



## 第二章 **PLC**控制与编程基础

- 输入输出元件
- PLC的工作原理
- PLC编程语言



# PLC的工作原理

- PLC的循环扫描工作过程
- PLC的I/O滞后现象
- PLC对输入点计数的频率问题



# A 循环扫描工作过程

## 1. PLC的工作过程

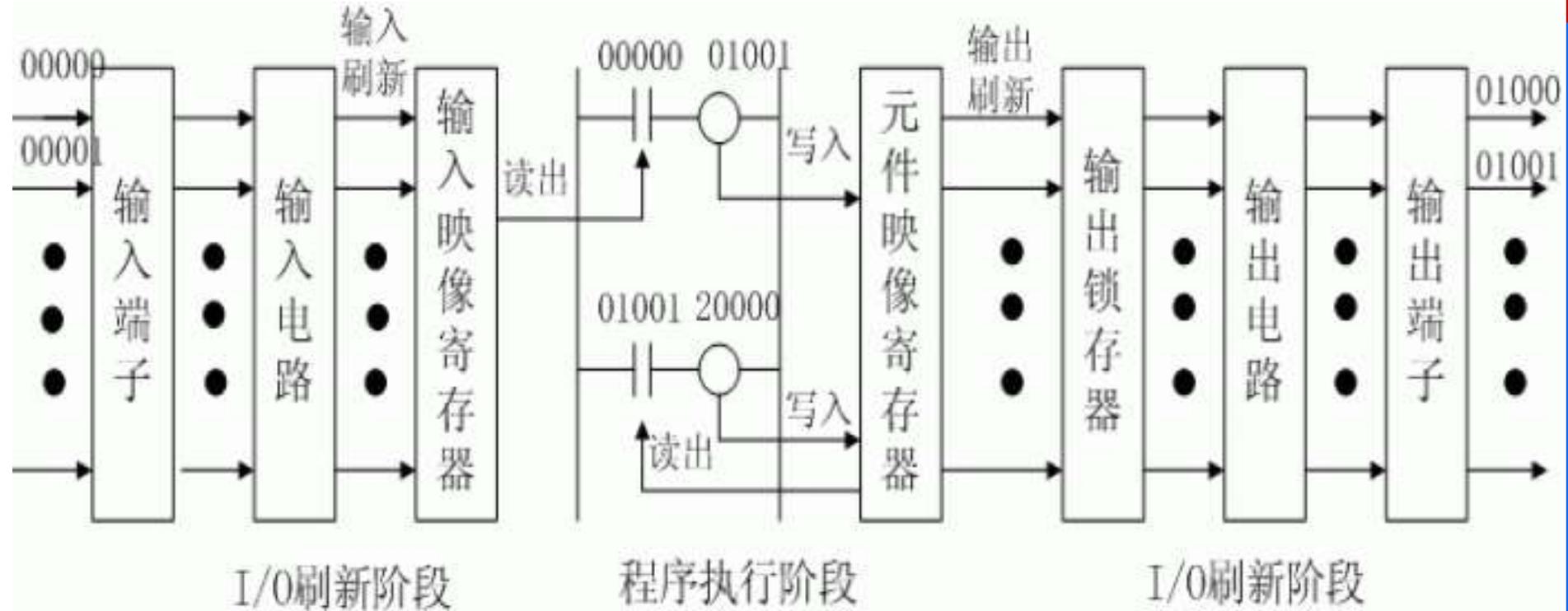
欧姆龙公司的小型机CPM1A

完成的任务如下：

- (1) 公共处理  
(硬件检查、异常报警)
- (2) 程序执行
- (3) 扫描周期计算处理
- (4) I/O刷新
- (5) 外设端口服务



## 2.信号传递过程（从输入到输出）



- I/O刷新阶段---CPU从输入电路的输出端读出各路状态，并将其写入输入映像寄存器；
- 程序执行阶段---CPU从输入映像寄存器和元件映像寄存器中读出各继电器的状态，并根据此状态执行用户程序，执行结果再写入元件映像寄存器中；
- 紧接着的下一个I/O刷新阶段---将输出映像寄存器的状态写入输出锁存电路，再经输出电路传递到输出端子，从而控制外接器件动作。



### 3.死循环自诊断功能

- PLC内部设置了一个监视定时器WDT,其定时时间可由用户设置为大于用户程序的扫描周期,PLC在每个扫描周期的公共处理阶段将监视定时器复位。
- 正常情况下,监视定时器不会动作,如果由于CPU内部故障使程序执行进入死循环,那么,扫描周期超过监视定时器的定时时间时,监视定时器动作,运行停止,以示用户。



## B I/O滞后现象

### 1. I/O滞后现象的原因

- (1) 输入滤波器有时间常数
- (2) 输出继电器有机械滞后
- (3) PC循环操作时，进行公共处理、I/O刷新和执行用户程序等产生扫描周期
- (4) 程序语句的安排，也影响响应时间。

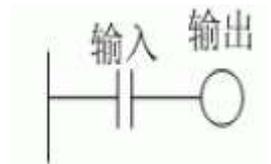


## 2. I/O响应时间

----从输入触点闭合到输出触点闭合有一段延迟时间，称为I/O响应时间

(1)最小I/O响应时间

(2)最大I/O响应时间

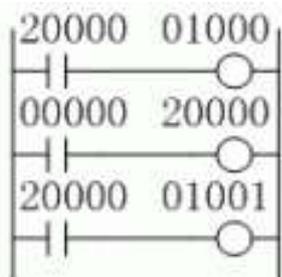




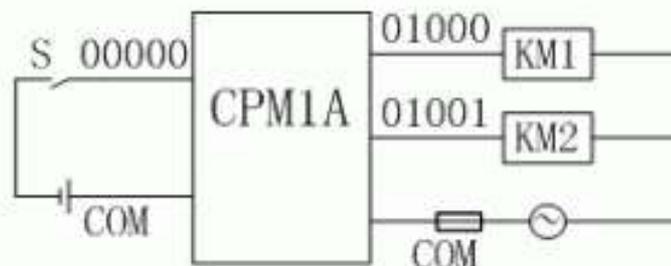
$$\text{最大I/O响应时间} = \text{输入ON延时} + \text{扫描时间} \times 2 + \text{输出ON延时}$$



