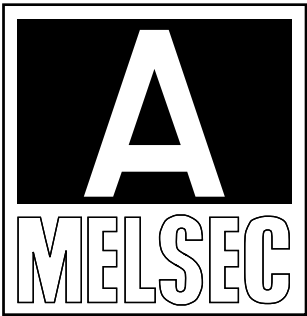
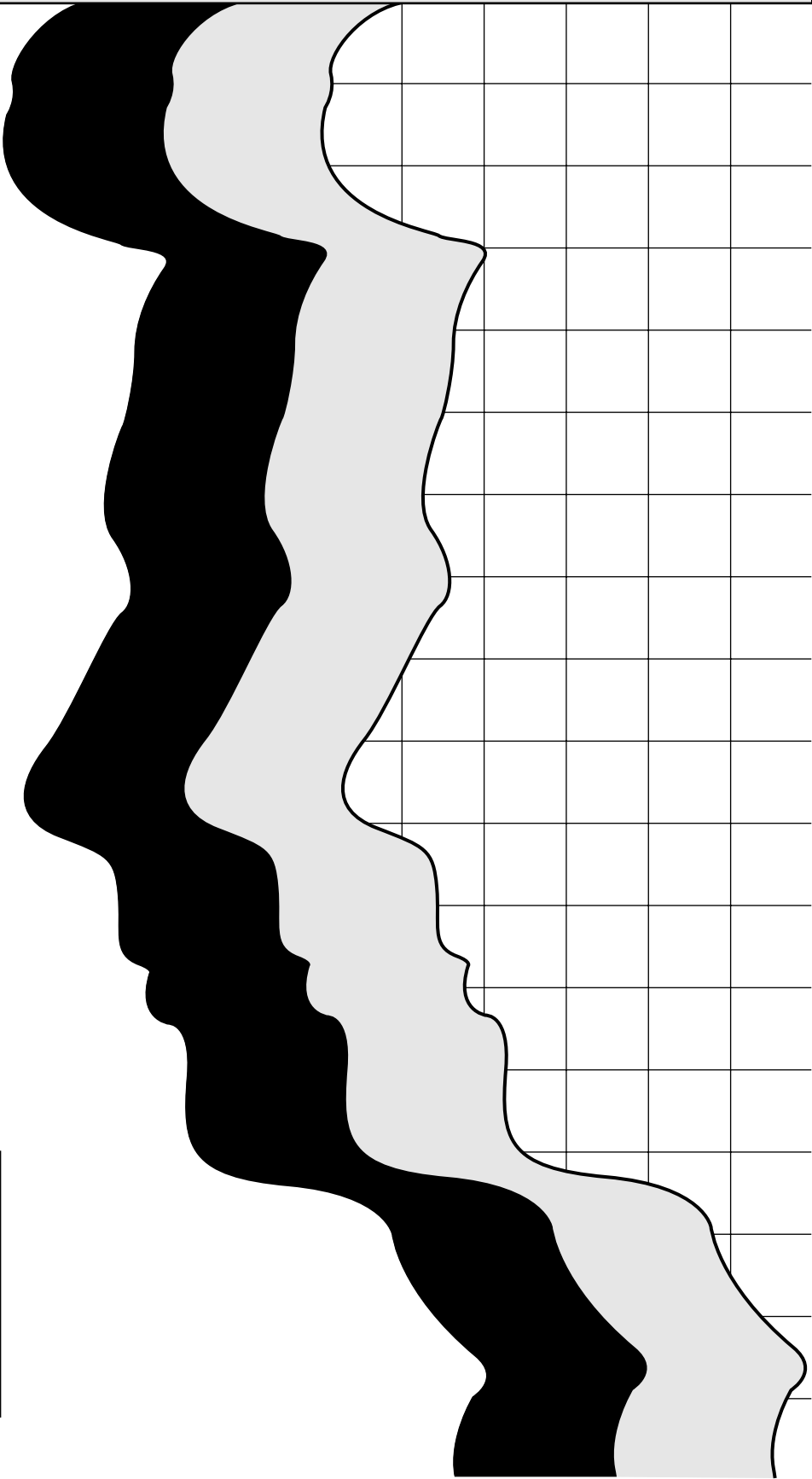


ACPU園殼返過(听喘鐙)

園殼歌深返過



辛園殼陣崙勾

勘 误 表

页码	错误	修正
3-10	第1个图下部的 “Index qualification”	变址
7-84	第2个图未译	见勘误第八页第二个图符
7-85	倒数第2行的 “需求的点数等于远程 I/O站中所用的特殊”	需求的点数等于远程 I/O站中所用的特殊功能模块数。
9-2	所有图表中 “stop”	停止
9-3	所有图表中 “stop”	停止
9-10	第一行的图表头	表9.2 AnA (-F) 的出错代码一览表 (续)
9-14	图表未译	见勘误第一页
9-15	图表未译	见勘误第二页
9-16	图表未译	见勘误第三页
9-17	图表未译	见勘误第四页
9-18	图表未译	见勘误第五页
9-19	图表未译	见勘误第六页
9-22	图表未译	见勘误第七页
9-23	图表未译	见勘误第八页

修订

※ 手册编号在封底左下侧给出

印刷日期	手册编号	修 订
1990. 10月	SH(NA) -080268C-A	第一版

序 言

感谢您选用三菱**MELSEC-A**系列通用型可编程序控制器。请仔细阅读本手册。这会令您使用时更得心应手。本手册的印刷本应直接送至最终用户。

目 录

1. 概述.....	1-1
2. 指令.....	2-1 ~ 2-24
2.1 分类.....	2- 1
2.2 指令清单.....	2- 2
2.2.1 指令清单的解释.....	2- 2
2.2.2 顺控指令.....	2- 5
2.2.3 基本指令.....	2- 8
2.2.4 应用指令.....	2-16
3. 指令结构.....	3-3 ~ 3-20
3.1 指令结构.....	3- 1
3.2 位处理.....	3- 3
3.2.1 1 位处理.....	3- 3
3.2.2 数字指定处理.....	3- 3
3.3 数值处理.....	3- 6
3.4 32位数据存储.....	3- 8
3.5 变址修饰.....	3-10
3.6 子集处理.....	3-12
3.7 操作出错.....	3-12
3.8 使用A□A的注意事项.....	3-14
3.8.1 指令步数的变化.....	3-14
3.8.2 可变功能的指令.....	3-16
3.8.3 扩展定时器和计数器的设定值.....	3-17
3.8.4 使用变址须知.....	3-17
3.8.5 存储32位数据的变址寄存器.....	3-20
4. 指令格式.....	4-1 ~ 4-3
5. 顺控指令.....	5-1 ~ 5-41
5.1 触点指令.....	5- 2
5.1.1 操作起始、串联、并联 (LD, LDI, AND, ANI, OR, ORI).....	5- 2
5.2 连接指令.....	5- 5
5.2.1 阶梯图电路块串联、块并联 (ANB, ORB).....	5- 5
5.2.2 操作结果进栈、读、出栈 (MPS, MRD, MPP).....	5- 9
5.3 输出指令.....	5-14
5.3.1 位软元件定时器计数器输出 (OUT).....	5-14
5.3.2 位软元件置位、复位 (SET, RST).....	5-19
5.3.3 边沿触发微分输出 (PLS, PLF).....	5-23
5.3.4 位软元件输出翻转 (CHK).....	5-25
5.4 移位指令.....	5-27
5.4.1 位软元件移位 (SFT, SFTP).....	5-27
5.5 主控指令.....	5-29

5.5.1 主控置位和主控复位 (MC, MCR)	5-29
5.6 终止指令	5-33
5.6.1 主程序结束 (FEND)	5-33
5.6.2 顺控程序结束 (END)	5-35
5.7 其它指令	5-37
5.7.1 顺控程序停止 (STOP)	5-37
5.7.2 无操作 (NOP, NOPLF)	5-39
6. 基本指令	6-1 ~ 6-88
6.1 比较指令	6- 2
6.1.1 16位数据比较 (=, <>, >, <=, <, >=)	6- 4
6.1.2 32位数据比较 (D=, D<>, D>, D<=, D<, D>=)	6- 6
6.2 算术运算指令	6- 8
6.2.1 16位二进制数加、减 (+, +P, -, -P)	6-10
6.2.2 32位二进制数加、减 (D+, D+P, -D, D-P)	6-13
6.2.3 16位二进制数乘、除 (*, *P, /, /P)	6-16
6.2.4 32位二进制数乘、除 (D*, D*P, /, D/P)	6-19
6.2.5 4位BCD数加、减 (B+, B+P, B-, B-P)	6-22
6.2.6 8位BCD数加、减 (DB+, DB+P, DB-, DB-P)	6-25
6.2.7 4位BCD数乘、除 (B*, B*P, B/, B/P)	6-28
6.2.8 8位BCD数乘、除 (DB*, DB*P, DB/, DB/P)	6-31
6.2.9 16位二进制数递增、递减 (INC, INCP, DEC, DECP)	6-34
6.2.10 32位二进制数递增、递减 (DINC, DINCP, DDEC, DDECP)	6-34
6.3 BCD $\leftrightarrow$ 二进制数转换指令	6-38
6.3.1 二进制数 $\rightarrow$ BCD 4位、8位数的转换 (BCD, BCDP, DBCD, DBCDP)	6-39
6.3.2 BCD 4位、8位 $\rightarrow$ 二进制数的转换 (BIN, BINP, DBIN, DBINP)	6-42
6.4 数据传送指令	6-46
6.4.1 16位、32位数据传送 (MOV, MOVP, DMOV, DMOVP)	6-47
6.4.2 16位、32位数据取反传送 (CML, CMLP, DCML, DCMLP)	6-49
6.4.3 16位数据块传送 (BMOV, BMOVP, FMOV, FMOVP)	6-52
6.4.4 16位、32位数据交换 (XCH, XCHP, DXCH, DXCHP)	6-56
6.5 程序分支指令	6-58
6.5.1 有条件跳转, 无条件跳转 (CJ, SCJ, JMP)	6-58
6.5.2 子程序调用, 返回 (CALL, CALLP, RET)	6-62
6.5.3 中断允许, 禁止, 返回 (EI, DI, RET)	6-64
6.5.4 微机程序调用 (SUB, SUBP)	6-67
6.6 程序切换指令	6-69
6.6.1 主 $\leftrightarrow$ 副程序切换 (CHG)	6-69
6.7 通讯刷新指令	6-82
6.7.1 通讯刷新 (COM)	6-82
6.7.2 通讯刷新允许, 禁止 (EI, DI)	6-84
6.7.3 部分刷新 (SEG)	6-86
7. 应用指令	7-1 ~ 7-127
7.1 逻辑运算指令	7- 2
7.1.1 16位、32位数据逻辑乘 (WAND, WANDP, DAND, DANDP)	7- 3
7.1.2 16位、32位数据逻辑加 (WOR, WORP, WDOR, DORP)	7- 7

7.1.3	16位、32位数据逻辑异或 (WXOR, WXORP, DXOR, DXORP)	7-11
7.1.4	16位、32位数据逻辑异或非 (WOR, WORP, DWOR, DWORP)	7-15
7.1.5	16位二进制数补码 (NEG, NEGP)	7-19
7.2	旋转指令	7-21
7.2.1	16位数右旋 (ROR, RORP, RCR, RCRP)	7-22
7.2.2	16位数左旋 (ROL, ROLP, RCL, RCLP)	7-24
7.2.3	32位数右旋 (DROR, DRORP, DRCR, DRCRP)	7-26
7.2.4	32位数左旋 (DROL, DROLP, DRCL, DRCLP)	7-28
7.3	移位指令	7-30
7.3.1	16位数据中n位右移, 左移 (SFR, SFRP, SFL, SFLP)	7-31
7.3.2	n位数据中1位右移, 左移 (BSFR, BSFRP, BSFL, BSFLP)	7-33
7.3.3	n字数据中1字右移, 左移 (DSFR, DSFRP, DSFL, DSFLP)	7-35
7.4	数据处理指令	7-37
7.4.1	16位数据搜索 (SER, SERP)	7-38
7.4.2	16位、32位数据校验 (SUM, SUMP, DSUM, DSUMP)	7-40
7.4.3	8 $\leftrightarrow$ 256位解码, 编码 (DECO, DECOP, ENCO, ENCOP)	7-42
7.4.4	7段译码 (SEG)	7-44
7.4.5	字软元件中位置位、复位 (BSET, BSETP, BRST, BRSTP)	7-46
7.4.6	16位数据分离、组合 (DIS, DISP, UNI, UNIP)	7-48
7.4.7	ASCII码转换 (ASC)	7-51
7.5	先入先出指令	7-53
7.5.1	先入先出表写、读 (FIFW, FIFWP, FIFR, FIFRP)	7-54
7.6	缓冲存储区存取指令	7-58
7.6.1	特殊功能模块单字、双字读 (FROM, FROP, DFRO, DFROP)	7-59
7.6.2	特殊功能模块单字、双字写 (TO, TOP, DTO, DTOP)	7-61
7.6.3	远程模块单字、双字读 (FROM, PRC, FROP, PRC, DFRO, PRC, DFROP, PRC)	7-63
7.6.3	远程模块单字、双字写 (TO, PRC, TOP, PRC, DTO, PRC, DTOP, PRC)	7-67
7.7	FOR ~ NEXT 指令	7-71
7.7.1	FOR ~ NEXT (FOR, NEXT)	7-71
7.8	就地、远程I/O站读写指令	7-73
7.8.1	就地站数据读、写 (LRDP, LWTP)	7-74
7.8.2	远程I/O站数据读、写 (RFRP, RTOP)	7-80
7.9	显示指令	7-86
7.9.1	ASCII字符打印指令 (PR, PRC)	7-88
7.9.2	ASCII码注释显示指令 (LED, LEDC)	7-94
7.9.3	字符显示指令 (LEDA, LEDB)	7-97
7.9.4	信号报警器复位指令 (LEDR)	7-99
7.10	其它指令	7-101
7.10.1	WDT复位 (WDT, WDTTP)	7-102
7.10.2	特定格式故障检测 (CHK)	7-104
7.10.3	状态锁存置位、复位 (SLT, SLTR)	7-111
7.10.4	采样跟踪置位、复位 (STRA, STRAR)	7-113
7.10.5	进位标志置位、复位 (STC, CLC)	7-115
7.10.6	脉冲生成指令 (DUTY)	7-117
7.11	伺服程序指令	7-119
7.11.1	伺服程序启动 (DSFRP)	7-120

7.11.2 当前位置数据和速度改变指令 (DSFLP).....	7-124
8. 微机模式.....	8-1 ~ 8-14
8.1 微机模式特性.....	8- 1
8.2 使用应用程序.....	8- 2
8.3 使用用户编写的微机程序.....	8- 4
8.3.1 存储器映象.....	8- 6
8.3.2 数据存储区地址配置.....	8- 6
8.3.3 数据存储区配置.....	8- 7
9. 出错代码清单.....	9-1 ~ 9-23
9.1 读取出错代码.....	9- 1
9.2 CPU (除AnACPU (-F) 和AnU外) 的出错代码清单.....	9- 1
9.3 AnACPU (-F) 的出错代码清单.....	9- 6
9.4 AnUCPU的出错代码清单.....	9-14
10. 附录.....	APP-1 ~ APP-56
附录1 特殊继电器和特殊寄存器.....	APP- 1
1.1 特殊继电器 M.....	APP- 1
1.2 通讯用特殊继电器.....	APP- 8
1.3 特殊寄存器.....	APP-11
1.4 通讯用特殊寄存器.....	APP-25
附录2 操作处理时间.....	APP-30
附录3 ASCII码表.....	APP-49
附录4 程序纸的格式.....	APP-50

1. 引言

本手册阐述如何使用MELSEC-A系列可编程控制器的顺控指令和微机程序。
MELSEC-A系列可编程控制器具有一组用来指定其功能和软元件使用范围的参数，其功能与软元件使用范围由参数值来决定。
CPU组件的参数一般按缺省约定值设置。如果该缺省约定值可以满足用户要求，就无必要再去设置参数。MELSEC-A系列PLC的用户程序可分类如下：

ACPU编程手册(基础篇)给出可用于CPU组件的程序。

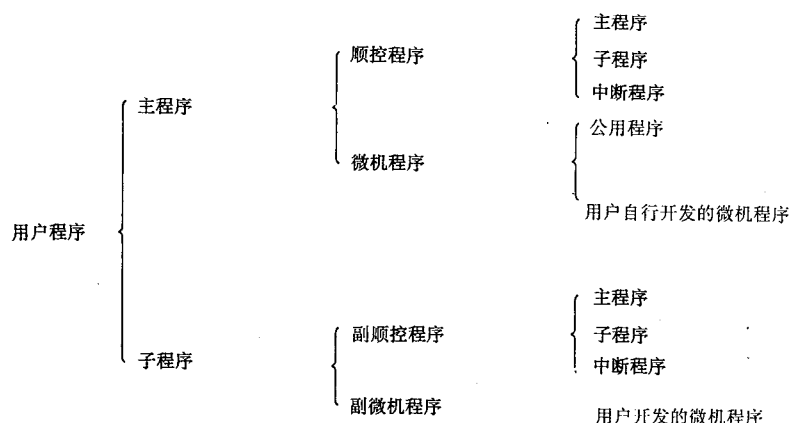


表1.1给出适用的CPU组件和用在本手册中的缩写。

表1.1 适用CPU与在本手册中用的缩写

本手册中用的缩写		适用CPU组件
An	A1	A1CPU(P21/R21)
	A2(-S1)	A2CPU(P21/R21), A2CPU(P21/R21)-S1
	A3	A3CPU(P21/R21)
AnN	A1N	A1NCPU(P21/R21)
	A2N(-S1)	A2NCPU(P21/R21), A2NCPU(P21/R21)-S1
	A3N	A3NCPU(P21/R21)
AnN-F	A2N(-S1)-F	A2NCPU(P21/R21)-F, A2NCPU(P21/R21)-S1-F
	A3N-F	A3NCPU(P21/R21)-F
A3H		A3HCPU(P21/R21)
A3M		A3MCPU(P21/R21)
A3V		A3VCPU(P21/R21)
AnA	A2A(-S1)	A2ACPU(P21/R21), A2ACPU(P21/R21)-S1
	A3A	A3ACPU(P21/R21)
AnA-F	A2A(-S1)-F	A2ACPU(P21/R21)-F, A2ACPU(P21/R21)-S1-F
	A3A-F	A3ACPU(P21/R21)-F
A0J2H		A0J2HCPU(P21/R21)
A1S		A1SCPU, A1SCPU-S1
A2C		A2CCPU(P21/R21), A2CCPUDC24, A2CCPUC24
A3N'board		A7BDE-A3N-PT32-S3
A73		A73CPU(P21/R21)
A52G		A52GCPU(T21B)
AnU	A2U(-S1)	A2UCPU, A2UCPU-S1
	A3U	A3UCPU
	A4U	A4UCPU

表1.2 外部设备和在本手册中用的缩写

本手册中用的缩写	外部设备
GPP	A6GPP IBM PC/AT(GPP 功能)
	A6HGP
	A6PHP A7PHPE(GPP 功能)

要点

本手册不可用于A0J2CPU(P23/R23)。
关于A0J2CPU(R23/P23)用的指令, 参见A0J2CPU编程手册。

对A系列PLC编程时还可查阅以下手册:

主题	内容	参考手册
CPU 性能	• CPU模块的用户程序容量和软元件数目 • 电源模块和基板性能规格	各CPU模块的用户手册
CPU 功能	• PLC的系统组态 • CPU模块的性能和功能 • CPU模块的处理 • 软元件和参数清单	
编写程序	• 编程步骤 • 软元件和参数的描述 • 程序类型 • 存储区结构	ACPU编程手册(公用指令篇) IB(NA)-66249
使用 A2A(S1) 和 A3ACPU	• 专用指令(扩展应用指令)的描述	AnACPU/AnUCPU 编程手册 (专用指令篇) IB(NA)-66251
	• AD57控制指令的描述	AnACPU/AnUCPU 编程手册 (AD57指令篇) IB(NA)-66257
	• PID控制指令的描述	AnACPU/AnUCPU 编程手册 (PID指令篇) IB(NA)-66258
使用 A73CPU	• 定位控制 • 编写伺服程序 • 辅助和应用功能的描述	A73CPU 参考手册 IB(NA)-66233

2. 指令

2.1 分类

MELSEC-A系列的指令大致分成3类: 顺控指令, 基本指令, 应用指令, 见表2-1。

指令的分类		说明	页次
顺控指令	触点指令	运算起始, 串联, 并联	5-2 至 5-4
	连接指令	阶梯图块连接, 运算结果存取	5-5 至 5-13
	输出指令	位软元件输出, 脉冲输出, 输出取反	5-14 至 5-26
	移位指令	位软元件移位	5-27 至 5-28
	主控指令	主控	5-29 至 5-32
	结束指令	程序结束	5-33 至 5-36
	其它指令	程序停止, 空操作	5-37 至 5-42
基本指令	比较指令	比较, 如“=”, “>”和“<”	6-2 至 6-7
	算术运算指令	二进制数和BCD数的加, 减, 乘, 除	6-8 至 6-37
	BCD $\leftrightarrow$ 二进制数	二进制和BCD数相互转换	6-38 至 6-45
	数据传送指令	传送特定的数据	6-46 至 6-57
	程序分支指令	程序跳转, 子程序 / 中断程序调用	6-58 至 6-68
	程序切换指令	主副程序切换	6-69 至 6-81
	通讯刷新指令	通讯刷新, 部分刷新	6-82 至 6-88
应用指令	逻辑运算指令	逻辑运算, 如逻辑加, 逻辑乘	7-2 至 7-20
	旋转指令	指定数据旋转	7-21 至 7-29
	移位指令	指定数据移位	7-30 至 7-36
	数据处理指令	数据处理, 如16位数据搜索译码, 解码	7-37 至 7-52
	先入先出指令	先入先出表写 读able	7-53 至 7-57
	缓冲存储区存取指令	特殊功能模块读写, 远程模块读写	7-58 至 7-70
	FOR~NEXT指令	FOR / NEXT之间程序反复执行	7-71 至 7-72
	就地站、远程I / O站 读写指令	就地站、远程I / O站数据读写	7-73 至 7-85
	显示指令	ASCII字符打印, 字符LED显示	7-86 至 7-100
	其它指令	以上分类尚未包括的指令, 如WDT复位 进位标志置位、复位	7-101 至 7-117
	伺服程序指令	伺服程序执行, 设定值改变等	7-118 至 7-126

2.2 指令一览表

2.2.1 指令一览表的解释

2.2.2 ~ 2.2.4 节的指令一览表有如下格式

分类	单位	指令符	表达符号	处理内容	执行 条件	步 序	数 址	子 操	适用 CPU	页次
16位二进制加/减	16 位	+	- + S D	(D) + (S) → (D)		5	●	●	○	6-10
		+P	- +P S D	(D) + (S) → (D)		5	●	●	○	6-10
		+	- + S1 S2 D	(S1) + (S2) → (D)		7	●	●	○	6-10
		+P	- +P S1 S2 D	(S1) + (S2) → (D)		7	●	●	○	6-10
		-	- - S D	(D) - (S) → (D)		5	●	●	○	6-10
		-P	- -P S D	(D) - (S) → (D)		5	●	●	○	6-10
		-	- - S1 S2 D	(S1) - (S2) → (D)		7	●	●	○	6-10
		-P	- -P S1 S2 D	(S1) - (S2) → (D)		7	●	●	○	6-10
①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪

表2.2 指令清单的解释

解 释

①..... 按应用对指令分类。

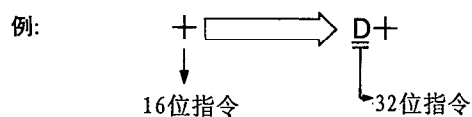
②..... 表示指令执行时处理的单位。

处理的批量	元件	点数
16 位	X, Y, M, L, F, B	4点1组, 最多16点
	T, C, D, W, R, A, Z, V	1 点
32 位	X, Y, M, L, F, B	4点1组, 最多32点
	T, C, D, W, R, A0, Z	2 点

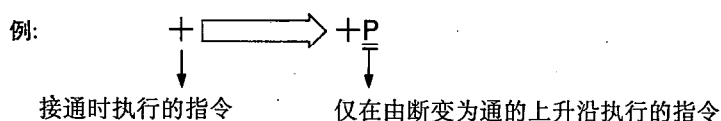
③.....表示用来编程的指令助记符。

指令符号以16位指令为基础。32位指令和上跳沿执行的指令如下所示:

32位指令..... 在指令的第一个字符前加 "D"。



从OFF到ON的上跳沿执行的指令..... 在指令的末尾加 "P"。



④.....表示电路中的符号图。

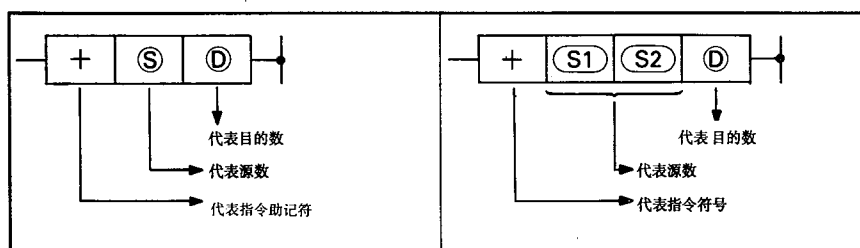


图2.2 阶梯图的符号表达

目的操作数: 表示操作后存放数据的地址

源操作数: 存放操作前的数据

⑤.....表示每条指令的处理

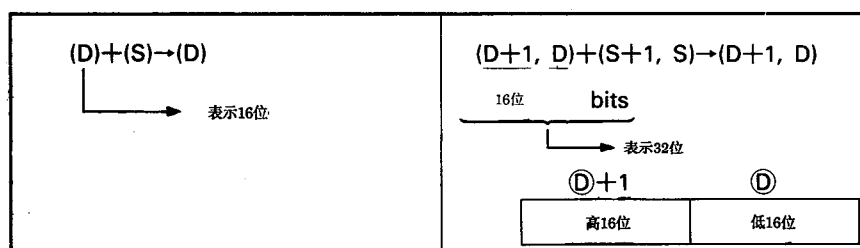


图2.3 每条指令的处理

⑥表示每条指令的执行条件,详述如下:

