

第一章 P L C 概述

一、可编程控制器的产生及定义

- ① 1969 年美国数字设备公司 (DEC) 研制出世界第一台可编程控制器，并成功地应用在美国通用汽车公司 (GM) 的生产线上。但当时只能进行逻辑运算，故称为可编程逻辑控制器，简称 PLC (programmable logic controller)。
- ② 70 年代后期，随着微电子技术和计算机技术的迅猛发展，使 PLC 从开量的逻辑控制扩展到数字控制及生产过程控制域，真正成为一种电子计算机工业控制装置，故称为可编程控制器，简称 PC (programmable controller)。但由于 PC 容易与个人计算机 (personal computer) 相混淆，故人们仍习惯地用 PLC 作为可编程控制器的缩写。
- ③ 1985 年国际电工委员会 (IEC) 对 PLC 的定义如下：
可编程控制器是一种进行数字运算的电子系统，是专为在工业环境下的应用而设计的工业控制器，它采用了可以编程序的存储器，用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令，并通过数字或模拟式的输入和输出，控制各种类型机械的生产过程。
- ④ PLC 是由继电器逻辑控制系统发展而来，所以它在数学处理、顺序控制方面具有一定优势。继电器在控制系统中主要起两种作用：(1) 逻辑运算 (2) 弱电控制强电。
- ⑤ PLC 是集自动控制技术、计算机技术和通讯技术于一体的一种新型工业控制装置，已跃居工业自动化三大支柱 (PLC、ROBOT、CAD/CAM) 的首位。

二、可编程控制器的分类及特点

(一) 分类

(1) 从组成结构形式分

- ① 一体化整体式 P L C
- ② 模块式结构化 P L C

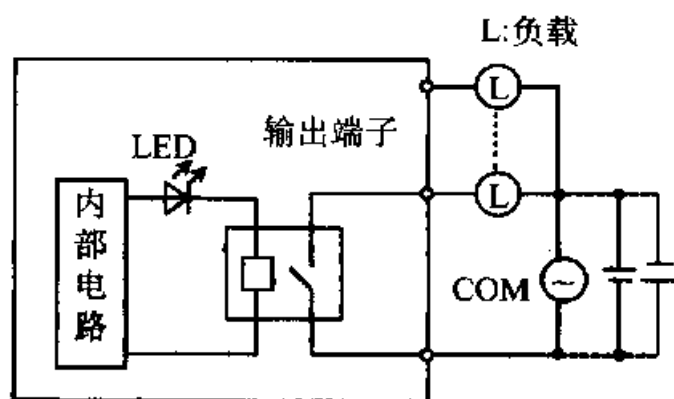
(2) 按 I / O 点数及内存容量分

- ① 超小型 P L C
- ② 小型 P L C
- ③ 中型 P L C
- ④ 大型 P L C
- ⑤ 超大型 P L C

(3) 按输出形式分

- ① 继电器输出

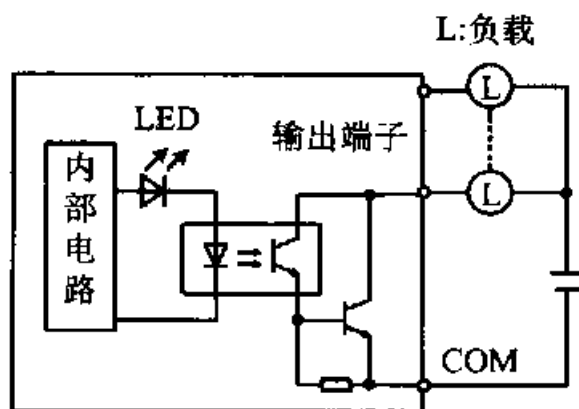
为有触点输出方式，适用于低频大功率直流或交流负载



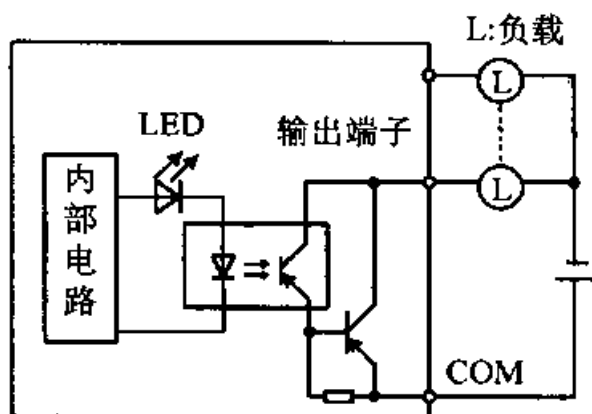
继电器输出型

②晶体管输出

为无触点输出方式，适用于高频小功率直流负载



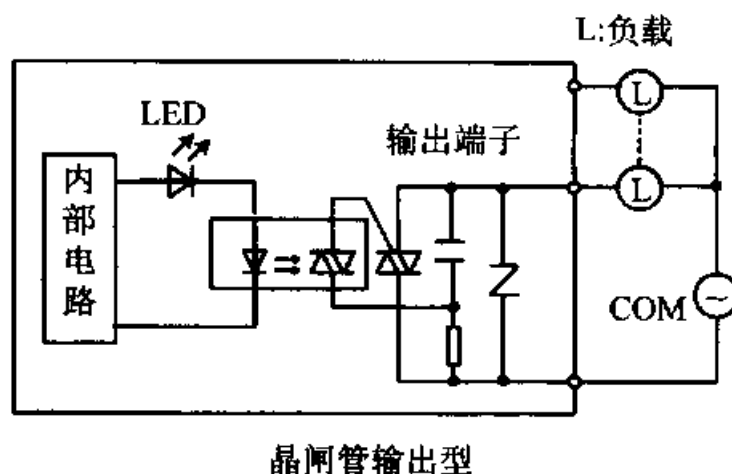
晶体管输出型（NPN 集电极开路）



晶体管输出型（PNP 集电极开路）

③晶闸管输出

为无触点输出方式，适用于高速大功率交流负载



(二) 特点

- ①可靠性高、抗干扰能力强
- ②编程简单、使用方便
- ③设计、安装容易，维护工作量少
- ④功能完善、通用性好，可实现三电一体化

PLC 将电控（逻辑控制）、电仪（过程控制）和电结（运动控制）这三电集于一体。

- ⑤体积小、能耗低
- ⑥性能价格比高

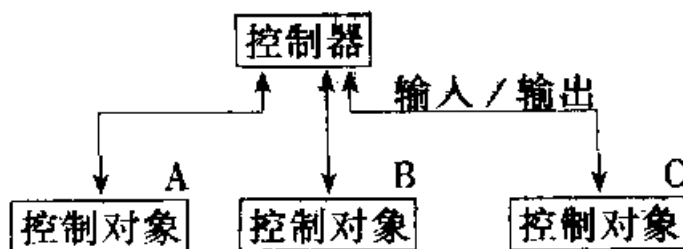
三、可编程控制器的应用

- ①开关量的逻辑控制
- ②位置控制
- ③过程控制
- ④数据处理
- ⑤通信联网
- ⑥CIMS 的应用

四、PLC 控制系统的分类

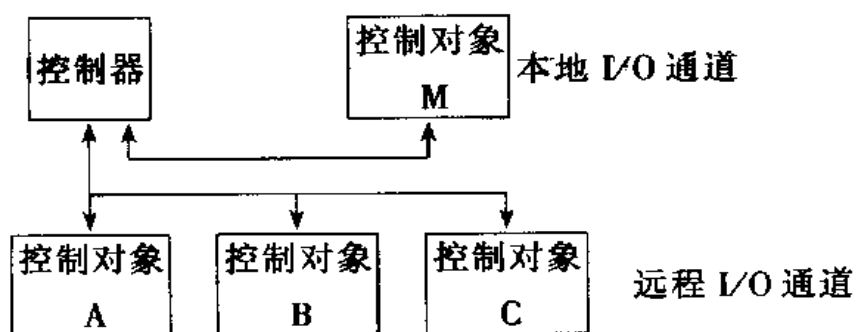
(一)、集中式控制系统

集中式控制系统是用一个 PLC 控制一台或多个被控设备。主要用于输入、输出点数较少，各被控设备所处的位置比较近，且相互间的动作有一定联系的场合。其特点是控制结构简单。



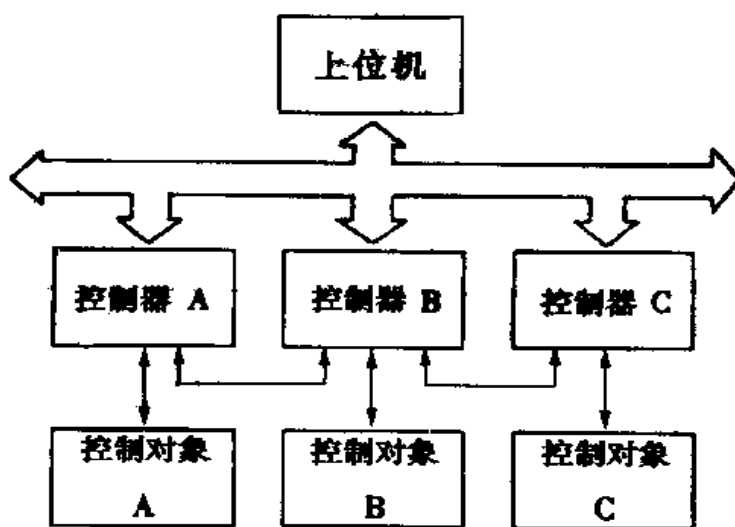
(二)、远程式控制系统

远程式控制系统是指控制单元远离控制现场，PLC 通过通信电缆与被控设备进行信息传递。该系统一般用于被控设备十分分散，或工作环境比较恶劣的场合。其特点是需要采用远程通信模块，提高了系统的成本和复杂性。



(三) 分布式控制系统

分布式控制系统即采用几台小型 PLC 分别独立控制某些被控设备，然后再用通信线将几台 PLC 连接起来，并用上位机进行管理。该系统多用于有多台被控设备的大型控制系统，其各被控设备之间有数据信息传送的场合。其特点是系统灵活性强、控制范围大，但需要增加用于通信的硬件和软件，系统的复杂性也更大。

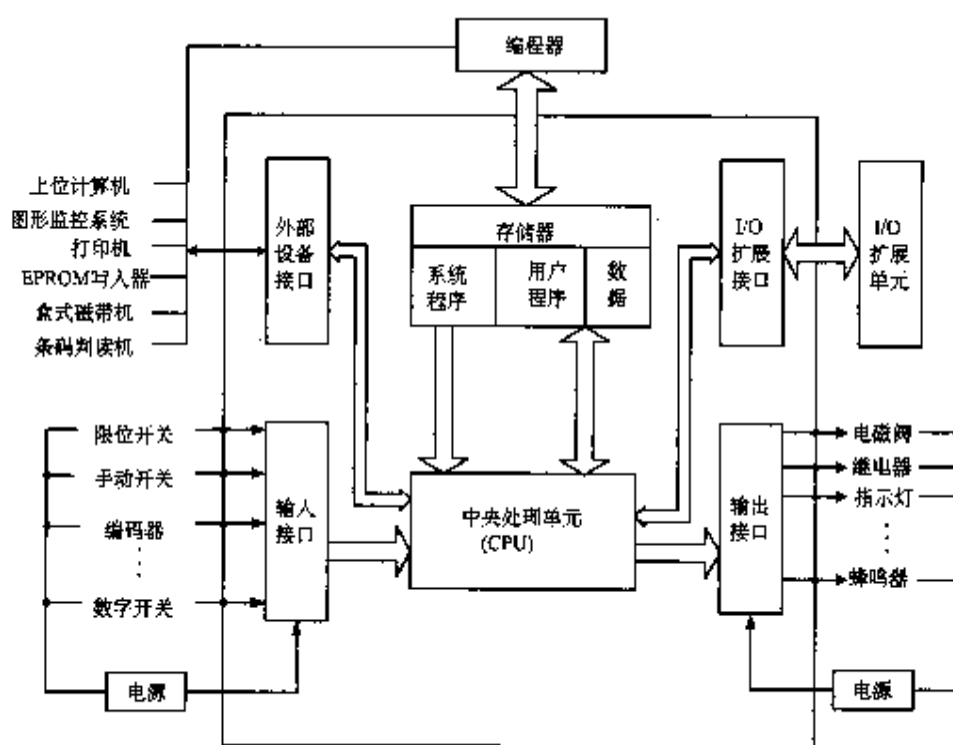


2.1 PLC 的组成与基本结构

2.1.1 P L C 的基本组成

P L C 主要由中央处理单元、输入接口、输出接口、通信接口等部分组成，其中 C P U 是 P L C 的核心，I / O 部件是连接现场设备与 C P U 之间的接口电路，通信接口用于与编程器和上位机连接。

对于整体式 P L C，所有部件都装在同一机壳内；对于模块式 P L C，各功能部件独立封装，称为模块或模板，各模块通过总线连接，安装在机架或导轨上。不同厂商生产的不同系列产品在每个机架上可插放的模块数是不同的，一般为 3 — 1 0 块。可扩展的机架数也不同，一般为 2 — 8 个机架。基本机架与扩展机架之间的距离不宜太长，一般不超过 1 0 M。



PLC 硬件结构图

2.1.2 P L C 各组成部分

1、中央处理单元 C P U

C P U 通过输入装置读入外设的状态，由用户程序去处理，并根据处理结果通过输出装置去控制外设。

一般的中型可编程控制器多为双微处理器系统，一个是字处理器，它是主处理器，由它处理字节操作指令，控制系统总线，内部计数器，内部定时器，监视扫描时间，统一管理编程接口，同时协调位处理器及输入输出。另一个为位处理器，也称布尔处理器，它是从处理器，它的主要作用是处理位操作指令和在机器操作系统的管理下实现 P L C 编程语言向机器语言转换。

C P U 处理速度是指 P L C 执行 1 0 0 0 条基本指令所花费的时间。

2、存储器

存储器主要存放系统程序，用户程序及工作数据。

P L C所用的存储器基本上由P R O M，E P R O M，E E P R O M及R A M等组成。

3、输入 / 输出部件

输入 / 输出部件又称 I / O 模块。P L C通过 I / O 接口可以检测被控对象或被控生产过程的各种参数，以这些现场数据作为P L C对控对象进行控制的信息依据。同时P L C又通过 I / O 接口将处理结果送给被控设备或工业生产过程，以实现控制。

4、编程装置和编程软件

P L C是以顺序执行存储器中的程序来完成其控制功能的。

5、电源部件

2.2 P L C的基本工作原理

2.2.1 P L C的循环扫描工作过程

（一）P L C的循环扫描

P L C的C P U是采用分时操作的原理，每一时刻执行一个操作，随着时间的延伸一个动作接一个动作顺序地进行，这种分时操作进程称为C P U对程序的扫描。P L C的用户程序由若干条指令组成，指令在存储器中按序号顺序排列。C P U从第一条指令开始，顺序逐条地执行用户程序，直到用户程序结束，然后返回第一条指令开始新一轮扫描。

（二）P L C工作过程

1、公共操作

公共操作是在每次扫描程序前进行的自检。

2、数据 I / O 操作

数据 I / O 操作也称为 I / O 状态刷新。它包括两种操作：

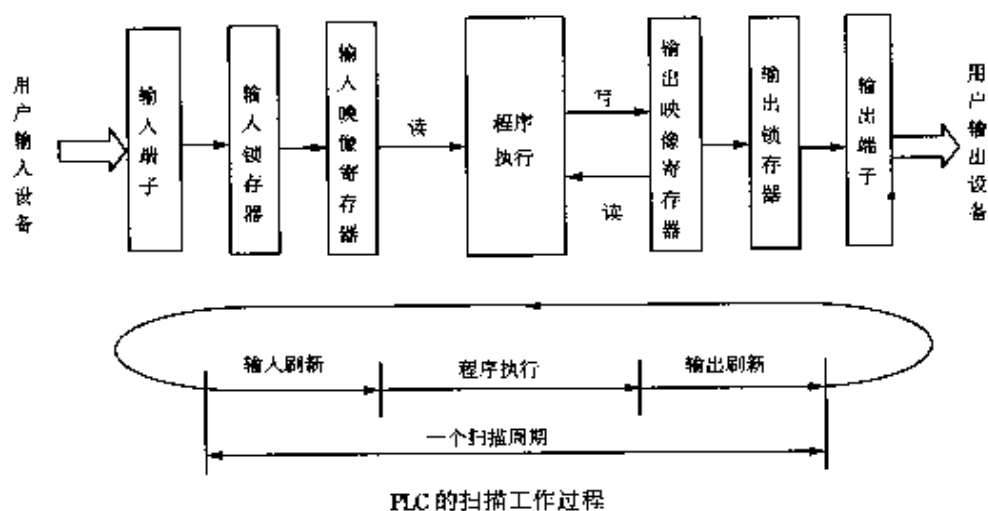
①采样输入信号，即刷新输入状态表的内容

②送出处理结果，即用输出状态表的内容刷新输出电路

3、执行用户程序操作

4、处理外设请求操作

外设的请求命令包括操作人员的介入和硬件设备的中断



2.2.2 PLC的 I / O 滞后现象

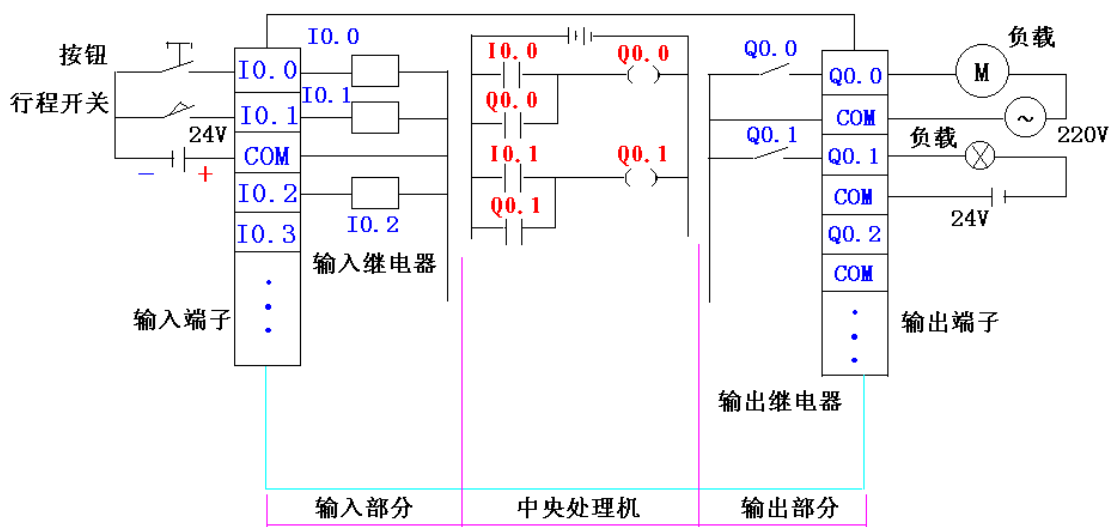
造成 I / O 响应滞后的原因：

①扫描方式

②电路惯性

输入滤波时间常数和输出继电器触点的机械滞后

③与程序设计安排有关



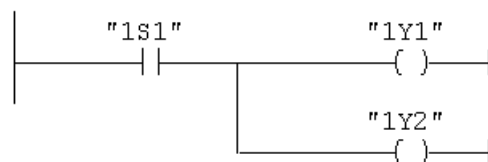
PLC的等效电路示意图

2.3 PLC 的编程语言

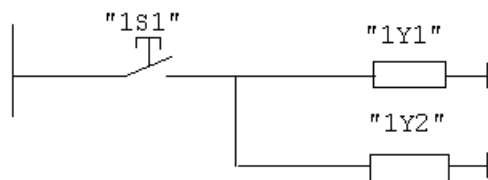
2.3.1 梯形图编程

(一) PLC 的编程特点

1、程序的执行顺序

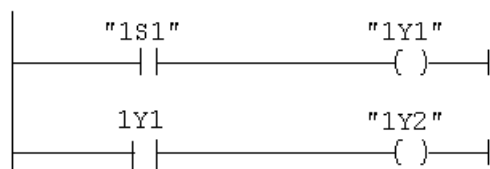


PLC梯形图

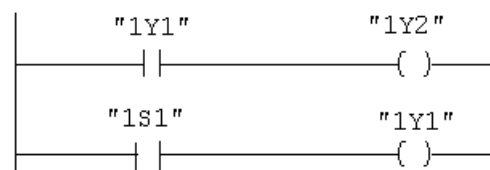


继电器梯形图

两图实现相同的功能。当 1S1 闭合时，1Y1、1Y2 输出。系统上电之后，当 1S1 闭合时，继电器梯形图中的 1Y1、1Y2 会同时得电，若不考虑继电器触点的延时，则 1Y1、1Y2 会同时输出。但在 PLC 梯形图中，因为 PLC 的程序是顺序扫描执行的，PLC 的指令按从上向下，从左向右的扫描顺序执行，整个 PLC 的程序不断循环往复。PLC 的“继电器”的动作顺序由 PLC 的扫描顺序和在梯形图中的位置决定，因此，当 1S1 闭合时，1Y1 先输出而 1Y2 后输出。即继电器采用并行的执行方式，而 PLC 则采用串行的执行方式。

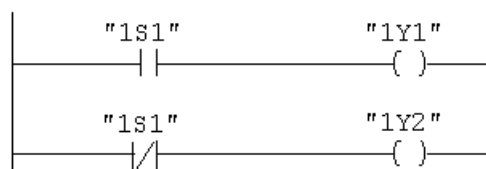


1Y1、1Y2在同一个周期内动作

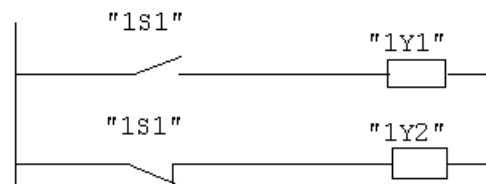


1Y1、1Y2在两个周期内动作

2、继电器自身的延时效应



1Y1、1Y2同时动作



1Y1、1Y2不同时动作

传统的继电器的触点在线圈得电后动作时有一个微小的延时，并且常开和常闭触点的动作之间有一微小的时间差。而 PLC 中的继电器都为软继电器，不会有延时效应，当然，这里忽略了 PLC 的扫描时间。

3、PLC 中的软继电器

每个继电器有无数个常开和常闭触点。

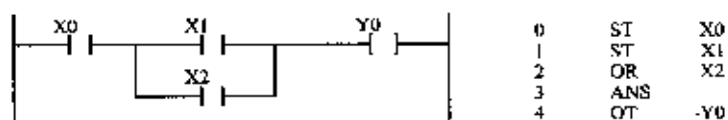
(二) PLC 编程的基本原则

- (1) 每个梯形图网络由多个梯级组成，每个输出元素可构成一个梯级，每个梯级可由多个支路组成。
- (2) 梯形图每一行都是从左母线开始，而且输出线圈接在最右边，输入触点不能放在输出线圈的右边。
- (3) 输出线圈不能直接与左母线连接。
- (4) 多个的输出线圈可以并联输出。
- (5) 在一个程序中各输出处同一编号的输出线圈若使用两次称为“双线圈输出”。双线圈输出容易引起误动作，禁止使用。
- (6) PLC 梯形图中，外部输入 / 输出继电器、内部继电器、定时器、计数器等器件的触点可多次重复使用。
- (7) 梯形图中串联或并联的触点的个数没有限制，可无限次的使用。
- (8) 在用梯形图编程时，只有在一个梯级编制完整后才能继续后面的程序编制。
- (9) 梯形图程序运行时其执行顺序是按从左到右，从上到下的原则。

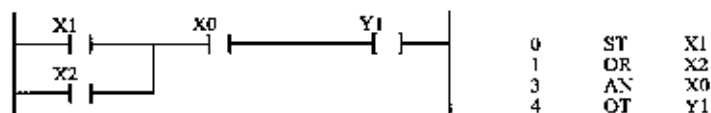
(二) 编程技巧及原则“上重下轻，左重右轻，避免混联”

- (1) 梯形图应把串联触点较多的电路放在梯形图上方
- (2) 梯形图应把并联触点较多的电路放在梯形图最左边
- (3) 为了输入程序方便操作，可以把一些梯形图的形式作适当变换

图 4-9 是一段“左大右小”的梯形图,图(a)和图(b)仅相差一步,实际上执行步数差了 25%。



(a)不符合左大右小的电路,共 5 步



(b)符合左大右小的电路,共 4 步

图 4-9

2.3.2 语句表编程

P L C 的语句：操作码+操作数

操作码用来指定要执行的功能，告诉 C P U 该进行什么操作；操作数内包含为执行该操作所必需的信息，告诉 C P U 用什么地方的数据来执行此操作。

操作数的分配原则：

- (1) 为了让 C P U 区别不同的编程元素，每个独立的元素应指定一个互不重复的地址
- (2) 所指定的地址必须在该型机器允许的范围之内。

2.3.3 其它编程语言

功能图编程．高级编程语言（C 语言．Pascal 语言等）

编程语言	用户类	应用
语句表（STL）	愿意用类似于机器码语言编程的用户	程序在运行时间和存贮空间要求上最优
梯形图（LAD）	习惯电路图的用户	编写逻辑控制程序
功能图（FBD）	熟悉布尔代数逻辑图的用户	编写逻辑控制程序
SCL（结构控制语言）可选软件包	用高级语言。如 PASCAL 或 C 语言编程的用户	数据处理任务程序
S7 Graph（顺序控制）可选软件包	有技术背景，没有 PLC 编程经验的用户	以顺序过程的描述很方便
S7 HiGraph（状态图形）可选软件包	有技术背景，没有 PLC 编程经验的用户	以异步非顺序过程的描述很方便

CFC（连续功能图）可选软件包	有技术背景，没有 PLC 编程经验的用户	适用于连续过程的描述
-----------------	----------------------	------------

第三章 SIMATIC S7-300 PLC 系统特性及硬件构成

3.1 S7-300 PLC 系统结构

3.1.1 S7-300 PLC 的组成

主要组成部分有导轨（RACK）、电源模块（PS）、中央处理单元 CPU 模块、接口模块（IM）、信号模块（SM）、功能模块（FM）等，通过 MPI 网的接口直接与编程器 PG、操作员面板 OP 和其它 S7 PLC 相连。

3.1.2 S7-300 的扩展能力

CPU314 一个机架上最多只能再安装八个信号模块或功能模块，最多可以扩展为四个机架。中央处理单元总是在 0 机架的 2 号槽位上，1 号槽安装电源模块，3 号槽总是安装接口模块，槽号 4 至 11，可自由分配信号模块、功能块。

3.1.3 S7-300 模块地址的确定

数字 I / O 模块每个槽划分为 4 Byte（等于 32 个 I / O 点），模拟 I / O 模块每个槽划分为 16 Byte（等于 8 个模拟量通道），每个模拟量输入或输出通道的地址总是一个字地址。

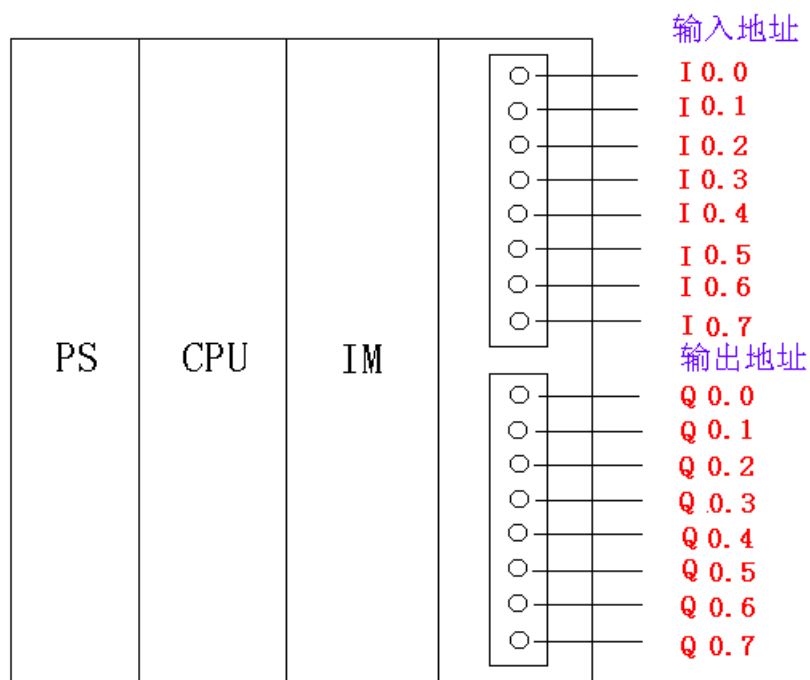
机架	模板起始地址	槽号										
		1	2	3	4	5	6	7	8	9	10	11
0	数字量	PS	CPU	IM	0	4	8	12	16	20	24	28
	模拟量				256	272	288	304	320	336	352	368
1	数字量			IM	32	36	40	44	48	52	56	60
	模拟量				384	400	416	432	448	464	480	496
2	数字量			IM	64	68	72	76	80	84	88	92
	模拟量				512	528	544	560	576	592	608	624
3	数字量			IM	96	100	104	108	112	116	120	124
	模拟量				640	656	672	688	704	720	736	752

怎样确定信号模板的地址

（一）确定数字量模板的地址

一个数字量模板的输入或输出地址由字节地址和位地址组成。字节地址取决于其模板起始地址。

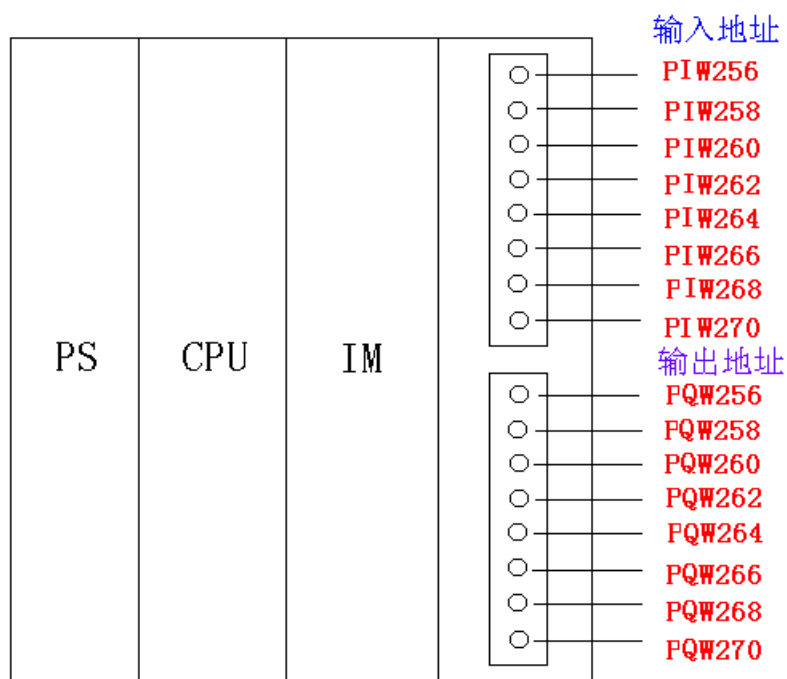
例如：如果一块数字量模板插在第 4 槽里，其地址分配如下：



(二) 确定模拟量模板的地址

模拟量输入或输出通道的地址总是一个字地址。通道地址取决于模板的起始地址。

例如：如果第一块模拟量模板插在 4 号槽，其地址分配如下：



3.2 S7-300 PLC 存储区简介

3.2.1 S7-300 编程方式简介

S7-300 PLC 的编程软件是 STEP 7。

用户程序由组织块(OB)、功能块(FB, FC)、数据块(DB)构成。

OB 是系统操作程序与用户应用程序在各种条件下的接口界面，用于控制程序的运行。OB1 是主程序循环块，在任何情况下，它都是需要的。

功能块(FB, FC)实际上是用户子程序，分为带“记忆”的功能块 FB 和不带“记忆”的功能块 FC。前者有一个数据结构与该功能块的参数表完全相同的数据块(DB)附属该功能块，并随着功能块的调用而打开，随着功能块的结束而关闭。该附属数据块(DB)叫做背景数据块，存在背景数据块中的数据在 FB 块结束时继续保持，也即被“记忆”。功能块 FC 没有背景数据块，当 FC 完成操作后数据不能保持。

数据块(DB)是用户定义的用于存放数据的存储区。

S7 CPU 还提供标准系统功能块(SFB, SFC)。

FB与FC的区别

	名称	背景数据块	定义静态变量
FB	功能块	需要	可以
FC	功能	不需要	不可以

3.2.2 S7-300 PLC 的存储区

S7-300 CPU 有三个基本存储区：

- (1) 系统存储区：RAM 类型，用于存放操作数据（I / O、位存储、定时器、计数器等）。
- (2) 装载存储区：物理上是 CPU 模块中的部分 RAM，加上内置的 EEPROM 或选用的可拆卸 EEPROM 卡，用于存放用户程序。
- (3) 工作存储区：物理上是占用 CPU 模块中的部分 RAM，其存储内容是 CPU 运行时，所执行的用户程序单元（逻辑块和功能块）的复制件。

CPU 程序所能访问的存储区为系统存储区的全部、工作存储区中的数据块 DB、暂时局部数据存储区、外设 I / O 存储区等。

程序可访问的存储区及功能

名称	存储区	存储区功能
输入(I)	输入过程映象表	扫描周期开始，操作系统读取过程输入值并录入表中，在处理过程中，程序使用这些值 每个 CPU 周期，输入存储区在输入映象

		表中所存放的输入状态值，它们是外设输入存储区头 128Byte 的映象
输出 (Q)	输出过程映象表	在扫描周期中，程序计算输出值并存放该表中，在扫描周期结束后，操作系统从表中读取输出值，并传送到过程输出口，过程输出映象表是外设输出存储区的头 128Byte 的映象
位存储区 (M)	存储位	存放程序运算的中间结果
外设输入 (PI) 外设输出 (PQ)	I/O: 外设输入 I/O: 外设输出	外设存储区允许直接访问现场设备（物理的或外部的输入和输出），外设存储区可以字节，字和双字格式访问，但不可以位方式访问
定时器 (T)	定时器	为定时器提供存储区 计时时钟访问该存储区中的计时单元，并以减法更新计时值 定时器指令可以访问该存储区和计时单元
计数器 (C)	计数器	为计数器提供存储区，计数指令访问该存储区
临时本地数据 (L)	本地数据堆栈（L 堆栈）	在 F B、F C 可 O B 运行时设定。在块变量声明表中声明的暂时变量存在该存储区中，提供空间以传送某些类型参数和存放梯形图中间结果。块结束执行时，临时本地存储区再行分配。不同的 CPU 提供不同数量的临时本地存储区
数据块 (DB)	数据块	D B 块存放程序数据信息，可被所有逻辑块公用（“共享”数据块）或（被 F B 特定占用“背景”数据块）

3.3 S7-300 PLC 中央处理单元 CPU 模块

3.3.1 CPU 模块概述

中央处理单元 C P U 的主要特性，包括存储器容量、指令执行时间、最大 I / O 点数、各类编程元件（位存储器、计数器、定时器、可调用块）数量等。

S7-300 可编程控制器 **CPU314** 的技术数据

程序存储量	24K
每 1K 语句执行时间	0.3ms

计数器	64 个 (C0~C63) 计数范围: 0~999
定时器	128 个 (T0~T127) 定时范围: 10ms~9990s
通讯接口	MPI
编程软件	STEP7
位存储器	2048 个 (MB0~MB255)
数据块	最多 127 (DB0 保留) 大小: 最大 8KB 嵌套深度: 8 层
机架	最多 4 个 每个机架的信号模块数: 最多 8 个
应用场合	对编程范围和操作处理速度有高要求的大型设备

3.3.2 CPU 模块的方式选择开关和状态指示二极管

S7-300 的 CPU 有四种工作方式, 通过可卸的专用钥匙控制:

- (1) RUN—P: 可编程运行方式。
- (2) RUN: 运行方式。
- (3) STOP: 停机方式。
- (4) MRES: CPU 清零

用钥匙开关进行程序的清除

在开始一个新的编程工作时, 我们需要将中央处理器进行清零处理。它将很容易地通过操作 CPU 上的钥匙开关来实现。为此我们必须进行以下的操作步骤:

1. 接通 PLC 工作电源, 并等待至 CPU 的自检测运行完成
2. 转动钥匙开关至 MRES 位置, 并保持这个状态, 直至 STOP 发光二极管从闪动转为常亮状态
3. 钥匙开关转至 STOP 位置并迅速转回 MRES 位置, 保持这个状态, STOP 发光二极管开始快速闪动
4. STOP 发光二极管的快速闪动, 表示 CPU 已被清零
5. 松开钥匙开关, 这时钥匙会自动返回 STOP 位置
6. 可编程控制器已被清零, 并可以传输新的控制程序

程序的下传只能是钥匙开关在 STOP 或 RUN—P 位置进行

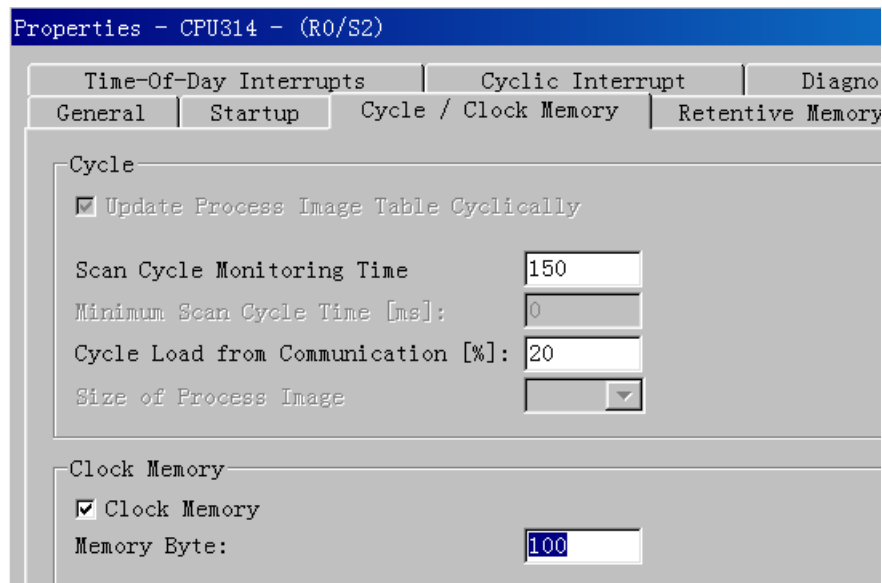
3.3.3 CPU 单元的参数设置

- (1) 时钟存储器

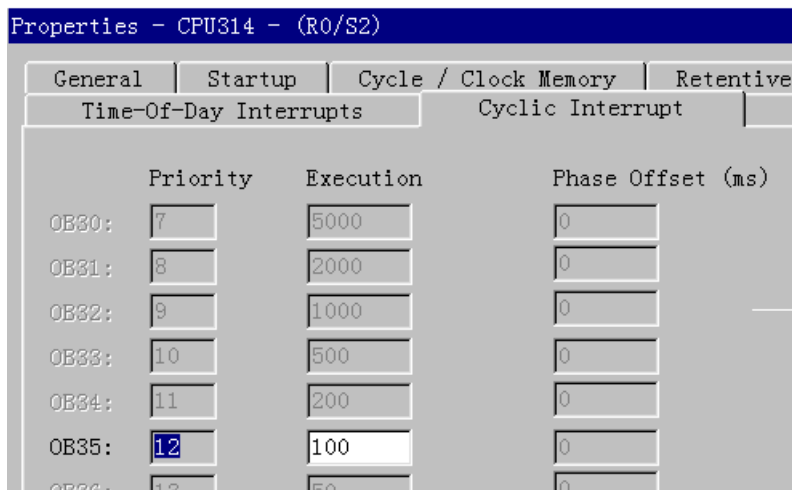
S7-300 有 8 个时钟存储器, 每个频率都不一样。可以在 0 — 2 5 5 范围内定义任一字节为时钟存储器字节。

A period duration/frequency is assigned to each bit of the clock memory byte:

