

# 生产线控制与维修技术

PLC 功能模块介绍、三菱 PLC 编程软件、指令系统和控制应用

**学习任务：** 状态转移图 SFC 的应用；  
选择性分支、并行性分支的应用；

**适用年级 / 专业 / 班级：** 20 机电 1、2、3 班；

4 学时

任课教师：姚贵发

**SOC2：**设计三菱 PLC 自动感应启动控制生产线  
输送产品到达位置触碰自动停止完成一个循环程  
序

**成果形式：**要有 I/O 分配、程序表、外部接线图和外形结构图。

**考核方法：**内容正确完整，符合设计目标要求 / 得  
实平台提交

**本次课学习重点与难点：**

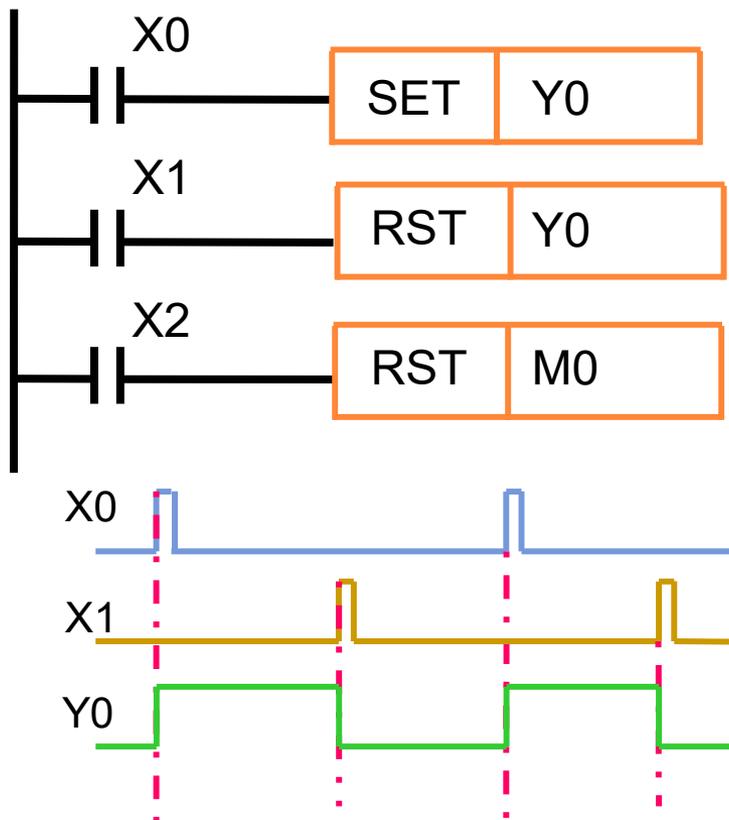
**重点：**状态转移图 SFC 的应用；

**难点：**选择性分支、并行性分支的应用；



# 一、教学回顾

## 1. 指令的梯形图



### ❖ 指令表程序

步序	指令	地址
0	LD	X0
1	SET	Y0
2	LD	X1
3	RST	Y0
4	LD	X2
5	RST	M0



## 2. 数据寄存器 D

通用：D0 ~ D199，200 点。

保持：D200 ~ D511，312 点。

特殊：D8000 ~ D8255（共 256 个）。

PLC 在进行输入输出处理、模拟量控制、位置控制时，需要许多数据寄存器以存储数据和参数。数据寄存器为 16 位，最高位为符号位。32 位数据可用两个数据寄存器来存储，（如 D1D0）。数据寄存器有：通用数据寄存器、保持数据寄存器、特殊数据寄存器、文件寄存器。

### 通用数据寄存器

通用数据寄存器在 PLC 由运行 (RUN) 变为停止 (STOP) 时，其数据全部清零。如果将特殊继电器 M8033 置 1，则 PLC 由运行变为停止时，数据可以保持。

## 文件数据寄存器 (D1000 ~ D2999)

文件寄存器是用于存放大量数据的专用数据寄存器。例如：用于存放采集数据、统计计算数据、多组控制参数等。文件寄存器占用用户程序存储器内的某一存储区间，可用编程器或编程软件进行写操作；PLC 运行时，可用 BMOV 指令将文件寄存器内容读到通用数据寄存器中，但不能用指令将数据写入文件寄存器。

FX 系列 PLC 的数据寄存器编号如表所

示。 表 FX 系列 PLC 的数据寄存器编号

	FX0S	FX1S	FX0N	FX1N	FX2N (FX2NC)
16 位普通 用	D0 ~ D29	D0 ~ D127	D0 ~ D127	D0 ~ D127	D0 ~ D199
16 位保持 用	D30、D31	D128 ~ D255	D128 ~ D255	D128 ~ D7999	D200 ~ D7999
16 位特殊 用	D8000 ~ D8069	D8000 ~ D8255	D8000 ~ D8255	D8000 ~ D8255	D8000 ~ D8195

### 3. 状态器 S

状态器 S 是构成 SFC 图的重要软元件，在步进顺控程序中起着重要的作用，它与后述的步进指令配合使用。一般有五个类型：初始用状态器、返回原点用状态器 (FX2N)、普通状态器、保持状态器、报警用状态器 (FX2N)，如表所示。不使用步进指令时，状态器也可当做辅助继电器使用。

表 器 FX 系列 PLC 的状态

	FX0S	FX1S	FX0N	FX1N	FX2N (FX2NC)
初始用	S0~S9	S0~S9	S0~S9	S0~S9	S0~S9
返回原点用	—	—	—	—	
普通用	S0~S63	S10~S127	S10~S127	S10~S999	S20~S499
保持用	—	S0~S127	S0~S127	S0~S999	S500~S899
报警用	—	—	—	—	S900~S999

## 4. 计数器 C

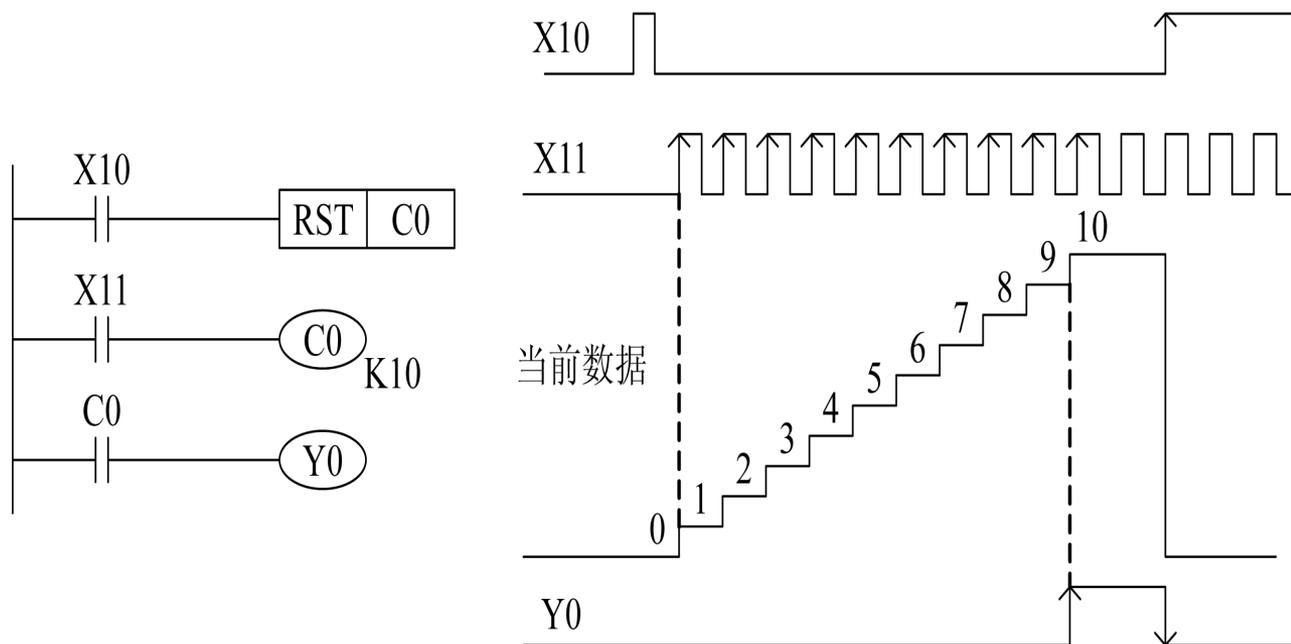


图 16 位增计数器工作原理图

注：16 位增计数器分为通用型和保持型两种。



16 位增计数器的地址编号如表所示。

表 FX 系列 PLC 的 16 位增计数器地址  
编号

	FX0S	FX1S	FX0N	FX1N	FX2N(FX2NC)
普通	C0-C13	C0-C15	C0-C15	C0-C15	C0-C99
保持	C14-C15	C16-C31	C16-C31	C16-C199	C100-C199

## 32 位可逆计数器

这类计数器与 16 位增计数器除位数不同外，还在于它能通过控制实现加 / 减双向计数（限 FX1N/2N/2NC）。增 / 减计数由对应的特殊辅助继电器设定（C△△△—M8△△△），对应的特殊辅助继电器被置为 ON 时为减计数，置为 OFF 时为增计数。



32 位增 / 减可逆计数器地址编号如表所示。

表 FX 系列 PLC 的 32 位增 / 减可逆计数器地址编号

	FX0S	FX1S	FX0N	FX1N	FX2N(FX2NC)
普通	—	—	—	C200-C219	
保持	—	—	—	C220-C234	

## 高速计数器

什么是高速计数器？

(1) 高速计数器是采用中断方式进行高速计数的，与 PLC 的扫描周期无关。

(2) 高速计数器是对特定的输入进行计数（如 FX0N 为 X0~X3）。

(3) 高速计数器为 32 位增 / 减计数型，设定值范围：  
- 2 147 483 648 ~ +2 147 483 647 。

## 5. STL 指令

FX 系列 PLC 的步进顺控指令有两条：一条是步进触点（步进开始）指令 STL，一条是步进返回（也叫步进结束）指令 RET。

### 5.1 STL 指令

STL 步进触点指令用于“激活”开始某个状态，其梯形图符号为。

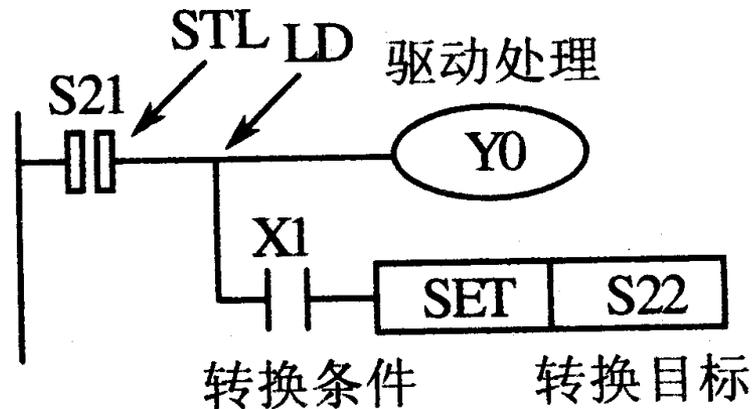
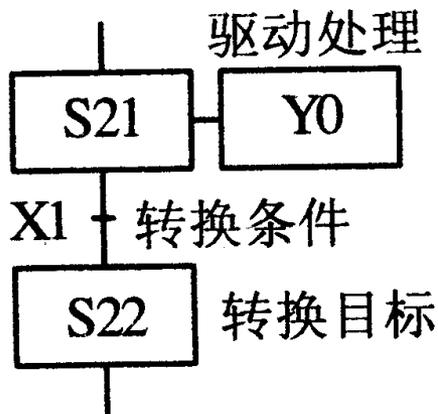