符号	执行条件
无	不管ON / OFF条件指令都执行。当前面条件是OFF时,执行OFF处理。
	指令只在ON时执行。当前面的条件为OFF时,既不执行也不处理。
	指令只在为ON时执行1次。当前面条件发生正跳变时,如从ON→OFF,指令才执行。此后即使条件为ON,指令也不执行,不处理。
	指令在为OFF时执行1次。当前面条件发生负跳变时,如从OFF→ON,指令才执行。此后即使条件为OFF,指令也不执行,不处理。

⑦表示每个指令的步数。由于步数因条件而变,所以有2种表达。具体请查每个指令。

要点

对于AnA, AnA-F和AnU, 若使用扩展软元件,或在指令中要执行位软元件的变址,步数要增加。详见3. 8. 1节

⑧..... ○表示指令可用变址(Z, V)。

▲表示指令只有在A□ACPU可用变址

⑨..... ○表示指令可作子集处理

▲表示此子集处理只可在A3H, A3M和A□ACPU中执行。

⑩..... 表示适用的CPU

●表示适用于所有的CPU

△表示适用于部分的CPU

—表示适用的特定的CPU

⑪..... 该指令的详细叙述所在的页次

2.2.2 顺控指令

(1) 触点指令

分类	单位	指令符号	表达符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
触点		LD		逻辑操作开始 (常开触点操作开始)		1	*2 ▲	○		5-2
		LDI		逻辑非操作开始 (常闭触点操作开始)		1	*2 ▲	○		5-2
		AND		逻辑乘 (常开触点串联)		1	*2 ▲	○		5-2
		ANI		逻辑乘非 (常闭触点串联)		1	*2 ▲	○		5-2
		OR		逻辑加 (常开触点并联)		1	*2 ▲	○		5-2
		ORI		逻辑加非 (常闭触点并联)		1	*2 ▲	○		5-2

表2.3 触点指令

(2) 连接指令

分类	单位	指令符号	表达符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
Connection		ANB		AND逻辑块与(块的串联)		1		○		5-5
		ORB		OR逻辑块或(块的并联)		1		○		5-5
		MPS		存储操作结果		1		○		5-9
		MRD		从MPS读取操作结果		1		○		5-9
		MPP		从MPS读取操作结果,并清除结果		1		○		5-9

表2.4 连接指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	表达符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
输出	—	OUT		软元件输出		1 3	*2 ▲	○		5-14
		SET		软元件置位	*	1 3	*2 ▲	○		5-19
		RST		软元件复位	*	1 3	*2 ▲	○		5-19
		PLS		在输入信号的上升沿 产生1个扫描周期的脉冲		3	*2 ▲	○		5-23
		PLF		在输入信号的下降沿 产生1个扫描周期的脉冲		3	*2 ▲	○		5-23
		CHK		软元件输出翻转 只在I/O刷新模式有效		5		△	不适用于 An, A3V, A2C, A3H, A3M, A52G, AnA, AnA-F 和 AnU.	5-25

表2.5 输出指令

备 注

(3)输出指令中标有*的执行条件:



当用于F(信号报警器)



当用于其它元件

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
移位	—	SFT		元件移1位		3	*2 ▲	○		5-27
		SFTP				3	*2 ▲	○		5-27

表2.6 移位指令

(5) 主控指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
主控	—	MC		主控开始		5	*2 ▲	○		5-29
		MCR		主控复位		3		○		5-29

表2.7 主控指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行。

(6) 结束指令

分类	位 单	指令符号	符号	处理内容	执行条件	数 步	变 址	子 集	适用 CPU	页次
程序结束	—	FEND		用于结束主程序,以终止处理		1		○		5-33
	—	END	—	总用于所有顺控程序的末尾, 返回第0步		1		○		5-35

表2.8 结束指令

(7) 其它指令

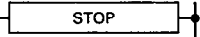
分类	位 单	指令符号	符号	处理内容	执行条件	数 步	变 址	子 集	适用 CPU	页次
停止	—	STOP		输入有效时输出复位,停止顺控程序. 钥匙开关转到RUN,则程序重新开始		1		○		5-37
空操作	—	NOP	—	无操作,用于程序删除和空行		1		○		5-39

表 2.9其它指令

- *1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。
- *2: 变址栏内的▲表示指令只可在A□ACPU使用变址
- *3: 子集栏内的▲表示此子集处理只可在A3H, A3M和A□ACPU中执行

2.2.4 基本指令

(1) 比较指令

分类	单位	指令符号	符号	处理内容	执行条件	源 地址	目标 地址	子集	适用 CPU	页次
16位数据 比较	16 位	LD=		当(S1)=(S2),接通 当(S1)≠(S2),断开		5 7	●	●	○	6-4
		AND=				5 7	●	●	○	6-4
		OR=				5 7	●	●	○	6-4
		LD<>		当(S1)≠(S2),接通 当(S1)=(S2),断开		5 7	●	●	○	6-4
		AND<>				5 7	●	●	○	6-4
		OR<>				5 7	●	●	○	6-4
		LD>		当(S1)>(S2),接通 当(S1)≤(S2),断开		5 7	●	●	○	6-4
		AND>				5 7	●	●	○	6-4
		OR>				5 7	●	●	○	6-4
		LD<=		当(S1)≤(S2),接通 当(S1)>(S2),断开		5 7	●	●	○	6-4
		AND<=				5 7	●	●	○	6-4
		OR<=				5 7	●	●	○	6-4
		LD<		当(S1)<(S2),接通 当(S1)≥(S2),断开		5 7	●	●	○	6-4
		AND<				5 7	●	●	○	6-4
		OR<				5 7	●	●	○	6-4
		LD>=		当(S1)>(S2),接通 当(S1)≤(S2),断开		5 7	●	●	○	6-4
		AND>=				5 7	●	●	○	6-4
		OR>=				5 7	●	●	○	6-4

表2.10 比较指令(待续)

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU使用变址

*3: 子集栏内的▲表示此子集处理只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	符号	处理内容	Execution Condition	步数	取指	操作	适用 CPU	页次
32位数据比较	32 位	LDD=		当(S1+1,S1)=(S2+1,S2),接通 当(S1+1,S1)≠(S2+1,S2),断开		11	●	○		6-6
		ANDD=				11	●	○		6-6
		ORD=				11	●	○		6-6
		LDD<>		当(S1+1,S1)≠(S2+1,S2),接通 当(S1+1,S1)=(S2+1,S2),断开		11	●	○		6-6
		ANDD<>				11	●	○		6-6
		ORD<>				11	●	○		6-6
		LDD>		当(S1+1,S1)>(S2+1,S2),接通 当(S1+1,S1)<(S2+1,S2),断开		11	●	○		6-6
		ANDD>				11	●	○		6-6
		ORD>				11	●	○		6-6
		LDD<=		当(S1+1,S1)<(S2+1,S2),接通 当(S1+1,S1)>(S2+1,S2),断开		11	●	○		6-6
		ANDD<=				11	●	○		6-6
		ORD<=				11	●	○		6-6
		LDD<		当(S1+1,S1)<(S2+1,S2),接通 当(S1+1,S1)>(S2+1,S2),断开		11	●	○		6-6
		ANDD<				11	●	○		6-6
		ORD<				11	●	○		6-6
		LDD>=		当(S1+1,S1)>(S2+1,S2),接通 当(S1+1,S1)<(S2+1,S2),断开		11	●	○		6-6
		ANDD>=				11	●	○		6-6
		ORD>=				11	●	○		6-6

表2.10 比较指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(2) 算术运算指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
二进制 16位 加/减	16 位			$(D) + (S) \rightarrow (D)$		5	●	●	○	6-10
						5	●	●	○	6-10
				$(S1) + (S2) \rightarrow (D)$		7	●	●	○	6-10
						7	●	●	○	6-10
				$(D) - (S) \rightarrow (D)$		5	●	●	○	6-10
						5	●	●	○	6-10
				$(S1) - (S2) \rightarrow (D)$		7	●	●	○	6-10
						7	●	●	○	6-10
二进制 32位 加/减	32 位			$(D + 1, D) + (S + 1, S) \rightarrow (D + 1, D)$		9	●	●	○	6-13
						9	●	●	○	6-13
				$(S1 + 1, S1) + (S2 + 1, S2) \rightarrow (D + 1, D)$		11	●	●	○	6-13
						11	●	●	○	6-13
				$(D + 1, D) - (S + 1, S) \rightarrow (D + 1, D)$		9	●	●	○	6-13
						9	●	●	○	6-13
				$(S1 + 1, S1) - (S2 + 1, S2) \rightarrow (D + 1, D)$		11	●	●	○	6-13
						11	●	●	○	6-13
二进制 16位 乘/除	16 位			$(S1) \times (S2) \rightarrow (D + 1, D)$		7	●	●	○	6-16
						7	●	●	○	6-16
				$(S1) \div (S2) \rightarrow$ 商存于(D), 余数存于(D+1)		7	●	●	○	6-16
						7	●	●	○	6-16

表2.11 算术运算指令(待续)

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行。

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
二进制32位乘/除	32位	D*	$\text{D*} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1} + 1, \text{S1}) \times (\text{S2} + 1, \text{S2}) \rightarrow (\text{D} + 3, \text{D} + 2, \text{D} + 1, \text{D})$		11	●	●	○	6-19
		D*P	$\text{D*P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			11	●	●	○	6-19
		D/	$\text{D/} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1} + 1, \text{S1}) \div (\text{S2} + 1, \text{S2}) \rightarrow$ 商存于 $(\text{D} + 1, \text{D})$, 余数存 $(\text{D} + 3, \text{D} + 2)$		11	●	●	○	6-19
		D/P	$\text{D/P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			11	●	●	○	6-19
4位BCD数加/减	4位BCD数	B+	$\text{B+} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$	$(\text{D}) + (\text{S}) \rightarrow (\text{D})$		7	●	▲ ^{*3}	○	6-22
		B+P	$\text{B+P} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$			7	●	▲ ^{*3}	○	6-22
		B+	$\text{B+} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1}) + (\text{S2}) \rightarrow (\text{D})$		9	●		○	6-22
		B+P	$\text{B+P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			9	●		○	6-22
		B-	$\text{B-} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$	$(\text{D}) - (\text{S}) \rightarrow (\text{D})$		7	●	▲ ^{*3}	○	6-22
		B-P	$\text{B-P} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$			7	●	▲ ^{*3}	○	6-22
		B-	$\text{B-} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1}) - (\text{S2}) \rightarrow (\text{D})$		9	●		○	6-22
		B-P	$\text{B-P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			9	●		○	6-22
8位BCD数加/减	8位BCD数	DB+	$\text{DB+} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$	$(\text{D} + 1, \text{D}) + (\text{S} + 1, \text{S}) \rightarrow (\text{D} + 1, \text{D})$		9	●		○	6-25
		DB+P	$\text{DB+P} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$			9	●		○	6-25
		DB+	$\text{DB+} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1} + 1, \text{S1}) + (\text{S2} + 1, \text{S2}) \rightarrow (\text{D} + 1, \text{D})$		11	●		○	6-25
		DB+P	$\text{DB+P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			11	●		○	6-25
8位BCD数乘/除	8位BCD数	DB-	$\text{DB-} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$	$(\text{D} + 1, \text{D}) - (\text{S} + 1, \text{S}) \rightarrow (\text{D} + 1, \text{D})$		9	●		○	6-25
		DB-P	$\text{DB-P} \begin{array}{ c c } \hline \text{S} & \text{D} \\ \hline \end{array}$			9	●		○	6-25
		DB-	$\text{DB-} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$	$(\text{S1} + 1, \text{S1}) - (\text{S2} + 1, \text{S2}) \rightarrow (\text{D} + 1, \text{D})$		11	●		○	6-25
		DB-P	$\text{DB-P} \begin{array}{ c c c } \hline \text{S1} & \text{S2} & \text{D} \\ \hline \end{array}$			11	●		○	6-25

表2.11 算术运算指令(待续)

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	乘子	适用 CPU	页次
4位BCD数 乘/除	4位 BCD	B*		$(S1) \times (S2) \rightarrow (D+1, D)$		9	●	▲ ^{*3}		6-28
		B*P								6-28
		B/		$(S1) \div (S2) \rightarrow$ 商存于(D), 余数存于(D+1)		9	●	▲ ^{*3}		6-28
		B/P								6-28
8位BCD数 乘/除	8位 BCD	DB*		$(S1+1, S1) \times (S2+1, S2) \rightarrow$ $(D+3, D+2, D+1, D)$		11	●			6-31
		DB*P								6-31
		DB/		$(S1+1, S1) \div (S2+1, S2) \rightarrow$ 商存(D+1, D), 余数存(D+3, D+2)		11	●			6-31
		DB/P								6-31
二进制数 加1	16 位	INC		$(D) + 1 \rightarrow (D)$		3	●	●		6-34
		INCP								6-34
	32 位	DINC		$(D+1, D) + 1 \rightarrow (D+1, D)$		3	●	●		6-36
		DINCP								6-36
二进制数 减1	16 位	DEC		$(D) - 1 \rightarrow (D)$		3	●	●		6-34
		DECP								6-34
	32 位	DDEC		$(D+1, D) - 1 \rightarrow (D+1, D)$		3	●	●		6-36
		DDECP								6-36

表2.11 算术运算指令

- *1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。
- *2: 变址栏内的▲表示指令只有在A□ACPU可用变址
- *3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
BCD转换	16 位	BCD		BCD转换 $(S) \rightarrow (D)$ 二进制(0~9999)		5	●	●	○	6-39
		BCDP		BCD转换 $(S) \rightarrow (D)$ 二进制(0~9999)		5	●	●	○	6-39
	32 位	DBCD		BCD转换 $(S+1, S) \rightarrow (D+1, D)$ 二进制(0~99999999)		9	●	▲	○	6-39
		DBCDP		BCD转换 $(S+1, S) \rightarrow (D+1, D)$ 二进制(0~99999999)		9	●	▲	○	6-39
二进制转换	4/BCD数	BIN		二进制转换 $(S) \rightarrow (D)$ BCD (0~9999)		5	●	●	○	6-42
		BINP		二进制转换 $(S) \rightarrow (D)$ BCD (0~9999)		5	●	●	○	6-42
	8/BCD数	DBIN		二进制转换 $(S+1, S) \rightarrow (D+1, D)$ BCD (0~99999999)		9	●	○	○	6-42
		DBINP		二进制转换 $(S+1, S) \rightarrow (D+1, D)$ BCD (0~99999999)		9	●	○	○	6-42

表2.12 BCD ↔ 二进制转换

(4) 数据传送指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
传送	16 位	MOV		$(S) \rightarrow (D)$		5	●	●	○	6-47
		MOVP		$(S) \rightarrow (D)$		5	●	●	○	6-47
	32 位	DMOV		$(S+1, S) \rightarrow (D+1, D)$		7	●	●	○	6-47
		DMOVP		$(S+1, S) \rightarrow (D+1, D)$		7	●	●	○	6-47
取反传送	16 位	CML		$\overline{(S)} \rightarrow (D)$		5	●	●	○	6-49
		CMLP		$\overline{(S)} \rightarrow (D)$		5	●	●	○	6-49
	32 位	DCML		$\overline{(S+1, S)} \rightarrow (D+1, D)$		7	●	●	○	6-49
		DCMLP		$\overline{(S+1, S)} \rightarrow (D+1, D)$		7	●	●	○	6-49

表2.13 数据传送指令(待续)

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只有在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	乘子	适用 CPU	页次
块传送	16 位	BMOV				9	●	▲ ^{*3}	○	6-52
		BMOVP				9	●	▲ ^{*3}	○	6-52
		FMOV				9	●	▲ ^{*3}	○	6-52
		FMOVP				9	●	▲ ^{*3}	○	6-52
交换	16 位	XCH		$(D1) \leftrightarrow (D2)$		5	●	●	○	6-56
		XCHP				5	●	●	○	6-56
	32 位	DXCH		$(D1 + 1, D1) \leftrightarrow (D2 + 1, D2)$		7	●	●	○	6-56
		DXCHP				7	●	●	○	6-56

表2.13 数据传送指令

(5) 程序分支指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	乘子	适用 CPU	页次
跳转	—	CJ		输入条件满足后跳转到 P** 处		3	●	▲ ^{*3}	○	6-58
		SCJ		输入条件满足后紧接着的扫描周期, 跳到 P** 处		3	●	▲ ^{*3}	○	6-58
		JMP		无条件跳到 P** 处		3	●	▲ ^{*3}	○	6-58
调用子程序	—	CALL		输入条件满足后执行 P** 处的子程序		3	●	▲ ^{*3}	○	6-62
		CALLP				3	●	▲ ^{*3}	○	6-62
		RET		从子程序返回顺控程序		1			○	6-62
调用中断程序	—	EI		允许中断, A□N 中 M9053 OFF 时有效		1		△	不适用于 A3V, A2C 和 A52G.	6-64
		DI		禁止中断, A□N 中 M9053 OFF 时有效		1		△	不适用于 A3V, A2C 和 A52G.	6-64
		IRET		从中断程序返回顺控程序		1		△	不适用于 A3V, A2C 和 A52G.	6-64
调用微程序	—	SUB		执行 n 指定的微机程序		3	●	△	不适用于 AnA 和 AnU.	6-67
		SUBP				3	●	△	不适用于 AnA 和 AnU.	6-67

表2.14 程序分支指令

*1: 使用扩展软元件或在 A□ACPU 的位软元件上使用变址而引起的步数变化请查阅 3.8.1 节。

*2: 变址栏内的 ▲ 表示指令只有在 A□ACPU 可用变址

*3: 子集栏内的 ▲ 表示此子集指令只可在 A3H, A3M 和 A□ACPU 中执行

(6) 程序切换指令

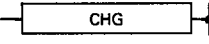
分类	单位	指令符号	Symbol	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
切换	—	CHG		在主副程序间切换	 A3H, A3M, A3A  其它CPU	1		△	不适用于 A1S, A1, A2(S1), A1N, A2N(S1), A2N(S1)-F, A2A(S1), A2A(S1)-F, A2C, A0J2H 和 A52G.	6-69

表2.15 程序切换指令

(7) 刷新指令

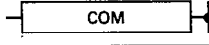
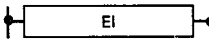
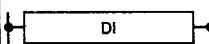
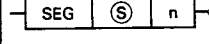
分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
通讯刷新	—	COM		执行通讯刷新,常规数据处理		3		△	不适用 A3V	6-82
通讯刷新 允许,禁止	—	EI		允许通讯刷新,M9053 ON时有效		1		△	不适用于 A3H,A3M,A3V, AnA(-F)和AnU	6-84
		DI		禁止通讯刷新,M9053 ON时有效		1		△	不适用于 A3H,A3M,A3V, AnA(-F)和AnU	6-84
部分刷新	—	SEG		对应软元件的刷新,仅执行1个扫描周期,M9052 ON时有效		7	*2 ▲	△	不适用于 An和A3N.	6-86

表2.16 刷新指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集处理只可在A3H, A3M和A□ACPU中执行

2.2.4 应用指令

(1) 逻辑运算指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
逻辑乘	16 位	WAND		(D) AND (S) → (D)		5	●	●	○	7-3
		WANDP				5	●	●	○	7-3
		WAND		(S1) AND (S2) → (D)		7	●		○	7-3
		WANDP				7	●		○	7-3
	32 位	DAND		(D + 1, D) AND (S + 1, S) → (D + 1, D)		9	●		○	7-3
		DANDP				9	●		○	7-3
逻辑加	16 位	WOR		(D) OR (S) → (D) *		5	●	●	○	7-7
		WORP				5	●	●	○	7-7
		WOR		(S1) OR (S2) → (D)		7	●		○	7-7
		WORP				7	●		○	7-7
	32 位	DOR		(D + 1, D) OR (S + 1, S) → (D + 1, D)		9	●		○	7-7
		DORP				9	●		○	7-7
逻辑异或	16 位	WXOR		(D) XOR (S) → (D)		5	●	●	○	7-11
		WXORP				5	●	●	○	7-11
		WXOR		(S1) XOR (S2) → (D)		7	●		○	7-11
		WXORP				7	●		○	7-11
	32 位	DXOR		(D + 1, D) XOR (S + 1, S) → (D + 1, D)		9	●		○	7-11
		DXORP				9	●		○	7-11

表2.17 逻辑运算指令 (待续)

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
逻辑异或非	16 位	WXNR		$\overline{(D)} \text{ XOR } (S) \rightarrow (D)$		5	●	●	○	7-15
		WXNRP				5	●	●	○	7-15
		WXNR		$\overline{(S1)} \text{ XOR } (S2) \rightarrow (D)$		7	●	○		7-15
		WXNRP				7	●	○		7-15
	32 位	DXNR		$\overline{(D + 1, D)} \text{ XOR } (S + 1, S) \rightarrow (D + 1, D)$		9	●	○		7-15
		DXNRP				9	●	○		7-15
二进制补码	16 bits	NEG		$0 - (D) \rightarrow (D)$		3	●	○		7-19
		NEGP				3	●	○		7-19

表 2. 17逻辑运算指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3. 8. 1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(2) 旋转指令

分类	Unit	指令符号	Symbol	处理内容	执行条件	步数	变址	标志	适用 CPU	页次
右旋	16 位	ROR				3	●	○		7-22
		RORP		n位向右转		3	●	○		7-22
		RCR				3	●	○		7-22
		RCRP		n位向右转		3	●	○		7-22
左旋		ROL				3	●	○		7-24
		ROLP		n位向左转		3	●	○		7-24
		RCL				3	●	○		7-24
		RCLP		n位向左转		3	●	○		7-24
右旋	32 位	DROR				3	●	○		7-26
		DRORP		n位向右转		3	●	○		7-26
		DRCR				3	●	○		7-26
		DRCRP		n位向右转		3	●	○		7-26
左旋		DROL				3	●	○		7-28
		DROLP		n位向左转		3	●	○		7-28
		DRCL				3	●	○		7-28
		DRCLP		n位向左转		3	●	○		7-28

表2.18 旋转指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(3) 移位指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	操作	适用 CPU	页次	
n位移位	16 位	SFR				5	●	●	○		7-31
		SFRP				5	●	●	○		7-31
		SFL				5	●	●	○		7-31
		SFLP				5	●	●	○		7-31
1位移位	n 位	BSFR				7	●		○		7-33
		BSFRP				7	●		○		7-33
		BSFL				7	●		○		7-33
		BSFLP				7	●		○		7-33
1字移位	n word	DSFR				7	●	▲ ^{*3}	△	不适用于 A73.	7-35
		DSFRP				7	●	▲ ^{*3}	△	不适用于 A73.	7-35
		DSFL				7	●	▲ ^{*3}	△	不适用于 A73.	7-35
		DSFLP				7	●	▲ ^{*3}	△	不适用于 A73.	7-35

表2. 19 移位指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3. 8. 1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(4) 数据处理指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	地址	操作	适用 CPU	页次
数据搜索	位	SER	SER (S1) (S2) n	(S1) (S2) n A0: 符合设定值 A1: 符合的个数		9	●	○		7-38
		SERP	SERP (S1) (S2) n			9	●	○		7-38
位检查	16	SUM	SUM (S)	(S) 15 0 A0: "1"的个数		3	●	▲ ^{*3}	○	7-40
		SUMP	SUMP (S)			3	●	▲ ^{*3}	○	7-40
	32	DSUM	DSUM (S)	(S+1) (S) A0: 1的个数		3	●	○		7-40
		DSUMP	DSUMP (S)			3	●	○		7-40
译码 编码	2 ⁿ	DECO	DECO (S) (D) n	8 → 256 位译码 (S) (D) 2 ⁿ bits		9	●	○		7-42
		DECOP	DECOP (S) (D) n			9	●	○		7-42
		ENCO	ENCO (S) (D) n	256 → 8 位编码 (S) (D) 2 ⁿ bits		9	●	○		7-42
		ENCOP	ENCOP (S) (D) n			9	●	○		7-42
7段译码		SEG	SEG (S) (D) n	(S) 3 0 7段译码 (D) n 0 A□N, A3H中M9052 OFF时有效		7	●	▲ ^{*3}	△	不适用于 A3V. 7-44
对字中 某位置位 复位	位	BSET	BSET (D) n	(D) 15 n 0		7	●	○		7-46
		BSETP	BSETP (D) n			7	●	○		7-46
		BRST	BRST (D) n	(D) 15 n 0		7	●	○		7-46
		BRSTP	BRSTP (D) n			7	●	○		7-46
组合 分离	16	DIS	DIS (S) (D) n	4 位 S D+1 D+2 当n=3		9	●	○		7-48
		DISP	DISP (S) (D) n			9	●	○		7-48
		UNI	UNI (S) (D) n	4 位 S+1 S+2 当n=3		9	●	○		7-48
		UNIP	UNIP (S) (D) n			9	●	○		7-48
ASCII 转换	—	ASC	ASC Alphanumeric character (D)	将字母和数字转换成ASCII码,存入以D为首地址的4个字软元件		13	●	○		7-51

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(5) 先入先出指令

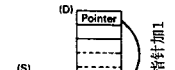
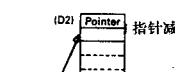
分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
写	16 位	FIFW	FIFW (S) (D)			7	●	○		7-54
		FIFWP	FIFWP (S) (D)			7	●	○		7-54
读		FIFR	FIFR (D1) (D2)			7	●	○		7-54
		FIFRP	FIFRP (D1) (D2)			7	●	○		7-54

