Cat No:OEZ-ZCP97201A

OMRON

可编程序控制器

C200HX/C200HG/C200HE

编 程 手 册

可编程序控制器

C200HX/C200HG/C200HE

编 程 手 册

C200HX/HG/HE

可编程序控制器
编程手册

1997.1 修订

目　录

第一章　介绍 …… 1
1－1　概述 …… 2
1－2　PC 逻辑起源 …… 2
1－3　PC 术语 …… 2
1－4　OMRON 产品术语 …… 3
1－5　PC 操作概述 …… 3
1－6　外围设备 …… 4
1－7　参考手册 …… 5
1－8　C200HX/HG/HE 特性 …… 6
第二章　硬件 …… 11
2－1　CPU 部件 …… 12
2－2　PC 配置 …… 15
2－3　CPU 功能 …… 15
2－4　存储器盒 …… 16
2－5　CPU DIP 开关 …… 19
第三章　存储器区域 …… 20
3－1　介绍 …… 22
3－2　数据区域结构 …… 23
3－3　IR(内部继电器)区域 …… 26
3－4　SR(专用继电器)区域 …… 29
3－5　AR(辅助继电器)区域 …… 46
3－6　DM(数据存储器)区域 …… 52
3－7　HR(保持继电器)区域 …… 66
3－8　TC(定时器/计数器)区域 …… 66
3－9　LR(链接继电器)区域 …… 66
3－10　UM 区域 …… 67
3－11　TR(暂存继电器)区域 …… 67
3－12　UM(扩展数据存储器)区域 …… 68
第四章　编写和输入程序 …… 70
4－1　基本步骤 …… 72
4－2　指令术语 …… 72
4－3　程序容量 …… 72
4－4　基本梯形图 …… 73
4－5　编程器 …… 84
4－6　操作准备 …… 86
4－7　程序的输入、修改和检查 …… 97
4－8　位状态的控制 …… 112
4－9　工作位(内部继电器) …… 113

4－10　编程注意事项 …… 115
4－11　程序执行 …… 116
4－12　特殊 I/O 单元接口程序 …… 116
4－13　模拟定时器单元编程 …… 120
第五章　指令集 …… 125
5－1　注意事项 …… 130
5－2　指令格式 …… 130
5－3　数据区，定义符值和标志 …… 130
5－4　微分型指令 …… 131
5－5　扩展指令 …… 132
5－6　右侧指令规则 …… 134
5－7　指令集表 …… 136
5－8　梯形图指令 …… 142
5－9　位控制指令 …… 143
5－10　联锁和联锁解除指令 …… 148
5－11　跳转和跳转终了指令 …… 150
5－12　结束指令 …… 151
5－13　空操作指令 …… 151
5－14　定时器及计数器指令 …… 152
5－15　数据移位指令 …… 162
5－16　数据传送指令 …… 171
5－17　数据比较指令 …… 183
5－18　数据转换指令 …… 195
5－19　BCD 码运算指令 …… 217
5－20　二进制运算指令 …… 233
5－21　特殊算术指令 …… 247
5－22　逻辑指令 …… 264
5－23　子程序和中断控制指令 …… 268
5－24　步进指令 …… 278
5－25　特殊指令 …… 286
5－26　网络指令 …… 303
5－27　串行通信指令 …… 310
5－28　高级 I/O 指令 …… 316
5－29　特殊 I/O 单元指令 …… 329
第六章　程序执行时间 …… 333
6－1　扫描周期 …… 334
6－2　计算扫描周期 …… 338
6－3　指令执行时间 …… 340
6－4　I/O 响应时间 …… 349
第七章　程序监视和执行 …… 361
7－1　监视操作和修改数据 …… 362
7－2　编程器操作 …… 362

第八章　串行通信 …… 389
8－1　引言 …… 390
8－2　上位机链接通信 …… 390
8－3　RS－232C 通信 …… 397
8－4　一对一 PC 链接 …… 400
8－5　NT 链接 …… 402
8－6　通信协议宏功能 …… 403
第九章　故障排除 …… 411
9－1　报警指示灯 …… 412
9－2　可编程报警和错误信息 …… 412
9－3　读取和清除错误及其信息 …… 412
9－4　错误信息 …… 412
9－5　错误标志 …… 415
9－6　上位机链接错误 …… 417
第十章　上位机链接命令 …… 419
10－1　上位机链接命令汇总 …… 422
10－2　上位机链接结束码 …… 423
10－3　上位机链接命令 …… 425
附录
A、 标准模块 …… 471
B、 编程指令 …… 485
C、 错误和算术标志操作 …… 491
D、 字分配记录表 …… 494
E、 程序代码表 …… 499
F、 数据转换表 …… 502
G、 扩展 ASCII …… 504
术语 …… 506

关于本手册的说明

本手册讲述了 C200HX/HG/HE 可编程控制器的操作，它包括下列所述各章。有关安装说明见《C200HX/HG/HE 可编程控制器安装手册》。在第一章"介绍"中提供了可以和本手册一起使用的其它手册表格。第二章"硬件"叙述了 C200HS CPU 与本手册所讲的新 CPU 之间的区别。

请先阅读完本手册并理解所有内容后再操作 C200HX/HG/HE。

第 1 章"介绍"

解释背景及用于梯形图编程中的一些基本术语，同时概述了编程过程及 PC 操作，并解释了用于 OMRON PC 的基本术语。最后讲述了用于 C200HX/HG/HE PC 的外围设备以及和本手册一起针对特殊 PC 应用的其它手册表格。

第 2 章"硬件"

解释了 PC 总体配置的基本方面，并叙述了指示器(参照本手册其它章)，介绍如何使用存储器盒管理 UM 和 IOM 数据。

第 3 章"存储器区域"

简述存储器区域的划分及分配方法，并解释了所提供的信息来帮助编程。它解释了存储器区域中如何管理 I/O，以及存储器区域中的位如何对应特殊 I/O 点。它还提供了在系统 DM 中的信息，这是在 C200HX/HG/HE PC 中，使用户能灵活地控制 PC 操作参数的特殊区域。

第 4 章"编程和输入程序"

解释梯形图编程的基本概念，组成梯形图程序的图素和如何控制程序执行。同时还解释如何把梯形图转变为助记符，使程序能通过编程器输入。

第 5 章"指令集"

叙述程序中所用的全部指令。

第 6 章"程序执行时间"

解释用于执行程序的循环过程以及如何协调输入和输出使其在适当时间出现。

第 7 章"程序监视及执行"

解释了用于输入和调式程序以及监视和控制操作编程器的过程。

第 8 章"串行通信"

介绍 C200HX/HG/HE 提供的通信特点的概述。

第 9 章"故障排除"

提供错误指示信息及其它缩短停机时间的方法。本节的内容也有助于调试程序。

第 10 章"上位机链接命令"

解释了运用 C200HX/HG/HE 的通信口与上位机链接通信的上位机链接命令。

附录

提供了能用于 C200HX/HG/HE PC 的标准 OMRON 产品列表、指令参考表，有助于编程和参数输入的编码表以及其它有助于 PC 操作的信息。

警告 如未能阅读和理解本书内容可能导致人员伤亡或设备故障。请在进行任何操作前阅读每一章的全部，确认已理解该章以及相关章所提供的信息。

第一章 介 绍

本章概述了可编程控制器的历史并解释了用在梯形图程序编制中的术语。同时也提供了编程和操作PC的过程，并解释了用在OMRON PC中的基本术语。而且还提供了用于C200HX/HG/HE的外围设备，针对特殊PC应用而和本手册一起运用的其它手册列表，以及C200HX/HG/HE新的特性。

1－1 概述…… 2
1－2 PC逻辑起源 …… 2
1－3 PC术语 …… 2
1－4 OMRON产品术语…… 3
1－5 PC操作概述 …… 3
1－6 外围设备…… 4
1－7 参考手册…… 5
1－8 C200HX/HG/HE特性 …… 6
1－8－1 C200HS和C200HX/HG/HE功能 …… 6
1－8－2 程序兼容 …… 7

1－1 概 述

PC(可编程控制器)从根本上说是一个装有程序并与输入输出(I/O)设备相连的 CPU(中央处理单元)。PC 受程序控制,当输入设备转为 ON 信号输入时就会产生适当的响应,该响应通常是向某种输出设备发一个 ON 信号。输入设备可以是光电传感器、控制板上的按钮、限位开关或其它能产生信号并输入到 PC 的设备。输出设备可以是螺线管、控制指示灯的开关、起动马达的继电器,或其它能被 PC 输出信号所控制的设备。

例如当传感器检测到有产品经过时就向 PC 输入 ON 信号,PC 产生响应发出 ON 信号使机械手把产品推到另一传输带去作进一步加工,另一个传感器(安装在前一个传感器的上面)向 PC 输入一个不同的 ON 信号,表示产品太高。PC 产生响应发出 ON 信号使另一个机械手(安装在上面所提到的机械手的前面)把这个过高的产品推入废品箱。

尽管本例只牵涉到两个输入和两个输出,但却是 PC 所能达到的典型控制操作。实际上,本例要比上面描述的复杂多了,这是因为还要考虑时序。例如怎样知道何时去起动机械手。当然,还有其它更复杂的操作。问题是如何在适当的时间获取来自输入设备的控制信号。

为达到适当的控制,C200HX/HG/HE 采用 PC 逻辑图,即梯形图编制程序。本手册解释了梯形图编程并且是为准备编程和操作 PC 的读者而编写的。

1－2 PC 逻辑起源

PC 起源于继电器控制系统,尽管集成电路和 PC 的内部逻辑替代了分立继电器、定时器、计数器和其它设备,实际的 PC 处理过程就如同这些分立设备仍在起作用,但 PC 提供的计算机容量和精度使其比继电器具有更大的灵活性和可靠性。

用来描述 PC 操作的符号及其它控制概念也来自于继电器控制系统,并形成梯形图编程法的基础。但是描述这些符号和概念的术语来自于计算机术语。

继电器与 PC 术语对照

本手册使用的术语与继电器控制系统的术语有点不同,但概念是一样的。

下表列出了继电器术语和 OMRON PC 使用的 PC 术语间的关系。

继电器术语	对应 PC 的术语
接点	输入或条件
线圈	输出或工作位
NO 继电器	常开条件
NC 继电器	常闭条件

实际上这些术语并不完全对等。术语“条件”一般仅用于描述梯形图程序,特定地等同于某组基本指令。术语“输入”和“输出”本身并不用于编程,除非是指定给进入或离开 PC 的输入输出信号的 I/O 位参考号。常开条件及常闭条件在第 4－4 节基本梯形图中解释。

1－3 PC 术语

尽管本书的后面也提供了术语表,但下列术语对理解 PC 操作很重要,故详细解释如下。

PC

因为 C200HX/HG/HE 是机架式 PC，没有一种产品称之为 C200HX/HG/HE PC，这就是讨论 PC 配置的原因，PC 是由较小单元配置起来的。

为使 PC 获得某种功能，必须准备一个 CPU 机架，并至少要安装一个提供 I/O 点的单元。在提到 PC 时，总是讨论 CPU 以及 CPU 通过程序直接控制的所有单元，并不包括连到 PC 输入输出的 I/O 设备。

如果对上述描述的 PC 术语不熟悉，请参见第二章“硬件”的解释。

输入和输出

连接到 PC 并向 PC 发送信号的设备称为输入设备，发送的信号称为输入信号。信号是通过单元的端子或连接器上的插脚进入 PC。信号进入 PC 的地点称为输入点，输入点在存储器中占有位置以反映其状态，即 ON(开)或 OFF(关)，该存储位置称为输入位。CPU 在其正常的处理周期中监视所有输入点的状态，并相应地在对应输入位置 ON 或 OFF。

单元上的输出点在存储器中也有相应的输出位，通过输出点向输出设备发送输出信号，即某一个输出位置 ON，就通过输出点向输出设备送出信号。CPU 根据输出位的状态周期性地置输出点为 ON 或 OFF。

在描述 PC 操作的不同方面时，就会用到这些术语。在编程时关心存储器里的信息，因此用到 I/O 位。在谈到连接 PC 和受控系统的单元以及在这些单元上信号进入和离开的地点时，就会用到 I/O 点。在连接这些 I/O 点时，就会用到 I/O 点相应的实物：端子或连接器的插脚。在谈到进入或离开 PC 中的信号时，也就会用到输入信号或输出信号。有时简称为输入或输出，这取决于谈论 PC 操作的哪个方面。

被控系统和控制系统

控制系统用来控制外部系统，包括 PC 以及所有的 I/O 设备。传感器提供达到控制的信息。它是一个输入设备，是控制系统的一部分。被控系统是由 PC 程序通过这些 I/O 设备控制的外部系统，I/O 设备有时也可以认为是被控系统的一部分，例如驱动传输带的马达。

1－4 OMRON 产品术语

OMRON 产品按通用名称分成几种功能组。附录 A 标准模块按这些组列出了产品。术语单元用于指 OMRON PC 的所有产品。一个单元可以是指任何一个组成 C200HX/HG/HE PC 的模块，它的意义通常受提到的安装在机架上单元的上下文的限制。大部分产品都在其名称后加了“单元”两字。

OMRON 产品中最大的一组是 I/O 单元。包括所有提供通用的输入和输出点的机架安装单元。I/O 单元提供了各种接法及规格的产品。

高密度 I/O 提供高密度 I/O 容量，包括组 2 高密度 I/O 单元和特殊 I/O 高密度 I/O 单元。

特殊 I/O 单元是专用单元，用来满足特殊要求，包括一些高密度 I/O 单元，位置控制单元，高速计数器单元和模拟量 I/O 单元。

链接单元，用于建立链接系统，具有 PC 间的链接或单个 PC 与远程 I/O 点的链接。链接单元包括远程 I/O 单元，PC 链接单元，上位机链接单元，SYSMAC NET 链接单元和 SYSMAC LINK 单元，SYSMAC NET 链接和 SYSMAC LINK 单元只能用于 CPU11－E。

其它产品组包括编程设备、外围设备、以及 DIN 导轨产品。

1－5 PC 操作概述

下列是编程和操作 C200HX/HG/HE 的基本步骤。购买了一个或更多的 PC 后，肯定已经对下面简述的步骤 1 和 2 中所需的信息已经有了了解。步骤 3－6、8、9 在本手册中都有详细解释，提供更多信息的相应章节列在这些步骤后面。

1．确定被控系统做什么，按什么顺序，何时动作。

2．确定需要哪些机架和单元，参见（C200HX/HG/HE 安装手册）。如果需要链接系统，参见相应的（系统手册）。

3．把输入输出设备分配给单元上的 I/O 点并确定如何分配 I/O 记录下来，如果 PC 包括特殊 I/O 单元或链接系统，有关 I/O 位分配的细节参见个别的（操作手册）或（系统手册）。（第三章“存储器区域”）

4．用继电器梯形图符号编写程序，把操作次序及其内部联系表示出来。同时要求程序能对各种可能发生的紧急事件作出相应的响应。（第四章“编写和输入程序”、第五章“指令集”、第六章“程序执行时间”）

5．把程序及所需要的操作参数输入 PC。（第 4－7 节“程序的输入、修改和检查”。）

6．调试程序。首先消除语法错误，然后查找运行中的错误。（第 4－7 节“程序的输入、修改和检查”，第七章“程序监视和执行”、第九章“故障排除”）

7．连接 PC 和被控系统。这一步骤实际上在步骤 3 完成后即可开始。参见（C200HX/HG/HE 安装手册）和提供单个单元细节的（操作手册）及（系统手册）。

8．在实际的控制环境下调试程序，在需要时进行细调。（第七章“程序监视和执行”和第九章“故障排除”）

9．拷贝两份已完成的主机中的程序，并把它们在放在不同的安全区域。（第 4－7 节“程序的输入、修改和检查”）

控制系统设计

在自动控制处理中第一步是设计控制系统。只有全面理解了控制系统后才能对 PC 编程和操作。设计控制系统首先是要对被控系统有全面的了解，控制系统设计的第一步就是要确定被控系统的要求。

输入输出要求

被控系统所需输入输出点数必须先确定下来，这可以用确认每一个向 PC 送输入信号的设备或接受 PC 输出信号的设备来做到。可用的 I/O 点数取决于 PC 配置。有关 I/O 容量和对应 I/O 点的 I/O 位分配的细节参见 3－3“IR 区域”。

顺序、时序及其关系

接着确定所要发生的控制操作的顺序以及操作的相关时序。确认 I/O 设备之间以及在它们之间发生的各种响应的物理联系。

例如，一个光电开关通过 PC 内的计器数与一个马达相联系。当 PC 接收到一个起动开关的输入信号时，就起动马达。当计数器从光电开关接收到一定量的输入信号时，PC 就关闭马达。

从控制操作开始到结束每一个相关的任务都必须类似地决定。

单元需求

根据 I/O 设备的需求，必须确定要安装或连接到 PC 机架的实际单元，同时还要考虑硬件规格，如电压和电流等级以及考虑特殊功能配备，如是否需要特殊 I/O 单元或链接系统。在许多情况中，特殊 I/O 单元，智能 I/O 单元、链接系统能极大地减轻编程负担。有关这些单元及链接系统的细节参见相应的（操作手册）和（系统手册）。

一旦设计了整个控制系统，就能开始本手册的其余章节中描述的编程、调试和操作任务。

1－6 外围设备

下列外围设备可用于编程或输入/调试/监视 PC 程序或 PC 向外部设备输出程序和存储器中的数据。下列设备的型号参见附录 A“标准模块”。在下列描述中，写成粗体的是 OMRON 产品名称。

编程器

编程器是 OMRON PC 最简单的编程设备。所有的编程器均可直接连至 CPU，无需额外接口。

SYSMAC 支持软件：SSS(中文版)

SSS(中文版)可在 IBM AT/XT 兼容机上运行。除了可以象编程器那样操作外，还增加了许多功能。PC 程序可以用梯形图或助记符形式写在屏幕上。程序写完时可在显示屏上显示，使确认或修改快速而且容易。在程序装入 PC 前还能进行语法检查。

SSS(中文版)有 3.5 英寸软盘。

运行 SSS(中文版)的计算机用 CQM1－CIF02 或 CV500－CIF01 电缆连接至 C200HX/HG/HE PC 外设端口。

1－7 参考手册

下表列出了编程/或操作 C200HX/HG/HE PC 可能要用到的其它手册。每个单元也附有操作手册和操作指南，在接线和查其它规格时需要。

名　　称	编　号	内　　容
GPC 操作手册	W84	应用 GPC(图形编程器)编程步骤
FIT 操作手册	W150	应用 FIT(工厂智能终端)编程步骤
SSS(中文版)操作手册	W247/W248	应用 SSS 编程步骤
数据存取器操作指南	W173	应用数据存取器监视数据区域和数据修改
打印机接口单元操作指南	W107	PC 与打印机接口操作步骤
PROM 写入器操作指南	W155	将程序写入 EPROM 芯片操作步骤
软盘驱动器接口单元操作指南	W119	PC 与软盘驱动器接口操作步骤
线型远程 I/O 系统手册(SYSMAC BUS)	W120	建立线型远程 I/O 系统以实现远程 I/O 功能
光型远程 I/O 系统手册(SYSMAC BUS)	W136	建立光型远程 I/O 系统，实现远程 I/OI/O 功能
PC 链接系统手册	W135	建立 PC 链接系统，以实现在 PC 间自动转送数据
上位机链接系统手册(SYSMAC WAY)	W143	建立上位机链接系统，通过上位计算机管理 PC
SYSMAC NET 链接单元操作手册	W114	建立 SYSMAC NET 链接系统，由许多 PC 和计算机及其它外围设备组成一个光型 LAN
SYSMAC LINK 系统手册	W174	建立 SYSMAC LINK 系统，使系统中的 PC 能自动地进行数据传送，编程和编程数据传送
高速计数器单元操作手册	W141	高速计数器单元
位置控制单元操作手册	NC111：W137 NC112：W128 NC211：W166	位置控制单元
模拟量 I/O 单元操作指南	W127	C200H－AD001，C200H－DA001 模拟量 I/O 单元
模拟量输入单元操作手册	W229	C200H－AD002 模拟量输入单元
温度传感器单元操作指南	W124	温度传感器单元
ASCII 单元操作手册	W165	ASCII 单元
ID 传感器单元操作手册	W153	ID 传感器单元
声音单元操作手册	W172	声音单元
模糊逻辑单元操作手册	W208	模糊逻辑单元
模糊支持软件操作手册	W210	支持模糊逻辑单元的模糊支持软件
温度控制单元操作手册	W225	温度控制单元
加热/冷却温度控制单元操作手册	W240	加热和冷却温度控制单元
PID 控制单元操作手册	W241	控制单元
凸轮定位器单元操作手册	W224	凸轮定位器单元

1－8 C200HX/HG/HE 特性

C200HX/HG/HE 有许多新特性，在新 CPU 上，仍可以使用 C200H 和 C200HS 程序。

1－8－1 C200HS 和 C200HX/HG/HE 功能

下表列出了 C200HX/HG/HE PC 的新功能并与 C200HS 作比较。

功能		容量	
		C200HX/HG/HE	C200HS
存储器	用户存储器（UM）	C200HE－CPU11－E：3.2K 字 C200HE－CPU□2－E：7.2K 字 C200HG－CPU□3－E：15.2K 字 C200HX－CPU□4－E：31.2K 字	15.2K 字
	普通 DM	6.144 字（DM0000～DM6143） （C200HE－CPU11－E 没有 DM4000～DM5999。）	6.144 字 （DM0000～DM6143）
	固定 DM	512 字（DM6144～DM6655）	512 字（DM6144～DM6655）
	扩展 DM	0～3000 字（DM7000～DM9999）	0～3000 字（DM7000～DM9999）
	扩展 DM 存储器（EM）	6.144 字（EM0000～EM6143） C200HE：无 C200HG：6.144 字×1 组 200HX：6.144 字×3 组	无
I/O 分配	扩展机架	3 个机架 （在 C200HE－CPU□□－E 或 C200HX/HG－CPU3□－E/4□－E 中为 2 个）	2 个机架
	组－2 多点 I/O 单元	单元号 0～9，A～F （不能用于 C200HE－CPU11－E （在 C200HE－CUP□2－E，C200HX/HG－CPU3□－E/4□－E 中单元号为 0～9）	单元号 0～9
	特殊 I/O 单元	单元号 0～9，A～F （不能用于 C200HE－CPU11－E） （在 C200HE－CPU□□－E，C200HX/HG－CUP3□－E/4□－E 中单元号为 0～9）	单元号 0～9
执行时间	基本指令	0.104μs（C200HX） 0.156μs（C200HG） 0.313μs（C200HE）	0.375μs
	MOV(21)	0.417μs（C200HX） 0.625μs（C200HG） 1.250μs（C200HE）	19.00μs
	ADD(30)	16.65μs（C200HX/HG） 31.45μs（C200HE）	40.10μs
	其它指令	C200HX/HG：C200HS 执行时间的 1/3～2/3 C200HE：C200HS 执行时间 的 3/4～4/5	40.10μs
	共同处理（ENX01）处理）	0.7ms（C200HX/HG） 2.1ms（C200HE）	0.7ms
	I/O 刷新时间	和 C200HS 相同，部分特殊 I/O 刷新时间为 C200HS 的 1/2～2/3	0.7ms

（续表）

功能		容量	
		C200HX/HG/HE	C200HS
CPU 功能	RS－232C 端口	用于 C200HX/HG/HE－CPU4□－E/6□－E	用于 C200HS－CPU2□－E/3□－E
	时钟功能	用于除 C200HE－CPU11－E 之外的所有型号	用于各种型号
	SYSMAC NET 链接和 SYSMAC LINK 功能	通信板可安装在除 C200HE－CUP11－E 之外的所有 PC 上。 （板的型号为 C200HW－COM01/04－E）	用于 C200HS－CPU3□－E
通信板	SYSMAC NET 链接和 SYSMAC LINK 功能	通信板可安装在除 C200HE－CPU11－E 之外的所有 PC 上。这些板提供下列功能： SYSMAC NET 链接和 SYSMAC LINK，通信端口（端口 1 和 2），以及通信协议宏功能	用于 C200HS－CPU3□－E
特殊 I/O 单元	SYSMAC NET 链接和 SYSMAC LINK 功能	IORD(一)和 IOWR(一)指令允许存取特殊 I/O 单元的数据。	用于 C200HS－CPU3□－E
中断	中断输入单元	2 单元(16 输入)	1 单元(8 输入)
	通信板中断	可以设置	1 单元(8 输入)
	响应特性	和 C200HS 相同，但 1ms 响应只在 C200HW－SLK□□中有可能	正常模式：10ms 高速模式：1ms （使用 SYSMAC NET 链接或 SYSMAC LINK 时总是 10ms）
PT	响应特性	NT 链接(1：1)或 NT 链接(1：N)（通过 RS－422/485 链接适配器，在 RS－232C 端口上最多可连接 8 个 PT。当在 C200HE－CUP□□－E 上用一块通信板时，最多只可连 3 个 PT）	NT 链接(1：1)
SYSMAC LINK	服务时间	最多 3.5ms(1 级操作)	最多 10.8ms(1 级操作)
	远程编程	可以通过外设端口或 RS－232C 端口 （包括通信板）	可以通过外设端口
	对响应时间影响	无	任何模式均为 10ms

1－8－2 程序兼容

C200HS 程序和存储器盒可使用在 C200HX/HG/HE 中，为 C200H 开发的程序也可以很容易地传送到 C200HX/HG/HE 中使用。

SSS(中文版)操作手册中有传送程序每一步的详细过程。还需要一根 CQM1－CIF02 连接电缆，连接运行 SSS(中文版)的计算机和 C200HS。

注意事项

传送 C200H 程序到 C200HX/HG/HE 时，遵照下列注意事项。

·如果传送到 C200HX/HG/HE 的 C200H 程序中包括设置系统(SET SYSTEM) 指令(SYS(49))，该指令设置的操作参数应传送到 C200HX/HG/HE 的 PC 设置区域(DM6600，DM6601，和 DM6655)并改写任何当前设置。在送入程序运用 PC 之前要确认这些字中设置是正确的。

·如果 C200H 程序存取 C200H 在 DM0969～DM0999 中的错误记录，存取字的地址必须改为 DM6000～DM6030，它们是 C200HX/HG/HE 中的错误记录区域。

·任何依赖于周期执行时间(即依赖于执行某一部分程序所需的时间)的程序，在执行时间快得多的 C200HX/HG/HE 上运行时必须进行调整。

使用内部存储器

下面过程概括了传送 C200H 程序到 C200HX/HG/HE 中用户存储器的步骤。

1．传送程序和其它需要的数据到 SSS(中文版)的工作区。这一数据传送可以通过 C200H CPU，或软盘，或 C200HS 的存储器单元。

C200H CPU 传送。先在 SSS(中文版)上设置 PC 为 C200H，连接 SSS(中文版)和 C200H，联机，然后将程序和其它需要的数据送到 SSS(中文版)的工作区。如果在 C200H 上已生成有 I/O 表，或许还要传送 DM 数据和 I/O 表。

软盘传送。在 SSS(中文版)上设置 PC 为 C200H 和脱机方式，再将程序和需要的数据装入 SSS(中文版)的工作区。如果在 C200H 上已生成有 I/O 表，或许还要装载 DM 数据和 I/O 表。

C200H－MP831 传送。在 SSS(中文版)上设置 PC 为 C200H 和脱机方式，从存储单元上读数据到 SSS(中文版)工作区。

2．如果 SSS(中文版)还处于联机方式，则脱机。

3．将 SSS(中文版)上 PC 设置改为 C200HX/HG/HE。

4．如打算将 I/O 注释和程序一起送入 C200HX/HG/HE，为 I/O 注释分配 UM 区域。

5．连接 SSS(中文版)和 C200HX/HG/HE 并联机。

6．确认 C200HX/HG/HE CPU 上 DIP 开关的插脚 1 为 OFF，允许写入 UM 区域。

7．传送程序和所需数据到 C200HX/HG/HE，如果 C200H 上已生成有 I/O 表，或许还要传送 DM 数据和 I/O 表。

8．关断 C200HX/HG/HE，再通电，使其复位。

9．在实际操作之前，试验程序的运行。

使用存储器盒

下面的过程概括了 C200H 程序通过 EEPROM 或 EPROM 存储器盒传送到 C200HX/HG/HE 的步骤。在 C200HX/HG/HE 启动时，它还自动地从存储器盒读入程序数据。过程的前 4 步和直接传送到 C200HX/HG/HE 内部存储器(UM 区域)的相同。

1．传送程序和其它需要的数据到 SSS(中文版)的工作区。这一数据传送可以通过 C200H CPU，或从软盘，或从存储器单元。

C200H CPU 传送。先在 SSS(中文版)上设置 PC 为 200H，连接 SSS(中文版)和 C200H，联机，再将程序和其它需要的数据送到 SSS(中文版)的工作区。如果在 C200H 上已生成有 I/O 表，或许还要传送 DM 数据和 I/O 表。

软盘传送。在 SSS(中文版)上设置 PC 为 200H 和脱机方式，再将程序和其它需要的数据装入 SSS(中文版)的工作区。如果在 C200H 上已生成有 I/O 表，或许还要传送 DM 数据和 I/O 表。

C200H－MP831 传送。在 SSS(中文版)上设置 PC 为 200H 和脱机方式，从存储器单元读数据到 SSS(中文版)工作区。

2．如果 SSS(中文版)还在联机方式，则脱机。

3．将 SSS(中文版)上 PC 设置改为 C200HX/HG/HE。

4．如打算将 I/O 注释和程序一起送入 C200HX/HG/HE，为 I/O 注释分配 UM 区域。

5．在 SSS(中文版)上用 UM 分配操作，在 UM 区域中分配扩展 DM 字，从 DM7000～DM7999。

6．复制 DM1000～DM1999 中内容到 DM7000～DM7999。

7．在 DM6602 中写入“0100”，使得启动时自动将 DM7000～DM7999 的内容传送到 DM1000～DM1999。

8．用下述步骤，传送给 EEPROM 存储器盒。

a)连接 SSS(中文版)和 C200HX/HG/HE 并联机。

b)确认 C200HX/HG/HE CPU 上 DIP 开关的插脚 1 为 OFF，允许写入 UM 区域。

c)传送程序和其它需要数据到 C200HX/HG/HE。如果 C200H 已生成有 I/O 表，或许还要传送 DM 数据和 I/O 表。如果需要，确已设定传送扩展 DM 区域和 I/O 注释区域。

d)在 SSS(中文版)上置 SR27000 为 ON，传送 UM 数据到存储器盒再从第 9 步继续。或用下述步骤，传送给 EPROM 存储器盒。

a)连接 SSS(中文版)和 PROM 写入器，执行 SSS(中文版)中 EPROM 写操作将数据写入 EPROM 芯片。

e)按正确的容量设置存储器盒上 ROM 类型选择开关。

f)ROM 芯片装入存储器盒。

g)EPROM 存储器盒装入 C200HX/HG/HE。

9．C200HX/HG/HE CPU 上 DIP 开关的插脚 2 置 ON，使存储器盒的数据在启动时自动传送到 CPU。

10．关断 C200HX/HG/HE，再通电，使其复位，传送存储器盒中数据到 CPU。

11．在实际操作之前，试验程序的运行。

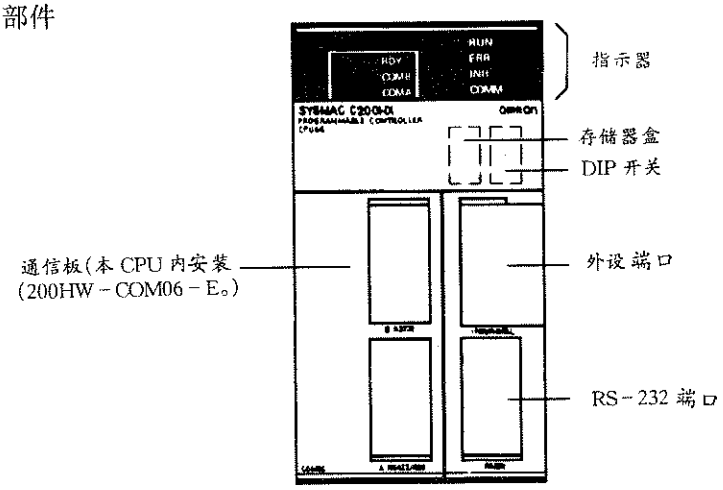
第二章　硬件

本章阐述C200HX/HG/HE中跟编程及软件操作有关的硬件情况。包括CPU部件，基本PC配置，CPU功能，以及存储器盒，这些在“C200HX/HG/HE安装手册”中都有详细说明。

2－1　CPU部件 …… 12
2－1－1　CPU指示器 …… 13
2－1－2　外围设备连接 …… 13
2－2　PC配置 …… 15
2－3　CPU功能 …… 15
2－4　存储器盒 …… 16
2－4－1　硬件和软件设置 …… 16
2－4－2　读/写UM区域 …… 17
2－4－3　读/写IOM数据 …… 17
2－5　CPU DIP开关 …… 19

2－1 CPU 部件

下图表示主要 CPU 部件



存储器盒

CPU 有一个舱,用以连接存储器盒和 CPU。存储器盒作为一个 RAM 和 CPU 的内置 RAM 一同工作。

外设端口

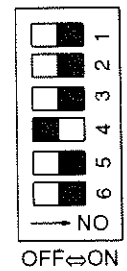
该外设端口连接一个外围设备。

RS－232C 端口

CPU 有一个内置 RS－232C 端口。

通信板

CPU 有一个舱,用以连接通信板和 CPU。



DIP 开关

PC 按照 CPU 上 DIP 开关的设置工作。C200HX/HG/HE CPU 上 DIP 开关有 6 个插脚。每个插脚的功能见下表(PC 出厂前,6 个插脚全部置为 OFF。)

脚	设置	功　能
1	ON	数据不能写入 UM 区域。
	OFF	数据可以写入 UM 区域。
2	ON	启动时,存储器盒数据自动读取。
	OFF	启动时,存储器盒数据不自动读取。
3	ON	编程器以英文显示信息
	OFF	编程器以日文显示信息
4	ON	可以设置扩展指令
	OFF	不可设置扩展指令(缺省设置)
5	ON	通信端口设置下列通信条件(包括连接 CQMI－CIFO2 连接电缆的外设端口): 1 个起动位;7 个数据位;偶校验;2 个停止位;波特率 9,600bps
	OFF	取消上述设置
6	ON	编程器处于扩展终端方式(AR0712 置 ON)
	OFF	编程器处于普通方式(AR0713 置 OFF)

2－1－1 CPU 指示器

CPU 指示器提供 PC 运行的可视信息。尽管这些指示器不能代替使用存储器数据区中标志和其它错误指示所编程的错误信息，但给正确运行提供即时的确认。

指示器	含 义
RUN(绿)	当 PC 正常运行时亮。
ERR(红)	如 PC 检测到运行中非严重错误时闪烁，PC 继续运行。如 PC 检测到运行中严重错误时亮，PC 停止工作。在 PC 停止工作之后，PUN 指示器熄灭同时输出单元的所有输出信号中断(关断)。
INH(橙)	当负载关断标志(AR 位)为 ON 时亮，此时输出单元的所有输出信号中断(关断)。
COMM(橙)	当 CPU 与连接外设端口或 RS－232C 端口的设备通信时闪烁。

2－1－2 外围设备连接

可以使用编程器或运行在 IBM PC/AT 上运行 SSS(中文版)来编程和监控 C200HX/HG/HE PC。

编程器

可以如图示连接 C200H－PRO27－E 或 CQM1－PR001－E 型编程器。C200H－PRO27－E 通过 C200H－CN222 或 C200H－CN422 连接电缆(必须单独订购)连接。CQM1－PRO01－E 和一根连接电缆同时供货。

数据存取器

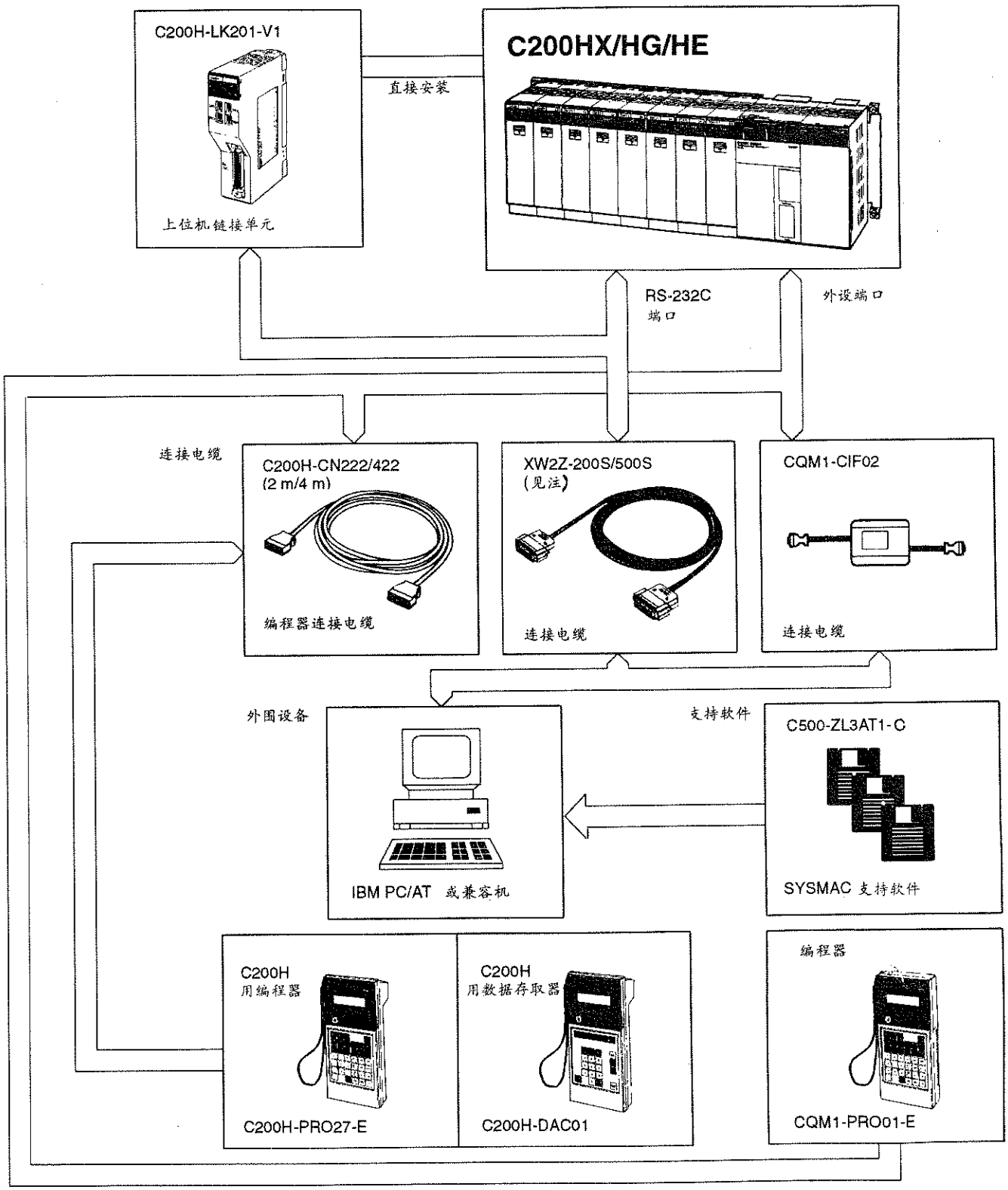
C200H－DAC01 型数据存取器通过 C200H－CN222 或 C200H－CN422 连接电缆(必须单独订购)连接。当 C200HX/HG/HE 使用 C200H－DAC01，不能执行下列操作：

设置值(SV)读取和修改

错误信息显示

用 SSS(中文版)的 IBM PC/AT

如下图所示连接运行 SYSMAC 支持软件的 IBM PC/AT 或兼容机。



注 XW2E－200S/500S 连接电缆的连接器是一只 25 芯针脚插头。为了配 IBM PC/AT 或兼容机上 9 芯针脚 D 型插头需要 1 个适配器。

2－2 PC 配置

PC 基本配置有二种类型机架组成:CPU 机架和扩展 I/O 机架。扩展 I/O 机架不是基本系统的必需部分。它是用来增加 I/O 点数。在 3－3"IR 区域"中有这些机架的列表说明。当 PC 连接远程 I/O 系统,要使用第三种类型机架,即从站机架。

CPU 机架

C200HX/HG/HE CPU 机架由三部分组成:(1)CPU 底板,在上面安装 CPU 和其它单元。(2)CPU,它执行程序和控制 PC。(3)其它单元,如 I/O 单元,特殊 I/O 单元,链接单元,由它们提供和 I/O 点相对应的实际 I/O 端点。

C200HX/HG/HE CPU 机架可以单独使用或者连接其它机架提供增加的 I/O 点。CPU 机架按使用的底板,有 3 槽,5 槽,8 槽或 10 槽,用以安装其它单元。

扩展 I/O 机架

扩展 I/O 机架可以看作是 PC 的扩展,因为它提供可以安装其它单元的附加槽。它是建立在扩展 I/O 底板上的,在上面安装电源和最多 10 个单元。

扩展 I/O 机架总是通过底板上连接器和 CUP 机架连接,以使两机架间进行通信。最多有三个扩展 I/O 机架(C200HE PC 只有二个)能与 CPU 机架串级相连。

单元安装位置

在从站机架上只可安装 I/O 单元和特殊 I/O 单元。在其它机架任何槽上都可安装所有 I/O 单元,特殊 I/O 单元,组－2 高密度 I/O 单元,远程 I/O 主站单元,PC 和上位机链接单元。中断输入单元必须装在带有"－VZ"后缀的型号底板上。

有关哪些槽可用于哪些单元以及 PC 配置的其它细节,请参阅"C200HX/HG/HE 安装手册"。单元各 I/O 点在存储器中分配的方法在 3－3 节"IR 区域"中叙述。

2－3 CPU 功能

下表列出 C200X/HG/HE CPU 功能。 CPU4□－E 和 CPU6□－E 装有 RS－232C 端口。

项 目	C200HE－		C200HG－		C200HX－	
	CPU11－E	CPU 32－E /42－E	CPU 33－E /43－E	CPU 53－E /63－E	CPU 34－E /44/E	CPU 54－U /64－E
程序容量	3.2K 字	7.2K 字	15.2K 字		31.2K 字	
DM 容量	4K 字	6K 字	6K 字		6K 字	
EM 容量	无		6K 字×1 组		6K 字×3 组	
基本指令执行时间	最小 0.3μs		最小 0.15μs		最小 0.1μs	
扩展 I/O 机架最大数量	2 机架		2 机架	3 机架	2 机架	3 机架
组－2 高密度 I/O 单元最大数量	无	10 单元	10 单元	16 单元	10 单元	16 单元
特殊 I/O 单元最大数量	10 单元		10 单元	16 单元	10 单元	16 单元
时钟功能	无	有	有		有	
通信板槽	无	有	有		有	

2－4 存储器盒

C200HX/HG/HE 装有内置 RAM，存放用户程序，即使没有存储器盒，还可以生成正常程序。一个任选的存储器盒用以存放程序，PC 设置，I/O 注释，DM 区域和其它数据区域的数据。存储器盒的安装方法参阅“C200HX/HG/HE 安装手册”。

存储器盒的功能

存储器盒用以存入和取出 UM 和 IOM 数据，存在存储器盒中的 UM 可以与 PC 中 UM 进行比较检查。

1.UM(用户存储器)的内容可以存放在存储器盒内，供今后取出和校验。如果 CPU 上 DIP 开关的插脚 2 置 ON，存储器盒中的内容在 PC 通电时自动取出。

UM 区域内有梯形图程序，固定 DM(如 PC 设置)，扩展 DM，I/O 注释，I/O 表，和 UM 区域分配信息。

2.PC 的 I/O 存储器(IOM)的内容可以存放在存储器盒中供今后取出。

IOM 包括 IR 区域，SR 区域，LR 区域，HR 区域，AR 区域，定时器和计数器 PV 值，DM1000～DM6143，和 EM000～EM6143。

C200HX/HG/HE 和 C200HS 之间 UM 和 IOM 数据完全兼容，除了 C200HX/HG/HE 部分数据区超出 C200HS 的容量范围，以及 C200HS CPU 不支持的新指令(BXF2(一)，IEMS(一)，IORD(一)，IOWR(一)，PMCR(一)，STUP(一)，和 XFR2(一)。C200HS 不支持的数据区域地址和指令不能用于 C200HS。IOM 数据只有当存储器上 IOM 的容量和 PC 中 IOM 容量一致时才能返回到 PC 的 RAM 中。

兼容的存储器盒

可以使用两种类型的存储器盒：EEPROM 和 EPROM。下表列出可用于 C200HX/HG/HE 的存储器盒。

存储器	容 量	型 号	注 解
EEPROM	4K 字	C200HW－ME04K	EEPROM 存储器盒可用以从 CPU 读取和写入 UM 和 I/O 数据。它不需要任何后备电源，即使从 CPU 上取下，仍能保留其中的数据。
	8K 字	C200HW－ME08K	
	16K 字	C200HW－ME16K	
	32K 字	C200HW－ME32K	
EPROM	16K 或 32K 字	C200HS－MP16K	存储器盒中不包括 EPROM 芯片，它必须单独订购。 27256 相当于 ROM－JD－B:16K 27512 相当于 ROM－KD－B:32K

注 1.EEPROM 在改写过 50,000 次后，存放其中的数据将不可靠。

2. 使用标准 PROM 写入器，将程序写入 EPROM 存储器盒内。先将 EPROM 接到 EPROM 存储器盒内，再安装 EPROM 存储器盒到 CPU 中。EPROM 存储器盒从 CPU 中取出将不会丢失它的数据。

2－4－1 硬件和软件设置

下面叙述与存储器盒操作有关连的硬件和软件设置

开关设置

在存储器盒出厂之前，存储器盒上开关 1 置于 OFF。安装之前核对开关 1 上设置。

存储器盒	开关 1 设置	功 能
EEPROM	ON	存储器盒中数据处于写保护。
	OFF	存储器盒中数据可以改写。
EPROM	ON	相当于 27512 的 ROM－KD－B EPROM(32K 字，150 ns 存取时间)
	OFF	相当于 27256 的 ROM－JD－B EPROM(16K 字，150 ns 存取时间)

SR 区域标志和控制位

SR269～SR273 包含与存储器盒中内容和操作有关的标志和控制位。详见 3－4“SR(专用继电器)区域”。

2－4－2 读/写 UM 区域

用下列步骤从存储器盒取出或写入 UM 数据(数据写入 EPROM 存储器盒需要 PROM 写入器。详见“SYSMAC 支持软件操作手册”。)

注 UM 内有梯形图程序,固定 DM(如 PC 设置),扩展 DM,I/O 注释,I/O 表,和 UM 区域分配信息。

UM 数据写入存储器盒

用下列步骤,将 UM 数据写入 EEPROM 存储器盒。

1. C200HX/HG/HE 通电之前,确认存储器盒上开关 1 为 OFF。
2. C200HX/HG/HE 通电,写入梯形图程序或从数据盘上读现有程序。
3. C200HX/HG/HE 切换到编程方式。
4. 用运行 SSS(中文版)的上位计算机或编程器,置 SR27000(保存 UM 至存储器盒位)ON。数据从 PC 写入存储器盒。数据传送完成后,SR27000 自动变回 OFF。
5. 如果对存储器盒上数据进行写保护,关断 PC,置存储器盒上开关 1 为 ON。当该位为 ON,即使 SR27000 为 ON,存储器盒中数据仍将保持不变。

从存储器盒读 UM 数据

从存储器盒中读 UM 数据有二种方法:启动时自动传送,或使用外围设备一次传送。

(存储器盒没有自动写入数据的功能。)

启动时自动传送:

1. CPU 上 DIP 开关插脚 2 置 ON。
2. 存有数据的存储器盒装入 C200HX/HG/HE。
3. C200HX/HG/HE 通电。存储器盒中内容自动传送到 CPU。如果数据不能送入,将出现存储器错误。

使用外围设备一次传送:

1. 存有数据的存储器盒装入 C200HX/HG/HE。
2. C200HX/HG/HE 通电,并切换到编程方式。
3. 用运行 SSS(中文版)的上位计算机或编程器置 AR2700(从存储器盒读 UM 的装载位)ON。数据从存储器盒读入至 PC。数据传送完成之后,SR27001 自动变回 OFF。

比较存储器盒上 UM 数据

用下列步骤比较存储器盒上 UM 数据和 PC 中 UM 数据。

1. 切换 C200HX/HG/HE 到编程方式。
2. 使用运行 SSS(中文版)的上位计算机或编程器,置 SR27002(比较 UM 和存储器盒位)ON。PC 和存储器盒之间进行数据比较。数据比较完成之后,SR27002 自动变回 OFF。
3. 用运行 SSS(中文版)的上位计算机或编程器,检查 SR27003(比较结果标志)的状态。

注 如果不在编程方式进行数据检查,将产生运行断续错误(FAL90)且 SR27002 变 ON。这时 SR27003 虽然为 ON,但不进行比较。如果未安装存储器盒,执行数据比较,SR27003 将为 ON。

2－4－3 读/写 IOM 数据

使用下面步骤从存储器盒取出或写入 IOM 数据(数据写入 EPROM 存储器盒需要 PROM 写入器。详见“SYSMAC 支持软件操作手册”。)

IOM 包括 IR 区域,SR 区域,LR 区域,HR 区域,AR 区域,定时器和计数器 PV 值,DM0000～DM6143,

和 EM0000～EM6143。

当 IOM 数据传送至存储器盒或从存储器盒取出时，存储器盒的容量必须和 CPU 的存储器容量相匹配。存储器要求如下：

写 IOM：CPU 容量≤存储器盒容量

读 IOM：CPU 容量＝存储器盒中 IOM 数据量

注 在 C200HS PC 中，即使存储器容量不匹配，仍进行数据传送其错误容易被忽略。

下表列出存储 1 组或更多组 EM 需要存储器盒的容量。

存储器盒容量	EM 组数
4K 字	无(不能用 1 个 4K 字存储器盒来存储其它 IOM 数(据)
8K 字	无
16K 字	1 组(只能存储 EM 组 0)
32K 字	3 组(能存储 EM 组 0～组 2)

SR273 中位 08～15 表示储存在存储器盒的 IOM 中 EM 组号。

SR27308～SR27315 中内容	含　义
00	未安装存储器盒，或存储器盒中无 IOM 数据，或存储器盒中无 EM 数据
01	存储器盒中有 IOM 数据，其中仅包括 EM 组 0(由 C200HG－CPU□□－E 存储)
04	存储器盒中有 IOM 数据，其中包括 EM 组 0～EM 组 2(由 C200HX－CPU□□－E 存储)

IOM 数据写入存储器盒

使用下列步骤将 IOM 数据写入 EEPROM 存储器盒

1．在 C200HX/HG/HE 通电之前，确信存储器盒上开关 1 为 OFF。

2．C200HX/HG/HE 通电并切换到编程方式。

3．用运行 SSS(中文版)的上位计算机或编程器，置 SR27300(保存 IOM 至存储器盒位)ON。数据从 PC 写入存储器盒。数据传送完成之后，SR27300 自动变回 OFF。

4．如果想对存储器盒上数据写保护，关断 PC，置存储器盒上开关 1 为 ON。

当该位为 ON，即使 SR27300 为 ON，存储器盒中数据仍将保持不变。

从存储器盒读 IOM 数据

使用下列步骤从存储器盒读 IOM 数据。错误记录(DM6000～DM6030)中的内容是不能从存储器盒读取的。

注 没有从存储器盒自动读取 IOM 数据的功能。

1．将存有 IOM 数据的存储器盒装入 C200HX/HG/HE。

2．接通 C200HX/HG/HE 电源，并切换到编程方式。

3．利用运行 SSS(中文版)的上位计算机或编程器，置 SR27301(从存储器盒装载 IOM 位)ON。数据从存储器盒读入至 PC。数据传送完成之后，SR27301 自动变回 OFF。

2－5 CPU DIP 开关

DIP 开关上 6 个插脚控制 CPU 的 6 个操作参数

脚	项　目	设　置	功　能
1	存储器保护	ON	外围设备不能改写 UM 区域
		OFF	外围设备可以改写 UM 区域
2	存储器盒内容自动传送	ON	启动时，存储器盒中内容自动传送到内部 RAM
		OFF	内容不自动传送
3	信息语言	ON	编程器信息用英文显示
		OFF	编程器信息用存在系统 ROM 中语言显示 （用日文版系统 ROM，将用日文显示信息）
4	扩展指令设置	ON	扩展指令由用户设置，用上位机编程/监视时为常 ON。
		OFF	扩展指令设置为缺省值。
5	通信参数	ON	通信端口（包括连接 CQM1－CIF01 连接电缆的外设端口） 起动位：1；数据长度：7 位；偶校验；停止位：2； 波特率：9，600bps
		OFF	通信不按缺省设置参数
6	当 AR0712 为 ON，设置扩展终端模式	ON	扩展终端方式（编程器）；AR0712：ON。
		OFF	普通方式（编程器）；AR0712：OFF。

注　1．UM 区域包括梯形图程序，固定 DM（包括 PC 设置），扩展 DM，I/O 注释，I/O 表，和 UM 区域分配信息。

2．PC 出厂前，6 个插脚全部设置为 OFF。

第三章　存储器区域

为实现有效及正确控制，需要各种类型的数据，而为了管理这些数据上的方便，PC 提供了各类存储器区域，每个区域都有各自的功能，用户通过编程可以存取数据的存储器区域称为数据区域，其它存储器区域是 UM 区域，用户的程序实际上就存储在该区域。本节分别讲述了各个区域以及使用时所需的内容。严格地说，TR 区域并不是存储器区域。但按照习惯，本节还是讲述了 TR 区域。

3－1　介绍 …… 22
3－1－1　数据区域概述 …… 22
3－1－2　IR/SR 区域概述 …… 23
3－2　数据区域结构 …… 23
3－3　IR(内部继电器)区域 …… 26
3－4　SR(专用继电器)区域 …… 29
3－4－1　SYSMAC NET/SYSMAC LINK 系统 …… 35
3－4－2　远程 I/O 系统 …… 36
3－4－3　链接系统标志及控制位 …… 36
3－4－4　强制状态保持位 …… 39
3－4－5　I/O 状态保持位 …… 39
3－4－6　输出关闭位 …… 39
3－4－7　FAL(故障报警)区域 …… 39
3－4－8　电池低压标志 …… 40
3－4－9　周期时间错误标志 …… 40
3－4－10　I/O 检验错误标志 …… 40
3－4－11　第一个周期标志 …… 40
3－4－12　时钟脉冲位 …… 40
3－4－13　单步标志 …… 41
3－4－14　组－2 单元错误标志 …… 41
3－4－15　特殊单元错误标志 …… 41
3－4－16　指令执行错误标志，ER …… 41
3－4－17　算术标志 …… 41
3－4－18　中断子程序区域 …… 42
3－4－19　RS－232C 端口通信区域 …… 42
3－4－20　外设端口通信区域 …… 42
3－4－21　存储器盒区域 …… 43
3－4－22　数据传送错误位 …… 44
3－4－23　梯形图存储器区域 …… 44
3－4－24　存储器错误标志 …… 44
3－4－25　数据保存标志 …… 45
3－4－26　传送错误标志 …… 45
3－4－27　PC 设置错误标志 …… 45
3－4－28　时钟和键盘映象 …… 45
3－4－29　组－2 高密度 I/O 单元错误标志 …… 45

3-4-30 特殊 I/O 单元重启动位和错误标志 …… 45
3-5 AR(辅助继电器)区域 …… 46
3-5-1 特殊 I/O 单元重启动 …… 48
3-5-2 从站机架错误标志 …… 48
3-5-3 组-2 单元错误标志 …… 48
3-5-4 光缆型 I/O 单元和 I/O 终端错误标志 …… 48
3-5-5 SYSMAC LINK 系统数据链接设定 …… 49
3-5-6 错误记录位 …… 49
3-5-7 有效节点标志 …… 49
3-5-8 SYSMACLINK/SYSMAC NET 链接系统服务时间 …… 50
3-5-9 日历/时钟区域和位 …… 50
3-5-10 终端方式键位 …… 51
3-5-11 电源关闭计数器 …… 51
3-5-12 SYSMAC LINK-外围设备标志 …… 51
3-5-13 周期时间标志 …… 51
3-5-14 链接单元安装标志 …… 51
3-5-15 CPU 安装型设备安装标志 …… 52
3-5-16 FPD 触发位 …… 52
3-5-17 数据跟踪标志及控制位 …… 52
3-5-18 周期时间指示字 …… 52
3-6 DM(数据存储器)区域 …… 52
3-6-1 扩展 DM 区域 …… 53
3-6-2 特殊 I/O 单元数据 …… 54
3-6-3 错误记录区域 …… 54
3-6-4 PC 设置 …… 56
3-6-5 通信板设置 …… 62
3-6-6 特殊 I/O 单元区域设置 …… 65
3-7 HR(保持继电器)区域 …… 66
3-8 TC(定时器/计数器)区域 …… 66
3-9 LR(链接继电器)区域 …… 66
3-10 UM 区域 …… 67
3-11 TR(临时继电器)区域 …… 67
3-12 EM(扩展数据存储器)区域 …… 68
3-12-1 使用 EM 区域 …… 68
3-12-2 当前 EM 组 …… 69

3－1 介　绍

3－1－1 数据区域概述

下表列出了每个区域的名称、大小、范围。数据和存储器区域的常用简称，如 IR 区域、SR 区域等等。

区　域	大　小	范　围	注　释
内部继电器区域 1	3,776 位	IR000～IR235	详见 3－1－2“IR/SR 区域概述”和 3－3“IR(内部继电器)区域”。
特殊继电器区域 1	312 位	SR236～SR255	详见 3－1－2“IR/SR 区域概述”和 3－4“SR(特殊继电器)区域”。
特殊继电器区域 2	704 位	SR256～SR299	
内部继电器区域 2	3,392 位	IR300～IR511	详见 3－1－2“IR/SR 区域概述”和 3－3“IR(内部继电器)区域”。
临时继电器区域	8 位	TR00～TR07	当编制某种类型的分支梯形图时，可用于临时存储和读取执行条件。
保持继电器区域	1,600 位	HR00～HR99	当 PC 掉电时，用于存储数据和保留数值。
辅助继电器区域	448 位	AR00～AR27	包括标志位和特殊功能位，掉电时保留状态。
链接继电器区域	1,024 位	LR00～LR63	用于链接系统中数据链接。(如不用于 PC 链接系统，这些位可用作工作字或工作位。)
定时器/计数器区域	512 个定时器/计数器	TC000～TC511	用于定义定时器和计数器以及存取结束标志，PC 和 SV 值。 TIM000～TIM015 作为高速定时器通过中断处理刷新。
数据存储器(DM)区域	6,144 字	DM0000～DM6143	读/写
	1,000 字	DM0000～DM0999	普通 DM
	2,600 字	DM1000～DM2599	特殊 I/O 单元区域
	3,400 字	DM2600～DM5999	普通 DM
	31 字	DM6000～DM6030	历史记录
	(44 字)	DM6100～DM6143	链接测验区域(保留)
固定 DM 区域	512 字	DM6144～DM6599	固定 DM 区域(只读)
	56 字	DM6600～DM6655	PC 设置
扩展数据存储器(EM)区域	6,144 字	EM0000～EM6143	EM 区域存储容量取决于使用 PC 的型号。PC 可以是无 EM 区，1 组 6,144 字 EM 区和 3 组 6,144 字 EM 区。和 DM 一样，EM 只能按字存取，当 PC 掉电时，EM 区域中数据被保留。

工作位和工作字

数据区域中的某些字和位不用于固有用途时，在编程中可以用它们去控制其它位。用于这种功能的字或位称为工作字或工作位。大多数(不是全部)没有用的位都可用作工作位，本章后面将一个个区域地叙述可以使用的位。第四章“编写和输入程序”讲述工作位和工作字实际应用。

标志位和控制位

一些数据区域包括标志位和(或)控制位。标志位能自动置“ON”或“OFF”来反映特别的操作状态。有一些标志位可由用户设置为 ON 和 OFF，但大多数标志位只能读，而不能被直接控制。

控制位是由用户设置 ON 和 OFF 来控制特定操作的位。用字“位”而非字“标志”具名的任何位就是控制位，例如重起动位就是控制位。

3－1－2 内部继电器(IR)/特殊继电器(SR)区域概述

当指定一个数据区域时，除了 IR 和 SR 外任何区域都需要指出该区域的简称。虽然为了解释清晰，也给出 IR 和 SR 区域简称，但在编程时不需要指明也不需要输入。

IR 和 SR 区域被分成两个 256 字段，两段的分界线位于 SR 区域中 SR255 和 SR256 之间。当 SR 区域在程序中用作为一个操作数，该操作数不可越过这个分界线。另外基本指令存取第二段中的位(SR25600～IR51115)，要花费稍长的指令执行时间。

区域		范围	注释
IR 区域 1	I/O 区域 1	IR000～IR029	按槽位置分配给 CPU 机架和扩展 I/O 机架的 I/O 字。
	组－2 高密度 I/O 单元和 B7A 接口单元区域 1	IR030～IR049	分配给组－2 高密度 I/O 单元和组－2B7A 接口单元(0～9)。
	SYSMAC BUS 区域	IR050～IR099	分配给远程 I/O 从站机架(0～4)。
	特殊 I/O 单元区域 1	IR100～IR199	分配给特殊 I/O 单元(0～9)。
	光缆 I/O 单元和 I/O 终端区域	IR200～IR231	分配给光缆 I/O 单元和 I/O 终端。
	工作区	IR232～IR235	用于程序中工作位。
SR 区域 1		SR23600～SR25507	包含系统时钟，标志，控制位，和状态信息。
SR 区域 2		SR256～SR299	包含标志，控制位和状态信息。SR290～SR297 由 MCRO(99)用作 I/O 字。
IR 区域 2	I/O 区域 2	IR300～IR309	按槽位置分配给第三个扩展 I/O 机架的 I/O 字。
	工作区	IR310～IR329	用于程序中工作位。
	组－2 高密度 I/O 单元区域 2	IR330～IR341	分配给组－2 高密度 I/O 单元。
	工作区	IR342～IR349	用于程序中工作位。
	特殊 I/O 单元区域 2	IR350～IR399	未分配。
		IR400～IR459	分配给特殊 I/O 单元(A～F)。
	工作区	IR460～IR511	用于程序中工作位。

注 1. 有关 IR 区域内容详见 3－3 节“IR 区域”。

2. IR 区 1 和 IR 区 2 中的位若未被使用，可在编程中作为工作位使用。

3－2 数据区域结构

设计数据区时，通常应给出数据区域的简称，但 IR 区域和 SR 区域不同，尽管在文章解释时经常很明确地指出 IR 区域和 SR 区域，但在编程中并不要求指定和输入。无简称的数据区域总是指 IR 区域或 SR 区域。因为 IR 区域和 SR 区域的地址是顺序编址的，字或位的地址足以区分这两个区域。

除了 TC 区域，其它数据区域的数据位置都通过地址确定。地址指定了所需数据在区域中的位或字。TC 区域由 TC 号组成，每个号用于程序中定义不同的定时器或计数器。TC 号的细节参见第 3－8 节“TC 区域”，TC 的应用参见第 5－14 节“定时器和计数器指令”。

其它的数据区域(如 IR，SR，HR，DM，AR 和 LR 区域)由字组成，每个字由 16 位组成，依次从右到左编号为 00 到 15。下面列出了 IR 字 000 和 001，每个字的内容都为 0，位 00 称为最右位，位 15 称为最左位。

术语“最高位”常指最左位，“最低位”常指最右位。在本手册中不采用。因为数据的字常分成几部分，每部分用作不同的参数或操作数，如果把不同字上的位接起来组成新字时，有可能使最右位实际上变成最高位。

位序号	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
字 IR000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
字 IR001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DM 区域只能存取字，而不能定义其中的某一位。在 IR、SR、HR、AR 和 LR 区域中可以存取数据的字或位，这取决于操作数据的指令。按字指定存储区域时，应包括存储区域简称（如有必要）和 2、3 或 4 位的字地址。如按位指定存储区域时，字地址加上位序号一同组成单个的 4 或 5 位数的地址。如下表所示。最右的 2 位数在 00～15 之间。

相同的 TC 号可用来指定定时器或计数器的 PV 以及定时器或计数器的完成标志位。这一点在第 3－8 节“TC 区域”有更详细的说明。

区　域	字　指　定	位　指　定
IR	000	00015（字 000 的最左位）
SR	252	25200（字 252 的最右位）
DM	DM1250	不能用
TC	TC215（指 PV）	TC215（指完成标志）
LR	LR12	LR1200

数据结构

以十进制形式输入的数据用 BCD 码形式存储，以十六进制输入的数据用二进制形式存储。字中每 4 位代表一位数，每一个十进制数或十六进制数都可以等价地表示成四位二进制值。这样，一个字就能表示四位数字，这四位数依次从右向左编号。数字字与二进制位号的相对应关系如下图所示：

数字序号	3				2				1				0			
位序号	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
内容	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

对于整个字，数字序号为 0 的数称为最右位数字，数字序号为 3 的数称为最左位数字。输入数据时，应按要求输入适当的形式，这在指定单个位时不会出问题，因为位只能为“ON”（相当于二进制“1”）或“OFF”（相当于二进制“0”）。而按字输入数据时，是用十进制还是用十六进制输入就很重要了。这取决于所用的指令。第 5 节〈指令集〉详述了如何处理指令所要求的特定数据形式。

不同形式数据转换

二进制数与十六进制数很容易转换，因为四位二进制数等价于一位十六进制数。按次序从右边开始把二进制数 0101111101011111 转换至十六进制数，二进制数 1111 等价于十六进制数 F，二进制数 0101 等价于十六进制数 5，因此该二进制数等价于十六进制的 5F5F 或十进制的 24，415（$16^3 \times 5 + 16^2 \times 15 + 16 \times 5 + 15$）。

十进制数与 BCD 也很容易转换。每一个 BCD 码数字（即 4 个 BCD 位的组）等价于一个十进制数字，例如 BCD 位 0101011101010111 以每 4 位为一组从右开始转换，二进制数 0101 对应十进制数 5，二进制数 0111 对应十进制数 7，因此对应的十进制数为 5757。必须注意到，这与十六进制数 0101011101010111 在数值上不同，它用十六进制数表示为 5757 或十进制数表示为 22，359（$16^3 \times 5 + 16^2 \times 7 + 16 \times 5 + 7$）。

因为每 4 位 BCD 码对应一个十进制数字，故大于 9 的 4 位二进制数不能用。例如 1011 不允许出现，因为它对应十进制数的 11，不能用一个十进制数字表示，当然在十六进制中允许用 1011，因为它等价于十六进制数的 C。

指令中具有 BCD 与十六进制数之间互相转换的指令。参见第 5－18 节“数据转换指令”。在附录中提供了二进制与十六进制及 BCD 数字间的转换表。

十进制小数点

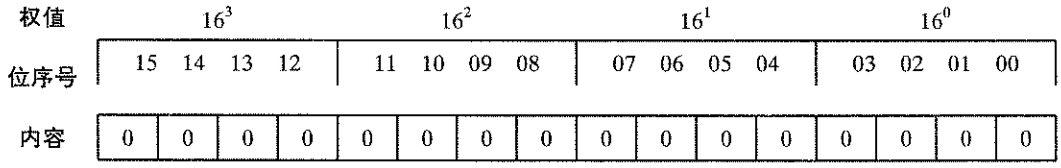
十进制小数点仅用于定时器，最低位代表十分之一秒，所有的算术指令只能操作整数。

带符号及不带符号的二进制数据

本节解释了带符号和无符号的二进制数据格式，许多指令既可以用带符号数据也可以用无符号数据，一些指令(CPS(一)，CPSL(一)，DBS(一)，BDSL(一)，MBS(一)和 MBSL(一))仅能用带符号数据。

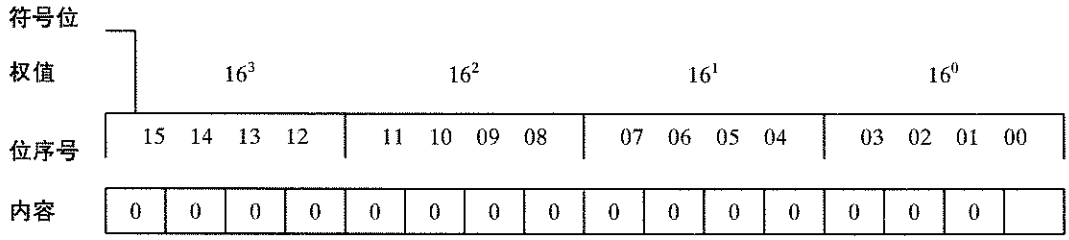
无符号二进制数

无符号二进制数在 OMRON PC 中是标准格式，除非特别声明，在本手册中都是指无符号数。无符号数总是表示正的，范围为 0(0000H)～65,535(FFFFH)，8 位数字的值范围从 0(0000 0000H)～4,294,967,295(FFFF,FFFFH)。



带符号二进制数

带符号二进制数可正可负，符号由位 15 表示，位 15 为“OFF”时表示正，位 15 为“ON”时表示负。正数范围从 0(0000H)到 32,767(7FFF)，负数范围为－32,768(8000H)到－1(FFFFH)。



8 位数字的正值范围从 0(0000 0000H)到 2,147,483,647(7FFF FFFH)，8 位数字的负值范围为－2,147,483,648(8000 0000H)到－1(FFFF FFFFH)。

下表列出了十进制数和 16 位的十六进制数及 32 位的十六进制数的对应值。

十进制数	16 位十六进制数	32 位十六进制数
2147483647	－	7FFFFFFF
2147483646	－	7FFFFFFE
・	・	・
・	・	・
・	・	・
32768	－	00008000
32767	7FFF	00007FFF
32766	7FFE	00007FFE
・	・	・
・	・	・
2	・	・
1	0002	00000002
0	0001	00000001
－1	0000	00000000
－2	FFFF	FFFFFFF
・	FFFE	FFFFFFE
・	・	・
・	・	・
－32767	・	・
－32768	8001	・
－32769	8000	FFFF8001
	－	FFFF8000
・	・	FFFF7FFF
・	・	・
・	・	・
－2147483647	－	80000001
－2147483648	－	80000000

十进制数转换到带符号二进制数

正的带符号二进制数(最大 32,767),与无符号二进制数相等,可以用 BIN(23)来转换。下列过程是把在－32,768～－1 之间的十进制负数转换为带符号二进制数。以－12345 转换为 CFC7 为例。

1. 第一步取绝对值(12345)并转换为无符号二进制数。

位序号	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
内容	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1

2. 取反

位序号	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
内容	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0

3. 最后加 1

位序号	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
内容	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1

如要把负的二进制数转换到十进制数,可把上述过程反过来用。

3－3 IR(内部继电器)区域

IR 区域既可用作控制 I/O 点的数据,也可用作内部处理和存储数据的工作位。它可以按位或按字存取。在 C200HX/HG/HE PC 中,IR 区域由字 IR000～IR235(IR 区域 1)和 IR300～IR511(IR 区域 2)组成。存取 IR 区域 2 的基本指令执行时间比存取 IR 区域 1 的基本指令执行时间稍长些。

在 IR 区域中,用来控制 I/O 点的字称 I/O 字,I/O 字中的位称为 I/O 位。在 IR 区域中未用作 I/O 位的其它位都可用作工作位。IR 区域的工作位在电源断开或 PC 操作停止时被复位。

区域		范围
IR 区域 1	I/O 区域 1	IR000～IR029
	组－2 高密度 I/O 单元区域 1 和 B7A 接口单元区域	IR030～IR049
	SYSMAC BUS 区域	IR050～IR099
	特殊 I/O 单元区域 1	IR100～IR199
	光缆 I/O 单元和 I/O 终端区域	IR200～IR231
	工作区域	IR232～IR235
IR 区域 2	I/O 区域 2	IR300～IR309
	工作区域	IR310～IR329
	组－2 高密度 I/O 单元区域 2	IR330～IR341
	工作区域	IR342～IR349
	特殊 I/O 单元区域 2	IR350～IR459
	工作区域	IR460～IR511

I/O 字

如单元输入信号至 PC,那么分配给该单元的位为输入位。如单元从 PC 输出信号,那么分配给单元的位为输出位。要使输出通电,相应的输出位就要置“ON”,当输入接通时,相应的输入位就会置“ON”。这样

就可在程序中通过 I/O 位读取输入状态,及控制输出状态。

输入位使用

输入位可直接输入外部信号至 PC,并在编程中不用考虑次序。为达到有效而适当的控制,每个输入位可根据需要被任意次使用。但输入位不能用于控制位状态的指令,如 OUTPUT, DIFFERENTIATION UP 和 KEEP 指令。

输出位使用

输出位用于输出程序执行结果并在编程中不用考虑次序。因为在每个周期(即每执行一次程序)输出位仅刷新一次,故每一位输出位仅能用于一条控制其状态的指令。

这些指令包括 OUT, KEEP(11), DIFU(13), DIFD(14)和 SFT(10)等。

如果一个输出位用于多条这样的指令,由 PC 实际输出的只是最后一条指令确定的状态。在两条'位控制'的指令中使用同一输出位的例子,见 5-15-1"移位寄存器指令-SFT(10)"。

机架的字分配

I/O 字根据槽位置分配给 CPU 机架和扩展 I/O 机架。如下表所示,每个槽分配一个 I/O 字。因此,3 槽机架只能使用最前面的 3 个字,5 槽机架只能使用最前面的 5 个字,8 槽机架只能使用最前面的 8 个字,分配给不用的或不存在的槽的字可用作工作字。

机架左侧 10 槽机架右侧

机 架	槽 1	槽 2	槽 3	槽 4	槽 5	槽 6	槽 7	槽 8	槽 9	槽 10
CPU	IR000	IR001	IR002	IR003	IR004	IR005	IR006	IR007	IR008	IR009
扩展机架 1	IR010	IR011	IR012	IR013	IR014	IR015	IR016	IR017	IR018	IR019
扩展机架 2	IR020	IR021	IR022	IR023	IR024	IR025	IR026	IR027	IR028	IR029
扩展机架 3	IR300	IR301	IR302	IR303	IR304	IR305	IR306	IR307	IR308	IR309

未使用的字

分配给某单元的字如不用,则在编程中可用作工作字和工作位,这些单元包括链接单元,如:上位机链接单元,PC 链接单元 SYSMAC NET 链接单元,等,特殊 I/O 单元,远程 I/O 主单元,组-2 高密度 I/O 单元,组 2 B7A 接口单元,以及辅助电源单元。

特殊 I/O 单元和从站机架的分配

在大多数 C200HX/HG/HE PC 中,CPU 机架或扩展 I/O 机架上的任何槽最多可安装 16 个特殊 I/O 单元(在远程 I/O 从站机架上,也可安装有限数量的特殊 I/O 单元)。每个特殊 I/O 单元根据它的单元号(0~F)分配有 10 个字。

在 C200HE-CPU□□-E 和 C200HG/HX-CPU3□-E/4□-E PC 中,最多可安装 10 个 I/O 特殊单元。每个单元根据它的单元号(0~9)分配有 10 个字。

单元号	I/O 字	PC 限制
0	IR100~IR109	无
1	IR110~IR119	
2	IR120~IR129	
3	IR130~IR139	
4	IR140~IR149	
5	IR150~IR159	
6	IR160~IR169	
7	IR170~IR179	
8	IR180~IR189	
9	IR190~IR199	

单元号	I/O 字	PC 限制
A	IR400~IR409	C200HE-CPU□□-E 和 C200HG/HX-CPU3□-E/4□-E 中不能使用
B	IR410~IR419	
C	IR420~IR429	
D	IR430~IR439	
E	IR440~IR449	
F	IR450~IR459	

注 未分配给特殊 I/O 单元的 I/O 字可作为工作字使用。

不论使用 1 个或 2 个主站单元,都可以使用最多 5 个从站机架。下表列出按单元上单元号分配给每个从站机架的 IR 区域字。

单元号	I/O 字
0	IR050～IR059
1	IR060～IR069
2	IR070～IR079
3	IR080～IR089
4	IR090～IR099

可以使用 C500－RT001/002－(P)V1 远程 I/O 从站机架,但它需要 20 个 I/O 字,而不是 10 个。因而占用了足以分配给 2 个 C200H 从站机架的 I/O 字,即设置在机架上的单元号,以及下一个单元号的 I/O 字。使用 C200HX/HG/HE CPU 时,不要把 C500 从站机架上的单元号设为 4,因为没有单元号 5。在 C500 从站机架上,I/O 字从左到右分配给安装的单元,而不是象 C200HXHG/HE 机架那样分配给槽。

光缆型 I/O 单元和 I/O 终端分配

IR200 至 IR231 的 I/O 字根据单元号分配给光缆型 I/O 单元和 I/O 终端;分配给每个单元的 I/O 字为 IR200＋n,其中 n 是单元上设置的单元号。

远程 I/O 主单元和链接单元分配

远程 I/O 主单元及上位机链接单元并不使用 I/O 字,PC 链接单元使用 LR 区域,因此分配给装有这些单元的槽的字可作为工作字。

I/O 单元的位分配

根据所用的型号,I/O 单元可以用 8～16 位,对大多数 I/O 单元,不用于输入输出的位可用作工作位。但晶体管输出单元 C200H－OD213 和 C200H－OD411,以及双向可控硅输出单元 C200H－OA221 用位 08 设置保险丝熔断标志,晶体管输出单元 C200H－OD214 用位 08～11 设置报警标志。因此,在分配给这些单元的字中,位 08～15 是不能用作工作位的。

中断输入单元位分配

在 CPU 机架上的中断输入单元只使用分配给它所在槽位的第一个 I/O 字的前 8 位(安装在扩展 I/O 机架上的中断输入单元象普通的输入单元一样工作),分配给这个槽位的其它 24 位可用作工作位。

组－2 高密度 I/O 单元和 B7A 接口单元的分配

组－2 高密度 I/O 单元和 B7A 接口单元不使用所在的槽号所分配到的字,而是根据设置的 I/O 号分配 IR030－IR049 中的字,对于 32 点的单元,每个单元分配 2 个字,对于 64 点的单元,每个单元分配 4 个字。下表列出了分配给每个 I/O 号的字,不用于输入输出的字或部分,在编程中可用作工作字或工作位。

32 点单元		64 点单元	
I/O 号	字	I/O 号	字
0	IR30－IR31	0	IR30－IR33
1	IR32－IR33	1	IR32－IR35
2	IR34－IR35	2	IR34－IR37
3	IR36－IR37	3	IR36－IR39
4	IR38－IR39	4	IR38－IR41
5	IR40－IR41	5	IR40－IR43
6	IR42－IR43	6	IR42－IR45
7	IR44－IR45	7	IR44－IR47
8	IR46－IR47	8	IR46－IR49
9	IR48－IR49	9	不能用

当给高密度 I/O 单元和 B7A 接口单元设置 I/O 号时，应注意不要把同一个字分配给一个以上的单元，例如：I/O 号 0 分配给 64 点单元后，那么 I/O 号 1 就不能用于系统中的任何单元。

组－2 高密度 I/O 单元和 B7A 接口单元不归类于特殊 I/O 单元，不管用了多少号，并不影响系统中特殊 I/O 单元使用数量的限制。

下表列出了分配给组－2 高密度 I/O 单元的字与单元上连接器的对应关系。

单　元	字	连接器/列
32 点单元	第 1 个	A 列
	第 2 个	B 列
64 点单元	第 1 个	CN1，A 列
	第 2 个	CN1，B 列
	第 3 个	CN2，A 列
	第 4 个	CN2，B 列

注　组－2 高密度 I/O 单元和 B7A 接口单元不能用在从站机架上。

3－4　SR(专用继电器)区域

SR 区域包括标志位和控制位。用来监视 PC 操作，存取时钟脉冲以及显示错误信号。SR 区域的字地址从 236～299，位地址从 23600～29915。

SR 区域分为二段，第一段结束于 SR255，第二段从 SR256 开始。当 SR 区域字用作指令操作数，该操作数不可超越这分界线。存取 SR 区域 2 的基本指令执行时间长些。

区　域	范　围
SR 区域 1	SR23600～SR25507
SR 区域 2	SR25600～SR29915

下表列出了 SR 区域标志位及控制位的功能，大多数位在后面还有详细的描述。除了链接系统位以组叙述，其它位按位顺序叙述。

除非特别声明，标志位一般为“OFF”，只有在预定条件出现时，才置“ON”，重起动位通常“OFF”，但当用户把某位置“ON”后再置“OFF”，就会重起动所指定的链接单元。其它控制位为“OFF”，直到用户设置。

并非所有的 SR 字及位都可由用户改写，故如要在程序中用到它们，应检查一下每一个位或字的功能。

字	位	功　能
236	00～07	0 级 SYSMAC NET 链接系统节点环路状态输出区域
	08～15	1 级 SYSMAC NET 链接系统节点环路状态输出区域
237	00～07	0 级 SYSMAC LINK/SYSMAC NET 链接系统执行 SEND(90)/RECV(98)后的完成码输出标志
	08～15	1 级 SYSMAC LINK/SYSMAC NET 链接系统执行 SEND(90)/RECV(98)后的完成码输出标志
238 和 241	00～15	0 级 SYSMAC LINK 或 SYSMAC NET 链接系统数据链接状态输出区域
242 和 245	00～15	1 级 SYSMAC LINK 或 SYSMAC NET 链接系统数据链接状态输出区域
246	00～15	不用
247 和 248	00～07	用于单元 16～31 的 PC 链接单元运行标志或 1 级操作中的数据链接标志
	08～15	用于单元 16～31 的 PC 链接单元错误标志或 1 级操作中的数据链接标志
249～250	00～07	用于单元 00～15 的 PC 链接单元运行标志或 0 级操作中的数据链接标志
	08～15	用于单元 00～15 的 PC 链接单元错误标志或 0 级操作中的数据链接标志

（续表）

字	位	功　能
251（写允许）	00	远程 I/O 错误读出位
	01～02	不用
	03	远程 I/O 错误标志
	04～06	远程 I/O 单元，光缆型 I/O 单元或 I/O 终端的从站机架号和单元号出错
	07	不用
	08～15	远程 I/O 单元，光缆型 I/O 单元或 I/O 终端分配字(十六进制)和主站单元号出错
252	00	0 级 SYSMAC LINK 或 SYSMAC NET 链接系统 SEND(90)/RECV(98)错误标志
	01	0 级 SYSMAC LINK 或 SYSMAC NET 链接系统 SEND(90)/RECV(98)允许标志
	02	0 级数据链接操作标志
	03	1 级 SYSMAC LINK 或 SYSMAC NET 链接系统 SEND(90)/RECV(98)错误标志
	04	1 级 SYSMAC LINK 或 SYSMAC NET 链接系统 SEND(90)/RECV(98)允许标志
	05	1 级数据链接操作标志
	06	机架安装型上位机链接单元 1 级通信错误标志
	07	机架安装型上位机链接单元 1 级重起动位
	08	RS－232C 端口错误标志
	09	RS－232C 重启动位
	10	PC 设置清除位
	11	强制状态保持位
	12	数据保持控制位
	13	机架安装型上位机链接单元 0 级重起动位
	14	不用
	15	输出关闭(OFF)位
253	00－07	FAL 号输出区域(见其它部分提供的错误信息)
	08	电池电压过低标志
	09	周期时间错误标志
	10	I/O 检验错误标志
	11	机架安装型上位机链接单元 0 级通信错误标志
	12	远程 I/O 错误标志
	13	常通(ON)标志
	14	常断(OFF)标志
	15	第一个周期标志

字	位	功　能
254	00	1分钟时钟脉冲位
	01	0.02秒时钟脉冲位
	02	负(N)标志
	03	MTR执行标志
	04	上溢出标志(用于带符号的二进制数运算)
	05	下溢出标志(用于带符号的二进制数运算)
	06	微分监视结束标志
	07	单步标志
	08	HKY执行标志
	09	7SEG执行标志
	10	DSW执行标志
	11	中断输入单元错误标志
	12	第一周期标志
	13	中断程序错误标志
	14	组－2错误标志
	15	特殊单元错误标志(特殊I/O,PC链接单元,上位机链接单元,远程I/O主单元)
255	00	0.1秒时钟脉冲位
	01	0.2秒时钟脉冲位
	02	1.0秒时钟脉冲位
	03 指令执行错误标志	执行END(01)指令时,这些标志都置OFF,故不能用编程器监视它们的状态。 参见附录C中表,显示哪些指令影响这些标志。
	04 进位(CY)标志	
	05 大于(GR)标志	
	06 相等(EQ)标志	
	07 小于(LE)标志	
	08～15 系统保留(用作TR位)	
256～261	00～15	系统保留
262	00～15	最长的中断子程序执行时间(0.1ms)
263	00～15	用最长执行时间进行的中断子程序号(8000－8255)(位15:中断标志)

字	位	功　能
264	00～03	RS－232C 端口错误码 0:没有错误　4:FCS 错误 1:奇偶校验错误　5:超时错误 2:帧错误　6:校验和错误 3:运行错误　7:命令错误
	04	RS－232C 端口通信错误
	05	RS－232C 端口发送准备好标志
	06	RS－232C 端口接收完成标志
	07	RS－232C 端口接收溢出标志
	08～11	外设端口错误码(普通 I/O 方式) 0:没有错误　3:运行错误　6:校验和错误 1:奇偶校验错误　4:FCS 错误　7:命令错误 2:帧错误　5:超时错误
	12	外设端口通信错误(普通 I/O 方式)
	13	外设端口发送准备标志(普通 I/方式)
	14	外设端口接收结束标志(普通 I/O 方式)
	15	外设端口接收溢出标志(普通 I/O 方式)
265	00～15	NT 链接(1:N)方式 位 00～07:与单元 0～7 的 PT 标志位通信 位 08～15:登记单元 0～7 的 PT 奇偶校验标志位 RS－232C 方式 位 00～15:RS－232C 端口接收计数器
266	00～15	RS－232C 方式下外设接收计数器
267	00～04	系统保留(用户不能存取)
	05	上位机链接 0 级发送准备标志
	06～12	系统保留(用户不能存取)
	13	上位机链接 1 级接收准备标志
	14～15	不用
268	00－15	通信板错误信息
269	00～07	存储器盒内容 00:(空);01:UM;02:IOM;03:HIS
	08～10	存储器盒容量 0:OK 字(没有存贮器盒);2:4～8K 字;3:16K 字;4:32K 字
	11～13	系统保留(用户不能存取)
	14	EEPROM 存储器盒保护或 EPROM 存储器盒安装标志
	15	存储器盒标志

字	位	功能	
270	00	保存 UM 至存储器盒位	在编程方式中该位置“ON”时，数据传送到存储器盒。该位能自动置“OFF”。如在其它方式中置“ON”，发生非致命错误。
	01	从存储器盒装入 UM 位	
	02	比较 UM 和存储器盒位	
	03	比较结果 0：内容相同； 1：内容不同或者不能比较	
	04～10	不用	
	11	传送错误标志：在有效数据链接期间，传送 SYSMAC NET 数据链接表到 UM。	如有出错(除了帧校验和错误外)，数据将不会从 UM 传送至存储器盒。存储器盒的校验和错误的详细信息不会输出至 SR272，因为不需要这些信息。如果置 SR27015 位为“ON”，则重复执行传送。
	12	传送错误标志：非编程方式	
	13	传送错误标志：只能读	
	14	传送错误标志：容量不足或没有 UM	
	15	传送错误标志：帧校验和错误	
271	00～07	存储器盒中的梯形图程序容量大小 梯形图文件：04：4K 字，08：8K 字，12：12K 字，…(32：32K 字)，00：无梯形图程序或存贮有非梯形图程序的文件	
	08～15	在 CPU 中的梯形图程序容量大小及类型(定义同位 00～位 07)	
272	00－10	不用。	
	11	存储器错误标志：PC 设置校验和错误。	
	12	存储器错误标志：梯形图校验和错误。	
	13	存储器错误标志：指令变化矢量区域校验和错误。	
	14	存储器错误标志：存储器盒在线时断开。	
	15	存储器错误标志：自动引导错误。	
273	00	把 IOM 保存至存储器盒位	在编程方式中该位置“ON”时数据传送到存储器盒。该位能自动置“OFF”，如在其它方式中置“ON”，就会出错。
	01	从存储器盒装入 IOM 位	
	02	该位置 0。	
	03～07	不用。	
	08～11	含 EM 组号，当存储器盒存有 IOM 数据时。	
	12	传送错误标志：非编程方式	如有出错(除了只读错误外)，数据不能从 IOM 传送至存储器盒。
	13	传送错误标志：只能读	
	14	传送错误标志：容量不足或无 IOM	
	15	总 0	

字	位	功能	
274	00	0 特殊 I/O 单元重启动位	重启动处理过程中,这些标志变为 ON。对从站机架上单元,这些标志不会变 ON。
	01	#1 特殊 I/O 单元重启动位	
	02	#2 特殊 I/O 单元重启动位	
	03	#3 特殊 I/O 单元重启动位	
	04	#4 特殊 I/O 单元重启动位	
	05	#5 特殊 I/O 单元重启动位	
	06	#6 特殊 I/O 单元重启动位	
	07	#7 特殊 I/O 单元重启动位	
	08	#8 特殊 I/O 单元重启动位	
	09	#9 特殊 I/O 单元重启动位	
	10	#10 特殊 I/O 单元重启动位	
	11	#11 特殊 I/O 单元重启动位	
	12	#12 特殊 I/O 单元重启动位	
	13	#13 特殊 I/O 单元重启动位	
	14	#14 特殊 I/O 单元重启动位	
	15	#15 特殊 I/O 单元重启动位	
275	00	PC 设置错误(DM6600～DM6605)	
	01	PC 设置错误(DM6613～DM6623)	
	02	PC 设置错误(DM6635～DM6655)	
	03	不用	
	04	修改 RS－232C 设置标志	
	05	不用	
	06～07	系统保留(用户不能存取)	
	08～15	不用	
276	00～07	分(00～59)	以 BCD 码表示当前时间
	08～15	小时(00～23)	
277～279	00～15	键盘映象,见 P383	
280	00～15	组－2 高密度 I/O 单元(单元 0～F)错误标志 (AR0205～AR0214 也有单元 0～9 错误标志的功能)	
281	00～15	特殊 I/O 单元(单元 0～F)重启动位 (单元 0～9 也可用特殊 I/O 单元重启动位 AR0100～AR0109 重启动)	
282	00～15	特殊 I/O 单元(单元 0～F)错误标志 (AR0000～AR0009 也有单元 0～9 错误标志的功能)	
283～286	00～15	通信板监视区域	
287～288	00～15	通信板中断数据区域	
289	00～07	通信板一般监视区域	
	08	通信板端口 A 指令执行标志	
	09～10	由通信板端口 A 指令使用	
	11	通信板端口 A 指令中止位	
	12	通信板端口 B 指令执行标志	
	13～14	由通信板端口 B 指令使用	
	15	通信板端口 B 指令中止位	
290～293	00～15	宏区域输入	
294～297	00～15	宏区域输出	
298～299	00～15	系统保留(用户不能存取)	

3－4－1 SYSMAC NET/SYSMAC LINK 系统

环状态

SR236 提供 SYSMAC NET 系统的局部节点环状态如下所示

	SR236 中的位							
0 级	07	06	05	04	03	02	01	00
1 级	15	14	13	12	11	10	09	08
状态/含义	1	1	中央电源 0:连接 1:不连接	1	环状态 11:正常环 10:向下返回环 01:向上返回环 00:环出错		接收状态 0:允许接收 1:不允许接收	1

完成代码

SR23700～SR23707 提供 SEND/RECV 0 级操作完成代码;SR23708～SR23715 提供 SEND/RECV 1 级操作完成代码。下表给出这些完成代码。

SYSMAC LINK

代 码	项 目	含 义
00	正常结束	处理正常结束。
01	参数错误	网络通信指令参数超出允许范围。
02	不能发送	处理命令过程中单元复位或局部节点不在网络中。
03	目标不在网络中	目标节点不在网络中。
04	占线错误	目标节点正在处理数据,不能接收命令。
05	应答超时	超出响应监控时间。
06	应答错误	从目标节点来的应答中有错误。
07	通信控制器错误	通信控制器出现错误。
08	设置错误	节点地址设置错误。
09	PC 错误	目标节点的 CPU 出现错误。

SYSMAC NET

代 码	项 目	含 义
00	正常结束	处理正常结束。
01	参数错误	网络通信指令参数超出允许范围。
02	路径错误	连接远程网络的路径表中有错误。
03	占线错误	目标节点正在处理数据,不能接收命令。
04	发送错误(丢失令牌)	从线服务器上接收不到令牌。
05	环出错	通信环中出现错误。
06	无应答	不存在目标节点或超出响应监控时间。
07	应答错误	在应答格式中有错误。

数据链接状态标志

SR238～SR245 内含 SYSMAC LINK/SYSMAC NET 系统的数据链接状态。数据结构取决于生成数据链接的系统。

SYSMAC LINK

0 级 操 作	1 级 操 作	位			
		12～15	08～11	04～07	00～03
SR238	SR242	节点 4	节点 3	节点 2	节点 1
SR239	SR243	节点 8	节点 7	节点 6	节点 5
SR240	SR244	节点 12	节点 11	节点 10	节点 9
SR241	SR245	节点 16	节点 15	节点 14	节点 13

最左位 最右位

1:数据链接操作	1:通信错误	1:PC CPU 错误	1:PC 运行状态

SYSMAC NET

0 级 操 作	1 级 操 作	位(下面是节点号)															
		15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
SR238	SR242	8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
SR239	SR243	16	15	14	13	12	11	10	9	16	15	14	13	12	11	10	9
SR240	SR244	24	23	22	21	20	19	18	17	24	23	22	21	20	19	18	17
SR241	SR245	32	31	30	29	28	27	26	25	32	31	30	29	28	27	26	25

1:PC CPU 错误 1:PC 运行状态

3－4－2 远程 I/O 系统

位 SR25312 置“ON”表示在远程 I/O 系统中发生错误。ALM/ERR 指示灯将闪烁。但 PC 操作仍然继续。SR251 含有错误的来源及类型,以及 AR0014、AR0015 含有 SYSMAC LINK 的状态,下面讲述了 SR251 中每一位的功能,详细情况参见《光型和线型远程 I/O 系统手册》。

SR25100－错误检查位

如果不止一个远程 I/O 单元出错,字 251 仅包含第一个错误单元的错误信息,其余单元的数据保存在存储器中,并可以通过置错误检查位“ON”和“OFF”来读取,当查看下一个错误时应确信前一个错误已被记下来,因为后者将被清除。

SR25101 和 SR25102

不用。

SR25103

远程 I/O 错误标志;位 03 置“ON”表示远程 I/O 单元出错。

SR25104～SR25115

位 04－06 表示 3 位二进制数(04:2^0,05:2^1,06:2^2),位 08－15 表示 2 位十六进制数(位 08～11:16^0,位 12～15:16^1)

如果位 12－15 的内容是 B,则表示远程 I/O 主单元或从单元发生了错误。位 08～11 指出主单元的单元号(0 或 1),此时,位 04－06 保存从站机架的单元号。

如果位 12－15 的内容是从 0 至 31 的数,则说明光缆型 I/O 单元或 I/O 终端发生了错误,该数实际上就是光缆型 I/O 单元或 I/O 终端的单元号,如果单元已分配给位 08～15,则位 04 置“ON”,如分配给位 00～07,则位 04 置“OFF”

3－4－3 链接系统标志及控制位

使用下列 SR 位取决于 PC 所在的链接系统的配置。当链接单元(如 PC 链接单元,远程 I/O 单元或上位机链接单元)安装到 PC 机架或 CPU 上时就要使用这些标志及控制位,详细说明可查询有关特定单元的系统手册。

当 PC 并不属于与它们相连的链接系统时,下列位可用作工作位。

上位机连接系统

上位机连接系统具有错误标志及重起动位。错误标志置“ON”时表示上位机链接单元出错。重启动位先置 ON,再置 OFF,使上位机链接单元重新启动。

下表列出了用于上位机链接系统的 SR 位。

机架安装型上位机链接单元重启动位对多重机架安装型上位机链接单元不起作用。详情参见(上位机链接系统手册)。

位	标　志
25206	机架安装型上位机链接单元 1 级错误标志
25207	机架安装型上位机链接单元 1 级重启动位
25213	机架安装型上位机链接单元 0 级重启动位
25311	机架安装型上位机链接单元 0 级错误标志

PC 链接系统

PC 链接单元错误和运行标志

当 PC 属于 PC 链接系统时,字 247～250 用来监视所有连至 PC 链接系统中的 PC 链接单元的操作状态,这包括最大的 32 个 PC 链接单元,如 PC 是在多级 PC 链接系统中,一半的 PC 链接单元在 PC 链接子系统 0 级中操作,另一半在子系统 1 级中操作。位的实际分配取决于 PC 是在单级 PC 链接系统还是在多级 PC 链接系统。详情参见(PC 链接系统手册)。下面讲述错误及运行标志位的分配。

每个字的位 00～07 是运行标志。当 PC 链接单元在运行(RUN)方式时这些位置"ON",位 08～15 是错误标志,当 PC 链接单元出错时,这些位置"ON",下表列出了单级和多级 PC 链接系统的位分配。

单级 PC 链接系统

标志类型	位序号	SR247	SR248	SR249	SR250
运行标志	00	单元 24	单元 16	单元 8	单元 0
	01	单元 25	单元 17	单元 9	单元 1
	02	单元 26	单元 18	单元 10	单元 2
	03	单元 27	单元 19	单元 11	单元 3
	04	单元 28	单元 20	单元 12	单元 4
	05	单元 29	单元 21	单元 13	单元 5
	06	单元 30	单元 22	单元 14	单元 6
	07	单元 31	单元 23	单元 15	单元 7
错误标志	08	单元 24	单元 16	单元 8	单元 0
	09	单元 25	单元 17	单元 9	单元 1
	10	单元 26	单元 18	单元 10	单元 2
	11	单元 27	单元 19	单元 11	单元 3
	12	单元 28	单元 20	单元 12	单元 4
	13	单元 29	单元 21	单元 13	单元 5
	14	单元 30	单元 22	单元 14	单元 6
	15	单元 31	单元 23	单元 15	单元 7

多级 PC 链接系统

标志类型	位序号	SR247	SR248	SR249	SR250
运行标志	00	单元 8 1 级	单元 0 1 级	单元 8 0 级	单元 0 0 级
	01	单元 9 1 级	单元 1 1 级	单元 9 0 级	单元 1 0 级
	02	单元 10 1 级	单元 2 1 级	单元 10 0 级	单元 2 0 级
	03	单元 11 1 级	单元 3 1 级	单元 11 0 级	单元 3 0 级
	04	单元 12 1 级	单元 4 1 级	单元 12 0 级	单元 4 0 级
	05	单元 13 1 级	单元 5 1 级	单元 13 0 级	单元 5 0 级
	06	单元 14 1 级	单元 6 1 级	单元 14 0 级	单元 6 0 级
	07	单元 15 1 级	单元 7 1 级	单元 15 0 级	单元 7 0 级
错误标志	08	单元 8 1 级	单元 0 1 级	单元 8 0 级	单元 0 0 级
	09	单元 9 1 级	单元 1 1 级	单元 9 0 级	单元 1 0 级
	10	单元 10 1 级	单元 2 1 级	单元 10 0 级	单元 2 0 级
	11	单元 11 1 级	单元 3 1 级	单元 11 0 级	单元 3 0 级
	12	单元 12 1 级	单元 4 1 级	单元 12 0 级	单元 4 0 级
	13	单元 13 1 级	单元 5 1 级	单元 13 0 级	单元 5 0 级
	14	单元 14 1 级	单元 6 1 级	单元 14 0 级	单元 6 0 级
	15	单元 15 1 级	单元 7 1 级	单元 15 0 级	单元 7 0 级

应用例子

如 PC 是在多级 PC 链接系统中，并且字 248 的内容是 02FF。那么在 PC 链接子系统 1 级中操作的 PC 链接单元 0～7 处在运行(RUN)方式，而在该系统中 PC 链接单元 1 有错误，下表列出了字 248 的十六进制数的对应的二进制位。

位序号	15……	……	……	00
二进制	0000	0010	1111	1111
十六进制	0	2	F	F

3－4－4 强制状态保持位

SR25211 决定了在编程和监视方式之间切换来进行起动或停止操作时，被强制置位或复位的位状态是否维持，如果 SR25211 为“ON”，则位状态保持，如果 SR25211 为“OFF”，当操作起动或停止时所有位返回到原缺省值。强制状态保持位只有在 PC 设置允许时才有效。

SR25211 位状态不受电源中断影响，除非 I/O 表被登记，在这种情况下，SR25211 将变为“OFF”。

在运行(RUN)方式下，SR25211 不起作用。

对 SR25211 的操作必须通过外围设备，如编程器或 SSS(中文版)。

上电期间维持状态

在 PC 设置中允许强制状态保持位，则在关掉再打开电源时，位 SR25211 及强制置位或复位的位状态可以保持下来，如果允许强制状态保持位，在电源关掉再打开后，位 SR25211 就保持原状态，如果这时 SR25211 是“ON”，强制置位或复位的位状态也被保持下来，如下表所示。

断电前状态	下一次上电时的状态	
SR25211	SR25211	强制置位/复位的位
ON	ON	状态保持
OFF	OFF	复位

注 参见第 3－6－4 节“PC 设置”中有关允许强制状态保持位。

3－4－5 I/O 状态保持位

当在编程方式与监视或运行方式间切换使操作起动或停止时，SR25212 决定 IR 和 LR 区域的位状态是否保持。如果 SR25212 是“ON”，则保持位状态，如 SR25212 是“OFF”，则所有的 IR 及 LR 区域的位被复位，I/O 状态保持位在 PC 设置允许才有效。

SR25212 状态不受电源中断影响，除非 I/O 表被登记，这种情况下，SR25212 变为“OFF”。

对 SR25212 的操作必须通过外围设备，如编程器或 SSS(中文版)。

上电期间维持状态

在 PC 设置中允许 I/O 状态保持位，则在关掉再打开电源时，位 SR25212 及 IR 和 LR 区域的位状态可以保持下来。如果允许 I/O 状态保持位，在电源关掉再打开后位 SR25212 就保持原状态，如果这时位 SR25212 为“ON”，IR 和 LR 区域的位状态也被保持下来，如下表所示。

断电前状态	下一次上电时的状态	
SR25212	**SR25212**	**IR 和 LR 区域**
ON	ON	状态保持
OFF	OFF	复位

注 参见第 3－6－4 节“PC 设置”中有关允许 I/O 状态保持位。

3－4－6 输出关闭位

SR 位 25215 为“ON”时使所有的 PC 输出为“OFF”。此时在 CPU 面板上的 OUT INHIBIT(输出禁止)指示灯亮。当输出关闭位为“OFF”时，按正常方式刷新所有的输出位。

当电源中断或 PC 操作停止时，输出关闭位状态保持。除非 I/O 表被登记，或 I/O 表被登记并且在 PC 设置时没允许强制状态保持位或 I/O 状态保持位。

3－4－7 FAL(故障报警)区域

当 FAL 或 FALS 指令执行时，一个二位的 BCD FAL 代码输出至位 25300～25307，这些代码由用户定

义,用于错误诊断,PC 也把 FAL 代码输出至这些位,比如说由于电池电压过低引起的故障。

执行操作数为 00 FAL 指令或在编程器上执行故障读操作,可以使这个区域复位。

3－4－8 电池低压标志

CPU 后备电池电压下降时,SR 位 25308 置"ON",CPU 面板 ALM/ERR(报警/错误)指示灯闪烁。

SR 位 25308 可以通过编程来产生一个电池电压过低的外部报警。

如果有必要,可以在 PC 设置中使电池报警不允许,详细说明参见第 3－6－4 节"PC 设置"。

3－4－9 周期时间错误标志

当周期时间超过 100ms 时,SR 位 25309 置"ON",CPU 板上的 ALM/ERR(报警/出错)指示灯闪烁,程序执行并不停止,除非超出监视定时器设置的最大时间限制。周期时间超过 100ms 后,计时就不精确了。

3－4－10 I/O 检验错误标志

装在系统上的单元与登记在 CPU 中的 I/O 表不一致时,SR 位 25310 置"ON",CPU 面板上的 ALM/ERR 指示灯闪烁,但 PC 操作仍继续。

一旦该位为"ON"时,为确保正确操作,应停止 PC 操作,检查单元以及修正 I/O 表。

3－4－11 第一个周期标志

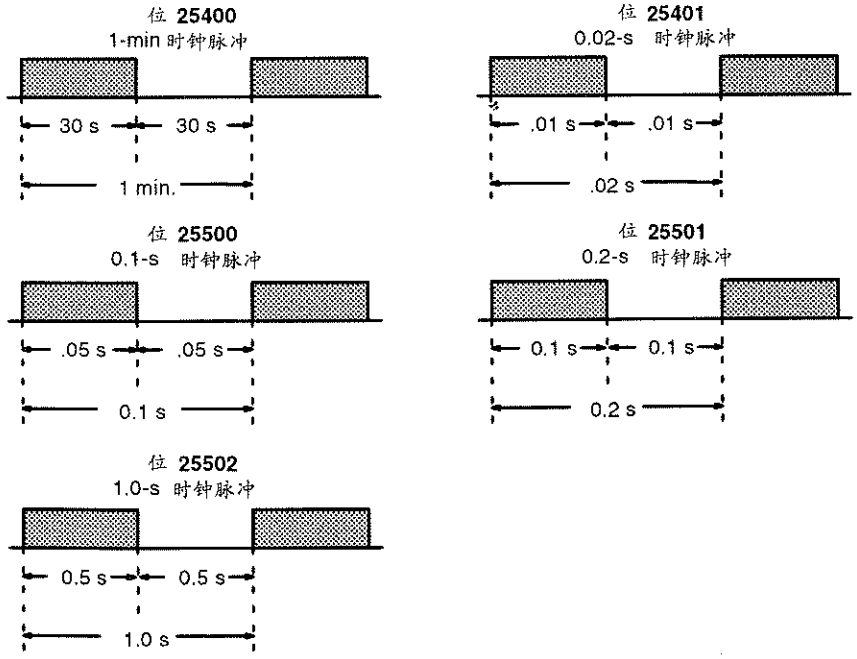
SR 位 25315 在 PC 操作开始时置"ON",一个周期后又置为"OFF":第一周期标志在初始化计数器及其它操作时很有用。第 5－14 节"定时器和计数器指令"中提供了例子。

3－4－12 时钟脉冲位

五个时钟脉冲可用于控制程序计时,每个时钟脉冲位在脉冲的前半部时间内置"ON",后半部时间内置"OFF",换句话说,每个时钟脉冲占空比为 50%。

这些时钟脉冲位常和计数器一起用于建立定时器。例子参见第 5－14 节"定时器和计数器指令"。

脉冲周期	1min	0.02s	0.1s	0.2s	1.0s
位	25400	25401	25500	25501	25502



注 因为 0.1 秒和 0.02 秒时钟脉冲位只有 50 毫秒和 10 毫秒是"ON",故如果程序执行时间过长,CPU 可能不能精确读出脉冲。

3－4－13 单步标志

当用 STEP(08)指令起动单步执行时,SR 位 25407 在一个周期里置“ON”。

3－4－14 组－2 单元错误标志

组 2 高密度 I/O 单元和 B7A 接口单元发生下列错误时 SR 位 25414 置“ON”:同一个 I/O 号设置两次;同一个字分配给一个以上的单元;刷新错误。当发生一个错误时,单元停止工作,同时 ALARM(报警)灯闪烁,但 PC 照常运行。

当组－2 单元错误标志位为 ON 时,相应的出错单元号保存在 AR0205～AR0214。如果正确设置了 I/O 号并适当安装了单元,而仍不能起动单元,则可能是保险丝断了或单元有硬件故障,此时应用备件更换单元,再起动系统。

AR 区域内也有一个高密度 I/O 单元和 B7A 接口单元错误标志位,即 AR0215。

3－4－15 特殊单元错误标志

SR 位 25415 置 ON 表示下列单元中错误:特殊 I/O,PC 链接,上位机链接和远程 I/O 主站单元。下述任何一个错误都会使 SR 位 25415 变 ON。

- 不止一个特殊 I/O 单元设相同单元号。
- 特殊 I/O 单元和 PC 的 CPU 之间刷新数据时出错。
- 上位机链接单元和 PC 的 CPU 之间出错。
- 远程 I/O 主单元上出现错误。

虽然 PC 在 SR25415 为 ON 时继续运行,但出错的单元停止工作,ALM 指示灯闪烁。检查 AR0000～AR0015 状态,找出出错单元的单元号并查找故障起因。

使用重启动位(AR0100～AR0115,SR25207 和 SR25213)可以重新启动单元操作,倘若相同单元号设置给一个以上特殊 I/O 单元,重新启动不起作用。关断电源,纠正单元号设置,再通电重启动。

即使 AR0100～AR0115(重启动位)置 ON,SR25415 仍不会变为 OFF。用编程器读取错误或用梯形程序执行 FAL(06)00 指令,SR25415 才会变 OFF。

3－4－16 指令执行错误标志,ER

如果用不正确的操作数来执行指令,SR 位 25503 就会置“ON”,指令出错的一般原因是应该用 BCD 数据时而没用,或间接寻址的 DM 字不存在。当 ER 标志位置“ON”时,当前指令将不执行。

3－4－17 算术标志

下列标志位用于数据移位、算术运算以及比较指令,一般缩写为两个字母。

注

当执行 END(01)指令时所有这些标志位都将复位,因此不能通过编程设备监视。

详见第 5－15 节“数据移位指令”,第 5－17 节“数据比较指令”,第 5－19 节“BCD 运算指令”以及第 5－20 节“二进制数运算指令”。

负标志(N)

当计算结果是负的,SR 位 25402 置 ON。

上溢出标志(OF)

当二进制加或减的结果超出 7FFF 或 7FFFFFFF 时,SR 位 25404 置 ON。

下溢出标志(UF)

当带符号二进制加或减的结果超出 8000 或 80000000 时,SR 位 25405 置 ON。

进位标志(CY)

当算术运算的结果产生进位或循环/移位指令把“1”移入 CY 时,SR 位 25504 置“ON”。CY 的内容也用于一些算术运算,例如把它与其它操作数一起进行加或减运算,在程序中可以通过设置进位和清除进位指令

设置/清除该标志位。

大于标志(GR)

两数操作比较中当前者大于后者时 SR 位 25505 置“ON”。

等于标志(EQ)

两数操作比较的结果相等时,或算术运算的结果是 0 时,SR 位 25506 置“ON”。

小于标志(LE)

两数操作比较中当前者小于后者时,SR 位 25507 置“ON”。

注 当执行 END(01)指令时,上述四个算术标志置“OFF”。

3－4－18 中断子程序区域

下列区域用于中断处理子程序。

中断子程序最大处理时间区域

SR 位 26200～26215 用来设置中断子程序的最大处理时间。处理时间的增量为0.1ms。

最大处理时间的中断子程序号区域

SR 位 26300～26315 包含最大处理时间的中断子程序号,当有中断时,位 15 置“ON”。

3－4－19 RS－232C 端口通信区域

RS－232C 端口错误代码

当出现 RS－232C 端口错误时,SR 位 26400～26403 置位。

设 置	错误类型
0	正常
1	奇偶校检错误
2	帧错误
3	运行错误
4	FCS 错误
5	超时错误
6	校验和错误
7	命令错误

RS－232C 端口通信错误位

当出现 RS－232C 端口通信错误时,SR 位 26404 置 ON。

RS－232C 端口发送准备标志

当 PC 准备传送数据时,SR 位 26405 置 ON。

RS－232C 端口接收完成标志

当 PC 完成从 RS－232C 设备读取数据时,SR 位 26406 置 ON。

RS－232C 端口接收溢出标志

数据接收后出现数据溢出时,SR 位 26407 置 ON。

RS－232C 接收计数器

SR26500～SR26515 保存 RS－232C 端口在普通 I/O 方式下接收次数。

上位机链接口级发送准备标志

当 PC 准备向上位机链接单元传送时,SR 位 26705 置 ON。

上位机链接 1 级发送准备标志

当 PC 准备上位机链接传送时,SR 位 26713 置 ON。

3－4－20 外设端口通信区域

外设端口错误代码

当出现外设端口错误时普通 I/O 方式,SR 位 26408～26411 置位。

设　置	错误类型
0	正常
1	奇偶校检错误
2	帧错误
3	运行错误
4	FCS 错误
5	超时错误
6	校验和错误
7	命令错误

外设端口通信错误位

当外设端口通信错误时，位 SR26412 置 ON(普通 I/O 方式有效)。

外设端口发送准备标志

当 PC 在普通 I/O 方式下准备传送数据时，SR 位 26413 置“ON”。

外部端口接收完成标志

当 PC 完成从外围设备读数据时，SR 位 26414 置 ON(普通 I/O 方式有效)。

外设端口接收溢出标志

数据接收后如有数据溢出产生，SR 位 26415 置“ON”(普通 I/O 方式有效)。

外设接收计数器

SR26600～SR26615 保存在普通 I/O 方式(BCD)下外设端口接收次数。

上位机链接 0 级发送准备标志

当 PC 准备传送给上位机链接单元时，SR 位 26705 置“ON”。

上位机链接 1 级接收准备标志

当 PC 准备从上位机链接接收数据时，SR 位 26713 置“ON”。

3－4－21　存储器盒区域

存储器盒内容

SR26900～SR26907 显示存储器盒内的存储器类型。

存储器类型	代　码
无	00
UM	01
IOM	02
HIS	03

存储器容量

SR26908～SR26910 显示存储器盒内的存储器容量

容　量	代　码
0K 字(没有安装)	0
4K 字/8K 字	2
16K 字	3
32K 字	4

EEPROM/EPROM 存储器盒安装标志

当 EEPROM 存储器盒处于保护方式或装有 EPROM 存储器盒时，SR 位 26914 置“ON”。

存储器盒标志

当装有存储器盒时，SR 位 26915 置“ON”。

保存 UM 至存储器盒标志

在编程方式中把 UM 数据读至存储器盒时，SR 位 27000 置“ON”。读完该位自动变为“OFF”。如在其它方式中置该位为“ON”，就会出错。

从存储器盒中装入 UM 标志

在编程方式中数据从存储器盒装入 UM 时，SR 位 27001 置 ON。装完该位自动变为“OFF”。如在其它方式中置该位为“ON”，就会出错。

核对(DM 与存储器盒之间)

在 DM 与存储器盒之间比较数据时，SR 位 27002 置“ON”。当一致时，SR 位 27003 置“ON”，不一致时 SR 位 27003 置“OFF”。

3－4－22 数据传送错误位

当有错误时(除帧校验和错误外)，数据就不会从 UM 传送至存储器盒。发生在存储器盒中的校验和错误信息并不输出至 SR272。因为该信息并不需要。如 SR 27015 置“ON”，重复传送。

传送错误标志：

在数据链接运行过程中，准备传送 UM 中 SYSMAC NET 数据链接表时，SR 位 27011 置 ON。

传送错误标志：非编程方式

当 PC 不处于编程方式且准备传送数据时，SR 位 27012 置“ON”。

传送错误标志：只能读

当 PC 处于只读方式且准备传送数据时，SR 位 27013 置“ON”。

传送错误标志：容量不足或无 UM

当准备传送数据而 UM 容量不足时，SR 位 27014 置“ON”。

传送错误标志：帧校验和错误

当准备传送数据而帧校验和出错时，SR 位 27015 置“ON”。

3－4－23 梯形图程序存储器区域

存储器盒梯形图容量区域

SR27100～SR27107 表示存储于存储器盒中的梯形图程序容量。

仅梯形图文件：04：4K 字，08：8K 字，12：12K 字……(32：32K 字)

00：无梯形图程序或无文件。

CPU 梯形图程序容量和类型

SR27108～SR27115 表示 CPU 中梯形图程序容量和类型。

与位 00～07 的规定相同。

3－4－24 存储器错误标志

存储器错误标志：PC 设置错误

当出现 PC 设置校验和错误时，位 SR27211 置“ON”。

存储器错误标志：梯形图校验和错误

当出现梯形图校验和错误时，SR 位 27212 置“ON”。

存储器错误标志：指令改变错误

当指令改变矢量区域出错时，SR 位 27213 置“ON”。

存储器错误标志:存储器断开错误

当存储器盒在操作期间连接或断开,SR 位 27214 置“ON”。

存储器错误标志:自动引导错误

当发生自动引导错误时,SR 位 27215 置“ON”。

3－4－25 数据保存标志

在编程方式下该位置 ON 时数据传送至存储器盒。该位自动变为“OFF”,在其它方式下置该位为“ON”,就会出错。

保存 IOM 至存储器盒位

当 IOM 保存至存储器盒时,SR 位 27300 置“ON”。

从存储器盒装入 IOM 位

当从存储器盒中装入 IOM 时,SR 位 27301 置“ON”。

3－4－26 传送错误标志

当发生错误时(除只读错误外),数据不会从 IOM 传送到存储器盒。

传送错误标志:非编程方式

当在非编程方式准备传送数据时,SR 位 27312 置“ON”。

传送错误标志

当在只读方式下准备传送数据时,SR 位 27313 置“ON”。

传送错误标志

准备传送数据而 IOM 容量不足时,SR 位 27314 置“ON”。

3－4－27 PC 设置错误标志

PC 设置起动错误

发生 PC 设置起动错误时(DM6600～DM6605),SR 位 27500 置“ON”。

PC 设置运行错误

发生 PC 设置运行错误时(DM6613～DM6623),SR 位 27501 置“ON”。

PC 设置通信/错误设置/混杂错误

当 PC 设置通信、错误设置或混杂错误发生时(DM6635～DM6655),SR 位 27502 置“ON”。

3－4－28 时钟和键盘映象

时钟(SR276)

SR276 包含当前时间。分钟(0－59)存在 SR 位 27600～27607,小时(0－23)存在 SR 位 27608～27615。

键盘映象(SR277)

SR277～SR279 用作键盘映象。

3－4－29 组－2 错误标志

SR 位 28000～28015 用作组－2 高密度 I/O 单元(单元号 0～F)的错误标志。当单元出错,相应的错误标志置 ON。在 AR 区域中 10 个位(AR0205～AR0214)也用作单元 0～9 的错误标志。

3－4－30 特殊 I/O 单位重启动位和错误标志

SR 位 28100～28115 用作特殊 I/O 单元(单元号 0～F)的重起动位。重新起动 1 个特殊 I/O 单元,先置相应的位 ON,再置 OFF。在 AR 区域中 10 个位(AR0100～AR0109)也用作单元 0～9 的重起动位。

SR 位 28200～28215 用作特殊 I/O 单元(单元号 0～F)的错误标志。当单元出错,相应的错误标志置 ON。在 AR 区域中 10 个位(AR0000～AR0009)也用作单元 0～9 的错误标志。

3-5 AR(辅助继电器)区域

AR 字的地址从 AR00～AR27;AR 位的地址从 AR0000～AR2715。大多数 AR 区域的字和位都有特定的用处,例如传输计数器,标志及控制位,而且字 AR00～AR07 以及 AR23～AR27 不能用于其它用途。AR08～AR17 的字或位如果不用于下列目的,可以作为工作字或工作位。

字	用 途
AR08～AR15	SYSMAC LINK 单元
AR16, AR17	SYSMAC LINK 和 SYSMAC NET 链接单元

当电源中断,或从监视/运行方式切换至编程方式或当 PC 操作停止时,AR 区域能保持状态,下表中列出了位分配,并在下面按位顺序详细讲述了每个位。

AR 区标志和控制位

字	位	功 能
00	00～09	特殊 I/O 单元 0～9 和 PC 链接单元 0～9 错误标志 (这些标志的功能和 SR28200～SR28209 的相同。)
	10	SYSMAC LINK 或 SYSMAC NET 链接系统 1 级操作错误标志
	11	SYSMAC LINK 或 SYSMAC NET 链接系统 0 级操作错误标志
	12	上位机至机架安装型上位机链接单元(1 级)错误标志
	13	上位机至机架安装型上位机链接单元(0 级)错误标志
	14	远程 I/O 主单元 1 错误标志
	15	远程 I/O 主单元 0 错误标志
01	00～09	特殊 I/O 单元 0～9 和 PC 链接单元 0～9 的重起动位 (这些位的功能和 SR28100～SR28109 的相同。)
	10	SYSMAC LINK 或 SYSMAC NET 链接系统 1 级操作重起动位
	11	SYSMAC LINK 或 SYSMAC NET 链接系统 0 级操作重起动位
	12、13	不用
	14	远程 I/O 主单元 1 重起动标志
	15	远程 I/O 主单元 0 重起动标志
02	00～04	从站机架错误标志(0～4)
	05～14	组－2 错误标志(位 05～14 对应于单元 0～9。)
	15	组 2 错误标志
03	00－15	光缆型 I/O 单元 I/O 终端 0－7 错误标志
04	00－15	光缆型 I/O 单元 I/O 终端 8－15 错误标志
05	00－15	光缆型 I/O 单元 I/O 终端 16－23 错误标志

字	位	功　能
06	00－15	光缆型 I/O 单元 I/O 终端 24－31 错误标志
07	00～03	SYSMAC LINK 系统 0 级操作数据链接设置
	04～07	SYSMAC LINK 系统 1 级操作数据链接设置
	08	TERMINAL(终端)方式输入取消位
	09～11	不用。
	12	同 CPU 上 DIP 开关插脚 6 的状态
	13	错误记录覆盖位
	14	错误记录重启动位
	15	错误记录允许位
08～11	00～15	0 级操作的 SYSMAC LINK 系统节点中有效节点标志
12～15	00～15	1 级操作的 SYSMAC LINK 系统节点中有效节点标志
16	00～15	SYSMAC LINK/SYSMAC NET 链接系统 0 级操作每个循环服务时间
17	00～15	SYSMAC LINK/SYSMAC NET 链接系统 1 级操作每个循环服务时间
18	00～07	秒:00～59
	08～15	分钟:00～59
19	00～07	小时:00～23(24 小时制)
	08～15	日:01～31(根据月份和闰年调整)
20	00～07	月:1～12
	08～15	年:00～99(年份的右 2 位)
21	00～07	星期:00～06(00:星期天;01:星期一;02:星期二;03:星期三;04:星期四;05:星期五;06:星期六)
	08～12	不用
	13	30 秒补偿位
	14	时钟停止位
	15	时钟设置位
22	00～15	键盘映象
23	00～15	电源关闭计数器(BCD)
24	00	SYSMAC LINK－RS－232C 外围设备标志
	01	SYSMAC LINK－端口 A 外围设备标志
	02	SYSMAC LINK－端口 B 外围设备标志
	03	SYSMAC LINK－外围设备初始化位
	04	不用
	05	周期时间标志
	06	1 级操作 SYSMAC LINK 系统网络参数标志
	07	0 级操作 SYSMAC LINK 系统网络参数标志
	08	SYSMAC－SYSMAC NET 链接单元 1 级安装标志
	09	SYSMAC－SYSMAC NET 链接单元 0 级安装标志
	10～12	不用。
	13	机架安装型上位机链接单元 1 级安装标志
	14	机架安装型上位机链接单元 0 级安装标志
	15	CPU 安装型器件安装标志

字	位	功　能
25	00～07	存取在线编辑不允许位的口令 (该字节包含 5A 时,在线编辑不允许位有效。)
	08	FPD(一)触发位
	09	在线编辑不允许位
	10	在线编辑等待标志
	11	不用
	12	跟踪结束标志
	13	跟踪标志
	14	跟踪触发位(可写)
	15	跟踪开始位(可写)
26	00～15	最大周期时间(0.1ms)
27	00～15	当前周期时间(0.1ms)

3－5－1　特殊 I/O 单元重起动

AR 位 0100～0109 对应特殊 I/O 单元号 0～9。重启动特殊 I/O 单元(包括 PC 链接单元),置相应的位 ON 和 OFF(或接通和断开电源)。在特殊 I/O 单元重新起动过程中,不要存取刷新的数据(见 34 页上 SR27400～SR27409)。

注

SR 位 28100～28115 还用作特殊 I/O 单元 O～F 的重启动位。

3－5－2　从站机架错误标志

AR 位 0200～AR0204 对应于远程 I/O 从站单元＃0～＃4。如果同一个号同时分配给一个以上的从站或当起动系统时发生传输错误,这些位将置“ON”,系统正常起动后发生错误,可参考 SR251。

3－5－3　组－2 错误标志

AR 位 0205～AR0215 对应于组 2 高密度 I/O 单元 0～9 和 B7A 接口单元(I/O 号)。当同一号分配给一个以上的单元,或同一个字分配给一个以上的单元,或设置一个 64 点单元的 I/O 号为 9,或在晶体管高密度 I/O 单元中保险丝烧断了,这些位将置“ON”,当单元不作为组 2 高密度 I/O 单元时,AR 位 0215 置“ON”。

注

SR 位 28000～28015 也用作组－2 高密度 I/O 单元(单元号 O～F)的错误标志。

3－5－4　光缆型 I/O 单元和 I/O 终端错误标志

AR03～AR06 保存光缆型 I/O 单元和 I/O 终端的错误标志,错误指示多重设置单元号。多达 64 个光缆型 I/O 单元可连至 PC,根据单元号 0～31 和字母 L 或 H 来区分单元。下表列出了字的分配。

光缆型 I/O 单元和 I/O 终端错误标志:

位	AR03 分配	AR04 分配	AR05 分配	AR06 分配
00	0L	8L	16L	24L
01	0H	8H	16H	24H
02	1L	9L	17L	25L
03	1H	9H	17H	25H
04	2L	10L	18L	26L
05	2H	10H	18H	26H
06	3L	11L	19L	27L
07	3H	11H	19H	27H
08	4L	12L	20L	28L

（续表）

位	AR03 分配	AR04 分配	AR05 分配	AR06 分配
09	4H	12H	20H	28H
10	5L	13L	21L	29L
11	5H	13H	21H	29H
12	6L	14L	22L	30L
13	6H	14H	22H	30H
14	7L	15L	23L	31L
15	7H	15H	23H	31H

3－5－5 SYSMAC LINK 系统数据链接设定

AR0700－AR0703 和 AR0704－AR0707 被用于指定 SYSMAC LINK 系统中 0 级操作和 1 级操作的字分配，分配或者按 SSS(中文版)的设置或者按在 LR 和/或 DM 区的自动设置进行。如果指定成自动分配，字的编号被指定分配到每一个节点。这些设置表示如下：

外部/自动分配。

0 级操作		1 级操作		设 置	
AR0700	AR0701	AR0704	AR0705		
0	0	0	0	外部设置字(SSS(中文版))	
1	0	1	0	自动分配	仅 LR 区
0	1	0	1		仅 DM 区
1	1	1	1		LR 和 DM 区

每节点的字

如果是自动指定上述分配，下面的设置是必要的。

0 级操作		1 级操作		每节点的字		最大节点数
AR0702	AR0703	AR0706	AR0707	LR 区	DM 区	
0	0	0	0	4	8	16
1	0	1	0	8	16	8
0	1	0	1	16	32	4
1	1	1	1	32	64	2

在 SYSMAC LINK 系统工作时，上述的设置每个周期都要读。

3－5－6 错误记录位

用户可通过设置 AR0713(错误记录覆盖位)为“ON”或“OFF”来控制 DM 区域错误记录区改写。置 AR0713 为“ON”时，每次出错记录计满 10 后，则覆盖最早的记录。当 AR0713 为“OFF”时，只记录清除错误记录后的前十个错误。

用户可通过置 AR 位 0714(错误记录复位位)为“ON”或“OFF”来复位错误记录指针(DM6000)从而可在记录区的开始重新记录错误。

用户置 AR0715(错误记录允许位)为 ON 时，允许错误记录存储，为“OFF”时，不允许存储错误记录。

详情参见第 3－6 节 DM 区域的错误记录区域。

错误记录位每一个周期被刷新。

3－5－7 有效节点标志

AR08－AR11 和 AR12－AR15 提供了表示当前 SYSMAC LINK 系统中哪些是有效节点的标志。SYS-

MAC LINK 系统工作时这些标志每个周期都刷新。

下表的内容表示了分配到每位的节点号。如果位为 ON,这个节点现时是有效的。

0 级	1 级	位(表中指示节点号)															
		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
AR08	AR12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
AR09	AR13	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
AR10	AR14	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
AR11	AR15	49	50	51	52	53	54	55	56	57	58	59	60	61	62	*	* *

* 通信控制器错误标志

* * EEPROM 错误标志

3－5－8 SYSMAC LINK/SYSMAC NET 链接系统服务时间

当 SYSMAC LINK 单元和/或 SYSMAC NET 链接单元安装到机架上,每一周期里 AR16 都提供分配给 0 级 SYSMAC LINK 系统和/或 SYSMAC NET 链接系统的工作时间。

当 SYSMAC LINK 单元和/或 SYSMAC NET 链接单元安装到机架上,每一周期里 AR 17 都提供分配给 1 级 SYSMAC LINK 系统和/或 SYSMAC NET 链接系统的工作时间。

这些时间以 4 位 BCD 码记录,精确到十分之一毫秒(000.0－999.9ms),每个周期都要刷新。

位			
15－12	11－08	07－04	03－00
10^2	10^1	10^0	10^{-1}

3－5－9 日历/时钟区域和位

日历/时钟区域

C200HX/HG/HE CPU 内置时钟。如果 AR2114(时钟停止位)为 OFF,日期、天和时间将有效,并以 BCD 码形式存储在 AR18～AR20 以及 AR2100～AR2108,如下表所示。这一区域也可以通过 AR2113(30 秒补偿位)和 AR2115(时钟设置位)来控制。

日历/时钟位

位	内 容	允 许 值
AR1800～AR1807	秒	00～59
AR1808～AR1815	分钟	00～59
AR1900～AR1907	小时	00～23(24 小时制)
AR1908～AR1915	日	01～31(根据月和闰年调整)
AR2000～AR2007	月	01～12
AR2008～AR2015	年	00～99(年份中右二位)
AR2100～AR2107	星期	00～06(00:星期天;01:星期一;02:星期二;03:星期三;04:星期四;05:星期五;06:星期六)

30 秒补偿位

AR2113 为"ON"时,将日历/时钟区的秒数"四舍五入"。例如,当时钟的秒数为 29 或更小时,就置 00 秒,如秒数为 30 或大于 30 时,则在分钟数上加 1,而置秒数为 00。

时钟停止位

AR2114 为"OFF"时,允许日历/时钟区域工作,而为"ON"时停止时钟工作。

时钟设置位

AR2115 用来设置日历/时钟区域,如下所述。数据必须是 BCD 码并且在上表规定的范围内。

1. 置 AR2114(停止位)为 ON。

2. 设置所需的日期和时间,当设置星期时,要小心不能置 AR2114(时钟停止位)为"OFF"(它们在同一字内)。(在编程器上,用位/数字监视和强制置位/复位操作来设置这些数据是最简便的方法)

3. 置 AR2115(时钟设置位)为 ON,根据已设置值,日历/时钟自动开始工作。而 AR2114 和 AR2115 都置"OFF"。

工作时日历/时钟区域和位每周期刷新一次。

3－5－10 终端方式键位

如果编程器安装在 PC 上,并处于 TERMINAL(终端)方式,任何一个键 0～9(包括字母 A～F,即用 SHIFT 键和 0～5)的输入都将置 AR22 中相应的位为 1,通过编程器操作可进入 TERMINAL(终端)方式。

AR22 中的位对应编程器上输入如下:

位	编程器输入	位	编程器输入
AR2200	0	AR2208	8
AR2201	1	AR2209	9
AR2202	2	AR2210	A
AR2203	3	AR2211	B
AR2204	4	AR2212	C
AR2205	5	AR2213	D
AR2206	6	AR2214	E
AR2207	7	AR2215	F

有关 TERMINAL(终端)方式的细节,参见第七章"程序监视和执行"。

3－5－11 电源关闭计数器

AR23 用 4 位 BCD 码记录 PC 电源关闭的次数。如有必要,可从编程器中通过 PV 改变 1 操作复位这个计数器。(详见第 7－2－4 节"十六进制/BCD 码数据修改")每次电源接通时,刷新电源关闭计数器。

3－5－12 SYSMAC LINK－外围设备标志

每次只能从一个端口通过 SYSMAC LINK 系统使用 1 个外围设备。要改变外围设备正在使用的端口时,先要置 SYSMAC LINK－外围设备初始化位(AR2403)为 ON。

位	功 能
AR2400	SYSMAC LINK－RS－232C 外围设备标志 当 SYSMAC LINK 系统通过 RS－232C 端口连接正使用一个外围设备时,该标志置 ON。)
AR2401	SYSMAC LINK－端口 A 外围设备标志 (当 SYSMAC LINK 系统通过通信板端口 A 连接正使用一个外围设备时,该标志置 ON。)
AR2402	SYSMAC LINK－端口 B 外围设备标志(当 SYSMAC LINK 系统通过通信板端口 B 连接正使用一个外围设备时,该标志置 ON。)
AR2403	SYSMAC LINK－外围设备初始化位 (该位置 ON,对 SYSMAC LINK 使用外围设备进行初始化)

3－5－13 周期时间标志

当用 SCAN(18)设置的周期时间小于实际的周期时间,AR2405 为"ON"。

PC 在运行或监控方式下每个周期都刷新 AR2405。

3－5－14 链接单元安装标志

下列标志表示特定的链接单元安装在机架上。(对 CPU 安装型上位机链接单元参见第 3－5－15 节

"CPU 安装型设备安装标志")。这些标志每个周期刷新一次。

名　称	位	链接单元
SYSMAC LINK/SYSMAC NET 链接单元 1 级安装标志	AR2408	SYSMAC LINK/SYSMAC NET 链接单元 1 级操作
SYSMAC LINK/SYSMAC NET 链接单元 0 级安装标志	AR2409	SYSMAC LINK/SYSMAC NET 链接单元 0 级操作
机架安装型上位机链接单元 1 级	AR2413	机架安装型上位机链接单元 1 级操作
机架安装型上位机链接单元 0 级	AR2414	机架安装型上位机链接单元 0 级操作

3－5－15　CPU 安装型设备安装标志

当有设备直接安装至 CPU 时，AR2415 置 ON。这包括 CPU 安装型上位机链接单元，编程器以及接口单元。这个标志位每个周期刷新一次。

3－5－16　FPD 触发位

AR2508 用来自动调整 FPD(一)的监视时间，详情参见第 5－25－12 节"故障点检测指令－FPD(一)。"

3－5－17　数据跟踪标志及控制位

用 TRSM(45)进行数据跟踪时，要用到下列控制位及标志。在跟踪操作期间，跟踪标志置 ON，当被跟踪的数据已占满跟踪区时，跟踪完成标志置 ON。

位	名称
AR2512	跟踪完成标志
AR2513	跟踪标志
AR2514	跟踪完成标志(可写)
AR2525	采样起动位(可写)

注　参见第 5－25－3 节"跟踪存储器采样指令－TRSM(45)"。

3－5－18　周期时间指示字

AR26 保存从程序执行开始以来的最大周期时间，AR27 保存当前的周期时间。

两个时间为以 0.1ms 为单位的 4 位 BCD 数，0000ms～999.9ms，并且每周期刷新一次。

3－6　DM(数据存储器)区域

DM 区域被分成如下表所示的各个部分。一部分 UM(多达 3000 字，以 1000 字为单位)可分配扩展 DM。

地　址	用户读/写	用　途
DM0000－DM0999	读/写	普通 DM
DM1000－DM1999		特殊 I/O 单元区域
DM2000－DM5999		普通 DM
DM6000－DM6030		历史记录
DM6100－DM6143		保留
DM6144－DM6599	只读	系统设置
DM6600－DM6655		PC 设置
DM7000－DM9999		扩展 DM[2]

注　1. PC 设置可以设置用 DM7000～DM8599 作为特殊 I/O 区域来代替 DM1000～DM2599，详情参见第 3－6－4 节"PC 设置"。

2．UM 区域分配编程器操作可用来分配多达 3000 字的 UM 作为扩展 DM。

尽管 DM 区域和其它数据区域一样，每字包含 16 位，但 DM 区域的数据不能通过位定义用于位操作数的指令。通过程序可以写 DM0000～DM6143。但 DM6144～DM6655 只能从外围设备改写，如编程器或带 SSS(中文版)的上位计算机。

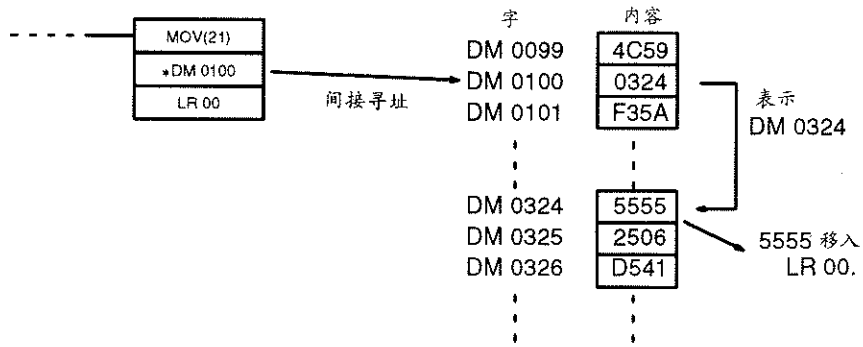
掉电时 DM 区域的状态可以保留下来。

间接寻址

一般地，当数据区域中字的内容定义给指令时，该指令可直接对该字的内容进行操作。例如，假设指令 MOV(21)把 DM0100 作为第一个操作数，把 LR20 作为第二个操作数，则当该指令执行时，DM0100 的内容移入 LR20。

注 扩展 DM 不能用于间接寻址。

然而，许多指令的操作数能使用间接 DM 地址。为指明间接 DM 地址，输入＊DM 操作数地址。用间接地址时，操作数内容不是实际所有的数据，它的内容为另一个 DM 字的地址，该地址中的内容才真正地用于指令。如在上例中采用＊DM0100，并且 DM0100 的内容是 0324，则＊DM0100 表示指令的操作数是 DM0324 中的内容，DM0324 的内容将移至 LR00。



3－6－1 扩展 DM 区域

设置扩展 DM 区域用来存储操作参数及链接单元和特殊 I/O 单元的其它操作数据。通过编程器或 SSS(中文版)的 UM 分配操作，多达 3000 字的 UM 可用作扩展 DM(以 1K 字为单位)，扩展 DM 区域的地址范围是 DM7000～DM9999。

起动 PC 或通过编程指令可将扩展 DM 区域数据传送至特殊 I/O 单元缺省区域(DM1000～DM1999)，这样就很容易改变操作参数，实现控制过程间的快速切换。扩展 DM 区还可用来存储连接至 PC 系统的其它设备的参数。例如可编程终端字符串或数字表。

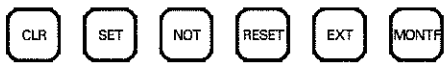
扩展 DM 区域用来存储操作参数，但不能象普通 DM 区域用于编程。扩展 DM 只能通过外围设备改写，在电源中断期间保持状态，且不能用于间接寻址。

UM 区域可以 1K 字为单元作为扩展 DM 区分配。一旦建立了扩展 DM 区域，就可作为程序的一部分进行保存和传送，即保存或传送程序时不需特别的过程。

UM 分配操作

下面列出了用编程器的 UM 分配操作的过程。参见第 4－6－3 节“清除内存储器”中的 DATA CLEAR(数据清除)和 UM ALLOCATION(UM 分配)指令。

1．清除存储器



注　除非先清除存储器，否则不能分配 UM。

2. 扩展 DM 区域可设置成 0，1，2 或 3K 字，下列键操作顺序生成一个 2K 字的扩展区 DM 区域(DM7000～DM8999)。



按 0 键取消扩展 DM 区域(0KW)。

或按 1 键分配 DM7000～DM7999(1KW)。

或按 2 键分配 DM7000～DM8999(2KW)。

或按 3 键分配 DM7000～DM9999(3KW)。

3－6－2　特殊 I/O 单元数据

按 PC 设置中字 DM6602 设置的值，DM 区域的 1000 个字或 1600 个字分配给特殊 I/O 单元。DM6602 设定值决定特殊 I/O 单元数据区域设成 10 单元还是 16 单元以及数据存储在读/写 DM 区(DM1000～DM2599)还是只读 DM 区(DM7000～DM8599)。

单元	地　址
0	DM 1000～DM 1099 或 DM 7000～DM 7099
1	DM 1100～DM 1099 或 DM 7100～DM 7199
2	DM 1200～DM 1099 或 DM 7200～DM 7299
3	DM 1300～DM 1099 或 DM 7300～DM 7399
4	DM 1400～DM 1099 或 DM 7400～DM 7499
5	DM 1500～DM 1099 或 DM 7500～DM 7599
6	DM 1600～DM 1099 或 DM 7600～DM 7699
7	DM 1700～DM 1099 或 DM 7700～DM 7799
8	DM 1800～DM 1099 或 DM 7800～DM 7899
9	DM 1900～DM 1099 或 DM 7900～DM 7999
A	DM 2000～DM 1099 或 DM 8000～DM 8099
B	DM 2100～DM 1099 或 DM 8100～DM 8199
C	DM 2200～DM 1099 或 DM 8200～DM 8299
D	DM 2300～DM 1099 或 DM 8300～DM 8399
E	DM 2400～DM 1099 或 DM 8400～DM 8499
F	DM 2500～DM 1099 或 DM 8500～DM 8599

注　当 DM 字不分配给特殊 I/O 单元可用作其它用途。

3－6－3　错误记录区域

DM6000～DM6030 用于存储 PC 出错的 10 个记录，显示 PC 中发生错误的性质、时间和日期。

只要 AR0715(错误记录允许位)为“ON”时，错误记录区将存储由系统或 FAL(06)/FALS(07)产生的错误代码，有关错误代码详见第九章“故障排除”。

区域结构

每个错误记录占用位于 DM6001～DM6030 中的三个字，可以通过 DM6000(错误记录指针)的内容获取所存的最后的一个记录。下面列出了 10 个记录中每一个记录号、DM 字、指针值。

记　录	地　　址	指　针　值
无	无	0000
1	DM6001～DM6003	0001
2	DM6004～DM6006	0002
3	DM60007～DM6009	0003
4	DM6010～DM6012	0004
5	DM6013～DM6015	0005
6	DM6016～DM6018	0006
7	DM6019～DM6021	0007
8	DM6022～DM6024	0008
9	DM6025～DM6027	0009
10	DM6028～DM6030	000A

尽管每个记录包含不同的内容,但它们的结构是一样的:第一字包含错误代码,第二、三字表示日期和时间,错误代码由系统产生或由 FAL(06)/FALS(07)产生,时间和日期来自 AR18 和 AR19(日历/日期区域),错误代码记录还指出是致命还是非致命的。记录结构如下:

字	位	内　　容
第 1 字	00～07	错误代码
	08～15	00(非致命的)或 80(致命的)
第 2 字	00～07	秒
	08～15	分
第 3 字	00～07	小时
	08～15	日

下表列出了可能的错误代码及对应的错误。

错误严重性	错误代码	错　　误
严重的错误	01～99 或 9F	系统错误(FALS)
	C0～C2	I/O 总线错误
	E0	输入输出 I/O 表错误
	E1	单元过多
	F0	没有 END(01)指令
	F1	存储器错误
非严重错误	01～99	系统错误(FAL)
	8A	中断输入错误
	8B	中断程序错误
	9A	组 2 高密度 I/O 错误
	9B	PC 设置错误
	9C	通信板错误
	9D	UM 存储器盒传送错误
	B0～B1	远程 I/O 错误
	D0	特殊 I/O 错误
	E7	I/O 表校检错误
	F7	电池错误
	F8	周期时间超时

操作

当 AR0715(错误记录允许位)为 ON,产生第一个错误代码时,相应的数据存至记录指针所指的记录中

(初始记录为1),然后指针增1,以后产生的错误代码依次放入连续的记录中一直到最后一个记录用完。此后如再出错,则错误记录的处理取决于AR0713的状态(错误记录覆盖位)。

如果AR0713为"ON"且指针为000A,则下一个错误代码写入记录10,而记录10的内容移向记录9,……,依次下移,直到记录1的内容被移走丢失。这个区域的功能看起来象移位寄存器,而记录指针一直停留在000A。

如果AR0713为"OFF"且指针为000A,则错误记录中的内容保持不变,以后产生的任何错误不被记录,除非AR0713置"ON"或把错误记录区复位。

先置AR0714(错误记录复位位)为"ON",然后再置"OFF"则可把错误记录区复位。此时指针为0000,错误记录区被复位(清除),从错误记录区起始处记录以后的错误代码。只有当AR0715(错误允许位)为"ON"时才可复位错误记录区。

3－6－4 PC设置

PC设置(DM6600～DM6655)包括决定PC操作的设置,PC设置的数据能通过编程器或SSS(中文版)来改变,这必须在UM不是写保护(CPU的DIP开关插脚1)的条件下实现。有关改变DIP开关插脚设定细节参见19页。

DM6600～DM6634中数据只有当PC处于编程方式下才能设置或修改。当PC处于编程或监控模式时,可以设置或修改DM6635～DD6655中数据。下列字通过SYSMAC支持软件中PC设置菜单修改。(PC必须处于编程方式。)

1. 启动方式(DM6600)
2. 启动方式指定(DM6601)
3. 周期监视时间(DM6618)
4. 周期时间设置(DM6619)
5. RS－232C端口设置(DM6645～DM6649)
6. 外设端口设置(DM6650～DM6654)

PC能在缺省PC设置下工作,只有当用户改变PC操作环境满足应用需要时才变更PC设置。下表列出了PC设置的参数。

字	位	功　能	缺省值
起动处理(DM6600～DM6612),下面的设置只当PC通电时才存取一次。			
DM6600	00～07	启动方式(当位08～15设置成02时有效) 00:PROGRAM(编程): 01:MONITOR(监控) 02:RUN(运行)	编程状态
	08～15	启动方式指定 00:编程器开关 01:电源关闭前所用的连续操作方式 02:在00位～07位中设置	编程器开关
DM6601	00～07	保留	…
	08～11	IOM保持位(SR25212)状态 0:清除;1:维持	清除
	12～15	强制状态保持位(SR25211)状态 0:清除;1:维持	

字	位	功　能	缺省值
DM6602	00～07	不用	…
	08～15	特殊 I/O 单元区域(详见 3－4－8 特殊 I/O 单元区域设置) 00:C200H 兼容的 RAM 方式(缺省)… DM1000～DM2599 用作特殊 I/O 单元区域。 01:C200H 兼容的 ROM 方式 1 启动时,DM7000～DM7999 的内容传送到 DM1000～DM1999 并使用 DM1000～DM1999。 02:DM 线性方式 1 DM7000～D7999 用作特殊 I/O 单元区域。 11:C200H 兼容的 ROM 方式 2 启动时,DM7000～DM8599 的内容传送到 DM100～DM2599 并使用 DM1000～DM2599。 12:DM 线性方式 2 DM7000～DM8599 用作特殊 I/O 单元区域。	DM1000～DM2599
DM6603～DM6604	00～15	不用	……
DM6605	00～07	瞬时断电时间(0－10ms) 设置瞬时断电时间 00～10,BCD 码	0ms
	08～15	不用	…
DM6606～DM6612	00～15	不用	…
通信和周期时间设置(DM6613～DM6619) 下面的设置只在程序开始执行时存取一次。			
DM6613	00～07	通信板端口 B 的服务时间 (当位 08～15 设为 01 时有效) 00～99(BCD 码):使用端口 B 服务占循环时间的百分比。 最小:0.228ms;最大:58.254ms	没有设置(0000)
	08～15	通信板端口 B 允许服务设置 00:不设服务时间(固定 5%,最小 0.228ms) 01:使用位 00～07 中时间。 当操作停止时,服务时间 10ms,与设置无关。	
DM6614	00～07	通信板端口 A 的服务时间 (当位 08～15 设为 01 时有效) 00～99(BCD 码):使用端口 A 服务占循环时间的百分比。 最小:0.228ms;最大:58.254ms	没有设置(0000)
	08～15	通信板端口 A 允许服务设置 00:不设服务时间(固定 5%,最小 0.228ms) 01:使用位 00～07 中时间。 当操作停止时,服务时间 10ms,与设置无关。	
DM6615	00～15	保留	……

字	位	功　能	缺省值
DM6616	00～07	RS-232C端口的服务时间 (当位08～15设为01时有效) 00～99(BCD码):使用RS-232C端口服务占循环时间的百分比。 最小:0.228ms;最大:58.545ms	没有设置(0000)
	08～15	RS-232C端口允许服务设置。 00:不设服务时间(固定5%,最小0.228ms) 01:使用位00～07中时间。 当操作停止时,服务时间10ms,与设置无关。	
DM6617	00～07	外设端口的服务时间 (当位08～15设为01时有效) 00～09(BCD码):使用外设端口服务占循环时间的百分比。 最小:0.228ms;最大:58.254ms	没有设置(0000)
	08～15	外设端口允许服务设置。 00:不设服务时间(固定5%,最小0.228ms) 01:使用位00～07中时间。 当操作停止时,服务时间10ms,与设置无关。	
DM6618	00～07	循环监控时间(当位08～15设为01,02或03时有效) 00～99(BCD码)x设置单位(见位08～15)	00
	08～15	循环监控允许(位00～07)中设置×设置单元;最大99秒) 00:120ms(位00～07中设置无效) 01:设置单位10ms 02:设置单位100ms 03:设置单位1s	00:120ms
DM6619	00～15	循环时间 0000:可变(无最小时间) 0001～9999(BCD码):最小时间,单位ms	可变
中断/刷新处理(DM6620～DM6623) 下面的设置只在程序开始执行时存取一次			
DM6620	00～09	特殊I/O单元循环刷新(位号对应单元号,包括PC链接单元) 0:允许主程序中循环刷新和I/O刷新(IORF(97)) 1:不允许(仅允许中断程序中I/O刷新) 01设置(不允许)仅当中断响应设为高速响应模式时才有效,对正常中断响应或安装在从站机架上的特殊I/O单元无效。	允许

字	位	功　能	缺 省 值
DM6620	10～11	保留	…
	12～15	中断响应 0:正常((C200H 兼容) 在上位机链接服务,指令执行,远程 I/O 处理,或特殊 I/O 处理过程中,中断不接收。在上述处理完成后执行中断子程序。 1:高速响应(C200HS 或 C200HX/HG/HE)在上位机链接,指令执行,远程 I/O 处理,或特殊 I/O 处理过程中,接收中断。如有中断输入,当前的处理中断,执行中断子程序。	正常
DM6621	00～07	保留	…
	08～15	特殊 I/O 单元刷新(包括 PC 链接单元) 00:允许所有特殊 I/O 单元刷新 01:不允许所有特殊 I/O 单元刷新(但对从站机架上单元不起作用) 01 设置(不允许)对安装在从站机架上的特殊 I/O 单元无效	允许
DM6622	00～07	定时中断时间单位 00:10ms 01:1ms	10ms(0000)
	08～15	定时中断时间单位允许 00:不允许(10ms) 01:允许位 00～07 中设置	
DM6623	00～15	特殊 I/O 单元循环刷新(包括 PC 链接单元) (位号 00～15 对应单元号 O～F) 0:允许主程序中循环刷新和 I/O 刷新 (IORF(97) 1:不允许(仅允许中断程序中 I/O 刷新)01 设置(不允许)只有当中断响应设为高速响应方式时才有效,对正常中断响应或安装在从站机架上特殊 I/O 单元无效。	允许
DM6624～DM6644	00～15	保留	…
RS－232C 端口设置(DM6645－DM6649) 下面的设置在 PC 运行时连续存取			

字	位	功　能	缺 省 值
DM6645	00～03	端口设置 0:标准(1个启动位,7个数据位,偶校验,2个停止位,9,600bps) 1:DM6646中设置	标准
	04～07	CTS控制设置 0:不允许CTS控制 1:允许CTS控制	不允许
	08～11	1:1链接的链接字 0:LR00～LR63;1:LR00～LR31; 2:LR00～LR15 1:N NT链接的最大PT节点号 1～7BCD码数(对C200HE－CPU□□－E PC为1～3)	LR00～LR63
	12～15	通信方式 0:上位机;1:RS－232C;2:1:1链接从站;3:1:1链接主站;4:NT链接(1:1);5:NT链接(1:N)	上位机链接
DM6646	00～07	波特率 00:1.2K;01:2.4K;02:4.8K 03:9.6K;04:19.2K	1.2K
	08～15	帧格式 起动　长度　停止　校验 00:　1位　7位　1位　偶 01:　1位　7位　1位　奇 02:　1位　7位　1位　无 03:　1位　7位　2位　偶 04:　1位　7位　2位　奇 05:　1位　7位　2位　无 06:　1位　8位　1位　偶 07:　1位　8位　1位　奇 08:　1位　8位　1位　无 09:　1位　8位　2位　偶 10:　1位　8位　2位　奇 11:　1位　8位　2位　无	1个起动位 7位数据 1个停止位 偶校验
DM6647	00～15	传输延迟 0000～9999:BCD码,单位10ms	0ms
DM6648	00～07	节点号(上位机链接) 00～31(BCD码)	0
	08～11	开始代码允许(RS－232C) 0:不允许　1:设置	不允许
	12～15	结束代码允许(RS－232C) 0:不允许(接收字节数) 1:设置(指定结束代码) 2:CR,LF	不允许

字	位	功　能	缺省值
DM6649	00～07	开始代码(RS－232C) 00～FF(二进制码)	不用(0000)
	08～15	DM6648 的位 12～15 设为 0:接收字节数 00:缺省设置(256 字节) 01～FF;1～255 字节 DD6648 的位 12～15 设为 1: 结束代码(RS－232C) 00～FF(二进制码)	
外设端口设置(DM6650～DM6654) 下面的设置在 PC 运行时连续存取			
DM6650	00～03	端口设置 0:标准(1 个起动位,7 位数据,偶校验,2 个停止位,9,600bps) 1:DM6651 中设置	标准
	04～11	保留	
	12～15	通信方式 0:上位机链接　1:RS－232C	上位机链接
DM6651	00～07	波特率 00:1.2K;01:2.4K;02:4.8K 0.3:9.6K;04:19.2K	1.2K
	08～15	帧格式 起动　长度　停止　校验 00:　1 位　7 位　1 位　偶 01:　1 位　7 位　1 位　奇 02:　1 位　7 位　1 位　无 03:　1 位　7 位　2 位　偶 04:　1 位　7 位　2 位　奇 05:　1 位　7 位　2 位　无 06:　1 位　8 位　1 位　偶 07:　1 位　8 位　1 位　奇 08:　1 位　8 位　1 位　无 09:　1 位　8 位　2 位　偶 10:　1 位　8 位　2 位　奇 11:　1 位　8 位　2 位　无	1 个起动位 7 位数据 1 个停止位 偶校验
DM6652	00～15	传输延迟(上位机链接) 0000～9999:单位 10ms	0ms
DM6653	00～07	节点号(上位机链接)00～31(BCD 码)	0
	08～11	开始代码允许(RS－232C) 0:不允许　1:设置	不允许
	12～15	结束代码允许(RS－232C) 0:不允许(接收字节数) 1:设置(指定结束码) 2:CR,LF	不允许

字	位	功　能	缺 省 值
DM6654	00～07	开始代码(RS～232C) 00～FF(二进制码)	0000
	08～15	DM6653 中位 12～15 设为 0： 接收字节数 00：缺省值(256 字节) 01～FF：1～255 字节 DM6653 中位 12～15 设为 1： 结束代码(RS－232C) 00～FF(二进制码)	
错误设置(DM6655) 下面的设置在 PC 运行时连续存取			
DM6655	00～03	中断程序错误允许 0：检测中断程序错误 1：不检测	检测
	04～07	保留	…
	08～11	循环时间监视允许 0：检测长周期为非严重错误 1：不检测长周期	检测
	12～15	电池低电压故障允许 0：检测电池低电压为非严重错误 1：不检测电池低电压	检测

3－6－5　通信板设置

DM6550～DM6554 包含通信板端口 B 的通信设置，DM6555～DM6559 包含通信板端口 A 的通信设置。

字	位	功　能	缺 省 值
通信板端口 B 设置(DM6550～DM6554) 下面的设置在 PC 运行时连续存取			
DM6550	00～03	端口设置 0：标准(1 个起动位，7 位数据，偶校验，2 个停止位，9，600bps)	标准
	04～07	CTS 控制允许 0：CTS 控制不允许 1：CTS 控制允许	不允许
	08～11	1：1 链接的链接字(一旦设为 1：1 链接主站，不可更改) 0：LR00～LR63；1：LR00～LR31；2：LR00～LR15 1：N PT 链接的最大 PT 节点数 1～7BCD 码数(对 C200HE－CPU□□－E PC 为 1～3)	LR00～LR63
	12～15	通信方式 0：上位机链接；1：RS－232C； 2：1：1 链接从站；3：1：1 链接主站； 4：NT 链接(1：1)；5：NT 链接(1：N)； 6：通信协议宏	上位机链接

字	位	功　能	缺省值
DM6551	00～07	波特率 00:1.2K;01:2.4K;02:4.8K;03:9.6K 04:19.2K	1.2K
	08～15	帧格式 起动　长度　停止　校检 00:　1位　7位　1位　偶 01:　1位　7位　1位　奇 02:　1位　7位　1位　无 03:　1位　7位　2位　偶 04:　1位　7位　2位　奇 05:　1位　7位　2位　无 06:　1位　8位　1位　偶 07:　1位　8位　1位　奇 08:　1位　8位　1位　无 09:　1位　8位　2位　偶 10　1位　8位　2位　奇 11:　1位　8位　2位　无	1个起动位 7位数据 1个停止位 偶校验
DM6552	00～15	传输延迟 0000～9999:BCD码,单位10ms	0ms
DM6553	00～07	节点号(上位机链接) 00～31(BCD码)	0
	08～11	开始代码允许(RS－232C) 0:不允许　1:设置	不允许
	12～15	结束代码允许(RS－232C) 0:不允许(接收字节数) 1:设置(指定结束代码) 2:CR,LF	不允许
DM6554	00～07	开始代码(RS－232C) 00～FF(二进制码)	
	08～15	DM6553中位12～15设为0: 接收字节数 00:缺省设置(256字节) 01～FF:1～255字节 DM6553中位12～15设为1: 结束代码(RS－232C) 00～FF(二进制码)	
通信板端口A设置(DM6555～DM6559) 下面的设置在PC运行时连续存取			

字	位	功　能	缺 省 值
DM6555	00～03	端口设置 0:标准(1 个起动位,7 位数据,偶校验,2 个停止位,9,600bps) 1:DM6556 中设置	标准
	04～07	CTS 控制设置 0:CTS 控制不允许 1:CTS 控制允许	不允许
	08～11	1:1 链接的链接字(一旦设为 1:1 链接主站,不能更改) 0:LR00～LR63;1:LR00～LR31;2:LR00～LR15 1:N NT 链接的最大 PT 节点数 1～7BCD 码数(对 C200H－CPU□□－E PC 为 1～3)	LR00～LR63
	12～15	通信方式 0:上位机链接;1:RS－232C; 2:1:1 链接从站;3:1:1 链接主站; 4:NT 链接(1:1);5:NT 链接(1:N); 6:通信协议宏	上位机链接
DM6556	00～07	波特率 00:1.2K;01:2.4K;02:4.8K 03:9.6K;04:19.2K	1.2K
	08～15	帧格式 起动　长度　停止　校检 00:　1 位　7 位　1 位　偶 01:　1 位　7 位　1 位　奇 02:　1 位　7 位　1 位　无 03:　1 位　7 位　2 位　偶 04:　1 位　7 位　2 位　奇 05:　1 位　7 位　2 位　无 06:　1 位　8 位　1 位　偶 07:　1 位　8 位　1 位　奇 08:　1 位　8 位　1 位　无 09:　1 位　8 位　2 位　偶 10:　1 位　8 位　2 位　奇 11:　1 位　8 位　2 位　无	1 个起动位 7 位数据 2 个停止位 偶校验
DM6557	00～15	传输延迟 0000～9999:BCD 码,单位 10ms。	0ms
DM6558	00～07	节点号(上位机链接) 00～31(BCD 码)	0
	08～11	开始代码允许(RS－232C) 0:不允许;1:设置	不允许
	12～15	结束代码允许(RS－232C) 0:不允许(接收字节数) 1:设置(指定结束代码) 2:CR,LF	不允许

字	位	功　能	缺省值
DM6559	00～07	开始代码(RS－232C) 00～FF(二进制码)	0000
	08～15	DM6558 中位 12～15 设为 0： 接收字节数 00：缺省设置(256 字节) 01～FF：1～255 字节 DM6558 中位 12～15 设为 1： 结束代码(RS－232C) 00～FF(二进制码)	

3－6－6　特殊 I/O 单元区域设置

DM6602 中位 08～15 的设置决定特殊 I/O 单元区域的大小和位置，如下表所示。

设置	方　式	功　能
00	C200H 兼容 RAM 方式	DM1000～DM2599 用作特殊 I/O 单元区域。 ·特殊 I/O 单元区域中数据可以改写。 ·数据不可转换到 ROM。
01	C200H 兼容 ROM 方式 1	启动时 DM7000～DM7999 中内容传送到 DM1000～DM1999，DM1000～DM1999 用作特殊 I/O 单元区域。 ·必须预先执行 UM 区域分配操作。 ·在 C200H 中，这个方式适合于使用 EEPROM 和 EPROM。 ·数据通过 DM7000～DM7999 转换，可以间接存入 ROM。
02	DM 线性 方式 1	DM7000～DM7999 用作特殊 I/O 单元区域。 ·DM1000～DM1999 可以用作常规 DM。 ·DM7000～DM7999 可以存入 ROM。
11	C200H 兼容 ROM 方式 2	启动时，DM7000～DM8599 中内容传送到 DM1000～DM2599，DM1000～DM2599 用作特殊 I/O 单元区域。 ·必须预先执行 UM 区域分配操作。 ·数据通过 DM7000～DM8599 转换，可以间接存入 ROM。
12	DM 线性 方式 2	DM7000～DM8599 用作特殊 I/O 单元区域。 ·DM1000～DM2599 可以用作常规 DM。 ·DM7000～DM8599 可以存入 ROM。

DM7000～DM9999 是不能直接用编程器读取和改写的。要从程序中读取这些数据，需使用扩展 DM 读指令－XDMR(一)将数据复制到另一数据区或常规数据区中。

当设置 C200H 兼容 ROM 方式或 DM 线性方式时，必须预先执行 UM 区域分配操作，将部分所用梯形图程序区变为扩展 DM。如果存储器不分配一个扩展 DM 区，会发生系统错误(FAL9B)。有关 UM 区域分配操作细节参见 7－2－15“UM 区域分配”。

当设置 DM 线性方式时，特殊 I/O 单元数据区从 DM7000 而不是 DM1000 开始，故在特殊单元操作手册里出现的 DM 地址上需加上 6000。

当特殊 I/O 单元区域设置为 01，02，11 或 12，并且分配的扩展 DM 不能超出 DM8000，那末单元号为 A～F 的特殊 I/O 单元在存取它的分配区域时，会产生错误。

3－7 HR(保持继电器)区域

HR区域用于存储/操作各种数据并可按字或按位存取，字地址从HR00～HR99，位地址HR0000～HR9915。HR位可按任何次序使用并和普通位一样用于程序。

当系统操作方式改变、电源中断或PC操作停止，HR区域保持状态。

HR区域的位和字可用于PC操作中止时保留数据，HR位还有各种特殊应用。如用KEEP指令产生一个锁存继电器及形成自保持输出，第四章编写及输入程序和第五章指令集，讨论了这些功能。

注 使用SYSMAC NET系统时，路径表和监视定时器需要的字分配在HR00～HR42之中。

3－8 TC(定时器/计数器)区域

TC区域用来生成和编制定时器和计数器并保存定时器/计数器的结束标志，设定值(SV)和当前值(PV)，通过TC号(TC000～TC511)可存取这些数。使用下列指令：TIM，TIMH，CNT，CNTR(12)和TTIM(87)，每一个TC号可定义一个定时器或一个计数器。在定时器或计数器指令中使用TC号时不需前缀。

一旦用其中的一条指令定义好TC号，该号码就不能再在程序的其它地方用相同的指令或其它指令再定义，否则在程序检测时会出错。TC号可按任何次序定义。

TC号一旦定义好，除了上述指令外，它可作为操作数用于其它指令。当定义定时器时，该操作数是TC号再加上前缀TIM。使用加缀TIM与用以定义该定时器的指令无关。当定义成计数器时，该操作数是TC号再加上前缀CNT。使用加前缀CNT也与用以定义该计数器的指令无关。

TC号可指定成位数据或字数据形式的操作数。当指定成位数据形式的操作数时，TC号存取定时器或计数器的结束标志。当定义成字数据形式操作数时，TC号存取在定时器或计数器PV的存储器。

TC号也可通过编程设备存取定时器和计数器的SV。有关使用编程器存取的过程详见第7－1节“监视操作和修改数据”。

电源中断时，TC区保存定时器和计数器的SV。PC开始工作或在联锁程序部分复位定时器的PV，但计数器的PV不复位。有关联锁程序部分的定时器及计数器操作参见第5－10节“联锁和联锁解除指令－IL(02)和ILC(03)”。

注 编程时TIM000用来指下列三件事：定时器指令定义TC号为000，该定时器的结束标志以及该定时器的PV，用上下文能区分这些意义。例如，第一个总是指令，第2个总是一个位以及第3个总是一个字。上述适用于其它带前缀TIM或CNT的TC号。

3－9 LR(链接继电器)区域

LR区域用作普通数据区，在PC间传送信息，通过PC链接系统可实现这种传送。

对每个PC，某些字分配为写区域。这些字由PC写入且自动传送至系统中其它PC相同的LR字，而其它PC的写区域传送至该PC作为读区域中，故在PC链接系统中每个PC都可读取由其它PC写入的数据，只有分配给特定PC的写区域才能写，其它字只能读。参见PC链接系统手册。

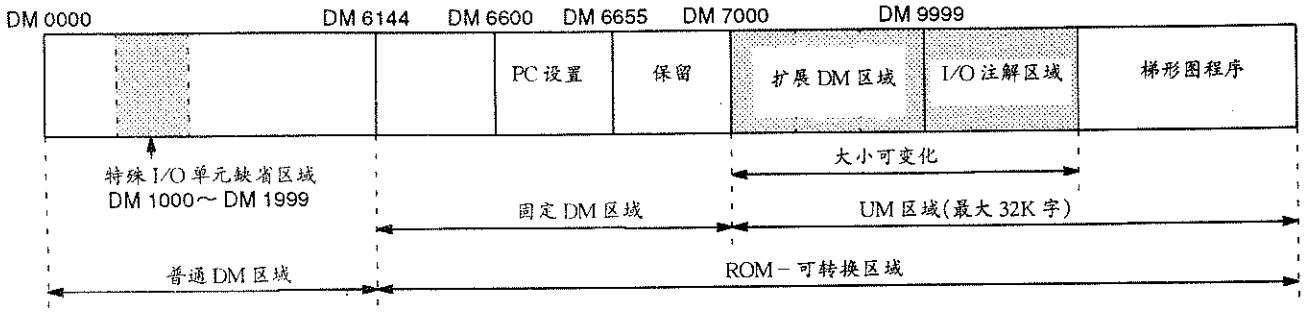
LR区域可按字或按位存取。LR区域字地址是从LR00～LR63，LR区域位地址是从LR0000～LR6315。任何一部分LR区域只要PC链接系统不用，则都可用作工作字，或用于SYSMAC NET链接或SYSMAC LINK系统。

当PC变换至编程方式，或电源中断或在联锁程序部分中复位时，LR区域数据不能保留下来，详见第5－10节“联锁及和联解除指令－IL(02)及ILC(03)”。

3－10 UM区域

在C200HX/HG/HE PC中,UM区域包含梯形图程序。部分UM区域可分配用作扩展DM或I/O注释区。UM使用容量范围从C200HE－CPU11－E中3.2K字到C200HX－CPU□4－E中31.2K字。

分配扩展DM区可以使用编程器或SYSMAC支持软件(SSS(中文版)),但是I/O注释区的分配只能使用SSS(中文版)。DM和UM区域结构示于下图。



注

用SSS(中文版)分配UM给扩展DM或I/O注释区,详见SYSMAC支持软件(SSS(中文版))手册。用编程器分配UM给扩展DM,详见7－2－15"UM区域分配"。

区 域	功 能
普通DM	这区域可以自由用于运算和编程指令。DM只能按字存取。当使用特殊I/O单元时,DM1000～DM2599分配给特殊I/O单元,而当PC设置(DM6602)中将特殊I/O单元区域设置在DM7000～DM8599,它们才可用作普通DM。
PC设置	PC设置包含控制PC运行的各种设置。
保留	这区域为系统使用保留。它不能由用户存取。
扩展DM	这区域包含初始化数据,如:特殊I/O单元数据,PT的数字表或字符串表,和运算数据。 扩展DM中数据不能像正常DM中的那样直接读取。 扩展DM可以通过编程器上十六进制/BCD码数据修改操作或用SYSMAC支持软件传送可编辑DM数据来改写。
I/O注释	这区域用于储存I/O注释。它可以和程序一起存储。I/O注释可以不通过它的检索操作进行检查。
梯形程序	这区域用于储存用户编制的梯形程序。分配给扩展DM和/或注释的区域取自于该区域。

注 1. 当UM区域字分配给扩展DM和/或I/O注释区时,梯形图程序区相应减少。在存储器分配给扩展DM和/或注释区时,确信在梯形图程序区中有足够充足的容量。

2. UM区域的缺省设置没有分配任何存储器给扩展DM或I/O注释区。如需要,必须由用户分配存储器。

3－11 TR(临时继电器)区域

TR区域只给LD和OUT指令提供了8个位,用于某些分支类型梯形图程序。TR位的用法见第四章"编写和输入程序"。

TR地址是从TR0～TR7。只要同一TR位在同一指令块中没有重复使用,每位可以任意次序并任意次

数使用。

3－12 EM(扩展数据存储器)区域

C200HG 和 C200HX PC,除了大容量 DM 区域外,还有可存储多达 18K 字数据的 EM 区域。EM 区域分成三组,每组有 6,144 字(EM0000～EM6143)。C200HG PC 有一组(0 组),C200HX PC 有三组(0,1,和 2 组)。有效的组称为当前组。

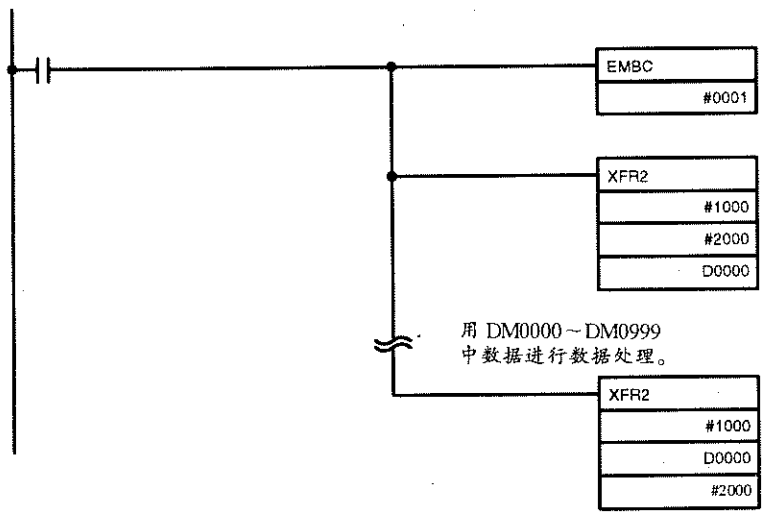
3－12－1 使用 EM 区域

大多数指令不能直接存取 EM 区域,但 PC 提供 EMBC(一),XFR2(一),BXF2(一),和 IEMS(一)指令来管理 EM 区域数据。

指 令	功 能
EMBC(一)	当前组转换到指定的组号。
XFR2(一)	在当前 E 组内或在当前 EM 组和一个常规数据区之间传送数据。
BXF2(一)	在指定的 EM 组和另一个 EM 组成一个常规数据之间传送数据。
IEMS(一)	转换 DM 间接寻址的目标到指定的 EM 组。也可以将目标转回到 DM。