### 5.2 RET 指令

RET 指令用于返回主母线，其梯形图符号为。

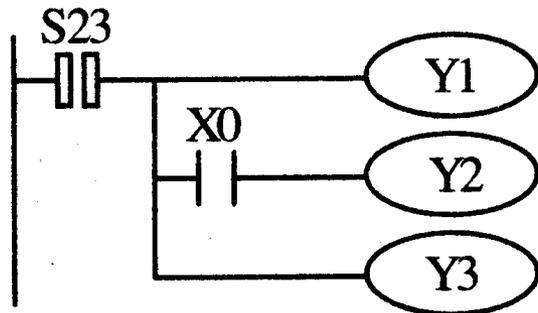


# STL 指令的编程方法

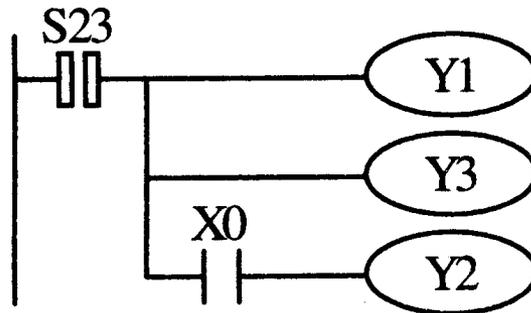


STL S21  
 OUT Y0  
 LD X1  
 SET S22

## STL 指令



(a) 错误的梯形图



(b) 正确的梯形图

梯形图



## STL 指令的特点：

1. 与 STL 触点相连的触点应使用 LD/LDI 指令。
2. STL 触点可以直接驱动或通过别的触点驱动 Y、M、S、T 等元件的线圈，STL 触点也可以使 Y、M、S 等元件置位或复位。
3. CPU 只执行活动步对应的程序。
4. 使用 STL 指令时允许双线圈输出。即不同 STL 触点可以分别驱动同一编程元件的一个线圈。但是同一元件的线圈不能在可能同时为活动步的 STL 区内出现，在有并行步的顺序功能图中，应特别注意这一问题。一个状态寄存器的 STL 触点在梯形图中只能出现一次。

6. 在 STL 触点驱动的电块中不能使用 MC 和 MCR 指令，可以使用 CJP/EJP 指令，当执行 CJP 指令跳入某一个 STL 触点的电块时，不管该 STL 触点是否接通，均执行对应的 EJP 指令之后的电块。
7. 可以对状态寄存器使用 LD 、 LDI 、 AND 、 ANI 、 OR 、 ORI 、 S 、 R 、 OUT 等指令。
8. 对状态寄存器置位的指令，如果不在 STL 触点驱动的电块内置位时，系统程序不会自动将前级步对应的状态寄存器复位。



