

PLC与变频应用技术

主讲：陈丽娟

项目三：基于PLC的生产线次品统计系统

任务目标

- ❑ 任务1：利用算术运算指令的统计次品数量
- ❑ 任务2：利用传送比较指令实现次品数量超标警
- ❑ 任务3：利用触点比较指令实现次品数量超标警
- ❑ 任务4：利用程序流程类指令实现控制模式切换

知识、能力目标

- ❑ 1、掌握功能指令的基本规则、表示方式、数据长度、位组件、执行方式和变址操作等。
- ❑ 2、掌握各类功能指令及运用功能指令编程的方法。
- ❑ 3、掌握PLC控制系统的设计原则和设计步骤。
- ❑ 4、学会用PLC解决实际问题的思路，进一步熟悉编程软件的使用方法，通过练习，提高编程技巧。

项目三：基于PLC的生产线次品统计系统

任务1：利用算术运算指令的统计次品数量

学习目标

- 1、掌握功能指令的基本规则、表示方式、数据长度、位组件、执行方式和变址操作等。
- 2、掌握算术运算指令的编程方法。
- 3、掌握功能指令设计程序的原则和步骤。
- 4、掌握ADD、SUB、MUL、DIV、DEC、INE指令的使用。

PLC的功能指令系统

【项目引入】

- ❑ 对于一般的传统工业控制电路，利用前面学过的基本指令和步进指令来编程就可以满足要求了，基本指令和步进指令主要用于逻辑处理。
- ❑ **PLC**作为工业控制用的计算机，仅仅进行逻辑处理是不够的，现代工业控制在许多场合需要进行数据处理，用来对数据的传送、运算、变换及程序控制等，这使**PLC**成为真正意义上的计算机。
- ❑ 本模块介绍功能指令（应用指令）。许多功能指令有强大的功能，往往一条指令就能实现几十条基本指令才可以实现的功能，还有很多功能指令具有基本指令难以实现的功能。

PLC的功能指令系统

分类

- 1、程序流程指令：如CJ(条件转移)、CALL(子程序调用)。
 - 2、传送与比较指令：如CMP(比较)、MOV(传送)。
 - 3、四则运算指令：如ADD(二进制加法)、WOR(逻辑字或)。
 - 4、循环移位指令：如ROR(循环右移)、SFTR(位右移)。
 - 5、数据处理指令：如ZRST(批次复位)、DECO(译码)。
 - 6、高速处理指令：如MTR(矩阵输入)、PLSY(脉冲输出)。
 - 7、方便指令：如IST(初始化)、SORT(数据排序)。
 - 8、外围设备（I/O）指令：如TKY(数字键0-9输入)、FROM、TO。
 - 9、外围设备（SER）指令：如BS(串型数据传送)、CCD(校验码)。
- 等等.....

