考点分布

一、上机操作题

1、做法：

1）创建保存文件（10分）

2）接线绘制IO分配表（10分）

3）修改程序增加功能（20分）

4）验证功能或画出波形或状态转移图，需验收（30分）

二、上机操作题

2、考点：

1）自锁互锁、定时器、计数器、边沿指令

2）功能指令（数据移动、数码管、移位、比较等）

3）流水灯、闪烁灯、交通灯、电机控制、数码管、模拟量采集

4）经验设计法、顺序功能图法、混合

三、上机操作题

举例：

1）数码管倒计时的同时，灯组顺时针流水。

2）课件的交通灯，显示每个阶段的时间倒计时

3）步进电机的正反转控制、速度切换

4）模拟量显示

四、设计题（30分）

举例：

1）一键多用，控制流水的模式

2）人行道交通灯

3）步进电机控制

4）数码管实现流水灯

题目类型

1、上级操作题（70分）

2、设计题 （30分）

开卷机试

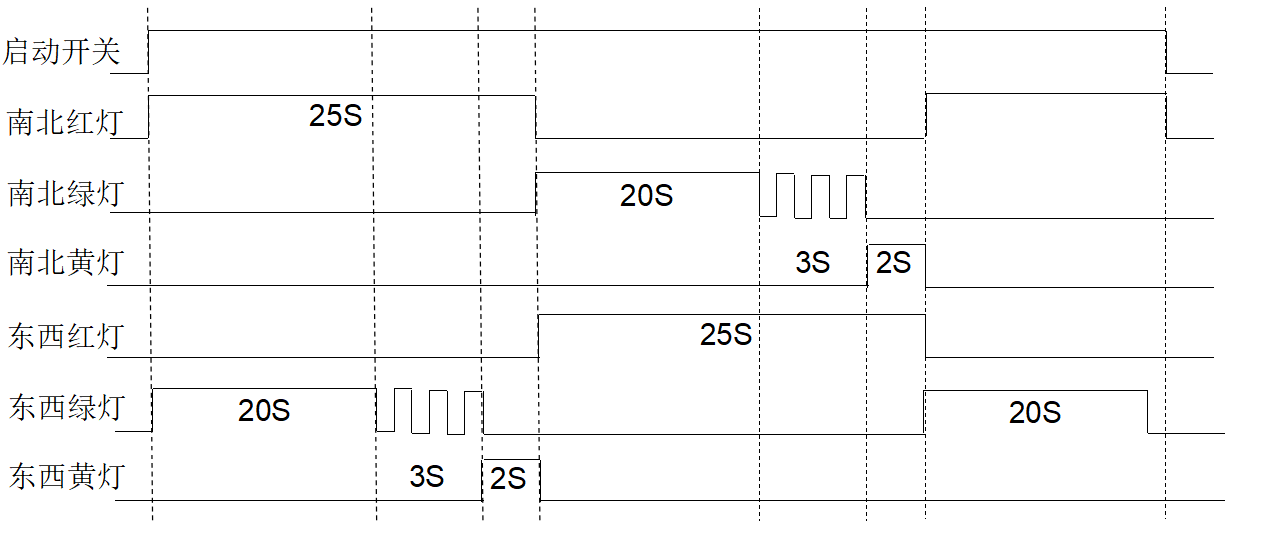
允许携带：

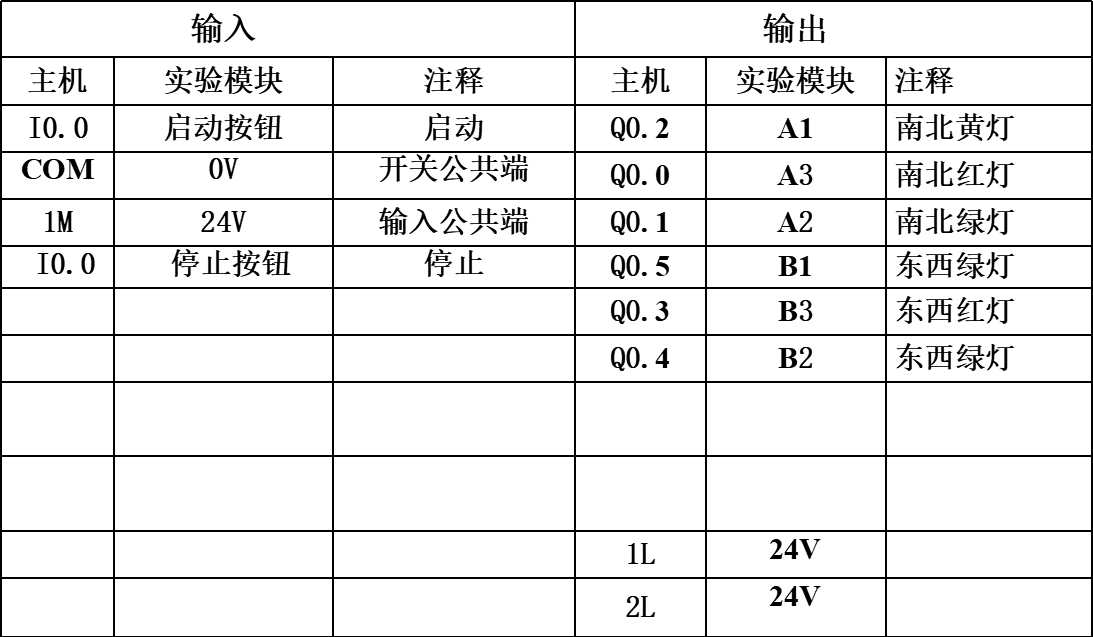
课本

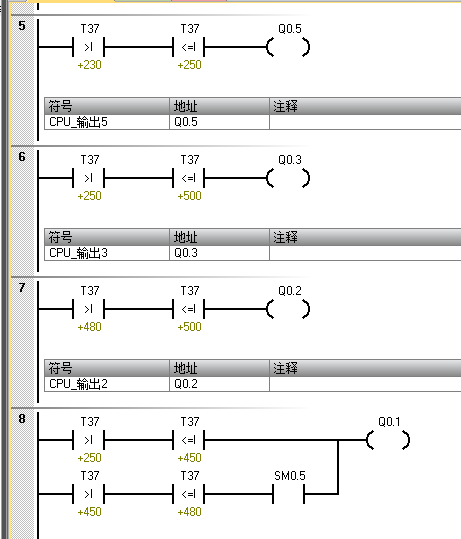
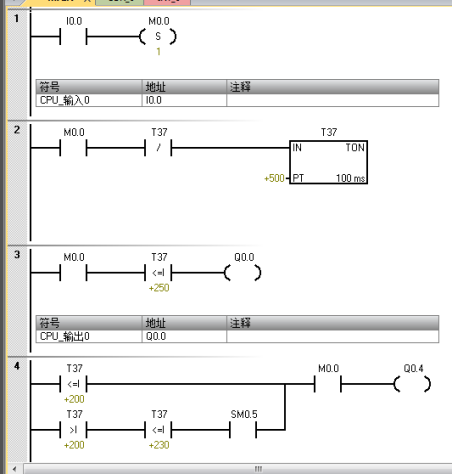
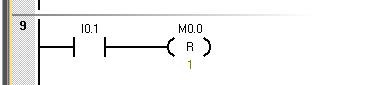
20页以内A4大小纸质资料（装订）

**（ 交通灯比较指令）**

信号灯受一个启动开关控制，当启动开关接通时，信号灯系统开始工作，且先南北红灯亮，东西绿灯亮。当启动开关断开时，所有信号灯都熄灭。南北红灯亮维持25秒。东西绿灯亮维持20秒。到20秒时，东西绿灯闪亮，闪亮3秒后熄灭。在东西绿灯熄灭时，东西黄灯亮，并维持2秒。到2秒时，东西黄灯熄灭，东西红灯亮，同时，南北红灯熄灭，绿灯亮。