表2.21 先入先出指令

(6) 内存缓冲区存取指令

分类	单位	指令符号	符号	处理内容	执行条件	*1 步数	变址	子集	适用 CPU	页次
数据读	1 单字	FROM	FROM n1 n2 (D) n3	从特殊功能模块读取数据		9	●	△	不适用于 A2C 和 A52G.	7-59
		FROMP	FROMP n1 n2 (D) n3			9	●	△	Not applicable to A2C 和 A52G.	7-59
	2 双字	DFRO	DFRO n1 n2 (D) n3			9	●	△	不适用于 A2C and A52G.	7-59
		DFROP	DFROP n1 n2 (D) n3			9	●	△	不适用于 A2C and A52G.	7-59
数据写	1 单字	TO	TO n1 n2 (S) n3	向特殊功能模块写数据		9	●	△	不适用于 A2C and A52G	7-61
		TOP	TOP n1 n2 (S) n3			9	●	△	不适用于 A2C and A52G	7-61
	2 双字	DTO	DTO n1 n2 (S) n3			11	●	△	不适用于 A2C and A52G.	7-61
		DTOP	DTOP n1 n2 (D) n3			11	●	△	不适用于 A2C and A52G.	7-61
数据读	1 单字	FROM	FROM n1 n2 (D) n3 PRC m1 n	从远程 I/O 站读数据		9	●	—	A2CCPU 专用	7-63
		FROMP	FROMP n1 n2 (D) n3 PRC m1 n			9	●	—	A2CCPU 专用	7-63
	2 双字	DFRO	DFRO n1 n2 (D) n3 PRC m1 n			9	●	—	A2CCPU 专用	7-63
		DFROP	DFROP n1 n2 (D) n3 PRC m1 n			9	●	—	A2CCPU 专用	7-63

表2.22 内存缓冲区存取指令 (待续)

*1: 使用扩展软元件或在 A□ACPU 的位软元件上使用变址而引起的步数变化请查阅 3.8.1 节。

*2: 变址栏内的 ▲ 表示指令只可在 A□ACPU 可用变址

*3: 子集栏内的 ▲ 表示此子集指令只可在 A3H, A3M 和 A□ACPU 中执行

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
数据写	1 字	TO	TO n1 n2 (S) n3 PRC m1 n	向从远程终端写数据		9	●	—	A2CCPU专用	7-67
		TOP	TOP n1 n2 (S) n3 PRC m1 n			9	●	—	A2CCPU专用	7-67
	2 字	DTO	DTO n1 n2 (S) n3 PRC m1 n			11	●	—	A2CCPU专用	7-67
		DTOP	DTOP n1 n2 (S) n3 PRC m1 n			11	●	—	A2CCPU专用	7-67

表2.22 内存缓冲区存取指令

(7) FOR/NEXT指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
重复	—	FOR	FOR n	执行FOR和NEXT之间的程序N次		3	*2 ▲	*3 ▲	○	7-71
		NEXT	NEXT			1	*2 ▲		○	7-71

表2.23 FOR/NEXT指令

(8) 就地站, 远程I/O站存取指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	子集	适用 CPU	页次
就地站数据读写	1 字	LRDP		从就地站读数据		11	●	○		7-74
		LWTP		向就地站写数据		11	●	○		7-74
RFRP			从远程I / O站的特殊功能模块读数据		11	●	△	不适用于 A3V.	7-80	
RTOP			向远程I / O站的特殊功能模块写数据		11	●	△	不适用于 A3V.	7-80	

表2.24 就地, 远程I/O站存取指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(9) 显示指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	乘数	适用 CPU	页次
打印	—	PR	PR (S) (D)	从指定的8点字软元件输出16个字符的ASCII码		7	●	△	不适用于 A2C 和 A52G.	7-88
		PR	PR (S) (D)	顺序向输出模块输出ASCII码, 直到结束符NUL(00H)		7	●	△	不适用于 A3V, A2C 和 A52G.	7-88
		PRC	PRC (S) (D)	将字软元件的注释转换成ASCII码, 且输出至输出组件, 软元件1中的注释可输出		7	●	△	不适用于 A2C 和 A52G.	7-88
显示	—	LED	LED (S)	(S) 显示, (S)+7 16个字符		3	●	—	适用于 A3, A3N, A3N-F, A3H, A3M, A3A, A3A-F, A3U, A4U, A73 和 A3N 板	7-94
		LEDA	LEDA Alphanumeric character	显示指定的英文数字字符		13	—	—	适用于 A3, A3N, A3N-F, A3H, A3M, A73 和 A3N 板	7-97
		LEDB	LEDB Alphanumeric character	(LEDA: 前8个字符 LEDB: 后8个字符)		13	—	—	适用于 A3, A3N, A3N-F, A3H, A3M, A73 和 A3N 板	7-97
		LEDC	LEDC (S)	显示软元件S的注释		3	●	—	适用于 A3, A3N, A3N-F, A3H, A3M, A3A, A3A-F, A3U, A4U, A73 和 A3N 板	7-94
显示复位	—	LEDR	LEDR	显示复位		1	—	○		7-99

表 2.25 显示指令

(10) 其它指令

分类	单位	指令符号	符号	处理内容	执行条件	步数	变址	乘数	适用 CPU	页次
复位	—	WDT	WDT	在顺控程序WDT复位		1	—	○		7-102
		WDTP	WDTP			1	—	○		7-102
故障检测	—	CHK	CHK (D1) (D2)	故障: D1, ON; D2, 故障号 正常: D1, OFF; D2, 0 若用A□N, 须I/O直接方式		5	*2 ▲	○		7-104
状态锁存	设置	SLT	SLT	按参数设置的条件, 数据被锁存		1	—	△	不适用于 A1 和 A1N.	7-111
	复位	SLTR	SLTR	状态锁存复位, 且允许执行SLT指令		1	—	△	不适用于 A1 和 A1N.	7-111
采样跟踪	设置	STRA	STRA	按参数设置的条件, 采样数据存入		1	—	△	不适用于 A1 and A1N.	7-113
	复位	STRAR	STRAR	采样跟踪复位, 且允许执行		1	—	△	不适用于 A1 and A1N.	7-113
进位标志	位置	STC	STC	进位标志(M9012)ON		1	—	○		7-115
	复位	CLC	CLC	进位标志(M9012)OFF		1	—	○		7-115
用户定义时钟	1 位	DUTY	DUTY n1 n2 (D)	生成如下所示时钟 		7	*2 ▲	○		7-117

表 2.26 其它指令

*1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。

*2: 变址栏内的▲表示指令只可在A□ACPU可用变址

*3: 子集栏内的▲表示此子集指令只可在A3H, A3M和A□ACPU中执行

(1) 伺服指令

分类	单位	指令符号	符号	处理内容	执行条件	数据步	变址	子集	适用 CPU	页次
请求启动	D 或 I	DSFRP	DSFRP (D) n	请求启动伺服程序		7		—	A73CPU 专用	7-119
数据修改	I	PSFLP	PSFLP (D) n	在定位和步进操作中修改未动作轴 向的当前坐标值和轴向进给率		7		—	A73CPU 专用	7-124

表2.27 伺服指令

- *1: 使用扩展软元件或在A□ACPU的位软元件上使用变址而引起的步数变化请查阅3.8.1节。
- *2: 变址栏内的 ▲ 表示指令只可在A□ACPU可用变址
- *3: 子集栏内的 ▲ 表示此子集指令只可在A3H, A3M和A□ACPU中执行

3. 指令结构

3.1 指令结构

1) 许多指令如下可以分成1个指令部分和1个软元件部分, 表示如下:

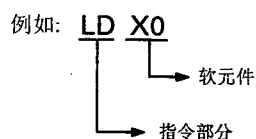
$\left\{ \begin{array}{ll} \text{指令部分} & \cdots \cdots \text{表示功能} \\ \text{软元件} & \cdots \cdots \text{表示指令要用到的数据} \end{array} \right.$

2) 指令结构大致可按指令和软元件的组合分类如下:

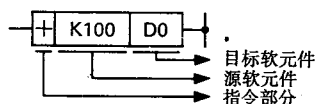
a) $\boxed{\text{指令部分}} \cdots \cdots$ 保留软元件状态, 且主要用于控制程序的执行

例如: END, FEND

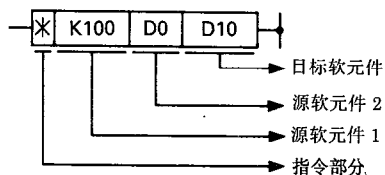
b) $\boxed{\text{指令部分}} + \boxed{\text{软元件}} \cdots \cdots$ 按软元件的状态, 程序的分支等, 切换软元件的ON/OFF, 控制执行条件。



c) $\boxed{\text{指令部分}} + \boxed{\text{源软元件}} + \boxed{\text{目标软元件}} \cdots \cdots$ 用目标软元件与源软元件执行运算, 运算结果存入目标软元件。



d) $\boxed{\text{指令部分}} + \boxed{\text{源软元件 1}} + \boxed{\text{源软元件 2}} + \boxed{\text{目标软元件}} \cdots \cdots$ 用源软元件1和源软元件2执行运算, 运算结果存入目标软元件。



e) 其它..... a)到d)的组合

(1) 源数据 (S)

1) 源数据用于操作运算

2) 源数据遵循如下规定

. 常数..... 指定用于操作运算的数字值. 这个值只能在写程序时确定, 程序运行时不能再修改。

. 位软元件, 字软元件..... 指定用于存储数据的软元件. 因此操作运算初始化以前, 必须向指定的软元件中存入数据. 当程序运行时修改指定软元件中的数据, 指令所用的数据就可被修改。

(2) 目标 (D)

1) 储存操作运算执行后的结果. 当指令由指令部分+源软元件+目标软元件组成时, 在操作开始之前必须将用于操作的数据存入目标软元件。

2) 存储数据的软元件必须在目标软元件中加以指定

备 注

1) 此手册中源和目标的表示方式如下:

源 ⑤
源1 ⑤1
源2 ⑤2
目标 ⑥

3.2 位处理

当已指定位软元件 (X, Y, M, L, S, B, F) 时, 就进行位处理. 可以选择或是1位处理, 或是用16, 32位的指令去进行数字规格处理.

3.2.1 1位处理

当使用顺控指令时, 不能指定超过1位的位处理.

例如: LD X0, OUT Y20

3.2.2 数字规格处理

当使用基本和应用指令时, 可以用数字的数目去指定位软元件的数目. 16位指令中可以每4点为增量单位, 最多可指定16点. 32位指令中最多可指定32位软元件.

(1) 16位指令: K1 ~ 4 (4 ~ 16点)

例如: 用16位数据的数字规格设定范围, X0 ~ F.

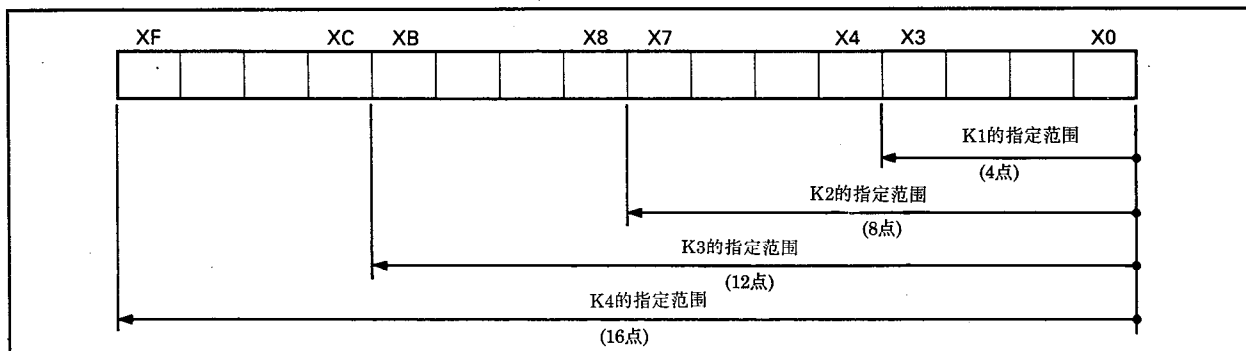


图3.1 16位指令的数字规格范围

(a) 当源数据 (S) 处有数字指定时, 数值范围按源数据处理, 如下表3.1所示

指定数字	16位指令
K1 (4点)	0 至 15
K2 (8点)	0 至 255
K3 (12点)	0 至 4095
K4 (16点)	-32768 至 32767

表3.1 数字指定规格和所处理的数值

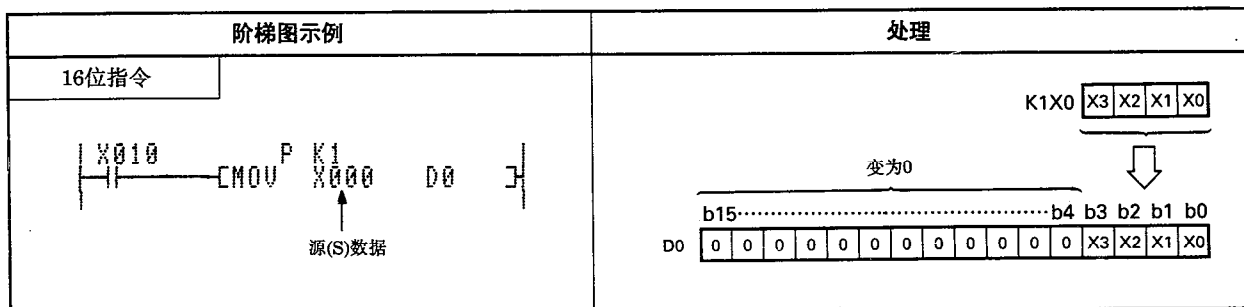


图3.2 阶梯图示例和处理

b) 当目标侧有数字规定, 点数就按照目标侧的数字规定。

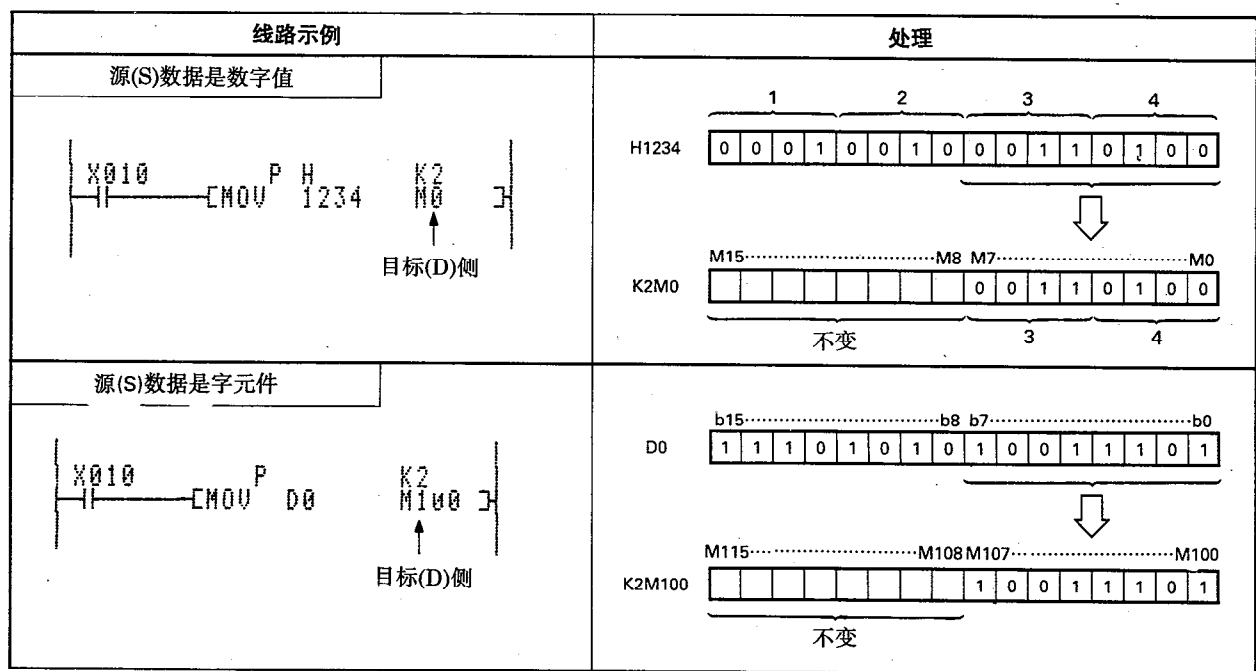


图3.3 梯形图示例和处理

(2) 32位指令: K1 ~ 8 (4 ~ 32点)

例如: 32位的数字规定范围是 X0 ~ F

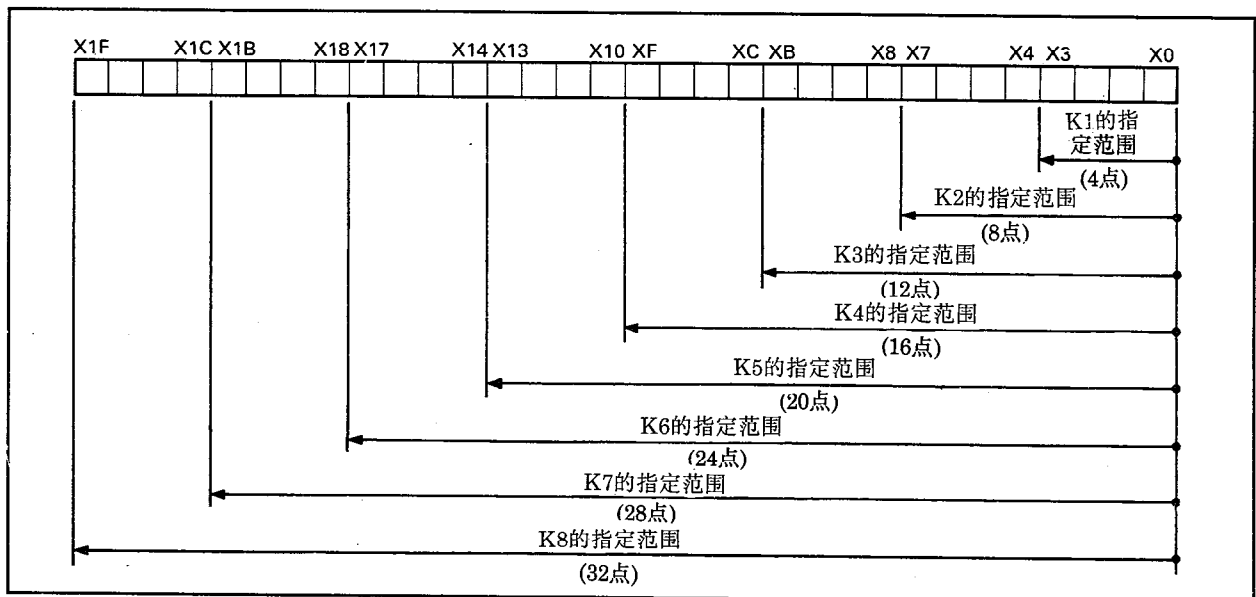


图3.4 32位指令的数字指定范围

指定数字	32位指令	指定数字	32位指令
K1 (4点)	0 至 15	K5 (20点)	0 至 1048575
K2 (8点)	0 至 255	K6 (24点)	0 至 167772165
K3 (12点)	0 至 4095	K7 (28点)	0 至 268435455
K4 (16点)	0 至 65535	K8 (32点)	-2147483648 至 2147483647

表 3.2 数字规定和处理的数值

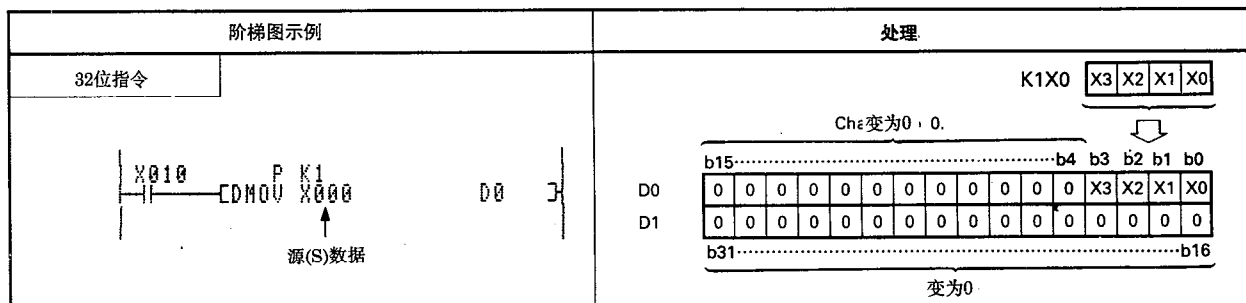


图3.5 阶梯图示例和处理

(4) 当目标侧有数字指定, 点数就由用于目标侧的数字规格处理。

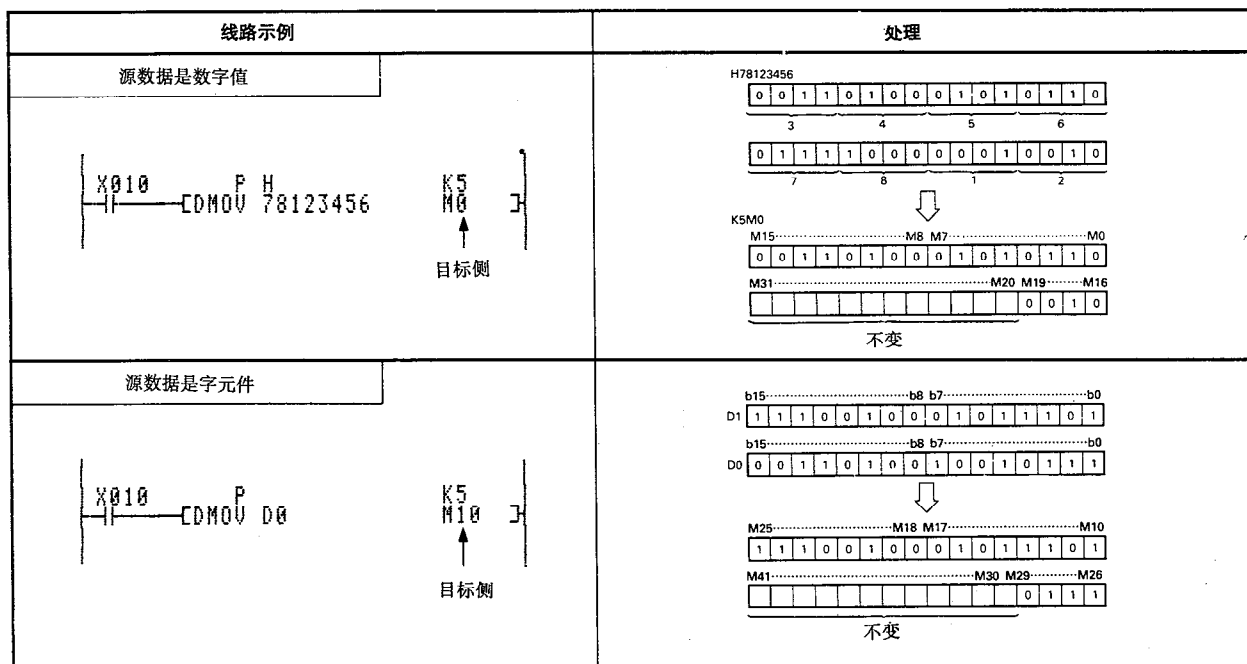


图3.6 阶梯图示例和处理

要点

对于数字规格处理, 任何值都可作为位软元件的首地址值。

3.3 数值处理

在A系列中,有处理16位和32位数值的指令,16位或32位的最高位用于判断正负.所以用16位和32位处理的数值如下:

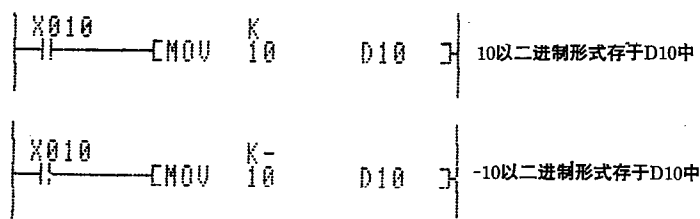
16位: -32768 ~ 32767

32位: -2147483648 ~ 2147483647

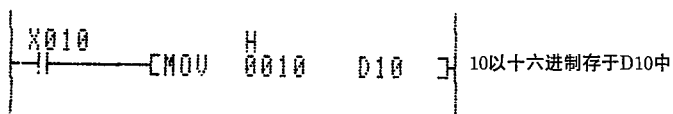
注意

(1) 数值设定步骤

1) 十进制

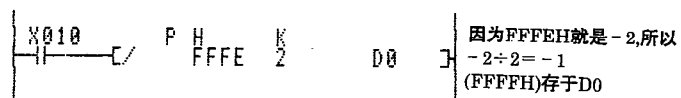


2) 16进制

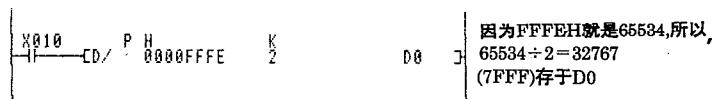


(2) 当FFFEH 被2除时,会发生如下情况

16位指令



32位指令



当用16位或32位处理的数值超过了规定(上溢,下溢),细节可看下表

	16位数据处理		32位数据处理	
	十进制表示	十六进制表示	十进制表示	十六进制表示
上溢	↑ -32765	8 0 0 3 _H	↑ -2147483645	8 0 0 0 0 0 0 3 _H
	-32766	8 0 0 2 _H	-2147483646	8 0 0 0 0 0 0 2 _H
	-32767	8 0 0 1 _H	-2147483647	8 0 0 0 0 0 0 1 _H
	-32768	8 0 0 0 _H	-2147483648	8 0 0 0 0 0 0 0 _H
	32767	7 F F F _H	2147483647	7 F F F F F F F _H
	32766	7 F F E _H	2147483646	7 F F F F F F E _H
	32765	7 F F D _H	2147483645	7 F F F F F F D _H
	32764	7 F F C _H	2147483644	7 F F F F F F C _H
	⋮		⋮	
下溢	-32765	8 0 0 3 _H	-2147483645	8 0 0 0 0 0 0 3 _H
	-32766	8 0 0 2 _H	-2147483646	8 0 0 0 0 0 0 2 _H
	-32767	8 0 0 1 _H	-2147483647	8 0 0 0 0 0 0 1 _H
	-32768	8 0 0 0 _H	-2147483648	8 0 0 0 0 0 0 0 _H
	32767	7 F F F _H	2147483647	7 F F F F F F F _H
	32766	7 F F E _H	2147483646	7 F F F F F F E _H
	32765	7 F F D _H	2147483645	7 F F F F F F D _H
	32764	7 F F C _H	2147483644	7 F F F F F F C _H
	⋮		⋮	

表3.3 超出允许数值范围的处理

即使发生上溢或下溢,进位标志和出错标志也不改变.
十进制对应十六进制表示如下表

十进制表示	十六进制表示
5	0 0 0 5 _H
4	0 0 0 4 _H
3	0 0 0 3 _H
2	0 0 0 2 _H
1	0 0 0 1 _H
0	0 0 0 0 _H
-1	F F F F _H
-2	F F F E _H
-3	F F F D _H
-4	F F F C _H
-5	F F F B _H
⋮	⋮
-32768	8 0 0 0 _H

要点

当要使用大于等于32768或小于等于-32768的十进制数时,请用32位数据处理.

