

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

:

逻辑字与、字或、字或非指令；
子程序调用程序指令、比较指令；

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

SOC3：设计三菱 PLC 控制生产线输送产品，检测产品到达位置并自动将产品抓取放到另外一条生产线程序

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：逻辑字与、字或、字或非指令；

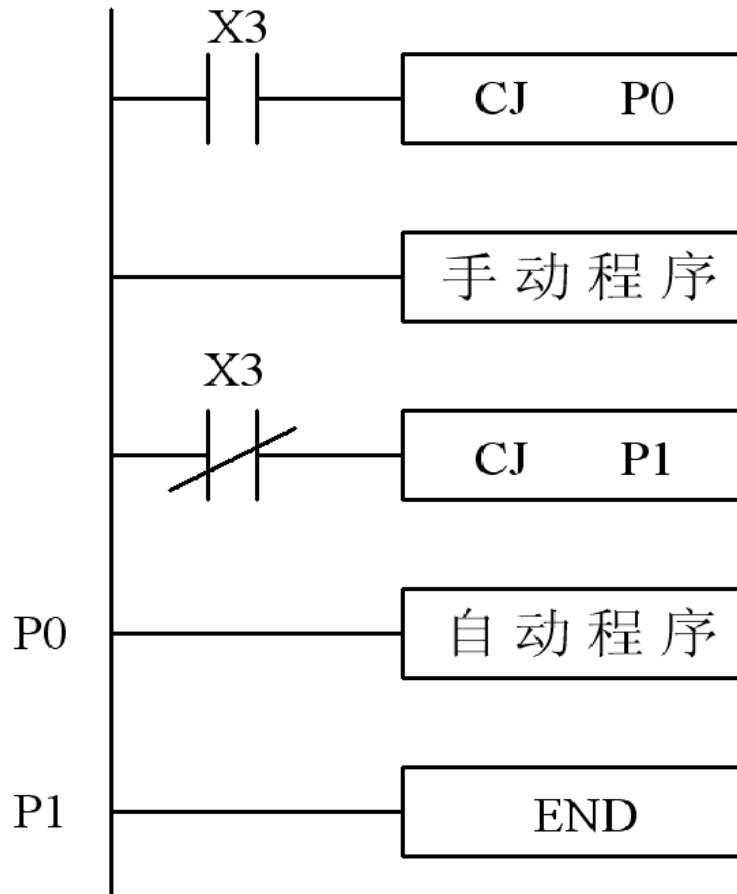
难点：子程序调用程序指令、比较指令；



一、教学回顾

1. 用跳转指令实现选择运行程序段

应用跳转指令的程序结构如图所示。X3 是手动 / 自动选择开关的信号输入端。当 X3 未接通时，执行手动程序段，反之执行自动程序段。X3 的常开 / 常闭接点起联锁作用，使手动、自动两个程序段只能选择其一。



手动 / 自动程序跳转

条件跳转指令 CJ

表：CJ 指令

条件跳转指令		操 作 数	程 序 步
P	FNC0 CJ	标号 P0 ~ P127 P63 表示跳到 END	CJ 3 步 标号 P 1 步

标号 P 的说明

(1) FX2N 系列 PLC 的标号 P 有 128 点 (P0 ~ P127) , 用于分支和跳转程序。

(2) 标号 P 放置在左母线的左边, 一个标号只能出现一次, 如出现两次或两次以上, 程序报错。标号 P 占一步步长。

跳转指令 CJ 的说明

(1) 如果跳转条件满足, 则执行跳转指令, 程序跳到以标号 P 为入口的程序段中执行。否则不执行跳转指令, 按顺序执行下一条指令。

(2) 多个跳转指令可以使用同一个标号。

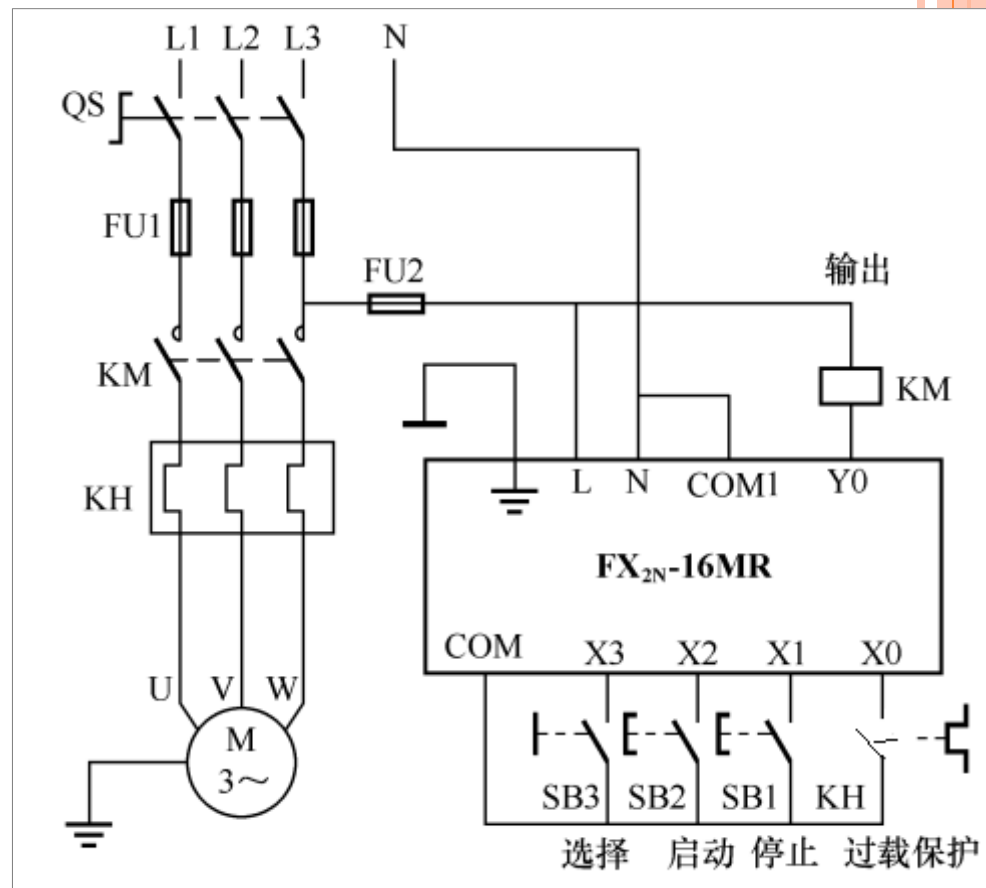
(3) 如果用 M8000 作为控制跳转的条件, CJ 则变成无条件跳转指令。

条件跳转指令应用举例

某台设备具有手动 / 自动两种操作方式。SB3 是操作方式选择开关，当 SB3 处于断开状态时，选择手动操作方式；当 SB3 处于接通状态时，选择自动操作方式，不同操作方式进程如下：

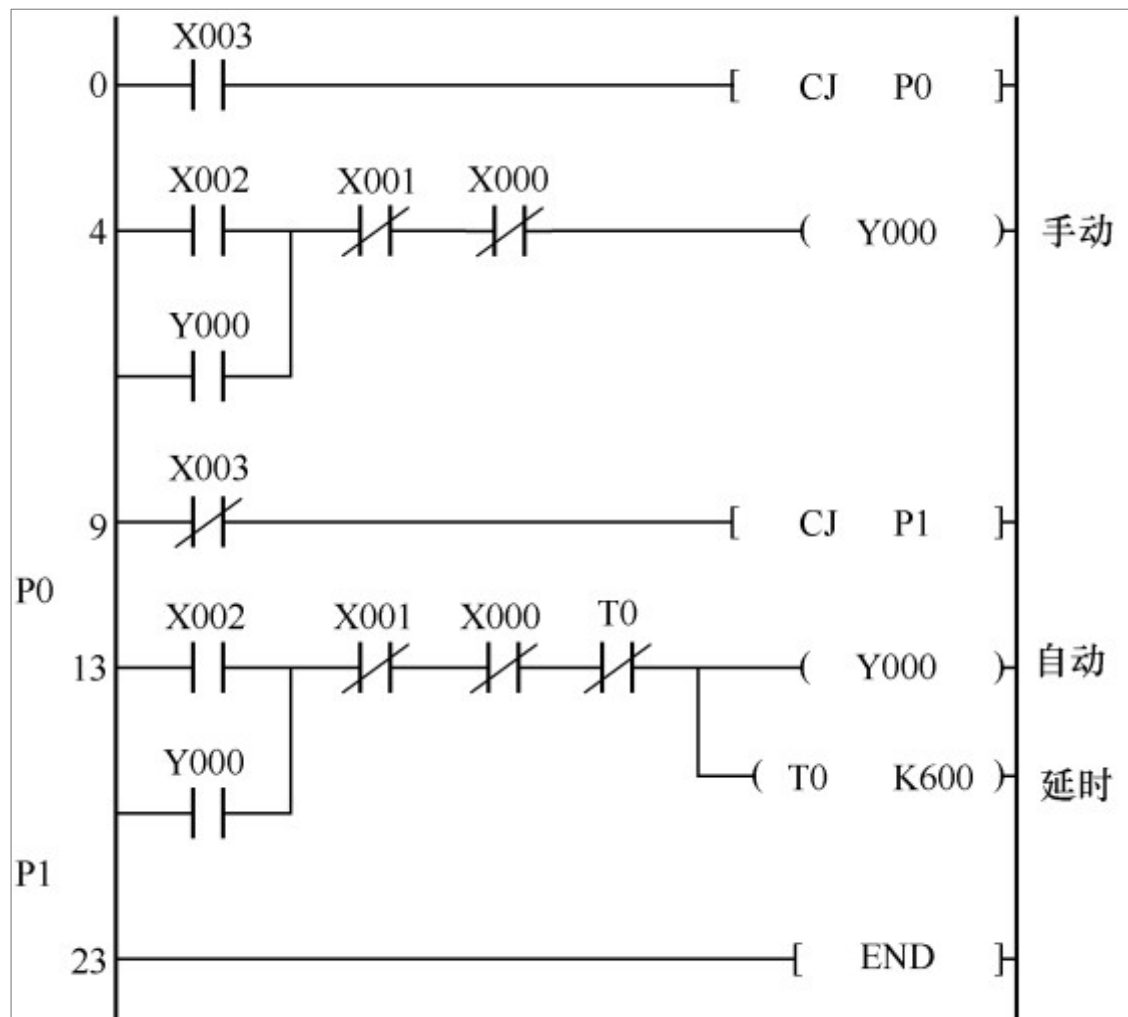
手动操作方式进程：按启动按钮 SB2，电动机运转；按停止按钮 SB1，电动机停机。

自动操作方式进程：按启动按钮 SB2，电动机连续运转 1min 后，自动停机。按停止按钮 SB1，电动机立即停机。



例：控制线路图

根据控制要求，设计程序梯形图如图所示。



例：程序梯形图

2. 加法指令 ADD

ADD 指令

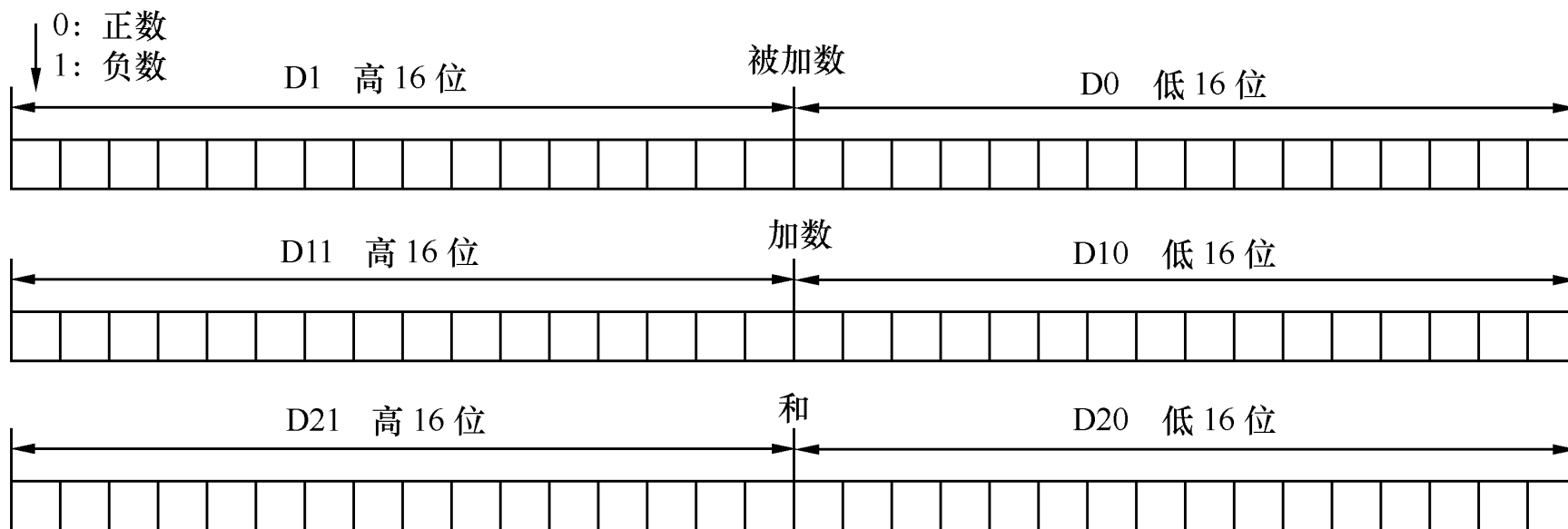
加 法 指 令		操 作 数	
D	FNC20 ADD	S1 、 S2	K、 H、 KnX、 KnY、 KnM、 KnS、 T、 C、 D 、 V、 Z
P		D	KnY、 KnM、 KnS、 T、 C、 D、 V、 Z

加法指令 ADD 的说明

- (1) 加法运算是代数运算。
- (2) 若相加结果为 0 ， 则零标志位 $M8020 = 1$ ， 可用来判断
两个数是否为相反数。
- (3) 加法指令可以进行 32 位操作方式。