**例 1：**某信号灯控制系统，初始状态仅红灯亮，按下启动按钮 X0，4 秒后红灯灭，绿灯亮，6 秒后绿灯和黄灯亮，再过 5 秒后，绿灯和黄灯灭，红灯亮。请设计顺序功能图，并用步进指令编程。

## 解题要点

一、分析问题：（属于单周期工作方式）

1、确定编程元件

（1）用 Y0、Y1、Y2 分别控制红灯、绿灯和黄灯。

（2）时间继电器：T0，定时 4S，T1，定时 6S，T2，定时 5S

2、步的划分：根据题意将一个工作循环划分为 4 步，即初始步、4 秒步、6 秒步、5 秒步，分别采用编程元件采用 S10、S11、S12 和 S13 来代表。

### 3、转换条件：

进入初始步的条件： $M8002=1$ ； $T2=1$ 。

进入4秒步的条件： $X0=1$ 。

进入6秒步的条件： $T0=1$ 。

进入5秒步的条件： $T1=1$ 。

特别注意初始步的激活问题：用  $M8002$  的常开触点将初始步的编程元件置位。

### 4、各步的动作：

初始步的动作： $Y0$ 。

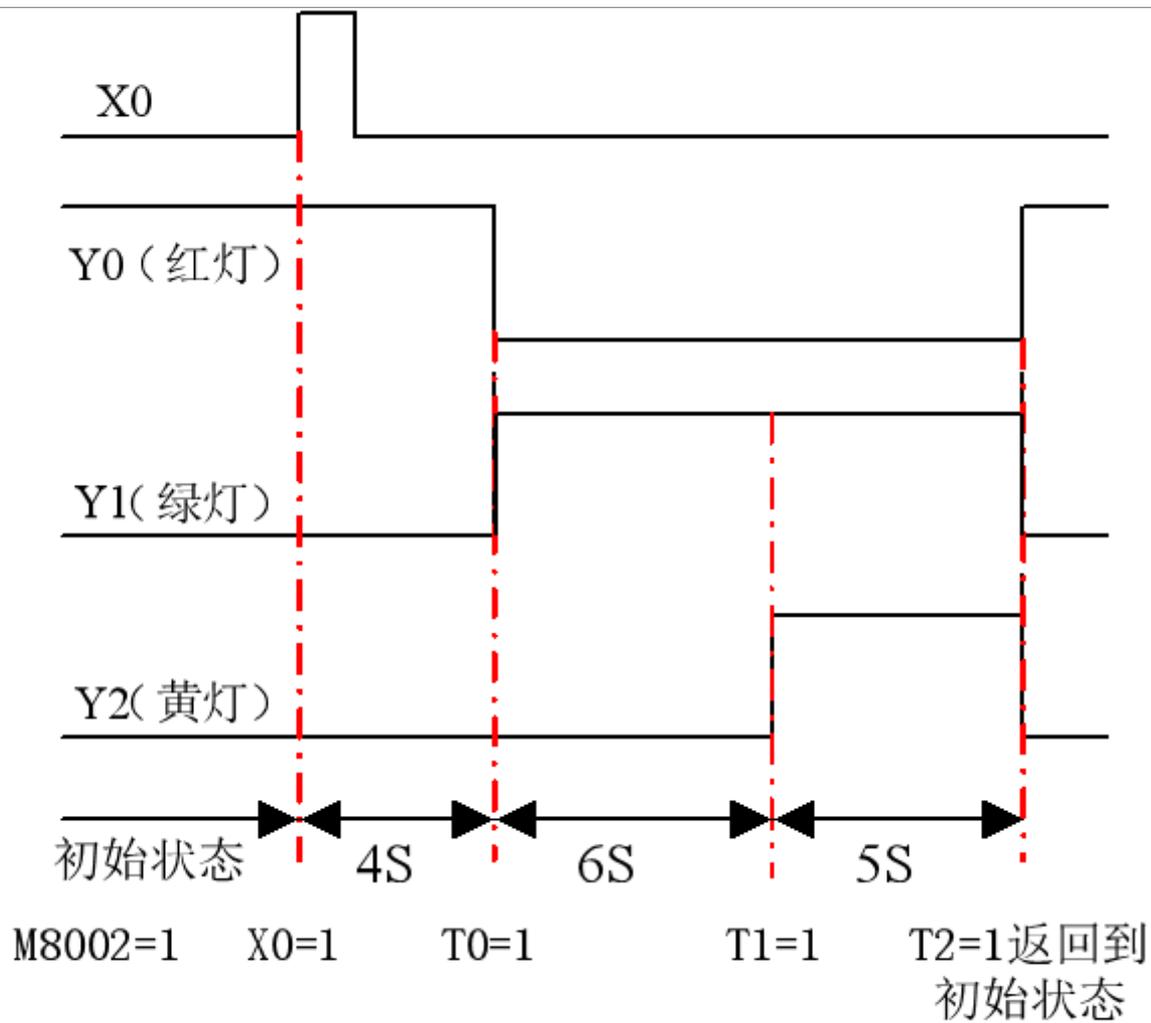
4秒步的动作： $Y0$ 、 $T0$ 。

6秒步的动作： $Y1$ 、 $T1$ 。

5秒步的动作： $Y1$ 、 $Y2$ 、 $T2$ 。

二、根据分析画出时序图（见图所示）

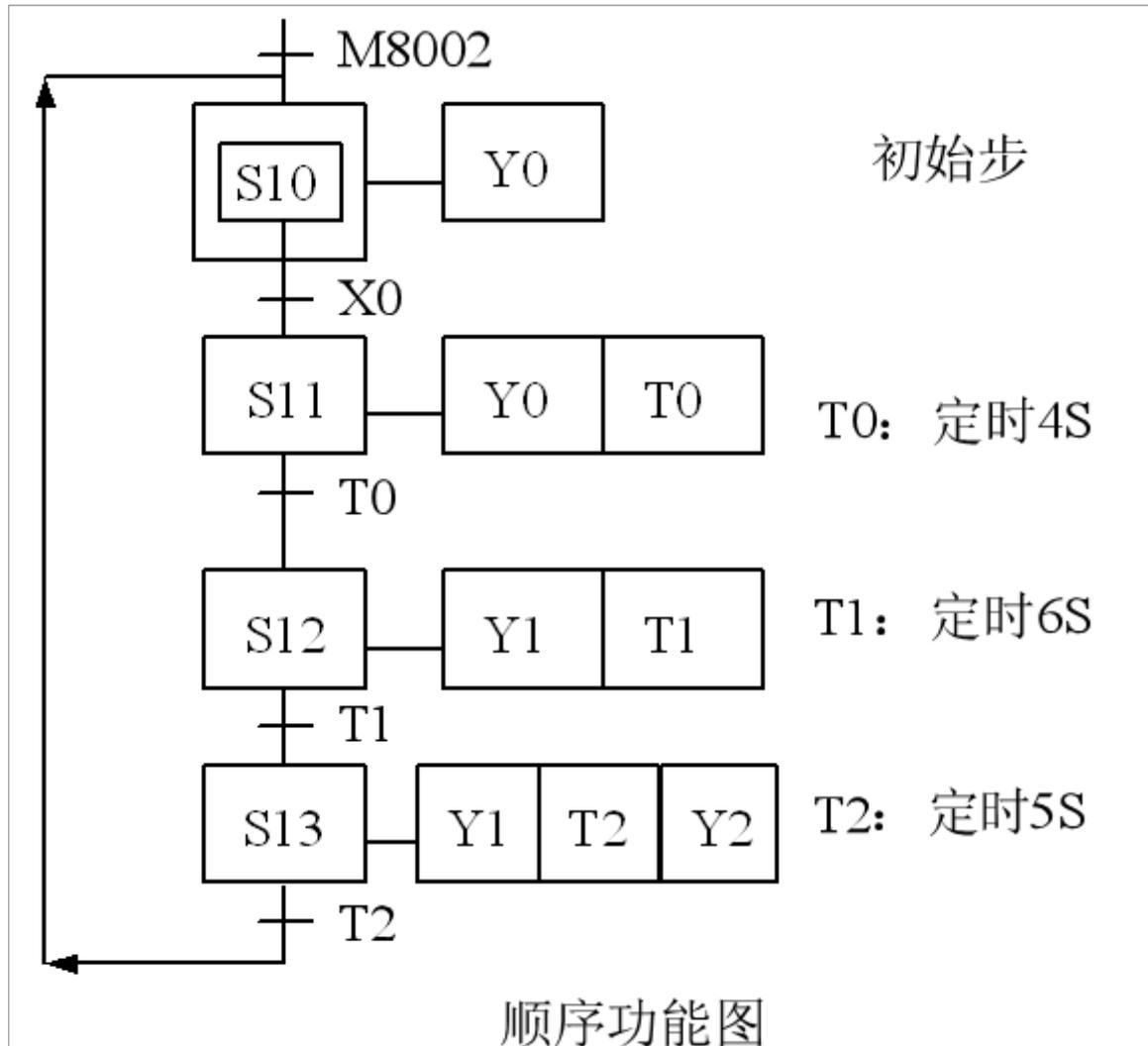




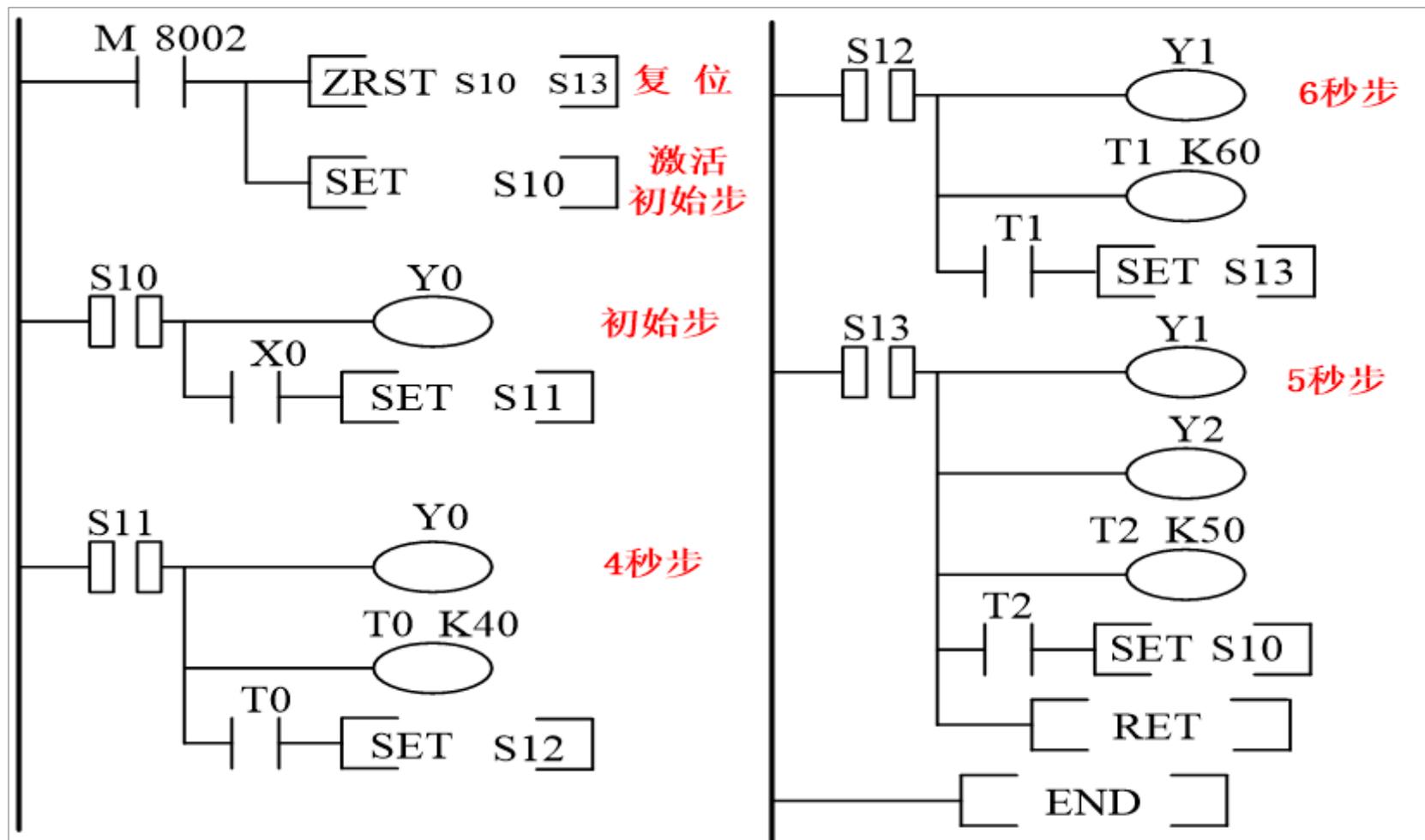
信号灯控制系统时序图



### 三、根据时序时画出顺序功能图（如下图示）



#### 四、根据顺序功能图设计梯形图程序（如下图示）

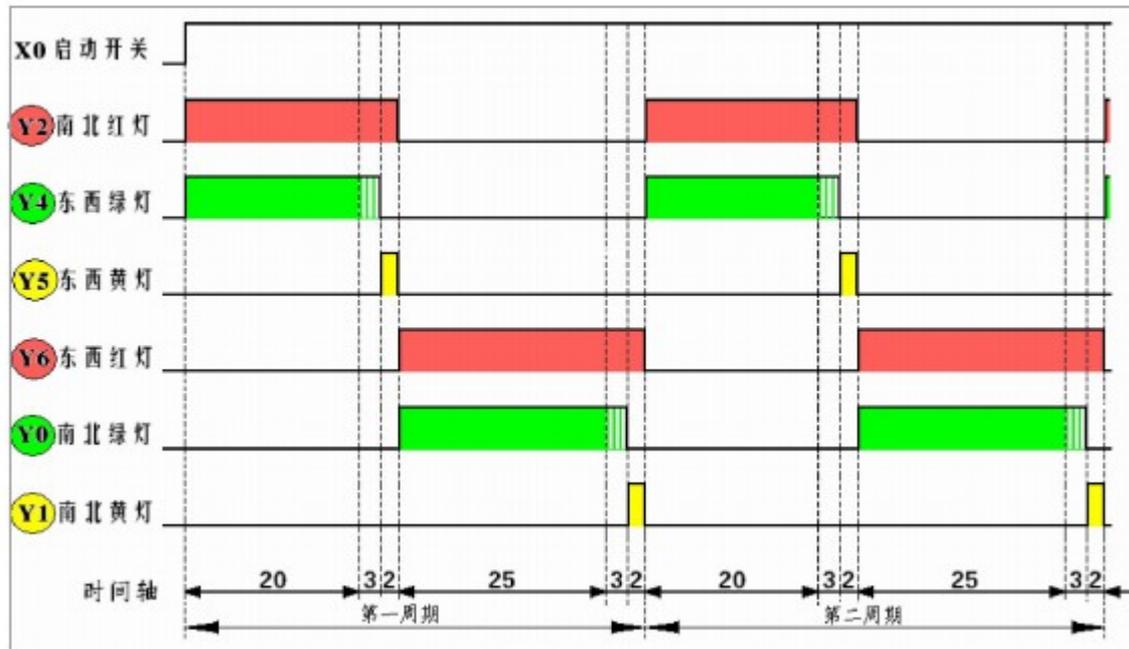


信号灯控制系统的梯形图程序

## 例 2 :

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。

**工作时序：**南北红灯亮维持 25 秒，在南北红灯亮的同时东西绿灯也亮，并维持 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 30 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。



1. I/O 分配:

输入地址		输出地址	
启动 SD	X0	南北方向绿灯	Y0
		南北方向黄灯	Y1
		南北方向红灯	Y2
		东西方向绿灯	Y3
		东西方向黄灯	Y4
		东西方向红灯	Y5

2. 实训步骤:

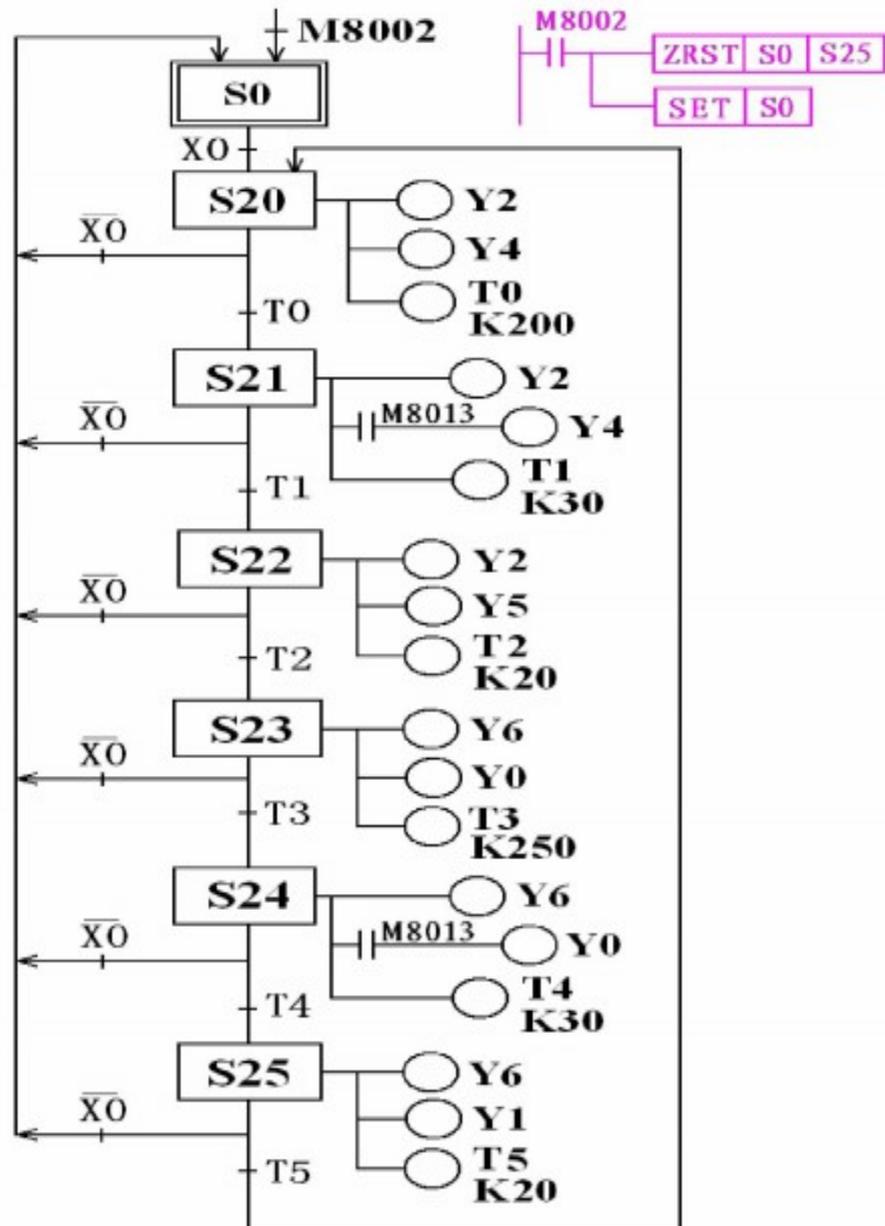
(1)按照上述 I/O 分配设计程序梯形图:

(2)连接模拟电路, 调试程序:

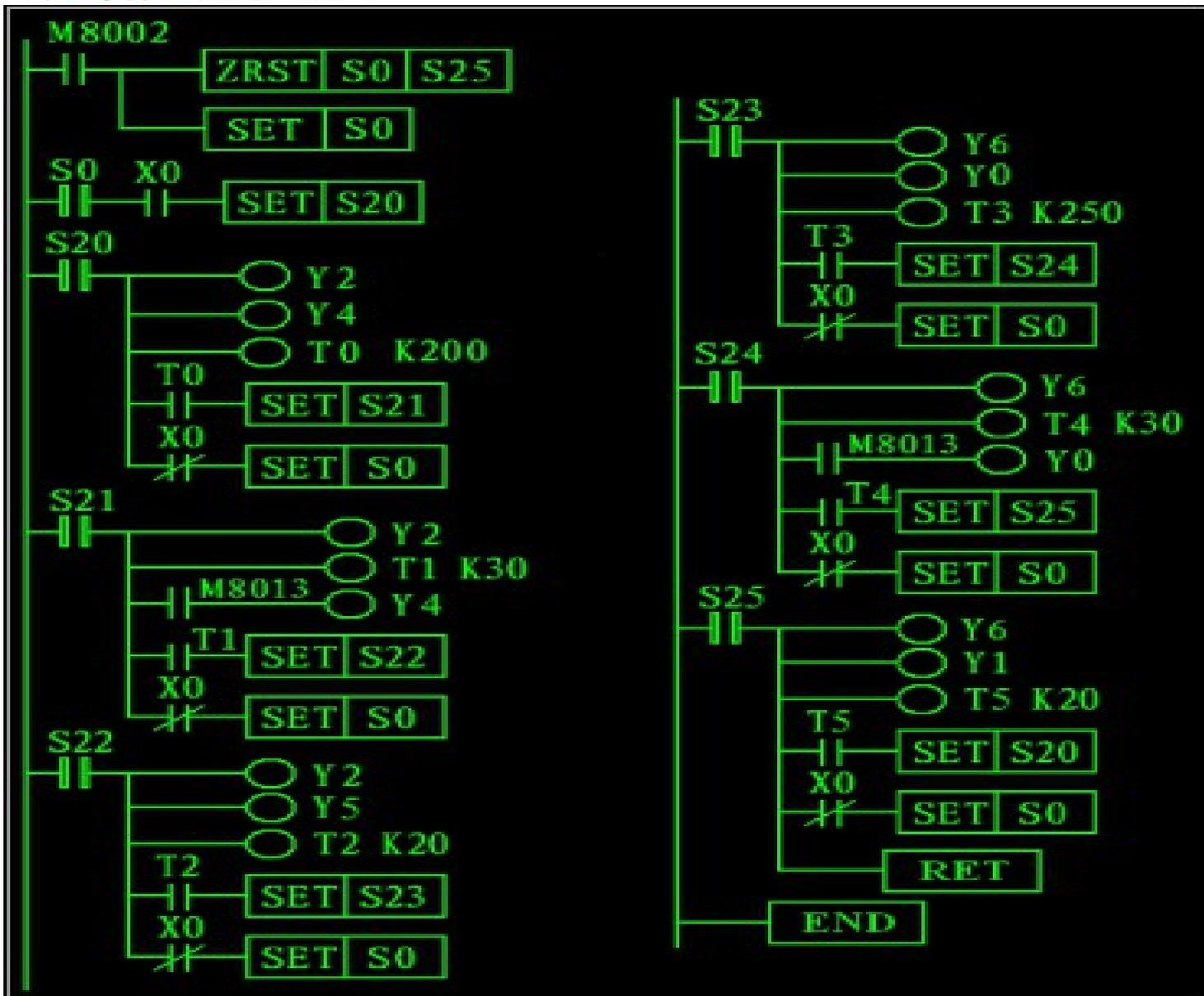
(3)将调试运行正常的梯形图写入实验报告。

(实验报告包括: 1、实验目的 2、控制要求 3、I/O 口分配 4、梯形图 5、实验结果)。

2. 参考 SFC 图如下:



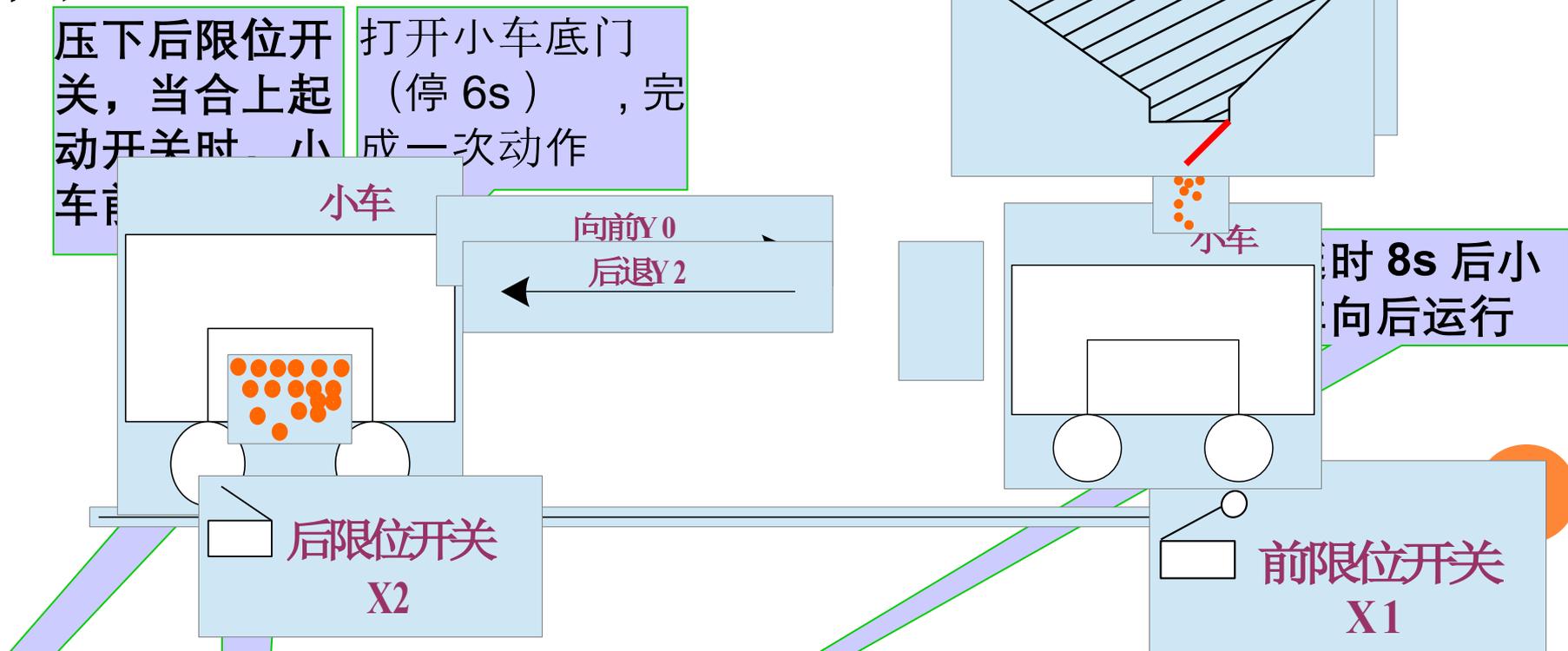
3. 参考梯形图(见下):