3.4 32位数据存储

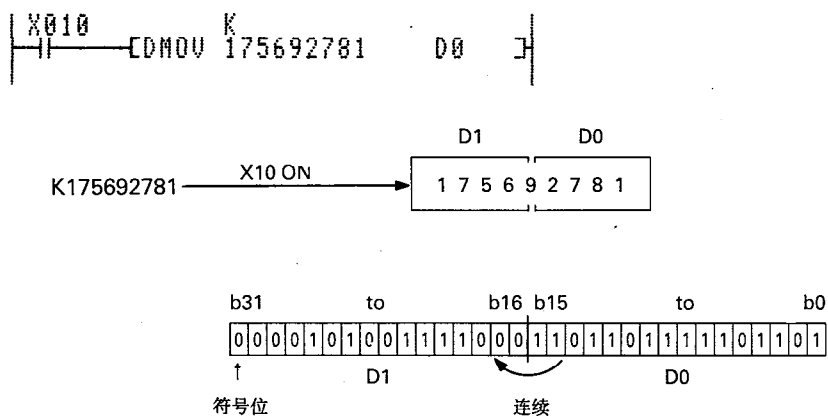
32位数据存入位软元件时使用数字规格K1~K8, 用字软元件存储时使用2个地址连续的字软元件。

(1) 存入位软元件中

参阅3.2.2(2)节

(2) 存入字软元件中

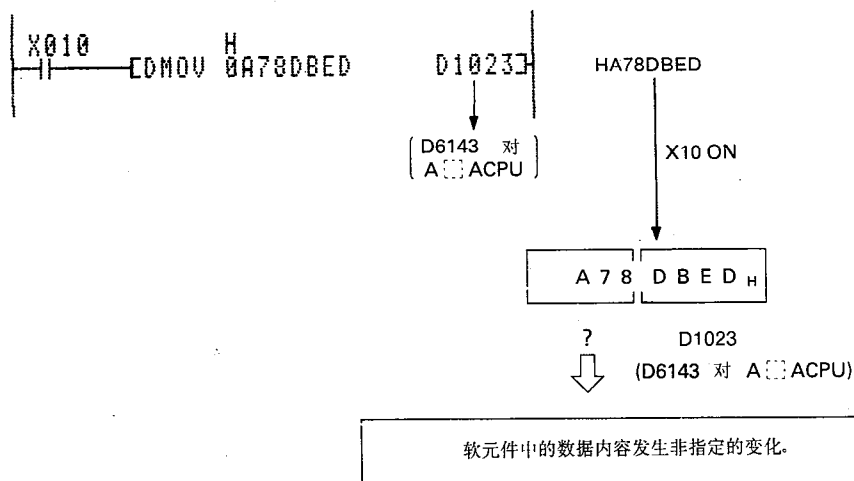
(a) 2个连续的字软元件可用来存储32位数据



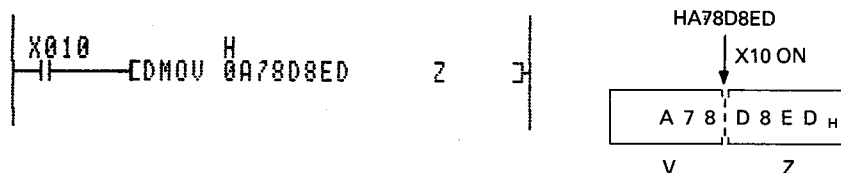
(b) 在用32位指令将具有数字规格K1~K8的位软元件数据存入字软元件时, 参阅3.2.2(1)节

(c) 注意

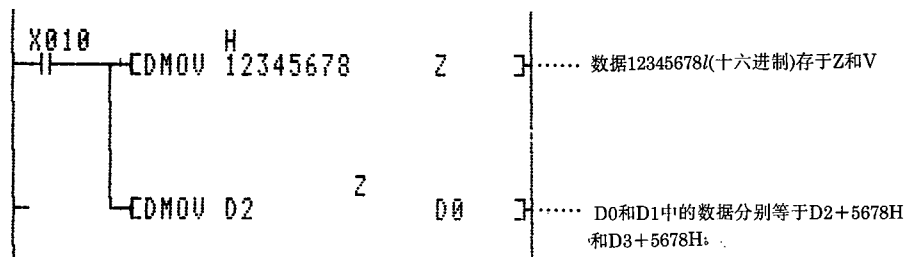
1) 即使将32位数据赋予最大地址的软元件, 也不会出错, CPU会自动将高位数据存入某个尚未定义的软元件。



- 2) 当V和Z成对使用时, 变址寄存器可处理32位数据. 这时Z作为低16位软元件, 因此V不能再用于32位指令. (在编程程序不能键入)



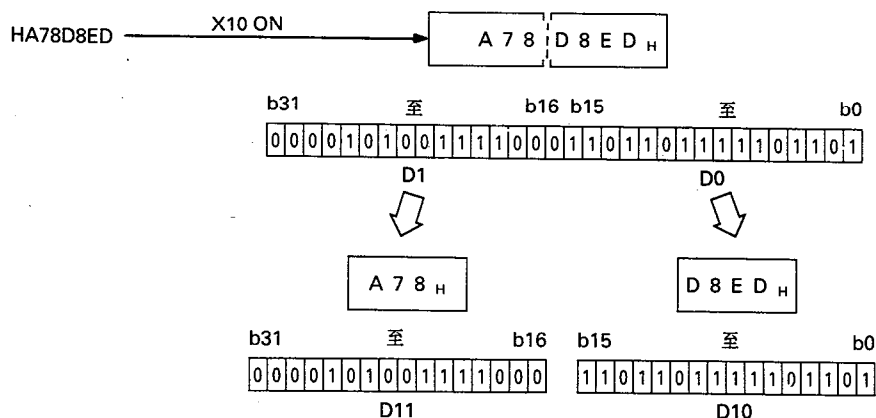
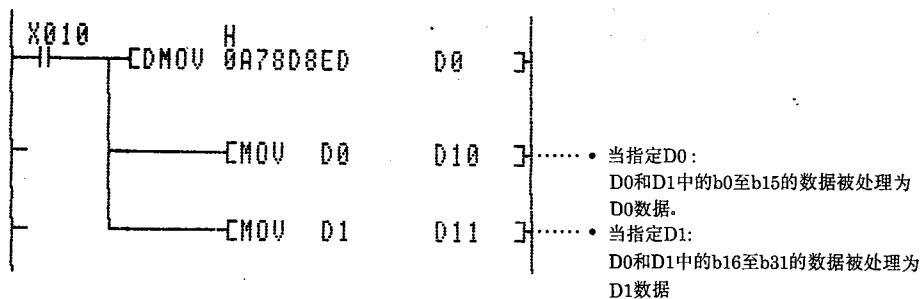
在指令中V, Z之一被指定成变址指针时, 即使V, Z被赋以32位数值, 它们只能分别作为16位数据加以处理.



备注:

在A□ACPU中使用扩展变址寄存器Z1~Z6, V1~V6处理32位数据时, 请查阅3.8.5节.

- 3) 如2个连续的字软元件存有32位数据, 其中之一用于16位指令, 它的处理过程如下:



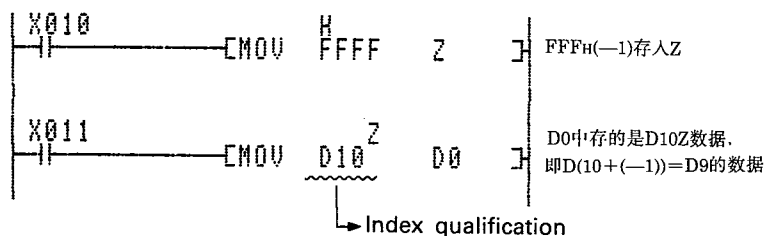
3.5 变址

(1)用变址指定的软元件地址号由指定软元件号加上变址寄存器(Z, V)的内容得到。

(2)可用变址指定的软元件有X, Y, M, L, S, B, F, T, C, D, R, W, K, H和P。

(3)变址寄存器的值有符号位, 其范围是-32768 ~ 32767

(4)变址指定的例子如下:



示例:

在执行变址设定时, 实际处理的软元件过程如下所示:

阶梯图例子	实际软元件处理
	解释: $\begin{cases} K100Z \cdots K(100 + 20) = K120 \\ W53V \cdots W(53 - 5) = W4E \end{cases}$ 十六进制
	解释: $\begin{cases} K2X50Z \cdots K2X(50 + 14) = K2X64 \\ K1M38 \cdots K1M(38 - 5) = K1M33 \end{cases}$ K20转换成十六进制
	解释: $\begin{cases} D0Z \cdots D(0 + 20) = D20 \\ K3Y12F \cdots K3Y(12F - 5) = K3Y12A \end{cases}$ 十六进制

图3.7 阶梯图示例和实际处理的软元件

(5) 在下列情况下基本指令和应用指令会产生运算出错:

(a) 在执行变址且软元件地址号超过范围时, 当然这不包括K和H.

变址	线路示例	判断
$Z = -10$		因为 $T(9 - (-10)) = T-1$, 所以操作出错
$Z = 10$		因为 $D(1020 + 10) = D1030$, 超过了 $D0 \sim 1023$ 的范围, 所以操作运算出错.
$Z = 10$		因为 $K(32767 + 10) = K-32759$, 不产生操作运算出错 $(32767 + 10) \rightarrow (7FFF_H + A_H) \rightarrow (8009_H) \rightarrow -32759$

图3.8 阶梯图示例和判断

(b) 在执行变址设定且位软元件的首地址超过了该软元件地址范围.

变址	线路示例	判断
$Z = 15$		虽然指定了 $K4B3FF (B(3F0 + F) = B3FF)$, 但未发生操作运算出错
$Z = 16$		因为 $K4B400 (B(3F0 + 10) = B400)$ 超过了 B 的地址范围, 所以出错

图3.9 阶梯图示例和判断

要点

使用A□ACPU时, 以上指定不会引起出错, 但顺控程序不会正确执行(参见3.8.4节)

(6) 使用A□ACPU时, 变址可用于位软元件, 支持LD, OUT等指令.

3.6 子集处理

子集处理用于在基本和应用指令中指定软元件时,若满足下列条件可提高处理速度.指令符号与常规处理的指令符号一样.

CPU 类型	变址规定	位软元件	字软元件
An AnN(-F) A3V, A2C, A52G A0J2H, A1S A73, A3N board	不能用	- 数字规格必须为K4(16位处理)或K8(32位处理) - 指定的位元件号必须是8的倍数	无此类情况
A3H, ASM	不能用	- 数字规格必须为K4(16位处理)或K8(32位处理) - 指定的位元件号必须是8的倍数	文件寄存器(R)必不可用
AnA(-F), AnU	不能用于位软元件	- 数字规格必须为K4(16位处理)或K8(32位处理) - 指定的位元件号必须是8的倍数	不能用于文件寄存器(R)和变址寄存器(Z, V)(当对字软元件使用变址时, V, Z就排除在外)

表3.4 子集处理条件

3.7 操作运算出错

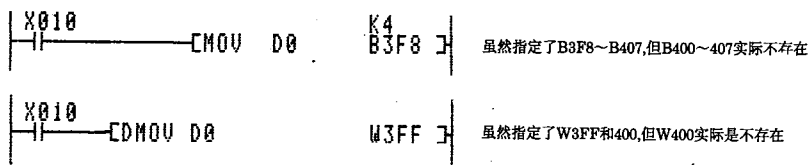
(1) 在下列情况下,基本和应用指令会发生操作运算出错.

(a) 在发生指令描述性的错误

(b) 在变址的结果包含错误时(见3.5(5)节)

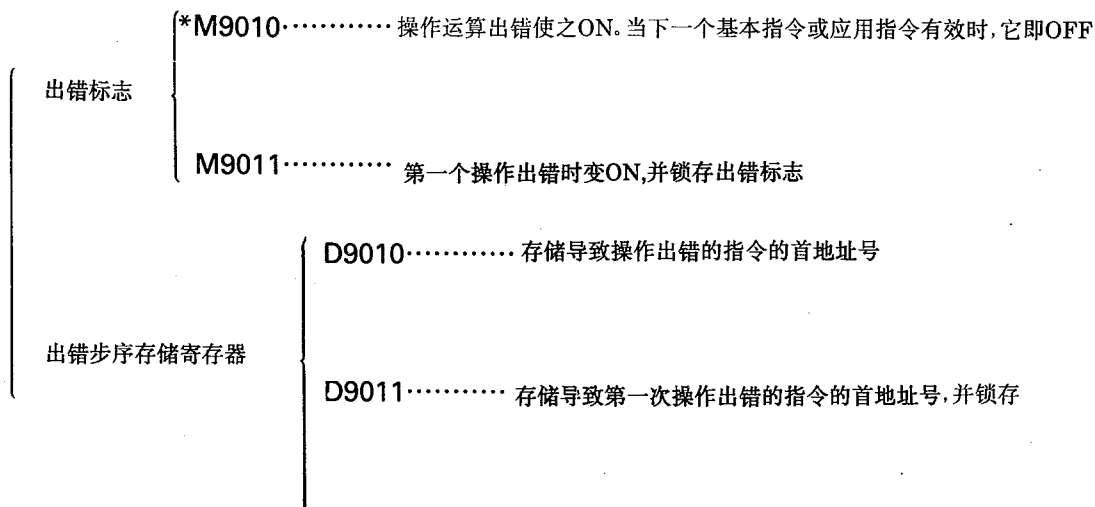
要点

当软元件的地址指定超过了允许值,数据会被自动写到未被指定的软元件中而不出错,因此要多加注意



(2) 出错处理

当基本或应用指令执行时发生操作出错, 出错标志 (M9010, M9011) 会变ON, 出错步序号会存入寄存器D9010, D9011.



*A3H, A3M, A□ACPU无此M9010.

- 1) 当程序出错时, M9011从OFF到ON, D9011存储出错指令的步序号. 因此如果M9011保持ON, 则D9011的内容不变.
- 2) 对M9011和D9011复位的程序如下.

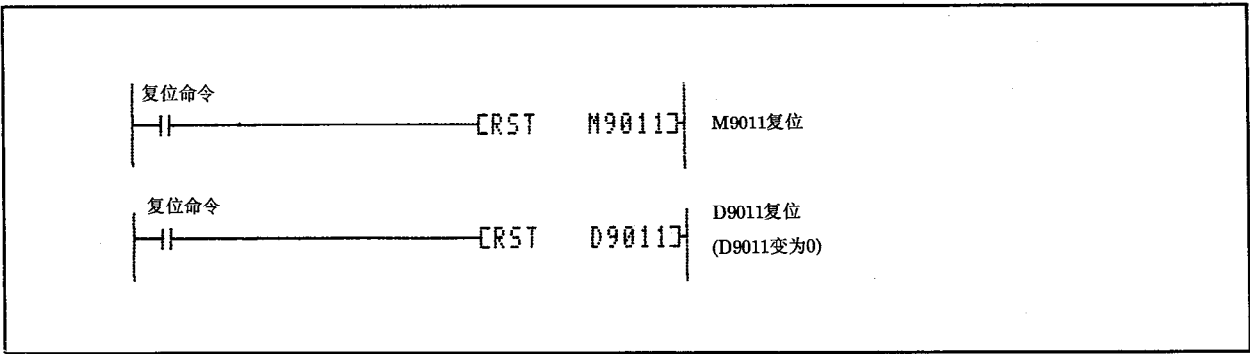


图3. 10 特殊继电器和寄存器的复位

- 3) 若操作出错发生时, 按参数设置的选择可使程序继续运行或停止. 详细请看ACPU编程手册(基础篇).

3.8 使用AnA(-F)和AnU注意事项

本节提出了使用AnA(-F)和AnU的注意事项

3.8.1 指令步数

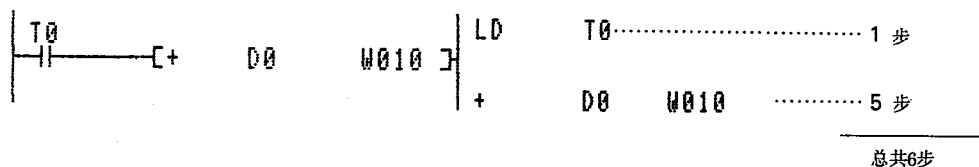
(1) 当使用如下软元件(用AnACPU(-F)和AnU扩展的软元件)时, 每用一次, 步数加1.

软元件名称		范围	
		AnA(-F)	AnU
内部继电器	M, L, S	2048 至 8191	
定时器	T	256 至 2047	
计数器	C	256 至 1023	
通讯用继电器	B	400 至 FFF	400 至 1FFF
数据寄存器	D	1024 至 6143	1024 至 8191
通讯用寄存器	W	400 至 FFF	400 至 1FFF
信号报警器	F	256 至 2047	
变址寄存器	Z	1 至 6	
变址寄存器	V	1 至 6	

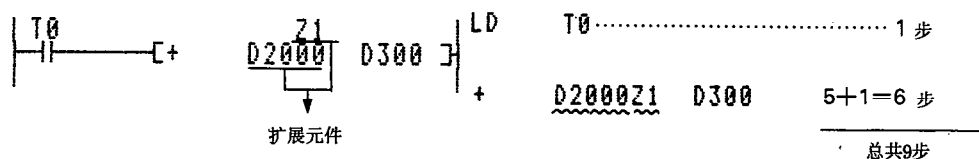
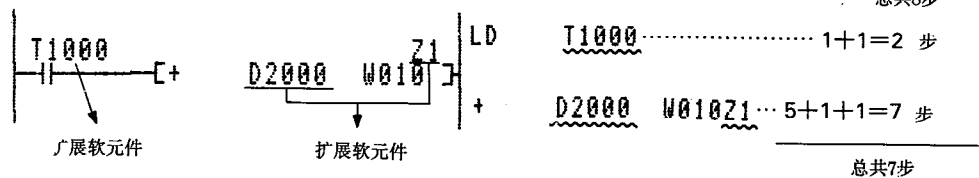
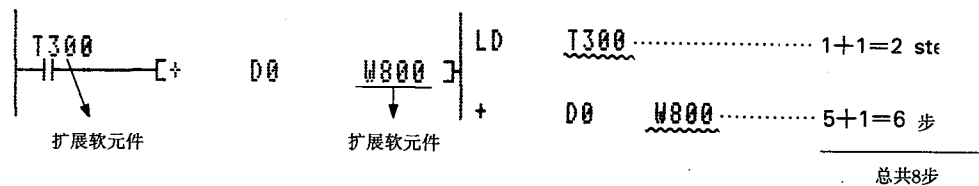
若对扩展软元件执行变址, 且使用扩展变址寄存器, 步数仅增加1步。

举 例

· 仅使用基本软元件



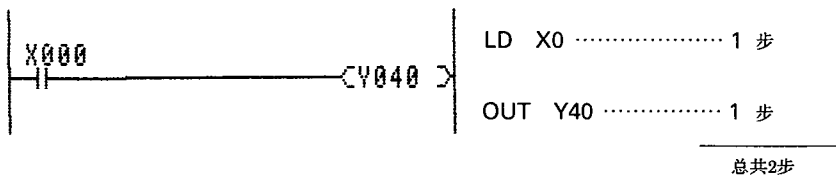
· 使用扩展指令



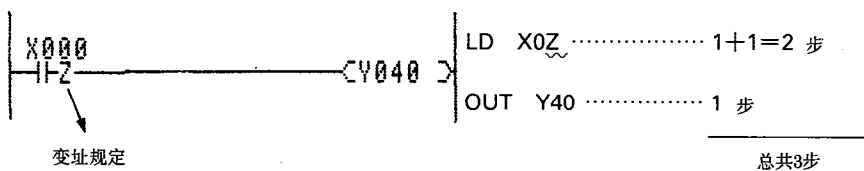
(2) 当变址用于步数仅为1步的顺控指令(如LD, OUT), 步数加1.

例 子

· 未用变址



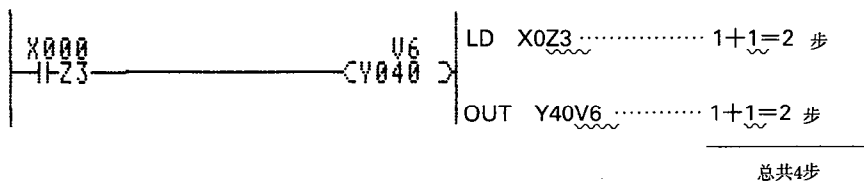
· 使用变址



备 注

当A□ACPU中的扩展变址寄存器(Z1-Z6, V1-V6)用于步数仅为1的顺控指令(如LD、OUT), 步数只增加1步。

例 子



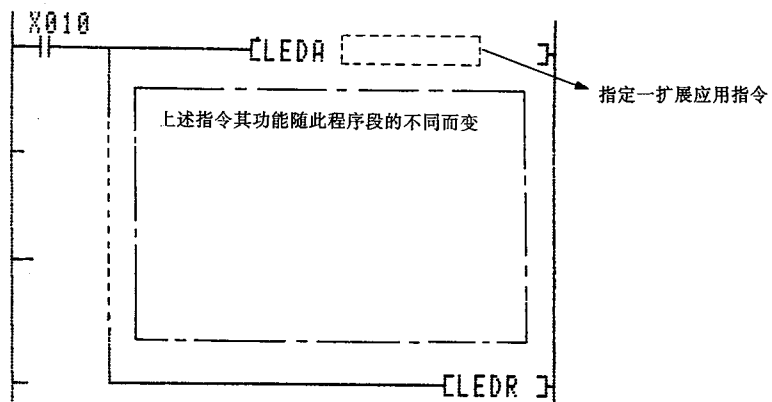
3.8.2 可变功能的指令

下列指令用于AnA(-F)和AnU扩展应用指令块时,其处理内容是可变的.具体请看AnACPU/AnUCPU编程手册(专用指令编).

指令	通常	在扩展指令块中
PRC	注释输出	MELSECNET/MINI-S3 的支持指令
FROM DFRO TO DTO	特殊功能模块缓冲寄存器存取	MELSECNET/MINI-S3 的支持指令
LEDA LEDB	未用	扩展应用指令起始
LEDC	LED 注释显示	指定软元件
DXNR	逻辑同或指令	指定32位常数
LEDR	LED 和信号报警器复位	扩展应用指令终止
SUB	未用	指定16位常数

备 注

A□ACPU中的扩展应用指令块如下图所示.



上面没有提及的指令不可用于扩展应用指令块中.

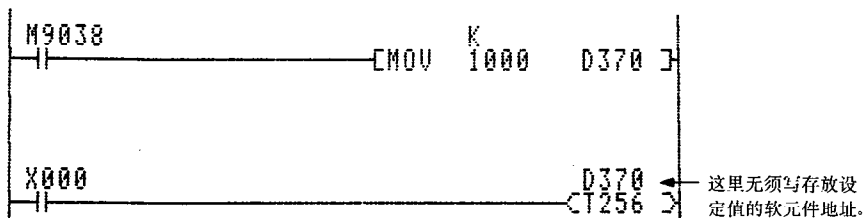
3.8.3 扩展定时器和计数器的设定值

在A□ACPU中使用扩展定时器和计数器,其设定值由参数设置初始化时所指定的字软元件D, W还是R确定. 具体请看A2A (S1) / A3ACPU用户手册(控制功能篇)。

定时器 T	256 至 2047
计数器 C	256 至 1023

例 子

在参数设置中指定D370存放T256的设定值



例如:

当  T256和
 键入, T256的设定
 值软元件D370自动显示

3.8.4 使用变址的须知

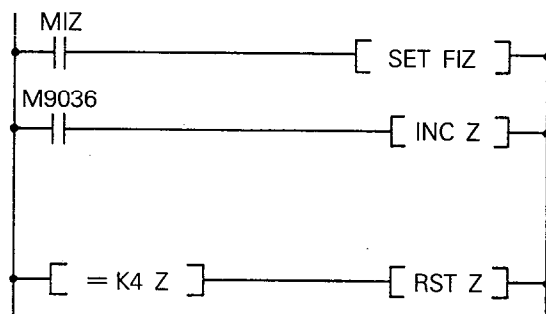
- (1) 为增快运算速度而使用变址时, A□ACPU不校核软元件地址号. 因此无法检测到因变址而导致的错误. 当发生变址出错时, 未经指定的软元件中的数据会被改变. 所以程序中含有变址时须多加小心.

