

项目 2 基本逻辑指令及应用

基本逻辑指令是 PLC 中应用最频繁的指令，是程序设计的基础。本项目主要介绍三菱 FX_{2N} 系列 PLC 的基本逻辑指令及其编程使用。

2.1 三相异步电动机的点动、连续运行控制



任务目标

- ① 学习并初步掌握常用基本逻辑指令的应用。
- ② 学习并熟悉 FX_{2N} 型 PLC 的 I/O 接线。
- ③ 掌握 FXGP/WIN-C 和 GPPW 编程软件的使用。



任务分析

① 电动机点动正转控制。点动正转控制线路是用按钮、接触器来控制电动机运转的最简单的正转控制线路。按下按钮，电动机就得电启动；松开按钮，电动机就失电停止。

② 电动机连续运行控制。电动机单向运行的启动/停止控制是最基本、最常用的控制。按下启动按钮，电动机就得电启动运行，按下停止按钮，电动机就失电停止。

③ 为了解电动机的运行状况，可以分别用绿色指示灯 HL1 和红色指示灯 HL2 表示电动机启动和停止状态。



相关知识

2.1.1 逻辑取及驱动线圈指令 LD/LDI/OUT

逻辑取及驱动线圈指令如表 2-1 所示。

表 2-1 逻辑取及驱动线圈指令表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
LD 取	常开触点逻辑运算起始		X, Y, M, T, C, S	1
LDI 取反	常闭触点逻辑运算起始		X, Y, M, T, C, S	1



续表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
OUT 输出	线圈驱动		Y, M, T, C, S	Y、M: 1, 特 M: 2, T: 3, C: 3~5

1. 用法示例

逻辑取及驱动线圈指令的应用如图 2-1 所示。

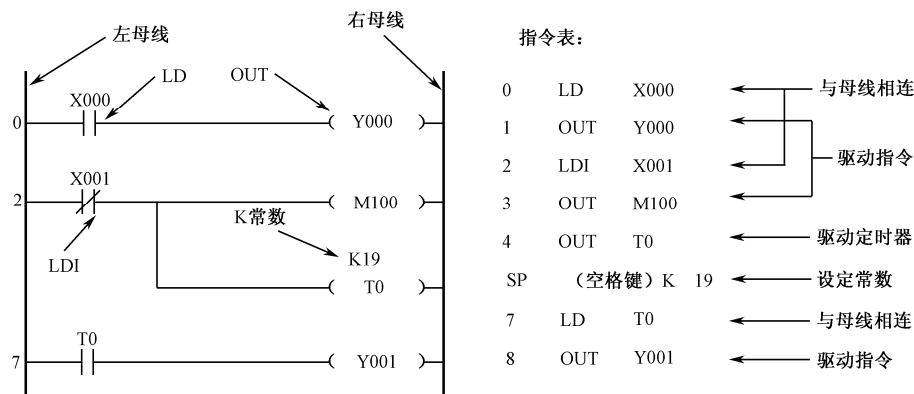


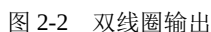
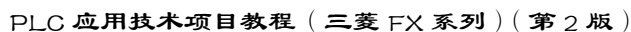
图 2-1 逻辑取及驱动线圈指令的用法

2. 使用注意事项

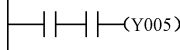
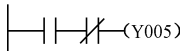
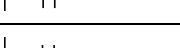
- ① LD 是电路开始的常开触点，连接到母线上，可以用于 X、Y、M、T、C 和 S。
- ② LDI 是电路开始的常闭触点，连接到母线上，可以用于 X、Y、M、T、C 和 S。
- ③ OUT 是驱动线圈的输出指令，可以用于 Y、M、T、C 和 S。
- ④ LD 与 LDI 指令对应的触点一般与左侧母线相连，若与后述的 ANB、ORB 指令组合，则可用于串、并联电路块的起始触点。
- ⑤ 线圈驱动指令可并行多次输出（即并行输出），如图 2-1 梯形图中的 OUT M100、OUT T0 K19。
- ⑥ 输入继电器 X 不能使用 OUT 指令。
- ⑦ 对于定时器的定时线圈或计数器的计数线圈，必须在 OUT 指令后设定常数。

3. 双线圈输出

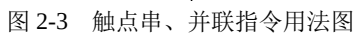
线圈一般不能重复使用（也称双线圈输出），如图 2-2 所示为同一线圈 Y3 多次使用的情况。设 X1=ON，X2=OFF，最初因 X1=ON，Y3 的映像寄存器为 ON，输出 Y4 也为 ON，然后紧接着又因 X2=OFF，Y3 的映像寄存器改写为 OFF，因此，最终的外部输出 Y3 为 OFF，Y4 为 ON。所以，若输出线圈重复使用，则后面线圈的动作状态对外输出有效。



触点串、并联指令如表 2-2 所示。

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
AND 与	常开触点串联连接		X, Y, M, S, T, C	1
ANI 与非	常闭触点串联连接		X, Y, M, S, T, C	1
OR 或	常开触点并联连接		X, Y, M, S, T, C	1
ORI 或非	常闭触点并联连接		X, Y, M, S, T, C	1

触点串、并联指令的应用如图 2-3 所示。





2. 使用注意事项

① AND 是常开触点串联连接指令, ANI 是常闭触点串联连接指令, OR 是常开触点并联连接指令, ORI 是常闭触点并联连接指令。这 4 条指令后面必须有被操作的元件名称及元件号, 都可以用于 X、Y、M、T、C 和 S。

② 单个触点与左边的电路串联, 使用 AND 和 ANI 指令时, 串联触点的个数没有限制, 但是因为图形编程器和打印机的功能有限制, 所以建议尽量做到一行不超过 10 个触点和 1 个线圈。

③ OR 和 ORI 指令是从该指令的当前步开始, 对前面的 LD、LDI 指令并联连接, 并联连接的次数无限制, 但是因为图形编程器和打印机的功能有限制, 所以并联连接的次数不超过 24 次。

