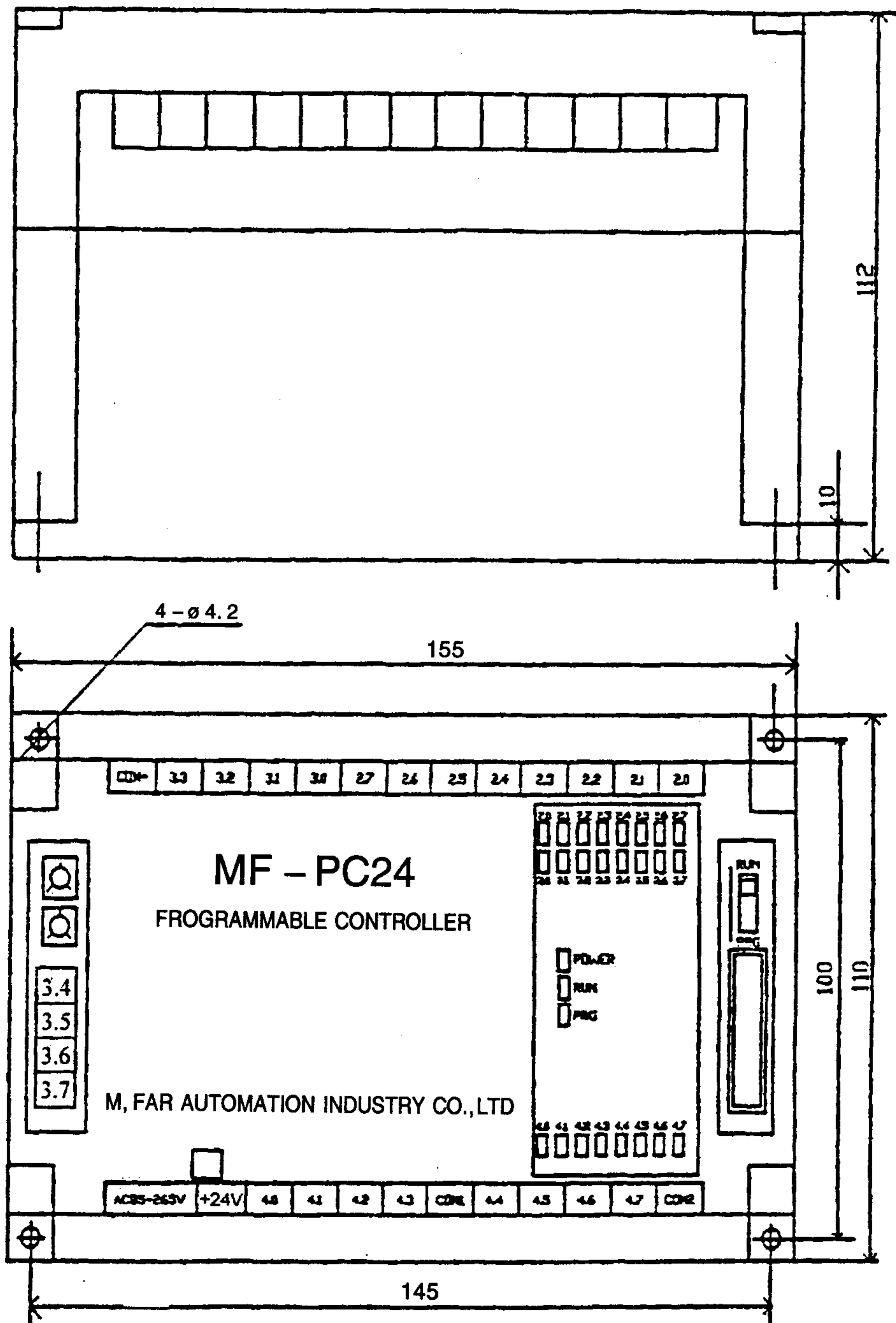


图 1 外形尺寸