东西红灯亮维持25秒。南北绿灯亮维持20秒，然后闪亮3秒后熄灭。同时南北黄灯亮，维持2秒后熄灭，这时南北红灯亮，东西绿灯亮，周而复始。





二、**步进电机**

要求：1）连接步进电机各相至PLC的输出接口Q0；

2）使用一个按键（非拨码开关）作为系统运行的启动条件，另一个按键（非拨码开关）作为系统运行的停止条件，连接至PLC的输入接口I0；

3）连接数码管至PLC的输出接口Q1。

要求：1）系统启动时，数码管显示0，电机不转；

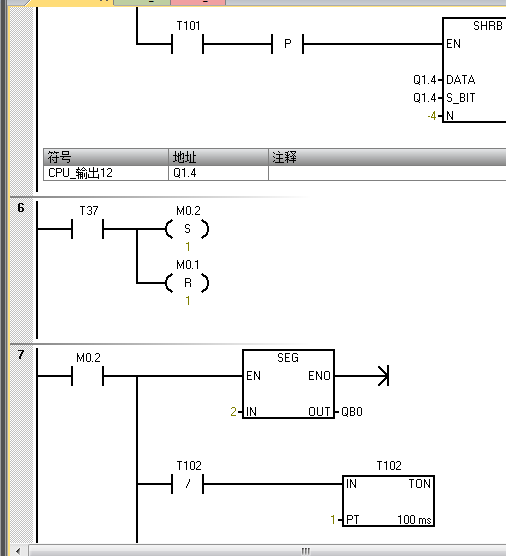
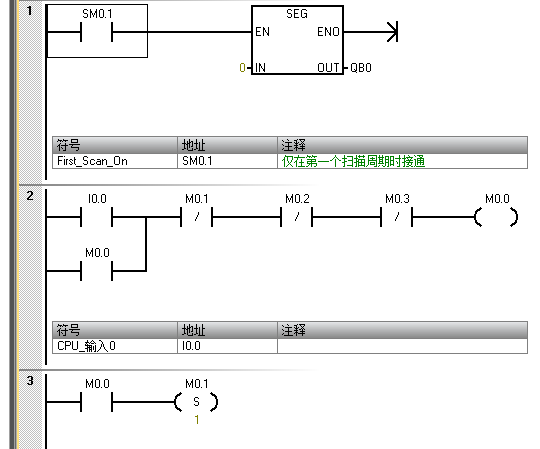
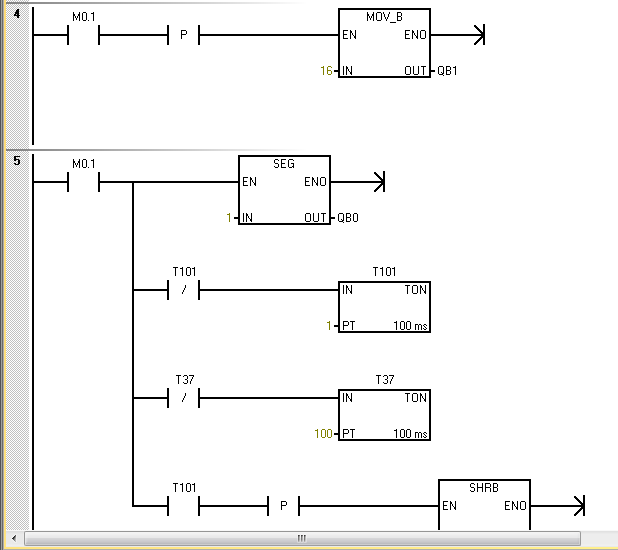
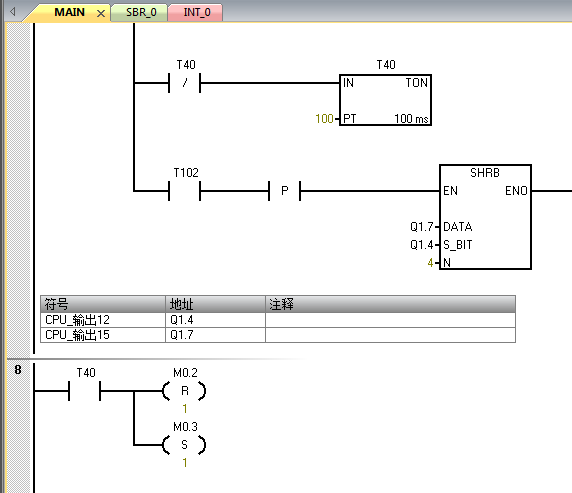
2）按下启动按钮，电机开始顺时针旋转，数码管显示1；

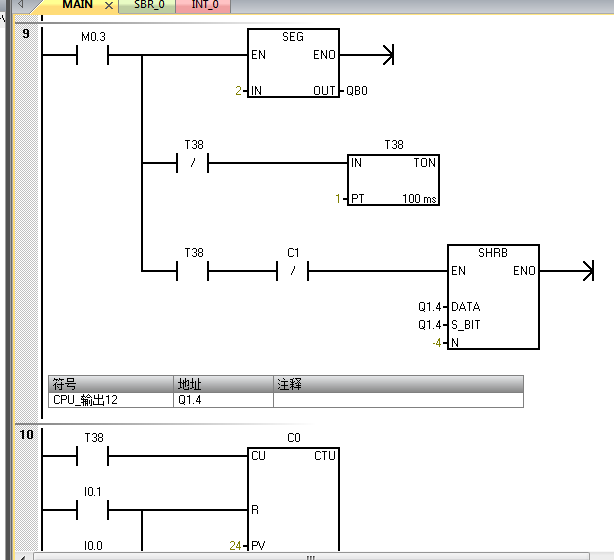
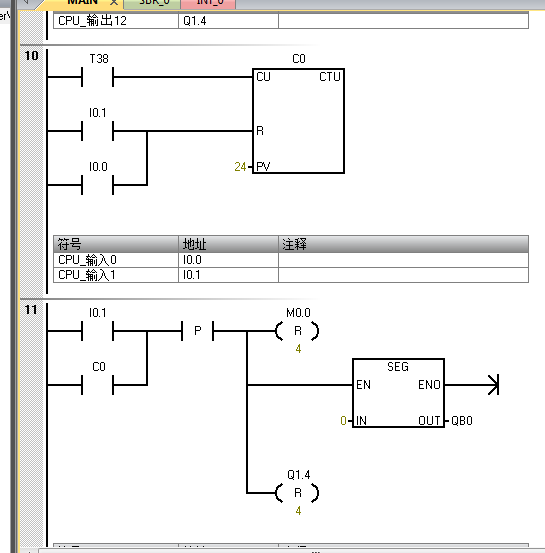
3）10秒钟后，电机逆时针旋转，数码管显示2；

4）10秒钟后，电机顺时针旋转，转1圈后停止，数码管显示0；

5）任何时刻按下停止按钮，电机停止，数码管显示0。

说明：1）启动按钮按下一次后，在电机停止之前再次按下时此按键无效；

1. 步进电机正转和反转的速度可以一样，每一步走动时间小于0.1S。



程序段1:启动PLC数码管显示零 程序段2:启动按钮，启动顺序控制各个M继电器

程序段3:M0.1继电器置位（持续接通得电） 程序段4:用上升沿触发Q1.4～Q1.7

程序段5:因M0.1得电，七段译码指令赋值1，数码管显示1，T101定时器产生0.1s的脉冲

T37用于定时10s，位移寄存器SHRB使步进电机逆转。  
程序段6:T37定时时间到接通，使M0.2得电，M0.1复位(断电)执行下一步。  
程序段7，M0.2接通，数码管显示2，T40定时10s，位移寄存器使电机顺时针转  
程序段8:T40定时到，复位M0.2，置位M0.3执行下一步  
程序段9:数码管显示2，电机逆时针转  
程序段10:用于计数器C1计数电机走的步数，24步转一周，T38触点用于产生计数脉冲I0.0、 I0.1使计数器计数复位