④ OR 和 ORI 用于单个触点与前面电路的并联, 并联触点的左端接到该指令所在的电路块的起始点 (LD) 上, 右端与前一条指令对应的触点的右端相连, 即单个触点并联到它前面已经连接好的电路的两端 (两个以上触点串联连接的电路块的并联连接时, 要用后续的 ORB 指令)。以图 2-3 中的 M110 的常开触点为例, 它前面的 4 条指令已经将 4 个触点串、并联为一个整体, 因此 OR M110 指令对应的常开触点并联到该电路的两端。

3. 连续输出

如图 2-4 (a) 所示, OUT M1 指令之后通过 X1 的触点去驱动 Y4, 称为连续输出。串联和并联指令用来描述单个触点与别的触点或触点 (而不是线圈) 组成的电路的连接关系。虽然 X1 的触点和 Y4 的线圈组成的串联电路与 M1 的线圈是并联关系, 但是 X1 的常开触点与左边的电路是串联关系, 所以对 X1 的触点应使用串联指令。只要按正确的顺序设计电路, 就可以多次使用连续输出, 但是因为图形编程器和打印机的功能有限制, 所以连续输出的次数不超过 24 次。

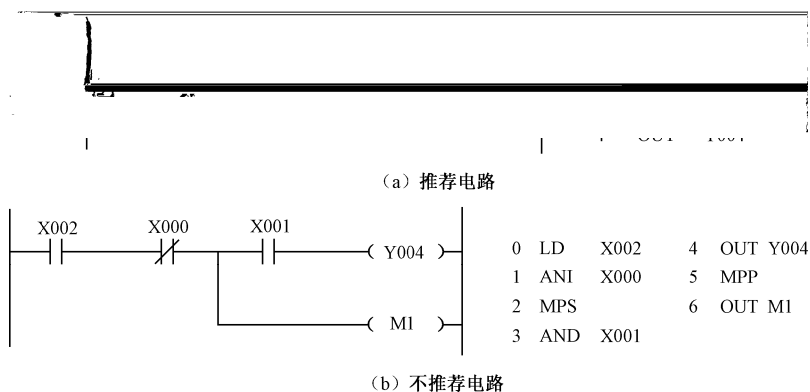


图 2-4 连续输出电路

应该指出, 如果将图 2-4 (a) 中的 M1 和 Y4 线圈所在的并联支路改为如图 2-4 (b) 所示的电路 (不推荐), 就必须使用后面要讲到的 MPS (进栈) 和 MPP (出栈) 指令。

2.1.3 PLC 编程语言的国际标准

PLC 编程语言标准 (IEC 61131-3) 中有 5 种编程语言, 即顺序功能图 (Sequential Function



chart)、梯形图 (Ladder diagram)、功能块图 (Function block diagram)、指令表 (Instruction list) 和结构文本 (Structured text)。其中顺序功能图 (SFC)、梯形图 (LD)、功能块图 (FBD) 是图形编程语言, 指令表 (IL)、结构文本 (ST) 是文字语言。

(1) 顺序功能图 (SFC)

顺序功能图用来描述开关量控制系统的功能, 是一种位于其他编程语言之上的图形语言, 用于编制顺序控制程序。顺序功能图提供了一种组织程序的图形方法, 根据它可以很容易地画出顺序控制梯形图程序, 本书将在项目 3 中做详细介绍。

(2) 梯形图 (LD)

梯形图是一种以图形符号及其在图中的相互关系来表示控制关系的编程语言, 是从继电器电路图演变过来的, 是使用最多的 PLC 图形编程语言。梯形图与继电器控制系统的电路图很相似, 直观易懂, 很容易被熟悉继电器控制的电气人员掌握, 特别适用于开关量逻辑控制。梯形图由触点、线圈和应用指令等组成, 触点代表逻辑输入条件, 如外部的开关、按钮和内部条件等; 线圈通常代表逻辑输出结果, 用来控制外部的指示灯、交流接触器等。

梯形图通常有左右两条母线 (有的时候只画左母线), 两母线之间是内部继电器常开、常闭的触点及继电器线圈组成的一条条平行的逻辑行 (或称梯级), 每个逻辑行必须以触点与左母线连接开始, 以线圈与右母线连接结束。

(3) 功能块图 (FBD)

这是一种类似于数字逻辑门电路的编程语言, 有数字电路基础的人很容易掌握。该编程语言用类似与门、或门的方框来表示逻辑运算关系, 方框的左侧为逻辑运算的输入变量, 右侧为输出变量, 输入、输出端的小圆圈表示“非”运算, 方框被“导线”连接在一起, 信号自左向右流动。国内很少使用功能块图语言。

(4) 指令表 (IL)

PLC 的指令是一种与微型计算机的汇编语言中的指令相似的助记符表达式, 由指令组成的程序叫做指令表程序。指令表程序较难阅读, 其中的逻辑关系很难一眼看出, 所以在设计时一般使用梯形图语言。如果使用手持式编程器, 必须将梯形图转换成指令表后再写入 PLC。在用户程序存储器中, 指令按步序号顺序排列。

(5) 结构文本 (ST)

结构文本 (ST) 是为 IEC 61131—3 标准创建的一种专用的高级编程语言。与梯形图相比, 它能实现复杂的数学运算, 编写的程序非常简捷和紧凑。IEC 标准除了提供几种编程语言供用户选择外, 还允许编程者在同一程序中使用多种编程语言, 这使编程者能选择不同的语言来适应特殊的工作。

2.1.4 梯形图的主要特点

(1) PLC 梯形图中的某些编程元件沿用了继电器这一名称, 如输入继电器、输出继电器、内部辅助继电器等, 它们不是真实的物理继电器 (即硬件继电器), 而是在梯形图中使用的编程元件 (即软元件)。每一软元件与 PLC 存储器中元件映像寄存器的一个存储单元相对应。以辅助继电器为例, 如果该存储单元为 0 状态, 则梯形图中对应的软元件的线圈“断电”, 其常开触点断开, 常闭触点闭合, 称该软元件为 0 状态, 或称该软元件为 OFF (断开)。如果该存储单元为 1 状态, 则对应软元件的线圈“有电”, 其常开触点接通, 常闭触点断开, 称该软元件为 1 状态, 或称该软元件为 ON (接通)。