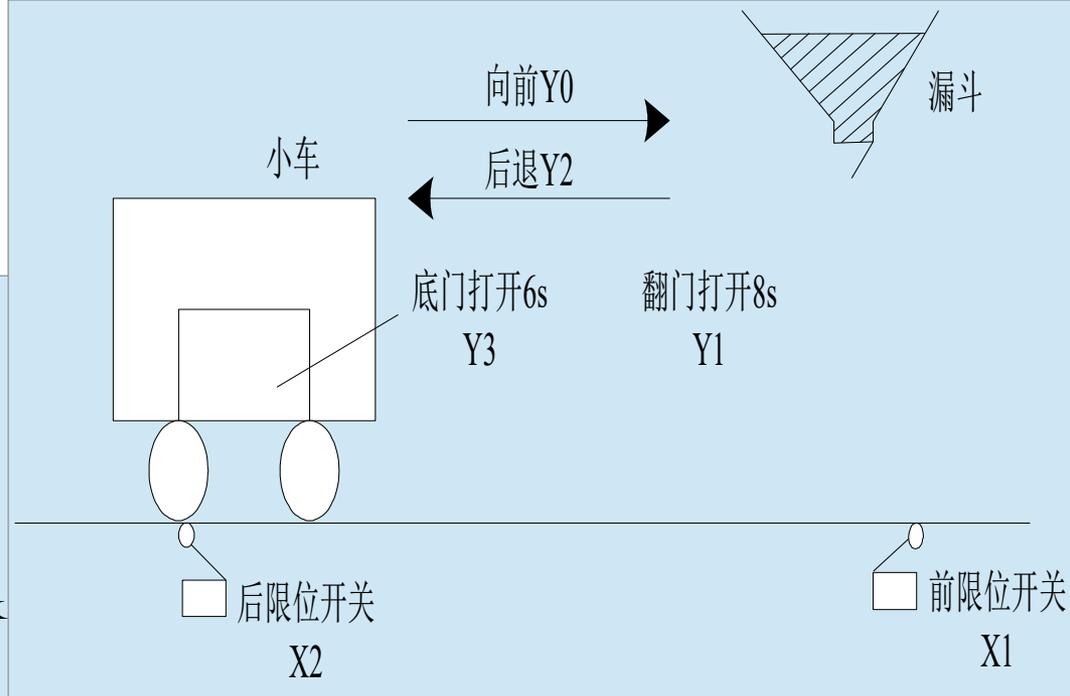
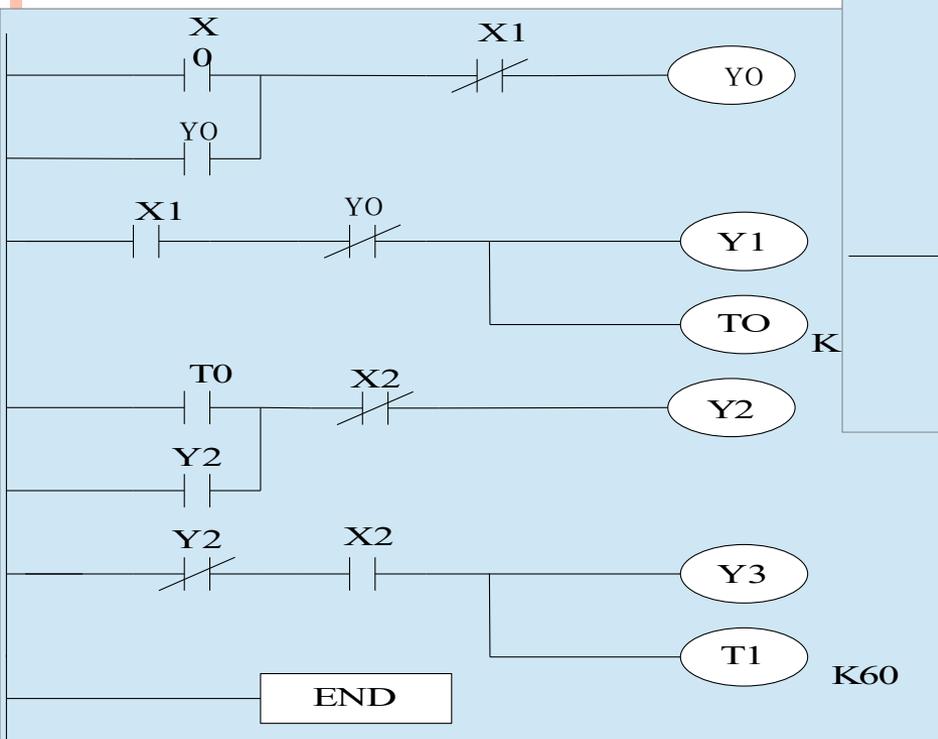
## 二、新课教学内容

### 1. 状态转移图 SFC 的应用

#### 1.1 小车运行过程示意图



## 梯形图

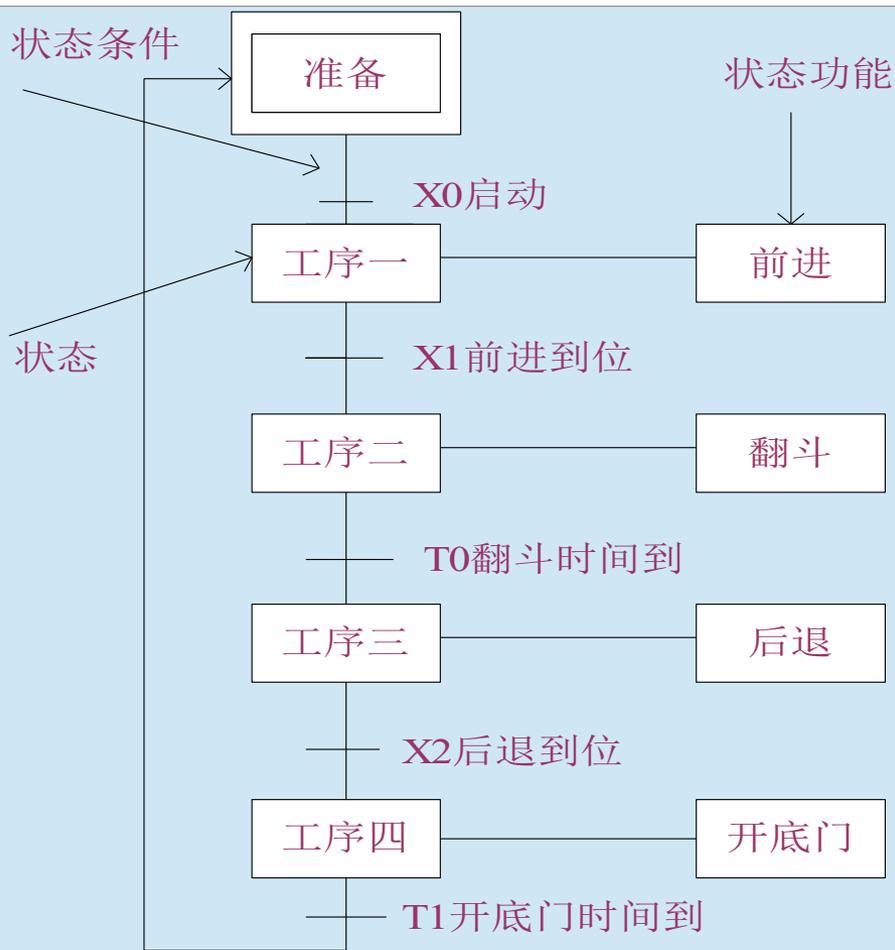


编制的程序存在以下一些问题：

- (1) 工艺动作表达繁琐。
- (2) 梯形图涉及的联锁关系较复杂，处理起来较麻烦。
- (3) 梯形图可读性差，很难从梯形图看出具体控制工艺过程。

为了程序编制的直观性和复杂控制逻辑关系的分解与综合。提出了状态转移图。





将复杂的任务或过程分解成若干个工  
(状态)

控制任务实现了简化

只要弄清各工序成立的条件、工序转  
条件和转移的方向，就可进行这类图形  
设计。

可读性很强，能清晰地反映全部控制  
过程。

图中的“工序”更换为“状态”，就得到了状  
态图

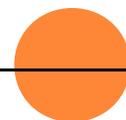
编程的一般思想为：

将一个复杂的控制过程分解为若干个工  
作状态，弄清各状态的工作细节（状态的功  
能、转移条件和转移方向），再依据总的控  
制程序要求，将这些状态联系起来，形成状  
态图，进而编绘梯形图程序。



# FX2 的状态元件分类

类别	元件编号	个数	用途及特点
初始状态	S0 ~ S9	10	用作 SFC 的初始状态
返回状态	S10 ~ S19	10	多运行模式控制当中，用作返回原点的状态
一般状态	S20 ~ S499	480	用作 SFC 的中间状态
掉电保持状态	S500 ~ S899	400	具有停电保持功能，停电恢复后需继续执行的场合，可用这些状态元件
信号报警状态	S900 ~ S999	100	用作报警元件使用



(1) 将整个工作过程按工作步序进行分解，每个工序对应一个状态，其状态分配如下：

- |        |     |       |
|--------|-----|-------|
| ① 初始状态 | S0  | ④ 后退  |
| ② 前进   | S20 | S22   |
| ③ 翻斗车  | S21 | ⑤ 开底门 |
|        |     | S23   |

(2) 理解每个状态的功能、作用

- S0 PLC上电作好工作准备
- S20 前进（输出 Y0，驱动电动机 M正转）
- S21 翻斗车（输出 Y1，同时计时 T0 开始工作）
- S22 后退（输出 Y2，驱动电动机 M反转）
- S23 开底门（输出 Y3，同时计时 T1 开始工作）

