

PLC与变频应用技术

主讲：陈丽娟

温故知新

基本指令

1、LD、LDI——取指令

LD：“取指令”，使常开触点与左母线相连。

LDI：“取反指令”，使常闭触点与左母线相连。

操作元件：X、Y、M、S、T、C等

2、OUT——驱动线圈输出指令

将程序段的逻辑运算结果去驱动一个指定的线圈。

可驱动对象：Y、M、C、T、S

3、触点串联——用于程序中串联单个触点

AND：串常开触点

ANI：串常闭触点

4、触点并联——用于程序中并联单个触点

OR：串常开触点

ORI：串常闭触点

5、SET：置位（置1）指令——线圈接通并保持。

操作元件：Y、M、S

6、RST：复位（清0）指令——线圈断开。

操作元件：Y、M、S、T、C、D、V、Z等

操作元件：
X、Y、M、
S、T、C

温故知新

基本步骤

- 1、分析题目
- 2、I/O分配：
- 3、I/O端口接线：
- 4、编程
- 5、调试、运行

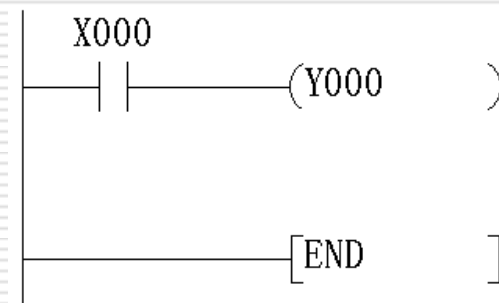
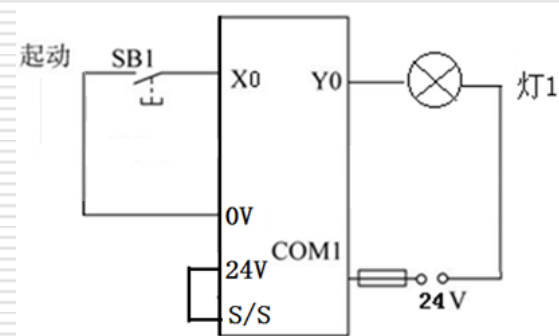
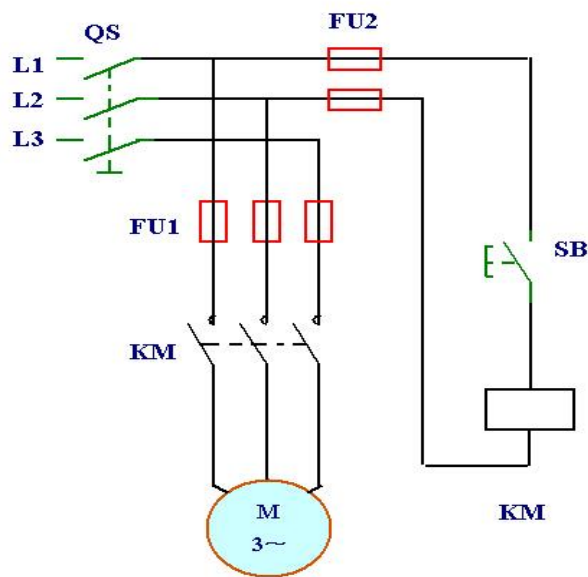
PLC编程的基本规则

- 1、触点不能右母线相连。
- 2、线圈只能与右母线相连，不能与左母线相连。
- 3、画梯形图时，右母线可以省略
- 4、线圈可以并联，不能串联。
- 5、避免双线圈输出。

温故知新

点动控制

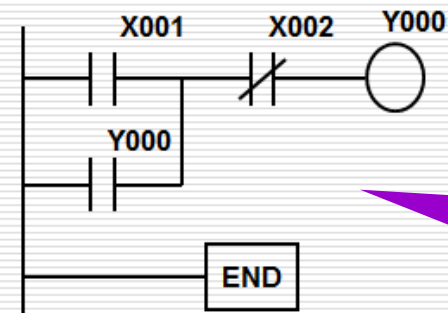
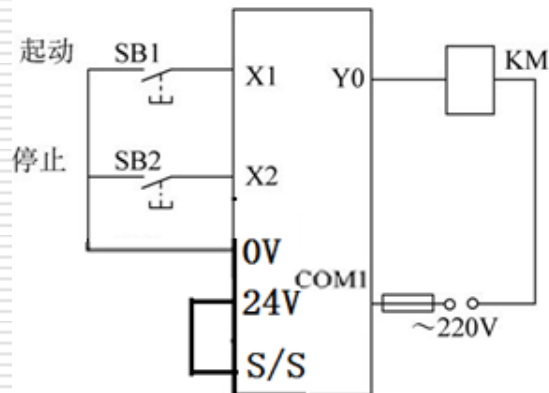
输入元件	输入继电器	输出元件	输出继电器
点动按钮SB1	X0	交流接触器KM	Y0



