



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 第3章 PLC基本指令及编程





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 3.1 基本逻辑指令





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 逻辑取及线圈驱动

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
LD	取指令	X,Y,M,S,T,C	1
LDI	取反指令	X,Y,M,S,T,C	1
OUT	线圈驱动指令	Y,M,S,T,C	Y,M: 1; S,特殊M: 2; T: 3; C: 3~5

## 2. 指令功能及说明

- LD, LDI指令用于将**常开/常闭触点**连接到母线上, 其他用法与后述的ANB指令组合, 在**分支起点处**也可使用
- OUT指令是对输出继电器、辅助继电器、状态、定时器、计数器的**线圈驱动指令**
- 并列的OUT命令能**多次连续使用**
- 对于定时器的计时线圈或计数器线圈, 使用OUT指令后, **必须设定常数K**

# 定时器、计数器的设定与步数

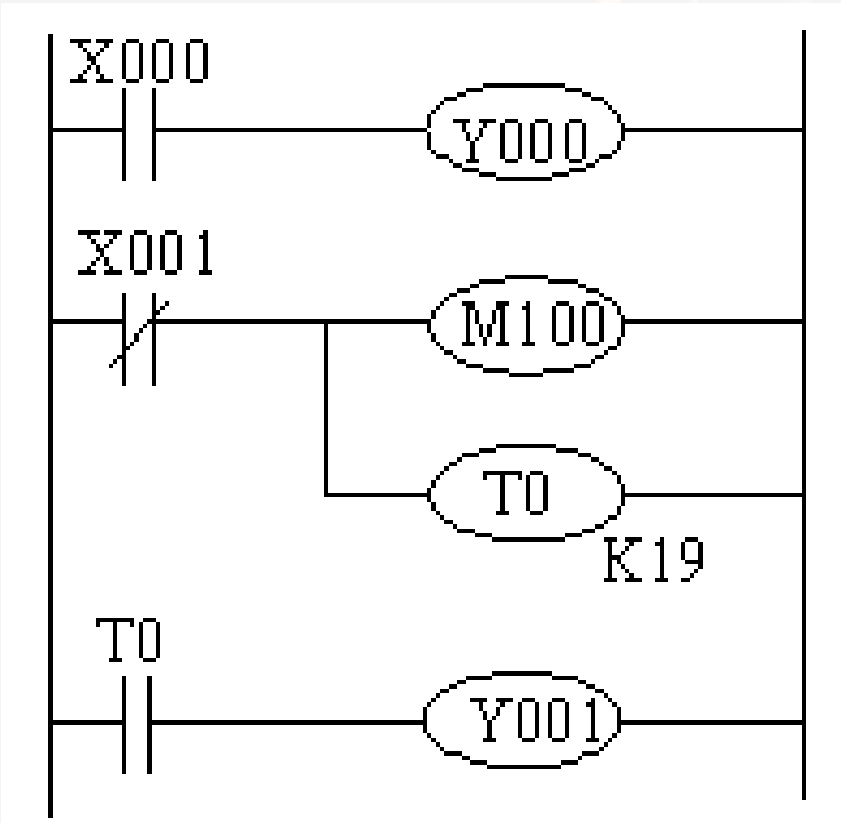
定时器 计数器	K的设定范围	实际的设定值	步数
1ms定时器	1~32,767	0.001~32.767	3
10ms定时器	1~32,767	0.01~327.67	3
100ms定时器		0.1~3276.7	
16位计数器	1~32,767	同左	3
32位计数器	-2,147,483,648~ +2,147,483,647	同左	5



# ◆【例】 LD, LDI及OUT指令的梯形图和指令程序



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X000; X000常开触点与母线连接

1 OUT Y000; 驱动Y000

2 LDI X001; X001常闭触点与母线连接

3 OUT M100; 驱动M100

4 OUT T0; 驱动T0

K 19; 设定时常数

7 LD T0; T0常开触点与母线连

8 OUT Y001; 驱动Y001

占用3步

输入输出？



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 触点串联指令



# 触点串联指令**AND, ANI**

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
AND	与	X,Y,M,S,T,C	1
ANI	与非	X,Y,M,S,T,C	1

□ 注：使用M1536-M3071时，程序步加1

## 2. 指令功能及说明

- 用AND/ANI指令可串联连接单个常开/常闭触点。串联触点数量不受限制，该指令可多次使用。
- OUT指令后，通过触点对其他线圈使用OUT指令，称之为**纵接输出**。若驱动顺序**相反**时，则必须使用（**栈存储**）**MPS指令**。



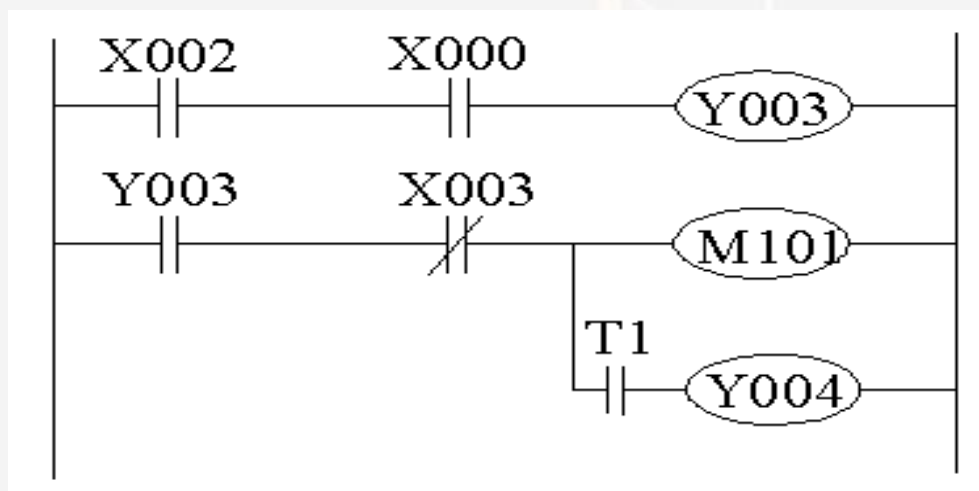


图3.2 AND,ANI指令

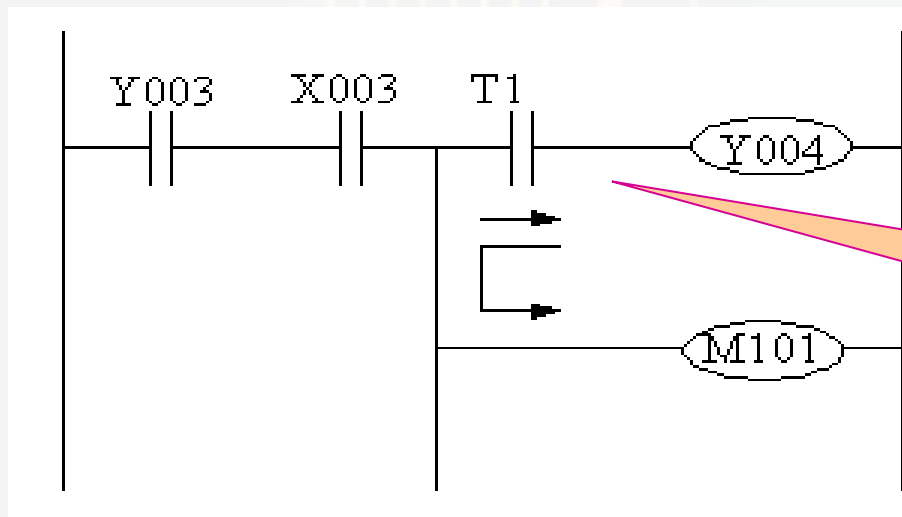


图3.3 改变图3.2中的Y004驱动顺序

0 LD X002

1 AND X000; 串联触点

2 OUT Y003

3 LD Y003

4 ANI X003; 串联触点

5 OUT M101

6 AND T1; 串联触点

7 OUT Y004; 纵接输出

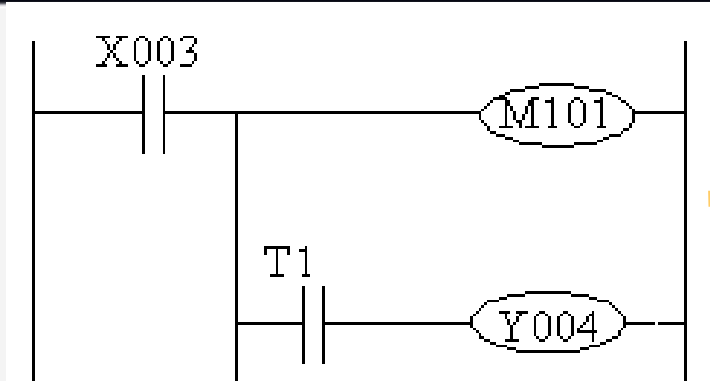
使用MPS 指令

# 比较图A和 图B中的Y004的驱动顺序



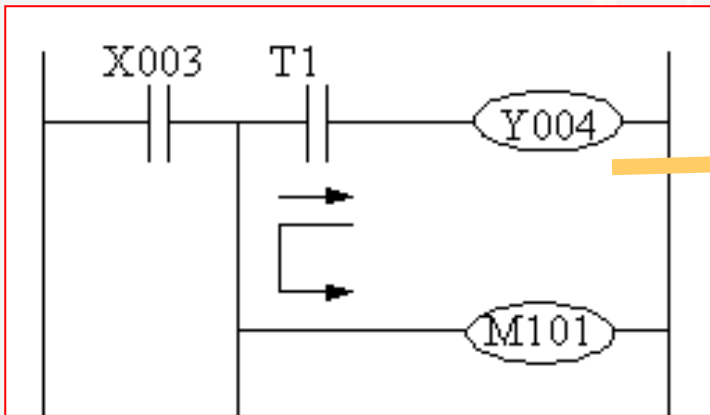
武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

图A

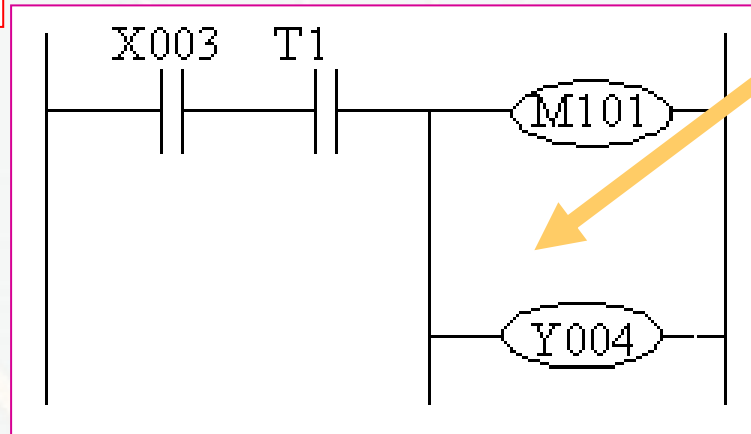


1	LD X003
2	OUT M101
3	AND T1
4	OUT Y004

图B



1	LD X003
2	AND T1
3	OUT Y004
4	OUT M101







武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 触点并联指令OR,ORI

# 触点并联指令**OR,ORI**

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
OR	或	X,Y,M,S,T,C	1
ORI	或非	X,Y,M,S,T,C	1

✓ 注：使用M1563-M3071时，程序步加1



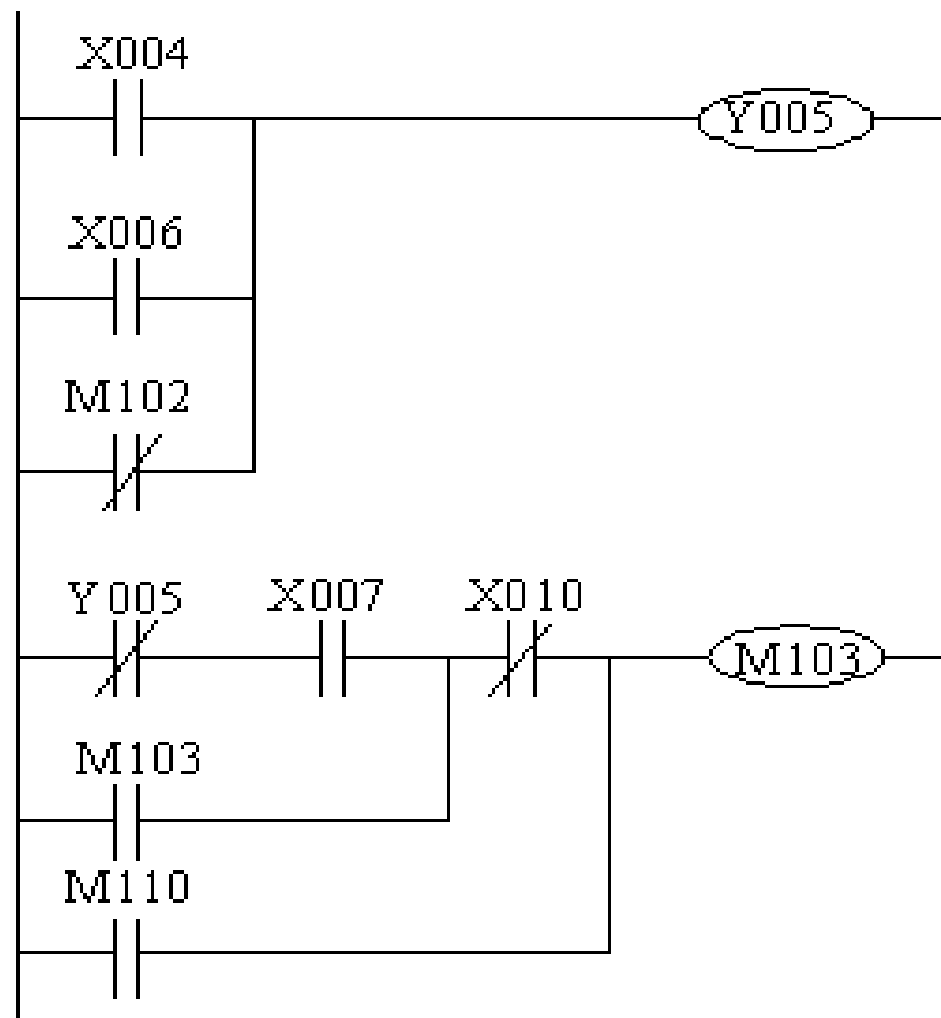
## 2. 指令功能及说明

- OR, ORI被用作**单个常开/常闭触点**的并联连接指令。
- OR, ORI是指**从该指令的步**开始, 与前述的LD、LDI指令步进行**并联**连接。

# 【例】 OR/ORI指令应用的梯形图和指令程序



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X004

1 OR X006

2 ORI M102; 并联连接

3 OUT Y005

4 LDI Y005

5 AND X007

6 OR M103

7 ANI X010

8 OR M110; 并联连接

9 OUT M103

图3.4 OR,ORI指令



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

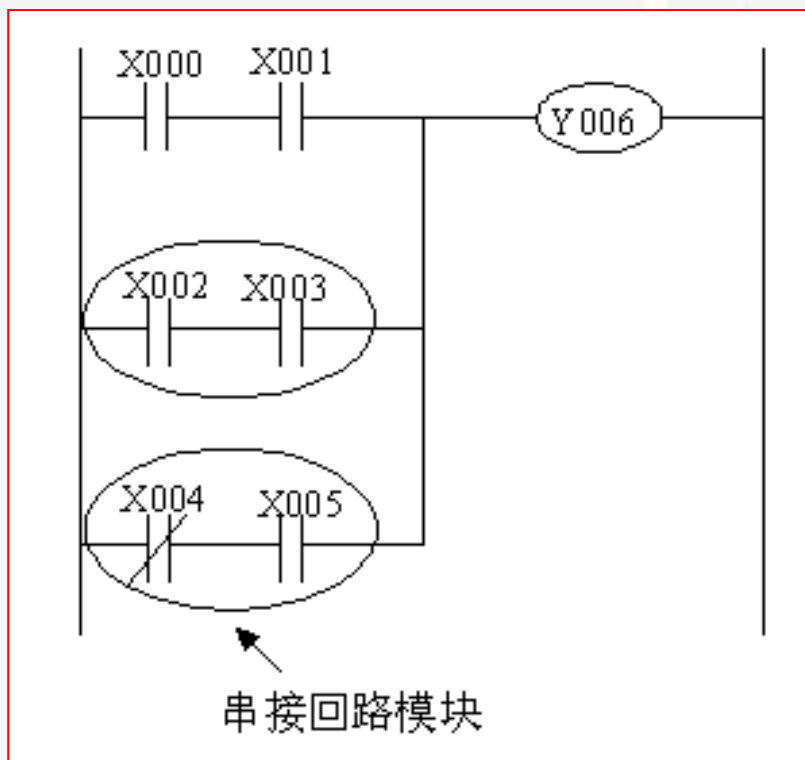
# 串联回路块并联指令ORB



# 串联回路块并联指令ORB

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
ORB	回路块或	串联回路块	1



### ➤ 串联回路块:

- ✓ 由2个以上的触点**串联连接**的回路。
- ✓ 将串联回路块**并联**连接时，**分支开始**用**LD, LDI**指令，**分支结束**用**ORB**指令。

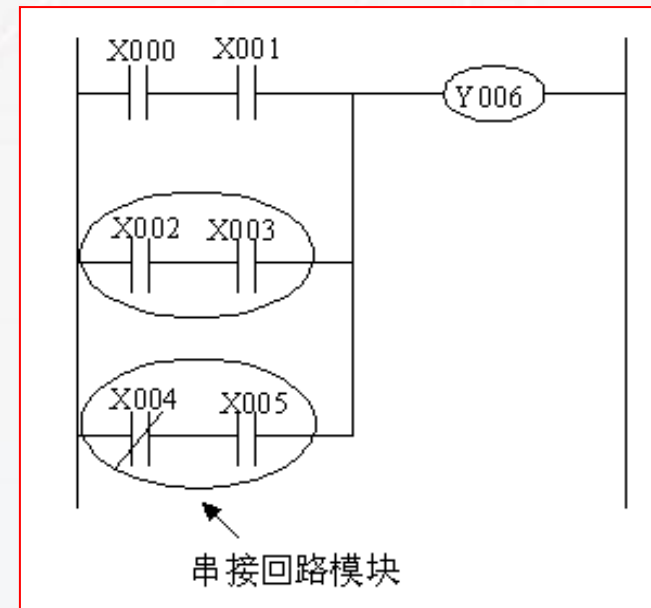
## 2. 指令功能及说明

➤ 分支开始用**LD,LDI**指令，分支结束用**ORB**指令。

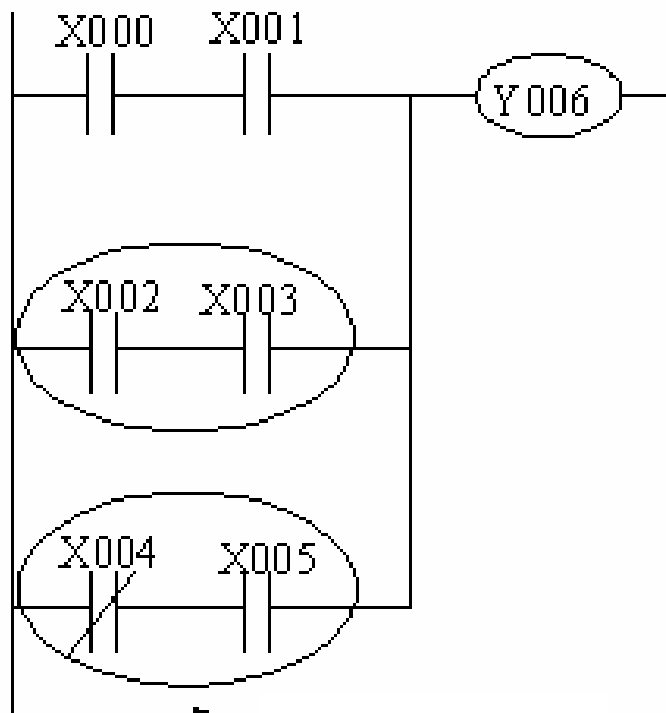
➤ **ORB**指令不带软元件编号的**独立**指令。

➤ 多个串联回路时，如对**每个回路块使用ORB指令**，则串联回路没有限制。

➤ **ORB**指令也可**成批使用**，但是由于**LD,LDI指令的重复次数限制在8次以下**，因此编程时必须注意。



# 【例】 ORB指令的梯形图和指令程序



串接回路模块

ORB指令

正确的程序

0 LD X000

1 AND X001

2 LD X002

3 AND X003

4 ORB; ORB分开使用

5 LDI X004

6 AND X005

7 ORB

8 OUT Y006

不佳的程序

0 LD X000

1 AND X001

2 LD X002

3 AND X003

4 LDI X004

5 AND X005

6 ORB; ORB成批使用

7 ORB

8 OUT Y006





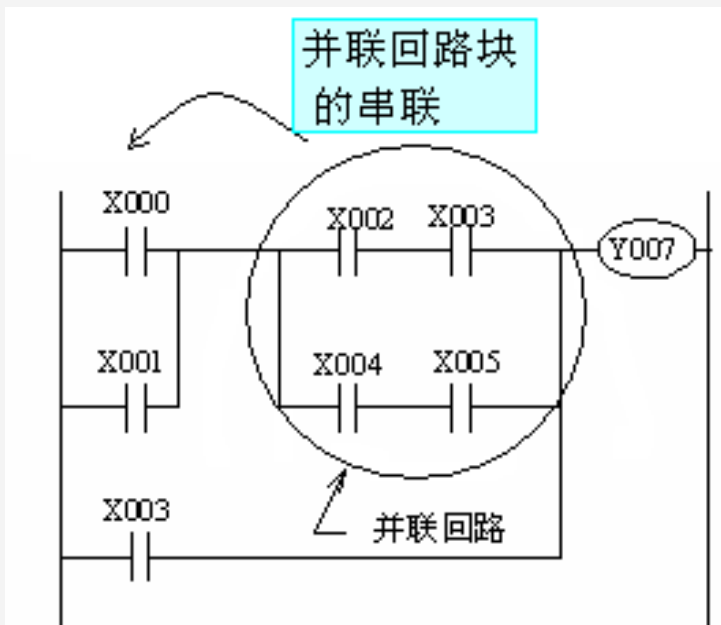
武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 并联回路块串联指令ANB

# 并联回路块串联指令ANB

## 1. 指令定义及应用对象

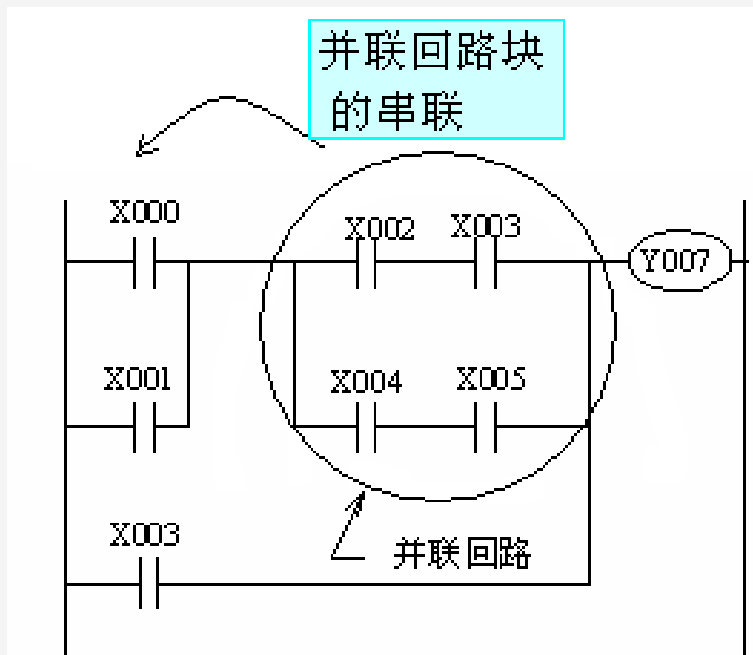
指令符	名称	指令对象	程序步
ANB	回路块与	并联回路块	1



### ➤ 并联回路块：

- ✓ 由2个以上的触点并联连接的回路。当并联回路块与前面的回路串联连接时，使用ANB指令。
- ✓ 分支起点用LD或LDI指令，并联回路块结束用ANB指令与前面的回路串联连接。

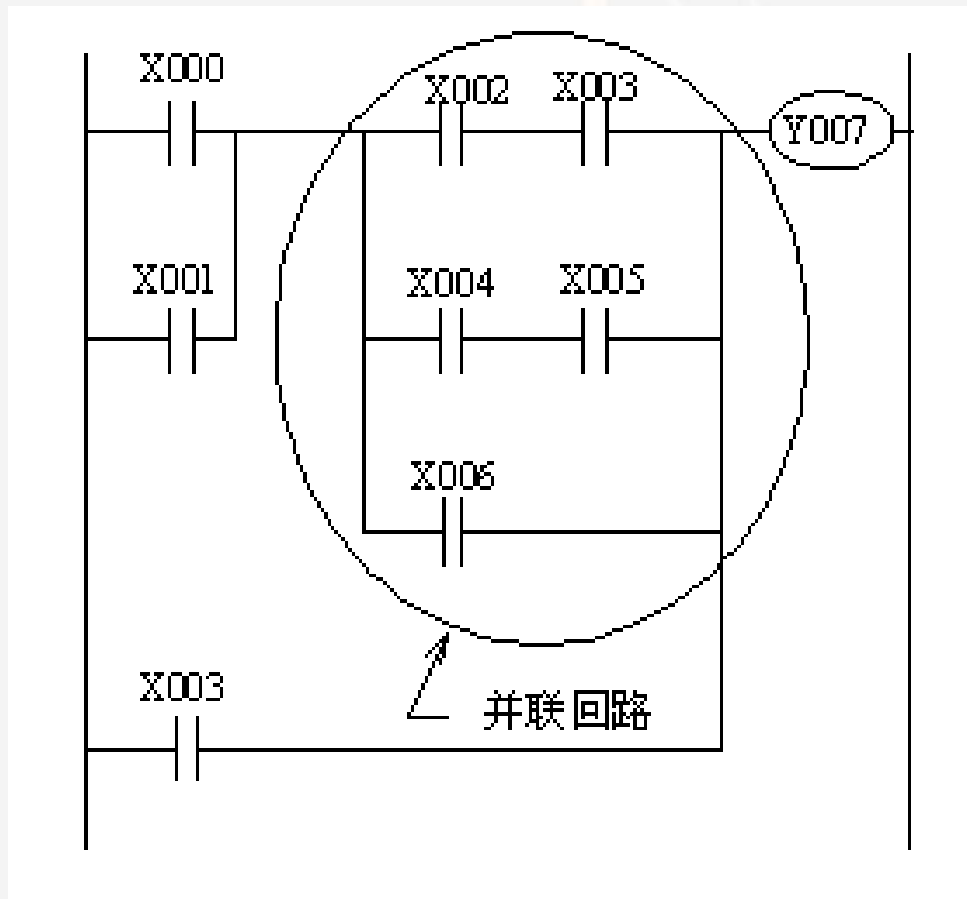
## 2. 指令功能及说明



- 当并联回路块与前面的回路**串联连接**时，使用**ANB指令**。
- **分支起点用LD或LDI指令**，并联回路块结束用**ANB指令**。
- **多个并联回路块按顺序和前面的回路串联时**，ANB指令的使用**次数没有限制**。
- **成批**使用ANB指令，LD,LDI指令的使用次数限制在8次以下。



# 【例】ANB指令的梯形图和指令程序



ANB指令

0 LD X000

1 OR X001

2 LD X002

3 AND X003

4 LD X004

5 AND X005

6 ORB

7 OR X006

8 **ANB**

9 OR X003

10 OUT Y007

分支起点

并联回路块结束

与前面的回路串联

**?去掉X003和X005, 指令**



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---

写出梯形图对应指令代码。







武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 边沿检出指令

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
LDP	取脉冲上升沿	X,Y,M,S,T,C	2
LDF	取脉冲下降沿	X,Y,M,S,T,C	2
ANDP	与脉冲上升沿	X,Y,M,S,T,C	2
ANDF	与脉冲下降沿	X,Y,M,S,T,C	2
ORP	或脉冲上升沿	X,Y,M,S,T,C	2
ORF	或脉冲下降沿	X,Y,M,S,T,C	2

## 2. 指令功能及说明

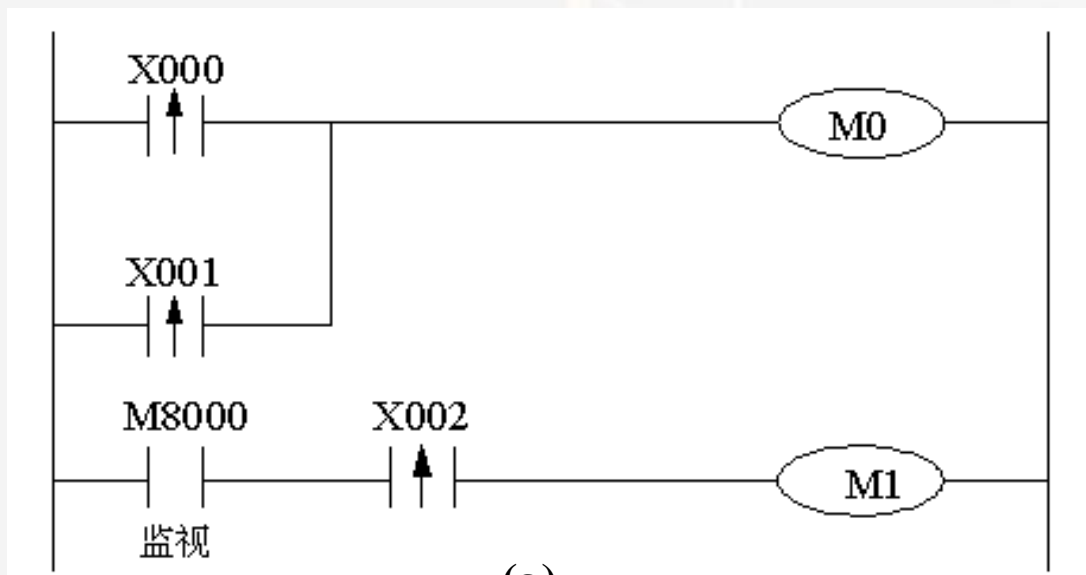
- LDP、ANDP、ORP指令是进行上升沿检出的**触点指令**，**仅在**指定位软元件的**上升沿**时（OFF→ON变化时）**接通一个扫描周期**。
- LDF、ANDF、ORF指令是进行下降沿检出的**触点指令**，**仅在**指定位软元件的**下降沿**时（ON→OFF变化时）**接通一个扫描周期**。
- 利用**上升沿检出和下降沿检出**这一特性，可以利用**同一信号进行状态转移**。



