

第3部分 三菱FX系列PLC基本指令

主要介绍三菱FX2系列PLC的20条基本逻辑指令,这20条指令功能十分强大,已经能解决一般的继电接触控制问题。

本章还重点介绍梯形图和助记符语言以及其程序设计方法。

三菱FX系列PLC的程序设计语言

三菱FX系列PLC的编程语言一般以梯形图语言为主，同时还有助记符语言、流程图语言。

一、梯形图（Ladder）

梯形图一种图形编程语言，是面向控制过程的一种“自然语言”，它沿用继电器的触点、线圈、串并联等术语和图形符号，同时也增加了一些继电器控制系统中没有的特殊符号，以便扩充PLC的控制功能。

三菱FX系列PLC的程序设计语言

从继电接触控制图到梯形图

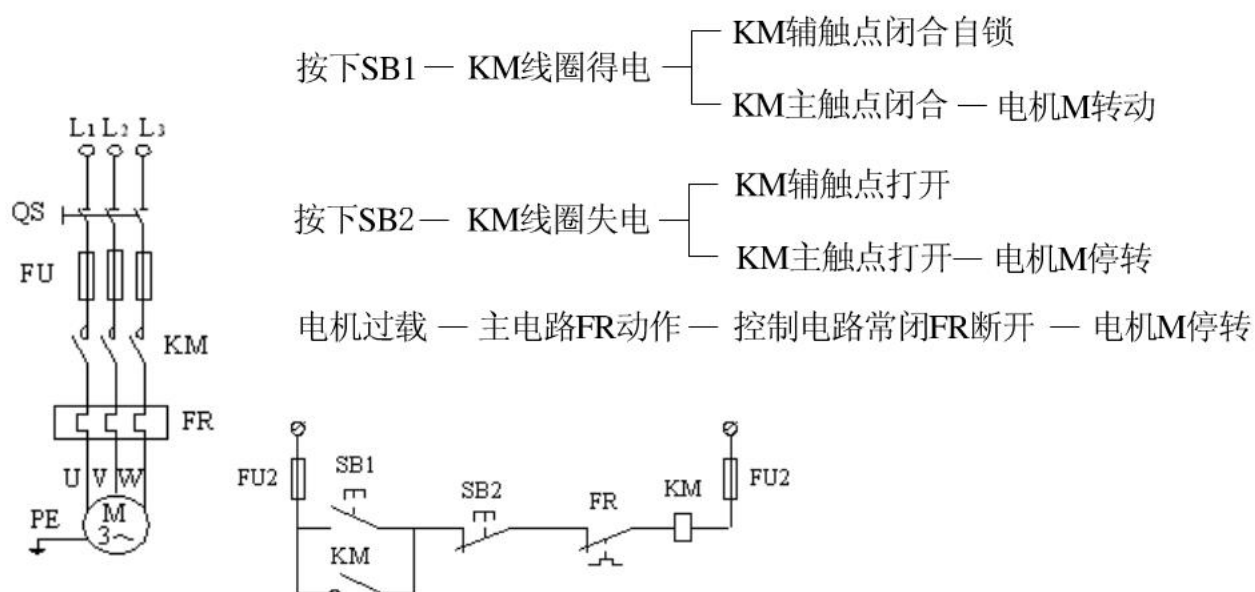


图1 电机启—保—停控制电路图

三菱FX系列PLC的程序设计语言

与图1等效PLC控制梯形图如图3。

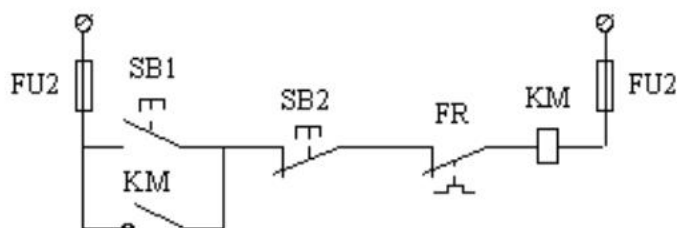


图1 电机启—保—停继电器控制电路图

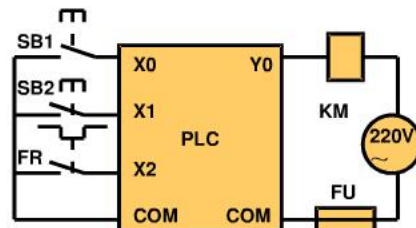


图2 PLC控制电机启—保—停连线图

表1 输入、输出点分配表

输 入			输 出		
名 称		输入点	名 称		输出点
启动按钮	SB1	X000	输出接触器	KM	Y000
停止按钮	SB2	X001			
热继电器常闭	FR	X002			

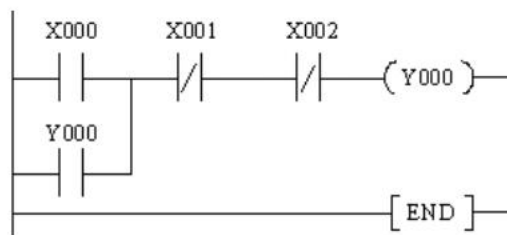


图3 电机启—保—停控制梯形图

三菱FX系列PLC的程序设计语言

1. 梯形图中的图形符号

① 对应继电器的各种符号。

表3.2 梯形图中的图元符号与继电器接触控制图中的图形符号比较

名称	梯形图中的图元符号	继电器接触控制图中的符号
常开		
常闭	 	
线圈	  	

② 其它指令符号：

三菱FX系列PLC的程序设计语言

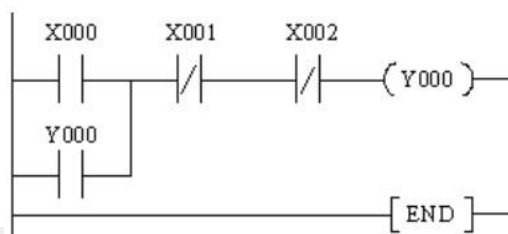
2. 梯形图的格式:

① 左边垂直线为起始母线（相当于电源正级，右边垂直线为终止母线电源负极（可省）。

③ 同名常开、常闭触点可多次使用；同名线圈只能使用一次。

② 每一逻辑行由一个或几个支路组成，左边由若干触点组成，表示控制元件；右边为线圈或其它指令，表示控制结果。

④ 最后一行以“END”指令结束。



★ PLC编程软件安装

1、安装MELSOFT环境

先点击 “EnvMEL” 文件夹 下面的 “SETUP.EXE”

2、安装 Gx Developer8.86Q

点击 “Develope” 文件夹 下面的 “SETUP.EXE”

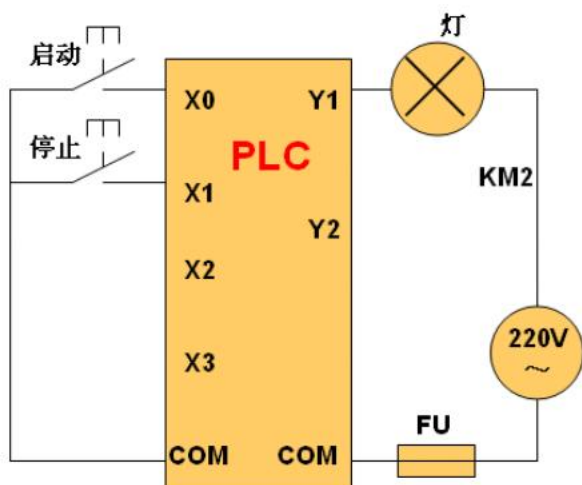
“监视专用” 那里千万不要打勾

★ PLC仿真软件安装

PLC应用练习

1、按过启动按钮后，灯亮；按过停止按钮后，灯灭。

①硬件连接图：



② I/O分配表：

输入		输出	
输入元件	输入端口	输出元件	输出端口
启动按钮	X0	灯	Y1
停止按钮	X1		

③梯形图：

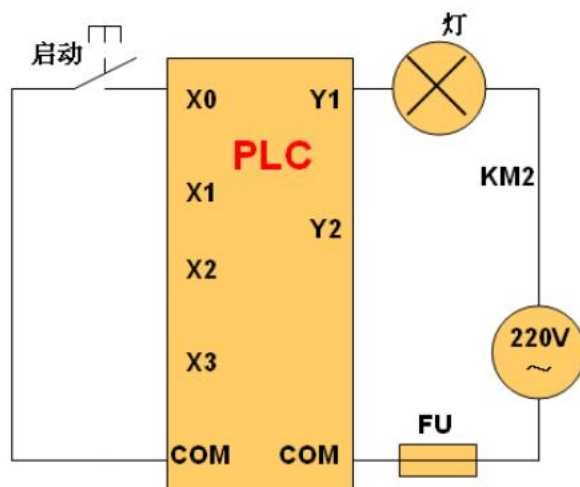
PLC应用练习

2、按过启动按钮后，灯亮；10秒后，灯灭。

① I/O分配表：

输入		输出	
输入 元件	输入 端口	输出 元件	输出 端口
启动 按钮	X0	灯	Y1

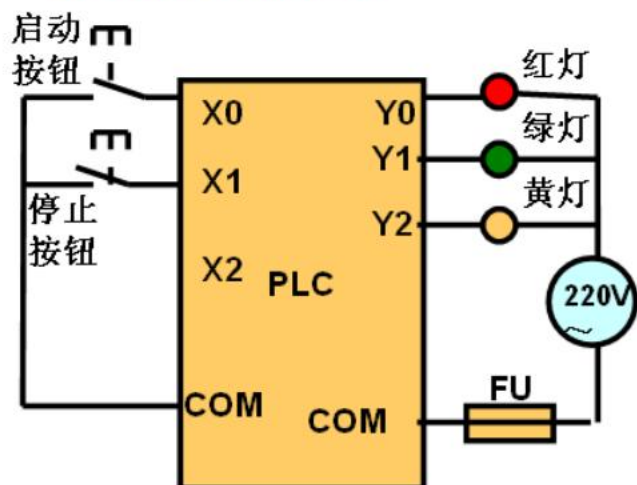
②梯形图：



3、按下启动按钮后，红灯亮**15**秒，而后绿灯亮**10**秒，而后黄灯亮**5**秒；而后红灯亮，依次反复；按下停止按钮后，所有灯都熄灭。

①硬件连接图：

②梯形图：



4、按下启动按钮后，红灯亮15秒，而后绿灯亮10秒，而后黄灯亮5秒；反复5次；按下停止按钮后，所有灯都熄灭。

①I/O分配表：

②梯形图：

输入		输出	
启动按钮	X0	红灯	Y0
停止按钮	X1	绿灯	Y1
		黄灯	Y2

二、助记符语言 (Mnemonic)

在现场调试时，小型PLC往往只配备显示屏只有几行宽度的简易编程器，这时，梯形图就无法输入了，但助记符指令却可以一条一条的输入，滚屏显示。

助记符指令组成：操作码+操作数。

操作码用便于记忆的助记符表示，用来表示指令的功能，告诉CPU要执行什么操作。

人工将图3梯形图转换成指令表方法：也是按梯形图的逻辑行和逻辑组件的编排顺序自上而下、自左向右依次进行。

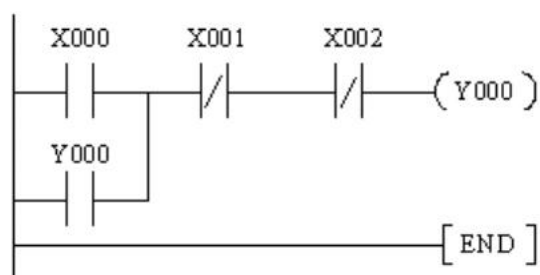


表3.4 对应图3.3梯形图的指令表

图3电机启—保—停控制梯形图

步序	操作码	操作数	说明
0	LD	X000	逻辑行开始，输入 X000 常开接点
1	OR	Y000	并联 Y000 的自保接点
2	ANI	X001	串联 X001 的常闭接点
3	ANI	X002	串联 X002 的常闭接点
4	OUT	Y000	输出 Y000 线圈
5	END		逻辑行结束