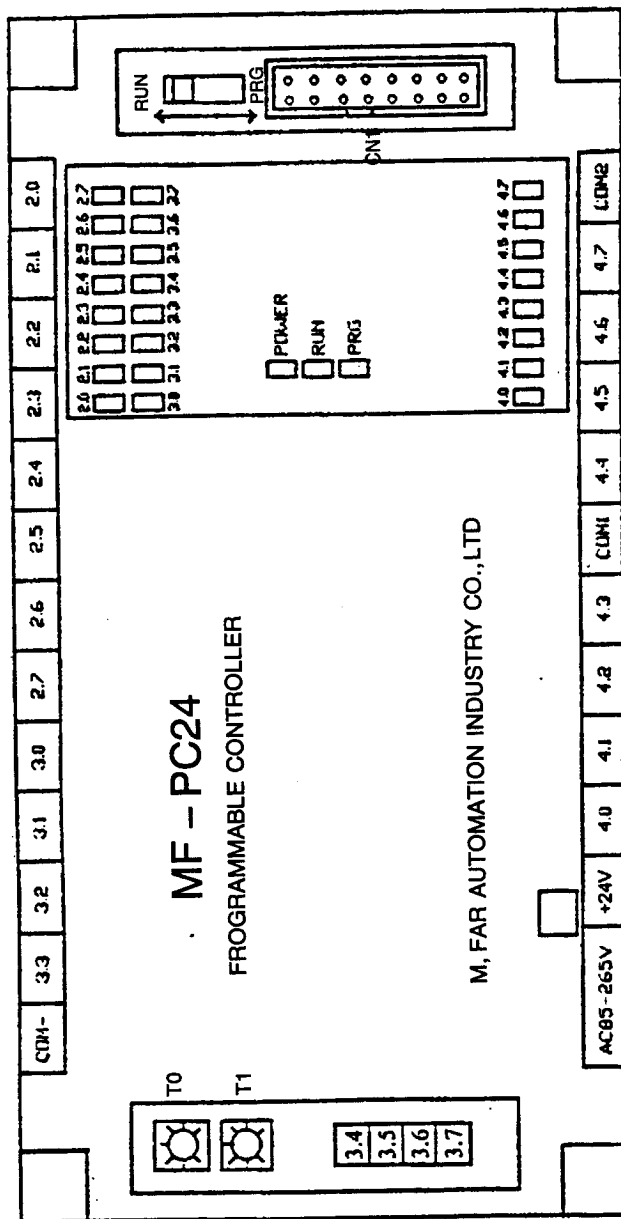


图 2 外部接口