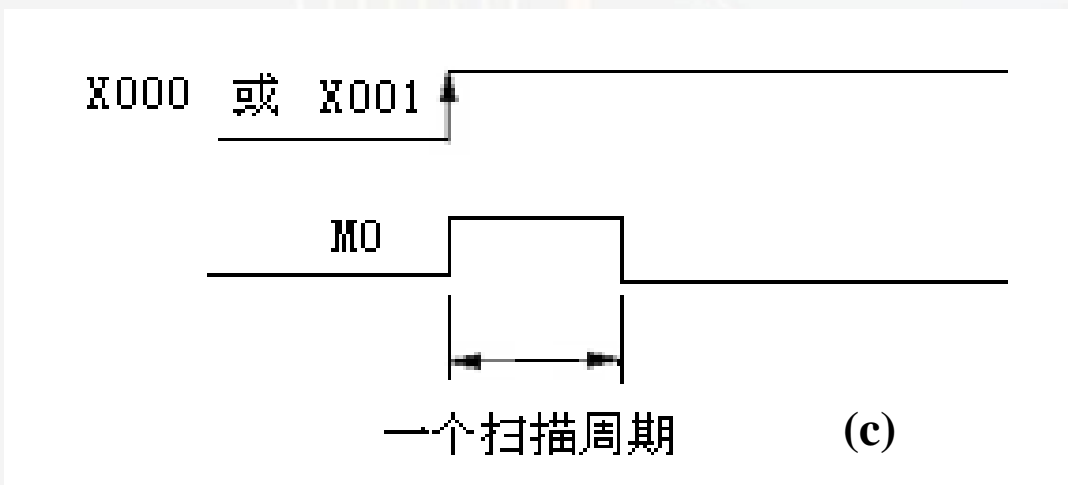
# 【例】 LDP,LDF,ANDP,ANDE,ORP,PRF指令的梯形图和指令程序



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



(a)  
梯形图



时序图

## 指令程序

0 LDP X000

2 ORP X001

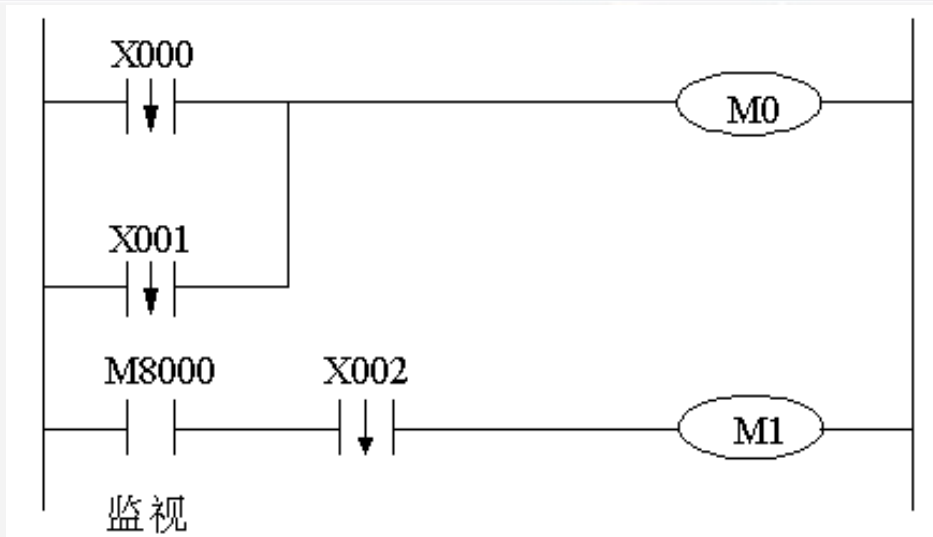
4 OUT M0

5 LD M8000

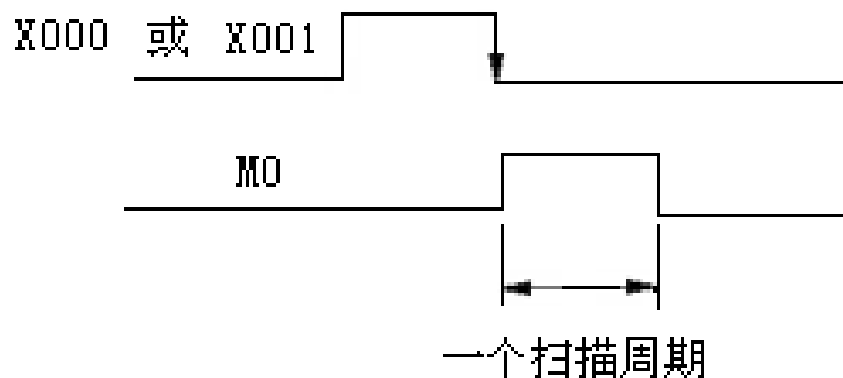
6 ANDP X002

8 OUT M1

# LDF, ANDF, ORF指令



梯形图



时序图

## 指令程序

0 LDF X000

2 ORF X001

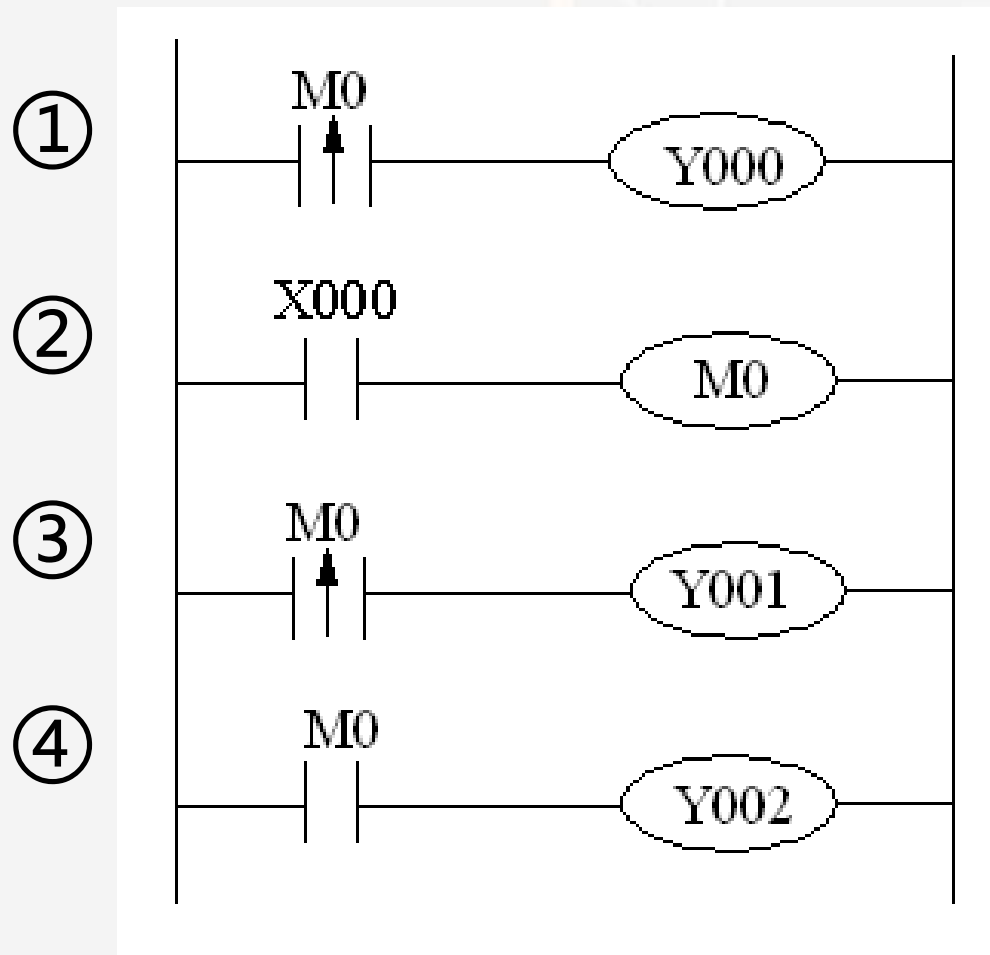
4 OUT M0

5 LD M8000

6 ANDF X002

8 OUT M1

# 【例】LDP指令应用的梯形图

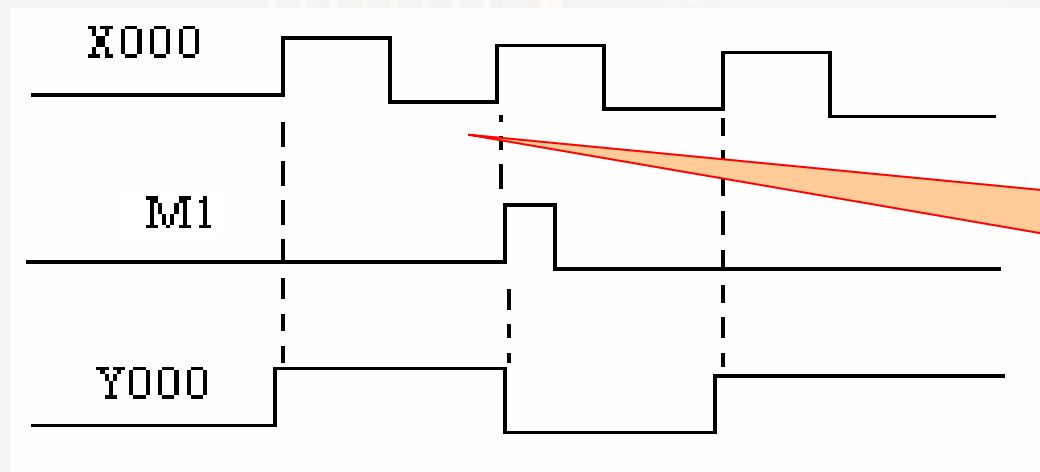
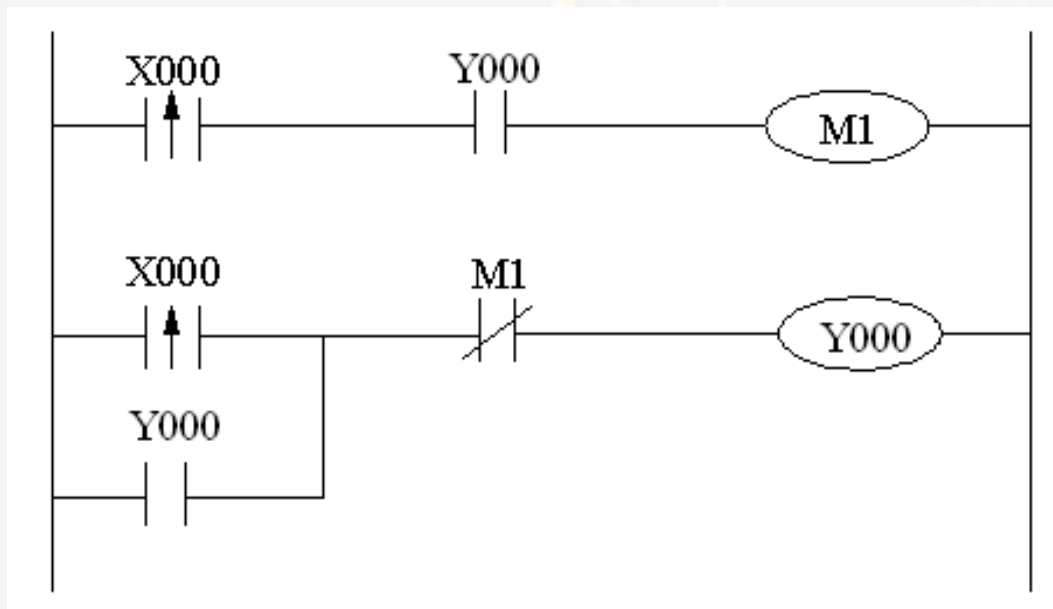


➤ 当X000驱动M0后，① ③  
执行M0的**上升沿检出**功能

➤ 而④为LD指令，因此在  
**M0接通**时，Y002接通



# 【例】双稳态电路程序设计。



0 LDP X000

2 AND Y000

3 OUT M1

4 LDP X000

6 OR Y000

7 ANI M1

8 OUT Y000

注意X000的  
第1和 第2次  
周期



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 脉冲指令PLS/PLF

# 脉冲指令PLS, PLF

## 1. 指令定义及应用对象

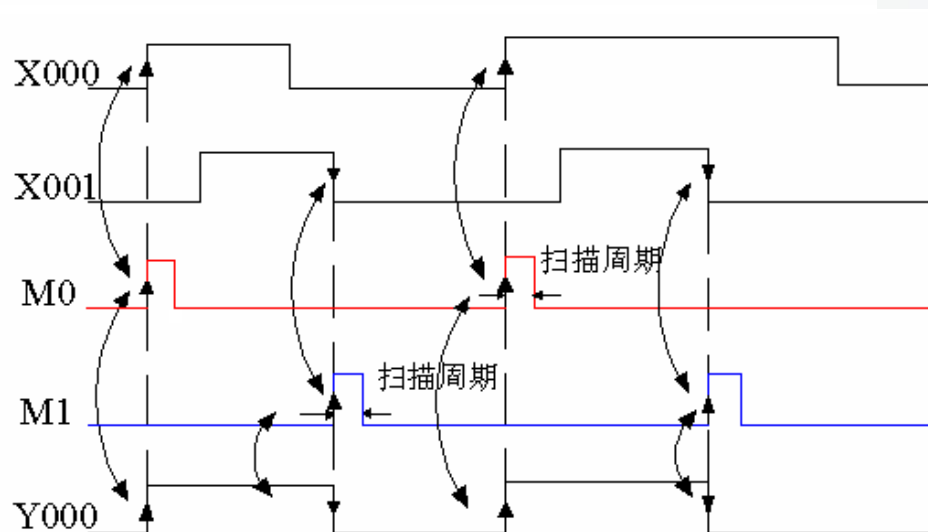
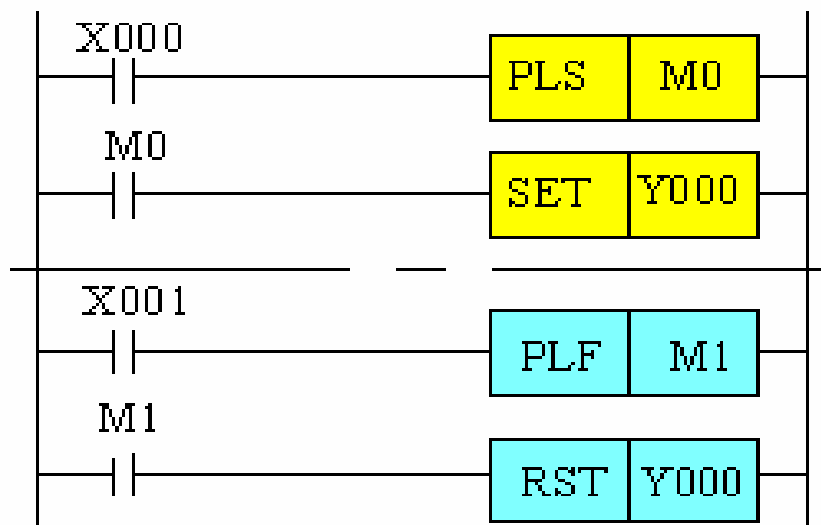
指令符	名称	指令对象	程序步
PLS	上升沿脉冲	除特殊的M以外的M、Y	1
PLF	下降沿脉冲	除特殊的M以外的M、Y	1



## 2. 指令功能及说明

- PLS指令，仅在**驱动输入为ON**后的一个扫描周期内，软元件Y，M动作。
- PLF指令，仅在**驱动输入为OFF**后的一个扫描周期内，软元件Y，M动作。

# 【例】X000、X001作为脉冲指令的触发信号



0 LD X000

1 PLS M 0

2 LD M 0

3 SET Y000

4 LD X001

5 PLF M 1

6 LD M 1

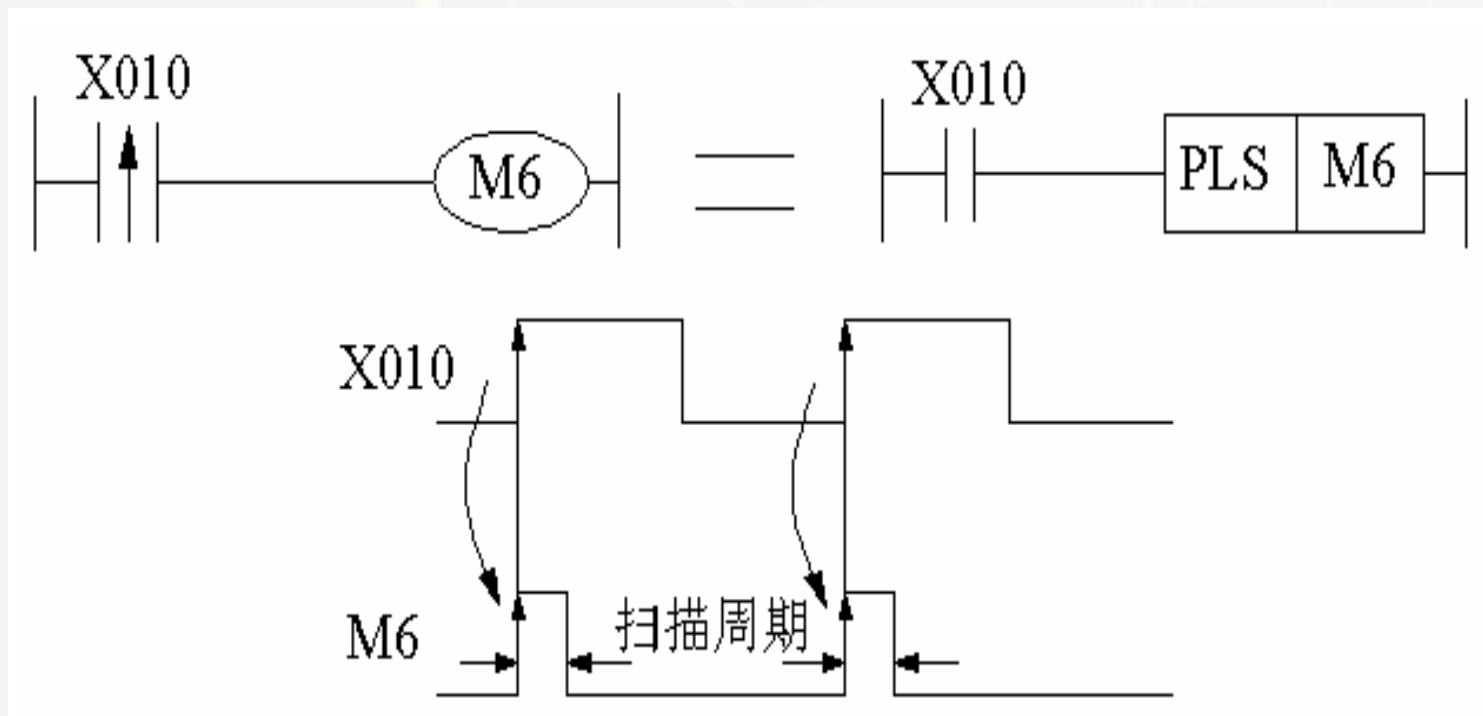
7 RST Y000

# 边沿检出指令与脉冲指令的比较

## ➤ 相同动作的指令程序

LDP指令

PLS脉冲指令



➤ **OUT指令与脉冲指令**两种情况都在X010由**OFF→ON**变化时，**M6接通一个扫描周期**





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 置位与复位指令SET/RST

# 置位与复位指令SET, RST

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
SET	置位	Y, M, S	Y,M : 1 S,特殊M : 2
RST	复位	Y, M, S, T, C, D, V, Z	T,C : 2 D,V, Z,特殊D: 3

□ 注：用M1563-M3071时，程序步加1

## 2. 指令功能及说明

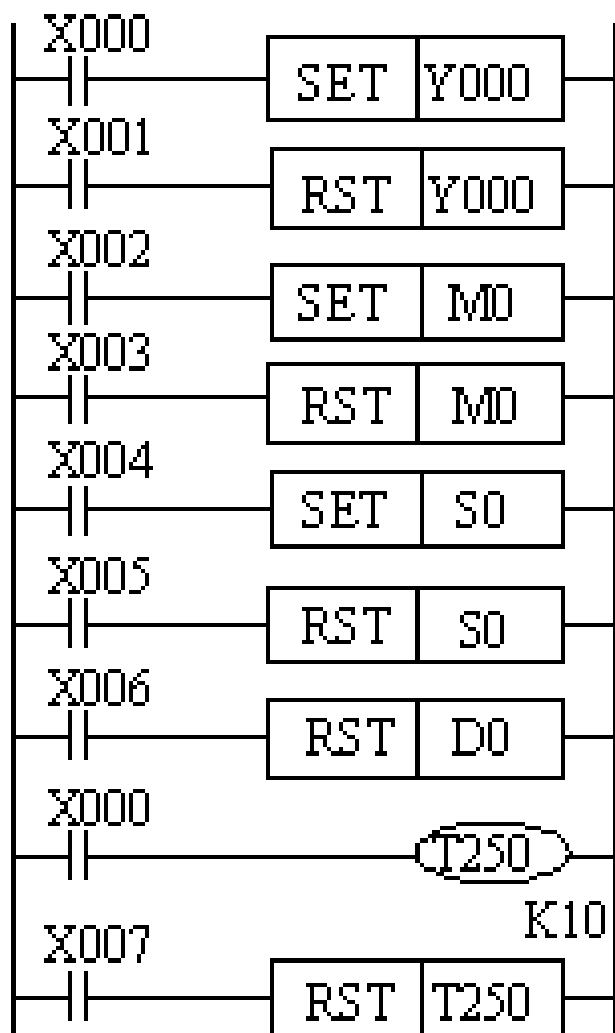
- 同一软元件，SET、RST可**多次使用**，顺序也可随意，但**最后执行者有效**。
- 数据寄存器（D）、变址寄存器（V）、（Z）的内容**清零**时，可用**RST指令**。
- **累积定时器T246~T255的当前值复位以及触点复位可用RST指令**。



# 【例】置位与复位指令梯形图与指令程序。



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X000

1 SET Y000

2 LD X001

3 RST Y000

4 LD X002

5 SET M0

6 LD X003

7 RST M0

8 LD X004

9 SET S0

11 LD X005

12 RST S0

14 LD X006

15 RST D0

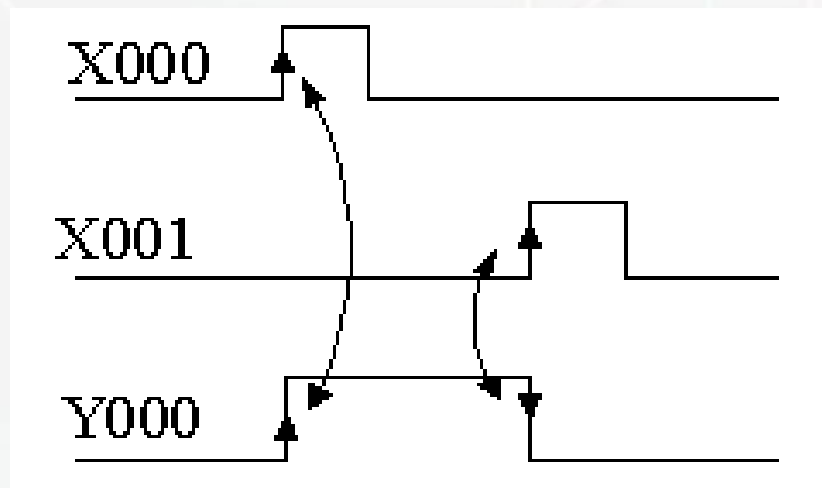
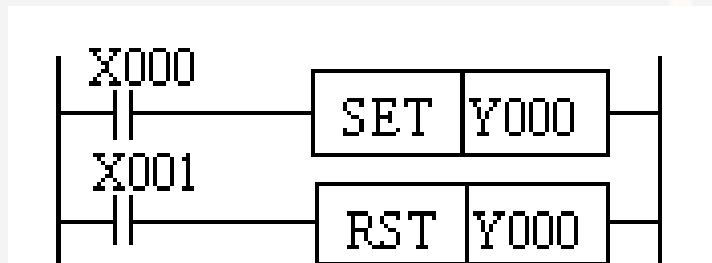
18 LD X000

19 OUT T250  
K10

22 LD X007

23 RST T250

## 例程中（SET和RST）：



- X000一旦接通后，即使它再断开，Y000仍然继续动作。
- X001一旦接通时，即使它再断开，Y000仍然保持不被驱动。
- 对于M，S也是一样的。



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 反转指令INV



# 反转指令INV

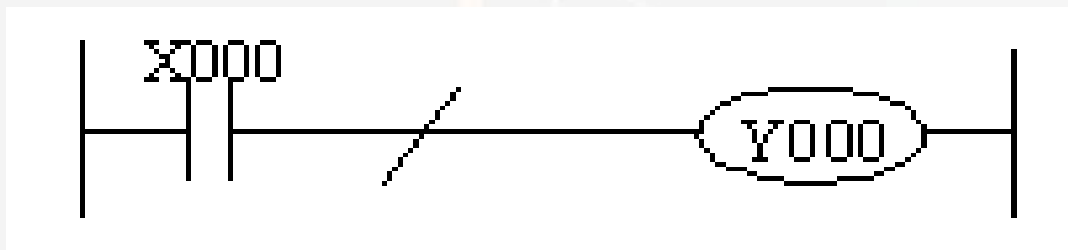
## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
INV	反转	—	1

## 2. 指令功能及说明

- INV指令是将INV指令执行之前的运算结果**反转**的指令。**不需要指定软元件号**。
- 在能输入AND或ANI, ANDP, ANDF指令步的相同位置处，可编写INV指令。
- INV指令**不能**象指令LD, LDI, LDP, LDF那样**与母线连接**，也**不能**象指令OR, ORI, ORP, ORF指令那样**单独使用**。

# 【例】 INV指令的梯形图与指令程序

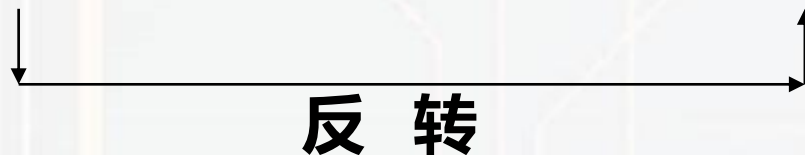


0 LD X000

1 INV

2 OUT Y000

执行INV指令前的运算结果	执行INV指令后的运算结果
OFF	ON
ON	OFF



如果X000断开，则Y000为ON，如果X000接通，则Y000断开。





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

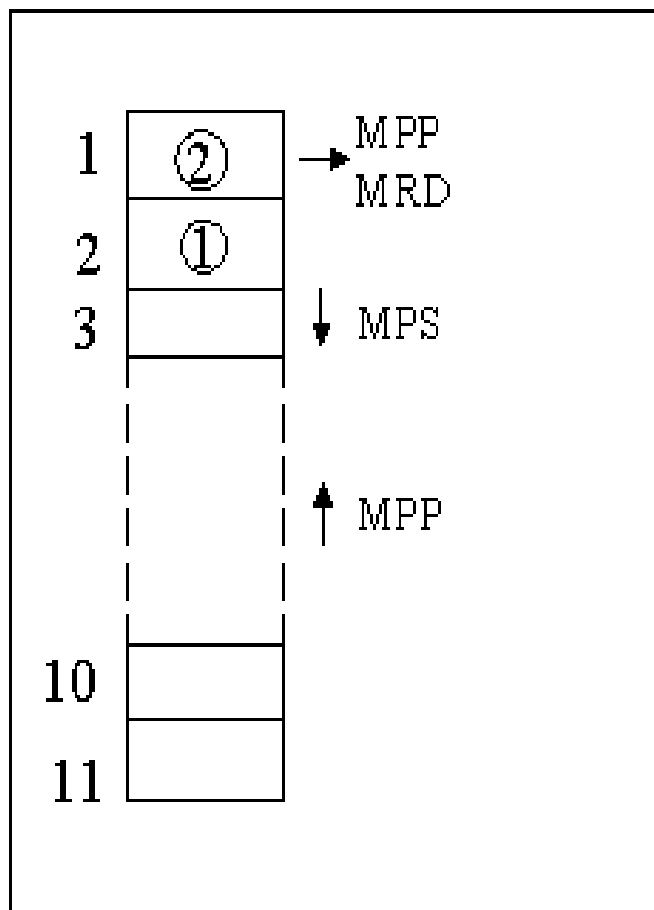
# 栈操作指令



# 栈操作指令MPS, MRD, MPP

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
MPS	进栈		1
MRD	读栈		1
MPP	出栈		1



➤ 有11个记忆运算中间结果的**存储器**(称为**栈**)。

➤ 使用**MPS指令**，即将**此时运算结果**送入栈的第一段存储。再用**MPS指令**，又将该时刻运算结果送入栈的第一段存储，先前送入存储的数据依次移到栈的下一段。

➤ 使用**MPP指令**，各数据按顺序向上移动，将**最上端的数据**读出，同时该数据就**从栈中消失**。

➤ **MRD指令**读出**最上端**所存的最新数据，栈内的数据不发生移动。

➤ 不带软元件编号的**独立指令**。

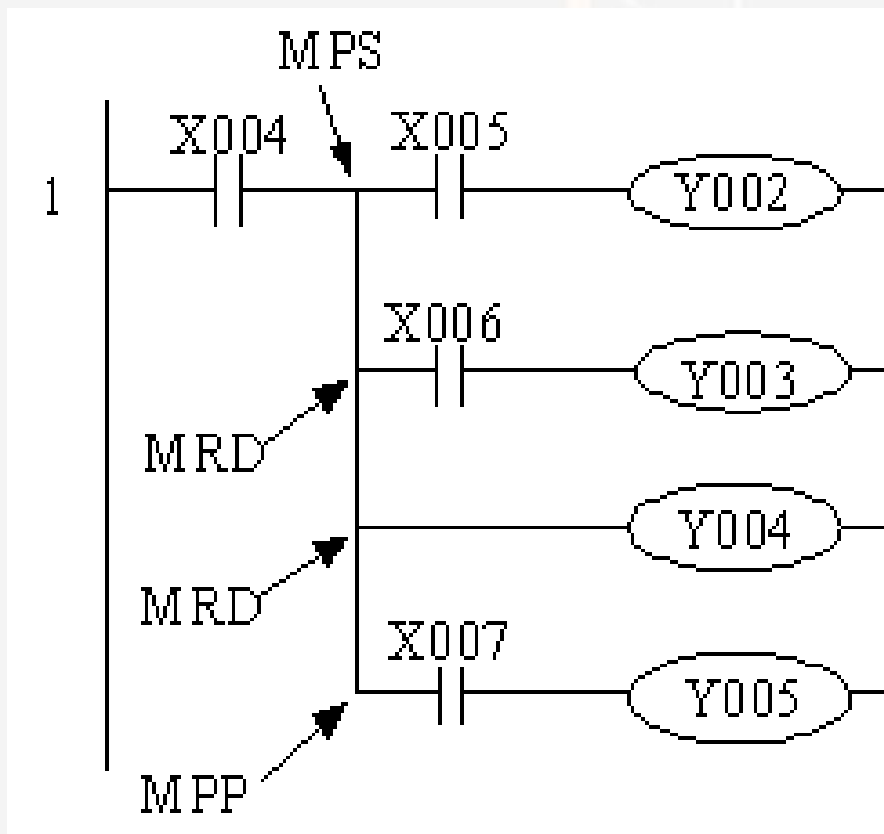
## 2. 指令功能及说明

- MPS指令可重复使用，MPS指令与MPP指令的数量差额少于11，但最终二者的**指令数要一样**。
- MRD指令可多次编程，但是在打印、图形编程面板的画面显示方面有限制（并联回路24行以下）。
- 最终输出回路以MPP指令替代MRD指令。从而在读出上述存储的同时将它复位。