例 1

下例使用 EMBC(一)指令将组 1 设为当前组,使用 XFR2(一)指令传送 EM2000～EM2999 中内容到 DM0000～DM0999。执行一个程序段之后,将 DM0000～DM0999 的内容传回 EM2000～EM2999。



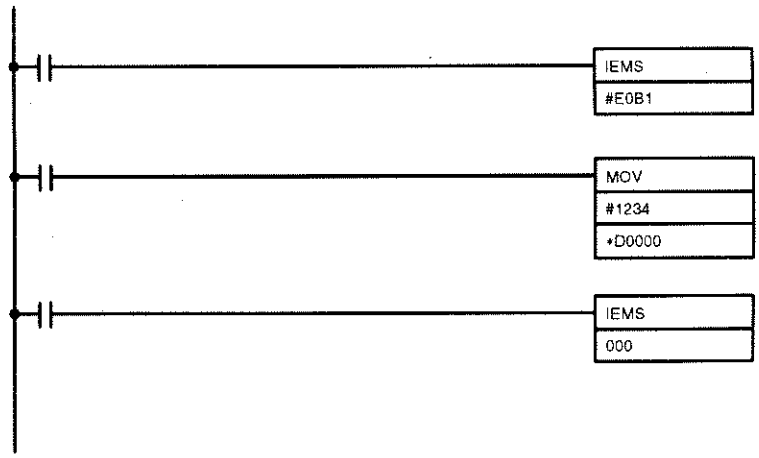
注

如果用 BXF2(一)指令执行数据传送,可以指定任意组号,就不需要 EMBC(一)指令来选择 EM 组 1。

例 2

下列使用 IEMS(一)指令将间接寻址的目标(＊DM)转换到 EM 组 1。这条指令执行之后,＊DM 操作数存取 EM 组 1 中数据而不再是 DM 区域数据。此时,在 MOV(21)指令的第二操作数传送＃1234 到 EM 组中一个字(在例中,＃1234 被传送到 EM0100,如果 DM0000 内容是 0100 的话。)

后面,程序用操作数 0000 执行 IEMS(一)指令,将间接寻址的目标(＊DM)又转回到 DM 区域。



注 1. 必要时,一定要将间接寻址的目标返回到缺省值(DM 区域)。
在下一个周期开始处,目标自动回到 DM 区域。
2. 间接寻址的目标在中断子程序开始时回复到 DM 区域,但在子程序中可以改变。当控制返回主程序时,目标又返回到原有设定。

3－12－2 当前 EM 组

PC 通电时,当前 EM 组设置为组 0,当前 EM 组可用 EMBC(一)或 IEMS(一)指令更改。和间接寻址的目标不一样,当前组号在周期开始或中断子程序开始时不初始化。

在 PC 通电之后,切换 PC 方式或完成了中断子程序的执行,将恢复原已转换过的 EM 组的状态。

第四章　编写和输入程序

本章解释了有关编写基本的梯形图程序，将其送入存储器并执行的基本步骤和概念，介绍了用于建立梯形图基本结构并控制其执行的指令。完整的编程指令将在第五章“指令集”中介绍。

4－1　基本步骤 …… 72
4－2　指令术语 …… 72
4－3　程序容量 …… 72
4－4　基本梯形图 …… 73
4－4－1　基本术语 …… 73
4－4－2　助记符 …… 74
4－4－3　梯形指令 …… 75
4－4－4　输出和反相输出(OUTPUT/OUTPUT NOT) …… 77
4－4－5　结束(END)指令 …… 77
4－4－6　逻辑块指令 …… 77
4－4－7　对多个右侧指令编写 …… 84
4－5　编程器 …… 84
4－5－1　键盘 …… 85
4－5－2　PC 方式 …… 86
4－5－3　信息显示选择开关 …… 86
4－6　操作准备 …… 86
4－6－1　输入口令字 …… 87
4－6－2　蜂鸣器 …… 87
4－6－3　清除内存 …… 87
4－6－4　登记 I/O 表 …… 90
4－6－5　清除错误信息 …… 91
4－6－6　检验 I/O 表 …… 91
4－6－7　读 I/O 表 …… 93
4－6－8　清除 I/O 表 …… 95
4－6－9　SYSMAC NET 链接表传送 …… 96
4－7　程序的输入、修改和检查 …… 97
4－7－1　从程序存储器地址中读写 …… 97
4－7－2　输入和编辑程序 …… 98
4－7－3　检查程序 …… 100
4－7－4　显示扫描周期时间 …… 103
4－7－5　程序搜索 …… 103
4－7－6　插入、删除指令 …… 105
4－7－7　指令行的分支 …… 107
4－7－8　跳转 …… 110
4－8　位状态的控制 …… 112
4－8－1　上升沿微分/下降沿微分指令(DIFU/DIFD) …… 112
4－8－2　保持(KEEP)指令 …… 112

4－8－3　自保持位 …… 113
4－9　工作位(内部继电器) …… 113
4－10　编程注意事项 …… 115
4－11　程序执行 …… 116
4－12　特殊 I/O 单元接口程序 …… 116
4－12－1　特殊 I/O 单元重启动 …… 116
4－12－2　特殊 I/O 单元出错处理程序 …… 117
4－12－3　改变特殊 I/O 单元设置 …… 117
4－12－4　特殊 I/O 单元的 I/O 刷新间隔时间 …… 118
4－12－5　减少扫描周期时间 …… 119
4－13　模拟定时器单元编程 …… 120
4－13－1　运行 …… 120
4－13－2　位分配和 DIP 开关设置 …… 120
4－13－3　示例程序 …… 121

4－1 基本步骤

编写程序有一些基本步骤，附录 D“字分配记录表”和附录 E“程序代码表”可以为编程提供帮助。

1. 首先要获得一张列出所有 I/O 设备和所分配 I/O 点的目录，并准备一张表，它能表明分配给每台I/O设备的 I/O 位。

2. 如果 PC 有这样的单元：该单元在数据区域而不是在 IR 区域被分配了字或被分配了 IR 字，而字中每位的功能被单元所确定，准备类似的表来表示“何字用于何单元以及字中每一位是何种功能”。这些单元包括特殊 I/O 单元和链接单元。

3. 确定提供给工作位的字，并准备一张表示“字的分配情况”的表。

4. 再准备 TC 号和跳转号的表，以便使用时对它们进行分配。注意：在程序中，一个 TC 号只能被定义一次，从 01～99 的每一个跳转号也只能被用一次。（TC 号在 5－14“定时器和计数器指令”介绍，跳转号在本节的后面介绍。）

5. 画梯形图

6. 将程序输入 CPU。当使用编程器时，需将程序转化为助记符形式。

7. 检查程序有无语法错误并改正。

8. 运行该程序以检查是否存在执行错误，并改正。

9. 当整个控制系统安装好并可用时，运行该程序，若有必要的话，进行细调。

10. 做程序备份。

有关梯形图编程及向助记符方式转换的基本内容将在 4－4“基本梯形图”中介绍。通过编程器准备和输入程序在 4－5“编程器”至 4－7“程序的输入、检查和修改”中介绍。第四章其余内容将涉及“高级编程、编程注意事项和程序执行。”所有特殊应用指令在第五章“指令集”中介绍。程序调试在第七章“程序监视和执行”中介绍，第九章“故障检查”，也将介绍有关程序调试的内容。

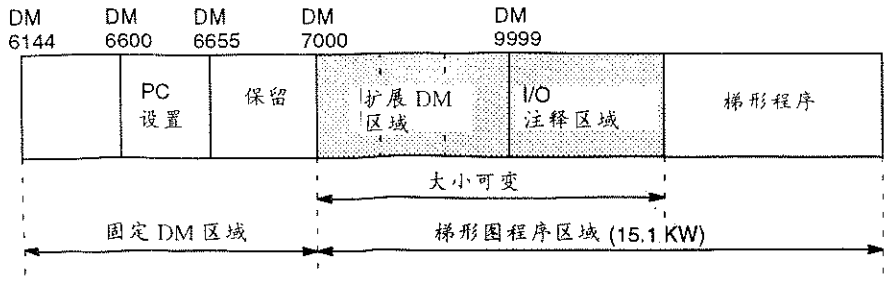
4－2 指令术语

有两类基本指令用于梯形图编程：一类指令用来代表梯形图上的条件，它们仅在将程序转化为助记符时以指令形式使用。另一类指令用在梯形图的右边并且按照指令行上的条件执行。多数指令有至少一个或更多操作数与之联系。操作数为一条将要执行的指令指明或提供数据。这些操作数有时是以实际数字形式输入。但通常是包含了需用数据的数据区域里的地址字或地址位。例如，一条以 IR000 为其源操作数的 MOVE(传送)指令将 IR000 的内容传送至其它位置，这个位置也将被指定为操作数。如果位的地址被指令为操作数，则该位称作操作数位，如果字的地址被指定为操作数，则该字称作操作数字。如果实际值以常量输入，其前面要标“＃”以表示它不是地址。

其它描述指令的术语在第五章“指令集”中介绍。

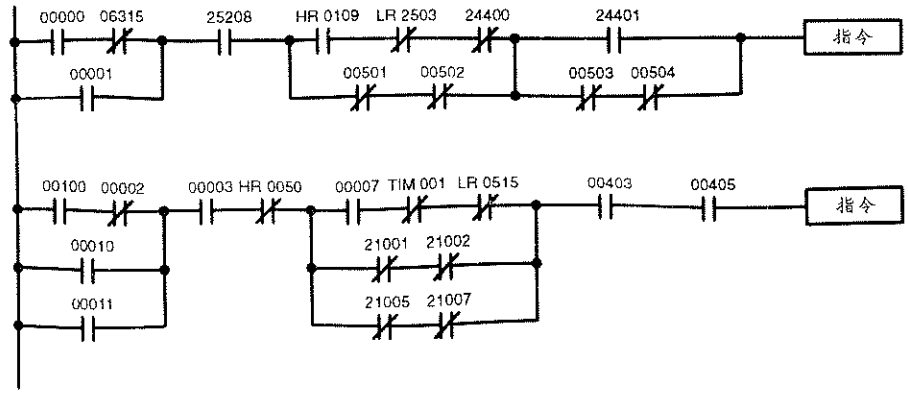
4－3 程序容量

用户程序的最大容量取决于分配到扩展 DM 和 I/O 注解区域的 UM 总量。如下图所示，当分配给扩展 DM、I/O 注解区域的内存分别为 3KW 和 2KW 时，大约有 10.1KW 内存可提供给梯形图。有关 UM 配置的深层信息可参阅 3－10“UM 区域”。



4－4 基本梯形图

一个梯形图由左边一条垂直向下的线及若干条与之相交并向右延伸的分支线组成。左边的线称为母线，分支线称为指令行或回路。沿着指令行设置各种条件，它们连向右边的其它指令。这些条件的逻辑组合决定右边指令何时及怎样执行。下面是一个梯形图。



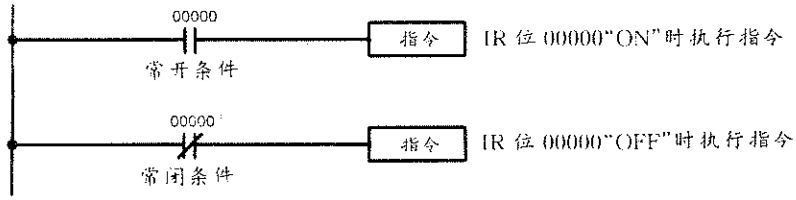
如上图所示，指令行可以分支，可以会合。指令行上垂直的一对线称为条件。无斜线穿过的条件(⊣⊢)称为常开条件，对应着 LOAD(取)、AND(与)、OR(或)指令。有斜线穿过的条件(⊣/⊢)称常闭条件，对应 LOAD NOT(取非)、AND NOT(与非)和 OR NOT(或非)指令。每个条件上方的数字表示指令的操作数位。它是与条件相关的位的状态，而每个条件决定后面指令的执行条件。对应于条件的每个指令的操作方式在下面介绍。在考虑这些问题之前，介绍几条基本术语。

注意：当以 SSS(中文版)显示梯形图时，在梯形图的右边显示第二条母线，与所有右侧指令相连。这不会改变梯形图程序的任何功能。在右边的指令和右边母线之间不能放置条件，即右边的指令必须直接与右边的母线相接。详细情况请参阅“SSS(中文版)操作手册：C 系列”。

4－4－1 基本术语

常开/常闭条件：

梯形图中每个条件为“ON”或“OFF”，取决于分配给它的操作数位的状态。如果操作数位是“ON”，则常开条件为“ON”。操作数位是 OFF，则常开条件为“OFF”。操作数位状态为“ON”，则常闭条件为“OFF”。操作数位状态为“OFF”，则常闭条件为“ON”。一般说，常开条件用在某位为“ON”时执行某操作，而常闭条件用在某位为“OFF”时执行某操作。



执行条件

在梯形图编程中，一条指令前面的"ON"及"OFF"的逻辑组合决定了组合条件，指令在此条件下执行。这个为 ON 或 OFF 的组合条件称作指令的执行条件。除 LOAD 指令以外的所有指令都有执行条件。

操作数位

指定给任何梯形指令的操作数位可以是以下区域的任何位。这些区域是 IR 区域、SR 区域、HR 区域、AR 区域、LR 区域或 TC 区域。这意味着梯形图中的条件可以被 I/O 位、标志、工作位、定时器/计数器等确定。LOAD，OUTPUT 指令也能使用 TR 区域的位，但只能用在特殊用途。详细情况可参阅 4－7－7"指令行的分支"。

逻辑块

与指令行相连的条件间的关系决定了条件对应于什么指令。任何一组能共同产生一个逻辑结果的条件组合称为逻辑块。尽管可以在没有经实际分析的单独逻辑块情况下写出梯形图，但明白逻辑块对于更有效编程是必要的。而且在程序要以助记符形式输入时，逻辑块是至关重要的。

4－4－2 助记符

梯形图不能直接通过编程器输入 PC，此时需要 SSS(中文版)。要从编程器输入梯形图，必须把它转化为助记符形式。助记符可以提供与梯形图完全相同的内容，但该形式可以直接被送入 PC。实际上可以直接用助记符编程，这对于初学者或编写复杂程序来说有一定难度。不管什么编程设备，程序是以助记符形式存贮在内存中的，因而掌握助记符十分重要。

把编程器作为外围设备以及助记符在透彻理解程序方面都有着重要意义，因此把助记符和梯形图放在一起介绍。如果通过 SSS(中文版)输入，就不必使用助记符，(用 SSS(中文版)时也可使用助记符)。

程序内存结构

程序是被送入程序内存中的地址，程序内存中的地址和其它内存区域稍有不同。因为这里的每一个地址，没必要保持同量的数据，每一个地址保存一个指令和所需要的定义及操作数。(后面详细介绍)。因为有些指令不要求操作数，而有些则最多要求三个操作数，故程序内存地址可以是 1 个至 4 个字长。

程序内存地址从 00000 开始直到程序内存容量用完为止。每个地址的第一个字定义了指令，而任何用于指令的定义也包含在第一个字中。如果一条指令只要求单个位操作数(无定义)，那么位操作数和该指令在同一指令行上编程。被一条指令要求的其它字包含着操作数，该操作数规定了什么数据将被使用。当向助记符形式转换时，除了梯形图指令以外的部分都要写成同一种形式，一个字对应一行，如同它们在梯形图符号中出现一样。下面是一个助记符的例子。其中所用的指令将在手册后面部分介绍。

地址	指令	操作数
00000	LD	HR 0001
00001	AND	00001
00002	OR	00002
00003	LD NOT	00100

（续表）

地址	指令	操作数
00004	AND	00101
00005	AND LD	00102
00006	MOV(21)	
		000
		DM 0000
00007	CMP(20)	
		DM 0000
		HR 00
00008	LD	25505
00009	OUT	00501
00010	MOV(21)	
		DM 0000
		DM 0500
00011	DIFU(13)	00502
00012	AND	00005
00013	OUT	00503

助记符表中地址和指令列只填入指令中，而其它行的左侧两例为空。如果指令不需定义或位操作数，第一条指令行的操作数列为空。略过空的数据列区域（对所有不要求数据的指令字），可以迅速扫描数据列以检查是否有地址被遗漏。

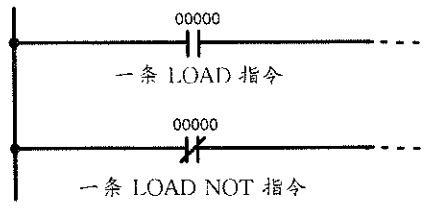
编程时，除非因某种原因，指令要求一个不同的位置外，地址一般会自动显示而无需输入，当向助记符转化时，最好从程序内存地址 00000 开始，除非有某种特殊原因要求从其它地方开始。

4－4－3 梯形指令

在梯形图上对应于条件的指令是梯形指令。梯形指令既可与逻辑块指令（下面介绍）独立，又可与之结合构成执行条件。所有其它指令的执行就建立在该执行条件上。

LOAD/LOAD NOT（取/取非）

启动梯形图中任何逻辑块的第一条指令便是 LD 或 LD NOT 指令。每一条这种指令需要一条助记符行。下面例子中的“指令”是一个代表，它可以是本手册后面要介绍的右侧指令中的任一条。

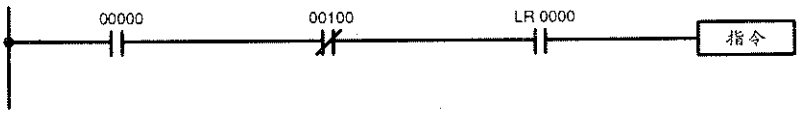


地址	指令	操作数
00000	LD	00000
00001	指令	
00002	LD NOT	00000
00003	指令	

当这是指令行上的唯一条件且为“ON”时，则右侧指令的执行条件为“ON”。对于 LD 指令（常开条件），当 IR00000 为 ON 时，则执行条件为“ON”；对于 LD NOT 指令（常闭条件），当 IR00000 为“OFF”时，执行条件为“ON”。

AND/AND NOT(与/与非)

当同一指令行上存在串联的两个或更多条件时,那么第一个条件对应于一条 LD 或 LD NOT 指令;余下条件对应 AND 或 AND NOT 指令。下面例子有三个条件,按顺序(从左至右)分别对应一条 LD 指令,一条 AND NOT 指令和一条 AND 指令,而每条指令都需要一行助记符表示。



地址	指令	操作数
00000	LD	00000
00001	AND NOT	00100
00002	AND	LR 0000
00003	指令	

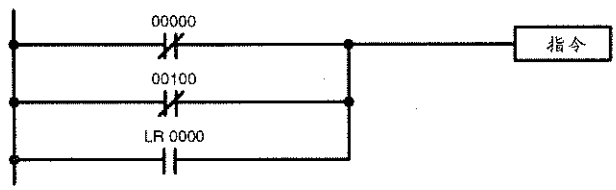
只有当三个条件都为"ON"时,上例指令的执行条件才为"ON",即 IR00000 为"ON",IR00100 为"OFF"、LR0000 为"ON"。

串联中的 AND 指令可以单独考虑,每一条 AND 指令对其执行条件(至该点的所有条件之和)和 AND 指令的操作数位的状态进行逻辑"与"。如果两者都为"ON"则为下一条指令产生一个"ON"执行条件。如果两者中任一者为 OFF,则结果都是"OFF"。串联的第一条 AND 指令的执行条件是指令行上第一个条件。

当串联时,每一条 AND NOT(与非)指令将对其执行条件以及它的操作数位的"非"进行逻辑"与"。

OR/OR NOT(或/或非)

当两个或更多条件是放置在相互独立的指令行上,且这些指令行并联相接,那么第一个条件对应一条 LD 或 LD NOT 指令;其它条件对应 OR 或 OR NOT 指令。下面例子显示了三个条件,分别对应(按顺序自顶而下)LD NOT、OR NOT 及 OR 指令。而且,其中每条指令需要一行助记符表示。

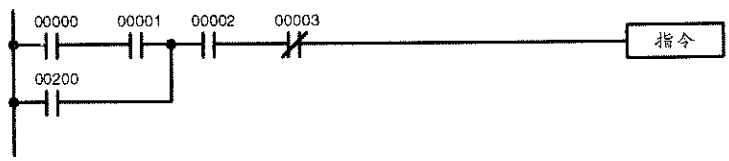


地址	指令	操作数
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	指令	

只要三个条件中任一个为 ON(即 IR00000 为 OFF 或 IR00100 为 OFF 或 LR0000 为 ON),则指令的执行条件为 ON,OR 和 OR NOT 指令的作用可以单独考虑,每一条 OR/OR NOT 指令对其执行条件及其操作数位的状态进行逻辑"或"。若两者中任一者为 ON,则下一条指令的执行条件为"ON"。

AND 和 OR 指令组合使用

在更加复杂的梯形图中对"AND"和"OR"指令进行结合时,有时它们可以被单独考虑,每一条指令都会对执行条件和操作数位的状态进行逻辑操作。下面是一个例子。仔细研究该例,直到了解助记符和梯形图遵循同一逻辑。



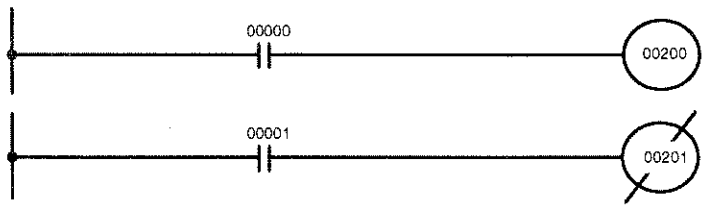
地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	OR	00200
00003	AND	00002
00004	AND NOT	00003
00005	指令	

这里先对 IR00000 与 IR00001 两者状态进行 AND，以作为与 IR00200 状态进行 OR 的执行条件。OR 操作的结果作为与 IR00002 的状态进行 AND 的执行条件，而此结果又作为与 IR00003 状态的非进行 AND 的执行条件。

在更复杂的梯形图中，在为最后一条指令确定一个执行条件之前必须考虑逻辑块，此时要使用 AND LD 和 OR LD 指令。考虑更复杂的梯形图之前，要寻找所需指令以完成一个简单的“输入——输出”程序。

4－4－4 输出和输出非(OUTPUT/OUTPUT NOT)

输出执行条件的组合结果的最简单方法是使用 OUTPUT 和 OUTPUT NOT 指令将它直接输出。这些指令用于按照执行条件控制指令的操作数位的状态。在 OUTPUT 指令作用下，只要执行条件为“ON”或“OFF”，则操作数位将相应为“ON”或“OFF”。在 OUTPUT NOT 指令作用下，只要执行条件为 OFF 或 ON，则操作数位即相应变为“ON”或“OFF”。这些情况如下所示。在助记符形式中，上述指令中的每一条需要一行助记符。



地址	指令	操作数
00000	LD	00000
00001	OUT	00200

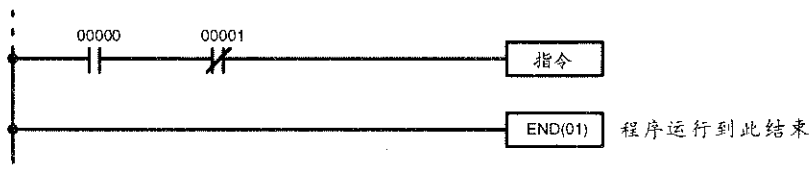
地址	指令	操作数
00000	LD	00001
00001	OUT NOT	00201

上例中，只要 IR00000 为“ON”，则 IR00200 将变为“ON”；只要 IR00001 为“ON”，则 IR00201 变为“OFF”。此处，IR00000 和 IR00001 是输入位，而 IR00200 和 IR00201 是输出位，这些输出位分配给受 PC 控制的单元，即通过分配在 IR00000 和 IR00001 的输入点进来的信号分别控制分配在 IR00200 和 IR00201 的输出点。

OUTPUT 或 OUTPUT NOT 指令与 TIMER(定时器)指令的结合可以控制输出“ON”或“OFF”一段时间。详细情况可参阅 5－14－1“定时器指令——TIM”中的例子。

4－4－5 结束(END)指令

用为完成一个简单程序的最后一条指令是 END(结束)指令。当 CPU 扫描程序时，它执行所有的指令，直到第一条 END 指令为止，然后回到程序起始再次执行。END 指令可以放于程序中的任何位置，特别在调试程序时会这样做，在第一条 END 指令后的指令不会被执行(除非将其移掉)。在助记符形式中，END 指令后面的数字是其功能代码，当输入大多数指令到 PC 中时会用到它，这些在后面介绍。END 指令不要求操作数，而且 END 指令的指令行上不能设置条件。



地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	指令	
00003	END(01)	

如果在程序中没有 END 指令，那么程序将不会被执行。

到此已介绍了所有用于编写简单输入—输出程序的指令。在完成梯形图基本内容并转入向 PC 输入程序之前，介绍逻辑块指令(AND LD 和 OR LD)，这些指令即使对简单梯形图有时也是必需的。

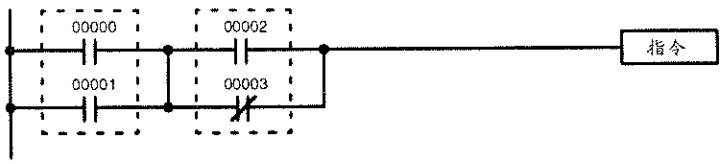
4－4－6 逻辑块指令

逻辑块指令不对应梯形图上特定的条件；它们只描述逻辑块之间的关系。AND LD 指令对两个逻辑块

产生的执行条件逻辑“与”。OR LD 指令对两个逻辑块产生的执行条件逻辑“或”。

AND LD

尽管下面梯形图看上去较简单，但它需要一条 AND LD 指令。



地址	指令	操作数
00000	LD	00000
00001	OR	00001
00002	LD	00002
00003	OR NOT	00003
00004	AND LD	
00005	指令	

上图中两个逻辑块用虚线表示。研究这个例子可以看出产生一个“ON”执行条件的前提是：左边逻辑块中的任一个条件为“ON”（即 IR00000 为 ON 或 IR00001 为“ON”），同时右边逻辑中任一条件为“ON”（即 IR00002 为 ON 或 IR00003 为 OFF）。

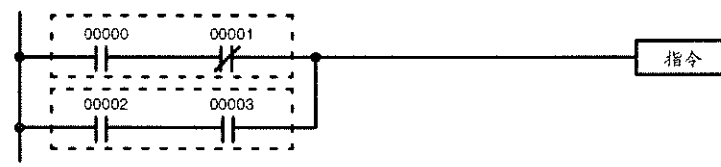
仅仅使用 AND 和 OR 指令不能把上面梯形图转化为助记符形式，如果对 IR00002 与 IR0000 和 IR00001 间“OR”操作的结果进行 AND 操作，这样就无法进行 IR00002 与 IR00003 之间的 OR NOT 操作，而变成最终在 IR00003 与 IR00002 和第一个 OR 操作间 AND 操作的结果之间进行 OR NOT 操作。而程序要做的是独立地进行 OR(NOT)操作，然后将所有结果结合起来。

要做到这一点，可以在指令行的中间使用“LD”或“LD NOT”指令。当按此方式使用“LD”或“LD NOT”指令时，当前执行条件保存在一个特殊缓冲区，而且逻辑程序重新开始，要将当前执行条件的状态和一个先前“未使用”的执行条件的状态进行结合，使用一条“AND LD”或“OR LD”指令。此处“LD”用于使最后一个未使用的执行条件被取出。除了指令行上第一个条件外，都能使用“LD”或“LD NOT”指令，产生一个未使用的执行条件。

根据助记符指令，分析上面梯形图，IR00000 的条件是一条“LD”指令，其下面条件是 IR00000 与 IR00001 状态之间的一条“OR”指令。IR00002 的条件是另一条 LD 指令，其下面条件是一条“OR NOT”指令，即在 IR00002 的状态和 IR00003 状态的“非”之间的一条“OR”指令。要得到右端指令的执行条件，必须对由上面两个逻辑块得到的执行条件进行逻辑与“AND LD”指令可做到这一点。该梯形图对应的助记符表如下所示。“AND LD”指令本身不需要操作数，因为它依先前确定的执行条件而操作。表中划线表示无操作数被指定或输入。

OR LD

下面梯形图中，在上部与下部逻辑块间需要一条“OR LD”指令。两种情况都会为右端指令产生一个“ON”执行条件；这两种情况是 IR00000 为“ON”同时 IR00001 为“OFF”，或者 IR00002、IR00003 皆为“ON”。除了当前执行条件和最后“未使用”执行条件之间进行的是“OR”操作外，“OR LD”指令的操作及其助记符和“AND LD”指令的操作完全一致。



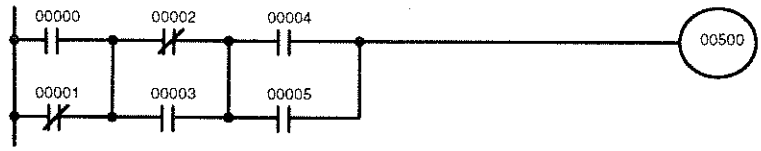
地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR LD	
00005	指令	

当然，一些梯形图将同时需要“AND LD”、“OR LD”指令。

逻辑块指令串联

对串联逻辑块指令梯形图编写时，梯形图必须分成若干逻辑块。每块使用一条“LD”指令编写第一个条件，然后使用“AND LD”或者“OR LD”指令对若干逻辑块进行逻辑组合。用“AND LD”和“OR LD”指令时，可以有两条途径实现上述目的。一条途径是在前面两个逻辑运算后编写逻辑块指令，然后在每增加一个逻辑块后面编写逻辑块指令。另一条途径是对所有要结合的逻辑块编写，先用“LD”或“LD NOT”指令启动每个逻辑块，然后编写用于结合逻辑块的逻辑块指令。在此过程中，用于最后一对逻辑块将首先被结合，然后前面的每一逻辑块被结合，逐次递推回第一个逻辑块。尽管上面两种方法都产生完全相同的结果，但第二种对所有逻辑块指令一起编写的方法，仅仅用在有 8 个(或更少)逻辑块要被结合的情况，也即要用 7 条(或更少)逻辑块指令的情况。

下面梯形图转化为助记符时，要求“AND LD”，因为有三对并联条件串联连接。对程序编写的两种方法都显示如下。



地址	指令	操作数
00000	LD	00000
00001	OR NOT	00001
00002	LD NOT	00002
00003	OR	00003
00004	AND LD	——
00005	LD	00004
00006	OR	00005
00007	AND LD	——
00008	OUT	00500

地址	指令	操作数
00000	LD	00000
00001	OR NOT	00001
00002	LD NOT	00002
00003	OR	00003
00004	LD	00004
00005	OR	00005
00006	AND LD	——
00007	AND LD	——
00008	OUT	00500

用上面右边办法进行编写时，可以被结合的程序块最多为 8 个。而采用第一种办法时，可被结合的程序块数没有限制。

下面梯形图在转化为助记符时，要求“OR LD”指令，因为有三对串联条件并联连接。



每对条件中的第一个条件转换为“LD”用分配的位作其操作数，然后和该对中另一个条件进行“与”操作。先编写前面二个逻辑块，紧跟一条 OR LD 指令，然后编写最后逻辑块，再用一条 OR LD 指令连接。或者首先对前面三个逻辑块编写，接下来是两条“OR LD”指令。两种方法的助记符表示如下所示。

地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	OR LD	——
00005	LD	00004
00006	AND	00005
00007	OR LD	——
00008	OUT	00501

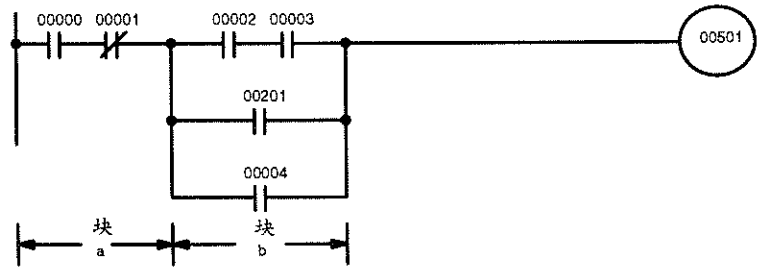
地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	LD	00004
00005	AND	00005
00006	OR LD	——
00007	OR LD	——
00008	OUT	00501

同样地，采用上图右边的方法，可被结合的逻辑块最多为 8 个，而采用第一种办法，可被结合的逻辑块数没有限制。

AND LD 和 OR LD 指令的结合

只要被结合的逻辑块数不超过 8 个，则使用“AND LD”及“OR LD”指令，上述两种方法都可采用。

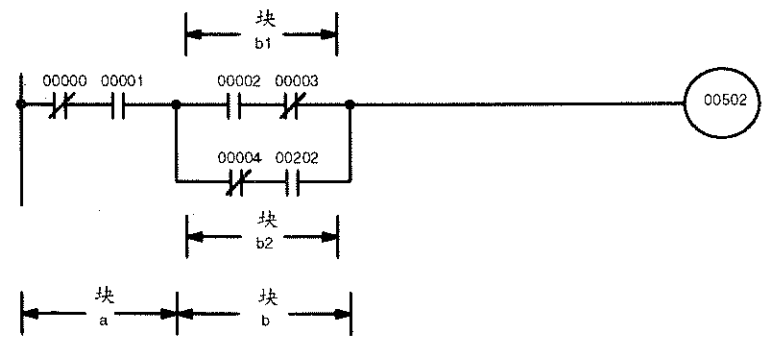
下面梯形图仅包含两个逻辑块。不需要将逻辑块 b 进一步分割，因为只要使用“AND”和“OR”指令便可对其编写。



地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR	00201
00005	OR	00004
00006	AND LD	——
00007	OUT	00501

尽管下面梯形图和上面一个很相似，但如果不把图中逻辑块 b 分割成两个用“OR LD”结合的逻辑块，那么就不能对其编写。在该例中，首先分为三个逻辑块，接着使用“OR LD”指令来结合最后的两个块，接下去是“AND LD”指令，结合由“OR LD”产生的执行条件和逻辑块 a 的执行条件。

在组合逻辑块末尾，一起使用逻辑块指令编写时，必须按相反顺序进行，就是说，首先对最后两个逻辑块编写逻辑块指令，接下来对产生于第一个逻辑块指令的执行条件以及倒数第三个逻辑块的执行条件进行编写，然后继续同样步骤，直到第一个组合的逻辑块为止。



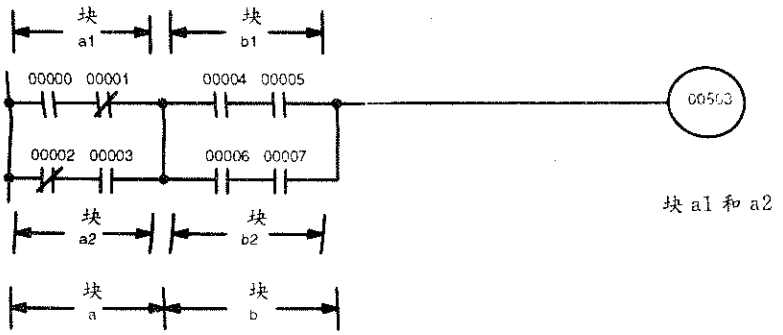
地址	指令	操作数
00000	LD NOT	00000
00001	AND	00001
00002	LD	00002
00003	AND NOT	00003
00004	LD NOT	00004
00005	AND	00202
00006	OR LD	——
00007	AND LD	——
00008	OUT	00502

复杂的梯形图

在决定一个梯形图中必须编写逻辑块指令时，常常需要将梯形图分割成大的逻辑块，接着连续对大的逻辑块分割，直到分割的若干逻辑块可以不需要逻辑块指令。接着对这些逻辑块编写，将小的逻辑块连接，再连接大一些的逻辑块。用“AND LD”指令，或者用“OR LD”指令连接逻辑块，这两条指令总是来连接存在着的最后两个执行条件，而不管这些条件是来自单个条件、逻辑块或来自于先前的逻辑块指令。

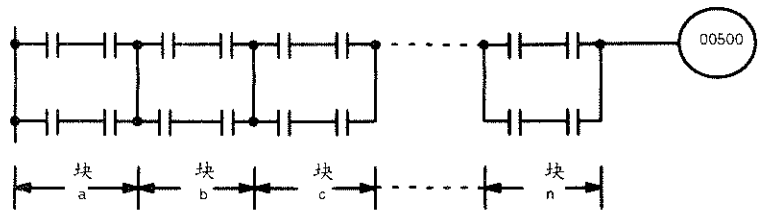
在编写复杂梯形图时，对逻辑块的编写开始于顶部左端，而且在移至另一逻辑块前，先在本块内下移。这通常意味着在可以选择时，先用“OR LD”编写，再用“AND LD”编写。

下面的梯形图必须被分割成两个大块，然后每个大块又被分割成两个小块，这样才可被编写。如下所示，块 a 和块 b 需要一个“AND LD”操作，而在此操作之前，必须使用“OR LD”操作将两边的顶部和底部块分别连接起来，即分别将 a_1 和 a_2；b_1 和 b_2 连接起来。

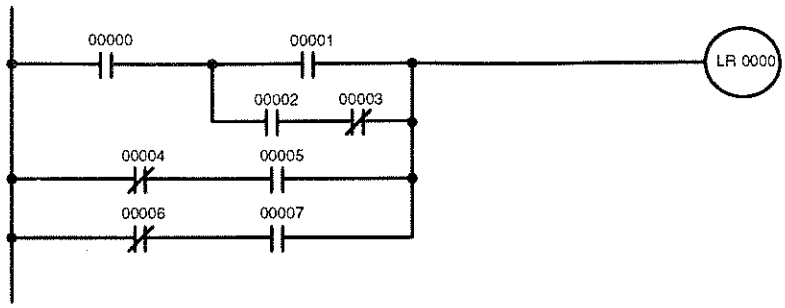


地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND	00003
00004	OR LD	——
00005	LD	00004
00006	AND	00005
00007	LD	00006
00008	AND	00007
00009	OR LD	——
00010	AND LD	——
00011	OUT	00503

可以较方便地对下面类型的梯形图编写，其前提是按顺序对每个逻辑块编写，即：先上→下，再左→右。下图中，要使用“AND LD”操作连接块 a 和 b，如上例所示。然后对块 c 编写，接着使用第二个“AND LD”操作将其和产生于第 1 个“AND LD”的执行条件连接起来。再接下来，要对块 D 编写，然后使用第 3 个“AND LD”操作将产生于块 D 的执行条件和来自第 2 个“AND LD”操作的执行条件连接起来，持续这一过程直至块 n。

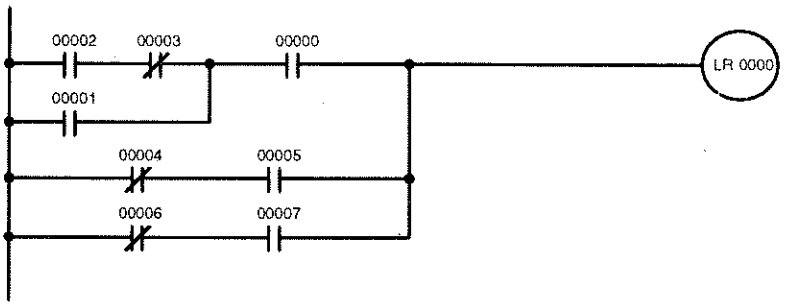


下面梯形图先需要一个“OR LD”操作，紧接着是一个“AND LD”操作，这样对顶部三个逻辑块编写，然后再用另两个“OR LD”操作来完成向助记符转化。



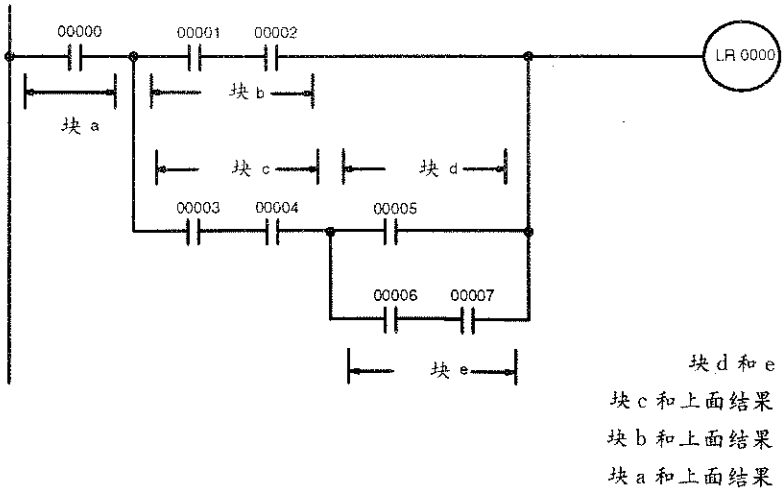
地址	指令	操作数
00000	LD	00000
00001	LD	00001
00002	LD	00002
00003	AND NOT	00003
00004	OR LD	——
00005	AND LD	——
00006	LD NOT	00004
00007	AND	00005
00008	OR LD	——
00009	LD NOT	00006
00010	AND	00007
00011	OR LD	——
00012	OUT	LR 0000

尽管程序可以按上面所写的执行，但该程序可以改写为如下形式，从而省去第一条“OR LD”和“AND LD”指令，以便简化程序、节省内存空间。



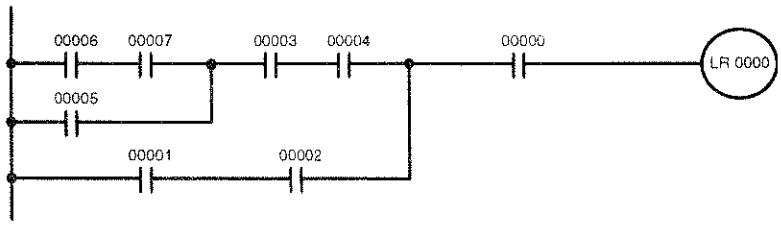
地址	指令	操作数
00000	LD	00002
00001	AND NOT	00003
00002	OR	00001
00003	AND	00000
00004	LD NOT	00004
00005	AND	00005
00006	OR LD	——
00007	LD NOT	00006
00008	AND	00007
00009	OR LD	——
00010	OUT	LR 0000

下面梯形图需要 5 个逻辑块，这里先按顺序对 5 个块编写，然后利用“OR LD”和“AND LD”指令从最后两个块开始往前将它们连接起来。程序中地址为 00008 的“OR LD”指令连接块 d 和 e，随后的“AND LD”指令将上述过程产生的执行条件和块 c 的执行条件连接起来……



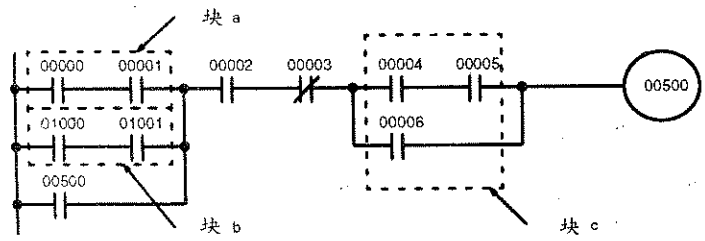
地址	指令	操作数
00000	LD	00000
00001	LD	00001
00002	AND	00002
00003	LD	00003
00004	AND	00004
00005	LD	00005
00006	LD	00006
00007	AND	00007
00008	OR LD	——
00009	AND LD	——
00010	OR LD	——
00011	AND LD	——
00012	OUT	LR 0000

与前述类似地，上面这个梯形图可以改写如下示，以便简化程序结构，简化编写并节省内存空间。

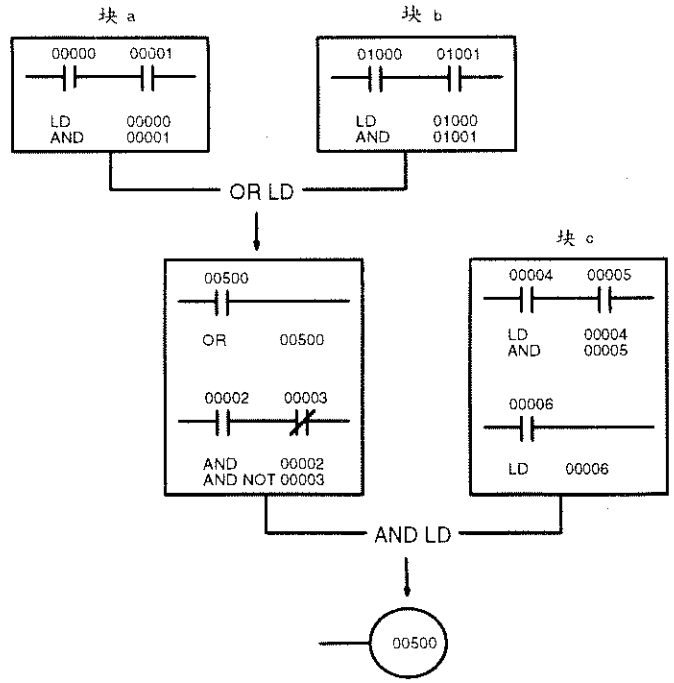


地址	指令	操作数
00000	LD	00006
00001	AND	00007
00002	OR	00005
00003	AND	00003
00004	AND	00004
00005	LD	00001
00006	AND	00002
00007	OR LD	——
00008	AND	00000
00009	OUT	LR 0000

下例及最后一例中梯形图看上去很复杂，但是仅用两条逻辑块指令即可对其编写。梯形图如下所示：



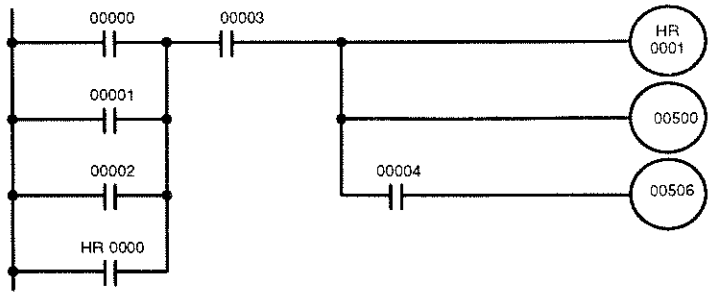
第一个逻辑块指令用来连接产生于块 a 和 b 的执行条件，第二条指令用来将逻辑块 C 的执行条件和产生于 IR00003 的常闭条件的执行条件连接起来。梯形图的余下部分可以通过“OR”、“AND”及“AND NOT”指令进行编写。上述过程的逻辑流程和编写结果如下所示。



地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	LD	01000
00003	AND	01001
00004	OR LD	——
00005	OR	00500
00006	AND	00002
00007	AND NOT	00003
00008	LD	00004
00009	AND	00005
00010	OR	00006
00011	AND LD	——
00012	OUT	00500

4－4－7 对多个右侧指令编写

如果有不止一条右侧指令按同一执行条件执行，那么从指令行上最后一个条件之后对这些指令连续编写。下面例子中，最后一个指令行还含有一个与 IR00004 AND 的条件。



地址	指令	操作数
00000	LD	00000
00001	OR	00001
00002	OR	00002
00003	OR	HR 0000
00004	AND	00003
00005	OUT	HR 0001
00006	OUT	00500
00007	AND	00004
00008	OUT	00506

4－5 编程器

本节和下节介绍编程器以及必须为输入程序所做的准备操作。4－7“程序的输入、修改和检查程序”介绍了将程序输入内存的实际步骤。

尽管可以用编程器编写梯形图，但基本上用于支持“SSS(中文版)”操作，且对现场编辑和维护非常有用。编程器的主要功能如下：

1. 显示操作信息和诊断检查的结果。
2. 读、写梯形图，插入、删除指令，搜索数据或指令，以及监视 I/O 位的状态。
3. 监视 I/O 状态，强制置位/复位。
4. 在电源 ON 状态下，编程器可以连接上 PC 或从 PC 取下来。
5. 编程器可用于 C 系列可编程控制器。

6.支持“终端方式”,它允许显示 32 个字符信息。还支持键盘映射功能的操作。详细情况请参阅 5－25－6“终端方式指令”—TERM(—)”。

注意:编程器并不支持全部的 SSS(中文版)的操作,仅支持用于现场编辑和维护的 LSS 操作。

4－5－1 键盘

以键的颜色,按功能把编程器的键盘分为四个区:

白色:数字键

十个白色键用于输入数字型程序数据,如程序地址、数据区域地址及操作数值。它们也可与功能键(FUN)组合使用,输入带功能码的指令。

红色:CLR(清除)键

CLR(清除)键可清除显示及取消编程器的当前操作。

在开始编程并键入口令字(Password)时也使用清除键。按清除键可以取消任何编程器操作,尽管可能要按 2、3 次去取消操作或消除显示。

黄色:操作键

黄色键用于写入和修改程序。有关这些键功能的详细解释在本节的后面。

灰色:指令和数据区域键

在输入或修改程序时,除了右上端的 SHIFT 键,灰色键用于输入指令、指定数据区域前缀。SHIFT 键与打字机的上挡键很相似,它用于改变下一个被按键的功能。(不必按住 SHIFT 键不放,只要按一次该键,然后按要用的键。)

除了“SHIFT”键,灰色键有指令的助记符名称,或者有数据区域的缩写(这些名称写在键上。)这些键的功能介绍如下。



通过功能码输入一指令时,在功能码之前按该键。



按该键输入 SFT(移位寄存器指令)



在一功能码后面输入以指定一条微分型式指令,或在一条梯形图指令后输入,以指定一个相反条件。



按该键以输入 AND(与指令)或与 NOT 使用以输入 AND NOT



按该键以输入 OR(或指令)或与 NOT 合用以输入 OR NOT



按该键以输入 CNT(计数指令)或指定一个已被定义为计数器的 TC 号



按该键以输入 LD(取指令)或与 NOT 合用以输入 LD NOT(取非指令)。按该键也可指示一个输入位。



按该键以输入 OUT(输出指令)或与 NOT 合用以输入 OUT NOT(输出非指令)。按该键也可指示一个输出位。



按该键以输入 TIM(定时指令)或指定一个已被定义为定时器的 TC 号

TR 在 TR 区域(暂存继电器区域)指定一个地址之前按该键

LR 在 LR 区域(链接继电器区域)指定一个地址之前按该键

HR 在 HR 区域(保持继电器区域)指定一个地址之前按该键

SHIFT HR 在 AR 区域(辅助继电器区域)指定一个地址之前按这两个键

DM 在 DM 区域(数据存储区)指定一个地址之前按该键

SHIFT DM 在 EM 区域指定一个地址之前按这两个键

CH * 在指定一个数据存储器间接地址之前按该键

SHIFT CH * 在指定一个字地址之前按这两个键

CONT # 在指定一个操作数作为一个常量之前按该键

SHIFT CONT # 在指定一个位地址之前按这两个键

SHIFT 在块编程指令的功能码前按该键,这些功能码位于尖括号＜＞之间

4－5－2 PC 方式

编程器上装有转换开关以控制 PC 主机的工作方式。用方式开关可分别选择三种工作方式:运行、监视、编程。所选择的方式将确定 PC 主机的操作及工作步骤。

RUN(运行)方式用于运行程序。当方式开关设在“RUN”,且 CPU 电源接通时,则 CPU 开始执行程序内存的程序。在 RUN 方式下,尽管通过编程器可监视 PC 操作,但不能输入或改变内存区域的数据。

监视方式下,可以直接监视程序运行状况,控制 I/O 状态,改变当前值(PV)或设定值(SV)等,此方式下,对 I/O 的处理与在运行方式(RUN)下相同。该方式通常用于系统的试运行及最后程序的调整。

“编程”方式下,PC 不执行程序。该方式用于编制,修改程序,清除存贮器区域以及登记和改变 I/O 表。编程方式还提供一种特殊的“调试”操作,以便在系统试运行之前检查程序以便修正。

注意

运行方式下,不要用加长电缆连接编程器和 PC,因为加长电缆所产生的噪声会进入 PC,影响程序和被控系统。

4－5－3 信息显示选择开关

CPU 上 DIP 开关的第 3 插脚决定编程器显示的信息为日文或英文。出厂时该开关置在“ON”,此时显示信息为英文。

4－6 操作准备

本节介绍了使编程器工作的步骤,包括输入口令字、清除内存、错误信息解除和编制 I/O 表。在其它一

些场合也要编制 I/O 表。例如在改变 PC 单元配置时。

注意

当给连着编程器的 PC 通电时，一定要确认编程器置于“编程”方式，除非需要用其它方式实现某种特定功能。PC 通电时，如果编程器在“运行”方式下，就会执行程序，而可能会导致受 PC 控制的系统开始工作。

下面的操作顺序必须在开始输入程序之前完成。

1．将方式转换开关的钥匙插入编程器

2．将方式开关置在“编程”方式(设在编程方式时，钥匙拔不出来)

3．接通 PC 电源

注意：如果安装了 I/O 单元，也接通其电源，否则编程器不工作。

4．确认 CPU 的电源(Power)灯亮，并且在编程器屏幕上出现如下的显示。(如果 ALM/ERR 灯亮或闪烁，或者有错误信息显示，清除发生的错误)

```
<PROGRAM>
PASSWORD!
```

5．输入口令　详见 4－6－1“输入口令字”

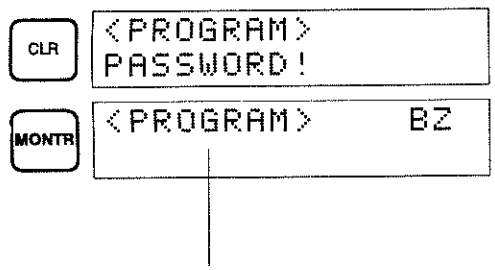
6．清除内存　如果不需要清除程序，则跳过该步。详见 4－6－3“清除内存。”

4－6－1　输入口令字

要获得利用 PC 编程功能的权利，首先必须输入口令字，口令字可防止未经授权而进入程序。

编程器连上 PC 后，当 PC 电源接通或已接通时，PC 会要求输入口令字。当“PASSWORD!”出现时，按“CLR”键，接着按“MONTR”键，进入系统，然后按“CLR”键以清除显示。

如果安装编程器时，PC 电源已接通，则下面的第一个显示表明的是安装编程器之前 PC 的状态。“输入口令字之前确认 PC 处于编程方式”。此时输入口令字，则 PC 的工作方式会随方式开关而定。如方式开关已设置“运行”或“监视”，则 PC 开始投入工作，即输入口令字后，在方式开关作用下，PC 会进入“运行”或“监视”工作方式。



表示由方式选择开关设定的工作方式

4－6－2　蜂鸣器

输入口令字后，或改变工作方式后，紧接着先后按下“SHIFT”键和“1”键可以切换蜂鸣器声音 ON/OFF。在按动编程器键盘时，蜂鸣器会发声。如果在显示器右上角显示“BZ”，说明蜂鸣器工作，否则，蜂鸣器不工作。

一旦 PC 工作出错，蜂鸣器也发声，蜂鸣器因 PC 工作出错而发出声响，不受上面设置的影响。

4－6－3　清除内存

内存清除操作可以全部或部分清除 UM 区域(RAM 或 EEPROM)和 IR、HR、AR、DM、EM 及 TC 区域的内容。除非另作特别说明，清除操作会全部清除上面的所有内存区。如果写保护开关(CPU 的 DIP 开关中第 1 插脚)置于“ON”，则不清除 UM 区域。

在初次进行编程或安装一个新程序时，通常要清除全部内存。在清除内存前，要检查内存是否有需要的

程序，如果有，则仅清除不需要的内存区域。记住在使用该内存中程序之前，必须顺序按键检查它。有关检查程序的步骤将在本节后面介绍。更进一步的调试方法，参阅第七章“程序的监视和执行”。为了清除全部内存区域，连续按“CLR”键直到显示全为 0，接着按照下面所给的最上面一行全清除击键顺序输入。下面的支线为部分清除按键顺序。部分清除、内存清除下面介绍。

只有在编程方式下才可清除内存。下表说明了三种清除内存操作中（全清除、部分清除、内存清除），分别清除哪些存贮的区域。

存贮器区域	全清除	部分清除	存贮器清除
I/O 字	清除	清除	清除
工作字	清除		清除
HR、AR、TC、DM、固定 DM	清除	清除	清除
扩展 DM	清除		清除
EM	清除	清除	清除
I/O 注释	清除		
梯形程序	清除	清除	清除
UM 分配信息	清除		

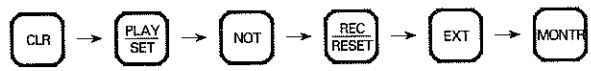
注：1. 在清除 DM 区域时，错误信息记录区域（DM6000～DM6030）将不被清除 2。

2. 当 PC 设置区域（固定 DM 区域中 DM6600～DM6655）被清除时，则设定值将变回出厂设定缺省值。

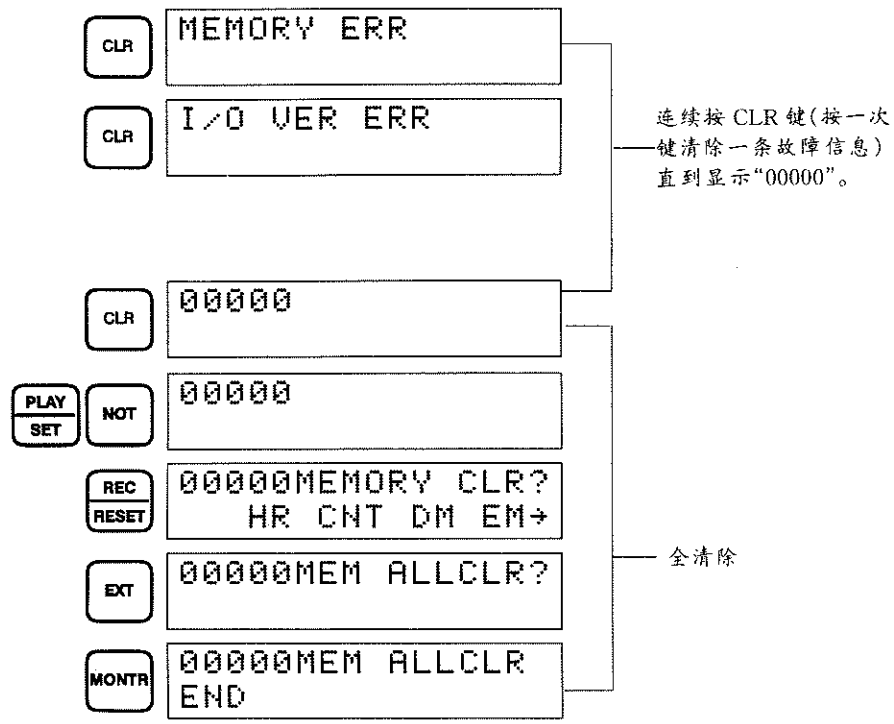
3. 全清除后，梯形程序区域将完全供梯形图程序使用，（扩展 DM 区域、I/O 注释区域被设为 0KW）。所有 EM 组也被清除。

全清除

全清除的键操作顺序如下



下列操作过程用于清除所有存储器

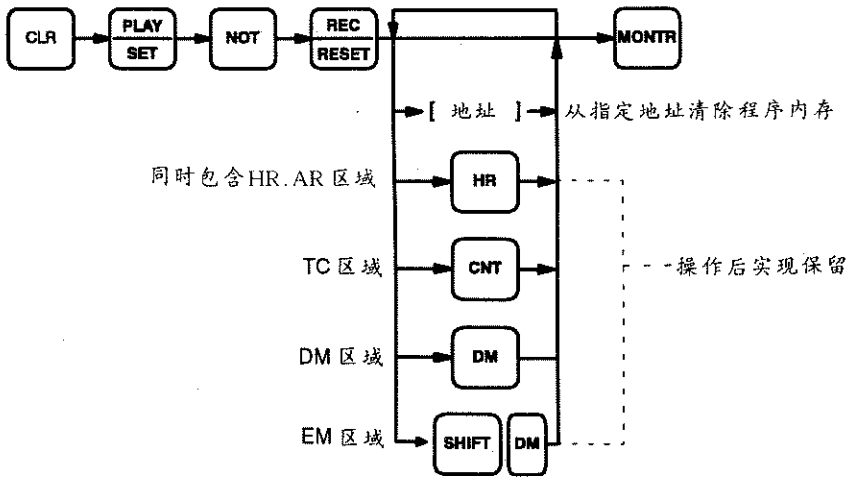


部分清除

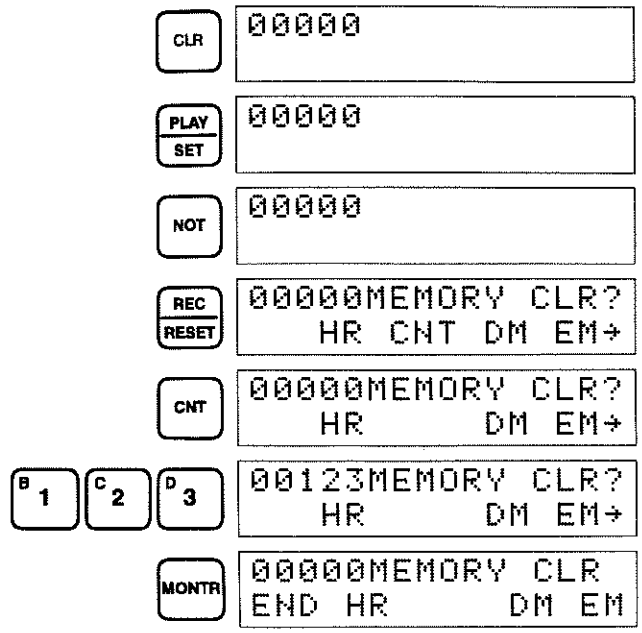
可以保留特定区域数据和部分梯形程序。要保留在 HR、AR、TC、DM 和/或 EM 区域的数据，则在键入 REC/RESET 键后再按适当的键。按“HR”键指定 HR、AR 区域。换句话说，若保留 HR 区域，同时也保留了 AR 区域，如果没有指定保留，则两种区域都将被清除。“CNT”键用于保留整个 TC 区域。按“SHIFT”键和“DM”键指定 EM 区域。屏幕会显示将被清除的内存区域。

可以保留一些 EM 组而清除其它组。参见 90 页“清除选择的 EM 组”。可以保留从开始到一特定地址的部分梯形程序，方法是设定要保留的数据区域后，指定要清除程序的起始地址。例如，要保留地址为 00000～00122 的部分程序，清除地址从 00123 到结束的程序内存，则输入 00123。

“部分清除”的键操作如下所示。



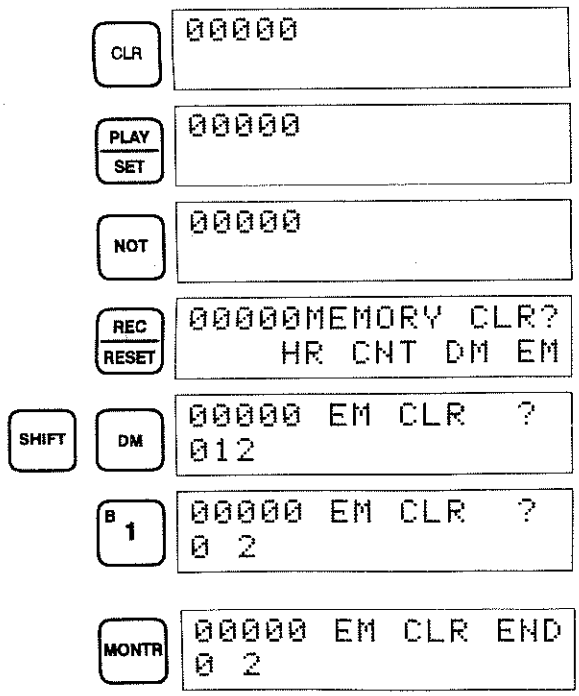
保留 TC 区域且保留程序内存地址为 00000～00122 的内容，输入如下：



清除选择的 EM 组

在执行部分清除操作时，可以选择特定 EM 组而不是全部 EM 组被清除。在下例中，选择清除 EM0 组和 2 组。

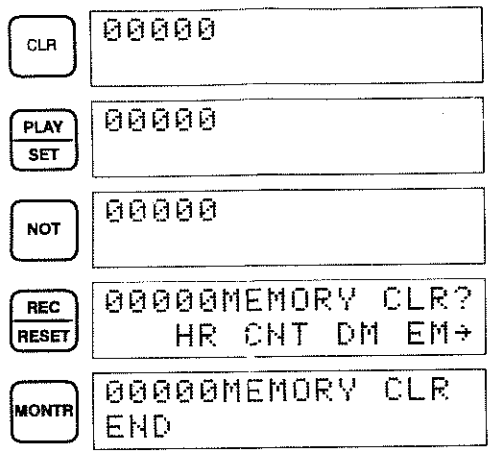
编程器上屏幕显示如下：



内存清除

“内存清除”将清除全部内存区域(I/O 注释，UM 分配信息除外)。

编程器屏幕显示如下：



注意：当写保护开关(CPU 的 DIP 开关中插脚 1)设在“ON”时，将保留 UM 区域(从 DM6144 到梯形程序)，其它数据区域，如 HR、AR、CNT 和 DM 区域(从 DM0000～DM6143)将被清除。

4－6－4 登记 I/O 表

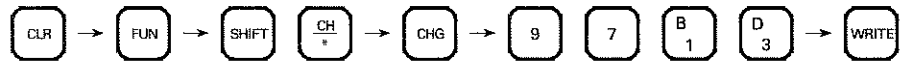
“登记 I/O 表”操作能记录受 PC 控制的 I/O 单元的类型及 I/O 单元的机架位置，也会清除全部 I/O 位。

C200HX/HG/HE 型机没有必要一定进行“登记 I/O 表”的操作。如果未登记 I/O 表，则在接通电源时 PC 将按所安装的 I/O 单元进行操作。不会发生 I/O 检验/设置错误。

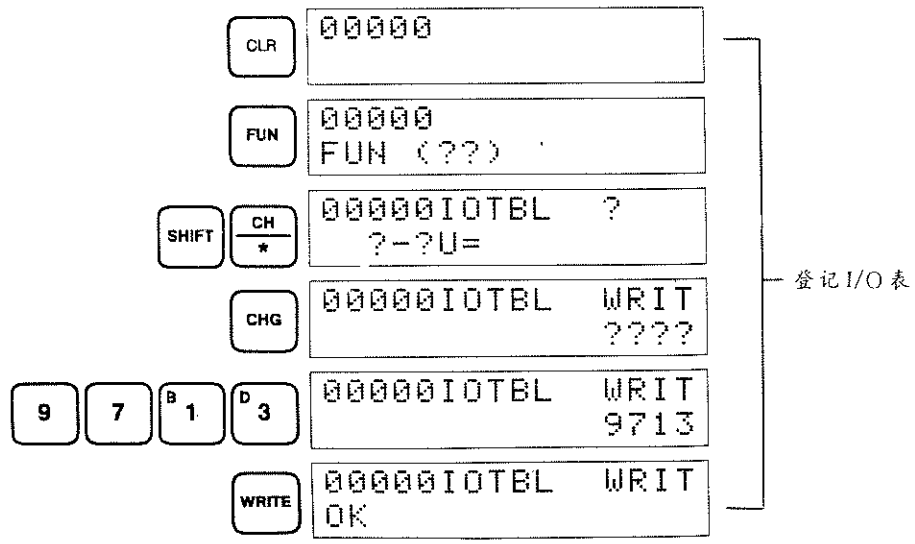
如果改变 I/O 单元，必须登记 I/O 表，否则开始编程时，会有 I/O 检验错误信息出现，即“I/O VER ERR”或“I/O SET ERROR”。

只有在编程方式下，且写保护开关(CPU 的 DIP 开关插脚 1)设在“OFF”(OFF＝可写入)，才能登记 I/O 表。

键操作如下：



初始 I/O 表登记



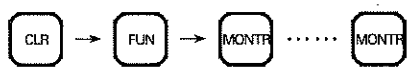
4－6－5 清除错误信息

完成了登记 I/O 表后，记录在内存中的错误信息应该被清除。此处假定已经注意到引起错误的原因(该错误导致错误信息的出现)。当要清除一错误信息，且听到“哔”的声音，则排除错误原因，然后清除错误信息(参阅第九章“故障检查”)

要显示记录的错误信息，按“CLR”、“FUN”和“MONRT”键。第一个信息会出现在屏幕上，再按“MONTR”键将清除当前的信息且显示下一个出错误信息。连续按“MONRT”键直到清除了所有错误信息。

尽管在任何方式下都能显示严重故障的错误信息，但只有在“编程”方式下才可以消除它们。

键操作顺序

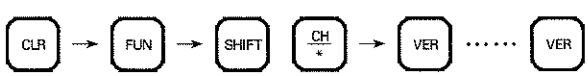


4－6－6 检验 I/O 表

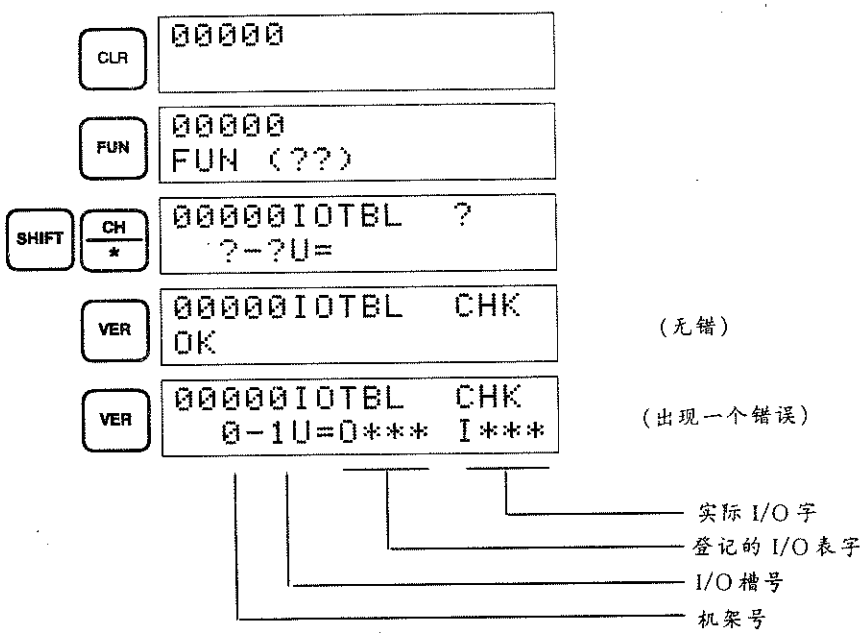
检验 I/O 表是为了检查内存中已登记的 I/O 表是否与实际的 I/O 单元排列一致。第一个不一致显示如下。每按一下“VER”键会显示下一个不一致。

注：只有当已登记了 I/O 表后才可执行此操作。

键操作顺序：



示例：



显示的含义

下面显示表明，在一个远程 I/O 从站机架上，C500、C1000H/C2000H 和 C200H/C200HS 或 C200HX/HG/HE 有相同单元号。

```
00000I/OTBL CHK
  *-*U=----
```

下列显示表明光缆型 I/O 单元中单元号重复

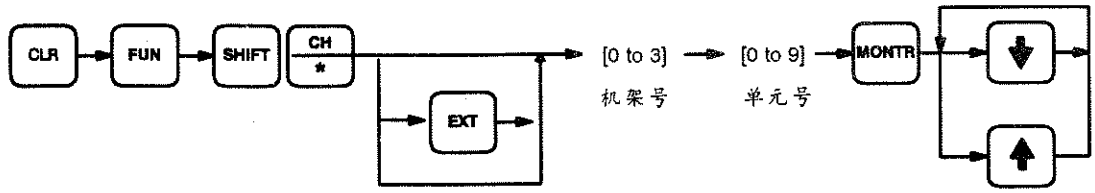
```
00000I/OTBL CHK
 2**HU=R*-I  R*-W
```

表明单元号重复

4－6－7 读 I/O 表

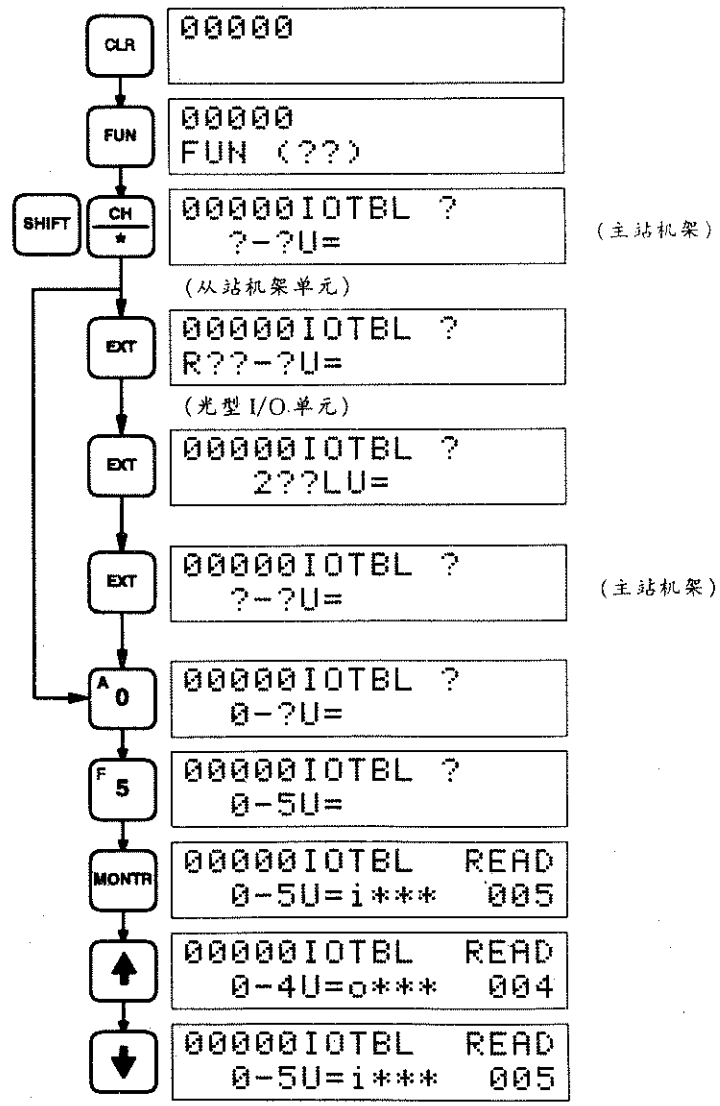
读 I/O 表是为了获得在 CPU 内存中当前登记的 I/O 表，该操作可在 PC 的任何方式下进行。

键操作顺序



按 EXT 键，选择远程 I/O
从站机架或光型 I/O 单元

例



显示的含义

显示指定的 I/O 单元(参阅远程从站机架中的 I/O 单元, P98)

C500, 1000H/C2000H I/O 单元

I/O 点的数量	输入单元	输出单元
16	I * * *	O * * *
32	I I * *	O O * *
64	I I I I	O O O O

C200H, C200HS I/O 单元

I/O 点的数量	输入单元	输出单元
8	i(*)* *	o * * *
16	i i * *	o o * *

注意:(*)为"i"表示无严重错误或 F

I/O 单元

```
00000IOTBL   READ
  *-*U=****   ***
```

- I/O 字号
- I/O 类型:i(输入)o(输出)
- 单元号(0—9)
- 机架号(0—3)

中断输入单元

```
00000IOTBL   READ
  *-*U=****
```

- INT0:安装在 CPU 机架
- INT1:安装在 I/O 扩展机架

特殊 I/O 单元

```
00000IOTBL   READ
  *-*U=$***
```

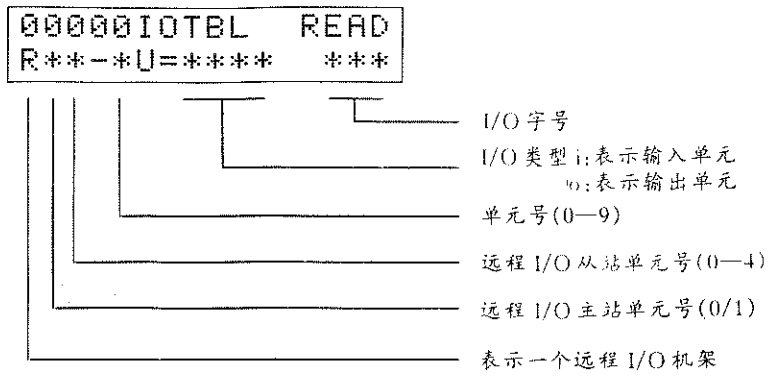
- 空:唯一的单元 1
- W:唯一的单元 2
- 特殊 I/O 单元类型
  - C:高速计数器
  - N:位置控制单元
  - A:其它
- 单元号(0—F)
- 表示特殊 I/O 单元

远程 I/O 主站单元

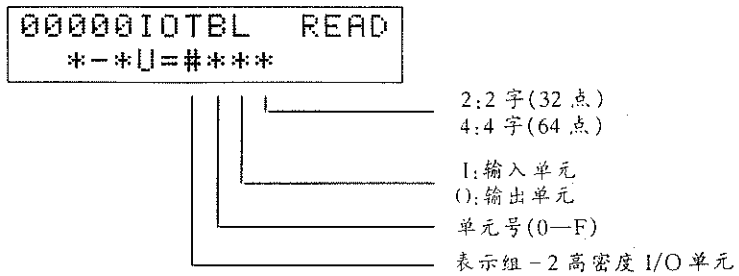
```
00000IOTBL   READ
  *-*U=RMT*
```

- 远程 I/O 主站号:(0 或 1)

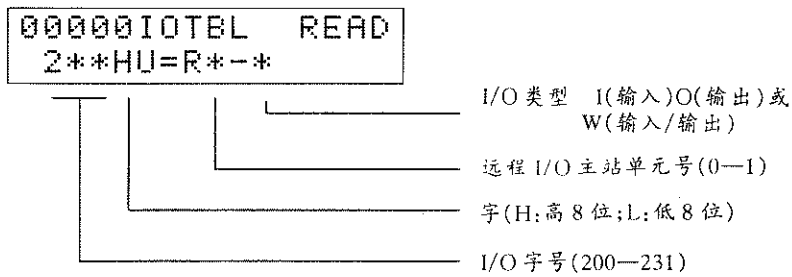
远程 I/O 从站机架



组－2 高密度 I/O 单元



光缆型 I/O 单元和远程终端



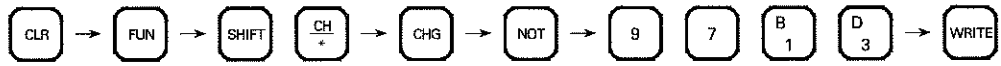
4－6－8 清除 I/O 表

如果想删除当前在 CPU 存贮单元中的 I/O 表的内容，可以使用 I/O 表清除操作。该操作执行后，PC 就可设置为所安装的 I/O 单元。

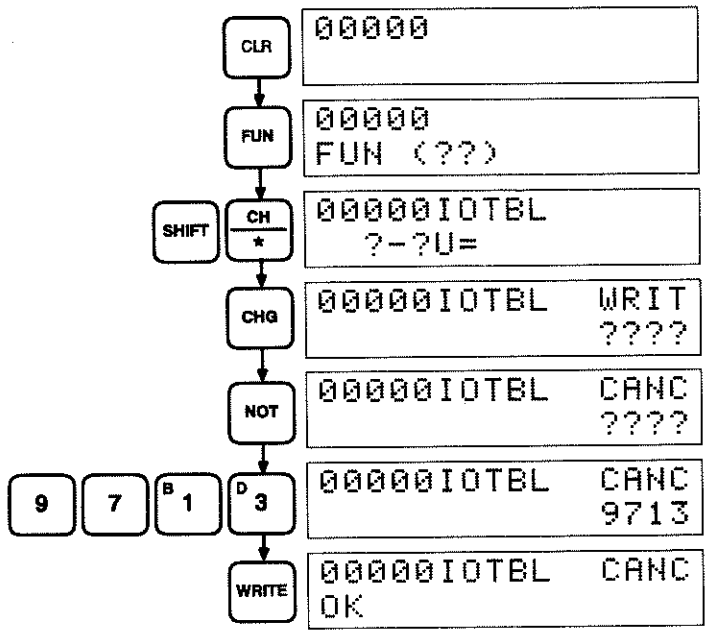
I/O 表清除操作，同时会对所有安装的特殊 I/O 单元和链接单元复位。当上位机链接单元，PC 链接单元，远程 I/O 主站单元，高速计数器单元，位置控制单元或其它特殊 I/O 单元在运行时，不可使用 I/O 表清除操作。

注意：这一操作只能在编程方式下进行，并且要把写保护开关(CPU DIP 开关的插脚 1)置成 OFF(OFF ="可写")

键操作顺序



例：



4－6－9 SYSMAC NET 链接表传送

SYSMAC NET 链接表传送操作能将 SYSMAC NET 链接数据链接表复制到 UM 程序存贮器。这就允许用户程序和 SYSMAC NET 链接表一起写到 EPROM 中。

数据链接表必须通过 SYSMAC 支持软件 SSS(中文版)生成，在拷贝到程序存贮器之前，先传送到 PC。

数据链接表从 SYSMAC 支持软件传送到 PC 后，被存贮在 RAM 内。当后备电池耗尽，表也将丢失。为了防止丢失，建议将程序(含链接表)写入 EPROM 或 EEPRON 存贮器内。

注

当电源加到 PC 上，而这个 PC 在它的程序存贮器里已经有 SYSMAC NET 链接表的拷贝，这个 CPU 的 SYSMAC NET 链接表会被复盖。SYSMAC NET 链接表的改变不影响程序存贮器内 SYSMAC NET 链接表的复制；要改变程序存贮器中的拷贝，必须重复传送 SYSMAC NET 链接表。

如果 SYSMAC NET 链接表传送操作不进行，是因为：

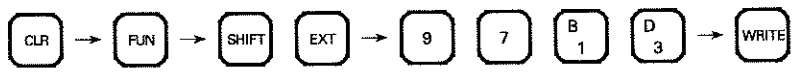
·存贮器单元不是 RAM，或 EEPROM，或写保护开关未设置在“可写入”位置。

·没有 END(01)指令。

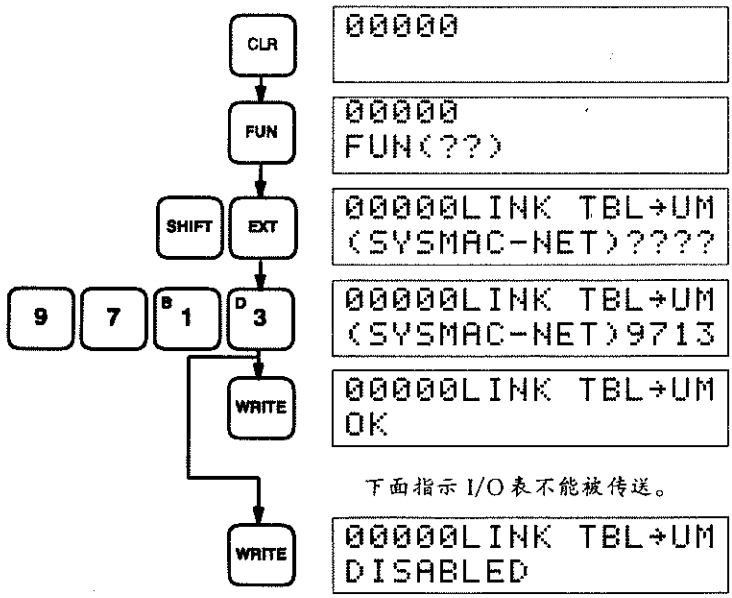
·程序存贮器的容量超过 14.7K 字。当存贮器分配给扩展 DM 或 I/O 注释区域时，程序容量将减少。在 END(01)指令后，为存贮数据链接表，大约需要 0.5K 字程序存贮空间。

SYSMAC NET 链接表只能在编程方式下传送。

键操作顺序



例



4－7 程序的输入、修改和检查

只要程序是用助记符编写的，就可以从编程器直接输入到 PC 中。助记符是从编程器键入程序存贮地址的。程序的检查包括语法检查，它检查程序是否按语法规则编写。一旦语法错误被改正，就可开始试运行，最终可在实际操作情况下，完成修正。

输入程序所必须的操作将在下面介绍。本节还将介绍如何修改已存入内存的程序，以及如何得到当前周期时间的步骤。

在输入程序以前，应先检查是否已经装入了别的程序，如果已装入不需要的程序，先用清除程序存贮器的操作步骤将它清除，然后输入新程序，如果还需要以前的程序，应该用程序检查键检查它，按需要修改它。更深入的程序调试方法将在第七章“程序的监视和执行”中介绍。

4－7－1 从程序存储器地址中读写：

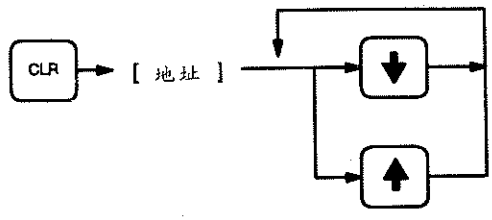
第一次输入程序时，一般是从程序存储器的地址 00000 开始写起，因为每次清屏时都会显示该地址，所以没必要去寻找。

如果不是从地址 00000 开始输入程序或读出、修改已存入存储器的程序，那么就必须指定所要求的地址。指定地址的操作为：先按“CLR”，再输入要求的地址。地址前面的“0”可以省略，例如要指定地址为 00053，只需输入 53 即可，在按了“向下键”后，指定地址中的内容才被显示。

按了“向下键”显示指定地址的内容后，还可以按“向上键”或“向下键”来显示整个程序存储器的内容。每按一次键，程序存储器中的上一个或下一个字就会显示出来。

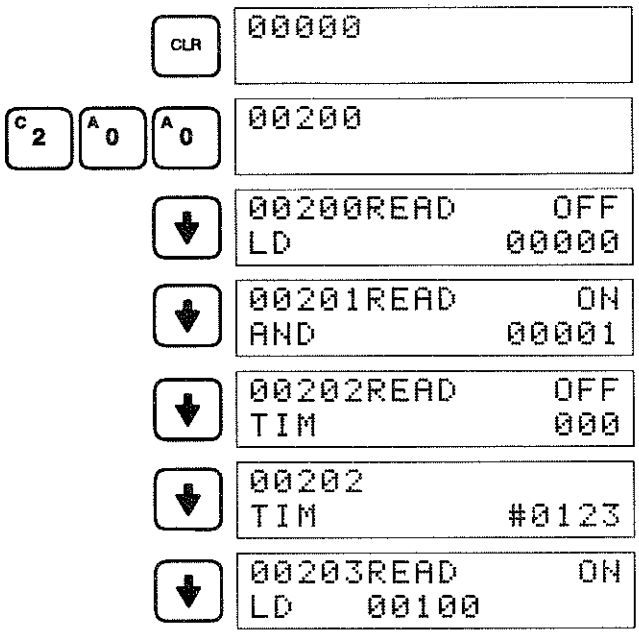
如果在“运行”或“监视”方式下读程序存储器，那么每个被显示“位”(bit)的 ON/OFF 状态也将显示出来。

键操作顺序



例

如果以下助记符已被输入程序存储器，那么按以下的键会产生图示的显示。



地址	指令	操作数
00200	LD	00000
00201	AND	00001
00202	TIM	000
		#0123
00203	LD	00100

4－7－2 输入和编辑程序

只有在编程方式下，并且写保护开关(CPU 的 DIP 开关中的插脚 1)置于 OFF(OFF＝“可写入”)的情况下，才可以输入并编辑程序。

第一次输入程序和编辑一个已存在的程序的步骤是完全一致的。在两种情况下，都是程序存储器中的当前内容被覆盖，只不过在没有旧程序的情况下，在每个地址中被覆盖的是 NOP(00)指令。

确认设置了正确的地址后，开始输入程序，一步步地输入梯形图中产生的助记符。一旦显示正确的地址。则输入第一个指令字并按“WRITE”键。接下来输入所要求的操作数，每输入一个操作数，按一下“WRITE”键，即在每一行助记符结束按“WRITE”键，只有这样，指令和操作数才会被送入，下一个显示才会出现。如果指令需要二个或更多的字，则接下来显示的是所要求的操作数并为它提供一个缺省值。如果指令仅需一个字，则显示下一个地址。连续输入每一行助记符，直到程序全部送入为止。

当输入操作数的数值时，不必输入字首的“0”。只有当输入功能代码时，才需输入字首的零(见后面)。在指定操作数时，必须为所有操作数指定数据区域(但 SR 和 IR 的地址除外)，而指定数据区域时是通过按对应的数据区域键完成的。用“CONT/＃”键来指定每个常数。对于计数器或定时器的 SV 则不必按此键(见下面)。AR 区域的指定是先按 SHIFT 键，再按 HR 键。TC 号作为位操作(即：结束标志)数被指定时，则在地址前按“TIM”键或“CNT”键，这取决于 TC 号是被用于定义一个定时器还是一个计数器。要指定一个间接 DM 地址，在地址前按 CH/＊键。(间接 DM 地址不必按 DM 键)。

为计数器和定时器输入 SV(设定值)

尽管可以输入含有设定值(SV)字的地址的方法来输入设定值，但通常是将常量赋给计数器或定时器。当作为常量输入设定值时，不需按“CONT/＃”键，只要送入数值再按“WRITE”键即可。若要通过指定字的方法来输入 SV，则是按“CLR”键，再输入字的地址即可。

指定指令

最基本的指令可用编程器上的键来输入。所有其它的指令使用功能代码输入。这些代码总是跟在指令的助记符后面。如果指令的助记符后面没有功能代码，则编程器上有该指令的键。在功能代码后按“NOT”

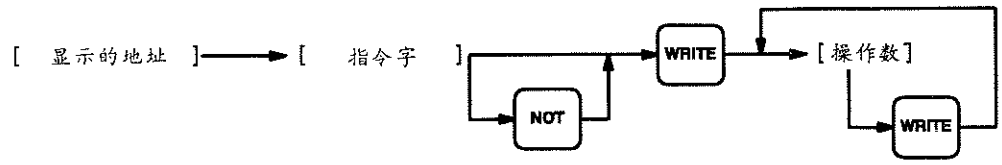
键可以指定指令的微分形式。

用功能代码输入指令时，先设置地址，按 FUN 键，再输入包括字首的 0 在内的功能代码。如果需要指令的微分形式，则按 NOT 键，接着输入指令所需的每个位操作数或定义，最后按“WRITE”键。

注意

细心输入功能代码，必要时按“SHIFT”键

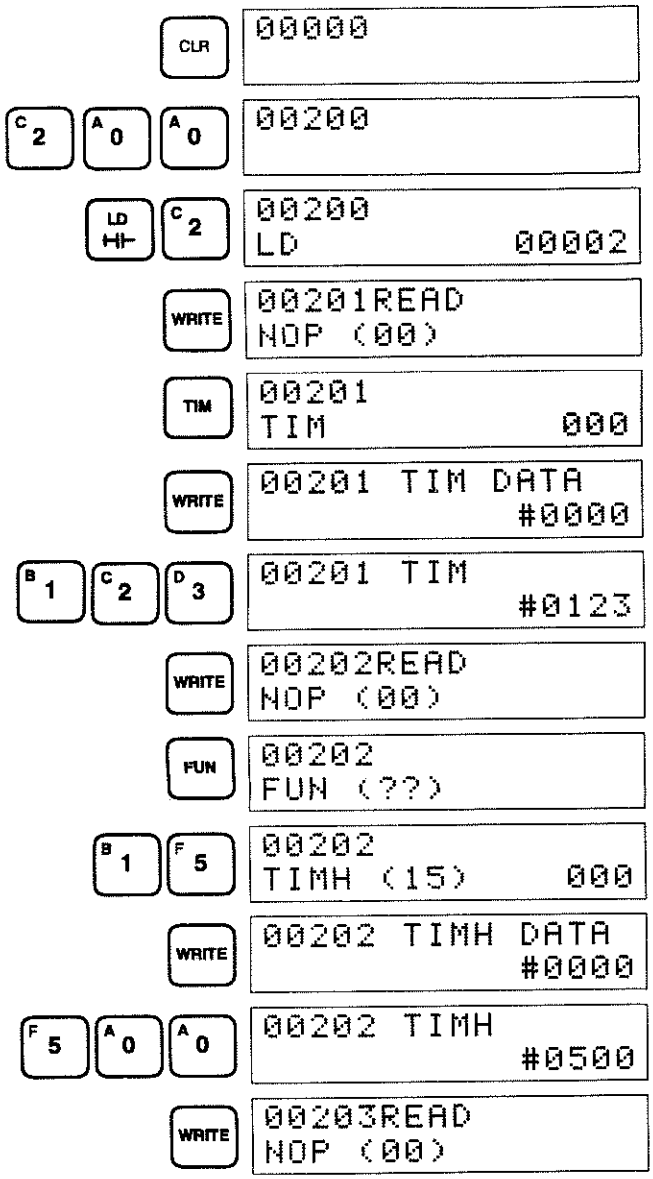
键操作顺序



例

使用所示的键可以输入以下程序。显示内容如下所示。

地址	指令	操作数
00200	LD	00002
00201	TIM	000
	TIMH	# 0123
00202		000
	(15)	# 0500



错误信息

当输入程序时，可能会出现以下错误信息。改正指出的错误后继续输入程序。下表中的星号代表实际显示出来的数值或地址。

信 息	原因和更正
* * * * REPL ROM	试图对具有写保护的 RAM 或 EEPROM 进行写操作。确信写保护开关置于 OFF。
* * * * PROG OVER	在存贮器中最后一个地址的指令不是 NOP(00)。清除程序末尾所有不需要的指令。
* * * * ADDR OVER	所置的地址超过了 UM 区域的最大地址。输入一个较小的地址。
* * * * SETDATA ERR	输入数据有格式错误或超过了规定的限制。例如，16 进制的值赋给了 BCD 码。重新输入数据。该错误会产生一个 FALS 00 的错误。
* * * * I/O NO. ERR	指定的数据区域地址超过了数据区域的限制，例如地址太大。确认指令所需的地址并重新输入它。

4－7－3 检查程序

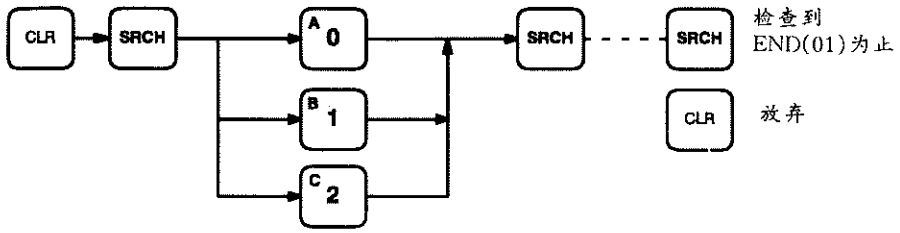
一旦输完了程序，就要检查语法，以验证它是否违反了程序规则。只要程序被修改，就有可能造成语法错误，也就必须要进行语法检查。

检查程序的按键顺序如下图所示。图中的数字表示检查级别(见下图)。当检查级别输入后，就开始对程序进行检查。如果发现了错误就停止检查，并显示错误。按“SRCH”键可使检查继续。如果未发现错误，就会检查到第一个结尾 END(01)，在每 64 个指令检查完后会显示信息。(例如：下表后的例子＃1 显示)

在检查开始后，可以按 CLR 键取消检查，对此在下面的例子中出现＃2 显示。当检查到第一个 END 时，会有如＃3 显示的信息。

只有在编程方式下才可以进行语法检查。

键操作顺序



(0,1,2 检查级别)

检查级别和错误信息

程序检查可以有三种级别。必须指定所要求的检查级别以表明要发现的错误类型。下表列出了错误类型，显示和各种语法错误的解释。0 级别检查 A、B、C 类的错误；1 级别检查 A、B 类的错误；2 级别只检查 A 类错误。

产生错误的地址也会显示出来。

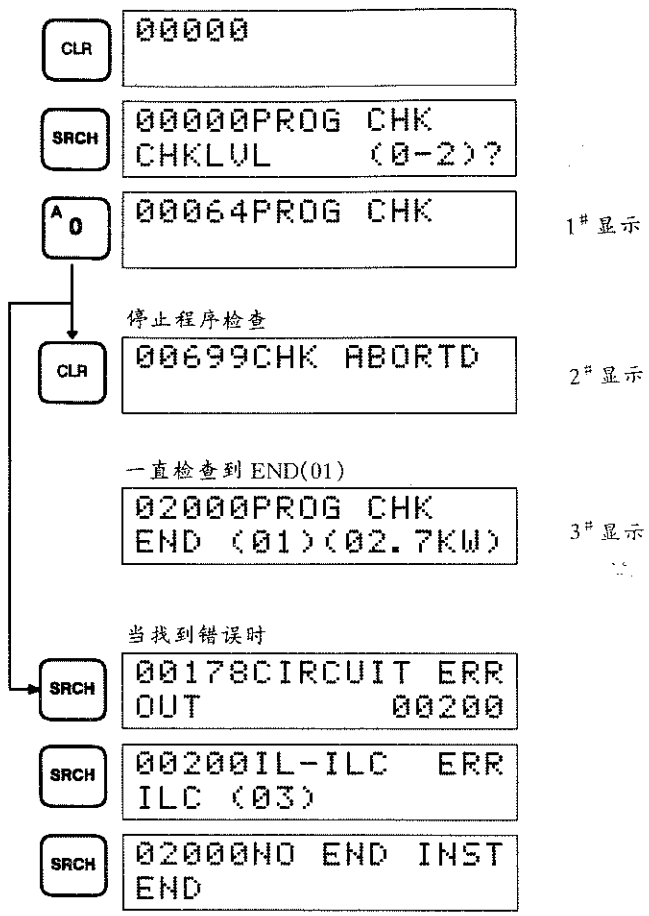
下表中的许多错误对应于还没详细说明的指令。具体细节可以参考 4－8“位状态控制”或第五章“指令集”。

类型	信息	含义及对应措施
A 类	?????	程序丢失，重新输入
	NO END INSTR	程序中没有 END(01)，在程序末尾地址写入 END(01)
	CIRCUIT ERR	逻辑块数与逻辑块指令不符。如用“LD”或“LD NOT”指令启动一个逻辑块，但该逻辑块的执行条件没有被其它指令使用。又如使用逻辑块指令却没有所需的逻辑块数。请检查程序。
	LOCN ERR	程序中某个指令的位置错误。检查指令的要求，修正程序。
	DUPL	同一跳转号或子程序号被使用两次。修改程序使得同一数字只被使用一次。(跳转号 00 可按需要多次使用)
	SBN UNDEFD	SBS(91)被用来为一个不存在的子程序号编程。修改子程序号或编写需要的子程序。
	JME UNDEFD	JMP(05)指令漏了 JME(04)指令。改正跳转号或插入适当的 JME(04)指令
	OPERAND ERR	为指令输入的常量不在定义的值域内。修改常数，使其在定义的值域内。
	STEP ERR	错误地使用了有分段号的 STEP(08)指令和无分段号的 STEP(08)。检查 STEP(08)指令的编程要求并修改程序。

类型	信息	含义及对应措施
B 类	IL－ILC ERR	IL(02)与 ILC(03)指令没有成对使用。修正程序使每个 IL(02)指令都有唯一的一个 ILC(03)指令。如果多个 IL(02)使用同一个 ILC(03)与其对应时，尽管会显示错误信息，但程序仍会按所写的执行。运行之前一定要确认程序是按要求编写。
	JMP－JME ERR	JMP(04)00 和 JME(05)00 没有成对使用。如果多个 JMP(04)00 与同一个 JME(05)00 对应使用，尽管此错误信息会显示，但程序仍会按所写的执行。在程序执行前确认它是按要求编写的。
	SBN－RET ERR	如果显示是 SBN(92)的地址，则有两个不同的子程序使用了同一子程序号。更改其中一个子程序的号或取消其中一个子程序。如果显示是 RET(93)的地址，则是 RET(93)未被正确使用。检查 RET(93)指令的编程要求并修改程序。
C 类	JMP UNDEFD	对同一跳转号，只使用了 JME(05)而未使用 JMP(04)指令。加入具有相同跳转号的 JMP(04)指令或删除未使用的 JME(05)指令。
	SBS UNDEFD	存在未被 SBS(91)调用的子程序。在适当位置编写一个调用子程序或删除不需要的子程序。
	COIL DUPL	多个指令(例 OUT，OUT NOT，DIFU(13)、DIFD(14)，KEEP(11)，SFT(10))控制同一位(置 ON 和/或 OFF)。尽管对一定的指令允许这种情况，但也要检查指令的要求以确保程序是正确的，或重新编写程序，使每个位只被一条指令控制。

例

以下的例子显示了程序检查过程中可能显示的一些信息。

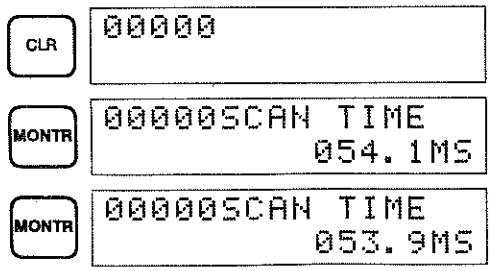


4－7－4 显示扫描周期时间

清除了程序中的语法错误后，应该检查它的扫描周期时间。这只有在“运行”或“监视”方式下运行程序时才有可能。有关扫描周期时间的具体细节可参看第六章“程序执行时间”。

要显示当前平均扫描周期时间，先按“CLR”键再按“MONTR”键。此操作后显示的是典型的扫描周期时间。显示值的差别取决于按下“MONTR”时的执行条件。

例



4－7－5 程序搜索

可在程序中搜索任意指定的指令或一条指令所使用的数据区域地址，搜索可以从当前显示的任意地址开始，或从清屏开始。

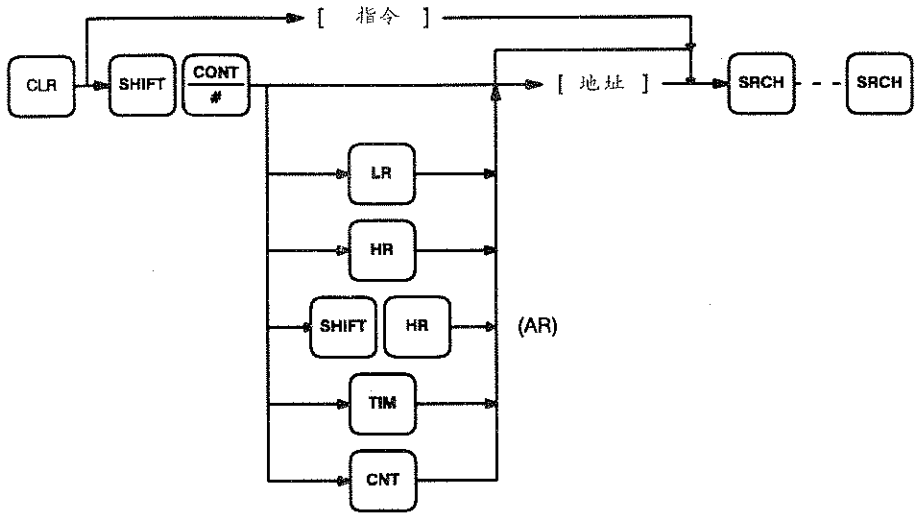
要指明一个位地址，先按“SHIFT”键，再按“CONT/＃”键，接着输入地址，包括所要指定的数据区域，再按“SRCH”键。

要指定一条指令，就象要输入程序时一样输入该指令，再按“SRCH”键即可，一旦指令或地址被找到，再按“SRCH”键，还可以找到其它相同的指令或地址。在搜索进行时，会显示“SRCH’G”。

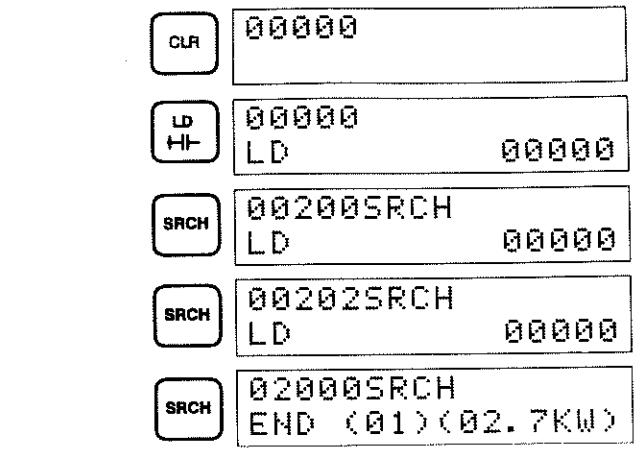
在搜索一个多字指令时，先显示指令的第一个字，在继续搜索前，按向下键即显示该指令的下一个字。

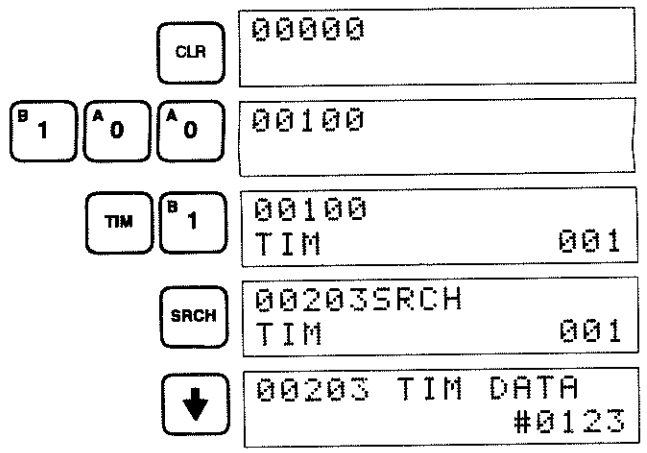
如果在“运行”或“监视”方式下读程序存储器，则每一位的 ON/OFF 状态也会显示出来。

键操作顺序

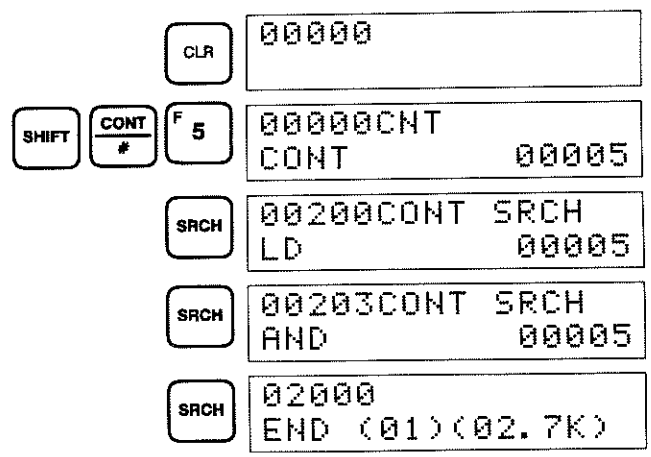


示例
指令搜索





示例
位搜索



4－7－6 插入、删除指令

在编程方式下，当前显示的任何指令都可以被删除，也可在它前面插入别的指令。这些操作只有在编程方式下，并且写保护开关(CPU 的 DIP 开关插脚 1)置 OFF 时才可以进行(OFF＝“可写入”)。

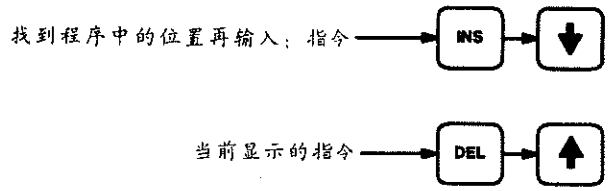
要在某指令前插入一条指令，先显示该指令，再象当初输入程序那样输入要插入的指令字，再按“INS”及“向下”键。如果该指令还需输入其它字，以输入程序的方式输入这些字即可。

要删除一条指令，显示该指令的指令字，再按“DEL”和“向上”键，则指定指令的所有字都将被删除。

注意

小心不要意外地删除指令，否则这些指令会永远地丢失，除非重新输入它们。

键操作顺序



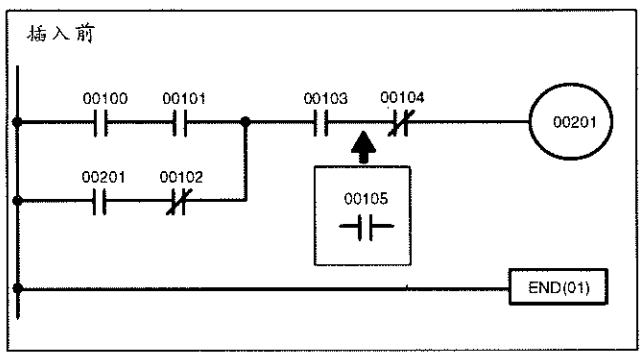
当删除或插入了一条指令后，程序存储器会自动调整后面指令的地址，就不会出现空地址或没有地址的指令。

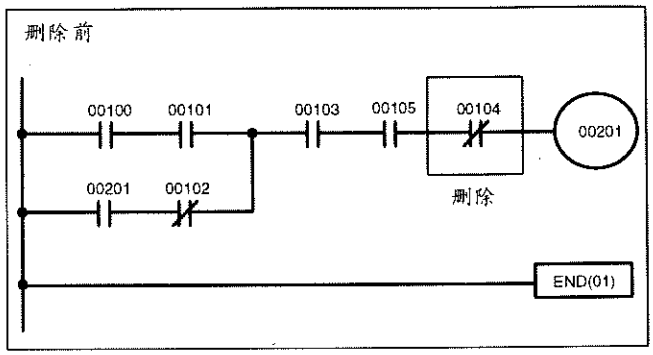
示例

以下助记符是表明顺序按键后程序的变更及相应的显示。

修改前程序

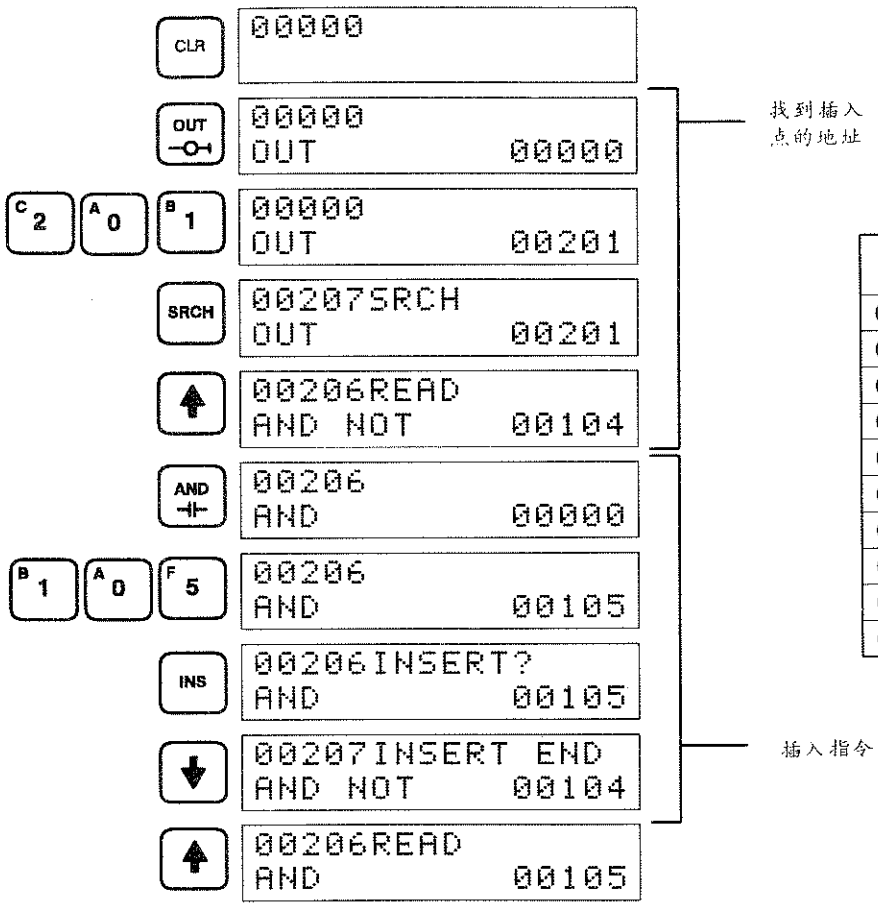
地址	指令	操作数
00000	LD	00100
00001	AND	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	—
00005	AND	00103
00006	AND NOT	00104
00007	OUT	00201
00008	END(01)	—





下面的键输入及显示说明程序按上面所示的改变的过程。

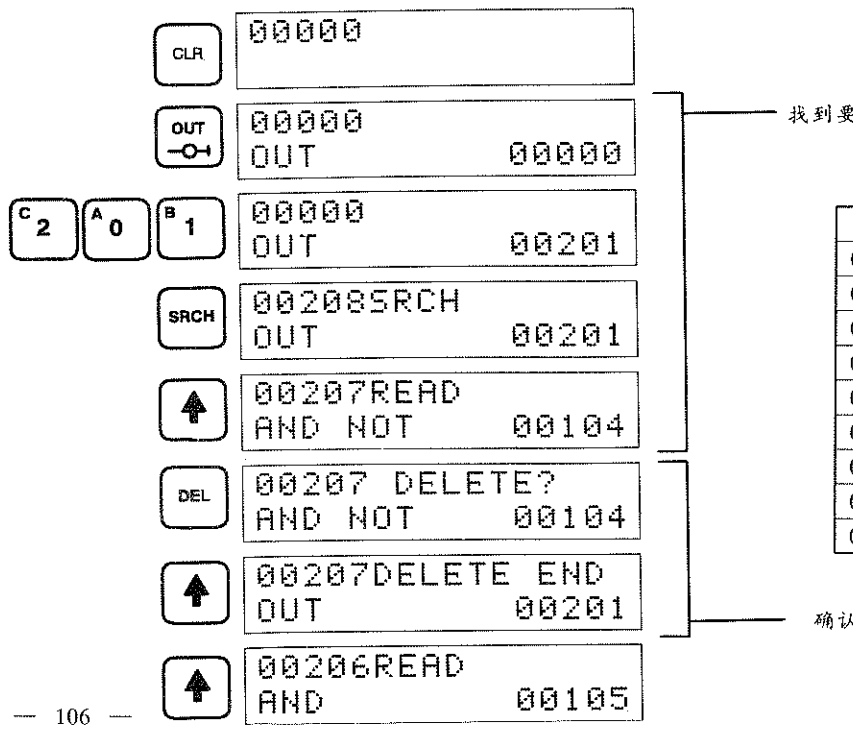
插入一条指令



插入后程序

地址	指令	操作数
00000	LD	00100
00001	AND	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	—
00005	AND	00103
00006	AND	00105
00007	AND NOT	00104
00008	OUT	00201
00009	END(01)	—

删除一条指令



删除后程序

地址	指令	操作数
00000	LD	00100
00001	AND NOT	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	—
00005	AND	00103
00006	AND	00105
00007	OUT	00201
00008	END(01)	—

4－7－7 指令行的分支

当一指令行分支成两行或更多行时，有时需使用联锁或 TR 位来保持分支点的执行条件。这是因为在返回分支点执行另一分支指令行前，至右侧指令的指令行已被执行。如果在指令行的分支点后面存在着一个条件，那么它就有可能改变执行条件，使程序不能执行应执行的操作。下图说明了这一点。在两张图中都是在返回分支点移到分支行去执行指令 2 之前，先执行了指令 1。

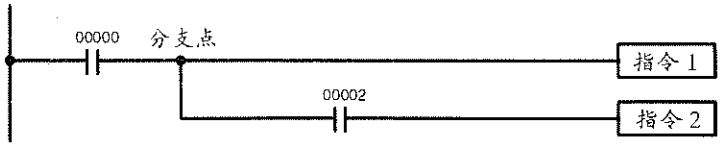


图 A：正确操作

地址	指令	操作数
00000	LD	00000
00001	指令 1	
00002	AND	00002
00003	指令 2	



图 B：错误操作

地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	指令 1	
00003	AND	00002
00004	指令 2	

图 A 中，在返回分支行前分支点处的执行条件不改变（右端的指令不改变执行条件），那么分支行就会被正确执行，因而不需要采取特殊的编程措施。

图 B 中，在分支点和顶部指令行的最后一条指令之间存在一个条件。那么在执行完顶部指令行后的执行条件和分支点的执行条件会有不同，这样就不能保证分支行的正确执行。

在编分支程序时有两种方法保持执行条件。一种用 TR 位，另一种是用联锁（IL(02)/ILC(03)）。

TR 位

TR 区域提供 8 个位（TR0 至 TR7），可用来暂时存储执行条件。如果一个 TR 位被设置于一个分支点处，则当前的执行条件就会存贮在指定的 TR 位中。当程序返回到分支点时，TR 位释放出执行状态，而该状态正是在程序第一次执行到该分支点处保存下来的。

前面的图 B 可用如下写法，以保证程序正确执行。在助记符中，把 TR 位当作“OUTPUT”指令的操作数，这样分支点的执行条件即可存入 TR 位中。在执行完（分支点）右边的指令后，再把该 TR 位当作“LD”指令的操作数，这样，这个执行条件又被释放出来。

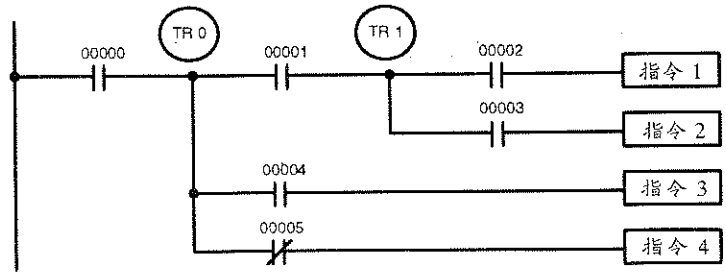


图 B：用 TR 位修正

地址	指令	操作数
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	指令 1	
00004	LD	TR 0
00005	AND	00002
00006	指令 2	

根据实际指令，上图应是如下过程，用一个 LD 指令装入 IR00000 的状态，从而建立初始执行条件。在分支点，用一个 OUTPUT 指令将该执行条件送入“TR0”存贮起来。接着执行条件和 IR00001 的状态进行“AND”操作，接着相应地执行指令 1。然后存贮在分支点的这个执行条件被调用（以 TR0 为操作数的 LD 指令），和 IR00002 的状态进行“AND”操作，相应地执行指令 2。

下面的例子说明了使用两个 TR 位的应用。



地址	指令	操作数
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	OUT	TR 1
00004	AND	00002
00005	指令 1	
00006	LD	TR 1
00007	AND	00003
00008	指令 2	
00009	LD	TR 0
00010	AND	00004
00011	指令 3	
00012	LD	TR 0
00013	AND NOT	00005
00014	指令 4	

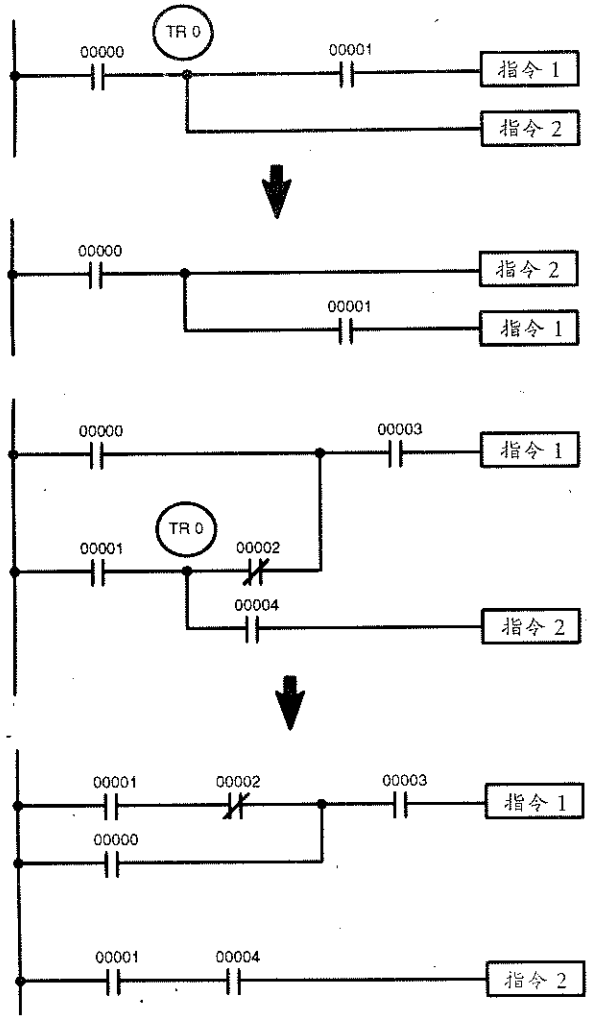
在本例中，TR0 和 TR1 用来存贮两个分支点的执行条件。执行完指令 1，TR1 内的执行条件被调出和 IR00003 的状态进行、“AND”操作。而存在 TR0 中的执行条件会被两次调用，第一次调出来和 IR00004 的状态进行 AND 操作，第二次则是与 IR00005 的状态的“非”进行 AND 操作。

TR 位可按要求多次使用，但在同一指令块中，不可多次使用同一 TR 位。程序每次开始执行新的指令块时都要返回母线，如果在单个指令块中，具有 8 个以上要存贮执行条件的分支点，那么必须使用联锁操作。（后面介绍）

在画梯形图时，除非必需一般不用 TR 位。画梯形图时不使用 TR 位往往可以减少程序的指令数，并使程序更易于理解。在下面两对梯形图中，后面的方法既不用 TR 位又减少了指令数。在第一个例子中，是通过重新组合指令块实现的；第二个例子是通过将第二个 OUTPUT 指令分开，并使用另一个 LD 指令来为它产生适当的执行条件实现的。

注

虽然简化程序总是受到关注的，但是指令执行的顺序有时在程序中也是至关重要的。例如，在执行一条二进制加（BINARY ADD）指令前，或许会用一条移动（MOVE）指令把适当的数据放入所需的操作数字中。在考虑简化程序之前，一定要确认执行顺序全部正确。



注

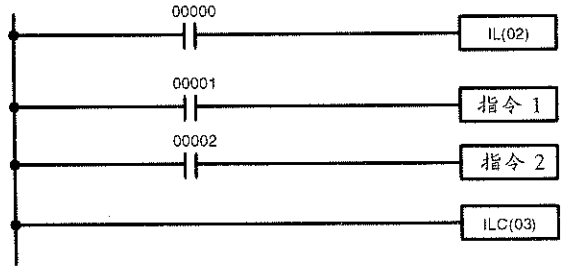
只有在用助记符编程时才使用 TR 位，在直接输入梯形图时，就不必使用 TR 位。但要注意在分支点所需 TR 位的最大数目的限制(8 个)，还要注意采用适当的方法以减少程序所使用的指令数。

联锁

使用 IL(02)(联锁)及 ILC(03)(解除联锁)指令也可以解决在分支点存贮执行条件的问题。IL(02)和 ILC(03)会完全消除分支点，它允许一个特殊的执行条件去控制一组指令。IL 和 ILC 指令总是在一起使用。

如果在一个梯形图的某一段前设置了 IL 指令，那么 IL 指令的执行条件将控制下一个 ILC 指令前的全部指令的执行。如果 IL 的执行条件是“OFF”，那么在该 IL 指令与下一个 ILC 指令之间的全部右侧指令都按“OFF”执行条件去执行，从而对梯形图的整个这一段复位。它对一些特定指令的影响将在 5—10“联锁及联锁解除指令——IL(02)、ILC(03)”中介绍。

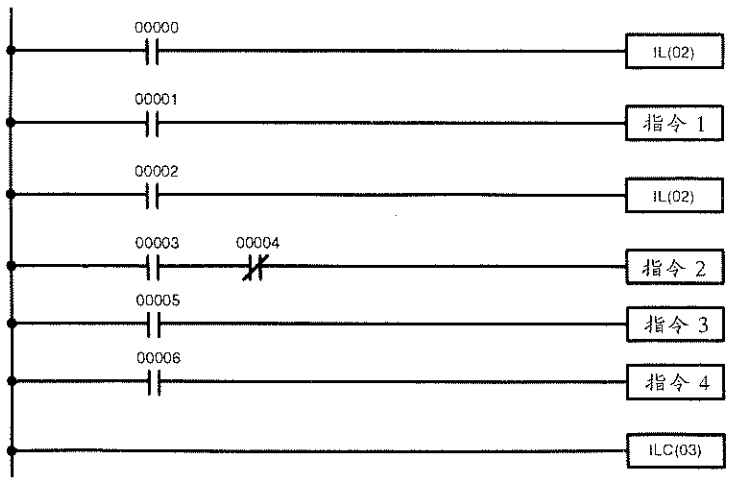
图 B 也可用联锁指令来修改。程序中分支点的执行条件被放在“IL”指令行上，所有原来从分支点分出的指令行重被写为独立的指令行，并且在最后另加一条 ILC 指令的指令行。ILC 指令行上不能有条件存在。注意 IL 和 ILC 指令都不需要操作数。



地址	指令	操作数
00000	LD	00000
00001	IL(02)	—
00002	LD	00001
00003	指令 1	
00004	LD	00002
00005	指令 2	
00006	ILC(03)	—

在上面修改过的程序中，如果 IR00000 的状态是 ON，那么 IR00001 及 IR00002 的状态将分别决定指令 1 和指令 2 的执行条件。因为 IR00000 的状态为 ON，那么它与各位的状态 AND 操作的结果是一样的。如果 IR00000 是 OFF，那么 IL 指令将为指令 1，指令 2 产生“OFF”执行条件，并从 ILC 指令后面的指令继续执行下去。

如下图所示，在一个指令中可以重复使用 IL 指令，每个指令的有效范围都在下一条 ILC 指令之前。



地址	指令	操作数
00000	LD	00000
00001	IL(02)	—
00002	LD	00001
00003	指令 1	
00004	LD	00002
00005	IL(02)	—
00006	LD	00003
00007	AND NOT	00004
00008	指令 2	
00009	LD	00005
00010	指令 3	
00011	LD	00006
00012	指令 4	
00013	ILC(03)	—

如果上图中 IR00000 为 OFF，(即如果第一条 IL 指令的执行条件是 OFF)那么指令 1 到指令 4 都按 OFF 执行条件执行且程序的执行会转到 ILC 指令后面的指令。如果 IR00000 是“ON”，则调出 IR00001 的状态作为指令 1 的执行条件，再调出 IR00002 的状态作为第二个 IL 指令的执行条件，如果 IR00002 的状态为“OFF”，则指令 2 到指令 4 将按 OFF 执行条件执行。如果 IR00002 为“ON”，则 IR00003、IR00005 和 IR00006 将确定新指令行中的第一个执行条件。

4－7－8 跳转

根据一个指定的执行条件，可以跳过程序中某一指定段。尽管在将 IL(联锁)指令的执行条件置“OFF”时也可类似实现这一功能，而使用跳转可以维持所有指令的操作数的状态，因此跳转常被用于控制需要保持连续输出的器件，比如气动装置，液压传动装置，而 IL(联锁)指令常用于控制不需要连续输出的器件中，比如电子仪器。

使用跳转(JMP(04))和跳转结束(JME(05))指令可实现跳转。如果一个跳转指令的执行条件是 ON，则程序就象没有跳转指令一样运行。如果跳转指令的执行条件是 OFF，则程序会立即跳转到跳转结束指令后继续执行，同时并不改变跳转与跳转结束指令间的各个状态。

跳转与跳转结束指令的跳转号范围是 00～99。有两种类型跳转，其类型决定于使用的跳转号。

在 01～99 范围内取号的跳转只能定义一次，即每个跳转号只能使用一次(跳转、跳转结束指令各一次)。当执行一条具有某一个跳转号的跳转指令时，程序会立即跳转至具有相同跳转号的跳转结束指令，就像这两条指令间不存在指令。在 TR 位和 IL(联锁)部分作的例子的图 B 可以使用跳转方法改写如下所示。尽管本例中跳转号用的是 01，其实可以任意选择 01～99 之间的数——只要它没有被程序的其它部分使用过。跳转、跳转结束指令不需其它另外的操作数，跳转结束指令行上也没有条件。

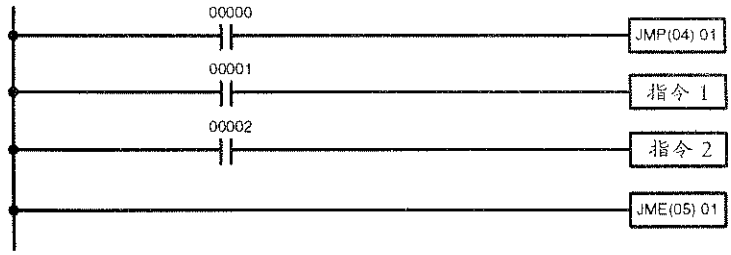


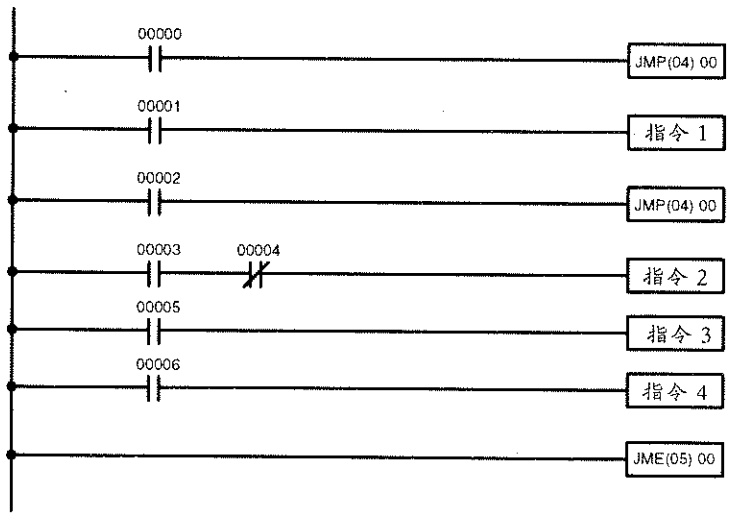
图 B：用一条 JUMP 指令修正

地址	指令	操作数
00000	LD	00000
00001	JMP(04)	01
00002	LD	00001
00003	指令 1	
00004	LD	00002
00005	指令 2	
00006	JME(05)	01

当 00000 为“OFF”时，这种图 B 程序的执行时间比任何其它方法编的程序都要短。

如果跳转号为 00，则是另一种类型跳转。用 00 作为跳转号可以产生所需要的许多次跳转。它允许多条跳转指令连续使用 00 作为跳转号，而它们之间没有以 00 为跳转号的跳转结束指令。它甚至允许所有 JUMP 00 指令将程序跳转至同一条 JUMP END 00 指令处，即在程序中所有的 JUMP 00 指令共用同一条 JUMP END 00 指令。当 00 作为一条跳转指令的跳转号时，程序会跳转至紧跟着的以 00 作为跳转号的跳转结束指令后的指令处。尽管同其它跳转指令一样，程序会跳过 JUMP 00 至 JUMP END 00 之间的所有指令，且不改变状态。但因为程序要用一定的时间寻找下一个 JUMP END 00 指令，所以它的执行时间比其它跳转指令稍长些。

程序中多条 JUMP 00 指令共用一条 JUMP END 00 指令，其执行与使用联锁(IL)指令的程序相似。下图与上面采用联锁的例子相同，只不过用 JUMP(跳转)方法改写而已。该图的执行与前述采用联锁的图不同(例如，前面图中联锁将会使联锁段中部分复位，而跳转则对 JUMP 至 JUMP END 指令之间的任何位不产生任何影响)。



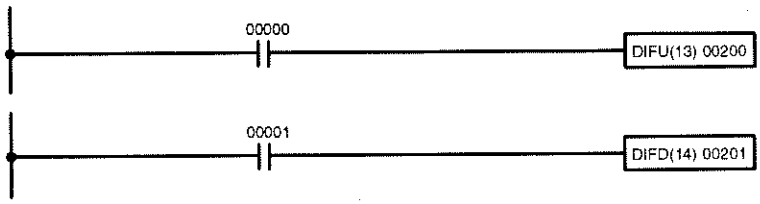
地址	指令	操作数
00000	LD	00000
00001	JMP(04)	01
00002	LD	00001
00003	指令 1	
00004	LD	00002
00005	JMP(04)	00
00006	LD	00003
00007	AND NOT	00004
00008	指令 2	
00009	LD	00005
00010	指令 3	
00011	LD	00006
00012	指令 4	
00013	JME(05)	00

4－8 位状态的控制

有5条指令可用来控制独立的位状态。它们是：OUTPUT、OUTPUT NOT、DIFFERENTIATE UP、DIFFERENTIATE DOWN和KEEP指令。所有这些指令都是以一个指令行的最后一条指令出现，它使用一个位地址作为操作数。尽管有关细节将在5－9“位控制指令”中介绍，但此处还是先介绍一下这些指令(除了已介绍过的OUTPUT和OUTPUT NOT指令)，因为它们在编程中非常重要。尽管这些指令常用于将IR区域的输出位置“ON”和“OFF”(即对外部设备送出或停止输出信号)，但也可用来在IR区域或其它数据区域控制其它位的状态。

4－8－1 上升沿微分/下降沿微分指令(DIFU/DIFD)

上升沿微分和下降沿微分指令用来使操作数位在一个周期时间内为ON。DIFU指令在执行条件OFF→ON时，使操作数位在一个周期时间内“ON”；DIFD指令在执行条件由ON→OFF时，使操作数位在一个周期内“ON”。这两条指令均只需一行助记符。



地址	指令	操作数
00000	LD	00000
00001	DIFU(13)	00200

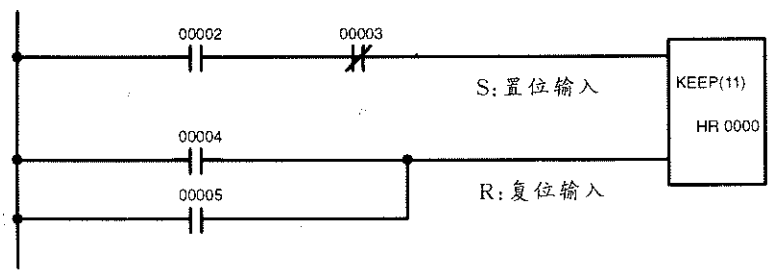
地址	指令	操作数
00000	LD	00001
00001	DIFD(14)	00201

此例中，IR00000变为ON时，IR00200在一个周期时间内ON。下一个周期时间DIFU(13)00200指令执行后，不论IR00000的状态怎样，IR00200均置“OFF”。当IR00001变为OFF时(此前，IR00201一直保持“OFF”)，在DIFD指令的作用下，IR00200在一个周期时间内ON，而在下一个周期时间执行了DIFD(14)00201指令后，IR00201又变为OFF。

4－8－2 保持(KEEP)指令

保持(KEEP)指令用来保持在两个执行条件下的操作数位的状态。要实现这一功能，保持指令要连接两条指令行。当第一条指令行末端的执行条件为“ON”时，保持(KEEP)指令的操作数位置“ON”，当第二条指令行末端的执行条件为“ON”，则“KEEP”指令的操作数位将置“OFF”。在梯形图中，保持指令的操作数的状态不会受联锁(IL)指令的影响。

下面的例子中，当IR00002“ON”，而IR00003“OFF”时，HR0000被置“ON”，它将保持“ON”状态直到IR00004或IR00005变为“ON”。对于保持指令来讲，和其它所有需要多条指令行的指令一样，它的指令行在被它们控制的指令之前先编写。



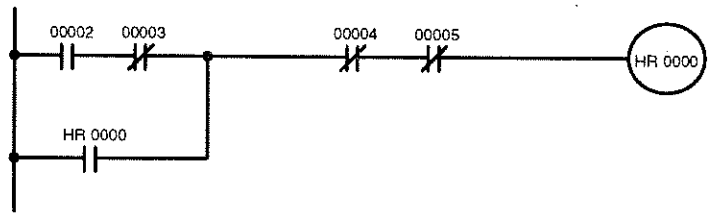
地址	指令	操作数
00000	LD	00002
00001	AND NOT	00003
00002	LD	00004
00003	OR	00005
00004	KEEP(11)	HR 000

4－8－3 自保持位(Seal)

尽管保持指令可以用来建立自保持位,但有时需要其它方法来建立自保持位,从而使它在 IL(联锁)的程序段中能被置"OFF"。

产生自保持位的方法是:将一条 OUTPUT 指令的操作数位作为同一条 OUTPUT 指令的"或"条件,这样 OUTPUT 指令的操作数位将一直保持着 ON 或 OFF,直到其它位发生改变为止。在 OUTPUT 指令前至少要使用一条其它条件用作复位功能。如果没有这个复位,就无法控制 OUTPUT 指令的操作数位了。

上面保持(KEEP)指令的图可改写如下。两张图的唯一差别在于当 IL(联锁)指令的执行条件为"ON"时,在联锁的程序段中的执行是不同的。与上面使用 KEEP 指令的图一样,本图中使用了两个"复位"位,即当 IR00004 或 IR00005 置"ON"时,HR0000 被置"OFF"。



地址	指令	操作数
00000	LD	00002
00001	AND NOT	00003
00002	OR	HR 0000
00003	AND NOT	00004
00004	AND NOT	00005
00005	OUT	HR 0000

4－9 工作位(内部继电器)

在编程中,将几个条件结合起来直接产生执行条件是很难的,但使用某些位间接触发其它指令,可以很容易地克服上述困难。使用工作位可以实现这种编程。为此有时会用到整个字,这些字称工作字。

PC 并不传送或接收工作位,它是编程者为了编程方便而自己选择的。如上所述,I/O 位和其它一些专用位不能用作为工作位。IR 区域内未被用作 I/O 位的所有位和 AR 区域未被使用的某些位可以用作工作位。注意保存一份有关怎样及何处使用了工作位的准确记录,它将有利于程序构思和编写,也有助于调试程序。

工作位的应用

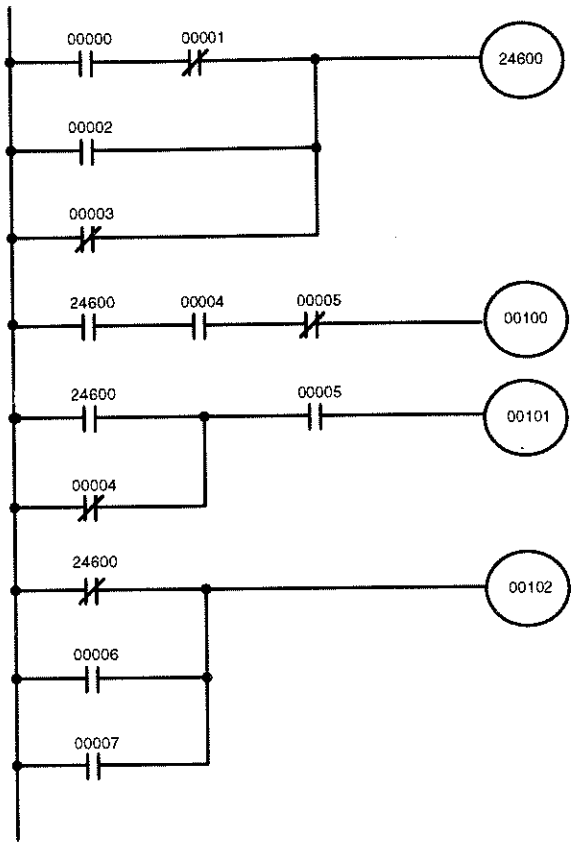
在本节后面的例子将介绍两种最通常的使用工作位的方法。但它们仅供参考,因为使用工作位的方法很多,不胜枚举。只要在编程中遇到控制方面的难题,就应想到使用工作位并考虑如何用它们以使编程简化。

工作位常使用的指令是:OUTPUT、OUTPUT NOT、DIFFERENTIATE UP、DIFFERENTIATE DOWN、和 KEEP 指令。工作位作为上述指令中其中之一的操作数位被首先使用,这样接下来它就可以作为条件来决定怎样执行其它指令。工作位还可与其它指令一起使用,例如移位寄存器(SFT (10))指令。工作位和工作字与 SFT(10)指令一起使用的例子在 5－15－1"移位寄存器——指令 SFT(10)"中介绍。

尽管未被作为工作位特别说明,但实际上在第五章"指令集"中的一些例子中许多用的位都是工作位。理解这些位的使用将有助于提高编程能力。

减少复杂条件

当一些条件的结合在与其它条件结合中重复出现时,可使用工作位来简化编程。下例中,IR00000、IR00001、IR00002 与 IR00003 合成一个逻辑块,用 IR24600 的状态存贮产生的执行条件,而 IR24600 又与其它各种条件结合来决定 IR00100、IR00101、IR00102 的输出条件,即决定这些位的输出置 ON 还是 OFF。

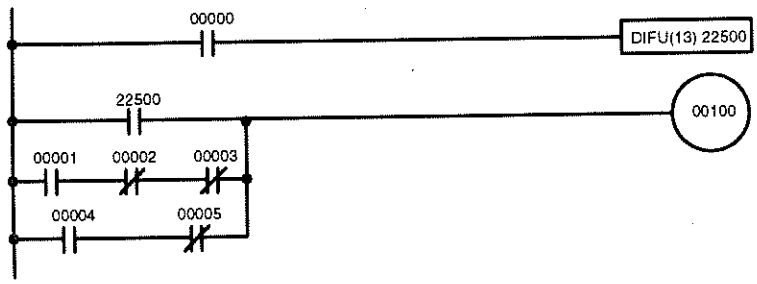


地址	指令	操作数
00000	LD	00000
00001	AND NOT	00001
00002	OR	00002
00003	OR NOT	00003
00004	OUT	24600
00005	LD	24600
00006	AND	00004
00007	AND NOT	00005
00008	OUT	00100
00009	LD	24600
00010	OR NOT	00004
00011	AND	00005
00012	OUT	00101
00013	LD NOT	24600
00014	OR	00006
00015	OR	00007
00016	OUT	00102

微分条件

如果一条指令的某些执行条件(不是全部)需要微分处理时,也可使用工作位。本例中,只要 IR00001 为 ON,同时 IR00002、IR00003 为 OFF,或者只要 IR00004 为 ON 而 IR00005 为 OFF,那么 IR00100 持续为 ON。每次当 IR00000 变为 ON 时,IR00100 置一个扫描周期时间 ON(除非上述条件之一一直使它为 ON)。

将 IR22500 作为微分指令(DIFU(13))的操作数的工作位,很容易实现该功能。当 IR00000“ON”,则 IR22500 在 DIFU(13)指令作用下,在一个周期时间内“ON”,下一周期时间变为“OFF”。如果假设控制 IR00100 的其它条件并不使它保持“ON”,则工作位 IR22500 将仅仅使 IR00100 在一个周期时间内“ON”。



地址	指令	操作数
00000	LD	00000
00001	DIFU(13)	22500
00002	LD	22500
00003	LD	00001
00004	AND NOT	00002
00005	AND NOT	00003
00006	OR LD	—
00007	LD	00004
00008	AND NOT	00005
00009	OR LD	—
00010	OUT	00100

4－10 编程注意事项

只要不超过 PC 的存贮容量，可用于串或并的条件数目并无限制。因此，用尽可能多的条件将程序图画清楚。尽管可以用指令行画出许多复杂的图，但严禁在两条指令行之间垂直线上加任何条件。例如下图 A 是错误的，应画成图 B 的形式。助记符只能用于图 B，对图 A 编码是不可能的。

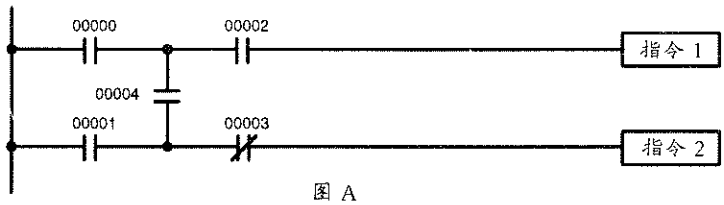


图 A

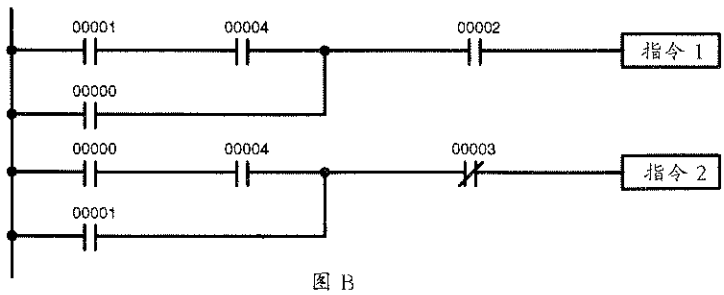


图 B

地址	指令	操作数
00000	LD	00001
00001	AND	00004
00002	OR	00000
00003	AND	00002
00004	指令 1	
00005	LD	00000
00006	AND	00004
00007	OR	00001
00008	AND NOT	00003
00009	指令 2	

指定为条件的特定位的使用次数也没有限制，根据需要，尽可能多地用这些位来简化程序。复杂的程序往往是试图减少所用位的使用次数而造成的。

除非指令本身不允许有条件(例如下面的 IL 指令，JUMP END 指令)，每条指令行上必须有至少一个条件以决定右边指令的执行条件。下面的图 A 也应改画成图 B 的形式。如果需要持续地执行一条执令(例如：程序运行时输出必须维持 ON)，那么可以使用 SR 区域的常“ON”标志。(SR25313)。

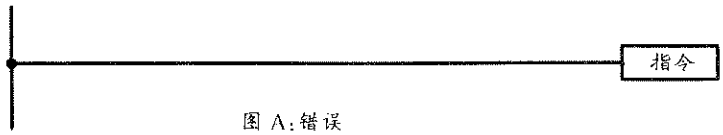


图 A：错误

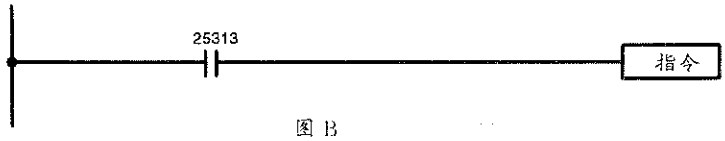


图 B

地址	指令	操作数
00000	LD	25313
00001	指令	

这一规则有几个例外，包括 ILC(联锁解除)、JME(跳转结束)和步指令。这些指令都是作为一对指令中的后一条而且受第一条指令的执行条件控制。在这些指令行上都不能有条件。详细情况请参考第五章“指令集”。

当画梯形图时，牢记所要输入的指令数很重要。下面图 A 中，要用一条 OR LD 指令将上下两指令行结合起来。而将图 A 改画成图 B 的形式，可以避免使用 AND LD 或 OR LD 指令。有关 AND LD、OR LD 详细情况可参阅 5－8－2“AND LD 和 OR LD 指令”，第七章“程序监视和执行”会有更多的例子。

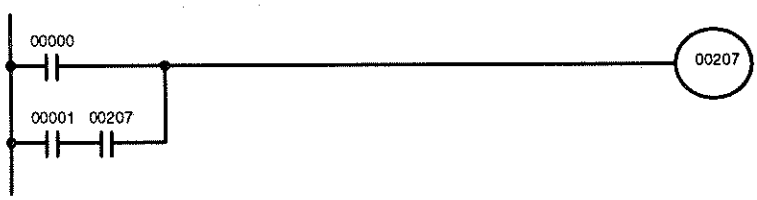


图 A

地址	指令	操作数
00000	LD	00000
00001	LD	00001
00002	AND	00207
00003	OR LD	—
00004	OUT	00207



图 B

地址	指令	操作数
00000	LD	00001
00001	AND	00207
00002	OR	00000
00003	OUT	002

4－11 程序执行

一旦程序启动执行，CPU 自上而下循环扫描，检查所有条件并运行所有到达母线的指令。将指令按适当顺序放置相当重要，例如，在指令要用到某个字之前，先把所需的数据送入该字。记住，一条指令行将完全执行到右边最后一条指令，然后再从分支处执行另一指令行。

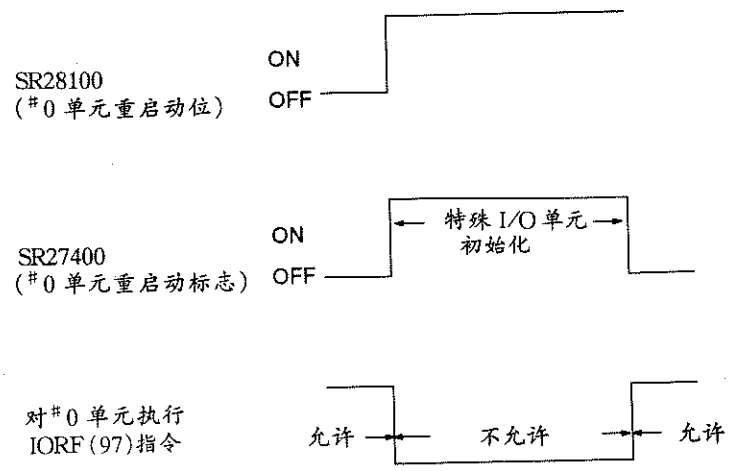
程序的执行实际上仅仅是 CPU 在扫描周期时间内所完成的任务之一，详细情况请参阅第六章“程序执行时间”。

4－12 特殊 I/O 单元接口程序

本节介绍特殊 I/O 单元的编程方法和注意事项。

4－12－1 特殊 I/O 单元重启动

特殊 I/O 单元重启动之后 IORF(97)停止执行直到特殊 I/O 单元完成初始化。



当重启动(SR27400)标志为“ON”,执行正常的 END 刷新且对特殊 I/O 单元进行初始化。上述过程处理与字 DM6620 和 DM6621 的设置无关,而这两个字的设定与特殊 I/O 单元刷新有关。在完成初始化之前,程序不会对该初始化的单元执行 IORF(97)指令。

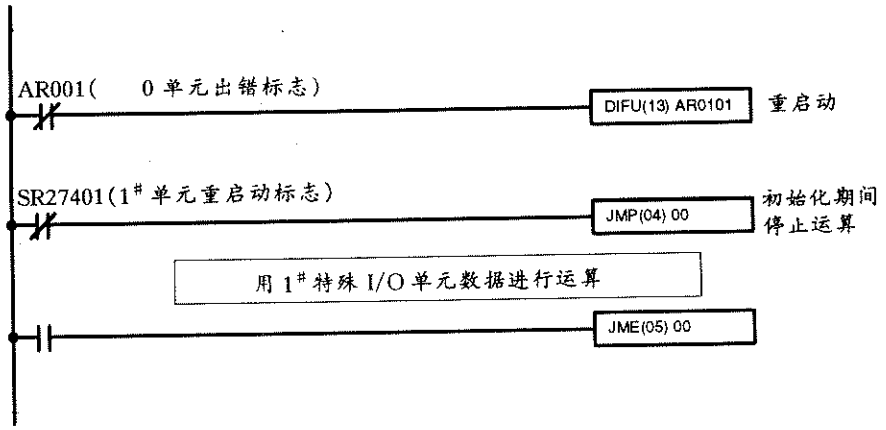
初始化过程中传送的特殊 I/O 单元数据可能会丢失。编写一个程序重启动特殊 I/O 单元要注意,当特殊 I/O 单元的重启动标志(SR27400－SR27415)为 ON 时,对于那些用来自正在初始化的特殊 I/O 单元的数据为计算数据的部分程序应予停止,而对不在初始化的单元,其程序可继续正常进行。

装在从站机架上特殊 I/O 单元的重启动标志不会变为“ON”。

标准特殊 I/O 单元的重启动时间为(20×扫描周期时间)。

4－12－2 特殊 I/O 单元出错处理程序

用下述的程序重启动出错的特殊 I/O 单元。例子中的程序重启动是单元 1。



4－12－3 改变特殊 I/O 单元设置

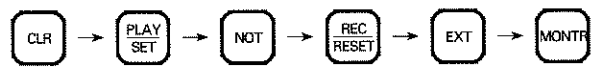
在 C200HX/HG/HE 型机中,可以用梯形指令在特殊 I/O 单元区域(DM1000－DM2599)内写入数据和修改特殊 I/O 单元的设置。对不同的生产过程需要不一样的设置时,改变设置是方便的。

在本例中,有两种生产过程,需要不同的特殊 I/O 单元设置。第一个生产过程的设置存在 DM7000－DM7999 中,而第二个存在非 DM8000－DM8999 中。

编程器操作

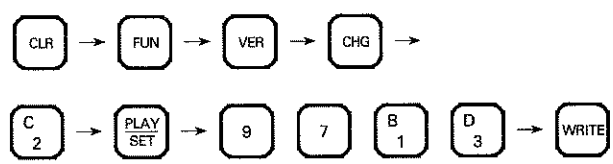
当程序使用 XFER(70)将固定 DM 区域(DM6144－DM6599)中的内容直接复盖到 DM1000－DM1999,下述步骤中第 1 至第 5 步就不需要了。在这种情况下,一旦 DM1000－DM1999 数据被复盖,程序即重启动单元。

1.清除内存(全清除)。



在存储器被清除之前,UM 区域分配操作不会执行。

2.执行 UM 区域分配操作,给扩展 DM 区域分配 2K 字(DM7000－DM7999)。

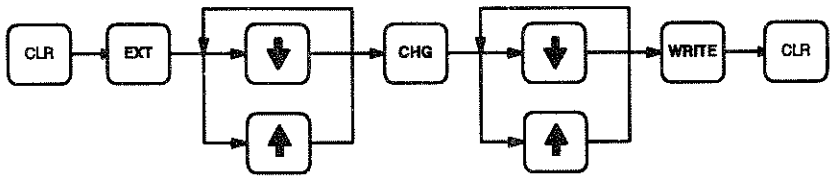


3.执行十六进制－BCD 码数据转换操作,在 DM6602 中写入＃0100,将特殊 I/O 单元设置在“C200H－

兼容 ROM 方式 1"的方式下，该方式在 PC 启动时自动传送 DM7000 - DM7999 的内容到 DM1000 - DM1999 中。PC 这个新的设置设定只有在 PC 先断电再通电后重新启动时方才有效。

4. CPU 的 DIP 开关插脚 4 设为 ON，该设置允许用户指定扩展指令功能码。

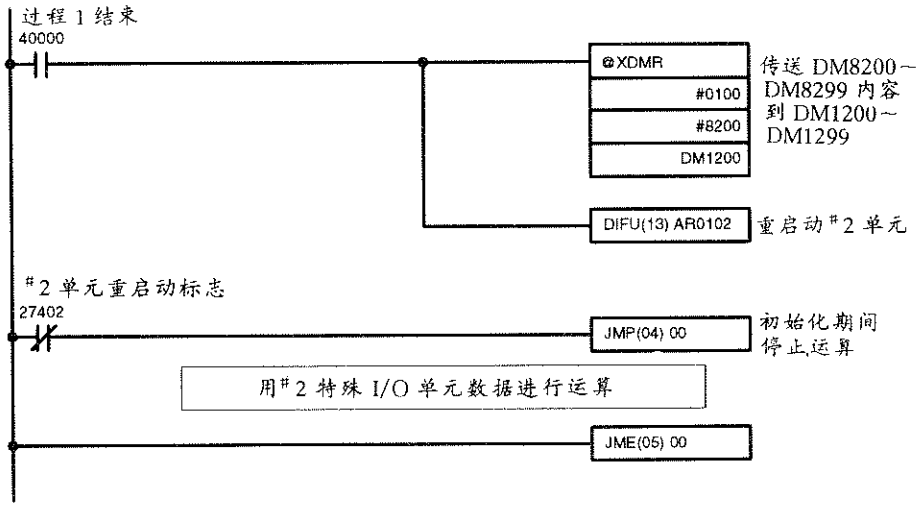
5. 执行扩展指令功能码设定操作，给 XDMR(－)指定一功能码。



6. 输入程序

程序举例

下述程序修改 2＃特殊 I/O 单元的区域设置，且重启动该单元，同时在单元初始化时暂停对 2＃单元来的数据进行计算。



4－12－4 特殊 I/O 单元的 I/O 单元刷新间隔时间

当 I/O 刷新间隔时间太短，会延缓对特殊 I/O 单元的处理，并导致特殊 I/O 单元出错或影响单元的正常操作。此时，使用下述的方法，以恢复单元的正常工作。

结束(END)刷新之间的短间隔时间

有两种方法延长 I/O 刷新之间的间隔时间，可使用其中任何一个。

1. 在 PC 设置(DM6621)中，不允许特殊 I/O 单元循环刷新，仅在需要时用 IORF(97)指令刷新特殊 I/O 单元的 I/O。

在 DM6621 中置入＃0100，可停止安装在 CPU 或扩展 I/O 机架上的所有特殊 I/O 单元的循环刷新。

2. 在 PC 设置(DM6619)中设一个最小的扫描周期时间或在程序中执行 SCAN(18)，以增加 PC 的扫描周期时间(SCAN(18)是一个缺省功能码为 18 的扩展指令)。

IORF(97)和结束(END)刷新之间的短间隔时间

用 IORF(97)刷新或结束(END)刷新指令改写程序。也可以在 PC 设置(DM6619)中设一个最小的扫描周期时间或在程序中执行 SCAN(18)，以增加 PC 的扫描周期的时间。

IORF(97)指令之间的短间隔时间

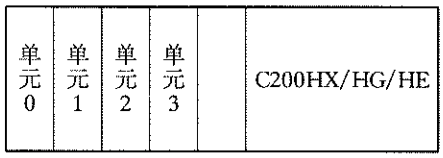
改写程序，增加 IORF(97)指令之间的间隔或者只使用一条 IORF(97)指令。

4－12－5 减少扫描周期时间

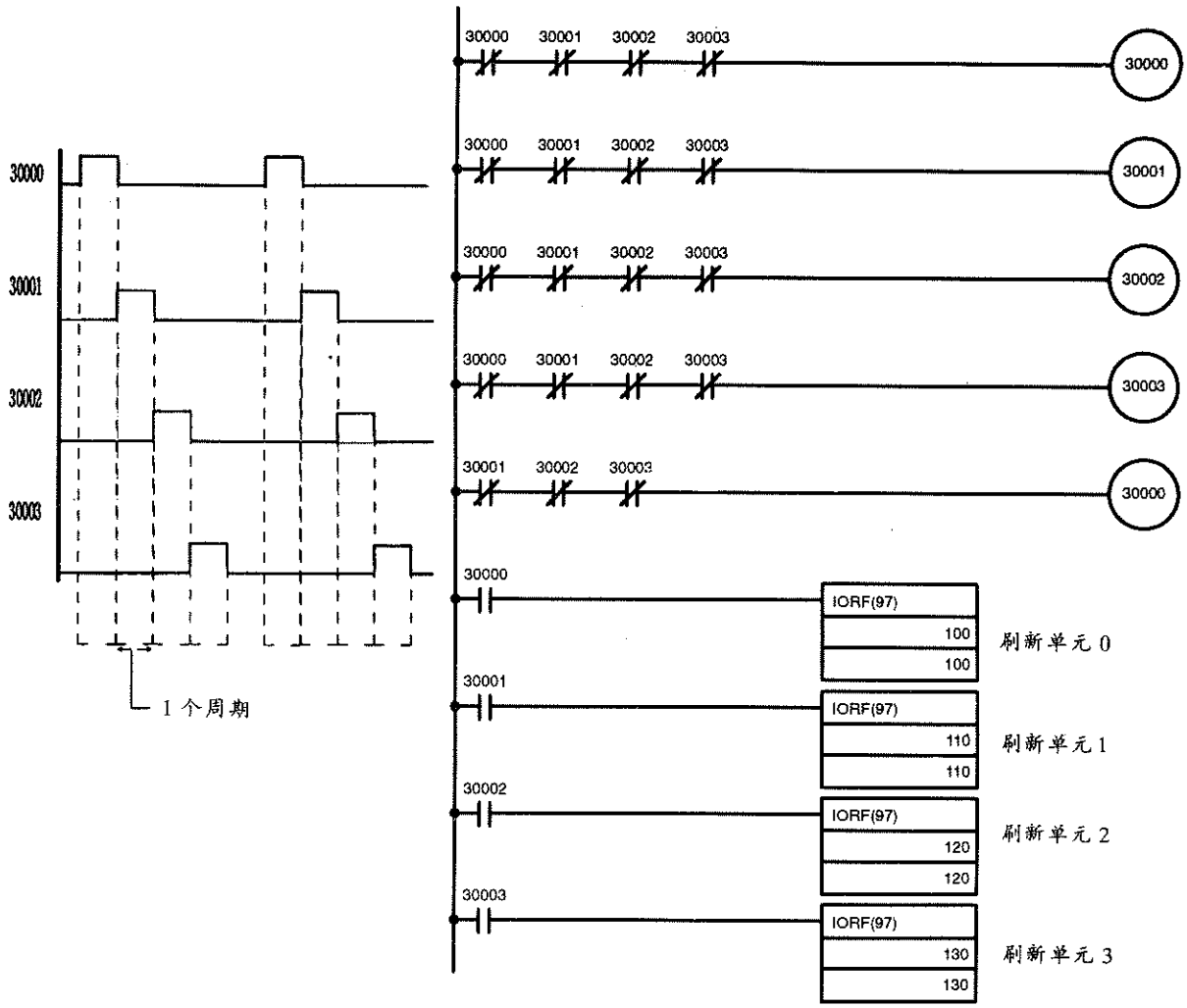
在 C200HX/HG/HE PC 机上安装了特殊 I/O 单元，每个扫描周期会自动执行结束(END)刷新，无需作任何特别的设置。当使用多个特殊 I/O 单元，因为这个自动的 I/O 刷新需要时间，可能使整个扫描周期时间变得很长。

为了减少这个 I/O 刷新时间，在 PC 的设置中停止特殊 I/O 单元循环刷新，代之以 IORF(97)指令来刷新特殊 I/O 单元。在 PC 设置中，用＃0100 置入 DM6621，可停止所有安装在 CPU 机架或扩展 I/O 机架上特殊 I/O 单元的 I/O 刷新。

在下面的例子中，一台装有 4 个特殊 I/O 单元 PC 的程序，采用每个扫描周期只刷新一个单元的方法，减少特殊 I/O 单元的刷新时间。单元按次序刷新：单元 0，单元 1，单元 2，单元 3，单元 0…



下列程序仅针对所有安装在 CPU 机架或扩展机架 I/O 上的特殊 I/O 单元，因为安装在从站机架上的特殊 I/O 单元总是执行结束(END)刷新指令，与 PC 设置无关。



注：在本程序的 OUT 指令中，IR30000 被使用了两次。这种类型的重复虽然在此处可以接受，但通常是不允许，除非有特殊原因并且保证运行正常。

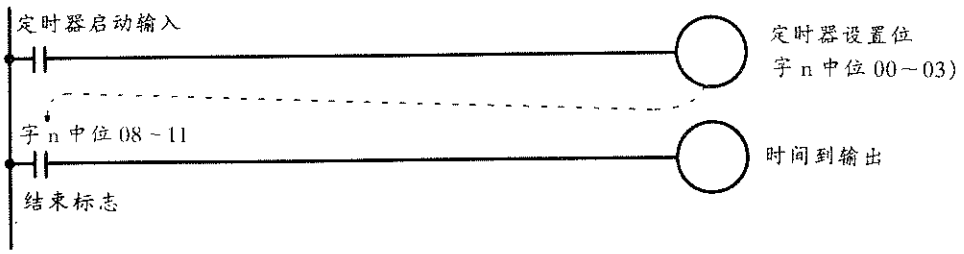
4－13 模拟定时器单元编程

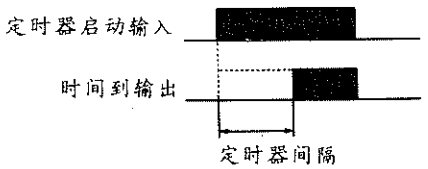
不用编程器，模拟定时器单元的设定值也可以容易地更改。由于单元设有外部可变电阻的连接器，因此在面板上可以安装一个可变电阻，与模拟定时器单元相连，用手动设定或调整设定值(SV)。

4－13－1 运行

当定时器的启动输入变 ON 时，模拟定时器单元的定时器设置位(字 n 的位 00－03)变为 ON，模拟定时器开始工作，同时单元上的定时器设置指示器(SET)亮。

当定时器设定值(内部或外部设置)时间过了，单元结束标志(字 n 的位 08－11)和时间到输出变 ON。同时在单元上的时间到指示器(TIME UP)亮。





有关定时器设定值内部和外部设置的转换，可变电阻的连接和开关的设置，详见模拟定时器操作手册。

4－13－2 位分配和 DIP 开关设置

下表示出了分配给模拟定时器单元字(n)的用途。该字的地址决定于单元安装的槽口位置。

位	I/O 类型	功能	注解
00	输入	0#定时器设置位	设置定时器时为 ON
01		1#定时器设置位	
02		2#定时器设置位	
03		3#定时器设置位	
04		0#定时器停止位	OFF：允许定时器工作 ON：停止定时器工作
05		1#定时器停止位	
06		2#定时器停止位	
07		3#定时器停止位	
08	输出	0#定时器结束位	定时器时间到时为 ON
09		1#定时器结束位	
10		2#定时器结束位	
11		3#定时器结束位	
12～15	…	未使用	…

定时器范围设置

定时器定时范围用单元面板上面一个 DIP 开关设置，每个定时器可以分别设置其范围。

定时范围	0# 定时器		1# 定时器		2# 定时器		3# 定时器	
	脚 8	脚 7	脚 6	脚 5	脚 4	脚 3	脚 2	脚 1
0.1～1 秒	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
1～10 秒	ON	OFF	ON	OFF	ON	OFF	ON	OFF
10～60 秒	OFF	ON	OFF	ON	OFF	ON	OFF	ON
1～10 分	ON	ON	ON	ON	ON	ON	ON	ON

内部/外部定时器 SV 设定的选择

内部或外部定时器 SV 设定用单元面板下面一个 DIP 开关设置。

设置	0# 定时器(脚 4)	1# 定时器(脚 3)	2# 定时器(脚 2)	3# 定时器(脚 1)
内部	ON	ON	ON	ON
外部	OFF	OFF	OFF	OFF

4－13－3 示例程序

单元配置

下表表明本例中单元分配到的字

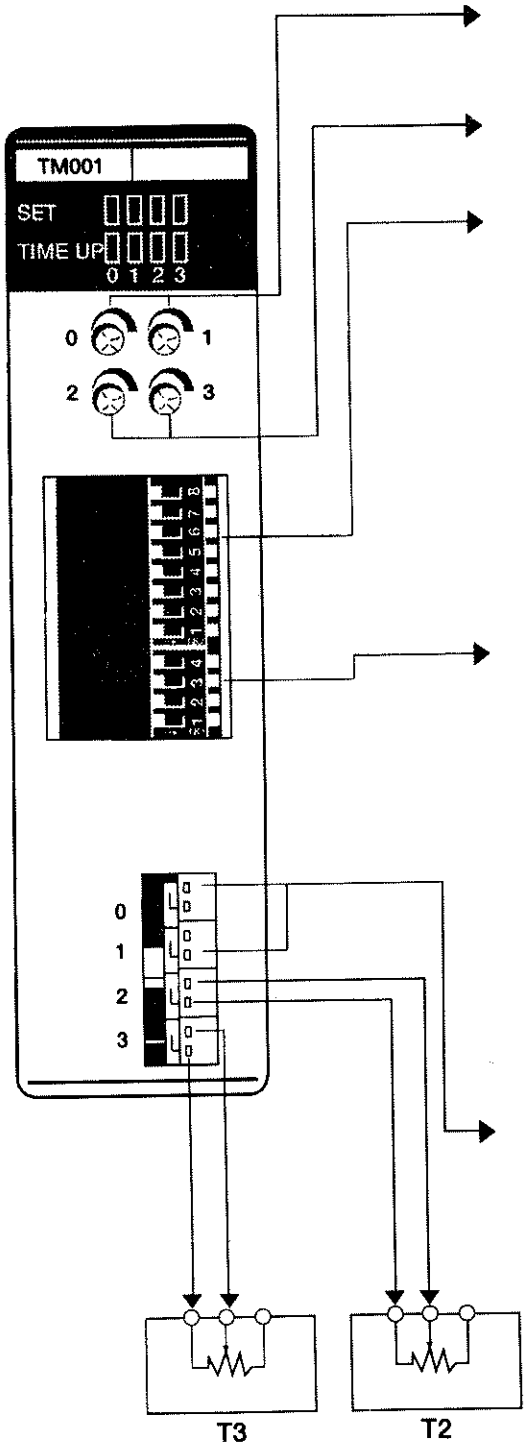
项 目	字
分配给模拟定时器的 IR 字	IR002
分配给输入单元的 IR 字	IR000
分配给输出单元的 IR 字	IR005

模拟定时器单元 SV 的设置和外部可变电阻的连接如下图所示。

定时器	设定值	范围	可变电阻器设置	可变电阻器控制
0#	0.6 秒	0.1～1 秒	60％顺时针	内部
1#	3 秒	1～10 秒	30％顺时针	内部
2#	20 秒	10～60 秒	20％顺时针	外部
3#	8 分	1～10 分	80％顺时针	外部

单元设置和连接

下图表示开关的设置和实现单元配置所需要的线路连接。



因为 0# 和 1# 定时器被设置为内部 SV 设定，这两个可变电阻器设定有效。

使用单元附带的螺丝刀调节可变电阻器。

因为 2# 和 3# 定时器被设置为外部 SV 设定，这两个可变电阻器不起作用。

定时器范围设置如下：

0# 定时器		1# 定时器		2# 定时器		3# 定时器	
0.1～1 秒		1～10 秒		10～60 秒		1～10 分	
脚 8	脚 7	脚 6	脚 5	脚 4	脚 3	脚 2	脚 1
OFF	OFF	ON	OFF	OFF	ON	ON	ON

内部/外部 SV(设定值)设置如下：

0# 定时器	1# 定时器	2# 定时器	3# 定时器
脚 4	脚 3	脚 2	脚 1
内部	内部	外部	外部
ON	ON	OFF	OFF

这两个连接器上不连接任何东西。

0# 和 1# 定时器被设置或内部 SV 设定，用单元顶端的可变电阻器设定它们的 SV(设定值)。

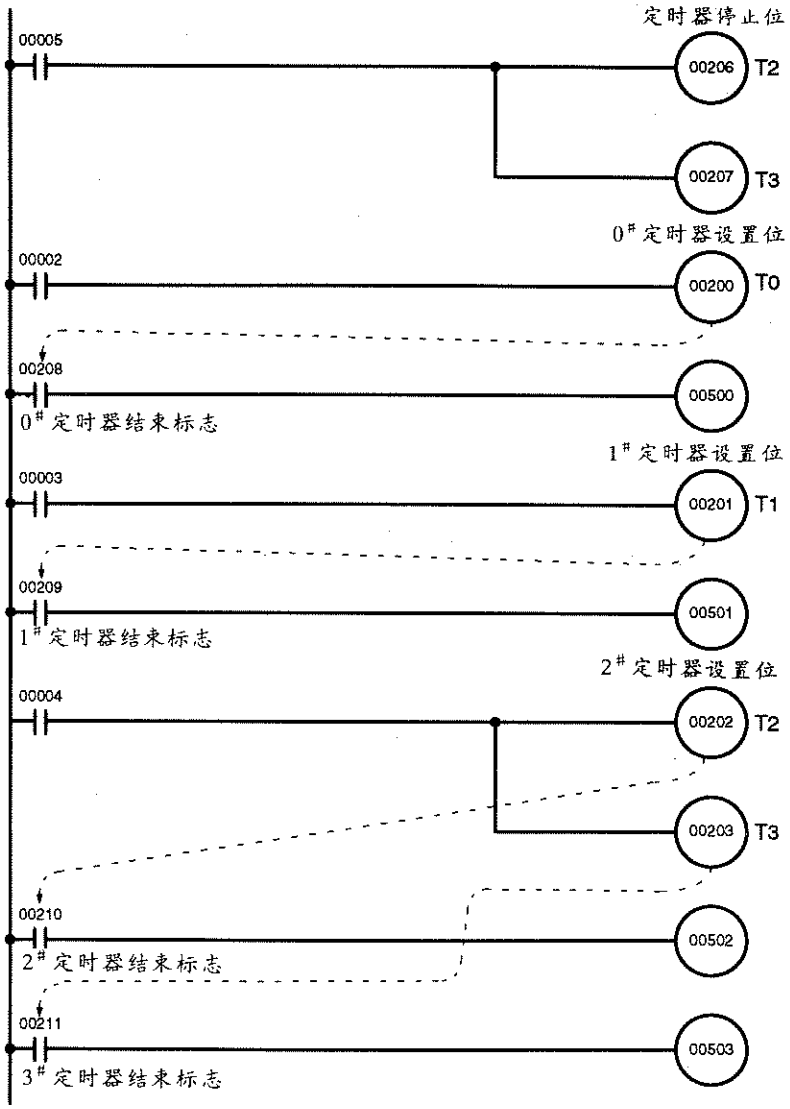
外部 SV 设定(0～20kΩ)