预备知识：1、数据的表示方法

FX3U系列PLC提供的数据表示方法有：

——位元件、字元件、位软元件组合等。

□位软元件：只有接通/断开两种状态的元件；如 X/Y/M 等。

位元件只能逐个编程。——用基本指令编程。

□字软元件：处理数据的元件；如数据寄存器 D等；字元件可以批量操作——用功能指令。

□位软元件组合（属于字元件）：

——4个“位元件”构成一个组，用Kn Mi表示，

Kn表示位元件的组数(十进制)，

M 表示位元件的类型，

i 表示位元件的首地址，一般从0号开始。

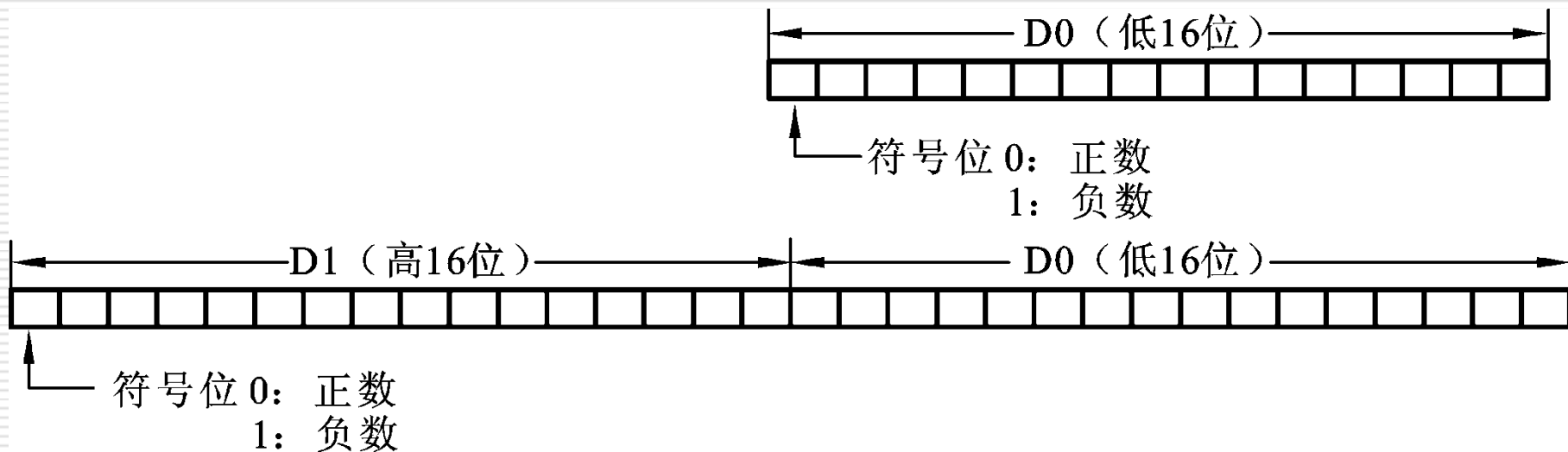
如：K1X0 K2Y0 K3M4 等



预备知识：1、数据的表示方法

一、数据寄存器（D）

- 用来储存数据和参数。
- 基本长度是16位(最高位是符号位)，如将两个相邻数据寄存器组合，可存储32位(最高位为符号位)的数值数据。



预备知识：1、数据的表示方法

数据寄存器（D）

- **通用数据寄存器：D0~D199，共200点。**
只要不写入其它数据，已写入的数据不会变化。
但当**PLC**状态由运行→停止时，全部数据均清零。
- **断电保持数据寄存器：D200~D7999，共7800点。**
只要不改写，原有数据不会丢失。
(其中**D200~D511**，可修改参数为非断电保持；)
(**D200~D7999**，为断电保持专用，不可修改参数)
- **特殊数据寄存器：D8000~D8255，共256点。**
这些数据寄存器供监视**PLC**中各种元件的运行方式用

预备知识：1、数据的表示方法

常数 (K/H)