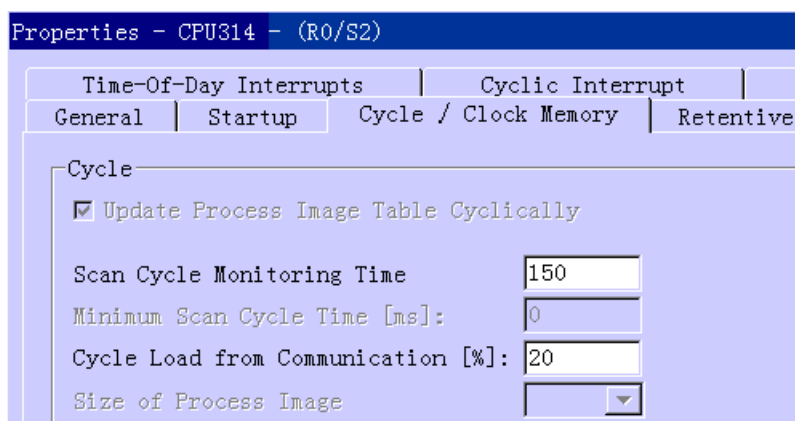
Bit	7	6	5	4	3	2	1	0
Period								
duration (s):	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Frequency								
(Hz):	0.5	0.625	1	1.25	2	2.5	5	10



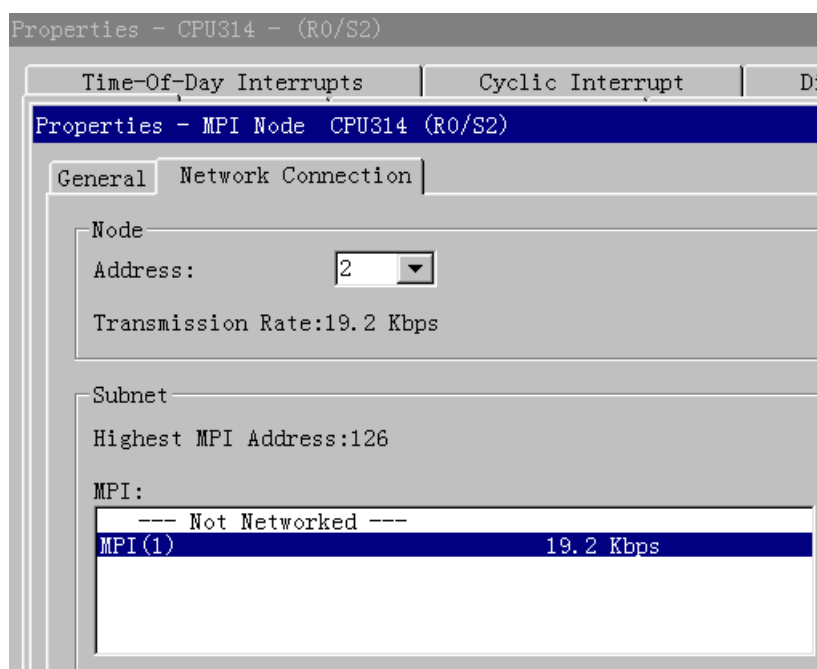
(2) 循环中断参数



(3) 最长循环时间



(4) M P I 参数



3.3.4 CPU 的循环时间计算

循环时间是一个程序循环所占用的时间，循环时间由过程映像传送时间、操作系统的执行时间和用户程序的执行时间三大部分组成

练习：

一、填空题：

- (1) 美国数字设备公司于（ ）年研制出世界第一台 P L C。
- (2) P L C 从组成结构形式上可以分为（ ）和（ ）两类。
- (3) P L C 以（ ）工作方式工作的。
- (4) 一般由程序控制的数字电子设备产生的故障常有两种，一种是（ ）；另一种是（ ）。
- (5) P L C 是由（ ）逻辑控制系统发展而来的，它在（ ）

-)、()方面具有一定优势。
- (6) P L C的基本组成主要由()、()、()、()等部分组成。
- (7) P L C的处理速度是指P L C执行()条基本指令所花费的时间。
- (8) P L C所用的存储器基本上由()、()、()等组成。
- (9) P L C运行时，内部要进行一系列操作，大致可分为四大类：()操作、()操作、()操作、()操作。
- (10) 常用的P L C编程语言有()、()、()等。
- (11) S7-300 PLC 314CPU 一个机架最多可安装()个信号模块，最多可扩展为()个机架，机架之间的通讯距离最大不超过()，最大数字量 I / O点数()，支持的可保持的定时器最多为()个，计数器最多为()个。
- (12) CPU314 支持的 OB、FB、FC、DB 的容量均不大于() KB。
- (13) 确定机架 0 的 6 号槽上 SM323 DI8/D08 的地址范围()以及 5 号槽上 SM334 AI4/A02 的地址范围()。
- (14) 手编器一般采用()语言编辑。
- (15) 高速、大功率的交流负载，应选用()输出的输出接口电路。
- (16) PLC 产品手册中给出的“存储器类型”和“程序容量”是针对()存储器而言。
- (17) PLC 控制系统分()、()、()三大类。
- (18) S7-300 CPU 所用的存储区基本上由()、()、()组成。
- (19) C P U 程序所能访问的存储区为()、()、暂时局部数据存储区、外设 I / O 存储区等

二、画出下面程序正确的梯形图

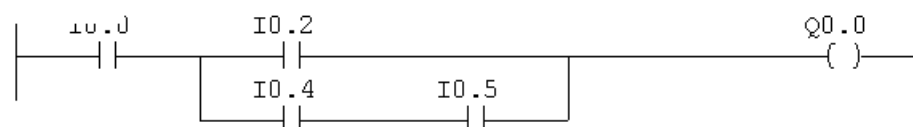
(1)



(2)



(3)



三、判断题

- 1、输入继电器只能由外部信号驱动，而不能由内部指令来驱动。
()
- 2、输出继电器可以由外部输入信号或 PLC 内部控制指令来驱动。
()
- 3、内部继电器既可以供内部编程使用，又可供外部输出。
()
- 4、PLC 内部的“软继电器”（即由 PLC 内部的存储单元构成的，包括定时器、计数器）均可提供无数副常开、常闭触点供编程使用。()
- 5、PLC 的 I/O 地址编号可以随意设定。
()

答案：1、√ 2、X 3、X 4、√ 5、X

第四章 编程语言

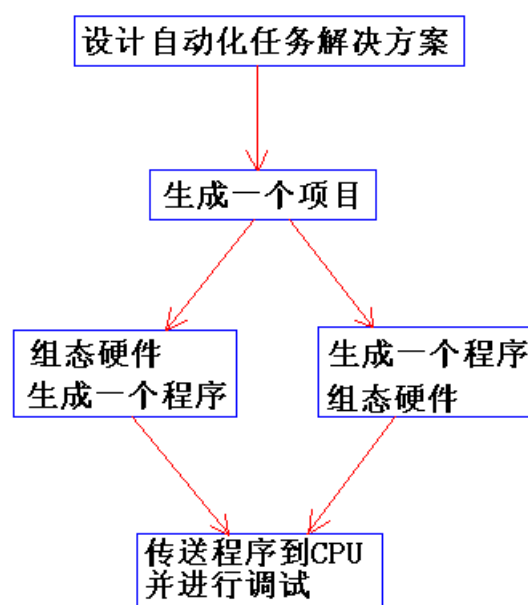
PLC 的程序

- 操作系统：操作系统由 PLC 的生产厂提供，它支持用户程序的运行
- 用户程序：用户程序是用户为完成特定的控制任务而编写的应用程序

PLC 常用的编程语言

- 梯形图编程 (LAD)
- 语句表编程 (STL)
- 功能图编程 (FBD)

使用STEP 7的基本步骤



注：对于数字量模块不一定要进行硬件组态，而对于模拟量模块一定要进行硬件组态。

4.1 指令及其结构

指令是程序的最小独立单位, 用户程序是由若干条顺序排列的指令构成.

4.1.1 指令的组成

1. 语句指令

语句指令用助记符表示 PLC 要完成的操作。

指令:操作码+操作数

操作码用来指定要执行的功能，告诉 CPU 该进行什么操作；操作数内包含为执行该操作所必需的信息，告诉 CPU 用什么地方的数据来执行此操作。

例如：操作码 操作数

0 I0.0

0 I0.1

= Q0.0

有些语句指令不带操作数, 因为它们的操作对象是唯一的。

例如: 操作码 操作数

NOT

SET

2. 梯形图指令

梯形图指令用图形元素表示 PLC 要完成的操作。在梯形图指令中, 其操作码是用图素表示的, 该图素形象表明 CPU 做什么, 其操作数的表示方法与语句指令相同。

例如:



梯形图指令也可不带操作数。

例如:



4.1.2 操作数

1. 标识符及标识参数

主标识符(操作数存放的存储器的区域): I
Q、PI、PQ、M、T、C、L、DB

标识符 

操作数  辅助标识符(操作数的位数长度): X、B、W、D

标识参数(操作数在该存储区域内的具体位置)

注释: I: 输入过程映像存储区

Q: 输出过程映像存储区

PI: 外部输入

PQ: 外部输出

M: 位存储区

T: 定时器

C: 计数器

L: 本地数据

DB: 数据块

X: 位

B: 字节

W: 字

D: 双字

注意：①PLC 物理存储器是以字节为单位的。

②当操作数长度是字或双字时，标识符后给出的标识参数是字或双字内的最低字节单元号。

③当使用宽度是字或双字的地址时，应保证没有生成任何重叠的字节分配，以免造成数据读写错误。

2. 操作数的表示法

操作数的表示法 物理地址(绝对地址)
符号地址(必须先定义后使用，而且符号名必须是唯一的)

关于定义符号地址的几点说明：

(一) 当你在表中输入符号地址时，应注意以下几点：

列	注意
符号	在整个符号表中名字必须唯一。当你确认该区域的输入或退出该区域时，不唯一的符号则被标定出来。符号名最长可达 24 个字符。引号（“”）不允许使用。
地址	当你确认该区域的输入或退出该区域时，程序会自动检查该地址输入是否是允许的。
数据类型	当你确认或退出地址时，该区域被自动地赋予一个缺省数据类型。如果你修改这个缺省类型，程序会检查你的数据类型是否与地址相匹配。
注释	你可以输入注释简单地解释该符号的功能（最多 80 个字符）。

(二) 你必须区分局域（块定义）符号和共享符号

	共享符号	局域符号
有效性	- 在整个用户程序中有效 - 可以被所有的块使用 - 在所有的块中含义是一样的 - 在整个用户程序中是唯一的	- 只在定义的块有效 - 相同的符号可在不同的块中用于不同的目的
允许使用的字符	- 字母、数字及特殊字符。 - 除 0X00, 0XFF 及引号以外的强调用号 - 如使用特殊字符，则符号须写出在引号内。	- 字母 - 数字 - 下划线（_）（注意：不允许使用两个连续的下划线）

使用	你可以为以下各项定义共享符号： • I/O 信号（I，IB，IW，ID，Q，QB，QW，QD） • I/O 输入与输出（PI，PQ） • 存储位（M，MB，MW，MD） • 定时器（T）/计数器（C） • 逻辑块（FB，FC，SFB，SFC） • 数据块（DB） • 用户定义数据类型（UDT） • 变量表（VAT）	你可以为以下各项定义局域符号： • 块参数（输入，输出和输入输出参数） • 块的静态数据 • 块的临时数据
在 哪 里 定 义	符号表	块的变量声明表

（三）显示共享或局域符号

你可以在程序的指令部分区分开共享符号和局域符号。

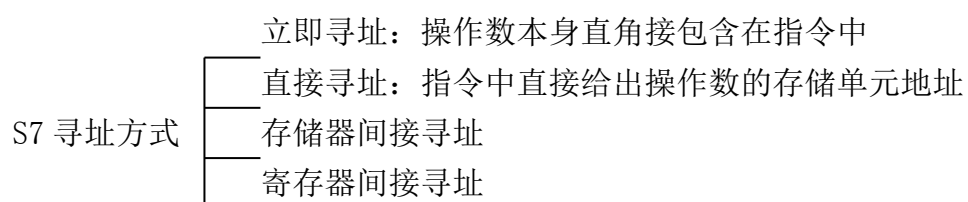
- 符号表中定义的符号（共享）显示在引号内。
- 块变量声明表中的符号（局域）显示时前面加上“#”。

提示：

使用菜单命令 View>Display>Symbolic Representation，你可以在所有声明的符号地址和绝对地址之间进行切换。

4.1.3 寻址方式

寻址方式是指令得到操作数的方式。



4.1.4 数据类型

数据类型决定了你以什么方式或格式理解或访问存储区中的数据。

状态字用于表示 CPU 执行指令时所具有的状态。

15	.	.	.	.	9	8	7	6	5	4	3	2	1	0
						BR	CC1	CC0	OS	OV	OR	STA	RLO	$\overline{PC}$
						.	.	.	.	.	.	.	.	.
						.	.	.	.	.	.	.	.	.
						.	.	.	.	.	.	.	.	.
						二进制结果位	条件码 1	条件码 0	溢出状态保持位	溢出位	或位	状态位	逻辑操作结果位	首次检测位

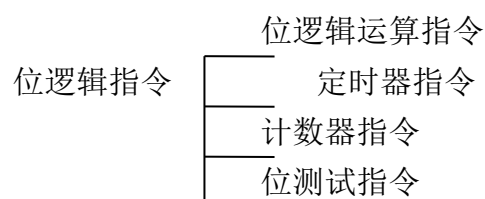
- 1、在段中选择一点，你想在该点后面插入一个梯形图组件。
- 2、用下列方法之一，在段中插入所需的组件：
 - 在菜单“Insert”中选择合适的菜单命令，例如，
Insert>LAD Element>Normally Open Contact
 - 用功能键 F2、F3 或 F7 输入一个常开触点、常闭触点或输出线圈。
 - 选择菜单命令 Insert>Program Elements 打开“program Elements（编程组件）”对话框并在目录中选择所需的组件。

所选的梯形图组件被插入，问号被用来表示地址和参数。

- 1、通过点击灰色注释框下面的任意区域就可打开正文框（或者若不显示段注释则在段标题的下面）。
- 2、输入指令、按空格键，然后是地址（直接或间接地址）。
- 3、按空格键并输入以双斜线//开始的注释（可选）。
- 4、在完成一条（一行）带注释或不带注释的语句后按 RETURN。

一行完成后，运行语法检查，这条语句形成并显示，指令中或绝对地址中的任何小写字母都转换为大写。任何查到的语法错误都显示为红色斜体，在存储该逻辑块之前必须修改所有错误。

4.2 位逻辑指令



4.2.1 位逻辑运算指令

1. “与”、“或”、“异或”指令

(1) 语句指令

布尔逻辑串内的真值表（根据下列表可以确定第二条布尔位操作后的 RLO）

助记符	指令	指令前 RLO	地址状态	RLO 结果
A	与	0	0	0
		0	1	0
		1	0	0
		1	1	1
AN	与非	0	0	0
		0	1	0
		1	0	1
		1	1	0
O	或	0	0	0
		0	1	1
		1	0	1
		1	1	1
ON	或非	0	0	1
		0	1	0
		1	0	1
		1	1	1
X	异或	0	0	0
		0	1	1
		1	0	1
		1	1	0
XN	异或非	0	0	1
		0	1	0
		1	0	0
		1	1	1

布尔逻辑串开始的真值表

助记符	指令	地址状态	RLO 结果
A	与	0	0
		1	1
AN	与非	0	1
		1	0
O	或	0	0
		1	1
ON	或非	0	1
		1	0
X	异或	0	0
		1	1
XN	异或非	0	1
		1	0

(2) 梯形图逻辑指令

①常开接点（动合触点）元素和参数

LAD元素	参数	数据类型	存储区	说明
地址 	地址	BOOL, TIMER, COUNTER	I, Q, M, T C, L, D	地址指明要检查信号状态的位

②常闭接点（动断触点）元素和参数

LAD元素	参数	数据类型	存储区	说明
地址 	地址	BOOL, TIMER, COUNTER	I, Q, M, T C, L, D	地址指明要检查信号状态的位

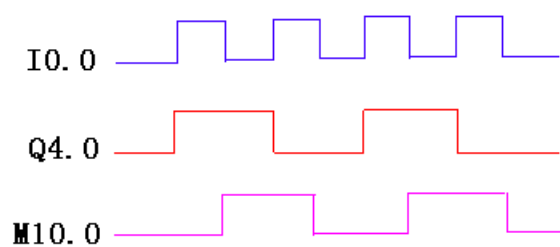
2. 输出指令

输出指令把状态字中 RLO 的值赋给指定的操作数

STL 指令	LAD 指令	功能	操作数	数据类型	存储区
=<地址>	<地址> --- ()	逻辑串赋值输出	<位地址>	BOOL	I, Q, M, D, L
	<地址> -- (#) --	中间结果赋值输出	<位地址>	BOOL	I, Q, M, D, L

例一 二分频器

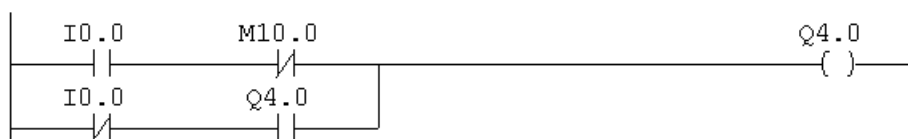
二分频器是一种具有一个输入端和一个输出端的功能单元, 输出频率为输入频率的一半。如下, 输入为 I0.0, 输出为 Q4.0。



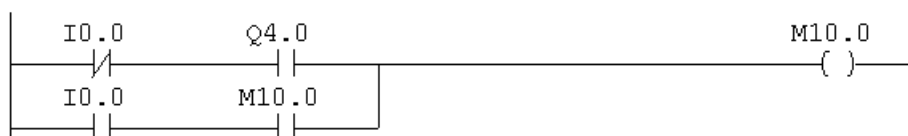
二分频时序图

梯形图程序:

Network 1:



Network 2:



语句表程序:

Network 1:

A	I	0.0
AN	M	10.0
○		
AN	I	0.0
A	Q	4.0
=	Q	4.0

Network 2:

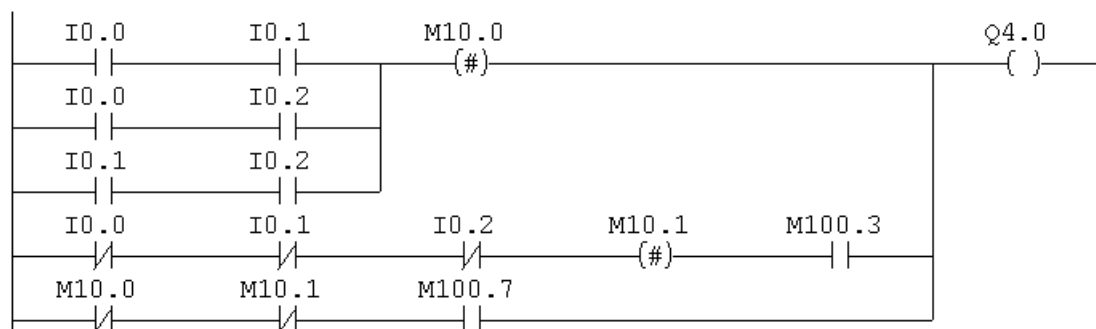
AN	I	0.0
A	Q	4.0
○		
A	I	0.0
A	M	10.0
=	M	10.0

例二 风机监控程序

某设备有三台风机，当设备处于运行状态时，如果风机至少有两台以上转动，则指示灯常亮；如果仅有一台风机转动，则指示灯以 0.5Hz 的频率闪烁；如果没有任何风机转动，则指示灯以 2Hz 的频率闪烁。当设备不运行时，指示灯不亮。

梯形图程序:

Network 1:



语句表程序:

Network 1:

```

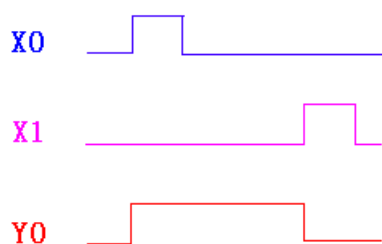
A(
A    I    0.0
A    I    0.1
O
A    I    0.0
A    I    0.2
O
A    I    0.1
A    I    0.2
)
=    M    10.0
A    M    10.0
O(
AN   I    0.0
AN   I    0.1

AN   I    0.2
=    M    10.1
A    M    10.1
A    M    100.3
)
O
AN   M    10.0
AN   M    10.1
A    M    100.7
=    Q    4.0
    
```

输入位 I0.0, I0.1, I0.2 分别表示风机 1, 2, 3。存储位 M100.3 为 2Hz 的频率信号, M100.7 为 0.5Hz 的信号。风机转动状态指示灯由 Q4.0 控制。存储位 10.0 为 1 时用于表示至少有两台风机转动, M10.1 为 1 时表示没有风机转动。

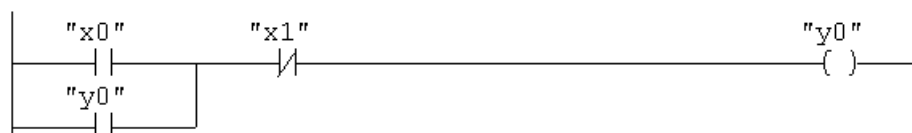
例三 启动和自锁程序

程序功能: 输入 X0 闭合时, 输出 Y0 闭合且自锁。只有在 X1 闭合时, 其动断触点打开, Y0 断开。其时序图如下。



梯形图程序:

Network 1:



语句表程序:

Network 1:

```

A (
  O      "x0"
  O      "y0"
)
AN      "x1"
=        "y0"
    
```

练习: 请指出以下实现输出 Q0.0、Q0.1 互锁程序的错误, 并改正。

Network 1: Title:



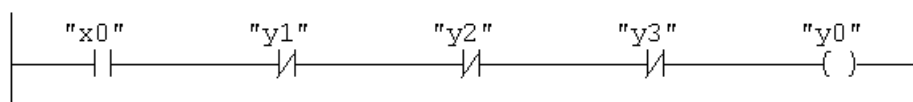
Network 2: Title:



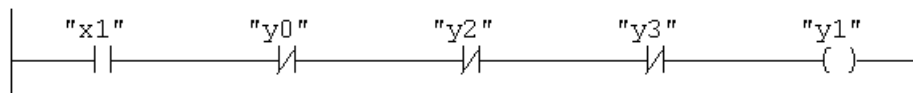
例四 优先程序

优先程序执行时, 能在多个输入信号中仅接收最先一个输入信号作出反映, 其后的输入信号不接收。此原则常用于抢答器中。

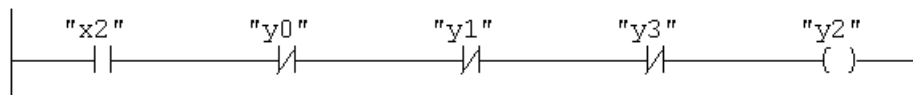
Network 1 :



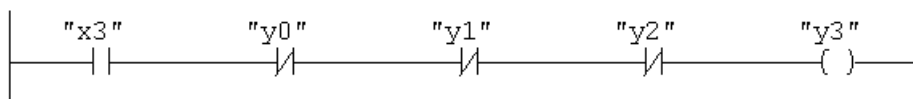
Network 2 :



Network 3 :



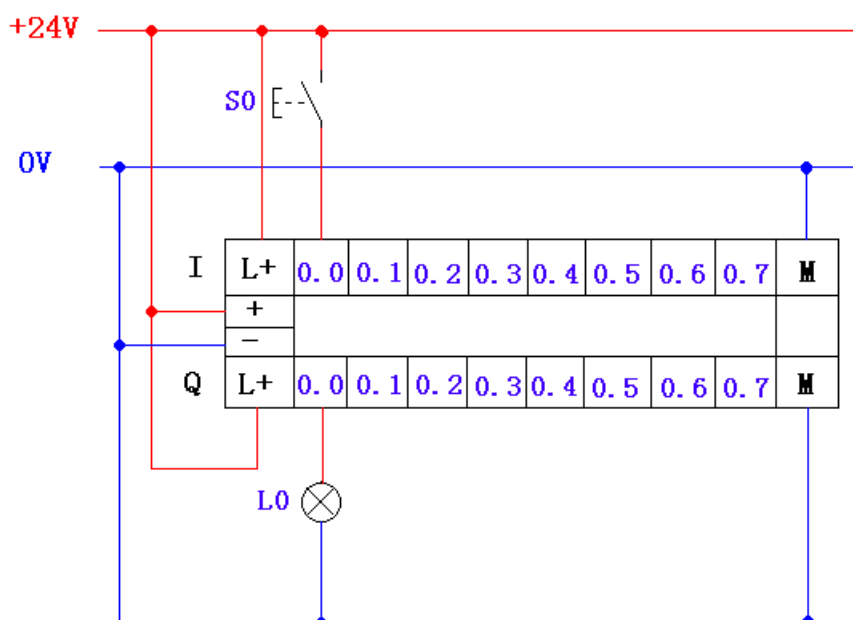
Network 4 :



例五 灯泡控制程序

一盏灯泡由一个按钮来控制，已知第一次按下按钮，灯泡亮，第二次按下按钮，灯光灭。

(一) PLC 接线图



(二) 定义符号地址

符号地址	绝对地址	数据类型	说明
S0	I0.0	BOOL	按钮

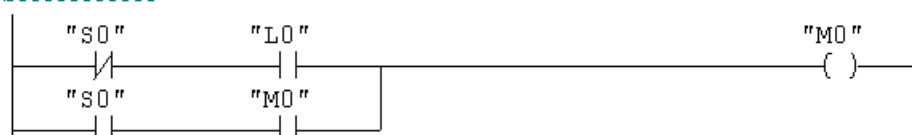
L0	Q0.0	BOOL	灯泡
M0	M0.0	BOOL	标标位

(三) 梯形图程序

Network 1:



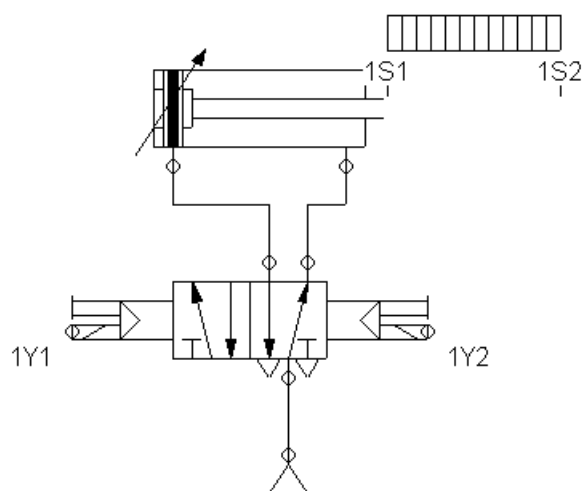
Network 2:



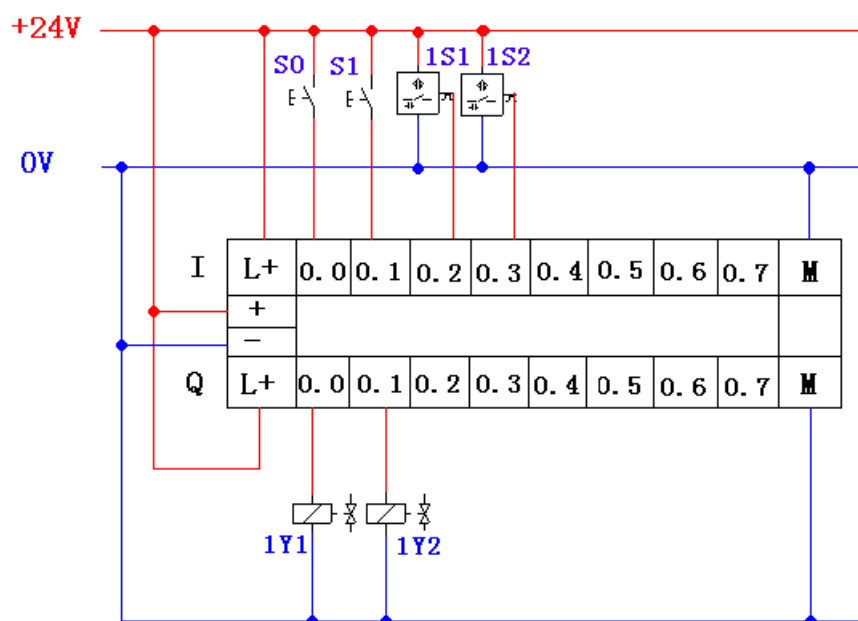
例六 双作用气缸连续往复运动控制

按启动按钮双作用气缸连续往复运动，按停止按钮，停止运动。

(一) 气控回路



(二) PLC 接线



(三) 定义符号地址

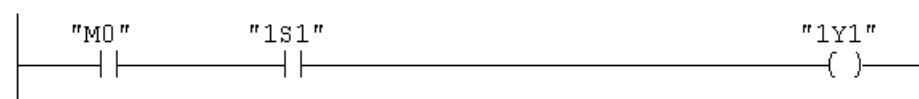
符号地址	绝对地址	数据类型	说明
S0	I0.0	BOOL	启动按钮
S1	I0.1	BOOL	停止按钮
1S1	I0.2	BOOL	位置传感器
1S2	I0.3	BOOL	位置传感器
1Y1	Q0.0	BOOL	换向阀电磁线圈
1Y2	Q0.1	BOOL	换向阀电磁线圈
M0	M0.0	BOOL	启动线圈

(四) 梯形图程序

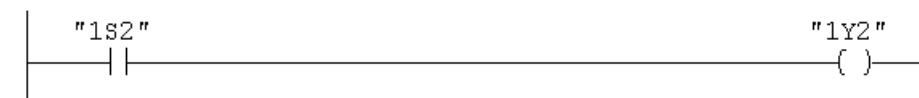
Network 1:



Network 2:



Network 3:



练习：编制电动机点动和连续运转控制程序。

输入端	控制触点	控制功能	输出端	执行器件	实现功能
I0.0	停止按钮 (常开)	停止	Q0.0	KM1	控制电动机电源
I0.1	点动按钮 (常开)	点动	Q0.1	保护指示灯	当保护动作时闪亮
I0.2	连续运转按钮(常开)	连续运转启动			
I0.3	热继电器触点(常闭)	过载保护动作 使得电机停止			

参考程序：

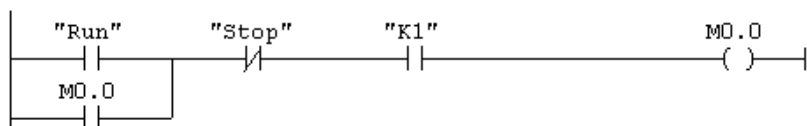
(一) 定义符号地址

	Symbol	Address	Data Type	Comment
1	K1	I 0.3	BOOL	热继电器
2	Lamp	Q 0.1	BOOL	保护指示灯
3	Motor	Q 0.0	BOOL	电机
4	Run	I 0.2	BOOL	连续按钮
5	Start	I 0.1	BOOL	点动按钮
6	Stop	I 0.0	BOOL	停止按钮

(二) LAD 程序

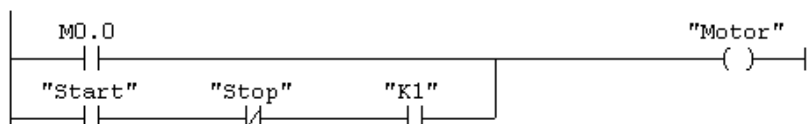
Network 1 : Title:

Comment:



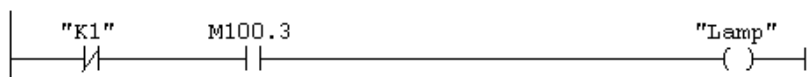
Network 2 : Title:

Comment:



Network 3 : Title:

Comment:



练习：编制双控灯控制程序，要求开关 K1 或 K2 任意一个开关的开和关的操作均可以控制灯 L 的亮和灭。完成后可再编制三控灯。

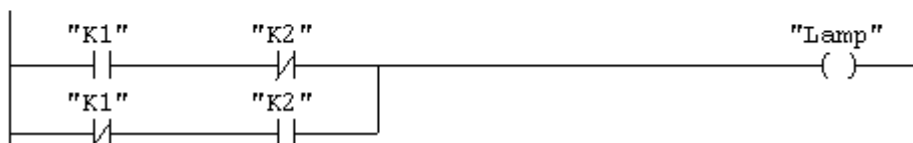
参考程序

程序 1:

	Symbol	Address	Data Type	Comment
1	K1	I 0.0	BOOL	开关K1
2	K2	I 0.1	BOOL	开关K2
3	Lamp	Q 0.0	BOOL	灯泡

Network 1 : Title:

Comment:



3. 置位 / 复位指令

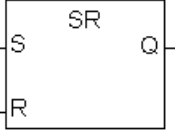
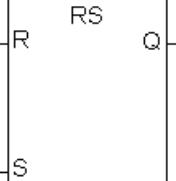
STL 指令	LAD 指令	功能	操作数	数据类型	存储区
S<位地址>	<位地址> ---(S)	置位输出	<位地址>	BOOL	I, Q, M, D, L

R<位地址>	<位地址> —(R)	复位输出	<位地址>	BOOL, TIMER, COUNTER	I, Q, M, D , L, T, C
--------	---------------	------	-------	-------------------------	-------------------------

复位/置位指令根据 RLO 的值，来决定被寻址位的信号状态是否需要改变。

若 RLO 的值为 1, 被寻址位的信号状态被置 1 或清 0; 若 RLO 的值为 0, 被寻址位的信号保持原状态不变。这一特性又称为静态的置位 / 复位。相应地，赋值输出被称为动态赋值输出。**在 LAD 中置位 / 复位指令要放在逻辑串最右端，而不能放在逻辑串中间。**

4. RS 触发器

置位复位触发器	复位置位触发器	参数	数据类型	存储区
		<位地址> 需要 置位-复位的位 S 允许置位输入 R 允许复位输入 Q 地址的状态	BOOL	I, Q, M, D, L

置位复位触发器真值表

S	R	Q
0	0	—
0	1	0
1	0	1
1	1	0

复位置位触发器真值表

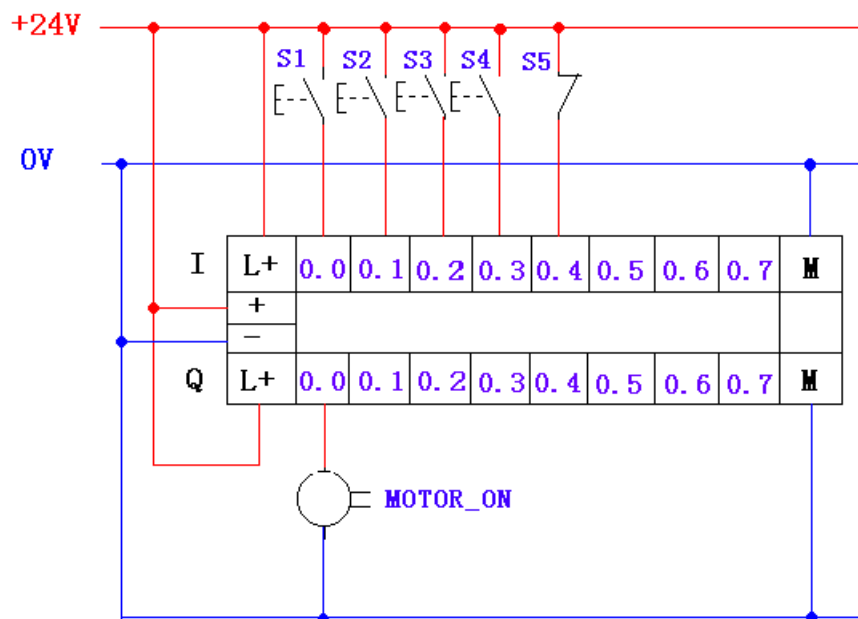
R	S	Q
0	0	—
0	1	1
1	0	0
1	1	1

在 LAD 中，RS 触发器可以用在逻辑串最右端，结束一个逻辑串，也可用在逻辑串中，影响右边的逻辑操作结果。

例一 控制传送带

一个由电气启动的传送带，在传送带的起点有两个按钮开关：用于 START 的 S1 和 STOP 的 S2。在传送带的尾部也有两个按钮开关：用于 START 的 S3 和 STOP 的 S4。可以从任何一端启动或停止传送带。另外，当传送带上的物件到达末端时，传感器 S5 使传送带停机。

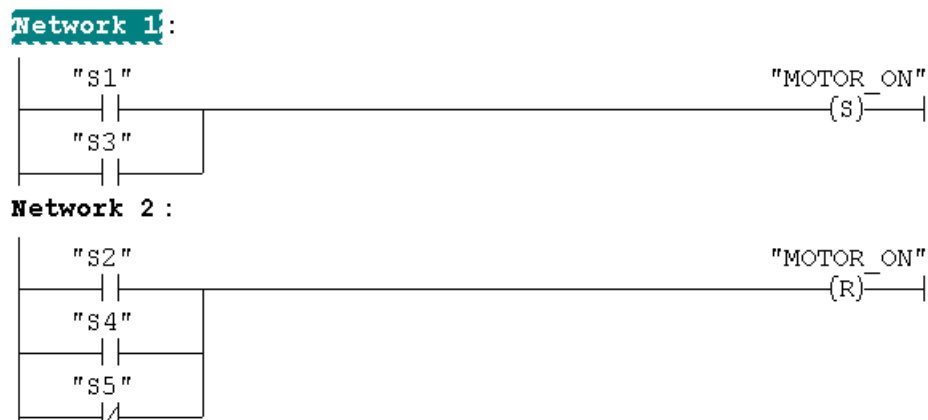
(一) PLC 接线



(二) 定义符号地址

符号地址	绝对地址	数据类型	说明
S1	I0.0	BOOL	起点启动按钮
S2	I0.1	BOOL	起点停机按钮
S3	I0.2	BOOL	尾部启动按钮
S4	I0.3	BOOL	尾部停机按钮
S5	I0.4	BOOL	末端传感器
MOTOR_ON	Q0.0	BOOL	电机

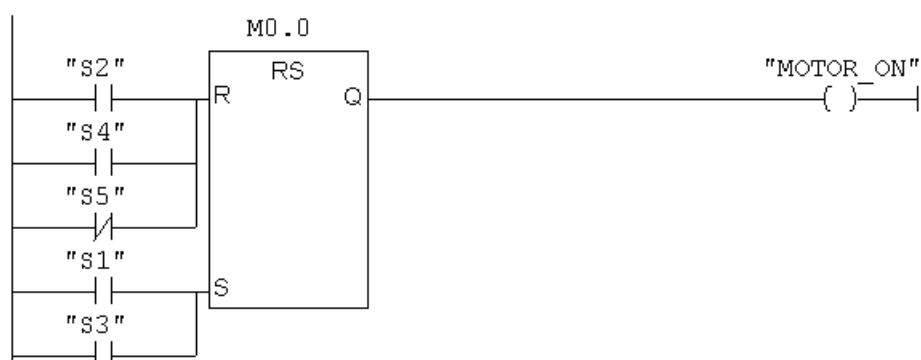
(三) 梯形图程序



思考题：

如下的程序有什么不足之处？应如何改正？

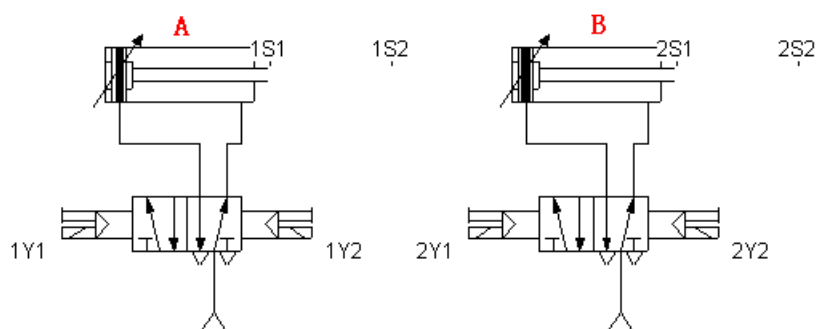
Network 1:



