

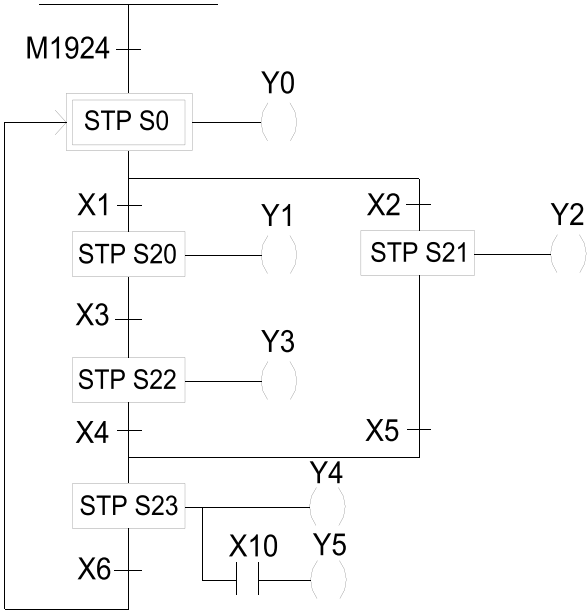
# 第 8 章：步进指令说明

结构化程序设计是软件设计的一大趋势，它的特点是可读性高、维护、更新容易，软件质量可靠性大大提升。尤其当控制偏向机械流程控制时，使用传统的设计方式来设计程序，往往令第三者难以着手，也就是程序可读性低、更新、维护风险较高。因此，专门针对机械动作流程的顺序控制，如果能结合现有广泛的梯形图语言，再加上步进执行指令辅助，将使这方面的设计工作更省时，更省力，且软件掌握度更高。我们将这种结合流程控制与梯形图语言的设计方式称为步进梯形图（STEP LADDER）语言。

步进梯形图是以一个步进点（STEP）为最小单元。一个步进点相当于机械部件中的一个步序（站），每个步序都有动作输出，整台机械或是整个顺序控制的流程，便是一个一个步进点逻辑串联或并联组成，其一步接一步循序执行的环境，使人对机械的运作一目了然，在设计、操作、维护上都相当便捷容易。

## 8.1 步进梯形图工作原理

### 【范例】

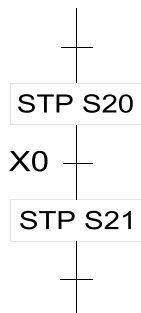


### 【说明】

1. **STP Sxxx** 是步进点（STEP）Sxxx 的表示符号，其中 Sxxx 可从 S0~S999。当执行到此步进点时（此点 ON），便会执行连在右边的梯形图，而前一个步进点及输出都会变 OFF。
2. M1924 为一开机 ON 一个扫描时间的接点，因此，一开机即进入初始步进点 S0（S0 ON）这一站，而其它步进点都不动作，Y1~Y5 皆 OFF。即 M1924 ON→S0 ON→Y0 ON，Y0 会维持到 X1 或 X2 其中一个接点先 ON 为止。
3. 假设 X2 先 ON，就会执行 S21 这条路径，即  $X2\ ON \Rightarrow \begin{cases} S21\ ON \\ S0\ OFF \end{cases} \Rightarrow \begin{cases} Y2\ ON \\ Y0\ OFF \end{cases}$ ，而 Y2 会维持到 X5 ON 为止。
4. 假设 X5 ON，就会前进到步进点 S23 这一站，即  $X5\ ON \Rightarrow \begin{cases} S23\ ON \\ S21\ OFF \end{cases} \Rightarrow \begin{cases} Y4\ ON \\ Y2\ OFF \end{cases}$ ，Y4 和 Y5 会维持到 X6 ON 为止。  
※如 X10 ON，则 Y5 也会 ON。
5. 假设 X6 ON，就会前进到步进点 S0 这一站，即  $X6\ ON \Rightarrow \begin{cases} S0\ ON \\ S23\ OFF \end{cases} \Rightarrow \begin{cases} Y0\ ON \\ Y4、Y5\ OFF \end{cases}$ ，如此便完成一个循环的控制流程，而进入下一循环的控制流程。

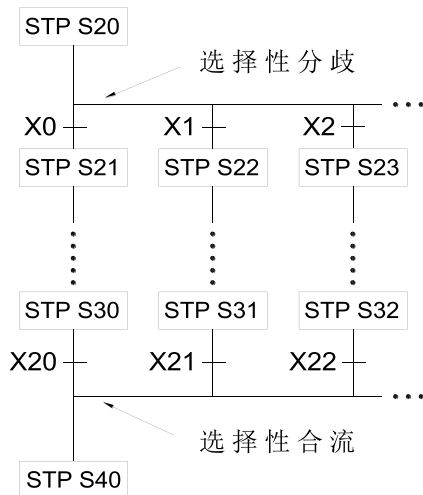
## 8.2 步进梯形图基本组成

### ① 单一回路



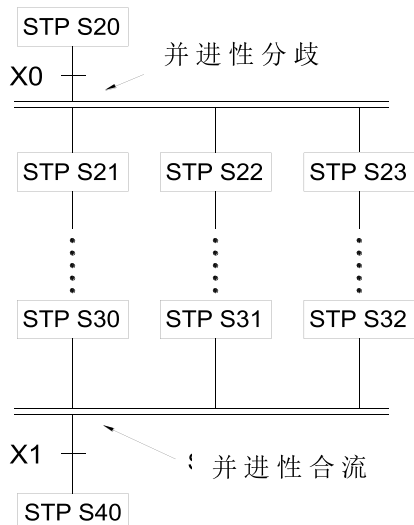
- 步进点 S20 单独经 X0 前进到步进点 S21。
- X0 可改为其它接点的串、并联组合。

### ② 选择性分歧/合流



- 步进点 S20 选择其下先 ON 的接点，当作唯一执行的回路，例如 X2 先 ON，则只执行步进点 S23 这条回路。
- 一个分歧最多有 8 条回路。
- X1、X2.....X22 都可改为其它接点的串、并联组合。

### ③ 并进式分歧/合流

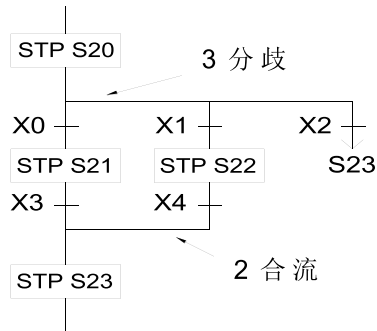


ICE

- 步进点 S20 在 X0 ON 后，同时执行其下的所有回路，即 S21、S22、S23.....都动作。
- 在合流处上的所有分歧回路皆执行到最后一个步进点（如 S30、S31、S32），在等到 X1 ON，即可转到步进点 S40 执行。
- 并进式分歧的分歧回路数和合流回路数需一致，且最多是 8 条回路。

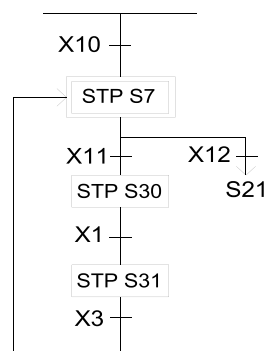
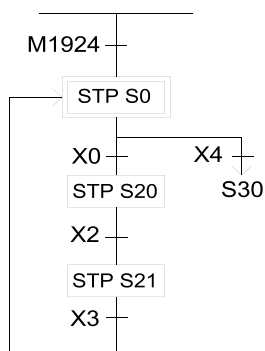
#### ④ 跳跃

### a. 同一步进流程



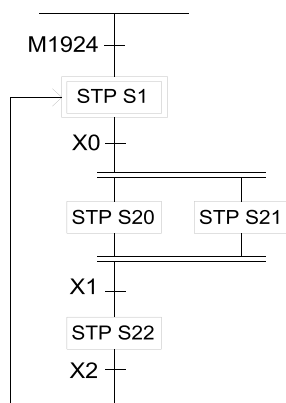
- 如左边步进点 S20 下有三条路径,假设 X2 ON, 则直接跳跃到步进点 S23 执行, 不需要选择性合流的过程。
- 但并进式分歧的路径不能跳跃执行。