●常数的表示方法：

K：表示十进制常数，如常数123 = K123

H：表示十六进制常数，如常数345 = H159

●各数制间的转换。

●二进制 $\longleftrightarrow$ 十进制：

●二进制 $\longleftrightarrow$ 十六进制：

●十进制 $\longleftrightarrow$ 十六进制：

预备知识：2、功能指令格式

❖ 功能指令是由**操作码**与**操作数**两部分组成。

操作码：表示指令的功能。

可以用指令助记符或指令编号表示。

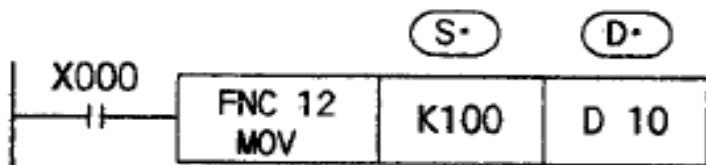
指令编号：FNC00~FNC299，有些编号是预留的。

操作数：指明参与操作的对象。

❑ **源操作数S**：执行指令后数据不变的操作数。

❑ **目标操作数D**：执行指令后数据被刷新的操作数。

❑ **其它操作数m、n**：用作补充注释的常数，
用K（十进制）和H（十六进制）表示。



预备知识：2、功能指令格式

功能指令 的 前缀后缀

前缀（D）：数据长度指示；

无**D**：表示**16**位数据操作；

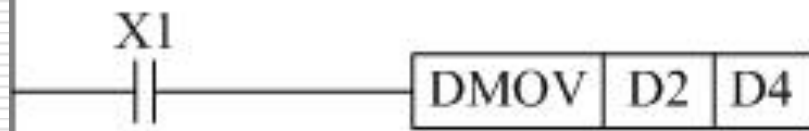
有**D**：表示**32**位数据操作。

FX3u系列PLC提供的数据长度分为16位和32位两种。默认为16位二进制数据；如果是32位数据，则在操作码前加加上前缀 **D (Double)** 表示，此时只写出元件的低16位数据地址，高16位数据放在比首地址高一位的地址中。

16位
与32
位数
据传
送



$(D0) \rightarrow (D1)$



$(D3D2) \rightarrow (D5D4)$

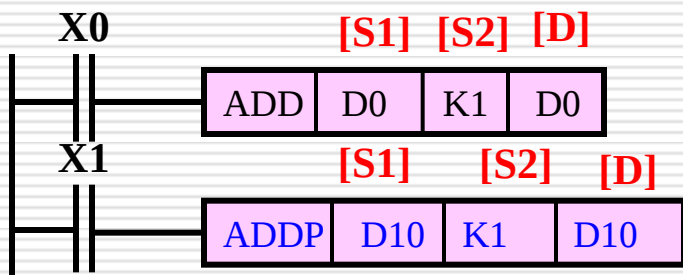
预备知识：2、功能指令格式

功能指令 的 前缀后缀

后缀（P）：脉冲/连续执行指示；

无**P**：表示每个扫描周期都执行一次。

有**P**：表示为脉冲执行指令，条件满足时执行一个扫描周期



连续执行方式与脉冲执行方式

□在X0接通时，**每个**扫描周期都执行一次（D0增加1）；

□在X1接通时，只在程序**第一个**扫描周期执行一次；

特别在算术运算之类中，通常使用脉冲执行方式。

第二类、算术运算类指令及其应用

四 则 运 算	20	ADD	BIN 加法
	21	SUB	BIN 减法
	22	MUL	BIN 乘法
	23	DIV	BIN 除法
	24	INC	BIN 加 1
	25	DEC	BIN 减 1
	26	WAND	逻辑字 与
	27	WOR	逻辑字 或
	28	WXOR	逻辑字 异或
	29	NEG	求补码

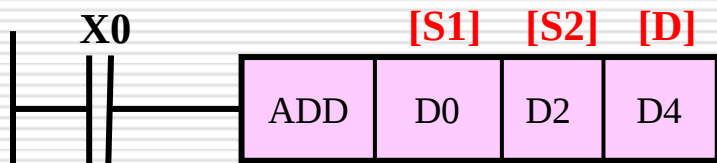
1. 加/减/乘/除运算指令

加法: **ADD** (编号FNC20)

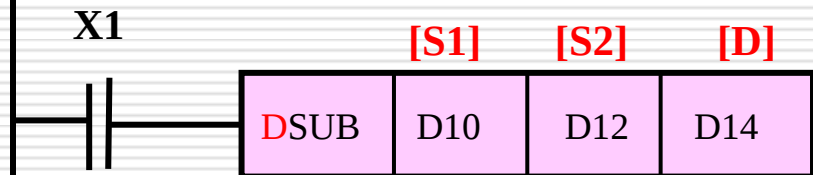
乘法: **MUL** (编号FNC22)

减法: **SUB** (编号FNC21)

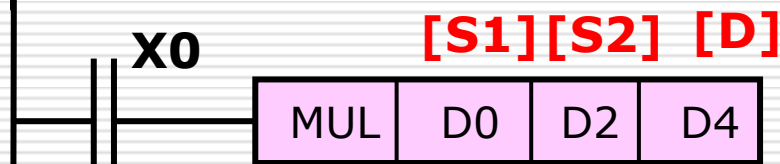
除法: **DIV** (编号FNC23)