这两个连接器上连接 2# 和 3# 定时器的可变电阻器。有关设置的细节参阅模拟定时器单元操作手册。

梯形图程序

下面为本例的梯形图程序

1. 输入 IR00002 变 ON 后约 0.6 秒，输出 IR00500(T0)变 ON。
2. 输入 IR00003 变 ON 后约 3 秒，输出 IR00501(T1)变 ON。
3. 输入 IR00004 变 ON 后约 20 秒，输出 IR00502(T2)变 ON。
   输入 IR00004 为 ON 后约 8 分钟，输出 IR00503(T3)变 ON。
4. 输入 IR00005，停止定时器 2 和定时器 3。



当急停输入变 ON，2# 和 3# 定时器停止工作。

当 IR00002 变 ON 时，0# 定时器(00200)开始工作，单元 SET 指示器亮。

当内部可变电阻上设定的时间过了，结束标志(00208)变 ON 单元时间到指示器亮，同时输出(IR00500)接通。

当 IR00003 变 ON 时，1# 定时器(00201)开始工作，单元 SET 指示器亮。

当内部可变电阻器上设定的时间过了，结束标志(00209)变 ON，单元到时间指示器亮，同时输出(IR00501)接通。

当 IR00004 变 ON 时，2# 定时器(00202)和 3# 定时器(00203)开始工作，单元 SET 指示器亮。

当外部可变电阻器上设定时间过了，结束标志(00210)变 ON，单元到时间指示器亮，同时输出(IR00502)接通。

当外部可变电阻器上设定时间过了，结束标志(00211)变 ON，单元到时间指示器亮，同时输出(IR00503)接通。

第五章　指令集

C200HX/HG/HE PC有大量的用于复杂控制处理的简便编程指令集。这一篇节分别说明各指令和给出梯形图符号、数据区及使用的各个标志。

C200HX/HG/HE可以处理100多条需要功能码的指令，但是系统只有100个功能码(00～99)可用。一些称作扩展指令的指令，它们没有固定的功能码。这些指令在使用前必须用作为扩展指令的18个功能码进行定义。

C200HX/HG/HE所提供的许多指令被分成下列分章节中的指令组。这些组包括梯形图指令，位控制指令，定时器和计数器指令，数据移位指令，数据传送指令，数据比较指令，数据转换指令，BCD码运算指令，二进制运算指令，逻辑指令，子程序指令，特殊指令，网络指令，串行通信指令，高级I/O指令，和特殊I/O单元指令。

有些指令如定时器和计数器指令可用于控制其它指令的执行，例如，当定时器设定的周期时间结束时，定时器的结束标志可用来接通一个位。虽然这些指令常通过输出指令来控制输出位，但它们也可用于控制其它指令的执行。因此，本手册中所例举的输出指令通常可用为特殊应用而优化程序的其它指令来取代，并不直接控制输出位。

5－1　注意事项…… 130
5－2　指令格式…… 130
5－3　数据区，定义符值和标志 …… 130
5－4　微分型指令…… 131
5－5　扩展指令…… 132
5－6　右侧指令规则…… 134
5－7　指令集表…… 136
　5－7－1　功能码 …… 136
　5－7－2　助记符字母表 …… 138
5－8　梯形图指令…… 142
　5－8－1　取(LOAD)，取非(LOAD NOT)，与(AND)，与非(AND NOT)，或(OR)，或非(OR NOT)指令 …… 142
　5－8－2　逻辑块与(AND LOAD)和逻辑块或(OR LOAD)指令 …… 142
5－9　位控制指令…… 143
　5－9－1　输出和输出非指令——OUT和OUT NOT …… 143
　5－9－2　上升沿微分和下降沿微分指令——DIFU(13)和DIFD(14) …… 144
　5－9－3　置位和复位指令——SET和RSET …… 145
　5－9－4　保持指令——KEEP(11) …… 146
5－10　联锁和联锁解除指令——IL(02)和ILC(03) …… 148
5－11　跳转和跳转终了指令——JMP(04)和JME(05) …… 150
5－12　结束指令——END(01) …… 151
5－13　空操作指令——NOP(00) …… 151
5－14　定时器和计数器指令 …… 152
　5－14－1　定时器指令——TIM …… 152
　5－14－2　高速定时器指令——TIMH(15) …… 156
　5－14－3　累加定时器指令——TTIM(87) …… 157

5－14－4　计数器指令——CNT …… 158
5－14－5　可逆计数器指令——CNTR(12) …… 161
5－15　数据移位指令 …… 162
5－15－1　移位寄存器指令——SFT(10) …… 162
5－15－2　可逆移位寄存器——SFTR(84) …… 165
5－15－3　算术左移指令——ASL(25) …… 166
5－15－4　算术右移指令——ASR(26) …… 167
5－15－5　循环左移指令——ROL(27) …… 167
5－15－6　循环右移指令——ROR(28) …… 168
5－15－7　数字(4 bit)左移——SLD(74) …… 168
5－15－8　数字(4 bit)右移——SRD(75) …… 169
5－15－9　字移位指令——WSFT(16) …… 169
5－15－10　异步移位寄存器——ASFT(17) …… 170
5－16　数据传送指令 …… 171
5－16－1　传送指令——MOV(21) …… 171
5－16－2　取反传送指令——MVN(22) …… 172
5－16－3　块设定指令——BSET(71) …… 172
5－16－4　块传送指令——XFER(70) …… 174
5－16－5　数据交换指令——XCHG(73) …… 174
5－16－6　单字数据分配指令——DIST(80) …… 175
6－16－7　数据抽出指令——COLL(81) …… 176
5－16－8　位(Bit)传送指令——MOVB(82) …… 179
5－16－9　数(4－bit)传送指令——MOVD(83) …… 179
5－16－10　多位传送指令——XFRB(62) …… 180
5－16－11　EM 块传送指令——XFR2(－) …… 182
5－16－12　EM 组传送指令——BXF2(－) …… 182
5－17　数据比较指令 …… 183
5－17－1　多字比较指令——MCMP(19) …… 183
5－17－2　单字比较指令——CMP(20) …… 185
5－17－3　双字比较指令——CMPL(60) …… 187
5－17－4　块比较指令——BCMP(68) …… 189
5－17－5　表格比较指令——TCMP(85) …… 190
5－17－6　区域比较指令——ZCP(88) …… 191
5－17－7　双字区域比较指令——ZCPL(－) …… 193
5－17－8　带符号二进制比较指令——CPS(－) …… 193
5－17－9　带符号双字二进制比较指令——CPSL(－) …… 194
5－18　数据转换指令 …… 195
5－18－1　BCD 码→二进制指令——BIN(23) …… 195
5－18－2　双字 BCD 码→双字二进制指令——BINL(58) …… 196
5－18－3　二进制→BCD 码指令——BCD(24) …… 196
5－18－4　双字二进制→双字 BCD 码指令——BCDL(59) …… 197
5－18－5　小时→秒指令——SEC(65) …… 198

5－18－6　秒→小时指令——HMS(66) …… 199
5－18－7　4→16/8→256 译码指令——MLPX(76) …… 199
5－18－8　16→4/256→8 编码指令——DMPX(77) …… 202
5－18－9　七段译码指令——SDEC(78)…… 205
5－18－10　ASCII 码转换指令——ASC(86) …… 207
5－18－11　ASCII 码→十六进制指令——HEX(－)…… 209
5－18－12　线性比例指令——SCL(－) …… 211
5－18－13　列→行转换指令——LINE(63) …… 213
5－18－14　行→列转换指令——COLM(64)…… 214
5－18－15　二进制求补指令——NEG(－) …… 215
5－18－16　双字二进制求补指令——NEGL(－) …… 216
5－19　BCD 码运算指令 …… 217
5－19－1　增量指令——INC(38) …… 217
5－19－2　减量指令——DEC(39) …… 218
5－19－3　置进位标志指令——STC(40) …… 218
5－19－4　清进位标志指令——CLC(41) …… 218
5－19－5　BCD 码加法指令——ADD(30) …… 218
5－19－6　双字 BCD 码加法指令——ADDL(54) …… 219
5－19－7　BCD 码减法指令——SUB(31) …… 221
5－19－8　双字 BCD 码减法指令——SUBL(55) …… 223
5－19－9　BCD 码乘法指令——MUL(32)…… 225
5－19－10　双字 BCD 码乘法指令——MULL(56) …… 226
5－19－11　BCD 码除法指令——DIV(33)…… 226
5－19－12　双字 BCD 码除法指令——DIVL(57)…… 227
5－19－13　浮点数除法指令——FDIV(79) …… 228
5－19－14　平方根指令——ROOT(72) …… 231
5－20　二进制运算指令 …… 233
5－20－1　二进制加法指令——ADB(50) …… 233
5－20－2　二进制减法指令——SBB(51) …… 235
5－20－3　二进制乘法指令——MLB(52) …… 237
5－20－4　二进制除法指令——DVB(53) …… 238
5－20－5　双字二进制加法指令——ADBL(－) …… 239
5－20－6　双字二进制减法指令——SBBL(－)…… 241
5－20－7　带符号二进制乘法指令——MBS(－) …… 243
5－20－8　带符号双字二进制乘法指令——MBSL(－) …… 244
5－20－9　带符号二进制除法指令——DBS(－) …… 245
5－20－10　带符号双字二进制除法指令——DBSL(－) …… 246
5－21　特殊算术指令 …… 247
5－21－1　寻找最大值指令——MAX(－) …… 247
5－21－2　寻找最小值指令——MIN(－) …… 248
5－21－3　求平均值指令——AVG(－) …… 249
5－21－4　求和指令——SUM(－) …… 251

5－21－5　数学处理指令——APR(69) …… 253
5－21－6　PID 控制指令——PID(－) …… 256
5－22　逻辑指令 …… 264
5－22－1　求反指令——COM(29) …… 264
5－22－2　“逻辑与”指令——ANDW(34) …… 265
5－22－3　“逻辑或”指令——ORW(35) …… 266
5－22－4　“异或“的逻辑指令——XORW(36) …… 267
5－22－5　“异或非”的逻辑指令——XNRW(37) …… 267
5－23　子程序和中断控制指令 …… 268
5－23－1　子程序指令 …… 268
5－23－2　中断指令 …… 268
5－23－3　子程序调用指令——SBS(91) …… 271
5－23－4　子程序定义和返回指令——SBN(92)/RET(93) …… 273
5－23－5　宏指令——MCRO(99) …… 273
5－23－6　中断控制指令——INT(89) …… 275
5－24　步进指令 …… 278
5－24－1　步的定义和起动指令——STEP(08)/SNXT(09) …… 278
5－25　特殊指令 …… 286
5－25－1　出错报警指令和严重故障报警指令 FAL(06)/FALS(07) …… 286
5－25－2　循环时间指令——SCAN(18) …… 287
5－25－3　跟踪存贮器采样指令——TRSM(45) …… 288
5－25－4　信息显示指令——MSG(46) …… 289
5－25－5　长信息指令——LMSG(47) …… 290
5－25－6　终端方式指令——TERM(48) …… 291
5－25－7　监视定时器刷新指令——WDT(94) …… 292
5－25－8　I/O 刷新指令——IORF(97) …… 292
5－25－9　组－2 高密度 I/O 刷新指令——MPRF(61) …… 293
5－25－10　位(bit)计数器指令——BCNT(67) …… 293
5－25－11　帧校验指令——FCS(－) …… 294
5－25－12　故障点检测指令——FPD(－) …… 296
5－25－13　数据搜索指令——SRCH(－) …… 299
5－25－14　扩展 DM 读指令——XDMR(－) …… 300
5－25－15　EM 间接寻址指令——IEMS(－) …… 301
5－25－16　选择 EM 组指令——EMBC(－) …… 302
5－26　网络指令 …… 303
5－26－1　网络发送——SEND(90) …… 303
5－26－2　网络接收——RECV(98) …… 305
5－26－3　关于网络通信 …… 307
5－27　串行通信指令 …… 310
5－27－1　接收指令——RXD(－) …… 310
5－27－2　发送指令——TXD(－) …… 311
5－27－3　修改变 RS－232C 设置指令——STUP(－) …… 314

5－27－4　通信协议宏指令——PMCR(－) …… 315
5－28　高级 I/O 指令 …… 316
5－28－1　七段码显示输出指令——7SEG(－) …… 317
5－28－2　数字开关输入指令——DSW(－) …… 319
5－28－3　十六进制键输入指令——HKY(－) …… 323
5－28－4　十进制键输入指令——TKY(－) …… 325
5－28－5　矩阵输入指令——MTR(－) …… 327
5－29　特殊 I/O 单元指令 …… 329
5－29－1　特殊 I/O 单元读指令——IORD(－) …… 330
5－29－2　特殊 I/O 单元写指令——IOWR(－) …… 331

5－1 注意事项

在本操作手册的下面部分中，所有指令都将用其助记符表示。例如，输出指令用 OUT；逻辑块写指令用 AND LD。如果你对使用的助记符指令不太清楚，可参见附录 B 编程指令。

如果一条指令用功能码指定，功能码在助记符后用括号( )标明。功能码是两位十进制数，这些功能码是将大部分指令送到 CPU。功能码的含意在下面作概要说明，具体说明在 4－7 章节“程序的输入、修改和检查”中，附录 B 以功能码顺序方式给出指令表。

助记符前冠以@符号表示那条指令是微分型指令。微分型指令在 5－4 章节中加以说明。

5－2 指令格式

大多数指令至少有一个或多个与它相关的操作数。操作数指定或提供指令执行的数据。这些操作数的输入有时是实际的数值(例如常数)，但一般是含有所用数据的数据区字或位的地址。其地址指定为操作数的位称为操作数位。其地址指定为操作数的字称为操作数字。在一些指令中，一条指令的字地址表示含有所需数据的初始地址。

每条指令在程序存贮器中需要一个或多个字。指令的第一个字是定义指令和包含所有定义符(下面叙述)或者是指令所需操作数位的指令字。指令所需其它操作数包含在后续字中，每个字一个操作数，有些指令需要多达四个字。

定义符是与指令相关的操作数，它包括在指令字中。它些操作数只定义指令并不指明取用什么数据。定义符的例子是 TC 编号，这些编号用于生成定时器和计数器的定时器和计数器指令中，同样，象跳转号，它用于定义跳转和跳转结束指令对。尽管位操作数未被看作定义符，但它也包含在指令本身字中。

5－3 数据区，定义符值和标志

本节中，每条指令的描述包括了指令的梯形图符号，指令操作数可使用的数据区域和定义符数值。详细的数据区由操作数名称和每一操作数所要求的数据类型规定(例如字或位，对字而言是十六进制还是二—十进制)。

在指定的数据区内，并非所有的地址都可允许用于操作数，例如，有一个操作数需要两个字，但由于用于一个单独操作数的所有字必须在同一数据区内，所以数据区的最后一个字不能被定义为操作数的起始字。同样，并非 SR 和 DM 区的所有字都可写作操作数(详见第三章存贮器区域)。指令的其它特别限制在限制说明部分给出。参见第三章存贮区寻址约定，标志地址和控制位。

注意

IR 和 SR 区可看作不相连的二个数据区。如果一个操作数访问一个数据区，它并不意味这同一操作数将会访问另一个数据区。但是一个单独的操作数可以穿越 IR 和 SR 区域之间分界线，即，只要 SR 区域也允许被这操作数使用，一个需要多字的操作数可定义在 IR 区域的最后位。

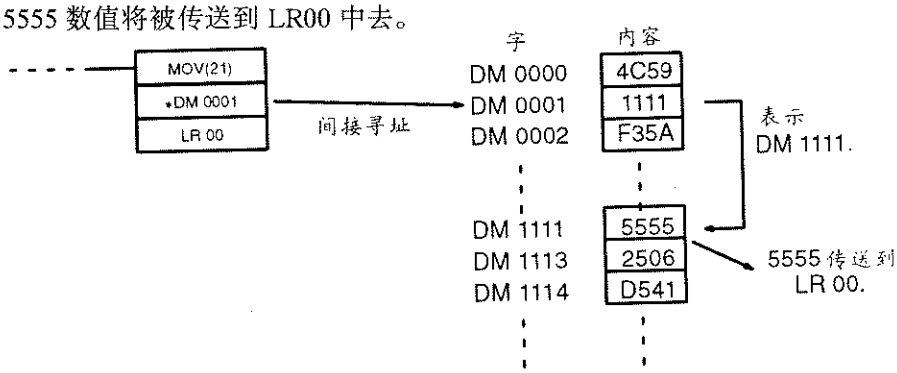
标志段部分列出了受指令的执行影响的标志，这些标志包括下列 SR 区域的标志位

缩　写	名　称	位
ER	指令执行出错标志	25503
CY	进位标志	25504
GR	大于标志	25505
EQ	等于标志	25506
LE	小于标志	25507
N	负标志	25402
OF	上溢标志	25404
UF	下溢标志	25405

ER 是用于监视指令执行最常用的标志。当 ER 变为 ON，这表示正要执行的当前指令出错。在每条指令的标志段落中列出了 ER 变为 ON 可能的原因。如果操作数输入不正确，ER 位变 ON，当 ER 为 ON，指令不执行。指令和受指令影响的标志位列表在附录 C 出错和算术标志操作中给出。

间接寻址

当一个操作数定义在 DM 区，可采用间接寻址方式。定义 DM 间接寻址只要在 DM 前加一星号（＊DM）。当定义间接 DM 地址时，指定的 DM 字中含有用于指令操作数的数据的 DM 地址。例如，＊DM0001 被定义为 MOV(21)的第一个操作数，LR00 作为 MOV(21)指令的第二个操作数，DM0001 的内容是 1111，DM1111 的内容是 5555，那么 5555 数值将被传送到 LR00 中去。



当采用间接寻址时，所需字的地址必须是 BCD 码，它所定义的字必须在 DM 区域内。上例中，＊DM0000 的内容必须在 0000～6655BCD 码范围内。

可使用 IEMS(－)指令将＊DM 的目标从 DM 区移至 EM 区中一个组。详见 5－25－15“EM 区间接寻址－IEMS(－)”。

定义常数

尽管数据区地址常以操作数形式给出，但许多操作数和所有定义符以常数形式输入。任何一个定义符或操作数的适用值范围取决于各使用它的指令。常数输入也必须是指令所要求的形式，即以 BCD 码或十六进制形式。

5－4　微分型指令

大多数指令都有非微分型和微分型两种形式。微分型指令是在指令助记符前加@标记。

只要执行条件为 ON，非微分型指令在每个循环周期都将执行。而微分型指令仅在执行条件由 OFF 变为 ON 后方执行一次。如果执行条件不发生变化，或者从上一个指令循环周期的 ON 变为 OFF，微分型指令是不执行的。下面两个例子说明了 MOV(21)和@MOV(21)指令是如何工作的。两条指令都是将第一个操作数指定的地址中的数据传送到第二个操作数据指定的地址中去。

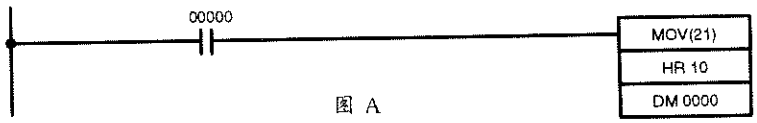


图 A

地址	指令	操作数
0000	LD	00000
00001	MOV(2)	
		HR 10
		DM 0000

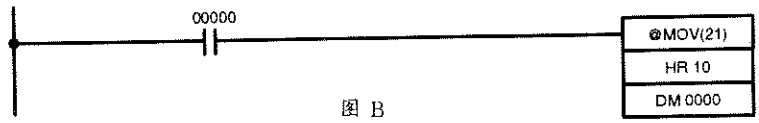


图 B

地址	指令	操作数
0000	LD	00000
0001	@MOV(21)	
		HR 10
		DM 0000

图 A 中，非微分型指令 MOV(21)根据每次循环 00000 条件将 HR10 的内容传送到 DM0000 中去。如果程序循环周期 80ms，00000 条件在 2.0s 内保持 ON 状态，那么数据传送工作将做 25 次，而仅最后一次送到 DM0000 的数据保存在那里。

图 B 中，微分型指令@MOV(21)仅仅在 00000 状态变为 ON 后将 HR10 中的内容传送一次到 DM0000，尽管程序的循环周期也是 80ms，00000 保持 ON 状态也是 2s，在 00000 由 OFF 变为 ON 状态的第一个循环周期中，数据传送操作只执行一次。由于在 00000 保持为 ON 状态的 2s 时间内，HR10 的内容可能变化很大，2s 以后 DM0000 的最终内容会由于采用 MOV(21)或@MOV(21)数据传送指令而不相同。

所有操作数，梯形图符号和指令的其它说明都与是采用微分型还是非微分型指令无关。指令输入时，两者功能码是一样的，区别在于微分型指令在功能码输入后输入 NOT。指令中，极大多数指令具有微分形式。

有关联锁对微分型指令的影响，参见 5－10 章节“联锁和联锁清除指令—IL(02)和 IL(103)”。C200HX/HG/HE 还提供了微分指令：DIFU(13)和 DIFD(14)。DIFU(13)指令操作和微分型指令一样，但此指令用于一位(bit)接通一个循环周期，DIFD(14)也是接通一位一个循环周期，不过它是在执行条件由 ON 变为 OFF 时执行，详见 5－9－2 章节“上升微分和下降微分指令——DIFU(13)和 DIFD(14)”。

注

微分型指令中不可使用 SR25313 和 SR25315，这些位不会改变状态，也不会触发微分型指令。

5－5 扩展指令

C200HX/HG/HE 有许多指令，需要 121 个功能码，这超过系统提供的 100 个功能码，所以有些指令就没有固定的功能码。这些指令被称之为扩展指令，(见下表)，包含具有缺省功能码的扩展指令。

采用编程器上的扩展功能码设定操作方法可设定这 18 个功能码中的某一个为某扩展指令的功能码。这 18 个功能码是：17，18，19，47，48，60～69，87，88 和 89。有关功能码设定，参见 7－2－14 章节“扩展指令功能码分配”。

码	助记符	名称	页
17	(@)ASFT	异步移位存器指令	170
18	(@)SCAN	循环时间指令	287
19	(@)MCMP	多字比较指令	283
47	(@)LMSG	长信息指令	290
48	(@)TERM	终端方式指令	291
60	CMPL	双字比较指令	187
61	(@)MPRF	组－2 高密度 I/O 刷新指令	293

（续表）

码	助记符	名称	页
62	(@)XFRB	多位传送指令	180
63	(@)LINE	列→行转换指令	213
64	(@)COLM	行→列转换指令	214
65	(@)SEC	小时→秒指令	198
66	(@)HMS	秒→小时指令	199
67	(@)BCNT	位计数器指令	293
68	(@)BCMP	块比较指令	189
69	(@)APR	数学处理指令	253
87	TTIM	累加定时器指令	157
88	ZCP	区域比较指令	191
89	(@)INT	中断控制指令	275
…	7SEG	七段码显示输出指令	317
…	(@)ADBL	双字二进制加法指令	239
…	AVG	求平均值指令	249
…	@BXF2	EM 组传送指令	182
…	CPS	带符号二进制比较指令	193
…	CPSL	带符号双字二进制比较指令	194
…	(@)DBS	带符号二进制除法指令	245
…	(@)DBSL	带符号双字二进制除法指令	246
…	DSW	数字开关输入指令	319
…	(@)EMBC	选择 EM 组指令	302
…	(@)FCS	帧校验指令	294
…	FPD	故障点检测指令	296
…	(@)HEX	ASCII 故障→十六进制指令	209
…	HKY	十六进制键输入指令	323
…	(@)IEMS	EM 间接寻址指令	301
…	(@)IORD	特殊 I/O 单元读指令	330
…	(@)IOWR	特殊 I/O 单元写指令	331
…	(@)MAX	寻找最大值指令	247
…	(@)MBS	带符号二进制乘法指令	243
…	(@)MBSL	带符号双字二进制乘法指令	244
…	(@)MIN	寻找最小值指令	248
…	MTR	矩阵输入指令	327
…	(@)NEG	二进制求补指令	215

（续表）

码	助记符	名称	页
…	(@)NEGL	双字二进制求补指令	216
…	PID	PID 控制指令	256
…	(@)PMCR	通信协议宏指令	315
…	(@)RXD	接收指令	310
…	(@)SBBL	双字二进制减法指令	241
…	SCL	线性比例指令	211
…	(@)SRCH	数据搜索指令	299
…	(@)STUP	修改 RS－232C 设置指令	314
…	(@)SUM	求和指令	251
…	(@)TKY	十进制键输入指令	325
…	(@)TXD	发送指令	311
…	(@)XDMR	扩展 DM 读指令	300
…	(@)XFR2	EM 块传送指令	182
…	ZCPL	双字区域比较指令	193

5－6 右侧指令规则

第四章编写和输入程序描述了表示梯形图指令的助记符书写准则。如下所述，表示所有其它指令的梯形图符号的信息转换格式是一样的，故不对每一指令一一说明。

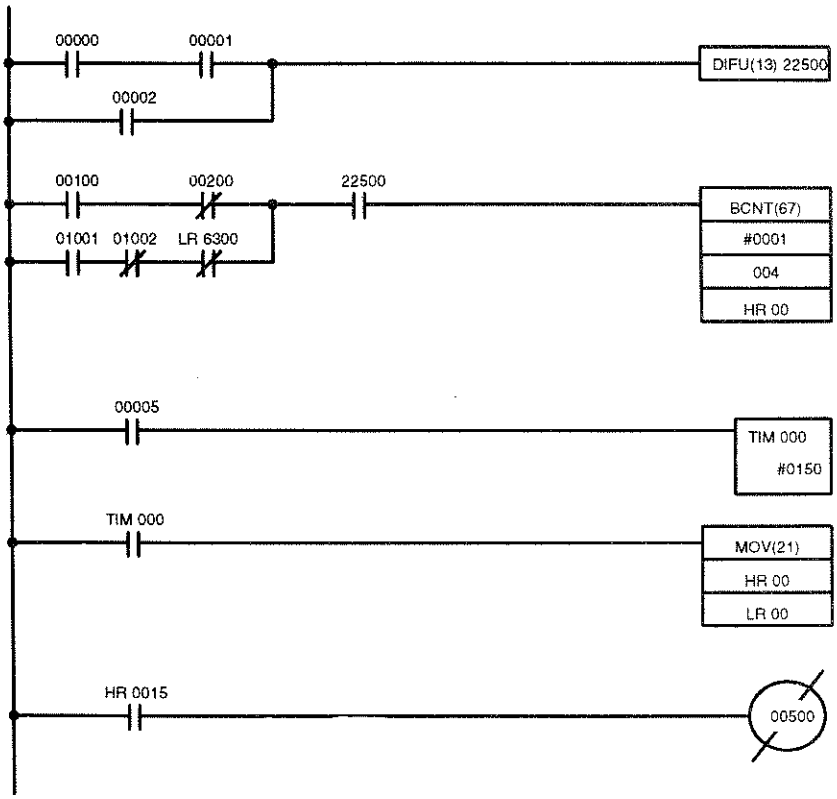
所有指令的第一个字是定义该指令和提供所有定义符。如果该指令仅不需不带定义符的一个单独位操作数，那么这位操作数就象助记符一样也放在梯形图同一行上，而所有其它操作数则放在指令行下面的行上，每行一个操作数，放置顺序如指令梯形符所示。

助记符编码表中直址和指令列只填入指令字，对于所有其它行，左面两列处于空栏。如果指令没有定义符或只有位操作数，指令第一行的数据栏处于空栏。对于所有不需要数据的指令字可容易地从空白的数据列观察到是否有地址遗漏。

如果在数据栏里放 IR 或 SR 地址，那么该栏左侧部分留下空栏。若使用其它数据区，数据区的缩写名称放在该栏左侧。指令字中作为定义符的任何号码在右栏中不需要号码符，对 TC 位，一旦定义为定时器或计数器，将 TIM 或 CNT 作为前缀。

当对一条有功能码的指令编码时，务必正确地写入功能码，此功能码可由编程器输入。请不要忘记用@符号定义微分型指令。

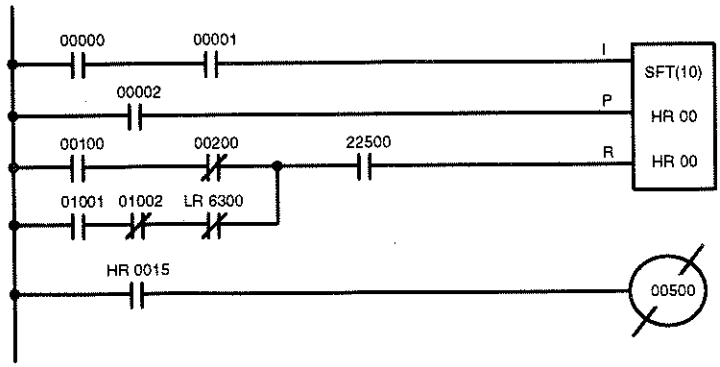
下面的图和相应的助记符说明了上述几点。



地址	指令		操作数
00000	LD		00000
00001	AND		00001
00002	OR		00002
00003	DIFU(13)		22500
00004	LD		00100
00005	AND NOT		00200
00006	LD		01001
00007	AND NOT		01002
00008	AND NOT	LR	6300
00009	OR LD		–
00010	AND		22500
00011	BCNT(67)		–
		#	0001
			004
		HR	00
00012	LD		00005
00013	TIM		000
		#	0150
00014	LD	TIM	000
00015	MOV(21)		–
		HR	00
		LR	00
00016	LD	HR	0015
00017	OUT NOT		00500

多指令行

如果右侧指令需要多指令行(如 KEEP(11)指令),所有用于该指令的行在右侧指令前输入。用于指令的各行用 LD 或 LD NOT 开始编码,与右侧指令结合形成"逻辑块",下例用 SFT(10)指令加以说明。



地址	指令	操作数	
00000	LD		00000
00001	AND		00001
00002	LD		00002
00003	LD		00100
00004	AND NOT		00200
00005	LD		01001
00006	AND NOT		01002
00007	AND NOT	LR	6300
00008	OR LD		－
00009	AND		22500
00010	SFT(10)		－
		HR	00
		HR	00
00011	LD	HR	0015
00012	OUT NOT		00500

END(01) 当你结束编程时,请别忘记在最后地址上加入 END(01)指令。

5－7 指令集表

这一节给出了 C200HX/HG/HE 的指令表。第一张表由功能码查找指令。第二张由助记符查找指令。两张表中,@均表示具有微分形式的指令。

注

扩展功能指令参见 5－5 章节"扩展指令"。

5－7－1 功能码

下列列出的指令均有固定的功能码。每条指令均以助记符和指令名形式列出。表中最左栏和项栏的数字分别作为功能码的左右位。

左数字	右数字									
	0	1	2	3	4	5	6	7	8	9
0	NOP 空操作指令	END 结束指令	IL 联锁指令	ILC 联锁解除指令	JMP 跳转指令	JME 跳转终了指令	(@)FAL 出错报警指令	FALS 严重故障报警指令	STEP 步进定义指令	SNXT 步启动指令
1	SFT 移位寄存器指令	KEEP 保持指令	CNTR 可逆计数器指令	DIFU 上升沿微分指令	DIFD 下降沿微分指令	TIMH 高速定时器指令	(@)WSFT 字移位指令	(@)ASFT 异步移位寄存器指令	(@)SCAN 循环时间指令	(@)MCMP 多字比较指令
2	CMP 单字比较指令	(@)MOV 传送指令	(@)MVN 取反传送指令	(@)BIN BCD码→二进制指令	(@)BCD 二进制→BCD码指令	(@)ASL 算术左移指令	(@)ASR 算术右移指令	(@)ROL 循环左移指令	(@)ROR 循环右移指令	(@)COM 求反指令
3	(@)ADD BCD码加法指令	(@)SUB BCD码减法指令	(@)MUL BCD码乘法指令	(@)DIV BCD码除法指令	(@)ANDW 逻辑与指令	(@)ORW 逻辑或指令	(@)XORW 异或指令	(@)XNRW 异或非指令	(@)INC 增量指令	(@)DEC 减量指令
4	(@)STC 置进位标志指令	(@)CLC 清进位标志指令				TRSM 跟踪存贮器采样指令	(@)MSG 信息显示指令	(@)LMSG 长信息指令	(@)TERM 终端方式指令	
5	(@)ADB 二进制加法指令	(@)SBB 二进制减法指令	(@)MLB 二进制乘法指令	(@)DVB 二进制除法指令	(@ADDL 双字BCD码加法指令	(@)SUBL 双字BCD码减法指令	(@)MULL 双字BCD码乘法指令	(@)DNL 双字BCD码除法指令	(@)BINL 双字BCD码→二进制指令	(@)BCDL 双字二进制→BCD码指令
6	CMPL 双字比较指令	(@)MPRF 组一2高密度I/O刷新指令	(@)XFRB 多位传送指令	(@)LINE 列→行转换指令	(@)COLM 行一列转换指令	(@)SEC 小时→秒指令	(@)HMS 秒→小时指令	(@)BCNT 位计数器指令	(@)BCMP 块比较指令	(@)APR 数学处理指令
7	(@)XFER 块传送指令	(@)BSET 块设定指令	(@)ROOT 平方根指令	(@)XCHG 数据交换指令	(@)SLD 数(4－bit)左移指令	(@)SRD 数(4－bit)右移指令	(@)MLPX 4－16/8－256译码指令	(@)DMPX 16－/256－8编码指令	(@)SDEC 七段译码指令	(@)FDN 浮点数除法指令
8	(@)DIST 单字数据分配指令	(@)COLL 数据调用指令	(@)MOVB 位传送指令	(@)MOVD 数传送指令	(@)SFTR 可逆移位寄存器指令	(@)TCMP 表格比较指令	(@)ASC ASCII码转换指令	TTIM 累加定时器指令	ZCP 区域比较指令	(@)INT 中断控制指令
9	(@)SEND 网络发送指令	(@)SBS 子程序调用指令	SBN 子程序定义指令	RET 子程序返回指令	(@)WDT 监视定时器刷新指令			(@)IORF I/O刷新指令	(@)RECV 网络接收指令	(@)MCRO 宏指令

5－7－2 助记符按字母顺序排列的指令表

助记符	码	字节	名称	页
7SEG	–	4	七段显示码输出指令	317
ADB(@)	50	4	二进制加法指令	233
ADBL(@)	–	4	双字二进制加法指令	239
ADD(@)	30	4	BCD 码加法指令	218
ADDL(@)	54	4	双字 BCD 码加法指令	219
AND	无	1	与指令	142
AND LD	无	1	逻辑块与指令	142
AND NOT	无	1	与非指令	142
ANDW(@)	34	4	字逻辑与指令	265
APR(@)	69	4	数学处理指令	253
ASC(@)	86	4	ASCII 码转换指令	206
ASFT(@)	17	4	异步移位寄存器指令	170
ASL(@)	25	2	算术左移指令	166
ASR(@)	26	2	算术右移指令	167
AVG(@)	–	4	求平均值指令	249
BCD(@)	24	3	二进制→BCD 码指令	196
BCDL(@)	59	3	双字二进制→双字 BCD 码指令	197
BCMP(@)	68	4	块比较指令	189
BCNT(@)	67	4	位计数器指令	293
BIN(@)	23	3	BCD 码→二进制指令	195
BINL(@)	58	3	双字 BCD 码→双字二进制指令	196
BSET(@)	71	4	块设定指令	172
BXF2(@)	–	4	EM 组传送指令	182
CLC(@)	41	1	清进位指令	218
CMP	20	3	单字比较指令	185
CMPL	60	4	双字比较指令	187
CNT	无	2	计数器指令	158
CNTR	12	3	可逆计数器指令	161
COLL(@)	81	4	数据调用指令	176
COLM(@)	64	4	行→列转换指令	214
COM(@)	29	2	求反指令	264
CPS	–	4	带符号二进制比较指令	193
CPSL	–	4	以字带符号二进制比较指令	194
DBS(@)	–	4	带符号二进制除法指令	245
DBSL(@)	–	4	带符号双字二进制除法指令	246
DEC(@)	39	2	减量指令	218

（续表）

助记符	码	字节	名称	页
DIFD	14	2	下降沿微分指令	144
DIFU	13	2	上升沿微分指令	144
DIST(@)	80	4	单字数据分配指令	175
DIV(@)	33	4	BCD 码除法指令	226
DIVL(@)	57	4	双字 BCD 码除法指令	227
DMPX(@)	77	4	16→4/256→8 编码指令	202
DSW	－	4	数字开关输入指令	319
DVB(@)	53	4	二进制除法指令	238
EMBC(@)	－	2	选择 EM 组指令	302
END	01	1	结束指令	151
FAL(@)	06	2	出错报警指令	286
FALS	07	2	严重故障报警指令	286
FCS(@)	－	4	帧校验指令	294
FDIV(@)	79	4	浮点数除法指令	228
FPD	－	4	故障点检测指令	296
HEX(@)	－	4	ASCII 码→十六进制指令	209
HKY	－	4	十六进制键输入指令	323
HMS(@)	66	4	秒→小时指令	199
IEMS(@)	－	2	EM 间接寻址指令	301
IL	02	1	联锁指令	148
ILC	03	1	联锁解除指令	148
INC(@)	38	2	增量指令	217
INT(@)	89	4	中断控制指令	275
IORD(@)	－	4	特殊 I/O 单元读指令	330
IORF(@)	97	3	I/O 刷新指令	292
IOWR(@)	－	4	特殊 I/O 单元写指令	331
JME	05	2	跳转终了指令	150
JMP	04	2	跳转指令	150
KEEP	11	2	保持指令	146
LD	无	1	取指令	142
LD NOT	无	1	取非指令	142
LINE(@)	63	4	列→行转换指令	213
LMSG(@)	47	4	长信息指令	290
MAX(@)	－	4	寻找最大值指令	247
MBS(@)	－	4	带符号二进制乘法指令	243
MBSL(@)	－	4	带符号双字二进制乘法指令	244

（续表）

助记符	码	字节	名称	页
MCMP(@)	19	4	多字比较指令	183
MCRO(@)	99	4	宏指令	273
MIN(@)	－	4	寻找最小值指令	248
MLB(@)	52	4	二进制乘法指令	237
MLPX(@)	76	4	4→16/8→256 译码指令	199
MOV(@)	21	3	传送指令	171
MOVB(@)	82	4	位传送指令	179
MOVD(@)	83	4	数(4－bit)传送指令	179
MPRF(@)	61	4	组－2 高密度 I/O 刷新指令	293
MSG(@)	46	2	信息显示指令	289
MTR	－	4	矩阵输入指令	327
MUL(@)	32	4	BCD 码乘法指令	225
MULL(@)	56	4	双字 BCD 码乘法指令	226
MVN(@)	22	3	取反传送指令	172
NEG(@)	－	4	二进制求补指令	215
NEGL(@)	－	4	双字二进制求补指令	216
NOP	00	1	空操作指令	151
OR	无	1	或指令	142
OR LD	无	1	逻辑块或指令	142
OR NOT	无	1	或非指令	142
ORW(@)	35	4	逻辑或指令	266
OUT	无	2	输出指令	143
OUT NOT	无	2	输出非指令	143
PID(@)	－	4	PID 控制指令	256
RECV(@)	98	4	网络接收	305
RET	93	1	子程序返回指令	273
ROL(@)	27	2	循环左移指令	167
ROOT(@)	72	3	平方根指令	231
ROR(@)	28	2	循环右移指令	168
RSET	无	2	复位指令	145
RXD(@)	－	4	接收指令	310
SBB(@)	51	4	二进制减法指令	235
SBBL(@)	－	4	双字二进制减法指令	241
SBN	92	2	子程序定义指令	273
SBS(@)	91	2	子程序调用指令	271
SCAN(@)	18	4	循环时间指令	287

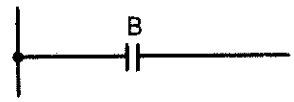
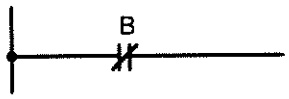
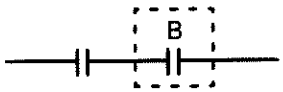
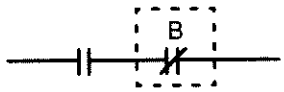
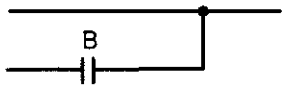
（续表）

助记符	码	字节	名称	页
SCL(@)		4	线性比例指令	211
SDEC(@)	78	4	七段译码指令	205
SEC(@)	65	4	小时→秒指令	198
SEND(@)	90	4	网络发送	303
SET	无	2	置位指令	145
SFT	10	3	移位寄存器指令	162
SFTR(@)	84	4	可逆移位寄存指令	165
SLD(@)	74	3	数(4－bit)左移指令	168
SNXT	09	2	步启动指令	278
SRCH(@)	－	4	数据搜索指令	299
SRD(@)	75	3	数(4－bit)右移指令	169
STC(@)	40	1	置进位指令	218
STEP	08	2	步定义指令	278
SUB(@)	31	4	BCD 码减法指令	221
SUBL(@)	55	4	双字 BCD 码减法指令	223
SUM(@)	－	4	求和指令	251
TCMP(@)	85	4	表格比较指令	191
TERM(@)	48	4	终端方式指令	291
TIM	无	2	定时器指令	152
TIMH	15	3	高速定时器指令	156
TKY(@)	－	4	十进制键输入指令	325
TRSM	45	1	跟踪存贮器采样指令	288
TTIM	87	4	累加定时器指令	157
TXD(@)	－	4	发送指令	311
WDT(@)	94	2	监视定时器刷新指令	292
WSFT(@)	16	3	字移位指令	169
XCHG(@)	73	3	数据交换指令	174
XDMR(@)	－	4	扩展 DM 读指令	300
XFER(@)	70	4	块传送指令	174
XFR2	－	4	EM 块传送	182
XFRB(@)	62	4	多位传送指令	180
XORW(@)	36	4	异或门逻辑指令	267
XNRW(@)	37	4	异或非门逻辑指令	267
ZCP	88	4	区域比较指令	191
ZCPL	－	4	双字区域比较指令	193

5－8 梯形图指令

梯形图指令包括梯形指令和逻辑块指令并和梯形图上的条件一致。逻辑块指令用于连结比较复杂的部分。

5－8－1 取(LOAD)、取非(LOAD NOT)，与(AND)、与非(AND NOT)、或(OR)、或非(OR NOT)指令

	梯形符号	操作数据区域
取指令－LD		B:位 IR, SR, AR, HR, TC, LR, TR
取非指令－LD NOT		B:位 IR, SR, AR, HR, TC, LR
与指令－AND		B:位 IR, SR, AR, HR, TC, LR
与非指令－AND NOT		B:位 IR, SR, AR, HR, TC, LR
或指令－OR		B:位 IR, SR, AR, HR, TC, LR
或非指令－OR NOT		B:位 IR, SR, AR, HR, TC, LR

限制

只要不超出 PC 的内存容量，这些指令的使用次数和顺序不受任何限制。

说明：

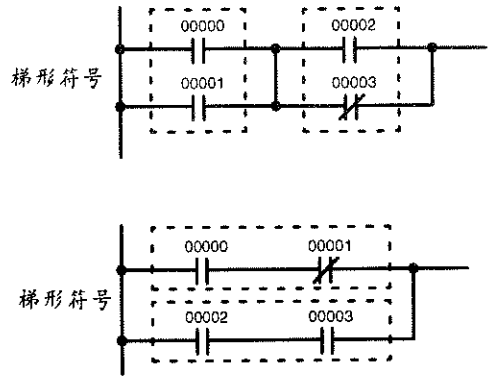
这六条基本指令对应于梯形图上的条件。如第四章编写和输入程序所述，每条指令指定位的状态决定其它所有指令的执行条件。每一条指令和每一位都可以根据需要重复使用。

用于 LD 或 LD NOT 指令的位操作数(B)的状态决定指令执行的第一个执行条件。AND 指令在操作条件和它的位操作之间起着逻辑与的作用；AND NOT 指令在执行条件与它的位操作数状态非之间起逻辑与作用。OR 指令在操作条件与它的位操作数状态间起着逻辑或作用；OR NOT 指令在执行条件与它的位操作数状态非之间起逻辑或作用。取 TR 位的梯形符号与上述指令不同，见 4－4－3 梯形指令章节。

标志

以上指令执行不影响标志位状态。

5－8－2 程序块与(AND LOAD)和程序块或(OR LOAD)指令



说明

不能简单地采用 OR 和 AND 逻辑连接块的组合，而应该采用 AND LD 和 OR LD 程序块与和或指令。AND 和 OR 操作用于执行条件与一位状态逻辑地连接，而 AND LD 和 OR LD 逻辑地连接两个执行条件，即当前执行条件与上一个还未使用的执行条件。

为了画梯形图，不必使用 AND LD 和 OR LD 指令，当直接输入梯形图，如用梯形图支持软件(LSS)输入时，也不必使用这些指令。只是在转换程序成助记符形式并输入时才需要它们。这些过程及不同过程的限制和举例在 4－7 章节“程序的输入，修改和检查”中论述。

为减少编程所需指令数，应基本理解逻辑块指令，指令详见 4－4－6“逻辑块指令”章节。

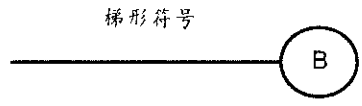
标志

这些指令的执行不影响标志位

5－9 位(Bit)控制指令

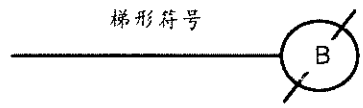
有五条常用以控制单个位(Bit)状态的指令，它们是 OUT，OUT NOT，DIFU(13)，DIFD(14)和 KEEP(11)指令。这些指令以不同的方法使位接通(ON)或断开(OFF)。

5－9－1 输出和输出非指令——OUT 和 OUT NOT



操作数据区域

B：位
IR，SR，AR，HR，TC，LR，TR



操作数据区域

B：位
IR，SR，AR，HR，TC，LR

限制

任何输出位(Bit)通常只能用于一条控制它状态的指令。详细请参阅 3－3“IR 区域”。

说明

根据指令执行条件，OUT 和 OUT NOT 指令用于控制指定位的状态。

当执行条件为 ON，OUT 指令使指定位输出 ON，执行条件为 OFF，OUT 指令指定位输出 OFF。对于 TR 位(Bit)，OUT 指令用于分支点而不是在指令行最后。详见 4－7－7“指令行的分支”。

当指令执行条件为 OFF，OUT NOT 指令使指定位输出为 ON，反之，执行条件为 ON，指定位输出 OFF。

通过改变梯形图中指定的条件位的 ON 和 OFF，控制执行 OUT 和 OUT NOT 指令，这样它们确定了其

它指令的执行条件。这一点是相当有用的;它可用一个复杂的控制条件去控制一个工作位(Bit)的状态,再由这工作位去控制其它指令。

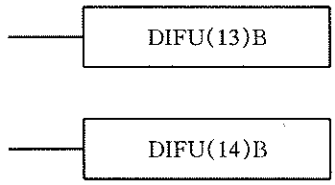
某一位 ON 或 OFF 状态时间的长短可由 OUT 和 TIM 或 OUT NOT 和 TIM 指令的组合来控制。详见 5－14－1 章节"定时器指令—TIM"例子。

标志

以上指令执行不影响标志状态。

5－9－2 上升沿微分和下降沿微分指令——DIFU(13)和 DIFD(14)

梯形符号



操作数数据区域

B:位
IR, AR, HR, LR

B:位
IR, AR, HR, LR

限制

任何输出位(Bit)通常只能用于一条控制其状态的指令。详见 3－3"IR 区域"。

说明

DIFU(13)和 DIFD(14)指令用于接通(ON)指定位(Bit)仅一个扫描周期。

每当执行 DIFU(13)指令,它总是将当前执行条件与前一次比较。如果前一次执行条件为 OFF,而当前执行条件为 ON,DIFU(13)指令使指定位变为 ON 状态。假如前次指令条件为 ON,当前执行条件是 ON 或 OFF,DIFU(13)指令使指令控制位为 OFF 或保持 OFF 状态(假定指定位原来已处于 OFF 状态)。假定每个扫描周期都执行 DIFU(13)指令,指令指定位为 ON 状态的时间永远不会大于一个扫描周期(见下面注意事项)。

每当执行 DIFD(14)指令,它总是将当前执行条件与前一次比较。如果前一次执行条件为 ON,当前执行条件为 OFF,DIFD(14)指令使指令位变成 ON 状态。假如前次指令条件为 OFF,当前执行条件为 ON 或 OFF,DIFD(14)指令使指令指定位为 OFF 或保持 OFF。假定每次扫描周期都执行 DIFD(14)指令,指令指定位为 ON 状态时间永远不会大于一个扫描周期(见下面注意事项)。

当希望一条特殊的指令执行一个扫描周期,而指令前冠以@标记的微分型指令又不能使用时,就可以使用上面介绍的微分指令。微分指令也可与非微分型指令(该指令具有微分形式)一起使用以使编程简化。见下面列子。

标志

以上指令执行不影响标志位状态。

注意事项

当 DIFU(13)和 DIFD 指令处于 IL 和 ILC 指令间,或处于 JMP 和 JME 间,以及在子程序中,指令的操作结果不定。详见 5－10"联锁和联锁解除指令—IL(02)和 ILC(03)";5－11"跳转和跳转终了指令—JMP(04)和 JME(05)"和 5－23"子程序与中断控制指令"章节。

例 1:无微分指令

下图 A 中,只要指令执行条件为 ON,CMP(20)指令就执行,它将 HR10 和 DM0000 两个操作数的内容作比较,并根据比较结果设置算术标志(GR,EQ 和 LE)。如果执行条件一直为 ON,被比较的一个或两个操作数内容发生变化,标志位状态在每个扫描周期中也会随之变化。但图 B 中的例子表明如何使用 DIFU(13)指令使得 CMP(20)指令只在每次所需的执行条件变成 ON 时才执行一次。

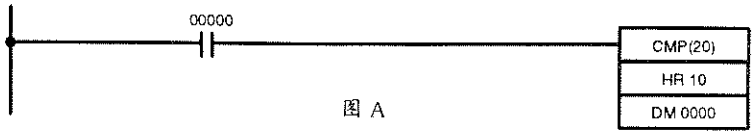


图 A

地址	指令	操作数
00000	LD	00000
00001	CMP(20)	
		HR 10
		DM 0000

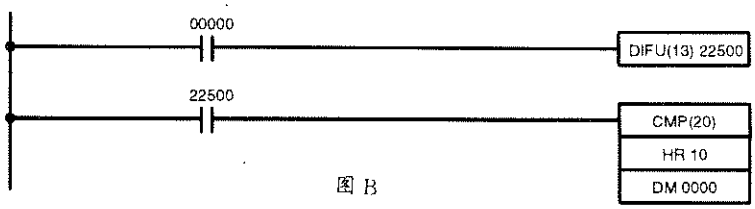
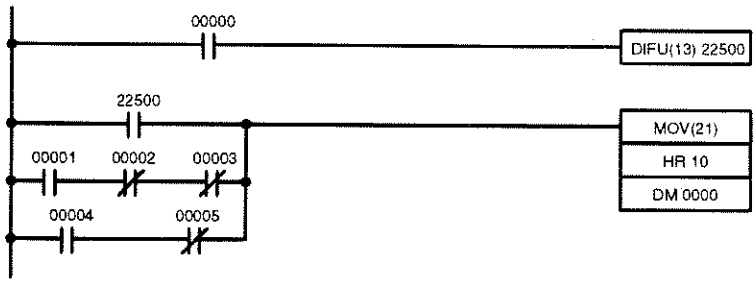


图 B

地址	指令	操作数
00000	LD	00000
00001	DIFU(13)	22500
00002	LD	22500
00003	CMP(20)	
		HR 10
		DM 0000

例 2:简化编程

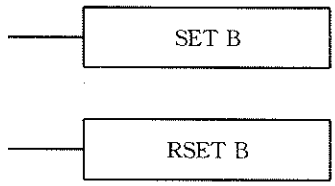
虽然 MOV(21)指令具有微分形式,由于数个执行条件中只有一个条件需要微分处理,用 MOV(21)微分型指令会使下面梯形图非常复杂。



地址	指令	操作数
00000	LD	00000
00001	DIFU(13)	22500
00002	LD	22500
00003	LD	00001
00004	AND NOT	00002
00005	AND NOT	00003
00006	OR LD	—
00007	LD	00004
00008	AND NOT	00005
00009	OR LD	—
00010	MOV(21)	
		HR 10

5－9－3 置位和复位指令——SET 和 RSET

梯形符号



操作数据区域

B:位
IR, SR, AR, HR, LR

B:位
IR, SR, AR, HR, LR

说明

当 SET 指令的执行条件为 ON,指令置操作数位(ON),当执行条件为 OFF,SET 指令不改变操作数位状态。当 RSET 指令执行条件为 ON,指令使操作数值复位(OFF),当执行条件为 OFF,RSET 指令不改变操作数值状态。

SET 指令的作用与 OUT 指令不同之处在于当 OUT 指令的执行条件为 OFF 时，OUT 指令会使操作数位变为 OFF。同理，RSET 指令也与 OUT NOT 指令不同，OUT NOT 指令在执行条件 OFF 时使操作数位变成 ON 状态。

注意事项

SET 和 RSET 指令编在 IL(02)和 ILC(03)或 JMP(04)和 JME(05)指令间，当联锁或跳转条件符合(即 IL(02)或 JMP(04)指令处于 OFF 执行条件)，SET 和 RSET 指令控制的操作数位不变。

标志

以上指令执行不影响标志位状态。

例

下例中演示了 OUT 和 SET/RSET 指令间的区别。在第一个例子(图 A)中，IR10000 状态随着 IR00000 ON/OFF 变化而对应变化。

第二个例子(图 B)中，IR10000 在 IR00001 接通(ON)时也变成 ON，并保持 ON 状态(即使 IR00001 变成 OFF 状态)直到 IR00002 从 ON 变成 OFF 为止。

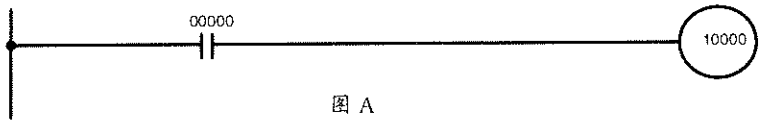


图 A

地址	指令	操作数
00000	LD	00000
00001	OUT	10000

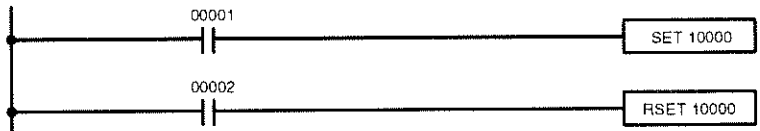
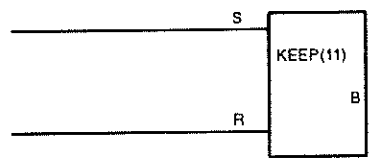


图 B

地址	指令	操作数
00000	LD	00001
00001	SET	10000
00002	LD	00002
00003	RSET	10000

5－9－4 保持指令——KEEP(11)

梯形符号



操作数据区域

B:位
IR, AR, HR, LR

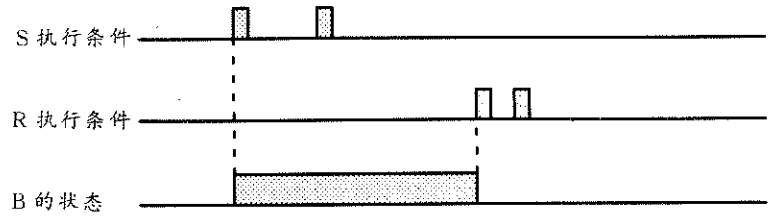
限制

任何输出位(Bit)通常只能用于一条控制其状态的指令。详见 3－3“IR 区域”。

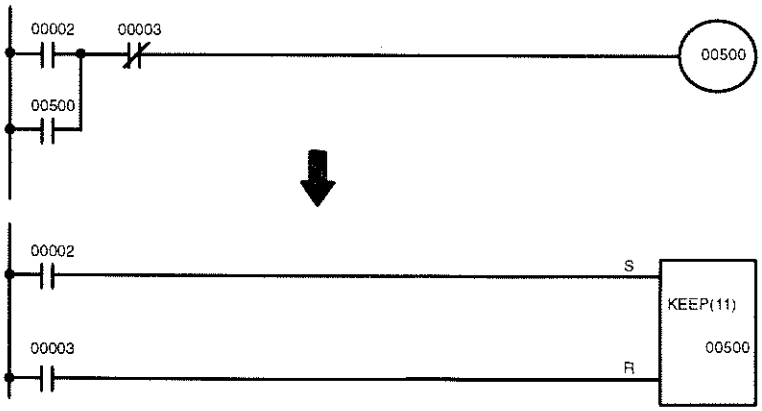
说明

根据两个执行条件，KEEP(11)指令用于保持指定位的状态。这两个执行条件标以 S 和 R 符号。S 是置位输入；R 是复位输入。KEEP(11)指令操作就象一个用 S 置位，用 R 复位的锁存继电器。

当 S 为 ON，指定位也变成 ON，此后无论 S 端保持 ON 状态还是变为 OFF 状态，指定位 ON 状态一直保持到复位信号有效。当 R 为 ON，指令定位变成 OFF 状态，此后无论 R 端保持 ON 还是变成 OFF 状态，指定位 OFF 状态一直保持到置位信号有效。S 与 R 执行条件和 KEEP(11)指令位状态之间关系见下图。



KEEP(11)指令操作如同 4－8－3“自保持位”章节所描述的。下面两图程序功能完全一样，然而用 KEEP(11)指令编程可少用一条指令，且即使指令在联锁程序段内，指令指定位也能保持其状态。



地址	指令	操作数
00000	LD	00000
00001	OR	00500
00002	AND NOT	00003
00003	OUT	00500

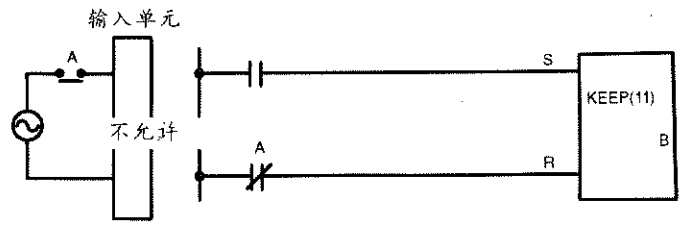
地址	指令	操作数
00000	LD	00002
00001	LD	00003
00002	KEEP(11)	00500

标志

本指令执行不影响标志位状态。

注意事项

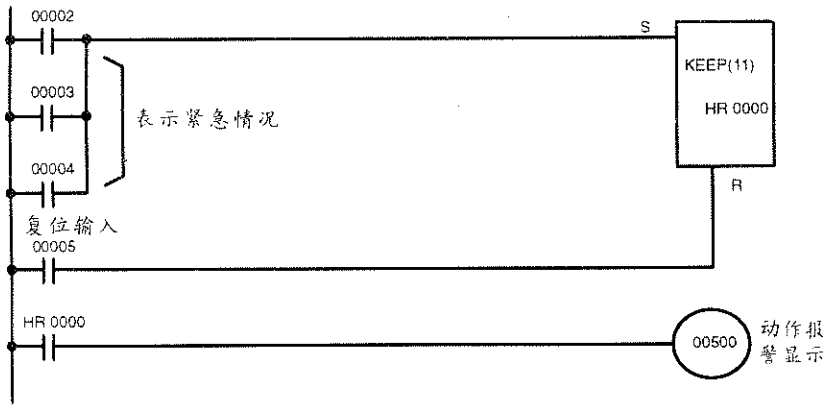
小心 KEEP 复位端由外部常闭元件控制时的异常现象。当输入元件用交流电源驱动时，千万别用输入位的非条件控制 KEEP(11)指令的复位端。关断 PC 的直流电源产生的电压延时消失(相对于输入元件用交流驱动)会引起 KEEP(11)指定位复位。情形如下图示。



在联锁程序段中的 KEEP 指令指定位不复位。详见 5－10“联锁和联锁解除指令—IL(02)和 ILC(03)”章节。

例

若使用了 HR 位或 AR 位，即使在电源中断期间位状态仍保持不变。因此，KEEP(11)指令可用于使位信号在电源中断后再起动 PC 仍保持不变的编程。如下图是一个用于紧急状态下关掉系统后处理报警指示的例子。00002，00003 和 00004 位变成 ON 状态，以指示某些故障，00005 位信号为 ON，则清除故障指示。当这三位中任何一位指示紧急状态，HR0000 就变成 ON，并通过 00500 位报警。



地址	指令	操作数
00000	LD	00002
00001	OR	00003
00002	OR	00004
00003	LD	00005
00004	KEEP(11)	HR 0000
00005	LD	HR 0000
00006	OUT	00500

KEEP(11)指令还可与定时器 TIM 结合，使接通(ON)和断开(OFF)产生延时。详见 5－14－1“定时器指令——TIM。”

5－10　联锁和联锁解除指令——IL(02)和 ILC(03)

梯形符号　IL(02)

梯形符号　ILC(03)

说明

IL(02)指令总是与 IL(03)指令一起使用，建立联锁功能。联锁和采用 TR 位一样都可用于分叉处理，但在 IL(02)指令执行条件不满足(OFF)时，处于 IL(02)和 ILC(03)之间的指令处理与 TR 位的处理结果不一样。如果 IL(02)执行条件为 ON，程序如所写的程序那样执行，执行条件满足(ON)，从 IL(2)所处的指令点到下一个 ILC(03)指令间的每行指令允许执行。参见 4－7－7“指令行的分支”中两种方法基本说明。

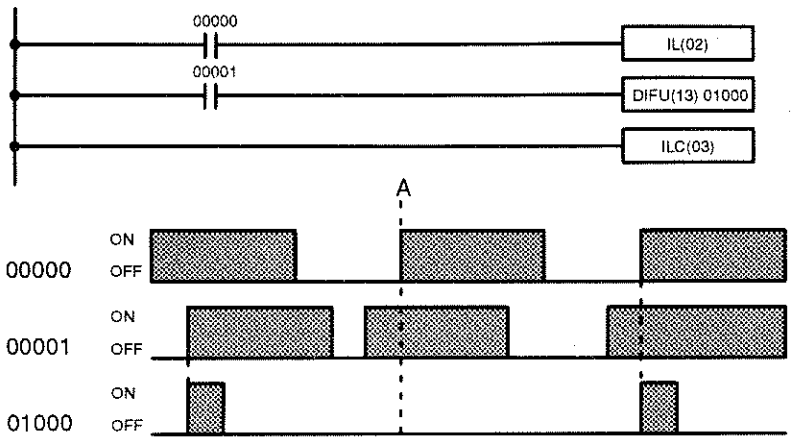
如果 IL(02)执行条件不满足(OFF)，在 IL(02)和 ILC(03)指令间的联锁部分的程序处理如下表所示：

指令	处理
OUT 和 OUT NOT	指定位变为 OFF
SET 和 RSET	位状态保持
TIM 和 TIMH(15)	复位
TTIM(87)	PV 保持
CNT.CNTR(12)	PC 保持
KEEP(11)	位状态保持
DIFU(13)和 DIFD(14)	不执行(见下面第二段)
其它指令	不执行

IL(02)和 ILC(03)并不一定要成对使用。IL(02)指令在一段程序中可使用多次，在每条 IL(02)指令到下一条 ILC(03)指令间产生联锁功能。除非在上一个 ILC(03)和现 ILC(03)指令之间存在至少一条 IL(02)指令，否则 ILC(03)指令不能使用。

DIFU(13)和 DIFD(14)指令在联锁中间

如果 DIFU(13)或 DIFD(14)指令在联锁程序部分中，且 IL(02)指令执行条件不满足(OFF)，那么 DIFU(13)或 DIFD(14)的执行条件变化情况不记存。IL(02)指令执行条件变为 ON 后，DIFU(13)或 DIFU(14)指令马上执行，DIFU(13)或 DIFD(14)指令的执行条件将与联锁变成有效前的执行条件比较(即 IL(02)联锁条件变成 OFF 前)。梯形图与位状态变化情况见下图。在 00000 处于 OFF 状态期间，联锁有效。注意，在标有 A 的这点，尽管 00001 已变为 OFF 状态又变回 ON 状态，01000 仍然不能变成 ON 状态。



地址	指令	操作数
00000	LD	00000
00001	IL(02)	
00002	LD	00001
00003	DIFU(13)	01000
00004	ILC(03)	

注意事项

在一条或多条 IL(02)指令后一定有一条 ILC(03)指令。

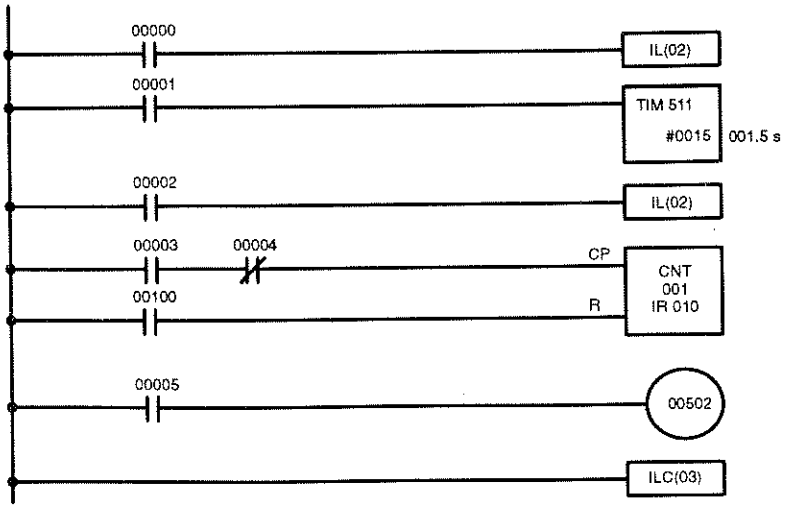
虽然多条 IL(02)指令后可只用一条 ILC(03)指令，但如果 ILC(03)指令间没有至少一条 IL(02)指令，ILC(03)指令不允许连续使用，即联锁指令不能嵌套使用。只要有一条 ILC(03)指令执行，在作用的 ILC(03)和先前的 ILC(03)间的所有联锁全部解除。如果多条 IL(02)指令后使用一条 ILC(03)指令，程序检查时会有出错信息显示，而动作还按程序正常进行。

标志

以上指令执行不影响标志位状态。

例

下面梯形图说明用两次 IL(02)指令与一条 ILC(03)指令情况。

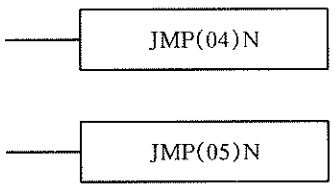


地址	指令	操作数
00000	LD	00000
00001	LD(02)	
00002	IL	00001
00003	TIM	511
		# 0015
00004	LD	00002
00005	IL(02)	
00006	LD	00003
00007	AND NOT	00004
00008	LD	00100
00009	CNT	001
		010
00010	LD	00005
00011	OUT	00502
00012	ILC(03)	

当第一个 IL(02)指令的执行条件(00000)为 OFF 状态，TIM511 被复位至 1.5 秒，CNT001 不变，00502 变为 OFF 状态，如果第一个 IL(02)执行条件为 ON 状态，第二个 IL(02)执行条件(00002)为 OFF 状态，TIM511 将根据执行条件 00001 状态而执行，CNT001 仍不变，00502 为 OFF 状态。如果两个 IL(02)指令的执行条件都为 ON，程序执行如同没有联锁指令一样。

5－11　跳转和跳转终了指令——JMP(04)和 JME(05)

梯形符号



定义符数值

N:跳转号
#(00～99)

N:跳转号
#(00～99)

限制

跳转号从 01 到 99 只能为 JMP(04)指令用一次和为 JME(05)指令用一次，即每个号只能定义一个跳转。跳转号 00 却可以根据需要次数多次使用。

说明

JMP(04)指令总是和 JME(05)指令结合使用产生跳转，即从梯形图中一点跳到另一点。JMP(04)指令定义跳转起始点；JME(05)指令定义跳转终了点。当 JMP(04)执行条件为 ON，跳转不执行，程序执行按书写程序连续执行。当 JMP(04)执行为 OFF，跳转至与 JMP(04)指令相同编号的 JME(05)指令处，执行 JME(05)后的指令。

如果 JMP(04)指令的跳转号在 01 至 99 间，当跳转执行，程序立即跳到与 JMP 相同跳转号的 JME(05)处，跳转指令间的任何指令都不执行。在 JMP(04)和 JME(05)间的定时器，计数器，OUT 和 OUT NOT 指定位状态以及指令控制的所有其它状态位都不发生变化。由于在 JMP(04)和 JME(05)间的所有指令被跳

过，所以跳转号 01 到 99 的跳转可用于减少周期时间。

如果 JMP(04)的跳转号是 00，CPU 将寻找一个跳转号为 00 的 JME(05)，为此跳转执行期间 CPU 必须在程序中搜索。相比其它转跳指令，该寻找工作(当转跳指令执行条件 OFF 时)会造成更长的扫描周期。JMP(04)00 和 JME(05)00 指令间的定时器、计数器、OUT 或 OUT NOT 指定位以及指令控制的所有其它状态不发生变化。跳转号 00 可根据需要多次使用。在程序中，JMP(04)00 总是跳转到下一个 JME(05)00 处，因此，可以连续使用 JMP(04)00 指令，与它们对应的用同一个 JME(05)00 匹配。然而连续使用 JME(05)00 是没有意义的，因为所有 JMP(04)00 只会跳转到第一条 JME(05)00 指令处。

DIFU(13)和 DIFD(14)指令在跳转指令间。

虽然 DIFU(13)和 DIFD(14)指令为一个扫描周期接通一指定的位，当它们在 JMP(04)和 JME(05)间时就不必定如此。一旦 DIFU(13)或 DIFD(14)使某位为 ON，ON 状态将一直保持到下一次 DIFU(13)或 DIFD(14)指令再次执行。一般程序中这意味下一次扫描。但在跳转程序中，这意味下一次不执行从 JMP(04)跳到 JME(05)的扫描周期，换而言之，如果 DIFU(13)或 DIFD(14)使某位为 ON，在下个周期中跳转执行，那么 DIFU(13)或 DIFD(14)指令被跳过不执行，这位状态将保持 ON 直到下一次扫描周期中 JMP(04)指令执行条件为 ON，跳转下执行。

注意事项

如果 JMP(04)和 JME(05)指令不成对使用，执行程序检查时会给出出错信息。虽然由此出现出错信息，程序仍按写入程序执行。

标志

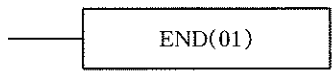
以上指令执行不影响标志位状态。

例子

跳转程序的例子在 4－7－8 章节“跳转”中给出。

5－12 结束指令——END(01)

梯形符号



说明

在任何程序中，END(01)用于程序最后一条指令。如果有子程序，END(01)放在最后一个子程序后。END(01)指令后面的任何指令都不执行。当有时需调试程序时，END(01)指令可插入到所需调试段处，执行该点前所有指令。但调试后必须删去 END(01)指令以执行后续程序。

程序中若无 END(01)指令，指令不执行且显示出错信息“NO END INST”。

标志

END(01)指令将 ER，CY，GR，EQ 和 LE 标志变为 OFF。

5－13 空操作指令——NOP(00)

说明

NOP(00)指令在程序中一般不需要，并且该指令无梯形符号。当程序中遇到 NOP(00)，PC 不做任何事，程序执行下一条指令。编程前内存无信息内容，NOP(00)写入所有地址。NOP(00)可由 00 功能码输入。

标志

指令执行不影响标志位状态。

5－14 定时器和计数器指令

TIM 和 TIMH 都是需要 TC 编号和设定值(SV)的递减接通延时定时器指令。

CNT 和 CNTR 分别是递减计数和可逆计数指令。两指令都需要 TC 编号和 SV，指令也与多指令行输入信号和复位信号相连。

任一 TC 编号不能重复定义，即一旦 TC 某一编号用作定时器或计数器指令定义符，此 TC 号不能再作它用。TC 编号定义后，除定时器和计数器指令外，它可根据需要作为指令操作数反复使用。

TC 编号范围从 000 到 511，TC 编号作为定时器或计数器指令定义符时不需要加前缀，一旦定义成定时器，TC 编号前加 TIM 以用作一些指令的操作数。使用前缀 TIM 和用于定义定时器的定时器指令无关。一旦定义成计数器，TC 编号前加 CNT 以用作一些指令的操作数。CNT 也与定义计数器的计数器指令无关。

TC 编号既可被指定为位操作数也可被指定为字操作数。当指定的操作数需要位数据，TC 编号访问表示定时/计数到的"结束标志"位，该位平时为 OFF 状态，当指定的设定值达到，状态变为 ON。当指定的操作数需要字数据，TC 编号访问存有定时器或计数器当前值(PV)的存储器。这样，定时器或计数器当前值可用作 CMP(20)比较指令的操作数或允许使用 TC 区域的其它指令。此过程是由定义为定时器或计数器的指定 TC 编号访问存放该定时器或计数器当前值存储器完成的。

注意"TIM000"用作以 TC000 编号定义的定时器指令，指定该定时器结束标志和当前值。这一组字符的含意应该很清楚，即，第一个是指令，第二个是位操作数，第三个总是字操作数。这对于其它冠以 TIM 和 CNT 的所有 TC 编号同样正确。

设定值(SV)可以一个常数或在数据区中字的地址形式输入。如果指定为输入单元的 IR 区域字被指定为字地址，输入单元可与拨盘开关或其它类似元件相连，这样设定值可由外部器件设定。以这种方法连线的定时器/计数器只有在 PC 处于运行(RUN)和监视(MONITOR)方式下，才可以从外部来输入其设定值。所有设定值包括外部设定，都必须是 BCD 码。

5－14－1 定时器指令——TIM

梯形符号



定义符数值

N:TC 编号
#(000～511)

操作数数据区域

SV:设定值(字,BCD 码)
IR, AR, DM, HR, LR, #

限制：

设定值范围 000.0～999.9，小数点不输入。

每一 TC 编号只能用于一个定时器或计数器指令定义符。

如果需要(TIMH(15))高速定时器指令，TC000～TC015 不能用作一般定时器，详见 5－14－2"高速定时器指令——TIMH(15)"章节。

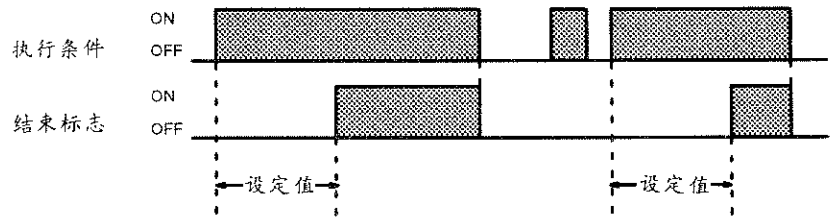
说明

当定时器执行条件为 ON，定时器定时，当执行条件为 OFF，定时器复位(当前值回复到设定值)。定时器一开始计时，以 0.1 秒为单位从设定值(SV)开始计数。

如果执行条件保持 ON 状态直到定时时间为零，TC 编号的定时结束标志变成 ON 并一直保持到定时器

复位(即定时器执行条件为 OFF 状态)。

下图说明定时器指令执行条件与对应定时结束标志间的关系。



注意事项

当 IL(02)联锁指令执行条件为 OFF,在联锁程序段中的定时器复位。定时器在电源中断时也将复位。如果希望定时器在以上条件下不复位,可借助 SR 区时钟脉冲位生成使用 CNT 的定时器。详见 5－14－4 "计数器指令——CNT"章节。如果设定值不是 BCD 码,程序们继续执行,但定时将不正确。

标志

ER:设定值不是 BCD 码。

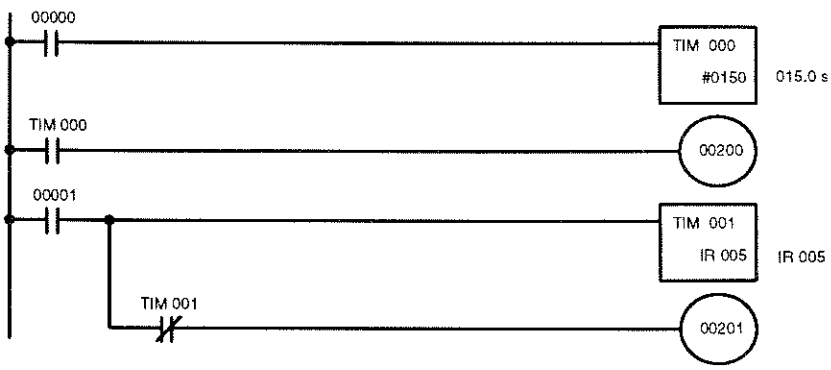
间接寻址 DM 字不存在。(* DM 字的内容不是 BCD 码或超出 DM 区域)

例

以下所有例子都在梯形图中使用 OUT,这些梯形图通常用于控制 IR 区域中的输出位,因此完全可以修改梯形图以控制其它指令的执行。

例 1:基本应用

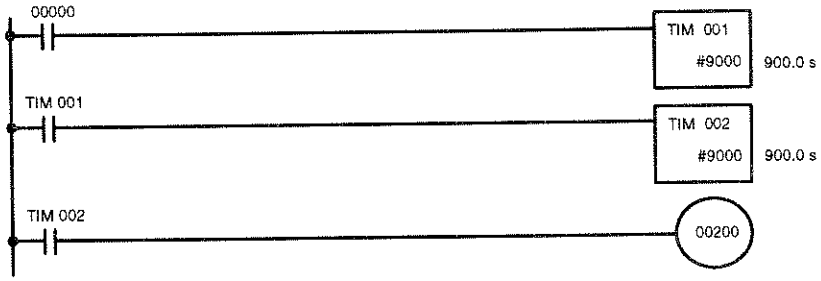
下例演示两个定时器,一定时器用常数设定,另一定时器通过输入字 005 设定定时值。在 00000 接通(ON)并至少保持 15 秒以上,00200 变为 ON。00000 变为 OFF 时,定时器复位,00200 变为 OFF 状态。当 00001 变为 ON,定时器 001 从 IR 字 005 提供设定值开始减值定时。00001 变为 ON 同时 00201 位也变为 ON。当 IR005 中的设定值定时满,00201 位变为 OFF。当 TIM001 复位(00001OFF),00201 位无论设定值是否定时到也变成 OFF 状态。



地址	指令	操作数
00000	LD	00000
00001	TIM	000
		# 0150
00002	LD	TIM 000
00003	OUT	00200
00004	LD	00001
00005	TIM	001
		005
00006	AND NOT	TIM 001
00007	OUT	00200

例 2:延长定时时间

有两种方法使定时器延时时间超过 999.9 秒。方法一是连续编制定时器,每个定时器定时结束标志用于起动下一个定时器。简单的例子是两个 900.0 秒(15 分钟)定时器结合成为一个 30 分钟功能定时器。



地址	指令	操作数
00000	LD	00000
00001	TIM	001
		# 9000
00002	LD	TIM 001
00003	TIM	002
		# 9000
00004	LD	TIM 002
00005	OUT	00200

例子中，00200 在 00000 变成 ON 的 30 分钟后接通。

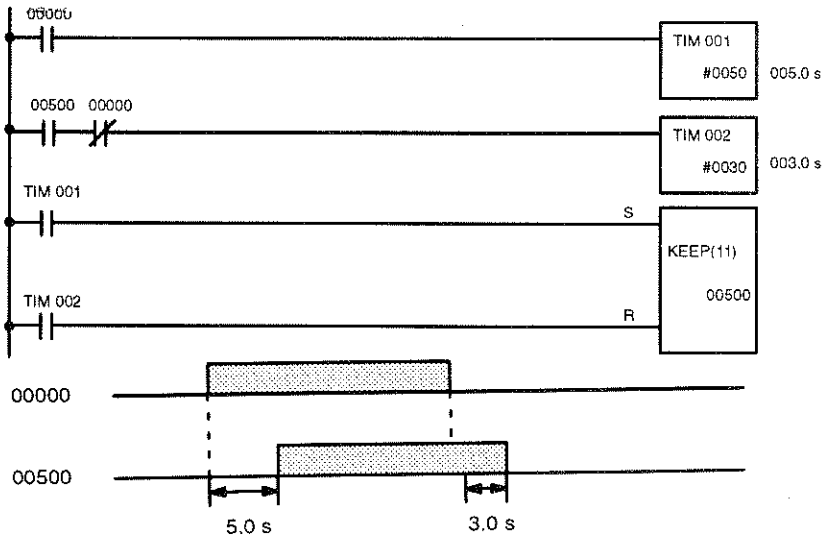
定时器可与计数器结合，也可采用计数器，对 SR 区域时钟脉冲位计数以延长定时器。例子在 5－14－4 指令“计数器指令—CNT”章节中。

例 3：接通/断开延时

TIM 可与 KEEP(11)指令结合，根据需要延时一个位的接通(ON)和断开(OFF)。KEEP(11)指令在 5－9－4“保持指令 KEEP (11)”中叙述。

为产生延时，两个定时器的计定结束标志用于控制 KEEP (11)指令指定位的置位和复位执行条件。要延时操作的位用于 KEEP (11)指令中，KEEP (11)指令指定位置位和复位延时由两个定时器设定值决定。自然，如果需要，两设定值完全可以一样。

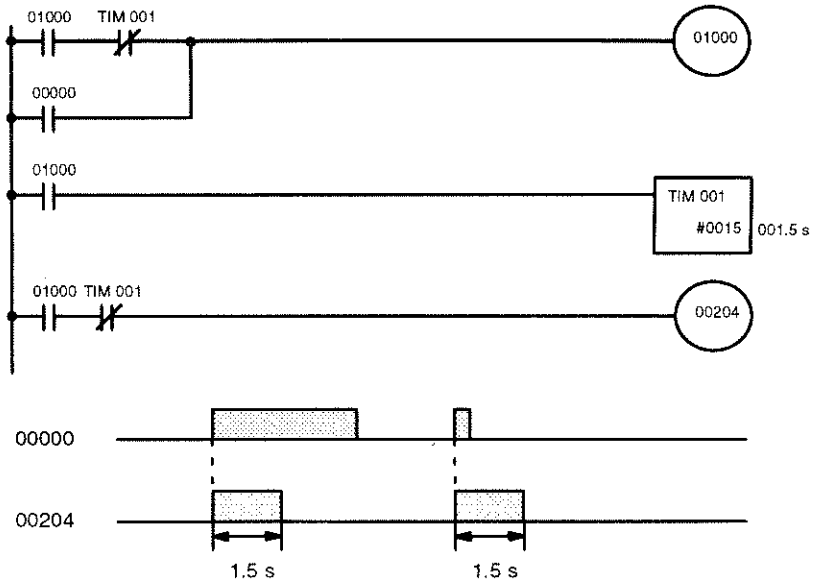
下例中，00500 在 00000 接通 5 秒后接通，在 00000 断开 3 秒后断开。必须用 00500 和 00000 两个条件决定 TIM002 执行条件；当 00000 接通，00000 的非状态对 TIM002 复位，当 00000 断开，需要 00500 起动 TIM002。



地址	指令	操作数
00000	LD	00000
00001	TIM	001
		# 0050
00002	LD	00500
00003	AND NOT	00000
00004	TIM	002
		# 0030
00005	LD	TIM 001
00006	LD	TIM 002
00007	KEEP(11)	00500

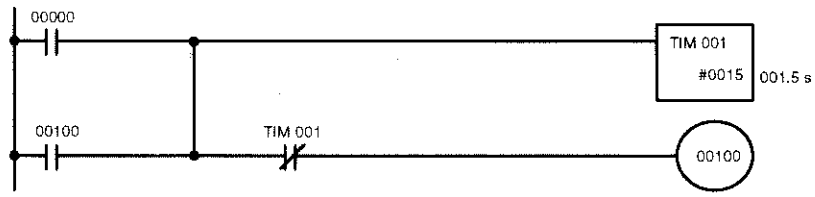
例 4：单稳位

某一位 ON/OFF 状态保持时间的长短可由定时器与 OUT 或 OUT NOT 的组合来控制。下图说明此功能如何实现。例子中，00204 在 00000 变为 ON 后保持 1.5 秒 ON 状态，与 00000 保持 ON 时间长短无关。因为 01000 在 00000 作用下自保，并由此接通 00204。当 TIM001 接通(即 TIM001 设定值定时到)。00204 通过 TIM001 控制变为断开(OFF 状态)(即 TIM001 接通，其非状态使 OUT00204 指令执行条件为 OFF)。



地址	指令	操作数
00000	LD	01000
00001	AND NOT	TIM 001
00002	OR	00000
00003	OUT	01000
00004	LD	01000
00005	TIM	001
		# 0015
00006	LD	01000
00007	AND NOT	TIM 001
00008	OUT	00204

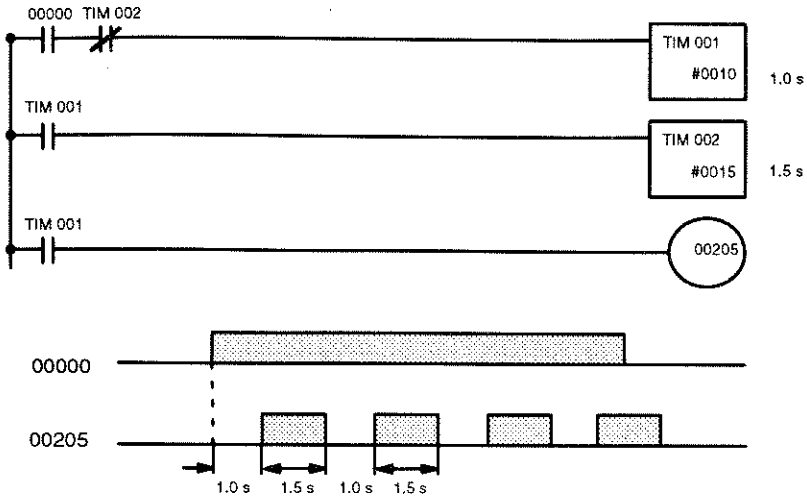
下面的单稳定时器可节省内存。



地址	指令	操作数
00000	LD	00000
00001	OR	00100
00002	TIM	001
		# 0015
00003	AND NOT	TIM 001
00004	OUT	00100

例 5：位闪烁

用两个定时器，在指定执行条件为 ON 期间可编程使位状态 ON 和 OFF 按规定的间隔翻转。一个定时器用以通断指定位，即这一定时器的完成标志位控制该指令位 ON/OFF。另一定时器功能是控制第一个定时器的操作，即当第一个定时器完成标志位为 ON，第二个定时器开始定时，当第二个定时器完成标志位为 ON，第一个定时器复位。



地址	指令	操作数
00000	LD	00000
00001	AND NOT	TIM 002
00002	TIM	001
		# 0010
00003	LD	TIM 001
00004	TIM	002
		# 0015
00005	LD	TIM 001
00006	OUT	00205

一个简单但不太灵活的闪烁的方法是将执行条件和一个 SR 区域的时钟脉冲位相与来实现位闪烁。当执行条件为 ON，位闪烁就工作。虽然这里没用定时器，但用此例仅为比较。由于 ON 和 OFF 时间必须一样，可用的时钟脉冲位取自于 SR 区域，所以这方法受到了一定的限制。

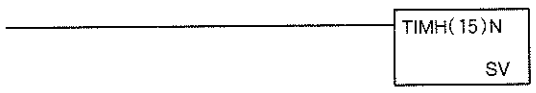
下例中使用了(25502)1 秒脉冲信号位，所以 00206 每秒 ON 和 OFF 翻转一次，即 ON 0.5 秒，OFF 0.5 秒。00206 精确定时和初始状态取决于 00000 为 ON 时时钟脉冲的状态。



地址	指令	操作数
00000	LD	00000
00001	AND	25502
00002	OUT	00206

5－14－2 高速定时器指令——TIMH (15)

梯形符号



定义符数值

N：TC 编号
#(000～511 优选 000～015)

操作数数据区域

SV：设定值(字，BCD 码)
IR，AR，DM，HR，LR，#

限制

设定值(SV)是在00.00～99.99之间,(虽然可以设置00.00和00.01数值,但00.00使定时器无法工作,即马上接通定时结束标志,00.01运行也不可靠。)数值中小数点不用输入。

作为定义符,每个TC编号只能用于一个定时器或计数器指令。

如果扫描周期大于10ms,必须使用TC000～TC015以保证计时准确。定时器000～015的当前值和结束标志每10ms刷新一次,但定时器016～511只在程序每执行TIMH(15)指令时才刷新一次。

说明

除TIMH的计量单位为0.01秒外,TIMH(15)操作与TIM指令一样。

如果使用TC016到TC511,扫描周期会影响TIMH(15)指令执行的准确性。若扫描周期大于10ms,请使用TC000～TC015。

详见5－14－1"定时器指令——TIM"章节中的内容与例子。除上面讲的以外,指令的所有方面都一样。

注意事项

联锁程序段中的定时器在IL(02)指令执行条件为OFF时被复位。电源中断时定时器也复位。如果希望定时器在以上条件下不复位,可采用SR区时钟脉冲位,生成使用CNT的定时器。详见5－14－4"计数器指令——CNT"章节。

如果设定值不是BCD码,程序仍继续执行,但定时不准确。

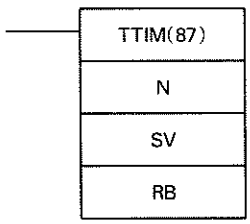
标志

ER:设定值不是BCD码。

间接寻址DM字不存在(＊DM内容不是BCD的或超出DM区域)

5－14－3 累加定时器指令——TTIM(87)

梯形符号



定义符数值

N:TC编号
#(000～511)

操作数数据区域

SV:设定值(字,BCD码)
IR, AR, DM, HR, LR

RB:复位位
IR, SR, AR, HR, LR

限制

SV数值范围在0000～9999 (000.0～999.9秒)之间,数值必须是BCD码。小数点不输入。

每一TC编号只能用作一个定时器或计数器指令的定义符。

说明

TTIM (87),指令用于建立一个以0.1秒为单元增加当前值到0.1～999.9秒间某一时间的定时器。TTIM (87)指令从零开始,增值单位为0.1秒,精度为＋0.0/－0.1秒。只要执行条件为"ON",定时器定时,直到计时值达到设定值或定时器复位控制位RB为ON为止。定时器只是每个扫描周期执行条件符合时定时,在联锁程序段和跳转指令执行的程序中不定时,但保持当前值不变。

定时器 000～015 的当前值和结束标志每 10ms 刷新一次，但定时器 016～511 只在程序执行 TTIM(87) 指令时才刷新一次。

注意事项

当 IL(02)指令执行条件 OFF(联锁执行)，在联锁程序段中的累加定时器当前值保持不变，不象定时器和高速定时器，累加定时器在跳转程序段中保持当前值不再定时。

电源中断使定时器复位。

累加定时器定满时将不会重新起动，除非修改的当前值数值比设定值小或者定时器复位信号有效。

定时器计满时，定时结束标志有时会延时一个扫描周期再接通。

标志

ER (SR25503)：当设置为 BCD 时，(＊DM 的内容不是 BCD 码。)

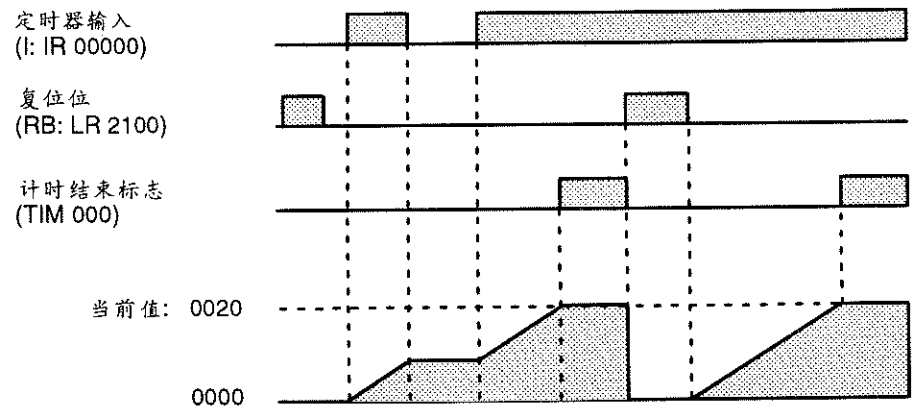
设定值不是 BCD 码。

例

下例说明一设定值为 2 秒的累加定时器执行条件，定时器当前值和定时结束标志间的相互关系。

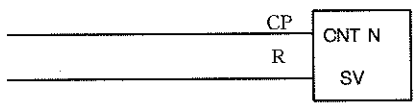


地址	指令	操作数
00000	LD	00000
00001	TTIM(87)	
		TIM 000
		# 0020
		LR 2100



5－14－4 计数器指令——CNT

梯形符号



定义符数值

N:TC 编号
#(000～511)

操作数数据区域

SV:(设定值)(字,BCD)
IR, AR, DM, HR, LR, #

限制

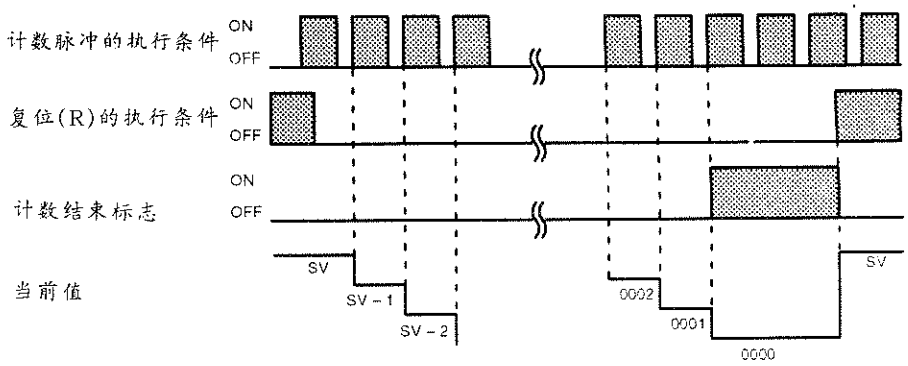
每一 TC 编号只能用作一条定时器或计数器指令的定义符。

说明

当计数脉冲 CP 的执行条件从 OFF 变成 ON，计数器就作减值计数，即，只要计数器 CP 执行条件为 ON，而上一扫描周期执行条件为 OFF，计数器作减一计数。如果 CP 端执行条件不变或由 ON 变为 OFF，计数器当前值不变。当计数器当前值计到零，计数器结束标志为 ON，并一直保持 ON 状态到计数器复位。

计数器的复位是由一复位输入 R 来实现的，当 R 由 OFF 变 ON，计数器当前值恢复为设定值。复位 R 为 ON 期间，计数器当前值不减值。当复位 R 变为 OFF 时，计数器从当前值开始递减，电源中断或处于联锁程序块内的计数器当前值不会复位。

接行条件的变化，结束标志和当前值的变化说明如下，当前值线高度只是说明当前值的变化情况。



注意事项

即使设定值不是 BCD 码，程序仍将继续执行，但设定值不准确。

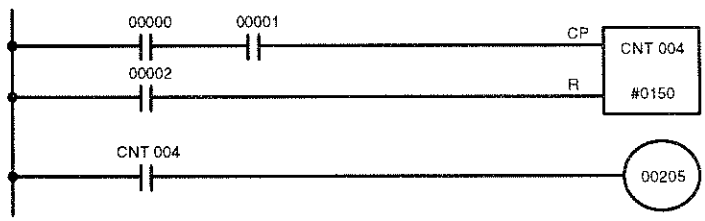
标志

ER：设定值不是 BCD 码。

间接寻址 DM 字不存在（＊DM 内容不是 BCD 码或超出 DM 区域）

例 1：基本应用

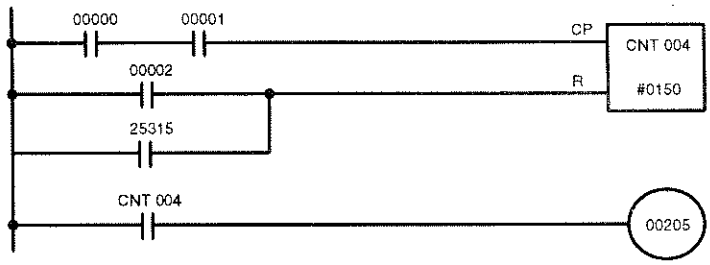
下例中，假定 00002 断开（OFF），且上次 CNT004 执行时 00000 或 00001 断开(OFF)，只要 00000 和 00001 同时为“ON”时 PV 将作减一计数。当 150 个脉冲计完(即当前值为零)，00205 接通。



地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	CNT	004
		# 0150
00004	LD	CNT 004
00005	OUT	00205

这里，00000 用于计数器计数控制，00001 用于由 OFF 变为 ON 计数信号位。

上面图计数器 CNT 可改成每次 PC 上电时，从 SV 值开始计数，可用 SR 区（25315）中的第一扫描周期标志位复位 CNT 的方法实现。梯形图如下图示。



地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	OR	25315
00004	CNT	004
		# 0150
00005	LD	CNT 004
00006	OUT	00205

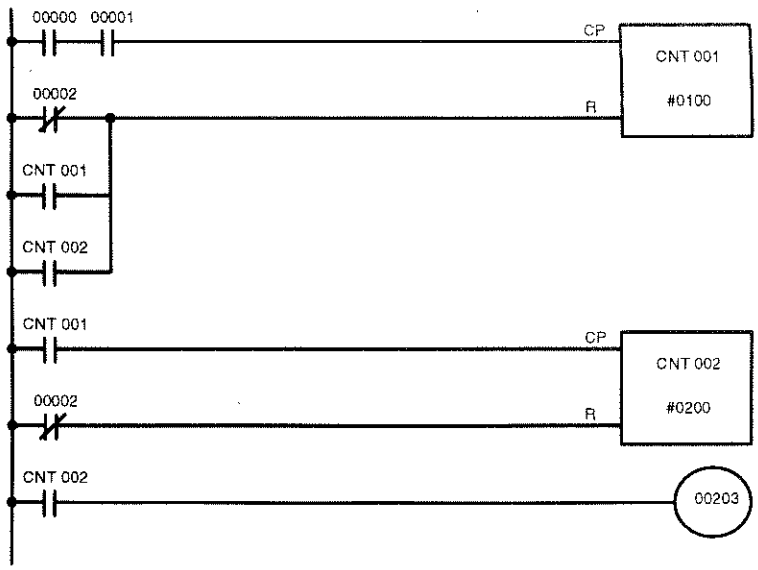
例 2:扩展计数器

计数值超过 9,999 的计数器编程可用一个计数器对另一个从设定值到零的动作次数进行计数的方法。

下例中,00000 用于控制 CNT001 计数器操作。当 00000 为 ON,计数器 001 对 00001 从 OFF 变为 ON 次数减值计数。CNT001 由自己的计数结束标志复位,即一旦计数器从当前值计到零就马上重新起动计数,计数器 002 对计数器 001 计数结束标志由 OFF 变为 ON 的次数进行计数。00002 位作为整个扩展计数器的复位信号,当 00002 断开(OFF),计数器 001 和 002 马上复位。当 CNT002SV 计数值满,计数器 CNT002 的计数结束标志也用作 CNT001 复位控制,在整个扩展计数器由 00002 复位之前,禁止 CNT001 操作。

因为例子中 CNT001 为设定值是 100,CNT002 设定值是 200,当 00001 中由 OFF→ON 变化次数达到 20000(100×200)次时,CNT002 计数结束标志 ON,使 00203 接通。

根据需要,可以将任意多个计数器级联使用以达到任何计数值。



地址	指令	操作数
00000	LD	00000
00001	AND	00001
00002	LD NOT	00002
00003	OR	CNT 001
00004	OR	CNT 002
00005	CNT	001
		# 0100
00006	LD	CNT 001
00007	LD NOT	00002
00008	CNT	002
		# 0200
00009	LD	CNT 002
00010	OUT	00203

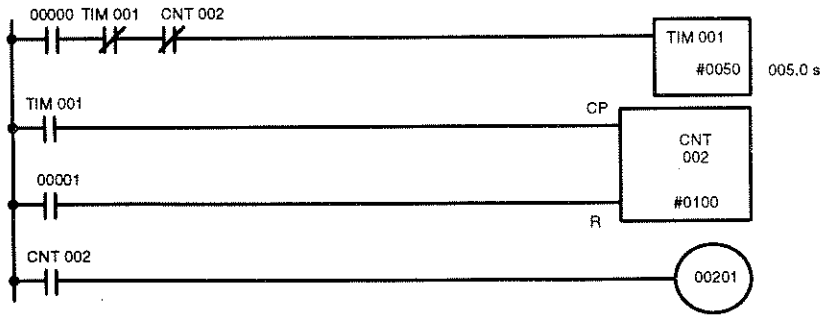
例 3:扩展时间的定时器

用计数器扩展定时范围有两种方法:计数器与定时器结合和采用 SR 区的时钟脉冲位信号计数。

下例中,CNT002 计数器对 TIM001 定时器从设定值到零的动作次数进行计数。TIM001 定时结束标志用作 TIM001 复位信号,这样,TIM001 可不断地定时并发定时结束信号。CNT002 对 TIM001 定时结束标志接通次数进行计数。每次当 TIM001 定时标志变 ON,并由其将 TIM001 复位时,CNT002 动作一次。CNT002 计数结束标志也对 TIM001 复位,00001 作为整个扩展定时器复位信号,所以 00001 将 CNT002 复位后,扩展定时器才能再次起动。

因为例子中 TIM001 定时器的设定值为 5.0 秒,CNT002 设定值为 100,当定时满 5 秒×100 次,

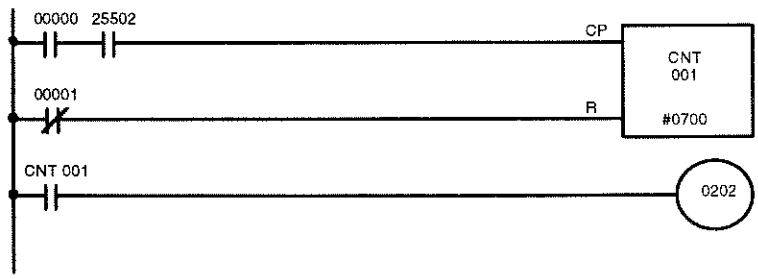
CNT002 计数结束标志接通(ON),即,500 秒(8 分 20 秒)定时结束。这一结果使 00201 接通。



地址	指令	操作数
00000	LD	00000
00001	AND NOT	TIM 001
00002	AND NOT	CNT 002
00003	TIM	001
		# 0050
00004	LD	TIM 001
00005	LD	00001
00006	CNT	002
		# 0100
00007	LD	CNT 002
00008	OUT	00201

在下例中,CNT001 对 1 秒时钟脉冲位(25502) 从 OFF 变 ON 次数进行计数。这里 00000 又用作计数器运行时的定时控制。

因为下列中 CNT001 的设定值是 700,在定时满 1 秒×700 次或 11 分 40 秒,计数器 CNT001 的结束标志接通 ON。这样也使得 00202 接通。



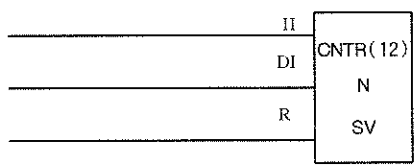
地址	指令	操作数
00000	LD	00000
00001	AND	25502
00002	LD NOT	00001
00003	CNT	001
		# 0700
00004	LD	CNT 001
00005	OUT	00202

注意

较窄的时钟脉冲不能产生精确的定时器。因为在长扫描周期内可能正好没读到脉冲的接通状态。严格地讲,0.02 秒和 0.1 秒时钟脉冲不能用于建立使用 CNT 指令的定时器。

5－14－5 可逆计数器指令——CNTR(12)

梯形符号



定义符数值

N:TC 编号
#(000～511)

操作数数据区域

SV:设定值(字,BCD)
IR, AR, DM, HR, LR, #

限制

每一个 TC 编号只能用作一个定时器或计数器的指令。

说明

CNTR (12)是可逆循环计数器,即,计数器根据两个执行条件:增值输入(II)和减值输入(DI),在零至设定值之间进行递增和递减计数。

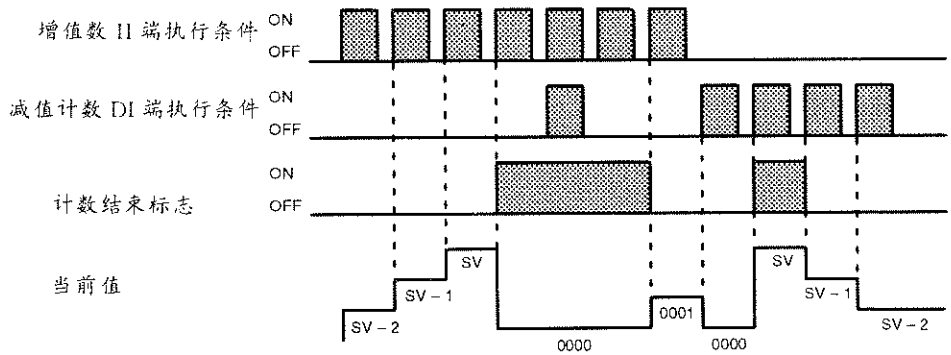
无论何时只要可逆计数器 II 端上一个扫描周期为 OFF,当前为 ON,执行指令,当前值(PV)加 1。无论何时只要 DI 端上的一个扫描周期为 OFF,当前为 ON,执行指令,当前值(PC)减一。如果从上一个扫描周期起 II 和 DI 都从 OFF 变为 ON,当前值不变。

如果 II 和 DI 执行条件不变,或者从 ON 变为 OFF,计数器当前值不变。

当从 0000 值减 1,当前值变成设定值,计数结束标志接通(ON)直到当前值再减一为止。当增值超过设定值,当前值置为 0000,标志位接通直到当前值再加 1 为止。

可逆计数器 CNTR(12)由输入复位位 R 复位,当 R 从 OFF 变 ON,当前值复位为零,在 R 为 ON 期间,当前值不增减。R 为 OFF,计数重新开始。电源中断或处于联锁程序段中,计数器不复位。

II 和 DI 执行条件,标志位和当前值变化说明见下例:计数器部分工作状况(即 CNTR(12)的 PV 值 CNTR(12)的 PV 值,复位,从零开始计数) 当前值顶线只是说明当前值变化情况。



注意事项

即使设定值不是 BCD 码,程序仍将继续运行,但设定值将不准确。

标志

ER:设定值(SV)不是 BCD 码。

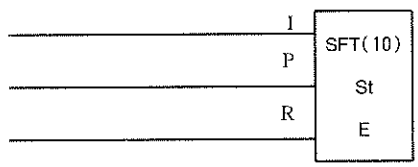
间接寻址 DM 字不存在(＊DM 字内容不是 BCD 码或超出 DM 区域。)

5－15 数据移位指令

这一节所描述的指令都用于数据移位,不同之处只是移位的量和方向。第一条移位指令是 SFT (10),它是将执行条件状态移入一个移位寄存器;其它移位指令则是对存贮器中的数据进行移位操作。

5－15－1 移位寄存器指令——SFT(10)

梯形符号



操作数数据区域

St:起始字
IR, SR, AR, HR, LR

E:结束字
IR, SR, AR, HR, LR

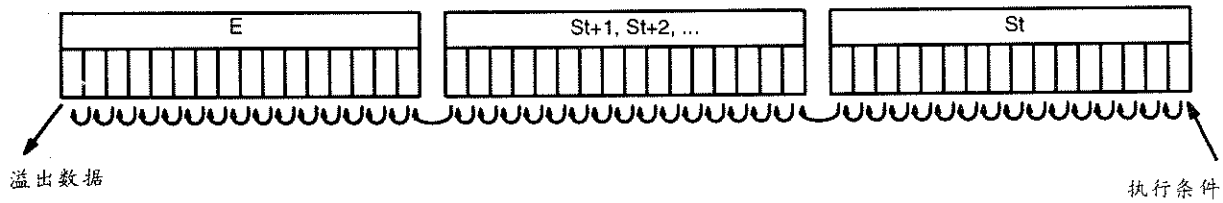
限制

St 值必须小于等于 E 值，且 St 和 E 必须在同一数据区。

如果某一字的位地址既用于移位寄存器又用于控制单独的位状态指令（如 OUT，KEEP（11）），一个出错信息：（线圈重复"Coil Dupl"）会在编程器或其它编程装置作程序语法检查时出现，但程序仍按程序写入的那样运行。见例 2：移位寄存器中的控制位（Bit）编程举例。

说明

SET（10）指令由三个执行条件 I、P 和 R 控制。如果 SFT（10）指令执行，且 1）P 端上次执行条件为 OFF，当前执行条件为 ON；2）R 端 OFF，则执行条件 I 移入 St 和 E 之间定义的移位寄存器的最低位。即，如果 I 接通（ON），一个 1 移入寄存器；如果 I 断开（OFF），一个 0 移入寄存器。当 I 状态被移入寄存器，以前在寄存器中所有位向左移，寄存器最左位（最高位）溢出丢失。



P 端功能执行条件象一条微分指令，即仅当 P 端上次 SFT（10）执行时为 OFF，当前 P 端为 ON，I 状态才移入寄存器，如 P 端执行条件不变，或者从 ON 变为 OFF 移位寄存器保持不变。

St 指定移位寄存器最低字；E 指定最高字。移位寄存器包括这两个字和两字之间的所有字。St 和 E 如指定同一字，则生成 16 位（即一个字）移位寄存器。

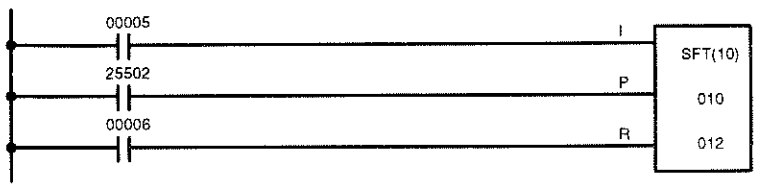
当 R 端执行条件变为 ON，移位寄存器所有位变为 OFF（即清零），移位寄存器在 R 再变为 OFF 前不工作。

标志

SFT(10)指令不影响标志位状态。

例 1：基本应用

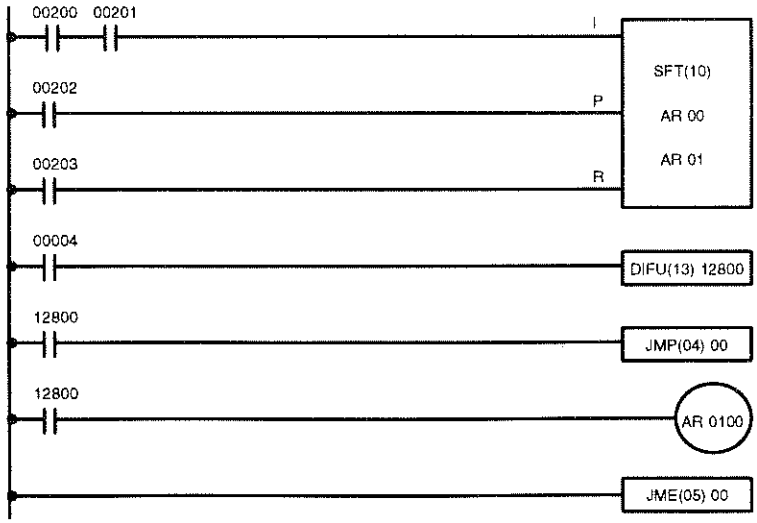
下例中采用 1 秒时钟脉冲位（25502），所以 IR010 和 IR012 间的三字移位寄存器每秒移入 00005 执行条件状态一次。



地址	指令	操作数
00000	LD	00005
00001	LD	25502
00001	LD	00006
00003	SFT(10)	
		010
		012

例 2：控制移位寄存器中的位

下面程序用于控制作用于 AR00－AR01 的移位寄存器第 17 位状态。在第 17 位需要设置时接通 00004。在那一扫描周期中，00004 接通使得 JMP（04）跳步不执行，AR0100（第 17 位）接通。当 12800 断开（OFF）（即除了在 00004 从 OFF 变为 ON 后的第一个扫描周期外所有时间），跳转执行，AR0100 状态不变。



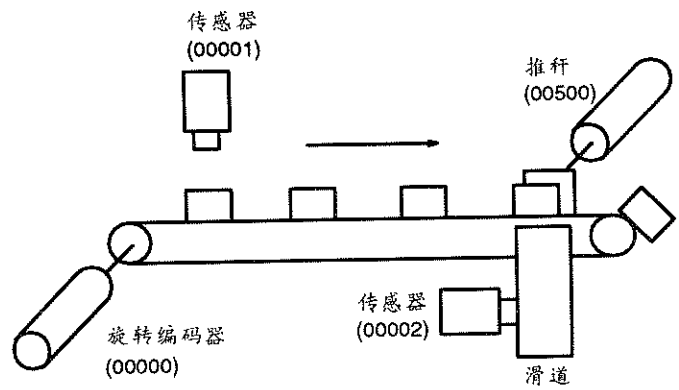
地址	指令	操作数
00000	LD	00200
00001	AND	00201
00002	LD	00202
00003	LD	00203
00004	SFT(10)	
		AR 00
		AR 01
00005	LD	00004
00006	DIFU(13)	12800
00007	LD	12800
00008	JMP(04)	00
00009	LD	12800
00010	OUT	AR 0100
00011	JME(05)	00

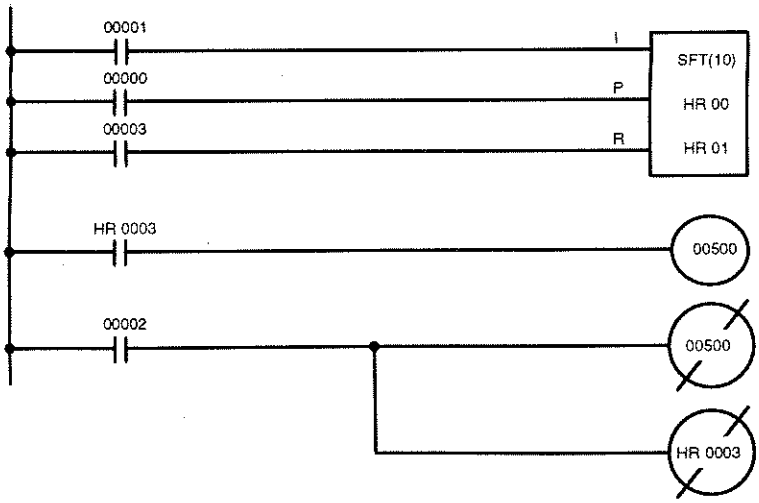
当移位寄存器中某一位被用于 OUT 指令(或其它位状态控制指令),程序检查时会出现语法出错字符,但程序仍可按写入的执行。

例 3:控制作用

下图程序控制一条传送线,如下图示,有缺陷的产品由传感器检测出并推下闩板,完成这一动作的条件由第一个传感器(00001)输入并存入移位寄存器:好的产品 00001 为 ON;坏的产品为 OFF。传送线的速度已调好,使得当有缺陷产品到达时,移位寄存器的 HR0003 驱动推杆(00500),即当 HR0003 变为 ON,00500 接通并驱动推杆。

如图编程,SFT (10)的执行由旋转编码器(00000)通过微分指令 DIFU (13)进行控制,每当一个产品通过第一个传感器时,旋转编码器就发出一个 ON 和 OFF 脉冲。另一传感器 (00002)用于检测在闩板上的不合格产品,输出推杆动作,并根据需要将 HR0003 复位。

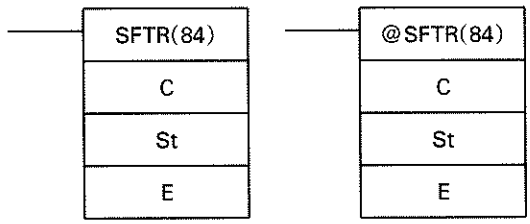




地址	指令	操作数
00000	LD	00001
00001	LD	00000
00002	LD	00003
00003	SFT(10)	
		HR 00
		HR 01
00004	LD	HR 0003
00005	OUT	00500
00006	LD	00002
00007	OUT NOT	00500
00008	OUT NOT	HR 0003

5－15－2 可逆移位寄存器指令——SFTR (84)

梯形符号



操作数数据区域

C:控制字
IR, AR, DM, HR, LR

St:起始字
IR, SR, AR, DM, HR, LR

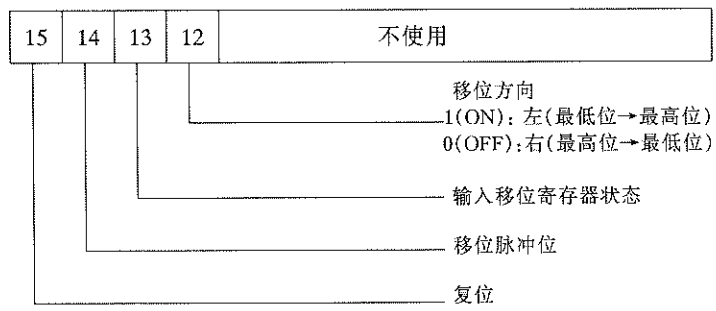
E:结束字
IR, SR, AR, DM, HR, LR

限制

St 和 E 必须在同一数据区，St 必须小于或等于 E。

说明

SFTR (84)指令用于生成一个可左、右移位的单字或多字移位寄存器。若 St，E 被赋于同一个字，则生成一个单字移位寄存器。控制字决定移位方向，输入移位寄存器的信号状态，移位脉冲和复位输入。控制字分配如下：



只要 SFTR (84) 指令执行条件满足，第 14 位为 ON，复位位为 OFF，按第 12 位指定的方向移位寄存器中数据移一位，移出位移入进位位 (CY)，第 13 位的状态移入移位寄存器为另一端。如果 SFTR (84) 执行

条件为 OFF 或者 SFTR (84) 执行时第 14 位为 OFF，移位寄存器将保持不变。如果 SFTR (84)执行条件为 ON，复位位(第 15 位)为 ON，整个移位寄存器所有位及 CY 位全部复零。

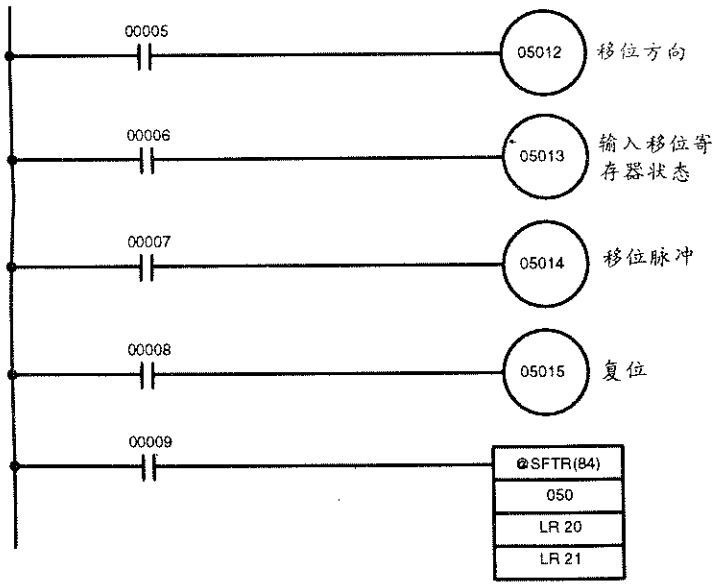
标志

ER：St 和 E 不在一个数据区或者 St 大于 E。间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

CY：根据移位方向，CY 接受 St 的第 00 位或 E 的第 15 位状态。

例

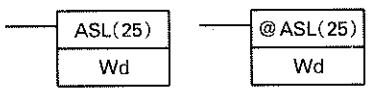
下例中，IR00005，IR00006，IR00007 和 IR00008 用以控制@SHIFT(84)指令中控制字 C 的各位。移位寄存器在 LR20 与 LR21 之间，指令由 IR00009 控制。



地址	指令	操作数
00000	LD	00005
00001	OUT	05012
00002	LD	00006
00003	OUT	05013
00004	LD	00007
00005	OUT	00514
00006	LD	00008
00007	OUT	05015
00008	LD	00009
00009	@SFT(10)	
		050
		LR 20
		LR 21

5－15－3 算术左移指令——ASL(25)

梯形标志

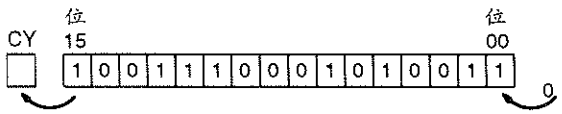


操作数数据区域

Wd：移位字
IR，SR，AR，DM，HR，LR

说明

当指令执行条件为 OFF 时，ASL (25)指令不执行。当执行条件为 ON 时，ASL(25) 将“0”移入字(Wd)的 00 位，字位依次左移一位，第 15 位状态移入 CY 位。



标志

ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

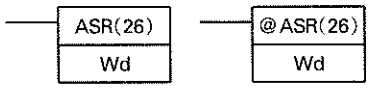
CY：接受第 15 位状态。

EQ：当字内容是零，EQ 为 ON，反之 EQ 为 OFF。

N：当“1”移入字的 15 位，N 为 ON。

5－15－4 算术右移指令－ASR(26)

梯形符号

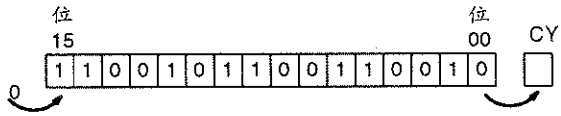


操作数数据区

Wd:移位字
IR, SR, AR, DM, HR, LR

说明

指令执行条件为 OFF 时,ASR(26)指令不执行。执行条件为 ON 时,ASR(26)将“0”移入字的第 15 字,字位依次右移一位。00 位状态移入 CY 位。



标志

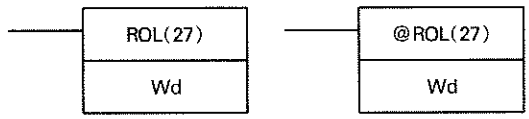
ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码,或超出 DM 区域)

CY:接受 00 位数据。

EQ:当字内容是零,EQ 为 ON;反之 EQ 为 OFF。

5－15－5 循环左移指令－ROL(27)

梯形符号

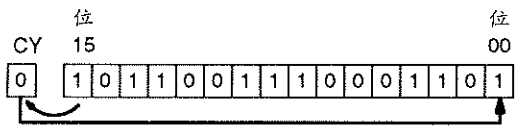


操作数据区

Wd:循环字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF 时,ROL(27)指令不执行,当执行条件为 ON 时,ROL(27)将字中各位左移一位,CY 移入 00 位,字第 15 位状态移入 CY。



注意事项

在执行循环操作前为保证执行 ROL(27)指令前 CY 含有正确状态,使用 STC(41)使 CY 置位或用 CLC(41)清除 CY 的状态。

进位标志 CY 状态在每个循环周期结束时(当执行 END(01)指令时)清“0”。

标志

ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 或超出 DM 区域)

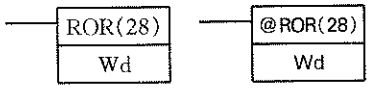
CY:接受字中第 15 位数据。

EQ:字内容是零,EQ 为 ON;反之为 OFF。

N:当“1”移入字的 15 位,N 为 ON。

5－15－6 循环右移指令－ROR(28)

梯形符号

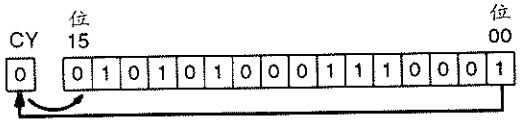


操作数数据区域

Wd：循环字
IR，SR，AR，DM，HR，LR

说明

当指令执行条件为 OFF 时，ROR(28)指令不执行。当执行条件为 ON 时，ROR(28)指令将字中各位右移一位，CY 状态移入字的 15 位，00 位状态移入 CY。



注意事项

在执行循环操作之前，为保证执行 ROR(28)指令前 CY 含有正确的状态，可用 STC(40)将 CY 置“1”，或用 CLC(41)将 CY 清“0”。

进位位标志 CY 状态在每个循环周期结束时(当执行 END(01)时) 清“0”。

标志

ER：间接寻址 DM 字不存在。(* DM 字内容不是 BCD 码或超出 DM 区域)

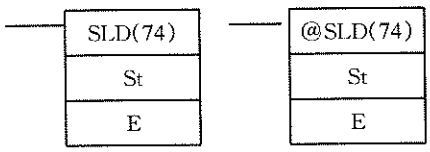
CY：接受字第 00 位数据。

EQ：字内容是零，EQ 为 ON，反之为 OFF。

N：当“1”移入字的 15 位，N 为 ON。

5－15－7 数(4 bits)左移——SLD(74)

梯形符号



操作数数据区域

St：起始字
IR，SR，AR，DM，HR，LR

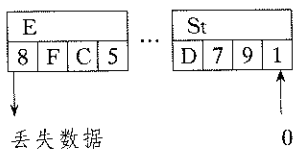
E：结束字
IR，SR，AR，DM，HR，LR

限制

St 和 E 必须在同一数据区，St 必须小于等于 E。

说明

当执行条件为 OFF 时，SLD(74)不执行，当执行条件为 ON 时，SLD(74)将包括 St 和 E 及它们之间的数(4 位二进制)左移。“0”写入 St 的最低位数，E 的最高位数丢失。



注意事项

若在超过 50 个字的移位执行过程中发生电源故障，移位操作可能不能完成。

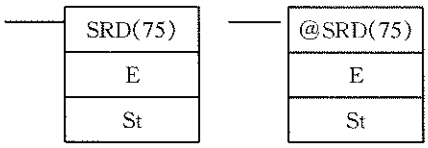
标志

ER：St 和 E 不在同一数据区或者 St 大于 E。

间接寻址 DM 字不存在。(* DM 字内容不是 BCD 码或超出 DM 区域)

5－15－8 数(4 bits)右移指令——SRD(75)

梯形符合



操作数数据区域

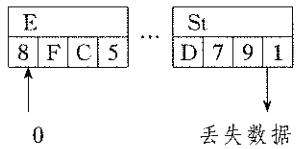
E:结束字
IR, SR, AR, DM, HR, LR
St:起始字
IR, SR, AR, DM, HR, LR

限制

St 和 E 必须在同一数据区，St 必须不于等于 E。

说明

当指令执行条件为 OFF 时，SRD(75)不执行。当执行条件为 ON 时，SRD(75)将包括 St 和 E 及它们之间的数(4 位二进制)右移。“0”写入 St 最高位数，E 的最低位数丢失。



注意事项

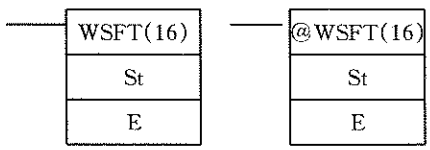
超过 50 个字的移位执行过程中若发生电源故障，移位操作可能不能完成。设置 E 和 St 之间字最多为 50 个。

标志

ER：St 和 E 不在同一数据区或者 St 大于 E。间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

5－15－9 字移位指令——WSFT(16)

梯形符号



操作数数据区域

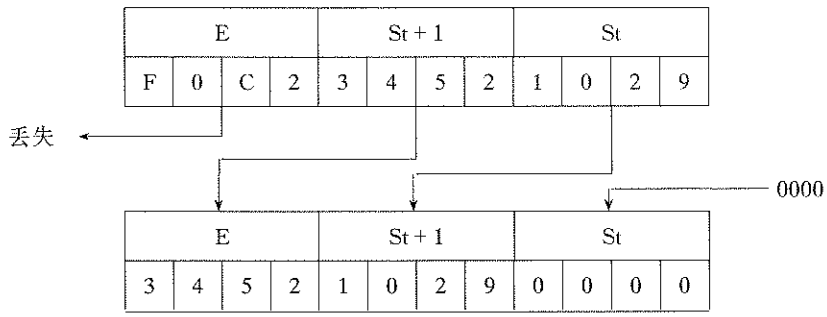
St:起始字
IR, SR, AR, DM, HR, LR
E:结束字
IR, SR, AR, DM, HR, LR

限制

St 和 E 必须在同一区域中，St 必须小于等于 E。

说明

当指令执行条件为 OFF 时，WSFT(16)不执行。当执行条件为 ON 时，WSFT(16)指令将 St 和 E 间的数据按字左移。“0”写入 St，E 内容丢失。



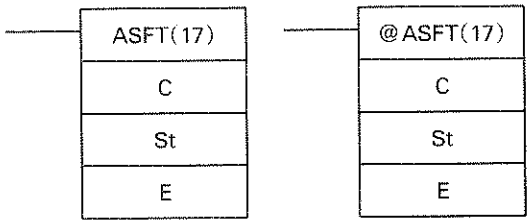
标志

ER：St 和 E 不在同一区域或 St 大于 E。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

5－15－10 异步移位寄存器指令——ASFT(17)

梯形符号



操作数数据区域

C：控制字
IR，SR，AR，DM，HR，LR

St：起始字
IR，SR，AR，DM，HR，LR

E：结束字
IR，SR，AR，DM，HR，LR

限制

St 和 E 必须在同一区域且 St 小于等于 E。

说明

当指令执行条件为 OFF 时，ASFT(17)不做任何事，程序执行下一条指令。当执行条件为 ON 时，ASFT(17)用于生成和控制一个在 St 和 E 之间字的异步可逆移位寄存器。寄存器只有在寄存器中的下一字是零的情况下才移位。例如，如果寄存器中没有一个字为零，就不进行移位操作。同样，对寄存器中含有另的每个字，只移位一个字。当字的内容移到下一字，移掉内容的原先字填入零。实质上，当寄存器移位，寄存器中的零与下一个字交换（见下例）

移位方向（即“下一个字”是高位或低位字）由控制字 C 指定。C 也用于寄存器复位。用 St 和 E 指定所需的部分，就可对寄存器所有的部分进行复位。

控制字

该控制字 C 的 00 位到 12 位没有使用。第 13 位是移位方向位：13 位为 ON，向下移（朝低地址字的方向），13 位为 OFF，向上移（朝高地址字的方向）。14 位移位允许位：14 位为 ON，允许移位寄存器根据 13 位移位方向操作。14 位为 OFF，寄存器不允许移位。15 位是复位位：当 ASFT(17)执行且 15 位为 ON，寄存器从 St 到 E 之间字复位。15 位 OFF 用于正常操作。

控　制　字	功　能
＃4000	向上移（朝高地址字的方向）
＃6000	向下移（朝低地址字的方向）
＃8000	St 至 E 之间字清零

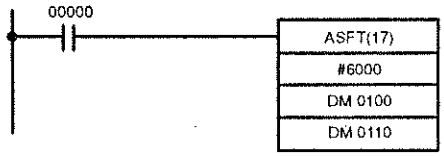
标志

ER：St 和 E 在不同区域或 St 大于 E。

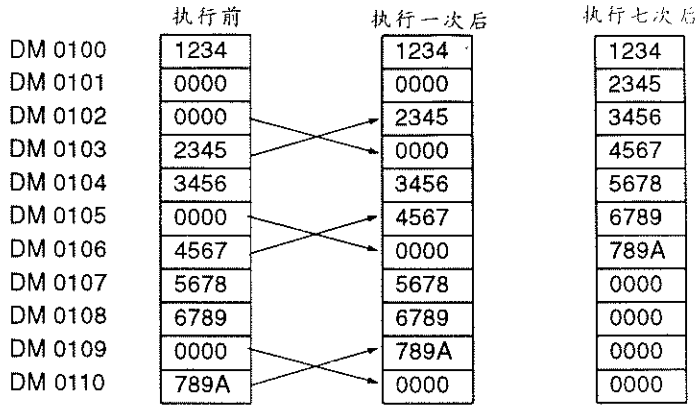
间接寻址字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

例子

下例表明 ASFT(17)用于移位由 DM0100 至 DM0110 给定的 11 字移位寄存器，控制字为＃6000（13 位和 14 位为 ON）。给定移位寄存器数据变化和控制字内容见下面。



地址	指令	操作数
00100	LD	00000
00101	ASFT(17)	
		# 6000
		DM 0100
		DM 0110

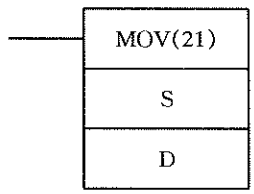


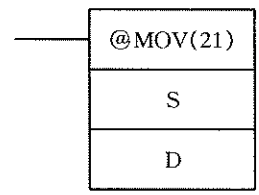
5－16 数据传送指令

这节叙述的指令用于在数据区中不同地址之间传送数据。数据传送可以由编程决定在同一数据区或在不同数据区进行。数据传送对于 PC 所有数据区的使用非常重要，链接系统的有效通信也需要数据传送。所有这些指令只是传送数据字的目的地址的内容变化，即，在任一数据传送指令执行前后，源字的内容不变。

5－16－1 传送指令——MOV(21)

梯形符号





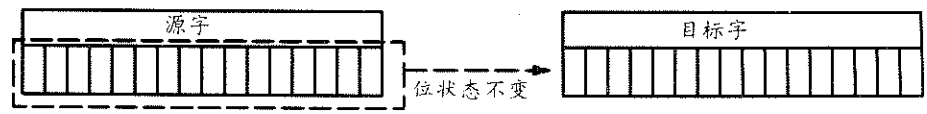
操作数数据区域

S：源字
IR, SR, AR, DM, HR, TC, LR, #

D：目标字
IR, SR, AR, DM, HR, LR

说明

当执行条件为 OFF，MOV(21)不执行。当执行条件为 ON，MOV(21)将 S 内容拷贝到 D。



注意事项

TC 编号不能指定为 D 以改变定时器或计数器当前值。要改变定时器或计数器当前值可使用 BSET

(71)指令。

标志

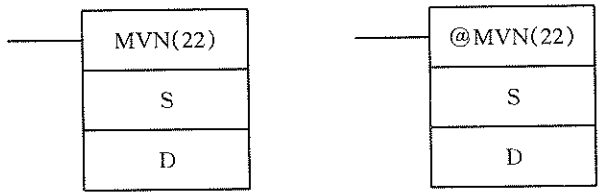
ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ：全零状态传送到 D 时，EQ 为 ON。

N　：当 D 的 15 位置“1”，N 为 ON。

5－16－2　取反传送指令——MVN(22)

梯形符号

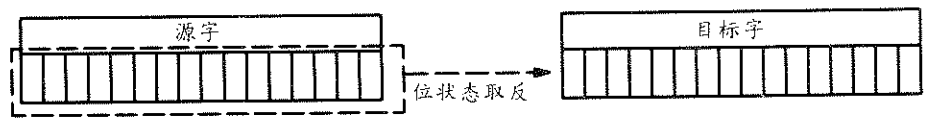


操作数数据区域

S：源字
IR，SR，AR，DM，HR，TC，LR，＃
D：目标字
IR，SR，AR，DM，HR，LR

说明

指令执行条件 OFF，MVN(22)不执行。当执行条件为 ON，MVN(22)将 S 内容(指定字或 4 位十六进制常数)取反送到 D 中，即 S 中所有 ON 状态位对应传送到 D 中为 OFF 状态，S 中所有 OFF 状态位对应传送到 D 中为 ON 状态。



注意事项

不能将 TC 编号定义 D 用以改变定时器或计数器当前值，改变当前值可采用 BSET(71)指令。

标志

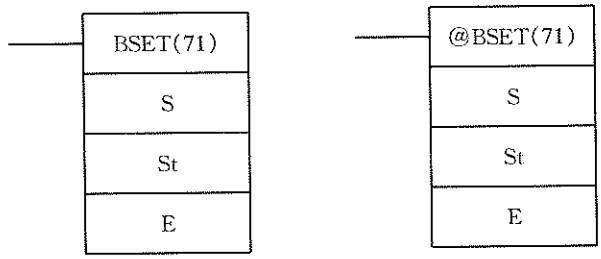
ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ：全零状态传送到 D 时，EQ 为 ON。

N　：当 D 的 15 位置“1”，N 为 ON。

5－16－3　块设置指令——BSET(71)

梯形符号



操作数数据区域

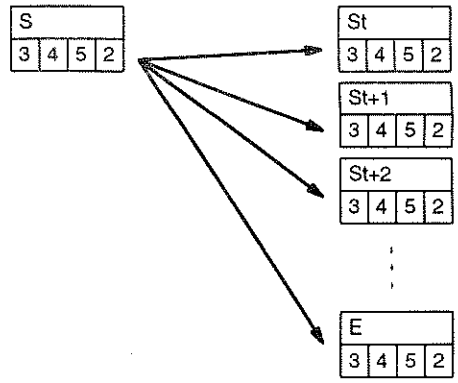
S：源数据
IR，SR，AR，DM，HR，TC，LR，＃
St：起始字
IR，SR，AR，DM，HR，TC，LR
E：结束字
IR，SR，AR，DM，HR，TC，LR

限制

St 必须小于等于 E 且 St 和 E 必须在同一数据区。

说明

当指令执行条件 OFF，BSET(71)指令不执行，当执行条件为 ON，BSET(71)将 S 的内容传送到从 St 到 E 所有字中去。



BSET(71)指令可用以改变定时器/计数器当前值(改变当前值不能用 MOV(21)或 MVN(22)指令)。BSET(71)也可用于数据区即 DM 区域部分段清零，为执行其它指令作准备。

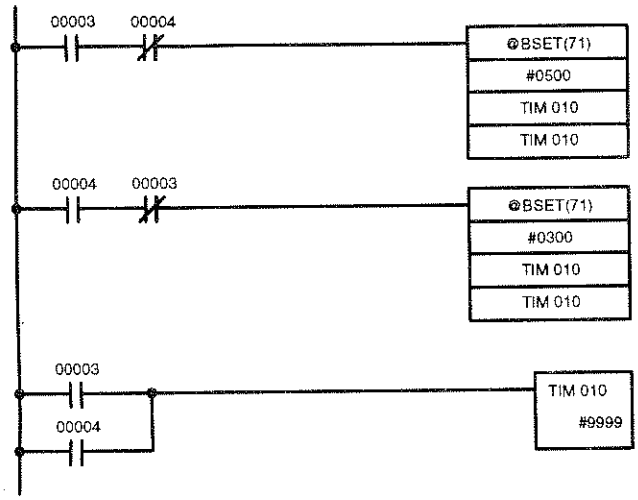
标志

ER：St 和 E 不在同一区域或 St 大于 E。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

例子

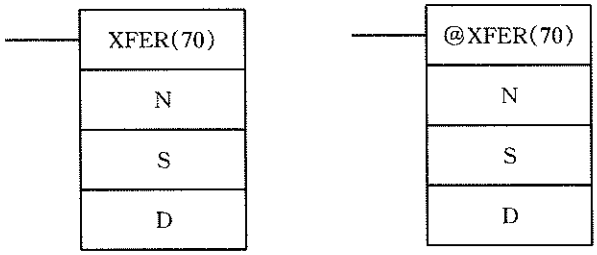
下例说明根据 IR00003 和 IR00004 状态如何使用 BSET(71)改变定时器当前值。当 IR00003 接通，定时器 TIM010 作为一个 50 秒定时器运行；当 IR00004 接通，定时器 010 作为一个 30 秒定时器运行。



地址	指令	操作数
00000	LD	00003
00001	AND NOT	00004
00002	@BSET(71)	
		# 0500
		TIM 010
		TIM 010
00003	LD	00004
00004	AND NOT	00003
00005	@BSET(71)	
		# 0300
		TIM 010
		TIM 010
00006	LD	00003
00007	OR	00004
00008	TIM	010
		# 9999

5－16－4 块传送指令——XFER(70)

梯形符号



操作数数据区域

N:字数(BCD码)
IR, SR, AR, DM, HR, TC, LR, #
S:源起始字
IR, SR, AR, DM, HR, TC, LR
D:目标起始字
IR, SR, AR, DM, HR, TC, LR

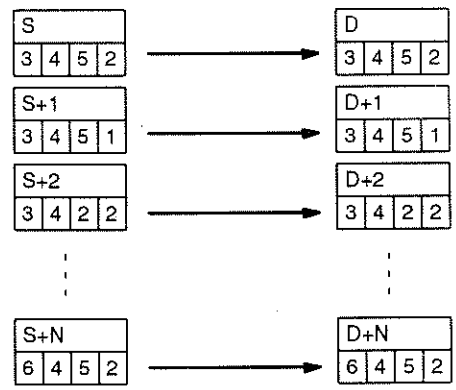
限制

S和D可在同一数据区域,但各自块区域一定不能重叠。

S和S+N及D和D+N必须在同一数据区,N必须是0000～6144间的BCD码。

说明

当指令执行条件OFF,XFET(70)指令不执行。当执行条件为ON,XFER(70)拷贝S,S+1,……,S+N内容到D,D+1,……,D+N中。



标志

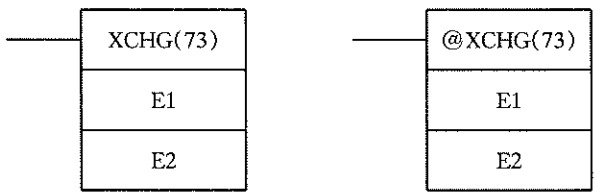
ER:N不是0000～2000间的BCD码。

S和S+N或D和D+N不在同一数据区。

间接寻址DM字不存在。(*DM字内容不是BCD码或超出DM区域)

5－16－5 数据交换指令——XCHG(73)

梯形符号



定义符数值

E1:交换字1
IR, SR, AR, DM, HR, TC, LR
E2:交换字2
IR, SR, AR, DM, HR, TC, LR

说明

当指令执行条件为OFF,XCHG(73)指令不执行。当执行条件为ON,XCHG(73)交换E_1和E_2内容。



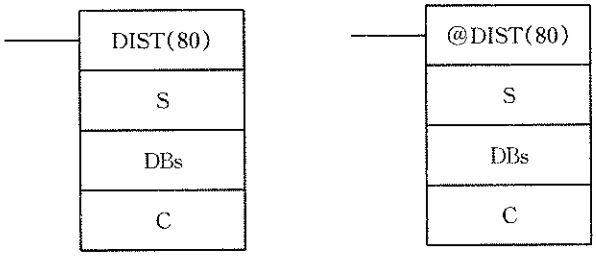
如果要交换超过一个字的块内容，可执行三次 XFER(70)块传送指令，用工作字作为一个中间缓冲来保留其中一个数据块。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

5－16－6 单字数据分配指令——DIST(80)

梯形符号



操作数数据区域

S：源数据
IR, SR, AR, DM, HR, TC, LR, #
DBs：目标基准字
IR, AR, DM, HR, TC, LR
C：控制字(BCD)
IR, SR, AR, DM, HR, TC, LR, #

限制

控制字 C 必须是 BCD 码。如果 C≤6655，DBs 和 DBs＋C 必须在同一数据区。如果 C≥9000，DBs 和 DBs＋C－9000 必须在同一数据区。

说明

取决于 C 的数值，DIST(80)可用作数据分配指令或堆栈指令。如果 C 值在 0000～6655 之间，DIST(80)可用作数据配指令，将 S 中内容传送到 DBs＋C 中去。如果 C 的最高位数是 9，DEST(80)用作堆栈指令并生成一个 C 低 3 位定义编号的堆栈。

注意事项

如果定义的堆栈长度与上一次执行 DIST(80)或 COLL(81)时定义长度不一样，堆栈操作可能会出错。

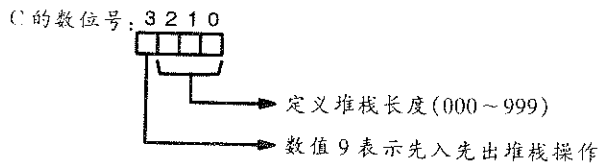
数据分配操作(C＝0000～6655)

当指令执行条件为 OFF，DIST(80)指令不执行。执行条件为 ON，DIST(00)将 S 内容传送到 DBs＋C 中去，即，C 加上 DBs 确定目标字。



堆栈操作(C＝9000～9999)

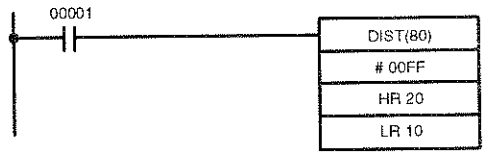
当指令执行条件为 OFF，DIST(80)指令不执行。当执行条件为 ON，DIST(80)操作 DBs 到 DBs＋C—9000 堆栈。DBs 堆栈指针，所以 S 内容拷贝到 DBs 指定的字中，指针 DBs 加 1，还改变负标志。



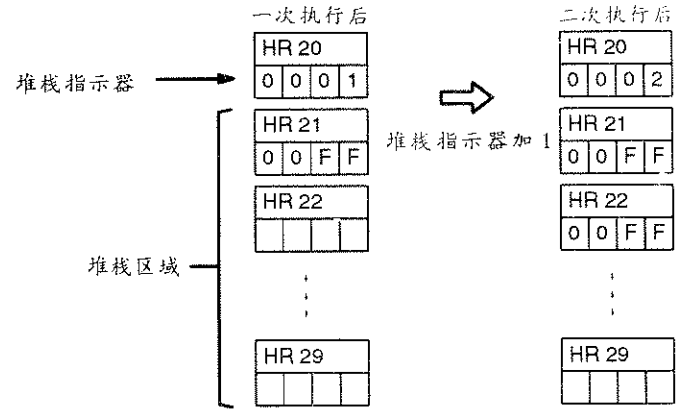
在堆栈区未满前数据可以进栈。DIST(80)(入栈)指令通常与 COLL(81)出栈指令一起用，它可被设定先入先出(FIFO)或后入先出(LIFO)的读栈方式。详见 5－16－7“数据调用指令—COLL(81)”章节。

堆栈操作例子

下例中，C(LR10)的内容是 9010，DIST(80)用于写数字数据＃00FF 到 HR20 到 HR29 的 10 字长的堆栈中。在 IR00001 为 ON 第一个扫描周期中，数据写入 DBs＋1(HR21)，堆栈指针加 1、第二个扫描周期时数据写入 DBs＋2(HR22)，堆栈指针加 1，依次类推。



地址	指令	操作数
00000	LD	00001
00001	DIST(80)	
		# 00FF
		HR 20
		LR 10



标志

ER：C 的内容不是 BCD 码或者 6655＜C＜9000。

当 C≤6655，DBs 和 DBs＋C 不在同一数据区。

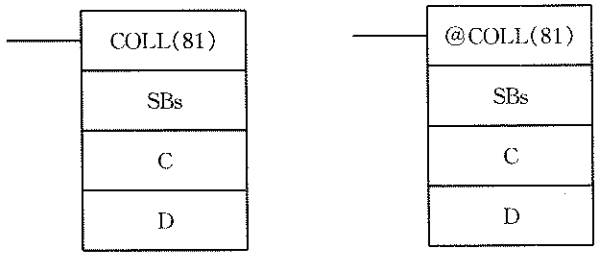
当 C≥9000，DBs 和 DBs＋C－9000 不在同一数据区。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当 S 内容为零，EQ 为 ON；反之 EQ 为 OFF。

5－16－7 数据调用指令——COLL(81)

梯形符号



操作数数据区域

SBs：源基准字
IR，SR，AR，DM，HR，TC，LR
C：偏移数据(BCD)
IR，SR，AR，DM，HR，TC，LR，#
D：目标字
IR，SR，AR，DM，HR，TC，LR

限制

C 必须是 BCD 码。如果 C≤6655，SBs 与 SBs＋C 必须在同一数据区内。如果 C 最高位数是 8 或 9，SBs 与 SBs＋N(N 等于 C 的低 3 位数)必须在同一数据区内。

说明

根据 C 的数值，COLL(81)可作为取数据指令(出栈)先入先出或后入先出堆栈指令使用。如果 C 值在 0000～6655 间，COLL(81)用作取数据指令，将 SBs＋C 内容传送到 D。

如果 C 的最高位数是 9，COLL(81)用作先入先出的出栈指令。如果 C 的最高位数是 8，COLL(81)用作

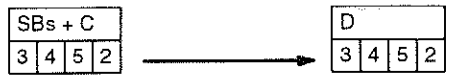
后入先出的出栈指令。两种堆栈指令使用的堆栈都从 SBs 开始，栈的长度由 C 的低 3 位数值定义。

注意事项

如果定义栈的长度与上一次执行 DIST(80)或 COLL(81)时定义栈长不同，堆栈操作可能会不可靠。

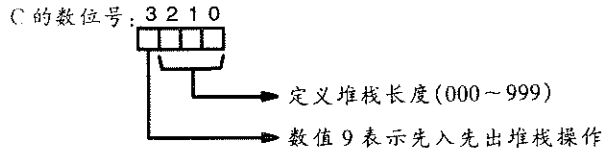
数据调用操作(C＝0000～6655)

当指令执行条件为 OFF，COLL(81)不执行。执行条件为 ON，COLL(81)传送 SBs＋C 的内容到 D，即 C 加上 SBs 以确定源字。



先入先出堆栈操作(C＝9000～9999)

当指令执行条件为 OFF，COLL(81)指令不执行。当执行条件为 ON，COLL(81)传送最早存入栈内的数据到 D，堆栈指针 SBs 减 1。

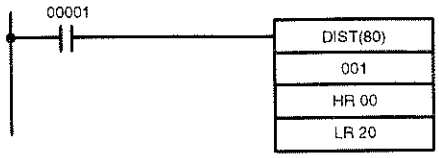


COLL(81)可与 DIST(80)指令一起使用。详见 5－16－6“单字数据分配指令——DIST(80)”章节。

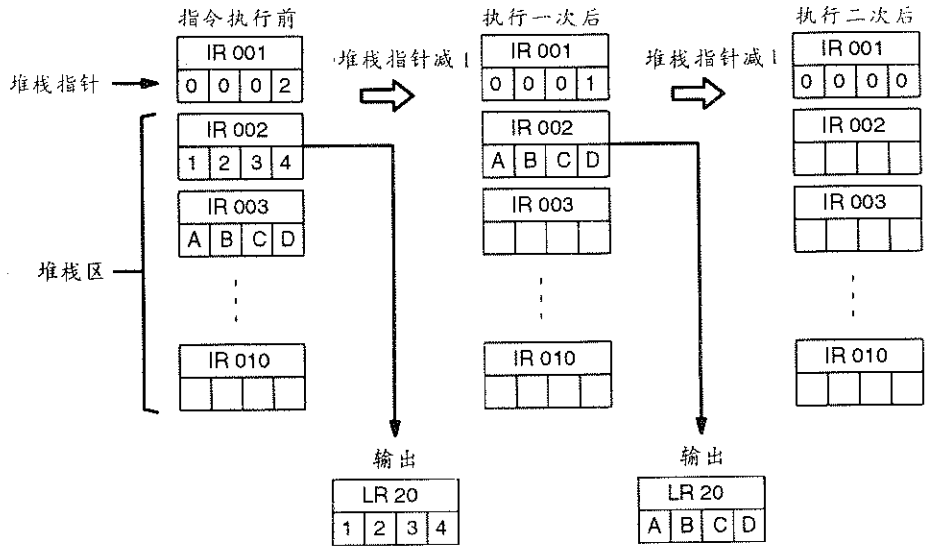
注 FIFO 表示先入先出。

例

下例中，C(HR00)内容是 9010，COLL(81)指令将最先存入 10 字长的堆栈(IR001～IR010)的内容传送到 LR20 中。

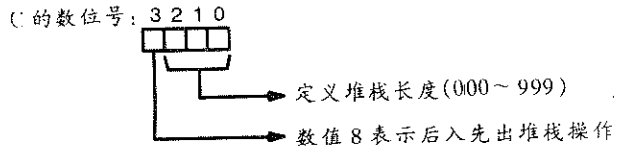


地址	指令	操作数
00000	LD	00001
00001	COLL(81)	
		001
		HR 00
		LR 20



后入先出栈操作(C＝8000～8999)

当指令执行条件为 OFF，COLL(81)指令不执行，执行条件为 ON，COLL(81)传送最近存入栈的内容到 D，堆栈指针 SBs 减 1。



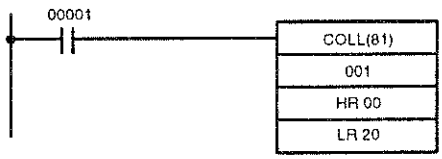
堆栈尚未存满前数据可加入栈中。DIST(80)的入栈操作可与 COLL(81)的出栈操作一起使用。COLL(81)可设置成先入先出或后入先出出栈方式。

详见 5－16－6 指令单数据分配指令章节。

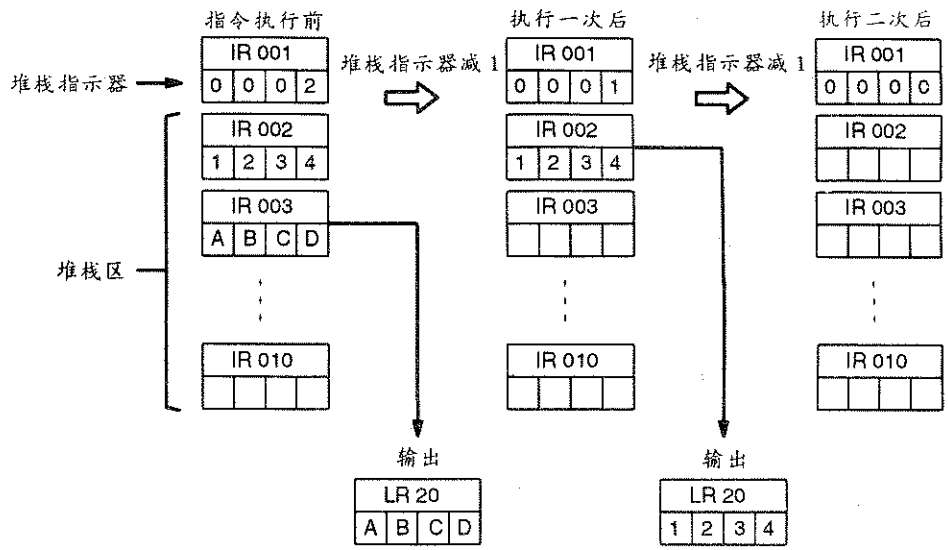
注 LIFO 表示后入先出。

例

下例中，C(HR00)内容是 8010，COLL(81)指令传送最近存入 10 个字长的堆栈内的容到 LR20。



地址	指令	操作数
00000	LD	00001
00001	COLL(81)	
		001
		HR 00
		LR 20



标志

ER：C 的内容不是 BCD 码或者 6655＜C＜8000。

当 C≤6655，SBs 和 SBs＋C 不在同一数据区。

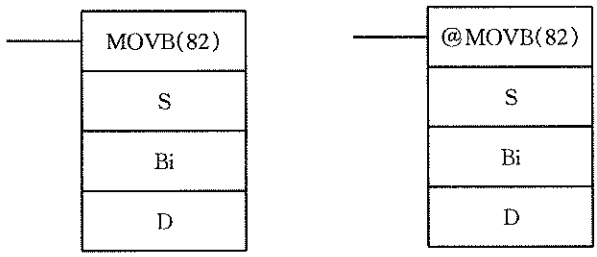
当 C≥8000，堆栈的顶和底不在同一数据区或堆栈指针值超出堆栈长度。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:传输数据是零,EQ 为 ON;反之 EQ 为 OFF。

5－16－8 位(bit)传送指令－MOVB(82)

梯形符号



操作数数据区域

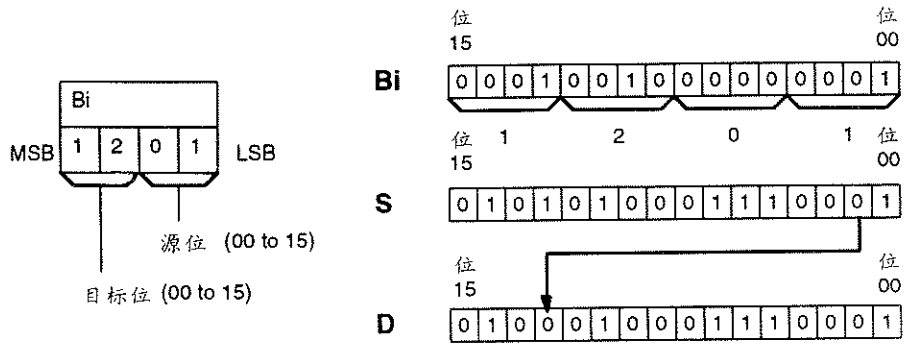
S:源字
IR, SR, AR, DM, HR, LR, #
Bi:位指示器(BCD)
IR, SR, AR, DM, HR, TC, LR, #
D:目标字
IR, SR, AR, DM, HR, LR

限制

Bi 的最低两位和最高两位数都必须在 00 和 15 之间。

说明

当指令执行条件为 OFF, MOVB(82)指令不执行。当执行条件为 ON, MOVB(82)传送 S 指定的位到 D 指定的位。S 和 D 中的位由 Bi 指定。Bi 低 2 位数指定源位;Bi 高 2 位数指定目标位。



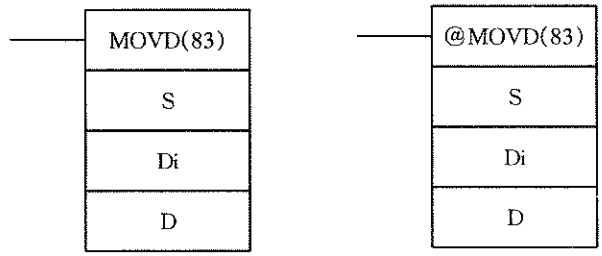
标志

ER:位指示器 C 不是 BCD 码或指定的位不存在。(即,位定义值必须在 00～15 之间。)

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

5－16－9 数(4bits)传送指令－MOVD(83)

梯形符号



操作数数据区域

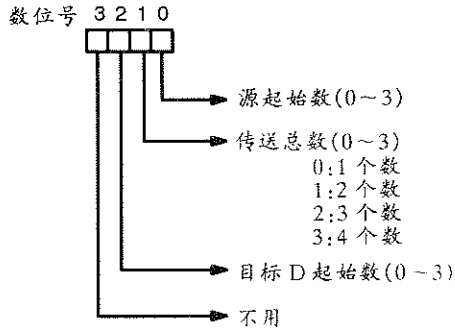
S:源字
IR, SR, AR, DM, HR, TC, LR, #
Bi:位指示器(BCD)
IR, SR, AR, DM, HR, TC, LR, #
D:目标字
IR, SR, AR, DM, HR, TC, LR

限制

Di 最低 3 位数都必须在 0～3 之间。

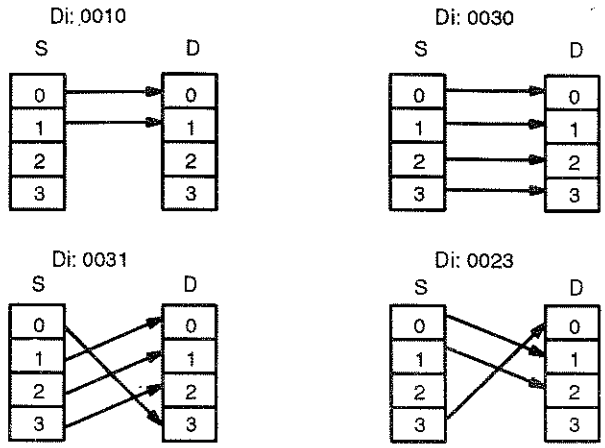
说明

当指令执行条件为 OFF，MOVD(83)指令不执行。当执行条件为 ON，MOVD(83)传送 S 中指定数(4bits)的内容到 D 中指定数(4bits)，每次最多传送 4 个数(4×4bits)。首传送数，传送总数和首接收数在 Di 中指定如下，S 中数连续传送到 D，其起始数位和传送数由 Di 定义。如果已经是 S 或 D 的最后数，下一位则回复到 0 位数开始。



位指示器

下图说明各种 Di 值的数据传送的例子。



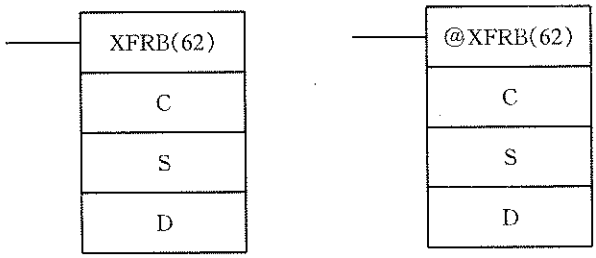
标志

ER:Di 最低 3 位数中至少有一位数内容不在 0～3 之间。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

5－16－10 多位传送指令——XFRB(62)

梯形符号



操作数数据区域

C:控制字
IR, SR, AR, DM, HR, TC, LR, #
S:源起始字
IR, SR, AR, DM, HR, TC, LR
D:目标起始字
IR, SR, AR, DM, HR, LR

限制

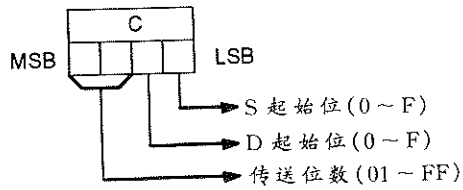
所有指定的源位必须在同一数据区内。

所有指定的目标位必须在同一数据区内。

说明

当指令执行条件为 OFF，XFRB(62)指令不执行。当执行条件为 ON，XFRB(62)指令传送指定的源位到指定的目标位中去。

用 C 的最低两位数定义源 S 和目标 D 的起始位(bit)；用 C 的最高 2 位数表示将传送的位数。

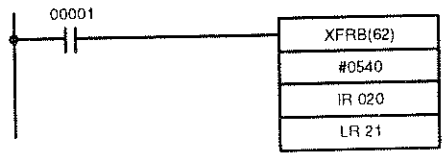


注

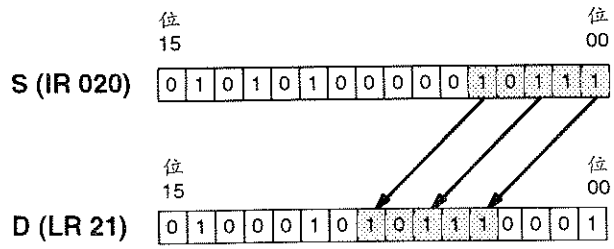
一次传送不能超过 255(FF)位。

例

下例中，当 IR00001 为 ON，XFRB(62)用于传送 IR020 中 5 位(bits)到 LR21。IR20 起始位是 0，LR21 起始位是 4，所以 IR02000～IR02004 状态传送到 LR2104～LR2108。



地址	指令	操作数
00000	LD	00001
00001	XFRB(62)	
		# 0540
		020
		LR 21



标志

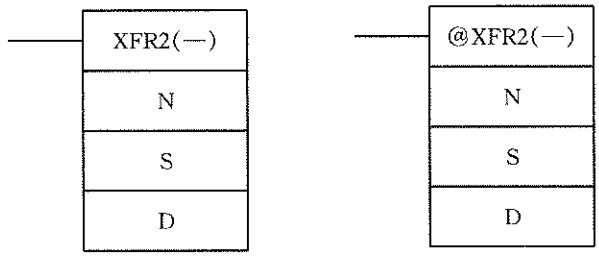
ER：所有指定的源位不都在同一数据区。

所有指定的目标位不都在同一数据区。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

5－16－11 EM 块传送指令——XFR2(—)

梯形符号



操作数数据区域

N:字数(BCD)
IR, SR, AR, DM, HR, TC, LR, #

S:源起始字
IR, SR, AR, DM, HR, TC, LR, #

D:目标起始字
IR, SR, AR, DM, HR, TC, LR, #

限制

S 和 S+N 必须在同一数据区，D 和 D+N 也必须在同一数据区。

N 必须是 BCD 码。

说明

当指令执行条件为 OFF，XFR2(—)指令不执行。当执行条件为 ON，XFR2(—)传送 S，S+1，…，S+N 中内容到 D，D+1，…，D+N 中。如果 S 或 D 为常数，该常数指定当前 EM 组中的地址。

标志

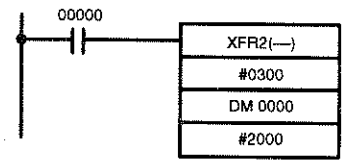
ER：N 不是 BCD 码。

S 和 S+N 或 D 和 D+N 不在同一数据区内。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

例

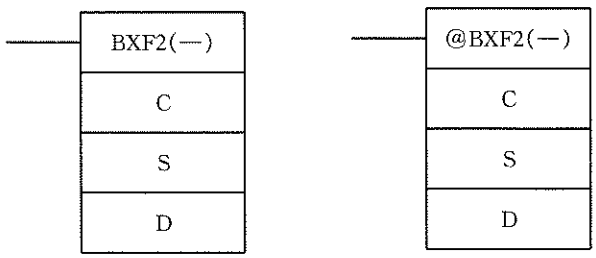
下例传送 DM0000 至 DM0299 间 300 字的内容到当前 EM 组中 EM2000 至 EM2299。



地址	指令	操作数
00200	LD	00000
00201	XFR2(—)	
		# 0300
		DM 0000
		# 2000

5－16－12 EM 组传送指令——BXF2(—)

梯形符号



操作数数据区域

C:第一控制字
IR, SR, AR, DM, HR, TC, LR, #

S:源起始字
IR, SR, AR, DM, HR, TC, LR, #

D:目标起始字
IR, SR, AR, DM, HR, TC, LR, #

限制

C＋1 字的值必须是 1～6144 间的 BCD 数。

S 和 S＋N 必须在同一数据区，D 和 D＋N 也必须在同一数据区。

说明

当指令执行条件为 OFF，BXF2(—)指令不执行。当执行条件为 ON，BXF2(—)传送 S，S＋1，…，S＋N 的内容到 D，D＋1，…，D＋N 中。如果 S 或 D 为常数，该常数指定源地址或目标地址在由 C 指定的 EM 组内。

控制字

当数据传送来自或送往 EM 时，控制字 C 包含源和目标 EM 组号。如果 S 或 D 不是常数，EM 组号可忽略。

C＋1 字包含传送字数，必须是 BCD 码(1～6144)。

控制字	位	功　能
C	0～7	指定源 EM 组号(00～02)
	8～15	指定目标 EM 组号(00～02)
C＋1	0～15	指定传送的字数(0～6144)

标志

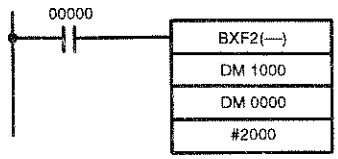
ER：N 不是 BCD 码。

S 和 S＋N 或 D 和 D＋N 不在同一数据区内。

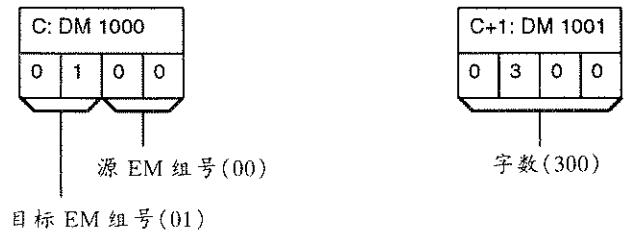
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

例

下例传送 DM0000 至 DM0299 间 300 字内容到 EM 组 1 中 EM2000 至 EM2299。(因为 S 不是常数，EM 组 00 不作为源字)。



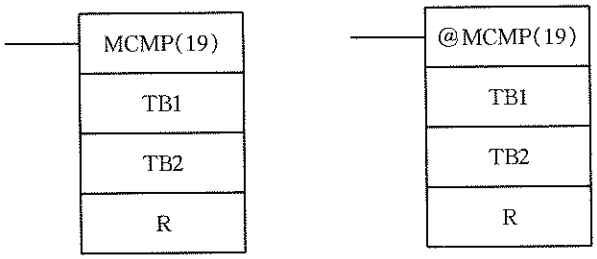
地址	指令	操作数
00200	LD	00000
00201	BXF2(—)	
		DM　1000
		DM　0000
		#　2000



5－17　数据比较指令

5－17－1　多字比较指令——MCMP(19)

梯形符号



操作数数据区域

TB1：表 1 起始字
IR, SR, AR, DM, HR, TC, LR

TB2：表 2 起始字
IR, SR, AR, DM, HR, TC, LR

R：结果字
IR, AR, DM, HR, TC, LR

限制

TB1 和 TB1＋15 必须在同一数据区，TB2 和 TB2＋15 也必须在同一数据区。

说明

当指令执行条件为 OFF，MCMP(19)指令不执行。当执行条件为 ON，MCMP(19)比较 TB1 和 TB2 内容，TB1＋1 和 TB2＋1 内容，……依次比较直到 TB1＋15 和 TB2＋15 内容比较。如果第一对比较内容相同，结果 R 的第一位(bit)为 OFF，等等……即，如果 TB1 的内容与 TB2 内容相同，00 位变为 OFF，如果 TB1＋1 和 TB2＋1 内容相同，01 位为 OFF，等等。R 中的其余位变为 ON。

标志

ER：其中一个表(即，TB1～TB1＋15，或 TB2～TB2＋15)超出数据区。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区。)

例

下例中表明 MCMP(19)指令的比较和比较结果情况。这里，当 00000 为 ON 的每个扫描周期都执行比较指令。



地址	指令	操作数
00000	LD	00000
00001	MCMP(19)	
		100
		DM 0200
		DM 0300

TB1：IR 100

IR 100	0100
IR 101	0200
IR 102	0210
IR 103	ABCD
IR 104	ABCD
IR 105	ABCD
IR 106	ABCD
IR 107	0800
IR 108	0900
IR 109	1000
IR 110	ABCD
IR 111	ABCD
IR 112	ABCD
IR 113	1400
IR 114	0210
IR 115	1212

TB2：DM 0200

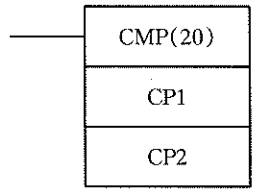
DM 0200	0100
DM 0201	0200
DM 0202	0210
DM 0203	0400
DM 0204	0500
DM 0205	0600
DM 0206	0210
DM 0207	0800
DM 0208	0900
DM 0209	1000
DM 0210	0210
DM 0211	1200
DM 0212	1300
DM 0213	1400
DM 0214	0210
DM 0215	1600

R：DM 0300

DM 030000	0
DM 030001	0
DM 030002	0
DM 030003	1
DM 030004	1
DM 030005	1
DM 030006	1
DM 030007	0
DM 030008	0
DM 030009	0
DM 030010	1
DM 030011	1
DM 030012	1
DM 030013	0
DM 030014	0
DM 030015	1

5－17－2 单字比较指令——CMP(20)

梯形符号



操作数数据区域

CP1：第一比较内容字
IR, SR, AR, DM, HR, TC, LR, #

CP2：第二比较内容字
IR, SR, AR, DM, HR, TC, LR, #

限制

当与定时器或计数器当前值比较时，比较值必须是 BCD 码。

说明

当指令执行条件为 OFF，CMP(20)指令不执行。当执行条件为 ON，CMP(20)比较 CP1 和 CP2 内容并将比较结果输出到 SR 区的 GR，EQ 和 LE 标志。

注意事项

在 CMP(20)指令和访问 EQ，LE，和 GR 标志的指令间插入其它指令，可能会使这些标志状态发生变化。必须在这些所需的状态变化前使用它们。

CMP(20)不能用于比较带符号二进制数据。可用 CPS(—)指令取代 CMP(20)。详见 5－17－8“带符号二进制比较指令—CPS(—)”章节。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域。）