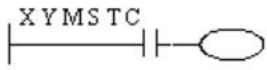
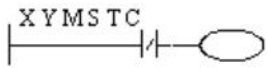
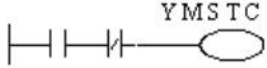
3.2 三菱FX系列PLC的基本逻辑指令 1

3.2.1 逻辑取与输出线圈驱动指令LD、LDI、OUT

1. 指令用法

- (1) **LD** (取常开)：常开接点与母线连接指令。
- (2) **LDI** (取常闭)：常闭接点与母线连接指令。
- (3) **OUT** (线圈驱动)：线圈驱动指令。

表3.4 逻辑取与输出线圈驱动指令

助记符名称	操作功能	梯形图与目标组件	程序步数
LD 取	常开接点 运算开始		1
LDI 取反	常闭接点 运算开始		1
OUT 输出	线圈驱动		YM: 1
			S、特M: 2
			T、C 16位: 3
			C 32位: 5

3.2.1 逻辑取与输出线圈驱动指令LD、LDI、OUT 2

2. 指令说明

(1) LD和LDI指令用于接点与母线相连。与ANB和ORB指令配合，还作为分支起点指令。目标组件：X、Y、M、T、C、S。

(2) OUT指令用于驱动输出继电器、辅助继电器、定时器、计数器、状态继电器和功能指令，但是不能用来驱动输入继电器，目标组件：Y、M、T、C、S和功能指令线圈F。

(3) OUT指令可以并行输出，相当于线圈是并联的，如图3.6中的M100和T1就是并联的。注意，输出线圈不能串联使用。

(4) 在对定时器、计数器使用OUT指令后，须设置时间常数K，或指定数据寄存器的地址。如图3.6中T1的

3.2.1 逻辑取与输出线圈驱动指令LD、LDI、OUT 3

时间常数设置为K10。时间常数K的设定，要占一步。表3.6中给出了时间常数K的设定值范围与对应的时间实际设定值范围，及以T、C为目时OUT指令所占步数。

例3.3 阅读图3.6中的梯形图，试解答：

- (1) 写出图3.6中梯形图所对应的指令表。
- (2) 指出各指令的步序并计算程序的总步数。
- (3) 计算定时器T1的定时时间。

表3.5 定时器/计数器时间常数K的设定

定时器/计数器	时间常数 K 范围	实际设定值范围	步数
1ms	1~32767	0.001~32.767s	3
10ms		0.01~327.67s	3
100ms		0.1~3276.7s	3
16 位计数器	1~32767	1~32767	3
32 位计数器	-2147483648~+2147483647	-2147483648~+2147483647	5

3.2.1 逻辑取与输出线圈驱动指令LD、LDI、OUT 4

时间常数设置为K10。时间常数K的设定，要占一步。表3.6中给出了时间常数K的设定值范围与对应的时间实际设定值范围，及以T、C为目时OUT指令所占步数。

例3.3 阅读图3.7中的梯形图，试解答：

- (1) 写出图3.7中梯形图所对应的指令表。
- (2) 指出各指令的步序并计算程序的总步数。
- (3) 计算定时器T1的定时时间。

解：

(1) 从梯形图到指令表，按自上而下、自左向右依次进行转换，得到对应图3.7梯形图的指令表如表3.7所示。

(2) 总的程序步为10步。各指令的步序如表3.7第1列所示。

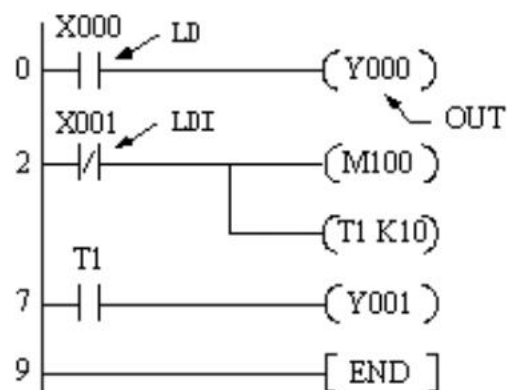


图3.7 LD、LDI和OUT指令应用举例

3.2.1 逻辑取与输出线圈驱动指令LD、LDI、OUT 5

(3) 由附录中的表A.1可知T1是100ms定时器, 所以T1定时时间为 $10 \times 0.1 = 1\text{s}$ 。

表3.6 对应图3.7梯形图的指令表

步序	操作码	操作数	说明
0	LD	X000	将 X000 常开接点与母线相连
1	OUT	Y000	驱动线圈 Y000
2	LDI	X001	将 X001 的常闭接点与母线相连
3	OUT	M100	驱动线圈 M100
4	OUT	T1	驱动定时器线圈 T1
		K10	设定定时时间 1s
7	LD	T1	将 T1 常开接点与母线相连
8	OUT	Y001	驱动线圈 Y001
9	END		逻辑行结束

3.2.2 接点串联指令AND、ANI 1

1. 指令用法

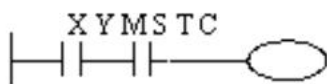
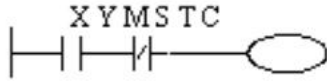
(1) **AND** (串常开)：常开接点串联指令。

(2) **ANI** (串常闭)：常闭接点串联指令。

2. 指令说明

(1) **AND**和**ANI**指令用于单个接点串联，串联接点的数量不限，重复使用指令次数不限。目为X、Y、M、T、C、S。

表3.7 接点串联指令

助记符名称	操作功能	梯形图与目标组件	程序步数
AND 与	常开接点 串联连接		1
ANI 与非	常闭接点 串联连接		1

3.2.2 接点串联指令AND、ANI 2

(2) 在执行OUT指令后，通过接点对其它线圈执行OUT指令，称为“连续输出”（又称纵接输出）。

正确：图3.8中紧接OUT M101后，通过接点T1输出OUT Y001。

错误：图3.9中M101与T1和Y001交换，出错。非要这样纵接，要使用后述的MPS和MPP指令。

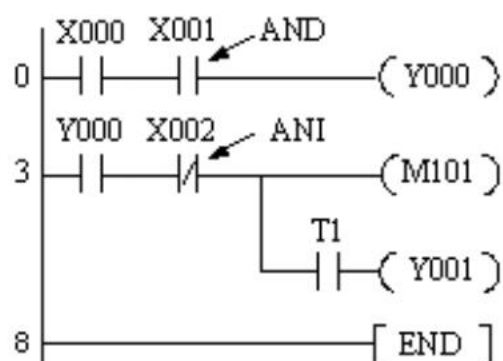


图3.8 AND与ANI指令应用举例

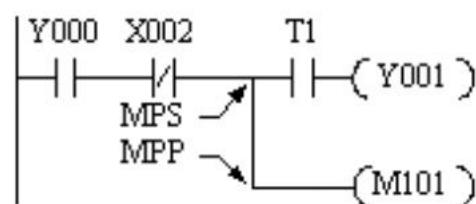


图3.9 纵接错误举例

3.2.2 接点串联指令AND、ANI 3

例3.4 阅读图3.8中的梯形图，试解答：

- (1) 写出图3.8梯形图所对应的指令表。
- (2) 指出各指令的步序并计算程序的总步数。

解：(1) 对应图3.8梯形图的指令表如表3.9所示。

- (2) 各指令步序如表3.9。程序总的占9步。

表3.8 对应图3.8梯形图的指令表

步序	操作码	操作数	说明
0	LD	X000	将 X000 常开接点与母线相连
1	AND	X001	串联常开接点 X001
2	OUT	Y000	驱动线圈 Y000
3	LD	Y000	将 Y000 常开接点与母线相连
4	ANI	X002	串联常闭接点 X002
5	OUT	M101	驱动线圈 M101
6	AND	T1	串联常开接点 T1
7	OUT	Y001	驱动线圈 Y001
8	END		逻辑行结束

3.2.3 接点并联指令OR、ORI 1

1. 指令用法

(1) **OR** (并常开)：常开接点并联指令

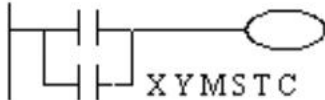
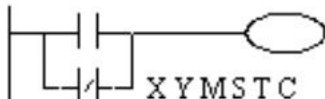
(2) **ORI** (并常闭)：常闭接点并联指令。

例3.5 阅读图3.10 (a) 中的梯形图，试解答：

(1) 写出图3.10 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

表3.9 接点并联指令

助记符名称	操作功能	梯形图与目标组件	程序步数
OR 或	常开接点 并联连接	 XYMSTC	1
ORI 或非	常闭接点 并联连接	 XYMSTC	1

3.2.3 接点并联指令OR、ORI 2

解：（1）对应图3.10梯形图的指令表如图3.10（b）所示。

（2）各指令步序也如图3.10（b），各指令均为1步，所以程序总的占10步。

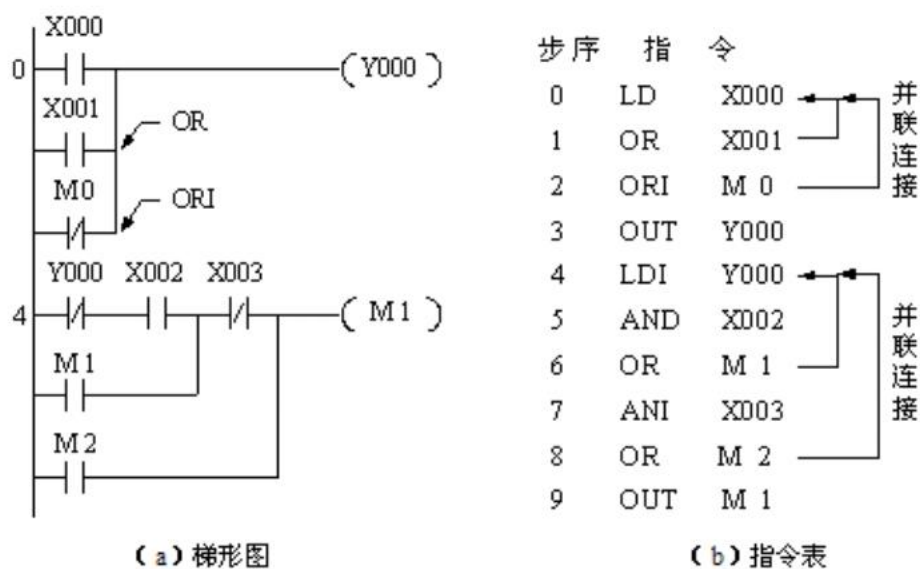


图3.10 OR与ORI指令举例

3.2.3 接点并联指令**OR**、**ORI** 3

2. 指令说明

(1) **OR**和**ORI**指令引起的并联，是从**OR**和**ORI**一直并联到前面最近的**LD**和**LDI**指令上，如图3.10 (a)，并联的数量不受限制。操作目标组件为**X**、**Y**、**M**、**T**、**C**、**S**。

(2) **OR**和**ORI**指令只能用于单个接点并联连接，若要将两个以上接点串联而成的电路块并联，要用后述的**ORB**指令。

3.2.4 串联电路块的并联指令ORB

1. 指令用法

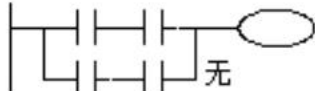
ORB（串联电路块）：将两个或两个以上串联块并联连接的指令。串联块：两个以上接点串联的电路。串联块并联，支路始端用LD和LDI，终端用ORB指令。