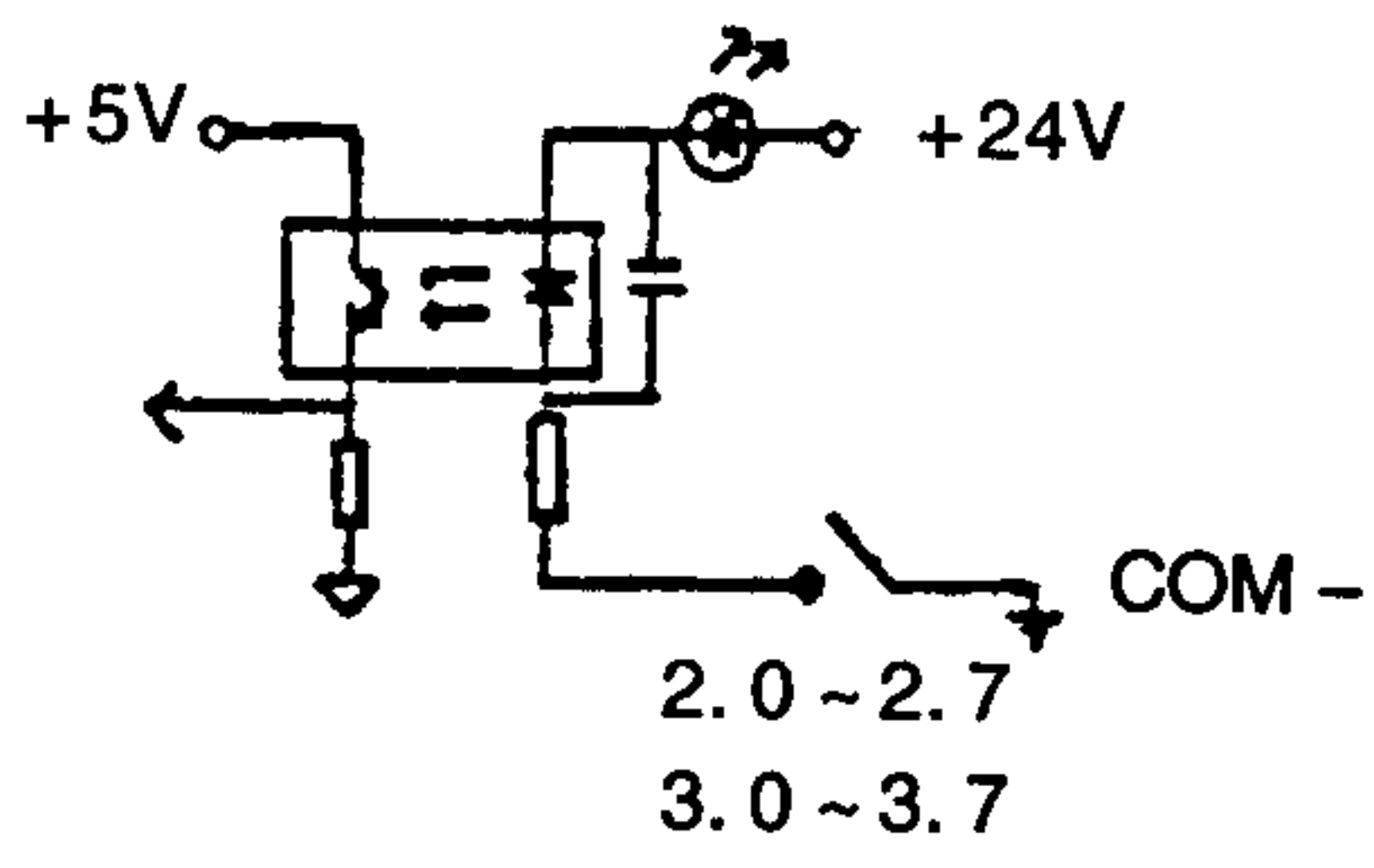


图3 INPORT(输入接口)