程序段11:当C1计数到使M0.0到M0.3复位、数码管显示零、Q1.4～Q1.7复位，电机停止!I0.1为停止按钮。

**I/O分配表**

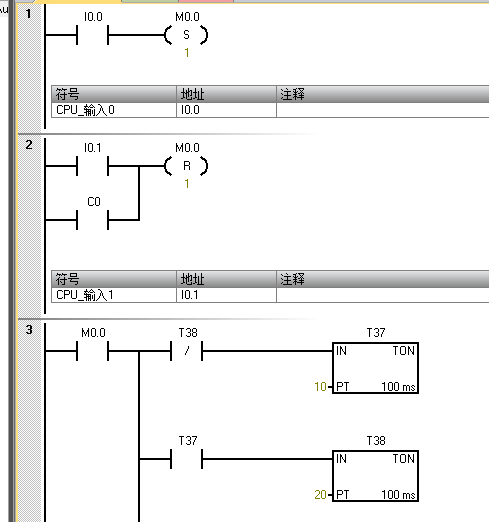
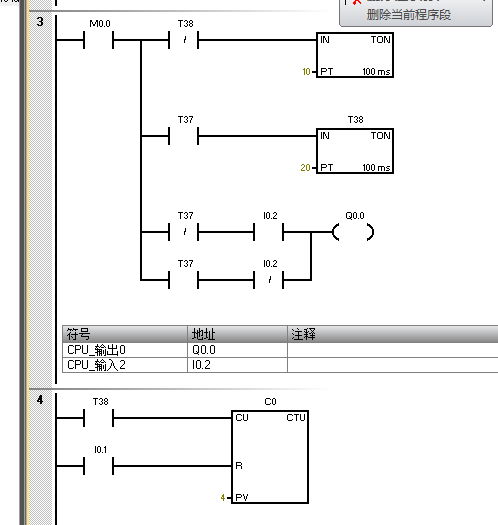


**三、1秒亮,2秒灭**

1）I0.0启动（按钮形式）Q0.0进行1S亮，2S灭的闪烁，I0.1停止（按钮形式）Q0.0的闪烁。

2）I0.2（开关形式）改变闪烁的方式为1S灭，2S亮。

3）闪烁4次后结束。



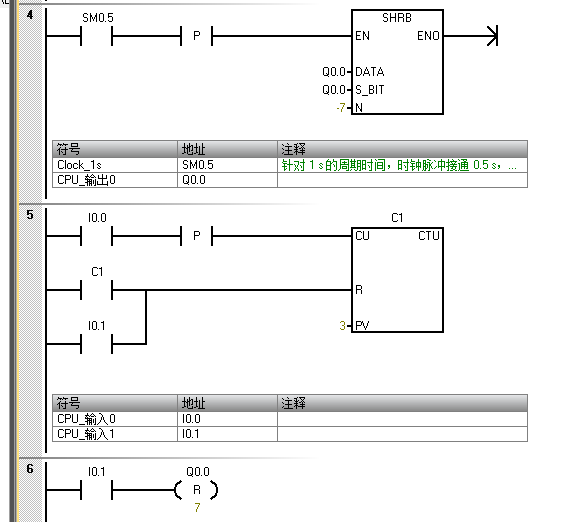
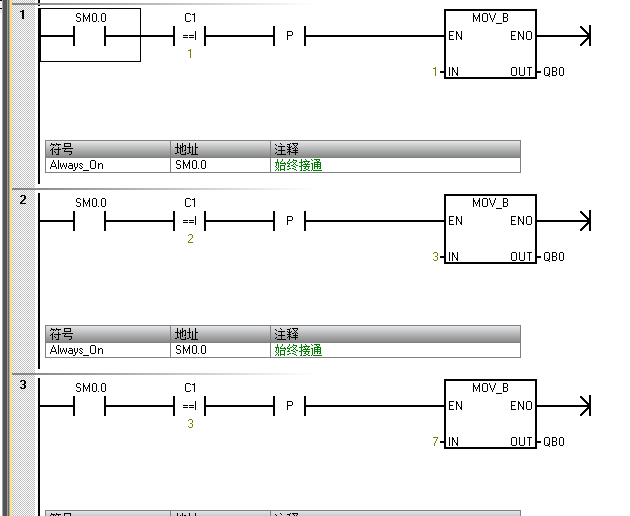
**I/O分配表**



程序段1:启动(I0.0)、M0.0得电。  
程序段2:停止(I0.1)、C0为计数器触点用于复位M0.0  
程序段3:T37定时10s，T38定时20s，I0.2(开关)改变闪烁方式1s灭，2s亮  
程序段4:计数器计四次T38产生的脉冲，四次后使M0.0复位，停止按钮I0.1使计数器复位

1. **一键多用**

用一个按钮控制流水灯，使其控制亮一盏、两盏、三盏灯之间切换

****

程序段1:启动PLC，当按下启动按钮I0.0加计数器加一，比较指令比较计数器当前值，当符合条件比较指令接通，MOB\_B赋值1，QB0的Q0.0得电，转程序段4位移寄存器开始以1s的时间间隔倒序流水

程序段2:当再次按下I0.0计数器加一，切换同时亮两盏灯流水，上段程序条件不符合断电

程序段3:同程序段2，依此类推

程序段4:用位移寄存器实现流水功能(如果想顺序流水修改位移寄存器的DATA为Q0.6，-7改7，即可

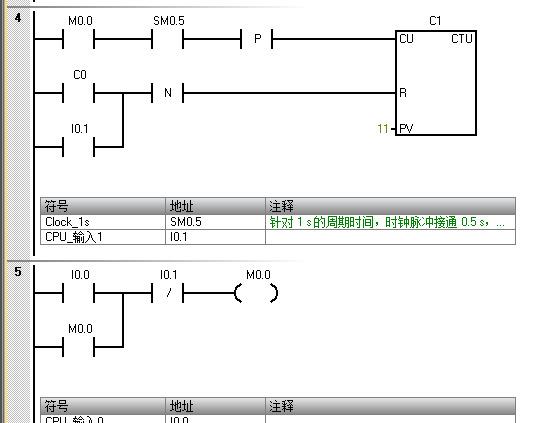
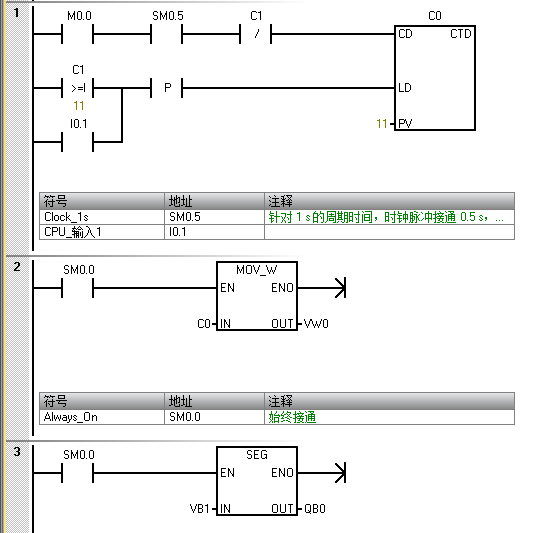
程序段5:I0.0用于启动，并使计数器加一，C1触点当计数为3时复位,I0.1使计数器复位