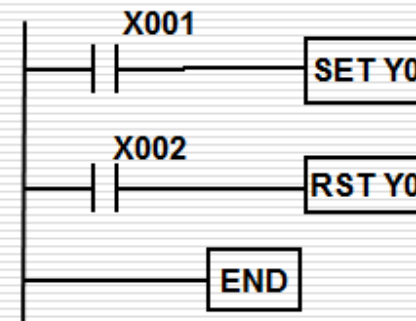
温故知新

自锁控制

输入元件	输入继电器	输出元件	输出继电器
启动按钮SB1	X1	交流接触器KM	Y0
停止按钮SB2	X2		



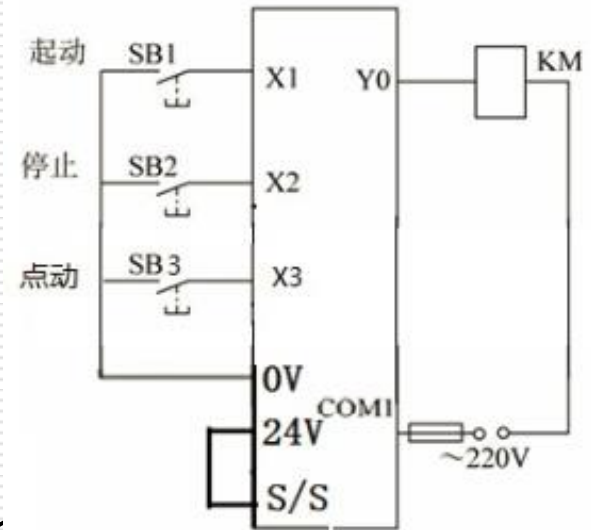
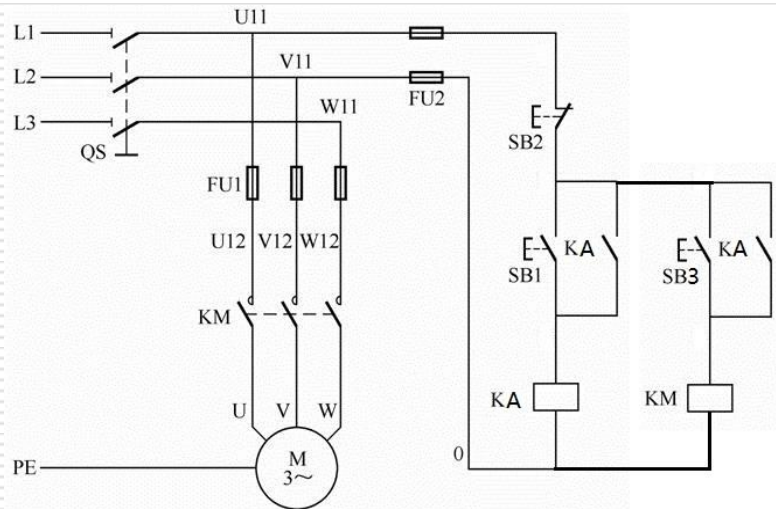
触点
串并联



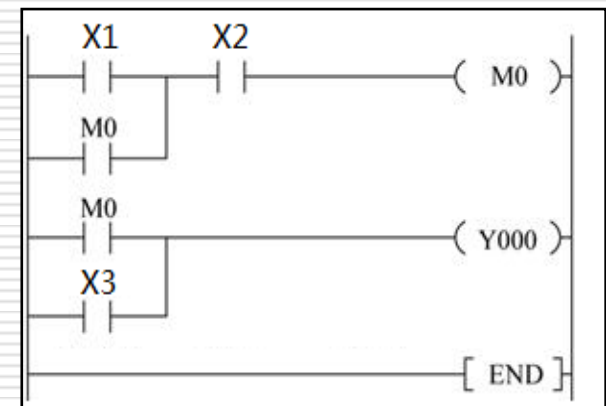
置位
复位

温故知新

混合控制



输入元件	输入继电器	输出元件	输出继电器
启动按钮SB1	X1	交流接触器KM	Y0
停止按钮SB2	X2		
点动按钮SB3	X3		



温故知新

辅助继电器 M

与中间继电器相似，用于中间状态变换、存储或中间信号变换等。

特点：辅助继电器的通断只能由内部程序驱动，并且不能直接驱动外部负载。

- 类型：
- (1) 通用型：**M0 ~ M499**
 - (2) 掉电保持型：**M500 ~ M3071**
(具有记忆功能)
 - (3) 特殊辅助继电器：**M8000~M8511**

温故知新

特殊辅助继电器

1、**触点型**：用户只可使用其触点，线圈由PLC自动驱动。

M8000：运行监视器（在PLC运行中接通）。

M8002：初始脉冲（仅在运行开始时瞬间接通）

M8011：产生10ms时钟脉冲。

M8012：产生100ms时钟脉冲。

M8013：产生1s时钟脉冲。

M8014：产生10min时钟脉冲。

在自锁控制中加入特殊辅助继电器，使LED以1s为周期闪烁

2、**线圈型**：由用户程序驱动线圈后PLC执行特定动作。

M8033：其线圈得电，PLC停止时保持输出映像寄存器和数据寄存器的内容。

M8034：其线圈得电，则将PLC的输出全部禁止。

M8039：其线圈得电，则PLC按D8039中制定的扫描时间工作。

项目一：运料小车自动往返控制

任务目标

- ❑任务1：三相异步电动机单向运行控制
- ❑任务2：三相异步电动机正反转控制
- ❑任务3：多台电机的PLC控制
- ❑任务4：电动机的定时运行控制
- ❑任务5：运料小车循环次数的控制

知识、能力目标

- ❑ 1、掌握PLC控制系统设计的基本步骤
- ❑ 2、掌握PLC控制系统硬件接线的方法
- ❑ 3、熟练操作编程软件
- ❑ 4、熟悉FX3U系列PLC的软元件使用方法
- ❑ 5、掌握三菱FX3U系列PLC的基本逻辑指令系统
- ❑ 6、掌握梯形图程序设计的编程规则、技巧和方法