### b.不同步进流程

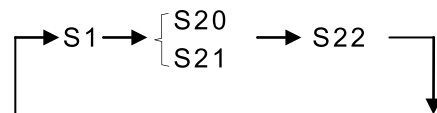


### ⑤ 闭环回路型和单循环型

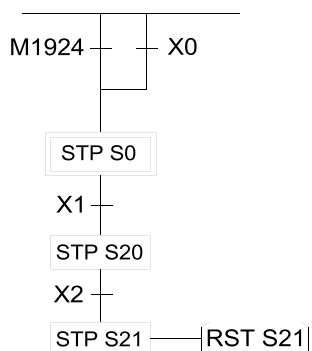
### a. 闭环回路型



- 一开机初始步进点 S1 ON, 往后只会如下做无限次循环。

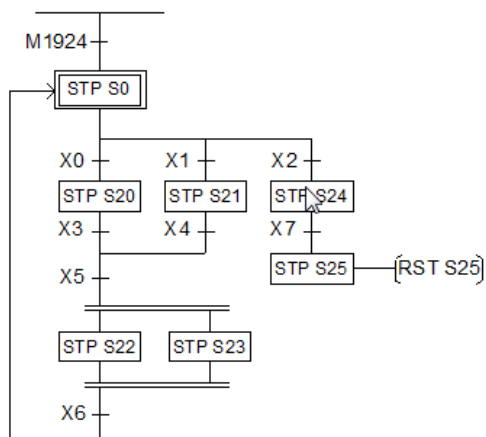


### b.单循环型

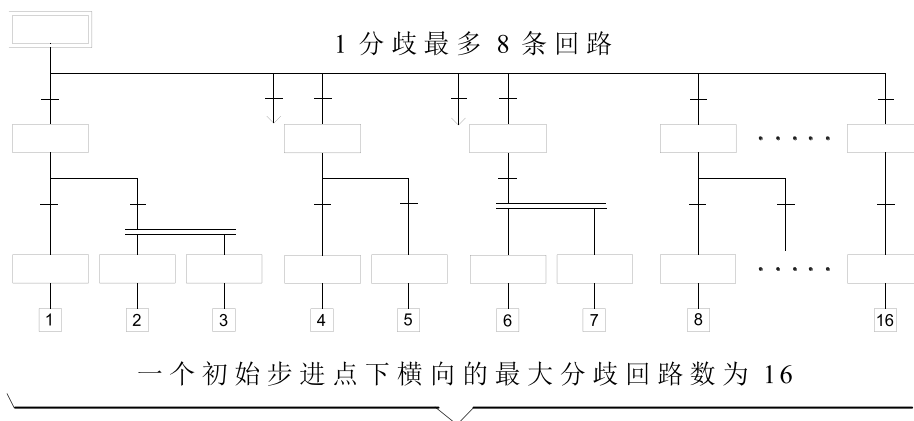


- 在步进点 S20 ON 时，如 X2 ON，S21 本应 ON，但被“RST S21”给 OFF，而结束此步进流程。

c.混合型流程



## ⑥ 综合应用



## 8.3 步进指令介绍：STP、FROM、TO、STPEND

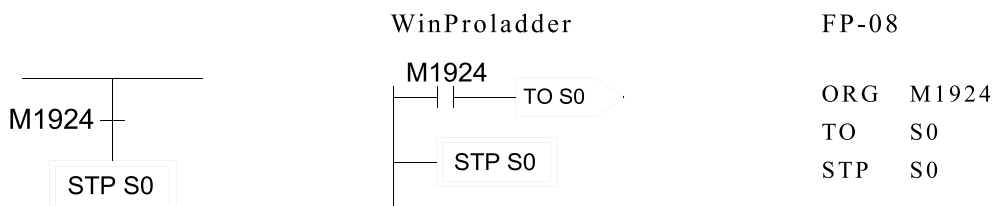
- **STP Sx** :  $S0 \leq Sx \leq S7$  (WinProladder 输入/显示格式)

或

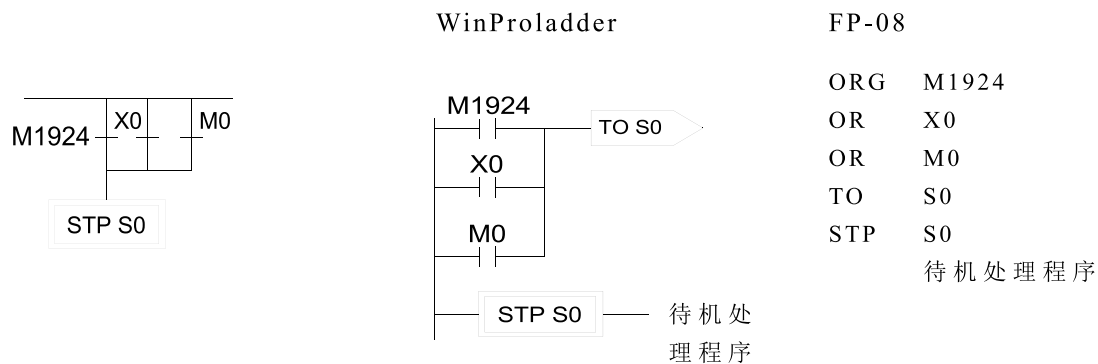
**STP Sx** :  $S0 \leq Sx \leq S7$  (FP-08 输入按 "STP" Key)

该指令为初始步进点 (Initial Step) 指令，由此指令才可往下衍生出各个机械流程的步进控制。FBs 系列最多可提供 8 个初始步进点，也就是说一台 PLC 最多可同时作 8 个流程控制。每一步进流程可独立运作或是产生运作结果供其它流程参考使用。

【范例一】每次开机启动初始步进点 S0



【范例二】每次开机或按手动钮或自动生产异常发生而在某特定时间内无人员处置自动进入初始步进点 S0 待机



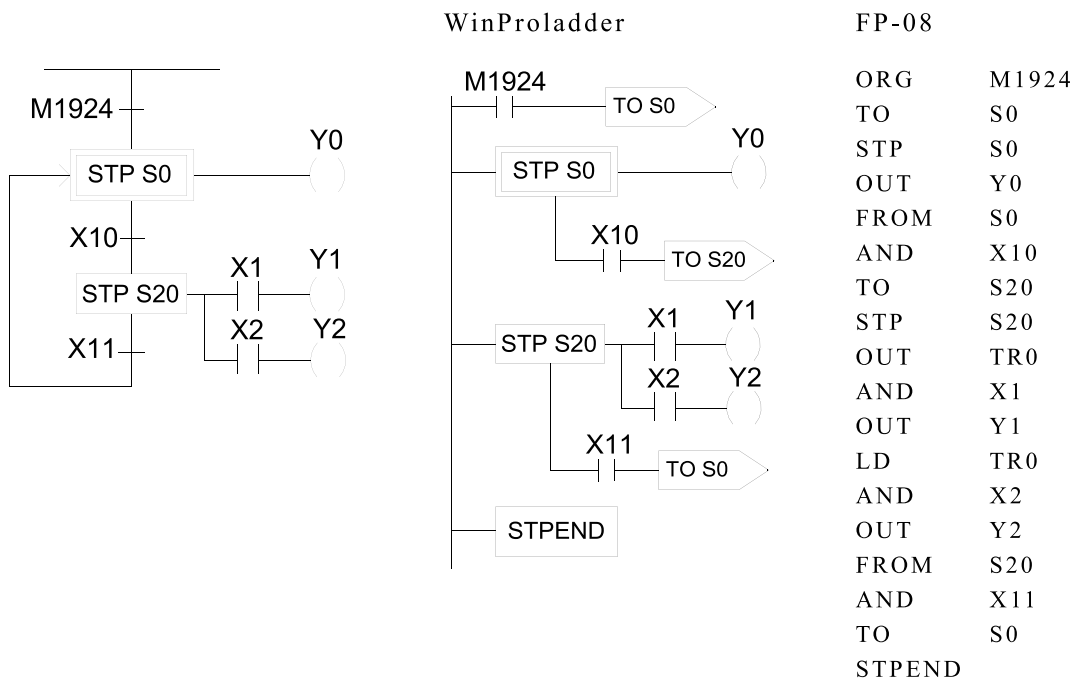
【说明】X0：手动钮；M0：异常的接点

- STP Sxxx

：S20≦Sxxx≦S999（WinProladder 输入/显示格式）
- 或
- STP Sxxx：S20≦Sxxx≦S999（FP-08 输入按“STP”Key）

该指令为流程中的步进点指令，每一步进点代表一个步序（站），ON 代表该步序作动，并会执行该步序下的梯形程序。

【范例】



- 【说明】1.开机时，初始步进点 S0 ON、Y0 ON。
- 2.当转进条件 X10（实际使用时，转进条件可由 X、Y、M、T、C 各接点的串、并联组合而成）ON 时，则步进点 S20 作动，当次扫描时间内系统会自动将 S0 OFF，且 Y0 自动清除为 OFF。

即 X10 ON⇒

S20 ON

S0 OFF

⇒

X1 ON→Y1 ON

X2 ON→Y2 ON

Y0 OFF

3.当转进条件 X11 ON 时，则步进点 S0 ON, Y0 也 ON, 同时 S20、Y1 和 Y2 变 OFF。

即  $X11\text{ ON} \Rightarrow \begin{cases} S0 & \text{ON} \\ S20 & \text{OFF} \end{cases} \Rightarrow \begin{cases} Y0 & \text{ON} \\ Y1 & \text{OFF} \\ Y2 & \text{OFF} \end{cases}$