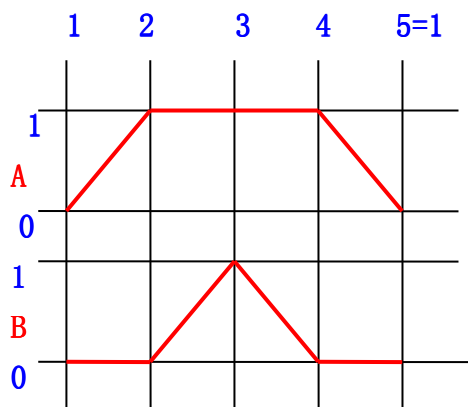
例二 双缸顺序动作控制程序

设计程序, 使两个气缸顺序动作, 其顺序为:A1B1B0A0。

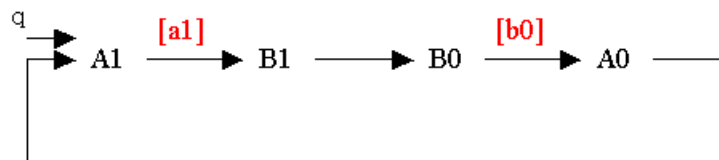
(一) 气控回路



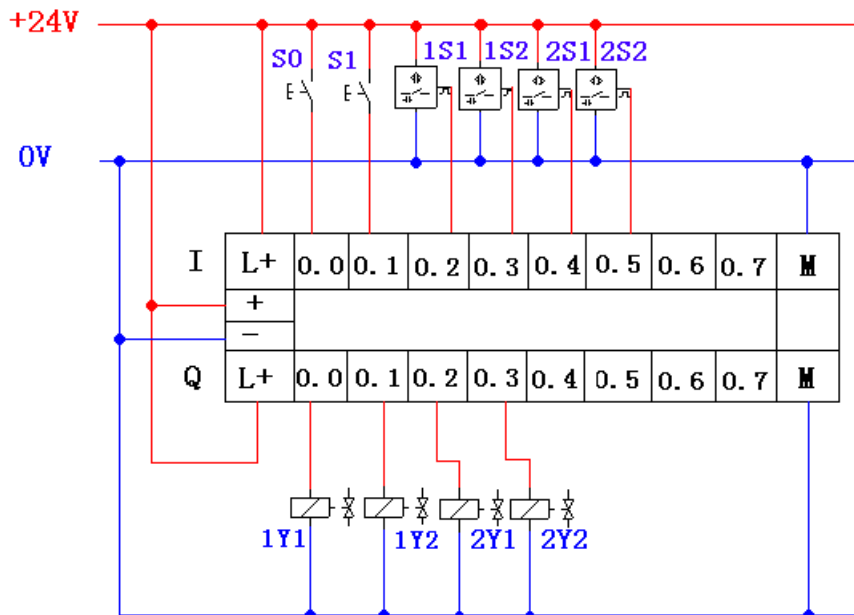
(二) 位移-步骤图



(三) I 型障碍信号分析



(四) PLC 接线

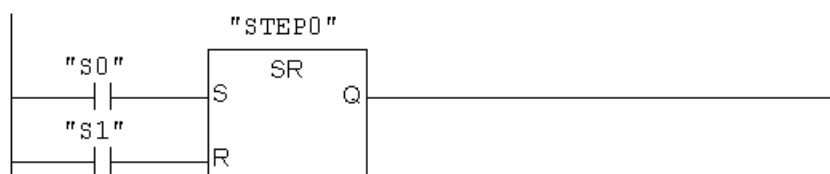


(五) 定义符号地址

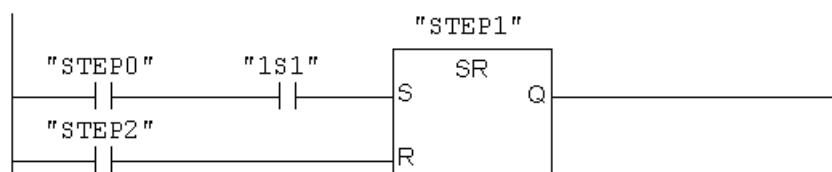
	Symbol	Address	Data Type
1	S0	I 0.0	BOOL
2	S1	I 0.1	BOOL
3	1S1	I 0.2	BOOL
4	1S2	I 0.3	BOOL
5	2S1	I 0.4	BOOL
6	2S2	I 0.5	BOOL
7	1Y1	Q 0.0	BOOL
8	1Y2	Q 0.1	BOOL
9	2Y1	Q 0.2	BOOL
10	2Y2	Q 0.3	BOOL
11	STEP0	M 0.0	BOOL
12	STEP1	M 0.1	BOOL
13	STEP2	M 0.2	BOOL
14	STEP3	M 0.3	BOOL
15	STEP4	M 0.4	BOOL

(六) 梯形图程序

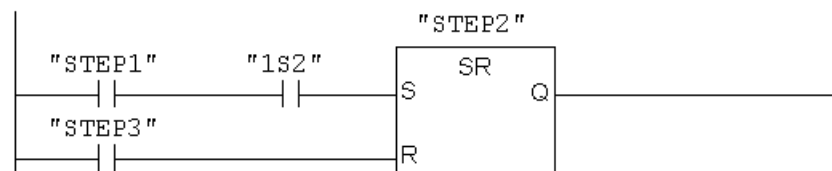
Network 1:



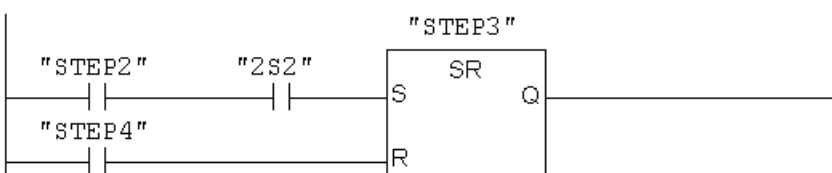
Network 2:



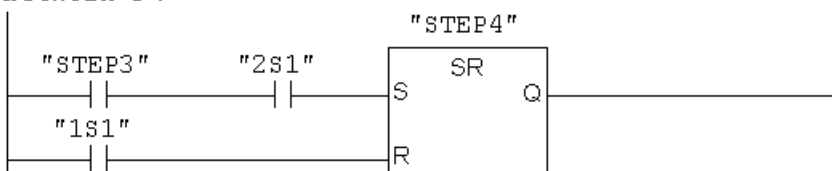
Network 3 :



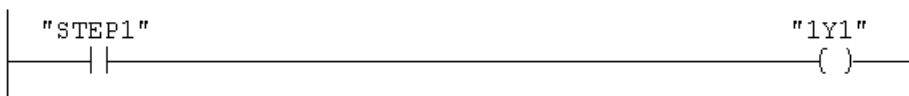
Network 4 :



Network 5 :



Network 6 :



Network 7 :



Network 8 :



Network 9 :



5. RLO 上升沿、下降沿检测指令

LAD 指令	STL 指令	功能	操作数	数据类型	存储区
<位地址> - (P) -	FP<位地址>	RLO 上升 沿检测	<位地址> 存储旧	BOOL	I、Q、M、 D、L
<位地址> - (N) -	FN<位地址>	RLO 下降 沿检测	RLO 的边 沿存储位	BOOL	I、Q、M、 D、L

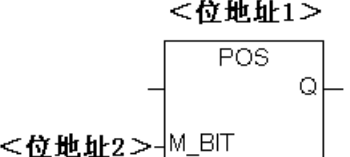
RLO 上升沿检测指令识别 RLO 从 0 至 1（上升沿）的信号变化，并且在操

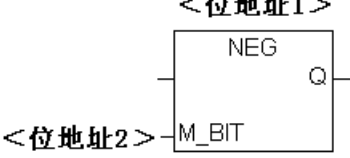
作之后以 $RLO=1$ 表示这一变化。用边沿存储位比较 RLO 的现在的信号状态与该地址上周期的信号状态，如果操作之前地址的信号状态是 0，并且现在 $RLO=1$ ，那么操作之后， RLO 将为 1（脉冲），所有其它的情况为 0。在该操作之前， RLO 存储于地址中。

RLO 下降沿检测指令识别 RLO 从 1 至 0（下降沿）的信号变化，并且在操作之后以 $RLO=1$ 表示这一变化。用边沿存储位比较 RLO 的现在的信号状态与该地址上周期的信号状态，如果操作之前地址的信号状态是 1，并且现在 $RLO=0$ ，那么操作之后， RLO 将为 1（脉冲），所有其它的情况为 0。在该操作之前， RLO 存储于地址中。

如果 RLO 在相邻的两个扫描周期中相同（全为 1 或 0），那么 FP 或 FN 语句把 RLO 位清 0。

6. 地址上升沿、下降沿检测指令

地址上升沿检测	参数	数据类型	存储区
	位地址1被检测的位	BOOL	I, Q, M, D, L
	位地址2存储被检测位上一个扫描周期的状态	BOOL	Q, M, D
	Q单稳输出	BOOL	I, Q, M, D, L

地址下降沿检测	参数	数据类型	存储区
	位地址1被检测的位	BOOL	I, Q, M, D, L
	位地址2存储被检测位上一个扫描周期的状态	BOOL	Q, M, D
	Q单稳输出	BOOL	I, Q, M, D, L

地址上升沿检测指令将<位地址 1>的信号状态与存储在<位地址 2>中的先前信号状态检查时的信号状态比较。如果有从 0 至 1 的变化的话，输出 Q 为 1，否则为 0。

地址下降沿检测指令将<位地址 1>的信号状态与存储在<位地址 2>中的先前信号状态检查时的信号状态比较。如果有从 1 至 0 的变化的话，输出 Q 为 1，否则为 0。

在梯形图中，地址跳变沿检测方块和 RS 触发器方块可被看作一个特殊常开触点。该常开触点的特性：若方块的 Q 为 1，触点闭合；若 Q 为 0，则触点断开。

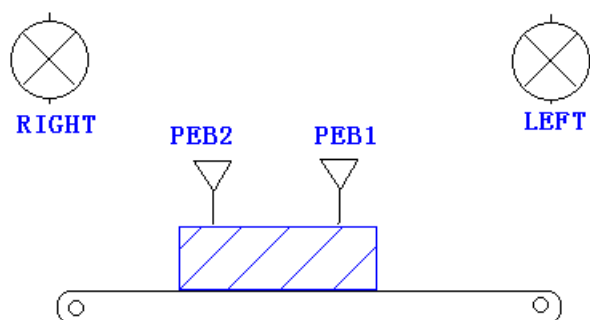
7. 对 RLO 的直接操作指令

LAD 指令	STL 指令	功能	说明
- NOT -	NOT	取反 RLO	在逻辑串中, 对当前的 RLO 取反; 取反指令或置位 STA

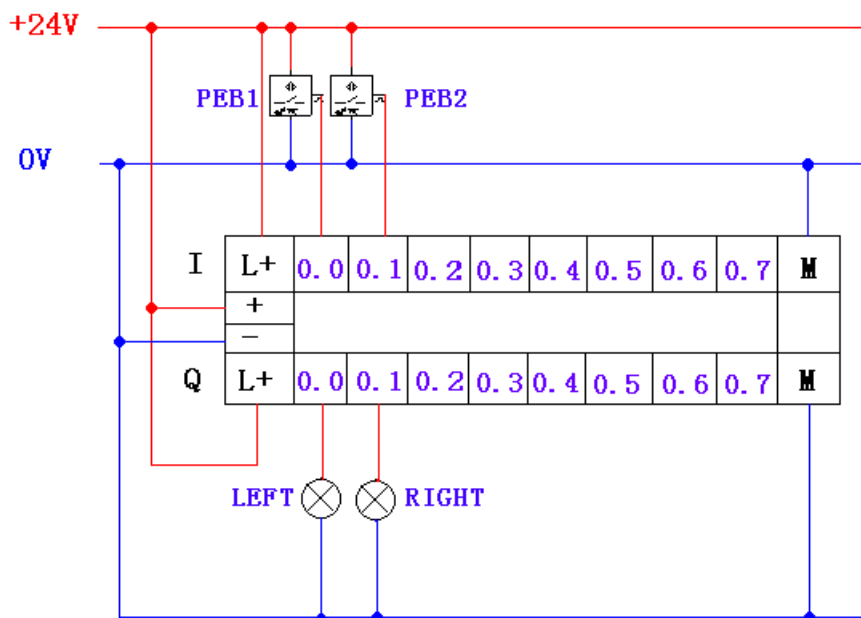
-----	SET	置位 RLO	把 RLO 无条件置 1 并结束逻辑串； 使 STA 置 1, OR FC 清 0
-----	CLR	复位 RLO	把 RLO 无条件清 0 并结束逻辑串； 清 0 STA, OR FC
-(SAVE)	SAVE	保存 RLO	把 RLO 存入状态字的 BR 位, 该指令 不影响其它状态位

例一 检测传送带的方向

装备有两个光电传感器(PEB1 和 PEB2)的传送带，该设计能够检测传送带上物件的运动方向，并通过左右两端的指示灯（LEFT 灯和 RIGHT 灯）显示。



(一) PLC 接线

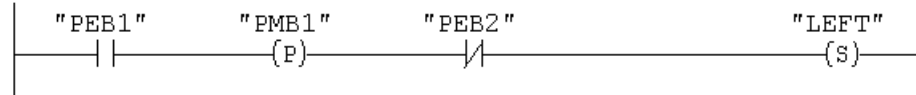


(二) 定义符号地址

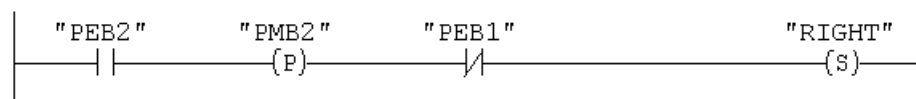
	Symbol	Address	Data Type
1	PEB1	I 0.0	BOOL
2	PEB2	I 0.1	BOOL
3	LEFT	Q 0.0	BOOL
4	RIGHT	Q 0.1	BOOL
5	PMB1	M 0.0	BOOL
6	PMB2	M 0.1	BOOL

(三) 梯形图程序

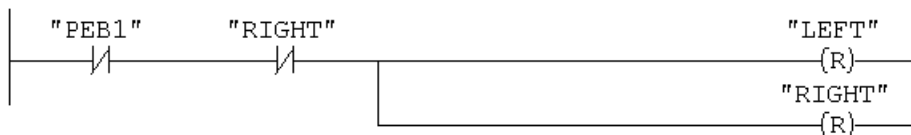
Network 1:



Network 2:

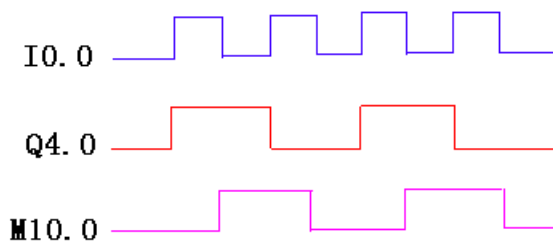


Network 3:



例二 二分频器

二分频器是一种具有一个输入端和一个输出端的功能单元, 输出频率为输入频率的一半。如下, 输入为 I0.0, 输出为 Q4.0。

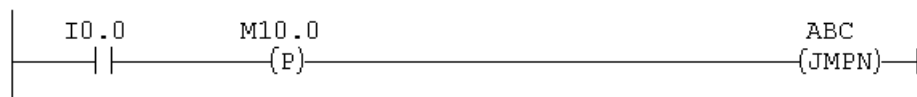


二分频时序图

分析二分频的时序图看到, 输入每有一个正跳沿, 输出便反转一次。据此, 可用跳变沿检测指令实现分频功能。

梯形图程序 1:

Network 1:



Network 2:



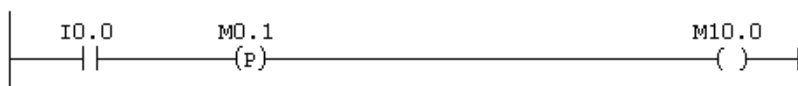
Network 3:



梯形图程序 2:

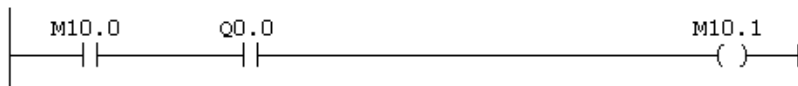
Network 1: Title:

Comment:



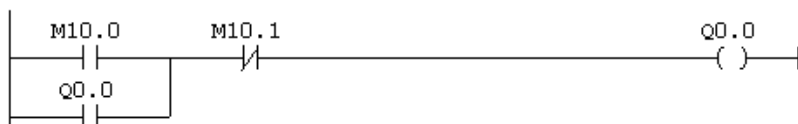
Network 2: Title:

Comment:



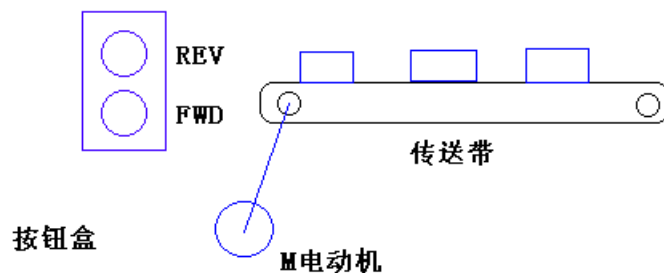
Network 3: Title:

Comment:

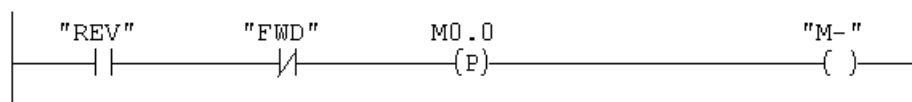
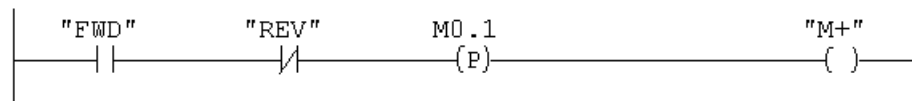


例三 传送带定位控制

一电动机带动一个传送带运动，要求移动传送带向前或向后到达某一确定的位置，其结构示意图如下，为了正确定位该传送带，有时需要按下向后（REV）或向前（FWD）按钮进行手动调整。

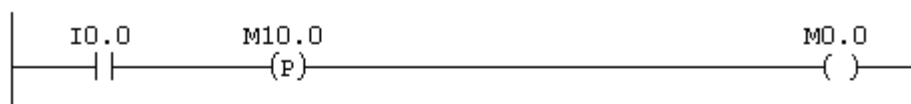
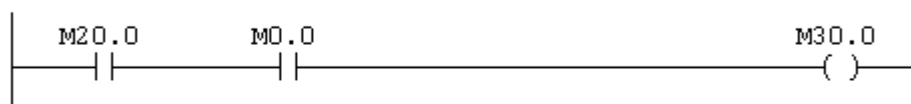
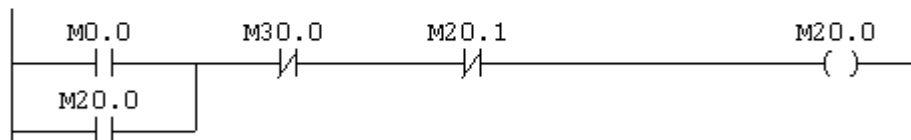
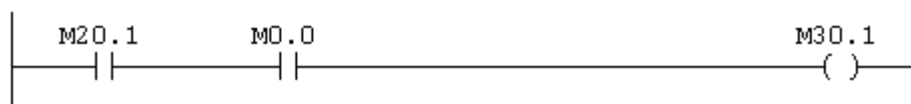
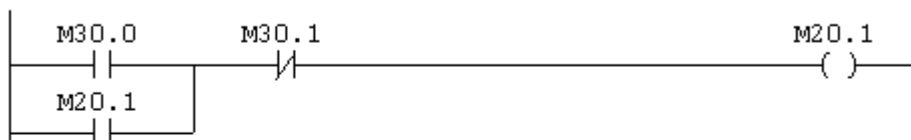
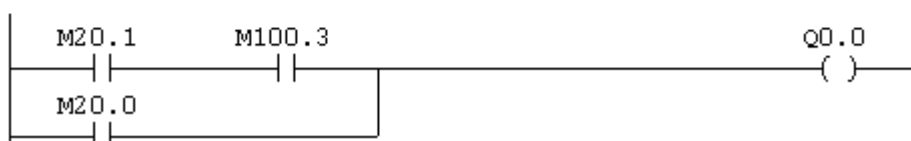


梯形图程序:

Network 1 :**Network 2 :**

一旦有按钮按下，立即驱动输出，电动机运转一个扫描周期。这也意味着按钮时间长短与电动机驱动的时间没有关系。

练习：第一次按按钮指示灯亮，第二次按按钮指示灯闪亮，第三次按下按钮指示灯灭，如此循环，试编写其 PLC 控制的 LAD 程序。

Network 1 : Title:**Network 2 : Title:****Network 3 : Title:****Network 4 : Title:****Network 5 : Title:****Network 6 : Title:**

4.2.2 定时器指令

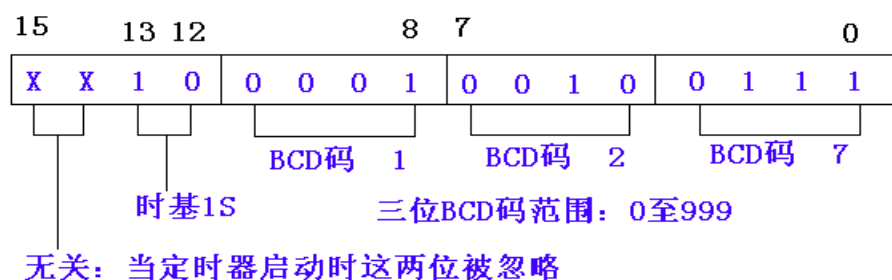
定时器可以提供等待时间或监控时间, 定时器还可产生一定宽度的脉冲, 亦可测量时间。定时器是一种由位和字组成的复合单元, 定时器的触点由位表示, 其定时时间值存储在字存储器中。

定时器的种类	脉冲定时器 (SP)
	扩展脉冲定时器 (SE)
	接通延时定时器 (SD)
	保持型接通延时定时器 (SS)
	关断延时定时器 (SF)

1. 定时器组成

在 CPU 的存储器中留出了定时器区域, 该区域用于存储定时器的定时时间值。每个定时器为 2Byte, 称为定时字。在 S7-300 中, 定时器区为 512Byte, 因此最多允许使用 256 个定时器。S7 中定时时间由时基和定时值两部分组成, 定时时间等于时基与定时值的乘积。当定时器运行时, 定时值不断减 1, 直至减到 0, 减到 0 表示定时时间到。定时时间到后会引引起定时器触点的动作。

定时器的第 0 到第 11 位存放二进制格式的定时值, 第 12, 13 位存放二进制格式的时基。



时基与定时范围

时基	时基的二进制代码	分辨率	定时范围
10ms	00	0.01s	10ms 至 9s_990ms
100ms	01	0.1s	100ms 至 1m_39s_900ms
1s	10	1s	1s 至 16m_39s
10s	11	10s	10s 至 2h_46m_30s

为累加器 1 装入定时时间值的表示方法:

(1) L W#16#wxyz

其中, w, x, y, z 均为十进制数;

w=时基，取值 0，1，2，3，分别表示时基为：10ms, 100ms, 1s, 10s；

xyz=定时值，取值范围：1 到 999。

(2) L S5T# aH_bbM_ccS_dddMS

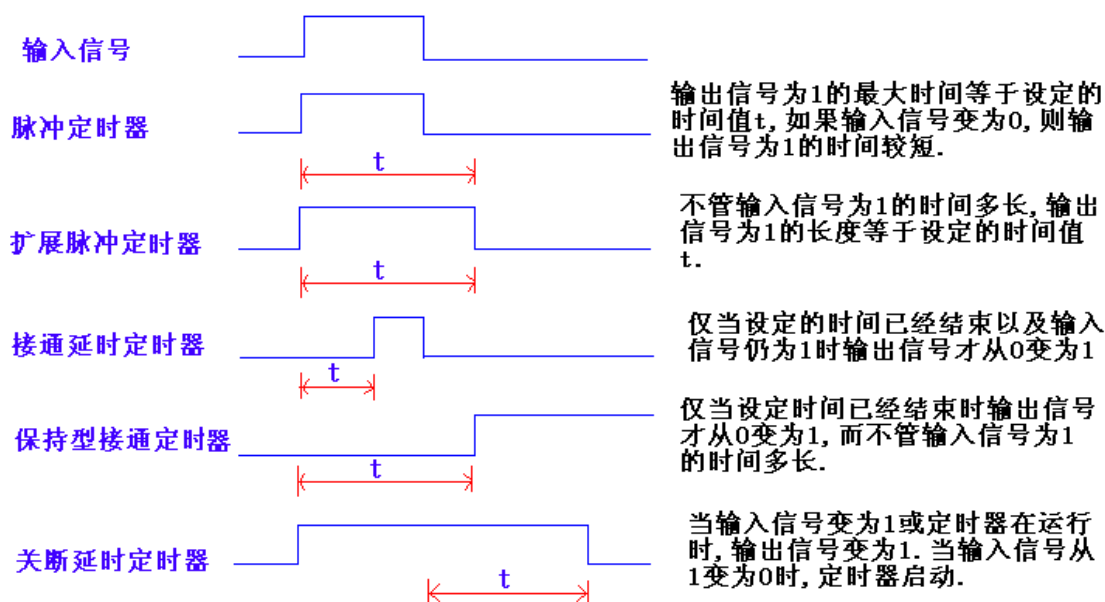
2. 定时器启动与运行

PLC 中的定时器相当于时间继电器。在使用时间继电器时，要为其设置定时时间，当时间继电器的线圈通电后，时间继电器被启动。若定时时间到，继电器的触点动作。当时间继电器的线圈断电时，也将引起其触点的动作。该触点可以在控制线路中，控制其它继电器。

3. 定时器启动指令

LAD 指令	STL 指令	功能
T no. ---(SP) 时间值	SP T no.	启动脉冲定时器
T no. ---(SE) 时间值	SE T no.	启动扩展脉冲定时器
T no. ---(SD) 时间值	SD T no.	启动接通延时定时器
T no. ---(SS) 时间值	SS T no.	启动保持型接通延时定时器
T no. ---(SF) 时间值	SF T no.	启动关断延时定时器
	FR T no.	允许再启动定时器

各种定时器的工作特点：



4. 定时器的梯形图方块指令

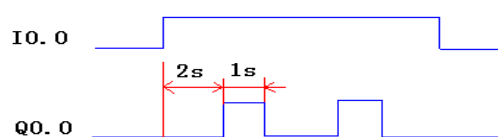
脉冲定时器	扩展脉冲定时器	接通延时定时器	保持接通定时器	关断延时定时器
Tno. S_PULSE Q TV BI R BCD	Tno. S_PEXT Q TV BI R BCD	Tno. S_ODT Q TV BI R BCD	Tno. S_ODTS Q TV BI R BCD	Tno. S_OFFDT Q TV BI R BCD

参数	数据类型	存储区	说明
NO.	TIMER	T	定时器标识号, 与 CPU 有关
S	BOOL	I, Q, M, D, L	启动输入
TV	S5TIME	I, Q, M, D, L	设定定时间 (S5TIME 格式)
R	BOOL	I, Q, M, D, L	复位输入
Q	BOOL	I, Q, M, D, L	定时器状态输出
BI	WORD	I, Q, M, D, L	剩余时间输出 (二进制格式)
BCD	WORD	I, Q, M, D, L	剩余时间输出 (BCD 码格式)

5. 定时器应用举例

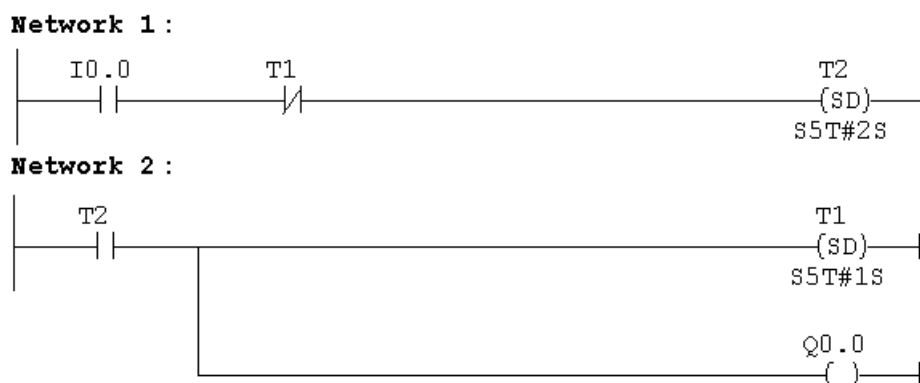
例一 脉冲发生器

用定时器可构成脉冲发生器, 这里用了两个定时器产生频率占空比均可设置的脉冲信号。如下图的脉冲发生器的时序图, 当输入 I0.0 为 1 时, 输出 Q0.0 为 1 或 0 交替进行, 脉冲信号的周期为 3s, 脉冲宽度为 1s。



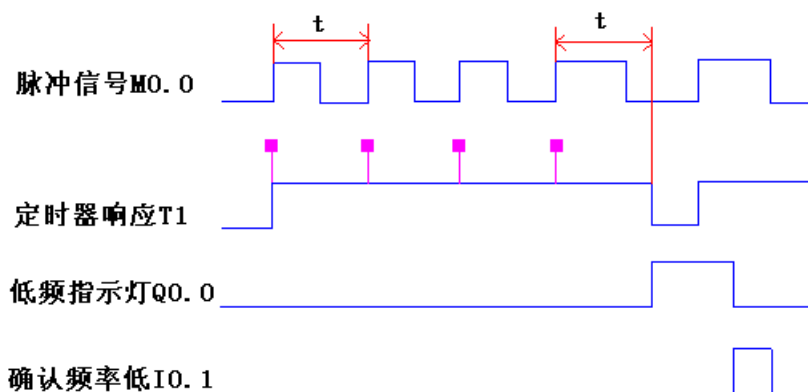
脉冲发生器时序

梯形图程序:



例二 频率监测器

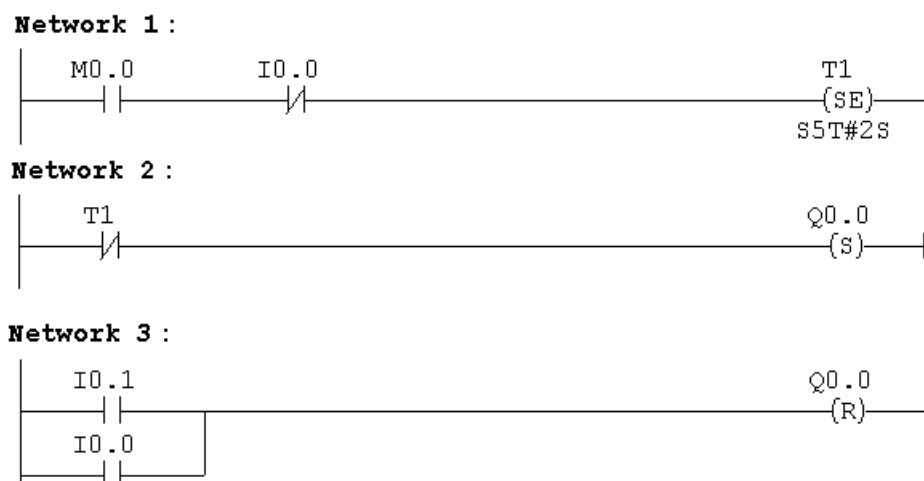
频率监测器用于监测脉冲信号的频率，若其低于下限，则指示灯亮，“确认”按键能使指示灯复位。为此，使用了一个扩展脉冲定时器，每当频率信号有一个上升沿就启动一次定时器。如果超过了定时时间没有启动定时器，则表明两个脉冲之间的时间间隔太长，即频率太低了。



■ 定时器再触发
t=定时时间

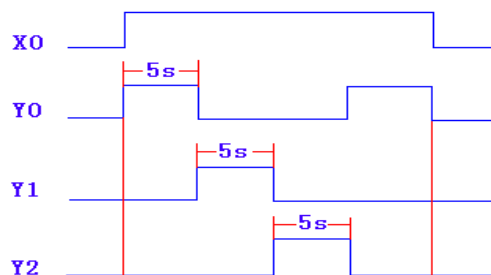
频率监测器时序

梯形图程序:



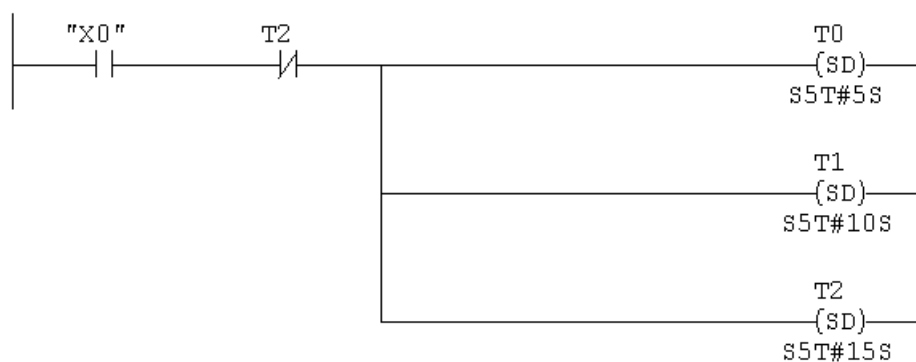
例三 顺序循环执行程序

当 X0 接通，灯 Y0 亮；经 5s 后，灯 Y0 灭，灯 Y1 亮；经 5s 后，灯 Y1 灭，灯 Y2 亮，再过 5s 后，灯 Y2 灭，灯 Y0 亮，如此顺序循环，其时序图如下。

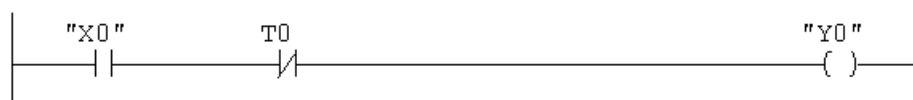


梯形图程序：

Network 1 :



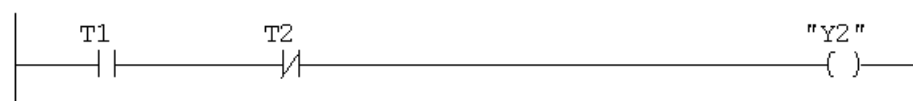
Network 2 :



Network 3 :



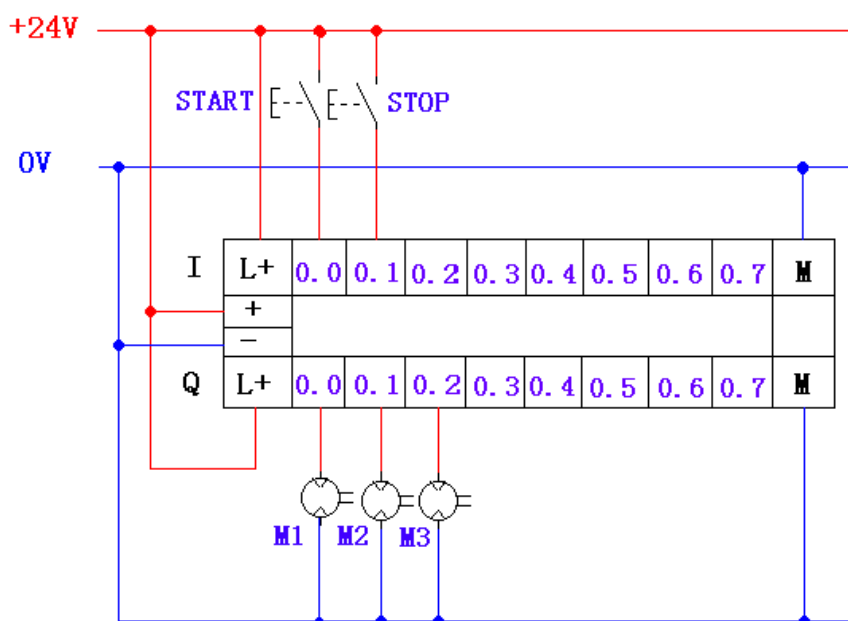
Network 4 :



例四 电动机顺序启动控制程序

有三台电动机 M1、M2、M3，按下启动按钮后 M1 启动，延时 5s 后 M2 启动，再延时 16s 后 M3 启动。

(一) PLC 接线

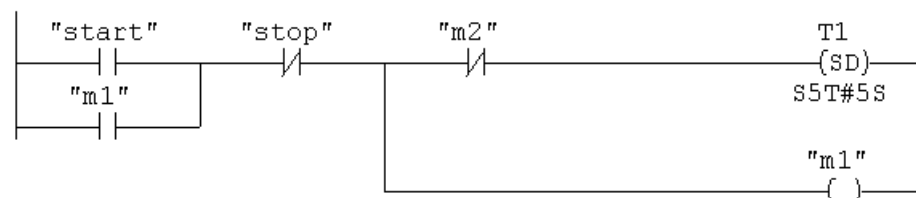


(二) 定义符号地址

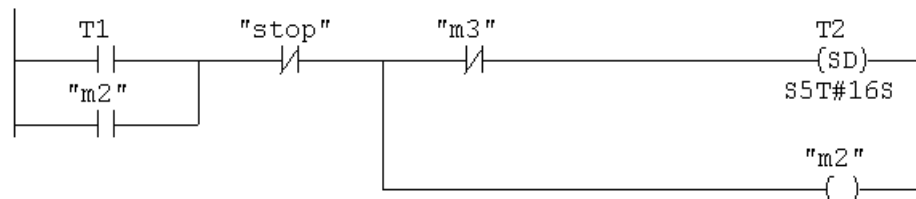
	Symbol	Address	Data Type
1	START	I 0.0	BOOL
2	STOP	I 0.1	BOOL
3	M1	Q 0.0	BOOL
4	M2	Q 0.1	BOOL
5	M3	Q 0.2	BOOL

(三) 梯形图程序

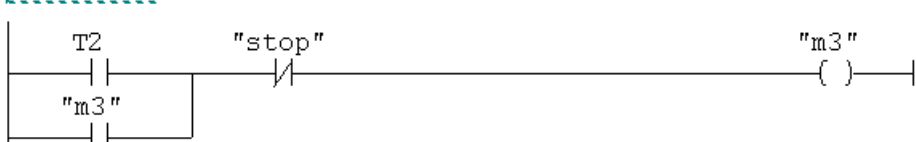
Network 1:



Network 2:

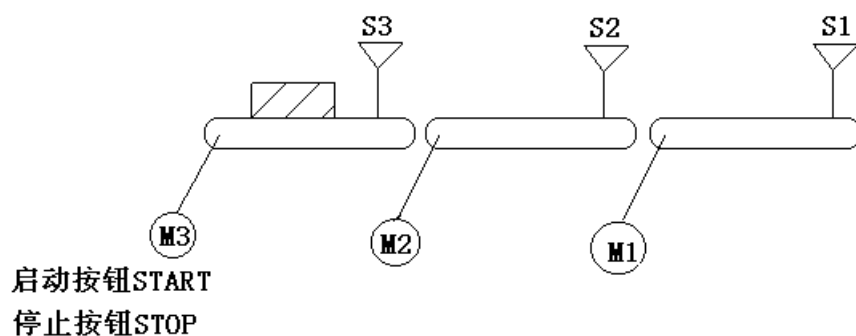


Network 3:

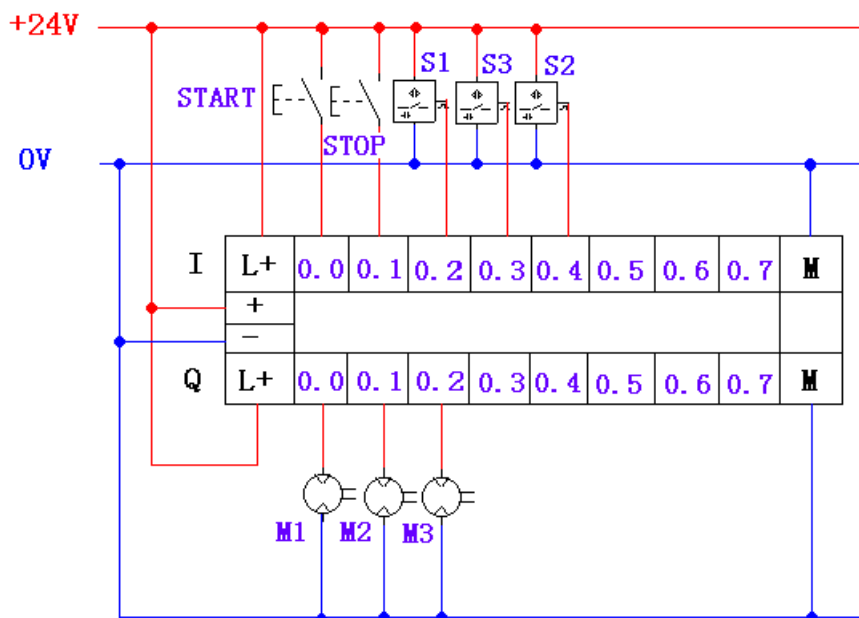


例五 分段传送带的电动机按制程序

为了节省能源的损耗，可使用 PLC 来启动和停止分段传送带的驱动电动机，使那些只载有物体的传送带运转，没有载物的传送带停止运行。金属板正在传送带上输送，其位置由相应的传感器检测。传感器安放在两段传送带相邻的地方，一旦金属板进入传感器的检测范围，PLC 便发出相应的输出信号，使后一段传送带的电动机投入工作；当金属板被送出检测范围时，PLC 内部定时器立即开始计时，在达到预定的延时时间后，前一段传送带电动机便停止运行。



(一) PLC 接线



(二) 定义符号地址

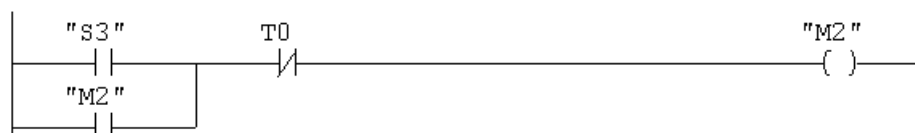
	Symbol	Address	Data Type
1	M1	Q 0.0	BOOL
2	M2	Q 0.1	BOOL
3	M3	Q 0.2	BOOL
4	START	I 0.0	BOOL
5	STOP	I 0.1	BOOL
6	S1	I 0.2	BOOL
7	S2	I 0.3	BOOL
8	S3	I 0.4	BOOL

(三) 梯形图程序

Network 1 :



Network 2 :



Network 3 :



Network 4 :



Network 5 :

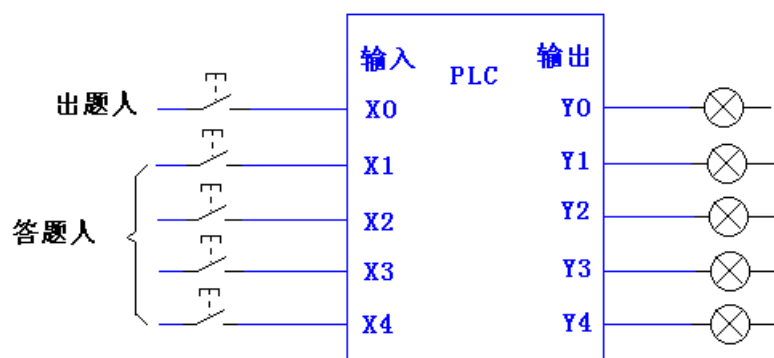


Network 6 :

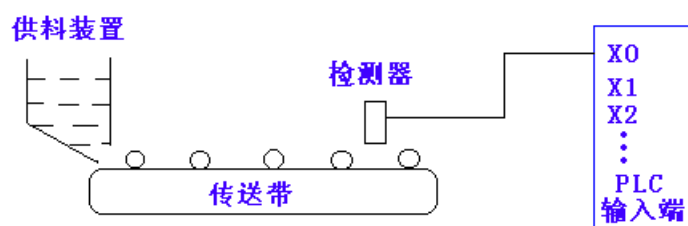


习题:

- (1) 有 4 个答题人，出题人提出问题，答题人按动按钮开关抢答，只有最早按动按钮的人有输出。出题人按复位按钮，引出下一个问题，试编写 PLC 梯形图程序。



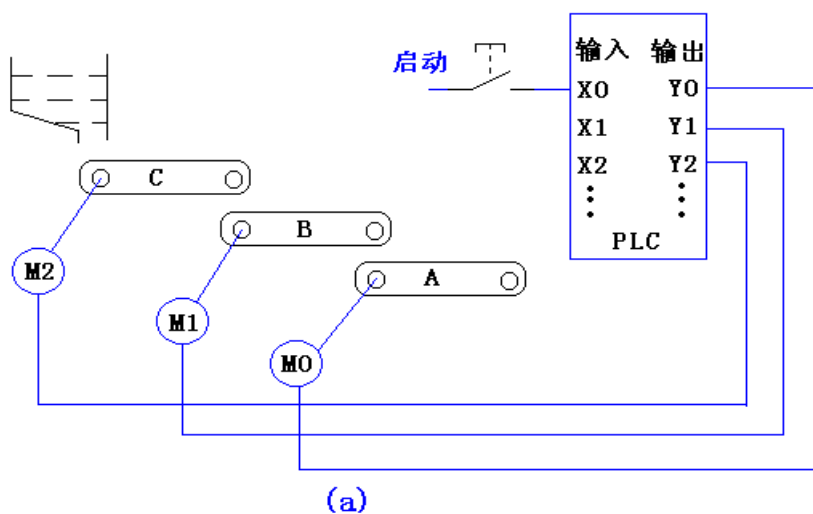
- (2) 若传送带上 30s 内无产品通过，检测器下的检测点则报警，试编写其梯形图程序。

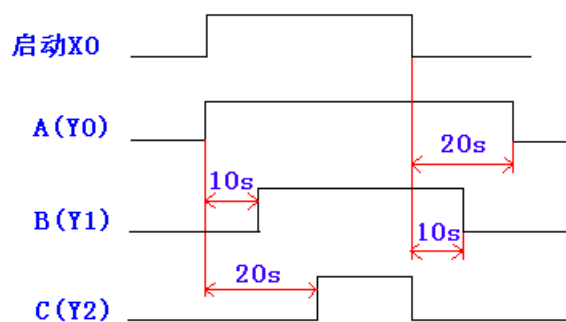


- (3) 试编写 PLC 梯形图程序具备下述功能：

- (a) 按钮接通后 A 灯先亮，过 5s 后 B 灯亮；
- (b) B 灯亮 5s 后，自动关闭；
- (c) B 灯灭 5s 后，A 灯关闭。

- (4) 如下图 (a) A, B, C 三条传输皮带，分别受 M1, M2, M3 三个电动机拖动；图 (b) 是此三条传输皮带运转的时序图。编写一个用 PLC 控制它们运转的梯形图程序。要求它们按 A—B—C 顺序启动，而后按 C—B—A 的顺序停止。

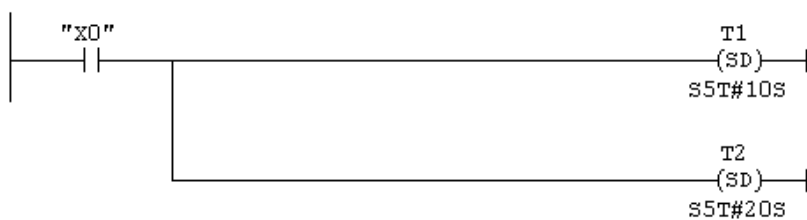




(b)

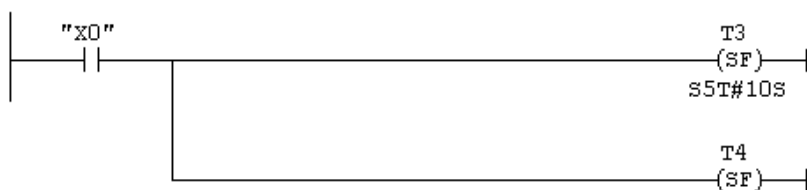
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:



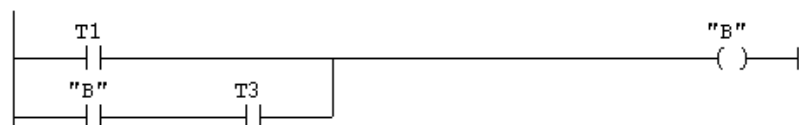
Network 3 : Title:

Comment:



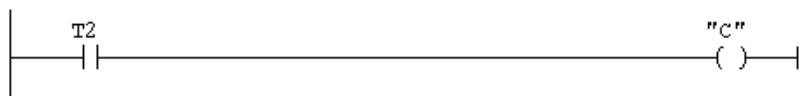
Network 4 : Title:

Comment:

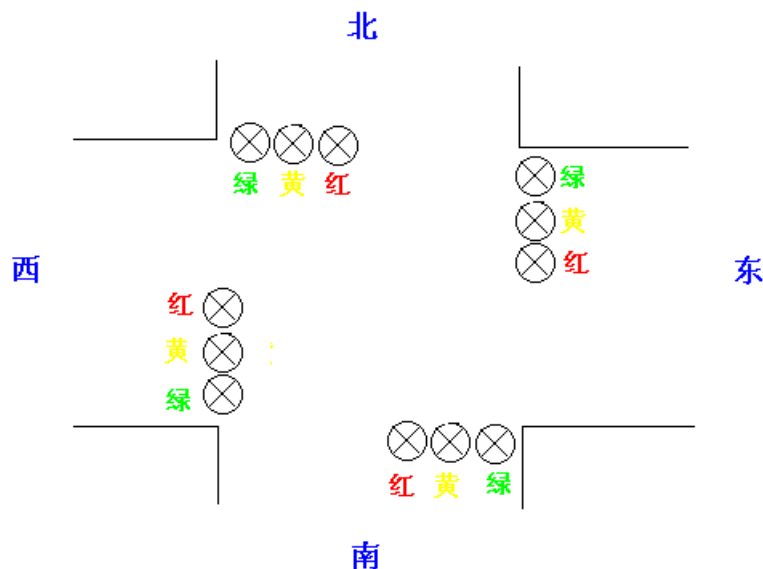


Network 5: Title:

Comment:

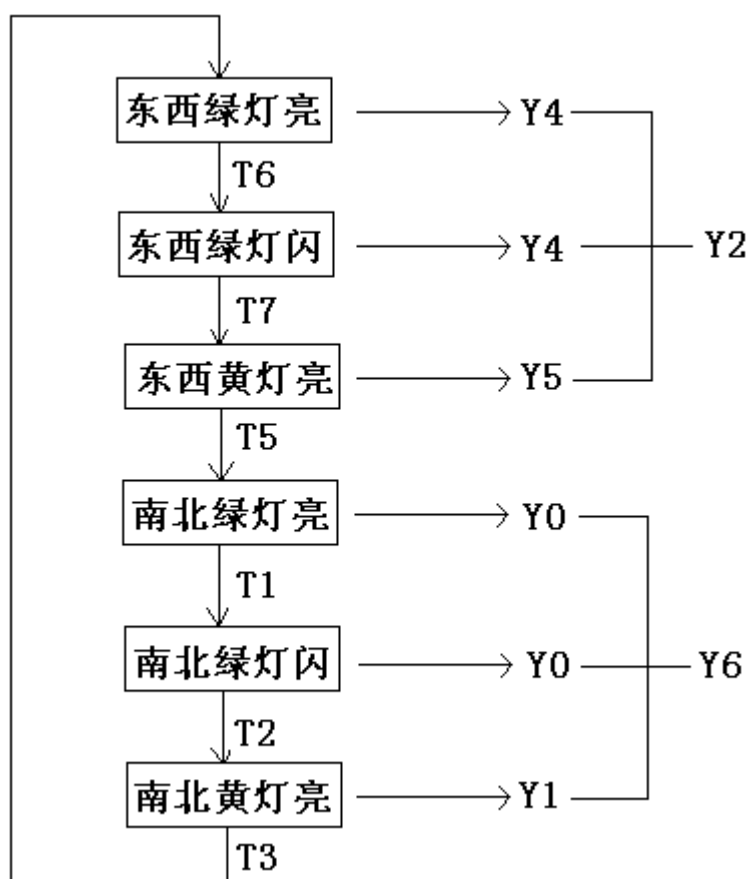
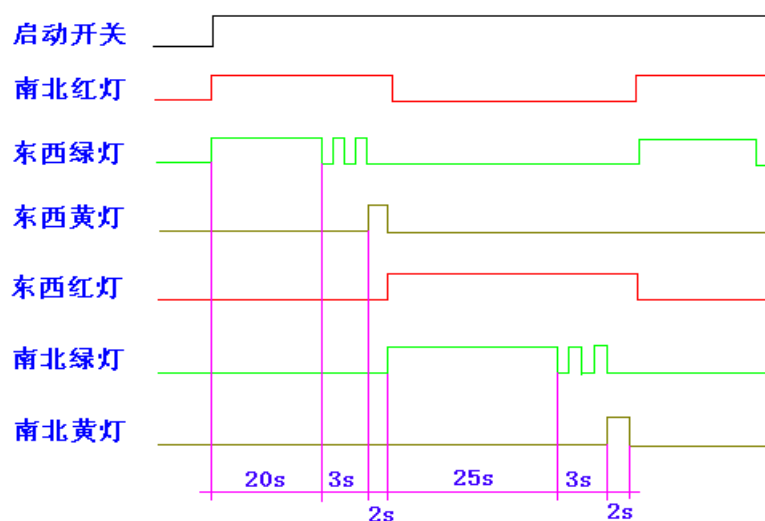


(5) 十字路口的交通指挥信号灯布置如下图:



一、控制要求

- (1) 信号灯系统由一个启动开关控制，当启动开关接通时，该信号灯系统开始工作，当启动开关关断时，所有信号灯都熄灭。
- (2) 南北绿灯和东西绿灯不能同时亮。如果同时亮应关闭信号灯系统，并立刻报警。
- (3) 南北红灯亮维持 25s。在南北红灯亮的同时东西绿灯也亮，并维持 20s。到 20s 时，东西绿灯闪亮，闪亮 3s 后熄灭，此时，东西黄灯亮，并维持 2s。到 2s 时，东西黄灯熄灭，东西红灯亮。同时，南北红灯熄灭，南北绿灯亮。
- (4) 东西红灯亮维持 30s。南北绿灯亮维持 25s，然后闪亮 3s 后熄灭。同时南北黄灯亮，维持 2s 后熄灭，这时南北红灯亮，东西绿灯亮。
- (5) 以上南北、东西信号灯周而复始地交替工作状态，指挥着十字路口的交通，其时序如下所示。

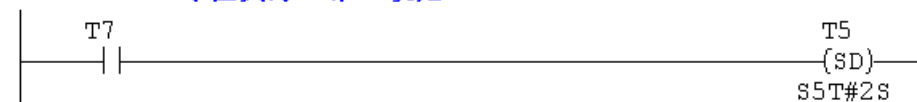


二、PLC 接线

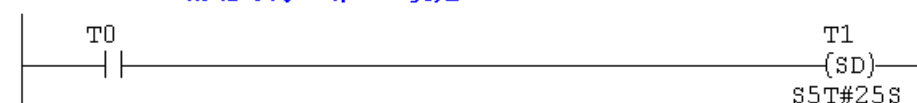
	Symbol	Address	Data Type
1	X0	I 0.0	BOOL
2	Y0	Q 0.0	BOOL
3	Y1	Q 0.1	BOOL
4	Y2	Q 0.2	BOOL
5	Y3	Q 0.3	BOOL
6	Y4	Q 0.4	BOOL
7	Y5	Q 0.5	BOOL
8	Y6	Q 0.6	BOOL

T6 (SD) T7
S5T#3S

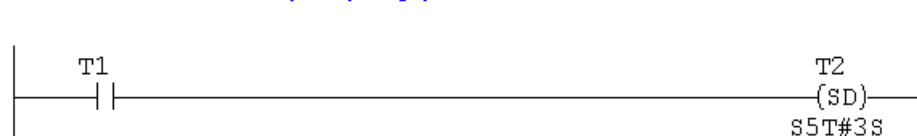
Network 5 : 东西黄灯工作2s 设定



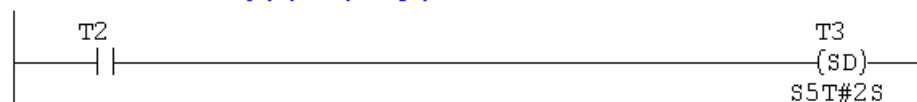
Network 6 : 南北绿灯工作25s 设定



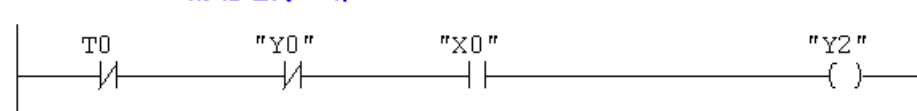
Network 7 : 南北绿灯闪烁3s 设定



Network 8 : 南北黄灯工作2s 设定



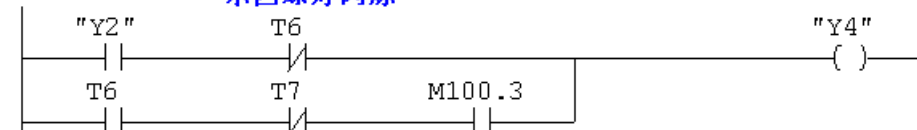
Network 9 : 南北红灯工作



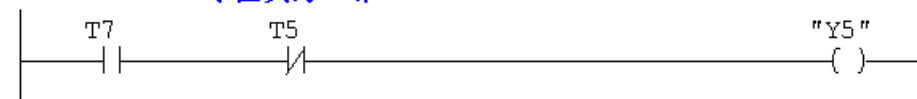
Network 10 : 东西红灯工作



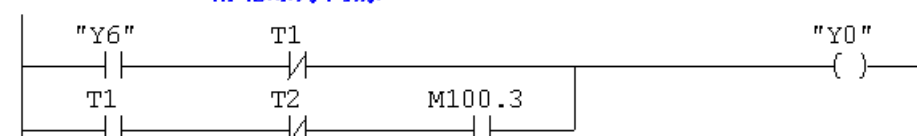
**Network 11 : 东西绿灯工作
东西绿灯闪烁**



Network 12 : 东西黄灯工作



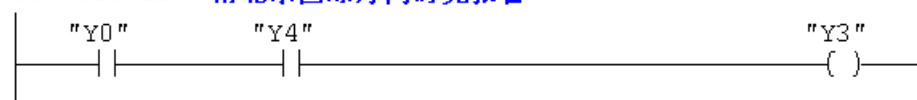
**Network 13 : 南北绿灯工作
南北绿灯闪烁**



Network 14 : 南北黄灯工作



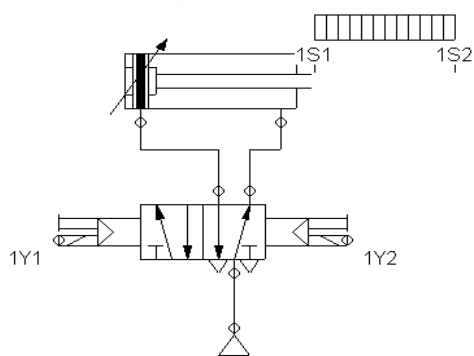
Network 15 : 南北东西绿灯同时亮报警



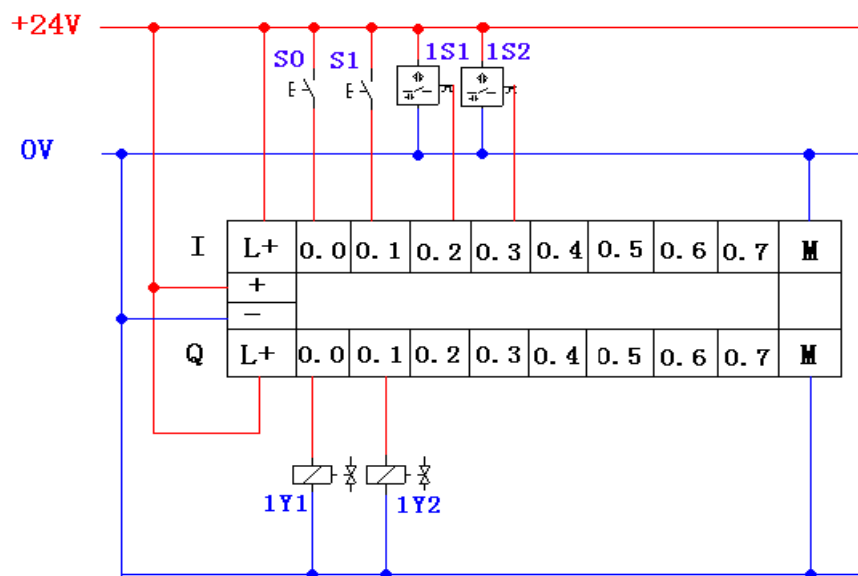
(6) PLC 控制的气缸延时控制回路，当气缸的活塞伸出到位停留 5s 后再返

回，退回到位停留 3s 再伸出，如此往复运动。

(一) 气控回路



(二) PLC 接线



(三) 定义符号地址

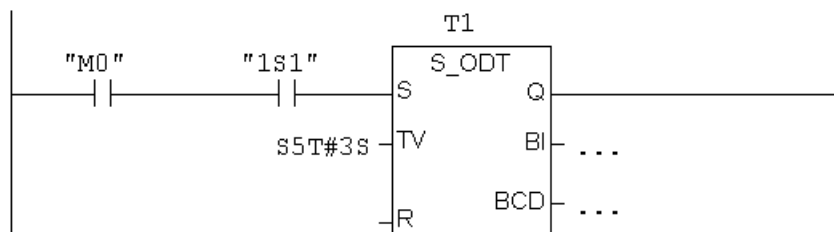
符号地址	绝对地址	数据类型	说明
S0	I0.0	BOOL	启动按钮
S1	I0.1	BOOL	停止按钮
1S1	I0.2	BOOL	位置传感器
1S2	I0.3	BOOL	位置传感器
1Y1	Q0.0	BOOL	换向阀电磁线圈
1Y2	Q0.1	BOOL	换向阀电磁线圈
M0	M0.0	BOOL	启动线圈

(四) 梯形图程序

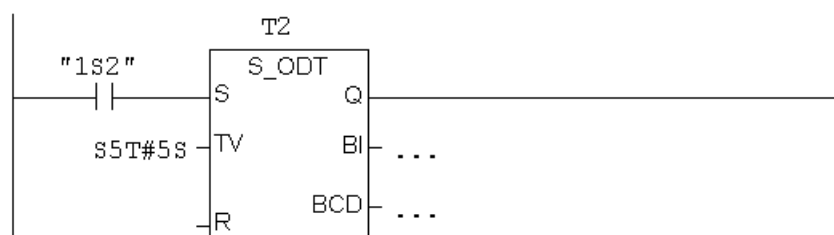
Network 1 :



Network 2 :



Network 3 :



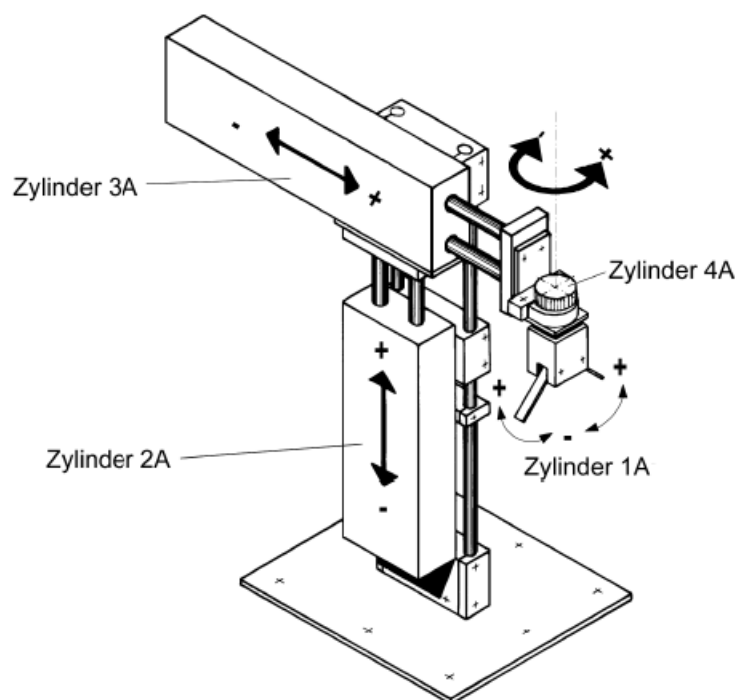
Network 4 :



Network 5 :



(7) 试编写以下机械手的 PLC 控制程序：



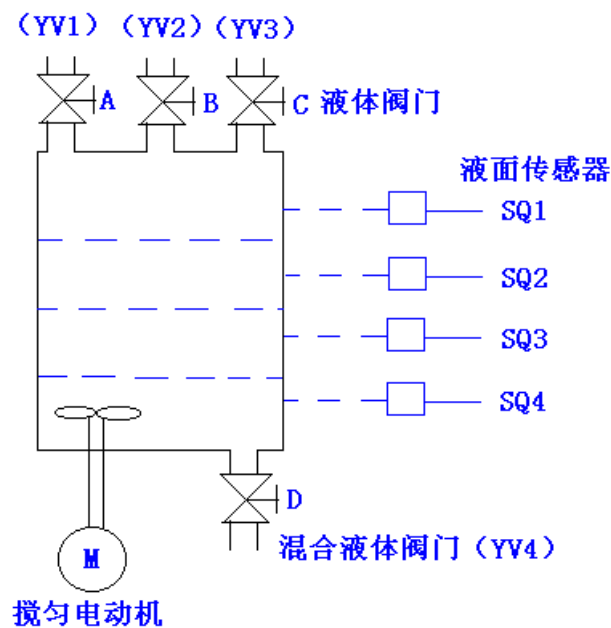
机械手

其动作的顺序如下，要求机械手在抓工件和松开工件时有 1s 的延时：

Arbeitszylinder	Bewegung	Aufgabe	Quittierung Signalgeber
Zylinder 1A	-	greifen	1S1
Zylinder 2A	+	Vertikalhub (heben)	2S2
Zylinder 3A	+	Horizontalhub (ausfahren)	3S2
Zylinder 4A	+	schwenken (nach links v. o.)	4S2
Zylinder 2A	-	Vertikalhub (absenken)	2S1
Zylinder 1A	+	greifen (loslassen)	1S2
Zylinder 2A	+	Vertikalhub (heben)	2S2
Zylinder 3A	-	Horizontalhub (rückfahren)	3S1
Zylinder 4A	-	schwenken (nach rechts v. o.)	4S1
Zylinder 2A	-	Vertikalhub (absenken)	2S1

(8) 多种液体自动混合装置的 PLC 控制

如图所示为三种液体混合装置，SQ1、SQ2、SQ3 和 SQ4 为液面传感器，液面淹没时接通，液体 A、B、C 与混合液阀由电磁阀 YV1、YV2、YV3、YV4 控制，M 为搅匀电动机，其控制要求如下：



1. 初始状态

装置投入运行时, 液体 A、B、C 阀门关闭, 混合液阀门打开 20s 将容器放空后关闭。

2. 起动作

按下启动按钮 SB1, 装置开始按下列给定规律运转：

- ①液体 A 阀门打开, 液体 A 流入容器。当液面达到 SQ3 时, SQ3 接通, 关闭液体 A 阀门, 打开液体 B 阀门。
- ②当液面达到 SQ2 时, 关闭液体 B 阀门, 打开液体 C 阀门。
- ③当液面达到 SQ1 时, 关闭液体 C 阀门, 搅匀电动机开始搅拌。
- ④搅匀电动机工作 1min 后停止搅动, 混合液体阀门打开, 开始放出

混合液体。

- ⑤当液面下降到 SQ4 时，SQ4 由接通变断开，再过 20s 后，容器放空，混合液阀门关闭，开始下一周期。

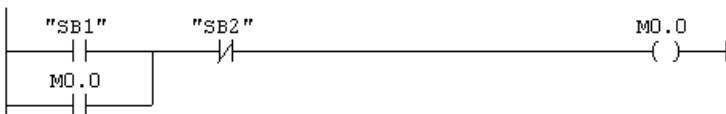
3. 停止操作

按下停止按钮 SB2 后，要将当前的混合操作处理完毕后，才停止操作（停在初始状态）

参考程序：

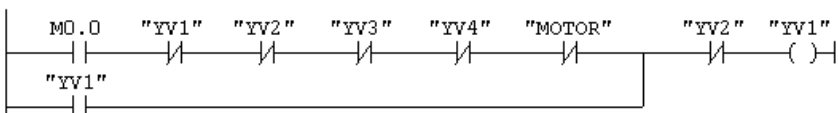
Network 1 : Title:

Comment:



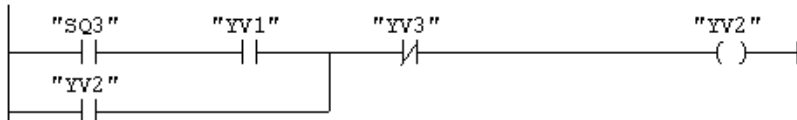
Network 2 : Title:

Comment:



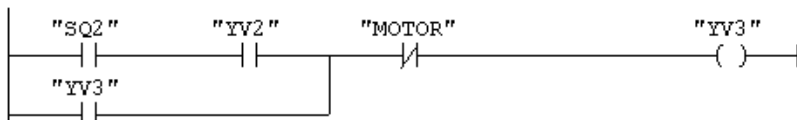
Network 3 : Title:

Comment:



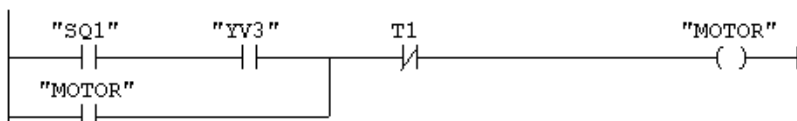
Network 4 : Title:

Comment:



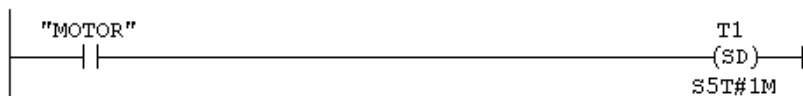
Network 5 : Title:

Comment:



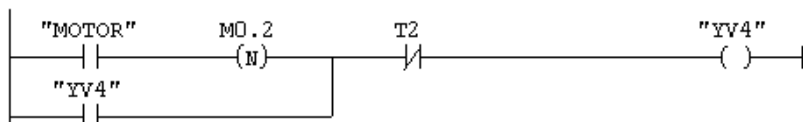
Network 6 : Title:

Comment:



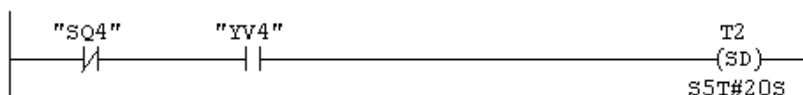
Network 7 : Title:

Comment:



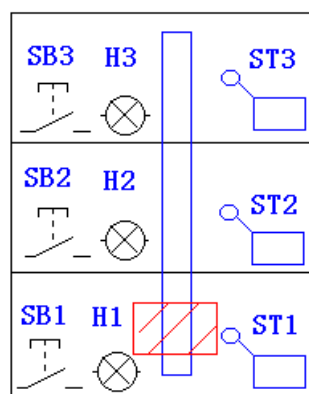
Network 8 : Title:

Comment:



(9) 如下

所示是三层楼电梯示意图。电梯的上升、下降由一台电动机控制；正转时电梯上升、反转时电梯下降。各层设一个呼叫开关（SB1、SB2、SB3）、一个呼叫指示灯（H1、H2、H3）、一个到位行程开关（ST1、ST2、ST3）。



三层楼电梯示意图

控制要求:

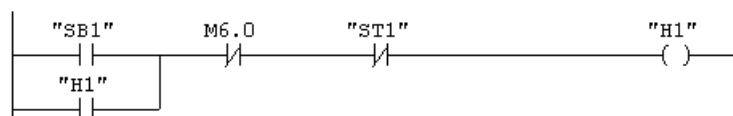
- (1) 各层的呼叫开关为按钮式开关，SB1、SB2 及 SB3 均为瞬间接通有效（即瞬间接通的即放开仍有效）。
- (2) 电梯箱体上升途中只响应上升呼叫，下降途中只响应下降呼叫，任何反方向呼叫均无效，简称为不可逆响应。具体动作要求，如下表。
- (3) 各楼层间有效运行时间应小于 10S，否则认为有故障、自动令电动机停转。

序号	输 入	输 出
----	-----	-----

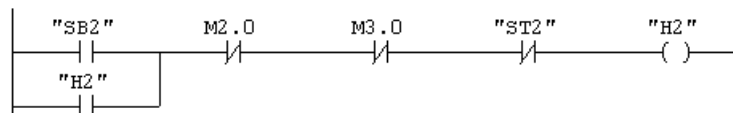
	原停层	呼叫层	运行方向	运行结果
1	1	3	升	上升到3层停,这期间经过2层时不停
2	2	3	升	上升到3层停
3	3	3	停	呼叫无效
4	1	2	升	上升到2层停
5	2	2	停	呼叫无效
6	3	2	降	下降到2层停
7	1	1	停	呼叫无效
8	2	1	降	下降到1层停
9	3	1	降	下降到1层停,这期间经过2层时不停
10	1	2、3	升	先升到2层暂停2S后,再升到3层停
11	2	1、3	降	下降到1层停
12	2	3、1	升	上升到3层停
13	3	2、1	降	先降到2层暂停2S后,再降到1层停
14	任意	任意	任意	楼层间运行时间必须小于10S,否则停

	Symbol	Address	Data Type	Comment
1	H1	Q 0.0	BOOL	1层呼叫指示灯
2	H2	Q 0.1	BOOL	2层呼叫指示灯
3	H3	Q 0.2	BOOL	3层呼叫指示灯
4	Motor-	Q 0.4	BOOL	电梯下降
5	Motor+	Q 0.3	BOOL	电梯上升
6	SB1	I 0.0	BOOL	1层呼叫开关
7	SB2	I 0.1	BOOL	2层呼叫开关
8	SB3	I 0.2	BOOL	3层呼叫开关
9	ST1	I 0.3	BOOL	1层到位开关
10	ST2	I 0.4	BOOL	2层到位开关
11	ST3	I 0.5	BOOL	3层到位开关

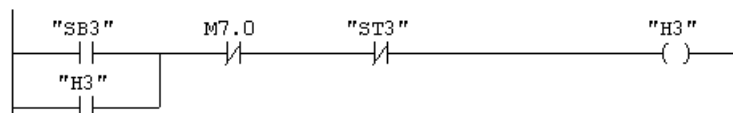
Network 1 : Title:



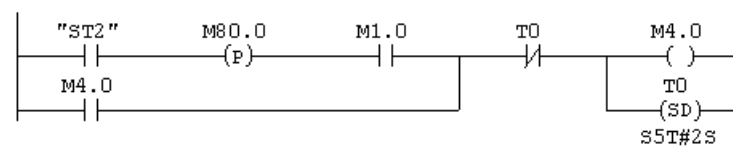
Network 2 : Title:



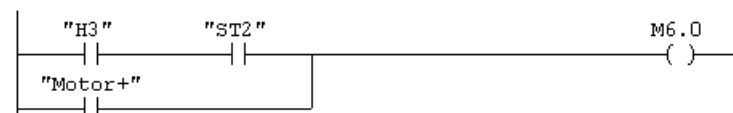
Network 3 : Title:



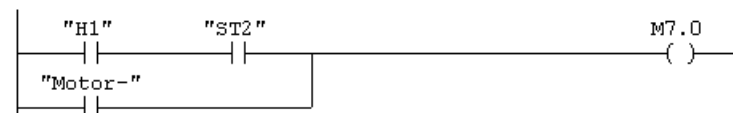
Network 4 : Title:



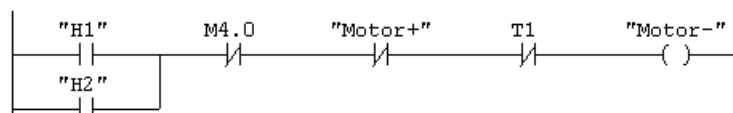
Network 5 : Title:



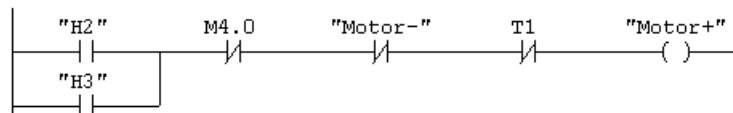
Network 6 : Title:



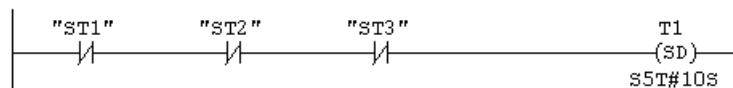
Network 7 : Title:



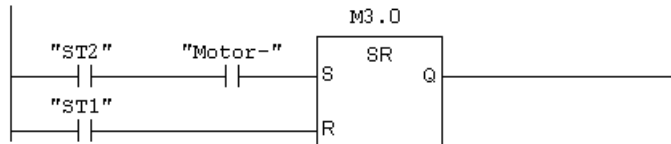
Network 8 : Title:



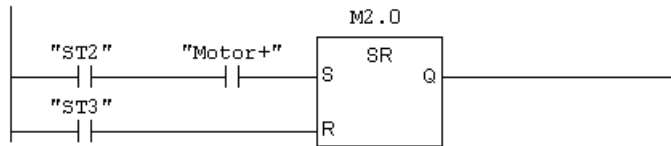
Network 9 : Title:



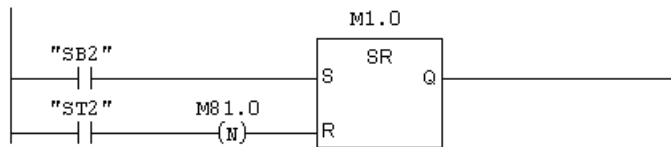
Network 10 : Title:



Network 11 : Title:

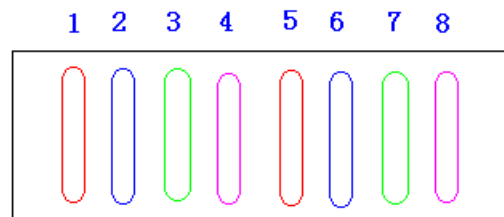


Network 12 : Title:



(10) 霓虹灯广告屏控制器的设计

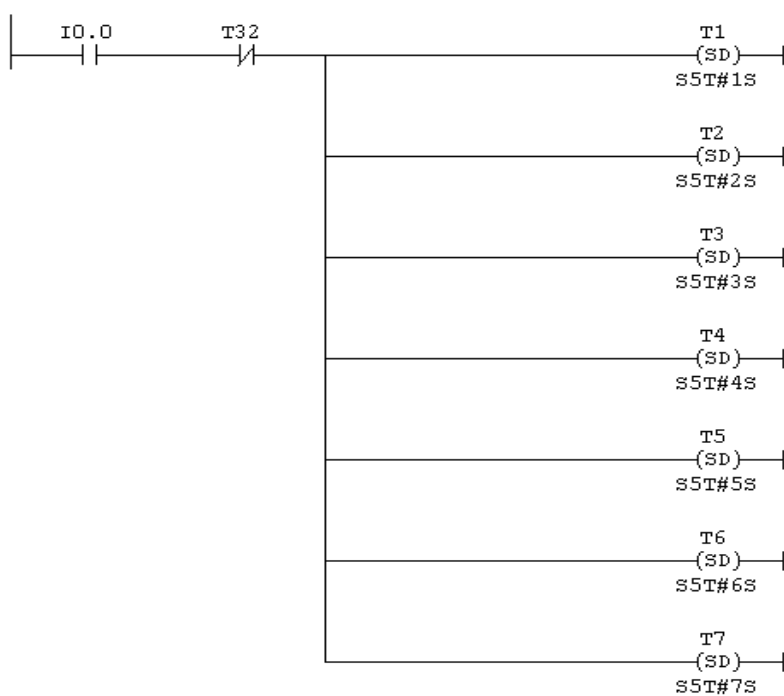
用 PLC 对霓虹灯广告屏实现控制，其具体要求如下：



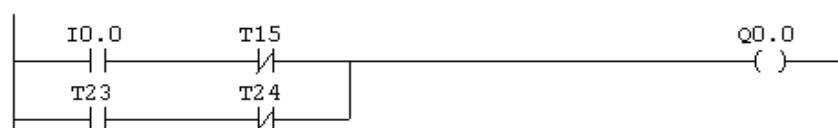
该广告屏中间 8 个灯管亮灭的时序为第 1 根亮→第 2 根亮→第 3 根亮→...→第 8 根亮，时间间隔为 1s，全亮后，显示 10s，再反过来从 8→7→...→1 顺序熄灭。全灭后，停亮 2s，再从第 8 根灯管开始亮起，顺序点亮 7→6→...→1，时间间隔为 1s，显示 20s，再从→2→...→8 顺序熄灭。全熄灭后，停亮 2s，再从头开始运行，周而复始。