2. 指令说明

（1）ORB指令无操作数，其后不跟任何软组件编号。

（2）多重并联电路中，ORB指令可以集中起来使用；
切记：在一条线上LD和LDI指令重复使用次数要 ≤ 8 。

表3.10 串联电路块的并联指令

助记符名称	操作功能	梯形图与目标组件	程序步数
ORB 块或	串联电路块 的并联连接		1

3.2.4 串联电路块的并联指令ORB 2

例3.5 阅读图3.11 (a) 中的梯形图，试解答：

- (1) 写出图3.11 (a) 梯形图所对应的指令表。
- (2) 指出各指令的步序并计算程序的总步数。

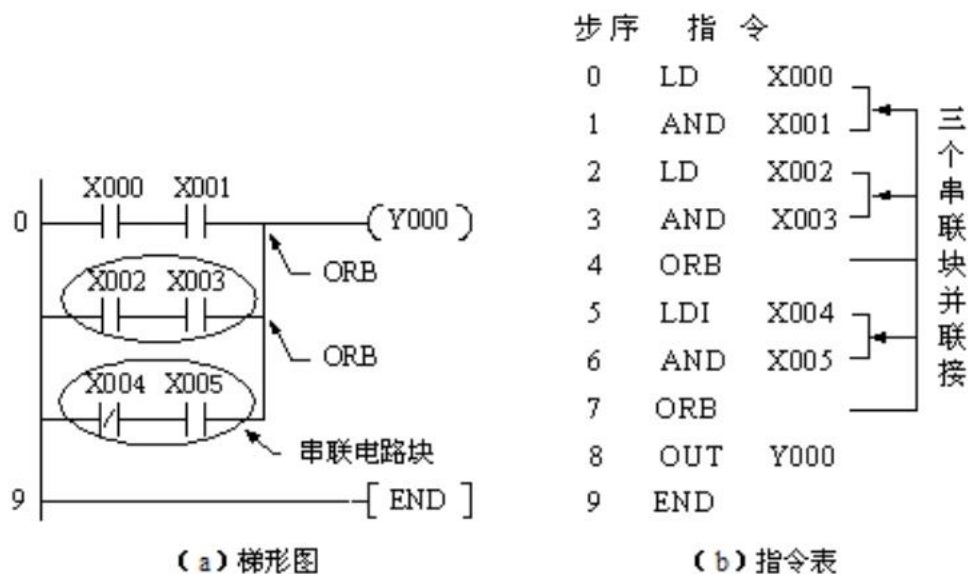


图3.11 ORB指令举例

3.2.4 串联电路块的并联指令ORB 3

解：

（1）对应图3.11（a）梯形图的指令表如图3.11（b）所示。按照两两并联的原则，在首次出现的两个串联块后应加一个ORB指令，此后每出现一个要并联的串联块，就要加一个ORB指令。

（2）各指令步序也如图3.11（b），各指令均为1步，所以程序总的占10步。

3.2.5 并联电路块的串联指令ANB 1

1. 指令用法

ANB（并联电路块）：将并联电路块的始端与前一个电路串联连接的指令。

并联块：两个以上接点并联的电路。

并联块串联时要用ANB指令，支路始端用LD和LDI，终端用ANB指令。

3.2.5 并联电路块的串联指令ANB 2

2. 指令说明

(1) **ANB**指令无操作数，其后不跟任何软组件编号。

(2) **ANB**指令可以集中起来使用，但是切记，此时在一条线上LD和LDI指令重复使用次数要 ≤ 8 。

例3.6 阅读图3.12 (a) 中的梯形图，试解答：

(1) 写出图3.12 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

解：(1) 对应图3.12 (a) 梯形图的指令表如图3.12 (b)。按两两串联原则，在首次出现的两并联块后应加一个**ANB**指令，

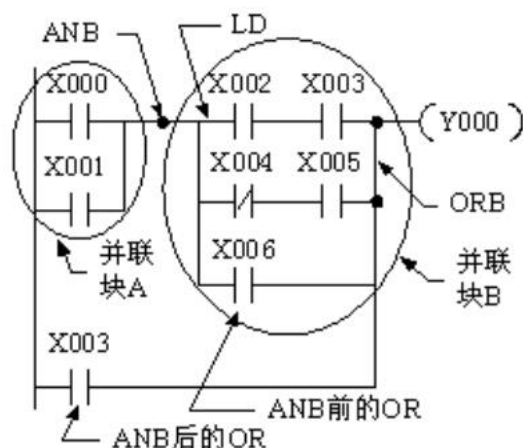
表3.11 并联电路块的串联指令

助记符名称	操作功能	梯形图与目标组件	程序步数
ANB 块与	并联电路块 的串联连接	 无	1

3.2.5 并联电路块的串联指令ANB 3

此后每出现一个并联块，就要加一个ANB。前一并联块结束时，应用LD或LDI指令开始后一并联块。

(2) 各指令步序也如图3.12 (b)，各指令均为1步，所以程序总的占11步。



(a) 梯形图

步序	指令
0	LD X000
1	OR X001
2	LD X002
3	AND X003
4	LDI X004
5	AND X005
6	ORB
7	OR X006
8	ANB
9	OR X003
10	OUT Y000

(b) 指令表

图3.12 ANB指令举例

3.2.6 多重输出指令MPS、MRD、MPP 1

1. 指令用法

(1) **MPS** (进栈)：进栈指令。

(2) **MRD** (读栈)：读栈指令。

(3) **MPP** (出栈)：出栈指令。

这组指令可将接点的状态先进栈保护，当需要接点状态时，再出栈恢复，以保证与后面的电路正确连接。

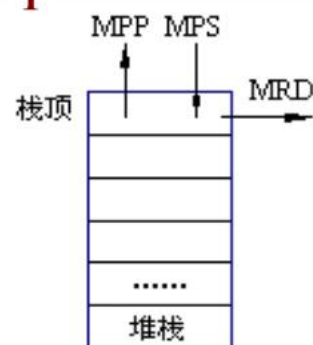


图3.13 栈操作示意

表3.12 多重输出指令

助记符名称	操作功能	梯形图与目标组件	程序步数
MPS 进栈	进栈		1
MRD 读栈	读栈		1
MPP 出栈	出栈		1

3.2.6 多重输出指令MPS、MRD、MPP 2

2. 指令说明

(1) PLC中，有11个可存储中间运算结果的存储器，它们相当于微机中的堆栈，是按照先进后出的原则进行存取的一段存储器区域。堆栈指令的操作如图3.13。

(2) 使用一次MPS指令，该时刻的运算结果就压入第一个单元中（栈顶）。再次使用MPS，当前结果压入栈顶，原先数据依次向栈的下一个单元推移。

(3) 使用MPP指令，各数据依次向上一个栈单元传送。栈顶数据在弹出后就从栈内消失。

(4) MRD是栈顶数据的读出专用指令，但栈内的数据不发生下压或上托的传送。

(5) MPS、MRD、MPP指令均无操作数。

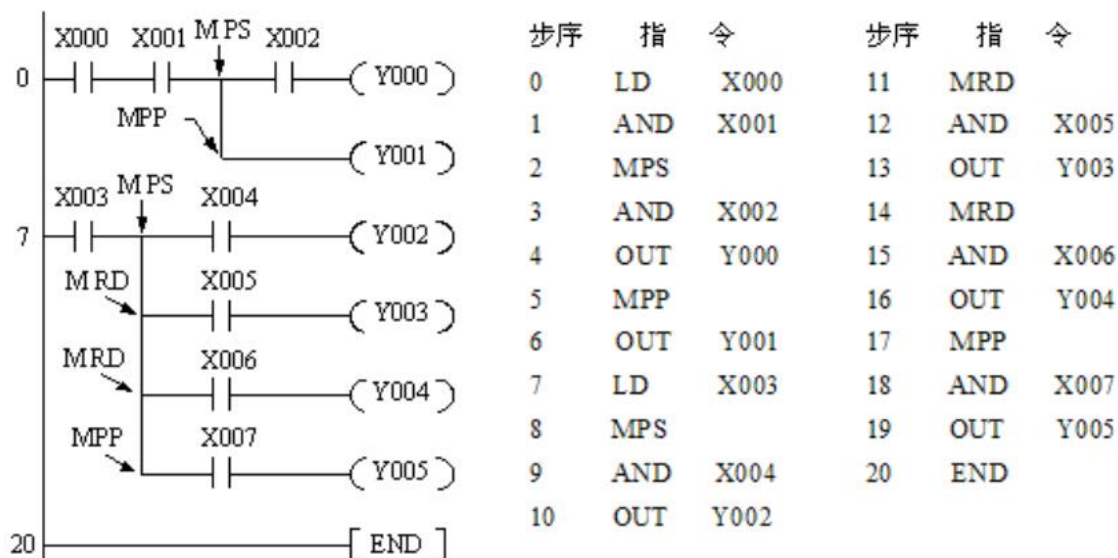
(6) MPS和MPP应配对使用，连续使用次数 ≤ 11 次。

3.2.6 多重输出指令MPS、MRD、MPP 3

例3.7 阅读图3.14 (a) 中一层堆栈的梯形图，试解答：

(1) 写出图3.14 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。



(a) 梯形图

(b) 指令表

图3.14 例3.7多重输出指令举例

3.2.6 多重输出指令MPS、MRD、MPP 4

解:

(1) 对应图3.14 (a) 梯形图的指令表如图3.14 (b)。注意, 栈操作指令在梯形图中并非显式可见的, 需要人工将它们加在指令表中。为了减少出错, 可用FXGPC软件先画好梯形图, 然后再将梯形图转换为指令。

(2) 用FXGP先画好梯形图, 然后用工具__转换命令, 即可得到图3.14 (b) 所示的指令表。各指令的步序已经在此程序中标出, 并可得到总的程序步为21步。

例3.8 阅读图3.15 (a) 中二层堆栈的梯形图, 试解答:

(1) 写出图3.15 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

解: (1) 用FXGP先画好梯形图, 然后用工具__转换命令, 即可得到对应图3.15 (a) 梯形图的指令表如图3.15 (b) 所示。

(2) 各指令的步序已经在此程序中标出, 并可得到总的程序步为18步。

3.2.6 多重输出指令MPS、MRD、MPP 5

(1) 对应图3.14 (a) 梯形图的指令表如图3.14 (b)。注意，栈操作指令在梯形图中并非显式可见，要人工将其加在指令表中。

(2) 用FXGP先画好梯形图，然后用工具__转换命令，即可得到图3.14 (b) 所示的指令表。各指令的步序已经在此程序中标出，并可得到总的程序步为21步。



(a) 梯形图

(b) 指令表

图3.15 例3.9多重输出指令举例

3.2.7 置位与复位指令SET、RST 1

1. 指令用法

(1) **SET** (置位)：置位指令

(2) **RST** (复位)：复位指令

用于各继电器Y、S和M等，置位和复位，还可在用户程序的任何地方对某个状态或事件设置或清除标志。

2. 指令说明

表3.13 置位与复位指令

助记符名称	操作功能	梯形图与目标组件	程序步数
SET 置位	线圈得电保持		YM: 1
			S 特 M: 2
RST 复位	线圈失电保持		STC: 2
			DVZ 特 D: 3

3.2.7 置位与复位指令SET、RST 2

(1) SET和 RST指令有自保功能，在图3.16 (a) 中，X000一旦接通，即使再断开，Y000仍保持接通。

(2) SET和 RST指令的使用没有顺序限制，并且 SET和 RST之间可以插入别的程序，但只在最后执行的一条才有效。

(3) RST指令的目标组件，除与SET相同的YMS外，还有TCD。

例3.10阅读图3.16 (a) 梯形图，试解答：

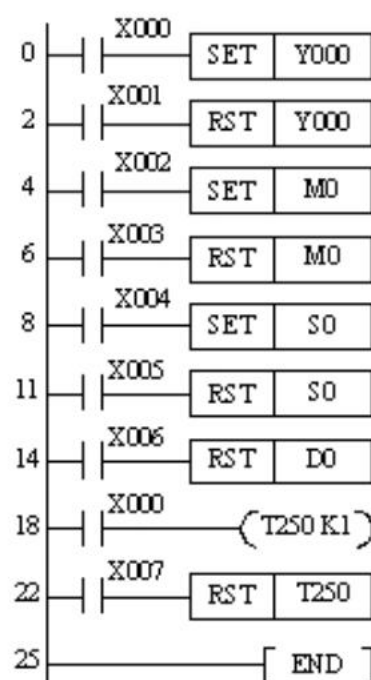
(1) 写出图3.16 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