各状态的功能是通过 PLC 驱动其各种负载来完成的。负载可由状态元件直接驱动，也可由其他软元件触点的逻辑组合驱动。

状态的转移条件可以是单一的，也可以是多个元件的串并联组合。



(3) 找出每个状态的转移条件。

即在什么条件将下个状态“激活”。状态转移图就是状态和状态转移条件及转移方向构成的流程图，弄清转移条件当然是必要的。

由工作过程可知，本例各状态的转移条件为：

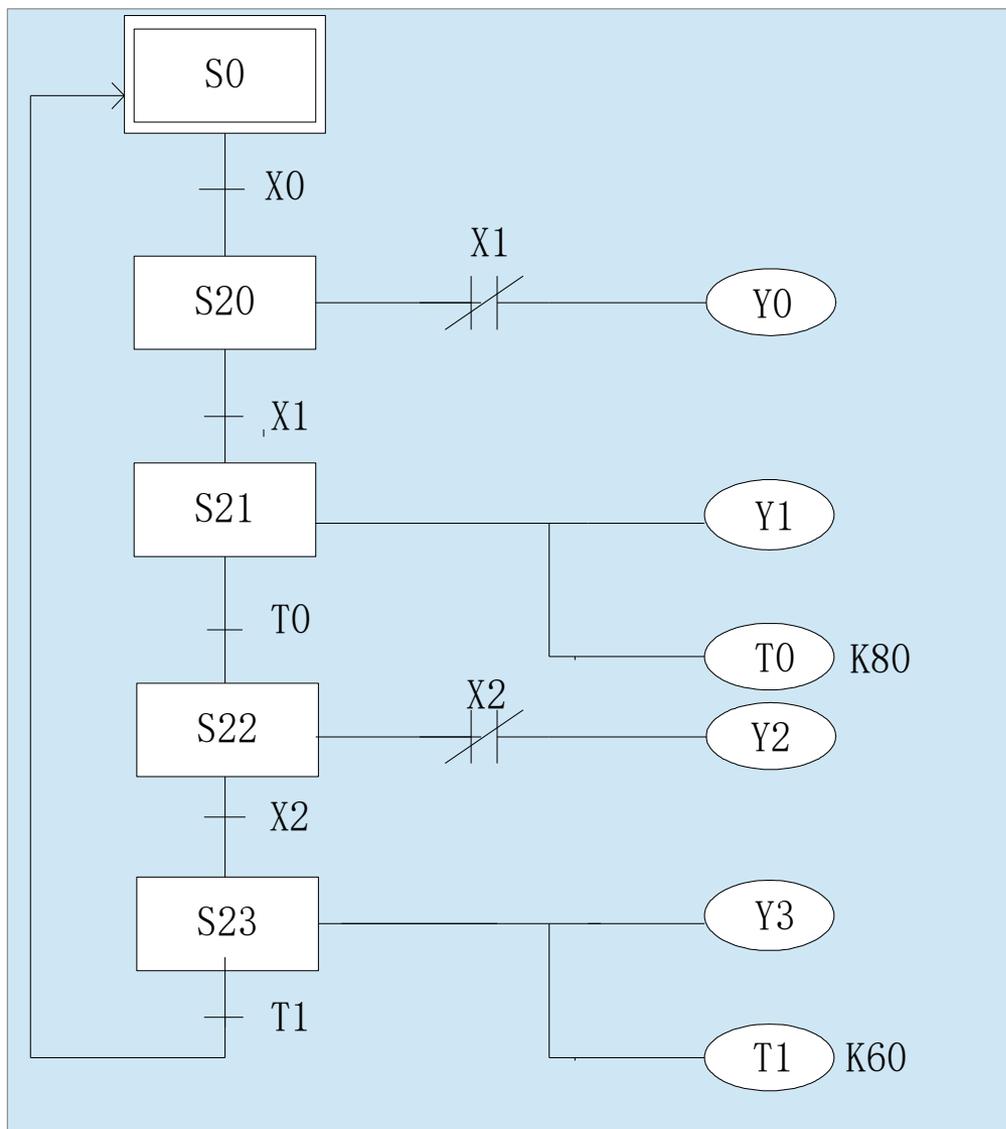
S20 X0      S21 X1

S22 T0      S23 X2

状态的转移条件可以是单一的，也可以是多个元件的串并联组合。

经过上述三步，可得  
小车自动往返控制的  
顺控状态图

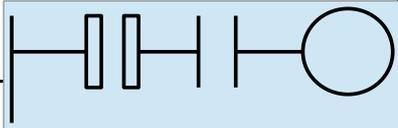




顺控状态图



## 步进指令（STL、RET）

符号、名称	功能	电路表示及操作元件	程序步
STL 步进阶梯	步进阶梯开始	元件：S 	1
RET 返回	步进阶梯结束	元件：无 	1

### 步进指令的说明：

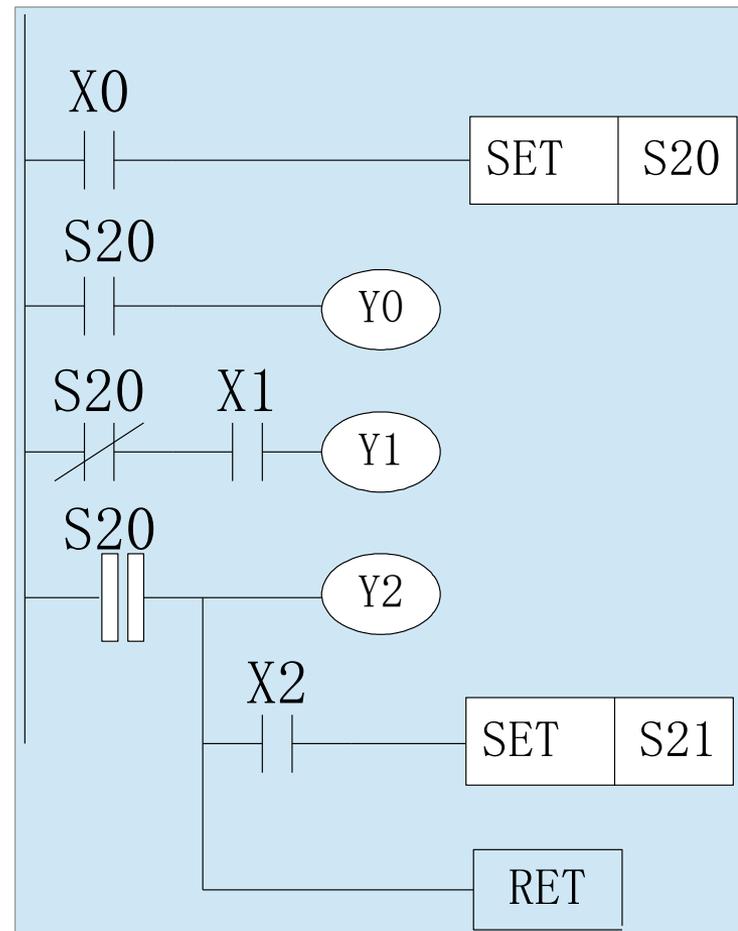
- ① 步进接点须与梯形图左母线连接。使用 STL 指令后，LD 或 LDI 指令点则被右移，所以当把 LD 或 LDI 点返回母线时，需要使用步进返回指令 RET。
- ② 使用 STL 指令后的状态继电器（有时亦称步进继电器），才具有步进控制功能。这时除了提供步进常开接点外，还可提供普通的常开接点与常闭接点，但 STL 指令只适用于步进接点。



## 步进指令的说明：

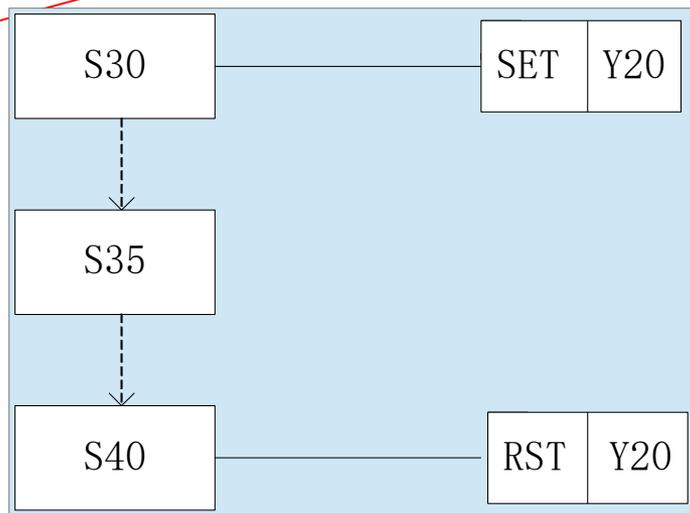
① 步进接点须与梯形图左母线连接。使用 STL 指令后，LD 或 LDI 指令点则被右移，所以当把 LD 或 LDI 点返回母线时，需要使用步进返回指令 RET。

② 使用 STL 指令后的状态继电器（有时亦称步进继电器），才具有步进控制功能。这时除了提供步进常开接点外，还可提供普通的常开接点与常闭接点，但 STL 指令只适用于步进接点。



只有 S40 接通时，Y20 才断开，即从 S30 接通开始到 S40 接通为止，这段时间为 Y20 持续接通时间。

③ 只有步进接点接通时，它后面的电路才能动作。如果步进接点断开，则其后面的电路将全部断开。当需保持输出结果时，可用 SET 和 RST 指令来实现。



④ 使状态继电器复位的方法。当使用 S500~S899 状态继电器时，具有断电保护功能，即断电后再次通电，动作从断电时的状态开始。但在某些情况下需要从初始状态开始执行动作，这时需要复位所有的状态。此时应使用功能指令区间复位指令 ZRST 实现状态复位操作。实际应用时，区间复位的起始值为设定复位开始器件的编号，区间复位的终止值为设定复位结束器件的编号。