参巧梯形图程序：

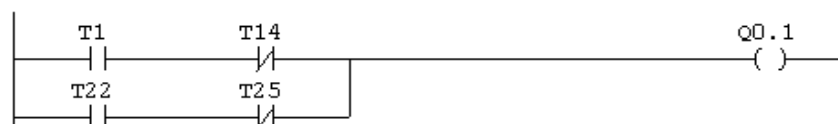
Network 1 : Title:



Network 2 : Title:



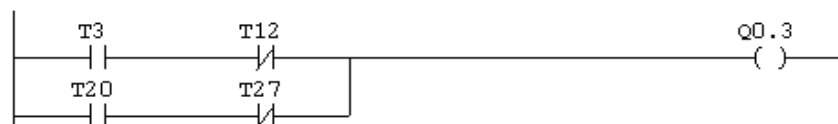
Network 3 : Title:



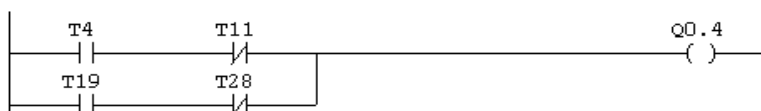
Network 4 : Title:



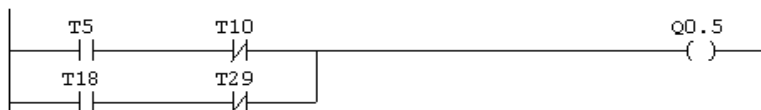
Network 5 : Title:



Network 6 : Title:



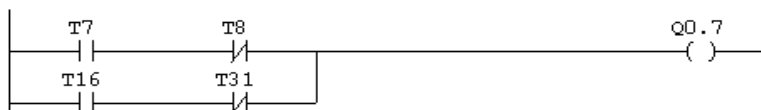
Network 7 : Title:



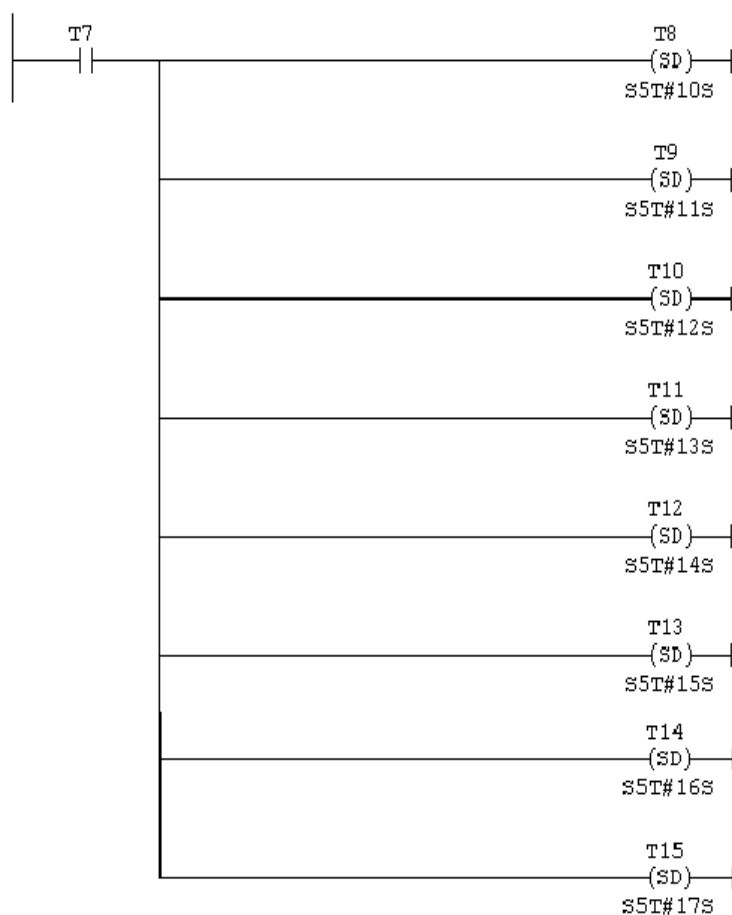
Network 8 : Title:



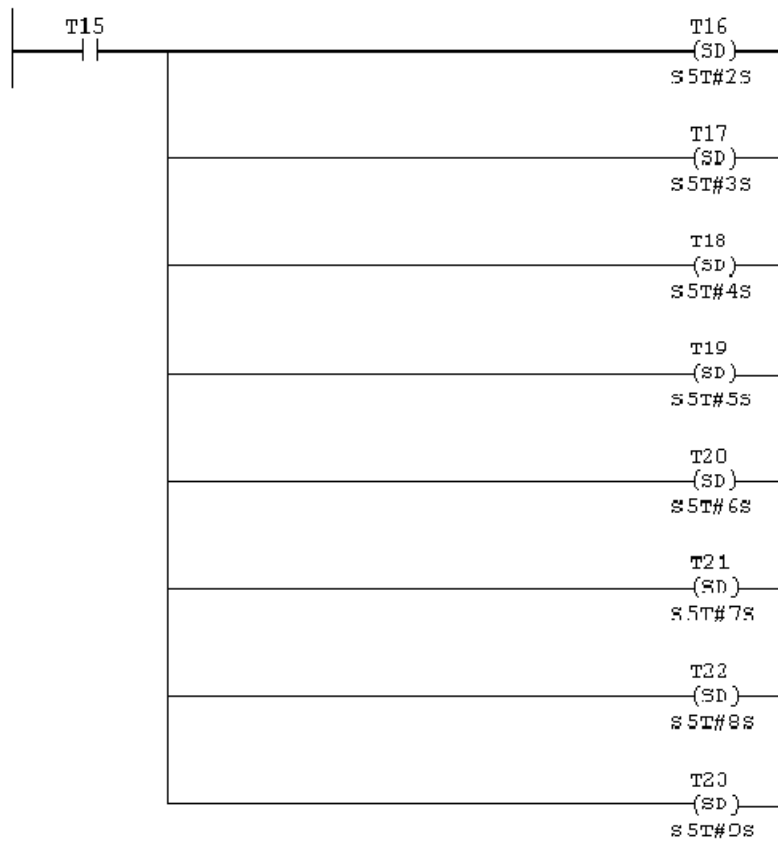
Network 9 : Title:



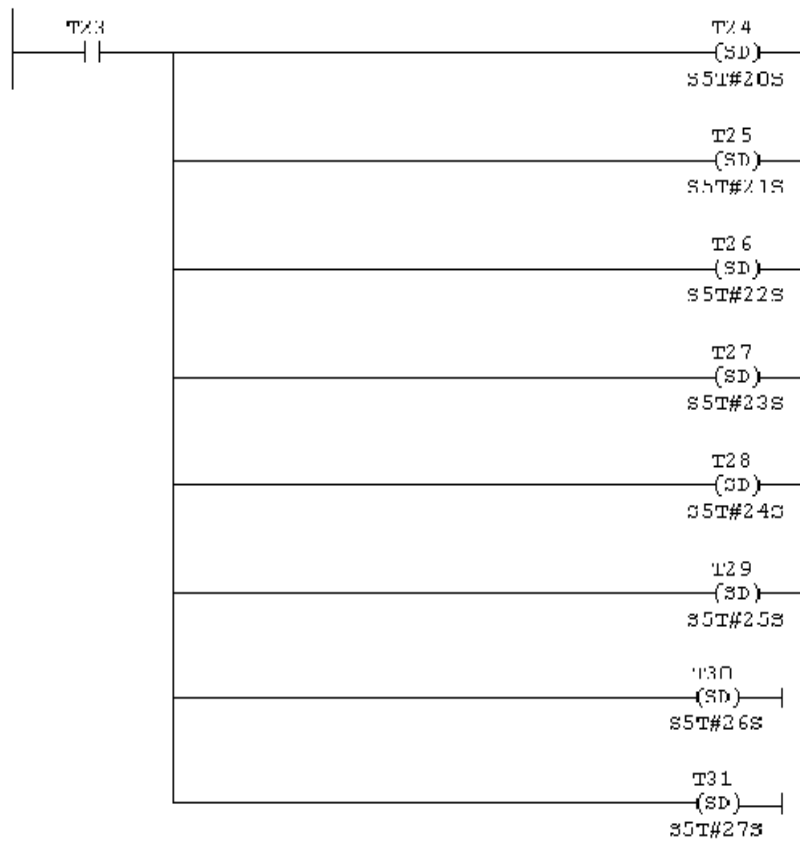
Network 10 : Title:



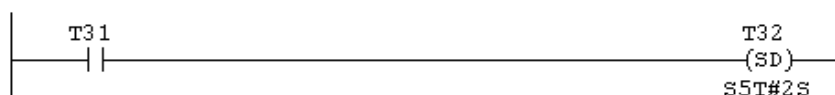
Network 11 : Title:



Network 12 : Title:

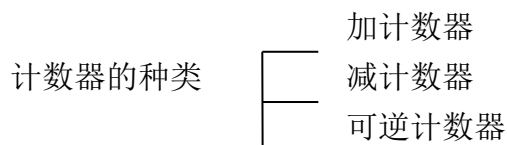


Network 13 : Title:



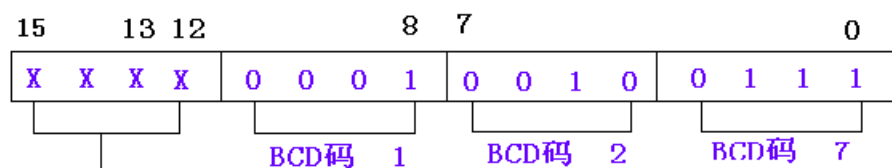
4.2.3 计数器指令

S7 中的计数器用于对 RLO 正跳沿计数。



1. 计数器的组成

在 CPU 中保留一块存储区作为计数器计数值存储区，每个计数器占用两个字节，称为计数器字。计数器字中的第 0 至 11 位表示计数值（二进制格式），计数范围是 0 到 999。当计数值达到上限 999 时，累加停止。计数值到达下限 0 时，将不再减小。



三位BCD码范围：0至999

无关：当计数器置数时这四位被忽略

累加器 1 低字的内容计数值 127

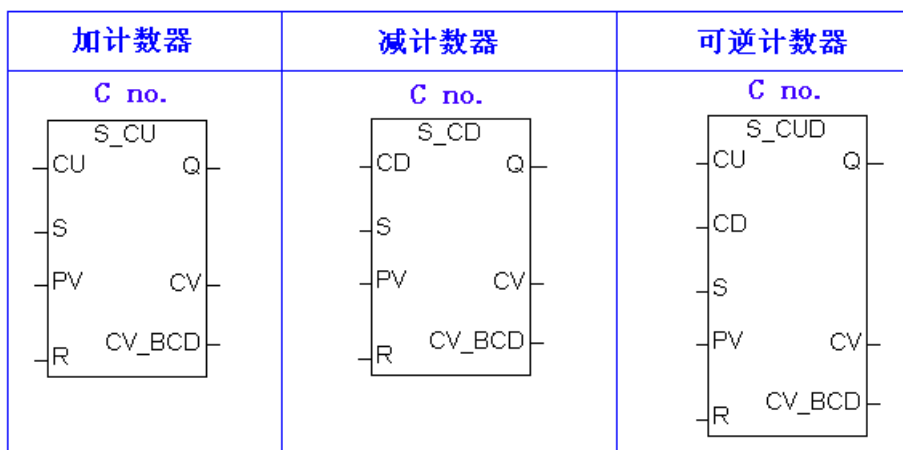
2. 计数器指令

LAD 指令	STL 指令	功能	说明
C no. — (SC) <预置值>	SC C no.	计数器置 初始值	该指令为计数器置初始值, 其中 no. 为计数器编号, 数据类型为 COUNTER; <预置值>的数据类型为 WORD, 可用存储区为 I、Q、M、D、L, 也可为常数; STL 指令的初始值在累加器 1 中.
C no. — (CU)	CU C no.	加计数	执行指令时, RLO 每有一个正跳沿计数值加 1. 若达上限 999, 则停止累加.
C no. — (CD)	CD C no.	减计数	执行指令时, RLO 每有一个正跳沿计数值减 1. 若达下限 0, 则停止减.
	FR C no.	允许计数 器再启动	若 RLO 为 1, 则初始值再次装入, 它不是计数器正常运行的必要条件.

当计数大于 0 时在输出 Q 上的 1 信号状态检查产生结果 1；当计数等

于 0 时，信号状态检查产生结果 0。

3. 计数器的梯形图方块指令



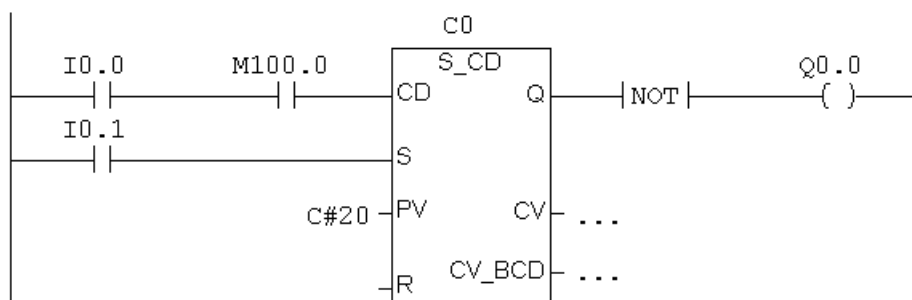
参数	数据类型	存储区	说明
NO.	COUNTER		计数器标识号
CU	BOOL	I, Q, M, D, L	加计数输入
CD	BOOL	I, Q, M, D, L	减计数输入
S	BOOL	I, Q, M, D, L	计数器预置输入
PV	WORD	I, Q, M, D, L	计数初始值 (0 至 999)
R	BOOL	I, Q, M, D, L	复位计数器输入
Q	BOOL	I, Q, M, D, L	计数器状态输出
CV	WORD	I, Q, M, D, L	当前计数值输出 (整数格式)
CV_BCD	WORD	I, Q, M, D, L	当前计数值输出 (BCD 格式)

4. 计数器应用举例

例一 计数器扩展为定时器

当定时器不够用时，可以用计数器扩展为定时器。程序中使用了 CPU 的时钟存储器，设置 MB100 为时钟存储器，则 M100.0 的变化周期为 0.1s。

Network 1 :

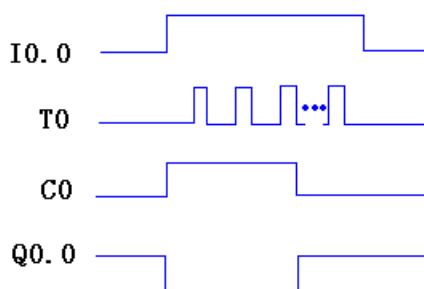


在程序中，如果 I0.1 的正跳沿为减计数器 C0 置数。若 I0.0 为 1，则 C0

每 0.1s 减 1。当 C0 减到 0 后, 输出 Q0.0 为 1. I0.1 的又一个正跳沿使 C0 置数并使输出为 0. 这样, 在 I0.0 为 1 后 2s ($20 \times 0.1\text{s} = 2\text{s}$), Q0.0 为 1, I0.1 的正跳沿使 Q0.0 复位.

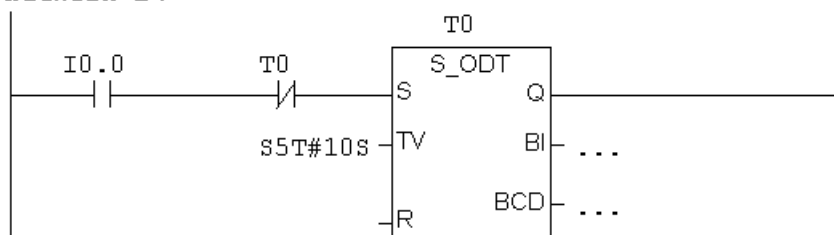
例二 长时间延时程序

采用定时器和计数器可以组成长时间延时程序, 如下:

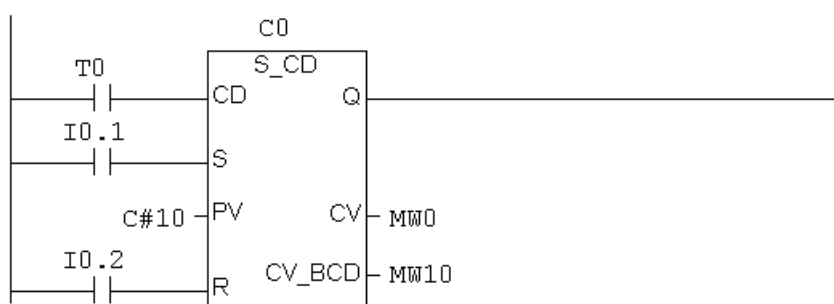


时序图

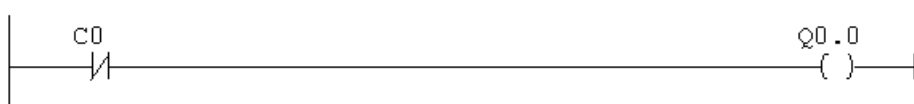
Network 1:



Network 2:



Network 3:

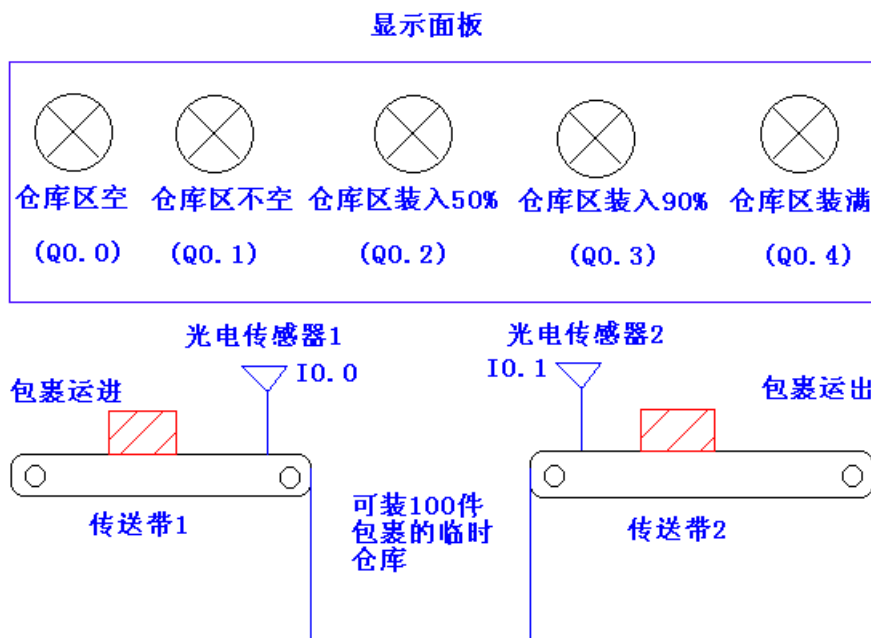


当输入 I0.0 接通时, 定时器 T0 经过 10s 时间延时后, 其动合触点 T0 闭合, 计数器 C0 开始递减运算, 与此同时 T0 的动断触点是断开的, 造成 T0 线圈断电, 使 T0 的动合触点断开, C0 仅计数一次, 而后 T0 线圈又接通, 如此循环. 当 C0 经过 $10\text{s} \times 10 = 100\text{s}$ 时间后, 计数器 C0 输出为 0, 输出 Q0.0 接通, 具有长时间延时的功能.

例三 货仓区的控制

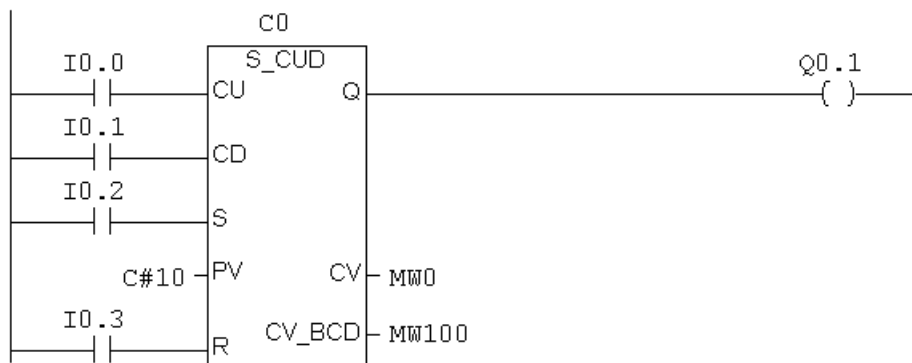
如下图, 装有两台传送带的系统, 在两台传送带之间有一个仓库区。传送带 1

将包裹运送至临时仓库区。传送带 1 靠近仓库区一端安装的光电传感器确定已有多少包裹运送至仓库区。传送带 2 将临时库区中的包裹运送至装货场，在这里货物由卡车运送至顾客。传送带 2 靠近仓库区一端安装的光电传感器确定已有多少包裹从库区运送至装货场。



梯形图程序：

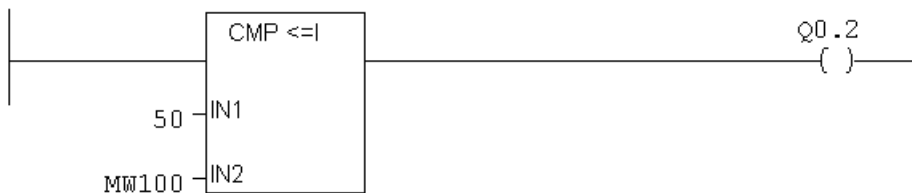
Network 1 :



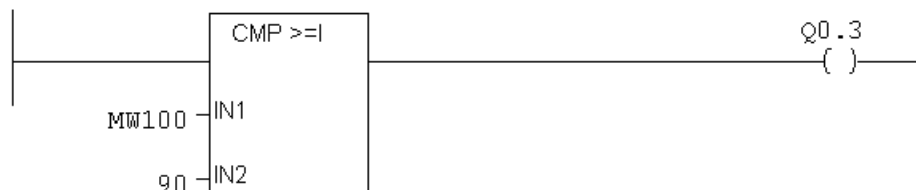
Network 2 :



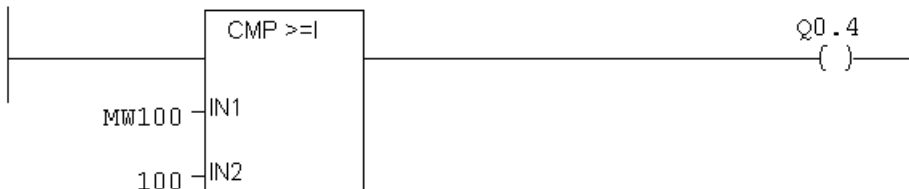
Network 3 :



Network 4 :



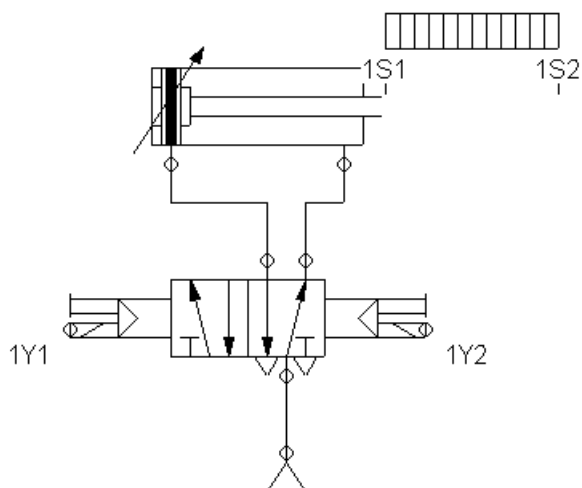
Network 5 :



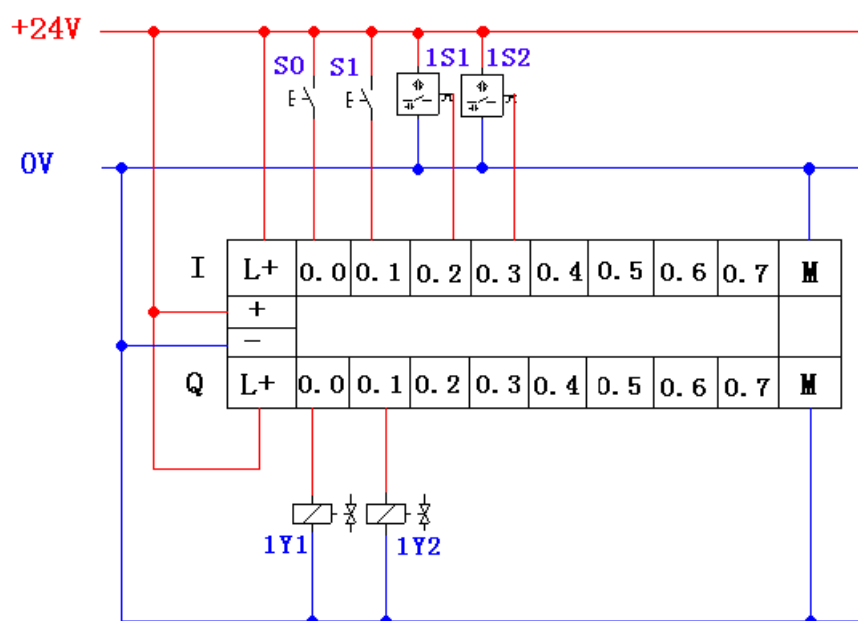
例四 气缸运动计数控制

要求：气缸连续往复运动 20 次便自动停止

(一) 气控回路



(二) PLC 接线



(三) 定义符号地址

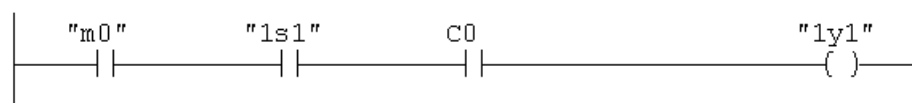
符号地址	绝对地址	数据类型	说明
S0	I0.0	BOOL	启动按钮
S1	I0.1	BOOL	停止按钮
1S1	I0.2	BOOL	位置传感器
1S2	I0.3	BOOL	位置传感器
1Y1	Q0.0	BOOL	换向阀电磁线圈
1Y2	Q0.1	BOOL	换向阀电磁线圈
M0	M0.0	BOOL	启动线圈

(四) 梯形图程序

Network 1 :



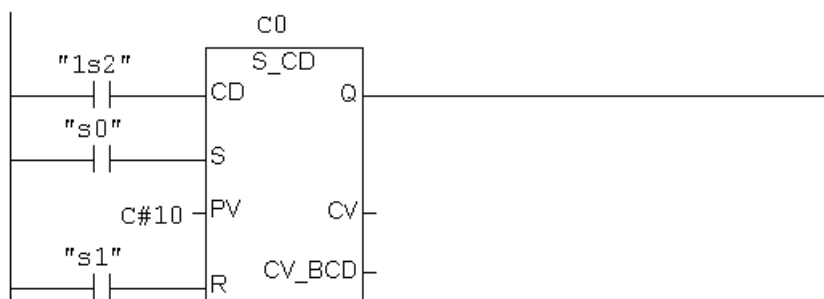
Network 2 :



Network 3 :

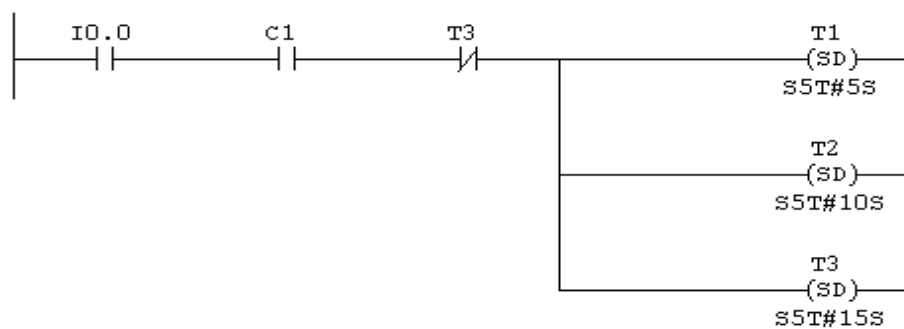


Network 4 :

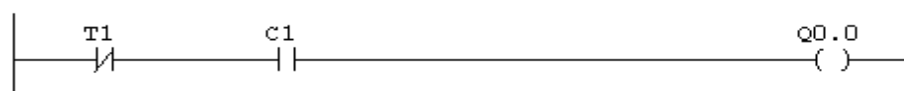


练习：当 X0 接通，灯 Y0 亮；经 5s 后，灯 Y0 灭，灯 Y1 亮；经 5s 后，灯 Y1 灭，灯 Y2 亮，再过 5s 后，灯 Y2 灭，灯 Y0 亮，如此顺序循环 10 次后自动停止。

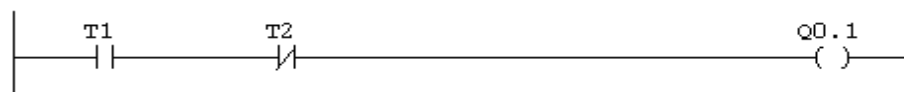
Network 1 : Title:



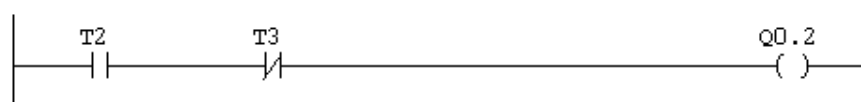
Network 2 : Title:



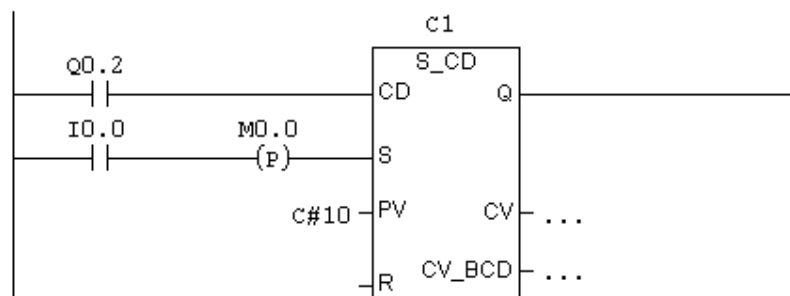
Network 3 : Title:



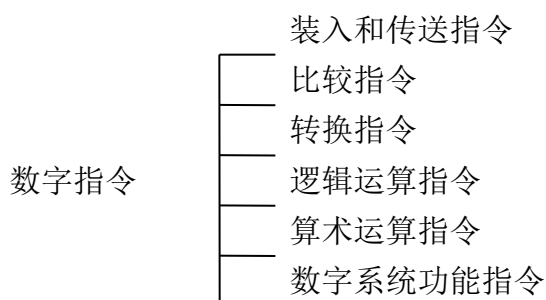
Network 4 : Title:



Network 5 : Title:



4.3 数字指令



4.3.1 装入和传送指令

装入(L)和传送(T)指令可以在存储区之间或存储区与过程输入、输出之间交换数据。CPU 执行这些指令不受逻辑操作结果 RLO 的影响。L 指令将源操作数装入累加器 1 中，而累加器原有的数据移入累加器 2 中，累加器 2 中原有的内容被覆盖。T 指令将累加器 1 中的内容写入目的存储区中，累加器的内容保持不变。L 和 T 指令可对字节、字、双字数据进行操作，当数据长度小于 32 位时，数据在累加器右对齐，其余各位填 0。

1. 对累加器 1 的装入和传送指令

L +5

T MW0

2. 读取或传送状态字

L STW //将状态字中 0-8 位装入累加器 1 中, 累加器 9-31 位被清 0

T STW //装累加器 1 中的内容传送到状态字中

3. 装入时间值或计数值

L T1 //将定时器 T1 中二进制格式的时间值直接装入累加器 1 的低字中

LC T1 //将定时器 T1 中的时间值和时基以 BCD 格式装入累加器 1 的低字中

L C1 //将计数器 C1 中二进制格式的计数值直接装入累加器 1 的低字中

LC C1 //将计数器 C1 中的计数值以 BCD 格式装入累加器 1 的低字中

4. 地址寄存器装入和传送

LAR1 //将操作数的内容装入地址寄存器 AR1

LAR2 //将操作数的内容装入地址寄存器 AR2

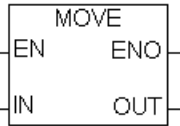
TAR1 //将 AR1 的内容传送给存储区或 AR2

TAR2 //将 AR2 的内容传送给存储区

CAR //交换 AR1 和 AR2 的内容

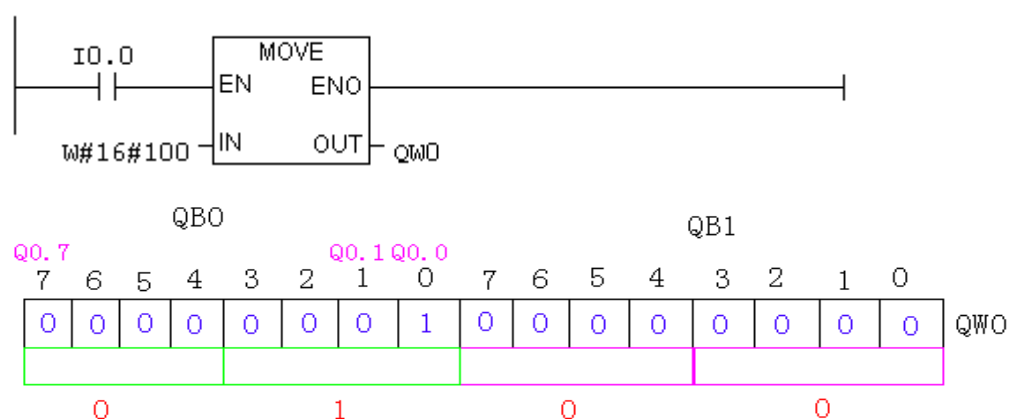
对于地址寄存器, 可以不经过累加器 1 而直接将操作数装入或传出, 或将两个地址寄存器的内容直接交换。

5. 梯形图方块传送指令

LAD指令	参数	数据类型	存储区	说明
	EN	BOOL	I、Q M、D L	允许输入
	ENO	BOOL		允许输出
	IN	8, 16, 32位长的所有数据类型		源数据, 可为常数
	OUT	8, 16, 32位长的所有数据类型		目的操作数

注意: 高位赋值给低字节, 低位赋值给高字节。

如下程序, 当按下按钮 I0.0, 输出 Q0.0 为 1.