# 【例】 栈操作指令的梯形图与指令程序



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X004

1 MPS

2 AND X005

3 OUT Y002

4 MRD

5 AND X006

6 OUT Y003

7 MRD

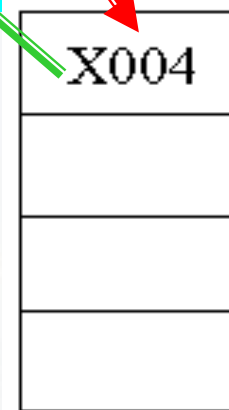
8 OUT Y004

9 MPP

10 AND X007

11 OUT Y005

12 END



## □ 分支多重输出回路编程

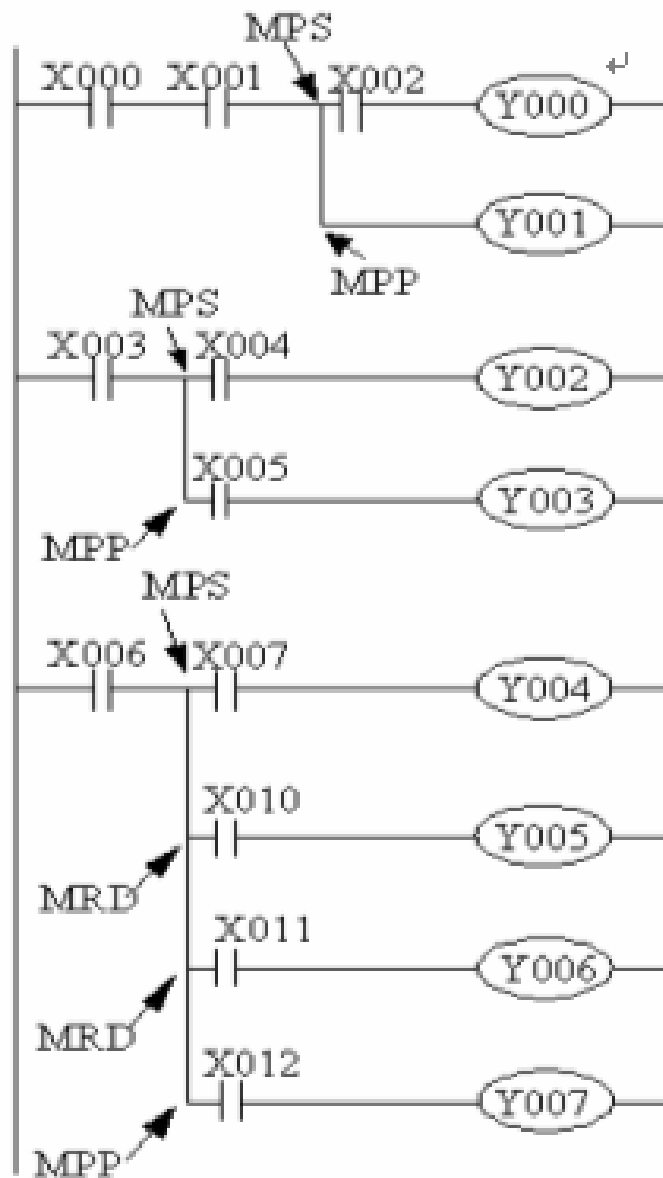
利用MPS指令存储得出的运算中间结果，然后驱动Y002。用MRD指令将该存储读出，再驱动输出Y003。



# 【例】一段堆栈的梯形图与指令程序



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X000

1 AND X001

2 MPS

3 AND X002

4 OUT Y000

5 MPP

6 OUT Y001

7 LD X003

8 MPS

9 AND X004

10 OUT Y002

11 MPP

12 AND X005

13 OUT Y003

14 LD X006

15 MPS

16 AND X007

17 OUT Y004

18 MRD

19 AND X010

20 OUT Y005

21 MRD

22 AND X011

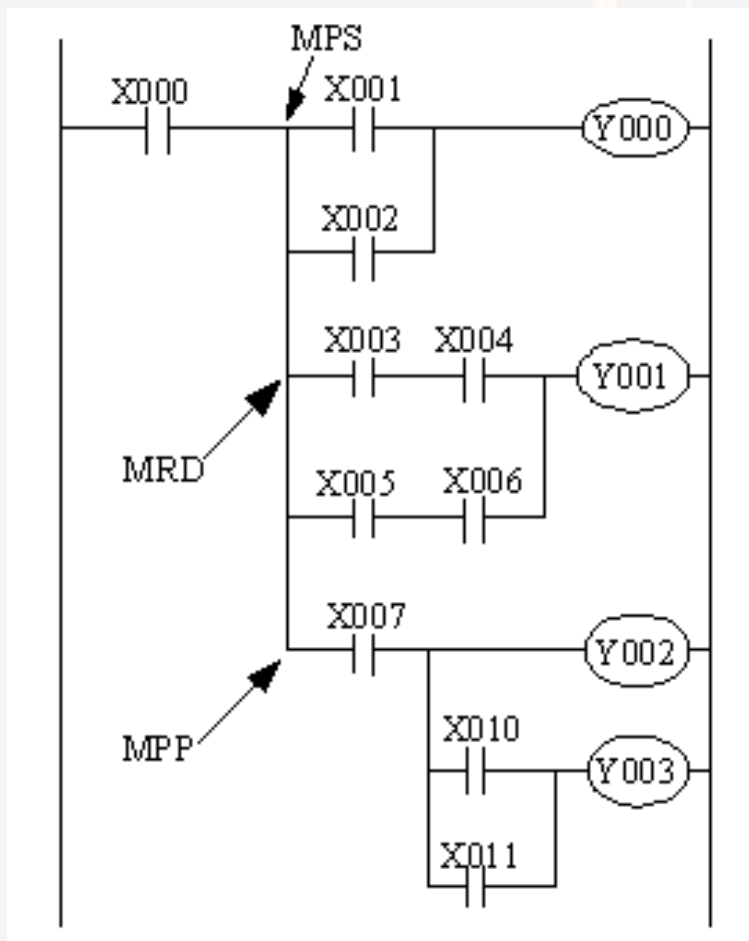
23 OUT Y006

24 MPP

25 AND X012

26 OUT Y007

# 【例】一段堆栈与ANB， ORB指令并用的梯形图与指令程序。



堆栈与ANB， ORB指令并用

0 LD X000

1 MPS

2 LD X001

3 OR X002

4 ANB

5 OUT Y000

6 MRD

7 LD X003

8 AND X004

9 LD X005

10 AND X006

11 ORB

12 ANB

13 OUT Y001

14 MPP

15 AND X007

16 OUT Y002

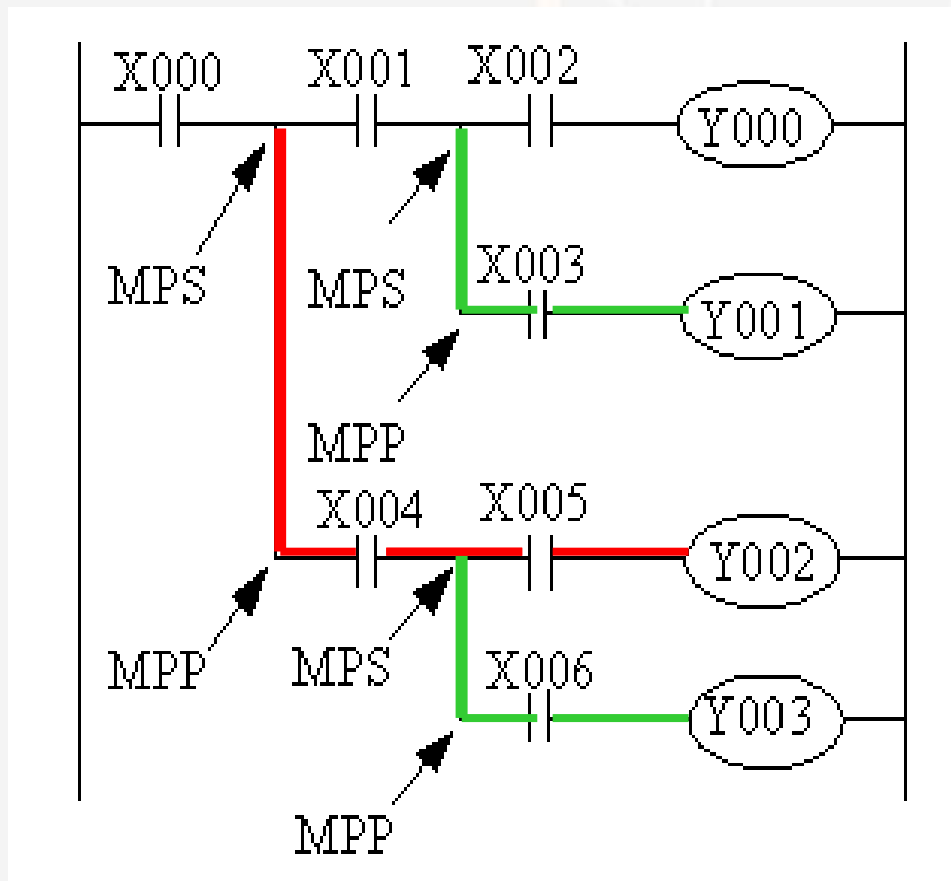
17 LD X010

18 OR X011

19 ANB

20 OUT Y003

# 【例】 两段堆栈的梯形图与指令程序



两段堆栈编程

0 LD X000

1 MPS

2 AND X001

3 MPS

4 AND X002

5 OUT Y000

6 MPP

7 AND X003

8 OUT Y001

9 MPS

10 AND X004

11 MPS

12 AND X005

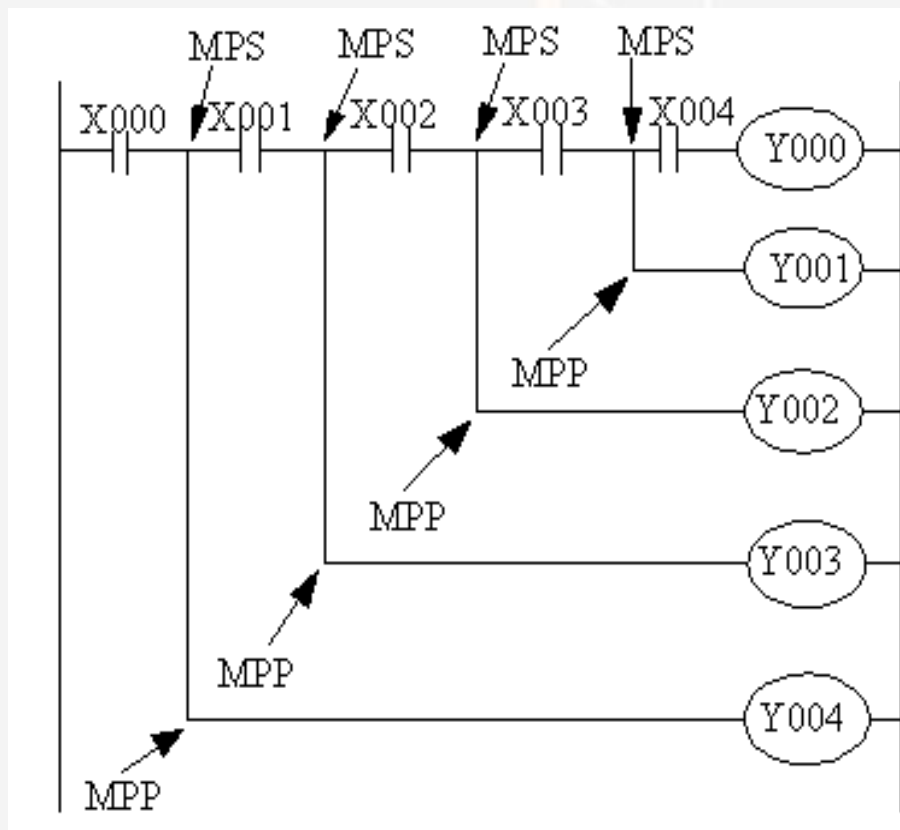
13 OUT Y002

14 MPP

15 AND X006

16 OUT Y003

# 【例】 四段堆栈的梯形图与指令程序



0 LD X000

1 MPS

2 AND X001

3 MPS

4 AND X002

5 MPS

6 AND X003

7 MPS

8 AND X004

9 OUT Y000

10 MPP

11 OUT Y001

12 MPP

13 OUT Y002

14 MPP

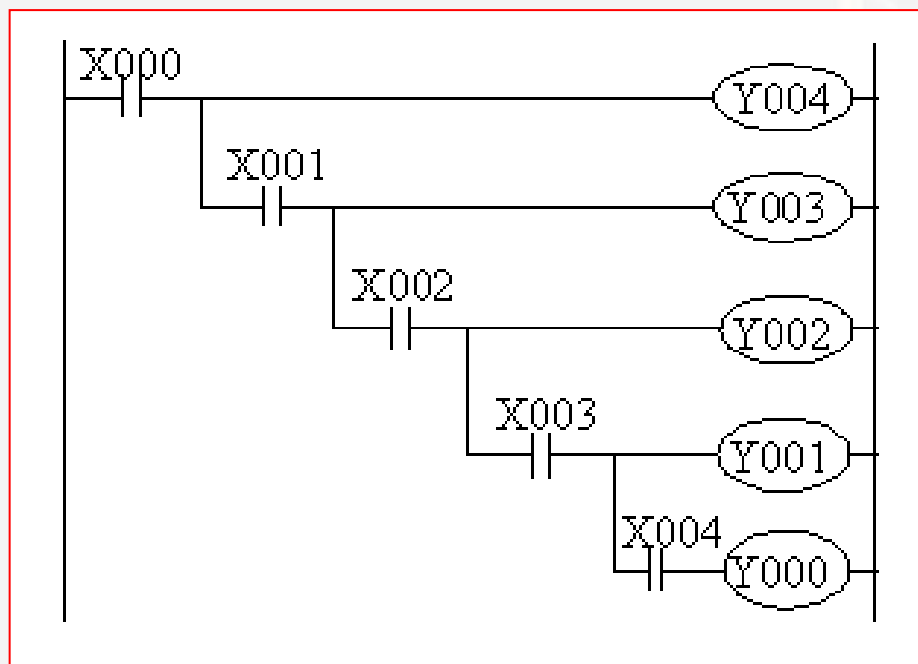
15 OUT Y003

16 MPP

17 OUT Y004

(a) 四段堆栈的编程





(b) 未采用MPS指令的编程

四段堆栈的编程

0 LD X000

1 OUT Y004

2 AND X001

3 OUT Y003

4 AND X002

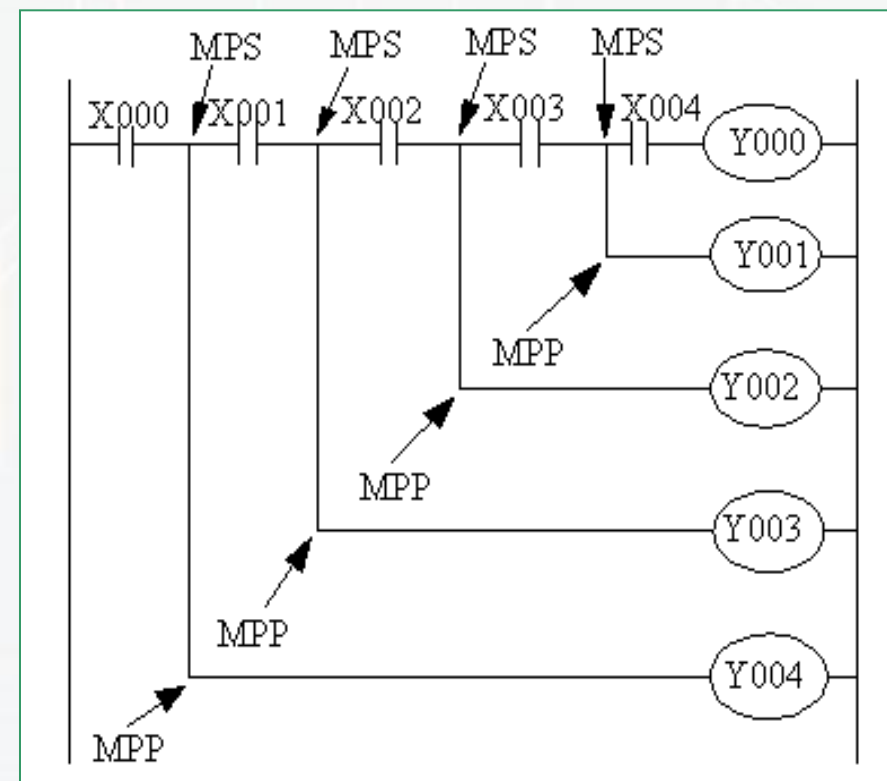
5 OUT Y002

6 AND X003

7 OUT Y001

8 AND X004

9 OUT Y000

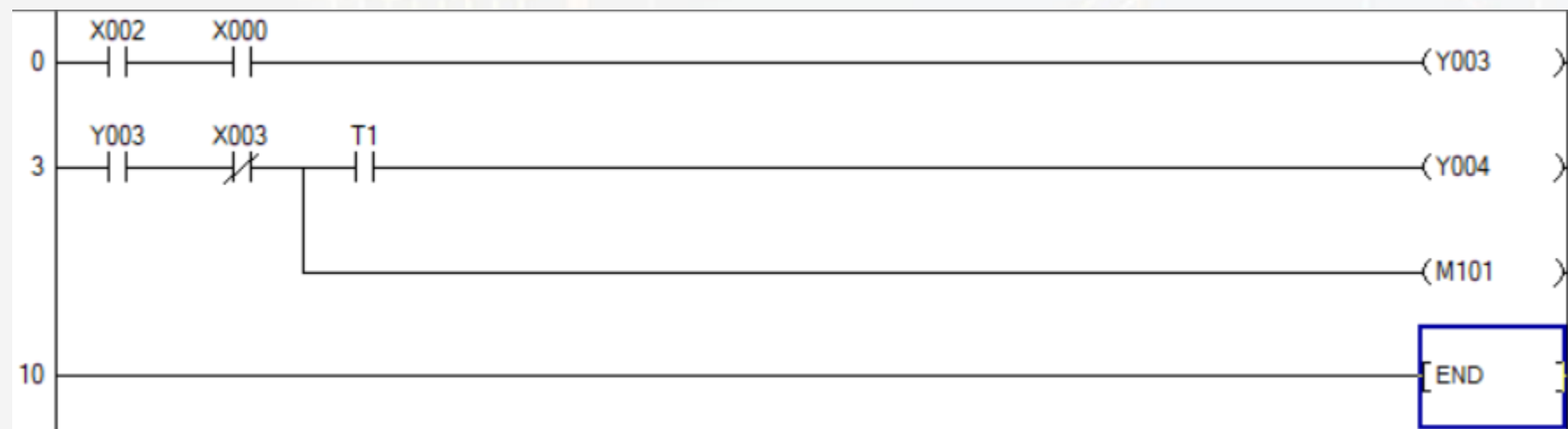
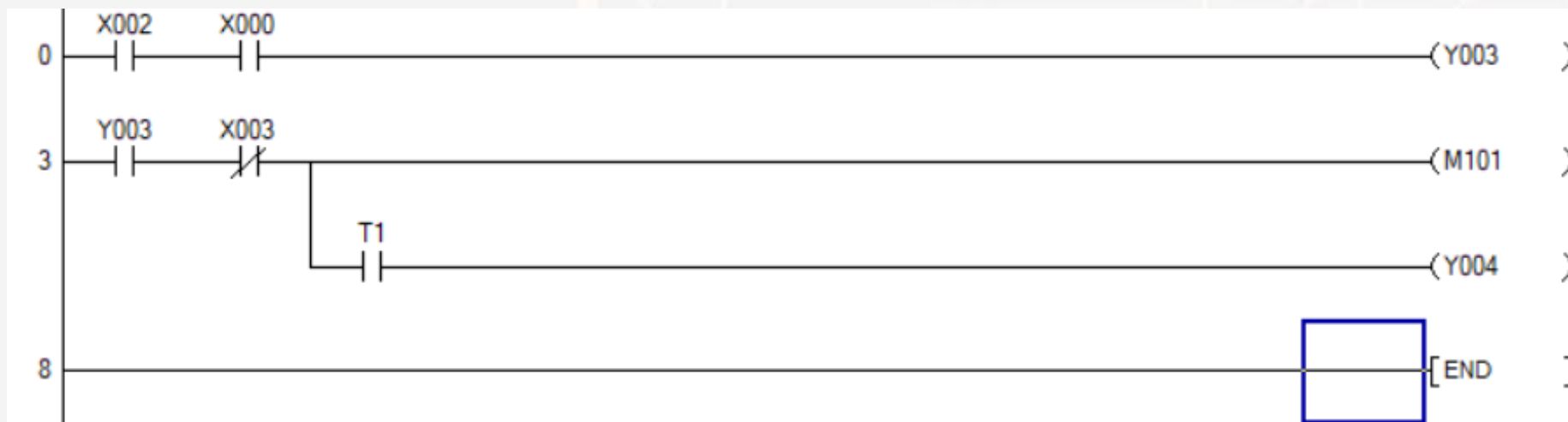




武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---



```
LD      X002
AND     X000
OUT     Y003
LD      Y003
ANI     X003
OUT     M101
AND     T1
OUT     Y004
END
```



```
LD    X002  
AND X000  
OUT Y003  
LD    Y003  
ANI   X003  
MPS  
AND T1  
OUT Y004  
MPP  
OUT M101  
END
```



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 主控触点指令MC, MCR

# 主控触点指令MC, MCR

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
MC	主控（回路块起点）	Y, M (除特殊继电器以外)	3
MCR	主控复位（回路块终点）		2

## 2. 指令功能及说明

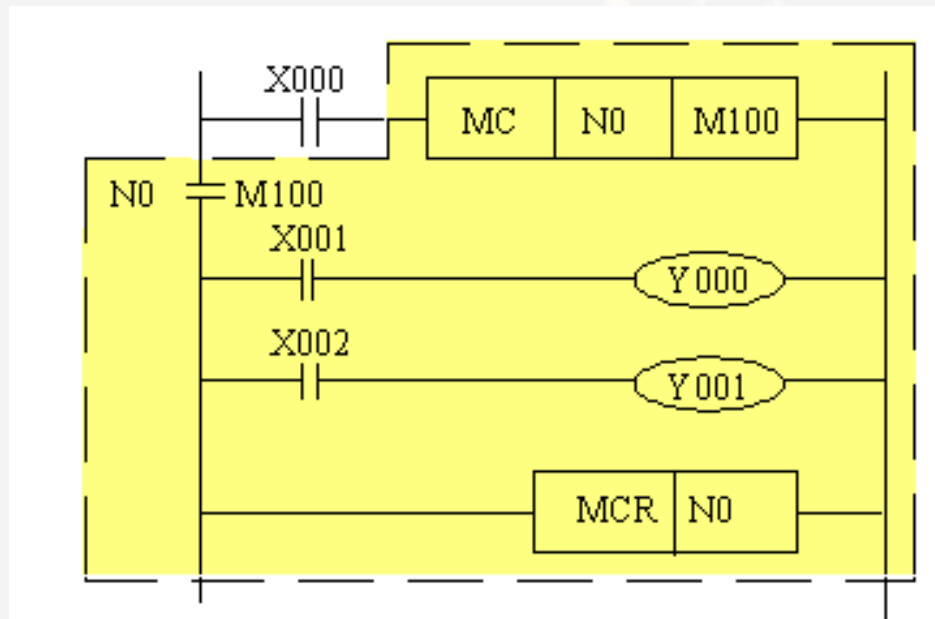
- MC主控指令：**母线转移**，只作用于输出继电器Y和辅助继电器。
- MCR主控复位指令：**母线复位**，主控区结束。
- MC/MCR：用于许多线圈同时受一个或一组触点控制，主控触点在**梯形图中与一般触点垂直**。节省存储单元。
- 执行**MC**指令后，**母线（LD，LDI）向MC触点后移动**，指令**MCR**将其**返回到原母线**。



# 【例】主控触点指令的梯形图与指令程序（无嵌套）



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



0 LD X000

1 MC N 0

M100

4 LD X001

5 OUT Y000

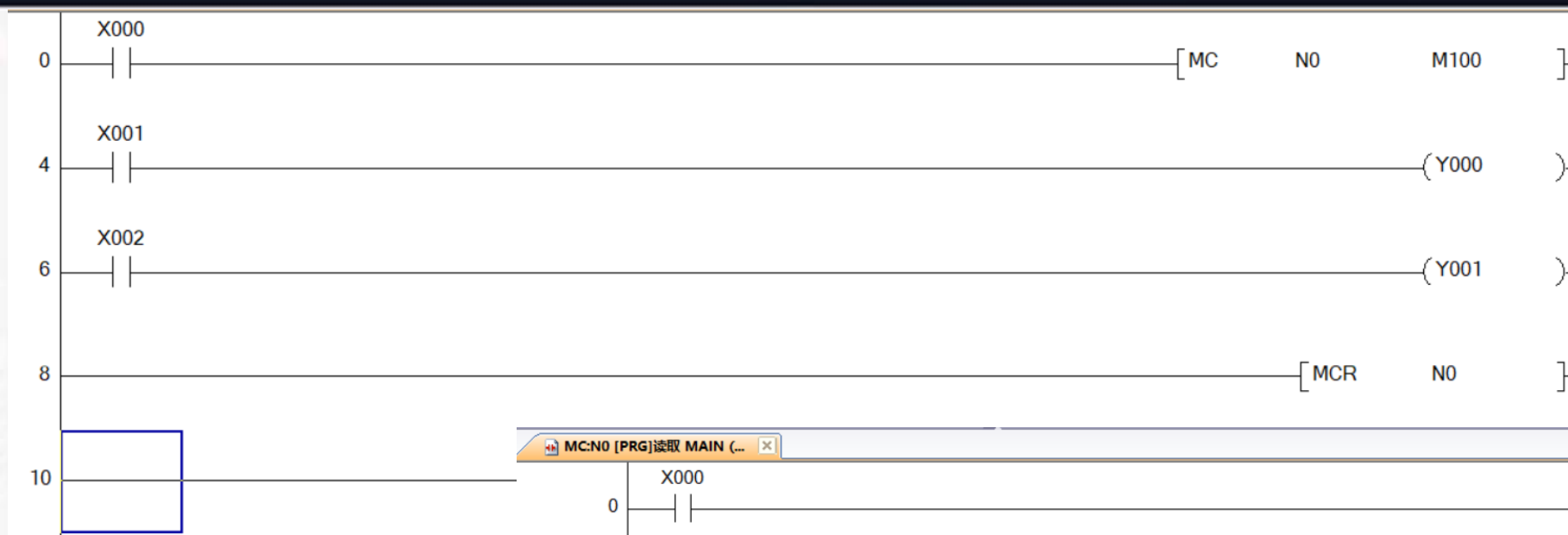
6 LD X002

7 OUT Y001

8 MCR N 0

← 母线返回 (N0为嵌套等级)

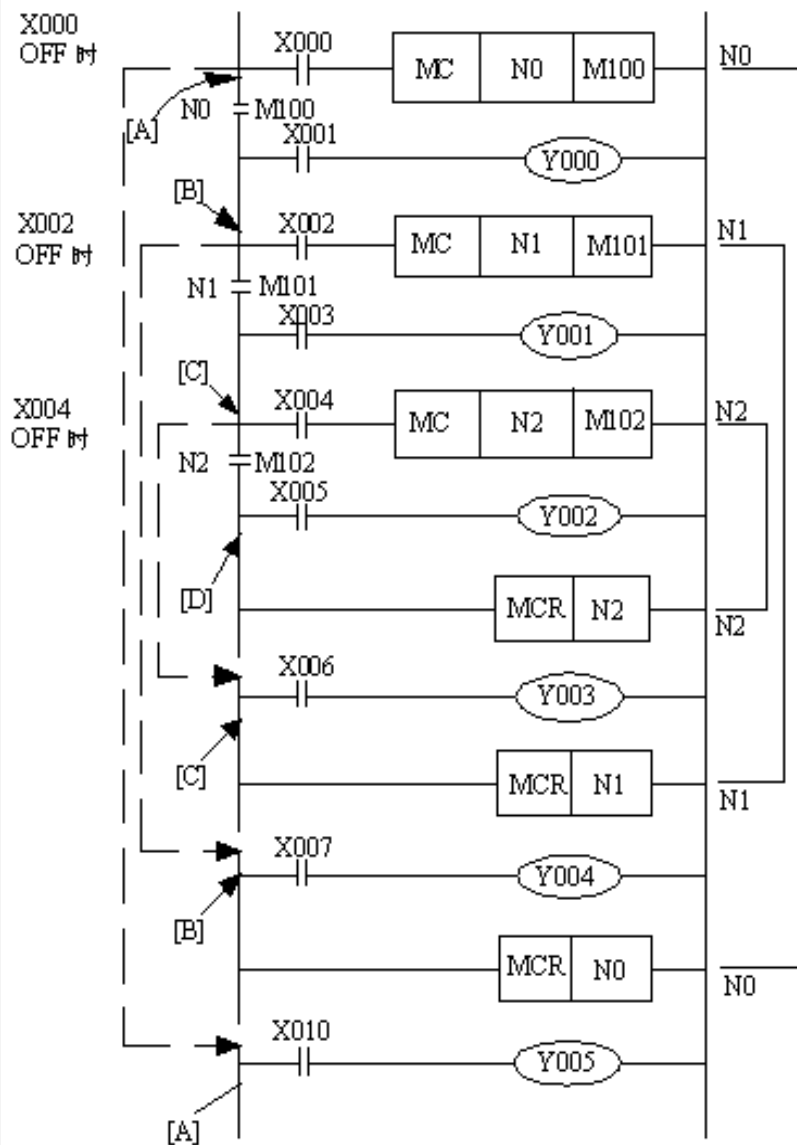
- 当X0接通时，M100置ON，主控段程序执行。
- 当X0断开时，M100置OFF，主控段程序不执行。
- 当M100置OFF时，主控段内由OUT指令输出部分复位，而由SET指令输出部分，保持原状态。