程序段6:停止按钮I0.1使Q0.0到Q0.6复位断电

**I/O分配表**



**五、晶体管10~0循环**



程序段1:以1s为周期产生一个减数计数脉冲，定计数为11  
程序段2:将计数器当前值送到VW0  
程序段3:将VB1数据送到七段译码指令，使数码管和减计数器同步  
程序段4:计数器C1用于复位计数器C0，两者相互关联使程序自动循环  
程序段5:用于启动和停止整个程序

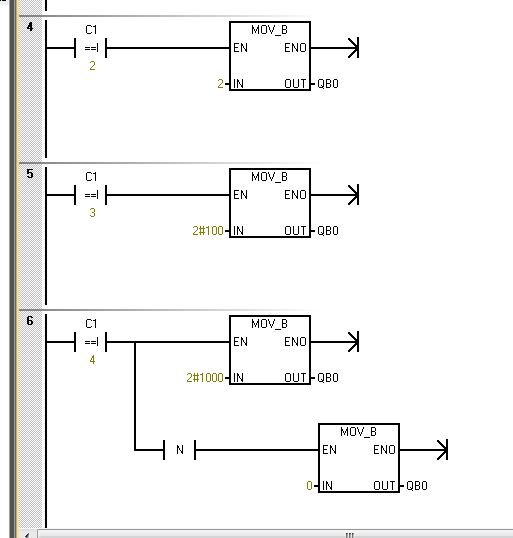
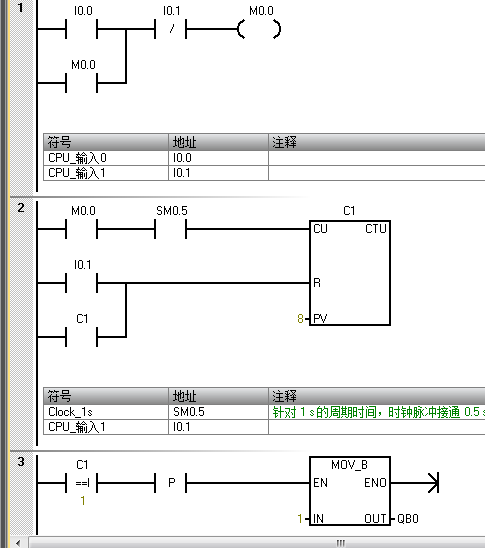
**I/O分配表**

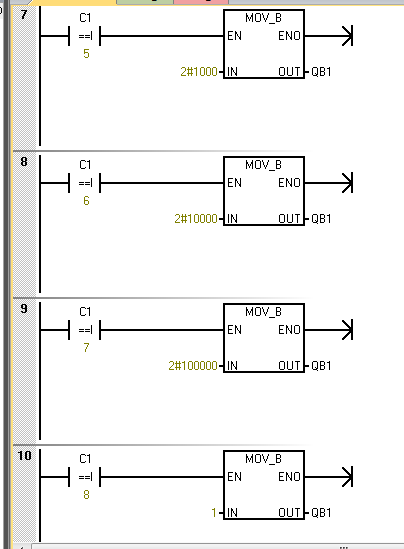
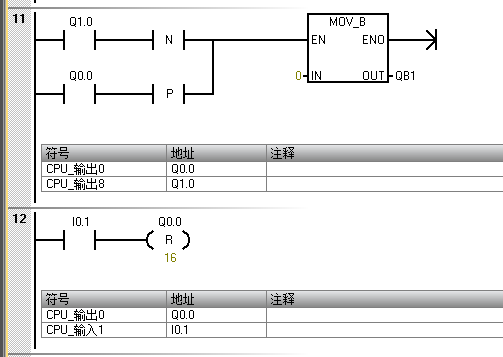
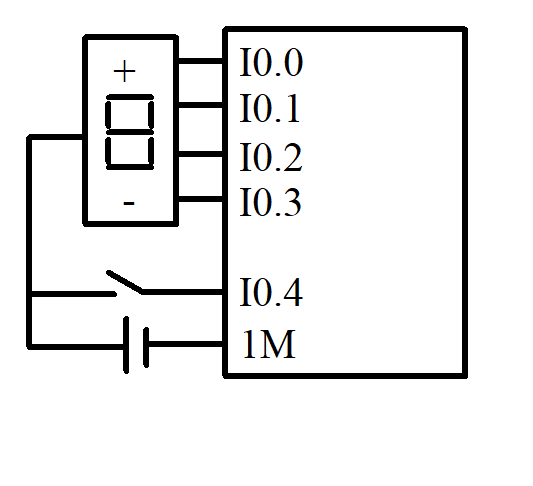