(3) X000和X001的波形如图3.17 (a)，画出Y000的波形图。

解：

3.2.7 置位与复位指令SET、RST 3



(a) 梯形图

步序	指令	步序	指令
0	LD X000	11	LD X005
1	SET Y000	12	RST S0 ← 2步
2	LD X001	14	LD X006
3	RST Y000	15	RST D0 ← 3步
4	LD X002	18	LD X000
5	SET M0	19	OUT T250
6	LD X003		← 3步
7	RST M0	22	LD X007
8	LD X004	23	RST T250 ← 2步
9	SET S0 ← 2步	25	END

(b) 指令表

图3.16 SET和RST指令举例

3.2.7 置位与复位指令SET、RST 4

(1) 用FXGP先画好梯形图，然后用工具__转换命令，即可得到图3.16 (b) 所示的指令表。各指令的步序已经在此程序中标出，并可得到总的程序步为21步。

(2) 各指令的步序已经在此程序中标出，并可得到总的程序步为26步。若人工计算，要注意图3.16 (b) 中步序15 RST D0，此指令为3个程序步。

(3) 根据SET和 RST指令功能，容易分析得出：常开X000接通时，线圈Y000得电并保持，一直至常开X001接通时，线圈Y000才失电并保持，所以Y000的波形如图3.17 (b) 所示。

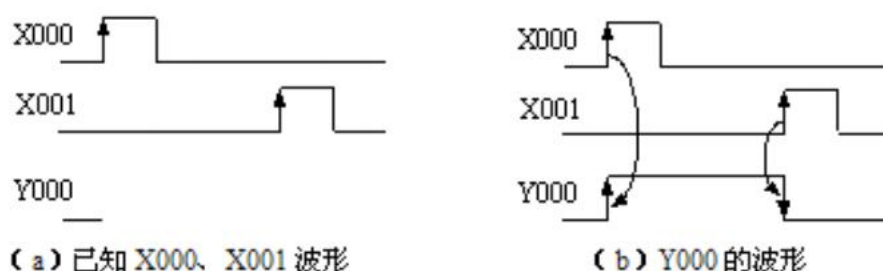


图3.17 输入/输出波形

3.2.8 脉冲输出指令PLS、PLF 1

1. 指令用法

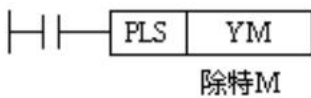
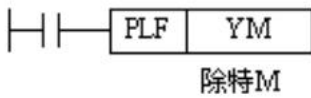
(1) **PLS** (脉冲)：微分输出指令，上升沿有效。

(2) **PLF** (脉冲)：微分输出指令，下降沿有效。

指令用于目标组件的脉冲输出，当输入信号跳变时产生一个宽度为扫描周期的脉冲。

2. 指令说明

表3.14 脉冲输出指令

助记符名称	操作功能	梯形图与目标组件	程序步数
PLS (升)	微分输出 上升沿有效		2
PLF (降)	微分输出 下降沿有效		2

3.2.8 脉冲输出指令PLS、PLF 2

(1) 使用PLS/PLF指令，组件Y、M仅在驱动输入接通/断开后一个扫描周期内动作。

(2) 特殊继电器M不能用作PLS或PLF的目标组件。

例3.11 阅读图3.18 (a) 梯形图，试解答：

(1) 写出图3.18 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

(3) X000和X001的波形如图3.19 (a) 所示，画出M0、M1和Y000的波形图。

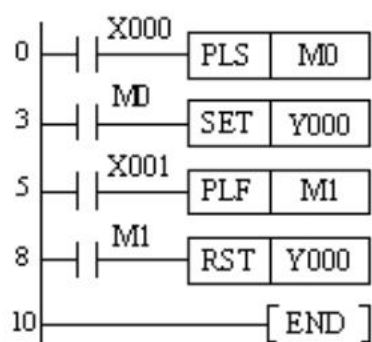
解：

(1) 用FXGP先画好梯形图，再用工具__转换命令，即可得到图3.18 (b) 所示的指令表。

(2) 各指令步序已在图3.18 (b) 程序中标出，总程序步为11步

(3) X000接通上升沿__M0线圈得电并保持一个扫描周期__M0常开闭合使Y000得电__X001接通下降沿__M1线圈得电并保持一个扫描周期，M1常开闭合使Y000复位。

3.2.8 脉冲输出指令PLS、PLF 3

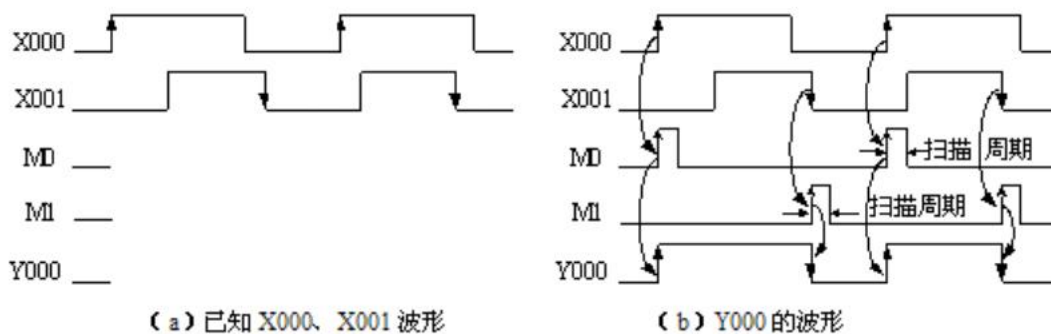


(a) 梯形图

步序	指令	步序	指令
0	LD X000	8	LD M1
1	PLS M0 ← 2步	9	RST Y000
3	LD M0	10	END
4	SET Y000		
5	LD X001		
6	PLF M1 ← 2步		

(b) 指令表

图3.18 PLS和PLF指令举例



(a) 已知 X000、X001 波形

(b) Y000 的波形

图3.19 输入/输出波形

3.2.9 主控与主控复位指令 MC、MCR 1

1. 指令用法

(1) **MC** (主控): 公共串联接点的连接指令 (公共串联接点另起新母线)。

(2) **MCR** (主控复位): MC指令的复位指令。
这两个指令分别设置主控电路块的起点和终点。

2. 指令说明

(1) 在图3.20 (a) 中, 当输入X000接通时, 执行MC

表3.15 主控与主控复位指令

助记符名称	操作功能	梯形图与目标组件	程序步数
MC 主控	公共串联接点 另起新母线	 N嵌套数:N0-N7	3
MCR 主控复位	公共串联接点 新母线解除	 N嵌套数:N0-N7	2

3.2.9 主控与主控复位指令 MC、MCR 2

与MCR之间的指令。当输入断开时，MC与MCR指令间各组件将为如下状态：计数器、累计定时器，用SET/RST指令驱动的组件，将保持当前的状态；非累计定时器及用OUT指令驱动的软组件，将处断开状态。

（2）执行MC指令后，母线（LD，LDI）移至MC接点，要返回原母线，用返回指令MCR。MC/MCR指令必须成对使用。

（3）使用不同的Y，M组件号，可多次使用MC指令。但是若使用同一软组件号，会出现双线圈输出。

（4）MC指令可嵌套使用，即在MC指令内再使用MC指令，此时嵌套级的编号就顺次由小增大。用MCR指令逐级返回时，嵌套级的编号则顺次由大减小，如图3.22（a）所示。嵌套最大不要超过8级(N7)。

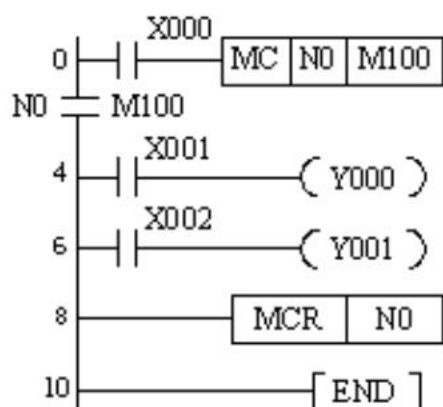
3.2.9 主控与主控复位指令 MC、MCR 3

例3.12 阅读图3.20 (a) 梯形图，试解答：

(1) 写出图3.20 (a) 梯形图所对应的指令表。

(2) 指出各指令的步序并计算程序的总步数。

解：(1) 用FXGP先画好图3.20 (a) 梯形图（串联在母线上的接点M100（嵌套级为N0）可以不必画），再用工具__转换命令



(a) 梯形图

步序 指令

0	LD	X000	↖ 3步
1	MC	NO	M100
4	LD	X001	
5	OUT	Y000	
6	LD	X002	
7	OUT	Y001	
8	MCR	NO	↖ 2步
10	END		

(b) 指令表

图3.20 MC和MCR指令举例

3.2.9 主控与主控复位指令 MC、MCR 4

梯形图将变为图3.21所示；同时可得到对应图3.20（a）梯形图的指令表如图3.20（b）所示。

（2）各指令的步序已经在图3.20（b）程序中标出，并可得到总的程序步为11步。注意图3.20（b）中两条主控指令：1 MC N0 M100和 8 MCR N0 分别为3个和2个程序步。

例3.13 分析图3.22（a）梯形图，指出主控嵌套级数，并且简述程序的执行过程。

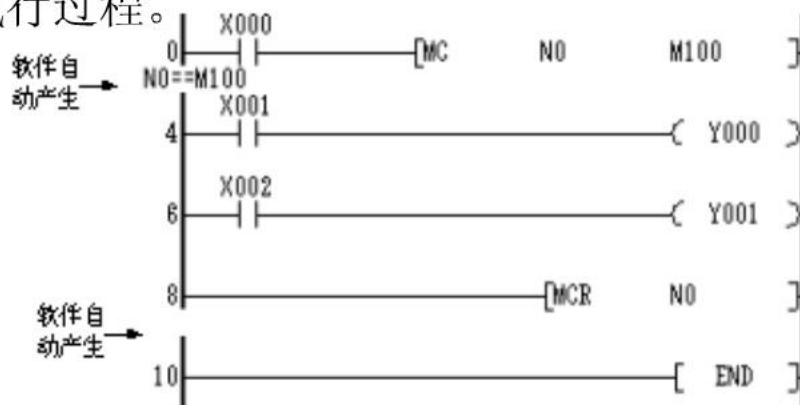
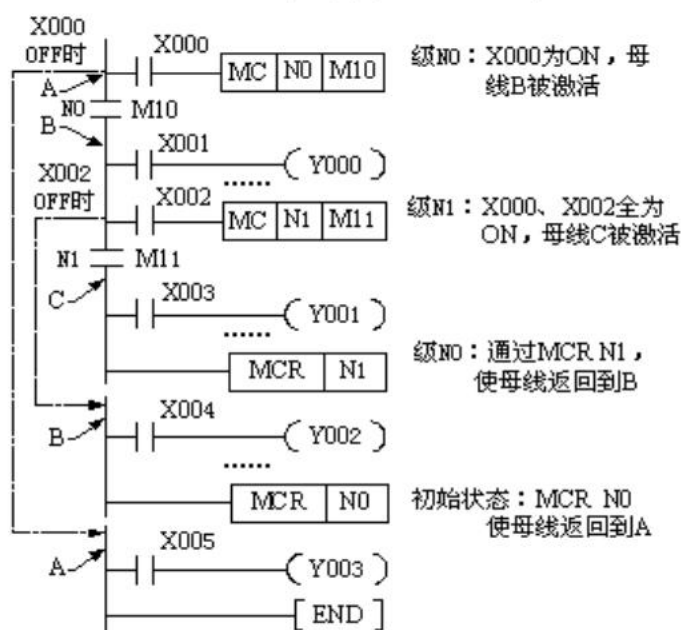