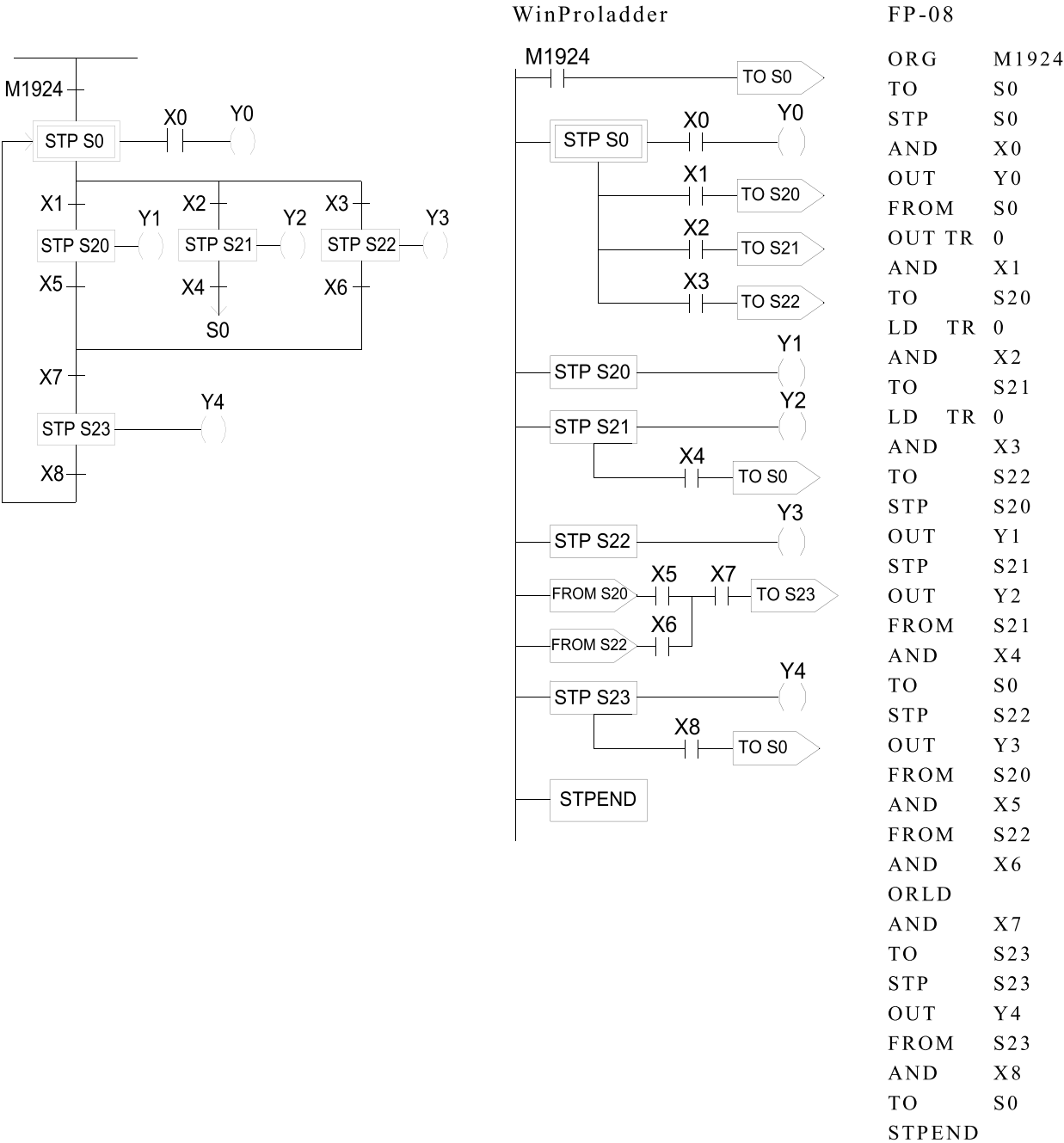
● FROM Sxxx : S0≡Sxxx≡S999 (WinProladder 输入/显示格式)

或

FROM Sxxx : S0≡Sxxx≡S999 (FP-08 输入按 "FROM" Key)

此指令描述转进的来源步进点，也就是要由步进点 Sxxx 配合转进条件前进到下一个步进点。

【范例】



**【说明】:** 1.开机时进入初始步进点 S0 ON; X0 ON 则 Y0 ON。

2. S0 ON 时, a.当 X1 ON 时, 则步进点 S20 ON、Y1 ON。

b.当 X2 ON 时, 则步进点 S21 ON、Y2 ON。

c.当 X3 ON 时, 则步进点 S22 ON、Y3 ON。

d.如果 X1、X2 和 X3 同时 ON, 则步进点 S20 优先 ON, S21 或 S22 不会 ON。

e.如果 X2 与 X3 同时 ON, 则步进点 S21 优先 ON, S22 不会 ON。

3. S20 ON, 当 X5 和 X7 同时 ON 时, 则步进点 S23 ON、Y4 ON、S20 OFF、Y1 OFF。

4. S21 ON, 当 X4 ON 时, 则步进点 S0 ON、S21 OFF、Y2 OFF。

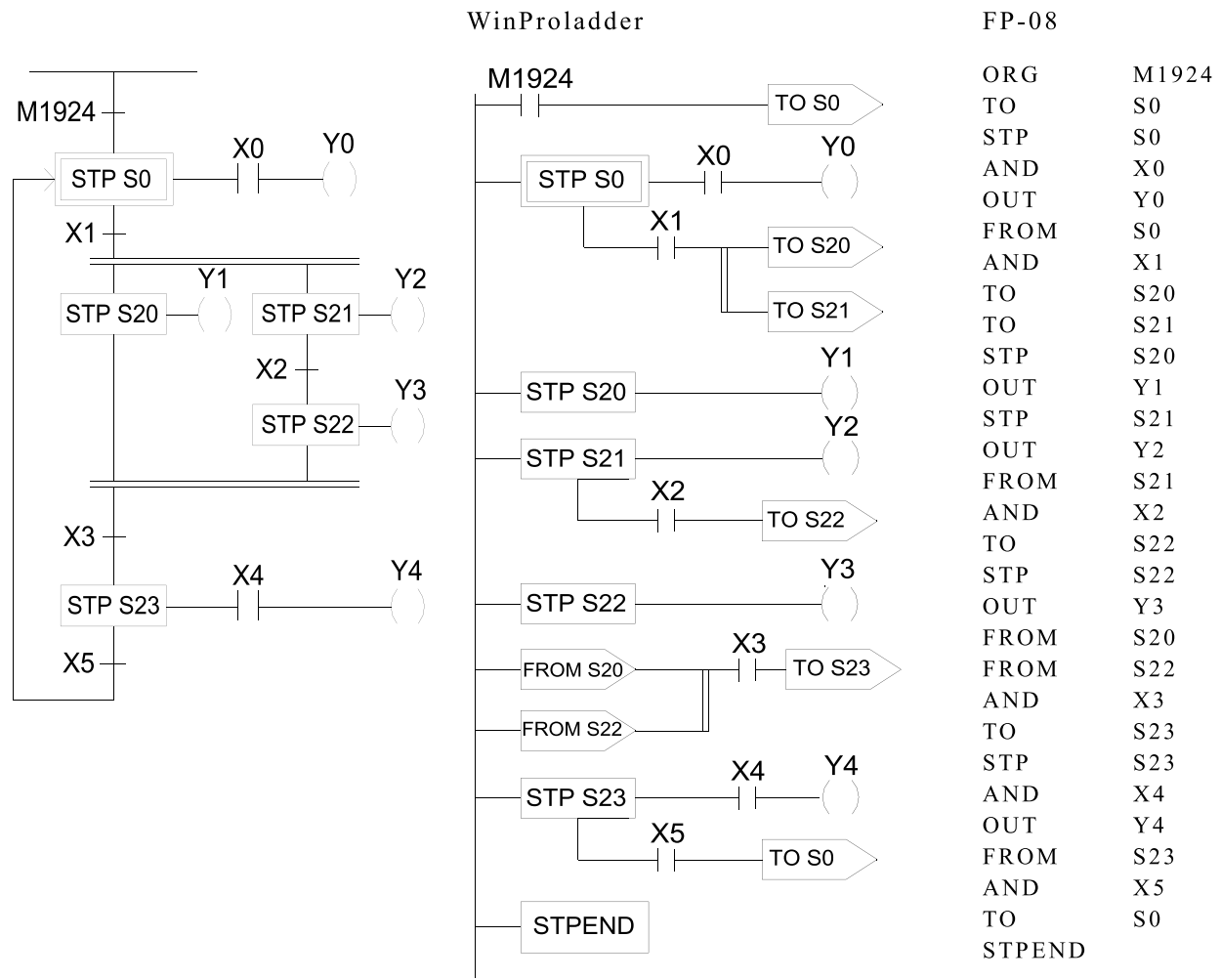
5. S22 ON, 当 X6 和 X7 同时 ON 时, 则步进点 S23 ON、Y4 ON、S22 OFF、Y3 OFF。

6. S23 ON, 当 X8 ON 时, 则步进点 S0 ON、S23 OFF、Y4 OFF。

- **TO Sxxx** :  $S0 \leq Sxxx \leq S999$  (Winproladder 输入/显示格式)  
或  
**TO Sxxx** :  $S0 \leq Sxxx \leq S999$  (FP-08 输入按 "TO" Key)

该指令描述要转往的步进点。

【范例】





● **STPEND** : (WinProladder 输入/显示格式)