(2) 根据梯形图中各触点的状态和逻辑关系, 求出图中各线圈对应的软元件的 ON/OFF 状态, 称为梯形图的逻辑运算。逻辑运算是按梯形图从上到下、从左至右的顺序进行的, 运算的结果可以马上被后面的逻辑运算所利用。逻辑运算是根据元件映像寄存器中的状态, 而不是根据运算瞬时外部输入触点的状态来进行的。

(3) 梯形图中各软元件的常开触点和常闭触点均可以无限多次地被使用。

(4) 输入继电器的状态唯一取决于对应的外部输入电路的通断状态, 因此在梯形图中不能出现输入继电器的线圈。

(5) 辅助继电器相当于继电控制系统中的中间继电器, 用来保存运算的中间结果, 不对外驱动负载, 负载只能由输出继电器来驱动。

2.1.5 任务实现：用 PLC 实现三相异步电动机的点动、连续运行控制

采用 PLC 进行电动机的控制, 主电路与传统继电器接触器控制的主电路一样, 不同的是控制电路。由于 PLC 的加入, 用户只需将输入设备 (如启动按钮 SB1、停止按钮 SB2、点动按钮 SB3、热继电器触点 FR) 接到 PLC 的输入端口, 输出设备 (如接触器线圈 KM、运行指示灯 HL1 和 HL2) 接到 PLC 的输出端口, 再接上电源、输入软件程序就可以了。具体线如何接, 程序如何编写, 编写好的程序如何输入及调试, 下面将详细介绍。

1. I/O 分配

在进行接线与编程前, 首先要确定输入/输出设备与 PLC 的 I/O 口的对应关系问题, 即要进行 I/O 分配工作。只有 I/O 分配工作结束后, 才能绘制 PLC 接线图, 也才能具体进行程序的编写工作。因此 I/O 分配是选择确定了输入/输出设备后首先要做的工作。

如何进行 I/O 分配呢? 这是一项十分简单的工作。具体来说, 就是将每一个输入设备对应一个 PLC 的输入点, 将每一个输出设备对应一个 PLC 的输出点。

为了绘制 PLC 接线图及运用 PLC 编程, I/O 分配后应形成一张 I/O 分配表, 明确表示出输入/输出设备有哪些, 它们各起什么作用, 对应的是 PLC 的哪些点, 这就是 PLC 的 I/O 分配。下面进行三相异步电动机的点动、连续运行 PLC 控制的 I/O 分配。

根据前面控制要求可知, 点动、连续运行控制的输入设备应是 4 个, 输出设备应是 3 个, 应选择与此输入/输出点数相适应的 PLC。三菱 FX_{2N}-16MR 有 8 个输入和 8 个输出点, 能满足此要求。点动、连续运行控制的 I/O 地址分配如表 2-3 所示。

表 2-3 点动、连续运行控制输入/输出地址分配

输 入			输 出		
输入继电器	电路元件	作用	输出继电器	电路元件	作用
X000	SB1	启动按钮	Y000	KM	电动机接触器
X001	SB2	停止按钮	Y001	HL1	启动绿色指示灯
X002	SB3	点动按钮	Y002	HL2	停止红色指示灯
X003	FR	过载保护	—	—	—

2. 硬件接线

输入设备接入 PLC 的方法十分简单, 即将两端输入设备的一个端点接到指定 PLC 的输



入端口上，另一个端点接到 PLC 的公共端上。输出设备的接线略复杂一些，主要应根据输出设备的工作特性（工作电压的类型及数值）做好分组工作，同时还应将电源接入电路。点动、连续运行控制的接线图如图 2-5 所示。

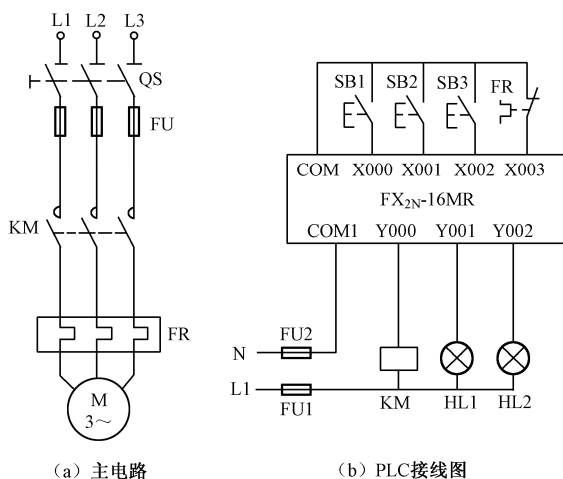


图 2-5 点动、连续运行控制的接线图

3. 编程

PLC 程序主要解决的是如何根据输入设备的信息（通、断信号）按照控制要求形成驱动输出设备的信号，使输出满足控制要求。PLC 的程序形式有多种，最常用的是梯形图，其次是指令形式，两者之间是可以互相转换的。程序的形式可以不同，但描述的内容是相同的，程序的实质是描述控制的逻辑关系。对于初学者来说，最关键是如何编写 PLC 程序。

编写 PLC 程序，最基本的方法是经验法。经验法要求编程者具有控制系统的设计经验，而作为初学者在控制系统设计方面的主要经验只有继电器接触器控制系统的初步设计经验，因此，对于继电器接触器控制系统中常用基本控制电路的理解及设计经验是十分宝贵的，它将给我们带来许多有关电动机控制程序设计的灵感，特别是继电器接触器控制中的启—保—停控制电路、正反转控制电路，这些将是编程的基本依据。下面将根据这些经验来初步构思并理解编写的程序。

点动实际上是利用输入的触点来控制输出的线圈，而连续控制则是典型的启—保—停控制电路，这两种基本控制电路控制的对象实际上是同一个线圈。如何使两者控制不发生冲突，最好的办法就是利用辅助继电器。将点动控制的对象改为一个辅助继电器，再将连续控制的对象改为另一个辅助继电器，最后再利用两个辅助继电器的触点来控制输出继电器。这就是采用 PLC 实现点动、连续运行控制的基本思路，然后再加入指示灯的控制程序，就形成了 PLC 的控制程序。梯形图程序如图 2-6 所示，指令表程序如表 2-4 所示。

