Cat. No. W483-CN5-07

SYSMAC CP 系列

CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □
CP2E-E □□ D □ - □
CP2E-S □□ D □ - □
CP2E-N □□ D □ - □

CP1E/CP2E CPU 单元

指令参考手册

OMRON

SYSMAC CP 系列
CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □
CP2E-E □□ D □ - □
CP2E-S □□ D □ - □
CP2E-N □□ D □ - □
CP1E/CP2E CPU 单元

指令参考手册

2019 年 10 月修订

引言

感谢您购买 SYSMAC CP 系列 CP1E/CP2E 可编程序控制器。

本手册包含使用 CP1E/CP2E 所需的信息，请务必在使用 CP1E/CP2E 前通读并理解本手册的内容。

面向读者

本手册主要供下列人员使用，这些人员必须具备电气系统相关知识 (电气工程师或同等水平者)。

・负责 FA 系统安装的人员
・负责 FA 系统设计的人员
・负责 FA 系统及设备管理的人员

适用产品

● CP1E CPU 单元

・基本型 CP1E-E □□ (S)D □ - □
CPU 单元的基本型，支持运用基本、传送、算术和比较等指令实现基本控制应用。
・应用型 CP1E-N/NA □□ (S □)D □ - □
CPU 单元的应用型，支持与可编程终端、变频器和伺服驱动器的连接。

● CP2E CPU 单元

・基本型 CP2E-E □□ D □ - □
CPU 单元的基本型，支持与可编程终端的连接，使用基本、传送、算术和比较等指令实现基本控制应用。
・标准型 CP2E-S □□ D □ - □
CPU 单元的标准型，支持与可编程终端、变频器和伺服驱动器的连接。
・网络型 CP2E-S □□ D □ - □
CPU 单元的网络型，支持 Ethernet 的连接，定位功能增强，可实现 4 轴直线插补和脉冲。

CP 系列以 CP1H、CP1L、CP1E 和 CP2E CPU 单元为核心，采用与 CS 和 CJ 系列相同的基本结构。

扩展 I/O 容量时，请务必使用 CP 系列扩展单元和 CP 系列扩展 I/O 单元。

CP1E/CP2E CPU 单元手册

以下手册中提供与 CP1E/CP2E CPU 单元相关的信息。

有关所需信息，请参考相应的手册。

CP1E CPU单元硬件
操作手册(手册编号：W479)
CP2E CPU单元硬件
操作手册(手册编号：W613)

CP1E CPU单元软件
操作手册(手册编号：W480)
CP2E CPU单元软件
操作手册(手册编号：W614)

本手册

CP1E/CP2E CPU单元指令
参考手册(手册编号：W483)

1 安装和设置硬件

- 所有单元的部件名称与功能
- CPU单元的基本系统配置
- 扩展I/O单元和扩展单元的连接方法

2 配线

- 电源配线方法
- 外部I/O设备和扩展I/O单元或扩展单元的配线方法

3 在线连接至PLC

CX–Programmer支持软件连接电缆

连接CX–Programmer支持软件的步骤

4 软件设定

CPU单元的软件设定方法(PLC设置)

5 创建程序

- 程序类型和基本信息
- CPU单元运行
- 内部存储器
- 内置CPU功能
- 各种设定

编程指令的详细信息

6 检查和调试操作

- 检查I/O接线、设定辅助区及执行试运转
- 利用CX–Programmer进行监控和调试

7 维护和故障检修

发生问题时的错误代码和纠正措施

手册构成

CP1E/CP2E CPU 操作手册由下表列出的章节构成。请根据需要参阅相关章节。

CP1E/CP2E CPU 单元指令参考手册 (手册编号：W483)(本手册)

章节	内容
第 1 章 指令摘要	本章节介绍了 CP1E/CP2E CPU 单元使用的指令摘要。
第 2 章 指令	本章节介绍了 CP1E/CP2E CPU 单元支持的功能、操作数和指令程序示例。
第 3 章 指令执行时间和步数	本章节介绍了 CP1E/CP2E CPU 单元支持的所有指令的执行时间。
第 4 章 循环时间的监控和计算	本章节介绍了如何监控和计算可在程序中使用的 CP1E/CP2E CPU 单元的循环时间。
附录	附录中包括 CP1E/CP2E CPU 单元使用的按助记符顺序编排的指令列表和 ASCII 码表。

CP1E CPU 单元软件操作手册 (手册编号：W480)

章节	内容
第 1 章 概述	本章节介绍了 CP1E 的概况及其应用步骤。
第 2 章 CPU 单元内部存储器	本章节介绍了 CP1E CPU 单元的内部存储器类型及保存的数据。
第 3 章 CPU 单元运行	本章节介绍了 CP1E CPU 单元的运行情况。
第 4 章 编程概念	本章节介绍了 CP1E CPU 单元梯形图程序设计的基本信息。
第 5 章 I/O 存储器	本章节介绍了 CP1E CPU 单元的 I/O 存储区类型及其详情。
第 6 章 I/O 分配	本章节介绍了用于 CP1E CPU 单元和其它单元之间数据交换的 I/O 分配。
第 7 章 PLC 设置	本章节介绍了 PLC 设置 (用于执行 CP1E CPU 单元的基本设定) 的详情。
第 8 章 内置功能和分配概述	本章节介绍了内置功能及其全面应用流程和功能分配。
第 9 章 快速响应输入	本章节介绍了可用于读取比循环时间更短的信号的快速响应输入。
第 10 章 中断	本章节介绍了 CP1E PLC 可使用的中断，包括输入中断和定时中断。
第 11 章 高速计数器	本章节介绍了高速计数器输入、高速计数器中断及频率测量功能。
第 12 章 脉冲输出	本章节介绍了定位功能，如梯形控制、点动和原点搜索。
第 13 章 PWM 输出	本章节介绍了可变占空比脉冲 (PWM) 输出情况。
第 14 章 串行通信	本章节介绍了不需要通信编程的可编程终端 (PT) 通信、通用器件的无协议通信以及与 Modbus-RTU 简易主站、串行 PLC 链接和上位计算机的连接。
第 15 章 模拟 I/O 功能	本章节介绍了 NA 型 CPU 单元的内置模拟功能。
第 16 章 其他功能	本章节介绍了 PID 温度控制、时钟功能、DM 备份功能和安全功能。
第 17 章 Ethernet 选件板	本章节介绍了 Ethernet 选件板的概况、设置方法、I/O 存储器分配、故障诊断、连接 CX-Programmer 以及安装 Ethernet 的方法。
第 18 章 模拟量选件板	本章节介绍了模拟量选件板的概况、安装和设置方法、存储器分配、启动运行、更新时间、故障诊断以及使用方法。
第 19 章 编程设备操作	本章节介绍了 CX-Programmer 的基本功能，如使用 CX-Programmer 编写梯形图程序来控制 CP1E CPU 单元、传送程序到 CP1E CPU 单元以及进行程序调试等。
附录	附录中介绍了编程指令列表、辅助区、循环时间响应性能及断电时的 PLC 性能。

CP1E CPU 单元硬件操作手册 (手册编号：W479)

章节	内容
第 1 章 概述及规格	本章节介绍了 CP1E 的概况、特性及规格。
第 2 章 基本系统配置和设备	本章节介绍了 CP1E 的系统配置及单元型号。
第 3 章 部件名称及功能	本章节介绍了 CP1E PLC 的 CPU 单元、扩展 I/O 单元和扩展单元的各部分的名称与功能。
第 4 章 编程设备	本章节介绍了用于 PLC 编程和调试的 CX-Programmer 的各项功能以及通过 USB 连接 PLC 与编程设备的方法。
第 5 章 安装与配线	本章节介绍了安装 CP1E 单元以及配线的方法。
第 6 章 故障检修	本章节介绍了 CP1E PLC 运行时出现故障的检修方法，其中包括 CP1E 单元显示的出错信息。
第 7 章 维护和检查	本章节介绍了定期检查、电池使用寿命以及更换电池的方法。
第 8 章 扩展单元及扩展 I/O 单元的使用	本章节介绍了扩展单元的应用方法。
附录	附录中介绍了有关 CP1E 的尺寸、配线图和串行通信配线的相关信息。

CP2E CPU 单元软件操作手册 (手册编号：W614)

章节	内容
第 1 章 概述	本章节介绍了 CP2E 的概况及其应用步骤。
第 2 章 CPU 单元内部存储器	本章节介绍了 CP2E CPU 单元的内部存储器类型及保存的数据。
第 3 章 CPU 单元运行	本章节介绍了 CP2E CPU 单元的运行情况。
第 4 章 编程概念	本章节介绍了 CP2E CPU 单元梯形图程序设计的基本信息。
第 5 章 I/O 存储器	本章节介绍了 CP2E CPU 单元的 I/O 存储区类型及其详情。
第 6 章 I/O 分配	本章节介绍了用于 CP2E CPU 单元和其它单元之间数据交换的 I/O 分配。
第 7 章 PLC 设置	本章节介绍了 PLC 设置 (用于执行 CP2E CPU 单元的基本设定) 的详情。
第 8 章 内置功能和分配概述	本章节介绍了内置功能及其全面应用流程和功能分配。
第 9 章 快速响应输入	本章节介绍了可用于读取比循环时间更短的信号的快速响应输入。
第 10 章 中断	本章节介绍了 CP2E PLC 可使用的中断，包括输入中断和定时中断。
第 11 章 高速计数器	本章节介绍了高速计数器输入、高速计数器中断及频率测量功能。
第 12 章 脉冲输出	本章节介绍了定位功能，如梯形控制、点动和原点搜索。
第 13 章 PWM 输出	本章节介绍了可变占空比脉冲 (PWM) 输出情况。
第 14 章 串行通信	本章节介绍了不需要通信编程的可编程终端 (PT) 通信、通用器件的无协议通信以及与 Modbus-RTU 简易主站、串行 PLC 链接、上位计算机和 Modbas-RTU 从站的连接。
第 15 章 内置 Ethernet 功能	本章节介绍了内置 Ethernet 功能的概况、规格及设定方法。
第 16 章 其它功能	本章节介绍了 PID 温度控制、时钟功能、DM 备份功能和安全功能。
第 17 章 模拟量选件板	本章节介绍了模拟量选件板的概况、安装和设置方法、存储器分配、启动运行、更新时间、故障诊断以及使用方法。
第 18 章 编程设备操作	本章节介绍了 CX-Programmer 的基本功能，如使用 CX-Programmer 编写梯形图程序来控制 CP2E CPU 单元、传送程序到 CP2E CPU 单元以及进行程序调试等。
附录	附录中介绍了编程指令列表、辅助区、循环时间响应性能、断电时的 PLC 性能、存储映射图以及 Ethernet 功能。

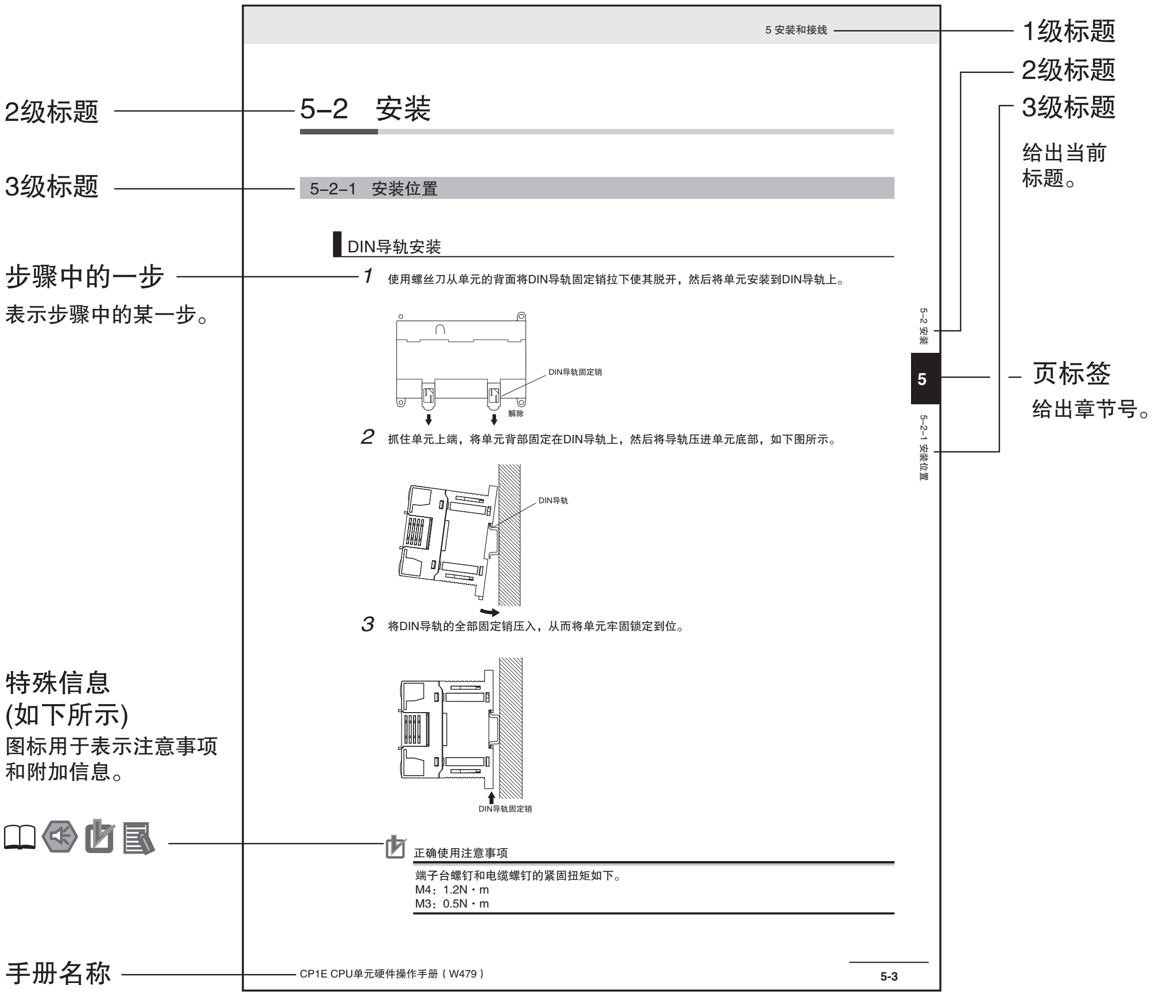
CP2E CPU 单元硬件操作手册 (手册编号：W613)

章节	内容
第 1 章 概述及规格	本章节介绍了 CP2E 的概况、特性及规格。
第 2 章 基本系统配置和设备	本章节介绍了 CP2E 的系统配置及单元型号。
第 3 章 部件名称及功能	本章节介绍了 CP2E PLC 的 CPU 单元、扩展 I/O 单元和扩展单元的各部分的名称与功能。
第 4 章 编程设备	本章节介绍了用于 PLC 编程和调试的 CX-Programmer 的各项功能以及通过 USB、Ethernet 和串行端口连接 PLC 与编程设备的方法。
第 5 章 安装与配线	本章节介绍了安装 CP2E 单元以及配线的方法。
第 6 章 故障检修	本章节介绍了 CP2E PLC 运行时出现故障的检修方法，其中包括 CP2E 单元显示的出错信息。
第 7 章 维护和检查	本章节介绍了定期检查、电池使用寿命以及更换电池的方法。
第 8 章 扩展单元及扩展 I/O 单元的使用	本章节介绍了扩展单元的应用方法。
附录	附录中介绍了有关 CP2E 的尺寸、配线图、串行通信配线、网络安装的相关信息以及 CP1E 与 CP2E 的比较。

手册结构

页面结构和图标

本手册采用下列页面结构和图标。



本图仅用作示例，本手册中可能无相关的文字描述。

特殊信息

本手册中的特殊信息分类如下：

安全使用注意事项
关于该做什么与不该做什么的注意事项，旨在确保产品的安全使用。

正确使用注意事项
关于该做什么与不该做什么的注意事项，旨在确保产品的正确操作和运转。

附加信息
加深理解和简化操作的附加信息

用于参考的详细信息或相关信息出处。

术语和注释

	术语	描述
CP1E	E 型 CPU 单元	CPU 单元基本型，支持使用基本、传送、算术和比较指令等实现基本控制应用。 本手册将 CPU 单元的基本型称为 “E □□ (S) 型 CPU 单元”。 E □□ (S) 型 CPU 单元的型号如下。 CP1E-E □□ D □ - □ CP1E-E □□ SD □ - □
	N 型 CPU 单元	CPU 单元应用型，支持与可编程终端、变频器和伺服驱动器的连接。 本手册将 CPU 单元的应用型称为 “N □□ (S) 型 CPU 单元”。 N □□ (S) 型 CPU 单元的型号如下。 CP1E-N □□ D □ - □ CP1E-N □□ SD □ - □ CP1E-N □□ S1D □ - □
	NA 型 CPU 单元	CPU 单元应用型，支持内置模拟器件以及与可编程终端、变频器和伺服驱动器的连接。 本手册将带内置模拟量的 CPU 单元的应用型号称为 “NA □□型 CPU 单元”。 NA □□型 CPU 单元的型号如下。 CP1E-NA □□ D □ - □
CP2E	E 型 CPU 单元	CPU 单元基本型，支持与可编程终端的连接，使用基本、传送、算术和比较等指令实现基本控制应用。 本手册将 CPU 单元的基本型称为 “E □□型 CPU 单元”。 E □□型 CPU 单元的型号如下所示。 CP2E-E □□ D □ - □
	S 型 CPU 单元	CPU 单元标准型，支持与可编程终端、变频器和伺服驱动器的连接。 本手册将 CPU 单元的标准型称为 “S □□型 CPU 单元”。 S □□型 CPU 单元的型号如下所示。 CP2E-S □□ D □ - □
	N 型 CPU 单元	CPU 单元网络型，支持 Ethernet 的连接，定位功能增强，可实现 4 轴直线插补和脉冲。 本手册将 CPU 单元的网络型称为 “N □□型 CPU 单元”。 N □□型 CPU 单元的型号如下所示。 CP2E-N □□ D □ - □
CX-One CX-Programmer		用于编程和调试 PLC 的编程设备。 ・CP1E 支持的版本 CX-Programmer 8.2 或更高版本支持 E20/30/40(S) 和 N20/30/40(S □) CPU 单元。 CX-Programmer 9.03 或更高版本支持 E10/14(S)、N14/60(S □) 和 NA20 CPU 单元。 CX-Programmer 9.42 或更高版本支持 E60S CPU 单元。 ・CP2E 支持的版本 CX-One 4.51 或更高版本以及 CX-Programmer 9.72 或更高版本支持 CP2E CPU 单元。

本手册中的章节

目录

协议条款和条件

保证及有限责任声明

保证声明

● **排他性保证**

OMRON 的排他性保证是指产品自售出之日起十二个月 (或 OMRON 书面确认的其它指定期间) 内在材料和工艺上无缺陷。OMRON 对于所有其它明示或暗示的保证概不负责。

● **有限责任**

OMRON 未以明示或暗示的方式表述或保证产品的非侵权性、适销性或特定用途的适用性。买方同意自主决定这些产品是否适当满足其预定用途。

OMRON 对任何由产品或知识产权侵权所产生的任何形式的索赔和费用概不承担责任。

● **买方补救措施**

按照本协议规定，OMRON 的责任仅限于以下几种形式且 OMRON 有权决定采取何种形式：(i) 更换不合格品 (欧姆龙只负责前期装运费用，后期拆卸或更换产品产生的劳务费由买方负责)、(ii) 维修不合格品或 (iii) 偿还买方等同于购买不合格品的价款；除非 OMRON 经分析后确认产品的使用、存放、安装和维护得当且未遭污染、滥用、误用或者不当改造或修理，否则在任何情况下，OMRON 对于与产品相关的保证、修理或其它主张不承担任何责任。买方必须在装运前征得 OMRON 的书面同意后方可将产品返还给 OMRON。OMRON 公司对其产品与任何电气或电子部件、电路、系统组件或其他任何材料、物质或环境组合使用时的适用性、非适用性及引起的后果概不负责。任何口头或书面形式的建议、推荐或信息均不得视为上述保证声明的修改或补充内容。

关于公布信息，请访问网站 http://www.omron.com/global/ 或垂询 OMRON 代理商。

有限责任等

OMRON 公司对于任何与产品相关的特殊、间接或直接损坏、利润损失或商业损失概不负责，不论此类索赔是基于合同、保证、疏忽还是严格责任。

此外，在任何情况下，OMRON 公司对于超出产品单价的索赔部分免责。

应用注意事项

适用性声明

OMRON 公司对于买方在其应用中的产品组合或产品使用的标准、规范或条例方面的合规性不承担任何责任。根据买方的要求，OMRON 将提供相应的第三方认证来明确适用于产品的额定值和使用限制。此信息本身不足以充分确定产品与终端产品、机器、系统及其它应用或用途组合的适用性。买方应自行负责确定该产品和相关应用、产品或系统的适用性。买方应始终承担应用责任。

如果产品整体设计不足以应对此类风险，且未在整个设备或系统内针对特定用途妥善调校并安装 OMRON 产品，则不得将产品用于存在严重人身或财产隐患的场合。

可编程产品

使用可编程产品时，OMRON 公司不对用户的程序或其引起的后果承担任何责任。

免责声明

性能数据

OMRON 公司网站、样本和其它材料中提供的性能数据仅供用户作为确定适用性的参考，并不予以担保。这些数据仅表示在 OMRON 测试条件下的结果，用户必须将其与实际应用条件相联系。实际性能遵守 OMRON 保证声明和有限责任条款的规定。

规格变更

基于产品改进和其它原因，产品规格及附件可能会随时变更。公司通常在公布规格、性能或重大结构变更后更改部件编号，但对某些产品规格进行变更时并不另行通知。在不确定规格时，我们会根据客户的要求为其应用场合指定特殊的部件编号或设立关键的规格。请随时垂询 OMRON 代理商以确认所购产品的实际规格。

错误与疏漏

OMRON 公司所述信息经仔细审核，力求准确无误；但对于笔误、排版或校对错误或疏漏，我方概不负责。

安全注意事项

安全注意信息的定义

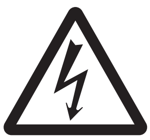
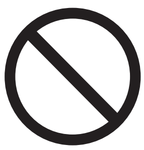
以下标识用于本手册中，以提供 CP 系列 PLC 安全使用所需的注意事项。安全注意事项对于安全使用至关重要。因此，请务必阅读并理解安全注意事项中包含的信息。

警告	表示紧迫的危险情况，如不加以避免，将会造成死亡或严重伤害。 此外，还可能导致严重的财产损失。
注意	表示潜在的危险状况，如不加以避免，可能会造成轻度或中度伤害或财产损失。

安全使用注意事项
表示该做什么与不该做什么的注意事项，旨在确保产品的安全使用。

正确使用注意事项
表示该做什么与不该做什么的注意事项，旨在确保产品的正确操作和运转。

符号

	该三角形符号表示注意事项(包括警告)。具体内容显示在三角形中并通过文本解释。该示例表示与触电相关的注意事项。
	圆圈和斜线符号表示应禁止执行的操作。具体内容显示在圆圈中并通过文本解释。
	实心圆圈符号表示应强制执行的操作。具体内容显示在圆圈中并通过文本解释。该示例表示必须加以执行的一般注意事项。
	该三角形符号表示注意事项(包括警告)。具体内容显示在三角形中并通过文本解释。该示例表示一般注意事项。
	该三角形符号表示注意事项(包括警告)。具体内容显示在三角形中并通过文本解释。该示例表示与灼热表面相关的注意事项。

警告

当传送程序、访问 I/O 存储器、执行修改 I/O 存储器的操作时，请务必充分确认目的地的安全。

否则，不论 CPU 单元处于何种运行模式下，连接至 PLC 输出端的设备都可能会产生误操作。

注意

针对 CP1E E □□ (S) 型 CPU 单元或无电池的 CP1E N/NA □□ (S) 型 CPU 单元，接通电源时，DM 区 (D)*、保持区 (H)、计数器当前值 (C)、计数器完成标志 (C) 状态和辅助区中 (A) 与时钟功能相关的位状态的内容可能会不稳定。

* 该情况不适于使用 DM 备份功能备份到 EEPROM 中的区。

如果使用了 DM 备份功能，请务必使用下列方法之一进行初始化。

1. 将所有区清零

在 PLC 设置*的"启动时数据读取"中*，选中*"清除保持的存储器 (HR/DM/CNT)"*复选框。

2. 将指定区清零或初始化到指定值通过梯形图程序进行设定。

如果数据未被初始化，则单元或设备可能会因数据不稳定而出现意外操作。

请务必在确认延长循环时间不会引起不良影响后，再执行在线编辑。

否则，可能会导致输入信号无法读取。

如果在 CP1E-N/NA □□ (S □)D □ – □ CPU 单元中安装了电池，则 DM 区 (D)、保持区 (H)、计数器完成标志 (C) 和计数器当前值 (C) 将通过电池进行保持。但当电池电压过低时，保持的 I/O 存储区 (包括 DM 区、保持区和计数器区) 将会变得不稳定。单元或设备可能会因数据不稳定而出现意外操作。

若外部输出由基于 DM 区或其它 I/O 存储区内容的梯形图程序来完成，则可通过电池错误标志或其它方法来停止输出。

若在梯形图窗口中监视 I/O 位状态或当前值，或者在监测窗口中监测当前值，则需进行充分的安全检查。

不管处于何种运行模式下，如果由于不小心按下快捷键而产生置位、复位、强制置位或强制复位，则连接至 PLC 输出端的设备可能会出现误操作。

⚠ 注意

采用字地址或符号指定偏移量时，请编写相关程序以确保不超出起始地址的存储区范围。

例如，编写程序时使用输入比较指令或其它指令，从而确保仅在间接指定没有导致末尾地址超出存储区范围时才执行程序。

如果间接指定导致末尾地址超出起始地址的存储区范围，则系统将访问其它区中的数据，并可能出现意外操作。

根据连接到单元的温度传感器类型设定温度范围。

如果温度范围与传感器不匹配，温度数据将无法被正确转换。

请勿将温度范围设定为指定温度范围以外的值。

错误的设定可能会导致运行错误。

安全使用注意事项

使用 CP 系列 PLC 系统时，请务必遵守以下注意事项。

● 使用

- 初始化 DM 区时，请使用以下方法之一将 DM 区的初始内容备份到内置 EEPROM 或闪存中。
  - 在“启动时数据读取”的 *“备份 DM 的 CH 数”*框中设置从 D0 开始的要备份的 DM 区字数。
  - 包括通过 A751.15(DM 备份保存起始位) 置 ON 而将 DM 区中指定字备份至内置 EEPROM 或闪存中的编程操作。
- 在单元上运行梯形图程序前，请确认其可以正确执行，否则可能会导致意外操作。
- CP1E/CP2E CPU 单元中的梯形图程序和参数区数据备份在内置 EEPROM 或闪存中。备份操作执行过程中，CPU 单元前面的 BKUP 指示灯将会亮起。此时，请勿关闭 CPU 单元的电源，否则，不仅无法备份数据，而且在下次接通电源时将会发生存储器错误。
- 对于 CP1E/CP2E CPU 单元，可将数据存储器中的内容备份到内置 EEPROM 或闪存中。备份操作执行过程中，CPU 单元前面的 BKUP 指示灯将会亮起。此时，请勿关闭 CPU 单元的电源，否则，不仅无法备份数据，而且在下一次接通电源时无法将数据传送到 RAM 内的 DM 区。
- 当时钟数据运用于程序时，请安装电池 （另售）。如果不安装电池，在电源切断时，时钟数据将被初始化，可能导致会程序误动作。
- 使用电池时，请设定 PLC 设置的 “检测电池低电量”。如果不更改默认设定，在电池耗尽时将无法检测出错误，从而导致使用时钟数据的程序误动作。
- 更换电池前，应向 CPU 单元持续供电至少 30 分钟，然后在关闭电源后的 5 分钟内换好电池。若未遵守该注意事项，可能会损坏时钟数据。
- 若参数设置不当，可能会造成设备意外操作。即使设置了适当的参数，也须在将参数传输至 CPU 单元前确认设备不会受到不良影响。
- 开始运行前，请确认 DIP 开关和 DM 区的设定准确无误。
- 在更换 CPU 单元后，请确保在恢复运行前已将 DM 区、保持区及其它存储区的必要数据传输至新 CPU 单元。
- 请勿试图拆解、修理或改造任何单元，否则可能导致误动作、起火或触电。
- 在进行以下任何一项操作前，请确认其不会对系统造成不良影响，否则可能会导致意外操作。
  - 改变 PLC 的运行模式 (包括启动运行模式的设置)。
  - 强制置位 / 强制复位存储器中的任意位；
  - 改变存储器中的任一字或设定值的当前值。

● 外部电路

- 请务必对外部电路进行配置，从而在接通 PLC 电源后再接通控制系统电源。若先接通控制系统电源后再接通 PLC 电源，则在接通 PLC 电源时，DC 输出单元和其它单元上的输出端子上的状态会瞬间变为 ON，从而导致控制系统信号临时出错。
- 若内部电路出现故障，则可能导致输出端子保持 ON 状态 (常见于继电器、晶体管及其它元器件)，因此客户须采取适当的防护措施以保障安全。
- 若 I/O 保持位置 ON，则当从 RUN 或 MONITIOR 模式切换到 PROGRAM 模式时，PLC 的输出不会置 OFF，并将保持其原有状态。请确保外部负载不会在上述过程中构成危险因素。(当操作因致命错误而停止时，包括 FALS 指令所产生的错误，PLC 的所有输出都会变为 OFF，且仅保持 CPU 单元的内部输出状态)。

规定和标准

商标

SYSMAC 为欧姆龙公司生产的可编程控制器的注册商标。

CX-One 为欧姆龙公司开发的编程软件的注册商标。

Windows 是美国微软公司的注册商标。

本文中所述的其它系统和产品名称分别为各家公司的商标或注册商标。

软件许可证和版权

本产品包含第三方软件。与该软件相关的许可证和版权，如下所示。

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIEDWARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

相关手册

以下手册与 CP1E/CP2E 密切相关，请与本手册一起使用。

<table>
<tr><th>手册名称</th><th>手册编号</th><th>型号</th><th>用途</th><th>内容</th></tr>
<tr><td>SYSMAC CP 系列
CP1E/CP2E CPU 单元
指令参考手册
(本手册)</td><td>W483</td><td>CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □</td><td>用于深入了解程序指令</td><td>本手册对各程序指令进行了详细说明。
编程时，请结合 CP1E CPU 单元软件操作手册 (手册编号：W480) 一起使用。</td></tr>
<tr><td rowspan="2">SYSMAC CP 系列
CP1E CPU 单元软件
操作手册</td><td rowspan="2">W480</td><td rowspan="2">CP1E-E □□ SD □ - □
CP1E-N□□S□D□-□
CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □</td><td>用于了解 CP1E PLC 的软件规格</td><td>本手册从以下几个方面对CP1E PLC进行了说明。
・ CPU 单元的操作
・ 内部存储器
・ 编程
・ 设定
・ CPU 单元内置功能
　・ 中断
　・ 高速计数器输入
　・ 脉冲输出
　・ 串行通信
　・ 模拟量 I/O 功能
　・ 其它功能</td></tr>
<tr><td colspan="2">请结合 CP1E CPU 单元硬件操作手册 (手册编号：W479) 和指令参考手册 (手册编号：W483) 一起使用。</td></tr>
<tr><td rowspan="2">SYSMAC CP 系列
CP1E CPU 单元硬件
操作手册</td><td rowspan="2">W479</td><td rowspan="2">CP1E-E □□ SD □ - □
CP1E-N□□S□D□-□
CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □</td><td>用于了解 CP1E PLC 的硬件规格</td><td>本手册从以下几个方面对CP1E PLC进行了说明。
・ 概要及特性
・ 基本系统配置
・ 部件名称及功能
・ 安装与设定
・ 故障诊断</td></tr>
<tr><td colspan="2">请结合 CP1E CPU 单元软件操作手册 (手册编号：W480) 和指令参考手册 (手册编号：W483) 一起使用。</td></tr>
<tr><td rowspan="2">SYSMAC CP 系列
CP2E CPU 单元软件
操作手册</td><td rowspan="2">W614</td><td rowspan="2">CP2E-E □□ D □ - □
CP2E-S □□ D □ - □
CP2E-N □□ D □ - □</td><td>用于了解 CP2E PLC 的软件规格</td><td>本手册从以下几个方面对CP2E PLC进行了说明。
・ CPU 单元的操作
・ 内部存储器
・ 编程
・ 设定
・ CPU 单元内置功能
　・ 中断
　・ 高速计数器输入
　・ 脉冲输出
　・ 串行通信
　・ 内置 Ethernet
　・ 其它功能</td></tr>
<tr><td colspan="2">请结合 CP2E CPU 单元硬件操作手册 (手册编号：W613) 和指令参考手册 (手册编号：W483) 一起使用。</td></tr>
<tr><td rowspan="2">SYSMAC CP 系列
CP2E CPU 单元硬件
操作手册</td><td rowspan="2">W613</td><td rowspan="2">CP2E-E □□ D □ - □
CP2E-S □□ D □ - □
CP2E-N □□ D □ - □</td><td>用于了解 CP2E PLC 的硬件规格</td><td>本手册从以下几个方面对CP2E PLC进行了说明。
・ 概要及特性
・ 基本系统配置
・ 部件名称及功能
・ 安装与设定
・ 故障诊断</td></tr>
<tr><td colspan="2">请结合 CP2E CPU 单元软件操作手册 (手册编号：W614) 和指令参考手册 (手册编号：W483) 一起使用。</td></tr>
</table>

<table>
<tr><th>手册名称</th><th>手册编号</th><th>型号</th><th>用途</th><th>内容</th></tr>
<tr><td rowspan="2">CS/CJ/CP/NSJ 系列
通信命令参考手册</td><td rowspan="2">W342</td><td rowspan="2">CS1G/H-CPU □□ H
CS1G/H-CPU □□ -V1
CS1D-CPU □□ H
CS1D-CPU □□ S
CS1W-SCU □□ -V1
CS1W-SCB □□ -V1
CJ1G/H-CPU □□ H
CJ1G-CPU □□ P
CJ1M-CPU □□
CJ1G-CPU □□
CJ1W-SCU □□ -V1</td><td>用于深入了解 CS/CJ/CP/NSJ 系列控制器的通信指令</td><td>详细内容
1) C 模式命令详解
2) FINS 命令详解
请阅读本手册，以便深入了解 C 模式和对 CPU 单元进行寻址的 FINS 指令。</td></tr>
<tr><td colspan="2">注 本手册仅对 CPU 单元寻址指令进行了说明，并未涉及对其它类型单元或端口进行寻址的指令(如CPU单元上的串行通信端口、串行通信单元/板上的通信端口及其它通信单元)。</td></tr>
<tr><td>SYSMAC CP 系列
CP1L/CP1E CPU 单元
入门手册</td><td>W461</td><td>CP1L-L10D □ - □
CP1L-L14D □ - □
CP1L-L20D □ - □
CP1L-M30D □ - □
CP1L-M40D □ - □
CP1L-M60D □ - □
CP1E-E □□ D □ - □
CP1E-N □□ D □ - □
CP1E-NA □□ D □ - □</td><td>用于了解 CP1L/CP1E PLC 的基本设置方法</td><td>本手册从以下几个方面对 CP1L/CP1E PLC 进行了说明。
・基本配置和部件名称
・安装和接线
・使用 CX-Programmer 进行编程、数据传输和调试
・应用程序示例</td></tr>
<tr><td>CX-One 设置手册</td><td>W463</td><td rowspan="5">CXONE-AL □□ D-V4</td><td>用于通过 CX-One 安装软件</td><td>本手册对 FA 集成工具包 CX-One 的概要和安装方法进行了说明。</td></tr>
<tr><td>CX-Programmer 操作手册</td><td>W446</td><td rowspan="2">用于了解 Windows 计算机用编程软件 CX-Programmer 的操作步骤</td><td rowspan="2">本手册对 CX-Programmer 的操作步骤进行了说明。</td></tr>
<tr><td>CX-Programmer 功能块 / 结构化文本操作手册</td><td>W447</td></tr>
<tr><td>CX-Simulator 操作手册</td><td>W366</td><td>用于了解 Windows 计算机用仿真软件 CX-Simulator 的操作步骤</td><td>本手册对 CX-Simulator 的操作步骤进行了说明。</td></tr>
<tr><td>CS/CJ/CP/NSJ 系列
CX-Integrator 操作手册</td><td>W464</td><td>用于网络设定和监控</td><td>本手册对 CX-Integrator 的操作步骤进行了说明。</td></tr>
</table>

1

指令摘要

本章节介绍了 CP1E/CP2E CPU 单元使用的指令摘要。

1-1 指令摘要

CP1E/CP2E 支持多达 220 种指令。

下表所示为按功能划分的各项指令。有关各项指令的详情，请参阅各参考页对应的内容。

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
顺序输入指令	载入	LD	–	指定一个逻辑开始，并根据指定操作位的 ON/OFF 状态建立一个 ON/OFF 执行条件。	○	○	2-7
		@LD	–				
		%LD	–				
		!LD	–				
		!@LD	–				
		!%LD	–				
	载入非	LD NOT	–	指定一个逻辑开始，并根据指定操作位的 ON/OFF 状态的取反结果建立一个 ON/OFF 执行条件。	○	○	2-7
		@LD NOT	–				
		%LD NOT	–				
		!LD NOT	–				
		!@LD NOT	–				
		!%LD NOT	–				
	与	AND	–	将指定操作位的状态和当前执行条件进行逻辑与操作。	○	○	2-9
		@AND	–				
		%AND	–				
		!AND	–				
		!@AND	–				
		!%AND	–				
	与非	AND NOT	–	将指定操作位的状态取反后和当前执行条件进行逻辑与操作。	○	○	2-9
		@AND NOT	–				
		%AND NOT	–				
		!AND NOT	–				
		!@AND NOT	–				
		!%AND NOT	–				
	或	OR	–	将指定操作位的 ON/OFF 状态和当前执行条件进行逻辑或操作。	○	○	2-11
		@OR	–				
		%OR	–				
		!OR	–				
		!@OR	–				
		!%OR	–				
	或非	OR NOT	–	将指定位的状态取反后和当前执行条件进行逻辑或操作。	○	○	2-11
		@OR NOT	–				
		%OR NOT	–				
		!OR NOT	–				
		!@OR NOT	–				
		!%OR NOT	–				
	逻辑块与	AND LD	–	在逻辑块之间进行逻辑与。	○	○	2-13
	逻辑块或	OR LD	–	在逻辑块之间进行逻辑或。	○	○	2-13
	非	NOT	520	执行条件取反。	○	○	2-16
	条件 ON	UP	521	当执行条件从 OFF → ON 时，UP(521) 将执行条件在一个循环内变为 ON。	○	○	2-17
	条件 OFF	DOWN	522	当执行条件从 ON → OFF 时，DOWN(522) 将执行条件在一个循环内变为 ON。	○	○	2-17
	LD 型・位测试	LD TST	350	指定位为 1 时，在下段进行 LD 连接。	–	○	2-18
	LD型・非位测试	LD TSTN	351	指定位为 0 时，在下段进行 LD 连接。	–	○	2-18
	AND 型・位测试	AND TST	350	指定位为 1 时，在下段进行 AND 连接。	–	○	2-20
	AND 型・非位测试	AND TSTN	351	指定位为 0 时，在下段进行 AND 连接。	–	○	2-20
	OR 型・位测试	OR TST	350	指定位为 1 时，在下段进行 OR 连接。	–	○	2-22
	OR型・非位测试	OR TSTN	351	指定位为 0 时，在下段进行 OR 连接。	–	○	2-22

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
顺序输出指令	输出	OUT	–	将逻辑运算结果 (执行条件) 输出到指定位。	○	○	2–24
		!OUT	–				
	反相输出	OUT NOT	–	将逻辑处理的结果 (执行条件) 取反后输出到指定位。	○	○	2–24
		!OUT NOT	–				
	TR 位	TR	–	当以助记符编程时，TR位用于临时保留程序中的执行条件的 ON/OFF 状态。	○	○	2–26
	保持	KEEP	011	运行方式类似于锁存继电器。	○	○	2–27
		!KEEP					
	上升沿微分	DIFU	013	当执行条件从 OFF → ON(上升沿) 时，DIFU(013) 将指定位在一个循环中变为 ON。	○	○	2–31
		!DIFU					
	下降沿微分	DIFD	014	当执行条件从 ON → OFF(下降沿) 时，DIFD(014) 将指定位在一个循环中变为 ON。	○	○	2–33
		!DIFD					
	置位	SET	–	当执行条件为 ON 时，SET 指令将操作数的位变为 ON。	○	○	2–35
		@SET	–				
		%SET	–				
		!SET	–				
		!@SET	–				
		!%SET	–				
	复位	RSET	–	当执行条件为 ON 时，RSET 指令将操作数的位变为 OFF。	○	○	2–35
		@RSET	–				
		%RSET	–				
		!RSET	–				
		!@RSET	–				
		!%RSET	–				
	多个位置位	SETA	530	SETA(530) 将指定的连续位数置 ON。	○	○	2–37
		@SETA					
	多个位复位	RSTA	531	RSTA(531) 将指定的连续位数置 OFF。	○	○	2–37
		@RSTA					
	单个位置位	SETB	532	当执行条件为 ON 时，SETB(532) 将指定字中的指定位置 ON。 与 SET 指令不同，SETB(532) 可用于对 DM 字中的位进行置位。	○	○	2–39
		@SETB					
		!SETB					
		!@SETB					
	单个位复位	RSTB	533	当执行条件为 ON 时，RSTB(533) 将指定字中的指定位置 OFF。 与 RSET 指令不同，RSTB(533) 可用于对 DM 字中的位进行复位。	○	○	2–39
		@RSTB					
		!RSTB					
		!@RSTB					

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
顺序控制指令	结束	END	001	表示一个程序结束。	○	○	2-44
	空操作	NOP	000	该指令无任何功能。(NOP(000) 不执行任何操作。)	○	○	2-45
	互锁	IL	002	当 IL(002) 的执行条件为 OFF 时，IL(002) 和 ILC(003) 之间的所有输出均互锁。	○	○	2-46
	互锁清除	ILC	003	当 IL(002) 的执行条件为 OFF 时，IL(002) 和 ILC(003) 之间的所有输出均被互锁。	○	○	2-46
	多路互锁微分保持	MILH	517	当 MILH(517) 的执行条件为 OFF 时，MILH(517) 和下一条 MILC(519) 指令之间的所有指令的输出均被互锁。	○	○	2-50
	多路互锁微分释放	MILR	518	当 MILH(518) 的执行条件为 OFF 时，MILH(518) 和下一条 MILC(519) 指令之间的所有指令的输出均被互锁。	○	○	2-50
	多路互锁清除	MILC	519	清除以 MILH(517) 或 MILR(518) 开始且具有相同互锁号的互锁。	○	○	2-50
	跳转	JMP	004	当 JMP(004) 的执行条件为 OFF 时，程序执行直接跳转至程序中具有相同跳转号的第一个 JME(005) 指令。	○	○	2-59
	跳转结束	JME	005	表明以 JMP(004) 或 CJP(510) 开始的跳转结束。	○	○	2-59
	条件跳转	CJP	510	CJP(510) 的作用与 JMP(004) 基本相反。当 CJP(510) 的执行条件为 ON 时，程序执行直接跳转至程序中具有相同跳转号的第一个 JME(005) 指令。	○	○	2-59
	FOR 循环	FOR	512	将 FOR(512) 和 NEXT(513) 之间的指令重复执行指定的次数。	○	○	2-62
	NEXT 循环	NEXT	513	将 FOR(512) 和 NEXT(513) 之间的指令重复执行指定的次数。	○	○	2-62
	循环中断	BREAK	514	用于 FOR-NEXT 循环语句中的编程，作用是对于给定的执行条件取消循环的执行。循环中剩余的指令作为 NOP(000) 指令处理。	○	○	2-65
定时器和计数器指令	100ms 定时器	TIM	-	TIM/TIMX(550) 定时器以 0.1s 为单位作减量计时。	○	○	2-72
		TIMX	550				
	10ms 定时器	TIMH	015	TIMH(015)/TIMHX(551) 定时器以 10ms 为单位作减量计时。	○	○	2-75
		TIMHX	551				
	1ms 定时器	TMHH	540	TMHH(540)/TMHHX(552) 定时器以 1ms 为单位作减量计时。	○	○	2-78
		TMHHX	552				
	累加定时器	TTIM	087	TTIM(087)/TTIMX(555) 定时器以 0.1s 为单位作增量计时。	○	○	2-80
		TTIMX	555				
	长定时器	TIML	542	TIML(542)/TIMLX(553) 定时器以 0.1s 为单位作减量计时。	○	○	2-83
		TIMLX	553				
	计数器	CNT	-	CNT/CNTX(546) 计数器作减量计数。	○	○	2-86
		CNTX	546				
	可逆计数器	CNTR	012	CNTR(012)/CNTRX(548) 操作一可逆计数器。	○	○	2-89
		CNTRX	548				
	复位定时器 / 计数器	CNR/ @CNR	545	CNR(545)/CNRX(547) 使指定的定时器或计数器号范围内的定时器或计数器复位。	○	○	2-92
		CNRX/ @CNRX	547				

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
比较指令	符号比较	=，<>，<，<=，>，>=	300 ~ 328	符号比较指令用于比较两个值的大小，并在比较结果为真时创建一个 ON 执行条件。	○	○	2-94
	时间比较	LD，AND，OR+=DT	341	时间比较指令比较两个 BCD 时间值，并在比较条件为真时生成一个 ON 执行条件。	○	○	2-97
		LD，AND，OR+<>DT	342				
		LD，AND，OR+<DT	343				
		LD，AND，OR+<=DT	344				
		LD，AND，OR+>DT	345				
		LD，AND，OR+>=DT	346				
	无符号比较	CMP	020	比较两个无符号二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。	○	○	2-101
		!CMP					
	双字无符号比较	CMPL	060	比较两个无符号双字二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。	○	○	2-101
	带符号二进制比较	CPS	114	比较两个带符号二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。	○	○	2-104
		!CPS					
	带符号双字二进制比较	CPSL	115	比较两个带符号双字二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。	○	○	2-104
	表比较	TCMP	085	将源数据与 16 个字的内容进行比较，并在内容相等时，对结果字中的对应位置 ON。	○	○	2-107
		@TCMP					
	无符号块比较	BCMP	068	将源数据与 16 个范围 (由 16 个下限和 16 个上限定义) 进行比较，并当源数据处于某个范围内时，对结果字中对应的位置 ON。	○	○	2-109
		@BCMP					
	区域范围比较	ZCP	088	将 CD 中的 16 位无符号二进制值 (字的内容或常数) 与由 LL 和 UL 定义的范围进行比较，并将结果输出至辅助区的算术标志中。	○	○	2-111
	双字区域范围比较	ZCPL	116	将 CD 和 CD+1 中的 32 位无符号二进制值 (字的内容或常数) 与由 LL 和 UL 定义的范围进行比较，并将结果输出至辅助区的算术标志中。	○	○	2-111

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
数据传送指令	传送	MOV @MOV !MOV !@MOV	021	传送数据的一个字到指定字中。	○	○	2-115
	双字传送	MOVL/ @MOVL	498	传送数据的两个字到指定字中。	○	○	2-115
	传送反	MVN/ @MVN	022	将一个字的数据的补码传送到指定字中。	○	○	2-115
	位传送	MOVB/ @MOVB	082	传送指定的位。	○	○	2-118
	数位传送	MOVD/ @MOVD	083	传送指定的数位。(每个数位由 4 个位组成。)	○	○	2-120
	多位传送	XFRB/ @XFRB	062	传送指定数目的连续位。	○	○	2-122
	块传送	XFER/ @XFER	070	传送指定数目的连续字。	○	○	2-124
	块设置	BSET/ @BSET	071	将同一个字复制到一个连续字的范围中。	○	○	2-126
	数据交换	XCHG/ @XCHG	073	交换两个指定字的内容。	○	○	2-128
	单字分配	DIST/ @DIST	080	将源字传送到目的字 (在源基址上加一个偏移值)。	○	○	2-130
	数据收集	COLL/ @COLL	081	将源字 (在源基址上加一个偏移值) 传送到目的字。	○	○	2-132
	索引寄存器设定	MOVR/ @MOVR	560	在指定的索引寄存器中设定指定 CH 编号、触点编号的 I/O 存储器有效地址。	–	○	2-134
	索引寄存器设定	MOVRW/ @MOVRW	561	在指定的索引寄存器中设定指定定时器 / 计数器当前值的 I/O 存储器有效地址。	–	○	2-134
数据移位指令	移位寄存器	SFT	010	操作移位寄存器。	○	○	2-137
	可逆数位寄存器	SFTR/ @SFTR	084	生成一个即可使数据左移又可使数据右移的移位寄存器。	○	○	2-139
	字移位	WSFT/ @WSFT	016	在 St 和 E 之间以字为单位使数据移位。	○	○	2-141
	算术左移	ASL/ @ASL	025	将 Wd 的内容左移一位。	○	○	2-143
	双字左移	ASLL/ @ASLL	570	将 Wd 和 Wd+1 的内容向左移一位。	–	○	2-143
	算术右移	ASR/ @ASR	026	将 Wd 的内容右移一位。	○	○	2-145
	双字右移	ASRL/ @ASRL	571	将 Wd 和 Wd+1 的内容向右移一位。	–	○	2-145
	循环左移	ROL/ @ROL	027	将Wd中包括进位标志(CY)在内的所有位左移一位。	○	○	2-147
	双字循环左移	ROLL/ @ROLL	572	将 Wd 和 Wd+1 中包括进位标志 (CY) 在内的所有位向左移一位。	–	○	2-147
	循环右移	ROR/ @ROR	028	将Wd中包括进位标志(CY)在内的所有位右移一位。	○	○	2-149
	双字循环右移	RORL/ @RORL	573	将 Wd 和 Wd+1 中包括进位标志 (CY) 在内的所有位向右移一位。	–	○	2-149
	一个数位左移	SLD/ @SLD	074	将数据左移一个数位 (4 个位)。	○	○	2-151
	一个数位右移	SRD/ @SRD	075	将数据右移一个数位 (4 个位)。	○	○	2-151
	左移 N 位	NASL/ @NASL	580	将指定的 16 位字数据左移指定的位数。	○	○	2-153
	双字左移 N 位	NSLL/ @NSLL	582	将指定的 32 位字数据左移指定的位数。	○	○	2-153
	右移 N 位	NASR/ @NASR	581	将指定的 16 位字数据右移指定的位数。	○	○	2-156
	双字右移 N 位	NSRL/ @NSRL	583	将指定的 32 位字数据右移指定的位数。	○	○	2-156

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
递增 / 递减指令	二进制递增	++/ @++	590	将指定字的 4 位数十六进制内容递增 1。	○	○	2-159
	双字二进制递增	++L/ @++L	591	将指定字的 8 位数十六进制内容递增 1。	○	○	2-159
	二进制递减	--/ @--	592	将指定字的 4 位数十六进制内容递减 1。	○	○	2-162
	双字二进制递减	--L/ @--L	593	将指定字的 8 位数十六进制内容递减 1。	○	○	2-162
	BCD 递增	++B/ @++B	594	将指定字的 4 位数 BCD 内容递增 1。	○	○	2-165
	双字 BCD 递增	++BL/ @++BL	595	将指定字的 8 位数 BCD 内容递增 1。	○	○	2-165
	BCD 递减	--B/ @--B	596	将指定字的 4 位数 BCD 内容递减 1。	○	○	2-168
	双字 BCD 递减	--BL/ @--BL	597	将指定字的 8 位数 BCD 内容递减 1。	○	○	2-168

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
四则运算指令	无进位带符号二进制加	+/ @+	400	4 位数 (单字) 十六进制数据和 / 或常数相加。	○	○	2-170
	无进位带符号双字二进制加	+L/ @+L	401	8 位数 (双字) 十六进制数据和 / 或常数相加。	○	○	2-170
	有进位带符号二进制加	+C/ @+C	402	4 位数 (单字) 十六进制数据和 / 或常数及进位标志 (CY) 相加。	○	○	2-172
	有进位带符号双字二进制加	+CL/ @+CL	403	8 位数 (双字) 十六进制数据和 / 或常数及进位标志 (CY) 相加。	○	○	2-172
	无进位 BCD 加	+B/ @+B	404	4 位 (单字)BCD 数据和 / 或常数相加。	○	○	2-174
	无进位双字 BCD 加	+BL/ @+BL	405	8 位 (双字)BCD 数据和 / 或常数相加。	○	○	2-174
	有进位 BCD 加	+BC/ @+BC	406	4 位数 (单字)BCD 数据和 / 或常数及进位标志 (CY) 相加。	○	○	2-176
	有进位双字 BCD 加	+BCL/ @+BCL	407	8位(双字)BCD数据和/或常数及进位标志(CY)相加。	○	○	2-176
	无进位带符号二进制减	–/ @–	410	4 位数 (单字) 十六进制数据和 / 或常数相减。	○	○	2-178
	无进位带符号双字二进制减	–L/ @–L	411	8 位数 (双字) 十六进制数据和 / 或常数相减。	○	○	2-178
	有进位带符号二进制减	–C/ @–C	412	4 位数 (单字) 十六进制数据和 / 或常数及进位标志 (CY) 相减。	○	○	2-182
	有进位带符号双字二进制减	–CL/ @–CL	413	8 位数 (双字) 十六进制数据和 / 或常数及进位标志 (CY) 相减。	○	○	2-182
	无进位 BCD 减	–B/ @–B	414	4 位数 (单字)BCD 数据和 / 或常数相减。	○	○	2-185
	无进位双字 BCD 减	–BL/ @–BL	415	8 位数 (双字)BCD 数据和 / 或常数相减。	○	○	2-185
	有进位 BCD 减	–BC/ @–BC	416	4 位数 (单字)BCD 数据和 / 或常数及进位标志 (CY) 相减。	○	○	2-188
	有进位双字 BCD 减	–BCL/ @–BCL	417	8 位数 (双字)BCD 数据和 / 或常数及进位标志 (CY) 相减。	○	○	2-188
	带符号二进制乘	*/ @*	420	4 位数带符号十六进制数据和 / 或常数相乘。	○	○	2-190
	带符号双字二进制乘	*L/ @*L	421	8 位数带符号十六进制数据和 / 或常数相乘。	○	○	2-190
	无符号二进制乘	*U/ @*U	422	4 位数无符号十六进制数据和 / 或常数相乘。	–	○	2-192
	无符号双字二进制乘	*UL/ @*UL	423	8 位数无符号十六进制数据和 / 或常数相乘。	–	○	2-192
	BCD 乘	*B/ @*B	424	4 位数 (单字)BCD 数据和 / 或常数相乘。	○	○	2-194
	双字 BCD 乘	*BL/ @*BL	425	8 位数 (双字)BCD 数据和 / 或常数相乘。	○	○	2-194
	带符号二进制除	/ @/	430	4位数(单字)带符号十六进制数据和/或常数相除。	○	○	2-196
	带符号双字二进制除	/L @/L	431	8位数(双字)带符号十六进制数据和/或常数相除。	○	○	2-196
	无符号二进制除	/U @/U	432	4位数(单字)无符号十六进制数据和/或常数相除。	–	○	2-198
	无符号双字二进制除	/UL @/UL	433	8位数(单字)无符号十六进制数据和/或常数相除。	–	○	2-198
	BCD 除	/B @/B	434	4 位 (单字)BCD 数据和 / 或常数相除。	○	○	2-200
	双字 BCD 除	/BL @/BL	435	8 位数 (双字)BCD 数据和 / 或常数相除。	○	○	2-200

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
转换指令	BCD →二进制	BIN/ @BIN	023	将 BCD 数据转换为二进制数据。	○	○	2-202
	双字 BCD →双字二进制	BINL/ @BINL	058	将 8 位数 BCD 数据转换为 8 位数十六进制 (32 位二进制) 数据。	○	○	2-202
	二进制→ BCD	BCD/ @BCD	024	将一个字的二进制数据转换成一个字的 BCD 数据。	○	○	2-204
	双字二进制→双字 BCD	BCDL/ @BCDL	059	将 8 位数十六进制 (32 位二进制) 数据转换为 8 位数 BCD 数据。	○	○	2-204
	二进制求补	NEG/ @NEG	160	计算一个字的十六进制数据的 2 的补码。	○	○	2-206
	数据译码	MLPX/ @MLPX	076	读取源字中指定位 (或字节) 的数值，将结果字 (或 16 字范围) 中相应的位变为 ON，并将结果字 (或 16 字范围) 中的所有其它位变为 OFF。	○	○	2-208
	数据编码	DMPX/ @DMPX	077	寻找源字范围 (或 16 字范围) 内的第一个或最后一个 ON 位的位置，并将该值写入结果字的指定数位 (或字节) 中。	○	○	2-213
	ASCII 转换	ASC/ @ASC	086	将源字中的 4 位十六进制数位转换成等值的 8 位 ASCII 码。	○	○	2-218
	ASCII →十六进制	HEX/ @HEX	162	将源字中的最多 4 个字节的 ASCII 数据转换成等值的十六进制数位，并将这些数位写入指定的目的字中。	○	○	2-222
逻辑指令	逻辑与	ANDW/ @ANDW	034	对单字数据和 / 或常数中的相应位作逻辑与运算。	○	○	2-227
	双字逻辑与	ANDL/ @ANDL	610	对双字数据和 / 或常数的相应位作逻辑与运算。	○	○	2-227
	逻辑或	ORW/ @ORW	035	对单字数据和 / 或常数的相应位作逻辑或运算。	○	○	2-229
	双字逻辑或	ORWL/ @ORWL	611	对双字数据和 / 或常数的相应位作逻辑或运算。	○	○	2-229
	异或	XORW/ @XORW	036	对单字数据和 / 或常数的相应位作逻辑异或运算。	○	○	2-231
	双字异或	XORL/ @XORL	612	对双字数据和 / 或常数的相应位作逻辑异或运算。	○	○	2-231
	求补	COM/ @COM	029	将 Wd 中的所有 ON 位变 OFF，并将所有 OFF 位变 ON。	○	○	2-233
	双字求补	COML/ @COML	614	将 Wd 和 Wd+1 中的所有 ON 位变 OFF，并将所有 OFF 位变 ON。	○	○	2-233
特殊算术指令	算术处理	APR/ @APR	069	计算源数据的正弦、余弦或线性外插。	○	○	2-235
	位计数器	BCNT/ @BCNT	067	对指定字中的所有 ON 位的总数计数。	○	○	2-244

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
浮点算术运算指令	浮点数→ 16 位	FIX/ @FIX	450	将一个 32 位浮点数据转换成 16 位带符号的二进制数据，并将结果放入指定的结果字中。	○	○	2-250
	浮点数→ 32 位	FIXL/ @FIXL	451	将一个 32 位浮点数据转换成 32 位带符号的二进制数据，并将结果放入指定的结果字中。	○	○	2-250
	16 位→浮点数	FLT/ @FLT	452	将一个 16 位带符号的二进制数据转换成 32 位浮点数据，并将结果放入指定的结果字中。	○	○	2-252
	32 位→浮点数	FLTL/ @FLTL	453	将一个 32 位带符号的二进制数据转换成 32 位浮点数据，并将结果放入指定的结果字中。	○	○	2-252
	浮点数加	+F/ @+F	454	将两个 32 位浮点数相加，并将结果放入指定的结果字中。	○	○	2-254
	浮点数减	–F/ @–F	455	将两个 32 位浮点数相减，并将结果放入指定的结果字中。	○	○	2-254
	浮点数乘	*F/ @*F	456	将两个 32 位浮点数相乘，并将结果放入指定的结果字中。	○	○	2-254
	浮点数除	/F @/F	457	将两个 32 位浮点数相除，并将结果放入指定的结果字中。	○	○	2-254
	浮点数符号比较	=F	329	比较指定的单精度数据 (32 位) 或常数，并在比较结果为真时产生一个 ON 执行条件。浮点数符号比较指令中可使用下述三种符号：LD(载入)、AND 和 OR。	○	○	2-258
		<>F	330				
		<F	331				
		<=F	332				
		>F	333				
		>=F	334				
	浮点数→ ASCII	FSTR/ @FSTR	448	将指定的单精度浮点数据 (32 位十进制小数或指数格式) 转换成文本字符串数据 (ASCII)，并将结果输出到目的字。	○	○	2-261
	ASCII →浮点数	FVAL/ @FVAL	449	将代表单精度浮点数据(十进制小数或指数格式)的文本字符串 (ASCII) 转换成 32 位单精度浮点数据，并将结果输出到目的字。	○	○	2-266
表数据处理指令	交换字节	SWAP/ @SWAP	637	将范围内所有字的最左字节和最右字节交换。	○	○	2-270
	最大值检索	MAX/ @MAX	182	检索指定表内的最大值。	–	○	2-272
	最小值检索	MIN/ @MIN	183	检索指定表内的最小值。	–	○	2-272
	帧校验和	FCS/ @FCS	180	计算指定范围的 ASCII FCS 值。	○	○	2-276
数据控制指令	带自整定功能的 PID 控制	PIDAT	191	根据指定的参数执行 PID 控制。PID 常数可通过 PIDAT(191) 进行自整定。	○	○	2-278
	时间比例输出	TPO	685	从指定字输入占空比或被控变量，根据指定参数将占空比转换成时间比例输出，然后从指定的输出参数输出结果。	○	○	2-290
	定标	SCL/ @SCL	194	根据指定的线性函数，将无符号二进制数据转换成无符号 BCD 数据。	○	○	2-297
	定标 2	SCL2/ @SCL2	486	根据指定的线性函数，将带符号二进制数据转换成带符号BCD数据。定义线性函数时可输入一个偏移值。	○	○	2-301
	定标 3	SCL3/ @SCL3	487	根据指定的线性函数，将带符号 BCD 数据转换成带符号二进制数据。定义线性函数时可输入一个偏移值。	○	○	2-305
	平均值	AVG	195	计算一个输入字在指定数目的循环中的平均值。	○	○	2-308
子程序指令	子程序调用	SBS/ @SBS	091	调用指定子程序号的子程序并执行该程序。	○	○	2-311
	子程序入口	SBN	092	表示指定子程序号的子程序的开始。	○	○	2-316
	子程序返回	RET	093	表示子程序的结束。	○	○	2-316

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
中断控制指令	设置中断屏蔽	MSKS/ @MSKS	690	为 I/O 中断或定时中断设置中断处理。	○	○	2-321
	清除中断	CLI/ @CLI	691	清除或保留 I/O 中断的已记录中断输入，或为定时中断设定首次定时中断时间。	○	○	2-324
	禁止中断	DI/ @DI	693	禁止执行所有中断任务 (除电源 OFF 中断除外)。	○	○	2-327
	允许中断	EI	694	允许执行被 DI(693) 指令所禁止的所有中断任务。	○	○	2-328
高速计数器和脉冲输出指令	模式控制	INI/ @INI	880	INI(880) 指令用于启动和停止目标值比较，改变高速计数器的当前值 (PV)，改变中断输入的 PV 值 (计数器模式)，改变脉冲输出的 PV 值或停止脉冲输出。	○	○	2-329
	读高速计数器的 PV 值	PRV/ @PRV	881	PRV(881) 指令用于读取高速计数器、脉冲输出或中断输入 (计数器模式) 的当前值 (PV)。	○	○	2-332
	比较表载入	CTBL/ @CTBL	882	CTBL(882) 指令用于对高速计数器的当前值 (PV) 执行目标值或范围比较。	○	○	2-336
	速度输出	SPED/ @SPED	885	SPED(885) 指令用于指定频率并执行无加速或减速的脉冲输出。	○	○	2-340
	设置脉冲	PULS/ @PULS	886	PULS(886) 用于设置脉冲输出的脉冲数。	○	○	2-344
	脉冲输出	PLS2/ @PLS2	887	PLS2(887) 用于设置脉冲频率和加速度 / 减速度，并用加速度 / 减速度 (用不同的减速度 / 减速度) 执行脉冲输出。仅定位控制允许进行。	○	○	2-346
	加速度控制	ACC/ @ACC	888	ACC(888) 用于设置脉冲频率和加速度 / 减速度，并用加速度 / 减速度 (用相同的减速度 / 减速度) 执行脉冲输出。定位和速度控制均允许进行。	○	○	2-352
	原点搜索	ORG/ @ORG	889	ORG(889) 用于执行原点搜索和返回。	○	○	2-357
	占空比可变脉冲	PWM/ @PWM	891	PWM(891) 用于输出占空比可变的脉冲。	○	○	2-360
	中断标准移动	IFEED/ @IFEED	892	加速后的等速控制期间，发生输入中断时，按指定脉冲量移动后减速停止。 无需创建中断任务。	-	○	2-362
	直线插补	ITPL/ @ITPL	893	使用指定的脉冲输出端口执行2 ~ 4轴的直线插补。	-	○	2-366
步指令	步启动	SNXT	009	SNXT(009) 指令用于以下 3 种情况： (1) 开始步程序执行。 (2) 进到下一个步控制位。 (3) 结束步程序执行。	○	○	2-372
	步定义	STEP	008	STEP(008) 指令视指令的位置和是否指定了控制位而定，有以下 2 个作用： (1) 开始一个指定的步。 (2) 结束步程序区 (例如步执行)。	○	○	2-372
基本I/O单元指令	I/O 刷新	IORF/ @IORF	097	刷新指定的 I/O 字。	○	○	2-382
	7 段译码	SDEC/ @SDEC	078	将指定数位的十六进制内容转换成 8 位、7 段显示代码，并将其放入指定目的字的高8位或低8位中。	○	○	2-384
	数字开关输入	DSW	210	读取设置在与输入单元或输出单元相连的外部数字开关 (或指轮开关) 上的值，并将 4 位数或 8 位数 BCD 数据储存到指定的字中。	○	○	2-387
	矩阵输入	MTR	213	从与输入单元和输出单元 (使用 8 点输入和 8 点输出) 相连的 8 × 8 矩阵输入最多 64 个信号，并将该 64 位数据储存到 4 个目的字中。	○	○	2-391
	7 段显示输出	7SEG	214	将源数据 (4 位数或 8 位数 BCD) 转换成 7 段显示数据，并将该数据输出至指定的输出字中。	○	○	2-395
串行通信指令	发送	TXD/ @TXD	236	从内置到 CPU 单元的 RS-232C 端口或串行通信选件板 (1.2 版或更高版本) 的串行端口输出指定字节数的数据。	○	○	2-399
	接收	RXD/ @RXD	235	从内置到 CPU 单元的 RS-232C 端口或串行通信选件板 (1.2 版或更高版本) 的串行端口读取指定字节数的数据。	○	○	2-404

指令类型	指令	助记符	功能编号	功能	CP1E	CP2E	页码
网络通信用指令	网络发送	SEND/ @SEND	090	对网络上的节点发送数据。	–	○	2-424
	网络接收	RECV/ @RECV	098	对网络上的节点请求发送，接收数据。	–	○	2-428
	指令发送	CMND/ @CMND	490	发行任意的 FINS 指令，接收响应。	–	○	2-431
时钟指令	日历加	CADD/ @CADD	730	将指定字中的日历数据和时间相加。	○	○	2-435
	日历减	CSUB/ @CSUB	731	从指定字中的日历数据减去时间。	○	○	2-435
	时钟调整	DATE/ @DATE	735	将内部时钟设定改为指定源字中的设定。	○	○	2-440
故障诊断指令	故障报警	FAL/ @FAL	006	产生或清除用户自定义的非致命错误。	○	○	2-442
	严重故障报警	FALS	007	产生用户自定义的致命错误。	○	○	2-449
其它指令	置进位	STC/ @STC	040	置进位标志 (CY)。	○	○	2-455
	清除进位	CLC/ @CLC	041	使进位标志 (CY) 变为 OFF。	○	○	2-455
	延长最大循环时间	WDT/ @WDT	094	延长最大循环时间，但仅对于执行该指令的循环有效。	○	○	2-456

2

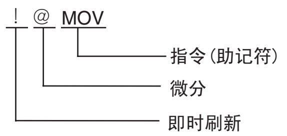
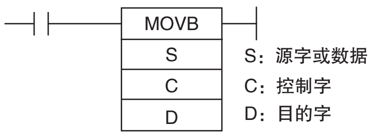
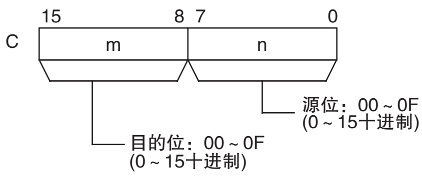
指令

本章节介绍了 CP1E/CP2E CPU 单元支持的功能、操作数和指令程序示例。

指令符号编排说明

指令按功能分组进行描述。有关按助记符编排的指令列表 (列出了各条指令在本章中的页码)，请参阅 “附录 A 按功能代码编排的指令列表”。

各条指令的描述按下表的形式组织编排。

<table>
<tr><th>项目</th><th>内容</th></tr>
<tr><td>指令</td><td>表示指令的名称。例如：位传送 (MOVE BIT)</td></tr>
<tr><td>助记符</td><td>表示助记符。
例如：MOVB(082)</td></tr>
<tr><td>变化</td><td>微分
@ 当执行条件变为 ON 时，执行微分操作的指令。
% 当执行条件变为 OFF 时，执行微分操作的指令。
即时刷新
! 执行指令时，刷新由操作数或高功能 I/O 单元中的字指定的 I/O 区中的数据。
</td></tr>
<tr><td>功能代码</td><td>表示功能代码。</td></tr>
<tr><td>功能</td><td>在章节的标题后描述了指令的基本用途。</td></tr>
<tr><td>符号</td><td>下面以 MOVE BIT 指令为例，说明了用于代表 CX-Programmer 中的指令的梯形图符号。各操作数的名称也用梯形图符号表示。
</td></tr>
<tr><td>适用程序区</td><td>指定了可使用指令的程序区。“OK”表示指令可在该程序区中使用。
<table>
<tr><th>区域</th><th>功能块定义内</th><th>步程序区</th><th>子程序区</th><th>中断任务程序区</th></tr>
<tr><td>能否使用</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td></tr>
</table></td></tr>
<tr><td>操作数</td><td>描述操作数的数据类型和大小。
必要时，还将描述特定操作数中所使用的字和位的含义 (如控制字)。
</td></tr>
<tr><td>区域类型</td><td>每个操作数可使用的存储区地址用下表形式列出。顶行的列标题中使用的字母与梯形图符号中使用的字母相同。当某个区无法指定为供某个操作数使用时，用 “---”表示。
<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">间接DM地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR直接</th><th>IR间接</th></tr>
<tr><td>S</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="2">OK</td><td rowspan="3">OK</td><td rowspan="3">---</td><td rowspan="3">OK</td><td rowspan="3">---</td><td rowspan="3">---</td><td rowspan="3">---</td></tr>
<tr><td>C</td></tr>
<tr><td>D</td><td>---</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>项目</th><th>内容</th></tr>
<tr><td>条件标志</td><td>标志表表示刚执行完指令之后的条件标志的状态。任何未列出的标志均不受指令的影响。“OFF”表示不论指令的执行结果如何，在刚完执行该指令之后，标志均变为 “OFF”。
<table>
<tr><th>名称</th><th>标记</th><th>操作</th></tr>
<tr><td>出错标志</td><td>ER</td><td>OFF</td></tr>
<tr><td>等于标志</td><td>=</td><td>・如果正在传送的数据(D)为0，则为ON。
・在所有其它情况下为OFF。</td></tr>
<tr><td>负标志</td><td>N</td><td>・如果正在传送的数据(D)的最左位为1，则为ON。
・在所有其它情况下为OFF。</td></tr>
</table></td></tr>
<tr><td>功能</td><td>表示指令的功能。</td></tr>
<tr><td>提示</td><td>表示除主要功能之外的补充说明。</td></tr>
<tr><td>注意事项</td><td>表示使用指令时的要点。</td></tr>
<tr><td>程序举例</td><td>给出了一个使用某指令和特定操作数的程序范例，用于进一步解释该指令的功能。</td></tr>
</table>

常数

作为操作数输入的常数如下所示。

操作数描述及区域类型

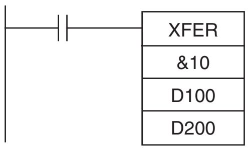
・ 指定位串的操作数 (通常作为十六进制输入)：
指定位串的操作数仅提供十六进制的表示形式，例如只能为 MOV(021) 指令的 S 操作数指定“#0000 ~ #FFFF”。但是在 CX-Programmer 中，可通过加前缀 & 来以十进制的形式输入位串。

・ 指定数值的操作数 (通常作为十进制输入，包括跳转号)：
指定数值的操作数能以十进制和十六进制这两种形式来表示，例如可以为 XFER(070) 指令的 N 操作数指定 “#0000 ~ #FFFF”和 “&0 ~ &65535”。

・ 表示控制号的操作数 (跳转号除外)：
为控制号提供了十进制的表示形式，例如可为 SBS(091) 指令的 N 操作数指定 “0 ~ 127”。

例

如下例所示，通过 CX-Programmer 的记号给定一个常数。例如通过加前缀 & 来以十进制的形式给定一个用于指定数值的操作数的值。



在编程设备中输入常数的方法如下表所示。

<table>
<tr><th>操作数</th><th>CX-Programmer</th></tr>
<tr><td>指定位串的操作数 (通常作为十六进制输入)</td><td rowspan="2">通过加前缀 & 以十进制输入或者通过加前缀 # 以十六进制输入。(见注)</td></tr>
<tr><td>指定数值的操作数 (通常作为十进制输入)</td></tr>
<tr><td>指定控制号的操作数 (跳转号除外)</td><td>带前缀 # 作为十进制输入。(见注)</td></tr>
</table>

注 当在 CX-Programmer 中输入操作数时，输入范围将连同适当的前缀一起显示。

条件标志

使用 CX-Programmer 时，应事先将条件标志作为全局符号注册，符号名之前应加 “P_”。

标志	CX-Programmer 标记
出错标志	P_ER
访问出错标志	P_AER
进位标志	P_CY
大于标志	P_GT
等于标志	P_EQ
小于标志	P_LT
负标志	P_N
上溢标志	P_OF
下溢标志	P_UF
大于等于标志	P_GE
不等于标志	P_NE
小于等于标志	P_LE
常 ON 标志	P_On
常 OFF 标志	P_Off

符号指令

某些 C/CV 系列 PLC 的指令已改为功能与 CP1E/CP2E 系列 PLC 的指令相同但名称不同的指令。

指令分组	C/CV 系列	CP1E/CP2E 系列
比较	EQU	AND=
数据传送	MOVQ	MOV
递增 / 递减	INC	++B
	INCL	++BL
	INCB	++
	INBL	++L
	DEC	--B
	DECL	--BL
	DECB	--
	DCBL	--L
四则运算	ADB	+C
	ADBL	+CL
	ADD	+BC
	ADDL	+BCL
	SBB	-C
	SBBL	-CL
	SUB	-BC
	SUBL	-BCL
	MBS	*
	MBSL	*L
	MLB	*U
	MUL	*B
	MULL	*BL
	DBS	/
	DBSL	/L
	DVB	/U
	DIV	/B
	DIVL	/BL
中断控制	INT	MSKS / CLIDI / EI

顺序输入指令

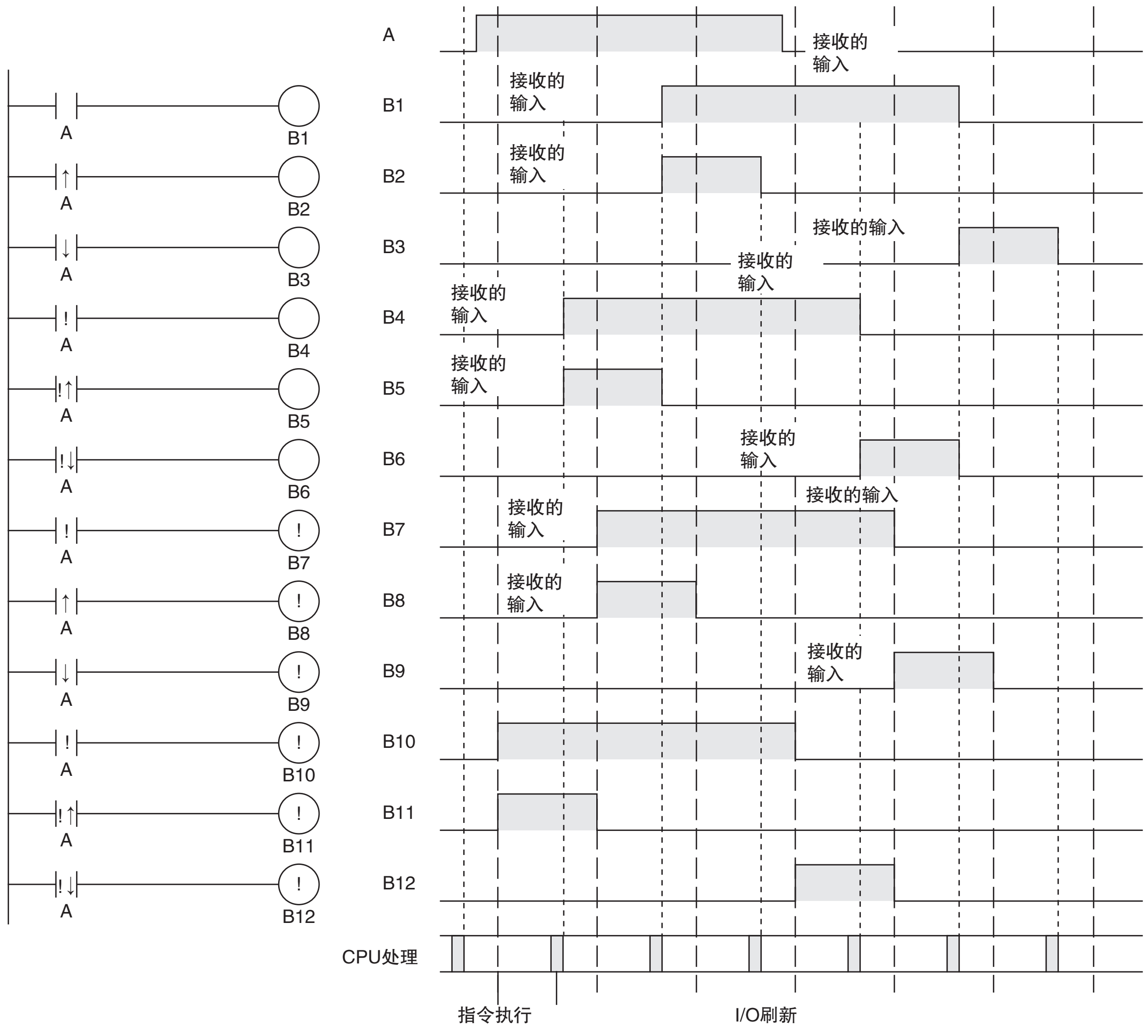
微分和即时刷新指令

- 载入、逻辑与和逻辑或指令除了普通形式之外，还有微分和即时刷新变化，而且另外还有两种组合情况。
- 载入非、与非、或非、输出和输出非指令除了普通形式之外，还有即时刷新变化。
- 普通指令、微分指令、即时刷新指令和即时刷新微分指令对数据的 I/O 定时的处理方式有所不同。
- 普通指令和微分指令使用由前一个 I/O 刷新处理所输入的数据来执行，并通过下一个 I/O 处理来输出结果。此处的 “I/O 刷新”是指 CPU 的内部存储器和 I/O 单元之间的数据交换。
- 除了上述 I/O 刷新之外，即时刷新指令与 I/O 单元就由该指令访问的字进行数据交换。除了指定位以外，即时刷新指令还可同时刷新 16 位数据。
  即时刷新指令 (即带 ! 的指令) 不可用于 CP 扩展单元或 CP 扩展 I/O 单元的 I/O 中。对于 CP 扩展单元或 CP 扩展 I/O 单元，请使用 IORF(097)。

指令变化	助记符	功能	I/O 刷新
普通形式	LD，AND，OR，LD NOT，AND NOT，OR NOT	指定位的 ON/OFF 状态由 CPU 通过循环刷新获取，并反映到下一个指令执行中。	循环刷新
	OUT，OUT NOT	执行指令后，指定位的 ON/OFF 状态通过下一个循环刷新来输出。	
上升沿微分	@LD，@AND，@OR	当指定位从 OFF 变为 ON 时执行一次指令，且 ON 状态保持一个循环。	
下降沿微分	%LD，%AND，%OR	当指定位从 ON 变为 OFF 时执行一次指令，且 ON 状态保持一个循环。	
即时刷新	!LD，!AND，!OR，!LD NOT，!AND NOT，!OR NOT	指定位的输入数据由 CPU 获取并执行指令。	执行指令之前
	!OUT，!OUT NOT	执行指令之后，输出指定位的数据。	执行指令之后
上升沿微分 / 即时刷新	!@LD，!@AND，!@OR	指定位的输入数据由 CPU 进行刷新，且在该位从 OFF 变为 ON 时执行该指令一次，并使 ON 状态保持一个循环。	执行指令之前
下降沿微分 / 即时刷新	!%LD，!%AND，!%OR	指定位的输入数据由 CPU 进行刷新，且在该位从 ON 变为 OFF 时执行该指令一次，并使 ON 状态保持一个循环。	

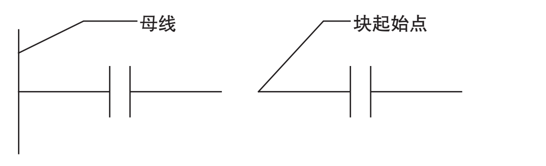
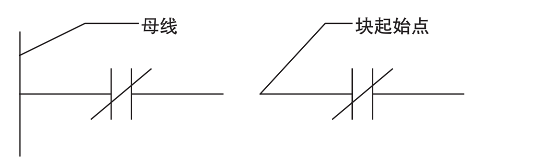
I/O 指令的工作时序

下图给出了从 LD 和 OUT 指令配置的程序中指令工作的时序差异。



LD/LD NOT

指令	助记符	变化	功能代码	功能
载入	LD	@LD，%LD，!LD，!@LD，!%LD	---	指定一个逻辑开始，并根据指定操作位的 ON/OFF 状态建立一个 ON/OFF 执行条件。
载入非	LD NOT	@LD NOT，%LD NOT，! LD NOT，!@LD NOT，!%LD NOT	---	指定一个逻辑开始，并根据指定操作位的 ON/OFF 状态的取反结果建立一个 ON/OFF 执行条件。