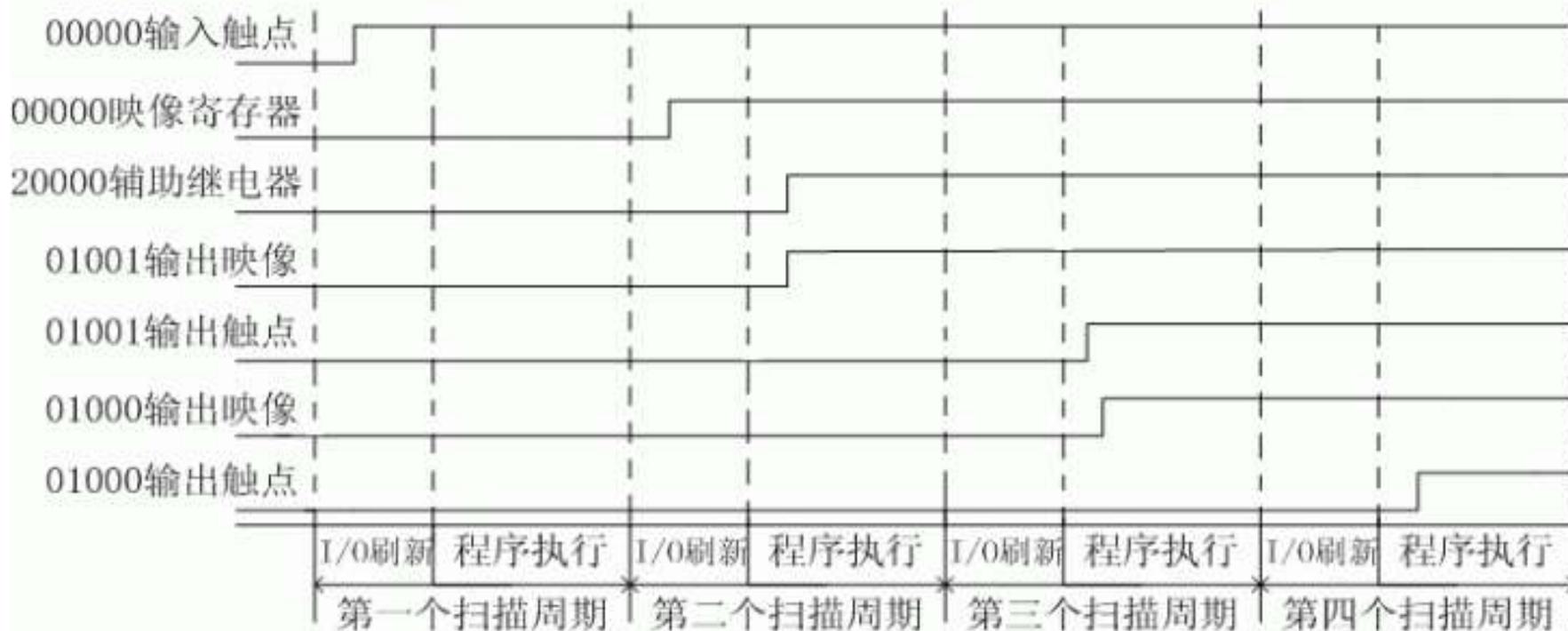
# 程序语句安排影响I/O响应时间?