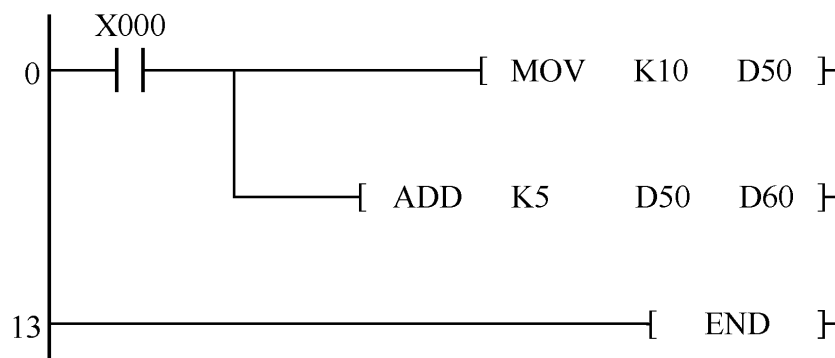
例如指令语句 “DADD D0 D10 D20” 的操作数构成如图所示。被加数的低 16 位在 D0 中，高 16 位在 D1 中；加数的低 16 位在 D10 中，高 16 位在 D11 中；“和” 的低 16 位在 D20 中，高 16 位在 D21 中。



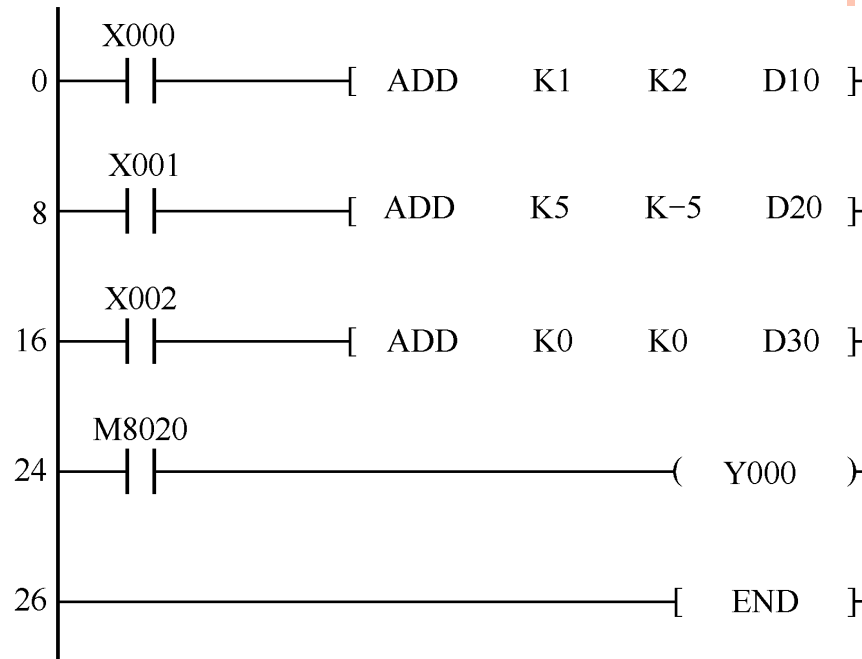
32 位加法指令操作数的构成



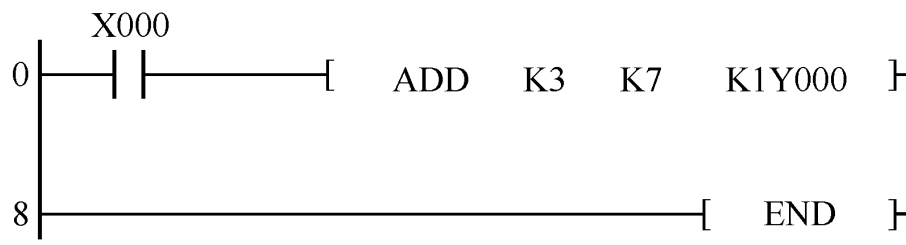
加法指令 ADD 举例



加法指令 ADD 的举例 1



加法指令 ADD 的举例 2



加法指令 ADD 的举例 3



M8020——零标志。
M8021——错位标志。
M8022——进位标志。
M8029——指令执行完毕标志。
M8046——STL 状态置 1。
M8246——C246 减计数监视。
M8247——C247 减计数监视。



3. 减法指令 SUB

SUB 指令

减 法 指 令		操 作 数	
D	FNC21 SUB	S1、S2	K、H、KnX、KnY、KnM、KnS、T、C、D、V、Z
P		D	KnY、KnM、KnS、T、C、D、V、Z

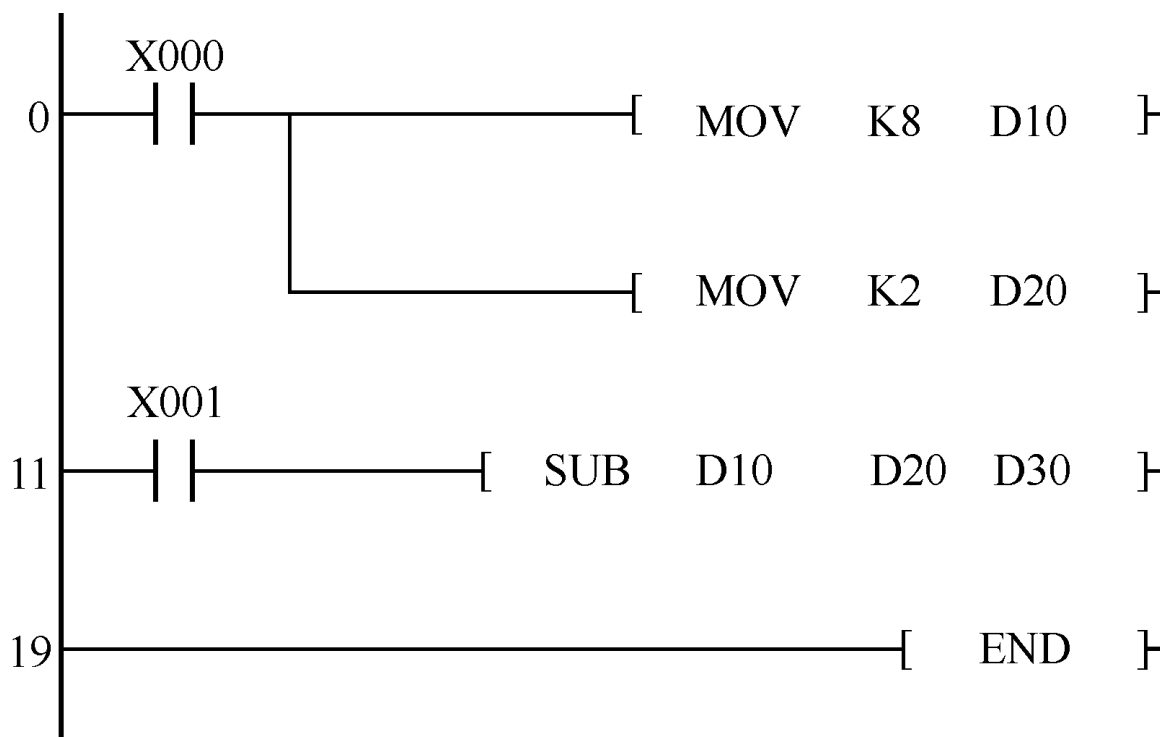
减法指令 SUB 的说明

- (1) 减法运算是代数运算。
- (2) 若相减结果为 0 时，则零标志位 $M8020 = 1$ ，可用来判断两个数是否相等。
- (3) SUB 可以进行 32 位操作方式，例如指令语句：DSUB D0 D10 D20。



减法指令 SUB 举例

两个数据寄存器中存储的数据相减，程序如图所示。如果 X0 接点闭合，执行数据传送指令。如果 X1 接点闭合，执行减法指令，减法运算的结果差（ $8-2=6$ ）存在 D30 中。



减法指令 SUB 的举例



4. 乘法指令 MUL

MUL 指

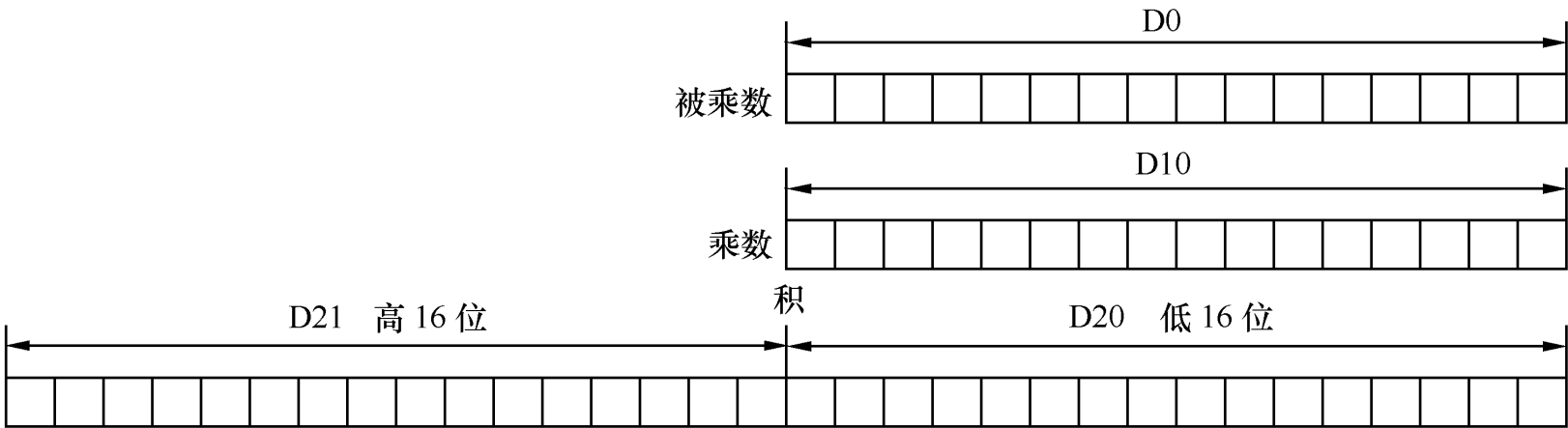
令
操作数

乘 法 指 令		操 作 数	
D	FNC22 MUL	S1、S2	K、H、KnX、KnY、KnM、KnS、T、C、D、V、Z
D		D	KnY、KnM、KnS、T、C、D、V、Z

乘法指令 MUL 的说明

- (1) 乘法运算是代数运算。
- (2) 16 位数乘法：源操作数 S1、S2 是 16 位，目标操作数 D 占用 32 位。

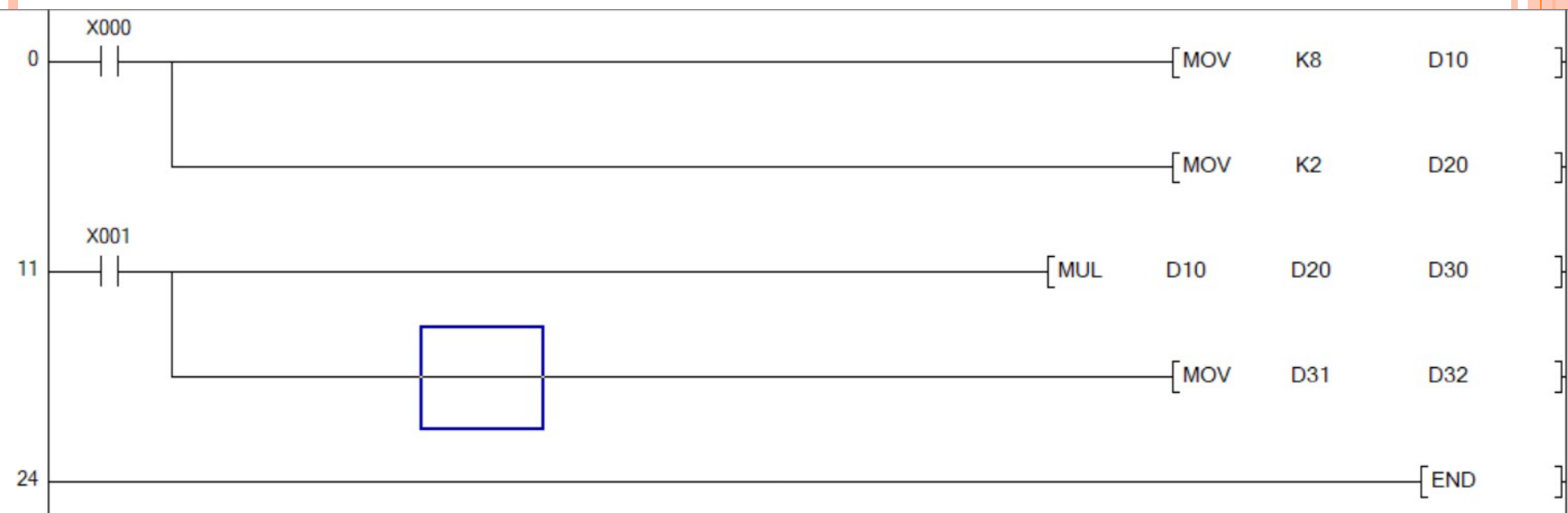
例如乘法指令语句 “MUL D0 D10 D20”，被乘数存储在 D0，乘数存储在 D10，积则存储在 D21、D20 组件中。操作数结构如图所示。



16 位乘法的积占用 32 位

乘法指令 MUL 举例

运行监控模式的程序梯形图如图所示。如果 X0 接点闭合，执行数据传送指令。如果 X1 接点闭合，执行乘法指令，乘法运算的结果（ $8 \times 2 = 16$ ）存储在 D31、D30 目标操作数中。图中 D31 存储的数据为 0，D30 存储的数据为 16。