EQ：如果 CP1 等于 CP2，EQ 为 ON。

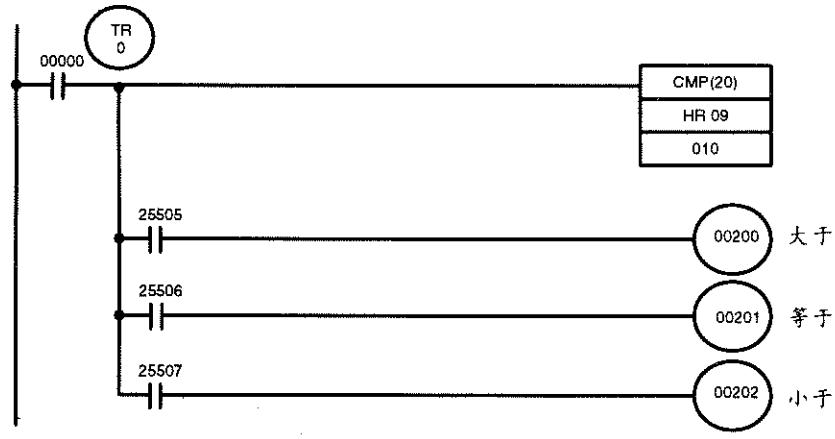
LE：如果 CP1 小于 CP2，LE 为 ON。

GR：如果 CP1 大于 CP2，GR 为 ON。

标志	地址	C1＜C2	C1＝C2	C1＞C2
GR	25505	OFF	OFF	ON
EQ	25506	OFF	ON	OFF
LE	25507	ON	OFF	OFF

例 1：保存 CMP(20)结果

下例中表明如何即时保存比较结果。如果 HR09 内容大于 010，00200 变为 ON；如果两内容相等，00201 变为 ON；如果 HR09 内容小于 010，00202 变为 ON，在一些应用中，三个输出中只有一个是必需的，那么就不必用 TRO 指令。用这样的编程方式，只有当 CMP(20)执行，00200，00201，00202 状态才会变化。



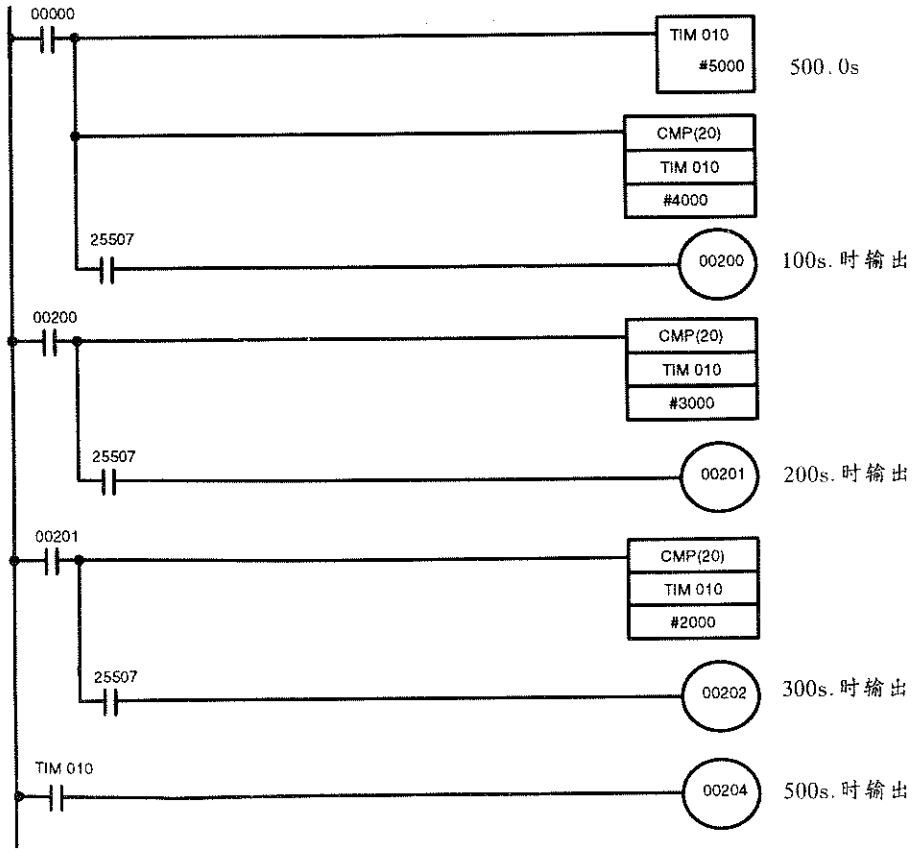
地址	指令	操作数
00000	LD	00000
00001	OUT	TR 0
00002	CMP(20)	
		010
		HR 09
00003	LD	TR 0
00004	AND	25505

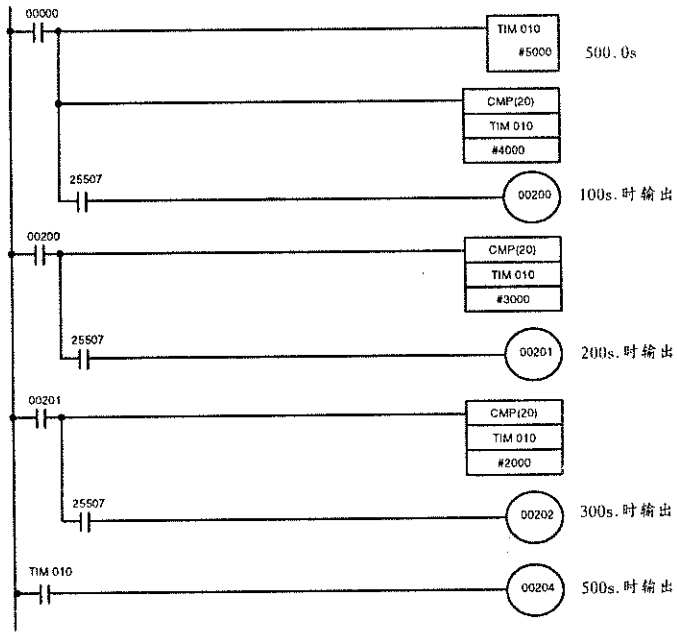
地址	指令	操作数
00005	OUT	00200
00006	LD	TR 0
00007	AND	25506
00008	OUT	00201
00009	LD	HR 0
00010	AND	25507
00011	OUT	00202

例 2:定时器操作过程中获得指示

下例中使用了定时器 TIM,比较指令 CMP(20)和小于标志 LE(25507),在定时器减计时的特定时间产生输出信号。00000 为 ON,定时器定时开始,00000 为 OFF,定时器 TIM010 复位,后面二个比较指令 CMP(20)不执行(即,在 OFF 条件下执行)。100 秒后 00200 产生输出;200 秒后 00201 输出;300 秒后 00202 输出;500 秒后 00204 输出。

为了保证 00200,00201,00202 定时减值过程正确地工作,梯形图的分支结构是重要的。因为这里所有的比较皆以定时器当前值 PV 为基准,所以每条 CMP(20)用到的其它操作数必须是 4 位 BCD 码。

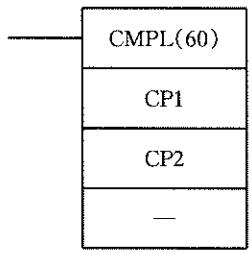




地址	指令	操作数
00000	LD	00000
00001	TIM	010
		# 5000
00002	CMP(20)	
		TIM 010
		# 4000
00003	AND	25507
00004	OUT	00200
00005	LD	00200
00006	CMP(20)	
		TIM 010
		# 3000
00007	AND	25507
00008	OUT	00201
00009	LD	00201
00010	CMP(20)	
		TIM 010
		# 2000
00011	AND	25507
00012	OUT	00202
00013	LD	TIM 010
00014	OUT	00204

5－17－3　双字比较指令－CMPL(60)

梯形符号



操作数数据区域

CP1：第一个比较字对的起始字
IR, SR, AR, DM, HR, TC, TR

CP2：第二个比较字对的起始字
IR, SR, AR, DM, HR, TC, LR

限制

CP1 和 CP1＋1 必须在同一数据区，CP2 和 CP2＋1 也必须在同一数据区。

说明

当指令执行条件为 OFF，CMPL(60)不执行。当执行条件为 ON，CMP2(60)指令将 CP1＋1 和 CP1，CP2＋1 和 CP2 的 4 位十六进制内容联接生成两个 8 位十六进制数 CP1＋1，CP1 和 CP2＋1，CP2 然后比较这两个 8 位数并将比较结果输出到 SR 区的大于(GR)，等于(EQ)，和小于(LE)标志位。

注意事项

在 CMPL(60)指令和访问 EQ，LE，和 GR 标志操作的指令间插入其它指令，可能会使这些标志状态发生变化。若需要，则应在这些状态变化之前使用它们。

CMPL(60)不能用于比较带符号的二进制数据。比较带符号二进制数据可用 CPSL(一)指令取代 CMP(20)。详见 5－17－9“带符号双字二进制比较指令——CPSL(一)”章节。

标志

ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

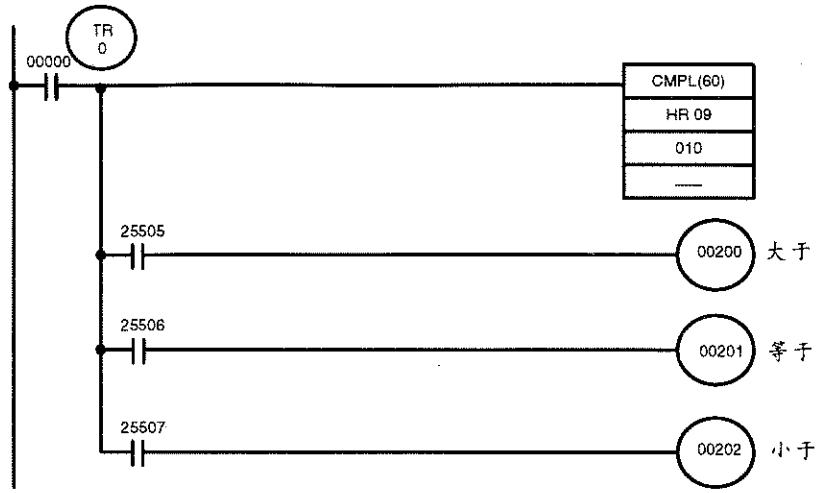
GR:如果 CP1＋1,CP1 大于 CP2＋1,CP2,GR 为 ON。

EQ:如果 CP1＋1,CP1 等于 CP2＋1,CP2,EQ 为 ON。

LE:如果 CP1＋1,CP1 小于 CP2＋1,CP2,LE 为 ON。

例:保存 CMPL(60)结果

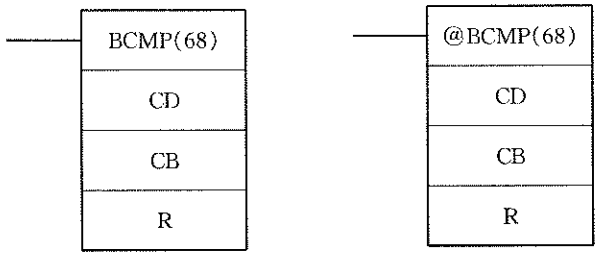
下例说明如何即时保存比较结果。如果 HR10,HR09 内容大于 011,010 的内容,这时 00200 变为 ON;如果两内容相等,00201 变为 ON;如果 HR10,HR09 内容小于 011,010,00202 变为 ON。在一些应用中,三个输出中只有一个是必需的,那么就不必用 TR0 指令。用这种编程方法,仅当 CMPL(60)执行时,00200,00201,00202 状态才会变化。



地址	指令	操作数
00000	LD	00000
00001	OUT	TR 0
00002	CMP(20)	
		HR 09
		010
00003	AND	25505
00004	OUT	00200
00005	LD	TR 0
00006	AND	25506
00007	OUT	00201
00008	LD	TR 0
00009	AND	25507
00010	OUT	00202

5－17－4 块比较指令——BCMP(68)

梯形符号



操作数数据区域

CD：比较数据
IR, SR, AR, DM, HR, TC, LR, #
CB：比较块起始字
IR, DM, HR, TC, LR
R：结果字
IR, SR, AR, DM, HR, TC, LR

限制

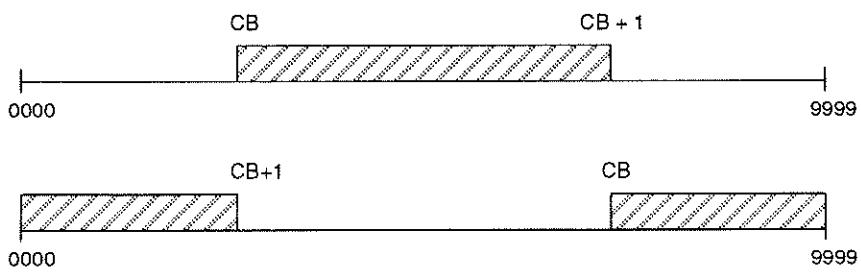
比较块中每个下限值必须小于等于上限值。

说明

当指令执行条件为 OFF，BCMP(68)不执行。当执行条件为 ON，BCMP(68)将数据(CD)与 CB，CB＋1，CB＋2……，CB＋32 组成的块所定义的区域比较。每一个区域由两个字定义，第一个作为下限，第二个作为上限。如果比较数据 CD 在这些范围的任何一个之中(包括上下限)，R 中的对应位为 ON。比较完成后，比较值在上下限范围内的 R 中相应的位为 ON，如下所示，R 中其余位为 OFF。

CB≤CD≤CB＋1	位 00
CB＋2≤CD≤CB＋3	位 01
CB＋4≤CD≤CB＋5	位 02
CB＋6≤CD≤CB＋7	位 03
CB＋8≤CD≤CB＋9	位 04
CB＋10≤CD≤CB＋11	位 05
CB＋12≤CD≤CB＋13	位 06
CB＋14≤CD≤CB＋15	位 07
CB＋16≤CD≤CB＋17	位 08
CB＋18≤CD≤CB＋19	位 09
CB＋20≤CD≤CB＋21	位 10
CB＋22≤CD≤CB＋23	位 11
CB＋24≤CD≤CB＋25	位 12
CB＋26≤CD≤CB＋27	位 13
CB＋28≤CD≤CB＋29	位 14
CB＋30≤CD≤CB＋31	位 15

通常每个区域的第 1 个字小于第 2 个字。如果第 1 个字小于第 2 个字，那末当 CD 在这二个字定义的范围之外时(如下图所示)，R 中相应位为 OFF。



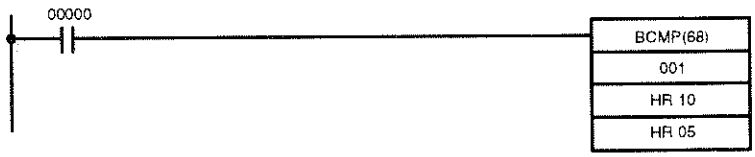
标志

ER：比较块(即 CB 到 CB＋31)超出数据区。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或者超出 DM 区域)

例

下例表明 BCMP(68)指令的执行和产生的结果。当执行条件 00000 为 ON 时每个循环周期都执行比较指令。



地址	指令	操作数
00000	LD	00000
00001	BCMP(68)	
		001
		HR 10
		HR 05

CD001	
001	0210

IR001 中的数据(内容为 0210)与给定范围比较。

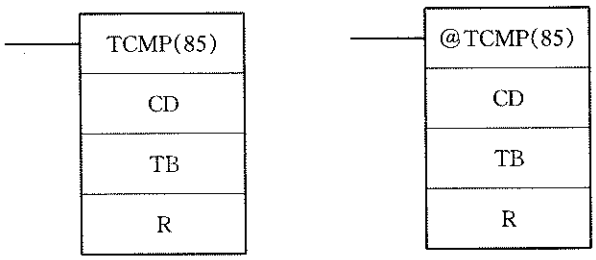
下限	
HR 10	0000
HR 12	0101
HR 14	0201
HR 16	0301
HR 18	0401
HR 20	0501
HR 22	0601
HR 24	0701
HR 26	0801
HR 28	0901
HR 30	1001
HR 32	1101
HR 34	1201
HR 36	1301
HR 38	1401
HR 40	1501

上限	
HR 11	0100
HR 13	0200
HR 15	0300
HR 17	0400
HR 19	0500
HR 21	0600
HR 23	0700
HR 25	0800
HR 27	0900
HR 29	1000
HR 31	1100
HR 33	1200
HR 35	1300
HR 37	1400
HR 39	1500
HR 41	1600

R：HR 05	
HR 0500	0
HR 0501	0
HR 0502	1
HR 0503	0
HR 0504	0
HR 0505	0
HR 0506	0
HR 0507	0
HR 0508	0
HR 0509	0
HR 0510	0
HR 0511	0
HR 0512	0
HR 0513	0
HR 0514	0
HR 0515	0

5－17－5 表格比较指令——TCMP(85)

梯形符号



操作数数据区域

CD：比较数据
IR, SR, AR, DM, HR, TC, LR, #
TB：比较表起始字
IR, AR, DM, HR, TC, LR
R：结果字
IR, SR, AR, DM, HR, TC, LR

限制

TB 和 TB＋15 必须在同一数据区内。

说明

当指令执行条件为 OFF，TCMP(85)指令不执行。当执行条件为 ON，TCMP(85)将数据 CD 和 TB，TB

+1, TB+2, ……TB+15 的内容进行比较。如果 CD 与这些字中任何一个内容相同, 则 R 中相应位置位(ON)。例如, 如果 CD 等于 TB 内容, R 中 00 位为 ON, 如果与 TB+1 相同, 01 位为 ON, 等等。R 中的其它位为 OFF。

标志

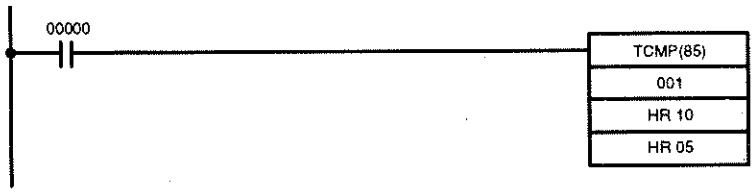
ER: 比较数(即, TB~TB+15)超出数据区。间接寻址 DM 字号不存在。

(*DM 字内容不是 BCD 码或超出 DM 区域)

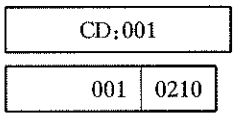
EQ: 如果比较表中没有一个字和 CD 相同, EQ 为 ON, 即 R 中所有位都为 OFF。

例

下例表示 TCMP(85)指令的执行和产生的结果。当执行条件 00000 为 ON 时, 每个循环周期都执行比较指令。



地址	指令	操作数
00000	LD	00000
00001	TCMP(85)	
		001
		HR 10
		HR 05



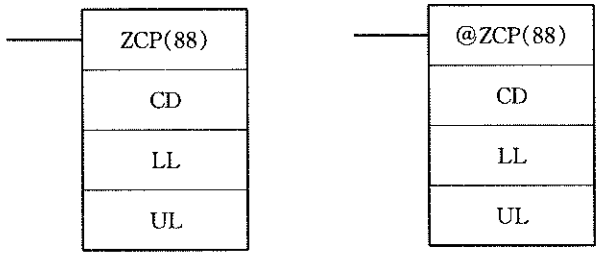
IR001 中的数据与表中数据比较

数据表	
HR 10	0100
HR 11	0200
HR 12	0210
HR 13	0400
HR 14	0500
HR 15	0600
HR 16	0210
HR 17	0800
HR 18	0900
HR 19	1000
HR 20	0210
HR 21	1200
HR 22	1300
HR 23	1400
HR 24	0210
HR 25	1600

R:HR 05	
HR 0500	0
HR 0501	0
HR 0502	1
HR 0503	0
HR 0504	0
HR 0505	0
HR 0506	1
HR 0507	0
HR 0508	0
HR 0509	0
HR 0510	1
HR 0511	0
HR 0512	0
HR 0513	0
HR 0514	1
HR 0515	0

5－17－6 区域比较指令——ZCP(88)

梯形符号



操作数数据区域

CD: 比较数据
IR, SR, AR, DM, HR, TC, LR, #
LL: 区间下限
IR, SR, AR, DM, HR, TC, LR, #
UL: 区间上限
IR, SR, AR, DM, HR, TC, LR, #

限制

LL 必须小于等于 UL。

说明

当指令执行条件为 OFF，ZCP(88)指令不执行。当执行条件为 ON，ZCP(88)将 CD 与下限 LL 和上限 UL 定义的区间进行比较，把结果输出到 SR 区的 GR，EQ 和 LE 标志中。结果标志状态如下表所示。

比较结果	标志状态		
	GR(SR 25505)	EQ(SR 25506)	LE(SR 25507)
CD＜LL	0	0	1
LL≤CD≤UL	0	1	0
UL＜CD	1	0	0

注意事项

在 ZCP(88)指令和接收 EQ，LE，GR 标志操作的指令间插入其它指令，可能会使这些标志状态发生变化。一定要在这些所需的状态变化之前使用它们。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

LL 大于 UL。

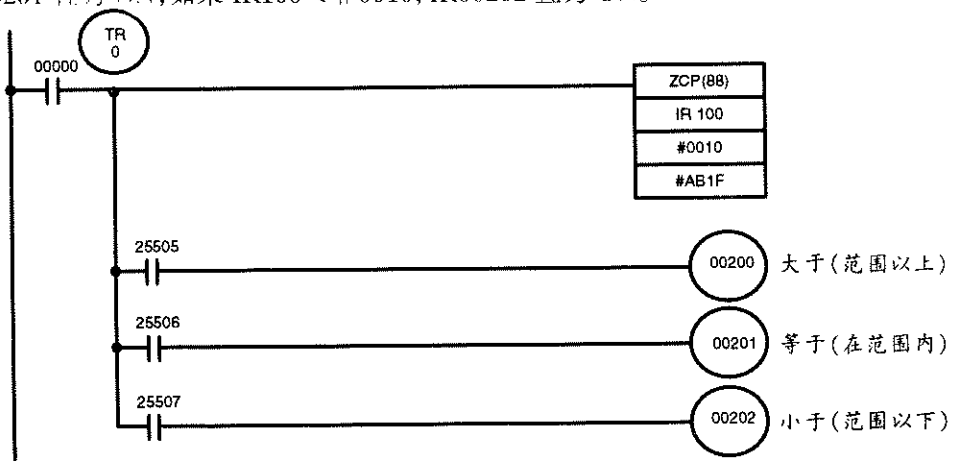
EQ：如果 LL≤CD≤UL，EQ 为 ON。

LE：如果 CD＜LL，LE 为 ON。

GR：如果 CD＞UL，GR 为 ON。

例：保存 ZCP(88)执行结果

下例说明如何即刻保存比较结果。如果 IR100＞＃AB1F，IR00200 置为 ON；如果＃0010≤IR100≤＃AB1F，IR00201 置为 ON；如果 IR100＜＃0010，IR00202 置为 ON。

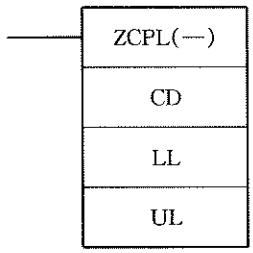


地址	指令	操作数	
00000	LD		00000
00001	OUT	TR	0
00002	ZCP(88)		
		IR	100
		#	0010
00003	LD	#	ABIF
00004	AND		25505

地址	指令	操作数	
0005	OUT		00200
0006	LD	TR	0
0007	AND		25506
0008	OUT		00201
0009	LD	TR	0
00010	AND		25507
00011	OUT		00202

5－17－7 双字区域比较指令——ZCPL(—)

梯形符号



操作数数据区域

CD：比较数据
IR, SR, AR, DM, HR, LR

LL：区间下限
IR, SR, AR, DM, HR, LR

UL：区间上限
IR, SR, AR, DM, HR, LR

限制

LL＋1, LL 组成的 8 位数值必须小于等于 UL＋1, UL 组成的 8 位数值。CD 和 CD＋1 必须在同一数据区，同样，LL 和 LL＋1, UL 和 UL＋1 也必须在同一数据区。

说明

当指令执行条件为 OFF, ZCPL(—)指令不执行，当执行条件为 ON, ZCPL(—)将 CD, CD＋1 中的 8 位数值与下限 LL＋1, LL 和上限 UL＋1, UL 定义的区间比较，把比较结果输出到 SR 区的 GR, EQ, LE 标志位中。结果标志状态如下表所示。

比较结果	标志状态		
	GR(SR 25505)	EQ(SR 25506)	LE(SR 25507)
CD, CD＋1＜LL＋1, LL	0	0	1
LL＋1, LL≤CD, CD＋1≤UL＋1, UL	0	1	0
UL＋1, UL＜CD, CD＋1	1	0	0

注事事项

在 ZCPL(—)指令和接收 EQ, LE, GR 标志操作的指令间插入其它指令，可能会使这些标志状态发生变化。一定要在这些所需要的状态变化之前使用它们。

标志

ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

LL＋1, LL 大于 UL＋1, UL。

EQ：如果 LL＋1, LL≤CD, CD＋1≤UL＋1, UL, EQ 为 ON。

LE：CD, CD＋1＜LL＋1, LL, LE 为 ON。

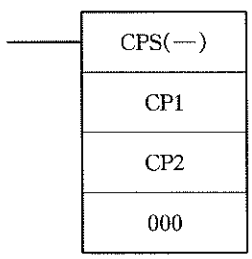
GR：CD, CD＋1＞UL＋1, UL, GR 为 ON。

例

参见 5－17－6“区域比较指令－ZCP(88)”章节中例子。ZCP(88)与 ZCPL(—)仅比较数据的位置不同。

5－17－8 带符号二进制比较指令－CPS(—)

梯形符号



操作数数据区域

CP1：第一比较内容通道号
IR, SR, AR, DM, HR, TC, LR, #

CP2：第二比较内容通道号
IR, SR, AR, DM, HR, TC, LR, #

第三个操作数：置为 000
—

说明

当指令执行条件为 OFF，CPS(—)指令不执行。当执行条件为 ON 时，CPS(—)比较 CP1 和 CP2 中 16 位(4 位数)带符号二进制数据，将比较结果输出到 SR 区的 GR，EQ，LE 标志位中。

注 1．有关 16 位带符号二进制数详见第 25 页。

2．有关保存比较结果详见 5－17－2“单字比较指令－CMP(20)。”

注意事项

在 CPS(—)指令和接收 EQ，LE，GR 标志操作的指令间插入其它指令，可能会使这些标志状态发生变化。一定要在这些所需的状态变化之前使用。

标志

ER：间接寻址 DM 字不存在，(＊DM 字内容不是 BCD 码或超出 DM 区域。)

EQ：如果 CP1 等于 CP2，EQ 为 ON。

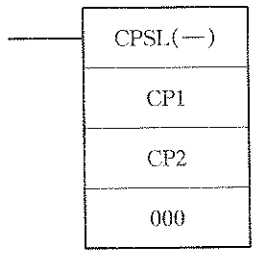
LE：如果 CP1 小于 CP2，LE 为 ON。

GR：如果 CP1 大于 CP2，GR 为 ON。

比较结果	标志状态		
	GR(SR 25505)	EQ(SR 25506)	LE(SR 25507)
CP1＜CP2	0	0	1
CP1＝CP2	0	1	0
CP1＝CP2	1	0	0

5－17－9 带符号双字二进制比较指令——CPSL(—)

梯形符号



操作数数据区域

CP1：第一比较内容字
IR，SR，AR，DM，HR，TC，LR
CP2：第二比较内容字
IR，SR，AR，DM，HR，TC，LR
第三操作数：置为 000
—

限制

CP1 和 CP1＋1 必须在同一数据区，同样，CP2 和 CP2＋1 也必须在同一数据区。

说明

当指令执行条件为 OFF，CPSL(—)指令不执行。当执行条件为 ON，CPSL(—)比较 CP1＋1，CP1 和 CP2＋1，CP2 中的 32 位(8 位数)带符号二进制数据，将比较结果输出到 SR 区的 GR，EQ，LE 标志位中。

注 1．有关 32 一位带符号二进制数据详见 25 页。

2．有关保存比较结果详见 5－17－2 单字“比较指令－CMP(20)。”

注意事项

在 CPSL(—)指令和接收 EQ，LE，GR 标志操作的指令间插入其它指令，可能会使这些标志状态发生变化。一定要在这些所需的状态变化之前使用。

标志

ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:如果 CP1＋1,CP1 等于 CP2＋1,CP2,EQ 为 ON。

LE:如果 CP1＋1,CP1 小于 CP2＋1,CP2,LE 为 ON。

GR:如果 CP1＋1,CP1 大于 CP2＋1,CP2,GR 为 ON。

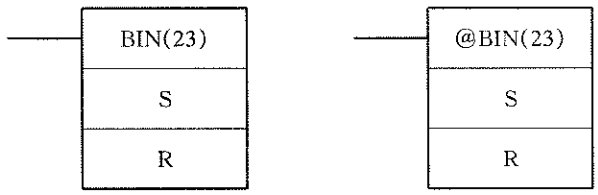
比较结果	标志状态		
	GR(SR 25505)	EQ(SR 25506)	LE(SR 25507)
CP1＋1,CP1＜CP2＋1,CP2	0	0	1
CP1＋1,CP1＝CP2＋1,CP2	0	1	0
CP1＋1,CP1＞CP2＋1,CP2	1	0	0

5－18 数据转换指令

数据转换指令是将字数据从一种形式转换为另一种形式,将将已转换的数据输出到指定的存放结果的字中。转换指令转换二进制(十六进制)和 BCD 码数据,转换 7 段显示数据,ASCⅡ码数据,转换编码数据和非编码数据。所有的转换指令只改变接收转换数据字的内容,即源字的内容在任何转换指令执行后保持不变。

5－18－1 BCD 码→二进制指令——BIN(23)

梯形符号



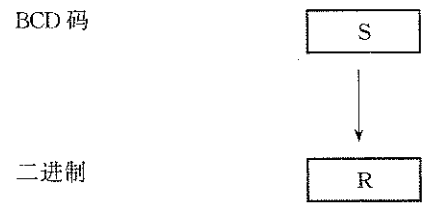
操作数数据区域

S:源字(BCD 码)
IR, SR, AR, DM, HR, TC, LR

R:结果字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF,BIN(23)指令不执行。当执行条件为 ON,BIN(23)将 S 中的 BCD 数据转换为等值的二进制数,输出到 R 中。转换过程中,S 的内容保持不变;仅 R 的内容发生变化。



BIN(23)指令可用于把 BCD 数据转换成二进制数据,这样在编程器或其它编程装置上的显示是十六进制而不是十进制数。也可用 BIN(23)的二进制转换实现二进制算术运算,而不采用 BCD 码的算术运算,例如:BCD 码和二进制数值必须相加时。

标志

ER:S 的内容不是 BCD 码。

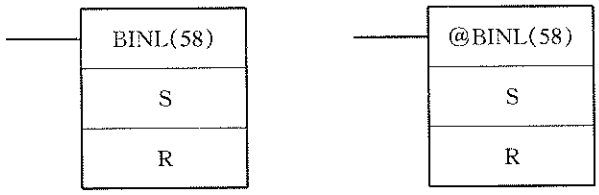
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:当转换结果是零,EQ 为 ON。

N:25402 总是 OFF。

5－18－2 双字 BCD 码→双字二进制指令——BINL(58)

梯形符号



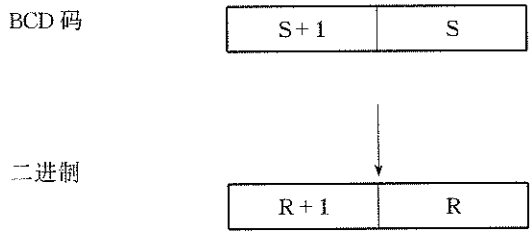
操作数数据区域

S:源起始字(BCD)
IR, SR, AR, DM, HR, TC, LR

R:结果起始字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF, BINL(58)不执行。当执行条件为 ON, BINL(58)把 S 和 S＋1 中的 8 位数转换为 32 位的二进制数据，并将转换结果输出到 R 和 R＋1 中。



标志

ER:S 和/或 S＋1 内容不是 BCD 码。

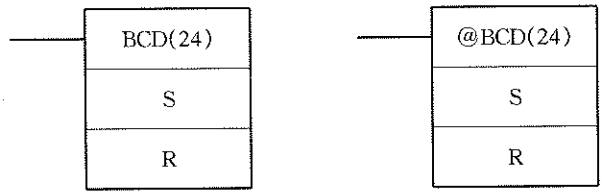
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:当转换结果是零，EQ 为 ON。

N:25402 总是 OFF。

5－18－3 二进制→BCD 码指令——BCD(24)

梯形符号



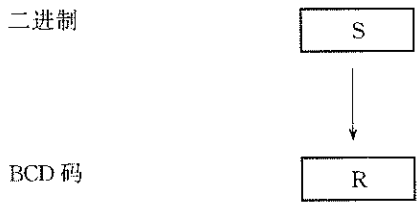
操作数数据区域

S:源字(二进制)
IR, SR, AR, DM, HR, LR

R:结果字
IR, SR, AR, DM, HR, LR

说明

BCD(24)指令将 S 的二进制(十六进制)数转换成等值的 BCD 数，并将 BCD 数据输出到 R 中。转换过程中，S 内容保持不变，仅 R 内容发生变化。



BCD(24)可用将二进制数转换成 BCD 数，这样在编程器或其它编程装置上的显示是十进制而不是十六进制数。也可用 BCD(24)的 BCD 码转换实现 BCD 码算术操作，而不采用二进制数算术操作，例如 BCD 码

和二进制数值必须相加时。

注

如果 S 的内容超过 270F，转换结果将会超过 9999，这样，BCD(24)指令将不执行。当指令不执行，R 中的内容保持不变。

带符号二进制数据

不能直接使用 BCD(24)将带符号二进制数据转换为 BCD 数据。要转换带符号的二进制数据时，首先要判断数据是正数这是负数。如果是正数，BCD(24)可将这二进制数转换成 BCD 数据，如果是负数，在 BCD(24)之前先用二进制求补指令 NEG(—)将数转换成无符号二进制数。有关带符号二进制数据详见 P25。

标志

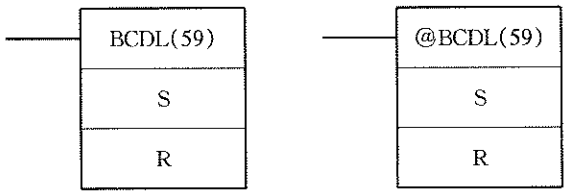
ER：S 大于 270F。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当转换结果是零，EQ 为 ON。

5－18－4 双字二进制→双字 BCD 码指令——BCDL(59)

梯形符号



操作数数据区域

S：源起始字(二进制)
IR，SR，AR，DM，HR，LR

R：结果起始字
IR，SR，AR，DM，HR，LR

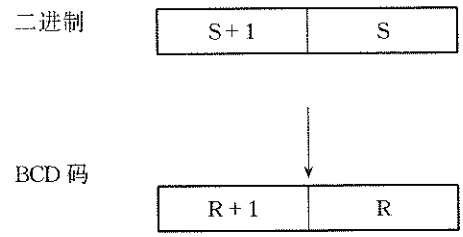
限制

如果 S 和 S＋1 的内容超过 05F5EOFF，转换结果将会大于 99999999，这样，BCDL(59)指令将不执行。当指令不执行，R 和 R＋1 的内容保持不变。

S 和 S＋1 必须在同一数据区，同样，R 和 R＋1 也必须在同一数据区。

说明

BCDL(59)将 S 和 S＋1 中的 32－位二进制内容转换成 8 位 BCD 码数据，转换后的数据输出到 R 和 R＋1 中。



带符号二进制数据

不能直接用 BCDL(59)将带符号的二进制数据转换成 BCD 数。要转换带符号的二进制数据时，首先要判断数据是正数还是负数。如果是正数，BCDL(59)可将这二进制数转换成 BCD 数。如果是负数，在执行 BCDL(59)之前先用双字二进制求补指令 NEGL(—)将该数转换成无符号二进制数。有关带符号二进制数据详见 25 页。

标志

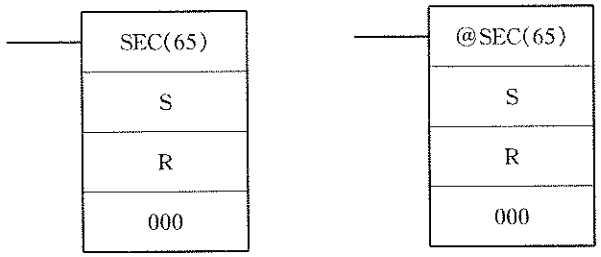
ER：R 和 R＋1 内容超过 99999999。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当转换结果是零，EQ 为 ON。

5－18－5 小时→秒指令——SEC(65)

梯形符号



操作数数据区域

S:源起始字(BCD 码)
IR, SR, AR, DM, HR, TC, LR
R:结果起始字(BCD 码)
IR, SR, AR, DM, HR, TC, LR
000:置 000
…

限制

S 和 S＋1 必须在同一数据区。R 和 R＋1 也必须在同一数据区。S 和 S＋1 必须是 BCD 数据，且还必须是小时/分/秒形式。

说明

SEC(65)指令用于将以小时/分/秒表示的时间转换成以秒表示的等值时间。

源数据中，秒数在 S 的 00 位到 07 位，分数在 S 的 08 位到 15 位。小时数在 S＋1 中。因此，可表示的最大值是 9,999 小时 59 分 59 秒。

转换的结果输出到 R 和 R＋1。可获得的最大值是 35,999,999 秒。

标志

ER:S 和 S＋1 或者 R 和 R＋1 不在同一数据区。

S 和/或 S＋1 内容不是 BCD 码。

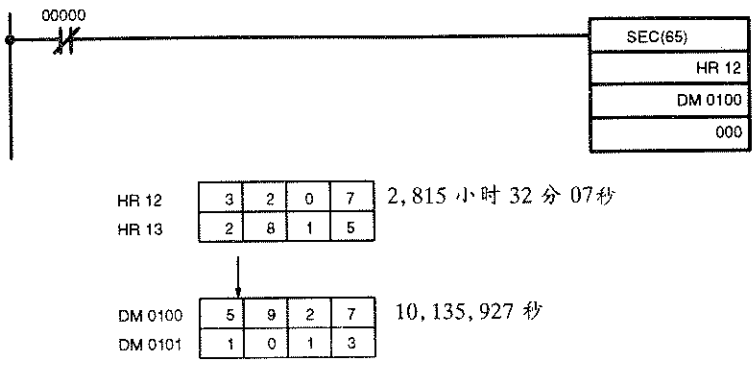
秒和/或分数值超过 59。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:当转换结果是零，EQ 为 ON。

例

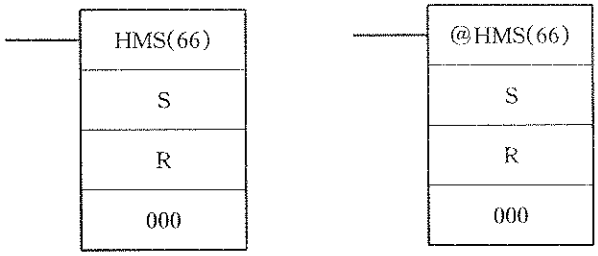
当 00000 为 OFF(即，指令执行条件为 ON)，下例中的指令将 HR12 和 HR13 中的小时、分、秒转换成秒并将转换结果存入 DM0100 和 DM0101 中，如下图所示。



地址	指令	操作数
00000	LD NOT	00000
00001	SEC(65)	
		HR 12
		DM 0100
		000

5－18－6 秒→小时指令－HMS(66)

梯形符号



操作数数据区域

S:源起始字(BCD)
IR, SR, AR, DM, HR, TC, LR
R:结果起始字(BCD)
IR, SR, AR, DM, HR, TC, LR
000:置 000
…

限制

S 和 S+1 必须在同一数据区。R 和 R+1 必须在同一数据区。S 和 S+1 必须是 BCD 数据，内容在 0～35,999,999 秒之间。

说明

HMS(66)指令用于将以秒表示的时间转换成以小时/分/秒表示的等值时间。

S 和 S+1 中指定的秒数转换成小时/分/秒表示的时间并存入 R 和 R+1 中。

转换结果中，秒数放在 R 的 00 位到 07 位；分数放在 R 的 08 位到 15 位，小时数放在 R+1。最大值将是 9,999 小时 59 分 59 秒。

标志

ER：S 和 S+1 或者 R 和 R+1 不在同一数据区。

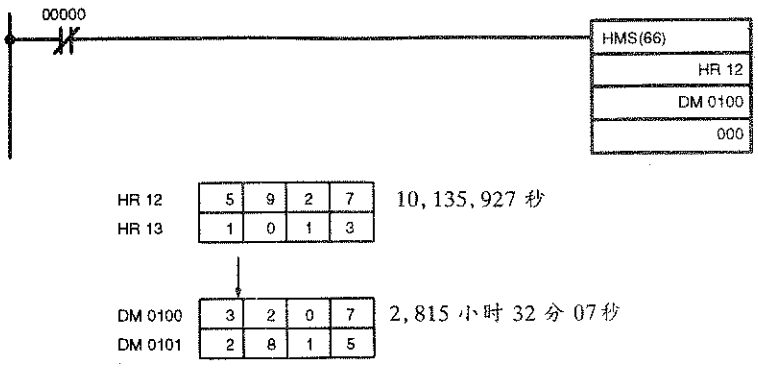
S 和/或 S+1 中内容不是 BCD 码或者超过 36,000,000 秒。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ：当转换结果是零，EQ 为 ON。

例

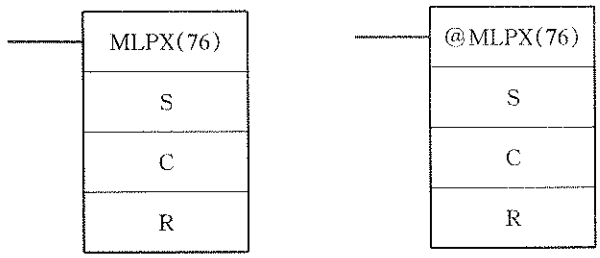
当 00000 为 OFF(即，指令执行条件为 ON)，下例指令将 HR12 和 HR13 中的秒数转换成小时、分和秒并将转换结果存入 DM0100 和 DM0101 中。



地址	指令	操作数
00000	LD NOT	00000
00001	HMS (66)	
		HR 12
		DM 0100
		000

5－18－7 4→16/8→256 译码指令——MLPX(76)

梯形符号



操作数数据区域

S:源字
IR, SR, AR, DM, HR, TC, LR
R:控制字
IR, SR, AR, DM, HR, TC, LR, #
R:结果起始字
IR, SR, AR, DM, HR, LR

限制

当控制字 C 的最高位数是“0”，则 C 的最低两位数都必须在 0～3 之间。

当控制字 C 的最高位数是“1”，则 C 的最低两位数都必须在 0～1 之间。

所有结果字必须在同一数据区。

说明

根据 C 的数值，MLPX(76)可作为 4－16 译码器或者 8－256 译码器。

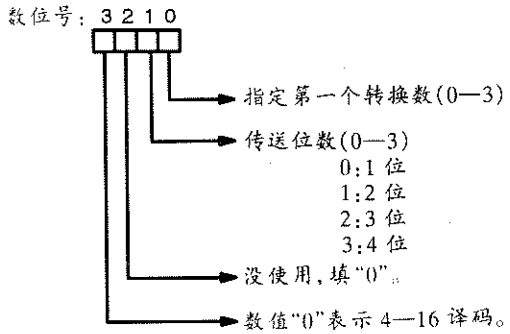
4－16 译码器

当控制字 C 的最高位数是“0”，MLPX(76)作为 4－16 译码器。S 中数的十六进制值用于指定多至 4 个结果通道中的位(bit)。在每个结果字中被指定的位置为 ON，每个字中其它 15 位变为 OFF。

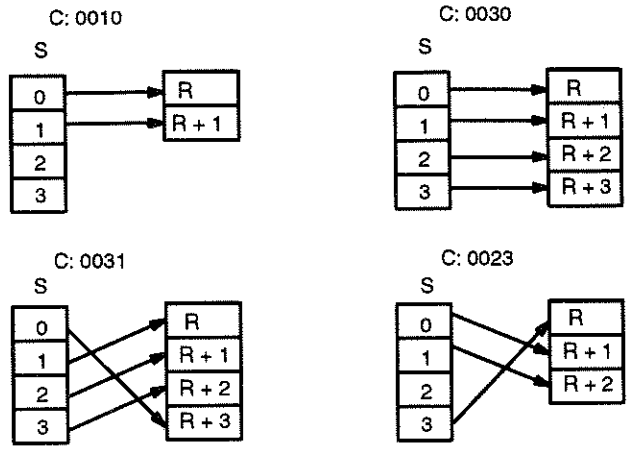
当指令执行条件 OFF，MLPX(76)不执行。当执行条件为 ON，MLPX(76)将 S 中多至 4 位数的 4bit 十六进制值转换成十进制的 0～15，这每个 10 进制的数用于指定位‘bit’的位置。这时结果字中与每一 10 进制数相对应的位变为 ON。如果指定的数不止一个，那么从 R 字开始顺序置位后继字中的相应位(见下例)。

控制字

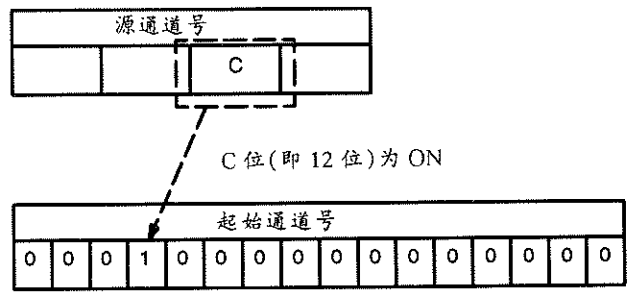
控制字 C 的位设置如下。设置 C 最高位数为“0”，则为 4→16 译码。



以下所示是一些控制字 C 值及由其控制的数到字的转换例子：



下面是取 S 的第二个数的一位数译码操作的例子，这里 C 为 0001。



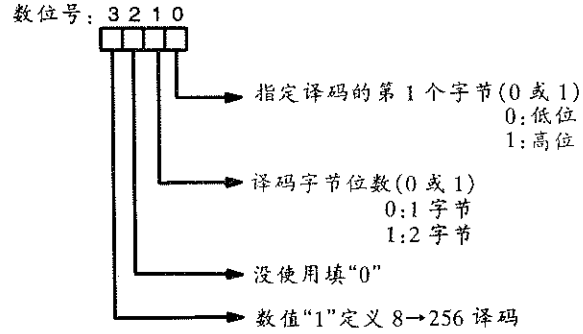
控制字 C 指定了从那位数开始转换及需转换的位数。如果定义的位数比 S 中余下的多(从第一个指定位开始数),则余下的位从 S 首位开始取。用于存放译码结果的最后一个字(R 加要译码的位数)必须和 R 在同一数据区,例如,译码两位,则不能指定在一个数据区里的最后一个字的地址。如果译码三位,则不能指定在一个数据区的最后二个字的地址。

8→256 译码

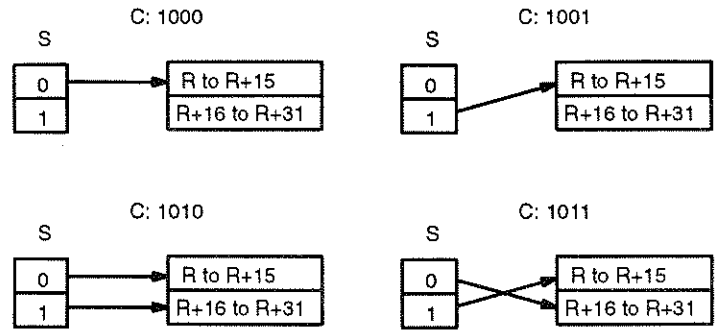
当控制字 C 的最高位数设置为 1,MLPX(76)作 8→256 译码操作。S 中的两字节十六进制值指定了 1 组或 2 组 16 个连续结果字(256bit)中的一位(bit),每组中的指定位置为 ON,该组中的其它 255 位为 OFF。

控制字

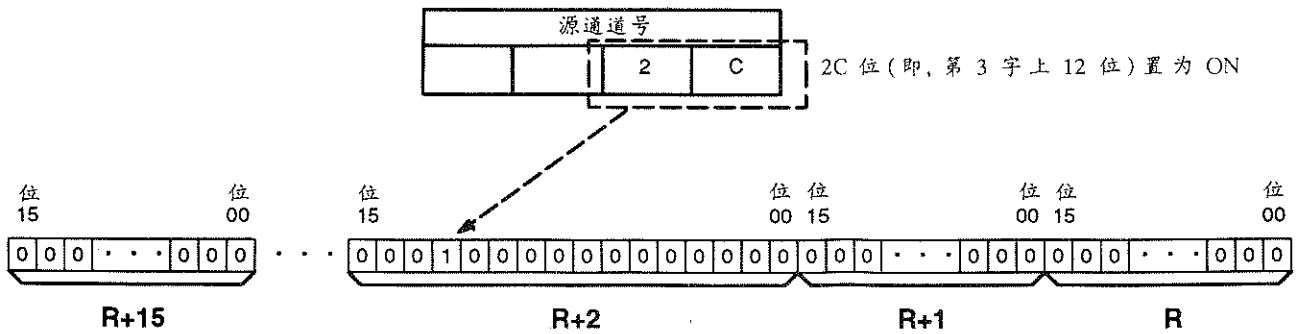
C 的控制数设置如下。设置 C 的最高位数为 1,则为 8→256 译码。



四个可能的 C 值和产生的译码如下所示(S 中 0 表示低位,1 表示高位)



下面是取 S 低位字节的单字节译码操作的例子(控制通道内容为 1000)



标志

ER:未定义控制字。

译码结果字不都在同一数据区。

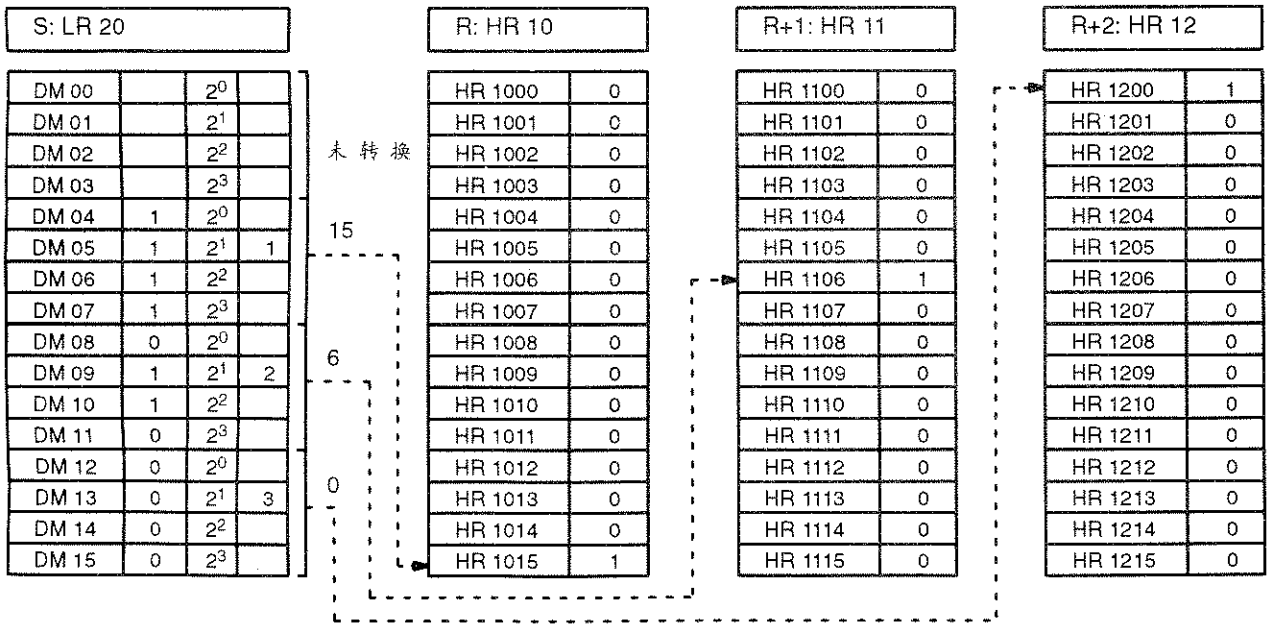
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

例 4→16 译码

下面程序将 LR20 的 3 位数转换成位的位置，并置 HR10 开始的连续 3 个字的对应位为 ON。



地址	指令	操作数
00000	LD	00000
00001	MLPX(76)	
		LR 20
		# 0021
		HR 10



S: LR 20			
DM 00		2^0	
DM 01		2^1	
DM 02		2^2	
DM 03		2^3	
DM 04	1	2^0	
DM 05	1	2^1	1
DM 06	1	2^2	
DM 07	1	2^3	
DM 08	0	2^0	
DM 09	1	2^1	2
DM 10	1	2^2	
DM 11	0	2^3	
DM 12	0	2^0	
DM 13	0	2^1	3
DM 14	0	2^2	
DM 15	0	2^3	

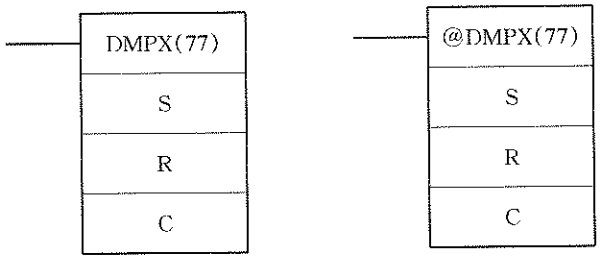
R: HR 10	
HR 1000	0
HR 1001	0
HR 1002	0
HR 1003	0
HR 1004	0
HR 1005	0
HR 1006	0
HR 1007	0
HR 1008	0
HR 1009	0
HR 1010	0
HR 1011	0
HR 1012	0
HR 1013	0
HR 1014	0
HR 1015	1

R+1: HR 11	
HR 1100	0
HR 1101	0
HR 1102	0
HR 1103	0
HR 1104	0
HR 1105	0
HR 1106	1
HR 1107	0
HR 1108	0
HR 1109	0
HR 1110	0
HR 1111	0
HR 1112	0
HR 1113	0
HR 1114	0
HR 1115	0

R+2: HR 12	
HR 1200	1
HR 1201	0
HR 1202	0
HR 1203	0
HR 1204	0
HR 1205	0
HR 1206	0
HR 1207	0
HR 1208	0
HR 1209	0
HR 1210	0
HR 1211	0
HR 1212	0
HR 1213	0
HR 1214	0
HR 1215	0

5－18－8 16→4/256→8 编码指令——DMPX(77)

梯形符号



操作数数据区域

S：源起始字
IR, SR, AR, DM, HR, TC, LR
R：结果字
IR, SR, AR, DM, HR, LR
R：控制字
IR, SR, AR, DM, HR, TC, LR, #

限制

当 C 的最高位数为 0，C 的最低两位数都必须在 0～3 之间。

当 C 的最高位数为 1，C 的最低两位数都必须在 0～1 之间。

所有源字必须在同一数据区。

说明

根据 C 的数值，DMPX(77)可用作 16→4 编码或 256→8 编码器操作。

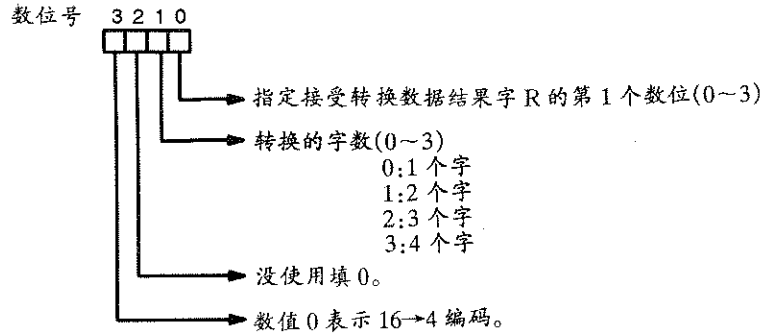
16→4 编码

当 C 的最高数位为 0，DMPX(77)用作 16→4 编码器操作。当指令执行条件为 OFF，DMPX(77)指令不执行。当执行条件为 ON，DMPX(77)根据源通道 S 中状态为 ON 的最高位位置，编码成相应的一位十六进制数，然后将这十六进制数传送到结果 R 中的指定数位。接受结果数位由 C 中设定值指定，C 设定值也指

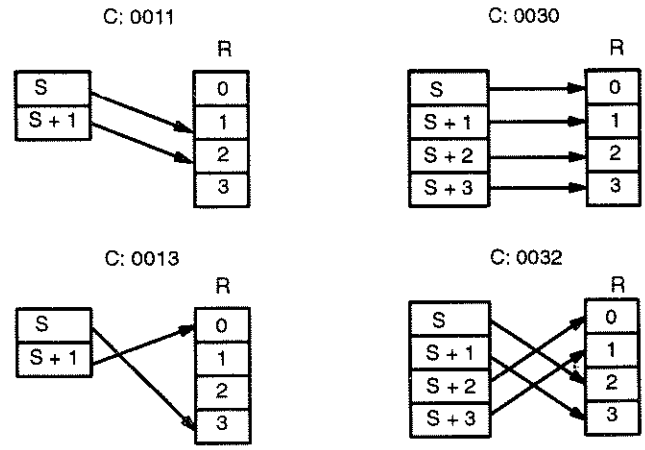
定了需编码的个数。

控制字

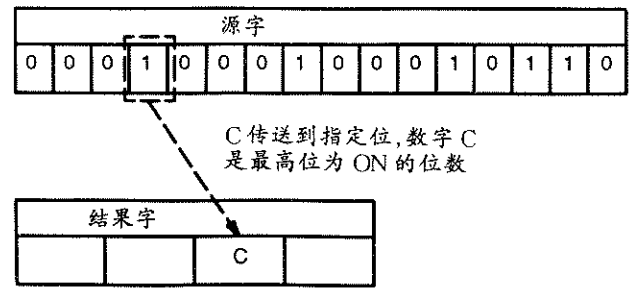
控制字 C 的设置如下。设置 C 的最高位数为“0”，则为 16→4 编码。



以下所示是一些控制字 C 值及由其产生的字到数的转换例子：



下面是一个存 R 第 1 数位的一位数编码操作的例子，这里 C 是 0001。



可以对从 S 开始多至 4 个连续字进行编码，并从指定的起始数位按顺序将这些结果写到结果 R 中。如果指定的位数比 R 中余下的多(从第一个指定位开始数。)则余下的位从 R 的开始位开始存放。转换源最后的字(S 加转换位数)必须和转换起始字 SB 在同一数据区。

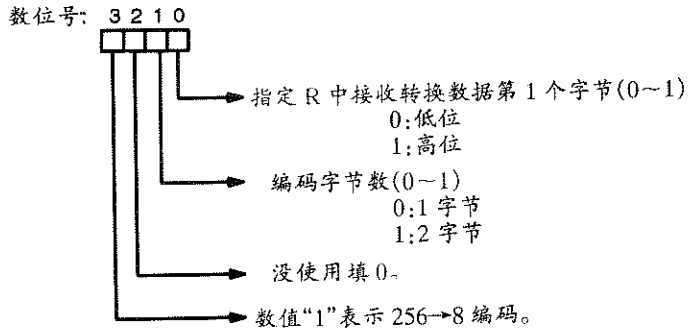
256→8 编码

当控制字 C 的最高位数是“1”，DMPX(77)用于 256→8 编码操作。

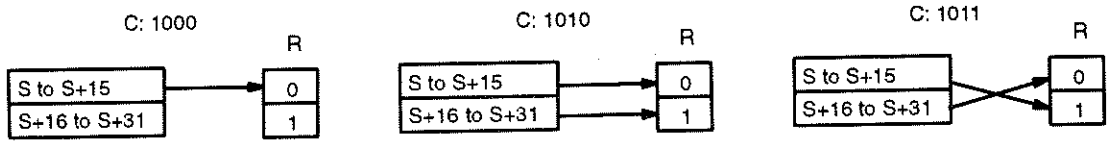
当指令执行条件为 OFF，DMPX(77)不执行，当执行条件为 ON，DMPX(77)根据 S～S＋15 或 S＋16～S＋31 十六个源字组中状态为 ON 的最高位位置，编码成相应的两位十六进制数值，然后将这十六进制数传送到 R 中指定字节。接收结果的字节位置由 C 指定，同时 C 中内容也指定了编码的字节数。

控制字

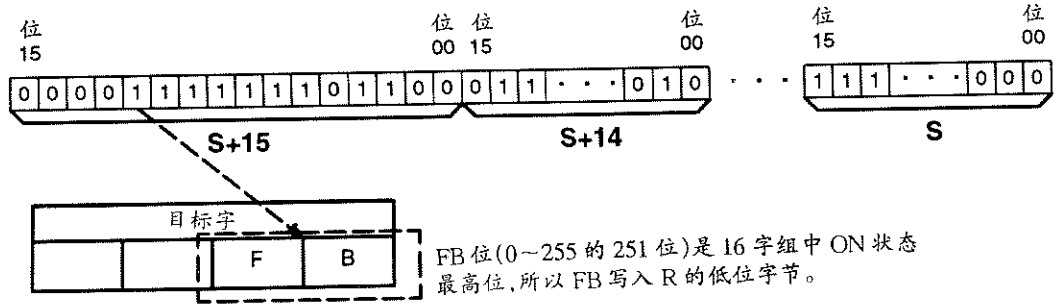
C 的设置如下示。设置 C 最高位数为“1”，则为 256→8 编码。



三种可能的 C 值和相应转换过程如下所示。(在 R 中,0 表示低位,1 表示高位)



下面是将结果送到 R 低位字节的单字节编码操作例子(在此例中,C 为 1000)。



标志

ER:未定义控制字。

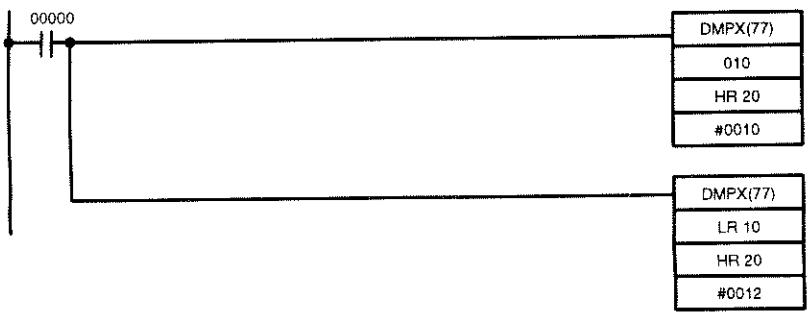
编码源字不都在同一数据区。

编码源字内容为零。(源字中没有一位为 ON。)

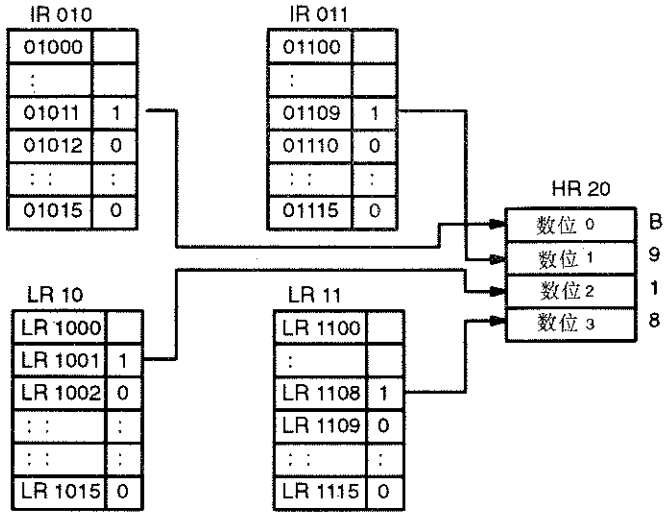
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

例 16→4 编码

当 00000 为 ON,例中梯形图将 IR 字 010 和 011 编码到 HR20 开始的两位数,然后将 LR10 和 11 编码到 HR20 的最后两位数。虽然每个源通道位状态没有都标明,但这里假设所表明 1 状态的位是对应字中 ON 的最高位。

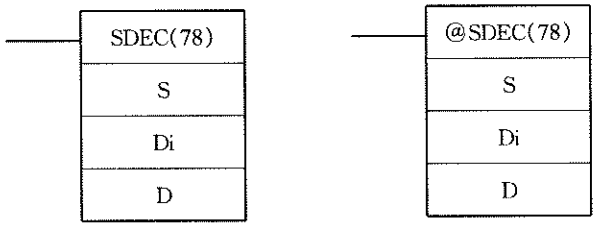


地址	指令	操作数
00000	LD	00000
00001	DMPX(77)	
		010
		HR 20
		# 0010
00002	DMPX(77)	
		LR 10
		HR 20
		# 0012



5－18－9 七段译码指令——SDEC(78)

梯形符号



操作数数据区域

S：源字(二进制)
IR, SR, AR, DM, HR, TC, LR
Di：数指定器
IR, SR, AR, DM, HR, TC, LR, #
D：目标起始字
IR, SR, AR, DM, HR, LR

限制

Di 必须在下面给定值之内。

所有目标字必须在同一数据区。

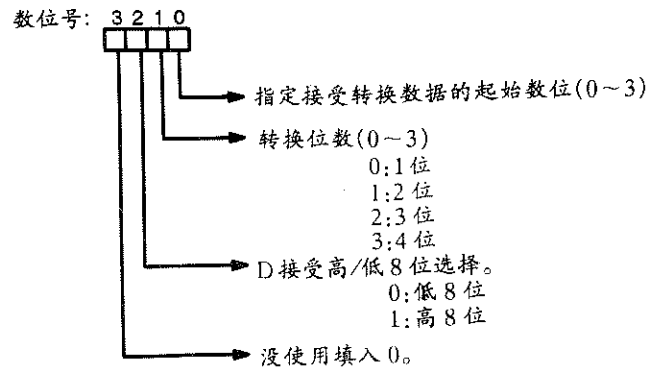
说明

当指令执行条件为 OFF，SDEC(78)指令不执行。当执行条件为 ON，SDEC (78)将 S 中的指定位的数转换成相应的 8－bit，7 段显示码，并将它们存入 D 起始的字中。

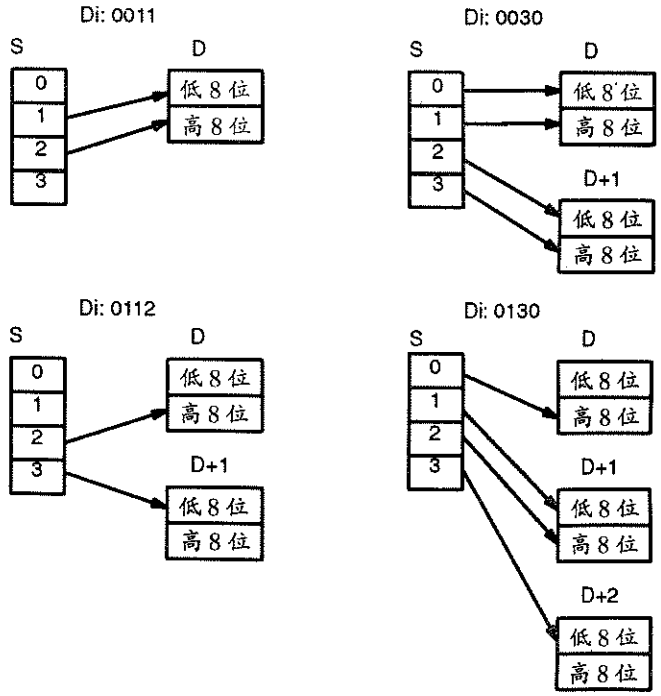
S 中的所有或任何位数可从指定的起始数位起按序转换。起始数位、转换位数和 D 中是高 8 位还是低 8 位接受第一个 7 段显示码由 Di 内容指定。如果指定多位数转换，转换后的 7 段显示码将从 D 中指定字节开始按序存放，每个码需要两数位(2×4bits)。如果指定的位数比 S 中余下的多(从指定的起始数位开始数)，则以后转换的位从 S 起始位开始。

数指定器

Di 数设置如下所示。

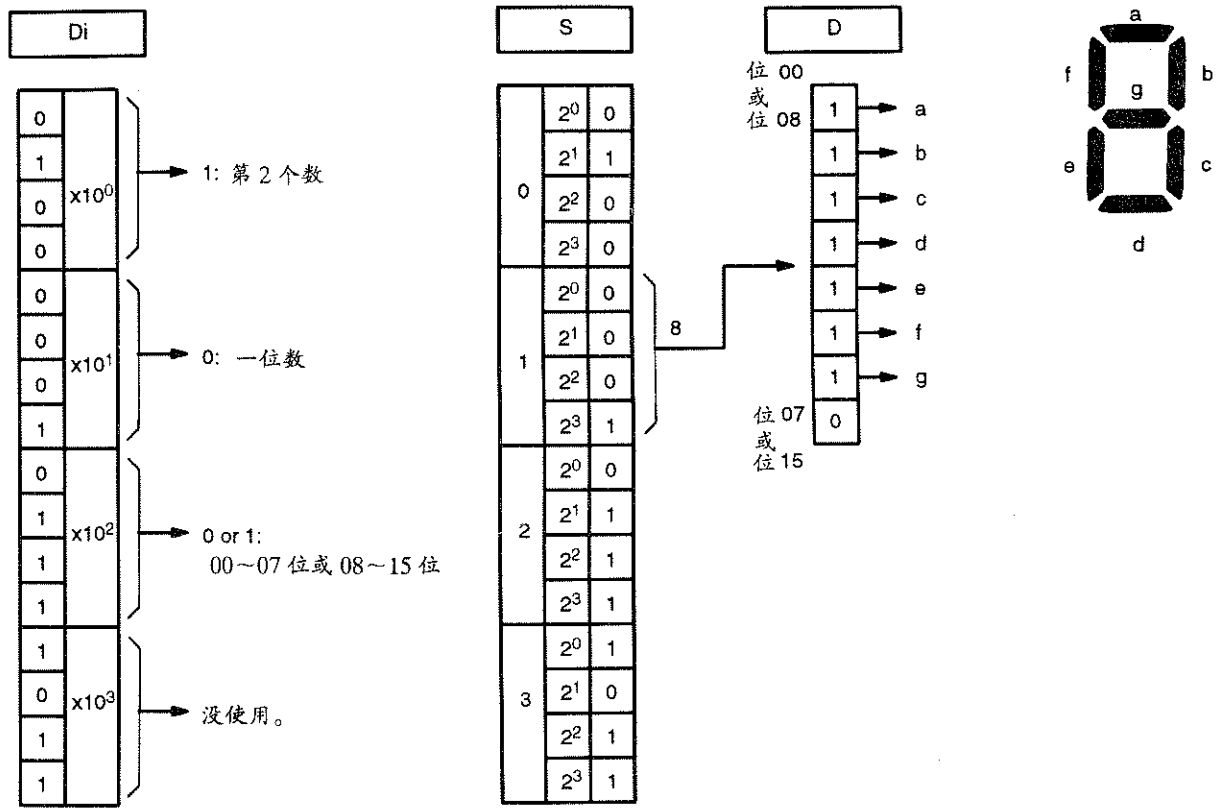


以下是一些 Di 值和相应的 4 位二进制到七段显示码的转换处理的例子。



例

下例为显示一个数字 8 的例子。例子中小写字母表示 7 段显示码的每一段对应哪一位。下面的表列出了所有 16 进制原始数据和它的转换码



源数据					转换后数据(段码)								显示
数	位				－	g	f	e	d	c	b	a	
0	0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	1
2	0	0	1	0	0	1	0	1	1	0	1	1	2
3	0	0	1	1	0	1	0	0	1	1	1	1	3
4	0	1	0	0	0	1	1	0	0	1	1	0	4
5	0	1	0	1	0	1	1	0	1	1	0	1	5
6	0	1	1	0	0	1	1	1	1	1	0	1	6
7	0	1	1	1	0	0	1	0	0	1	1	1	7
8	1	0	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	0	1	1	0	1	1	1	1	9
A	1	0	1	0	0	1	1	1	0	1	1	1	A
B	1	0	1	1	0	1	1	1	1	1	0	0	B
C	1	1	0	0	0	0	1	1	1	0	0	1	C
D	1	1	0	1	0	1	0	1	1	1	1	0	D
E	1	1	1	0	0	1	1	1	1	0	0	1	E
F	1	1	1	1	0	1	1	1	0	0	0	1	F

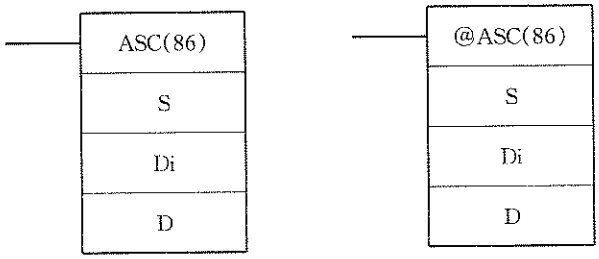
标志

ER:数指定器不正确或目标数据超出区域。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

5－18－10 ASCII 转换指令——ASC(86)

梯形符号



操作数数据区域

S:源字
IR, SR, AR, DM, HR, TC, LR

Di:数指定器
IR, SR, AR, DM, HR, TC, LR, #

D:目标起始字
IR, SR, AR, DM, HR, LR

限制

Di 必须在下面给定值之内。

所有目标字必须在同一数据区。

说明

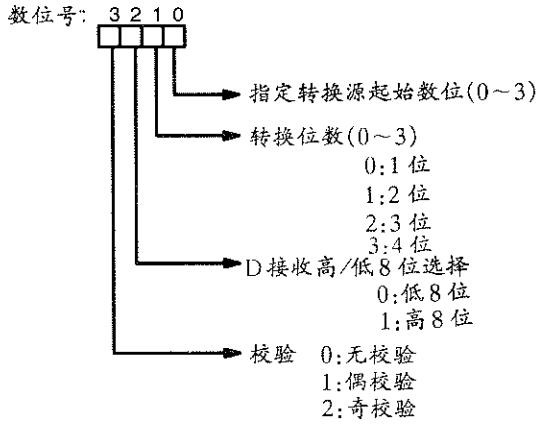
当指令执行条件为 OFF, ASC(86)指令不执行。当执行条件为 ON, ASC(86)将 S 的指定位的数转换成相应的 8—位 ASCII 码并存入从 D 开始的字内。

S 中的所有或任何位数可从指定的起始数位起按序转换。起始数位，转换位数和 D 中高 8 位还是低 8 位接受第一个 ASCII 码由 Di 内容定义。如果指定多位数转换，转换后的 ASCII 码将从 D 指定字节开始按序存放，每个码需要两数位(2×4bit)，如果指定的位数比 S 中余下多(从指定的起始数位开始数)，则以后转换的位从 S 起始位开始。

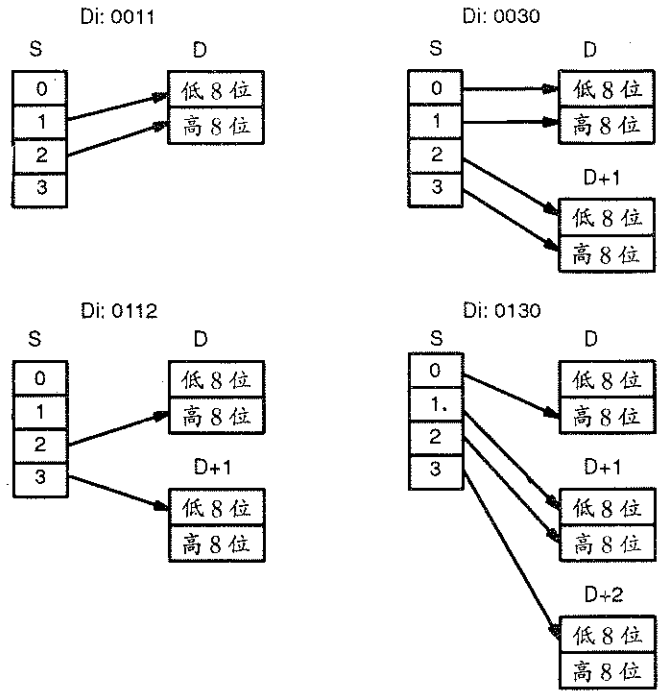
扩展 ASCII 码字符表请参见附录 G。

数指定器

Di 数设置如下所示。



以下是一些 Di 值和相应的 4 位二进制到 8 位 ASCII 转换处理的例子。



校验

每个 ASCII 码字符(2 位数)的最高位根据奇或偶校验自动调整。如果未指定校验，最高位总是为零。

当指定了偶校验，最高位会被调整，使 ON 位的总数为偶数，例如当设置成偶校验，ASCII“31”(00110001)被调整为“B1”(10110001 校验位 ON，形成偶数个 ON 位)；ASCII“36”(00110110)被调整为“36”(00110110：校验位为 OFF，因为 ON 位数已是偶数个)，校验位的状态不影响 ASCII 码的含意。

当设定奇校验，每个 ASCII 码字符的最高位会调整，形成奇数个 ON 位。

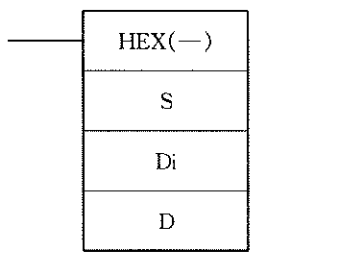
标志

ER：数指定器不正确或目标数据超出区域。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

5－18－11 ASCII 码→十六进制指令——HEX(—)

梯形符号



操作数数据区域

S：源起始字
IR，SR，AR，DM，HR，TC，LR
Di：数指定器
IR，SR，AR，DM，HR，TC，LR
D：目标字
IR，SR，AR，DM，HR，LR

限制

Di 必须在下列给定值之中。

所有源字必须在同一数据区中。

源字中的字节必须含有与 ASCII 码相应的十六进制数，即，30～39(0～9)，41～46(A～F)，或 61～66(a～f)。

说明

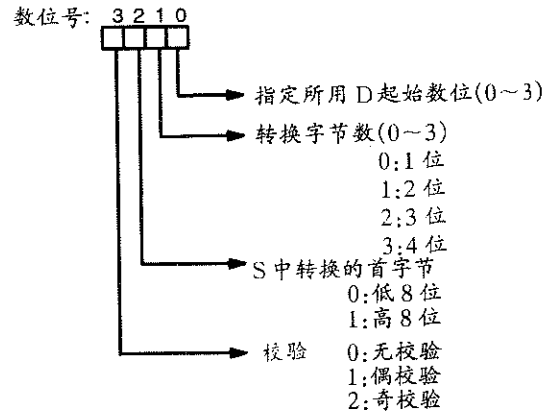
当指令执行条件为 OFF，HEX(—)不执行，当执行条件为 ON，HEX(—)将源字中指定的 ASCII 字节转换成相应的十六进制数并送入 D 中。

从 S 的起始字节起，可转换多至 4 个 ASCII 码。转换后的十六进制数从指定数位起按顺序存入 D 中，起始字节(高 8 位或低 8 位)，转换字节数，和 D 中接受第一个十六进制值的数位都由 Di 内容指定，如果指定多字节转换，必须从指定的 S 高 8 位或低 8 位开始到 S＋1，S＋2 按序转换。

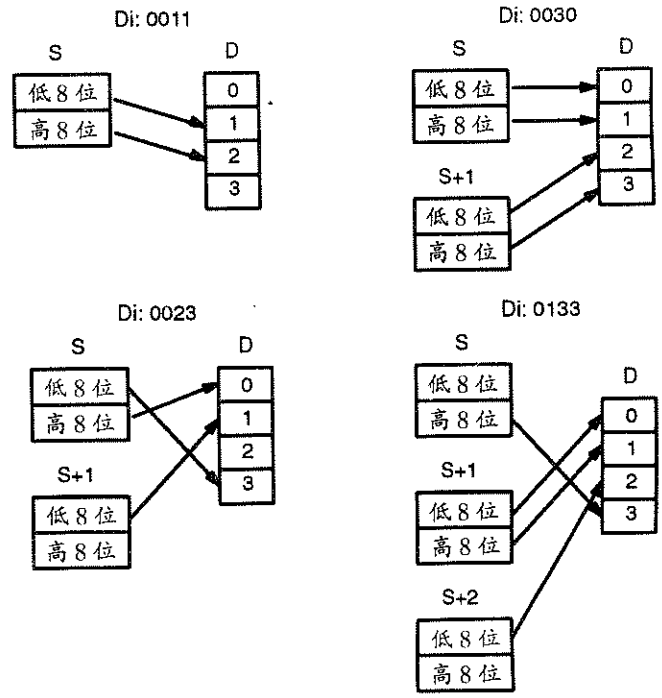
如果指定的位数比 D 中余下的多(从指定的起始数位开始数)，则以后转换的数从 D 起始位开始的。D 中未接受转换数据的数位内容不变。

数指定器

Di 的数设置如下所示。



以下是一些 Di 值和相应的 8—位 ASCII 码到 4—位十六进制转换处理的例子。



ASCII 代码表

下表给出了转换前的 ASCII 码和转换后的十六进制数值。

ASCII 字符表请参见附录 G。

原数据									转换后数据				
ASCII 码	位状态(见注)								数	位			
30	·	0	1	1	0	0	0	0	0	0	0	0	0
31	·	0	1	1	0	0	0	1	1	0	0	0	1
32	·	0	1	1	0	0	1	0	2	0	0	1	0
33	·	0	1	1	0	0	1	1	3	0	0	1	1
34	·	0	1	1	0	1	0	0	4	0	1	0	0
35	·	0	1	1	0	1	0	1	5	0	1	0	1
36	·	0	1	1	0	1	1	0	6	0	1	1	0
37	·	0	1	1	0	1	1	1	7	0	1	1	1
38	·	0	1	1	1	0	0	0	8	1	0	0	0
39	·	0	1	1	1	0	0	1	9	1	0	0	1
41	·	1	0	1	0	0	0	1	A	1	0	1	0
42	·	1	0	1	0	0	1	0	B	1	0	1	1
43	·	1	0	1	0	0	1	1	C	1	1	0	0
44	·	1	0	1	0	1	0	0	D	1	1	0	1
45	·	1	0	1	0	1	0	1	E	1	1	1	0
46	·	1	0	1	0	1	1	0	F	1	1	1	1

注

每一 ASCII 码的最高位根据检验调整。

校验

每个 ASCII 码字符(2 位数)的最高位根据奇或偶校验自动调整。

如果未指定校验,最高位总是零。当指定奇或偶校验,每个 ASCII 码字符的最高位会被调整,形成奇数或偶数个 ON 位。

如果 S 中的 ASCII 码的校验结果与 Di 中所定义的校验不符,则 ER 标志(SR25503)置为 ON,指令不被执行。

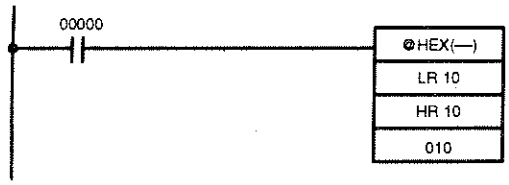
标志

ER:数指示器不正确或目标数据超出区域。

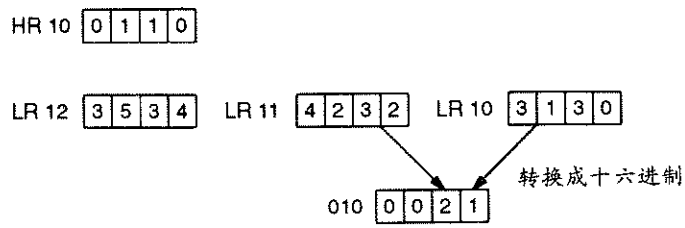
间接寻址 DM 字不存在。(* DM 字不是 BCD 码或超出 DM 区域)

例

下例中,LR10 的第二字节和 LR11 的第一字节转换成十六进制数,并将转换的数写入 IR010 的第一和第二位数中。

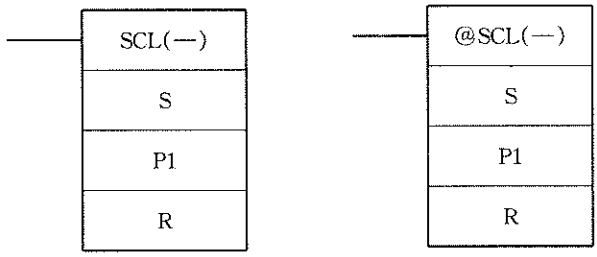


地址	指令	操作数
00000	LD	00000
00001	@HEX(—)	
		LR 10
		HR 10
		010



5－18－12 线性比例指令——SCL(—)

梯形符号



操作数数据区域

S:源字
IR, SR, AR, DM, HR, TC, LR, #

P1:参数起始字
IR, SR, AR, DM, HR, TC, LR

R:结果字
IR, SR, AR, DM, HR, LR

限制

P1 和 P1 + 2 必须是 BCD 数据。

P1 至 P1＋3 必须在同一数据区。

P1＋1 和 P1＋3 不能设置成相同数值。

说明

SCL(一)指令用于把 4 位十六进制数线性转换为 4 位 BCD 码数。

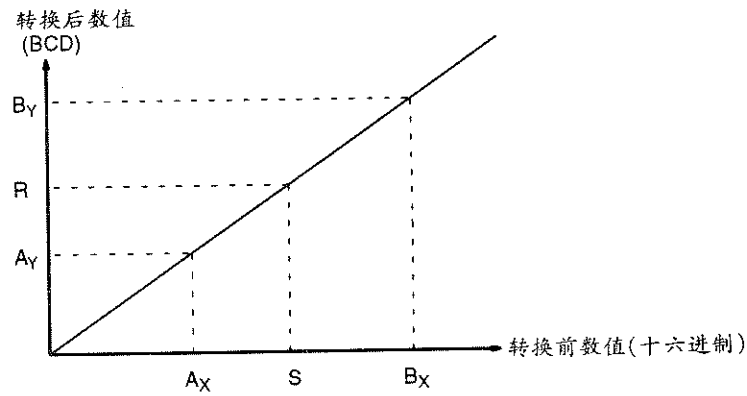
不象 BCD(24)指令是将 4 位十六进制数转换成等值的 4 位 BCD 数($S_{hex}\rightarrow S_{BCD}$)，SCL(一)可根据定义的线性关系转换十六进制数值。线性转换关系由 P1 到 P1＋3 参数字指定的两点所定义。

当指令执行条件为 OFF，SCL(一)指令不执行。当执行条件为 ON，SCL(一)将 S 中的 4 位十六进制数值根据(P1，P1＋1)和(P1＋2，P1＋3)定义的线性关系，转换成 4 位 BCD 数据，并将转换结果存入 R 中。结果按四舍五入取整。如果结果小于零，那么 0000 写入 R，如果结果大于 9999，9999 写入 R。

下表给出了参数的范围和功能：

参数	功能	范围	注释
P1	BCD 点＃1(A_Y)	0000～9999	…
P1＋1	十六进制.点＃1(A_X)	0000～FFFF	不可设置 P1＋1＝P1＋3
P1＋2	BCD 点＃2(B_Y)	0000～9999	…
P1＋3	十六进制. 点＃2(B_X)	0000～FFFF	不可设置 P1＋3＝P1＋1

下图给出了源字 S 根据(A_Y，A_X)和(B_Y，B_X)点定义的线性关系转换至 R 的直线图。



计算时，首先将所有数据转换成 BCD 码，然后用下面计算公式计算结果。

结果＝By－[(B_y－A_y)/(B_X－A_X)＊(B_X－S)]

标志

ER：P1＋1 数值等于 P1＋3。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

P1 和 P1＋3 设置错误或不在同一数据区。

EQ：当结果 R 是 0000，EQ 为 ON。

例

当 00000 为 ON，下例中 DM0100(＃0100)中的 BCD 根据 DM150 到 DM153 中的参数转换成十六进制数据，然后把转换结果(＃0512)写入 DM0200。



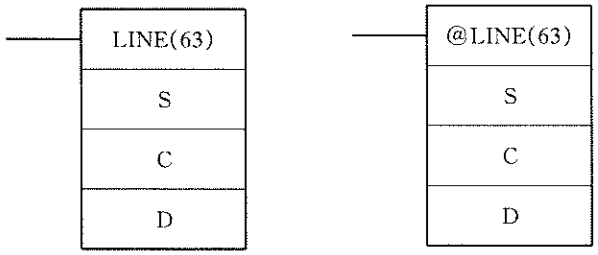
地址	指令	操作数
00000	LD	00000
00001	@SCL(—)	
		DM 0100
		DM 0150
		DM 0200

DM0150	0010
DM0151	0005
DM0152	0050
DM0153	0019

DM0100	0100
↓	
DM0200	0512

5－18－13 列→行转换指令——LINE(63)

梯形符号



操作数数据区域

S:16 字源组的起始字
IR, SR, AR, DM, HR, TC, LR
C:列位指定器(BCD 码)
IR, SR, AR, DM, HR, TC, LR, #
D:目标字
IR, SR, AR, DM, HR, TC, LR

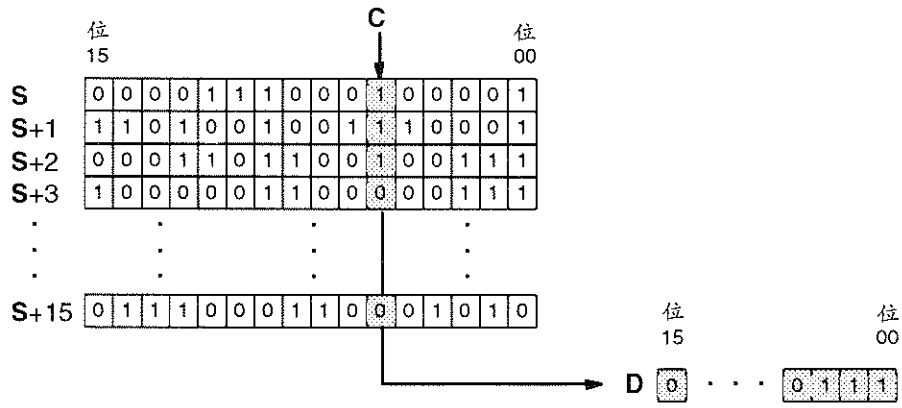
限制

S 和 S+15 必须在同一数据。

C 必须在 #0000 和 #0015 之间。

说明

当指令执行条件为 OFF, LINE(63)指令不执行, 当执行条件为 ON, LINE(63)将 16 个字组(S～S+15)中 C 列位拷贝到 D 的 16 个位(00～15)中。



标志

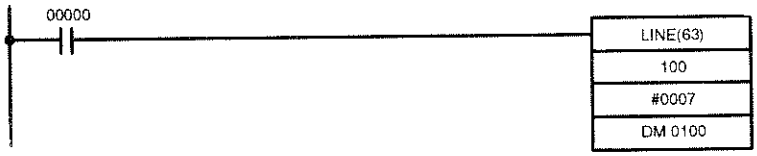
ER:列位指定器 C 不是 BCD 码或者指定位不存在。(即, 指定位必须是在 00～15 之间)

间接寻址 DM 字不存在。(＊DM 字不是 BCD 码或超出 DM 区域)

EQ:D 的内容为零,EQ 为 ON,反之为 OFF。

例

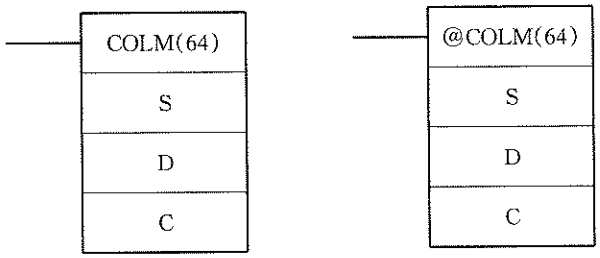
下例表明如何用 LINE(63)将设定的(IR100～IR115)中的 7 列位传送到 DM0100 中去。



地址	指令	操作数
00000	LD	00000
00001	LINE(63)	
		100
		# 0007
		DM 0100

5－18－14 行→列转换指令－COLM(64)

梯形符号



操作数数据区域

S:源字
IR, SR, AR, DM, HR, TC, LR

D:目标组起始字
IR, AR, DM, HR, TC, LR

C:列位指定器(BCD 码)
IR, SR, AR, DM, HR, TC, LR, #

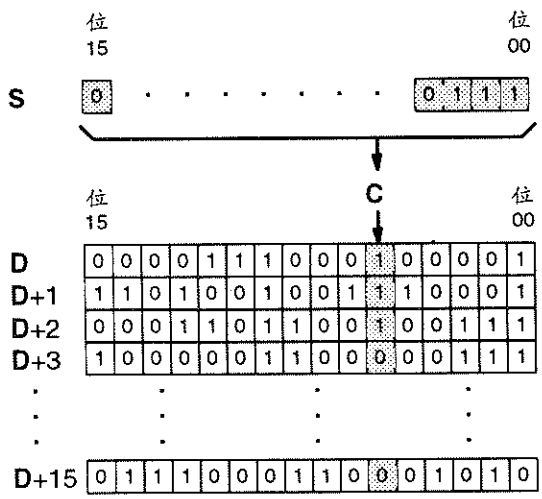
限制

D 和 D＋15 必须在同一数据区。

C 必须在＃0000 到＃0015 之间。

说明

当指令执行条件为 OFF,COLM(64)不执行。当执行条件为 ON,COLM(64)将字 S 的 16 位(00～15)拷贝到 16 个字组(D 到 D＋15)的第 C 列位中。



标志

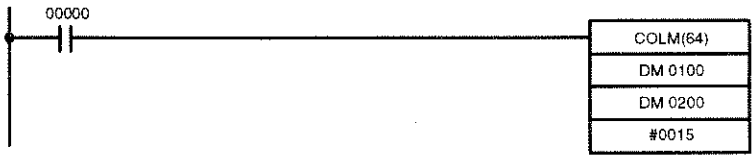
ER:列位指定器 C 不是 BCD 码或者指定列位不存在。(即,指定列位必须在 00～15 之间)

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:当 S 内容是零,EQ 为 ON;反之为 OFF。

例

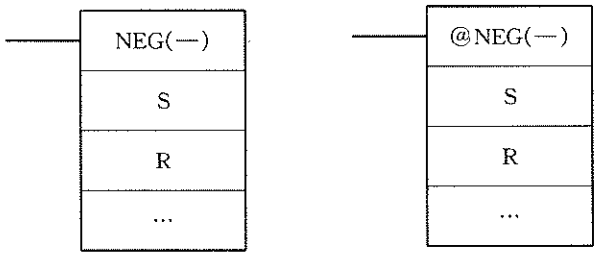
下例表明如何用 COLM(64)将 DM0100 的内容(00～15)传送到设定的字组(DM0200～DM0215)15 列位中。



地址	指令	操作数
00000	LD	00000
00001	COLM(64)	
		DM 0100
		DM 0200
		# 0015

5－18－15 二进制求补指令——NEG(－)

梯形符号



操作数数据区域

S:源字
IR, SR, AR, DM, HR, TC, LR, #

R:结果字
IR, SR, AR, DM, HR, LR

说明

源字 S 的 4 位十六进制内容转换成它的二进制补码,并将结果输出到结果字(R)中。指令的操作与 0000 减 S 内容再输出到 R 中的作用完全一样。

如 S 的内容是 0000,指令执行后 R 的内容也是 0000,而 EQ(SR25506)变为 ON。

如果 S 的内容是 8000,指令执行后 R 的内容也是 8000,而 UF(SR25405)变为 ON。

注

有关 16 位带符号二进制数详见 25 页。

标志

ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

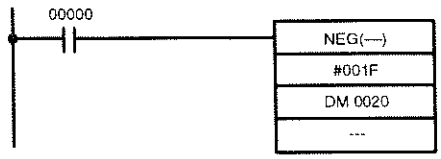
EQ:当 S 内容是 0000,EQ 为 ON;反之为 OFF。

UF:当 S 的内容是 8000,UF 为 ON;反之为 OFF。

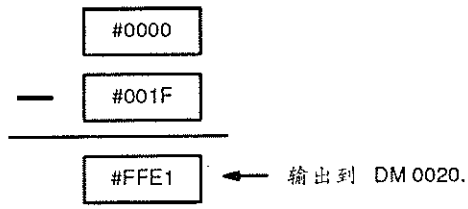
N:当 R 中 15 位置“1”,N 为 ON,反之为 OFF。

例

下例表明如何用 NEG(－)得到十六进制数 001F 的二进制补码并将结果输出到 DM0020。

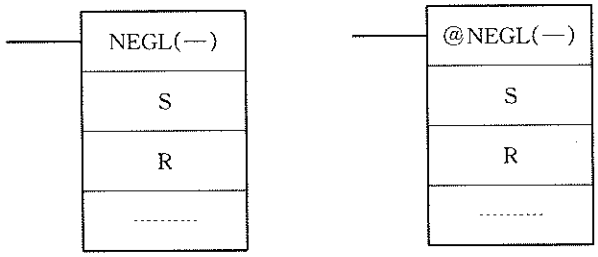


地址	指令	操作数
00000	LD	00000
00001	NEG(—)	
		# 001F
		DM 0020



5－18－16 双字二进制求补指令—NEGL(—)

梯形符号



操作数数据区域

S:源起始字
IR, SR, AR, DM, HR, TC, LR

R:结果起始字
IR, SR, AR, DM, HR, LR

限制

S 和 S+1 必须在同一数据区，R 和 R+1 也必须在同一数据区。

说明

源字(S 和 S+1)的 8 位十六进制内容转换成它的二进制补码并将结果输出到结果字(R 和 R+1)中。指令的操作与 00000000 减 S 和 S+1 内容再输出到 R 和 R+1 指令执行后一样。

如果 S 的内容是 0000 0000，指令执行后 R 的内容也是 0000 0000，而 EQ(SR25506)置为 ON。

如果 S 的内容是 8000 0000，指令执行后 R 的内容也是 8000 0000，而 UF(SR25405)置为 ON。

注

有关 32 位带符号二进制数据详见 25 页。

标志

ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域。)

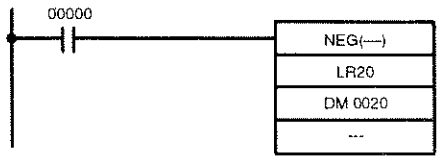
EQ:当 S+1，S 内容是 0000 0000，EQ 为 ON；反之为 OFF。

UF:当 S+1，S 内容是 8000 0000，UF 为 ON；反之为 OFF。

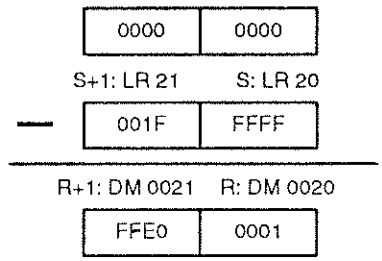
N:当 R+1 中 15 位置“1”，N 为 ON；反之为 OFF。

例

下例表明如何用 NEGL(－)得到 LR21，LR20 中十六进制数(001FFFFF)的二进制补码并将结果输出到 DM0021，DM0020 中。



地址	指令	操作数
00000	LD	00000
00001	NEGL(—)	
		LR 20
		DM 0020



5－19 BCD 码运算指令

BCD 码运算指令——INC(38), DEC(39), ADD(30), ADDL(54), SUB(31), SUBL(55), MUL(32), MULL(56), DIV(33), DIVL(57), FDIV(79)和 ROOT(72), 所有算术运算都用 BCD 数据。

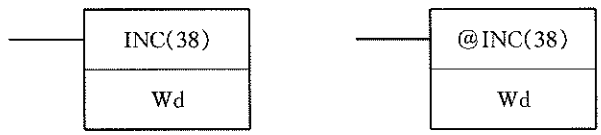
对于 INC(38)和 DEC(39)指令源字和结果字是同一个。即指令结果将源字的内容改写。所有其它指令只改变结果字的内容, 即所有其它 BCD 码运算指令源字的内容在指令执行前后不改变。

由于大多数 BCD 操作在它们的结果中使用了进位标志(CY), 所以 STC(40)和 CLC(41), 置进位和清进位标志指令也包含在这组指令中。二进制及移位操作也使用 CY。

就象结果中包含了进位位(CY), 在加法和减法指令计算中也使用了进位位(CY), 所以在计算中, 如果 CY 的原始状态不需要, 切记将 CY 清零, 并且如果需要, 在其它指令执行改变它的状态之前, 使用它。

5－19－1 增量指令－INC(38)

梯形符号



操作数数据区域

Wd:增量字(BCD 码)
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF, INC(38)指令不执行。当执行条件为 ON, INC(38)将 Wd 加一, 标志位 CY 不变。

标志

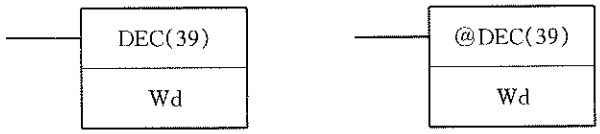
ER:Wd 不是 BCD 码。

间接寻址 DM 字不存在。(* DM 字内容不是 BCD 码或超出 DM 区域。)

EQ:当增加后数值是零, EQ 为 ON。

5－19－2 减量指令－DEC(39)

梯形符号



操作数数据区域

Wd：减量字(BCD 码)
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF，DEC(39)指令不执行。当执行条件为 ON，DEC(39)将 Wd 减一，标志位 CY 不变。DEC(39)指令操作和 INC(38)相同，只是数值减一而不是加一。

标志

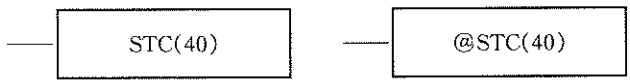
ER：Wd 不是 BCD 码。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域")

EQ：当减值结果是零，EQ 为 ON。

5－19－3 置进位标志指令——STC(40)

梯形符号



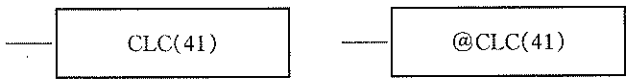
当指令执行条件为 OFF，STC(40)指令不执行。当执行条件为 ON，STC(40)将 CY(SR25504)置为 ON 状态。

注

影响 CY 的指令列表请参见附录 C"算术运算的标志和出错信息"。

5－19－4 清进位标志指令——CLC(41)

梯形符号



当指令执行条件为 OFF，CLC(41)不执行。当执行条件为 ON，CLC(41)将 CY(SR25504)变为 OFF 状态。

清标志用于复位(置为 OFF)CY(SR25504)为"0"状态。

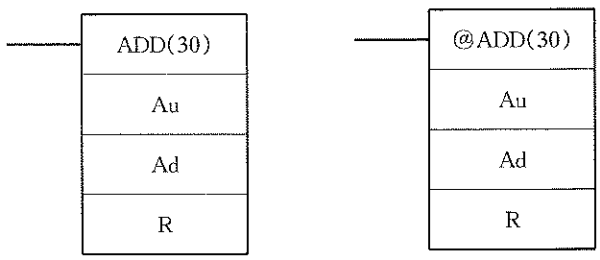
当结果指令 END(01)在每个循环周期结束执行时，CY 自动复位到"0"状态。

注

影响 CY 的指令列表请参见附录 C"算术运算标志和出错信息"。

5－19－5 BCD 码加法指令——ADD(30)

梯形符号



操作数数据区域

Au：被加数字(BCD)
IR, SR, AR, DM, HR, TC, LR, #

Ad：加数字(BCD)
IR, SR, AR, DM, HR, TC, LR, #

R：结果字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF，ADD(30)指令不执行。当执行条件为 ON，ADD(30)将 Au，Ad 内容和 CY 相加，并将和存入 R。如果和大于 9999，CY 置为 ON。

Au + Ad + CY → CY　R

标志

ER：Au 和/或 Ad 不是 BCD 码。

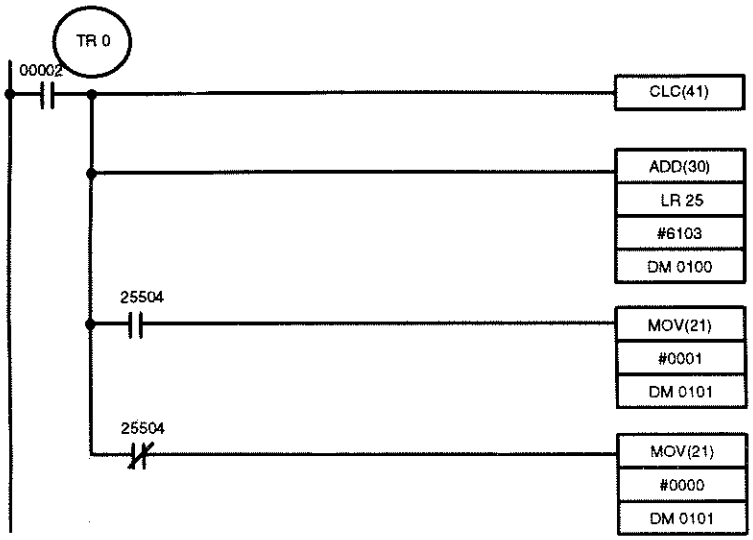
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

CY：当结果有进位，CY 为 ON。

EQ：当结果是零，EQ 为 ON。

例

如果 00002 为 ON，下面的梯形图程序采用 CLC(41)将 CY 复位，然后把 LR25 的内容与常数(6103)相加，并将结果存入 DM0100，然后根据 CY(25504)状态将 0000 或 0001 传送到 DM0101。保证了所得的数据的进位标志保存在 R＋1 中，这样运算结果以后可按 8 位数处理。

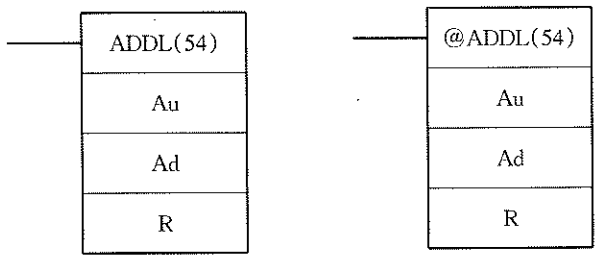


地址	指令	操作数	
00000	LR		00002
00001	OUT	TR	0
00002	CLC(41)		
00003	AND(30)		
		LR	25
		#	6103
		DM	0100
00004	AND		25504
00005	MOV(21)		
		#	0001
		DM	0101
00006	LD	TR	0
00007	AND NOT		25504
00008	MOV(21)		
		#	0000
		DM	0101

虽然两条 ADD(30)指令可一起使用以完成 8 位 BCD 码加法运算，但 ADDL(54)双字 BCD 码加法指令是专门用于双字节加法运算的。

5－19－6　双字 BCD 码加法指令——ADDL(54)

梯形符号



操作数数据区域

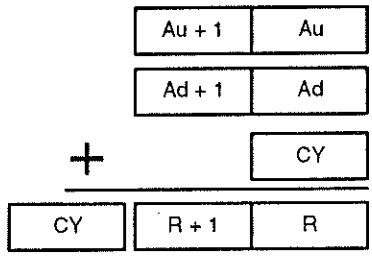
Au：被加数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR
Ad：加数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR
R：结果起始字
IR，SR，AR，DM，HR，LR

限制

Au 和 Au＋1，Ad 和 Ad＋1，R 和 R＋1 每对数据必须在同一数据区。

说明

当指令执行条件为 OFF，ADDL(54)不执行。当执行条件为 ON，ADDL(54)将 CY 内容，Au 和 Au＋1 中的 8 位被加数，Ad 和 Ad＋1 中的 8 位加数相加，并将结果存入 R 和 R＋1 中。如果相加和大于 99999999，CY 为 ON。



标志

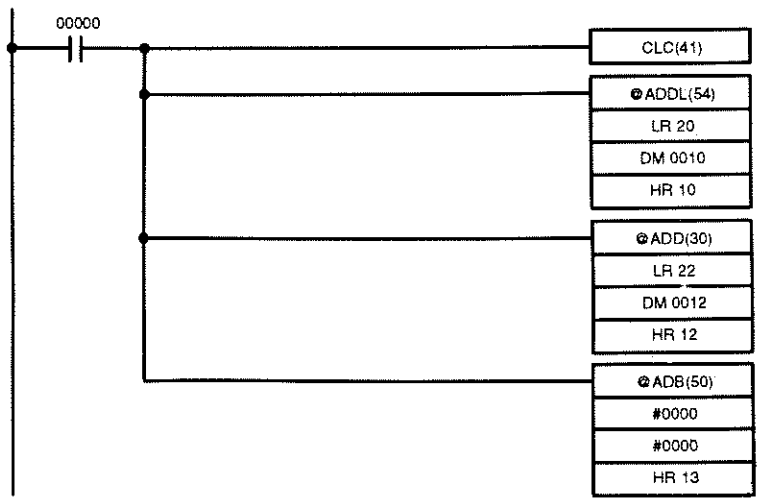
ER： Au 与/或 Ad 不是 BCD 码。
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或者超出 DM 区域。）

CY： 当结果中有进位，CY 为 ON。

EQ： 当结果是零，EQ 为 ON。

例

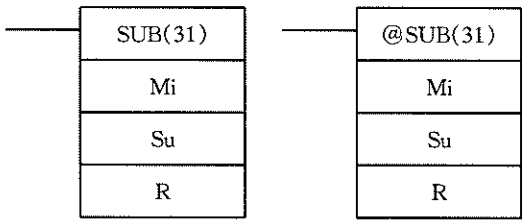
当 00000 为 ON，下面程序将两个 12 位数相加，第一个数在 LR20 到 LR22 中，第二个数在 DM0010 到 DM0012 中。计算结果存入 HR10 到 HR13 中。在第二条加法指令 ADD(30)运算时，第一条加法指令运算的进位也包含在运算中。微分型指令@ADB(50)用两个零相加（见 5－20－1“二进制加法指令——ADB(50)”）将第二条加法指令的进位位 CY 间接存入 HR13。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	@ADDL(54)	
		LR 20
		DM 0010
		HR 10
00003	@ADD(30)	
		LR 22
		DM 0012
		HR 12
00004	@ADB(50)	
		# 0000
		# 0000
		HR 13

5－19－7 BCD 码减法指令——SUB(31)

梯形符号



操作数数据区域

Mi：被减数字(BCD)
IR, SR, AR, DM, HR, TC, LR, #

Su：减数字(BCD)
IR, SR, AR, DM, HR, TC, LR, #

R：结果字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF，SUB(31)指令不执行。当执行条件为 ON，SUB(31)将 Mi 内容减去 Su 内容和 CY，并将结果存入 R。如果结果为负，CY 为 ON 且实际结果以 10 进制补码存入 R。为了将这 10 进制补码变成真值，用零减 R 的内容(见下例)。

Mi － Su － CY → CY　　R

注

二进制求补—NEG(－)指令只能用于转换二进制数，此指令不能用于 BCD 码数据求补。

标志

ER：　Mi 和/或 Su 不是 BCD 码。
　　间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

CY：　当运算结果是负值，CY 为 ON，即，当 Mi 小于 Su 加 CY。

EQ：　当运算结果是零，EQ 为 ON。

注意

如果 CY 先前的状态不需要，执行 SUB(31)指令前务必用 CLC(41)清除进位标志，执行 SUB(31)减法指令后检查 CY 状态。如果执行 SUB(31)后 CY 为 ON(即，如果减的结果是负值)，输出结果是真实结果的十进制补码形式。要获得真值，则采用零减 R 中数值方法。

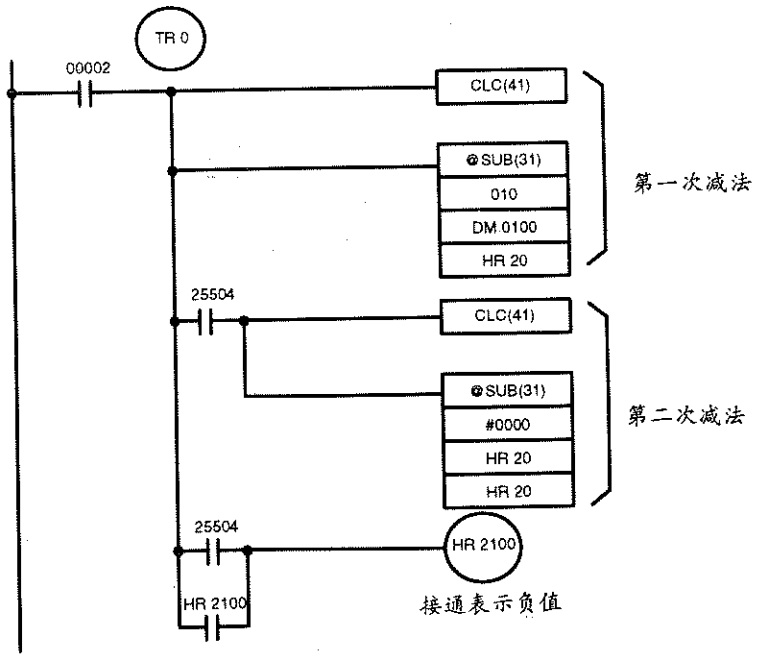
例

当 00002 为 ON，下面梯形图程序先清进位位 CY，后从 010 的内容中减去 DM0100 的内容和 CY，并将结果存入 HR20 中 。

如果执行 SUB(31)后 CY 为 ON，从零中减去 HR20 中结果(注意再需要使用 CLC(41)指令，以得到正确结果)，将差放回 HR20，置 HR2100 为 ON，表示结果为负值。

如果执行 SUB(31)后 CY 为 OFF，则表示结果为正值，第二次减法不执行，HR2100 不置为 ON。HR2100 编成保持位程序形式，所以当程序再次循环执行时 CY 状态改变就不能使 HR2100 变为 OFF。

例子中，采用 SUB(31)微分型指令，所以减法操作在每次 00002 变为 ON 时执行一次。当进行另一次减法操作时，00002 至少需要在一个循环中变为 OFF，(HR2100 复位)然后再变为 ON。



地址	指令	操作数
00000	LD	00002
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUB(31)	
		010
		DM 0100
		HR 20
00004	AND	25504
00005	CLC(41)	
00006	@SUB(31)	
		# 0000
		HR 20
		HR 20
00007	LD	TR 0
00008	AND	25504
00009	OR	HR 2100
00010	OUT	HR 2100

这个图中的第一和第二次减法运算由下面 IR010 和 DM0100 数据举例说明。

注

SUB(31)实际操作包括从 10,000 中减去 Su 和 CY，然后再加 Mi，结果为正值，最高位就截去；结果为负值，获得的是十进制补码。获得正确结果的处理过程如下所示。

第一次减法

IR 010 1029

DM0100 －3452

CY －0

HR20 7577 (1029＋(100001－3452))

CY 1 (负值结果)

第二次减法

0000

HR 20 －7577

CY －0 (0000＋(10000－7577))

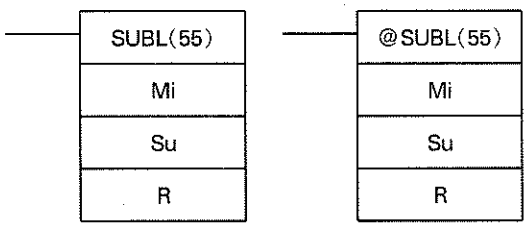
HR 20 2423 (负值结果)

CY 1

在上面情况中，程序使 HR2100 变为 ON，以表示 HR20 中的数据是负值。

5－19－8 双字 BCD 码减法指令——SUBL(55)

梯形符号



操作数据区域

Mi：被减数起始字(BCD)
IR, SR, AR, DM, HR, TC, LR

Su：减数起始字(BCD)
IR, SR, AR, DM, HR, TC, LR

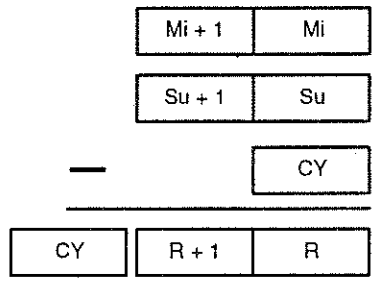
R：结果起始字
IR, SR, AR, DM, HR, LR

限制

Mi 和 Mi＋1，Su 和 Su＋1，R 和 R＋1 每一对数据必须在同一数据区。

说明

当指令执行条件为 OFF，SUBL(55)指令不执行。当执行条件为 ON，SUBL(55)从 Mi＋1 和 Mi 的 8 位数值中减去 Su＋1 和 Su 的 8 位数值及 CY，并将结果存入 R＋1 和 R。如果结果是负值，CY 为 ON，R 中存入实际结果的十进制补码。要将这十进制补码变为真值，采用零减 R 中内容的方法。由于 8 位常数不能直接进入运算，请采用 BSET(71)指令(参见 5－16－3“块设置—BSET(71)”)产生一个 8 位常数。



注

双字二进制求补指令—NEGL(—)只能用于二进制转换，不能用于 BCD 码转换。

标志

ER： Mi，Mi＋1，Su 或 Su＋1 不是 BCD 码。

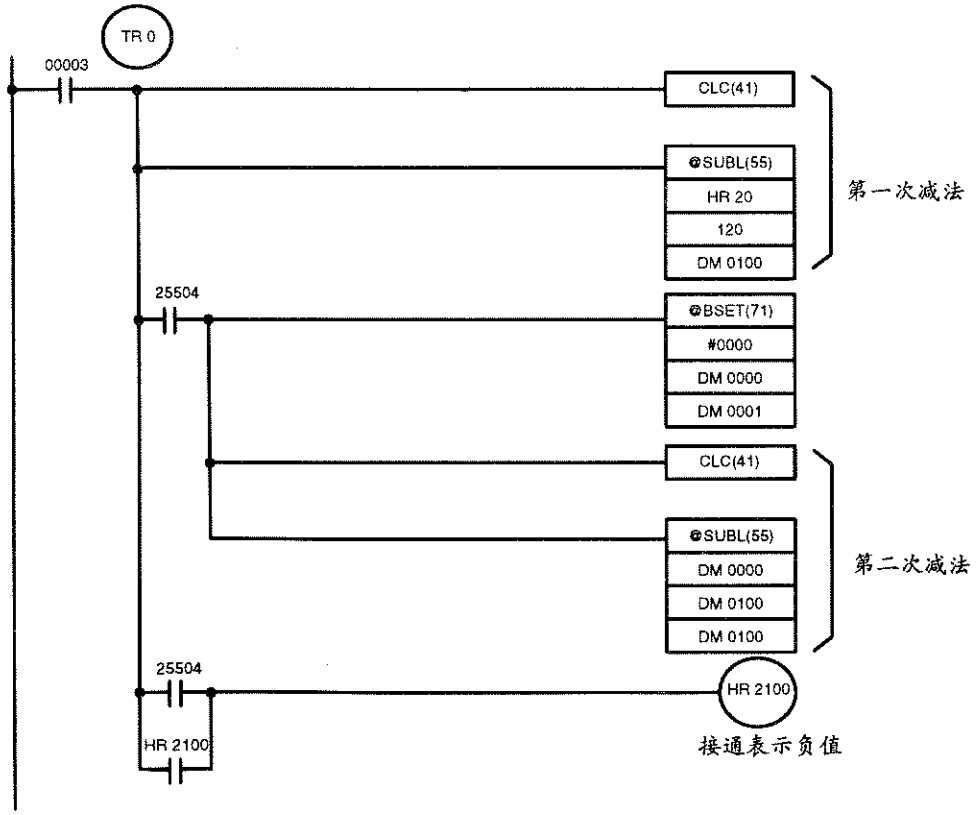
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

CY： 当运算结果为负值，CY 为 ON，即，Mi 小于 Su。

EQ： 当运算结果是零，EQ 为 ON。

下例的工作形式很象单字减法运算，然而，例子中用 BSET(71)将 DM0000 和 DM0001 清零，这样可从 0(不能直接输入 8 位常数)减去负值的结果。

例

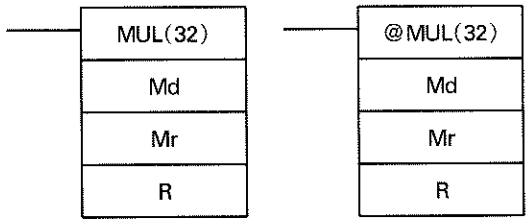


地址	指令	操作数
00000	LD	00003
00001	OUT	TR 0
00002	CLC(41)	
00003	@SUBL(55)	
		HR 20
		120
		DM 0100
00004	AND	25504
00005	@BSET(71)	
		# 0000
		DM 0000
		DM 0001

地址	指令	操作数
00006	CLC(41)	
00007	@SUBL(55)	
		DM 0000
		DM 0100
		DM 0100
00008	LD	TR 0
00009	AND	25504
00010	OR	HR 2100
00011	OUT	HR 2100

5－19－9 BCD 码乘法指令——MUL(32)

梯形符号



操作数据区域

Md：被乘数(BCD)
IR, SR, AR, DM, HR, TC, LR, #

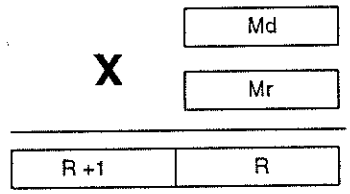
Mr：乘数(BCD)
IR, SR, AR, DM, HR, TC, LR, #
R：结果起始字
IR, SR, AR, DM, HR, LR

限制

R 和 R＋1 必须在同一数据区。

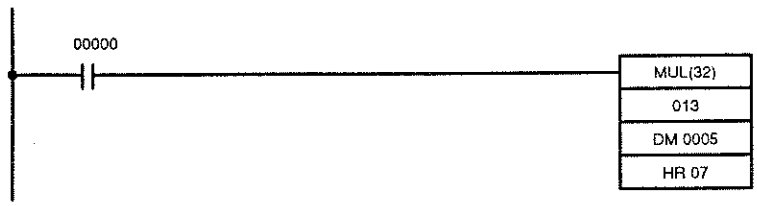
说明

当指令执行条件为 OFF，MUL(32)指令不执行。当执行条件为 ON，MUL(32)将 Mr 内容与 Md 相乘，并把结果存入 R 和 R＋1。

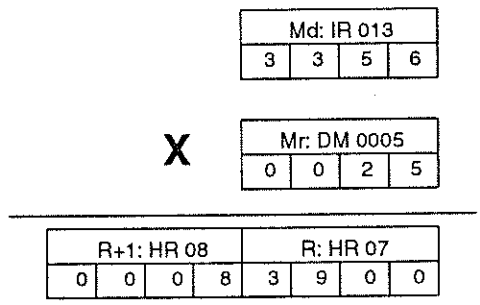


例

下面程序中，当 IR00000 为 ON，IR013 和 DM0005 内容相乘并将结果存入 HR07 和 HR08。举例的数据和运算如下面所示。



地址	指令	操作数
00000	LD	00000
00001	MUL(32)	
		013
		DM 00005
		HR 07



标志

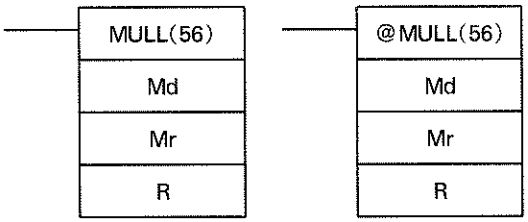
ER： Md 和/或 Mr 不是 BCD 码。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ： 当运算结果是零，EQ 为 ON。

5－19－10 双字 BCD 码乘法指令——MULL(56)

梯形符号



操作数据区域

Md：被乘数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR

Mr：乘数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR

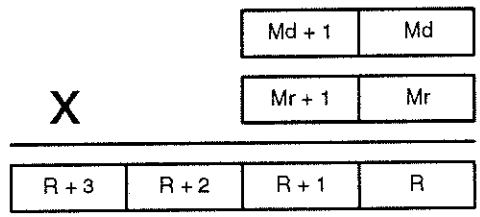
R：结果起始字
IR，SR，AR，DM，HR，LR

限制

Md 和 Md＋1 必须在同一数据区，Mr 和 Mr＋1 也必须在同一数据区，R 至 R＋3 必须在同一数据区。

说明

当指令执行条件为 OFF，MULL(56)不执行，当执行条件为 ON，MULL(56)将 Md 和 Md＋1 的 8 位数内容与 Mr 和 Mr＋1 的内容相乘，并将结果存入 R 至 R＋3。



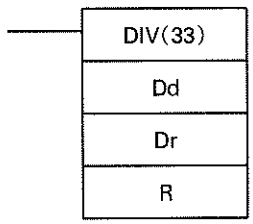
标志

ER： Md，Md＋1，Mr 或 Mr＋1 不是 BCD 码。
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ： 当运算结果是零，EQ 为 ON。

5－19－11 BCD 码除法指令——DIV(33)

梯形符号



操作数数据区域

Dd：被除数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR，＃

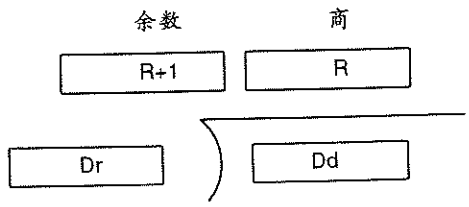
Dr：除数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR，＃
R：结果起始字(BCD)
IR，SR，AR，DM，HR，LR

限制

R 和 R＋1 必须在同一数据区。

说明

当指令执行为 OFF，DIV(33)指令不执行，程序执行下一条指令。当执条件为 ON，Dd 除以 Dr 并将结果存入 R 和 R＋1：商在 R 中，余数在 R＋1 中。



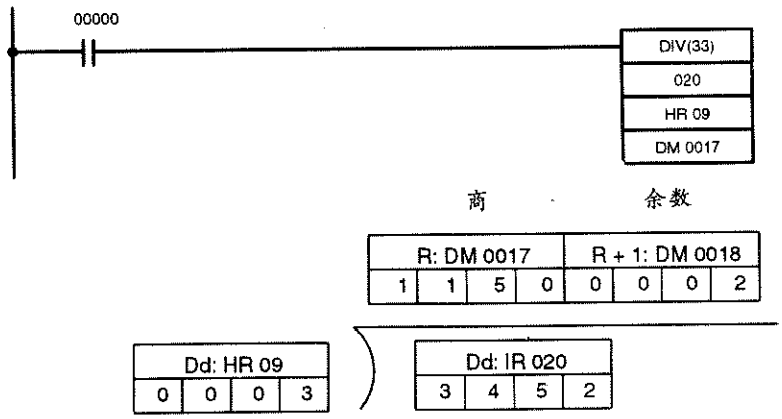
标志

ER： Dd 或 Dr 不是 BCD 码，或 Dr 为＃0000。
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ： 当运算结果是零，EQ 为 ON。

例

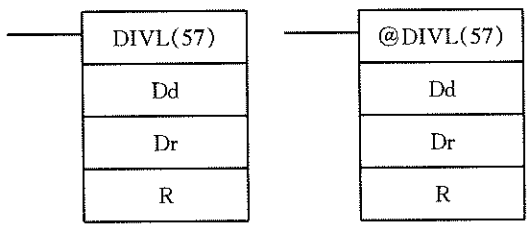
当下面程序 IR00000 为 ON，IR020 内容除以 HR09 的内容，结果存入 DM0017 和 DM0018 中。例示数据和计算示于程序下方。



地址	指令	操作数
00000	LD	00000
00001	DIV(33)	
		020
		HR 09
		DM 0017

5－19－12 双字 BCD 码除法指令——DIVL(57)

梯形符号



操作数据区域

Dd：被除数字(BCD)
IR, SR, AR, DM, HR, TC, LR

Dr：除数字(BCD)
IR, SR, AR, DM, HR, TC, LR

R：结果字
IR, SR, AR, DM, HR, LR

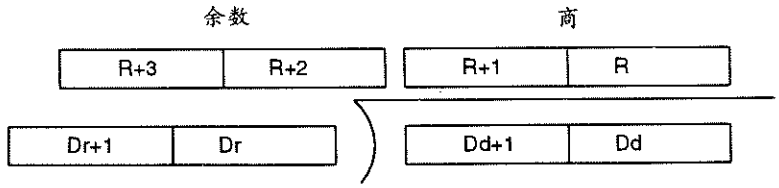
限制

Dd 和 Dd＋1 必须在同一数据区，Dr 和 Dr＋1 也必须在同一数据区。

R 至 R＋3 必须在同一数据区。

说明

当指令执行条件为 OFF，DIVL(57)指令不执行。当执行条件为 ON，DIVL(57)将 Dd 和 Dd＋1 的 8 位数内容除以 Dr 和 Dr＋1 的内容，并将商存入 R 和 R＋1，余数数存入 R＋2 和 R＋3。



标志

ER： Dr 和 Dr＋1 内容是零。

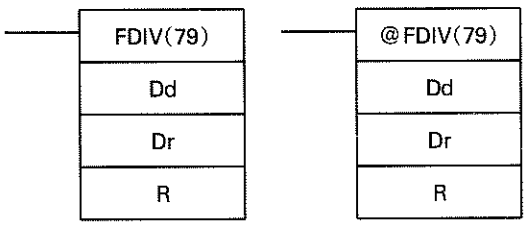
Dd，Dd＋1，Dr 或 Dr＋1 不是 BCD 码。

间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ： 当运算结果是零，EQ 为 ON。

5－19－13 浮点数除法指令——FDIV(79)

梯形符号



操作数据区域

Dd：被除数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR

Dr：除数起始字(BCD)
IR，SR，AR，DM，HR，TC，LR

R：结果起始字
IR，SR，AR，DM，HR，LR

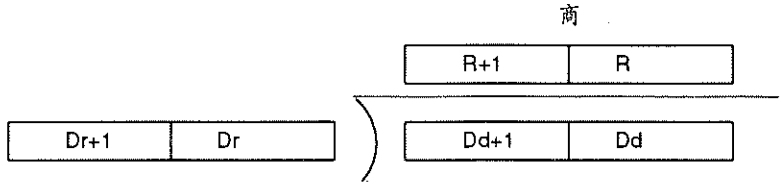
限制

Dr 和 Dr＋1 内容不为零。Dr 和 Dr＋1 必须在同一数据区，Dd 和 Dd＋1；R 和 R＋1 也必须在同一数据区。

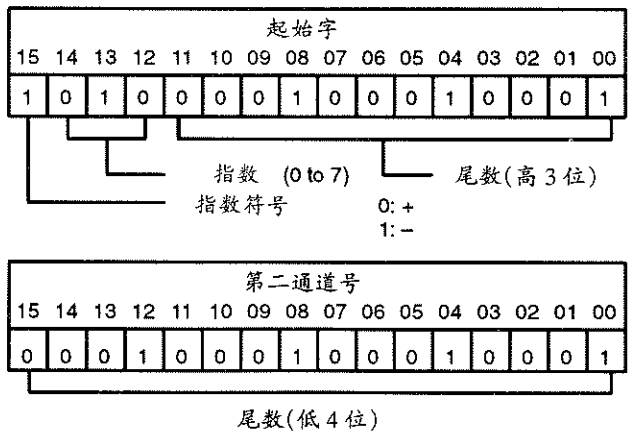
被除数和除数必须在 0.0000001×10^{-7} 至 0.9999999×10^{7} 之间。结果必须在 0.1×10^{-7} 至 0.9999999×10^{7} 之间。

说明

当指令执行条件为 OFF，FDIV(79)指令不执行。当执行条件为 ON，FDIV(79)将 Dd 和 Dd＋1 中的浮点数除以 Dr 和 Dr＋1 中的浮点数并将运算结果存入 R 和 R＋1。



如下所示，为了表达浮点数，BCD 码的低 7 位表示成尾数，最高位表示指数。尾数表示为一个小于 1 的数值，即，至小数点第七位。



$= 0.1111111 \times 10^{-2}$

标志

ER：Dr 和 Dr＋1 内容为零。

Dd，Dd＋1，Dr 或 Dr＋1 不是 BCD 码。

运算结果不在 0.1×10^{-7} 和 0.999999×10^{7} 数值之间。

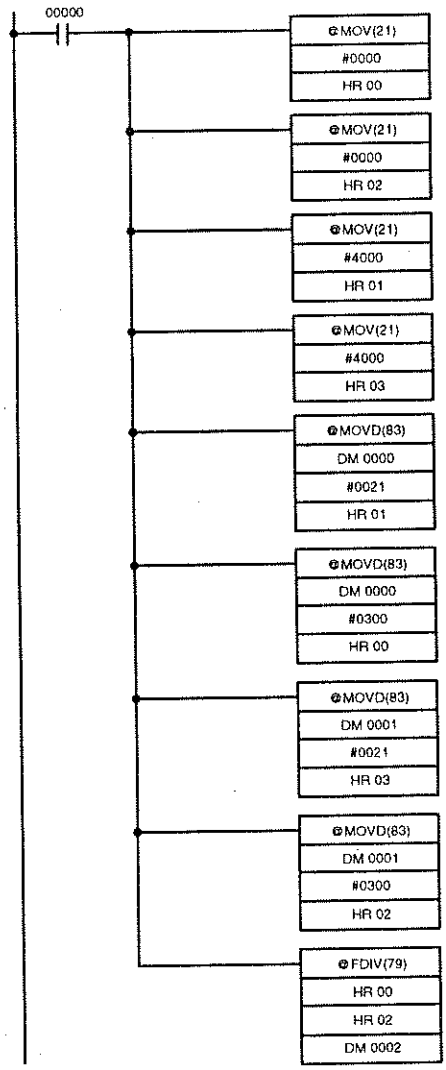
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

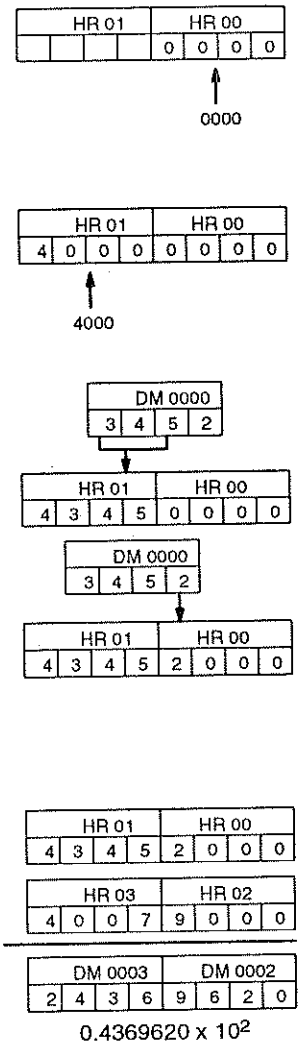
EQ： 当运算结果是零，EQ 为 ON。

例

下例给出如何进行两个四位整数的除法（即，无分数），获得浮点结果。

首先原始数必须转换成浮点数形式。因为原始数据没有小数点所以指数是 4（例如：$3452 = 0.3452\times10^{4}$）。包括指数和零在内，所有移动是将相应的数据放到连续字内用于最终的除法。Dd 和 Dd＋1 的数据移动见下面。Dr 和 Dr＋1 的移动和被除数基本相同。被除的原始数据在 DM0000 和 DM0001 中，最后的除法也已给出。



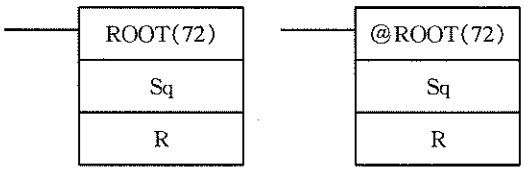


地址	指令	操作数
00000	LD	00000
00001	@MOV(21)	
		# 0000
		HR 00
00002	@MOV(21)	
		# 0000
		HR 02
00003	@MOV(21)	
		# 4000
		HR 01
00004	@MOV(21)	
		# 4000
		HR 03
00005	@MOVD(83)	
		DM 0000
		# 0021
		HR 01

地址	指令	操作数
00006	@MOVD(83)	
		DM 0000
		# 0300
		HR 00
00007	@MOVD(83)	
		DM 0001
		# 0021
		HR 03
00008	@MOVD(83)	
		DM 0001
		# 0300
		HR 02
00009	@FDIV(79)	
		HR 00
		HR 02
		DM 0002

5－19－14 平方根指令——ROOT(72)

梯形符号



操作数数据区域

Sq：源起始字(BCD)
IR，SR，AR，DM，HR，TC，LR

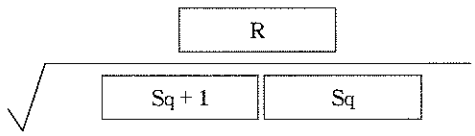
R：目标字
IR，SR，AR，DM，HR，LR

限制

Sq 和 Sq＋1 必须在同一数据区。

说明

当指令执行条件为 OFF，ROOT(72)指令不执行。当执行条件为 ON，ROOT(72)计算 Sq 和 Sq＋1 的 8 位 BCD 数的平方根。并将计算结果存入 R。小数部分舍去。



标志

ER： Sq 或 Sq＋1 不是 BCD 码。

间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

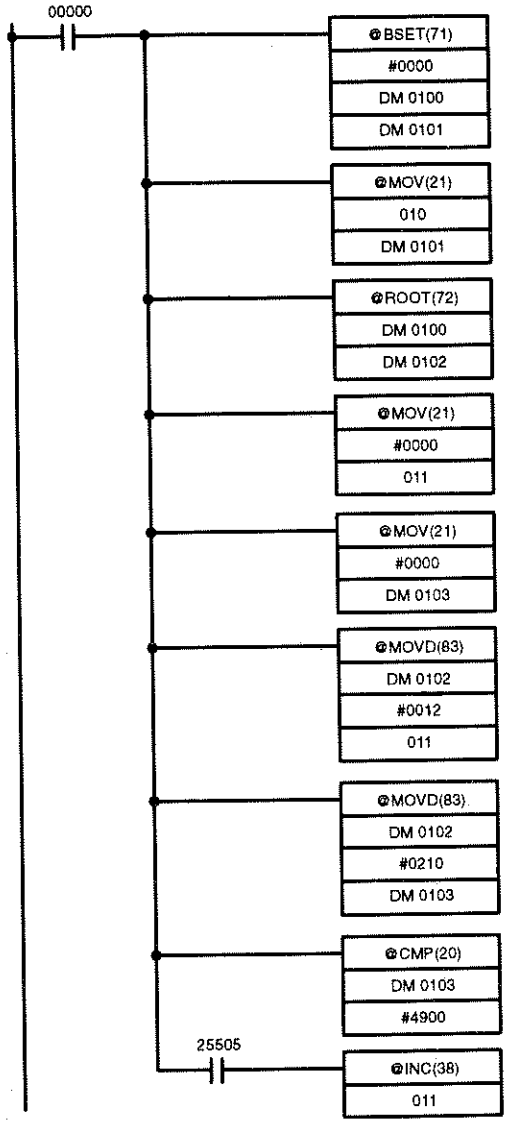
EQ： 当运算结果是零，EQ 为 ON。

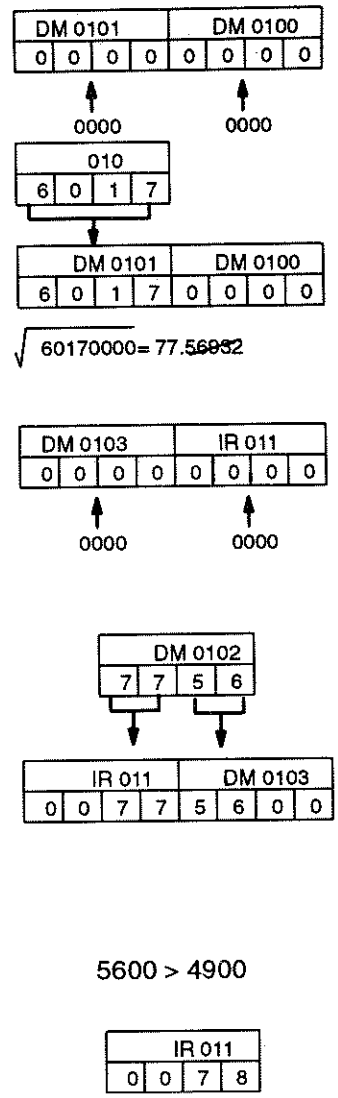
例

下例表明如何对四位 BCD 码数字开方，然后对结果作四舍五入处理。

首先所用的字清零，然后将被开方的数传送到 Sq＋1。结果的 2 倍位数(因为原始数据位数翻了一倍)存入 DM102，然后将结果数分开存入 2 个不同字内，高两位存入 IR011，低两位存入 DM103，用于对 IR011 中的结果作四舍五入修正。最后一步是比较 DM0103 中的数值，这样用大于标志使 IR011 加一。

在这个例子中，$\sqrt{6017}=77.56$，77.56 四舍五入到 78。





地址	指令	操作数
00000	LD	00000
00001	@BSET(71)	
		# 0000
		DM 0100
		DM 0101
00002	@MOV(21)	
		010
		DM 0101
00003	@ROOT(72)	
		DM 0100
		DM 0102
00004	@MOV(21)	
		# 0000
		011
00005	@MOV(21)	
		# 0000
		DM 0103

地址	指令	操作数
00006	@MOVD(83)	
		DM 0102
		# 0012
		011
00007	@MOVD(83)	
		DM 0102
		# 0210
		DM 0103
00008	@CMP(20)	
		DM 0103
		# 4900
00009	AND	25505
00010	@INC(38)	
		011

5－20 二进制运算指令

二进制运算指令—ADB(50), SBB(51), MLB(52), DVB(53), ADBL (—), SBBL (—), MBS (—), MBSL (—), DBS(—), 和 DBSL (—) 是以十六进制数据形式完成算术操作。

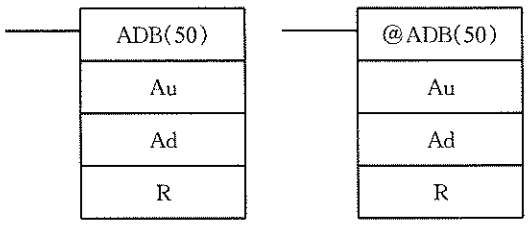
这些指令中的四条(ADB(50), SBB(51), ADBL (—), 和 SBBL (—))可处理一般不带符号和带符号数据,(MLB(52)和 DVB(53))两条指令只能处理一般不带符号数据,(MBS(—), MBSL (—), DBS(—), 和 DBSL (—))四条指令只能处理带符号二进制数据。

加法和减法在运算和结果中包括进位位 CY。如果 CY 的先前状态在运算中不需要,务必将 CY 复位,如果需要,则在 CY 被其它指令运行改变状态之前使用它。控制 CY 状态可用 STC (40) 和 CTC (41) 指令。参见 5－19"BCD 码运算指令"章节。

带符号二进制加 法和减法指令使用下溢和上溢标志(UF 和 OF)表示运算结果是否超出 16 位或 32 位带符号二进制数据的允许范围。有关带符号二进制数据内容详见 30 页。

5－20－1 二进制加法指令－ADB(50)

梯形符号



操作数数据区域

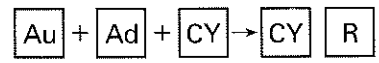
Au:被加数字(二进制)
IR, SR, AR, DM, HR, TC, LR, #

Ad:加数字(二进制)
IR, SR, AR, DM, HR, TC, LR, #

R:结果字
IR, SR, AR, DM, HR, LR

说明

当指令执行条件为 OFF, ADB(50)指令不执行,当执行条件为 ON, ADB(50)将 Au, Ad 内容和 CY 相加,并将结果存入 R。如果运算结果大于 FFFF, CY 为 ON。



ADB(50)也可用于带符号二进制数据相加。下溢和上溢标志(SR25404 和 SR25405) 表明运算结果是否超出 16 位带符号二进制数据范围的下限和上限。有关带符号二进制数据内容详见 25 页。

标志

ER: 间接寻址 DM 字不存在。(* DM 字内容不是 BCD 码或超出 DM 区域)

CY: 当运算结果大于 FFFF, CY 为 ON。

EQ: 当运算结果是零, EQ 为 ON。

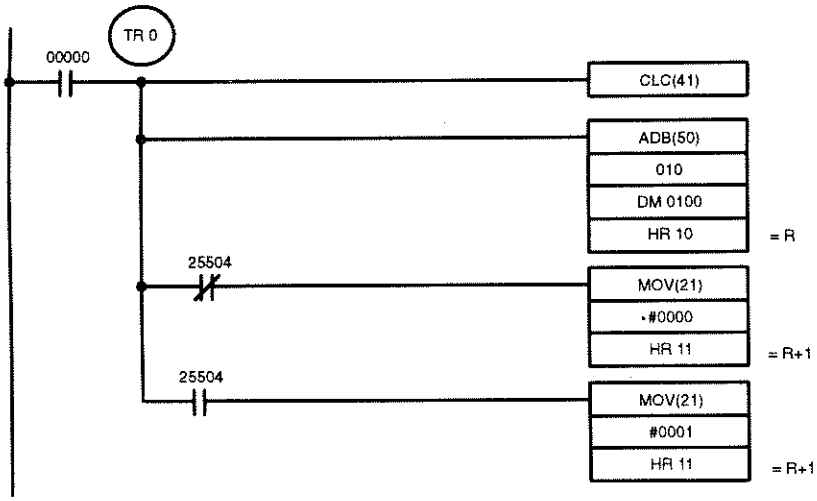
OF: 当运算结果超过 + 32, 767(7FFF), OF 为 ON。

UF: 当运算小于 － 32, 768(8000), UF 为 ON。

N: 当运算结果 15 位置"1", N 为 ON。

例 1：无符号数据加法

下例给出了四位带进位位的加法运算根据 CY 状态在 R＋1 中存入＃ 0000 和＃ 0001 以保存进位状态。



地址	指令	操作数
00000	LD	00000
00001	OUT	TR 0
00002	CLC(41)	
00003	ADB(50)	
		010
		DM 0100
		HR 10
00004	AND NOT	25504
00005	MOV(21)	
		＃ 0000
		HR 11
00006	LD	TR 0
00007	AND	25504
00008	MOV(21)	
		＃ 00001
		HR 11

下面情况中，A6E2＋80C5＝127A7。运算结果 5 位数，所以 CY（SR25504）＝1，R＋1 内容为＃0001。

Au：IR010			
A	6	E	2

＋

Ad：DM0100			
8	0	C	5

R＋1：HR 11				R：HR 10			
0	0	0	1	2	7	A	7

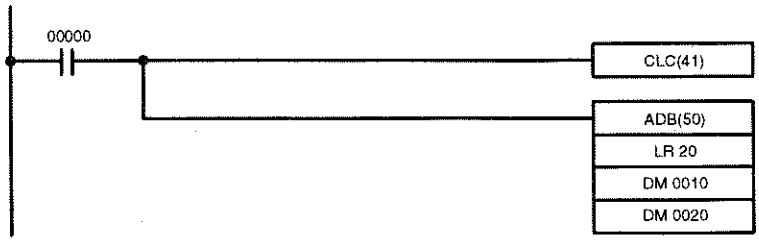
注

在加法运算过程中，UF 和 OF 也会变为 ON，但可忽略，因为 UF 和 OF 仅在带符号二进制加法中有用。

例 2：二进制符号加法

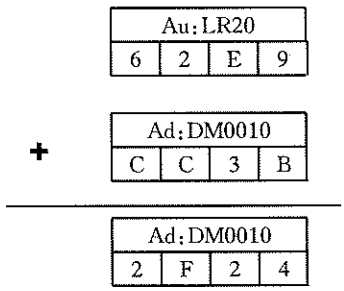
下例中，ADB（50）用于两个 16 位带符号二进制数值相加。（用二进制补码表示负值。）

带符号 16 位二进制数值的有效范围是－32，767（8000）到＋32，768（7FFF）。如果运算结果超过＋32，768（7FFF），上限溢进标志（OF：SR25404）为 ON，如果结果小于下限－32，767（8000），下限溢出标志（UF：SR25405）为 ON。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	ADB(50)	
		LR 20
		DM 0010
		DM 0020

在下面实例中，25，321＋(－13，253)＝12，068(62E9＋CC3B＝2F24)OF 和 UF 都不为 ON。

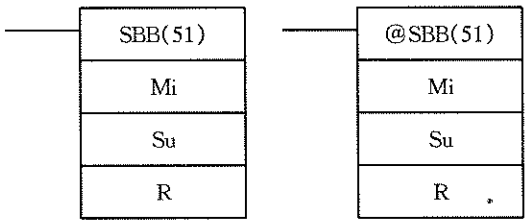


注

当带前符号二进制数相加时，进位位 CY 状态可忽略，因为进位位仅在无符号十六进制数值运算时有作用。

5－20－2 二进制减法指令－SBB(51)

梯形符号

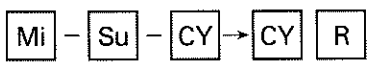


操作数数据区域

Mi：被减数字(二进制)
IR，SR，AR，DM，HR，TC，LR，#
Su：减数字(二进制)
IR，SR，AR，DM，HR，TC，LR，#
R：结果字
IR，SR，AR，DM，HR，LR

说明

当指令执行条件为 OFF，SBB (51) 指令不执行。当执行条件为 ON，SBB (51)将 Su 的内容和 CY 从 Mi 中减去，并将结果存入 R。如果运算结果为负值，CY 为 ON，实际结果的二进制补码存入 R。



SBB(51)也可用于带符号二进制数据减法运算。上限和下限溢出标志（SR25404 和 SR25405）指示运算结果是否超出 16 位二进制数据的上/下极限。有关带符号二进制数据参见 25 页。

标志

ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

CY：当运算结果为负值，CY 为 ON，即，当 Mi 小于 Su 加 CY。

EQ：当运算结果是零，EQ 为 ON。

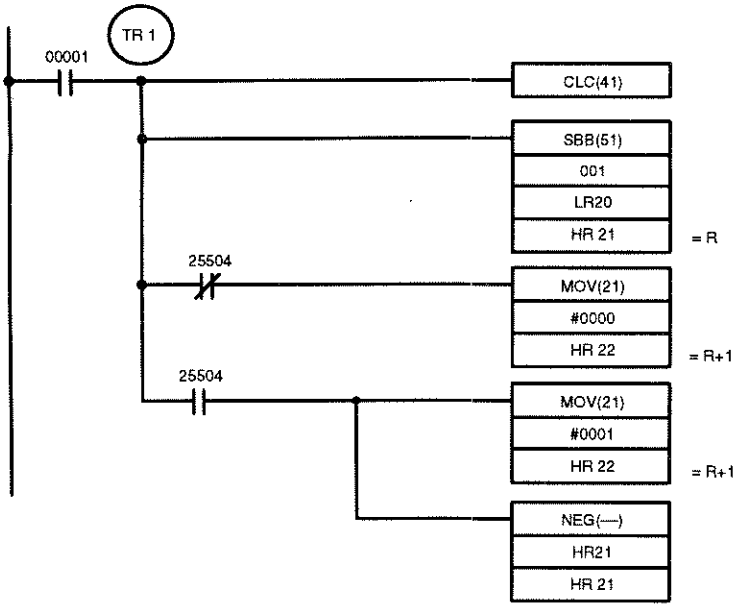
OF：当运算结果是超出＋32，767(7FFF)，OF 为 ON。

UF：当运算结果小于－32，768(8000)，UF 为 ON。

N：当运算结果 15 位置“1”，N 为 ON。

例 1：不带符号二进制减法

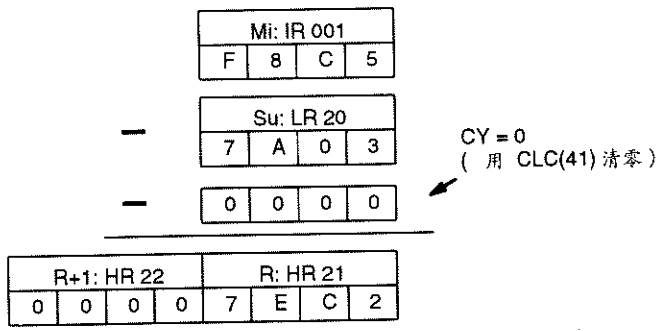
下例给出了四位带进位位的减法运算，根据 CY 状态在 R＋1 存入＃0000 或＃0001 以保存进位位状态。



地址	指令	操作数
00000	LD	00001
00001	OUT	TR 1
00002	CLC(41)	
00003	SBB(51)	
		001
		LR 20
		HR 21
00004	AND NOT	25504
00005	MOV(21)	
		# 0000
		HR 22
00006	LD	TR 1
00007	AND	25504
00008	MOV(21)	
		# 0001
		HR 22
00009	NEG(－)	
		HR 21
		HR 21

在下面实例中，IR001（＃F8C5）减去 LR20 的内容（＃7A03）和 CY。运算结果存入 HR21，HR22 的内容（＃0000）表示运算结果为正值。

如果运算结果为负值，CY 为 ON，＃0001 存入 HR22 中，运算结果将为它的二进制补码形式。



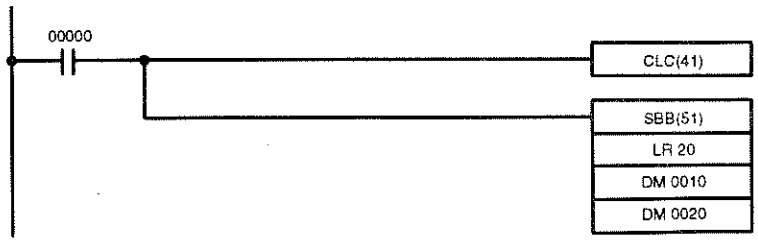
注

UF 和 OF 标志状态可忽略，因为上/下溢出标志仅在带符号二进制运算中起作用。

例 2：带符号二进制减法

在下例中，SBB（51）用于将从一个数中减去一个 16 位二进制带符号数。（二进制补码用以表示负值）。

16 位带符号二进制数值的有效范围是从－32，768（8000）到＋32，767（7FFF）。如果运算结果超过＋32，767（7FFF），上限溢出标志（OF：SR25404）为 ON；如果运算结果小于－32，768（8000），下限溢出标志（UF：SR25405）为 ON。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	SBB(51)	
		LR 20
		DM 0010
		DM 0020

下面的实例表明，30,020－(－15,238)＝45,258(7544－C47A＝60CA)。OF 标志变为 ON，表示运算结果超过 16 位带符号二进制数据范围的上限。(即，运算结果是一个超过 32,767 (7FFF) 正值，不是表示带符号二进制负数。)

Mi：LR20			
7	5	4	4

－

Su：DM0010			
C	4	7	A

R：DM0020			
B	0	C	A

下面实例表明，－30,000－3,000＝－33,000(8AD0－0BB8＝7F18)。UF 标志变为 ON，表示运算结果小于 16 位带符号二进制数据范围的下限。(即，运算结果是一个小于－32,768(8000)，不是表示带符号二进制的正值。)

Mi：LR20			
8	A	D	0

－

Su：DM0010			
0	B	B	8

R：DM0020			
7	F	1	8

真正运算结果的绝对值(80F8＝33,000)可用 NEG(－)对 7F18 求二进制补码获得。

注

当对带符号二进制数据作加法运算，CY 标志状态可以忽略，因为 CY 标志仅在不带符号十六进制加法运算时起作用。

5－20－3 二进制乘法指令－MLB(52)

梯形符号

MLB(52)
Md
Mr
R

@MLB(50)
Md
Mr
R

操作数数据区域

Md：被乘数字(二进制)
IR, SR, AR, DM, HR, TC, LR, #

Mr：乘数字(二进制)
IR, SR, AR, DM, HR, TC, LR, #

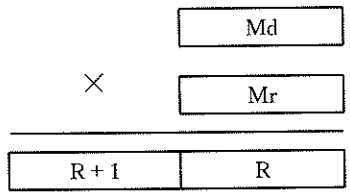
R：结果字
IR, SR, AR, DM, HR, LR

限制

R 和 R＋1 必须在同一数据区。

说明

当指令执行条件为 OFF，MLB（52）指令不执行。当执行条件为 ON，MLB（52）将 Md 的内容与 Mr 的内容相乘，并把结果存入 R 的低四位和 R＋1 的高四位。



注意事项

MLB(52)不能用于带符号二进制数据乘法。带符号二进制乘法采用 MBS(52) 指令。详见 5－20－7“带符号二进制乘法指令—MBS(52)”。

标志

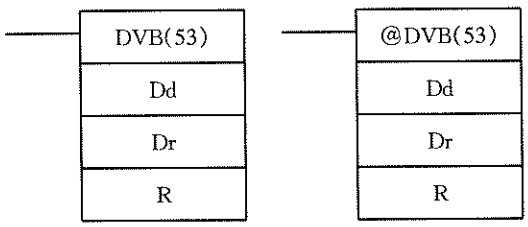
ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当运算结果是零，EQ 为 ON。

N：当 R＋1 中 15 位置“1”，N 为 ON。

5－20－4 二进制除法指令—DVB(53)

梯形符号



操作数数据区域

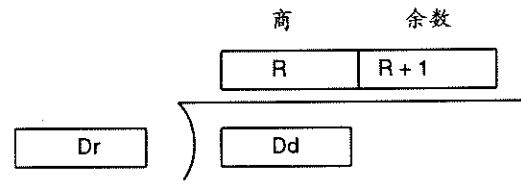
Dd：被除数字（二进制）
IR，SR，AR，DM，HR，TC，LR，＃

Dr：除数字（二进制）
IR，SR，AR，DM，HR，TC，LR，＃

R：结果字
IR，SR，AR，DM，HR，LR

说明

当指令执行条件为 OFF，DVB(53)指令不执行。当执行条件为 ON，DVB(53)将 Dd 的内容除以 Dr 的内容，并将运算结果存入 R 和 R＋1；商存入 R，余数存入 R＋1。



注意事项

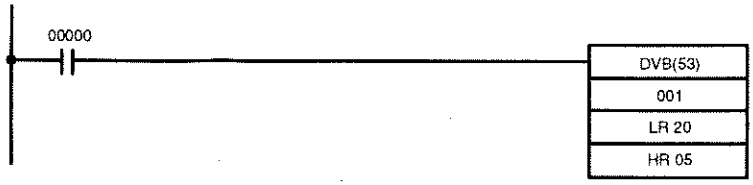
DVB(53)不能用于带符号二进制数除法。带符号二进制数除法可用 DBS（－）指令。详见 5－20－9 带符号二进制除法指令—DBS（－）。

标志

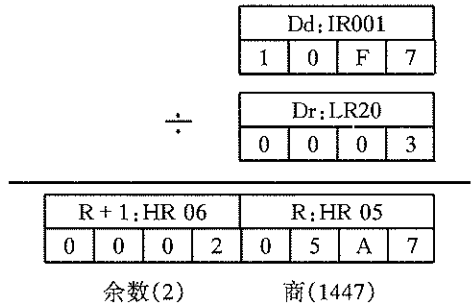
ER：Dr 内容是零。间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当运算结果是零，EQ 为 ON。

N：当 R 中 15 位置“1”，N 为 ON。

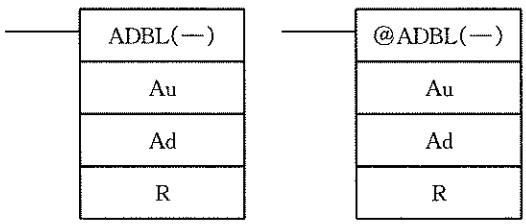


地址	指令	操作数
00000	LD	00000
00001	DVB(53)	
		001
		LR 20
		HR 05



5－20－5 双字二进制加法指令－ADBL(－)

梯形符号



操作数数据区域

Au：被加数起始字(二进制)
IR，SR，AR，DM，HR，LR，

Ad：加数起始字(二进制)
IR，SR，AR，DM，HR，LR

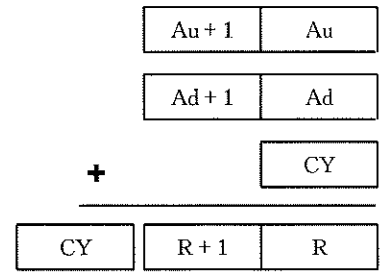
R：结果字
IR，SR，AR，DM，HR，LR

限制

Au 和 Au＋1 必须在同一数据区，Ad 和 Ad＋1 以及 R 和 R＋1 也必须在同一数据区。

说明

当指令执行条件为 OFF，ADBL（－）指令不执行。当执行条件为 ON，ADBL(－) 将 Au＋1 和 Au 中的八位数内容与 Ad＋1 和 Ad 八位数内容以及 CY 相加，并将结果存入 R 和 R＋1。如果运算结果大于 FFFFFFFF，CY 为 ON，有关带符号二进制数据详见 P25 页。



ADBL(－)指令也可用于带符号二进制数据相加。上限和下限溢出标志（SR25404 和 SR25405）指示运算结果超出 32 位二进制带符号数据上限或下限范围。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

CY：当运算结果超出 FFFF FFFF，CY 为 ON。

EQ：当运算结果是零，EQ 为 ON。

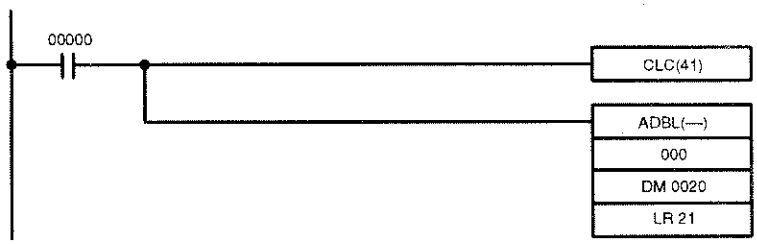
OF：当运算结果超过＋2,147,483,647(7FFF FFFF)，标志 OF 为 ON。

UF：当运算结果小于－2,147,483,648(8000 0000)，标志 UF 为 ON。

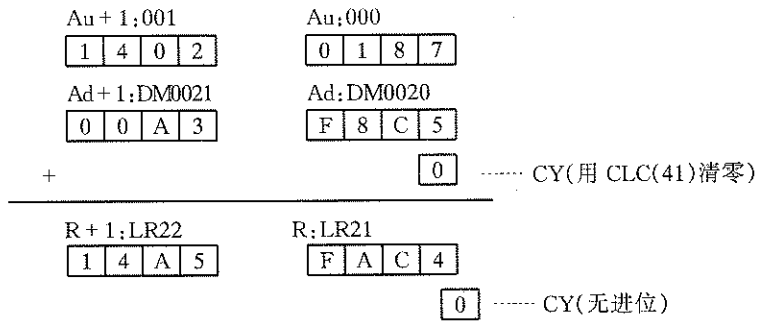
N：当 R 中 15 位置“1”，N 为 ON。

例 1：不带符号数据运算

下例给出八位带进位位 CY(SR25504)数加法，CY 用于表示第九位状态。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	ADBL(－)	
		000
		DM 0020
		LR 21



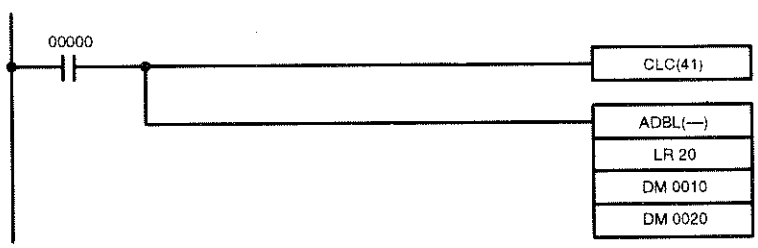
注

UF 和 OF 标志位状态可以忽略，因为它们仅在带符号二进制加法运算时起作用。

例 2：带符号二进制数据运算

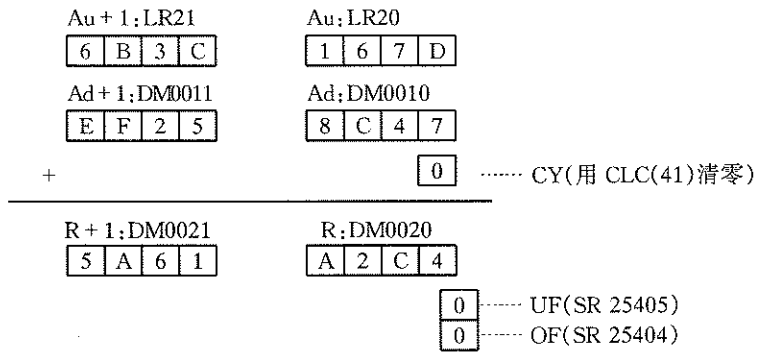
下例中，ADBL（－）用于将 32 位带符号二进制相加，并将 32－位带符号二进制存入 R 和 R＋1。（二进制补码用于表示负值）。

32 位带符号二进制数的有效范围是从－2,147,483,648（8000 0000）到＋2,147,483,647（7FFF FFFF）。如果运算结果超出＋2,147,483,647（7FFF FFFF），上限溢出标志（OF：SR25404）变为 ON。如果运算结果小于－2,147,483,648（8000 0000），下限溢出标志（UF：SR25405）变为 ON。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	ADBL(－)	
		LR 20
		DM 0010
		DM 0020

下面实例中，1，799，100，099＋（－282，751，929）＝1，516，348，100（6B3C167D＋EF258C47＝5A61A2C4）。OF 和 UF 标志都为 OFF。

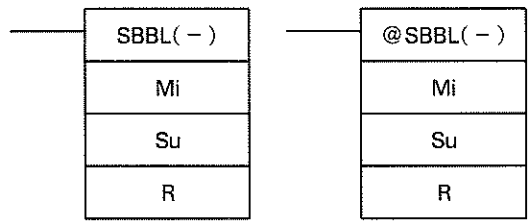


注

当作带符号二进制加法运算时可以忽略 CY 标志状态，因为它仅在不带符号十六进制加法运算中起作用。

5－20－6　双字二进制减法指令－SBBL(－)

梯形符号



操作数数据区域

Mi：被减数起始字(二进制)
IR，SR，AR，DM，HR，TC，LR

S：减数起始字(二进制)
IR，SR，AR，DM，HR，TC，LR

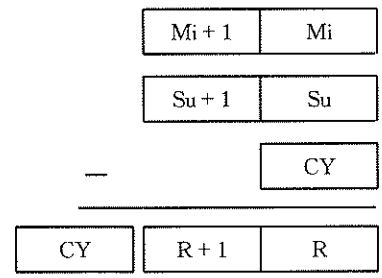
R：结果字
IR，SR，AR，DM，HR，LR

限制

Mi 和 Mi＋1 必须在同一数据区，Su 和 Su＋1 以及 R 和 R＋1 也必须在同一数据区。

说明

当指令执行条件为 OFF，SBBL(－)不执行。当执行条件为 ON，SBBL(－)将 Mi 和 Mi＋1 中的八位数值减去 CY 和 Su 及 Su＋1 中的八位数值，并将运算结果存入 R 和 R＋1。如果结果是负值，CY 为 ON，运算实际结果的二进制补码存入 R＋1 和 R。采用双字二进制求补指令将二进制补码变为真值。



SBBL(－)指令也可用于带符号二进制数据减法运算。上限/下限溢出标志（SR25404 和 SR25405）表示运算结果是否超出 32 位带符号二进制范围的上/下限。有关带符号二进制数据内容详见 25 页。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

CY：当运算结果是负值，CY 为 ON，即，Mi 小于 Su 加 CY。

EQ：当运算结果是零，EQ 为 ON。

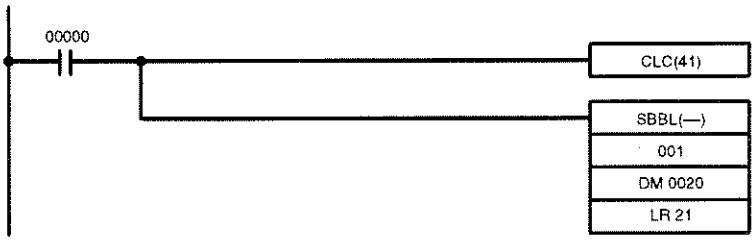
OF：当运算结果超出 +2,147,483,647(7FFFFFFF)，OF 为 ON。

UF：当运算结果小于 −2,147,483,648(8000 0000)，UF 为 ON。

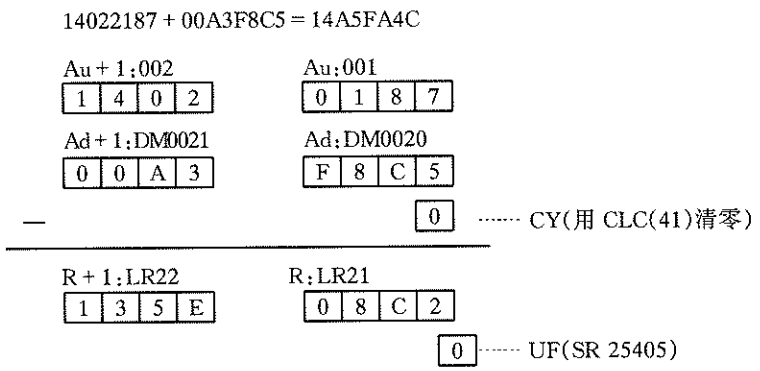
N：当 R+1 中 15 位置"1"，N 为 ON。

例 1：不带符号数据运算

本例中，IR002 和 IR001 中八位数减去 DM0021 和 DM0020 中的八位数，结果存入 LR22 和 LR21。如果运算结果为负值，CY（SR25504）变为 ON。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	SBBL(－)	
		001
		DM 0020
		LR 21



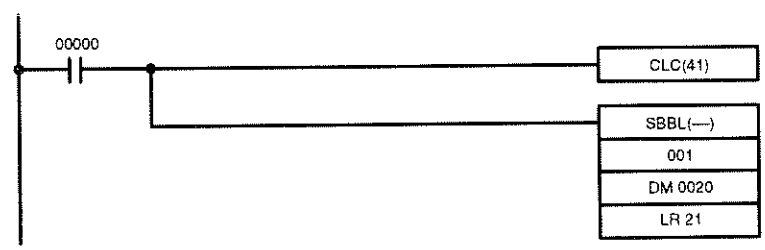
注

UF 和 OF 标志状态可忽略，因为它们仅在带符号二进制减法运算中起作用。

例 2：带符号二进制数据运算

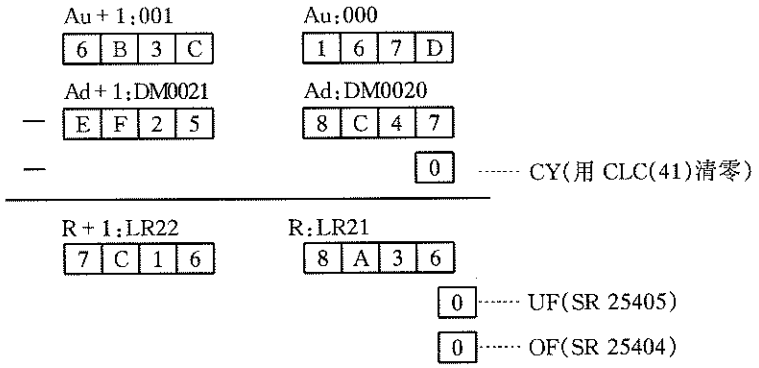
下例中，SBBL（－）用于将一个 32 位带符号二进制数从另一个数中减去，并将 32 位带符号二进制运算结果存入 R 和 R+1。

32 位带符号二进制数有效范围是从 −2,147,483,648(8000 0000)到 +2,147,483,647(7FFF FFFF)，当运算结果大于 +2,147,483,647(7FFF FFFF)，上限溢出标志（OF:SR25404）变为 ON。当运算结果小于 −2,147,483,648（8000 0000），下限溢出标志（UF：SR25405）变为 ON。



地址	指令	操作数
00000	LD	00000
00001	CLC(41)	
00002	SBBL(－)	
		001
		DM 0020
		LR 21

下面实例中，1，799，100，099－（－282，751，929）＝2，081，851，958（6B3C167D｛EF258C47－100000000｝＝7C168A36）。OF 和 UF 标志都为 OFF。

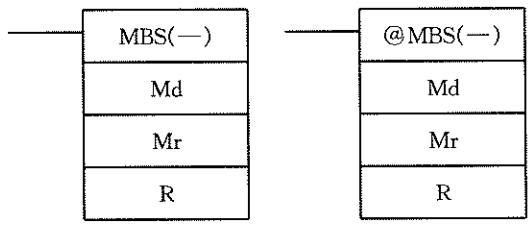


注

当作带符号二进制数加法运算时 CY 标志状态可忽略，因为 CY 仅在不带符号十六进制数加法运算有用。

5－20－7　带符号二进制乘法指令－MBS(－)