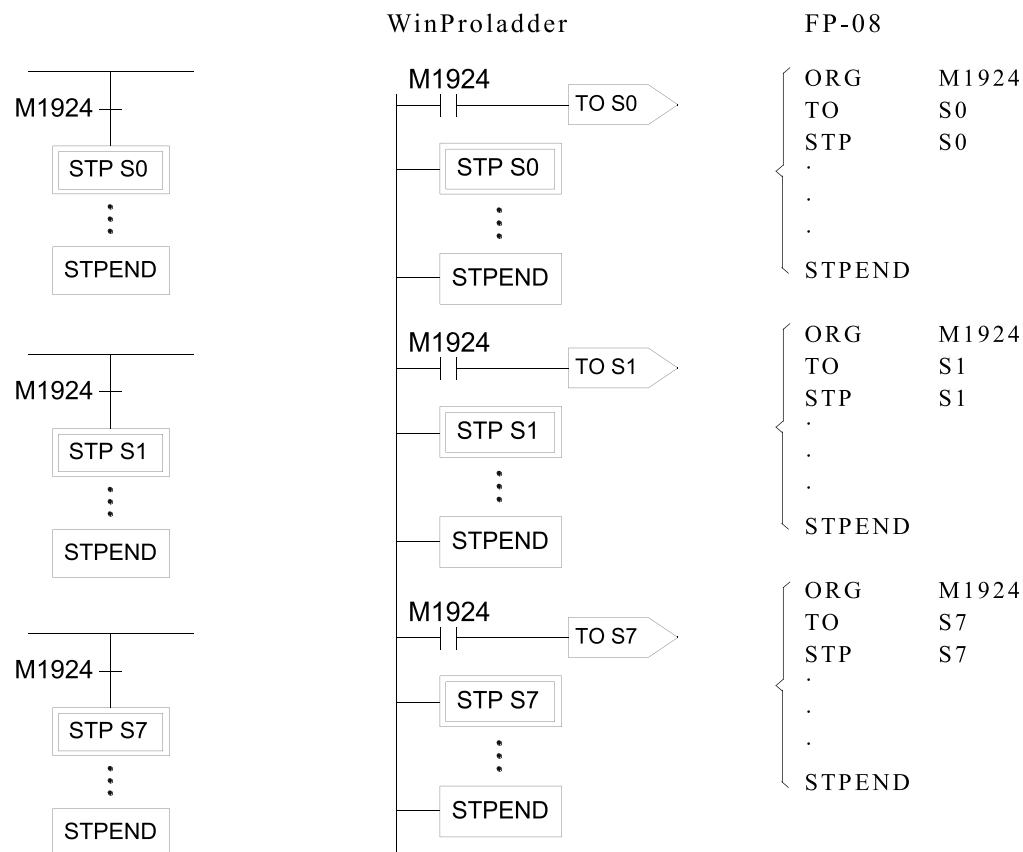
或

**STPEND** : (FP-08 输入按 “STP” 和 “END” Key)

该指令代表一个流程指令的结束，必须有此指令，所有流程才会正确运作。

PLC 最多有 8 个步进流程 (S0~S7) 可同时控制，所以最多有 8 个 **STPEND** 指令。

【范例】

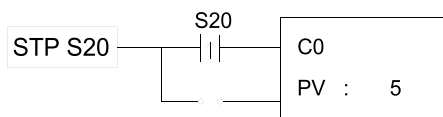


【说明】开机时 8 个步进流程同时作动。

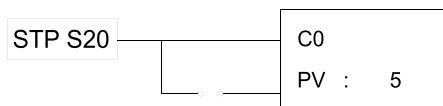
## 8.4 步进梯形图写法

### 【注意事项】

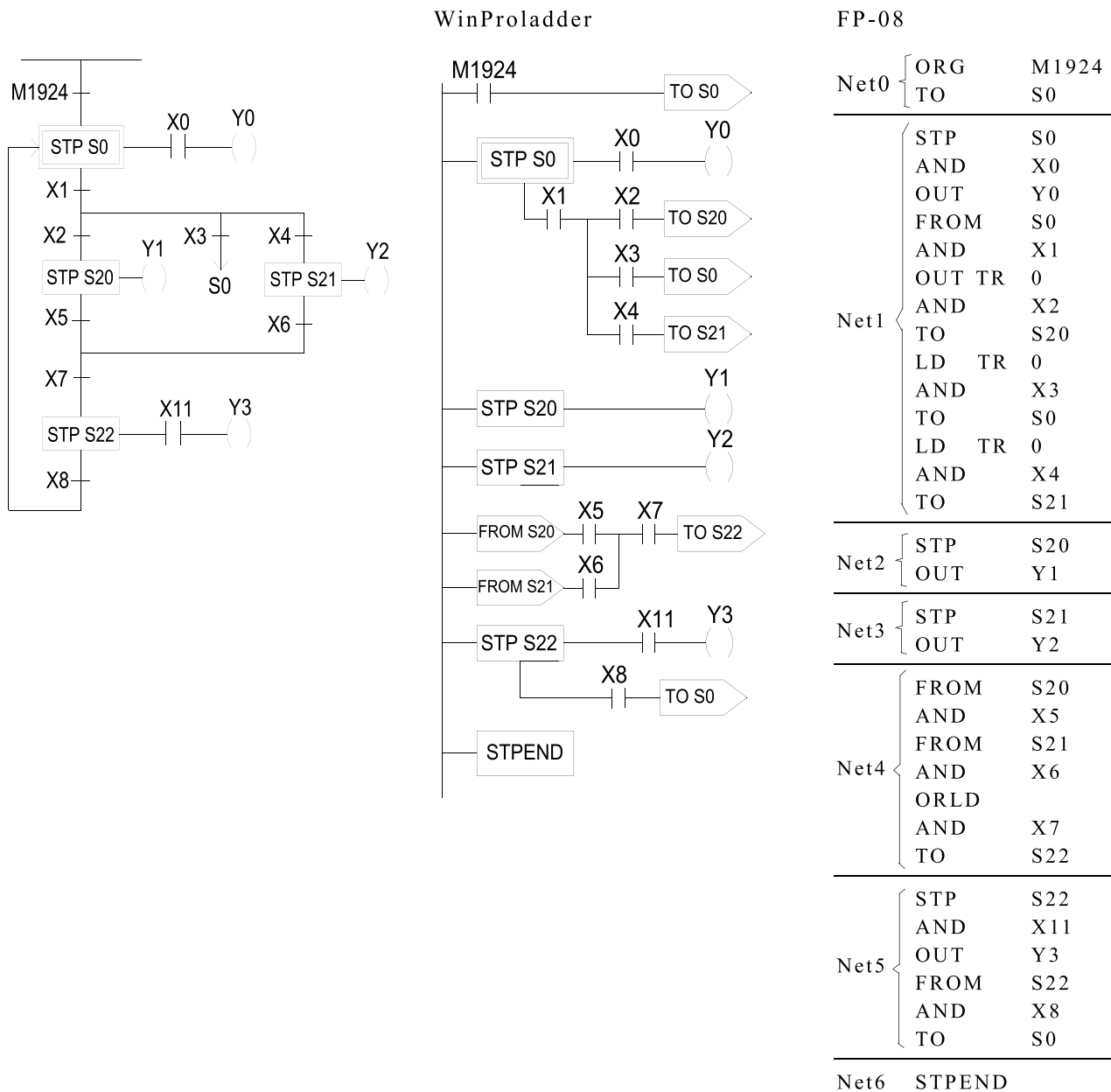
- 在实际的应用上，可将单纯的梯形图和步进梯形图组合使用。
- 作为开头的步进点我们称为初始步进点，共有 8 点，即 S0~S7。
- 要让初始步进点作动当然可以由任何一个步进点来加以触发，但 PLC 开始运转时，必须让初始步进点 ON；我们可利用系统提供的 M1924（第一次扫描 ON 信号）来触发初始步进点 ON。
- 除了初始步进点用上述方法触发启动外，其它的步进点的触发必须由另外一个步进点来驱动。
- 在步进梯形图程序当中必须有开头的初始步进点，及最后的 STPEND 指令，才算一个完整的步进流程程序。
- 一般步进点共有 980 点，由 S20~S999，可任意使用，无须按顺序，但号码绝不可重复使用。系统默认 S500~S999 为停机保持型（当然可由 USER 修改），机械流程在断电后如想继续断电前的动作，则可使用这些步进点。
- 一个步进点在基本上必须具备驱动步进点内输出负载、指定转进条件及转进目的地等三个功能。
- 在步进程序中不可使用 MC，SKP 指令；子程序区不能输入步进程序。当然 JMP 指令尽量少用。
- 如果步进转进后，输出点仍需保持 ON 则需用 SET 指令推动该输出点；要清除该输出点为 OFF，则需用 RST 指令。
- 从一个初始步进点往下看，横向分歧步进点最大 16 点，但一个分歧点最多只可作 8 个分歧回路。
- M1918=0（默认值）时，在 MC（FUN 0）或步进点程序中如需使用 PULSE 型功能指令，则必须在该功能指令前串接一个该步进点的 TU 指令，例如