# 【例】 主控触点指令的梯形图与指令程序 (有嵌套)



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



(级N0)

母线B在X000为ON时，呈激活状态

(级N1)

母线C在X000, X002为ON时，呈激活状态

(级N2)

母线D在X000,X002,X004都为ON时，呈激活状态

(级N1)

通过MCR N2，母线返回到C的状态

(级N0)

通过MCR N1，母线返回到B的状态

(初始态)

通过MCR N0，母线返回到初始的A状态

Y005的接通/断开只取决于X010的接通/断开状态，而与X000, X002, X004状态无关。

有嵌套的主控触点指令的编程



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 空操作指令NOP和END



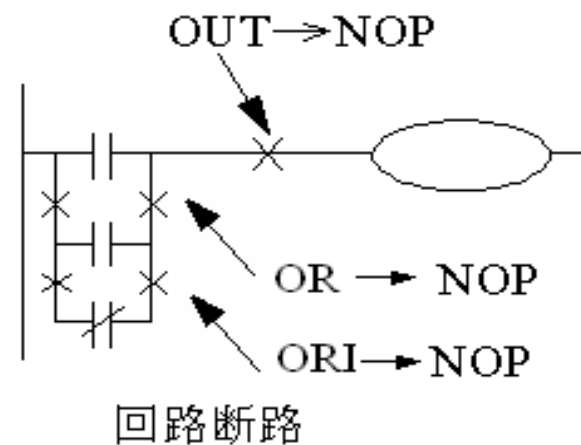
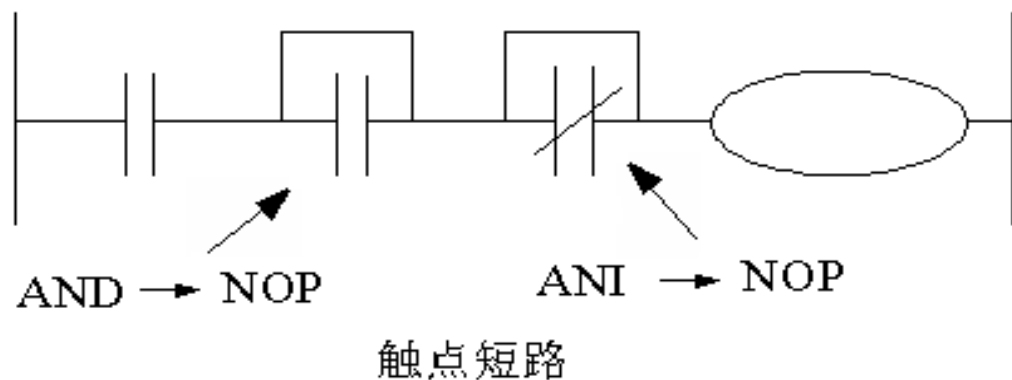
## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
NOP	空操作	—	1

## 2. 指令功能及说明

- 将程序全部消除时，全部指令成为NOP。
- 在普通的指令与指令之间加入NOP指令，则可编程控制器将无视其存在继续工作。
- 在程序中加入NOP指令，则在修改或追加程序时，可以减少步号的变化。
- 若将已写入的指令换成NOP指令，则回路会发生变化

# 【例】空操作指令的应用



用NOP取代AND，就表示删除AND。

# 程序结束指令END

## 1. 指令定义及应用对象

指令符	名称	指令对象	程序步
END	结束程序	—	1

## 2. 指令功能及说明

- 在调试阶段，在各程序段插入END指令，可依次检出各程序段的动作
- RUN开始时的首次执行，从执行END指令开始
- 执行END指令时，也刷新监视定时器（检查扫描周期是否过长的定时器）



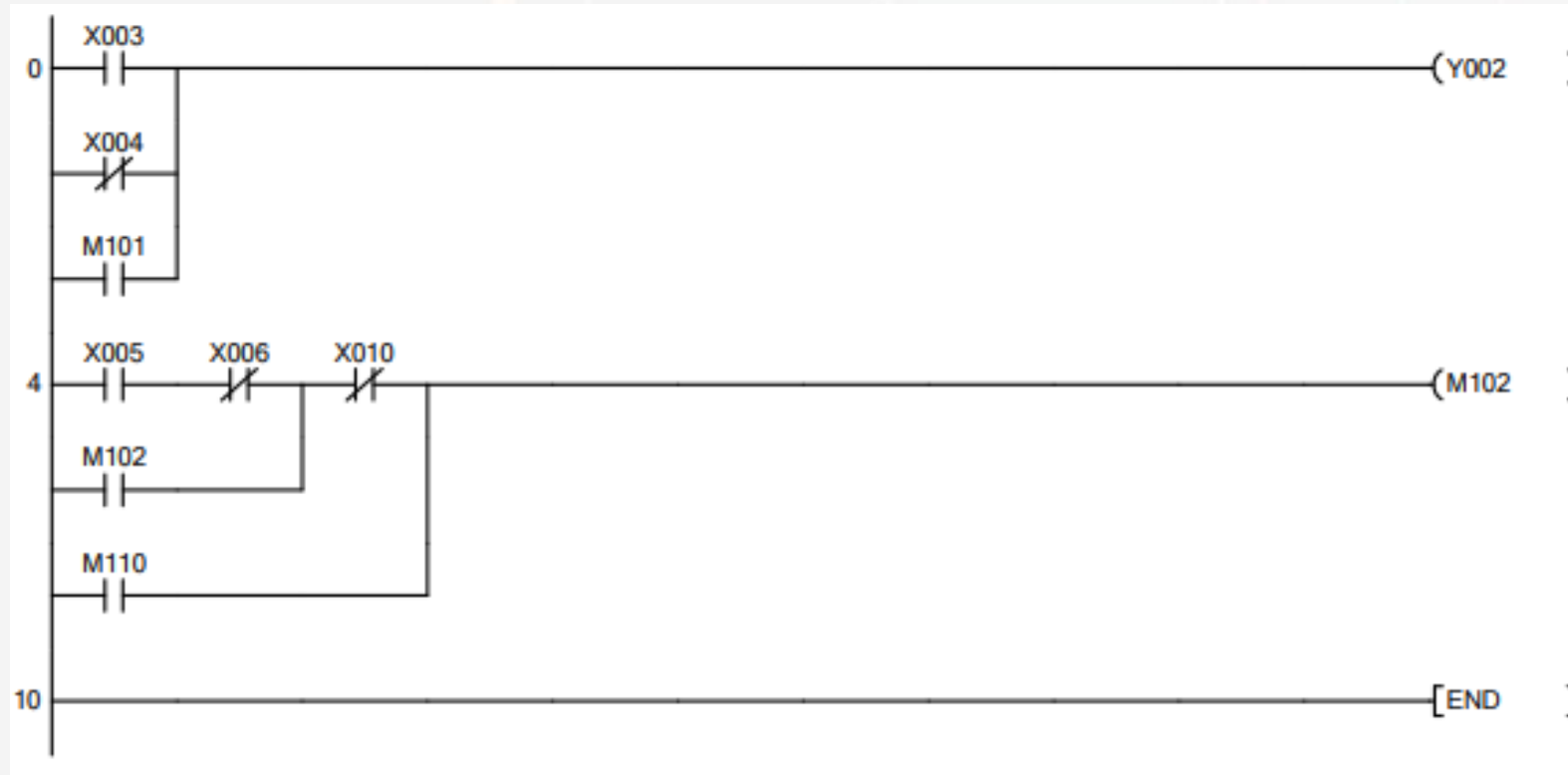


武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---

写出梯形图对应指令代码。







武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

## 3.2 梯形图编写规则





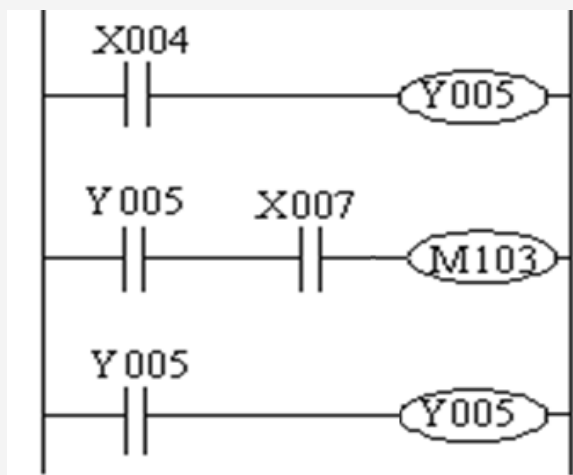
武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 梯形图编程格式

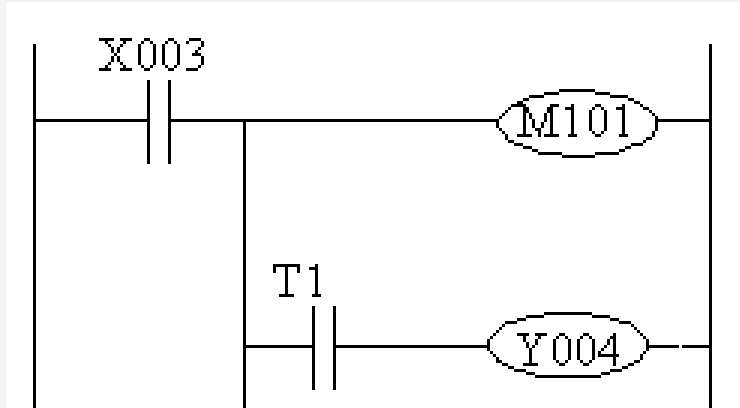


# 梯形图编程格式

- 每个梯形图程序由多个梯级组成，一个输出元素可构成一个梯级，每个梯级可由多个支路组成。
- 每个支路通常可容纳11个编程元素，最右边的元素不能是触点。
- 每个梯级最多允许16条支路。
- 梯形图编程时，只有在在一个梯级编制完后才能继续后面的编程。
- 输出线圈用圆形或椭圆形表示。



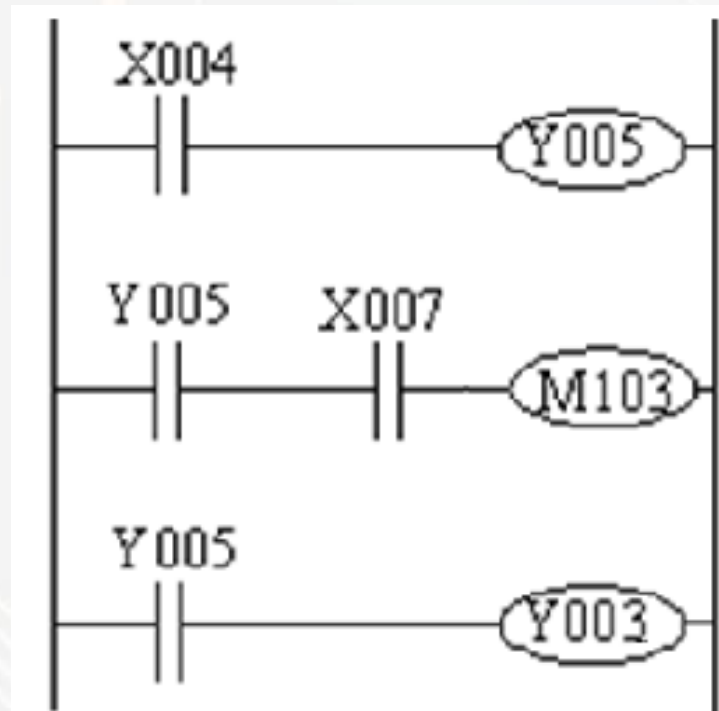
# 梯形图的特点

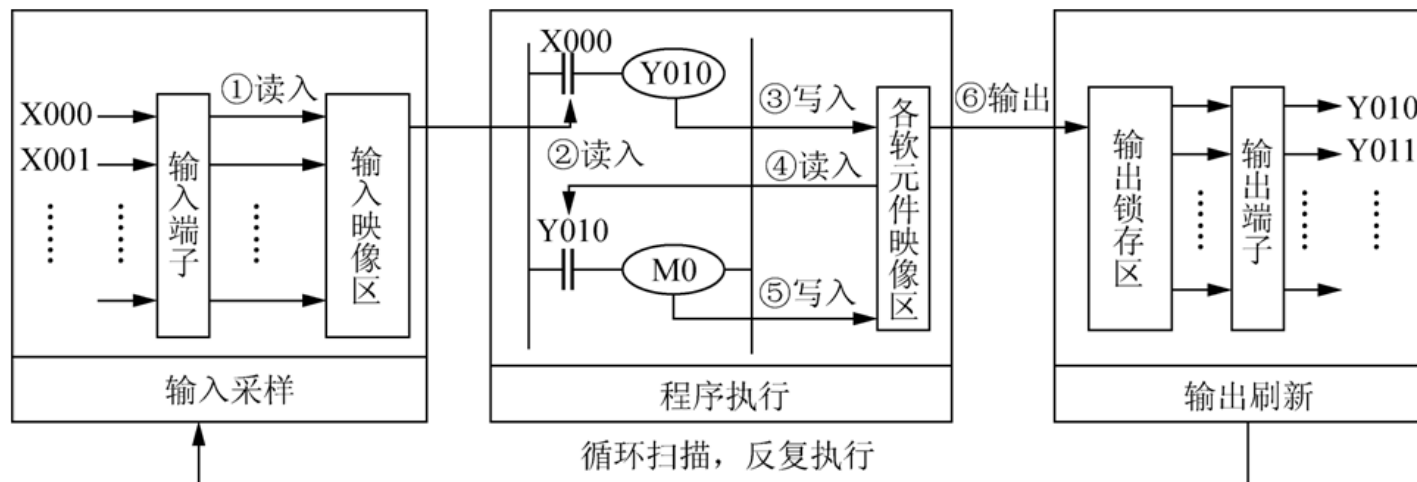


- 梯形图格式中每个继电器和输入接点均为存储器中的一位，相应位为“1”状态，表示继电器线圈通电或常开接点闭合或常闭接点断开。
- 梯形图中流过的电流是“概念”电流。“概念”电流只能从左向右流动。用户程序解算中满足输出执行条件的形象表示方式。



- 梯形图中的继电器触点可在程序中无限次引用，既可常开又可常闭。
- 梯形图中用户逻辑解算结果，可马上为后面用户程序的解算所用。





- 梯形图中**输入接点和输出线圈**不是物理接点和输出线圈，用户程序的解算是根据PLC内**I/O映象区每位**的状态，而不是解算时现场开关的实际状态。
- **输出线圈只对应输出映象区的相应位**，不能用该编程元件直接驱动现场机构，该位的状态必须**通过I/O模块上对应的输出单元**才能驱动现场执行机构。



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 梯形图设计规则

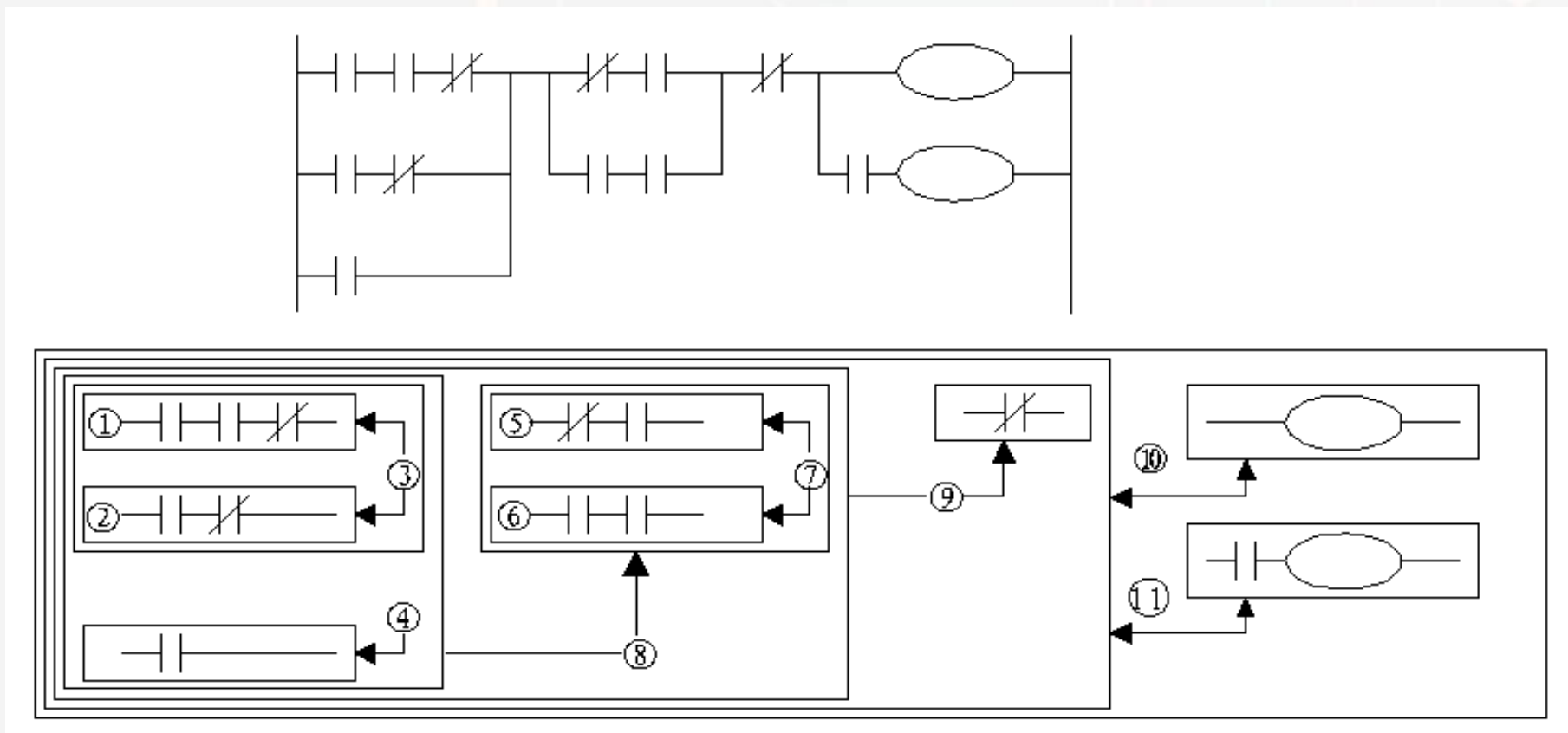
# 梯形图设计规则

- 梯形图按PLC在一个扫描周期内扫描程序的顺序，**从左到右、从上到下的顺序进行绘制**。与右边线圈相连的全部支路组成一个逻辑行。
- 逻辑行**起于左母线，终于右母线**（或终于线圈，或一特殊指令）。
- 不能在线圈与右母线之间接其他元件。



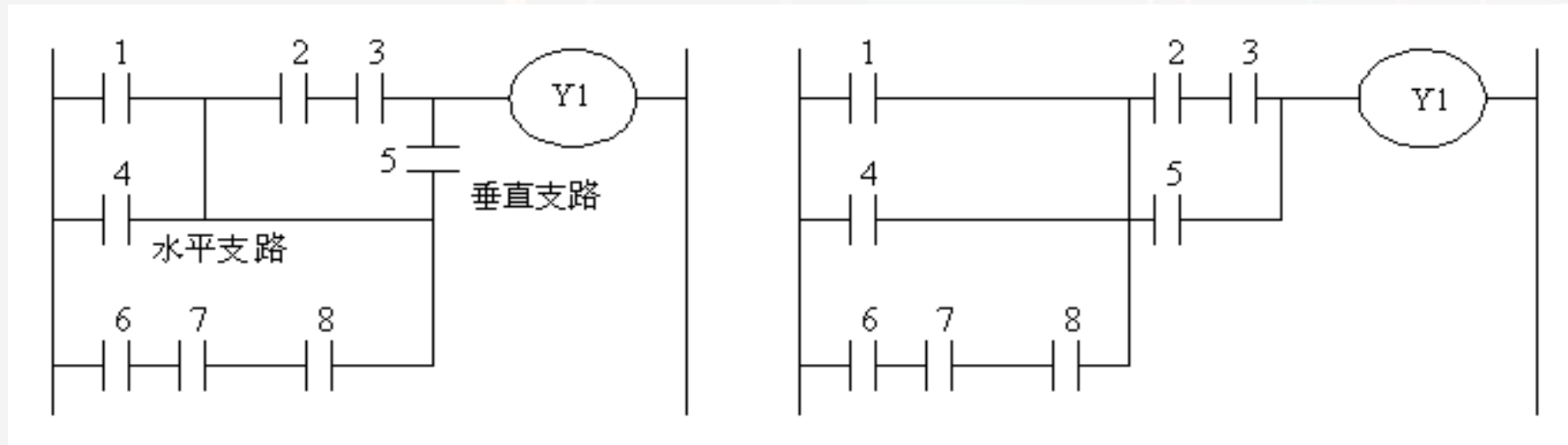
# 梯形图编程规则一

➤ **编程顺序:**一个逻辑行编程顺序则是从上到下，从左到右进行。如图所示。



# 编程规则二

➤ 触点应画在水平支路上，不能画在垂直支路上

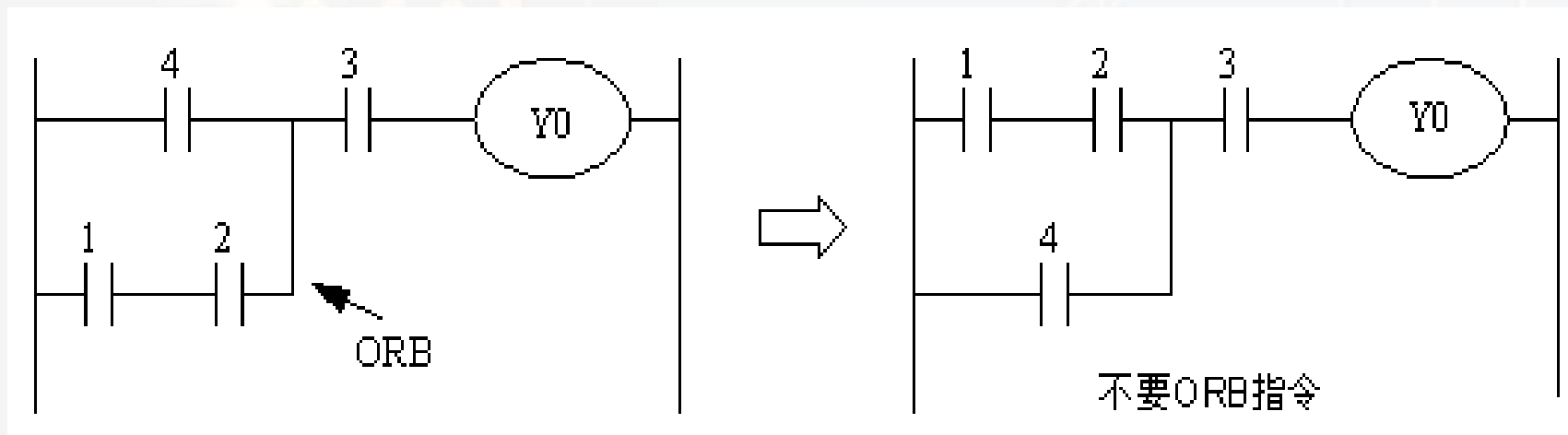


(a)

(b)

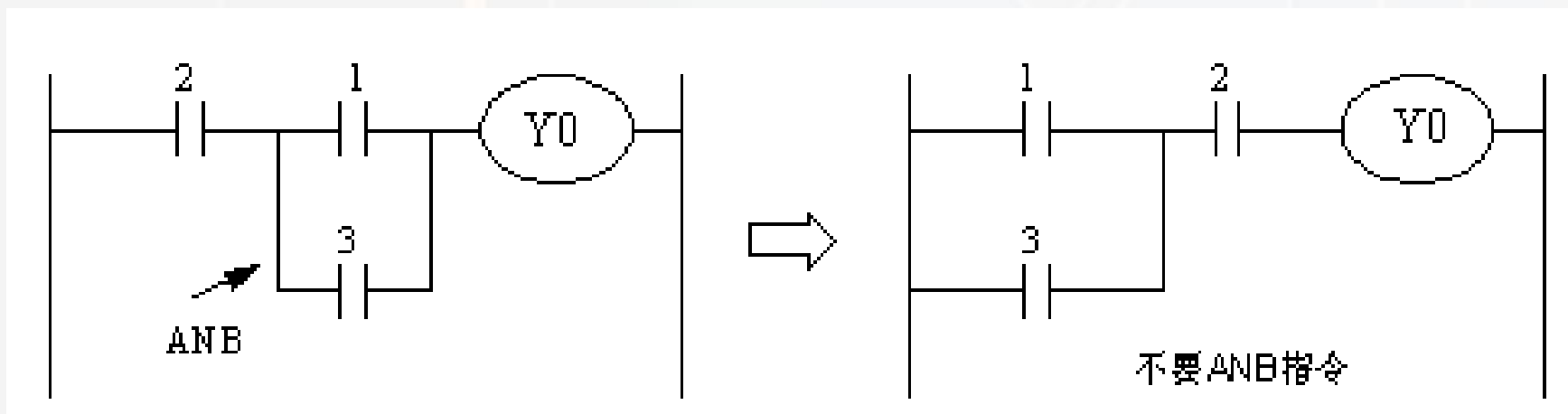
(a) 不正确 (b) 正确

- 几条支路并联时，串联触点多的，安排在上面（先画），如图所示



# 编程规则四

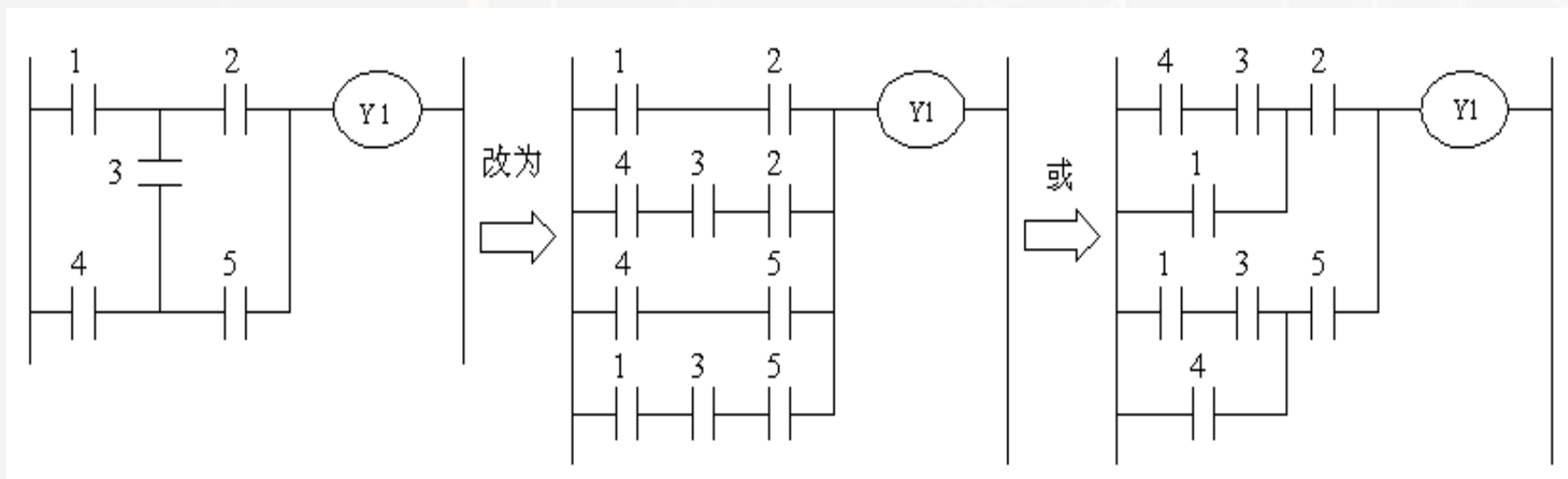
- 几个支路串联时，并联触点多的支路块安排在左面，如图所示





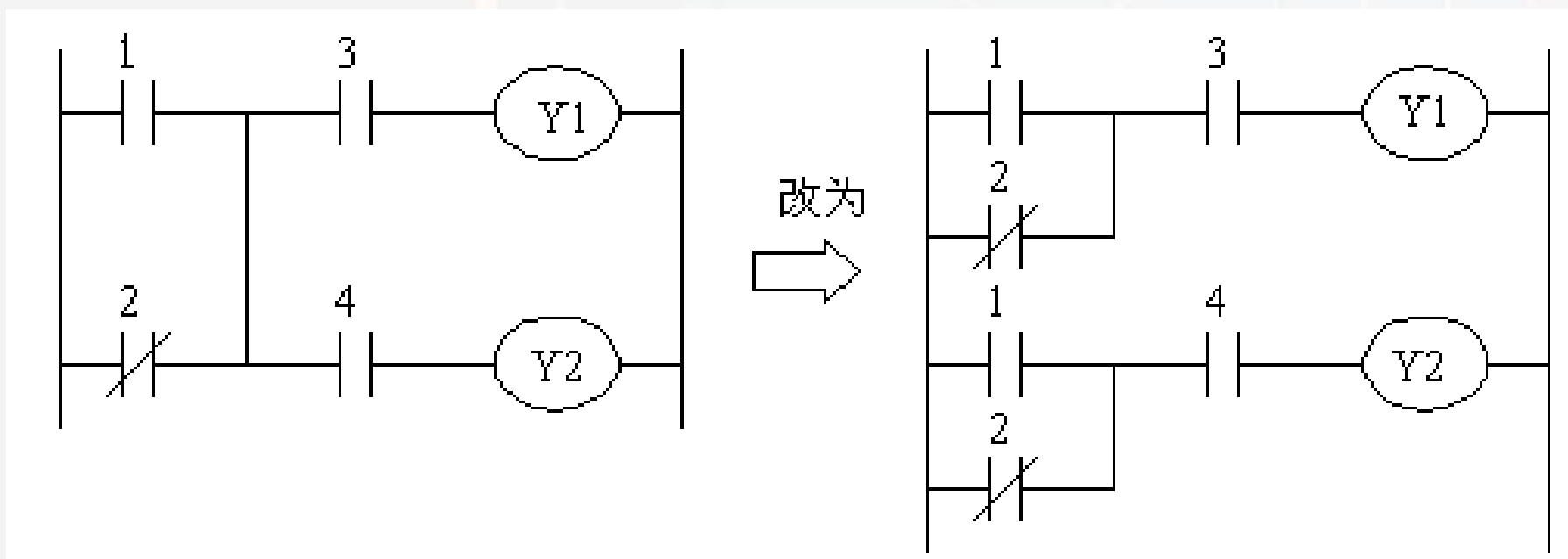
# 编程规则五

➤ 一个触点不允许有双向电流通过。当出现这种情况时，按图示例改

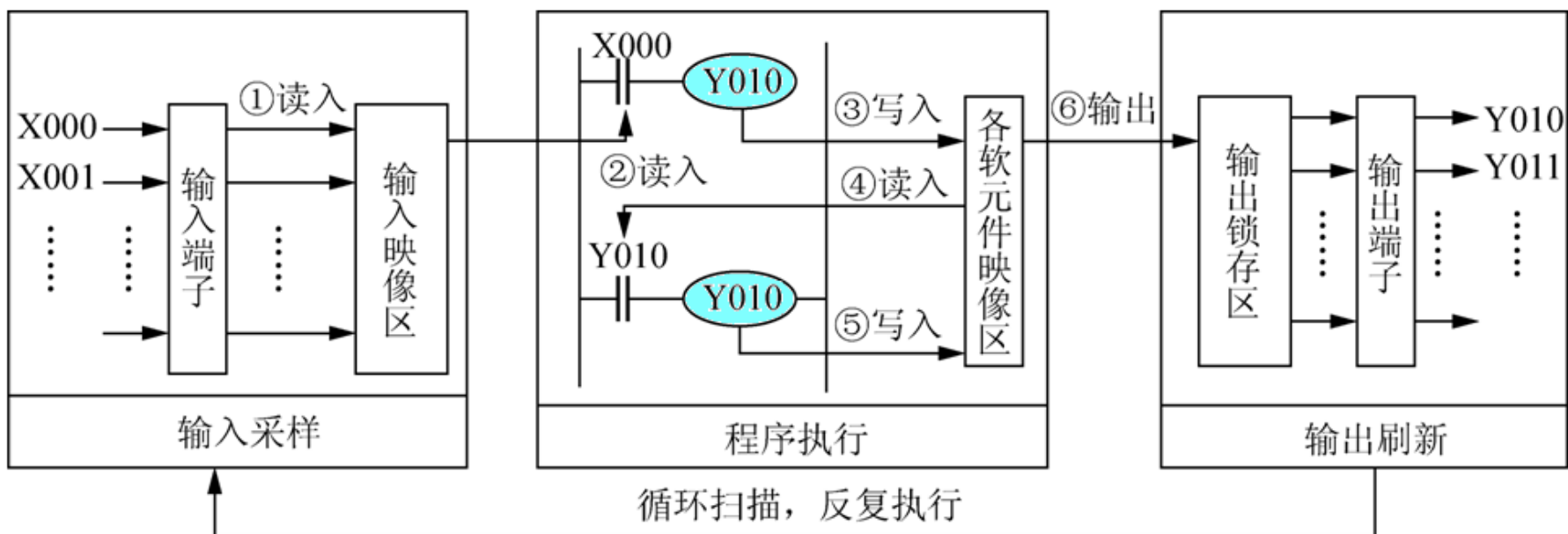


# 编程规则六

- 当两个逻辑行之间互有牵连时，如图所示，可按图示的方法加以改画



- 在梯形图中任一支路上的串联触点、并联触点以及内部并联线圈的个数一般不受限制，但有的PLC有自己的规定，应注意看说明书。
- 若在顺序控制中进行线圈的双重输出（双线圈），则后面的动作优先执行。