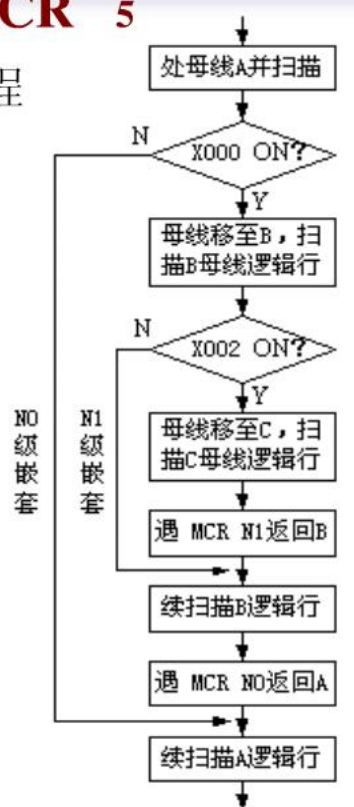
图3.21 MC和MCR指令举例梯形图

3.2.9 主控与主控复位指令 MC、MCR 5

解：图3.22 (a) 为2级主控嵌套，执行过程如图3.22 (b)。N1嵌套在N0之中。



(a) 2级主控嵌套示例梯形图



(b) 2级主控嵌套执行流程

图3.21 MC和MCR指令举例梯形图

3.2.10 空操作与程序结束指令NOP、END 1

1. 指令用法

(1) **NOP** (空操作): 空一条指令 (想删除一指令)

(2) **END** (程序结束): 程序结束指令。

调试中恰当使用NOP和END, 会带来许多方便。

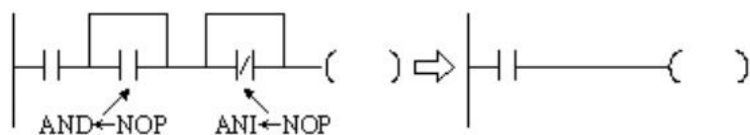
2. 指令说明

(1) 在程序中事先插入NOP指令, 以备在修改或增加指令时, 可使步进编号的更改次数减到最少。

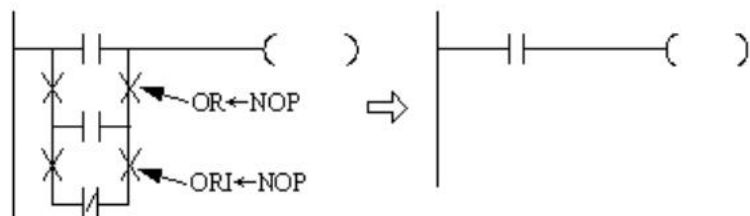
表3.16 NOP和END指令

助记符名称	操作功能	梯形图与目标组件	程序步数
NOP 空操作	空操作	无	1
END 结束	程序结束 返回 0 步	 无	1

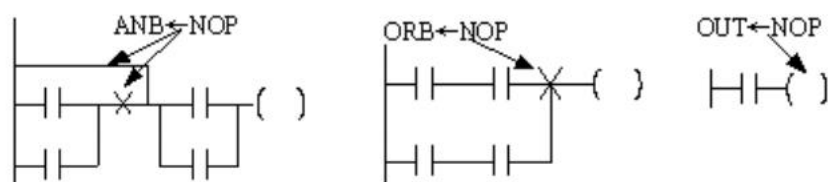
3.2.10 空操作与程序结束指令NOP、END 2



(a) 串联接点被短路



(b) 并联接点被开路



(c) 取消块串联后电路出错 (d) 切断并联电路后电路出错 (e) 取消输出电路出错

图3.23 用NOP指令取代已写入的指令引起电路改变

3.2.10 空操作与程序结束指令NOP、END 3

(2) 用NOP指令取代已写入的指令，从而修改电路。LD、LDI、AND、ANI、OR、ORI、ORB和ANB等指令若换成NOP指令，电路结构将会改变。

① AND和ANI指令改为NOP，相当于串联接点被短路，如图3.23 (a) 示例。

② OR和ORI指令改为NOP，相当于并联接点被开路，如图3.23 (b) 示例。

③ 如用NOP指令修改后的电路不合理，梯形图将出错，如图3.23 (c) ~ (e) 所示。

(3) NOP是一条空操作指令，CPU不执行目标指令。NOP在程序中占一个步序，该指令在梯形图中没有对应的软组件来表示它，但可从梯形图中的步序得到反映。

(4) 执行程序全清操作后，全部指令都变成NOP。

3.2.10 空操作与程序结束指令NOP、END 4

(5) END指令用于程序的结束，无目标操作数。END指令还可在程序调试中设置断点，先分段插入END指令，再逐段调试，调试好后，删去END指令。

例3.14 阅读图3.24 (a) 梯形图，试解答：

(1) 将图3.24 (a) 梯形图中的接点X001用 NOP指令代替，画出对应的梯形图。

(2) 将图3.24 (a) 梯形图中的接点X001和X003用 NOP代替，画出对应的梯形图。

(3) 比较图3.24 (a) 梯形图和作上述变换后的梯形图所对应的指令表。

解：(1) 将图3.24 (a) 中的接点X001用 NOP指令代替，与左母线相连的接点X001被取消，此时编程软件FXGP是将X003的左端接在上一逻辑行的Y000与X002相连处（同一逻辑层次点），得到的梯形图如图3.24 (b) 所示。

3.2.10 空操作与程序结束指令NOP、END 5

(2) 将图3.24 (a) 梯形图中的接点X001和X003都用 NOP指令代替, 在图3.24 (b) 中将X003短路, 得到的梯形图如图3.24 (c) 所示。

(3) 对应图3.24 (a) ~ (c) 梯形图的指令表, 分别如图3.25 (a) ~ (c) 所示。

3.3 梯形图程序设计方法

3.3.1 梯形图程序编程基本原则

梯形图程序设计规则

(1) 梯形图中的阶梯都是始于左母线，终于右母线。每行的左边是接点的组合，表示驱动逻辑线圈的条件，而表示结果的逻辑线圈只能接在右边的母线上，接点是不能出现在线圈的右边的。所以，图3.26 (a) 应改画为图3.26 (b)。

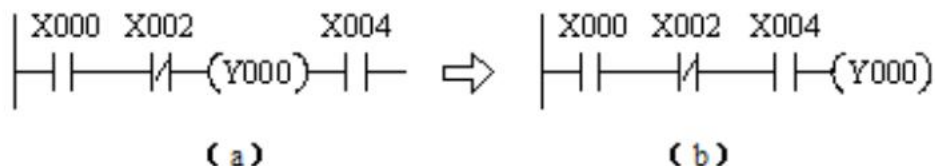


图3.26 接点不能出现在线圈的右边的原则

(2) 接点应画在水平线上，不要画在垂直线上。如图3.27 (a) 中接点X005与其它接点之间的连接关系不能识别，对此类桥式电路，要将其化为连接关系明确的电路。按从左至右，从上到下的单向性原则，可以看出有4条从左母线到达线圈Y000的不同支

3.3.1 梯形图程序编程基本原则 2

路，于是就可以将图3.27（a）不可编程的电路化为在逻辑功能上等效的图3.27（b）的可编程电路。

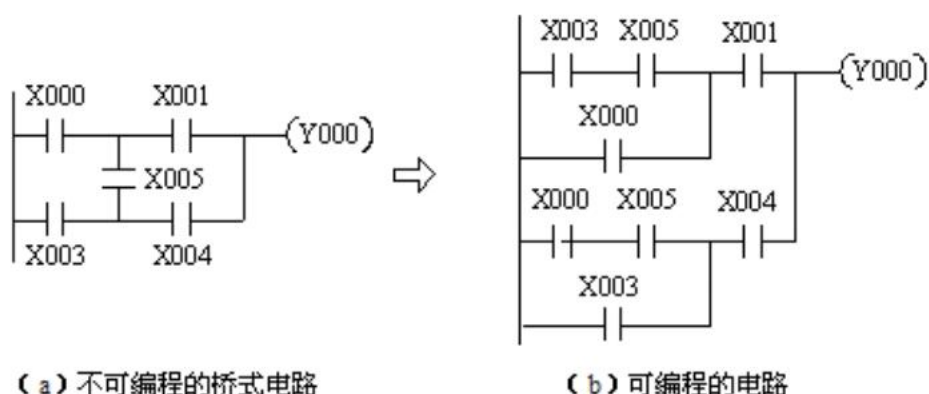


图3.27 不可编程的电路化为等效的可编程电路

（3）并联块串联时，应将接点多的支路放在梯形图的左方。串联块并联时，应将接点多的并联支路，放在梯形图的上方。这样安排，程序简洁，指令更少。图3.28（a）和图3.29（a）应分别改画为图3.28（b）和图3.29（b）为好。

3.3.1 梯形图程序编程基本原则 3

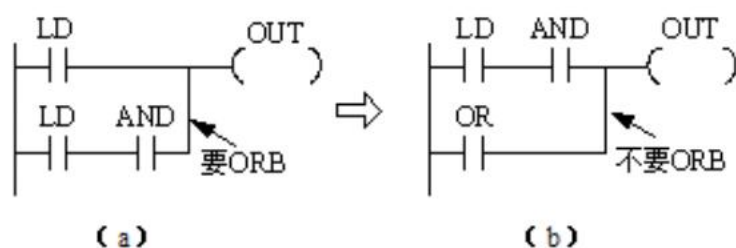


图3.28 上重下轻原则

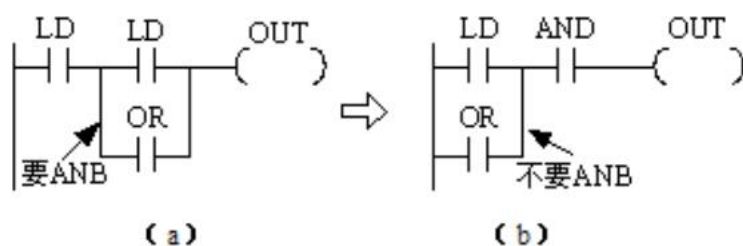


图3.29 左重右轻原则

(4) 双线圈输出不宜

若在同一梯形图中，同一组件的线圈使用两次或两次以上，称为双线圈输出。双线圈输出只有最后一次有效，一般不宜使用。

3.3.1 梯形图程序编程基本原则 4

设输入采样时，输入映象区中X001=ON，X002=OFF。第1次执行时，Y003=ON，Y004=ON；第2次执行时，X002=OFF，Y003=OFF；输出刷新时，实际输出，Y003=OFF，Y004=ON

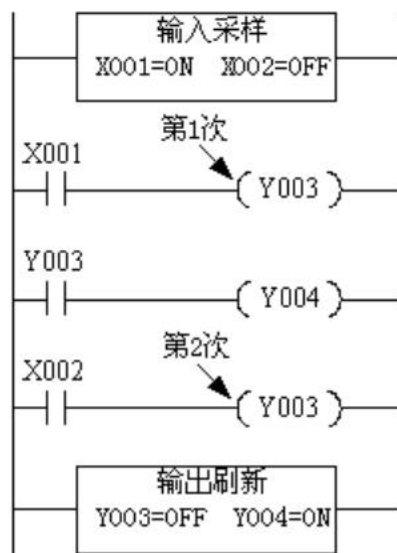


图3.30 不宜使用双线圈输出

3.3.2 梯形图的等效变换 1

在不改变逻辑关系的前提下，好的等效变换往往能化难为简、事半功倍。

(1) 在串联电路中，按梯形图设计规则改变组件的位置，使编程变为可能。如图3.26电路中，通过将线圈Y000移到右母处，应能使FXGP编译通过。

(2) 在电路块串并联电路中，按“左重右轻、上重下轻”的原则变换梯形图，使程序更优化。如图3.28和图3.29两电路，即为典型的实例。

(3) 在不易识别串并联关系的电路中，按从上到下、从左到右的单向性原则，找出所有能到达目标线圈的不同支路，变换梯形图为可编程电路，如图3.27电路即为典型的实例。