M1918=1 时，则不需加该步进点的 TU 指令，例如：

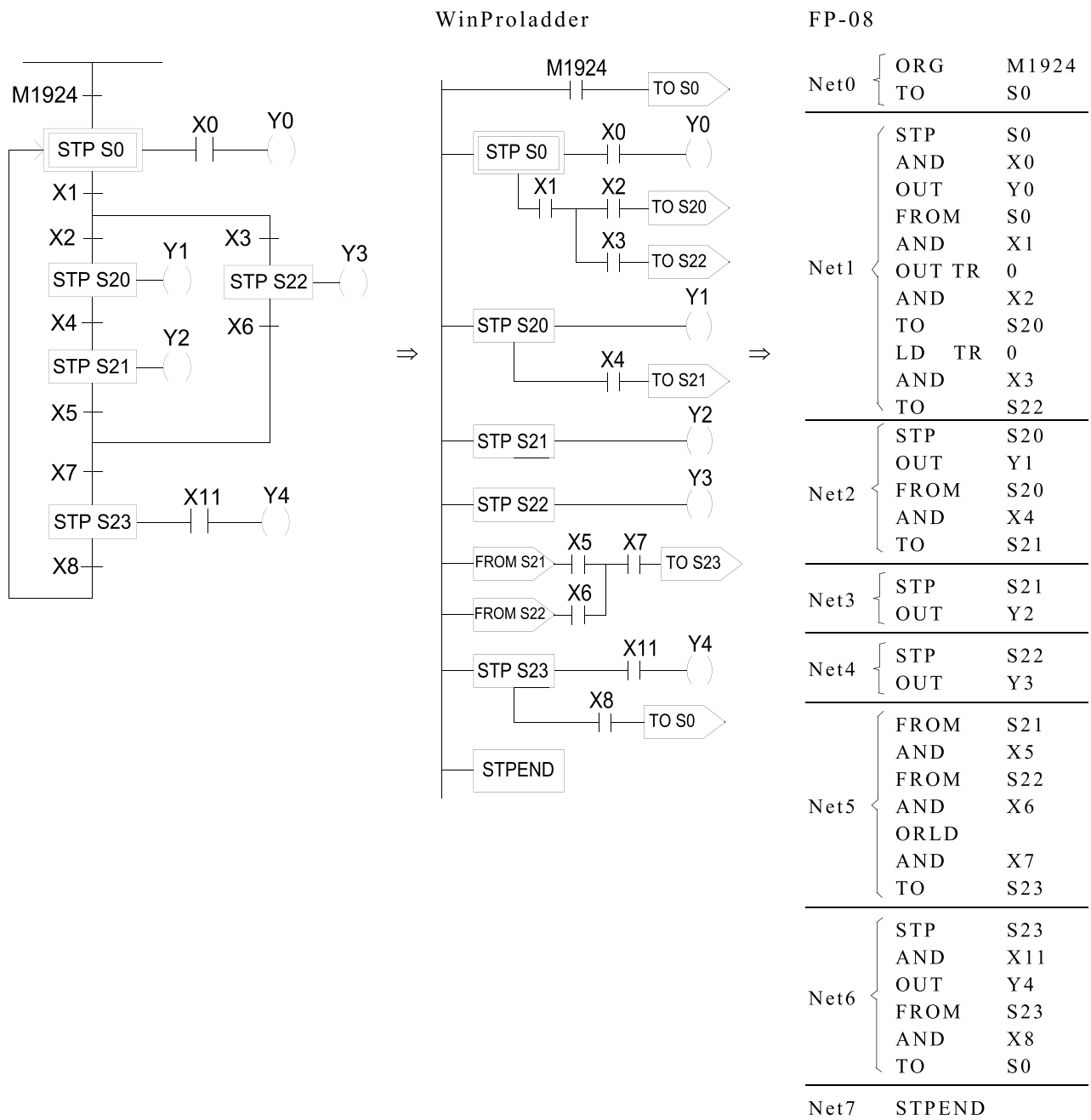


【范例 1】



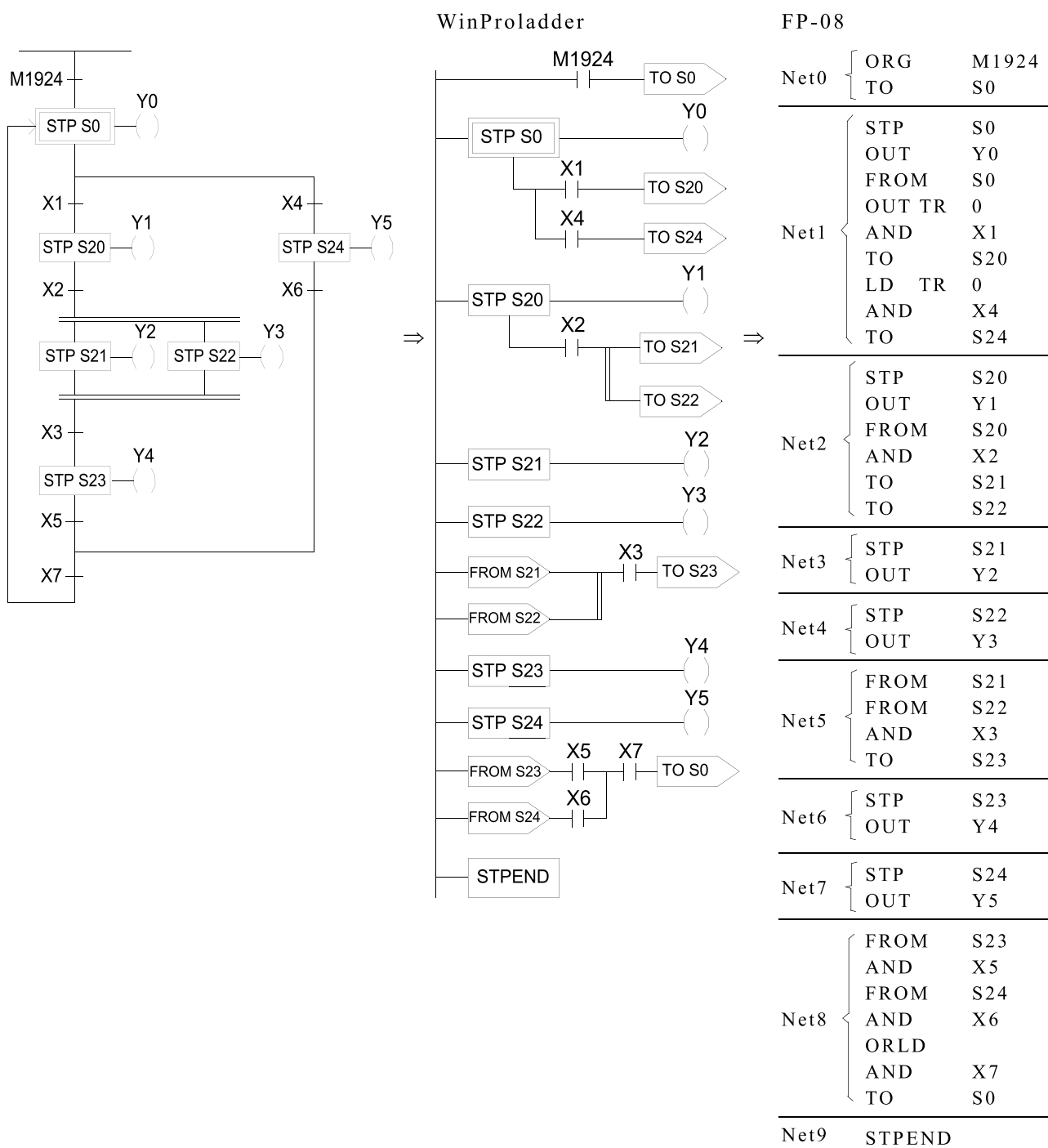
- 【说明】:
- 1.编辑初始步进点 S0
  - 2.编辑 S20、S0、S21 的分歧
  - 3.编辑 S20
  - 4.折返编辑 S21
  - 5.编辑 S20、S21 的合流
  - 6.往下编辑 S22

【范例 2】



- 【说明】:
- 1.编辑初始步进点 S0
  - 2.编辑 S20、S22 的分歧
  - 3.编辑 S20
  - 4.编辑 S21
  - 5.折返编辑 S22
  - 6.编辑 S21、S22 的合流
  - 7.往下编辑 S23