(2) 在变址时的通/断指令操作

在用AnA或AnU时,若通/断指令(PLS, PLF, SET□, RST□, □P)以变址方式指定其操作软元件,仅当它们的执行条件满足时指令才会执行。

例1

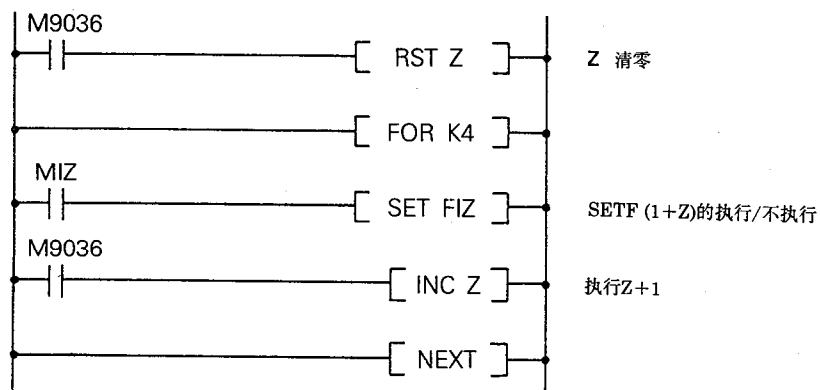
在以下所示电路中,当M1、M2和M4为ON,而M3为OFF时



扫描次数	MIZ*3		SET FIZ		FIZ	
	软元件地址	ON/OFF 状态	执行条件*1	执行/不执行状态	软元件地址	ON/OFF 状态
第1次扫描	M1	ON	*2	*2	F1	*2
第2次扫描	M2	ON	ON → ON (不满足)	不执行	F2	OFF
第3次扫描	M3	OFF	ON → OFF (不满足)	不执行	F3	OFF
第4次扫描	M4	ON	OFF → ON (满足)	执行	F4	ON

例2

在以下所示电路中,当M1、M2、M4当ON,而M3为OFF时



FOR 指令	MIZ		SET FIZ		FIZ	
	软元件地址	ON/OFF 状态	执行条件	执行/不执行状态	软元件地址	ON/OFF 状态
第1次	M1	ON	*2	*2	F1	*2
第2次	M2	ON	ON → ON (未满足)	不执行	F2	OFF
第3次	M3	OFF	ON → OFF (未满足)	不执行	F3	OFF
第4次	M4	ON	OFF → ON (满足)	执行	F4	ON

备 注

(1) * 1: 执行/不执行由软元件的当前状态与前一次扫描时的状态/上一次的状态进行比较来决定。

当前软元件

前一次扫描时/前一次的软元件

M1	M1	M4	M4
M2	M2	M1	M1
M3	M3	M2	M2
M4	M4	M3	M3

(2) * 2: 根据前一次扫描时M4的ON/OFF状态而变化。

前一次扫描 周期M4的状态	SET FIZ		FIZ	
	执行条件	执行/不执行状态	软元件地址	ON/OFF 状态
OFF	OFF → ON (满足)	执行	F1	ON
ON	ON → ON (未满足)	不执行		OFF

(3) * 3: 软元件状态以四次扫描中M1、M2、M3和M4的次序而变化, 同时, 在第5次扫描返回M1。

3.8.5 在变址寄存器存储32位数据

AnA(-F)和AnU的扩展变址寄存器(Z1~Z6, V1~V6)可存储32位数据。
下列变址寄存器可配对存储32位数据

- | | | |
|-------|---|----|
| 1) Z1 | 和 | V1 |
| 2) Z2 | 和 | V2 |
| 3) Z3 | 和 | V3 |
| 4) Z4 | 和 | V4 |
| 5) Z5 | 和 | V5 |
| 6) Z6 | 和 | V6 |

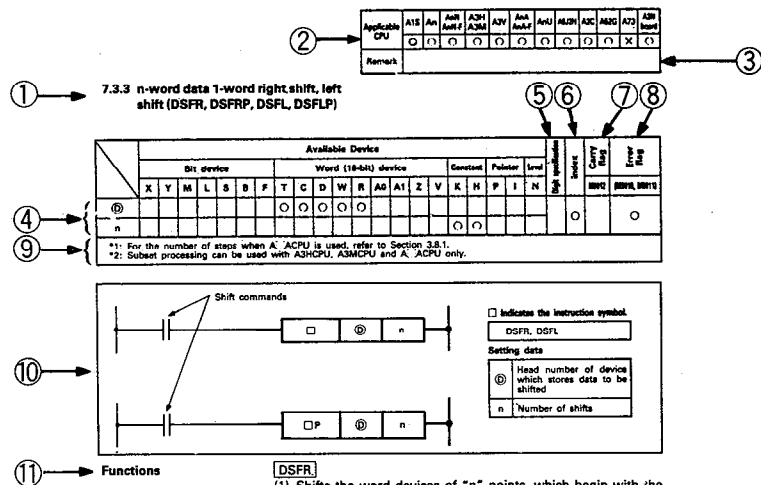
由于Zn被用作存放低16位数据的软元件, 所以Vn不能在32位指令中直接表现出来(即程序中不出现Vn)

除了上述配对的组合, 其它组合均不能存储32位数据. 如果配对软元件中的1个寄存器在某个指令中用作变址, 此时这两个寄存器就都只能作为变址用的16位数据寄存器.

4. 指令格式

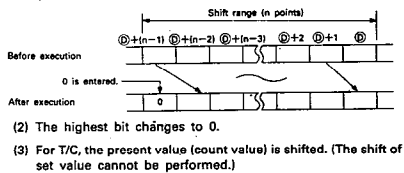
4. 指令格式

在以下章节中, 用本章所描述的如下格式解释指令.



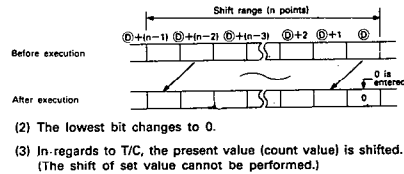
DSFR

(1) Shifts the word devices of "n" points, which begin with the device specified at ⑤, to the right by one bit.

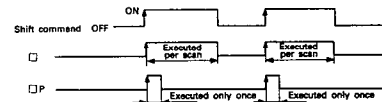


DSFL

(1) Shifts the word devices of "n" points, which begin with the device specified at ⑤, to the left by one bit.



⑫ → Execution Conditions



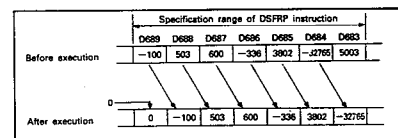
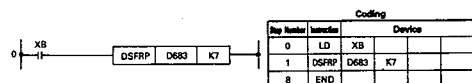
⑬ → Operation Error

In the following case, operation error occurs and the error flag turns on.
○ "n" is a negative value.

⑭ → Program Examples

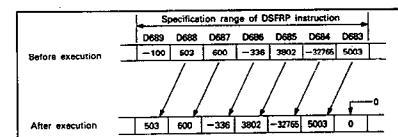
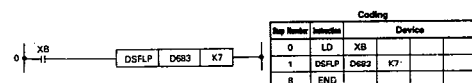
DSFR

Program which shifts the contents of D683 to 689 to the right when XB turns on.



DSFL

Program which shifts the contents of D683 to 689 to the left when XB turns on.



解 释

- (1) 表示指令所在的章节数, 标题和指令助记符
- (2) 表示适用的CPU.
 - : 适用
 - △: 对某些CPU适用或者使用需要一些特殊操作.
 - ×: 不适用.

如果指令适用于所有的CPU, 它的表示方式为:

适用	全部型号CPU
----	---------

- (3) 详细解释②中的内容. 对标有△的须特别注意.
- (4) 标有圆圈的软元件表示能用此指令
- (5) 表示当位软元件有数字规格时所能用的数字
- (6) 圆圈表示该指令可使用变址 (把V或Z的内容加上原地址), 三角表示在特定几种CPU上才有此功能.
- (7) 圆圈表示该指令能将进位标志置位
- (8) 圆圈表示该指令能在操作出错时将出错标志置位.
- (9) 提供与以上④到⑩有关的注意事项. 对标有○和△的须特别注意
- (10) 表示在阶梯图方式下的指令格式
- (11) 描述该指令
- (12) 表示该指令的执行条件
- (13) 表示导致操作出错的条件
- (14) 提供阶梯图和指令表方式下的程序举例

备注

以下所示是指令表方式下的程序

0	LD	M9036		
1	DBIN	K6X020	D9	
10	D*	D9	K10000	D5
21	BIN	K4X010	D3	
26	MOV	K0	D4	
31	D+	D3	D5	D0
42	END			

程序的输入步骤请查阅相关外部设备的操作手册.

5. 顺控指令

用于继电器控制电路的顺控指令, 按如下分类:

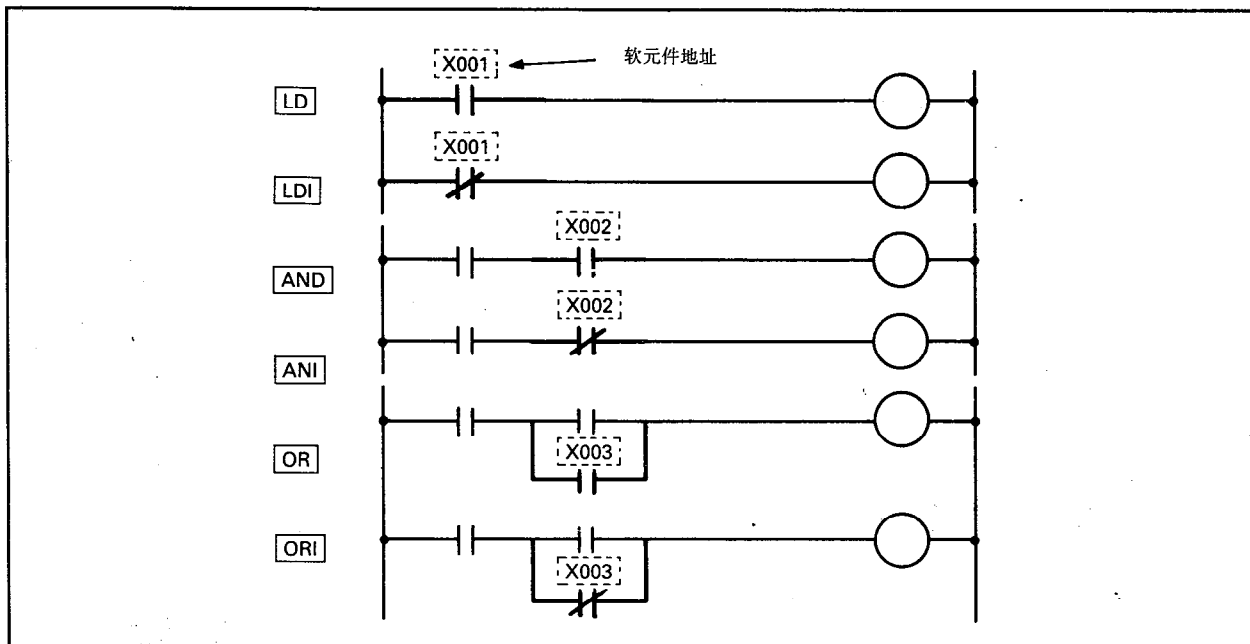
分类	说明	参见
触点指令	操作开始, 串联连接, 并联连接	5-2
连接指令	电路块串联, 并联连接, 操作结果存储	5-5
输出指令	位软元件输出, 微分输出, 置位, 复位, 输出取反	5-14
移位指令	位软元件移位	5-27
主控指令	主控置位, 复位	5-29
终止指令	顺控程序终止	5-33
其它指令	顺控程序停止, 空操作	5-37

5.1 触点指令

5.1.1 操作开始, 串联, 并联连接
(LD, LDI, AND, ANI, OR, ORI)

适用 CPU	全部型号CPU
-----------	---------

可用软元件																				数字规格	变址	总 地址 进位标志	出错标志	
位软元件								字(16位)软元件								常数		指针						故障级
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I					N
○	○	○	○	○	○	○	○	○														△		
• 1: 有关使用A□ACPU时的程序的步数, 参看3.8.1节																								



功能

LD, LDI

- (1) LD是常开触点操作起始指令, LDI是常闭触点操作起始指令。它们获得指定软元件的ON/OFF数据, 并以此作为操作结果。

AND, ANI

- (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。它们读取指定软元件的ON/OFF数据, 采用此数据与前面的操作结果进行与操作, 结果作为新的操作结果。
- (2) 使用AND, ANI无限制。然而, 在GPP的阶梯图模式中有下列限制:
- ① 写: 当AND 或ANI串联连接时, 最多可写21级电路。
 - ② 读: 当AND 或ANI串联连接时, 一次最多可显示24级电路。
若电路超过25级, 则一次只显示1到24级。

OR, ORI

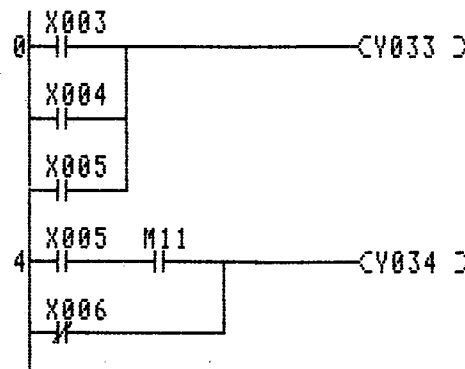
- (1) OR是常开触点并联连接指令, ORI是常闭触点并联连接指令。这些指令从指定的软元件获取ON/OFF数据, 并和前面的操作结果作或操作, 并将它用作新的操作结果。
- (2) 使用OR或ORI无限制。然而, 在GPP的阶梯图模式中有下列限制:
- ①写: 在一个电路中最多可连续连接23个OR或ORI指令。
 - ②读: 在一个电路中最多可显示23个连续连接的OR或ORI。一个电路中, 不能显示超过23个OR或ORI的电路。

执行条件

与软元件的状态和操作结果无关, 每个扫描周期都执行。

编程举例

LD, LDD, AND, ANI, OR, ORI

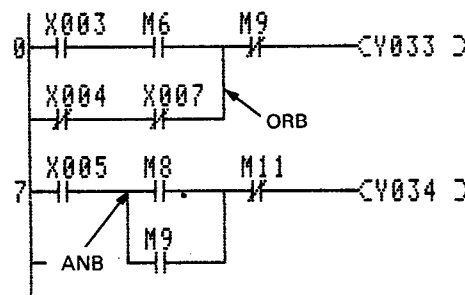


• 指令代码

```

0 LD    X003
1 OR    X004
2 OR    X005
3 OUT   Y033
4 LD    X005
5 AND   M11
6 ORI   X006
7 OUT   Y034
8 END

```

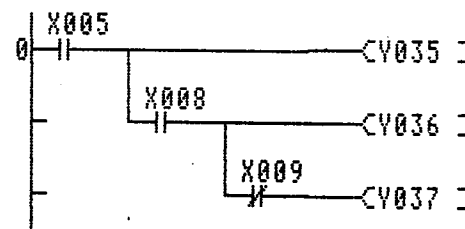


• 指令代码

```

0 LD    X003
1 AND   M6
2 LDI   X004
3 ANI   X007
4 ORB
5 ANI   M9
6 OUT   Y033
7 LD    X005
8 LD    M8
9 OR    M9
10 ANB
11 ANI   M11
12 OUT   Y034
13 END

```



• 指令代码

```

0 LD    X005
1 OUT   Y035
2 AND   X008
3 OUT   Y036
4 ANI   X009
5 OUT   Y037
6 END

```

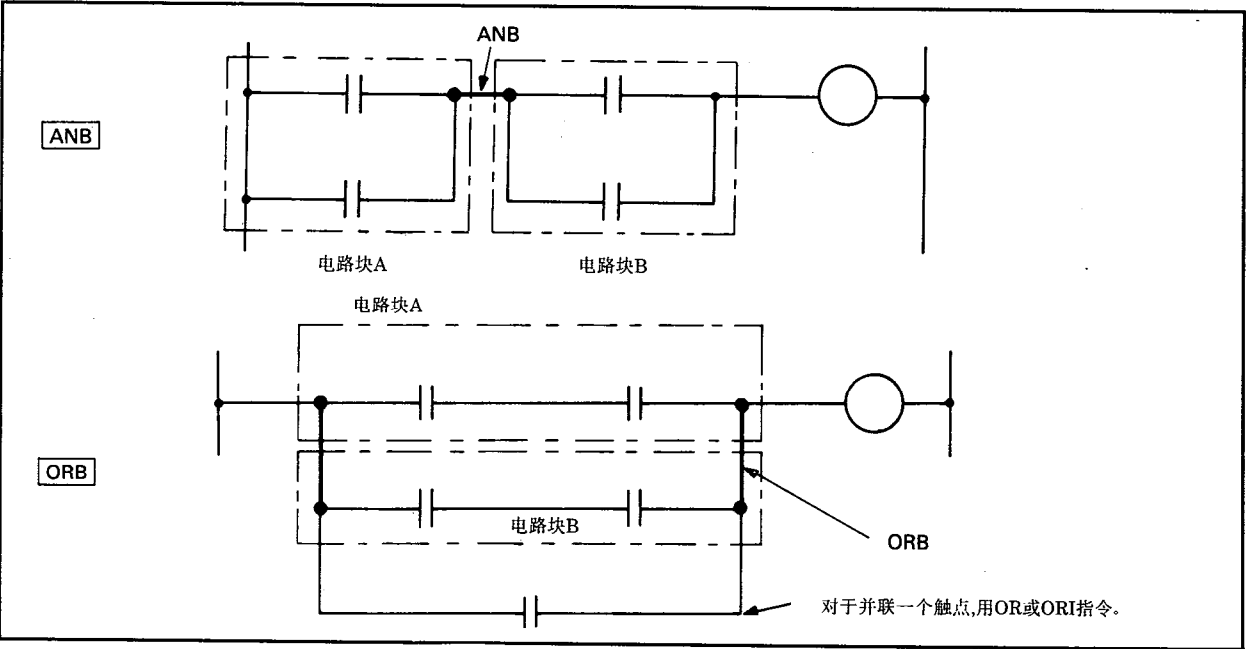
[illegible]

5.2 连接指令

5.2.1 电路块串联连接, 并联连接 (ANB, ORB)

适用CPU	所有型号的CPU组件
-------	------------

可用软元件																				D 数字规格	变址	进位标志	出错标志		
位软元件							字(16位)软元件								常数		指针		取番 级						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I					N	



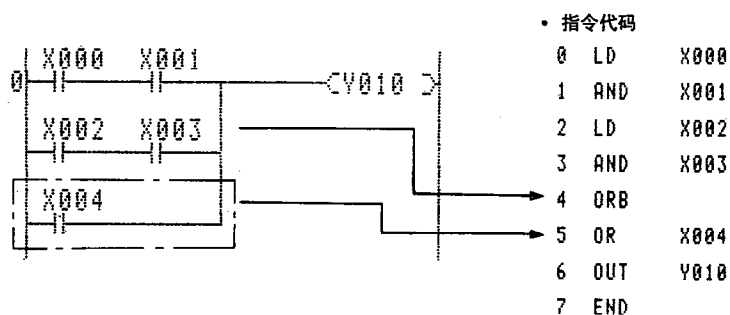
功能

ANB

- (1) 此指令执行电路块A与电路块B的与操作,并将它用作操作结果。
- (2) ANB不是触点的指令符而是连接的指令符号。
- (3) ANB最多连续可写的数目根据下面的情况决定:
 - 对A□ACPU: 15个指令 (16个块)
 - 对非A□ACPU: 7个指令 (8个块)若连续写ANB超过此数目, PLC将不能执行正常操作。

ORB

- (1) 此指令执行电路块A与电路块B的或操作, 并将它用作操作结果。
- (2) ORB执行有两个或更多触点的电路块的并联连接。对于只有一个触点的电路块的并联连接, 可用OR或ORI, 不需ORB (见下图)

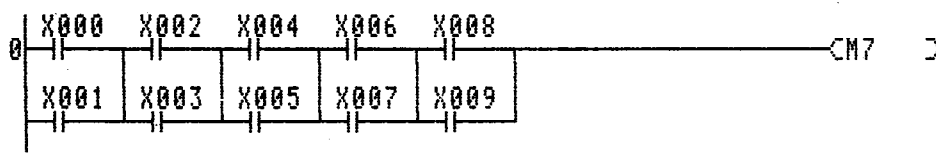


- (3) ORB不是触点的指令符而是连接的指令符。
- (4) ORB最多可连续写的数目根据下面的情况决定:
 对A□ACPU: 15个指令 (16个块)
 对非A□ACPU: 7个指令 (8个块)
 若连续写ORB超过此数目, PLC将不能执行正常操作。

编程举例

ANB

当电路块连续串联时, 程序的指令代码有两种形式. 但在执行时, 按照指令代码举例1处理。



• 指令代码1

```

0 LD X000
1 OR X001
2 LD X002
3 OR X003
4 ANB
5 LD X004
6 OR X005
7 ANB
8 LD X006
9 OR X007
10 ANB
11 LD X008
12 OR X009
13 ANB
14 OUT M7
15 END
  
```

• 指令代码2

```

0 LD X000
1 OR X001
2 LD X002
3 OR X003
4 LD X004
5 OR X005
6 LD X006
7 OR X007
8 LD X008
9 OR X009
10 ANB
11 ANB
12 ANB
13 ANB
14 OUT M7
15 END
  
```



所用的ORB指令的次数不加限制



若ORB指令连续写入超过下述的次数, PLC将不能执行正常操作

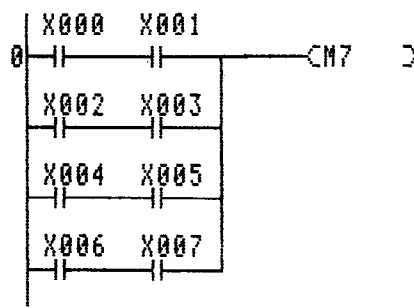
对于AnA(-F) 和 AnU: 15次(16个电路块)

对于除AnA(-F)和AnU外的CPU:

7次(8个电路块)

ORB

当电路块连续并联时,程序的指令代码有两种形式.但在执行时,按照指令代码举例1处理。



指令代码1

```

0 LD      X000
1 AND     X001
2 LD      X002
3 AND     X003
4 ORB
5 LD      X004
6 AND     X005
7 ORB
8 LD      X006
9 AND     X007
10 ORB
11 OUT    M7
12 END

```



所用的ORB指令的次数不加限制

• 指令代码2

```

0 LD      X000
1 AND     X001
2 LD      X002
3 AND     X003
4 LD      X004
5 AND     X005
6 LD      X006
7 AND     X007
8 ORB
9 ORB
10 ORB
11 OUT    M7
12 END

```



若ORB指令连续写入超过下述的次数,PLC将不能执行正常操作

对于AnA(-F) 和 AnU: 15次(16个电路块)

对于除AnA(-F)和AnU外的CPU:

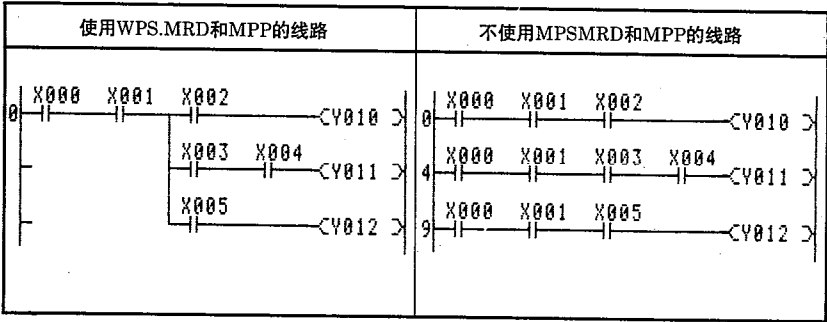
7次(8个电路块)

MPP

- (1) 读出由MPS指令存储的操作结果. 从下一步操作开始, 用此操作结果来继续操作。
- (2) 清除由MPS指令存储的操作结果。

要点

(1) 使用MPS, MRD和MPP, 与不使用这些指令的电路有不同, 如下所述:

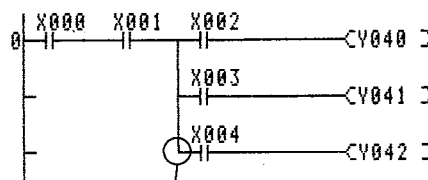


要点

(2) MPS指令和MPP指令的使用次数须一致, 若不一致则会产生以下问题:

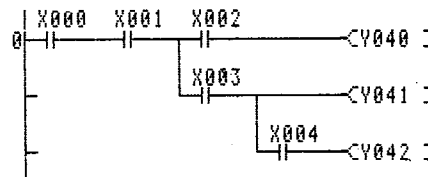
① 如果使用MPS指令的次数大于MPP指令, PLC按照改变了的电路执行。

改变以前



MPP改变为NOP

改变之后



• 指令代码

```

0 LD      X000
1 AND     X001
2 MPS
3 AND     X002
4 OUT     Y040
5 MRD
6 AND     X003
7 OUT     Y041
8 MPP
9 AND     X004
10 OUT    Y042
11 END

```

• 指令代码

```

0 LD      X000
1 AND     X001
2 MPS
3 AND     X002
4 OUT     Y040
5 MRD
6 AND     X003
7 OUT     Y041
8 NOP
9 AND     X004
10 OUT    Y042
11 END

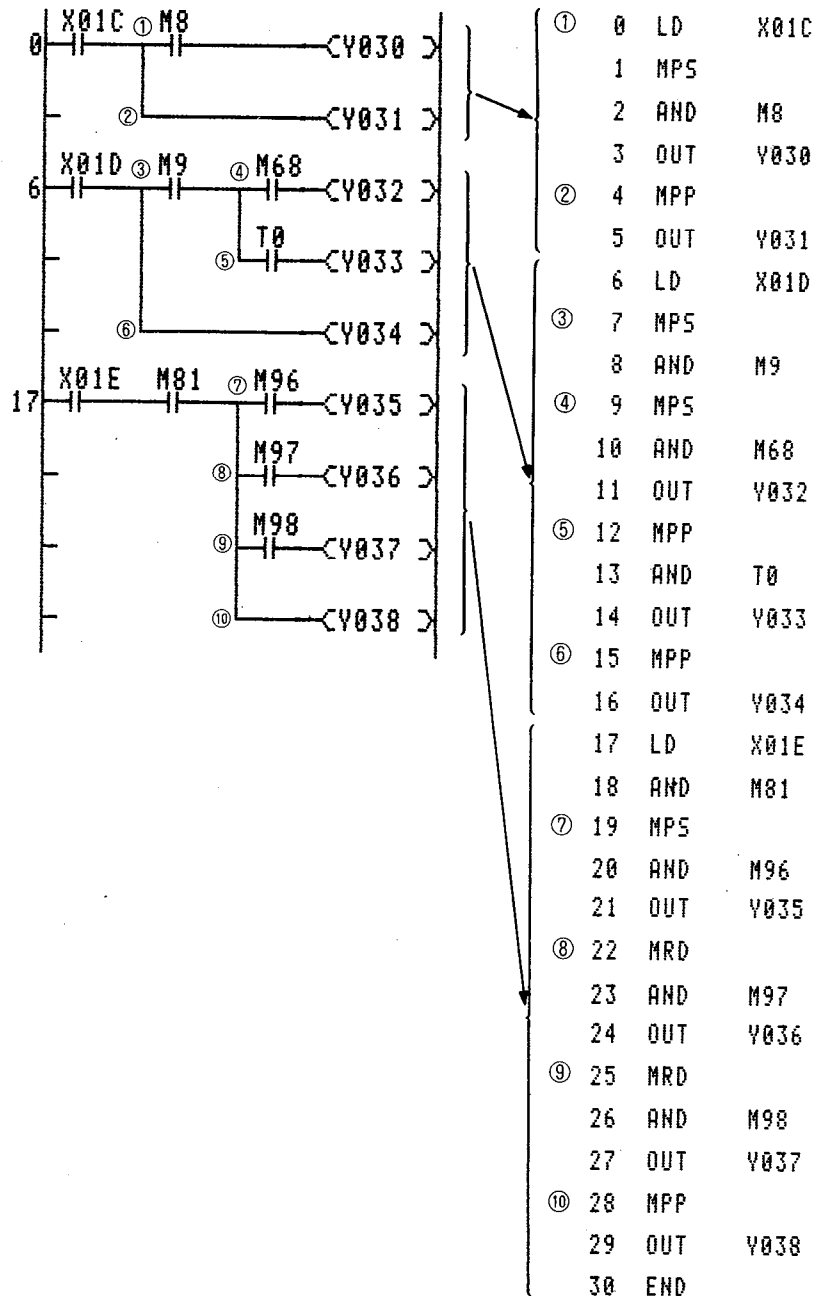
```