**（如修改9到1循环，只用修改有11的值改为10，将程序段4的下降沿改为上升沿即可）**

**六、双晶体管循环**

数码管实现流水灯



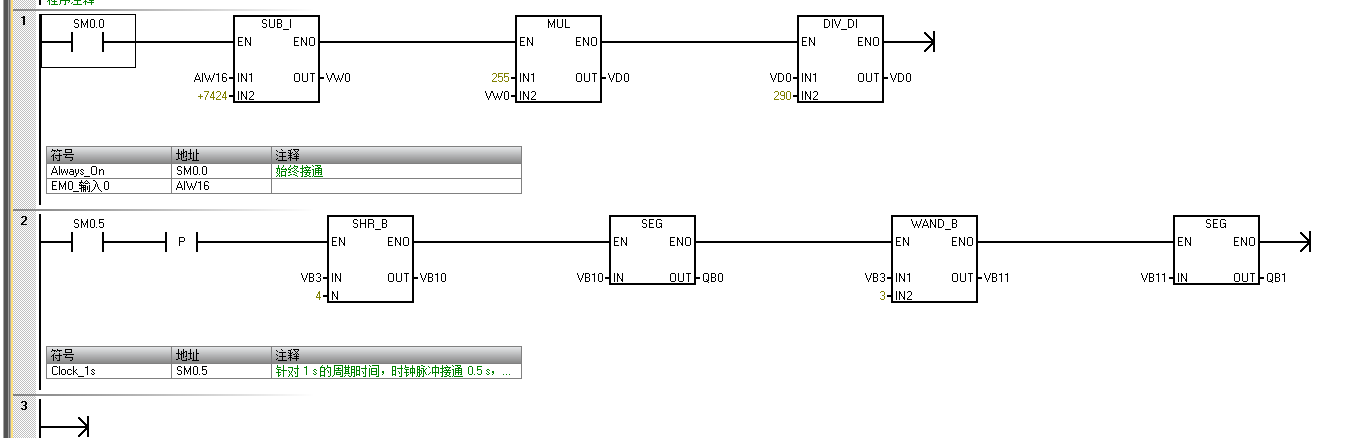


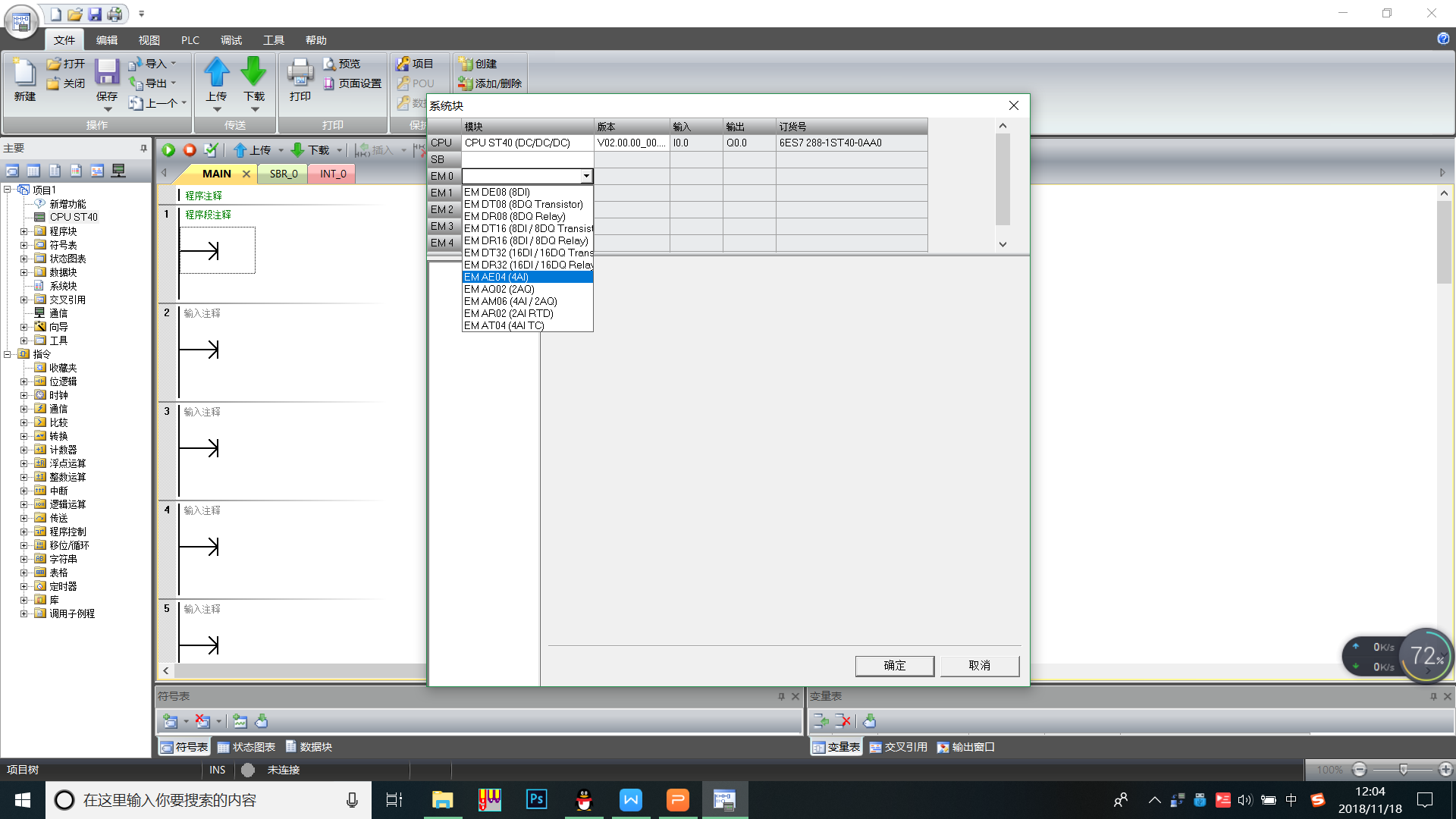
程序段1:启动程序  
程序段2:用于产生1s的时钟脉冲，并循环往复  
程序段3～5:当计数器开始计数时数右码管单段点并开始以1s为周期顺序流水  
程序段6:切换左第四段数码管，复位右数码管  
程序7～10:同上流水点亮  
程序段11:复位左数码管段，启动右数码管段，实现循环  
程序段12:停止按钮并复位Q0.0～Q1.7

**I/O分配表**



1. **模拟量（压力传感器）**