4. 调试

- ① 在断电状态下连接好电缆。
- ② 将 PLC 运行模式选择开关拨到 STOP 位置。
- ③ 使用编程软件进行编程并下载。

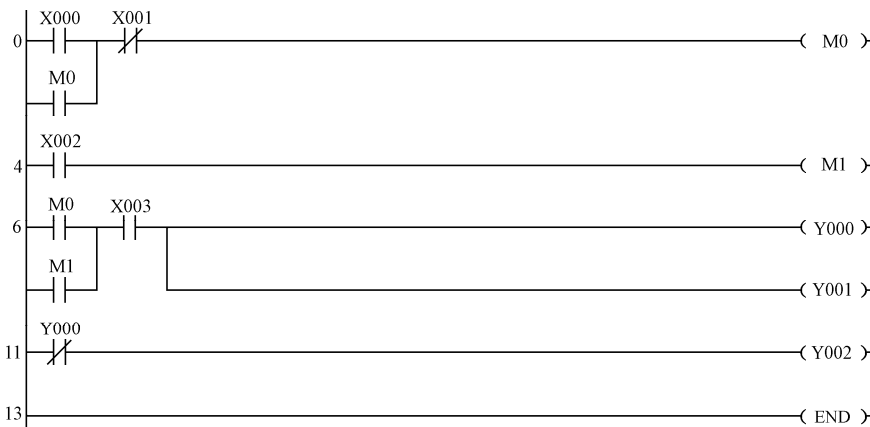


图 2-6 点动、连续运行控制的梯形图程序

表 2-4 点动、连续运行控制的指令程序

指令程序	指令程序	指令程序	指令程序
0 LD X000	4 LD X002	8 AND X003	12 OUT Y002
1 OR M0	5 OUT M1	9 OUT Y000	13 END
2 ANI X001	6 LD M0	10 OUT Y001	
3 OUT M0	7 OR M1	11 LDI Y000	

④ 将 PLC 运行模式选择开关拨到 RUN 位置或采用编程软件中的遥控运行。

⑤ 观察 PLC 中 Y2 的 LED 是否点亮，如果处于点亮状态，表明电动机处于停止状态。

⑥ 按下点动按钮 SB3，观察电动机是否可以运行。松开点动按钮 SB3，观察电动机是否能够停车。如果均可完成，则说明点动控制程序正确。同时观察在电动机运行时 Y1 是否点亮，若指示正常则程序正确。

⑦ 按下启动按钮 SB1 并松开后，如果系统能够启动一直运行，并能在按下停止按钮 SB2 后停车，则程序调试结束。

⑧ 如果出现故障，学生应独立检查，直至排除故障，使系统能够正常工作。



知识链接

1. 置位与复位指令 SET/RST

置位与复位指令如表 2-5 所示。

表 2-5 置位与复位指令表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
SET 置位	令元件自保持 ON		Y, M, S	Y, M: 1 S, 特 M: 2



续表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
RST 复位	令元件自保持 OFF 或清除数据寄存器的内容		Y, M, S, C, D, V, Z, 积 T	Y, M: 1 S, 特 M, C, 积 T: 2; D, V, Z: 3

(1) 指令用法示例

指令用法示例如图 2-7 所示。

(2) 使用注意事项

- ① 图 2-7 中的 X0 一旦接通，即使变成断开，Y0 也保持接通。X1 接通后，即使变成断开，Y0 也保持断开，对于 M、S 也是如此。
- ② 对同一元件可以多次使用 SET、RST 指令，顺序可任意，但对于外部输出，则只有最后执行的一条指令才有效。
- ③ 要使数据寄存器 D、计数器 C、积算定时器 T、变址寄存器 V、Z 的内容清零，也可用 RST 指令。

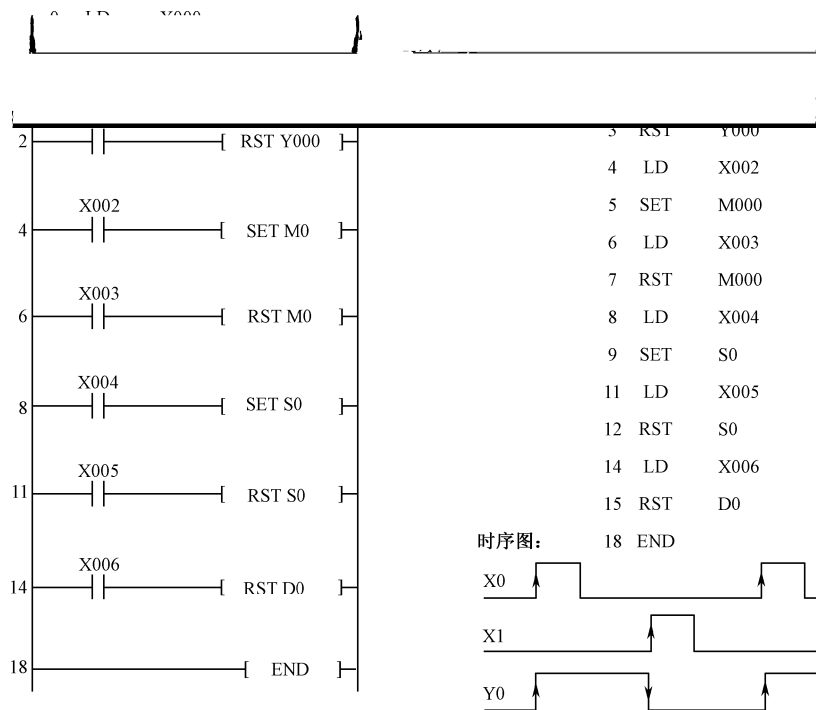


图 2-7 SET、RST 的使用示例

2. 常闭触点的输入信号处理

PLC 输入端口可以与输入设备不同类型的触点连接，但不同的触点类型设计出的梯形图程序不一样。

- ① PLC 外部的输入触点既可以接常开触点，也可以接常闭触点。当外部接常闭触点时，梯形图中的触点状态与继电器控制图中的状态是相反的。



② 教学中 PLC 的输入触点经常使用常开触点, 便于进行原理分析。但在实际控制中, 停止按钮、限位开关及热继电器等要使用常闭触点, 以提高安全保障。

③ 为了节省成本, 应尽量少占用 PLC 的 I/O 点, 因此有时也将 FR 常闭触点串接在其他常闭输入或负载输出回路中。



能力测试

设计一个能在两地启停控制的 PLC 控制系统。控制要求: 甲地按下启动按钮 SB1, 则电动机启动运行, 按下停止按钮 SB2, 则电动机停止。乙地按下启动按钮 SB3, 则电动机启动运行, 按下停止按钮 SB4, 则电动机停止运行; 任何时间若热继电器动作, 则电动机停止运行。