梯形符号



操作数数据区域

Md：被乘数字(二进制)
IR，SR，AR，DM，HR，TC，LR，#

Mr：乘数字(二进制)
IR，SR，AR，DM，HR，TC，LR，#

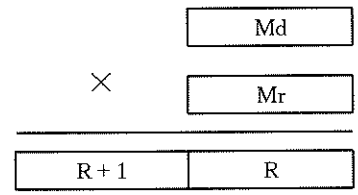
R：结果字
IR，SR，AR，DM，HR，LR

限制

R 和 R＋1 必须在同一数据区。

说明

MBS(－)指令将两个字的带符号二进制内容相乘，并将八位带符号的二进制数运算结果存入 R＋1 和 R。运算结果的低四位存入 R，高四位存入 R＋1。有关带符号二进制数据详见 25 页。



标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当运算结果是零，EQ 为 ON，反之 EQ 为 OFF。

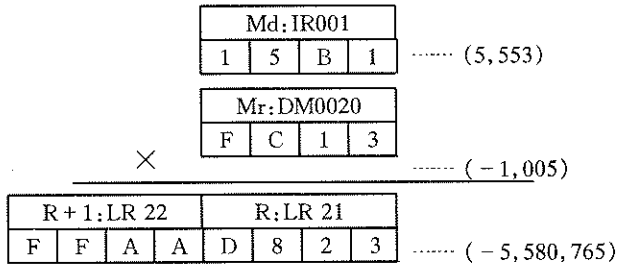
N：当 R＋1 中 15 位置“1”，N 为 ON。

例

下例中，MBS(－)用于将 IR001 中带符号二进制内容与 DM0020 中带符号二进制内容相乘，并将运算结果存入 LR21 和 LR22。

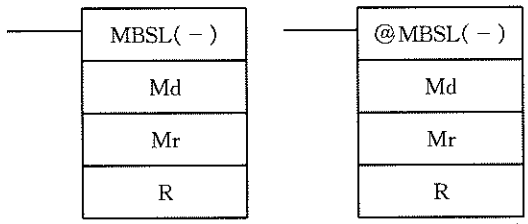


地址	指令	操作数
00000	LD	00000
00001	MBS(－)	
		001
		DM 0020
		LR 21



5－20－8 带符号双字二进制乘法指令－MBSL(－)

梯形符号



操作数数据区域

Md:被乘数起始字
IR, SR, AR, DM, HR, TC, LR

Mr:乘数起始字
IR, SR, AR, DM, HR, TC, LR

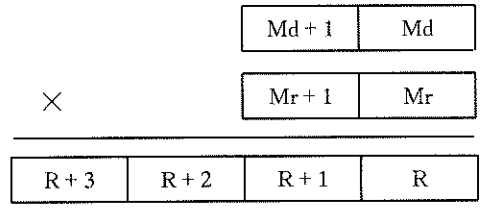
R:结果字
IR, SR, AR, DM, HR, LR

限制

Md 和 Md＋1 必须在同一数据区，Mr 和 Mr＋1 以及 R 至 R＋3 也必须同一数据区。

说明

MBSL(－)将 Md＋1 和 Md 中的 32 位(八位数)带符号二进制数据乘以 Mr＋1 和 Mr 中 32 位带符号二进制数据，并将 64 位(十六位数)带符号二进制运算结果存入 R＋3～R 中。有关带符号二进制数内容详见 25 页。



标志

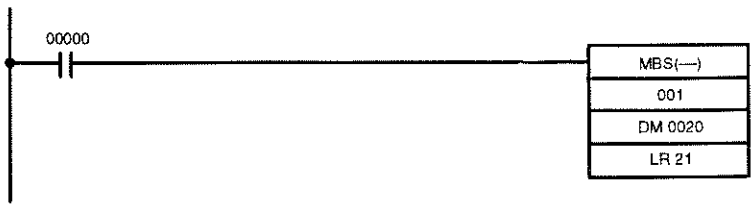
ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ:当运算结果是零(R＋3～R 所有内容都是零)，EQ 为 ON，反之 EQ 为 OFF。

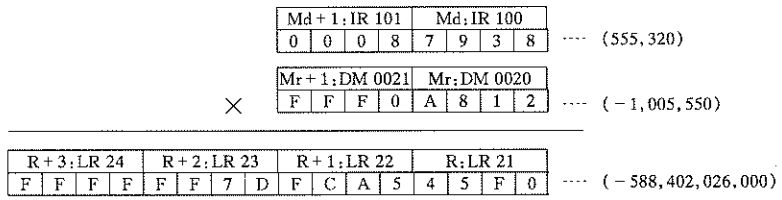
N：当 R＋3 中 15 位置“1”，N 为 ON。

例

下例中，MBSL(－) 用于将 IR101 和 IR100 中的带符号二进制内容乘以 DM0021 和 DM0020 中的带符号二进制内容，并将运算结果存入 LR24～LR21。

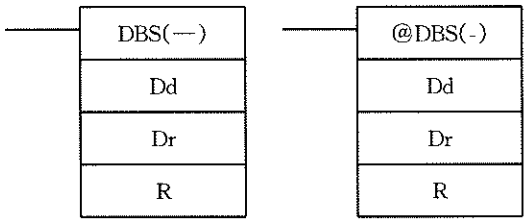


地址	指令	操作数
00000	LD	00000
00001	MBSL(－)	
		100
		DM 0020
		LR 21



5－20－9 带符号二进制除法指令－DBS(－)

梯形符号



操作数数据区域

Dd：被除数字
IR，SR，AR，DM，HR，TC，LR，#

Dr：除数字
IR，SR，AR，DM，HR，TC，LR，#

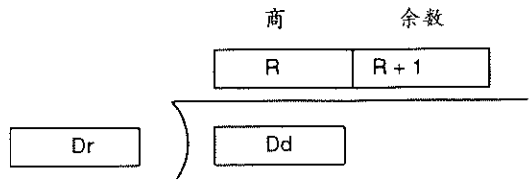
R：结果字
IR，SR，AR，DM，HR，LR

限制

R 和 R＋1 必须在同一数据区。

说明

DBS(－)将 Dd＋1 和 Dd 中的 32 位(八位数)带符号二进制数据除以 Dr＋1 和 Dr 中带符号二进制内容除以 Dr 中带符号二进制内容，并将八位带符号二进制数结果存入 R＋1 和 R。商存入 R，余数存入 R＋1。有关符号二进制数据详见 25 页。



标志

ER：Dr 内容是零。

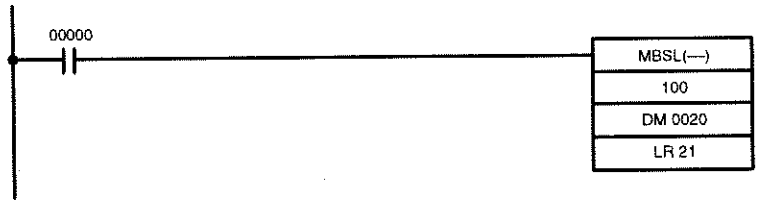
间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

EQ：当 R 的内容(商)是零，EQ 为 ON，反之 EQ 为 OFF。

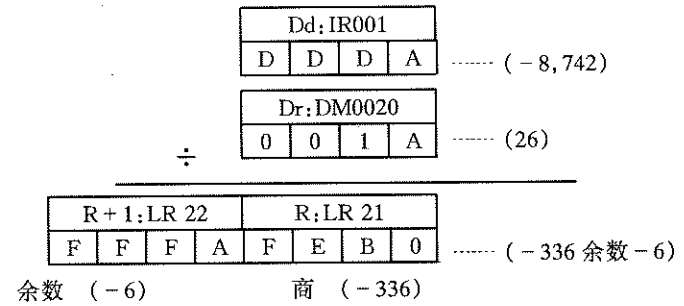
N：当 R 中 15 位置“1”，N 为 ON。

例

下例中，DBS(－)用于 IR001 的带符号二进制内容除以 DM0020 的带符号二进制内容，并将运算结果存入 LR21 和 LR22。

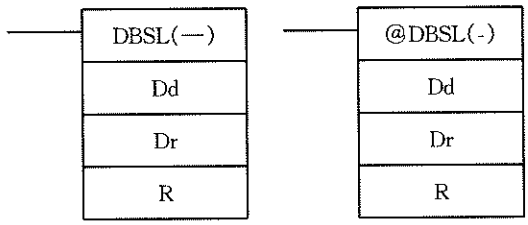


地址	指令	操作数
00000	LD	00000
00001	DBS(－)	
		001
		DM 0020
		LR 21



5－20－10 带符号双字二进制除法指令－DBSL(－)

梯形符号



操作数数据区域

Dd:被除数字(二进制)
IR, SR, AR, DM, HR, TC, LR

Dr:除数字(二进制)
IR, SR, AR, DM, HR, TC, LR

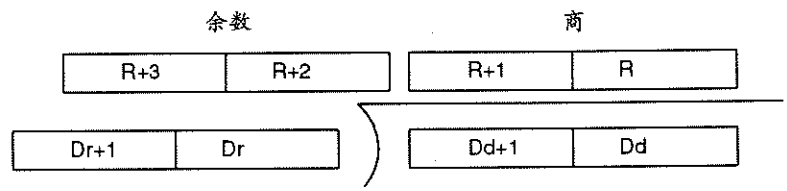
R:结果字
IR, SR, AR, DM, HR, LR

限制

Dd 和 Dd＋1 必须在同一数据区，Dr 和 Dr＋1 以及 R 至 R＋3 也必须在同一数据区。

说明

DBS(－)将 Dd＋1 和 Dd 中的 32 位(八位数)带符号二进制数据除以 Dr＋1 和 Dr 中 32 位带符号二进制数据，并将 64 位(十六位数)带符号二进制运算结果存入 R＋3～R。商存入 R＋1 和 R，余数存入 R＋3 和 R＋2。有关带符号二进制数据详见 25 页。



标志

ER：Dr＋1 和 Dr 内容是零。

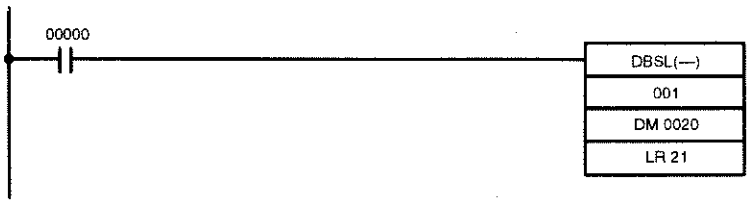
间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

EQ：当 R＋1 和 R 的内容(商)是零，EQ 为 ON，反之为 OFF。

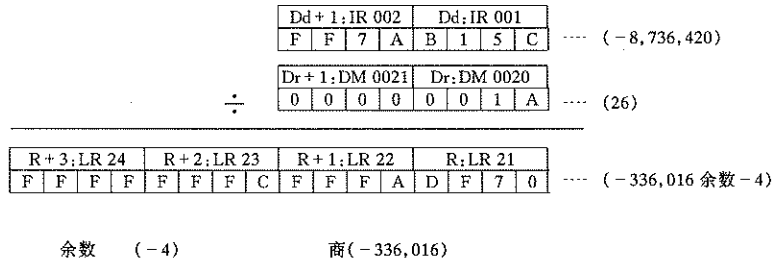
N：当 R＋1 中 15 位置“1”，N 为 ON。

例

下例中，DBSL（－）用于将 IR002 和 IR001 的带符号二进制内容除以 DM0021 和 DM0020 的带符号二进制内容，并将运算结果存入 LR24～LR21。



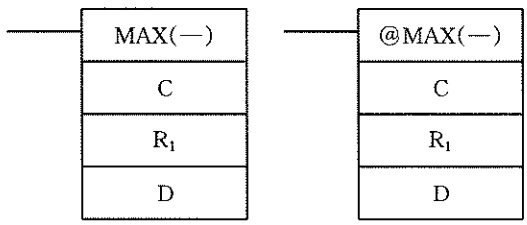
地址	指令	操作数
00000	LD	00000
00001	DBSL(－)	
		001
		DM 0020
		LR 21



5－21 特殊算术指令

5－21－1 寻找最大值指令—MAX(－)

梯形符号



操作数数据区域

C：控制数据
IR，SR，AR，DM，HR，LR，＃

R_1：范围起始字
IR，SR，AR，DM，HR，TC，LR

D：目标字
IR，SR，AR，DM，HR，LR

限制

C 中的 N 数据必须是 001～999 间的 BCD 码。

R_1 和 R_1＋N－1 必须在同一数据区。

说明

当指令执行条件为 OFF，MAX(－)指令不执行，当执行条件为 ON，MAX(－)搜索 R_1～R_1＋N－1 存储器范围内含有最大数值的地址，并将最大数值输出到目标字(D)中。

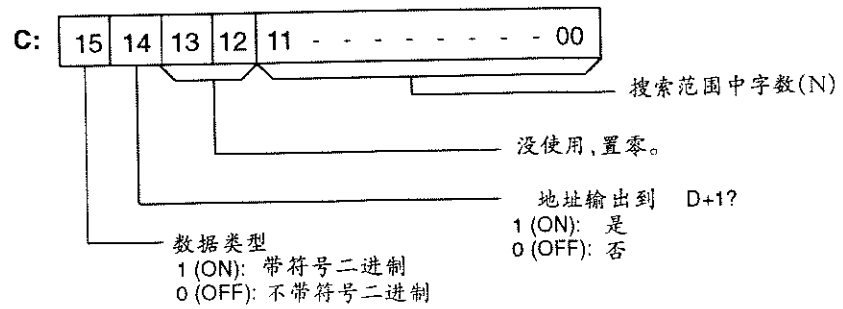
如果 C 的 14 位为 ON，MAX（－）把含有最大值字的地址放在 D＋1 中。对 DM 区而言，不同的地址确定在于：

1. 对于一个在 DM 区的地址，字地址写入 D＋1。例如，如果含有最大值的存储器地址是 DM0114，这时＃0114 写入 D＋1。
2. 对于一个在另一数据区的地址，从寻找开始的地址至最大值地址的个数被写入 D＋1。例如，要含有最大数值的地址 IR114，寻找范围的初始地址是 IR014，这时 D＋1 中为＃100。

如果 C 的 14 位为 ON，含有相同最大数值的地址不止一个，处于最小的地址被存入 D＋1。

搜索范围中字数(N)在 C 的低三位数中，该数必须是 001～999 内的 BCD 码。

当 C 的 15 位为 OFF 时，搜索范围中数据按不带符号二进制处理；为 ON 时，按带符号二进制处理。



注意

如果 C 的 15 位为 ON，超过＃8000 数值作为负数处理，所以搜索结果会根据数据类型定义不同而不同。应确信数据类型的定义是正确的。

标志

ER：间接寻地 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

在 C 中指定的字不是 BCD 码(000～999)。

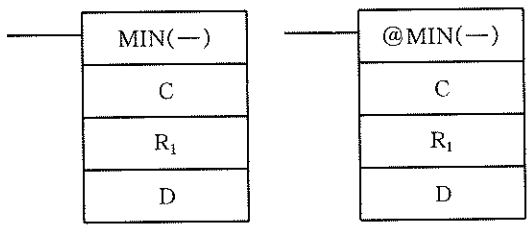
R_1 和 R_1＋N－1 不在同一数据区。

EQ：当最大值是＃0000，EQ 为 ON。

N：当 D 中 15 位置“1”，N 为 ON。

5－21－2 寻找最小值指令－MIN（－）

梯形符号



操作数数据区域

C：控制数据
IR，SR，AR，DM，HR，LR，＃

R_1：范围起始字
IR，SR，AR，DM，HR，TC，LR

D：目标字
IR，SR，AR，DM，HR，LR

限制

C 中的 N 必须是 001～999 间的 BCD 码。

R_1 和 R_1＋N－1 必须在同一数据区。

说明

当指令执行条件为 OFF，MIN（－）指令不执行，当执行条件为 ON，MIN（－）从 R_1～R_1＋N－1 存储器范围搜索含有最小数值内容的地址，并将最小值存入指定字 D 内。

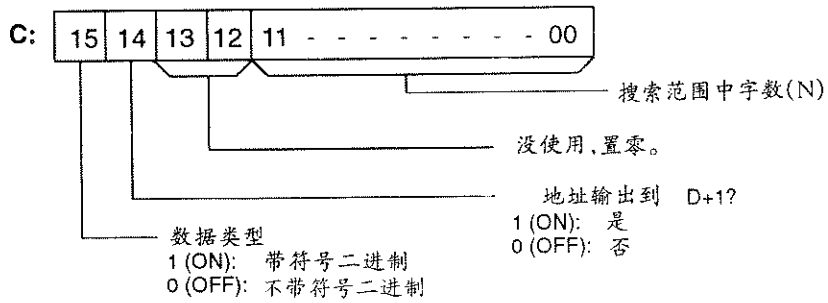
如果 C 的 14 位为 ON，MIN（－）把含有最小数值字的地址存入 D＋1。对于 DM 区域，不同的地址确定在于：

1. 对于一个在 DM 区中的地址，字地址写入 D＋1。例如含有最小值的地址是 DM0114，这时＃0114 写入 D＋1。
2. 对于一个处于另一数据区的地址，从开始搜索起的地址至最小值地址的个数写入 D＋1。例如，含有最小值的地址是 IR114，搜索范围的起始字是 IR014，这时＃0100 写入 D＋1。

如果 C 的 14 位为 ON，且含有相同最小值的地址不止一个，这些地址中处于最低的地址存入 D＋1。

搜索范围中字数在 C 的低 3 位数中，该数必须是 001～999 之间的 BCD 数。

当 C 的 15 位为 OFF，搜索范围内的数据作为不带符号二进制处理，当 C 的 15 位为 ON，数据作带符号数据处理。有关带符号二进制数据详见 25 页。



注意

如果 C 的 15 位为 ON，超过＃8000 数值作为负值处理，所以，搜索结果会根据数据类型定义不同而不同。应确信数据类型的定义是正确的。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

C 中指定的字数不是 000～999 间 BCD 码。

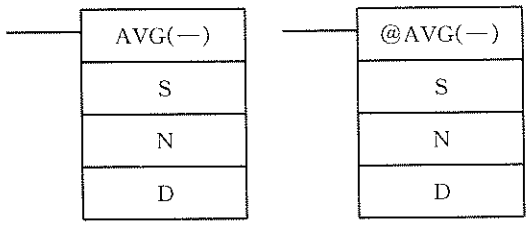
R_1 和 R_1＋N－1 不在同一数据区。

EQ：当最小值是＃0000，EQ 为 ON。

N：当 D 中 15 位置“1”，N 为 ON。

5－21－3 求平均值指令－AVG（－）

梯形符号



操作数数据区域

S：源字
IR，SR，AR，DM，HR，TC，LR，＃

N：循环数
IR，SR，AR，DM，HR，TC，LR，＃

D：目标起始字
IR，SR，AR，DM，HR，LR

限制

S 的数据必须是十六进制。

N 必须是＃0001～＃0064 间的 BCD 码。

D 和 D＋N＋1 必须在同一数据区。

说明

AVG（－）用于计算经过 N 个循环后 S 的平均值。

当指令执行条件为 OFF，AVG（－）指令不执行。

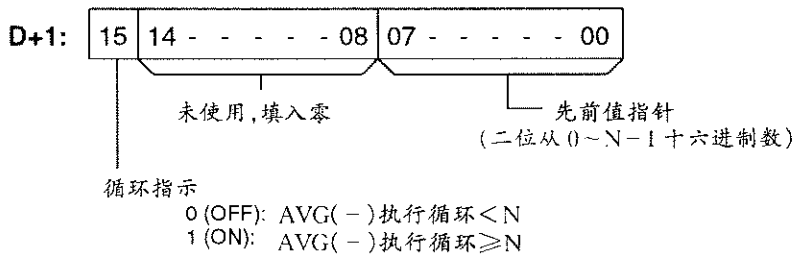
执行条件为 ON，开始的 N－1 次循环，AVG（－）将 S 的数值写入 D。每次 AVG（－）指令执行，S 的原先值存入字 D＋2～D＋N＋1。指令每执行一次，D＋1 的低二位数自动加一，其值作为指针指示原先值存放的地址。在开始的 N－1 个循环中，D＋1 的 15 位保持 OFF 状态。

第 N 次循环，首先 S 的原先值存入 D＋2～D＋N＋1 范围内的最后字内，然后计算出存贮在 D＋2～D＋N＋1 中的原先值的平均值，并将运算结果存入 D，并且将 D＋1 的 15 位置为 ON，先前值指针（D＋1 的低二位数）清零。每次 AVF（－）执行，均用 S 中的先前值改写指针指示的字的内容，并计算出新的平均值写入 D。循环 N－1 次后，指针又清零。

下图给出 D～D＋N＋1 字的功能。

D	平均值（N 或大于 N 次循环后）
D＋1	先前数值指针和特循环指示器
D＋2	先前数值＃1
D＋3	先前数值＃2
・ ・ ・	・　・　・ ・　・　・ ・　・　・
D＋N＋1	先前数值＃N

D＋1 中位的功能由下图给出，详细说明在后面。



原先值指针

相对于 D＋2 原先值指针指示存贮 S 最新数值存放的地方，即：指针值 0 指示 D＋2，指针值 1 指示 D＋3，等等。

循环指示器

AVG（－）执行 N 次后，循环指示器置为 ON。这时，D 将存有字 D＋2～D＋N＋1 中内容的平均值。平均值为四位十六进制，平均运算时数据被四舍五入到最相近的整数。

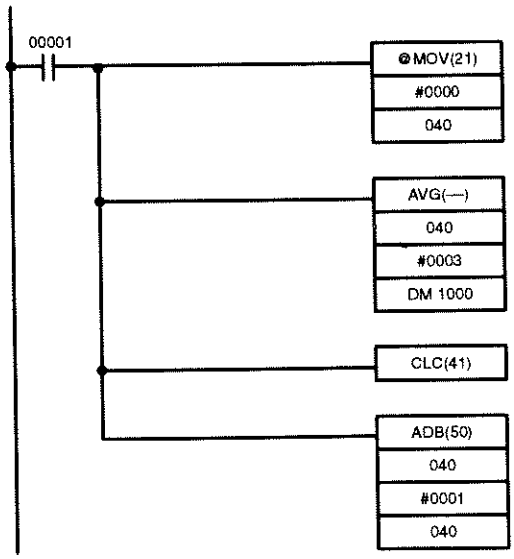
标志

ER：间接寻址 DM 字不存在。（＊DM 字的内容不是 BCD 码或超出 DM 区域）

一个或多个操作数设置不正确。

例

下例中，IR040 的内容设置成＃0000，每一循环内容加一。前两个循环，AVG（－）将 IR040 的内容传送到 DM1002 和 DM1003。DM1001 内容也将发生变化（这一变化用于确认 AVG（－）的结果已经变化）。AVG（－）在第三和最后一个循环计算 DM1002 到 DM1004 内容的平均数值，并将平均值存入 DM1000。



地址	指令	操作数
00000	LD	00001
00001	@MOV(21)	
		# 0000
		040
0002	AVG(—)	
		040
		# 0003
		DM 1000
00003	CLC(41)	
0004	ADS(50)	
		040
		# 0001
		040

第一次循环　第二次循环　第三次循环　第四次循环

IR040

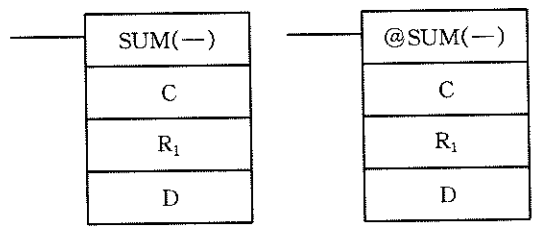
0000	0001	0002	0003

第一次循环　第二次循环　第三次循环　第四次循环

DM 1000	0000	0001	0001	0002	平均值
DM 1001	0001	0002	8000	8000	指针
DM 1002	0000	0000	0000	0003	
DM 1003	…	0001	0001	0001	IR040 三个先前
DM 1004	…	…	0002	0002	值

5－21－4　求和指令－SUM（－）

梯形符号



操作数数据区域

C：控制数据
IR，SR，AR，DM，HR，LR，#

R_1：范围内起始字
IR，SR，AR，DM，HR，TC，LR

D：目标起始字
IR，SR，AR，DM，HR，LR

限制

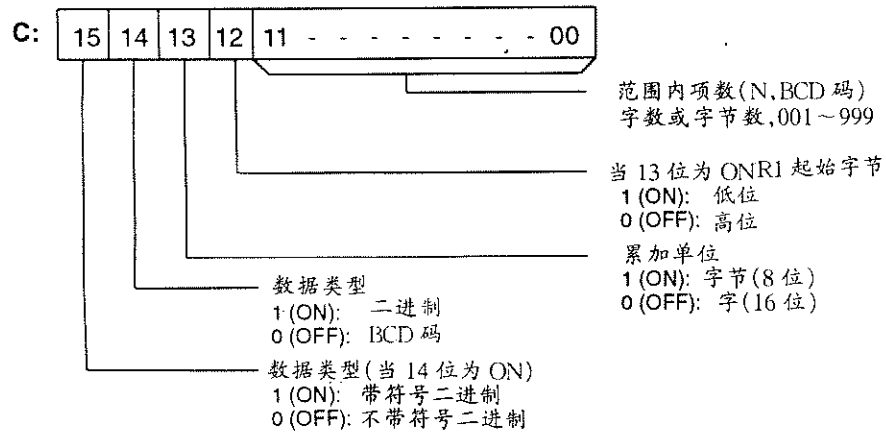
控制数据 C 的低三位数必须是 001～999 之间的 BCD 码。

如果 C 的 14 位为 OFF(设置成 BCD 码数据)，R_1 到 $R_1 + N - 1$ 范围内的所有数据必须是 BCD 码。

说明

当指令执行条件为 OFF，SUM（－）指令不执行，当执行条件为 ON，SUM（－）将 R_1 到 R_1+N-1 字的内容相加或者将 R_1 到 $R_1+N/2-1$ 字内的字节内容相加，并将和存入目标通道（D 和 D+1）。求和的数据可以是二进制或是 BCD 码数，输出也为对应的数据形式。二进制数据可以是带符号的也可以是不带符号的。

C 中位（bit）功能由下图给出，详细说明在后面。



范围中项数

在求和范围内的项数(N)包含在 C 的低三位数中，这数必须是 001 到 999 之间的 BCD 码。根据累加单位(13 位)，这数表示字(16 位)数或字节(8 位) 数。

累加单位

如果 C 的 13 位为 OFF，累加单位为字(16 位)，如果 C 的 13 位为 ON，累加单位为字节 (8 位)。

如果定义累加单位为字节，求和可以从 R_1 的低字节或高字节开始，如果 C 的第 12 位为 ON，R_1 的高字节不累加。

	高字节	低字节
R_1	1	2
R_1+1	3	4
R_1+2	5	6
R_1+3	7	8
· · ·	· · ·	· · ·

当 C 的 12 位为 OFF，求和以这样的顺序执行：1＋2＋3＋4……

当 C 的 12 位为 ON，求和以这样的顺序执行：2＋3＋4……

数据类型

当 C 的 14 位为 ON，15 位为 OFF，求和数据作无符号二进制数处理，当 C 的 14 位和 15 位均为 ON，求和数据作带符号二进制数处理。有关带符号二进制数据详见 25 页。

当 C 的 14 位为 OFF，不管 15 位处于什么状态，求和的数据作 BCD 码处理。

标志

ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

R_1 和 R_1+N-1 不在同一数据区。

C 中项数不是在 001 到 999 之间的 BCD 码。

当设定为 BCD 码数型，求和的数据不是 BCD 码。

EQ：当求和结果是零，EQ 为 ON。

N：当 D 中 15 位置“1”，N 为 ON。

例

下例中，当 IR00001 为 ON，从 DM0000 到 DM0009 间 10 个字的 BCD 码内容相加，结果写入 DM0100 和 DM0101。



地址	指令	操作数
00000	LD	00001
00001	@SUM(－)	
		#4010
		DM 0000
		DM 0100

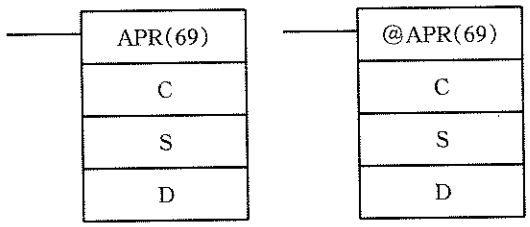
DM0000	3F2A
DM0001	51C3
DM0002	E02A
DM0003	7C9F
DM0004	2A20
DM0005	A827
DM0006	2A20
DM0007	E02A
DM0008	C755
DM0009	94DC

→

DM0100	2678
DM0101	0005

5－21－5 数学处理指令－APR (69)

梯形符号



操作数数据区域

C:控制字
IR, SR, AR, DM, HR, TC, LR, #

S:输入数据源字
IR, SR, AR, DM, HR, TC, LR

D:结果目标字
IR, SR, AR, DM, HR, TC, LR

限制

用作三角函数功能的 S 必须是 0000 到 0900 内的 BCD 码数（$0^\circ \leqslant \theta \leqslant 90^\circ$）

说明

当指令执行条件为 OFF，APR (69) 指令不执行。当执行条件为 ON，APR (69)的操作取决于控制字 C。如果 C 为 #0000 或 #0001，APR (69) 计算 Sin (θ)或 Cos(θ)。S 的 BCD 码值表示 θ，单位为 0.1 度。

如果 C 是地址，APR (69) 计算从 C 字开始的预设输入函数 f(x)。函数由一系列取决于算子的直线段构成（可近似一条曲线）。S 的 BCD 码或十六进制数表示输入 x。

标志

EQ：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

作为三角函数，x＞0900。（x 是 S 的内容）

C 指定的常数不是＃0000 或＃0001。

线性逼近数据不可读。

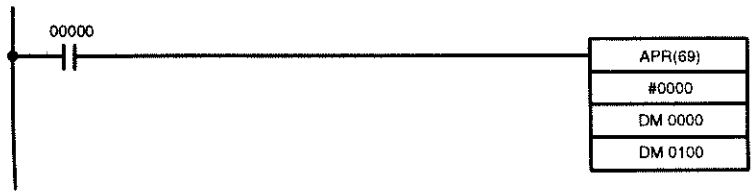
EQ：运算结果是零，EQ 为 ON。

N：当 D 中 15 位置“1”，N 为 ON。

例

Sin 函数

下例用 APR（69）Sin 函数功能计算 Sin30°。当 C 是＃0000，指令被定义为 Sin 函数功能。



地址	指令	操作数
00000	LD	00000
00001	APR(69)	
		# 0000
		DM 0000
		DM 0100

输入数据×

S：DM0000			
0	10^1	10^0	10^{-1}
0	3	0	0

输入数据是不超过＃0900 的 BCD 码数

结果数据

D：DM0100			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
5	0	0	0

结果数据有四个有效位，第五位及以上位忽略，Sin（90）将是 0.9999而不是 1。

Cos 函数

下例用 APR（69）Cos 函数功能计算 Cos30°。当 C 是＃0001，指令被定义为 Cos 函数功能。



地址	指令	操作数
00000	LD	00000
00001	APR(69)	
		# 0001
		DM 0010
		DM 0110

输入数据×

S：DM0010			
0	10^1	10^0	10^{-1}
0	3	0	0

输入数据是不超过＃0900 的 BCD 码数

结果数据

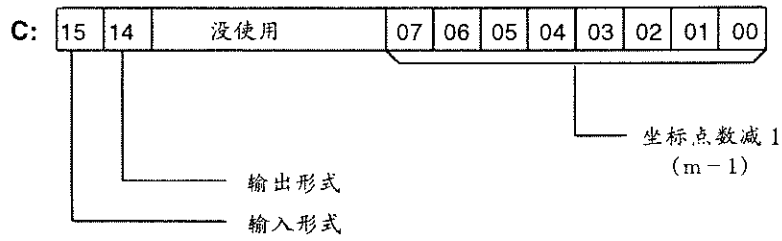
D：DM0110			
10^{-1}	10^{-2}	10^{-3}	10^{-4}
8	6	6	0

结果数据有四个有效位，第五位及以上位忽略，Cos（0）将是 0.9999而不是 1。

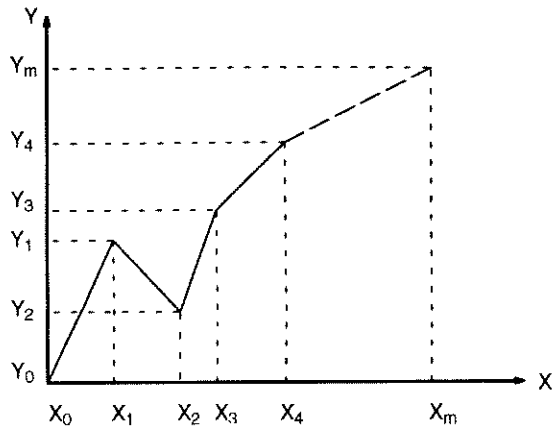
线性逼近

当 C 是一个存储器地址，APR（69）被定义为线性逼近功能。字 C 是含有线性逼近数据存储器块的起始字。

字 C 的内容指定逼近的线段数及指定输入和输出是 BCD 的还是二进制形式。00～07 位内容为线段数减 1（m－1）的二进制数据。14 位和 15 位分别定义输出、输入形式：0 定义为 BCD 码，1 定义为二进制。

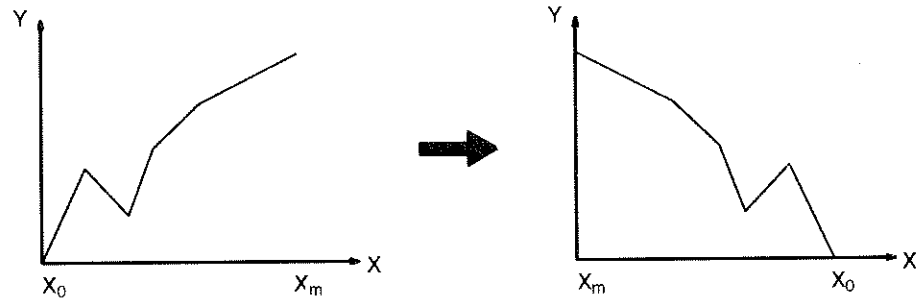


如下表所示，输入了定义 m 条线段的 m＋1 端点坐标。输入的所有坐标点均采用二进制形式。输入坐标点总是从输入最低的 X 值（X_i）到最高值（X_m）。X_0 是 0000 就不必输入。

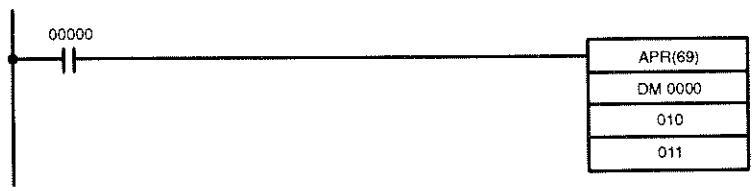


通道号	坐标点
C＋1	X_m(max, Xvalue)
C＋2	Y_0
C＋3	X_i
C＋4	Y_i
C＋5	X_2
C＋6	Y_2
↓	↓
C＋(2m＋1)	X_m
C＋(2m＋2)	Y_m

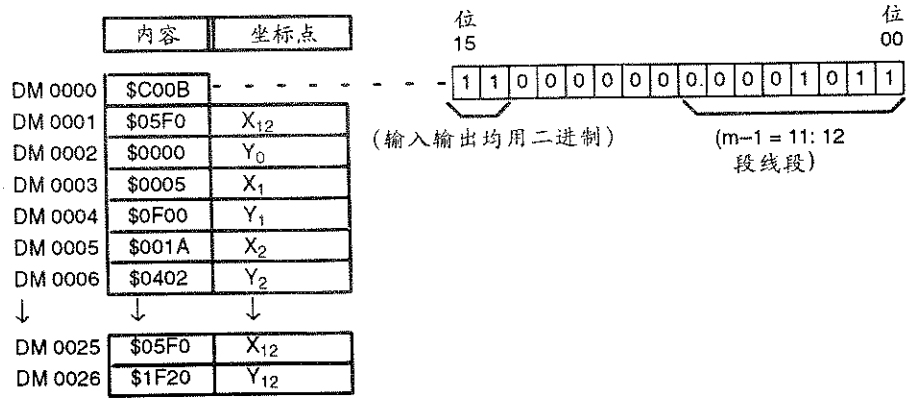
如果 C 的 13 位设为 ON，图形反射成从左到右，如下图示：



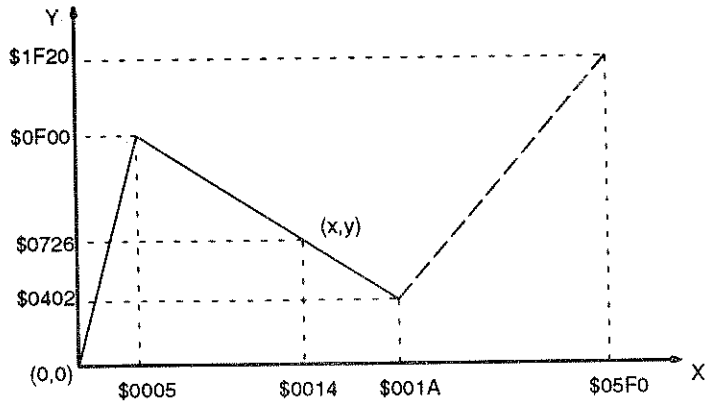
下例给出了用十二条直线段线性逼近的结构。数据块必须是从 DM0000 到 DM0026 连续地址。（C 到 C＋(2×12＋2)）。输入数据从 IR010 取，输出结果存入 IR011。



地址	指令	操作数
00000	LD	00000
00001	APR(69)	
		DM 0000
		010
		011

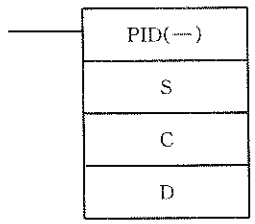


本实例中，通道号 IR010 输入数据，内容为＃0014，则 f（0014）＝＃0726 并输出到 R，IR011。



5－21－6　PID 控制指令－PID（－）

梯形符号



操作数数据区域

S:输入字
IR, SR, AR, DM, HR, LR

C:参数起始字
IR, SR, DM, HR, LR

D:输出字
IR, SR, AR, DM, HR, LR

限制

C 和 C＋32 必须在同一数据区。

注

在下列程序中不要使用 PID(－)指令，不然可能会产生不可预料的结果，它们是：中断程序，子程序，IL(02)和 ILC(03)之间，JMP(04)和 JME(05)之间，使用 STEP(08)和 SNXT(09)的步进程序。

说明

PID（一）指令根据指定参数进行 PID 控制。指令从输入字 S 的内容获取指定的二进制数据的输入范围，按照设定参数，进行 PID 运算，运算结果存放到输出字 D 中。

PID 参数字范围从 C～C＋32。PID 参数结构如下表。

字	15～12	11～8	7～4	3～0
C	设定值(SV)			
C+1	比例带(P)			
C+2	Tik=积分时间 T_i/采样周期 γ(见注 1)			
C+3	Tdk=微分时间 Td/采样周期 γ(见注 1)			
C+4	采样周期 γ			
C+5	2-PID 参数 α(见注 2)			PID 正向/反向设定
C+6	0	输入范围	时间单位	输出范围
C+7 to C+32	工作区域(不能被程序直接存取)			

注

1. 用 C+2 和 C+3 中所设定的值和 C+6 中设定的时间单元可以计算实际积分和微分时间。
2. 通常 2-PID 参数(α)设定值为 000,相应的 α 为 0.65。

参数设置

项 目	内 容	设定范围
设定值(SV)	受控过程的目标值	二进制数据(与指定输入范围位数相同)
比例带 P	是 P(比例)控制参数,为比例控制范围/整个控制范围。	0001～9999(4 位 BCD 码) (0.1%～999.9%,单位 0.1%)
TiK	是个常数,用来描述积分控制强度,该值越大,积分作用越小 时间单位参数决定设定方法	0001～8191(4 位 BCD 数); (9999 为无积分控制) 1×～8191× 当时间单位=0 或 1 0.1～819.1s 当时间单位=80.1～81.9s当时间单位=9
TdK	是个常数用来描述微分控制强度,该值越大,微分作用越强 时间单位参数决定设定方法	0001～8191(4 位 BCD 数) (0000 为无微分控制) 1×～8191× 当时间单位=0 或 1 0.1～819.1s 当时间单位=8 0.1～81.9s 当时间单位=9
采样周期	执行 PID 运算的周期	0001～9999(4 位 BCD 数) (0.01～99.99s,单位 0.01s)
PID 正向/反向设定	确定比例控制的方向	(一位 BCD 数)0:反向 1:正向
2-PID 参数(α)	输入滤波系数。通常使用 0.65(即设定值 000)。当系数接近 000 时,滤波作用减弱	000:α=0.65 若设定为 100～199,根据设定值低两位数决定:α=0.00～0.99(三位 BCD 数)
输入范围	输入数据的位数	0:8 位 5:13 位 1:9 位 6:14 位 2:10 位 7:15 位 3:11 位 8:16 位 4:12 位 (一位 BCD 数)
时间单位	指定积分/微分参数设定方法	0,1,8 或 9(1 位 BCD 数) 0 或 1:时间设定 8:时间设定(100ms/单位) 9:时间常数(10ms/单位)
输出范围	输出数据的位数	和输入设定范围相同

PID 控制运算

指令执行条件为 OFF

所有设定值保持不变，当执行条件为 OFF，通过把控制量写入输出字，可以进行手动控制。

执行条件的上升沿

根据设定的 PID 参数，工作区域被初始化，PID 控制运算开始，在刚开始运行时为避免控制系统受反向冲击(无冲击运行)，运算输出值不发生突变和大幅变化。

当 PID 参数更改时，指令执行条件从 OFF 变为 ON，更改参数才开始有效。

执行条件为 ON 状态。

PID 运算是按采样周期的时间间隔执行的，该采样周期的设置由 PID 参数决定的。

采样周期和 PID 执行时间

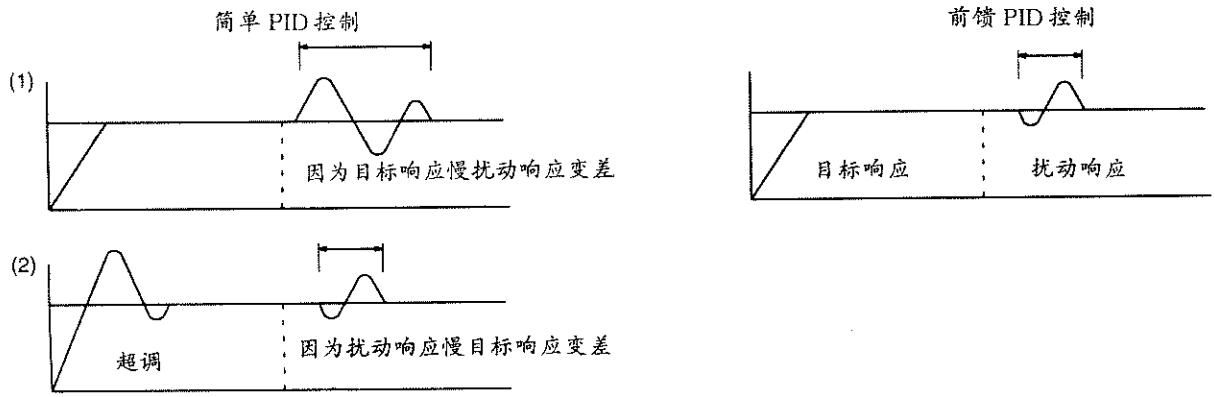
采样周期是采集测量数据，为执行 PID 运算用的间隔时间。然而，PID（一）指令根据 CPU 循环时间执行，所以可能有超过采样周期情况出现。这种情况下，指令执行要等到下次采样时间到。

PID 控制方法

PID 控制采用前馈 PID 控制方式（二级自由度）。

当采用简单 PID 方式为避免超调时，会引起扰动的稳态过程缓慢（见(1)）。另一方面，如果加快扰动的稳态过程，超调就会出现，达到目标值过程就会变长（见 (2)）。

用前馈 PID 控制，就可避免超调，加快达到目标值和扰动稳定过程(见(3))。

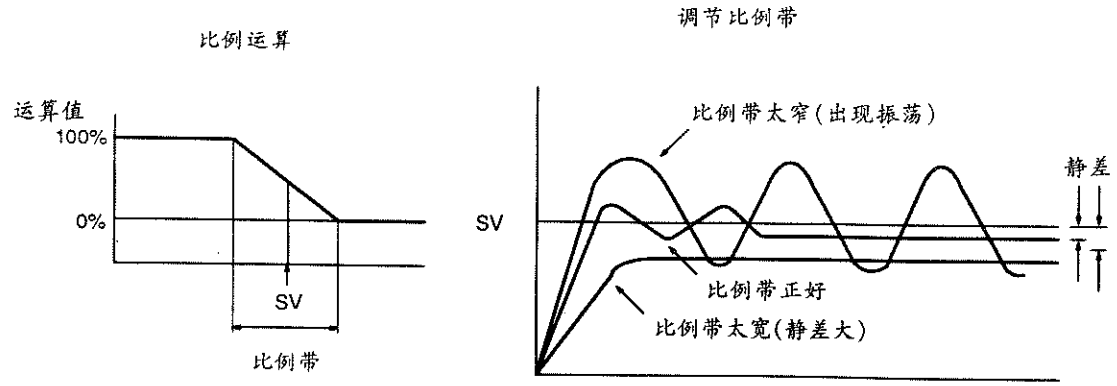


控制运算

比例运算（P）

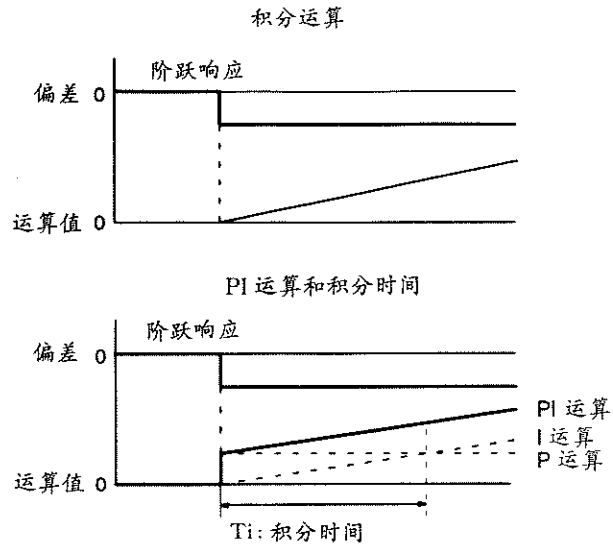
比例控制是建立与设定值（SV)相关的比例带的一种运算，并根据偏差比例在比例带求得运算值(控制输出量)。如果当前值（PV）小于比例带，运算值为 100%。如果当前值在比例带内，运算值根据偏差比例求得并逐步减小直到 SV 和 PV 区配（即，直到偏差为 0)，此时运算值回复到先前值(前馈运算)。

比例带表示为整个输入范围的百分比。若出现静差(残余偏差)，可用减小比例带方法减小残余偏差。如果比例带太小，反而会出现振荡。



积分运算 (I)

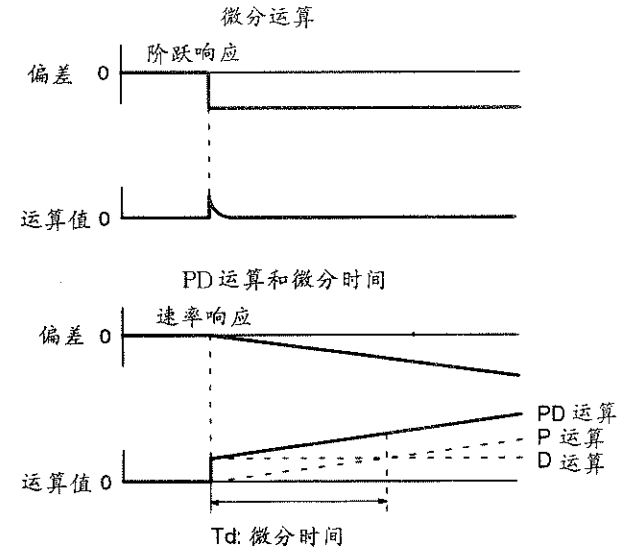
将积分与比例运算相结合，随着调节时间延续可减小静差。积分强度用积分时间表示，积分时间相当于积分运算值达到比例运算值在阶跃偏差响应下达到的作用所需要的时间，如下所示。积分时间值越小，积分运算的校正作用越强。但如果积分时间值太小，校正作用太强会产生振荡。



微分运算(D)

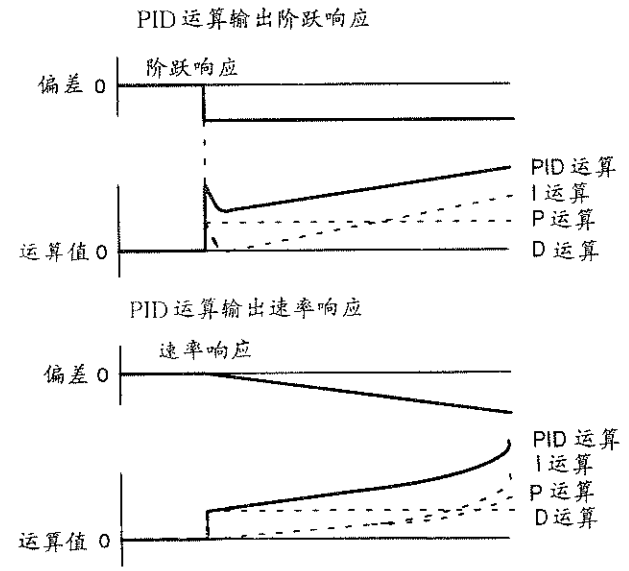
比例和积分运算都校正控制结果，所以不可避免地会产生响应延时现象。微分运算可弥补这些缺陷。在一个突发的干扰响应中，微分运算提供了一个很大的运算值，以使迅速恢复原始状态。微分运算采用一个正比于偏差变化率(微分系数)的运算值校正控制。

微分运算的强度由微分时间表示，如下所示，微分时间相当于微分运算值达到比例运算值在阶跃偏差响应下达到的作用所需的时间。微分时间值越大，微分运算的校正强度越强。



PID 运算

PID 运算是将比例(P),积分 (I),微分 (D) 运算结合。即使控制对象带有呆滞时间,PID 也能提供很好的控制结果。它用比例运算使控制平滑没有振荡,用积分运算自动修正静差,用微分运算加速干扰响应。



运算方向

当使用 PID 运算,选择以下两个控制方向中的任何一个。在任一方向,当给定值 SV 和当前值 PV 偏差增加,运算值增加。

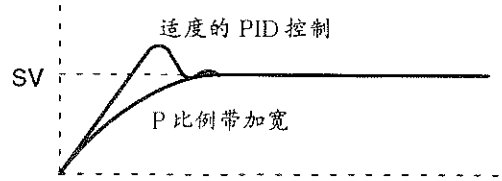
·正向运算:当 PV 大于 SV,控制值增加;

·反向运算:当 PV 小于 SV,控制值增加。

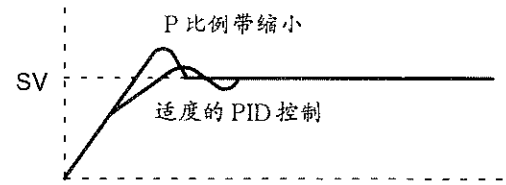
PID 参数调节

PID 参数间的一般关系和控制状态见下述。

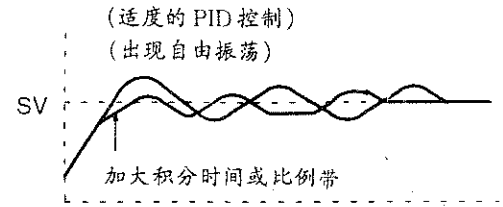
·如果对稳定时间没有要求,但不允许出现超调,那么,加宽比例带。



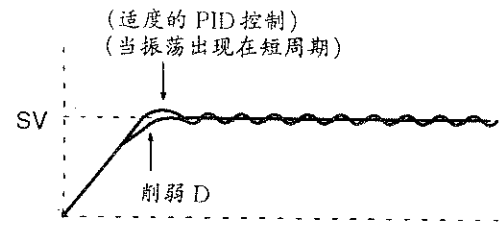
·如果有点超调没问题，但希望快速稳态控制，那么缩小比例带宽度。如果比例带太窄，那么就会出现振荡。



·当出现等幅振荡，或者运算受到过调和欠调限制，通常是由于积分运算太强。如果增加积分时间或加宽比例带，振荡会减弱。



·如果振荡出现，振荡周期短，可能是控制系统响应太快和微分运算太强。这种情况下，削弱微分运算强度。



标志

ER：＊DM 字内容不是 BCD 码，或者超出 DM 区域。

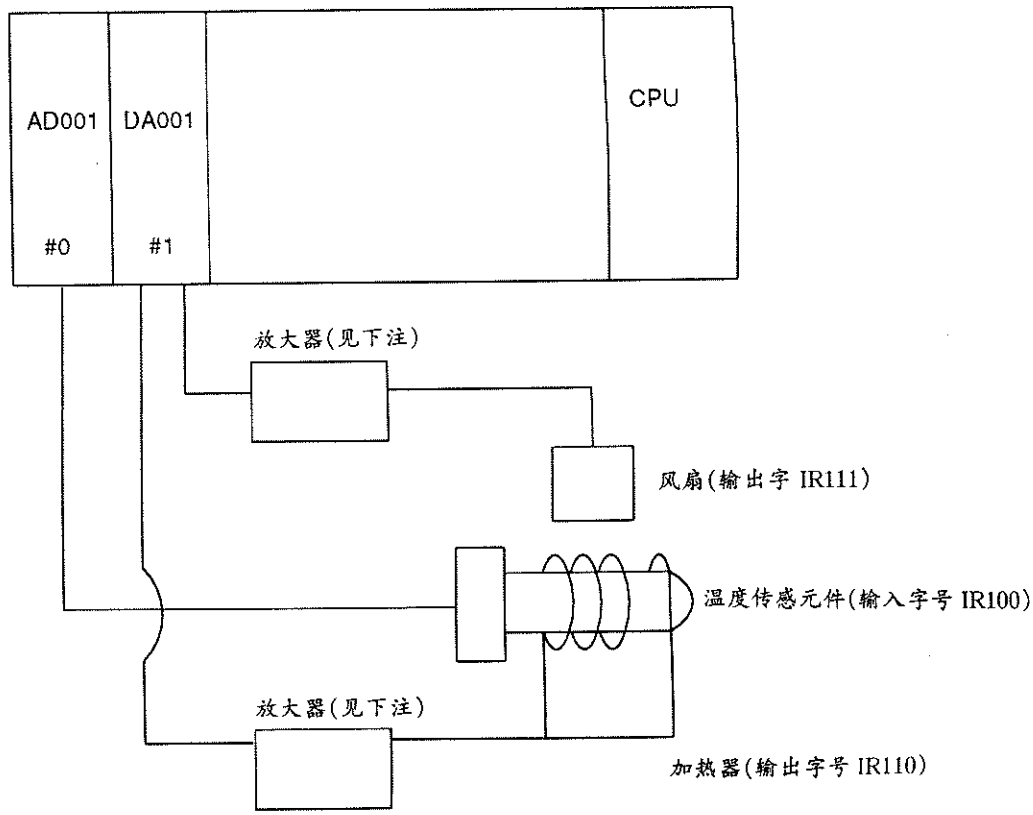
PID 参数 SV(设定值) 超出范围。

PID 运算执行，但运行时间是采样周期的两倍。这种情况下，即使 ER (SR25503)为 ON，PID (—)指令仍将执行。

CY：PID 运算正在执行，CY 为 ON。

例

这个例子给出了使用 PID (—) 指令的 PID 控制程序。



注

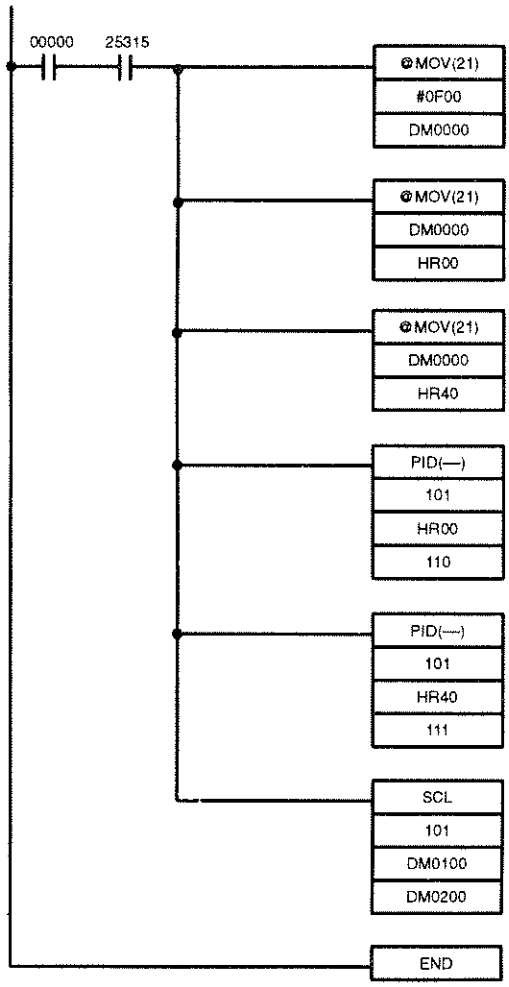
马达和加热器不能直接与模拟量输出单元连接,需要一个放大器(即一个电流放大器)

程序编制

程序编制按下面概要步骤所示。

1. 设定目标值(二进制 0000～0FFF)存入 DM0000。
2. 输入温度传感器元件的当前值 PV(二进制 0000～0FFF)至 IR101 字的 00～11 位中。
3. 根据下面程序中的第一条 PID (—)指令,输出加热器的运算值在 IR110 字的 00～11 位。
4. 根据下面程序中的第二条 PID (—)指令,在 IR111 通道号的 00～11 位输出风扇的运算值。
5. 用 SCL (—)指令将温度传感元件的当前值 PV(二进制 000～FFF)转换成温度数据(0000℃～0200℃),并输出到 DM0200。

程序



目标值

第一个 PID 指令参数字

第二个PID指令参数起始字

温度传感元件的当前值 PV

加热器运算值

风扇运算值

温度传感元件的当前值 PV
(二进制) 参数转换起始字
温度传感元件的当前温度(℃)

注意

在使用 PID(－)或 SCL (－)指令之前，应通过外围设备，例如编程器或 SSS(中文版)输入设定值。

加热器

HR00	(DM0000)	目标值 HR
HR01	0080	比例带
HR02	0200	积分时间/采样周期
HR03	0100	微分时间/采样周期
HR04	0001	采样周期
HR05	0000	正/反向标志/PID 参数
HR06	0404	I/O 范围和时间单位设置

SCL 参数

DM0100	0000
DM0101	0000
DM0102	0200
DM1003	0FFF

风扇

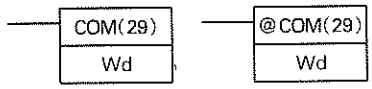
HR40	(DM0000)
HR41	0060
HR42	0150
HR43	0100
HR44	0001
HR45	0001
HR46	0004

5－22 逻辑指令

逻辑指令——COM (29)，ANDW (34)，ORW (35)，XORW (36)和 XNRW (37)——字数据执行逻辑操作

5－22－1 求反指令－COM(29)

梯形符号



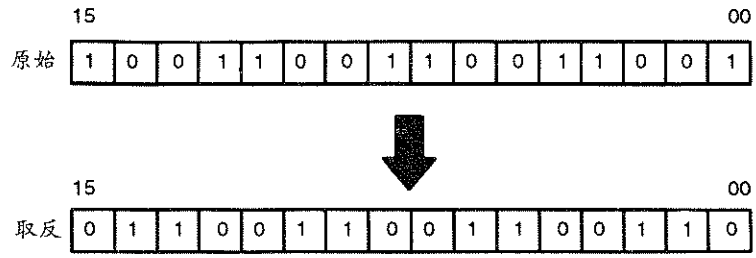
操作数数据区域

Wd：求反字
IR，SR，AR，DM，HR，LR

说明

当执行条件为 OFF，COM (29) 不执行。当执行条件为 ON 时，COM (29) 将字中所有 ON 位复位，将所有 OFF 位置位。

举例



标志

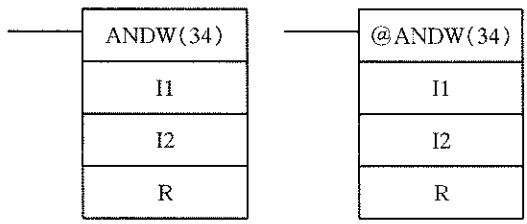
ER:间接寻址的 DM 字不存在。(＊DM 字内容不是 BCD 码,或者超出 DM 区域)

EQ:当结果为 0 时置 ON。

N:当字中 15 位置"1",N 为 ON。

5－22－2 "逻辑与"指令－ANDW (34)

梯形符号



操作数数据区域

I1:输入 1
IR, SR, AR, DM, HR, TC, LR, #

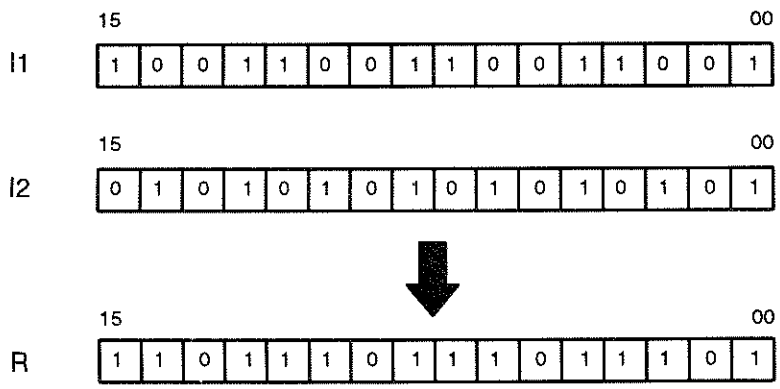
I2:输入 2
IR, SR, AR, DM, HR, TC, LR, #

R:结果字
IR, SR, AR, DM, HR, LR

说明

当执行条件为 OFF 时,ANDW (34) 不执行,当执行条件为 ON 时,ANDW (34) 将 I1 和 I2 的内容逐位逻辑与,并把结果存入 R 中。

举例



标志

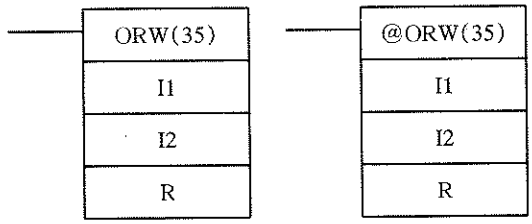
ER：间接寻址的 DM 字不存在。（＊DM 字内容不是 BCD 码，或者超出 DM 区域）

EQ：当结果为 0 时置 ON。

N：当 R 中 15 位置“1”，N 为 ON。

5－22－3 “逻辑或”指令－ORW (35)

梯形符号



操作数数据区域

I1：输入 1
IR, SR, AR, DM, HR, TC, LR, #

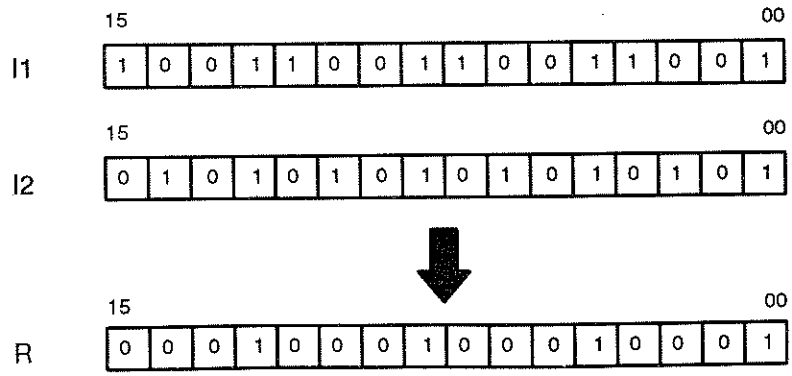
I2：输入 2
IR, SR, AR, DM, HR, TC, LR, #

R：结果字
IR, SR, AR, DM, HR, LR

说明

当执行条件为 OFF 时，ORW(35)不执行。当执行条件为 ON 时，ORW(35)将 I1 和 I2 的内容逐位逻辑或，并把结果存入 R 内。

举例



标志

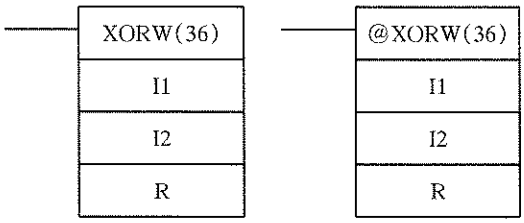
ER：间接寻址的 DM 字不存在。（＊DM 字内容不是 BCD 码，或者超出 DM 区域）

EQ：当结果为 0 时置为 ON。

N：当 R 中 15 位置“1”，N 为 ON。

5－22－4 “异或”门逻辑指令－XORW（36）

梯形符号



操作数数据区域

I1：输入 1
IR，SR，AR，DM，HR，TC，LR，#

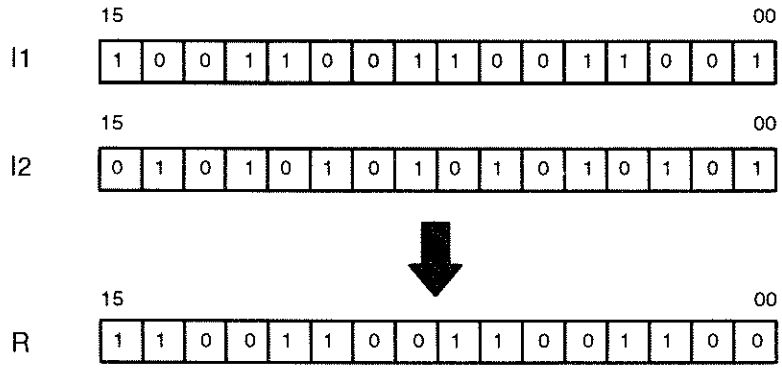
I2：输入 2
IR，SR，AR，DM，HR，TC，LR，#

R：结果字
IR，SR，AR，DM，HR，LR

说明

当执行条件为 OFF 时，XORW（36）不执行。当执行条件为 ON 时，XORW（36）将 I1 和 I2 的内容逐位进行异或运算并把结果存入 R 中。

举例



标志

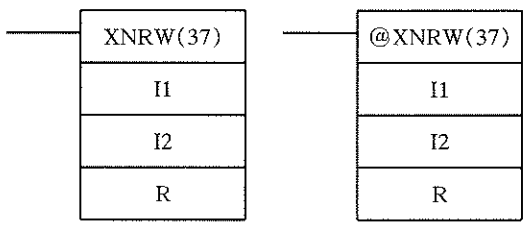
ER：间接寻址的 DM 字不存在。（＊DM 字的内容不是 BCD 码，或者超出 DM 区域）

EQ：当结果为 0 时置为 ON。

N：当 R 中 15 位置“1”，N 为 ON。

5－22－5 “异或非”门逻辑指令－XNRW（37）

梯形符号



操作数数据区域

I1：输入 1
IR，SR，AR，DM，HR，TC，LR，#

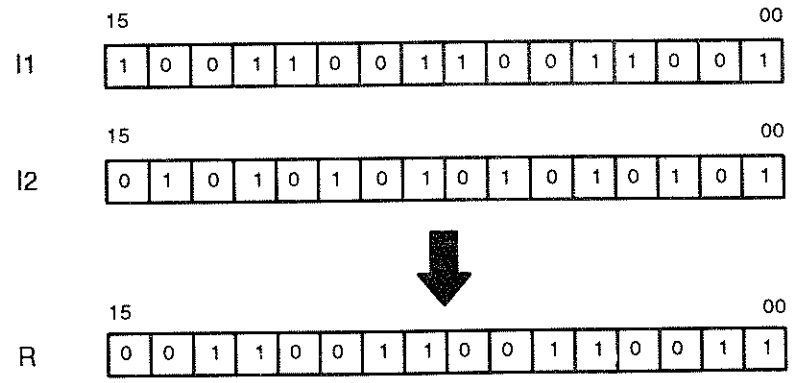
I2：输入 2
IR，SR，AR，DM，HR，TC，LR，#

R：结果字
IR，SR，AR，DM，HR，LR

说明

当执行条件为 OFF 时，XNRW（37）不执行。当执行条件为 ON 时，XNRW（37）将 I1 和 I2 的内容逐位进行“异或非”运算，并把结果存入 R 中。

举例



标志

ER：间接寻址的 DM 字不存在。（＊DM 字内容不是 BCD 码，或者超出 DM 区域）

EQ：当结果为 0 时置 ON。

N：当 R 中 15 位置“1”，N 为 ON。

5－23　子程序和中断控制指令

5－23－1　子程序指令

子程序将大量的控制任务分离成许多小块的控制任务，并可使你能重复使用一组给定的指令。当主程序调用一个子程序时，控制就转到子程序程序，执行子程序指令。子程序指令码编制与主程序指令码一样。当所有的子程序指令执行完毕，程序返回到主程序中调用子程序点后的一条指令处。（除非子程序另有指定）

子程序也可以通过中断或 MCRO（99）指令起动。

中断

同子程序的调用一样，在主程序运行流程过程中，中断引起主程序运行停顿，同样子程序执行完后，运行再从这点后继续下去。一个中断要么是由一个外部源如来自于中断输入入单元的一个输入信号产生的，要么就是由一个按定时中断产生的，在定时中断的情况下，中断信号按固定的时间间隔重复产生。

鉴于调用子程序是受控于主程序的，被中断起动的子程序在接收到中断信号时就起动了。在定时中断的情况下，中断之间的时间间隔是用户设置的，与 PC 的扫描周期无关，这个性能有利于周期监视或控制程序运行。

INT（89）用于控制从中断输入单元接收到的中断信号，也控制定时中断的安排。INT（89）提供中断屏蔽（这样中断被记忆下来但不响应）和中断清除功能。

有关中断详见 5－23－2“中断”指令章节。

宏(MCRO)(99)指令

宏指令允许一个单独的子程序（程序格式）替代几个具有相同结构不同操作数的子程序。因为一些类似程序段可用一个子程序处理，这样程序的步数可大大地减少。这条指令详见 5－23－5“宏指令—MCRO(99)”章节。

5－23－2　中断指令

C200HX/HG/HE PC 提供输入中断和定时中断。中断输入（来自一个中断输入单元）必须立即停止主程序去运行子程序，或必须周期执行子程序（定时中断）。

有二种中断方式。在一般中断方式，CPU 等到当前处理完成之后停止主程序。在高速中断方式，CPU 中断当前处理过程。在 C200HX/HG/HE PC 中，一般中断方式为缺省方式，可在 PC 设置中选择高速中断方式。

一般中断方式

以下设置用于一般中断方式：

DM6620 | 0 | 0 | 0 | 0 |

在一般中断方式中，下列过程一旦开始，必须全部完成，即使出现中断也仍将继续运行完毕，等该过程执行完毕则马上执行中断程序。

·上位机链接服务

·远程 I/O 服务

·特殊 I/O 单元服务

·单个的指令执行

当不修改地使用 C200H 中断子程序或考虑到中断的响应时间可以接受时使用这种方式。

注

一般中断方式是 C200HX/HG/HE 的缺省设置。

高速中断方式

如下设置用于高速中断方式

DM6620 | 1 | － | － | － |

中断方式
（1＝高速）

在高速中断方式中，当中断一产生，下述过程就会被中断并执行中断子程序。

·上位机链接服务

·远程 I/O 服务

·特殊 I/O 单元服务

·单个的指令执行

当中断响应时间必须精确到 1.0ms，使用这种方式。

如果使用高速中断方式，那么数据就不能有一致性要求。因为当上位机链接服务、特殊 I/O 单元服务，远程 I/O 服务和单独的指令执行开始工作时不一定运行完毕。当应用中要求这一点就必须编写出满足这一要求的程序。（要更多详情可阅数据一致性章节。）

输入中断

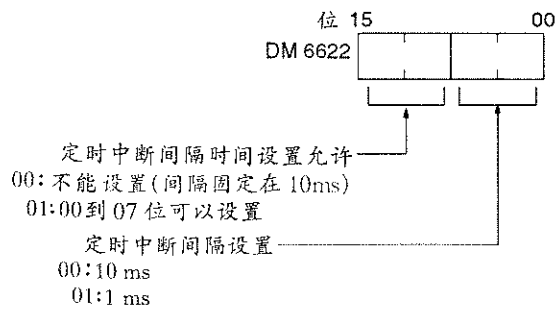
当外部中断在中断输入单元被接收时执行输入中断。在 CPU 机架上最多可安装二个中断输入单元，每个中断输入单元提供 8 个输入，IN0～IN7。

第 1 个中断输入单元上输入 IN0～IN7 产生＃00～＃07 中断，第 2 个中断输入单元上输入 IN0～IN7 产生＃08～＃15 中断。一般地说，当产生＃00～＃15 中断时，执行＃00～＃15 子程序。

定时中断

定时中断可以按设定的时间间隔执行，该时间间隔可以是 10ms 的倍数或 1ms 的倍数。使用中断＃99 并执行子程序＃99。

PC 设置中 DM6622 是设置定时中断时间间隔的单位。



中断优先权

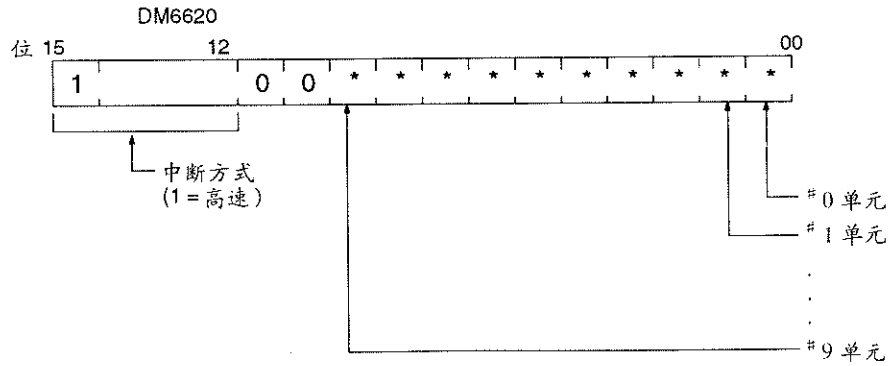
当中断产生时，就执行规定的子程序。在一个中断子程序执行期间如果产生更多的中断，它们只有在当前中断子程序运行完毕后才能被处理。如果同时产生一个以上的中断或同时等待运行，那么相应的子程序将按以下优先权的次序运行。

输入中断 1＞输入中断 2＞……＞输入中断 7＞定时中断

中断子程序中的特殊 I/O

在中断子程序中可使用 I/O 刷新(IORF)指令，刷新用于特殊 I/O 单元中的 I/O。如果用高速中断方式，常规扫描周期的刷新（主程序中的 END 刷新和 IORF 刷新)不允许对中断子程序才能刷新的特殊 I/O 单元刷新。如果相同的特殊 I/O 在中断程序和常规扫描周期中都刷新，中断程序错误就会产生(系统 FAL 错误码 8B)，特殊 I/O 也不会在中断子程序中得到刷新。

PC 设置将常规扫描中不允许对特殊的专用 I/O 单元刷新的设置安排在 DM6620 中。这个设置如下所示。



注

在高速中断方式下，由于在常规的扫描周期不允许对特殊 I/O 刷新。要对特殊 I/O 刷新需在中断子程序中，在一般中断方式下，特殊 I/O 在中断子程序和常规扫描周期中都会被刷新。因此不允许在常规扫描周期中刷新特殊 I/O 的设置是无用的。

如果使用高速中断方式且处理执行特殊 I/O 单元，上位机链接单元或远程 I/O 单元，中断子程序的执行时间必须小于 10ms。如果执行时间为 10ms 或大于 10ms，就会发生中断程序错误(系统 FAL 错误码 8B)。

执行时间最长的中断子程序执行时间被存放到 SR262，执行时间最长的子程序编号被输出到 SR263。

例如：子程序＃80 执行时间为 12.3ms

SR262 | 0 | 1 | 2 | 3 | 最长的中断子程序执行时间（0.1ms 为单位）

SR263 | 8 | 0 | * | * | 具有最长执行时间的中断子程序编号。

注

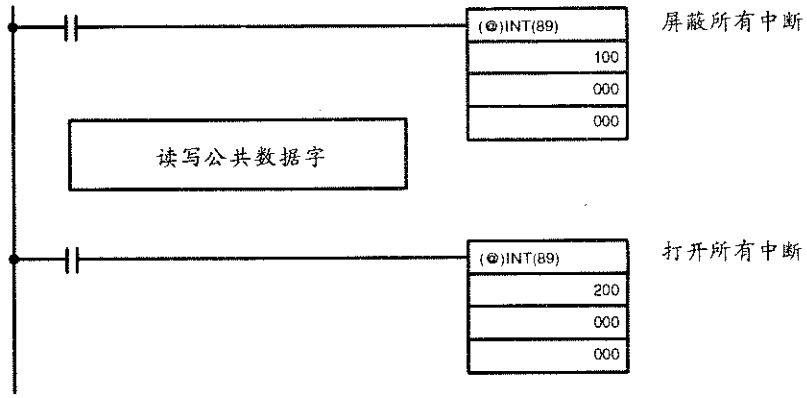
当采用一般中断方式或未安装上述单元时，上述 10ms 的限制也就不存在。

数据一致性

当采用高速中断方式时，尽管数据一致对于常规算术指令或比较指令的执行不是一个问题，但当执行多字节的较长指令时，例如块传输指令，同一数据被主程序和中断子程序同时处理时，数据一致性就会成为一个问题。

在两种不同条件下数据也许不一致：1）如果主程序中的数据写操作被中断，同一数据在中断子程序中又被读入，2）如果主程序中的数据读操作被中断，同一数据在中断子程序中又被写入。

如果你必须在主程序和一个中断子程序中处理同一数据时，采用如下所示的程序以确保数据一致性，即主程序在读/写中断子程序也要处理的数据时，屏蔽中断。



如果在特殊 I/O 单元，远程 I/O 或上位机链接系统服务处于数据传输期间产生了中断，数据一致也可能是一个问题，对于这些中的任一个而言，字节单元数据也可出现不一致。

采用以下方法中的任一个可维持上述条件下的数据一致性，第二个方法仅适用特殊 I/O 单元。

·当主程序传送从单元接收或发送不同字的数据，而在中断子程序中也使用这些变化的字时，主程序屏蔽中断。

·在中断子程序中使用 I/O 刷新指令刷新特殊 I/O 单元所需的 I/O，主程序读/写特殊 I/O 单元字时屏蔽中断。

5－23－3 子程序调用指令－SBS (91)

梯形符号

SBS (91) N

定义符数据区域

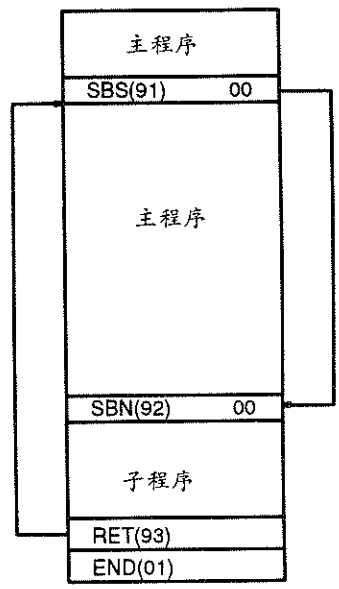
N：子程序编号
00 至 255

限制

子程序编号 00 至 15 与输入中断同时使用，子程序编号 99 被用于定时中断。

说明

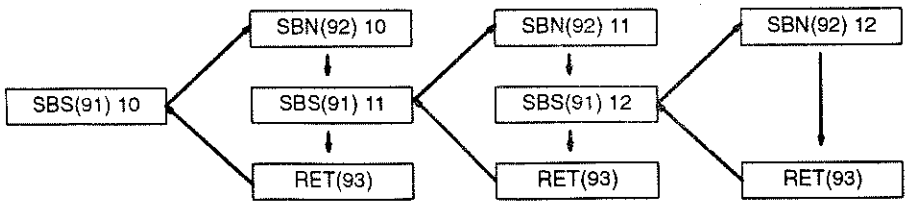
在主程序中将 SBS (91) 指令放在要求执行子程序处，这子程序将会执行。SBS(91)中采用的子程序编号表明了所要调用的子程序。当执行 SBS(91)时(即当它的执行条件为 ON 时)，PC 执行和 SBS(91)有相同子程序编号的 SBN (92)和下面第一条 RET(93)指令间的程序，并返回到调用的 SBS(91)指令后的程序中断点。



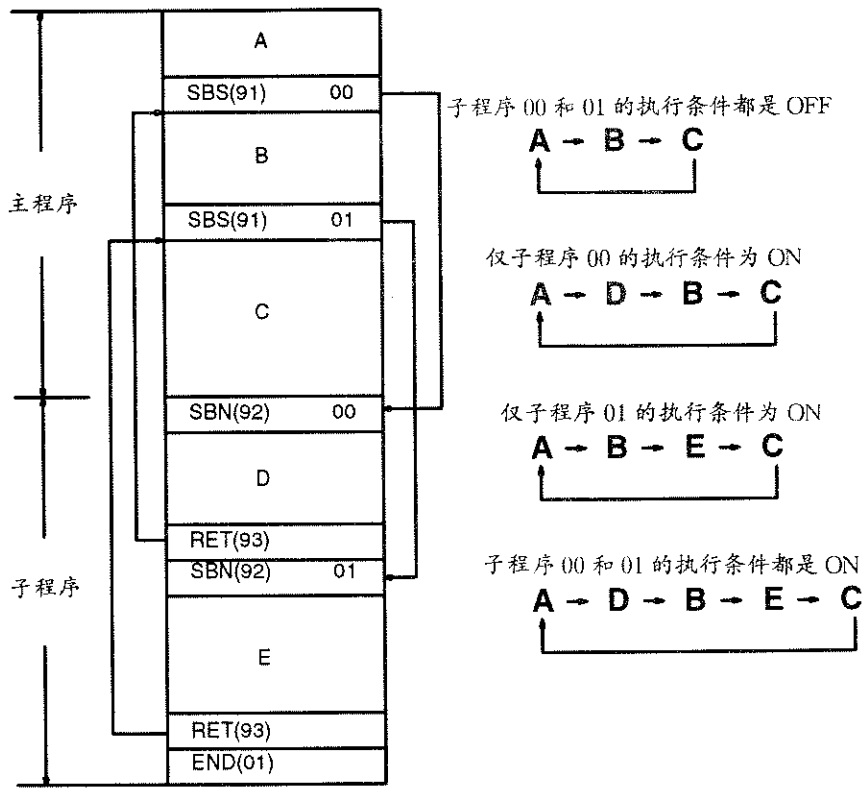
SBS(91)在程序中可被多次调用，即在程序中同一子程序可被不同地方调用。

SBS(91)也可被放于子程序中，使程序执行从一个子程序转移至另一个子程序，即子程序可以嵌套。当第二个子程序执行完毕时（即已执行到了 RET(93)），程序执行就会返回到调用源子程序，然后该子程序执行完毕后返回主程序。嵌套级数无限制，不象 C200HS PC 限制 16 级嵌套。

一个子程序不能调用其自身（例如在 SBN（92）00 的子程序中不能调用 SBS（91）00）。下图说明了 2 级嵌套执行过程。



下图说明二条 SBS（91）指令在各种执行条件下的程序流程。



注

如果一段子程序的执行时间超过 10ms，将会产生一个非致命错误（错误码 8B）。

标志

ER：指定子程序编号的子程序不存在。

子程序自调用。

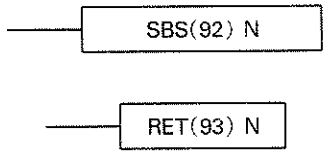
运行的子程序被调用。

注意

当 ER 为 ON 状态，SBS(91) 不能执行，子程序不能调用。

5－23－4 子程序定义和返回指令－SBN (92) /RET (93)

梯形符号



定义数据区

N:子程序编号
00 至 255

限制

在 SBN(92)指令中,每个子程序编号只能使用一次。

说明

SBN (92)用来表示一个子程序的开始;RET (93) 指令是表示一个子程序的结束。每一个子程序都用一个子程序编号 N 来标记,在编程时这编号 N 作为对 SBN (92)的定义符来使用,用 SBS (91) 指令调用同一子程序时使用相同的子程序编号 (参见 5－23－3"子程序调用指令——SBS (91)")。RET (93)指令不需要子程序编号。

所有的子程序都必须排在主程序的结尾。当编写一个或多个子程序,主程序执行到第一个 SBN (92)指令,返回 00000 地址,开始下一个周期。除非用 SBS (91) 指令来调用子程序,否则子程序不会执行。

END(01)指令必须放在最后子程序的末尾,即最后一条 RET(93)指令的后面。其他任何地方无需该指令。

注意事项

如果错将 SBN (92) 指令放置在主程序中,它将屏蔽此点以后的程序,即当遇到 SBN (92)指令时,程序就返回到开始处。

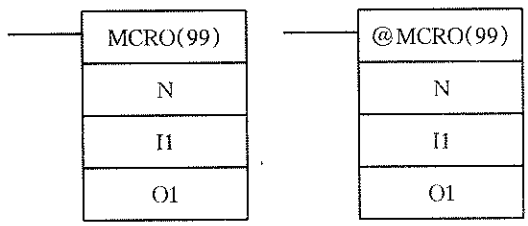
如果 DIFU(13)或 DIFU(14)设置在一段子程序内,直到下一次子程序执行前,操作数位将不会变为"OFF",即操作数位保持"ON"超过一个周期。

标志

这些指令不直接影响标志。

5－23－5 宏指令－MCRO (99)

梯形符号



操作数区

N:子程序编号
00～255

I1:第一个输入字
IR, SR, AR, DM, HR, TC, LR

O1:第一个输入字
IR, SR, AR, DM, HR, LR

限制

I1～I1＋3 必须在同一数据区,O1～O1＋3 也在同一数据区。

说明

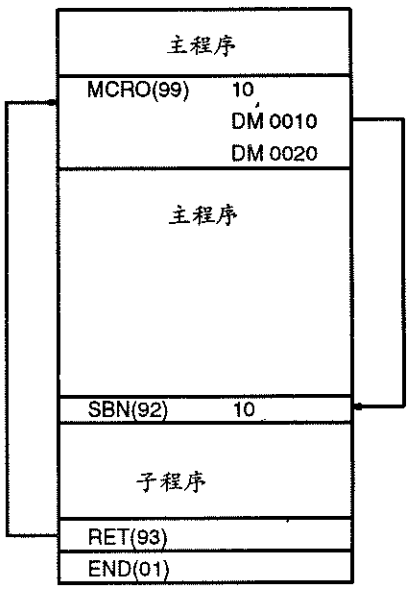
宏指令允许一单个子程序来取代数个子程序,这数个子程序须具有相同的结构,但操作数不同。有 4 个输入字(SR290～SR293)和 4 个输出字(SR294～SR297)分配给 MCRO (99)。这 8 个字是用于子程序,且当

子程序执行时可将其内容从 I1～I1＋3 和 O1～O1＋3 中取出。

当指令执行条件为"OFF"，MCRO (99)不执行：当执行条件为 ON，MCRO (99) 指令将 I1～I1＋3 中的内容复制到 SR290～SR293 中，将 O1～O1＋3 中的内容复制到 SR294～SR297 中。然后调用并执行编号为 N 的子程序。当子程序执行完了，SR294～SR297 中的内容将会在 MCRO (99)完成之前传送回 O1～O1＋3 中。

在下面例子中，DM0010～DM0013 的内容复制到 SR290～SR293 中，DM0020～DM0023 的内容复制到 SR294～SR297 中，并且调用和执行子程序 10。

当子程序执行后，SR294 到 SR297 的内容复制回 DM0020 到 DM0023 中



注

1．除 SR290～SR297 的内容是从指定的输入字或输出字中传送来之外，宏指令子程序编程与其它子程序相同。

2．不仅外部的 I/O 字，而且内部 I/O 字都能用于 I1 和 O1。

3．当 SR290～SR297 不用于宏程序时，它们可用来作为工作位。

4．MCRO(99)指令可以嵌套，但必须保留 I/O 数据，因为该指令使用 8 个相同 I/O 字(SR290～SR297)。

注意事项

MCRO (99)指令只能用于程序段，这程序段是用四个或少于四个连续的输入字和(或者)四个或少于四个连续的输出字来编写的，因此通常需一起考虑系统和程序设计，从而充分利用宏指令编程。

必须注意其输入字和输出字要严格地与宏输入字和宏输出字相对应。

标志

ER：指定编号的子程序不存在。

操作数超出数据区域。

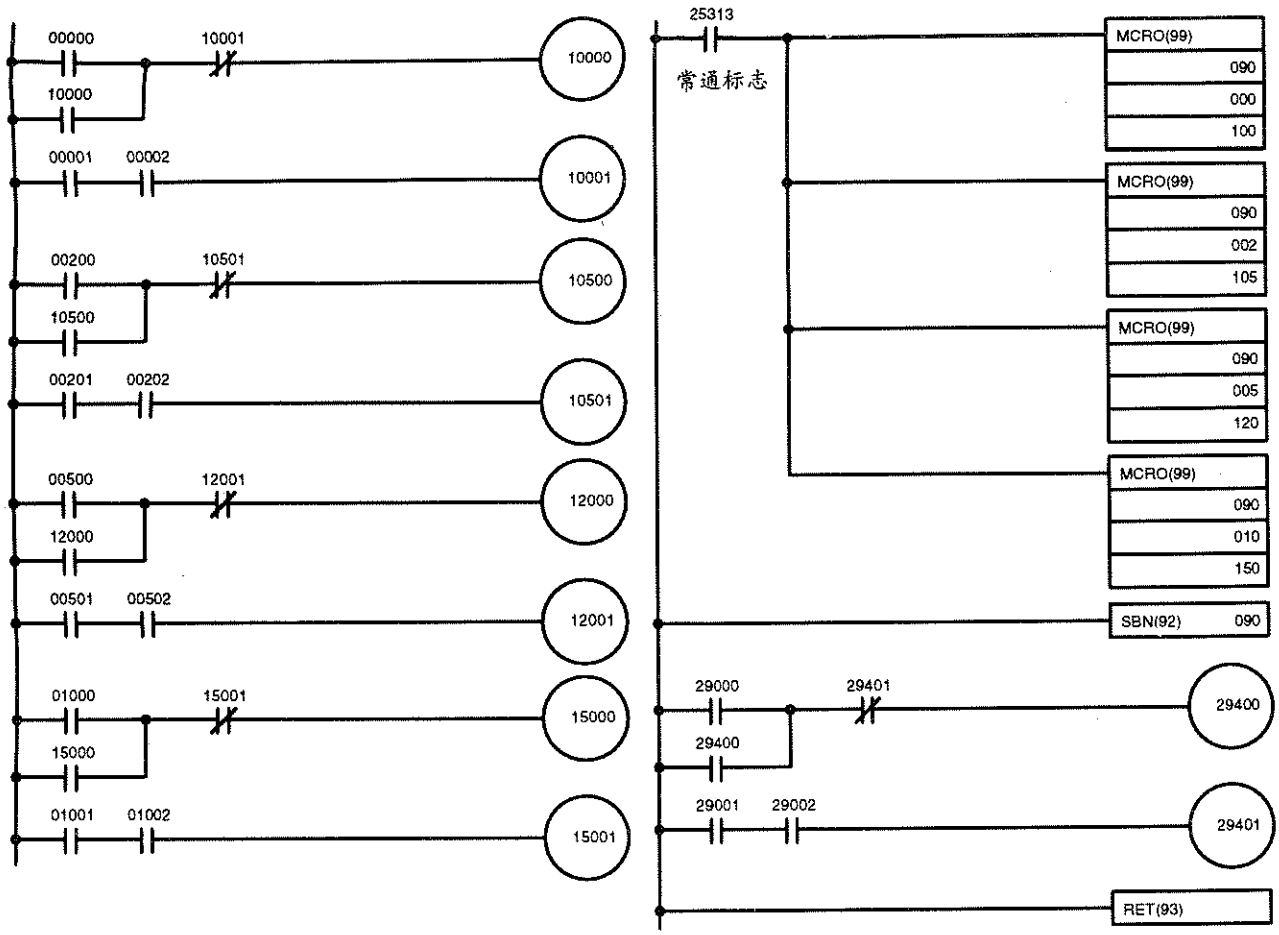
间接寻址 DM 字不存在。(如：＊DM 字的内容不是 BCD 码。或超出 DM 区域)

子程序自调用。

运行中的子程序被调用。

举例

下面的例子说明用四条 MCRO(99)指令进入相同的子程序，左边的程序段给出了不用 MCRO (99)指令但具有相同功能的程序。



5－23－6 中断控制指令－INT(89)

梯形符号

INT(89)
C
N
D

@INT(89)
C
N
D

操作数数据区域

C:控制码
#(000, 001, 002, 100, 或 200)

N:中断类型
#(000,001,或 004)

D:控制数据
IR, AR, DM, HR, TC, LR, #

限制

当 N＝000 和 C＝000 或 001 时，D 必须在＃0000 和＃00FF 之间。

当 N＝004 和 C＝000 或 001 时，D 必须是在＃0001 和＃9999 之间的 BCD 码。

说明

INT(89)用来控制中断，能完成 11 种中断功能的其中一种，这 11 种功能取决于 C 和 N 的值。如下表所

示，其中六种用作输入中断，三种用作定时中断，另外二种用于屏蔽或非屏蔽所有中断。

中断	C	INT(89)功能	内　容
来自中断输入单元 0 的中断输入(N＝000)	000	屏蔽/开放输入中断	D 的 00～07 位表示输入 00－07
	001	清除输入中断	
	002	读当前屏蔽状态	状态写入 D
来自中断输入单元 1 的中断输入(N＝001)	000	屏蔽/开放输入中断	D 的 00～07 位表示输入 00～07
	001	清除输入中断	
	002	读当前屏蔽状态	状态写入 D
定时中断(N＝004)	000	设置中断时间间隔	
	001	设定第一次中断时间	
	002	读中断时间间隔	

下面 2 种功能仅取决于 C 值。

C 值	INT(89)功能
100	屏蔽所有中断
200	开放所有中断

屏蔽/开放输入中断 (N＝000 或 001，C＝000)

对中断输入单元 0，设 N 为 000；对中断输入单元 1，设 N 为 001。该功能用于屏蔽和开放 00～07 输入中断。被屏蔽的输入将记录下来，但不响应。一旦某个被屏蔽输入被开放，它的中断程序马上执行（除非在它之前，执行了 C＝001 和 N＝000 的 INT(89)指令)。

设置 D 中与之对应的位为 0，则开放该输入；设置 D 中与之对应的位为 1，则屏蔽该输入。D 中的 00～07 对应 00～07 中断输入。

清除输入中断 (N＝000 或 001，C＝001)

对中断输入单元 0，设 N 为 000；对中断输入单元 1，设 N 为 001。本功能用来清除 00～07I/O 中断输入，由于中断输入已记录，被屏蔽的中断如不首先清除，当屏蔽去除时仍会响应中断。

设置 D 中相应的位为 1，清除一个中断输入。D 中的 00～07 对应于 00～07 中断输入。

读当前屏蔽状态 (N＝000 或 001，C＝002)

对中断输入单元 0，设 N 为 000；对中断输入单元 1，设 N 为 001。该功能用于将输入中断 00～07 的当前屏蔽状态写入字 D，如果输入被屏蔽，则对应的位是“ON”。D 中的 00～07 对应于 00～07 中断输入。

设置中断时间间隔 (N＝004，C＝000)

该功能用来设定定时中断之间的时间间隔，D 中的内容（BCD 码：0001～9999）与定时时间单位（1ms 或 10ms)相乘来获得这个中断的时间间隔。

中断定时时间单位在 PC 设置的 DM6622 中。有关时间单位设定的细节参见 3－6－4“PC 设置”。

设定首次中断的时间 (N＝004，C＝001)

该功能是用来设定首次定时中断的时间。D 中内容(BCD 码：0000～9999)与定时时间单位(1ms 或 10ms)相乘以获得第一次定时中断的时间。定时中断时间单位是设在 PC 设置的 DM6622 之中。设定该时间单位的方法参见 3－6－4 节“PC 设置”。

注意第一次中断时间的设置，如果这个时间没有设置，第一次中断的时间间隔（由 N＝004，C＝000 设定)就不确定。

当设定了首次中断时间（C＝001)，使用首次扫描标志(SR25315)作为执行 INT(89)指令的执行条件。如果连续设置 C＝001，则定时中断可能永远不会产生。

读中断时间间隔 （N＝004，C＝002）

该功能用来将当前的定时时间间隔写入 D。

屏蔽/开放所有中断 （C＝100/200）

该功能用于屏蔽和开放所有的中断处理。屏蔽的中断输入被记录下来，但不响应。一旦中断开放，这些被屏蔽的输入中断将会执行。该功能同时屏蔽或开放所有的中断，而且与其它功能所产生的屏蔽无关。

在本功能中不使用控制数据 D 和中断类型 N，它们都设为＃0000。

标志

ER：间接寻址 DM 字不存在。（＊DM 字的内容不是 BCD 码，或者超过 DM 区域）

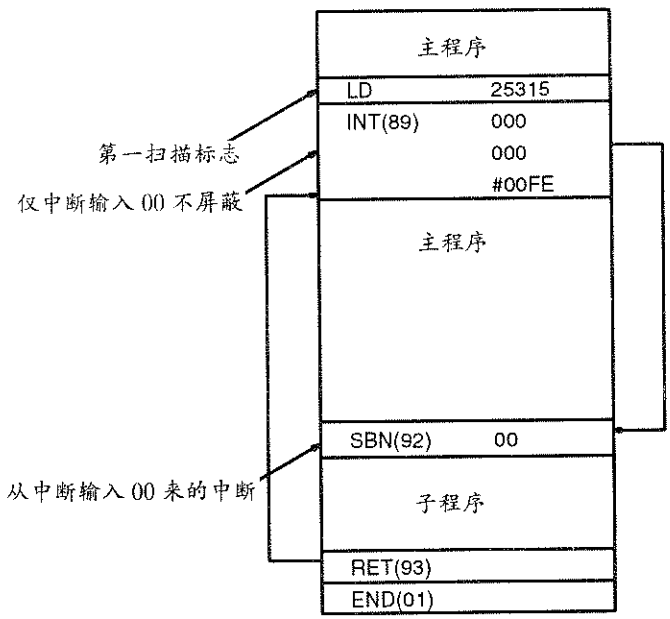
C 和/或 N 数值不在指定值之内。

子程序中所有中断被屏蔽或被开放。

举例 1：输入中断

该例说明如何开放某个特定的中断输入。当 CPU 收到相应的中断输入，无论 CPU 扫描处于何处与之对应的中断子程序都将执行。当使用长度不定的程序段时，例如事件程序，这些中断是很有用的。

在运行开始时所有的中断都被屏蔽，而所需的中断输入用 N＝000 和 C＝000 的 INT(89)指令来开放。如下图所示，如果从中断输入单元 0 的中断输入 00 有输入，当该输入中断开放时，子程序将被执行。



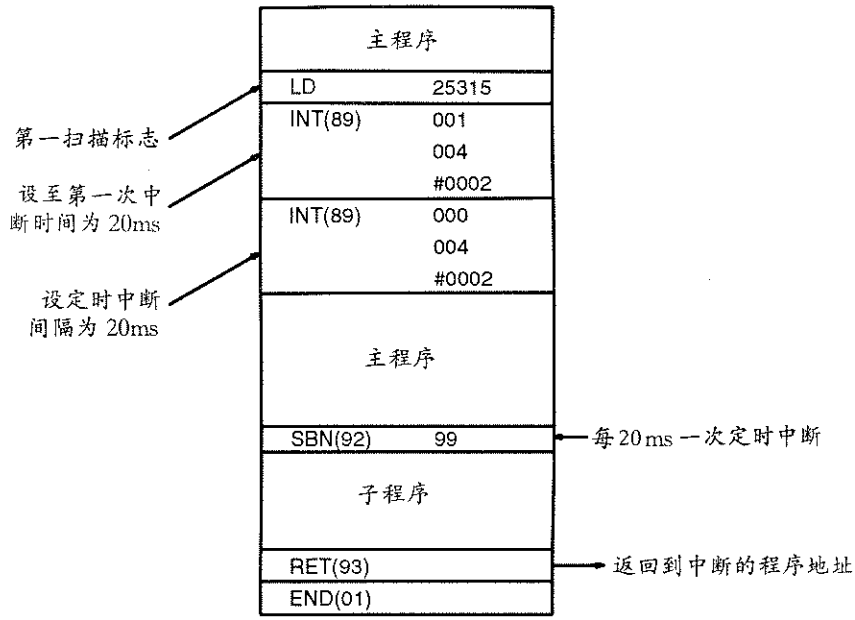
注

取决于 PC 设置中 DM6621 的设置，上位机链接服务、远程 I/O 服务，特殊 I/O 单元服务和单个指令的执行将在子程序执行前完成，细节参见 268 页。

举例 2：定时中断

该例子说明如何设置定时中断之间的时间间隔。无论 CPU 扫描处于何处，定时中断子程序都按固定时间间隔执行。在诸如常规监视程序段中该中断是很有用的。

运行开始时，定时中断被禁止（定时中断时间间隙为 0），因此到第一次中断的时间以及定时时间间隔必须用 N＝004 和 C＝001/000 的 INT(89)指令来设定。在下图中，假定 PC 设置中的 DM6622 设定为 10ms，子程序每隔 20ms 执行一次。



注

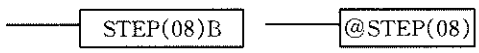
根据 PC 设置中的 DM6621 的设置，上位机链接服务、远程 I/O 服务、特殊 I/O 单元服务和单个指令的执行将在子程序执行前完成，细节参见 268 页。

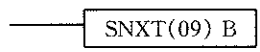
5－24 步指令

STEP(08)和 SNXT(09)步指令一起使用，在一个大型程序中程序段间设置断点。以便使这些程序段作为一个整体执行，并在执行完毕后复位。一段程序通常和实际应用中的一个过程相对应(参阅本节后面的应用例子)。除了某些指令（如 IL(02)/ILC(03)、JMP(04)/JME (05))之外，一条步指令如同普通的一个程序代码。

5－24－1 步的定义和起动指令－STEP(08)/SNXT(09)

梯形符号





定义数据区

B:控制位
IR, SR, AR, HR, LR

B:控制位
IR, SR, AR, HR, LR

限制

所有控制位必须在同一个字中，并且要连续。

IR29800～IR29915 不能用于 B。

说明

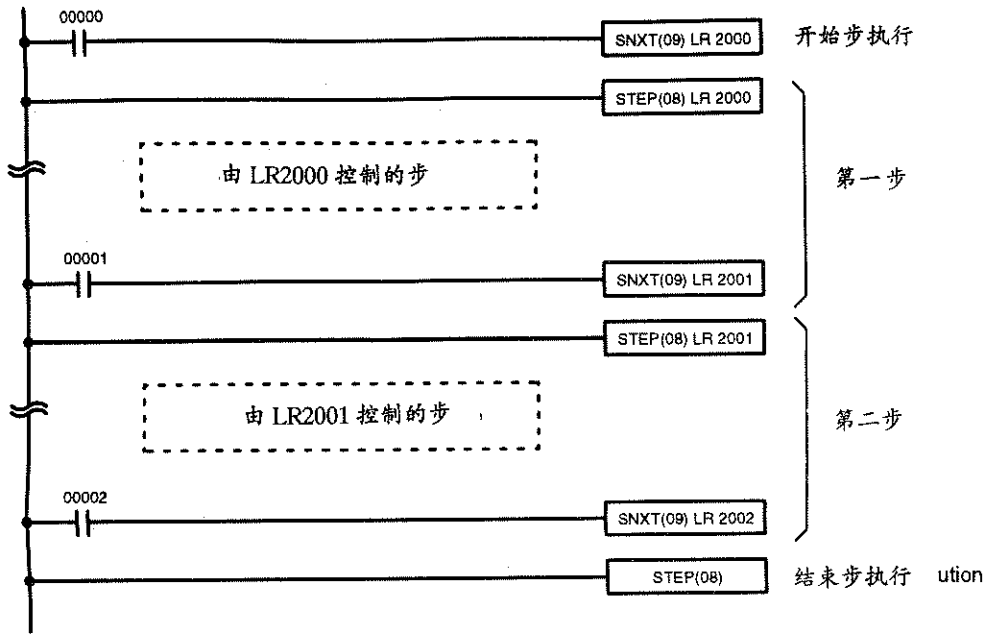
STEP (08) 用 IR 或 HR 区域内的一位控制位来定义一个程序段(称为一个步)的开始。STEP (08)无需执行条件，即其执行与否是由控制位来控制。为起动步的执行，需使用 SNXT (09)指令，其控制位和 STEP (08)指令一样。如果在“ON”执行条件下执行了 SNXT (09)指令，具有相同控制位的步也被执行。若

执行条件为“OFF”，则步就不执行。SNXT（09）指令必须写进程序，使得程序到达该步段前先执行 SNXT（09）指令。它可以用于步前的不同位置，从而按两种不同的条件来控制该步（见例 2）。程序中任何步若没有以 SNXT（09）指令开始，均不被执行。

一旦程序中使用了 SNXT（09），步执行将会持续，直到执行一条无控制位的 STEP（08）指令为止。但必须有一条具有虚控制位的 SNXT（09）指令作为无控制位的 STEP（08）指令的先导。虚控制位可以是任何未使用的 IR 或 HR 位，它不能是用于 STEP（08）中的控制位。

一个步动作可以通过下一个 SNXT（09）的执行或者关闭步中的控制位（见下面例 3）来完成。当步完成时，步中所有的 IR 和 HR 位都变 OFF，步中所有定时器（除 TTIM(－)）都被复位成它们的 SV 值。TTIM（－）、计数器、移位寄存器，用置位 SET 及复位 RSET 指令控制的位和用于保持 KEEP（11）中的位维持原状态。

两个简单的步过程显示如下：



地址	指令	操作数
00000	LD	00000
00001	SNXT(09)	LR 2000
00002	STEP(08)	LR 2000
由 LR2000 控制的步		
00100	LD	00001
00101	SNXT(09)	LR 2001

地址	指令	操作数
00102	STEP(08)	LR 2001
由 LR2001 控制的步		
00200	LD	00002
00201	SNXT(09)	LR 2002
00202	STEP(08)	——

步可以连续编程。每一步必须以 STEP（08）开始，而通常以 SNXT（09）结束（见下例 3 一个例外）。步指令以串行方式编程时，可能有 3 种执行类型：顺序、分支或并行。SNXT（09）的执行条件和位置决定了其类型，下面的 3 个例子说明这三种类型步执行情况。

注意事项

联锁、跳转、SBN（92）和 END（01）指令不能用于步程序中。

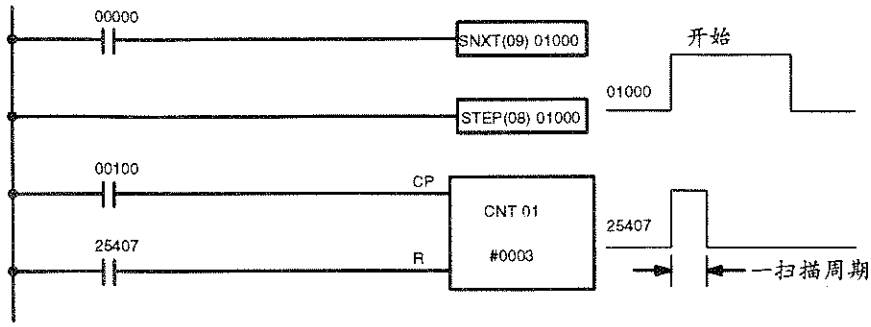
用于控制位的位不能用于程序中的其它任何地方，除非它们正用于控制步的运行（见下例 3）。所有的

控制位必须在相同的字里，且要按顺序使用。

如果 IR 或 LR 位用作控制位，当掉电时，它们的状态不会保存。若须保持这些状态，之后继续执行相同步，必须使用 HR 位。

标志

25407：步开始标志；当 STEP（08）执行时，在一扫描周期中，置 25407 为 ON，需要时，可用来复位计数器。（如下图）



地址	指令	操作数
00000	LD	00000
00001	SNXT(09)	01000
00002	STEP(08)	01000
00003	LD	00100

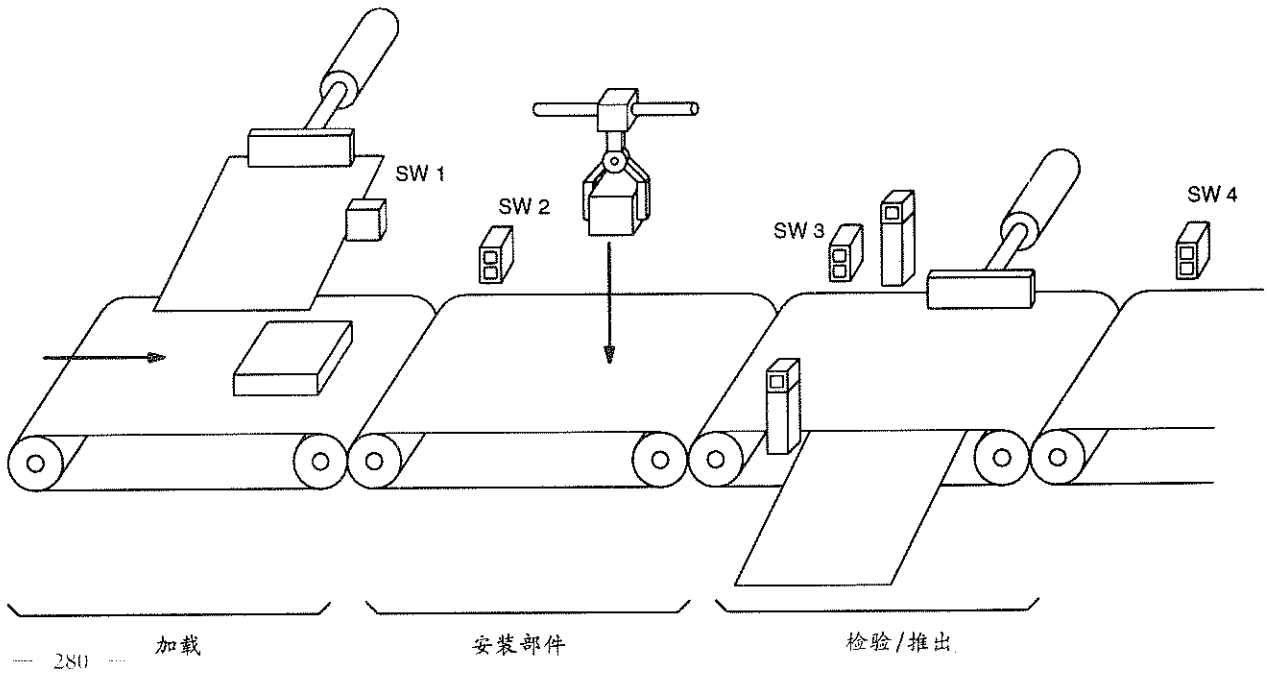
地址	指令	操作数
00004	LD	25407
00005	CNT	01
		# 003

举例

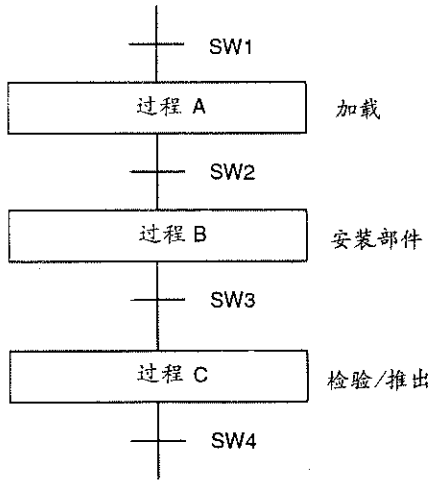
下例 3 个例子说明了步程序中三种不同执行控制的类型：例 1 是顺序执行；例 2 是分支执行；例 3 是并行执行。

例 1　顺序执行

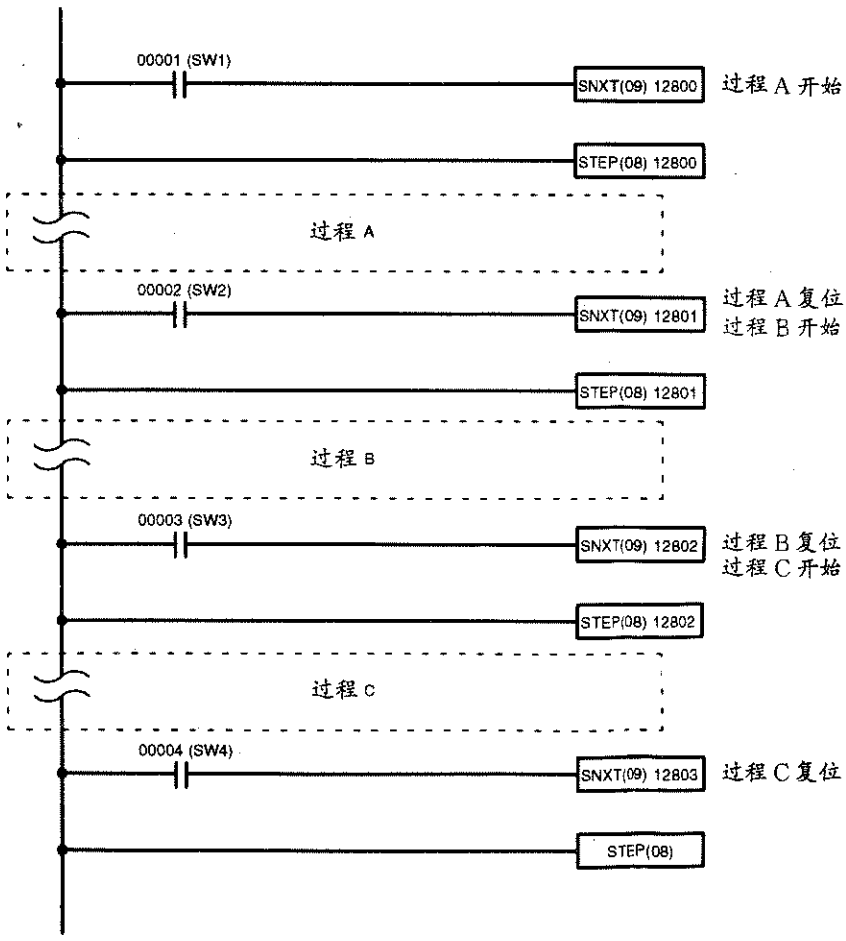
下列过程要求 3 个步骤：加载、安装部件和检验/推出，3 个步骤按顺序执行，在开始下一个过程前上一个过程被复位，当过程开始或结束时设置在不同位置的一组传感器（SW1、SW2、SW3 和 SW4）发出信号。



下图表示流程和用于控制执行的开关。



该控制过程的程序如下，它使用步编程的最基本类型：每一步用唯一的 SNXT(09)来完成并开始下一步。只有当表示前一步完成的开关接通后，下一步才开始。



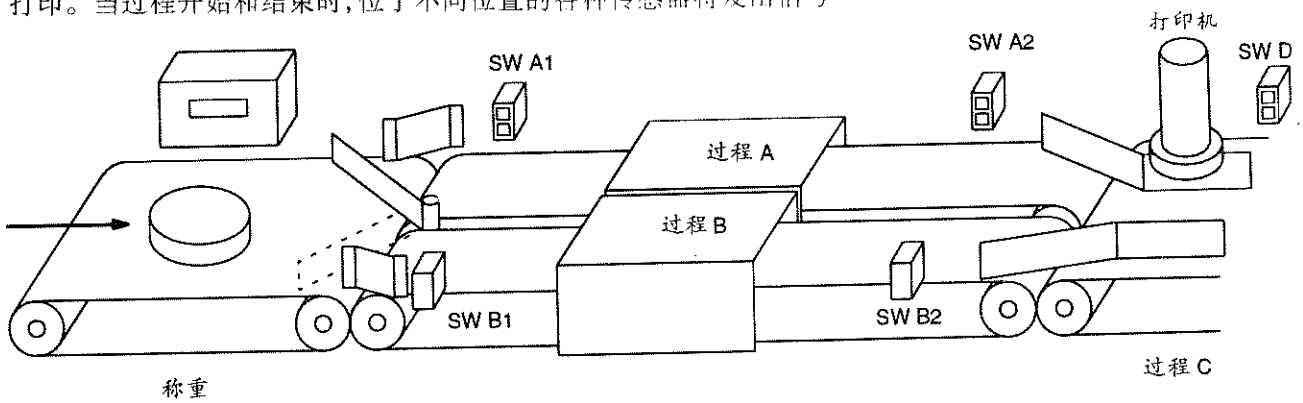
地址	指令	操作数
00000	LD	00001
00001	SNXT(09)	12800
00002	STEP(08)	12800
过程 A		
00100	LD	00002
00101	SNXT(09)	12801
00102	STPE(08)	12801

地址	指令	操作数
过程 B		
00100	LD	00003
00101	SNXT(09)	12802
00102	STEP(08)	12802
过程 C		
00200	LD	00004
00201	SNXT(09)	12803
00202	STEP(08)	——

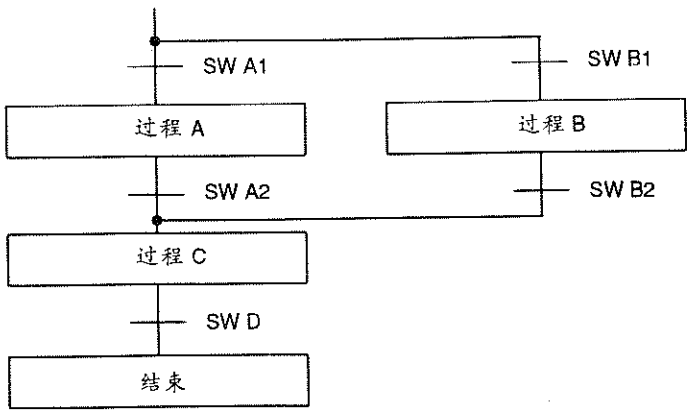
例 2

分支执行

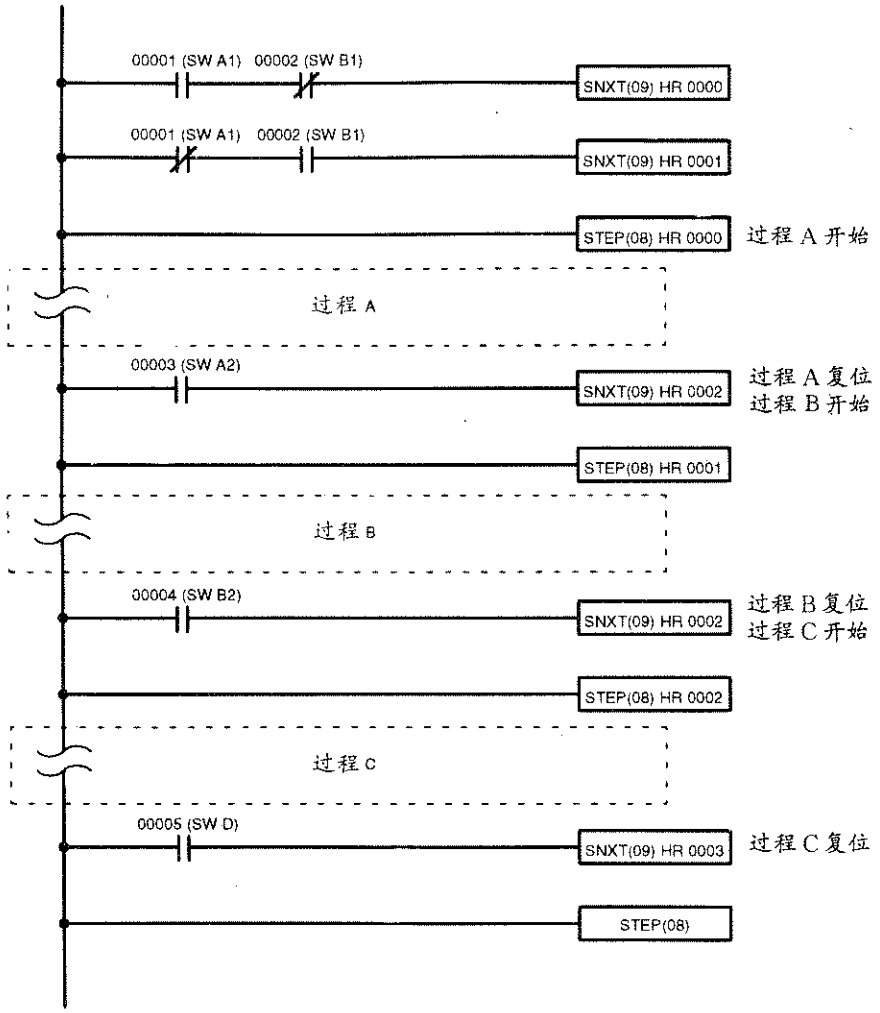
下面的过程要求某种产品在打印之前按照其重量在二种处理方式中选一种。但无论作那种处理，均需打印。当过程开始和结束时，位于不同位置的各种传感器将发出信号



下图表示执行控制的流程及用于执行控制的开关。这里用开关 SWA1 和 SWB1 的状态决定过程 A 或过程 B。



下图是该控制过程的程序，用两条 SNXT（09）指令开始过程 A 或 B。由于 00001(SWA1)和 00002(SWB1）被编入程序，启动过程 A 或 B，只要其中一个被执行。A 或 B 两个过程用同一条 SNXT（09）指令来结束，并开始过程 C。

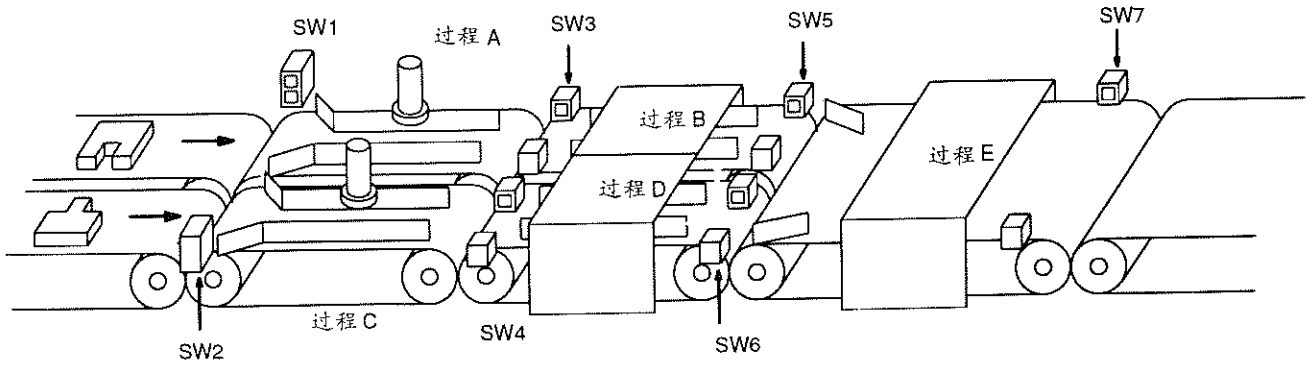


地址	指令	操作数
00000	LD	00001
00001	AND NOT	00002
00002	SNXT(09)	HR 0000
00003	LD NOT	00001
00004	AND	00002
00005	SNXT(09)	HR 0001
00006	STEP(08)	HR 0000
过程 A		
00100	LD	00003
00101	SNXT(09)	HR 00002
00102	STEP(08)	HR 0001

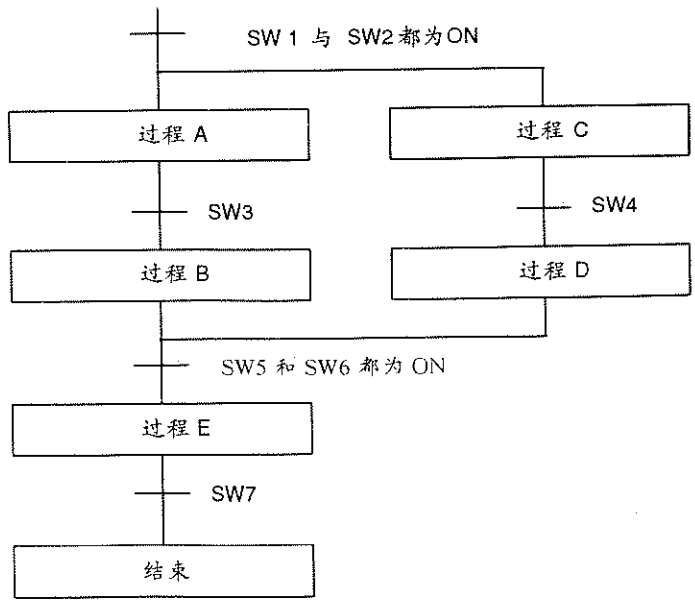
地址	指令	操作数
过程 B		
00100	LD	00004
00101	SNXT(09)	HR 0002
00102	STEP(08)	HR 0002
过程 C		
00200	LD	00005
00201	SNXT(09)	HR 0003
00202	STEP(08)	……

例 3　并行执行

下面的过程要求某种产品的两个另件同时进行两种不同加工处理，然后在第五道工序中会在一起。当过程开始或结束时，一组位于不同位置的传感器将发出信号。

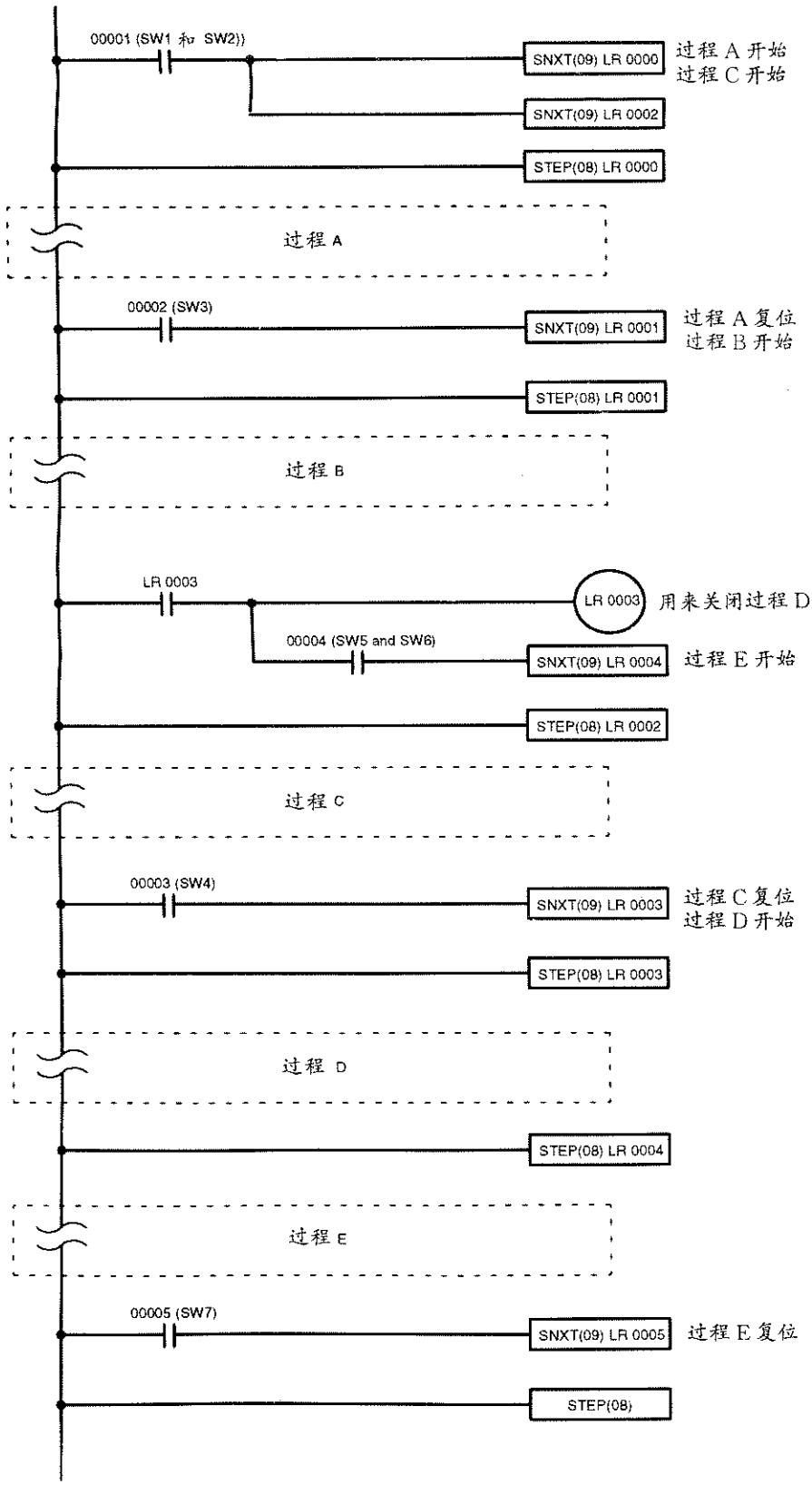


下图表示上面控制过程的流程及用作控制的开关。这里过程 A 和过程 C 一起开始，过程 A 结束时，过程 B 开始；过程 C 结束时，过程 D 开始。当过程 B 和过程 D 都结束时，过程 E 开始。



上述生产过程的程序如下图所示，程序从启动过程 A 和过程 C 的两条 SNXT(09)指令开始，它们从同一指令行上分开，同时执行开始过程 A 和过程 C 的步进。当过程 A 和过程 C 都结束时，过程 B 和过程 D 立即开始步进。

当过程 B 和过程 D 都结束时(即 SW5 和 SW6 都变为 ON 时)，在过程 B 的程序结束处，过程 B 和 D 一起被 SNXT(09)指令复位。虽然在过程 D 结束时没有 SNXT(09)指令，但它的控制位通过执行 SNXT(09) LR0004 指令而关闭，这是因为在该步中 LR0003 的输出是由 SNXT(09)LR0004 指令进行复位的。即当执行 SNXT (09) LR0004 指令时，LR0003 被关闭，这样过程 B 直接复位，而过程 D 在过程 E 步执行前被间接复位。


00001 (SW1 和 SW2))
SNXT(09) LR 0000
过程 A 开始
过程 C 开始
SNXT(09) LR 0002
STEP(08) LR 0000
过程 A
00002 (SW3)
SNXT(09) LR 0001
过程 A 复位
过程 B 开始
STEP(08) LR 0001
过程 B
LR 0003
LR 0003
用来关闭过程 D
00004 (SW5 and SW6)
SNXT(09) LR 0004
过程 E 开始
STEP(08) LR 0002
过程 C
00003 (SW4)
SNXT(09) LR 0003
过程 C 复位
过程 D 开始
STEP(08) LR 0003
过程 D
STEP(08) LR 0004
过程 E
00005 (SW7)
SNXT(09) LR 0005
过程 E 复位
STEP(08)

地址	指令	操作数
00000	LD	00001
00001	SNXT(09)	LR 0000
00002	SNXT(09)	LR 0002
00003	STEP(08)	LR 0000
过程 A		
00100	LD	00002
00101	SNXT(09)	LR 0001
00102	STEP(08)	LR 0001
过程 B		
00200	LD	LR 0003
00201	OUT	LR 0003
00202	AND	00004
00203	SNXT(09)	LR 0004

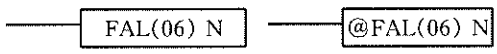
地址	指令	操作数
00204	STEP(08)	LR 0002
过程 C		
00300	LD	00003
00301	SNXT(09)	LR 0003
00302	STEP(08)	LR 0003
过程 D		
00400	STEP(08)	LR 0004
过程 E		
00500	LD	00005
00501	SNXT(09)	LR 0005
00502	STEP(08)	——

5－25 特殊指令

本节介绍的指令可用来进行各种操作，包括程序用户错误代码及信息、累计“ON”位数、设置监视定时器和在程序执行期间刷新 I/O。

5－25－1 出错报警指令－FAL(06)和严重故障报警指令——FALS(07)

梯形符号





定义的数据区

N:FAL 号
#(00～99)

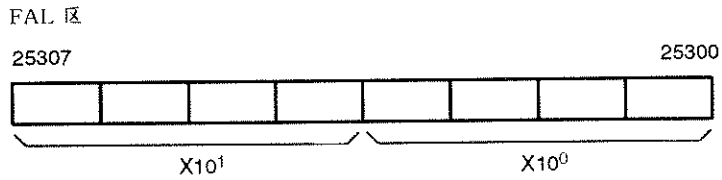
N:FAL 号
#(01～99)

限制

FAL(06)指令和 FALS(07)指令共享同一个 FAL 代码。记住 FAL(06)或 FALS(07)可分别使用同一个代码，但不能两者一起用。

说明

FAL(06)和 FALS(07)指令能为编程员在运行、维修和调试程序时提供错误的代码。在"ON"执行条件下执行时，两者任一个指令将在 SR253 的 00 位～07 位输出一个 FAL 代码。输出的代码在 01 与 99 之间，并作为定义符被输入到 FAL(06)或 FALS(07)的数据区内。带有 00 定义符的 FAL(06)指令是用来复位该区的。(见下图)



FAL(06)产生一个非致命的错误，而 FALS(07)则产生致命的错误。当 FAL(06)在 ON 执行条件执行时，CPU 面板上的 ALARM/ERROR 指示灯会闪烁，但 PC 将继续执行程序。当 FALS(07)指令在"ON"执行条件下执行，ALARM/ERROR 指示灯点亮，PC 停止执行程序。

系统产生一个错误代码放入 FAL 区。

清除错误

虽然在 FAL 区只可得到一个错误代码，但存贮器内可存所有 FAL 错误代码。若想获取其它代码，可执行 FAL(06)00 指令来复位 FAL 区。每执行 FAL(06)00 指令一次，另一个 FAL 错误代码则会进入 FAL 区，同时清除上一个代码。FAL 错误代码以下列次序记录和调出：产生的第 1 个代码；比第 1 个代码大的最小 FAL 号；比第 1 个代码小的最小 FAL 号。

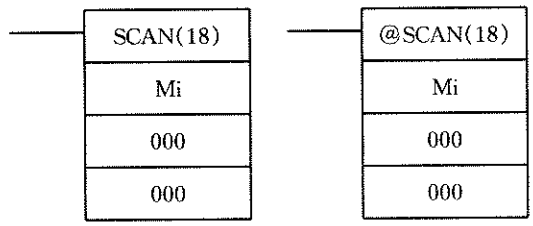
清除信息

FAL(06)00 指令与 MSG(46)指令一起使用还可用来清除信息。

如果 FAL 不能清零(一般执行 FALS(07)指令时出现该种情况)，先要清除出错原因，再通过编程器的操作面板清除 FAL 区。(见 4－6－5"清除错误信息")

5－25－2 循环时间指令－SCAN(18)

梯形符号



操作数区

Mi：乘数(BCD)
IR, SR, AR, DM, HR, TC, LR, #

000：不用

000：不用

限制

Mi 必须是 BCD 码，只用 Mi 的最右面 3 位数。

说明

SCAN(18)指令用来设定最小扫描周期，Mi 是最小扫描周期，以十分之一毫秒为单位。例如 Mi＝1200，最小扫描周期是 120.0ms，Mi 的设置范围是 000.0 到 999.0ms。

如果实际扫描周期小于 SCAN(18)指令设定的周期，CPU 将等待预置的周期到了之后再执行下一个周期。如果实际扫描周期大于设定周期，该设定周期无效，程序将执行到结束为止。

标志

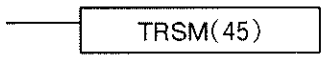
ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或者超出 DM 区域）

Mi 不是 BCD 码。

5－25－3 跟踪存贮器采样指令－TRSM(45)

数据跟踪用来便于诊断程序，要建立并使用数据跟踪需有一台上位机运行 SSS(中文版)程序。在编程器上不能进行数据跟踪，SSS(中文版)操作手册中详细介绍了数据跟踪。本节介绍 TRSM(45)指令的梯形符号并给出一个程序例子。

梯形符号



说明

TRSM(45)指令在程序中用来标明那些存于跟踪内存中的指定数据地址。可指定最多跟踪 12 个位和 3 个字(参见 SSS(中文版)操作手册)。

TRSM(45)指令不受执行条件控制，而是通过 AR 区中 AR2515 和 AR2514 两位来控制。AR2515 为采样开始位，该位“ON”时开始跟踪采样过程。采样开始位不是由程序来置 ON 的，而是由外围设备控制置 ON 的。AR2514 位跟踪起始位，当该位置“ON”时，指定的数据在跟踪存贮器中记录下来，跟踪起始位可由程序置“ON”，也可以由编程设备置“ON”，也可以设定一个正或负的延时，以改变跟踪开始的实际起始点。

数据可用三种方法之一记录下来。TRSM(45)指令可放在程序中的一处或多处，从而指定该处特定的数据将被跟踪。如果不用 TRSM(45)指令，则当执行 END(01)时，指定数据也被跟踪。第三种方法是由外设设定一个时间间隔，使数据以一个固定时间间隔记录下来，而与周期无关(参见 SSS(中文版)操作手册)。

TRSM(45)指令可放在程序的任何地方，使用任意次，跟踪存贮器的数据可以用编程器或上位机监视。

AR 控制位和标志

下面的控制位及标志在数据跟踪中使用。跟踪工作期间跟踪标志位为 ON。当足够多的数据填满跟踪内存时，跟踪结束标志为 ON。

标志	功能
AR2515	采样开始位
AR2514	跟踪起始位
AR2513	跟踪标志
AR2512	跟踪结束标志

注意事项

如果跳转指令条件为 OFF 时出现了 TRSM(45)，在 JMP(08)—JME(09)程序块中 TRSM(45)指令不会被执行。

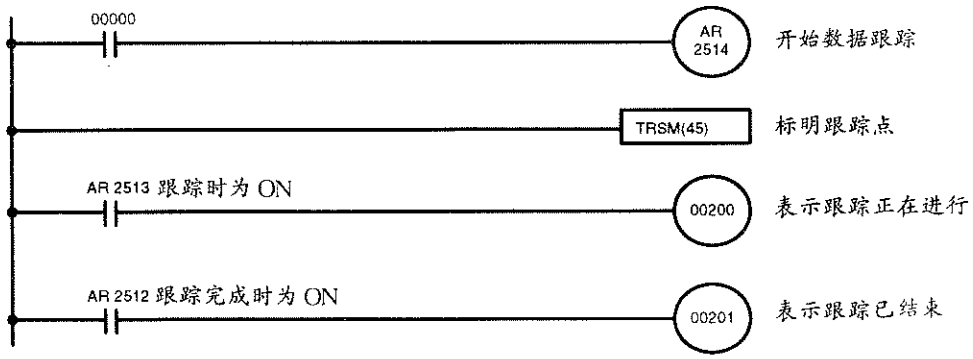
举例

下面例子说明数据跟踪的基本编程及操作。将采样开始位(AR2515)强迫置位，开始采样。采样开始位不能在程序中置为 ON。数据读出后存入跟踪存贮器中。

当 IR00000 为 ON，跟踪开始位(AR2514)也变为 ON，CPU 即查看延时时间并相应地在跟踪存贮器中打标记。这意味着已经采样到几个将存入跟踪内存(负延时)的数据或者在数据存入内存之前还需采几个数据(正延时)。

采样到的数据写入跟踪存贮器中，一旦存到最后的单元即跳到内存的开始单元，继续存入数据直到开始

标记。这说明先前记录的数据(即在内存标志之前的数据)会被覆盖(若正延时时,特别有可能产生)。负延时不会这样,因所需的数据在采样开始前已得到了。

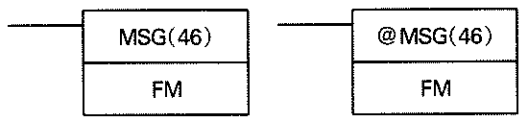


地址	指令	操作数
00000	LD	00000
00001	OUT	AR 2514
00002	TRSM(45)	
00003	LD	AR 2513

地址	指令	操作数
00004	OUT	00200
00005	LD	AR 2512
00006	OUT	00201

5－25－4 信息显示指令－MSG(46)

梯形符号



操作数数据区域

FM:信息起始字
IR, SR, AR, DM, HR, LR

限制

FM 和 FM＋7 必须在同一个数据区内。

说明

当 MSG(46)指令在 ON 执行条件下执行时,它从 FM～FM＋7 读取 8 个扩展 ASCII 码字,并在编程器上显示该信息,显示的信息最多达 16 个字符:即每个 ASCII 字符码需 8 位(二位数)。参阅附录 G“扩展 ASCII 码”。日本的片假名字符也包括在其中。

如果信息不足 8 个字节,可在任何要结束的地方输入“0D”。当信息中遇到“0D”时,它之后的数据不再读取,信息中的字可用于其它用途。

信算缓冲及优先权

最多有 3 组信息存于存贮器缓冲器中,一旦存入缓冲器,其显示的规则是“先进先出”。由于在一个扫描周期内可能有 3 个以上的 MSG(46)指令要执行,这些信息按存贮不同的区域而有一个优先权,以选择优先权高的信息进入缓冲区。

存放显示内容的存贮器优先权如下:

LR＞IR＞HR＞AR＞TC＞DM＞SR

对同一内存区的信息,以地址最低的内容优先权最高。

对间接寻址的信息(即＊DM),具有最低 DM 地址值的优先权最高。

清除信息

要清除显示的内容，可执行 FAL(06)00 指令，或用 4－6－5 节“清除错误信息”中步骤，通过编程器操作清除。

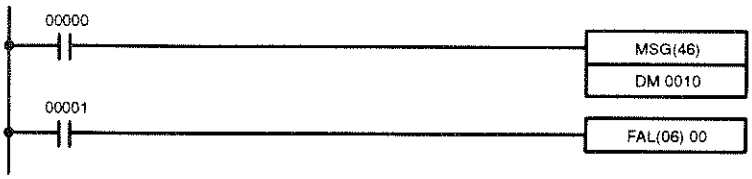
若信息正在显示时数据发生变化，显示也会改变。

标志

ER：间接寻址的 DM 字不存在。（＊DM 字内容不是 BCD 码或超出 DM 区域）

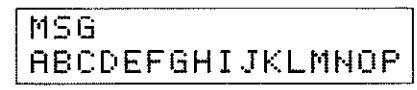
举例

下面的例子表示当 00000 是 ON 时，指令及数据将显示出来；如果 00001 变 ON 时，显示会被清除。



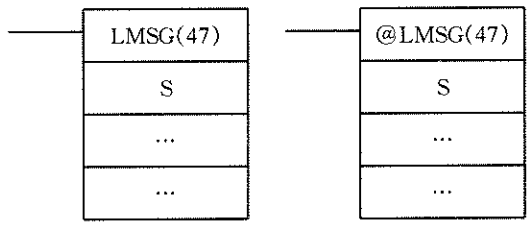
地址	指令	操作数
00000	LD	00000
00001	MSG(46)	
		DM 0010
00002	LD	00001
00003	FAL(06)	00

DM 内容					ASCII 对应字符	
DM 0010	4	1	4	2	A	B
DM 0011	4	3	4	4	C	D
DM 0012	4	5	4	6	E	F
DM 0013	4	7	4	8	G	H
DM 0014	4	9	4	A	I	J
DM 0015	4	B	4	C	K	L
DM 0016	4	D	4	E	M	N
DM 0017	4	F	5	0	O	P



5－25－5 长信息指令－LMSG(47)

梯形符号



操作数区

S：源起始字(ASCII)
IR，SR，AR，DM，HR，TC，LR

…：不用
设为 000

…：不用
设为 000

限制

S～S＋15 必须在同一数据区，且均为 ASCII 码。如果在 S 和 S＋15 间插有空字符(0D)，则信息将会缩短。

说明

LMSG(47)指令是用来将32个字符信息输出到编程器上，除了较短的信息以外，信息必须以ASCII码输出，且在S字开始，在S+15字结束。在字符串中的空字符(0D)能产生一个较短的信息，空字符不输出字符。

为输出到编程器，一定要设置成终端方式，虽然LMSG(47)会按普通方式执行，但信息不能正确地显示在编程器上，除非设置在终端方式。转换到终端方式的细节参阅5－25－6终端方式指令——TERM(48)。

如果CPU的DIP开头的第6脚是OFF，按CHG键或在程序中执行TERM(48)，编程器被转换到终端方式。如果CPU的DIP开头第6脚是ON，接通位AR0709，编程器被转换到扩展终端方式。

标志

ER：S和S+15不在同一数据区。

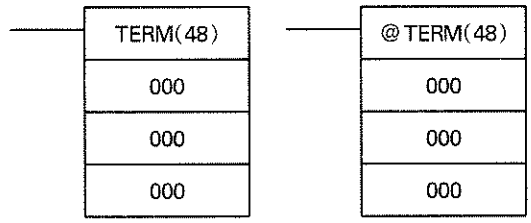
间接寻址的DM字不存在(＊DM字的内容不是BCD码，或者超出DM区域)

例子

虽然显示时间长，而且输出装置可选择，LMSG(47)的编码同于MSG(46)，参阅前面章节使用MSG(46)的例子。

5－25－6 终端方式指令－TERM(48)

梯形符号



说明

如果指令执行条件为OFF，不执行TERM(48)指令。如果执行条件为ON，TERM(48)指令把编程器转换到终端方式。(指令MSG(46)、LMSG/(47)、键盘唤象功能在终端方式执行。)

当再按CHG键编程器返回到编程器方式。在程序中没有指令能使编程器返回到编程器方式。

如CPU的DIP开关的第6脚是OFF，在输入口令字之前或显示方式时按下编程器上的CHG键也能使编程器转换到终端方式。当再按下CHG键，编程器会返回到编程器方式。

如果CPU的开关第6脚是ON，接通位AR0709，编程器被转换到扩展终端方式。

例子

如下例子，确信CPU的DIP开头第6脚是OFF，当00000是ON，用TERM(48)指令把编程器转换到终端方式。



地址	指令	操作数
00000	LD	00000
00001	TERM(48)	
		000
		000
		000

5－25－7 监视定时器刷新指令－WDT(94)

梯形符号



定义符数据区域

T:监视定时器值
(00～63)

说明

如果指令执行条件为 OFF,不执行 WDT(94),如果执行条件为 ON,WDT(94)以 100ms 的 T 倍扩展监视定时器设置时间(该时间 PC 设置在 DM6618),缺省设定为 120ms。

扩展的定时器时间 = 100ms × T

注意事项

扫描周期监视时间(监视定时器设定)可以在 10～640ms 范围内由 PC 设置设定。还可以使用 WDT(94)指令将扫描周期监视时间扩展到最大值 640ms,但只可扩展扫描周期中执行指令的部分时间。

一个扫描周期中可以多次执行 WDT(94)指令,但是总的执行时间不能超过 640ms,超出 640ms 任何延伸都无效。如果扫描周期时间已经超过 640ms,不再执行其他 WDT(94)指令。

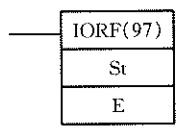
当扫描周期时间超过 100ms,定时器不能有真正的功能。如果使用 WDT(94)指令,在程序中用同一个定时器以小于 100ms 的时间间隔重复进行。TIMH(15)仅用在定时中断执行程序中,执行时间间隔为 10ms 或小于 10ms。

标志

这条指令对标志没有影响

5－25－8 I/O 刷新指令－IORF(97)

梯形符号



操作数数据区域

St:起始字
IR000～IR049, IR100～IR199, SR400～SR450

E:结束字
IR000～IR049, IR100～IR199, SR400～SR450

限制

IORF(97)仅用于刷新分配给安装在 CPU 机架和扩展 I/O 机架上 I/O 单元、特殊 I/O 单元和中断输入单元的 I/O 字。它不能用来刷新其它 I/O 字,例如:在从站机架上的 I/O 单元或者组－2 高密度 I/O 单元。

St 必须小于或等于 E。

说明

为了刷新分配给 CPU 或扩展 I/O 机架的 I/O 字(IR000～IR029 或 IR300～IR309),简单地指明要被刷新的起始(St)和结束(E)I/O 字。如果 IORF(97)的执行条件是 ON,在 St 和 E 之间的所有字都将被刷新。这将在 CPU 的扫描周期内附加在正常 I/O 刷新以外完成。

为了刷新分配给特殊 I/O 单元 0～9 的 I/O 字(IR100～IR199),指定 IR040～IR049,这些 I/O 字用来对应特殊 I/O 单元。执行 IORF(97)指令不会影响 IR040～IR049 中的内容。

例如,设 St 为 IR043,E 为 IR045,刷新分配给特殊 I/O 单元 3,4 和 5 的 I/O 字,当执行 IORF(97)指令时,刷新分配给这些单元的 I/O 字(IR130～IR159)。这将在 CPU 的扫描周期内附加在正常 I/O 刷新以外完成的。

要指明特定 10 字特殊 I/O 单元区域(IR100～IR190 或 IR400～IR450),输入 10 字区域的首字。(IR 字的最后一位数必定是 0)

关于刷新分配给组－2 高密度 I/O 单元的 I/O 字,详见 5－25－9"组－2 高密度 I/O 刷新指令—MPRF(61)"。

标志

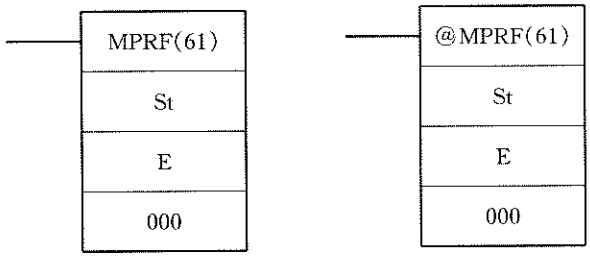
ER:St 和/或 E 不在适当的设定值范围内。(000～029,040～049,100～190,300～309 或 400～450)

St 和 E 不在同一设定范围内。

St 大于 E。

5－25－9 组－2 高密度 I/O 刷新指令—MPRF(61)

梯形符号



操作数数据区域

St:开始单元
0000～000F
E:结束单元
#0000～#000F
000:置 000
……

限制

MPRF(61)仅用于刷新分配给组－2 高密度 I/O 单元的 I/O 字,它不能用于其它 I/O 字。

St 和 E 必须在#0000～#000F 之间,St 必须小于或等于 E。

说明

如果指令执行条件是 OFF,不执行 MPRF(61)。如果执行条件是 ON,分配给组－2 高密度单元(I/O 单元号从 St 到 E)的 I/O 字被刷新。除正常 I/O 刷新之外,这将在 CPU 的扫描周期内完成。

不能用地址定义 I/O 字,只能用它们的单元号来规定。

执行时间:

MPRF(61)执行时间按下式计算:

T_{MPRF} = 指令执行时间 + Σ(组－2 高密度 I/O 单元 I/O 刷新时间)

参阅 6－1 节"扫描周期"中组－2 高密度 I/O 单元 I/O 刷新时间表。

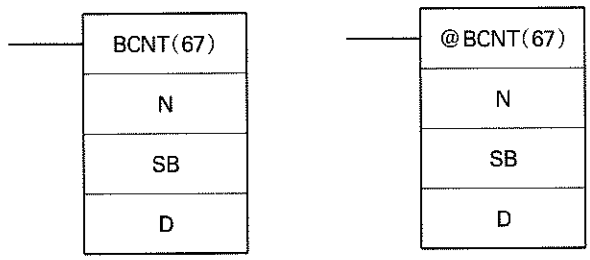
标志

ER:St 或 E 不是在#0000～#000F 之间的 BCD 码。

St 大于 E。

5－25－10 位计数器指令—BCNT(67)

梯形符号



操作数数据区域

N:字数(BCD)
IR, SR, AR, DM, HR, TC, LR, #
SB:源起始字
IR, SR, AR, DM, HR, TC, LR
D:目标字
IR, SR, AR, DM, HR, TC, LR

限制

N 必须是 0000～6656 之间的 BCD 码。

说明

如果指令执行条件为 OFF，不执行 BCNT(67)；如果执行条件为 ON，BCNT(67)对在 SB 和 SB＋(N－1)之间所有字中 ON 位的总数进行计数，并且把结果存入 D。

标志

ER：N 不是 BCD 码，或 N 是 0；SB 和 SB＋(N－1)不在同一区域。

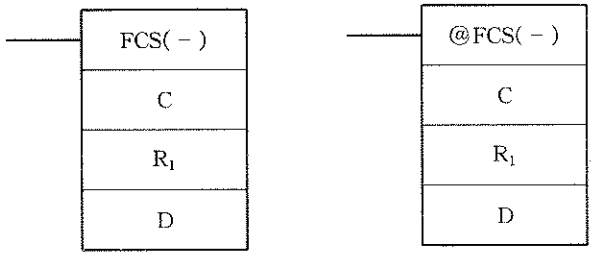
结果计数值超出 9999。

间接寻址的 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 的区域)

EQ：如果结果是 0 为 ON。

5－25－11 帧校检指令－FCS(—)

梯形符号



操作数数据区域

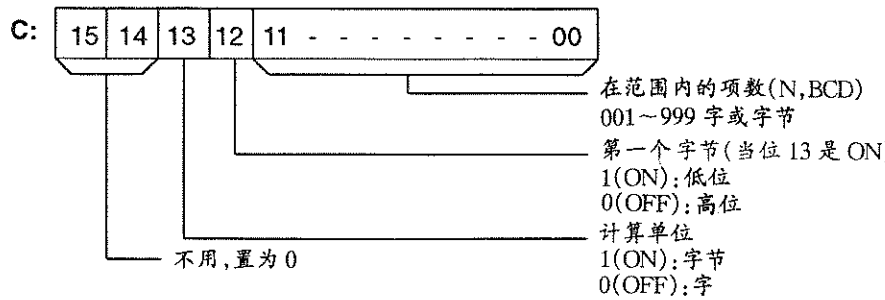
C：控制字
IR，SR，AR，DM，HR，LR，#
R_1：在范围内起始字
IR，SR，AR，DM，HR，TC，LR
D：目标起始字
IR，SR，AR，DM，HR，LR

说明

通过通信端口传送数据时，FCS(—)可用来检查出错情况。

如果指令执行条件是 OFF，不执行 FCS(—)，如果执行条件是 ON，FCS(—)按异或计算规定范围内的帧校检和，范围在 R1～R1＋N－1 字的内容或 R1～R1＋N－1 字的字节，然后将帧校验和(十六进制数)转换成 ASCII 码并输出至目标字(D 和 D＋1)。

C 的位功能如下图所示，而且后面会有更详细的解释。



范围内的项数

在范围内的项数(N)包括在 C 的最右面的 3 位数内，必须是在 001～999 之间的 BCD 码。

计算单位

如果位 13 是 OFF，计算字的帧校验和；如果位 13 是 ON，计算字节的帧校验和。

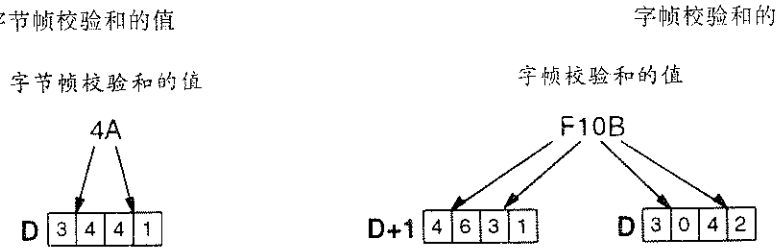
如果指定为字节计算，范围能由 R1 的高位或低位字节开始算起。如果位 12 是 ON，R1 的高位字节不包括在内。

	高位	低位
R1	1	2
R1 + 1	3	4
R1 + 2	5	6
R1 + 3	7	8
·	·	·
·	·	·
·	·	·

当位 12 是 OFF，字节按：1，2，3，4，…的次序进行或运算；当位 12 是 ON，字节按：2，3，4，5，…的次序进行或运算。

转换成 ASCII 码

字节帧校验和计算产生一个 2 位十六进制数，再转换成与它等效的 4 数位 ASCII 码。字帧校验和计算产生一个 4 位十六进制数，并且转换成与它等效的 8 数位 ASCII 码。如下所示：



标志

ER：间接寻址 DM 字不存在。（＊DM 字的内容不是 BCD 码或者超出 DM 区域）

项数不是 001～999 之间的 BCD 码。

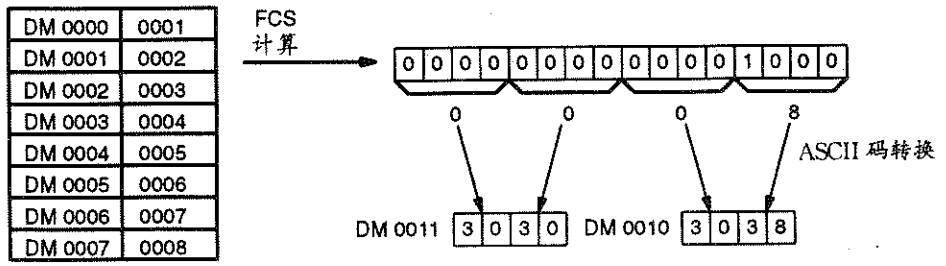
计算范围超出数据区域。

例子

如下例中 IR00000 为 ON，对从 DM0000～DM0007 的 8 个字计算出帧检验和(0008)，相应的 ASCII 码为(30 30 30 38)写到 DM0011 和 DM0010 中。

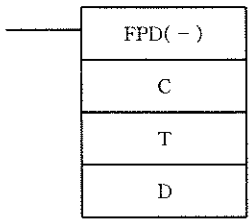


地址	指令	操作数
00000	LD	00000
00001	@FCS(－)	
		# 0008
		DM 0000
		DM 0010



5－25－12 故障点检测指令－FPD(－)

梯形符号



操作数数据区域

C:控制数据
#

T:监控时间(BCD)
IR, SR, AR, DM, HR, TC, LR, #

D:起始寄存器字
IR, AR, DM, HR, LR

限制

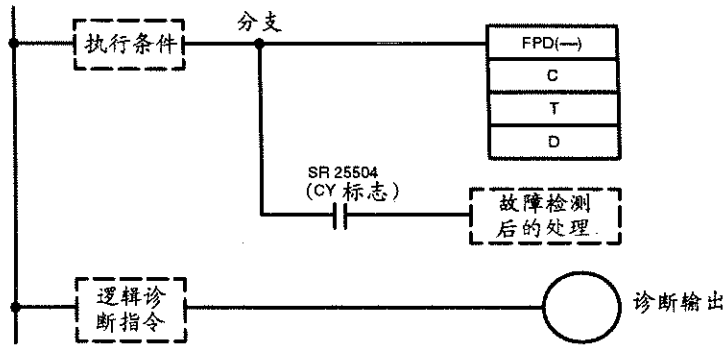
当 C 的位 15 是 ON, D 和 D＋8 必须在同一数据区域。

C 必须作为常数输入。

说明

根据需要, FPD(—)在程序中可多次使用, 但每次必须使用不同的 D。它用于监视在 FPD(—)执行和诊断输出执行之间的时间, 如果这时间超出 T, 产生一个 FAL(06)非致命错误, 并且带有由 C 指定的 FAL 数。

在下图中由虚线标记的程序段可根据特定的程序应用需要来编写指令。由进位位(CY)触发的处理程序段是任选的, 可用除 LD 和 LD NOT 的任何指令。逻辑诊断指令和执行条件可由包括所要求的常闭或常开条件任意组合。

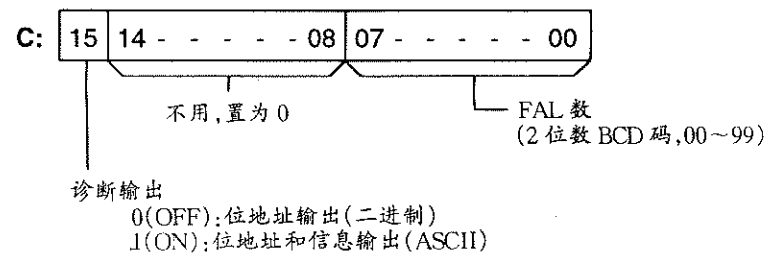


当执行条件是 OFF, 不执行 FPD(—); 当执行条件是 ON, FPD(—)监视时间直到逻辑诊断条件变成 ON, 并且把诊断输出转变成 ON。如果这时间超出 T, 将出现下列情况。

1. 产生一个由 C 中的开头二位数指定 FAL 数的 FAL(06)错误。如果指定为 00, 不会产生错误。
2. 逻辑诊断指令搜索第一个 OFF 输入条件, 而且这条件的位地址输出到 D 开始的目标字。
3. CY 标志(SR25504)变成 ON。如果需要, 使用这 CY 标志去执行错误处理程序段。
4. 如果 C 的位 15 是 ON, 最多 8 位 ASCII 字符的预置信息和在步骤 2 提到的位地址一起显示在外围设备上。

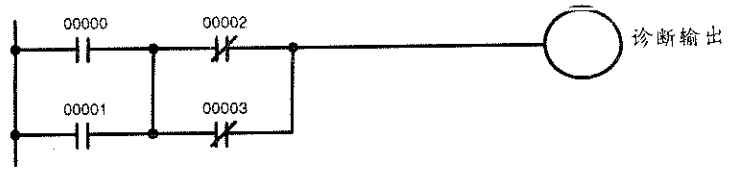
控制数据

C 的控制数据位功能如下图所示



逻辑诊断指令

如果逻辑诊断条件变成 ON 的时间超出 T，搜索逻辑诊断指令中 OFF 输入条件。若有一个以上输入条件是 OFF，则选择最高指令行和最靠近左母线的输入条件。



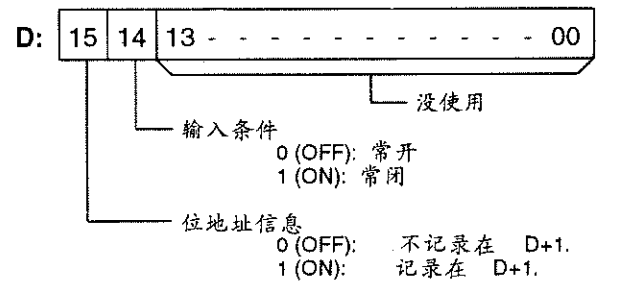
如果 IR00000～IR00003 都是 ON，寻找出常闭条件 IR00002 诊断输出没有变为 ON 的起因。

诊断输出

在逻辑诊断条件中，有二种方法输出检测 OFF 条件的位地址：

1. 位地址输出(当 C 的位 15 是 OFF 时使用)

D 的位 15 指示是否有位地址信息存储在 D+1。如果有的话，D 的位 14 指示输入条件是常开还是常闭的。



D+1 含有输入条件的位地址码，如下所示，字地址，位号和 TC 号都用二进制编码。

数据区域	D+1 位状态															
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IR, SR(见注 C)	1	0	0	0	字地址								位号			
	1	0	1	0	字地址								位号			
HR	1	0	0	1	1	字地址							位号			
LR	1	0	0	1	0	0	字地址						位号			
TC*	1	0	0	1	0	1	*	定时器或计数器号								

注

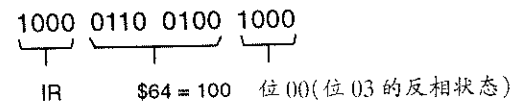
a) * 对于 TC 区域，D+1 的 09 位指示这数字是定时器号还是计数器号，"0"表示定时器，"1"表示计数器。

b)位数最左位(位 03)的状态是反相的。

c)位 13 为 OFF 表示 IR00000～SR25515，即位 13 为 ON 表示 SR25600～IR51115。

例子

如果 D+1 内容是 1000 0110 0100 1000，IR10000 会按下列所示。



2. 位地址和信息输出(当 C 的位 15 是 ON 时使用)。

D 的位 15 指示在 D+1～D+3 内是否有位地址信息存储。如果有信息的话，D 的位 14 指示输入条件是常开的还是常闭的。参阅下表。

如果执行 FPD(一)，用 ASCII 码将字 D+5～D+8 包含的信息和位地址一起显示在外围设备上。

D+5～D+8 由用户预置，其中包含的信息如下表所示：

字	位 15～08	位 07～00
D+1	20＝空	位地址的第一个 ASCII 字符
D+2	位地址的第二个 ASCII 字符	位地址的第三个 ASCII 字符
D+3	位地址的第四个 ASCII 字符	位地址的第五个 ASCII 字符
D+4	2D＝“－”	“0”＝常开 “1”＝常闭
D+5	信息的第一个 ASCII 字符	信息的第二个 ASCII 字符
D+6	信息的第三个 ASCII 字符	信息的第四个 ASCII 字符
D+7	信息的第五个 ASCII 字符	信息的第六个 ASCII 字符
D+8	信息的第七个 ASCII 字符	信息的第八个 ASCII 字符

注

如果在信息中不需要 8 个字符，可在最后字符后输入“0D”。

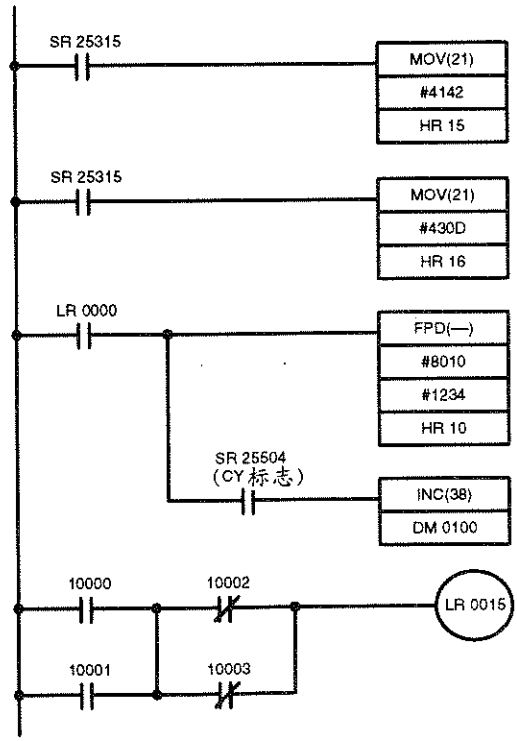
决定监控时间

在实际操作条件下，如果对 T 指定一个字操作数，按下面的步骤自动地设置监控时间 T；如果 T 设置为一常数，就不能使用这样的操作。

1. 把 PC 转换到监视方式操作。
2. 连接外围设备，如编程器。
3. 利用外围设备将控制位 AR2508 置为 ON。
4. 利用转换成 ON 的 AR2508 执行程序，如果 T 中的当前监控时间已超出的话，1.5 倍的实际监控时间就会存储到 T 中。在 AR2508 为 ON 期间内不出现 FAL(06)错误。
5. 当一个可接受值存储在 T 中后，AR2508 变成 OFF。

例

在下面例子中，当监控时间超出 123.4S，这 FPD(一)被设置来显示位地址和信息(“ABC”)。



地址	指令	操作数
00000	LD	25315
00001	MOV(21)	
		# 4142
		HR 15
00002	LD	25315
00003	MOV(21)	
		# 430D
		HR 16
00004	LD	LR 0000
00005	FPD(—)	
		# 8010
		# 1234
		HR 10
00006	AND	25504
00007	INC(38)	
		DM 0100
00008	LD	10000
00009	OR	10001
00010	LD NOT	10002
00011	OR NOT	10003
00012	AND LD	
00013	OUT	LR 0015

当 LR0000 成 ON 时，执行 FPD(一)，并且开始进行监控。如果在 123.4 秒范围内 LR0015 没有转换成 ON，而且 IR10000～IR10003 全是 ON，选择 IR10002 作为出错的起因，产生一个 FAL 号为 10 的 FAL(06) 错误，并且把位地址和预置信息("10002－1ABC")显示在外围设备上。

HR	10	0000		HR	10	C000	指示信息，常闭条件
HR	11	0000		HR	11	2031	"1"
HR	12	0000		HR	12	3030	"00"
HR	13	0000		HR	13	3032	"02"
HR	14	0000		HR	14	2D31	"－1"
HR	15	4142	→	HR	15	4142	"AB"
HR	16	430D		HR	16	430D	"C"，和 CR 编码
HR	17	0000		HR	17	0000	最后两个字被忽略
HR	18	0000		HR	18	0000	(作为空白显示)

标志

ER：T 不是 BCD 码。

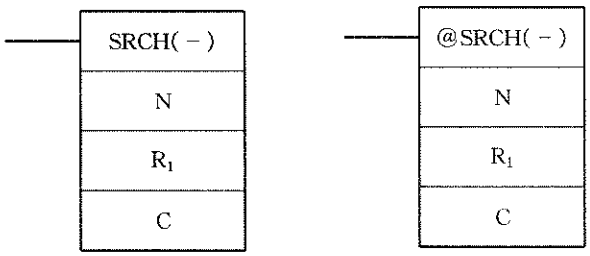
C 不是一常数或 C 的最右二位数不是 00～99 的 BCD 码。

间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 区域)

CY：当 FPD(一)的执行和诊断输出的执行之间的时间超出 T 时为 ON。

5－25－13 数据搜索指令－SRCH(－)

梯形符号



操作数数据区域

N：字数
IR，SR，AR，DM，HR，TC，LR，#
R_1：在范围内第一个字
IR，SR，AR，DM，HR，TC，LR
C：比较数据，结果字
IR，SR，AR，DM，HR，LR

限制

N 必须是 0001～6656 之间的 BCD 码。

R1 和 R1＋N－1 必须在同一数据区。

说明

当执行条件是 OFF，不执行 SRCH(－)；当执行条件是 ON，SRCH(－)搜索从 R1～R1＋N－1 存储范围内的地址，在这地址里包含有 C 中比较数据。如果一个或一个以上的地址包含有比较数据，EQ 标志(SR25506)就变为 ON，且包含比较数据的最低地址在 C＋1 中指明。对 DM 区而言，不同的地址确定在于：

1. DM 区域内的地址：字地址写入到 C＋1。例如，如果包含着比较数据的最低位地址是 DM0114，那末＃0114 写入到 C＋1。
2. 其它数据区域内地址：将搜索开始的地址个数写入到 C＋1。例如，如果包含着比较数据的最低位地址是 IR114，在查找范围内第一个字是 IR014，那末，把＃0100 写入 C＋1。

如果在这范围里没有包含比较数据的地址，EQ 标志(SR25506)变为 OFF，而 C＋1 的内容不变。

标志

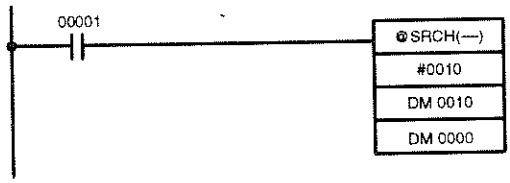
ER： 间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码，或者超出 DM 区域)

N 不是 0001～6655 之间的 BCD 码。

EQ： 当在搜索区域里找到比较数据相同的数据时，EQ 置 ON。

例

下例中，从 DM0010 到 DM0019 这 10 个字范围内查找含有与 DM0000(＃FFFF)相同数据的地址。由于 DM0012 包含同样数据，EQ 标志(SR25506)变为 ON，并且＃0012 写入到 DM0001。



地址	指令	操作数
00000	LD	0001
00001	@SRCH(－)	
		＃ 0010
		DM 0010
		DM 0000

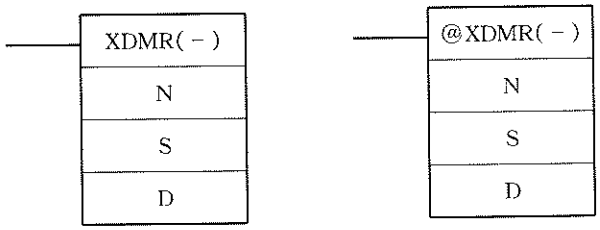
DM0010	0000
DM0011	9898
DM0012	FFFF
DM0013	9797
DM0014	AAAA
DM0015	9595
DM0016	1414
DM0017	0000
DM0018	0000
DM0019	FFFF

→

DM0000	FFFF
DM0001	0012

5－25－14 扩展 DM 读指令－XDMR(－)