****



（注:通道要选用上图所示，默认通道零即可

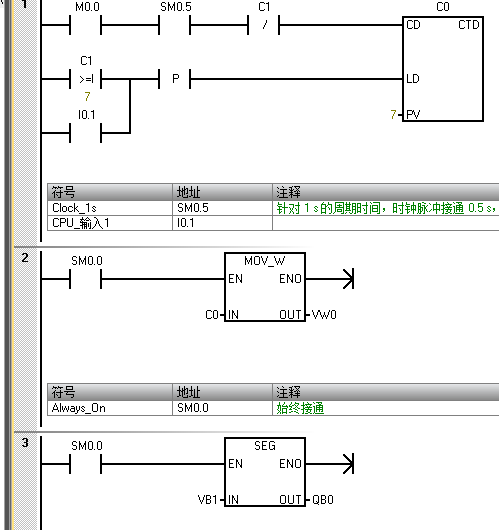
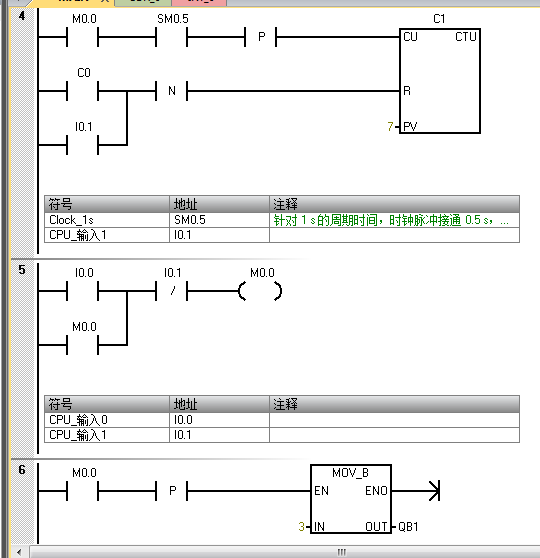
程序段1:SUB\_I的7424值要修改为和AIW16的值一样，使压力传感器输出值归零

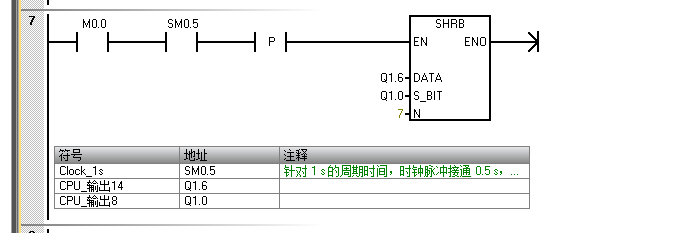
其他不再解释）

**I/O分配表**



1. **晶体管6~0循环加艺术灯流水**

****

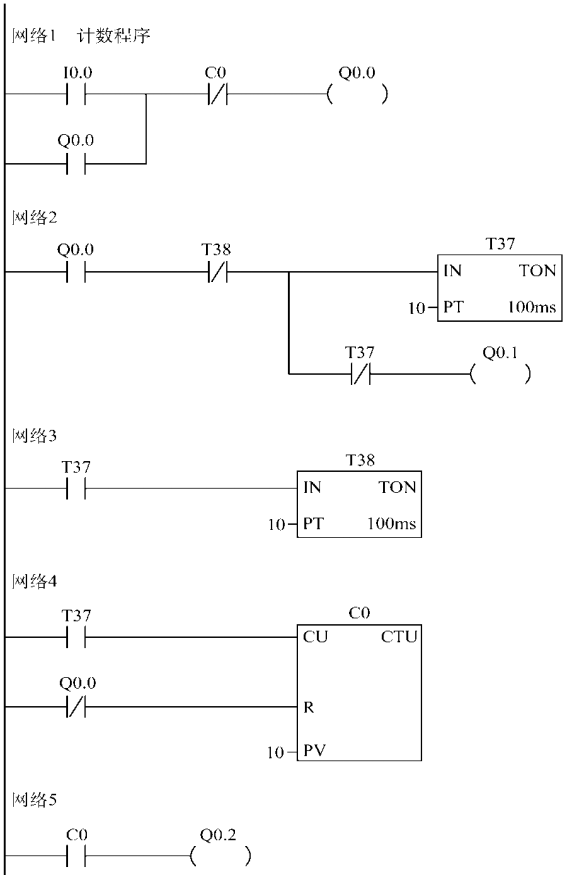
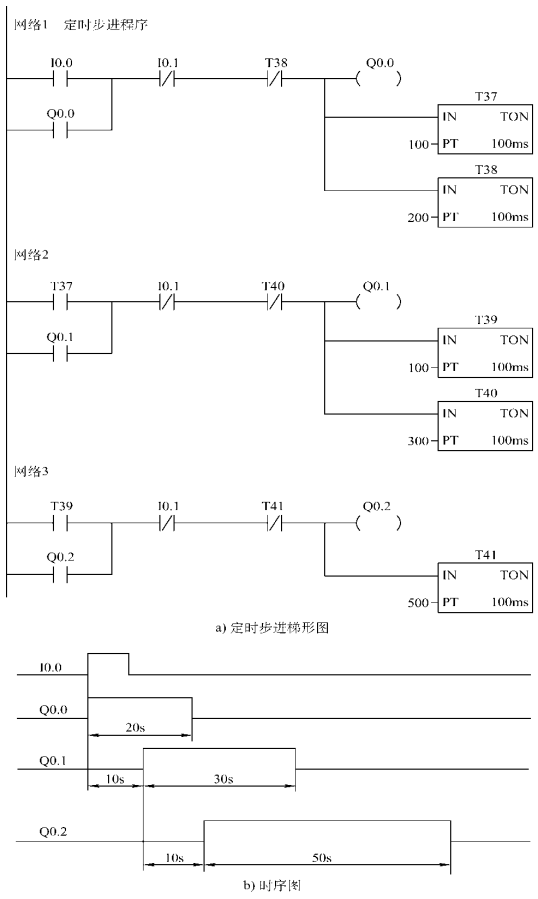
****