② 如果使用MPS指令的次数大于MPP指令, 会导致电路出错, PLC不能正确执行操作。

编程举例

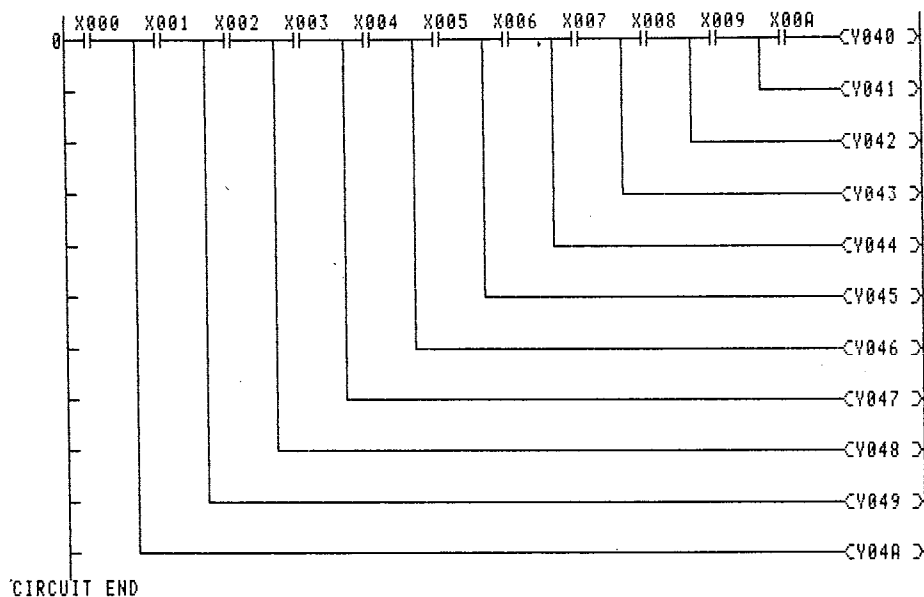
MPS, MRD, MPP

(1) 采用MPS, MRD, MPP编程



(2) 用MPS和MPP指令的打印举例

○打印出的电路



○打印出的指令清单

0	LD	X000	22	MPP	
1	MPS		23	OUT	Y041
2	AND	X001	24	MPP	
3	MPS		25	OUT	Y042
4	AND	X002	26	MPP	
5	MPS		27	OUT	Y043
6	AND	X003	28	MPP	
7	MPS		29	OUT	Y044
8	AND	X004	30	MPP	
9	MPS		31	OUT	Y045
10	AND	X005	32	MPP	
11	MPS		33	OUT	Y046
12	AND	X006	34	MPP	
13	MPS		35	OUT	Y047
14	AND	X007	36	MPP	
15	MPS		37	OUT	Y048
16	AND	X008	38	MPP	
17	MPS		39	OUT	Y049
18	AND	X009	40	MPP	
19	MPS		41	OUT	Y04A
20	AND	X00A	42	END	
21	OUT	Y040			

[illegible]

5.3 输出指令

适用
CPU

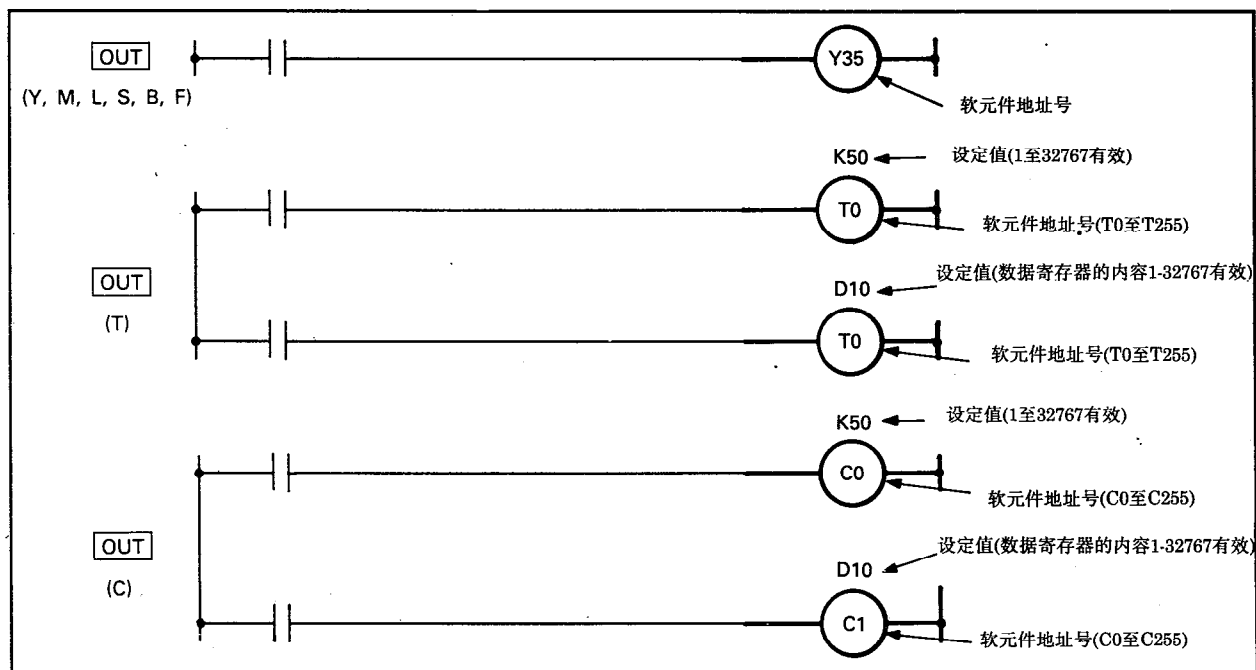
所有型号的CPU组件

5.3.1 位软元件, 定时器, 计数器输出 (OUT)

	可用软元件																				数字规格	变址	地址	出错标志		
	位软元件							字(16位)软元件								常数		指针		数据级						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I			N			
位元件			○	○	○	○	○	○																		
	元件							○																		
	设定值									○ ²							○ ²									
	元件								○																	
	设定值									○ ²							○ ²									
*1: 使用A□ACPU时,才可使用变址 *2: 在A□ACPU中使用扩展定时器或扩展计数器时,参看3.8.3节.																										

*1: 使用A□ACPU时,才可使用变址

*2: 在A□ACPU中使用扩展定时器或扩展计数器时,参看3.8.3节。



功能

OUT (Y, M, L, S, B, F)

此指令输出位于OUT指令前面的元件的操作结果。

指令	操作结果		
	线圈	触点	
		常开触点	常闭触点
OFF	OFF	断开	接通
ON	ON	接通	断开

要点

- (1) 当F(信号报警器)接通时,在CPU模块上的LED显示窗和ERROR LED点亮,所有接通的信号报警器的点数存储于特殊寄存器中。详情请参考ACPU编程手册(基础篇)。
- (2) 若用OUT指令来接通信号报警器时,其线圈的状态并不与LED显示窗的内容一致。为避免这种情况,使用SET指令来接通信号报警器。
若使用OUT指令来使信号报警器接通时,当OUT指令前面的操作结果变为OFF时,信号报警器线圈也变为OFF。但是,在CPU模块上的LED显示窗中的内容和ERROR LED(出错指示灯)以及特殊寄存器的内容并不改变。详情请参考ACPU编程手册(基础篇)。

备 注

在OUT指令中用了下述任一软元件时,步数为3:

- 特殊辅助继电器(M)
- 信号报警器(F)

OUT (T)

- (1) 当OUT指令前操作结果为ON时, 定时器的线圈接通, 并一直计数到设定值为止。当定时器到达其设定值后 (计数值 $\geq$ 设定值), 触点状态如下所示:

常开触点	接通
常闭触点	断开

- (2) 当OUT指令前的操作结果从ON变为OFF时, 产生下列情况:

定时器类型	定时器线圈	定时器当前值	设定值到达之前		设定值到达之后	
			常开触点	常闭触点	常开触点	常闭触点
100ms定时器	OFF	0	断开	接通	断开	接通
10ms定时器						
带保持的100ms定时器	OFF	保持当前值	断开	接通	接通	断开

- (3) 当定时器到达其设定值后, 有保持功能的定时器的触点状态一直不变, 直到执行RST指令才复位。
- (4) 在A□ACPU中使用T256---T2047时, 按照3. 8. 3节所述指定设定值。
- (5) 负数 (-32768 ~ -1) 不能作为设定值。
- (6) 当设定值为0时, 认为设定值是无限大, 因此定时器永远不会到达其设定值。
- (7) 关于定时器的计数处理, 参考ACPU编程手册 (基础篇)。

OUT (C)

- (1) 当OUT指令前的操作结果从OFF变为ON时, 计数器当前值 (计数值) 加1, 当计数器计数值到达其设定值时 (计数值=设定值), 触点状态如下所示:

常开触点	接通
常闭触点	断开

- (2) 当OUT指令前的操作结果保持接通, 不执行计数 (不需要将计数输入转换为一个脉冲)。
- (3) 计数器计数到之后, 计数值和触点状态不变, 直到执行RST指令才复位。
- (4) 若在A□ACPU中使用C256 ~ C1023, 参照3. 8. 3节作设定值的设定。
- (5) 负数 (-32768 ~ -1) 不能作为设定值, 当设定值为零时, 执行与1相同的处理。
- (6) 关于计数器的计数处理, 参照ACPU编程手册 (基础篇)。

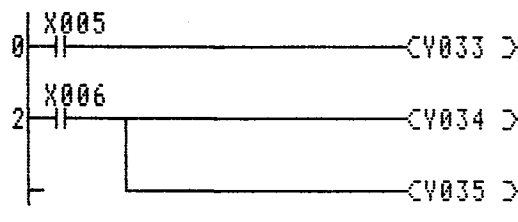
执行条件

不管OUT指令前面的操作结果如何,每次扫描都要执行本指令。

编程举例

OUT

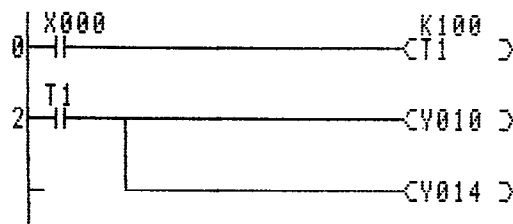
1) 开关输出组件上输出点的程序



• 指令代码

0	LD	X005
1	OUT	Y033
2	LD	X006
3	OUT	Y034
4	OUT	Y035
5	END	

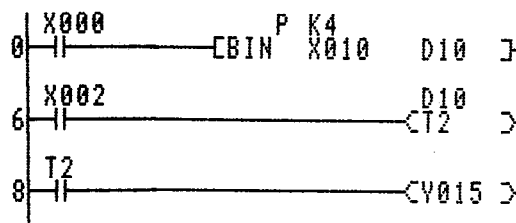
2) 在X0接通10秒后使Y10和Y14得电



• 指令代码

0	LD	X000
1	OUT	T1 K100
2	LD	T1
3	OUT	Y010
4	OUT	Y014
5	END	

3) 用X10至X1F的BCD数据作为定时器的设定值



X10至X1F的数据转换为二进制(BIN),并存入D10

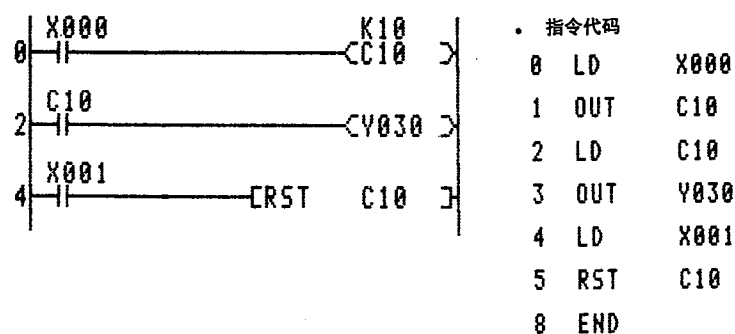
当X2接通,存入D10的数据用作T2的设定值

当T2到达所设定的时间值,Y15得电

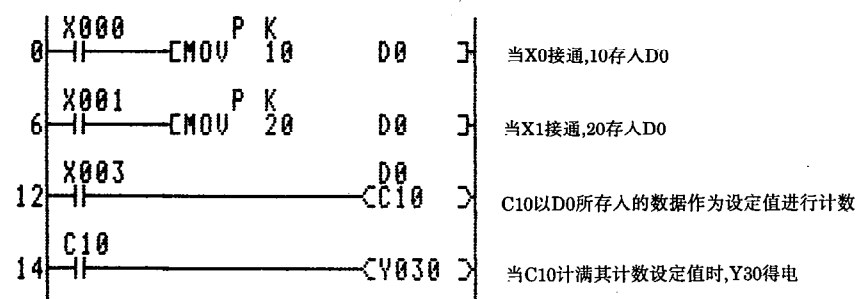
• 指令代码

0	LD	X000
1	BINP	K4X010 D10
6	LD	X002
7	OUT	T2 D10
8	LD	T2
9	OUT	Y015

4) 当X0接通10次后, Y30得电; 当X1接通, Y30失电。



5) C10的设定值在X0接通时变为10, 而在X1接通时却变为20。



• 指令代码

```

0 LD      X000
1 MOVP    K10      D0
6 LD      X001
7 MOVP    K20      D0
12 LD     X003
13 OUT    C10      D0
14 LD     C10
15 OUT    Y030
16 END

```

[illegible]

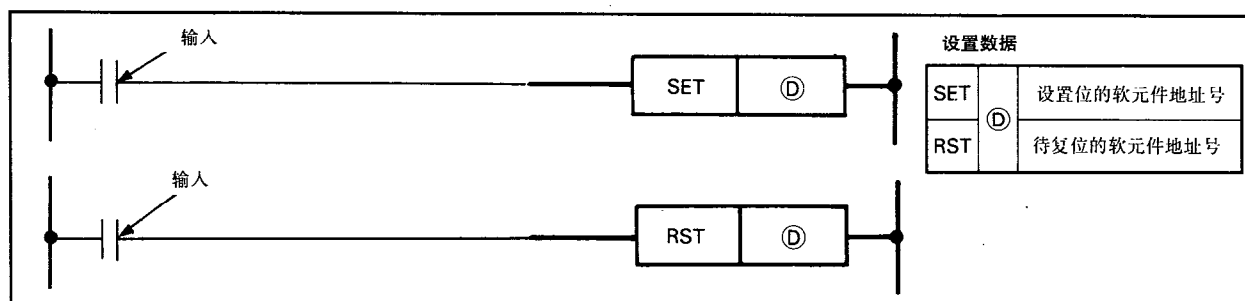
5.3.2 位软元件置位与复位 (SET, RESET)

适用
CPU

所有型号的CPU组件

		可用之软元件																			数字规格	变址	标志	出错标志
		位软元件							字(16位)软元件								常数		指针					
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P			I	N
SET	①		○	○	○	○	○	○																
RST			○	○	○	○	○	○	○	○	○	○	○	○	○									

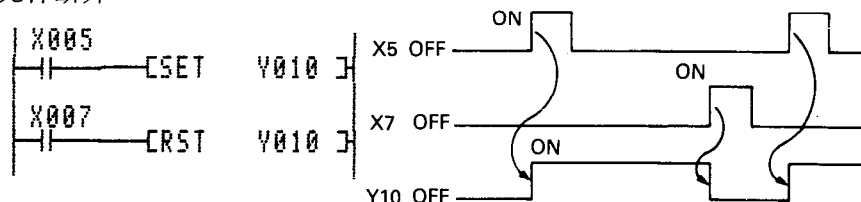
*1: 在使用A□ACPU时的步数,参见3.8.1节



功能

SET

- (1) 当SET输入接通时, 所指定的软元件接通.
- (2) 即使SET输入断开时, 所接通的软元件其状态仍然保持, 用RST指令可将该软元件断开.



- (3) 当SET指令输入断开, 该软元件的状态不会改变.

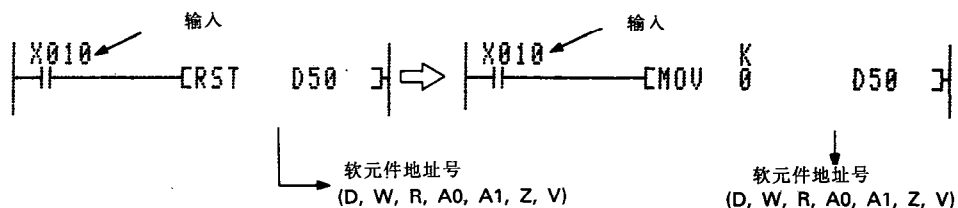
RST

- (1) 当RST输入接通, 所指定的软元件其变化如下所述:

软元件	状态
Y, M, L, S, B, F	线圈失电, 触点断开
T, C	当前值置0, 线圈失电, 触点断开
D, W, R, A0, A1, Z, V	内容置为0

- (2) 当RST输入断开, 软元件的状态不会变化.

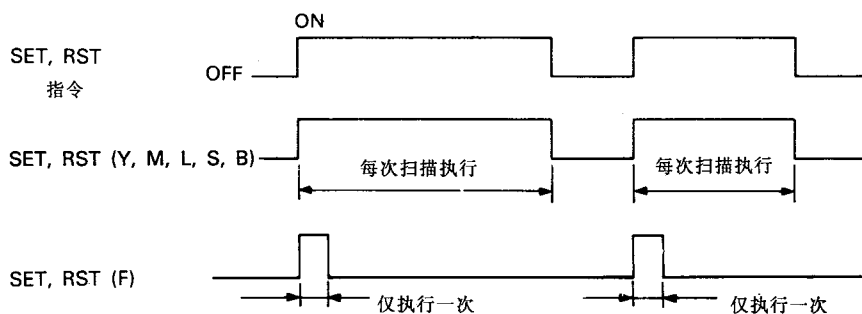
(3) RST(D, W, R, A0, A1, Z, V) 的功能与下述电路的功能相同



若信号报警器接通/断开, CPU组件上的LED显示窗及ERROR(出错)LED所显示的内容以及特殊寄存器的内容均发生变化, 详见ACPU编程手册(基础篇)。

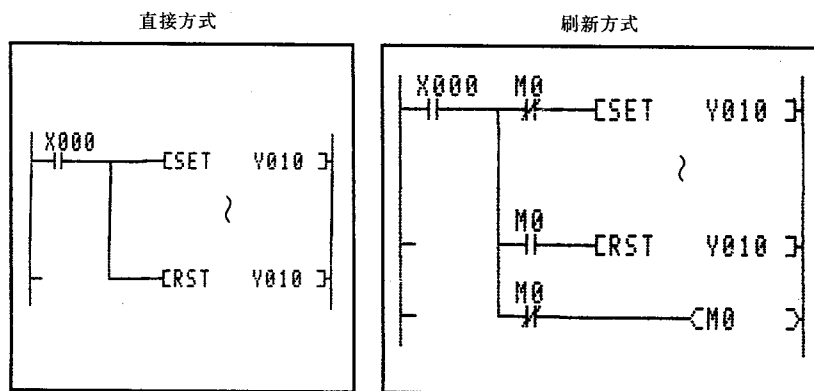
执行条件

(1) SET RST指令按以下条件执行



(2) SET RST指令

在刷新方式下, SET/RST指令不可用在在一个扫描周期输出一个脉冲信号的程序. 若有此要求, 输出(Y)必须改为直接方式, 或者程序必须加以修改, 使软元件每次扫描时进行接通/断开之切换(如下所示)。



备注

在用以下软元件时, 步数为3:

SET指令 特殊继电器(M)

通信继电器(B)

信号报警器(F)

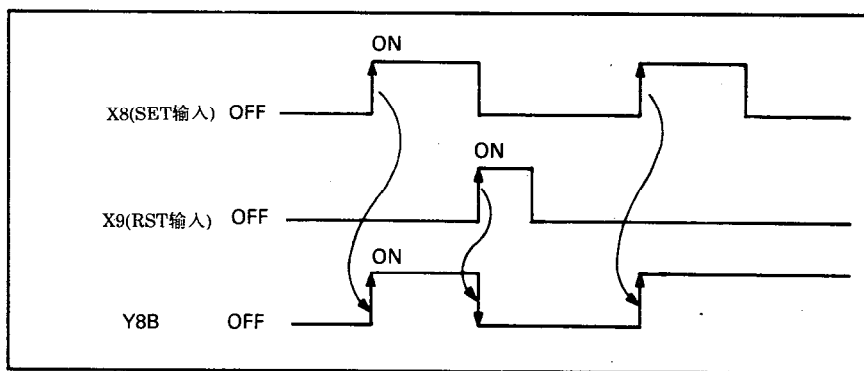
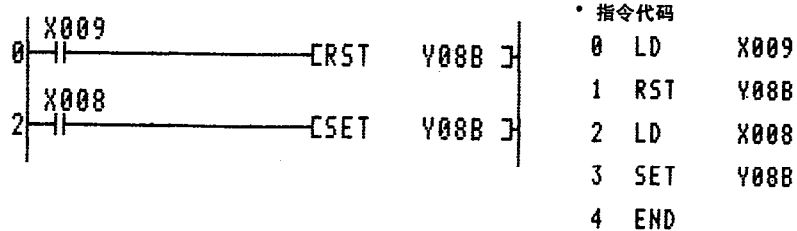
RST指令 特殊继电器(M)

字软元件(全部)

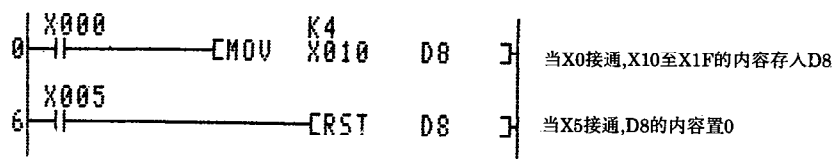
编程举例

SET, RST

1) 当X8接通, Y8B置位(接通), 而X9接通, Y8B复位(断开)。



2) 设置数据寄存器的内容为0



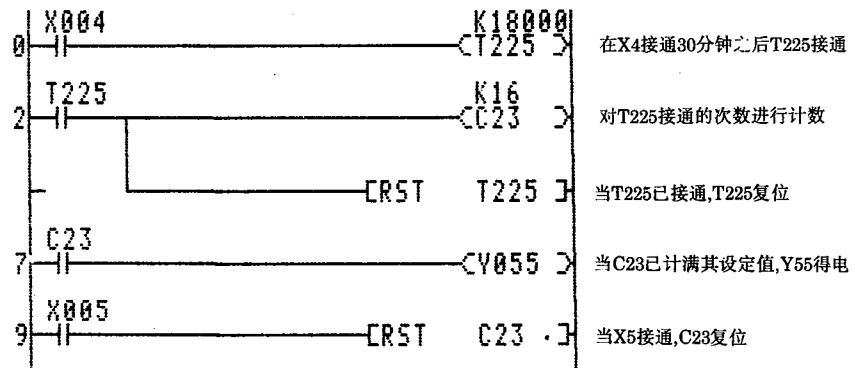
• 指令代码

```

0 LD X000
1 MOV K4X010 D8
6 LD X005
7 RST D8
10 END

```

3) 将100ms带保持的定时器及计数器复位



• 指令代码

0	LD	X004	
1	OUT	T225	K18000
2	LD	T225	
3	OUT	C23	K16
4	RST	T225	
7	LD	C23	
8	OUT	Y055	
9	LD	X005	
10	RST	C23	
13	END		

5. 顺控指令

MELSEC-A

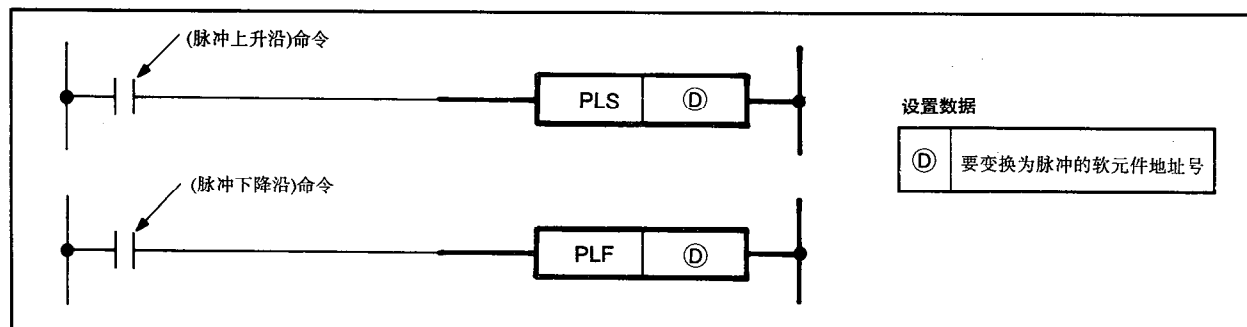
5.3.3 沿口触发微分指令 (PLS, PLF)

适用
CPU

所有型号的CPU组件

	可用之软元件																			D 数字规格	变址	特殊 进位标志	出错标志
	位软元件							字(16位)软元件								常数		指针					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N		
①		○	○	○	○	○	○																