(a) 梯形图



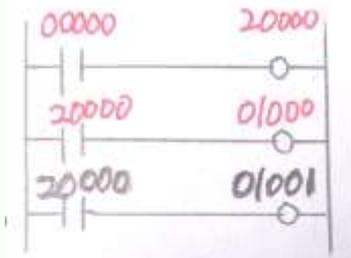
(b) 接线图



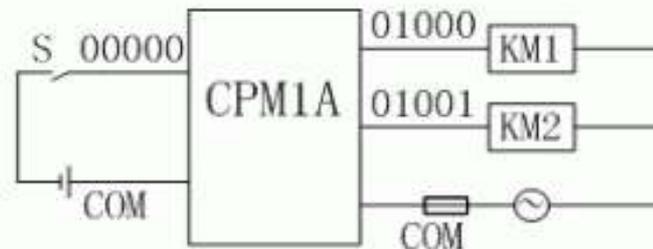
(c) I/O波形图



# 程序语句安排影响I/O响应时间?



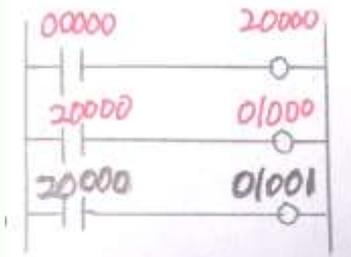
(a) 梯形图



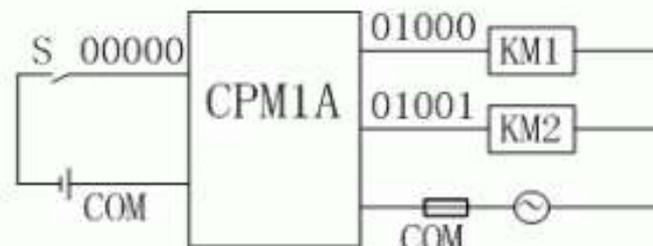
(b) 接线图



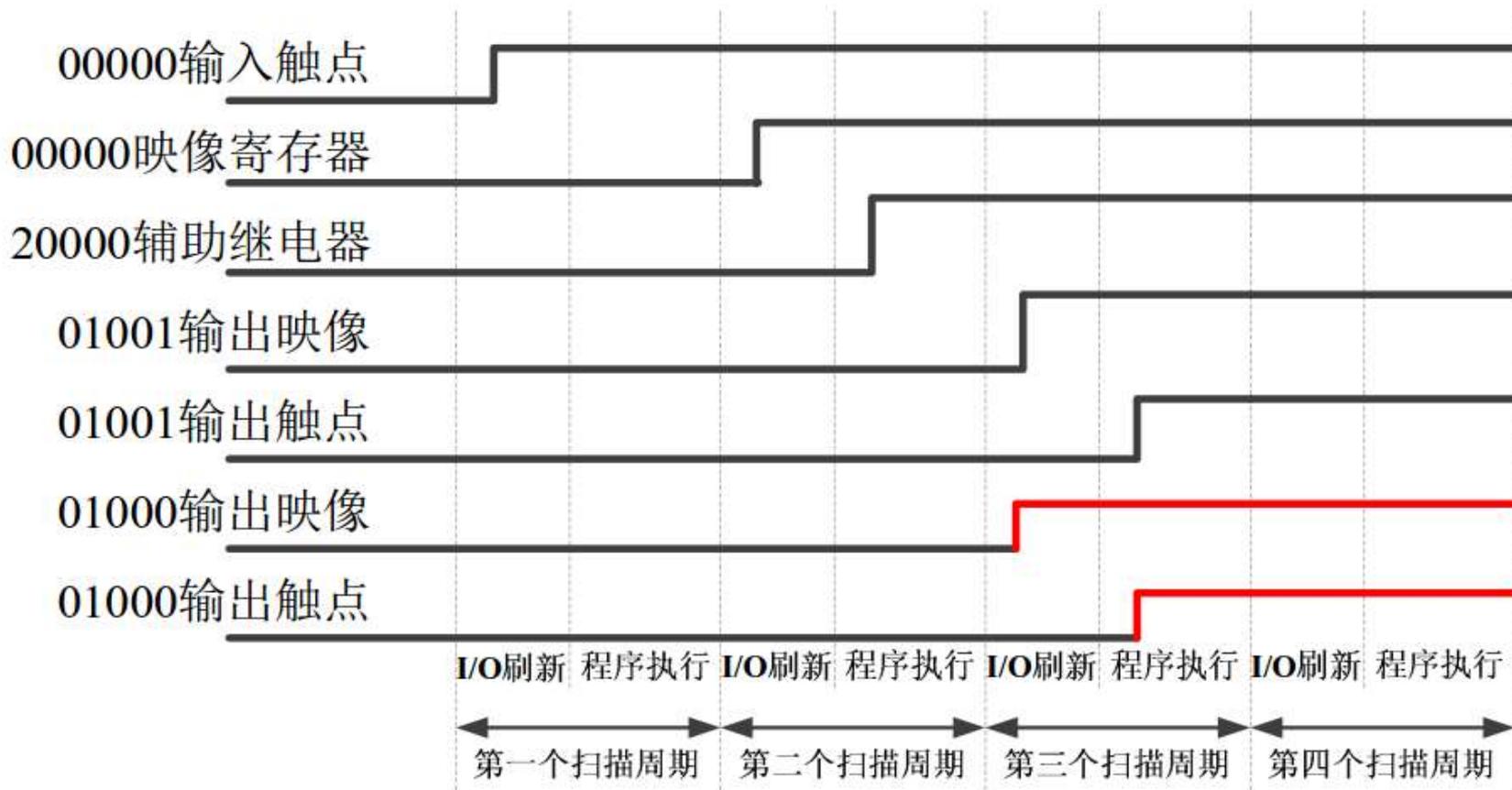
# 程序语句安排影响I/O响应时间?