1. 设计梯形图 (40 分)

自行设计程序的梯形图。

2. 设计系统接线图 (20 分)

- ① 设计 PLC 接线图 (10 分)。
- ② 设计电动机的主电路图 (10 分)。

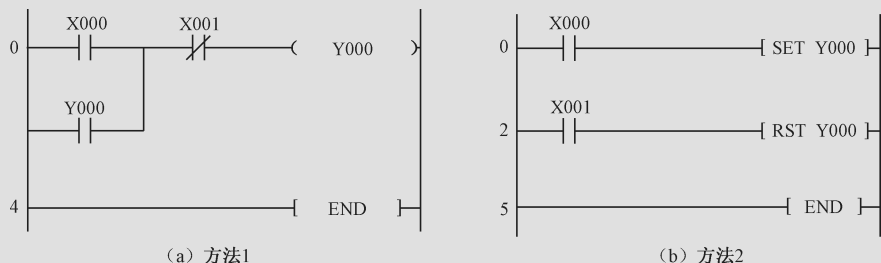
3. 系统调试 (40 分)

- ① 程序输入 (5 分)。
- ② 不接负载调试 (15 分)。
- ③ 带负载调试 (10 分)。
- ④ 其他测试 (10 分)。



研讨与练习

启—保—停电路可以用普通输入、输出触点与线圈完成 (如图 2-8 (a) 所示), 也可用 SET、RST 指令实现。若用 SET、RST 指令编程, 启—保—停电路包含了梯形图程序的两个要素: 一个是使线圈置位并保持的条件, 本例为启动按钮 X0 为 ON; 另一个是使线圈复位并保持的条件, 本例为停止按钮 X1 为 ON。因此, 其梯形图为启动按钮 X0、停止按钮 X1 分别驱动 SET、RST 指令。当要启动时, 按启动按钮 X0, 使输出线圈置位并保持; 当要停止时, 按停止按钮 X1, 使输出线圈复位并保持, 如图 2-8 (b) 所示。



(a) 方法1

(b) 方法2

图 2-8 电动机的启—保—停梯形图 (停止优先)



比较以上电路可知，方法 2 的设计思路更简单明了，是最佳设计方案。

注意：

① 在方法 1 的梯形图中，用 X1 的动断点；而在方法 2 中，用 X1 的动合点，但它们的外部输入接线却完全相同。

② 上述的两个梯形图都为停止优先，即如果启动按钮 X0 和停止按钮 X1 同时被按下，则电动机停止；若要改为启动优先，则梯形图如图 2-9 所示。

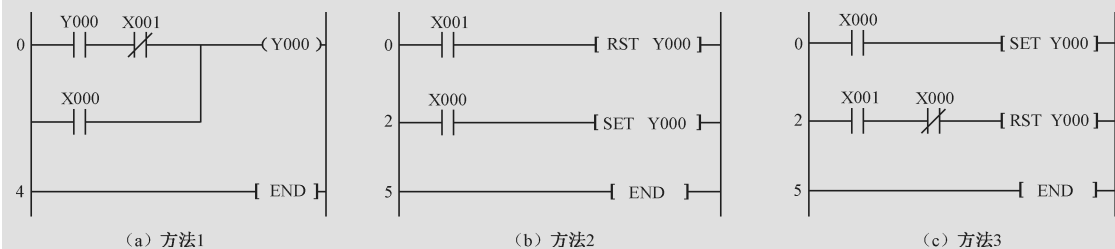


图 2-9 电动机的启—保—停梯形图（启动优先）



思考与练习

请读者分析以上梯形图，并体会其设计思路，然后将梯形图改写成对应的指令表程序。

2.2 三相异步电动机的正、反转控制



任务目标

- ① 能利用“启—保—停”电路中的基本逻辑指令、置位/复位指令及堆栈指令分别实现电动机正、反转运行。
- ② 能将已学指令应用于灯光控制电路等。
- ③ 进一步熟悉 PLC 的内部结构和外部接线方法。



任务分析

三相异步电动机正、反转继电器接触器控制运行电路如图 2-10 所示，KM1 为电动机正向运行交流接触器，KM2 为电动机反向运行交流接触器，SB2 为正转启动按钮，SB3 为反转启动按钮，SB1 为停止按钮，FR 为热保护继电器。当按下 SB2 时，KM1 的线圈保持吸合，KM1 主触点闭合，电动机开始正向运行，同时 KM1 的辅助常开触点闭合而使 KM1 的线圈保持吸合，实现了电动机的正向连续运行，直到按下停止按钮 SB1；反之，当按下 SB3 时，KM2 的线圈通电吸合，KM2 主触点闭合，电动机开始反向运行，同时 KM2 的辅助常开触点闭合而使 KM2 线圈保持吸合，实现了电动机的反向连续运行，直到按下停止按钮 SB1；KM1、KM2 线圈互锁确保不同时通电。本任务研究用 PLC 实现三相异步电动机的正、反转控制。

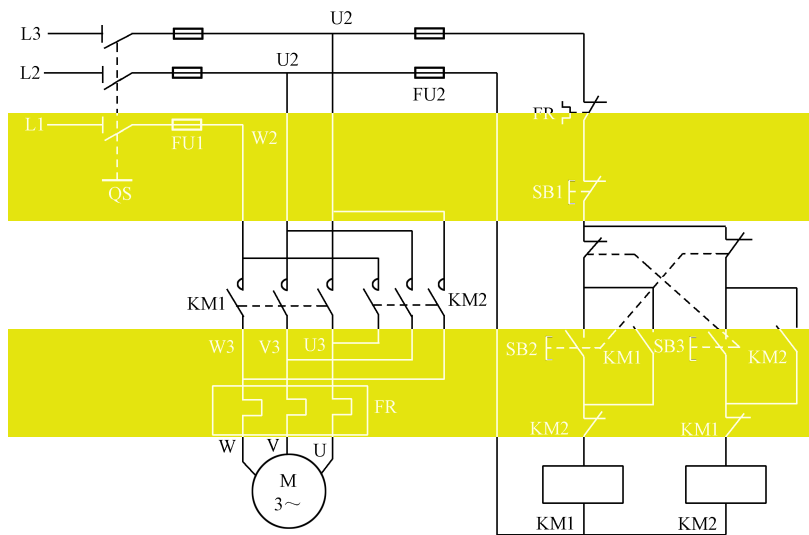


图 2-10 三相异步电动机正、反转继电器接触器控制运行电路



相关知识

2.2.1 电路块连接指令 ORB/ANB

电路块连接指令如表 2-6 所示。

表 2-6 电路块连接指令表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
ORB 电路块或	串联电路的并联连接		无	1
ANB 电路块与	并联电路的串联连接		无	1