梯形符号



操作数数据区域

N：字数
IR, SR, AR, DM, HR, TC, LR, ＃
S：第一个扩展 DM 字
IR, SR, AR, DM, HR, TC, LR, ＃
D：第一个目标字
IR, SR, AR, DM, HR, LR

限制

N 必须是 0001～3000 之间的 BCD 码。

S 必须是 7000～9999 之间的 BCD 码。

S 和 S＋N－1 必须在同一数据区，D 和 D＋N－1 也必须在同一数据区。

说明

如果执行条件是 OFF，不执行 XDMR(－)；如果执行条件是 ON，XDMR(－)将扩展 DM 字 S～S＋N－1 之间的内容拷贝至目标字 D～D＋N－1。

注意事项

程序使用该指令前，必须先在 PC 设置中设置好扩展 DM 区。不要超出扩展 DM 区域的设置范围。

只要出现电源中断，优先执行 XDMR(－)。

标志

ER： 指定的扩展 DM 字不存在。确信指定的字已分配给扩展 DM。详细参阅 7－2－15“UM 区域分配”。

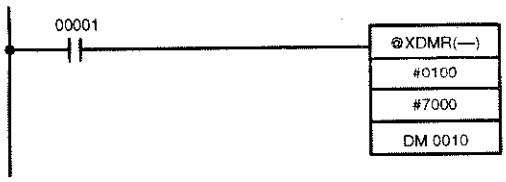
间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 区域)

N 不是 0001～3000 之间的 BCD 码。

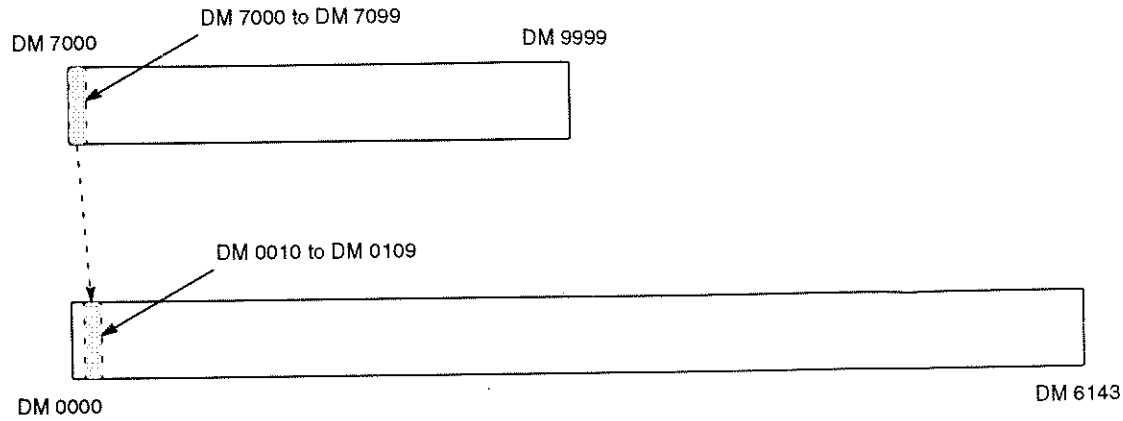
S 不是 7000～9999 之间的 BCD 码。

例

在下面例子，当 IR00001 为 ON，将 DM7000～DM7099 间 100 字范围里的内容拷贝到 DM0010～DM0109 里。

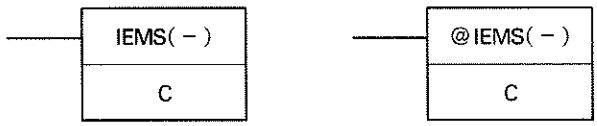


地址	指令	操作数
00000	LD	00001
00001	@XDMR(－)	
		# 0100
		# 7000
		DM 0010



5－25－15 EM 间接寻址指令－IEMS(－)

梯形符号



操作数数据区域

C：控制字
000，#E000，或 E0B1～#E0B3

限制

C 必须是 000，#E000，#E0B0，#E0B1，或 #E0B2

说明

在 ON 执行条件下执行指令时，IEMS(－)将 DM 间接寻址(＊DM)的目标转换到指定的 EM 组。当间接寻址转换到 EM 时，可以改变当前 EM 组号。

在启动中断子程序时，＊DM 的目标自动转回到 DM 区域。在每次扫描开始时，＊DM 的目标也自动转回到 DM 区域。

下表给出了 C 的允许数值及其功能

C	IEMS(－)操作
000	＊DM 目标转换到 DM 区域。
#E000	＊DM 目标转换到 EM 区域的当前组。
#E0B0	＊DM 目标转换到 EM 区域的 #0 组。
#E0B1	＊DM 目标转换到 EM 区域的 #1 组。
#E0B2	＊DM 目标转换到 EM 区域的 #2 组。

DM6031 中内容指明当前的＊DM 目标和当前的 EM 组号，如下表。

字	位 00～07	位 08～15
DM6031	当前的 EM 组号(00～02)	＊DM 目标(00：DM，01：EM)

注

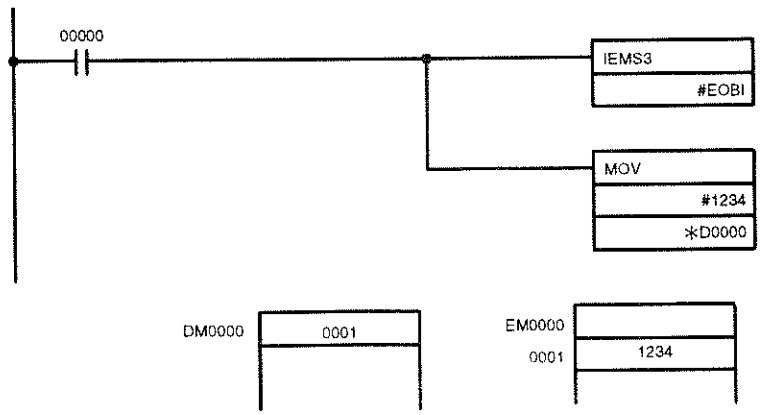
当使用替换指令时，第二和第三操作数中输入 000。

标志

ER：C 不是一个允许的数值。

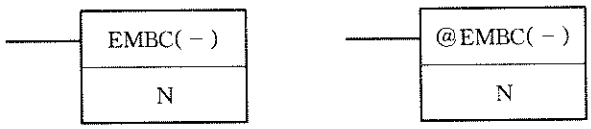
例

下例中，IEMS(－)将＊DM 目标转换至 EM 中 1＃组并使用间接寻址指令传送＃1234 到 EM1＃组中 EM0001。



5－25－16 选择 EM 组指令－EMBC(－)

梯形符号



操作数数据区域

N：组号
IR，SR，AR，DM，HR，LR，＃

限制

N 必须是 0000，0001，或 0002 并且 N 指定组号的组必须存在于 CPU 中。

说明

在 ON 执行条件下执行指令时，EMBC(－)将当前 EM 组转至指定的组号中。CPU 只能访问当前 EM 组，组号由 DM6031 中位 00～07 指示。

如果 CPU 中不存在指定的 EM 组号，会出现错误，EMBC(－)指令不执行。

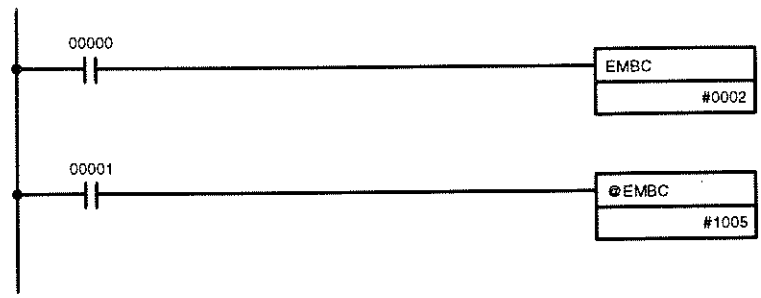
标志

ER：间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或超出 DM 区域)

CPU 中不存在 N 指定的组号。

例

在下例中，当 IR00000 为 ON，EMBC(－)转换当前组到 2＃组。

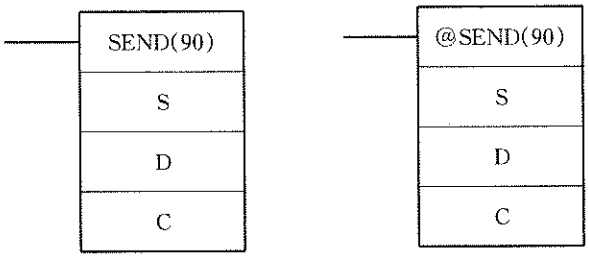


5－26 网络指令

网络指令用于通过 SYSMAC NET 链接系统，SYSMAC LINK 系统，或以太网系统与网上的其它 PC. BASIC 单元或上位机。

5－26－1 网络发送－SEND(90)

梯形符号



操作数数据区域

S：源起始字
IR，SR，AR，DM，HR，TC，LR
D：目标起始字
IR，SR，AR，DM，HR，TC，LR
C：第一控制数据字
IR，SR，AR，DM，HR，TC，LR

限制

C～C＋2 必须在同一数据区域内，必须在下面规定值以内。为了使用 SEND(90)，系统必须安装有 SYSMAC NET 链接单元，SYSMAC LINK 单元或 PC 卡单元。

说明

当执行条件为 OFF，SEND(90)指令不执行。当执行条件为 ON，SEND(90)将起始于字 S 的数据传送到 SYSMAC NET 链接，SYSMAC LINK 系统，或以太网系统指定的节点中字 D 所开始的地址里。以字 C 开始的控制字决定了传送的字数，目标节点和其它参数。控制数据的内容取决于数据是在 SYSMAC NET 链接系统或在 SYSMAC LINK 系统，还是在以太网系统中传送。

字 C＋1 的位 15 状态决定指令是用于 SYSMAC NET 链接系统，或 SYSMAC LINK/以太网系统。

控制数据

以太网系统

目标节点号设为 0，则将数据传送到所有节点。

详见“PC 卡单元操作手册”。

字	位 00～07	位 08～15
C	传送字的数目(0～1000，用 4 位十六进制，即 0000H～03E8H)	
C＋1	响应时向限制(0.1 和 25.5 秒，0.1 秒为增量单元，用 2 位十六进制，无小数点，即 01H～FFH) 缺省值为 00H(2.2 秒)	位 08～11：重复次数(0～15，用十六进制，即 OH～FH) 位 12：ON：间接寻址 OFF：直接寻址 位 13：ON：响应不返回 OFF：响应返回 位 14：ON：0 级操作 OFF：1 级操作 位 15：置为 1。
C＋2	目标节点(0－127，用二位十六进制，即 00H～7FH)	位 08～12：目标节点的单元地址 置为 00H 位 13～15：置为 0

SYSMAC NET 链接系统

目标端口总是设为 0。目标节点号设为 0，则将数据传送到所有节点。网络号设为 0，则将数据传送到同一子系统(即网络)上的一个节点。详见“SYSMAC NET 链接系统手册”。

字	位 00～07	位 08～15
C	传送字的数目(0～1000，用 4 位十六进制，即 0000H～03E8H)	
C＋1	网络号(0～127，用 2 位十六进制，即 00H～7FH)	位 14 ON：0 级操作 OFF：1 级操作 位 08～13 和 15：置为 0
C＋2	目标节点(0－126，用 2 位十六进制，即 00H～7EH)＊	目标端口 NSB：00 NSU：01/02

＊执行网络发送操作的 PC 的节点数可能被设置。

SYSMAC LINK 系统

目标节点号设为 0，则将数据传送给所有节点。(详见 SYSMAC LINK 系统手册)

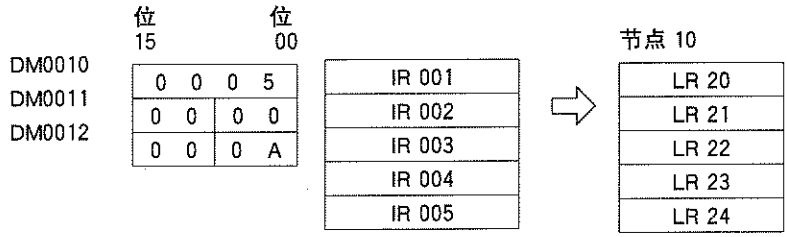
字	位 00～07	位 08～15
C	传送字的数目(0～1000，用 4 位十六进制，即 0000H～03E8H)	
C＋1	响应时间限制(0.1 和 25.4 秒，用 2 位十六进制，无小数点，即 00H～FFH) 注：如果限制设为 OH，响应时间为 2 秒。如果设为 FFH，就设有时间限制。	位 08～11：重复次数(0～15，用十六进制表示，即 0H～FH) 位 12：置为 0 位 13 为ON：响应不返回 OFF：应返回 位 14 为ON：0 级操作 OFF：1 级操作 位 15：置为 1。
C＋2	目标节点(0－62，用 2 位十六进制表示，即 00H～3EH)＊	置为 0

＊执行网络发送操作的 PC 的节点数不能被设置。

例：这个例子用于 SYSMAC NET 链接系统。当 0000 为 ON，下列程序将从 IR001 到 IR005 的内容传送到在节点 10 上的 LR20 到 LR24。



地址	指令	操作数
00000	LD	00000
00001	SEND(90)	
		001
		LR 20
		DM 0010



标志

ER：指定的节点号大于 126(在 SYSMAC NET 链接系统中)，或大于 62(在 SYSMAC LINK 系统中)，或大于 127(在以太网系统中)。

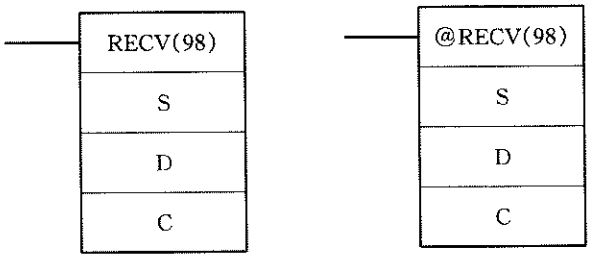
被传送数据超出数据区域。

间接寻址的 DM 字不存在。(＊DM 字不是 BCD 码，或超出 DM 区域)

没有 SYSMAC NET 链接/SYSMAC LINK/PC 卡单元。

5－26－2 网络接收－RECV(98)

梯形符号



操作数数据区域

S:源起始字
IR, SR, AR, DM, HR, TC, LR
D:目标起始字
IR, SR, AR, DM, HR, TC, LR
C:第一控制数据字
IR, SR, AR, DM, HR, TC, LR

限制

C～C＋2 必须在同一数据区域内，必须在下面规定值以内。为了使用 RECV(98)，系统必须安装有 SYSMAC NET 链接单元，SYSMAC LINK 单元，或 PC 卡单元。

说明

当执行条件为 OFF，RECV(98)不执行。当执行条件为 ON，RECV(98)将 SYSMAC NET 链接，SYSMAC LINK，以太网系统一个节点中起始于 S 的数据传送到起始于 D 的字中。以字 C 开始的控制字提供接收字数，源节点和其它传送参数。

字 C＋1 位 15 的状态决定指令用于 SYSMAC NET 链接系统或 SYSMAC LINK/以太网系统。

控制数据

以太网系统

详见“PC 卡单元操作手册”。

字	位 00～07	位 08～15
C	接收字数(0～1000，用 4 位十六进制，即 0000H～03E8H)	
C＋1	响应时间限制(0.1 和 25.5 秒，0.1 秒为增量单元，用 2 位十六进制，无小数点，即 00H～FFH) 缺省值为 00H(2.2 秒)	位 08～11:重复次数(0～15，用十六进制，即 0H～FH) 位 12:ON:间接寻址 OFF:直接寻址 位 13:ON:响应不返回 OFF:响应返回 位 14:ON:0 级操作 OFF:1 级操作 位 15:置为 1。
C＋2	源节点(0－127，用 2 位十六进制，即 00H～7FH)	位 08～12:源节点的单元地址 置为 OOH 位 13～15:置为 0

SYSMAC NET 链接系统 源端口号总是设为 0。网络号设为 0，则接收同一子系统(网络)中的一个节点的数据，详见“SYSMAC NET 链接系统手册”。

字	位 00～07	位 08～15
C	接收字数(0～1000,用 4 位十六进制,即 0000H～03E8H)	
C＋1	网络号(0～127,用 2 位十六进制,即 00H～7FH)	位 14 ON:0 级操作 OFF:1 级操作 位 08～13 和 15:置为 0
C＋2	源节点(1－126,用 2 位十六进制,即 01H～7EH)	源端口 NSB:00 NSU:01/02

SYSMAC LINK 系统

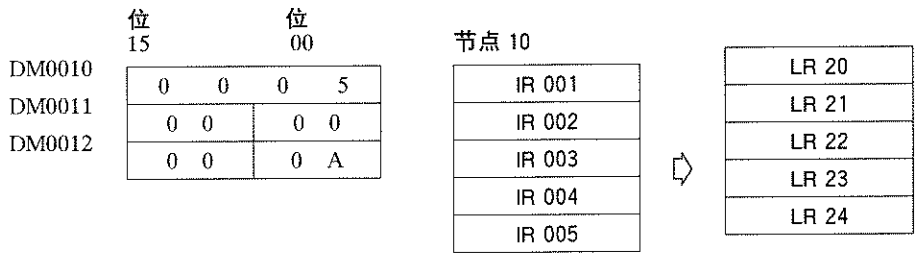
详见“SYSMAC LINK 系统手册”。

字	位 00～07	位 08～15
C	接收字数(0～256,用 4 位十六进制,即 0000H～0100H)	
C＋1	响应时间限制(0.1 和 25.4 秒,2 位十六进制,无小数点,即 00H～FFH) 注:如果限制设到 0H,响应时间为 2 秒。如果设到 FFH,就没有时间限制。	位 08～11:重复次数(0～15,用十六进制表示,即 OH～FH) 位 12:置为 0 位 13:置为 0 位 14 为ON:0 级操作 为 OFF:1 级操作
C＋2	源节点(0－62 用 2 位十六进制表示,即 00H～3EH)	位 15:置为 0

例:这个例子用于 SYSMAC NET 链接系统。当 0000 为 ON,下列程序就将 IR001～IR005 的内容传送到节点 10 中的 LR20～LR24。



地址	指令	操作数
00000	LD	0000
00001	RECV(98)	
		001
		LR 20
		DM 0010



标志

ER:指定的节点号大于 126(在 SYSMAC NET 链接系统中),大于 62(在 SYSMAC LINK 系统中),或大于 127(在以太网系统中)。

被接收的数据超出数据区域。

间接寻址的 DM 字不存在。(* DM 字不是 BCD 码或超出 DM 区域)

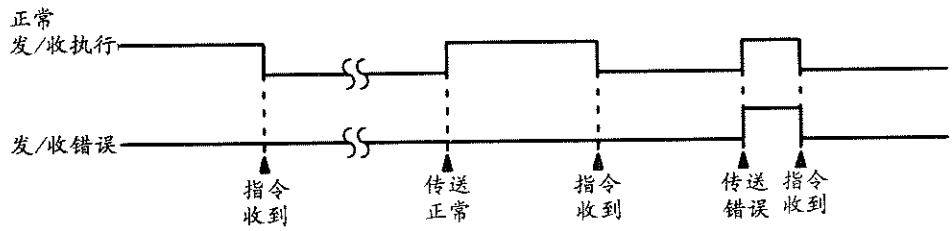
没有 SYSMAC NET 链接/SYSMAC LINK/PC 卡单元。

5－26－3 关于网络通讯

SEND(90)和 RECV(98)是在命令/响应处理基础上,也就是,只有在发送节点收到并确认目标节点传来的响应才能完成一个传送。注意 SEND(90)/RECV(98)的允许标志只有在传送结束后第一个 END(01)时才能为 ON。关于命令/响应操作详见"SYSMAC NET 链接系统手册"或"SYSMAC LINK 系统手册。"

如果需要重复使用 SEND(90)/RECV(98),必须用下列标志,以保证在进一步执行 SEND(90)/RECV(98)操作时,以前的操作已经完成。

SR 标志	功　能
SEND(90)/RECV(98)允许标志 (SR 25201, SR25204)	SEND(90)/RECV(98)执行期间为 OFF(包括命令响应处理)。如果标志不为 ON,不能开始 SEND(90)/RECV(98)操作。
SEND(90)/RECV(98)错误标志 (SR25200, SR25203)	SEND (RECV)正常完成(即接收到响应信号)后为 OFF。 SEND (RECV)错误后为 ON,直到下一个 SEND(90)/RECV(98)操作。 错误类型: 超时(命令/响应时间大于 1 秒) 传送数据错误

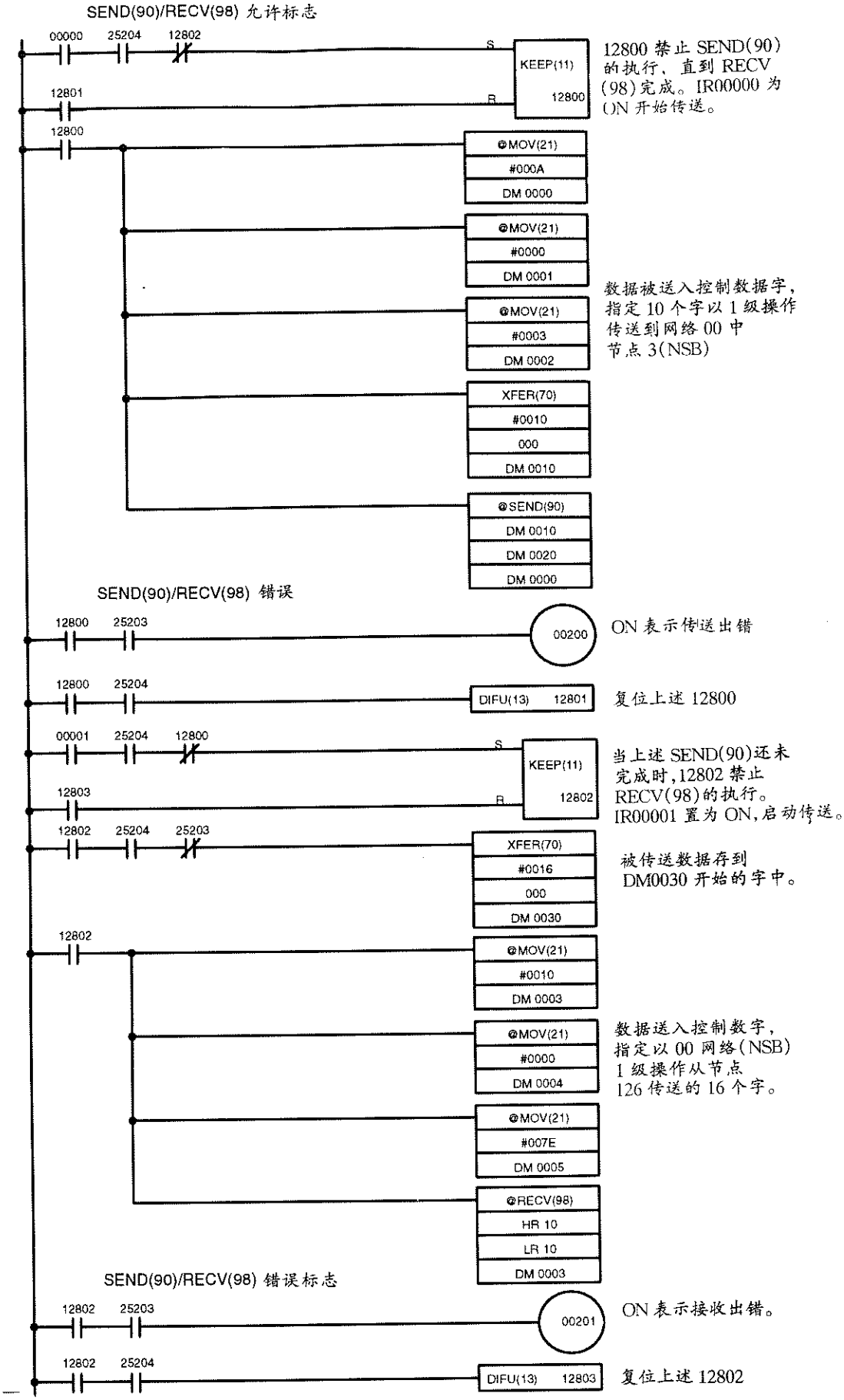


SEND(90)/RECV(98)的数据处理

当 SEND(90)/RECV(98)执行时,根据 SEND (90)和 RECV (98)指令,数据被传送给所有 PC。在对外围设备和链接单元的服务过程中完成,对传送/接收做最后的处理。

编程举例　重复使用 SEND(90)/RECV(98)

为了保证成功地操作 SEND(90)/RECV(98),你的程序必须使用 SEND(90)/RECV(98)允许标志和错误标志,以保证执行。下列程序是用于 SYSMAC NET 链接系统的一个例子。


SEND(90)/RECV(98) 允许标志
00000
25204
12802
S
KEEP(11)
12800
12801
R
12800 禁止 SEND(90)的执行，直到 RECV(98)完成。IR00000 为 ON 开始传送。
12800
@MOV(21)
#000A
DM 0000
@MOV(21)
#0000
DM 0001
@MOV(21)
#0003
DM 0002
数据被送入控制数据字，指定 10 个字以 1 级操作传送到网络 00 中节点 3(NSB)
XFER(70)
#0010
000
DM 0010
@SEND(90)
DM 0010
DM 0020
DM 0000
SEND(90)/RECV(98) 错误
12800
25203
00200
ON 表示传送出错
12800
25204
DIFU(13)
12801
复位上述 12800
00001
25204
12800
S
KEEP(11)
12802
12803
R
当上述 SEND(90)还未完成时，12802 禁止 RECV(98)的执行。IR00001 置为 ON，启动传送。
12802
25204
25203
XFER(70)
#0016
000
DM 0030
被传送数据存到 DM0030 开始的字中。
12802
@MOV(21)
#0010
DM 0003
@MOV(21)
#0000
DM 0004
数据送入控制数字，指定以 00 网络(NSB)1 级操作从节点 126 传送的 16 个字。
@MOV(21)
#007E
DM 0005
@RECV(98)
HR 10
LR 10
DM 0003
SEND(90)/RECV(98) 错误标志
12802
25203
00201
ON 表示接收出错。
12802
25204
DIFU(13)
12803
复位上述 12802

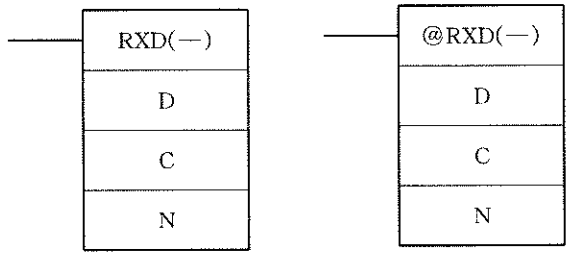
地址	指令	操作数
00000	LD	00000
00001	AND	25204
00002	AND NOT	12802
00003	LD	12801
00004	KEEP(11)	12800
00005	LD	12800
00006	@MOV(21)	
		# 000A
		DM 0000
00007	@MOV(21)	
		# 0000
		DM 0001
00008	@MOV(21)	
		#0003
		DM 00002
00009	@XFER(70)	
		# 0010
		000
		DM 0002
00010	@SEND(90)	
		DM 0010
		DM 0020
		DM 0000
00011	LD	12800
00012	AND	25203
00013	OUT	00200
00014	LD	12800
00015	AND	25204
00016	DIFU(13)	12801
00017	LD	00001
00018	AND	25204

地址	指令	操作数
00019	AND NOT	12800
00020	LD	12803
00021	KEEP(11)	12802
00022	LD	12802
00023	AND	25204
00024	AND NOT	25203
00025	XFER(70)	
		# 0016
		000
		DM 0030
00026	LD	12802
00027	@MOV(21)	
		# 0010
		DM 0003
00028	@MOV(21)	
		# 0000
		DM 0004
00029	@MOV(21)	
		# 007E
		DM 0005
00030	@RECV(98)	
		HR 10
		LR 10
		DM 0003
00031	LD	12802
00032	AND	25203
00033	OUT	00201
00034	LD	12802
00035	AND	25204
00036	DIFU(13)	12803

5－27 串行通信指令

5－27－1 接收指令－RXD(－)

梯形符号



操作数数据区域

D:第一个目标字
IR, SR, AR, DM, HR, TC, LR
C:控制字
IR, SR, AR, DM, HR, TC, LR, #
N:字节数
IR, SR, AR, DM, HR, TC, LR, #

限制

D 和 D+(N÷2)－1 必须在同一数据区。

N 必须是从＃0000～＃0256 的 BCD 码。

说明

如果执行条件是 OFF,不执行 RXD(－)。如果执行条件是 ON,RXD(－)读出 N 个在外设端口接收到的数据字节,然后把那些数据写入到字 D～D+(N÷2)－1。一次最多能读出 256 字节数据。

如果接收到字节少于 N 个,那末读出接收到的全部数据。

注

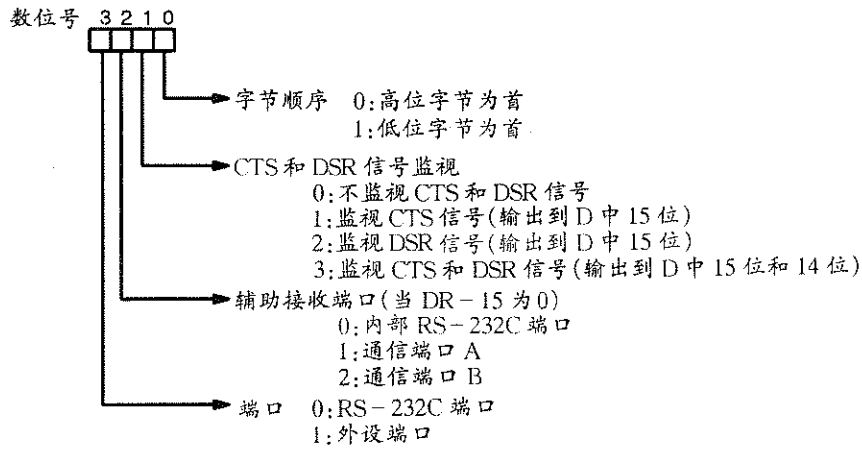
只有在经外设端口或 RS－232C 端口接收数据时需要 RXD(－)。从上位计算机到上位机链接单元的传送数据是自动处理的,不需要进行编程。

注意

一旦接收到 256 字节的数据,但用 RXD(－)指令读不出该接收数据,那 PC 就不能接收更多的数据。在接收完成标志转变为 ON(SR26414 用于外设端口,SR26406 用于 RS－232C 端口)。则尽快读出数据。

控制字

控制字的数值决定哪一个端口读出数据和采用什么顺序把数据写入到存储器。



数据写入存储器的次序取决于 C 中 0 数位的数值。

数据 12345678…八个字节用下列方式写入。

0 位数＝0

	高位	低位
D	1	2
D＋1	3	4
D＋2	5	6
D＋3	7	8
・	・	・
・	・	・
・	・	・

0 位数＝1

	高位	低位
D	2	1
D＋1	4	3
D＋2	6	5
D＋3	8	7
・	・	・
・	・	・
・	・	・

标志

ER：CPU 没装 RS－232 端口。

另一装置没有与指定的端口相连。

通信设置(PC 设置中)或操作数设置上有错。

间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 区域)

目标字(D～D＋(N÷2)－1)超出数据区域。

外设端口

26414：当在外设端口上数据被正常接收，SR26414 变为 ON，当执行 RXD(－)指令时，数据被读入，SR26414 复位。

266： SR266 包含外设端口上接收到的字节数，当执行 RXD(－)指令时，SR266 复位为 0000。

RS－232C

26406：当在 RS－232C 端口上数据被正常接收，SR26404 变为 ON，当执行 RXD(－)指令时数据被读入，SR26404 复位。

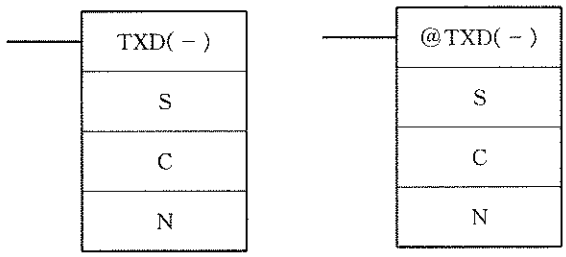
265： SR265 包含 RS－232C 端口上接收到的字节数，当执行 RXD(－)指令时，SR265 复位至 0000。

注

给 N 设定 0000 或使用端口复位位(SR25208，用于外设端口，SR25209 用于 RS－232C 端口)都使通信标志和计数器清零。

5－27－2 发送指令－TXD(－)

梯形符号



操作数数据区域

S：第一个源字
IR，SR，AR，DM，HR，TC，LR
C：控制字
IR，SR，AR，DM，HR，TC，LR，＃
N：字节数
IR，SR，AR，DM，HR，TC，LR，＃

限制

S 和 S＋(N÷2)－1 必须在同一数据区。

N 必须是从＃0000～＃0256 的 BCD 码。(在上位机链接方式中是＃0000～＃0061)

说明

当执行条件是 OFF，不执行 TXD(－)指令；当执行条件是 ON，TXD(－)指令从字 S～S＋(N÷2)－1 的 N 个字节里的读出数据，转换成 ASCII 码，并且从指定的端口输出数据。

TXD(－)指令的操作在上位机链接方式中和 RS－232C 方式中是不同的，因此要分别解释这些方式。

注

下面的标志为 ON，表明可能通过各种端口进行通讯，确认在执行 TXD(－)前响应标志为 ON

SR26405：RS－232C 端口

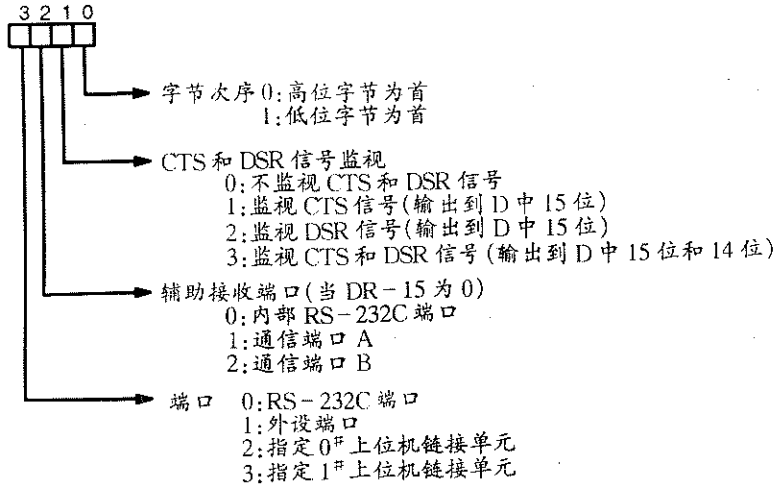
SR26413：外设端口

SR26705：＃0 上位机链接单元

SR26713：＃1 上位机链接单元

上位机链接方式

N 必须是＃0000～＃0061 之间的 BCD 码(即相当于最多 122 个字节)。控制字的数值决定数据以哪个端口输出，如下所示

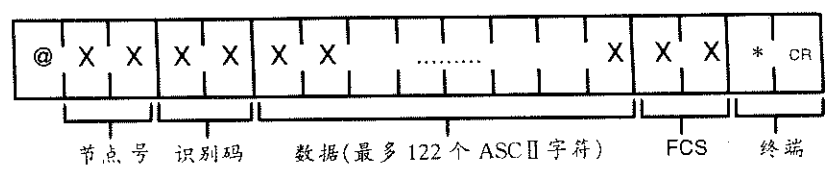


从 S－S＋(N÷2)－1 中读出指定的字节数，转换成 ASII 码，并通过指定的端口传送。

下面给出按：12345678…顺序传送源数据字节。

	高位	低位
S	1	2
S＋1	3	4
S＋2	5	6
S＋3	7	8
・	・	・
・	・	・
・	・	・

下图所示从 PC 送至上位机链接命令(TXD)格式。C200HX/HG/HE 和 C200HS 根据它们的设定自动地加上前缀和后缀，例如；节点号，识别码和 FCS。

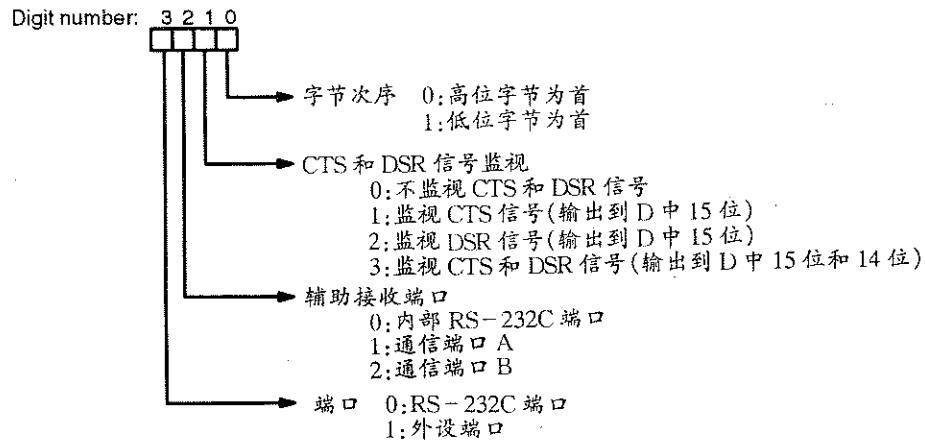


RS－232C 方式

N 必须是＃0000～＃0256 之间的 BCD 码。控制字的数值决定数据从哪一个端口输出和数据写进存储器的顺序。

控制字

控制字的数字决定数据从哪一个端口读出和数据按什么次序写入存储器。



从 S～S＋(NP2)－1 中读出指定的字节数，并且通过指定的端口传送。

	高位	低位
S	1	2
S＋1	3	4
S＋2	5	6
S＋3	7	8
・	・	・
・	・	・
・	・	・

当 C 的 0 位数是 0，上面所示源数据字节传送次序是：1 2 3 4 5 6 7 8

当 C 的 0 位数是 1，上面所示源数据字节传送次序是：2 1 4 3 6 5 8 7

注：

当指定开始和结束编码时，包括开始和结束编码的数据总长度应该是 256 节。

标志

ER：另一装置没有与外设端口相连。

在通信设置(PC 设置中)或者操作数设置上有错误。

间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 区域)

源字(S～S＋(N÷2)－1)超出数据区域。

26405：RS－232C 端口通信有效标志

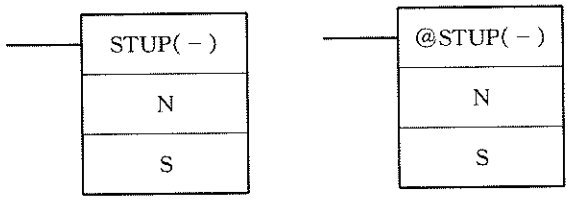
26413：外设端口通信有效标志

26705：＃0 上位机链接单元通信有效标志

26713：＃1 上位机链接单元通信有效标志

5－27－3 修改 RS－232C 设置指令－STUP(－)

梯形符号



操作数数据区域

N：RS－232C 端口指定器
IR000，IR001，或 IR002

S：第一个源字
IR，SR，AR，DM，HR，TC，LR，＃

限制

N 必须是 IR000，IR001，或 IR002。

S 和 S＋4 必须在同一数据区。

(置 S 为 0，将 RS－232C 设置转为缺省设置)

如果 DIP 开关插脚 2 为 ON，不能为 RS－232C 端口执行 STUP(－)指令。

在中断子程序中不能执行 STUP(－)指令。

说明

当指令执行条件为 OFF，不执行 STUP(－)指令。当执行条件为 ON，STUP(－)修改 PC 设置中由 N 指定的端口的设置。

N 确定 RS－232C 设置中哪一部分变更。

N	指定端口
IR000	内置 RS－232C 端口(PC 设置：DM6645～DM6649)
IR001	通信板端口 A(PC 设置：DM6555～DM6559)
IR002	通信板端口 B(PC 设置：DM6550～DM6554)

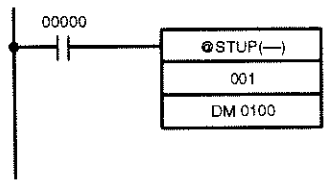
如果 S 是一个字地址，字 S～S＋4 中内容复制 PC 设置中 5 个字中，这 5 个字存有由 N 指定端口的设定值。

如果 S 输入常数＃0000，指定端口的设置回到缺省设定值。

S	功　能
字地址	S～S＋4 中内容复制到含有 N 指定端口设定值的 PC 设置区
常数＃0000	由 N 指定的端口设置回复到缺省设定

应用例子

例子表示将 DM0100～DM0104 中内容传送到通信板端口 A 的 PC 设置区(DM6555～DM6559)



地址	指令	操作数
00000	LD	00000
00001	@STUP(－)	
		001
		DM　0100

设定值传送如上所示。当传送完成，修改 RS－232C 设置标志(SR27504)变为 OFF。

DM0100	1001	DM6555	1001
DM0101	0803	DM6556	0803
DM0102	0000	DM6557	0000
DM0103	2000	DM6558	2000
DM0104	0000	DM6559	0000

下表说明被传送的设置数据的作用

字	内　容	作　　用
DM0100	1001	允许 DM0101 中通信设置，通信方式设为 RS－232C。
DM0101	0803	设置下列通信设定 9,600bPS,1 起动位,8 位数据,1 停止位,无校验
DM0102	0000	无传输延时(oms)
DM0103	2000	允许使用结束码 CR,LF
DM0104	0000	…

标志

ER:间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码或超出 DM 区域)

端口指定器(N)不是 IR000,IR001,或 IR002。

指定端口 A,但 DIP 开关上插脚 2 为 ON。

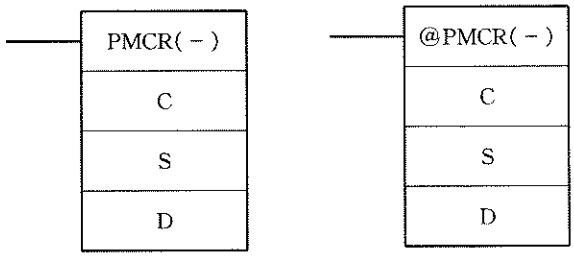
PC 设置写保护。(DIP 开关上插脚 1 为 ON)

指定源字超出数据区域。

在中断程序中执行 STUP(－)指令。

5－27－4　通信协议宏指令－PMCR(－)

梯形符号



操作数数据区域

C:控制字
IR, SR, AR, DM, HR, TC, LR, #
S:第一输出字
IR, SR, AR, DM, HR, TC, LR, #
D:第一输入字
IR, SR, AR, DM, HR, TC, LR

限制

C 必须是＃1000～＃2999 之间的 BCD 码。

DM6144～DM6655 不能用于 D。

说明

当执行条件为 OFF,不执行 PMCR(－)指令。当执行条件为 ON,PMCR(－)调用并执行指定的通信序列(通信协议数据),这序列已在安装在 PC 上通信板上登记过。

如果 S 和 D 没有设定在 DM 区,由通信板上登记的通信序列发送/接收的信息必须设到读出或写入字数据中。当不须为第一个输出字设数据字时,可用一个常数。

如通信序列不需要输入字,任意指定一个字地址。在指定的地址内并不存入数据,还保留原有内容。当

通信序列不需要数个输入字。请指定不在程序其他地方使用的字。

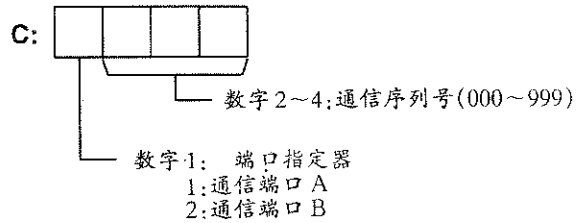
输入和输出字(S 和 D)可以在通信板上登记的通信序列中设置。

注

有关通信板详见“通信板操作手册”,有关通信序列详见“通信协议软件操作手册”。

控制字

控制字的第 1 个数(1 或 2)指定通信端口,最后 3 个数指定通信序列号(000～999),如下所示。



标志

ER:间接寻址 DM 字不存在。(* DM 字的内容不是 BCD 码或超出 DM 区域)

D 不是 BCD 码或使用字 DM6144～DM6655。

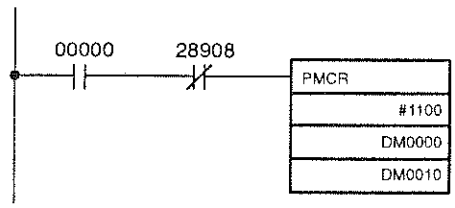
在执行指令时,另一条 PMCR(－)指令正在进行。

端口指定器不是 1 或 2。

例

当 IR00000 为 ON, SR28908(通信板端口 A 指令执行标志)为 OFF 时,调用通信板上通信序列 100,通过通信板端口 A 传送数据。

从始于 DM0000(第一输出字)的范围读出发送数据,在从 DM0010(第一输入字)开始的范围内存入接收数据。



地址	指令	操作数
00200	LD	00000
00201	AND NOT	28908
00202	PMCR(－)	
		# 1100
		DM 0000
		DM 0010

5－28 高级 I/O 指令

高级 I/O 指令能用单一指令控制先前的包含外部 I/O 设备(数字开关,7 段显示器等等)的复杂操作。

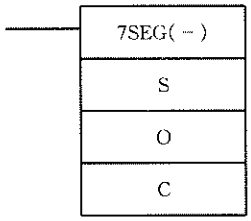
如下表所示,有五条高级 I/O 指令。它们全部是扩展指令,在使用前必须赋于功能码。

名称	记忆符号	功 能
七段显示器输出	7SEG(－)	BCD 码输出到 7 段显示器
数字开关输入	DSW(－)	从数字开关输入数据
十六进制键输入	HKY(－)	从十六进键输入十六进制数
十进制键输入	TKY(－)	从十进制键输入 BCD 码
矩阵输入	MTR(－)	从一个 8×8 矩阵输入数据

虽然 TKY(－)指令仅用于简化程序,但其它高级 I/O 指令被用来缩短扫描周期时间,减少特殊 I/O 单元的需要从而减少系统费用。除 TKY(－)指令之外,高级 I/O 指令在程序中仅能使用一次,并且不能用于安装在从站机架上 I/O 单元,因此在那里必须使用特殊 I/O 单元。

5－28－1 七段显示输出指令－7SEG(－)

梯形符号



操作数数据区域

S:第一个源字
IR, SR, AR, DM, HR, TC, LR

O:输出字
IR, SR, AR, DM, HR, LR

C:控制数据
000～007

限制

S 和 S＋1 必须在同一数据区。

DM0000～DM6143 可以用于输出字 0。

不能置 C000～007 以外,其它数值。

概述

当执行条件是 OFF,不执行 7SEG(－)指令;当执行条件是 ON,7SEG(－)指令读出源数据(4 位数或 8 位数),把它转换成 7 段显示数据,输出数据送到由 O 指示的 7 段显示器上。

C 的数值指定了源数据的位数及输入和输出单元的逻辑关系。如下表所示:

源数据	显示器的数据输入逻辑	显示器的锁存输入逻辑	C
4 位数(S)	同输出单元	同输出单元	000
		不同于输出单元	001
	不同于输出单元	同输出单元	002
		不同于输出单元	003
8 位数 (S,S＋1)	同输出单元	同输出单元	004
		不同于输出单元	005
	不同于输出单元	同输出单元	006
		不同于输出单元	007

如果有 8 位源数据,则放入 S 和 S＋1,最高位放入 S＋1;如果只有 4 位源数据,则放入 S。

7SEG(－)指令用 12 个循环周期显示这 4 位或 8 位数据,然后重复显示这些数据。

对每一个显示数据,7 段显示器必须提供四条数据线和一条锁存信号线。

注

1.当设计系统时,要考虑扫描周期时间和 7 段显示器的特性。

2.这里不用的输出位,能按照普通输出位使用。

注意事项

为保证 7SEG(－)指令的有效执行,每次都必须完成对 7SEG(－)指令所使用的全部I/O 点的刷新。每当在子程序中使用 7SEG(－)指令,为保证每次执行 7SEG(－)时能刷新所有用到的 I/O 点,I/O 刷新指令

必须和 7SEG(－)指令一起使用。参阅 330 页这种类型程序的例子。

每当启动程序执行,包括电源中断后的再启动,7SEG(－)从第一个扫描周期开始执行。

在程序中 7SEG(－)指令只能使用一次。

7SEG(－)不能用于安装在从站机架上的 I/O 单元。

硬件

这条指令将字数据输出到 7 段显示器,对于 4 位数它使用 8 个输出位,对于 8 位数它使用 12 个输出位。7 段显示器与输出单元的连线如下图所示,对于 4 位数的显示,数据输出(D0～D3)与输出点 0～3(分配字 O)相连接,而锁存输出(CS0～CS3)连到输出点 4～7。当数据显示一周以后,输出点 12(对于 8 点显示)或者输出点 8(对于 4 点显示)被置成 ON。它们的连接并不是必须的,除非使用上有要求。

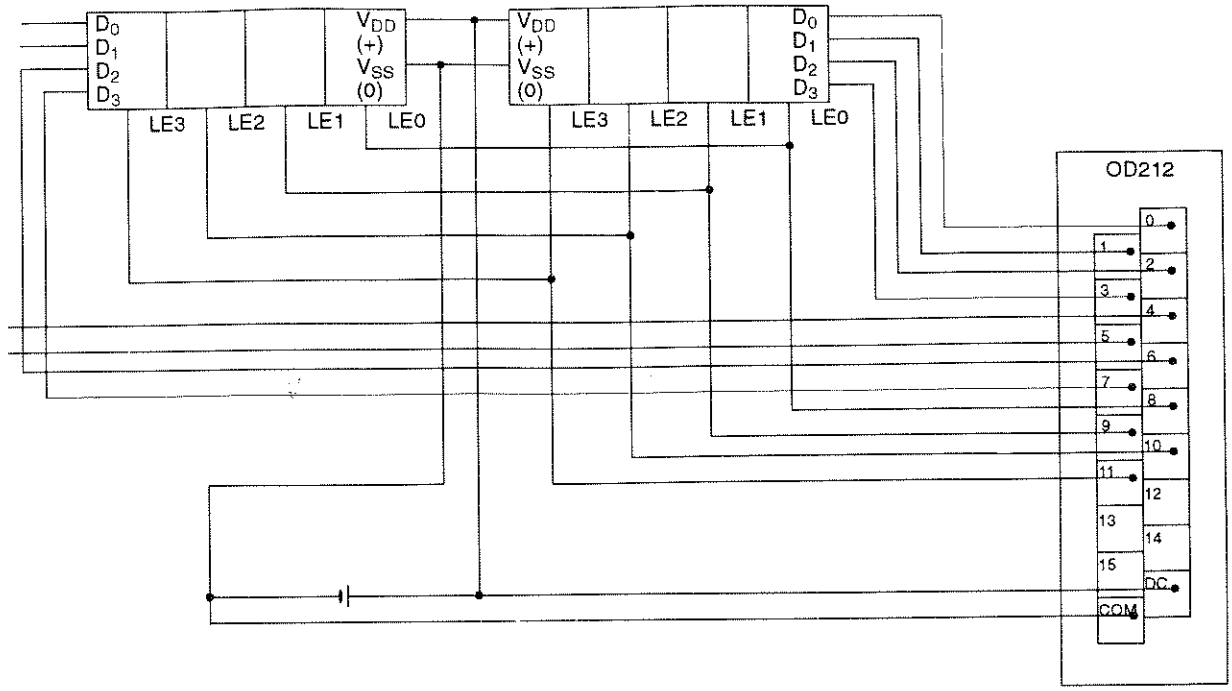
对于 4 位数位的显示必须使用带有 8 点或更多输出点的输出单元,而对于 8 位数位的显示,必须使用带有 16 点或更多输出点的输出单元。可以使用基本输出,特殊 I/O,或高密度输出单元。

注

1.输出单元的输出一般使用负逻辑。(仅当 PNP 输出类型时使用正逻辑)。

2.7 段显示器需要正还是负逻辑,取决其类型。

3.7 段显示器在每位数字上必须有四根数据信号线和一根锁存信号线。



使用指令

如果由 S 来指定保存显示数据的第一个字,输出字由 O 来指定,而从下表得到的 SV 由 C 来指定,那末当程序执行时,操作如下示进行。如果仅仅显示 4 位数,只使用 S 字。

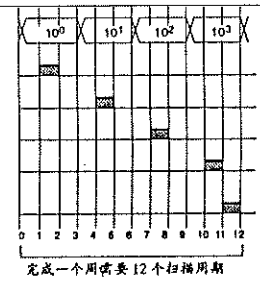
数据存贮格式

最左 4 位数	最右 4 位数
S+1	S

时序

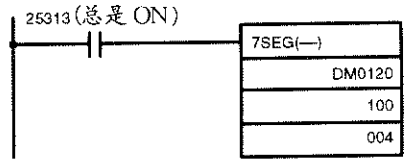
输出数据的时序如下表所示。"S"是保存显示数据的第 1 个字,"O"是输出字。

功　能	O 中的位		输出状态(数据和锁存逻辑取决于 C)
	4 位数一组	4 位数二组	
数据输出	00～03	00～03 04～07	注 0～3：字 S 数据输出 4～7：字 S＋1 数据输出
锁存输出 0	04	08	
锁存输出 1	05	09	
锁存输出 2	06	10	
锁存输出 3	07	11	
一周标志：	08	12	



应用例子

这例子给出了在 7 段 LED 显示器上显示 8 位 BCD 码数字的程序。假设 7 段显示器与输出字 IR100 相连，同时假设输出单元使用负逻辑，而用于数据信号、锁存信号的 7 段显示器逻辑也是负逻辑。



借助 7SEG(－)指令，在 DM0120(最右四位数)和 DM0121(最左四位数)中 8 位 BCD 码数据始终显示着。当 DM0120 和 DM0121 的内容改变，显示器的内容也随之改变。

标志

ER：S 和 S＋1 不在同一数据区。(当设置 8 位显示数据时)

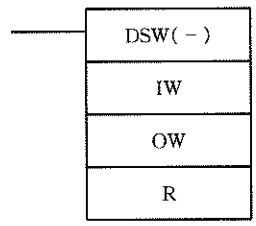
间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码，或者超出 DM 区域)

在操作数设置上有错误。

25409：在执行 7SEG(－)指令时 SR25409 将是 ON。

5－28－2　数字开关输入指令－DSW(－)

梯形符号



操作数数据区域

IW：输入字
IR, SR, AR, DM, HR, LR

OW：输输出字
IR, SR, AR, DM, HR, LR

R：第一结果字
IR, SR, AR, DM, HR, LR

概述

DSW(－)用来读出与 I/O 单元相连的数字开关设置值。当执行条件是 OFF，不执行 DSW(－)指令。当执行条件是 ON 时，DSW(－)指令从 IW 中读出数字开关上设定的 8 位数，并将结果存到 R 中。

这 8 位数值存入 R 和 R＋1，最高位放入在 R＋1。

DSW(－)执行 20 次，读出这八位数，然后重新开始继续读出数据。

数字开关必须为每一个输入数位提供 4 条数据线和一条锁存信号线及一条读出信号线。

注意事项

为保证 DSW(－)指令的有效执行，每次都必须完成对 DSW(－)指令所使用的全部 I/O 点的刷新。每当在子程序中使用 DSW(－)指令时，为了保证每次执行 DSW(－)指令时能刷新所有用到的 I/O 点，I/O 刷新指令必须与 DSW(－)一起使用。参阅 330 页这种类型程序的例子。

每当启动程序执行，包括电源中断后的再起动，DSW(－)从第一个扫描周期开始执行。

在程序中 DSW(－)只能使用一次。

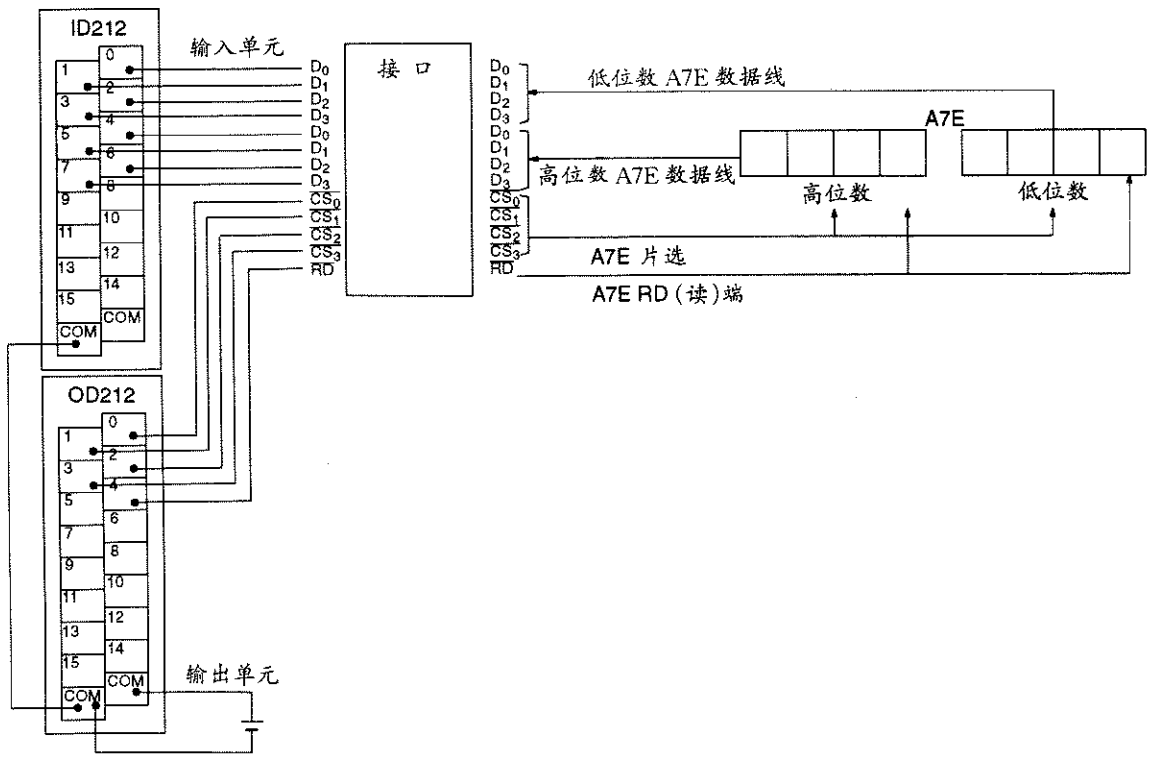
DSW(－)不能用于装在从站机架上的 I/O 单元。

注

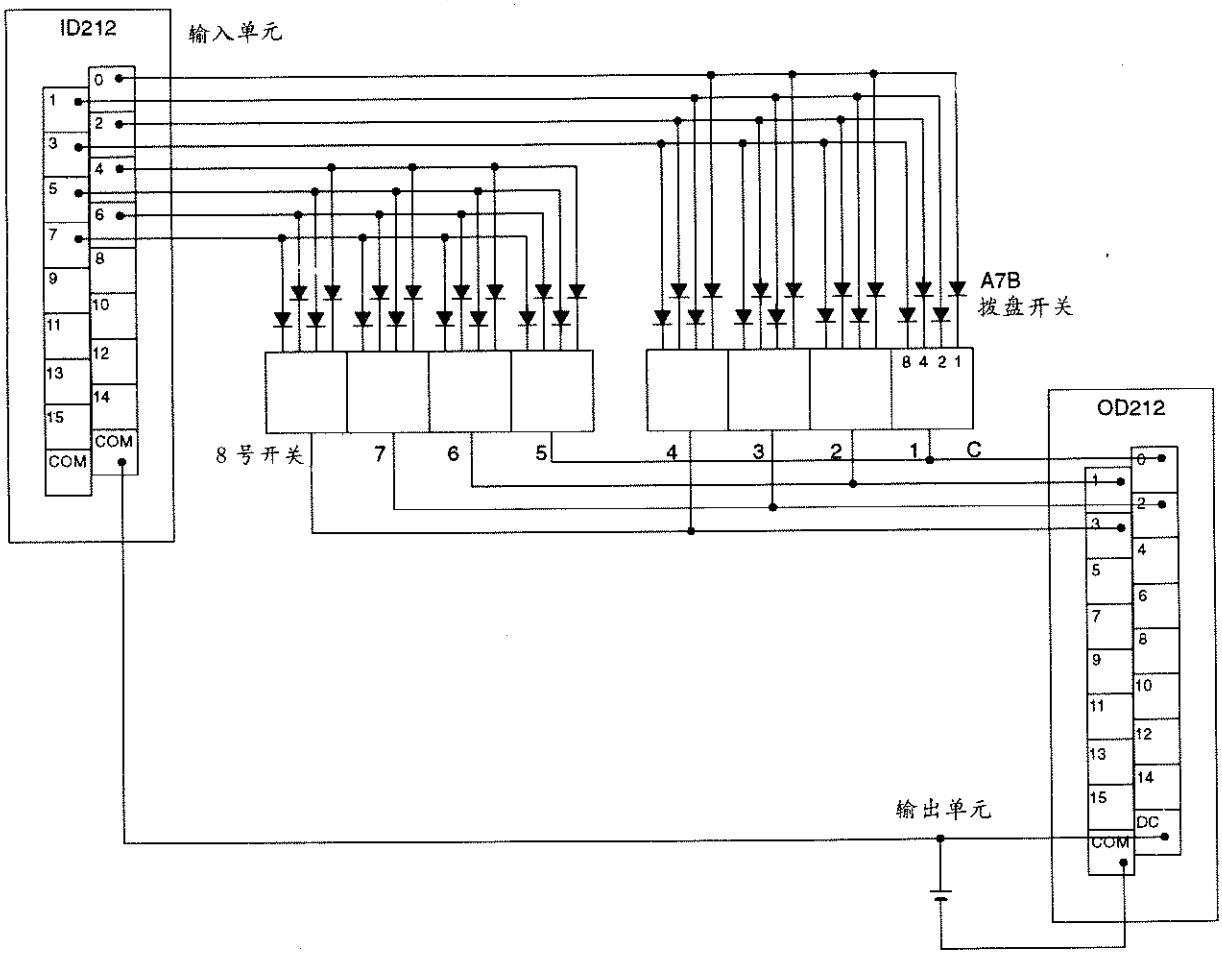
这里未用到的输入和输出位可以用作为普通输入和输出位。

硬件

用这条指令，从数字开关上读取 8 位 BCD 码设置值。DSW(－)使用 5 个输出位和 8 个输入位。数字开关和输入输出单元的连接如下图所示。当一个数据读出循环后，输出点 5 转变成 ON，若无应用上需要，不必连接该输出点。



下图例子说明：A7B 拨盘开关的连接



注

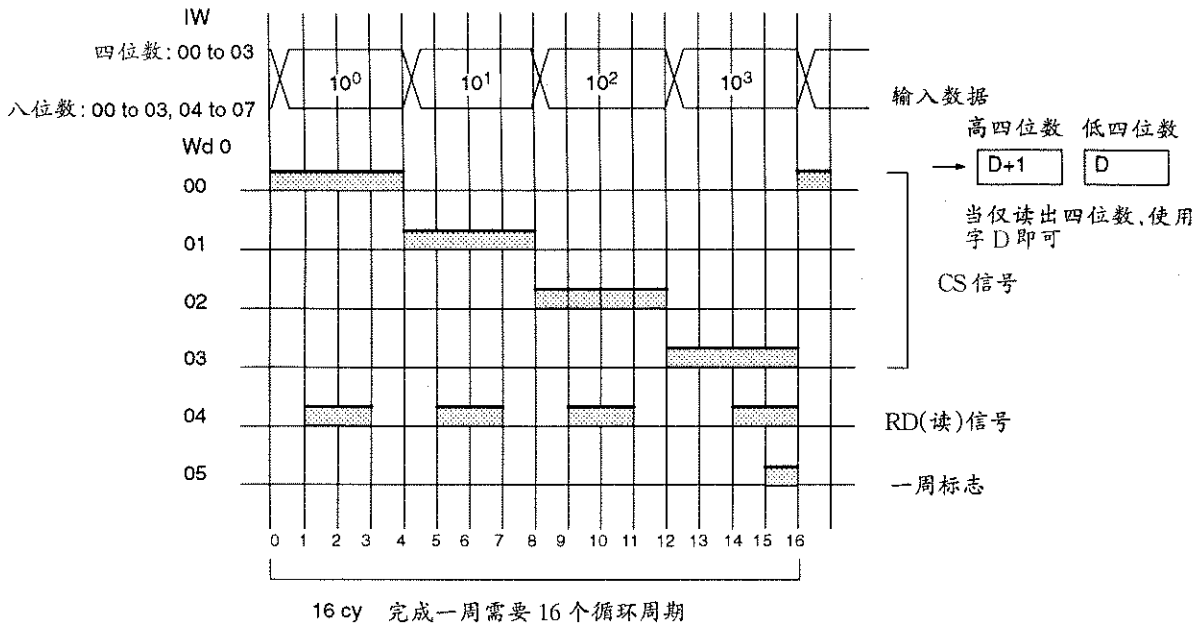
在这个例子中不需要数据读出信号。

输入必须与有 8 个或更多输入点的直流输入单元相连；

输出必须与有 8 个或更多输出点的晶体管输出单元相连。

指令使用

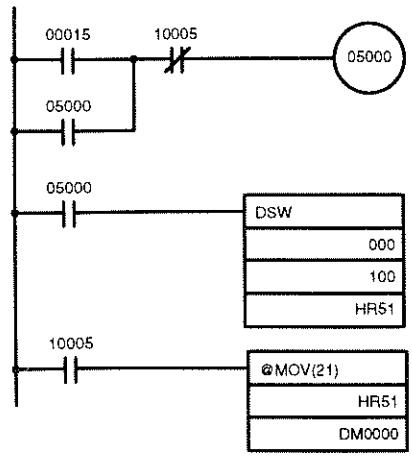
如果连接数字开关的输入字由 IW 指定，输出字由 OW 来指定，那末执行程序时，操作将按如下所示进行。



16 cy 完成一周需要 16 个循环周期

应用例子

这例子给出了一个从数字开关读出 8 位 BCD 码的程序，假设数字开关与 IR000(输入)和 IR100(输出)相连。



当 IR00015 变成 ON，IR05000 一直保持为 ON，直到由于 DSW(－)完成一周的读操作，一周标志(IR10005)变成 ON。

用 DSW(－)把从数字开关上设置的数据存到 HR51。

当读出数据完成后，一周标志(IR 10005)转换成 ON，储存在 HR51 的数字传送到 DM0000。

标志

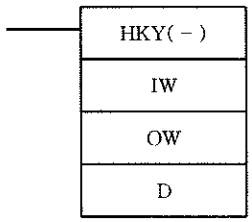
ER：间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码，或者超出 DM 区域)

R 和 R＋1 不在同一个数据区。

25410：当执行 DSW(－)时 25410 为 ON。

5－28－3 十六进制键盘输入指令－HKY(－)

梯形符号



操作数数据区域

IW:输入字
IR, SR, AR, DM, HR, LR

OW:控制信号输出字
IR, SR, AR, DM, HR, LR

D:第一个寄存器字
IR, SR, AR, DM, HR, LR

限制

D 和 D＋2 必须在同一数据区。

概述

当指令执行条件为 OFF, HKY(－)指令不执行;当执行条件为 ON, HKY(－)从十六进制键盘输入数据,这键盘连到由 IW 指定的输入上。

数据输入采用两种方式:

1. 在 D 和 D＋1 中生成一个 8 位移位寄存器。当十六进制键盘某键按下,对应的十六进制数移入 D 的最低位。D 和 D＋1 的其它位逐位左移,D＋1 的最高位溢出丢失。
2. D＋2 的位和 OW 的位 4 指示键输入,当键盘上(0～F)某一键按下,D＋2 中的对应位(00～15)和 OW 位 4 变为 ON。

注 1. 当键盘上某一键按下,键盘上的其它键输入无效。

2. 没用于键盘的其它输入输出位仍可用为一般输入输出使用。

用这条指令读入一个输入键要 4 到 13 个扫描周期。因为只有在测试这些键确实是接通了,输出才为 ON,所以需要不止一个扫描周期。

十六进制键盘输入装置必须连成一个 4×4 矩阵。

注意事项

为保证有效操作,每次 HKY(－)指令执行都必须完成对它使用的所有 I/O 点刷新。每当 HKY(－)用于子程序中,为保证每次指令执行刷新所有的 I/O 点,I/O 刷新指令必须和 HKY(－)一起使用。参见 330 页这类程序的举例。

每当程序开始执行,包括电源中断后重新起动,HKY(－)指令从第一个扫描周期执行。

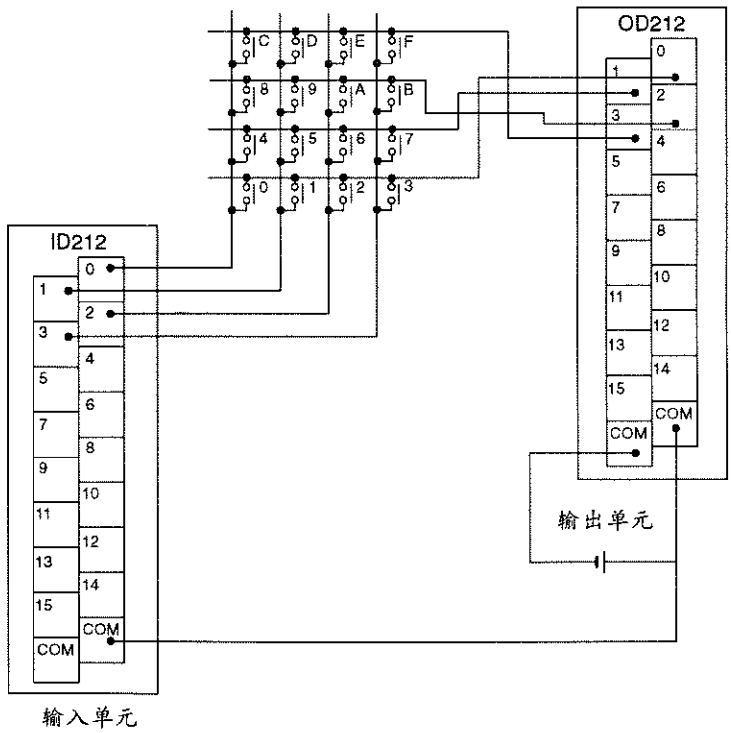
程序中不能重复使用 HKY(－)指令。

HKY(－)指令不能用于安装在从站机架上的 I/O 单元。

硬件

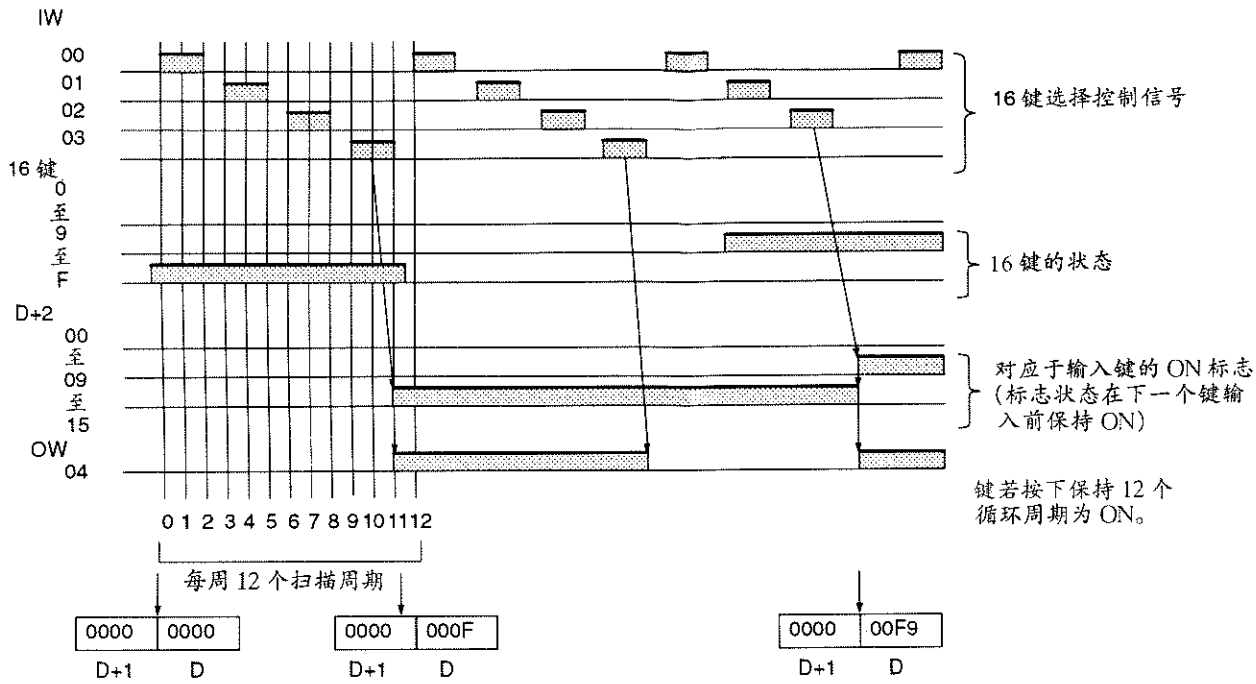
这条指令从十六进制键盘输入 8 位十六进制数。它采用 5 个输出位和 4 个输入位。准备一个十六进制键盘,象下图所示那样,将 0 到 F 数字开关键连接到 0～3 输入点和 0～3 输出点上。当有一个键被按下了输出点 4 变为 ON,除非应用中有需要,不必连接这一输出点。

连接到输入端的输入单元必须用 8 点或 8 点以上的直流输入单元,连接到输出端的输出单元必须用 8 点或 8 点以上的晶体管输出单元。



指令使用

如果连接十六进制键盘的输入字由 IW 定义，输出字由 OW 定义，当程序执行时，运行操作按如下所示进行。



标志

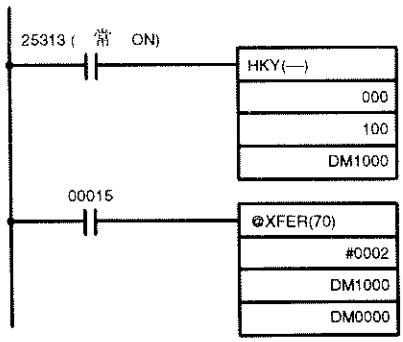
ER：间接寻址 DM 字不存在。（＊DM 字内容不是 BCD 码或者超出 DM 区域）

D 和 D＋2 不在同一数据区。

SR25408：当 HKY(－)正在执行时为 ON。

例

这个例子给出了从十六进制键盘输入数字的程序。假设十六进制键盘已和 IR000(输入)和 IR100(输出)相连。

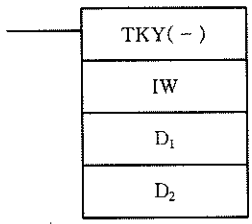


输入到 IR000 的十六进制键信息由 HKY(－)转换成十六进制数并存入字 DM1000 和 DM1001 中。

IR00015作为 ENTER 键使用，当 IR00015 为 ON 时，存贮在 DM1000 和 DM1001 中的数传送到 DM0000 和 DM0001 中。

5－28－4 十进制键盘输入指令－TKY(－)

梯形符号



操作数数据区域

IW：输入字
IR, SR, AR, DM, HR, LR

D_1：第一个寄存器字
IR, SR, AR, DM, HR, LR

D_2：键输入字
IR, SR, AR, DM, HR, LR

限制

D1 和 D1＋1 必须在同一数据区。

概述

当指令执行条件为 OFF，TKY(－)指令不执行；当执行条件为 ON，TKY(－)从十进制键盘输入数据，这键盘连到由 IW 指定的输入上。

TKY(－)指令可以用在程序中不同地方，只需改变它的输入字，IW。

数据输入采用两种方式：

1. 在 D1 和 D1＋1 中生成一个 8 位移位寄存器。当十进制键盘上某键按下对应十进制移入 D1 最低位。D1 和 D1＋1 的其它位逐位左移，D1＋1 的最高位溢出丢失。
2. D2 的第一个 10 位指示键的输入。当键盘上(0～9)某一键按下，D2 的对应位(00～09)变为 ON。

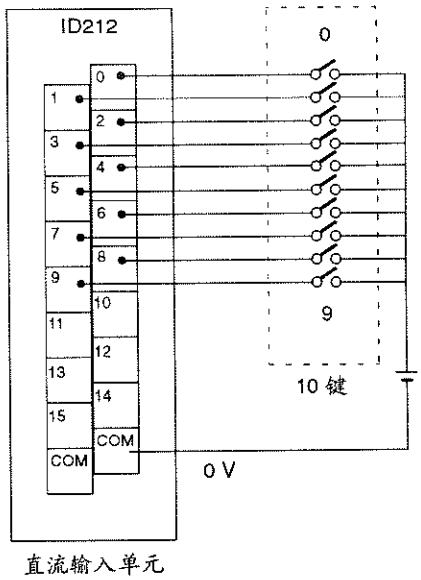
注 1. 当键盘上某一键按下，键盘的其它键输入无效。

2. 如果输入超过 8 位数，最左边位开始的数将被删除。

3. 指令中没使用的输入点仍可用作一般输入点来用。

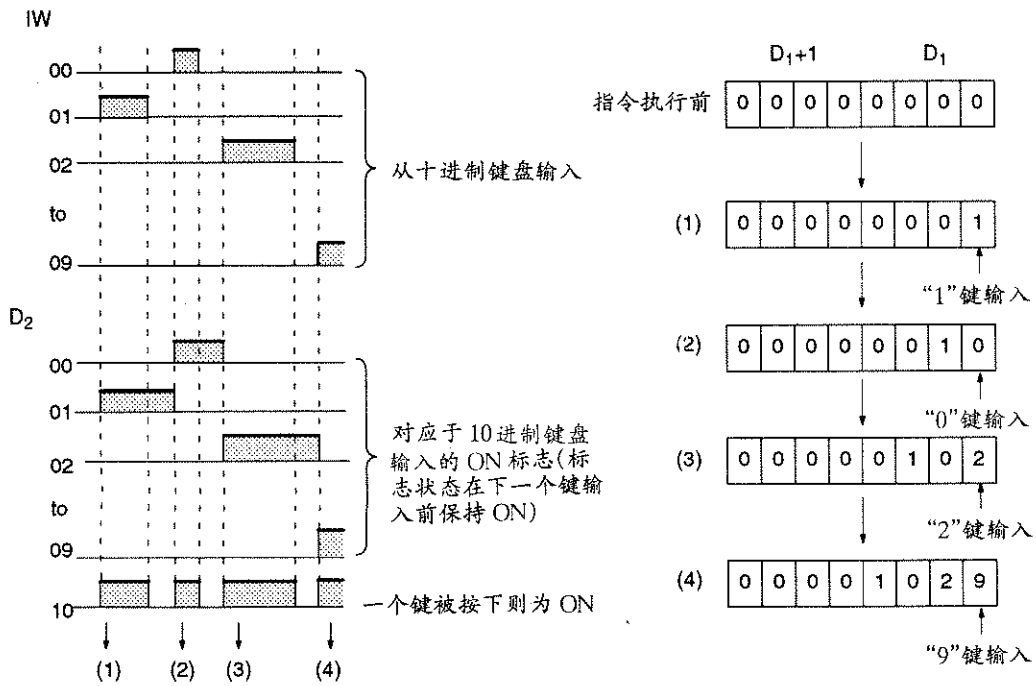
硬件

这条指令用十个输入点从十进制键盘输入 8 位 BCD 码数。配备一个十进制键，并如下图连接，把数值键 0～9 连到 0～9 输入点上。使用 16 点或更多点的直流输入单元。



指令使用

如果由 IW 来指定联接十进制键的输入字，当程序执行时，操作处理如下所示。



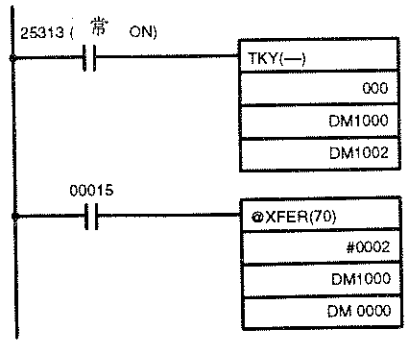
标志

ER：间接寻址 DM 字不存在。（＊DM 字的内容不是 BCD 码或超出 DM 区域）

D1 和 D1＋1 不在同一数据区。

例

例子给出了从十进制键盘中输入数字的程序。假设十进制键盘与 IR000 相连。

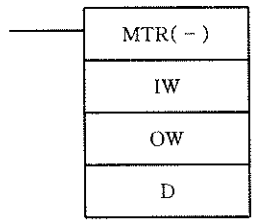


由 TKY(－)输入到 IR000 的十进制键盘信息转换成 BCD 码并存入 DM1000 和 DM1001，键信息存入 DM1002。

IR00015 用作“ENTER 键”，当 IR00015 变为 ON，存在 DM1000 和 DM1001 的数据将传送到 DM0000 和 DM0001。

5－28－5 矩阵输入指令－MTR(－)

梯形符号



操作数数据区域

IW：输入字
IR，SR，AR，DM，HR，LR

OW：输出字
IR，SR，AR，DM，HR，LR

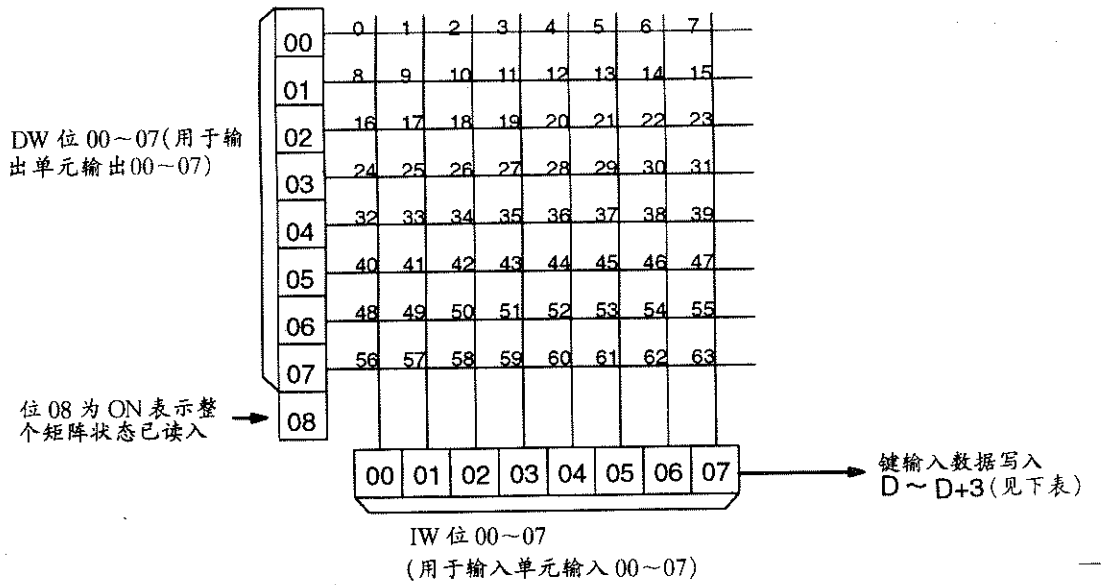
D：第一个目标字
IR，SR，AR，DM，HR，LR

限制

D 和 D＋3 必须在同一数据区。

概述

当指令执行条件为 OFF，MTR(－)指令不执行；当执行条件为 ON，MTR(－)从 8×8 矩阵中输入数据并将数据记录在 D～D＋3。即使联接的键少于 64 个，矩阵中的所有 64 点数据都被记录下来。



选择信号连续 3 个扫描周期输出到 OW 的 00～07 位(bit)。在任一时刻，只有一位输出为 ON。OW 的 08 位在 07 位后 3 个扫描周期变为 ON，表示读距阵循环结束。

当 64 键中某一键按下，输入位中的一位接受到输入信号。由输出信号的输出和接受到的输入信号比较，以确定按下的键。

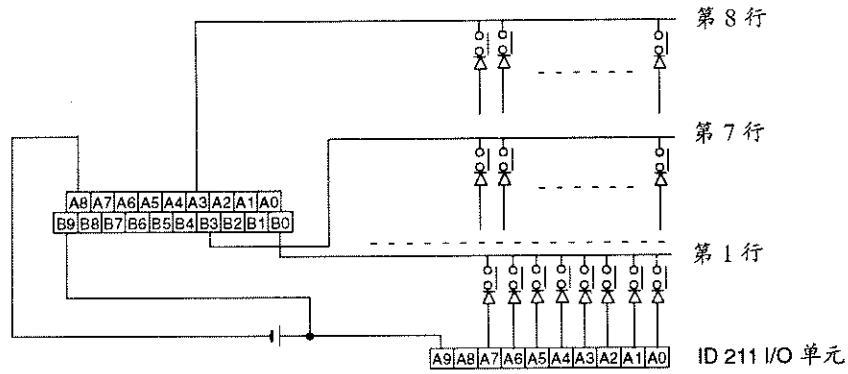
当一个键输入信号被检测到了，在 D～D＋3 中的对应位变为 ON。下表给出了键和 D～D＋3 中位之间的对应关系。

字	位	对 应 键
D	00～15	0～15
D＋1	00～15	16～31
D＋2	00～15	32～47
D＋3	00～15	48～63

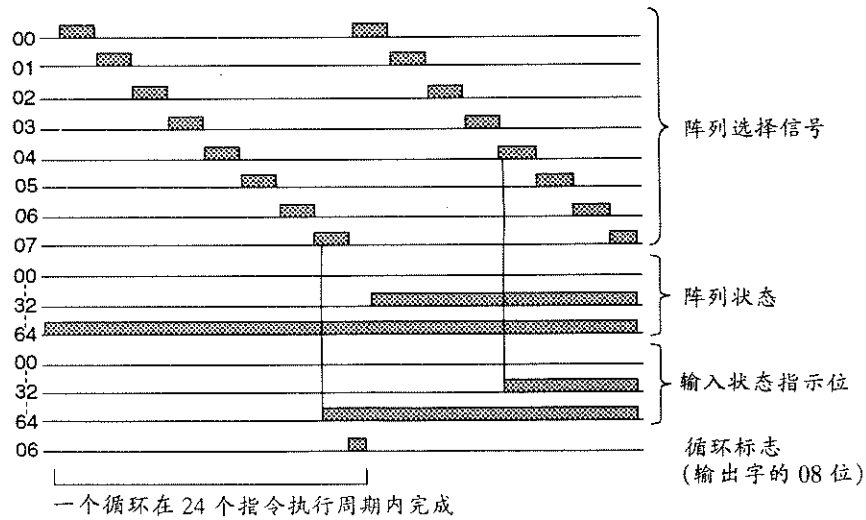
硬件

这条指令从一个 8 点输入和 8 点输出的 8×8 矩阵中输入多达 64 个信号。任何 8×8 矩阵都可使用。输入点必须连到 8 点或 8 点以上直流输入单元，输出点必须连到 8 点或 8 点以上的晶体管输出单元，下图给出了 MTR(—)指令的基本连线和时序。

连线



时序图



注意事项

64 键可分成 8 行连续扫描(包括 OW 字的 08 位的行)。因为每行扫描需 3 扫描周期,为了输入某个键,在键的给定行被扫描到之前,要产生多达 25 个扫描周期的延时。

为保证有效操作,每次 MTR(—)指令执行都必须对它所用的 I/O 点刷新。每当在子程序中使用 MTR(—)指令,为保证每次指令执行刷新所用的 I/O 点,输入/输出刷新指令必须和 MTR(—)指令一起使用。

只要程序开始执行,包括电源中断后重新起动,MTR(—)指令从第一个扫描周期执行。

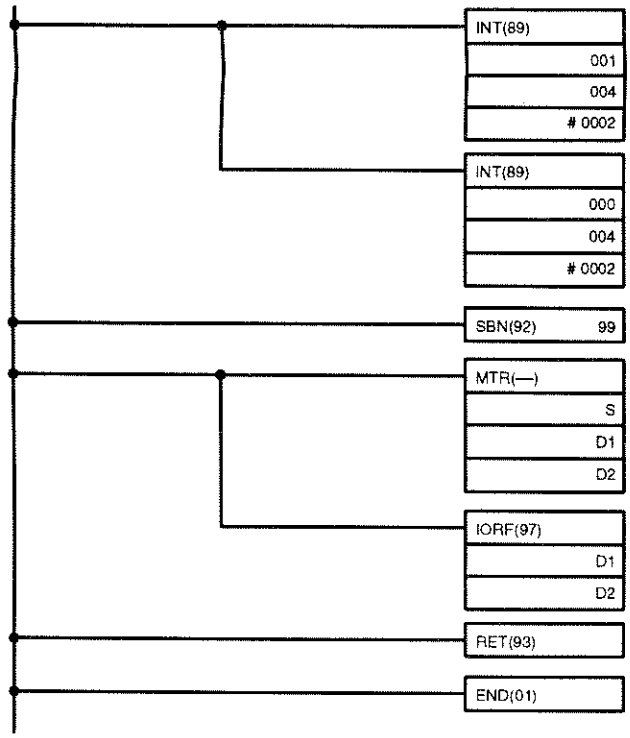
当 MTR(—)执行期间,SR25403 变为 ON,在联锁程序段中复位。MTR(—)指令在互锁程序部分中不执行。

程序中不能重复使用 MTR(—)指令。

MTR(—)指令不能用于安装在从站机架上的 I/O 单元。

例

下例给出了 MTR(—)指令在定时子程序中的编程。这里用到的 IORF(97)指令可保证每次 MTR(—)执行时能刷新指令所用到的 I/O 字。



标志

ER:间接寻址 DM 字不存在。(＊DM 字内容不是 BCD 码或超出 DM 区域)

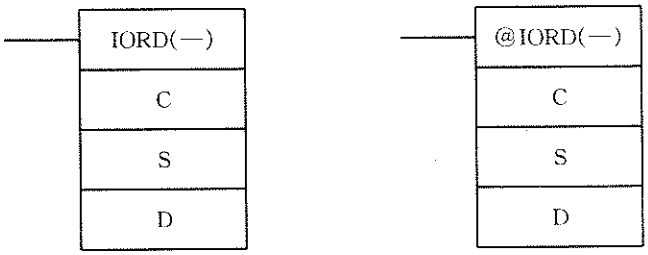
SR25403:MTR(—)指令执行期间为 ON。

5－29 特殊 I/O 单元指令

特殊 I/O 指令用于对指定的特殊 I/O 单元的存储器传送或取出数据。

5－29－1 特殊 I/O 单元读指令— IORD(－)

梯形符号



操作数数据区域

C:控制代码
IR, SR, AR, DM, HR, TC, LR, #
S:源信息
IR, SR, AR, DM, HR, TC, LR, #
D:第一目标字
IR, SR, AR, DM, HR, TC, LR

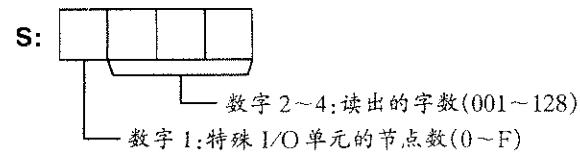
限制

只可指定安装在 PC 的 CPU 机架或扩展 I/O 机架上的特殊 I/O 单元。

S 的最后三位数必须是从 001～128 的 BCD 码。

说明

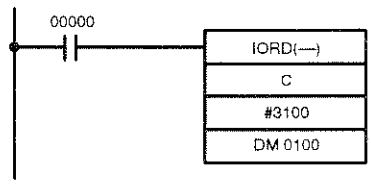
当执行条件为 OFF, IORD(—)指令不执行。当执行条件为 ON, IORD(—)将指定特殊 I/O 单元存储区的数据传送到由 D 开始的字中。源信息提供特殊 I/O 单元的节点号和读出的字数,如下图所示。



控制代码(C)取决于被指定的特殊 I/O 单元。详见单元操作手册。

例

当 IR00000 从 OFF 变为 ON 时,下述指令将 3# 特殊 I/O 单元存储区中 100 字传送到 DM 0100～DM 0199。



地址	指令	操作数
00200	LD	00000
00201	@IORD(－)	
		C
		# 3100
		DM 0100

标志

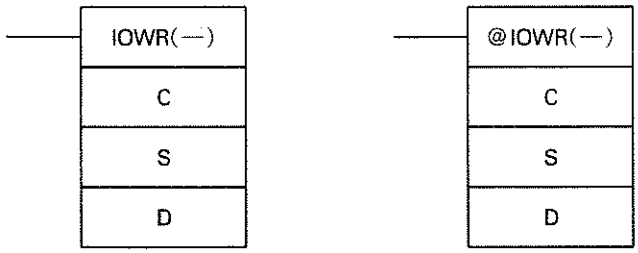
ER:S 最后三位数(指令字数)不是 BCD 码或不在 001～128 范围内。1 间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码或超出 DM 区域)

源单元的单元号不是 0～F 或源单元安装在从站机架上。接收数据溢出一个数据区域。

EQ:数据正常读入时为 ON,反之为 OFF。

5－29－2 特殊 I/O 单元写指令— IOWR(－)

梯形符号



操作数数据区域

C:控制代码
IR, SR, AR, DM, HR, TC, LR, #
S:第一源字
IR, SR, AR, DM, HR, TC, LR
D:目标信息
IR, SR, AR, DM, HR, TC, LR, #

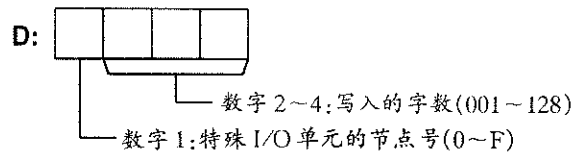
限制

只可指定安装在 PC 的 CPU 机架或扩展 I/O 机架上的特殊 I/O 单元。

D 中最后三位数必须是 001～128 的 BCD 码。

说明

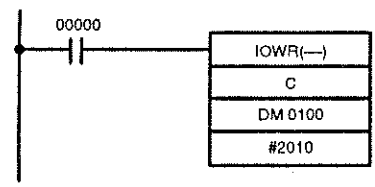
当执行条件为 OFF, IOWD(—)指令不执行。当执行条件为 ON, IOWR 将由 D 开始的字中内容传送到指定特殊 I/O 单元的存储区。目标信息提供特殊 I/O 单元的节点号和写入的字数，如下图所示。



控制代码(C)取决于指定的特殊 I/O 单元。详见单元操作手册。

例

当 IR00000 从 OFF 变为 ON, 下述指令将 DM 0100～DM0109 间 10 个字的内容写到 $2^{\#}$ 特殊 I/O 单元的存储区。



地址	指令	操作数
00200	LD	00000
00201	@IOWD(－)	
		C
		DM 0100
		# 2010

标志

ER:D 中最后三位数(指定字数)不是 BCD 码或不在 001～128 范围之内。

间接寻址 DM 字不存在。(＊DM 字的内容不是 BCD 码或超出 DM 区域)

目标单元的单元号不是 0～F 或目标单元安装在从站机架上。指令没有正常完成。

EQ:当数据正常读入时为 ON, 反之为 OFF。

第六章　程序执行时间

当编写程序和调试程序时，应考虑各种操作的时序。为了在正确的时间里取得所希望的控制作用，执行程序和 CPU 操作所需的时间是非常重要的，该时间就是为了完成所需控制作用进出 PC 的时间。本节说明扫描周期以及如何计算扫描周期和 I/O 响应时间。

6－1　扫描周期 …… 334
6－2　计算扫描周期 …… 338
6－2－1　仅有 I/O 单元的 PC …… 338
6－2－2　具有上位机链接单元和远程 I/O 主站单元的 PC …… 339
6－3　指令执行时间 …… 340
6－4　I/O 响应时间 …… 349
6－4－1　基本系统 …… 349
6－4－2　远程 I/O 系统 …… 350
6－4－3　上位机链接系统 …… 352
6－4－4　PC 链接系统 …… 353
6－4－5　一对一链接 I/O 响应时间 …… 355
6－4－6　中断响应时间 …… 357

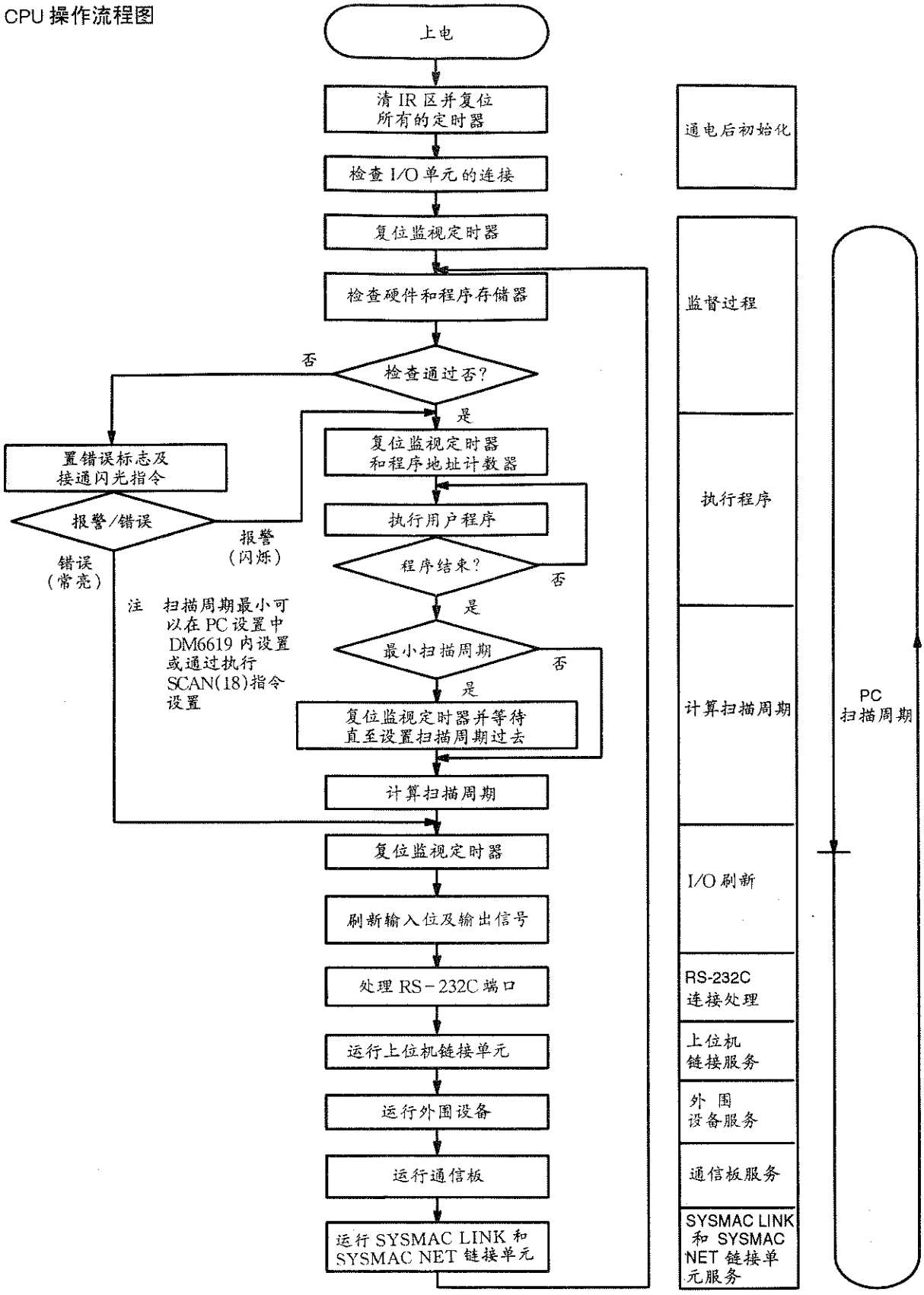
6－1 扫描周期

为了便于 PC 操作，平均、最大、最小扫描周期能够显示在编程器上或任何其它编程装置上，且最大扫描周期和当前扫描周期值保存在 AR26 和 AR27 中。然而，为有效地编程和进行 PC 操作，搞清一个扫描期内的各种操作以及影响扫描周期的各种因素是很重要的。

决定程序时间的主要因素是扫描周期和 I/O 响应时间。CPU 运行一周称之为一个扫描。运行一周所需要的时间称扫描周期。

CPU 操作的总流程示于以下流程图。

CPU 操作流程图



上电后，只要 PC 一通电，就立即执行最初的三个操作。剩下的各个操作以扫描的形式执行。

扫描周期也就是 CPU 完成其中一个扫描所要求的时间，这个扫描包括九项基本操作。

1．监督检查
2．执行程序
3．计算扫描周期
4．I/O 刷新
5．上位机链接单元服务
6．RS－232C 端口服务
7．外围设备服务
8．通信板服务
9．SYSMAC NET/SYSMAC LINK 服务

扫描周期是 PC 执行这九项操作所要求的总时间。第三项操作(扫描周期计算)所需时间很小，在实际计算中可以忽略不计。

操　作	所需时间	功　能
1.监督	0.7ms(对 C200HE－CPU11－E，2.1ms)	复位监视定时器，检查 I/O 总线，程序储存器；时钟刷新。
2.程序执行	执行所有指令的总时间随程序的大小使用的指令和执行条件而变化，详见 6－3 指令执行时间。	执行程序
3.计算扫描周期	忽略。如果设置了最小扫描周期，会产生一段等待时间，使循环时间达到最小设定值。	计算扫描周期，当执行 SCAN(18)指令，则等待直到达到设置的时间，并复位监视定时器。
4.I/O 刷新	下述时间的总和 每个字节输入(8 点)　20μs。 每个字节输出(8 点)　20μs。 (12 点输出单元按 16 点计算) PC 链接单元 I/O 刷新时间。 特殊 I/O 单元刷新时间。 远程 I/O 主单元每个 1.1ms＋从站机架上每个 I/O 字　0.17ms。 组－2(高密度)I/O 单元刷新时间。 PC 链接单元，特殊 I/O 单元和组－2 高密度 I/O 单元刷新时间详见下面列表。	输入位根据输入信号状态设置。输出按照存贮器中输出位的状态发送信号。 远程 I/O 系统中输入和输出刷新。运行特殊 I/O 单元。 运行组－2 高密度 I/O 单元。
5.上位机链接单元服务	每个单元最多 6ms。	处理来自连接机架安装上位机链接单元的计算机的命令。
6.RS－232C 端口服务	不连接设备，为 0ms。 最小 0.26ms 或者 TX0.05，其中 T 为第三项计算的扫描周期	处理连接 RS－232C 端口设备的通信。
7.外围设备服务	不连接设备，为 0ms。 最小 0.26ms 或者 TX0.05，其中 T 为第三项计算的扫描周期	处理编程设备来命令(计算机，编程器等)。
8.通信板服务	0.5ms＋每只端口处理时间。 每个端口处理时间如下： 最小 0.26ms 或者 TX0.05，其中 T 为第三项计算的扫描周期。	处理通信板来命令(RS－232C，RS－422 或 RS－485)。
9.SYSMAC LINK/SYSMAC NET LINK 服务	不安装通信单元，为 0ms。 对于 C200HS－SLK□□或 C200HS－SNT□□： 0.8ms＋每个单元最多 15ms。 对于 C200HW－SLK□□： 每个单元最多 3.5ms	处理连到 SYSMAC MET LINK/SYSMAC LINK 的计算机和其它设备来的命令。

PC 链接单元刷新

刷新的 I/O 部件	所需时间(ms)
512	7.4
256	4.1
128	2.7
64	1.7

特殊 I/O 单元刷新

单　元	每个单元所需时间
C200H－ID501/215	0.6ms
C200H－OD501/215	0.6ms　设为 32 点 I/O 时
C200H－MD501/215	1.6ms　设为动态 I/O 时
C200H－CT001－V1/CT002	2.0ms
C200H－NC111/NC112	2.1ms
C200H－NC211	5.0ms
C200H－AD001	1.1ms
C200H－DA001	0.9ms
C200H－TS001/TS101	1.2ms
C200H－ASC02	1.9ms　一般,对微分形式 5.0ms
C200H－IDSO1－V1/IDS21	2.0ms　一般,对命令传送 5.5ms
C200H－OVOO1	3.3ms
C200H－FZ001	2.0ms
C200H－TC□□□	2.7ms
C200H－CP114	2.0ms
C200H－AD002	1.4ms
C200H－LS101	2.3ms
C200H－PID□□□	2.7ms
C200H－DA002	1.0ms
C200HW－SRM21	0.44ms　当从站数设为最多 16 个 0.88ms　当从站数设为最多 32 个
C200HW－DRM21	1.72ms＋0.022ms×字数

组－2 高密度 I/O 单元刷新

单　元	所需时间
C200H－ID216	0.18ms
C200H－OD218	0.14ms
C200H－ID217	0.31ms
C200H－OD219	0.23ms

NT 链接

如果 PC 通过 C200HX/HG/HE 接口单元连接可编程终端(PT)，下表列出刷新 PT 的 I/O 所需的时间。

PT 输入表数目	I/O 刷新时间
最小设定：字符串表：0 数字表：0	2.5ms
最大设定：字符串表：32 数字表：128	5.4ms

监视定时器和长扫描周期

在 PC 内，监视定时器检测扫描周期并和设定值比较。如果扫描周期超过监视定时器的设定值，产生 FALS9F 的错误并且 CPU 停止工作。WDT(94)能够用于延长监视定时器的设定值。

即使扫描周期不超过监视定时器的设定值，如下表所示，长扫描周期对系统操作的精确度不利。

扫描周期 (ms)	可能不利的影响
10 或更长	当使用 TC016～TC511 时 TIMH(15)不精确(使用 TC000－TC015 不影响精度)
20 或更长	不能精确读 0.02 秒时钟脉冲(SR 25401)
100 或更长	不能精确读 0.1 秒时钟脉冲(SR 25500)，并且扫描周期错误标志(SR 25309)接通
200 或更长	不能精确地读 0.2 秒时钟脉冲(SR 25501)
6,500 或更长	不管监视定时器的设定如何，产生 FALS 码 9F，且系统停止。

在线编辑

在编程设备上进行在线编辑时，操作最多中断 80ms，重新输入时用户程序要屏蔽中断。在中断期间，将不产生长扫描周期的警告。在线编辑程序之前，核查它对 I/O 响应时间的影响。

当 AR25 中位 00～07 中含有口令代码“5A”时，不允许在线编辑。CPU 在禁止在线编辑位(AR 2509)为 ON 条件下将处于等待状态。CPU 处于等待状态时，在线编辑等候标志(AR 2510)为 ON。当 AR 2509 变为 OFF，处理执行(AR 2510 也变为 OFF)。

注意

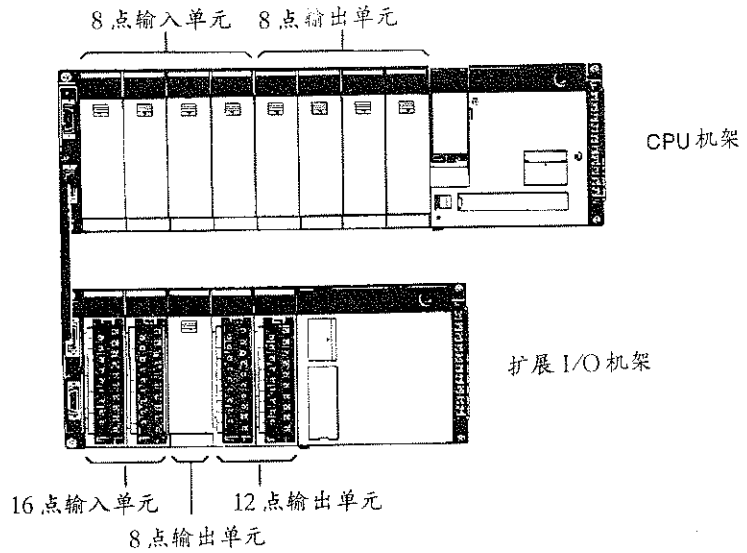
程序在线编辑会延长 I/O 响应时间，而在线编辑造成长扫描周期系统又不发出警告，因此在线编辑之前，必须确认 I/O 响应时间延缓不会给被控系统造成危险的后果。

6－2 计算扫描周期

当计算扫描周期时，必须考虑 PC 的结构、程序和程序执行的条件。也就是要考虑诸如 I/O 的点数、所使用的编程指令和是否使用外围设备装置等。本节给出了一些计算基本扫描周期的例子。为了简化例子，在程序中所采用的指令都假设是 LD 或是 OUT，指令的平均执行时间为 0.156μs(执行时间由 6－3“指令执行时间”中的表给出)。

6－2－1 仅有 I/O 单元的 PC

这里我们将计算简单 PC 的扫描周期，CPU 仅控制在 CPU 机架上的 8 个 I/O 单元和在 5 槽扩展 I/O 机架上的 5 个 I/O 单元，该 PC 结构显示如下。假设程序包含 5000 条指令，平均每条指令的执行时间需0.156μs。



计算

上述扫描周期的等式如下：

扫描周期＝监督时间＋程序执行时间＋I/O刷新时间＋外围设备服务时间

过　程	计　算	接外围设备	不接外围设备
监督	固定	0.7ms	0.7ms
执行程序	0.156μs/每条指令 ×5,000指令	0.78ms	0.78ms
I/O刷新	见下面	0.34ms	0.34ms
外围设备服务	最小时间	0.26ms	0.0ms
扫描周期	上述之和	2.08ms	1.82ms

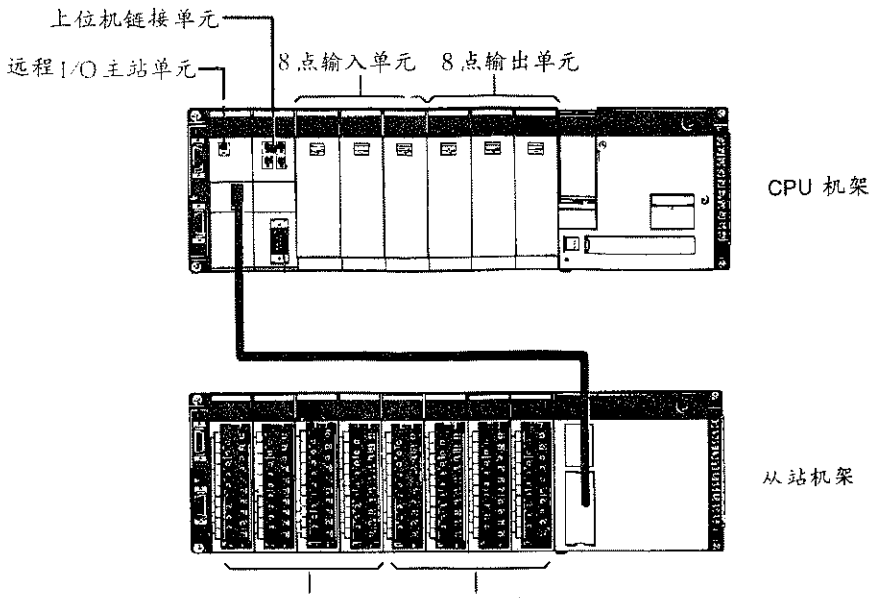
PC控制二个16点输入单元，四个8点输入单元，二个12点输出单元(12点单元作16点处理)和五个8点输出单元的I/O刷新时间如下所述：

$$\frac{(16\text{点}\times 2)+(8\text{点}\times 4)}{8\text{点}}\times 20\mu\text{s}+\frac{(16\text{点}\times 2)+(8\text{点}\times 5)}{8\text{点}}\times 20\mu\text{s}=0.34\text{ms}$$

6－2－2　具有上位机链接单元和远程I/O主站单元的PC

这里，我们将计算具有上位机链接单元和远程I/O主站单元的PC的扫描周期，该PC的结构显示如下，CPU控制3个8点输入单元，3个8点输出单元，一个上位机链接单元和一个接远程I/O从站的远程I/O主站单元，该远程I/O从站单元包括4个16点输入单元和4个12点输出单元。

我们假设程序有5000条指令，且每条指令平均执行时间为0.156μs，以及在RS－232C端口不连任何，且不安装SYSMAC NET和SYSMAC LINK单元。



计算

扫描周期的算式如下：

扫描周期＝监督时间＋程序执行时间＋I/O刷新时间＋上位机链接单元服务时间＋外围设备服务时间。

过　程	计　算	接外围设置	不接外围设备
监督	固定	0.7ms	0.7ms
执行程序	0.156μs/每条指令 ×5,000 指令	0.78ms	0.78ms
I/O刷新	见下面	2.58ms	2.58ms
上位机链接服务	固定	6.0ms	6.0ms
外围设备服务	0.7＋0.78＋2.58＋6＝10.06 10.06×0.05＝0.50	0.50ms	0.0ms
扫描周期	上述之和	10.56ms	10.06ms

PC控制CPU机架上三个8点输入单元，三个8点输出单元和从站架上八个单元的I/O刷新时间如下：

$$\frac{(8\text{点}\times 3)+(8\text{点}\times 3)}{8\text{点}}\times 20\mu s+1.1\text{ms}+8\text{单元}\times 0.17\text{ms}=2.58\text{ms}$$

6－3　指令执行时间

下表列出了可用于C200HX/HG/HE的所有指令的执行时间，给出了最大和最小的执行时间及造成最大最小执行时间的相关条件，当条件一栏里涉及到“字”时，表示除间接寻址DM字外任何字的内容。间接寻址DM地址由“＊DM”表示，当使用它时，会产生一个较长的执行时间。

对大多数指令而言，执行时间取决于它们具有接通还是断开的执行条件。梯形图OUT和OUT NOT指令是个例外，不管执行条件如何，OUT和OUT NOT需要相同的时间。对一条指令而言，OFF的执行时间也会根据情况发生变化。例如，它是否在一个联锁程序段内和IL的执行条件是否为OFF，它是否在JMP(04) 00和JME(05) 00之间以及JMP(04) 00的执行条件是否为OFF。或者它是否被一个OFF执行条件复位。“R”“IL”和“JMP”用于表示这三个时间。

除非另外注明，所有的执行时间都以微秒为单位给出。

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
LD	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.312	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
LD NOT	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.313	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
AND	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.312	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
AND NOT	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.313	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
OR	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.313	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
OR NOT	对 IR 和 SR 23600～SR 25515	0.104	0.156	0.313	0.104	0.156	0.313
	对 SR 25600～SR 51115	0.208	0.313	0.626	0.104	0.156	0.313
AND LD	…	0.104	0.156	0.313	0.104	0.156	0.313
OR LD	…	0.104	0.156	0.313	0.104	0.156	0.313
OUT	对 IR 和 SR 23600～SR 25515	0.208	0.313	0.626	0.208	0.313	0.626
	对 SR 25600～SR51115	0.313	0.468	0.936	0.208	0.313	0.626
OUT NOT	对 IR 和 SR 23600～SR25515	0.208	0.313	0.626	0.208	0.313	0.626
	对 SR 25600～SR 51115	0.313	0.468	0.936	0.208	0.313	0.626
TIM	SV 是常数	0.417	0.625	1.25	0.417	0.625	1.25
	SV 是＊DM				22.45	22.45	37.15
	指定字 256～511				0.417	0.625	1.25
CNT	SV 是常数	0.417	0.625	1.25	0.417	0.625	1.25
	SV 是＊DM	0.417	0.625	1.25	22.55	22.55	37.25
	指定字 256～511	0.417	0.625	1.25	0.417	0.625	1.25
SET	对 IR 和 SR 23600～SR 25515	0.208	0.313	0.626	0.208	0.313	0.626
	对 SR 25600～SR 51115	0.313	0.468	0.936	0.208	0.313	0.626
RSET	对 IR 和 SR 23600～SR 25515	0.208	0.313	0.626	0.208	0.313	0.626
	对 SR 25600～SR 51115	0.313	0.468	0.936	0.208	0.313	0.626
NOP(00)	…	0.104	0.156	0.312	0.313	0.469	0.938
END(01)	…	24.75		39.45	0.313	0.469	0.938
IL(02)	…	7.55		22.25	0.313	0.469	0.938
ILC(03)	…	9.25		23.95	0.313	0.469	0.938

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
JMP(04)	…	7.65		22.35	0.313	0.469	0.938
JME(05)	…	7.95		22.65	0.313	0.469	0.938
FAL(06)	FAL 号 01～99	88.6		88.6	0.313	0.469	0.938
	FAL 号 00	86.6		86.8	0.313	0.469	0.938
FALS(07)	…	…	见注 1	见注 2	0.313	0.469	0.938
STEP(08)	…	33.1		47.8	0.313	0.469	0.938
		15.3		30			
SNXT(09)	…	9.25		23.95	0.313	0.469	0.938
SFT(10)	1 字移位寄存器	21.05		35.75	R:8.05 IL:8.05 JMP:8.05		
	100 字移位寄存器	117.8		132.5	R:8.05 IL:8.05 JMP:8.05		
	250 字移位寄存器	262.5		277.2	R:8.05 IL:8.05 JMP:8.05		
KEEP(11)	对 IR 和 SR 23600～SR 25515	0.208	0.313	0.625	0.208	0.319	0.625
	对 SR 25600～SR 51115	0.318	0.469	0.938			
CNTR(12)	SV 是常数	19.15		33.85	R:16.95 IL:11.65 JMP:11.65		
	SV 是＊DM	31.45		46.15	R:16.95 IL:11.65 JMP:11.65		
DIFU(13)	…	13.75		28.45	Normal:13.75 IL:13.65 JMP:11.95		
DIFD(14)	…	13.65		28.35	Normal:13.65 IL:13.55 JMP:11.85		
TIMH(15)	SV 是中断常数	18.35		33.05	R:25.05 IL:24.05 JMP:14.45		
	正常扫描	16.55		31.25	R:21.95 IL:21.05 JMP:11.25		
	SV 是中断＊DM	18.35		33.05	R:37.1 IL:36.5 JMP:14.45		
	正常扫描	16.55		31.25	R:34.1 IL:33.3 JMP:11.25		
WSFT(16)	移位 1 个字	16.45		31.15	.313	0.469	0.938
	用＊DM 移位 6,144 个字	6.45ms		见注 2			
CMP(20)	一个字和常数比较	0.417	0.625	1.25	0.313	0.469	0.938
	二个字比较	0.521	0.781	1.56			
	两个＊DM 字比较	35.2		49.9			
MOV(21)	一个常数传送给一个字	0.417	0.625	1.25	0.313	0.469	0.938
	比较两个字	0.625	0.937	1.87			
	＊DM 传送给＊DM	33.7		48.4			
MVN(22)	一个常数取反后传送给一个字	0.417	0.625	1.25	0.313	0.469	0.938
	比较两个字	0.625	0.937	1.87			
	＊DM 传送给＊DM	34.3		49			
BIN(23)	一个字转换到另一个字	19.65		34.35	0.313	0.469	0.938
	＊DM 转换到＊DM	40.5		55.2			
BCD(24)	一个字转换到另一个字	18.25		32.95	0.313	0.469	0.938
	＊DM 转换到＊DM	39.1		53.8			
ASL(25)	移位一个字	12.25		26.95	0.313	0.469	0.938
	移位＊DM	23.35		38.05			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
ASR(26)	移位一个字	11.95		26.65	0.313	0.469	0.938
	移位＊DM	22.95		37.65			
ROL(27)	循环一个字	13.15		27.85	0.313	0.469	0.938
	循环＊DM	24.25		38.95			
ROR(28)	循环一个字	13.15		27.85	0.313	0.469	0.938
	循环＊DM	24.25		38.95			
COM(29)	一个字取反	11.45		26.15	0.313	0.469	0.938
	＊DM 字取反	22.65		37.35			
ADD(30)	常数＋字→字	16.65		31.35	0.313	0.469	0.938
	字＋字→字	18.45		33.15			
	＊DM＋＊DM→＊DM	50.1		64.8			
ADD(31)	常数－字→字	16.65		31.35	0.313	0.469	0.938
	字－字→字	18.45		33.15			
	＊DM－＊DM→＊DM	50.1		64.8			
MUL(32)	常数×字→字	31.15		45.85	0.313	0.469	0.938
	字×字→字	32.95		47.65			
	＊DM×＊DM→＊DM	64.7		79.4			
DIV(33)	字÷常数→字	30.15		44.85	0.313	0.469	0.938
	字÷字→字	32.35		47.05			
	＊DM÷＊DM→＊DM	64.1		78.8			
ANDW(34)	常数与字→字	14.35		29.05	0.313	0.469	0.938
	字与字→字	15.25		29.95			
	＊DM 与＊DM→＊DM	46.7		61.4			
ORW(35)	常数或字→字	14.35		29.05	0.313	0.469	0.938
	字或字→字	15.25		29.95			
	＊DM 或＊DM→＊DM	46.7		61.4			
XORW(36)	常数异或字→字	14.35		29.05	0.313	0.469	0.938
	字异或字→字	15.25		29.95			
	＊DM 异或＊DM→＊DM	46.7		61.4			
XNRW(37)	常数异或非字→字	14.55		29.25	0.313	0.469	0.938
	字异或非字→字	15.45		30.15			
	＊DM 异或非＊DM→＊DM	46.9		61.6			
INC(38)	一个字递增	11.55		26.25	0.313	0.469	0.938
	＊DM 字递增	22.45		37.15			
DEC(39)	一个字递减	11.45		26.15	0.313	0.469	0.938
	＊DM 字递减	22.35		37.05			
STC(40)	…	7.22		21.92	0.313	0.469	0.938
CLC(41)	…	7.22		21.92	0.313	0.469	0.938
TRSM(45)	…	18.65		33.35	0.313	0.469	0.938
MSG(46)	指字为 DM	12.05		26.75	0.313	0.469	0.938
	指字为＊DM	21.95		36.65			
ADB(50)	常数＋字→字	19.15		33.85	0.313	0.469	0.938
	字＋字→字	20.05		34.75			
	＊DM＋＊DM→＊DM	51.7		66.4			
SBB(51)	常数－字→字	18.95		33.65	0.313	0.469	0.938
	字－字→字	19.85		34.55			
	＊DM－＊DM→＊DM	51.7		66.4			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
MLB(52)	常数×字→字	16.95		31.65	0.313	0.469	0.938
	字×字→字	17.85		32.55			
	*DM×*DM→*DM	49.3		64			
DVB(53)	字÷常数→字	17.15		31.85	0.313	0.469	0.938
	字÷字→字	18.05		32.75			
	*DM÷*DM→*DM	49.7		64.4			
ADDL(54)	字+字→字	22.45		37.15	0.313	0.469	0.938
	*DM+*DM→*DM	53.9		68.6			
SUBL(55)	字－字→字	22.45		37.15	0.313	0.469	0.938
	*DM－*DM→*DM	53.9		68.6			
MULL(56)	字×字→字	110.6		125.3	0.313	0.469	0.938
	*DM÷*DM→*DM	142.6		157.3			
DIVL(57)	字÷字→字	105.8		120.5	0.313	0.469	0.938
	*DM÷*DM→*DM	135.8		150.5			
BINL(58)	一些字转换到另一些字	35.15		49.85	0.313	0.469	0.938
	*DM 转换到*DM	55.9		70.6			
BCDL(59)	一些字转换到另一些字	25.75		40.45	0.313	0.469	0.938
	*DM 转换到*DM	46.5		61.2			
XFER(70)	传送一个字	45.3		44.2	0.313	0.469	0.938
	用*DM 传送 1,024 个字	655		653.9			
	用*DM 传送 6,144 个字	3.66ms		见注 3			
BSET(71)	将一个常数设给一个字	19.75		34.45	0.313	0.469	0.938
	用*DM 将*DM 设给 1,024 个字	40.9		55.6			
	用*DM 将*DM 设给 6,144 个字	52.3		67			
ROOT(72)	取字的平方根,存入一个字	41.7		56.4	0.313	0.469	0.938
	取*DM 字的平方根,存入*DM	85.5		100.2			
XCHG(73)	字与字之间交换	14.45		29.15	0.313	0.469	0.938
	*DM 字之间交换	36.45		51.15			
SLD(74)	移位 1 个字	15.65		30.35	0.313	0.469	0.938
	用*DM 移位 1,024 个 DM 字	2.72ms		见注 2			
	用*DM 移位 6,144 个 DM 字	16.2ms		见注 2			
SRD(75)	移位 1 个字	15.65		30.35	0.313	0.469	0.938
	用*DM 移位 1,024 个 DM 字	2.72ms		见注 2			
	用*DM 移位 6,144 个 DM 字	16.2ms		见注 2			
MLPX(76)	将字译码到字	47.3		62	0.313	0.469	0.938
	将*DM 字译码到*DM 字	103.8		118.5			
DMPX(77)	将字编码到字	28.45		43.15	0.313	0.469	0.938
	将*DM 字编辑到*DM 字	111.8		126.5			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
SDEC(78)	将字译码到字	26.95		41.65	.313	0.469	0.938
	将 2 个数 * DM 译码到 * DM	63.3		78			
	将 4 个数 * DM 译码到 * DM	71.7		86.4			
FDIV(79)	字÷字→字(等于 0)	62.3		77	0.313	0.469	0.938
	字÷字→字(不等于 0)	499		513.7			
	* DM÷ * DM→ * DM	843		857.7			
DIST(80)	常数→(字+(字))	27.65		42.35	0.313	0.469	0.938
	* DM→(* DM+(* DM))	61.5		76.2			
COLL(81)	(字+(字))→字	28.75		43.45	0.313	0.469	0.938
	(* DM+(* DM))→ * DM	64.3		79			
MOVB(82)	常数传送给字	17.35		32.05	0.313	0.469	0.938
	字传送给字	19.45		34.15			
	* DM 传送给 * DM	52.9		67.6			
MOVD(83)	常数传送给字	16.95		31.65	0.313	0.468	0.938
	字传送给字	18.25		32.95			
	* DM 传送给 * DM	54.3		69			
SFTR(84)	移位 1 个字	20.05		34.75	0.313	0.469	0.938
	用 * DM 移位 1,024 个 * DM 字	1.1ms		(见注 2)			
	用 * DM 移位 6,144 个 * DM 字	6.37ms		(见注 2)			
TCMP(85)	与指定表格中的字比较	37.25		51.95	0.313	0.469	0.938
	与指定表格中的字比较	38.1		52.8			
	将 * DM 与 * DM 指定的表格字比较	69.1		83.8			
ASC(86)	在字之间	30.1		44.8	0.313	0.469	0.938
	在 * DM 字之间	78.3		93			
SEND(90)	传送 1 个字	60.9		75.6	0.313	0.469	0.938
	传送 1,000 个字	99.2		113.9			
SBS(91)	…	22.1		36.8	0.313	0.469	0.938
SBN(92)	…	…		…	0.313	0.469	0.938
RET(93)	…	20.9		35.6	0.313	0.469	0.938
WDT(94)	…	10.55		25.25	0.313	0.469	0.938
IORF(97)	刷新 1 个字	110(输入),170(输出)		110(输入),170(输出)	0.313	0.469	0.938
	刷新 30 个字	2002		2000.9			
RECV(98)	刷新 1 个字	67.1		67.1	0.313	0.469	0.938
	刷新 1,000 个字	105.8		105.8			
MCRO(99)	指定—字参数	38.5		53.2	0.313	0.469	0.938
	指字— * DM 字参数	58.2		72.9			
ASFT(—)缺省值:(17)	复位 1 个字	24.95		39.65	0.313	0.469	0.938
	用 * DM 移位 1 个字	29.05		43.75			
	用 * DM 移位 10 个字	39.9		54.6			
SCAN(—)缺省值:(18)	乘数为常数	18.65		33.35	0.313	0.469	0.938
	乘数为 * DM	30.45		45.15			
MCMP(—)缺省值:(19)	比较 2 个字,产生字	60.3		75	0.313	0.469	0.938
	比较 2 个 * DM 字,产生 * DM 字	93		107.7			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
LMSG(—) 缺省值:(47)	SV 为字	17.95		32.65	0.313	0.469	0.938
	SV 为 * DM	27.65		42.35			
TERM(—) 缺省值:(48)	…	8.55		23.25	0.313	0.469	0.938
CMPL(—) 缺省值:(60)	字与字比较	16.55		31.25	0.313	0.469	0.938
	* DM 与 * DM 比较	38.5		53.2			
MPRF(—) 缺省值(61)	1 个单元	2.00		16.7	0.313	0.469	0.938
	10 个单元	13.00		27.7			
XFRB(—) 缺省值(62)	从字到字发送 1 位	22.45		37.15	0.313	0.469	0.938
	从字到字发送 256 位	142.6		157.3			
LINE(—) 缺省值(63)	从几个字传送到 1 个常数	59.7		74.4	0.313	0.469	0.938
	从几个字传送到 1 个字	62.1		76.8			
	从 * DM 传送到 * DM	182.5		197.2			
COLM(—) 缺省值:(64)	从 1 个常数传送到几个字	72.7		87.4	0.313	0.469	0.938
	从 1 个字传送到几个字	74.9		89.6			
	从 * DM 传送到 * DM	190.5		205.2			
SEC(—) 缺省值:(65)	DM→DM	35.35		50.05	0.313	0.469	0.938
	* DM→ * DM	56.3		71			
HMS(—) 缺省值:(66)	DM→DM	36.5		51.2	0.313	0.469	0.938
	* DM→ * DM	57.7		72.4			
BCNT(—) 缺省值:(67)	SV 为常数	39.1		53.8	0.313	0.469	0.938
	SV 为 * DM	26.5ms		见注 2			
BCMP(—) 缺省值:(68)	常数与字指定表比较	62.1		76.8	0.313	0.469	0.938
	与字比较后到另一个字	63.1		77.8			
	* DM 与 * DM 指定表比较	98.2		112.9			
APR(—) 缺省值:(69)	规定正弦函数	24.25		38.95	0.313	0.469	0.938
	SV 为 * DM	473		487.7			
TTIM(—) 缺省值(87)	设置常数	27.55		42.25	常数输入 OFF: 23.15	R:24.15 IL:20.55	JPM: 19.55
	设置 * DM	35.5		50.2	常数输入 OFF: 35.30	R:36.3 IL:32.7	JPM: 31.9
ZCP(—) 缺省值(88)	常数和一个字比较	16.75		31.45	0.313	0.469	0.938
	一个字和一个字比较	17.65		32.35			
	* DM 和一个 * DM 比较	49.7		64.4			
INT(—) 缺省值:(89)	SV 为字	19.90～198.50		见注 3	0.313	0.469	0.938
	SV 为 * DM	19.90～213.5		见注 3			
TKY(—)	输入到 DM	25.65		40.35	0.313	0.469	0.938
	输入到 * DM	46.7		61.4			
RXD(—)	指定一个字	31.5		30.4	0.313	0.469	0.938
	指定 * DM	73.5		72.4			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
TXD(—)	指定 1 个字	56.1		70.8	0.313	0.469	0.938
	指定 * DM	99.4		114.1			
7SEG(—)	字指定 4 个数	19～22		见注 2	0.313	0.469	0.938
	* DM 指定 4 个数	30～34		见注 2			
	字指定 8 个数	19～22		见注 2			
	* DM 指定 8 个数	30～34		见注 2			
FPD(—)	字指定,代码输出	74.60～89.40		见注 2	0.313	0.469	0.938
	* DM 指定,信息输出	105.0～142.2		见注 2			
SRCH(—)	SV 为常数	39.7		54.4	0.313	0.469	0.938
	SV 为 * DM	1.35ms		见注 2			
	SV 为 * DM	7.73ms		见注 2			
MAX(—)	搜索 DM	31.75		46.45	0.313	0.469	0.938
	搜索 * DM	1.31ms		见注 2			
MIN(—)	搜索 DM	31.75		46.45	0.313	0.469	0.938
	搜索 * DM	1.31ms		见注 2			
SUM(—)	DM 累加	26.55		41.25	0.313	0.469	0.938
	* DM 累加	1.30ms		见注 2			
FCS(—)	加 1 个字→字	26.75		41.45	0.313	0.469	0.938
	加 999 个字→ * DM	10.5ms		见注 2			
HEX(—)	DM 转换	36.95		51.65	0.313	0.469	0.938
	* DM 转换	102.6		117.3			
AVG(—)	1 个操作数的平均值	33.05		47.75	0.313	0.469	0.938
	64 个操作数的平均值	133.8		148.5			
PID(—)	指定 1 个字	48.1		62.8	0.313	0.469	0.938
	指定 * DM	89.4		104.1			
XDMR(—)	SV 为常数	39.9		54.6	0.313	0.469	0.938
	SV 为字	1.44ms		见注 2			
	SV 为 * DM	4.19ms		见注 2			
MTR(—)	输入 DM	29～34		见注 2	0.313	0.469	0.938
	输入 * DM	45～51		见注 2			
ADBL(—)	DM＋DM→DM	27.35		42.05	0.313	0.469	0.938
	* DM＋ * DM→ * DM	60.1		74.8			
SBBL(—)	DM－DM→DM	26.95		41.65	0.313	0.469	0.938
	* DM－ * DM→ * DM	59.7		74.4			

指令	条件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
MBS(—)	常数×字→字	20.85		35.55	0.313	0.469	0.938
	DM×DM→DM	21.65		36.35			
	*DM×*DM→*DM	53.5		68.2			
DBS(—)	常数÷字→字	21.55		36.25	0.313	0.469	0.938
	DM÷DM→DM	22.45		37.15			
	*DM÷*DM→*DM	54.9		69.6			
MBSL(—)	DM×DM→DM	45.3		60	0.313	0.469	0.938
	*DM×*DM→*DM	77.1		91.8			
DBSL(—)	DM÷DM→DM	86.6		101.3	0.313	0.469	0.938
	*DM÷*DM→*DM	118.6		133.3			
CPS(—)	比较二个常数	13.95		28.65	0.313	0.469	0.938
	DM 与 DM 比较	15.65		30.35			
	DM 与 *DM 比较	36.9		51.6			
CPSL(—)	比较二个 DM	18.65		33.35	0.313	0.469	0.938
	比较二个 *DM	40.7		55.4			
NEG(—)	常数转换到字	15.05		29.75	0.313	0.469	0.938
	一个字转换到一个字	16.85		31.55			
	*DM 转换到 *DM	37.1		51.8			
NEGL(—)	字转换到字	18.15		32.85	0.313	0.469	0.938
	*DM 转换到 *DM	39.7		54.4			
ZCPL(—)	二个字比较	19.95		34.65	0.313	0.469	0.938
	二个 *DM 比较	53.3		68			
SCL(—)	SV 为字	58.3		73	0.313	0.469	0.938
	SV 为 *DM	89.4		104.1			
HKY(—)	指定一个字	23.55		38.25	0.313	0.469	0.938
	指字 *DM	44.3		59			
DSW(—)	DM CS 输出	35		49.7	0.313	0.469	0.938
	DM RD 输出	35		49.7			
	DM 数据恢复	45		59.7			
	*DM CS 输出	44		58.7			
	*DM RD 输出	44		58.7			
	*DM 数据恢复	53		67.7			
BXFR(—)	指定字(1 个字)	71.5		70.4	0.313	0.469	0.938
	指定 *DM(1,024 个字)	3.68ms		见注 3			
	指定 *DM(6,144 个字)	8.75ms		见注 3			
EMBC(—)	指定常数	18.45		33.15	0.313	0.469	0.938
	指定字	19.95		34.65			
	指定 *DM	33.45		48.15			
XFR2(—)	指定字(1 个字)	46.4		45.3	0.313	0.469	0.938
	指定 *DM(1,024 个字)	656.2		655.1			
	指定 *DM(6,144 个字)	3.66ms		见注 3			
BXF2(—)	指定字(1 个字)	72.8		71.7	0.313	0.469	0.938
	指定 *DM(1,024 个字)	3.68ms		见注 3			
	指定 *DM(6,144 个字)	8.75ms		见注 3			

指　令	条　件	ON 执行时间(μs)			OFF 执行时间(μs)		
		C200HX	C200HG	C200HE	C200HX	C200HG	C200HE
IEMS(—)	指定常数(转换到 DM)	19.25		18.15	0.313	0.469	0.938
	指定字(转换到 EM 组)	24.95		23.85			
IORD(—)	…	…	见注 1	见注 2	0.313	0.469	0.938
IOWR(—)	…	…	见注 1	见注 2	0.313	0.469	0.938
STUP(—)	指定缺省 RS－232C 条件	30.9		29.8	0.313	0.469	0.938
	通信板端口 A＊DM 指定	61		59.9			
PMCR(—)	端口和序列号为常数输入和输出字为 DM	41		39.9	0.313	0.469	0.938
	端口和序列号为常数输入和输出字为＊DM	56		54.9			
	端口,序列号,输入和输出字均为＊DM	74.2		73.1			

注　1. 和 C200HX 指令执行时间相同。

2. C200HX 指令执行时间加 14.7μs。

3. C200HX 指令执行时间加 1.1μs。

6－4　I/O 响应时间

I/O 响应时间是在接受到一个输入信号以后,到 PC 输出一个控制信号所需的时间。当 CPU 接受对应于输入刷新周期的输入信号时,用于响应的时间取决扫描周期。

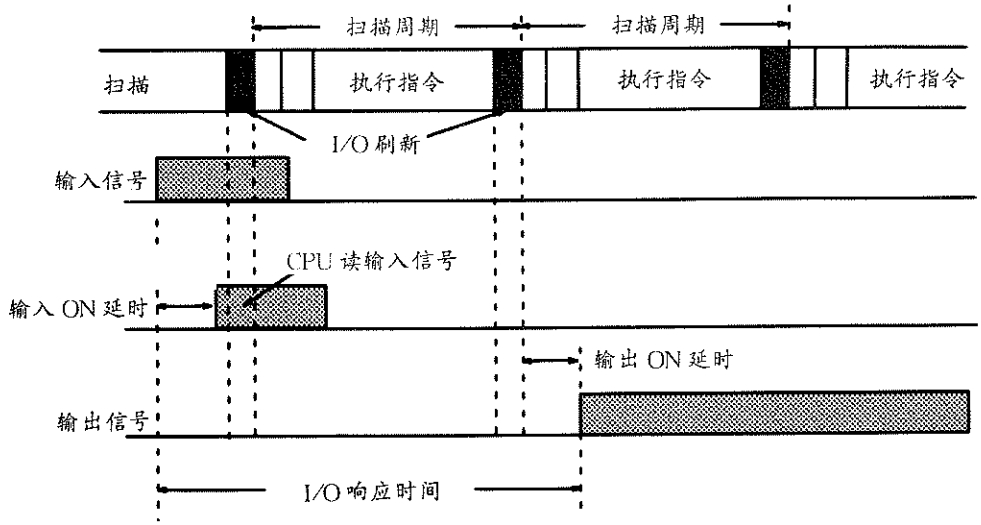
以下说明了最小和最大 I/O 响应时间的计算,IR00000 是接受信号的输入位,IR00200 是相应输出点的输出位。



6－4－1　基本系统

最小 I/O 响应时间

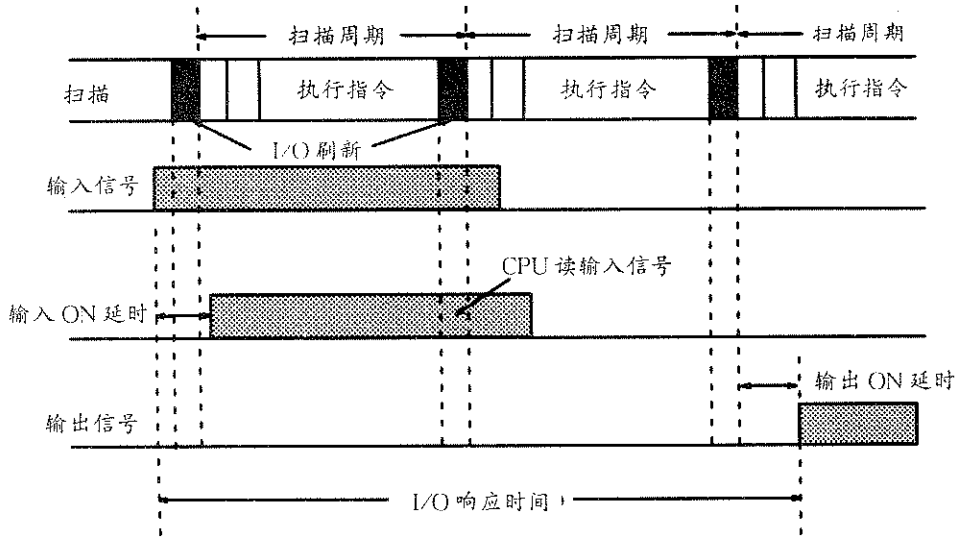
PC 恰好在扫描期 I/O 刷新时接受一输入信号,其响应速度最快。一旦与其信号对应的输入位接通,程序就立即执行,接通要输出信号的输出位,然后执行 I/O 刷新操作,刷新该输出位。这种情况的 I/O 响应时间由输入接通延时时间、扫描周期和输出接通延时时间之和来得到。该情况说明如下:



最小 I/O 响应时间＝输入接通延时＋扫描周期＋输出接通延时。

最大 I/O 响应时间

当 PC 恰好在扫描的 I/O 刷新周期之后接受到输入信号，其响应时间最长。在这种情况下，直到下个循环结束之前 CPU 不承认输入信号。最大响应时间比最小 I/O 响应时间长了一个扫描周期，只是不包括 I/O 刷新时间，因为输入信号正好在它后面到来而不是在它之前。



最大 I/O 响应时间＝输入接通延时＋(循环时间×2)＋输出接通延时。

计算举例

下表格中的数据，按下面的计算产生出最小和最大扫描周期。

项　目	时　间
输入 ON 延时	1.5ms
输出 ON 延时	15ms
扫描周期	20ms

最小 I/O 响应时间＝1.5＋20＋15＝36.5ms；

最大 I/O 响应时间＝1.5＋(20×2)＋15＝56.5ms；

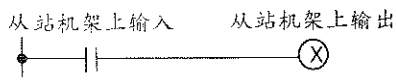
注

本例中，I/O 刷新时间忽略不计，未包括在最小 I/O 响应时间内。

6－4－2　远程 I/O 系统

带有远程 I/O 系统的 PC，在计算 I/O 响应时间时只要远程 I/O 传输时间，小于扫描周期，可以忽略，仅需考虑 PC 的循环时间。但是，远程 I/O 系统增加了扫描周期。

本节提供确定从输入到输出最大和最小响应时间的过程及其计算的方法。计算是假定输入和输出都在远程 I/O 系统的从站机架上的。这个计算对光缆型 I/O 单元，I/O 链接单元，I/O 终端等单元上 I/O 点是相同的。



如果需要，可以用更精确的算式，而下列计算的算式未考虑扫描中的细小部分。

观察下面时序图时，记住 PC 扫描期间处理过程的次序，尤其是程序被执行之前，输入不会产生程序动作，这一点是很重要的。

在计算来自通过 I/O 链接单元连接的 CPU 的输入和输出有关的响应时间时，必须同时考虑控制 CPU

扫描周期和与 I/O 链接单元连接的 CPU 扫描周期。

注意：噪声可能增加 I/O 延时。

远程 I/O 传输时间

按下式计算远程 I/O 传输时间

T_{RM} = 一个主站上总的从站传输时间

$= \Sigma T_{RT} + \Sigma T_{TT}$

T_{RT} = 每个从站传输时间

$= 1.4ms + (0.2ms \times n)$

式中 n = 从站机架上 I/O 字数。

T_{TT} = 光缆型 I/O 单元/I/O 终端传输时间

$= 2ms \times m$

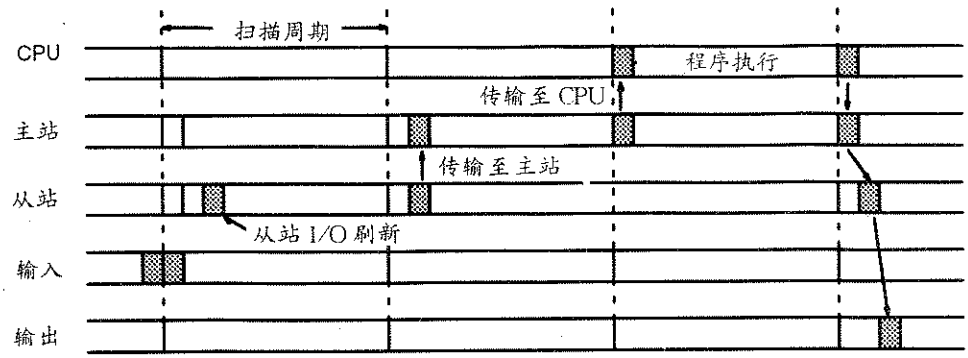
式中 m = 光缆型 I/O 单元/I/O 终端的数量

最小 I/O 响应时间

当所有信号一接收到，立即进行处理，这时响应时间为最小。这里，需要经过 3 次扫描，程序被执行，如下图所示。

时间 = 输入接通延时 + 扫描周期 × 3 + 输出接通延时

扫描周期 > 远程 I/O 传输时间。



最大 I/O 响应时间

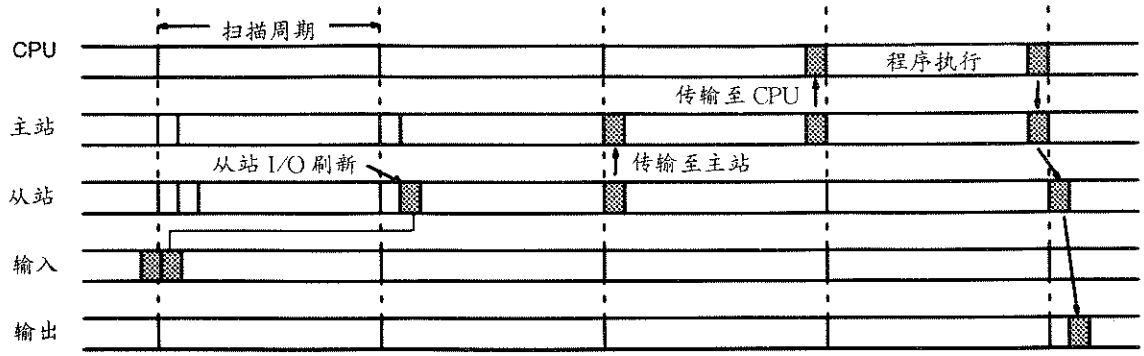
当输入恰好错过扫描的程序执行期，也就是等候下一次传输，然后再下一次扫描（第四次扫描）才进行处理，这时响应时间为最大。

时间 = 输入接通延时 + 扫描周期 × 4 + 输出接通延时。

扫描周期 > 远程 I/O 传输时间。

注

计算最大 I/O 响应时间时，使用输出到 AR26 的最大扫描周期。



计算举例

对于输入接通延时为 1.5ms，输出接通延时为 15ms 和扫描周期为 20ms 的响应计算如下所示

最小响应时间

时间＝1.5ms＋(20ms×3)＋15ms＝76.5ms

最大响应时间

时间＝1.5ms＋(20ms×4)＋15ms＝96.5ms

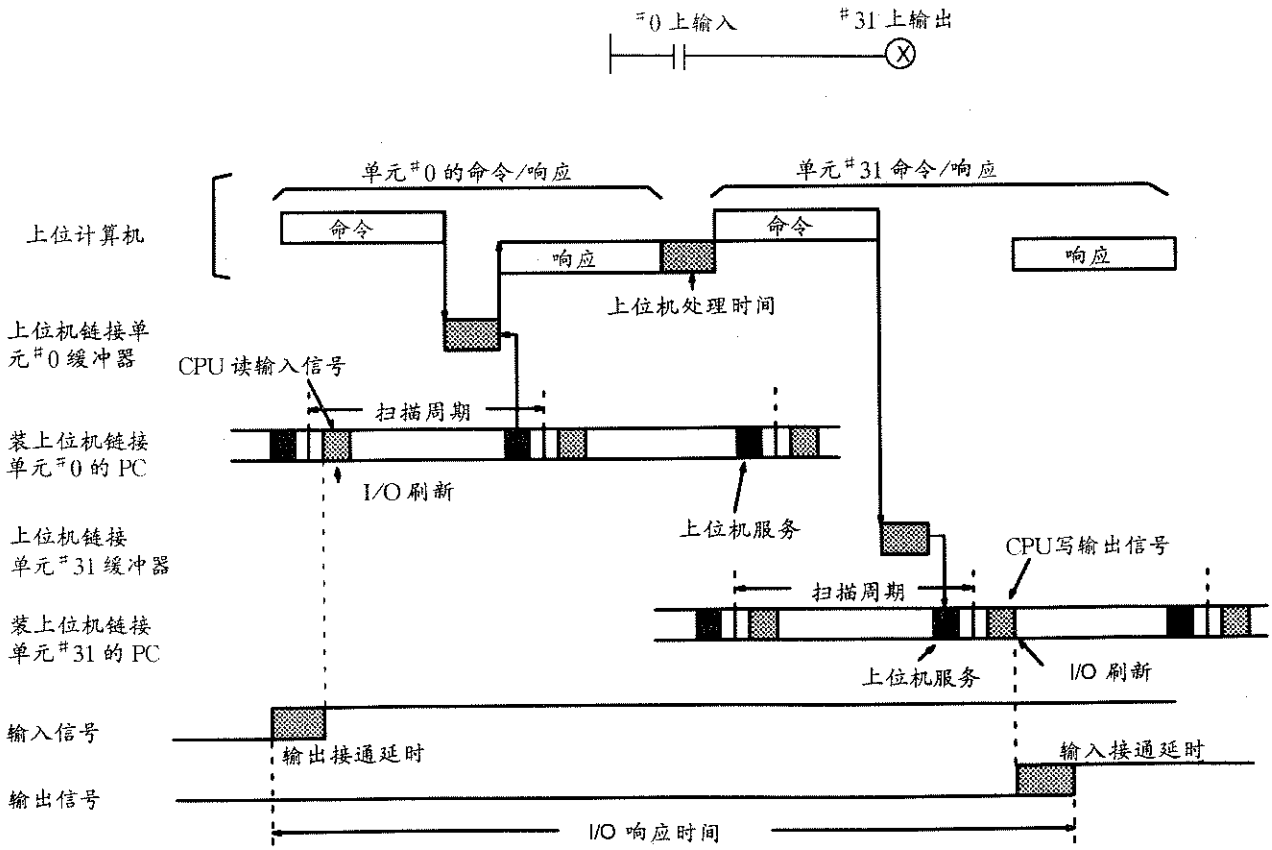
注 1. 当从站机架上装有特殊 I/O 单元，扫描周期可能小于或等于远程 I/O 传输时间。如果是这样，可能有些扫描，在主站和 C200HX/HG/HE CPU 之间没有进行 I/O 刷新。

2. 对主站，每次扫描只进行一次刷新，然后只有在确认完成一个远程扫描之后才作下一次刷新。

3. 在处理远程 I/O 系统时，除非采取合理的编程，由微分型指令产生的短时 ON/OFF 状态会引起不正确的信号。

6－4－3 上位机链接系统

下图说明了一台 PC 上的输入通过上位机链接系统接通另一台 PC 上输出的过程。详细说明可参阅上位机链接系统资料。



下面给出计算最大和最小扫描次数的算式。每台 PC 所需的扫描次数取决于读/写数据的数量。

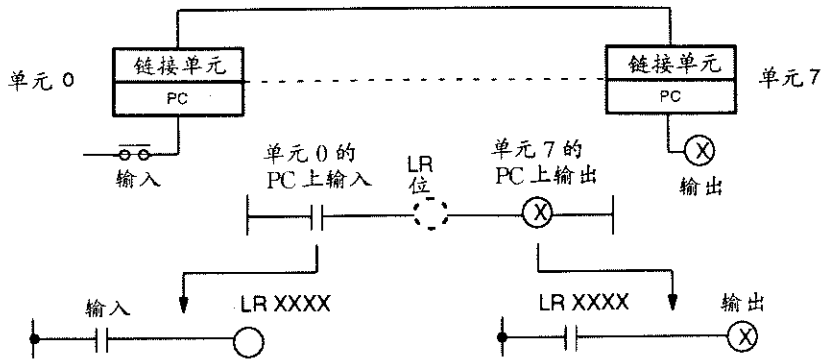
最小响应时间＝输入接通延时＋命令传输时间＋(单元＃0 上 PC 的扫描周期×3)＋响应传输时间＋上位机处理时间＋命令传输时间＋(单元＃31 上 PC 的扫描周期×3)＋输出接通延时

最大响应时间＝输入接通延时＋命令传输时间＋(单元＃0 上 PC 的扫描周期×10)＋响应传输时间＋上位机处理时间＋命令传输时间＋(单元＃31 上 PC 的扫描周期×10)＋输出接通延时。

6－4－4 PC 链接系统

本节提供确定从输入到输出最大和最小响应时间的过程及其计算方法，后面所有例子都使用下述系统和 I/O 程序。该系统包括 8 个 PC 链接单元。

在观察下面时序图时，记住 PC 扫描期间处理过程的次序，尤其是程序被执行之前，输入不会产生程序动作这一点是很重要的。



注

噪声可能增加 I/O 延时。

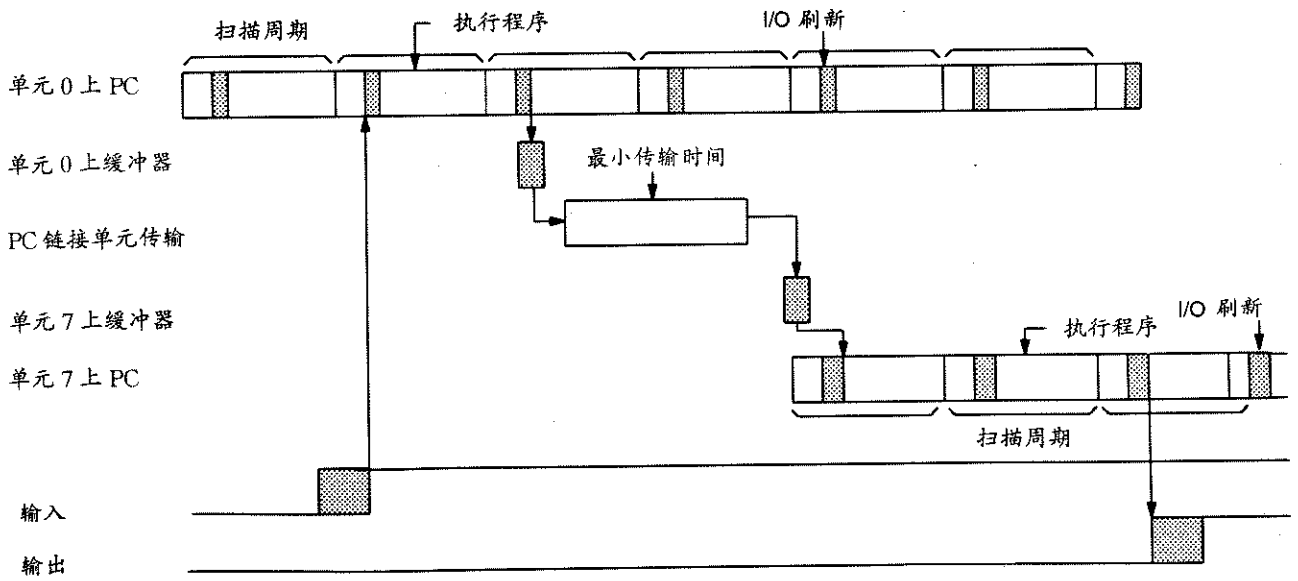
PC 链接条件

在本例中使用的 PC 链接系统组成：

- 链接的 PC 数：8 个
- 链接的 LR 点数：每台 PC128 点
- 最多 PC：8 个
- 使用 LR 点数：1,024 点

最小响应时间

下图表示一种数据流向，由它产生最小响应时间，即所有信号和数据传输一出现就得到处理所获得的时间。



计算最小 I/O 响应时间的算式如下：

响应时间＝输入接通延时＋单元 0 上 PC 的扫描周期＋最小传输时间＋(单元 7 上 PC 的扫描周期×2)＋输出接通延时

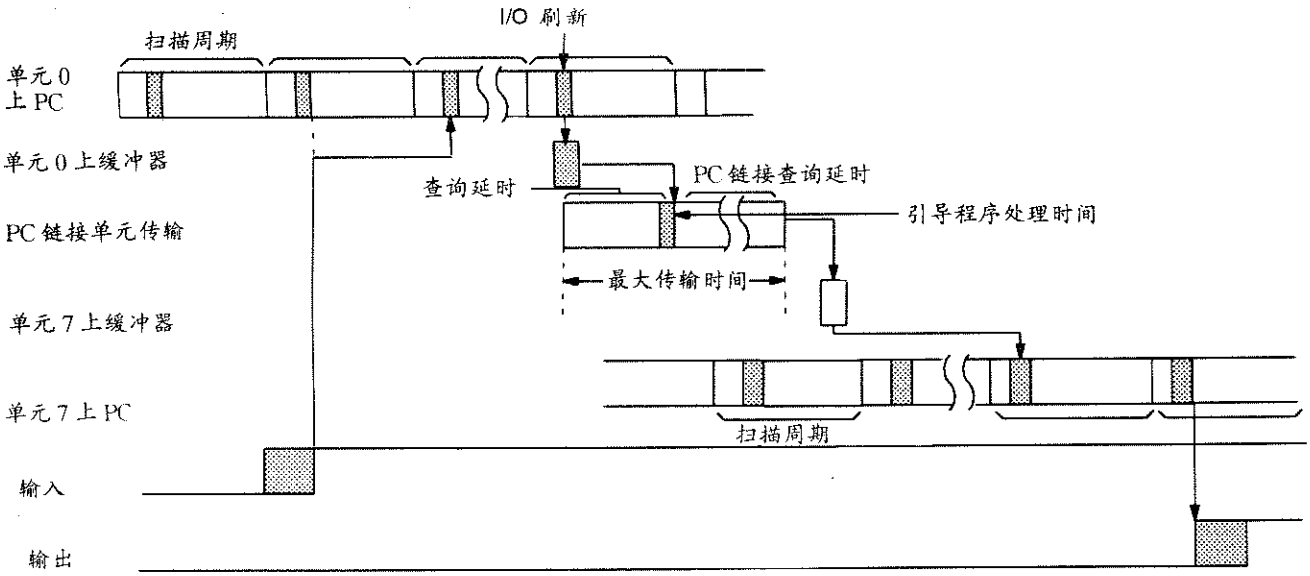
算式中代入下列数值，得到最小 I/O 响应时间149.3ms：

输入接通延时	1.5ms
输出接通延时	1.5ms
单元 0 上 PC 扫描周期	20ms
单元 7 上 PC 扫描周期	50ms
最小传输时间	2.8ms＋10ms＝12.8ms

最大响应时间

下图表示产生最大响应时间的数据流向。因为信号或数据恰好在对它们要进行处理之后接收到或者数据在处理期间才发送，造成延迟，不论哪种情况，处理都将等到下次循环/询问周期。

送到查询单元缓冲器的第一个输出被设置每次扫描刷新 LR 位的数量延时，当 LR 数据到达单元 7，也遇到类似的延时。查询延时是 LR 数据在自身的 PC 中被更新的结果。这个更新是在前一个 LR 数据被送主 PC 链接单元缓冲器之后立即进行的。它造成了等候下一个查询周期的延时。数据传送到 PC 链接单元 7 中缓冲器，需要 1 个以上查询周期。



计算最大 I/O 响应时间的算式如下：

响应时间＝输入接通延时＋[单元 0 上 PC 扫描周期×(LR 传送位数÷I/O 刷新位)]＋α＋(PC 链接查询时间＋引导程序处理时间)＋{单元 7 上 PC 扫描周期×[LR 传送位数÷I/O 刷新位]×2＋1}＋β＋输出接通延时

如果单元 0 上 PC 扫描周期＞PC 链接查询时间，α＝单元 0 上 PC 扫描周期。如果单元 0 上 PC 扫描周期＜PC 链接查询时间，α＝PC 链接查询时间。

如果单元 7 上 PC 扫描周期＞PC 链接查询时间，β＝单元 7 上 PC 扫描周期。如果单元 7 上 PC 扫描周期＜PC 链接查询时间，β＝PC 链接查询时间。

上式中代入下列数值，得到最 I/O 响应时间661.3ms。

输入接通延时：	1.5ms
输出接通延时：	15ms
单元 0 上 PC 扫描周期：	20ms

单元 7 上 PC 扫描周期：50ms

PC 链接查询时间： 2.8ms×8(PC)＋10ms＝32.4ms

引导程序处理： 15ms×(8PC－8PC)＝0ms

单元 0 I/O 刷新位： 256

单元 7 I/O 刷新位： 256

减小响应时间

可以在程序中使用 IORF(97)，并设置大量的刷新位以缩短 I/O 响应时间。(记住，增加在面板背面设置的刷新位数会缩短响应时间，但却增加了 PC 的扫描周期。)

下述计算最大响应时间使用相同的例子，对单元 0 和 7 的 PC 编程中，每个 PC 扫描周期中，执行 IORF(97)对 PC 链接单元进行刷新。最大响应时间算式如下：

响应时间＝输入接通延时＋[单元 0 上 PC 扫描周期×(LR 传送位数÷I/O 刷新位数÷2)]＋α＋PC 链接查询时间＋引导程序处理时间＋{单元 7 上 PC 扫描周期×[(LR 传送位数÷I/O 刷新位数÷2)×2＋1]}＋β＋输出接通延时

如果单元 0 上 PC 扫描周期＞PC 链接查询时间，α＝单元 0 上 PC 扫描周期。如果单元 0 上 PC 扫描周期＜PC 链接查询时间，α＝PC 链接查询时间。

如果单元 7 上 PC 扫描周期＞PC 链接查询时间，β＝单元 7 上 PC 扫描周期。如果单元 7 上 PC 扫描周期＜PC 链接查询时间，β＝PC 链接查询时间。

举例的系统配置中所需数据为：

输入接通延时	1.5ms
输出接通延时	15ms
单元 0 上 PC 扫描周期	20ms＋5.7ms＝25.7ms (执行 IORF 需要 5.7ms)
单元 7 上 PC 扫描周期	50ms＋5.7ms＝55.7ms (执行 IORF 需要 5.7ms)
PC 链接单元数	8
LR 位数	1,024
单元 0 的刷新位数	256
单元 7 的刷新位数	256
PC 查询时间	2.8ms×8(PC)＋10ms＝32.4ms
引导程序处理时间	15ms(8PC－8PC)＝0ms

将这些值代入上式，得到最大 I/O 响应时间 466.9ms，比不用 IORF 大约缩短 200ms。

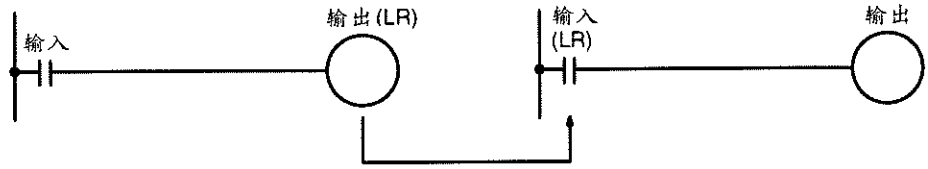
6－4－5 1 对 1 链接 I/O 响应时间

当二台 C200HXHG/HE 采用 1 对 1 链接，I/O 响应时间就是在一台 C200HX/HG/HE 上执行的输入通过 1 对 1 链接通讯输出到另一台 C200HXHG/HE 所需的时间。

1 对 1 链接在主站和从站之间实现互相通信。相应的传输时间取决于使用 LR 的字数，列于下表。

使用字数	传输时间
64 字(LR00～LR63)	39ms
32 字(LR00～LR31)	20ms
16 字(LR00～LR15)	10ms

下面以主站和从站上执行下述指令作为例子表示最小和最大 I/O 响应时间



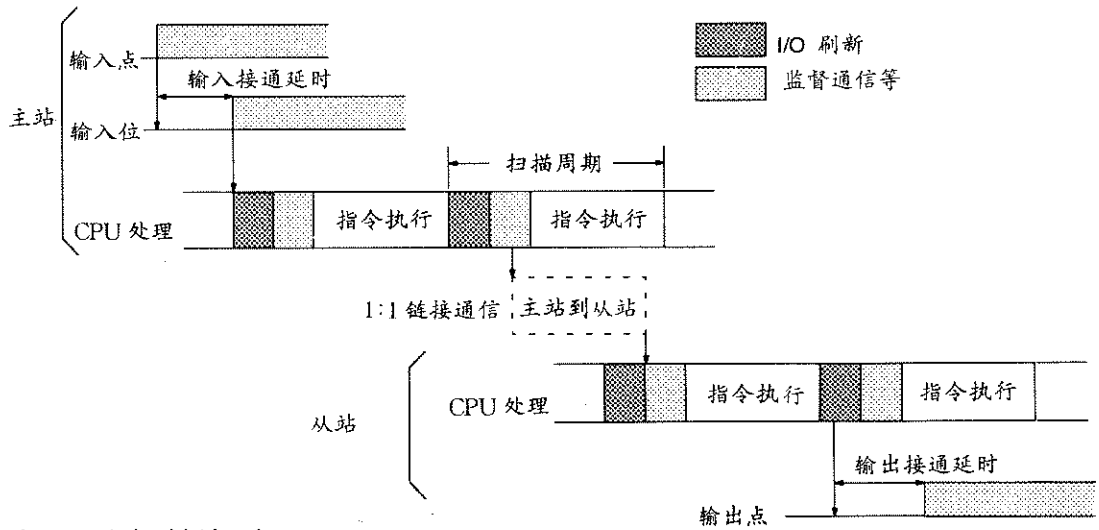
用下列条件作为计算 I/O 响应时间的例子

输入接通延时 8ms
主站扫描周期 10ms
从站扫描周期 14ms
输出接通延时 10ms
LR 字数 64 字

最小 I/O 响应时间

C200HX/HG/HE 在下列情况下响应最快：

1. C200HX/HG/HE 恰好在扫描期输入刷新阶段前接收输入信号。
2. 主站至从站的传输立即开始。
3. 通信完成之后，从站立即执行通信服务。



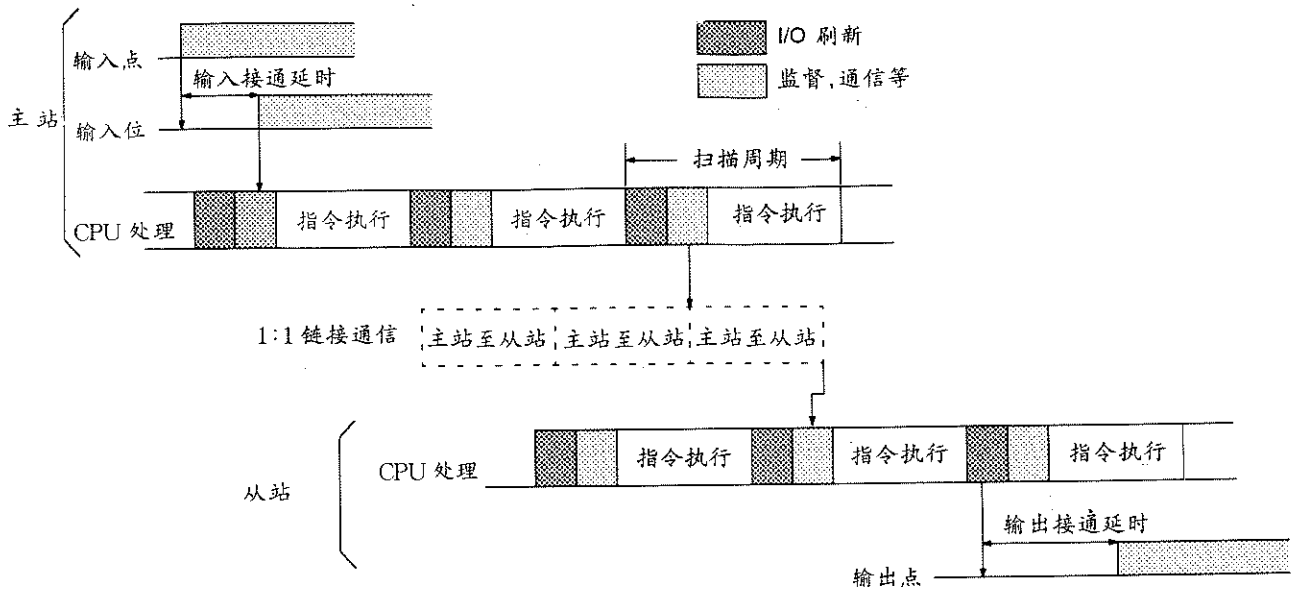
最小 I/O 响应时间如下：

输入接通延时	8ms
主站扫描周期	10ms
传输时间	39ms
从站扫描周期	15ms
＋ 输出接通延时	10ms
最小 I/O 响应时间	82ms

最大 I/O 响应时间

C200HX/HG/HE 在下列情况下响应时间最长：

1. C200HX/HG/HE 恰好在扫描期输入刷新阶段之后接收到输入信号。
2. 主站至从站通信不立即开始。
3. 通信恰好在从站执行通信服务之后完成。



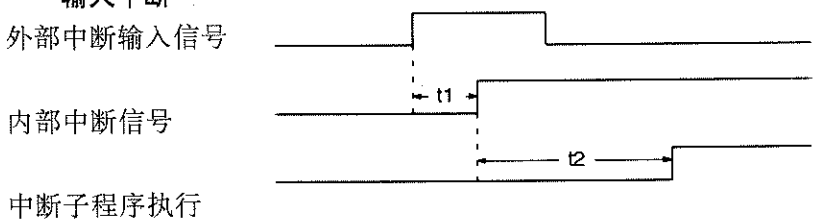
最大 I/O 响应时间如下：

	输入接通延时	8ms
	主站扫描周期	10ms×2
	传输时间	39ms×2
	从站扫描周期	15ms×2
+	输出接通延时	10ms
	最大 I/O 响应时间	185ms

6－4－6 中断响应时间

从接收到中断输入到执行中断子程序结束的响应时间叙述如下。

输入中断



t_1 = 中断输入单元接通延时

t_2 = 软件中断响应时间

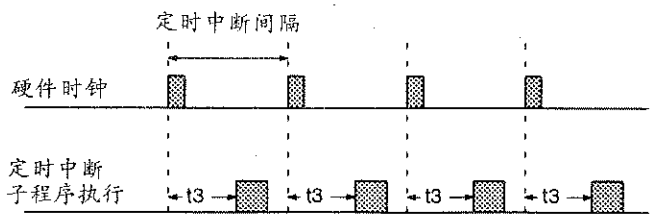
总的中断响应时间 = $t_1 + t_2$

中断输入单元接通延时是等于或小于 0.2ms。

软件中断响应时间取决于 PC 设置 DM6620 中所设置的中断响应参数。如果 DM6620 设置成 C200H 兼容方式(0000)，软件中断响应时间小于 10ms。如果 DM6620 设置成 C200HX/HG/HE 方式(1×××)，软件中断响应时间小于 1ms。总的中断响应时间如下表所示。

中断响应设置	总的中断响应时间
C200H 兼容方式(正常响应)	0.2ms＋(下述总和:特殊 I/O 处理时间,远程 I/O 处理时间,上位机链接服务时间,指令执行时间)
C200HS 方式(高速响应)	等于或小于 1.2ms

定时中断



t3＝软件中断响应时间

总的中断响应时间＝t_3(软件中断响应时间)

软件中断响应时间取决于 PC 设置 DM6620 中所设定的中断响应参数。如果 DM6620 设置成 C200H 兼容方式(0000),软件中断响应时间小于 10ms。如果 DM6620 设置成 C200HS 方式(1×××),软件中断响应时间小于 1ms。总的中断响应时间如下表所示。

中断响应设置	总的中断响应时间
C200H 兼容方式(正常响应)	下述总和:特殊 I/O 处理时间,远程 I/O 处理时间,上位机链接服务时间,指令执行时间
C200HS 方式(高速响应)	等于或小于 1.0ms

注 1. 当使用 C200H 兼容方式,程序中有某条指令执行时间长于 10ms,则总的中断响应时间等于这条需要 10ms 以上的指令执行时间。

2. 上述计算假定某个时间只需要执行一个中断。如果多个中断同时发生,那除第 1 个之外的其它中断都将等待,上面给出的响应时间将增加。

3. 在 C200HS－SLK□□或 C200HS－SNT□□正在服务时发出中断,中断一定要等到 SYSMAC NET/SYSMAC LINK 单元服务完成之后才被处理。此时响应时间按下表,不受中断响应设置所影响。