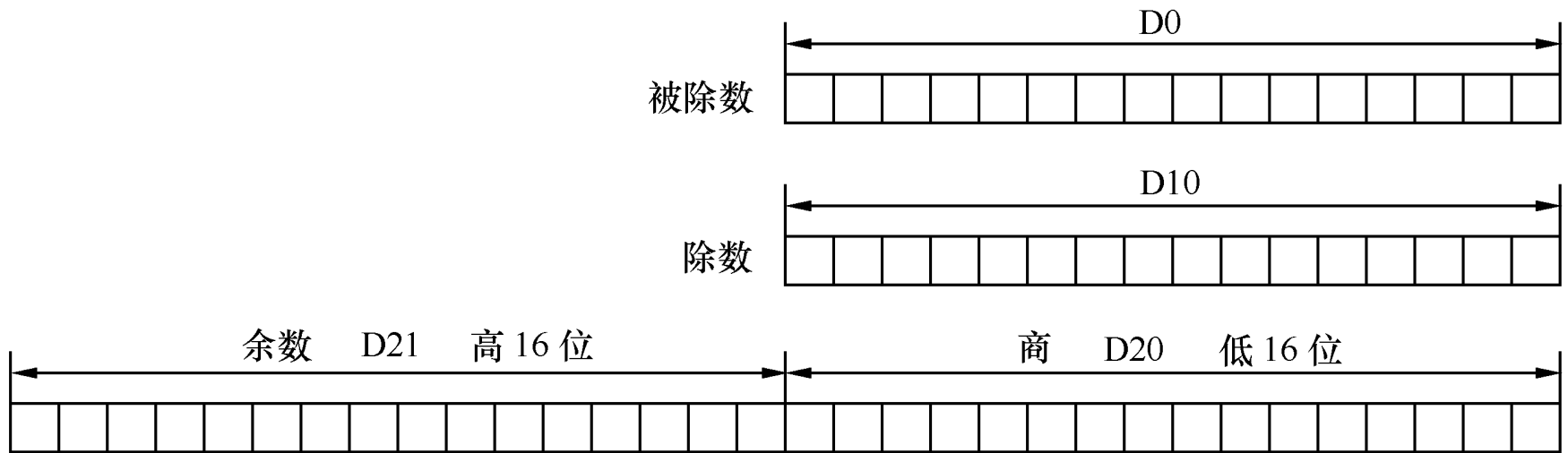
5. 除法指令 DIV

DIV 指令			
除 法 指 令		操 作 数	
D	FNC23 DIV	S1、S2	K、H、KnX、KnY、KnM、KnS、T、C、D、V、Z
P		D	KnY、KnM、KnS、T、C、D、V、Z

除法指令 DIV 的说明

- (1) 除法运算是代数运算。
- (2) 16 位数除法：源操作数 S1、S2 是 16 位，目标操作数 D 占用 32 位。除法运算的结果商存储在目标操作数的低 16 位，余数存储在目标操作数的高 16 位中。
- (3) 32 位除法：源操作数 S1、S2 是 32 位，但目标操作数却是 64 位。除法运算的结果商存储在目标操作数的低 32 位，余数存储在目标操作数的高 32 位。

例除法指令语句 “DIV D0 D10 D20”，被除数存储在 D0，除数存储在 D10，商存储在 D20，余数存储在 D21，操作数的结构如图所示。

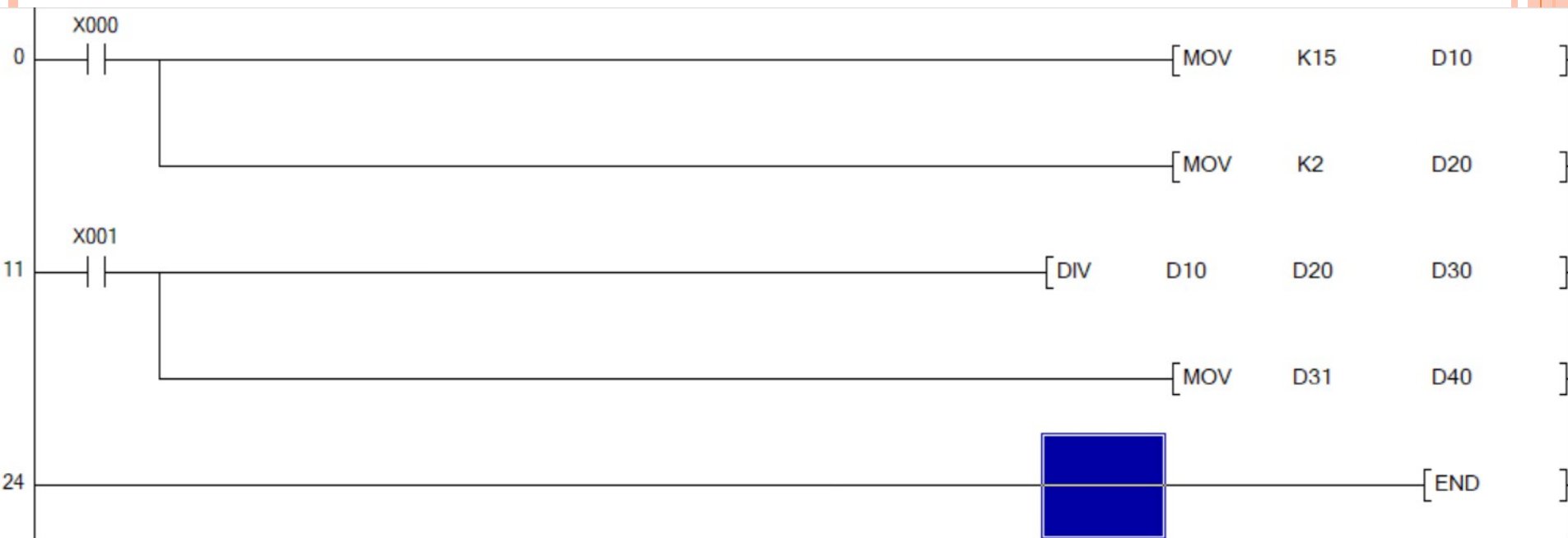


16 位除法的商和余数构成 32 位目标操作数



除法指令 DIV 举例

运行监控模式的程序梯形图如图所示。如果 X0 接点闭合，执行数据传送指令。如果 X1 接点闭合，执行除法指令。除法运算结果的商 7 存储在 D30，余数 1 存储在 D31。可以看出，数据除 2 后根据余数为 1 或为 0 可判断数据的奇偶性。



除法指令 DIV 的举例

6. 加 1 指令 INC

加 1 指令		INC 指令 操作数	
D	FNC24 INC	D	KnY、KnM、KnS、T、C、D、V、Z
P			

加 1 指令 INC 的说明

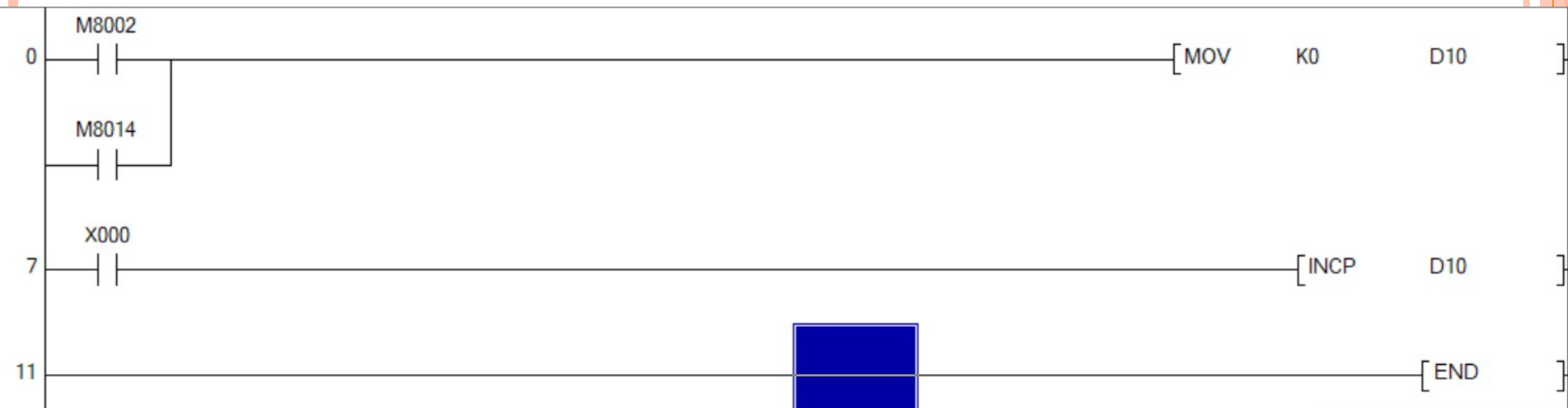
- （1）INC 指令的执行结果不影响零标志位 M8020。
- （2）在实际控制中通常不使用每个扫描周期目标操作数都要加 1 的连续执行方式，所以，INC 指令经常使用脉冲操作方式。脉冲 P 与上升沿 / 下降沿

减 1 指令 DEC 和加 1 指令 INC 执行方式相似。

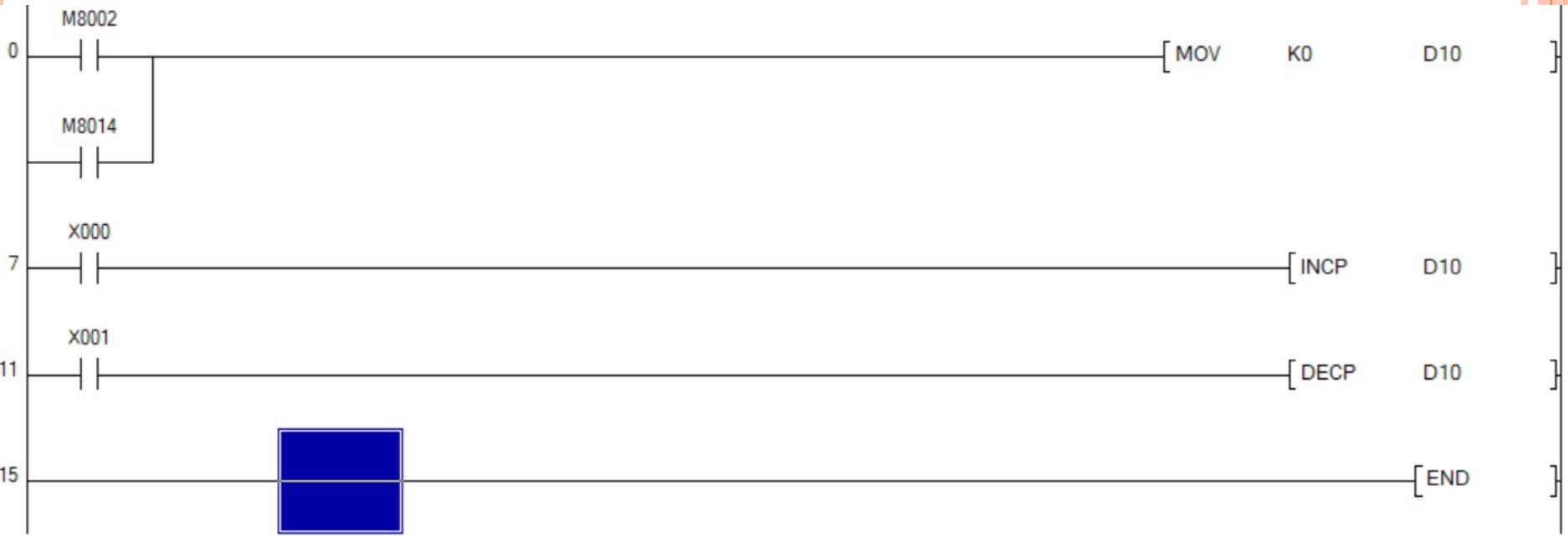


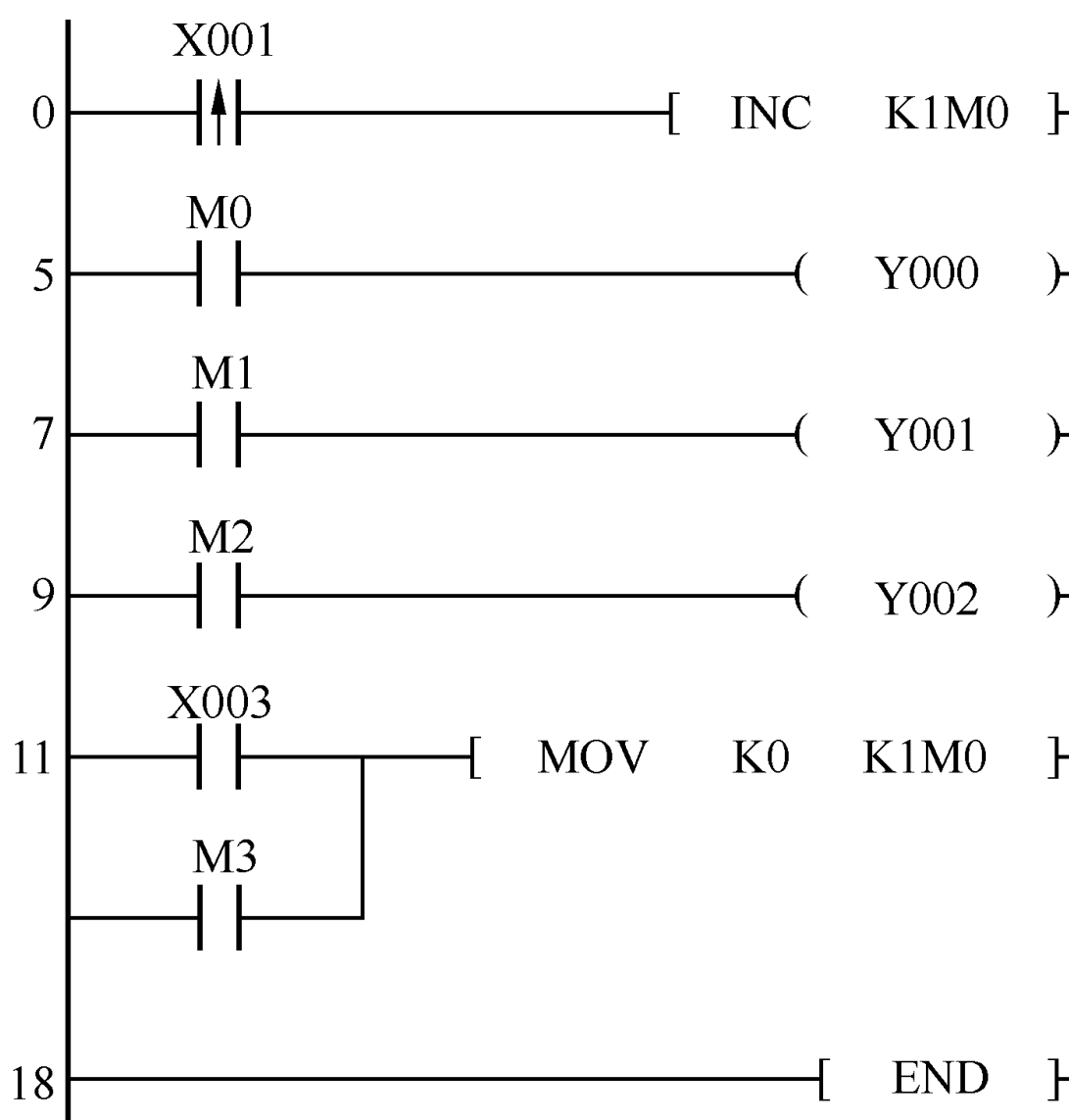
加 1 指令 INC 举例

运行监控模式的程序梯形图如图所示。开机初始脉冲 M8002 将数据寄存器 D10 清 0。在 X0 接点闭合的那个扫描周期执行加 1 指令，D10 的数据被加 1 后存储，即 $(D10) + 1 \rightarrow (D10)$ 。图中 X0 共接通 5 次，D10 中存储的数据由 0 增加到 5。