	LD	LD NOT
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
---	---	BOOL	---

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
LD	OK	OK	OK	OK	OK	OK	---	---	---	---	---	---	OK	OK	OK	OK
LD NOT																---

条件标志

该指令不影响任何标志。

功能

● LD

LD 指令用于从母线开始的第一个常开位或从逻辑块开始的第一个常开位。如果没有即时刷新规定，则读取 I/O 存储器中的指定位。如果有即时刷新规定，则读取和使用 CPU 单元的内置输入端子的状态。

● LD NOT

LD NOT 指令用于从母线开始的第一个常闭位或从逻辑块开始的第一个常闭位。如果没有即时刷新规定，则读取 I/O 存储器中的指定位并取反。如果有即时刷新规定，则读取 CPU 单元的内置输入端子的状态，并在取反后使用。

提示

- LD/LD NOT 指令在下述情况下用作表示逻辑开始的指令。

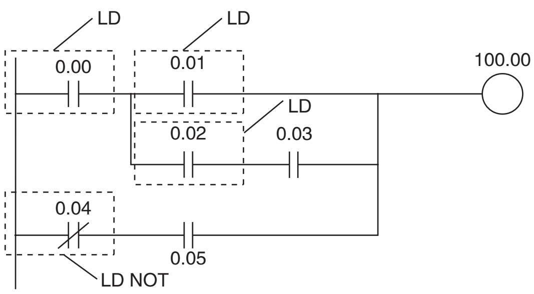
  1. 当直接连接至母线时。
  2. 当逻辑块通过 AND LD 或 OR LD 相连时 (即在逻辑块的起始处)。
     逻辑块与和逻辑块或指令用于以串联或并联方式连接以 LD 或 LD NOT 开始的逻辑块。

- 当与输出相关的指令无法直接连接到母线时，至少需要一条载入或载入非指令用于执行条件。 如果没有载入或载入非指令，则将通过外围设备的程序检查功能产生一个编程错误。
- 当逻辑块通过逻辑块与或逻辑块或指令连接时，逻辑块与 / 逻辑块或指令的总数必须等于载入 / 载入非指令的总数减 1。否则将产生编程错误。有关详情，请参阅 “逻辑块与：AND LD 和逻辑块或：OR LD”。

注意事项

- 可为 LD 指定上升沿微分 (@) 或下降沿微分 (%)。如果指定上升沿微分 (@)，则仅在操作位的状态从 OFF 变为 ON 后，执行条件在一个循环中变为 ON。如果指定下降沿微分 (%)，则仅在操作位的状态从 ON 变为 OFF 后，执行条件在一个循环中变为 ON。
- 可为 LD/LD NOT 指定即时刷新 (!)。即时刷新指令在即将从 CPU 单元执行指令之前更新内置输入位的状态。
- 对于 LD，允许即时刷新和上升沿或下降沿微分 (!@ 或 !%) 的组合。如果指定二者之一，则在即将执行指令之前从基本输入单元对输入进行刷新，且仅当状态从 OFF 变为 ON 或者从 ON 变为 OFF 之后，执行条件在一个循环中变为 ON。

程序举例

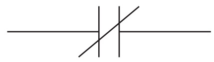


指令	操作数
LD	0.00
LD	0.01
LD	0.02
AND	0.03
OR LD	––
AND LD	––
LD NOT	0.04
AND	0.05
OR LD	––
OUT	100.00

AND LD
OR LD
OR LD

AND/AND NOT

指令	助记符	变化	功能代码	功能
与	AND	@AND， %AND，!AND， !@AND， !%AND	---	将指定操作位的状态和当前执行条件进行逻辑与操作。
与非	AND NOT	@AND NOT， %AND NOT， !AND NOT， !@AND NOT， !%AND NOT	---	将指定操作位的状态取反后和当前执行条件进行逻辑与操作。

符号	AND	AND NOT
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
---	---	BOOL	---

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
AND AND NOT	OK	OK	OK	OK	OK	OK	---	---	---	---	---	---	OK	OK	OK	---

条件标志

该指令不影响任何标志。

功能

● AND

AND 用于常开位的串联连接。AND 无法直接连接到母线，也无法用于逻辑块的起始处。如果没有即时刷新规定，则读取 I/O 存储器中的指定位。如果有即时刷新规定，则读取 CPU 单元的内置输入端子的状态。

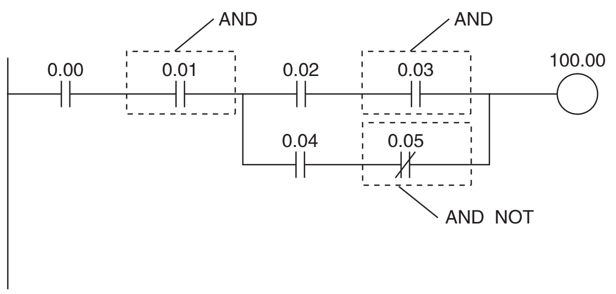
● AND NOT

AND NOT 用于常闭位的串联连接。AND NOT 无法直接连接到母线，也无法用于逻辑块的起始处。如果没有即时刷新规定，则读取 I/O 存储器中的指定位。如果有即时刷新规定，则读取 CPU 单元的内置输入端子的状态。

注意事项

- 可为 AND 指定上升沿微分 (@) 或下降沿微分 (%)。如果指定上升沿微分 (@)，则仅在操作位的状态从 OFF 变为 ON 后，执行条件在一个循环中变为 ON。如果指定下降沿微分 (%)，则仅在操作位的状态从 ON 变为 OFF 后，执行条件在一个循环中变为 ON。
- 可为 AND/AND NOT 指定即时刷新 (!)。即时刷新指令在即将从 CPU 单元执行指令之前更新内置输入位的状态。
- 对于 AND，允许即时刷新和上升沿或下降沿微分 (!@ 或 !%) 的组合。如果指定二者之一，则在即将执行指令之前从基本输入单元对输入进行刷新，且仅当状态从 OFF 变为 ON 或者从 ON 变为 OFF 之后，执行条件在一个循环中变为 ON。

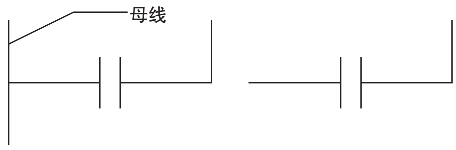
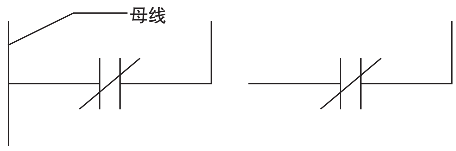
程序举例



指令	操作数
LD	0.00
AND	0.01
LD	0.02
AND	0.03
LD	0.04
AND NOT	0.05
OR LD	--
AND LD	--
OUT	100.00

OR/OR NOT

指令	助记符	变化	功能代码	功能
或	OR	@OR，%OR，!OR，!@OR，!%OR	---	将指定操作位的 ON/OFF 状态和当前执行条件进行逻辑或操作。
或非	OR NOT	@OR NOT，%OR NOT，!OR NOT，!@OR NOT，!%OR NOT	---	将指定位的状态取反后和当前执行条件进行逻辑或操作。

	OR	OR NOT
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
---	---	BOOL	---

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
OR OR NOT	OK	OK	OK	OK	OK	OK	---	---	---	---	---	---	OK	OK	OK	---

条件标志

该指令不影响任何标志。

功能

● OR

OR 用于常开位的并联连接。一个常开位和一个以载入或载入非指令开始的逻辑块 (连接到母线或位于逻辑块的起始处) 形成一个逻辑或。如果没有即时刷新规定，则读取 I/O 存储器中的指定位。如果有即时刷新规定，则读取 CPU 单元的内置输入端子的状态。

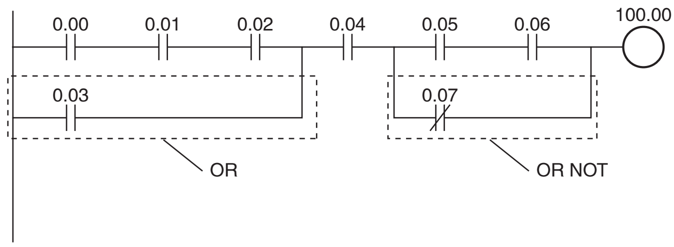
● OR NOT

OR NOT 用于常闭位的并联连接。一个常闭位和一个以载入或载入非指令开始的逻辑块 (连接到母线或位于逻辑块的起始处) 形成一个逻辑或。如果没有即时刷新规定，则读取 I/O 存储器中的指定位。如果有即时刷新规定，则读取 CPU 单元的内置输入端子的状态。

注意事项

- 可为 OR 指定上升沿微分 (@) 或下降沿微分 (%)。如果指定上升沿微分 (@)，则仅在操作位的状态从 OFF 变为 ON 后，执行条件在一个循环中变为 ON。如果指定下降沿微分 (%)，则仅在操作位的状态从 ON 变为 OFF 后，执行条件在一个循环中变为 ON。
- 可为 OR/OR NOT 指定即时刷新 (!)。即时刷新指令在即将从 CPU 单元执行指令之前更新内置输入位的状态。
- 对于 OR，允许即时刷新和上升沿或下降沿微分的组合 (!@ 或 !%)。如果指定二者之一，则在即将执行指令之前从基本输入单元对输入进行刷新，且仅当操作位的状态从 OFF 变为 ON 或者从 ON 变为 OFF 之后，执行条件在一个循环中变为 ON。

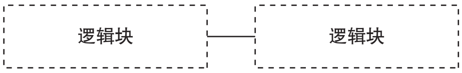
程序举例



指令	操作数
LD	0.00
AND	0.01
AND	0.02
OR	0.03
AND	0.04
LD	0.05
AND	0.06
OR NOT	0.07
AND LD	--
OUT	100.00

AND LD/OR LD

指令	助记符	变化	功能代码	功能
逻辑块与	AND LD	---	---	在逻辑块之间进行逻辑与。
逻辑块或	OR LD	---	---	在逻辑块之间进行逻辑或。

	AND LD	OR LD
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

条件标志

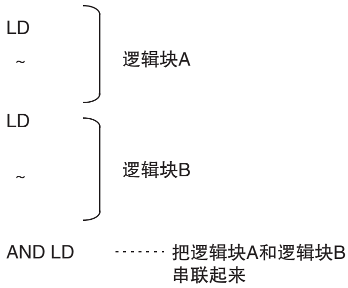
该指令不影响任何标志。

功能

● AND LD

AND LD 将紧邻该指令之前的两个逻辑块串联。

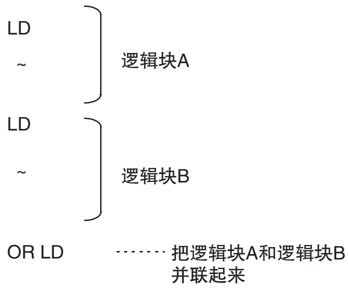
逻辑块由从载入或载入非指令开始到同一梯级中下一个载入或载入非指令为止的所有指令组成。



● OR LD

OR LD 将紧邻该指令之前的两个逻辑块并联。

逻辑块由从载入或载入非指令开始到同一梯级中下一个载入或载入非指令为止的所有指令组成。



提示

● AND LD

・使用该指令可以串联三个或三个以上的逻辑块。方法是首先连接两个逻辑块，然后按顺序连接后续的逻辑块。在三个或三个以上的逻辑块之后还能继续使用该指令进行串联。

● OR LD

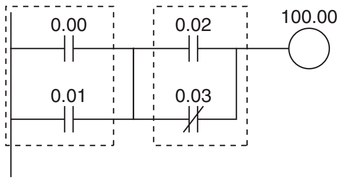
・使用该指令可以并联三个或三个以上的逻辑块。方法是首先连接两个逻辑块，然后按顺序连接后续的逻辑块。在三个或三个以上的逻辑块之后还能继续使用该指令进行并联。

注意事项

当逻辑块通过逻辑块与或逻辑块或指令连接时，逻辑块与 / 逻辑块或指令的总数必须等于载入 / 载入非指令的总数减 1，否则将产生编程错误。

● AND LD

在下述程序中，两个虚线框内即为两个逻辑块。研究该示例后可以发现，当左逻辑块中的任一执行条件为 ON(即当 CIO 0.00 或 CIO 0.01 为 ON 时) 且右逻辑块中的任一执行条件为 ON(即当 CIO 0.02 为 ON 或 CIO 0.03 为 OFF 时) 时，将产生一个 ON 执行条件。



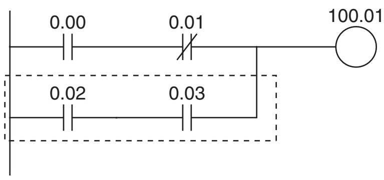
编程

指令	操作数
LD	0.00
OR	0.01
LD	0.02
OR NOT	0.03
AND LD	---
OUT	100.00

第二个 LD 指令：在与前一个块串联的下一个块的起始位处使用。

● OR LD

下图中，在上、下逻辑块之间需要一个逻辑块或指令。当 CIO 0.00 为 ON 且 CIO 0.01 为 OFF，或者当 CIO 0.02 和 CIO 0.03 均为 ON 时，将产生一个 ON 执行条件。除了将当前执行条件与上次未使用的执行条件进行逻辑或操作之外，逻辑块或指令的助记符操作与逻辑块与指令完全相同。



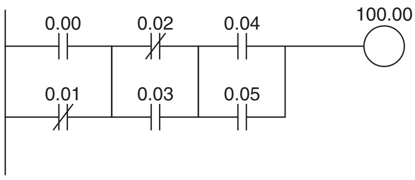
编程

指令	操作数
LD	0.00
AND NOT	0.01
LD	0.02
AND	0.03
OR LD	---
OUT	100.01

第二个 LD 指令：在与前一个块串联的下一个块的起始位处使用。

程序举例

● AND LD



编程例 (1)

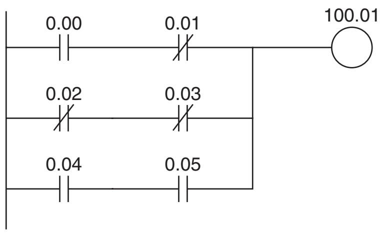
指令	操作数
LD	0.00
OR NOT	0.01
LD NOT	0.02
OR	0.03
AND LD	---
LD	0.04
OR	0.05
AND LD	---
. .	. .
OUT	100.00

编程例 (2)

指令	操作数
LD	0.00
OR NOT	0.01
LD NOT	0.02
OR	0.03
LD	0.04
OR	0.05
. .	. .
AND LD	---
AND LD	---
. .	. .
OUT	100.00

- 逻辑块与指令可重复使用。但用上述方法 (2) 编程时，逻辑块与指令的数目将比其前面的载入和载入非指令的数目少 1。
- 用方法 (2) 编程时，请务必确保逻辑块与指令之前的载入和载入非指令的总数不超过 8 个。
- 若要使用 9 个或 9 个以上，则请使用方法 (1) 进行编程。
- 如果用方法 (2) 编程时该指令有 9 个或 9 个以上，则通过外围设备进行程序检查时将产生一个编程错误。

● OR LD



编程例 (1)

指令	操作数
LD	0.00
AND NOT	0.01
LD NOT	0.02
AND NOT	0.03
OR LD	---
LD	0.04
AND	0.05
OR LD	---
. .	. .
OUT	100.01

编程例 (2)

指令	操作数
LD	0.00
AND NOT	0.01
LD NOT	0.02
AND NOT	0.03
LD	0.04
AND	0.05
. .	. .
OR LD	---
OR LD	---
. .	. .
OUT	100.01

- 逻辑块或指令可重复使用。但用上述方法 (2) 编程时，逻辑块或指令的数目将比其前面的载入和载入非指令的数目少 1。
- 用方法 (2) 编程时，请务必确保逻辑块与指令之前的载入和载入非指令的总数不超过 8 个。
- 若要使用 9 个或 9 个以上，则请使用方法 (1) 进行编程。
- 如果用方法 (2) 编程时该指令有 9 个或 9 个以上，则通过外围设备进行程序检查时将产生一个编程错误。

NOT

指令	助记符	变化	功能代码	功能
非	NOT	---	520	执行条件取反。

符号	NOT
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

条件标志

NOT(520) 指令不影响任何标志。

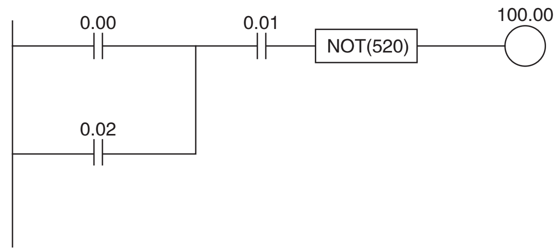
功能

NOT(520) 指令放在一个执行条件与另一个指令之间，用于对执行条件取反。

注意事项

NOT(520) 为中间指令，即该指令无法用作右侧指令。请务必在 NOT(520) 之后编入一个右侧指令。

程序举例



下例中，NOT(520)对执行条件取反

0.00	0.01	0.02	100.00
1	1	1	0
1	1	0	0
1	0	1	1
0	1	1	0
1	0	0	1
0	1	0	1
0	0	1	1
0	0	0	1

UP/DOWN

指令	助记符	变化	功能代码	功能
条件 ON	UP	---	521	当接收到的执行条件从 OFF → ON 时，UP(521) 将下一条指令的执行条件在一个循环中变为 ON。
条件 OFF	DOWN	---	522	当接收到的执行条件从 ON → OFF 时，DOWN(522) 将下一条指令的执行条件在一个循环中变为 ON。

符号	UP	DOWN
	UP(521)	DOWN(522)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

条件标志

UP(521) 和 DOWN(522) 指令不影响任何标志。

功能

● UP

UP(521) 指令放在一个执行条件与另一个指令之间，用于将执行条件变为上升沿微分条件。当执行条件从 OFF → ON 时，UP(521) 指令使连接指令只执行一次。

● DOWN

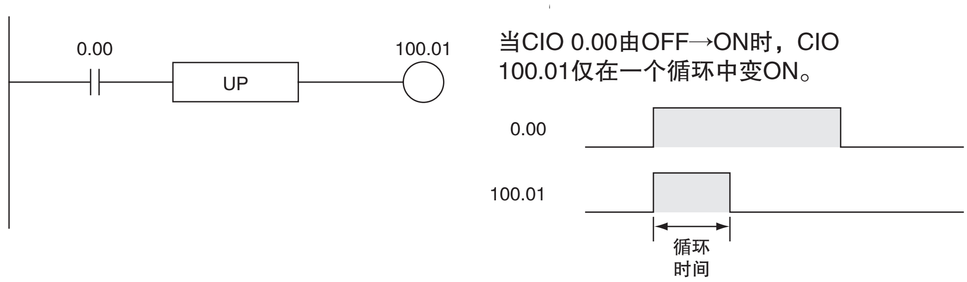
DOWN(522) 指令放在一个执行条件与另一个指令之间，用于将执行条件变为下降沿微分条件。当执行条件从 ON → OFF 时，DOWN(522) 指令使连接指令只执行一次。

注意事项

- UP(521) 和 DOWN(522) 指令的运行不仅取决于指令的执行条件，还取决于当指令编入互锁程序段、跳转程序段或子程序时的程序段执行条件。
- 如果同一个子程序在同一个循环中执行一次以上，则 UP(521) 和 DOWN(522) 的运行将不一致。
- 当子程序的输入条件为 OFF 时，子程序将不执行。在功能块定义中使用 UP(521) 和 DOWN(522) 时，应特别注意。详情请参考 SBS(091) 的相关说明。

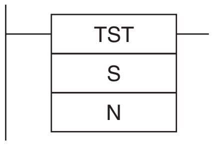
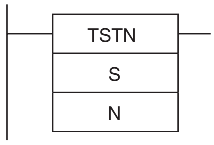
程序举例

● UP



LD TST/LD TSTN

指令	助记符	变化	功能代码	功能
LD 型・位测试	LD TST	–	350	指定位为 1 时，在下段进行 LD 连接。
LD 型・非位测试	LD TSTN	–	351	指定位为 0 时，在下段进行 LD 连接。

	LD TST	LD TSTN
符号	 TST / S / N S：测试数据CH编号 N：位位置数据	 TSTN / S / N S：测试数据CH编号 N：位位置数据

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	测试数据 CH 编号	WORD	1
N	位位置数据	UINT	1

N：位位置数据
　10 进制 &0 ～ 15 或 16 进制 #0000 ～ 000F
　指示通道时，如果为范围外的值，则仅将低位 4 位 （16 进制 #0 ～ F）设为有效。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

● LD TST

将 S 的 N 位的 ON/OFF 内容反映到输入条件中。（使用方法与 LD 指令相同，接在其后对其他指令进行编程。）

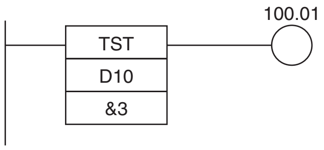
● LD TSTN

对 S 的 N 位的 ON/OFF 内容取反并反映到输入条件中。（使用方法与 LD 指令相同，接在其后对其他指令进行编程。）

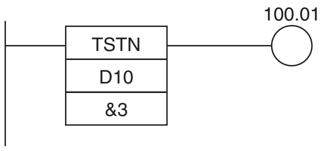
注意事项

- 请在本指令的末尾段附加输出类指令（OUT 类指令与下段连接型指令以外的应用指令）。
- 本指令无法用于电路的末尾段。
- 可在 CP2E CPU 单元中使用 LD TST/LD TSTN 指令。不能在 CP1E CPU 单元中使用。

程序举例



检查D10的位3为ON还是OFF，
如果为ON，则100.01变为ON。



检查D10的位3为ON还是OFF，
如果为OFF，则100.01变为ON。

AND TST/AND TSTN

指令	助记符	变化	功能代码	功能
AND 型・位测试	AND TST	–	350	指定位为 1 时，在下段进行 AND 连接。
AND 型・非位测试	AND TSTN	–	351	指定位为 0 时，在下段进行 AND 连接。

	AND TST	AND TSTN
符号	AND TST S N S：测试数据CH编号 N：位位置数据	AND TSTN S N S：测试数据CH编号 N：位位置数据

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	测试数据 CH 编号	WORD	1
N	位位置数据	UINT	1

N：位位置数据
10 进制 &0 ～ 15 或 16 进制 #0000 ～ 000F
指定通道时，如果为范围外的值，则仅将低位 4 位 （16 进制 #0 ～ F）设为有效。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

● AND TST

将 S 的 N 位的 ON/OFF 内容与输入条件的逻辑积反映到输入条件中。（使用方法与 AND 指令相同，接在其后对其他指令进行编程。）

● AND TSTN

对 S 的 N 位的 ON/OFF 内容取反并将与输入条件的逻辑积反映到输入条件中。（使用方法与 AND 指令相同，接在其后对其他指令进行编程。）

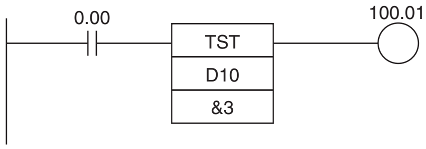
注意事项

・ 请在本指令的末尾段附加输出类指令（OUT 类指令与下段连接型指令以外的应用指令）。

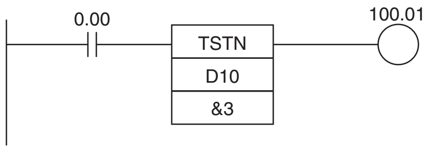
・ 本指令无法用于电路的末尾段。

・ 可在 CP2E CPU 单元中使用 AND TST/AND TSTN 指令。不能在 CP1E CPU 单元中使用。

程序举例



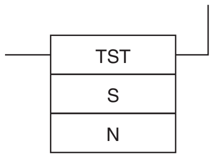
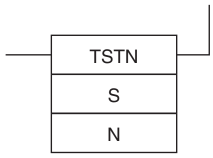
0.00为ON时，检查D10的位3为ON还是OFF。
如果检查后位3为ON，则100.01变为ON。



0.00为ON时，检查D10的位3为ON还是OFF。
如果检查后位3为OFF，则100.01变为ON。

OR TST/OR TSTN

指令	助记符	变化	功能代码	功能
OR 型・位测试	OR TST	–	350	指定位为 1 时，在下段进行 OR 连接。
OR 型・非位测试	OR TSTN	–	351	指定位为 0 时，在下段进行 OR 连接。

	OR TST	OR TSTN
符号	 S：测试数据CH编号 N：位位置数据	 S：测试数据CH编号 N：位位置数据

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	测试数据 CH 编号	WORD	1
N	位位置数据	UINT	1

N：位位置数据

10 进制 &0 ～ 15 或 16 进制 #0000 ～ 000F

指定通道时，如果为范围外的值，则仅将低位 4 位 （16 进制 #0 ～ F）设为有效。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

● OR TST

将 S 的 N 位的 ON/OFF 内容与输入条件的逻辑和反映到输入条件中。（使用方法与 OR 指令相同，接在其后对其他指令进行编程。）

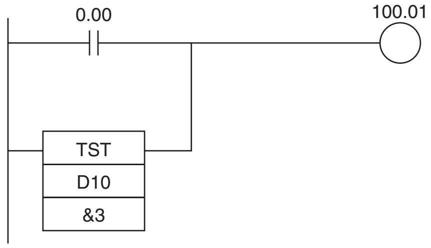
● OR TSTN

对 S 的 N 位的 ON/OFF 内容取反并将与输入条件的逻辑和反映到输入条件中。（使用方法与 OR 指令相同，接在其后对其他指令进行编程。）

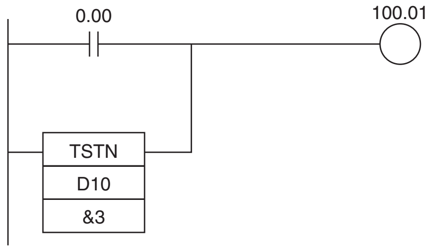
注意事项

- 请在本指令的末尾段附加输出类指令（OUT 类指令与下段连接型指令以外的应用指令）。
- 本指令无法用于电路的末尾段。
- 可在 CP2E CPU 单元中使用 OR TST/OR TSTN 指令。不能在 CP1E CPU 单元中使用。

程序举例



将0.00与D10的位3的逻辑和输出到100.01中。

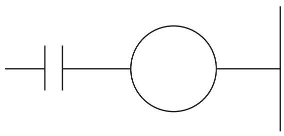
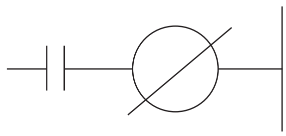


将0.00与D10的位3的非触点的逻辑和输出到100.01中。

顺序输出指令

OUT/OUT NOT

指令	助记符	变化	功能代码	功能
输出	OUT	!OUT	---	将逻辑处理的结果 (执行条件) 输出到 指定位。
反相输出	OUT NOT	!OUT NOT	---	将逻辑处理的结果 (执行条件) 取反后输出到指定位。

	OUT	OUT NOT
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
---	---	BOOL	---

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
OUT OUT NOT	OK	OK	OK	OK	---	---	OK	---	---	---	---	---	OK	---	---	OK

条件标志

该指令不影响任何标志。

功能

● OUT

如果没有即时刷新规定，则将执行条件 (能流) 的状态写入 I/O 存储器中的指定位中。如果有即时刷新规定，则除了将执行条件 (能流) 的状态写入 I/O 存储器中的输出位之外，还会写入 CPU 单元的内置输出输出端子。

● OUT NOT

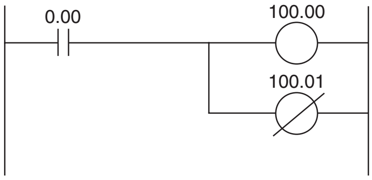
如果没有即时刷新规定，则将执行条件 (能流) 的状态取反后写入 I/O 存储器中的指定位中。如果有即时刷新规定，则除了将执行条件 (能流) 的状态取反后写入 I/O 存储器中的输出位之外，还会写入 CPU 单元的内置输出端子。

提示

・可为 OUT 和 OUT NOT 指定即时刷新 (!)。即时刷新指令在刚对 CPU 单元执行完指令后更新内置输出端子的状态，同时将执行条件 (能流) 的状态写入 I/O 存储器中的指定位。

・SET/RSET 和 OUT 的区别
对于 OUT，当输入条件变为 ON 时操作位变为 ON，而当输入条件变为 OFF 时操作位变为 OFF。对于 SET 和 RSET，当输入条件变为 ON 时，操作位分别变为 ON 或 OFF ；而当输入条件变为 OFF 时，操作位不变。

程序举例



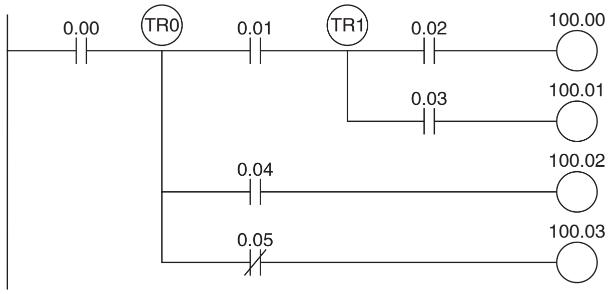
指令	操作数
LD	0.00
OUT	100.00
OUT NOT	100.01

TR

指令	助记符	变化	功能代码	功能
TR 位	TR	---	---	当以助记符编程时，TR 位用于临时保留程序中的执行条件的 ON/OFF 状态。

功能

当以助记符编程时，TR 位用于临时保留程序中的执行条件的 ON/OFF 状态。当直接以梯形图的形式编程时，不使用 TR 位，因为处理步骤将通过外围设备自动执行。下图所示为使用两个 TR 位编程的简单应用。



编程

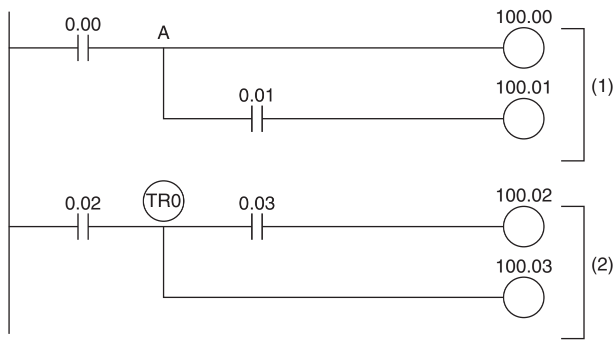
指令	操作数
LD	0.00
OUT	TR0
AND	0.01
OUT	TR1
AND	0.02
OUT	100.00
LD	TR1
AND	0.03
OUT	100.01
LD	TR0
AND	0.04
OUT	100.02
LD	TR0
AND NOT	100.00
OUT	100.03

继电器地址	
临时继电器	TR0 ~ TR15

● 使用 TR0 ~ TR15

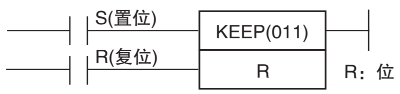
- TR0 ~ TR15 仅用于载入和输出指令。
- 位地址的使用顺序没有限制。
- 有时可重写程序使得不再需要 TR 位，从而可简化编程。下图所示为一个不需要TR位的例子和一个需要 TR 位的例子。

在指令块 (1) 中，A 点的 ON/OFF 状态与输出 CIO 100.00 相同，因此可编写代码 AND 0.01 和 OUT 100.01 而无需 TR 位。在指令块 (2) 中，分支点的状态和输出 CIO 100.02 的状态不一定相同，因此必须使用一个 TR 位。这种情况下，可使用指令块 (1) 代替指令块 (2) 来减少程序中的步数。



KEEP

指令	助记符	变化	功能代码	功能
保持	KEEP	!KEEP	011	运行方式类似于锁存继电器。

符号	KEEP
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
R	位	BOOL	---

● 区域类型

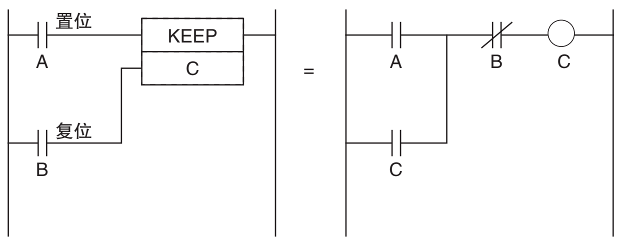
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
R	OK	OK	OK	OK	---	---	---	---	---	---	---	---	OK	---	---	OK

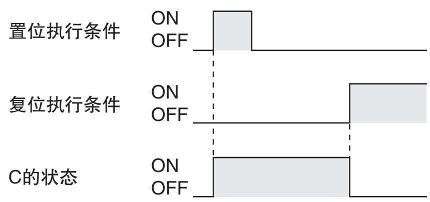
条件标志

KEEP(011) 指令不影响任何标志。

功能

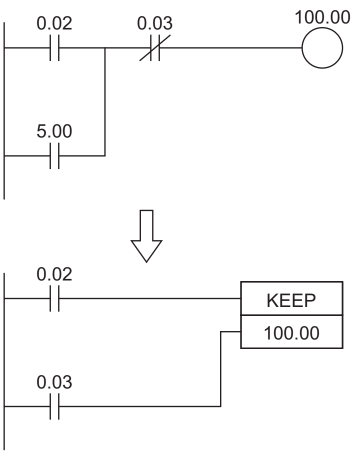
当 S 变为 ON 时，指定位将变为 ON，并且不论 S 是保持 ON 还是变为 OFF，指定位均保持 ON 直到被复位。当 R 变为 ON 时，指定位将变为 OFF。执行条件和 KEEP(011) 位状态之间的关系如右下图所示。



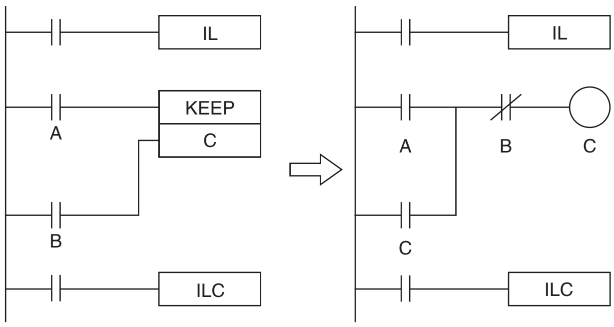


提示

- KEEP(011) 有即时刷新变化 (!KEEP(011))。当在 !KEEP(011) 指令中对 R 指定了一个 CPU 单元的内置输出位时，则执行 !KEEP(011) 指令时对 R 的任何改变都将被刷新并即时反映到输出位中。
- KEEP(011) 指令的运行类似于自保持位，但在 KEEP(011) 指令中使用自保持位编程时，所需指令少一条。



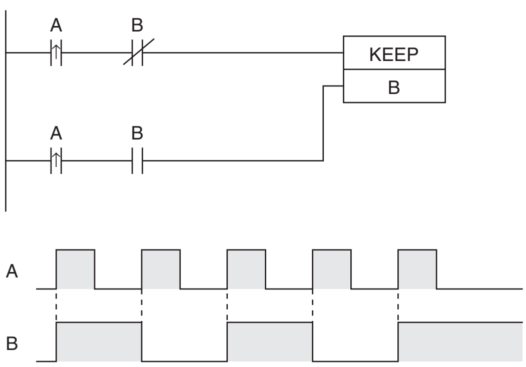
与不通过 KEEP(011) 指令编程的自保持位不同，通过 KEEP(011) 编程的自保持位即使在互锁的程序段中也将保持状态。



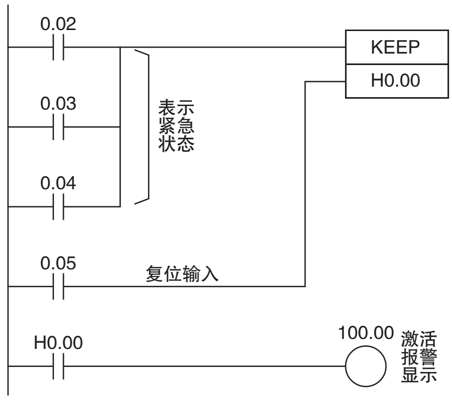
输出位C在互锁中
将保持其前状态

输出位C在互锁中
将变OFF

- KEEP(011) 可用于产生如下所示的触发器。



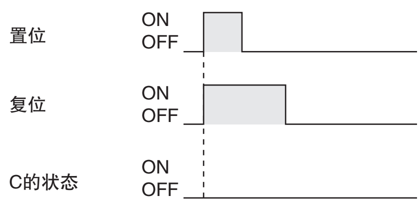
- 如果对 R 使用了一个保持位，则即使在电源中断期间位状态也将保留。因而 KEEP(011) 可用于位信号在电源中断后再重启 PLC 时仍保持不变的编程。下图是一个用于紧急状态下关断系统后产生报警显示信息的例子。



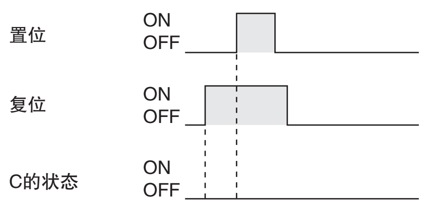
- 通过将 IOM 保持位变为 ON 并在 PLC 设置中将 IOM 保持位设定为“保持”，即可在电源中断事件中保持 I/O 区位的状态。在这种情况下，电源中断后重启 PLC 时，KEEP(011) 中使用的 I/O 区位的状态将如保持位一样得以保持。请务必在改变 PLC 的设置之后重启 PLC，否则将不采用新设定。

注意事项

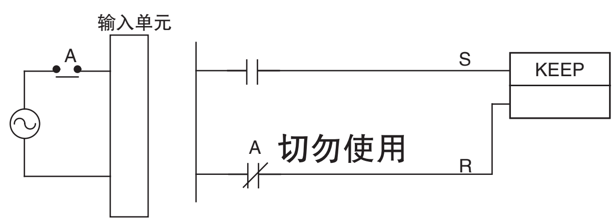
- 如果 S 和 R 同时为 ON，则复位输入的优先级更高。



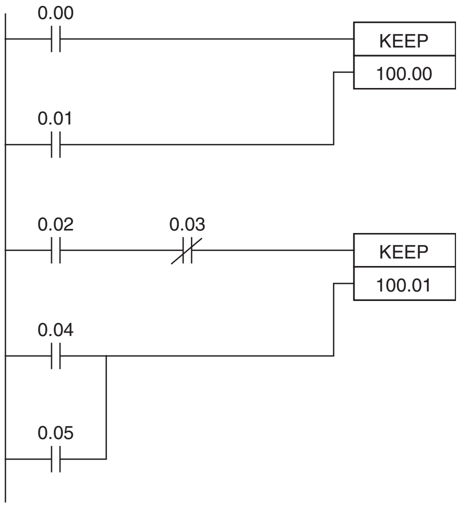
- 当 R 为 ON 时，无法接收置位输入 (S)。



- 当输入设备使用交流电源时，切勿在对 KEEP(011) 进行复位 (R) 的常闭条件中使用输入位。关闭PLC的直流电源时的延迟(相对于输入设备的交流电源而言) 可导致 KEEP(011) 的操作位被复位。该情况如右图所示。



程序举例



左例中，当 CIO 0.00 变为 ON 时，CIO 100.00 变为 ON。CIO 100.00 保持 ON，直到 CIO 0.01 变为 ON。
左例中，当 CIO 0.02 变为 ON 且 CIO 0.03 变为 OFF 时，CIO 100.01 变为 ON。CIO 100.01 保持 ON，直到 CIO 0.04 或 CIO 0.05 变为 ON。

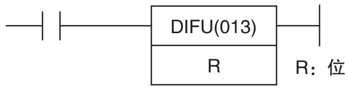
编程

指令	操作数
LD	0.00
LD	0.01
KEEP(011)	100.00
LD	0.02
AND NOT	0.03
LD	0.04
OR	0.05
KEEP(011)	100.01

注 KEEP(011) 以梯形图和助记符形式输入时，顺序不同。在梯形图中，先输入置位输入、KEEP(011)，然后输入复位输入。以助记符形式输入时，先输入置位输入、复位输入，然后输入 KEEP(011)。

DIFU

指令	助记符	变化	功能代码	功能
上升沿微分	DIFU	!DIFU	013	当执行条件从 OFF → ON(上升沿) 时，DIFU(013) 将指定位在一个循环中变为 ON。

符号	DIFU
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
R	位	BOOL	---

● 区域类型

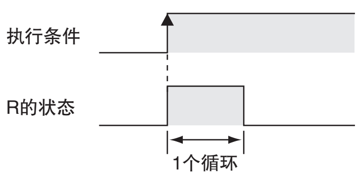
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
R	OK	OK	OK	OK	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

DIFU(013) 指令不影响任何标志。

功能

当执行条件从 OFF → ON 时，DIFU(013) 指令将 R 变为 ON。当 DIFU(013) 到达下一个循环时，R 变为 OFF。



提示

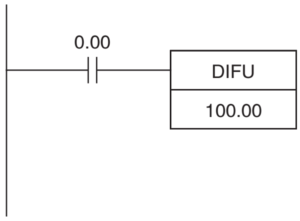
- 当执行条件从 OFF → ON 时，UP(521) 可用于使指令执行一个循环。
- DIFU(013)有即时刷新变化(!DIFU(013))。当在该指令中对R指定了一个CPU单元的内置输出位时，则执行该指令时对 R 的任何改变都将被刷新并即时反映到输出位中。

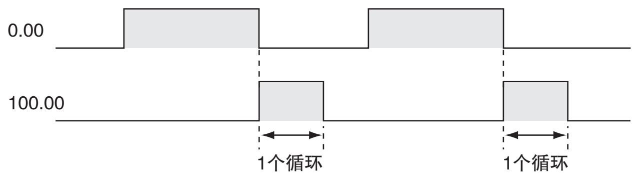
注意事项

- DIFU(013) 指令的运行不仅取决于指令自身的执行条件，还取决于当指令编入互锁程序段、跳转程序段或子程序时的程序段执行条件。
- 当子程序的输入条件为 OFF 时，子程序将不执行。在功能块定义中使用 DIFU(013) 时，应特别注意。详情请参考 SBS(091) 的相关说明。
- 如果同一个子程序在同一个循环中执行一次以上，则 DIFU(013) 的运行将不一致。

程序举例

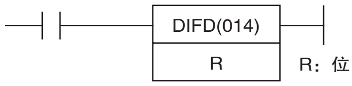
下例中，当 CIO 0.00 从 ON → OFF 时，CIO 100.00 将仅在一个循环中变为 ON。





DIFD

指令	助记符	变化	功能代码	功能
下降沿微分	DIFD	!DIFD	014	当执行条件从 ON → OFF(下降沿) 时，DIFD(014) 将指定位在一个循环中变为 ON。

	DIFD
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
R	位	BOOL	---

● 区域类型

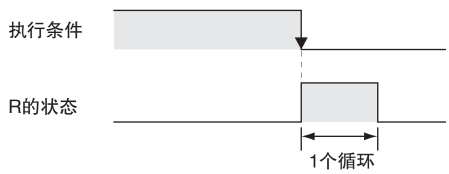
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
B	OK	OK	OK	OK	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

DIFD(014) 指令不影响任何标志。

功能

当执行条件从 ON → OFF 时，DIFD(014) 指令将 R 变为 ON。当 DIFD(014) 到达下一个循环时，R 变为 OFF。



提示

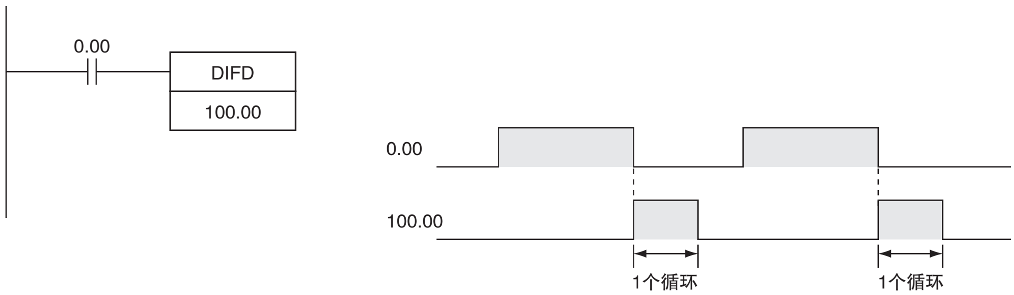
- 当执行条件从 ON → OFF 时，DOWN(522) 可用于使指令执行一个循环。
- DIFD(014) 指令的运行不仅取决于指令自身的执行条件，还取决于当指令编入互锁程序段、跳转程序段或子程序时的程序段执行条件。
- DIFD(014) 有即时刷新变化 (!DIFD(014))。当在该指令中对 R 指定了一个 CPU 单元的内置输出位时，则执行该指令时对 R 的任何改变都将被刷新并即时反映到输出位中。

注意事项

- 如果同一个功能块实例在同一个循环中执行一次以上，则 DIFD(014) 的运行将不一致。
- 当子程序的输入条件为 OFF 时，子程序将不执行。在功能块定义中使用 DIFD(014) 时，应特别注意。详情请参考 SBS(091) 的相关说明。

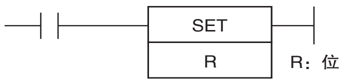
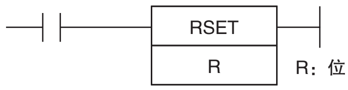
程序举例

下例中，当 CIO 0.00 从 ON → OFF 时，CIO 100.00 将仅在一个循环中变为 ON。



SET/RSET

指令	助记符	变化	功能代码	功能
置位	SET	@SET，%SET，!SET，!@SET，!%SET	---	当执行条件为 ON 时，SET 指令将操作数的位变为 ON。此后，无论输入条件的 ON/OFF 状态如何，指定的操作位均保持 ON。
复位	RSET	@RSET，%RSET，!RSET，!@RSET，!%RSET	---	当执行条件变为 ON 时，RSET 指令将操作位变为 OFF。此后，无论输入条件的 ON/OFF 状态如何，指定的操作位均保持 OFF。

	SET	RSET
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
R	位	BOOL	---

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
R	OK	OK	OK	OK	---	---	---	---	---	---	---	---	OK	---	---	---

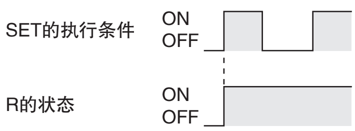
条件标志

SET 和 RSET 指令不影响任何标志。

功能

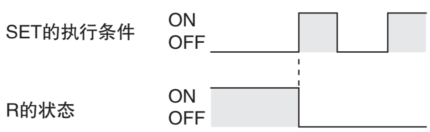
● SET

当执行条件为 ON 时，SET 指令将操作位变为 ON ；而当执行条件为 OFF 时，则不影响操作位的状态。可使用 RSET 指令将被 SET 指令置为 ON 的位变为 OFF。



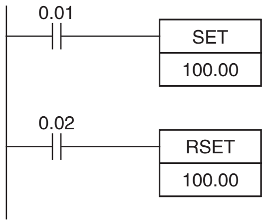
● RSET

当执行条件为 ON 时，RSET 指令将操作位变为 OFF ；而当执行条件为 OFF 时，则不影响操作位的状态。可使用 SET 指令将被 RSET 指令置为 OFF 的位变为 ON。



提示

- OUT/OUT NOT 和 SET/RSET 的区别
  SET 的运行与 OUT 的区别在于 OUT 指令在其执行条件为 OFF 时将操作位变为 OFF。与此类似，RSET 的运行与 OUT NOT 区别在于 OUT NOT 指令在其执行条件为 OFF 时将操作位变为 ON。对于 OUT，当输入条件变为 ON 时操作位变为 ON，而当输入条件变为 OFF 时操作位变为 OFF。对于 SET 和 RSET，当输入条件变为 ON 时，操作位分别变为 ON 或 OFF ；而当输入条件变为 OFF 时，操作位不变。



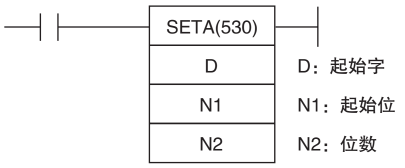
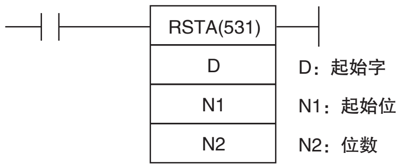
- KEEP(011) 指令的置位和复位输入必须用该指令编在一起，但 SET 和 RSET 指令在编程时则可完全独立。此外，同一个位可在任意数目的 SET 和 RSET 指令中用作操作数。
- SET 和 RSET 有即时刷新变化 (!SET 和 !RSET)。当在该指令中对 R 指定了一个 CPU 单元的内置输出位时，则执行该指令时对 R 的任何改变都将被刷新并即时反映到输出位中。
  如果通过 !SET(或 !RSET) 指令为 R 指定了外部输出，则 R 刚变为 ON(或 OFF) 时，将被 OUT 刷新。变为 ON(或 OFF) 之后的 R 将保持 ON(或 OFF) 作为常态，直到执行了一条 RSET 指令 (或 SET 指令) 为止。

注意事项

- 无法使用 SET 和 RSET 指令来对定时器和计数器进行置位和复位。当在 IL(002) 和 ILC(003) 指令或者 JMP(004) 和 JME(005) 指令之间编入 SET 或 RSET 指令时，如果程序段被互锁或跳转，则指定位的状态将不会改变。

SETA/RSTA

指令	助记符	变化	功能代码	功能
多个位置位	SETA	@SETA	530	SETA(530) 将指定的连续位数置 ON。
多个位复位	RSTA	@RSTA	531	RSTA(531) 将指定的连续位数置 OFF。

	SETA	RSTA
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
D	起始字	UINT	可变
N1	起始位	UINT	1
N2	位数	UINT	1

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
N1，N2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

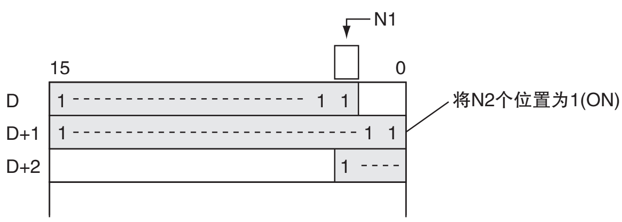
条件标志

操作数	描述	数据类型
出错标志	P_ER	· N1 不在 0000 ~ 000F(&0 ~ &15) 的指定范围内时为 ON。 · 其它情况下 OFF。

功能

● SETA

SETA(530) 指令将从 D 的 N1 位开始往左 (较高位) 的 N2 个位变为 ON。所有其它位均保持不变。(如果将 N2 置为 0，则不作改变。)

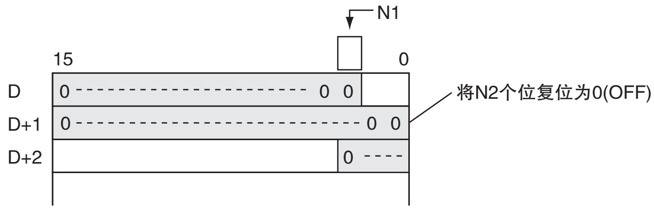


由 SETA(530) 指令变为 ON 的位可通过任何其它指令 (而不仅仅是 RSTA(531)) 变为 OFF。

● RSTA

RSTA(531) 指令将从 D 的 N1 位开始往左 (较高位) 的 N2 个位变为 OFF。所有其它位均保持不变。(如果将 N2 置为 0，则不作改变。)

由 RSTA(531) 指令变为 OFF 的位可通过任何其它指令 (而不仅仅是 SETA(530)) 变为 ON。



提示

● SETA

・ SETA(530) 可用于将通常只能通过字访问的数据区 (例如 DM 区) 中的位变为 ON。

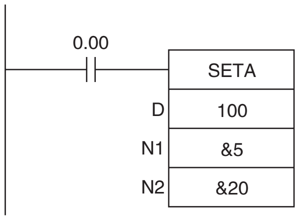
● RSTA

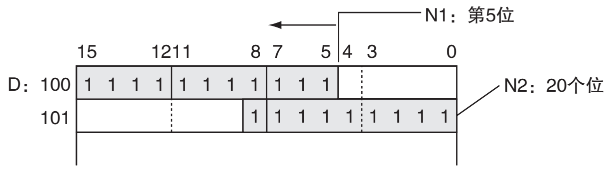
・ RSTA(531) 可用于将通常只能通过字访问的数据区 (例如 DM 区) 中的位变为 OFF。

程序举例

● SETA

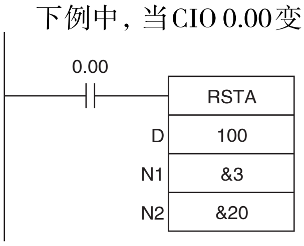
下例中，当 CIO 0.00 变为 ON 时，从 CIO 100 的第 5 位开始的 20 个位 (十六进制的 0014) 均变为 ON。

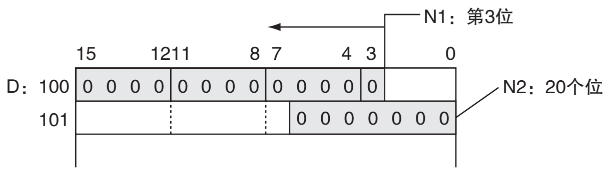




● RSTA

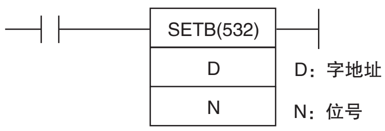
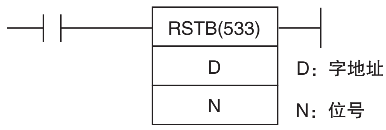
下例中，当 CIO 0.00 变为 ON 时，从 CIO 100 的第 3 位开始的 20 个位 (十六进制的 0014) 均变为 OFF。





SETB/RSTB

指令	助记符	变化	功能代码	功能
单个位置位	SETB	@SETB，!SETB，!@SETB	532	SETB(532) 将指定的位变为 ON。
单个位复位	RSTB	@RSTB，!RSTB，!@RSTB	533	RSTB(533) 将指定的位变为 OFF。

	SETB	RSTB
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
D	字地址	UINT	1
N	位号	UINT	1

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

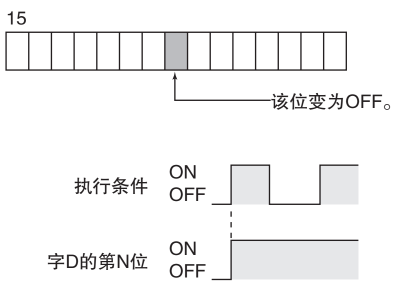
条件标志

操作数	描述	数据类型
出错标志	P_ER	· N 不在 0000 ～ 000F(&0 ～ &15) 的指定范围内时为 ON。 · 其它情况下 OFF。

功能

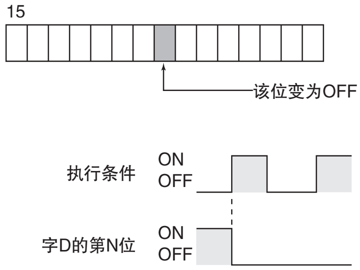
● SETB

当执行条件为 ON 时，SETB(532) 指令将字 D 的位 N 变为 ON。当执行条件为 OFF 时，该位的状态不受影响。



● RSTB

当执行条件为 ON 时，RSTB(533) 指令将字 D 的位 N 变为 OFF。当执行条件为 OFF 时，该位的状态不受影响。(用 SETB(532) 将该位变为 ON。)



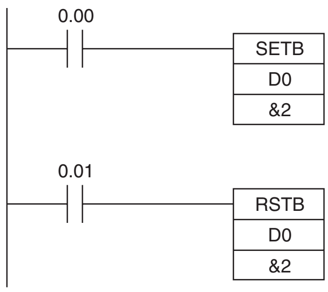
提示

- SET/RSET 和 SETB(532)/RSTB(533) 的区别
  当指定的位在 CIO、W、H 或 A 区时，这些指令的用法相同。
  SETB(532) 和 RSTB(533) 指令可控制 DM 区中的位，这一点与 SET 和 RSET 不同。
- KEEP(011) 指令的置位和复位输入必须用该指令编在一起，但 SETB(532) 和 RSTB(533) 指令在编程时则可完全独立。此外，同一个位可在任意数目的 SETB(532) 和 RSTB(533) 指令中用作操作数。

注意事项

- 由 SETB(532) 指令变为 ON 的位可通过任何其它指令 (而不仅仅是 RSTB(533)) 变为 OFF。
  由 RSTB(533) 指令变为 OFF 的位可通过任何其它指令 (而不仅仅是 SETB(532)) 变为 ON。
- SETB(532) 和 RSTB(533) 不能对定时器和计数器进行置位 / 复位。
- 当在 IL(002) 和 ILC(003) 指令或者 JMP(004) 和 JME(005) 指令之间编入 SETB(532) 或 RSTB(533) 指令时，如果程序段被互锁或跳转 (即互锁条件或跳转条件为 OFF 时)，则指定位的状态将不会改变。
- SETB(532) 和 RSTB(533) 有即时刷新变化 (!SETB(532) 和 !RSTB(533))。当在这些指令之一中指定了一个 CPU 单元的内置输出位时，则执行该指令时对指定位的任何改变都将被刷新并即时反映到输出位中。
- 如果通过 !SETB(或 !RSTB) 指令为字 D 的位地址 N 指定了一个 CPU 单元的内置输出，则已变为 ON(或 OFF) 字 D 的位地址 N 将在该点 (执行指令时) 被 OUT 刷新。变为 ON(或 OFF) 之后的字 D 的位地址 N 将保持 ON(或 OFF) 作为常态，直到执行了一条 RSTB 指令 (或 SETB 指令) 为止。

程序举例



当 CIO 0.00 为 ON 时，D0 的第 02 位变为 ON。

当 CIO 0.01 为 ON 时，D0 的第 02 位变为 OFF。

顺序控制指令

互锁指令概要

● 互锁指令

可使用以下指令组合来使程序段中的输出互锁。

- 互锁和互锁清除 (IL(002) 和 IL(003))
- 多路互锁微分保持和多路互锁清除 (MILH(517) 和 MILC(519))*

注　MILH(517) 保持微分标志的状态，因而在互锁清除后，将执行被互锁的微分指令。

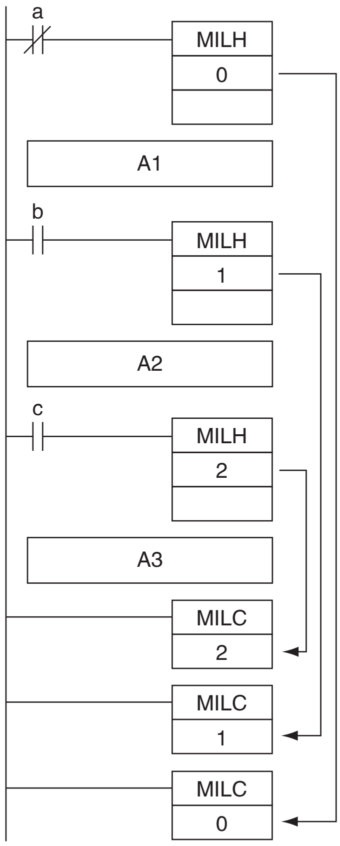
- 多路互锁微分释放和多路互锁清除 (MILR(518) 和 MILC(519))*

注　MILR(518) 不保持微分标志的状态，因而在互锁清除后，不执行被互锁的微分指令。

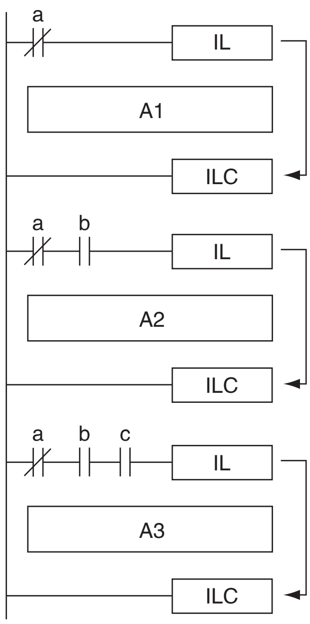
● 互锁和多路互锁的区别

常规的互锁 (IL(002) 和 IL(003)) 无法嵌套，但多路互锁 (MILH(517)、MILR(518) 和 MILC(519)) 则可嵌套。通过使多路互锁嵌套，可简化梯形图编程，如下图所示。

通过MILH和MILC互锁



通过IL和ILC互锁



● MILH(517) 和 MILR(518) 的区别

微分指令 (DIFU、DIFD 或者带前缀 @ 或 % 的指令) 的运行与通过 MILH(517) 和 MILR(518) 创建的互锁有所不同。

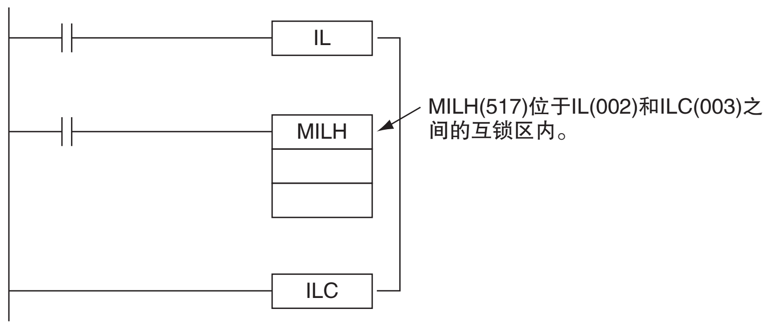
微分指令在通过 MILH(517) 创建的互锁中的运行与在通过 IL(002) 创建的互锁中的运行相同。

有关详情，请参考 *"多路互锁微分保持、多路互锁微分释放和多路互锁清除：MILH(517)、MILR(518) 和 MILC(519)"*。

● 注意事项

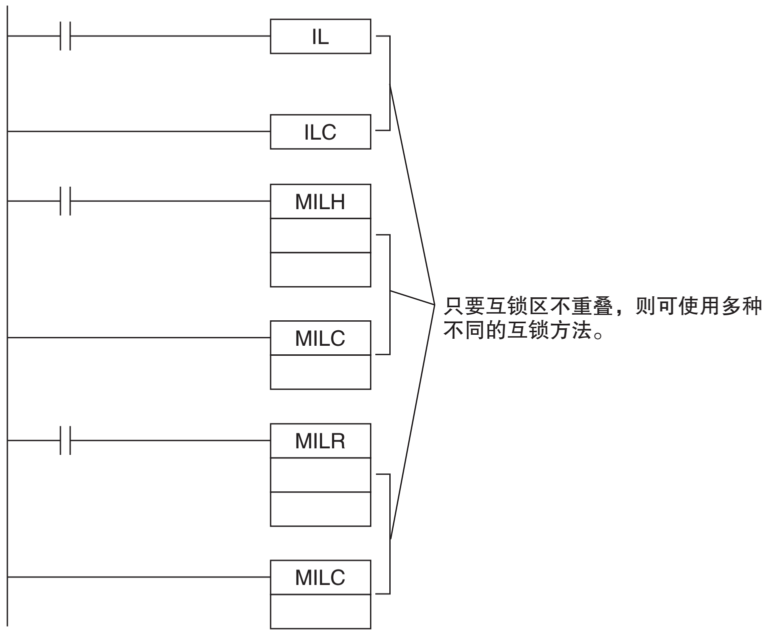
请勿组合用不同的互锁指令 (IL-ILC、MILH-MILC 和 MILR-MILC) 创建的互锁。如果一起使用不同的互锁方法，则互锁可能无法运行。有关组合指令的详情，请参考 *"多路互锁微分保持、多路互锁微分释放和多路互锁清除：MILH(517)、MILR(518) 和 MILC(519)"*。

例如，不能将 MILH(517) 指令插在 IL(002) 和 IL(003) 之间。



注 只要互锁程序段不重叠，则可一起使用不同的互锁 (IL-ILC、MILH-MILC 和 MILR-MILC)。

例如，下述三种互锁方法在不相互重叠时可一起使用，如下图所示。



● 互锁和跳转的区别

下表列出了互锁 (通过 IL(002)/ILC(003)、MILH(517)/MILC(519) 或 MILR(518)/MILC(519) 指令创建) 和通过 JMP(004)/JME(005) 创建的跳转之间的区别。

项目	IL(002)/ILC(003)、MILH(517)/MILC(519) 或 MILR(518)/MILC(519) 中的处理	JMP(004)/JME(005) 中的处理
指令执行	除 OUT、OUT NOT 和定时器指令之外，其它所有指令均不执行。	不执行任何指令。
指令的输出状态	除 OUT、OUT NOT 和定时器指令的输出之外，其它所有输出均保持其前状态。	所有输出均保持其前状态。
OUT、OUT NOT 中的位	OFF	所有输出均保持其前状态。
定时器指令 (TTIM(087)、TTIMX(555)、MTIM(543) 和 MTIMX(554) 除外) 的状态	复位	正在运行的定时器 (仅限 TIM、TIMX(550)、TIMH(015)、TIMHX(551)、TMHH(540) 和 TMHHX(552)) 继续计时，因为即使不执行定时器指令时也会更新 PV。

END

指令	助记符	变化	功能代码	功能
结束	END	---	001	表示一个程序结束。

符号	END
	END(001)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	---	---	OK

条件标志

该指令不影响任何标志。

功能

END(001) 完成一个循环内的程序执行。END(001) 后面的任何指令均不执行。

注意事项

- 请务必在每个程序的结尾处放置 END(001) 指令。如果程序中没有 END(001) 指令，则将产生编程错误。

NOP

指令	助记符	变化	功能代码	功能
空操作	NOP	---	000	该指令无任何功能。

符号	NOP
	(NOP(000) 没有梯形图符号)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

条件标志

NOP(000) 指令不影响任何标志。

功能

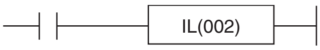
- NOP(000) 指令不执行任何处理，但该指令可用于在程序中将来要插入指令处留出程序行的位置。
- NOP(000) 指令只能用于助记符中，而不能用于梯形图程序中。

提示

- 将来插入指令时，程序的地址不会有变化。

IL/ILC

指令	助记符	变化	功能代码	功能
互锁	IL	---	002	当 IL(002) 的执行条件为 OFF 时，IL(002) 和 ILC(003) 之间的所有输出均互锁。
互锁清除	ILC	---	003	表示互锁范围的结束。

符号	IL	ILC
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	---	OK	OK

条件标志

该指令不影响任何标志。

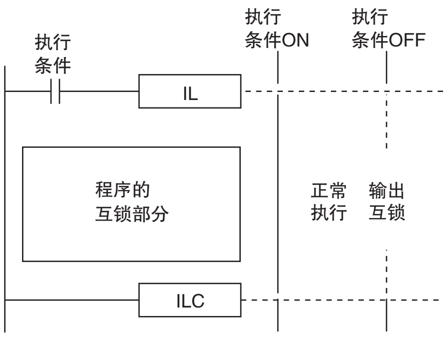
功能

当 IL(002) 的执行条件为 OFF 时，IL(002) 和 ILC(003) 之间的所有指令的输出均被互锁。当 IL(002) 的执行条件为 ON 时，IL(002) 和 ILC(003) 之间的所有指令均正常执行。

下表列出了对 IL(002) 和 ILC(003) 之间的互锁程序段中的各种输出的处理。

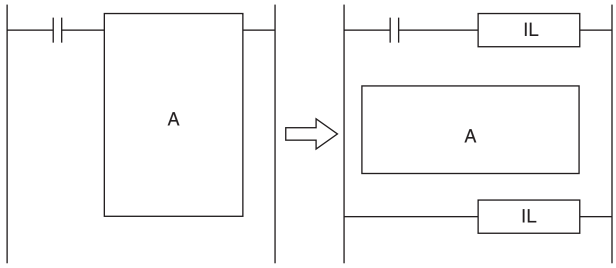
<table>
<tr><th colspan="2">指令</th><th>处理</th></tr>
<tr><td colspan="2">OUT、OUT NOT 中的指定位</td><td>OFF</td></tr>
<tr><td rowspan="2">TIM、TIMX(550)、TIMH(015)、TIMHX(551)、TMHH(540)、TMHHX(552)、TIML(542) 和 TIMXL(553)</td><td>完成标志</td><td>OFF(复位)</td></tr>
<tr><td>PV</td><td>时间设定值 (复位)</td></tr>
<tr><td colspan="2">在所有其它指令中指定的位 / 字 (见注)</td><td>保持前状态</td></tr>
</table>

注 所有其它指令 (包括 TTIM(087)、TTIMX(555)、SET、RSET、CNT、CNTX(546)、CNTR(012)、CNTRX(548)、SFT 和 KEEP(011)) 中的位和字均保持其前状态。



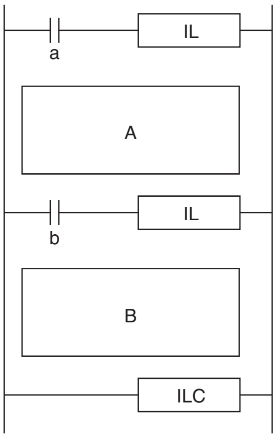
提示

- 如果想要在一个互锁程序段中使某些位保持 ON，可通过紧靠 IL(002) 之前的 SET 指令将这些位置为 ON。
- 通过 IL(002) 和 ILC(003) 指令来切换程序段的效率通常更高。当通过相同的执行条件来控制多个过程时，将这些过程放入 IL(002) 和 ILC(003) 指令之间执行所用的程序步数更少。



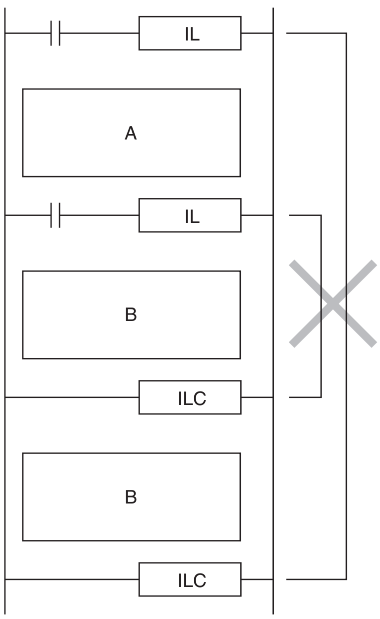
注意事项

- 当程序的一部分被互锁后，循环时间不会缩短，因为被互锁的指令在内部执行。
- 尽管可将一个以上的 IL(002) 与单个 ILC(003) 配合使用，但 IL(002) 和 ILC(003) 通常成对使用，如下图所示。如果 IL(002) 和 ILC(003) 不成对使用，则当执行程序检查时将报错，但程序仍可正常执行。



执行条件		程序段	
a	b	A	B
OFF	ON	互锁	互锁
OFF	OFF	互锁	互锁
ON	OFF	不互锁	互锁
ON	ON	不互锁	不互锁

- IL(002) 和 ILC(003) 不能嵌套，如下图所示。(必须使互锁嵌套时，请使用 MILH(517)/MILR(518) 和 MILC(519) 指令。)

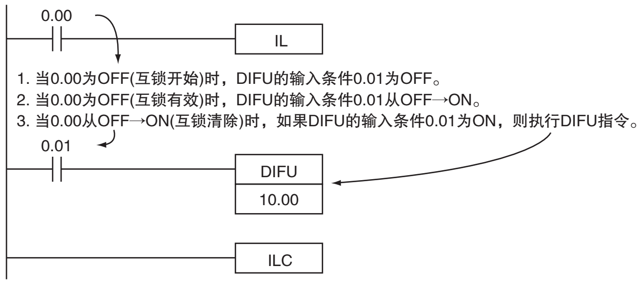


● 微分指令的运行

如果在 IL(002) 和 ILC(003) 指令之间有一条微分指令 (DIFU、DIFD 或者带 @ 或 % 前缀的指令)，则当因互锁开始到清除期间的输入条件发生变化而使得指令的微分条件成立时，该指令将在互锁清除后执行。

例：

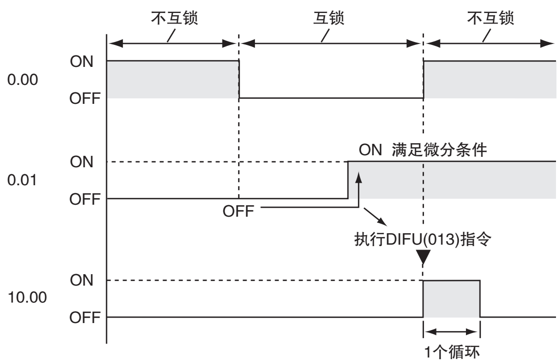
当使用上升沿微分 (DIFU(013)) 指令且在互锁开始时输入条件为 OFF、而在互锁清除时输入条件为 ON 时，则在互锁清除时将执行上升沿微分 (DIFU(013)) 指令。



参考：

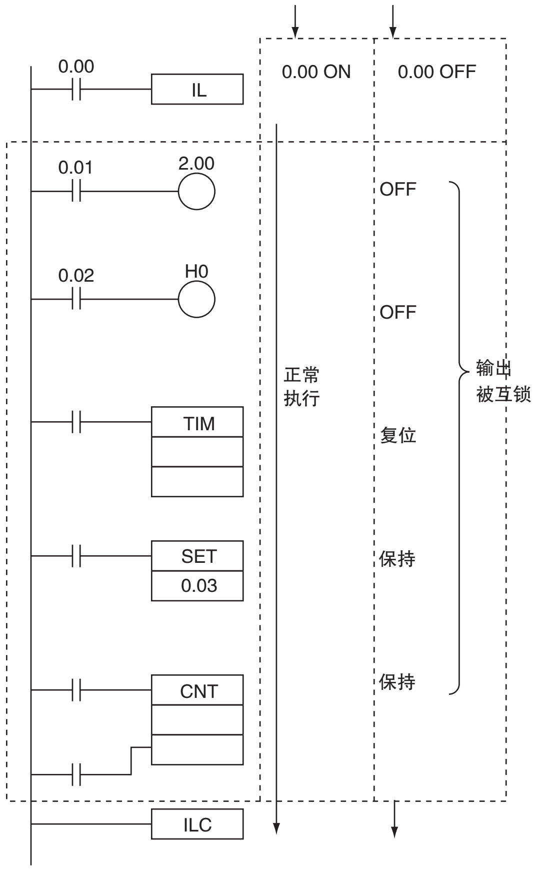
就微分指令的运行方式而言，IL(002) 指令与 MILH(517) 指令相同。

时序图



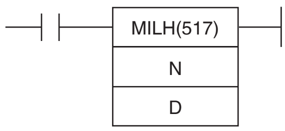
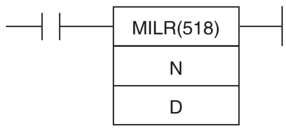
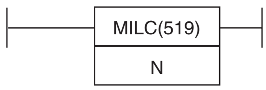
程序举例

右例中，当 CIO 0.00 为 OFF 时，IL(002) 和 ILC(003) 之间的所有输出均被互锁。右例中，当 CIO 0.00 为 ON 时，IL(002) 和 ILC(003) 之间的所有指令均正常执行。



MILH/MILR/MILC

指令	助记符	变化	功能代码	功能
多路互锁微分保持	MILH	---	517	当MILH(517)的执行条件为OFF时，MILH(517)和MILC(519)之间的所有输出均互锁。
多路互锁微分释放	MILR	---	518	当MILR(518)的执行条件为OFF时，MILR(518)和MILC(519)之间的所有输出均互锁。
多路互锁清除	MILC	---	519	表示通过MILH或MILR指令创建的具有相同互锁号的多路互锁范围的结束。

<table>
<tr><th rowspan="2">符号</th><th>MILH</th><th>MILR</th><th>MILC</th></tr>
<tr><td>
N：互锁号
D：互锁状态位</td><td>
N：互锁号
D：互锁状态位</td><td>
N：互锁号</td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	---	OK	OK

操作数

操作数	描述	数据类型	大小
N	互锁号	--	1
D	互锁状态位	BOOL	--

N：互锁号

互锁号必须介于 0 ～ 15 之间。MILH(517)(或 MILR(518)) 指令中的互锁号需与对应 MILC(519) 指令中的互锁号保持相同。

互锁号的使用顺序不限。

D：互锁状态位

・ 当程序段不互锁时为 ON。

・ 当程序段互锁时为 OFF。

处于互锁状态时，可对互锁状态位强行置位，以释放互锁。与之相反，不处于互锁状态时，可对互锁状态位强行复位，以进入互锁。

● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>N</td><td>---</td><td>---</td><td>---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>D</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	OFF

功能

当互锁号为 N 的 MILH(517)(或 MILR(518)) 的执行条件为 OFF 时，MILH(517)/MILR(518) 指令和互锁号为 N 的下一条 MILC(519) 指令之间的所有指令的输出均被互锁。
当互锁号为 N 的 MILH(517)(或 MILR(518)) 的执行条件为 ON 时，MILH(517)/MILR(518) 指令和互锁号为 N 的下一条 MILC(519) 指令之间的所有指令均正常执行。

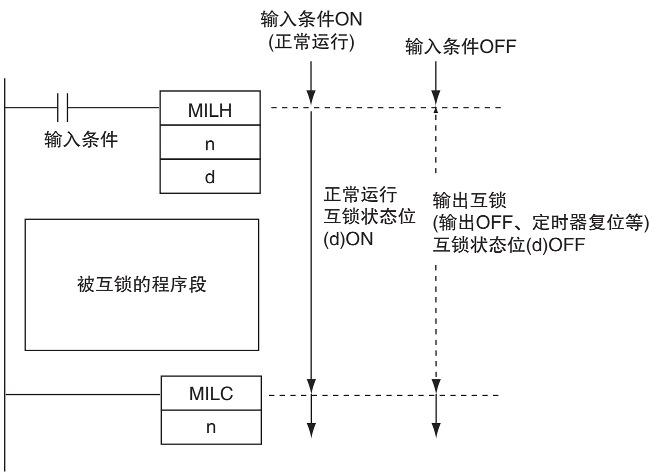
● 互锁状态

下表列出了对 MILH(517)/MILR(518) 指令和下一条 MILC(519) 指令之间的互锁程序段中的各种输出的处理。

指令		处理
OUT、OUT NOT 中的指定位		OFF
TIM、TIMX(550)、TIMH(015)、TIMHX(551)、TMHH(540)、TMHHX(552)、TIML(542) 和 TIMXL(553)	完成标志	OFF(复位)
	PV	时间设定值 (复位)
在所有其它指令中指定的位 / 字 (见注)		保持前状态

注 所有其它指令 (包括 TTIM(087)、TTIMX(555)、SET、RSET、CNT、CNTX(546)、CNTR(012)、CNTRX(548)、SFT 和 KEEP(011)) 中的位和字均保持其前状态。

当处于互锁状态时，MILH(517)/MILR(518) 指令将互锁状态位 (操作数 D) 变为 OFF，而当未处于互锁状态时，则将其变为 ON。因此可对互锁状态位进行监视，以检查给定互锁号的某个互锁是否处于互锁状态中。

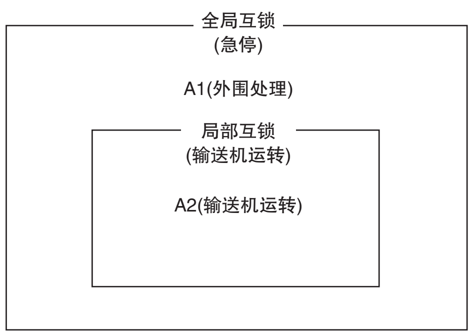


● 嵌套

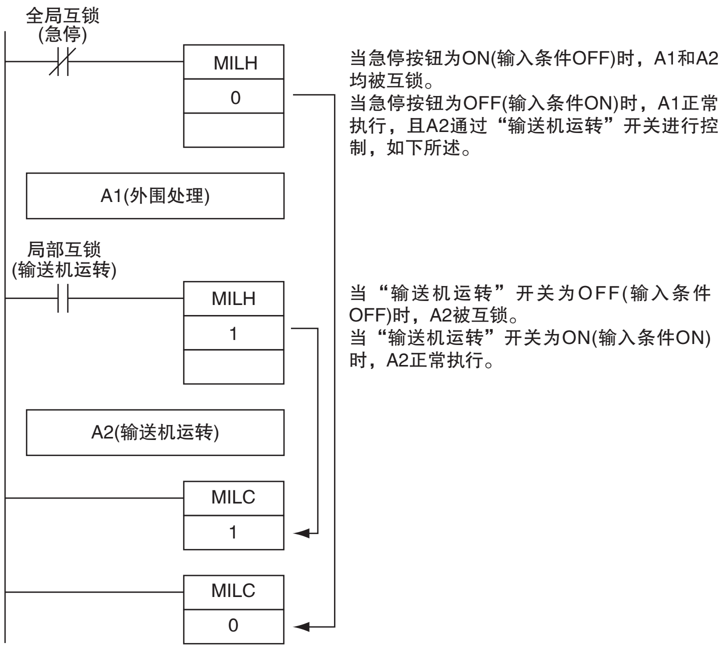
当将某个被互锁的程序段 (MILH(517)/MILR(518) 和 MILC(519) 组合) 放在另一个被互锁的程段 (MILH(517)/MILR(518) 和 MILC(519) 组合内时互锁即嵌套。互锁最多可嵌套 16 层。
嵌套可用于以下场合。

例 1

用某个条件互锁整个程序，并用另一个条件互锁程序的某一部分 (1 层嵌套)

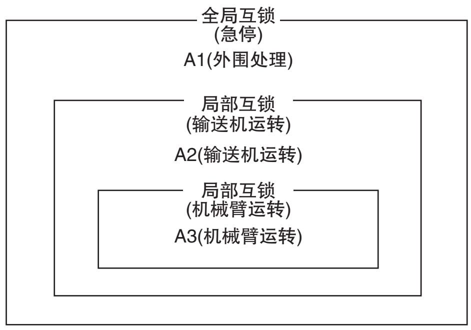


· 当急停按钮为 ON 时，A1 和 A2 被互锁。
· 当 “输送机运转” 为 OFF 时，A2 被互锁。

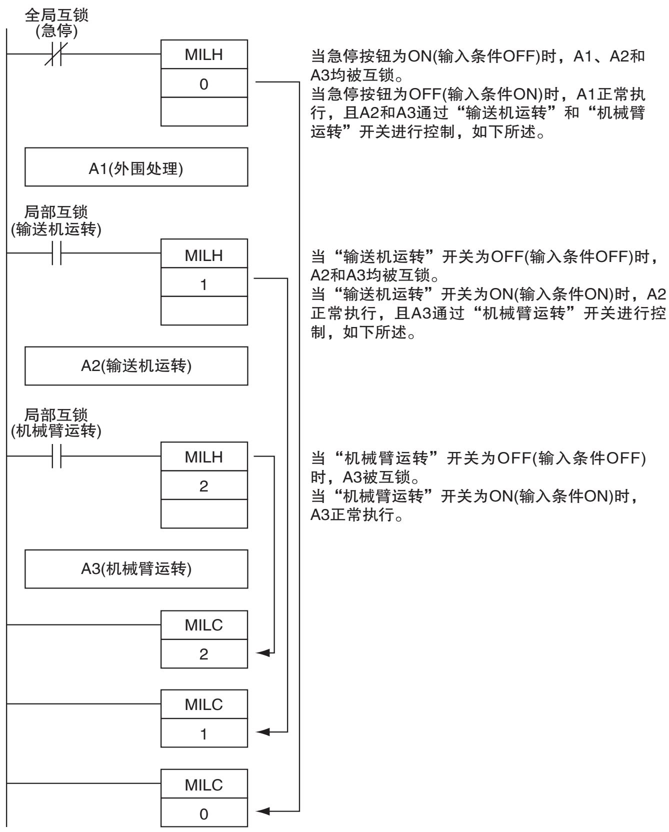


例 2

用某个条件互锁整个程序，并用另一个条件互锁程序中的两个重叠部分 (2 层嵌套)