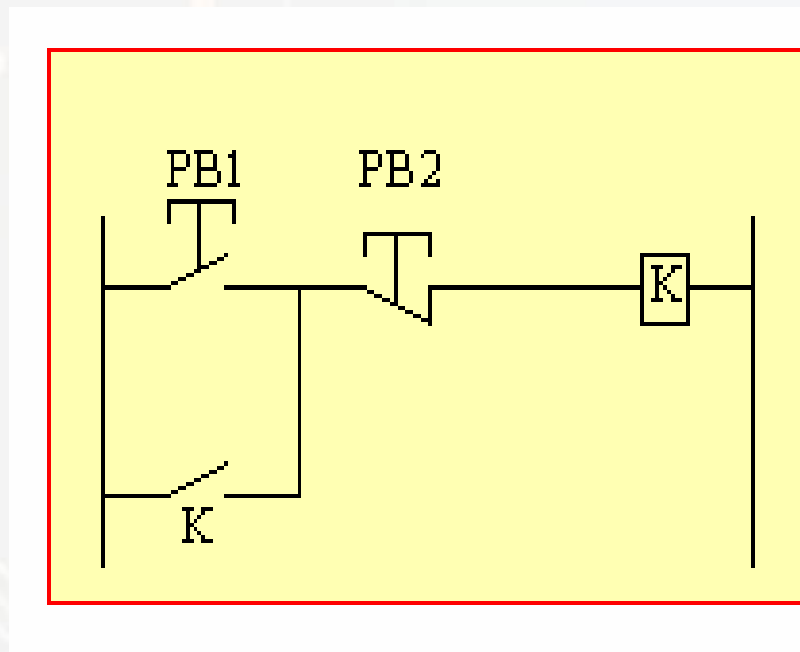
# 外部输入触点与梯形图触点处理



➤ 绘图时,应注意PLC外部所接“输入信号”的触点状态,与梯形图中所采用内部输入触点(X编号的触点)的关系。

✓ 继电器控制电路

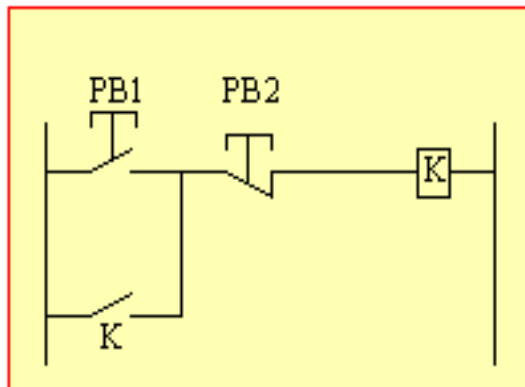
启动按钮PB1用**常开**按钮, 停止按钮PB2用**常闭**按钮。



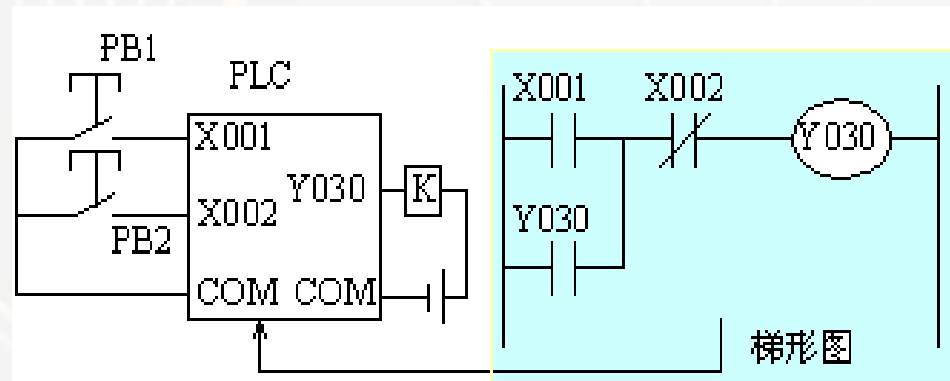
转换成PLC控制,  
如何?

## □ 外部输入条件与PLC的连接形式之一

- ✓ 继电器控制电路中启动按钮PB1用常开按钮，停止按钮PB2用常闭按钮。

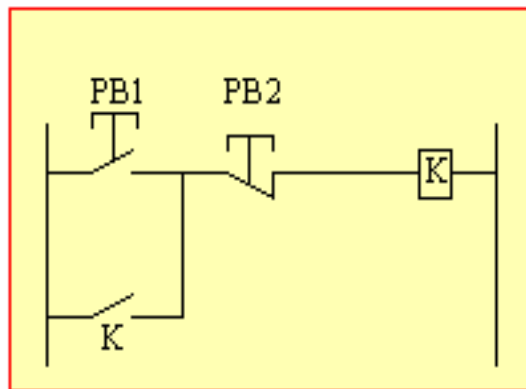


- ✓ 接入PLC时，PB1用常开按钮，PB2也用常开按钮时，则在梯形图设计时X001用常开触点，X002用常闭触点。

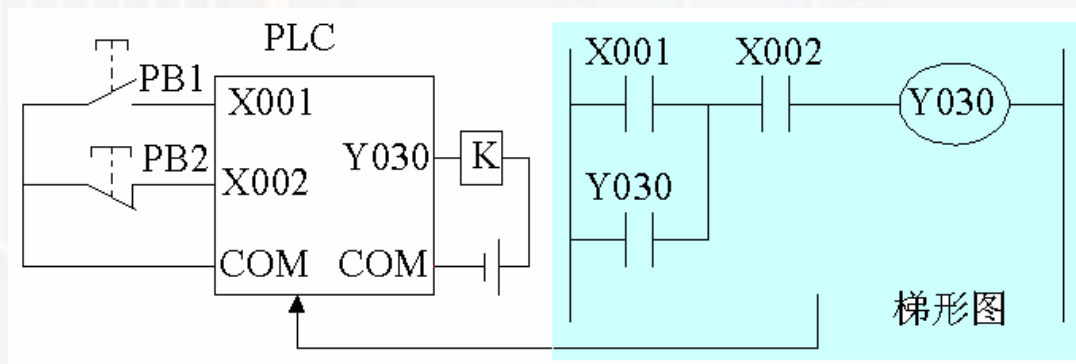


## □ 外部输入条件与PLC的连接形式之二

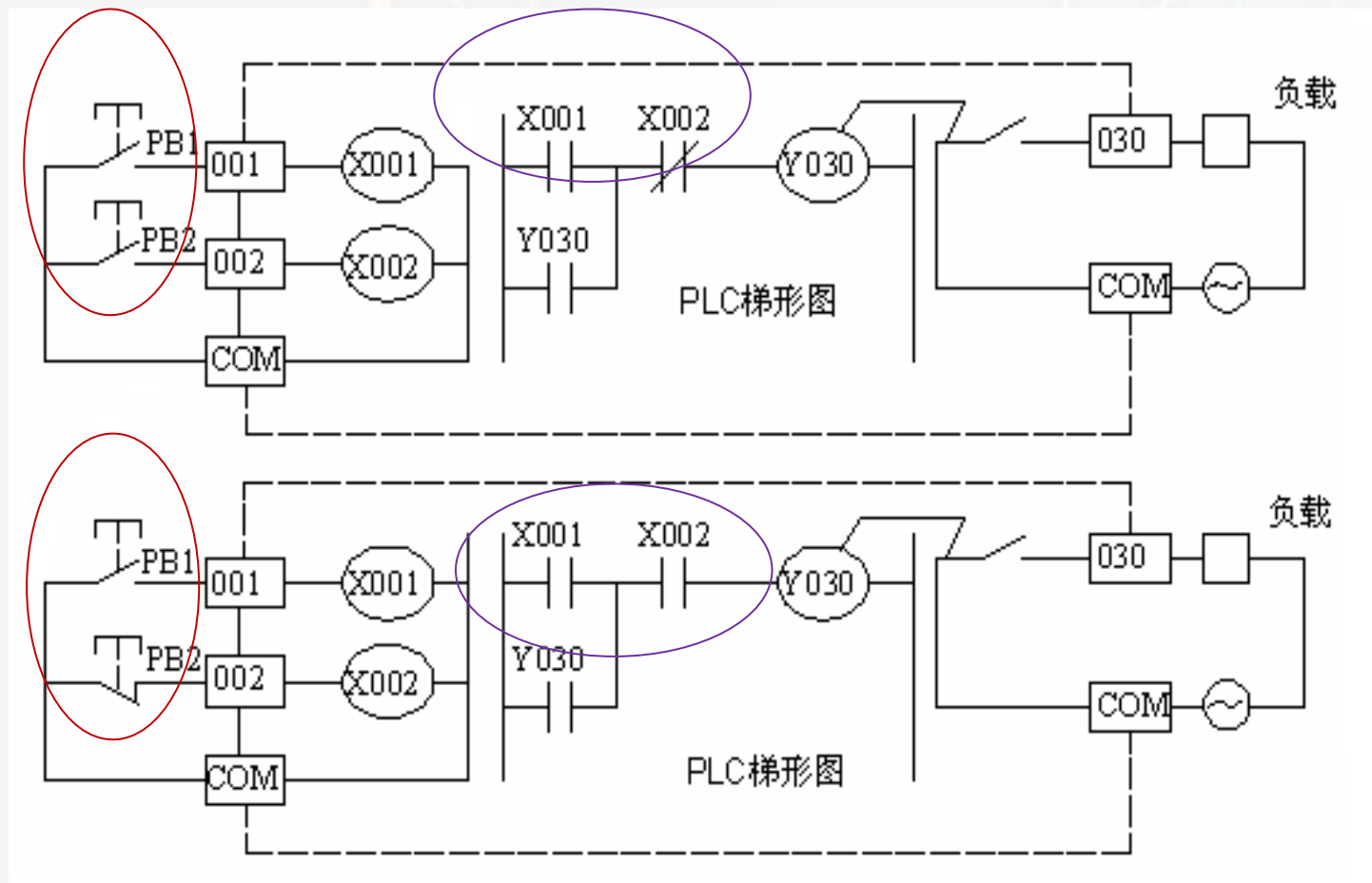
- ✓ 启动按钮PB1用常开按钮，停止按钮PB2用常闭按钮，



- ✓ 接入PLC，PB1用**常开**按钮，PB2用**常闭**按钮，则在梯形图设计时X001用**常开**触点，X002也应用**常开**触点。



输入得电？



控制逻辑

两种外部输入条件与PLC梯形图的关系

外部输入条件与梯形图编程的关系



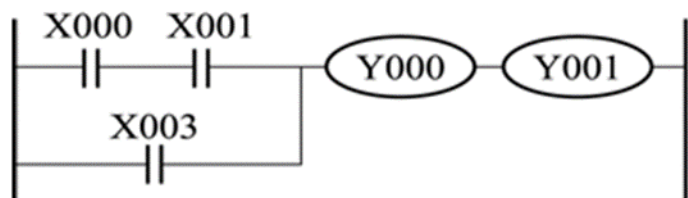


武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

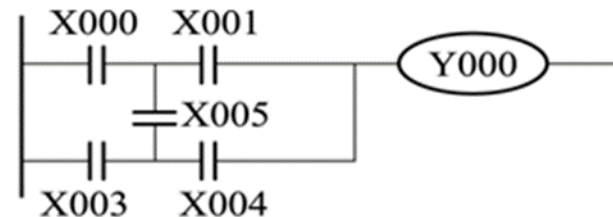
# 测试

---

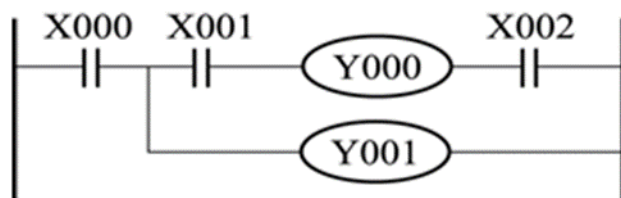
指出图中所示梯形图的错误，并在不改变逻辑关系的前提下画出正确的梯形图。



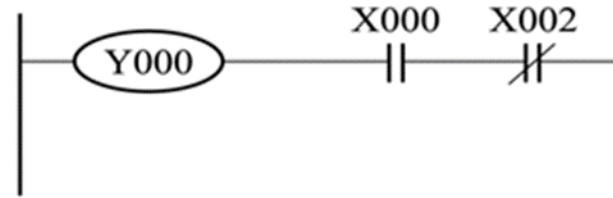
(a)



(b)



(c)



(d)





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

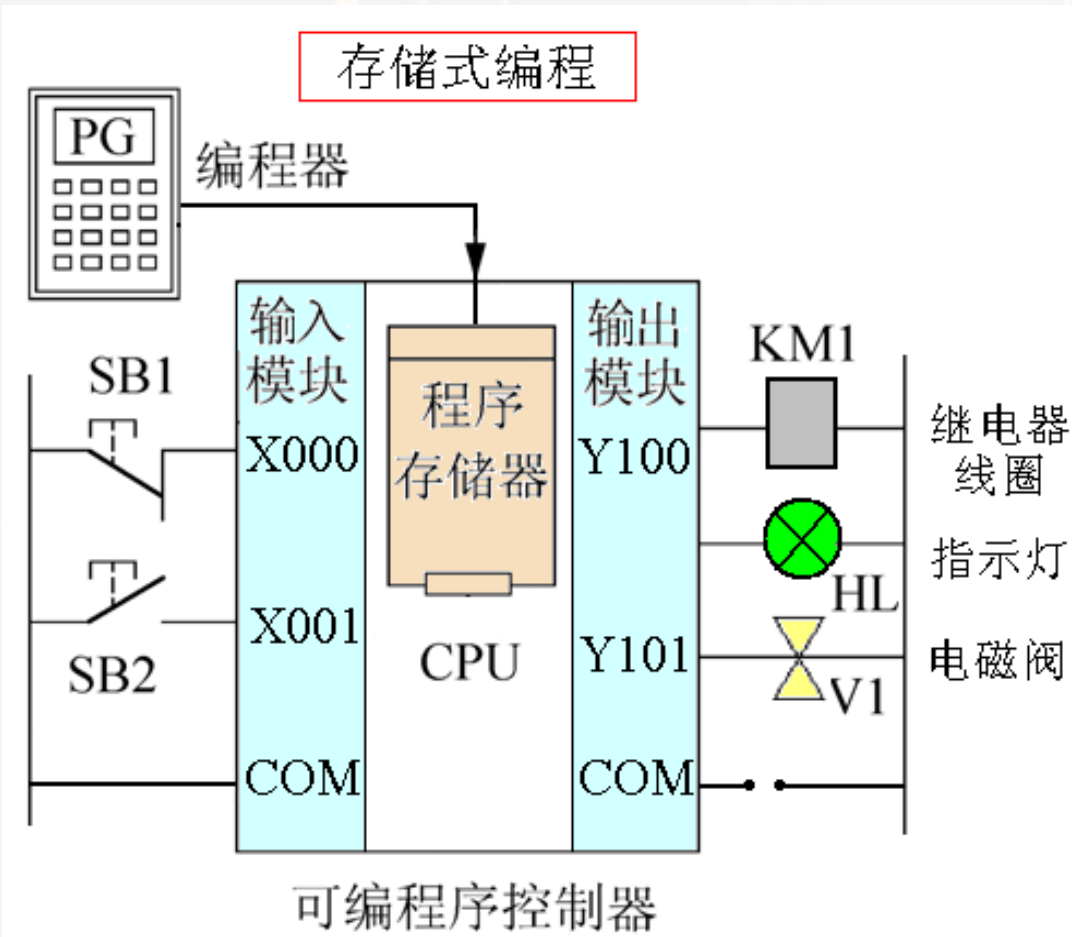
# 梯形图设计的基本步骤

# ◆根据控制系统控制要求确定PLC机型

- 分析被控对象的具体情况（生产过程，技术特点，工艺方法，环境条件），研究对控制系统的要求。
- 根据被控对象状态参数的数目和被采集信号的数目，确定PLC的输出/输入点数，以此作为选择PLC机型的条件。
- 根据被采集及被控制信号的特点（数字量，模拟量）以及所需电源的情况，确定输入器件，输出执行器件及接线方式。结合上面的条件选择PLC的型号。



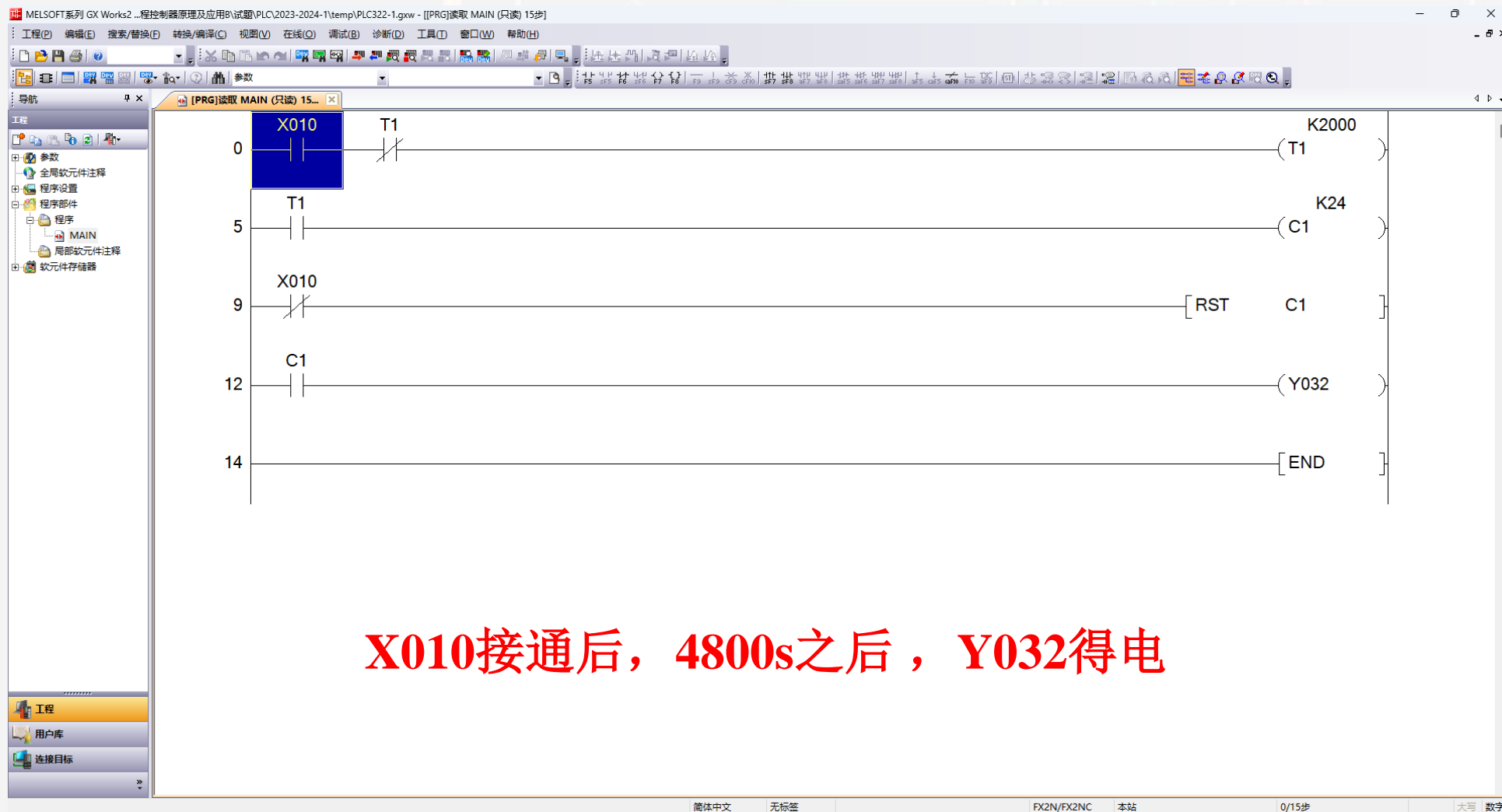
# ◆设计PLC输入/输出信号连接图



PLC的外围接线图

# ◆ 控制程序编写

## GX Works2-工具



**X010接通后，4800s之后，Y032得电**

# ◆输入并编辑程序

The screenshot displays the MELSOFT GX Works2 software interface. On the left, the 'File' menu is open, showing options like 'New (N)...', 'Open (O)...', 'Save (S)', and 'Print (P)'. The main window shows the 'New' dialog box with the following settings:

- 系列(S): FXCPU
- 机型(T): FX2N/FX2NC
- 工程类型(P): 简单工程
- 程序语言(G): 梯形图

The 'Use Label' (使用标签) checkbox is unchecked. The 'OK' (确定) and 'Cancel' (取消) buttons are at the bottom right of the dialog box.

Below the dialog box, the main editor window is visible, showing a ladder logic program. The program starts with a normally open contact labeled '0' connected to a coil labeled 'END'. The status bar at the bottom indicates the current step is '[PRG]写入 MAIN 1步'.

# ◆输入并编辑程序

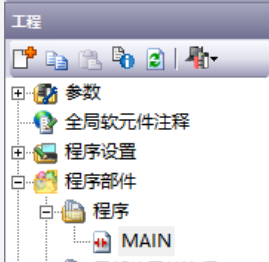
MELSOFT系列 GX Works2 (工程未设置) - [[PRG]写入 MAIN 1步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

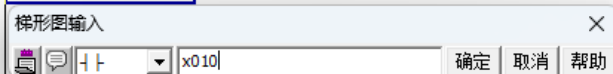
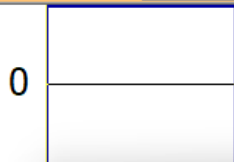


参数

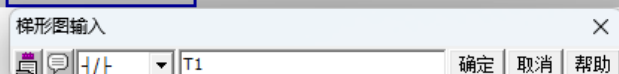
导航



[PRG]写入 MAIN 1步



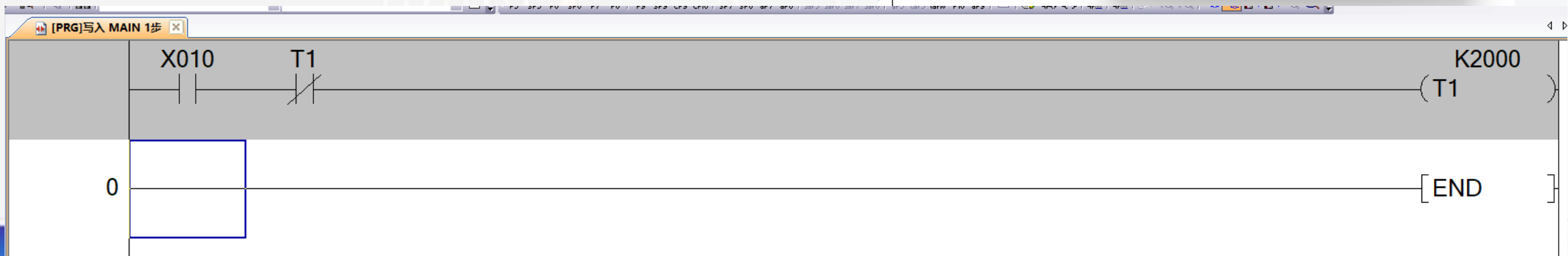
MAIN 1步



0



MAIN 1步





# ◆ 程序编写

MELSOFT系列 GX Works2 ...程控制原理及应用B\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]读取 MAIN (只读) 15步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

参数

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
    - 局部软元件注释
  - 软元件存储器

[[PRG]读取 MAIN (只读) 15...

0 X010 T1 K2000 (T1)

5 T1 K24 (C1)

9 X010 [ RST C1 ]

12 C1 (Y032)

14 [ END ]

工程

用户库

连接目标

简体中文 无标签 FX2N/FX2NC 本站 0/15步 大写 数字

**X010接通后，4800s之后，Y032得电**

# ◆ 程序编写-转换

MELSOFT系列 GX Works2 ...程控制器原理及应用\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]写入 MAIN 15步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

转换(B) F4  
转换+RUN中写入(O) Shift+F4  
转换(所有程序)(R) Shift+Alt+F4

导航

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
  - 局部软元件注释
- 软元件存储器

0 X001 T1 K2000 (T1)

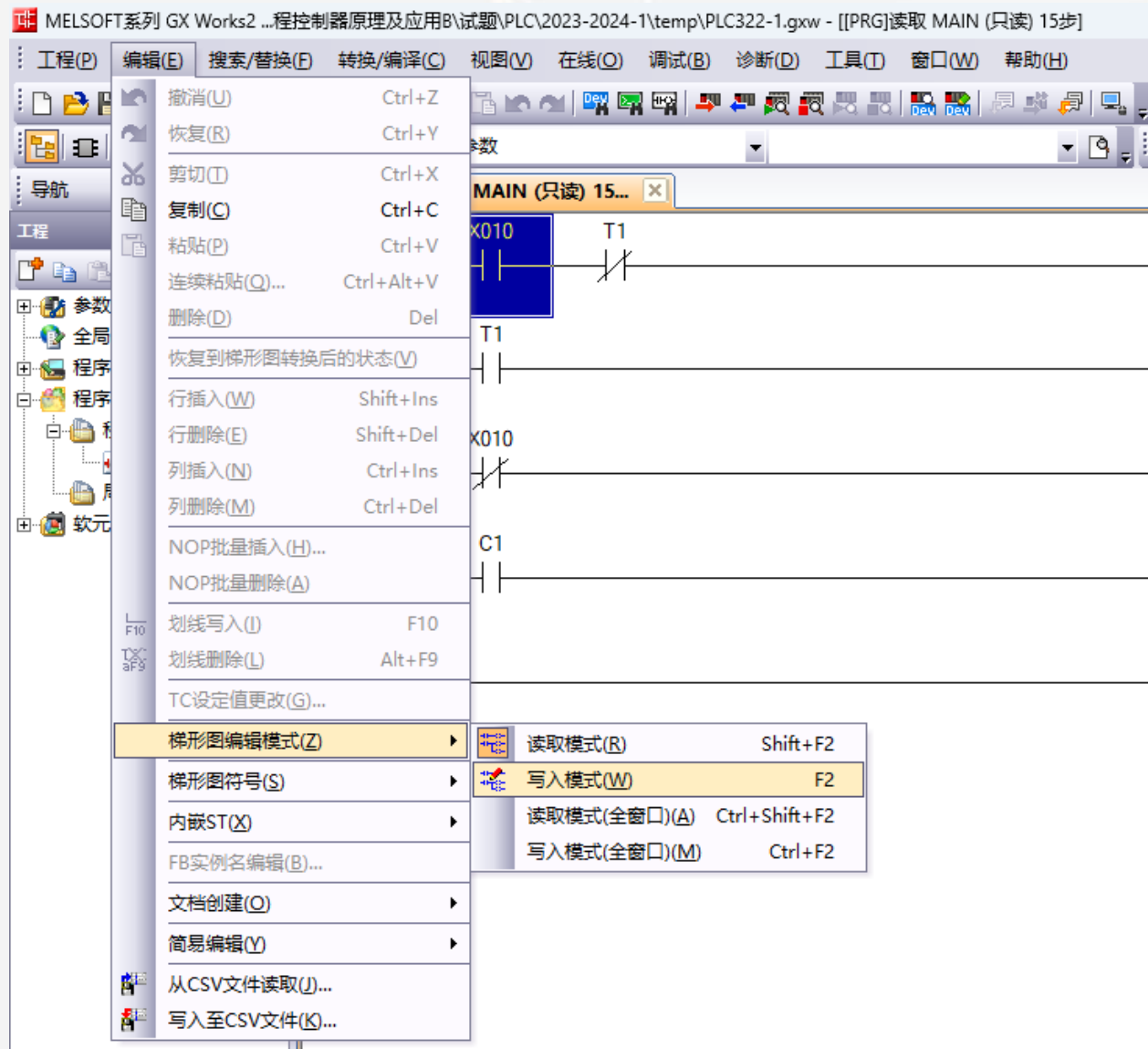
5 T1 K24 (C1)

9 X001 [RST C1]

12 C1 (Y032)

14 [END]

# ◆ 程序编写-写入模式



# ◆程序调试

# PLC仿真测试

MELSOFT系列 GX Works2 ...程控制器原理及应用B\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]写入 MAIN 15步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

模拟开始/停止(S)

- 显示模拟不支持的指令(N)
- 当前值更改(M)...
- 强制输入输出登录/解除(F)...
- 附带执行条件的软元件测试(E) ▶
- 采样跟踪(A) ▶
- 扫描时间测定(C)...
- 步执行(X) ▶
- 中断设置(B) ▶
- 跳过设置(K) ▶

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
- 局部软元件注释
- 软元件存储器

[PRG]写入 MAIN 15步

0 X001 T1

5 T1

9 X001

12 C1

14

K2000 (T1)

K24 (C1)

[RST C1]

(Y032)

[END]



# PLC仿真测试

MELSOFT系列 GX Works2 ...程控制器原理及应用B\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]写入 MAIN 15步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

参数

导航

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
    - 局部软元件注释
- 软元件存储器

[PRG]写入 MAIN 15步

0 X001 T1 K2000

5 T1 K24

9 X001

12 C1

14

RST C1

Y032

END

GX Simulator2

工具(T) 选项(O)

开关

STOP RUN

LED

POWER RUN ERR.