⑤ 如果不用 STL 步进接点时，状态继电器可作为普通辅助（中间）继电器 M 用，这时其功能与 M 相同。

⑥ 步进指令后面可以使用 CJP/EJP 指令，但不能使用 MC/MCR 指令。

⑦ 在时间顺序步进控制电路中，只要不是相邻步进工序，同一个定时器可在这些步进工序中使用，这可节省定时器。

**1. 状态的三要素：驱动负载、指定转移目标、指定转移条件**

**2. 状态的开启与关闭及状态转移图执行的特点**

开启可以理解为该段程序被扫描执行。而关闭则可以理解为该段程序被跳过，未能扫描执行

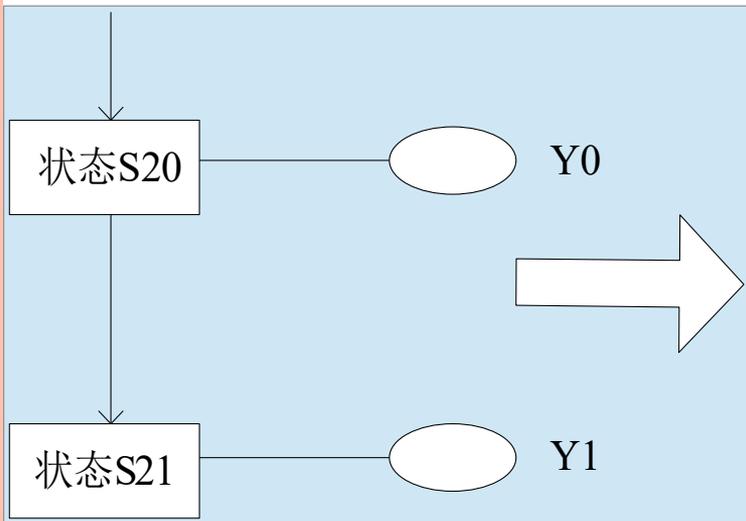
**3. 步进梯形图：使用步进接点指令和步进返回指令梯形图的形式将**

状态转移图以梯形图的形式表述出来。

步进指令常用于控制时间和位移等顺序的操作过程。

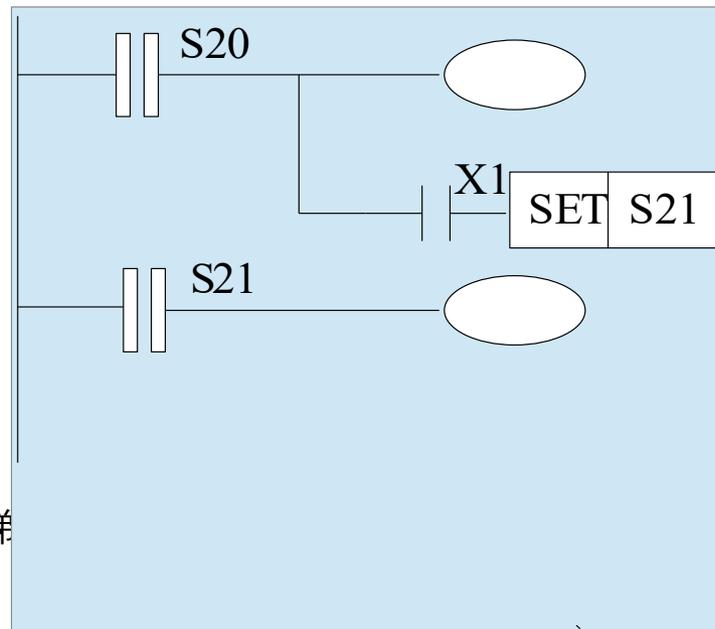
步进接点只有常开接点，而没有常闭接点。





状态转移图

步进梯



```

STL S20
OUT Y0
LD X1
SET S21
STL S21
OUT Y1
  
```



## 2. 多分支状态转移图的处理与梯形图的转换

在步进顺序控制过程中，有时需要将同一控制条件转向多条支路，或把不同条件转向同一支路，或跳过某些工序或重复某些操作。以上这些称之为多分支状态转移图。

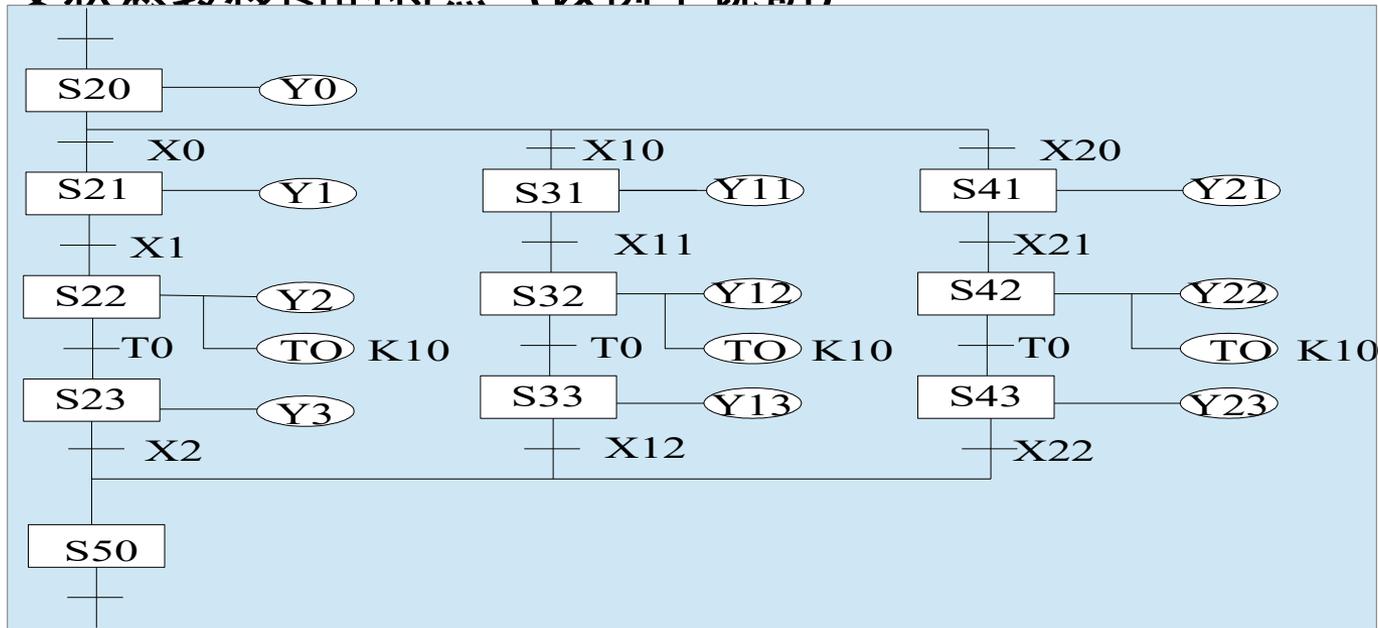
这种多种工作顺序的状态流程图为：分支、汇合流程图。

根据转向分支流程的形式，可分为：选择性分支与汇合流程图与并行分支与汇合流程图。



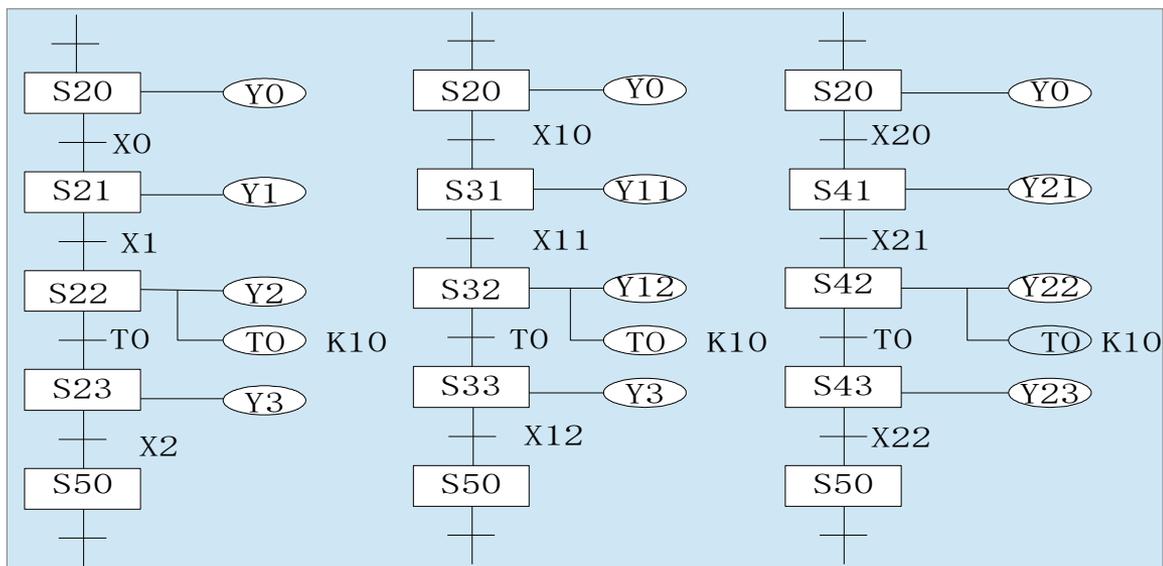
## 选择性分支与汇合的处理（从多个流程顺序中选择执行哪一个流程）

### 选择性分支状态转移图的特点（以例子说明）



- (1) S20 为分支状态。（根据不同的条件（X0，X10，X20），选择且只能选择执行其中的一个流程。）
- (2) S50 为汇合状态，可由 S23、S33、S43 任一状态驱动。
- (3) 该状态转移图有三个流程顺序。





## 1、首先进行分支状态元件的处理

分支状态的处理方法是：首先进行分支状态的输出连接，然后依次按照转移条件置位各转移分支的首转移状态元件

## 2、再依顺序进行各分支的连接

## 3、最后进行汇合状态的处理

汇合状态的处理方法是：先进行汇合前的驱动连接，再依顺序进行汇合状态的连接）。

# 选择性分支与汇合状态转移图的编程方法

编程原则是先集中处理分支状态，然后再集中处理汇合状态。

## 分支状态的编程

编程方法是先进行分支状态的驱动处理，再依顺序进行转移处理。

按分支状态的编程方法，首先对 S20 进行驱动处理（OUT Y0），然后按 S21、S31、S41 的顺序进行转移处理。

STL	S20	
OUT	Y0	驱动处理
LD	X0	
SET	S21	转移到第一分支状态
LD	X10	
SET	S31	转移到第二分支状态
LD	X20	
SET	S41	转移到第三分支状态



# 汇合状态的编程

编程思想：先进行汇合前状态的驱动处理，再依顺序进行汇合状态的转移处理。

第一分支汇合前处理

第二分支汇合前处理

第三分支汇合前处理

汇合前的驱动输出进行

```

STL S21
OUT Y1
LD X1
SET S22
STL S22
OUT Y2
OUT T0
    K10
LD T0
SET S23
STL S23
OUT Y3
    
```

```

OUT Y11
LD X11
SET S32
STL S32
OUT Y12
OUT T0
    K10
LD T0
SET S33
STL S33
OUT Y13
    
```

```

STL S41
OUT Y21
LD X21
SET S42
STL S42
OUT Y22
OUT T0
    K10
LD T0
SET S43
STL S43
OUT Y23
    
```

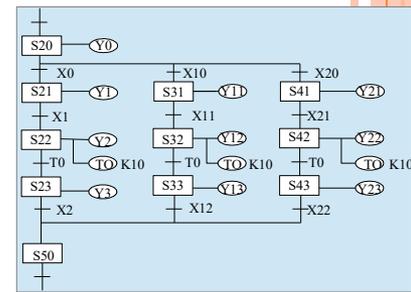
```

STL S23
LD X2
SET S50
STL S33
LD X12
SET S50
STL S43
LD X22
SET S50
    
```