· 当急停按钮为 ON 时，A1、A2 和 A3 被互锁。
· 当 “输送机运转” 为 OFF 时，A2 和 A3 被互锁。
· 当 “机械臂运转” 为 OFF 时，A3 被互锁。



● MILH(517) 和 MILR(518) 的区别

微分指令 (DIFU、DIFD 或者带前缀 @ 或 % 的指令) 的运行与通过 MILH(517) 和 MILR(518) 创建的互锁有所不同。

当一个程序段被 MILR(518) 互锁之后，即使在互锁期间微分条件被激活，互锁清除后微分指令也将无法执行 (将互锁开始时的执行条件状态与互锁清除后的执行条件状态作对比)。

当一个程序段被 MILH(517) 互锁之后，如果互锁期间某个微分条件被激活，则互锁清除后将执行该微分指令 (将互锁开始时的执行条件状态与互锁清除后的执行条件状态作对比)。

指令	微分指令的运行
MILH(517) 多路互锁微分保持	如果在指令被互锁期间该指令的某个微分条件成立，则在互锁清除后**将执行**微分指令 (DIFU、DIFD 或者带 @ 或 % 前缀的指令)。(将互锁开始时的执行条件状态与互锁清除时的执行条件状态作对比。)
MILR(518) 多路互锁微分释放	即使在指令被互锁期间该指令的某个微分条件成立，则在互锁清除后也**不执行**微分指令 (DIFU、DIFD 或者带 @ 或 % 前缀的指令)。

● 微分指令在 MILH(517) 互锁中的运行

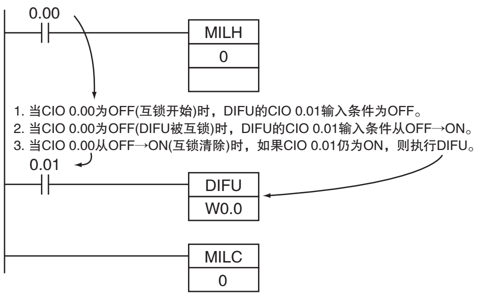
如果在 MILH(517) 及其对应的 MILC(519) 指令之间有一条微分指令 (DIFU、DIFD 或者带 @ 或 % 前缀的指令)，则如果该指令的微分条件成立，则在互锁清除后将执行该指令。

同样，如果某条微分指令的执行条件在互锁开始或清除的同时成立，则将执行该微分指令。

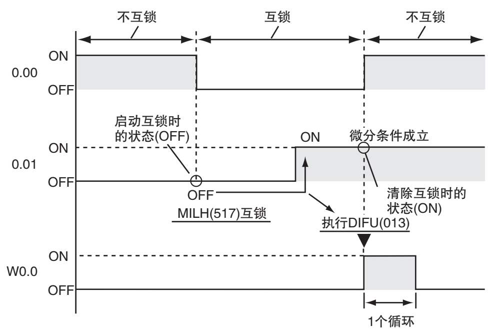
即使微分条件在互锁期间成立，但程序中的许多其它条件也可能会造成该微分条件复位。在这种情况下，当互锁清除时将不执行该微分指令。

例：

当正在使用上升沿微分 (DIFU(013)) 指令且在互锁开始时输入条件为 OFF、而在互锁清除时输入条件为 ON 时，则在互锁清除时将执行 DIFU(013) 指令。(微分指令在 MILH(517) 互锁中的运行与在 IL(002) 互锁中相同。)



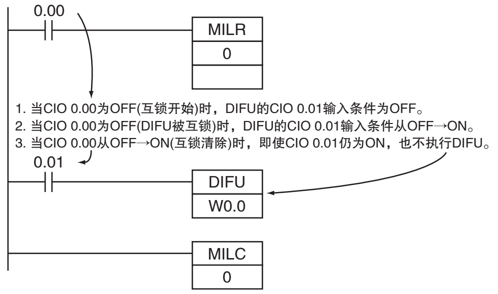
时序图



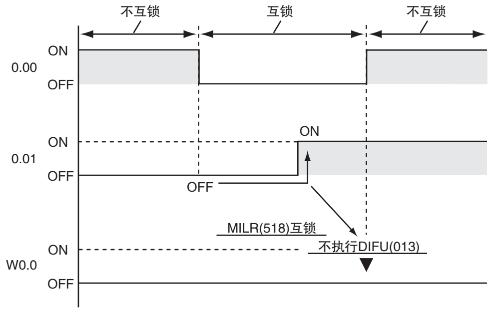
● 微分指令在 MILR(518) 互锁中的运行

如果在 MILR(518) 及其对应的 MILC(519) 指令之间有一条微分指令 (DIFU、DIFD 或者带 @ 或 % 前缀的指令)，则即使该指令的微分条件成立，互锁清除后也不会执行该指令。
同样，如果某条微分指令的执行条件在互锁开始或清除的同时成立，则将不执行该微分指令。

例：
当正在使用上升沿微分 (DIFU(013)) 指令且在互锁开始时输入条件为 OFF、而在互锁清除时输入条件为 ON 时，则在互锁清除时将不执行 DIFU(013) 指令。



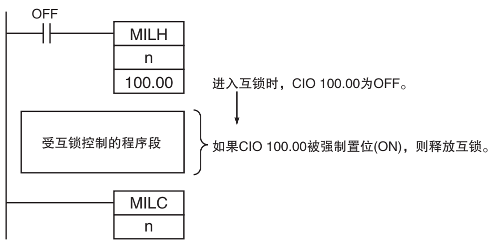
时序图



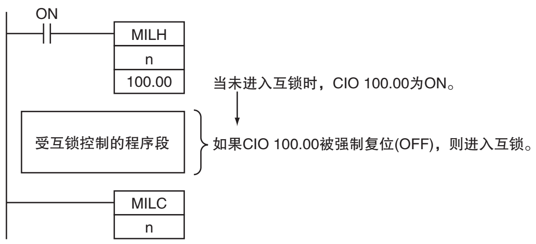
● 从编程设备控制互锁状态

可通过从编程设备对互锁状态位进行强制置位或强制复位 (用 MILH(517) 和 MILR(518) 的操作数 D 指定) 的方式来手动进入互锁或释放互锁。互锁状态位的强制状态有优先权，可覆写通过程序执行所计算得出的互锁状态。

强制置位：释放互锁

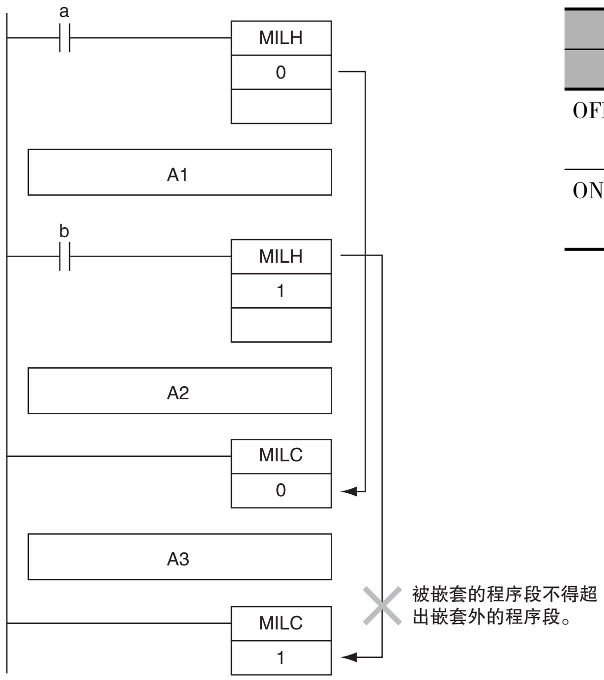


强制复位：进入互锁



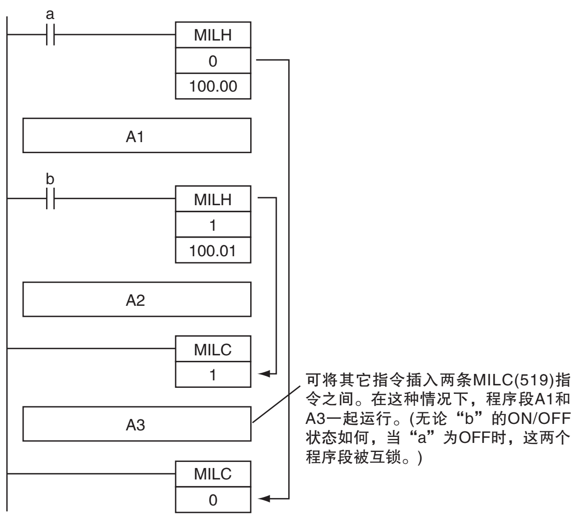
提示

- 当程序的一部分被 MILH(517) 或 MILR(518) 互锁后，循环时间不会缩短，因为被互锁的指令在内部执行。
- 要嵌套互锁时，分配给被嵌套的程序段的互锁号不能超过分配给嵌套外的程序段的互锁号。

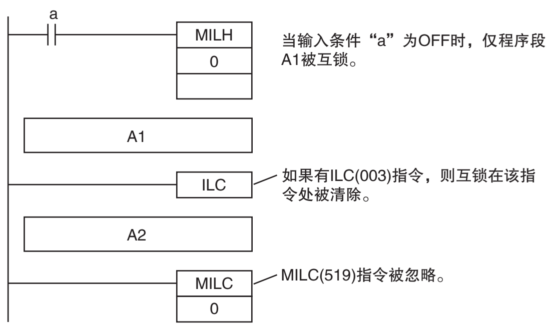


执行条件		程序段		
a	b	A1	A2	A3
OFF	ON	互锁	互锁	互锁
	OFF			
ON	OFF	不互锁	互锁	互锁
	ON	不互锁	不互锁	不互锁

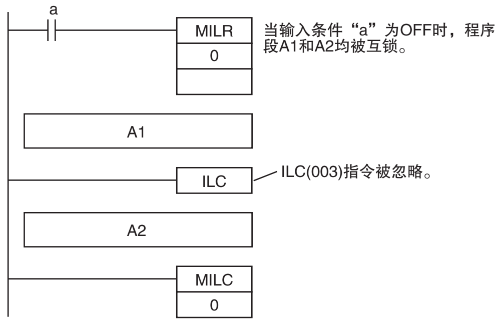
- 在 MILC(519) 指令之间可输入其它指令，如下图所示。



- 如果在一对 MILH(517) 和 MILC(519) 指令之间有一条 ILC(003) 指令，则 MILH(517) 和 ILC(003) 之间的程序段将被互锁。

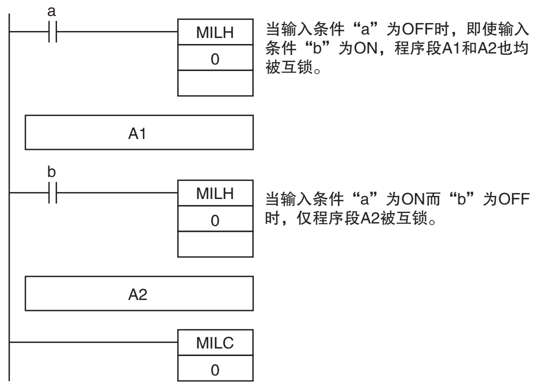


・如果在一对 MILR(518) 和 MILC(519) 指令之间有一条 ILC(003) 指令，则 ILC(003) 指令将被忽略，且 MILR(518) 和 MILC(519) 之间的整个程序段将被互锁。



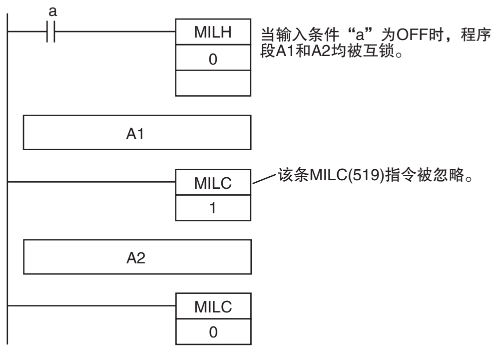
・如果在一对 MILH(517) 和 MILC(519) 指令之间有另一条互锁号相同的 MILH(517) 或 MILR(518) 指令，且已进入第一条 MILH(517) 指令的互锁，则第二条MILH(517)/MILR(518) 将不执行。

・如果在一对 MILH(517) 和 MILC(519) 指令之间有另一条互锁号相同的 MILH(517) 或 MILR(518) 指令，且未进入第一条 MILH(517) 指令的互锁，则第二条MILH(517)/MILR(518) 将正常执行。



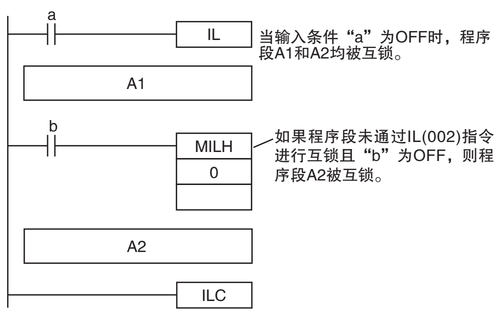
注 如果在一对 MILR(518) 和 MILC(519) 指令之间有另一条互锁号相同的 MILH(517) 或 MILR(518) 指令，则 MILR(518) 互锁的运行方式不变。

・如果在一对MILH(517)/MILR(518)和MILC(519) 指令之间有一条互锁号不同的 MILC(519) 指令，则这条 MILC(519) 指令将被忽略。

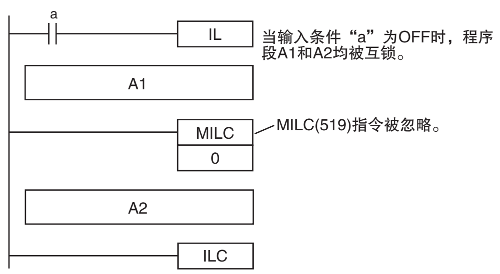


・如果在一对 IL(002) 和 ILC(003) 指令之间有一条 MILH(517) 指令且已进入 IL(002) 互锁，则 MILH(517) 指令不起作用。在这种情况下，位于 IL(002) 和 ILC(003) 之间程序段将被互锁。

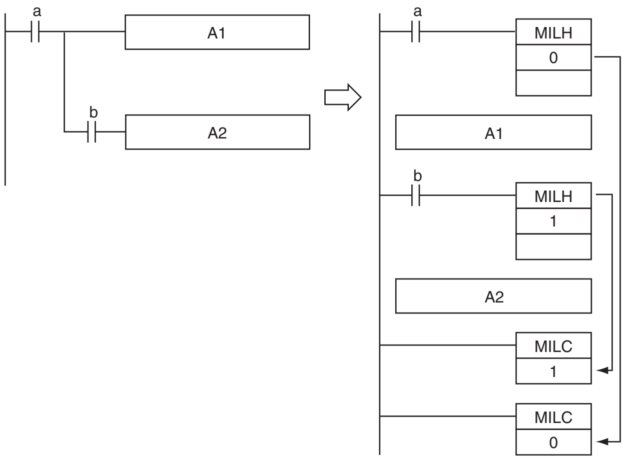
如果未进入 IL(002) 互锁且 MILH(517) 指令的执行条件 (在该情况下为 b) 为 OFF，则 MILH(517) 和 ILC(003) 之间的程序段将被互锁。



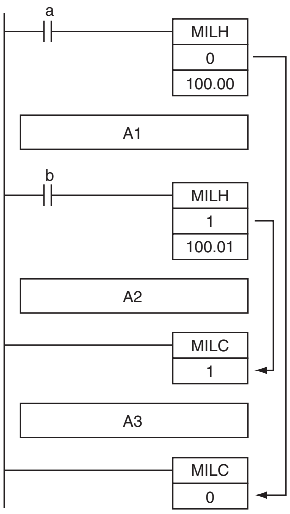
・如果在一对 IL(002) 和 ILC(003) 指令之间有一条 MILC(519) 指令，则该 MILC(519) 指令将被忽略且 IL(002) 和 ILC(003) 之间的整个程序段将被互锁。



- 使用 MILH(517) 或 MILR(518) 创建的互锁可提高切换程序运行的效率。
  可在各过程之前插入一条 MILH(517) 或 MILR(518) 指令并在各过程之后插入一条 MILC(519) 指令即可，而无需通过复合条件来切换处理。

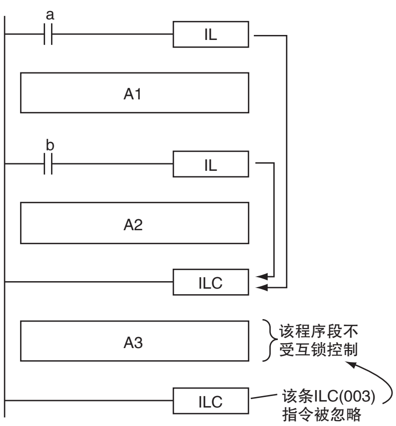


- 与 IL(002) 互锁指令不同，MILH(517) 和 MILR(518) 指令可以嵌套。因此如果使用 MILH(517) 或 MILR(518) 来替代 ILC(002)，相似程序的运行将不同。
  - 带有 MILH(517)/MILC(519) 互锁的程序



执行条件		程序段		
a	b	A1	A2	A3
OFF	ON	互锁	互锁	不互锁
	OFF			
ON	OFF	不互锁	互锁	不互锁
ON	ON	不互锁	不互锁	不互锁

  - 带有 IL(002)/ILC(003) 互锁的程序



执行条件		程序段		
a	b	A1	A2	A3
OFF	ON	互锁	互锁	不互锁 (不受 IL(002)/ILC(003) 互锁控制)
	OFF			
ON	OFF	不互锁	互锁	
ON	ON	不互锁	不互锁	

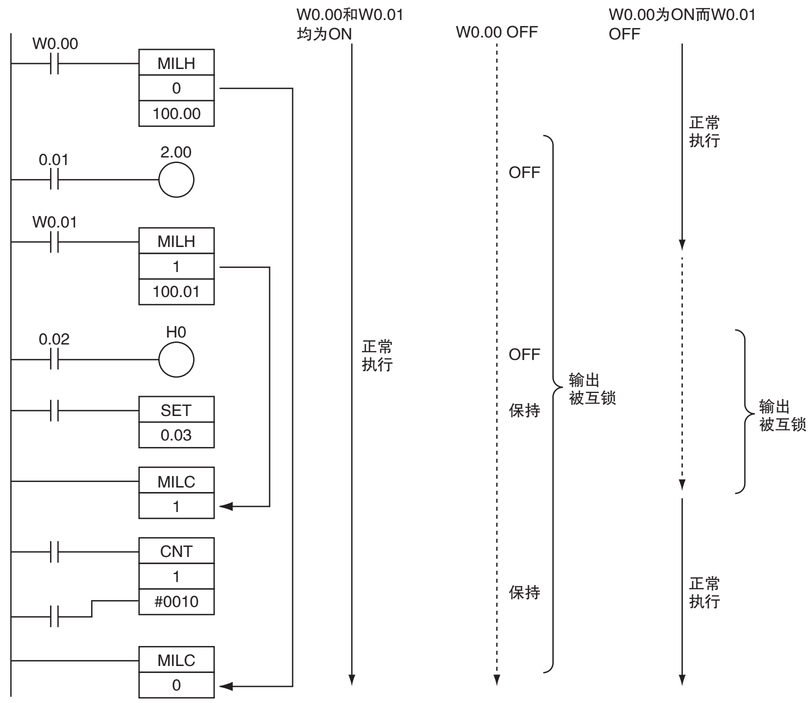
- 如果想要在一个被 MILH(517) 或 MILR(518) 互锁的程序段中使某些位保持 ON，可通过紧靠 MILH(517) 或 MILR(518) 指令之前的 SET 指令将这些位置为 ON。

程序举例

当 W0.00 和 W0.01 均为 ON 时，互锁号为 0 的 MILH(517) 指令和互锁号为 0 的 MILC(519) 指令之间的指令将正常执行。

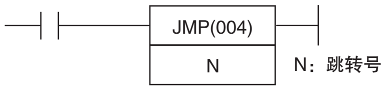
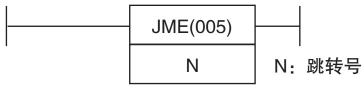
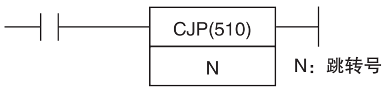
当W0.00为OFF时，互锁号为0的MILH(517)指令和互锁号为0的MILC(519)指令之间的指令被互锁。

当 W0.00 为 ON 而 W0.01 为 OFF 时，互锁号为 1 的 MILH(517) 指令和互锁号为 1 的 MILC(519) 指令之间的指令被互锁。其它指令将正常执行。



JMP/CJP/JME

指令	助记符	变化	功能代码	功能
跳转	JMP	---	004	当 JMP(004) 的执行条件为 OFF 时，程序执行直接跳转至程序中具有相同跳转号的第一个 JME(005) 指令。
条件跳转	CJP	---	510	当 CJP(510) 的执行条件为 ON 时，程序执行直接跳转至程序中具有相同跳转号的第一个 JME(005) 指令。
跳转结束	JME	---	005	表示通过 JMP 或 CJP 指令设定的跳转过程的结束位置。

符号	JMP	JME
		
	CJP	
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	---	OK	OK

操作数

操作数	描述	数据类型	大小
N	跳转号	UINT	1

N：跳转号

跳转号必须介于 0000 ～ 007F(十进制的 &0 ～ &127) 之间。

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
JMP/CJP	N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
JME	N	---	---	---	---	---	---	---	---	---		---		---			

条件标志

● JMP/CJP

名称	标记	操作
出错标志	P_ER	· 当 N 不在 0000 ～ 007F 的指定范围内时为 ON。 · 当程序中有一条 JMP(004) 但无同一跳转号的 JME(005) 时为 ON。 · 其它情况下 OFF。

● JME

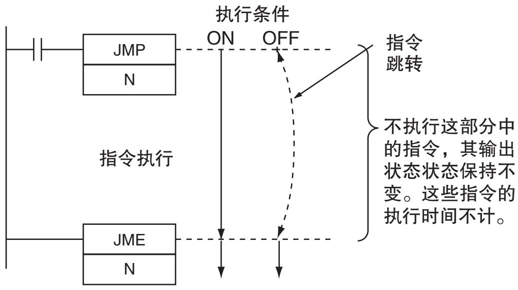
该指令不影响任何标志。

功能

● JMP

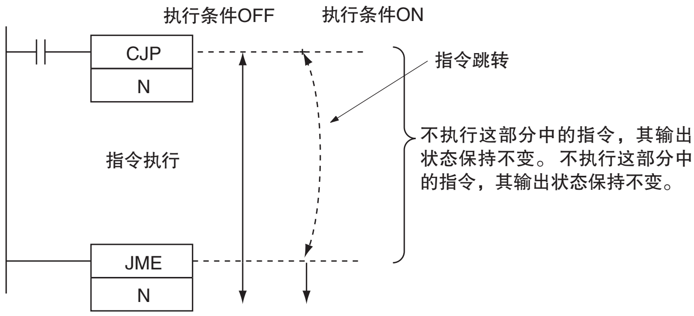
当JMP(004)的执行条件为ON时，不进行跳转，且程序按编写的顺序连续执行。

当JMP(004)的执行条件为OFF时，程序执行直接跳转至程序中具有相同跳转号的第一个JME(005)指令。JMP(004)和JME(005)之间的指令将不执行，因此JMP(004)和JME(005)之间的输出状态得以保持。在块程序中，无论执行条件的状态如何，JMP(004)和JME(005)之间的指令均被跳过。



● CJP

当 CJP(510) 的执行条件为OFF时，不进行跳转，且程序按编写的顺序连续执行。当CJP(510)的执行条件为ON时，程序执行直接跳转至程序中具有相同跳转号的第一条JME(005)指令。



提示

- 由于在 JMP(004) 的执行条件为 OFF 时，JMP(004)/CJP(510) 和 JME(005) 之间的所有指令均被跳过，因此循环时间将减去被跳过的指令的总执行时间。与此相反，JMP0(515) 和 JME0(516) 之间的指令所需的处理时间与 NOP(000) 处理相同，因此循环时间减少的量少于上述跳转指令。
- 下表给出了各种跳转指令的对比。

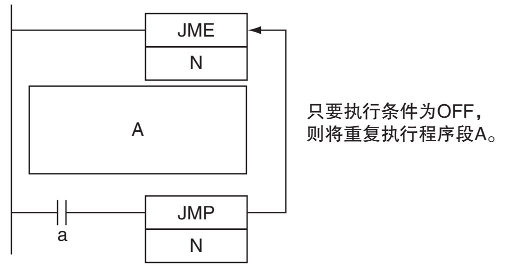
项目	JMP(004) JME(005)	CJP(510) JME(005)
跳转的执行条件	OFF	ON
允许的次数	128	
跳转时的指令处理	不执行	
跳转时的指令执行时间	无校验	
跳转时的输出(位和字)状态	位和字保持其前状态	
跳转时正在运行的定时器的状态	正在运行的定时器继续计时	
在块程序中的处理	始终跳转	ON时跳转

注意事项

- 被跳转的指令中的所有输出（位和字）均保持其前状态。正在运行的定时器 (TIM、TIMX(550)、TIMH(015)、TIMHX(551)、TMHH(540) 和 TMHHX(552)) 继续计时，因为即使不执行定时器指令时也会更新 PV。

・当存在两条或两条以上跳转号相同的 JME(005) 指令时，仅地址较低的指令有效。地址较高的 JME(005) 指令将被忽略。

・CJP(510) 在执行条件为 ON 时跳转至第一条 JME(005) 指令，而 JMP(004) 在执行条件为 OFF 时跳转至第一条 JME(005) 指令。



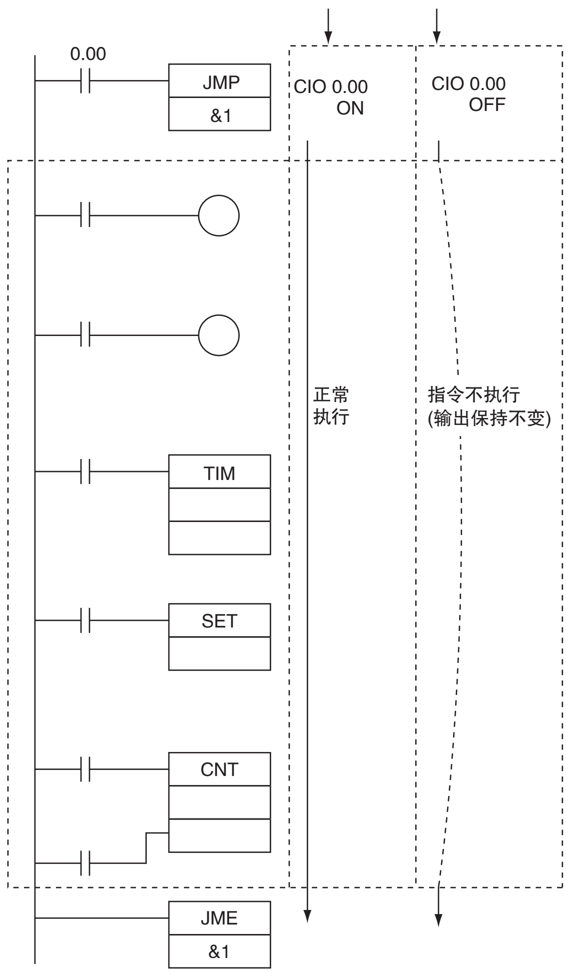
・如果 JME(005) 在程序中的位置在 JMP(004)/CJP(510) 之前，则只要 JMP(004)/CJP(510) 的执行条件为 OFF，JME(005) 和 JMP(004)/CJP(510) 之间的指令将反复执行。如果在最大循环时间内执行条件未变为 ON 或者未执行 END(001) 指令，则将产生 "循环时间超长"的错误。

・当 DIFU(013)、DIFD(014) 和微分指令编在 JMP(004)/CJP(510) 和 JME(005) 指令之间时，这些指令并不完全依赖于执行条件。当 DIFU(013)、DIFD(014) 或微分指令在 JMP(004)/CJP(510) 的执行条件刚变为 ON 时即在某个跳转的程序段中执行，则 DIFU(013)、DIFD(014) 或微分指令的执行条件将与跳转生效前 (即 JMP(004) 的执行条件变为 OFF 之前) 的执行条件作比较。

程序举例

右例中，当 CIO 0.00 为 OFF 时，JMP(004) 和 JME(005) 之间的指令将不执行，且输出保持其前状态。

右例中，当 CIO 0.00 为 ON 时，JMP(004) 和 JME(005) 之间的所有指令均正常执行。



FOR/NEXT

指令	助记符	变化	功能代码	功能
---	FOR	---	512	将 FOR(512) 和 NEXT(513) 之间的指令重复执行指定的次数。
	NEXT	---	513	

	FOR	NEXT
符号	FOR(512) N　N：循环次数	NEXT(513)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	---	OK	OK

操作数

操作数	描述	数据类型	大小
N	循环次数	UINT	1

N：循环次数

循环次数必须介于 0000 ～ FFFF(十进制的 0 ～ 65,535) 之间。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

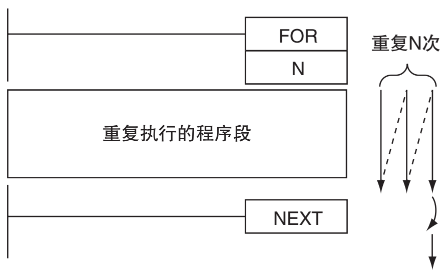
名称	标记	操作
出错标志	P_ER	· 循环嵌套超过 15 层时为 ON。 · 其它情况下 OFF。
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

将 FOR(512) 和 NEXT(513) 之间的指令重复执行 N 次，然后程序继续执行NEXT(513)之后的指令。BREAK(514) 指令可用于取消循环。

如果 N 被置为 0，则 FOR(512) 和 NEXT(513) 之间的指令将作为 NOP(000) 指令处理。

循环可通过最少的编程量来实现对数据表的处理。



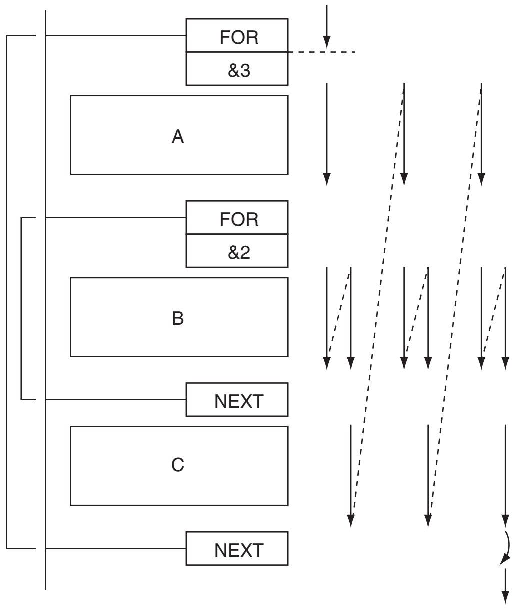
提示

有两种方法可重复执行某个程序段，直到输入了给定的执行条件为止。

- 带 BREAK 的 FOR-NEXT 循环
  开始一个最多重复 N 次的 FOR-NEXT 循环。以所需的执行条件在循环中编入 BREAK(514) 指令。如果输入了上述执行条件，则循环将在重复 N 次之前结束。
- JME(005)-JMP(004) 循环
  在 JMP(004) 之前插入 JME(005) 指令来编写循环。只要 JMP(004) 的执行条件为 OFF，则 JME(005) 和 JMP(004) 之间的指令将反复执行。(如果在最大循环时间内执行条件未变为 ON 或者未执行 END(001) 指令，则将产生 “循环时间超长”的错误。)

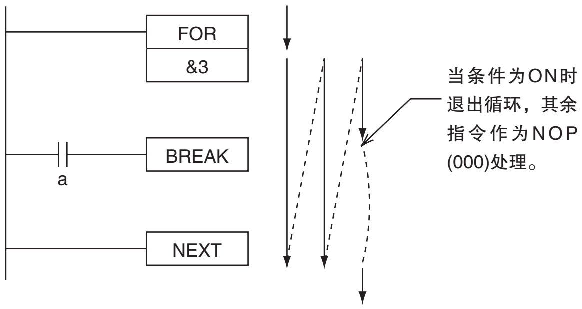
注意事项

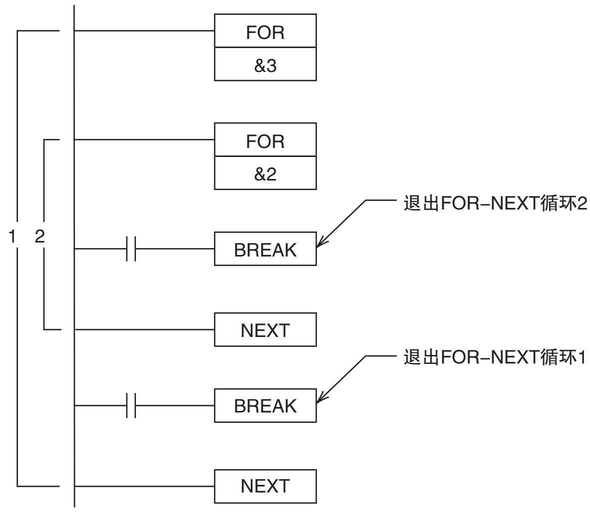
- 应将 FOR(512) 和 NEXT(513) 编入同一个任务中。如果这两条指令不在同一个任务中，则将不重复执行。
- 如果循环在一个周期内重复执行，且在 FOR-NEXT 循环中使用一条微分指令，则该指令仅将执行一次，而无法执行与循环相同的次数。
  - UP(521)、DOWN(522)
  - DIFU(013)、DIFD(014)
  - 上升沿微分指令 (微分变化：@)
  - 下降沿微分指令 (微分变化：%)
- FOR-NEXT 循环最多可嵌套 15 层。



在上例中，程序段 A、B 和 C 的执行顺序如下：
A → B → B → C, A → B → B → C，以及 A → B → B → C

- 可使用 BREAK(514) 退出 FOR-NEXT 循环。从嵌套的循环中退出需要多条 BREAK(514) 指令 (嵌套的层数)。
- 循环中 BREAK(514) 之后的其余指令作为 NOP(000) 指令处理。



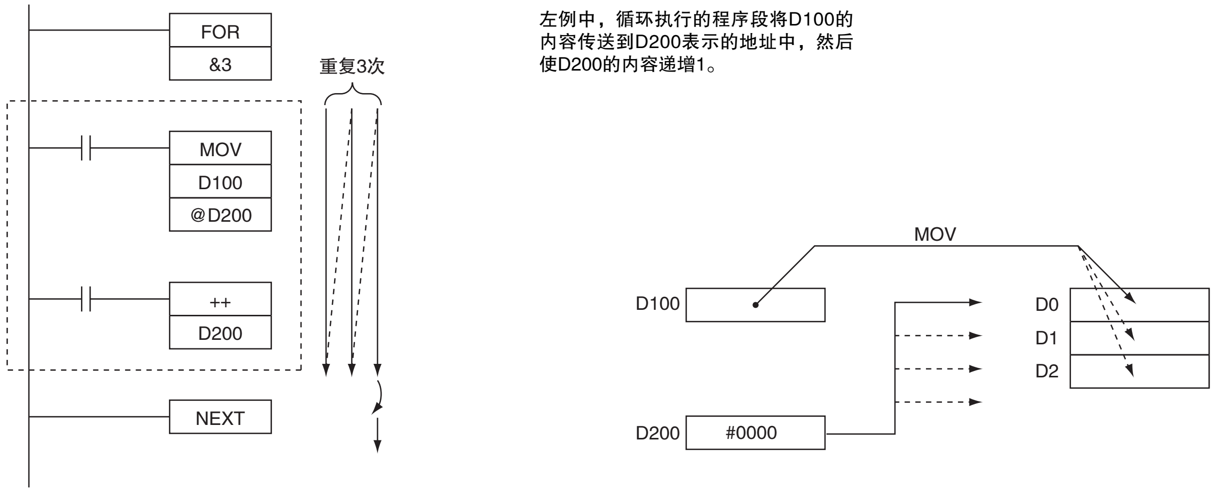


・ JMP(004) 之类的跳转指令可在 FOR-NEXT 循环内部执行，但不能跳出 FOR-NEXT 循环之外。

・ 下述指令不能在 FOR-NEXT 循环内部使用。

  ・ 步定义和步开始：STEP(008)/SNXT(009)

程序举例



BREAK

指令	助记符	变化	功能代码	功能
循环中断	BREAK	---	514	用于 FOR–NEXT 循环语句中的编程，作用是对于给定的执行条件取消循环的执行。循环中剩余的指令作为 NOP(000) 指令处理。

符号	BREAK
	─┤├──BREAK(514)──┤

适用程序区

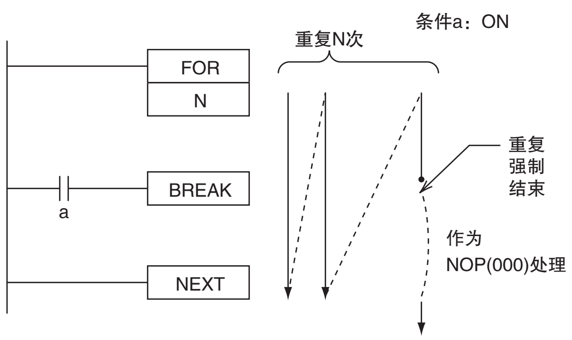
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	---	OK	OK

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

在FOR(512)和NEXT(513)之间编入BREAK(514)指令，从而在执行 BREAK(514) 时取消 FOR–NEXT 循环。当 BREAK(514) 指令执行时，到 NEXT(513) 为止的其余指令作为 NOP(000) 处理。



注意事项

- 由于一条 BREAK(514) 指令只能取消一个循环，因此从嵌套的循环中退出需要多条 BREAK(514) 指令 (嵌套的层数)。
- BREAK(514) 只能用于 FOR–NEXT 循环。

定时器和计数器指令

定时器 / 计数器 PV 的刷新方法

● 概要

刷新定时器 / 计数器相关指令的 PV 有两种方法，即 “BCD” 和 “二进制”。

方法	描述	设定范围	设定值
BCD	以 BCD 码设定定时器的设定值。	0~9.999	#0000~9999
二进制	以二进制码设定定时器的设定值。	0~65.535	&0 ~ 65535或#0000 ~ FFFF

PLC 设置所有定时器 / 计数器相关指令。在间接设定 SV(即通过存储器字的内容进行设定) 时，刷新方法同样有效。(也就是说，根据所设定的刷新方法，读取被寻址的字中的内容作为 BCD 或二进制数据。)

● 适用指令

分类	指令	助记符	
		BCD	二进制
定时器 / 计数器指令	100ms 定时器	TIM	TIMX(550)
	10ms 定时器	TIMH(015)	TIMHX(551)
	1ms 定时器	TMHH(540)	TMHHX(552)
	累加定时器	TTIM(087)	TTIMX(555)
	长定时器	TIML(542)	TIMLX(553)
	计数器	CNT	CNTX(546)
	可逆计数器	CNTR(012)	CNTRX(548)
	复位定时器 / 计数器	CNR(545)	CNRX(547)

● PV 刷新的设定方法

可在同一个项目中同时刷新 BCD 和二进制 PV。此时，在 PLC 设置中设定的 PV 刷新方法将被忽略。

● 定时器的基本规定

项目		TIM/TIMX (550)	TIMH(015)/ TIMHX(551)	TMHH(540)/ TMHHX(552)	TTIM(087)/ TTIMX(555)	TIML(542)/ TIMLX(553)
定时方式		递减	递减	递减	递增	递减
定时单位		100ms	10ms	1ms	100ms	100ms
最大 SV		TIM：999.9s TIMX：6,553.5s	TIMH：99.99s TIMHX：655.35s	TMHH：9.999s TMHHX：65.535s	TTIM：999.9s TTIMX：6,553.5s	TIML：115 天 TIMLX：49,710 天
输出 / 指令		1	1	1	1	1
定时器号		使用	使用	使用	使用	不使用
完成标志刷新		执行指令时	执行指令时	执行指令时	执行指令时	执行指令时
定时器 PV 刷新		执行指令时	执行指令时	执行指令时	执行指令时	执行指令时
复位后的值	完成标志	OFF	OFF	OFF	OFF	OFF
	PV	SV	SV	SV	0	SV

● 运行模式

项目		TIM/TIMX(550)	TIMH(015)/TIMHX(551)	TMHH(540)/TMHHX(552)	TTIM(087)/TTIMX(555)	TIML(542)/TIMLX(553)
运行模式改变		PV=0 完成标志 =OFF				---
电源中断 / 复位		PV=0 完成标志 =OFF				---
CNR(545)/CNRX(547) 的执行		二进制：PV=FFFF，完成标志 =OFF BCD：PV=FFFF 或 9999，完成标志 =OFF				不适用
在跳转程序段 (JMP(004)–JME(005)) 中的处理		正在运行的定时器继续计时			定时器状态保持	
在互锁的程序段 (IL(002)–ILC(003)) 中的处理		PV=SV 完成标志 =OFF			保持定时器状态	PV=SV 完成标志 =OFF
强制置位	完成标志	ON				---
	PV	置为 0				---
强制复位	完成标志	OFF				---
	PV	复位到 SV			置为 0	---

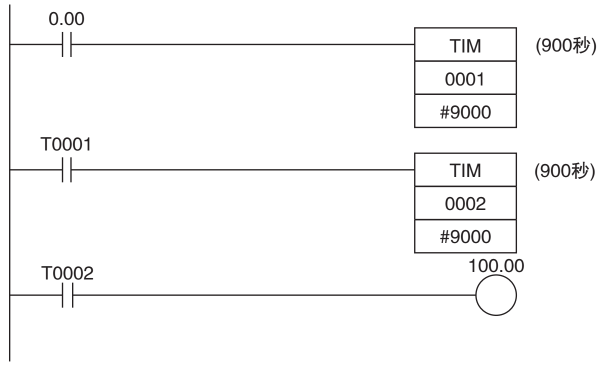
● 定时器和计数器的应用实例

例 1：长时间定时器

下述程序范例所示为通过标准的 TIM 和 CNT 指令来生成长时间定时器的三种方法。

1) 两条 TIM 指令

在该例中，组合了两条 TIM 指令以生成一个 30 分钟定时器。

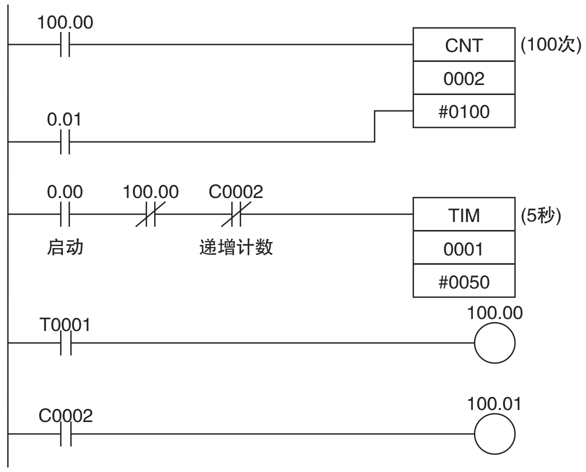


指令	操作数
LD	0.00
TM	1
	#9000
LD	T0001
TM	2
	#9000
LD	T000
OUT	100.00

2)TIM 和 CNT 指令

在该例中，组合了一条 TIM 指令和一条 CNT 指令以生成一个 500 秒定时器。

TIM 0001 每 5 秒产生一个脉冲，而 CNT 0002 则对这些脉冲进行计数。该组合的设定值为定时器间隔 × 计数器 SV。在这种情况下，定时器 SV 为 5 秒 ×100=500 秒。使用该组合时，长时间定时器的 PV 实际上为计数器的 PV(计数器的 PV 在电源中断时得以保持)。

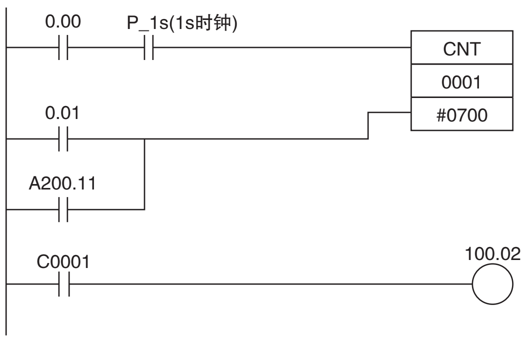


指令	操作数
LD	100.00
LD	0.01
CNT	2
	#0100
LD	0.00
AND NOT	100.00
AND NOT	C0002
TIM	1
	#0050
LD	T0001
OUT	100.00
LD	C0002
OUT	100.01

3) 时钟脉冲和 CNT 指令

在该例中，一条 CNT 指令从 1 秒时钟脉冲开始对脉冲进行计数，以生成一个 700 秒定时器。

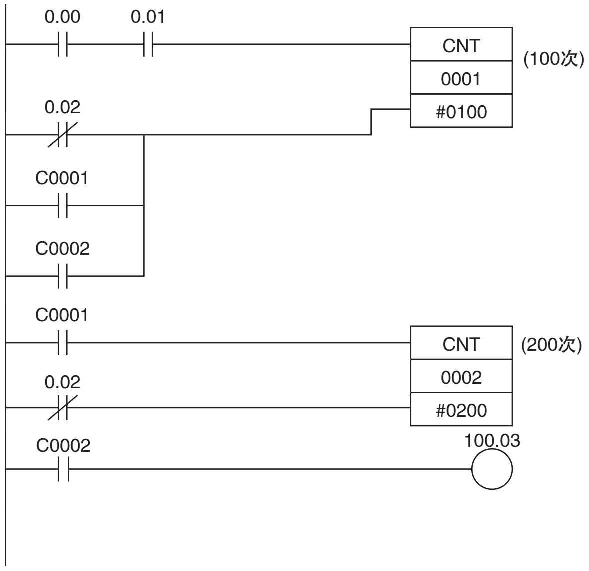
如果将第一循环标志 (A200.11) 与计数器的复位输入 (CIO 0.01) 进行逻辑或运算，则在程序开始执行时，计数器的 PV 将被复位为 SV(0700) 而非从 PV 前值开始重续计数。



指令	操作数
LD	0.00
AND	1s
LD NOT	0.01
CNT	1
	#0700
	C0001
	100.02

例 2：两级计数器

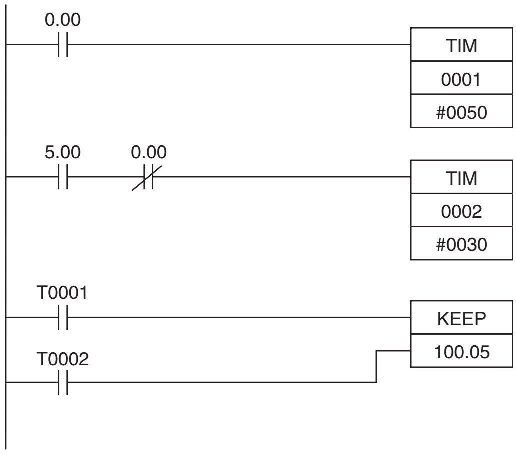
当需要超过 9999 的 SV 时，可组合两个计数器，如下例所示。在该例中，组合了两条 CNT 指令以生成一个 SV 为 20,000 的 BCD 计数器。



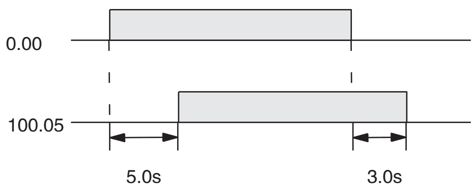
指令	操作数
LD	0.00
AND	0.01
LD NOT	0.02
OR	C0001
OR	C0002
CNT	1
	#0100
LD	C0001
LD NOT	0.02
CNT	2
	#0200
LD	C0002
OUT	100.03

例 3：ON/OFF 延迟

在该例中，通过 KEEP(011) 组合了两个 TIM 定时器，以产生一个 ON 延迟和一个 OFF 延迟。CIO 5.00 将在 CIO 0.00 变 ON 的 5.0 秒之后变 ON，并在 CIO 0.00 变 OFF 的 3.0 秒之后变 OFF。

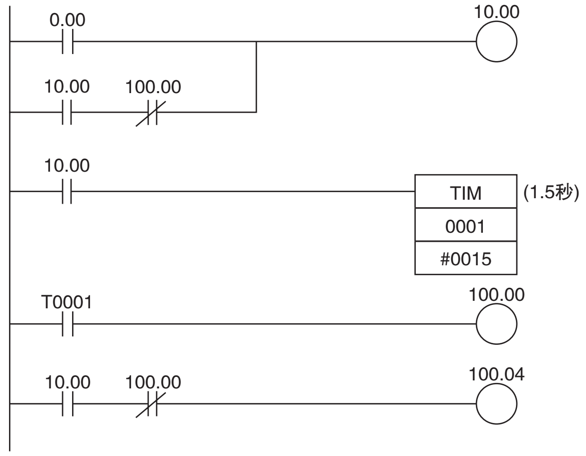


指令	操作数
LD	0.00
TIM	1
	#0050
LD	5.00
LD NOT	0.00
TIM	2
	#0030
LD	T0001
LD	T0002
KEEP(011)	100.05

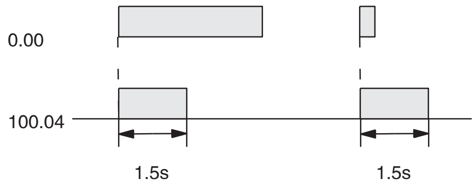


例 4：单触发位

可将 TIM 定时器与 OUT 或 OUT NOT 进行组合，以控制某个特定位保持 ON 或 OFF 的时间长度。在该例中，CIO 2.04 将在 CIO 0.00 变 ON 之后保持 ON 达 1.5 秒 (T0001 的 SV)。



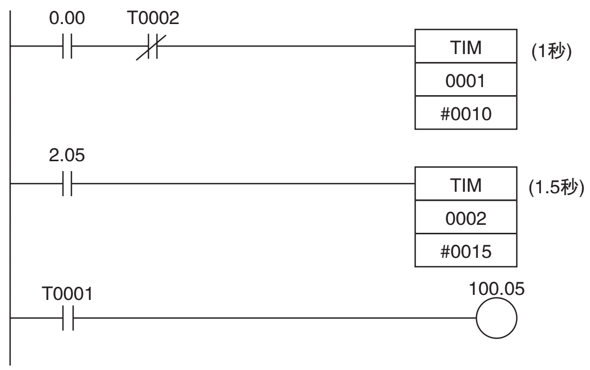
指令	操作数
LD	0.00
LD	10.00
AND NOT	100.00
OR LD	--
OUT	10.00
LD	10.00
TIM	1
	#0015
LD	T0001
OUT	100.00
LD	10.00
AND NOT	100.00
OUT	100.04



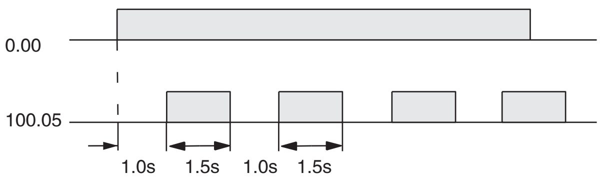
例 5：闪烁位

1) 两条 TIM 指令

可组合两个 TIM 定时器，以在执行条件为 ON 时生成以固定间隔变 ON 和 OFF 的位。在该例中，只要 CIO 0.00 为 ON，CIO 2.05 将保持 OFF 达 1.0 秒，然后再保持 ON 达 1.5 秒。

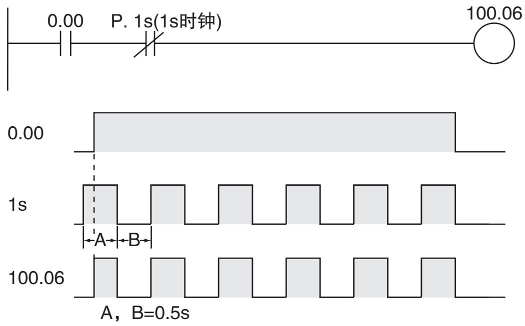


指令	操作数
LD	0.00
AND LD	T0002
TIM	1
	#0010
LD	2.05
TIM	2
	#0015
LD	T0001
OUT	100.05



2) 时钟脉冲

可将所需的执行条件与一个时钟脉冲进行组合，以模拟时钟脉冲 (0.1s、0.2s 或 1.0s)。



指令	操作数
LD	0.00
AND	1s
OUT	100.06

· 可使用内部时钟脉冲 (0.1s、0.2s、1s) 来简化闪烁电路的编程。

● 定时器的复位方法

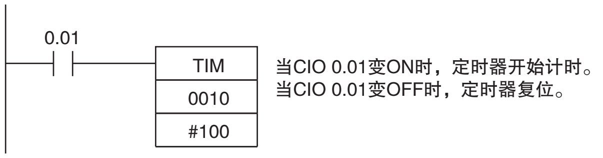
复位定时器指令的方法有两种。

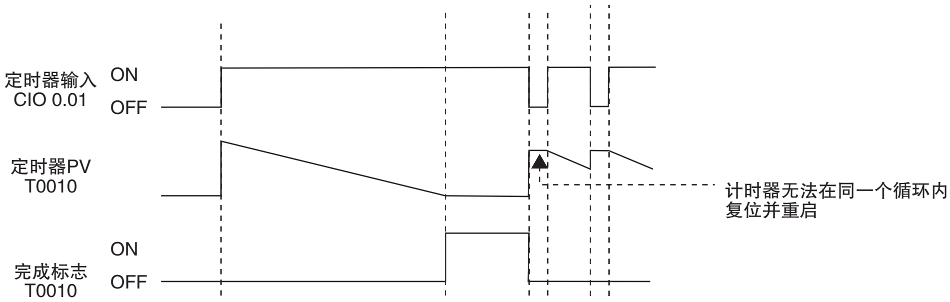
1. 使定时器指令的执行条件变 OFF

当执行条件变 OFF 时，定时器将被复位。

当执行条件变 ON 时，定时器将重新开始计时。

采用该方法时，无法在同一个循环中对定时器进行复位然后再重启。



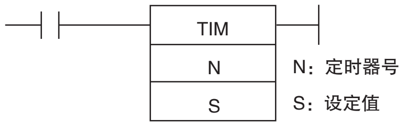
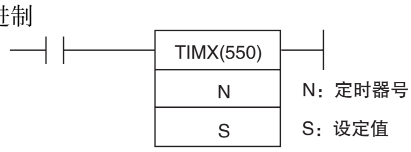


2. 使用复位定时器 / 计数器指令

执行复位定时器 / 计数器指令 (CNR/CNRX) 时，指定的指令将被复位。

TIM/TIMX

指令	助记符	变化	功能代码	功能
100ms 定时器	TIM/TIMX	---	550	TIM 或 TIMX(550) 定时器以 0.1s 为单位作减量计时。

符号	TIM	TIMX
	BCD 	二进制 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型		大小
		TIM	TIMX	
N	定时器号	TIMER	TIMER	1
S	设定值	WORD	UINT	1

N：定时器号

定时器号必须介于 0000 ~ 0255(十进制) 之间。

S：设定值 (以 100ms 为单位)

TIM(BCD)：#0000 ~ #9999

TIMX(二进制)：&0 ~ &65535(十进制) 或 #0000 ~ #FFFF(十六进制)

● 区域类型

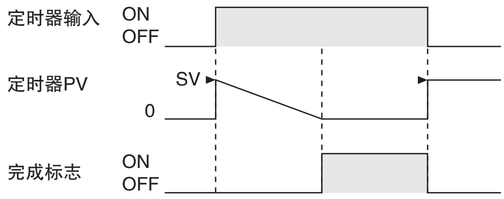
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	OK	---	---	---	---	---	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

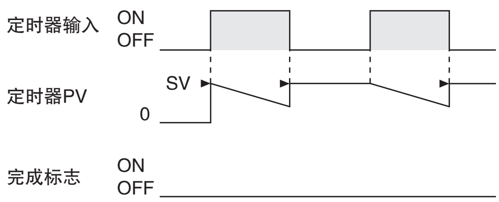
名称	标记	操作
出错标志	P_ER	· 如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 · 其它情况下 OFF。

功能

- 当定时器输入为 OFF 时，通过 N 指定的定时器被复位，即定时器的 PV 被复位为 SV 且完成标志变 OFF。
- 当定时器输入从OFF→ON时，TIM/TIMX(550) 开始使 PV 递减。只要定时器输入保持 ON，则 PV 将保持减量计时，且当 PV 到达 0 时，定时器的完成标志将变 ON。
- 定时器的PV和完成标志的状态在定时器计时完成后将保持。若要重启定时器，则定时器输入必须变 OFF 然后再变 ON，或者必须将定时器的 PV 变为一个非零值 (例如通过 MOV(021) 指令)。
- 对于 TIM，设定值 (SV) 的设定范围为 0 ～ 999.9s ；对于 TIMX(550)，设定范围为 0 ～ 6,553.5s。
- 定时器的精度为 –0.01 ～ 0s。



以下时序图所示为定时器计时完成之前，当定时器输入变 OFF 时的定时器 PV 和完成标志的状态。

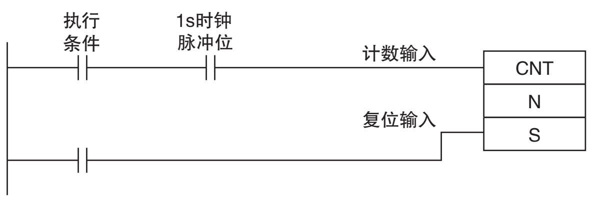


提示

- 根据所使用的定时器号而定，TIM/TIMX(550) 指令的 PV 和完成标志可通过以下方法刷新。

刷新定时	描述
TIM/TIMX(550) 执行时	· 每执行一次 TIM/TIMX(550) 指令均会更新 PV。 · 如果 PV 为 0，则完成标志变 ON。 如果 PV 不为 0，则完成标志变 OFF。

- 定时器通过电源中断被复位 (PV=SV，完成标志 OFF)，除非 IOM 保持位 (A500.12) 为 ON且该保持位在PLC设置中受保护。另外还可使用时钟脉冲位和计数器指令来编写一个在电源中断事件中将保持其PV的定时器，如下图所示。
- 如果定时器的设定值为 #0000，则在执行指令时将发生 “计时结束”错误。



注意事项

- 定时器号与其它定时器指令共享。如果两个定时器共享相同的定时器号但不同时使用该定时器号，则将在检查程序时报重复错误，但定时器将正常运行。共享相同定时器号的定时器若同时使用，则无法正常运行。
- 由于当 CPU 单元的循环时间超过 4s 时，定时器将无法正常运行，因此，请在循环时间不超过 4s 时使用定时器指令。
- 以下情况下，定时器将复位或暂停。(定时器复位时，其 PV 将被复位为 SV，且其完成标志将变 OFF。)

条件	PV	完成标志
当运行模式从 RUN 或 MONITOR 模式转换到 PROGRAM 模式或者反过来操作时 *1	0	OFF
电源断开和复位	0	OFF
执行 CNR(545)/CNRX(547)(复位定时器 / 计数器指令) 时 *2	BCD：9999 二进制：FFFF	OFF
在互锁的程序段 (IL(002)–ILC(003)) 中的处理	复位到 SV	OFF
在跳转的程序段 (JMP(004)–JME(005)) 中的处理	保持前状态	保持前状态

*1　如果 IOM 保持位 (A500.12) 已变 ON，则在运行模式改变时，定时器完成标志和 PV 的状态将保持。

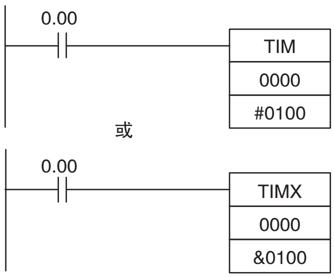
*2　执行 TIM/TIMX(550) 指令时，PV 将被设定为 SV。

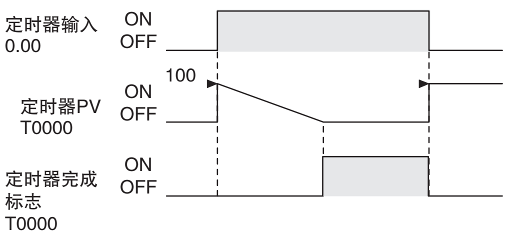
- 当 TIM/TIMX(550) 位于 IL(002) 和 ILC(003) 之间的程序段且该程序段被互锁时，则 PV 将被复位为 SV，且完成标志将变 OFF。
- 当在跳转的程序段 (JMP(004)、CJP(510)、JME(005)) 中运行 TIM/TIMX(550) 定时器时，定时器的 PV 将被刷新。
- 对 TIM/TIMX(550) 定时器进行强制置位时，定时器的完成标志将变 ON，且其 PV 将被置位为 0000。对 TIM/TIMX(550) 定时器进行强制复位后，定时器的完成标志将变 OFF，且其 PV 将被复位为 SV。
- 仅当执行 TIM/TIMX(550) 时对定时器的完成标志进行刷新，因此在定时器计时完成后使完成标志变 ON 可能最长需要一个循环的延迟。
- 如果采用在线编辑方式重写定时器指令，请务必使完成标志复位。 除非使完成标志复位，否则定时器将无法正常运行。

程序举例

下例中，当定时器输入 CIO 0.00 从 OFF → ON 时，定时器 PV 将从 SV 开始减量计时。当 PV 计到 0 时，定时器完成标志 T0000 将变 ON。

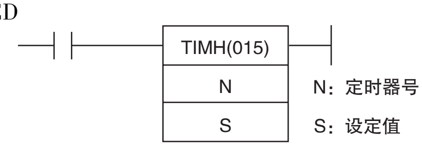
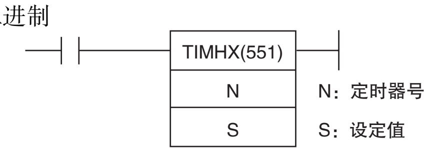
当 CIO 0.00 变 OFF 时，定时器 PV 将被复位为 SV，且完成标志将变 OFF。





TIMH/TIMHX

指令	助记符	变化	功能代码	功能
10ms 定时器	TIMH	---	015	TIMH(015)/TIMHX(551) 定时器以 10ms 为单位作减量计时。
	TIMHX	---	551	

符号	TIMH	TIMHX
	BCD 	二进制 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型		大小
		TIMH	TIMHX	
N	定时器号	TIMER	TIMER	1
S	设定值	WORD	UINT	1

N：定时器号

定时器号必须介于 0000 ~ 0255(十进制) 之间。

S：设定值

TIMH(BCD)：#0000 ~ #9999

TIMHX(二进制)：&0 ~ &65535(十进制) 或 #0000 ~ #FFFF(十六进制)

● 区域类型

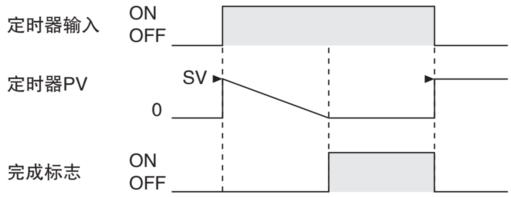
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	OK	---	---	---	---	---	---	---	OK	---	---	---
S	OK	OK	OK	OK		OK	OK	OK	OK	OK	OK					

条件标志

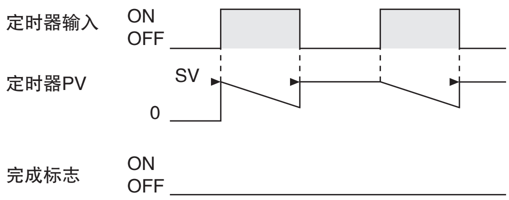
名称	标记	操作
出错标志	P_ER	・如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 ・其它情况下 OFF。

功能

- 当定时器输入为 OFF 时，通过 N 指定的定时器被复位，即定时器的 PV 被复位为 SV 且完成标志变 OFF。
- 当定时器输入从OFF→ON时，TIMH(015)/TIMHX(551)开始使 PV 递减。只要定时器输入保持 ON，则 PV 将保持减量计时，且当 PV 到达 0000 时，定时器的完成标志将变 ON。
- 定时器的 PV 和完成标志的状态在定时器计时完成后将保持。若要重启定时器，则定时器输入必须变 OFF 然后再变 ON，或者必须将定时器的 PV 变为一个非零值 (例如通过 MOV(021) 指令)。
- 对于 TIMH(015)，设定值 (SV) 的设定范围为 0 ～ 99.99s ；对于 TIMHX(551)，设定范围为 0 ～ 655.35s。
- 定时器的精度为 0 ～ 0.01s。



以下时序图所示为定时器计时完成之前，当定时器输入变 OFF 时的定时器 PV 和完成标志的状态。



提示

根据所使用的定时器号而定，TIMH(015)/TIMHX(551) 指令的 PV 和完成标志可通过以下方法刷新。

刷新定时	描述
TIMH(015)/TIMHX(551) 执行时	· 每执行一次 TIMH(015)/TIMHX(551) 指令均会更新 PV。 · 如果 PV 为 0，则完成标志变 ON。 如果 PV 不为 0，则完成标志变 OFF。

注意事项

- 定时器号与其它定时器指令共享。如果两个定时器共享相同的定时器号但不同时使用该定时器号，则将在检查程序时报重复错误，但定时器将正常运行。共享相同定时器号的定时器若同时使用，则无法正常运行。
- 由于当 CPU 单元的循环时间超过 4s 时，定时器将无法正常运行，因此，请在循环时间不超过 4s 时使用定时器指令。
- 以下情况下，定时器将复位或暂停。(定时器复位时，其 PV 将被复位为 SV，且其完成标志将变 OFF。)

条件	PV	完成标志
当运行模式从 RUN 或 MONITOR 模式转换到 PROGRAM 模式或者反过来操作时 *1	0	OFF
电源断开和复位	0	OFF
执行 CNR(545)/CNRX(547)(复位定时器 / 计数器指令) 时 *2	BCD：9999 二进制：FFFF	OFF
在互锁的程序段 (IL(002)–ILC(003)) 中的处理	复位到 SV	OFF
在跳转的程序段 (JMP(004)–JME(005)) 中的处理	保持前状态	保持前状态

*1 如果 IOM 保持位 (A500.12) 已变 ON，则在运行模式改变时，定时器完成标志和 PV 的状态将保持。

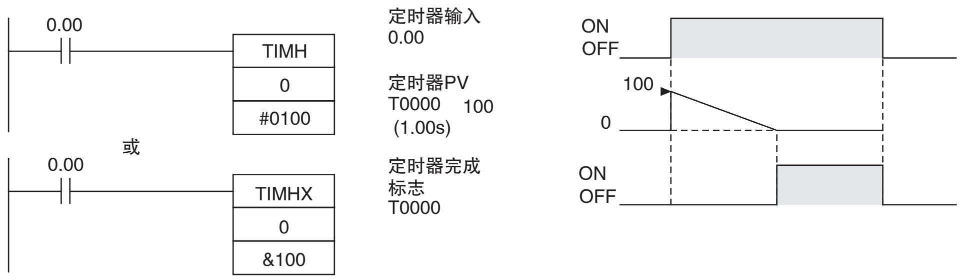
*2 执行 TIMH(015)/TIMHX(551) 指令时，PV 将被置位为 SV。

- 当在跳转的程序段 (JMP(004)、CJP(510)、JME(005)) 中运行 TIMH(015)/TIMHX(551) 定时器时，定时器的 PV 将被刷新。
- 当 TIMH(015)/TIMHX(551) 位于 IL(002) 和 ILC(003) 之间的程序段且该程序段被互锁时，则 PV 将被复位为 SV，且完成标志将变 OFF。
- 对 TIMH(015)/TIMHX(551) 定时器进行强制置位时，定时器的完成标志将变 ON，且其 PV 将被置位为 0000。对 TIMH(015)/TIMHX(551) 定时器进行强制复位后，定时器的完成标志将变 OFF，且其 PV 将被复位为 SV。
- 仅当执行 TIMH(015)/TIMHX(551) 时对定时器的完成标志进行刷新，因此在定时器计时完成后使完成标志变 ON 可能最长需要一个循环的延迟。
- 如果采用在线编辑方式重写定时器指令，请务必使完成标志复位。除非使完成标志复位，否则定时器将无法正常运行。

程序举例

下例中，当定时器输入 CIO 0.00 从 OFF→ON 时，定时器 PV 将从 SV 开始减量计时 (#0064=100=1.00s)。当 PV 计到 0 时，定时器完成标志 T0000 将变 ON。

当 CIO 0.00 变 OFF 时，定时器 PV 将被复位为 SV，且完成标志将变 OFF。



TMHH/TMHHX

指令	助记符	变化	功能代码	功能
1ms 定时器	TMHH	---	540	TMHH(540)/TMHHX(552) 定时器以 1ms 为单位作减量计时。
	TMHHX	---	552	

符号	TMHH	TMHHX
	BCD TMHH(540) N　N：定时器号 S　S：设定值	二进制 TMHHX(552) N　N：定时器号 S　S：设定值

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	OK	OK	---

操作数

操作数	描述	数据类型		大小
		TMHH	TMHHX	
N	定时器号	TIMER	TIMER	1
S	设定值	WORD	UINT	1

N：定时器号

定时器编号必须介于 0000 ~ 0015(十进制) 之间。

S：设定值

TMHH(BCD)：#0000 ~ #9999

TMHHX(二进制)：&0 ~ &65535(十进制) 或 #0000 ~ #FFFF(十六进制)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	OK	---	---	---	---	---	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 · 其它情况下 OFF。

功能

- 当定时器输入为 OFF 时，通过 N 指定的定时器被复位，即定时器的 PV 被复位为 SV 且完成标志变 OFF。
- 当定时器输入从 OFF → ON 时，TMHH(540)/TMHHX(552) 开始使 PV 递减。只要定时器输入保持 ON，则 PV 将保持减量计时，且当 PV 到达 0000 时，定时器的完成标志将变 ON。

- 定时器的 PV 和完成标志的状态在定时器计时完成后将保持。若要重启定时器，则定时器输入必须变 OFF 然后再变 ON，或者必须将定时器的 PV 变为一个非零值 (例如通过 MOV(021) 指令)。
- 对于TMHH(540)，设定值(SV)的设定范围为0 ~ 9.999s；对于TMHHX(552)，设定范围为0 ~ 65.535s。
- 定时器的精度为 –0.001 ～ 0s。

提示

在 TMHH/TMHHX 指令中使用的定时器 PV 和 “计时结束”在以下定时进行刷新。

刷新定时	描述
执行各条指令时	· 每执行一次指令均会更新 PV。 · 当 PV 为 0 时，计时结束标志为 ON ；否则为 OFF。

注意事项

- 定时器号与其它定时器指令共享。如果两个定时器共享相同的定时器号但不同时使用该定时器号，则将在检查程序时报重复错误，但定时器将正常运行。共享相同定时器号的定时器若同时使用，则无法正常运行。
- 仅当执行 TMHH(540)/TMHHX(552) 指令时更新完成标志。因此完成标志与实际设定值相比最多可延迟一个循环的时间。
- 即使任务处于待机状态也不刷新定时器的当前值。
- 以下情况下，定时器将复位或暂停。(定时器复位时，其 PV 将被复位为 SV，且其完成标志将变 OFF。)

条件	PV	完成标志
当运行模式从 RUN 或 MONITOR 模式转换到 PROGRAM 模式或者反过来操作时 *1	0	OFF
电源中断和复位	0	OFF
执行 CNR(545)/CNRX(547)(复位定时器 / 计数器指令) 时 *2	BCD：9999 二进制：FFFF	OFF
在互锁的程序段 (IL(002)–ILC(003)) 中的处理	复位到 SV	OFF
在跳转的程序段 (JMP(004)–JME(005)) 中的处理	保持前状态	保持前状态

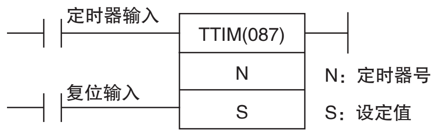
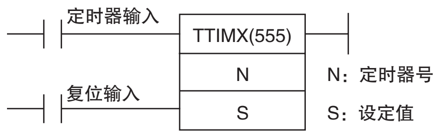
*1 如果 IOM 保持位 (A500.12) 已变 ON，则在运行模式改变时，定时器完成标志和 PV 的状态将保持。

*2 执行 TMHH(540)/TMHHX(552) 指令时，PV 将被设定为 SV。

- 即使所有运行的定时器位于被 JMP(004)、CJP(510)、JME(005) 指令跳转的程序段中，其当前值也将不被刷新。
- 当 TMHH(540)/TMHHX(552) 位于 IL(002) 和 ILC(003) 之间的程序段且该程序段被互锁时，则 PV 将被复位为 SV，且完成标志将变 OFF。
- 对 TMHH(540)/TMHHX(552) 定时器进行强制置位时，定时器的完成标志将变 ON，且其 PV 将被置位为 0。对 TMHH(540)/TMHHX(552) 定时器进行强制复位后，定时器的完成标志将变 OFF，且其 PV 将被复位为 SV。
- 如果采用在线编辑方式重写定时器指令，请务必使完成标志复位。 除非使完成标志复位，否则定时器将无法正常运行。

TTIM/TTIMX

指令	助记符	变化	功能代码	功能
累加定时器	TTIM	---	087	TTIM(087)/TTIMX(555) 定时器以 0.1s 为单位作增量计时。
	TTIMX	---	555	

	TTIM	TTIMX
符号	BCD 	二进制 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型		大小
		TTIM	TTIMX	
N	定时器号	TIMER	TIMER	1
S	设定值	WORD	UINT	1

N：定时器号

定时器号必须介于 0000 ～ 0255(十进制) 之间。

S：设定值

TTIM(BCD)：#0000 ～ #9999

TTIMX (二进制)：&0 ～ &65535(十进制) 或 #0000 ～ #FFFF(十六进制)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	OK	---	---	---	---	---	---	---	OK	---	---	---
S	OK	OK	OK	OK		OK	OK	OK	OK	OK	OK					

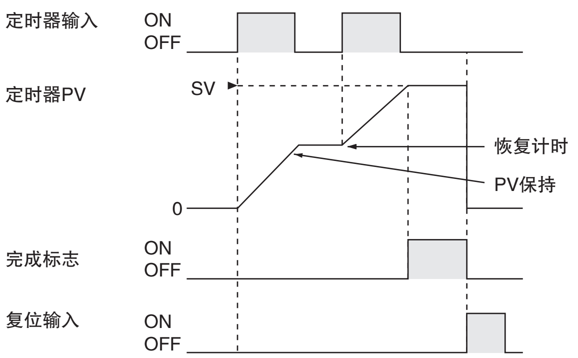
条件标志

名称	标记	操作
出错标志	P_ER	· 如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 · 其它情况下 OFF。

功能

・当定时器输入为 ON 时，TTIM(087)/TTIMX(555) 使 PV 递增。当定时器输入变 OFF 时，定时器将停止使 PV 递增，但 PV 的值将保持。当定时器输入再次变 ON 时，PV 将重续计时。当 PV 到达 SV 时，定时器完成标志将变 ON。

・定时器的 PV 和完成标志的状态在定时器计时完成后将保持。重启定时器的方法有三种：将定时器的 PV 改为非零值 (例如通过 MOV(021) 指令)、使复位输入变 ON，或者执行 CNR(545)/CNRX(547) 指令。



・对于 TTIM(087)，设定值 (SV) 的设定范围为 0 ～ 999.9s；对于 TTIMX(555)，设定范围为 0 ～ 6,553.5s。

・定时器的精度为 0 ～ 0.01s。

提示

・TIM/TIMX(550)之类的典型定时器使计数器减量计数，其PV表示到定时器计时结束前剩余的时间。而 TTIM(087)/TTIMX(555) 的 PV 则表示已经过的时间，因此其 PV 可在许多计算和显示输出中使用而不会改变。

注意事项

・定时器号与其它定时器指令共享。如果两个定时器共享相同的定时器号但不同时使用该定时器号，则将在检查程序时报重复错误，但定时器将正常运行。共享相同定时器号的定时器若同时使用，则无法正常运行。

・以下情况下，定时器将复位或暂停。(当使 TTIM(087)/TTIMX(555) 定时器复位时，其 PV 将被复位为 0，且其完成标志将变 OFF。)

条件	PV	完成标志
当运行模式从 RUN 或 MONITOR 模式转换到 PROGRAM 模式或者反过来操作时 *1	0	OFF
电源中断和复位	0	OFF
执行 CNR(545)/CNRX(547)(复位定时器 / 计数器指令) 时 *2	BCD：9999 二进制：FFFF	OFF
在互锁的程序段 (IL(002)–ILC(003)) 中的处理	保持前状态	保持前状态
在跳转的程序段 (JMP(004)–JME(005)) 中的处理	保持前状态	保持前状态

*1　如果 IOM 保持位 (A500.12) 已变 ON，则在运行模式改变时，定时器完成标志和 PV 的状态将保持。

*2　执行 TTIM(087)/TTIMX(555) 指令时，PV 将被设定为 SV。

・当 TTIM(087)/TTIMX(555) 位于 IL(002) 和 ILC(003) 之间的程序段且该程序段被互锁时，则 PV 将保持前值 (不会被复位)。当将 TTIM(087)/TTIMX(555) 编入 IL(002) 和 ILC(003) 之间的程序段时，请考虑上述情况。

・当正在运行的 TTIM(087)/TTIMX(555) 位于 JMP(004) 和 JME(005) 之间的程序段且该程序段被跳转时，则 PV 将保持前值。当将 TTIM(087)/TTIMX(555) 编入 JMP(004) 和 JME(005) 之间的程序段时，请考虑上述情况。

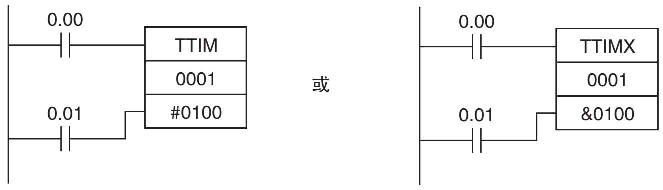
- 对 TTIM(087)/TTIMX(555) 定时器进行强制置位后，定时器的完成标志将变 ON，且其 PV 将被复位为 0。对 TTIM(087)/TTIMX(555) 定时器进行强制复位后，定时器的完成标志将变 OFF，且其 PV 将被复位为 0。强制置位和强制复位操作的优先级高于定时器和复位输入的状态。
- 仅当执行 TTIM(087)/TTIMX(555) 指令时，定时器的 PV 被刷新。因此当循环时间超过 100ms 时，定时器无法正常运行，因为定时器的递增单位为 100ms。
- 仅当执行 TTIM(087)/TTIMX(555) 时对定时器的完成标志进行刷新，因此在定时器计时完成后使完成标志变 ON 可能最长需要一个循环的延迟。

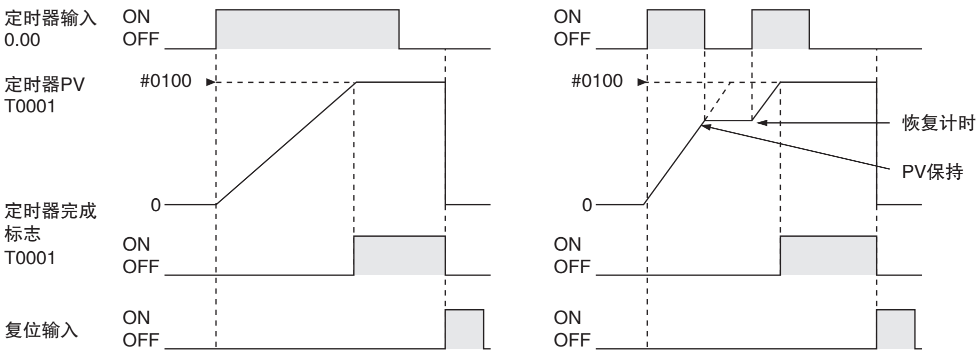
程序举例

下例中，当定时器输入 CIO 0.00 为 ON 时，定时器 PV 将从 0 开始增量计数。当 PV 到达 SV 时，定时器完成标志 T0001 将变 ON。

如果复位输入变 ON，则定时器 PV 将被复位为 0，且完成标志 (T0001) 将变 OFF。(通常会使复位输入变 ON 以使定时器复位，然后使定时器输入变 ON 以开始计时。)

如果在到达 SV 之前定时器输入变 OFF，则定时器将停止计时，但 PV 将保持。当定时器再次变 ON 时，定时器将从其前 PV 开始重续计时。

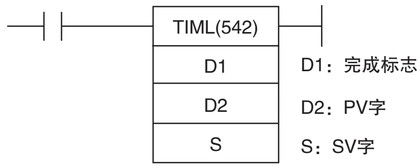


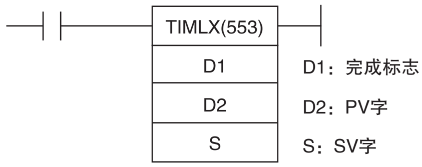


TIML/TIMLX

指令	助记符	变化	功能代码	功能
长定时器	TIML	---	542	TIML(542)/TIMLX(553) 定时器以 0.1s 为单位作减量计时。
	TIMLX	---	553	

	TIML	TIMLX
符号	BCD	二进制





适用程序区

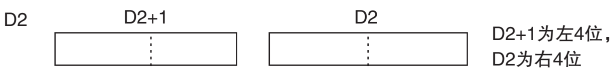
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

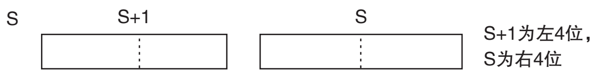
操作数	描述	数据类型		大小
		TIML	TIMLX	
D1	完成标志	WORD	UINT	1
D2	PV 字	DWORD	UDINT	2
S	SV 字	DWORD	UDINT	2