三菱FX_{3u}系列PLC的基本逻辑指令系统

一、取指令 LD、LDI

二、触点串联指令 AND、ANI

三、触点并联指令 OR、ORI

四、输出指令 OUT

五、电路块串并联指令ANB、ORB

六、堆栈指令MPS、MRD、MPP

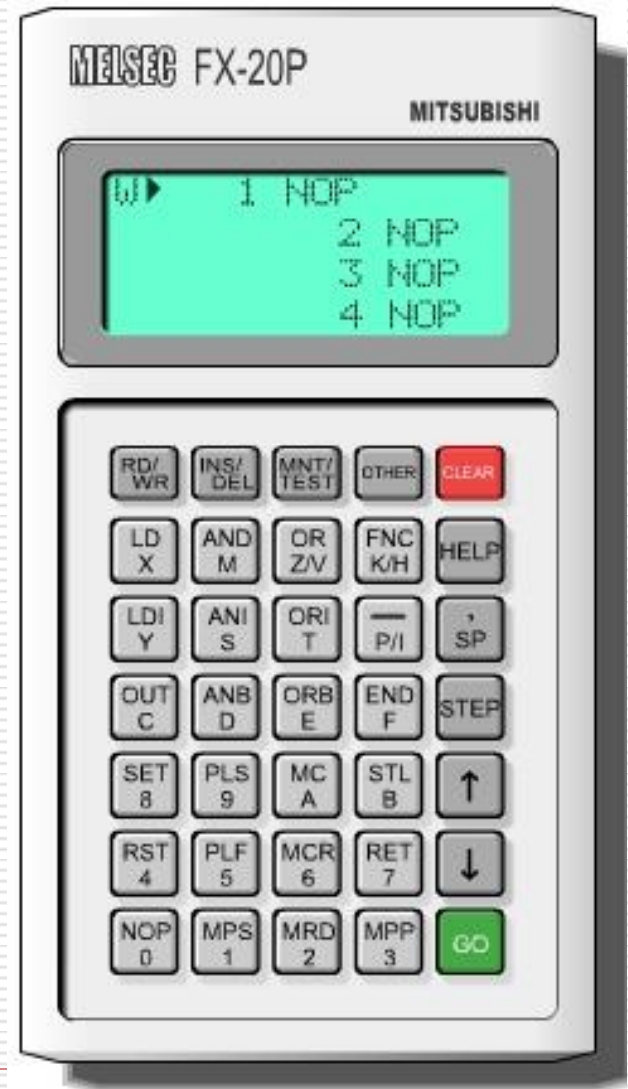
七、主控指令MC、MCR

八、置位、复位指令SET、RST

九、脉冲输出指令PLS、PLF

十、边沿检测指令**P、**F

十一、空操作和结束指令NOP、END



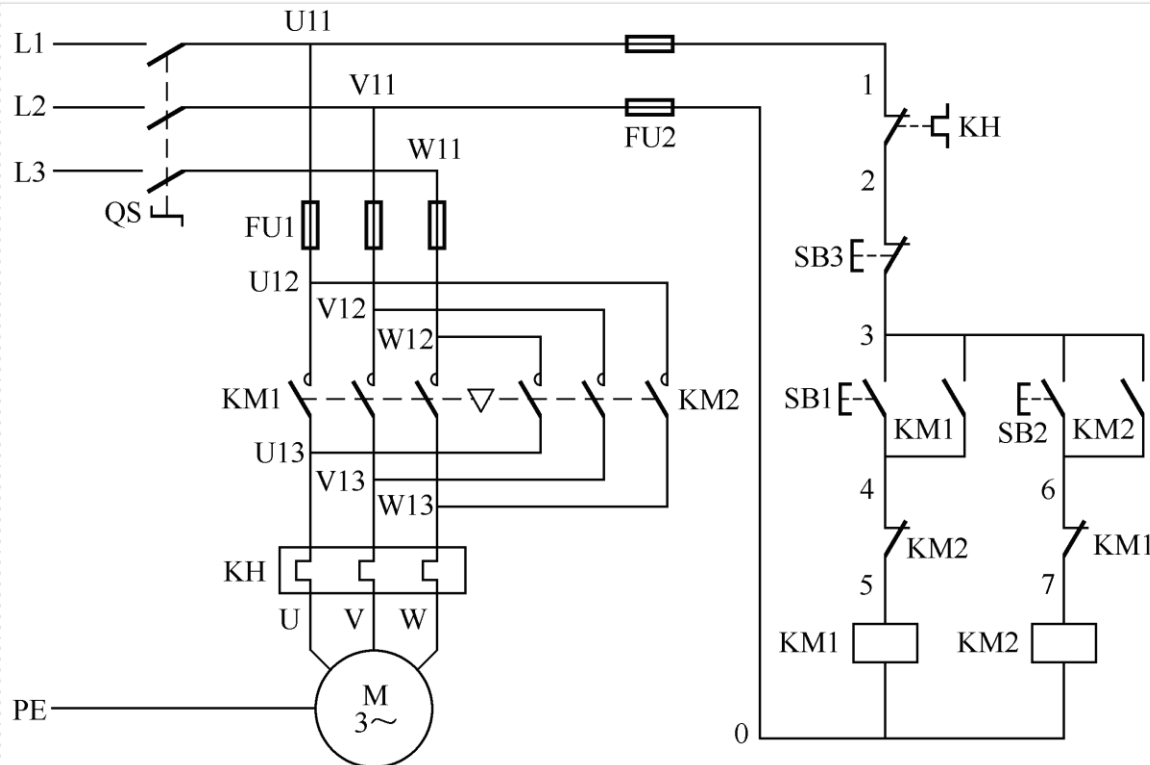
任务2：三相异步电动机正反转控制

- ❑ 某系统，含有多个控制对象，这些对象不能同时运行，每个时刻之间有一个对象运行。
- ❑ 这时，各控制对象之间需要互锁（联锁）。

学习目标

- ❑ 掌握互锁程序的写法