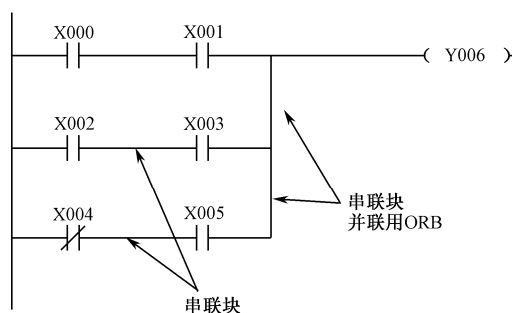
1. 用法示例

电路块连接指令的应用如图 2-11 和图 2-12 所示。

2. 使用注意事项

① ORB 是串联电路块的并联连接指令，ANB 是并联电路块的串联连接指令。它们都没有操作元件，可以多次重复使用。

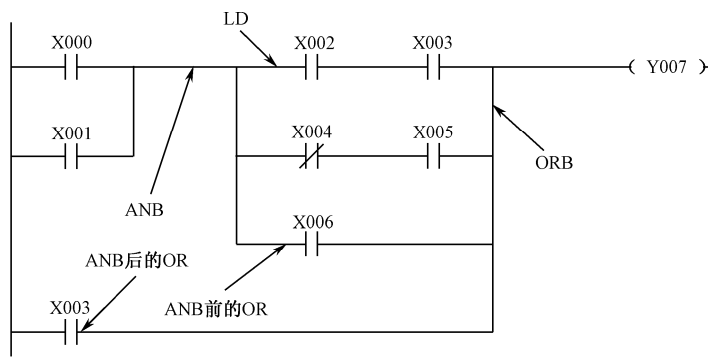
② ORB 指令将串联电路块与前面的电路并联，相当于电路块间右侧的一段垂直连线。并联的电路块的起始触点使用 LD 或 LDI 指令，完成了电路块的内部连接后，用 ORB 指令将它与前面的电路并联。



指令表:

0	LD	X000	5	LDI	X004
1	AND	X001	6	AND	X005
2	LD	X002	7	ORB	
3	AND	X003	8	OUT	Y006
4	ORB				

图 2-11 串联电路块并联示例



指令表:

0	LD	X000	6	ORB	← 串联块并联
1	OR	X001	7	OR	X006

图 2-12 并联电路块串联示例

③ ANB 指令将并联电路块与前面的电路串联，相当于两个电路之间的串联连线。串联的电路块的起始触点使用 LD 或 LDI 指令，完成了电路块的内部连接后，用 ANB 指令将它与前面的电路串联。

④ ORB、ANB 指令可以多次重复使用，但是连续使用时应限制在 8 次以下，所以最好按图 2-11 和图 2-12 所示的方法写指令。

2.2.2 多重输出电路指令 MPS/MRD/MPP

多重输出电路指令如表 2-7 所示。



表 2-7 多重输出电路指令表

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
MPS 进栈	进栈		无	1
MRD 读栈	读栈		无	1
MPP 出栈	出栈		无	1