中　断	总的中断响应时间
输入中断	最大 10.2ms
定时中断	最大 10ms

当 C200HX/HG/HE 上使用 C200HW－SLK□□,这个限制不适用。

中断处理时间

从接收中断输入,程序执行直至返回原程序总的处理时间叙述如下。

使用定时中断的周期或输入中断作为计数输入所造成的计数频率极限是由中断处理时间所决定。

中断处理时间＝总的中断响应时间＋中断程序执行时间＋中断返回时间

中断程序执行时间决定于中断子程序的内容。如果只执行 SBN(92)和 RET(93)这个时间可以忽略不计。

中断返回时间是 0.04ms。

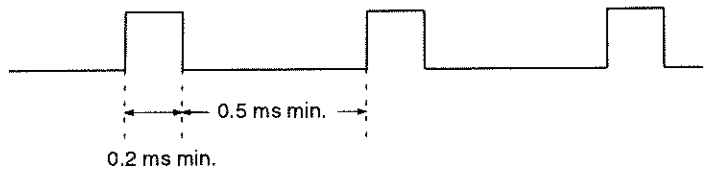
注 1. 如果有几个单元会产生中断或者中断期小于平均中断处理时间,则只执行中断子程序,而主程

序将不执行。这会造成扫描监视时间超过，产生 FALS9F 错误，PC 停止运行。

2．最大中断程序的执行时间保存在 SR262 和 SR263 中。

中断输入脉冲宽度

中断输入单元的输入脉冲宽度必须设置在下图所示的范围之内。



接通时间：最小 0.2ms。

关断时间：最小 0.5ms。

第七章　程序监视和执行

本节提供了通过一个编程器来监视和控制 PC 的方法和步骤。如果你使用计算机运行 SSS(中文版),使用 SSS(中文版)的步骤可参阅 SYSMAC 支持软件的“操作手册”。

7－1　监视操作和修改数据…… 362
7－2　编程器操作…… 362
7－2－1　位/字监视…… 362
7－2－2　强制设置位/复位…… 366
7－2－3　强制设置位/复位的取消…… 367
7－2－4　十六进制/BCD 码数据修改 …… 368
7－2－5　十六进制/ASCⅡ码显示转换…… 369
7－2－6　四位十六进制/十进制显示转换…… 369
7－2－7　八位十六进制/十进制显示转换…… 370
7－2－8　微分监视 …… 371
7－2－9　三字监视 …… 372
7－2－10　三字数据修改…… 373
7－2－11　二进制监视…… 373
7－2－12　二进制数据修改…… 375
7－2－13　改变定时器/计数器 SV …… 378
7－2－14　扩展指令功能码的配置…… 379
7－2－15　UM 区域分配…… 380
7－2－16　读取和设置时钟…… 382
7－2－17　扩展键盘映象…… 383
7－2－18　键盘映象…… 383

7－1 监视操作和修改数据

监视操作的最简单的形式是显示地址，而该地址的操作数位状态是由采用程序读或搜索操作来进行监视的。只要以运行方式或监视方式执行操作，就可指示出被显示的任何一位的状态。

本节还提供监视和修改存在于数据区内数据的其它方法。任何定时器和计数器的数据都可修改，包括修改 PV(当前值)和 SV(设定值)。

本节中所有监视操作，可采用运行、监视、编程方式执行，并可按 CLR 键取消监视操作的执行。

除了修改定时器和计数器 SV 以外，所有的数据都在先执行监视器操作之后，才能进行修改操作。无论采用监视方式或者编程方式，都能进行数据修改，但不能在运行方式下执行数据修改。

7－2 编程器操作

7－2－1 位/字监视

采用以下操作可监视任何数据区中任何位或字的状态，虽然可在任何方式下进行操作，但是对于位来说，只能在监视或运行方式下提供 ON/OFF 状态的显示。可以由指定第一个被监视的位或字从已清除的显示进入位/数字的监视操作，也可以从程序的任何地址显示要监视的位或字的地址进入监视操作。这两种进入监视操作的方式都要按 MONTR 键。

当监视一个位时，它的 ON/OFF 状态也被显示(采用监视或运行方式)。除定时器或计数器以外，当设置一个字地址时，将显示字的数字内容；当设置定时器或计数器号的时候，显示定时器的 PV 值。如果定时器或计数器的完成标志为 ON 时，会显示出一个小方块。当监视多个字时，为了帮助辩认不同地址之间的区别，一个脱字符(∧)会显示在设定地址最左边的数字下面。当执行 END(01)时，不能监视 TR 位和 SR 标志位(例如算术标志)的状态。

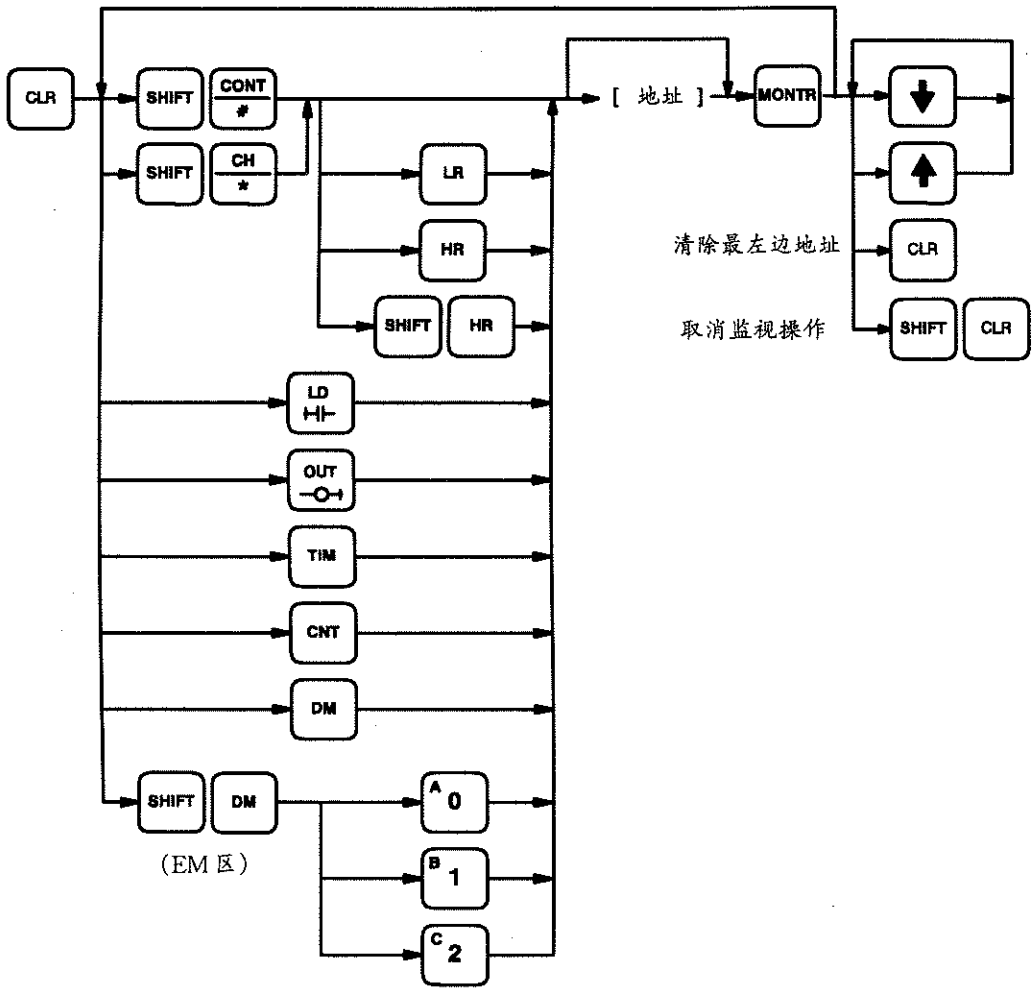
可一次监视多达 6 个存储器地址，它们可以是位、字或者字位的混合，但在任一时间，只显示这其中的三个。对于多地址的监视，可返回步骤的起始处继续设定地址。除非设定超过 6 个地址，都可保持全部设定地址的监视。如果设定多于六个地址，正被监视的地址中最左边的那个地址就会被取消。

为了显示被监视的，而又不在当时编程器上所显示的地址，在未设定另一个地址时，按 MONTR 键，使正被监视的地址向右移位。按下 MONTR 键时，正监视的地址的继续向右移，直到最右面的地址被移回到左面显示。

在监视操作期间，按向上(UP)、向下(DOWN)键，递增和递减显示器上最左边显示的地址。按 CLR 键，取消在显示器上所监视的最左边的地址。如果取消最末的地址，就会取消监视操作。当按下 SHIFT 键，然后按 CLR 键，不管被监视的地址号是什么，监视操作都被取消。

LD 和 OUT 只能用于设定要被显示的第一个地址，当已经在监视一个地址的时候，就不能再使用它们了。

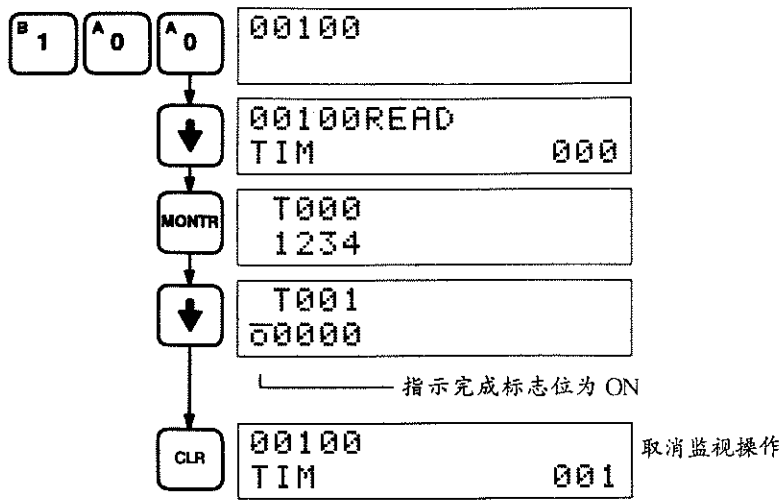
键操作顺序



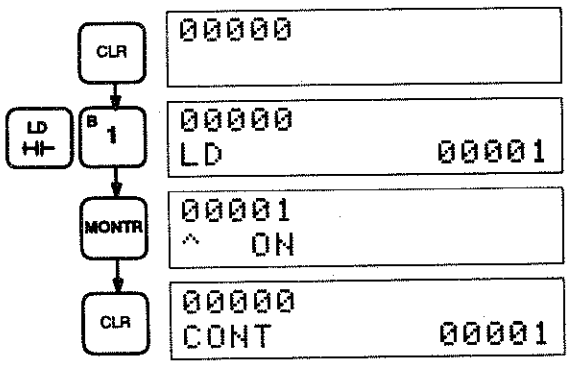
例

以下例子给出该监视操作的各种应用

读程序并监视

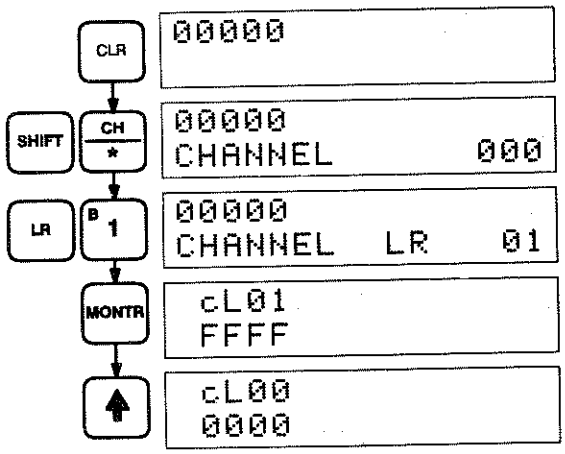


位监视

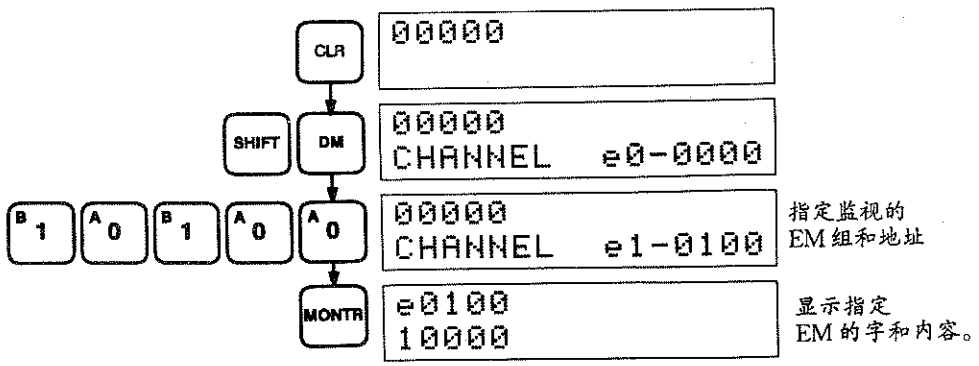


注
当执行 END(01)时，TR 位和 SR 标志位 SR25503～25507(例如算术标志)状态清除，不能监视。

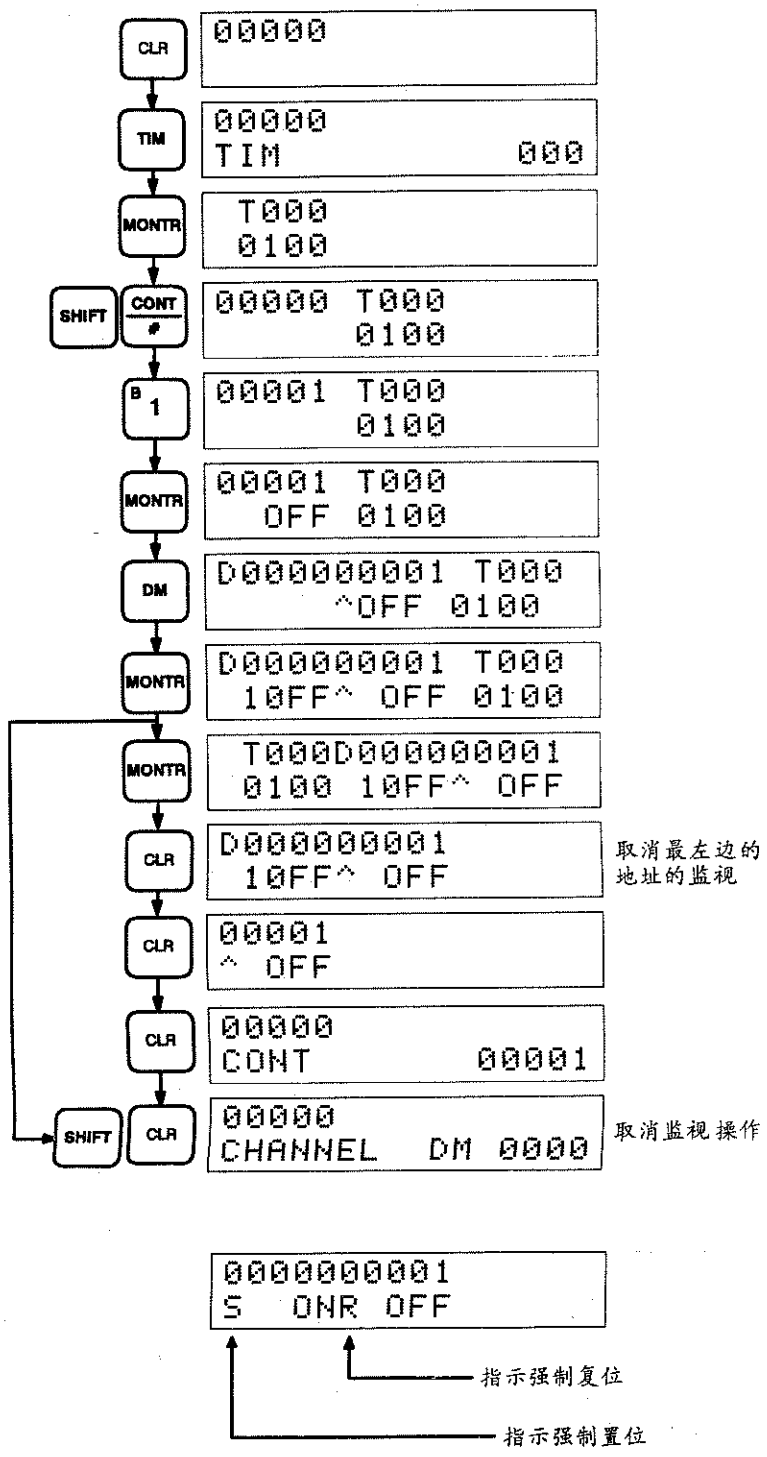
字监视



EM 区字监视



多地址监视：


CLR
00000
TIM
00000
TIM 000
MONTR
T000
0100
SHIFT
CONT
#
00000 T000
0100
B
1
00001 T000
0100
MONTR
00001 T000
OFF 0100
DM
D0000000001 T000
^OFF 0100
MONTR
D0000000001 T000
10FF^ OFF 0100
MONTR
T000D0000000001
0100 10FF^ OFF
CLR
D0000000001
10FF^ OFF
取消最左边的地址的监视
CLR
00001
^ OFF
CLR
00000
CONT 00001
SHIFT
CLR
00000
CHANNEL DM 0000
取消监视操作
0000000001
S ONR OFF
指示强制复位
指示强制置位

7－2－2 强制置位/复位

当执行位/数监视操作，并且位、定时器或计数器地址显示在显示屏最左边时，可按下 PLAY/SET 键；接通位、启动定时器或增加计数器值；而欲断开位或复位定时器、计数器可按 REC/RESET 键。定时器在编程方式下不工作，SR 位也不能在这方式下通(ON)或断(OFF)。

只要一直按下键不放，位状态就维持在 ON 或 OFF，一旦释放键，马上返回到原来状态。如果启动一个定时器，而设定值(SV)已达到，则它的完成标志位通。

按下 SHIFT 和 PLAY/SET 或 SHIFT 和 REC/RESET 键后释放键仍可维持位状态，直到按下 NOT 键或满足以下条件之一，该位才会返回它原来的状态。

1．执行强迫状态清除操作；

2．改变 PC 方式(见注)；

3．由于致命错误或电源中断造成操作停止；

4．执行 I/O 表格登记操作。

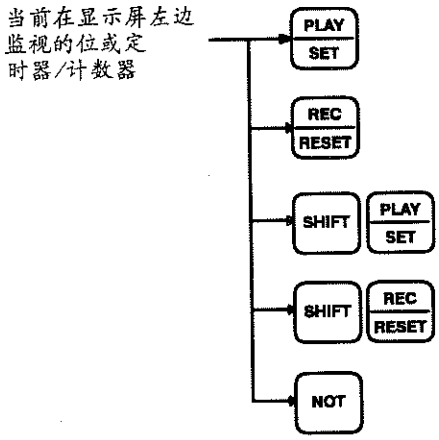
为了在实际程序执行之前检查从 PC 的输出接线，该操作可在监视方式下进行，而不能在运行方式下进行。

注

当把开关从编程转到监视方式时，如果强迫状态保持位为 ON，并且 PC 设置中 DM6601 被设置成保持位状态，则强迫置位/复位的位状态就被保持。

详见 3－6－4“PC 设置”。

键操作顺序



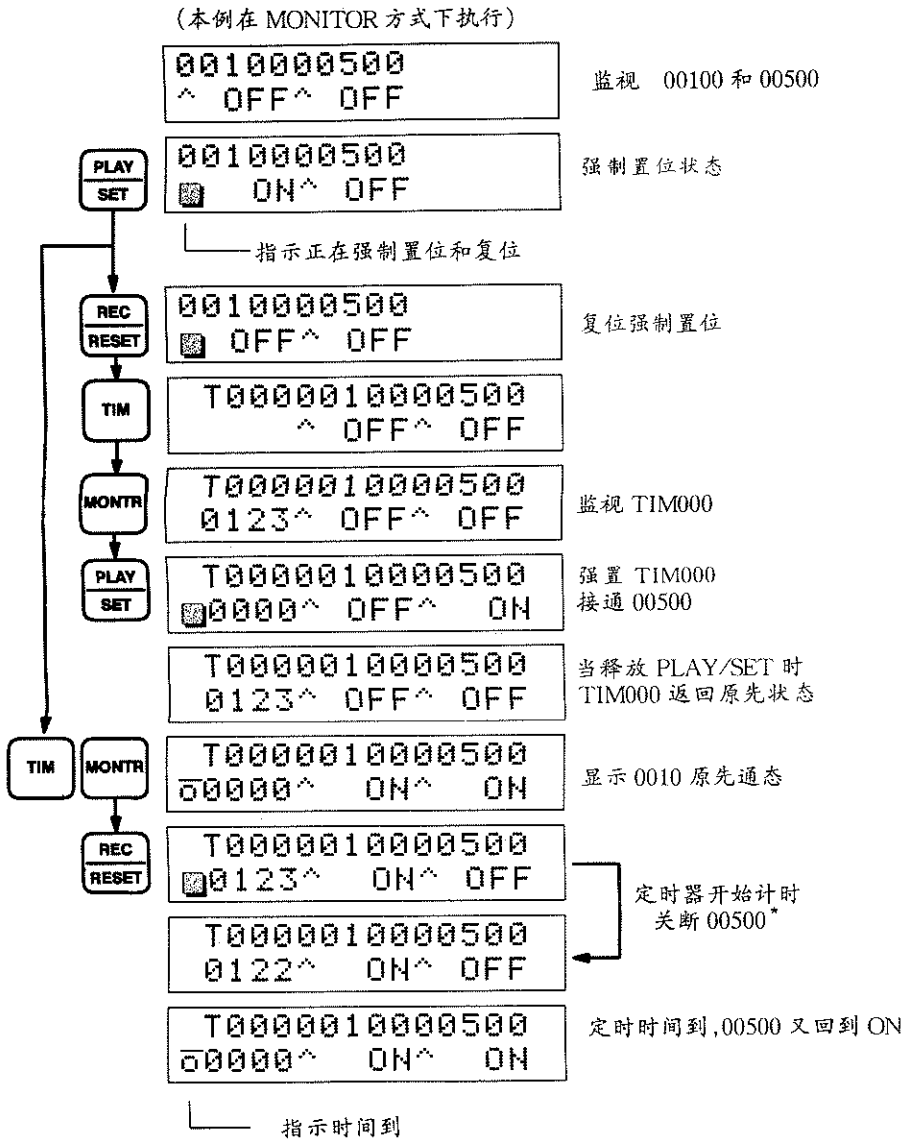
例

以下例子给出如何用强制置位/复位操作来控制位或定时器。如下图所示



地址	指令	数据
00200	LD	00100
00201	TIM	000
		# 0123
00202	LD	TIM 000
00203	OUT	00500

下图显示了当用 00100 OFF 设置 TIM000 时发生的情况(例如 00500 被接通),以及当用 00100ON TIM000 复位时发生的情况(例如定时器开始工作,关断 00500,当定时器完成递减计数到 SV 时,00500 返回到 ON)。



* 计时!不能在 PROGRAM 方式下进行

7－2－3 强制置位/复位的取消

该操作恢复在 I/O、IR、TIM、CNT、HR、AR 或 LR 区域中所有被强制置位或复位的位状态。这个操作可在编程方式或监视方式下执行。

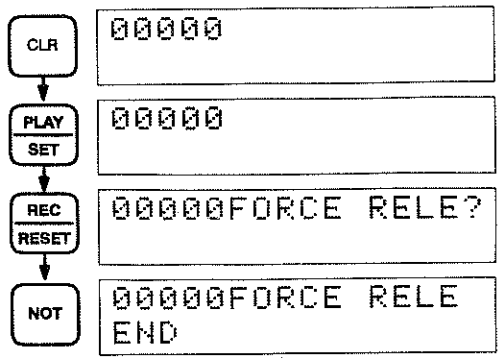
键操作顺序



当按下 PLAY/SET 和 REC/RESET 键时,蜂鸣器发声。如果你错按了键,可按 CLR 键并重新开始。

例

以下例子给出正常执行恢复状态时,所出现的显示:



7－2－4 十六进制/BCD 码数据修改

当执行位/数监视操作，并有 BCD 或十六进制值显示在显示屏的最左边时，按 CHG 键可用以修改数值，SR 字不能被修改。

如果在显示屏最左边显示的是定时器或计数器，则 PV 值被显示并可修改。修改定时器和计数器 SV 的方法，参见 7－2－13 改变“定时器/计数器 SV”。当定时器或计数器工作时，只能在监视方式下修改 PV。

为了修改最左边字地址的内容，按 CHG 键，输入所要修改的值，然后按 WRITE 键。

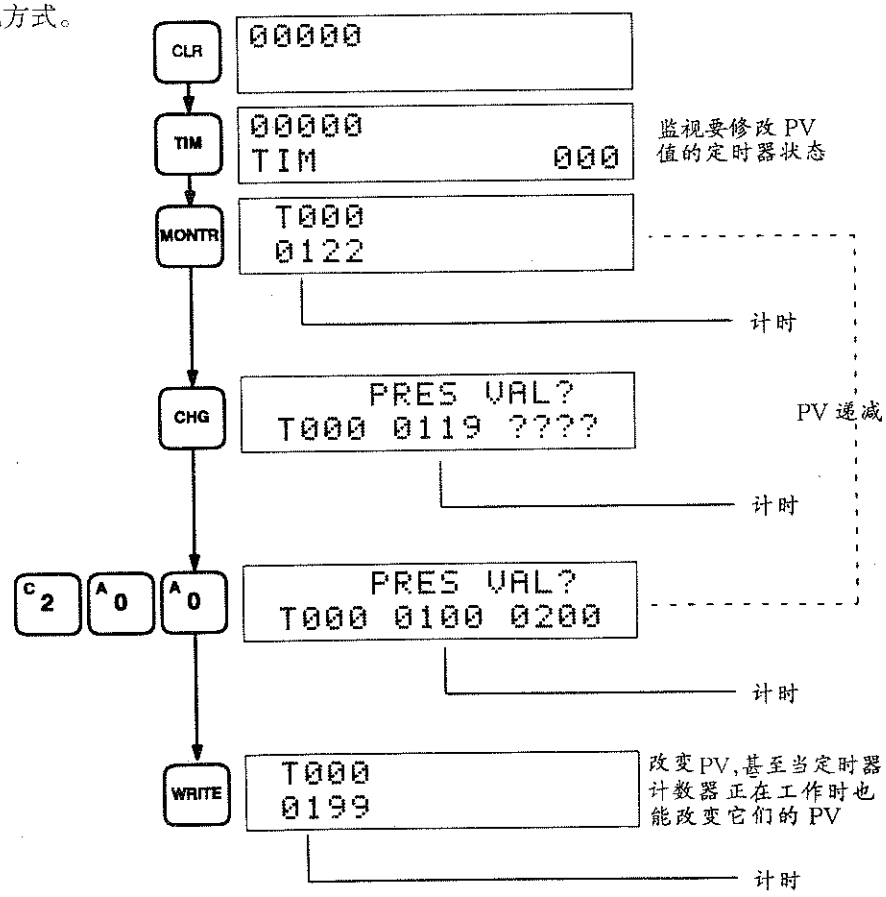
键操作顺序

正在显示器左边所监视的字 → CHG → 数据 → WRITE

例

下例给出改变定时器 PV 的情况。

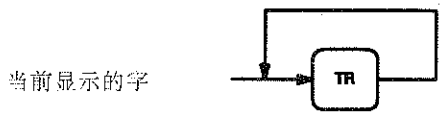
本例处于监视方式。



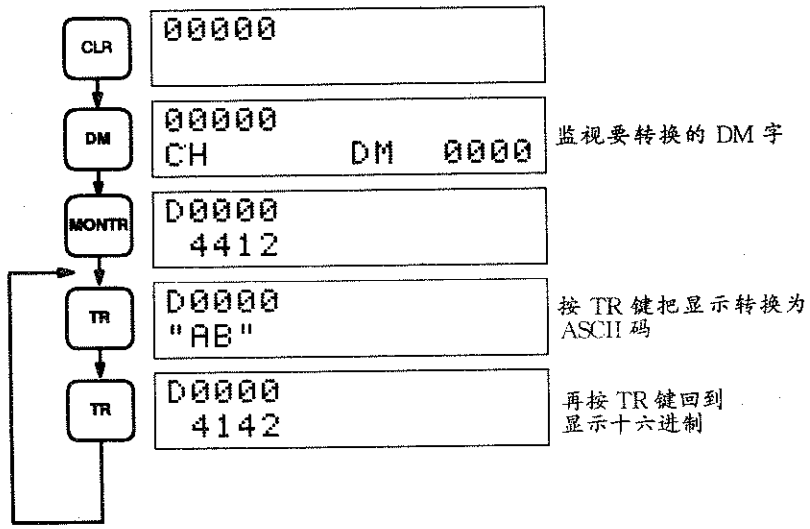
7－2－5 十六进制/ASCⅡ码显示转换

该操作把 DM 数据的显示从 4 位十六进制的数据转换到 ASCⅡ码，反之亦然。

键操作顺序



例

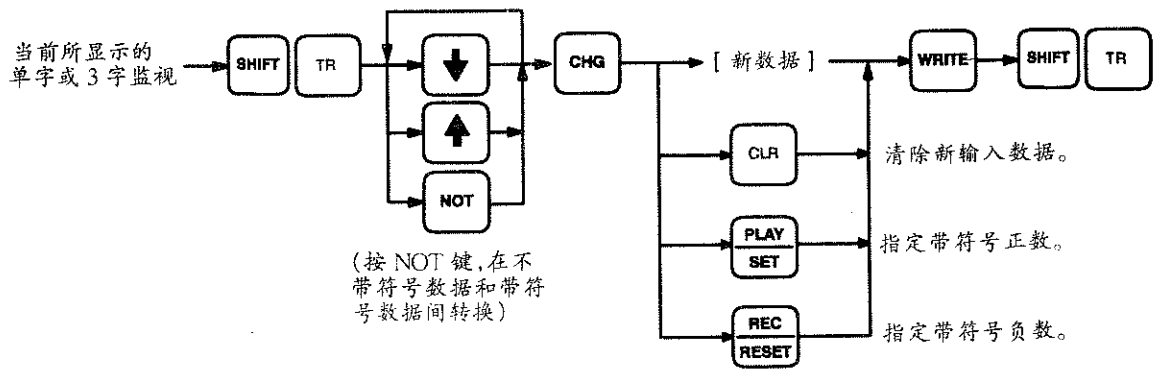


7－2－6 四位十六进制/十进制显示转换

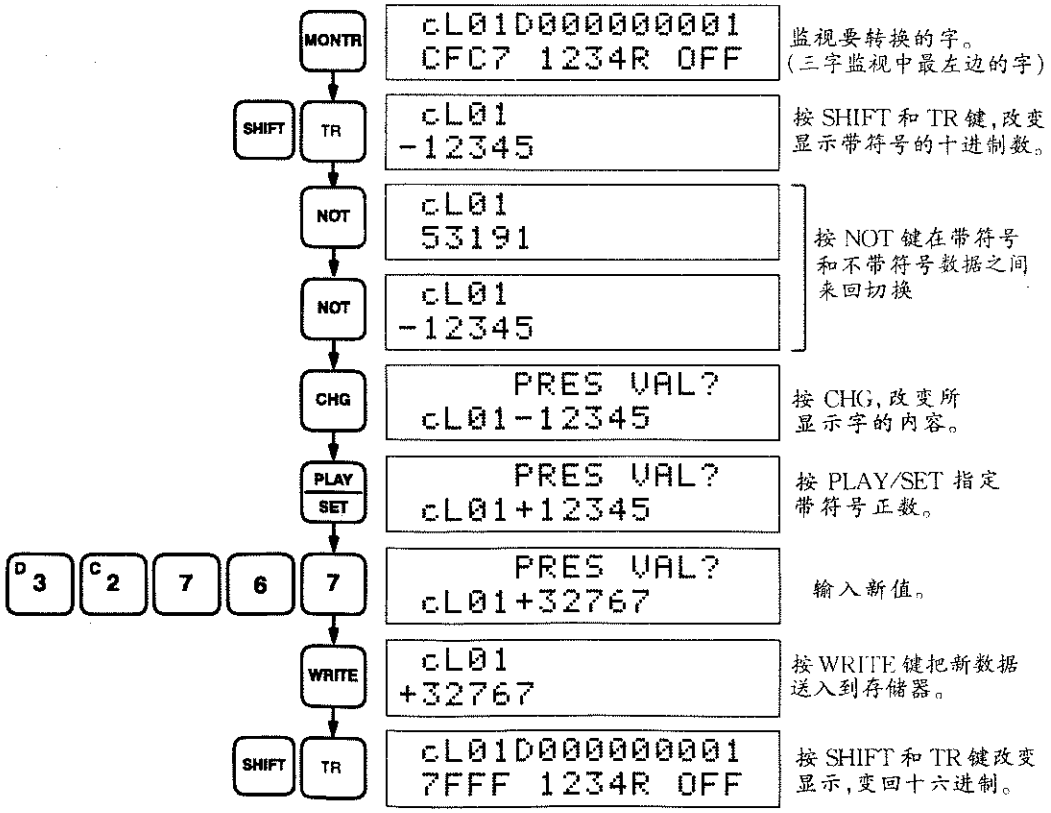
该操作把不带符号的或带符号的四位十六进制数据显示转换到十进制数显示，反之亦然。

当输入不带符号的四位十六进制数时，对应十进制数有效范围为 0～65,535；当输入带符号四位十六进制数据时，对应十进制数有效范围为 －32,768～＋32,767。

键操作顺序



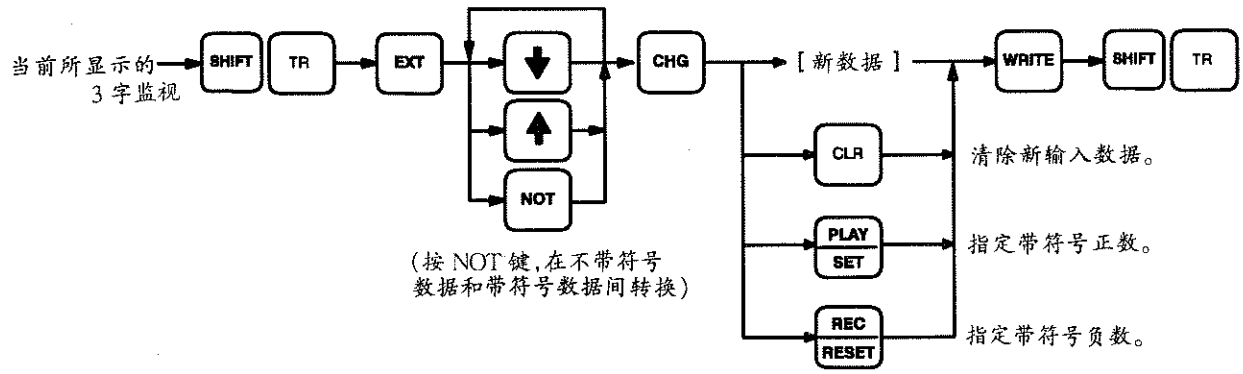
例

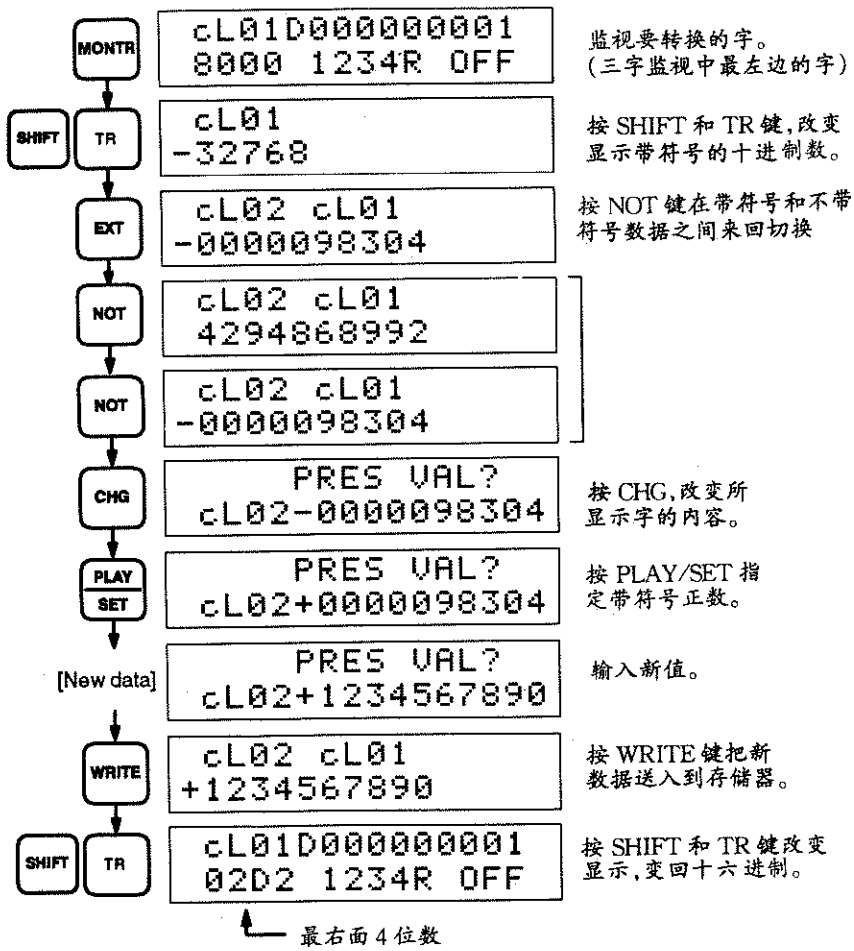


7－2－7 八位十六进制/十进制显示转换

该操作把不带符号的或带符号的 4 位或 8 位十六进制数据显示转换到十进制数据显示,反之亦然。

当输入不带符号的 8 位十六进制数据时,对应十进制数的有效范围为 0～4,294,967,295;当输入带符号的 8 位十六进制数据时,对应的十进制数的有效范围为 2,147,483,648,2,147,483,647。



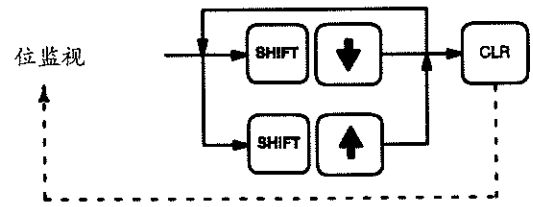


7－2－8 微分监视

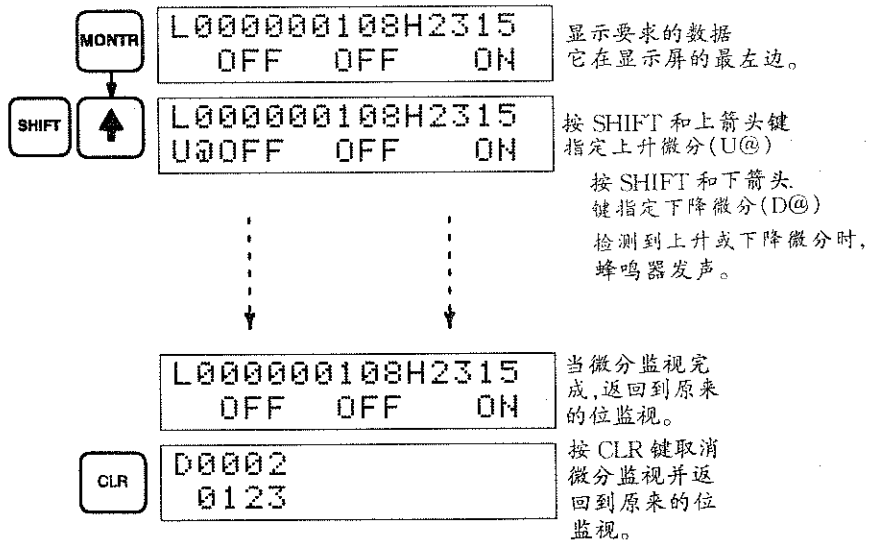
该操作用于监视在 IR、SR、AR、LR、HR 和 TC 区中各位上升沿或下降沿微分的状态。为了监视上升沿或下降沿微分状态,在位监视显示时,把要取微分的位显示在最左边,然后按 SHIFT 和向上箭头键或向下箭头键。

按 CLR 键把微分监视操作转换回到普通位监视显示。

键操作顺序



例

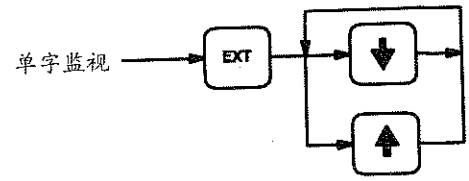


7－2－9 三字监视

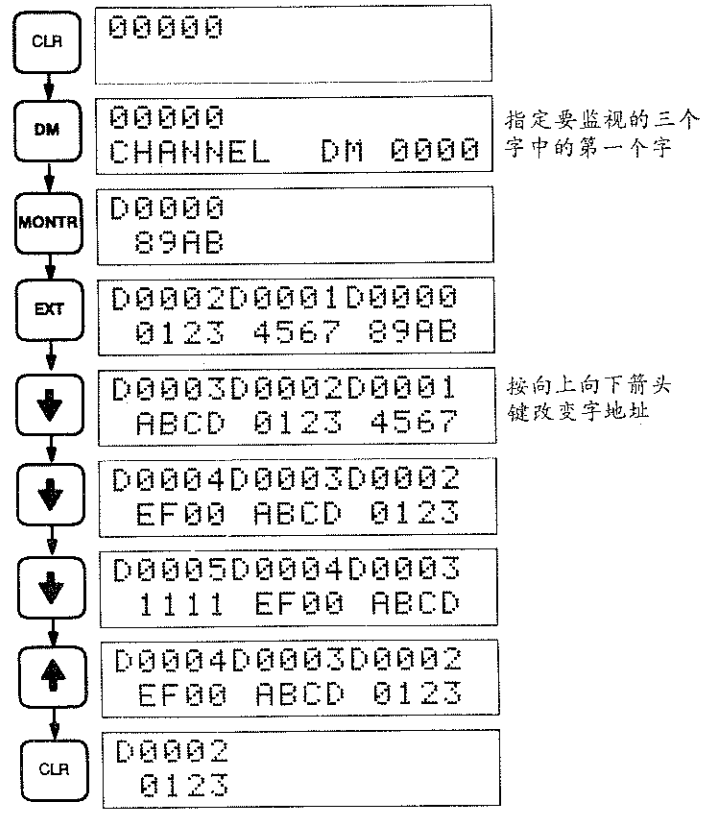
为了监视三个连续的字，指定最小编号的字，按 MONTR 键然后按 EXT 键，显示所指定的字和紧接在后面两个字的数据。

按 CLR 键把三字监视操作转换到单字显示。

键操作顺序



例

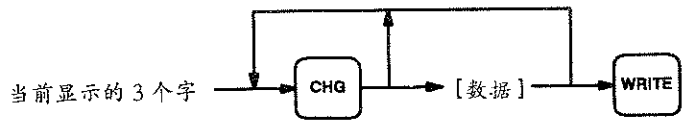


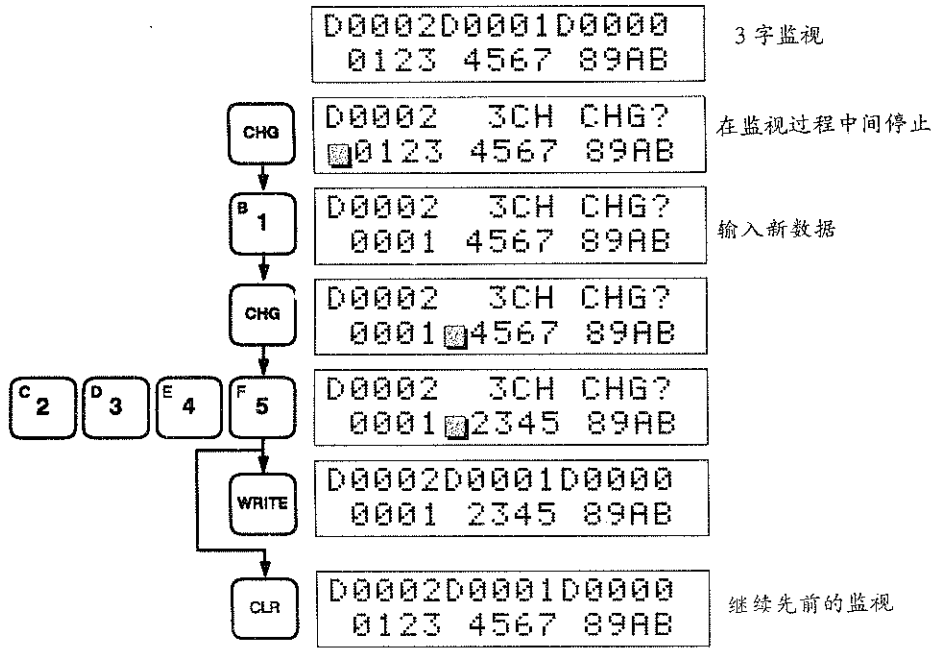
7－2－10 三字数据修改

在三字监视操作期间，该操作修改一个字的内容，闪烁的光标指标能修改该处的数据。键入新值以后，按 WRITE 键使得原来的数据被新的数据改写。如果按 WRITE 键之前按下 CLR 键则取消修改操作，然后继续前面的三字监视操作。

该操作不能用于修改 SR253～SR255。只有那些在三字监视操作中所显示的字才能被修改。

键操作顺序

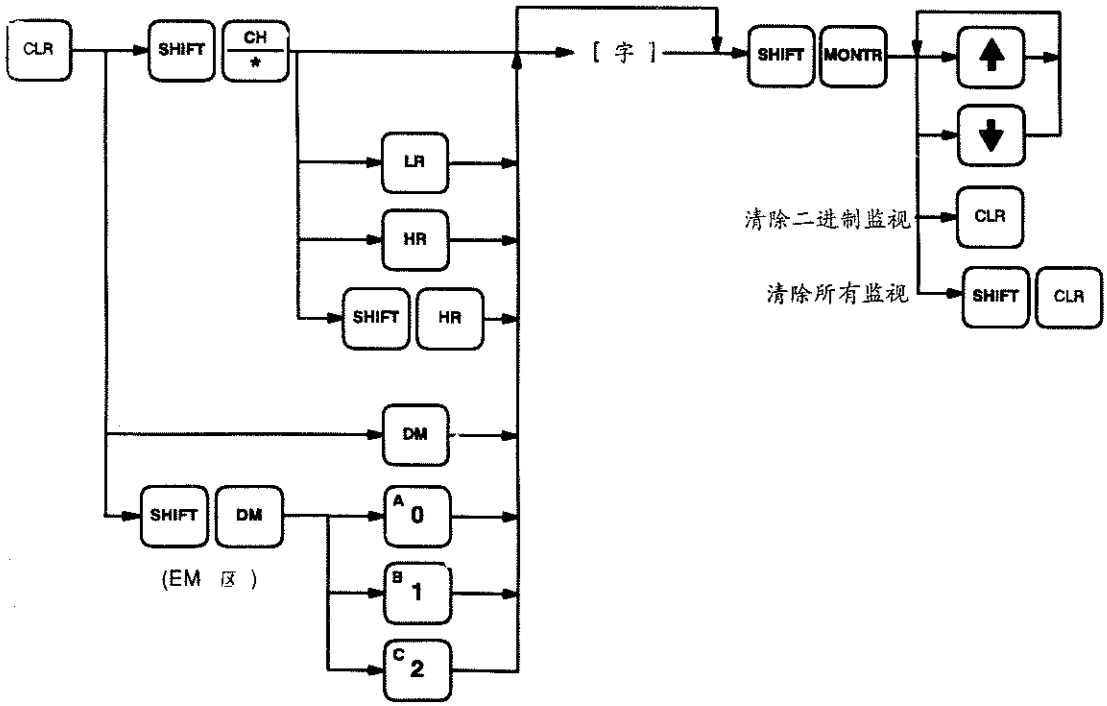




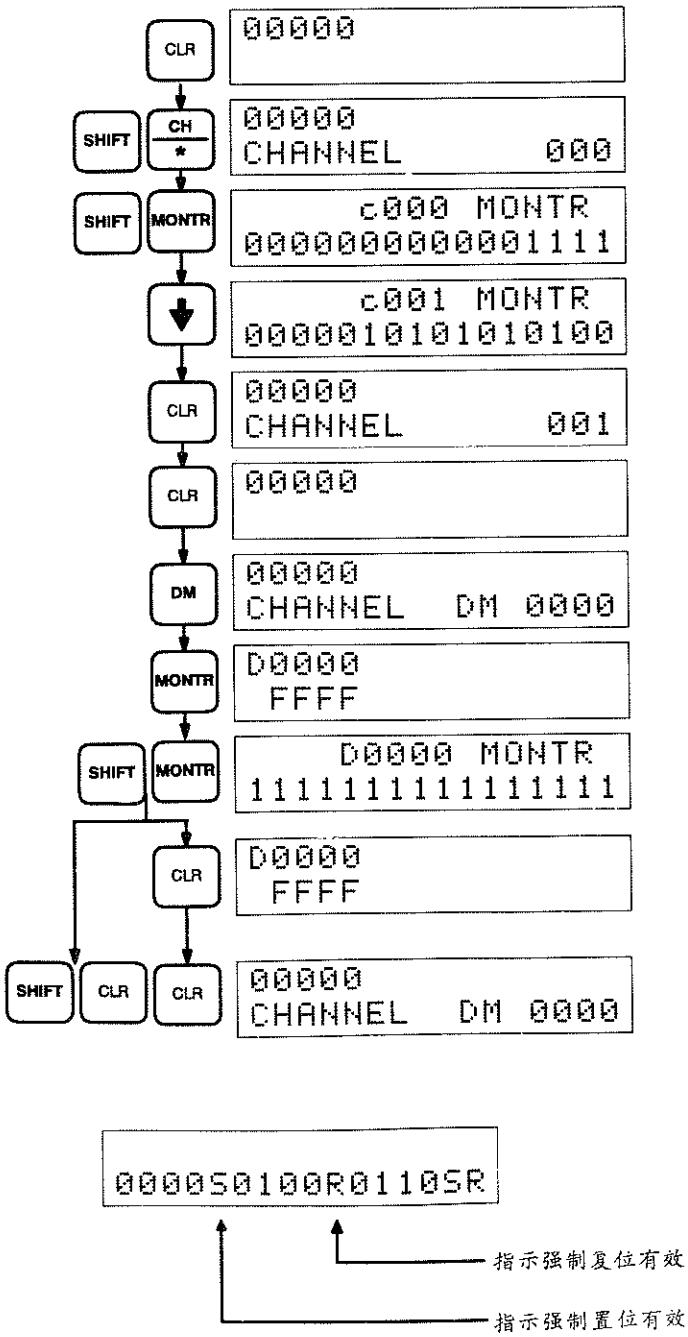
7－2－11 二进制监视

在字地址被输入以后按下 SHIFT 和 MONTR 键，可指定用二进制显示监视字内容。使用向上和向下键，递增和递减显示字的地址，可连续地监视多个字。按 CLR 键清除二进制显示。

键操作顺序


CLR
SHIFT
CH
*
LR
HR
SHIFT
HR
[字]
SHIFT
MONTR
清除二进制监视
CLR
清除所有监视
SHIFT
CLR
DM
SHIFT
DM
(EM 区)
A
0
B
1
C
2
(EM组0,1,或2)

例

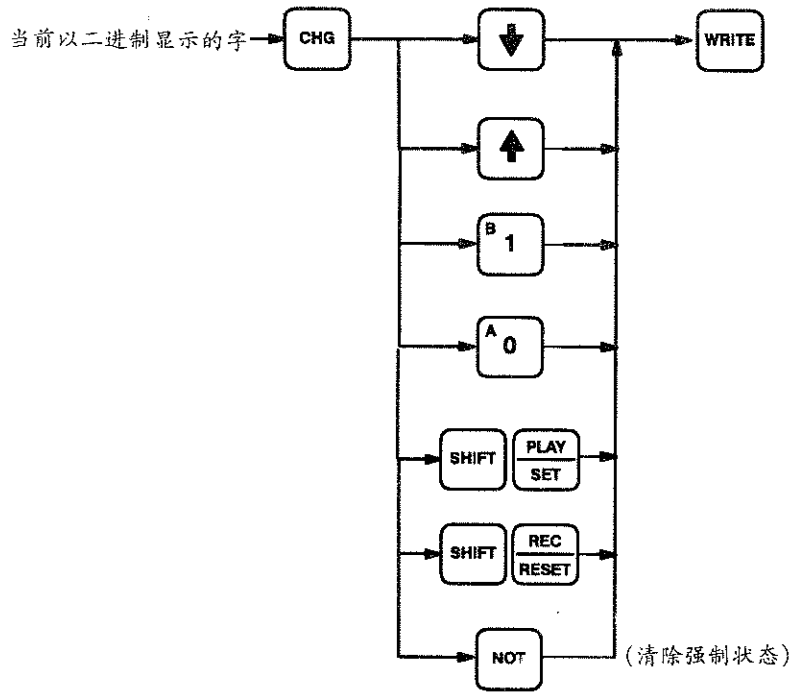


7－2－12 二进制数据修改

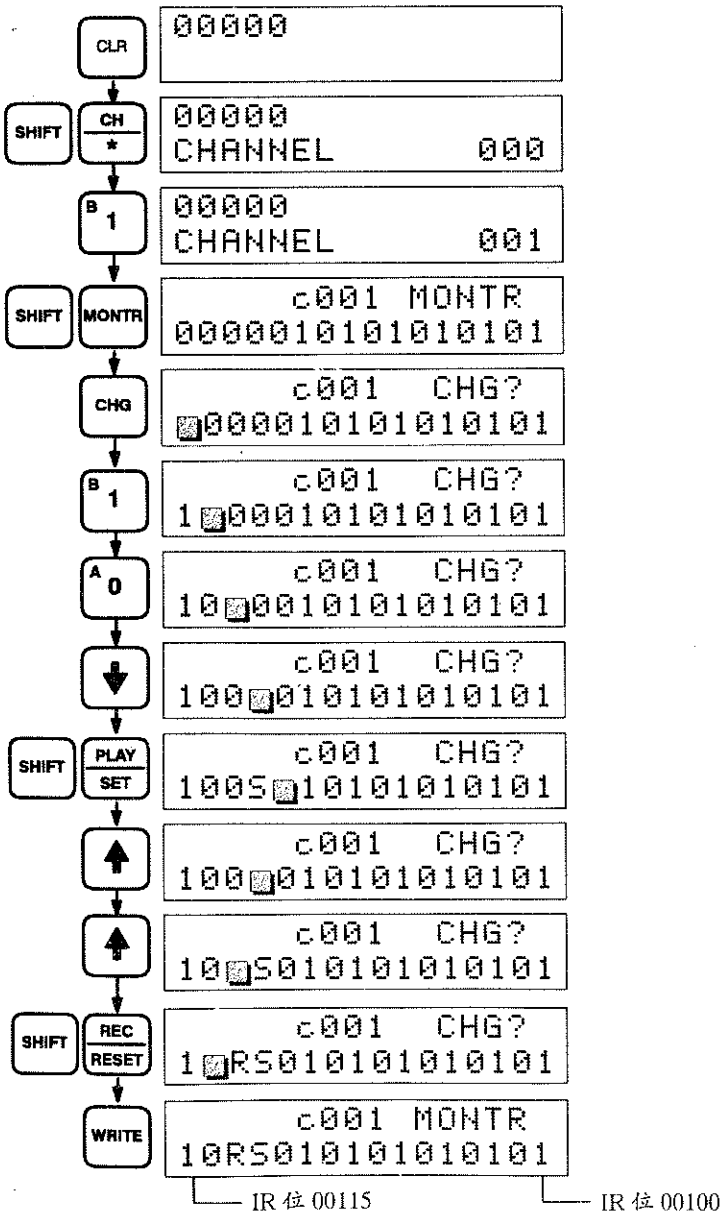
该操作对一个 IR、HR、AR、DM、EM、或 LR 字赋予一个新的 16 位二进制数值。

用向上键可把光标移动到左边，用向下键可把光标移至右边，该光标指明了能改变位的位置。在修改位定位之后，0 和 1 可作为新的位值。按 SHIFT 以及 PLAY/SET 或是 REC/RESET 键，也能够强制置位或复位。此时字母 S 或 R 将出现在位的位置上。按 NOT 键将清除强制状态，S 将变为 1；R 变为 0。在位数值改变以后，闪烁的光标将出现在改变位右边的位置上。

键操作顺序


当前以二进制显示的字
CHG
WRITE
B
1
A
0
SHIFT
PLAY
SET
SHIFT
REC
RESET
NOT
（清除强制状态）

例


CLR
00000
SHIFT
CH
*
00000
CHANNEL 000
B
1
00000
CHANNEL 001
SHIFT
MONTR
c001 MONTR
0000010101010101
CHG
c001 CHG?
0000010101010101
B
1
c001 CHG?
1 000010101010101
A
0
c001 CHG?
10 00101010101
c001 CHG?
100 0101010101
SHIFT
PLAY
SET
c001 CHG?
100S 10101010101
c001 CHG?
100 0101010101
c001 CHG?
10 S0101010101
SHIFT
REC
RESET
c001 CHG?
1 RS0101010101
WRITE
c001 MONTR
10RS0101010101
IR 位 00115
IR 位 00100

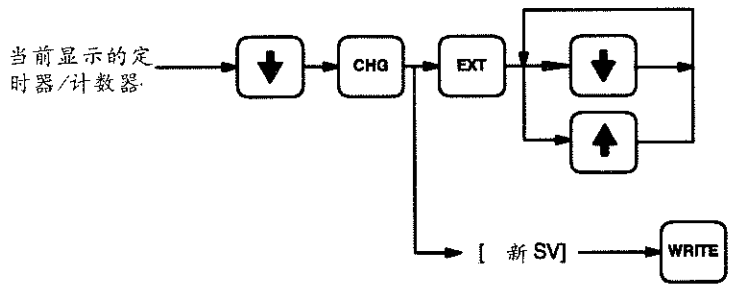
7－2－13 改变定时器/计数器 SV

有两种方法可改变定时器或计数器的 SV，即采用输入一个新值或者是递增或递减当前的 SV。它们只能用监视方式或编程方式来实现。在监视方式下，执行程序期间，可改变 SV。当 SV 为常数输入时，才能递增或递减。

为了使用其中一个方法，首先显示要改变 SV 值的定时器或计数器的地址，按向下箭头再按 CHG 键，以数字输入新值，按 WRITE 键改变 SV。或按 EXT 键接着按向上向下键递增和递减当前的 SV。在 SV 递增和/或递减时，按下 CLR 键一次 SV 值变为递增或递减后的值，并且保持按 EXT 键时出现的显示。而当按 CLR 键二次时，从 SV 新值回到原先的显示。

这个操作能用于将 SV 值从常数指定改变为地址指定。

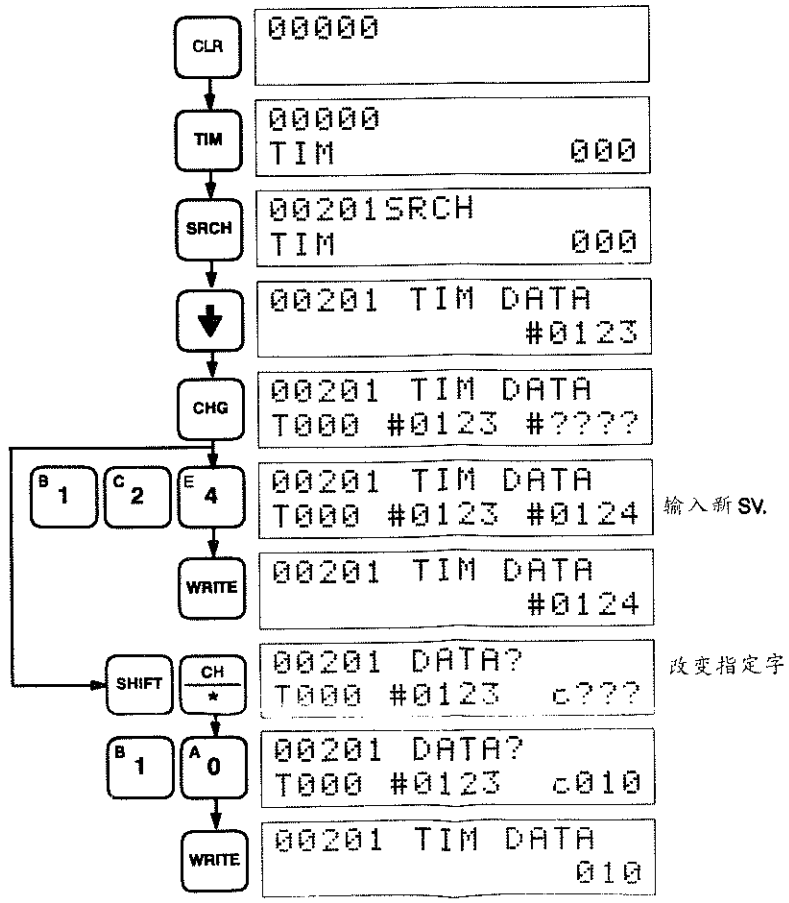
键操作顺序



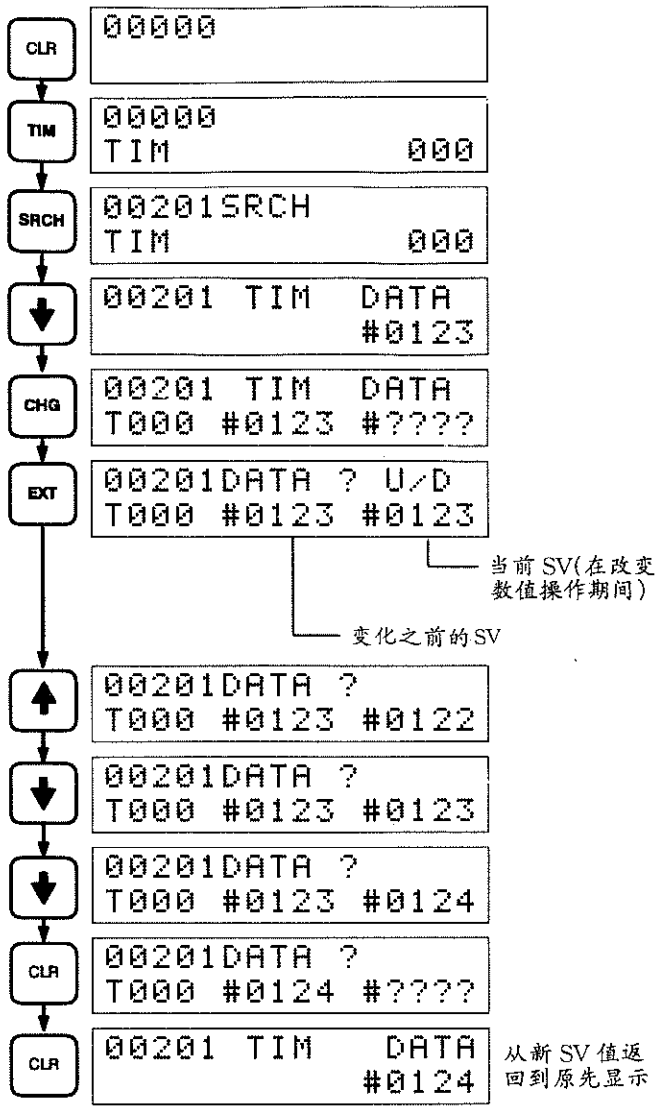
例

以下例子说明输入一个新的常数，把常数改变成地址。并且递增到一个新的常数。

输入新 SV 并改变为字设定。



递增和递减



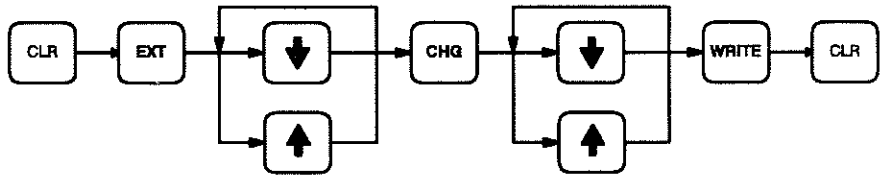
7－2－14 扩展指令功能码的配置

本操作用于读取或改变分配给扩展指令的功能码。有 18 个分配给扩展指令的功能码：17、18、19、47、48、60～69，和 87～89。一个扩展指令可分配有多个功能码。

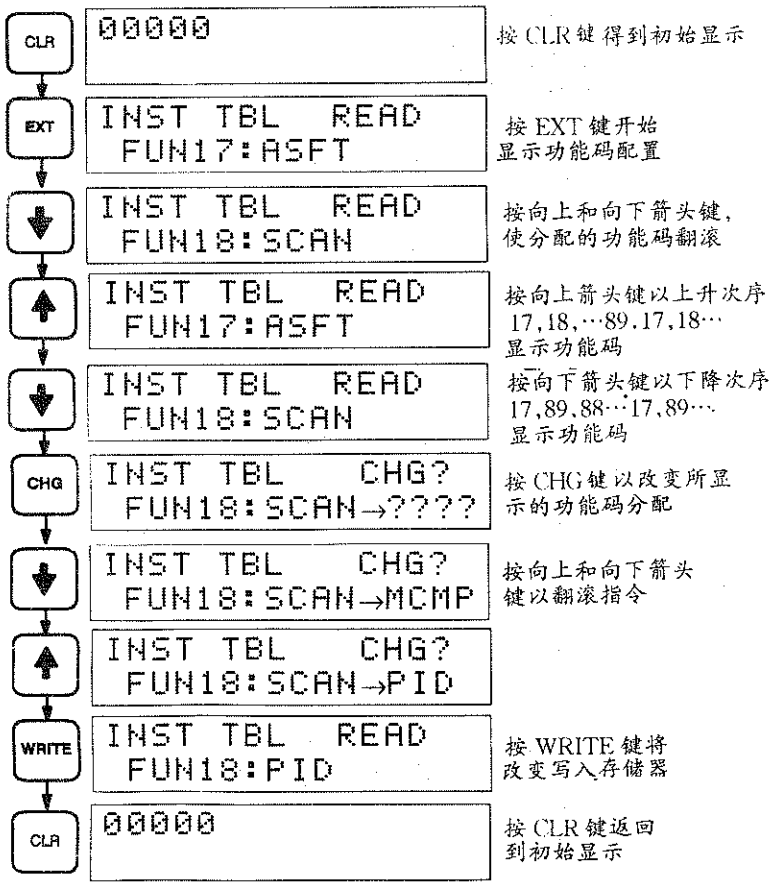
注 可采用任何方式读功能码的配置，但只能在编程方式下改变它。

只有当 DIP 开关插脚 1 为 0FF，插脚 4 为 ON 时，才能配置功能码

键操作顺序



例



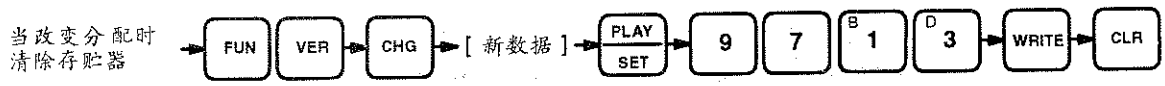
7－2－15 UM 区域分配

该操作用于分配部分 UM 区域作为扩展 DM，它只能在编程方式下执行。从梯形图程序区中扣除被分配为扩展 DM 的存储器区域。

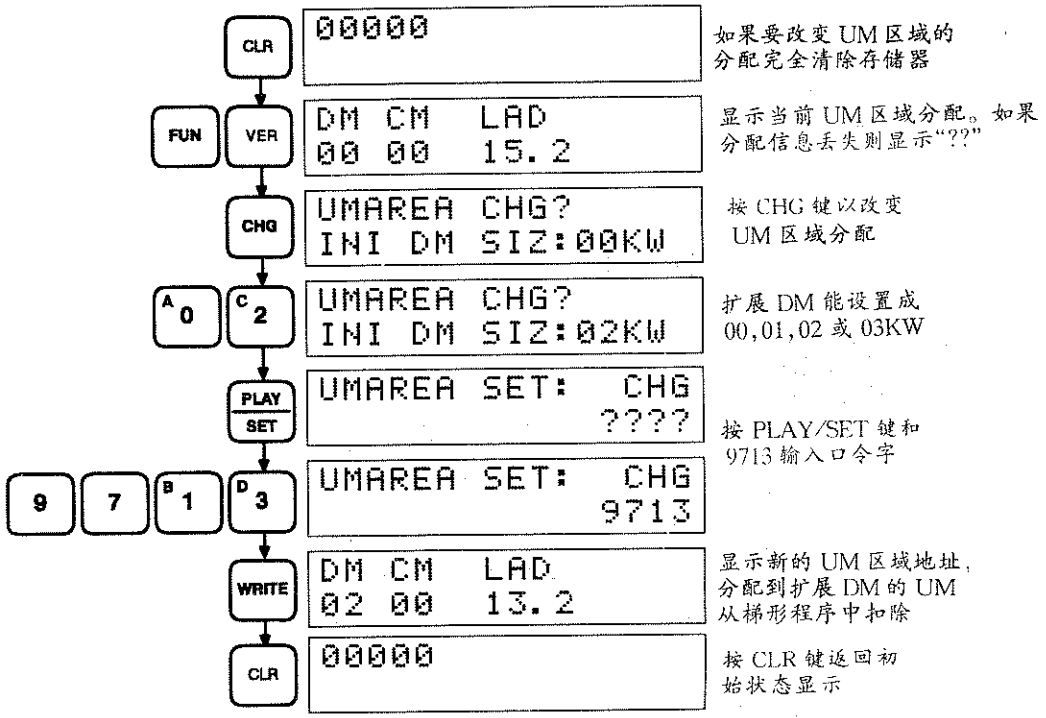
梯形图程序可使用的存储器容量取决于 CPU 内 RAM 的容量，16KW 的 RAM 可用到大约 15.2KW 的存储容量，32KW 的 RAM 可用到大约 31.2KW 的存储容量。

该操作不能用于把 UM 分配作为 I/O 注释区，只能在装有 LSS(V6 或更高)的上位计算机上，才可把 UM 分配作为 I/O 注释区。

键操作顺序



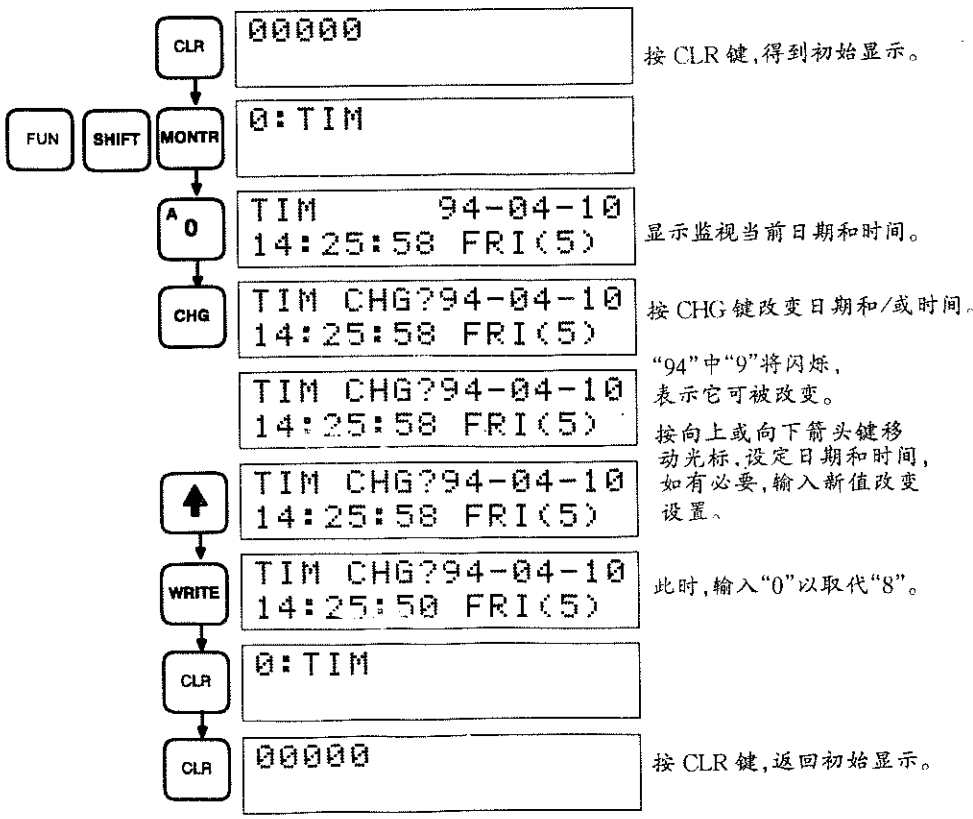
例


CLR
00000
如果要改变 UM 区域的
分配完全清除存储器
FUN
VER
DM CM LAD
00 00 15.2
显示当前 UM 区域分配。如果
分配信息丢失则显示"??"
CHG
UMAREA CHG?
INI DM SIZ:00KW
按 CHG 键以改变
UM 区域分配
A 0
C 2
UMAREA CHG?
INI DM SIZ:02KW
扩展 DM 能设置成
00,01,02 或 03KW
PLAY
SET
UMAREA SET: CHG
????
按 PLAY/SET 键和
9713 输入口令字
9
7
B 1
D 3
UMAREA SET: CHG
9713
WRITE
DM CM LAD
02 00 13.2
显示新的 UM 区域地址，
分配到扩展 DM 的 UM
从梯形程序中扣除
CLR
00000
按 CLR 键返回初
始状态显示

7－2－16 读取和设置时钟

该操作用于读取或设置 CPU 的时钟。可用任何方式读取时钟，但是只能在监视或编程方式下设置时钟。

CPU 将拒绝范围以外的数据输入。即，月份的范围是 01～02，日期的范围是 01～31，星期的范围是 00～06，或秒的范围是 00～60 秒。而且不承认不存在的日期，如 2/31。



7－2－17 扩展键盘映象

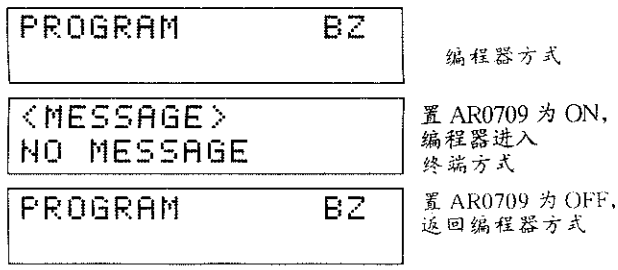
用该操作可在编程器键盘上按键控制位 SR27700～SR27909 的 ON/OFF 状态。C200HX/HG/HE 也支持键盘映象的操作，此操作控制 AR22 区的位状态。可在用任何 PC 方式下执行这些操作，但是编程器必须采用终端方式或扩展终端方式。

为了能够操作扩展键盘映象，CPU 上 DIP 开关的 6 脚和 AR0709 必须是 ON，而 AR0708 必须为 OFF。

用这种操作接通的位可用触发 AR0708 使之关断。置 AR0709 为 OFF，停止扩展键盘映象，并且把编程器从扩展终端方式转换到编程器方式。

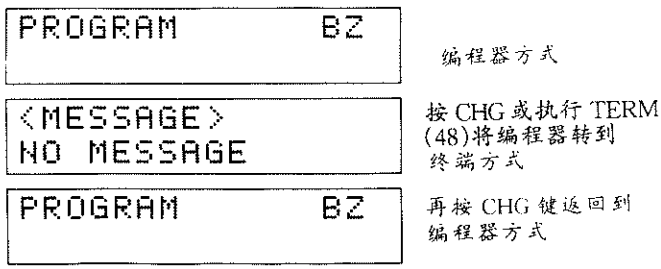
终端方式

按 CHG 键或执行程序中的 TERM(48)，可使编程器进入终端方式。这时，CPU 上 DIP 开关的 6 脚必须关断(OFF)。



扩展终端方式

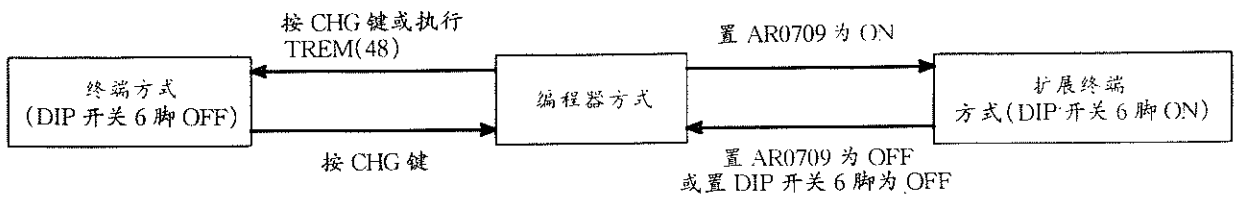
置 AR0709 为 ON，编程器进入扩展终端方式。CPU 上 DIP 开关的 6 脚必须为通(ON)。



7－2－18 键盘映象

C200HX/HG/HE 支持扩展键盘映象以及普通键盘映象，扩展键盘映象控制 SR27700～SR27909 的 41 个位状态。而普通键盘映象仅控制 AR22 中的 16 个位。当编程器采用终端方式或扩展终端方式时，按相应的编程器键，能控制这些位的状态。

下图说明在编程器方式(一般操作方式)和终端或扩展终端方式之间编程器是如何切换的。



终端方式

按 CHG 或在程序中执行 TERM(48)，编程器能进入终端方式。此时，必须置 CPU 的 DIP 开关的 6 脚 OFF。

再按 CHG 即返回编程器方式。

当编程器处于终端方式时，它能执行普通键盘映象，并且显示由 MSG(46)或 LMSG(47)输出的信息，按下编程器键盘上的 0～F，键盘映象 AR22 的 00～15 位接通。释放以后编程器键仍有一位保持为 ON。

AR0708 变为 ON 时，将关断 AR22 中所有的位；AR0708 为 ON 时，不能进行键盘映象输入。

除了键盘映象功能以外，终端方式允许将 MSG(46)和 LMSG(47)输出的信息显示在编程器上。如果编程器转回编程器方式，这些信息会被擦除。

扩展终端方式

置 AR0709 为 ON，编程器进入扩展终端方式。此时 CPU 上 DIP 开关的 6 脚必须为 ON。

置 AR0709 或 CPU 的 DIP 开关 6 脚为 OFF，编程器返回编程器方式。当编程器处于终端方式，它能执行扩展键盘映象，并且显示由 MSG(46)或 LMSG(47)输出信息。按下编程器键盘上相应的键，扩展键盘映象把 SR27700～SR27909 位转为 ON，释放编程器键以后，仍有一位保持接通(ON)。

AR0708 为 ON 时，关断 SR27700～SR27909 所有的位。

AR0708 为 ON 时，不执行扩展键盘映象输入。

除了键盘映象功能以外，扩展终端方式允许将 MSG(46)和 LMSG(47)输出信息显示在编程器上。如果编程器回到编程器方式，这些信息会被擦除。

下图给出了编程器键位置与 AR 区中各位之间的对应关系。每个键用一位，一一对应，不承认移位输入。键 0～15 对应于 AR27700～AR27715 位，键 16～31 对应于 AR27800～AR27815 位，以及键 32～41 对应于 AR27900～AR27909 位。

功能键→

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35
36	37	38	39	40	41

下表给出了编程器上各个键与 SR27700～SR27909 各位之间的对应关系。

SR 字	位	对应的键
277	00	FUN
	01	SET
	02	NOT
	03	* 1

SR 字	位	对应的键
277	04	* 2
	05	SHIFT
	06	AND
	07	OR
	08	CNT
	09	TR
	10	LR
	11	HR
	12	LD
	13	OUT
	14	TIM
	15	DM
278	00	CH / *
	01	CONT / #
	02	7
	03	8
	04	9
	05	EXT
	06	CHG

SR 字	位	对应的键
278	07	SRCH
	08	E 4
	09	F 5
	10	6
	11	PLAY / SET
	12	DEL
	13	MONTR
	14	D 1
	15	C 2

279	00	D 3
	01	REC RESET
	02	INS
	03	↑
	04	A 0
	05	* 3
	06	CLR
	07	VER
	08	WRITE
	09	↓

第八章　串行通信

本章概述了通过 RS－232C, RS－422/485, 和外设端口进行的串行通信(上位机链接, RS－232C, 1:1 链接, NT 链接和通信协议宏功能。

8－1　引言…… 390
8－2　上位机链接通信…… 390
　8－2－1　上位机链接命令简述 …… 390
　8－2－2　上位机链接通信 …… 393
　8－2－3　程序举例 …… 395
8－3　RS－232C 通信 …… 397
　8－3－1　通信帧结构 …… 398
　8－3－2　通信步骤 …… 398
　8－3－3　应用实例 …… 400
8－4　一对一 PC 链接 …… 400
8－5　NT 链接 …… 402
8－6　通信协议宏功能…… 403
　8－6－1　引言 …… 403
　8－6－2　通信板设置 …… 406
　8－6－3　通信步骤 …… 407
　8－6－4　应用实例 …… 410

8－1 引 言

装在 C200HX/HG/HE PC 的 CUP 内的 RS－232C 端口和外设端口支持下列通信功能：

· 与编程设备(如编程器，LSS，或 SSS(中文版))进行通信。
· 与个人计算机和其它外部设备进行上位机链接通信。
· 与个人计算机和其它外部设备进行 RS－232C(无通信协议)。
· 与其它 PC 进行 1:1
· 与装有 NT 链接接口的可编程终端(PT)进行 NT 链接通信。(1:1，或 1:N)。

除了上述通信功能外，在大多数 C200HX/HG/HE PC 中还可安装支持通信协议宏功能的通信板。

下表概括这些通信功能。

功能	连接设备	连接	摘要	相关指令
上位机链接	上位机或 PT	1:1 或 1:N	上位机和 PC 间提供通信。 由上位机监视 PC 操作状态和数据区内容。 使用 TXD(一)传输 PC 数据区中数据到上位机。	上位机链接命令，TXD(一)
RS－232C(无通信协议)	上位机或其它 RS－232C 设备	1:1	使用 TXD(一)和 RXD(一)管理简单的数据传送序列，如条形码读入器输入或输出至打印机。 通过指令控制 RS，CS，ER 和 DR 信号。	TXD(一)，RXD(一)
一对一链接	PC	1:1	使用 PC 中 LR 区域实现二台 PC 之间一对一链接。	…
NT 链接	PT	1:1 或 1:N	在 PC 和一个或多个 PT 之间提供数据传送。	…
通信协议宏功能	其它串行设备	1:1 或 1:N	通信协议宏功能允许用户定义单独的数据传送序列和传送信息。 可以登记多达 1000 个通信序列。提供一个支持程序，简化通信序列的生成。	PMCR(一)

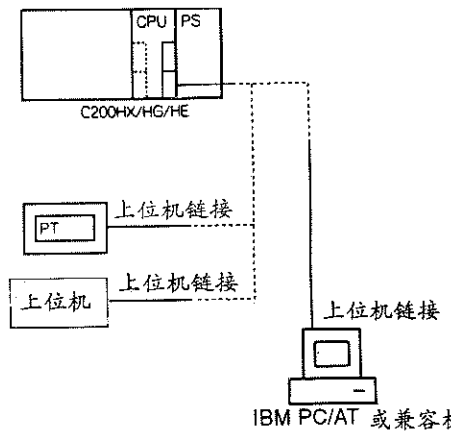
8－2 上位机链接通信

8－2－1 上位机链接命令简述

上位机链接通信用来在 PC 和上位机(个人计算机或 PT)之间传送数据，使得上位机可以使用上位机链接命令监视 PC 的运行状态和 PC 数据区的内容。还可以使用梯形程序中 TXD(一)指令将 PC 的 IOM 数据区(IR 区，SR 区，LR 区，HR 区，AR 区，定时器和计数器 PV 值，DM0000～DM6143，和 EM0000～EM6143)中数据传送到上位机。

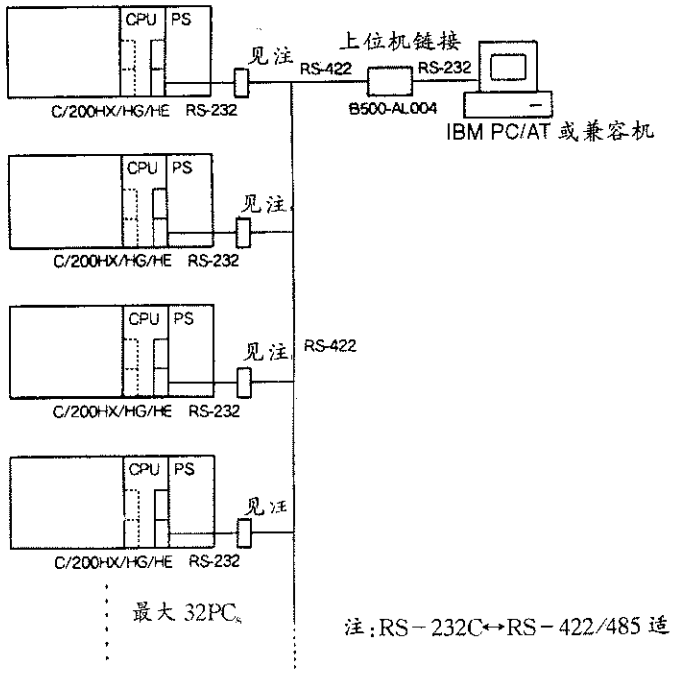
RS－232C 连接(1:1)

当使用 RS－232C 连接实现上位机链接时，上位机只能与 1 台 PC 连接(1:1 连接)。



RS－422/485 连接(1:N)

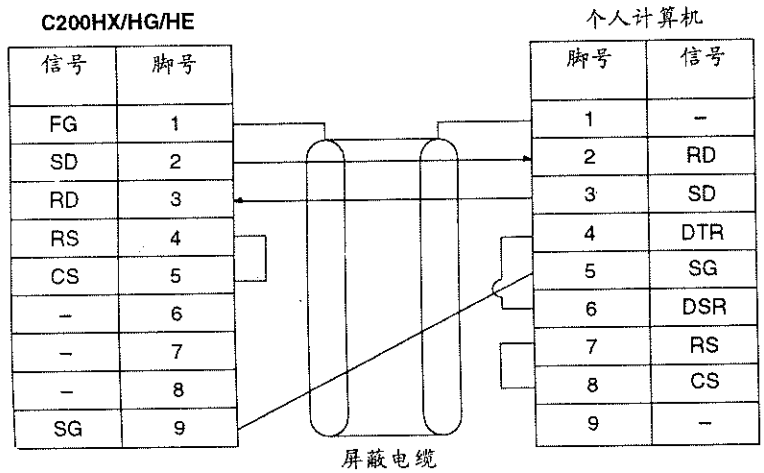
当使用 RS－422/485 连接实现上位机链接时，上位机可以连接多至 32 台 PC(1:N 连接)。



端口接线

使用下面接线图作为端口与外部设备接线的指南

下面所示的是 C200HX/HG/HE 和一台个人计算机之间连接的一个例子。



适用的连接器

下列是可选用的连接器，在 CPU 中包括有一只插头和一只插头连接器。

插头：XM2D－0901(孔型)，用于 IBM PC/AT 或兼容机(OMRON)或其它等同型号

插头连接器：XM2S－0911(OMRON)或其它等同型号

注

在 PC 上 FG 端子和计算机处接地，电阻小于等于 100Ω。

详见 C200HX/HG/HE 安装手册和您的计算机方面资料。

上位机链接参数

要启动上位机链接通信，必须预先在 PC 设置中设置下述参数

通信方式

设置通信方式为上位机链接通信(这是缺省设置)。

RS－232C 端口:置 DM6645 的位 12～15 为 0。

外设端口:置 DM6650 的位 12～15 为 0。

节点号设置

当使用 1:N 连接时,设置 00～31 之间一个节点号(唯一的)

当使用 1:1 连接时,设 PC 节点号为 00。

RS－232C 端口:置在 DM6648 的位 00～07 中。

外设端口:置在 DM6653 的位 00～07 中。

标准端口设置

标准设置或用户设置是针对 RS－232C 和外设端口的,当下述位置 0 时,使用标准设置(下面解释用户设置)。

RS－232C 端口:DM6645 中位 00～03(0:标准;1:用户)。

外设端口:DM6650 中位 00～03(0:标准;1:用户)。

下表列出标准设置。

项	设　置
起动位	1
数据长度	7
停止位	2
奇偶校验	偶
波特率	9,600 bps

用户端口设置

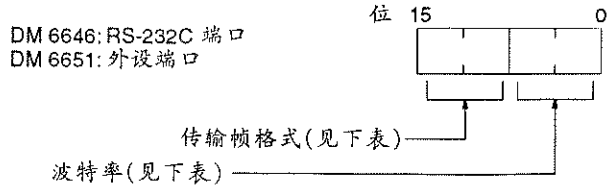
标准设置或用户设置　用于 RS－232C 和外设端口。当下述位置 1 时,使用用户设置。

RS－232C 端口:DM6645 中位 00～03(0:标准;1:用户)。

外设端口:DM6650 中位 00～03(0:标准;1:用户)。

RS－232C 端口的用户设置在 DM6646 中定义,外设端口的用户设置在 DM6651 中定义。

只有当 CPU 上 DIP 开关的脚 J 置 OFF,下述设置才有效。通信两侧的通信参数必须有相同的设置。



参　数	设置	起动位	数据长度	停止位	奇偶校验
传输帧格式	00	1	7	1	偶
	01	1	7	1	奇
	02	1	7	1	无
	03	1	7	2	偶
	04	1	7	2	奇
	05	1	7	2	无
	06	1	8	1	偶
	07	1	8	1	奇
	08	1	8	1	无
	09	1	8	2	偶
	10	1	8	2	奇
	11	1	8	2	无

参　数	设置	波　特　率
波特率	00	1,200 bps
	01	2,400 bps
	02	4,800 bps
	03	9,600 bps
	04	19,200 bps

传输延迟时间

根据 RS－232C 所连接的设备，也许必须要有一个传输时间，此时应设置传输延时以规定允许的时间量。传输延时时间设置单元为 100ms。

RS－232C 端口：设置在 DM6647，从 0000～9999(0～99.99 秒延时)。

外设端口：设置在 DM6652，从 0000～9999(0～99.99 秒延时)。

注

如果 CPU 上 DIP 开关的脚 5 置为 ON，不论 PC 设置中如何设置，将使用下面所列出的标准通信设置：

项　目	设　置
节点号	00
起动位	1
数据长度	7
停止位	2
奇偶校验	偶
波特率	9,600 bps
传输延迟时间	无

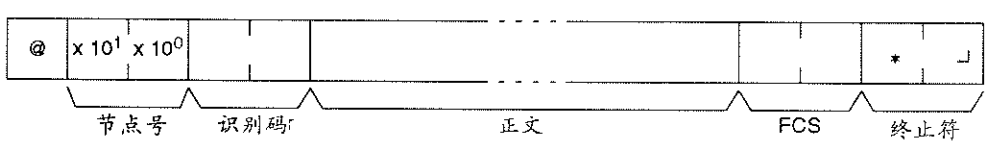
8－2－2　上位机链接通信

上位机链接通信通过在上位机和 PC 之间交换命令和应答实现的。在一次交换中传输的命令或应答数据称之为一帧，一个帧最多可包含 131 个数据字符。

下面叙述由上位机发送的上位机链接命令帧格式和由 PC 返回的应答帧格式。当 PC 接收到从上位机发来的 ASCII 码命令时自动返回 ASCII 码应答。上位机必须有一个能控制命令和应答的传送和接收的程序。

命令帧格式

从上位机发送一个命令时，按下示的格式排列命令数据。



识别码和正文取决于传输的上位机链接命令。当传送一个组合命令时，还将有第二个识别码(子识别码)。

FCS(帧检查顺序)码由上位机计算，并设置在命令帧中，本节后面将叙述 FCS 的计算。

命令帧可以有最多 131 个字符长。一个等于或大于 132 字符的命令必须分成若干帧。命令分段，使用回车定界符(┘，CHR $ (13))，而不是终止符。终止符必须用在最后帧的末尾。

在对执行写操作的命令(如 WR，WL，WC 或 WD)分段时，应注意不要将写入单独一个字的数据分在不

同帧中,还有帧的分段应和字与字之间的分段一致。

项　目	功　能
@	@符号必须置于每个命令的开头。
节点号	按该节点号辨识 PC,它设置在 PC 设置的 DM6648 中。
标识码	设置 2 字符的命令代码。
正文	设置命令参数。
FCS	设置 2 字符的帧检查顺序码。
终止符	设置"＊"和回车(CHR＄(13))两字符,表示命令结束。

应答帧格式

来自 PC 的应答按下示格式返回。请准备一个程序,翻译并处理应答数据。



识别码和正文取决于接收到的上位机链接命令。

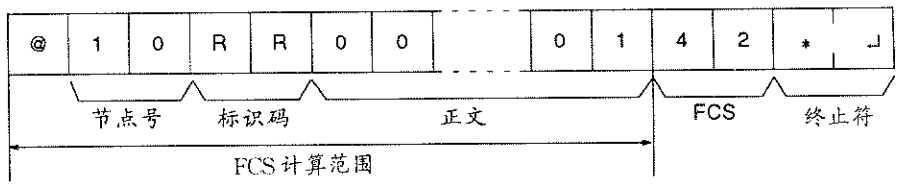
结束码表示命令完成的状态(即,是否有错误发生。)

当应答超过 132 字符,它必须分成若干帧。在每个帧的末尾将自动设置一个定界符(□,CHR(13)),代替终止符。终止符必须设置在最后帧的末尾。

项　目	功　能
@	@符号必须置于每个应答的开头。
节点号	设在 PC 设置的 DM6648 中的 PC 节点号。
标识码	返回 2 字符的命令代码
正文	返回命令的结果
FCS	返回 2 字符帧检查顺序码
终止符	"＊"和回车(CHR＄(13))两字符,表示应答结束。

FCS(帧检查顺序)

当传送一个帧时,在定界符或终止符前面安排一个 FCS 码,以检查传送时是否存在数据错误。FCS 是一个转换成 2 个 ASCII 字符的 8 位数据。这 8 位数据为从帧开始到帧正文结束(即 FCS 之前)所有数据执行"异或"操作的结果。每次接收到一帧,计算 FCS,与帧中所包含的 FCS 作比较从而检查帧中间的数据错误。



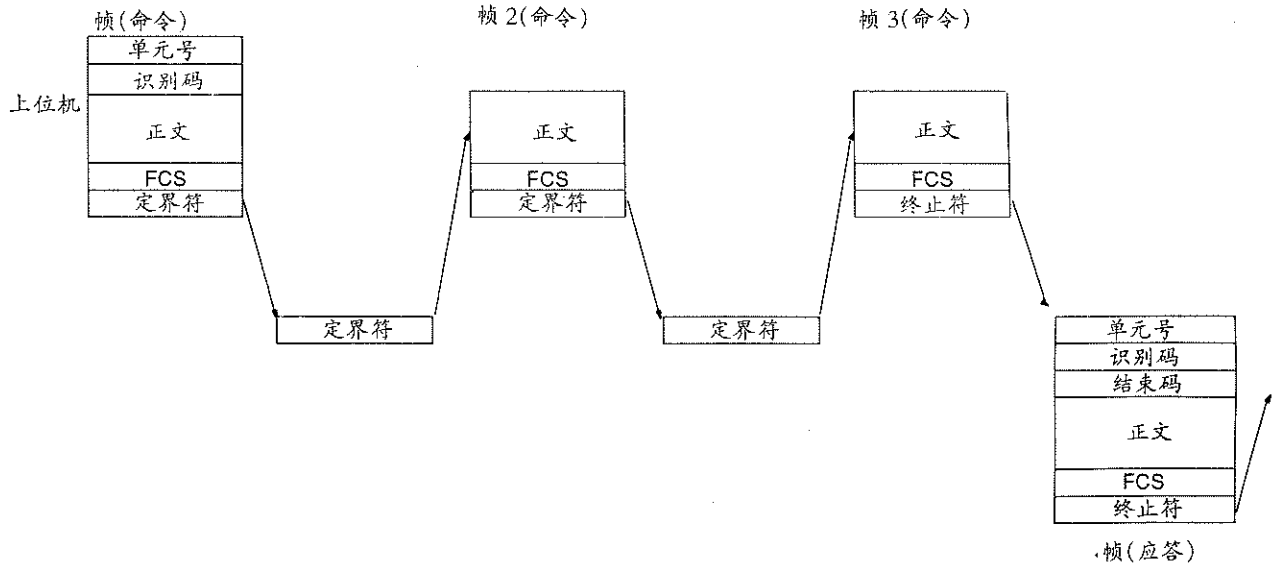
	ASCII 码		
@	40	0100	0000
		EOR	
1	31	0011	0001
		EOR	
0	30	0011	0000
		EOR	
R	52	0101	0010
		⋮	
1	31	0011	0001
计算结果		0100	0010
		i	i
		4	2

转换为十六进制作 ASCII 字符处理

通信顺序

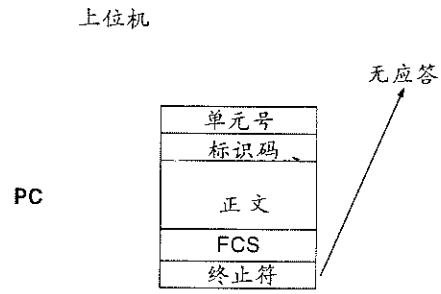
发送帧的权利称为“传输权”。具有传输权的单元是可以在某一给定时间发送帧的单元。每次发送 3 帧之后，这个传输权在上位机和 PC 之间来回交换。下面叙述一个上位机和 PC 之间通信顺序的例子。

- 上位机在命令的第一帧末端设定界符并送出该帧。
- 当 PC 接收到定界符，返回相同的定界符给上位机。
- 上位机收到来自 PC 的定界符之后，传送下一帧。
- PC 在第一应答帧的末端设定界符，传送帧。
- 当上位机收到定界符，返回相同的定界符给 PC。
- 接收到上位机来的定界符，PC 发送下一帧。
- 长传送利用这样方式交换定界符进行管理。最后一帧以终止符结尾。



使用 TXD(—)指令

TXD(－)指令可用来传送 PC 数据区中数据到上位机。上位机不传送应答。如果一个上位机链接命令的应答正返回给上位机时要执行 TXD(－)指令，必须在传送应答之后才可执行 TXD(－)指令。



8－2－3 程序举例

命令传输：在上位机，必须准备下述类型的程序以接收数据。

在上位机链接正在执行读取命令，从 PC 读数据的同时，这个程序在计算机可以读入并显示从 PC 接收的数据。

```
10  'C200HX/HG/HE SAMPLE PROGRAM FOR EXCEPTION
20  CLOSE
```

```
30  CLS
40  OPEN "COM:E73"AS#1
50  *KEYIN
60  INPUT"DATA－－－－－－－－－",S$
70  IF S$ ="  "THEN GOTO 190
80  PRINT"SEND DATA=";S$
90  ST$ =S$
100  INPUT"SEND OK? Y or N? =",B$
110  IF B$ ="Y"THEN GOTO 130 ELSE GOGO *KEYIN
120  S$ =ST$
130  PRINT #1,S$                          '发送命令给 PC
140  INPUT #1,R$                          '接收 PC 来的应答
150  PRINT"RECV DATA=";R$
160  IF MID$(R$,4,2)="EX"THEN GOTO 210'识别 PC 来的命令
170  IF RIGHT$(R$,1)<>"*"THEN S$ ="  ";GOTO 130
180  GOTO *KEYIN
190  CLOSE 1
200  END
210  PRINT"EXCEPTION!! DATA"
220  GOTO 140
```

FCS 处理程序举例

该例子是一个对上位机接收到的帧进行 FCS 检查的 BASIC 子程序。

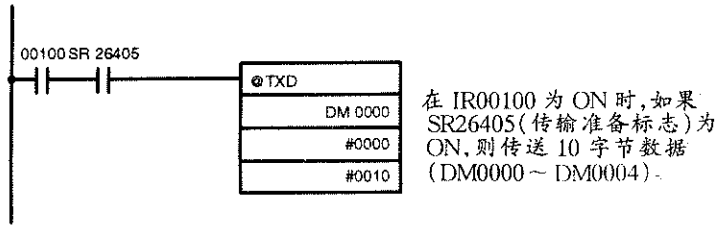
```
400  *FCSCHECK
410  L=LEN(RESPONSE$   '……传送和接收到的数据
420  Q=0;FCSCK$ ="  "
430  A$ =RIGHT$(RESPONSE$,1)
440  PRINT RESPONSE$,AS,L
450  IR A$ ="*"THEN LENGS=LEN(RESPONSE$)－3
                ELSE LENGS=LEN(RESPONSE$)－2
460  FCSP$ =MID$(RESPONSE$,LENGS+1,2) '…接收到的 FCS 数据。
470  FOR I=1TO LENGS '……FCS 中字符数
480  Q=ASC(MID$(RESPONSE$,I,1))XOR Q
490  NEXT I
500  FCSD$ =HEX$(Q)
510  IF LEN(FCSD$)=1 THEN FCSD$ ="0"+FCSD$   'FCS 结果
520  IF  FCSD$ <>FCSD$ THEN FCSCK$ ="ERR"
530  PRINT"FCSD$ =";FCSD$,"FCSP$ =";FCSP$,"FCSCK$ =";FCSCK$
540  RETURN
```

注

1. 正常接收数据包括 FCS,定界符或终止符等。当发生传输错误,可能不再包含 FCS 或其它数据,系统编程时要考虑这种可能性。
2. 该程序例子中,在 RESPONSE$ 中未包含回车(CR)码(CHR$(13))。如加入 CR 码,要修改 430 和 450 行。

TXD(—)应用举例

假设所有 PC 设置均为缺省值(即:RS－232C 用于上位机链接方式,节点号为 00,采用标准通信参数)。

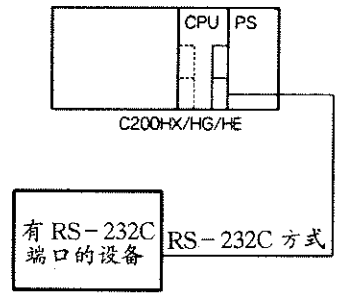


8－3　RS－232C 通信

本节介绍 RS－232C 通信,RS－232C 通信可使用 TXD(－)和 RXD(－)指令将数据输出到打印机,从条形码读入器输入数据或向有 RS－232C 端口的其它设备发送上位机链接命令。

RS－232C 连接

下图表示在 RS－232C 方式(无通信协议)下使用 RS－232C 端口时的 RS－232C 连接。



PC 设置参数

要启动通过 RS－232C 或外设端口的 RS－232C 方式通信,必须预先在 PC 设置中设置下述参数。

通信方式

设置通信方式为 RS－232C 方式

RS－232C 端口:DM6645 的位 12～15 置 1。

外设端口:DM6650 的位 12～15 置 1。

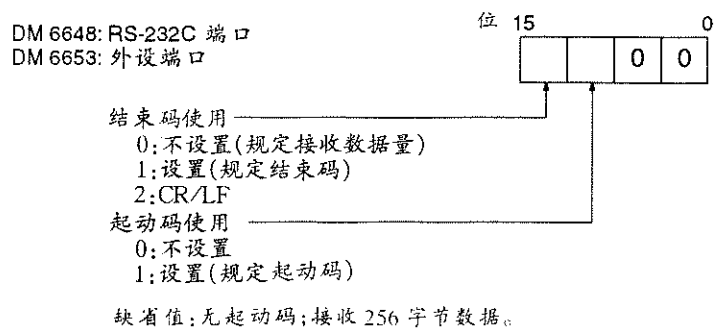
标准端口设置

标准设置或用户设置用于 RS－232C 和外设端口。关于标准通信设置详见 P392。

用户端口设置

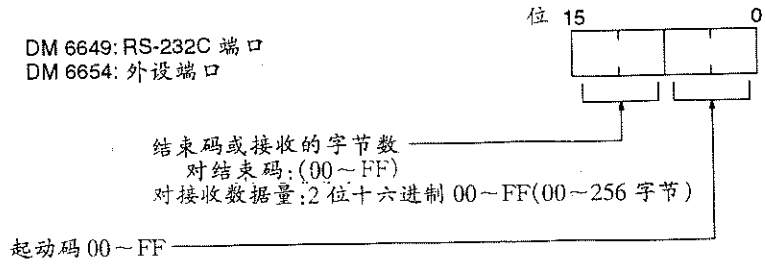
标准设置或用户设置用于 RS－232C 和外设端口。关于用户通信设置详见 P392。

允许起动码和结束码



指定在数据开始处是否设起动码，在结束是否设结束码。可以规定完成接收操作之前要接收的字节数，代替设置结束码。两种代码和接收字节数设置在 DM6649 或 DM6654。

设置起动码，结束码和接收数据量



缺省值：无起动码，接收 256 字节数据。

8－3－1 通信帧结构

执行一条 TXD(－)或 RXD(－)指令，可以发送最多 259 字节数据(包括起动码和结束码)。当有二个或更多起动码时，将使用第一个起动码。同样，有二个或更多结束码时，将使用第一个结束码。

避免使用常用字符作为结束码。如果要截短数据传输，因为结束码插入在传输数据当中，请使用 CR 和 LF 作为结束码。

· 无起动码和结束码

数据(规定字节数)

· 仅有起动码

ST	数据(规定字节数)

· 仅有结束码

数据(等于或小于 256 字节)	ED

· 有起动码和结束码

ST	数据(等于或小于 257 字节)	ED

· 规定 CR，LF 为结束码

数据(等于或小于 258 字节)	CR	F

· 有起动码(00～FF)和结束码(CR，LF)

ST	数据(等于或小于 259 字节)	CR	LF

8－3－2 通信步骤

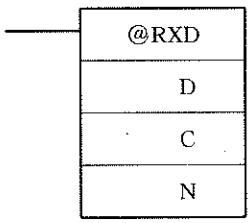
下面简要叙述 TXD(－)和 RXD(－)。

传输(TXD(－))

详见 5－27－2“发送指令－TXD(－)”。为了保证在进行传送之前，指定端口的传送准备标志为 ON，将该标志包含在 TXD(－)指令的执行条件中。

1．检查 SR26405(RS－232C 端口发送准备标志)，SR26413(外设端口发送准备标志)，SR28305(通信板端口 A 发送准备标志)，或 SR28313(通信板端口 B 发送准备标志)是否为 ON。

2．用 TXD(－)指令发送数据。(仅当位 12～15 置为 0，位 08～11 才有效)。



S:传送数据首字地址
C:控制数据
位 00～03
0:高位字节为首
1:低位字节为首
位 04～07
0:正常数据传送操作。
1:传输数据最左面字的位 15 状态反映到相应端口的 RTS 上。
2:传输数据最左面字的位 15 状态反映到相应端口的 DTR 上。
3:传输数据最左面字的位 14 和 15 反映到相应端口的 RTS 和 DTR 上。
位 08～11
0:内置 RS－232C 端口。
1:通信板端口 A
2:通信板端口 B
位 12～15
0:RS－232C 端口
1:外设端口
2:上位机链接单元＃1
3:上位机链接单元＃2
N:传输字节数(4 位 BCD 码),0000～0256(起动和结束位)

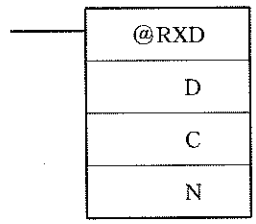
3. 从开始执行指令到数据传输结束这段时间,发送准备标志(SR26405, SX26413, SR28305, 或 SR28313)保持 OFF。数据传输结束时,它变回 ON。

接收(RXD(—))

详见 5－27－1"接收指令－RXD(－)"。

1. 检查 SR26406(RS－232C 端口接收结束标志)或 SR26414(外设端口接收结束标志)是否为 ON。

2. 用 RXD(－)指令接收数据。(仅当位 12～15 置为 0 时位 08～11 才有效。)



D:存贮接收数据的首字地址
C:控制数据
位 00～03
0:高位字节为首
1:低位字节为首
位 04～07
0:正常数据接收操作
1:读相应端口的 CTS 状态,并写入接收数据,最左面字的位 15。
2:读相应端口的 DSR 状态,写入接收数据最左面字的位 15。
3:读相应端口的 CTS 和 DSR 状态,写接收数据最左面字的位 14 和 15
位 08～11
0:内置 RS－232C 端口
1:通信板端口 A
2:通信板端口 B
位 12～15
0:RS－232C 端口
1:外设端口
N:存贮字节数(4 位 BCD 码),0000～0256(起动位和结束位)

3. 当执行 RXD(－)指令时,接收到的数据传送到指定的字中(不含起动码和结束码),同时接收完成标志置 OFF。接收起动和结束说明如下:

起动:如果不设起动码,连续接收。

如果设置起动码,当接收到起动码,开始接收。

结束:当接收到结束码或已经接收到 259 个数据字节,接收结束。

4. 读取接收的数据而产生的状态存贮在 SR 区域。检查操作是否顺利完成。这些位的状态在每次执行 RXD(－)指令时被复位。

RS－232C 端口	外 设 端 口	错 误
SR26400～SR26403	SR26408～SR26411	通信端口错误代码(1 位数 BCD 码) 0:正常完成 1:奇偶校验错误 2:帧错误 3:运行错误
SR26404	SR26412	通信错误标志
SR26407	SR26415	接收操作错误(接收完成之后,用 RXD(一)指令读入数据之前,接收到连续数据。
SR265	SR266	接收字节数(不包括起动位和结束位)

注

复位 RS－232C 端口(即恢复初始状态),SR25209 置 ON。复位通信板端口 A,SR28900 置 ON。复位通信板端口 B,SR28901 置 ON。复位之后,这些位自动回到 OFF。

8－3－3 应用实例

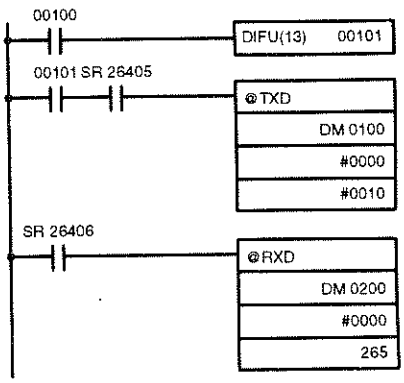
该实例展示一个程序,它使用 RS－232C 端口在 RS－232C 方式下发送 10 个字节的数据(DM0100～DM0104)到计算机,并将从计算机接收来的数据存贮在以 DM0200 为首址的 DM 数据区中。执行该程序之前必须进行下述 PC 设置:

DM6645:1000(RS－232C 方式下 RS－232C 端口,标准设置)

DM6648:2000(无起动码;结束码 CR/LF)

假定其它 PC 设置均为缺省值。上位机必须有相同的通信设置,并且有一个程序以接收从 PC 传送来的数据。

数据 3454 存贮在 DM0100－DM0104 的每个字中。



在 IR00100 为 ON 时,如果 SR26405(发送准备标志)为 ON,发送 10 个字节数据(DM0100～DM0104),高位字节为首。

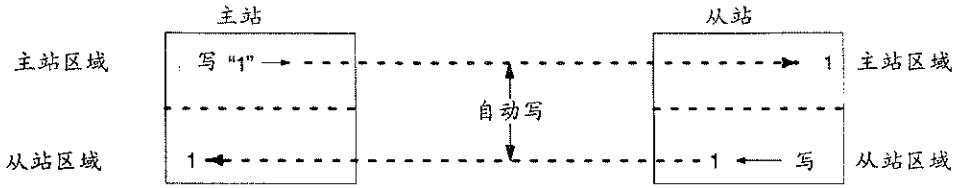
当 SR26406(接收准备标志)变 ON,由 SR265 指定字节数的数据从 PC 接收缓冲器中读入,存贮到 DM0200 开始的存储器中,高位字节为首。

这个数据是:"34543454345434543454 CRLF"

8－4 一对一 PC 链接

如果二台 PC 通过各自 RS－232C 端口连接构成一对一链接,它们可以共享公共 LR 区域。当二台 PC 一对一链接,其中一台为主站,而另一台为从站。

如下图所示,当在被链接的一个单元的 LR 区域一个字内写入数据,该数据将同样自动地写到另一单元的相同字中。每台 PC 指定可写入的字和由另一台 PC 写入的字。每台 PC 可以读,但不可写入由另一台 PC 写的字。



接线

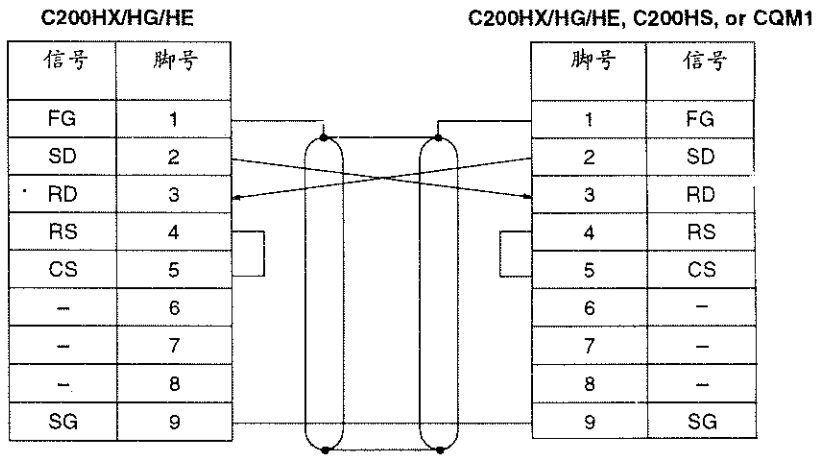
用下列连接器按下图连接电缆。

适用的连接器

下列是可选用的连接器，在 CPU 中包括有一只插头和一只插头连接器。

插头：XM2A－0901(OMRON)或其它等同型号。

插头连接器：XM2S－0911(OMRON)或其它等同型号。



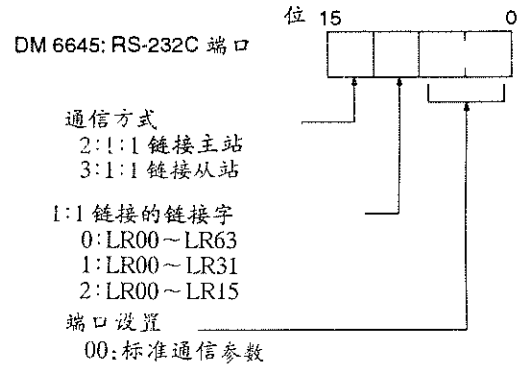
注

在 PC 上 FG 端子和计算机处接地，电阻小于等于 100Ω。

PC 设置

使用 1:1 链接，只须设置通信方式和链接字。

设置一台 PC 机的通信方式为 1:1 链接主站，另一台 PC 为 1:1 链接从站，然后在指定为主站的 PC 上设置链接字。位 08～11 仅对 1:1 链接的主站有效。



下表列出根据主站，从站和链接字设定，每台 PC 使用的链接字

DM6645 设置	LR00～LR63	LR00～LR31	LR00～LR15
主站字	LR00～LR31	LR00～LR15	LR00～LR07
从站字	LR32～LR63	LR16～LR31	LR08～LR15

通信步骤

如果主站和从站设置正确，PC 合上电源时，一对一链接自动起动。

应用实例

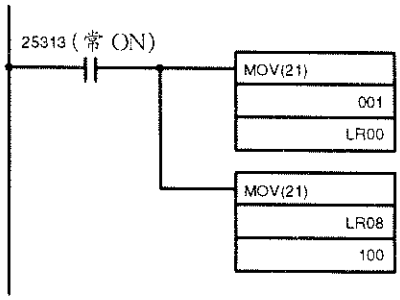
该实例展示一个程序用于检查使用 RS－232C 端口执行一对一链接的状态。执行程序之前，设置下述 PC 设置参数：

主站：DM6645：3200（一对一链接主站，链接字：LR00～LR15）

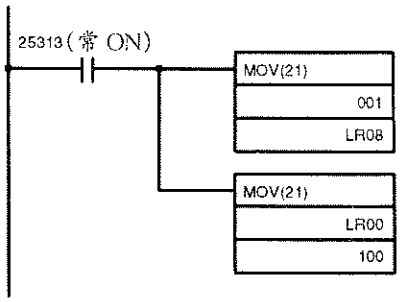
从站：DM6645：2000（一对一链接从站）

当在主站和从站执行下述程序，每个单元 IR001 的状态反映在另一单元的 IR100 上。IR001 是输入字，IR100 是输出字。

在主站中



在从站中



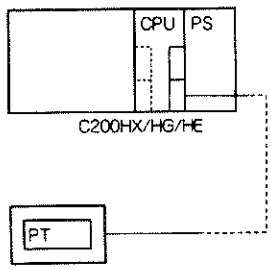
8－5 NT 链接

一对一 NT 链接是将 PC 的 RS－232C 端口与可编程终端（PT）上的 RS－232C 端口连接建立起来的，使用 NT 链接命令。

一对 N 的 NT 链接是通过 RS－422/485 电缆连接 PC 和可编程终端（PT）建立起来的，使用 NT 链接命令。

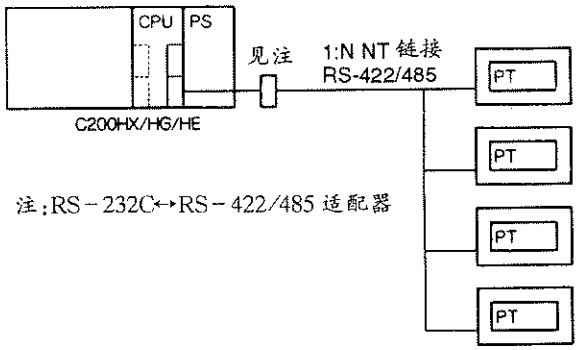
一对一 NT 链接

下图给出一对一 NT 链接的连接



一对 N NT 链接

下图给出一对 N NT 链接的连接图。只要 PC 不是 C200HE－CPU □□－E，可以最多连接 8 个可编程终端。用 C200HE－CPU□□－E，最多可连接 4 个可编程终端（包括通过通信板连接的 PT）。



注

RS－232C↔RS－422/485 适配器

PC 设置

当建立一个 NT 链接时，需设置如下：

链　接	端　口	设　置
一对一	内置 RS－232C 端口	DM6645 中位 12～15 置 4。
	通信板端口 B	DM6650 中位 12～15 置 4。
	通信板端口 A	DM6655 中位 12～15 置 4。
一对 N	内置 RS－232C 端口	DM6655 中位 12～15 置 5。
	通信板端口 B	DM6550 中位 12～15 置 5。 MD6550 中位 08～11 置最大节点号(1～7)。
	通信板端口 A	DM6555 中位 12～15 置 5。 DM6555 中位 08～11 置最大节点号(1～7)。

应用

实际 NT 链接的应用详见为 NT 链接接口单元提供的资料。

8－6　通信协议宏功能

本节说明如何使用通信协议宏功能

8－6－1　引言

通信协议宏功能是一种通信协议，控制同各种通信设备和装有 RS－232C 或 RS－422/485 端口的通用器件的数据传送。用户可以用 OMRON 通信协议支持软件很容易地修改数据传送步骤(通信序列)并在梯形图程序中使用 PMCR(－)指令执行该通信序列。

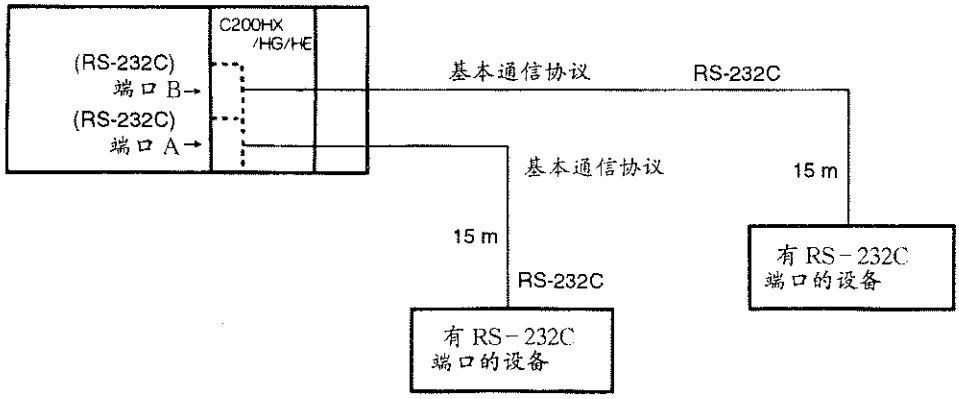
通信板已装有七种通信步骤，这些标准序列可以按常规使用或进行修改以符合特定应用的需求。

注

关于通信板详见“通信板操作手册”，有关通信协议支持软件参阅“通信协议支持软件操作手册”。

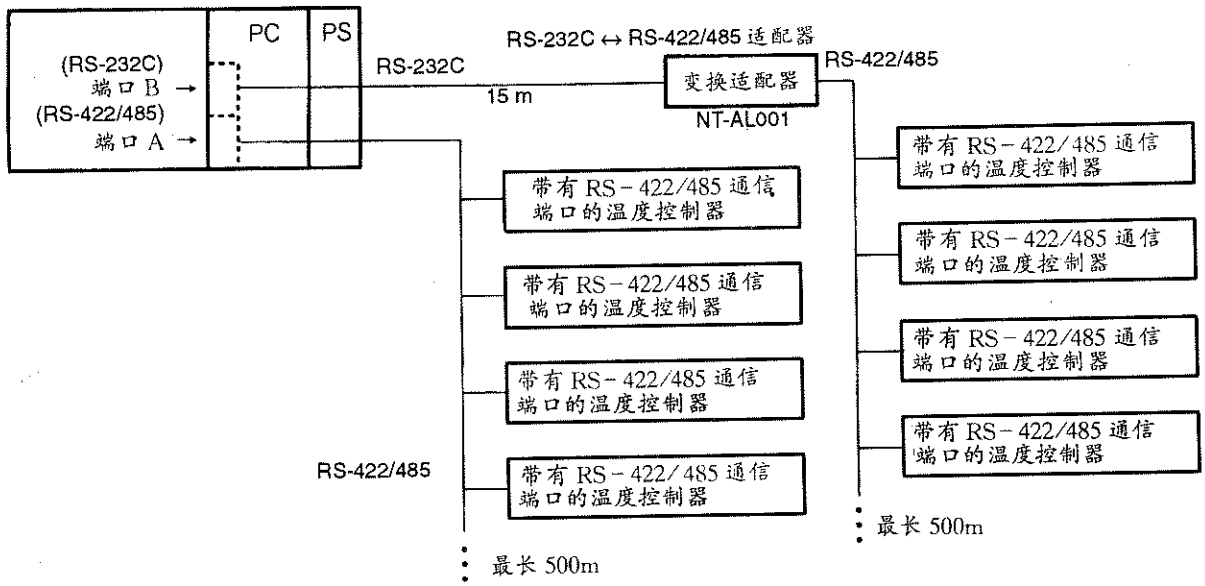
RS－232C 连接(一对一)

采用 RS－232C 连接，只可连接一个设备，RS－232C 电缆最长 15 米。



RS－422/485 连接(一对 N)

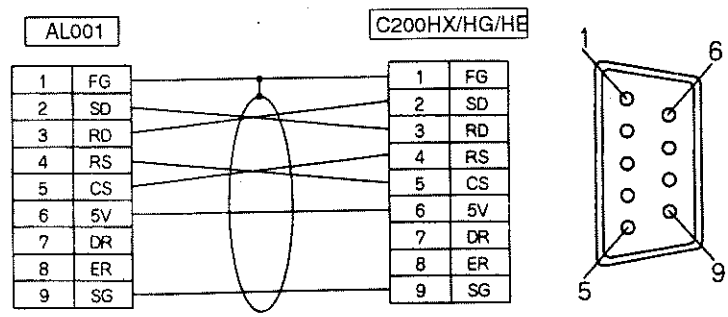
RS－422/485 连接允许连接 2 个或更多的设备，电缆长度最长可达 500 米。RS－422/485 连接还可用于远距离的一对一连接。



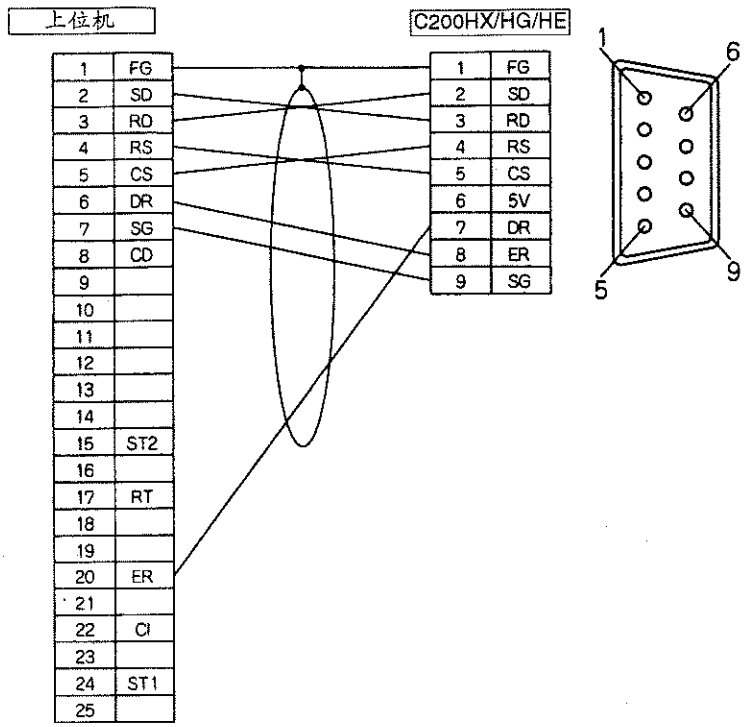
连接电缆接线

下图表示用于通信协议宏功能通信的电缆接线。

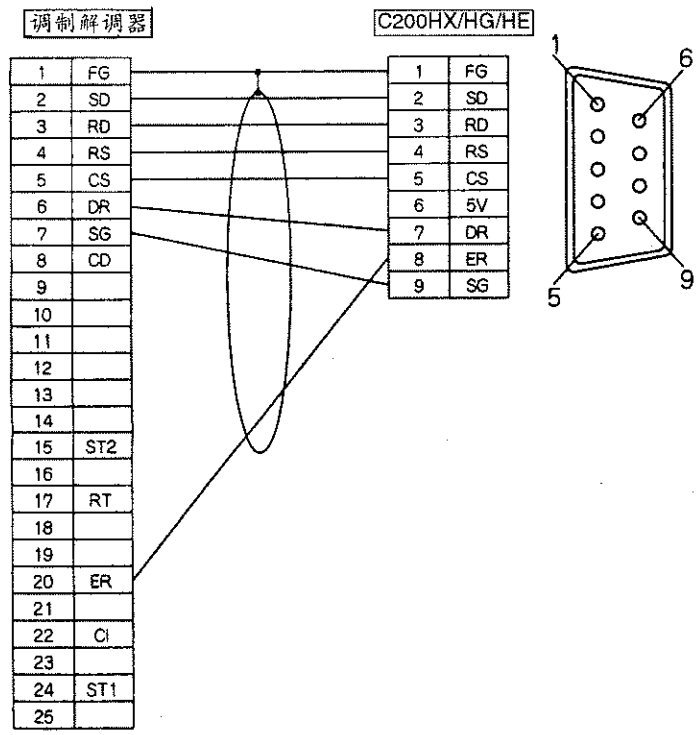
RS－422/485 适配器连接(NT－AL001)



通用设备/计算机连接(RS/CS 流,交叉连接)



调制介调器连接(直线连接)



注

在 PC 上 FG 端子和计算机处接地,电阻小于或等于 100Ω。

详见"C200HX/HG/HE 安装手册"和其它设备有关资料。

8－6－2 通信板设置

使用通信板上通信协议宏功能，必须预先设置下述参数。

通信方式

设置通信方式为通信协议宏功能。

端口 B：DM6550 中位 12～15 置 6。

端口 A：DM6555 中位 12～15 置 6。

标准端口设置

标准设置或用户设置用于端口 A 和 B。当下列位设置为 0，使用标准设置。

端口 B：DM6550 中位 00～03(0：标准设置；1：用户设置)。

端口 A：DM6555 中位 00～03(0：标准设置；1：用户设置)。

项　目	设　置
起动位	1
数据长度	7
停止位	2
奇偶校验	偶
波特率	9,600 bps

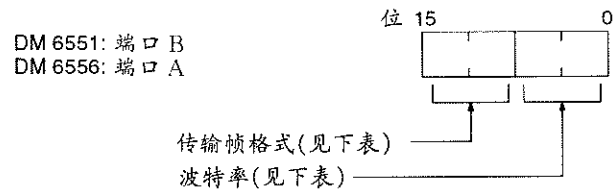
用户端口设置

标准设置或用户设置用于端口 A 和 B。当下列位设置为 1，使用用户设置。

端口 B：DM6550 中位 00～03(0：标准设置；1：用户设置)

端口 A：DM6555 中位 00～03(0：标准设置；1：用户设置)

端口 B 用户设置由 DM6551 定义，端口 A 用户设置由 DM6556 定义。



参　数	设　置	起　动　位	数据长度	停　止　位	奇偶校验
传输帧格式	00	1	7	1	偶
	01	1	7	1	奇
	02	1	7	1	无
	03	1	7	2	偶
	04	1	7	2	奇
	05	1	7	2	无
	06	1	8	1	偶
	07	1	8	1	奇
	08	1	8	1	无
	09	1	8	2	偶
	10	1	8	2	奇
	11	1	8	2	无

参　数	设置	波　特　率
波特率	00	1,200 bps
	01	2,400 bps
	02	4,800 bps
	03	9,600 bps
	04	19,200 bps

8－6－3　通信步骤

必须预先运用通信协议支持软件生成通信协议宏通信序列，再传送到通信板。在 PC 内，使用 PMCR(－)指令执行存储在通信板中的通信序列。

通信序列结构

用通信协议支持软件可以建立多达 1000 个通信序列，序列号从 000～999。

每个通信序列最多由 16 步组成。下表列出通信序列的设置。

项　目		功　能	参　数　设　置
序列设置	发送控制	设置发送控制方法，如 X－on/X－off 流控制或 RS/CS 流控制。	X－on/X－off，RS/CS，调制介调器控制，定界控制，或争议控制
	链接字	设置 PC 和通信板之间数据链接的链接字	IR/SR，LR，HR，AR，DM，和 EM 区域
	监视时间	设置通信处理的监视时间(监视定时器)	接收等待，接收完成，发送完成单位：0.01 秒，0.1 秒，1 秒和 1 分
	响应告示	设置写接收数据的时间	扫描告示或中断告示
步设置	重复计数器	设置重复步的次数	常数 0～255 IR/SR，LR，HR，AR，DM 和 EM 区域
	命令	设置通信命令	发送，接收，或发送与接收
	重复数	设置在执行发送和接收命令时发生错误时重复执行次数	0～9
	发送等候时间	设置在发送期间等候发送数据的时间	单位 0.01 秒，0.1 秒，1 秒和 1 分
	发送信息	设置用于发送命令或发送和接收命令的发送数据	识别码，地址，长度，数据，错误检查码，和终止符
	接收信息	设置用于接收命令或发送和接收命令的期望接收数据	识别码，地址，长度，数据，错误检查码，和终止符
	接收阵列	设置用于接收命令或发送和接收命令的期望接收数据(最多 15 种类型)并按数据类型调整处理方法	识别码，地址，长度，数据，错误检查码，终止符，和下一步处理
	响应告示	设置是否写接收数据	是/否
	下一步处理	设置当前步顺利结束时转往的下一步	END，GOTO，NEXT，或 ABORT
	错误处理	设置当前步中出现错误时转往的下一步	END，GOTO，NEXT，或 ABORT

发送/接收信息结构

发送信息和接收信息的结构如下。

识别码	地址	长度	数　据	错误检查	终止符

项　目	功　能
识别码	设置表示信息开始的数据
地址	设置节点号或其它表示信息送往目标的标志符
长度	数据长度(字节数)自动附加
数据	设置信息内容
错误检查码	设 SUM, LRC, 或 CRC 作为错误检查码。发送时指定的错误码自动附加。接收时，按信息中指定的错误检查码自动执行错误控制同时接收指定量(长度)的数据。
终止符	设置表示信息结束的数据

在发送和接收信息的每一栏可以设置下述属性。缩写"RM"表示接收信息，"TM"表示发送信息。

数据属性		识别码		地址		长度		数据		错误检查		终止符	
		TM	RM	TM	RM	TM	RM	TM	RM	TM	RM	TM	RM
常数	ASCII "□□□□" 十六进制 [□□□□]	Yes	Yes	Yes	Yes	…	…	Yes	Yes	…	…	Yes	Yes
	特殊字符 CR, STX, 等	Yes	Yes	…	…	…	…	Yes	Yes	…	…	Yes	Yes
非变量变换	一阶方程 使用变量(N)	…	…	Yes	Yes	…	…	Yes	Yes	…	…	…	…
	统配符(*)	…	…	…	Yes		…	…	…	Yes	…	…	…
	读字(R)	…	…	Yes	Yes	…	…	Yes	Yes		…	…	…
	写字(W)	…	…	…	Yes	…	…	…	Yes	…	…	…	…
	自动变量 LNG	…	…	…	…	Yes	Yes	…	…	…	…	…	…
	自动变量 SUM, LRC, 和 CRC	…	…	…	…	…	…	…	…	Yes	Yes	…	…
变量ASCII变换	一阶方程 使用变量(N)	…	…	Yes	Yes	…	…	Yes	Yes	…	…	…	…
	统配符(*)	…	…	…	Yes	…	…	…	Yes	…	…	…	…
	读字(R)	…	…	Yes	Yes	…	…	Yes	Yes	…	…	…	…
	写字(W)	…	…	…	Yes	…	…	…	Yes	…	…	…	…
	自动变量 LNG	…	…	…	…	Yes	…	…	…	…	…	…	…
	自动变量 SUM, LRC, 和 CRC	…	…	…	…	…	…	…	…	Yes	…	…	…
变量HEX变换	一阶方程 使用变量(N)	…	…	Yes	Yes	…	…	Yes	Yes	…	…	…	
	统配符(*)	…	…	…	Yes	…	…	…	Yes	…	…	…	…
	读字(R)	…	…	Yes	Yes	…	…	Yes	Yes	…	…	…	…
	写字(W)	…	…	…	Yes	…	…	…	Yes	…	…	…	…
	自动变量 LNG	…	…	…	…	…	Yes	…	…	…	…	…	…
	自动变量 SUM, LRC, 和 CRC	…	…	…	…	…	…	…	…	…	Yes	…	…

读字(R)

对发送和接收信息中的"地址"或"数据"通过设置所需的属性,来读字数据。设置了属性,可从指定字中读到地址或数据,有三种方法指定该字:

1. 使用 PMCR(－)指令中第二个操作数(S,第一个输出字)。

例:R(1)

当命令是"发送",从 PMCR(－)指令的第二个操作数所指定字后第 1 个字中读数据。

2. 使用通信序列链接区域中输入和输出字。

例:R(I1+5)

指定链接区域中第 1 个接收字后面第五个字。

例:R(O2+1)

指定链接区域中第 2 个发送字后面第 1 个字。

3. 可以直接指定数据区字地址

例:R(DM0000+2)

指定 DM0000 后第二个字。

统配符(＊)和写字(W)

在接收数据时可以对"地址"或"数据"设置统配符(＊)和写字。下面解释它们的功能

1. 用 PMCR(－)指令中第三操作数指定字地址(首地址)。

2. 在接收信息的地址中设统配符,不论目标为何处接收任何信息,其结果是一种广播通信。

3. 在接收信息的数据中设统配符,接收所有信息。

4. 在接收信息的地址中设置写字属性,不论目标为何处接收任何信息,同时写该信息到由接收信息的地址所指定的数据区域。

5. 在接收信息数据中设置写字属性,接收所有信息,同时写该信息到由接收信息的地址所指定的数据区域。

使用变量 N 的一阶方程

包含变量 N 的一阶方程用于地址和数据的引入。每当由通信序列的步中指定的重复计数器重复一步时,变量 N 加 1。使用带 N 变量的方程计算地址或数据,可实现下面例子中的一种动态要求。

例:R(2N+6)

指定 PMCR(－)指令的第二个操作数后面第六个字作为"地址"或"数据",在每重复一步时,再加上二个字。

公用信息
通信数据
通信数据
通信数据
通信数据
通信数据

用 2N+6(使用 N 的一阶方程)
指定第一个地址
第 6 个字(N=0)
第 8 个字(N=1)
第 10 个字(N=2)
第 12 个字(N=3)
第 14 个字(N=4)
第 16 个字(N=5)

错误检查码和长度

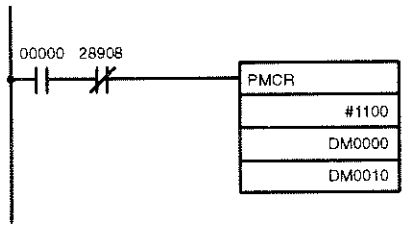
当发送信息时,信息中自动加入数据长度和错误检查码 SUM,LRC,或 CRC。当接收信息时,使用错误检查码检查数据的传输错误同时接收由数据长度指定字节数的数据。

接收阵列

如果在接收信息中设有一个接收阵列,则可以设置最多 15 种类型的接收信息,对每一种接收信息,可以指定不同的过程和错误处理方法。

8－6－4 应用实例

在程序中可以通过 PMCR(－)指令调用并执行通信序列。下例表示一个通信序列，它从 PMCR(－)指令第二个操作数后面的第 1 个字开始一个字接一个字地发送 5 个字的数据，然后将接收到的数据存贮在第



三操作数指定的字中。1. 当 IR00000 为 ON，SR28908(通信板端口 A 指令执行标志)为 OFF 时，通过通信板端口 A 进行数据通信。

2. DM0000 是发送数据的首字，它根据通信序列的重复计数器连续发送(5 次)。

3. 接收数据被写至 DM0010 起始的 DM 区域内。

发送命令必须设置在发送步中，重复计数器设置 5。在用 PMCR(－)指令第二个操作数的发送信息中必须给地址设置读字属性，并且设置一阶方程为 R(N＋1)。

为了将接收数据写到由 PMCR(－)指令中第三个操作数指定的数据区域地址，必须将写接收数据的时间设置在序列设置中响应告示参数中。在每个接收步中必须设有 Recv(接收)命令，并在步设置的响应告示参数中设置“Yes”。

在接收信息中设统配符(＊)这样就可以接收所有数据，在下一个处理过程的发送步和接收步中设置“End”，为处理错误，在发送步和接收步中设置“Abort”。

第九章　故障排除

C200HX/HG/HE 提供自诊断功能，可检查系统的多种不正常情况。这些功能可使故障时间减至最少，并能快速排除和校正错误。

本章提供在 PC 工作期间的有关硬件和软件错误的信息。程序输入错误在 4－7 节“程序的输入、修改和检查”中有说明。虽然在第 3 节存储器区中有过对 SR 和 AR 区的标志和其它错误信息的说明，在 9－5 “错误标志”一节中还是列出了。

9－1　报警指示器…… 412
9－2　可编程报警和错误信息…… 412
9－3　读取和清除错误及其信息…… 412
9－4　错误信息…… 412
9－5　错误标志…… 415
9－6　上位机链接错误…… 417

9－1 报警指示器

在CPU面板上的ALM/ERR报警/错误指示器提供可视的PC异常的指示。当指示器为ON(错误)时，表明已经发生了致命错误(即该错误停止PC运行)；当指示器闪烁时(报警)，则表明发生了非致命错误。在2－1－1节"CPU指示器"中，有该指示器的说明。

警告

对于大部分硬件错误和某些致命的软件错误，或在程序(见下页的表格)执行FALS(07)时，PC将接通ALM/ERR(报警/错误)指示器，停止程序的执行并关断PC的所有输出。对所有其余的错误，PC操作将继续进行。用户有责任采取适当的措施来确保因发生致命错误而导致的系统自动关机，不会产生危险的后果，并且要采取适当的步骤来对付使系统不自动关机的故障。系统标志和其它系统和/或用户程序错误指示器，可用到程序的适当的步骤中。

9－2 可编程报警和错误信息

FAL(06)，FALS(07)和MSG(46)可编在程序中，以提供在出错情况下用户编程信息。用这三条指令，用户可修改所诊断的错误，有助于排除故障。

FAL(06)使用除了00以外的FAL号，在FAL(06)执行时，FAL号输出到SR区。执行FAL(06)不会停止PC的运行，或者直接影响任何来自PC的输出。

FALS(07)也使用FAL号，当执行FALS(07)时，FAL号输出到SR区中相同的地址。执行FALS(07)会停止PC的运行，并关闭PC所有的输出。

当FAL(06)和00功能号一起使用时，如果系统已在存储器中存储了较多的数据，就会清除放在SR区内的当前的FAL号，而由另一个所取代。

当使用MSG(46)时，指定数据区字的信息将显示在编程器或其他编程装置上。

这些指令的详细说明在第五章"指令集"中。

9－3 读取和清除错误及其信息

系统出错信息可显示在编程器上或其它编程设备上。

在编程器上，可按CLR、FUN和MONTR键。如果系统存有多个错误信息，可再按MONTR键以得到下一个信息。如果系统处在编程方式，按MONTR键将清除错误信息。因此当你读错误信息时候必须记下所有信息。(处在运行或监视方式，是不会清除错误或者信息的，而PC必须在编程方式下才能做到)。所有信息清除以后，会显示"ERR CHK OK"。

有关从编程器得到错误信息的详细内容参见7－1节"监控操作和修改数据"。LSS或SSS(中文版)的方法分别参见"LSS操作手册"或"SSS(中文版)操作手册：C系列。"

9－4 错误信息

一般来说，可显示三种类型的错误信息：初始化错误、非致命操作错误和致命操作错误。这些信息中大部分都能由FAL号来指示，而这FAL号被传送到SR区中FAL区内。

下面叙述了三种类型的错误，由CPU上的指示器可迅速检测错误的类型。如果在叙述中没有提及到指示器的状态，那么它是亮或暗就没有区别了。

排除了错误以后，在重新开始操作之前，从存储器中清除错误信息。

下表中错误信息中的星号表示可变的数值。而实际数字显示在显示器上。

初始化错误

以下错误信息出现在程序开始执行之前。对这一类型中的任一种错误，电源指示灯亮，而运行指示灯熄灭。

错误和信息	FAL 号	原因	处置
等待特殊 I/O 或中断输入单元 CPU WAIT'G	无	特殊 I/O 单元或中断输入单元没有初始化。	执行读 I/O 表操作以便检查单元号。如果在 I/O 表中仅指示为“$”，更换部件。（高密度 I/O 单元不出现在对所有的外围设备读 I/O 表的显示中）。
等待远程 I/O CPU WAIT'G	无	远程 I/O 单元的电源关断，或未找到终端。	检查远程 I/O 单元的电源，把远程 I/O 单元和所设定的终端连接起来。

非致命操作错误

对于程序启动执行之后发生的错误出现下面的信息。在一个或多个这样错误发生以后，程序仍将继续执行。对于每个错误，POWER 和 RUN 指示灯点亮，而 ALM/ERR 指示器将闪烁。

错误和信息	FAL 号	原因	处置
FAL 错误 SYS FAIL FAL**	01～99	在程序中 FAL(06)已被执行，检查 FAL 号确定会引起执行的条件。（由用户设定）	根据由 FAL 号所指示的原因，纠正之。 （由用户设定）
中断输入单元错误 SYS FAIL FAL8A	8A	错误发生在中断输入单元和 CPU 之间的数据传送中。	更换中断输入单元。
中断子程序错误 SYS FAIL FAL8B	8B	在远程 I/O 单元作 I/O 刷新或上位机链接服务期间，执行长于 10ms 的中断子程序。	检查 SR262 和 SR263 内容，验证中断子程序处理时间小于 10ms。
		按特殊 I/O 单元循环刷新设置类别，试图执行不同类型的 I/O 刷新。	修改程序或特殊 I/O 单元循环刷新设置，使用相同的刷新方法。
高密度 I/O 单元错误 SYS FAIL FAL9A	9A	错误发生在高密度 I/O 单元与 CPU 之间的数据传输中。	检查 AR0205～AR0214 以辨认有毛病的单元，更换单元重新起动 PC。
PC 设置错误 SYS FAIL FAL9B	9B	在 PC 设置中，已测到错误，当读设置和第一次使用时，该错误发生。	检查、纠正 PC 的设置。 （SR27500～27502 指示哪部分 PC 设置是错的）。
通信板错误 SYS FAIL FAL9C	9C	错误发生在一通信板上。	详见通信板的操作手册。
存储器盒传输错误 SYS FAIL FAL9D	9D	错误发生在 UM 和存储器盒之间的数据传送期间，因为： ·未处于编程方式 ·UM 或存储器为只读 ·UM 或存储器容量不够 ·在存储器盒中发生校验和错误	确认 PC 处于编程（PROGRAM）方式。 确认存储器盒不处于写保护。 确保 UM 和存储器盒有足够容量。 确保传送期间 SYSMAC NET 数据链接不工作。 再次传送数据。

错误和信息	FAL 号	原因	处置
循环时间超过 CYCLE TIME OVER	F8	监视定时器超过 100ms。	程序扫描周期比建议值长，如果可能的话减少扫描周期。
I/O 表格检验错误 I/O VER ERR	E7	单元已被拆除或由不同种类的单元取代，使得 I/O 表不正确。	使用 I/O 表检验操作来检查 I/O 表，连接虚设单元或再次登记 I/O 表。
远程 I/O 错误 REMOTE ERR Remote I/O Master Unit number *	B0 或 B1	错误发生在远程 I/O 单元之间的传输中。	检查在 PC 和主站以及和远程 I/O 单元间的传输线。
特殊 I/O 部件错误 SIOU ERR	D0	错误发生在 PC 链接单元；远程 I/O 主单元；上位机链接，SYSMAC LINK 或 SYSMAC NET 链接单元和 CPU 之间；或在特殊 I/O 单元和 CPU 之间刷新中。	查明引起错误单元的单元号（AR00 或 SR282），纠正错误，触发 AR01，SR250，或 SR252 中适当的再启动位。如果单元不能重新启动，更换之。
电池错误 BATT LOW	F7	备用电池失电或者电压下降。	检查电池，如有必要更换电池。

致命操作错误

以下错误信息显示了错误发生在执行程序开始之后。当以下任何错误发生时，停止 PC 操作和程序的执行，并且关断 PC 所有的输出。对于电源中断错误 CPU 指示灯不亮。对于其它的致命操作错误，电源和 ALM/ERR 指示灯将被点亮，运行输出将被关闭。

错误和信息	FAL 号	原因	处置
电源中断 No message.	无	电源被中断至少 10ms。	检查电源电压和电源线，重新加电。
存储器错误 MEMORY ERR	F1	SR27211 ON 状态：在 PC 设置中出现代码和检验错误（DM6600～DM6655）。	检查 PC 的设置。
		SR27212 ON 状态：在程序中发生了代码和检验错误，指出一条不正确的指令。	检查程序。
		SR2713 ON：在扩展指令改变时发生了代码和检验错误。	
		SR27214 ON：电源接通时安装或拆除存储器盒。	正确安装存储器盒。
		SR27215 ON：引导错误。	检查 CPU 存储器是否加保护或者在存储器盒中发生了代码和检验错。

错误和信息	FAL 号	原因	处置
没有 END(01)指令 NO END INST	F0	程序中任何地方都未写 END(01)。	在程序的最末地址上写上 END(01)。
I/O 总线错误 I/O BUS ERR * Rack no.	C0～C3	错误发生在 CPU 和 I/O 装置之间的总线中。	FAL 号最右边的数字会指示已查出错误的机架号。检查机架之间的连接电缆。
单元太多 I/O UNIT OVER	E1	两个或更多的特殊 I/O 单元或组－2 高密度单元设有相同的单元号。	执行 I/O 表读操作检查单元号，消除重复的号码。
		需要两个字的特殊 I/O 单元的单元号被设置为最后一个单元号(9 或 F)。	需要两个字的单元的单元号不能设置最后一个单元号。单元号改为允许的设定值。
		64 点组－2 高密度 I/O 单元的 I/O 号被设置为最后一个单元号(9 或 F)。	64 点组－2 高密度 I/O 单元不能设置最后一个单元号。单元号改为有效的设定值。
		两个 SYSMAC NET 链接或 SYSMAC LINK 单元共用相同操作级。	检查 SYSMAC NET 链接和 SYSMAC LINK 单元操作级，消除重复。
		安装三个或更多中断输入单元。	最多可安装二个中断输入单元。
		特殊 I/O 单元或高密度 I/O 单元的单元号不在允许设定范围之内。	在允许的设定范围之内设置单元号。
		第三个扩展机架连接到 PC，而该 PC 只能容纳二个。	断开第三个机架，除非 PC 可以容纳三个机架。
输入－输出 I/O 表错误 I/O SET ERROR	E0	登记在 I/O 表中所指定的输入和输出字与实际安装单元的所要求的输入输出字不一致	用 I/O 表的检验操作检查 I/O 表，检查所有的单元直到使其为正确的结构。当系统确证无误时，再次登记 I/O 表。
FALS 错误 SYS FAIL FALS**	01 ～ 99 或 9F	FALS 已由程序执行。检查 FAL 号以决定会引起的执行条件(由用户或由系统设置)	根据由 FAL 号所指示的原因纠正之。如果 FAL 号为 9F，检查监视定时器和扫描周期可能太长。

通信错误

如果通信出错，外设端口和 RS－232C 端口的指示灯(COMM)将不亮。检查连线和两端(C200HX/HG/HE 和外设)的程序，然后使用复位位(RS－232C 端口：25209)复位端口。

其它错误信息

在本手册中详述了其它错误的号码。程序输入和调试中的错误在第四章中“编写和输入程序”中有说明。

9－5 错误标志

下表列出了标志和 SR 和 AR 区所提供的能用于故障排除的其它信息。详情在 3－4 节“SR 区域”和 3－5 节“AR 区域”中提供。

SR 区

地　址	功　　能
23600～23615	SYSMAC NET 链接系统的节点回路状态
23700～23715	在 SYSMAC LINK/SYSMAC NET 链接系统中 SEN(90)/RECV(98)完成/错误码输出区
24700～25015	PC 链接单元运行和错误标志
25100～25115	远程 I/O 单元错误标志
25200	SYSMAC LINK/SYSMAC NET 链接 0 级 SEND(90)/RECV(98)错误标志
25203	SYSMAC LINK/SYSMAC NET 链接 1 级 SEND(90)/RECV(98)错误标志
25206	机架安装上位机链接单元 1 级错误标志
25208	RS－232C 端口错误标志
25300～25307	FAL 号输出区
25308	电池低电压标志
25309	扫描周期错误标志
25310	I/O 检查错误标志
25311	机架安装上位机链接单元 0 级错误标志
25312	远程 I/O 错误标志
25411	中断输入单元错误标志
25413	中断程序错误标志
25414	组－2 高密度 I/O 单元错误标志
25415	特殊单元错误标志(特殊 I/O,PC 链接,上位机链接,远程 I/O 主站,SYSMAC NET 链接,或 SYSMAC LINK 单元错误标志)
25503	指令执行错误(ER)标志
26400～26403	RS－232C 端口错误代码
26404	RS－232C 端口通信出错
26408～26411	外设端口错误代码(除外设方式外)
26412	外设端口通信错误标志(除外设方式外)
26800～26815	通信板错误信息
27011	UM 传送错误标志:数据链接表传送期间 SYSMAC NET 数据链接工作
27012	UM 传送错误标志:不是编程方式
27013	UM 传送错误标志:只读
27014	UM 传送错误标志:容量不够或者无 UM
27015	UM 传送错误标志:主板校验和错误
27211	存储器错误标志:PC 设置校验和错误
27212	存储器错误标志:UM 或梯形码校验和错误
27213	存储器错误标志:扩展指令码转换区校验和错误
27214	存储器错误标志:存储器盒在线拆卸
27215	存储器错误标志:引导错误
27312	IOM 传送错误标志:不是编程方式
27313	IOM 传送错误标志:只读
27314	IOM 传送错误标志:容量不够
27500	PC 设置错误(DM6600～DM6605)
27501	PC 设置错误(DM6613～DM6623)
27502	PC 设置错误(DM6635～DM6655)
28000～28015	组－2 高密度 I/O 单元(0～F)错误标志
28200～28215	特殊 I/O 单元(0～F)错误标志

AR 区

地　址	功　　能
0000～0009	特殊 I/O 或 PC 链接单元错误标志
0010	SYSMAC LINK/SYSMAC NET 链接 1 级系统错误标志
0011	SYSMAC LINK/SYSMAC NET 链接 0 级系统错误标志
0012	机架安装上位机链接单元 1 级错误标志
0013	机架安装上位机链接单元 0 级错误标志
0014	远程 I/O 主单元 1 级错误标志
0015	远程 I/O 主单元 0 级错误标志
0200～0204	从站机架 0 到 4 的错误标志
0205～0214	组－2 高密度 I/O 单元错误标志(AR0205～AR0214 对应于 I/O 号 0～9)
0215	组－2 高密度 I/O 单元错误标志
0300～0315	光缆型 I/O 单元(0～7)错误标志
0400～0415	光缆型 I/O 单元(8～15)错误标志
0500～0515	光缆型 I/O 单元(16～23)错误标志
0600～0615	光缆型 I/O 单元(24～31)错误标志
0713～0715	错误历史位
1114	通信控制器错误标志 0 级
1115	EEPROM 0 级操作错误标志
1514	通信控制器错误标志 1 级
1515	EEPROM 1 级操作错误标志

9－6　上位机链接错误

这是一些在响应格式中可能返回的响应代码(结束代码)。当出现两个或更多错误时,将返回第一个错误的结束代码。

结束码	内　容	可　能　的　原　因	改　正　措　施
00	正常完成	…	…
01	运行方式下不执行	PC 在运行方式下,不能执行发送的命令。	检查命令和 PC 模式之间关系。
02	监控方式下不执行	PC 在监控方式下,不能执行发送的命令。	
03	UM 写保护	PC 的 UM 处于写保护。	PC 上 DIP 开关的脚 1 置 OFF。
04	地址超出区域	设置在 SV 读命令或 SV 修改命令中程序地址超出 65,535 最大值。	改正程序的地址设置,再传送命令。
13	FCS 错误	FCS 错误,或是 FCS 计算错,或是噪声干扰。	检查 FCS 计算方法。如是噪声干扰,再传送命令。
14	格式错误	命令格式错误或者不可分割的命令被分割。	检查格式,再传送命令。
15	入口码数据错误	数据超出规定范围或太长。	修改数据,再传送命令。
16	命令不支持	程序中不存在 SV 读命令或 SV 修改命令中指定操作数。	改正命令和程序。
18	帧长度错误	超过最大的帧长度 132 个字节。(如帧超过 280 个字节,接收溢出标志变 ON,不再返回应答。)	检查命令,必要时分成若干个帧。
19	不执行	读的项没有用组合命令(QQ)登记过。	在批形式读之前,执行 QQ 命令登记读的项。
23	用户存储器写保护	C200HX/HG/HE 上 DIP 开关脚 1 为 ON。	置 CPU 上 DIP 开关的脚 1 为 OFF。
A3	传送数据时因 FCS 错误引起中止	在第二或后面的帧中出现 FCS 错误。	检查 FCS 计算方法。如果噪声干扰,再传送命令。
A4	传送数据时因格式错误引起中止	命令格式与第二或后面的帧中字节数不匹配。	检查格式,再传送命令。

（续表）

结束码	内　容	可　能　的　原　因	改　正　措　施
A5	传送数据时因入口码数据错误引起中止	在第二或后面的帧中有入口码数据错误。	修改数据，再传送命令。
A8	传送数据时因帧长度错误引起中止	第二或后面帧的长度超过最大值 132 个字节。	保持帧不超过 132 字节。

无应答错误

由于某些错误，造成无应答，与命令无关。这些错误列于下表。

错　　误	PC　操　作
命令接收期间，奇偶校验或帧错误	通信错误标志置 ON，错误代码被记录，接收复位。（如通信重新起动正常，错误自动清除。）
280 个字节中未接收到“回车”（CR）。	SR26407（RS－232C 端口接收溢出标志）变 ON。当传送正常命令，通信将恢复正常。
接收到第一帧开始处无ⓐ字符的命令。	接收复位。
错误节点号（不是局部节点，为十六进制数，或超过 31）	命令废除，接收复位。
写命令的中间帧或最后帧只有 1 个或 2 个数据字节。	FCS 错误。

第十章　上位机链接命令

本章介绍通过 C200HX/HG/HE 端口进行上位机链接通信的上位机链接命令。关于使用上位机链接命令的步骤和上位机链接命令有关的错误方面的资料，参阅 8－2 节“上位机链接通信”。

10－1　上位机链接命令汇总 …… 422

10－2　上位机链接结束码 …… 423

10－2－1　结束码汇总 …… 423

10－2－2　命令/结束代码表 …… 424

10－3　上位机链接命令 …… 425

10－3－1　读 IR/SR 区——RR …… 425

10－3－2　读 LR 区——RL …… 426

10－3－3　读 HR 区——RH …… 426

10－3－4　读 PV 值——RC …… 427

10－3－5　读 TC 状态——RG …… 428

10－3－6　读 DM 区——RD …… 429

10－3－7　读 AR 区——RJ …… 430

10－3－8　读 EM 区——RE …… 431

10－3－9　写 IR/SR 区——WR …… 432

10－3－10　写 LR 区——WL …… 433

10－3－11　写 HR 区——WH …… 434

10－3－12　写 PV 值——WC …… 435

10－3－13　写 TC 状态——WG …… 436

10－3－14　写 DM 区——WD …… 437

10－3－15　写 AR 区——WJ …… 438

10－3－16　写 EM 区——WE …… 439

10－3－17　SV 读 1——R＃ …… 440

10－3－18　SV 读 2——R＄ …… 441

10－3－19　SV 读 3——R％ …… 443

10－3－20　SV 改变 1——W＃ …… 445

10－3－21　SV 改变 2——W＄ …… 446

10－3－22　SV 改变 3——W％ …… 448

10－3－23　读状态——MS …… 450

10－3－24　写状态——SC …… 452

10－3－25　读错误——MF …… 453

10－3－26　强制置位——KS …… 454

10－3－27　强制复位——KR …… 456

10－3－28　多重强制置位/复位——FK …… 457

10－3－29　强制置位/复位取消——KC …… 458

10－3－30　读 PC 型号——MM …… 459

10－3－31　测试——TS …… 460

10－3－32　读程序——RP …… 461

10－3－33　写程序——WP …… 462
10－3－34　I/O 表生成——MI …… 463
10－3－35　组合命令——QQMR …… 464
10－3－36　组合命令读数据——QQIR …… 466
10－3－37　中止——XZ …… 467
10－3－38　初始化——＊＊ …… 467
10－3－39　TXD 应答——EX …… 468
10－3－40　未定义命令——IC468 …… 468

10－1 上位机链接命令汇总

命令图表

下列图表中列出的命令可用于上位机与 C200HX/HG/HE 之间的通讯。

识别码	PC 方式			名称	页数
	运行	监视	编程		
RR	有效	有效	有效	读 IR/SR 区	425
RL	有效	有效	有效	读 LR 区	426
RH	有效	有效	有效	读 HR 区	426
RC	有效	有效	有效	读 PV	427
RG	有效	有效	有效	读 TC 状态	428
RD	有效	有效	有效	读 DM 区	429
RJ	有效	有效	有效	读 AR 区	430
RE	有效	有效	有效	写 RM 区	431
WR	无效	有效	有效	写 IR/SR 区	432
WL	无效	有效	有效	写 LR 区	433
WH	无效	有效	有效	写 HR 区	434
WC	无效	有效	有效	写 PV	435
WG	无效	有效	有效	写 TC 状态	436
WD	无效	有效	有效	写 DM 区	437
WJ	无效	有效	有效	写 AR 区	438
WE	无效	有效	有效	写 EM 区	439
R#	有效	有效	有效	SV 读 1	440
R$	无效	有效	有效	SV 读 2	441
R%	有效	有效	有效	SV 读 3	443
W#	无效	有效	有效	SV 改变 1	445
W$	无效	有效	有效	SV 改变 2	446
W%	无效	有效	有效	SV 改变 3	448
MS	有效	有效	有效	读状态	450
SC	有效	有效	有效	写状态	452
MF	有效	有效	有效	读错误	453
KS	无效	有效	有效	强制置位	454
KR	无效	有效	有效	强制复位	456
FK	无效	有效	有效	多重强制置位/复位	457
KC	无效	有效	有效	强制置位/复位的取消	458
MM	有效	有效	有效	读 PC 类型	459
TS	有效	有效	有效	测试	460
RP	有效	有效	有效	读程序	461
WP	无效	无效	有效	写程序	462
MI	无效	无效	有效	I/O 表生成	463
QQ	有效	有效	有效	组合命令	464
XZ	有效	有效	有效	中止(仅用于命令)	467
* *	有效	有效	有效	初始化(仅用于命令)	467
EX	有效	有效	无效	TXD 应答(仅用于应答)	468
IC	——	——	——	未定义命令(仅用于应答)	468

10－2 上位机链接结束码

10－2－1 结束码汇总

下面是在应答帧中返回的结束码。当出现二个或以上的错误时，返回第一个错误的结束码。

结束码	内 容	可 能 的 原 因	改 正 措 施
00	正常完成	…	…
01	运行方式下不执行	PC 在运行方式下，不能执行发送的命令。	检查命令和 PC 方式之间的关系。
02	监控方式下不执行	PC 在监控方式下，不能执行发送的命令。	
03	UM 写保护	PC 的 UM 处于写保护。	PC 上 DIP 开关的脚 1 置 OFF。
04	地址超出区域	设置在 SV 读命令或 SV 修改命令中程序地址超出 65,535 最大值。	修改程序的地址设置，再传送命令。
13	FCS 错误	FCS 错误，或是 FCS 计算错，或是噪声干扰。	检查 FCS 计算方法。如是噪声干扰，再传送命令。
14	格式错误	命令格式错误或者不可分割的命令被分割。	检查格式，再传送命令。
15	入口码数据错误	数据超出规定范围或太长。	修改数据，再传送命令。
16	命令不支持	程序中不存在 SV 读命令或 SV 修改命令中指定的操作数。	改正命令和程序。
18	帧长度错误	超过最大的帧长度 132 个字节。（如帧超过 280 个字节，接受溢出标志变 ON，不再返回应答。）	检查命令，必要时分成若干个帧。
19	不执行	读的项没有用组合命令(QQ)登记过。	在批形式读之前，执行 QQ 命令登记读的项。
20	不能生成 I/O 表	无法辨认的远程 I/O 单元，I/O 字太多，或远程 I/O 单元节点号重复。	检查远程 I/O 系统和 I/O 字数。
23	用户存储器写保护	C200HX/HG/HE 上 DIP 开关脚 1 为 ON。	置 CPU 上 DIP 开关脚 1 为 OFF。
A3	传送数据时因 FCS 错误引起中止	在第二或后面的帧中出现 FCS 错误。	检查 FCS 计算方法。如果噪声干扰，再传送命令。
A4	传送数据时因格式错误引起中止	命令格式与第二或后面的帧中字节数不匹配。	检查格式，再传送命令。
A5	传送数据时因入口码数据错误引起中止	在第二或后面的帧中有入口码数据错误。	检查数据，再传送命令。
A8	传送数据时因帧长度错误而引起中止	第二或后面的帧的长度超过最大值 132 个字节。	保持帧不超过 132 个字节。

无应答错误

由于某些错误，造成无应答，与命令无关。下表列出这些错误。

错 误	PC 操 作
命令接收期间，奇偶校验或帧错误。	通信错误标志置 ON，记录错误代码，接收复位。（如通信重新启动正常，错误自动清除。）
280 个字节中间未接收到“回车”(CR)。	SR26407(RS－232C 端口接收溢出标志)变 ON。当传送正常命令，通信将恢复正常。
接收到第一帧开头无@字符的命令。	接收复位。
错误节点号(不是局部节点，为十六进制数，或超过 31)	命令废除，接收复位。
写命令的中间帧或最后帧只有 1 个或 2 个字节数据。	FCS 错误。

10－2－2 命令/应答代码表

下表列出了每个命令返送的结束码。

识别符	可能的结束代码	注解
RR	00 13 14 15 18 A3 A8	…
RL	00 13 14 15 18 A3 A8	…
RH	00 13 14 15 18 A3 A8	…
RC	00 13 14 15 18 A3 A8	…
RG	00 13 14 15 18 A3 A8	…
RD	00 13 14 15 18 A3 A8	…
RJ	00 13 14 15 18	…
RE	00 13 14 15 18 A3 A8	…
WR	00 01 13 14 15 18 A3 A4 A5 A8	…
WL	00 01 13 14 15 18 A3 A4 A5 A8	…
WH	00 01 13 14 15 18 A3 A4 A5 A8	…
WC	00 01 13 14 15 18 A3 A4 A5 A8	…
WG	00 01 13 14 15 18 A3 A4 A5 A8	…
WD	00 01 13 14 15 18 23 A3 A4 A5 A8	…
WJ	00 01 13 14 15 18 A3 A4 A5 A8	…
WE	00 01 13 14 15 18 A3 A4 A5 A8	…
R#	00 13 14 15 16 18 23	…
R$	00 04 13 14 15 16 18 23	…
R%	00 04 13 14 15 16 18 23	…
W#	00 01 13 14 15 16 18 23	…
W$	00 01 04 13 14 15 16 18 23	…
W%	00 01 04 13 14 15 16 18 23	…
MS	00 13 14 18	…
SC	00 13 14 15 18 19	…
MF	00 13 14 15 18	…
KS	00 01 13 14 15 18	…
KR	00 01 13 14 15 18	…
FK	00 01 13 14 15 18	…
KC	00 13 14 18	…
MM	00 13 14 18	…
TS	13 14 18	…
RP	00 13 14 18 23 A3 A8	…
WP	00 01 02 13 14 15 18 19 23 A3 A4 A5 A8	…
MI	00 01 02 03 13 14 18 20	…
QQ	00 13 14 15 18 19 A3 A4 A5 A8	…
XZ	…	无应答
* *	…	无应答
EX	…	无结束代码
IC	…	无结束代码

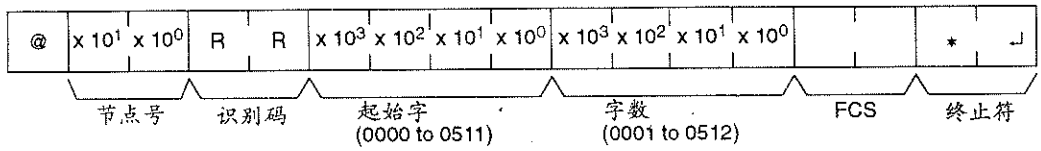
10－3 上位机链接命令

本节介绍由上位机发往 PC 的各种上位机链接命令。关于使用上位机链接命令的步骤和上位机链接命令有关的错误方面的资料，参阅 8－2 节“上位机链接通信”。

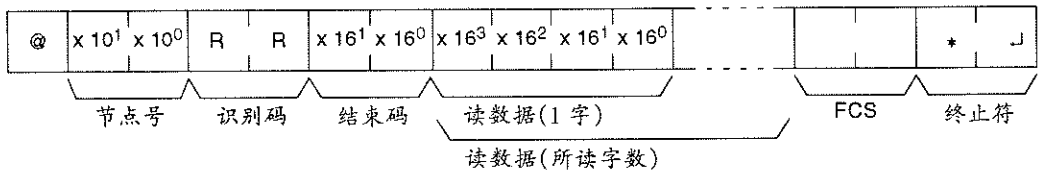
10－3－1 读 IR/SR 区——RR

读指定的若干 IR 和 SR 字的内容，以指定字为起始。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

对该命令的多帧应答，可用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

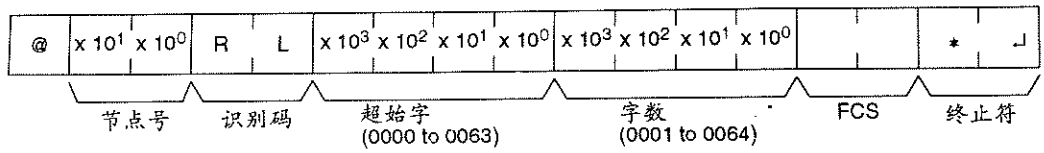
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

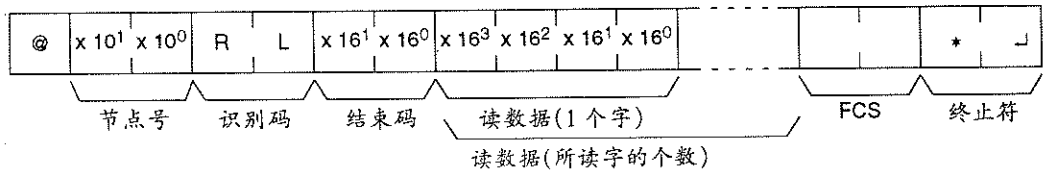
10－3－2 读 LR 区——RL

读指定的若干 LR 字内容，以指定字为起始。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

对该命令的多帧应答，可用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

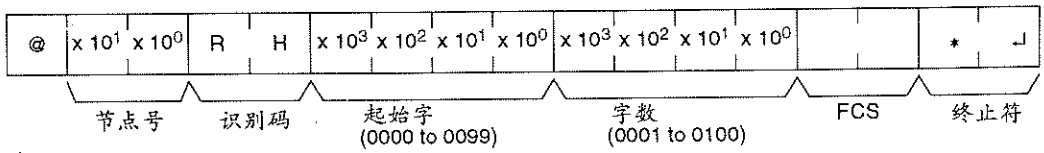
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

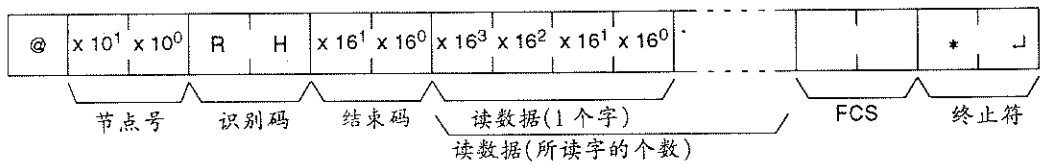
10－3－3 读 HR 区——RH

读指定的若干 HR 字的内容，以指定字为起始。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

对该命令的多帧应答，可用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

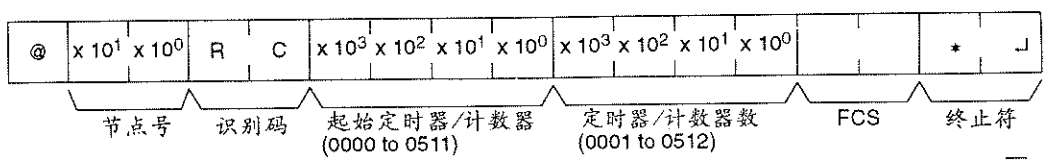
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

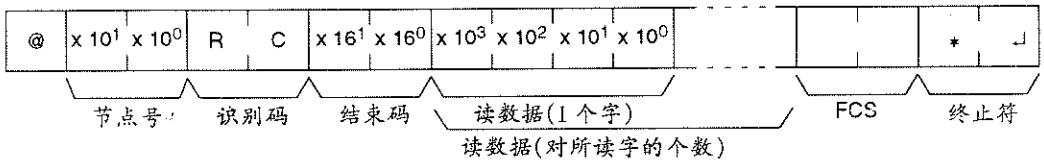
10－3－4 读 PV 值——RC

读指定的若干个定时器/计数器当前值(PV)的内容，以指定定时器/计数器为起始。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

对该命令的多帧应答，可用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

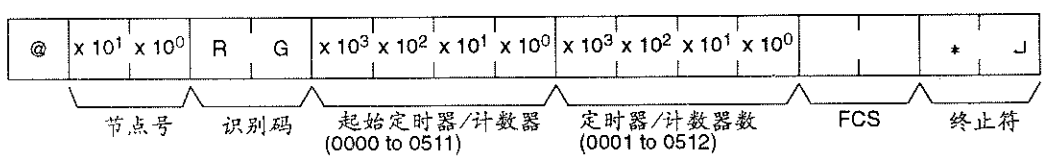
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