D1：完成标志

D2：PV 字



S：SV 字



对于 TIML(542) 指令，PV 和 SV 的范围可从 #00000000 ~ #99999999；对于 TIMLX(553) 指令，范围可从 &00000000 ~ &4294967294(十进制)或#00000000 ~ #FFFFFFFF(十六进制)。

注 S、S+1、D2 和 D2+1 必须位于同一个数据区。

● 区域类型

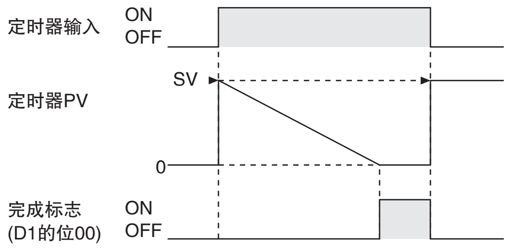
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
D1，D2	OK	OK	OK	OK	---	---	OK	OK	OK	---	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・如果处于 BCD 模式且 D2 中不含 BCD 数据时，该标志为 ON。 ・如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 ・其它情况下 OFF。

功能

- 当定时器输入为 OFF 时，定时器被复位，即定时器的 PV 被复位为 SV 且完成标志变 OFF。
- 当定时器输入从OFF→ON时，TIML(542)/TIMLX(553)开始使 D2+1 和 D2 中的 PV 递减。只要定时器输入保持 ON，则 PV 将保持减量计时，且当 PV 到达 0 时，定时器的完成标志将变 ON。
- 定时器的PV和完成标志的状态在定时器计时完成后将保持。若要重启定时器，则定时器输入必须变OFF 然后再变ON，或者必须将定时器的 PV 变为一个非零值 (例如通过 MOV(021) 指令)。



- TIML(542)/TIMLX(553) 指令中，TIML(542) 最多可计时 115 天，而 TIMLX(553) 最多可计时 4,971 天。
- 定时器的精度为 0 ~ 0.01s。

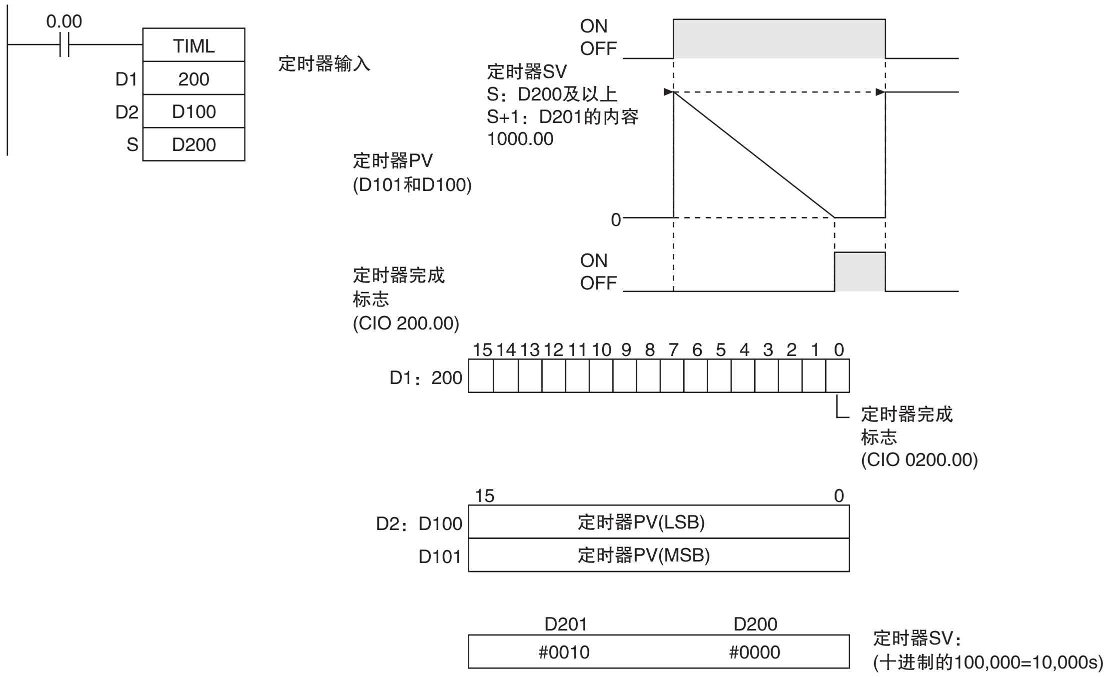
注意事项

- 与大多数定时器不同，TIML(542)/TIMLX(553) 不使用定时器号。(对于 TIML(542)/TIMLX(553) 指令，不执行定时器区 PV 的刷新。)
- 由于 TIML(542)/TIMLX(553) 的完成标志位于数据区中，因此可对其像其它位一样进行强制置位或强制复位，但 PV 不变。
- 仅当执行 TIML(542)/TIMLX(553) 指令时，定时器的 PV 被刷新。因此当循环时间超过 100ms 时，定时器无法正常运行，因为定时器的递增单位为 100ms。
- 仅当执行 TIML(542)/TIMLX(553) 时对定时器的完成标志进行刷新，因此在定时器计时完成后使完成标志变 ON 可能最长需要一个循环的延迟。
- 当 TIML(542)/TIMLX(553) 位于 IL(002) 和 ILC(003) 之间的程序段且该程序段被互锁时，则 PV 将被复位为 SV，且完成标志将变 OFF。
- 当正在运行的 TIML(542)/TIMLX(553) 位于 JMP(004) 和 JME(005) 之间的程序段且该程序段被跳转时，则 PV 将保持前值。当将 TIML(542)/TIMLX(553) 编入 JMP(004) 和 JME(005) 之间的程序段时，请考虑上述情况。
- 请务必确保为完成标志和 PV(D1、D2 和 D2+1) 指定的字不用于其它指令中。如果这些字受到其它指令的影响，则定时器可能无法正常结束计时。

程序举例

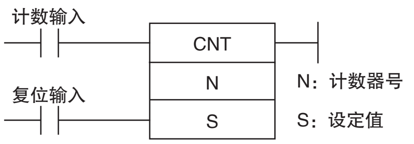
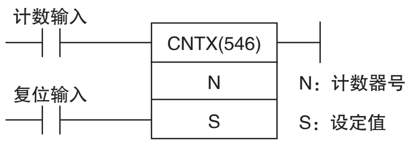
下例中，当定时器输入 CIO 0.00 为 ON 时，定时器 PV(在 D201 和 D200 中) 将被置位为 SV(在 D101 和 D100 中)，且 PV 将开始减量计时。当 PV 计到 0 时，定时器完成标志 (CIO 200.00) 将变 ON。

当 CIO 0.00 变 OFF 时，定时器 PV 将被复位为 SV，且完成标志将变 OFF。



CNT/CNTX

指令	助记符	变化	功能代码	功能
计数器	CNT/CNTX	---	546	CNT/CNTX(546) 计数器作减量计数。

<table>
<tr><th rowspan="3">符号</th><th>CNT</th><th>CNTX</th></tr>
<tr><td>BCD</td><td>二进制</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th rowspan="2">大小</th></tr>
<tr><th>CNT</th><th>CNTX</th></tr>
<tr><td>N</td><td>计数器号</td><td>COUNTER</td><td>COUNTER</td><td>1</td></tr>
<tr><td>S</td><td>设定值</td><td>WORD</td><td>UINT</td><td>1</td></tr>
</table>

N：计数器号

计数器号必须介于 0000 ～ 0255(十进制) 之间。

S：设定值

CNT(BCD)：#0000 ～ #9999

CNTX(二进制)：&0 ～ &65535(十进制) 或 #0000 ～ #FFFF(十六进制)

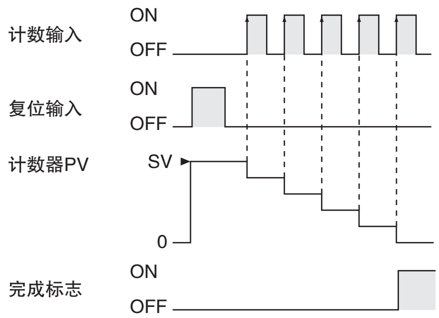
● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>N</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td rowspan="2">OK</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">OK</td><td>---</td><td>---</td><td rowspan="2">---</td></tr>
<tr><td>S</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td></tr>
</table>

条件标志

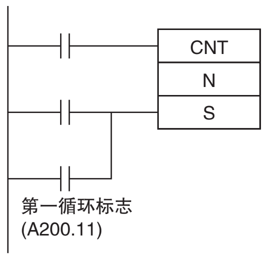
名称	标记	操作
出错标志	P_ER	· 如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 · 其它情况下 OFF。

功能



- 每当计数输入从 OFF → ON 时，其 PV 均递减 1。当 PV 计到 0 时，完成标志变 ON。
- 一旦完成标志变 ON，应通过将复位输入变 ON 或者使用 CNR(545)/CNRX(547) 指令的方法使计数器复位。否则，计数器无法重新启动。
- 当复位输入为 ON 时，计数器被复位且计数输入被忽略。(当计数器被复位时，其 PV 将被复位为 SV，且完成标志将变 OFF。)
- 对于 CNT，设定范围为 0 ～ 9,999 ；对于 CNTX(546)，设定范围为 0 ～ 65,535。

提示



- 即使发生电源中断，计数器 PV 也将保留。如果想要从 SV 而非所保留的 PV 开始重新计数，可将第一循环标志 (A200.11) 作为复位输入添加到计数器中。

注 1 若 CP1E CPU 被备份至电容器中且电源保持 OFF 超过下述时间，则计数器的 PV 和完成标志将不固定。
E 型 CPU 单元
9 小时 (60 ℃)，50 小时 (25 ℃)
N/NA 型 CPU 单元
7 小时 (60 ℃)，40 小时 (25 ℃)

2 在 PLC 设置中设定“清除保持存储器”后，计数器的 PV 值和完成标志将在电源每次置 ON 时被清空。此时，DM 区 (D) 和保持区 (H) 将同时被清空。

3 CP1E N/NA 型 CPU 单元 (CP1E-N/NA □□ D □ - □) 可安装电池。安装电池后，计数器的 PV 和完成标志可在电源置 OFF 时进行保持。

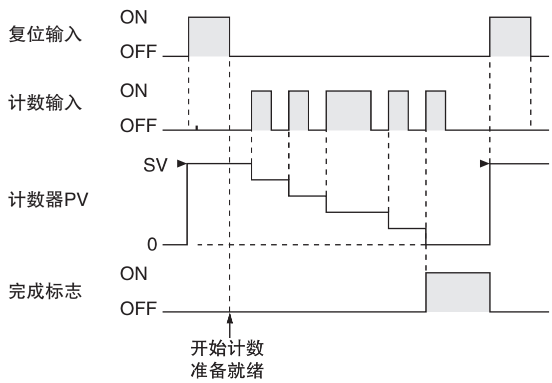
4 CP2E CPU 单元在未安装电池的情况下，即使电源置 OFF，计数器的 PV 和完成标志仍将保持。

注意事项

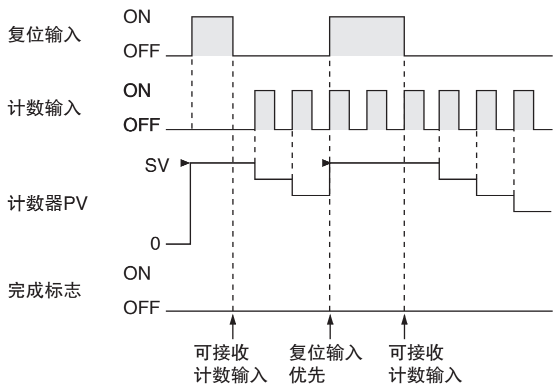
- 计数器号由 CNT、CNTX(546)、CNTR(012) 和 CNTRX(548) 指令共享。如果两个计数器共享相同的计数器号但不同时使用，则将在检查程序时报重复错误，但计数器将正常运行。共享相同计数器号的计数器若同时使用，则无法正常运行。

- 当计数输入从 OFF → ON 时计数器的 PV 将被刷新，而每次执行 CNT/CNTX(546) 时，完成标志将被刷新。如果 PV 为 0，则完成标志将变 ON ；如果 PV 不为 0，则完成标志将变 OFF。

- 对 CNT/CNTX(546) 计数器进行强制置位时，计数器的完成标志将变 ON，且其 PV 将被复位为 0000。对 CNT/CNTX(546) 计数器进行强制复位后，计数器的完成标志将变 OFF，且其 PV 将被置位为 SV。
- 在通过计数输入开始计数时，请务必使复位输入从 OFF → ON → OFF，从而使计数器复位，如下图所示。如果复位输入为 ON，则无法接收计数输入。



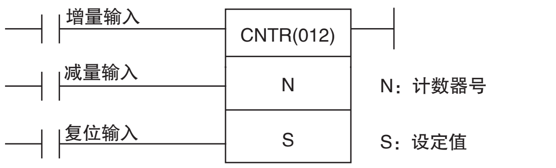
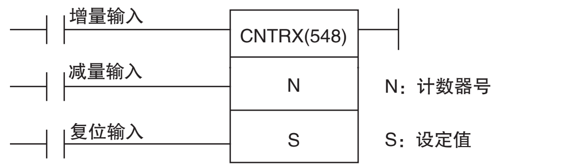
- 如果复位输入和计数输入同时为 ON，则复位输入的优先级更高，且计数器将被复位。(PV 将被复位为 SV，且完成标志将变 OFF。)



- 如果采用在线编辑来添加计数器，则计数器必须复位后才可正常运行。如果计数器不进行复位，则计数器的前值将作为当前值 (PV) 使用，且计数器被写入后可能无法正常运行。

CNTR/CNTRX

指令	助记符	变化	功能代码	功能
可逆计数器	CNTR	---	012	---
	CNTRX	---	548	

	CNTR	CNTRX
符号	BCD 	二进制 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小
		CNTR	CNTRX	
N	计数器号	COUNTER	COUNTER	1
S	设定值	WORD	UINT	1

N：计数器号

计数器号必须介于 0000 ~ 0255(十进制) 之间。

S：设定值

CNTR(BCD)：#0000 ~ #9999
CNTRX(二进制)：&0 ~ &65535(十进制) 或 #0000 ~ #FFFF(十六进制)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	---	OK	---	---	---	---	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK		OK	OK	OK	OK	OK					

条件标志

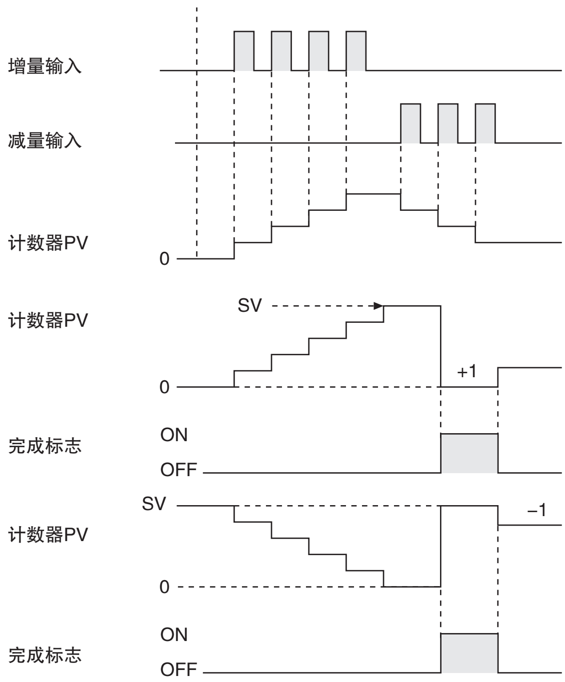
名称	标记	操作
出错标志	P_ER	· 如果处于 BCD 模式且 S 中不含 BCD 数据时，该标志为 ON。 · 其它情况下 OFF。

功能

每当增量输入从 OFF → ON 时，计数器 PV 将递增 1；而每当减量输入从 OFF → ON 时，计数器PV将递减1。PV可在0 ~ SV之间波动。

递增的情况下，当 PV 从 SV 变回 0 时，完成标志将变 ON；而当 PV 从 0 递增到 1 时，完成标志又将再次变 OFF。

递减的情况下，当 PV 从 0 递减为 SV 时，完成标志将变 ON；而当 PV 从 SV 递减到 SV−1 时，完成标志又将再次变 OFF。



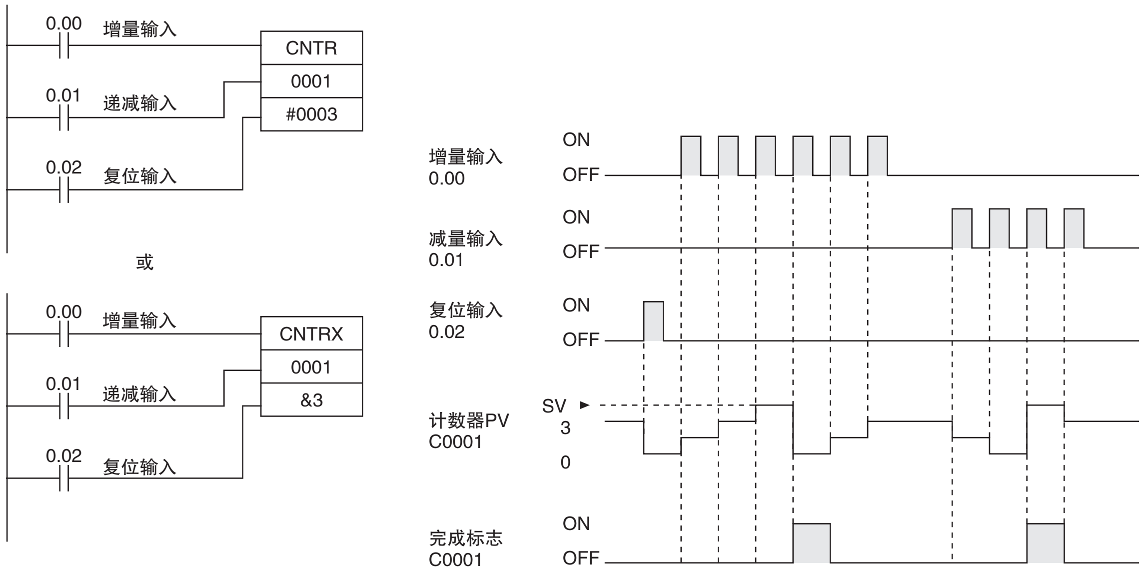
注意事项

- 计数器号由 CNT、CNTX(546)、CNTR(012) 和 CNTRX(548) 指令共享。如果两个计数器共享相同的计数器号但不同时使用，则将在检查程序时报重复错误，但计数器将正常运行。共享相同计数器号的计数器若同时使用，则无法正常运行。
- 如果增量和减量输入同时从 OFF → ON，则 PV 不变。如果复位输入为 ON，则 PV 将被复位为 0，且这两个计数输入均会被忽略。
- 仅当 PV 从 SV 递增到 0 或者从 0 递减到 SV 时，完成标志将为 ON；在其它所有情况下则为 OFF。
- 当通过助记符输入 CNTR(012)/CNTRX(548) 指令时，应首先输入增量输入 (II)，然后输入减量输入 (DI) 和复位输入 (R)，最后再输入 CNTR(012)/CNTRX(548) 指令。当通过梯形图输入时，应首先输入增量输入(II)，然后输入CNTR(012)/CNTRX(548)指令，再输入减量输入(DI)，最后输入复位输入(R)。

程序举例

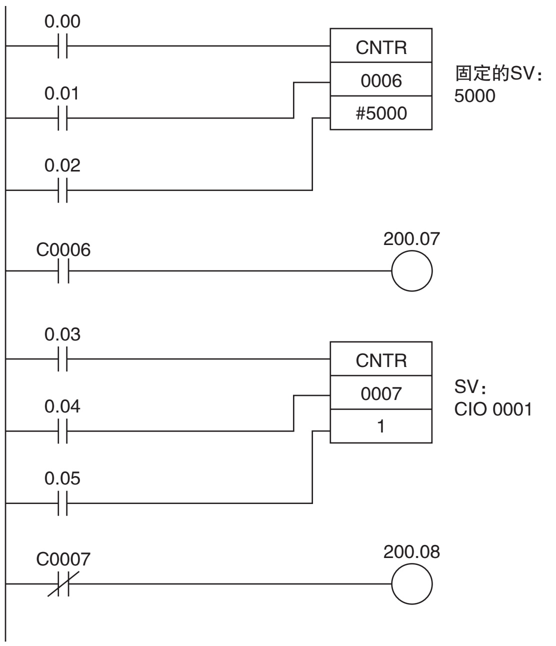
通过使复位输入 (CIO 0.02) 变 ON 和变 OFF 来将计数器 PV 复位为 0。每当增量输入 (CIO 0.00) 从 OFF → ON 时，其 PV 均递增 1。当 PV 从 SV(3) 递增时，将被自动复位为 0，且完成标志将变 ON。

与此类似，每当减量输入 (CIO 0.01) 从 OFF → ON 时，其 PV 均递减 1。当 PV 从 0 递减时，将被自动置位为 SV(3)，且完成标志将变 ON。

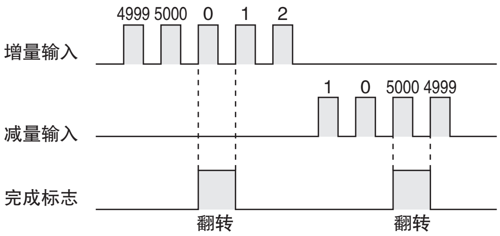


当信号上升 (OFF → ON) 时，加法和减法计数输入会使计数增大 / 减小一次。当上述两个输入同时变 ON 时，既不增大也不减小计数。如果复位输入变 ON，则 PV 将变为 0，且不接受计数输入。

下例中，CNTR(012) 0007 的 SV 由 CIO 0001 的内容决定。当通过外部开关来控制 CIO 0001 的内容时，则可从该开关手动更改设定值。

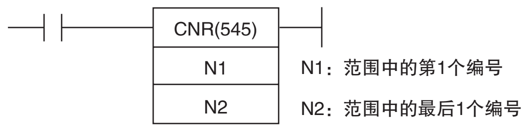
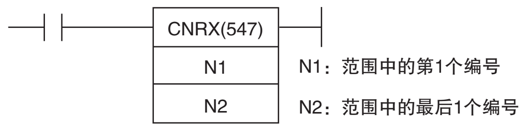


指令	操作数
LD	0.00
LD	0.01
LD	0.02
CNTR(012)	0006
	#5000
LD	200.07
OUT	0.03
LD	0.04
LD	0.05
LD	0007
CNTR(012)	1
LD NOT	C0007
OUT	200.08



CNR/CNRX

指令	助记符	变化	功能代码	功能
复位定时器 / 计数器	CNR	@CNR	545	使指定的定时器或计数器号范围内的定时器或计数器复位。
	CNRX	@CNRX	547	

	CNR	CNRX
符号	BCD 	二进制 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N1	范围中的第 1 个编号	TIMER/COUNTER*1	可变
N2	范围中的最后 1 个编号	TIMER/COUNTER*1	可变

* 仅 N1 和 N2 为相同变量时有效。

N1：范围中的第 1 个编号

N1 必须为介于 T000 ～ T255 之间的定时器号或者介于 C000 ～ C255 之间的计数器号。

N2：范围中的最后 1 个编号

N2 必须为介于 T000 ～ T255 之间的定时器号或者介于 C000 ～ C255 之间的计数器号。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N1，N2	---	---	---	---	OK	OK	---	---	---	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当 N1 和 N2 不在同一数据区时为 ON。 · 其它情况下 OFF。

功能

CNR(545)/CNRX(547) 指令使从 N1 ～ N2 的所有定时器或计数器的完成标志复位。与此同时，PV 将被全部置为最大值 (对于 BCD 为 9999，对于二进制为 FFFF)。(下次执行定时器或计数器指令时，PV 将被置为 SV。)

注意事项

・定时器 / 计数器的复位操作如下表所示。

	指令复位		CNR(545) 的运行
BCD	TIM： TIMH(015)： TMHH(540)： TTIM(087)： CNT： CNTR(012)：	100ms 定时器 10ms 定时器 1ms 定时器 累加定时器 计数器 可逆计数器	PV 被置为其最大值 (9,999 BCD) 且完成标志变 OFF。

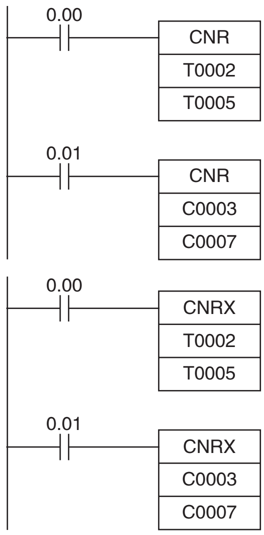
	指令复位		CNRX(547) 的运行
二进制	TIMX(550)： TIMHX(551)： TMHHX(552)： TTIMX(555)： CNTX(546)： CNTRX(548)：	100ms 定时器 10ms 定时器 1ms 定时器 累加定时器 计数器 可逆计数器	PV 被置为其最大值 (FFFF Hex) 且完成标志变 OFF。

・因为 TIML(542) 和 TIMLX(553) 定时器不采用定时器号，所以 CNR(545)/CNRX(547) 指令不会对其进行复位。

・CNR(545)/CNRX(547) 指令不会使定时器 / 计数器指令本身复位，而只是使分配给这些指令的 PV 和完成标志复位。在大多数情况下，执行 CNR(545)/CNRX(547) 的效果与直接使指令复位的效果不同。例如，当直接使TIM/TIMX(550)指令复位时，其PV将被置为SV；但当通过CNR(545)/CNRX(547) 指令使其复位时，其 PV 将被置为最大值 (对于 BCD 为 9999 ；对于二进制为 FFFF)。

・当被指定的 N1 和 N2 之间的关系为 N1>N2 时，仅该定时器 / 计数器号的完成标志被复位。

程序举例



本例中，当CIO 0.00为ON时，定时器T0002～T0005的完成标志将变OFF，且定时器的PV将被置为最大值(对BCD为9999)。

当CIO 0.01为ON时，计数器C0003～C0007的完成标志将变OFF，且计数器的PV将被置为最大值(对BCD为9999)。

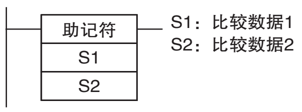
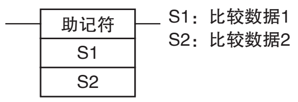
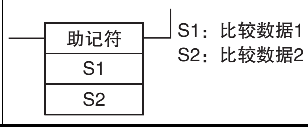
本例中，当CIO 0.00为ON时，定时器T0002～T0005的完成标志将变OFF，且定时器的PV将被置为最大值(对二进制为FFFF)。

当CIO 0.01为ON时，计数器C0003～C0007的完成标志将变OFF，且计数器的PV将被置为最大值(对二进制为FFFF)。

比较指令

=，<>，<，<=，>，>=

指令	助记符	变化	功能代码	功能
输入比较指令	=，<>，<，<=，>，>=	---	300 ~ 328	输入比较指令比较两个值 (常数和 / 或指定字的内容)，并在比较条件为真时生成一个 ON 执行条件。 输入比较指令可用来比较单字或双字长带符号或无符号数据。

符号	=, <>, <, <=, >, >=		
	LD 连接 	AND 连接 	OR 连接 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型				大小	
		无符号	无符号双字长	带符号	带符号双字长	单字	双字
S1	比较数据 1	UINT	UDINT	INT	DINT	1	2
S2	比较数据 2	UINT	UDINT	INT	DINT	1	2

● 区域类型

区域			字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
			CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
单字	无符号	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
单字	带符号	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
双字	无符号	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	OK	---	---	---
双字	带符号	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---

条件标志

名称	标记	操作	
		数据长度：单字	数据长度：双字
出错标志	P_ER	OFF 或不变	OFF 或不变
大于标志	P_GT	· 单字数据 $S_1>S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1>S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	· 单字数据 $S_1 \geqslant S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1 \geqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 单字数据 $S_1=S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1=S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	· 单字数据 $S_1 \neq S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1 \neq S_2+1$，S_2 时 ON。 · 其它情况下 OFF。

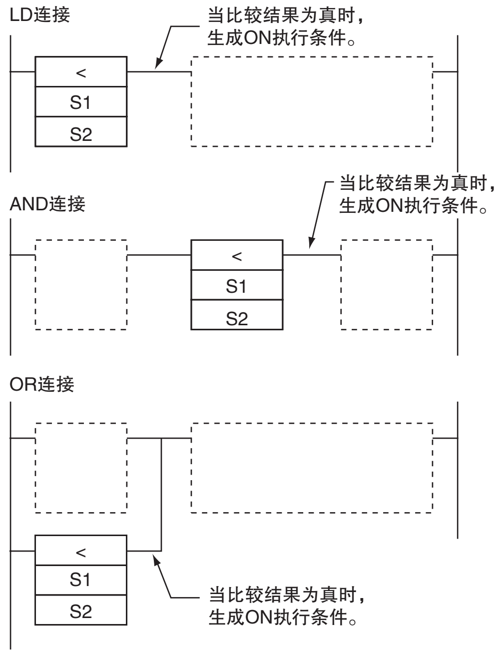
名称	标记	操作	
		数据长度：单字	数据长度：双字
小于标志	P_LT	· 单字数据 $S_1 < S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1 < S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
小于等于标志	P_LE	· 单字数据 $S_1 \leqslant S_2$ 时 ON。 · 其它情况下 OFF。	· 双字数据 S_1+1，$S_1 \leqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
负标志	P_N	OFF 或不变	OFF 或不变

功能

输入比较指令将 S1 和 S2 当作带符号或无符号值进行比较，并在比较条件为真时生成一个 ON 执行条件。

对输入比较指令的处理类似于 LD、AND 和 OR 指令，用于控制随后的指令执行。

输入类型	操作
LD	该指令可直接连接到左边的母线。
AND	该指令不能直接连接到左边的母线。
OR	该指令可直接连接到左边的母线。



● 选项

输入比较指令可比较带符号或无符号数据，还可比较单字或双字值。如果未指定选项，则作为单字无符号数据比较。指令有三种输入方式和两个选项，共有 72 个不同的输入比较指令。

符号	选项 (数据格式)	选项 (数据长度)
= （等于） < >(不等于) < （小于） <= (小于等于) > （大于） >= (大于等于)	无：无符号数据 S：带符号数据	无：单字数据 L：双字数据

功能	助记符	名称	代码
C1 = C2 时为真	LD/AND/OR =	等于	300
	LD/AND/OR =L	双字等于	301
	LD/AND/OR =S	带符号等于	302
	LD/AND/OR =SL	双字带符号等于	303
C1 ≠ C2 时为真	LD/AND/OR <>	不等于	305
	LD/AND/OR <>L	双字不等于	306
	LD/AND/OR <>S	带符号不等于	307
	LD/AND/OR <>SL	双字带符号不等于	308
C1 < C2 时为真	LD/AND/OR <	小于	310
	LD/AND/OR <L	双字小于	311
	LD/AND/OR<S	带符号小于	312
	LD/AND/OR <SL	双字带符号小于	313

功能	助记符	名称	代码
C1 ≤ C2 时为真	LD/AND/OR <=	小于等于	315
	LD/AND/OR<=L	双字小于等于	316
	LD/AND/OR <=S	带符号小于等于	317
	LD/AND/OR <=SL	双字带符号小于等于	318
C1 ＞ C2 时为真	LD/AND/OR >	大于	320
	LD/AND/OR >L	双字大于	321
	LD/AND/OR>S	带符号大于	322
	LD/AND/OR >SL	双字带符号大于	323
C1 ≥ C2 时为真	LD/AND/OR >=	大于等于	325
	LD/AND/OR>=L	双字大于等于	326
	LD/AND/OR >=S	带符号大于等于	327
	LD/AND/OR >=SL	双字带符号大于等于	328

无符号输入比较指令 (即指令无 S 选项) 可处理无符号二进制或 BCD 数据。有符号输入比较指令 (即指令有 S 选项) 可处理带符号二进制。

提示

・ 与 CMP(020) 和 CMPL(060) 之类的指令不同，输入比较指令的结果将直接反映为执行条件，因此无需通过算术标志访问比较结果，从而使得程序更加简捷。

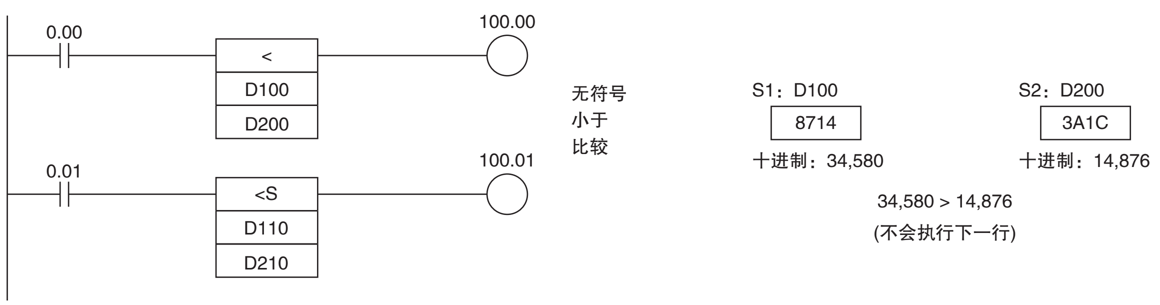
注意事项

・ 输入指令不能作为右侧指令使用，也就是说这些指令和右侧母线之间必须要有其它指令。

程序举例

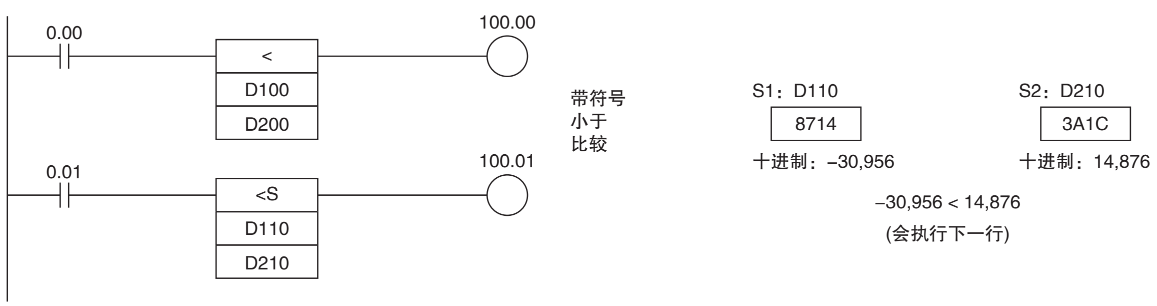
与小于：AND<(310)

下例中，当 CIO 0.00 为 ON 时，D100 和 D200 中的内容作为无符号二进制数据比较。如果 D100 中的内容小于 D200 中的内容，CIO 100.00 将变 ON，且执行进到下一行。如果 D100 中的内容不小于 D200 中的内容，此行余下的指令将被跳过，且执行移到下一行。



与带符号小于：AND<S(312)

下例中，当 CIO 0.01 为 ON 时，D110 和 D210 中的内容作为带符号二进制数据比较。如果 D110 中的内容小于 D210 中的内容，CIO 100.01 将变 ON，且执行进到下一行。如果 D110 中的内容不小于 D210 中的内容，此行余下的指令将被跳过，且执行移到下一行。



=DT，<>DT，<DT，<=DT，>DT，>=DT

指令	助记符	变化	功能代码	功能
时间比较指令	=DT <>DT <DT <=DT >DT >=DT	---	341 342 343 344 345 346	时间比较指令比较两个 BCD 时间值，并在比较条件为真时生成一个 ON 执行条件。

符号	=DT，<>DT，<DT，<=DT，>DT，>=DT		
	LD 符号 C S1 S2 C：控制字 S1：当前时间的首字 S2：比较时间的首字	AND 符号 C S1 S2 C：控制字 S1：当前时间的首字 S2：比较时间的首字	OR 符号 C S1 S2 C：控制字 S1：当前时间的首字 S2：比较时间的首字

适用程序区

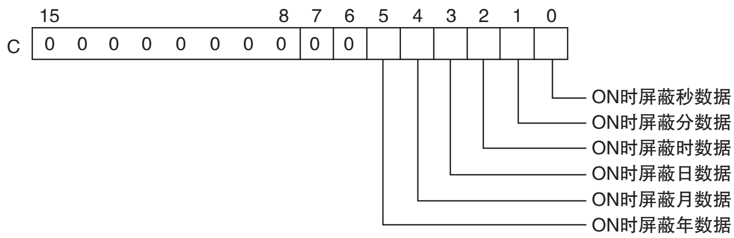
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
C	控制字	WORD	1
S1	当前时间的首字	WORD	3
S2	比较时间的首字	WORD	3

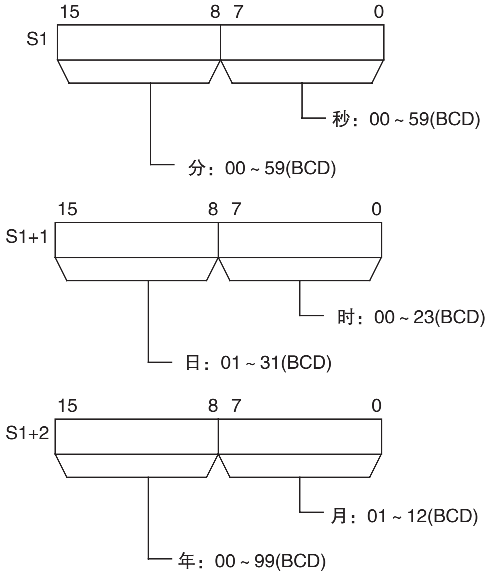
C：控制字

C 的 00 ~ 05 位指定是否屏蔽时间数据以进行比较。00 ~ 05 位分别屏蔽秒、分、时、日、月和年。如果这所有 6 个值均被屏蔽，则指令将不执行，执行条件将为 OFF，且出错标志将变 ON。



S_1 ~ S_1+2：当前时间数据

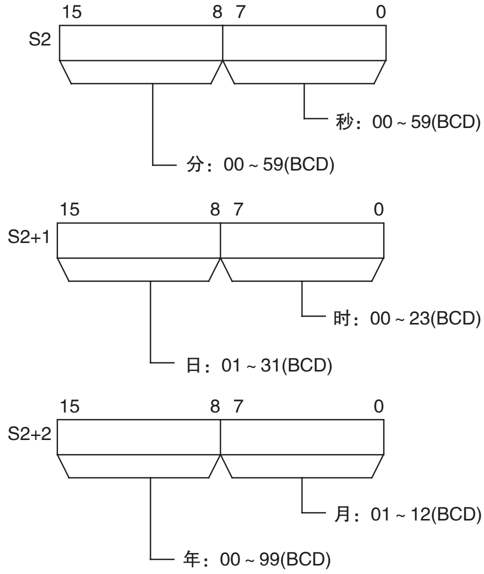
S_1 ~ S_1+2 中包含当前时间数据。S_1 ~ S_1+2 必须位于同一个数据区。



注 当将 CPU 单元的内部时钟数据用于比较时，请将 S1 设定为 A351，以指定 CPU 单元的内部时钟数据 (A351 ~ A353)。

S_2 ~ S_2+2：比较时间数据

S_2 ~ S_2+2 中包含比较时间数据。S_2 ~ S_2+2 必须位于同一个数据区。



注 年份值表示年份的后两位。值00 ~ 97表示2000 ~ 2097。值 98 和 99 表示 1998 和 1999。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	OK	---	---	---
S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 全部 6 个屏蔽位 (C 的 00 ~ 05 位) 为 ON 时 ON。 · 其它情况下 OFF。
大于标志	P_GT	· $S_1 > S_2$ 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	· $S_1 \geqslant S_2$ 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· S_1=S_2 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	· $S_1 \neq S_2$ 时 ON。 · 其它情况下 OFF。
小于标志	P_LT	· $S_1 < S_2$ 时 ON。 · 其它情况下 OFF。
小于等于标志	P_LE	· $S_1 \leqslant S_2$ 时 ON。 · 其它情况下 OFF。

功能

时间比较指令将 S_1 ～ S_1+2 中当前时间数据的未屏蔽值 (C 中被置为 0 的相应位) 与 S_2 ～ S_2+2 中的比较时间数据进行比较，并在比较条件为真时产生一个 ON 执行条件。与此同时，时间比较指令的结果反映到算术标志 (=，<>，<，<=，>，>=) 中。

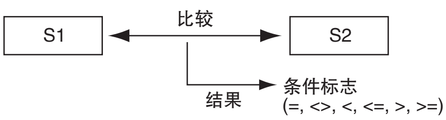
对时间比较指令的处理类似于 LD、AND 和 OR 指令，用于控制随后的指令执行。

时间比较指令有 18 种可能的组合。

比较中不包括在控制字 (C) 中被屏蔽的任何时间值。

下表所示为各比较结果的每个标志的 ON/OFF 状态。

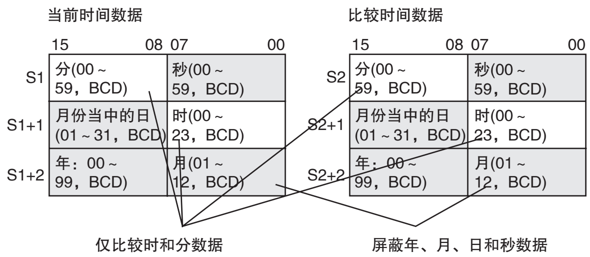
结果	标志状态					
	=	<>	<	<=	>	>=
S_1=S_2	ON	OFF	OFF	ON	OFF	ON
S_1 > S_2	OFF	ON	OFF	OFF	ON	ON
S_1 < S_2	OFF	ON	ON	ON	OFF	OFF



● 屏蔽时间值

可在比较运算中单独指定屏蔽或不屏蔽某些时间值。若要屏蔽某个时间值，可将控制字 (C) 中的相应位置为 1。C 中的位 00 ～ 05 位分别屏蔽秒、分、时、日、月和年。

例：
当 C= 十六进制的 39 时，最右边的 6 位为 111001(年 =1，月 =1，日 =1，时 =0，分 =0，秒 =1)，因此只比较时和分。该屏蔽设定可用于在每天的特定时间 (时和分) 执行特定的操作。



提示

・ 先前的数据比较指令以 16 位为单位比较数据。时间比较指令限于比较 8 位时间值。
  下表所示为 CPU 单元的内部日历 / 时钟区的结构。

地址	内容
A351.00 ～ A351.07	秒 (00 ～ 59，BCD)
A351.08 ～ A351.15	分 (00 ～ 59，BCD)
A352.00 ～ A352.07	时 (00 ～ 23，BCD)
A352.08 ～ A352.15	月份当中的日 (01 ～ 31，BCD)
A353.00 ～ A353.07	月 (01 ～ 12，BCD)
A353.08 ～ A353.15	年 (00 ～ 99，BCD)

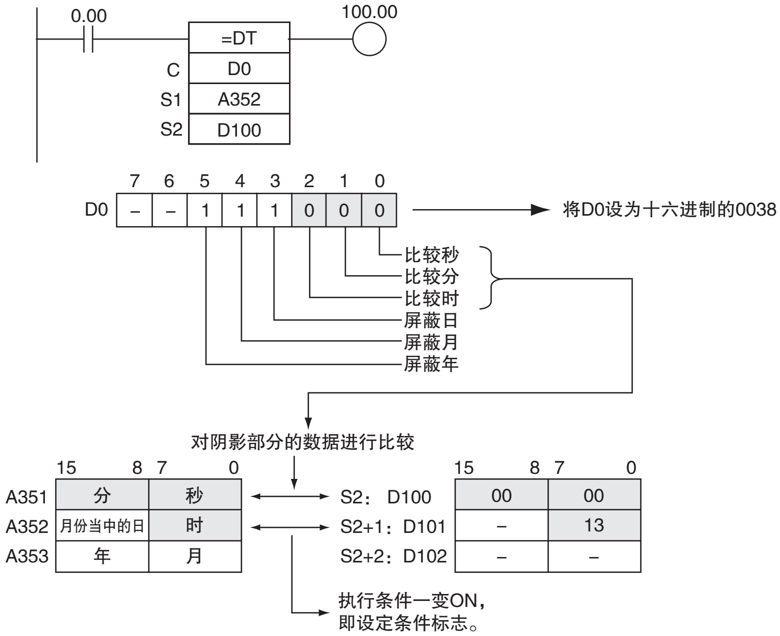
・ 可通过编程设备 (包括编程器)、DATE(735) 指令或“CLOCK WRITE”FINS 命令 (十六进制的 0702) 来设定日历 / 时钟区。

注意事项

- 时间指令不能作为右侧指令使用，也就是说这些指令和右侧母线之间必须要有其它指令。
- CP1E/CP2E E 型 CPU 单元不具有时钟功能。
  CPU 单元内部的时钟数据始终为 01-01-01 01:01:01 的形式。

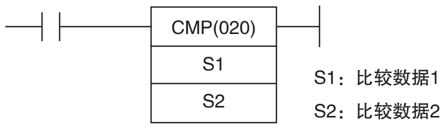
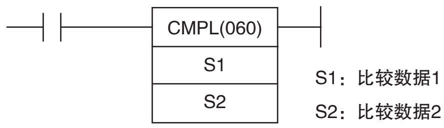
程序举例

当 CIO 0.00 为 ON 且时间为 13:00:00 时，CIO 100.00 变 ON。A351 ～ A353(CPU 单元的内部日历 / 时钟数据) 中的内容用作当前时间数据，而 D100 ～ D102 中的内容用作比较时间数据。年、月和日的值被屏蔽，因此只比较时、分和秒数据。



CMP/CMPL

指令	助记符	变化	功能代码	功能
比较	CMP	!CMP	020	比较两个无符号二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。
双字比较	CMPL	---	060	比较两个无符号双字二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。

	CMP	CMPL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		CMP	CMPL	CMP	CMPL
S1	CMP：比较数据 1 CMPL：比较数据 1，最右边的字地址	UINT	UDINT	1	2
S2	CMP：比较数据 2 CMPL：比较数据 2，最右边的字地址	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
CMP	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
CMPL	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	OK	---	---	---

条件标志

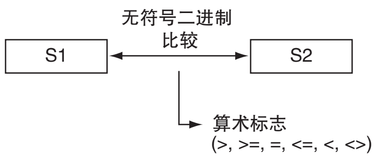
名称	标记	操作	
		CMP	CMPL
出错标志	P_ER	不变	不变
大于标志	P_GT	· $S+1 > S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 > S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	· $S_1 \geqslant S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \geqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· $S_1=S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1=S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	· $S_1 \neq S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \neq S_2+1$，S_2 时 ON。 · 其它情况下 OFF。

名称	标记	操作	
		CMP	CMPL
小于标志	P_LT	· $S_1 < S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 < S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
小于等于标志	P_LE	· $S_1 \leqslant S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \leqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
负标志	P_N	不变	不变

● 下表所示为执行 CMP(020) 指令之后的算术标志的状态。

CMP(020) 的结果	标志状态					
	>	> =	=	< =	<	< >
$S_1 > S_2$	ON	ON	OFF	OFF	OFF	ON
$S_1=S_2$	OFF	ON	ON	ON	OFF	OFF
$S_1 < S_2$	OFF	OFF	OFF	ON	ON	ON

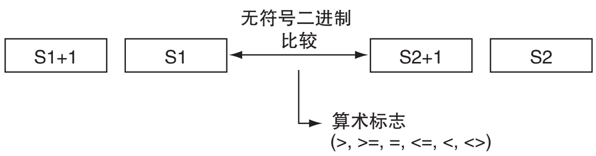
* 状态 “---” 表示标志可能为 ON 也可能为 OFF。



● 下表所示为执行 CMPL(060) 指令之后的算术标志的状态。

CMPL(060) 的结果	标志状态					
	>	> =	=	< =	<	< >
S_1+1，$S_1 > S_2+1$，S_2	ON	ON	OFF	OFF	OFF	ON
S_1+1，$S_1=S_2+1$，S_2	OFF	ON	ON	ON	OFF	OFF
S_1+1，$S_1 < S_2+1$，S_2	OFF	OFF	OFF	ON	ON	ON

* 状态 “---” 表示标志可能为 ON 也可能为 OFF。



功能

● CMP

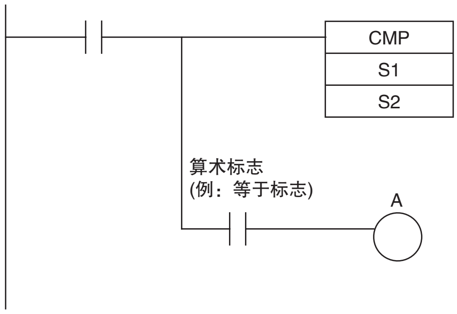
CMP(020) 比较 S_1 和 S_2 中的无符号二进制数据，并将结果输出到辅助区中的算术标志 (大于、大于等于、等于、小于等于、小于和不等于标志) 中。

● CMPL

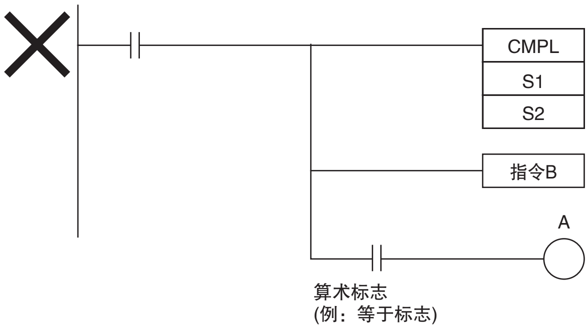
CMPL(060) 比较 S_1+1，S_1 和 S_2+1，S_2 中的无符号二进制数据，并将结果输出到辅助区中的算术标志 (大于、大于等于、等于、小于等于、小于和不等于标志) 中。

注意事项

· 在程序中使用 CMP(020) 的结果
执行 CMP(020)/CMPL(060) 时，结果将反映到算术标志中。用与控制 CMP(020)/CMPL(060) 指令相同的输入条件分支来控制所需的输出或右侧指令，如下图所示。在这种情况下，当 S_1=S_2 或 S_1+1，S_1=S_2+1，S_2 时，等于标志和输出 A 将变 ON。



· 在程序中使用 CMP(020) 的结果
请勿在 CMP(020)/CMPL(060) 指令和通过算术标志控制的指令之间编入其它指令，因为编入的其它指令可能会改变算术标志的状态。

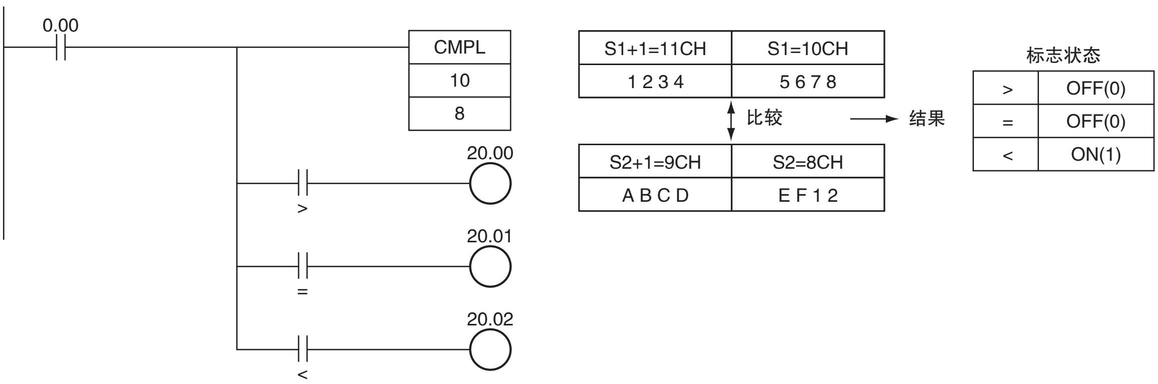


在这种情况下，指令 B 的结果可能会改变 CMP(020) 的结果。

· 即时刷新变化 (!CMP(020)) 指令可用于在 S_1 和 / 或 S_2 中指定的分配给 CPU 单元的内置输入字。执行 (!CMP(020)) 指令时，将对在 S_1 和 / 或 S_2 中指定的外部输入字执行输入刷新，并比较刷新后的值。

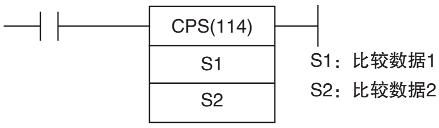
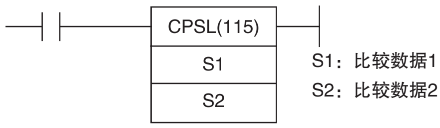
程序举例

· 下例中，当 CIO 0.00 为 ON 时，CIO 0011 和 CIO 0010 中的 8 位数无符号二进制数据与 CIO 0009 和 CIO 0008 中的 8 位数无符号二进制数据进行比较，并将结果输出到算术标志中。大于、等于和小于标志中记录的结果将即时保存到 CIO 20.00(大于)、CIO 20.01(等于) 和 CIO 20.02(小于) 中。



CPS/CPSL

指令	助记符	变化	功能代码	功能
带符号二进制比较	CPS	!CPS	114	比较两个带符号二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。
带符号双字二进制比较	CPSL	---	115	比较两个带符号双字二进制值 (常数和 / 或指定字的内容) 并将结果输出至辅助区的算术标志中。

符号	CPS	CPSL
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		CPS	CPSL	CPS	CPSL
S1	CMP：比较数据 1 CMPL：比较数据 1，最右边的字地址	INT	DINT	1	2
S2	CMP：比较数据 2 CMPL：比较数据 2，最右边的字地址	INT	DINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
CPS	S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
CPSL												---					

条件标志

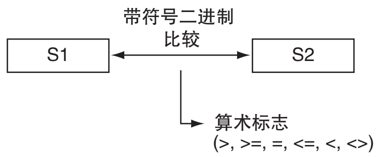
名称	标记	操作	
		CPS	CPSL
出错标志	P_ER	不变	OFF 或不变
大于标志	P_GT	· $S_1 > S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 > S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	· $S_1 \geqslant S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \geqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· $S_1=S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1=S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	· $S_1 \neq S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \neq S_2+1$，S_2 时 ON。 · 其它情况下 OFF。

名称	标记	操作	
		CPS	CPSL
小于标志	P_LT	· $S_1 < S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 < S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
小于等于标志	P_LE	· $S_1 \leqslant S_2$ 时 ON。 · 其它情况下 OFF。	· S_1+1，$S_1 \leqslant S_2+1$，S_2 时 ON。 · 其它情况下 OFF。
负标志	P_N	不变	OFF 或不变

● 下表所示为执行 CPS(114) 指令之后的算术标志的状态。

结果	标志状态					
	>	> =	=	< =	<	< >
$S_1 > S_2$	ON	ON	OFF	OFF	OFF	ON
$S_1=S_2$	OFF	ON	ON	ON	OFF	OFF
$S_1 < S_2$	OFF	OFF	OFF	ON	ON	ON

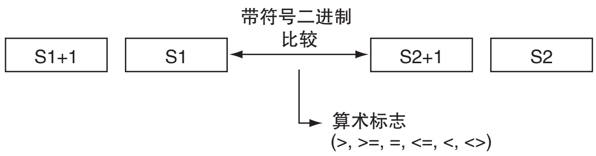
* 状态 “---”表示标志可能为 ON 也可能为 OFF。



● 下表所示为执行 CPSL(115) 指令之后的算术标志的状态。

结果	标志状态					
	>	> =	=	< =	<	< >
S_1+1，$S_1 > S_2+1$，S_2	ON	ON	OFF	OFF	OFF	ON
S_1+1，$S_1=S_2+1$，S_2	OFF	ON	ON	ON	OFF	OFF
S_1+1，$S_1 < S_2+1$，S_2	OFF	OFF	OFF	ON	ON	ON

* 状态 “---”表示标志可能为 ON 也可能为 OFF。



功能

● CPS

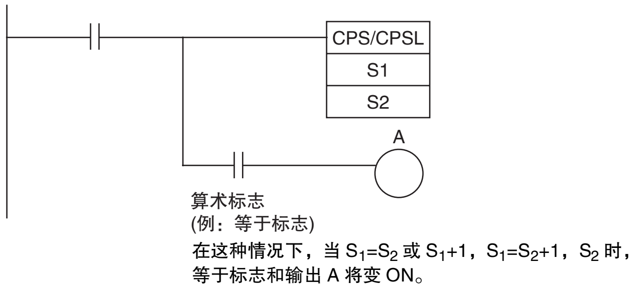
CPS(114) 比较 S_1 和 S_2 中的带符号二进制数据，并将结果输出到辅助区中的算术标志 (大于、大于等于、等于、小于等于、小于和不等于标志) 中。

● CPSL

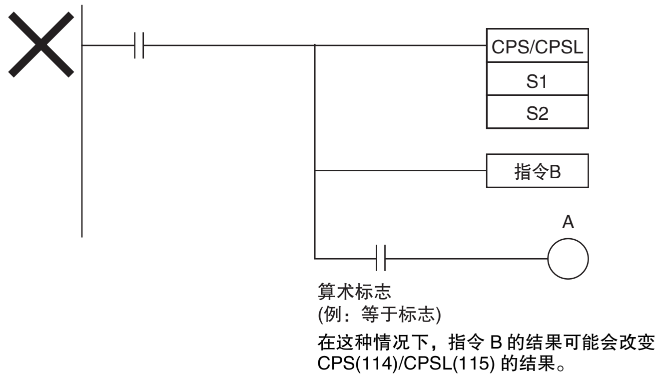
CPSL(115) 比较 S_1+1，S_1 和 S_2+1，S_2 中的带符号双字二进制数据，并将结果输出到辅助区中的算术标志 (大于、大于等于、等于、小于等于、小于和不等于标志) 中。

注意事项

- 执行 CPS(114)/CPSL(115) 时，结果将反映到算术标志中。用与控制 CPS(114)/CPSL(115) 指令相同的输入条件分支来控制所需的输出或右侧指令，如下图所示。



- 请勿在 CPS(114)/CPSL(115) 指令和通过算术标志控制的指令之间编入其它指令，因为编入的其它指令可能会改变算术标志的状态。

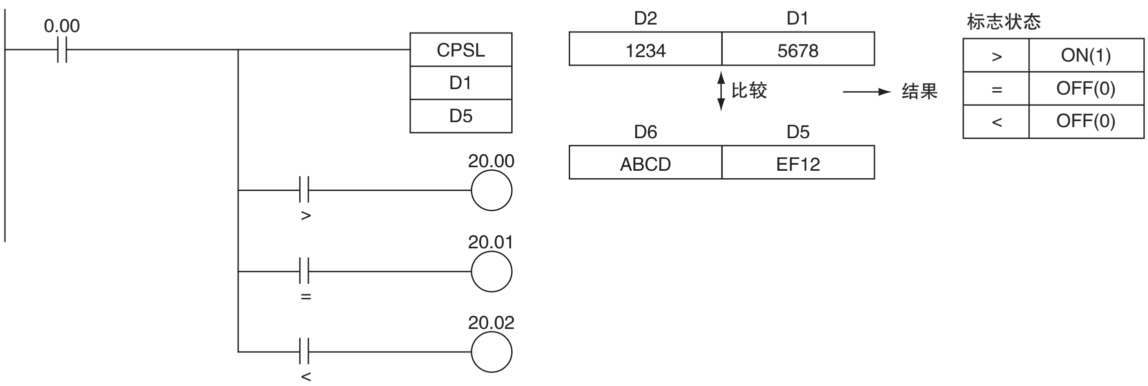


- 即时刷新变化 (!CPS(114)/!CPSL(115)) 指令可用于在 S_1 和 / 或 S_2 中指定的分配给 CPU 单元的内置输入字。执行 !CPS(114)/!CPSL(115) 指令时，将对在 S_1 和 / 或 S_2 中指定的外部输入字执行输入刷新，并比较刷新后的值。

程序举例

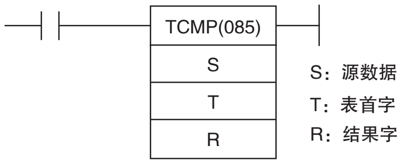
下例中，当 CIO 0.00 为 ON 时，D2 和 D1 中的 8 位数带符号二进制数据与 D6 和 D5 中的 8 位数带符号二进制数据进行比较，并将结果输出到算术标志中。

- 如果 D2 和 D1 中的内容大于 D6 和 D5 中的内容，则大于标志将变 ON，从而使得 CIO 20.00 变 ON。
- 如果 D2 和 D1 中的内容等于 D6 和 D5 中的内容，则等于标志将变 ON，从而使得 CIO 20.01 变 ON。
- 如果 D2 和 D1 中的内容小于 D6 和 D5 中的内容，则小于标志将变 ON，从而使得 CIO 20.02 变 ON。



TCMP

指令	助记符	变化	功能代码	功能
表比较	TCMP	@TCMP	085	比较源数据和 16 个连续字的内容，当两字的内容**相等**时，使结果字中的相应位变 ON。

符号	TCMP
	

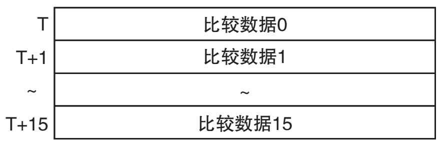
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

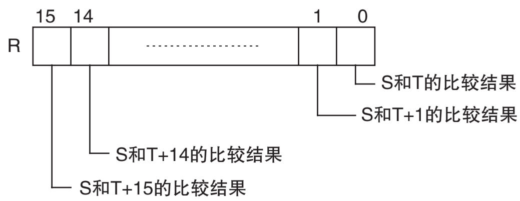
操作数

操作数	描述	数据类型	大小
S	源数据	WORD	1
T	表首字	WORD	16
R	结果字	UINT	1

T：表首字



R：结果字



● 区域类型

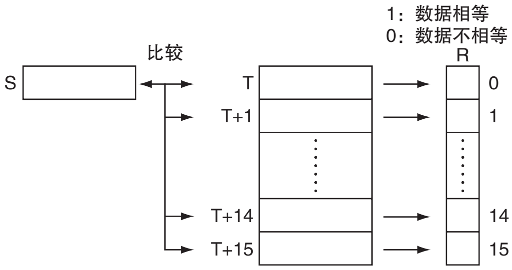
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S										OK	OK					
T	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
R											OK					

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果字为 0000 时 ON。 (这两个 16 字的数据集中包含相同的数据。) · 其它情况下 OFF。

功能

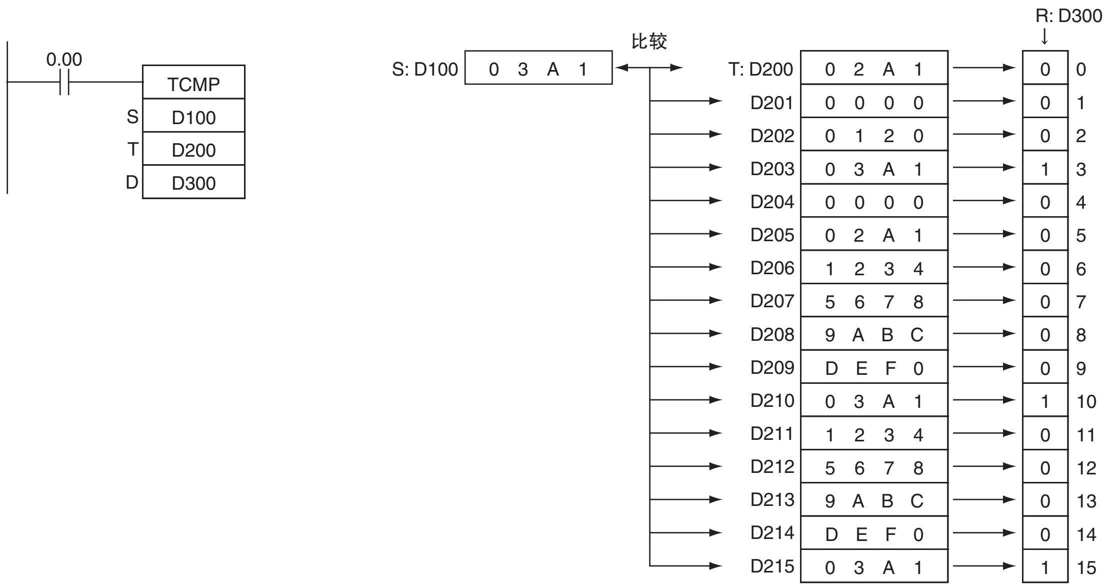
TCMP(085) 将源数据 (S) 与 16 字 (T ～ T+15) 中的每一个进行比较，如果数据相等，则字 R 中的相应位变 ON。如果 T+n 中的内容等于 S，则 R 的位 n 变 ON；如果不相等则变 OFF。



将 S 与 T 的内容作比较，如果相等，则 R 中的位 00 变 ON，如果不相等则变 OFF；将 S 与 T+1 中的内容作比较，如果相等，则 R 中的位 01 变 ON，如果不相等则变 OFF；……将 S 与 T+15 的内容作比较，如果相等，则 R 中的位 15 变 ON，如果不相等则变 OFF。

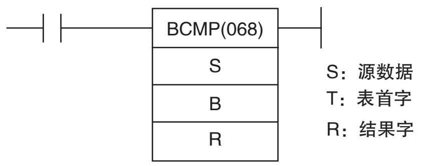
程序举例

下例中，当 CIO 0.00 为 ON 时，TCMP(085) 将字 D100 的内容与字 D200 ～ D215 中的内容作比较，当内容相等时，使 D300 中的相应位变 ON；当内容不相等时，使 D300 中的相应位变 OFF。



BCMP

指令	助记符	变化	功能代码	功能
块比较	BCMP	@BCMP	068	将源数据与 16 个范围 (由 16 个下限和 16 个上限定义) 进行比较，当源数据在某个范围内时，使结果字中的相应位变 ON。

符号	BCMP
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

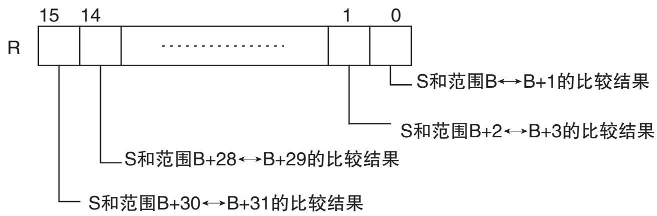
操作数

操作数	描述	数据类型	大小
S	源数据	WORD	1
B	块首字	WORD	32
R	结果字	UINT	1

B：块首字

B	下限值0
B+1	上限值0
B+2	下限值1
B+3	上限值1
~	~
B+30	下限值15
B+31	上限值15

R：结果字



● 区域类型

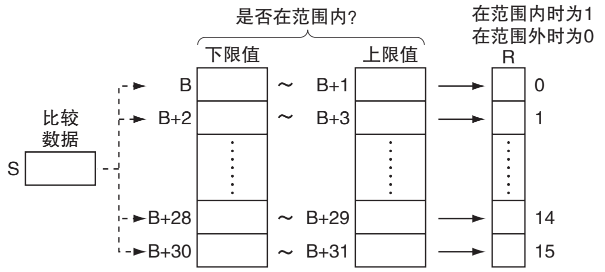
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
B	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果字为 0000 时 ON。 (S 不在 16 个范围中的任一个范围内。) · 其它情况下 OFF。

功能

BCMP(068) 将源数据 (S) 和 B ~ B+31 中由 16 对下限和上限值定义的 16 个范围进行比较。每对的第一个字 (B+2n) 为范围 n 提供下限，第二个字 (B+2n+1) 为范围 n 提供上限 (n=0 ~ 15)。如果 S 在这些范围 (包含上、下限) 中的任一范围之内，则 R 中的相应位变 ON。 如果 S 不在这些范围之外，则 R 中的相应位变 OFF。

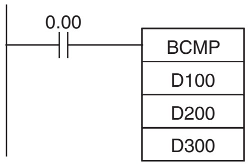


例如，如果 S 位于第一个范围 (B ≤ S ≤ B+1) 之内，则 R 的位 00 变 ON ；如果 S 位于第二个范围 (B+2 ≤ S ≤ B+3) 之内，则 R 的位 01 变 ON ；…… 如果 S 位于第 15 个范围 (B+30 ≤ S ≤ B+31) 之内，则 R 的位 15 变 ON。R 中的所有其它位均变 OFF。

注 如果下限大于上限，不会发生错误，但会将 0(不在范围内) 输出到 R 的相应位。

程序举例

下例中，当 CIO 0.00 为 ON 时，BCMP(068) 将 D100 中的内容与在 D200 ~ D231 定义的 16 个范围作比较，当 S 在范围内时，使 D300 中的相应位变 ON ；当 S 不在范围内时，使 D300 中的相应位变 OFF。

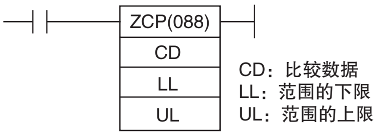
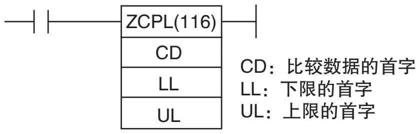


S:D100 03A0

					R: D300	
D200	0 3 A 1	~	D201	0 4 0 0	0	0
D202	0 0 0 0	~	D203	0 3 9 0	0	1
D204	0 0 0 0	~	D205	0 4 0 0	1	2
D206	0 3 A 1	~	D207	0 4 0 0	0	3
D208	0 3 A 0	~	D209	0 4 0 0	1	4
D210	0 0 0 0	~	D211	0 4 0 0	1	5
D212	0 0 0 0	~	D213	0 3 9 0	0	6
D214	0 3 A 1	~	D215	0 4 0 0	0	7
D216	0 0 0 0	~	D217	0 5 0 0	1	8
D218	0 0 0 0	~	D219	0 3 A 0	1	9
D220	0 0 0 0	~	D221	0 3 9 0	0	10
D222	0 3 A 1	~	D223	0 4 0 0	0	11
D224	0 0 0 0	~	D225	0 4 0 0	1	12
D226	0 0 0 0	~	D227	0 0 0 0	0	13
D228	0 3 A 1	~	D229	0 0 0 0	0	14
D230	0 0 0 0	~	D231	0 0 0 0	0	15

ZCP/ZCPL

指令	助记符	变化	功能代码	功能
区域范围比较	ZCP	---	088	将一个 16 位无符号二进制值 (CD) 与由下限 LL 和上限 UL 所定义的范围作比较，并将结果输出至算术标志中。
双字区域范围比较	ZCPL	---	116	将一个 32 位无符号二进制值 (CD+1，CD) 与由下限 (LL+1，LL) 和上限 (UL+1，UL) 所定义的范围作比较，并将结果输出至算术标志中。

	ZCP	ZCPL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ZCP	ZCPL	ZCP	ZCPL
CD	ZCP：比较数据 (单字数据) ZCPL：比较数据 (双字数据)	UINT	UDINT	1	2
LL	ZCP：下限 ZCPL：下限的最左字	UINT	UDINT	1	2
UL	ZCP：上限 ZCPL：上限的最右字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ZCP	CD，LL，UR	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
ZCPL	CD，LL，UR	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	OK	---	---	---

条件标志

名称	标记	操作	
		ZCP	ZCPL
出错标志	P_ER	LL>UL 时 ON。	LL+1，LL>UL+1，UL 时 ON。
大于标志	P_GT	· CD>UL 时 ON。 · 其它情况下 OFF。	· CD+1，CD>UL+1，UL 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	保持不变。	保持不变。
等于标志	P_EQ	· LL ≤ CD ≤ UL 时 ON。 · 其它情况下 OFF。	· LL+1，LL ≤ CD+1，CD ≤ UL+1，UL 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	保持不变。	保持不变。

名称	标记	操作	
		ZCP	ZCPL
小于标志	P_LT	・CD<LL 时 ON。 ・其它情况下 OFF。	・CD+1，CD<LL+1，LL 时 ON。 ・其它情况下 OFF。
小于等于标志	P_LE	保持不变。	保持不变。
负标志	P_N	保持不变。	保持不变。

功能

● ZCP

ZCP(088) 将 CD 中的 16 位带符号二进制数据与由 LL 和 UL 所定义的范围作比较，并将结果输出到辅助区内的大于、等于和小于标志中。(小于等于、大于等于和不等于标志保持不变。)

如下所示，当 CD>UL 时，> 标志变 ON。
当 LL ≤ CD ≤ UL 时，= 标志变 ON。当 CD<LL 时，< 标志变 ON。

<table>
<tr><th rowspan="2">ZCP(088) 的结果</th><th colspan="3">标志状态</th></tr>
<tr><th>></th><th>=</th><th><</th></tr>
<tr><td>CD>UL</td><td>ON</td><td>OFF</td><td rowspan="4">OFF</td></tr>
<tr><td>CD=UL</td><td rowspan="4">OFF</td><td rowspan="3">ON</td></tr>
<tr><td>LL<CD<UL</td></tr>
<tr><td>CD=LL</td></tr>
<tr><td>CD<LL</td><td>OFF</td><td>ON</td></tr>
</table>

● ZCPL

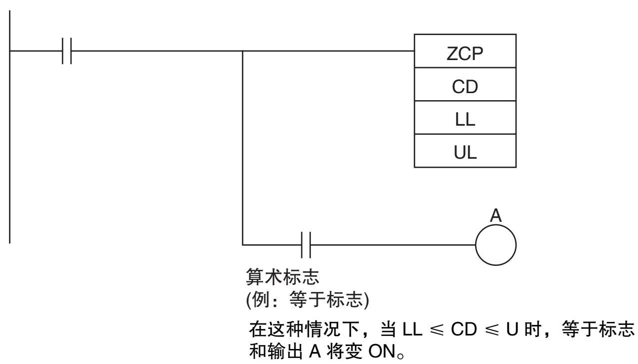
ZCPL(116) 将 CD+1，CD 中的 32 位带符号二进制数据与由 LL+1，LL 和 UL+1，UL 所定义的范围作比较，并将结果输出到辅助区内的大于、等于和小于标志中。(小于等于、大于等于和不等于标志保持不变。)

如下所示，当 CD+1，CD>UL+1，UL 时，> 标志变 ON。
当 LL+1，LL ≤ CD+1，CD ≤ UL+1，UL 时，= 标志变 ON。
当 CD+1，CD<LL+1，LL 时，< 标志变 ON。

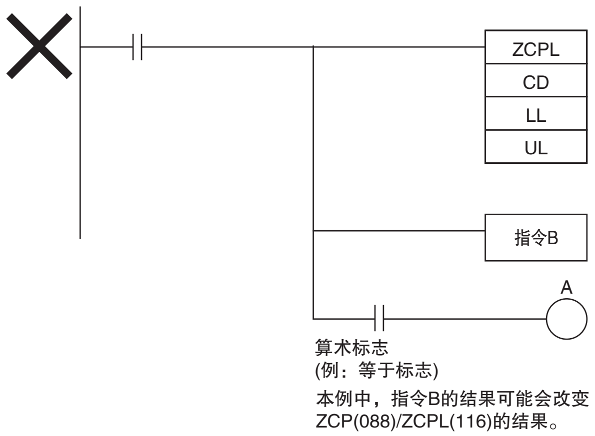
<table>
<tr><th rowspan="2">ZCPL(116) 的结果</th><th colspan="3">标志状态</th></tr>
<tr><th>></th><th>=</th><th><</th></tr>
<tr><td>CD+1，CD>UL+1，UL</td><td>ON</td><td>OFF</td><td rowspan="4">OFF</td></tr>
<tr><td>CD+1，CD=UL+1，UL</td><td rowspan="4">OFF</td><td rowspan="3">ON</td></tr>
<tr><td>LL+1，LL<CD+1，CD<UL+1，UL</td></tr>
<tr><td>CD+1，CD=LL+1，LL</td></tr>
<tr><td>CD+1，CD<LL+1，LL</td><td>OFF</td><td>ON</td></tr>
</table>

注意事项

・执行 ZCP(088)/ZCPL(116) 时，结果将反映到算术标志中。用与控制 ZCP(088)/ZCPL(116) 指令相同的输入条件分支来控制所需的输出或右侧指令，如下图所示。

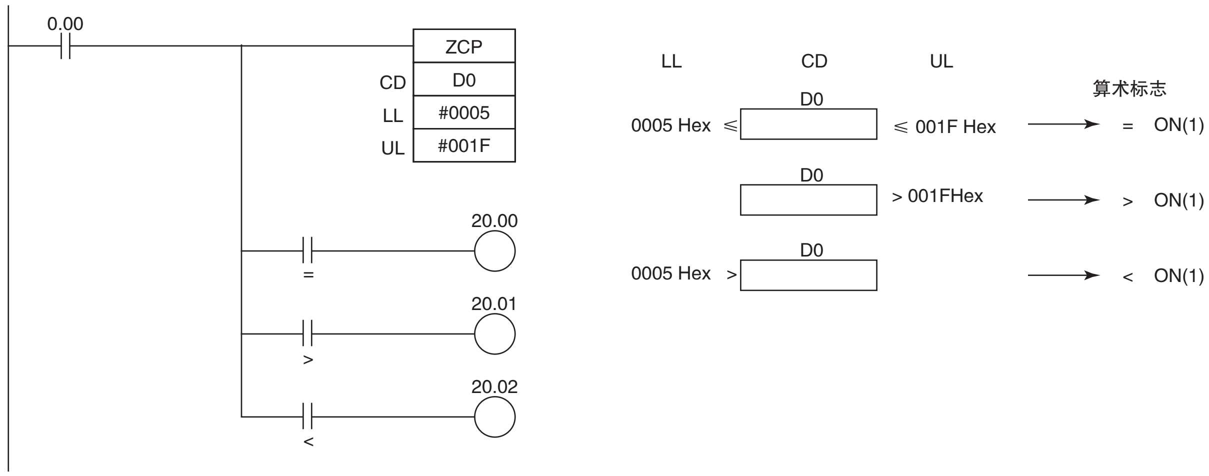


· 请勿在 ZCP(088)/ZCPL(116) 指令和通过算术标志控制的指令之间编入其它指令，因为编入的其它指令可能会改变算术标志的状态。



程序举例

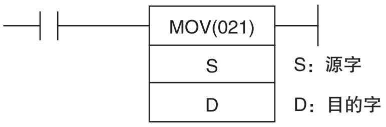
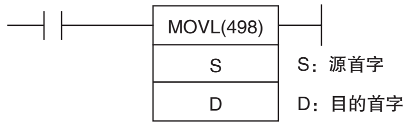
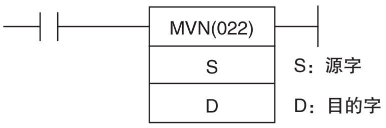
· 下例中，当 CIO 0.00 为 ON 时，D0 中的 16 位无符号二进制数据与十六进制的 0005 ～ 001F(十进制的 5 ～ 31) 范围进行比较，并将结果输出到算术标志中。
如果 0005 Hex ≤ D0 的内容≤ 001F Hex，则 CIO 20.00 变 ON。
如果 D0 的内容 >001F Hex，则 CIO 20.01 变 ON。
如果 D0 的内容 <0005 Hex，则 CIO 20.02 变 ON。



数据传送指令

MOV/MOVL/MVN

指令	助记符	变化	功能代码	功能
传送	MOV	@MOV, !MOV, !@MOV	021	传送数据的一个字到指定字中。
双字传送	MOVL	@MOVL	498	传送数据的两个字到指定字中。
传送反	MVN	@MVN	022	将一个字的数据的补码传送到指定字中。

符号	MOV	MOVL
		
	MVN	
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>MOV/MVN</th><th>MOVL</th><th>MOV/MVN</th><th>MOVL</th></tr>
<tr><td>S</td><td>MOV/MVN：源字
MOVL：源首字</td><td>WORD</td><td>DWORD</td><td>1</td><td>2</td></tr>
<tr><td>D</td><td>MOV/MVN：目的字
MOVL：目的首字</td><td>WORD</td><td>DWORD</td><td>1</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th rowspan="2" colspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">MOV/ MOVL</td><td>S</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>D</td><td>---</td></tr>
<tr><td rowspan="2">MVN</td><td>S</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td></tr>
<tr><td>D</td><td>---</td></tr>
</table>

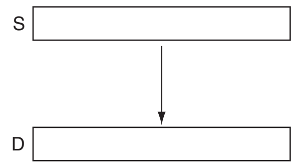
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	・正在传送的数据 (D) 为 0 时 ON。 ・其它情况下 OFF。
负标志	P_N	・正在传送的数据 (D) 的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

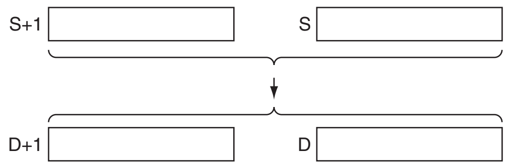
● MOV

将 S 传送到 D。如果 S 是一个常数，则该值可用作数据设定。



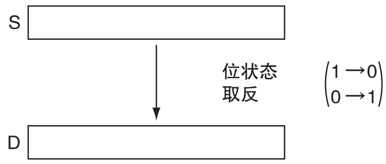
● MOVL

MOVL(498) 将 S+1 和 S 传送到 D+1 和 D。如果 S+1 和 S 是常数，则该值可用作数据设定。



● MVN

MVN(022) 指令对 S 中的位进行取反，并将结果传送到 D 中。S 中的内容保持不变。



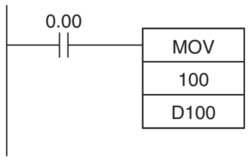
注意事项

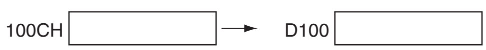
MOV(021) 具有即时刷新变化 (!MOV(021))。可分别为 S 和 D 指定 CPU 单元的内置输入位和外部输出位。用于 S 的输入位在即将执行前刷新，用于 D 的输出位在刚执行后刷新。

当为 S 指定了 CPU 单元的内置输入时，S 的值将在指令执行时进行内部刷新并传送到 D。当为 D 指定了外部输出时，S 的值被传送到 D，并在指令执行时立即进行外部刷新。还可同时对 S 进行内部刷新和对 D 进行外部刷新。