图3.31 梯形图的等效变换

3.3.2 梯形图的等效变换 3

例3.16 对图3.32 (a) 梯形图作等效变换，使成为合理的梯形图
解：方法1：双线圈输出时，只有最后一次才有效。因此，在图3.32 (a) 的虚线框中的逻辑行可忽略，剩下的梯形图与原梯形图是等效的。

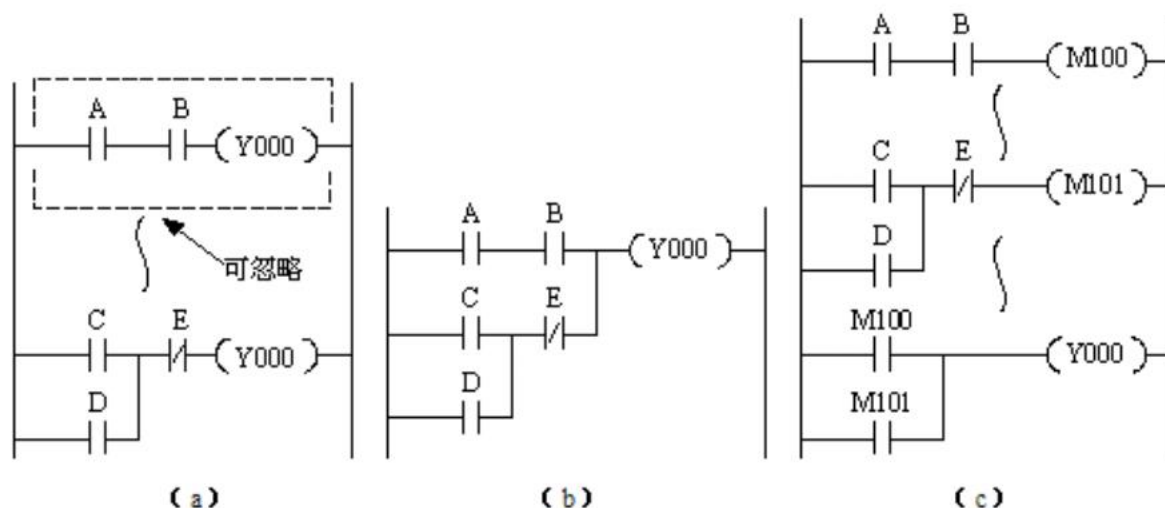


图3.32 梯形图的等效变换

3.3.2 梯形图的等效变换 4

方法2:

将图3.32 (a) 梯形图作相应变换, 得到单线圈的梯形图如图3.32 (b) 所示。图3.32 (b) 梯形图对Y000的逻辑控制关系与原梯形图是等效的。

方法3:

引入辅助寄存器是一个常用的好方法, 往往能使一些难以解决的逻辑控制问题迎刃而解。在图3.32 (c) 中, A和B接点控制M100, C、E和D接点控制M101, 再由M100和M101接点的并联组合去控制Y000。这样逻辑关系没有变, 却把双线圈梯形图变为单线圈的梯形图。所以图3.32 (c) 所示梯形图是与原梯形图等效的。

3.3.3 输入信号的最高频率 1

输入信号的最高频率限制：由于PLC采用集中I / O刷新的扫描工作方式，导致了在程序执行阶段和输出刷新阶段，即使输入信号发生变化，输入映象区的内容也不会改变。如果输入信号变化过快，有可能被PLC检测不到。所以输入信号的变化周期必须比PLC的扫描周期长，因此输入信号的最高频率就受到了限制。

设扫描周期一般为10ms，输入滤波器的响应延迟也为10ms，则输入脉冲的宽度至少为20ms，即其周期至少为40ms。可以估算出，输入脉冲频率应 $\leq 1 / 40\text{ms} = 25\text{Hz}$ 。这种滞后响应，在一般的工业控制场合是完全允许的，但对于要求响应速度快场合就不适应了。

3.3.3 输入信号的最高频率 2

对于高速场合，PLC除了提高扫描速度，还在软硬件上采取相应的措施，以提高I/O的响应速度。如在硬件方面，选用快速响应模块，高速计数模块。FX2系列PLC还提供X0~X7共8个高速输入端，其RC滤波器时间常数仅为 $50\mu\text{s}$ 。在软件方面采用I/O立即信息刷新方式、中断传送方式和能用指令修改的数字式滤波器等。因此，可以处理的输入信号的最高频率有很大提高。FX2N系列PLC是小型化，高速度，高性能和所有方面都是相当于FX系列中最高档次的机型，它的1个基本指令运行时间只需 $0.08\mu\text{s}$ ，可读取最大 $50\mu\text{s}$ 的短脉冲输入，可见输入信号的最高频率可以达到20KHz。

3.4 基本指令应用程序举例 1

例3.17 参照图3.33设计一个三相异步电机正反转PLC控制系统。
设计步骤

- (1) 功能要求：
- ① 当接上电源时，电机M不动作。
 - ② 按下SB1，电机正转；按SB3，电机停转。
 - ③ 按下SB2，电机反转；按SB3，电机停转。
 - ④ 热继电器触点FR断，电机过载保护停转。

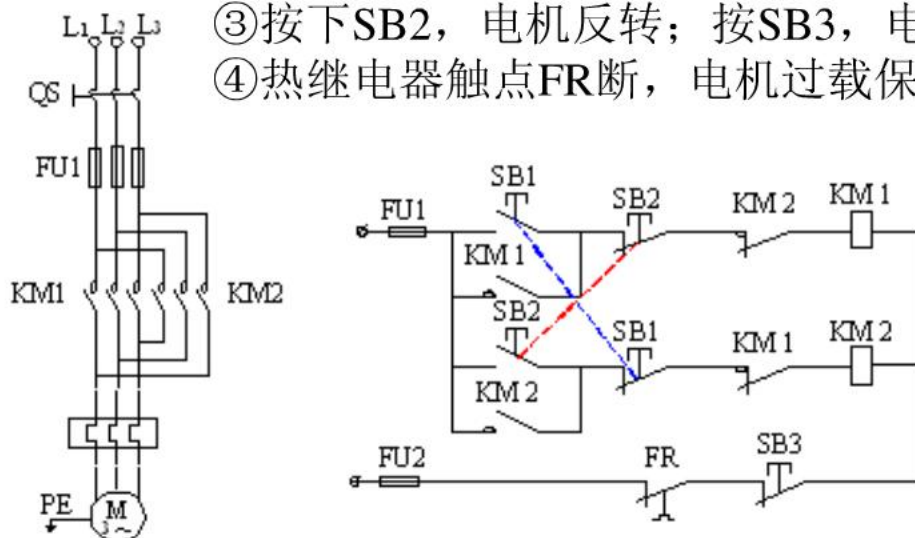


图3.33 三相异步电机正反转控制线路

3.4 基本指令应用程序举例 2

③按下SB2，电机M反转；按SB3，电机M停转。

④热继电器触点FR动作，电机M因过载保护而停止。

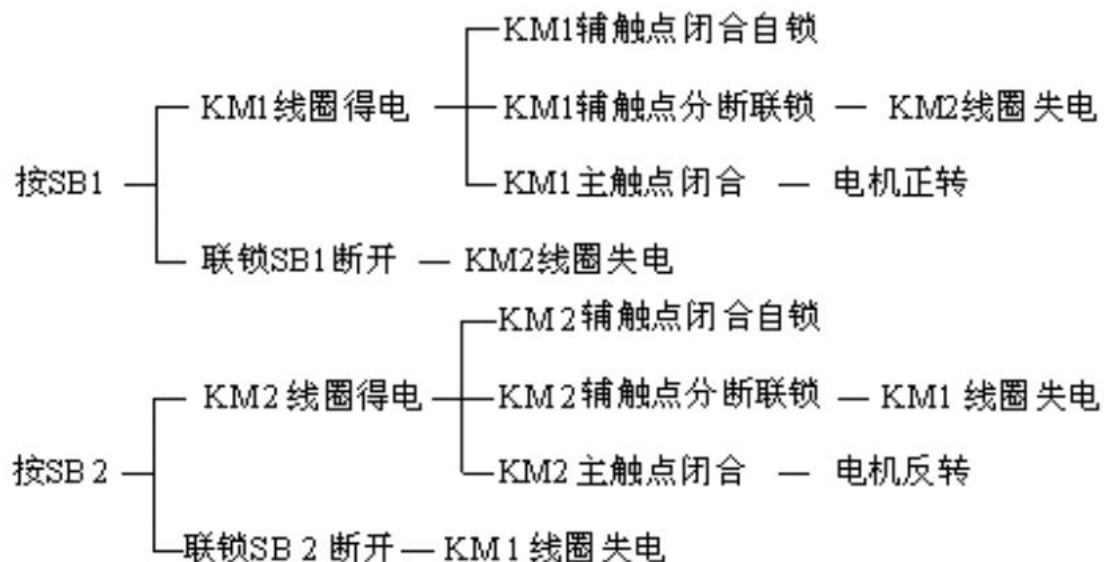


图3.34 三相异步电机正反转控制线路的动作顺序

3.4 基本指令应用程序举例 3

(2) 输入/输出端口设置

表3.17 三相异步电机正反转PLC控制I/O端口分配表

输 入			输 出		
名称		输入点	名称		输出点
正转启动按钮	SB1	X001	正转接触器	KM1	Y001
反转启动按钮	SB2	X002	反转接触器	KM2	Y002
停止按钮	SB3	X003			
热继电器触点	FR	X004			

(3) 梯形图

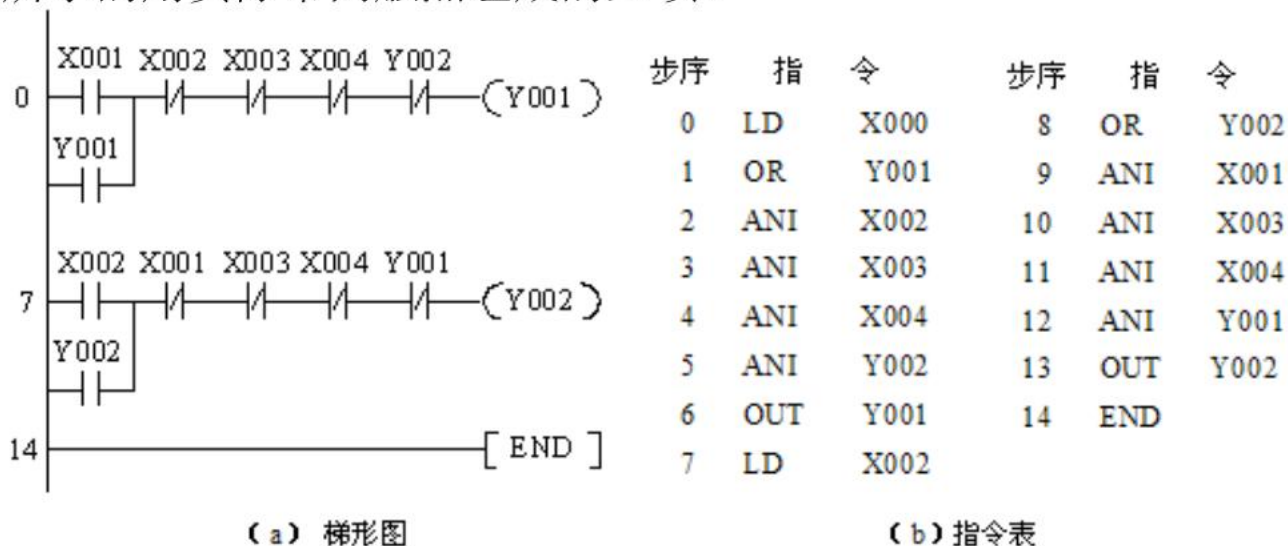
三相异步电机正反转控制系统梯形图如图3.35 (a)，其动作顺序完全符合表3.17，只要按表3.17的I/O分配作相应替换即行。