PLC写入

参数 写入 : 完成

程序 (MAIN) 写入 : 完成

PLC写入 : 结束

☐ 处理结束时, 自动关闭窗口。

关闭

# PLC仿真测试

MELSOFT系列 GX Works2 ...程控制器原理及应用B\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]监视 执行中 MAIN (只读) 15步]

工程(P) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

100ms

参数

导航

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
    - 局部软元件注释
- 软元件存储器

[PRG]监视 执行中 MAIN (...)

0 X001 T1 K2000

5 T1 K24

9 X001 RST C1

12 C1 Y032

14 [END]

GX Simulator2

工具(T) 选项(O)

开关

STOP RUN

LED

POWER RUN ERR.

# PLC仿真测试

MELSOFT系列 GX Works2 ...程控制原理及应用8\试题\PLC\2023-2024-1\temp\PLC322-1.gxw - [[PRG]监视 执行中 MAIN (只读) 15步]

工程(E) 编辑(E) 搜索/替换(F) 转换/编译(C) 视图(V) 在线(O) 调试(B) 诊断(D) 工具(T) 窗口(W) 帮助(H)

100ms

参数

导航

工程

- 参数
- 全局软元件注释
- 程序设置
- 程序部件
  - 程序
    - MAIN
    - 局部软元件注释
    - 软元件存储器

[PRG]监视 执行中 MAIN (...)

0 X001 T1 K2000

5

9 X001 C1 K24

12

14

隐藏梯形图块(D)

显示梯形图块(L)

搜索(F)

视图(V)

在线(O)

调试(G)

交叉参照(S)

软元件使用列表(I)

登录至监督窗口(H)

登录至软元件批量更改(A)

打开指令帮助(R)...

GX Works2帮助(W)...

从CSV文件读取(J)...

写入至CSV文件(K)...

当前值更改(M)...

强制输入输出登录/解除(F)...

附带执行条件的软元件测试登录(R)...

附带执行条件的软元件测试批量解除(B)

附带执行条件的软元件测试登录确认(C)...

附带执行条件的软元件测试登录解除(D)...

中断点设置/解除(P)

跳过范围设置/解除(S)

中断点启用/禁用(E)

[PRG]监视 执行中 MAIN (...)

0 X001 T1 K2000

5 T1 C1 K24

9 X001

12 C1

14

当前值更改

软件/标签 | 缓冲存储器 |

软件/标签(E)

X001

数据类型(T) Bit

ON OFF ON/OFF取反(I)

可输入范围

执行结果(R) ▲ 关闭

执行结果(L)

软件/标签	数据类型	设定值
X001	Bit	ON

反映到输入栏(B) 删除(C)

GX Simulator2

工具(T) 选项(O)

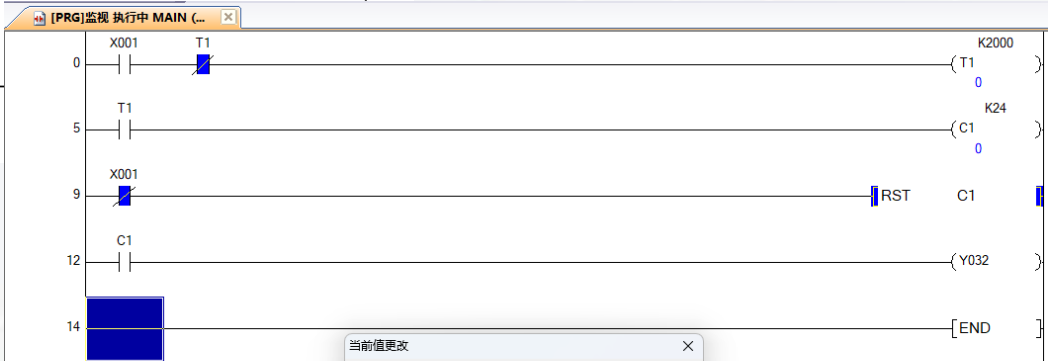
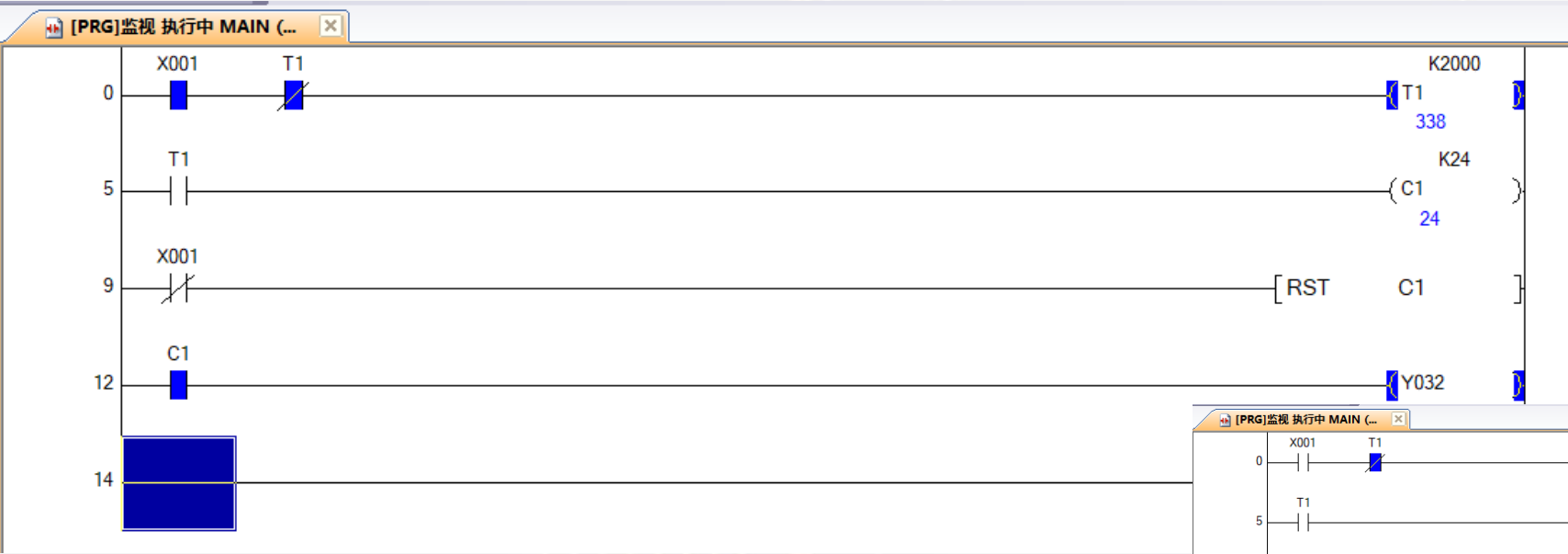
开关

STOP RUN

LED

POWER RUN ERR.

# PLC仿真结果



当前值更改

软元件/标签 | 缓冲存储器 |

软元件/标签(E)  
X001

数据类型(T) Bit

ON OFF ON/OFF取反(!)

可输入范围

执行结果(R)▲ 关闭

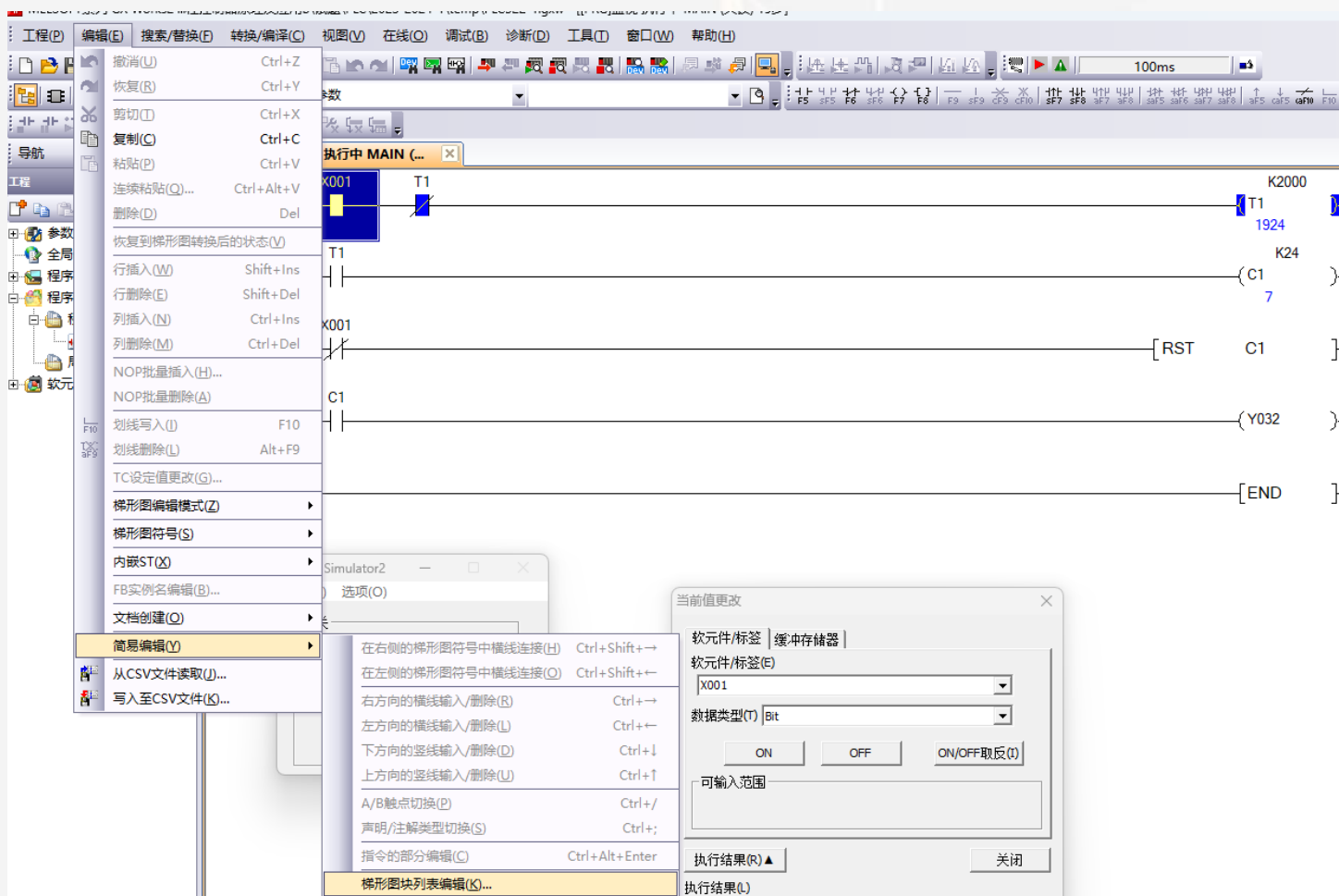
执行结果(Q)

软元件/标签	数据类型	设定值
X001	Bit	OFF
X001	Bit	ON

反映到输入栏(B) 删除(C)



# 梯形图与指令列表



The screenshot shows the Siemens STEP 7 software interface. The main window displays a ladder logic diagram with the following components:

- Step 0: LD X001
- Step 1: ANI T1
- Step 2: OUT T1 K2000
- Step 3: T1 (Timer T1, K2000)
- Step 4: C1 (Counter C1, K24)
- Step 5: RST C1
- Step 6: Y032 (Output Y032)
- Step 7: END

The 'Simple Edit' menu is open, showing various editing options:

- 撤消(U) Ctrl+Z
- 恢复(R) Ctrl+Y
- 剪切(I) Ctrl+X
- 复制(C) Ctrl+C
- 粘贴(P) Ctrl+V
- 连续粘贴(Q) Ctrl+Alt+V
- 删除(D) Del
- 恢复到梯形图转换后的状态(V)
- 行插入(W) Shift+Ins
- 行删除(E) Shift+Del
- 列插入(N) Ctrl+Ins
- 列删除(M) Ctrl+Del
- NOP批量插入(H)...
- NOP批量删除(A)
- 划线写入(I) F10
- 划线删除(L) Alt+F9
- TC设定值更改(G)...
- 梯形图编辑模式(Z)
- 梯形图符号(S)
- 内嵌ST(X)
- FB实例名编辑(B)...
- 文档创建(O)
- 简易编辑(Y)
- 从CSV文件读取(I)...
- 写入至CSV文件(O)...

The 'Simple Edit' menu also includes a 'Ladder Block List Edit (Read)' option, which is highlighted.

梯形图块列表编辑(读取)

步	列表
0	LD X001
1	ANI T1
2	OUT T1 K2000

梯形图块列表编辑(读取)

步	列表
5	LD T1
6	OUT C1 K24

梯形图块列表编辑(读取)

步	列表
9	LDI X001
10	RST C1

梯形图块列表编辑(读取)

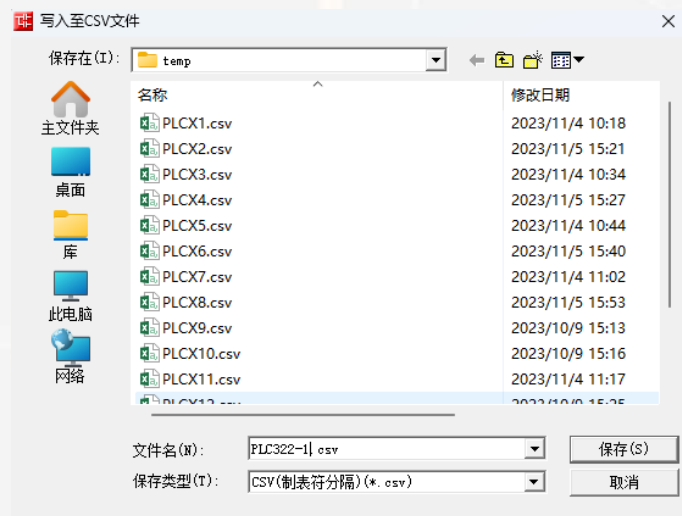
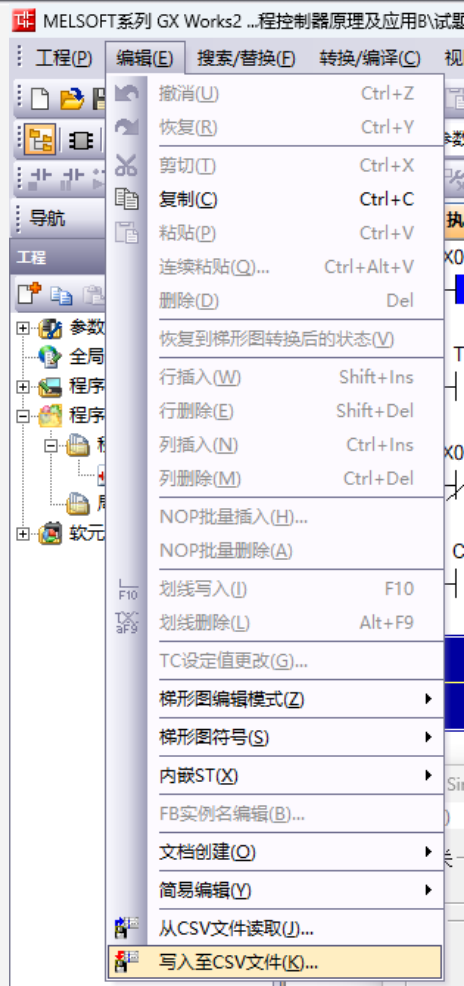
步	列表
12	LD C1
13	OUT Y032

Shift + Insert: 行插入 Delete: 行删除  
Ctrl + C: 复制 Ctrl + X: 剪切 Ctrl + V: 粘贴

确定

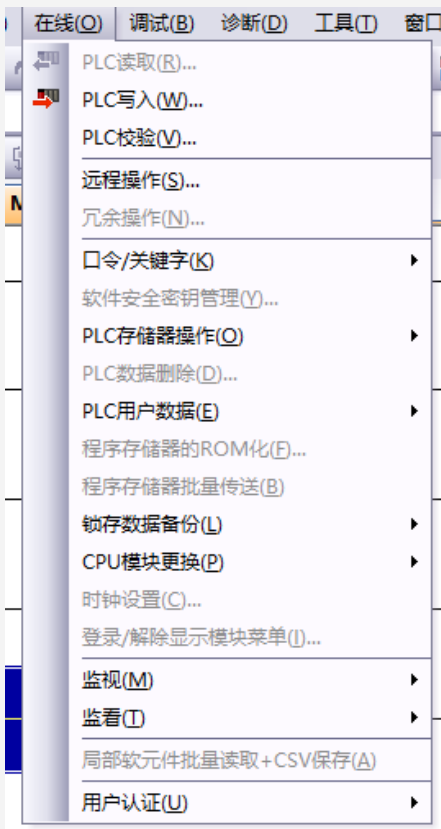
取消

# 梯形图转换成CSV指令文件



PLC322-1						
	A	B	C	D	E	F
1	PLC322-1					
2	PLC信息:	FXCPU FX2N/FX2NC				
3	步号	行间声明	指令	I/O(软元件	空白栏	PI声明
4	0		LD	X001		注解
5	1		ANI	T1		
6	2		OUT	T1		
7				K2000		
8	5		LD	T1		
9	6		OUT	C1		
10				K24		
11	9		LDI	X001		
12	10		RST	C1		
13	12		LD	C1		
14	13		OUT	Y032		
15	14		END			

# ◆程序传送存储







武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---



根据指令表画出对应的梯形图。

步 序	指 令	步 序	指 令
0	LD X000	6	ORB
1	OR X001	7	OR X006
2	LD X002	8	ANB
3	AND X003	9	OR X003
4	LDI X004	10	OUT Y000
5	AND X005	11	END





武汉理工大学

WUHAN UNIVERSITY OF TECHNOLOGY

## 3.3 基本逻辑指令应用举例





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-延时释放



# 【例】延时释放电路的PLC程序设计。

控制要求：

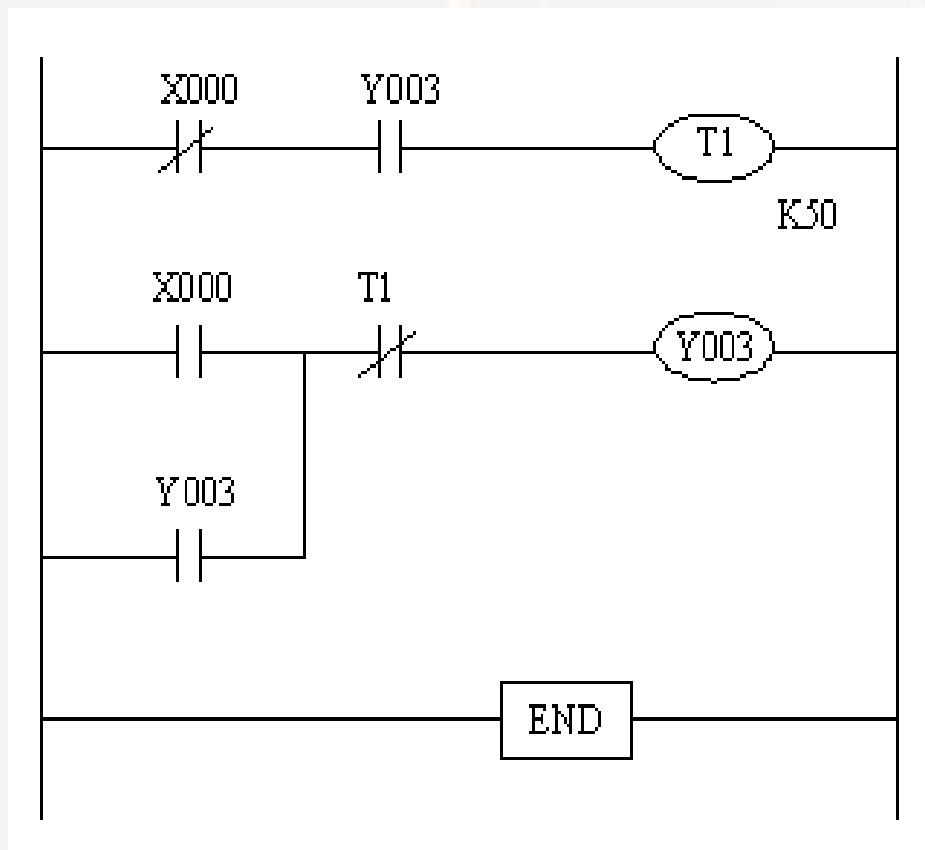
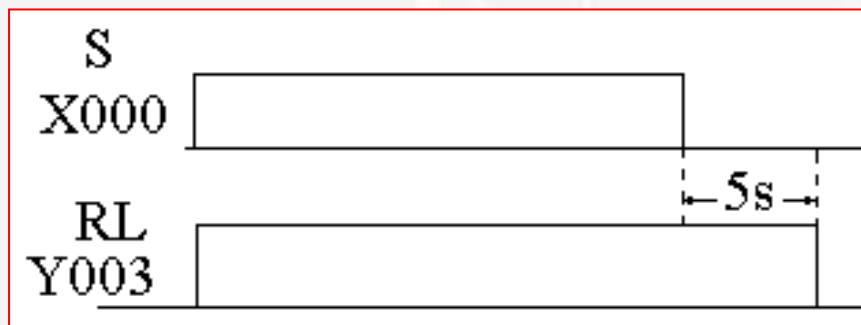
➤ 控制开关：  $\begin{cases} S = \text{ON} & \checkmark \text{ 负载： } RL = \text{ON} \\ S = \text{OFF} & \checkmark \text{ 5秒后： } RL = \text{OFF} \end{cases}$

解： (1) 输入/输出接点分配表

输入装置	PLC输入端口号	输出装置	PLC输出端口号
S	X000	RL	Y003



## (2) 控制逻辑梯形图



**LDI X000**

**AND Y003**

**OUT T1  
K50**

**LD X000**

**OR Y003**

**ANI T1**

**OUT Y003**

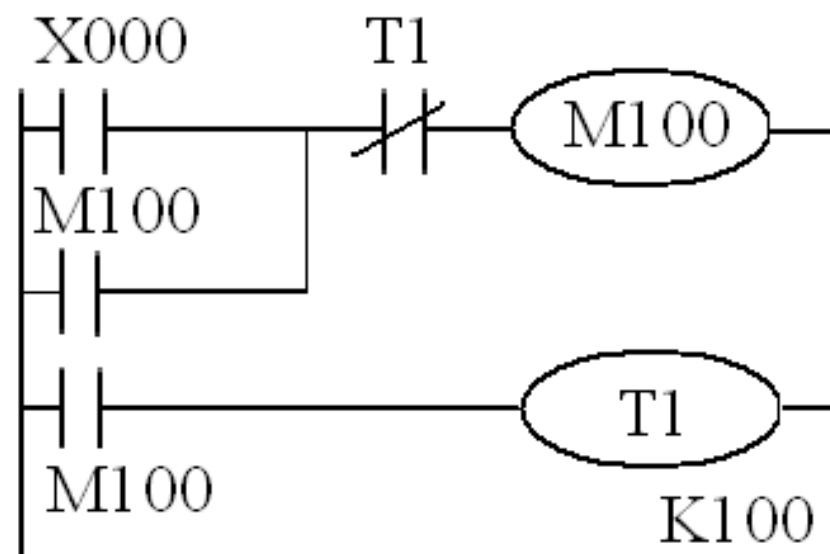
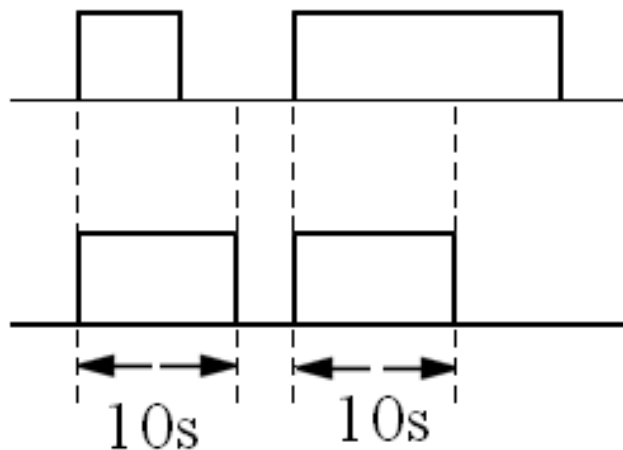
**END**

# 【例】输入信号保持定时输出的PLC程序设计

**控制要求：将输入信号X000的ON状态保持一定时间。  
不输出。**

输入信号  
X000

保持定时  
输出信号





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-触发器电路



# 【例】 触发器电路（分频电路）的PLC程序设计

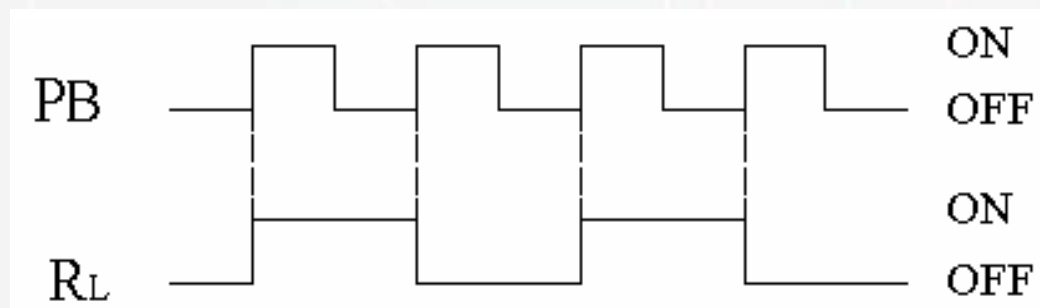


武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

控制要求：

PB为控制按钮

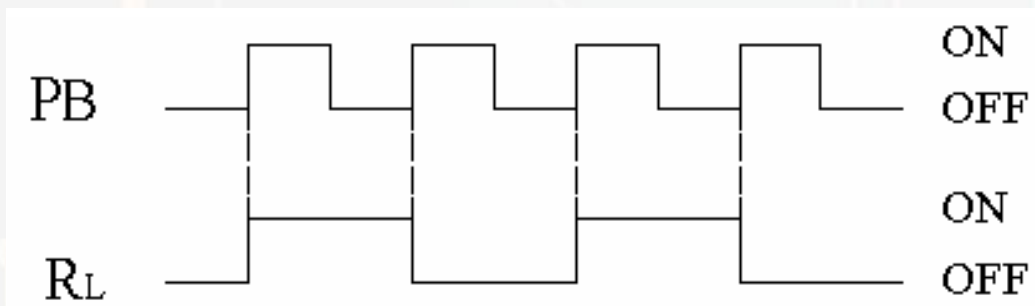
RL为输出负载



时序图

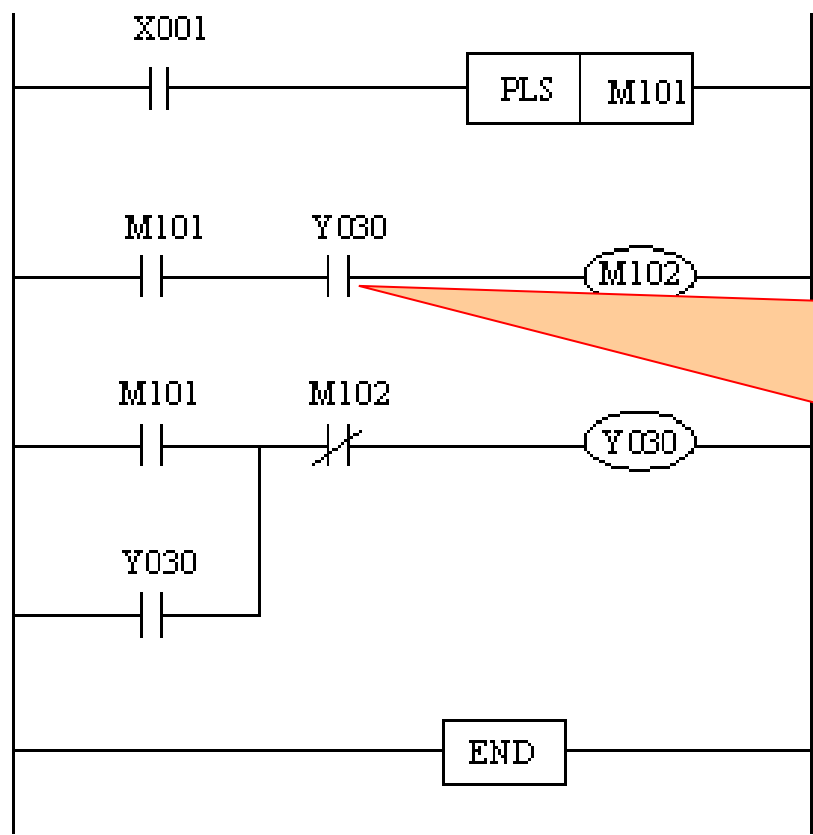
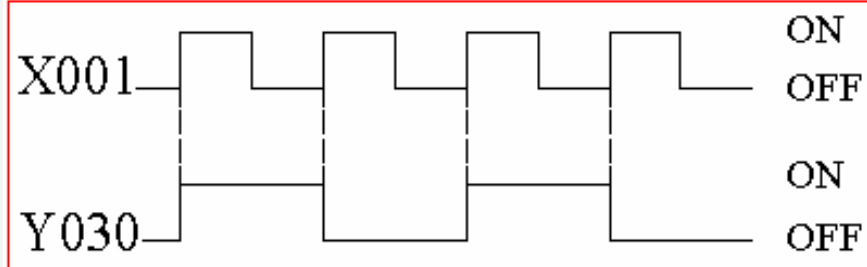
# 【例】 触发器电路（分频电路） 的PLC程序设计