(a) 梯形图



(b) 接线图



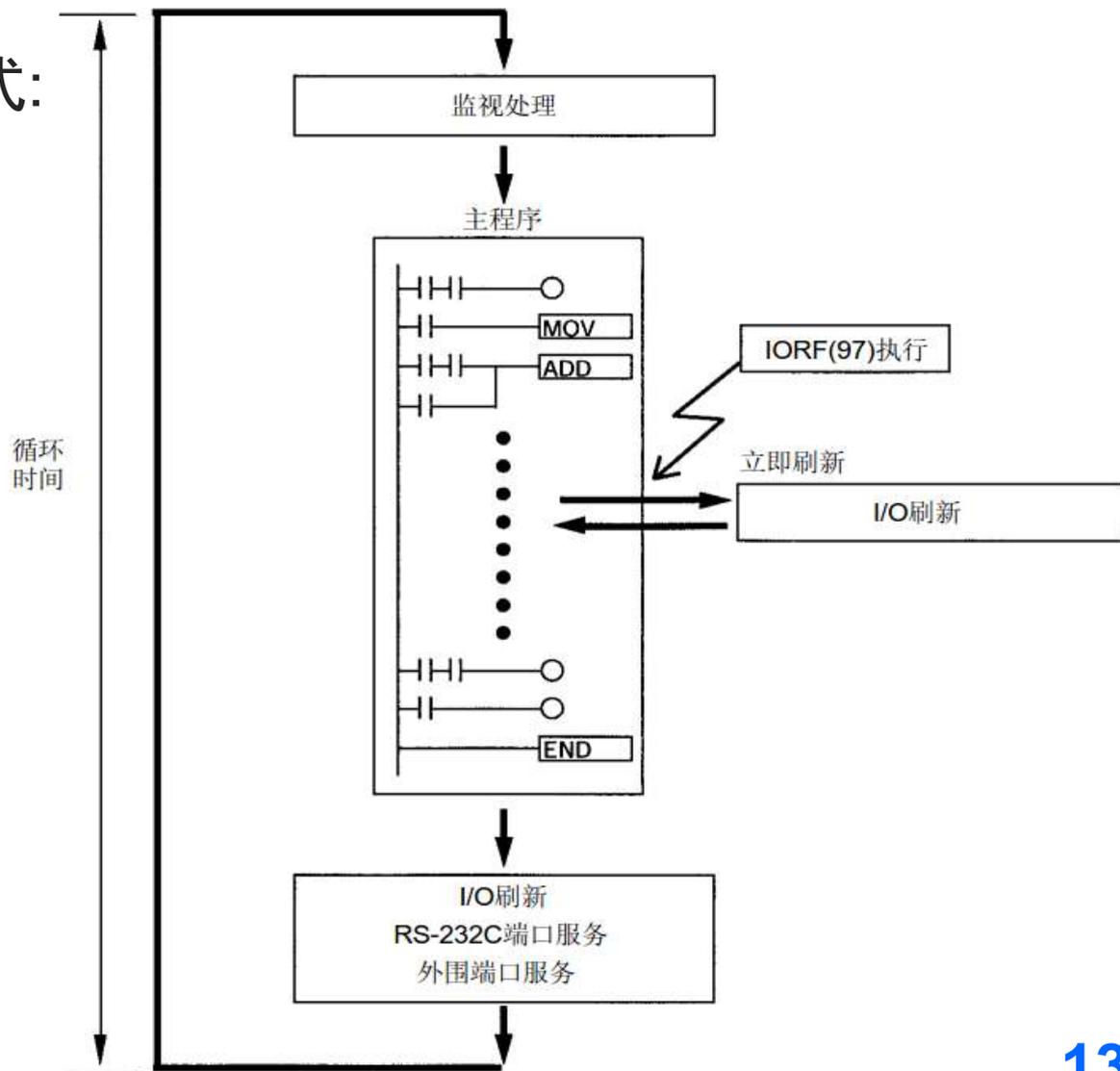


# C PLC对输入点计数的频率问题

PLC计数有两种方式:

## 1 高速计数

-----在高速计数方式下，输入信号不经输入滤波器直接送到CPU，计数不受输入滤波器时间常数、扫描周期的影响，计数频率可以很高。





## 2 普通计数

在普通计数方式下，输入信号经过滤波器后在PLC扫描周期的I/O刷新阶段被CPU读入，因此，计数频率受输入滤波时间常数和扫描周期限制，不可能很高。



## 2 普通计数

为了保证CPU能够可靠地读入开关接通或断开的状态，不丢失脉冲数，输入滤波后的信号其有效高电平和低电平持续时间不能少于一个扫描周期，即应满足

$$T' \geq \tau + T_S$$

$T'$  -----输入开关接通或断开的时间

$\tau$  -----输入滤波器时间常数

$T_S$  -----PLC的扫描周期



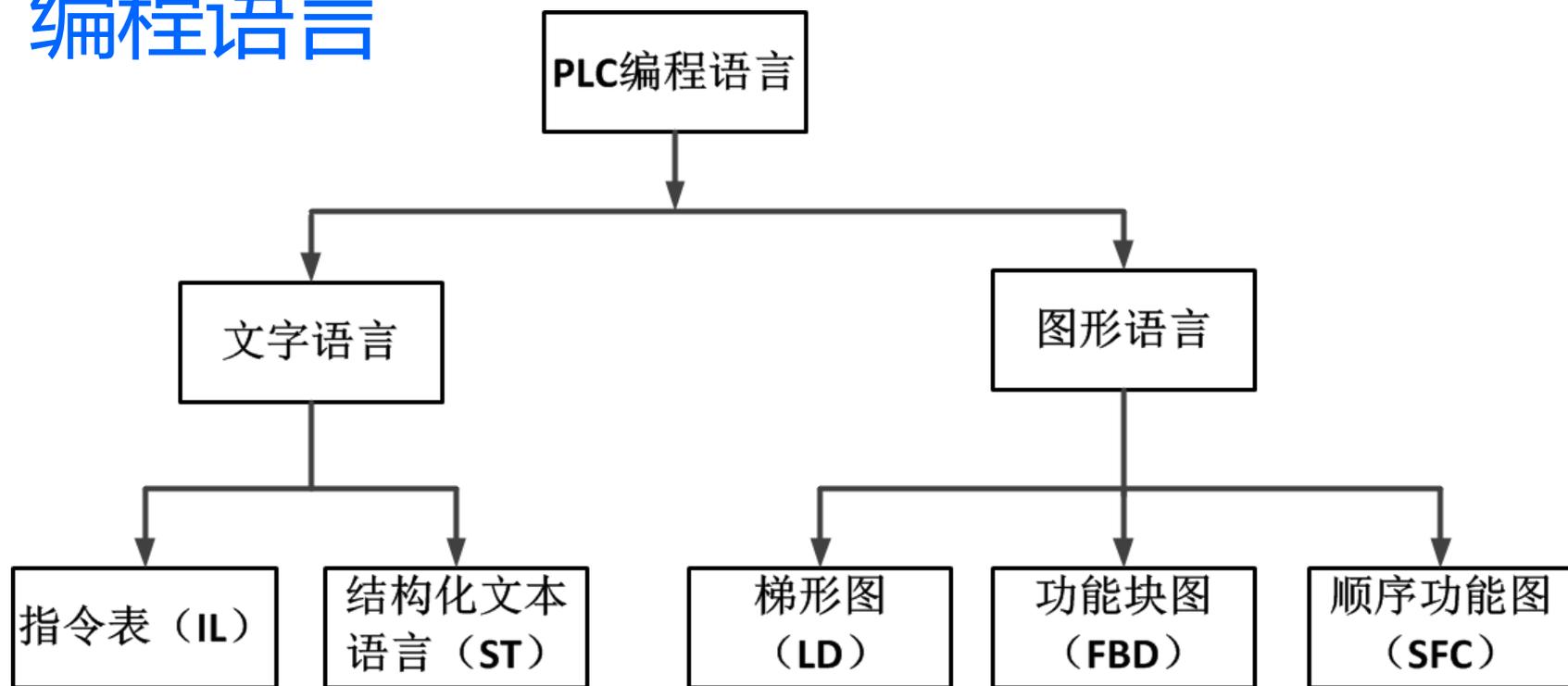
## 以CPM1A普通计数为例

$$f_{max} = \frac{1}{2T'_{min}} \\ = \frac{1}{2(\tau + T_s)} = \frac{1}{2 \times 10ms} = 50Hz$$