1. 用法示例

多重输出电路指令的应用如图 2-13 和图 2-14 所示。

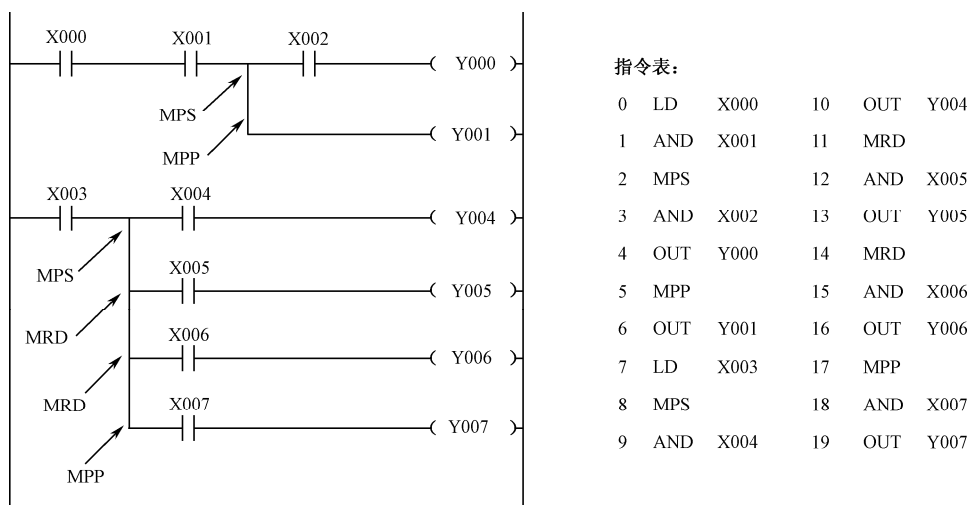


图 2-13 简单 1 层栈示例

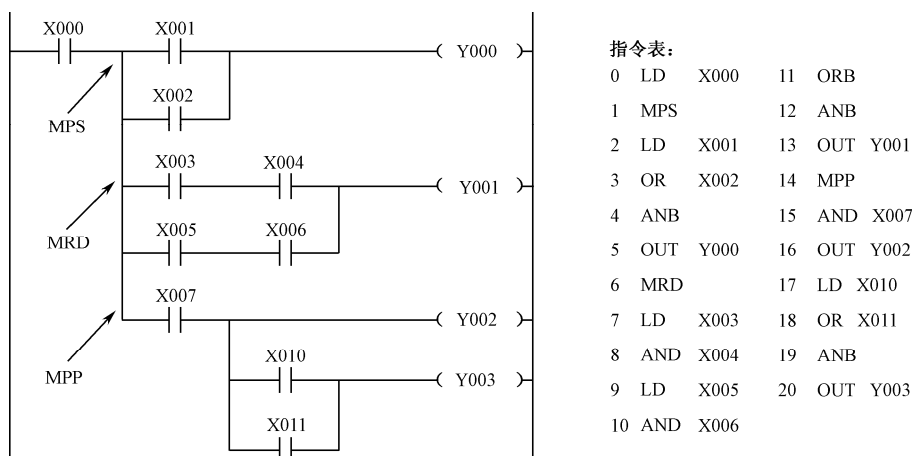


图 2-14 复杂 1 层栈示例



2. 使用注意事项

① **MPS** 指令可将多重电路的公共触点或电路块先存储起来，以便后面的多重输出支路使用。多重电路的第一个支路前使用 **MPS** 进栈指令，多重电路的中间支路前使用 **MRD** 读栈指令，多重电路的最后一个支路前使用 **MPP** 出栈指令。该组指令没有操作元件。

② **FX** 系列 PLC 有 11 个存储中间运算结果的堆栈存储器，堆栈采用先进后出的数据存取方式。每使用一次 **MPS** 指令，当时的逻辑运算结果压入堆栈的第一层，堆栈中原来的数据依次向下一层推移。

③ **MRD** 指令读取存储在堆栈最上层（即电路分支处）的运算结果，将下一个触点强制性地连接到该点。读栈后堆栈内的数据不会上移或下移。

④ **MPP** 指令弹出堆栈存储器的运算结果，首先将下一个触点连接到该点，然后从堆栈中去掉分支点的运算结果。使用 **MPP** 指令时，堆栈中各层的数据向上移动一层，最上层的数据在弹出后从栈内消失。

⑤ 处理最后一条支路时必须使用 **MPP** 指令，而不是 **MRD** 指令，且 **MPS** 和 **MPP** 的使用必须少于 11 次，并且要成对出现。

2.2.3 主控触点指令 MC/MCR

在编程时，经常会遇到许多线圈同时受一个或一组触点控制的情况，如果在每个线圈的控制电路中都串入同样的触点，将占用很多存储单元，主控指令可以解决这一问题。使用主控指令的触点称为主控触点，它在梯形图中与一般的触点垂直，主控触点是控制一组电路的总开关。主控触点指令如表 2-8 所示。

表 2-8 主控触点指令表

符号、名称	功 能	电路表示及操作元件	程 序 步
MC 主控	主控电路块起点		3
MCR 主控复位	主控电路块终点		2

1. 用法示例

主控触点指令的应用如图 2-15 所示。

2. 使用注意事项

(1) **MC** 是主控起点，操作数 **N**（0~7 层）为嵌套层数，操作元件为 **M**、**Y**，特殊辅助继电器不能用做 **MC** 的操作元件。**MCR** 是主控结束，主控电路块的终点，操作数为 **N**（0~7）。**MC** 与 **MCR** 必须成对使用。

(2) 与主控触点相连的触点必须用 **LD** 或 **LDI** 指令，即执行 **MC** 指令后，母线移到主控触点的后面，**MCR** 使母线回到原来的位置。

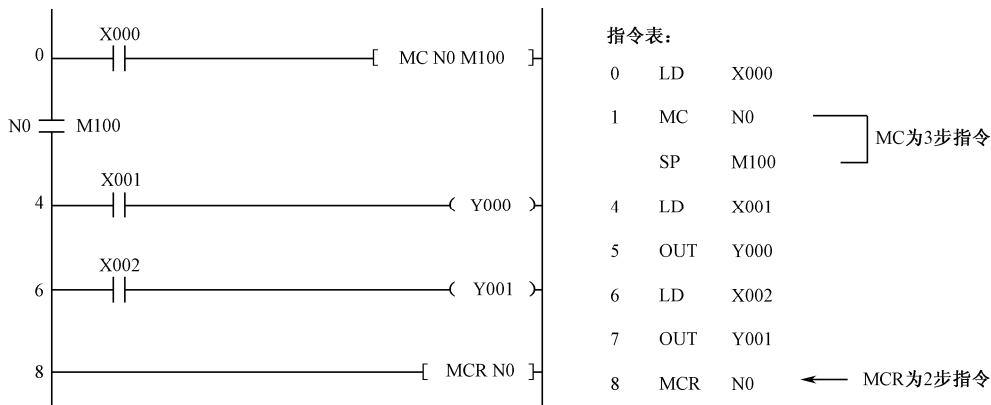


图 2-15 主控触点指令的应用示例

(3) 图 2-15 中 X0 的常开触点接通时, 执行从 MC 到 MCR 之间的指令; MC 指令的输入电路 (X0) 断开时, 不执行上述区间的指令。其中的积算定时器、计数器、用复位/置位指令驱动的软元件保持其当时的状态, 其余的元件被复位, 如非积算定时器和用 OUT 指令驱动的元件变为 OFF。

(4) 在 MC 指令内再使用 MC 指令时, 称为嵌套, 嵌套层数 N 的编号依次增大; 主控返回时用 MCR 指令, 嵌套层数 N 的编号依次减小。

2.2.4 逻辑运算结果取反指令 INV

逻辑运算结果取反指令如表 2-9 所示。

表 2-9 逻辑运算结果取反指令 INV

符号、名称	功 能	电 路 表 示	操 作 元 件	程 序 步
INV 取反	逻辑运算结果取反		无	1

INV 指令在梯形图中用一条 45° 的短斜线来表示, 它将使无该指令时的运算结果取反, 如运算结果为 0 则将它变为 1, 如运算结果为 1 则将它变为 0。如图 2-16 所示, 如果 X0 为 ON, 则 Y0 为 OFF; 反之, Y0 为 ON。