(4) 指令表指令表如图3.35 (b) 所示。

(5) 接线图接线图如图3.36所示。

3.4 基本指令应用程序举例 4

为防止正反转启动按钮同时按下危险情况，一方面，在梯形图中设置了互锁，将常闭X001和Y001串联在反转电路中，将常闭X002和Y002串联在正转电路中。另一方面，在外部也设置了如图3.36所示的用实际常闭触点组成的互锁。



(a) 梯形图

(b) 指令表

图3.35 三相异步电机正反转控制

3.4 基本指令应用程序举例 5

为防止正反转启动按钮同时按下危险情况，一方面，在梯形图中设置了互锁，将常闭X001和Y001串联在反转电路中，将常闭X002和Y002串联在正转电路中。另一方面，在外部也设置了如图3.36所示的用实际常闭触点组成的互锁。

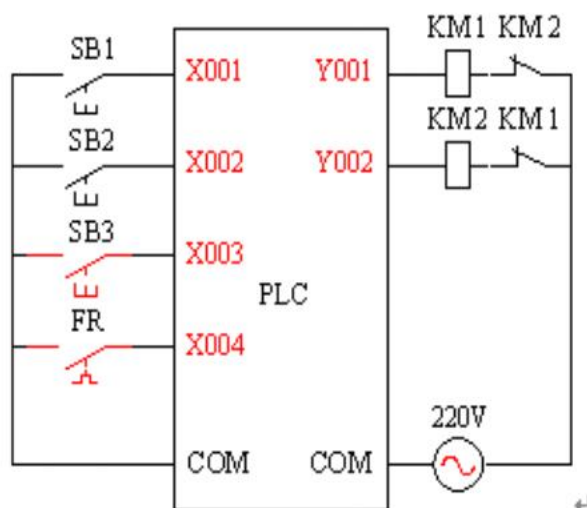


图3.36 PLC控制的接线图

3.4 基本指令应用程序举例 6

例3.18 设计一个用FX1S-20MT的输出端子直接驱动直流小电动机正反转控制系统。直流电机的规格在12V/0.5A以下。

直流电机正反转驱动电路，是通过电源极性的切换来控制电机转向，可参照桥式整流电路来设计。只要将桥式整流电路中的四个整流二极管用四个继电器的触点来取代，负载则用直流电机来取代，如图3.37（a）所示。

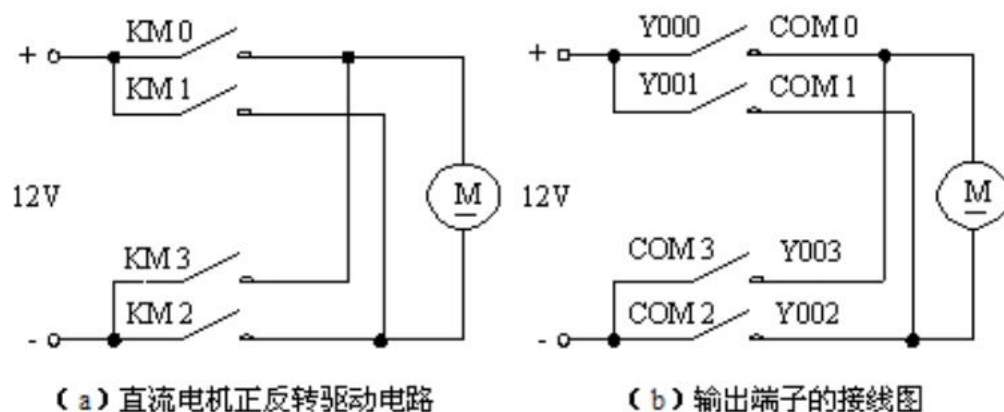


图3.37 直流电机正反转驱动与接线

3.4 基本指令应用程序举例 7

控制电路设计可参照例3.17交流异步电机的控制，不同的是要控制的继电器线圈有4个，动作过程，参看图3.38的动作顺序表。

(1) 功能要求

① 当接上电源时，电机M不动作。

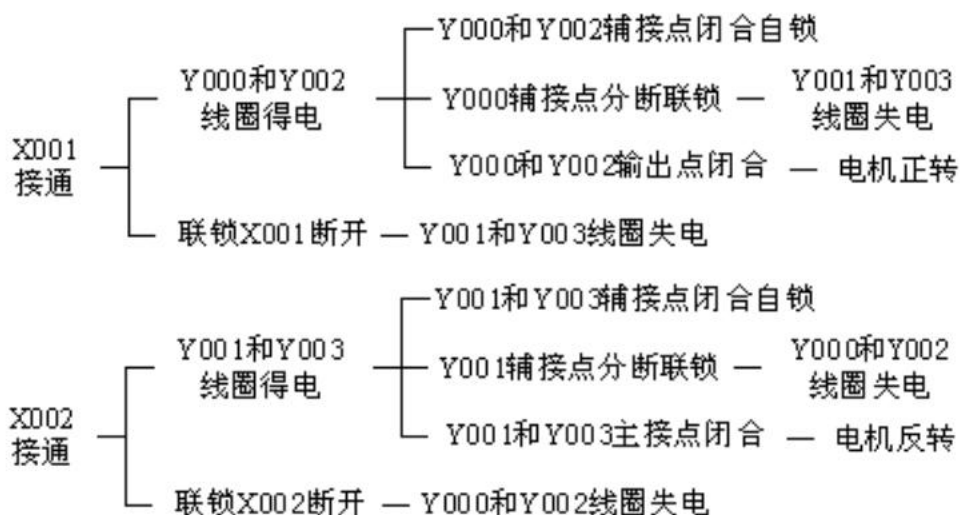


图3.38 直流电机正反转控制动作顺序表

3.4 基本指令应用程序举例 8

- ②按下SB1后, 电机正转; 再按SB3后, 电机停转。
③按下SB2后, 电机反转; 再按SB3后, 电机停转。
④热继电器触点FR动作后, 电机M因过载保护而停止。

(2) 输入/ 输出端口设置

(3) 梯形图

(4) 指令表

(5) 接线图

表3.18 直流电机正反转PLC控制I/O端口分配表

输 入			输 出		
名称		输入点	名称		输出点
正转启动按钮	SB1	X001	正转输出端	Y000	Y000
反转启动按钮	SB2	X002	正转输出端	Y002	Y002
停止按钮	SB3	X003	反转输出端	Y001	Y001
热继电器触点	FR	X004	反转输出端	Y003	Y003

3.4 基本指令应用程序举例 9

FX1S-20MT是晶体管输出，输出结构如图3.41所示。当晶体管截止时，输出端子Y0与公共端COM0断开。当晶体管导通时，Y0与COM0接通，要注意的是导通是单向的，即导通时的电流流向只能是从Y0流向COM0。所以图3.37（a）中4个开关的实际接法应如图3.37（b）所示。图3.40就是按此画出的接线图。

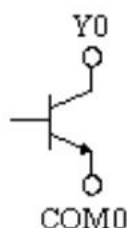


图3.41 晶体管输出结构

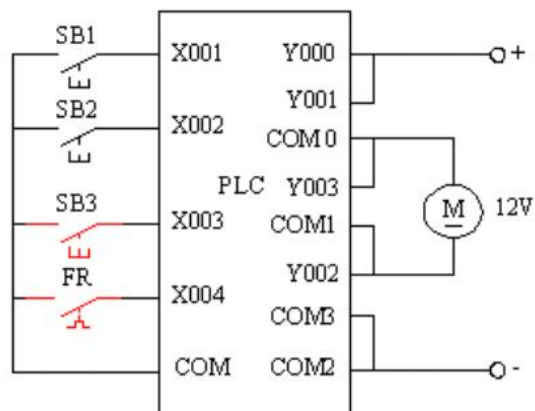


图3.40 PLC控制的接线图

3.4 基本指令应用程序举例 10

例3.19 流水行云——设计一个彩灯控制的PLC系统。

(1) 功能要求

- ①合启动按钮SB2，彩灯HL0~HL7(Y000~Y007)按间隔2s点亮。
- ②至彩灯HL0~HL7全亮，维持5s；此后全熄，维持3s；自动重复下一轮循环。

(2) 输入/输出端口设置

表3.19 彩灯PLC控制的I/O端口分配表

输 入			输 出		
名称		输入点	名称		输出点
启动开关	SB2	X010	3号彩灯	HL3	Y003
输 出			4号彩灯	HL4	Y004
名称		输出点	4号彩灯	HL4	Y004
0号彩灯	HL0	Y000	5号彩灯	HL5	Y005
1号彩灯	HL1	Y001	6号彩灯	HL6	Y006
2号彩灯	HL2	Y002	7号彩灯	HL7	Y007

3.4 基本指令应用程序举例 11

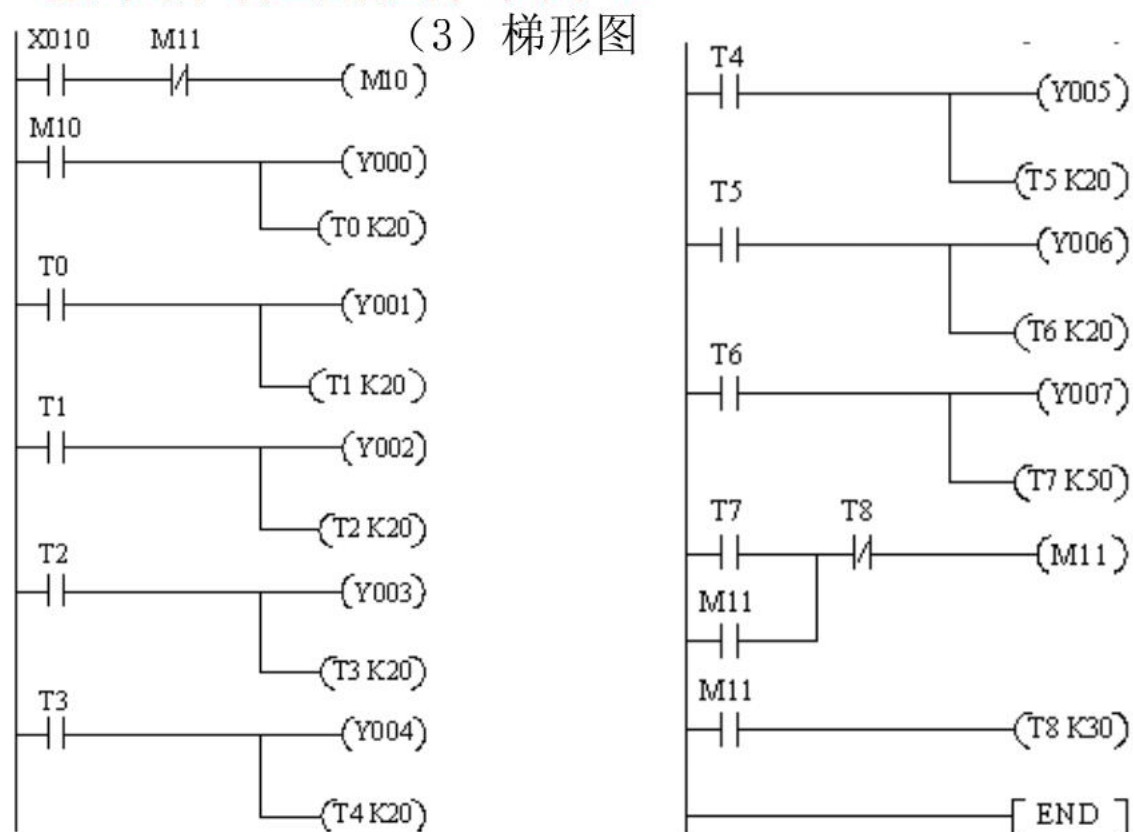


图3.42 (a) 彩灯PLC控制的梯形图

3.4 基本指令应用程序举例 12

(4) 指令表

步序	指	令	
0	LD	X010	28 LD T4
1	ANI	M11	29 OUT Y005
2	OUT	M10	30 OUT T5 K20
3	LD	M10	33 LD T5
4	OUT	Y000	34 OUT Y006
5	OUT	T0 K20	35 OUT T6 K20
8	LD	T0	38 LD T6
9	OUT	Y001	39 OUT Y007
10	OUT	T1 K20	40 OUT T7 K50
13	LD	T1	43 LD T7
14	OUT	Y002	44 OR M11
15	OUT	T2 K20	45 ANI T8
18	LD	T2	46 OUT M11
19	OUT	Y003	47 LD M11
20	OUT	T3 K20	48 OUT T8 K30
23	LD	T3	
24	OUT	Y004	
25	OUT	T4 K20	51 END