- 公共处理时间+I/O刷新时间=2ms (1)
- 程序执行时间=0ms (2)
- CPM1A的扫描周期  $T_s = (1) + (2) = 2ms$
- 输入滤波器时间常数  $\tau = 8ms$  (缺省设置)



# 编程语言



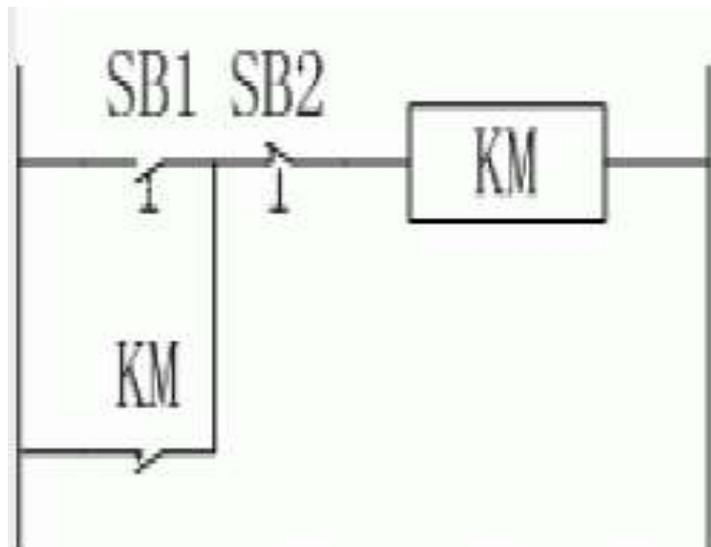
IEC 61133-3规定了二大类编程语言：文本化编程语言和图形化编程语言。前者包括指令清单语言（IL）和结构化文本语言（ST），后者包括梯形图语言（LD）和功能块图语言（FBD）。

在北美和日本，普遍运用梯形图（LD）语言编程；在欧洲，则使用功能块图（FBD）和顺序功能图（SFC）；德国和日本，又常常采用指令表（IL）对PLC进行编程。为了扩展PLC的功能，特别是加强它的数据处理、文字处理，以及通信功能的能力，许多PLC还允许使用高级语言（如BASIC语言、C语言），即ST。