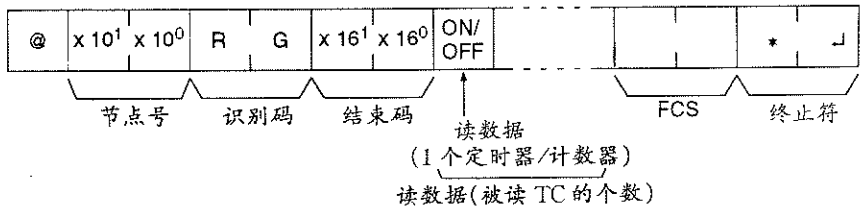
10－3－5 读 TC 状态——RG

读指定的若干个定时器/计数器的结束标志状态，以指定的定时器/计数器为首。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 121 个字。如果读取字超过 121 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 124 个字。

对该命令的多帧应答，可使用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

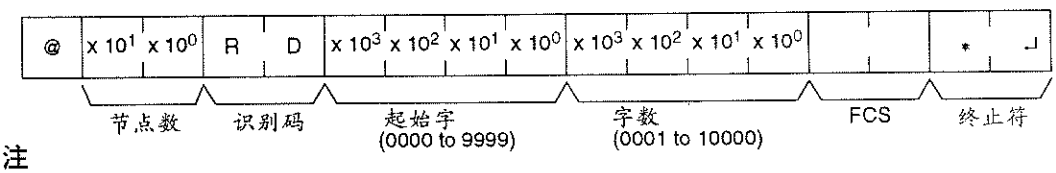
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

10－3－6 读 DM 区——RD

读指定的若干个 DM 字的内容，以指定字为起始。

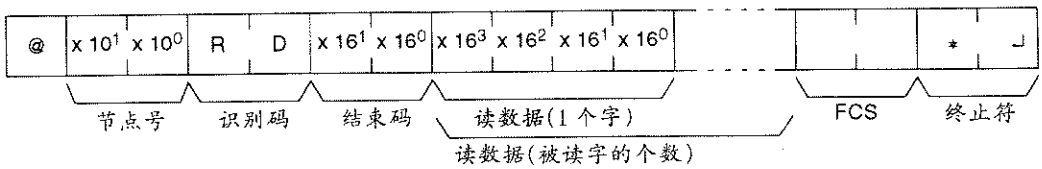
命令格式



注

若读 10,000 个字，被读字数用 0000 表示。

应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

不存在 DM6656～DM6999，如果试图读这些字，不出现错误，而返送“0000”作为应答。同样，如果 UM 区域中未分配扩展 DM(DM7000～DM9999)区，“0000”作为读取扩展 DM 的应答返回。

对该命令的多帧应答，可用初始化(INITIALIZE)和中止(ABORT)命令代替定界符。如发送其它命令，它们同样作为定界符处理。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

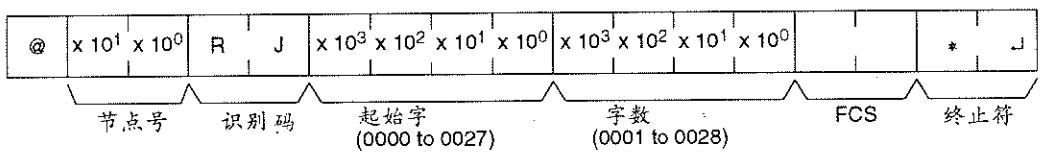
如果命令长度不对，返回结束码 14(格式错误)，如果指定的字超出数据区域，或不是 BCD 码，返回结束码(入口码数据错误)。

结束代码	内容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

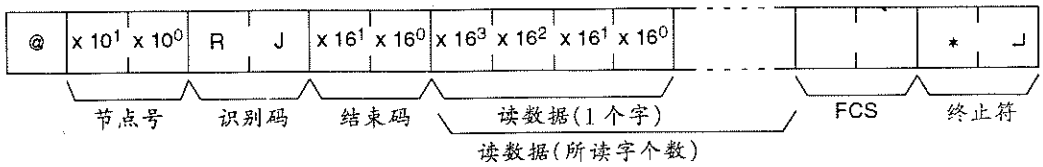
10－3－7 读 AR 区——RJ

读指定的若干个 AR 字的内容，以指定字为起始。

命令格式



应答格式



限制

第一应答帧的正文最多可包含 30 个字。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

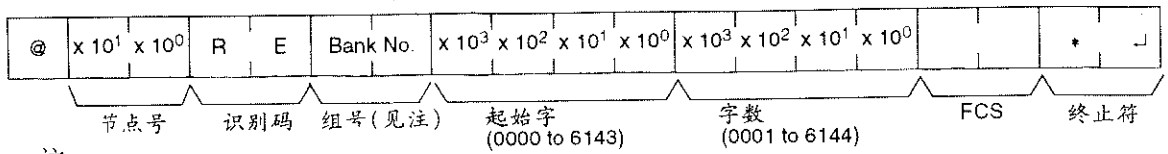
如果命令长度不对，返回结束码 14(格式错误)，如果指定字超出数据区域，或不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误

10－3－8 读 EM 区——RE

读指定的若干个 EM 字的内容，以指定 EM 组中指定字为起始。

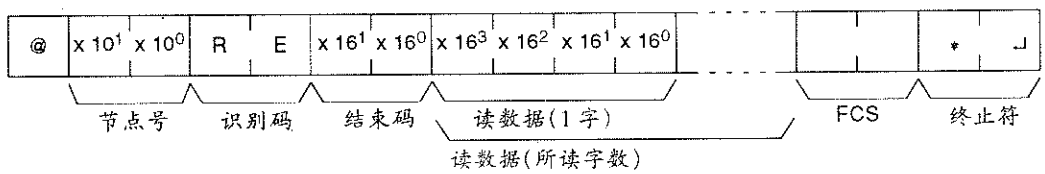
命令格式



注

输入 00，01，或 02 表示组号 0，1，或 2。输入二个空格表示当前组。

应答格式



限制

第一应答帧的正文最多可包含 30 个字。如果读取字超过 30 个，分若干帧返送数据。

在第二和之后帧中，应答的正文最多可包含 31 个字。

如指定有效的组号，但 PC 没有装入该 EM 组，“0000”作为读取数据返回。

该命令不能用以修改当前组号。

对该命令的多帧应答，可用初始化（INITIA LIZE）和中止（ABORT）命令代替分界符。如发送其它命令，它们同样作定界符处理。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

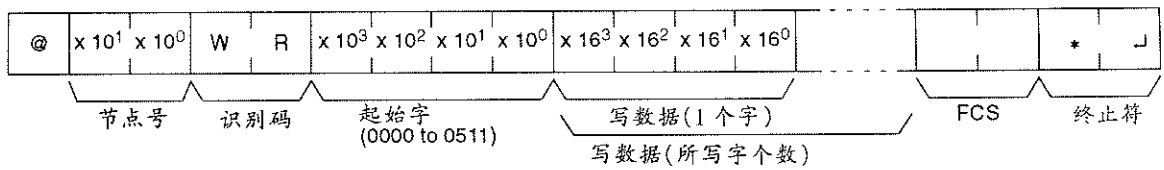
如果命令长度不对，返回结束码 14（格式错误），如果指定的字超出数据区域，或不是 BCD 码，或指定了无效的组号，返回结束码 15（入口码数据错误）。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

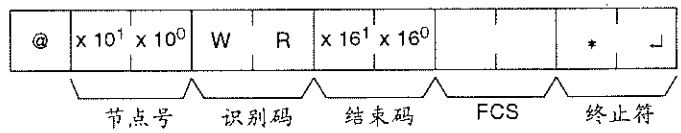
10－3－9 写 IR/SR 区——WR

将数据写入 IR 和 SR 区中，以指定字为起始，按字依次写入。

命令格式



应答格式



限制

字 253～255 不能写入数据。如试图对这些字写操作，不出现错误，但这些字中不写入任何内容。除了第一个写入数据字外，其它写入数据可以分成多帧。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写数据的第一个字不在第一帧中，返回结束码 14(格式错误)。

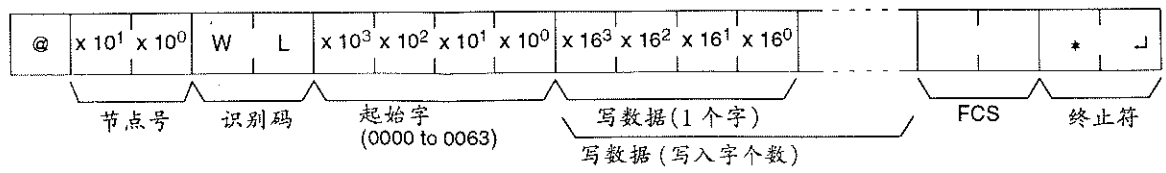
如果指定的写数据超出数据区域，或起始字不是 BCD 码，或写数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写数据返回结束码 A5，而不是 15。)

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

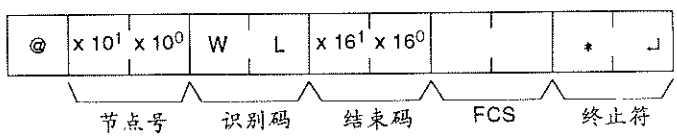
10－3－10 写 LR 区——WL

将数据写入 LR 区，以指定字为起始，按字依次写入。

命令格式



应答格式



限制

除了第一个写入数据字之外，其它写入数据可以分成多帧。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写入数据的第一个字不在第一帧中，返回结束码 14(格式错误)。

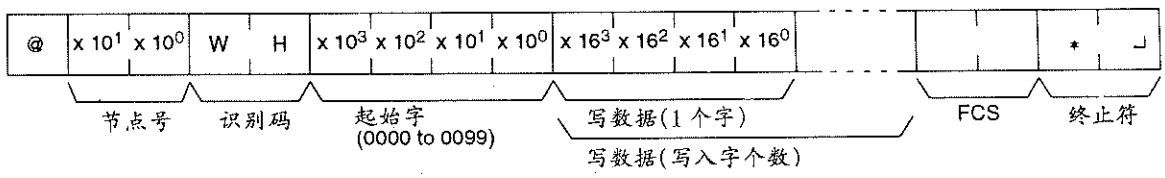
如果指定的写入数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写入数据，返回结束码 A5，而不是 15。)

结束代码	内 容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

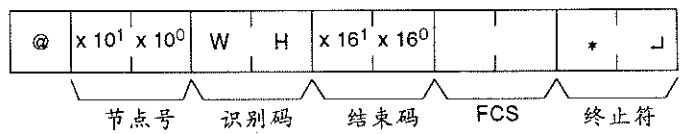
10－3－11 写 HR 区——WH

写数据到 HR 区，以指定字为起始，按字依次写入。

命令格式



应答格式



限制

除第一个写入数据字外，其它写入数据可以分成多帧。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如命令长度不对或写入数据的第一个字不在第一帧中，返回结束码 14(格式错误)。

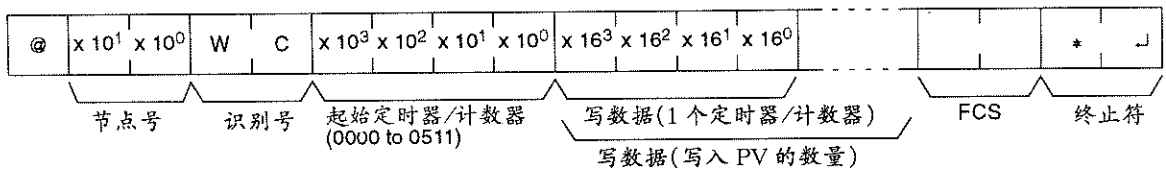
如果指定的写数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写入数据，返回结束码 A5，而不是 15。)

结束代码	内 容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

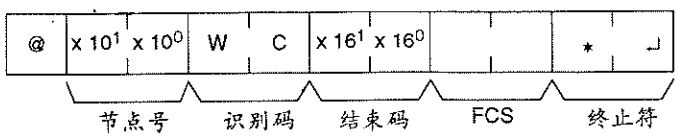
10－3－12 写 PV 值——WC

写定时器/计数器的 PV 值(当前值)，从指定的定时器/计数器开始。

命令格式



应答格式



限制

除第一个写入数据字外，其它写入数据可以分成多帧。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写入数据第一个字不在第一帧中，返回结束码 14(格式错误)。

如果指定的写入数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写入数据，返回结束码 A5，而不是 15。)

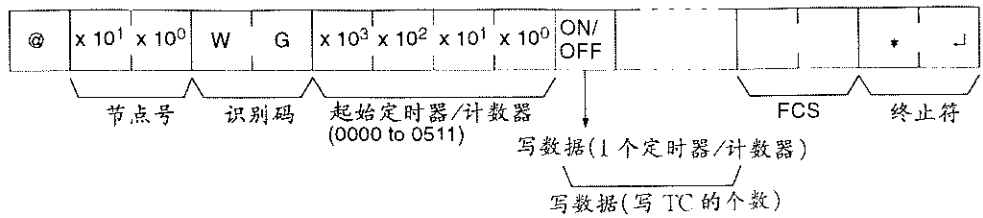
用上位机链接单元，如写入数据不是 BCD 码，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

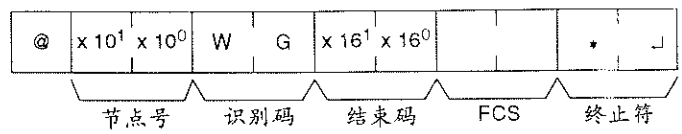
10－3－13 写 TC 状态——WG

将定时器/计数器的结束标志的状态写入 TC 区，从指定的定时器/计数器号开始，按号依次写入。结束标志 ON 表示定时器或计数器已计满。

命令格式



应答格式



PC设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写入数据第一个字不在第一帧中，返回结束码14(格式错误)。

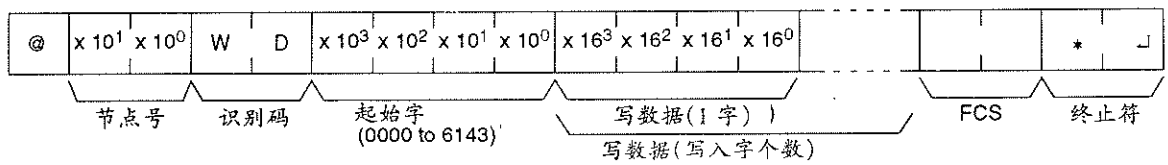
如果写入数据的数不是0或1，或指定的写入数超出数据区域，或起始字不是BCD码，或写入数据不是BCD码，返回结束码15(入口码数据错误)。(对多重命令帧中非十六进制写入数据，返回结束码A5，而不是15。)

结束代码	内 容
00	正常结束
01	在运行方式下不执行
13	FCS错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因FCS错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

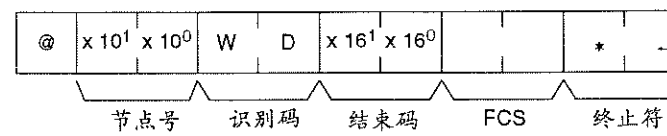
10－3－14 写DM区——WD

向DM区写数据，从指定字开始，按字依次写入。

命令格式



应答格式



限制

除第一个写入数据字外，其它写入数据可以分成多帧。

不存在DM6656～DM6999，这些字中写数据，不出现错误，同样，往UM区域中未分配的扩展DM字(DM7000～DM9999)中写数据，也不出现错误。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对，或写入数据第一个字不在第一帧中，返回结束码 14(格式错误)。

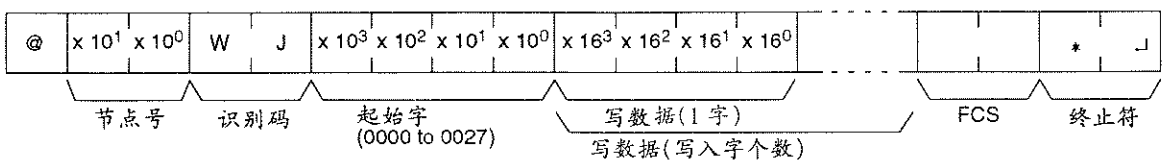
如果指定的写入数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写入数据返回结束码 A5，而不是 15。)

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
23	用户存储器处于写保护
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

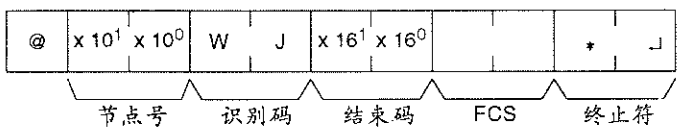
10－3－15 写 AR 区——WJ

向 AR 区写数据，从指定字开始，按字依次写入。

命令格式



应答格式



限制

除第一个写入数据字外，其它写入数据可以分成多帧。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写入数据第一个字不在第一帧中，返回结束码 14(格式错误)。

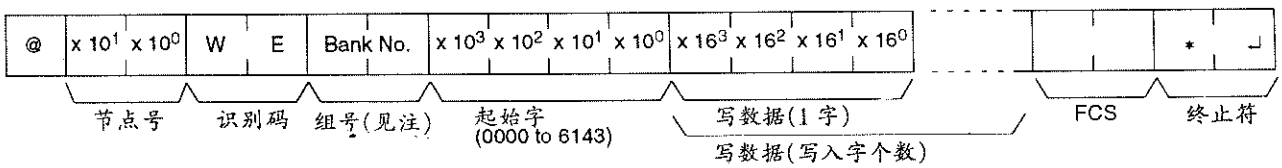
如果指定的写入数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中非十六进制写入数据，返回结束码 A5，而不是 15。)

结束代码	内 容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

10－3－16 写 EM 区——WE

将数据写入指定的 EM 区，从指定字开始，按字依次写入。

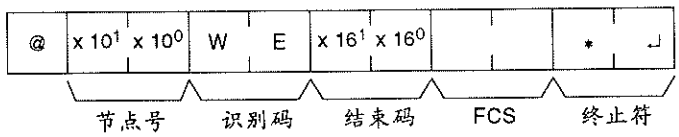
命令格式



注

输入 00，01，或 02 表示组号 0，1，或 2。输入两个空格表示当前组。

应答格式



限制

除了第一个写数据字外，其它写入数据可以分成多帧。

如果指定有效的组号，但 PC 未装入该 EM 组，命令正常完成，但未写入任何数据。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或写入数据第一个字不在第一帧中，返回结束码 14(格式错误)。

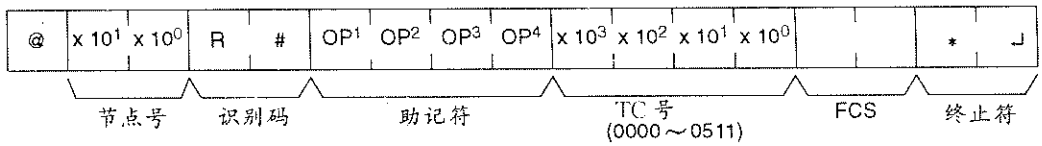
如果指定的写入数据超出数据区域，或起始字不是 BCD 码，或写入数据不是十六进制，返回结束码 15(入口码数据错误)。(对多重命令帧中，非十六进制写入数据，返回结束码 A5，而不是 15。)

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

10－3－17 SV 读 1——R#

用指定的 TC 号在用户程序中搜索指令 TIM，TIMH(15)，CNT，CNTR(12)或 TTIM(87)第一处，并读 PV 值，该值被假设为作为常数设置的值。被读的 SV 是 4 位十进制数(BCD)。

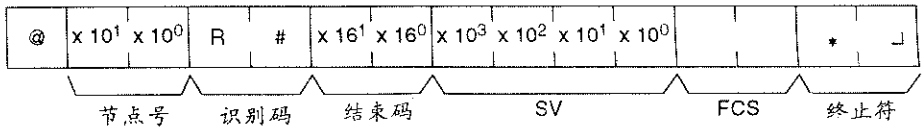
命令格式



指定定时器或计数器指令的助记符，必须是 4 个字符，在 TIM 或 CNT 后面需加一个空格以凑满 4 个字符。

指令名称	助记符				TC 号范围
	OP1	OP2	OP3	OP4	
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

应答格式



限制

指令的第二个字必须是从 0000～0511 的 BCD 常数。

如果一条指令在程序中使用一次以上，读第一个的 SV 值。

如果 UM 区读保护，命令不能执行。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	…

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对返回结束码 14(格式错误)。

如果指令助记符不对或 TC 号不正确，返回结束码 15(入口码数据错误)。

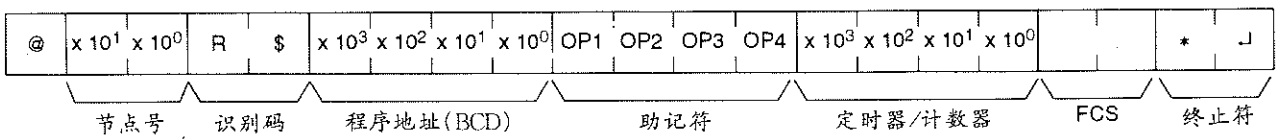
如果指定的指令在程序中不存在，返回结束码 16(命令不支持。)

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护

10－3－18 SV 读 2——R $

读常数 SV 或 SV 的存贮字地址，所读 SV 为一个十进制数(BCD 码)。在用户程序指定的程序地址中，该数作为指令 TIM、TIMH(15)、CNT、CNTR(12)或 TTIM(87)的第二操作数写入。

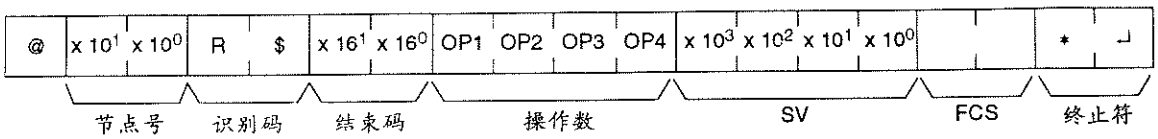
命令格式



指定定时器或计数器指令的助记符必须是4个字符，在TIM或CNT后面要加一个空格以凑满4个字符。

指令名称	助记符				TC号范围
	OP1	OP2	OP3	OP4	
定时器	T	1	M	（空格）	0000～0511
高速定时器	T	1	M	H	
计数器	C	N	T	（空格）	
可逆计数器	C	N	T	R	
累加定时器	T	T	1	M	

应答格式



“操作数”表示存贮SV的数据区或一个常数。“SV”表示字地址或者SV自身，如果SV是一个常数。

操作数				分类	常数或字地址
OP1	OP2	OP3	OP4		
C	I	O	（空格）	IR或SR	0000～0511
L	R	（空格）	（空格）	LR	0000～0063
H	R	（空格）	（空格）	HR	0000～0099
A	R	（空格）	（空格）	AR	0000～0027
D	M	（空格）	（空格）	DM	0000～6655
D	M	*	（空格）	DM(间址)	0000～6655
C	O	N	（空格）	常数	0000～9999
E	M	（空格）	（空格）	EM	0000～6143
E	M	*	（空格）	EM(间址)	0000～6143

限制

当UM只设置给梯形图程序区域，命令才有效。

如果UM区处于读保护、命令不能执行。

PC设置

PC方式			UM区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	…

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果程序地址高于程序最高地址，但小于 65,536(对 C200HS 为 32,767)，返回结束码 04(地址超出区域)。

如果命令长度不对，返回结束码 14(格式错误)。

如果程序地址不是 BCD 码或操作数/SV 参数不正确，返回结束码 15(入口码数据错误)。

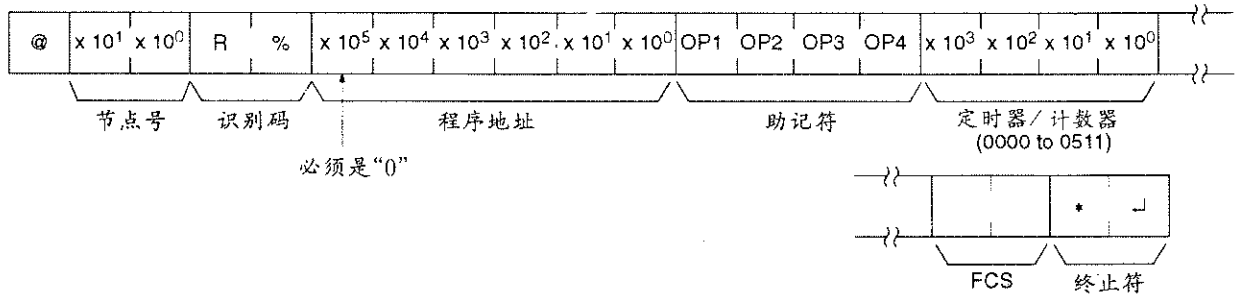
如果指定的指令在程序中不存在，返回结束码 16(命令不支持)。

结束代码	内　　容
00	正常结束
04	地址超出区域
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护

10－3－19　SV 读 3——R%

读常数 SV 或 SV 的存贮字地址，所读 SV 为一个十进制数(BCD 码)。在用户程序指定的程序地址中，该数写入指令 TIM、TIMH(15)、CNT、CNTR(12)或 TTIM(87)中的第二字。

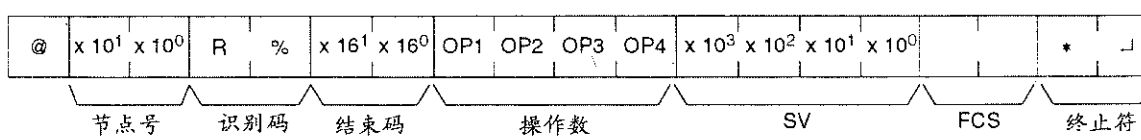
命令格式



指定定时器或计数器指令的助记符必须是 4 个字符。在 TIM 或 CNT 后面要加一个空格以凑满 4 个字符。

指　令　名　称	助　记　符				TC 号范围
	OP1	OP2	OP3	OP4	
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

应答格式



"操作数"表示存贮 SV 的数据区或一个常数。"SV"表示字地址或者 SV 自身，如果 SV 是一个常数。

操作数				分类	常数或字地址
OP1	OP2	OP3	OP4		
C	I	O	(空格)	IR 或 SR	0000～0511
L	R	(空格)	(空格)	LR	0000～0063
H	R	(空格)	(空格)	HR	0000～0099
A	R	(空格)	(空格)	AR	0000～0027
D	M	(空格)	(空格)	DM	0000～6655
D	M	*	(空格)	DM(间址)	0000～6655
C	O	N	(空格)	常数	0000～9999
E	M	(空格)	(空格)	EM	0000～6143
E	M	*	(空格)	EM(间址)	0000～6143

限制

当 UM 只设置给梯形图程序区域，命令才有效。

如果 UM 区处于读保护、命令不能操作。

不能指定 SR253～SR255。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	…

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

如果程序地址高于程序最高地址，但小于 65,536(对 C200HS 为 32,767)，返回结束码 04(地址超出区域)。

如果命令长度不对，返回结束码 14(格式错误)。

如果程序地址超过最大值 65,535(对 C200HS 为 32,767)，或程序地址不是 BCD 码，或助记符/SV 操作数不正确，返回结束码 15(入口码数据错误)。

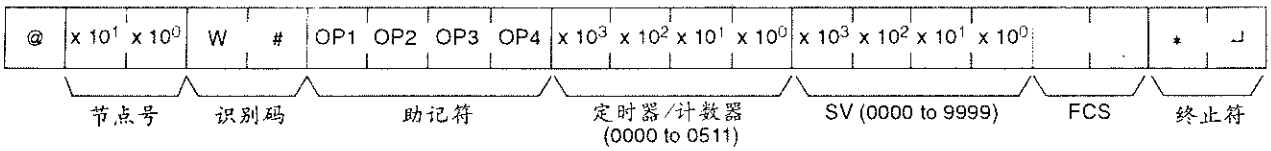
如果指定的指令在程序中不存在，返回结束码 16(命令不支持)。

结束代码	内容
00	正常结束
04	地址超出区域
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护

10－3－20 SV改变1——W#

在用户程序中搜索指定的TIM、TIMH(15)、CNT、CNTR(12)或TTIM(87)指令第一处，并将SV改变成在指令的第二字中指定的新常数。

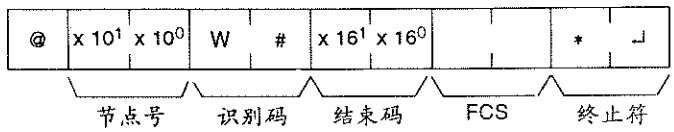
命令格式



指定定时器或计数器指令的助记符必须是4个字符。在TIM或CNT后面要加一个空格以凑满4个字符。

指令名称	助记符				TC号范围
	OP1	OP2	OP3	OP4	
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

应答格式



“操作数”表示存贮SV的数据区或一个常数。“SV”表示字址或者SV自身，如果SV是一个常数，。

操作数				分类	常数或字地址
OP1	OP2	OP3	OP4		
C	I	O	(空格)	IR或SR	0000～0511
L	R	(空格)	(空格)	LR	0000～0063
H	R	(空格)	(空格)	HR	0000～0099
A	R	(空格)	(空格)	AR	0000～0027
D	M	(空格)	(空格)	DM	0000～6655
D	M	*	(空格)	DM(间址)	0000～6655
C	O	N	(空格)	常数	0000～9999
E	M	(空格)	(空格)	EM	0000～6143
E	M	*	(空格)	EM(间址)	0000～6143

限制

如果SV不是从0000～9999的BCD数，命令不能执行。

如果UM区域处于写保护，命令不能执行。

如果相同指令在程序中使用一次以上，改变第一处指令中SV值。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

如果 SV 不是 BCD 码或助记符/TC 号不正确，返回结束码 15(入口码数据错误)。

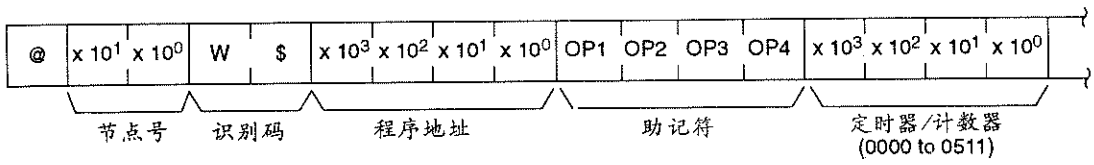
如果指定的指令在程序中不存在，返回结束码 16(命令不支持)。

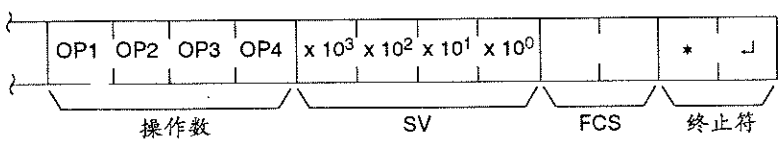
结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护。

10－3－21 SV 改变 2——W $

在用户程序中指定的程序地址上改变 TIM、TIMH(15)、CNT、CNTR(12)或 TTIM(87)的第二字内容，仅在程序长度小于 10K 时才执行命令。

命令格式





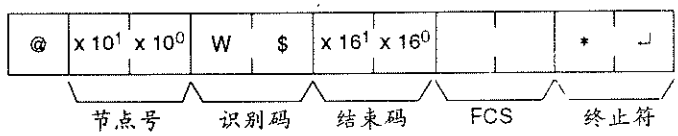
指定定时器或计数器指令的助记符必须是 4 个字符。在 TIM 或 CNT 后面要加一个空格以凑满 4 个字符。

指令名称	助记符				TC号范围
	OP1	OP2	OP3	OP4	
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

“操作数”表示存贮SV的数据区或一个常数。“SV”表示字地址或者SV自身，如果SV是一个常数。

操作数				分类	常数或字地址
OP1	OP2	OP3	OP4		
C	I	O	(空格)	IR或SR	0000～0511
L	R	(空格)	(空格)	LR	0000～0063
H	R	(空格)	(空格)	HR	0000～0099
A	R	(空格)	(空格)	AR	0000～0027
D	M	(空格)	(空格)	DM	0000～6655
D	M	*	(空格)	DM(间址)	0000～6655
C	O	N	(空格)	常数	0000～9999
E	M	(空格)	(空格)	EM	0000～6143
E	M	*	(空格)	EM(间址)	0000～6143

应答格式



限制

当UM只设置给梯形图程序区域时，命令才有效。

不能指定SR253～SR255。

如果UM区域处于写保护，命令不能执行。

PC设置

PC方式			UM区域	
运行	监控	编程	写保护	读保护
…	允许	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

如果程序地址不是 BCD 码，或程序地址超过程序中最大地址，或助记符/TC 号不正确，或 SV 不正确，返回结束码 15(入口码数据错误)。

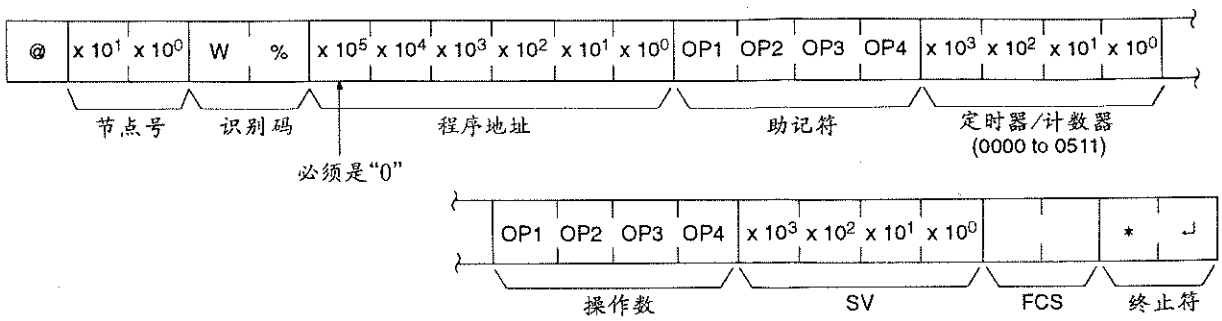
如果指定的指令在程序中不存在，返回结束码 16(命令不支持)。

结束代码	内　容
00	正常结束
01	在运行方式下不执行
04	地址超出区域
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护

10－3－22　SV 改变 3——W%

在用户程序中指定的程序地址上改变 TIM、TIMH(15)、CNT、CNTR(12)或 TTIM(87)的第二字内容。

命令格式



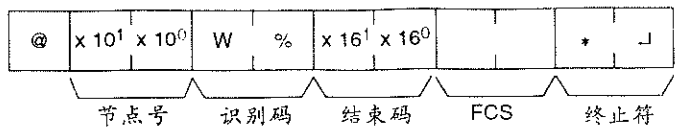
指定定时器或计数器指令的助记符必须是 4 个字符。在 TIM 或 CNT 后面要加一个空格以凑满 4 个字符。

指　令　名　称	助　记　符				TC 号范围
	OP1	OP2	OP3	OP4	
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

"操作数"表示存贮 SV 的数据区或一个常数。"SV"表示字地址或者 SV 自身，如果 SV 是一个常数。

操作数				分类	常数或字地址
OP1	OP2	OP3	OP4		
C	I	O	(空格)	IR 或 SR	0000～0511
L	R	(空格)	(空格)	LR	0000～0063
H	R	(空格)	(空格)	HR	0000～0099
A	R	(空格)	(空格)	AR	0000～0027
D	M	(空格)	(空格)	DM	0000～6655
D	M	*	(空格)	DM(间址)	0000～6655
C	O	N	(空格)	常数	0000～9999
E	M	(空格)	(空格)	EM	0000～6143
E	M	*	(空格)	EM(间址)	0000～6143

应答格式



限制

当 UM 只设置给梯形图程序区域时，命令才有效。

不能指定 SR253～SR255。

当 UM 区域处于写保护，命令不能执行。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

如果程序地址不是 BCD 码，或程序地址超过程序最大地址，或助记符/TC 号不正确，或 SV 值不正确，返回结束码 15(入口码数据错误)。

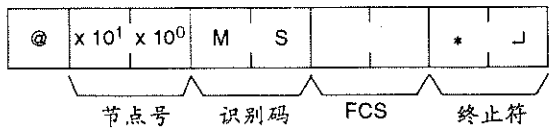
如果指定的指令在程序中不存在，返回结束码 16(命令不支持)。

结束代码	内　容
00	正常结束
01	在运行方式下不执行
04	地址超出区域
13	FCS 错误
14	格式错误
15	入口码数据错误
16	命令不支持
18	帧长度错误
23	用户存储器处于写保护

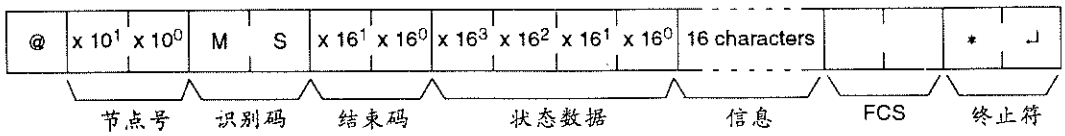
10－3－23　读状态——MS

读 PC 运行状态。

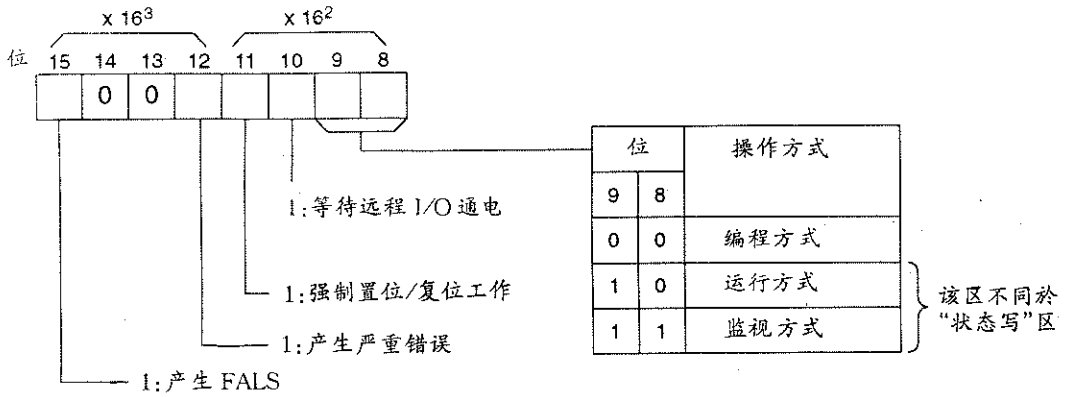
命令格式

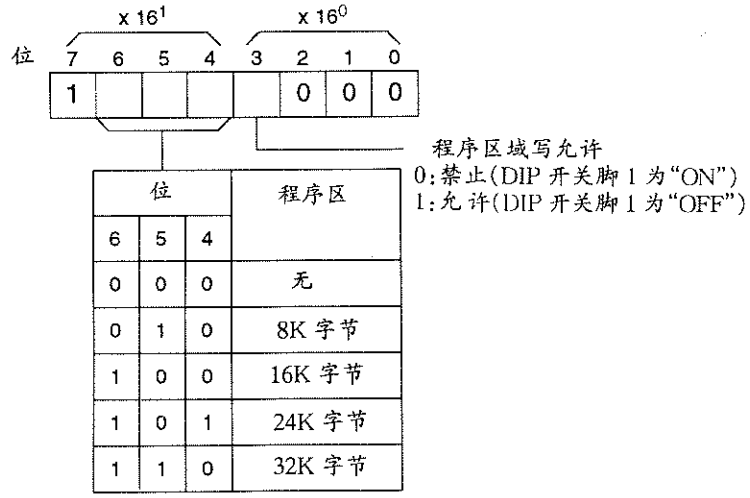


应答格式



“状态数据”由 4 个 16 进制数(2 字节)组成，最左字节表示 CPU 操作方式，最右字节则表示程序区长度。





PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

限制

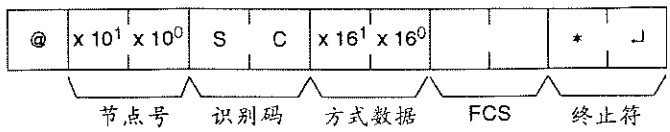
如果命令长度不正确，返回结束码 14(格式错误)。

结束代码	内　容
00	正常结束
13	FCS 错误
14	格式错误
18	帧长度错误

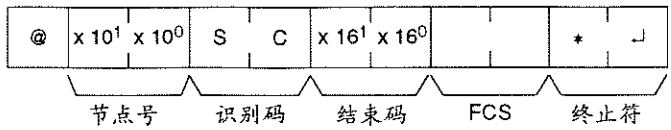
10－3－24　写状态——SC

改变 PC 的操作方式。

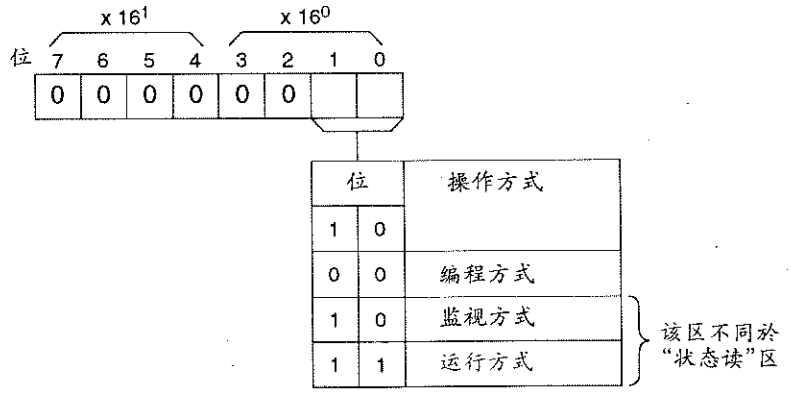
命令格式



应答格式



“方式数据”由 2 个 16 进制数(1 字节)组成，最左两位表示 PC 操作方式，其余位皆设置为“0”。



PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	…

结束码

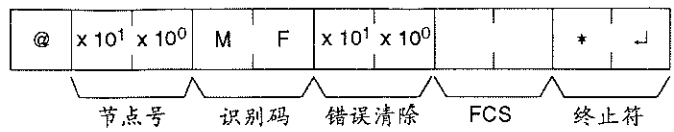
如果命令长度不正确，返回结束码 14(格式错误)。

结束代码	内　　容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
19	不执行

10－3－25　读错误——MF

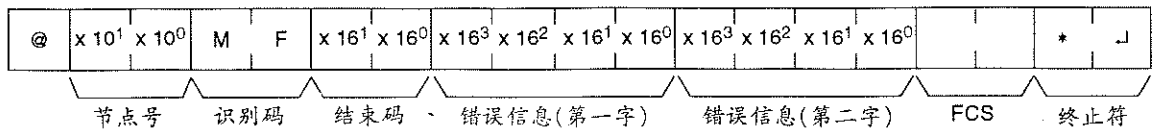
读并清除 PC 错误，同时检查先前的错误是否已擦除。

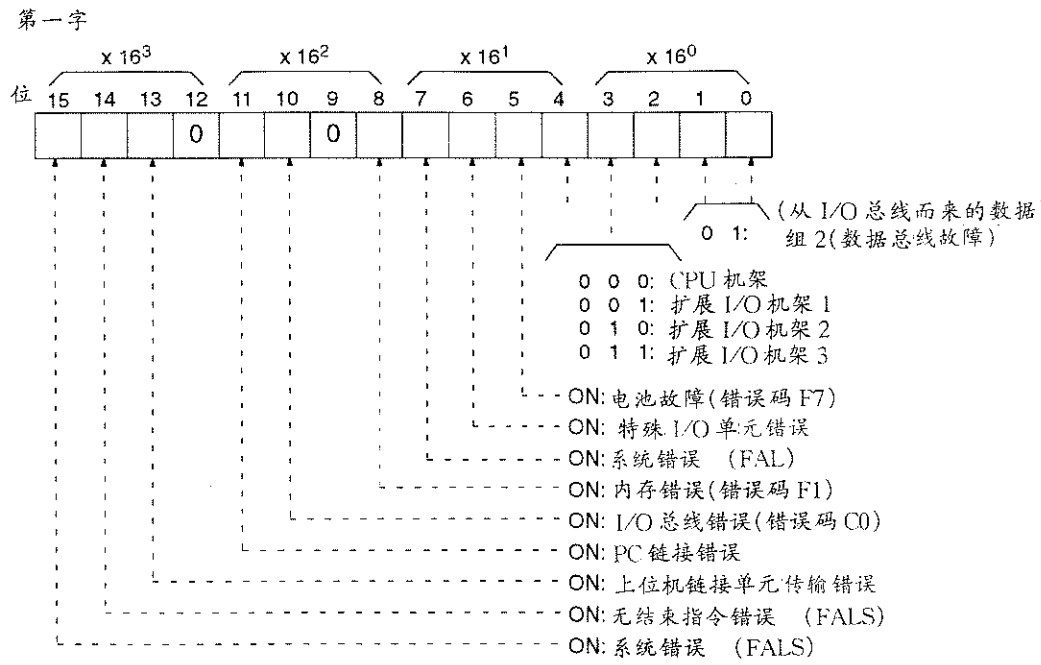
命令格式

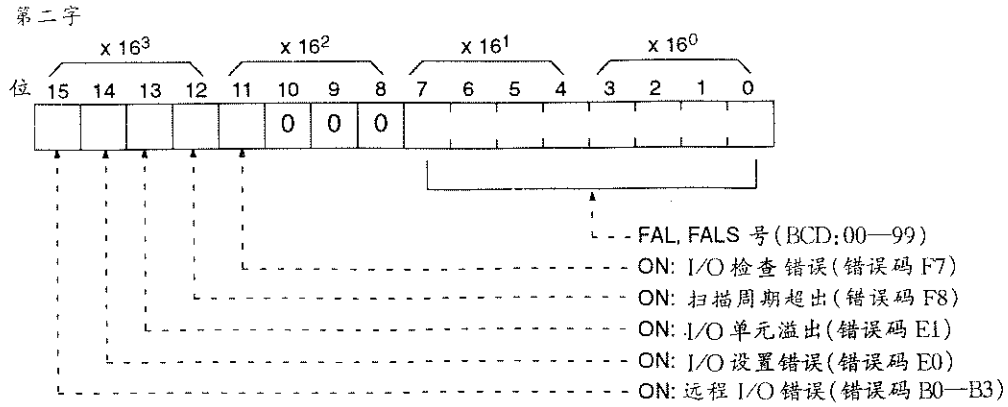


"错误清除"中 01 表示清除错误，00 表示不清除错误(BCD)。严重错误只有在 PC 处于编程方式才可清除。

应答格式







限制

当错误正被清除("错误清除"＝01)时,错误信息只有在错误清除功能执行完成之后才被读取。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对,返回结束码 14(格式错误)。

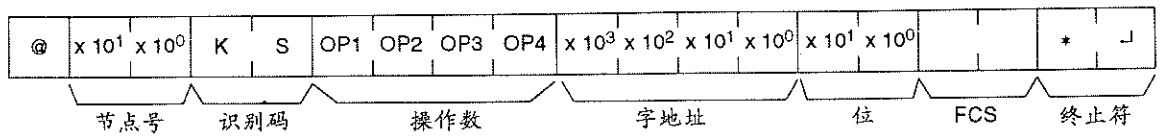
如果错误清除参数不是 0 或 1,返回结束码 15(入口码数据错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误

10－3－26 强制置位——KS

对 IR、SR、LR、HR、或 AR 位,或定时器或计数器结束标志强制置位,一旦某位被强制置位或复位,在强制置位/复位取消(FORCED SET/RESET CANCEL (KC))传送之前,其状态将被保持。

命令格式

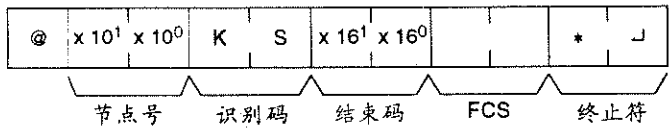


对 IR, SR, LR, HR, 和 AR 数据区,"操作数"表示强制置位的位所在的数据区,"字地址"表示字的地址。

对 TC 区域,"操作数"表示定时器或计数器指令的助记符,"字地址"表示 TC 号。

数据区 指令	操作数				字地址	位
	OP1	OP2	OP3	OP4		
IR 或 SR	C	I	O	(空格)	0000～0511	00～15
LR	L	R	(空格)	(空格)	0000～0063	
HR	H	R	(空格)	(空格)	0000～0099	
AR	A	R	(空格)	(空格)	0000～0027	
定时器	T	I	M	(空格)	0000～0511	总是 00
高速定时器	T	I	M	H		
计数器	C	N	T	(空格)		
可逆计数器	C	N	T	R		
累加定时器	T	T	I	M		
传送标志	T	N	(空格)	(空格)	0000～1023	

应答格式



限制

不能指定 SR253～SR255 中的位。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不正确，返回结束码 14(格式错误)。

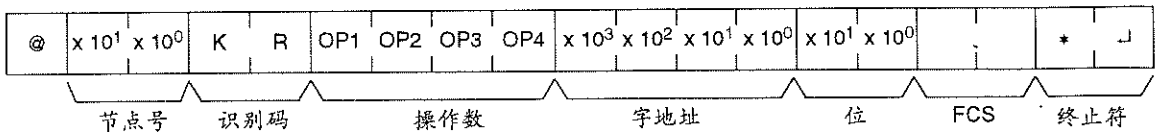
发果操作数，字地址或位地位设置不正确，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误

10－3－27 强制复位——KR

对 IR、SR、LR、HR、或 AR 位或定时器或计数器结束标志强制复位，一旦某位被强制置位或复位，在强制置位/复位取消传送之前，其状态保持不变。

命令格式

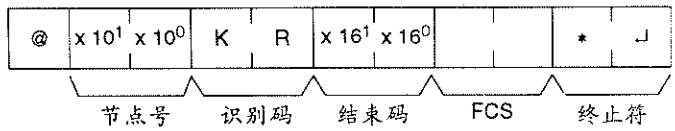


对 IR、SR、LR、HR 和 AR 数据区，“操作数”表示强制置位的位所在的数据区，“字地址”表示字的地址。

对 TC 区域，“操作数”表示定时器或计数器指令的助记符，“字地址”表示 TC 号。

数据区 指令	操作数				字地址	位
	OP1	OP2	OP3	OP4		
IR 或 SR	C	I	O	(空格)	0000～0511	00～15
LR	L	R	(空格)	(空格)	0000～0063	
HR	H	R	(空格)	(空格)	0000～0099	
AR	A	R	(空格)	(空格)	0000～0027	
定时器	T	I	M	(空格)	0000～0511	总为 00
高速定时器	T	I	M	H		
计数器	C	N	T	(空格)		
可逆计数器	C	N	T	R		
累加定时器	T	T	I	M		
传送标志	T	N	(空格)	(空格)	0000～1023	

应答格式



限制

不能指定 SR253～SR255 中的位。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

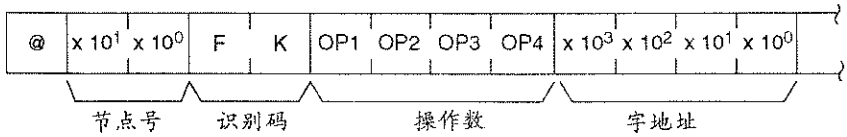
如果操作数，字地址，或位地位设置不正确，返回结束码 15(入口码数据错误)。

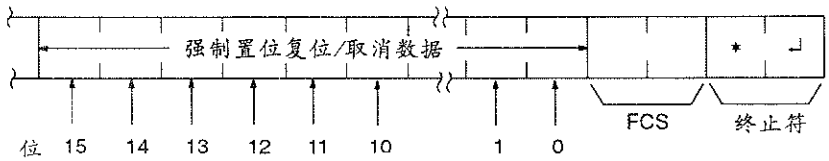
结束代码	内　容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误

10－3－28　多重强制置位/复位——FK

对 IR、SR、HR 或 AR 区中一个字的位或一个定时器/计数器结束标志状态进行强制置位、强制复位及取消。

命令格式





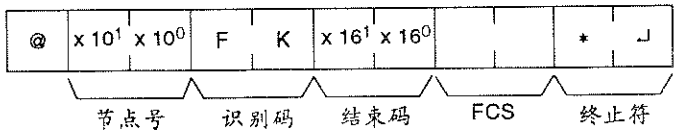
对 IR、SR、LR、HR 和 AR 数据区，“操作数”表示强制置位的位所在的数据区，“字地址”表示字的地址。对 TC 区域，“操作数”表示定时器或计数器指令的助记符，“字地址”表示 TC 号。

数据区 指令	操作数				字地址	位
	OP1	OP2	OP3	OP4		
IR 或 SR	C	I	O	（空格）	0000～0511	00～15
LR	L	R	（空格）	（空格）	0000～0063	
HR	H	R	（空格）	（空格）	0000～0099	
AR	A	R	（空格）	（空格）	0000～0027	
定时器	T	I	M	（空格）	0000～0511	总为 00
高速定时器	T	I	M	H		
计数器	C	N	T	（空格）		
可逆计数器	C	N	T	R		
累加定时器	T	T	I	M		
传送标志	T	N	（空格）	（空格）	0000～1023	

强制置位/复位/取消的 16 位数据，对指定的数据区域字或结束标志规定所要作的操作。如果指定定时器或计数器，则用位 15 对结束标志的状态强制置位或强制复位，其余位可忽略。仅对定时器或计数器才可用强制置位和强制复位。

强制置位/复位/取消数据	操　作	强制置位/复位/取消数据	操　作
0000	位状态不变	0100	强制复位
0010	复位	0101	强制置位
0011	置位	1000	强制置位/复位取消

应答格式



限制

不能指定 SR253～SR255 中的位。

仅有 15 个定时器/计数器或 15 个传送标志可以置位/复位。

当指定传送标志时，不可检查 UM 设置，即只要传送标志地址不超过 1023，即使远程 PC 中不存在指定的标志，命令仍照常执行。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。(强制置位/复位/取消数据长 16 字节。)

注

用上位机链接单元，如果数据位 15 指定定时器/计数器或传送标志，将不返回结束码 14(格式错误)。(强制置位/复位/取消数据可以单字节或 16 字节长。)

如果操作数，字地址或位地址不正确，返回结束码 15(入口码数据错误)。如果给定时器或计数器结束标志，或传送标志使用数据 0 或 1，同样返回结束码 15。

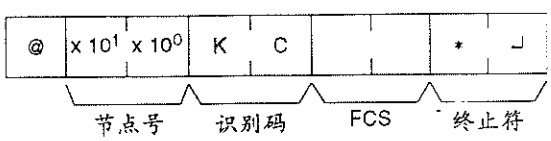
如果设置 TC 时，使用[$C, $D]为强制数据，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
01	在运行方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误

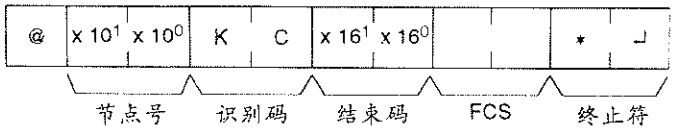
10－3－29 强制置位/复位取消——KC

取消所有被强制设置及强制复位的位(包括被强制置位、强制复位及多重强制置位/复位所设置的那些位)。如多个位被设置，其所有强制状态将被取消。

命令格式



应答格式



PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
…	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	…

结束码

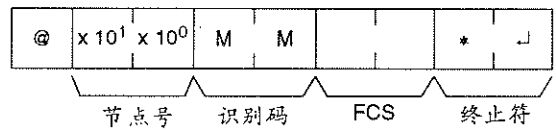
如果命令长度不对，返回结束码 14(格式错误)。

结束代码	内 容
00	正常结束
13	FCS 错误
14	格式错误
18	帧长度错误

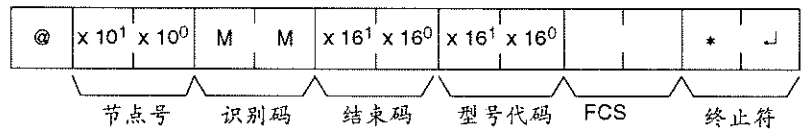
10－3－30 读 PC 型号——MM

读 PC 的型号。该命令返回代码和 NT 链接代码不一样。

命令格式



应答格式



"型号代码"用二位十六进制数表示 PC 的型号。

型号代码	型 号
01	C250
02	C500
03	C120
0E	C2000

（续表）

型号代码	型　　号
10	C1000H
11	C2000H/CQM1
12	C20H/C28H/C40H, C200H, C200HS, C200HX/HG/HE
20	CV500
21	CV1000
22	CV2000
40	CVM1－CPU01－E
41	CVM1－CPU11－E
42	CVM1－CPU21－E

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

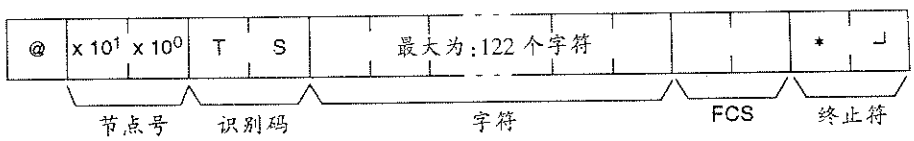
结束代码	内　　容
00	正常结束
13	FCS 错误
14	格式错误
18	帧长度错误

10－3－31　测试——TS

从上位机返传一个数据块，但内容不改变。

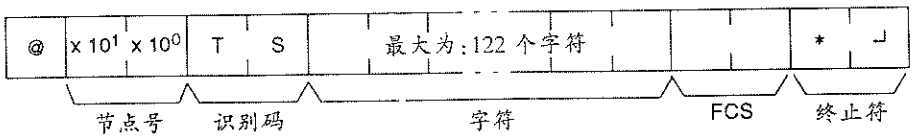
命令格式

规定除回车(CHR $ (13))外任何字符。



应答格式

如果测试顺利，返传命令中规定的字符，内容不变。



PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

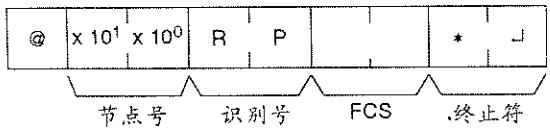
如果第一帧中未接收到终止符，返回结束码 14(格式错误)。

结束代码	内容
13	FCS 错误
14	格式错误
18	帧长度错误

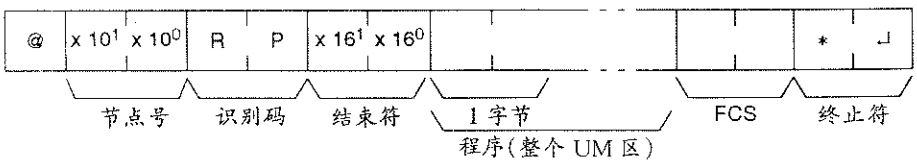
10－3－32 读程序——RP

读用机器语言(目标码)写的 PC 用户程序区域中内容。该内容从头至尾以块形式读出。

命令格式



应答格式



限制

当 UM 只设置给梯形图程序区域时，命令才有效。

如果 UM 区域处于写保护，命令不能执行。

数据从梯形图区域开头起一直读至程序区域的末尾(例如，对 16K 字的 UM 容量从 $A06C0 至 $A7FBF，UM 区域不分配给扩展 DM 或 I/O 注释。)

起始读地址和应答长度取决于 UM 区域设置(如分配给扩展 DM 的存储器容量)。

应答第一帧最多可以包含 30 个程序数据字，第二和以后各帧最多可包含 31 字。

对该命令的多帧应答，可用初始化(INITIALIEE)和中止(ABORT)命令代替定界符。如果发送其它命令，它们同样作定界符处理。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	…

执行条件

命　令		应　答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

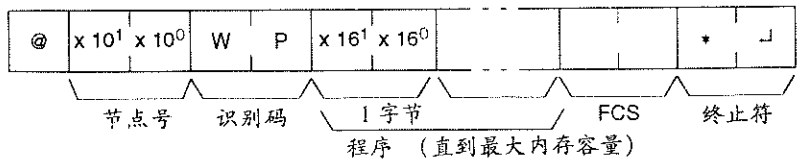
如果命令长度不对，返回结束码 14(格式错误)。

结束代码	内　容
00	正常结束
13	FCS 错误
14	格式错误
18	帧长度错误
23	用户存储器处于写保护
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

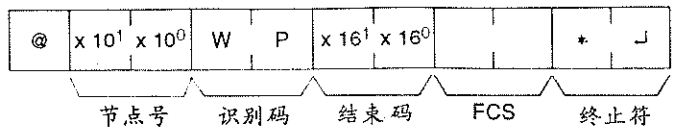
10－3－33　写程序——WP

将由上位机传送来的机器语言(目标码)程序写入 PC 用户程序区。该内容从头开始，以块形式写入。

命令格式



应答格式



限制

当 UM 只设置给梯形图程序区域时，命令才有效。

如果 UM 区域处于写保护，命令不能执行。

数据从梯形图区域开始处写至程序区域最大极限处。下面有二个例子。

1. 当 UM 容量为 16K 字，从 $A06C0 至 $A7FBF，UM 区域未分配给扩展 DM 或 I/O 注释。

2. 当 UM 容量为 16K 字，从 $A06E0 至 $A7FBF 有 1000 个字分配给扩展 DM，未分配给 I/O 注释。

起始写地址和程序最大容量取决于 UM 区域设置(如分配给扩展 DM 的存储器容量)。

如果试图超出程序区域最大范围写入程序，不会产生错误。

程序数据可以分成多帧。

PC 设置

PC　方　式			UM　区　域	
运行	监控	编程	写保护	读保护
…	…	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

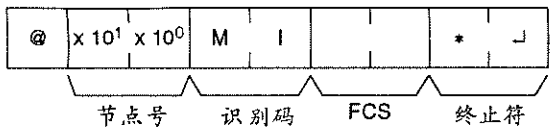
如果命令长度不对(整个程序字节数为奇数)或者第一帧中无程序数据,返回结束码 14(格式错误)。
如果指定的写数据不是十六进制,返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
01	在运行模式下不执行
02	在监控方式下不执行
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
19	不执行
23	用户存储器处于写保护
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

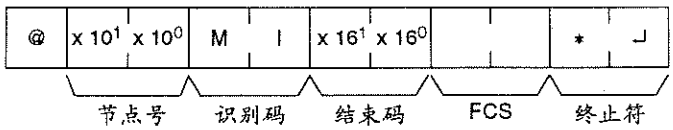
10—3—34 I/O 表生成——MI

修正寄存的 I/O 表,以匹配实际的 I/O 表。

命令格式



应答格式



PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
…	…	允许	…	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

结束码

如果命令长度不对，返回结束码 14(格式错误)。

结束代码	内　容
00	正常结束
01	在运行方式下不执行
02	在监控方式下不执行
03	UM 写保护
13	FCS 错误
14	格式错误
18	帧长度错误
20	不能生成 I/O 表

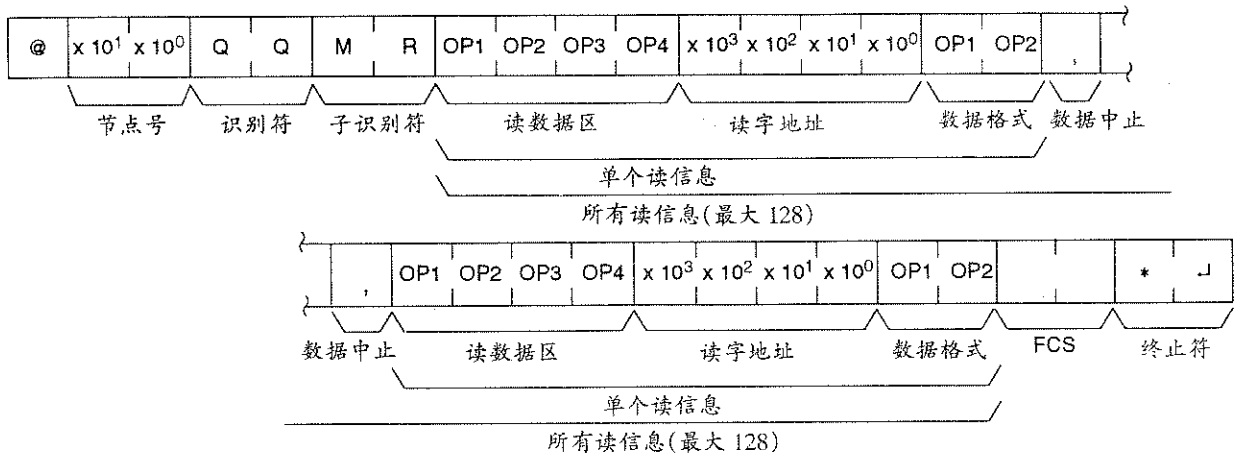
注

该命令的写保护错误代码与其它命令的不一样。

10－3－35　组合命令——QQMR

在 PC 上寄存所有将被读取的位，字，或定时器/计数器，并以批形式读所有的这些状态。寄存信息保存在 PC 中，直至被组合命令改写或 PC 断电。

命令格式



“读数据区”表示数据区，“读字地址”表示字地址，“数据格式”表示位号或整个字(CH)。

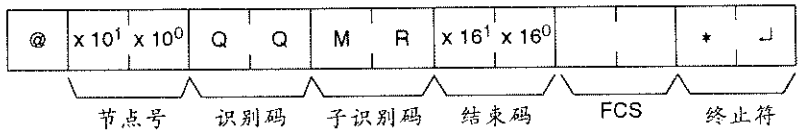
数据区 指令	操　作　数				字　地　址
	OP1	OP2	OP3	OP4	
IR 或 SR	C	I	O	(空格)	0000～0511
LR	L	R	(空格)	(空格)	0000～0063
HR	H	R	(空格)	(空格)	0000～0099
AR	A	R	(空格)	(空格)	0000～0027
定时器	T	I	M	(空格)	0000～0511
高速定时器	T	I	M	H	
计数器	C	N	T	(空格)	
可逆计数器	C	N	T	R	
累加定时器	T	T	I	M	

（续表）

数据区 指令	操作数				字地址
	OP1	OP2	OP3	OP4	
DM	D	M	（空格）	（空格）	0000～6655
EM(当前组)	E	M	（空格）	（空格）	0000～6143
EM(组 0)	E	M	0	0	0000～6143
EM(组 1)	E	M	0	1	0000～6143
EM(组 2)	E	M	0	2	0000～6143

每个字或每个位，每次用一个中止代码(,)分隔指定为一个项。可被分项的最大项数为 128，如果指定了 EM 区域中的一个地址，最大项数则为 100。（当指定定时器/计数器的 PV 值，同时返回结束标志的状态，因此必须按二项计算。）

应答格式



限制

数据从头开始登记，直到某一错误，假如，命令试图登记 129 项目，将产生格式长度错误(结束码 18)，但前面 128 个项目被登记。

DM6656～DM6999 不存在，如果试图登记这些字，不会出现错误。

位和字可以任意次序登记，它们将以指定的次序被登记。

数据可以分别为若干帧。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	允许	允许	…

结束码

如果命令长度不对或二项之间的数据中止符“,”被遗漏，返回结束码 14(格式错误)。

如果“读数据区”，“读字地址”，或“数据格式”设定不正确，返回结束码 15(入口码数据错误)。

结束代码	内容
00	正常结束
13	FCS 错误
14	格式错误
15	入口码数据错误
18	帧长度错误
19	不执行

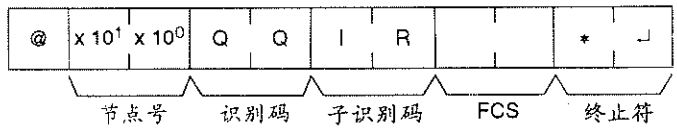
（续表）

结束代码	内　容
A3	传送数据时因 FCS 错误引起中止
A4	传送数据时因格式错误引起中止
A5	传送数据时因入口码错误引起中止
A8	传送数据时因帧长度错误引起中止

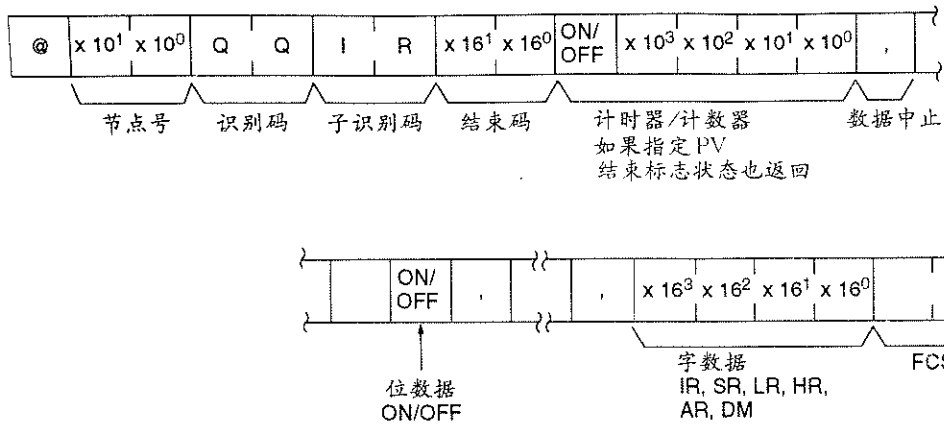
10－3－36　组合命令读数据——QQIR

按照用 QQMR 命令登记的读信息，以批形式读取位，字和定时器/计数器的状态。

命令格式



应答格式



限制

数据以 QQMR 登记的相同次序读取。

对不存在的字，如 DM 6656～DM 6999，未分配的扩展 DM，或不存在的 EM 组，返回应答数据“0000”。

对该命令的多帧应答可使用初始化（INITIALIEE）和中止（ABORT）命令代替定界符。如果发送其它命令，它们同样作定界符处理。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
允许	…	允许	允许

结束码

如果命令长度不对，返回结束码 14（格式错误）。

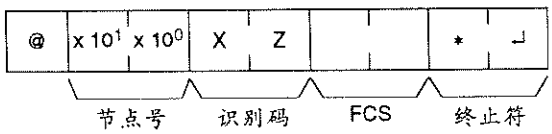
如果没有登记过的数据，返回结束码 19（不执行）。

结束代码	内　　容
结束代码	内　　容
00	正常结束
13	FCS 错误
14	格式错误
18	帧长度错误
19	不执行
A3	传送数据时因 FCS 错误引起中止
A8	传送数据时因帧长度错误引起中止

10－3－37　中止——XZ

中止当前正在执行的上位机链接命令，允许接收下一命令。中止命令不接收应答信号。

命令格式



限制

用该命令可以取消一个命令的多帧应答。

该命令即使没有 FCS 码和终止符仍然有效。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命　令		应　答	
单个	多重	单个	多重
允许	…	…	…

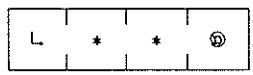
结束码

该命令没有结束码。

10－3－38　初始化——＊＊

对所有与上位机连接的 PC 的传输控制过程进行初始化。初始化命令不使用节点号或 FCS，也不接收应答信号。

命令格式



限制

用该命令可以取消一个命令的多帧应答。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	…	…

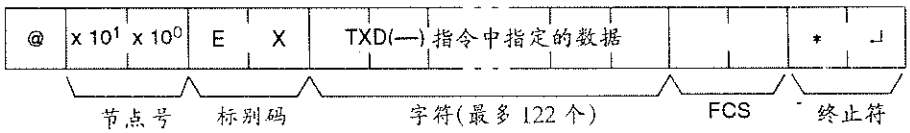
结束码

该命令没有结束码。

10－3－39 TXD 应答——EX

这是 PC 在上位机链接方式下执行 TXD(一)指令时的应答格式。(TXD(一)转换指定数据的 ASCII 码,然后用这一格式传送至上位机。)

应答格式



限制

该格式最多可包括 122 个字符。(TXD(一)指令不支持多帧格式。)

没有和 EX 相呼应的命令。

PC 设置

PC 方式			UM 区域	
运行	监控	编程	写保护	读保护
允许	允许	…	允许	允许

执行条件

命令		应答	
单个	多重	单个	多重
允许	…	允许	…

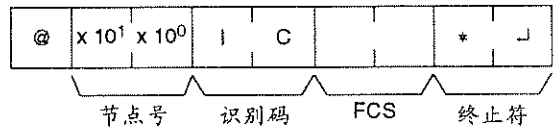
结束码

该命令没有结束码。

10－3－40 未定义命令——IC

如果命令的识别码无法解码,该应答信号返回,请检查识别码。

应答格式



限制

如果一个命令的@字符和终止符之间数据少于 6 个字节,或使用不正确的标识码,或格式有错误,返回

这个出错应答。

没有和 IC 相呼应的命令。

PC 设置

PC 方 式			UM 区 域	
运行	监控	编程	写保护	读保护
允许	允许	允许	允许	允许

执行条件

命 令		应 答	
单个	多重	单个	多重
…	…	允许	…

结束码

该命令没有结束码。

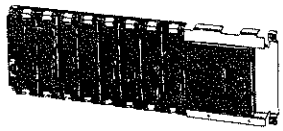
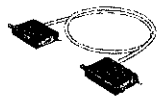
附　录A
标准模块

CPU机架

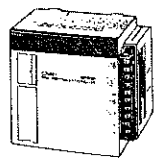
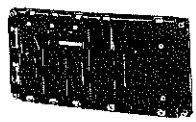
名　称	规　格				型　号
CPU(除CPU－11E,所有型号都具有时钟功能和通信板槽口。) 	UM	DM	I/O点	RS－232C	—
	3.2K字	4K字	640	No	C200HE－CPU11－E
	7.2K字	6K字	880	No	C200HE－CPU32－E
				Yes	C200HE－CPU42－E
	15.2K字	12K字	880	No	C200HG－CPU33－E
				Yes	C200HG－CPU44－E
			1,184	No	C200HG－CPU53－E
				Yes	C200HG－CPU63－E
	31.2K字	24K字	880	No	C200HX－CPU34－E
				Yes	C200HX－CPU43－E
			1,184	No	C200HX－CPU54－E
				Yes	C200HX－CPU64－E
电源单元 	100～120/200～240VAC				C200HW－PA204
	100～120/200～240VAC(带24VDC输出端子)				C200HW－PA204S
	24VDC				C200HW－PD024
CPU I/O底板 	3槽				C200HW－BC031
	5槽				C200HW－BC051
	8槽				C200HW－BC081
	10槽				C200HW－BC101
通信板 	用于SYSMAC LINK和SYSMAC NET链接单元的通信端口				C200HW－COM01
	RS－232C端口				C200HW－COM02
	RS－422/485端口				C200HW－COM03
	用于SYSMAC LINK和SYSMAC NET链连单元的通信端口和一个通信协议宏功能				C200HW－COW04－E
	二个RS－232C端口和一个通信协议等功能				C200HW－COM05－E
	一个RS4212/485端口,一个RS－232C端口和一个通信协议宏功能				C200HW－COM06－E
存储器盒 	EEPROM	4K字			C200HW－ME04K
		8K字			C200HW－ME08K
		16K字			C200HW－ME16K(见注)
		32K字			C200HW－ME32K
	EPROM	16K字/32K字			C200HS－MP16K
		相当于27256,150ns,12.5V			ROM－JD－B
		相当于27512,150ns,12.5V			ROM－KD－B

注:C200HW－ME16K不久将推出。EEPROM存储器盒可以使用C200HS－ME16K。

扩展 I/O 机架

名　称	规　格		型　号
电源单元 	100～120/200～240 VAC		C200HW－PA204
	100～120/200～240 VAC(带 24VDC 输出端子)		C200HW－PA204S
	24 VDC		C200HW－PD024
扩展 I/O 底板 	3 槽		C200HW－BI031
	5 槽		C200HW－BI051
	8 槽		C200HW－BI081
	10 槽		C200HW－BI101
I/O 连接电缆 	30 cm	一个网络中连接电缆总长最长 12 米。	C200H－CN311
	70 cm		C200H－CN711
	200 cm		C200H－CN211
	500 cm		C200H－CN521
	1,000 cm		C200H－CN131

从站机架

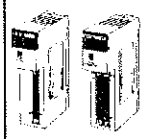
名　称		规　格		型　号
从站机架	远程 I/O 从站单元 	100～120/200～240 VAC(可选)	APF/PCF	C200H－RT001－P
		24 VDC		C200H－RT002－P
		100～120/200～240 VAC(可选)	线缆	C200H－RT221
		24 VDC		C200H－RT202
底板 		3 槽		C200H－BC101－V2
		5 槽		C200H－BC081－V2
		8 槽		C200H－BC051－V2
		10 槽		C200H－BC101－V2
I/O 组		输入	规定 12 VDC 或 21 VDC。	G71－IC16
		输出		G71－OD16
I/O 端子	AC 输入	规定 100 VAC 或 200 VAC。		G7TC－IA16
	DC 输入	规定 12 VDC 或 24 VDC。		G7TC－ID16
	输出	规定 12 VDC 或 24 VDC。		G7TC－OC16

I/O 单元

名称		规格		型号
输入单元	AC 输入单元	8 点	100～120 VAC	C200H－IA121
		16 点	100～120 VAC	C200H－IA122/IA122V
		8 点	200～240 VAC	C200H－IA221
		16 点	200～240 VAC	C200H－IA222/IA222V
	DC 输入单元	8 点	12～24 VDC	C200H－ID211
		16 点	24 VDC	C200H－ID212
	AC/DC 输入单元	8 点	12～24 VAC/DC	C200H－IM211
		16 点	24 VAC/DC	C200H－IM212
	中断输入单元（见注）	8 点	12～24 VDC	C200HS－INT01
输出单元	继电器输出单元	8 点	2A, 250 VAC/24 VDC(阻性负载)	C200H－OC221
		12 点	2 A, 250 VAC/24 VDC(阻性负载)	C200H－OC222
		5 点	2 A, 250 VAC/24 VDC(阻性负载) 独立公共端	C200H－OC223
		8 点	2 A, 250 VAC/24 VDC(阻性负载) 独立公共端	C200H－OC224
		16 点	2 A, 250 VAC/24 VDC(阻性负载) (见注)	C200H－OC225
	双向可控硅输出单元	8 点	1 A, 120 VAC	C200H－OA121－E
		8 点	1 A, 200 VAC	C200H－OA221
		12 点	0.3 A, 200 VAC	C200H－OA222
	晶体管输出单元	8 点	1 A, 12～48 VDC	C200H－OD411
		12 点	0.3 A, 24 VDC	C200H－OD211
		16 点	0.3 A, 24 VDC(见注)	C200H－OD212
		8 点	2.1 A, 24 VDC	C200H－OD213
		8 点	0.8A, 24VDC;有源型(PNP);带负载短路保护	C200H－OD214
		8 点	0.3 A, 5～24 VDC;有源型(PNP)	C200H－OD126
		12 点	0.3 A, 5～24 VDC;有源型(PNP)	C200H－OD217
		16 点	1.0A, 24VDC;有源型(PNP);带负载短路保护	C200H－OD21A
模拟定时器单元		4 定时器	0.1～1 s/1～10 s/10～60 s/1 min～10 min (可选)	C200H－TM 001
	可变电阻连接器	1 个外部可变电阻连接器带 2 米连线		C4K－CN223
B7A 接口单元		15 或 16 输入点	连接 B7A 链接终端。标准传输延时。	C200H－B7AI 1
		16 输出点	连接 B7A 链接终端。标准传输延时。	C200H－B7AO1(见注)

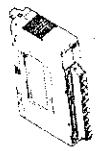
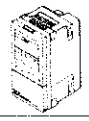
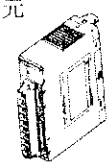
注：如果中断输入单元安装在扩展 I/O 机架上，不能使用其中断功能，中断输入单元仅作为一般 8 点输入单元。但是，中断输入单元不能用在从站机架上。另外，中断输入单元要求 CPU 机架使用 2 型底板(即带－V2 后缀的型号)若使用早期型号底板，不能使用其中断功能。当底板上安装 C200H－OC225, C200H－OD212, 或 C200H－B7AO1, 确认底板型号带后缀"－V1"或"－V□"。C200H－OC225, C200H－OD212 或 C200H－B 7AO1 不能安装在任何型号中不带后缀"－V1"或"－V□"的底板。

组－2 I/O 单元

名称		规格		型号
	DC 输入单元	64 点	12 VDC	C200H－ID111
		32 点	24 VDC	C200H－ID216
		64 点		C200H－ID217
	晶体管输出单元	32 点	16 mA4.5V～100 mA26.4V	C200H－OD218
		64 点		C200H－OD219
B7A 接口单元		32 输入点	连接 B7A 接口终端，标准或高速传输延时	C200H－B7A12
		32 输出点		C200H－B7A02
		16 输入点和 16 输出点		C200H－B7A21
		32 输入点和 32 输出点		C200H－B7A22

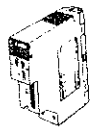
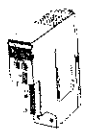
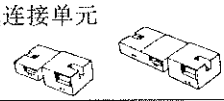
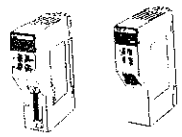
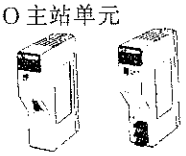
特殊 I/O 单元

名称		规格		型号
高密度 I/O 单元(见注)	DC 输入单元	32 点	5 VDC(TTL 输入)；带高速输入	C200H－ID501
		32 点	24VDC；带高速输入	C200H－ID215
	晶体管输出单元	32 点	0.1A，24VDC(可用作 128 点动态输出单元)	C200H－OD215
		32 点	35mA，5VDC(TTL 输出)(可用作 128 点动态输出单元)	C200H－OD501
	DC 输入/晶体管输出单元	16 输入点和 16 输出点	24VDC 输入；带高速输入；0.1A，24VDCVDC 输出(可用作 128 点动态输入单元)	C200H－MD215
		16 输入点和 16 输出点	5VDCTTL 输入；带高速输入；35mA，5VDCTTL 输出(可用作 128 点动态输入单元)	C200H－MD501
		16 输入点和 16 输出点	12VDCTTL 输入；带高速输入；12VDCTTL 输出(可用作 128 点动态输入单元)	C200H－MD115
模拟 I/O 单元	模拟输入单元	4～20 mA，1～5v/0～10v(可选)；4 输入；12 位		C200H－AD001
		4～20 mA，1～5v/0～10v/－10v～＋10v(可选)：8 输入；12 位或 BCD 码		C200H－AD002
	模拟输出单元	4～20 mA，1～5v/0～10v(可选)；2 输出		C200H－DA001
模糊逻辑单元		用模糊支持软件编程，最多 8 点输入和 4 点输出。		C200H－FZ001
	模糊支持软件	可提供 3.5 或 5.25 软盘		C500－SU981－E
温度传感器单元		热电偶	K(CA)或 J(IC)，可转换；4 输入	C200H－TS001
			K(CA)或 L(Fe－CuNi)DIN 标准；4 输入	C200H－TS002
		铂热电阻	Pt 100Ω；4 输入	C200H－TS101
			Pt 100Ω；4 输入；DIN 和 1989JIS 标准	C200H－TS102
温度控制单元		热电偶	晶体管输出	C200H－TC001
			电压输出	C200H－TC002
			电流输出	C200H－TC003
		铂热电阻	晶体管输出	C200H－TC101
			电压输出	C200H－TC102
			电流输出	C200H－TC103

名　称	规　格		型　号
热/冷温度控制单元 	热电偶	晶体管输出	C200H－TV001
		电压输出	C200H－TV002
		电流输出	C200H－TV003
	铂热电阻	晶体管输出	C200H－TV101
		电压输出	C200H－TV102
		电流输出	C200H－TV103
凸轮定位器单元 	通过分相器检测旋转角度，并在规定角度提供 ON 和 OFF 信号。最多可使用 48 个凸轮输出(16 个外部输出和 32 个内部输出)。		C200H－CP 114
数据设定器 	用来设置和监控温度控制单元，凸轮定位器单元，PID 控制单元和热/冷温度控制单元中数据。		C200H－DSC01
连接电缆	2m		C200H－CN225
	4m		C200H－CN425
PID 控制单元 	晶体管输出；4～20mA/1～5v/0～5v/0～10v 输入(可选)		C200H－PID01
	电压输出；4～20mA/1～5v/0～5v/0～10v 输入(可选)		C200H－PID02
	电流输出；4～20mA/1～5v/0～5v/0～10v 输入(可选)		C200H－PID03
位置控制单元 	1 轴	脉冲输出；速度 1～100,000PPS	C200H－NC111
	1 轴	脉冲输出；直接连伺服驱动器，与线驱动器匹配；速度；1～250,000PPS	C200H－NC112
	2 轴	脉冲输出；速度 1～250,000PPS，每轴 53 点	C200H－NC211
高速计数器单元 	1 轴	脉冲输出；计数速度：50KCPS，5VDC/12VDC/24VDC	C200H－CT001－V1
	1 轴	脉冲输入；计数速度：75KCPS；RS－422 线驱动器	C200H－CT002
	2 轴	脉冲输入；计数速度：75KCPS；RS－422 线驱动器；7 位 BCD 码	C200H－CT021
ASCII 单元 	内置 24K 字节 RAM 和 24K 字节 EEPROM。		C200H－ASC02
ID 传感器单元	本地使用，电磁耦合		C200H－IDS01－V1
	远程使用；微波发送		C200H－IDS21
读/写头 	电磁型		V600－H 系列
	微波型		V620－H 系列
数据载体	SRAM 型配置 V600－H 系列		V600－D□□R□□
	EEPROM 型 V600－H 系列		V600－D□□P□□
语音单元 	最多 60 条信息；信息长度 32.48，或 64 秒(可选)		C200H－OV001
连接电缆	RS－232C		C200H－CN224

注：高密度 I/O 单元作为特殊 I/O 单元安装在从站机架上时，远程 I/O 主站必须是 C200H－RM001－PV1 或 C200H－RM201。

通信单元

名称	规格		型号
SYSMACLINK 单元（同轴电缆）	总线连接单元必须单独订货。	数据链接表:918 字	C200HW－SLK23
		数据链接表:2,966 字	C200HW－SLK24
 终端	系统末端的每个节点 1 个。		C1000H－TER01
连接夹	随 SYSMACLINK 单元一起提供。		C200H－TL001
适配器	接网络		C100H－CE001
适配器盖	接网络		C100H－COV01
SYSMACLINK 单元（光纤电缆）	用 H－PCF 光缆连接。总线连接单元必须单独订货。	数据链接表:918 字	C200HW－SLK13
		数据链接表:2.966 字	C200HW－SLK14
 电源适配器	提供后备电源时需要。	用于 1 个或 2 个单元	C200H－APS03
电源电缆	连接电源适配器和 SYSMACNET 链接单元	用于 1 个单元	C200H－CN111
		用于 2 个单元	C200H－CN211
SYSMACLINK 支持板（同轴电缆）	连接 IBM PC/AT 或兼容机作为 SYSMACLINK 系统中的节点		3G8F5－SLK21－E
SYSMACNET 链接单元	总线连接单元必须单独订货。		C200HS－SNT32
 电源适配器	供应后备电源时需要	用于 1 个单元	C200H－APS01
		用于 2 个单元	C200H－APS02
电源电缆	连接电源适配器和 SYSMACNET 链接单元	包括在 C200H－APS01 内	用于 1 个单元	C200H－CN001
		包括在 C200H－APS02 内	用于 2 个单元	C200H－CN002
总线连接单元 	连接 SYSMACLINK 单元或 SYSMACNET 链接单元和 C200HW－COM 01/COM 04－E 通信板		用于 1 个单元	C200HW－CE 001
			用于 2 个单元	C200HW－CE002
上位机链接单元 	机架安装	C200H, C200HS, C200HE, C200HG, C200HX	APF/PCF	C200H－LK101－PV1
			RS－422	C200H－LK202－V1
			RS－232C	C200H－LK201－V1
PC 链接单元 	单级:32 个单元 多级:16 个单元		RS－485	C200H－LK401
远程 I/O 主站单元 	每个 PC 最多二个主站,5 个从站单元		APF/PCF	C200H－PM001－PV1
			线缆	C200H－RM201
远程 I/O 从站单元	见产品目录开头部分的“机架”			

SYSMACNET/SYSMACLINK 硬件

名称	规格	型号
SYSMACNET 网络支持板	用于 IBM PC/AT 或兼容机	S3200－NSB11－E
SYSMACLINK 网络支持板	用于 IBM PC/AT 或兼容机,同轴电缆连接器	3G8F5－SLK21－E

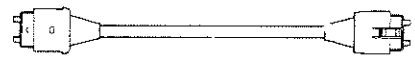
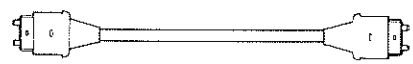
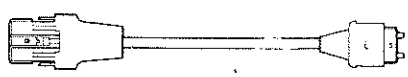
链接适配器

名　称	规　格	型　号
链接适配器 	RS－422 连接器 3 个	3G2A9－AL001
	光连接器(AFP/PCF)3 个	3G2A9－AL002－PE
	光连接器(PCF)3 个	3G2A9－AL002－E
	RS－232C 连接器 1 个，RS－422 连接器 2 个	3G2A9－AL003
	APF/PCF，RS－422 和 RS－232C 连接器各 1 个	3G2A9－AL004－PE
	PCF，RS－422 和 RS－232C 连接器各 1 个	3G2A9－AL004－E
	APF/PCF 和 AGF 连接器各 1 个	3G2A9－AL005－PE
	PCF 和 AGF 连接器各 1 个	3G2A9－AL005－E
	APF/PCF 连接器 1 个，AGF 连接器 2 个	3G2A9－AL006－PE
	PCF 连接器 1 个，AGF 连接器 2 个	3G2A9－AL006－E
	O/E 变换器，RS－485 和 APF/PCF 连接器各 1 个	B500－AL007－P
	用于 SYSMAC，NET 链接系统中的 SYSMACNET 链接单元在线拆卸，SYSMACNET 光缆链接适配器，HPF/PCF 连接器 3 个	B700－AL001

光纤产品

用于 SYSMACNET 和 SYSMACLINK 光纤电缆

带连接器的 H－PCF 光纤电缆

系　统	外　观	型　号
SYSMAC NET SYSMAC LINK		S3200－CN□□□－20－20
		S3200－CN□□□－20－25
		S3200－CN□□□－20－62
		S3200－CN□□□－25－25
		S3200－CN□□□－25－62
		S3200－CN□□□－62－62

型号

上述电缆型号规定了电缆类型,长度和连接器型号。

$$\underset{1}{\text{S3200-CN}}\underset{2}{\square\square\square}\underset{3}{-20-25}$$

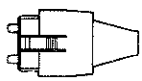
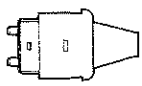
1. S3200－CN 指定 H－PCF 光纤电缆
2. 方框(□□□)用表示标准电缆长度的代码替代,如下所示。
   欲购更长电缆,请与 OMRON 代理商议。定购长电缆时略去方框表示的部分,以米为单位单独指明其长度。例如,S3200－CN－20－20,30m。

代 码	长 度	代 码	长 度
201	2m	152	15m
501	5m	202	20m
102	10m	省略	大于 20m

3. 型号最后一部分(如,20～25)确定连接器,如下所示。

代 码	连 接 器
20	S3200－COCF2011
25	S3200－COCF2511
62	S3200－COCH 62M

可使用的光纤连接器

型号/外观	可连接单元	
	SYSMAC NET	STSMAC LINK
S3200－COCF2011 	CV500－SNT31	CV500－SLK11 C1000H－SLK11
S3200－COCF2511 	C200HS－SNT32	C200HW－SLK13/14
S3200－COCH62M 	S3200－LSU03－01E S3200－NSUA1－00E S3200－NSUG4－00E S3200－NSB11－E C500－SNT31－V4 B 700－AL001	…

用于 SYSMACBUS 的全塑光纤电缆

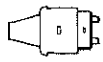
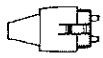
名 称	规 格	型 号	标 准
全塑光纤电缆	仅光缆;订货要求长度 5～100 m 间,以 5 米为增加单位,或以 200 米或 500 米为增加单位。	3G5A2－PF002	…
光连接器 A	二个光连接器(棕色),用于 APF(10 m max.)	3G5A2－CO001	
光连接器 B	二个光连接器(黑色),用于 APF(8～20 m)	3G5A2－CO002	
全塑光纤电缆组	1 米光缆,两端各有一个光连接器 A	3G5A2－PF101	
光纤加工工具	附件:125mm 摄子(Muromoto Tekko's 550M)用于 APF	3G2A9－TL101	

用于 SYSMACBUS 的塑料护套光纤电缆

名 称	规 格		型 号	标 准
塑料护套光纤电缆(户内型)	0.1m,带连接器	环境温度: -10°~70°C	3G5A2 - OF011	…
	1m,带连接器		3G5A2 - OF101	
	2m,带连接器		3G5A2 - OF201	
	3m,带连接器		3G5A2 - OF301	
	5m,带连接器		3G5A2 - OF501	
	10m,带连接器		3G5A2 - OF111	
	20 m,带连接器		3G5A2 - OF211	
	30m,带连接器		3G5A2 - OF311	
	40m,带连接器		3G5A2 - OF411	
	50 m,带连接器		3G5A2 - OF511	
	仅光缆,订货要求的长度 1~500 m 间,以 1 米为增加单位。		3G5A2 - OF002	
	仅光缆,订货要求的长度 501~800 m 间,以 1 米为增加单位。	环境温度: 0°~55°C (无日光直接照射)		

H-PCF 光纤电缆(用于 SYSMAC NET, SYSMAC LINK 和 SYSMAC BUS)

名 称	规 格		型 号	标 准
光纤电缆 SYSMAC NET, SYSMAC LINK	10m,黑色	混合电缆,包括二芯光缆和二芯电力电缆	S3200 - HCLB101	…
	50m,黑色		S3200 - HCLB501	
	100m,黑色		S3200 - HCLB102	
	500m,黑色		S3200 - HCLB502	
	1,000m,黑色		S3200 - HCLB103	
	10m,橙色		S3200 - HCLO101	
	50m,橙色		S3200 - HCLO501	
	100m,橙色		S3200 - HCLO102	
	500m,橙色		S3200 - HCLO502	
	1,000m,橙色		S3200 - HCLO103	
光纤电缆 SYSMAC NET, SYSMAC LINK, SYSMAC BUS, SYSMAC WAY	10m,黑色	二芯光缆	S3200 - HCLB101	…
	50m,黑色		S3200 - HCCB501	
	100m,黑色		S3200 - HCCB102	
	500m,黑色		C3200 - HCCB502	
	1000m,黑色		C3200 - HCCB103	
	10m,橙色		S3200 - HCCO101	
	50m,橙色		S3200 - HCCO501	
	100m,橙色		S3200 - HCCO102	
	500m,橙色		S3200 - HCCO502	
	1,000m,橙色		S3200 - HCCO103	
	10m,黑色	二芯光缆	S3200 - HBCB101	…
	50m,黑色		S3200 - HRCB501	
	100m,黑色		S3200 - HBCB102	
	500m,黑色		S3200 - HBCB502	
	1,000m 黑色		S3200 - HBCB103	

名 称	规 格		型 号	标 准
光纤电缆连接器 	SYSMAC NET: S3200－LSU03－01E B700－AL001 C500－SNT31－V4	用于 NSU, NSB 和 C500 SYSMAC NET 链接单元的全锁连接器	S200－COCH62M	…
	SYSMAC BUS: C200HH－RM 001－PV1 C200H－RT001/RT002－P C500－RM 001－(P)V1 C500－RT001/RT002(P)V1 3G2A9－□□□(－P)	用于远程 I/O主站，远程 I/O 从站，上位链接单元，和链接适配器的半锁定连接器	S3200－COCH82	
	SYSMAC NET/SYSMAC LINK C200HS－SNT32 C200HW－SLK13/14	半锁定连接器	S3200－COCF2511	
	SYSMACNET/SYSMACLINK CV500－SNT31 CV500－SLK11 CV1000H－SLK11	全锁定连接器	S3200－COCF2011	
	连接所有 SYSMAC NET 节点的中继	COCF62M 和 COCF62F 成对使用	S3200－COCF62M	
			C3200－COCF62F	

注：1. 光纤电缆必须由专业人员配备和连接。

2. 如用户自行配置和连接光纤电缆，用户必须接受由 Samitomo 由气工业有限公司主办的培训，并获得合格证书。

3. 连接光纤电缆需要光功率测试仪，源头单元，主光纤仪和光纤装配工具。

4. 您也可使用前二页所列出的带连接器的塑料护套光纤电缆/全塑光纤电缆。

光纤装配工具

名 称	规 格	型 号	标 准
光纤装配工具	为光传输系统，如 SYSMAC C－和 CV－系列 SYSMAC BUS, SYSMAC LINK 和 SYSMAC NET 系统，连接 H－PCF 和压切式连接器用。	S3200－CAK1062	…

注：1. 光纤电缆必须由专业人员配备和连接。

2. 连接光纤电缆需要光功率测试仪，源头单元，主光纤仪和光纤装配工具。

光功率测试仪

名　称	规　格	源头单元	型　号	标　准
光功率测试仪(见注) (附连接转换器,光源单元,小型单脚插头,硬机箱,和 AC 转换器)	SYSMAC NET: CV500 – SNT31 C200HS – SNT32	S3200 – CAT2002(随同测试仪供货)	S3200 – CAT2000	…
	SYSMAC LINK: CV500 – SLK11 C200HW – SLK13/14 CV1000H – SLK11	S3200 – CAT2702(随同测试仪供货)	S3200 – CAT2700	
	SYSMAC BUS: C200H – RM001 – PV1 C200H – RT001/RT002 – P C500 – RM 001 – (P)V1 C500 – RT001/RT002 – (P)V1	S3200 – CAT2822(随同测试仪供货)	S3200 – CAT2820	
	SYSMAC NET: S3200 – LSU03 – 01E S500 – SNT31 – V4	S3200 – CAT3202(随同测试仪供货)	S3200 – CAT3200	

注:源头单元与用光源单元和光功率测试仪的连接转换器相同。

源头单元

名　称	规　格	型　号	标　准
源头单元(包括光源单元和连接适配器组成)(见注)	SYSMAC NET: CV500 – SNT31 C200HS – SNT32	S3200 – CAT2002	…
	SYSMAC LINK: CV500 – SLK11 C200HW – SLK13/14 CV1000H – SLK11	C3200 – CAT2702	
	SYSMAC BUS: C500 – RM 001 – (P)V1 C500 – RT001/RT002 – (P)V1 C200H – RM 001 – PV1 C200H – RT001/RT002 – PV1	S3200 – CAT2822	
	SYSMAC NET: S3200 – LSU 03 – 01E C500 – SNT31 – V4	S3200 – CAT3202	

注:对要用的光槽块,需用适当的光源头单元。如果使用二种类型的光模块(单元型和插件型)请订购光功率测试仪外加合适型号的光源头单元。

主光纤仪

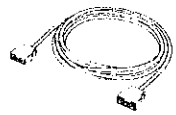
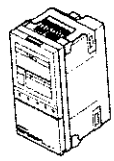
名　称	规　格	型　号	标　准
主光纤仪(lm)	S3200 – CAT3202(SYSMAC NET, NSB, NSU, 网桥)	S3200 – CAT3201	
	S3200 – CAT2002/CAT2702 (SYSMAC NET, SYSMAC LINK)	S3200 – CAT2001H	
	S3200 – CAT2822(SYSMAC BUS)	S3200 – CAT2821	

注:1. 主光纤仪和光功率测试仪一起使用,检测与光纤连接器连接的光纤电缆的光纤能级。

2. 光纤电缆必须由专业人员配备和连接。

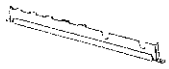
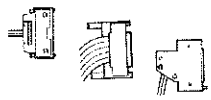
3. 连接光纤电缆,需要光功率测试仪,光源头单元,主光纤仪和光纤装配工具。

编程设备

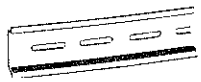
名 称	规 格		型 号	标 准
编程器 	手持式,带背光		C200H－PRO27－E	U,C
	包括一根二米长连接电缆		CQM1－PRO01－E	U,C
编程器安装支架 	用来将手持式编程器固定在面板上。		C200H－ATT01	…
编程器 连接电缆 	用于手持式编程器	2m	C200H－CN222	…
		4m	C200H－CN422	…
数据设定器 	用于 C200H－TC□□□, C200H－TV□□□, C200H－CP114,和 C200H－PID0□等单元数据输入和过程数值显示。		C200H－DSC01	
数据设定器连接电缆	用于 C200H－DSC01	2m	C200H－CN225	
		4m	C200H－CN425	
连接电缆 	连接 IBM PC/AT 或兼容机和 C200HX/HG/HE。	3.3m	CQM 1－CIF02	…

可选用产品

名 称	规 格	型 号	标 准
I/O 单元盖 	10 脚端子板罩盖	C200H－COV11	…
端子板盖 	10 脚端子板短路保护(每包 1 个),8 点	C200H－COV02	
	19 脚端子板短路保护(每包 10 个);12 点	C200H－COV03	
连接器盖 	用来连接电缆连接器的保护盖。	C500－COV01	
空单元 	用于空槽	C200H－SP 001	N,L
电池组 	仅用于 C200H 或 C200HRAM 存储器单元	C 200H－BAT09	…
继电器 	24VDC	GBB－1174P－FD－US	…

名 称	规 格	型 号	标 准
底板绝缘板 	用于 C200HW－BC031(3 槽 CPU 底板)	C200H－ATT31	…
	用于 C200HW－BC051(5 槽 CPU 底板)	C200H－ATT51	
	用于 C200HW－BC081(8 槽 CPU 底板)	C200H－ATT81	
	用于 C200HW－BC101(10 槽 CPU 底板)	C200H－ATTA1	
	用于 C200HW－BI031(3 槽 I/O 底板)	C200HW－ATT32	
	用于 C200HW－BI051(5 槽 I/O 底板)	C200HW－ATT52	
	用于 C200HW－BI081(8 槽 I/O 底板)	C200HW－ATT82	
	用于 C200HW－BI101(10 槽 I/O 底板)	C200HW－ATTA2	
I/O 支架 	用于 3 槽底板	C200H－ATT33	N
	用于 5 槽底板	C200H－ATT53	
	用于 8 槽底板	C200H－ATT83	
	用于 10 槽底板	C200H－ATTA3	…
外部连接器 	焊接端子;40 脚和 1 个连接器外壳	C500－CE 401	…
	非焊接端子;40 脚和 1 个连接器外壳(压切型)	C500－CE402	…
	压接端子;40 脚	C500－CE403	…
	焊接端子;40 脚和 1 个连接器外壳(水平型)	C500－CE404	…
	压切型端子;40 脚和 1 个连接器外壳(水平型)	C500－CE405	…

安装导轨和附件

名 称	规 格	型 号	标 准
DIN 导轨安装支架 	1 组(含 2 件)	C200H－DIN01	…
DIN 导轨 	长:50 cm;高:7.3 cm	PFP－50N	
	长:1 cm;高:7.3 cm	PFP－100N	
	长:1 cm;高:16 cm	PFP－100N2	
顶端板 	…	PFP－M	
定位架 	…	PFP－S	

注:DIN 导轨,顶端板和定位架订货,以十件为一个单位。

支持软件(SSS 中文版)

名 称	规 格	型 号	标 准
SYSMAC 支持软件 (用于 C20,C□□P,C□□K,C120, C□□H,C200H,C200HS,C200HE, C200HG,C200HX,C500,C1000H, C2000H,CQM1,和 CVM1)	3.5",2HD 软盘用于 IBM PC/AT 兼容机 注:1.0 版本不支持 C200HX/HG/HE 的附加功能。	C500－ZL3AT1－C	…

通信协议支持软件

名 称	规 格	型 号	标 准
通信协议支持软件	3.5",2HD 用于 IBM PC/AT 兼容机	C200HW－ZW3AT1－E	…

培训材料

名 称	规 格	型 号	标 准
SYSMAC 培训系统	包括教科书,磁带,和输入开关板	C200H－ETL01－E	…
模糊培训系统	包括模糊培训系统手册,一个主要单元,C200H－MR831 存储器单元,C200H－PRO27－E 编程器,编程器用电缆 C200H－CN222,C500－SU981－E 模糊培训软件,RS－232C 电缆,和一根传送带。	C200H－ETL13－E	…

附　录 B

编程指令

PC 指令既可通过键入相对应的编程器控制键来输入(例如 LD, AND, OR, NOT),也可使用功能代码来输入。用功能代码输入一条指令时,需按下 FUN 键、功能代码,然后按下 WRITE 键。下面列出了详细的编程指令。

代　码	助 记 符	名　称	功　能	页　数
—	AND	与	指定位的状态和执行条件逻辑与。	142
—	AND LD	逻辑块与	原先块的结果逻辑与。	142
—	AND NOT	与非	指定位取反且和执行条件逻辑与。	142
—	CNT	计数器	减计数器。	158
—	LD	取	由指定位状态开始指令行或用 AN-DLD 和 OR LD 定义逻辑块。	142
—	LD NOT	取非	用指定位状态取反开始指令行。	142
—	OR	或	指定位的状态和执行条件逻辑或。	142
—	OR LD	逻辑块或	原先块的结果逻辑或。	142
—	OR NOT	或非	指定位的状态取反且和执行条件逻辑或。	142
—	OUT	输出	条件为 ON 时,接通操作数位,条件为 OFF 时,断开操作数位。	143
—	OUT NOT	输出非	条件为 ON 时,断开操作数位,条件为 OFF 时,接通操作数位。	143
—	RSET	复位	条件为 ON 时,断开操作数位,条件为 OFF 时,不影响操作数位的状态。	145
—	SET	置位	条件为 ON 时,接通操作数位,条件为 OFF 时,不影响操作数位的状态	145

（续表）

代 码	助 记 符	名 称	功 能	页 数
—	TIM	定时器	接通延迟型(减1)定时器操作。	152
00	NOP	空操作	空操作，不执行任何内容，程序移到下条指令。	151
01	END	结束	程序结束时使用。	151
02	IL	联锁	如果联锁条件是OFF，所有的输出断开，在这条IL指令和下条ILC指令之间所有定时器的当前值(PV)复位，其它指令都作NOP指令处理，计数器当前值(PV)保持不变。	148
03	ILC	联锁解除		148
04	JMP	跳转	如果转移条件是OFF，在JMP指令和相对应的JME指令之间的所有指令被忽略。	150
05	JME	跳转终了		150
(@)06	FAL	错误报警与复位	产生一个非致命错误信息，把已设定的FALS号输出到编程器。	286
07	FALS	严重故障报警	产生一个致命错误信息，把已设定的FALS号输出到编程器。	286
08	STEP	步定义	当使用一个控制位时，定义一个新步的起始并清除前面一步，当不带N使用时，定义步执行的结束。	278
09	SNXT	步启动	使用控制位来指示步执行的结束，清除步，开始下一个步执行。	278
10	SFT	移位寄存器	建立一个移位寄存器。	162
11	KEEP	保持	定义一位作为一个锁存器，通过置位和复位输入来控制。	146
12	CNTR	可逆计数器	每当增量输入信号或减量输入信号分别由OFF变成ON，那么PV就加“1”或减“1”。	161
13	DIFU	上升沿微分	输入信号的上升沿接通指定的位一个周期。	144
14	DIFD	下降沿微分	输入信号的下降沿接通指定的位一个周期。	144
15	TIMH	高速定时器	高速，接通延迟型定时器。	156
(@)16	WSFT	字移位	在起始字和结束字之间以字为单元移动数据，把零写入起始字。	169
17～19	用于扩展指令。			
20	CMP	单字比较	比较两个字的内容，结果输出到GR、EQ和LE标志位。	185
(@)21	MOV	传送	把源数据(字或常数)复制到目标字。	171
(@)22	MVN	取反传送	把源数据(字或常数)取反，然后复制到目标字。	172
(@)23	BIN	BCD码→二进制	把源字的4位BCD码数据转换成16位二进制数据，且将转换好的数据输出到结果字中。	195
(@)24	BCD	二进制→BCD码	把源字的二进制数据转换成BCD码，且将转换好的数据输出到结果字中。	196
(@)25	ASL	算术左移	单字中的每位向左移动一位，带进位CY。	166

（续表）

代 码	助 记 器	名 称	功 能	页 数
(@)26	ASR	算术右移	单字中的每位向右移动一位，带进位 CY。	167
(@)27	ROL	循环左移	单字中的每位循环左移一位，带进位 CY。	167
(@)28	ROR	循环右移	单字中的每位循环右移一位，带进位 CY。	168
(@)29	COM	求反	一个字的数据各位状态取反。	264
(@)30	ADD	BCD 码加法	两个 4 位 BCD 码数相加且加上 CY 的值，将结果输出到指定的结果字中。	218
(@)31	SUB	BCD 码减法	从另一个 4 位的 BCD 码数中减去一个 4 位的 BCD 码数和 CY 的值，且将结果输出到指定的结果字中。	221
(@)32	MUL	BCD 码乘法	两个 4 位的 BCD 码数相乘，且将结果输出到指定字中。	225
(@)33	DIV	BCD 码除法	两个 4 位的 BCD 码数相除，且将结果输出到指定字中。	226
(@)34	ANDW	逻辑与	两个 16 位的输入字逻辑与，如果两个输入字中的对应位都是 ON，那么在结果字中的对应位置“1”。	265
(@)35	ORW	逻辑或	两个 16 位的输入字逻辑或，若两个输入字中的对应位有一位为 ON 或两位都为 ON，则结果字中的对应位置“1”。	266
(@)36	XORW	异或门逻辑	两个 16 位的输入字异或，当两个输入字中相对应位的状态相反时，结果字中的对应位置“1”。	267
(@)37	XNRW	异或非门逻辑	两个 16 位的输入字异或非，当两个输入字中相对应位的状态相同时，结果字中的对应位置“1”。	267
(@)38	INC	BCD 码加 1	4 位 BCD 码字加“1”。	217
(@)39	DEC	BCD 码减 1	4 位 BCD 码字减“1”。	218
(@)40	STC	置进位	进位标志置“1”(即 CY 为 ON)。	218
(@)41	CLC	清进位	进位标志清零(即 CY 为 OFF)。	218
45	TRSM	跟踪存贮器采样	跟踪数据定位。	288
(@)46	MSG	信息显示	在编程器的显示器上显示 16 个字符的信息。	289
47～48	用于扩展指令			
(@)50	ADB	二进制加法	两个 4 位十六进制数和 CY 的值相加，结果输出到指定字中。	233
(@)51	SBB	二进制减法	从一个 4 位十六制数减去另一个 4 位十六进制数和 CY 的值，结果输出到指定字中。	235
(@)52	MLB	二进制乘法	两个 4 位十六进制数相乘，结果输出到指定字中。	237
(@)53	DVB	二进制除法	两个 4 位十六进制数相除，结果输出到指定字中。	238
(@)54	ADDL	双字 BCD 码加法	两个 8 位的 BCD 码数(每个数有 2 个字)相加且加上 CY 的值，结果输出到指定字中。	219
(@)55	SUBL	双字 BCD 码减法	从另一个 8 位的 BCD 码数中减去一个 8 位的 BCD 码数和 CY 的值，结果输出到指定字中。	223
(@)56	MULL	双字 BCD 码乘法	两个 8 位的 BCD 码相乘，结果输出到指定字中。	226
(@)57	DIVL	双字 BCD 码除法	两个 8 位的 BCD 码相除，结果输出到指定字中。	227

（续表）

代 码	助 记 器	名 称	功 能	页 数
(@)58	BINL	双字 BCD 码→双字二进制	将两个连续源字中的 BCD 码数转换成二进制数且将转换好的数输出到两个连续结果字中。	196
(@)59	BCDL	双字二进制→双字 BCD 码	将两个连续源字中的二进制换成 BCD 码数，且将转换好的数输出到两个连续结果字中。	197
60～69	用于扩展指令			
(@)70	XFER	块传送	将连续几个源字的内容传送到连续的目标字中。	174
(@)71	BSET	块设定	将一个字或常数的内容复制到几个连续字中。	172
(@)72	ROOT	平方根	计算 8 位 BCD 码数的平方根，将四台五入后的 4 位整数结果输出到指定字中。	231
(@)73	XCHG	数据交换	两个不同字的内容互相交换。	174
(@)74	SLD	数左移	将起始字和结果字之间的数据左移一个数字位(4 位)。	168
(@)75	SRD	数右移	将起始字和结果字之间的数据右移一个数字位(4 位)。	169
(@)76	MLPX	4－16/8－256 译码	将源字中的 4 位十六进制数转换成 0～15 的十进制值，在结果中将转换后的值所对应的位置“1”，也可转换 8 位的十六进制数且将结果字 R～R＋15 的对应位置“1”。	199
(@)77	DMPX	16－4/256－8 编码	确定源字中最高接通位(置“1”位)的位置且在 R 中将接通位的位置(0～F)写成数字，也可在一组或二组(16 个字)(S～S＋15，S＋16～S＋31)中确定最高接通位的位置且将接通位的位置(00～FF)以字节写入 R。	202
(@)78	SDEC	七段译码	将源字中 16 进制数转换成七段显示码数据。	205
(@)79	FDIV	浮点数除法	一个浮点数(Dd＋1，Dd)和另一个浮点数(Dr＋1，Dr)相除，结果输出到 R＋1 和 R。	228
(@)80	DIST	单字数据分配	把源数据中的一个字传送到目标字中，目标字的地址由目标基字的地址加上偏移量得到。	175
(@)81	COLL	数据调用	从源字中抽出数据，将它写到目标字中。	176
(@)82	MOVB	位传送	把源字或常量中指定的位传送到目标字的指定位中。	179
(@)83	MOVD	数传送	把指定源数中 4 位 16 进制内容传送到指定的目标数字中，最多 4 个数。	179
(@)84	SFTR	可逆移位寄存器	在指定的字或字序列中将数据向左或向右移动。	165
(@)85	TCMP	表格比较	把 4 个数字的 16 进制值和 16 个字组成的表格里的值进行比较。	191
(@)86	ASC	ASCII 码转换	把源字中的 16 进制数转换成 8 位 ASCII 码，放在起始目标字的最左或最右字节。	206
87～89	用于扩展指令			
(@)90	SEND	网络发送	用于通过 SYSMACNET 链接系统或 SYSMACLINK 系统同其它 PC 通信。	303
(@)91	SBS	子程序调用	调用和执行子程序 N。	271
92	SBN	子程序定义	设置子程序 N 的起始标志。	273
93	RET	子程序返回	设置子程序 N 的结束标志，把控制权返回给主程序。	273
(@)94	WDT	监视定时器刷新	增加监视定时器当前值(PV)，从 0～6300 ms。	292
(@)97	IORF	I/O 刷新	刷新起始字和结束字之间的所有 I/O 字。	292
(@)98	RECV	网络接收	用于通过 SYSMACNET 链接系统或 SYSMACLINK 系统同其它 PC 的通讯。	305
(@)99	MCRO	宏	调用和执行一个可替换 I/O 字的子程序。	273

扩展指令

下表列出的指令可当作为扩展指令使用，对有缺省值代码的指令给出了缺省的功能代码。

代 码	助 记 器	名 称	功 能	页 数
17	(@)ASFT	异步移位寄存器	产生一个移位寄存器，当相邻两个字中一个为零，另一个不为零时，互相交换彼此的内容。	170
18	(@)SCAN	循环时间	设置最小的扫描周期(0～999.0ms)。	287
19	(@)MCMP	多字比较	把一16个连续字块和另一16个连续字块进行比较。	183
47	(@)LMSG	长信息	把32位字符信息输出到编程器。	290
48	(@)TERM	终端方式	将编程器转换到终端方式，以执行标准键盘映象的操作。	291
60	CMPL	双字比较	比较两个8位的十六进制数。	187
61	(@)MPRF	组－2高密度I/O刷新	刷新分配给组－2高密度I/O单元的I/O字。	293
62	(@)XFRB	多位传送	将多达255个指定源位中的状态复制到指定的目标位中。	180
63	(@)LINE	列→行转换	将从16个连续字中得到的一个位列复制到一个指定字中。	213
64	(@)COLM	行→列转换	将从指定字中得到的16位数复制到16个连续字的一个位列中。	214
65	(@)SEC	小时→秒	把小时和分转换成秒	198
66	(@)HMS	秒→小时	把秒转换成小时和分	199
67	(@)BCNT	位计数器	在指定的字块中对所有状态为ON的位计数。	293
68	(@)BCMP	块比较	判断一个字的值是否在16个范围内(定义了其下限和上限。)	189
69	(@)APR	数学处理	完成正弦、余弦或线性近似的计算。	253
87	TTIM	累加定时器	产生一个累加定时器	157
88	(@)ZCP	区域比较	将一个字和已定义的一个上下限范围进行比较，比较后的结果输出到GR、EQ和LE标志位。	191
89	(@)INT	中断控制	完成中断控制，屏蔽或不屏蔽I/O中断的中断位。	275
…	7SEG	七段码显示输出	把4位或8位的BCD码转换成7段码的显示格式，然后输出转换好的数据。	317
…	(@)ADBL	双字二进制加法	两个8位的二进制数相加(不带符号或带符号数)，且将结果输出到R和R＋1中。	239
…	AVG	求平均值	把指定数目的十六进制数相加，计算平均值，十进制小数点后保留4位。	249
…	(@)BXF2	EM组传送	将几个连续源字中内容传送到连续目标字中。当前EM组中的字可指定为源字或目标字。	182
…	CPS	带符号二进制比较	比较两个16位(4个数)带符号二进制数，结果输出到GR、EQ和LE标志位。	193
…	CPSL	带符号双字二进制比较	比较两个32位(8个数)带符号二进制数，结果输出到GR、EQ和LE标志位。	194
…	(@)DBS	带符号二进制数除法	两个16位带符号二进制数相除，将32位带符号二进制数结果输出到R＋1和R。	245
…	(@)DBSL	带符号双字二进制数除法	两个32位带符号二进制数相除，将64位带符号二进制数结果输出到R＋3和R。	246

（续表）

代 码	助 记 符	名 称	功 能	页 数
…	DSW	数字开关输入	从一个数字开关中输入4位或8位BCD码。	319
…	(@)EMBC	选择EM组	将当前EM组转换到指定的组号。	302
…	(@)FCS	帧校验	对通过一个上位机链接命令传送的数据检查错误。	294
…	FPD	故障点检测	在一个指令块中找错误。	296
…	(@)HEX	ASCII码→十六进制	把ASCII码数据转换成十六进制数。	209
…	(@)IEMS	EM间接寻址	将DM间接寻址的目标(＊DM)转换到DM或指定的EM组。该指令可用于变更当前组号。	301
…	(@)IORD	特殊I/O单元读	将指定特殊I/O单元存储器中数据传送到PC的字中。	330
…	(@)IOWR	特殊I/O单元写	将PC字中数据传送到指定特殊I/O单元的存储器中。	331
…	(@)MAX	寻找最大值	在指定数据区中找最大值，且将它输出到另一个字中。	247
…	(@)MBS	带符号二进制乘法	两个带符号二进制数相乘，将8位带符号二进制结果输出到R+1和R。	243
…	(@)MBSL	带符号双字二进制数乘法	两个32位(8个数)带符号二进制数相乘，将16位数带符号二进制结果输出至R+3～R	244
…	(@)MIN	寻找最小值	在指定数据区中找最小值，且将它输出到另一个字中。	248
…	MTR	矩阵输入	从一个8×8点的矩阵中输入数据，且将数据记录于D～D+3。	327
…	(@)NEG	二进制求补	把源字中一个4位十六进制数转换成相应的二进制补码，且将结果输出到R。	215
…	(@)NEGL	双字二进制求补	把源字中8位十六进制数转换成相应的二进制补码，且将结果输出到R和R+1。	216
…	(@)PID	PID控制	根据操作数和预置的PID参数完成PID控制。	256
…	(@)PMCR	通信协议宏	调用和执行在通信板中登记过的指定通信序列。(通信协议)	315
…	(@)RXD	接收	从一个通信端口接收数据。	310
…	(@)SBBL	双字二进制减法	两个8位的二进制数(不带符号或带符号数)相减，结果输出到R和R+1。	241
…	(@)SCL	线性比例	完成已计算好数据的线性变换。	211
…	(@)SRCH	数据搜索	在内存指定区域里寻找指定的数据，且将在此区域里包含这个数据的字地址输出。	299
…	(@)STUP	修改RS-232C设置	修改PC设置中指定端口的设定值。	314
…	(@)SUM	求和	在指定的内存区域中求字内容的和。	251
…	(@)TKY	十进制键输入	从10个键的键盘输入8位BCD码数。	325
…	(@)TXD	发送	通过一个通信端口发送数据。	311
…	(@)XDMR	扩展DM读	读指定扩展DM数据区中若干指定字的内容，且输出到PC的目标字中。	300
…	(@)XFR2	EM块传送	将几个连续源字中内容传送到连续目标字中。任何有效EM组内的字可指定为源字或目标字。该指令可用于变更当前组号。	182
…	ZCPL	双字区域比较	把一个8位(数字位)数和已定义了上下限的范围进行比较，结果输出到GR，EQ和LE标志位。	193

附　录C

出错和算术标志操作

下面表格列出了影响N, OF, UF, ER, CY, GR, LE和EQ标志的指令。

一般, N表示负的结果, OF表示16位计算结果大于32, 767(7FFF)或32位计算结果大于2, 147, 483, 647(7FFFFFFF), UF表示16位计算结果小于－32, 768(8000)或32位计算结果小于－2, 147, 483, 648(80000000), 详见第五章"指令集"。

ER表示操作数不在要求的范围内。CY表示运算或数据移位结果。GR表示比较值大于某个标准值, LE表示小于和EQ表示等于某个标准值。EQ还可表示算术运算的零结果。详见第五章"指令集"。

表格中垂直箭头表示标志位根据指令执行结果置ON和OFF。

当ER为ON时, 尽管梯形图指令TIM和CNT仍执行, 但在ER栏底下有垂直箭头的其它指令不被执行。当ER为ON时, 下表中所有其它标志也不操作。

在表格中未列出的指令不影响任何标志位。尽管表中只列出指令的非微分形式, 指令的微分形式以相同方式影响标志位。

执行END(01)指令时, 8个标志都为OFF, 因此它们的状态不能通过编程器监视。

指令	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)	页数
TIM	↕	…	…	…	…	…	…	…	152
CNT	↕	…	…	…	…	…	…	…	158
END(01)	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	151
SFT(10)	OFF	…	…	…	…	…	…	…	162
CNTR(12)	↕	…	…	…	…	…	…	…	161
TIMH(15)	↕	…	…	…	…	…	…	…	156
WSFT(16)	↕	…	…	…	…	…	…	…	169
CMP(20)	↕	…	↕	↕	↕	…	…	…	185
MOV(21)	↕	…	…	↕	…	…	…	↕	171
MVN(22)	↕	…	…	↕	…	…	…	↕	172
BIN(23)	↕	…	…	↕	…	…	…	OFF	195
BCD(24)	↕	…	…	↕	…	…	…	…	196
ASL(25)	↕	↕	…	↕	…	…	…	↕	166
ASR(26)	↕	↕	…	↕	…	…	…	OFF	167
ROL(27)	↕	↕	…	↕	…	…	…	↕	167
ROR(28)	↕	↕	…	↕	…	…	…	↕	168
COM(29)	↕	…	…	↕	…	…	…	↕	264
ADD(30)	↕	↕	…	↕	…	…	…	…	218
SUB(31)	↕	↕	…	↕	…	…	…	…	221
MUL(32)	↕	…	…	↕	…	…	…	…	225
DIV(33)	↕	…	…	↕	…	…	…	…	226
ANDW(34)	↕	…	…	↕	…	…	…	↕	265
ORW(35)	↕	…	…	↕	…	…	…	↕	266
XORW(36)	↕	…	…	↕	…	…	…	↕	267
XNRW(37)	↕	…	…	↕	…	…	…	↕	267
INC(38)	↕	…	…	↕	…	…	…	…	217
DEC(39)	↕	…	…	↕	…	…	…	…	218
STC(40)	…	ON	…	…	…	…	…	…	218

（续表）

指令	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)	页数
CLC(41)	…	…	…	…	…	…	…	…	218
MSG(46)	↕	…	…	…	…	…	…	…	289
ADB(50)	↕	↕	…	↕	…	↕	↕	↕	233
SBB(51)	↕	↕	…	↕	…	↕	↕	↕	235
MLB(52)	↕	…	…	↕	…	…	…	↕	237
DVB(53)	↕	…	…	↕	…	…	…	↕	238
ADDL(54)	↕	↕	…	↕	…	…	…	…	219
SUBL(55)	↕	↕	…	↕	…	…	…	…	223
MULL(56)	↕	…	…	↕	…	…	…	…	226
DIVL(57)	↕	…	…	↕	…	…	…	…	227
BINL(58)	↕	…	…	↕	…	…	…	OFF	196
BCDL(59)	↕	…	…	↕	…	…	OFF	…	197
XFER(70)	↕	…	…	…	…	…	…	…	174
BSET(71)	↕	…	…	…	…	…	…	…	172
ROOT(72)	↕	…	…	↕	…	…	…	…	231
XCHG(73)	↕	…	…	…	…	…	…	…	174
SLD(74)	↕	…	…	…	…	…	…	…	168
SRD(75)	↕	…	…	…	…	…	…	…	169
MLPX(76)	↕	…	…	…	…	…	…	…	199
DMPX(77)	↕	…	…	…	…	…	…	…	202
SDEC(78)	↕	…	…	…	…	…	…	…	205
FDIV(79)	↕	…	…	…	↕	…	…	…	228
DIST(80)	↕	…	…	…	↕	…	…	↕	175
COLL(81)	↕	…	…	…	↕	…	…	↕	176
MOVB(82)	↕	…	…	…	…	…	…	…	179
MOVD(83)	↕	…	…	…	…	…	…	…	179
SFTR(84)	↕	↕	…	…	…	…	…	…	165
TCMP(85)	↕	…	…	↕	…	…	…	…	190
ASC(86)	↕	…	…	…	…	…	…	…	207
SEND(90)	↕	…	…	…	…	…	…	…	303
SBS(91)	↕	…	…	…	…	…	…	…	271
SBN(92)	↕	↕	↕	↕	↕	↕	↕	↕	273
WDT(94)	↕	…	…	…	…	…	…	…	292
IORF(97)	↕	…	…	…	…	…	…	…	292
RECV(98)	↕	…	…	…	…	…	…	…	305
MCRO(99)	↕	…	…	…	…	…	…	…	273

扩展指令

对具有缺省功能代码的指令标出了该功能码。

指令	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)	页数
7SEG(—)	↕	…	…	…	…	…	…	…	317
ADBL(—)	↕	↕	…	↕	…	↕	↕	↕	239
APR(69)	↕	…	…	↕	…	…	…	↕	253
ASFT(17)	↕	…	…	…	…	…	…	…	170
AVG(—)	↕	…	…	…	…	…	…	…	249
BCMP(68)	↕	…	…	…	…	…	…	…	189
BCNT(67)	↕	…	…	↕	…	…	…	…	293

（续表）

指令	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)	页数
BXF2(—)	↕	…	…	…	…	…	…	…	182
CMPL(60)	↕	…	↕	↕	↕	…	…	…	187
COLM(64)	↕	…	…	↕	…	…	…	…	214
CPS(—)	↕	…	↕	↕	↕	…	…	…	193
CPSL(—)	↕	…	↕	↕	↕	…	…	…	194
DBS(—)	↕	…	…	↕	…	…	…	↕	245
DBSL(—)	↕	…	…	↕	…	…	…	↕	246
DSW(—)	↕	…	…	…	…	…	…	…	319
EMBC(—)	↕	…	…	…	…	…	…	…	302
FCS(—)	↕	…	…	…	…	…	…	…	294
FPD(—)	↕	↕	…	…	…	…	…	…	296
HEX(—)	↕	…	…	…	…	…	…	…	209
HKY(—)	↕	…	…	…	…	…	…	…	323
HMS(66)	↕	…	…	↕	…	…	…	…	199
IEMS(—)	↕	…	…	…	…	…	…	…	301
INT(89)	↕	…	…	…	…	…	…	…	275
IORD(—)	↕	…	…	↕	…	…	…	…	330
IOWR(—)	↕	…	…	↕	…	…	…	…	331
LINE(63)	↕	…	…	↕	…	…	…	…	213
LMSG(47)	↕	…	…	…	…	…	…	…	290
MAX(—)	↕	…	…	↕	…	…	…	↕	247
MBS(—)	↕	…	…	↕	…	…	…	↕	243
MBSL(—)	↕	…	…	↕	…	…	…	↕	244
MCMP(19)	↕	…	…	↕	…	…	…	…	183
MIN(—)	↕	…	…	↕	…	…	…	↕	248
MPRF(61)	↕	…	…	…	…	…	…	…	293
MTR(—)	↕	…	…	…	…	…	…	…	327
NEG(—)	↕	…	…	↕	…	…	↕	↕	215
NEGL(—)	↕	…	…	↕	…	…	↕	↕	216
PID(—)	↕	↕	…	…	…	…	…	…	256
PMCR(—)	↕	…	…	…	…	…	…	…	315
RXD(—)	↕	…	…	…	…	…	…	…	310
SBBL(—)	↕	↕	…	↕	…	↕	↕	↕	241
SCAN(18)	↕	…	…	…	…	…	…	…	287
SCL(—)	↕	…	…	↕	…	…	…	…	210
SEC(65)	↕	…	…	↕	…	…	…	…	198
SRCH(—)	↕	…	…	↕	…	…	…	…	299
STUP(—)	↕	…	…	…	…	…	…	…	314
SUM(—)	↕	…	…	↕	…	…	…	↕	251
TERM(48)	↕	…	…	…	…	…	…	…	291
TKY(—)	↕	…	…	…	…	…	…	…	325
TTIM(87)	↕	…	…	…	…	…	…	…	157
TXD(—)	↕	…	…	…	…	…	…	…	310
XDMR(—)	↕	…	…	…	…	…	…	…	300
XFR2(—)	↕	…	…	…	…	…	…	…	182
XFRB(62)	↕	…	…		…	…	…	…	180
ZCP(88)	↕	…	↕	↕	↕	…	…	…	191
ZCPL(—)	↕	…	↕	↕	↕	…	…	…	193

附　录 D

字分配记录表

本附录所列表格可由程序员复制，用来记录 I/O 位的分配和终配设置以及工作位、数据存储区、定时器及计数器的详细说明。

I/O 位

编程员： 程序： 日期 页

字：	单元：	
位	现 场 设 备	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

字：	单元：	
位	现 场 设 备	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

字：	单元：	
位	现 场 设 备	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

字：	单元：	
位	现 场 设 备	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

工 作 位

编程员： 程序： 日期 页

区：	字：	
位	作 用	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

位：	字：	
位	作 用	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

区：	字：	
位	作 用	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

区：	字：	
位	作 用	注 释
00		
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		

数 据 存 储

编程员： 程序： 日期 页

字	内 容	注 释

定时器及计数器

编程员： 程序： 日期 页

定时器或计数器地址	定时器或计数器	设置值	注 释

附　录 E

程序代码表

可复制下页用来登记梯形图程序编程。表的设计非常灵活，允许用户输入所要求的地址及指令。

编程程序代码时，一定要指定指令的功能代码和操作数的数据区(或常数)。当使用编程器或其他外部设备输入程序时，这些代码是非常必须的。

程序代码表

编程员： 程序： 日期 页

地址	指　令	操作数

附　　录 F

数据转换表

不带符号数据

十进制码	二——十进制码	十六进制码	二进制码
00	00000000	00	00000000
01	00000001	01	00000001
02	00000010	02	00000010
03	00000011	03	00000011
04	00000100	04	00000100
05	00000101	05	00000101
06	00000110	06	00000110
07	00000111	07	00000111
08	00001000	08	00001000
09	00001001	09	00001001
10	00010000	0A	00001010
11	00010001	0B	00001011
12	00010010	0C	00001100
13	00010011	0D	00001101
14	00010100	0E	00001110
15	00010101	0F	00001111
16	00010110	10	00010000
17	00010111	11	00010001
18	00011000	12	00010010
19	00011001	13	00010011
20	00100000	14	00010100
21	00100001	15	00010101
22	00100010	16	00010110
23	00100011	17	00010111
24	00100100	18	00011000
25	00100101	19	00011001
26	00100110	1A	00011010
27	00100111	1B	00011011
28	00101000	1C	00011100
29	00101001	1D	00011101
30	00110000	1E	00011110
31	00110001	1F	00011111
32	00110010	20	00100000

带符号二进制数

十进制数	十六位十六进制数	三十二位十六进制数
2147483647	—	7FFFFFFF
2147483646	—	7FFFFFFE
.	.	.
.	.	.
.	.	.
32768	—	00008000
32767	7FFF	00007FFF
32766	7FFE	00007FFE
.	.	.
.	.	.
.	.	.
5	0005	00000005
4	0004	00000004
3	0003	00000003
2	0002	00000002
1	0001	00000001
0	0000	00000000
−1	FFFF	FFFFFFFF
−2	FFFE	FFFFFFFE
−3	FFFD	FFFFFFFD
−4	FFFC	FFFFFFFC
−5	FFFB	FFFFFFFB
.	.	.
.	.	.
.	.	.
−32767	8001	FFFF8001
−32768	8000	FFFF8000
−32769	—	FFFF7FFF
.	.	.
.	.	.
.	.	.
−2147483647	—	80000001
−2147483648	—	80000000

附　录 G
扩展的 ASCII 码

编程器显示

位 0～3		位 4～7													
二进制		0000	0001	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
	十六进制	0	1	2	3	4	5	6	7	A	B	C	D	E	F
0000	0	NUL	DLE	Space	0	∂	P	⋱	p		0	∂	P	⋱	p
0001	1	SOH	DC_1	!	1	A	Q	a	q	!	1	A	Q	a	q
0010	2	STX	DC_2	"	2	B	R	b	r	"	2	B	R	b	r
0011	3	ETX	DE_3	#	3	C	S	c	s	#	3	C	S	c	s
0100	4	EOT	DC_4	$	4	D	T	d	t	$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	¥	l	|	,	<	L	¥	l	|
1101	D	CR	GS	-	=	M	]	m	}	-	=	M	]	m	}
1110	E	S0	RS	·	>	N	⋰	n	←	·	>	N	⋰	n	
1111	F	S1	US	/	?	O	-	o	→	/	?	O	-	o	→

术　语

地址(address)

在存储区中,存放数据的存储单元。在数据区中,地址是由表示数据区号的两个字母和表示字(或位)位置的一个数字组成。在 UM(程序)区,地址指明了指令的位置。对 FM 区,地址指明了块的位置,等。

存贮分配(allocation)

PC 存贮器中的某些位和字赋于不同功能的过程,包括将各 I/O 位与器件上 I/O 点相对应。

与(AND)

一种逻辑操作。只有当二者都为"真"时其结果才为真,在梯形图编程中,位的状态通常约定为 ON 或 OFF,或者是这些位状态的逻辑组合,即称之为执行条件。

全塑光缆(APF)

"全塑光缆"英文"all plastic fiber－optic cable"的首字母缩写。

辅助继电器区(AR area)

PC 中分配给标志位、控制位及工作位的数据区。

算术移位(arithmetic shift)

进位位也一起移位的移位操作。

美国信息交换标准码(ASCⅡ)

"美国信息交换标准码"英文"American Standard Code for Information Inerchange"的首字字母缩写,对字符进行 ASCII 编码,可用来输出到打印机及其他外部设备。

ASCⅡ单元(ASCⅡ Unit)

用 BASIC 语言编程的一种智能 I/O 单元。在网络链接系统中将此单元接到网络服务单元(NSU)上,就可将命令传送到其他节点上去。

底板(Backplane)

可在上面安装单元形成机架的底板。底板上有安装各种单元的槽,各单元沿着走线连到 CPU 上。底板上还有与其他底板相连接的接插口。有些系统中,不同的底板用于不同的机架,在另一些系统中,机架仅按照安装的单元不同来区分。

二－十进制码(BCD)

"十—十进制码"英文为"binary－coded decimal"的首字母缩写。

十－十进制计算(BCD calculation)

用十—十进制码进行算术运算。

二进制(binary)

所有的数以 2 为基数表示的数制,即任何一个数可用反码及补码来表示。每个四位二进制数相当于一个十六进制数。

二进制计算(binary calculation)

用二进制码进行算术运算。

十－十进制码(binary－coded decimal)

一种表示数的方法,用四位二进制数表示一位十进制数。

位(bit)

一个二进制位。在二进制记数法中每一个数据单位称为"位"(bit),是 PC 中用电信号存储信息的最小单位。"位"的状态不是 ON 就是 OFF。在特定地址中的不同位,有特殊的用途。例如由外部设备输入的一种保持状态。而编程时,"位"一般可用来组成程序及数。

位地址(bit address)

存储器中存储数据一个位所占的存储单元。位地址必须指定数据区、字及位号,有时也可缺省。

位指定器(bit designator)

一个操作数,用来指明指令使用的字中某一位或某几位。

位编号(bit number)

指示一个字中各位的次序。"00"为最右侧(最低)位,"15"为最左侧(最高)位。

模块化 PC(building - block PC)

由独立部件或"模块"构成 PC。对一台由模块构成的 PC 来说。没有一个单元可单独称之为 PC,因此,更确切地说,PC 是一个功能部件的组合。

母线(bus bar)

在梯形图左侧(有时右侧)画的一根垂直线,指令沿这根母线逐条向下执行,母线又总是所有指令行的起点。

调用(call)

从执行主程序转到子程序的处理过程。一个子程序可被一条指令或一个中断所调用。

进位标志(carry flag)

该标志用于加法、乘法操作中的进位,或者在减法操作中表示其结果为负数。进位标志也能用于某些移位操作中。

时钟脉冲(clock pulse)

利用存储器中的某位产生计时操作的脉冲,各种时钟脉冲可有不同的脉冲宽度。

时钟脉冲位(clock pulse bit)

内存储器中能够产生计时操作脉冲的位。不同的时钟脉冲位可产生不同的脉冲宽度,因而形成不同的频率。

公用数据(common data)

存储在 PC 的 LR(链接)区并且可被同一系统中的其他 PC 共享的数据。每台 PC 都有确定的 LR 区段。而且每台 PC 的 LR 区的内存分配是相同的。

通信板(Communication Board)

插在 C200HX/HG/HE CPU 的任意槽内的板。CPU 使用通信板,可以通过 RS - 232C, RS - 422 或 RS - 485 与 SYSMACLINK 单元, SYSMACNET 链接单元,可编程终端,温度控制器,个人计算机,条形码读入器或其他外围设备通信。

条件(condition)

指令行中,执行终端指令(在右侧)的信息,每一条件都在内存中分配一个决定其状态的位,反过来,位的状态也决定了梯形图右侧的终端指令的执行条件。

常数(constant)

由用户指定实际数值的操作数,它存储在数据存储器特定地址中。

控制位(control bit)

为特定目的通过程序或编程器在存储器中设定的状态位。如重新启动位,可开启或关闭一个单元。

控制系统(Control System)

用来控制其他设备的所有硬件和软件组合的系统,一个控制系统可以包括 PC 系统,PC 程序及所有的 I/O 设备。I/O 设备可用来控制被控系统或从被控系统中得到反馈信息。

被控系统(controlled system)

受 PC 控制的设备。

控制信号(control signal)

PC 发出的对被控系统进行操作的信号。

计数器(counter)

存储器中一专用的数组或字组,用来计算一个过程发生的次数,或存储器中一个通过计数位(TC)存取单元,用来计算位状态次数或从 OFF 到 ON 时执行条件变化次数。

中央处理单元(CPU)

"中央处理单元"英文"Central Processing Unit"的首字母缩写。在 PC 系统中,CPU 能执行程序,处理 I/O 信号及与外部设备通信等。

CPU 底板(CPU Backplane)

用来构成 CPU 机架的底板。

CPU 机架(CPU Rack)

是模块化 PC 的组成部分。它包括 CPU、电源及其他单元,对大多数 PC 来说,它是一个单独的机架,上面有供安插其他单元的链接槽。

清除－发送(CTS)

"清除－发送"英文"Clear－to－send"的首字母缩写。用于电子设备相互间通讯的信号,表示接受设备已准备好接收发来的数据。

周期(cycle)

执行一个梯形图程序的过程。程序从开始到结束顺序执行,每条指令都根据相应的条件依次执行。

扫描周期(cycle time)

执行梯形图一个扫描所化的时间。

数据区(data area)

PC 存储器中用来存储各类数据的区域,如在链接区(LR)中存储的是 PC 链接系统中的公用数据。存放程序的内存区不作为数据区。

数据区边界(data area boundary)

数据区的最大地址。当一个操作数需要多个字时,要确保最大地址不能超过数据区界面地址。

数据共享(data sharing)

在 PC 链接系统及网络链接系统的数据链接中二个或多个 PC 之间建立公用数据区或公用数据字的特性。

排除错误(debug)

纠正设计程序中的错误,达到预期的过程操作。排除错误包括纠正语法错误,调整时序及控制操作间的协调。

十进制(decimal)

以 10 为基数表示的数制。在 PC 中,所有的十进制数最终都以二进制形式存储。常用四个二进制位表示一个十进制数,即二—十进制编码(BCD)。

减量(decrement)

减少一个数值。

缺省(default)

用户没有设置而由 PC 自动设置的值。许多设备通电时,都设定了缺省条件。

定义符(definer)

指令操作数的一个数,用来定义指令本身,并非指令所操作的数。定义符包括转移号数,子程序号等。

延迟(delay)

在跟踪中,定义一个延时值,即对应激发延迟该定义时间再开始跟踪。延迟可为正延迟或负延迟,即可表明触发器条件向任一方向偏移。

目标地址(destination)

由指令设置正在进行操作的数据的存储单元。与指令中用来取数据的存储单元相反,从中取数的存储单元,称为源地址。

微分型指令(differentiated Instruction)

只有当执行条件从 OFF 到 ON 时才执行一次的指令。非微分型指令在执行条件为 ON 每个周期都执行。

微分指令(differentiation Instruction)

保证操作位在大于一个周期的时间内不为 ON 的指令。当执行条件从 OFF 到 ON 执行一个上升沿微分指令或当执行条件从 ON 到 OFF 执行一个下降沿微分指令。

数(digit)

在存储器中用四个二进位组成的一个存储单元。

数指定器(digit designator)

一个操作数,用来指定指令使用的字中某位数或某几位数。

分布控制(distributed control)

一个自动化的概念,它将部分控制系统置于被控设备处进行控制,即对整个系统而言,控制被"分散"(分布式)。分布控制是 PC 系统中的一个基本概念。

数据存贮器区域(DM area)

在内存储器中只存储数据字的数据区。该区中的字,不能按位存取,而是按字存取。

下载(download)

将程序或数据从上位计算机传到下位计算机或 PC 的处理过程。

电噪声(electrical noise)

使一个或多个电气特性,如电压、电流或数据发生随机变化,可能影响设备的正常操作。

EM 区(EM area)

扩展数据存贮区,同 DM 一样,EM 区中字只能按字存取,当 PC 的电源断开时 EM 区的数据将保留。

错误代码(error code)

表示有错误存在的一种数码,有些代码还能指出错误的性质。有些出错代码是由系统产生,另一些出错代码可由程序员在程序中定义。

异或(exclusive OR)

一种逻辑操作,当一个而且只有一个条件为"真"时,其结果为"真"。在梯形图编程中常用 ON 或 OFF 状态位来表示,或用这些状态的逻辑组合表示,称为执行条件。

异或非(exclusive NOR)

一种逻辑操作。二者都为"真",或二者都为"假"时,其结果为"真"。在梯形图编程中,用 ON 或 OFF 状态位来表示,或用这些状态的逻辑组合来表示,称为执行条件。

执行条件(exection condition)

使指令执行的 ON 或 OFF 状态。执行条件取决于该指令行中直止执行指令前所有条件的逻辑组合。

执行时间(execution time)

CPU 执行一条指令或执行整个程序所化的时间。

扩展 I/O 底板(Expansion I/O Backplane)

用于构成扩展 I/O 机架的底板。

扩展 I/O 机架(Expansion I/O Back)

是模块化 PC 的一部分。一个扩展 I/O 机架可与 CPU 机架或与另一个扩展 I/O 机架相连,增加的槽数可以配置更多的单元。

扩展计数器(extended counter)

在程序中，用二条或多条连续的计数指令构成的计数器。如此构成的计数器要比单个指令的标准计数器具有更大计数范围。

扩展定时器(extended timer)

在程序中用二条或多条连续的计时指令构成的定时器。它的计时时间要比单个指令的标准定时器计时的时间要长。

工厂智能终端(Factory Intelligent Terminal)

能提供先进的编程方法与调试功能，方便 PC 操作的一种编程设备。工厂智能终端还可提供连接各种外部设备的接口，如软盘驱动器等。

严重出错(fatal error)

能使 PC 终止工作的一种出错。只有在排除这个出错以后才能再行操作。

工厂智能终端(FIT)

"工厂智能终端"英文"Factory Intelligent Terminal"的首字母缩写。

标志(flag)

在存储器中为指出一些操作状态类别而由系统设置的专门位。有些标志，如进位标志，可由操作者设置，也可通过程序设置。

闪烁位(flicker bit)

由程序控制的按一定频率通、断的位。

十进制浮点数(floating point decimal)

一种十进制数，表示为 0～1(尾数)间的数值乘以 10 的次方，例 0.538×10^{-5}。

软盘接口单元(Floppy Disk Interface Unit)

用于将软盘连接到 PC 上去的一种连接单元。以便将程序和/或数据存储到软盘上去。

强制复位(force reset)

用编程器强行将一个位置 OFF 的过程。程序执行结束，该位通常为 OFF。

强制置位(force set)

用编程器强行将一个位置 ON 的过程。程序执行结束，该位通常为 ON。

功能代码(function code)

用来将指令输入到 PC 中去的一个二位数字。

硬件出错(hardware error)

出于 PC 硬件结构(电子器件)出错。相反，"软件出错"出自软件(即程序)。

十六进制 (hexadecimal)

以 16 为基数的数字表示数制。在 PC 中所有数据最终是以二进制形式存储，然而为简化操作，编程器显示及输入，常用 16 进制表示。每四个二进制位组相当于一个十六进制数。

上位机链接系统(Host Link System)

由一台或多台上位计算机通过上位机链接单元连接一台或多台 PC，这样，上位机可将数据传到 PC，PC 也能将数据传到上位机。上位机链接系统能够集中管理及控制 PC 系统。

上位机链接单元(Host Link Unit)

在上位机链接系统中，用于将 PC 连到上位机的接口单元。

上位计算机(host computer)

在上位机链接系统中，可与 PC 相互传送程序或数据的计算机。上位计算机用于数据管理及整个系统的控制。上位计算机一般为个人计算机或商用计算机。

保持继电器区(HR area)

存储和处理数据的一个数据区，当 PC 关闭(断电)时，仍能保持这些数据。

增量(increment)

增加一个数值。

间接地址(indirect address)

是一种地址,其内容指出另一个地址,第二个地址内容可当为操作数用。间接寻址只能在 DM 区(数据存储区)中进行。

初始化出错(initialization error)

PC 启动(初始化)时,硬件或软件出错。

初始化(initialize)

是启动过程的一部分,对某些内存区清"0"、检查系统设置及设置缺省值等。

输入(input)

从外部设备将信号送到 PC 中去。该词常用来抽象地、概括地表示"信号输入"。

输入位(input bit)

输入区中配置保持输入状态的位。

输入设备(input device)

把信号送到 PC 中的外部设备。

输入点(input point)

输入信号加到 PC 上的那一点。输入点实质上是一个终端或连接件的一个引脚。

输入信号(input signal)

进入到 PC 连接点状态的变化。一般,连接点的电压由低到高、从不导通到导通,认为输入产生信号了。

指令(instruction)

程序中的一条命令,告诉 PC 实现某种动作及执行动作所用的数据。一条简单的指令,可使一位转为 ON 或 OFF。复杂的指令可转换和/或传输一大块数据。

指令块(instruction block)

梯形图程序中逻辑上相关的指令组。虽然任何逻辑上相关的指令组可称为指令块,但这名词常用于表示称为逻辑块的指令块,要求逻辑块指令与其他指令或其他逻辑块相关。

指令执行时间(instruction execution time)

执行一条指令的时间。任何一条指令执行的时间不尽相同是由于指令执行条件及使用的操作数不同所致。

指令行(instruction line)

在梯形图中处于同一水平线上的一组条件。指令行可以分离及统联形成一个指令块。

接口(interface)

是系统与设备之间概念上的界面。通常经过这接口,传输数据的表示格式要改变。接口设备(如 NSB)能完成码转换、变换格式或改变数据传输速度的操作。

联锁(interlock)

对指令成组处理的一种编程方法,当组内指令不需单独执行时,可对整组指令一起复位。一个联锁程序段通常在 ON 条件下执行,在 OFF 执行条件下可部分复位。

中断(信号)(interrupt(signal))

停止执行正常程序并转去执行一个子程序的信号。

中断输入单元(interrupt Input Unit)

用来将外部中断信号输入到 PC 中去的机架安装单元。

取反条件(inverse condition)

当一位赋值 OFF 时,产生一个 ON 执行条件。当一位赋值 ON 时,产生一个 OFF 执行条件。

I/O 容量(I/O capacity)

PC 能处理的输入和输出点数。小型 PC 输入输出量约 100 点,大型 PC 约 2000 点。

I/O 控制单元(I/O Control Unit)

在某些 PC 中,安装在 CPU 机架上用来监视、控制扩展 I/O 单元上 I/O 点的单元。

I/O 设备(I/O devices)

连接 I/O 单元、特殊 I/O 单元,或智能 I/O 单元端子的设备。如 I/O 设备有控制其他设备的功能,则它可成为控制系统的一部分或成为受控系统的一部分。

I/O 接口单元(I/O interface Unit)

在某些 PC 中,安装在扩展 I/O 机架上使扩展 I/O 机架与 CPU 机架相连的单元。

I/O 链接(I/O Link)

在光纤远程 I/O 系统中,能在 PC 之间产生一个或两个直接输入/输出状态字,这些字是控制的主站 PC 和通过 I/O 链接单元或 I/O 链接机架同远程 I/O 系统相连的 PC 之间的输入/输出。

I/O 链接单元(I/O Link Unit)

在光纤远程 I/O 系统中,同构成 I/O 联网的 PC 一起使用的单元。

I/O 点(I/O point)

输入信号进入 PC 或 PC 输出信号的地方。I/O 点可能是单元上一个端子,也可能是单元连接器上的一个插脚。在编程时,I/O 点是指在 I/O 区域中相应的 I/O 位。

I/O 响应时间(I/O response time)

PC 从外部设备收到输入信号到发出输出信号所需的时间。

I/O 表(I/O table)

在 PC 内存储器中建立的一个表,列出分配给 PC 系统中各个单元的 IR 区域字。该表可由编程器来建立或修改。

I/O 单元(I/O Unit)

安装在底板上形成机架的最基本的单元。I/O 单元包括输入单元及输出单元。各单元按说明书要求使用。I/O 单元不包括特殊 I/O 单元、链接单元等。

I/O 字(I/Oword)

在 IR 区域中分配给 PC 单元的一种字。

内部继电器区(IR area)

一个数据区,其主要功能是保持进入系统的输入状态和系统发送的输出状态。在该区中这样使用的位和字称为 I/O 位及 I/O 字。在该区中其余的位是工作位。

日本工业标准(JIS)

"日本工业标准"英文"Japanese Industrial Standards"的首字母缩写。

跳转(jump)

一种编程方法,不是按顺序执行指令,而是从程序的一个点直接跳转到另一个点。跳转通常是有条件执行的。

跳转号(jump number)

定义从哪条指令跳转到哪条指令的定义符。

梯形图(程序)(ladder diagram (program))

由继电器控制系统引伸的一种编程方式,用回路表示编程指令的逻辑流程,图形像一个梯形,故称梯形图。

梯形图符号(ladder diagram symbol)

在梯形图程序中使用的符号。

梯形指令(ladder instruction)

在梯形图程序中表示梯级"横档"部分的指令,在梯形图上其他指令沿梯形图右端向下排,称终端指令。

梯形图支持软件(ladder Support Software)

是一个为 IBM－AT、IBM－XT 或兼容微机提供工厂智能终端大部分功能的软件包。

局域网(LAN)

"局域网"英文"Local area network"的首字母缩写。

最左(最高)位/字(leftmost (bit/word))

一组位,通常是一个字中编号最大的位,或一个字组中编号最大的字。这些位或字常称为"最高"的位或字。

链接适配器(Link Adapter)

用来连接通信线路的单元。可以是分支线路或不同类型电缆之间的换接。有二种适配器:分支链接适配器及转换链接适配器。

链接(link)

两个单元(设备)间的硬件、软件连接。"链接"可以表示两个单元之间物理连接(如连线远程 I/O 系统中的光缆链接)或通过到另一个存储单元中数据的软件连接(如网络数据链接)。

链接槽(linkable slot)

在 CPU 底板或扩展 I/O 底板上安装链接单元的槽。不同的底板可安装链接单元槽数是不一样的。

链接系统(Link System)

它包括以下一个或几个系统的系统:远程 I/O 系统、PC 链接系统、上位机链接系统或网络链接系统。

链接单元(Link Unit)

用来将 PC 连到链接系统上的单元。有远程 I/O 单元,I/O 链接单元、PC 链接单元、上位机链接单元及网络链接单元。

装载(load)

从外部设备或存储区将数据传到系统执行部件的处理过程,如显示缓冲寄存器。连接在 PC 上的外部设备又称负载。

局域网(local area network)

在较小范围内由若干节点组成的网络。每个节点可以是一台设备,这台设备能与另一台设备传送数据。

逻辑块(logic block)

与梯形图程序逻辑上相关的一种指令组,要求逻辑块指令能与其他指令或其他逻辑块发生联系。

逻辑块指令(logic block instruction)

是一种指令,用来逻辑组合逻辑块中得到的执行条件和当前执行条件,当前执行条件可以是单个条件或另一个逻辑块。ANDLoad 及 OR Load 就是两条逻辑块指令。

逻辑指令(logic instruction)

用来逻辑组合两个字的内容且输出逻辑结果到指定结果字的指令。逻辑指令把二个字中同样编号的位组合起来,输出结果到指定结果字的同样编号位中。

循环(loop)

按执行条件或位状态能重复执行多次的一组指令。

链接继电器区(LR area)

用于 PC 链接系统的一个数据区。该区的数据可在二台或多台 PC 间进行传输,如不使用 PC 链接系统,则该区域可作为工作区。

梯形图支持软件(LSS)

"梯形图支持软件"英文"Ladder Support Software"的首字母缩写。

主程序(main program)

除子程序以外的所有程序。

屏蔽(masking)

"覆盖"住一个中断信号使之无效,只有当这个屏蔽去掉后,此中断才起作用。

主站(Master)

是"远程 I/O 的主站"的简写。

存储区(memory area)

PC 用于存储数据或程序的存储区域。

助记符代码(mnemonic code)

梯形图程序的一种表达形式,不用梯形图而用有序的指令表来表示。用编程器将程序输入到 PC 中去,就需要用助记符代码。

监视方式(MONITOR mode)

PC 操作的一种方式,此时可执行一般的程序并允许修改存储器中的数据。用于 PC 的监视及调试。

最高位/字(most - significant (bit/word))

见最左位/字。

常闭输入(NC input)

一种常闭输入,当同输入相连的电路打开时,才认为有输入信号。

嵌套(nest)

编程时,一个循环中包含另一个循环,调用的子程序中包含了另一个子程序,或在 IF - ELSE 语句中包含了另一个 IF - ELSE 语句。

网络链接系统(Net Link Unit)

由网络链接单元连接多台 PC 组成的光纤局域网(LAN)。一般网络链接系统包括连接计算机及其他外围设备的节点。网络链接系统中的 PC 能够向里和向外传送数据、接受从其他连接计算机发来的命令并与连接的外围设备共享。

网络链接单元(Net Link Unit)

将 PC 连到网络链接系统上的单元。全名是"SYSMAC Net Link Unit"。

网络服务板(Network Service Board)

是一种设备,设有把 PC 以外的设备与网络链接系统连接的接口。

网络服务单元(Network Service Unit)

提供连接外围设备和网络链接系统的两种接口的单元。

节点(node)

计算机局域网上的一个"点",各个"点"对应一个设备,可与所有节点上的设备相互进行数据通信。每一个节点上的设备可用一个节点号码来表示。一个网络链接系统(OMRON'S 局域网)最多可包括 126 个节点。每个网络链接单元占用一个节点,这些单元装到 PC 或安装到提供计算机或其他外围设备接口的设备上。

常开输入(NO input)

一种常开输入,当同输入相连的电路接通时,才认为有输入信号。

噪声干扰(noise interference)

由电噪声引起信号的扰动。

非严重出错(nonfatal error)

只作出提示警告而不停止 PC 运行的硬件或软件出错。

正常条件(normal condition)

状态位赋值 ON 时，为 ON 执行条件；赋值 OFF 时，为 OFF 执行条件。

非(NOT)

操作数状态取反的逻辑操作，如 ANDNOT 表示对操作数实际状态取反后再进行与(AND)操作。

网络服务板(NSB)

"网络服务板"英文"Network Service Boand"的首字母缩写。

网络服务单元(NSU)

"网络服务单元"英文"Network Service Unit"的首字母缩写。

关断(OFF)

当信号不存在时的输入或输出状态。OFF 常用低电压及非导通来表示，但也可用相反的状态来定义。

关断延迟(OFF delay)

由输入设备或 PC 发出关闭信息开始，到接收设备(输出设备或 PC)可以读出其信息关闭状态之间的时间。

导通(ON)

当信号存在时的输入或输出状态。ON 常用高电压及导通来表示，但也可用相反的状态来表示。

导通延迟(ON delay)

由输入设备或 PC 发出打开信号开始，到接收设备(输出设备或 PC)可以读出其信息打开状态之间的时间。

单稳位(one - shot bit)

保持 ON 或 OFF 状态一段确定的间隔时间的位，这个时间大于一个周期。

在线拆卸(one - line removal)

当 PC 运行时，为更替单元或维修，拆卸机架上的单元。

操作数(operand)

标明作为指令数据的位和字。操作数可表示为常数值或表示为使用数据在内存分配的地址。

操作数位(operand bit)

指令操作数的位。

操作数字(operand word)

指令操作数的字。

操作出错(operating error)

与操作前初始化出错不一样，这里发生在 PC 实际运行过程中的操作出错。

光缆 I/O 单元(Optical I/O Unit)

连接在光缆远程 I/O 系统上的一种单元，它提供 8 个 I/O 点。光缆 I/O 单元不安装在机架上。

光缆从站机架(Optical Slave Rack)

通过光缆远程 I/O 从站单元连接的从站机架。

或(OR)

一种逻辑操作。如果二者之一为"真"或二者全为"真"，则结果为"真"。在梯形图程序中，常用 ON/OFF 来表示位的状态或这些状态的逻辑组合做为指令执行条件。

输出(output)

从 PC 发送到外围设备的信号。输出这个词，常抽象地概括地表示发出的信号。

输出位(output bit)

在 IR 区中设置的送到输出设备的保持状态位。

输出设备(output device)

接收从 PC 发来信号的外部设备。

输出点(output point)

PC 系统中输出信号的地方。输出点常为一个端子或连接插头一个引脚。

输出信号(output signal)

送到外部设备的信号。通常,连接点从低电压到高电压或由不导通状态转变到导通状态,作为发送了输出信号。

监视(overseeing)

由 CPU 实现的部分处理功能,包括操作 PC 所需的各种步骤。

改写(overwrite)

修改内存赋予的内容,以前的内容将不复存在。

奇偶性(parity)

调整字中或其他单元中 ON 的位数,使其总数为奇数或偶数。奇偶性常用于数据传输的校验,方法是确认传输后 ON 的位数是否仍为奇数或偶数。

可编程序控制器(PC)

"可编程序控制器"英文"Programmable Controller"的首字母编写。

印制电路板(PCB)

"印刷电路版"英文"Printed Circuit Board"的首字母编写。

PC 配置(PC configuration)

把一些单元装配和互连在一起组成一台具有一定功能的 PC。

塑料护套光缆(PCF)

"塑料护套光缆"英文"Plastic - clad optical fiber cable"的首字母缩写。

PC 链接系统(PC Link System)

通过 PC 链接单元连接多台 PC 使能共享公用数据的系统,即每台 PC 都能向链接数据区(LR 区)写一些"字",又可接受写在与这台 PC 相连的其他 PC"字"中的数据。

PC 链接单元(PC Link Unit)

在 PC 链接系统中用来链接各 PC 的链接单元。

PC 系统(PC System)

在模块化 PC 中的所有机架和与之直接连接的独立单元,但是不包括 I/O 外部设备。PC 系统的界面是 PC 和上一个层次 CPU 中的程序;下一个层次中的 I/O 单元、特殊 I/O 单元、光缆 I/O 单元,远程终端等。

外围设备(peripheral device)

有助于系统操作而连接在 PC 系统上的设备。外围设备包括打印机、编程器、外存储器等。

端口(port)

PC 或计算机上用来连接外围设备的一个连接器。

当前值(present value)

缩写为 PV。操作时,随时寄存在设备中的现行值。

印制电路板(printed circuit board)

安装在计算机或电气设备中印有电气线路的板。

打印机接口单元(Pinter Interface Unit)

连接打印机的接口单元,可打出梯形图及各种数据。

程序(program)

"告诉"可编程控器要实现控制动作次序的指令表。

可编程序控制器(Programmable Controller)

一种计算机化设备,它能根据存储器中的程序从外部设备接受输入信号,并送出输出信号。PC 可用于

对外围设备的自动控制。虽然单组件 PC 也能使用，但模块化的 PC 是由分列组件构成的。只有当提供足量的分列组件时，才能组装成功能齐全的 PC，也就是一个单独单元不能称为 PC。

程序报警(Programmed alam)

由执行程序中的指令结果而发出的警报。另外，还有系统报警。

程序出错(programmed error)

由执行程序中的指令结果而产生的错误。另外，还有系统出错。

程序信息(programmed message)

由执行程序中的指令结果而产生的信息，另外，还有系统信息。

编程器(Programming Console)

PC 中使用的一种最简单的编程设备。编程器可以是手持式的，也可以安装在 CPU 上。

编程设备(Programming Device)

是一种将程序输入到可编程控器中或者修改、监视 PC 中已有程序的外部设备。有专用编程设备，如编程器和非专用编程设备，如上位计算机。

编程方式(PROGRAM mode)

能实现输入程序和调试程序的操作方式。此时，不允许执行正常程序。

PROM 写入器(PROM Writer)

为了永久性存储和使用，将程序和其它数据写到只读存储器(ROM)中的外围设备。

提示(Prompt)

在显示屏上显示的信息或符号，提示操作人员输入数据。

当前值(PV)

“当前值”英文“Present Value”的首字母缩写。

机架(Rack)

在模块化 PC 系统中，为了形成一个功能单元而在底板上安装各种单元的装配组合。机架包括 CPU 机架、扩展 I/O 机架、I/O 机架及从站机架。

刷新(refresh)

修改发送到外围设备的输出状态过程。这样，使它与存储器中保持的输出位状态相一致，修改存储器中输入位，使它与从外部设备的输入状态位相一致。

继电器控制(relay - based control)

是 PC 早期的一种控制方式。在继电器控制中，一组继电器相互连接形成控制线路。在 PC 中它已被可编程电路所代替。

远程 I/O 主站单元(Remote I/O Master Unit)

远程 I/O 系统中的一种单元，通过它向其他远程 I/O 单元发送信号，该单元可安装在 CPU 机架上，或安装在与 CPU 机架相连的扩展 I/O 机架上。这单元常简称为主站。

远程 I/O 从站单元(Remote I/OS lave Unit)

安装在底板上以形成从站机架的一种单元，简称为从站。

远程 I/O 系统(Remote I/O Syetem)

通过安装在 CPU 机架或与它连接的 I/O 扩展机架上的主站控制远程 I/O 点的一个系统。

远程 I/O 单元(Remote I/O Unit)

远程 I/O 系统中的任何一个单元。远程 I/O 单元包括主站单元、从站单元、光缆 I/O 单元，I/O 链接单元及远程终端。

远程 I/O 字(remote I/Oword)

分配给远程 I/O 系统一个单元的 I/O 字。

复位(reset)

设置位或信号为 OFF 或者改变定时器、计数器的当前值到设定值或到"0"的过程。

返回(return)

从子程序返回到主程序(通常是调用子程序处)执行指令的过程。

可逆计数器(reversible counter)

按指定的条件可加可减的计数器。

可逆移位寄存器(reversible shift register)

可按指定的条件将数据进行两个方向移位的寄存器。

右侧指令(right - hand instruction)

是终端指令(terminal instruction)另一种说法。

最右(最低)位/字(rightmost(bit/word))

一组位(通常是一个字)中的编号最小的位或一组字中编号最小的字。这些位/字常称为最低位/字。

循环寄存器(rotate register)

一种移位寄存器,从一端移出的数据又返回到寄存器另一端。

运行方式(RUN mode)

PC 用来进行正常控制操作的操作方式。

定时中断(scheduled interrupt)

按特定时间由系统自动产生或由程序员专门设置的程序地址产生的一种中断。定时中断的结果是启动特定的子程序,可用来执行哪些必须按规定的间隔时间反复执行的指令。

自诊断(self diagnosis)

系统检验自身操作的过程,如发现有不正常现象,就会提示警告或发出出错信息。

自保持位(self - maintaining bit)

由程序保持位的状态(ON 或 OFF),除非"置位"或"复位",它是不会改变的。

服务(servicing)

PC 发出信号(数据)或从外部设备、远程 I/O 单元接受信号(数据)或为链接系统处理数据的过程。

置位(set)

设置位或信号为 ON 的过程。

设置值(set value)

减量计数器向下计数开始值,或者增量计数器向上计数最终值(最大值),或者定时器开始计时的初始值,缩写为 SV。

移位寄存器(shift register)

由一个或多个字组成的寄存器,其中的数据可按规定的位,数或字为单位,向左或向右移动规定的单位位数。在循环移位寄存器中,一端移出的数据位,返回进入另一端。在其他移位寄存器中新的数据(特定数据、"0"或"1")从一端移入,而从另一端移出的数据将丢失。

从站(Slave)

"远程 I/O 从站单元"英文"Remote I/OSlave Unit"的简写。

从站机架(Slave Rack)

包括一个远程 I/O 从站单元,并受远程 I/O 主站单元控制的机架,通常该机架和 CPU 机架分开布置。

槽(slot)

在机架(底板)上安装单元的地方。

软件出错(software error)

在软件程序中产生的错误。

软件保护(software protect)

不用硬件的开关或设定而用软件方法保护数据不改变的一种方法。

源(source)

指令用来取得数据的地址。反之,指令操作结果存放的单元地址,称为目标地址。

特殊 I/O 单元(Special I/O Unit)

为特殊用途而设计的特殊单元,包括位置控制单元,高速计数器单元,模拟量输入输出单元等。

专用数据区(SR area)

可编程控器中的一个数据区,主要用于存储标志位,控制位及有关 PC 操作的其他信息。其中只有少数位可由程序员控制,即多数位只能读,不能写。

SYSMAC 支持软件(SSS(中文版))

"SYSMAC 支持软件"英文"SYSMAC Support Software"的首字母缩写。

子程序(subroutine)

置于主程序后的一组指令,只能由主程序调用或由中断启动才能执行。

子程序号(subroutine number)

子程序调用或中断启动时用来指示子程序的定义符。

设置值(SV)

"设置值"英文"set value"的首字母缩写。

通断容量(switching capacity)

一个继电器可靠"开"和"关"时最大电压及电流。

语法出错(syntax error)

程序书写方法上的错误,语法错误包括拼词错误(即功能代码不存在)、规定操作数超出允许范围(如把 SR 保留位定义为目标地址)和指令用错(如调用不存在的子程序)。

SYSMAC 支持软件(SYSMAC Support Software)

安装在 IBM PC/AT 或兼容机上使其具有编程设备功能的软件包。

系统配置(system configuration)

系统中各单元连接布置。

系统出错(system error)

由系统产生的出错。不是执行程序中指令结果而产生的出错。

系统出错信息(system error message)

由系统产生的出错信息。不是执行程序中指令而产生的出错信息。

计数、定时数据区(TC area)

只用于定时器及计数器的数据区,该区中每一位可作为用该位定义的定时器或计数器的"设置值"(SV)、"当前值"(PV)和结束标志的存取点。

计数、定时号(TCnumber)

对应于 TC 区中一个位的定义符,将该位定义为一个定时器或计数器。

终端指令(terminal instruction)

位于梯形图右侧的指令,它使用一条指令行上最终的执行条件。

终止符(terminator)

由星号和回车组成的代码(*CR)。不论是单帧的还是多帧的数据块,*CR(回车)是一个数据块的终止符。在多帧数据块中,各帧间有间隔符分开。

定时器(timer)

存储器中由 TC 位存取的内存单元,用来从定时器设定值开始计时,并按执行条件将定时器打开及复

位。

跟踪结果区(TM area)

存储跟踪信息的存储区。

传输距离(transmission distance)

信号能被传输的距离。

暂存数据区(TR area)

存储执行条件的数据区。它能重新装入与其他指令一起使用。

跟踪(trace)

执行程序的一种操作方式。结果数据存储在跟踪区(TM 区),可用来一步步地分析及调试程序。

传输(transfer)

PC 中、或 PC 和外部设备之间,将数据从一个区转移到另一个区的过程。此时一般将数据复制到目标地址,而源地址中的数据不变。

触发地址(trigger address)

程序中定义开始跟踪点的地址。用定义触发器正延迟或负延迟来改变实际开始跟踪点。

程序存贮区(UM area)

存放现行程序的存贮区,即当前执行的程序。

单元(Unit)

在 OMRONPC 术语中大写的 Unit 是指本公司出售的产品。尽管大部份产品的名字以 Unit 为结束,但不全是这样,例如,远程终端(Remote Terminal)作为一个单元表示一器件、设备的"集合"。这样,便于上下文统一。

单元号(unit number)

当设置字或其他操作数时,为方便识别对一些链接单元及特殊 I/O 单元规定的编号。

监视定时器(watchdog timer)

系统中的一个定时器,保证扫描周期不超过规定极限值。当到达极限值时,将发出警告信号或终止 PC 操作,这取决于所到达的特定极限值。

连线从站机架(Wired Slave Rack)

由连线型远程 I/O 从站单元连接的从站机架。

字(word)

存储器中由 16 位组成的数据存储单元。所有的数据区包括若干个"字"。有些数据区只能按"字存取,另一些数据区既能按"字",又能按"位"存取。

字地址(word address)

存储器中存储一个数据字的内存单元。字地址必须说明哪个数据区和正在寻址的字的序号。字地址有时也可缺省。

字乘倍数(word multiplier)

将值(0~3)赋给远程 I/O 系统中的主站单元(Master),以便把这些字能分配给系统中非机架安装的单元,字地址加上 32 乘以字倍数,成为新的地址。

工作位(work bit)

工作字中的一位。

工作字(work word)

在存储器中的一个"工作空间",用于编程时数据计算或其他操作的字。内部数据区(IR 区)的大部分是用于存放工作字,没特殊要求的其他部分数据区也可用于工作字。如链接数据区(LR 区)不用于 PC 链接或网络链接系统时,也可作工作字用。

欧姆龙(中国)有限公司
欧姆龙亚洲有限公司

上海办事处	021-50372222
南京办事处	025-4726876
武汉办事处	027-65776566
苏州办事处	0512-8669277
北京办事处	010-83913005
山东办事处	0531-2929795
辽宁办事处	024-22566105
广州办事处	020-87320508
厦门办事处	0592-5117709
西安办事处	029-5381152
成都办事处	028-6765345
重庆办事处	023-63803720
昆明办事处	0871-5366019

授权经销商

技术咨询

电子邮件: omron@omron.com.cn
网　　址: http://www.omron.com
800免费技术咨询电话: 800-820-4535

本书在印刷前进行仔细校对，以期无误。各种规格、参数最终以产品说明书为准。

样本编号 OEZ-ZCP97201A

规格可能改变，恕不另行通知。

欧姆龙(中国)有限公司　　版权所有

上海印刷
200210S02