加 1 指令 INC 的举例

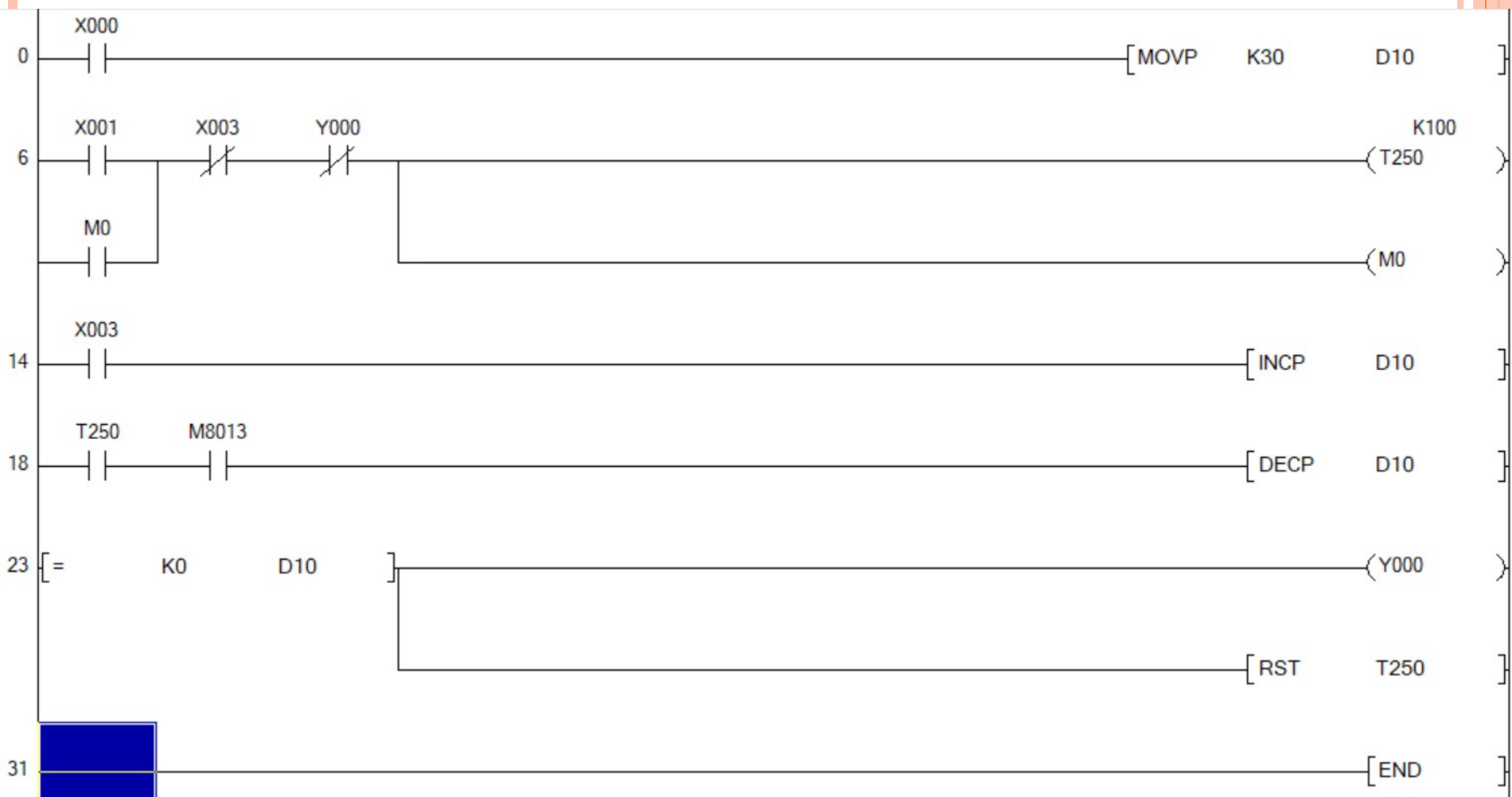




```

0    LDP    X001
2    INC    K1M0
5    LD     M0
6    OUT    Y000
7    LD     M1
8    OUT    Y001
9    LD     M2
10   OUT    Y002
11   LD     X003
12   OR     M3
13   MOV    K0    K1M0
18   END

```



二、讲授新内容

1. 逻辑字“与”指令 WAND

字“与”指令		操 作 数	
D	FNC26 WAND	S1、S2	K、H、KnX、KnY、KnM、KnS、T、C、D、V、Z
P		D	KnY、KnM、KnS、T、C、D、V、Z

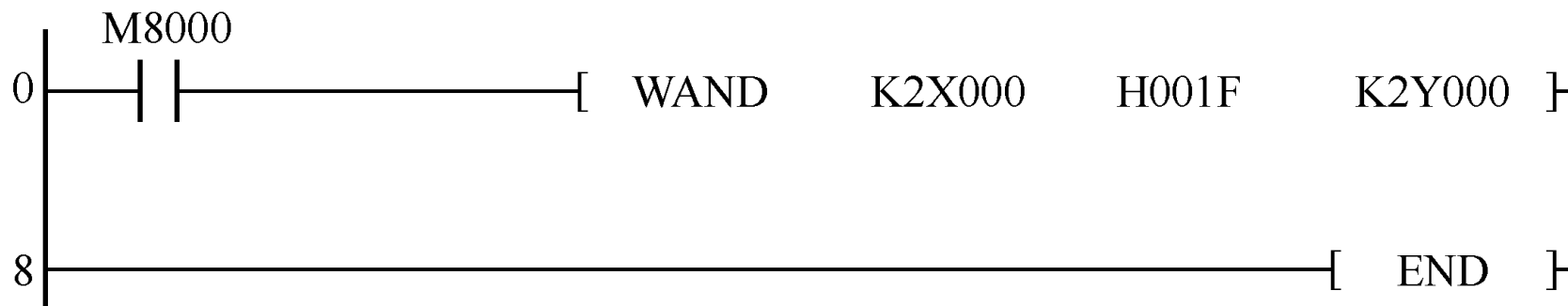
字“与”指令 WAND 的说明

(1) S1、S2 为作相“与”逻辑运算的源操作数，D 为存储“与”逻辑运算结果的目标操作数。

(2) 字“与”指令的功能是将两个源操作数的数据，进行二进制按位相“与”，并将运算结果存入目标操作数。

字“与”指令 WAND 举例

假设要求用输入继电器 X0 ~ X4 的位状态去控制输出继电器 Y0 ~ Y4，可用字元件 K2X0 去控制字元件 K2Y0。对字元件多余的控制位 X5、X6 和 X7，可与 0 相“与”进行屏蔽。程序如图 5.23 所示。



	X7	X6	X5	X4	X3	X2	X1	X0
K2X0	1	0	1	1	1	0	1	0

H1F	0	0	0	1	1	1	1	1
-----	---	---	---	---	---	---	---	---

K2Y0	0	0	0	1	1	0	1	0
------	---	---	---	---	---	---	---	---

与运算

全 1 出 1 ——— 有 0 出 0



2. 逻辑字“或”指令 WOR

字“或”指令		操 作 数	
D	FNC27 WOR	S1、S2	K、H、KnX、KnY、KnM、KnS、T、C、D、V、Z
P		D	KnY、KnM、KnS、T、C、D、V、Z

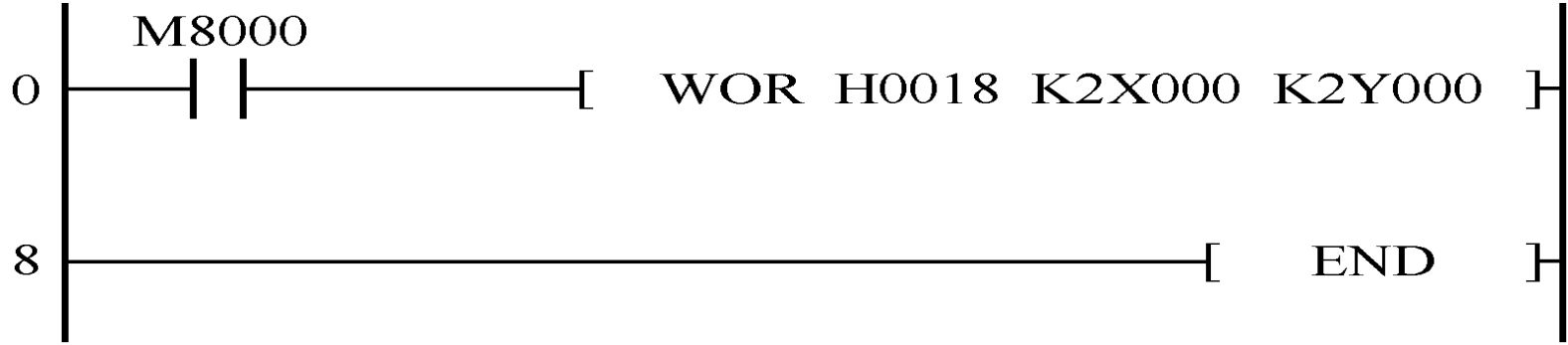
字“或”指令 WOR 的说明

（1）S1、S2 为两个相“或”的源操作数，D 为存储“或”逻辑结果的目标操作数。

（2）指令的功能是将两个源操作数的数据，进行二进制按位相“或”，并将运算结果存入目标操作数。

字“或”指令 WOR 举

^例要求用输入继电器组成的字元件 K2X0 去控制由输出继电器组成的字元件 K2Y0，但 Y3、Y4 位不受字元件 K2X0 的控制而始终处于 ON 状态。可用字“或”指令屏蔽 X3、X4 位，程序如图 5.25 所示。



	X7	X6	X5	X4	X3	X2	X1	X0
K2X0	1	0	1	0	1	0	1	0

H18	0	0	0	1	1	0	0	0
-----	---	---	---	---	---	---	---	---

或运算

K2Y0	1	0	1	1	1	0	1	0
------	---	---	---	---	---	---	---	---

有 1 出 1 全 0 出 0

辅助继电器 M

FX3U : 特殊用 M8000~M8511

512 触点型特殊辅助继电器

(1) M8000 (M8001) —— 运行监视用特殊辅助继电器。

PLC 运行时 M8000 得电 (M8001 断电)，PLC 停止时 M8000 失电 (M8001 得电)。如图 5 所示。

(2) M8002 (M8003) —— 初始脉冲特殊辅助继电器。

M8002 (M8003) 只在 PLC 开始运行的第一个扫描周期内得电 (断电)，其余时间均断电 (得电)，如图 6 所示。

常用 M8002 的触点作为一些软元件的初始化复位信号。

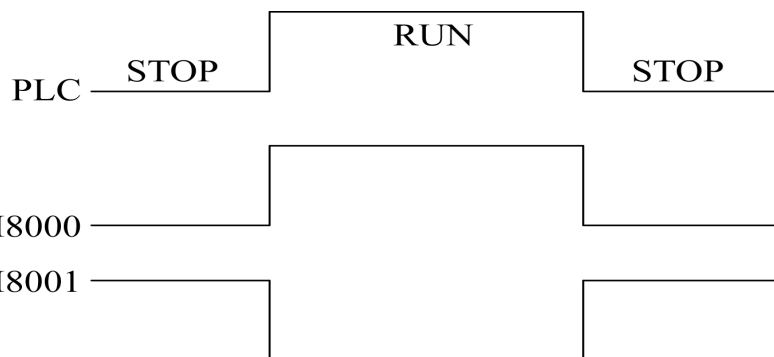


图 5 运行监视用特殊辅助继电器 M8000 (M8001)

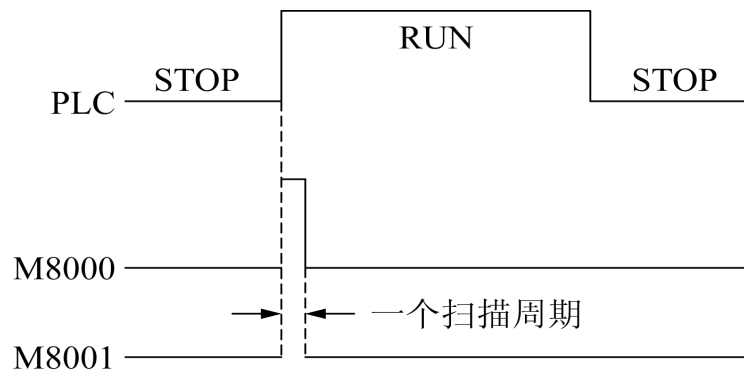


图 6 初始脉冲特殊辅助继电器 M8002 (M8003)