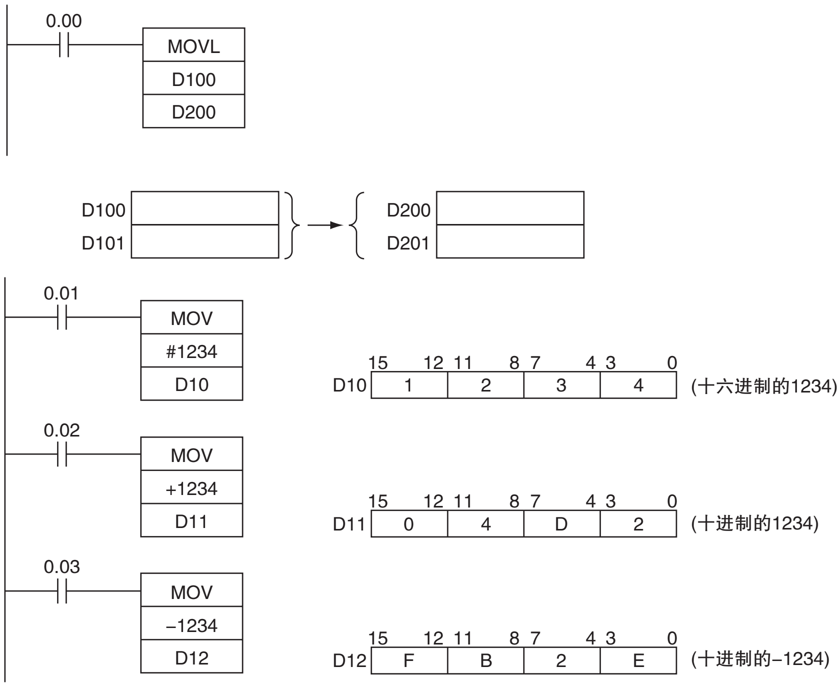
程序举例

下例中，当 CIO 0.00 为 ON 时，CIO 100 的内容被复制到 D100 中。

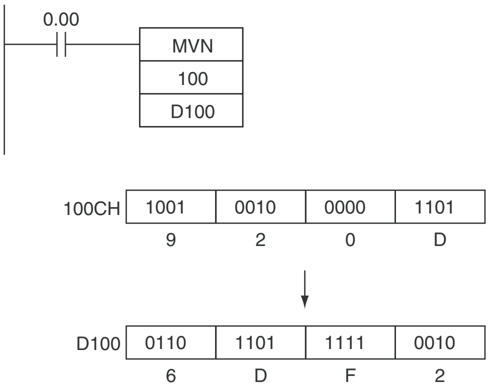




下例中，当 CIO 0.00 为 ON 时，D101 和 D100 的内容被复制到 D201 和 D200 中。

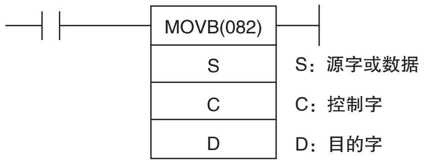


下例中，当 CIO 0.00 为 ON 时，对 CIO 100 中的位的状态取反，并将结果复制到 D100 中。



MOVB

指令	助记符	变化	功能代码	功能
位传送	MOVB	@MOVB	082	传送指定的位。

<table>
<tr><td rowspan="2">符号</td><td>MOVB</td></tr>
<tr><td>
</td></tr>
</table>

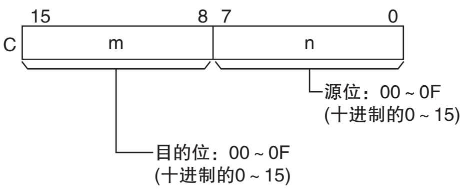
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字或数据	WORD	1
C	控制字	UINT	1
D	目的字	WORD	1

C：控制字



● 区域类型

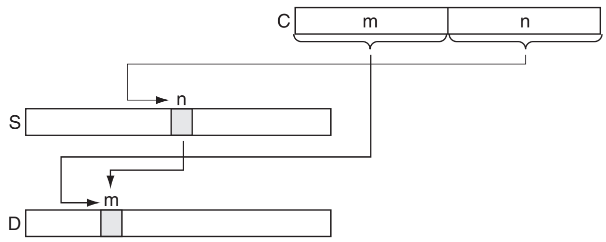
<table>
<tr><td rowspan="2">区域</td><td colspan="7">字地址</td><td colspan="2">DM 间接地址</td><td rowspan="2">常数</td><td colspan="3">寄存器</td><td rowspan="2">CF</td><td rowspan="2">脉冲位</td><td rowspan="2">TR 位</td></tr>
<tr><td>CIO</td><td>WR</td><td>HR</td><td>AR</td><td>T</td><td>C</td><td>DM</td><td>@DM</td><td>*DM</td><td>DR</td><td>IR 直接</td><td>IR 间接</td></tr>
<tr><td>S，C</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>D</td><td>---</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	・C 最右和最左的两个数位不在 00 ～ 0F 的指定范围内时 ON。 ・其它情况下 OFF。

功能

MOVB(082) 将 S 中的指定位 (n) 复制到 D 中的指定位 (m)。



提示

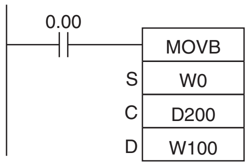
可为 S 和 D 指定同一个字，从而在同一个字内进行位复制。

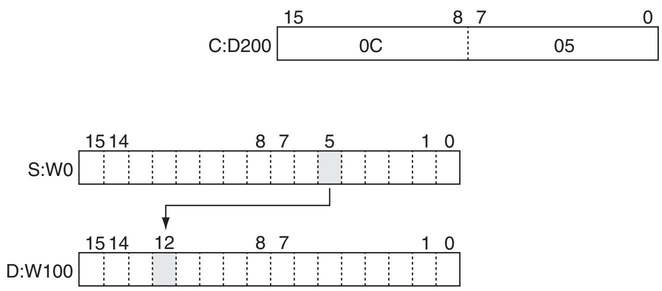
注意事项

目的字中的其它位均保持不变。

程序举例

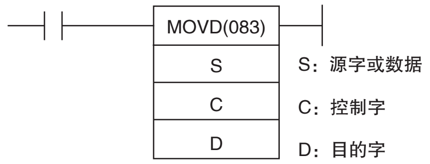
下例中，当 CIO 0.00 为 ON 时，依据控制字的值 0C05，将源字 (W0) 中的第 5 位复制到目的字 (W100) 的第 12 位中。





MOVD

指令	助记符	变化	功能代码	功能
数位传送	MOVD	@MOVD	083	传送指定的数位。 (每个数位由 4 个位组成。)

	MOVB
符号	

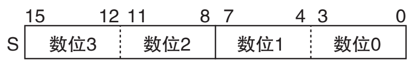
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

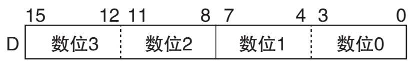
操作数	描述	数据类型	大小
S	源字或数据	WORD	1
C	控制字	UINT	1
D	目的字	UINT	1

S：源字



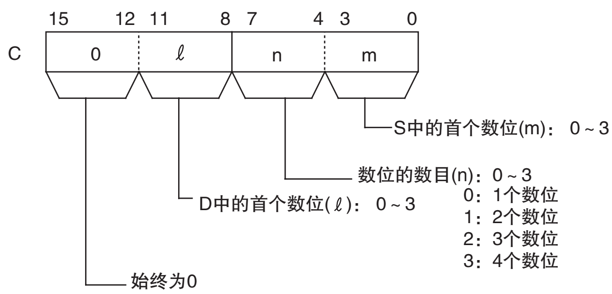
按从右到左的方向读取源数位，并在必要时绕回最右边的数位 (数位 0)。

D：目的字



按从右到左的方向写入目的数位，并在必要时绕回最右边的数位 (数位 0)。

C：控制字



● 区域类型

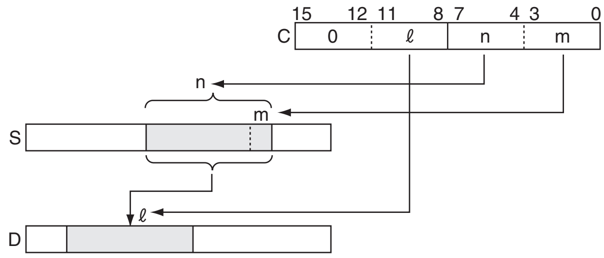
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S，C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・C 的前 3 个数位中的某个数位不在 0 ～ 3 的指定范围内时 ON。 ・其它情况下 OFF。

功能

MOVB(082) 将 S 中的指定位 (n) 复制到 D 中的指定位 (m)。目的字中的其它位均保持不变。

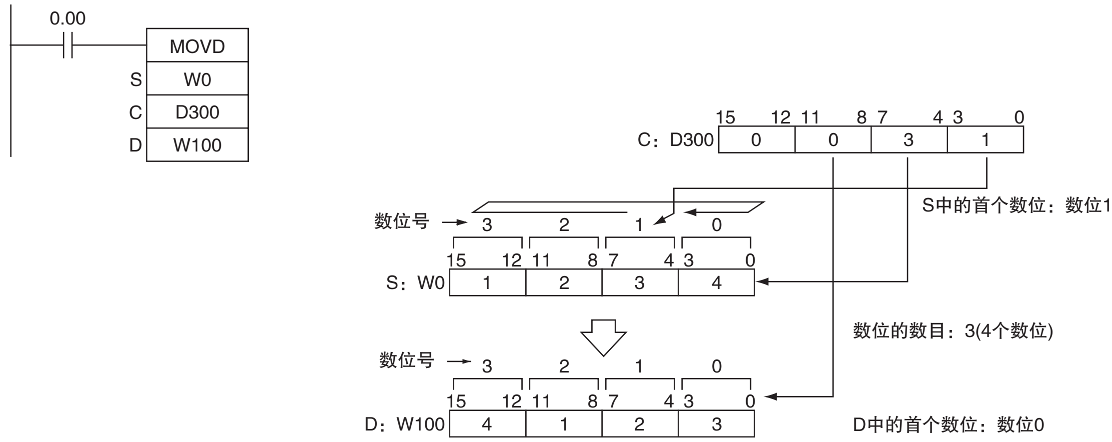


注意事项

如果读取或写入的数位个数超出 S 或 D 中最左边的数位，则 MOVD(083) 将绕回到这个字的最右边的数位。

程序举例

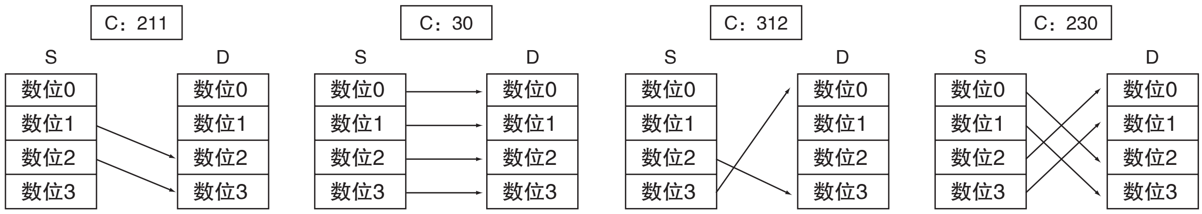
下例中，当 CIO 0.00 为 ON 时，数据中的 4 个数位将被从 W0 复制到 W100 中。依据控制字的值 31，将从 W0 的数位 1 和 W100 的数位 0 开始传送。



注 读完 S 最左边的数位 (数位 3) 之后，MOVD(083) 将绕回到最右边的数位 (数位 0)。

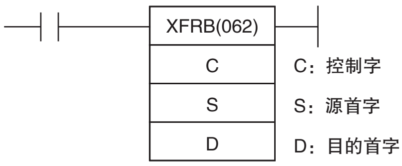
● 传送多个数位的编程举例

下图所示为在各种不同的 C 值下的数据传送示例。



XFRB

指令	助记符	变化	功能代码	功能
多位传送	XFRB	@XFRB	062	传送指定数目的连续位。

	XFRB
符号	

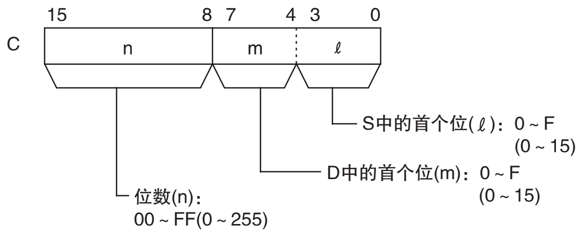
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

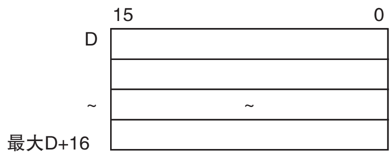
操作数	描述	数据类型	大小
C	控制字	UINT	1
S	源首字	WORD	可变
D	目的首字	WORD	可变

C：控制字

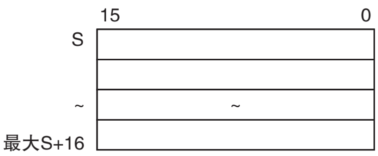


注 源字和目的字应各自在同一个数据区内。

D：目的首字



S：源首字



● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
S，D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

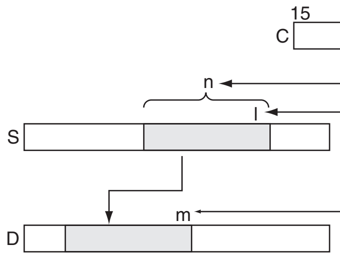
条件标志

名称	标记	操作
出错标志	P_ER	OFF

功能

XFRB(062) 最多可将源字中的 255 个连续位 (从 S 的位 1 开始) 传送到目的字中 (从 D 的位 m 开始)。

如图所示，开始位和位的个数在控制字 C 中指定。



提示

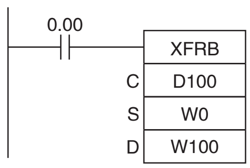
- 每执行一次 XFRB(062) 指令，最多可传送 255 位的数据。
- 源字和目的字可以重叠。通过传送重叠了几个字的数据，可对数据区内的数据进行更有效的打包。(在处理用于控制位置的位置数据时特别有用。)
- 由于源字与目的字可以重叠，因而可将 XFRB(062) 和 ANDW(034) 组合使用，从而通过 n 个空格来移动 m 个位。

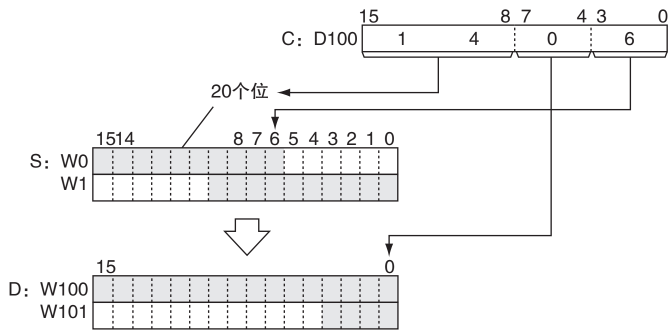
注意事项

- 务必确保源字和目的字不超出数据区的末端。
- 当传送的位数 (C 的 n) 为 0 时，不进行传送。
- 目的字中未被源字的位覆盖的位保持不变。

程序举例

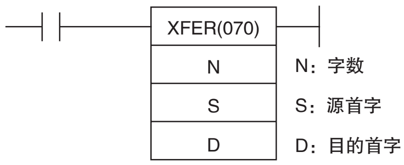
下例中，当 CIO 0.00 为 ON 时，将从 W0.06 开始的 20 个位复制到从 W100 开始的 20 个位中。





XFER

指令	助记符	变化	功能代码	功能
块传送	XFER	@XFER	070	传送指定数目的连续字。

符号	XFER
	

适用程序区

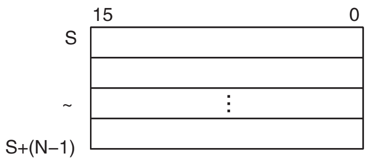
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

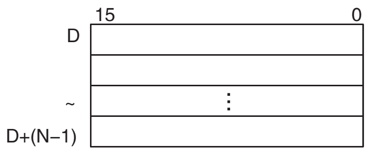
操作数

操作数	描述	数据类型	大小
N	字数	UINT	1
S	源首字	WORD	可变
D	目的首字	WORD	可变

N：字数

指定要传送的字的数目。N 的范围为 0000 ~ FFFF(十进制的 0 ~ 65,535)。





● 区域类型

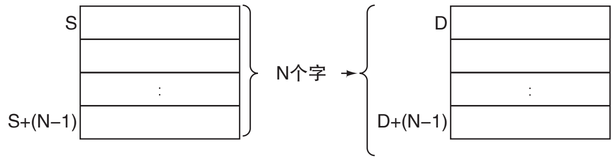
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
S，D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF

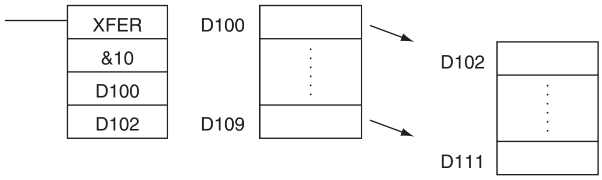
功能

XFER(070) 将从 S 开始的 N 个字 (S ～ S+(N−1)) 复制到从 D 开始的 N 个字 (D ～ D+(N−1))。



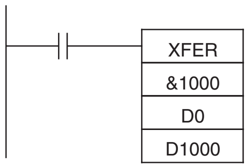
提示

- 源字和目的字可以重叠，因此 XFER(070) 指令可执行字移动操作。
- 指定的源和目的数据区可重叠 (字移动)。



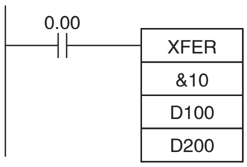
注意事项

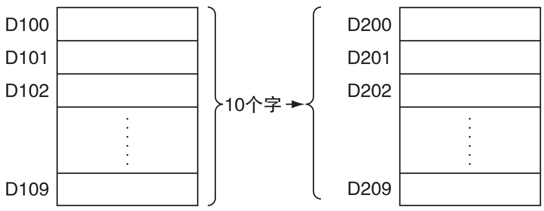
- 务必确保源字 (S ～ S+N−1) 和目的字 (D ～ D+N−1) 不超出数据区的末端。
- 当传送大量的字时，完成 XFER(070) 指令的执行需要一定的时间。即时发生中断，该指令的执行也不会中断，且中断任务将在 XFER(070) 指令执行完之后开始执行。如果在 XFER(070) 指令执行期间发生电源中断，则可能无法完成指令的执行，即可能无法传送全部的指定数据。



程序举例

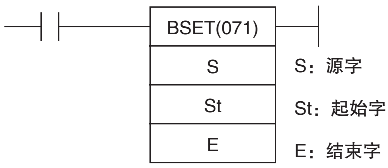
下例中，当 CIO 0.00 为 ON 时，将 D100 ～ D109 中的 10 个字的内容复制到 D200 ～ D209 中。





BSET

指令	助记符	变化	功能代码	功能
块设置	BSET	@BSET	071	将同一个字复制到一个连续字的范围中。

符号	BSET
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

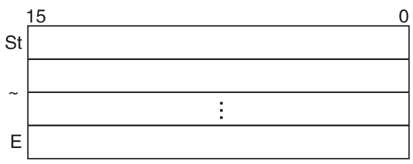
操作数	描述	数据类型	大小
S	源字	WORD	1
St	起始字	WORD	可变
E	结束字	WORD	可变

St：起始字

指定目的范围的起始字。

E：结束字

指定目的范围的结束字。



注 St 和 E 必须位于同一个数据区。

● 区域类型

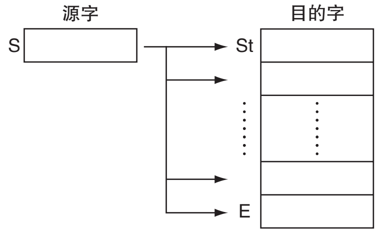
<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>S</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td>OK</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>St，E</td><td>---</td><td>---</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	· St 大于 E 时 ON。 · 其它情况下 OFF。

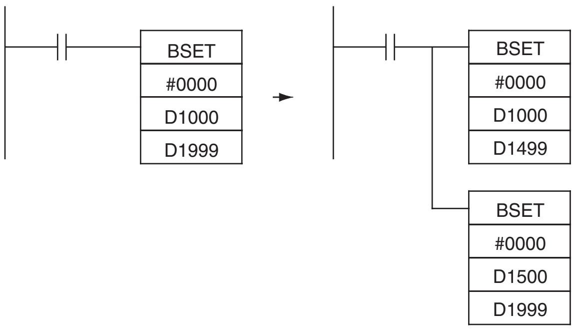
功能

BSET(071) 指令将同一个源字 (S) 复制到范围 St ~ E 中的所有目的字中。



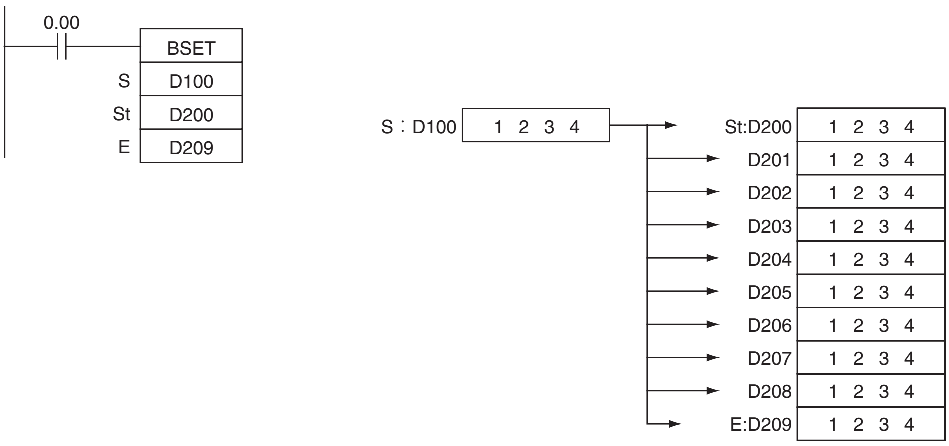
注意事项

· 当设定大量的字时，完成 BSET(071) 指令的执行需要一定的时间。即时发生中断，该指令的执行也不会中断，且中断任务将在 BSET(071) 指令执行完之后开始执行。如果在 BSET(071) 指令执行期间发生电源中断，则可能无法完成指令的执行，即可能无法设定全部的指定字。为避免该问题，可用两条 BSET(071) 指令来替代一条 BSET(071) 指令。



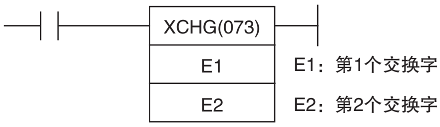
程序举例

下例中，当 CIO 0.00 为 ON 时，将 D100 中的源数据复制到 D200 ~ D209 中。



XCHG

指令	助记符	变化	功能代码	功能
数据交换	XCHG	@XCHG	073	交换两个指定字的内容。

符号	XCHG
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
E1	第 1 个交换字	WORD	1
E2	第 2 个交换字	WORD	1

● 区域类型

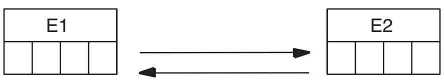
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
E1，E2	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	不变
等于标志	P_EQ	不变
负标志	P_N	不变

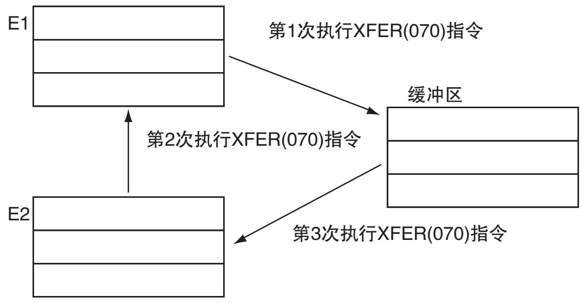
功能

XCHG(073) 指令将 E1 和 E2 中的内容进行交换。



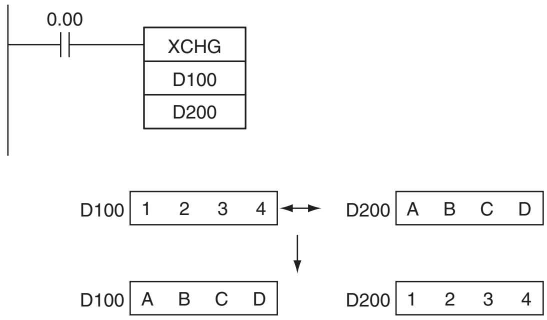
提示

若要交换3个或3个以上的字，可使用XFER(070)指令将字传送到第 3 组字 (缓冲区) 中，如下图所示。



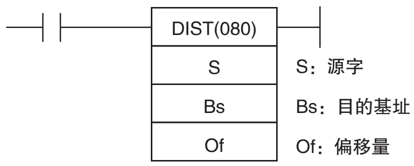
程序举例

下例中，当 CIO 0.00 为 ON 时，将 D100 的内容与 D200 的内容进行交换。



DIST

指令	助记符	变化	功能代码	功能
单字分配	DIST	@DIST	080	将源字传送到目的字 (在源基址上加一个偏移值)。

符号	DIST
	

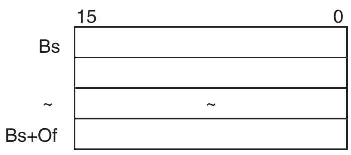
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	WORD	1
Bs	目的基址	WORD	1
Of	偏移量	UINT	1

Bs：目的基址



Of：偏移量

偏移量可以是 0000 ～ FFFF(十进制的 0 ～ 65,535) 之间的任何值。

注 Bs 和 Bs+Of 必须位于同一个数据区。

● 区域类型

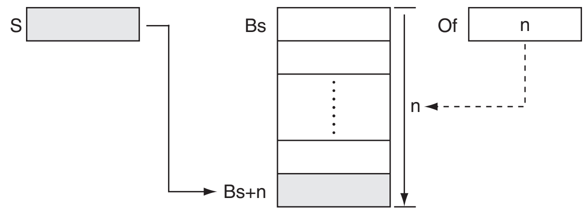
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S										OK	OK					
Bs	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
Of										OK	OK					

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 源数据为 0000 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 源数据的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

DIST(080) 指令将 S 复制到由 Of 和 Bs 相加计算得出的目的字中。



提示

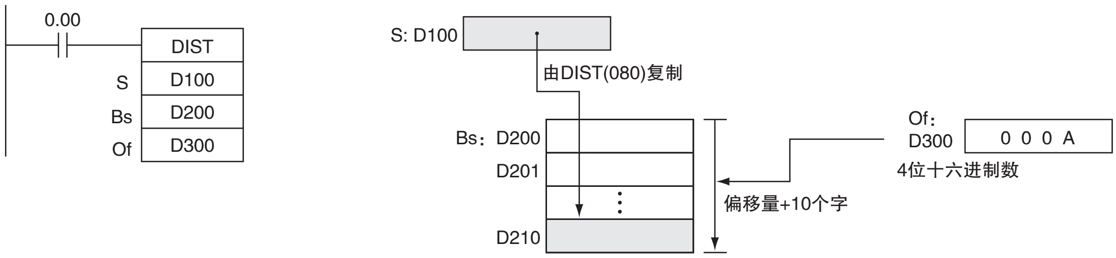
通过改变 Of 的值，可用同一条 DIST(080) 指令将源字分配到数据区中的多个字中。

注意事项

务必确保偏移量不超出数据区的末端，即 Bs 和 Bs+Of 必须在同一个数据区内。

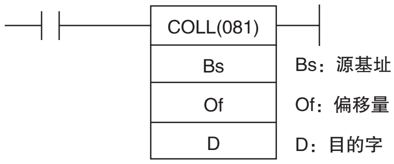
程序举例

下例中，当 CIO 0.00 为 ON 时，如果 D300 的内容为 10(十六进制的 0A)，则将 D100 的内容复制到 D210(D200+10) 中。通过改变 D300 中的偏移量，可将 D100 的内容复制到其它字中。



COLL

指令	助记符	变化	功能代码	功能
数据收集	COLL	@COLL	081	将源字 (在源基址上加一个偏移值) 传送到目的字。

符号	COLL
	

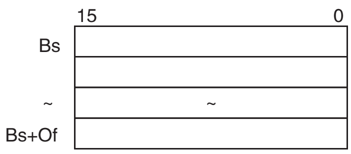
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
Bs	源基址	WORD	1
Of	偏移量	WORD	1
D	目的字	WORD	1

Bs：源基址



Of：偏移量

偏移量可以是 0000 ～ FFFF(十进制的 0 ～ 65,535) 之间的任何值。

注 Bs 和 Bs+Of 必须位于同一个数据区。

● 区域类型

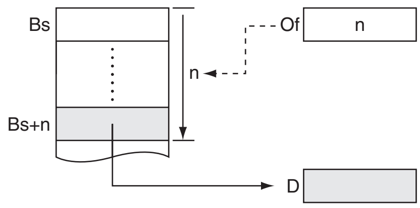
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
Bs	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
Of	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 源数据为 0000 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 源数据的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

COLL(081) 指令将源字 (由 Of 和 Bs 相加计算得出) 复制到目的字中。



提示

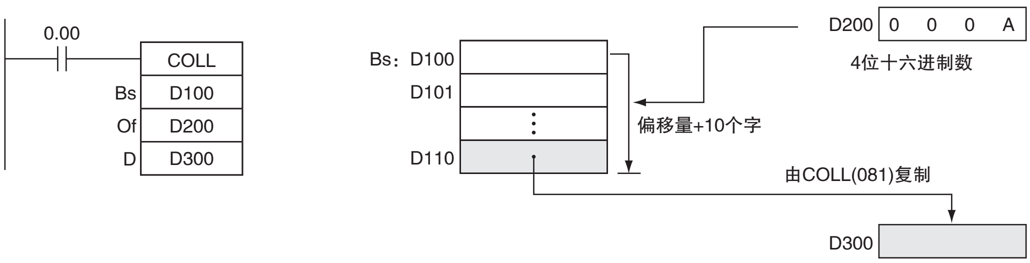
通过改变 Of 的值，可用同一条 COLL(081) 指令从数据区的多个源字中收集数据。

注意事项

务必确保偏移量不超出数据区的末端，即 Bs 和 Bs+Of 必须在同一个数据区内。

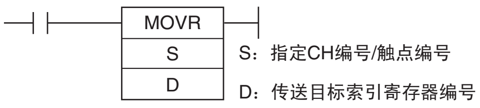
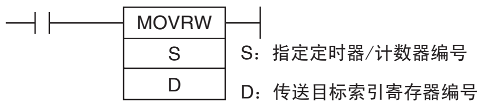
程序举例

下例中，当 CIO 0.00 为 ON 时，如果 D200 的内容为 10(十六进制的 0A)，则将 D110(D100+10) 的内容复制到 D300 中。通过改变 D200 中的偏移量，可将其它字的内容复制到 D300 中。



MOVR/MOVRW

指令	助记符	变化	功能代码	功能
索引 寄存器设定	MOVR	@MOVR	560	在指定的索引寄存器中设定指定 CH 编号、触点编号的 I/O 存储器有效地址。
索引 寄存器设定	MOVRW	@MOVRW	561	在指定的索引寄存器中设定指定定时器 / 计数器当前值的 I/O 存储器有效地址。

	MOVR	MOVRW
符号		

适用程序区

● MOVR

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

● MOVRW

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	OK	OK	OK

操作数

操作数	描述	数据类型		大小
		MOVR	MOVRW	
S	MOVR：指定 CH 编号 / 触点编号 MOVRW：指定定时器 / 计数器编号	BOOL	UINT	1
D	传送目标索引寄存器编号	WORD	WORD	2

● 区域种类

内容		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
MOVR	S	OK	OK	OK	OK	OK	OK	OK	---	---	---	---	---	---	---	---	---
	D	---	---	---	---	---	---	---					OK				
MOVRW	S	---	---	---	---	OK	OK	---	---	---	---	---	---	---	---	---	---
	D					---	---						OK				

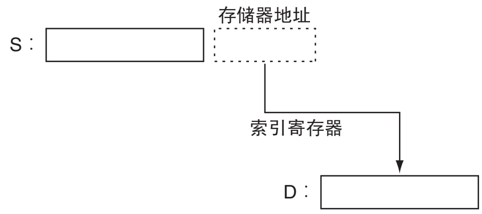
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	OFF
负标志	P_N	OFF

功能

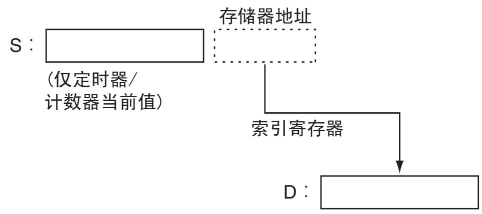
● MOVR

将 S 的 I/O 存储器有效地址传送到由 D 指定的索引寄存器中。



● MOVRW

将 S（定时器或计数器）的 I/O 存储器有效地址（定时器当前值区域或计数器当前值区域）传送到由 D 指定的索引寄存器中。



注意事项

可在 CP2E CPU 单元中使用 MOVR/MOVR 指令。不能在 CP1E CPU 单元中使用。

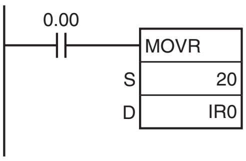
● MOVR

- 利用本指令设定索引寄存器 (IR0 ～ 15) 的 I/O 存储器有效地址 （定时器 / 计数器当前值除外）。
- 如果在通常的 I/O 存储器地址（各区域种类的地址）中指定 S，则自动将其转换为 I/O 存储器有效地址并保存到 D 中。
- 如果利用本指令在 S 中指定定时器 / 计数器，则将定时器 / 计数器上行标志的 I/O 存储器有效地址保存到 D 中。
- 利用 MOVRW 指令而非本指令设定定时器 / 计数器当前值的 I/O 存储器有效地址。
- 中断任务中的 IR 不确定。请务必在中断任务中使用本指令进行设定。另外，由中断任务执行的 IR/DR 不会对周期执行任务产生影响。

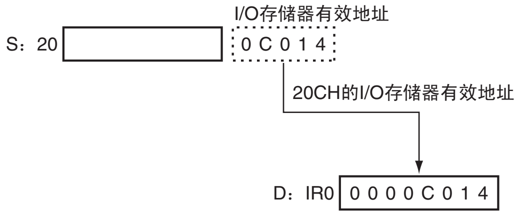
● MOVRW

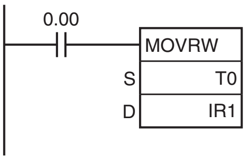
- 利用本指令设定索引寄存器 (IR0 ～ 15) 的定时器 / 计数器当前值的 I/O 存储器有效地址。
- 如果在通常的定时器 / 计数器地址中指定 S，则自动将其转换为定时器 / 计数器当前值的 I/O 存储器有效地址并保存到 D 中。
- 利用 MOVR 指令而非本指令设定定时器 / 计数器上行标志的 I/O 存储器有效地址。

程序举例

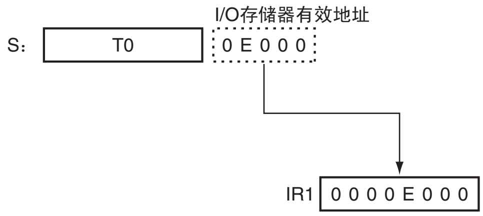


0.00为ON时，将20CH的I/O存储器有效地址保存到索引寄存器IR0中。





0.00为ON时，将定时器当前值T0000的I/O存储器有效地址保存到索引寄存器IR1中。

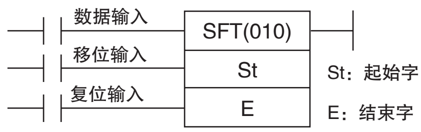


“SYSMAC CP 系列 CP2E CPU 单元用户手册软件篇”中记载有具体的 I/O 存储器有效地址。

数据移位指令

SFT

指令	助记符	变化	功能代码	功能
移位寄存器	SFT	---	010	操作移位寄存器。

	SFT
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
St	起始字	UINT	可变
E	结束字	UINT	可变

● 区域类型

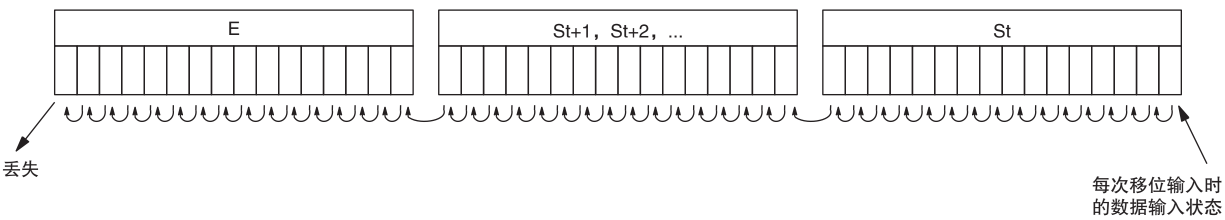
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
St，E	OK	OK	OK	---	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF

功能

・当移位输入的执行条件由 OFF 变 ON 时，St ~ E 的所有数据左移一位 (从最右位到最左位)，并且将数据输入的 ON/OFF 状态放在最右边位。



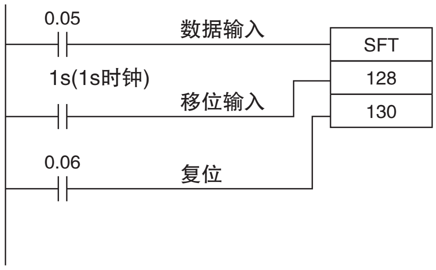
注意事项

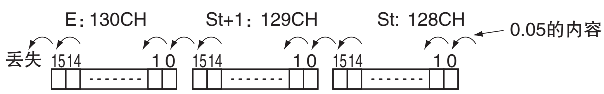
- 请勿使用一条以上的 SFT(010) 指令使移动的字重叠，否则结果将不确定。
- St 和 E 必须位于同一个数据区。
- 被移出移位寄存器的位数据将丢失。
- 当复位输入变 ON 时，移位寄存器中从最右边的指定字 (St) 到最左边的指定字 (E) 中的所有位将被复位 (即被置 0)。复位输入优先于其它输入。
- St 必须小于等于 E，但即使 St 被设为大于 E 也不会发生错误，并且 St 中数据的某个字将被移位。

程序举例

● 超出 16 位的移位寄存器

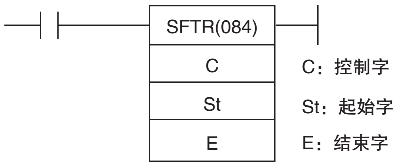
下例所示为一个使用字 CIO 128 ～ CIO 130 的 48 位移位寄存器。该寄存器使用 1s 时钟脉冲，这样每隔一秒钟，即会将由 CIO 0.05 产生的执行条件移入 CIO 128.00 和 CIO 130.15 之间的 3 字寄存器中。





SFTR

指令	助记符	变化	功能代码	功能
可逆移位寄存器	SFTR	@SFTR	084	生成一个即可使数据左移又可使数据右移的移位寄存器。

	SFTR
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
C	控制字	UINT	1
St	起始字	UINT	可变
E	结束字	UINT	可变

C：控制字

注 St 和 E 必须位于同一个数据区。

● **区域类型**

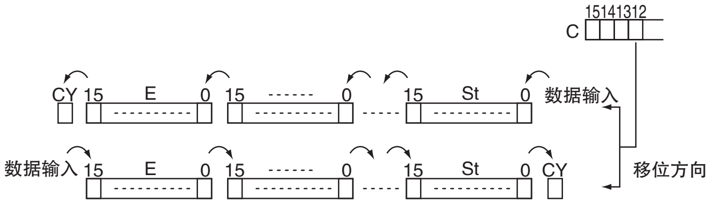
<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>C</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>St，E</td><td>---</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	· St 大于 E 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 移入了 1 时 ON。 · 移入了 0 时 OFF。 · 复位位被置为 1 时 OFF。

功能

当移位输入位 (C 中的位 14) 的执行条件变 ON 时，从 St ～ E 的所有数据均朝指定的移位方向 (由 C 中的位 12 指定) 移动 1 位，并且将数据输入的 ON/OFF 状态放在最右或最左位，而将被移出移位寄存器的位数据放在进位标志 (CY) 中。

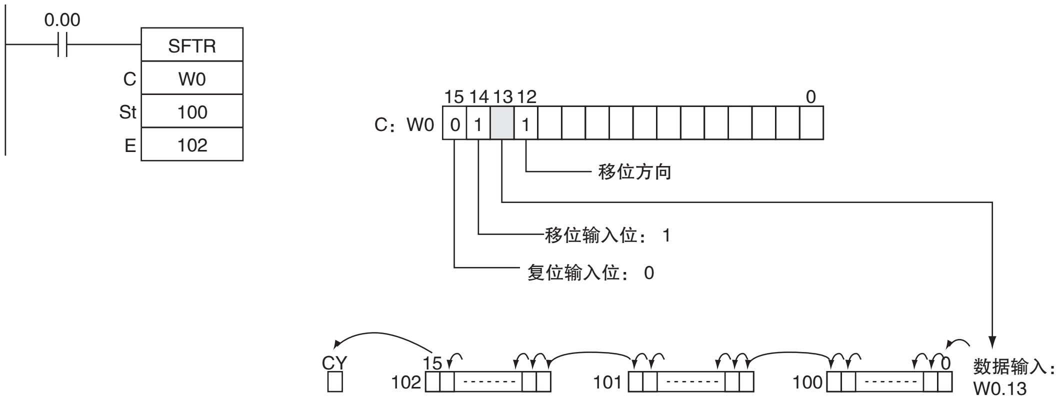


注 · 当复位位 (C 中的位 15) 置 OFF 时，即可使用上述移位操作。
· 当复位位 (C 中的位 15) 置 ON 时，移位寄存器中从 St ～ E 所有位将被复位 (即被置 0)。

程序举例

· 移位数据

当 CIO 0.00 为 ON 时，如果移位输入位 W0.14 为 ON 且复位位 W0.15 为 OFF，则从 CIO 100 ～ CIO 102 的字将朝由 W0.12 指定的方向移动一位 (例如 1 表示向右)，且输入位 W0.13 的内容将移入最右位 CIO 100.00 中。CIO 102.15 的内容将移到进位标志 (CY) 中。

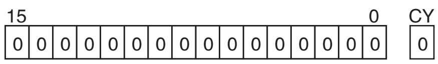


· 复位数据

当 CIO 0.00 为 ON 时，如果 W0.14 为 ON，且复位位 W0.15 为 ON，则字 CIO 100 ～ CIO 102 和进位标志将被复位为 OFF。

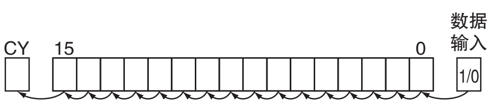
● 控制数据

复位数据



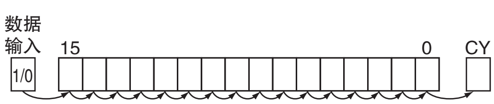
当复位输入位 (C 中的位 15) 为 ON 时，从 St ～ E 的所有位和进位标志均被置为 0，且不接受其它数据。

左移数据 (从最右位到最左位)



当移位输入位 (C 中的位 14) 为 ON 时，输入位 (C 中的位 13) 中的内容将被移到起始字的位 00 中，且其后的每一位均左移一位。结束字的位 15 的状态被移到进位标志中。

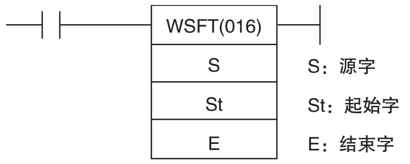
右移数据 (从最左位到最右位)



当移位输入位 (C 中的位 14) 为 ON 时，输入位 (C 中的位 13)(I/O) 中的内容将被移到结束字的位 15 中，且其后的每一位均右移一位。起始字的位 00 的状态被移到进位标志中。

WSFT

指令	助记符	变化	功能代码	功能
字移位	WSFT	@WSFT	016	在 St 和 E 之间以字为单位使数据移位。

	WSFT
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	控制字	WORD	1
St	起始字	UINT	可变
E	结束字	UINT	可变

● 区域类型

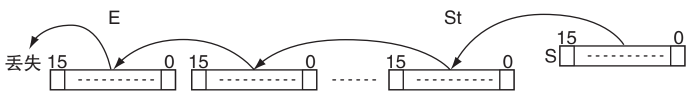
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
St，E	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· St 大于 E 时 ON。 · 其它情况下 OFF。

功能

WSFT(016) 以字为单位按从 St 到 E 的方向移动数据，且源字 S 中的数据将放在 St 中，而 E 中的数据将丢失。



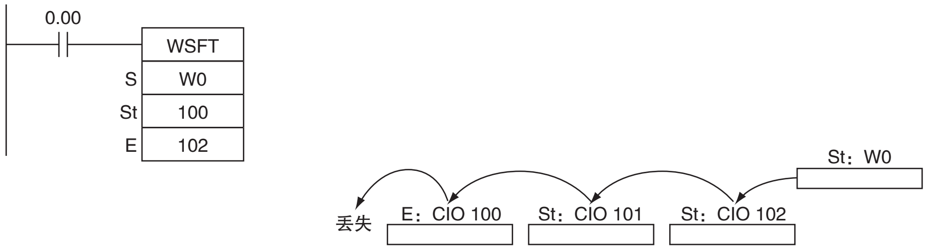
注意事项

· St 和 E 必须位于同一个数据区。

· 当移动大量数据时，指令执行的时间相当长。请务必确保在 WSFT(016) 指令执行期间不切断电源，以防止移位操作中途停止。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 ～ CIO 102 之间的数据将朝 E 移动一个字。W0 中的内容将被保存在 CIO 100 中，而 CIO 102 中的内容将会丢失。



ASL/ASLL

指令	助记符	变化	功能代码	功能
算术左移	ASL	@ASL	025	将 Wd 的内容向左移一位。
双字左移	ASLL	@ASLL	570	将 Wd 和 Wd+1 的内容向左移一位。

	ASL	ASLL
符号	ASL(025) Wd　Wd：字	ASLL(570) Wd　Wd：字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ASL	ASLL	ASL	ASLL
Wd	字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ASL	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
ASLL	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

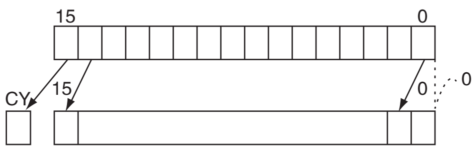
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	・移位结果为 0 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・将 1 移入进位标志 (CY) 时 ON。 ・其它情况下 OFF。
负标志	P_N	・移位结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

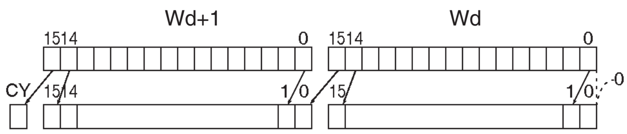
● ASL

ASL(025) 将 Wd 的内容左移一位 (从最右位到最左位)。在最右位中放置 “0”，并将最左位的内容移到进位标志 (CY) 中。



● ASLL

ASLL(570) 将 Wd 和 Wd+1 的内容左移一位 (从最右位到最左位)。在 Wd 的最右位中放置 “0”，并将 Wd 和 Wd+1 最左位的数据移到进位标志 (CY) 中。

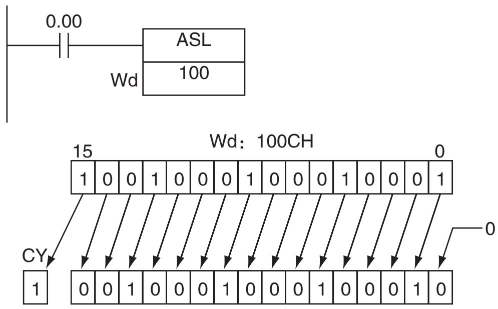


注意事项

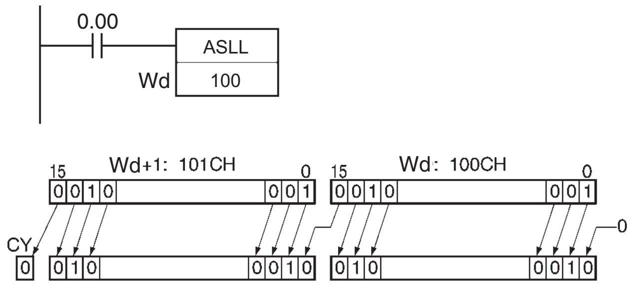
可在 CP2E CPU 单元中使用 ASLL(570) 指令。不能在 CP1E CPU 单元中使用。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 将向左移一位。将在 CIO 100.00 中放置“0”，并将 CIO 100.15 的内容移到进位标志 (CY) 中。



当 CIO 0.00 为 ON 时，CIO 100 和 CIO 101 将向左移一位。将在 CIO 100.00 中放置 “0”，并将 CIO 101.15 的内容移到进位标志 (CY) 中。



ASR/ASRL

指令	助记符	变化	功能代码	功能
算术右移	ASR	@ASR	026	将 Wd 的内容向右移一位。
双字右移	ASRL	@ASRL	571	将 Wd 和 Wd+1 的内容向右移一位。

	ASR	ASRL
符号	ASR(026) Wd Wd：字	ASRL(571) Wd Wd：字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ASR	ASRL	ASR	ASRL
Wd	字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ASR	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
ASRL												---					

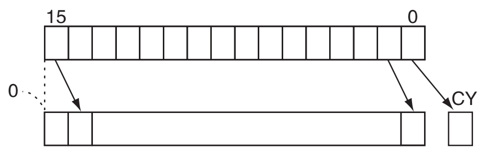
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 移位结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 将 1 移入进位标志 (CY) 时 ON。 · 其它情况下 OFF。
负标志	P_N	OFF

功能

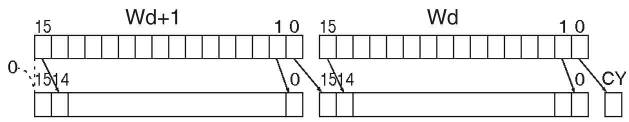
● ASR

ASR(026) 将 Wd 的内容右移一位 (从最左位到最右位)。在最左位中放置 “0”，并将最右位的内容移到进位标志 (CY) 中。



● ASRL

ASRL(571) 将 Wd 和 Wd+1 的内容右移一位 (从最左位到最右位)。在 Wd 的最左位中放置 “0”，并将 Wd 和 Wd+1 最右位的数据移到进位标志 (CY) 中。

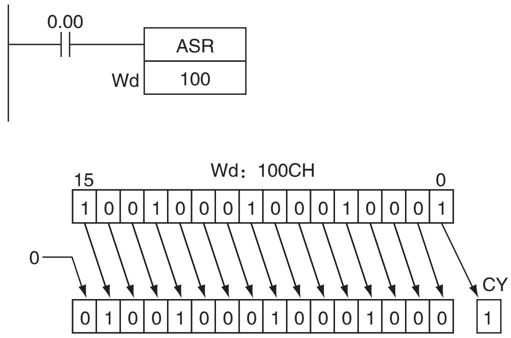


注意事项

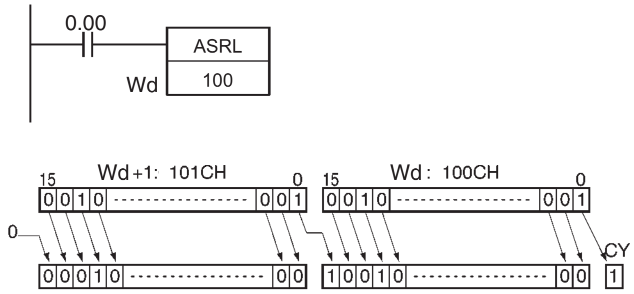
可在 CP2E CPU 单元中使用 ASRL(571) 指令。不能在 CP1E CPU 单元中使用。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 将向右移一位。将在 CIO 100.15 中放置 “0”，并将 CIO 100.00 的内容移到进位标志 (CY) 中。

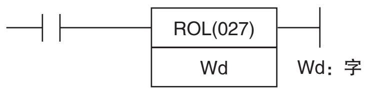
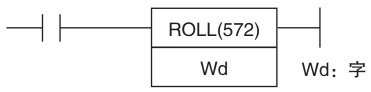


当 CIO 0.00 为 ON 时，CIO 100 和 CIO 101 将向右移一位。将在 CIO 100.15 中放置 “0”，并将 CIO 100.00 的内容移到进位标志 (CY) 中。



ROL/ROLL

指令	助记符	变化	功能代码	功能
循环左移	ROL	@ROL	027	将 Wd 中包括进位标志 (CY) 在内的所有位向左移一位。
双字循环左移	ROLL	@ROLL	572	将Wd和Wd+1中包括进位标志(CY)在内的所有位向左移一位。

符号	ROL	ROLL
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ROL	ROLL	ROL	ROLL
Wd	字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ROL	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
ROLL												---					

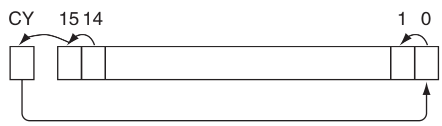
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	・移位结果为 0 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・将 1 移入进位标志 (CY) 时 ON。 ・其它情况下 OFF。
负标志	P_N	・移位结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

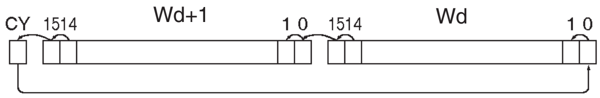
● ROL

ROL(027) 指令将 Wd 中包括进位标志 (CY) 在内的所有位向左移一位(从最右位到最左位)。



● ROLL

ROLL(572) 指令将 Wd 和 Wd+1 中包括进位标志 (CY) 在内的所有位向左移一位 (从最右位到最左位)。



提示

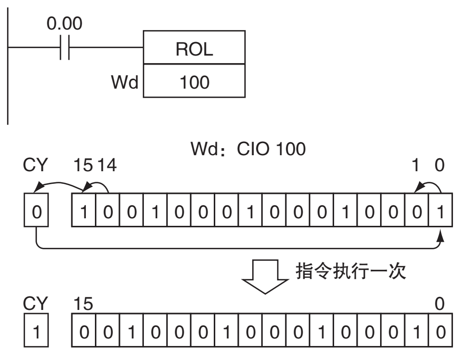
通过使用设置进位 (STC(040)) 或清除进位 (CLC(041)) 指令，可以在即将执行该指令之前，将进位标志的内容置为 1 或 0。

注意事项

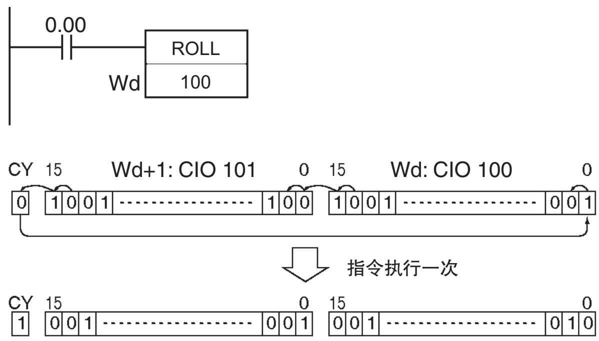
可在 CP2E CPU 单元中使用 ROLL(572) 指令。不能在 CP1E CPU 单元中使用。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 和进位标志 (CY) 将向左移一位。CIO 100.15 的内容将被移到进位标志 (CY) 中，且进位标志的内容将被移到 CIO 100.00 中。



当 CIO 0.00 为 ON 时，CIO 100、CIO 101 和进位标志 (CY) 将向左移一位。CIO 100.15 的内容将被移到进位标志 (CY) 中，且进位标志的内容将被移到 CIO 100.00 中。



ROR/RORL

指令	助记符	变化	功能代码	功能
循环右移	ROR	@ROR	028	将 Wd 中包括进位标志 (CY) 在内的所有位向右移一位。
双字循环右移	RORL	@RORL	573	将 Wd 和 Wd+1 中包括进位标志 (CY) 在内的所有位向右移一位。

	ROR	RORL
符号	ROR(028) Wd　Wd：字	RORL(573) Wd　Wd：字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ROR	RORL	ROR	RORL
Wd	字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ROR	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
RORL	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

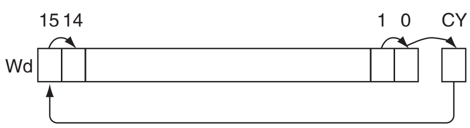
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	・移位结果为 0 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・将 1 移入进位标志 (CY) 时 ON。 ・其它情况下 OFF。
负标志	P_N	・移位结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

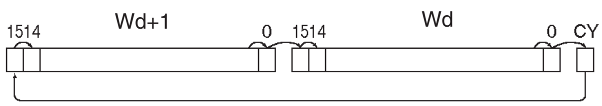
● ROR

ROR(028) 指令将 Wd 中包括进位标志 (CY) 在内的所有位向右移一位(从最左位到最右位)。



● RORL

RORL(573) 指令将 Wd 和 Wd+1 中包括进位标志 (CY) 在内的所有位向右移一位 (从最左位到最右位)。



提示

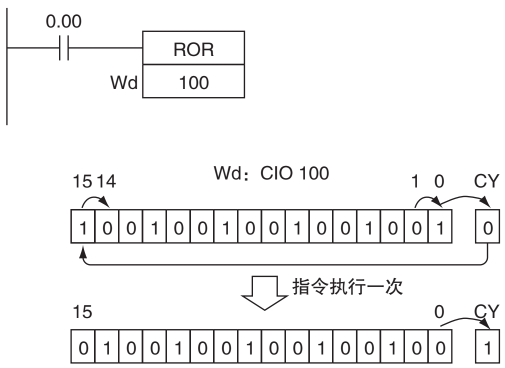
通过使用设置进位 (STC(040)) 或清除进位 (CLC(041)) 指令，可以在即将执行该指令之前，将进位标志的内容置为 1 或 0。

注意事项

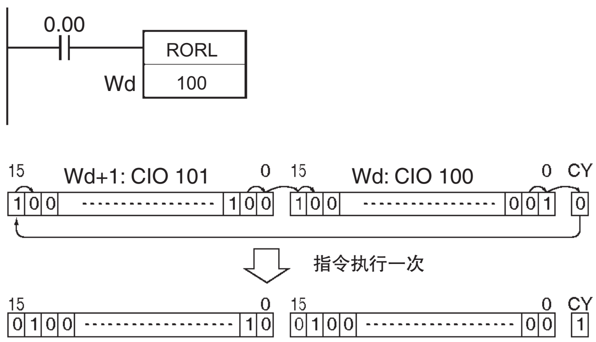
可在 CP2E CPU 单元中使用 RORL(573) 指令。不能在 CP1E CPU 单元中使用。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 和进位标志 (CY) 将向右移一位。CIO 100.00 的内容将被移到进位标志 (CY) 中，且进位标志的内容将被移到 CIO 100.15 中。

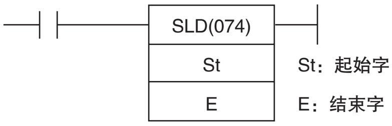
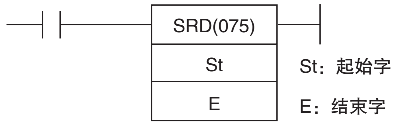


当 CIO 0.00 为 ON 时，CIO 100、CIO 101 和进位标志 (CY) 将向右移一位。CIO 101.00 的内容将被移到进位标志 (CY) 中，且进位标志的内容将被移到 CIO 101.15 中。



SLD/SRD

指令	助记符	变化	功能代码	功能
一个数位左移	SLD	@SLD	074	将数据左移一个数位 (4 个位)。
一个数位右移	SRD	@SRD	075	将数据右移一个数位 (4 个位)。

	SLD	SRD
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
St	起始字	UINT	可变
E	结束字	UINT	可变

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
St，E	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· St 大于 E 时 ON。 · 其它情况下 OFF。

功能

● SLD

SLD(074) 指令将 St ～ E 之间的数据向左移一个数位 (4 个位)。将 “0” 放置在最右边的数位 (St 的位 3 ～ 0)，而最左边数位 (E 的位 15 ～ 12) 的内容将丢失。

● SRD

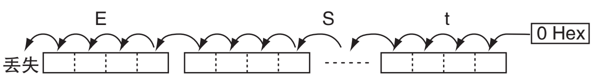
SRD(075) 指令将 St ～ E 之间的数据向右移一个数位 (4 个位)。将 “0” 放置在最左边的数位 (E 的位 15 ～ 12)，而最右边数位 (St 的位 3 ～ 0) 的内容将丢失。

注意事项

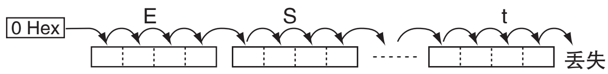
· St 和 E 必须位于同一个数据区。

· 当移动大量数据时，指令执行的时间相当长。请务必确保在 SLD(074) 和 SRD(075) 指令执行期间不切断电源，以防止移位操作中途停止。

■SLD



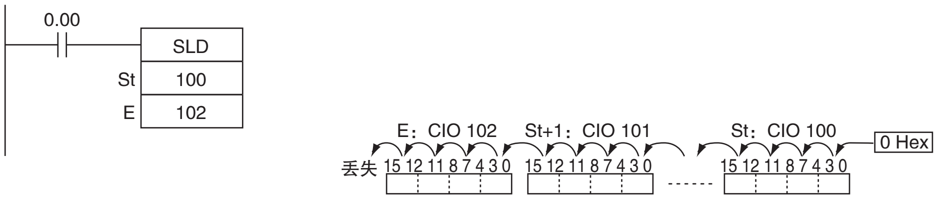
■SRD



程序举例

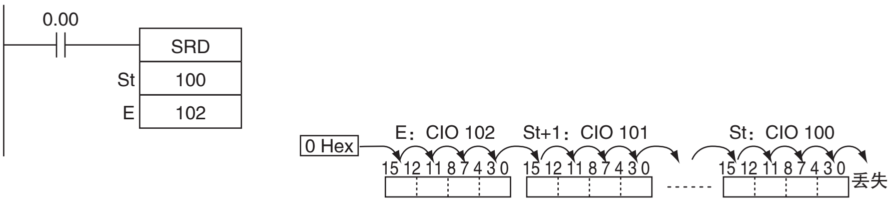
● SLD

当 CIO 0.00 为 ON 时，字 CIO 100 ～ CIO 102 将向左移一个数位 (4 个位)。将在字 CIO 100 的位 0 ～ 3 中放置 0，而 CIO 102 的位 12 ～ 15 的内容将丢失。



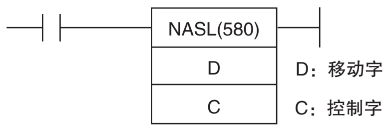
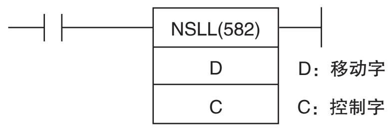
● SRD

当 CIO 0.00 为 ON 时，字 CIO 100 ～ CIO 102 将向右移一个数位 (4 个位)。将在 CIO 102 的位 12 ～ 15 中放置 0，而字 CIO 100 的位 0 ～ 3 的内容将丢失。



NASL/NSLL

指令	助记符	变化	功能代码	功能
左移 N 位	NASL	@NASL	580	将指定的 16 位字数据左移指定的位数。
双字左移 N 位	NSLL	@NSLL	582	将指定的 32 位字数据左移指定的位数。

	NASL	NSLL
符号		

适用程序区

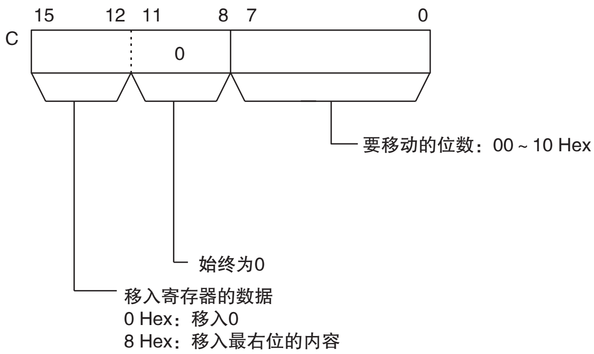
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

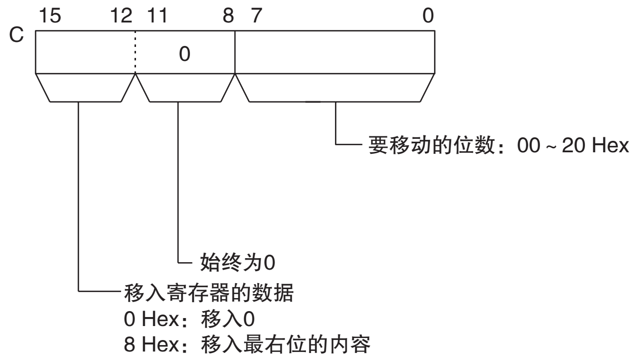
操作数	描述	数据类型		大小	
		NASL	NSLL	NASL	NSLL
D	移动字	UINT	UDINT	1	2
C	控制字	UINT	UDINT	1	1

C：控制字

■ NASL



■ NSLL



● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
NASL	D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
	C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
NSLL	D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
	C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

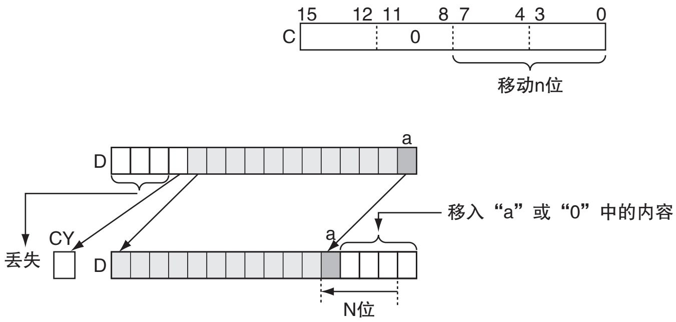
条件标志

名称	标记	操作
出错标志	P_ER	・当控制字 C(要移动的位数) 不在范围内时 ON。 ・其它情况下 OFF。
等于标志	P_EQ	・移位结果为 0 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・将 1 移入进位标志 (CY) 时 ON。 ・其它情况下 OFF。
负标志	P_N	・移位结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

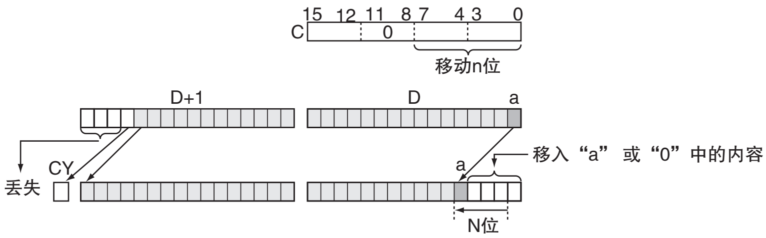
● NASL

NASL(580) 将 D(移动字) 向左 (从最右位到最左位) 移动指定的二进制位数 (在 C 中指定)。将在移动字从最右位开始的指定位数中放置零或最右位的值。



● NSLL

NSLL(582) 将 D 和 D+1(移动字) 向左 (从最右位到最左位) 移动指定的二进制位数 (在 C 中指定)。将在移动字从最右位开始的指定位数中放置零或最右位的值。



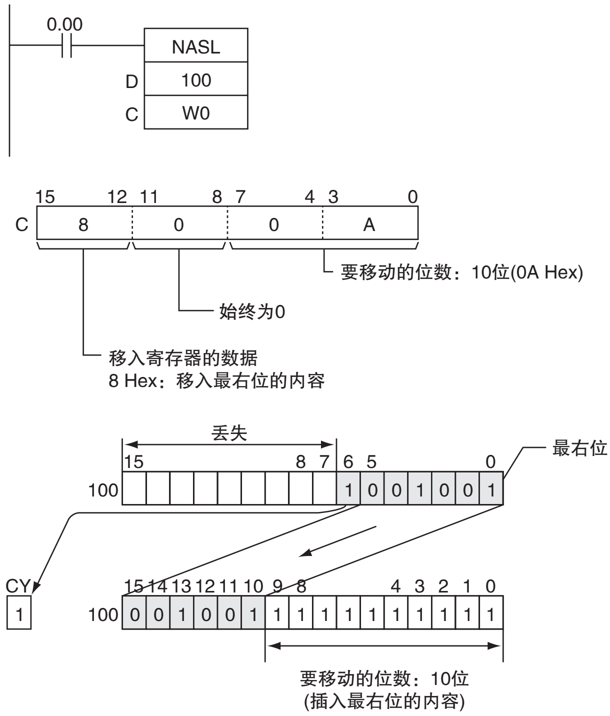
注意事项

・对于移出指定字的位，最后一位的内容将移到进位标志 (CY) 中，而其它所有数据将丢失。

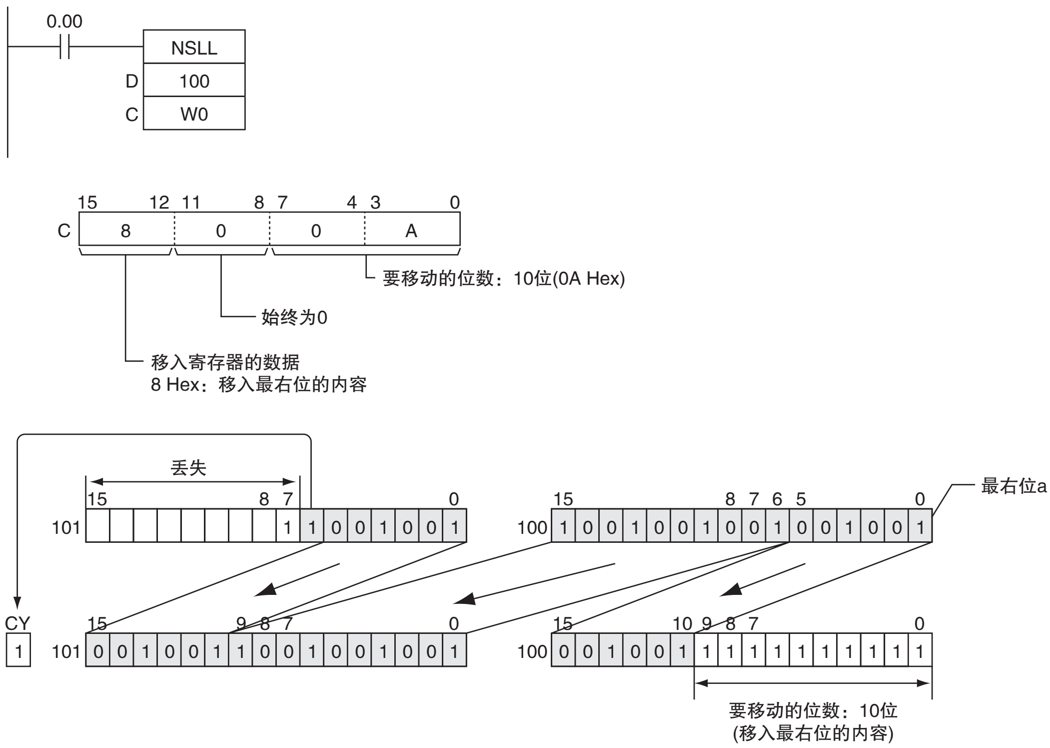
・当要移动的位数 (在 C 中指定) 为 “0”时，数据将不移动。但是根据指定字中的数据而定，相应的标志将变 ON 或变 OFF。

程序举例

当 CIO 0.00 为 ON 时，CIO 100 的内容将向左 (从最右位到最左位) 移 10 位。要移动的位数在字 W0(控制数据) 的位 0 ～ 7 中指定。CIO 100 的位 0 中的内容将被复制到数据被移动的位中，且被移出范围的最右位的内容将被移到进位标志 (CY) 中，所有其它数据则丢失。



当 CIO 0.00 为 ON 时，CIO 100 和 CIO 101 将向左 (从最右位到最左位) 移 10 位。要移动的位数在 W0(控制数据) 的位 0 ～ 7 中指定。CIO 100 的位 0 中的内容将被复制到数据被移动的位中，且被移出范围的最右位的内容将被移到进位标志 (CY) 中，所有其它数据则丢失。



NASR/NSRL

指令	助记符	变化	功能代码	功能
右移 N 位	NASR	@NASR	581	将指定的 16 位字数据右移指定的位数。
双字右移 N 位	NSRL	@NSRL	583	将指定的 32 位字数据右移指定的位数。

	NASR	NSRL
符号	NASR(581) D　D：移动字 C　C：控制字	NSRL(583) D　D：移动字 C　C：控制字

适用程序区

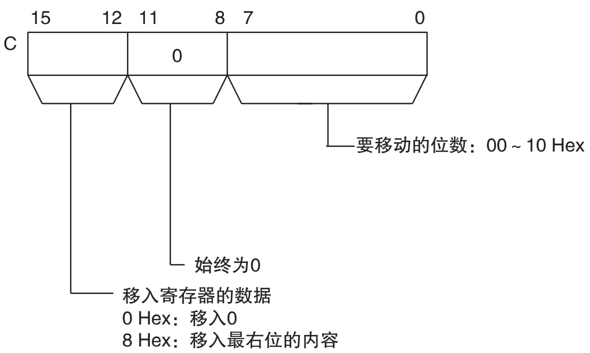
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

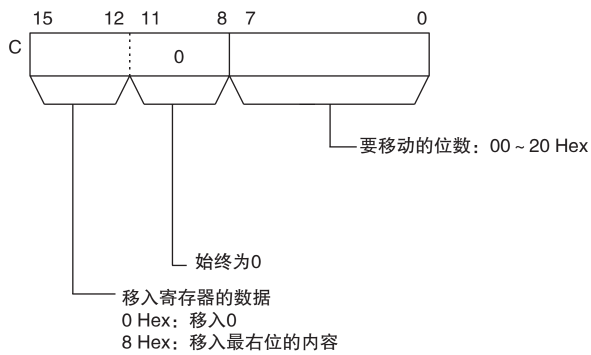
操作数	描述	数据类型		大小	
		NASR	NSRL	NASR	NSRL
D	移动字	UINT	UDINT	1	2
C	控制字	UINT	UDINT	1	1

C：控制字

■ NASR



■ NSRL



● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
NASR	D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
	C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
NSRL	D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
	C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

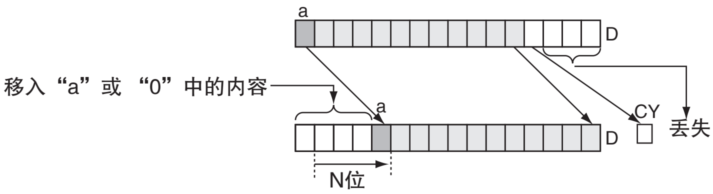
条件标志

名称	标记	操作
出错标志	P_ER	・当控制字 C(要移动的位数) 不在范围内时 ON。 ・其它情况下 OFF。
等于标志	P_EQ	・移位结果为 0 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・将 1 移入进位标志 (CY) 时 ON。 ・其它情况下 OFF。
负标志	P_N	・移位结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

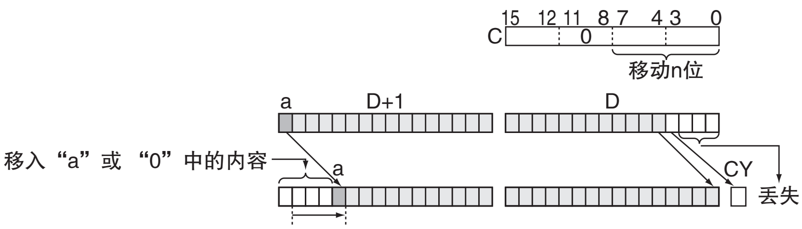
● NASR

NASR(581) 将 D(移动字) 向右 (从最右位到最左位) 移动指定的二进制位数 (在 C 中指定)。将在移动字从最右位开始的指定位数中放置零或最右位的值。



● NSRL

NSRL(583) 将 D 和 D+1(移动字) 向右 (从最左位到最右位) 移动指定的二进制位数 (在 C 中指定)。将在移动字从最右位开始的指定位数中放置零或最右位的值。



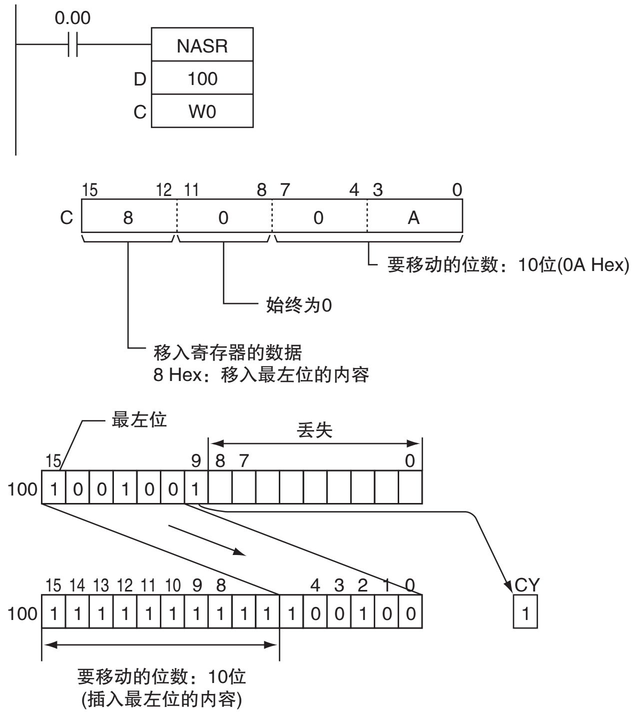
注意事项

・ 对于移出指定字的位，最后一位的内容将移到进位标志 (CY) 中，而其它所有数据将丢失。

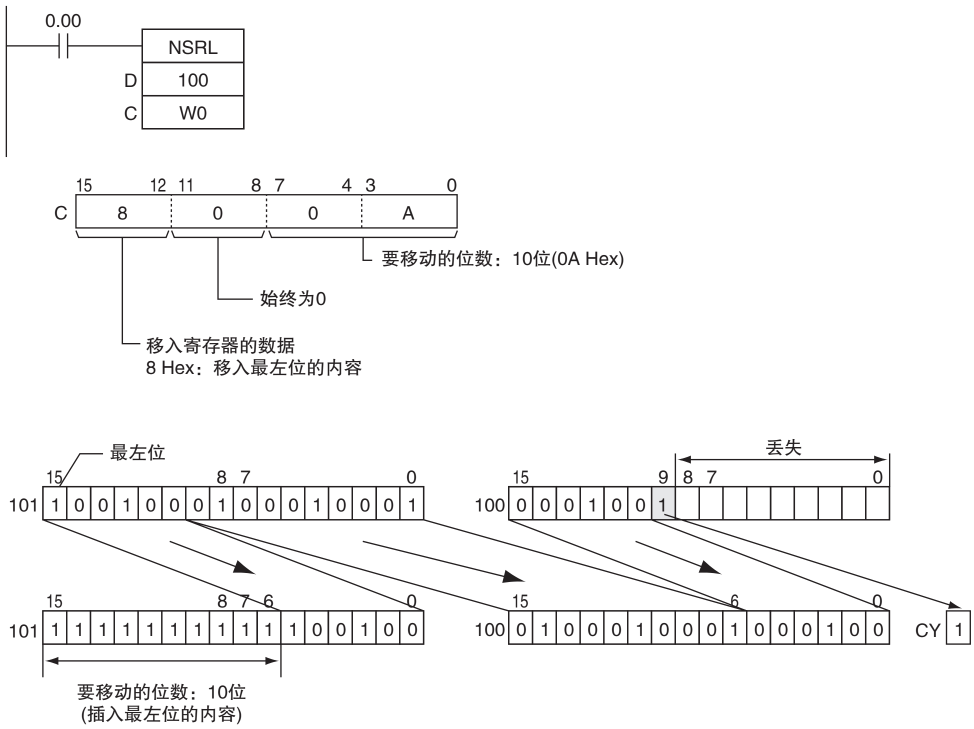
・ 当要移动的位数 (在 C 中指定) 为 “0”时，数据将不移动。但是根据指定字中的数据而定，相应的标志将变 ON 或变 OFF。

程序举例

- 当 CIO 0.00 为 ON 时，CIO 100 的内容将向右 (从最左位到最右位) 移 10 位。要移动的位数在 W0 的位 0 ～ 7 中指定。CIO 100 的位 15 中的内容将被复制到数据被移动的位中，且被移出范围的数据最左位的内容将被移到进位标志 (CY) 中，所有其它数据则丢失。



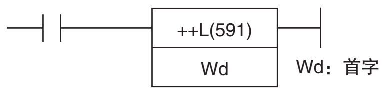
当 CIO 0.00 为 ON 时，CIO 100 和 CIO 101 的内容将向右 (从最左位到最右位) 移 10 位。要移动的位数在 W0(控制数据) 的位 0 ～ 7 中指定。CIO 的位 15 中的内容将被复制到数据被移动的位中，且被移出范围的数据最左位的内容将被移到进位标志 (CY) 中，所有其它数据则丢失。



递增 / 递减指令

++/++L

指令	助记符	变化	功能代码	功能
二进制递增	++	@++	590	将指定字的 4 位数十六进制内容递增 1。
双字二进制递增	++L	@++L	591	将指定字的 8 位数十六进制内容递增 1。

	++	++L
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		++	++L	++	++L
Wd	++：字 ++L：首字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
++	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
++L	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

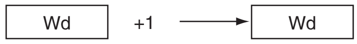
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 执行后结果为 0000/0000 0000 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 执行期间若 Wd/Wd+1 或 Wd 中的某个数位从 F 变为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 执行后若 Wd/Wd+1 的位 15 为 ON 时 ON。 · 其它情况下 OFF。

功能

● ++

++(590) 指令使 Wd 的二进制内容加 1。只要 ++(590) 的执行条件为 ON，每次循环时指定的字均会递增 1。当使用该指令的上升沿微分变化 (@++(590)) 时，仅在执行条件从 OFF 变 ON 时指定字递增。



● ++L

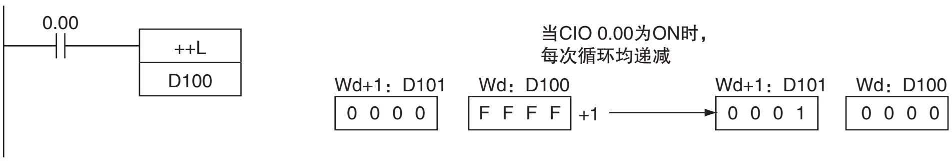
++L(591) 指令使 Wd+1 和 Wd 的 8 位数十六进制内容加 1。只要 ++L(591) 的执行条件为 ON，每次循环时指定字的内容均会递增 1。当使用该指令的上升沿微分变化 (@++L(591)) 时，仅在执行条件从 OFF 变 ON 时指定字的内容递增。