4.3.2 比较指令

比较指令用于比较累加器 2 与累加器 1 中的数据大小。比较时应确保两个数的类型相同, 数据类型可以是整数、长整数或实数。若比较的结果为真, 则 RLO 为 1, 否则为 0。

1. 比较两个整数或长整数

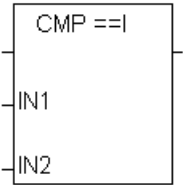
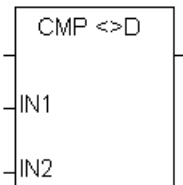
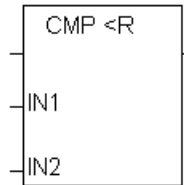
指令	说明
==I ==D	在累加器 2 低字中的整数是否等于累加器 1 低字中的整数 在累加器 2 中的长整数是否等于累加器 1 中的长整数
<>I <>D	在累加器 2 低字中的整数是否不等于累加器 1 低字中的整数 在累加器 2 中的长整数是否不等于累加器 1 中的长整数
>I >D	在累加器 2 低字中的整数是否大于累加器 1 低字中的整数 在累加器 2 中的长整数是否大于累加器 1 中的长整数
<I <D	在累加器 2 低字中的整数是否小于累加器 1 低字中的整数 在累加器 2 中的长整数是否小于累加器 1 中的长整数

$\geq I$ $\geq D$	在累加器 2 低字中的整数是否大于等于累加器 1 低字中的整数 在累加器 2 中的长整数是否大于等于累加器 1 中的长整数
$\leq I$ $\leq D$	在累加器 2 低字中的整数是否小于累加器 1 低字中的整数 在累加器 2 中的长整数是否小于累加器 1 中的长整数

2. 比较两个实数

指令	说明
$==R$	在累加器 2 中的 32 位实数是否等于累加器 1 中的实数
$<>R$	在累加器 2 中的 32 位实数是否不等于累加器 1 中的实数
$>R$	在累加器 2 中的 32 位实数是否大于累加器 1 中的实数
$<R$	在累加器 2 中的 32 位实数是否小于累加器 1 中的实数
$\geq R$	在累加器 2 中的 32 位实数是否大于等于累加器 1 中的实数
$\leq R$	在累加器 2 中的 32 位实数是否小于等于累加器 1 中的实数

3. 梯形图方块比较指令

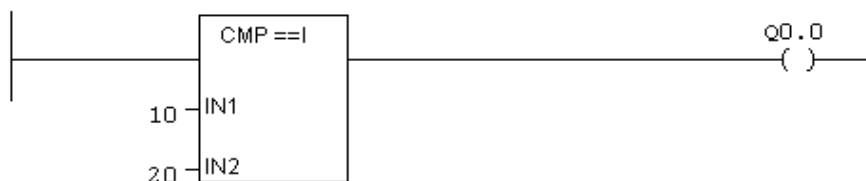
整数比较举例	长整数比较举例	实数比较举例
 等于	 不等于	 小于

参数	数据类型	存储区	说明
INT1	INT DINT REAL	I、Q、M、D、L	第一个参与比较数
INT2	INT DINT REAL	I、Q、M、D、L	第二个参与比较数

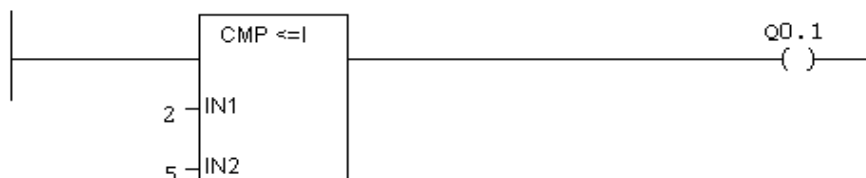
方块比较指令在逻辑串中，等效于一个常开触点，如果比较结果为“真”，则该常开触点闭合，否则触点断开。

例如：

Network 1 : Title:



Network 2 : Title:



4.3.3 算术运算指令

在 STEP 7 中可以对整数、长整数和实数进行加、减、乘、除算术运算。算术运算指令在累加器 1 和 2 中进行，在累加器 2 中的值作为被减数或被除数。算术运算的结果存在累加器 1 中，累加器 1 原有的值被运算结果覆盖，累加器 2 中的值保持不变。

1. 整数算术运算

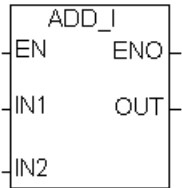
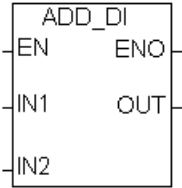
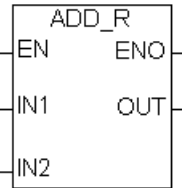
指令	说明
+I	将累加器 1, 2 低字中的 16 位整数相加, 16 位整数结果保存在累加器 1 低字中
-I	将累加器 2 低字中的 16 位整数减去累加器 1 低字中的内容, 结果保存在累加器 1 低字中
*I	将累加器 1, 2 低字中的 16 位整数相乘, 32 位整数结果保存在累加器 1 低字中
/I	将累加器 2 低字中的 16 位整数除以累加器 1 低字中的内容, 商为 16 位整数并保存在累加器 1 低字中, 余数存放在累加器 1 的高字中
+D	将累加器 1, 2 中的 32 位整数相加, 32 位整数结果保存在累加器 1 中

—D	将累加器 2 中的 32 位整数减去累加器 1 中的内容, 结果保存在累加器 1 中
*D	将累加器 1, 2 中的 32 位整数相乘, 32 位整数结果保存在累加器 1 中
/D	将累加器 2 中的 32 位整数除以累加器 1 中的内容, 商为 32 位整数并保存在累加器 1 中, 余数被丢掉
MOD	将累加器 2 中的 32 位整数除以累加器 1 中的内容, 余数保存在累加器 1 中, 商被丢掉
+	累加器 1 中加一个 16 位或 32 位整数常量, 结果保存在累加器 1 中

2. 实数算术运算

指令	说明
+R	将累加器 1, 2 中的 32 位实数相加, 32 位结果保存在累加器 1 中
—R	将累加器 2 中的 32 位实数减去累加器 1 中的内容, 结果保存在累加器 1 中
*R	将累加器 1, 2 中的 32 位实数相乘, 32 位乘积保存在累加器 1 中
/R	将累加器 2 中的 32 位实数除以累加器 1 中的内容, 商为 32 位实数并保存在累加器 1 中
ABS	对累加器 1 中的 32 位实数取绝对值

3. 梯形图算术运算方块指令

整数相加	长整数相加	实数相加
		

例一 解数学问题

用算术运算指令解以下的方程：

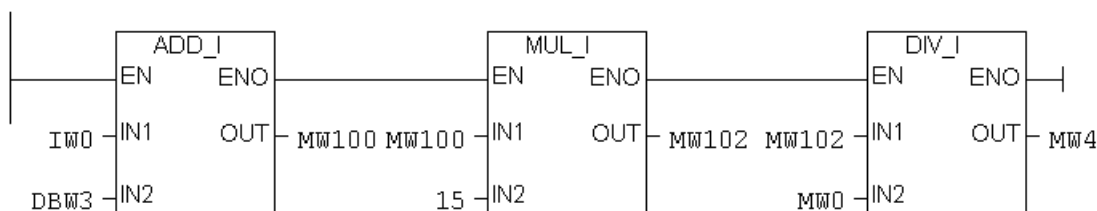
$$MW4 = \frac{(IW0 + DBW3) \times 15}{MW0}$$

梯形图程序：

Network 1:

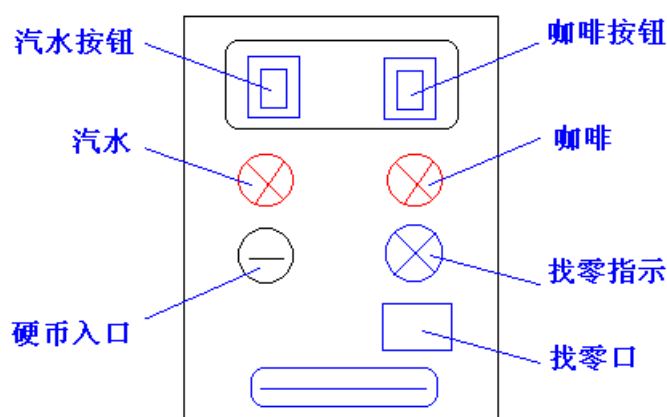


Network 2:



练习：(1) 自动售货机的 PLC 控制

如下图所示的自动售货机示意图，其工作要求如下：

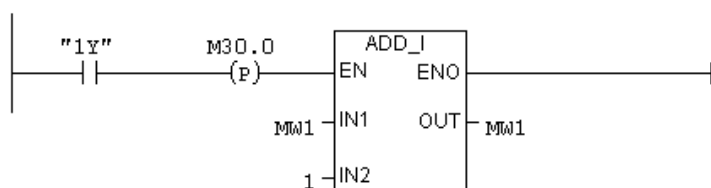


1. 此售货机可投入 1 元、5 元或 10 元硬币。
2. 当投入的硬币总值超过 12 元时，汽水按钮指示灯亮；当投入的硬币总值超过 15 元时，汽水及咖啡按钮指示灯都亮。
3. 当汽水按钮灯亮时，按汽水按钮，则汽水排出 7s 后自动停止，这段时间内，汽水指示灯闪动。
4. 当咖啡按钮灯亮时，按咖啡按钮，则咖啡排出 7s 后自动停止，这段时间内，咖啡指示灯闪动。
5. 若投入硬币总值超过按钮所需的钱数(汽水 12 元，咖啡 15 元)时，找钱指示灯亮, 表示找钱动作, 并退出多余的钱。

参考答案：

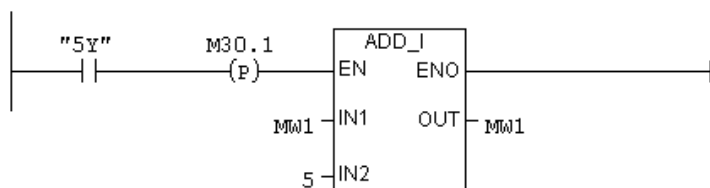
Network 1: Title:

Comment:



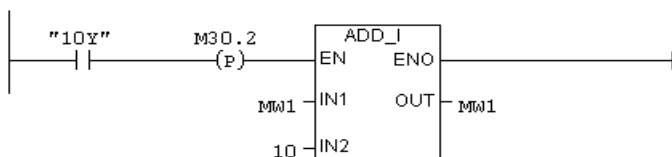
Network 2 : Title:

Comment:



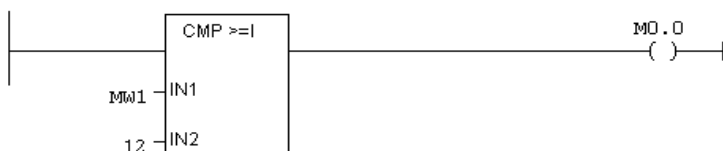
Network 3 : Title:

Comment:



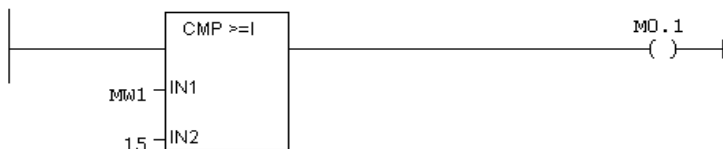
Network 4 : Title:

Comment:



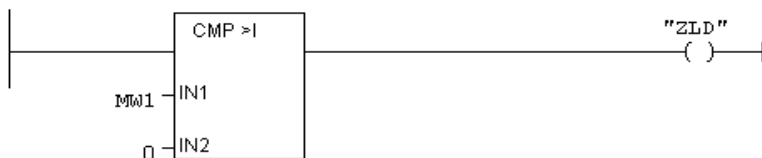
Network 5 : Title:

Comment:



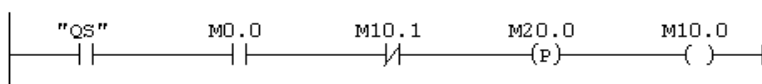
Network 6 : Title:

Comment:



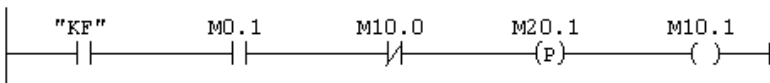
Network 7 : Title:

Comment:



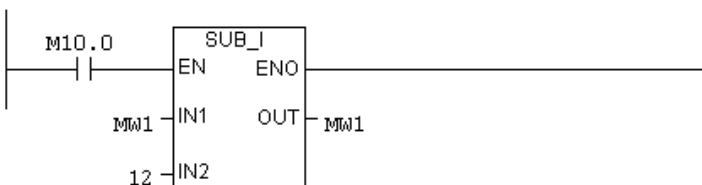
Network 8 : Title:

Comment:



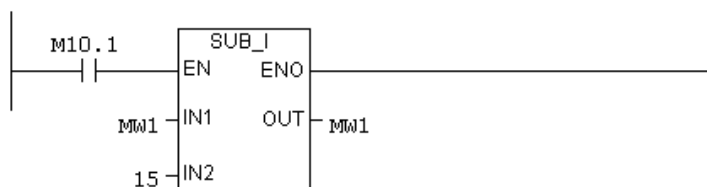
Network 9 : Title:

Comment:



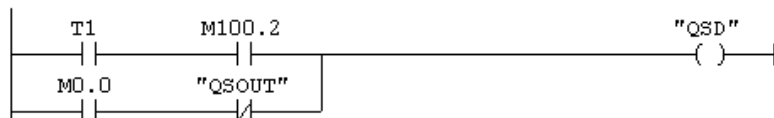
Network 10 : Title:

Comment:



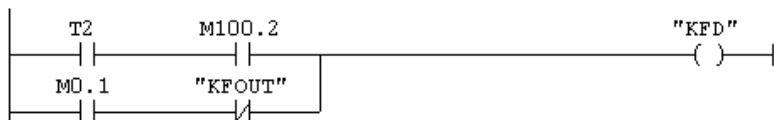
Network 11 : Title:

Comment:



Network 12 : Title:

Comment:



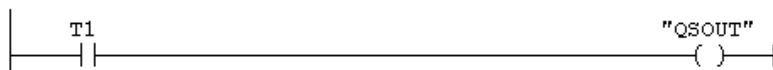
Network 13 : Title:

Comment:



Network 14 : Title:

Comment:



Network 15 : Title:

Comment:



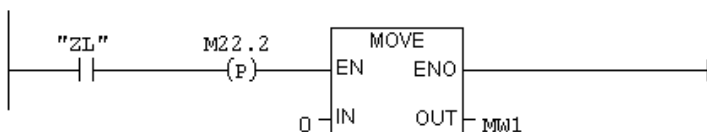
Network 16 : Title:

Comment:



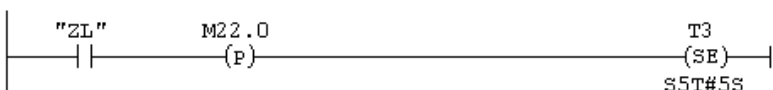
Network 17 : Title:

Comment:



Network 18 : Title:

Comment:



Network 19 : Title:

Comment:



(2) LED 数码显示 PLC 控制

利用 PLC 来控制一位七段 LED 数码管的显示，数码管的每一段都对应 PLC 的一个输出端子，PLC 输出端子的“1”、“0”状态对应于相应段的亮与灭。有两个按钮“+”、“-”，每按动一次“+”按钮，数字加 1，每按动一次“-”按钮，数字减 1。要求能正确显示数字 0-9 即可。

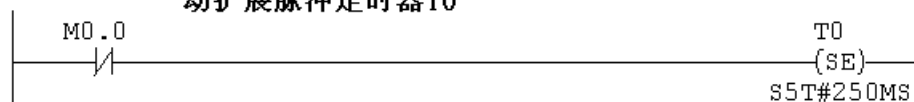
七段转换表

待转换的数据		七段显示的组成	用于七段显示的8-bit数据							七段显示
十六进制	二进制		g	f	e	d	c	b	a	
H0	0000		0	0	1	1	1	1	1	0
H1	0001		0	0	0	0	0	1	1	0
H2	0010		0	1	0	1	1	0	1	1
H3	0011		0	1	0	0	1	1	1	1
H4	0100		0	1	1	0	0	1	1	0
H5	0101		0	1	1	0	1	1	0	1
H6	0110		0	1	1	1	1	1	0	1
H7	0111		0	0	0	0	0	1	1	1
H8	1000		0	1	1	1	1	1	1	1
H9	1001		0	1	1	0	1	1	1	1
HA	1010		0	1	1	1	0	1	1	1
HB	1001		0	1	1	1	1	1	0	0
HC	1100		0	0	1	1	1	0	0	1
HD	1101		0	1	0	1	1	1	0	1
HE	1110		0	1	1	1	1	0	0	1
HF	1111		0	1	1	1	0	0	0	1

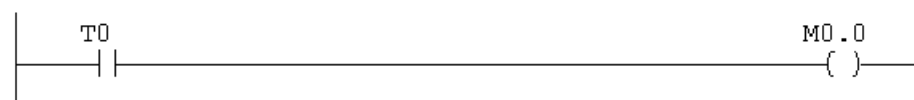
例二 时钟脉冲发生器

使用定时器实现自由设定时钟脉冲发生器功能(脉冲占空系数 1:1)。

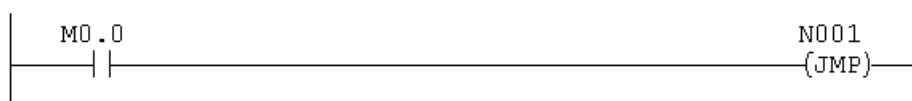
Network 1: 如果定时器T0的信号状态为0，则将时间值250MS装入T0并启动扩展脉冲定时器T0



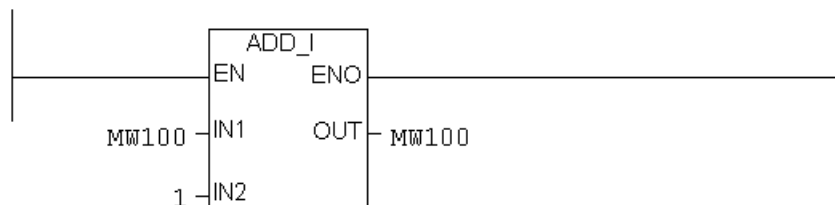
Network 2: 定时器状态暂时保存在辅助存储器标志中



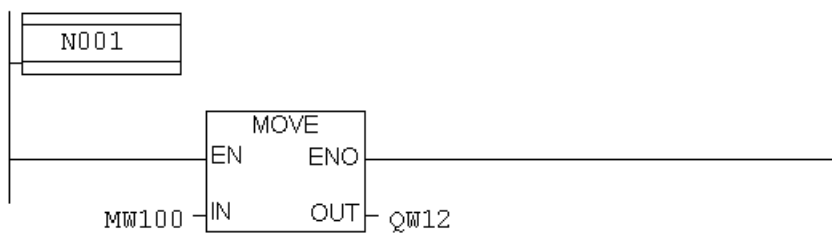
Network 3: 如果定时器T0的信号状态为1，则跳转至标号N001



Network 4: 当定时器T0信号状态为0时，存储器字100加1



Network 5: MOVE指令允许在输出Q12.0至13.7能够获得各种不同的时钟频率



存储字节 MB101 和 MB100 的单个位获得的频率：

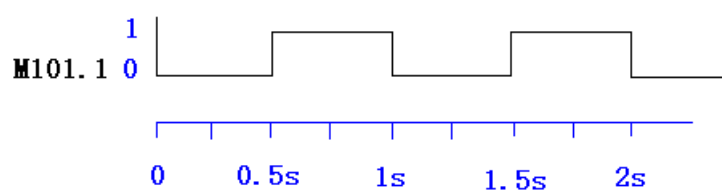
MB101/MB100	频率(Hz)	持续时间(S)
M101.0	2.0	0.5
M101.1	1.0	1
M101.2	0.5	2
M101.3	0.25	4
M101.4	0.125	8
M101.5	0.0625	16
M101.6	0.03125	32
M101.7	0.015625	64
M100.0	0.0078125	128
M100.1	0.0039062	256
M100.2	0.0019531	512
M100.3	0.0009765	1024
M100.4	0.0004882	2048
M100.5	0.0002441	4096
M100.6	0.000122	8192
M100.7	0.000061	16384

存储字节 MB101 各位的信号状态：

扫描 周期	存储字节 MB101 各位的信号状态								时 间 值 (MS)
	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	250
1	0	0	0	0	0	0	0	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	0	0	250
5	0	0	0	0	0	1	0	1	250

6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	0	0	250
9	0	0	0	0	1	0	0	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	0	0	250

M101.1 的信号状态:



练习:

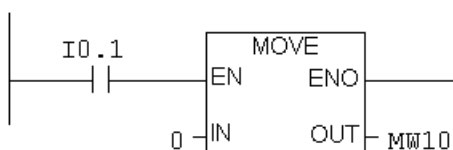
当按启动按钮时, L1 灯以 0.5s 的周期闪烁, L2 灯以 1s 的周期闪烁, L3 灯以 2s 的周期闪烁, L4 灯以 4s 的周期闪烁, 按下停止按钮, 所有灯熄灭。

参考答案:

Network 1:



Network 2:



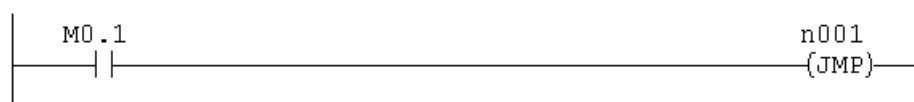
Network 3:



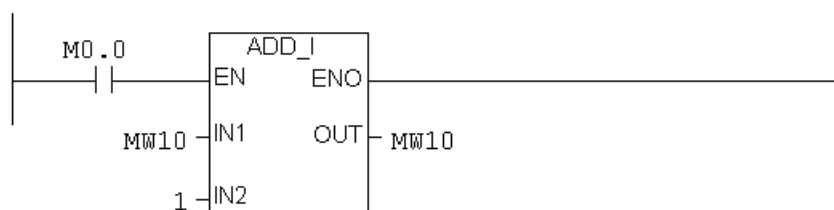
Network 4:



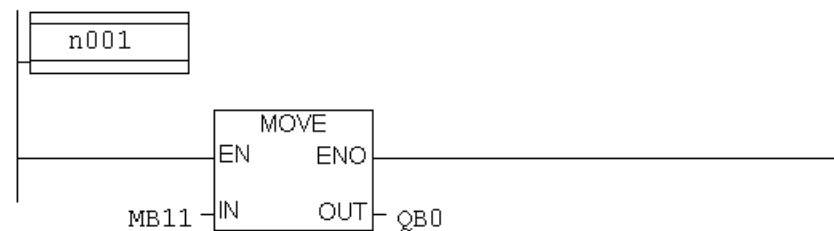
Network 5:



Network 6 :



Network 7 :



4.3.4 字逻辑运算指令

字逻辑指令将两个字(16 位)或双字(32 位)逐位进行逻辑运算.每个字或双字都必须分别存放在 2 个累加器中.操作的结果被存放在累加器 1 中,原有的内容被覆盖.

1. 字逻辑语句表指令

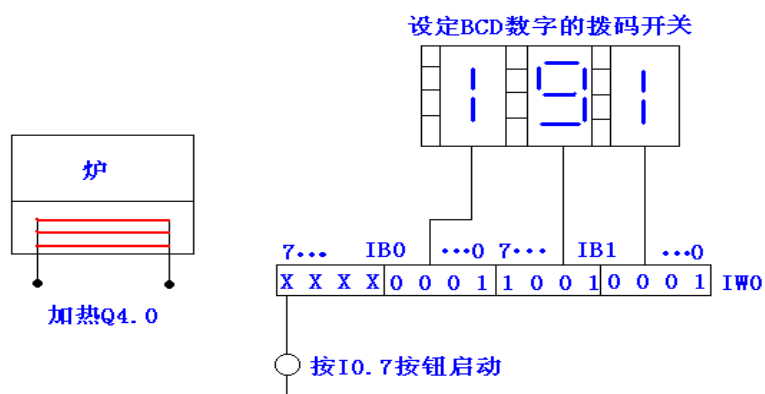
指令	说明
AW	两个字逐位进行 与 逻辑运算
OW	两个字逐位进行 或 逻辑运算
XOW	两个字逐位进行 异或 逻辑运算
AD	两个 双字 逐位进行 与 逻辑运算
OD	两个 双字 逐位进行 或 逻辑运算
XOD	两个 双字 逐位进行 异或 逻辑运算

2. 字逻辑方块指令

字与字相与	字与字相或	字与字相异或
双字相与	双字相或	双字相异或

例一 加热炉

操作员按启动按钮开始加热如下所示的加热炉.操作员能够使用如图所示的拨码开关设定加热时间.操作员设定的值以 **BCD** 格式用秒单位显示.



加热系统的元件和相应的绝对地址

系统元件	绝对地址
启动按钮	I0.7
个位数拨码开关	I1.0~I1.3
十位数拨码开关	I1.4~I1.7
百位数拨码开关	I0.0~I0.3
开始加热	Q4.0

梯形图程序:

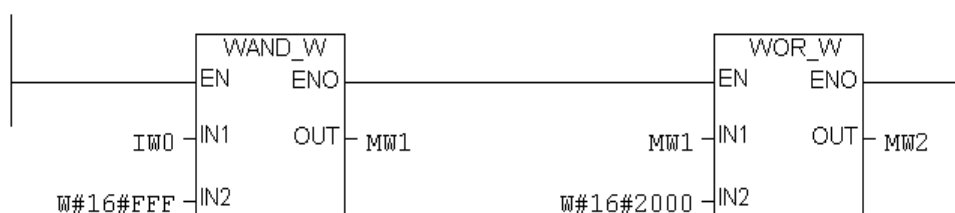
Network 1: 如果定时器运行, 则接通加热炉



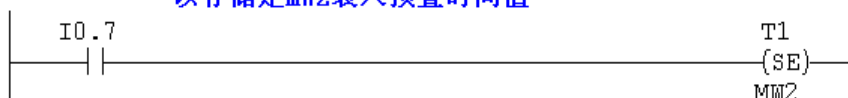
Network 2: 如果定时器运行, 则RETURN指令结束在此的处理



Network 3: 屏蔽输入位I0.4至I0.7, 不使用拨码开关输入这些位. 按照字与字相与指令将拨码开关输入的16位与W#16#FFF相结合. 结果装入存储字MW1. 为了设定以秒为单位的时基, 按照字与字相或指令将预置值和W#16#2000相结合, 将位13置1, 位12复0



Network 4: 如果按动启动按钮, 则启动定时器T1为扩展脉冲定时器, 以存储定MW2装入预置时间值



4.3.5 移位和循环移位指令

移位指令将累加器 1 低字中或整个累加器 1 的内容左移或右移.移动的位数在累加器 2 中或直接在指令中以常数给出.累加器 1 移位后空出的位, 填以 0 或符号位(0 代表正,1 代表负),被移动的最后一位保存在状态字中的 CC1 里,CC0 和 OV 被复位为 0.循环移位指令与一般移位指令的差别: 循环移位指令的空位填以从累加器中移出的位.

1. 无符号数移位指令

指令	说明
SLW	将累加器 1 低字中的内容逐位左移,空出位填以 0
SRW	将累加器 1 低字中的内容逐位右移,空出位填以 0
SLD	将累加器 1 整个内容逐位左移,空出位填以 0
SRD	将累加器 1 整个内容逐位右移,空出位填以 0

2. 有符号数移位指令

指令	说明
----	----

SSI	将累加器 1 低字中的内容逐位右移,空出位填以符号位(正填 0,负填 1)
SSD	将累加器 1 整个内容逐位右移,空出位填以符号位(正填 0,负填 1)

3. 循环移位指令

指令	说明
RLD	将累加器 1 整个内容逐位左移,空出位填以从累加器 1 移出的位
RRD	将累加器 1 整个内容逐位右移,空出位填以从累加器 1 移出的位
RLDA	将累加器 1 整个内容带 CC1 位逐位左移一位,空出位填以从 CC1 移出的位
RRDA	将累加器 1 整个内容带 CC1 位逐位右移一位,空出位填以从 CC1 移出的位

4. 移位和循环方块指令

字左移	字右移	双字左移
双字右移	整数右移	长整数右移

使用移位指令将输入 IN 中的数向左或右逐位移动,将输入 IN 中的内容左移相当于完成乘 2 加权,将输入 IN 中的内容右移相当于完成除以 2 加权的运算.

双字左循环移位	双字右循环移位

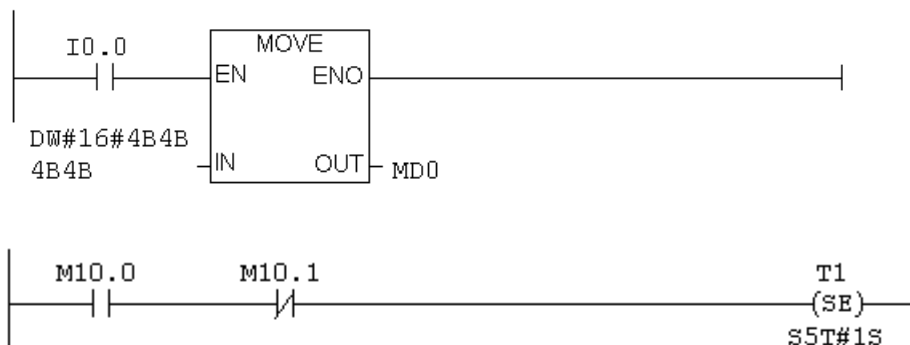
例一 彩灯控制

当按下启动按钮时，彩灯 L1、L2 同时亮；过 1s 后，L1 熄灭，L2 保持亮；过 1s 后，L1、L2 同时灭；过 1s 后，L1 亮,L2 保持灭；再过 1s 后，L1、L2 又同时亮，如此循环闪烁，直到按下停止按钮，彩灯工作终止。

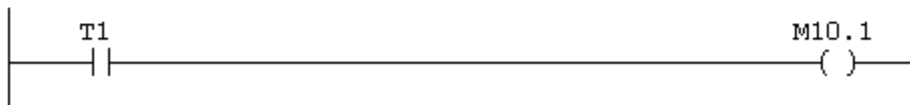
Network 1 :



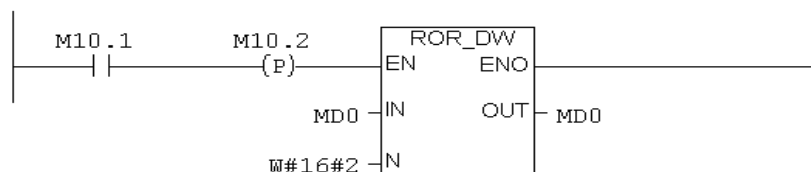
Network 2 :



Network 4 : Title:



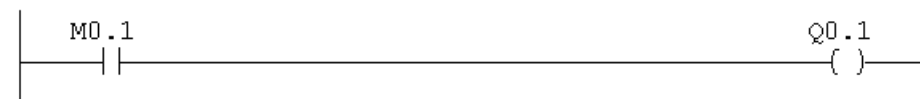
Network 5 :



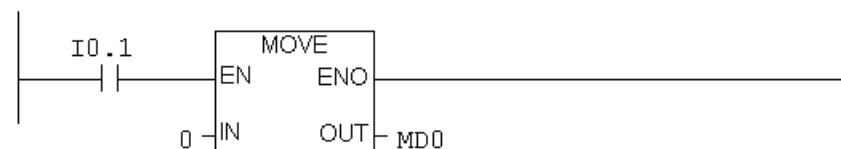
Network 6 :



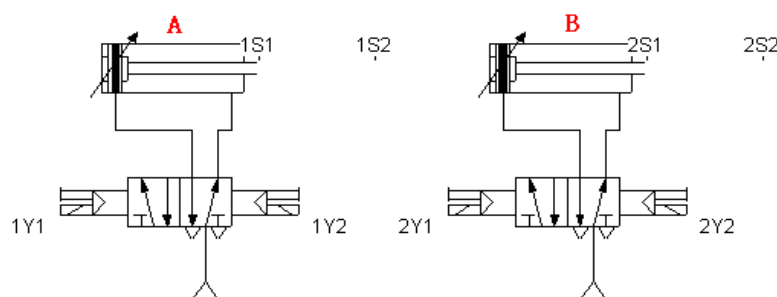
Network 7 :



Network8 :

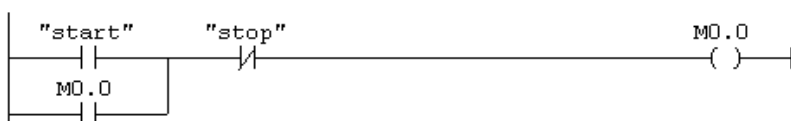


例二 双缸顺序动作回路 A1B1B0A0



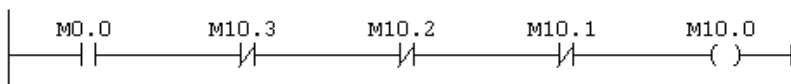
Network 1 : Title:

Comment:



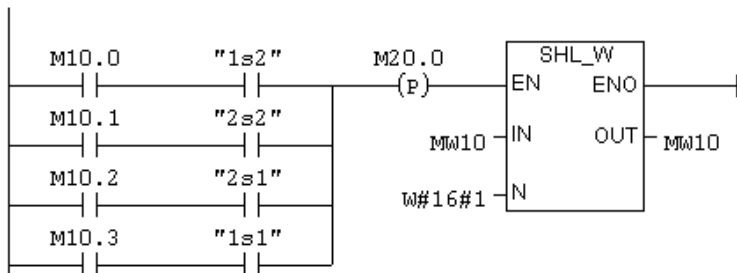
Network 2 : Title:

Comment:



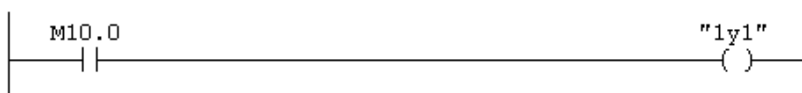
Network 3 : Title:

Comment:



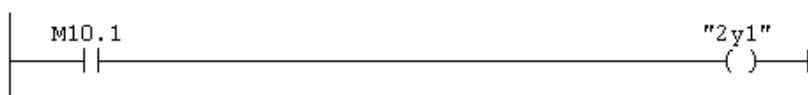
Network 4 : Title:

Comment:



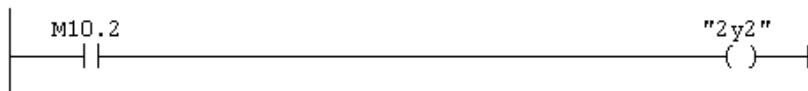
Network 5 : Title:

Comment:



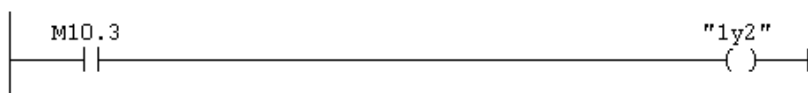
Network 6: Title:

Comment:



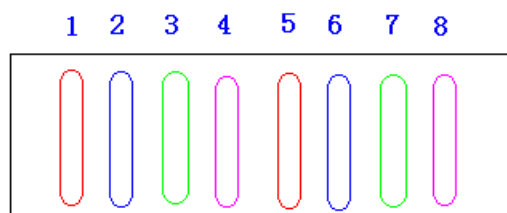
Network 7: Title:

Comment:



练习一：霓虹灯广告屏控制器的设计

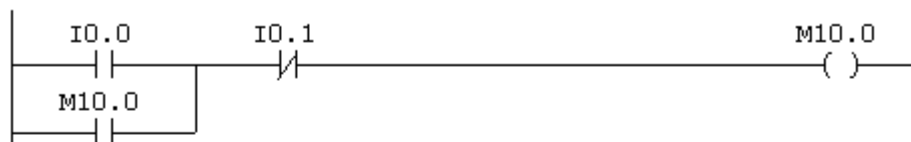
用 PLC 对霓虹灯广告屏实现控制，其具体要求如下：



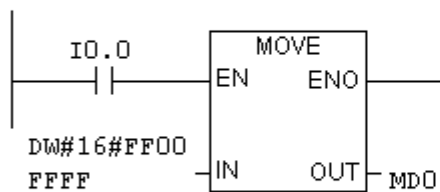
该广告屏中间 8 个灯管亮灭的时序为第 1 根亮→第 2 根亮→第 3 根亮→...→第 8 根亮，时间间隔为 1s，全亮后，显示 10s，再反过来从 8→7→...→1 顺序熄灭。全灭后，停亮 2s，再从第 8 根灯管开始亮起，顺序点亮 7→6→...→1，时间间隔为 1s，显示 20s，再从→2→...→8 顺序熄灭。全熄灭后，停亮 2s，再从头开始运行，周而复始。

参巧梯形图程序：

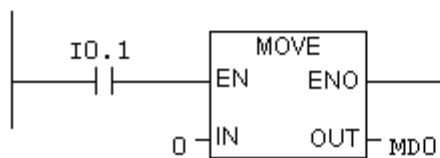
Network 1 : Title:



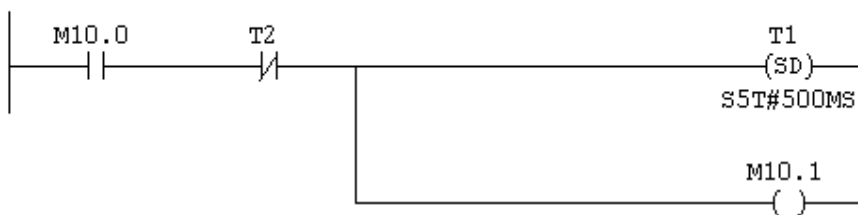
Network 2 : Title:



Network 3 : Title:



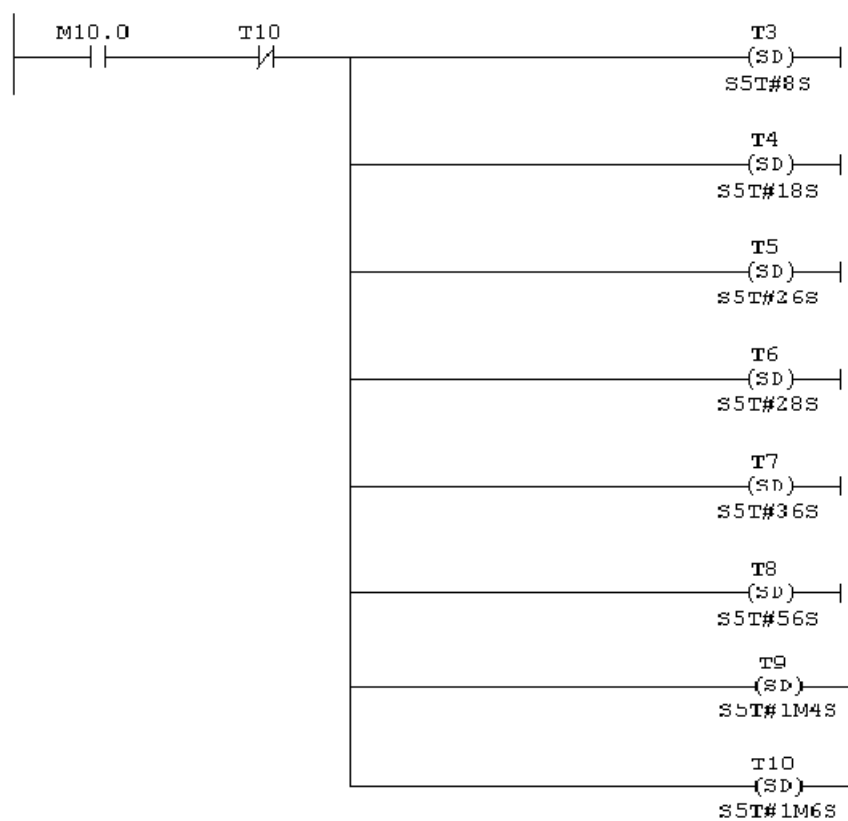
Network 4 : Title:



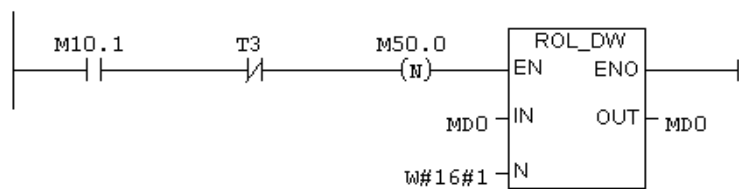
Network 5 : Title:



Network 6 : Title:



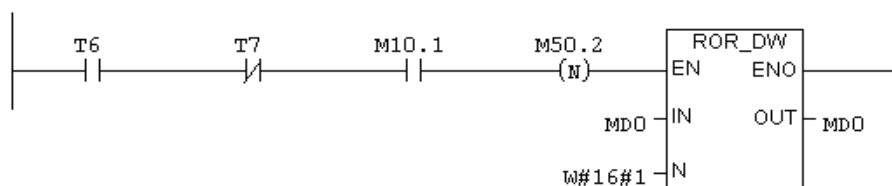
Network 7 : Title:



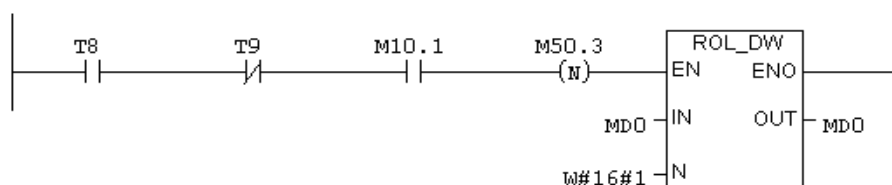
Network 8 : Title:



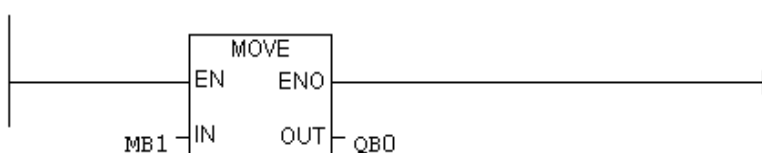
Network 9 : Title:



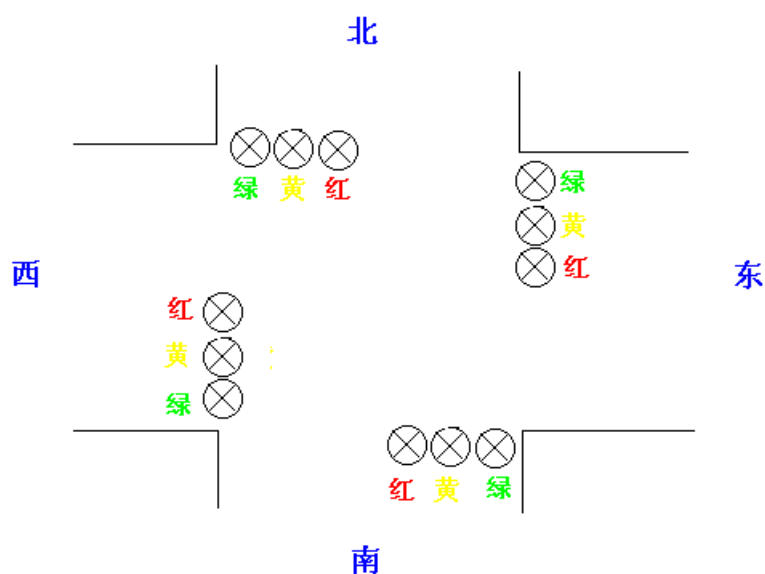
Network 10 : Title:



Network 11 : Title:



练习二：十字路口的交通指挥信号灯布置如下图：

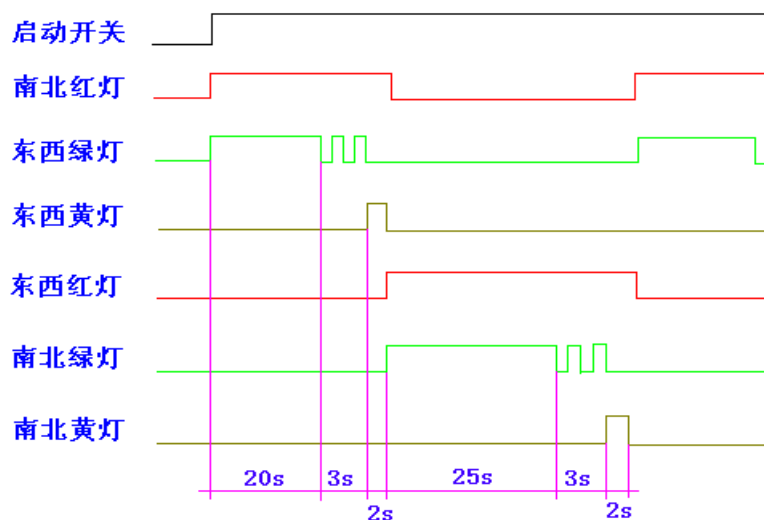


一、控制要求

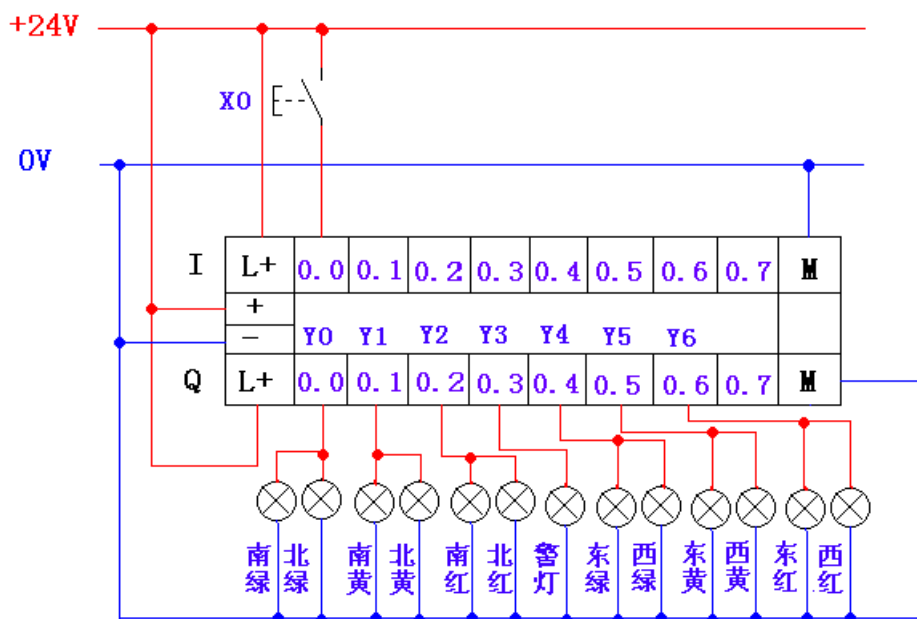
- (1) 信号灯系统由一个启动开关控制，当启动开关接通时，该信号灯系统开始工作，当启动开关关断时，所有信号灯都熄灭。
- (2) 南北绿灯和东西绿灯不能同时亮。如果同时亮应关闭信号灯系统，并立刻报警。
- (3) 南北红灯亮维持 25s。在南北红灯亮的同时东西绿灯也亮，并维持 20s。到 20s 时，东西绿灯闪亮，闪亮 3s 后熄灭，此时，东西黄

灯亮，并维持 2s。到 2s 时，东西黄灯熄灭，东西红灯亮。同时，南北红灯熄灭，南北绿灯亮。

- (4) 东西红灯亮维持 30s。南北绿灯亮维持 25s，然后闪亮 3s 后熄灭。同时南北黄灯亮，维持 2s 后熄灭，这时南北红灯亮，东西绿灯亮。
- (5) 以上南北、东西信号灯周而复始地交替工作状态，指挥着十字路口的交通，其时序如下所示。

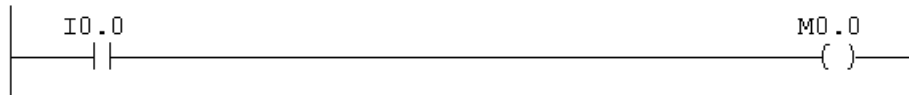


二、PLC 接线

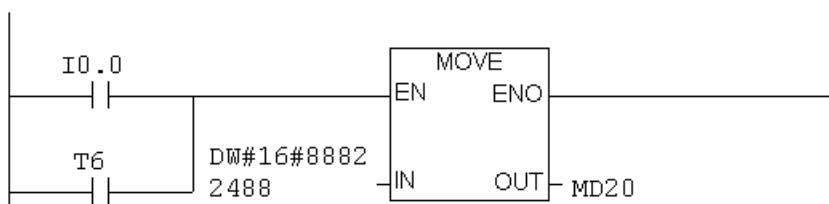


三、梯形图程序

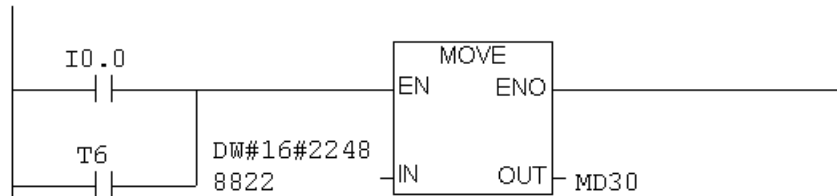
Network 1 :



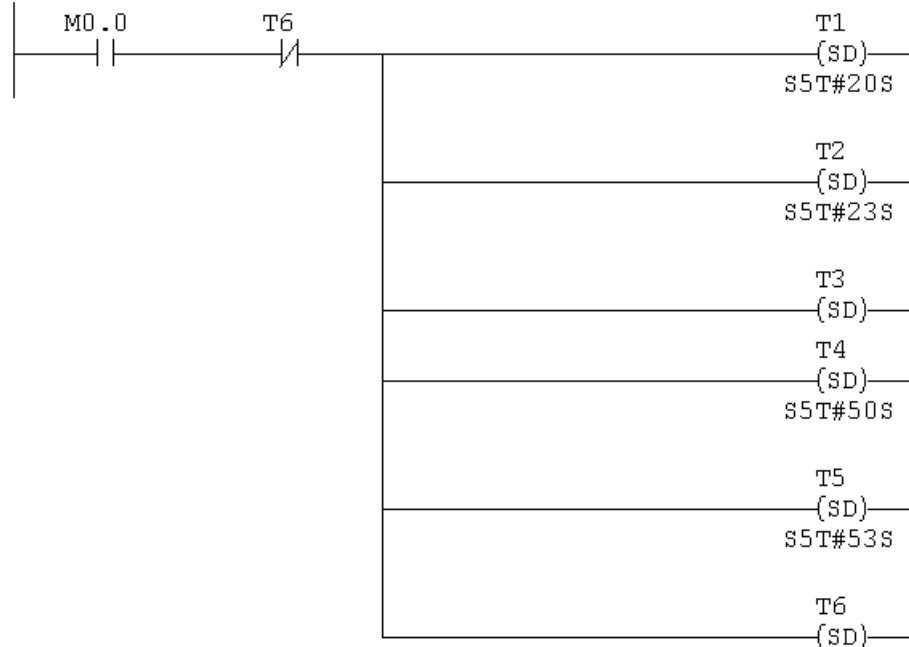
Network 2 :



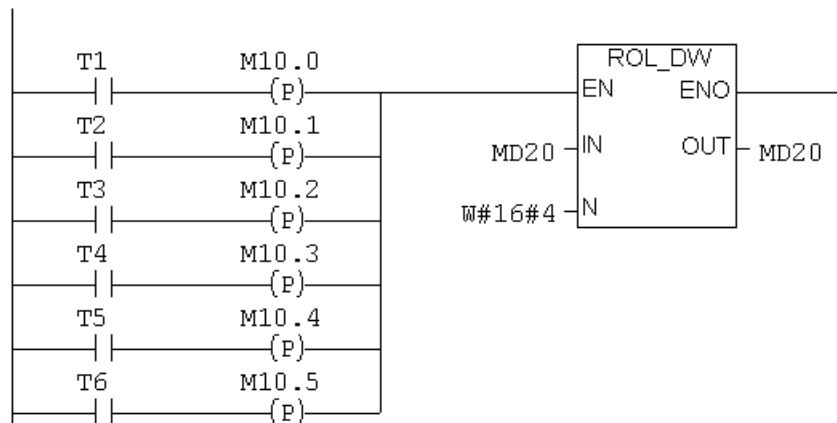
Network 3 :



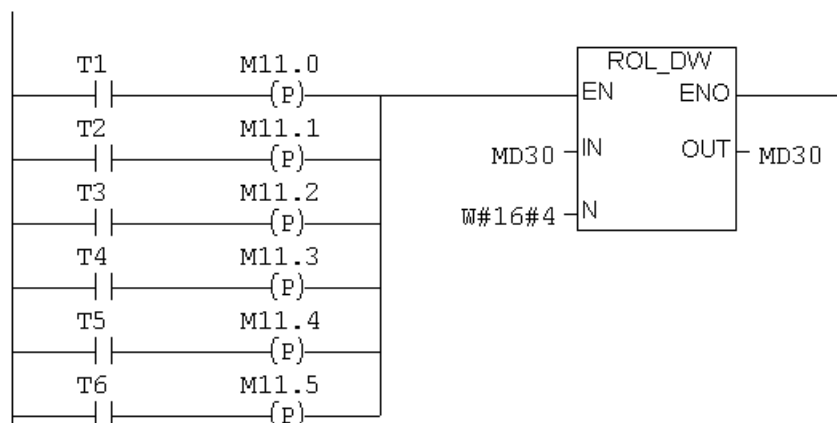
Network 4 :



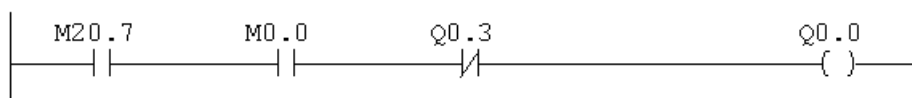
Network 5 :



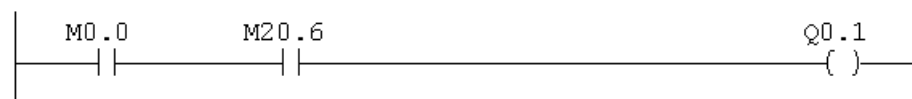
Network 6 :



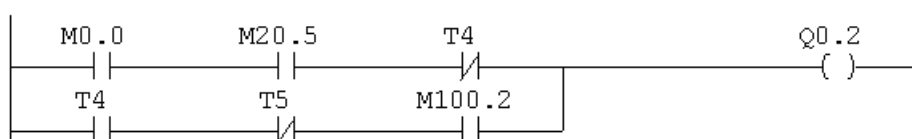
Network 7 :



Network 8 :



Network 9 :



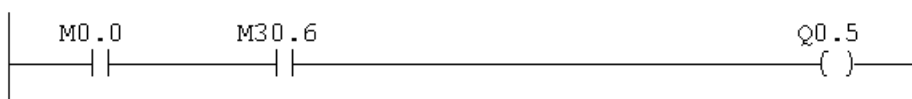
Network 10 :



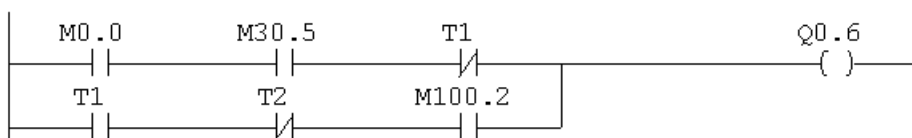
Network 11 :



Network 12 :



Network 13 :



练习 (1): 编制智力竞赛抢答器控制程序

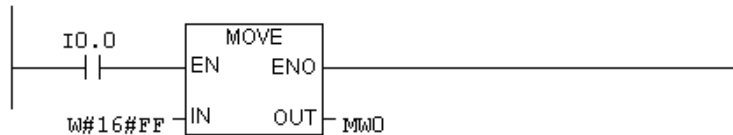
K0 为开始/复位控制开关 (断开为复位, 闭合为开始), L0 为抢答开始指示, K1, K2, K3, K4, K5, K6 为六组抢答按钮, L1, L2, L3, L4, L5, L6 为六组抢答指示。K7, K8, K9 分别为时间控制开关。KA 为手控时间到按钮。L7 为抢答时间到指示。当 K7, K8,

K9 全为断开时为手控时间控制（此时由人工记时间到时按下 KA），当开关 K7，K8，K9 处于其它不同状态时，对应于抢答设定时间分别为 5S，10S，15S，20S，25S，30S，35S。当 K0 处于复位状态时，如有某个按钮最先按下，该组的指示灯就“闪亮”，表示该组违例（其他组被锁定）。当 K0 处于开始状态时，如有某个按钮最先按下，该组的指示灯就“常亮”，表示该组抢答有效（其他组被锁定）。如在设定时间到未有人抢答，则 L7 灯亮，各组抢答按钮被锁定，直到开关 K0 复位。

练习（2）：依次按 8 次按钮 I0.1 时，8 盏指示灯依次亮，再依次按 8 次按钮 I0.1 时，8 盏指示灯依次灭，按 I0.0 开始新的循环操作，任何时候按 I0.2 时所有的灯灭。

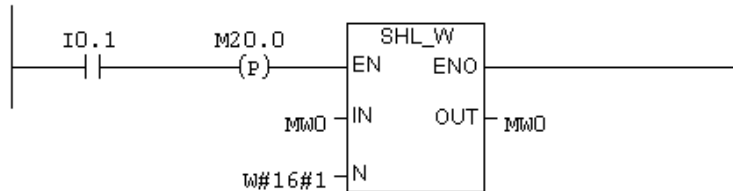
Network 1 : Title:

Comment:



Network 2 : Title:

Comment:



Network 3 : Title:

Comment:



Network 4 : Title:

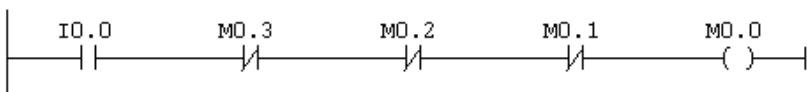
Comment:



练习（3）：按下开关 I0.0，L1、L2、L3、L4 依次亮灭，周而复始，时间间隔为 1S（即 L1 亮 1S 后灭，接着 L2 亮，如此循环）。

Network 1 : Title:

Comment:



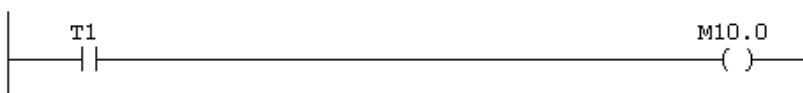
Network 2 : Title:

Comment:



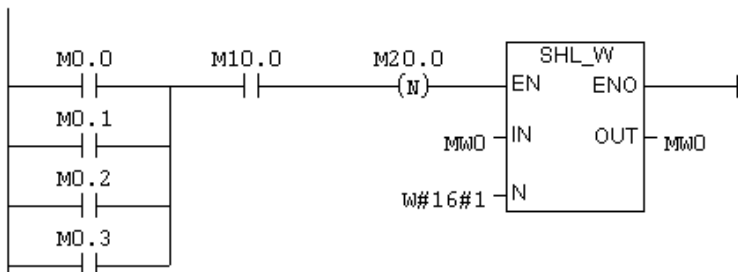
Network 3 : Title:

Comment:



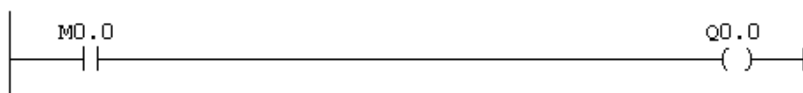
Network 4 : Title:

Comment:



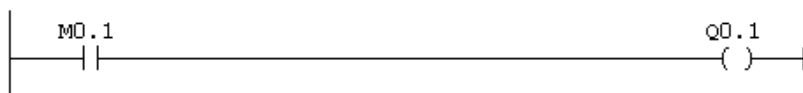
Network 5 : Title:

Comment:



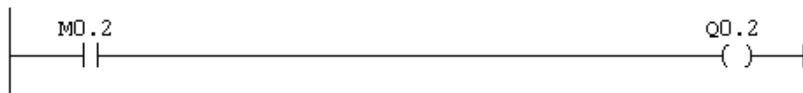
Network 6 : Title:

Comment:



Network 7 : Title:

Comment:



Network 8 : Title:

Comment:



练习(4):

试编写用 PLC 控制三个霓虹灯闪烁的程序。工作要求如下:

- (1) 首先 A 灯亮;
- (2) 1S 后 A 灯灭, B 灯亮;
- (3) 再过 1S 后 B 灯灭, C 灯亮;
- (4) 再过 1S 后 C 灯灭;
- (5) 再过 1S 后, A、B、C 三灯全亮;
- (6) 再过 1S 后, A、B、C 三灯全灭;

(7) 再过 1S 后，A、B、C 三灯全亮；

(8) 再过 1S 后，A、B、C 三灯全灭。

然后 (1) ~ (8) 步重复循环。要求用一个开关控制，当它闭合接通时霓虹灯工作，断开时停止工作。

	Symbol	Address	Data Type	Comment
1	L1	Q 0.0	BOOL	
2	L2	Q 0.1	BOOL	
3	L3	Q 0.2	BOOL	
4	Switch	I 0.0	BOOL	

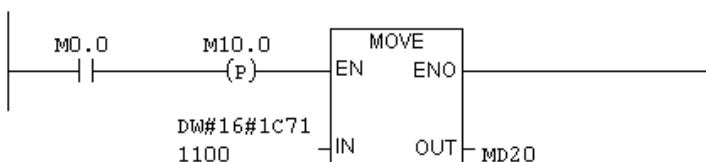
Network 1: Title:

Comment:



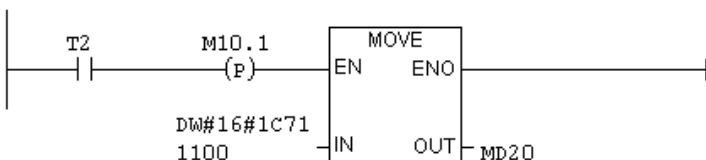
Network 2: Title:

Comment:



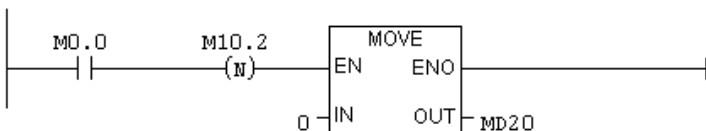
Network 3: Title:

Comment:



Network 4: Title:

Comment:



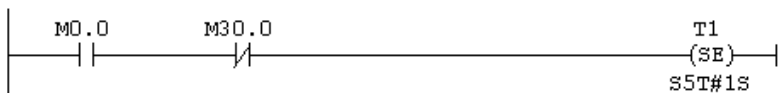
Network 5 : Title:

Comment:



Network 6 : Title:

Comment:



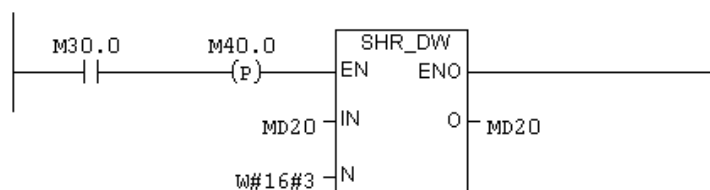
Network 7 : Title:

Comment:



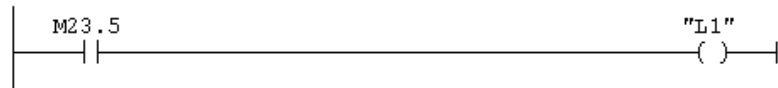
Network 8 : Title:

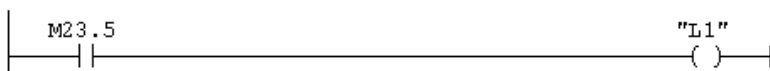
Comment:



Network 9 : Title:

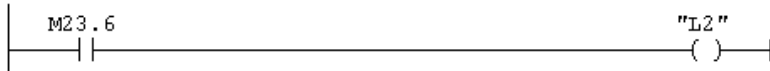
Comment:





Network 10 : Title:

Comment:



Network 11 : Title:

Comment:



4.3.6 打开数据块指令

LAD元素	参数	数据类型	说明
<DB号>或<DI号> -(OPN)-	<DB号>或 <DI号>	BLOCK_DB	DB和DI的序号范围与CPU有关

例如:

Network 1 :



Network 2 :



4.4 控制指令

4.4.1 逻辑控制指令

逻辑控制指令是指逻辑块内的跳转和循环指令，这些指令中止程序原有的线性逻辑流，跳到另一处执行程序。跳转或循环指令的操作数是地址标号，该地址标号指出程序要跳往何处，标号最多为4个字符，第一个字符必须是字母，其余字符可为字母或数字。与它相同的标号还必须写在程序跳转的目的地前，称为目标地址标号。在一个逻辑块内，目标地址标号不能重名。在语句表中，目标标号与目标指令用冒号分隔。在梯形图中目标标号必须在一个网络的开始。由于STEP7的跳转指令只能在逻辑块内跳转，所以，在不同逻辑块中的目标标号可以重名。

1. 无条件跳转指令

指令	说明
JU	无条件跳转
JL	跳转表格

无条件跳转指令（JU）无条件中断正常的程序逻辑流，使程序跳转到目标处继续执行。

跳转表格指令（JL）实质上是多路分支跳转语句，它必须与无条件跳转指令一起使用。多路分支的路径参数存放于累加器 1 中。

例：累加器中的值为 2，则程序执行时跳到 sub2，这时 Q0.2 灯亮。

```

L      2
JL     list
JU     sub0
JU     sub1
JU     sub2
JU     sub3
list:  JU     comm
sub0:  AN     M      0.0
      =      Q      0.0
      JU     comm
sub1:  AN     M      0.1
      =      Q      0.1
      JU     comm
sub2:  AN     M      0.2
      =      Q      0.2
      JU     comm
sub3:  AN     M      0.3
      =      Q      0.3
      JU     comm
comm:  AN     M      0.4
      =      Q      0.4

```

2. 条件跳转指令

指令	说明
JC	当 RLO=1 时跳转
JCN	当 RLO=0 时跳转
JCB	当 RLO=1 且 BR=1 时跳转，指令执行时将 RLO 保存在 BR 中
JNB	当 RLO=0 且 BR=0 时跳转，指令执行时将 RLO 保存在 BR 中
JB	当 BR=1 时跳转，指令执行时，OR、FC 清 0，STA 置 1
JNBI	当 BR=0 时跳转，指令执行时，OR、FC 清 0，STA 置 1
JO	当 OV=1 时跳转
JOS	当 OS=1 时跳转，指令执行时，OS 清 0
JZ	累加器 1 中的计算结果为 0 跳转
JN	累加器 1 中的计算结果为非 0 跳转
JP	累加器 1 中的计算结果为正跳转

JM	累加器 1 中的计算结果为负跳转
JMZ	累加器 1 中的计算结果小于等于 0 跳转
JPZ	累加器 1 中的计算结果大于等于 0 跳转
JUO	实数溢出跳转

3. 循环指令

使用循环指令（LOOP）可以多次重复执行特定的程序段，重复执行的次数存在累加器 1 中，即以累加器 1 为循环计数器。LOOP 指令执行时，将累加器 1 低字中的值减 1，如果不为 0，则回到循环体开始处继续循环过程，否则执行 LOOP 指令后面的指令。循环体是指循环标号和 LOOP 指令间的程序段。由于循环次数不能是负数，所以程序应保证循环计数器中的数为正整数（数值范围：0~32767）或字型数据（数值范围：W # 16 # 0000~W # 16 # FFFF）

4. 梯形图逻辑控制指令

指令	说明
<地址> —(JMP)	用于无条件跳转或以 RLO=1 为跳转条件。无条件跳转时不影响状态字，条件跳转时，清 OR、FC；置位 STA、RLO
<地址> —(JMPN)	当 RLO=0 时跳转，清 OR、FC；置位 STA、RLO

在 S7 中，没有根据算术运算结果直接转移的梯形逻辑指令。但通过使用反映字各位状态的常开常闭触点，并使用前面两条跳转指令，即可实现根据运算结果的跳转功能。

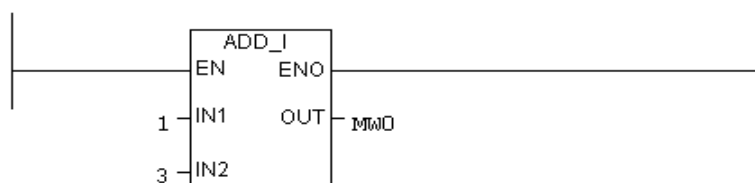
状态位常开常闭触点：

LAD指令		说明
$\begin{array}{c} >0 \\ \text{---} \end{array}$	$\begin{array}{c} >0 \\ \text{---} \end{array}$	算术运算结果大于0，则常开触点闭合常闭触点断开，该指令检查状态字条件码CC0和CC1的组合，决定结果与0的关系
$\begin{array}{c} \geq 0 \\ \text{---} \end{array}$	$\begin{array}{c} \geq 0 \\ \text{---} \end{array}$	算术运算结果大于等于0，则常开触点闭合常闭触点断开，指令检查状态字条件码CC0和CC1的组合，决定结果与0的关系
$\begin{array}{c} \text{UO} \\ \text{---} \end{array}$	$\begin{array}{c} \text{UO} \\ \text{---} \end{array}$	浮点算术运算结果溢出，则常开触点闭合常闭触点断开，指令检查状态字条件码CC0和CC1的组合
$\begin{array}{c} \text{BR} \\ \text{---} \end{array}$	$\begin{array}{c} \text{BR} \\ \text{---} \end{array}$	若状态字的位BR为1，常开触点闭合常闭触点断开

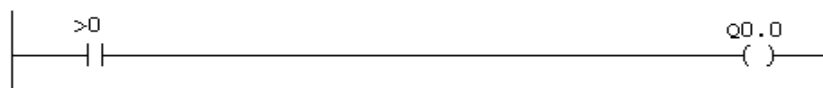
例：

OB1 : Title:

Network 1: Title:



Network 2: Title:



4.4.2 程序控制指令

程序控制指令是指功能块（FB、FC、SFB、SFC）调用指令和逻辑块（OB、FB、FC）结束指令。调用块或结束块可以有条件的或无条件的。STEP 7 中的功能块实质上就是子程序。

1. STL 程序控制指令

指令	说明
CALL	该指令在程序中无条件执行，调用 FB，FC，SFB，SFC
UC	该指令在程序中无条件调用功能块（一般是 FC 或 SFC），但不能传递参数
CC	RLO=1，调用功能块（一般是 FC），但不能传递参数
BEU	该指令无条件结束当前块的扫描，将控制返还给调用块
BEC	RLO=1，结束当前块的扫描，将控制返还给调用块，若 RLO=0，则将 RLO 置 1，程序继续在当前块内扫描

CALL 指令可以调用用户编写的功能块或操作系统提供的功能块，CALL 指令的操作数是功能块类型及其编号，当调用的功能块是 FB 块时还要提供相应的背景数据块 DB，使用 CALL 指令可以为被调用功能块中的形参赋以实际参数，调用时应保证实参与形参的数据类型一致。

例如：

CALL FB 40, DB 41 //调用 FB40，并指明背景数据块为 DB41

IN1:= I 1.0 //将实参 I1.0 分配给形参 IN1

IN2:= MW 2 //将实参 MW 2 分配给形参 IN2

OUT1:= MD 20 //给形参 OUT1 分配给实参 MD20

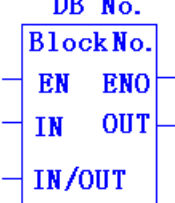
L MD 20 //调用结束后，FB40 的运行结果在 MD20 中。

UC 和 CC 指令用于不需传递参数的场合，如：

```
CC    FC 12          //当 RLO=1 时调用 FC 12
UC    FC 12          //不管 RLO 结果，调用 FC 12
```

2. 梯形图程序控制指令

梯形图调用块有两种方式：一是用线圈驱动指令调用功能块，这种方式相当于 STL 指令 UC 和 CC，不能实现参数传递；二是用方块指令调用功能块，相当于 STL 指令 CALL，可以传递参数。

LAD指令	参数	数据类型	说明
	No.	BLOCK_FC	No. 为被调用的不带参数的FC或SFC号数
	DB No. Block No.	BLOCK_DB BLOCK_FB BLOCK_FC	背景数据块号, 只在调用FB时提供这一数据被调用的功能块
	EN	BOOL	允许输入
	ENO	BOOL	允许输出
	无	—	块结束

4.4.3 主控继电器指令

主控继电器是一种美国梯形图逻辑主控开关,用来控制信号流(电流路径)的通断。

在 STEP 7 中与主控继电器相关的指令：

STL 指令	LAD 指令	说明
MCRA	—(MCRA)	激活 MCR 区,该指令表明一个按 MCR 方式操作区域的开始
MCRD	—(MCRD)	激活 MCR 区(应与 MCRA 成对使用),该指令表明一个按 MCR 方式操作区域的结束
MCR(	—(MCR<)	主控继电器,该指令将 RLO 保存于 MCR 堆栈中,产生一条子母线,其后的指令与子母线相连
)MCR	—(MCR>)	恢复 RLO,结束子母线,返回主母线

注意：

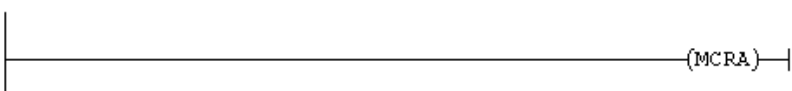
- ①在 MCRA 和 MCRD 指令之间的指令将根据 MCR 位的信号状态进行操作;在 MCRA 和 MCRD 之外的指令串中,即使有 MCR 位也不按其操作,而是视主控触点为闭合进行操作。

- ②若在 MCRA 和 MCRD 之间有 BEU 指令,则 CPU 执行到 BEU 指令时也结束 MCR 区域.
- ③若在激活的 MCR 区域中有块调用指令,则激活状态不能继承到被调用块中,必须在被调用块中重新激活 MCR 区,才能使指令根据 MCR 位操作.
- ④MCR(指令和)MCR 指令要成对使用,以表示受控子母线的形成与终止.MCR 指令可以嵌套使用,最大的嵌套深度是 8 层.
- ⑤在 STEP 7 中主控继电器动作与否,影响与子母线相连的控制逻辑的操作结果.

MCR 信号状 态	= (输出线圈或中 间输出)	S 或 R (置位或复位)	T (传送或赋值)
0	写入 0 模仿掉电时继 电器的静止状 态	不写入 模仿掉电时的 自锁继电器,使 其保持当前的 状态	写入 0 模仿一个元件, 在掉电时产生 0 值
1	正常执行	正常执行	正常执行

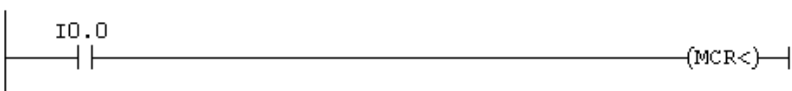
Network 1 : Title:

Comment:



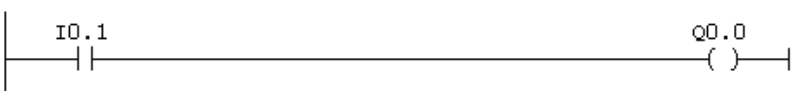
Network 2 : Title:

Comment:



Network 3 : Title:

Comment:



Network 4 : Title:

Comment:

(MCR>)

Network 5 : Title:

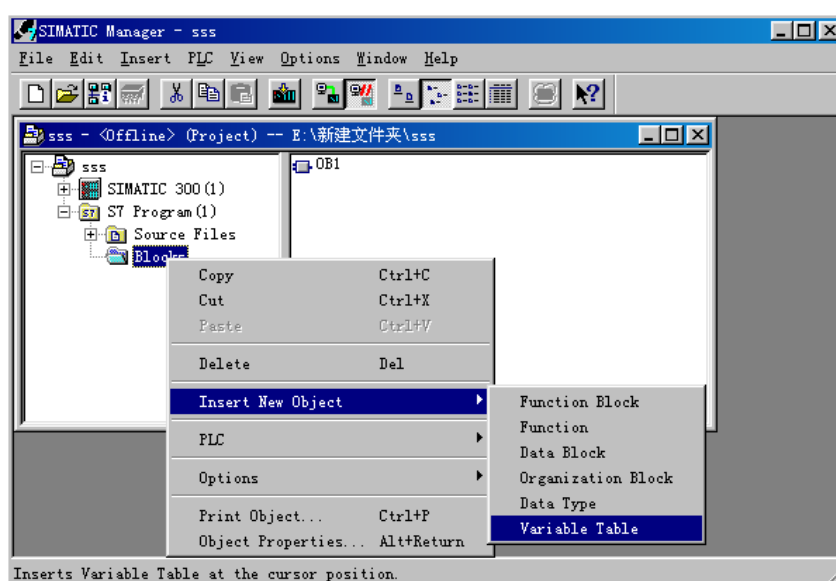
Comment:

(MCRD)