3. 逻辑字 “异或” 指令 WXOR

字 “异或” 指令		操 作 数	
D	FNC28 WXOR	S1 、 S2	K 、 H 、 KnX 、 KnY 、 KnM 、 KnS 、 T 、 C 、 D 、 V 、 Z
P		D	KnY 、 KnM 、 KnS 、 T 、 C 、 D 、 V 、 Z

字 “异或” 指令 WXOR 的说明

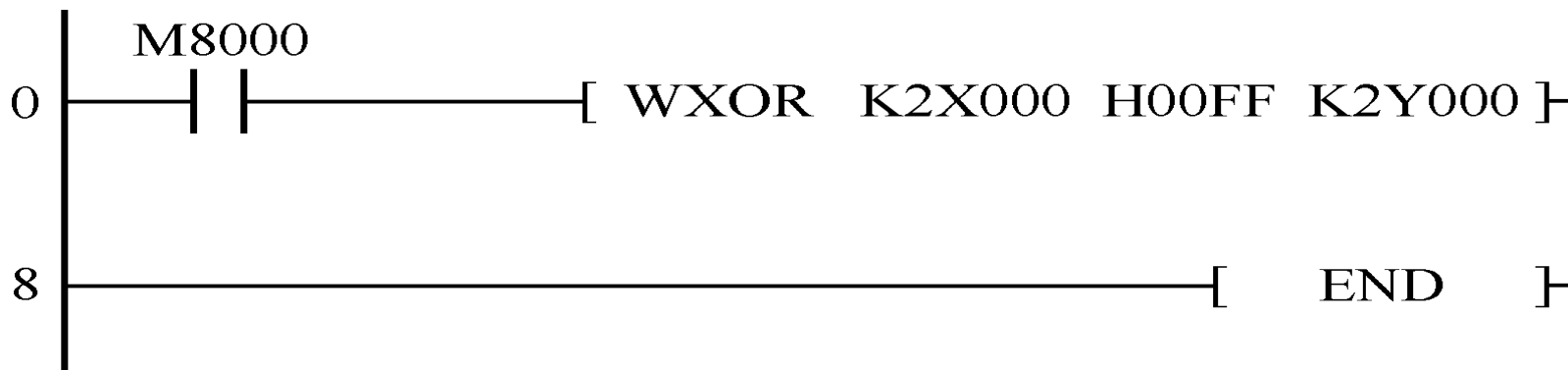
（1）S1 、 S2 为两个相 “异或” 的源操作数， D 为存储 “异或” 逻辑结果的目标操作数。

（2）指令的功能是将两个源操作数的数据，进行二进制按位相 “异或”，并将运算结果存入目标操作数。

字 “异或” 指令 WXOR

举例

要求用输入继电器组成的字元件 K2X0 的相反状态去控制由输出继电器组成的字元件 K2Y0，即 X 某位为 “1” 时，Y 的相应位为 “0”；X 某位为 “0” 时，Y 的相应位为 “1”。程序如图 5.27 所示。



	X7	X6	X5	X4	X3	X2	X1	X0
K2X0	1	0	1	0	1	0	1	0

H0FF	1	1	1	1	1	1	1	1
------	---	---	---	---	---	---	---	---

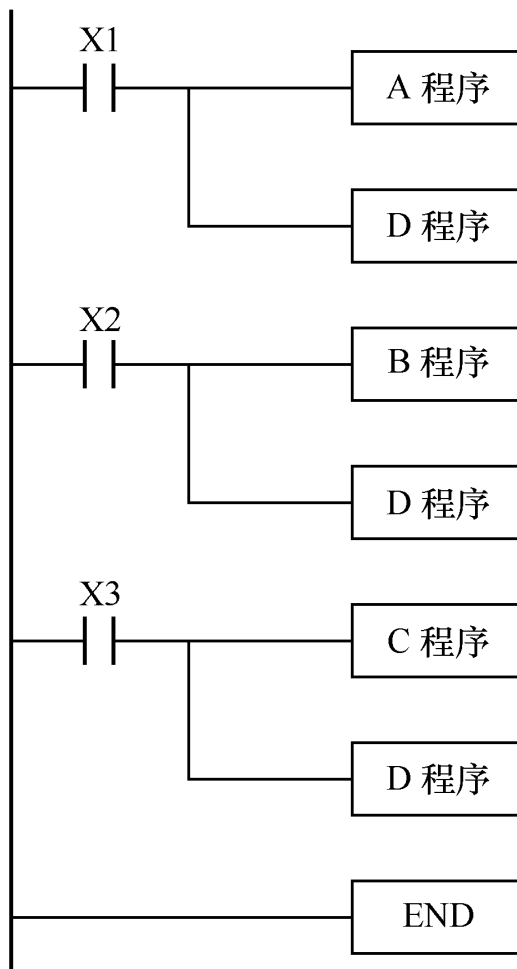
异或运算

K2Y0	0	1	0	1	0	1	0	1
------	---	---	---	---	---	---	---	---

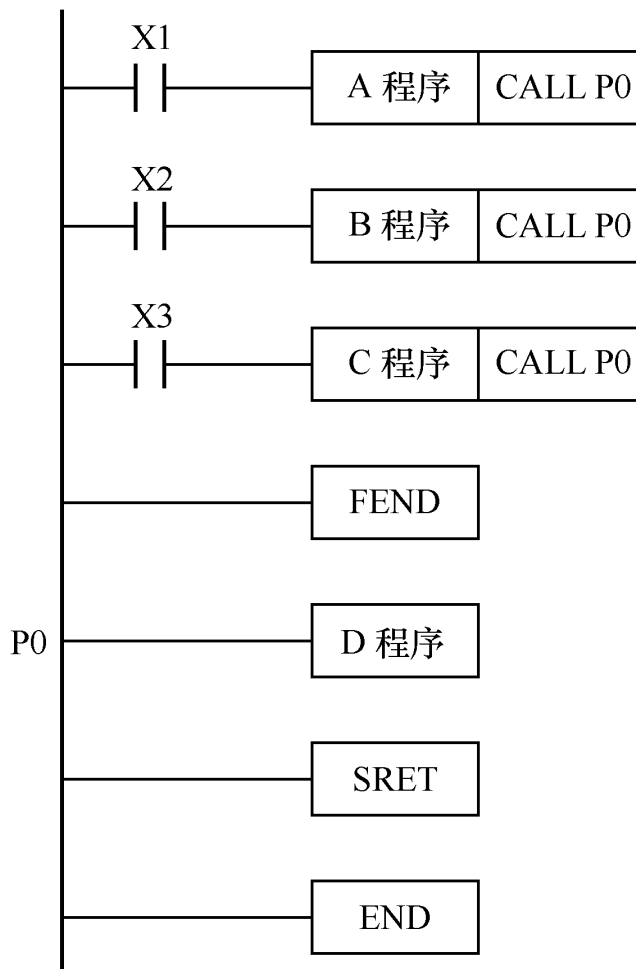
相异出 1

相同出 0

4. 子程序调用指令及应用



(a) 顺序控制程序



(b) 子程序调用程序



子程序指令 CALL、 SRET 与主程序结束指令 FEND

指令助记符		操 作 数	程 序 步
P	FNC1 CALL	标号 P0 ~ P62	CALL 3 步
		标号 P64 ~ P127	标号 P 1 步
	FNC2 SRET	无	SRET 1 步
	FNC6 FEND	无	FEND 1 步

FEND 指令表示主程序结束。**END** 是指整个程序（包括主程序和子程序）结束。一个完整的程序可以没有子程序，但一定要有主程序。

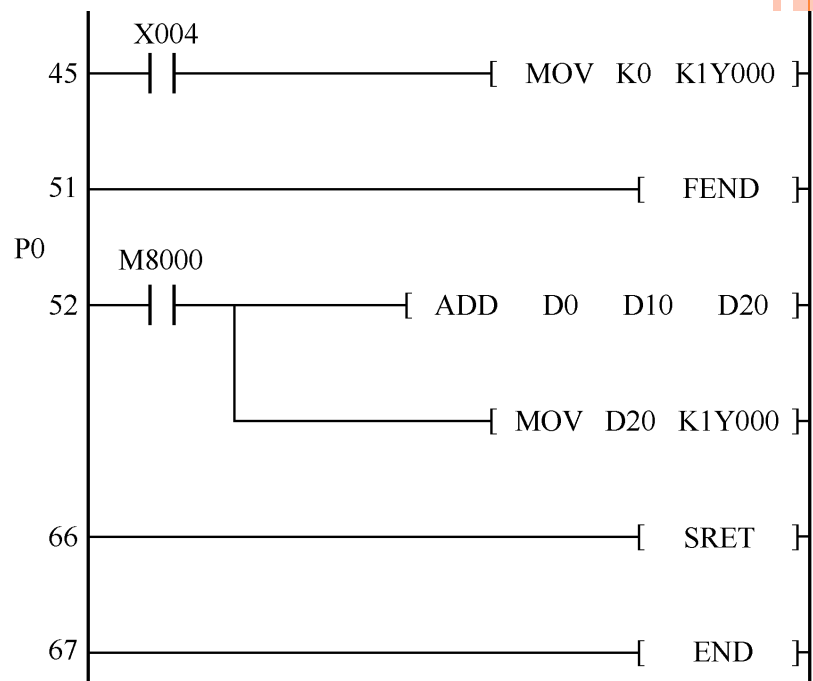
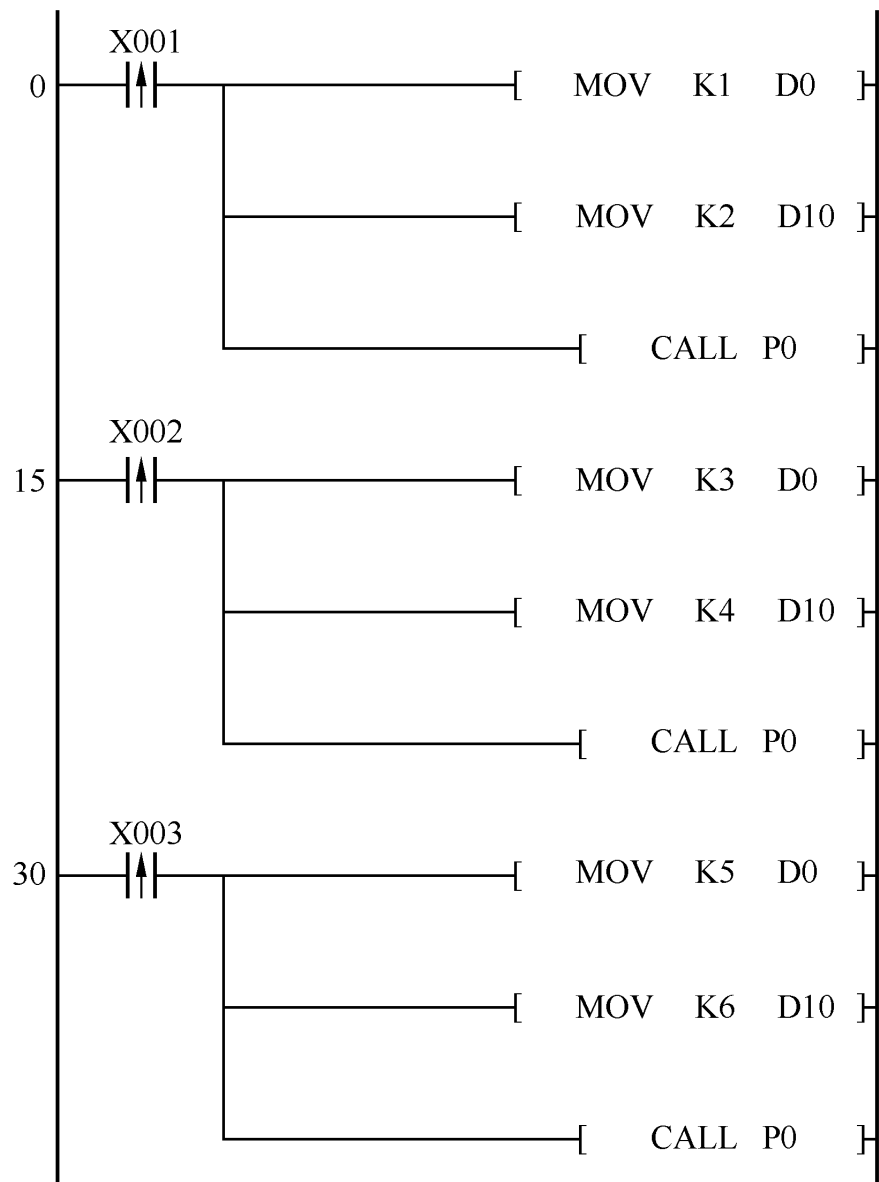
子程序编写在 **FEND** 指令的后面，以标号 **P** 开头，以返回指令 **SRET** 结束。

如果子程序调用条件满足，则中断主程序去执行子程序，标号是被调用子程序的入口地址。在子程序结束处一定要使用返回指令 **SRET**，意思是返回主程序中断处去继续执行主程序的下一条指令语句。