由第一分支转移到汇合点

由第二分支转移到汇合点

由第三分支转移到汇合点



### 3. 并行分支与汇合的编程

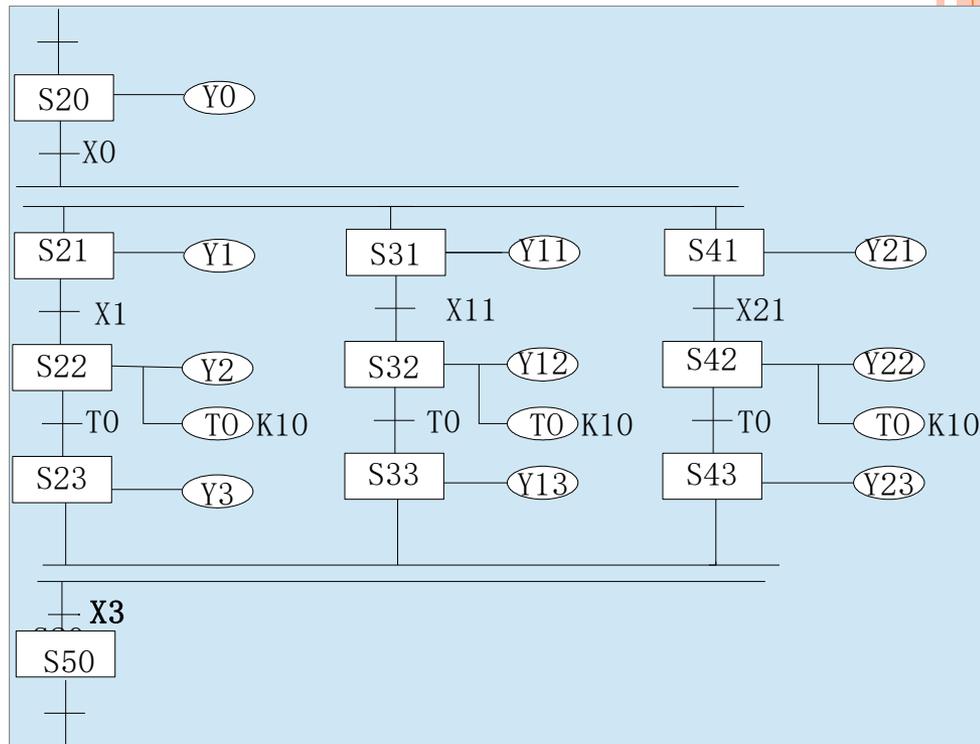
#### (一) 并行分支状态转移图

#### (二) 并行分支与汇合状态转移图与梯形图的转换

首先进行分支状态元件的处理

处理方法：

首先进行分支状态的输出连接，然后依次按照转移条件置位各转移分支的首转移状态元件。



再依顺序进行各分支的连接，最后进行汇合状态的处理

汇合状态的处理方法：

先进行汇合前的驱动连接，再依顺序进行汇合状态的连接。



# 选择性分支与汇合状态转移图的编程方法。

编程原则是先集中处理分支状态，然后再集中处理汇合状态。

## 1. 分支状态的编程

编程方法是先进行分支状态的驱动处理，再依顺序进行转移处理。

首先对 **S20** 进行驱动处理（**OUT Y0**），然后按 **S21**、**S31**、**S41** 的顺序进行转移处理。程序如下。

**STL S20**

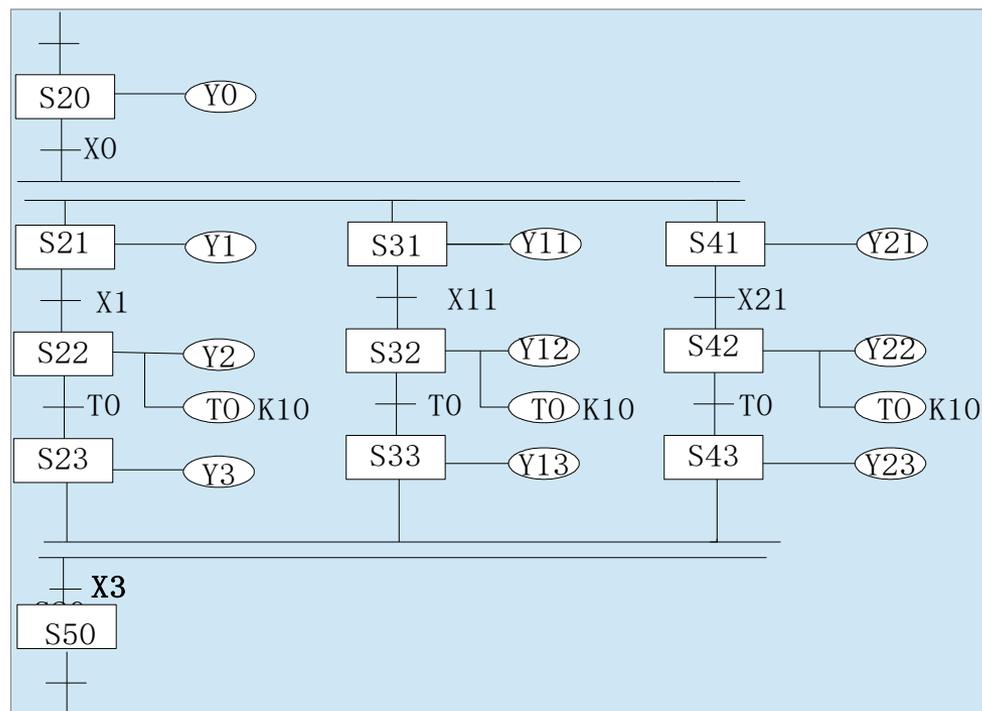
**OUT Y0**    驱动处理

**LD X0**

**SET S21**    向第一分支转移

**SET S31**    向第二分支转移

**SET S41**    向第三分支转移



# 汇合状态的编程

先进行汇合前状态的驱动处理，再依顺序进行向汇合状态的转移处理。

按照汇合状态的编程方法，依次将

第一分支汇合前处理

第二分支汇合前处理

第三分支汇合前处理

S4汇合前的驱动处理

输出进行 S43 (第

三分支) 向 S50 转移。

```

STL S21
OUT Y1
LD X1
SET S22
STL S22
OUT Y2
OUT T0
K10
LD T0
SET S23
STL S23
OUT Y3
    
```

```

STL S31
OUT Y11
LD X11
SET S32
STL S32
OUT Y12
OUT T0
K10
LD T0
SET S33
STL S33
OUT Y13
    
```

```

STL S41
OUT Y21
LD X21
SET S42
STL S42
OUT Y22
OUT T0
K10
LD T0
SET S43
STL S43
OUT Y23
    
```

```

STL S23
STL S33
STL S43
LD X3
SET S50
    
```



## 4. 步进指令应用举例

某店面名叫“彩云间”，这三个字的广告字牌要求实现闪烁，用HL1~HL3三个灯点亮“彩云间”三个字。其闪烁要求如下：在打开闪烁开关以后，首先是“彩”亮1秒，接着是“云”亮1秒，然后“间”亮1秒，在这之后“彩云间”三字闪烁以0.5秒为周期亮灭两次。

模拟过程：



采用状态编程方法实现：

1、可将过程分为以下几个状态：

S0 初始状态

S20 照亮字“彩”

S21 照亮字“云”

S22 照亮字“间”

S23 第一次三字全灭

S24 第一次三字全亮

S25 第二次三字全灭

S26 第二次三字全亮

2、其 I/O 地址分配如下：

X0 闪烁启动按钮 SB

Y0 点亮“彩”字的灯 HL1

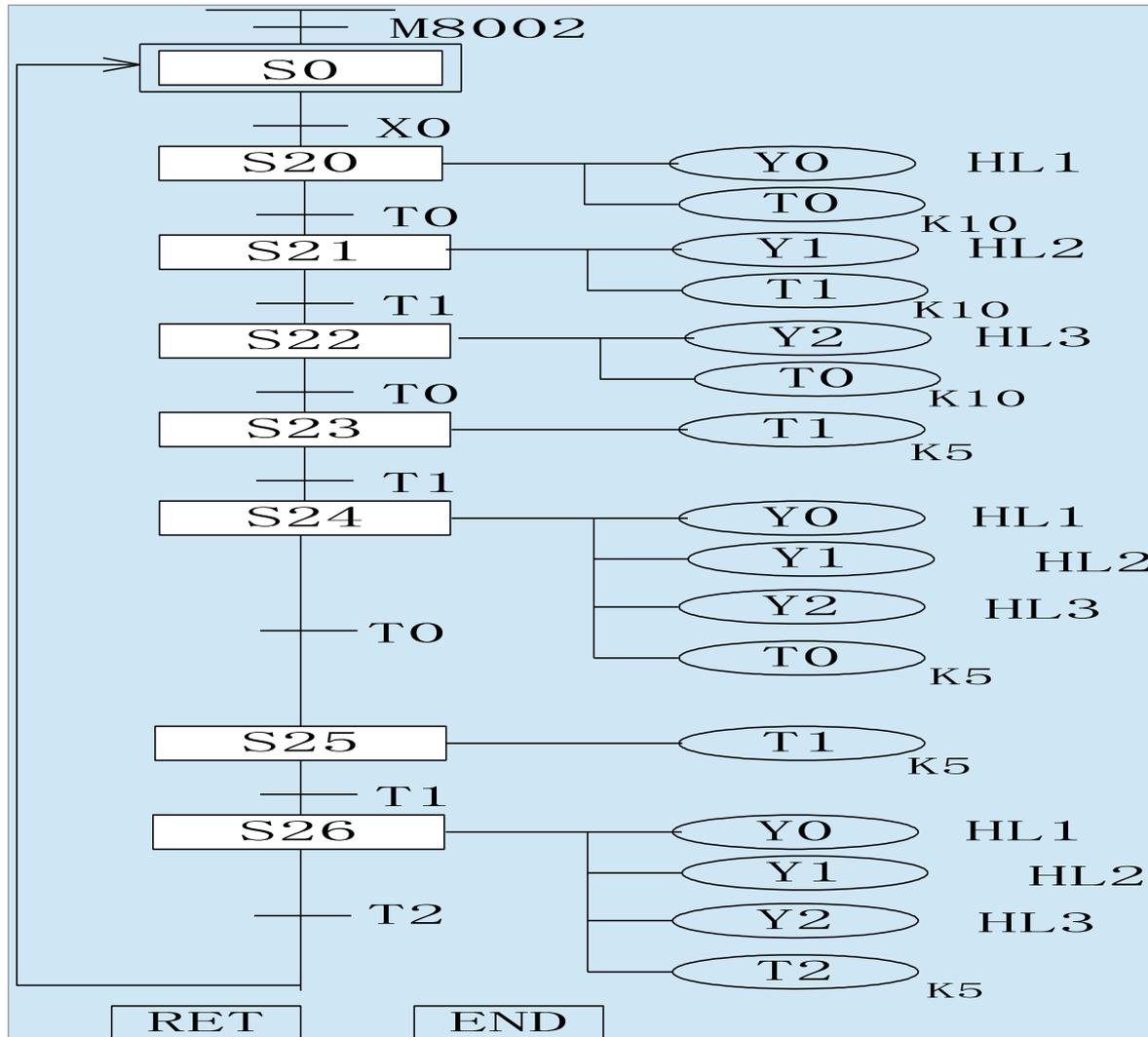
Y1 点亮“云”字的灯 HL2

Y2 点亮“间”字的灯 HL3



# 状态转移图

# 由梯形图得到指令程序



```

LD    M8002
SET   S0
STL   S0
LD    X0
SET   S20
STL   S20
OUT   Y0
OUT   T0
      K10
LD    T0
SET   S21
STL   S21
OUT   Y1
OUT   T1
      K10
LD    T1
SET   S22

```



## 三、本次课小结

:

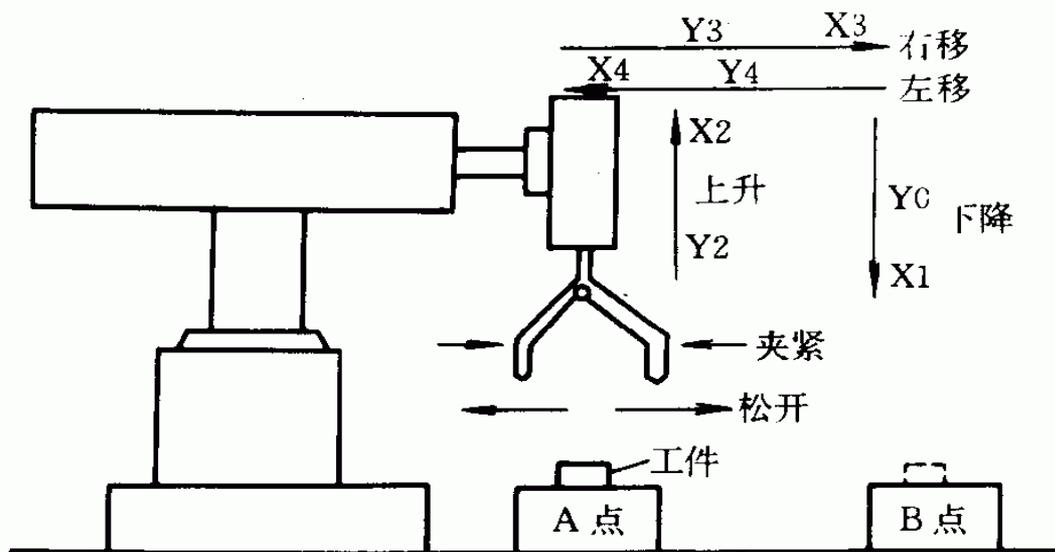
1. 学习了状态法也叫功能表图法，是程序编制的重要方法及工具；
2. 学习了 FX2 系列可编程控制器的步进顺控指令及大量的状态软元件就是为状态编程法安排的；
3. 一般先绘出状态转移图，再转换成梯形图及指令；
4. 编程思想、状态元件、状态三要素、状态指令及状态转移图；
5. 说明常见状态转移图的编程及调试方法；



## 四、布置作业

机械手是工业机器人系统中传统的任务执行机构，是机器人的关键部件之一。

机械手可避免人体受到高温、腐蚀及有毒气体等因素的危害，降低工人的劳动强度。



机械手工作示意图



**谢谢观看  
!**