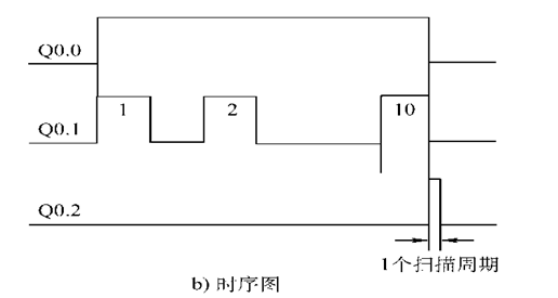
（该程序为数码管倒计时循环和流水功能修改，主要加上了艺术灯顺时流水，功能实现了解上步程序四、五）修改程序段6的3值为1、3、7可实现一、二、三盏同亮循环

**I/O分配表**

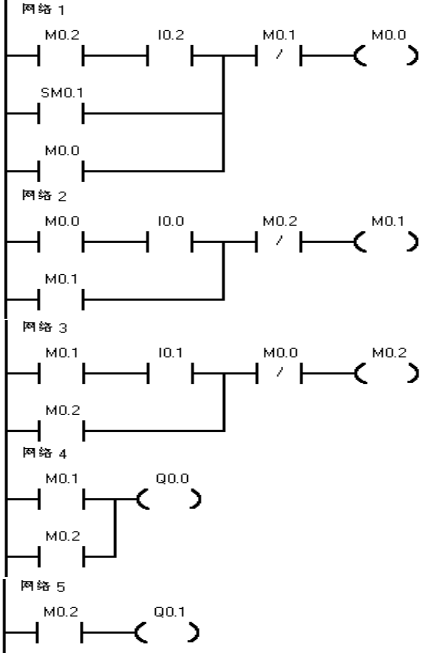


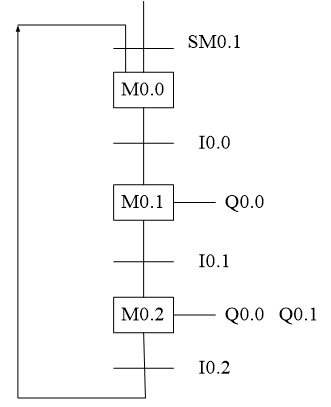
**例题1 例题2**

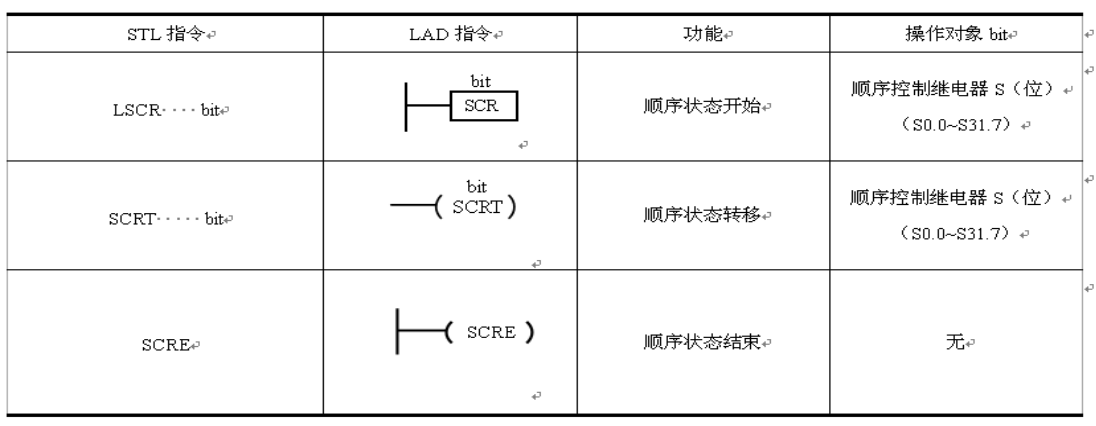
****



**1、基于起保停电路的顺序控制设计方法**

**例：根据图所示的顺序功能流程图，设计出梯形图程序。**



1. **顺序控制指令**

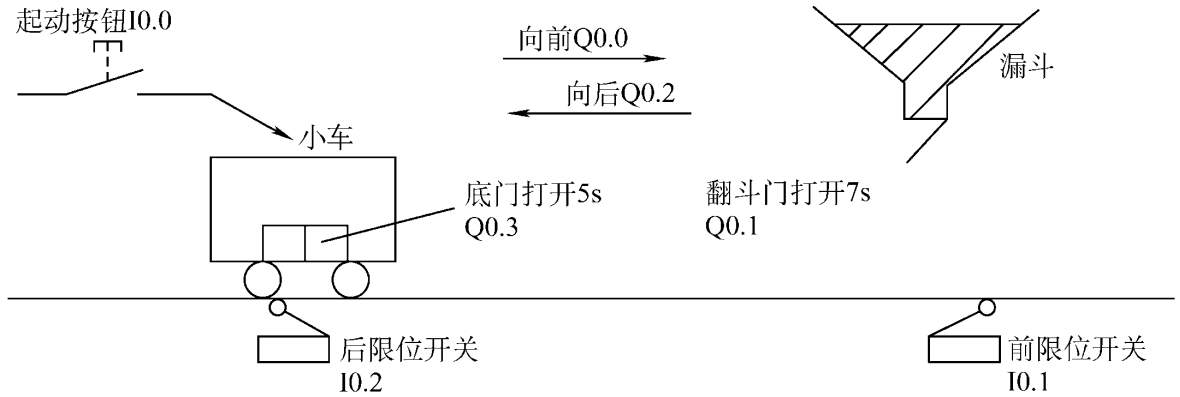
**LSCR指令（在前）：为功能图中一个状态的开始。**

**SCRE指令（在后）：为这个状态的结束。**

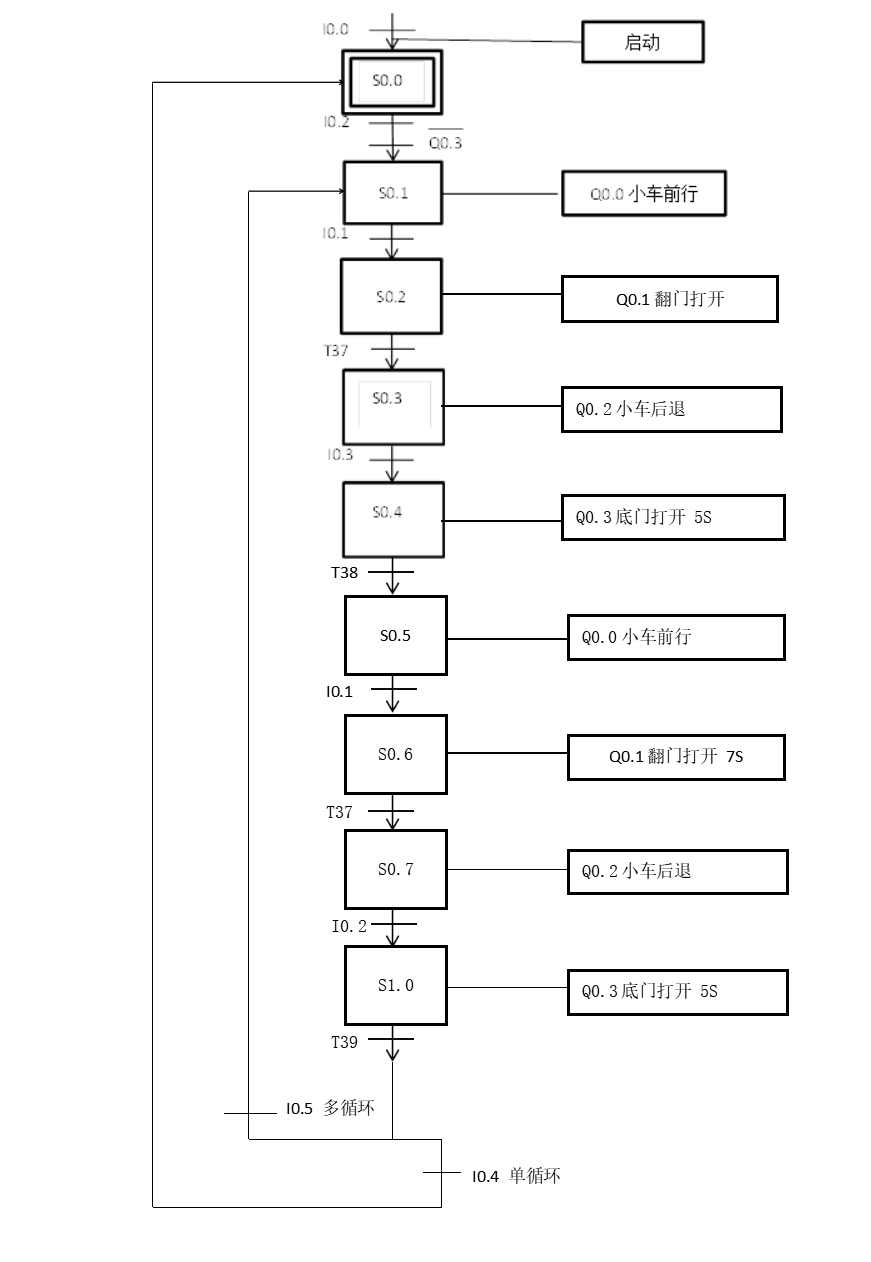
**LSCR指令操作对象bit为顺序控制继电器S中的某个位（范围为S0.0~S31.7），当某个位有效时，激活所在的SCR段。S中各位的状态用来表示功能图中的一种状态。**

**顺序状态转移指令SCRT：**

**该指令操作数bit置位激活下一个SCR段的状态，使下一个SCR段开始工作，同时使该指令所在段停止工作，状态器复位。**



I0.3



**比较指令与数据传送指令**

1．字节、整数、双整数和实数比较指令

比较触点中间的B、I、D、R、S分别表示无符号字节、有符号整数、有符号双整数、有符号实数和字符串比较。满足比较关系式给出的条件时，比较指令对应的触点接通。字符串比较指令的比较条件“x”只有==和<>。