在子程序中，使用定时器的范围是 **T192 ~ T199**。

如果在子程序中再调用其他子程序称为子程序嵌套，嵌套总数可达 **5 级**。
标号 **P63** 相当于 **END**。

子程序调用指令 **CALL** 与跳转指令 **CJ** 不能使用相同的标号。



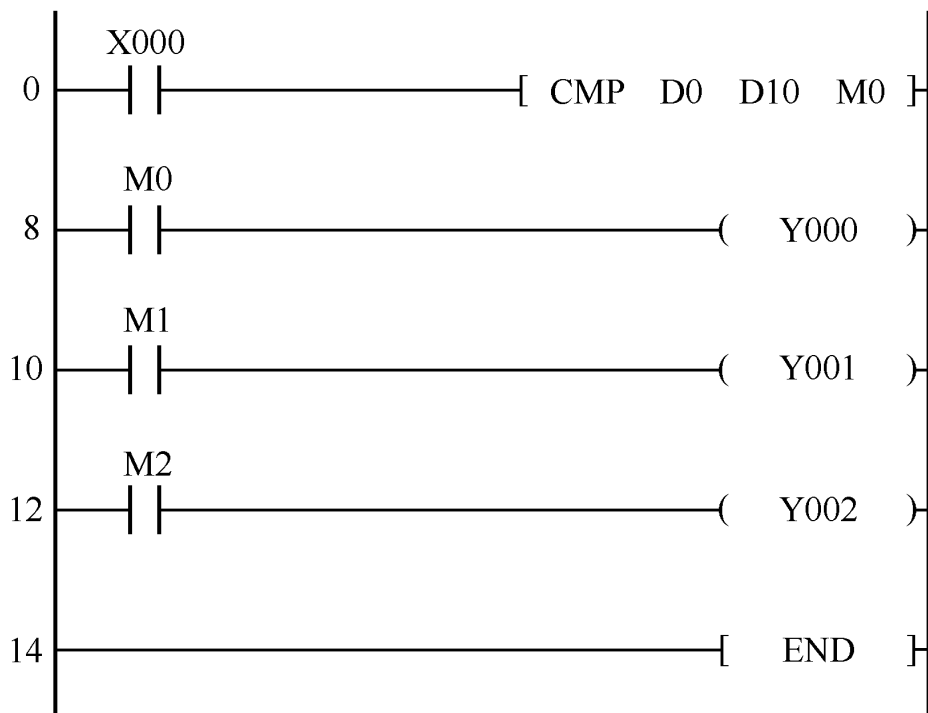
程序功能是：

X1、**X2**、**X3** 分别接通时，将相应的数据传送到 **D0**、**D10**，然后调用子程序；在子程序中，将 **D0**、**D10** 存储的数据相加，运算结果存储在 **D20**，用 **D20** 存储数据控制输出字元件 **K1Y0**。

5. 比较指令

组件比较指令 CMP

比 较 指 令		操 作 数	
D	FNC10 CMP	S1 、 S2	K、 H、 KnX、 KnY、 KnM、 KnS、 T、 C、 D、 V、 Z
P		D	Y、 M、 S



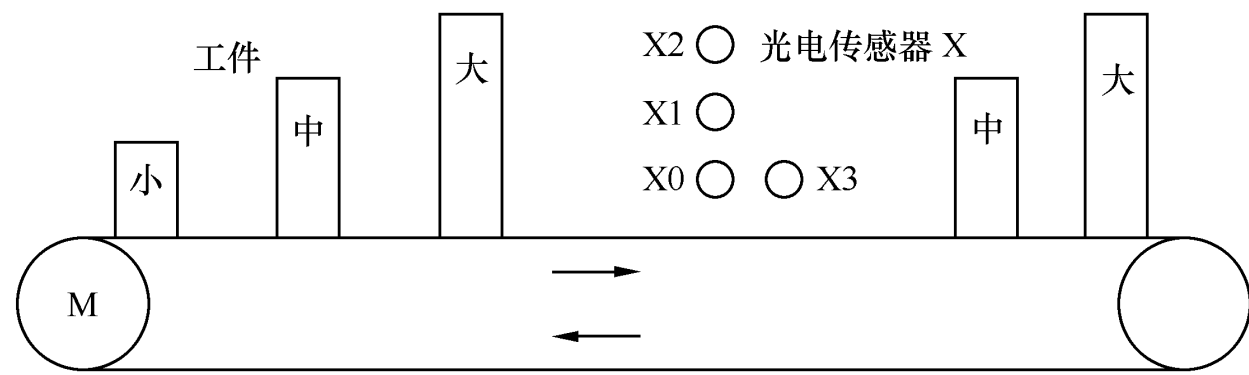
标志位的规则：

若 (D0) > (D10) ， 则
M0 置 1 ， M1 、 M2 为 0 ；

若 (D0) = (D10) ， 则
M1 置 1 ， M0 、 M2 为 0 ；

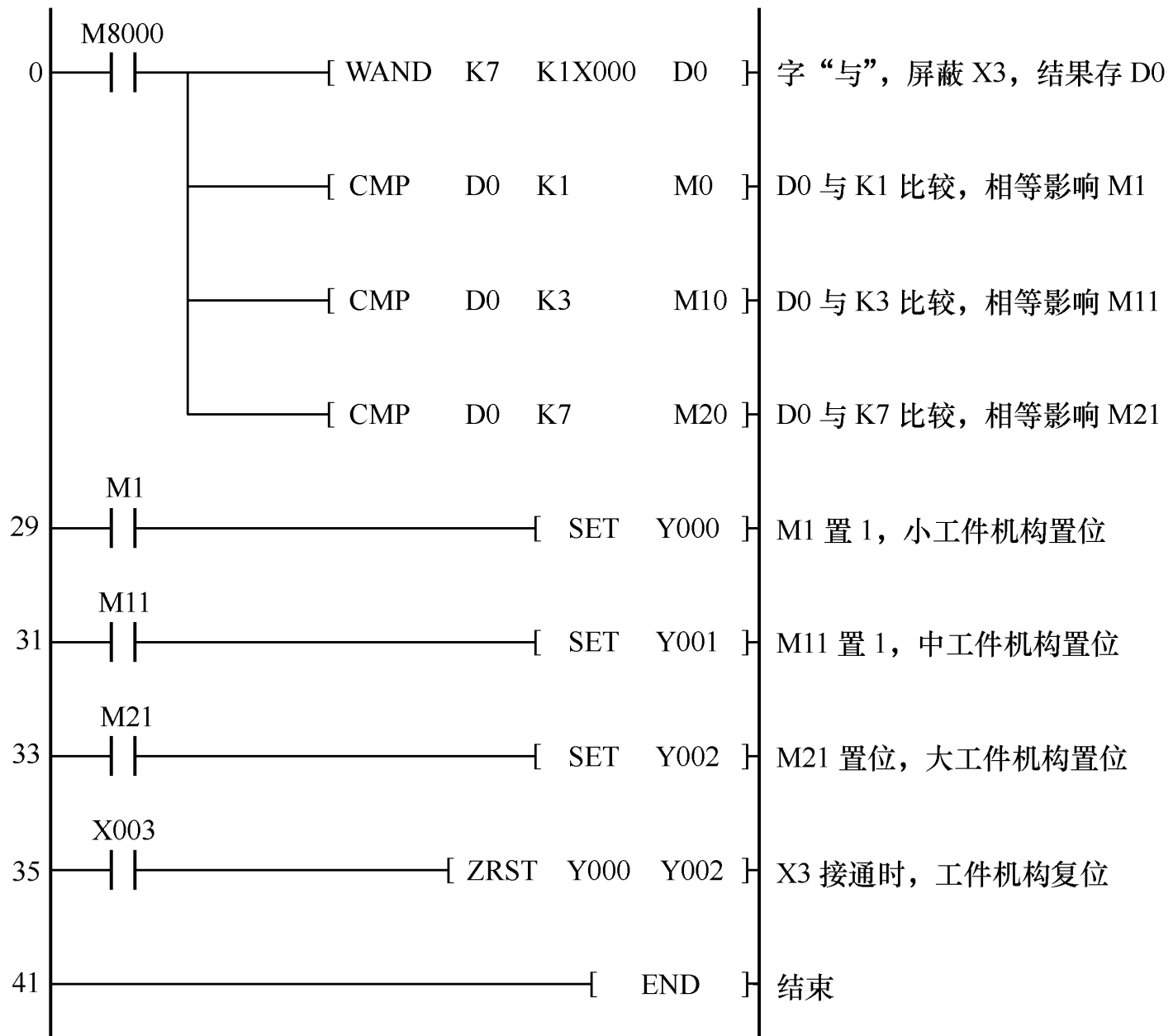
若 (D0) < (D10) ， 则
M2 置 1 ， M0 、 M1 为 0 。

如图所示的传送带输送大、中、小三种规格的工件，用连接 X0、X1、X2 端子的光电传感器判别工件规格，然后启动分别连接 Y0、Y1、Y2 端子的相应操作机构；连接 X3 的光电传感器用于复位操作机构。用比较指令 CMP 编写工件规格判别程序。



工件规格与光电信号转换关系

工 件 规 格	光电信号输入控制字 K1X0				光电转换数据
	X3	X2	X1	X0	
小	0	0	0	1	K1
中	0	0	1	1	K3
大	0	1	1	1	K7

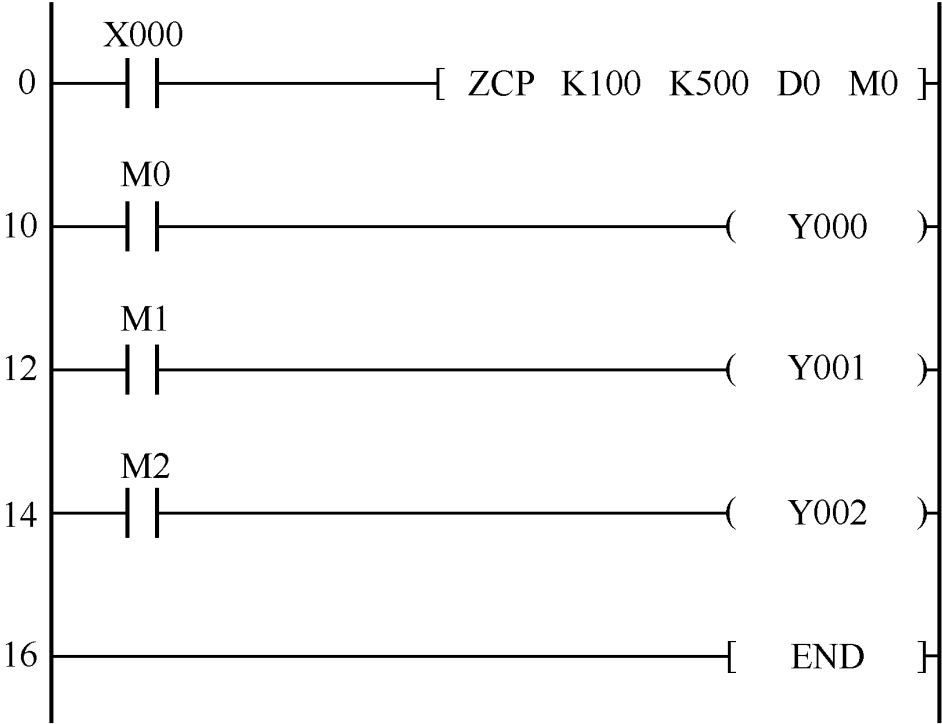


传送带工件规格判别程序



区域、区间比较指令

指令名称	助记符	指令代码 位数	操作数范围			程序步
			[S1 .]	[S2.]	[D.]	
区域比较	ZCP ZCP(P)	FNC11 (16/32)	K、H KnX、Kn Y、KnM 、KnS T、C、D 、V、Z	Y、 M、 S		ZCP、ZCPP ...9步 DZCP、DZC PP...17步

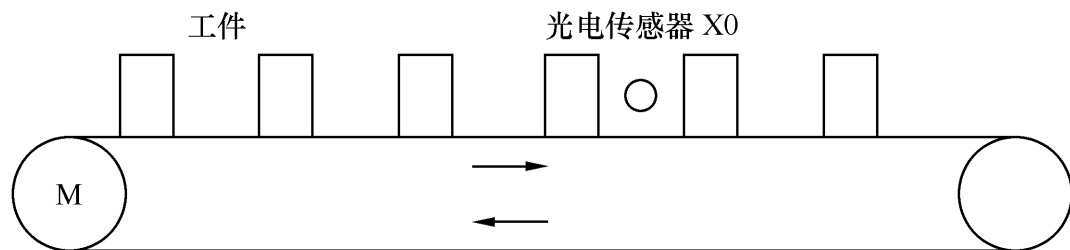


标志位的规则：
若 $K100 > (D0)$ ，则 M0 置 1，M1、M2 为 0；
若 $K100 \leq (D0) \leq K500$ ，则 M1 置 1，M0、M2 为 0；
若 $K500 < (D10)$ ，则 M2 置 1，M0、M1 为 0。