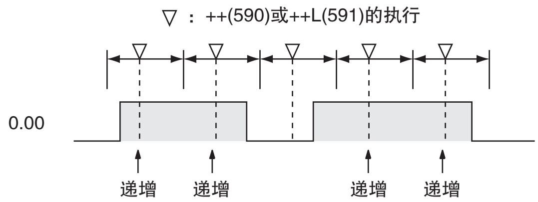
程序举例

● ++(590)/++L(591) 的操作

下例中，只要 CIO 0.00 为 ON，每次循环时 D100 的内容均会递增 1。

下例中，只要 CIO 0.00 为 ON，每次循环时 D100 的内容均会递增 1。

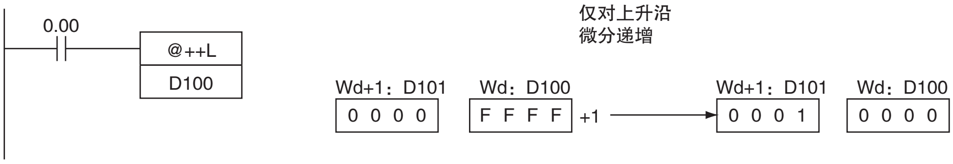


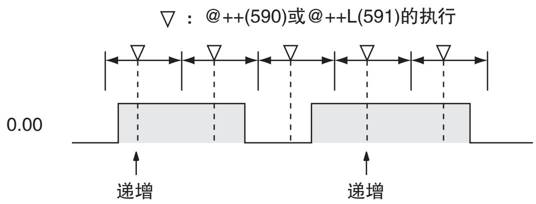


- @++(590)/@++L(591) 的操作

下例中使用了上升沿微分变化，因此此当 CIO 0.00 从 OFF 变 ON 时，D100 的内容将递增 1。

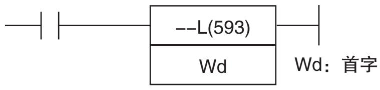
下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D101 和 D100 的内容将递增 1。





--/--L

指令	助记符	变化	功能代码	功能
二进制递减	--	@--	592	将指定字的 4 位数十六进制内容递减 1。
双字二进制递减	--L	@--L	593	将指定字的 8 位数十六进制内容递减 1。

	--	--L
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		--	--L	--	--L
Wd	--：字 --L：首字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
--	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
--L												---					

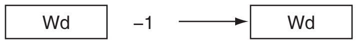
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 执行后结果为 0000/0000 0000 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 执行期间若 Wd/Wd+1 或 Wd 中的某个数位从 0 变为 F 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 执行后若 Wd/Wd+1 的位 15 为 ON 时 ON。 · 其它情况下 OFF。

功能

● --

--(592) 指令使 Wd 的二进制内容减 1。只要 --(592) 的执行条件为 ON，每次循环时指定的字均会递减 1。当使用该指令的上升沿微分变化 (@--(592)) 时，仅在执行条件从 OFF 变 ON 时指定字递减。



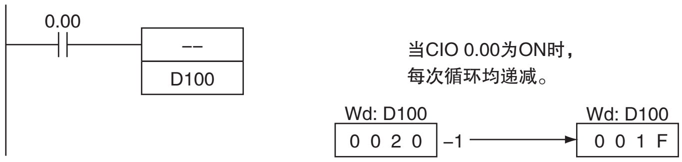
● --L

--L(593) 指令使 Wd+1 和 Wd 的 8 位数十六进制内容减 1。只要 --L(593) 的执行条件为 ON，每次循环时指定字的内容均会递减 1。当使用该指令的上升沿微分变化 (@--L(593)) 时，仅在执行条件从 OFF 变 ON 时指定字的内容递增减。

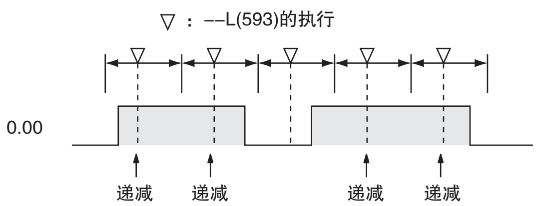
程序举例

● --(592)/--L(593) 的操作

下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D100 的内容将递减 1。

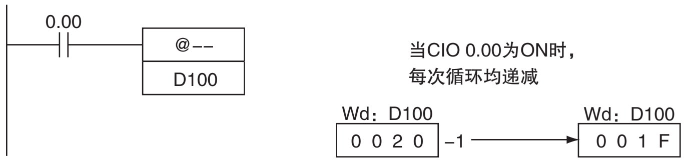


下例中，只要 CIO 0.00 为 ON，每次循环时 D101 和 D100 的 8 位数十六进制内容均会递减 1。

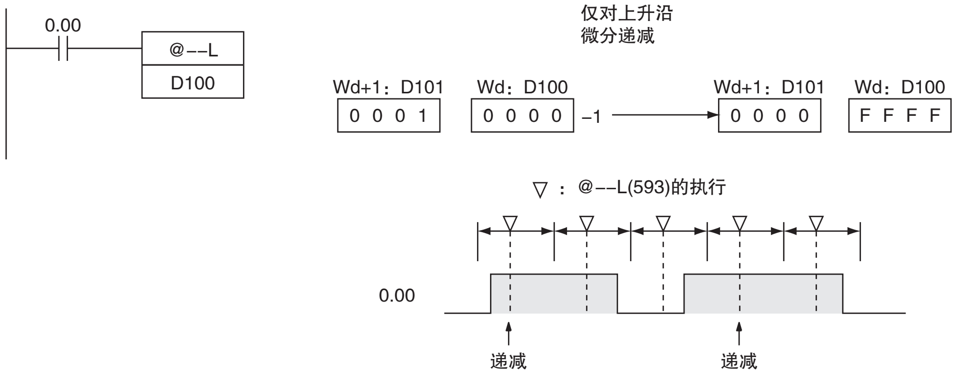


- **@−−(592)/@−−L(593) 的操作**

下例中，只要 CIO 0.00 为 ON，每次循环时 D100 的内容均会递减 1。

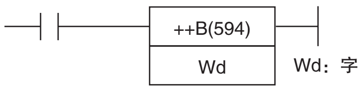
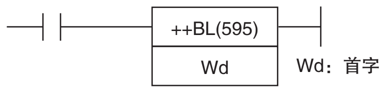


下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D101 和 D100 的内容将递减 1。



++B/++BL

指令	助记符	变化	功能代码	功能
BCD 递增	++B	@++B	594	将指定字的 4 位数 BCD 内容递增 1。
双字 BCD 递增	++BL	@++BL	595	将指定字的 8 位数 BCD 内容递增 1。

	++B	++BL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		++B	++BL	++B	++BL
Wd	++B：字 ++BL：首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
++B	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
++BL												---					

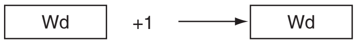
条件标志

名称	标记	操作
出错标志	P_ER	・Wd/Wd+1 和 Wd 的内容不是 BCD 为 ON。 ・其它情况下 OFF。
等于标志	P_EQ	・执行后结果为 0000/0000 0000 时 ON。 ・其它情况下 OFF。
进位标志	P_CY	・执行期间若 Wd/Wd+1 或 Wd 中的某个数位从 9 变为 0 时 ON。 ・其它情况下 OFF。

功能

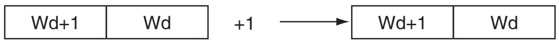
● ++B

++B(594) 指令使 Wd 的 BCD 内容加 1。只要 ++B(594) 的执行条件为 ON，每次循环时指定的字均会递增 1。当使用该指令的上升沿微分变化 (@++B(594)) 时，仅在执行条件从 OFF 变 ON 时指定字递增。



● ++BL

++BL(595) 指令使 Wd+1 和 Wd 的 8 位数 BCD 内容加 1。只要 ++BL(595) 的执行条件为 ON，每次循环时指定字的内容均会递增 1。当使用该指令的上升沿微分变化 (@++BL(595)) 时，仅在执行条件从 OFF 变 ON 时指定字的内容递增。

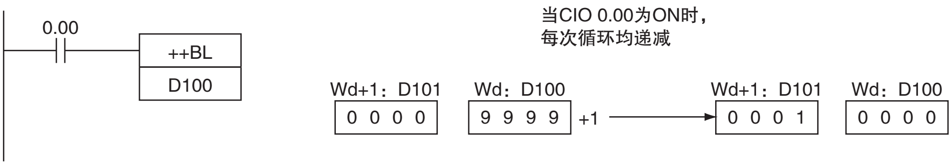


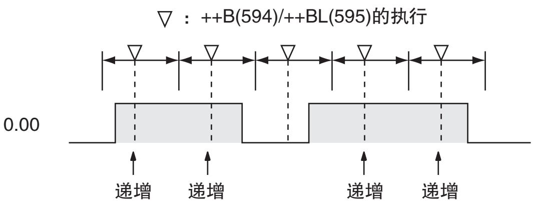
程序举例

● ++B(594)/++BL(595) 的操作

下例中，只要 CIO 0.00 为 ON，每次循环时 D100 的 BCD 内容均会递增 1。

下例中，只要 CIO 0.00 为 ON，每次循环时 D101 和 D100 的 8 位数 BCD 内容均会递增 1。

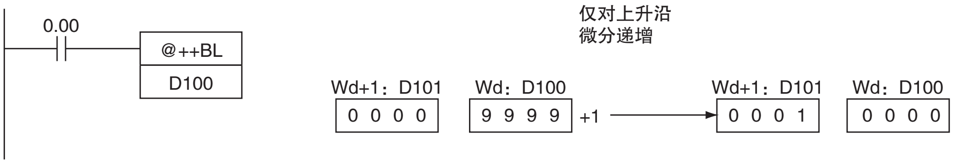


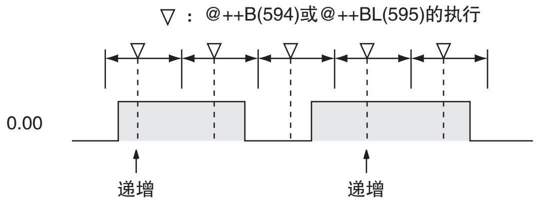


● @++B(594)/@++BL(595) 的操作

下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D100 的内容将递增 1。

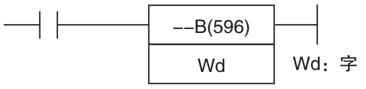
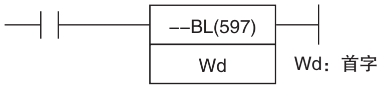
下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D101 和 D100 的 BCD 内容将递增 1。





--B/--BL

指令	助记符	变化	功能代码	功能
BCD 递减	--B	@--B	596	将指定字的 4 位数 BCD 内容递减 1。
双字 BCD 递减	--BL	@--BL	597	将指定字的 8 位数 BCD 内容递减 1。

	--B	--BL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		--B	--BL	--B	--BL
Wd	--B：字 --BL：首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
--B	Wd	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
--BL												---					

条件标志

名称	标记	操作
出错标志	P_ER	· Wd/Wd+1 和 Wd 的内容不是 BCD 为 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 执行后结果为 0000/0000 0000 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 执行期间若 Wd/Wd+1 或 Wd 中的某个数位从 0 变为 9 时 ON。 · 其它情况下 OFF。

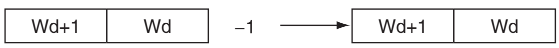
功能

● --B

--B(596) 指令使 Wd 的 BCD 内容减 1。只要 --B(596) 的执行条件为 ON，每次循环时指定的字均会递减 1。当使用该指令的上升沿微分变化 (@--B(596)) 时，仅在执行条件从 OFF 变 ON 时指定字递减。

● --BL

--BL(597) 指令使 Wd+1 和 Wd 的 8 位数 BCD 内容减 1。只要 --BL(597) 的执行条件为 ON，每次循环时指定字的内容均会递减 1。当使用该指令的上升沿微分变化 (@--BL(597)) 时，仅在执行条件从 OFF 变 ON 时指定字的内容递减。

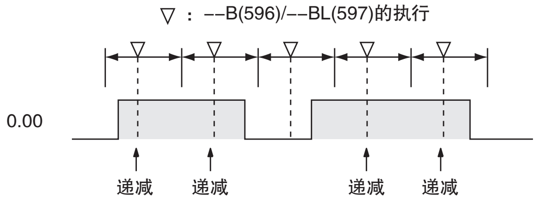


程序举例

● --B(596)/--BL(597) 的操作

下例中，只要 CIO 0.00 为 ON，每次循环时 D100 的 BCD 内容均会递减 1。

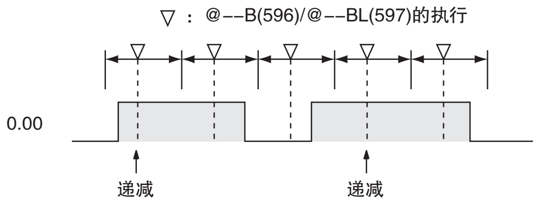
下例中，只要 CIO 0.00 为 ON，每次循环时 D101 和 D100 的 8 位数 BCD 内容均会递减 1。



● @--B(596)/@--BL(597) 的操作

下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D100 的 BCD 内容将递减 1。

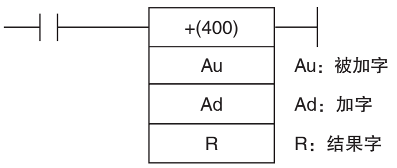
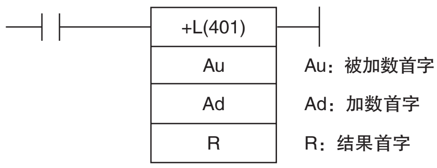
下例中使用了上升沿微分变化，因此当 CIO 0.00 从 OFF 变 ON 时，D101 和 D100 的 BCD 内容将递减 1。



四则运算指令

+/+L

指令	助记符	变化	功能代码	功能
无进位带符号二进制加	+	@+	400	4 位数 (单字) 十六进制数据和 / 或常数相加。
无进位带符号双字二进制加	+L	@+L	401	8 位数 (双字) 十六进制数据和 / 或常数相加。

	+	+L
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		+	+L	+	+L
Au	+: 被加字 +L：被加数首字	INT	DINT	1	2
Ad	+: 加字 +L：加数首字	INT	DINT	1	2
R	+: 结果字 +L：结果首字	INT	DINT	1	2

● 区域类型

<table>
<tr><th colspan="2" rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">+</td><td>Au，Ad</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td></tr>
<tr><td rowspan="2">+L</td><td>Au，Ad</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

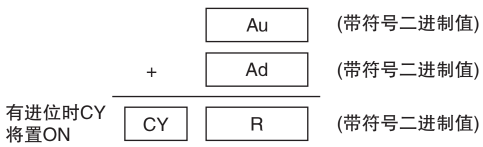
条件标志

名称	标记	操作	
		+	+L
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相加导致有进位时 ON。 · 其它情况下 OFF。	· 相加导致有进位时 ON。 · 其它情况下 OFF。
上溢标志	P_OF	· 两个正数的相加结果在 8000 ~ FFFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 两个正数的相加结果在 80000000 ~ FFFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
下溢标志	P_UF	· 两个负数的相加结果在 0000 ~ 7FFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 两个负数的相加结果在 00000000 ~ 7FFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

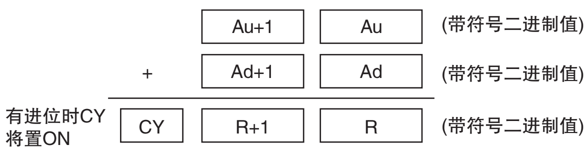
● +

+(400) 指令将 Au 和 Ad 中的二进制值相加，并将结果输出到 R。

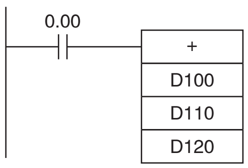


● +L

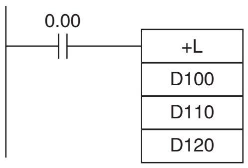
+L(401) 指令将 Au 和 Au+1 以及 Ad 和 Ad+1 中的二进制值相加，并将结果输出到 R 和 R+1 中。



程序举例



该例中，当CIO 0.00为ON时，将D100和D110作为4位带符号二进制值进行相加并将结果输出到D120中。



当CIO 0.00为ON时，将D101、D100和D111、D110作为8位带符号二进制值进行相加，并将结果输出到D121和D120中。

+C/+CL

指令	助记符	变化	功能代码	功能
有进位带符号二进制加	+C	@+C	402	4 位数 (单字) 十六进制数据和 / 或常数及进位标志 (CY) 相加。
有进位带符号双字二进制加	+CL	@+CL	403	8 位数 (双字) 十六进制数据和 / 或常数及进位标志 (CY) 相加。

符号	+C	+CL
	+C(402) Au　Au：被加字 Ad　Ad：加字 R　R：结果字	+CL(403) Au　Au：被加数首字 Ad　Ad：加数首字 R　R：结果首字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		+C	+CL	+C	+CL
Au	+C：　被加字 +CL：被加数首字	INT	DINT	1	2
Ad	+C：　加字 +CL：加数首字	INT	DINT	1	2
R	+C：　结果字 +CL：结果首字	INT	DINT	1	2

● 区域类型

<table>
<tr><th colspan="2" rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">+C</td><td>Au，Ad</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td></tr>
<tr><td rowspan="2">+CL</td><td>Au，Ad</td><td>OK</td><td rowspan="2">---</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

条件标志

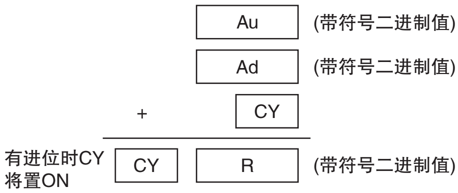
名称	标记	操作	
		+C	+CL
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 相加结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相加导致有进位时 ON。 · 其它情况下 OFF。	· 结果有进位时 ON。 · 其它情况下 OFF。

名称	标记	操作	
		+C	+CL
上溢标志	P_OF	・两个正数和 CY 的相加结果在 8000 ~ FFFF Hex 范围内时 ON。 ・其它情况下 OFF。	・两个正数和 CY 的相加结果在 80000000 ~ FFFFFFFF Hex 范围内时 ON。 ・其它情况下 OFF。
下溢标志	P_UF	・两个负数和 CY 的相加结果在 0000 ~ 7FFF Hex 范围内时 ON。 ・其它情况下 OFF。	・两个负数和 CY 的相加结果在 00000000 ~ 7FFFFFFF Hex 范围内时 ON。 ・其它情况下 OFF。
负标志	P_N	・结果的最左位为 1 时 ON。 ・其它情况下 OFF。	・结果的最左位为 1 时 ON。 ・其它情况下 OFF。

功能

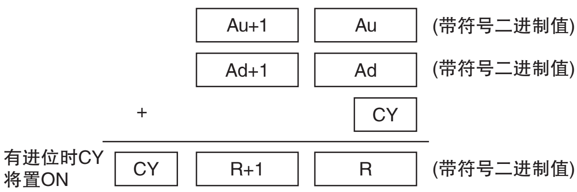
● +C

+C(402) 指令将 Au、Ad 和 CY 中的二进制值相加，并将结果输出到 R。



● +CL

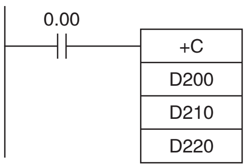
+CL(403) 指令将 Au 和 Au+1、Ad 和 Ad+1 以及 CY 中的二进制值相加，并将结果输出到 R 和 R+1。



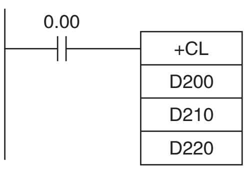
提示

・若要清除进位标志 (CY)，可执行清除进位 (CLC(041)) 指令。

程序举例



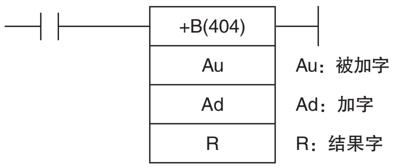
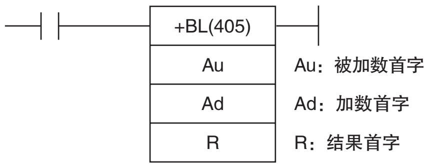
当CIO 0.00为ON时，将D200、D210和CY作为4位带符号二进制值进行相加，并将结果输出到D220中。



当CIO 0.00为ON时，将D201、D200、D211、D210和CY作为8位带符号二进制值进行相加，并将结果输出到D221和D220中。

+B/+BL

指令	助记符	变化	功能代码	功能
无进位 BCD 加	+B	@+B	404	4 位数 (单字)BCD 数据和 / 或常数相加。
无进位双字 BCD 加	+BL	@+BL	405	8 位数 (双字)BCD 数据和 / 或常数相加。

	+B	+BL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		+B	+BL	+B	+BL
Au	+B：　被加字 +BL：被加数首字	WORD	DWORD	1	2
Ad	+B：　加字 +BL：加数首字	WORD	DWORD	1	2
R	+B：　结果字 +BL：结果首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
+B	Au，Ad	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
+BL	Au，Ad	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

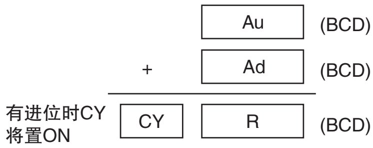
条件标志

名称	标记	操作	
		+B	+BL
出错标志	P_ER	· Au 不是 BCD 时 ON。 · Ad 不是 BCD 时 ON。 · 其它情况下 OFF。	· Au，Au+1 不是 BCD 时 ON。 · Ad，Ad+1 不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相加导致有进位时 ON。 · 其它情况下 OFF。	· 相加导致有进位时 ON。 · 其它情况下 OFF。

功能

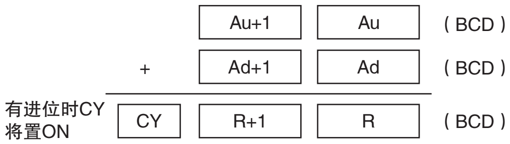
● +B

+B(404) 指令将 Au 和 Ad 中的 BCD 值相加，并将结果输出到 R。

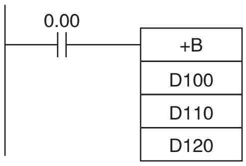


● +BL

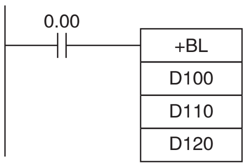
+BL(405) 指令将 Au 和 Au+1 以及 Ad 和 Ad+1 中的 BCD 值相加，并将结果输出到 R 和 R+1。



程序举例



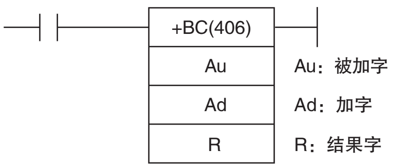
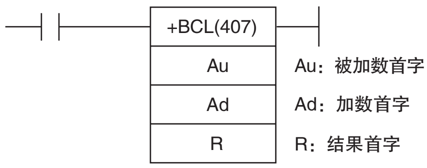
该例中，当CIO 0.00为ON时，将D100和D110作为4位BCD值进行相加，并将结果输出到D120中。



该例中，当CIO 0.00为ON时，将D101和D100以及D111和D110作为8位BCD值相加，并将结果输出到D121和D120中。

+BC/+BCL

指令	助记符	变化	功能代码	功能
有进位 BCD 加	+BC	@+BC	406	4 位数 (单字)BCD 数据和 / 或常数及进位标志 (CY) 相加。
有进位双字 BCD 加	+BCL	@+BCL	407	8 位数 (双字)BCD 数据和 / 或常数及进位标志 (CY) 相加。

	+BC	+BCL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		+BC	+BCL	+BC	+BCL
Au	+BC：　被加字 +BCL：被加数首字	WORD	DWORD	1	2
Ad	+BC：　加字 +BCL：加数首字	WORD	DWORD	1	2
R	+BC：　结果字 +BCL：结果首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
+BC	Au，Ad	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
+BCL	Au，Ad	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

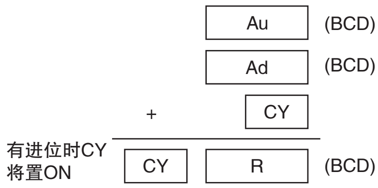
条件标志

名称	标记	操作	
		+BC	+BCL
出错标志	P_ER	· Au 不是 BCD 时 ON。 · Ad 不是 BCD 时 ON。 · 其它情况下 OFF。	· Au，Au+1 不是 BCD 时 ON。 · Ad，Ad+1 不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相加导致有进位时 ON。 · 其它情况下 OFF。	· 相加导致有进位时 ON。 · 其它情况下 OFF。

功能

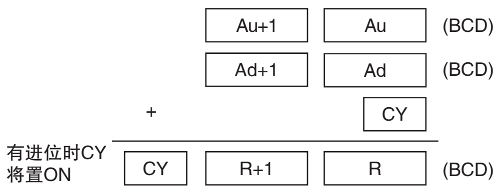
● +BC

+BC(406) 指令将 Au、Ad 和 CY 中的 BCD 值相加，并将结果输出到 R。



● +BCL

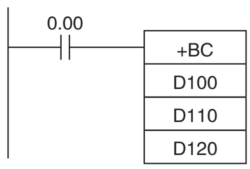
+BCL(407) 指令将 Au 和 Au+1、Ad 和 Ad+1 以及 CY 中的 BCD 值相加，并将结果输出到 R 和 R+1。



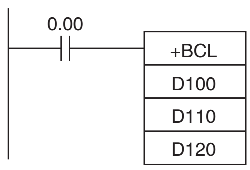
提示

・若要清除进位标志 (CY)，可执行清除进位 (CLC(041)) 指令。

程序举例



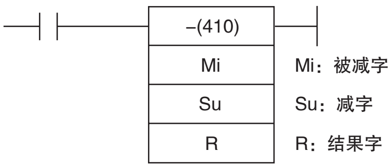
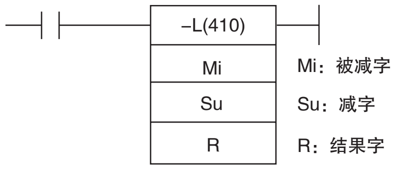
该例中，当CIO 0.00为ON时，将D101、D110、和CY作为4位BCD值进行相加，并将结果输出到D120中。



该例中，当CIO 0.00为ON时，将D101、D100、D111、D110和CY作为8位BCD值相加，并将结果输出到D121和D120中。

–/–L

指令	助记符	变化	功能代码	功能
无进位带符号二进制减	–	@–	410	4 位数 (单字) 十六进制数据和 / 或常数相减。
无进位带符号双字二进制减	–L	@–L	411	8 位数 (双字) 十六进制数据和 / 或常数相减。

<table>
<tr><th rowspan="2">符号</th><th>–</th><th>–L</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>–</th><th>–L</th><th>–</th><th>–L</th></tr>
<tr><td>Mi</td><td>–：被减字
–L：被减数首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
<tr><td>Su</td><td>–：减字
–L：减字首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
<tr><td>R</td><td>–：结果字
–L：结果首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th colspan="2" rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR
直接</th><th>IR
间接</th></tr>
<tr><td rowspan="2">–</td><td>Mi，Su</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td></tr>
<tr><td rowspan="2">–L</td><td>Mi，Su</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

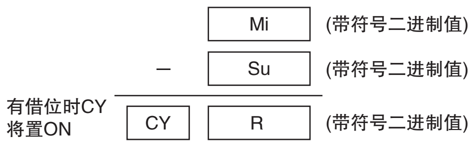
条件标志

名称	标记	操作	
		–	–L
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相减导致有借位时 ON。 · 其它情况下 OFF。	· 相减导致有借位时 ON。 · 其它情况下 OFF。
上溢标志	P_OF	· 从一个正数减去一个负数的结果在 8000 ~ FFFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 从一个正数减去一个负数的结果在 80000000 ~ FFFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
下溢标志	P_UF	· 从一个正数减去一个负数的结果在 0000 ~ 7FFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 从一个负数减去一个正数的结果在 00000000 ~ 7FFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

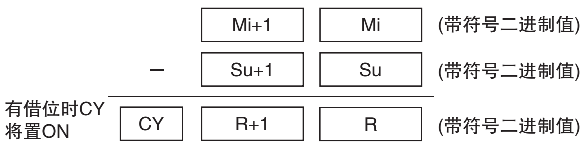
● –

–(400) 指令从 Mi 中减去 Su 中的二进制值，并将结果输出到 R。当结果为负时，将 2 的补码输出到 R。



● –L

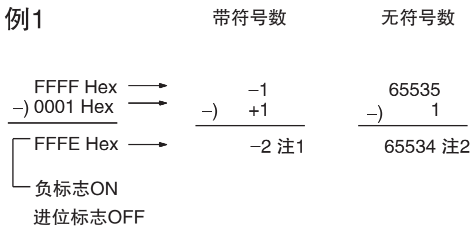
–L(411) 指令从 Mi 和 Mi+1 中减去 Su 和 Su+1 中的二进制值，并将结果输出到 R 和 R+1。当结果为负时，将 2 的补码输出到 R 和 R+1。



提示

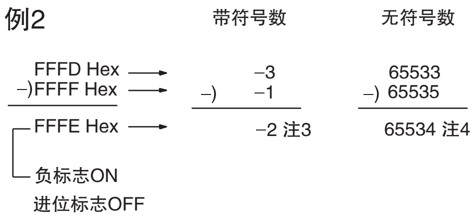
· 2 的补码

2 的补码是用 1 减去每个二进制数位再将结果加 1 得到的数。例如，1101 的 2 的补码计算如下：1111(十六进制的 F)–1101(十六进制的 D)+1(十六进制的 1)= 0011(十六进制的 3)。3039(十六进制) 的 2 的补码计算如下：FFFF(十六进制)–3039(十六进制)+ 0001(十六进制)–CFC7(十六进制)。因此，对于 4 位数十六进制值，2 的补码可计算如下：FFFF(十六进制)–a(十六进制)+0001(十六进制)= b(十六进制)。若要从 2 的补码 b(十六进制) 得到真值：a(十六进制)=10000(十六进制)–b(十六进制)。例如，若要从2的补码CFC7(十六进制)得到真值：10000(十六进制)–CFC7=3039。



注 1. 由于负标志为ON，因此结果(FFFE Hex)为负值(2的补码)，即–2。

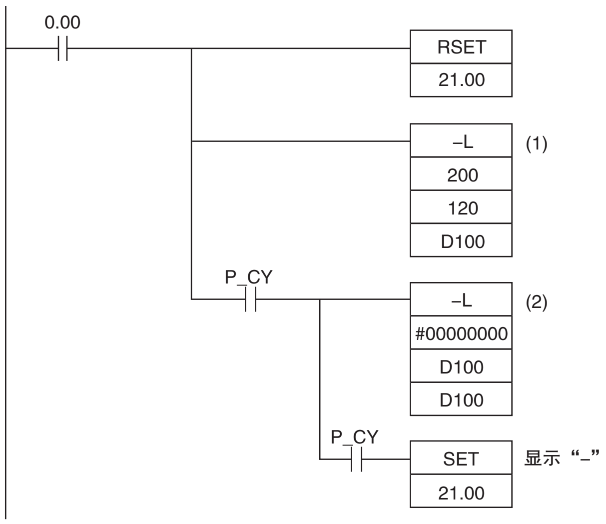
2. 由于进位标志为OFF，因此结果(FFFE Hex)为无符号正值，即65534。



3. 由于负标志为ON，因此结果(FFFE Hex)为负值(2的补码)，即–2。

4. 由于进位标志为ON，因此结果(FFFE Hex)为负值(2的补码)，转换成真值时为–2。

20F55A10—B8A360E3=–97AE06D3。(十六进制)



该例中，将 CIO 201 和 CIO 200 中的 8 位数二进制值减去 CIO 121 和 CIO 120 中的 8 位数二进制值，并将结果以 8 位数二进制值的形式输出到 D101 和 D100。如果结果为负，将执行 (2) 处的指令，并将实际结果输出到 D101 和 D100。

(1) 处的减法

	Mi+1：CIO 201	Mi：CIO 200
	2 0 F 5	5 A 1 0
	Su+1：CIO 121	Su：CIO 120
–	B 8 A 3	6 0 E 3
CY	R+1：D101	R+1：D100
1	6 8 5 1	F 9 2 D

进位标志 (CY) 为 ON，因此用 0000 0000 减去结果从而得到真值。

(2) 处的减法

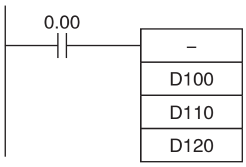
	0 0 0 0	0 0 0 0
	Su+1：D101	Su：D100
–	6 8 5 1	F 9 2 D
CY	R+1：D101	R+1：D100
1	9 7 A E	0 6 D 3

最终相减结果

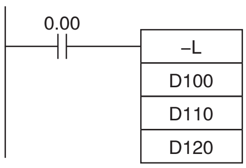
	Mi+1：CIO 201	Mi：CIO 200
	2 0 F 5	5 A 1 0
	Su+1：D101	Su：D100
–	6 8 5 1	F 9 2 D
CY	R+1：D101	R+1：D100
1	9 7 A E	0 6 D 3

进位标志 (CY) 被置 ON，因此真值为 –97AE06D3。由于 D101 和 D100 的内容为负，因此用 CY 来对 CIO 21.00 置 ON 以表示负值。

程序举例



该例中，当CIO 0.00为ON时，将D100作为4位数带符号二进制值减去D110，并将结果输出到D120中。



该例中，当CIO 0.00为ON时，将D101和D100作为8位带符号二进制值减去D111和D110，并将结果输出到D121和D120中。

如果相减的结果为负数 (Mi<Su 或 Mi+1，Mi<Su+1，Su)，则将结果以 2 的补码输出，并对进位标志 (CY) 置 ON，以表示相减的结果为负数。若要将 2 的补码转换成真值，必须将进位标志 (CY) 用作执行条件，用 0 减去结果。

–C/–CL

指令	助记符	变化	功能代码	功能
有进位带符号二进制减	–C	@–C	412	4 位数 (单字) 十六进制数据和 / 或常数及进位标志 (CY) 相减。
有进位带符号双字二进制减	–CL	@–CL	413	8 位数 (双字) 十六进制数据和 / 或常数及进位标志 (CY) 相减。

<table>
<tr><th rowspan="2">符号</th><th>–C</th><th>–CL</th></tr>
<tr><td>–C(412)
Mi　Mi：被减字
Su　Su：减字
R　R：结果字</td><td>–CL(413)
Mi　Mi：被减字
Su　Su：减字
R　R：结果字</td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>–C</th><th>–CL</th><th>–C</th><th>–CL</th></tr>
<tr><td>Mi</td><td>–C：　被减字
–CL：被减数首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
<tr><td>Su</td><td>–C：　减字
–CL：减数首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
<tr><td>R</td><td>–C：　结果字
–CL：结果首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th rowspan="2" colspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">–C</td><td>Mi，Su</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td></tr>
<tr><td rowspan="2">–CL</td><td>Mi，Su</td><td>OK</td><td rowspan="2">---</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

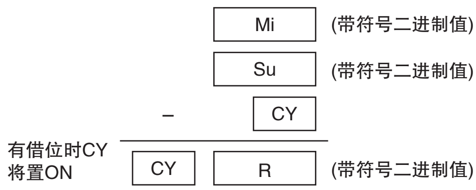
条件标志

名称	标记	操作	
		–C	–CL
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 相减的结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相减导致有借位时 ON。 · 其它情况下 OFF。	· 结果有借位时 ON。 · 其它情况下 OFF。
上溢标志	P_OF	· 从一个正数减去一个负数和 CY 的结果在 8000 ~ FFFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 从一个正数减去一个负数和 CY 的结果在 80000000 ~ FFFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
下溢标志	P_UF	· 从一个负数减去一个正数和 CY 的结果在 0000 ~ 7FFF Hex 范围内时 ON。 · 其它情况下 OFF。	· 从一个负数减去一个正数和 CY 的结果在 00000000 ~ 7FFFFFFF Hex 范围内时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

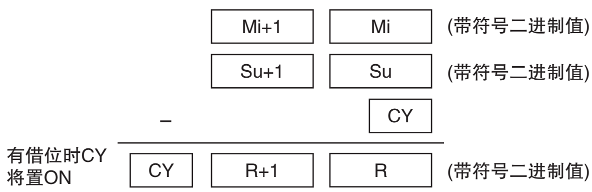
● –C

–C(412) 指令从 Mi 中减去 Su 中的二进制值和 CY，并将结果输出到 R。当结果为负时，将 2 的补码输出到 R。



● –CL

–CL(413) 指令从 Mi 和 Mi+1 中减去 Su 和 Su+1 中的二进制值和 CY，并将结果输出到 R 和 R+1。当结果为负时，将 2 的补码输出到 R 和 R+1。



提示

· 若要清除进位标志 (CY)，可执行清除进位 (CLC(041)) 指令。

· 2 的补码

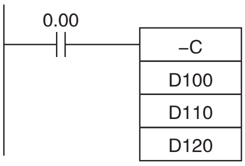
2 的补码是用 1 减去每个二进制数位再将结果加 1 得到的数。

例如： 二进制数 1101 的 2 的补码计算如下：
1111(十六进制的 F)–1101(十六进制的 D)+1(十六进制的 1)=0011(十六进制的 3)。

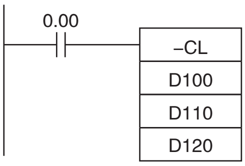
例如： 十六进制数 3039 的 2 的补码计算如下：
FFFF Hex–3039 Hex+0001 Hex=CFC7 Hex。
据此，4 位数的十六进制值 “a” 的 2 的补码计算如下：
FFFF Hex–a Hex+0001 Hex=b Hex。
若要从 2 的补码 “b” (十六进制) 得到真值 “a” (十六进制)：
a Hex+10000 Hex–b Hex。

例如： 若要从 2 的补码 CFC&(十六进制) 得到真值：
10000 Hex–CFC7 Hex=3039 Hex。

程序举例



该例中，当CIO 0.00为ON时，将D100作为4位数带符号二进制值减去D110和CY，并将结果输出到D120中。

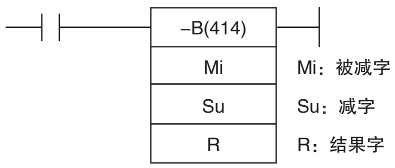
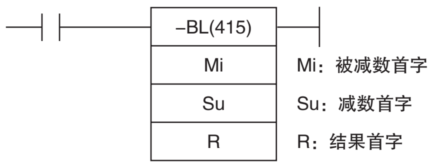


该例中，当CIO 0.00为ON时，将D101和D100作为8位数带符号二进制值减去D111和D110以及CY，并将结果输出到D121和D120中。

如果相减的结果为负数 (Mi<Su 或 Mi+1，Mi<Su+1，Su)，则将结果以 2 的补码输出。进位标志 (CY) 将变 ON。若要将 2 的补码转换成真值，必须将进位标志 (CY) 用作输入条件，用 0 减去结果。进位标志变 ON，用于表示相减的结果为负。

–B/–BL

指令	助记符	变化	功能代码	功能
无进位 BCD 减	–B	@–B	414	4 位数 (单字)BCD 数据和 / 或常数相减。
无进位双字 BCD 减	–BL	@–BL	415	8 位数 (双字)BCD 数据和 / 或常数相减。

	–B	–BL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		–B	–BL	–B	–BL
Mi	–B： 被减字 –BL：被减数首字	WORD	DWORD	1	2
Su	–B： 减字 –BL：减数首字	WORD	DWORD	1	2
R	–B： 结果字 –BL：结果首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
–B	Mi，Su	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
–BL	Mi，Su	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

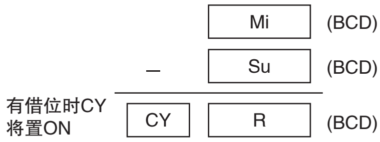
条件标志

名称	标记	操作	
		–B	–BL
出错标志	P_ER	· Mi 不是 BCD 时 ON。 · Su 不是 BCD 时 ON。 · 其它情况下 OFF。	· Mi 和 / 或 Mi+1 不是 BCD 时 ON。 · Su 和 / 或 Su+1 不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相减导致有借位时 ON。 · 其它情况下 OFF。	· 相减导致有借位时 ON。 · 其它情况下 OFF。

功能

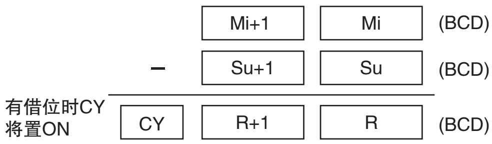
● –B

–B(414) 指令从 Mi 中减去 Su 中的 BCD 值，并将结果输出到 R。当相减的结果为负时，将结果以 10 的补码输出到 R。



● –BL

–BL(415) 指令从 Mi 和 Mi+1 中减去 Su 和 Su+1 中的 BCD 值，并将结果输出到 R 和 R+1。当结果为负时，将 10 的补码输出到 R 和 R+1。

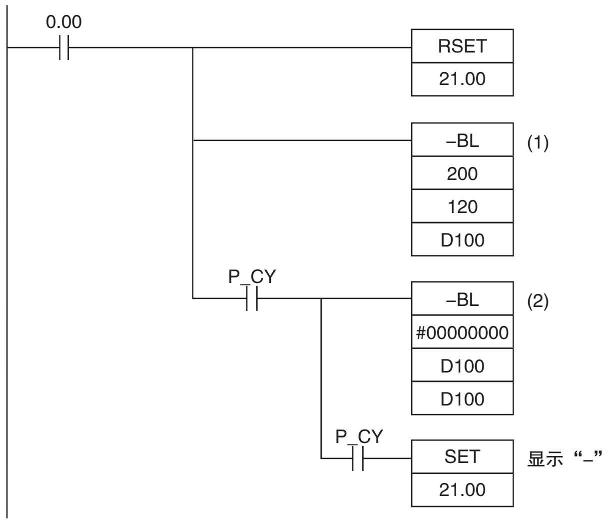


提示

・10 的补码

10 的补码是用 9 减去每个数位再将结果加 1 得到的数。例如，7556 的 10 的补码计算如下：9999–7556+1=2444。对于 4 位数 A，其 10 的补码为 9999–A+1=B。若要从 10 的补码 B 得到真值：A=10000–B。例如，若要从 10 的补码 2444 得到真值：10000–2444=7556。

例：9,583,960–17,072,641 = –7,488,681(BCD)



该例中，将 CIO 201 和 CIO 200 中的 8 位数 BCD 内容减去 CIO 121 和 CIO 120 中的 8 位数 BCD 内容，并将结果以 8 位数 BCD 的形式输出到 D101 和 D100。结果为负，因此将执行 (2) 处的指令，并将真值输出到 D101 和 D100。

(1) 处的减法

Mi+1：CIO 201　S1：CIO 200
| 0 | 9 | 5 | 8 | | 3 | 9 | 6 | 0 |

Su+1：CIO 121　S2：CIO 120
– | 1 | 7 | 0 | 7 | | 2 | 6 | 4 | 1 |

09583960+(100000000–17072641)

CY　R+1：D101　R+1：D100
| 1 | | 9 | 2 | 5 | 1 | | 1 | 3 | 1 | 9 |

进位标志 (CY) 为 ON，因此用 0000 0000 减去结果。

(2) 处的减法

| 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

Su+1：D101　Su：D100
– | 9 | 2 | 5 | 1 | | 1 | 3 | 1 | 9 |

00000000+(100000000–92511319)

CY　R+1：D101　R+1：D100
| 1 | | 0 | 7 | 4 | 8 | | 8 | 6 | 8 | 1 |

最终相减结果

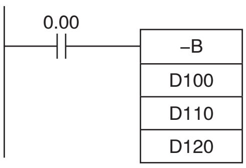
Mi+1：CIO 201　Mi：CIO 200
| 0 | 9 | 5 | 8 | | 3 | 9 | 6 | 0 |

Su+1：D101　Su：D100
– | 1 | 7 | 0 | 7 | | 2 | 6 | 4 | 1 |

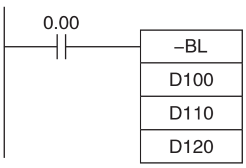
CY　R+1：D101　R+1：D100
| 1 | | 0 | 7 | 4 | 8 | | 8 | 6 | 8 | 1 |

进位标志 (CY) 被置 ON，因此真值为 –7,488,681。由于 D101 和 D100 的内容为负，因此用 CY 来对 CIO 21.00 置 ON 以表示负值。

程序举例



该例中，当CIO 0.00为ON时，将D100作为4位数BCD值减去D110，并将结果输出到D120中。



该例中，当CIO 0.00为ON时，将D101和D100作为8位数BCD值减去D111和D110，并将结果输出到D121和D120中。

如果相减的结果为负数 (Mi<Su 或 Mi+1，Mi<Su+1，Su)，则将结果以 10 的补码输出。进位标志 (CY) 将变 ON。若要将 10 的补码转换成真值，必须将进位标志 (CY) 用作输入条件，用 0 减去结果。进位标志变 ON，用于表示相减的结果为负。

–BC/–BCL

指令	助记符	变化	功能代码	功能
有进位 BCD 减	–BC	@–BC	416	4 位数 (单字)BCD 数据和 / 或常数及进位标志 (CY) 相减。
有进位双字 BCD 减	–BCL	@–BCL	417	8 位数 (双字)BCD 数据和 / 或常数及进位标志 (CY) 相减。

	–BC		–BCL	
符号	–BC(416)		–BCL(417)	
	Mi	Mi：被减字	Mi	Mi：被减数首字
	Su	Su：减字	Su	Su：减数首字
	R	R：结果字	R	R：结果首字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		–BC	–BCL	–BC	–BCL
Mi	–BC： 被减字 –BCL：被减数首字	WORD	DWORD	1	2
Su	–BC： 减字 –BCL：减数首字	WORD	DWORD	1	2
R	–BC： 结果字 –BCL：结果首字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
–BC	Mi，Su	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
–BCL	Mi，Su	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

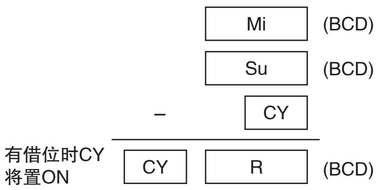
条件标志

名称	标记	操作	
		–BC	–BCL
出错标志	P_ER	· Mi 不是 BCD 时 ON。 · Su 不是 BCD 时 ON。 · 其它情况下 OFF。	· Mi 和 / 或 Mi+1 不是 BCD 时 ON。 · Su 和 / 或 Su+1 不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 相减导致有借位时 ON。 · 其它情况下 OFF。	· 相减导致有借位时 ON。 · 其它情况下 OFF。

功能

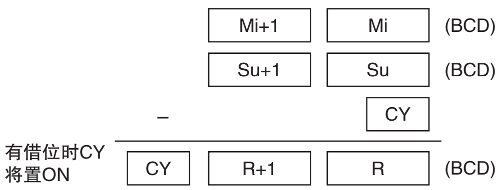
● –BC

–BC(416) 指令从 Mi 中减去 Su 中的 BCD 值和 CY，并将结果输出到 R。若结果为负，则将 10 的补码输出到 R。



● –BCL

–BCL(417) 指令从 Mi 和 Mi+1 中减去 Su 和 Su+1 中的 BCD 值以及 CY，并将结果输出到 R 和 R+1。当结果为负时，将 10 的补码输出到 R 和 R+1。



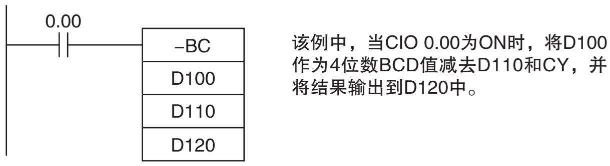
提示

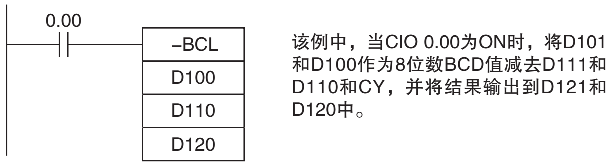
· 若要清除进位标志 (CY)，可执行清除进位 (CLC(041)) 指令。

· **10 的补码**

10 的补码是用 9 减去每个数位再将结果加 1 得到的数。例如，7556 的 10 的补码计算如下：9999–7556+1=2444。对于 4 位数 A，其 10 的补码为 9999–A+1=B。若要从 10 的补码 B 得到真值：A=10000–B。例如，若要从 10 的补码 2444 得到真值：10000–2444=7556。

程序举例

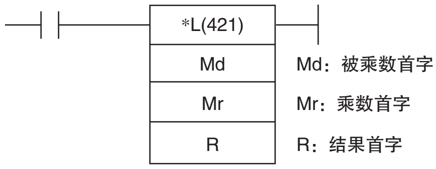




如果相减的结果为负数 (Mi<Su 或 Mi+1，Mi<Su+1，Su)，则将结果以 10 的补码输出。进位标志 (CY) 将变 ON。若要将 10 的补码转换成真值，必须将进位标志 (CY) 用作输入条件，用 0 减去结果。进位标志变 ON，用于表示相减的结果为负。

*/*L

指令	助记符	变化	功能代码	功能
带符号二进制乘	*	@*	420	4 位数带符号十六进制数据和 / 或常数相乘。
带符号双字二进制乘	*L	@*L	421	8 位数带符号十六进制数据和 / 或常数相乘。

	*	*L
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		*	*L	*	*L
Md	*：被乘字 *L：被乘数首字	INT	DINT	1	2
Mr	*：乘字 *L：乘数首字	INT	DINT	1	2
R	结果首字	DINT	LINT	2	4

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
*	Md，Mr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
*L	Md，Mr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

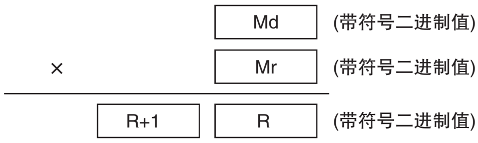
条件标志

名称	标记	操作	
		*	*L
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

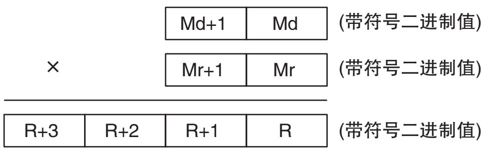
● *

*(420) 指令将 Md 和 Mr 中的带符号二进制值相乘，并将结果输出到 R 和 R+1。

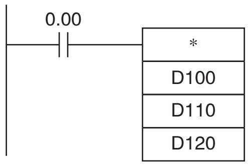


● *L

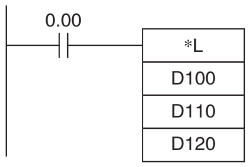
*L(421) 指令将 Md 和 Md+1 以及 Mr 和 Mr+1 中的带符号二进制值相乘，并将结果输出到 R、R+1、R+2 和 R+3。



程序举例



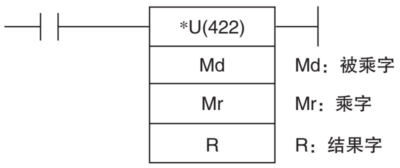
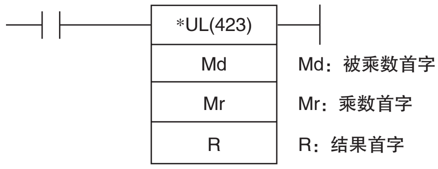
该例中，当CIO 0.00为ON时，将D100和D110作为4位带符号十六进制值相乘，并将结果输出到D120和D121中。



该例中，当CIO 0.00为ON时，将D101、D100、D111和D110作为8位带符号十六进制值相乘，并将结果输出到D123、D122、D121和D120中。

*U/*UL

指令	助记符	变化	功能代码	功能
无符号二进制乘	*U	@*U	422	4 位数无符号十六进制数据和 / 或常数相乘。
无符号双字二进制乘	*UL	@*UL	423	8 位数无符号十六进制数据和 / 或常数相乘。

	*U	*UL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		*U	*UL	*U	*UL
Md	*U:　被乘字 *UL：被乘数首字	UINT	UDINT	1	2
Mr	*U:　乘字 *UL：乘数首字	UINT	UDINT	1	2
R	结果首字	UDINT	ULINT	2	4

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
*U	Md，Mr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R										---	---					
*UL	Md，Mr										OK						
	R										---						

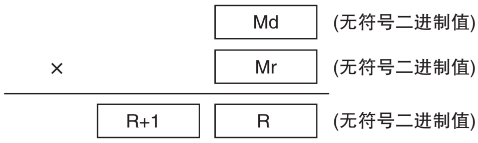
条件标志

名称	标记	操作	
		*U	*UL
出错标志	P_ER	OFF	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

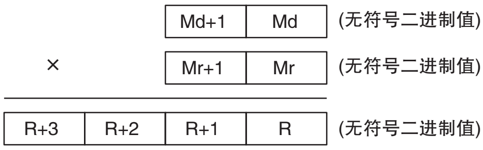
● *U

*U(422) 指令将 Md 和 Mr 中的无符号二进制值相乘，并将结果输出到 R 和 R+1。



● *UL

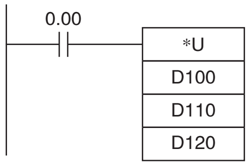
*UL(423) 指令将 Md 和 Md+1 以及 Mr 和 Mr+1 中的无符号二进制值相乘，并将结果输出到 R、R+1、R+2 和 R+3。



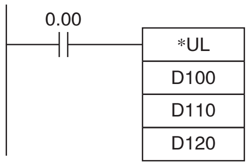
注意事项

可在 CP2E CPU 单元中使用 *U/*UL 指令。不能在 CP1E CPU 单元中使用。

程序举例



该例中，当CIO 0.00为ON时，将D100和D110作为4位无符号十六进制值相乘，并将结果输出到D120和D121中。

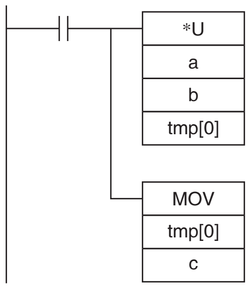


该例中，当CIO 0.00为ON时，将D101、D100、D111和D110作为8位无符号十六进制值相乘，并将结果输出到D123、D122、D121和D120中。

● 功能块定义内的使用示例

使用数组变量仅提取 1 个通道的乘法运算结果。

a * b → c



使用变量：
非乘法运算数据a(数据类型UNIT)
乘法运算数据b(数据类型UNIT)
结果c(数据类型UNIT)
临时变量tmp(数据类型WORD，数组2元素)

*B/*BL

指令	助记符	变化	功能代码	功能
BCD 乘	*B	@*B	424	4 位数 (单字)BCD 数据和 / 或常数相乘。
双字 BCD 乘	*BL	@*BL	425	8 位数 (双字)BCD 数据和 / 或常数相乘。

	*B	*BL
符号	*B(424) Md　Md：被乘字 Mr　Mr：乘字 R　R：结果字	*BL(425) Md　Md：被乘数首字 Mr　Mr：乘数首字 R　R：结果首字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		*B	*BL	*B	*BL
Md	*B：　被乘字 *BL：被乘数首字	WORD	DWORD	1	2
Mr	*B：　乘字 *BL：乘数首字	WORD	DWORD	1	2
R	结果首字	DWORD	LWORD	2	4

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
*B	Md，Mr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
*BL	Md，Mr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

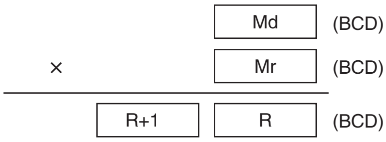
条件标志

名称	标记	操作	
		*B	*BL
出错标志	P_ER	· Md 不是 BCD 时 ON。 · Mr 不是 BCD 时 ON。 · 其它情况下 OFF。	· Md 和 / 或 Md+1 不是 BCD 时 ON。 · Mr 和 / 或 Mr+1 不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。

功能

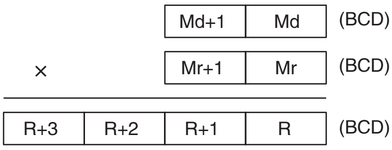
● *B

*B(424) 指令将 Md 和 Mr 中的 BCD 内容相乘，并将结果输出到 R 和 R+1。

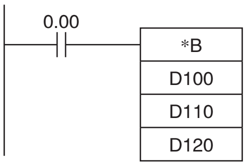


● *BL

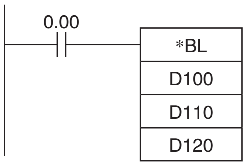
*BL(425) 指令将 Md 和 Md+1 以及 Mr 和 Mr+1 中的 BCD 值相乘，并将结果输出到 R、R+1、R+2 和 R+3。



程序举例



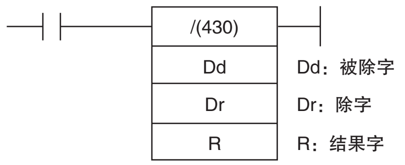
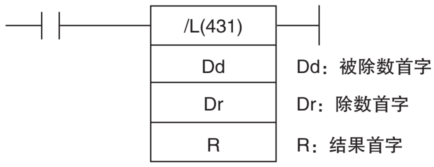
该例中，当CIO 0.00为ON时，将D100、和D110作为4位BCD值进行相乘，并将结果输出到D121和D120中。



该例中，当CIO 0.00为ON时，将D101、D100、D111和D110作为8位无符号BCD值相乘，并将结果输出到D123、D122、D121和D120中。

/，/L

指令	助记符	变化	功能代码	功能
带符号二进制除	/	@/	430	4 位数 (单字) 带符号十六进制数据和 / 或常数相除。
带符号双字二进制除	/L	@/L	431	8 位数 (双字) 带符号十六进制数据和 / 或常数相除。

符号	/	/L
		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		/	/L	/	/L
Dd	/: 被除字 /L：被除数首字	INT	DINT	1	2
Dr	/: 除字 /L：除数首字	INT	DINT	1	2
R	结果首字	DWORD	LWORD	2	4

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
/	Dd，Dr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
/L	Dd，Dr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

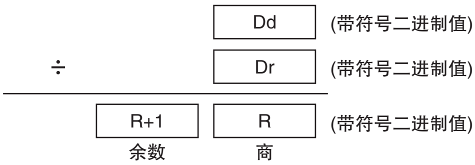
条件标志

名称	标记	操作
出错标志	P_ER	· 除数为 0 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 相除的结果 R/R+1 和 R 为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R/R+1 和 R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● /

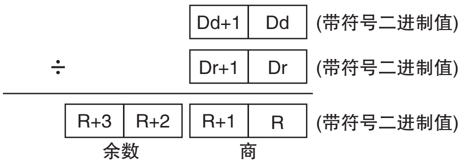
/(430) 指令将 Dd 中的带符号二进制 (16 位) 值除以 Dr 中的带符号二进制值，并将结果输出到 R 和 R+1。商放在 R 中，余数放在 R+1 中。



注 十六进制数 #8000 除以 #FFFF 的除法未定义。

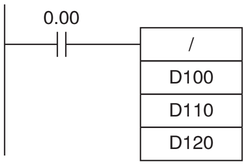
● /L

/L(431) 指令将 Dd 和 Dd+1 中的带符号二进制值除以 Dr 和 Dr+1 中的带符号二进制值，并将结果输出到 R、R+1、R+2 和 R+3。商输出到 R 和 R+1，余数输出到 R+2 和 R+3。

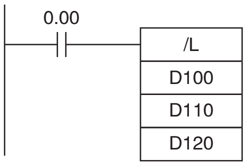


注 十六进制数 #80000000 除以 #FFFFFFFF 的除法未定义。

程序举例



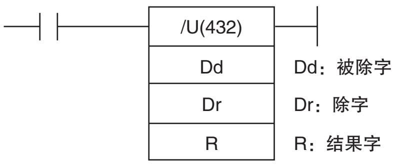
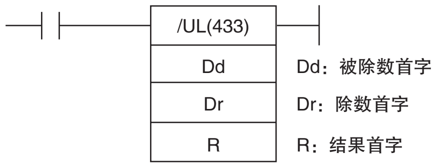
该例中，当CIO 0.00为ON时，将D110作为4位数带符号二进制值除以D100，并将商输出到D120，将余数输出到D121中。



该例中，当CIO 0.00为ON时，将D111和D110作为8位数带符号十六进制值除以D101和D100，并将商输出到D121和D120中，并将余数输出到D123和D122中。

/U，/UL

指令	助记符	变化	功能代码	功能
无符号二进制除	/U	@/U	432	4 位数 (单字) 无符号十六进制数据和 / 或常数相除。
无符号双字二进制除	/UL	@/UL	433	8 位数 (双字) 无符号十六进制数据和 / 或常数相除。

	/U	/UL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>/U</th><th>/UL</th><th>/U</th><th>/UL</th></tr>
<tr><td>Dd</td><td>/U:　被除字
/UL：被除数首字</td><td>UINT</td><td>UDINT</td><td>1</td><td>2</td></tr>
<tr><td>Dr</td><td>/U:　除字
/UL：除数首字</td><td>UINT</td><td>UDINT</td><td>1</td><td>2</td></tr>
<tr><td>R</td><td>结果首字</td><td>DWORD</td><td>LWORD</td><td>2</td><td>4</td></tr>
</table>

● 区域类型

<table>
<tr><th rowspan="2" colspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">/U</td><td>Dd，Dr</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td>OK</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td><td rowspan="3">---</td></tr>
<tr><td rowspan="2">/UL</td><td>Dd，Dr</td><td>OK</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

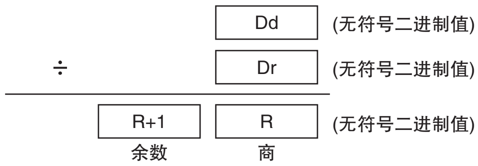
条件标志

名称	标记	操作
出错标志	P_ER	· 除数为 0 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 相除的结果 R/R+1 和 R 为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R/R+1 和 R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

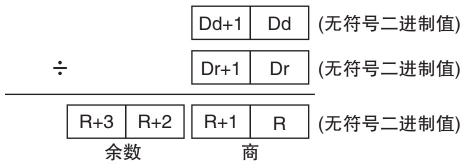
● /U

/U(432) 指令将 Dd 中的无符号二进制 (16 位) 值除以 Dr 中的无符号二进制值，并将结果输出到 R 和 R+1。商放在 R 中，余数放在 R+1 中。



● /UL

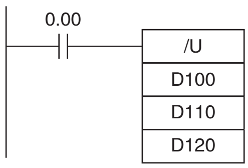
/UL(433) 指令将 Dd 和 Dd+1 中的无符号二进制值除以 Dr 和 Dr+1 中的无符号二进制值，并将结果输出到 R、R+1、R+2 和 R+3。商输出到 R 和 R+1，余数输出到 R+2 和 R+3。



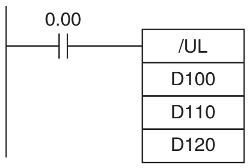
注意事项

可在 CP2E CPU 单元中使用 /U、/UL 指令。不能在 CP1E CPU 单元中使用。

程序举例



该例中，当CIO 0.00为ON时，将D110作为4位数无符号二进制值除以D100，并将商输出到D120，将余数输出到D121中。

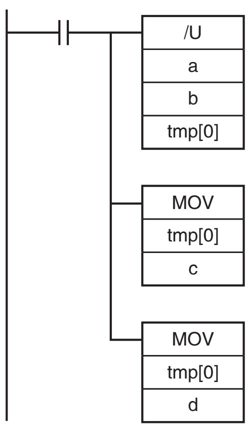


该例中，当CIO 0.00为ON时，将D111和D110作为8位数无符号十六进制值除以D101和D100，并将商输出到D121和D120中，并将余数输出到D123和D122中。

● 功能块定义内的使用示例

使用数组变量提取商和余数。

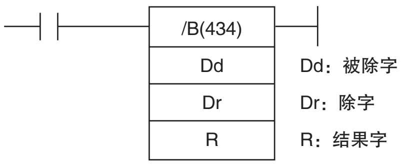
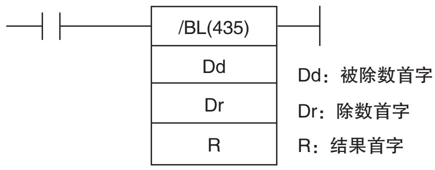
a / b → c · · · d



使用变量：
- 非除法运算数据a(数据类型UNIT)
- 除法运算数据b(数据类型UNIT)
- 商c(数据类型UNIT)
- 余数d(数据类型UNIT)
- 临时变量tmp(数据类型WORD，数组2元素)

/B，/BL

指令	助记符	变化	功能代码	功能
BCD 除	/B	@/B	434	4 位数 (单字)BCD 数据和 / 或常数相除。
双字 BCD 除	/BL	@/BL	435	8 位数 (双字)BCD 数据和 / 或常数相除。

	/B	/BL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		/B	/BL	/B	/BL
Dd	/B：　被除字 /BL：被除数首字	WORD	DWORD	1	2
Dr	/B：　除字 /BL：除数首字	WORD	DWORD	1	2
R	结果首字	DWORD	LWORD	2	4

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
/B	Dd，Dr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
/BL	Dd，Dr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

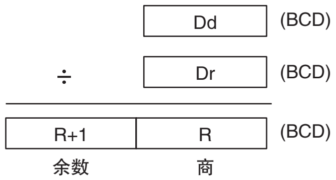
条件标志

名称	标记	操作	
		/B	/BL
出错标志	P_ER	· Dd 不是 BCD 时 ON。 · Dr 不是 BCD 时 ON。 · 除数为 0 时 ON。 · 其它情况下 OFF。	· Dd 和 Dd+1 不是 BCD 时 ON。 · Dr 和 Dr+1 不是 BCD 时 ON。 · 除数为 0 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 相除的结果 R 为 0 时 ON。 · 其它情况下 OFF。	· 相除的结果 R+1 和 R 为 0 时 ON。 · 其它情况下 OFF。

功能

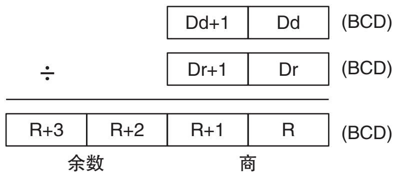
● /B

/B(434) 指令将 Dd 中的 BCD 内容除以 Dr 中的 BCD 内容，并将商输出到 R，将余数输出到 R+1。

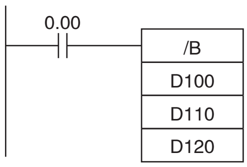


● /BL

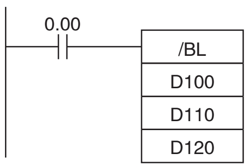
/BL(435) 指令将 Dd 和 Dd+1 中的 BCD 值除以 Dr 和 Dr+1 中的 BCD 值，并将商输出到 R 和 R+1，将余数输出到 R+2 和 R+3。



程序举例



该例中，当CIO 0.00为ON时，将D110作为4位数的BCD值除以D100，并将商输出到D120中，并将余数输出到D121中。



该例中，当CIO 0.00为ON时，将D111和D110作为8位数BCD值除以D101和D100，将商输出到D121和D120中，并将余数输出到D123和D122中。

转换指令

BIN/BINL

指令	助记符	变化	功能代码	功能
BCD →二进制	BIN	@BIN	023	将 BCD 数据转换为二进制数据。
双字 BCD →双字二进制	BINL	@BINL	058	将 8 位数 BCD 数据转换为 8 位数十六进制 (32 位二进制) 数据。

	BIN	BINL
符号	BIN(023) S R S：源字 R：结果字	BINL(058) S R S：源首字 R：结果首字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		BIN	BINL	BIN	BINL
S	BIN：源字 BINL：源首字	WORD	DWORD	1	2
R	BIN：结果字 BINL：结果首字	UINT	UDINT	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
BIN	S，R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
BINL												---					

条件标志

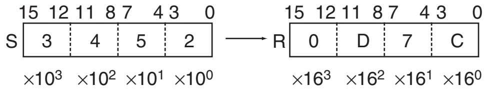
名称	标记	操作	
		BIN	BINL
出错标志	P_ER	・S 的内容不是 BCD 时 ON。 ・其它情况下 OFF。	・S+1 和 S 的内容不是 BCD 时 ON。 ・其它情况下 OFF。
等于标志	P_EQ	・结果为 0000 时 ON。 ・其它情况下 OFF。	・结果为 0 时 ON。 ・其它情况下 OFF。
负标志	P_N	OFF	OFF

功能

● BIN

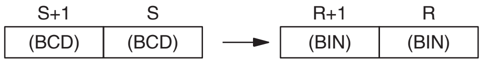
BIN(023) 指令将 S 中的 BCD 数据转换成二进制数据，并将结果写入 R 中。

下图所示为 BCD 到二进制转换举例。

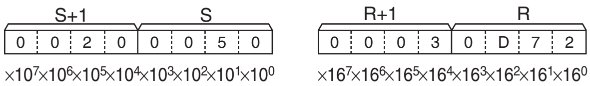


● BINL

BINL(058) 指令将 S 和 S+1 中的 8 位数 BCD 数据转换为 8 位数十六进制 (32 位二进制) 数据，并将结果写入 R 和 R+1 中。

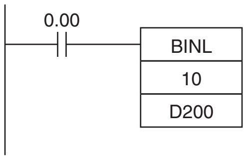


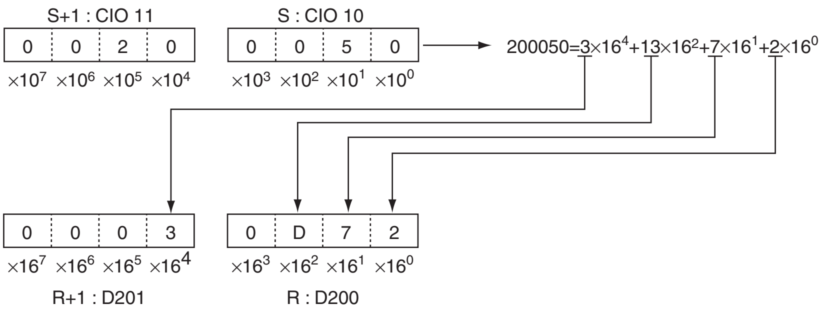
下图所示为 8 位数 BCD 到二进制转换举例。



程序举例

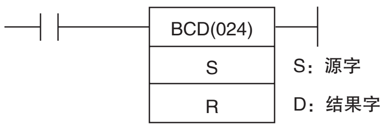
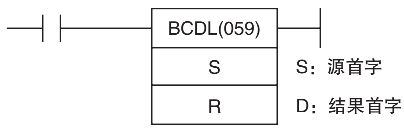
下例中，当 CIO 0.00 为 ON 时，CIO 0010 和 CIO 0011 中的 8 位数 BCD 值将转换成十六进制值并保存到 D200 和 D201 中。





BCD/BCDL

指令	助记符	变化	功能代码	功能
二进制→ BCD	BCD	@BCD	024	将一个字的二进制数据转换成一个字的 BCD 数据。
双字二进制→ 双字 BCD	BCDL	@BCDL	059	将 8 位数十六进制 (32 位二进制) 数据转换为 8 位数 BCD 数据。

	BCD	BCDL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		BCD	BCDL	BCD	BCDL
S	BCD：源字 BCDL：源首字	UINT	UDINT	1	2
R	BCD：结果字 BCDL：结果首字	WORD	DWORD	1	2

S：源字 (BCD)/ 源首字 (BCDL)

- BCD
  S 必须介于十六进制的 0000 ～ 270F(十进制的 0000 ～ 9999) 之间。
- BCDL
  S+1 和 S 的内容必须介于十六进制的 0000 0000 ～ 05F5 E0FF(十进制的 0000 0000 ～ 9999 9999) 之间。

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
BCD	S，R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
BCDL	S，R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

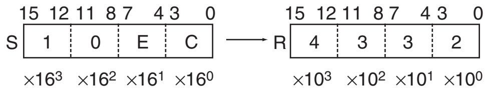
名称	标记	操作	
		BCD	BCDL
出错标志	P_ER	· S 的内容超过 270F(十进制的 9999) 时 ON。 · 其它情况下 OFF。	· S 和 S+1 的内容超过 05F5 E0FF(十进制的 9999 9999) 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0000 时 ON。 · 其它情况下 OFF。	· 结果为 0 时 ON。 · 其它情况下 OFF。

功能

● BCD

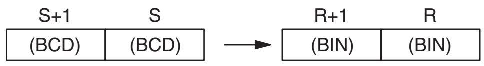
BCD(024) 指令将 S 中的二进制数据转换成 BCD 数据，并将结果写入 R 中。

下图所示为 BCD 到二进制转换举例。

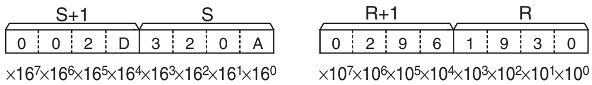


● BCDL

BCDL(059) 指令将 S 和 S+1 中的 8 位数十六进制 (32 位二进制) 数据转换为 8 位数 BCD 数据，并将结果写入 R 和 R+1 中。

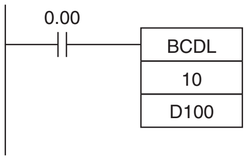


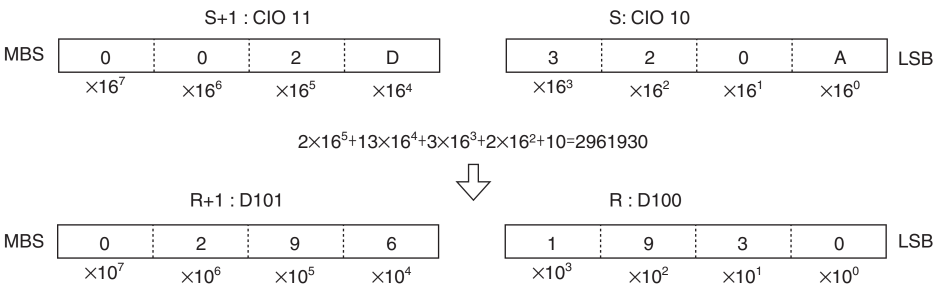
下图所示为 8 位数 BCD 到二进制转换举例。



程序举例

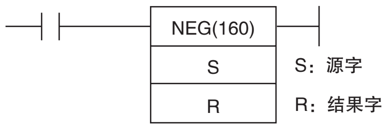
下例中，当 CIO 0.00 为 ON 时，CIO 11 和 CIO 10 中的十六进制值将转换成 BCD 值并保存到 D100 和 D101 中。





NEG

指令	助记符	变化	功能代码	功能
二进制求补	NEG	@NEG	160	计算一个字的十六进制数据的 2 的补码。

符号	NEG
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	WORD	1
R	结果字	UINT	1

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
R										---						

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果为 0000/0000 0000 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的位 15 为 ON 时 ON。 · 其它情况下 OFF。

功能

● NEG

NEG(160) 计算 S 的 2 的补码，并将结果写入 R 中。2 的补码的计算一般为 S 中各个位的状态取反后加 1。

$\overline{(S)} \xrightarrow{2的补码(补码+1)} (R)$

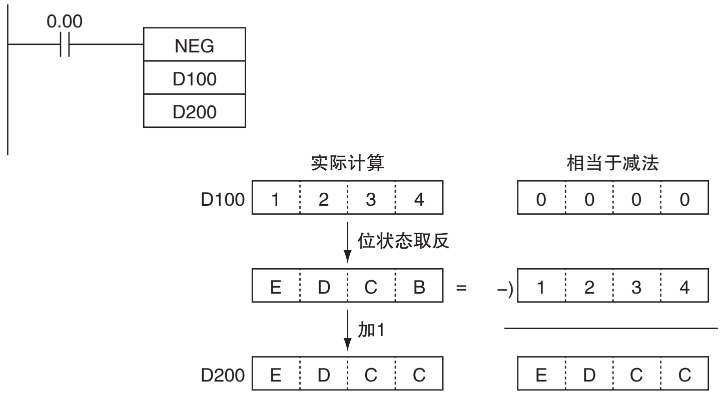
注 8000 Hex 的结果为 8000 Hex。

提示

· 操作 (将位的状态取反后加 1) 等同于从 0000/0000 0000 减去 S/S+1 和 S 的内容。

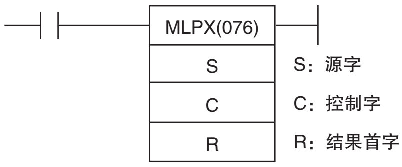
程序举例

下例中，当 CIO 0.00 为 ON 时，NEG(160) 计算 D100 中内容的 2 的补码，并将结果写入 D200 中。



MLPX

指令	助记符	变化	功能代码	功能
数据译码	MLPX	@MLPX	076	通过 4 → 16 位转换 (或 8 → 256 位转换) 读取源字中指定数位 (或字节) 的数值，并将结果字中相应的位置 ON，将结果字中的其余位置 OFF。

	MLPX
符号	

适用程序区

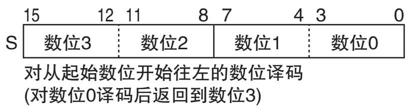
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	UINT	1
C	控制字	UINT	1
R	结果首字	UINT	可变

● 4 → 16 位译码

S：源字



R：结果首字

D：第 1 个译码数位的译码结果

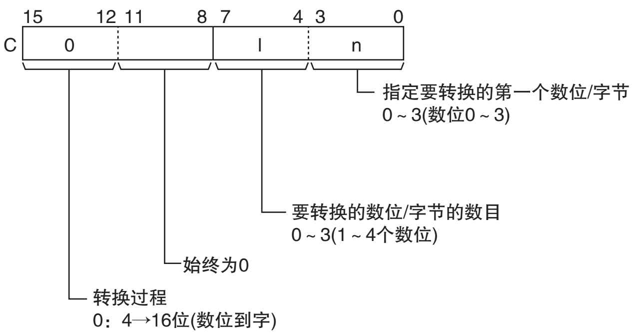
D+1：第 2 个译码数位的译码结果

D+2：第 3 个译码数位的译码结果

D+3：第 4 个译码数位的译码结果

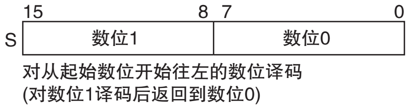
注 结果字必须位于同一个数据区。

C：控制字



● 8 → 256 位转换

S：源字



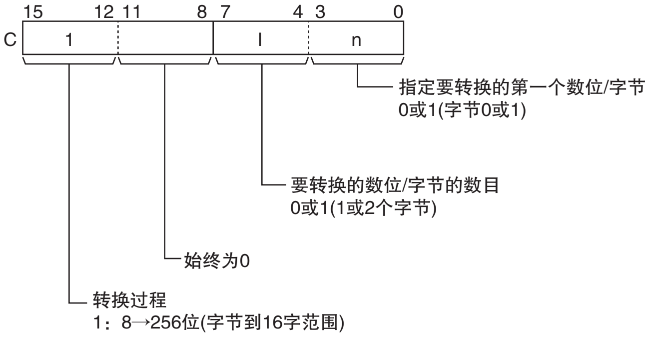
R：结果首字

D+15 ～ D：第 1 个译码数位的译码结果

D+31 ～ D+16：第 2 个译码数位的译码结果

注 结果字必须位于同一个数据区。

C：控制字



● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

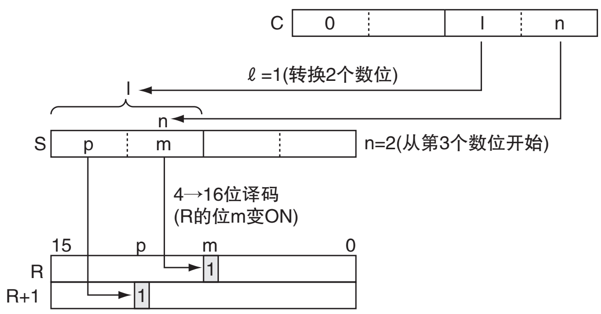
名称	标记	操作
出错标志	P_ER	· 当 C 不在指定范围内时 ON。 · 其它情况下 OFF。

功能

MLPX(076) 可执行 4 → 16 位或 8 → 256 位转换。将 C 的最左位置 0 时可指定 4 → 16 位转换，置 1 时可指定 8 → 256 位转换。

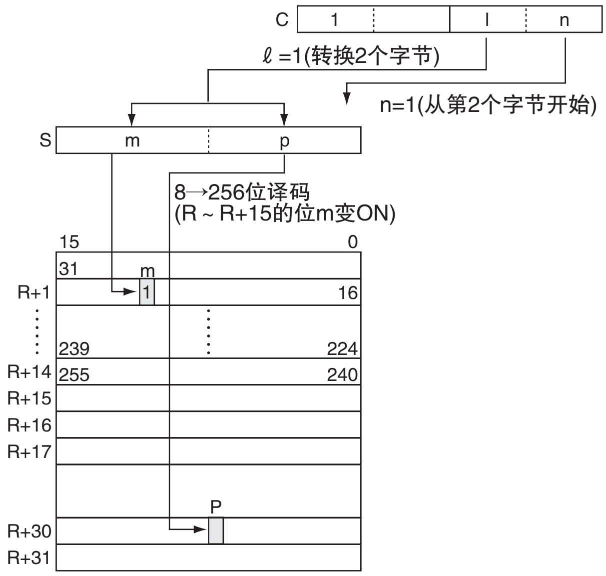
● 4 → 16 位转换

当 C 的最左位为 0 时，MLPX(076) 读取 S 中指定位的值 (0 ～ F)，并将结果字中的相应位置 ON。结果字中的其余位均置 OFF。最多可转换 4 个数位。



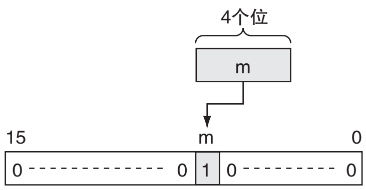
● 8 → 256 位转换

当 C 的最左位为 1 时，MLPX(076) 读取 S 中指定字节的值 (00 ～ FF)，并将 16 个结果字范围中的相应位置 ON。结果字中的其余位均置 OFF。最多可转换 2 个字节。