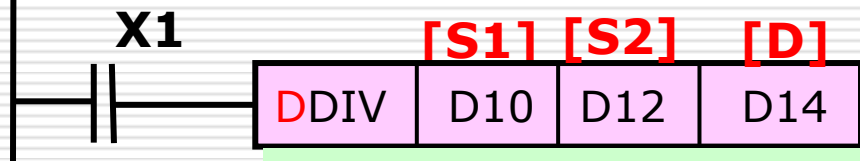
$$(D0) + (D2) \longrightarrow (D4)$$



$$(D11D10) - (D13D12) \longrightarrow (D15D14)$$



被乘数	乘数	积
$(D0) \times (D2)$		$(D5D4)$



被除数	除数	商	余数
$(D11D10) \div (D13D12)$		$(D15D14) \cdots (D17D16)$	

算术运算指令

□说明

- 算术运算指令是代数运算，数据的最高位为符号位。
- 进行16位运算时，数据范围为 $-32768 \sim +32767$ ；
- 32位运算时（在指令前加前缀“D”），
数据范围为 $-2147483648 \sim +2147483647$ 。
- 该指令可以进行连续/脉冲执行（加后缀 P）方式。
（通常都要使用脉冲执行方式）

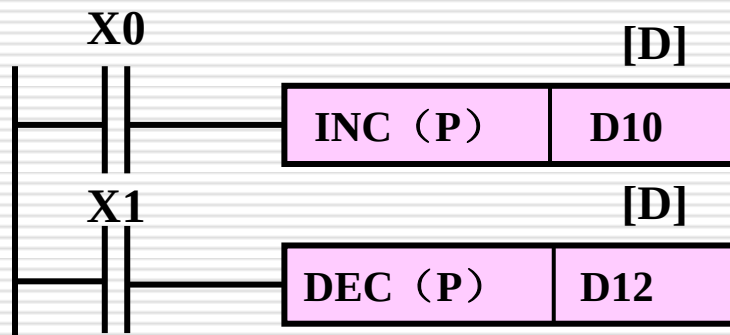
2. 二进制加1、减1指令（自加一、自减一）

加1指令：INC（编号FNC24）

减1指令：DEC（编号FNC25）

加1指令应用如图所示，X0每接通一次，目的操作数 (D) 中的软元件内容自动加1。在连续执行指令中，每个扫描周期都将执行加1运算。

减1指令应用如图所示，X0每接通一次，目的操作数 (D) 中的软元件内容自动减1。在连续执行指令中，每个扫描周期都将执行减1运算。



$(D10)+1 \Rightarrow (D10)$

$(D12)-1 \Rightarrow (D12)$

算术运算指令应用实例1：停车场车位统计

- 假设有一汽车停车场，为了表示停车场是否有空位，试用**PLC**加减指令及比较指令来实现如下功能：
- 设停车场有**50**个车位，在出入口各有一个红外探测器，当有车进入时，车位减**1**，反之，车位加**1**.
 - 若空余车位数多于**10**个，亮绿灯表示“车位充足”；
 - 若空余车位数介于**3~10**个，亮黄灯表示“车位紧张”；
 - 若空余车位数少于**3**个（不含**3**个），亮红灯表示“车位已满”；

算术运算指令应用实例2：彩灯控制

- 设有**16**盏灯，接于输出点**Y0~Y7**，**Y10~Y17**；
- 要求打下开关，灯光以正序每隔**1**秒轮流点亮，并循环。
- 断开启动开关，灯光熄灭。
- 提示：用乘法指令：

$1 \times 2 = 2 \rightarrow K4Y0$ Y1灯亮

$2 \times 2 = 4 \rightarrow K4Y0$ Y2灯亮

$4 \times 2 = 8 \rightarrow K4Y0$ Y3灯亮

⋮

灯组移位控制

 END