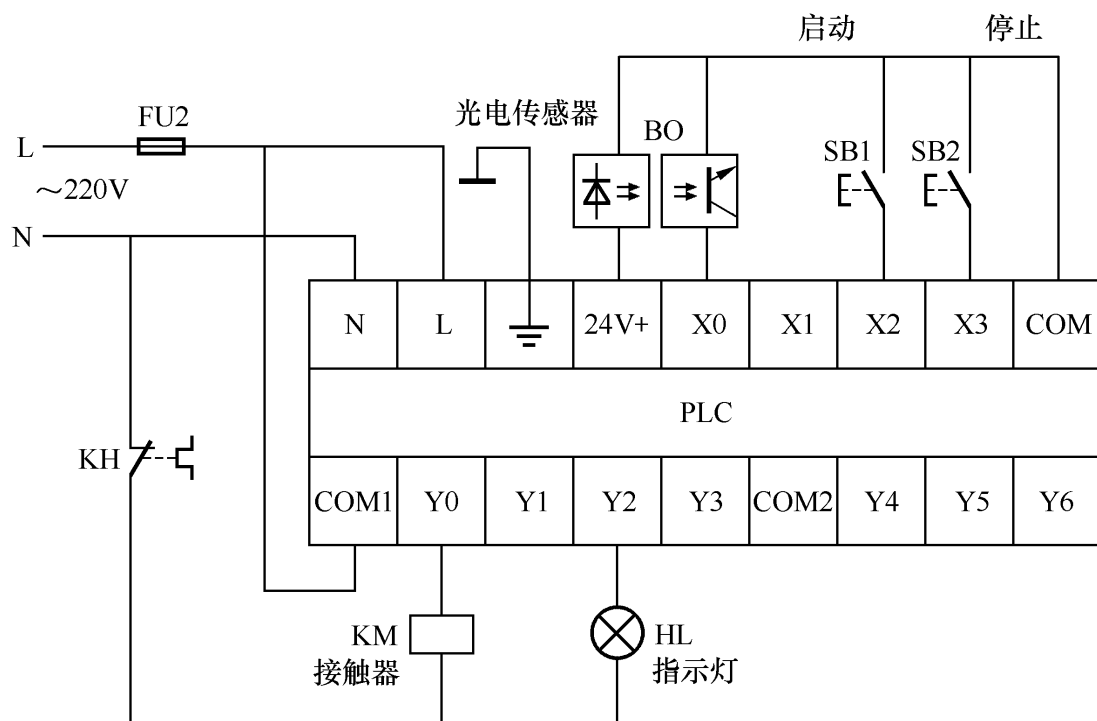


区间比较指令 ZCP 举例

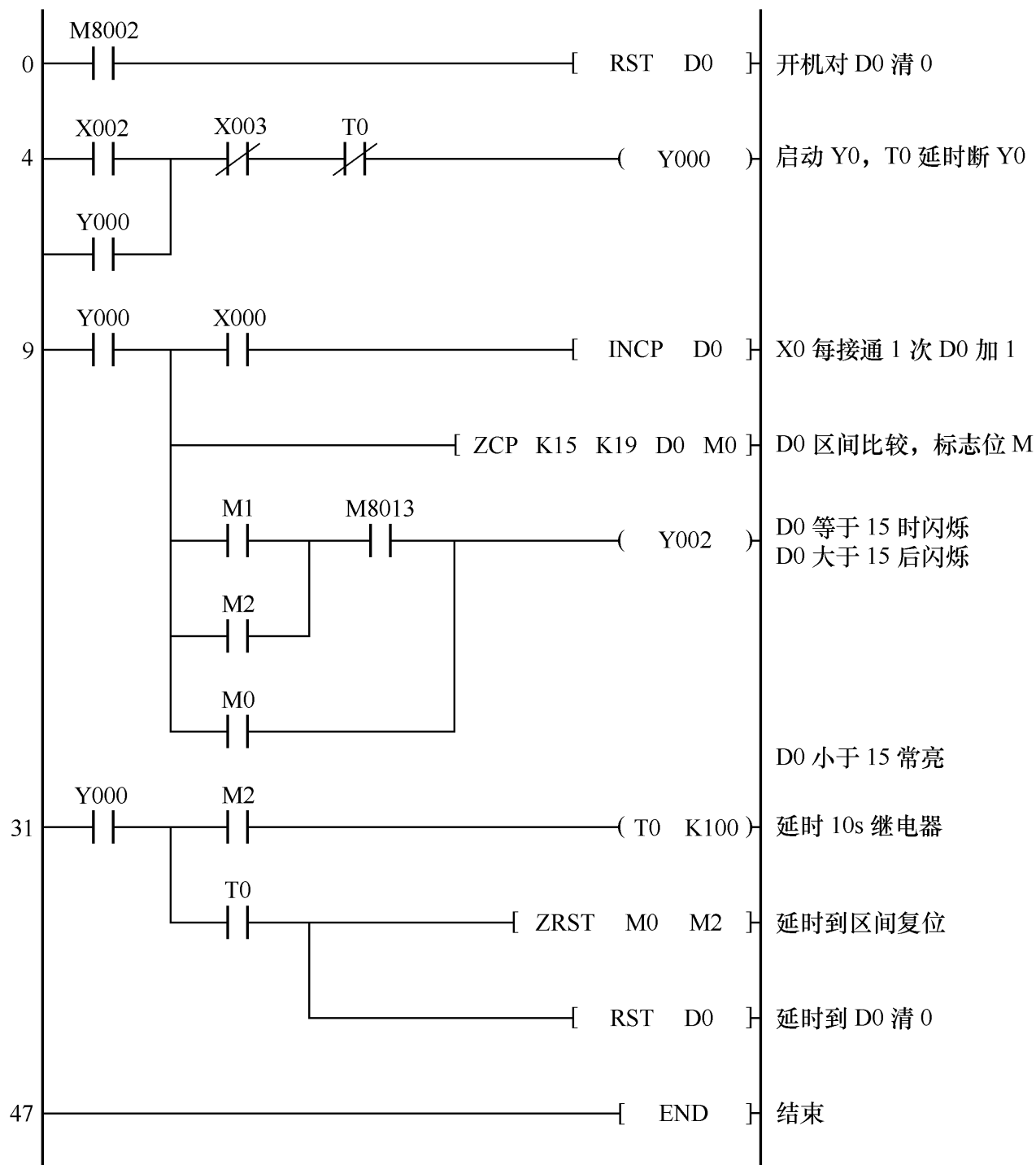
用如图所示的传送带输送工件，数量为 20 个。连接 X0 端子的光电传感器对工件进行计数。当计件数量小于 15 时，指示灯常亮；当计件数量等于或大于 15 以上时，指示灯闪烁；当计件数量为 20 时，10s 后传送带停机，同时指示灯熄灭。设计 PLC 控制线路并用区间比较指令 ZCP 编写程序。



传送带工作台



传送带的控制线路图



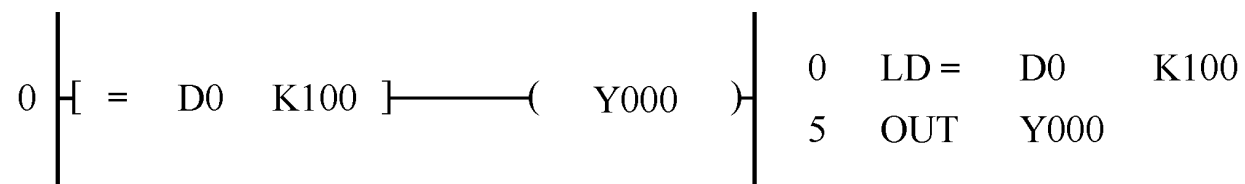
接点比较指令

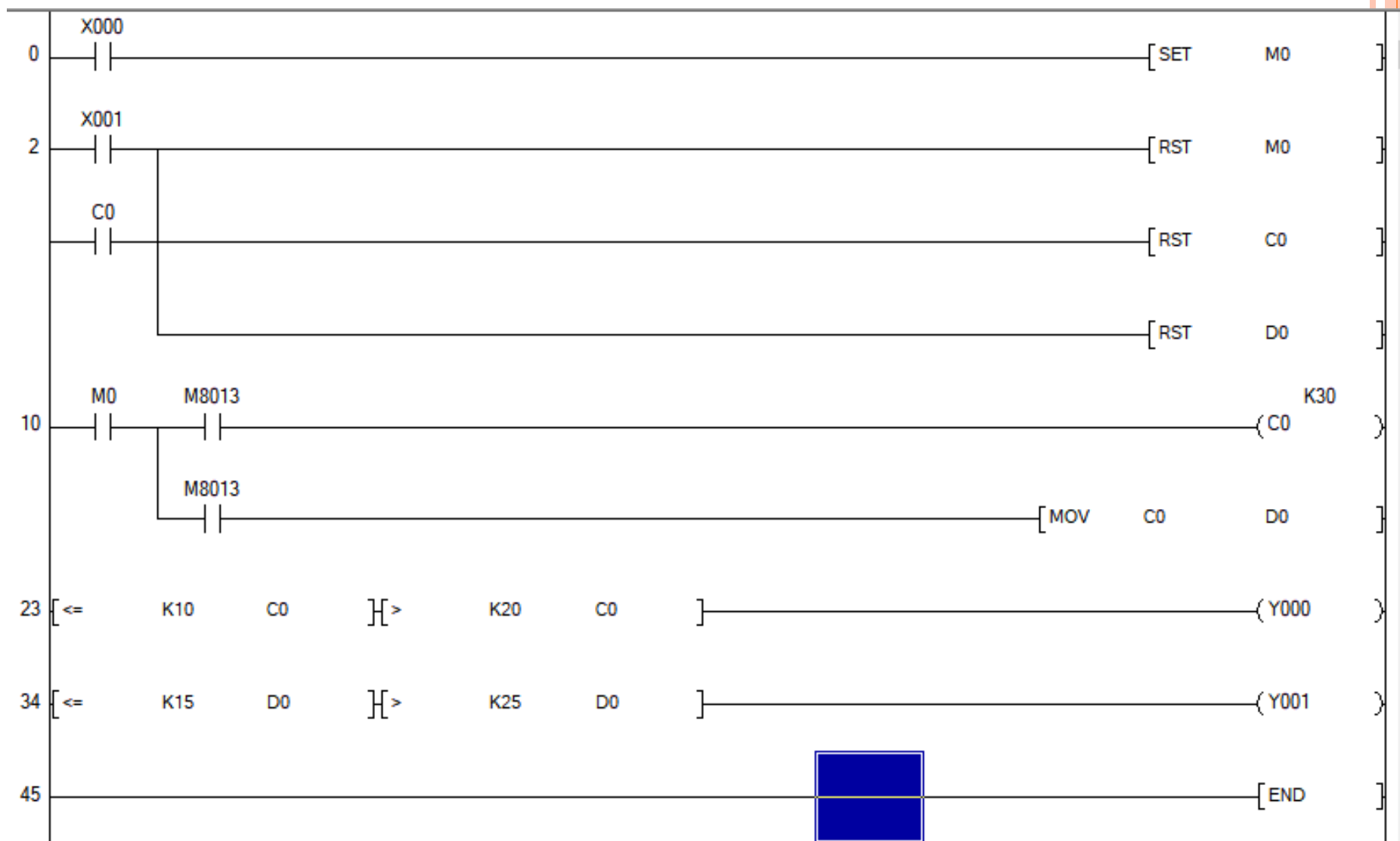
16 位数据接点比较指令表

	编号	助 记 符	比 较 条 件	逻 辑 功 能
取 比 较 接 点	224	LD=	$S1=S2$	S1 与 S2 相等
	225	LD>	$S1>S2$	S1 大于 S2
	226	LD<	$S1<S2$	S1 小于 S2
	228	LD<>	$S1\neq S2$	S1 与 S2 不相等
	229	LD<=	$S1\leq S2$	S1 小于等于 S2
	230	LD>=	$S1\geq S2$	S1 大于等于 S2



	编号	助 记 符	比 较 条 件	逻 辑 功 能
串联比较接点	232	AND=	S1=S2	S1 与 S2 相等
	233	AND >	S1>S2	S1 大于 S2
	234	AND <	S1<S2	S1 小于 S2
	236	AND <>	S1≠ S2	S1 与 S2 不相等
	237	AND <=	S1≤ S2	S1 小于等于 S2
	238	AND >=	S1≥ S2	S1 大于等于 S2
并联比较接点	240	OR=	S1=S2	S1 与 S2 相等
	241	OR>	S1>S2	S1 大于 S2
	242	OR<	S1<S2	S1 小于 S2
	244	OR<>	S1≠ S2	S1 与 S2 不相等
	245	OR<=	S1≤ S2	S1 小于等于 S2
	246	OR>=	S1≥ S2	S1 大于等于 S2





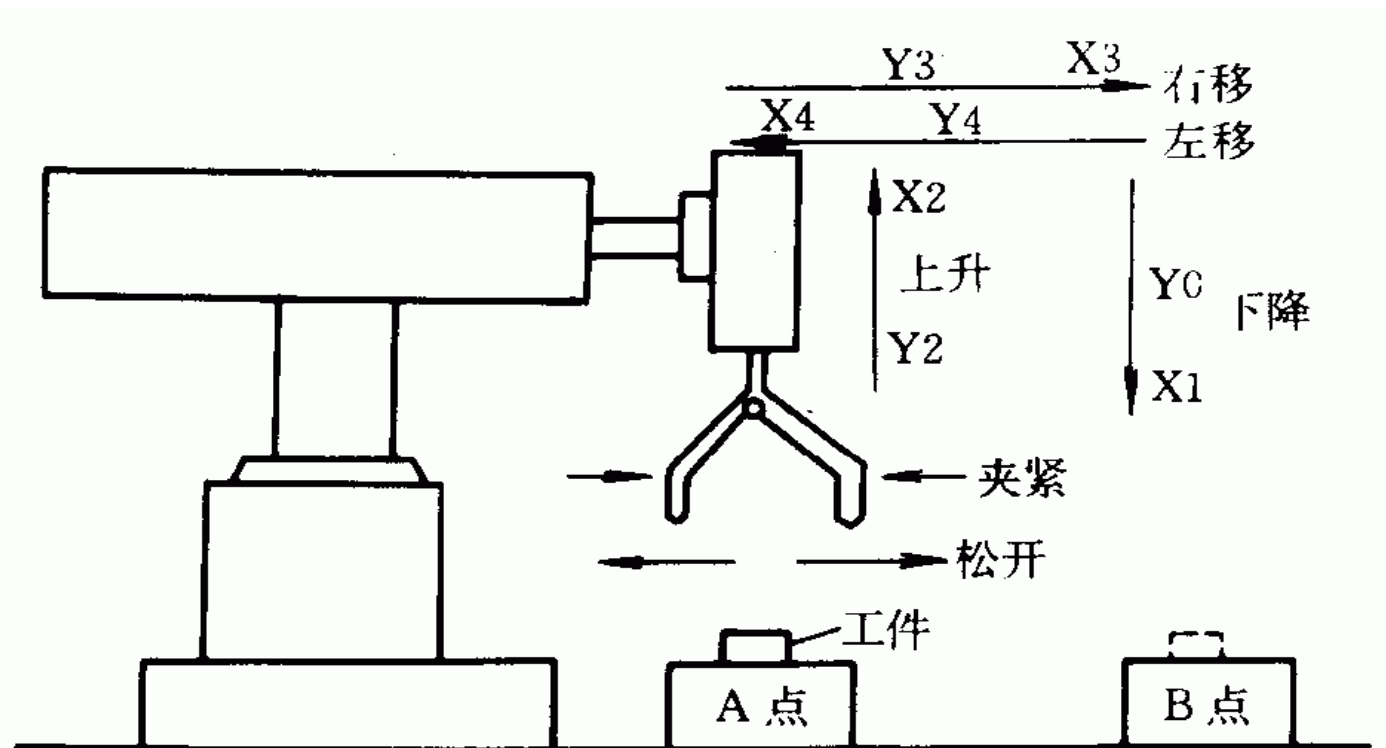
三、本次课小结

:

1. 逻辑字与、字或、字或非指令；
2. 子程序调用程序指令；
3. 比较指令；
4. 分析机械手



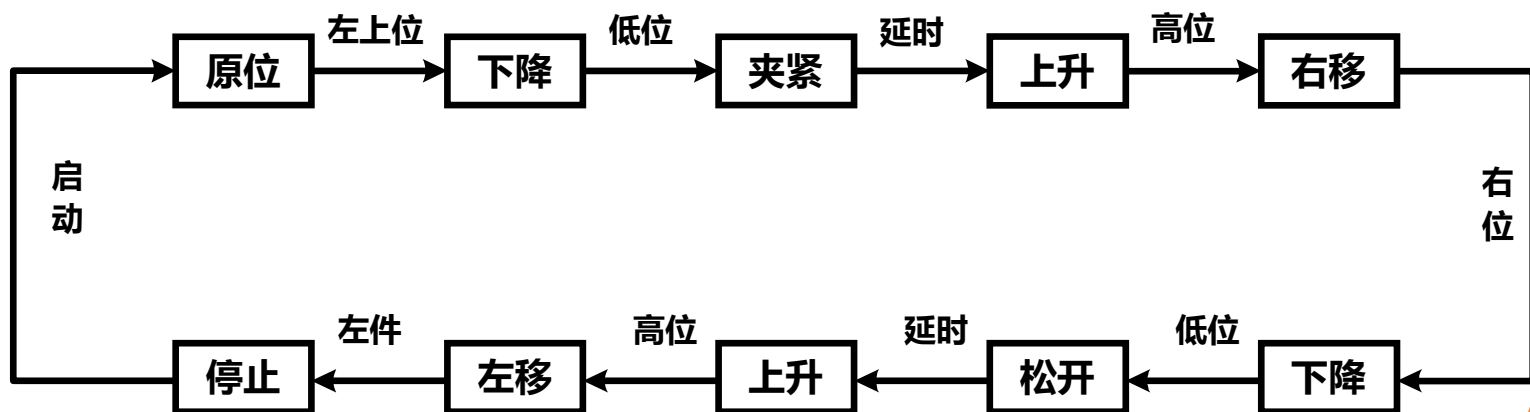
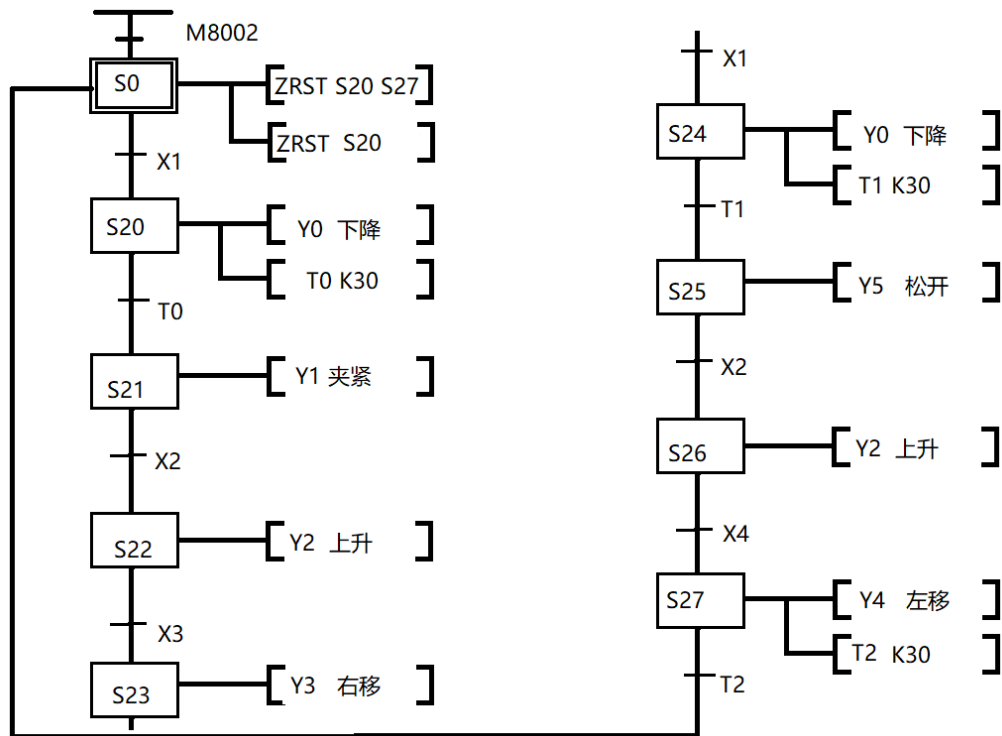
四、布置作业



提示如下

：

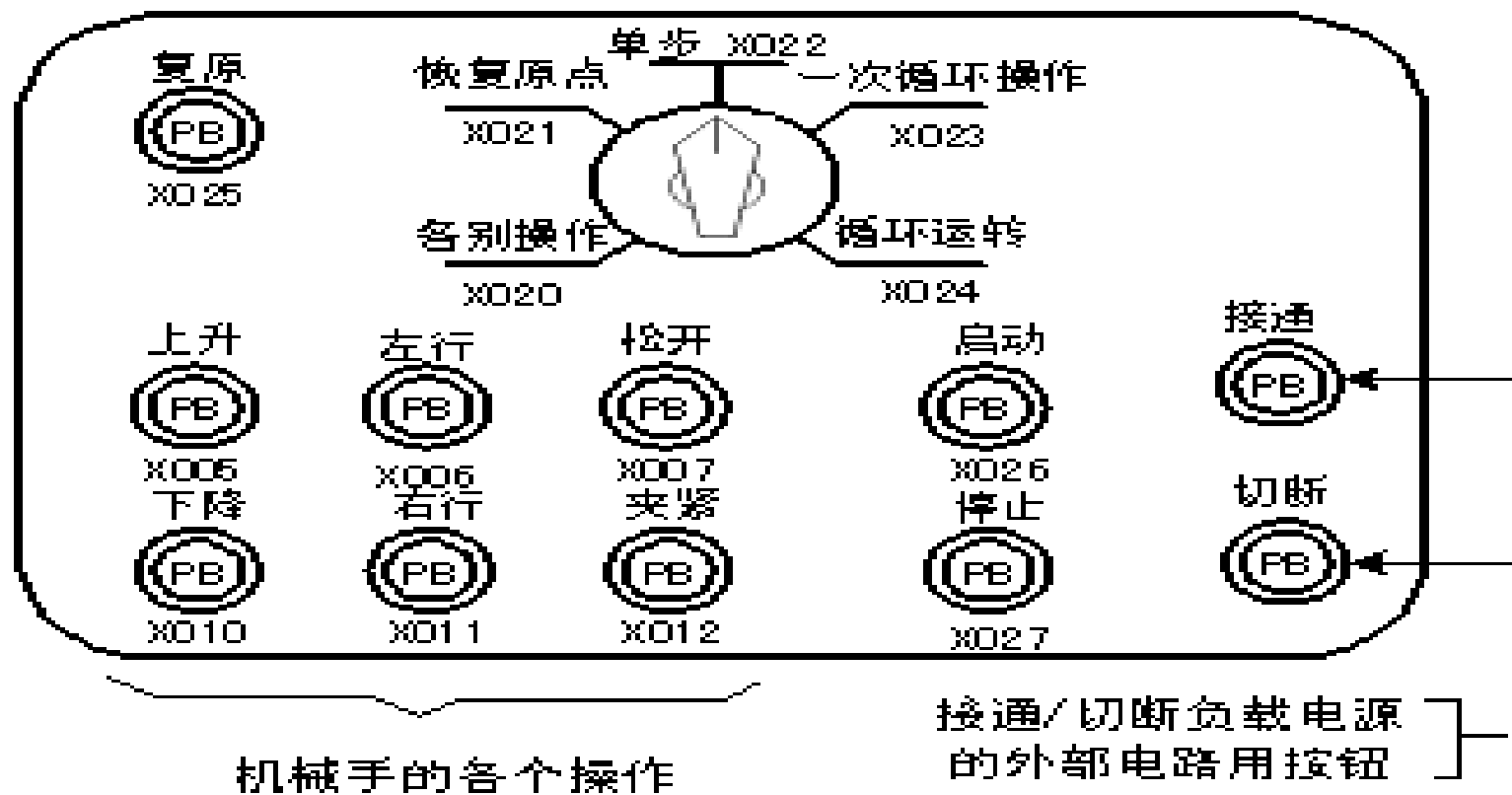




机械手控制流程图

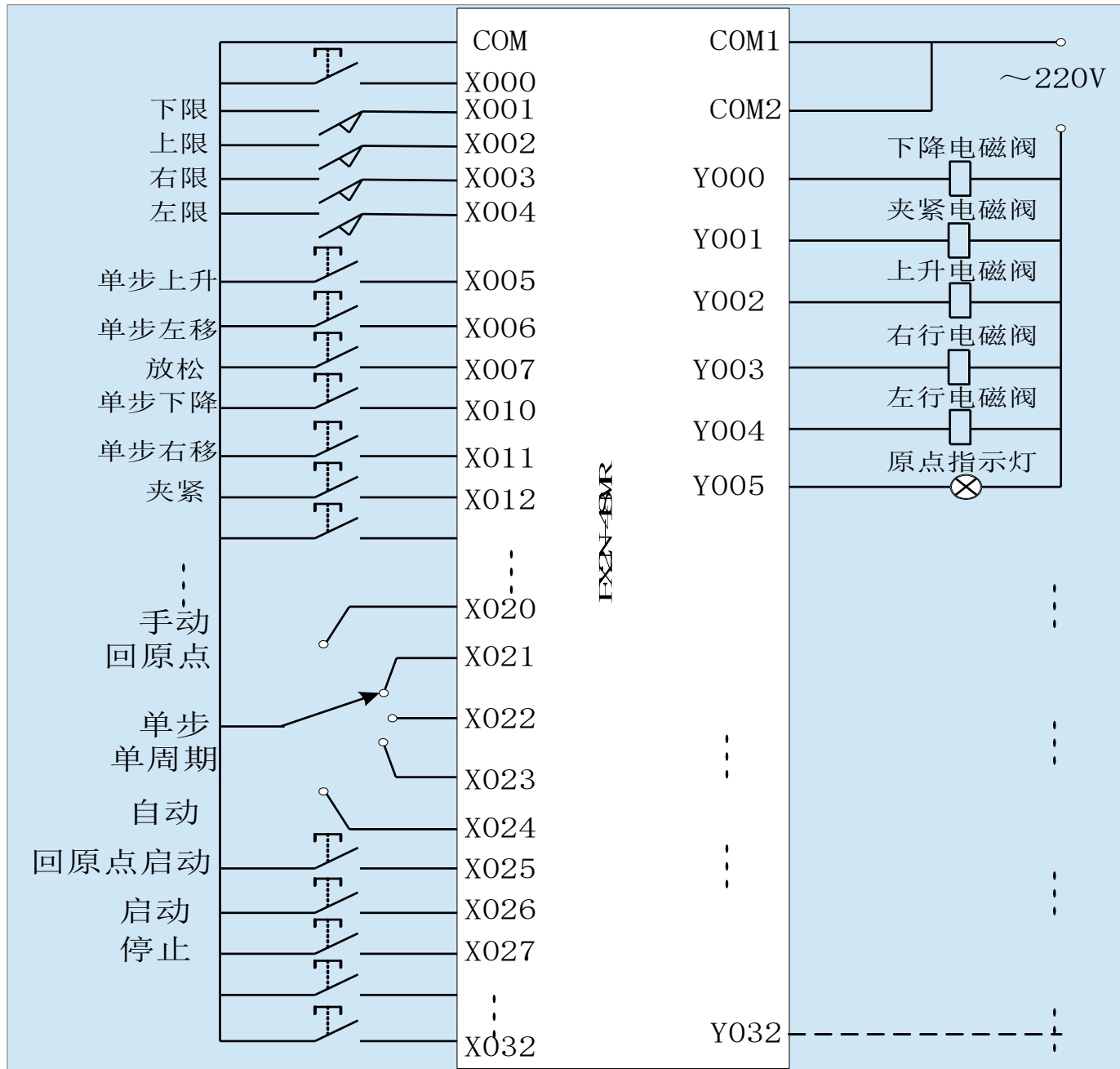
工作方式：手动、回原点、单步、一个周期和连续工作（自动）

- 手动：按钮单独操作机构上升、下降、左移、右移、放松、夹紧。供维修用。
- 回原点：按下此按钮，机械手自动回到原点。顺序控制中，自动运行要有一个起始点，这就是原点。机械手工作时应从原点位置按启动按钮。
- 单步运行：按动一次启动按钮，前进一个工步，供调试用。
- 单周期运行（半自动）：在原点位置按动启动按钮，自动运行一遍后回到原点停止。供首次检验用。若在中途按动停止按钮，则停止运行；再按启动按钮，从断点处继续运行，回到原点处自动停止。
- 自动控制：按下启动按钮，机构从原点位置开始，自动完成一个工作循环过程，并连续反复运行，若在中途按动停止按钮，运行到原点后停止。供正常工作用。



机械手操作面板





谢谢观看
!