- ❑ 掌握硬联锁、软连锁的使用
- ❑ 掌握梯形图优化的基本方法
- ❑ 了解电路块串/并联指令
- ❑ 了解多重输出指令

任务2：三相异步电动机正反转控制



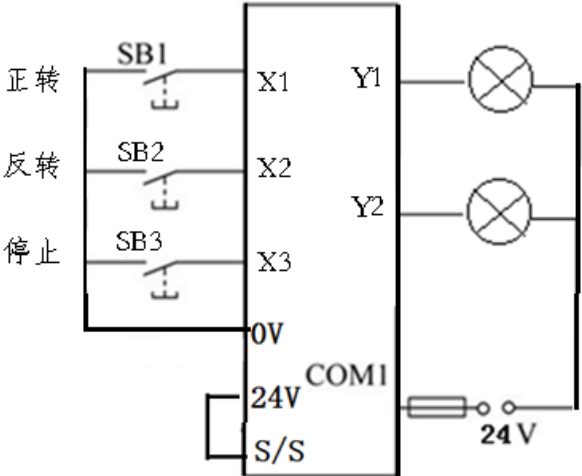
传统的接触器联锁正反转电路

任务2：三相异步电动机正反转控制

1、I/O分配：

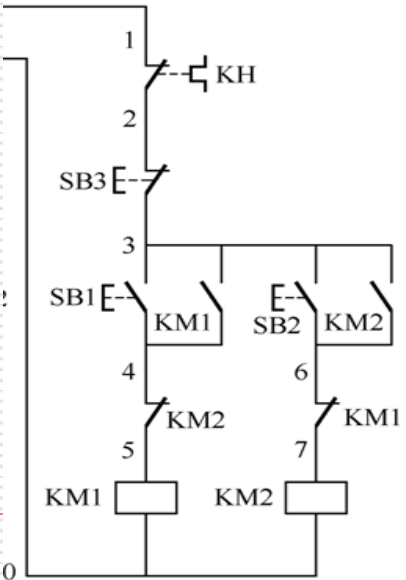
输入元件	输入继电器	输出元件	输出继电器
正转按钮SB1	X1	正转接触器KM1	Y1
反转启动SB2	X2	反转接触器KM2	Y2
停止按钮SB3	X3		

2、I/O接线：



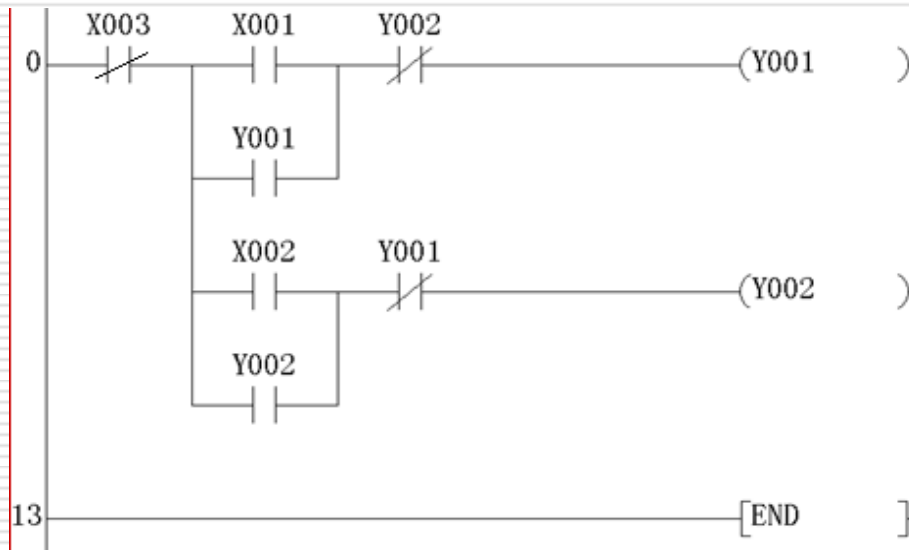
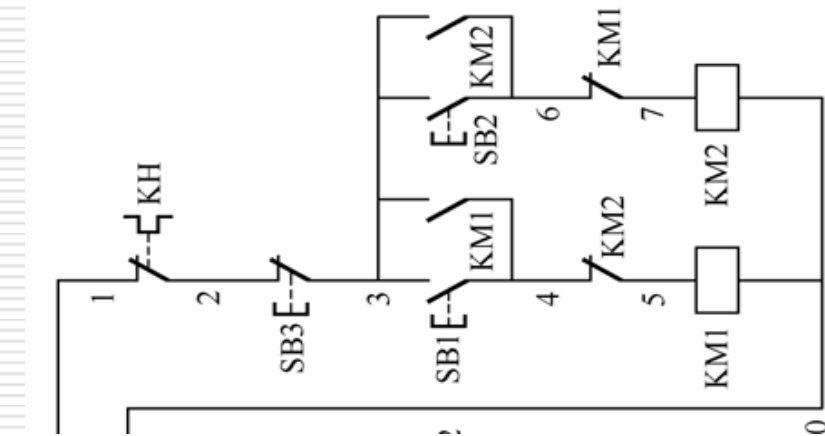
3、程序设计：

用“翻译法”