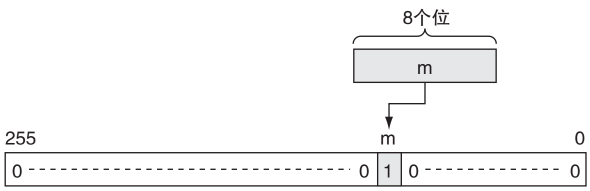


提示

如右图所示，4 → 16 位译码的操作是将 4位二进制值作为位号，并对该位号置1，而对 16 位中的其余位号置 0。



如右图所示，8 → 256 位译码的操作是将 8位二进制值作为位号，并对该位号置1，而对 256 位中的其余位号置 0。



注意事项

● 4 → 16 位转换

当转换 2 个或 2 个以上的数位时，MLPX(076) 指令将从右到左读取 S 中的数位，并在必要时从最左位绕回最右位。

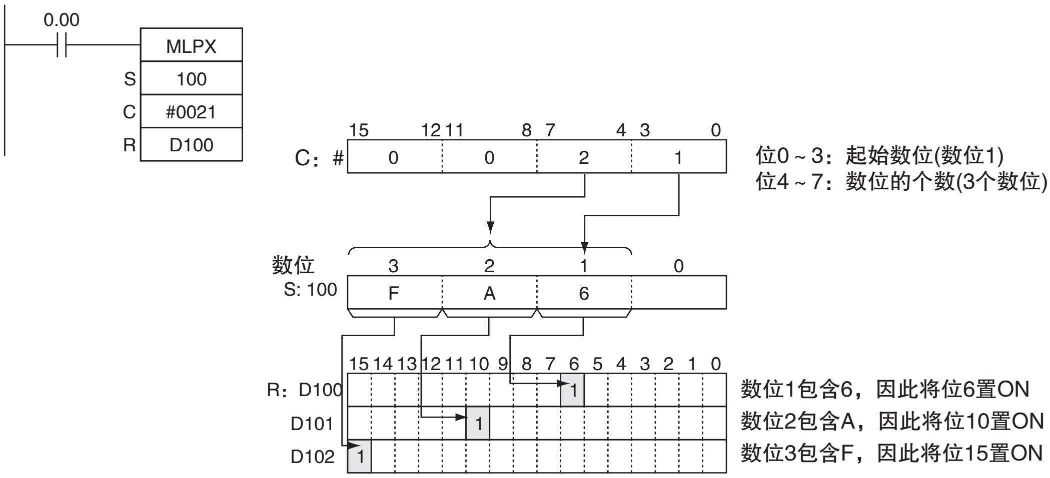
● 8 → 256 位转换

当转换 2 个字节时，MLPX(076) 指令将从右到左读取 S 中的字节，如果最左边的字节 (字节 1) 被指定为起始字节，则将绕回最右字节。

程序举例

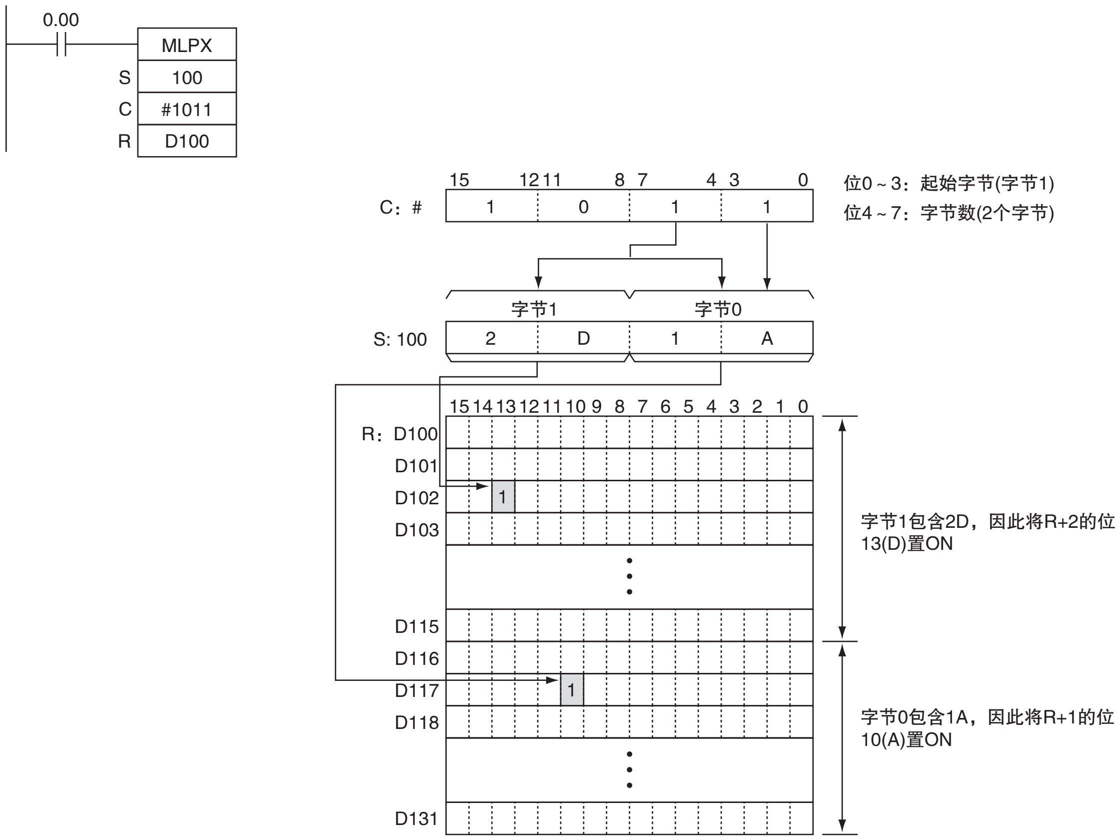
● 4→16 位转换

下例中，当 CIO 0.00 为 ON 时，MLPX(076) 指令将转换 S 中从数位 1(第 2 个数位) 开始的 3 个数位，如 C (#0021) 所指示。D100、D101 和 D102 中的相应位将变 ON。



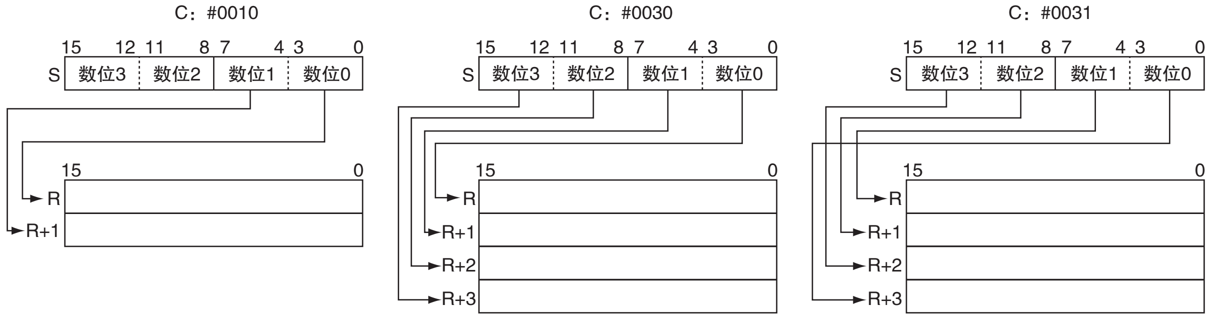
● 8→256 位转换

下例中，当 CIO 0.00 为 ON 时，MLPX(076) 指令将转换 S 中从字节 1(最左边的字节) 开始的 2 个字节，如 C(#1011) 所指示。D100 ～ D115 和 D116 ～ D131 中的相应位将变 ON。

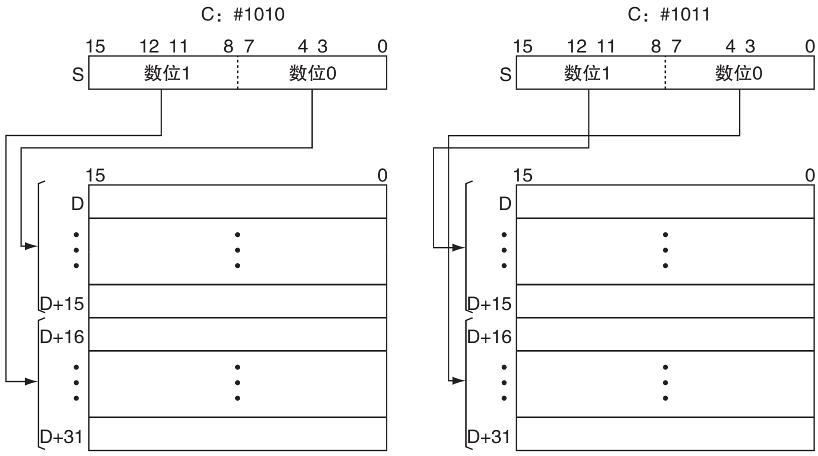


● 对多个数位译码的编程举例

· 4 → 16 位译码举例



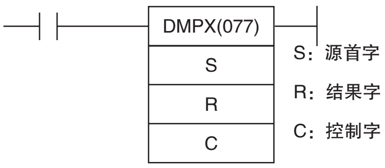
· 8 → 256 位译码举例



DMPX

指令	助记符	变化	功能代码	功能
数据编码	DMPX	@DMPX	077	通过 16→4 转换 (或 256→8 转换) 寻找源字范围内的第一个或最后一个 ON 位的位置，并将该值写入结果字的指定数位 (或字节) 中。

符号	DMPX



适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

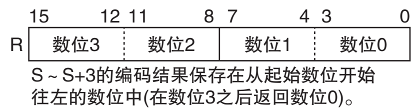
操作数	描述	数据类型	大小
S	源首字	UINT	可变
R	结果字	UINT	1
C	控制字	UINT	1

● 16→4 位转换

S：源首字

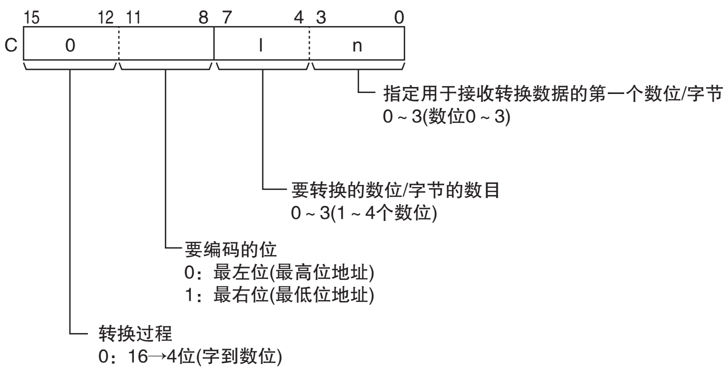
S：要编码的第 1 个数位
S+1：要编码的第 2 个数位
S+2：要编码的第 3 个数位
S+3：要编码的第 4 个数位

R：结果字



注 源字必须位于同一个数据区。

C：控制字



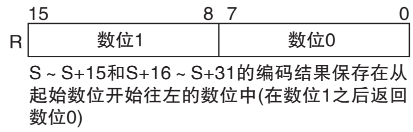
● 256 → 8 位转换

S：源首字

S+15 ～ S：要编码的第 1 个数位

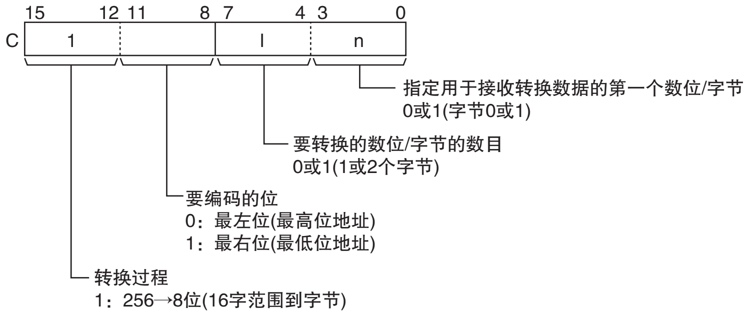
S+31 ～ S+16：要编码的第 2 个数位

R：结果字



注 源字必须位于同一个数据区。

C：控制字



● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
R											OK					
C										OK						

条件标志

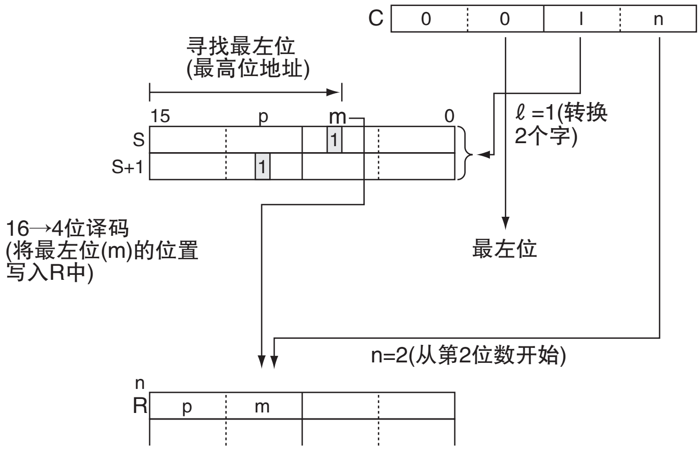
名称	标记	操作
出错标志	P_ER	· 当任何源字中含 0000 Hex(即没有要编码的位) 时 ON。 · 当 C 不在指定范围内时 ON。 · 其它情况下 OFF。

功能

DMPX(077) 可执行 16 → 4 位或 256 → 8 位转换。将 C 的最左位置 0 时可指定 16 → 4 位转换，置 1 时可指定 256 → 8 位转换。

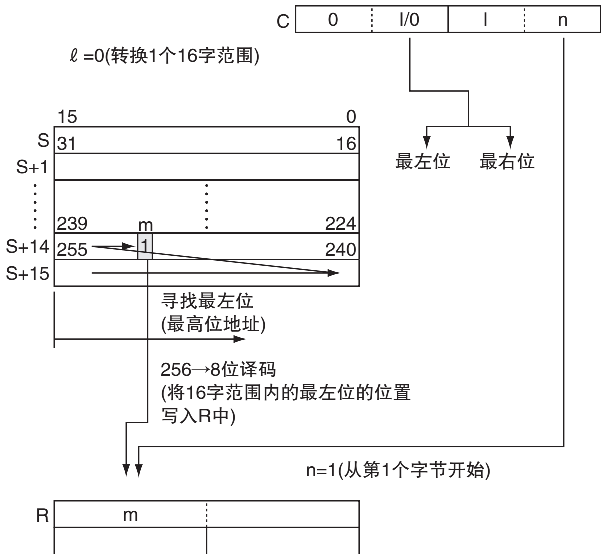
● 16 → 4 位转换

当 C 的第 4 个 (最左边) 数位为 0 时，DMPX(077) 在最多 4 个源字中寻找最左边或最右边的 ON 位的位置，并将这些位置写入 R 中从指定数位开始的各个数位中。



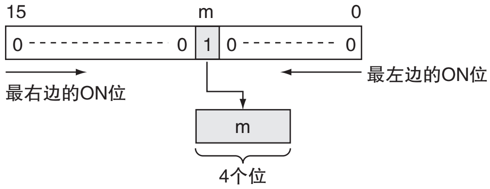
● 256 → 8 位转换

当 C 的第 4 个最左边数位为 1 时，DMPX(077) 在源字的 1 个或 2 个 16 字范围内寻找最左边 (最高位地址)或最右边(最低位地址)的ON位的位置。这些位的位置将被写入R中从指定字节开始的字节中。

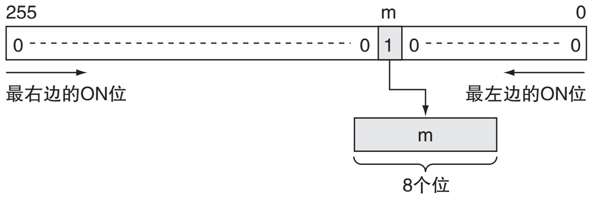


提示

如右图所示，16 → 4 位编码的操作是将 16 位中被置为 1 的最左位或最右位的位号 (m) 转换成 4 位二进制值。



如右图所示，256 → 8 位编码的操作是将 256 位中被置为 1 的最左位或最右位的位号 (m) 转换成 8 位二进制值。



注意事项

● 16 → 4 位转换

当转换 2 个或 2 个以上的数位时，DMPX(077) 从右到左将值写入 R 的数位中，并在必要时从最左位绕回最右位。

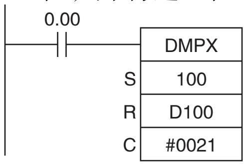
● 256 → 8 位转换

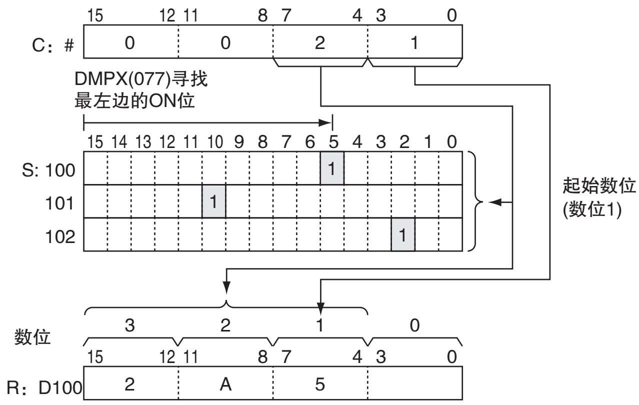
当转换 2 个字节时，DMPX(077) 从右到左将值写入 R 的字节中，如果最左边的字节 (字节 1) 被指定为起始字节，则将绕回最右字节。

程序举例

● 16 → 4 位转换

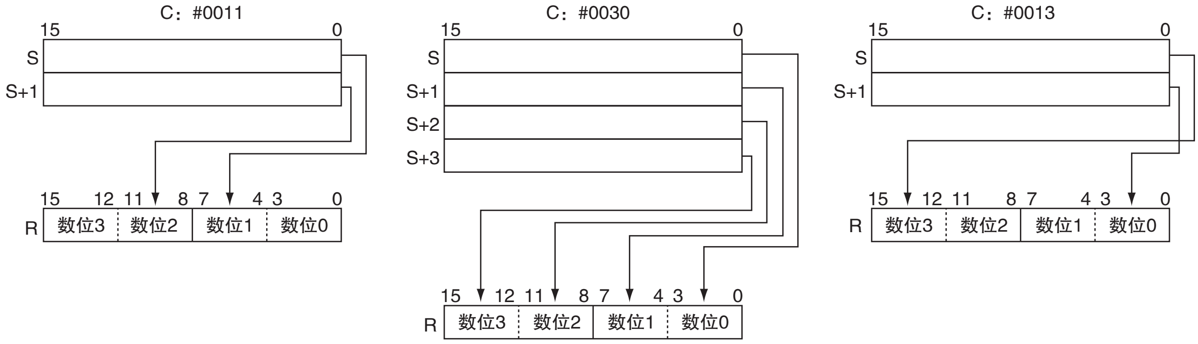
下例中，当 CIO 0.00 为 ON 时，DMPX(077) 指令将在 CIO 100、CIO 101 和 CIO 102 中寻找最左边的 ON 位，并将这些位置写入 R 中从数位 1(第 2 个数位) 开始的 3 个数位中，如 C (#0021) 所指示。

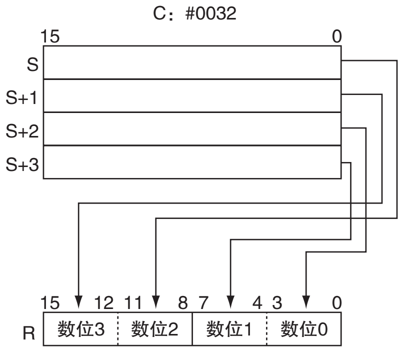




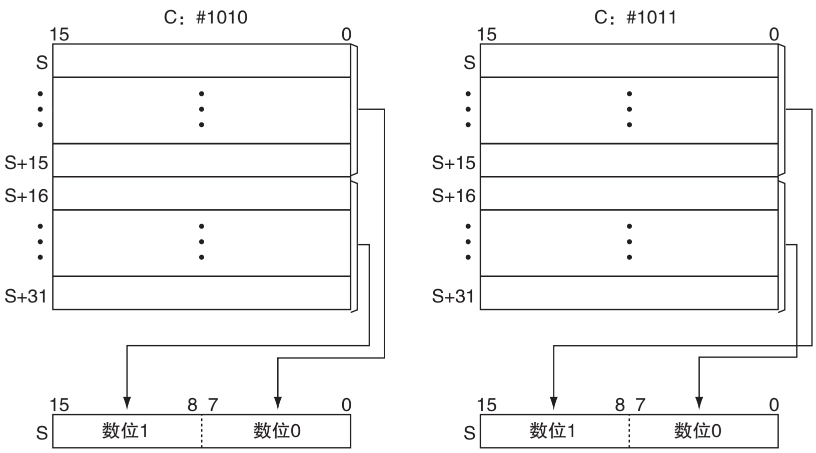
● 对多个数位译码的编程举例

· 16 → 4 位译码举例





● 256 → 8 位转换



如果转换数据中含 0000 Hex，但要对其它数据编码，则可通过多次使用 DMPX(077) 指令来分别转换数据。

DMPX(077) D0 D100 #0300

⇓

DMPX(077) D0 D100 #0000

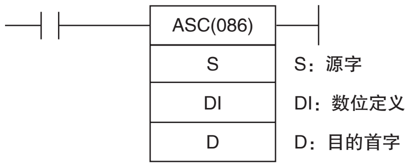
DMPX(077) D1 D100 #0001

DMPX(077) D2 D100 #0002

DMPX(077) D3 D100 #0003

ASC

指令	助记符	变化	功能代码	功能
ASCII 转换	ASC	@ASC	086	将源字中的 4 位十六进制数位转换成等值的 8 位 ASCII 码。

	ASC
符号	

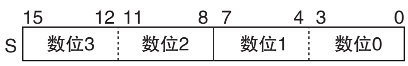
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	UINT	1
DI	数位定义	UINT	1
D	目的首字	UINT	可变

S：源字



将要转换的数据中从起始数位开始往左的数位
当做十六进制数据，并将这些数据转换为
ASCII码(在数位3之后返回数位0)

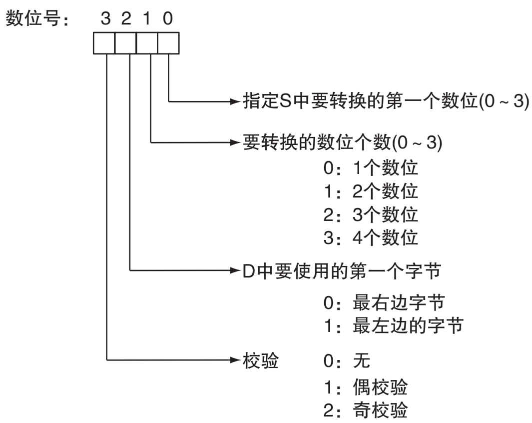
D：目的首字

ASCII码保存在左字中。
代码从D的输出起始字节开始
按照右字节、左字节的顺序保存。

注 目的字必须位于同一个数据区。

DI：数位定义

数位定义指定了用于转换的各种参数，如下图所示。



● 区域类型

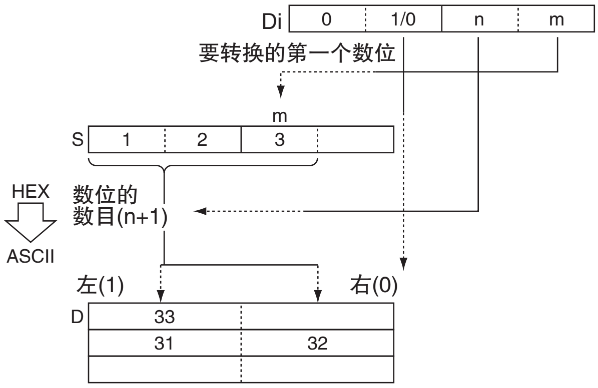
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
DI										OK						
D										---	---					

条件标志

名称	标记	操作
出错标志	P_ER	・当 DI 的内容不在指定范围内时 ON。 ・其它情况下 OFF。

功能

ASC(086) 指令将 S 的内容当做 4 个十六进制数位处理，将 S 中指定的数位转换成等值的 8 位 ASCII 码，并将该数据写入目的字中从由 D 指定的字节开始的字节中。在 ASCII 码数据中可规定将最左位用于校验 (K 的位 12 ～ 15)，且可转换成成奇校验位或偶校验位 (8 个位中值为 1 的位数调整为奇数或偶数)。



提示

・在数据后添加校验位，用于在数据传输时检测错误。添加校验位后，可指明数据中为 1 的位数是奇数还是偶数；如果接收到的数据中 1 的位数的奇偶性与之不同，则可认为发生了错误。

注意事项

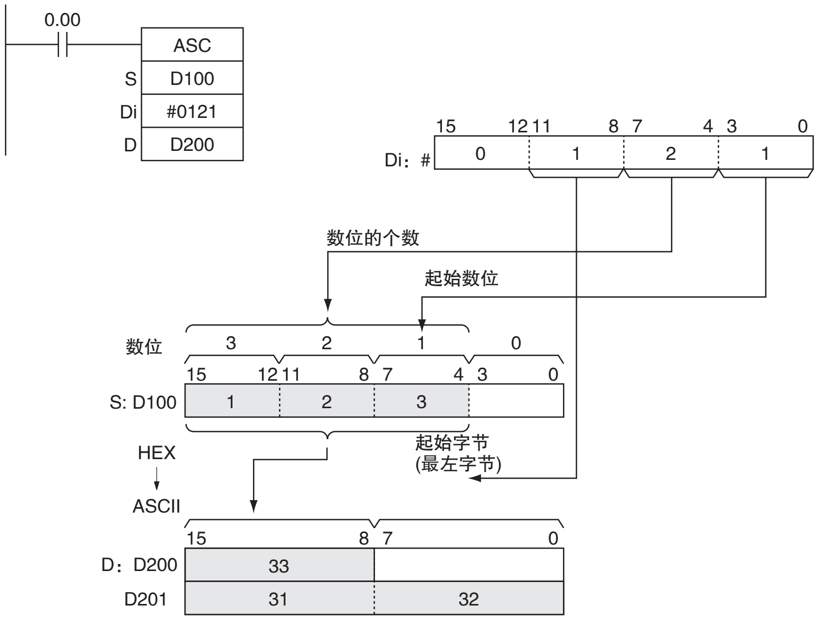
・当在要转换的数位个数 (K) 中指定了多个数位时，各数位将按照从起始转换位开始从右到左的顺序转换 (在数位 3 之后返回数位 0)，且转换结果按照从 D 的输出位置开始从右到左的顺序保存 (以 8 位为单位)。

・在转换结果输出字的各数据中，不输出的位置处的数据将保持不变。

・当转换多个数位时，请注意勿使 D+1 和 D+2CH 超出数据区。

程序举例

下例中，当 CIO 0.00 为 ON 时，ASC(086) 指令将 D100 中 (从数位 1 开始) 的三个十六进制数位转换成等值的 ASCII 码，并将转换后的数据写入 D200 和 D201 中从 D200 的最左边字节开始的字节中。在这种情况下，#0121 数位定义指定无校验，起始字节 (写入时)= 最左边的字节，要读取的数位个数 =3，且起始数位 (读取时)= 数位 1。



● ASCII 码转换举例

转换数据的数位内容					转换输出数据								
值	位内容				代码	(MSB) 位内容 (LSB)							
0	0	0	0	0	#30	*	0	1	1	0	0	0	0
1	0	0	0	1	#31	*	0	1	1	0	0	0	1
2	0	0	1	0	#32	*	0	1	1	0	0	1	0
3	0	0	1	1	#33	*	0	1	1	0	0	1	1
4	0	1	0	0	#34	*	0	1	1	0	1	0	0
5	0	1	0	1	#35	*	0	1	1	0	1	0	1
6	0	1	1	0	#36	*	0	1	1	0	1	1	0
7	0	1	1	1	#37	*	0	1	1	0	1	1	1
8	1	0	0	0	#38	*	0	1	1	1	0	0	0
9	1	0	0	1	#39	*	0	1	1	1	0	0	1
A	1	0	1	0	#41	*	1	0	0	0	0	0	1
B	1	0	1	1	#42	*	1	0	0	0	0	1	0
C	1	1	0	0	#43	*	1	0	0	0	0	1	1
D	1	1	0	1	#44	*	1	0	0	0	1	0	0
E	1	1	1	0	#45	*	1	0	0	0	1	0	1
F	1	1	1	1	#46	*	1	0	0	0	1	1	0

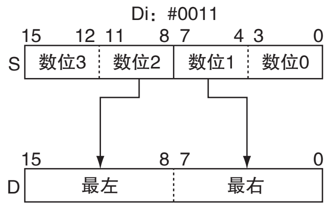
* 校验位 – 根据校验规定而改变。

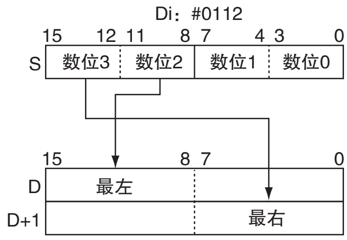
● 校验

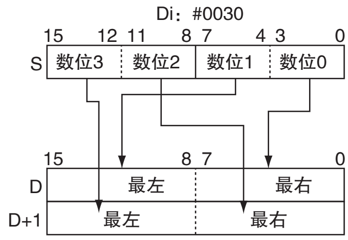
可以指定 ASCII 数据的校验位，用于在数据传输期间控制错误。每个 ASCII 字符的最左位将根据奇校验、偶校验或无校验进行自动调整。

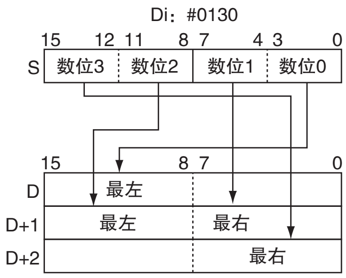
- 当指定了无校验 (0) 时，最左位将始终为零。当指定了偶校验 (1) 时，将对最左位进行调整，使得 ON 位的总位数为偶数。当指定了奇校验 (2) 时，将对每个 ASCII 字符的最左位进行调整，使得 ON 位的总位数为奇数。校验位的状态不影响 ASCII 码的意义。
- 偶校验举例：
  当对偶校验进行调整时，ASCII 码 “31”(00110001) 将为 “B1”(10110001：校验位置 ON，以产生偶数个 ON 位)； ASCII 码 “36”(00110110) 将为 “36”(00110110：校验位保持 OFF，因为 ON 的位数已经是偶数)。
- 奇校验举例：
  当对奇校验进行调整时，ASCII 码 “36”(00110110) 将为 “B6”(10110110：校验位置 ON，以产生奇数个 ON 位)； ASCII 码 “46”(01000110) 将为 “46”(01000110：校验位保持 OFF，因为 ON 的位数已经是奇数)。

● Di 举例



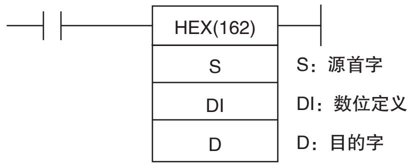






HEX

指令	助记符	变化	功能代码	功能
ASCII →十六进制	HEX	@HEX	162	将源字中的最多 4 个字节的 ASCII 数据转换成等值的十六进制数位，并将这些数位写入指定的目的字中。

	HEX
符号	

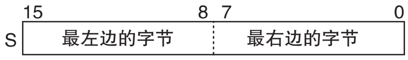
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

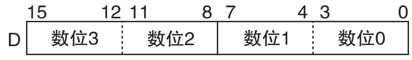
操作数	描述	数据类型	大小
S	源首字	UINT	可变
DI	数位定义	UINT	1
D	目的字	UINT	1

S：源首字



将从转换起始字节开始往左的字节作为ASCII码处理，并将其转换成十六进制数(在最左边字节后返回到最右边字节)。

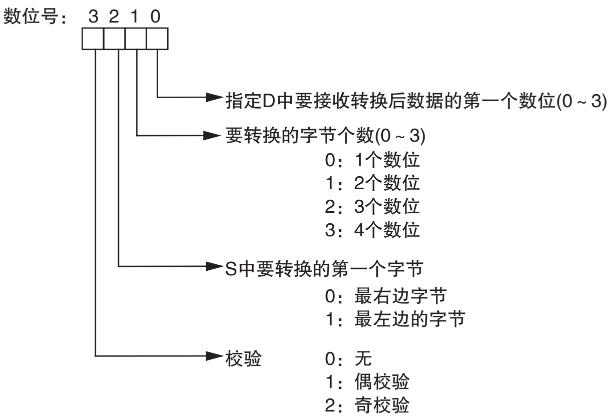
D：目的字



转换成十六进制的结果保存在从起始数位开始往左的数位中(在数位3之后返回数位0)。

DI：数位定义

数位定义指定了用于转换的各种参数，如下图所示。



- **区域类型**

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
DI	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

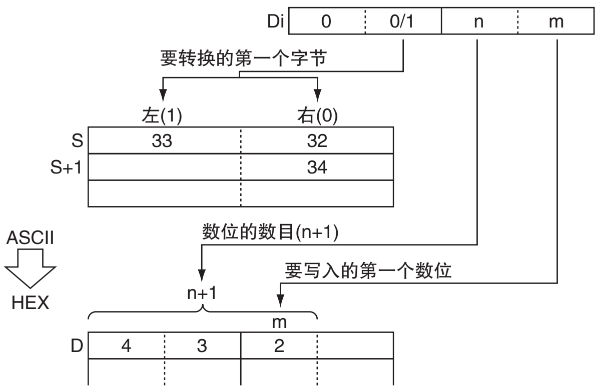
条件标志

名称	标记	操作
出错标志	P_ER	· 当 ASCII 数据中有校验错误时 ON。 · 当源字中的 ASCII 数据不等于十六进制数位时 ON。 · 当 DI 的内容不在指定范围内时 ON。 · 其它情况下 OFF。

功能

HEX(162) 指令将源字的内容当做表示十六进制数位 (0 ~ 9 和 A ~ F) 的 ASCII 数据处理，将指定个数的字节转换成十六进制数据，并将该十六进制数据写入目的字中从指定数位开始的数位中。

转换数据时，可根据校验规定，将 ASCII 码数据的最左位作为奇校验或偶校验位处理。



提示

· 在数据后添加校验位，用于在数据传输时检测错误。添加校验位后，可指明数据中为 1 的位数是奇数还是偶数；如果接收到的数据中 1 的位数的奇偶性与之不同，则可认为发生了错误。

注意事项

· 当在要转换的数位个数 (C) 中指定了多个数位时，各数位将按照 S 中从起始转换位 (C) 开始从右到左的顺序转换，且转换结果按照从 D 的输出起始位 (C) 开始从右到左的顺序保存 (在数位 3 之后返回数位 0)。

· 在转换结果输出字的各数据中，不输出的位中的数据将保持不变 (与先前相同)。

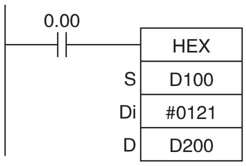
· 下表所示为可包含在源字 (不包括校验位) 和等值的十六进制数位中的 ASCII 数据。

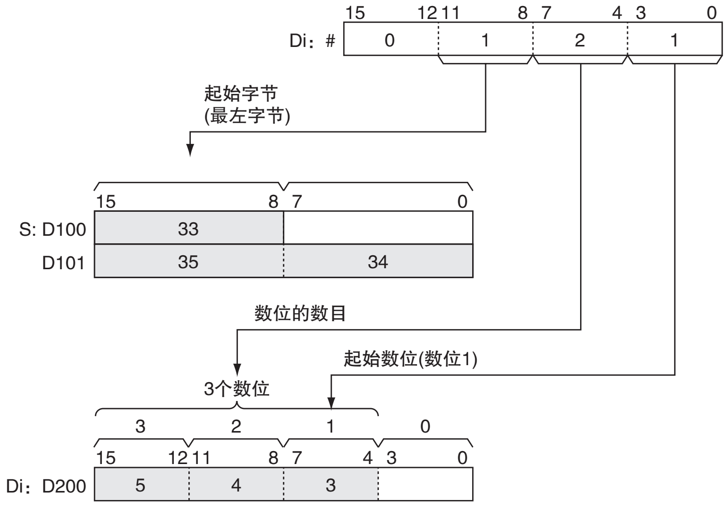
ASCII 数据 (2 个十六进制数位)	十六进制数位
30 ~ 39	0 ~ 9
41 ~ 46	A ~ F

程序举例

下例中，当 CIO 0.00 为 ON 时，HEX(162) 指令根据数位定义中的设定来转换 D100 和 D101 中的 ASCII 数据。(Di=#0121 指定无校验，起始字节 (读取时)= 最左边的字节，要读取的字节个数 =3，且起始数位 (写入时)= 数位 1。)

HEX(162) 指令将 D100 中从最左边的字节开始的三个字节的 ASCII 数据 (3 个字符) 转换为等值的十六进制数据，并将该数据写入 D200 中从数位 1 开始的数位。





● 校验

可以指定 ASCII 数据的校验位，用于在数据传输期间控制错误。各字节中的最左位为校验位。没有校验时，校验位应始终为零；偶校验时，校验位的状态应产生偶数个 ON ；奇校验时，校验位的状态应产生奇数个 ON 位。

下表所示为 HEX(162) 指令对于各种校验设定的操作。

校验设定 (Di 的最左位)	HEX(162) 的操作
无校验 (0)	仅当各字节中的校验位为 0 时，执行 HEX(162) 指令。如果校验位不为零，则将产生错误。
偶校验 (1)	仅当各字节中 ON 位的个数为偶数时，执行 HEX(162) 指令。如果某个字节中的 ON 位个数为奇数，则将产生错误。
奇校验 (2)	仅当各字节中 ON 位的个数为奇数时，执行 HEX(162) 指令。如果某个字节中的 ON 位个数为偶数，则将产生错误。

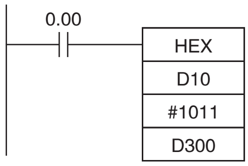
● 输出举例

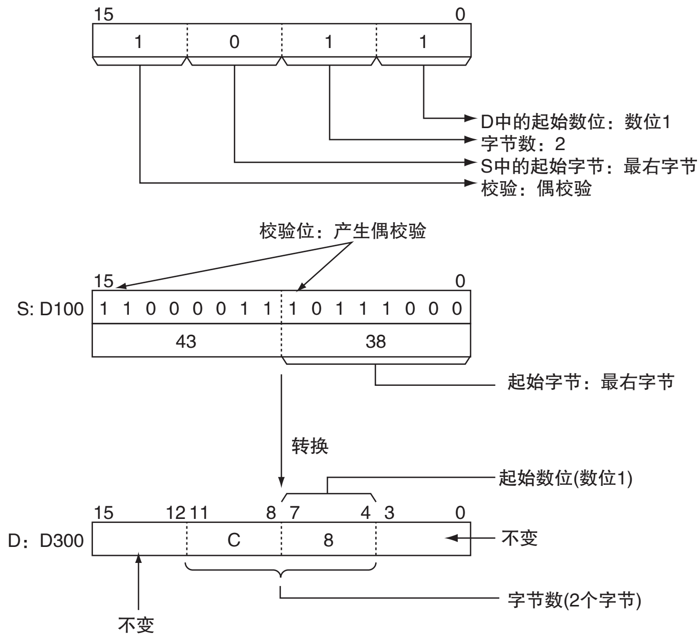
转换数据								输出结果 (十六进制数据)				
ASCII码	(MSB) 位内容 (LSB)							值	位内容			
#30	*	0	1	1	0	0	0	0	0	0	0	0	0
#31	*	0	1	1	0	0	0	1	1	0	0	0	1
#32	*	0	1	1	0	0	1	0	2	0	0	1	0
#33	*	0	1	1	0	0	1	1	3	0	0	1	1
#34	*	0	1	1	0	1	0	0	4	0	1	0	0
#35	*	0	1	1	0	1	0	1	5	0	1	0	1
#36	*	0	1	1	0	1	1	0	6	0	1	1	0
#37	*	0	1	1	0	1	1	1	7	0	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	0	0	0	0	1	A	1	0	1	0
#42	*	1	0	0	0	0	1	0	B	1	0	1	1
#43	*	1	0	0	0	0	1	1	C	1	1	0	0
#44	*	1	0	0	0	1	0	0	D	1	1	0	1
#45	*	1	0	0	0	1	0	1	E	1	1	1	0
#46	*	1	0	0	0	1	1	0	F	1	1	1	1

* 校验位 – 根据校验规定而改变。

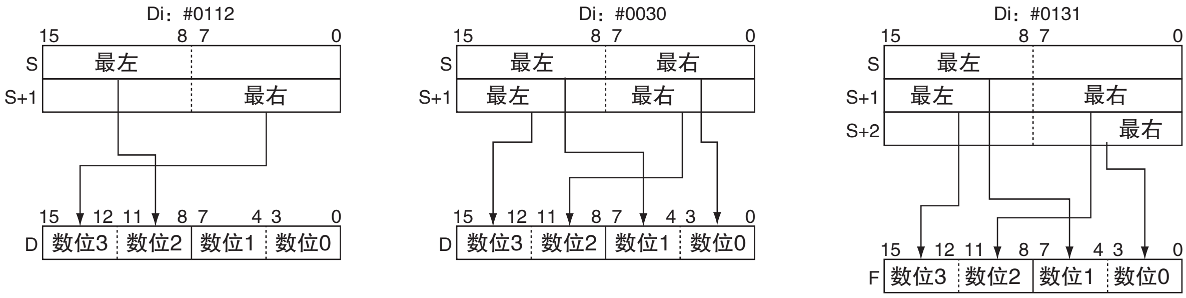
下例中，当 CIO 0.00 为 ON 时，HEX(162) 指令转换 D10 中从最右边字节开始的 ASCII 数据，并将转换后的等值十六进制数写入 D300 中从数位 1 开始的数位中。

#1011 的数位定义设定指定偶校验，起始字节 (读取时)= 最右边的字节，要读取的字节个数 =2，且起始数位 (写入时)= 数位 1。





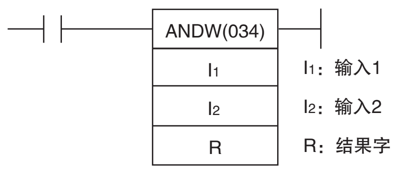
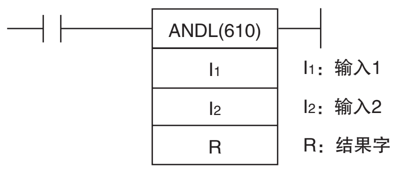
● 将多个字节的 ASCII 码转换成十六进制数举例



逻辑指令

ANDW/ANDL

指令	助记符	变化	功能代码	功能
逻辑与	ANDW	@ANDW	034	对单字数据和 / 或常数中的相应位作逻辑与运算。
双字逻辑与	ANDL	@ANDL	610	对双字数据和 / 或常数的相应位作逻辑与运算。

	ANDW	ANDL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ANDW	ANDL	ANDW	ANDL
I_1	输入 1	WORD	DWORD	1	2
I_2	输入 2	WORD	DWORD	1	2
R	结果字	WORD	DWORD	1	2

● 区域类型

<table>
<tr><th colspan="2" rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">ANDW</td><td>I1，I2</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td rowspan="2">OK</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td></tr>
<tr><td rowspan="2">ANDL</td><td>I1，I2</td><td>OK</td><td rowspan="2">---</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● ANDW

ANDW(034) 指令将 I_1 和 I_2 中指定的数据进行逻辑与运算，并将结果输出到 R。

I_1，$I_2 \rightarrow R$

I_1	I_2	R
1	1	1
1	0	0
0	1	0
0	0	0

● ANDL

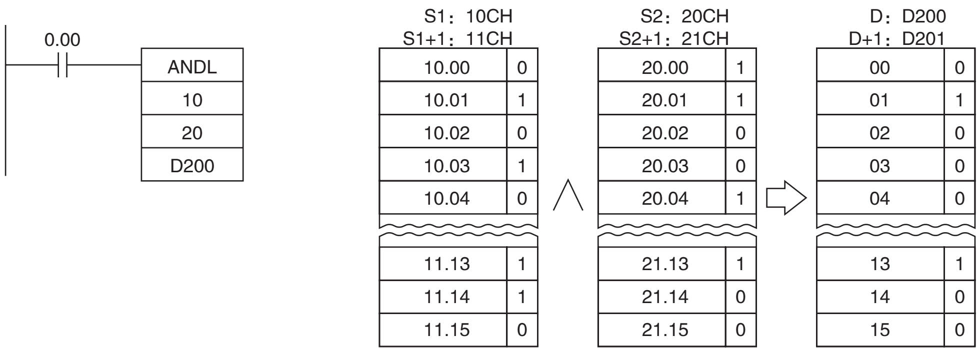
ANDL(610) 指令将 I_1，I_1+1 和 I_2，I_2+1 中指定的数据进行逻辑与运算并将结果输出到 R，R+1 中。

$(I_1, I_1+1) \cdot (I_2, I_2+1) \rightarrow (R, R+1)$

I_1，I_1+1	I_2，I_2+1	R，R+1
1	1	1
1	0	0
0	1	0
0	0	0

程序举例

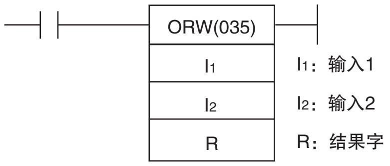
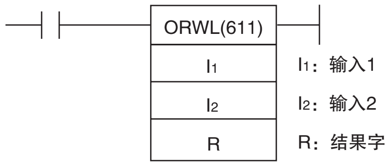
当执行条件 CIO 0.00 为 ON 时，取 CIO 11，CIO 10 和 CIO 21，CIO 20 中的相应位进行逻辑与运算，并将结果输出到 D201 和 D200 中的相应位。



注 纵向箭头表示逻辑与。

ORW/ORWL

指令	助记符	变化	功能代码	功能
逻辑或	ORW	@ORW	035	对单字数据和 / 或常数中的相应位作逻辑或运算。
双字逻辑或	ORWL	@ORWL	611	对双字数据和 / 或常数的相应位作逻辑或运算。

	ORW	ORWL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		ORW	ORWL	ORW	ORWL
I_1	输入 1	WORD	DWORD	1	2
I_2	输入 2	WORD	DWORD	1	2
R	结果字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
ORW	I_1，I_2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
ORWL	I_1，I_2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● ORW

ORW(035) 指令将 I_1 和 I_2 中指定的数据进行逻辑或运算，并将结果输出到 R 中。

I_1，I_2 → R

I_1	I_2	R
1	1	1
1	0	1
0	1	1
0	0	0

● ORWL

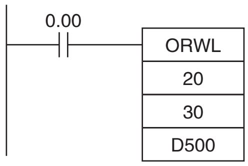
ORWL(611) 指令将 I_1 和 I_2 中的数据作为双字数据进行逻辑或运算，并将结果输出到 R，R+1 中。

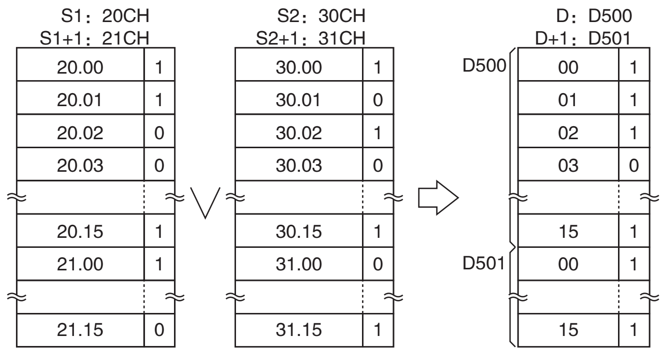
(I_1，I_1+1)+(I_2，I_2+1) → (R，R+1)

I_1，I_1+1	I_2，I_2+1	R，R+1
1	1	1
1	0	1
0	1	1
0	0	0

程序举例

当执行条件 CIO 0.00 为 ON 时，取 CIO 21，CIO 20 和 CIO 31，CIO 30 中的相应位进行逻辑或运算，并将结果输出到 D501 和 D500 的相应位。

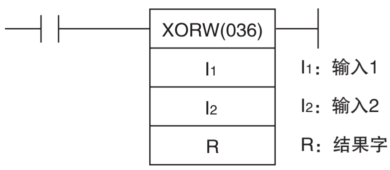
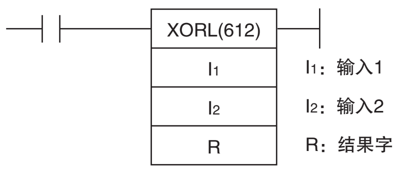




注 纵向箭头表示逻辑或。

XORW/XORL

指令	助记符	变化	功能代码	功能
异或	XORW	@XORW	036	对单字数据和 / 或常数的相应位作逻辑异或运算。
双字异或	XORL	@XORL	612	对双字数据和 / 或常数的相应位作逻辑异或运算。

	XORW	XORL
符号		

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型		大小	
		XORW	XORL	XORW	XORL
I_1	输入 1	WORD	DWORD	1	2
I_2	输入 2	WORD	DWORD	1	2
R	结果字	WORD	DWORD	1	2

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
XORW	I_1，I_2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
XORL	I_1，I_2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
	R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● XORW

XORW(036) 指令将 I_1 和 I_2 中指定的数据进行逻辑异或运算，并将结果输出到 R 中。

$$I_1 \cdot \overline{I_2} + \overline{I_1} \cdot I_2 \rightarrow R$$

I_1	I_2	R
1	1	0
1	0	1
0	1	1
0	0	0

● XORL

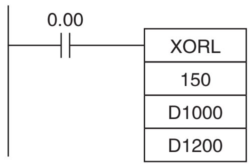
XORL(612) 指令将 I_1 和 I_2 中指定的数据作为双字数据进行逻辑异或运算，并将结果输出到 R，R+1 中。

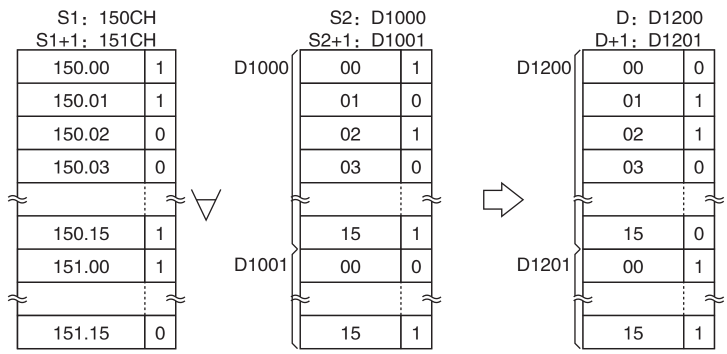
$$(I_1, I_1+1) \cdot \overline{(I_2, I_2+1)} + \overline{(I_1, I_1+1)} \cdot (I_2, I_2+1) \rightarrow (R, R+1)$$

I_1，I_1+1	I_2，I_2+1	R，R+1
1	1	0
1	0	1
0	1	1
0	0	0

程序举例

当执行条件 CIO 0.00 为 ON 时，取 CIO 151，CIO 150 和 D1001，D1000 中的相应位进行逻辑异或运算，并将结果输出到 D1201 和 D1200 中的相应位。





注 图中的符号表示逻辑异或。

COM/COML

指令	助记符	变化	功能代码	功能
求补	COM	@COM	029	将 Wd 中的所有 ON 位变 OFF，并将所有 OFF 位变 ON。
双字求补	COML	@COML	614	将 Wd 和 Wd+1 中的所有 ON 位变 OFF，并将所有 OFF 位变 ON。

	COM	COML
符号	COM(029) Wd Wd：字	COML(614) Wd Wd：字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>COM</th><th>COML</th><th>COM</th><th>COML</th></tr>
<tr><td>Wd</td><td>字</td><td>WORD</td><td>DWORD</td><td>1</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th colspan="2" rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>COM</td><td rowspan="2">Wd</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>COML</td><td>---</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R 的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● COM

COM(029) 指令对 Wd 中的每个指定位的状态取反。

$\overline{\text{Wd}}$ → Wd：1 → 0 和 0 → 1

注 使用 COM 指令时请注意，每个位的状态在执行条件为 ON 的各个循环中都将改变。

● COML

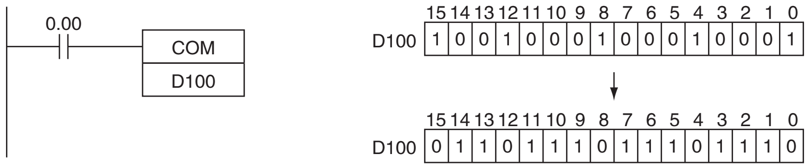
COML(614) 指令对 Wd 和 Wd+1 中的每个指定位的状态取反。

$\overline{(\text{Wd+1，Wd})}$ → (Wd+1，Wd)

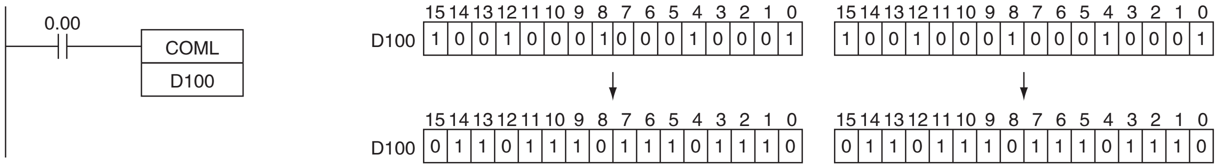
注 使用 COML 指令时请注意，每个位的状态在执行条件为 ON 的各个循环中都将改变。

程序举例

下例中，当 CIO 0.00 为 ON 时，D100 各个位的状态将取反。



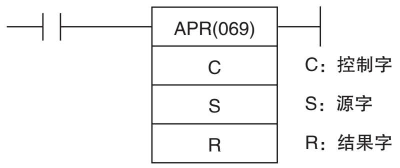
下例中，当 CIO 0.00 为 ON 时，D100 和 D101 中各个位的状态将取反。



特殊算术指令

APR

指令	助记符	变化	功能代码	功能
算术处理	APR	@APR	069	计算源数据的正弦、余弦或线性外插。

	APR
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
C	控制字	UINT	可变
S	源数据	WORD	1
R	结果字	WORD	1

● 正弦函数

操作数	值	数据范围
C	0000 Hex	---
S	0000 ～ 0900(BCD)	0° ～ 90°
R	0000 ～ 9999(BCD) 9999(BCD)	0.0000 ～ 0.9999 1.0000

- 正弦函数 (C=0000)
  当 C 为 0000 时，APR(069) 计算 SIN(S) 并将结果写入 R。S 的范围为 0000 ～ 0900 BCD(0.0° ～ 90.0°)，R 的范围为 0000 ～ 9999 BCD(0.0000 ～ 0.9999)。结果中小数点后第 4 位以后的数被舍去。

● 余弦函数

操作数	值	数据范围
C	0001 Hex	---
S	0000 ～ 0900(BCD)	0° ～ 90°
R	0000 ～ 9999(BCD) 9999(BCD)	0.0000 ～ 0.9999 1.0000

- 余弦函数 (C=0001)
  当 C 为 0001 时，APR(069) 计算 COS(S) 并将结果写入 R。S 的范围为 0000 ～ 0900 BCD(0.0° ～ 90.0°)，R 的范围为 0000 ～ 9999 BCD(0.0000 ～ 0.9999)。结果中小数点后第 4 位以后的数被舍去。

注 SIN(90°) 和 COS(0°) 的实际结果为 1，但将 9999(0.9999) 输出到 R。

● 线性外插函数

操作数	值	数据范围
C	数据区地址	---
S	16位无符号BCD数据	0000 ~ 9999
	16 位无符号二进制数据	0 ~ 65,535
	16 位带符号二进制数据	−32,768 ~ 32,767
	32 位带符号二进制数据	−2,147,483,648 ~ 2,147,483,647
	浮点数据	−∞, -3.402823×10^{38} ~ $-1.175494 \times 10^{-38}$, 1.175494×10^{-38} ~ 3.402823×10^{38}, +∞
R	16位无符号BCD数据	0000 ~ 9999
	16 位无符号二进制数据	0 ~ 65,535
	16 位带符号二进制数据	−32,768 ~ 32,767
	32 位带符号二进制数据	−2,147,483,648 ~ 2,147,483,647
	浮点数据	−∞, -3.402823×10^{38} ~ $-1.175494 \times 10^{-38}$, 1.175494×10^{-38} ~ 3.402823×10^{38}, +∞

・线性外插 (C= 数据区地址)
当 C 为字地址时，指定 APR(069) 指令执行线性外插操作。字 C 的内容指定从 C+2 开始的数据表中的坐标数，源数据的形式以及数据类型 (是 BCD 还是二进制)。

可使用以下 5 种 I/O 数据：

・16 位无符号 BCD 数据

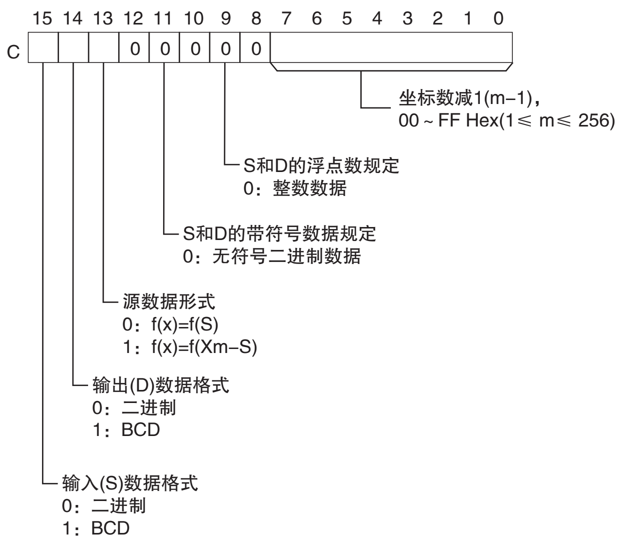
・16 位无符号二进制数据

・16 位带符号二进制数据

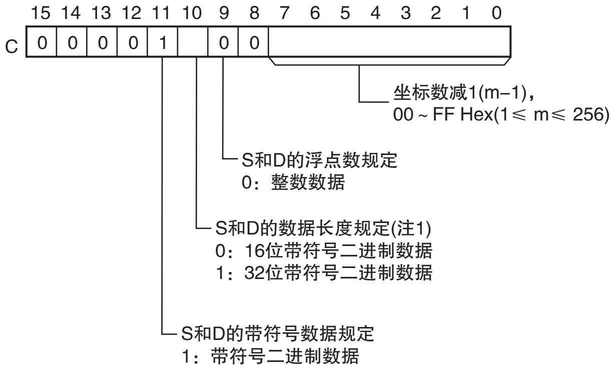
・32 位带符号二进制数据

・单精度浮点数据

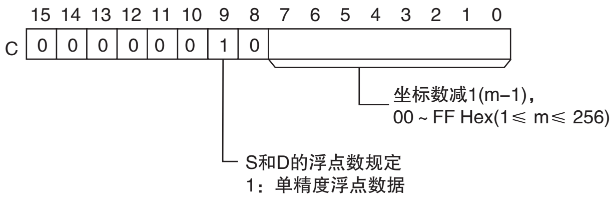
・无符号整数数据(二进制或BCD)



・ 带符号整数数据(二进制)



・ 单精度浮点数据



16位二进制(带符号或无符号)或16位BCD数据

C+1	X0(*1)
C+2	Y0
C+3	X1
C+4	Y1
C+5	X2
C+6	Y2
	Xn
	Yn
C+(2m+1)	Xm
C+(2m+2)	Ym

注：当S和D中的I/O数据包含无符号数据(C的位11=0)时，将Xm(表中最大的X值)写入字C+1。

32位带符号二进制数据

C+1	X0(最右边的16个位)
C+2	X0(最左边的16个位)
C+3	Y0(最右边的16个位)
C+4	Y0(最左边的16个位)
C+5	X1(最右边的16个位)
C+6	X1(最左边的16个位)
C+7	Y1(最右边的16个位)
C+8	Y1(最左边的16个位)
~	~
C+(4n+1)	Xn(最右边的16个位)
C+(4n+2)	Xn(最左边的16个位)
C+(4n+3)	Yn(最右边的16个位)
C+(4n+4)	Yn(最左边的16个位)
~	~
C+(4m+1)	Xm(最右边的16个位)
C+(4m+2)	Xm(最左边的16个位)
C+(4m+3)	Ym(最右边的16个位)
C+(4m+4)	Ym(最左边的16个位)

浮点数据

C+1	X0(最右边的16个位)
C+2	X0(最左边的16个位)
C+3	Y0(最右边的16个位)
C+4	Y0(最左边的16个位)
C+5	X1(最右边的16个位)
C+6	X1(最左边的16个位)
C+7	Y1(最右边的16个位)
C+8	Y1(最左边的16个位)
~	~
C+(4n+1)	Xn(最右边的16个位)
C+(4n+2)	Xn(最左边的16个位)
C+(4n+3)	Yn(最右边的16个位)
C+(4n+4)	Yn(最左边的16个位)
~	~
C+(4m+1)	Xm(最右边的16个位)
C+(4m+2)	Xm(最左边的16个位)
C+(4m+3)	Ym(最右边的16个位)
C+(4m+4)	Ym(最左边的16个位)

注： X坐标必须按升序排列：$X_1<X_2<\cdots<X_m$。 输入控制字C中指定的数据格式如何，将所有(X_n，Y_n)作为二进制数据

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

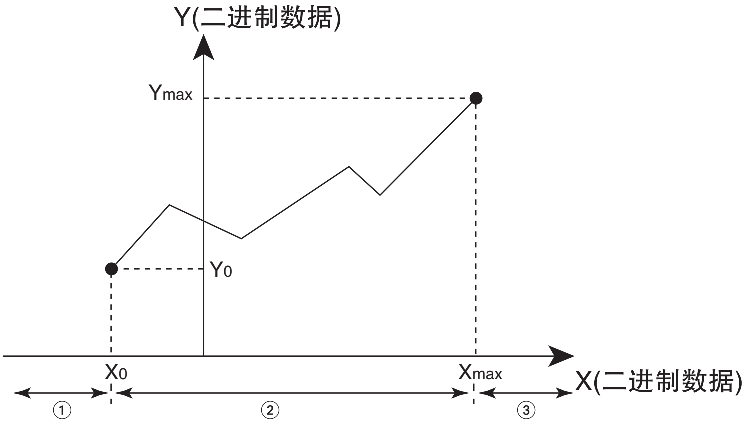
名称	标记	操作
出错标志	P_ER	· 当 C 为大于 0001 的常数时 ON。 · 当 C 为字地址，但 X 坐标不是升序排列 ($X_1 \leqslant X_2 \leqslant \cdots\cdots \leqslant X_m$) 时 ON。 · 当 C 为字地址且 C 的位 9、11 和 15 表示 BCD 输入，但 S 不是 BCD 数据时 ON。 · 当 C 为字地址且 C 的位 9 表示浮点数据，但 S 为单字常数时 ON。 · 当 C 为 0000 或 0001，但 S 不是介于 0000 ～ 0900 之间的 BCD 数据时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R 的位 15 为 ON 时 ON。 · 其它情况下 OFF。

功能

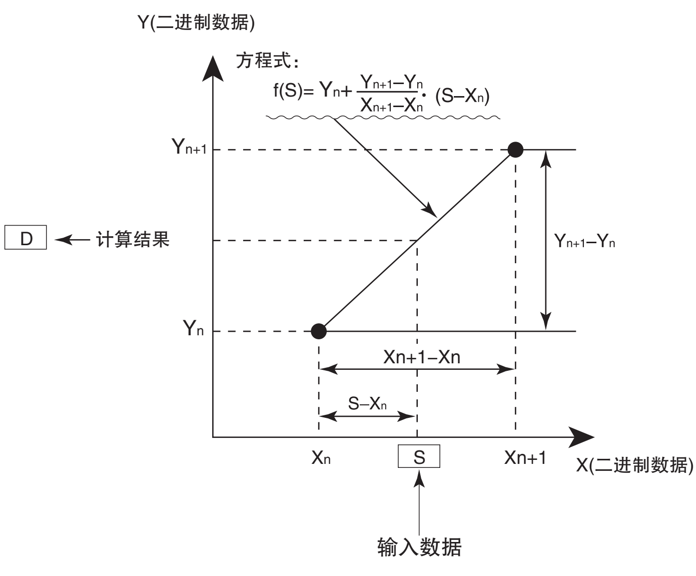
● 线性外插函数的操作

APR(069) 通过下述方程式处理在 S 中指定的输入数据和在以 C+1 开始的表中指定的线段数据 (X_n，Y_n)。结果输出到通过 D 指定的目的字中。

1. 对于 $S<X_0$
转换后的值 $=Y_0$
2. 对于 $X_0 \leqslant S \leqslant X_{max}$，如果 $X_n<S<X_{n+1}$
转换后的值 $=Y_n+[\{Y_n+1-Y_n\}/\{X_n+1-X_n\}] \times \{$ 输入数据 $S-X_n\}$
3. $X_{max}<S$
转换后的值 $=Y_{max}$



在以 C+1 开始的线段数据表中最多可存储 256 个端点。



- **16 位无符号 BCD 数据**

输入数据和 / 或输出数据可以是 16 位无符号 BCD 数据。另外，还可将线性插补函数设定为直接对 S 或 X_m–S 指定的值进行操作。(X_m 是线段数据中 X 的最大值。)

设定名称	C中的位	设定
输入数据 (S) 格式	15	0：二进制 1：BCD
输出数据 (D) 格式	14	0：二进制 1：BCD
源数据形式	13	0：对 S 操作 1：对 X_m–S 操作
S 和 D 的带符号数据规定	11	0：无符号数据
S 和 D 的数据长度规定	10	无效(固定为16个位)
浮点规定	09	0：整数数据

- **16 位无符号二进制数据**

输入数据和 / 或输出数据可以是 16 位无符号二进制数据。另外，还可将线性插补函数设定为直接对 S 或 X_m–S 指定的值进行操作。(X_m 是线段数据中 X 的最大值。)

设定名称	C中的位	设定
输入数据 (S) 格式	15	0：二进制 1：BCD
输出数据 (D) 格式	14	0：二进制 1：BCD
源数据形式	13	0：对 S 操作 1：对 X_m–S 操作
S 和 D 的带符号数据规定	11	0：无符号数据
S 和 D 的数据长度规定	10	无效(固定为16个位)
浮点规定	09	0：整数数据

- **16 位带符号二进制数据**

设定名称	C中的位	设定
输入数据 (S) 格式	15	0：二进制
输出数据 (D) 格式	14	0：二进制
源数据形式	13	0
S 和 D 的带符号数据规定	11	1：带符号数据
S 和 D 的数据长度规定	10	0：16 位带符号二进制数据
浮点规定	09	0：整数数据

- **32 位带符号二进制数据**

设定名称	C中的位	设定
输入数据 (S) 格式	15	0：二进制
输出数据 (D) 格式	14	0：二进制
源数据形式	13	0
S 和 D 的带符号数据规定	11	1：带符号数据
S 和 D 的数据长度规定	10	1：32 位带符号二进制数据
浮点规定	09	0：整数数据

注 如果 C 的位 10 中的 “S 和 D 的数据长度规定” 设定为 1 且为 S 输入了一个 16 位常数，则输入数据在进行线性插补计算之前先转换成 32 位带符号二进制数据。

● 浮点数据

设定名称	C中的位	设定
输入数据 (S) 格式	15	0
输出数据 (D) 格式	14	0
源数据形式	13	0
S 和 D 的带符号数据规定	11	0
S 和 D 的数据长度规定	10	0
浮点规定	09	1：浮点数据

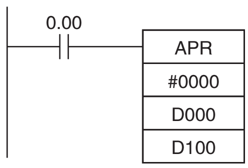
注 如果 C 的位 09 中的“浮点规定”设定为 1，则无法为 S 输入常数。

程序举例

● 正弦函数 (C：#0000)

下例所示为使用 APR(069) 指令来计算 30° 的正弦值。

(SIN(30)=0.5000)



源数据

S：D0			
0	$\times 10^{1}$	$\times 10^{0}$	$\times 10^{-1}$
0	3	0	0

将源数据的精度设定到10^{-1}度
(0000 ~ 0900，BCD)

结果

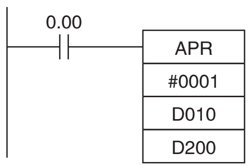
R：D100			
$\times 10^{-1}$	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-4}$
5	0	0	0

结果保留4位有效数位，
第5位及后面的数位被忽略
(0000 ~ 9999，BCD)

● 余弦函数 (C：#0001)

下例所示为使用 APR(069) 指令来计算 30° 的余弦值。

(COS(30)=0.8660)



源数据

S：D10			
0	$\times 10^{1}$	$\times 10^{0}$	$\times 10^{-1}$
0	3	0	0

将源数据的精度设定到10^{-1}度
(0000 ~ 0900，BCD)

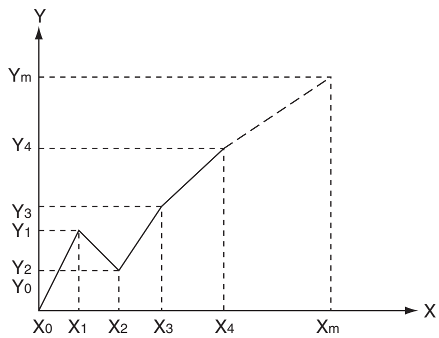
结果

R：D100			
$\times 10^{-1}$	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-4}$
8	6	6	0

结果保留4位有效数位，
第5位及后面的数位被忽略
(0000 ~ 9999，BCD)

● 线性外插 (C：字地址)

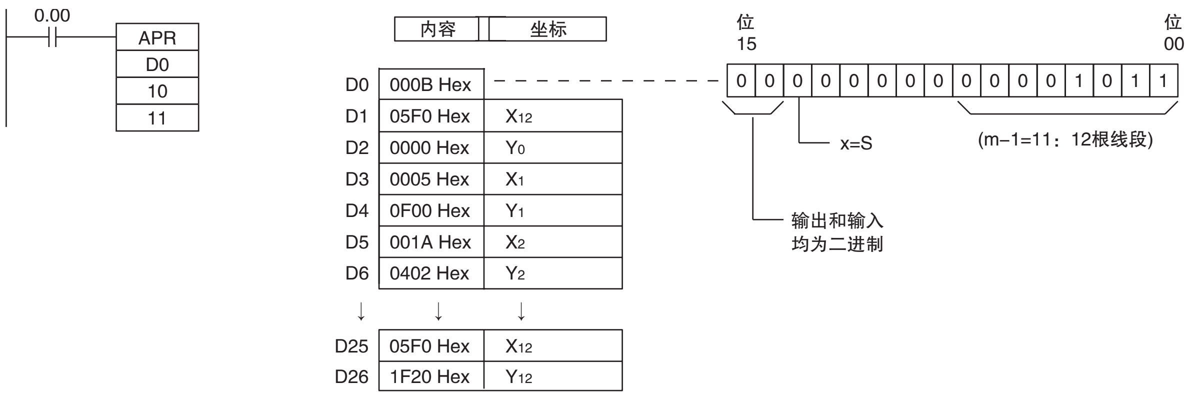
・使用16位无符号BCD或二进制数据APR(069)根据C中的控制数据来处理在S中指定的输入数据和在以C+1开始的表中指定的线段数据。结果输出到 D。



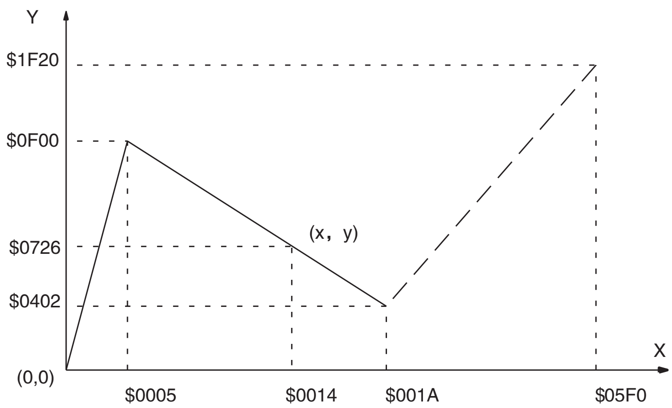
- Y_n=f(X_n)，Y_0=f(X_0)
- 务必确保所有情况下均满足 $X_{n-1}<X_n$。
- 将所有的 (X_n，Y_n) 值作为二进制数据输入。

字	坐标
C+1	Xm(最大的 X 值)
C+2	Y_0
C+3	X_1
C+4	Y_1
C+5	X_2
C+6	Y_2
↓	↓
C+(2m+1)	X_m(最大的 X 值)
C+(2m+2)	Y_m

下例所示为如何用 12 个坐标来建立一个线性外插。数据块根据要求是从 D0 ~ D26(C ~ C+(2 × 12+2)) 的连续地址。输入数据取自 CIO 10，结果则输出到 CIO 11。



在该例中，源字 CIO 0010 包含 0014，且 f(0014)=0726 输出到 R，即 CIO 0011。



线性外插的计算过程如下所示。

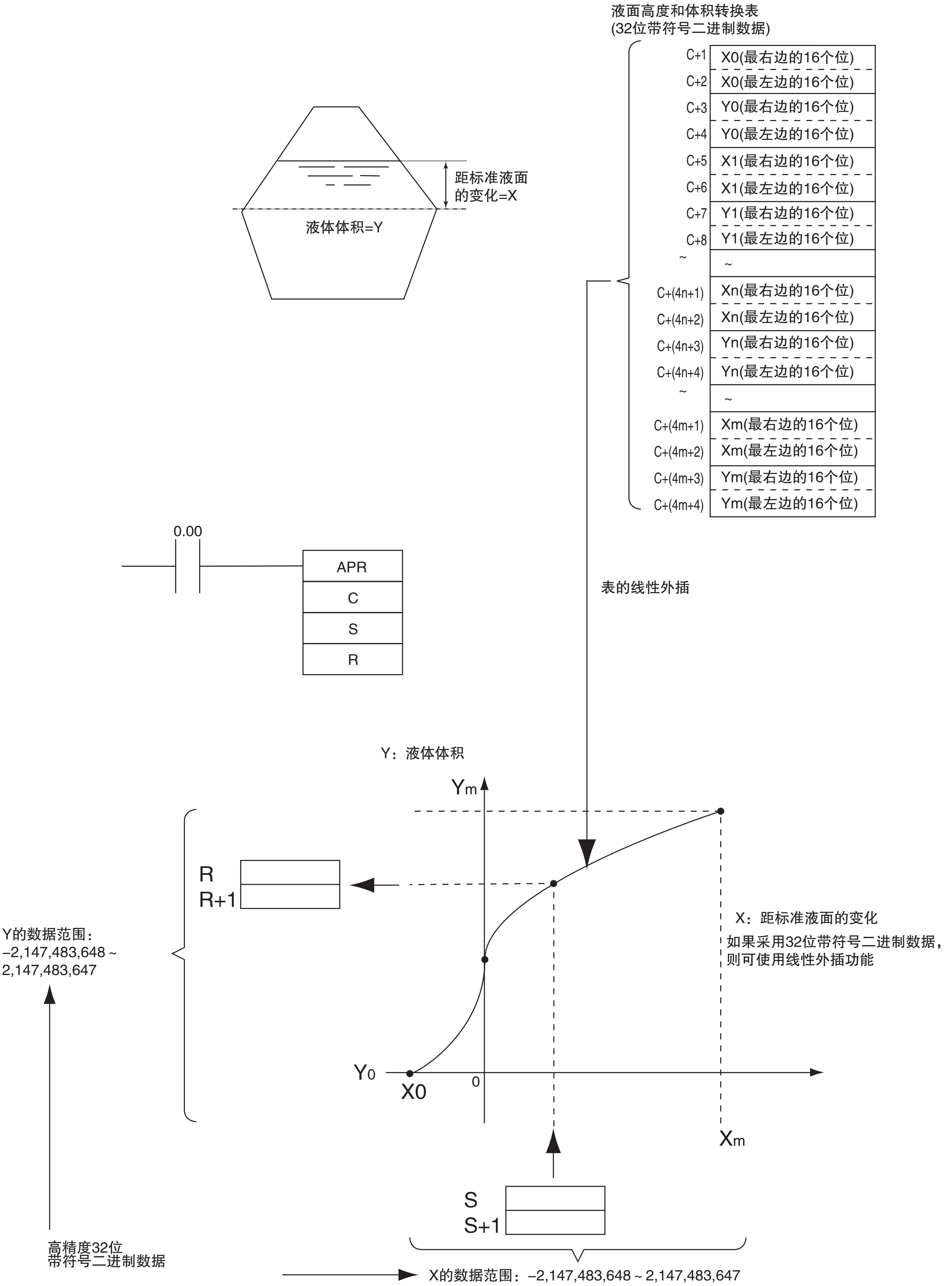
$$Y = 0F00 + \frac{0402 - 0F00}{001A - 0005} \times (0014 - 0015)$$

$$= 0F00 - (0086 \times 000F)$$

$$= 0726$$ 值全部为十六进制 (Hex)。

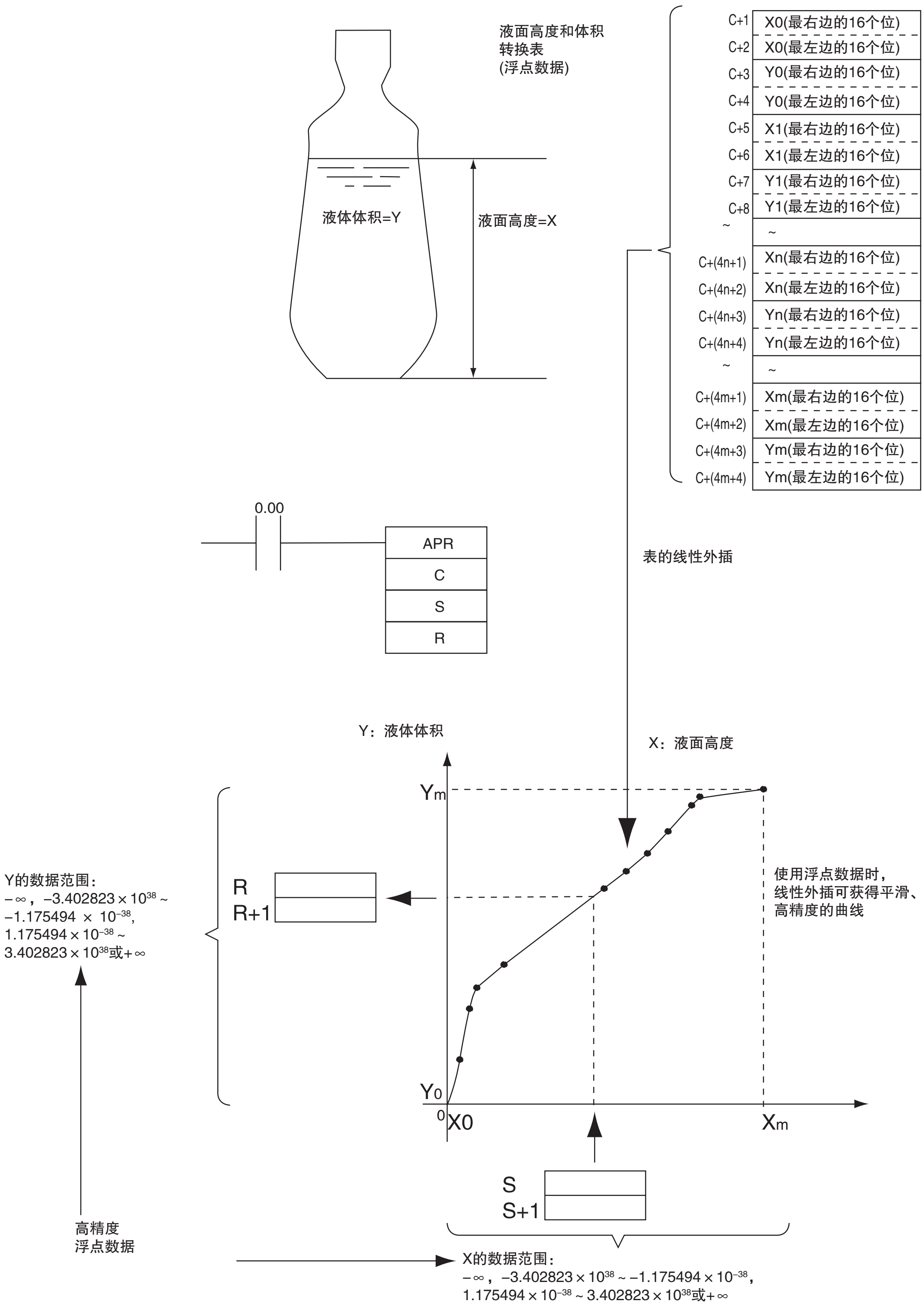
· 使用 32 位带符号二进制数据

该例中，使用 APR(069) 指令以根据容器的形状来将容器中液面的高度转换为液体的体积。



· 使用浮点数据

该例中，使用 APR(069) 指令以根据容器的形状来将容器中液面的高度转换为液体的体积。



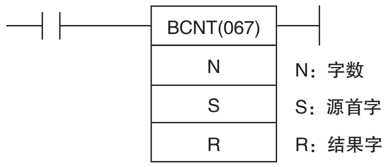
特殊算术指令

2

APR

BCNT

指令	助记符	变化	功能代码	功能
位计数器	BCNT	@BCNT	067	对指定字中的所有 ON 位的总数计数。

符号	BCNT
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	字数	UINT	1
S	源首字	UINT	可变
R	结果字	UINT	1

N：字数

字数必须介于 0001 ～ FFFF(1 ～ 65,535 个字) 之间。

● 区域类型

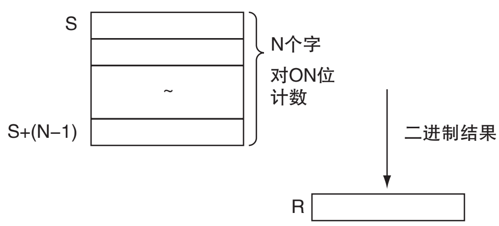
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· N 为 0000 时 ON。 · 结果超过 FFFF 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0000 时 ON。 · 其它情况下 OFF。

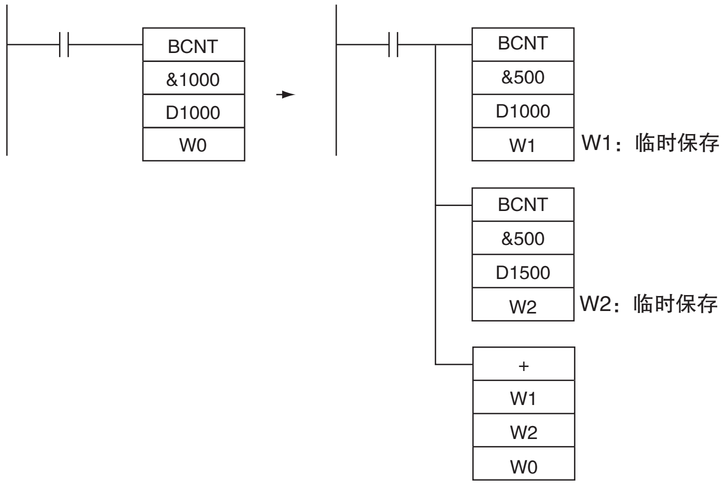
功能

BCNT(067) 计算 S ～ S+(N−1) 之间的所有字中的 ON 位总数，并将结果写入 R 中。



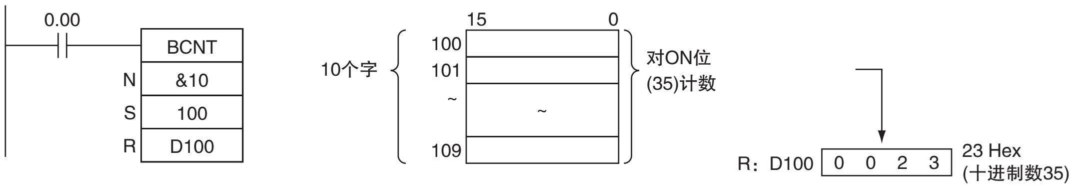
注意事项

- 指定了大量的字时，完成 BCNT(067) 指令的执行需要一定的时间。即使发生中断，该指令的执行也不会中断，且中断任务将在 BCNT(067) 指令执行完之后开始执行。 为避免该问题，可用两条 BCNT(067) 指令来替代一条 BCNT(067) 指令。