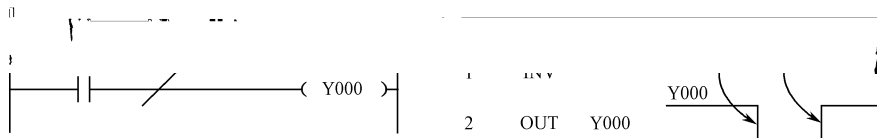


图 2-16 逻辑运算结果取反指令示例

2.2.5 空操作和程序结束指令 NOP/END

空操作和程序结束指令如表 2-10 所示。



表 2-10 空操作和程序结束指令表

符号、名称	功 能	电路表示	操作元件	程序步
NOP 空操作	无动作	无	无	1
END 结束	输入/输出处理，程序回到第 0 步		无	1

1. 空操作指令 NOP

① 若在程序中加入 NOP 指令，则改动或追加程序时，可以减少步序号的改变。

② 若将 LD、LDI、ANB、ORB 等指令换成 NOP 指令，电路构成将有较大幅度的变化，如图 2-17 所示。

③ 执行程序全清除操作后，全部指令都变成 NOP。

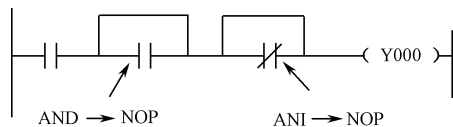


图 2-17 用 NOP 指令短路触点

2. 程序结束指令 END

PLC 按照循环扫描的工作方式，首先进行输入处理，然后进行程序处理，当处理到 END 指令时，即进行输出处理。所以，若在程序中写入 END 指令，则 END 指令以后的程序就不再执行，直接进行输出处理；若不写入 END 指令，则从用户程序存储器的第一步执行到最后一步。因此，若将 END 指令放在程序结束处，则只执行第一步至 END 之间的程序，可以缩短扫描周期。在调试程序时，可以将 END 指令插在各段程序之后，从第一段开始分段调试，调试好以后必须删除程序中间的 END 指令，这种方法对程序的查错也很有用处，而且，执行 END 指令时，也刷新警戒时钟。

2.2.6 梯形图的基本规则

梯形图作为 PLC 程序设计中最常用的编程语言，被广泛应用于工程现场的系统设计，为更好地使用梯形图语言，下面介绍梯形图的一些基本规则。

1. 线圈右边无触点

梯形图中每一逻辑行从左到右排列，以触点与左母线连接开始，以线圈、功能指令与右母线（可允许省略右母线）连接结束。触点不能接在线圈的右边，线圈也不能直接与左母线连接，必须通过触点连接，如图 2-18 所示。

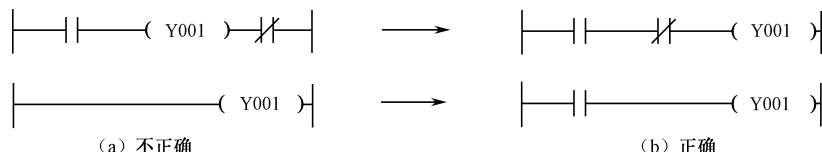


图 2-18 线圈右边无触点的梯形图

2. 触点可串可并无限制

触点可以用于串行电路，也可以用于并行电路，且使用次数不受限制，所有输出继电器



也都可以作为辅助继电器使用。

3. 线圈不能重复使用

在同一个梯形图中，如果同一元件的线圈被使用两次或多次，这时前面的输出线圈对外输出无效，只有最后一一次的输出线圈有效，所以，程序中一般不出现双线圈输出，如图 2-19 (a) 所示的梯形图必须改为如图 2-19 (b) 所示的梯形图。

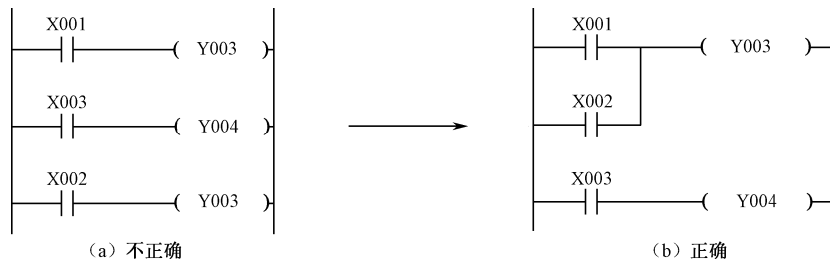


图 2-19 线圈不能重复使用的梯形图

4. 触点水平不垂直

触点应画在水平线上，不能画在垂直线上。如图 2-20 (a) 所示的 X3 触点被画在垂直线上，所以很难正确识别它与其他触点的逻辑关系，因此，应根据其逻辑关系改为如图 2-20 (b) 或图 2-20 (c) 所示的梯形图。

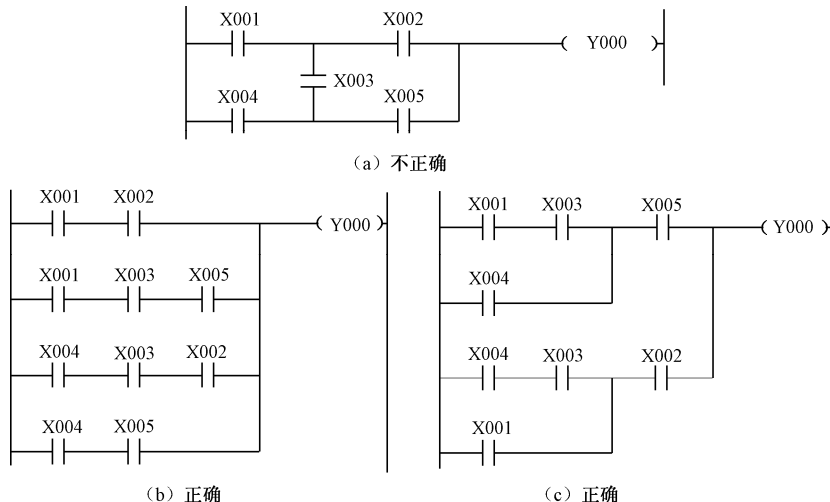


图 2-20 触点水平不垂直的梯形图

5. 触点多上并左

如果有串联电路块并联，应将串联触点多的电路块放在最上面；如果有并联电路块串联，应将并联触点多的电路块移近左母线，这样可以使编制的程序简洁，指令语句少，如图 2-21 所示。