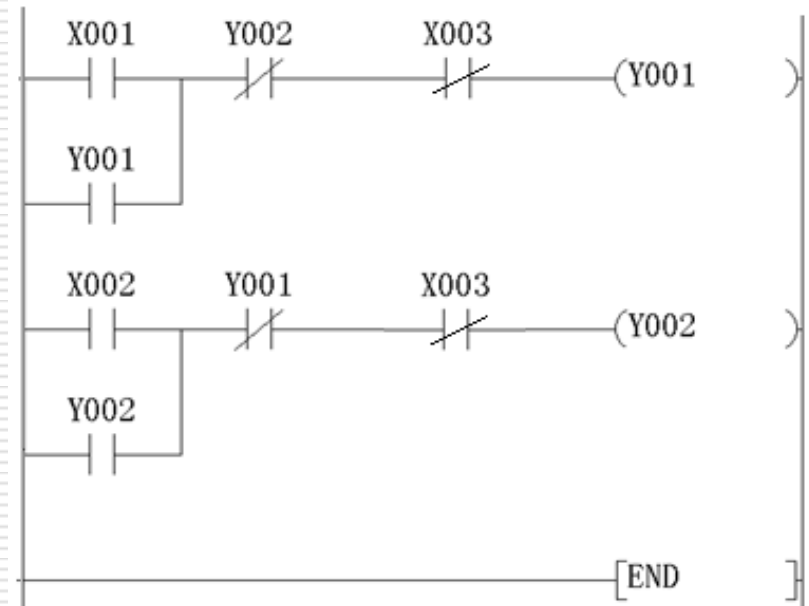


任务2：三相异步电动机正反转控制

2、程序设计：



□ 调整后：

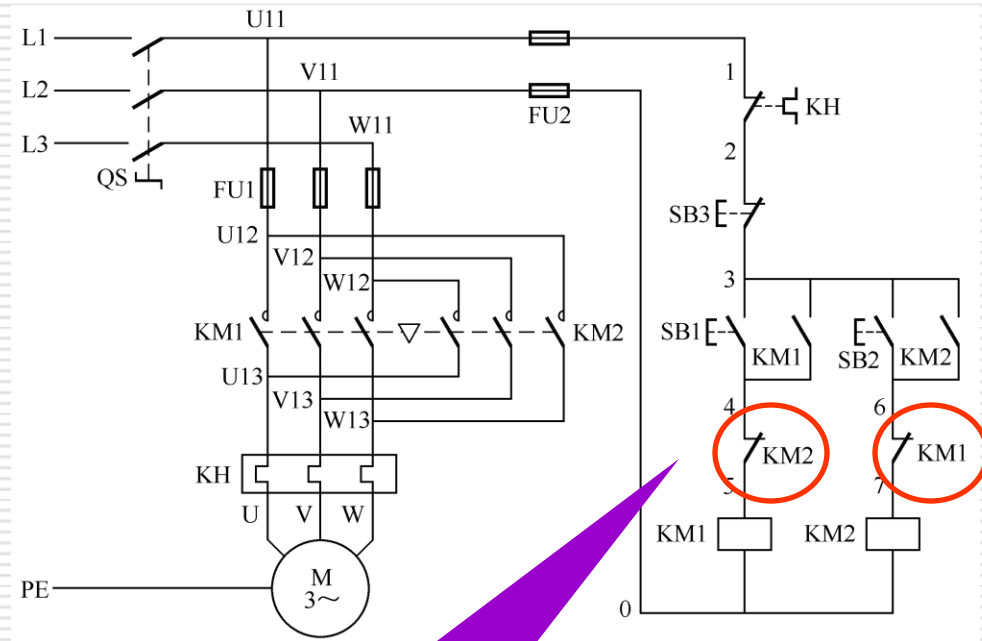
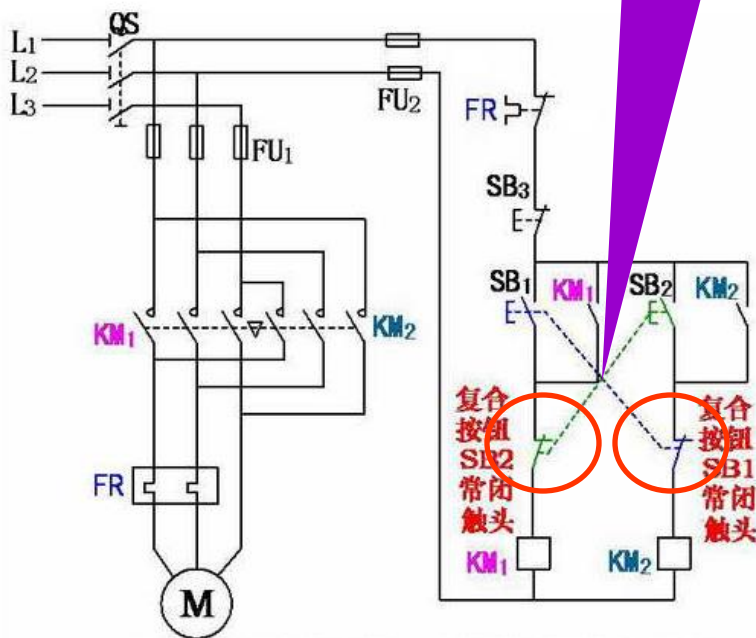