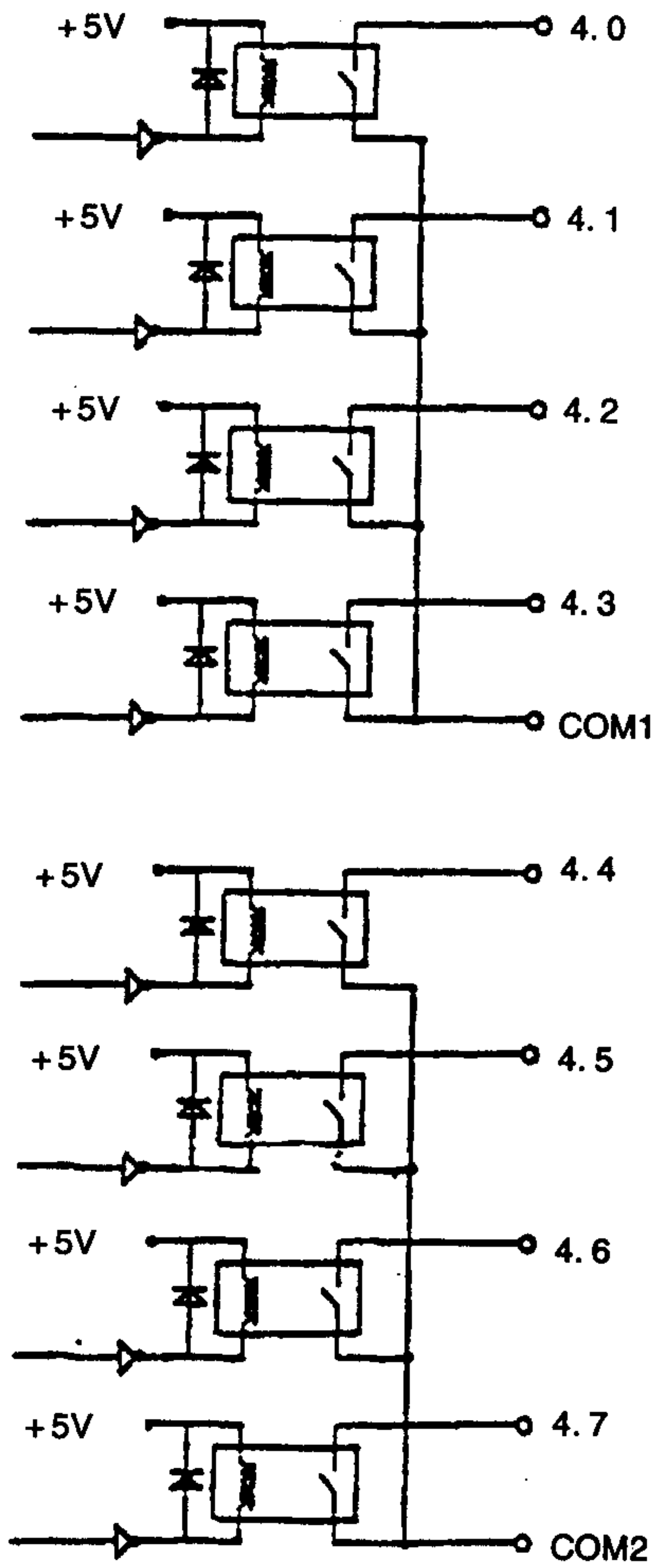


图4 OUTPORT(输出接口)

3.2 下部为输出部分,地址从 4.0~4.7。

输出全部采用继电器形式,输出容量为 AC5V~200V,电流 1A;DC5V~DC24V,电流 2A;阻性负载。

输出共分两组:4.0~4.3 为 1 组 公共点为 COM1

4.4~4.7 为 1 组 公共点为 COM2

两组可以采用不同电源,使用方便。

输出接口图见图 4

3.3 左部共有两部分组成,上部为两只旋转开关 T₀、T₁,它是专门用作定时或计数的设定单元。(注:该部分特殊订货时提供)

其地址代号分别为 CNT0/TIMO、CNT1/TIMI

它们根据所选系数可通过外部设备使计数值范围 0~15,延时值范围为 0.15~150 秒。编程方法参见 P28、P29 及 P47 页。

使用时,先由软件选定某一功能,并编好程序。在开机前,首先旋转开关,用小起子旋至所需位置(自左向右顺时针旋转可设定 O~F,共 16 档),然后接通电源。

注意:PLC 在每次通电瞬间将设置值读入,通电期间变更设置无效。

3.4 左部的下部分接线端子,它是四路输入,其引脚排列自上而下见下图。

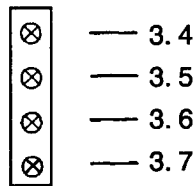


图 5 内藏式四路输入端

每路输入形式同 3.1 所述,且每路均有 LED 指示,参见图 3。

3.5 右部的上方有一拨动开关 S1,它是 PLC 的状态开关,将其拨在上面“RUN”位置,是运行状态,同时 PLC 中央的“RUN”LED 绿色指示灯发光,表示此时 PLC 正在工作,将开关拨在下面“PRG”位置,是编程状态,此时,PLC 中央的“PRG”LED 红色指示灯发光,表示此时 PLC 停止所有输出,进入编程状态。