解： （1） 输入/输出接点分配表



输入装置	PLC输入端子号	输出装置	PLC输出端子号
PB	X001	RL	Y030

## (2) 触发器控制逻辑梯形图



**注意Y030的第1和第2次周期。第2个周期Y030就一直通。**

**X001第二次  
ON,M101=ON,  
使M102 = ON,  
--Y030=OFF**

<b>LD</b>	<b>X001</b>
<b>PLS</b>	<b>M101</b>
<b>LD</b>	<b>M101</b>
<b>AND</b>	<b>Y030</b>
<b>OUT</b>	<b>M102</b>
<b>LD</b>	<b>M101</b>
<b>OR</b>	<b>Y030</b>
<b>ANI</b>	<b>M102</b>
<b>OUT</b>	<b>Y030</b>
<b>END</b>	



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

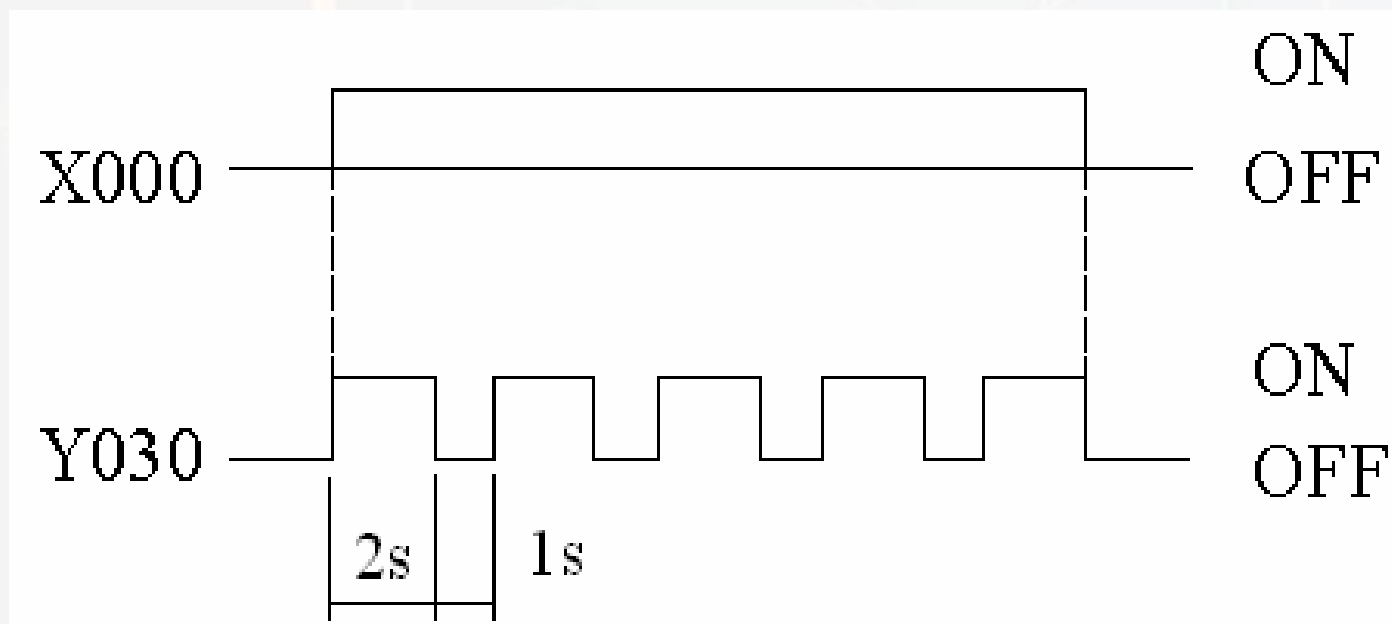
# 编程-振荡电路



# 【例】振荡电路的PLC程序设计

控制要求

- 输入信号X000 = ON，  
输出Y030按一定的周期通断。
- X000 = OFF，则Y030 = OFF

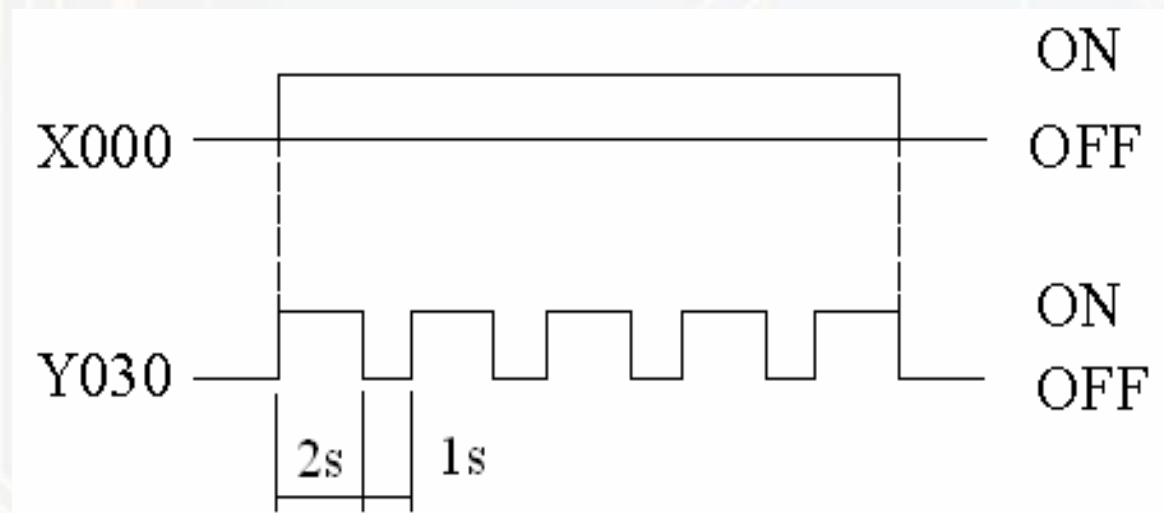


时序图

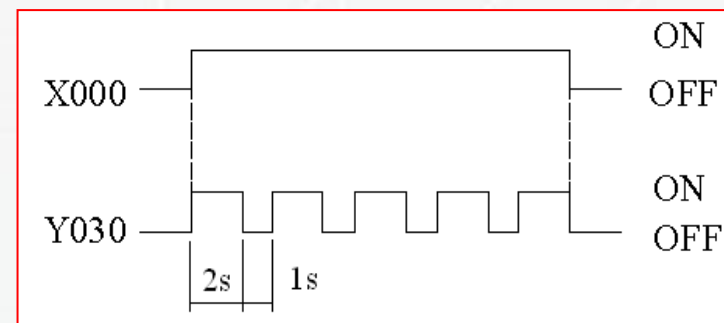
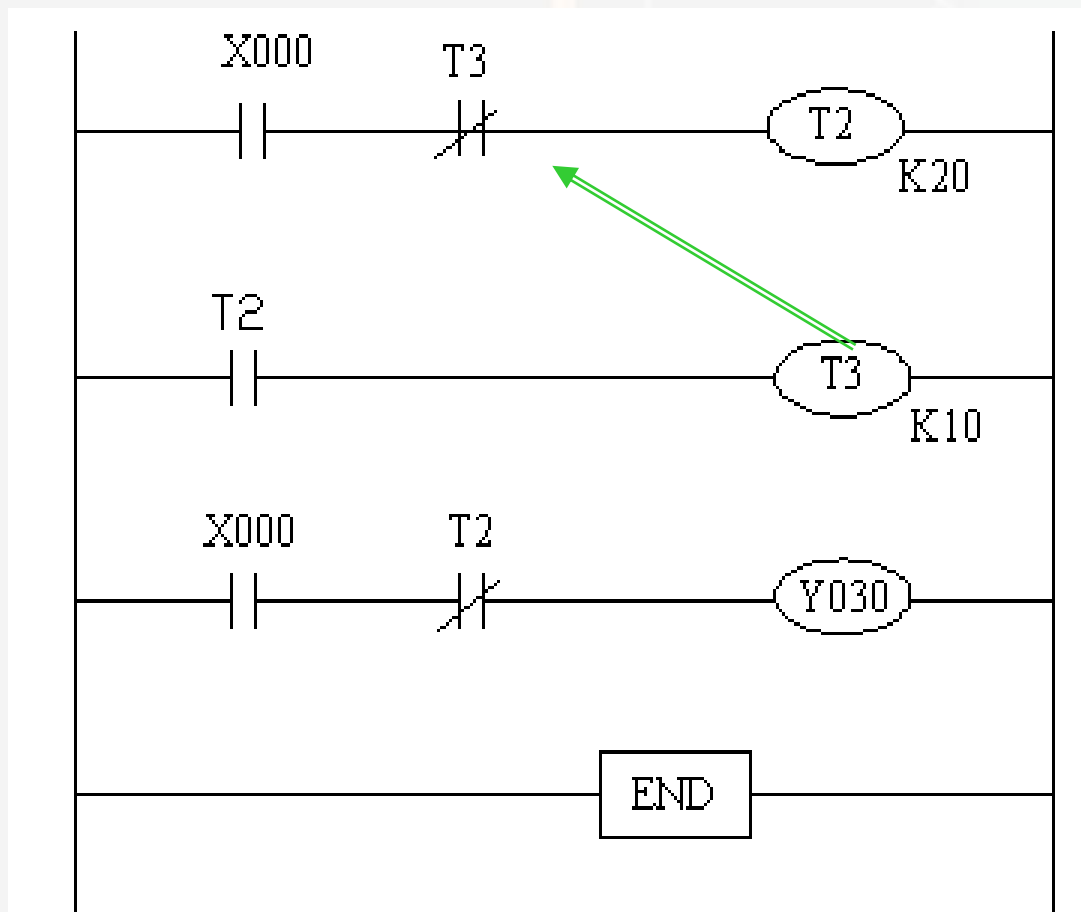
# 解：

## (1) I/O分配表

输入装置	PLC输入端子号	输出装置	PLC输出端子号
PB	X000	灯	Y030



## (2) 控制逻辑梯形图



通电时间(2s)

断电时间(1s)



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-报警电路



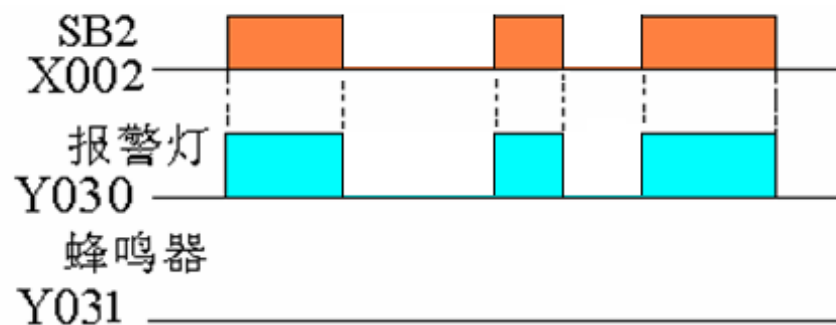
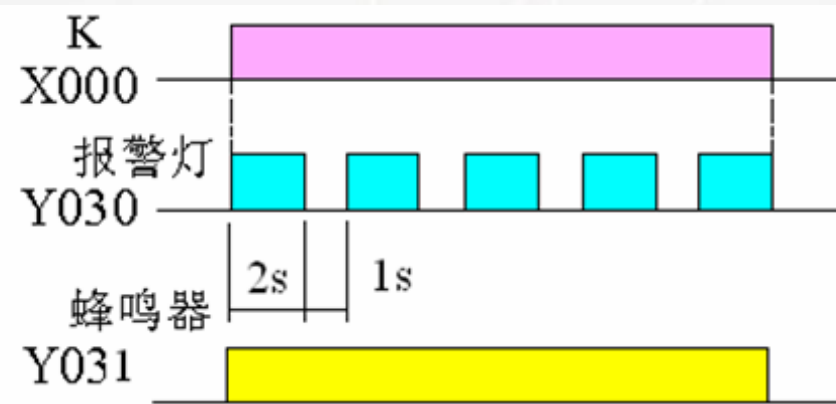
# 【例】报警电路的PLC程序设计

控制要求:

➤报警继电器**K** = ON,报警灯闪烁, 蜂鸣器叫。

➤当报警响应按钮**SB1**按下, 报警灯常亮, 蜂鸣器停叫。

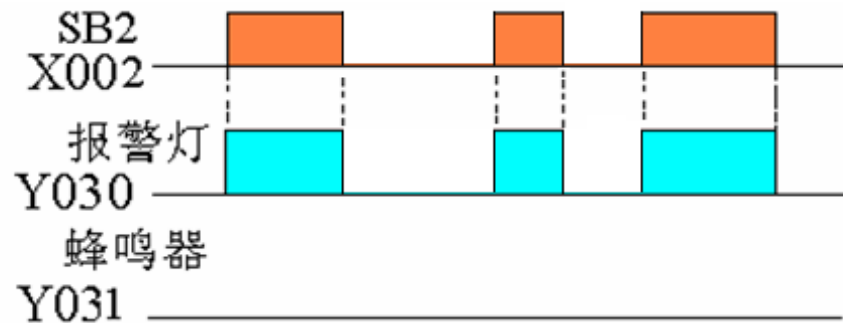
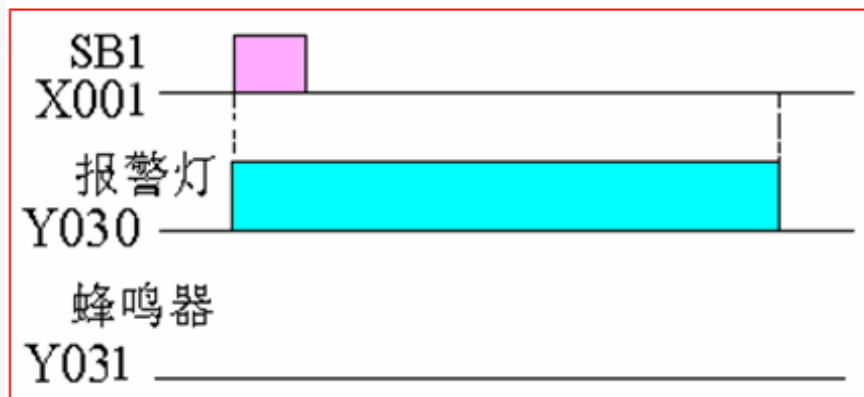
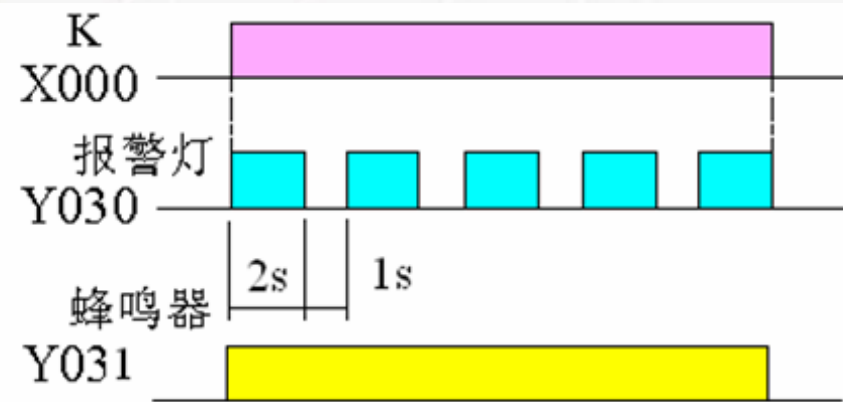
➤当报警灯测试按钮**SB2**按下时, 报警灯亮

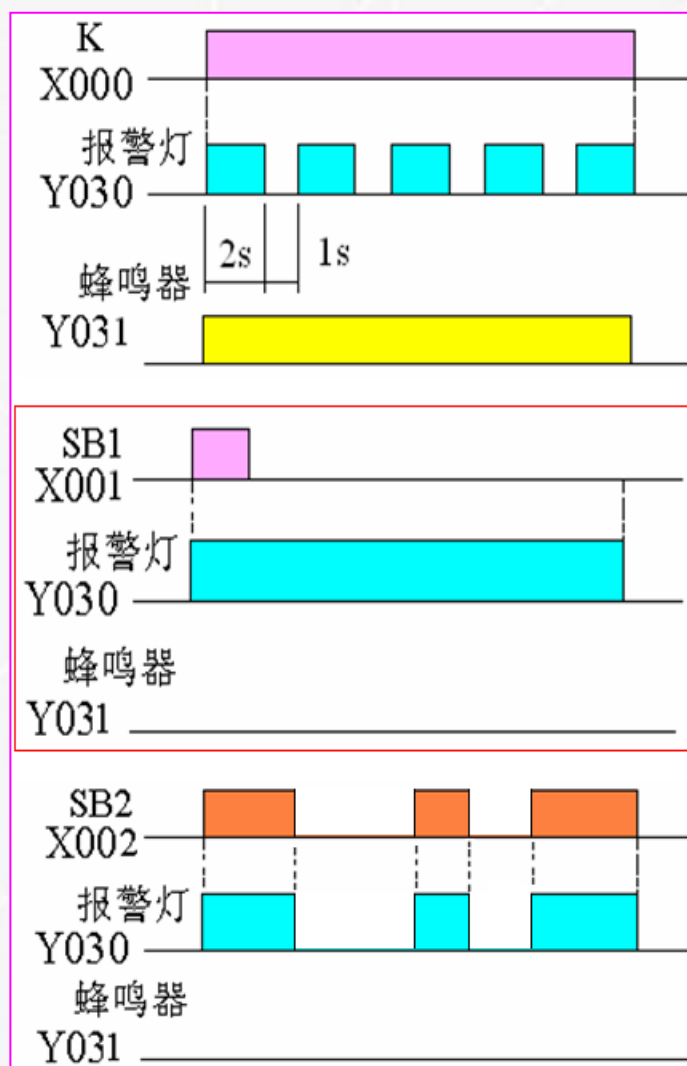
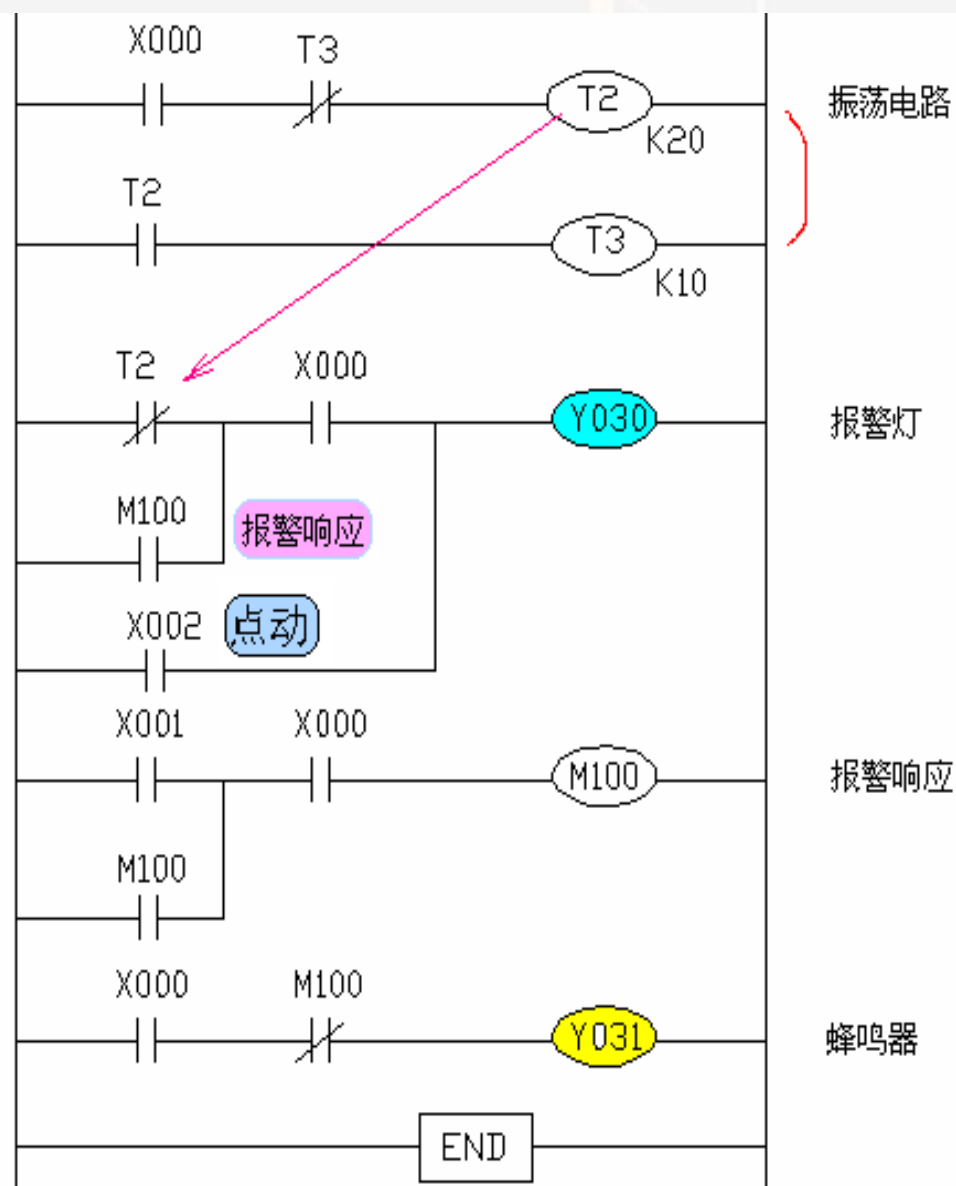


解:

## (1) 输入/输出接点分配表

输入装置	PLC输入端子号	输出装置	PLC输出端子号
K	X000	报警灯	Y030
SB1	X001	蜂鸣器	Y031
SB2	X002		







武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-长延时电路



# 【例1】长延时电路的PLC程序设计。

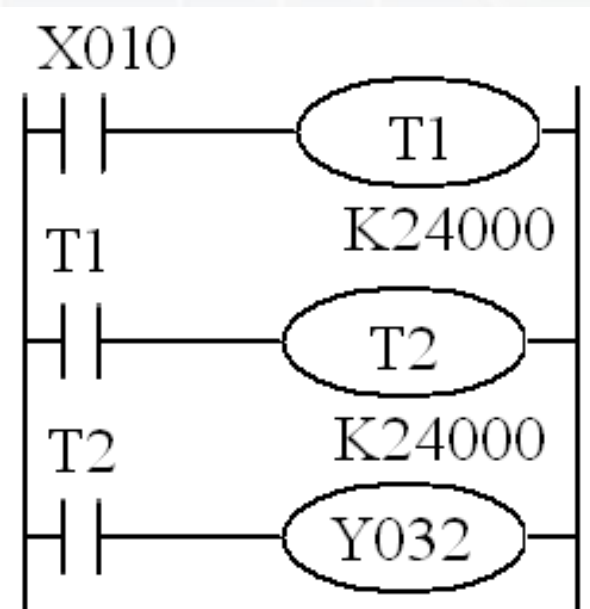
控制要求：

当输入信号X010接通1小时20分钟后，输出信号Y032接通。



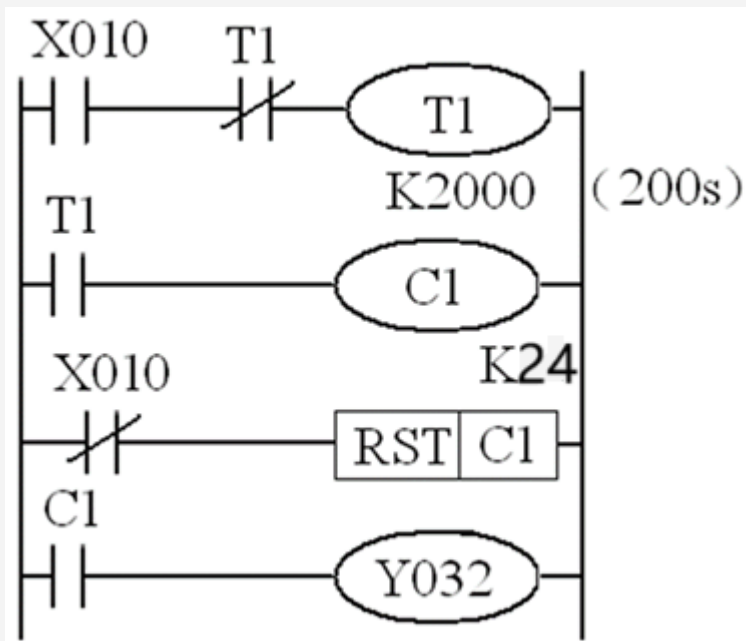
## 1) 两个定时器组合使用

- ✓ 选用普通计数器T1  
(100ms=0.1s) 时，最大延时  
时间为 $3276.7s < 1h$ 。1小时  
20分钟=4800s



## 2) 一个定时器和一个计数器组合

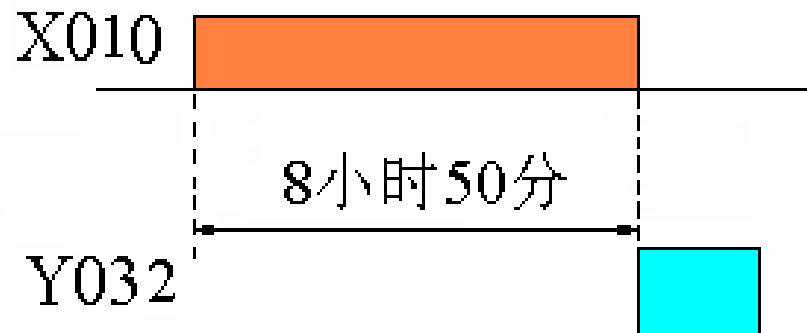
✓ 用一个定时器和一个计数器可构成一个4800s(200秒\*24) 的定时器，T1产生一个200秒脉冲，计数器C1计数到24次.



# 【例2】长延时电路的PLC程序设计。

**控制要求：**

**当输入信号X010接通8小时50分钟后，输出信号Y032接通。**



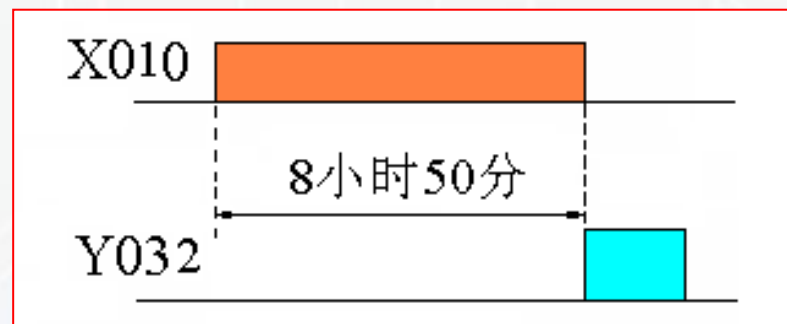
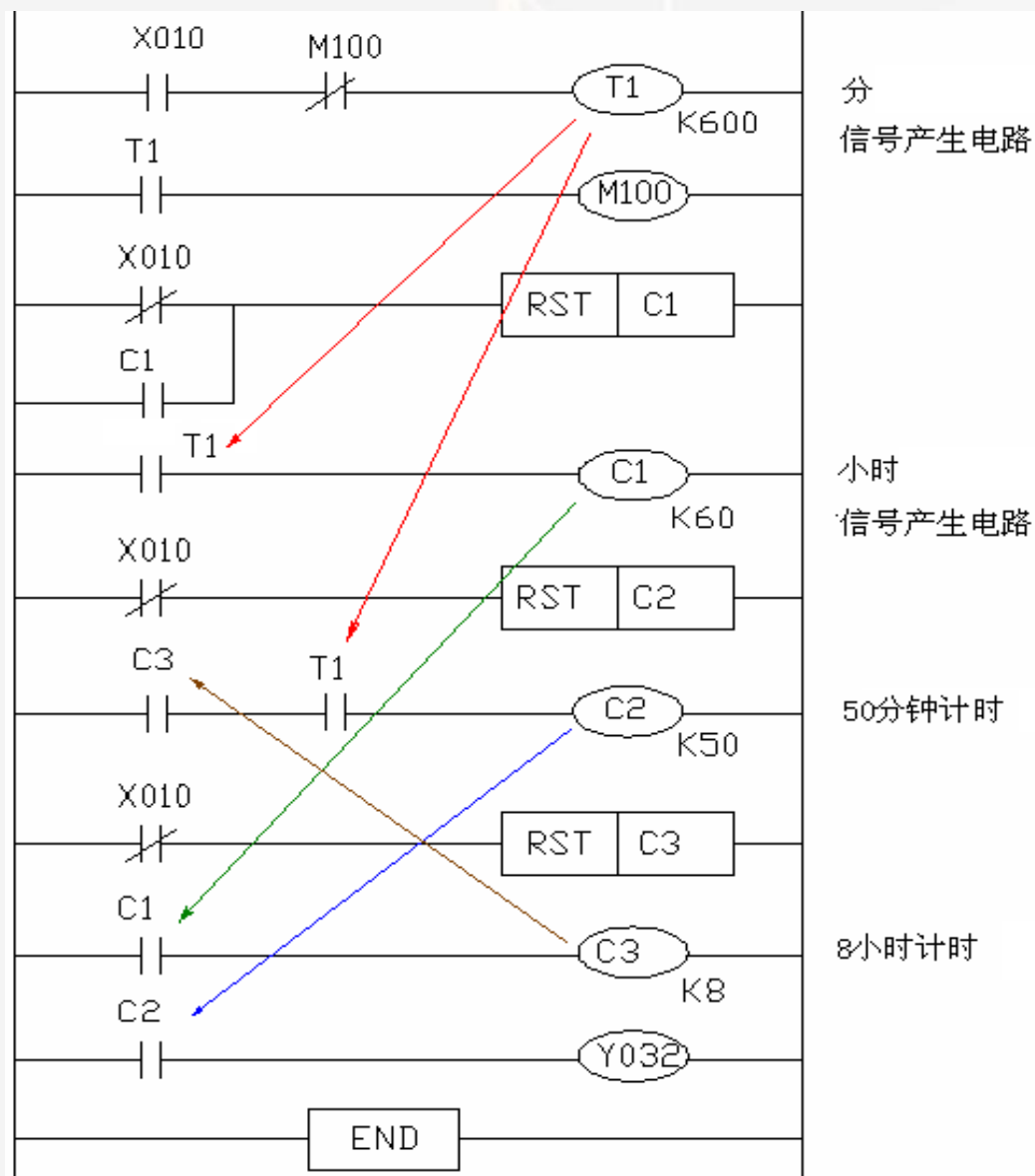
**解：(1) I/O分配表。**