任务2：三相异步电动机正反转控制

正反转控制有三种形式

- 正反停：按钮联锁（极少使用）
- 正停反：接触器联锁
- 正反停：双重联锁

按钮联锁

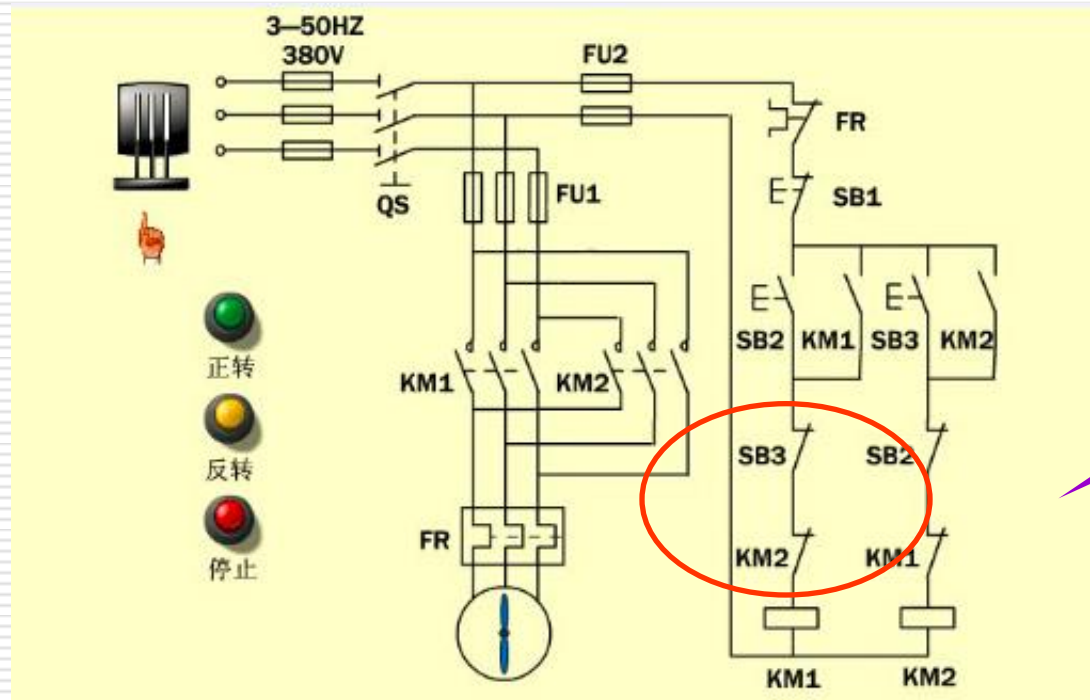


接触器联锁

任务2：三相异步电动机正反转控制

正反转控制有三种形式

- ❑ 正反停：按钮联锁（极少使用）
- ❑ 正停反：接触器联锁
- ❑ 正反停：双重联锁

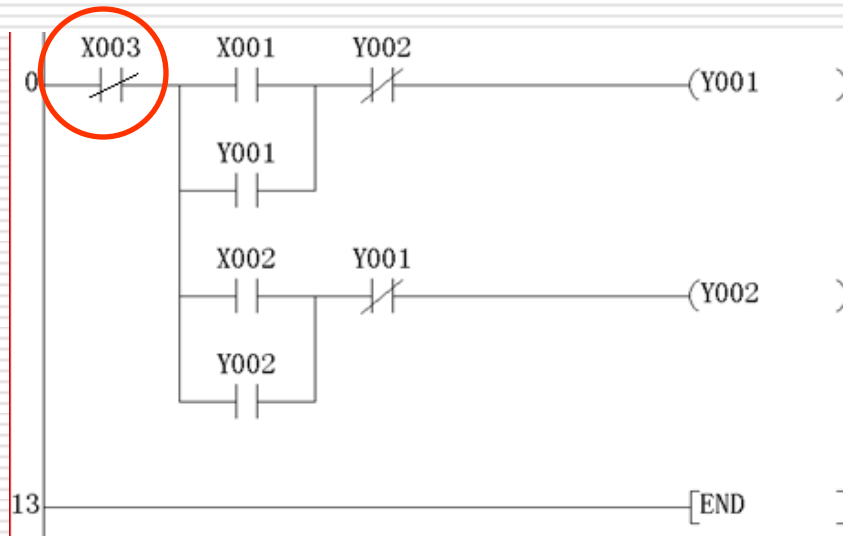
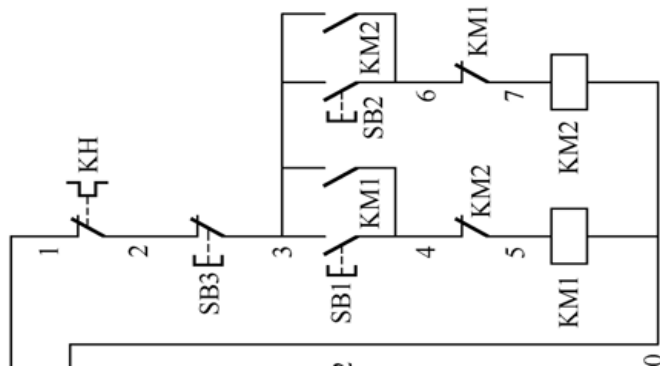


双重联锁

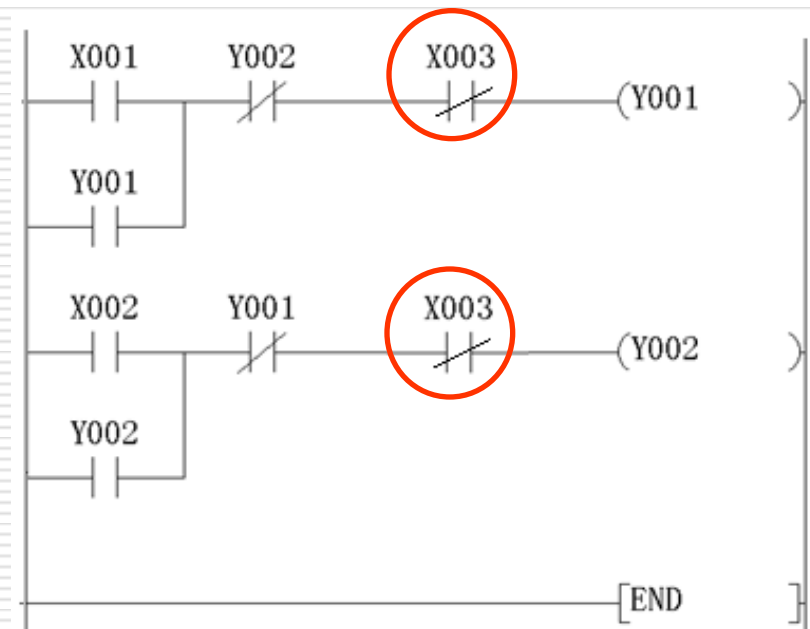
□ 梯形图的优化

任务2：三相异步电动机正反转控制

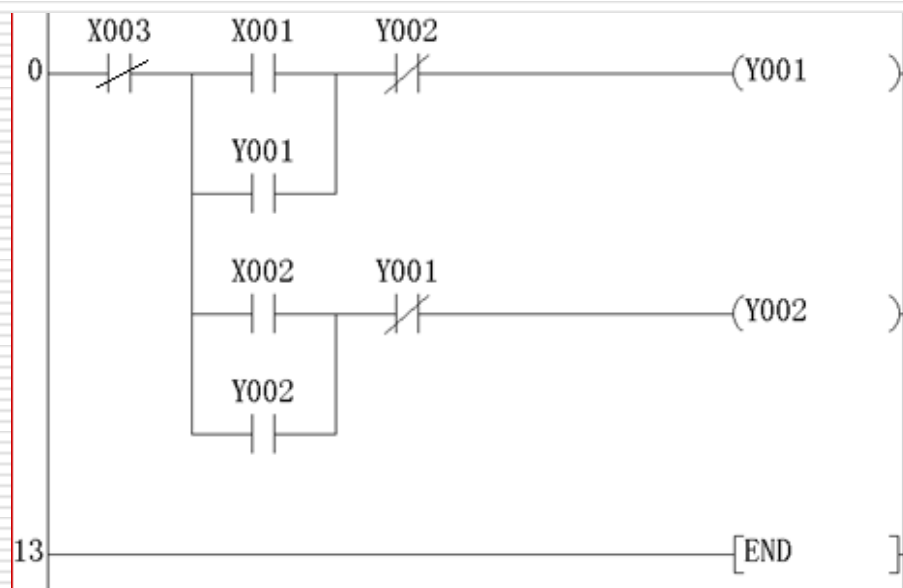
2、程序设计：



□ 调整后：



为什么要调整？？



调整前指令：

0	LD	X003
1	MPS	
2	LD	X001
3	OR	Y001
4	ANB	
5	ANI	Y002
6	OUT	Y001
7	MPP	
8	LD	X002
9	OR	Y002
10	ANB	
11	ANI	Y001
12	OUT	Y002
13	END	