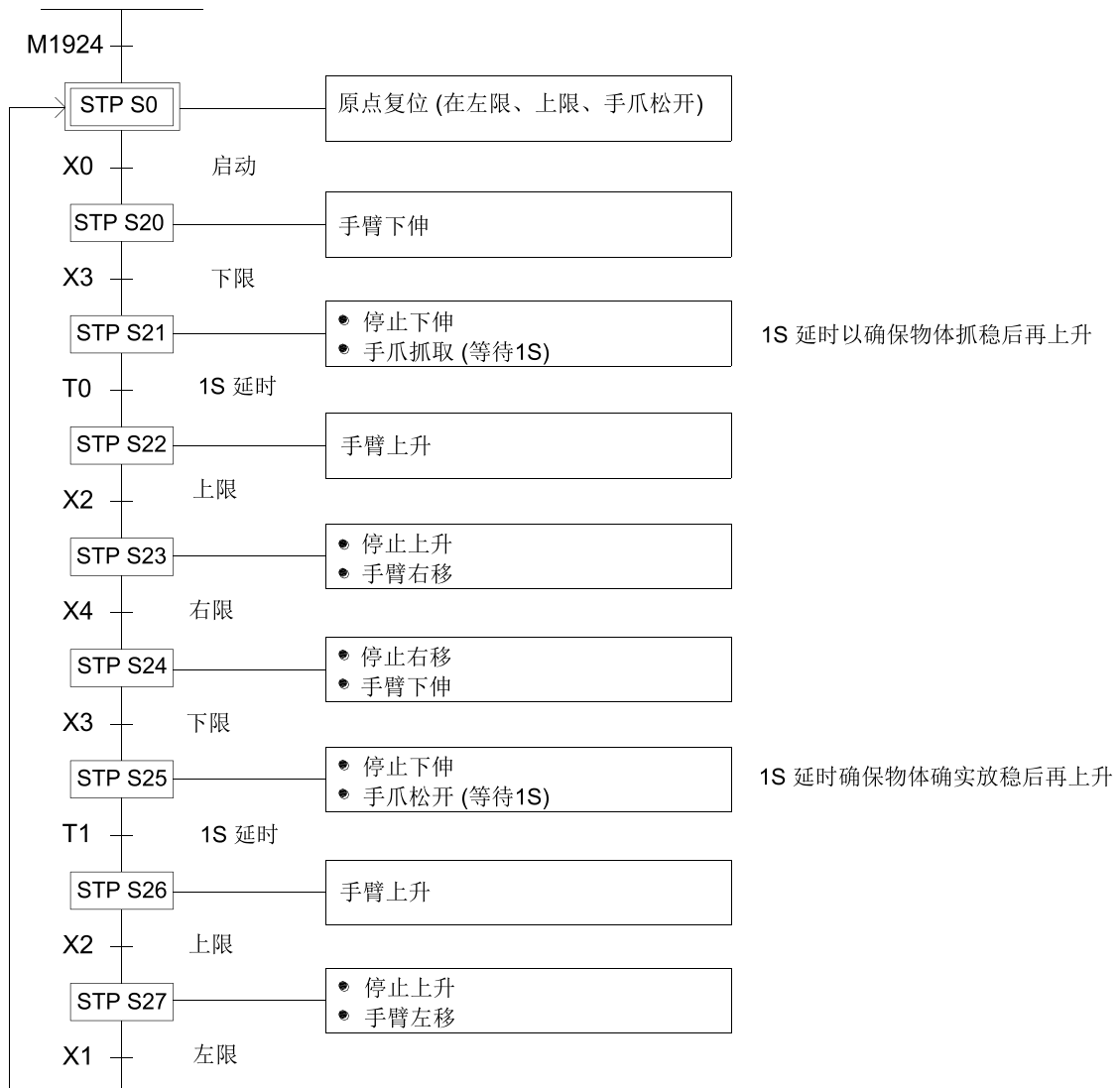
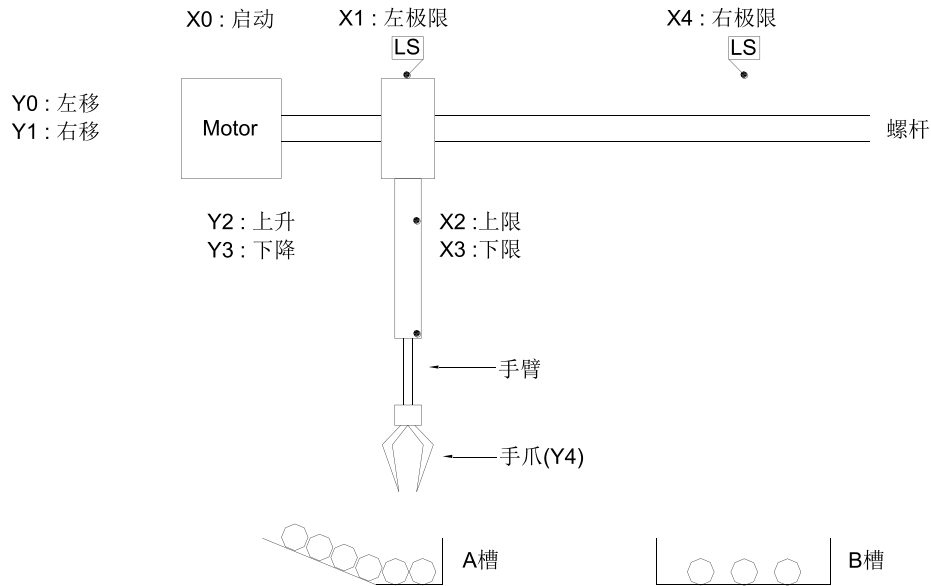
### 【 范例 3 】

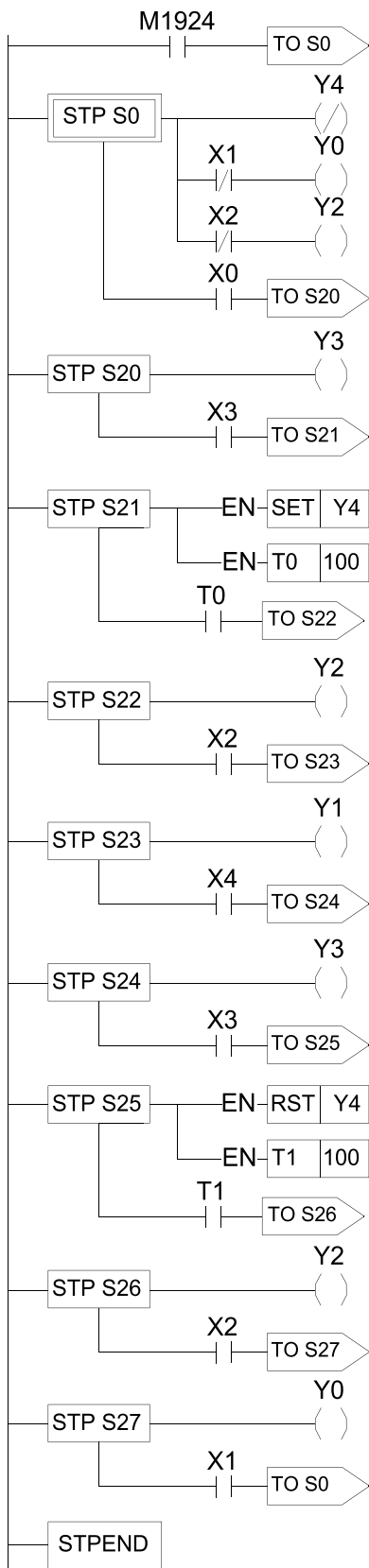


- 【 说明 】:
- 1.编辑初始步进点 S0
  - 2.编辑 S20、S24 的分歧
  - 3.编辑 S20
  - 4.编辑 S21、S22 的分歧
  - 5.编辑 S21
  - 6.折返编辑 S22
  - 7.编辑 S21、S22 的合流
  - 8.编辑 S23
  - 9.返回上层编辑 S24
  - 10.编辑 S23、S24 的合流

## 8.5 实际应用范例

### 【范例 1】从 A 槽抓取物体放到 B 槽内





释放手爪

回左限

回上限

启动开关 ON 后  
移行到 S20

手臂下伸

伸至下限后  
移行到 S21手爪抓取(因用 SET 指令  
故 STP S21 离开后, Y4  
仍保持 ON)