整数和双整数比较指令比较两个有符号数。IN1在触点的上面，IN2在触点下面。

比较指令是通过取指令LD、逻辑与指令A、逻辑或指令O操作码分别加上数据类型符号B（字节）、I（W）（整数）、D（双整数）、R（实数）进行组和实现编程的。

比较运算符有以下6种。

== （比较IN1是否等于IN2）

＜＞ （比较IN1是否不等于IN2）

> （比较IN1是否大于IN2）

< （比较IN1是否小于IN2）

>= （比较IN1是否大于等于IN2）

<= （比较IN1是否小于等于IN2）

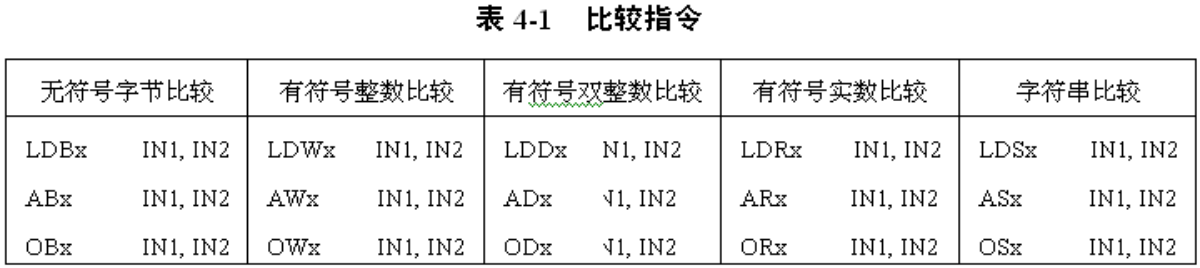
比较的数据类型应该相等。

字节型：Ｂ　　　　　0~255，-128~127

　　　整　型：Ｉ／Ｗ -32768~+32767

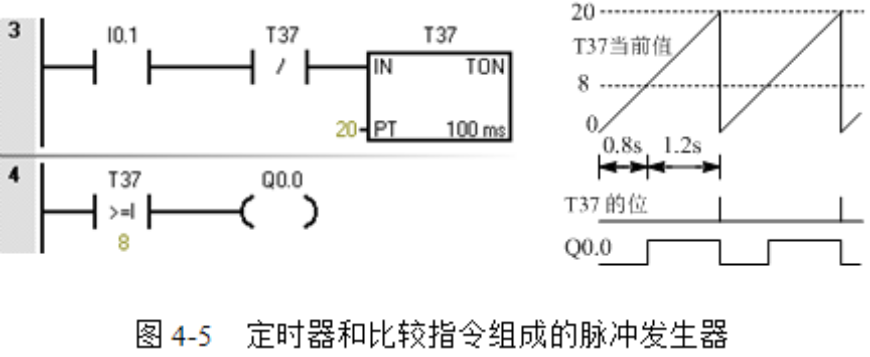
　　　双整数：Ｄ -2G~+2G

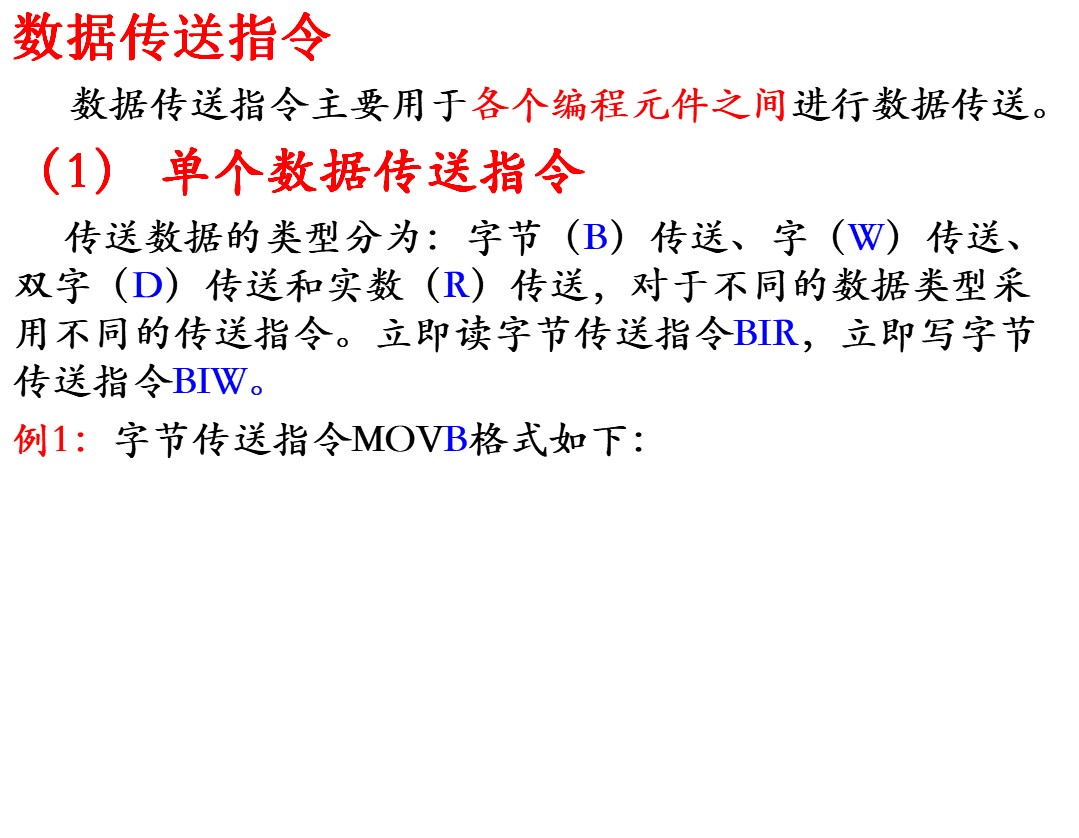
　　　浮点型：Ｒ -1.175495E-38~~+3.402823E+38

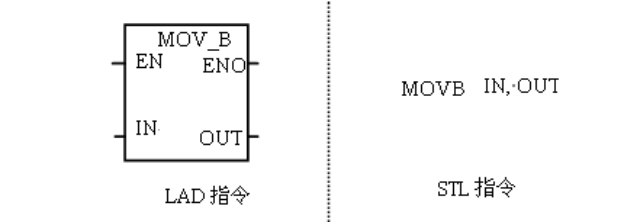


【例】 用接通延时定时器和比较指令组成占空比可调的脉冲发生器。

T37的常闭触点控制它的IN输入端，使T37的当前值按锯齿波变化。比较指令用来产生方波，Q0.0为OFF的时间取决于比较指令第2个操作数的值。

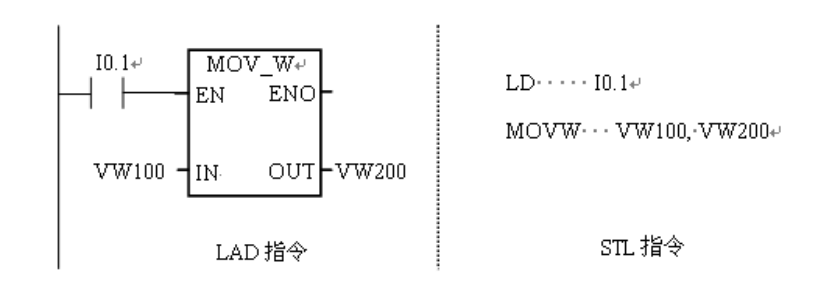
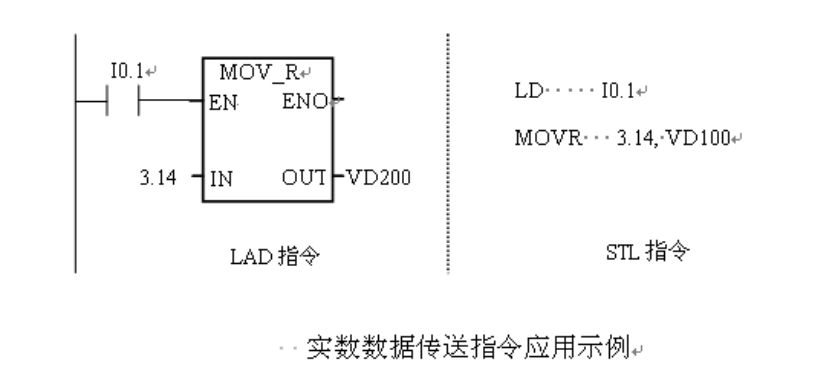






例1：上图（在I0.1为ON时）将VW100中的字数据传送到VW200中；

下图（在I0.1为ON时）将常数（实数）3.14传送到双字单元VD200中；

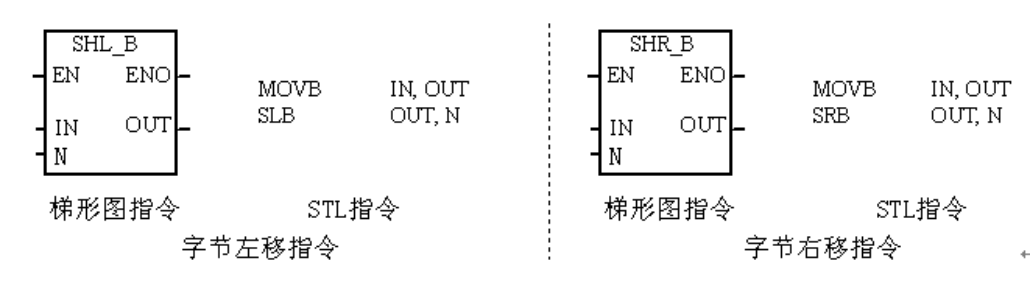


**移位指令**

移位指令的作用是对操作数按二进制位进行移位操作，移位指令包括：左移位、右移位、循环左移位、循环右移位以及移位寄存器指令。

（1）左移和右移指令

左移和右移指令的功能是将输入数据IN左移或右移N位，其结果送到OUT中。移位指令分字节、字、双字移位指令，其指令格式类同。



字节操作为无符号操作，有符号数的符号位同样被移动；

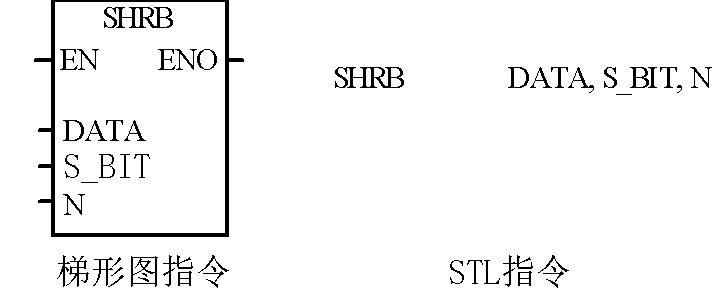
移出与SM1.1相连，移入补0；

移位次数N为字节型，N大于数据长度以实际长度为准；

影响SM1.0（结果为0）SM1.1（溢出）SM1.4（不正常态）。

**移位寄存器指令**

移位寄存器指令又称自定义位移位指令，可以由用户在指令数据部分设置移位寄存器的起始位和最高位。



最低位地址S\_BIT；最高位地址MSB+S\_Bit；

N的绝对值≤64，数据类型无字节型、字型、双字型之分;

如果N﹥0，正向移位，自低到高，最高位移出；

如果N﹤0，反向移位，自高到低，最低位移出；

移出端至SM1.1。

下图中的14位移位寄存器由V30.0～V31.5组成，在I0.3的上升沿，I0.4的值从移位寄存器的最低位V30.0移入，寄存器中的各位左移一位，最高位V31.5的值被移到溢出标志位SM1.1。 N为−14时，I0.4的值从最高位V31.5移入，寄存器中的各位右移一位，从最低位V30.0移到溢出标志位SM1.1。