在正常工作时,应将 PLC“S1”开关拨在上部“RUN”位置。

3.6 右部的下方是编程的插孔 CN1 它适合采用通用的 PRG001 编程器,在编程或监控时,必须将编程器电缆接头按照其缺口方向插入插座 CN1 中,其编程方法请参见 DK—PC80 使用手册的第 2 篇:编程指导。

#### 4、程序的写入及固化

MF—PC24 的程序采用 EEPROM 贮存,新写入的程序临时存入在 RAM 中,待联机调试完毕后,应立即固化到 EEPROM 中,以免掉电后丢失,带来不必要麻烦。

##### 4.1 程序写入:

与 DK—PC80 方法完全一样。

#### 4.2 程序固化:

- 1)将“S1”开关拨向下方,连接好编程器;
- 2)按一下编程器上的“SHF”,“SHF”灯亮;
- 3)按一下“EPROM”键,显示“EP”字样;
- 4)再按一下“EPROM”键,在编程器显示窗左侧闪烁“日”;
- 5)闪烁“日”完毕,将“S1”开关拨向上方即完毕。

在调试中,为了减少麻烦,机器断电前,应将修改的程序执行一下“4.2”的操作,一旦调试完毕,固化好后,将永久保存在 MF—PC24 中,不会丢失。

# MF-PC24 T

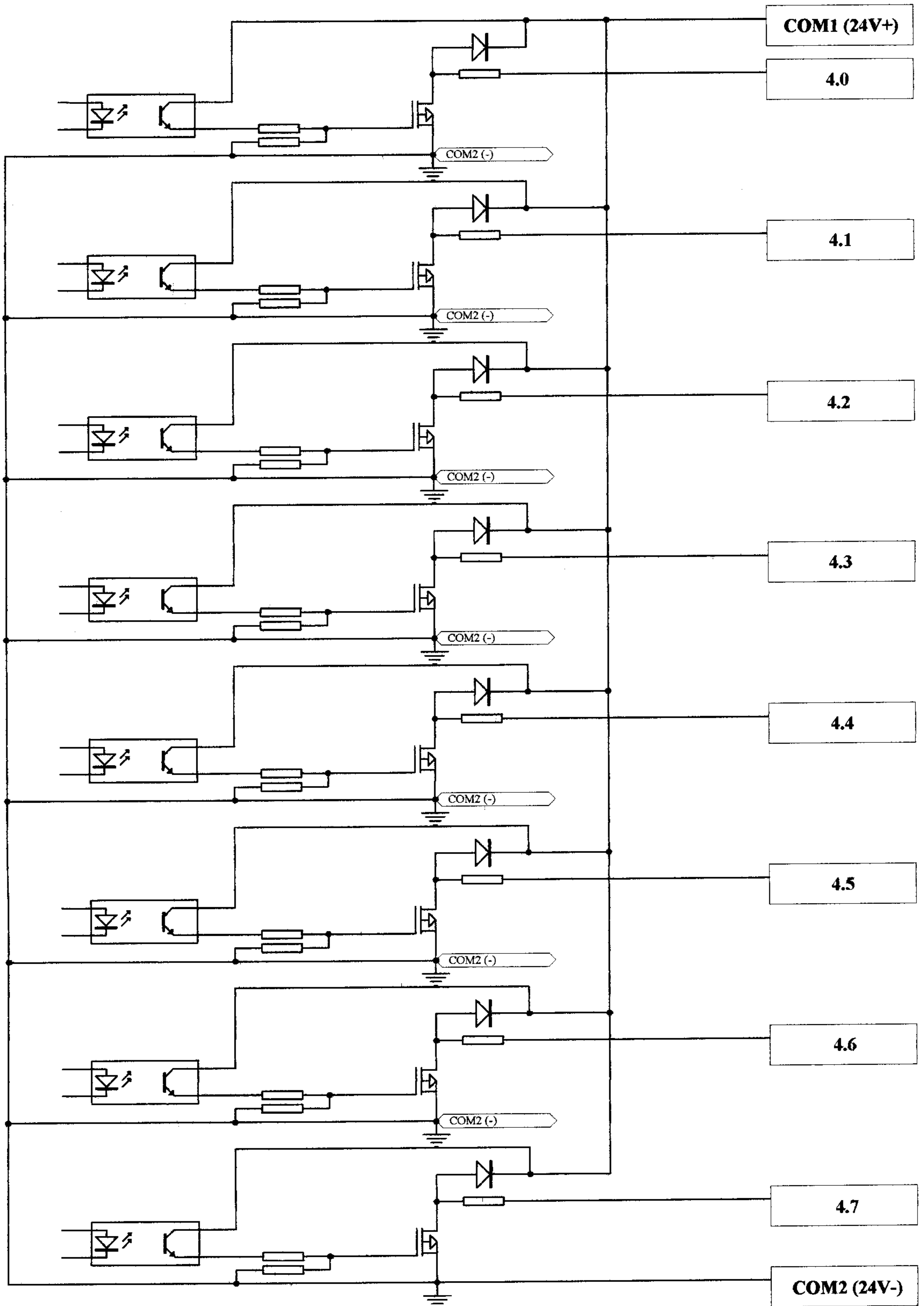


图6 晶体管输出接口图

## 附表 1 程序的检验及错误类别

在程序的输入过程及运行前,为保证最小限度产生不应有的错误,PC 控制器具有自动进行程序错误检验的能力,当输入的指令语句发生了语法错误等情况,将不接受 **WRITE** 键的操作,这时在“DATA”部分显示出错误信息,提示你修正输入的错误。下面给出编程上显示的错误信息及处理方法。

Err 1:输出继电器线圈重复使用。整个程序里,不允许相同卡号及位地址的线圈被重复使用。

Err 2:操作数错误,不同的指令码后要求的操作数类形不同,当发生错误时,将给出 Err 2 显示,另外操作数超出范围也导致发生此类错误。