输入	X010	输出	Y031
----	------	----	------

**(2) 控制逻辑梯形图**

- ✓ 选用T1 (100ms) 时，采用定时器与计数器联合编程的方法解决长时间定时的控制要求。
- ✓ 编程时，先**计小时**后**计分**。

## 长延时电路的梯形图

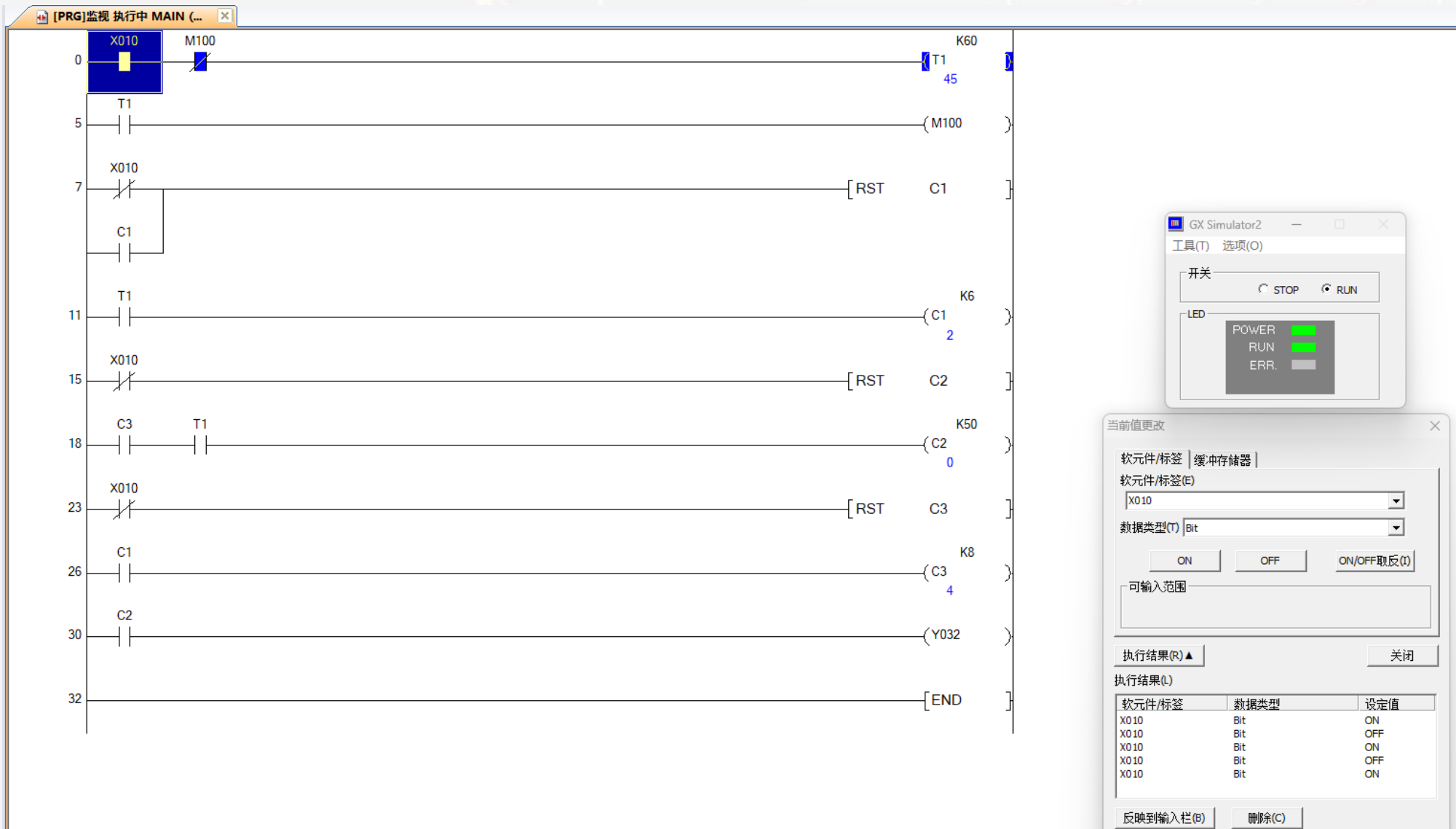




# 仿真运行



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



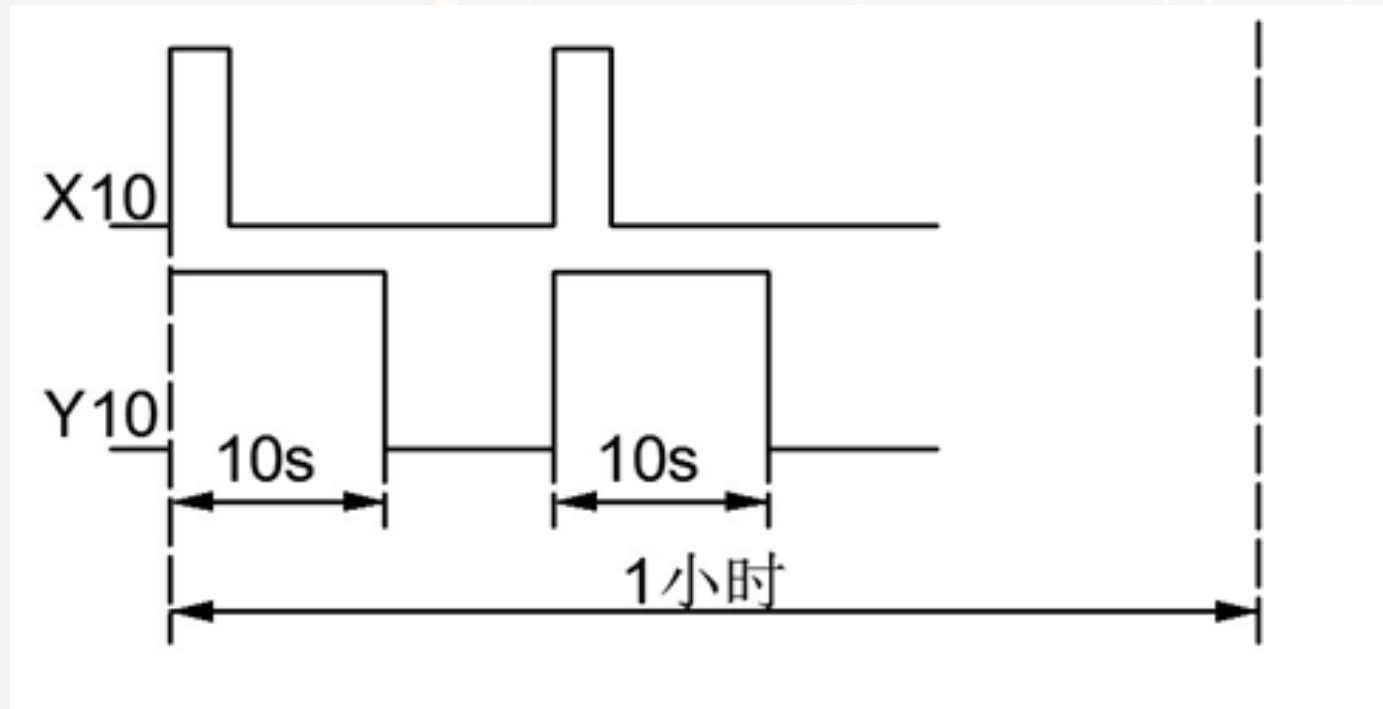


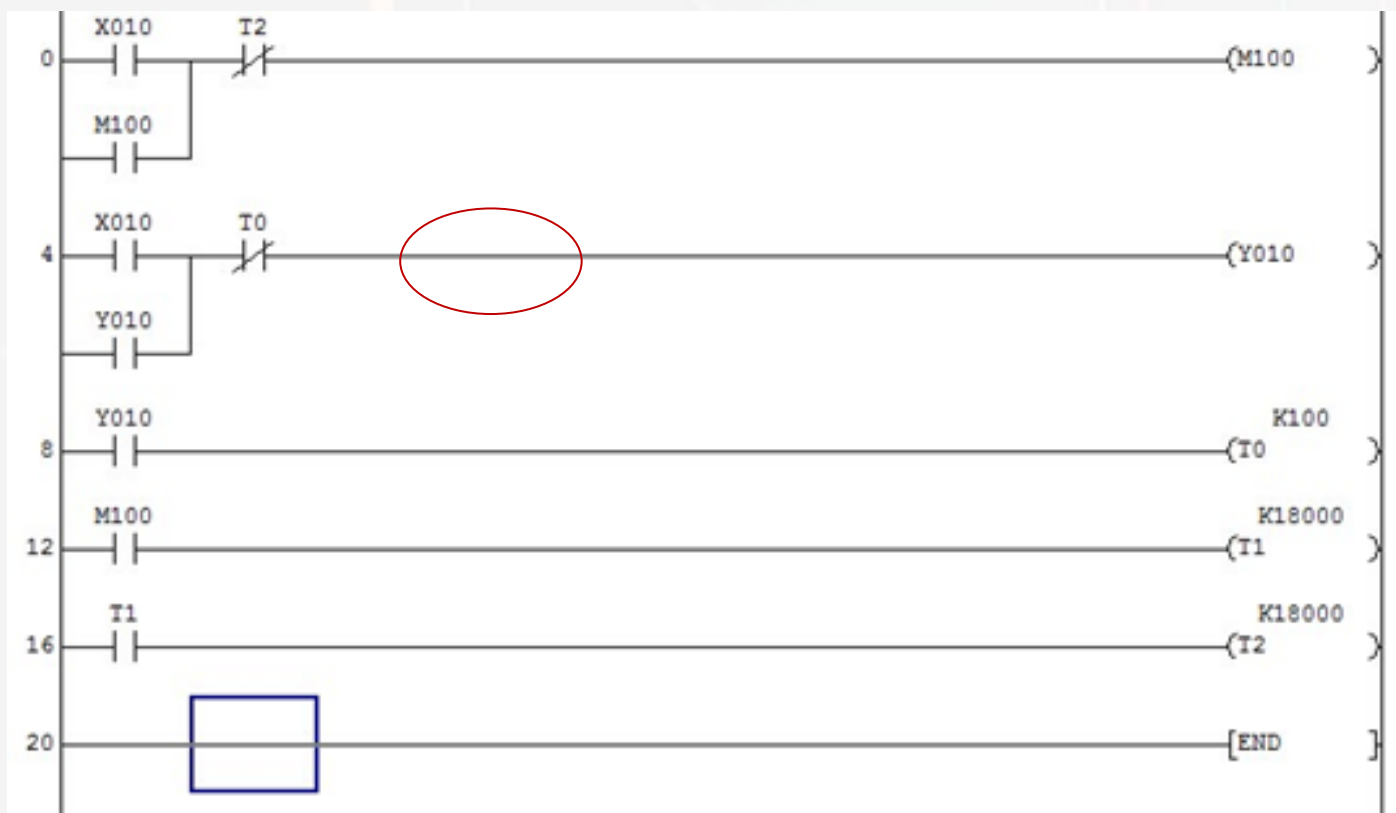
武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---

下图是输入、输出的时序波形图，试完成梯形图设计









武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-饲料箱盛料过少报警

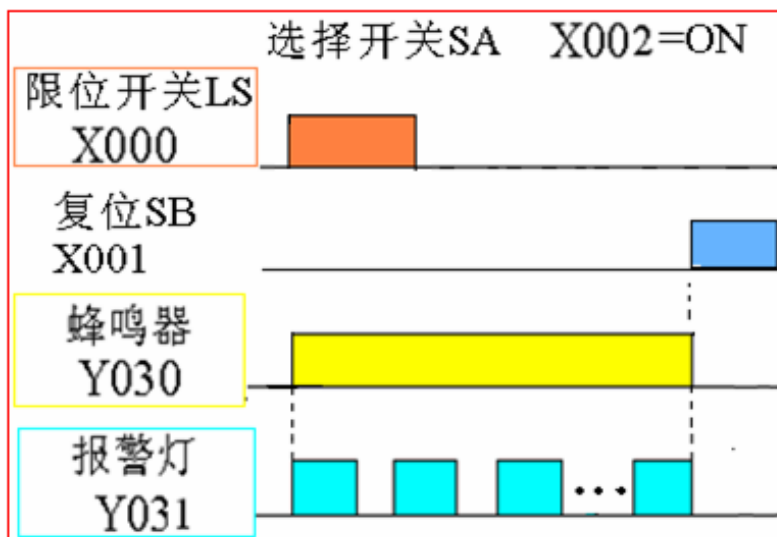
控制要求：

1. 方式选择 开关SA**断开**时，选择短时报警运行方式。

✓ 当盛料过少，限位开关LS变为ON后，蜂鸣器鸣叫，同时报警灯连续**闪烁20次**（亮1.5秒，灭2.5秒），此后蜂鸣器**停止鸣叫**，灯也**熄灭**。复位按钮SB可以使二者中止。

2. 方式选择开关SA**接通**时，选择连续报警运行方式。

✓ 当盛料过少，限位开关LS变为ON后，蜂鸣器鸣叫，同时灯开始闪烁。按下**复位按钮SB**时，二者中止。

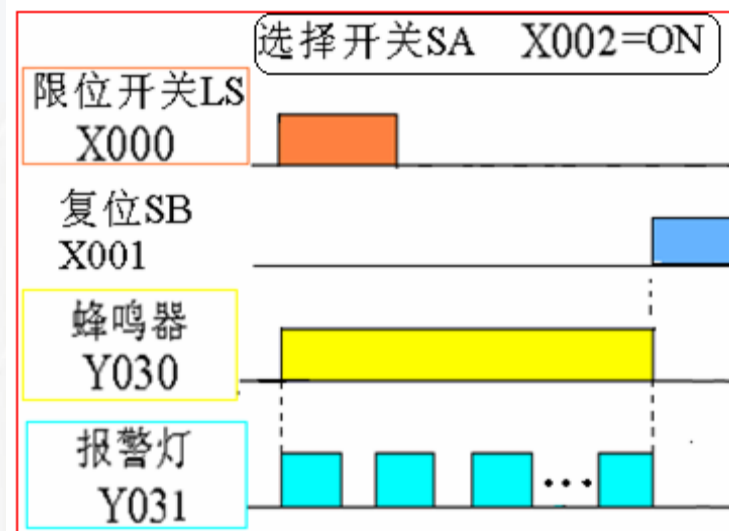
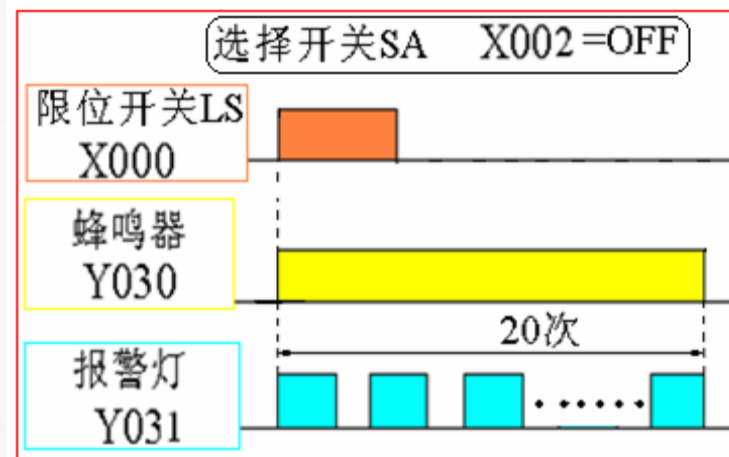
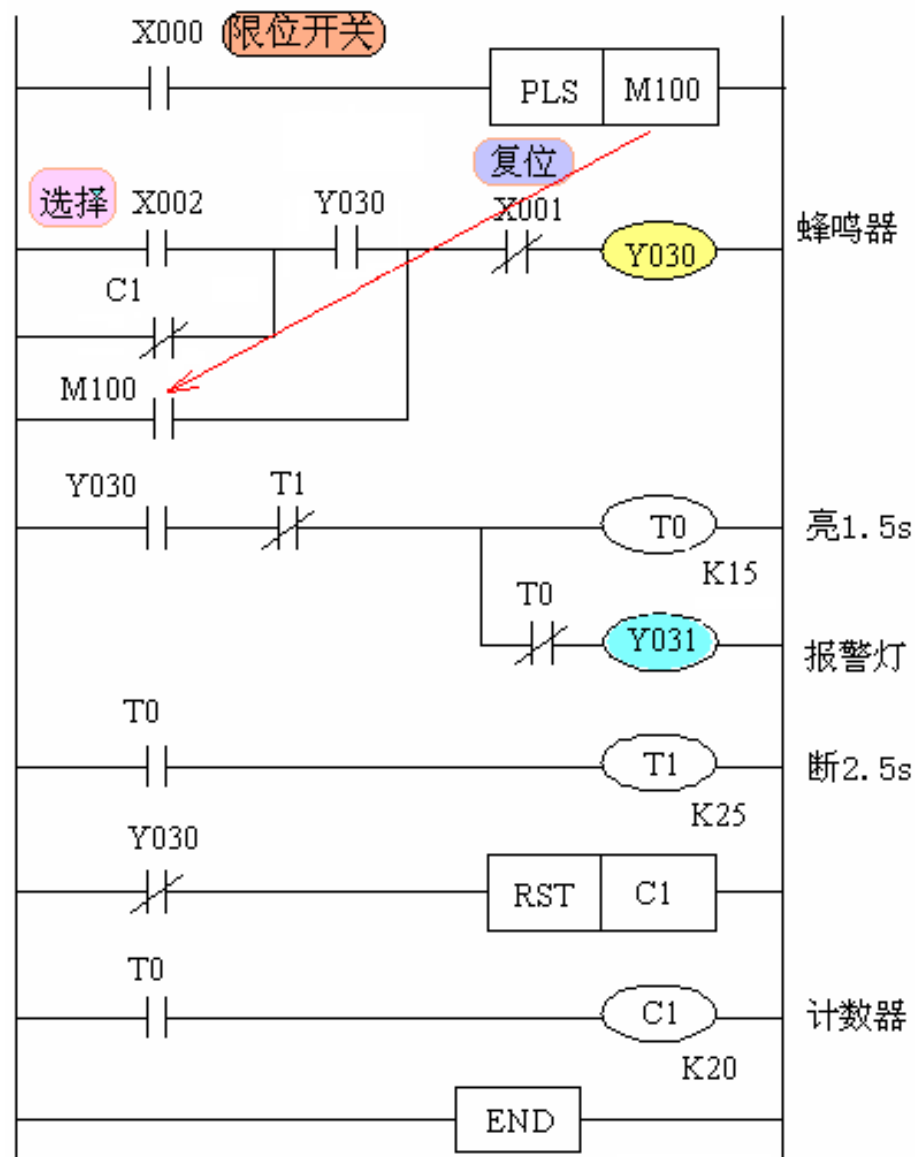


解：

## (1) 输入/输出接点分配表

输入装置	PLC输入端子号	输出装置	PLC输出端子号
限位开关LS	X000	蜂鸣器HA	Y030
复位开关SB	X001	报警灯HL	Y031
选择开关SA	X002		







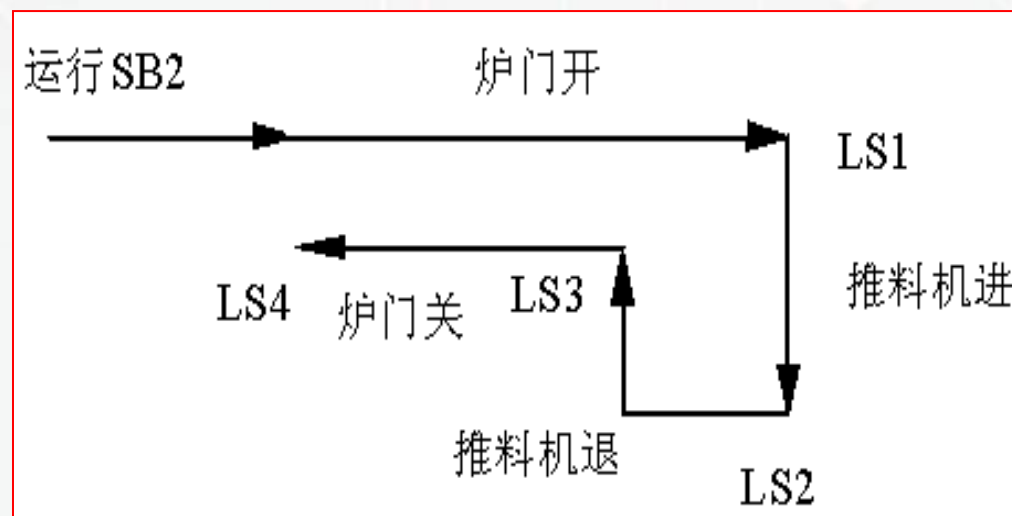
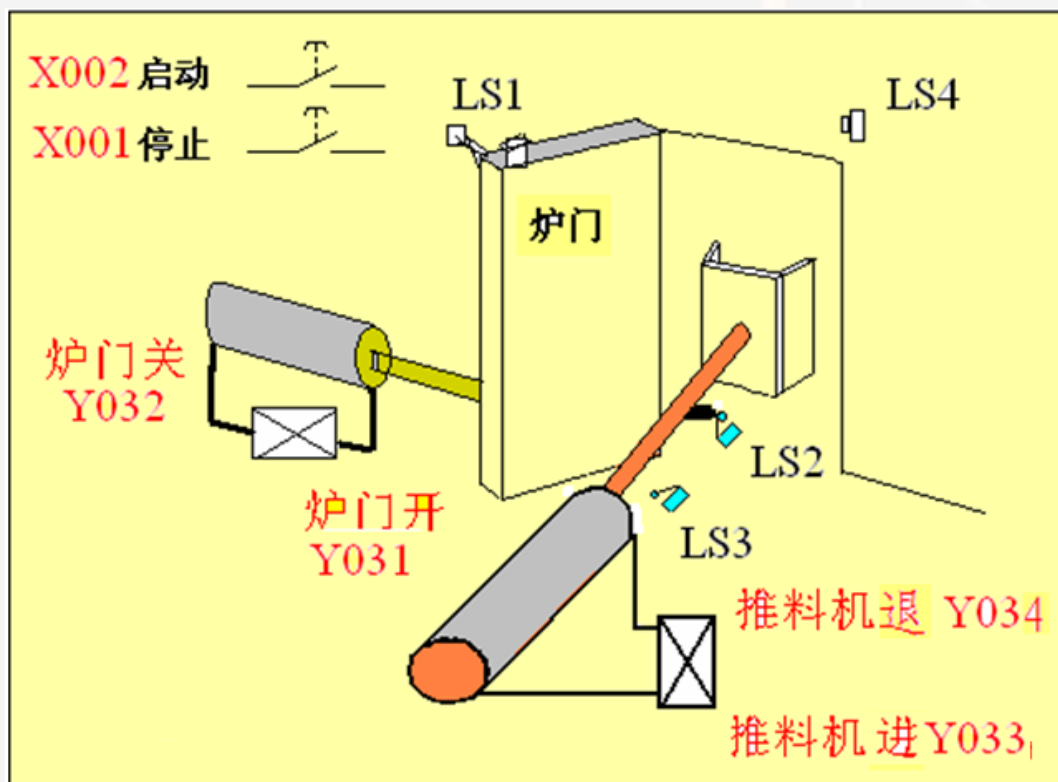
武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 编程-加热炉推料机自动上料

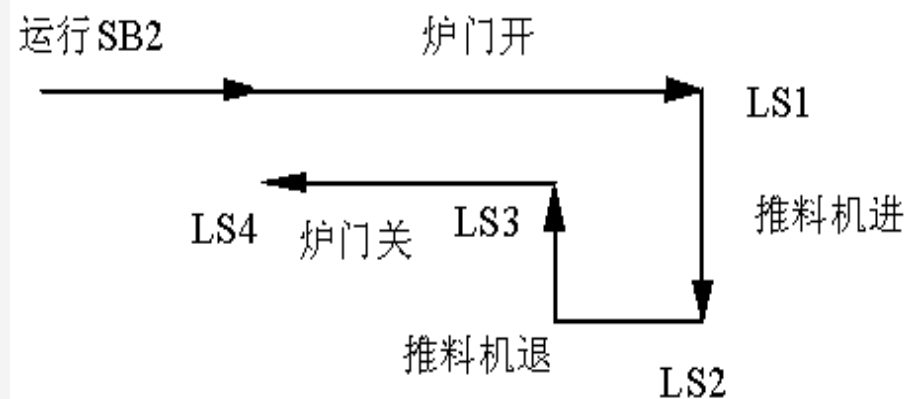
# 【例】加热炉推料机自动上料控制电路的PLC程序设计。



武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY



## 控制要求：

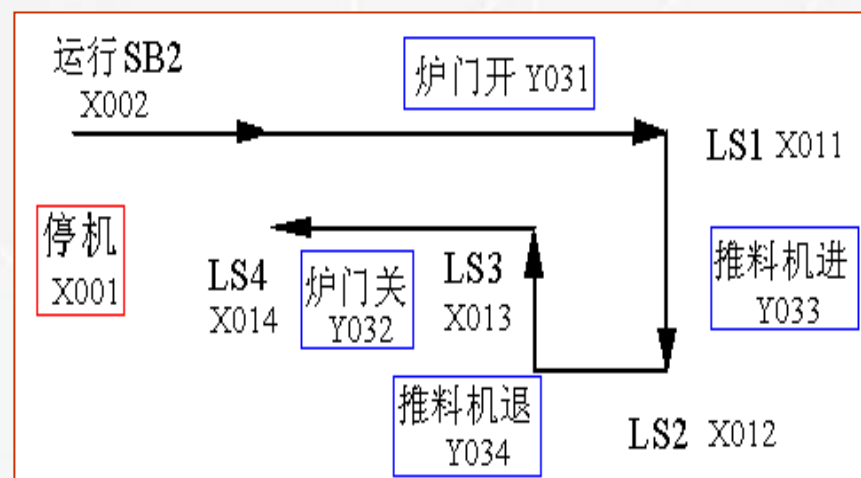
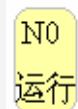


- ✓ 推料机在**原点**，加热**炉门关闭**，行程开关**LS4被压住**；
- ✓ 运行按钮**SB2被按下**后，驱动**炉门开启**；
- ✓ 开启到位，行程开关**LS1被压住**，然后推料机往炉门**推料**；
- ✓ 推到位，**压住行程开关LS2**后，推料机**后退**；
- ✓ 后退到位，**压住行程开关LS3**，然后炉门开始**关闭**；
- ✓ 炉门关闭后，行程开关**LS4被压住**，整个**工作流程结束**。
- ✓ 再按下SB2后，重复上述过程。按下停机按钮**SB1**后，马上停止。



## 解：（1）输入/输出接点分配表

输入装置	PLC输入 端子号	输出装置	PLC输出 端子号
停机SB1	X001	炉门开KM1	Y031
运行SB2	X002	炉门关KM2	Y032
门开LS1	X011	推料机进KM3	Y033
推料机进到位 LS2	X012	推料机退KM4	Y034
推料机退到位 LS3	X013		
门关，回原位 LS4	X014		





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

# 测试

---

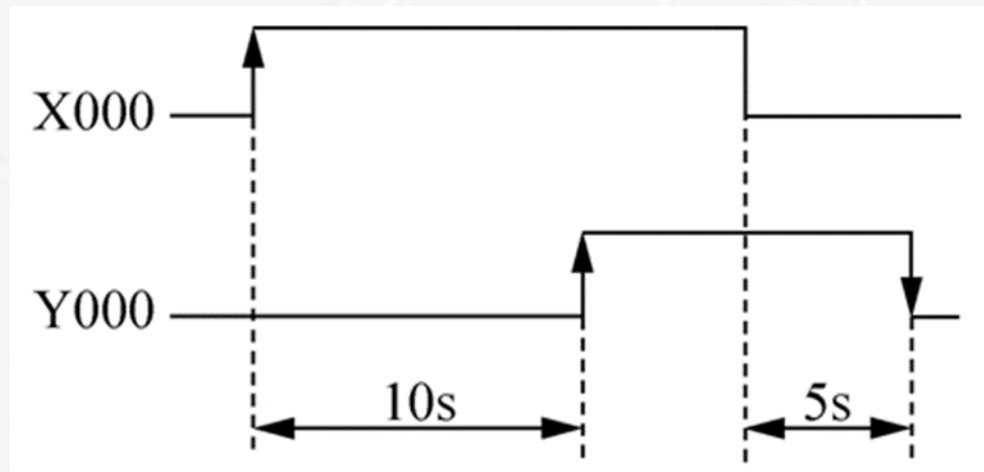
用PLC的内部定时器设计一个延时电路，图示是其时序图。

功能要求：

- (1) 当X000接通时，Y000延时10s后才接通。
- (2) 当X000断开时，Y000延时5s后才断开。

编程要求：

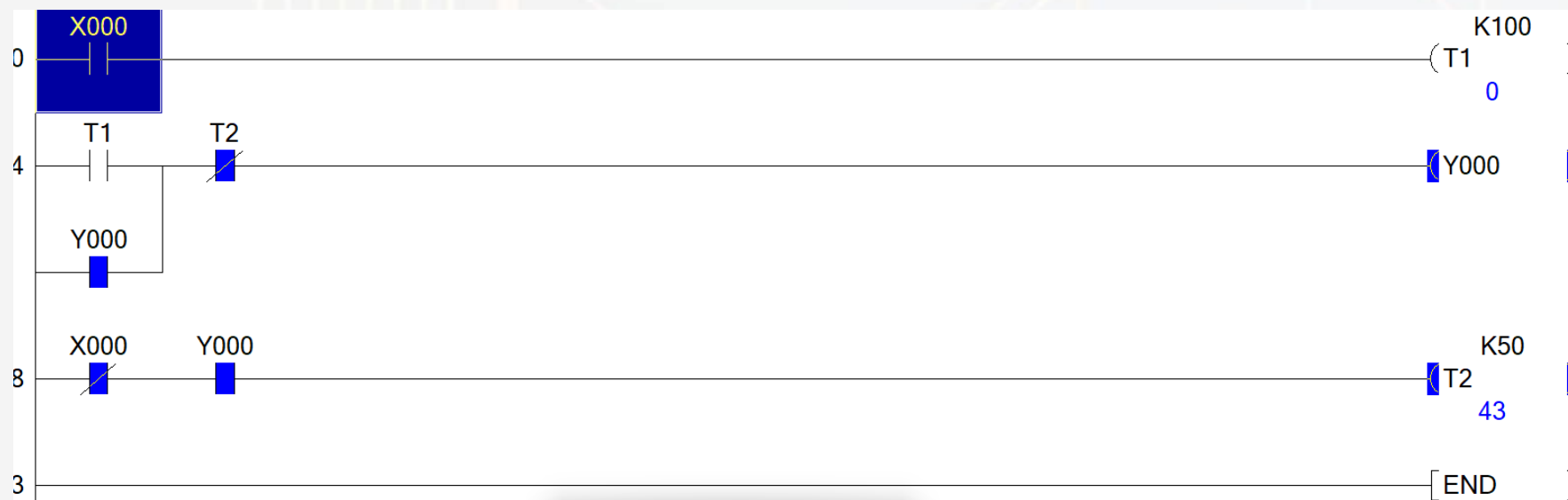
- (1) 列出输入/输出的端口分配表。
- (2) 画出梯形图和接线图。
- (3) 写出指令程序。







步序	指 令	步序	指 令
0	LD X000	7	OUT Y000
1	OUT T1 K100	8	LDI X000
4	LD T1	9	AND Y000
5	OR Y000	10	OUT T2 K50
6	ANI T2		END





武汉理工大学  
WUHAN UNIVERSITY OF TECHNOLOGY

谢谢观看