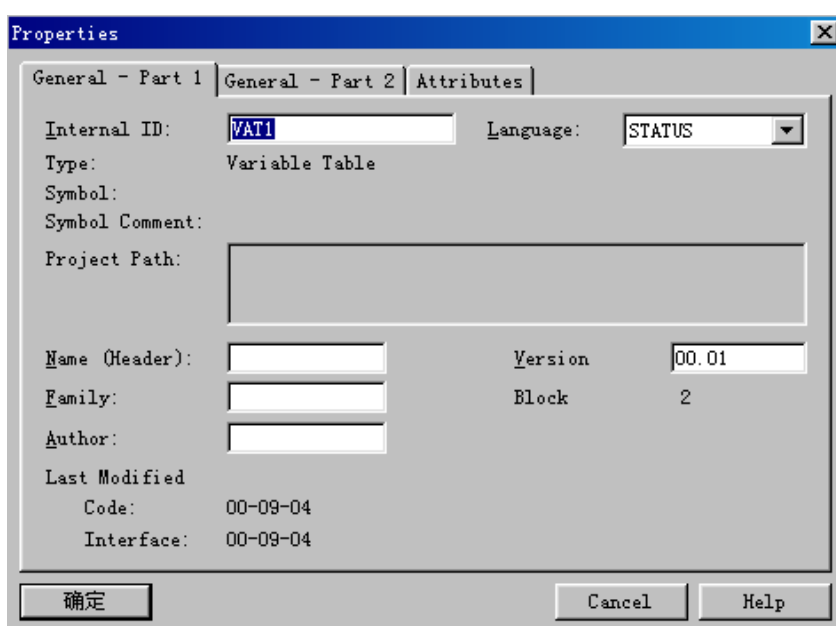
监控变量

和强制变量

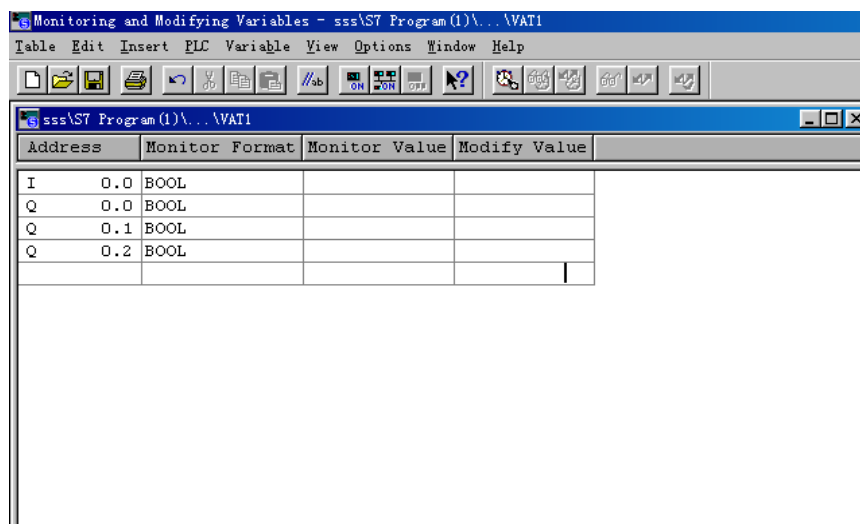
①创建一个的变量表



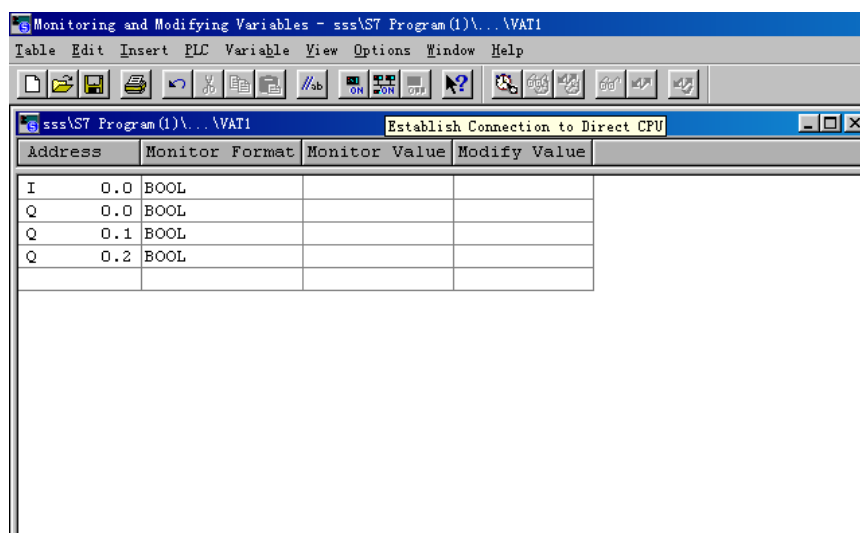
②给变量表命名



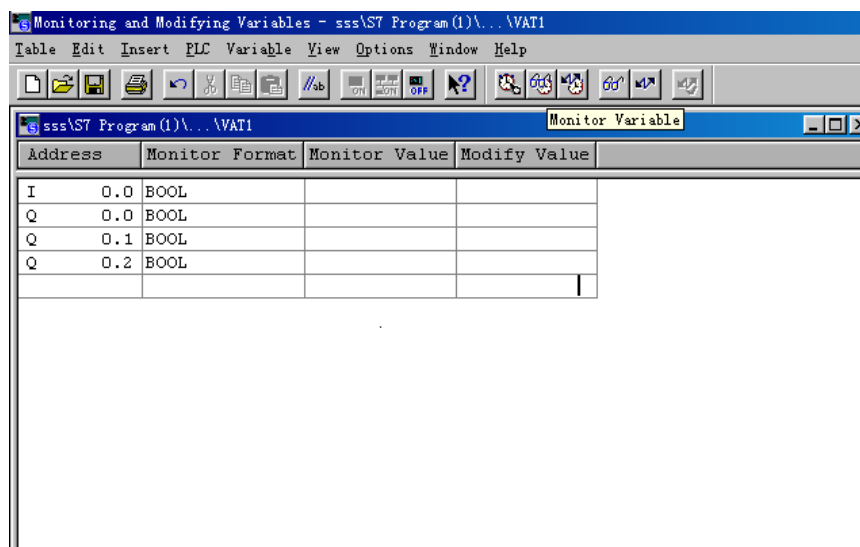
③打开变量表，输入监控变量



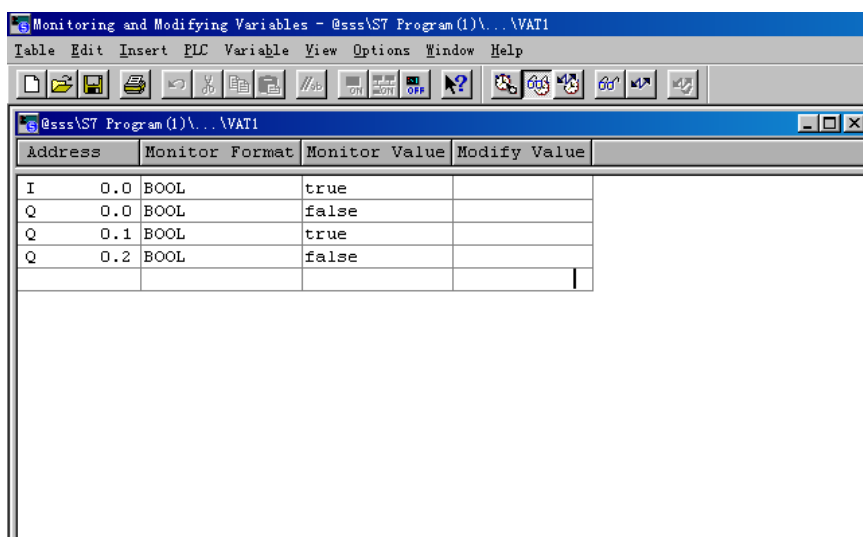
④建立 CPU 连接



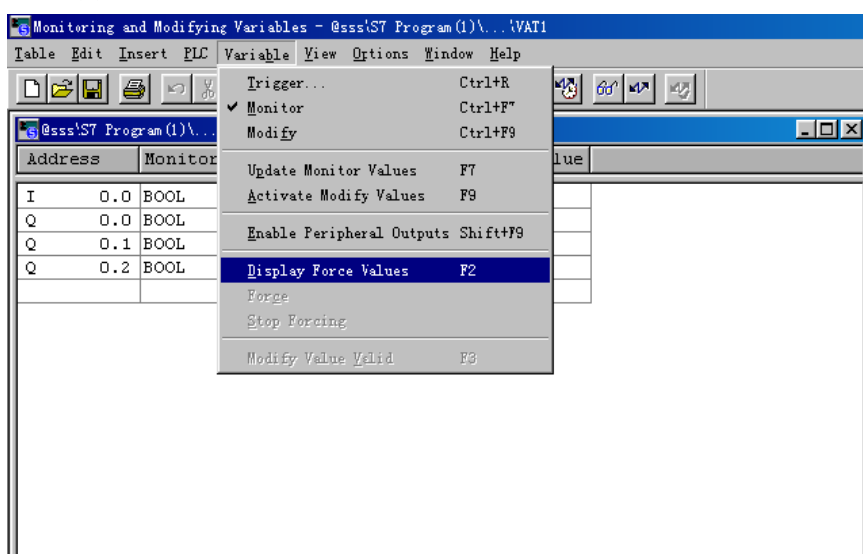
⑤启动监视



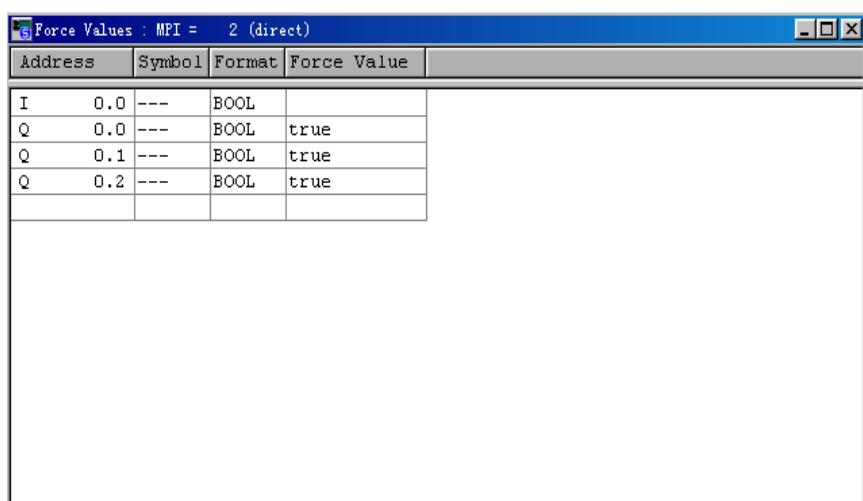
⑥监视情况



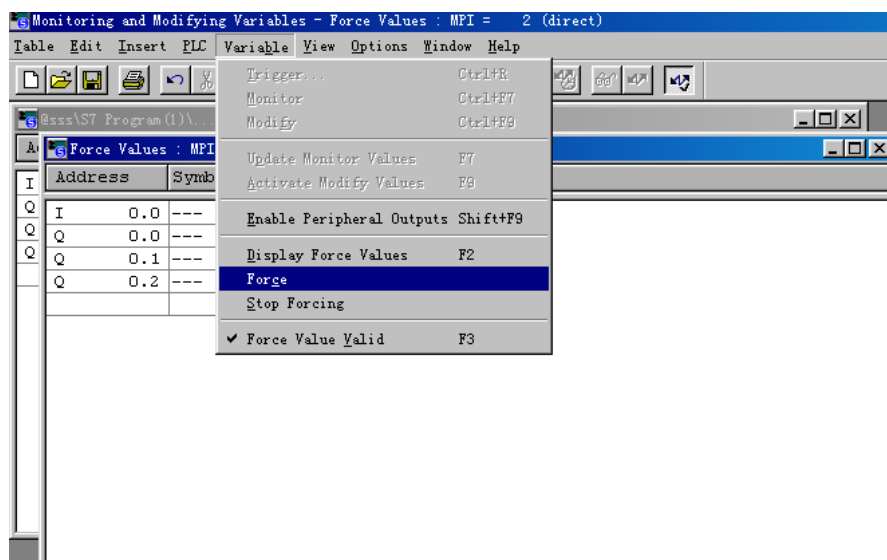
⑦显示强制变量表



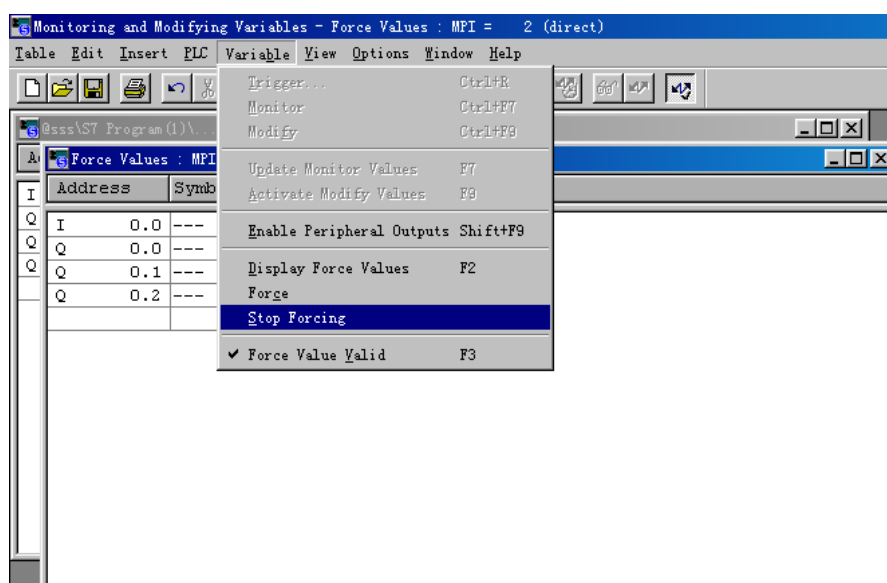
⑧设置强制值



⑨启动强制

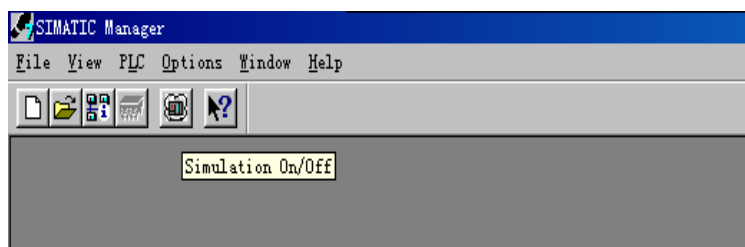


⑩停止强制

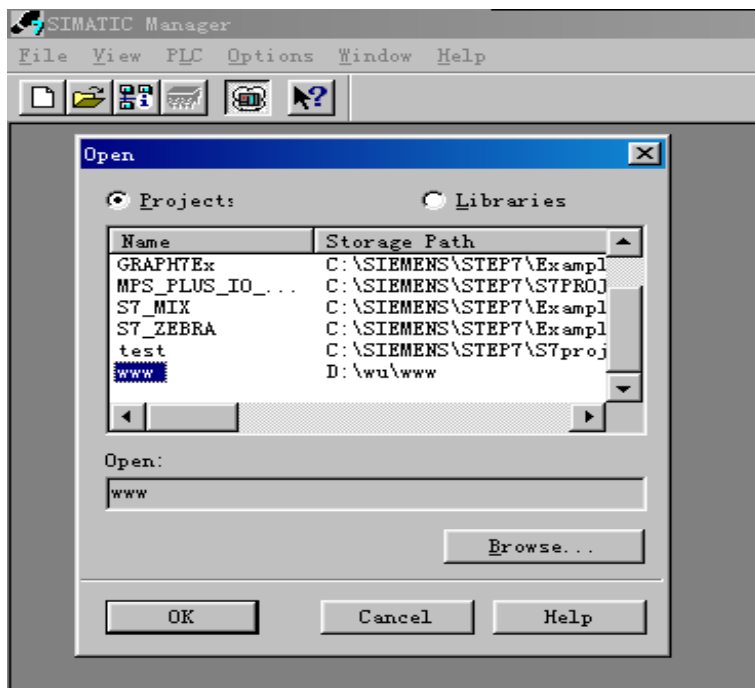


PLCSIM 的使用:

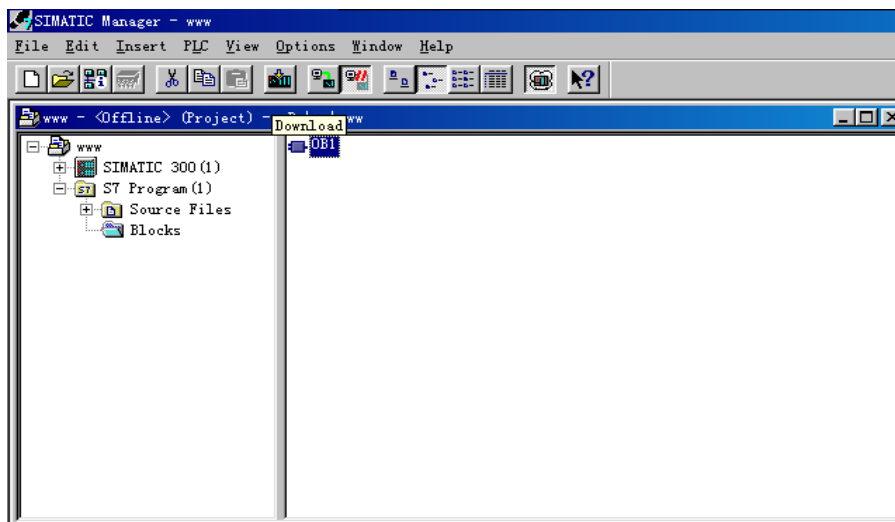
①进入 SIMATIC 管理器，点击“仿真”图标



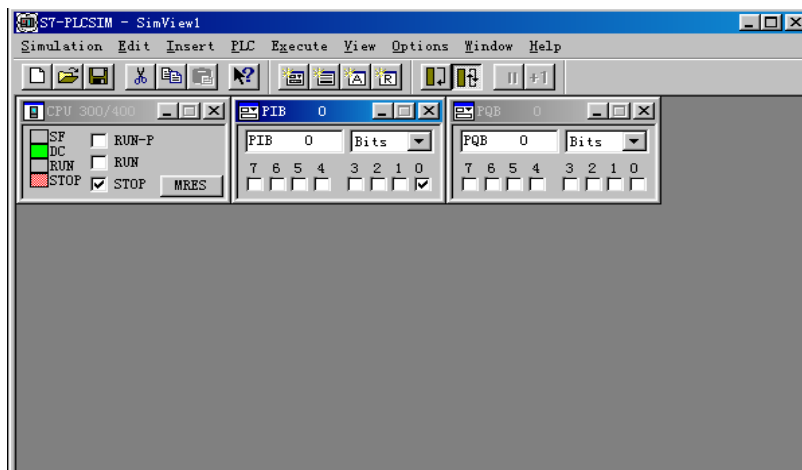
②打开要仿真的项目（如“WWW”）



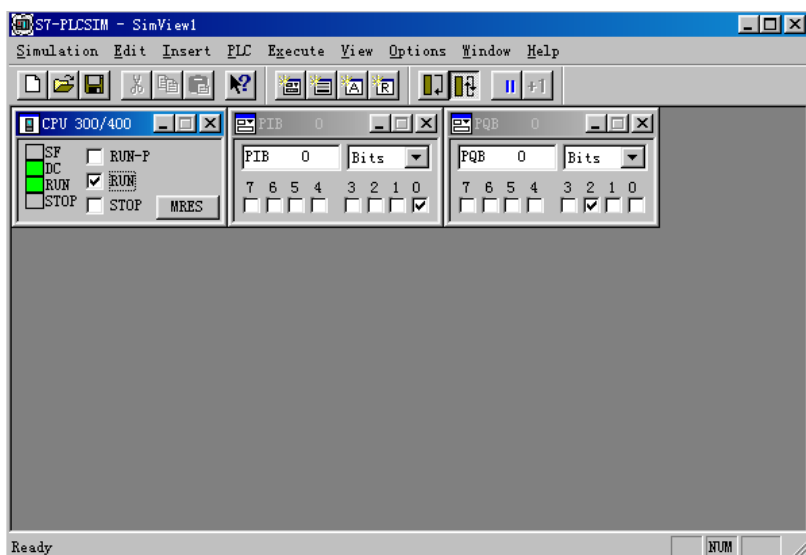
③下载组织块 OB1



④插入输入值、输出值表，并设定相应的输入值



⑤点击 CPU 模式表，将 CPU 工作在“RUN”或“RUN—P”模式

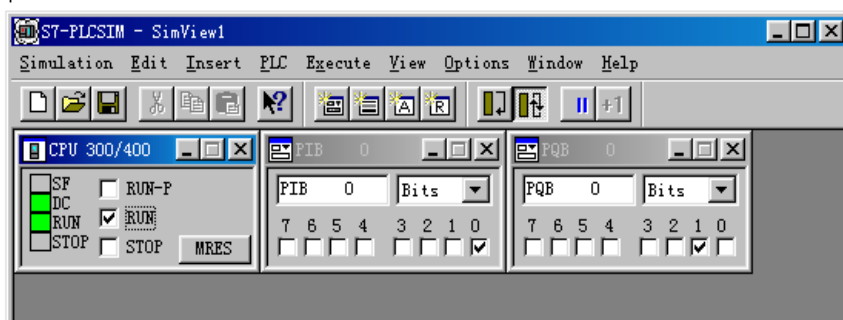
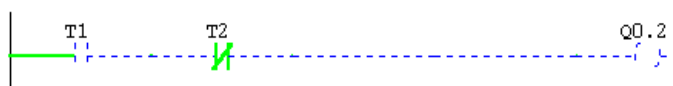


⑥打开程序，监控程序的运行

Network 3 : Title:



Network 4 : Title:



第五章 程序结构

三种编程方法：

（一）线性编程

线性编程将整个用户程序写在一个指令连续的块中，处理器线性地或顺序地扫描程序的每条指令。

（二）分部编程

分部式编程将用户程序分成相对独立的指令块、每个块包含给定的部件组或作业组的控制逻辑。

（三）结构化编程

结构化编程要求用户程序提供一些通用的指令块，以便控制一类相似或相同的部件，给通用指令提供的参数进一步说明各部件的控制差异。为了支持结构化程序设计，STEP 7 将用户程序分类归并为不同的块，根据程序要求，可选用组织块（OB）、功能块（FB）、或功能（FC）

三种类型的逻辑块，而数据块（DB）或（DI）则用来存储执行用户程序时所需的数据。

5.1 数据块及其数据结构

5.1.1 数据块

用户程序运行所需的大量数据或变量存储在数据块中，数据块也是实现各逻辑块之间交换、传递和共享数据的重要途径。

对于 CPU 314，用作数据块的存储器最多 8KB，用户定义的数据总量不能超过这个限制。对于数据块必须遵循先定义后使用的原则。

1. 定义数据块

定义内容包括数据块号及块中的变量（包括：变量符号名，数据类型以及初始值等）。数据块在使用前，必须作为用户程序的一部分下载到 CPU 中。

2. 访问数据块

访问时需要明确数据块号和数据块中的数据类型与位置。根据明确数据块号的不同方法，可以用以多种方法访问数据块中的数据。

（一）直接在访问指令中写明数据块号，如：

```
L DB10.DBW 0
L DB10.DBW 2
XOW
T DB10.DBW 4
```

（二）“先打开后访问”

```
OPN DB 5
L DBW 10
OPN DB 10
T DBW 20
```

由于有两个数据块寄存器（DB 和 DI 寄存器），所以，最多可以同时打开两个数据块。一个作为背景数据块，数据块的起始地址存储在 DI 寄存器中；另一个作为共享数据块，数据块的起始地址存储在 DB 寄存器中。打开背景数据块，在调用 FB 时可以自动实现，由于调用 FB 时使用 DI 寄存器，所以，一般不在 FB 程序中用 OPN DI n 指令打开数据块。

3. 背景数据块和共享数据块

任何 FB、FC 或 OB 均可读写存放在共享数据块中的数据。背景数据块是 FB 运行时的工作存储区，它存放 FB 的部分运行变量。调用 FB 时，必须指定一个相关的背景数据块。作为规则，只有 FB 块才能访问存放在背景数据块中的数据。一般情况下，每个 FB 都有一个对应的背景数据块，一个 FB 也可以使用不同的背景数据块。如果几个 FB 需要的背景数据完全相同，为节省存储器，则可以定义成一个背景数据块，供它们

分别使用。

5.1.2 数据结构

基本数据类型
数据结构形式 复式数据类型

复式数据类型

名称	类型	说明
日期-时间	DATE_AND_TIME	长度 8Byte
字符串	STRING	占用 256Byte
数组	ARRAY	可定义 6 维数组
构造	STRUCT	由多种数据类型组成的数据集合

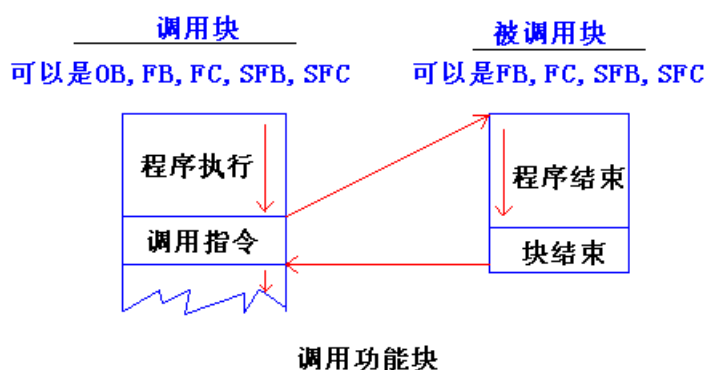
5.1.3 用户数据类型

STEP 7 允许将基本或复式组合成“用户”自己定义的数据类型，这种类型称为用户数据类型（UDT）。用户数据类型必须首先单独建立，并存放在称为 UDT 的特殊数据块中。

建立用户数据类型的目的，是为了将 UDT 作为一种数据类型使用，以方便定义多个结构相同的构造变量。

5.2 功能块编程及调用

一个程序由许多部分（子程序）组成，STEP 7 将这些部分称为逻辑块，并允许块间相互调用。



功能块由两个主要部分组成：一部分是每个功能块的变量声明表，变量声明表声明此块的局部数据；另一部分是逻辑指令组成的程序，程序要用到变量声明表中给出的局部数据。

当调用功能块时，需要提供块执行时要用到的数据或变量，也就是将外部数据传递给功能块，这称为参数传递。

5.2.1 变量声明表（局部数据）

局部数据分为参数和局部变量两大类，局部变量又包括静态变量和临时变量（暂态变量）两种。参数是在调用块和被调用块间传递的数据。静态变量和临时变量是仅供逻辑块本身使用的数据。

局部数据类型

变量名	类型	说明
输入参数	In	由调用逻辑块的块提供数据，输入给逻辑块的指令
输出参数	Out	向调用逻辑块的块返回参数，即从逻辑块输出结果数据
I / O 参数	In_Out	参数的值由调用块的块提供，由逻辑块处理修改，然后返回
静态变量	Stat	静态变量存储在背景数据块中，块调用结束后，其内容被保留
状态变量	Temp	临时变量存储在 L 堆栈中，块执行结束变量的值因被其它内容覆盖而丢失

5.2.2 逻辑块局部数据的数据类型

在变量声明表中，要明确局部数据的数据类型，这样操作系统才能给变量分配确定的存储空间。局部数据可以是基本数据类型或是复式数据类型，也可以是专门用于参数传递的所谓“参数类型”。

参数类型	大小	说明
定时器	2 Byte	在功能块中定义一个定时器形参，调用时赋予定时器实参
计数器	2 Byte	在功能块中定义一个计数器形参，调用时赋予定时器实参
块： Block_FB Block_FC Block_DB Block_SDB	2 Byte	在功能块中定义一个功能块或数据块形参变量，调用时给功能块类或数据块类形参赋予实际的功能块或数据块编号
指针	6 Byte	在功能块中定义一个形参，该形参说明的是内存的地址指针。例如，调用时可给形参赋予实参：P # M50.0，以访问内存 M500.0
ANY	10 Byte	当实参的数据未知时，可以使用该类型

STEP 7 对分配给块局部数据（在变量声明表中）的数据类型（基本、复式、参数）是有一定限制的。

声明类型	基本类型	复式类型	参数类型				
			定时器	计数器	块	指针	
			ANY				
OB 局部变量有效的数据类型							
Temp	可以	可以					可以
FB 局部变量有效的数据类型							
In	可以	可以	可以	可以	可以	可以	可以
Out	可以	可以					
In-Out	可以	可以					
Stat	可以	可以					
Temp	可以	可以					可以
FC 局部变量有效的数据类型							
In	可以	可以	可以	可以	可以	可以	可以
Out	可以	可以				可以	可以
In-Out	可以	可以				可以	可以
Temp	可以	可以					可以

STEP 7 允许用物理地址（绝对地址）或符号地址或常数等形式作为实参给形参赋值，但对于不同的形参类型，STEP 7 限制了赋值形式。

声明类型	绝对地址	符号地址	块局部符号	常数
基本数据类型				
In	可以	可以	可以	可以
Out	可以	可以	可以	不可
In-Out	可以	可以	可以	不可
复式数据类型				
In	不可	可以	可以	不可
Out	不可	可以	可以	不可
In-Out	不可	可以	可以	不可

5.2.3 功能块编程与调用举例

对功能块编程分两步进行:第一步工作是定义局部变量（填写局部变量表）；第二步是编写要执行的程序。

写功能块程序时，可以两种方式使用局部变量：①使用变量名，此时变量名前缀“#”，以区别于在符号表中定义的符号地址，增量方式下，前缀会自动产生；②直接使用局部变量的地址，这种方式只对背景数据块和 L 堆栈有效。

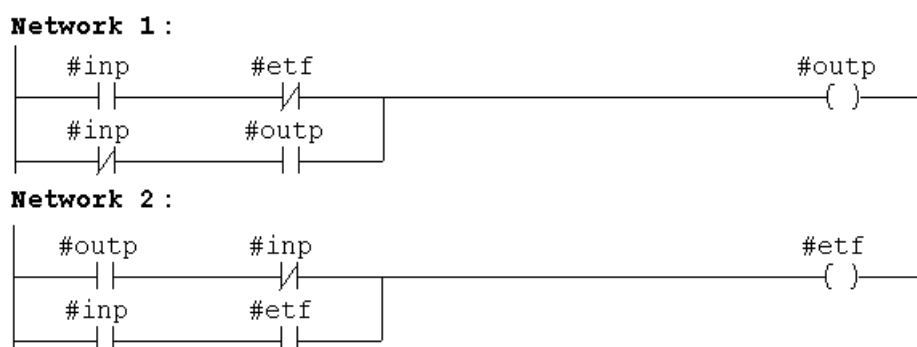
例一 二分频器

二分频器是一种具有一个输入端和一个输出端的功能单元, 输出频率为输入频率的一半。

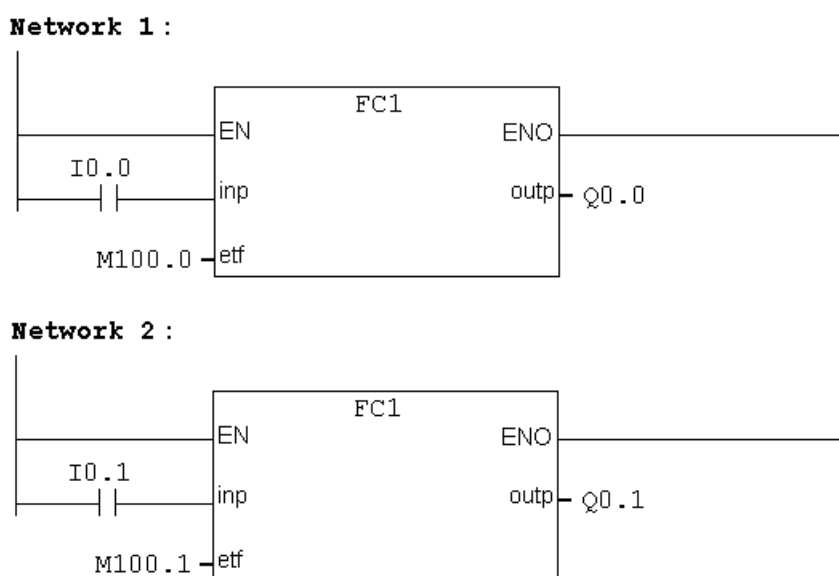
(1) FC1 的变量声明表

Address	Decl.	Name	Type	Initial Value
0.0	in	inp	BOOL	
2.0	out	outp	BOOL	
4.0	in_out	etf	BOOL	

(2) FC1 梯形图程序



(3) OB1 调用梯形图程序



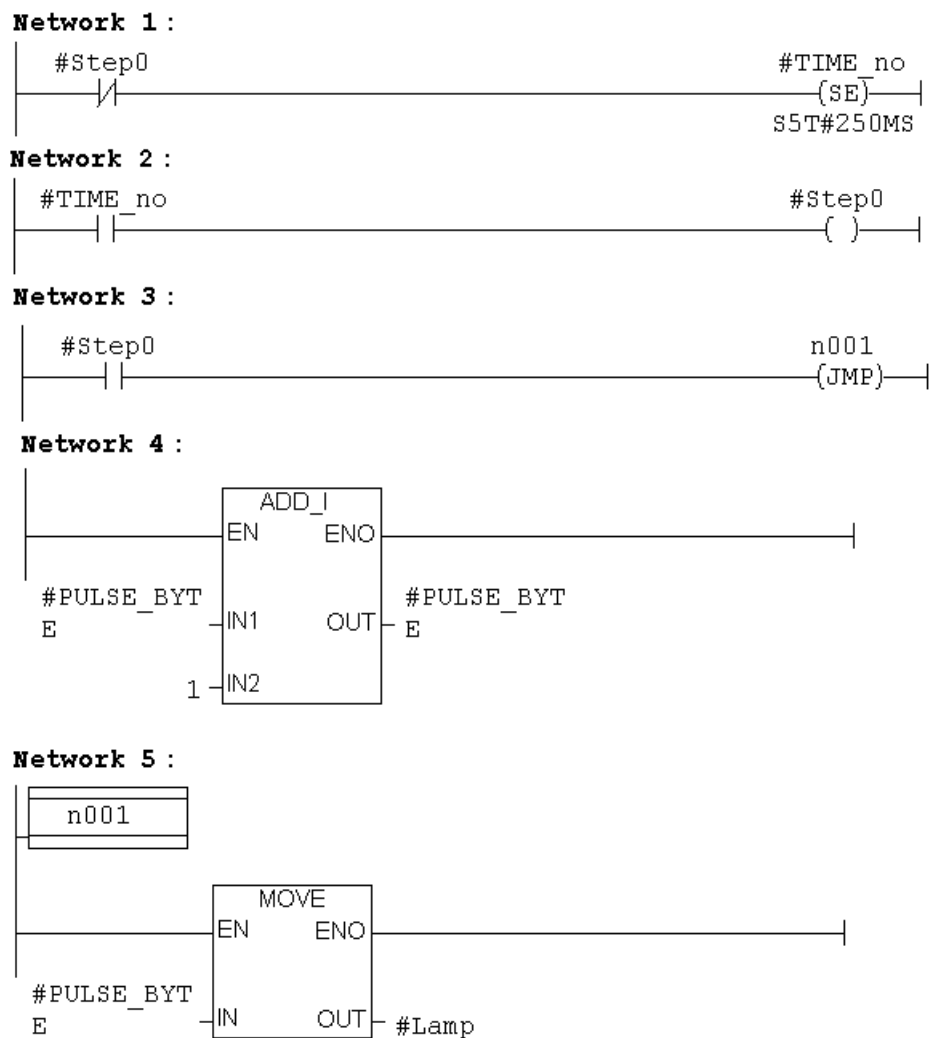
例二 时钟脉冲发生器

使用定时器实现自由设定时钟脉冲发生器功能(脉冲占空系数 1:1)。

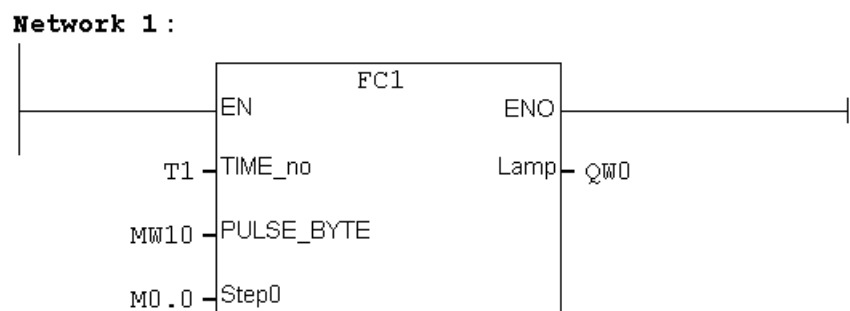
(1) FC1 变量声明表

Address	Decl.	Name	Type
0.0	in	TIME_no	TIMER
2.0	out	Lamp	WORD
4.0	in_out	PULSE_BYTE	INT
6.0	in_out	Step0	BOOL

(2) FC1 梯形图程序

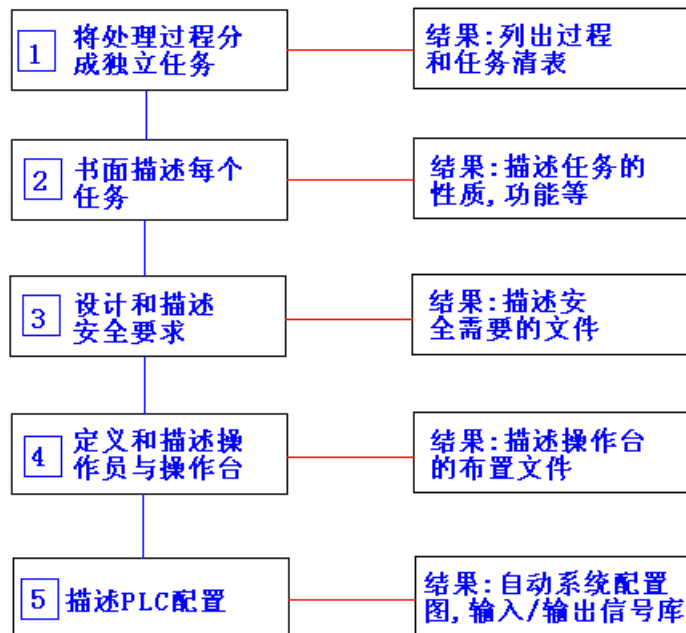


(3) OB1 调用梯形图程序



第六章 可编程序控制器应用设计

PLC 系统设计步骤:



MPI/PROFIBUS 地址

为了使所有节点能相互通讯,必须在组网前为它们分配地址:

- 在 MPI 网络中,每一个节点分配一个“MPI 地址”以及一个“最高 MPI 地址”。
- 在 PROFIBUS 网络中,每一个节点分配一个“PROFIBUS 地址”以及一个“最高 PROFIBUS 地址”。

通过编程器(某些 PROFIBUS-DP 从站必须使用在从站上的开关)给每个节点单独分配 MPI/PROFIBUS 地址。

注意

不能给 RS 485 中继器分配一个“MPI 地址”或“PROFIBUS 地址”。

表 5-1 允许的 MPI 和 PROFIBUS 地址。

MPI 地址	PROFIBUS 地址
0—126	0—125
为以下设备保留: 0 为 PG 1 为 OP 2 为 CPU	为以下设备保留: 0 为 PG

MPI/PROFIBUS 编址的规则

在分配 MPI/PROFIBUS 地址之前，要遵守以下规则：

- 在 MPI/PROFIBUS 网络上各节点的地址必须是不同的。
- 允许的最高 MPI/PROFIBUS 地址必须是 $\geq$ 实际的最大 MPI/PROFIBUS 地址，并且对所有节点应该是相同的(例外：将一台编程器接到几个节点，见 6.3.2 节)。

在连接网络的节点时，必须遵循以下规则：

- 在网络的各个节点互连以前，必须为每一个节点分配 MPI 地址和最高的 MPI 地址或 PROFIBUS 地址和最高 PROFIBUS 地址(RS 485 中继器除外)。
提示：在网络中所有节点的外壳上标明其地址，这样用户始终能看到系统中哪些节点已分配了哪些 MPI 地址，为此 CPU 的包装中附有地址标签条。
- “排队”连接 MPI 网络中的所有节点，这样使直接连在 MPI 网络中的编程器和操作员面板都被考虑到。

只在起动或维护时才使用的那些编程器/OP 用分支线将它们连接在网络上。

- 如网络上运行的节点多于 32 个，必须通过 RS 485 中继器以连接网络上的段(Segments)。

所有 PROFIBUS 网络上的段，在其间必须至少有一个主站和一个从站。

- 通过 RS 485 中继器连接非接地的总线段和接地的总线段(见模板规范参考手册中 RS 485 中继器的描述)。
- 每个 RS 485 中继器会减少该总线段的最大节点数，就是说，在一个总线段安装有一个 RS 485 中继器，则该段只能连接最多 31 个节点。但 RS 485 中继器的数量不影响总线的最大节点数。在一个队列中最多可安装 10 个 RS 485 中继器。
- 在段的第一个节点和最后一个节点接入终端电阻。
- 在将一个新的节点接入网络以前，必须切断该节点的供电电压。

MPI 网络举例

图 5-4 为按照上述规则组建的 MPI 网络。

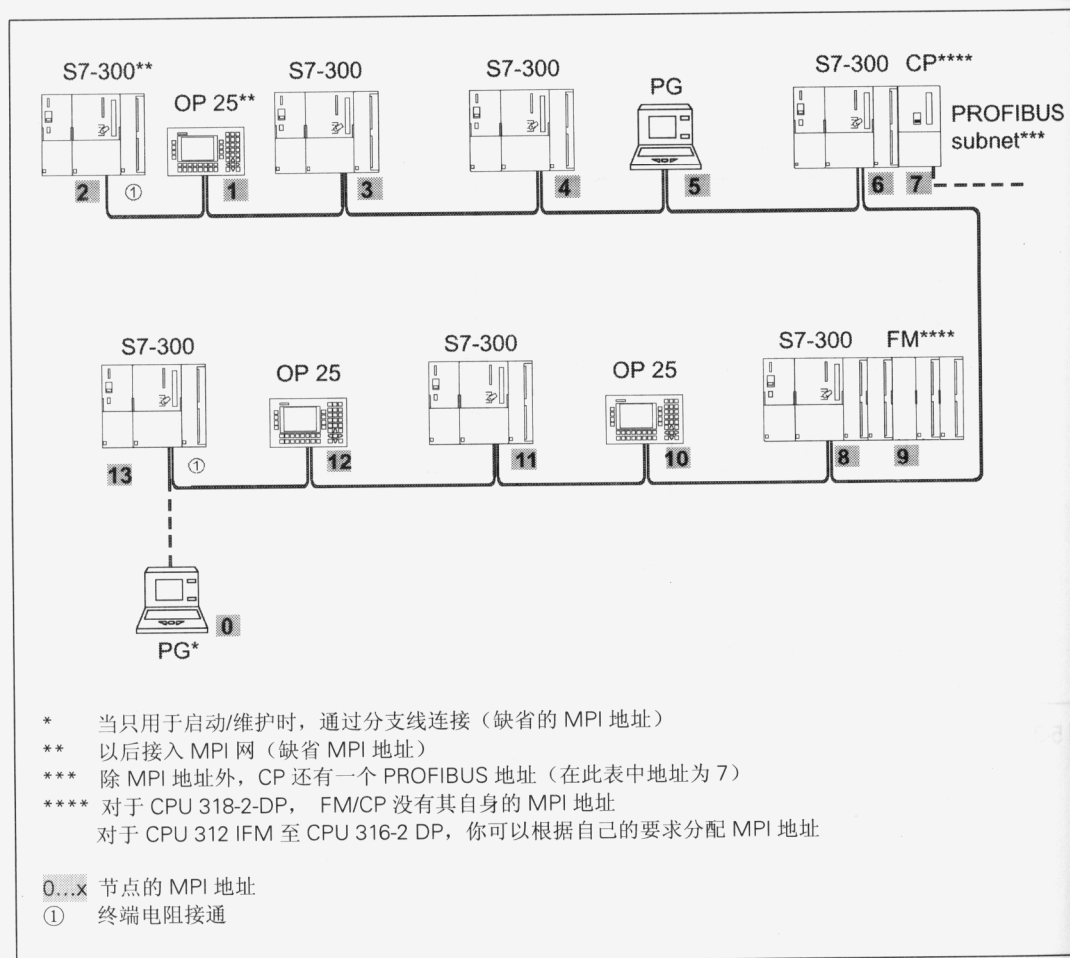


图 5-4 一个 MPI 网络的举例

PROFIBUS 网络举例

图 5-5 为按照上述规则组建的 PROFIBUS 网络。

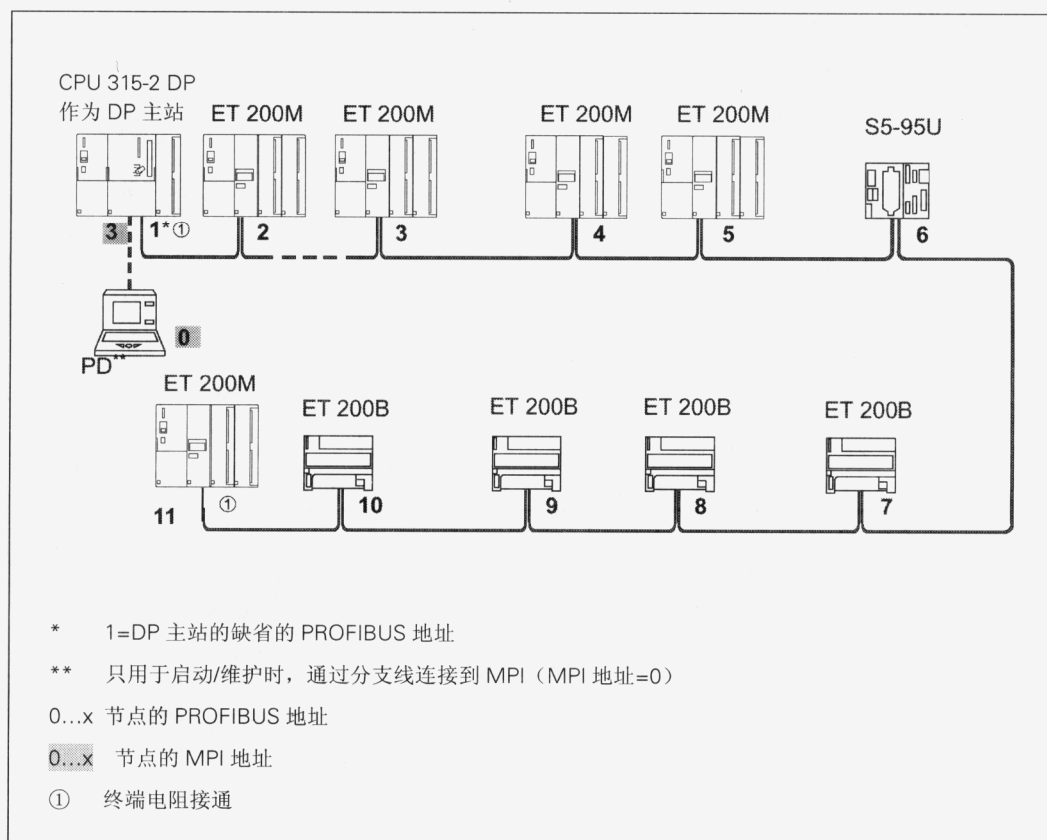


图 5-5 PROFIBUS 网络举例