1 秒后转进 S22

手臂上升

到上限后转进 S23

手臂右移

移到右限后转进 S24

手臂下伸

伸到下限后转进 S25

手爪松开

1 秒钟延时

1 秒钟后转进 S26

手臂上升

升到上限后转进 S27

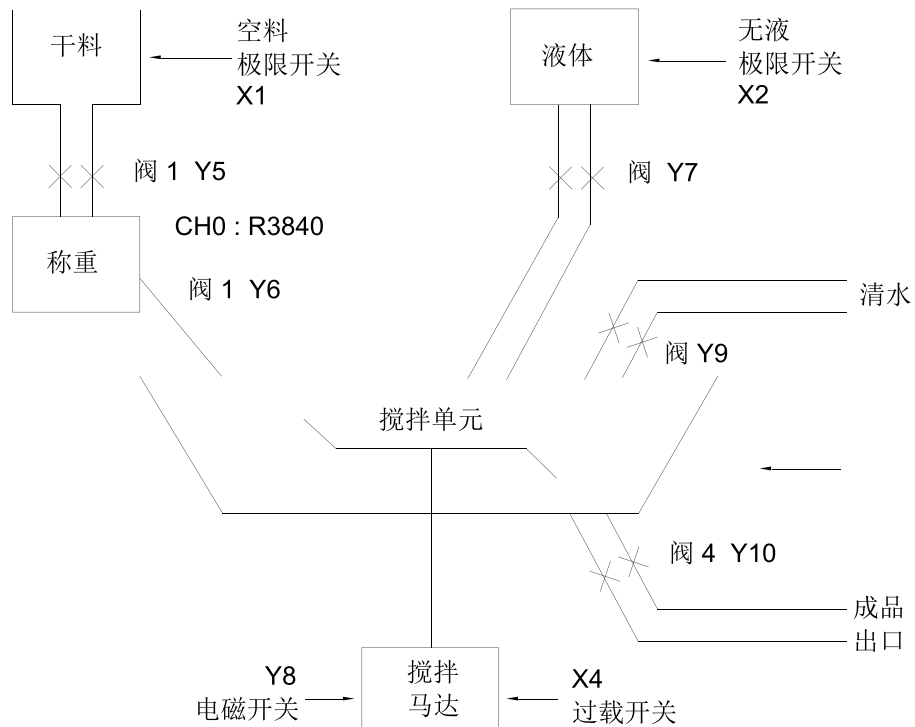
手臂左移

等待到左限后, 转进  
S0(一个完整 CYCLE)

```

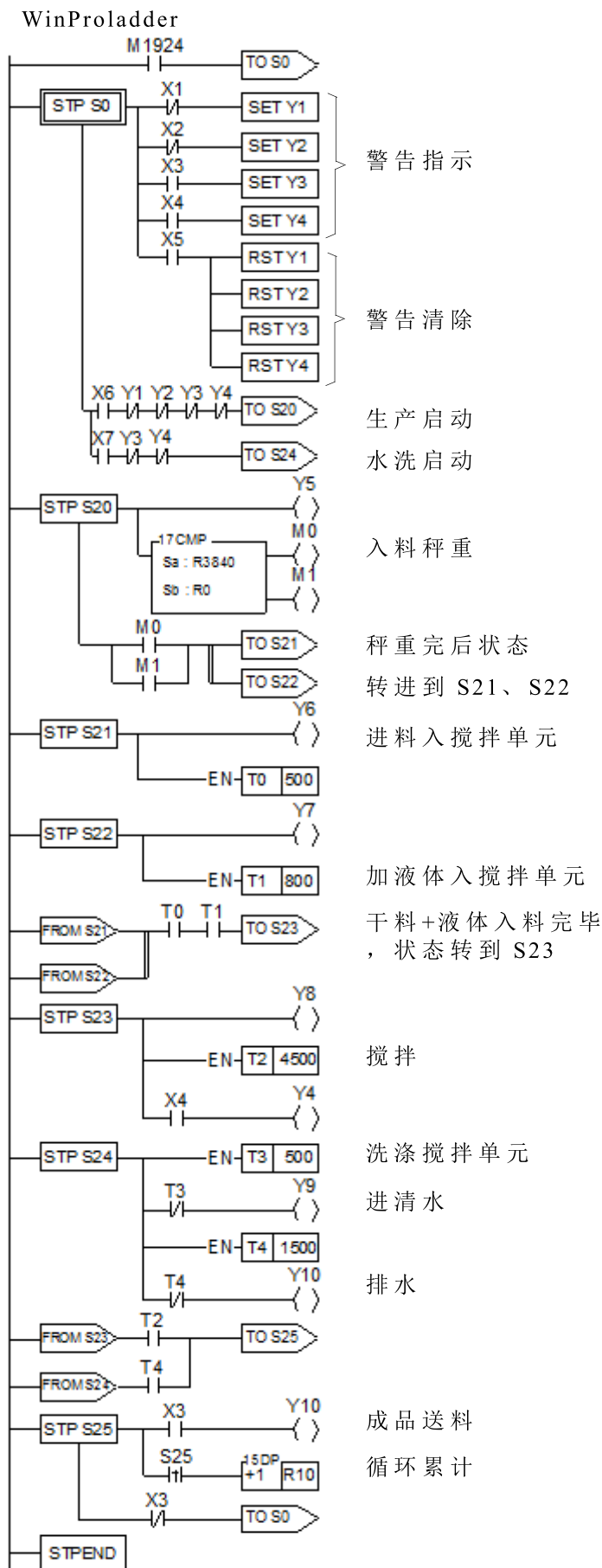
ORG      M1924
TO       S0
STP      S0
OUT TR   0
OUT NOT  Y4
AND NOT  X1
OUT      Y0
LD TR    0
AND NOT  X2
OUT      Y2
FROM     S0
AND      X0
TO       S20
STP      S20
OUT      Y3
FROM     S20
AND      X3
TO       S21
STP      S21
SET      Y4
T0       PV: 100
FROM     S21
AND      T0
TO       S22
STP      S22
OUT      Y2
FROM     S22
AND      X2
TO       S23
STP      S23
OUT      Y1
FROM     S23
AND      X4
TO       S24
STP      S24
OUT      Y3
FROM     S24
AND      X3
TO       S25
STP      S25
RST      Y4
T1       PV: 100
FROM     S25
AND      T1
TO       S26
STP      S26
OUT      Y2
FROM     S26
AND      X2
TO       S27
STP      S27
OUT      Y0
FROM     S27
AND      X1
TO       S0
STPEND
  
```

【范例 2】液体搅拌处理



- ◆ 输入点：
  - 空料极限开关 X1
  - 无液极限开关 X2
  - 空料极限开关 X3
  - 过载开关 X4
  - 警告清除钮 X5
  - 启动钮 X6
  - 水洗钮 X7
- ◆ 警告指示灯：
  - 干料空料 Y1
  - 液体缺液 Y2
  - 搅拌单元空料 Y3
  - 马达过载 Y4
- ◆ 输出点：
  - 干料入料阀 Y5
  - 干料入料阀 Y6
  - 液体入料阀 Y7
  - 启动马达电磁阀 Y8
  - 清水入水阀 Y9
  - 成品送料阀 Y10
- ◆ 秤重输入：CH0 (R3840)
- ◆ M1918=0

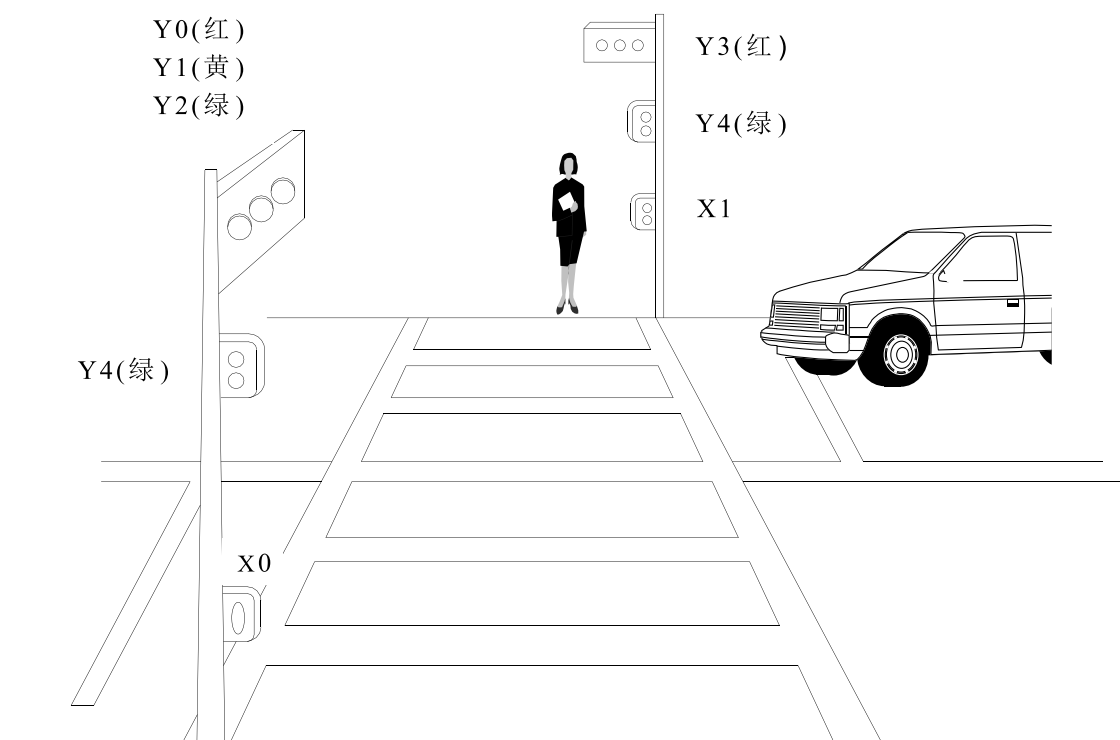




FP-08

ORG	M1924	STP	S22
TO	S0	OUT	Y7
STP	S0	T1	PV: 800
OUT TR	0	FROM	S21
AND NOT	X1	FROM	S22
SET	Y1	AND	T0
LD TR	0	AND	T1
AND NOT	X2	TO	S23
SET	Y2	STP	S23
LD TR	0	OUT TR	0
AND	X3	OUT	Y8
SET	Y3	LD TR	0
LD TR	0	T2	PV: 4500
AND	X4	LD TR	0
SET	Y4	AND	X4
LD TR	0	OUT	Y4
AND	X5	STP	S24
RST	Y1	OUT TR	0
RST	Y2	T3	PV: 500
RST	Y3	LD TR	0
RST	Y4	AND NOT	T3
FROM	S0	OUT	Y9
OUT TR	1	LD TR	0
AND	X6	T4	PV: 1500
AND NOT	Y1	LD TR	0
AND NOT	Y2	AND NOT	T4
AND NOT	Y3	OUT	Y10
AND NOT	Y4	FROM	S23
TO	S20	AND	T2
LD TR	1	FROM	S24
AND	X7	AND	T4
AND NOT	Y3	ORLD	
AND NOT	Y4	TO	S25
TO	S24	STP	S25
STP	S20	OUT TR	0
OUT	Y5	AND	X3
FUN	17	OUT	Y10
Sa:R3840		LD TR	0
Sb:R0		AND TU	S25
FO	0	FUN	15DP
OUT	M0	D:R10	
FO	1	FROM	S25
OUT	M1	AND NOT	X3
FROM	S20	TO	S0
LD	M0	STPEND	
OR	M1		
ANDLD			
TO	S21		
TO	S22		
STP	S21		
OUT	Y6		
T0	PV: 500		