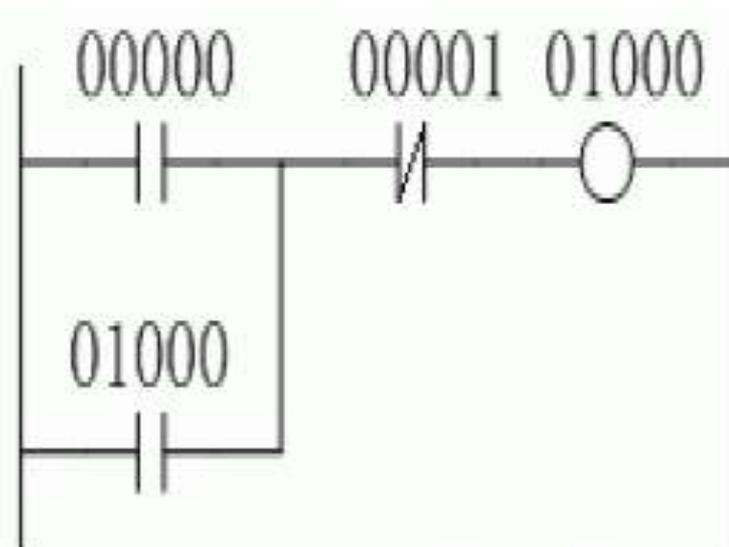


# 梯形图

----PLC的主要编程语言



(a) 电器控制梯形图



(b) PLC梯形图

PLC梯形图在形式上类似于继电器控制梯形图



# 梯形图的主要特点

(1) 梯形图中的继电器不是物理继电器，每个继电器是映像寄存器中的一位，被称为“软继电器”。  
因此：

一、用户程序解算时，输入触点和输出线圈的状态是从 I/O映像寄存器中读取的，而不是解算时现场开关的实际状态。

二、继电器状态可以反复读取，因此可以认为继电器有无限多个常开触点和常闭触点，程序中 can 反复引用。



# 梯形图的主要特点

(1) 梯形图中的继电器不是物理继电器，每个继电器是映像寄存器中的一位，被称为“软继电器”。

(2) 梯形图是PLC形象化的编程手段，梯形图两端是没有任何电源可接的。梯形图中并没真实的电流，而仅是“概念”电流，且只能从左向右流动。

(3) 输入继电器供PLC接收外部输入信号，因此，在梯形图中只出现输入继电器的触点，而不出现输入继电器的线圈。

(4) 输出继电器供PLC输出控制用。他通过开关量模块对应的输出开关（晶体管、双向晶闸管或继电器触点）去驱动外部负载。

(5) 当PLC处于运行状态时，PLC对梯形图是按扫描方式顺序执行程序。

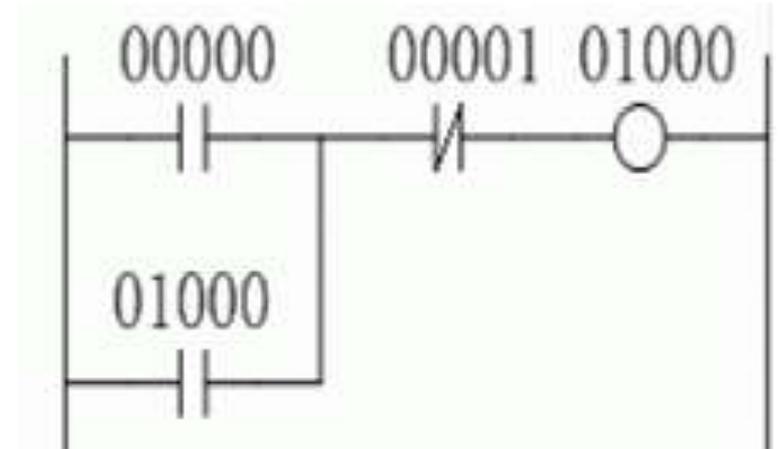


# 语句表 (指令表)

----一种与汇编类似的语言

语句表使用的助记符

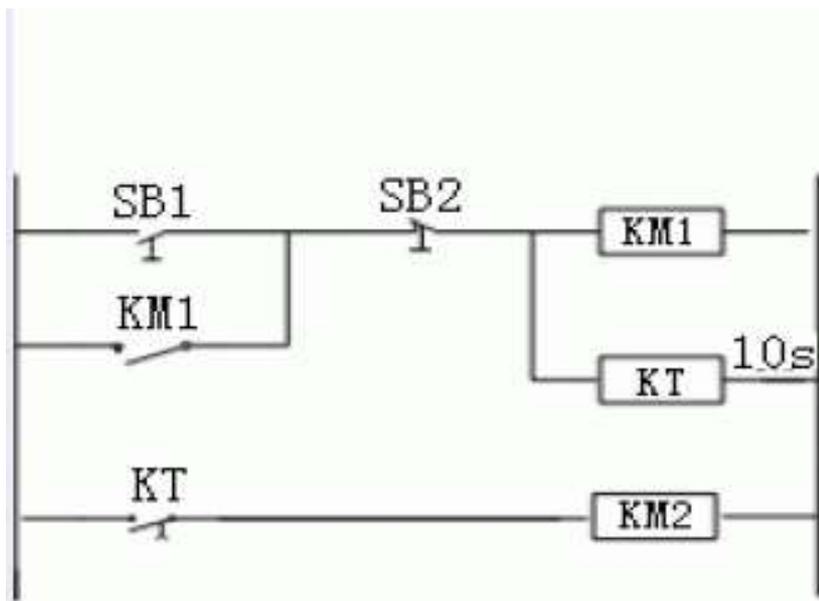
```
LD 00000  
OR 01000  
AND NOT 00001  
OUT 01000
```