图3.42 (b) 彩灯PLC控制的指令表

3.4 基本指令应用程序举例 13

(5) 接线图

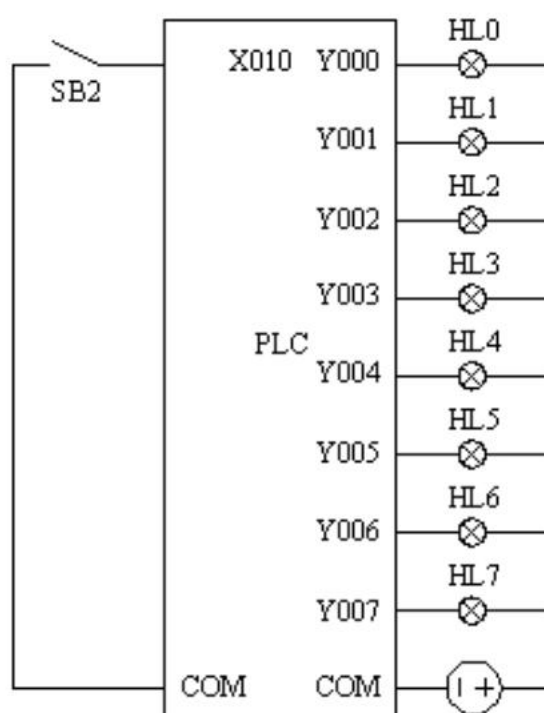


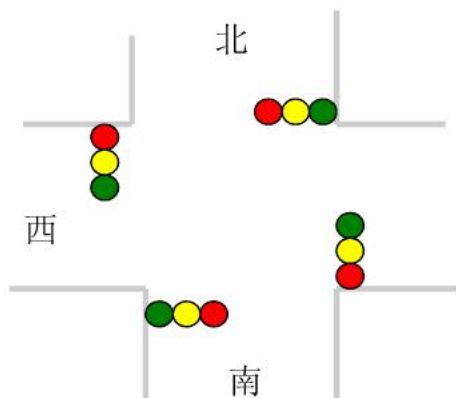
图3.43 PLC彩灯控制接线图

本章小结

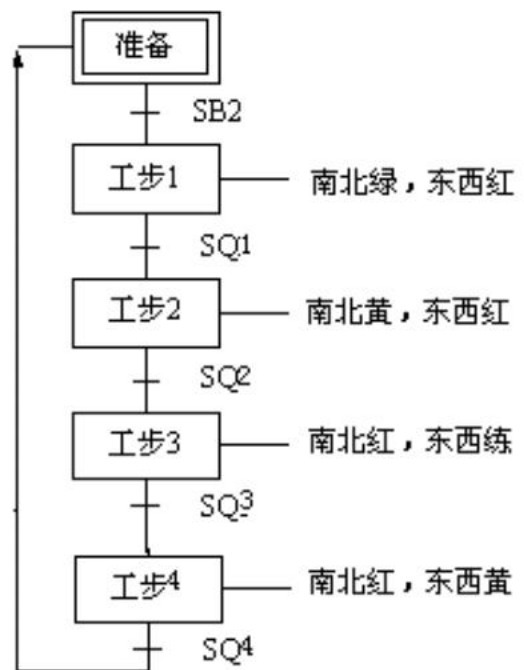
1. 三菱FX2系列PLC的编程组件如附录表A.1所示，供编程时查阅。
2. 常用的梯形图、指令表、流程图等编程语言，各有其特点。请根据需要选用。
3. FX2系列PLC共有**20条基本逻辑指令**，这些指令已经能解决一般的继电接触控制问题，要求能熟练掌握。对于20条基本逻辑指令，应当注意掌握每条指令的助记符名称、操作功能、梯形图、目标组件和程序步数。FX2系列PLC基本逻辑指令表，见附录B的“表B.1 FX2系列PLC基本逻辑指令表”，供查阅用。
4. 熟练掌握用梯形图进行程序设计方法。
5. 对于复杂的顺控系统，以SFC语言的状态转移图方式进行程序设计，是解决顺序控制问题的有效方法。

三、流程图语言（SFC， Sequential Function Chart）

SFC： 又称状态转移图，适合于编制复杂的顺序控制程序。



第一步：南北绿、东西红；
第二步：南北黄、东西红；
第三步：南北红、东西绿；
第四步：南北红、东西黄。



(2) SFC语言元素：状态、转移条件和有向线段。

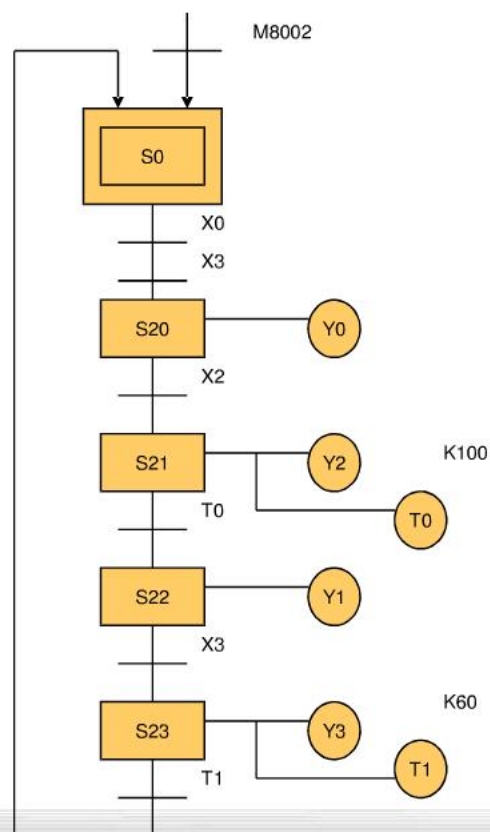
★ 状态：表示过程中的每一个工步。

双线框+组号，表示初始状态；

单线框+组号，表示各工作状态

★ 状态间要用有向线段连接，以表示转移方向，凡从上到下、从左到右的有向线段箭头可省去不画。

★ 有向线段上的垂直短线和它旁边标注的文字符号或逻辑表达式表示状态转移条件



3.1.3 流程图语言 (SFC) 5

(3) SFC流程图的基本形式

①单流程结构：其状态是一个接着一个地顺序进行，每个状态仅连接一个转移，每个转移也仅连接一个状态。

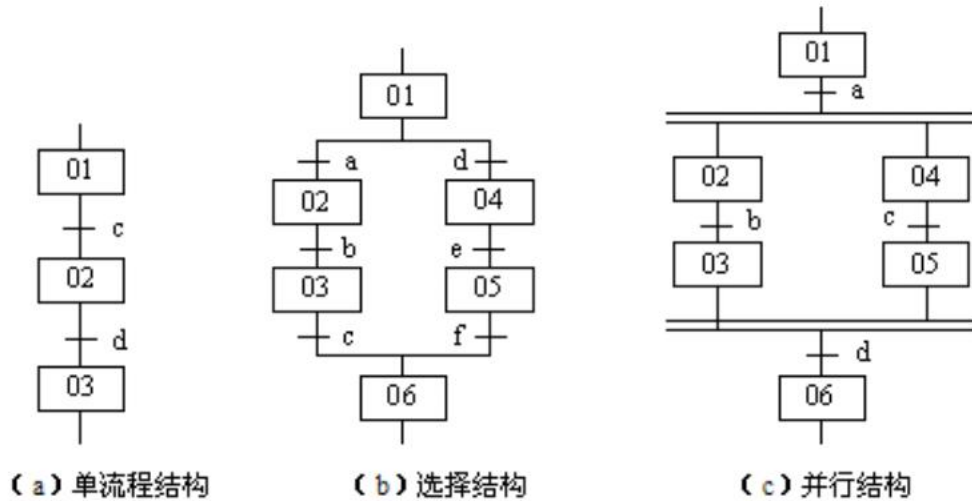
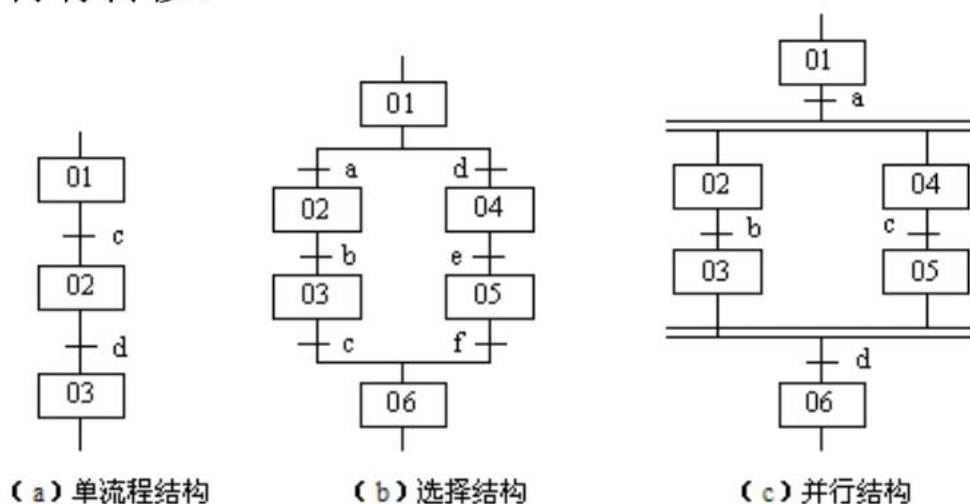


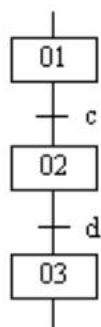
图3.6 SFC流程图的三种基本形式

3.1.3 流程图语言 (SFC) 6

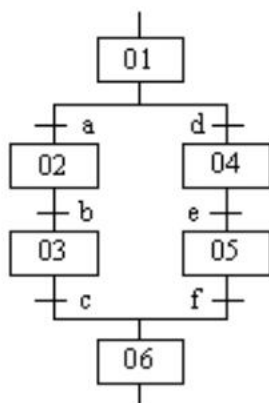
②**选择结构**：在某一状态后有几个单流程分支，当相应的转移条件满足时，一次只能选择进入一个单流程分支。选择结构的转移条件是在某一状态后连接一条水平线，水平线下再连接各个单流程分支的第一个转移。各个单流程分支结束时，也要用一条水平线表示，而且其下不允许再有转移。



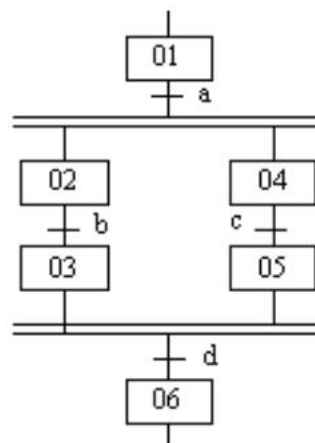
③**并行结构**是指在某一转移下，若转移条件满足，将同时触发并行的几个单流程分支，这些并行的顺序分支应画在两条双水平线之间。



(a) 单流程结构



(b) 选择结构



(c) 并行结构

三种程序设计语言比较：梯形图具有与传统继电接触控制相似的特征，编程直观、形象，易于掌握。助记符语言适合编程器在现场调试程序。SFC语言以状态转移图方式编程，适合于编制复杂的顺控程序。