进栈

块串联

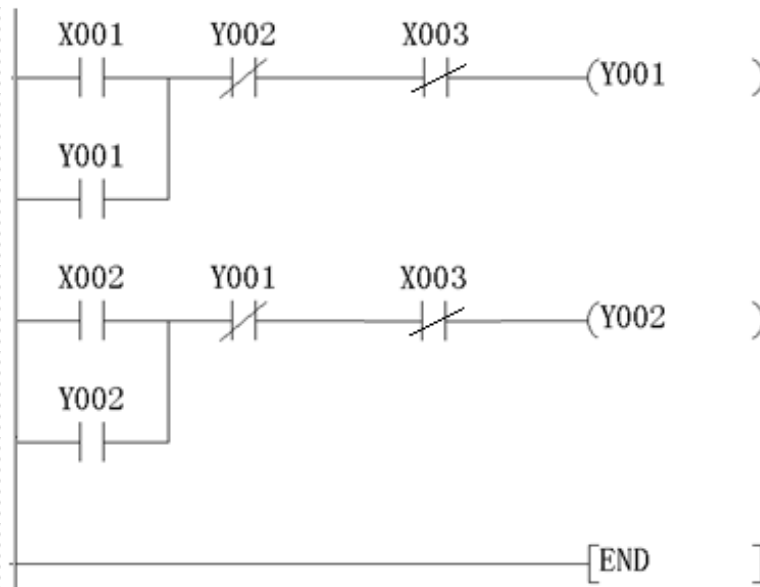
出栈

块串联

□ 调整后：

调整前指令：

* 优化原则：触点复用



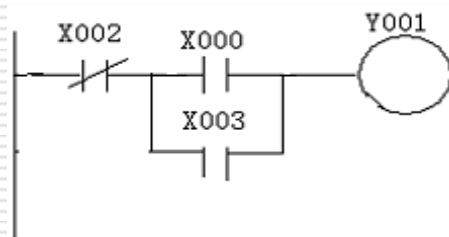
0	LD	X001
1	OR	Y001
2	ANI	Y002
3	ANI	X003
4	OUT	Y001
5	LD	X002
6	OR	Y002
7	ANI	Y001
8	ANI	X003
9	OUT	Y002
10	END	

*知识链接：电路块的串并联指令（略）

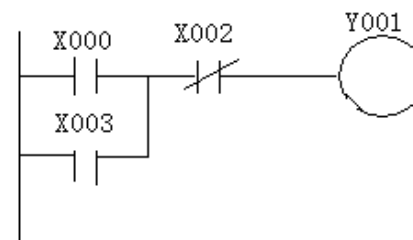
1、ANB指令

* 优化原则：左重右轻

- 用于“并联电路块”的串联连接
- 由两个或以上的触点并联而成的电路称为“并联电路块”
- “并联电路块”与前面的电路串联时，起点要用**LD/LDI**，终点要用“**ANB**”。



LDI	X002
LD	X000
OR	X003
ANB	
OUT	Y001



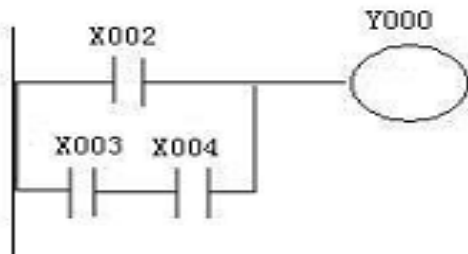
LD	X000
OR	X003
ANI	X002
OUT	Y001

*知识链接：电路块的串并联指令（略）

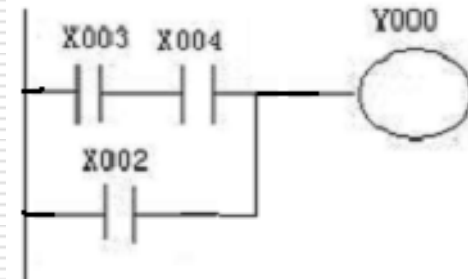
2、ORB指令

* 优化原则：上重下轻

- 用于“串联电路块”的并联连接
- 由两个或以上的触点串联而成的电路称为“串联电路块”
- 在并联“串联电路块”时，起点要用**LD/LDI**，终点要用**“ORB”**。



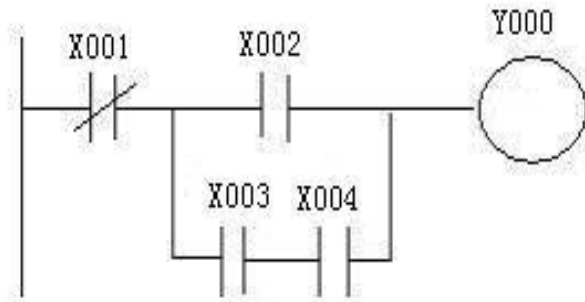
```
LD    X002
LD    X003
AND   X004
ORB
OUT   Y000
```