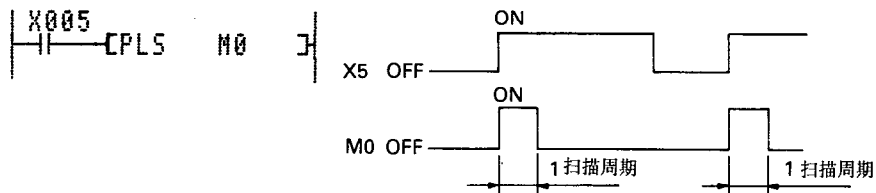
*1: 在使用A□ACPU时的步数,参见3.8.1节



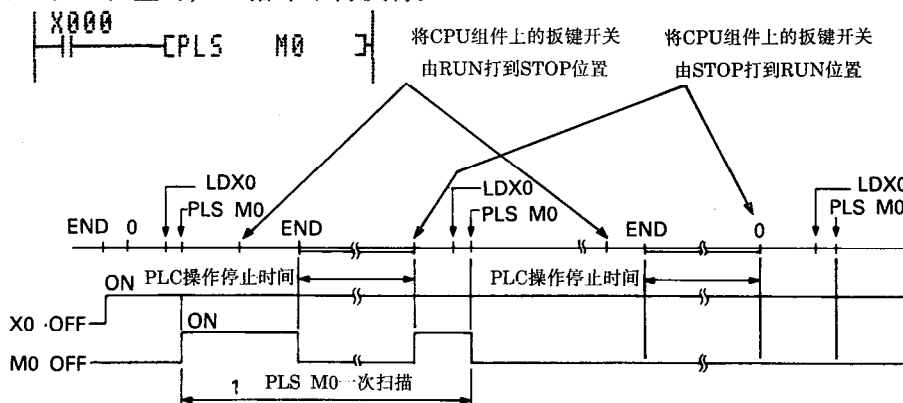
功能

PLS

- (1) 当PLS指令执行命令从断开到接通时, PLS指令使指定的软元件接通一个扫描周期。同样, 当该软元件处于接通状态时, PLS指令使之断开一个扫描周期。



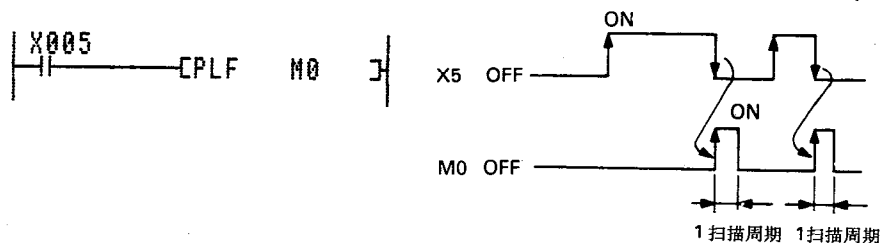
- (2) 如果产生脉冲的指令已经接通, RUN扳键开关从RUN打到STOP位置, 然后从新回到RUN位置时, PLS指令不再执行。



- (3) 当PLS指令使用锁存继电器(L)时, 电源恢复后原先的数据恢复输出。

PLF

- (1) 当PLF指令执行命令从断开到接通时, PLF指令使指定的软元件接通一个扫描周期。同样, 当该软元件处于接通状态时, PLF指令使之断开一个扫描周期。



- (2) 如果产生脉冲的指令已经断开, RUN扳键开关从RUN打到STOP位置, 然后从新回到RUN位置时, PLF指令不再执行。

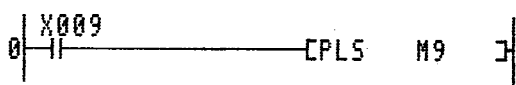
要点

如果由于使用CJ指令而跳过PLS或PLF指令时, 由(D)指定的软元件有时会接通大于一个扫描周期。

编程举例

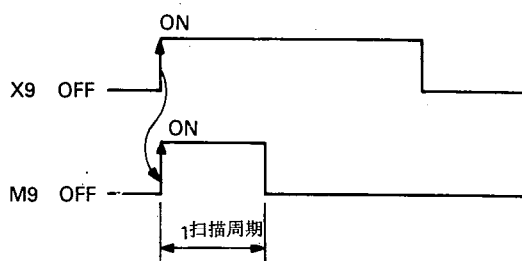
PLS

当M9接通, 执行PLS指令



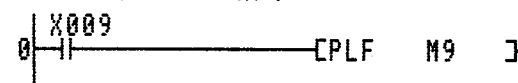
• 指令代码

```
0 LD X009
1 PLS M9
4 END
```



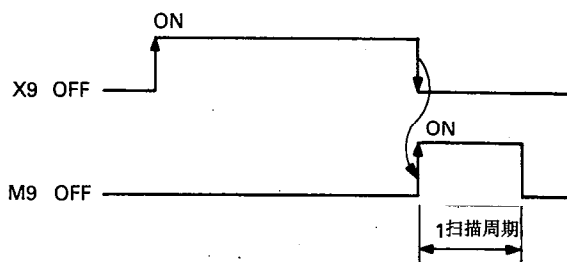
PLF

当M9接通, 执行PLF指令



• 指令代码

```
0 LD X009
1 PLF M9
4 END
```



5. 顺控指令

5.3.4 位软元件输出取反 (CHK)

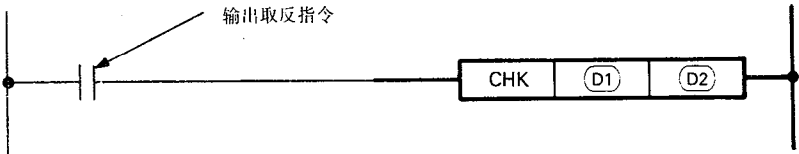
Applicable CPU	A1S	An	AnN AnN-F	A3H A3M	A3V	AnA AnA-F	AnU	A0J2H	A2C	A52G	A73	A3N board
	△*	×	△*	×	×	×	×	△*	×	×	△*	△*
适用	* 对刷新方式有效											

CHK指令的功能随I/O控制方式的变化而变化。如下表所示:

CPU	I/O 控制方式	
	直接方式	刷新方式 (当输入和输出任一或都处于刷新方式)
An	故障检查	—————
AnN, AnN-F, A1S, A0J2H, A73, A3N board	故障检查	位软元件输出取反
A3H, A3M	故障检查	故障检查
A3V, AnA, AnA-F, A2C, A52G, AnU	—————	故障检查

关于故障检查, 参考7.10.2节

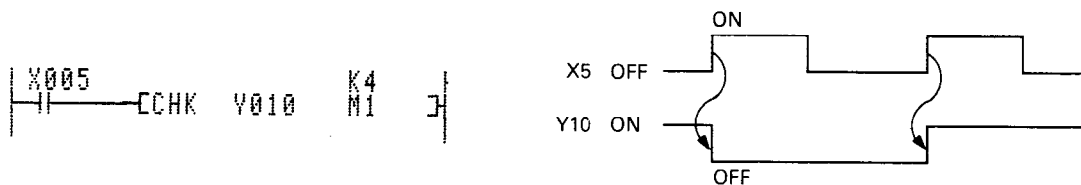
	可用之软元件																			数字规格	变址	进位标志 M9012	出错标志 (M9010, M9011)
	位软元件						字(16位)软元件								常数		指针		嵌套级				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H					
(D1)		○	○	○	○	○	○																
(D2)		△	△	△	△	△	△	△	△	△	△	△	△	△	△	△					K1 to K4		
* 1 (D2)中使用的元件是虚设数据, 在程序处理中不起作用																							



设定数据	
(D1)	所要求的软元件地址号
(D2)	虚设的数据 标有-的任何软元件地址号

功能

- (1) 在输出取反命令的上升沿对软元件D1的输出状态取反。
- (2) D2是一个虚设数据,可把任何带有△标志的软元件设为D2。如果设定一个位软元件D2,则其数字规格须设定为K1到K4。因为此数字规格值是虚设数据,可设定为任意值。



D2指定的软元件可任意用作其它用途。

- (3) CHK指令只可在刷新方式下执行。
- (4) 输出取反命令的ON/OFF时间必须等于或大于1个扫描周期。

编程举例

CHK

下面程序是当X9接通时,Y10的输出状态取反。

<pre> 0 X009 -----[CHK Y010 D0]----- </pre>	• 指令代码	<pre> 0 LD X009 1 CHK Y010 D0 6 END </pre>
---	---------------	--

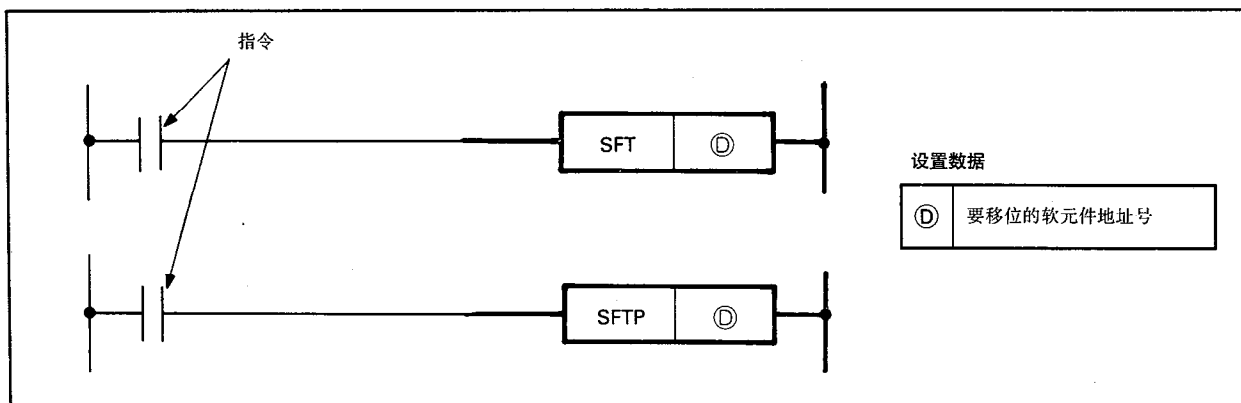
5.4 移位指令

5.4.1 位软元件移位 (SFT, SFTP)

适用 CPU	所有型号的CPU组件
-----------	------------

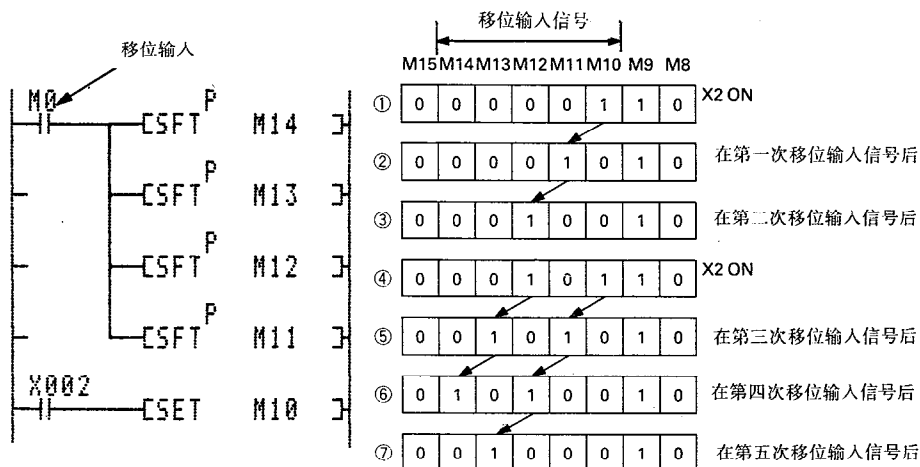
	可用之软元件																			数字规格	变址	标志	出错标志		
	位软元件							字(16位)软元件								常数		指针						嵌套级	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P			I	N		
①		○	○	○	○	○	○																△	M9012	(M9010, M9011)

※ 1 在使用A□ACPU时的步数,参见3.8.1节



功能

- (1) 此指令将某一地址号的软元件(定义为D-1)的ON/OFF状态移位到指定为D的软元件,并使最低地址号的软元件复位。
- (2) 用SET指令使要移位的首地址软元件接通。
- (3) 当连续使用SFT, SFTP指令时,对地址号较大的软元件应先编程。(见下面)

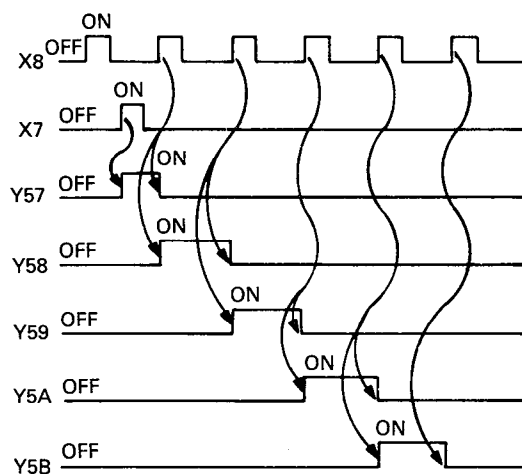
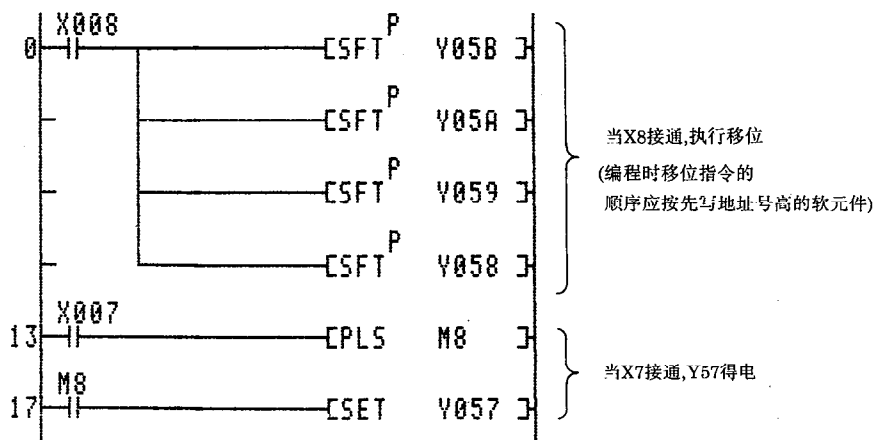


* : 对M8至M15,1表示ON 0表示OFF

编程举例

SFT

(1) 当X8接通时, Y57-Y5B移位的程序



• 指令代码

0	LD	X008
1	SFTP	Y05B
4	SFTP	Y05A
7	SFTP	Y059
10	SFTP	Y058
13	LD	X007
14	PLS	M8
17	LD	M8
18	SET	Y057
19	END	

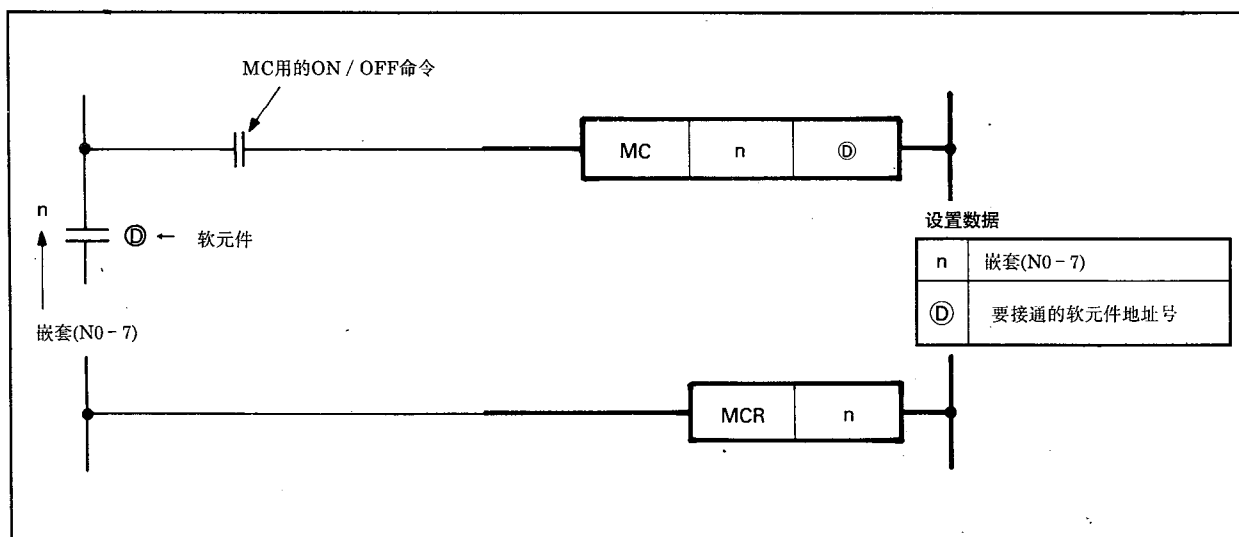
5.5 主控指令

5.5.1 主控置位, 复位 (MC, MCR)

适用 CPU	所有型号的CPU组件
-----------	------------

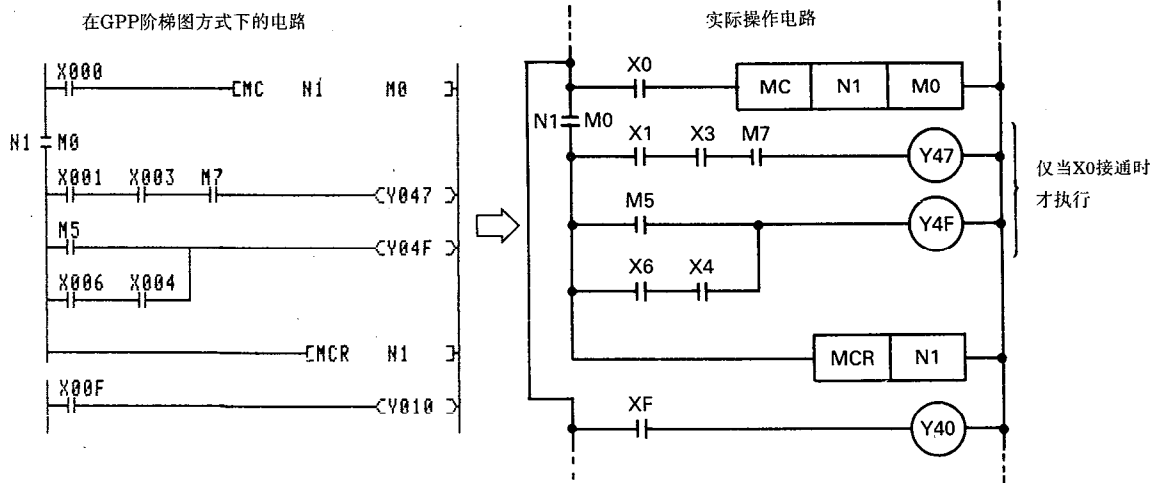
	可用之软元件																				数字规格	变址	地址标志	出错标志		
	位软元件							字(16位)软元件								常数		指针		嵌套级						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P						I	N
n																					○	△				
①		○	○	○	○	○	○																			

* 1 MC指令占5步,MCR指令占3步 在使用A□ACPU时的步数,参见3.8.1节



功能

MC指令通过切换公共母线的通断来使顺控程序有效执行程序。下图所示为当使用MC指令时的电路:



备注

当在GPP上阶梯图模式中编程时, 不必在母线上输入触点。那些触点通过转换自动产生。

功能

MC

- (1) MC是主控起始指令。当MC指令的ON/OFF命令为ON时,从MC到MCR之间程序的操作结果保持不变。
- (2) 即使MC指令的ON/OFF命令为OFF,仍然对MC和MCR之间的指令进行扫描。因此扫描时间并不缩短。当执行MC指令的ON/OFF命令为OFF时,MC和MCR之间操作结果如下所示:

100ms和10ms定时器	计数值变为0,线圈和触点变为OFF
100MS带保持定时器,计数器	线圈变为OFF,计数值和触点保持当前状态
OUT指令用到的软元件	全变为OFF
SET,RST,SFT指令中用到的软元件	保持当前状态

要点

在有MC指令的电路中,如果包含那些在其前面无须触点指令的指令(FOR-NEXT, EI, DI等),PLC不管MC指令的ON/OFF命令处于何种状态都执行这些指令。

- (3) 用改变①软元件的办法,MC指令可以重复使用相同的嵌套级数。
- (4) 若MC指令为ON,①指定的软元件的线圈置为ON。如果OUT指令两次使用了同一软元件,会认为是双线圈。为避免这种情况,不要在别处再使用①指定过的软元件。

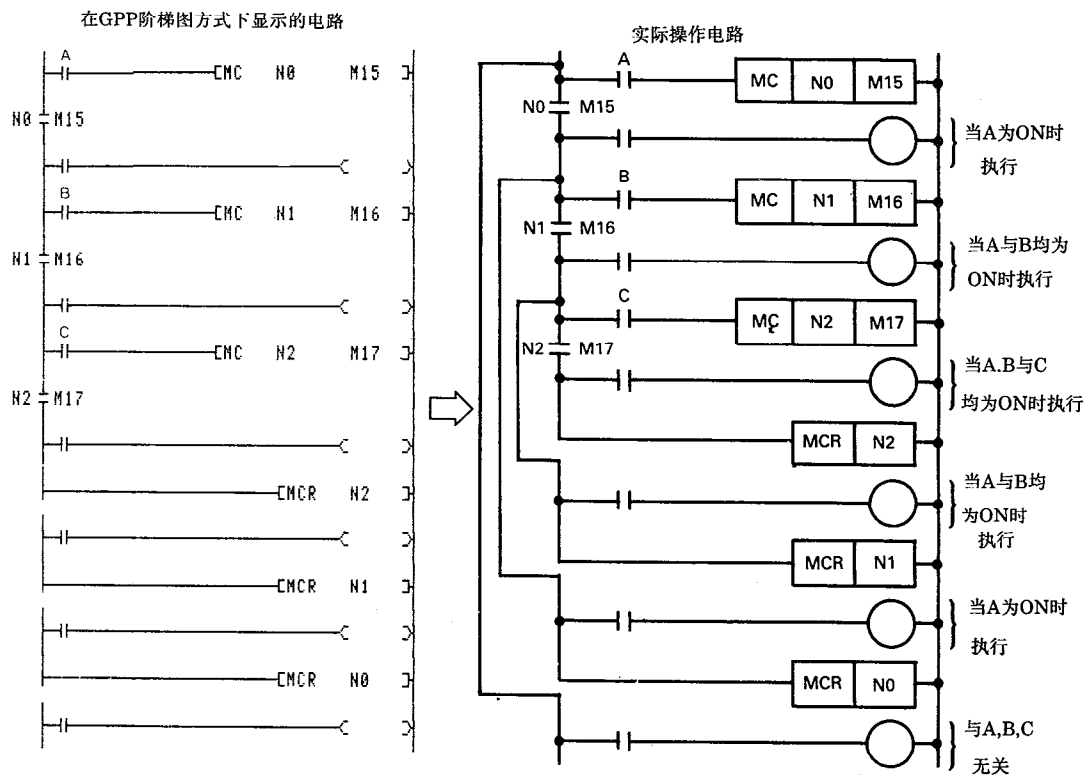
MCR

- (1) MCR是主控复位指令,指示主控范围的结束。
- (2) 不要在MCR指令前使用触点指令。

主控指令利用嵌套级号来使用。每个主控指令的范围由嵌套级号来标明。嵌套级号的范围为N0-N7。

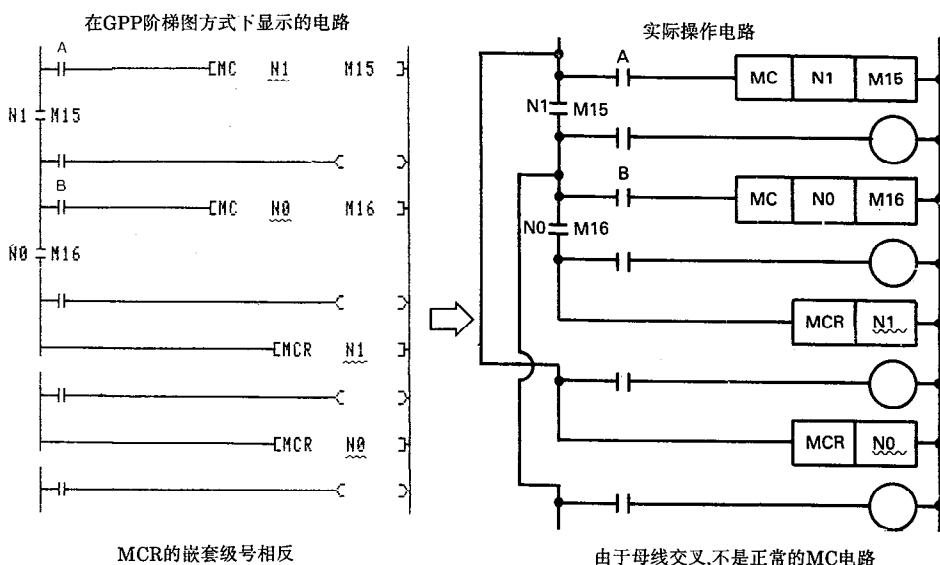
使用嵌套,可以对严格按照顺序条件执行的电路进行编程。

下图是一个使用嵌套的电路举例:

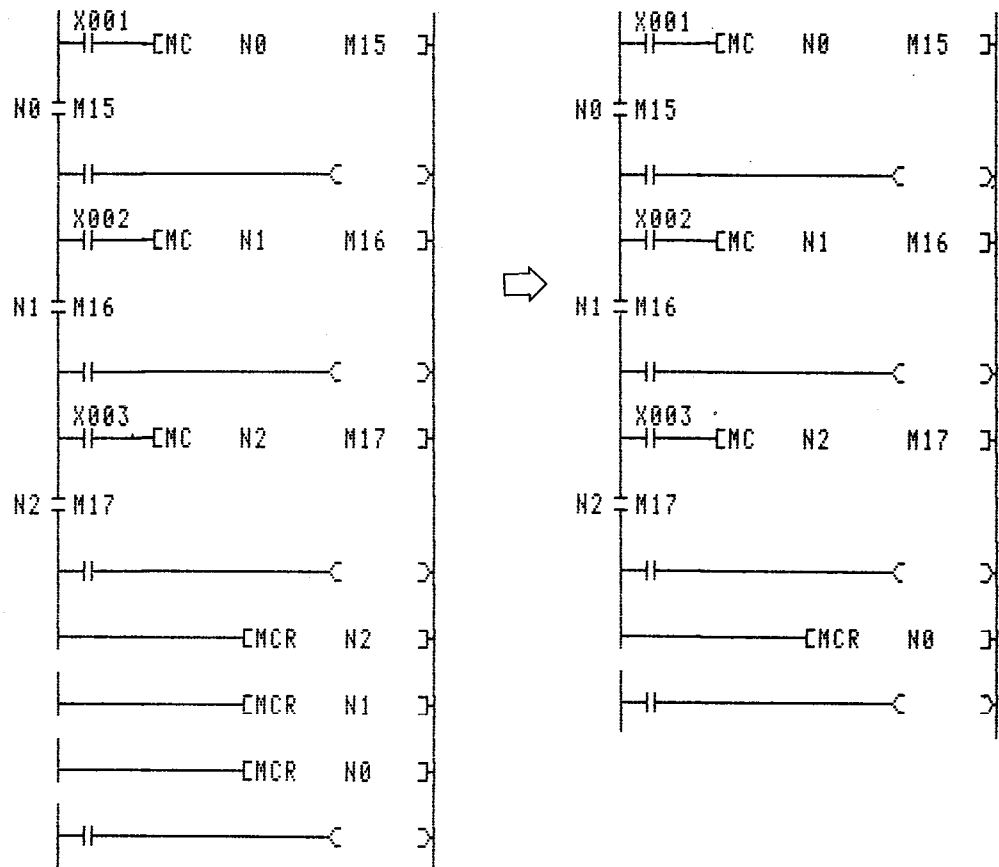


使用嵌套时应注意下面几点:

- (1) 嵌套最多可用8级,从N0到N7,MC嵌套级号从小级号开始,而MCR从最大的级号开始。如果嵌套级号用反了,不能构成正确的嵌套,并且PLC不能执行正确的操作。



- (2) 如果所有嵌套均在同一地方使用MCR指令, 只要使用最小的嵌套级号一次便可结束所有的MC指令。



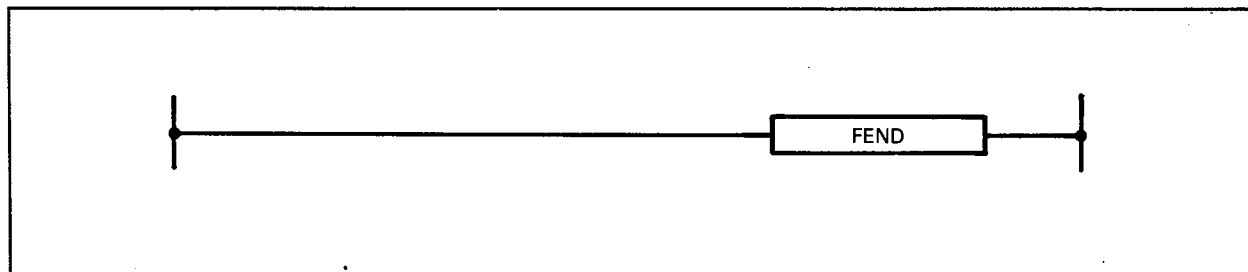
5.6 结束指令

5.6.1 主程序结束指令 (FEND)

适用
CPU

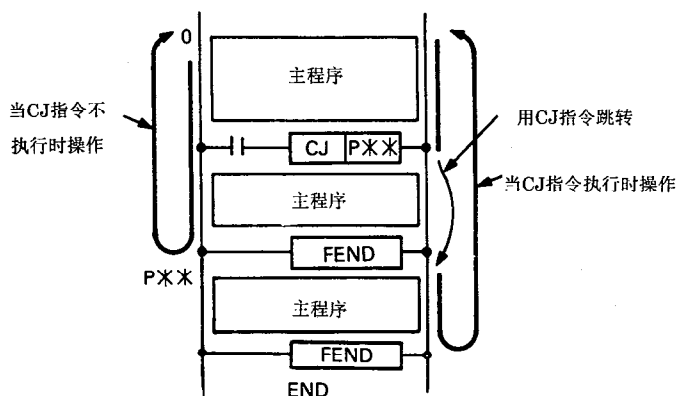
所有型号的CPU组件

可用之软元件																					数字规格	变址	进位标志	出错标志
位软元件							字(16位)软元件								常数		指针		状态级					
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N			M9012	(M9010, M9011)
																						○		

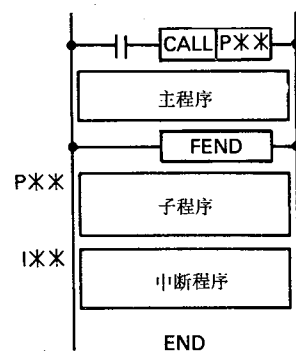


功能

- (1) 结束主程序。
- (2) 当执行FEND指令时, 在执行完END指令后的处理 (如定时器/计时器处理和自诊断) 后, PLC回到第0步再恢复操作。
- (3) 在FEND指令后的顺控程序也可以在GPP上显示 (GPP显示END指令前的所有电路)。



(a) 用CJ指令



(b) 有子程序和中断程序

操作出错

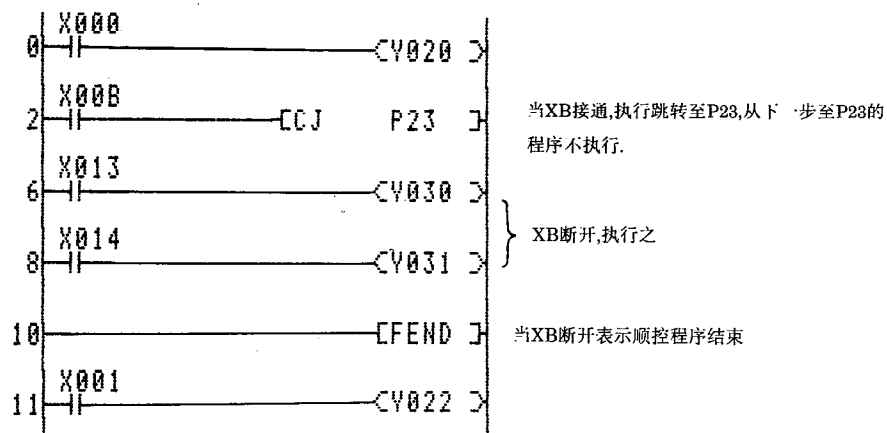
在下面的情况, 会出现操作出错, PLC停止其操作。

- 在执行完CALL (P) 指令后, 在执行RET指令前已执行FEND指令。
- 在执行FOR指令后, 在执行NEXT指令前已执行FEND指令。

编程举例

FEND

1) 用CJ指令的程序



• 指令代码

```

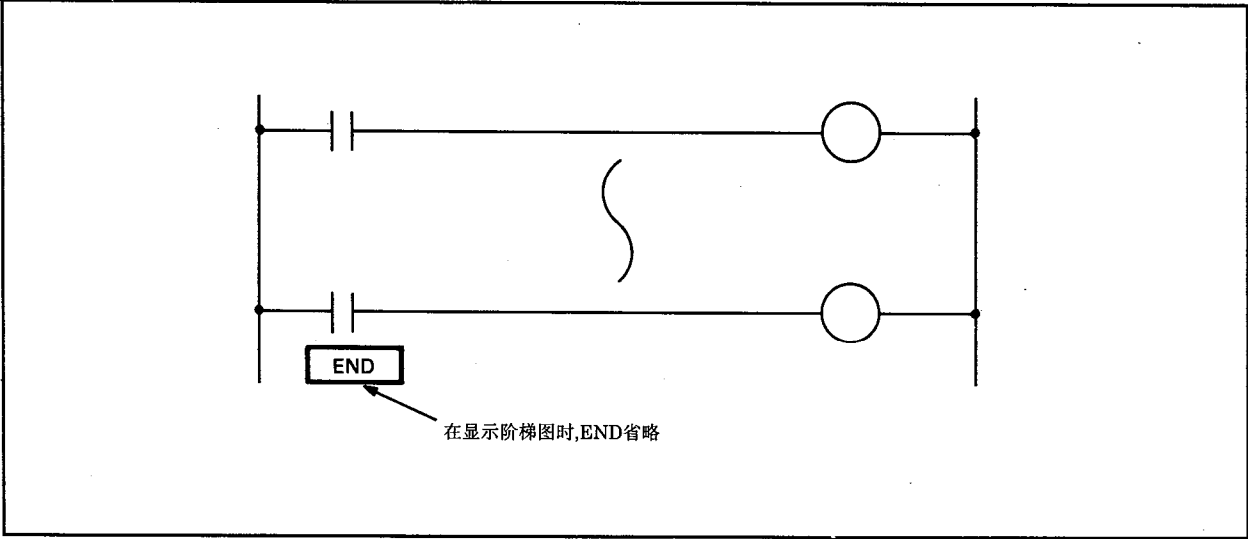
0  LD    X000
1  OUT    Y020
2  LD    X00B
3  CJ     P23
6  LD    X013
7  OUT    Y030
8  LD    X014
9  OUT    Y031
10 FEND
11 P23
12 LD    X001
13 OUT    Y022
14 END

```

5.6.2 顺控程序结束 (END)

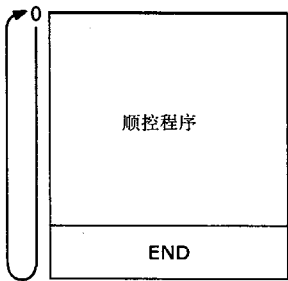
适用 CPU	所有型号的CPU组件
-----------	------------

可用之软元件																					数字规格	变址	进位标志	出错标志																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
位软元件							字(16位)软元件								常数		指针		散转级																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I	N																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												



功能

(1) 这个指令表示程序的结束。在这一步,扫描返回第0步。



- (2) END指令不能用在主顺控程序或副顺控程序的中间。如果在程序中间须进行END处理, 请使用FEND指令。
- (3) 如果程序在GPP上的阶梯图模式中输入, 不必要键入END指令, 通过执行转换可自动输入。

(4) 在主程序, 子程序, 中断程序和副程序中使用END和FEND指令, 如下所示:

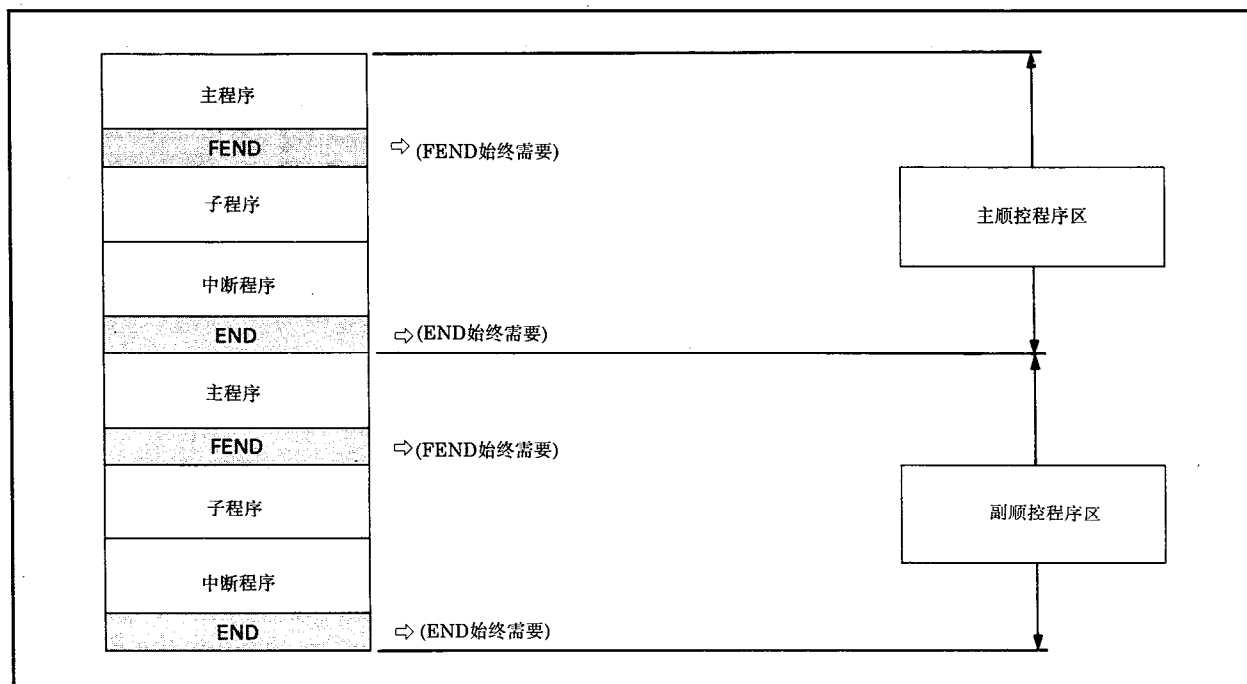


图5.1 END (FEND) 指令的使用

5) 如果在程序中缺少END指令, 会产生操作出错, PLC不会运行。如果在参数中设定了副程序的容量, 若在副程序中缺少END指令, 也会产生操作出错。

操作出错

在下例情况下会产生操作出错, PLC停止操作:

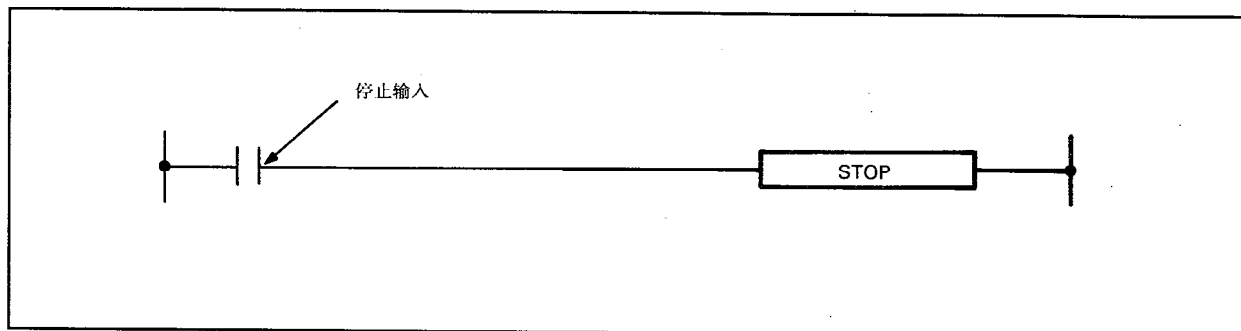
- (1) 程序用CJ, SCJ或JMP指令跳转到END指令后面的步。
- (2) 执行在END指令后面的子程序或中断程序。

5.7 其它指令

5.7.1 顺控程序停止 (STOP)

适用 CPU	所有型号的CPU组件
-----------	------------

可用之软元件																				数字规格	变址	特殊位址	出错标志					
位软元件							字(16位)软元件								常数		指针		兼寄存器									
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I					N				



功能

- (1) 当停止运行的输入变为ON时, 输出Y复位, 并停止PLC的操作。(与运行开关从RUN位置扳到STOP位置的功能相同。)
- (2) 当执行STOP指令时, 特殊寄存器D9015的b8位置为1

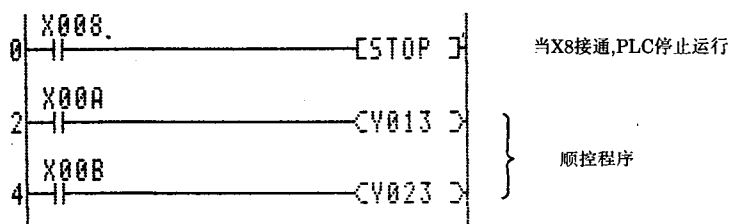
	b15	b14	b13	b12	b11	b10	b9	b8		b0
D9015	0	0	0	0	0	0	0	1		
								置1		

- (3) 在执行完STOP指令后要恢复PLC的操作, 将RUN开关由RUN位置扳到STOP位置, 然后再从STOP位置扳回到RUN位置。
- (4) 当已执行STOP指令时, 即使将RESET (复位) 开关扳到"LATCH CLEAR"位置, 也不执行锁存清除操作。要执行锁存清除, 把RUN开关扳到STOP位置, 然后将RESET开关扳到"LATCH CLEAR"位置。
- (5) 不要在中断程序, 子程序和FOR/NEXT指令中使用STOP指令。若使用了STOP指令, 会产生操作出错。

编程举例

STOP

(1) 当X8接通时停止PLC的操作。



• 指令代码

0	LD	X008
1	STOP	
2	LD	X00A
3	OUT	Y013
4	LD	X00B
5	OUT	Y023
6	END	

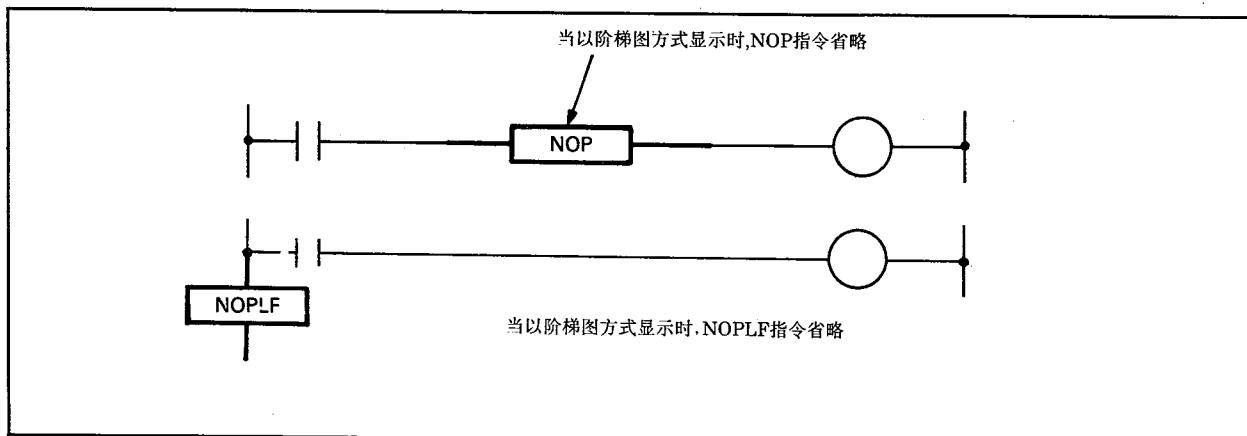
5.7.2 空操作 (NOP, NOPLF)

适用
CPU

所有型号的CPU组件

在GPP上使用SW4GP-GPPA或SW01X-GPPAE系统软件时,可使用NOPLF指令。

有效软元件																				数字规格	变址	进位标志	出错标志
位软元件							字(16位)软元件								常数		指针		寄存器				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H	P	I			N	M9012



功能

NOP

- (1) 这是一个空操作指令,对前面的操作没有影响。
- (2) 在下列情况下使用NOP指令。
 - ① 为顺控程序提供调试空间。
 - ② 删除一条指令而不改变步数。(用NOP重写)
 - ③ 临时删除一条指令。

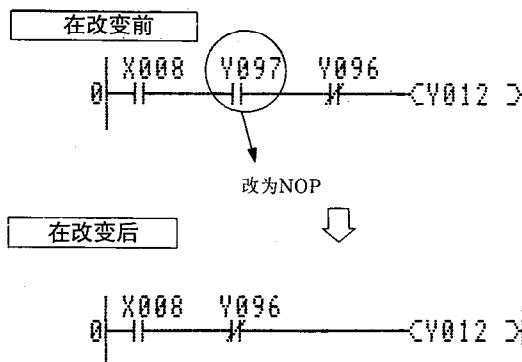
NOPLF

- (1) 这是一个空操作指令,对前面的操作没有影响。
- (2) NOPLF指令是在GPP打印机输出操作中,用于把所希望的某一点设定为一页的结尾。
 - ① 对于打印阶梯图
 - 如果在每个电路块的结尾使用NOPLF指令,将会换页。若NOPLF指令用于阶梯图电路的中间,它将被忽略。
 - 当在GPP阶梯图模式下进行转换时,在阶梯图电路块中使用的NOPLF指令的处理如下:
 - 当增加步数时删除掉
 - 当步数减少时转换成NOP
 - ② 指令表打印
 - 在打印NOPLF后换页。
- (3) 关于GPP打印机输出,参照外部设备的操作手册

编程举例

NOP

(1) 短路触点 (AND, ANI)



• 指令代码

0 LD X008

1 AND Y097

2 ANI Y096

3 OUT Y012

4 END

• 指令代码

0 LD X008

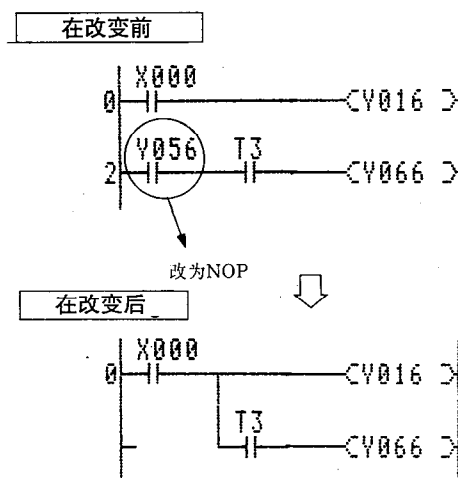
1 NOP

2 ANI Y096

3 OUT Y012

4 END

(2) 短路触点 (LD, LDI): 如果LD或LDI改为NOP时, 电路会完全改变, 因此必须注意。



• 指令代码

0 LD X000

1 OUT Y016

2 LD Y056

3 AND T3

4 OUT Y066

5 END

• 指令代码

0 LD X000

1 OUT Y016

2 NOP

3 AND T3

4 OUT Y066

5 END

• 指令代码

0 LD X000

1 OUT Y016

2 LD Y056

3 AND T3

4 OUT Y066

5 END

• 指令代码

0 LD X000

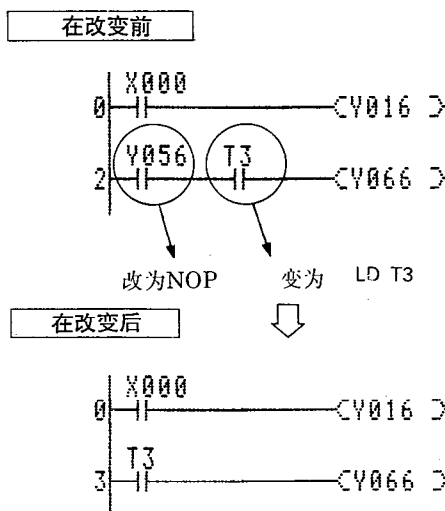
1 OUT Y016

2 NOP

3 LD T3

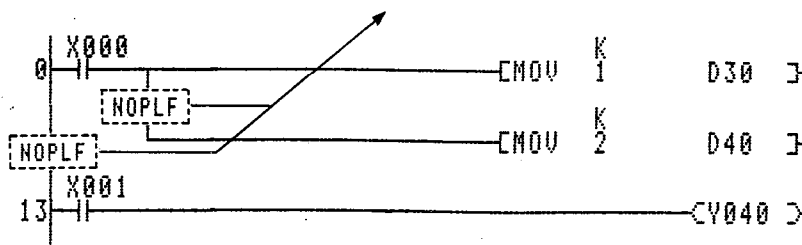
4 OUT Y066

5 END



NOPLF

在阶梯图方式下不显示



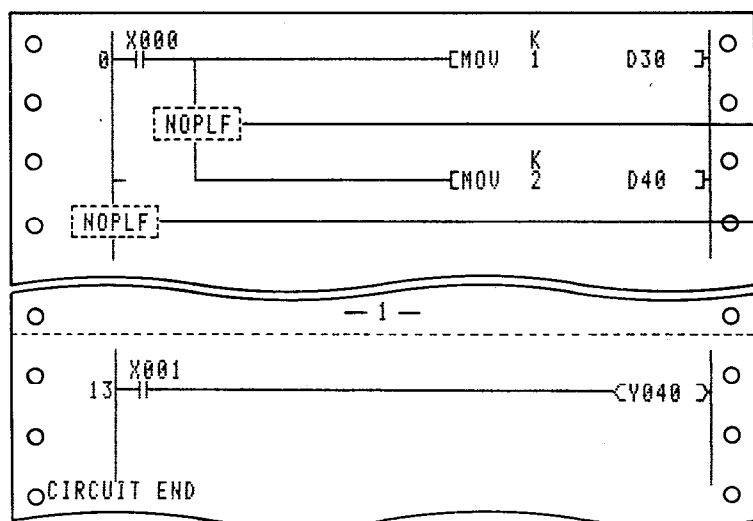
• 指令代码

```

0 LD      X000
1 MOV     K1      D30
6 NOPLF
7 MOV     K2      D40
12 NOPLF
13 LD      X001
14 OUT     Y040
15 END

```

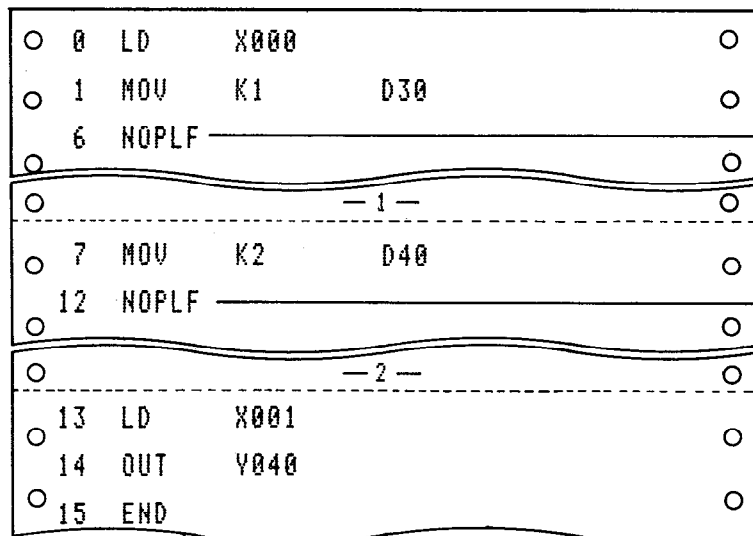
• 阶梯图打印举例



在阶梯图电路块中NOPLF指令
被忽略(打印阶梯图时不打印出来)

在阶梯图电路块末尾给出NOPLF指令时,将换页
(打印阶梯图时不打印出来)

• 指令清单打印举例



在NOPLF后换页,NOPLF仍打印出来.