Err 3:表示语句输入没有指令码,只有操作数。

Err 4:表示语句步号超过 1000,超出本控制器提供给用户语句步号。

上述四种错误类型的处理方法:

1、按动 **CE**,消去 ErrX 显示。

2、输入正确的键命令即可。

PC80 对用户程序的处理是将程序编译后运行,在控制器状态开关由“PRG”一侧打向“RUN”侧后,进行程序编译,这时程序结构(即相当于控制电器结构)产生错误时将给出错误信息。

Err 5:LD 指令过多。

a、主控母线开始前或输出指令之前有多余 LD 指令。

b、LD 指令多于(A)、(O)指令。

Err 6:LD 指令过少,LD 指令少于(A)、(O)指令。

Err 7:MCR 指令个数超过 MCS 指令或 MCR 指令前没有输出指令的语句行。

Err 8:MCR 指令个数少于 MCS 指令。

Err 9:指令码发生错误。

Err a: VAL 指令后没有 TIM 或 CNT 指令语句;或 TIM 或 CNT 指令前没有 VAL 指令。

上述 6 种错误发生后,控制器将不运行用户程序。处理时,首先将状态开关拨回到“PRG”一侧,然后按 **CE** 键消去错误显示,仔细查找程序结构上的错误。

Err d:EPROM 写入有错。将 RAM 区用户程序写入 EPROM 后,EPROM 里的程序和 RAM 里的程序不符。

处理方法:按动 **CE** 键,消去 Errd,编程器上显示第 0 步的内容。

1、EPROM 写入操作不当。

2、EPROM 片子本身不干净,或不是 2764A,写入电压不是 12.5V。

3、CPU 板上 EPROM 写入部分存在故障。



**无锡中远工业自动化有限公司  
无锡中远自动化研究所**

地址：无锡滨湖经济技术开发区泰运路

电话：0510—5628100 8821090

传真：0510—5629169

邮编：214131

http://www.m-far.com

E-mail: m.far@public1.wx.js.cn