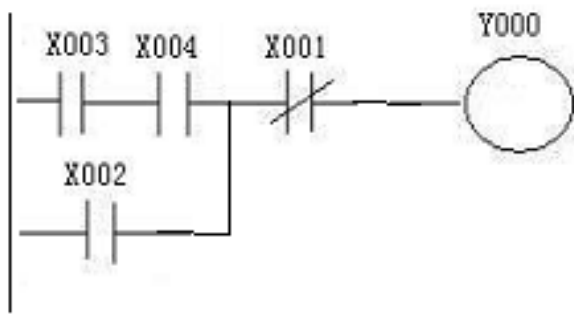
```
LD    X003
AND   X004
OR    X002
OUT   Y000
```

*知识链接：电路块的串并联指令（略）

练习：



```
LDI    X001
LD      X002
LD      X003
AND     X004
ORB
ANB
OUT     Y000
```



```
LD      X003
AND     X004
OR      X002
ANI     X001
OUT     Y000
```

* 上重下轻，左重右轻

*知识链接：电路块的串并联指令（略）

注：

- **1、ORB、ANB 不带操作数**
- **2、ORB/ANB有两种编程格式：**
 - （1）分散使用（不限次数）**
 - （2）集中使用（不能超过8次）**

*知识链接：栈操作指令 MPS MRD MPP

1. 进栈指令MPS

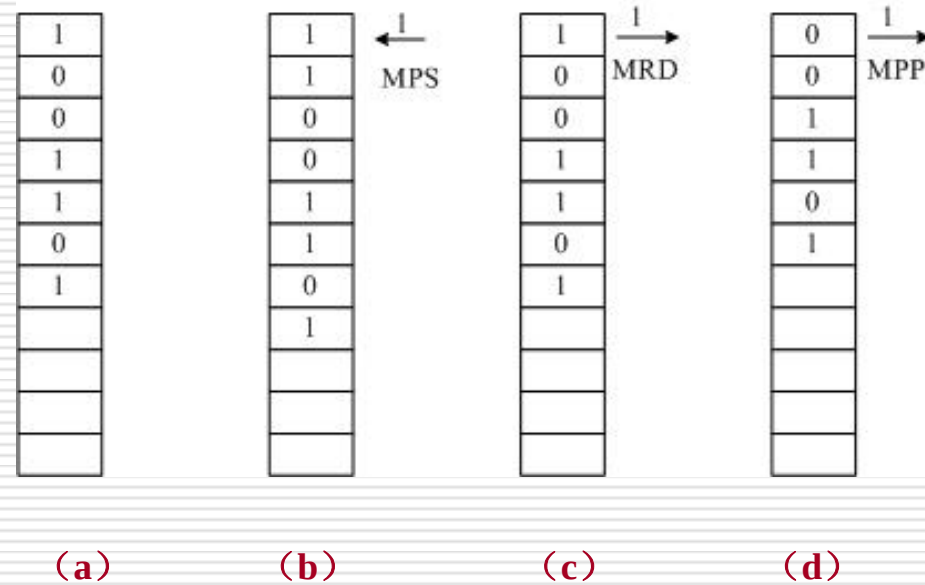
将该时刻的运算结果压入堆栈存储器的最上层，堆栈存储器原来存储的数据依次向下自动移一层

2. 读栈指令MRD

将堆栈存储器中最上层的数据读出。执行MRD指令后，堆栈存储器中的数据不发生任何变化

3. 出栈指令MPP

将堆栈存储器中最上层的数据取出，堆栈存储器原来存储的数据依次向上自动移一层。



- (a) 堆栈中原来情况
- (b) 执行MPS指令后的情况
- (c) 执行MRD指令后的情况
- (d) 执行MPP指令后的情况

*知识链接：栈操作指令 MPS MRD MPP

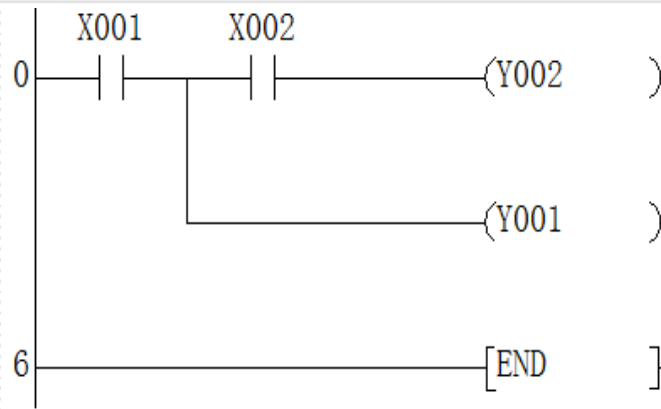
——多重输出的梯形图中使用的栈操作指令，用于存储编程时的中间运算结果。

MPS: **进栈**指令，将数据压入栈顶。用在**分支开始**处。

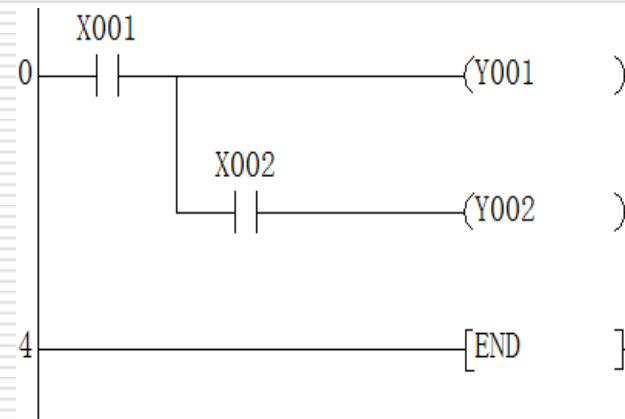
MRD: **读栈**指令，读取栈数据。用在MPS下继续的分支，表示**分支继续**。

MPP: **出栈**指令，取出栈顶数据。用在最后分支的地方，表示**分支结束**。

*知识链接：栈操作指令 MPS MRD MPP



0	LD	X001
1	MPS	
2	AND	X002
3	OUT	Y002
4	MPP	
5	OUT	Y001
6	END	



0	LD	X001
1	OUT	Y001
2	AND	X002
3	OUT	Y002
4	END	

*** 驱动输出时，有触点的分支放下面**

□ THE END