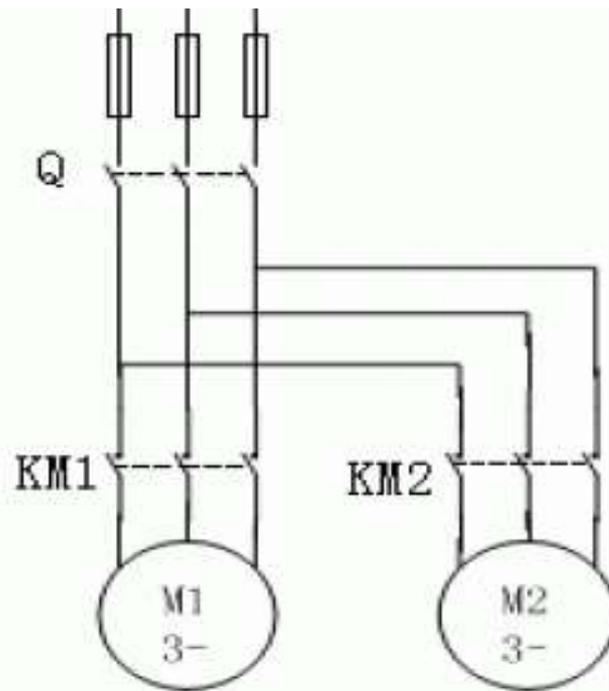
PLC的语句表达：操作码+操作数



# 另一个梯形图例子



(a) 控制线路

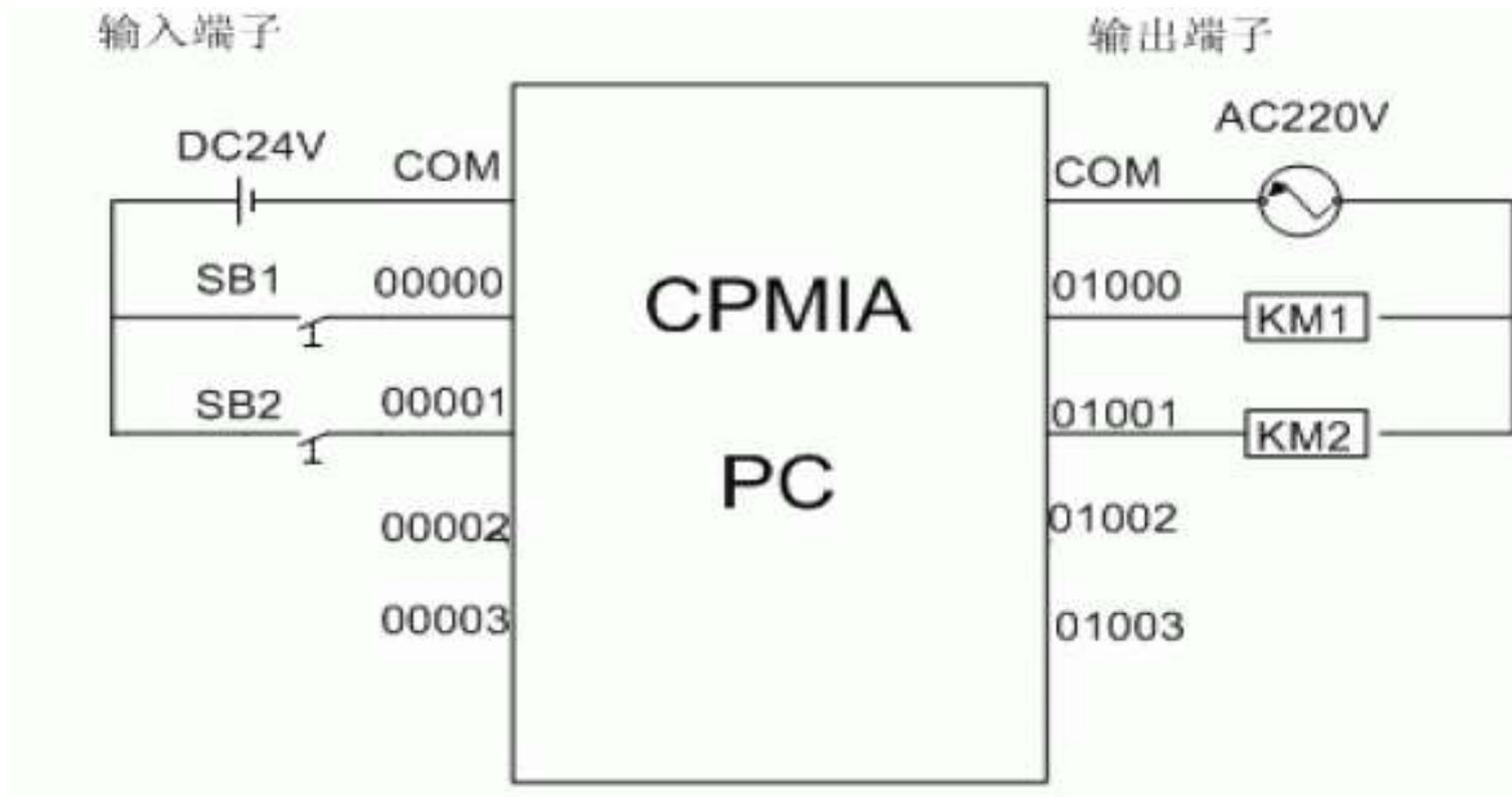


(b) 电机主电路

控制功能如下：按下启动按钮SB1，电机M1开始运转，过10秒钟后，电机M2开始运转；按下停止按钮SB2，电机M1、M2同时停止运转。



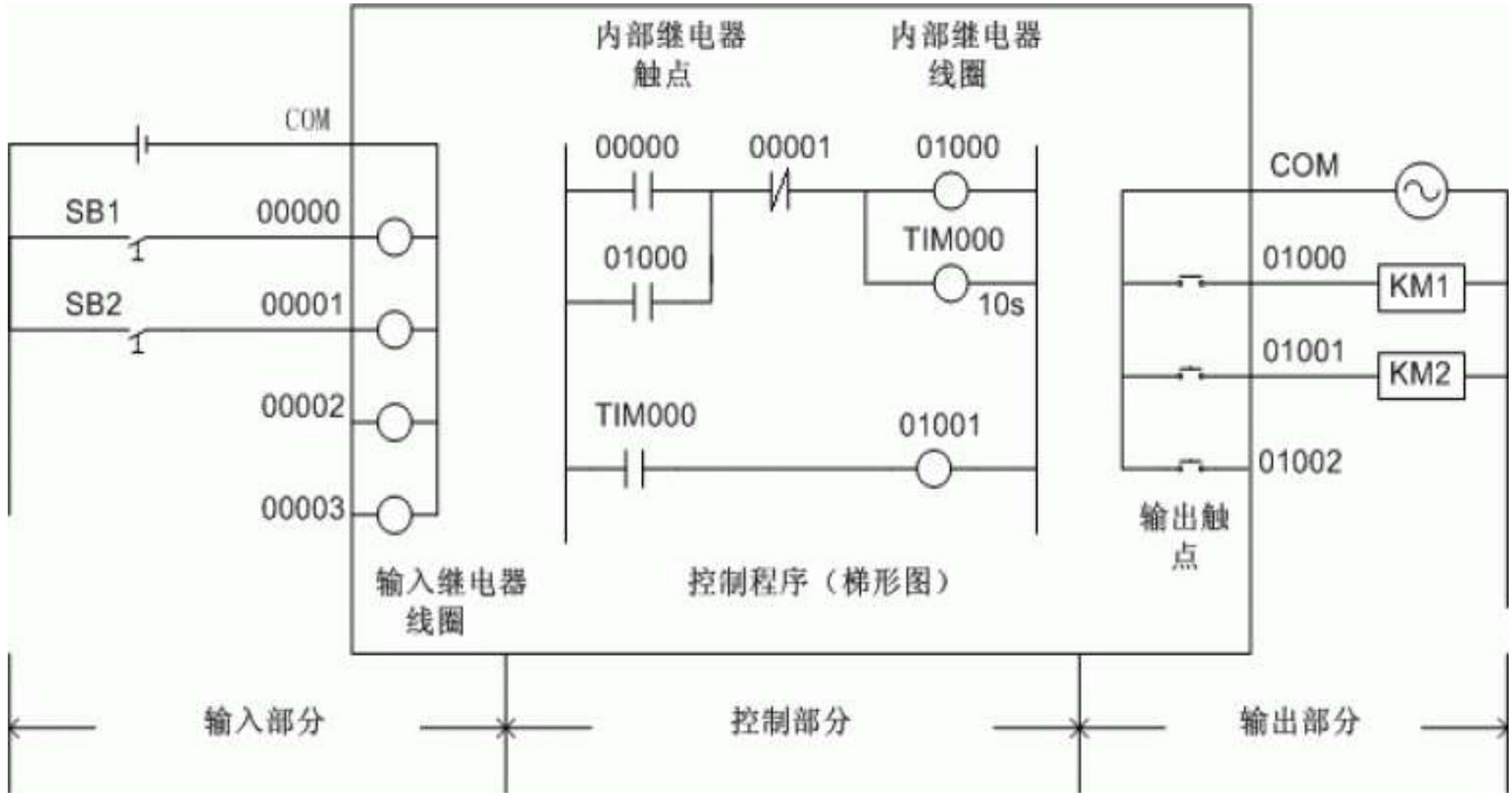
# 另一个梯形图例子



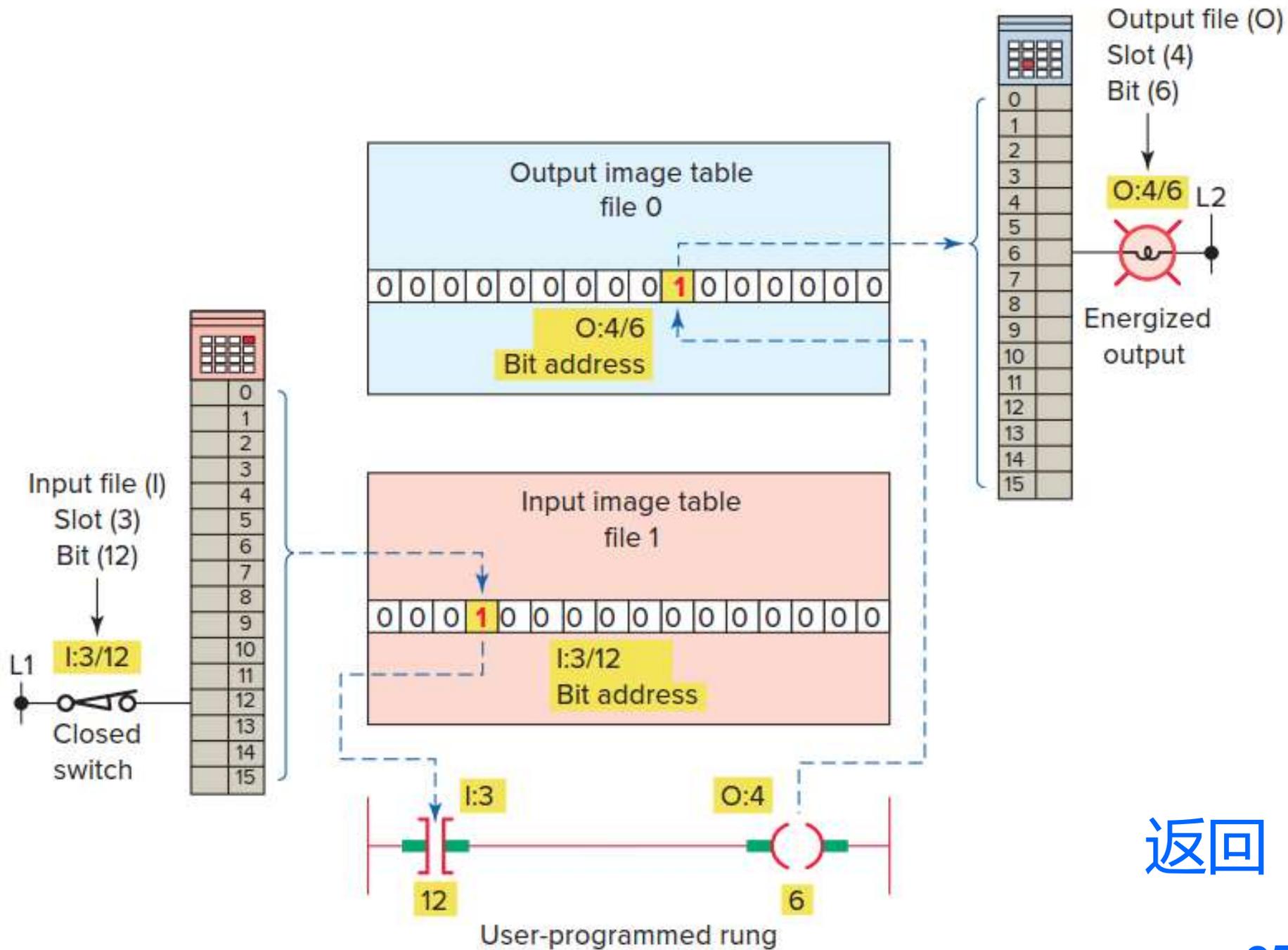
控制功能如下：按下启动按钮SB1，电机M1开始运转，过10秒钟后，电机M2开始运转；按下停止按钮SB2，电机M1、M2同时停止运转。



# 基于梯形图的控制等效电路图



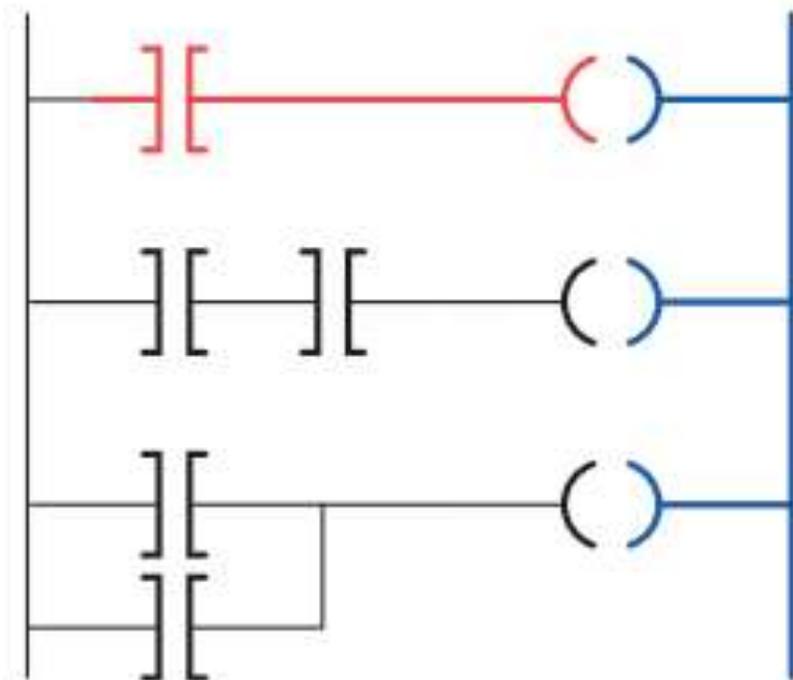
当按下SB2时，输入继电器00001的线圈通电，00001得常闭触点断开，01000、TIM000的线圈均断电，01001的线圈也断电，01000、01001两个硬输出触点随之断开，KM1、KM2断电，M1、M2停转。



返回



Horizontal scanning order



End of ladder

Vertical scanning order

Return for next scan



返回

扫描方式可以是纵向的或者横向的



结构式文件编程语言是支援块状结构（block structured）的高阶语言，以Pascal为基础，语法也类似Pascal。结构式文件编程语言类似于PASCAL及C语言，因此可利用与微电脑及个人电脑相同的程式设计技术，进行阶梯图所难以执行的复杂计算，完成程式的建立。常用的程式及回路可透过FB（功能区块）的建立轻易地重复利用。

返回