【范例 3】人行横道红绿灯

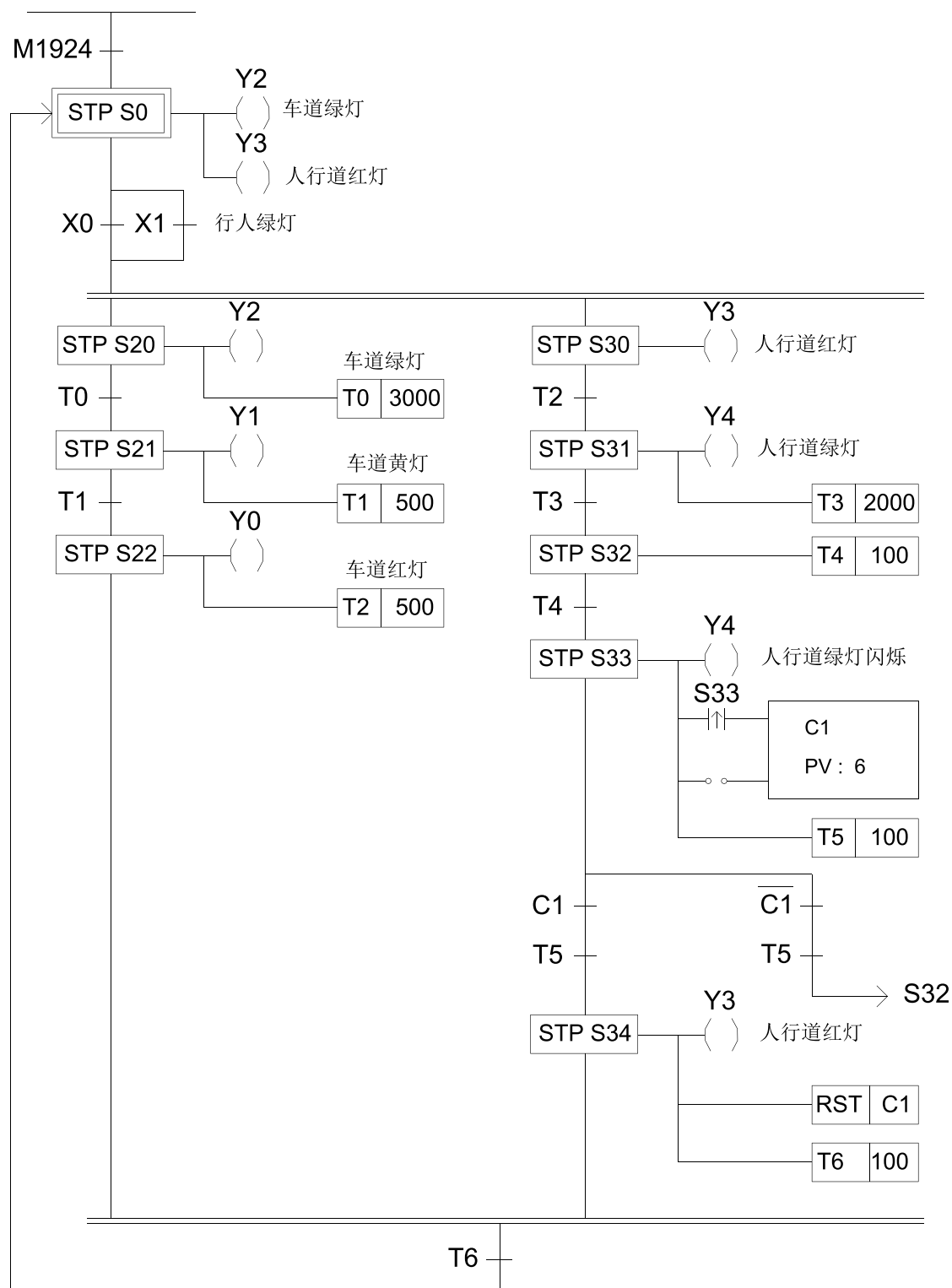


◆ 输入点：行人按钮 X0  
行人按钮 X1

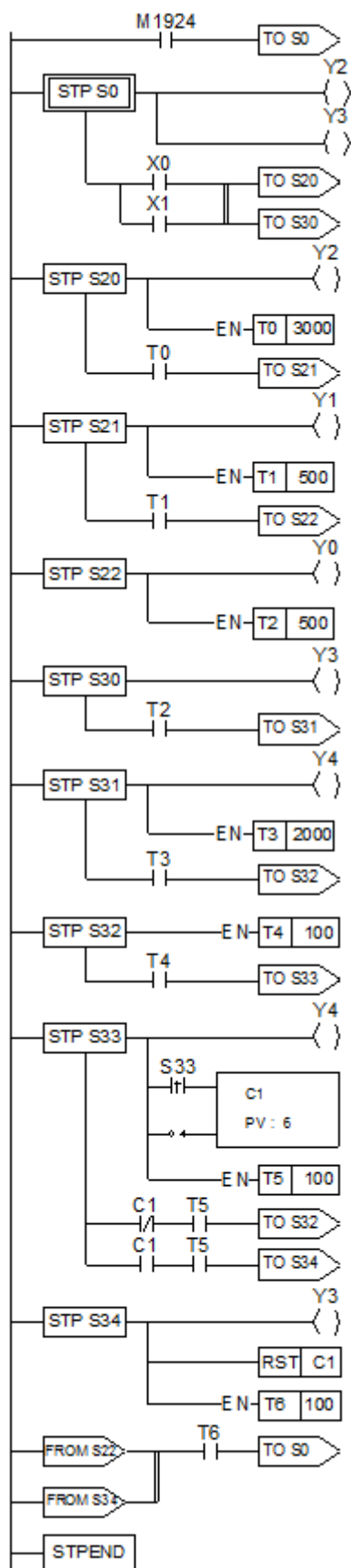
◆ 输出点：车道红灯 Y0  
车道黄灯 Y1  
车道绿灯 Y2  
人行横道红灯 Y3  
人行横道绿灯 Y4

◆ M1918=0

人行横道红绿灯控制流程图



● 人行横道红绿灯控制程序  
WinProladder



FP-08

```

ORG      M1924
TO       S0
STP      S0
OUT      Y2
OUT      Y3
FROM     S0
LD       X0
OR       X1
ANDLD
TO       S20
TO       S30
STP      S20
OUT      Y2
T0      PV: 3000
FROM     S20
AND      T0
TO       S21
STP      S21
OUT      Y1
T1      PV: 500
FROM     S21
AND      T1
TO       S22
STP      S22
OUT      Y0
T2      PV: 500
STP      S30
T2
TO       S31
STP      S31
EN-T3 2000
T3
TO       S32
STP      S32
EN-T4 100
T4
TO       S33
STP      S33
S33
C1      PV: 6
EN-T5 100
C1 T5
TO       S32
C1 T5
TO       S34
STP      S34
RST     C1
EN-T6 100
T6
FROM     S22
FROM     S33
STPEND

```

```

STP      S32
T4      PV:100
FROM     S32
AND      T4
TO       S33
STP      S33
OUT TR 0
OUT      Y4
LD       TR 0
AND TU   S33
LD       OPEN
C1      PV:6
LD       TR 0
T5      PV:100
FROM     S33
OUT TR 1
AND NOT C1
AND      T5
TO       S32
LD       TR 1
AND      C1
AND      T5
TO       S34
STP      S34
OUT      Y3
RST     C1
T6      PV:100
FROM     S22
FROM     S34
AND      T6
TO       S0
STPEND

```

## 8.6 步进程序语法检查错误码说明

步进语言程序、语法检查错误的编号如下：

- E51 : TO(S0~S20)必须以 ORG 为起始指令
- E52 : TO(S20~S999)不得以 ORG 为起始指令
- E53 : 同一个网络中, TO(S20~S999)前, 必需在有 FROM
- E54 : TO 之前一个指令, 必需为 TO、AND、OR、ANDLD、ORLD
- E56 : 此时 FROM 之前一个指令, 必需为 FROM 或 AND、OR、ANDLD、ORLD
- E57 : OUT、TMR、CTR、FUN 不与 TO(S0~S19)并存在同一个网络中
- E58 : OUT、TMR、CTR、FUN 前一个 STEP 指令必需为 STP
- E59 : 同一个网络中, TO 超过 8 个
- E60 : 同一个网络中, FROM 超过 8 个
- E61 : TO(S0~S19) 必需为网络第一列
- E62 : 接点占据 TO 位置
- E71 : 连续不完整(理应不会发生)
- E72 : TO Sxx 重复
- E73 : STP Sxx 重复
- E74 : FROM Sxx 重复
- E76 : 上一个 STP(S0~S19)缺乏 STPEND 或 STPEND 往前找不到相对应的 STP(S0~S19)
- E77 : STP(S0~S19)的前一个网络并非是以 ORG 为起始唯一的 TO(S0~S19)
- E78 : 尚未使用 STP(S0~S19)就使用 TO(S20~S999)、STP (S20~S999)、FROM
- E79 : 尚未使用 TO Sxx 就使用 STP Sxx 或 FROM Sxx
- E80 : 尚未使用 STP Sxx 就使用 FROM Sxx
- E81 : 同一时间, 尚未处理的分歧层数不得大于 16
- E82 : 同一时间, 分歧中尚未处理的分枝不得大于 16
- E83 : 单一步进点, 未依照 TO Sxx→STP Sxx→FROM Sxx 的顺序且紧密连续
- E84 : 进入分歧后, 需按照由左至右的顺序来处理分枝
- E85 : 合流时, 与先前的分歧情况不对应
- E86 : 尚未利用 TO 来完成上一个合流, 就使用 STP 或 FROM
- E87 : 尚未利用 FROM+TO 来转移上一个 STP, 就使用 STP 或 FROM
- E88 : 分歧中, STP Sxx 或 FROM Sxx, 在此分歧内, 往前找不到相对应的 TO Sxx
- E89 : 尚未利用 STP 来承接 TO 的处理, 就使用 FROM
- E90 : 并进式分歧的转接不合法
- E91 : 上一个 STP(S0~S19)尚未处理完全, 就使用 ORG、LBL、RTS、RTI、MCE、SKPE、FOR、NEXT、END