程序举例

下例中，当 CIO 0.00 为 ON 时，BCNT(067) 计算从 CIO 100 ～ CIO 109 这 10 个字中的 ON 位总数，并将结果写入 D100。



浮点算术运算指令

浮点算术运算指令用于转换数据并执行浮点算术操作。

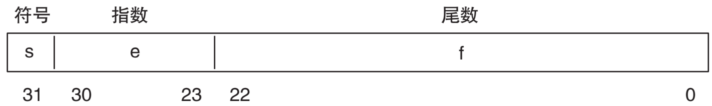
● 数据格式

浮点数据用符号、指数和尾数来表示实数。当数据以浮点格式表示时，采用以下公式。

实数 $=(-1)^{s}2^{e-127}(1.f)$

s：符号
e：指数
f：尾数

浮点数据格式符合 IEEE754 标准，数据以 32 位表示，如下所示：



数据	位数	内容
s：符号	1	0：正，1：负
e：指数	8	指数 (e) 值的范围为 0 ~ 255。指数的实际值是用 e 减去 127 得到的结果，范围为 −127 ~ 128。“e=0”和“e=255”表示特殊的数。
f：尾数	23	二进制浮点数据的尾数部分满足 2.0>1.f ≥ 1.0。

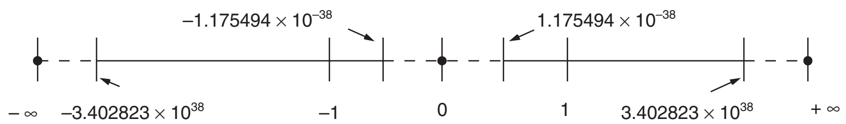
● 数位的个数

浮点数据的有效数位个数为 7 位 (十进制时)。

● 浮点数据

下列数据可用浮点数据表示：

- $-\infty$
- $-3.402823 \times 10^{38} \leq$ 值 $\leq -1.175494 \times 10^{-38}$
- 0
- $1.175494 \times 10^{-38} \leq$ 值 $\leq 3.402823 \times 10^{38}$
- $+\infty$
- 非数字 (NaN)



● 特殊数字

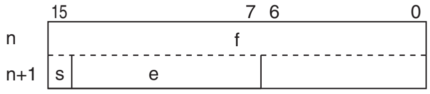
NaN、±∞ 和 0 的格式如下：

- NaN*：e=255，f ≠ 0
- +∞：e=255，f=0，s=0
- −∞：e=255，f=0，s=1
- 0： e=0，f=0

* NaN(非数字) 不是有效的浮点数字。执行浮点运算指令不会导致 NaN。

● 写入浮点数据

当在 CX-Programmer 的 I/O 存储器编辑显示器中将数据格式指定为浮点数时，在显示器中输入的标准十进制数字将自动转换成上述的浮点格式 (IEEE754 格式)，并写入 I/0 存储器中。在显示器上监控时，以 IEEE754 格式写入的数据将自动转换成标准的十进制格式。



读 / 写浮点数据时，用户无需了解 IEEE754 数据格式，只需记住每个浮点数占用两个字。

● 用浮点数表示的数字

可使用下列类型的浮点数。

尾数 (f)	指数 (e)		
	0	非 0 和非全 1	全 1(255)
0	0	规格化数	无穷大
非 0	非规格化数	规格化数	NaN

注 非规格化数是绝对值很小的数，无法用规格化数字来表示。非规格化数的有效数位很少。如果计算结果为非规格化数 (包括中间结果)，则有效数位的个数将减少。

(1) 规格化数

规格化数表示实数。符号位为 0 表示正数，符号位为 1 表示负数。

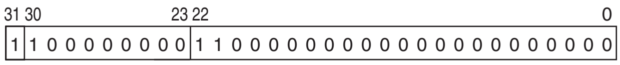
指数 (e) 用 1 ～ 254 表示，实际指数将小于 127，即 –126 ～ 127。

尾数 (f) 用 0 ～ 2^{33}–1 表示，并且假设实际尾数的位 2^{33} 为 1，且二进制小数点紧随其后。

规格化数表示如下：

$(-1)^{(\text{符号 s})} \times 2^{(\text{指数 e})-127} \times (1+\text{尾数} \times 2^{-23})$

例



符号： –

指数： 128–127=1

尾数： $1+(2^{22}+2^{21}) \times 2^{-23}=1+(2^{-1}+2^{-2})=1+0.75=1.75$

值： $-1.75 \times 2^{1}=-3.5$

(2) 非规格化数

非规格化数表示绝对值很小的实数。符号位为 0 表示正数，符号位为 1 表示负数。

指数 (e) 将为 0，而实际指数将为 −126。

尾数 (f) 用 1 ~ $2^{33}-1$ 表示，并且假设实际尾数的位 2^{33} 为 0，且二进制小数点紧随其后。

非规格化数表示如下：

$(-1)^{(符号 s)} \times 2^{-126} \times (尾数 \times 2^{-23})$

例

31	30 … 23	22 … 0
0	0 0 0 0 0 0 0 0	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

符号：　−

指数：　−126

尾数：　$0+(2^{22}+2^{21}) \times 2^{-23}=0+(2^{-1}+2^{-2})=0+0.75=0.75$

值：　-0.75×2^{-126}

(3) 零

值 +0.0 和 −0.0 可通过设置符号来表示，符号为 0 表示 +0.0，符号为 1 表示 −0.0。此时指数和尾数均为 0。+0.0 和 −0.0 均等于 0.0。由 0.0 的符号所产生的差别，请参考下述的浮点算术运算结果。

(4) 无穷大

值 +∞ 和 −∞ 可通过设置符号来表示，符号为 0 表示 +∞，符号为 1 表示 −∞。此时指数为 255(2^8-1)，尾数为 0。

(5) NaN

当计算 0.0/0.0、∞/∞ 或 ∞−∞ 等结果不等于一个数字或等于无穷大时将产生 NaN(非数字)。此时指数为 255(2^8-1)，尾数为非 0。

注 对于 NaN 的符号或尾数域的值 (不是非 0) 无规定。

● 浮点算术运算结果

(1) 对结果四舍五入

当浮点算术运算的精确结果中的位数超过内部处理表达式的有效位数时，将采用以下方法对结果进行四舍五入。

如果结果接近于两个内部浮点表达式中的某一个，则将适用接近的表达式。

如果结果介于两个内部浮点表达式中间，则将对结果四舍五入，使尾数的最后一个数位为 0。

(2) 上溢、下溢和非法计算

上溢将根据结果的符号，以正无穷或负无穷输出。下溢将根据结果的符号，以正 0 或负 0 输出。

非法计算将导致 NaN。非法计算包括使无穷大与相反符号的数相加、用相反符号的数减去无穷大、使 0 与无穷大相乘、用 0 除以 0 或者用无穷大除以无穷大。

当将浮点数转换成整数时，若产生溢出，则结果值可能会不正确。

(3) 处理特殊值时的注意事项

下列注意事项适用于处理 0、无穷大和 NaN。

- 正 0 和负 0 的和为正 0。
- 相同符号的 0 之差为正 0。
- 如果任何一个操作数为 NaN，则出错标志将被置 ON 且不执行测试。
- 正 0 和负 0 在比较时作为相等的数处理。
- 不执行对一个或一个以上 NaN 的比较或相等测试，且出错标志将被置 ON。

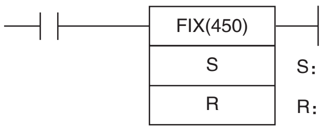
● 浮点计算结果

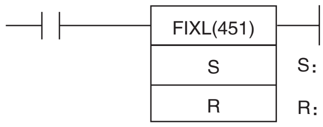
当结果的绝对值大于浮点数据所能表示的最大值时，上溢标志将变 ON，且结果将输出为 ±∞。如果结果为正，则将输出为 +∞；若为负，则将输出为 –∞。

当计算后指数 (e) 和尾数 (f) 均为 0 时，等于标志将变 ON。当结果的绝对值小于浮点数据所能表达的最小值时，计算结果也将输出为 0。此时，下溢标志将变 ON。

FIX/FIXL

指令	助记符	变化	功能代码	功能
浮点数→ 16 位	FIX	@FIX	450	将一个 32 位浮点数据转换成 16 位带符号的二进制数据，并将结果放入指定的结果字中。
浮点数→ 32 位	FIXL	@FIXL	451	将一个 32 位浮点数据转换成 32 位带符号的二进制数据，并将结果放入指定的结果字中。

<table>
<tr><th rowspan="2">符号</th><th>FIX</th><th>FIXL</th></tr>
<tr><td>

S：源首字
R：结果字</td><td>

S：源首字
R：结果首字</td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>FIX</th><th>FIXL</th><th>FIX</th><th>FIXL</th></tr>
<tr><td>S</td><td>源首字</td><td>REAL</td><td>REAL</td><td>2</td><td>2</td></tr>
<tr><td>R</td><td>结果首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th rowspan="2" colspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td rowspan="2">FIX</td><td>S</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td>---</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td><td>OK</td></tr>
<tr><td rowspan="2">FIXL</td><td>S</td><td>OK</td><td rowspan="2">---</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

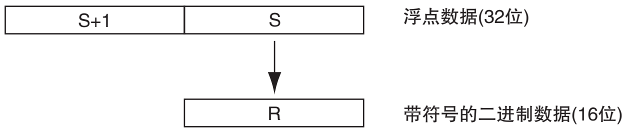
条件标志

名称	标记	操作
出错标志	P_ER	· FIX 当 S+1 和 S 的整数部分不在 –32,768 ～ 32,767 范围内时 ON。 · FIXL 当 S+1 和 S 的整数部分不在 –2,147,483,648 ～ 2,147,483,647 的范围内时 ON。 · 当 S+1 和 S 中的数据为非数字 (NaN) 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0000 时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果的最左位为 1 时 ON。 · 其它情况下 OFF。

功能

● FIX

FIX(450) 指令将 S+1 和 S 中的 32 位浮点数 (IEEE754 格式) 的整数部分转换成 16 位带符号二进制数据，并将结果放入 R 中。



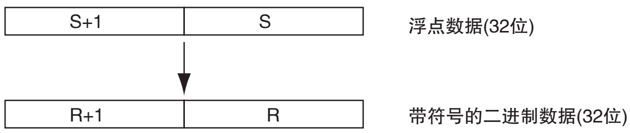
仅转换浮点数据的整数部分，而将小数部分舍去。

转换举例：

浮点值 3.5 将转换成 3。
浮点值 –3.5 将转换成 –3。

● FIXL

FIXL(451) 指令将 S+1 和 S 中的 32 位浮点数 (IEEE754 格式) 的整数部分转换成 32 位带符号二进制数据，并将结果放入 R+1 和 R 中。



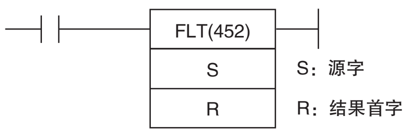
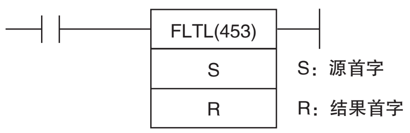
仅转换浮点数据的整数部分，而将小数部分舍去。

转换举例：

浮点值 2,147,483,640.5 将转换成 2,147,483,640。
浮点值 –214,748,340.5 将转换成 –214,748,340。

FLT/FLTL

指令	助记符	变化	功能代码	功能
16 位→浮点数	FLT	@FLT	452	将一个 16 位带符号的二进制数据转换成 32 位浮点数据，并将结果放入指定的结果字中。
32 位→浮点数	FLTL	@FLTL	453	将一个 32 位带符号的二进制数据转换成 32 位浮点数据，并将结果放入指定的结果字中。

<table>
<tr><th rowspan="2">符号</th><th>FLT</th><th>FLTL</th></tr>
<tr><td></td><td></td></tr>
</table>

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

<table>
<tr><th rowspan="2">操作数</th><th rowspan="2">描述</th><th colspan="2">数据类型</th><th colspan="2">大小</th></tr>
<tr><th>FLT</th><th>FLTL</th><th>FLT</th><th>FLTL</th></tr>
<tr><td>S</td><td>源首字</td><td>INT</td><td>DINT</td><td>1</td><td>2</td></tr>
<tr><td>R</td><td>结果首字</td><td>REAL</td><td>REAL</td><td>2</td><td>2</td></tr>
</table>

● 区域类型

<table>
<tr><th rowspan="2" colspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR
直接</th><th>IR
间接</th></tr>
<tr><td rowspan="2">FLT</td><td>S</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td rowspan="4">OK</td><td>OK</td><td>OK</td><td rowspan="4">---</td><td rowspan="4">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>R</td><td>---</td><td rowspan="3">---</td></tr>
<tr><td rowspan="2">FLTL</td><td>S</td><td>OK</td></tr>
<tr><td>R</td><td>---</td></tr>
</table>

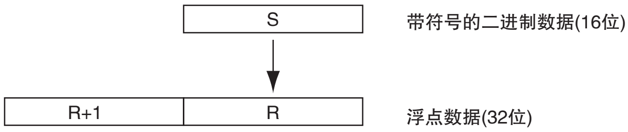
条件标志

名称	标记	操作
出错标志	P_ER	OFF
等于标志	P_EQ	・当结果的指数和尾数均为 0 时 ON。 ・其它情况下 OFF。
负标志	P_N	・结果为负时 ON。 ・其它情况下 OFF。

功能

● FLT

FLT(452) 指令将 S 中的 16 位带符号二进制值转换成 32 位浮点数据 (IEEE754 格式)，并将结果放入 R+1 和 R 中。在浮动数结果的小数点后将添加一位 0。



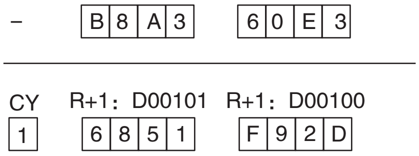
只能为S指定 –32,768 ~ 32,767的值。若要转换上述范围之外的带符号二进制数据，请使用FLTL(453) 指令。

转换举例：

带符号二进制值 3 将转换成 3.0。
带符号二进制值 –3 将转换成 –3.0。

● FLTL

FLTL(453) 指令将 S+1 和 S 中的 32 位带符号二进制值转换成 32 位浮点数据 (IEEE754 格式)，并将结果放入 R+1 和 R 中。在浮动数结果的小数点后将添加一位 0。



只能为 S+1 和 S 指定 –2,147,483,648 ~ 2,147,483,647 的带符号二进制数据。浮点值有 24 个有效数位 (位)。如果通过 FLTL(453) 来转换大于 16,777,215 的数 (可用 24 位表示的最大值)，则结果将不精确。

转换举例：

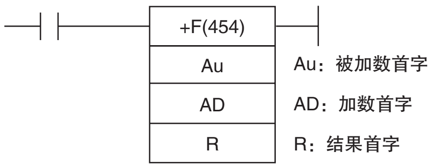
带符号二进制值 16,777,215 将转换成 16,777,215.0。
带符号二进制值 –16,777,215 将转换成 –16,777,215.0。

+F，–F，*F，/F

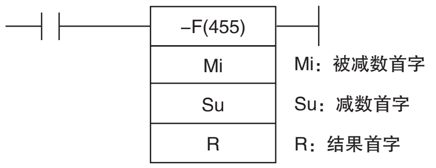
指令	助记符	变化	功能代码	功能
浮点数加	+F	@+F	454	将两个 32 位浮点数相加，并将结果放入指定的结果字中。
浮点数减	–F	@–F	455	将两个 32 位浮点数相减，并将结果放入指定的结果字中。
浮点数乘	*F	@*F	456	将两个 32 位浮点数相乘，并将结果放入指定的结果字中。
浮点数除	/F	@/F	457	将两个 32 位浮点数相除，并将结果放入指定的结果字中。

符号

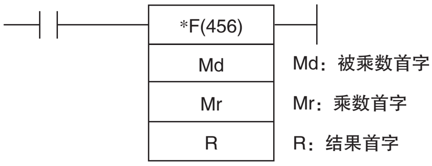
+F



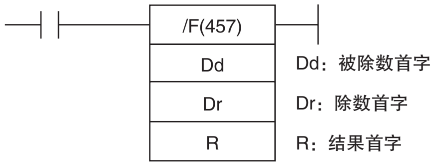
–F



*F



/F



适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数		描述	数据类型	大小
+F	Au	被加数首字	REAL	2
	AD	加数首字		
–F	Mi	被减数首字	REAL	2
	Su	减数首字		
*F	Md	被乘数首字	REAL	2
	Mr	乘数首字		
/F	Dd	被除数首字	REAL	2
	Dr	除数首字		
R		结果首字	REAL	2

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
Au，AD，Mi，Su，Md，Mr，Dd，Dr	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

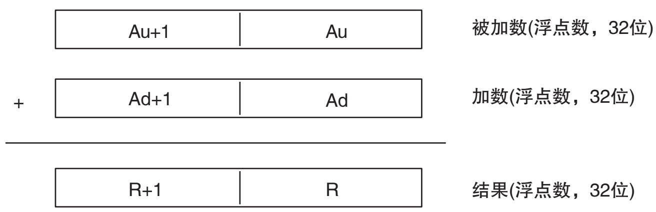
条件标志

名称	标记	操作
出错标志	P_ER	+F · 当被加数或加数为非数字 (NaN) 时 ON。 · 当将 +∞ 和 –∞ 相加时 ON。 –F · 当被减数或减数为非数字 (NaN) 时 ON。 · 当将 +∞ 减去 +∞ 时 ON。 · 当将 –∞ 减去 –∞ 时 ON。 *F · 当被乘数或乘数为非数字 (NaN) 时 ON。 · 当将 +∞ 和 0 相乘时 ON。 · 当将 –∞ 和 0 相乘时 ON。 /F · 当被除数或除数为非数字 (NaN) 时 ON。 · 当被除数和除数均为 0 时 ON。 · 当被除数和除数均为 +∞ 或 –∞ 时 ON。 其它情况下 OFF。
等于标志	P_EQ	· 当结果的指数和尾数均为 0 时 ON。 · 其它情况下 OFF。
上溢标志	P_OF	· 当结果的绝对值太大、无法用 32 位浮点数来表示时 ON。 · 其它情况下 OFF。
下溢标志	P_UF	· 当结果的绝对值太小、无法用 32 位浮点数来表示时 ON。 · 其它情况下 OFF。
负标志	P_N	· 结果为负时 ON。 · 其它情况下 OFF。

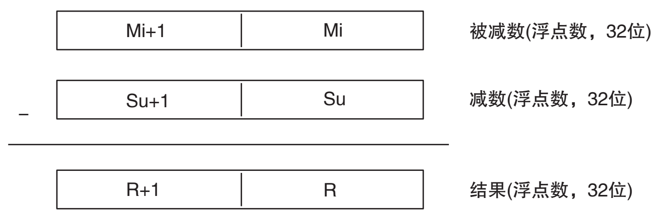
功能

将在 Au/Mi/Md/Dd 中指定的数据和在 AD/Su/Mr/Dr 中指定的数据作为单精度浮点数据 (32 位: IEEE754 格式) 相加 (+F)、相减 (–F)、相乘 (*F) 或相除 (/F)，并将结果输出到 R+1，R 中。

● +F



● –F



● *F

	Md+1	Md	被乘数(浮点数，32位)
×	Mr+1	Mr	乘数(浮点数，32位)
	R+1	R	结果(浮点数，32位)

● /F

	Dd+1	Dd	被除数(浮点数，32位)
÷	Dr+1	Dr	除数(浮点数，32位)
	R+1	R	结果(浮点数，32位)

・如果结果的绝对值大于浮点数据所能表示的最大值，上溢标志将变 ON，且结果将输出为 ±∞。

・如果结果的绝对值小于浮点数据所能表示的最小值，下溢标志将变 ON，且结果将输出为 0。

● 运算规则

根据浮点数的组合情况而定，输出的运算结果如下所示。

● 浮点数加 (+F)

	被加数				
加数	0	数字	+∞	−∞	NaN
0	0	数字	+∞	−∞	
数字	数字	见注 1	+∞	−∞	
+∞	+∞	+∞	+∞	ER	
−∞	−∞	−∞	ER	−∞	
NaN					ER

注 1 结果可能是 0(包括下溢)、数字、+∞ 或 −∞。

ER 出错标志将变 ON，且不执行指令。

● 浮点数减 (−F)

	被减数				
减数	0	数字	+∞	−∞	NaN
0	0	数字	+∞	−∞	
数字	数字	见注 1	+∞	−∞	
+∞	−∞	−∞	ER	−∞	
−∞	+∞	+∞	+∞	ER	
NaN					ER

注 1 结果可能是 0(包括下溢)、数字、+∞ 或 −∞。

ER 出错标志将变 ON，且不执行指令。

● 浮点数乘 (*F)

	被乘数				
乘数	0	数字	+∞	–∞	NaN
0	0	0	ER	ER	
数字	0	见注 1	+/–∞	+/–∞	
+∞	ER	+/–∞	+∞	–∞	
–∞	ER	+/–∞	–∞	+∞	
NaN					ER

注 1　结果可能是 0(包括下溢)、数字、+∞ 或 –∞。

ER 出错标志将变 ON，且不执行指令。

● 浮点数除 (/F)

	被除数				
除数	0	数字	+∞	–∞	NaN
0	ER	+/–∞	+∞	–∞	
数字	0	见注 2	+/–∞	+/–∞	
+∞	0	0(见注 1)	ER	ER	
–∞	0	0(见注 1)	ER	ER	
NaN					ER

注 1　下溢时结果将为 0。
　 2　结果可能是 0(包括下溢)、数字、+∞ 或 –∞。

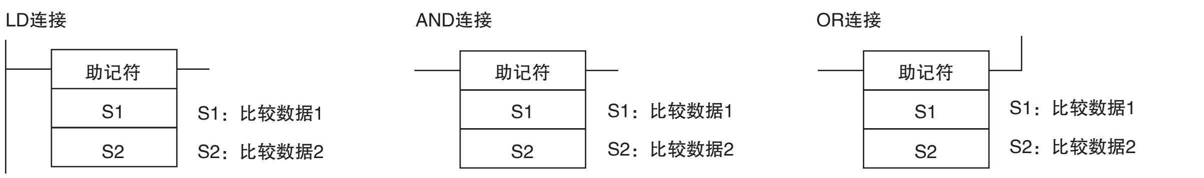
ER 出错标志将变 ON，且不执行指令。

=F，<>F，<F，<=F，>F，>=F

指令	助记符	变化	功能代码	功能
单精度浮点比较	=F <>F <F <=F >F >=F	---	329 330 331 332 333 334	这些输入比较指令比较两个单精度浮点值 (32 位 IEEE754 常数和 / 或指定字的内容)，并在比较条件为真时产生一个 ON 执行条件。

符号

单精度浮点比较



适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S1	比较数据 1	REAL	2
S2	比较数据 2	REAL	2

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S1，S2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当 S1+1，S1 或 S2+1，S2 为非数字 (NaN) 时 ON。 · 当 S1+1，S1 或 S2+1，S2 为 +∞ 时 ON。 · 当 S1+1，S1 或 S2+1，S2 为 –∞ 时 ON。 · 其它情况下 OFF。
大于标志	P_GT	· 当 S1+1，S1 > S2+1，S2 时 ON。 · 其它情况下 OFF。
大于等于标志	P_GE	· 当 S1+1，S1 ⩾ S2+1，S2 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 当 S1+1，S1 = S2+1，S2 时 ON。 · 其它情况下 OFF。
不等于标志	P_NE	· 当 S1+1，S1 ≠ S2+1，S2 时 ON。 · 其它情况下 OFF。
小于标志	P_LT	· 当 S1+1，S1 < S2+1，S2 时 ON。 · 其它情况下 OFF。
小于等于标志	P_LE	· 当 S1+1，S1 ⩽ S2+1，S2 时 ON。 · 其它情况下 OFF。
负标志	P_N	不变

功能

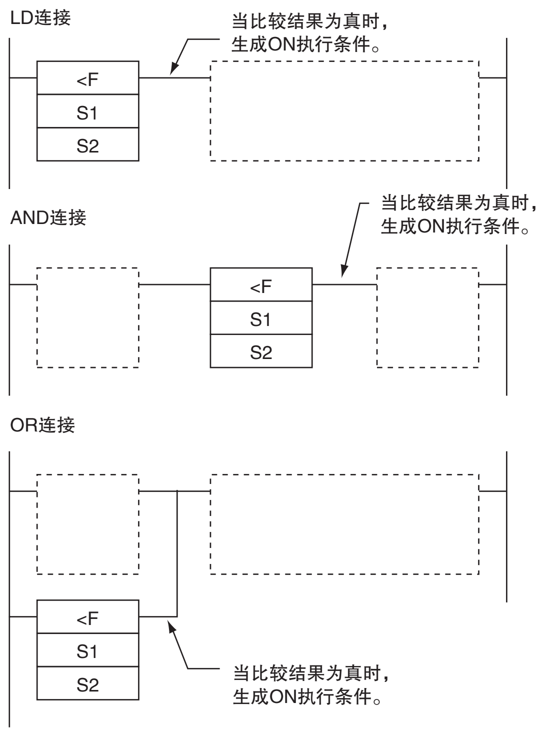
输入比较指令将在S1和S2中的指定的数据作为单精度浮点数 (32 位 IEEE754数据) 进行比较，并在比较条件为真时生成一个ON执行条件。当数据以字为单位存储时，S1 和 S2 指定包含 32 位数据的两个字中的第一个字。另外也可将浮点数据作为8位数的十六进制常数输入。

对输入比较指令的处理类似于 LD、AND和OR指令，用于控制随后的指令执行。

LD: 该指令可直接连接到左边的母线。

AND: 该指令不能直接连接到左边的母线。

OR: 该指令可直接连接到左边的母线。



● 选项

这些指令有 3 种输入方式和 6 个符号，可形成 18 种不同的组合。

符号 (LD、AND 和 OR 不能用于梯形图程序中)		选项 (数据格式)
LD=，AND=，OR=，LD<>，AND<>，OR<>，LD<，AND<，OR<，LD<=，AND<=，OR<=，LD>，AND>，OR>，LD>=，AND>=，OR>=	+	F：单精度浮点数据

代码	助记符	名称	功能
329	LD=F	载入浮点数不等于	当 S_1+1，S_1 = S_2+1，S_2 时为真
	AND=F	与浮点数等于	
	OR=F	或浮点数等于	
330	LD<>F	载入浮点数不等于	当 S_1+1，S_1 ≠ S_2+1，S_2 时为真
	AND<>F	与浮点数不等于	
	OR<>F	或浮点数不等于	
331	LD<F	载入浮点数小于	当 S_1+1，S_1 < S_2+1，S_2 时为真
	AND<F	与浮点数小于	
	OR<F	或浮点数小于	
332	LD<=F	载入浮点数小于等于	当 S_1+1，S_1 ≤ S_2+1，S_2 时为真
	AND<=F	与浮点数小于等于	
	OR<=F	或浮点数小于等于	
333	LD>F	载入浮点数大于	当 S_1+1，S_1 > S_2+1，S_2 时为真
	AND>F	与浮点数大于	
	OR>F	或浮点数大于	

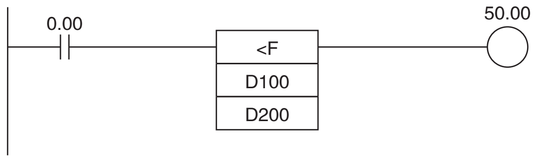
代码	助记符	名称	功能
334	LD>=F	载入浮点数大于等于	当 S_1+1，S_1 ≥ S_2+1，S_2 时为真
	AND>=F	与浮点数大于等于	
	OR>=F	或浮点数大于等于	

注意事项

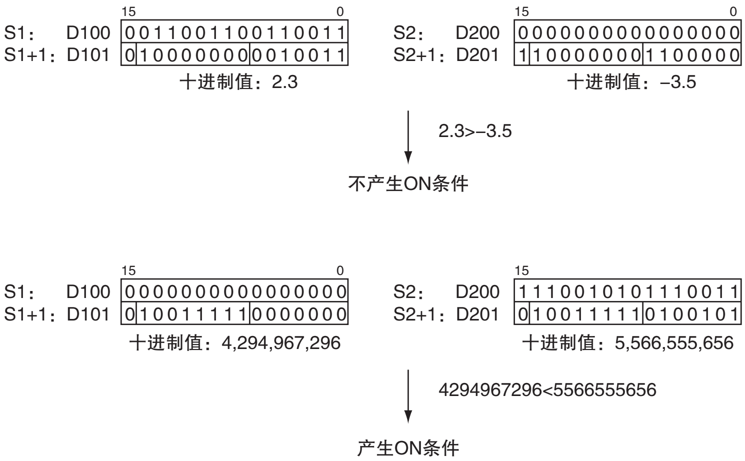
・输入指令不能作为右侧指令使用，也就是说这些指令和右侧母线之间必须要有其它指令。

程序举例

下例中，当 CIO 0.00 为 ON 时，将 D101，D100 中的浮点数据与 D201，D200 中的浮点数据进行比较。如果 D101，D100 中的内容小于 D201，D200 中的内容，则执行进到下一行，且 CIO 50.00 将变 ON。如果 D101，D100 中的内容不小于 D201，D200 中的内容，则执行不进到下一行。

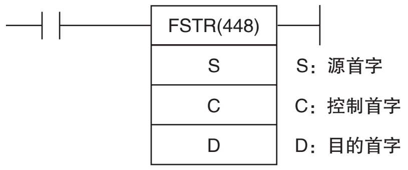


单精度浮点数小于比较 (<F)



FSTR

指令	助记符	变化	功能代码	功能
浮点数→ ASCII	FSTR	@FSTR	448	用标准十进制计数法或科学计数法来表示 32 浮点数 (IEEE754 格式)，并将该数值转换成 ASCII 文本。

符号	FSTR
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源首字	REAL	2
C	控制首字	UINT	3
D	目的首字	UINT	可变

C：控制首字

总字符	─	0 Hex：十进制格式 1 Hex：科学计数法
数据格式	─	2 ～ 18 Hex(2 ～ 24 个字符，见注)
小数位	─	0 ～ 7 Hex(见注)

注 总字符数和小数位数有限制。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
C，D										---						

条件标志

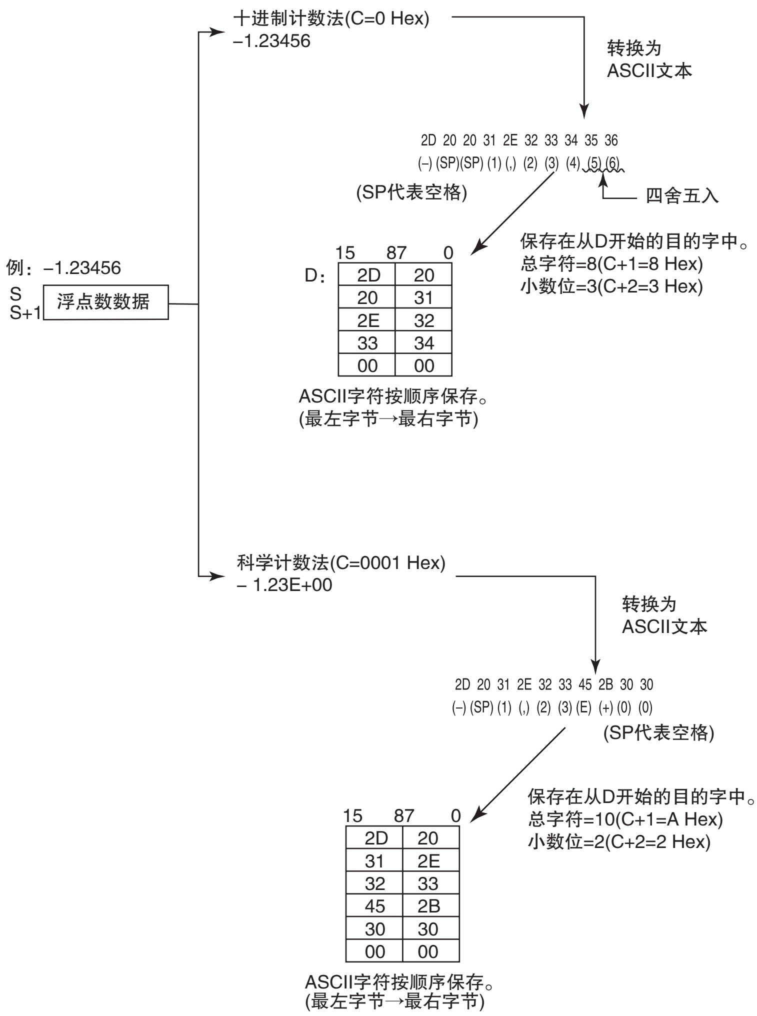
名称	标记	操作
出错标志	P_ER	· 当 S+1 和 S 中的数据不是一个合法的浮点数 (NaN) 时 ON。 · S+1 和 S 中的数据是 +0 或 –0 时 ON。 · 当 C 中的数据格式设定不是 0000 或 0001 时 ON。 · 当C+1中的总字符设定不在允许的范围内时ON。(详情请参阅上述的见注1.ASCII 字符的总数限制。) · 当 C+2 中的小数位设定不在允许的范围内时 ON。(详情请参阅上述的见注 3. 小数部分的位数限制。)
等于标志	P_EQ	· 当转换结果为 0 时 ON。 · 其它情况下 OFF。

功能

FSTR(448) 指令根据字 C ～ C+2 中的控制数据，将 S+1 和 S 中的 32 位浮点数 (IEEE754 格式) 用十进制计数法或科学计数法表示，将该数据转换成 ASCII 文本，并将结果输出到从 D 开始的目的字中。

- C 中的内容 (数据格式) 指定是以十进制计数法还是科学计数法表示 S+1，S 中的数。
  - 十进制计数法
    用整数和小数部分来表示一个实数。
    例：124.56
  - 科学计数法
    用整数部分、小数部分和指数部分来表示一个实数。
    例：1.2456E−2(1.2456 × 10^{-2})
- C+1 中的内容 (总字符) 指定转换后的 ASCII 字符数，包括符号、数字、小数点和空格。
- C+2 中的内容 (小数位) 指定小数点后的位 (字符) 数。

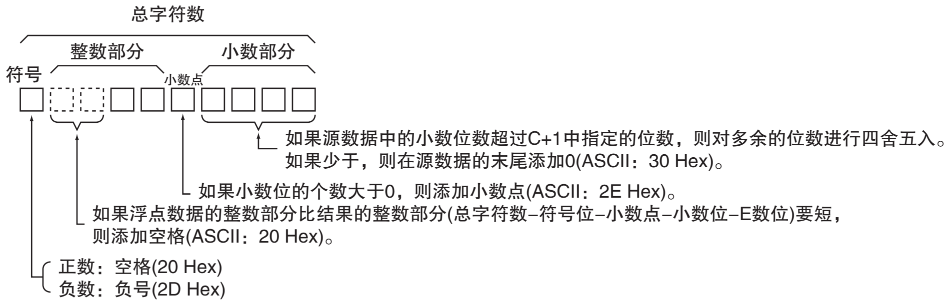
ASCII 文本按下述顺序存储在 D 及其后面的字中：D 的最左字节、D 的最右字节、D+1 的最左字节、D+1 的最右字节等。



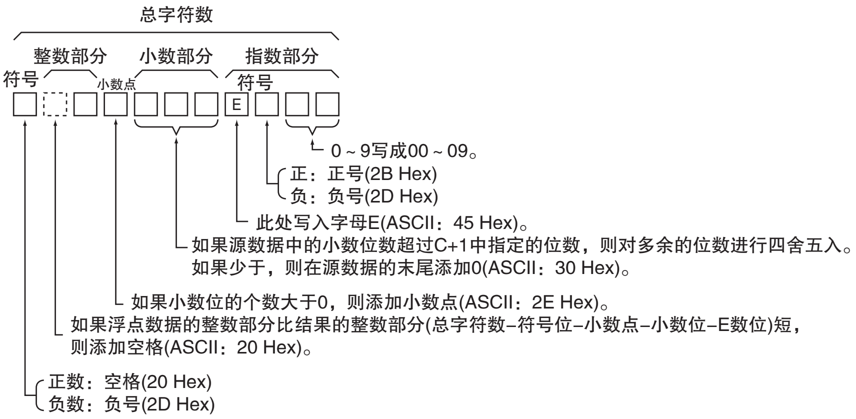
● ASCII 文本的存储

将浮点数转换成 ASCII 文本之后，将 ASCII 字符存储到从 D 开始的目的字中，如下图所示。十进制计数法和科学计数法分别采用不同的存储方法。

十进制计数法 (C=0 Hex)



科学计数法 (C=1 Hex)



注 在 ASCII 文本的末尾添加 1 到 2 个字节的 0 作为结束码。

- 总字符数为奇数：在 ASCII 文本之后放入 00 Hex。
- 总字符数为偶数：在 ASCII 文本之后放入 0000 Hex。

● ASCII 字符数的限制

对于转换后的数中的 ASCII 字符个数有限制。如果字符数超过允许的最大值，则出错标志将变 ON。

- ASCII 字符总数的限制

  1) 十进制计数法 (C=0 Hex)

  - 没有小数部分时 (C+2=0 Hex)：
    2 ≤总字符数≤ 24
  - 有小数部分时 (C+2=1 ~ 7 Hex)：
    (小数位数 +3) ≤总字符数≤ 24

  2) 科学计数法 (C=1 Hex)

  - 没有小数部分时 (C+2=0 Hex)：
    6 ≤总字符数≤ 24
  - 有小数部分时 (C+2=1 ~ 7 Hex)：
    (小数位数 +7) ≤总字符数≤ 24

- 整数部分中的位数限制

  1) 十进制计数法 (C=0 Hex)

  - 没有小数部分时 (C+2=0 Hex)：
    1 ≤整数位数≤ 24−1
  - 有小数部分时 (C+2=1 ～ 7 Hex)：
    1 ≤整数位数≤ (24− 小数位数 −2)

  2) 科学计数法 (C=1 Hex)
  1 位 (固定)

- 小数部分中的位数限制

  1) 十进制计数法 (C=0 Hex)

  - 小数位数≤ 7
  - 另外：小数位数≤ (ASCII 字符总数 −3)

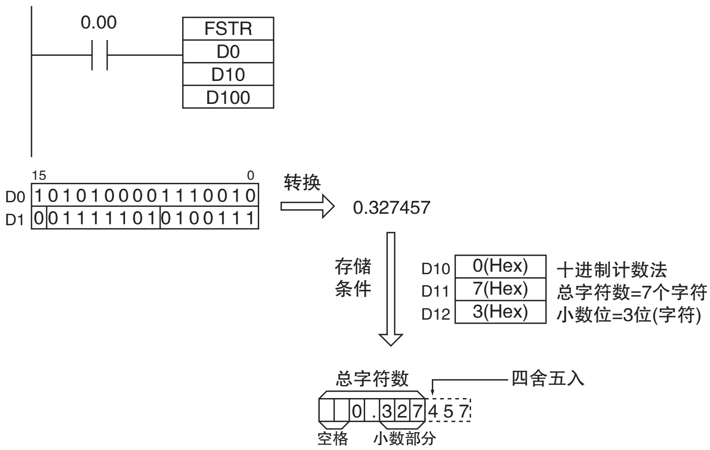
  2) 科学计数法 (C=1 Hex)

  - 小数位数≤ 7
  - 另外：小数位数≤ (ASCII 字符总数 −7)

程序举例

● 转换成用十进制计数法表示的 ASCII 文本

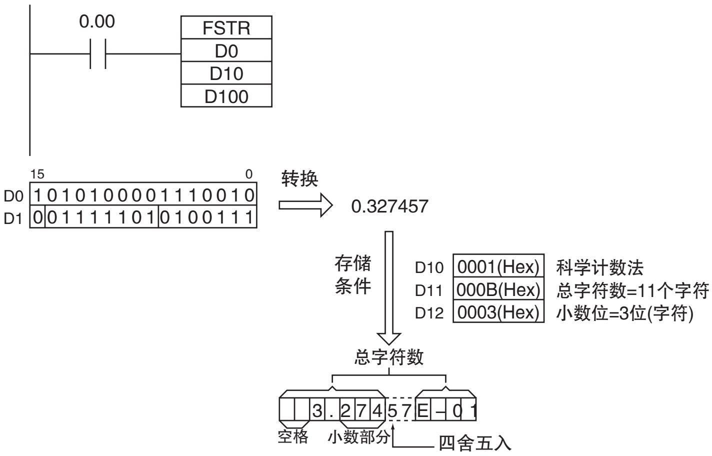
下例中，当 CIO 0.00 为 ON 时，FSTR(448) 指令将 D1 和 D0 中的浮点数据转换成用十进制计数法表示的 ASCII 文本，并将 ASCII 文本写入从 D100 开始的目的字中。控制字 (D10 ～ D12) 的内容指定有关数据格式的细节 (十进制计数法、总共 7 个字符、3 个小数位)。



D100	20(空格)	20(空格)
D101	30(0)	2E(.)
D102	33(3)	32(2)
D103	37(7)	00

● 转换成用科学计数法表示的 ASCII 文本

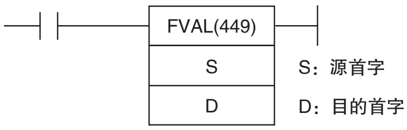
下例中，当 CIO 0.00 为 ON 时，FSTR(448) 指令将 D1 和 D0 中的浮点数据转换成用科学计数法表示的 ASCII 文本，并将 ASCII 文本写入从 D100 开始的目的字中。控制字 (D10 ~ D12) 的内容指定有关数据格式的细节 (科学计数法、总共 11 个字符、3 个小数位)。



D100	20(空格)	20(空格)
D101	33(3)	2E(.)
D102	32(2)	37(7)
D103	35(5)	45(E)
D104	2D(–)	30(0)
D105	31(1)	00

FVAL

指令	助记符	变化	功能代码	功能
ASCII →浮点数	FVAL	@FVAL	449	将用 ASCII 文本 (十进制或科学计数法) 表示的数转换成 32 位浮点值 (IEEE754 格式)，并将该浮点值输出到指定的字中。

	FVAL
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源首字	UINT	可变
D	目的首字	REAL	2

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S，D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当从 S 开始的源数据中的数位 (整数和小数部分) 不是 30 ～ 39 Hex(0 ～ 9) 时 ON。 · 当指数部分的前两位不包含 45 和 2B Hex(E+) 或 45 和 2D Hex(E–)(从 S 开始的源数据不是 30 ～ 39 Hex(0 ～ 9)) 时 ON。 · 当源数据中有 2 个或 2 个以上的指数部分时 ON。 · 当转换后的数据为 +∞ 或 –∞ 为 ON。 · 当文本数据中有 0 个字符时 ON。 · 当在前 25 个字符中未找到包含 00 Hex 的字节时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 当转换结果为 0 时 ON。 · 其它情况下 OFF。

功能

FVAL(449) 指令将指定的 ASCII 文本数据 (从字 S 开始) 转换成 32 位浮点数据 (IEEE754 格式)，并将结果输出到从 D 开始的目的字中。

满足下列条件时，FVAL(449) 指令可转换以十进制或科学计数法表示的 ASCII 文本：

最多 6 个有效字符，其中不包括符号、小数点和指数。超过第 6 位的任何字符均被忽略。

· 十进制计数法
  用整数和小数部分来表示一个实数。
  例：124.56

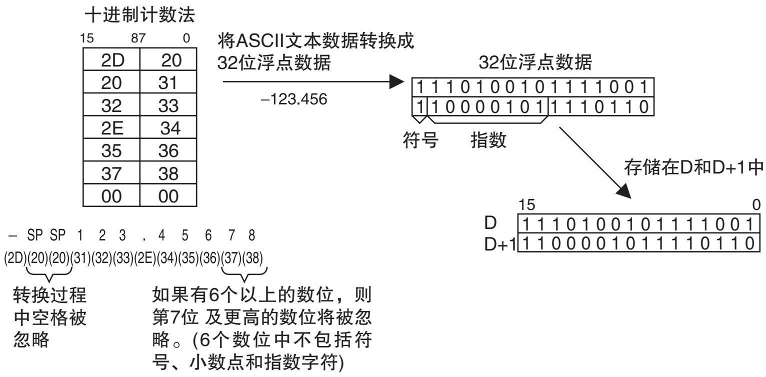
・科学计数法

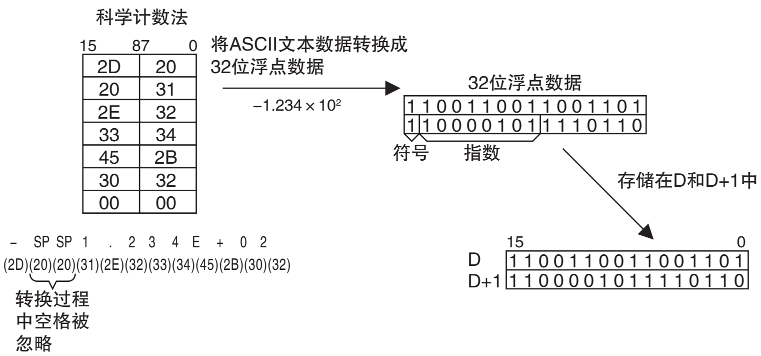
用整数部分、小数部分和指数部分来表示一个实数。

例：1.2456E−2(1.2456×10^{-2})

将自动检测数据格式 (十进制或科学计数法)。

ASCII 文本必须按下述顺序存储在 S 及其后面的字中：S 的最左字节、S 的最右字节、S+1 的最左字节、S+1 的最右字节等。

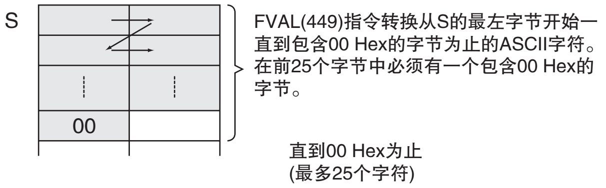




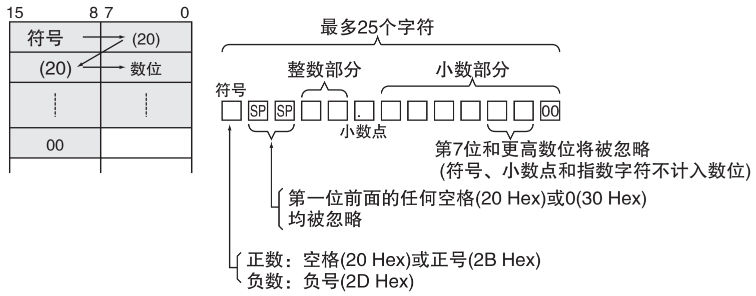
● ASCII 文本的存储

下图所示为将 ASCII 文本数据转换成浮点数据的方法。以十进制计数法和科学计数法存储的数据分别采用不同的转换方法。

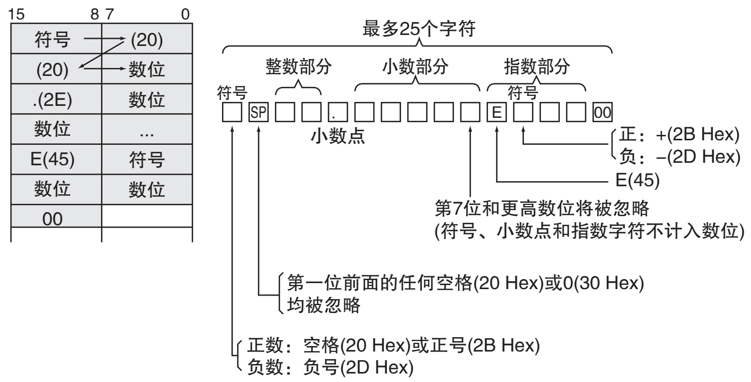
ASCII 字符的存储



十进制计数法



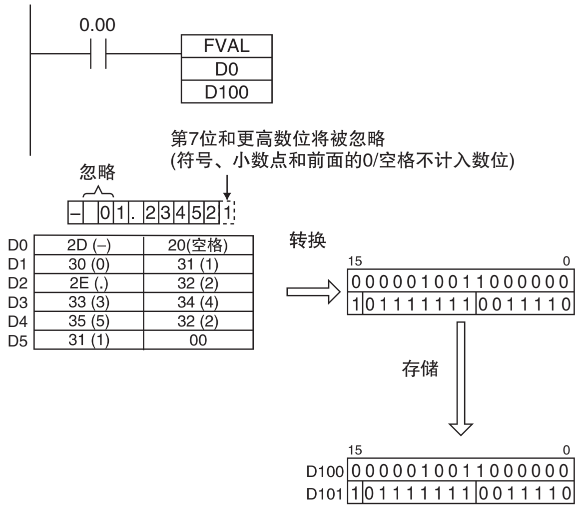
科学计数法



程序举例

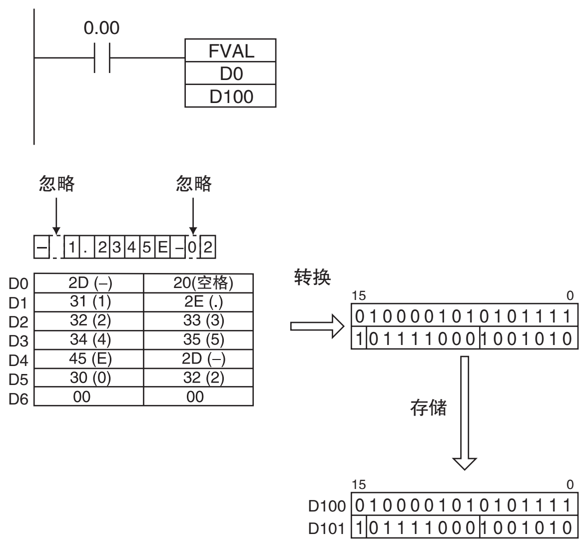
● 将用十进制计数法表示的 ASCII 文本转换成浮点数据

下例中，当 CIO 0.00 为 ON 时，FVAL(449) 指令将从 D0 开始的源数据中指定的用十进制计数法表示的 ASCII 文本转换成浮点数据，并将结果写入目的字 D100 和 D101 中。



● 转换用科学计数法表示的 ASCII 文本

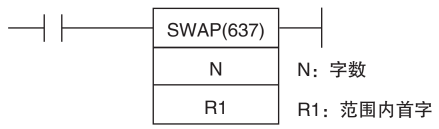
下例中，当 CIO 0.00 为 ON 时，FVAL(449) 指令将从 D0 开始的源数据中指定的用科学计数法表示的 ASCII 文本转换成浮点数据，并将结果写入目的字 D100 和 D101 中。



表数据处理指令

SWAP

指令	助记符	变化	功能代码	功能
交换字节	SWAP	@SWAP	637	将范围内所有字的最左字节和最右字节交换。

符号	SWAP
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

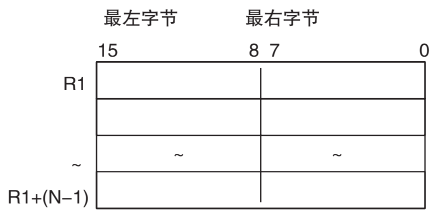
操作数

操作数	描述	数据类型	大小
N	字数	UINT	1
R1	范围内首字	UINT	可变

N：字数

N 指定范围内的字数，字数必须为十六进制的 0001 ～ FFFF(即 &1 ～ &65,535)。

R1：范围内首字



注 R1 和 R1+(N−1) 必须位于同一个数据区。

● 区域类型

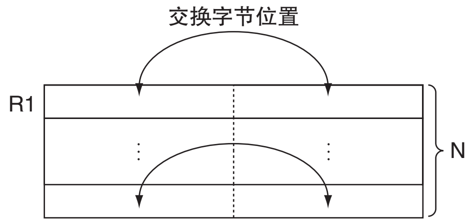
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
R1	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· N 为 0 时 ON。 · 其它情况下 OFF。

功能

SWAP(637) 指令使存储器 R1 ～ R1+N−1 范围内的所有字中的两个字节交换位置。

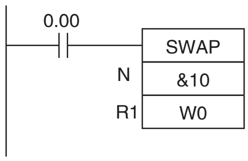


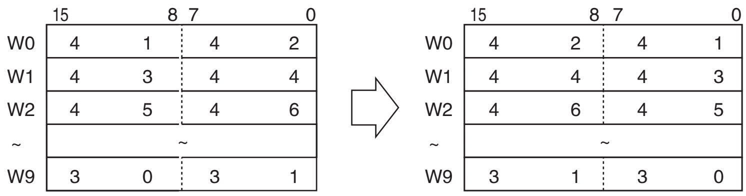
提示

· 该指令可用于使各个字中的 ASCII 码字符反序。

程序举例

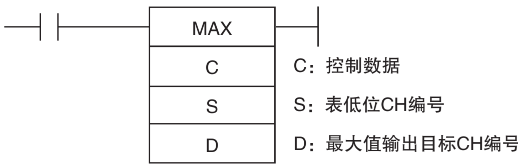
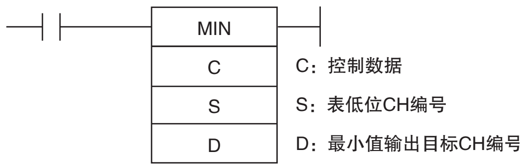
下例中，当 CIO 0.00 为 ON 时，SWAP(637) 指令使 W0 ～ W9 的 10 个字范围内的每个字的左字节中的数据和右字节中的数据交换位置。





MAX/MIN

指令	助记符	变化	功能代码	功能
最大值检索	MAX	@MAX	182	检索指定表内的最大值。
最小值检索	MIN	@MIN	183	检索指定表内的最小值。

	MAX	MIN
符号		

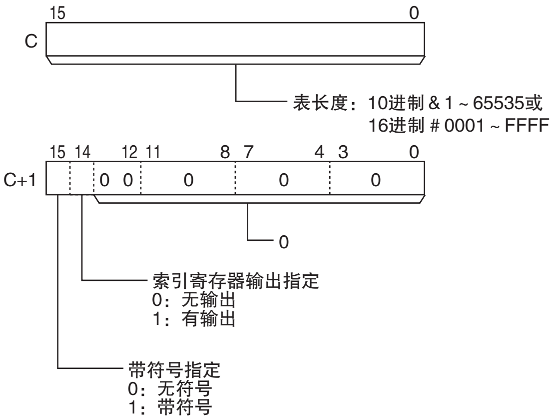
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

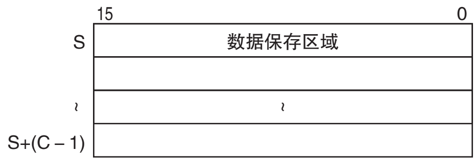
操作数

操作数	描述	数据类型	大小
C	控制数据	UDINT	2
S	表低位 CH 编号	UINT	可变
D	MAX：最大值输出目标 CH 编号 MIN：最小值输出目标 CH 编号	UINT	1

C：控制数据



S：表低位 CH 编号



注 C ～ C+1 和 S ～ S+（C － 1）各自的区域种类必须相同。

● 区域种类

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	内容	
		MAX	MIN
出错标志	P_ER	・C 的表长度数据（位 00 ～ 15）不在 10 进制 &1 ～ 65535 或 16 进制 #0001 ～ FFFF 范围内时，ON ・除此以外时，OFF	
等于标志	P_EQ	・最大值为 0 时，ON ・除此以外时，OFF	・最小值为 0 时，ON ・除此以外时，OFF
负标志	P_N	・最大值的最高位为 1 时，ON ・除此以外时，OFF	・最小值的最高位为 1 时，ON ・除此以外时，OFF

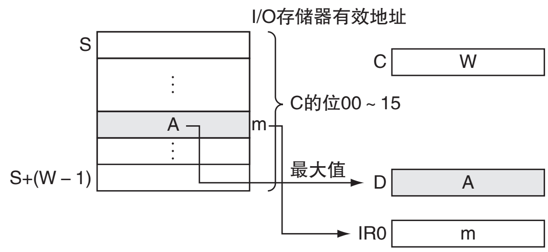
功能

● MAX

以由 S 指定的表低位 CH 编号开始的由 C 指定的表长度（字数）为表数据，从中检索最大值，输出至 D。

索引寄存器输出指定（C+1）指定为有输出时，将有最大值的 CH（存在多个 CH 时为低位 CH）的 I/O 存储器有效地址输出至 IR0。

带符号指定（C+1）时，将表的数据作为带符号 BIN 数据（负数为 2 的补数）进行处理。

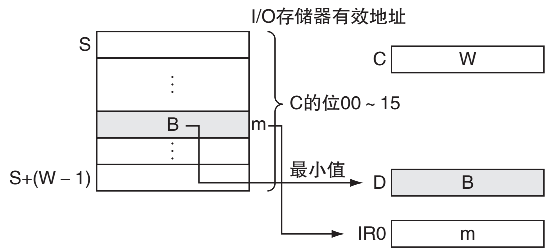


● MIN

以由 S 指定的表低位 CH 编号开始的由 C 指定的表长度（字数）为表数据，从中检索最小值，输出至 D。

索引寄存器输出指定（C+1）时，将有最小值的 CH（存在多个 CH 时为低位 CH）的 I/O 存储器有效地址输出至 IR0。

带符号指定（C+1）时，将表的数据作为带符号 BIN 数据（负数为 2 的补数）进行处理。

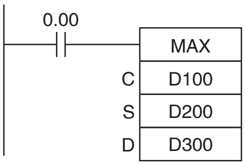


提示

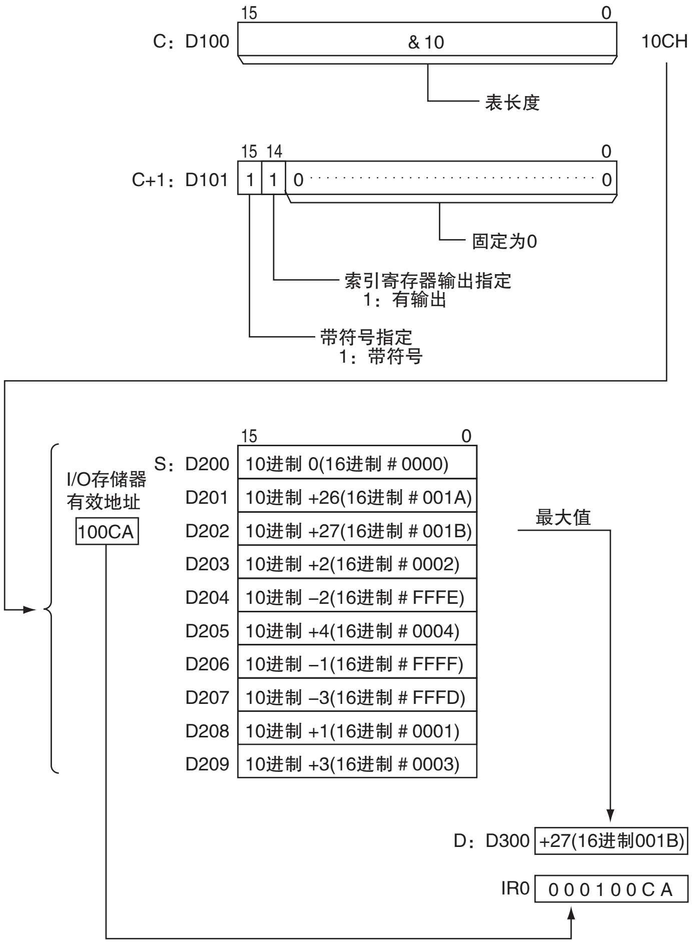
・在带符号指定（C+1 的位 15）中指定了带符号（1）时，将 16 进制 #8000 ～ FFFF 的数值作为负数进行处理。因此，检索结果因带符号指定的有无而异。

程序举例

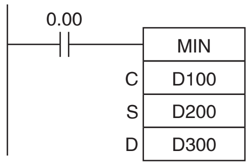
● MAX



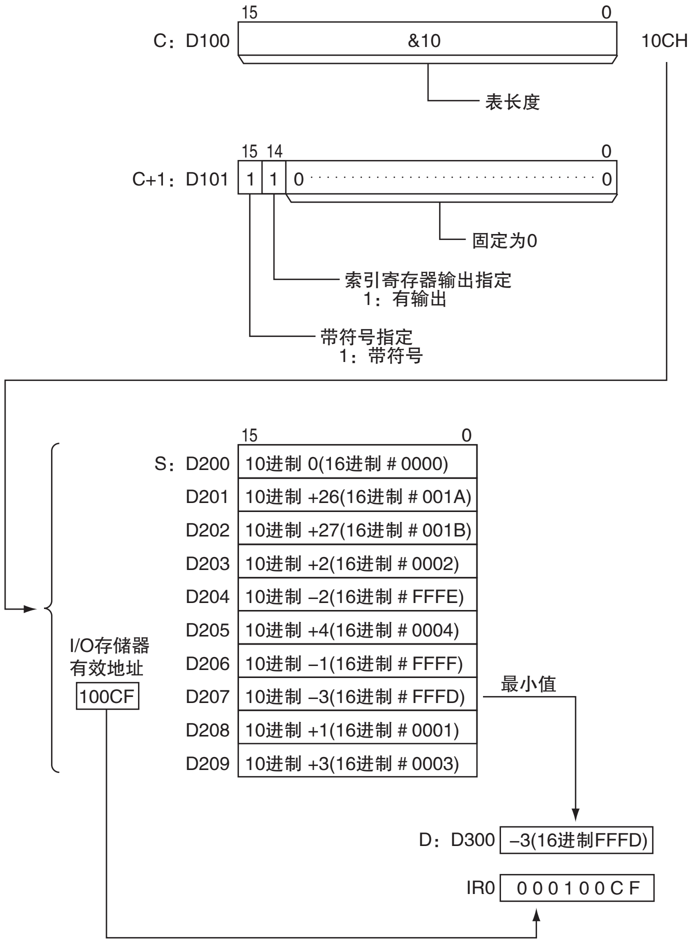
0.00为ON时，在从D200开始10CH(由D100指定)的表中检索最大值，将该值保存到D300。同时，将最大值所在地址D202的I/O存储器有效地址保存到IR0。



● MIN



0.00为ON时，在从D200开始10CH(由D100指定)的表中检索最小值，将该值保存到D300。同时，将最小值所在地址D207的I/O存储器有效地址保存到IR0。

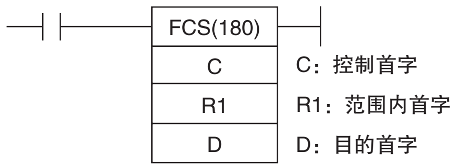


注意事项

可在 CP2E CPU 单元中使用 MAX/MIN 指令。不能在 CP1E CPU 单元中使用。

FCS

指令	助记符	变化	功能代码	功能
帧校验和	FCS	@FCS	180	计算指定范围的FCS值并以ASCII码输出结果。

	FCS
符号	

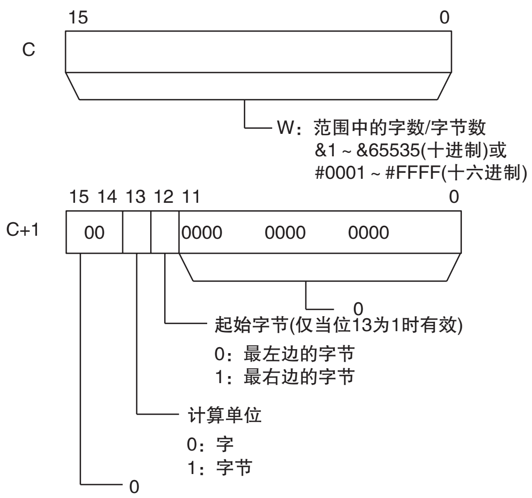
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

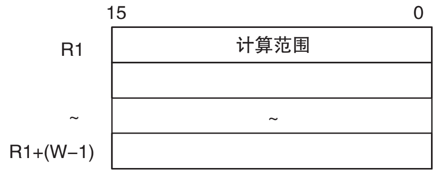
操作数

操作数	描述	数据类型	大小
C	控制首字	UDINT	2
R1	范围内首字	UINT	可变
D	目的首字	UINT	可变

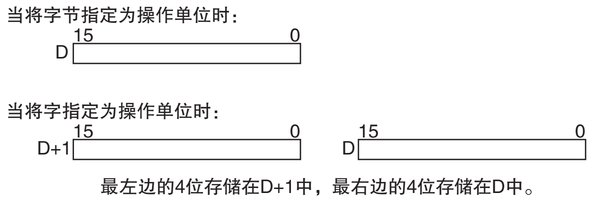
C：控制首字



R1：范围内首字



D：目的首字



注 C 和 C+1、计算范围内的所有字均必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
R1，D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

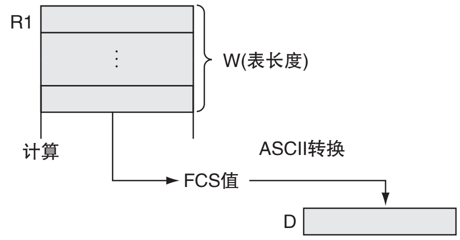
条件标志

名称	标记	操作
出错标志	P_ER	· 当 C 的内容不在 0001 ~ FFFF 的指定范围内时为 ON。 · 其它情况下 OFF。

功能

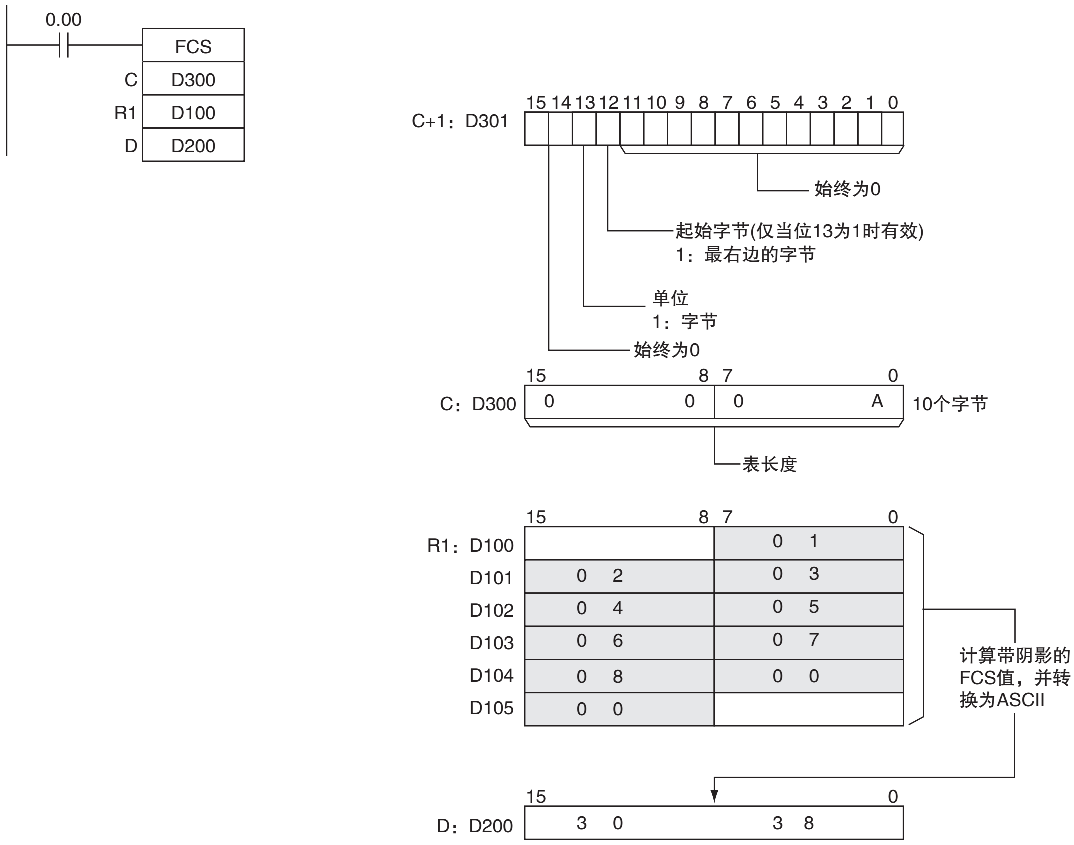
FCS(180) 计算从 R1 中的数据开始的 W 个单元的数据的 FCS 值，将该值转换成 ASCII 码，并将结果输出到 D(对于字节) 或 D+1 和 D(对于字) 中。C+1 中的设定决定相加时是以字还是字节为单位、是二进制数据 (带符号还是无符号) 还是 BCD 数据，以及若以字节为单位相加时，是从 R1 的右字节还是左字节开始。

当 C+1 的位 13 被置为 1 时，FCS(180) 指令以字节为单位对数据进行操作。在这种情况下，位 12 决定计算是从 R1 的最右字节 (位 12=1) 还是 R1 的最左字节 (位 12=0) 开始。



程序举例

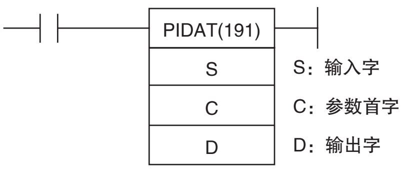
下例中，当 CIO 0.00 为 ON 时，FCS(180) 指令计算从 D100 的最右字节开始的 10 个字节的数据的 FCS 值，并将结果写入 D200 中。



数据控制指令

PIDAT

指令	助记符	变化	功能代码	功能
带自动整定的 PID 控制	PIDAT	---	191	根据指定的参数执行 PID 控制。PID 常数可进行自动整定。

	PID
符号	

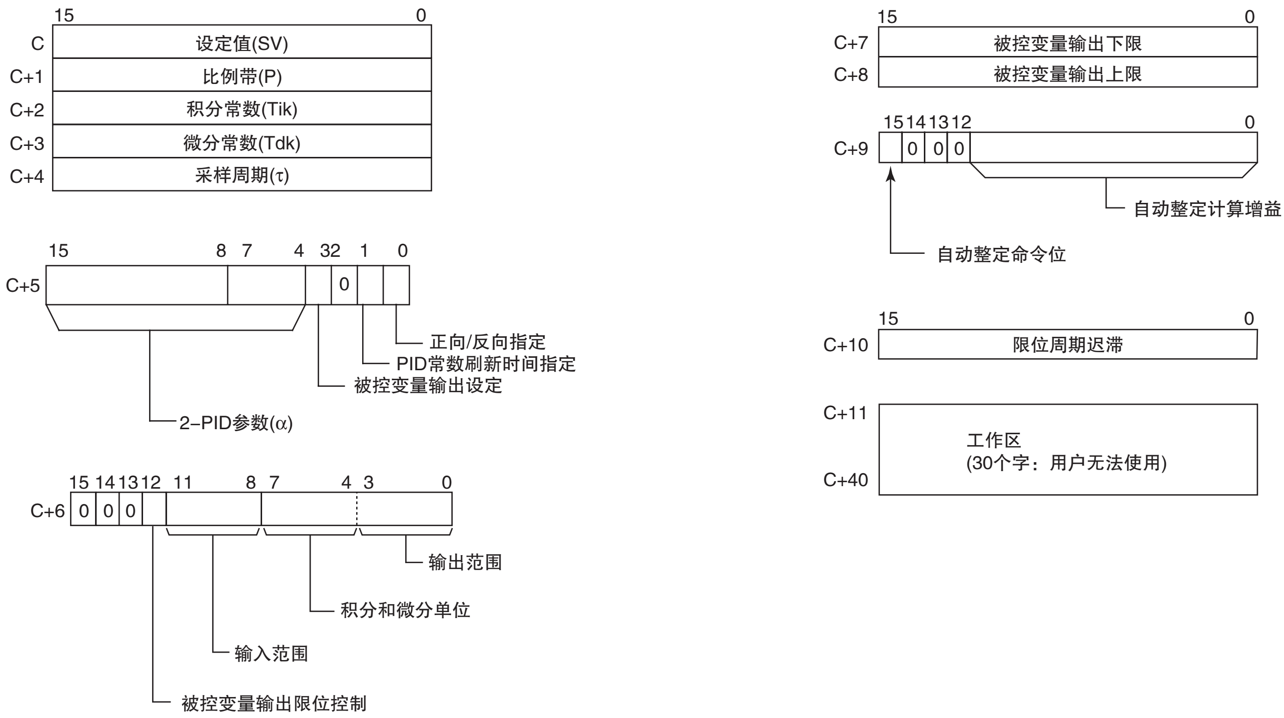
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型	大小
S	输入字	UINT	1
C	参数首字	WORD	41
D	输出字	UINT	1

C：参数首字



- **区域类型**

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

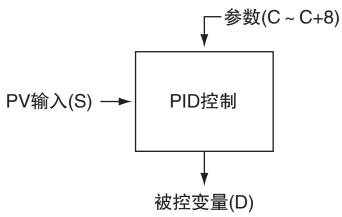
名称	标记	操作
出错标志	P_ER	· 当 C 数据超出范围时 ON。 · 当实际采样周期超过指定采样周期的 2 倍时 ON。 · 在自动整定期间，被控变量因限位周期而未在 9999 秒内发生变化时，则判定为错误并且 ON。 · 其它情况下 OFF。
大于标志	P_GT	· 当在 PID 作用后被控变量大于上限时 ON。 · 其它情况下 OFF。
小于标志	P_LT	· 当在 PID 作用后被控变量小于下限时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 正在执行 PID 控制时 ON。 · 其它情况下 OFF。

功能

当执行条件为 ON 时，PIDAT(191) 指令按照由 C 所指定的参数 (设定值、PID 常数等)，以两个自由度执行目标值过滤的 PID 控制。该指令采用输入字 S 的内容所指定的二进制数据输入范围，并按照所设定的参数执行 PID 作用。执行结果以被控变量的形式存入输出字 D 中。

当执行条件由 OFF 变 ON 时读取参数设定，而如果设定值超出许可范围，则出错标志将变 ON。
如果设定值在许可范围内，则将采用初始值执行 PID 处理。缓冲操作在此时不执行，而是用于后续 PID 处理过程中的被控变量。(缓冲操作过程是指为避免突然变化造成负面影响而逐渐和连续改变被控变量的过程。)

当执行条件变 ON 时 , 将输入指定采样周期的 PV, 并执行该处理。

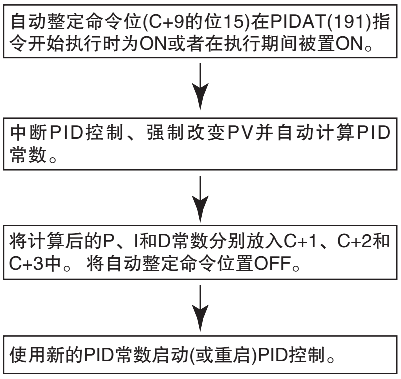


自动整定

每个循环均会对自动整定命令位 (C+9 的位 15) 的状态进行检查。如果在给定的循环中对该控制位置 ON，则 PIDAT(191) 指令将开始自动整定 PID 常数。(正在执行自动整定时，SV 的变化将不会反映出来。)
自动整定采用限位周期的方法。PIDAT(191) 指令强制改变被控变量 (最大被控变量 ↔ 最小被控变量) 并监控受控系统的特性。根据观察到的特性来计算 PID 常数，并将新的 P、I 和 D 常数自动存储到 C+1、C+2 和 C+3 中。此时，对自动整定命令位 (C+9 的位 15) 置 OFF，并使用 C+1、C+2 和 C+3 中新的 PID 常数来重续 PID 控制。

- 如果当 PIDAT(191) 指令开始执行时自动整定命令位为 ON，则将首先执行自动整定，然后再使用计算得出的 PID 常数来启动 PID 控制。
- 如果在 PIDAT(191) 指令执行期间对自动整定命令位置 ON，则 PIDAT(191) 指令将中断使用用户设定的PID常数所执行的PID控制，然后执行自动整定，再使用计算得出的PID常数来重续PID控制。

下列流程图所示为自动整定流程：



注 1 如果在自动整定期间通过对自动整定命令位置 OFF 的方法来中断自动整定，则 PID 控制将使用在自动整定开始之前所使用的 PID 常数来启动。

2 另外，如果产生自动整定执行错误，则 PID 控制将使用在自动整定开始之前所使用的 PID 常数来启动。

在注1和注2这两种情况下，如果PID常数在自动整定被中断之前即已计算得出，则这些常数将生效。

PID 控制

· PV 输入 (S) 的 16 个位中的有效输入数据位数由 C+6 的位 08 ～ 11 中的输入范围设定来指定。例如，如果为输入范围指定了 12 个位 (4 Hex)，则 PV 的有效范围将为 0000 Hex ～ 0FFF Hex。(大于 0FFF Hex 的值将作为 0FFF Hex 处理。)

· 设定值的范围同时还取决于输入范围。

· 测量值 (PV) 和设定值 (SV) 为从 0000 Hex ～输入范围最大值的无符号二进制数。

· 被控变量输出的 16 个位中的有效输出数据位数由 C+6 的位 00 ～ 03 中的输出范围设定来指定。例如，如果为输出范围指定了12个位(4 Hex)，则从0000 Hex ~ 0FFF Hex的范围将作为被控变量输出。

· 仅对于比例操作，当 PV 等于 SV 时的被控变量输出可指定如下：

0：输出 0%
1：输出 50%。

· 可将比例操作的方向指定为正向或反向。

· 可指定被控变量输出的上限和下限。

· 可以 10ms 为单位指定采样周期 (0.01 ～ 99.99s)，但实际的 PID 作用由采样周期和 PIDAT(191) 指令的执行时间 (每个循环) 共同决定。

· 使 PID 常数变化的时刻可设定为下列二者之一：1)PIDAT(191) 指令执行的开始时刻； 2)PIDAT(191) 指令执行和各个采样周期的开始时刻。在各个采样周期中 (即 PID 指令执行期间) 只能改变比例带 (P)、积分常数 (Tik) 和微分常数 (Tdk)。变化的时刻由 C+5 的位 1 进行设定。

· 当将积分和微分单位指定为 “1：采样周期倍数”，并且自动整定后积分常数 (Tik) 小于 1 时，则积分常数 (Tik) 变为 9999 (不执行积分运算)。同样的，当将积分和微分单位指定为 “1：采样周期倍数”，并且自动整定后微分常数 (Tdk) 小于 1 时，则微分常数 (Tdk) 变为 0 (不执行微分运算)。

提示

・PIDAT(191) 指令的执行条件类似于 “停止运行” 信号。对 C+11 ～ C+40 进行初始化后，当执行条件在下一循环保持 ON 时，将执行 PID 计算。因此，当使用常 ON 标志 (ON) 作为 PIDAT(191) 的执行条件时，请在操作开始时提供一个单独的过程对 C+11 ～ C+40 进行初始化。

注意事项

・多条 PIDAT 指令无法共享一个 PID 参数存储字。即使在多条 PIDAT 指令使用相同参数的情况下，也必须分别指定单独的存储字。

・当手动改变 PID 常数时，请将 PID 常数变化允许设定 (C+5 的位 1) 设为 1，从而使得 C+1、C+2 和 C+3 中的值在 PID 计算的每个采样周期中均进行刷新。另外，该设定还允许在自动整定之后手动调整 PID 常数。

・在 PID 参数 (C ～ C+40) 当中，只有下述参数在执行条件为 ON 时可以改变。若有任何其它值发生了改变，请务必将输入条件从 OFF 置为 ON，以使新的设定生效。

  ・C 中的设定值 (SV)
  (只能在 PID 控制期间改变。在自动整定期间的 SV 变化将不会反映出来。)

  ・PID 常数变化允许设定 (C+5 的位 1)

  ・C+1、C+2 和 C+3 中的 P、I 和 D 常数
  (仅当 PID 常数变化允许设定 (C+5 的位 1) 被设定为 1 时，对这些常数所作的改变才会在每个采样周期中反映出来。)

  ・自动整定命令位 (C+9 的位 15)

  ・自动整定计算增益 (C+9 的位 0 ～ 14) 和限位周期迟滞 (C+10)(这些值在自动整定开始时读入。)

执行规定

项目			规格
PID 控制方法		---	目标值过滤型两个自由度的 PID 控制方法 (正向 / 反向)
PID 控制的循环次数		---	无限制 (每个指令 1 个循环)
采样周期		τ	0.01 ～ 99.99s
PID 常数	比例带	P	0.1 ～ 999.9%
	积分常数	Tik	1 ～ 8191,9999(对采样周期倍数 9999，无积分作用)
	微分常数	Tdk	0 ～ 8191(对采样周期倍数 0，无微分作用)
设定值		SV	0 ～ 65535(一直到输入范围的最大值为止均有效)
测量值		PV	0 ～ 65535(一直到输入范围的最大值为止均有效)
被控变量		MV	0 ～ 65535(一直到输出范围的最大值为止均有效)

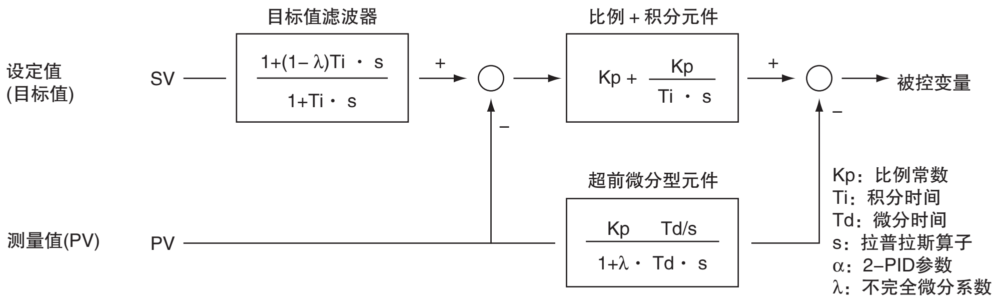
计算方法

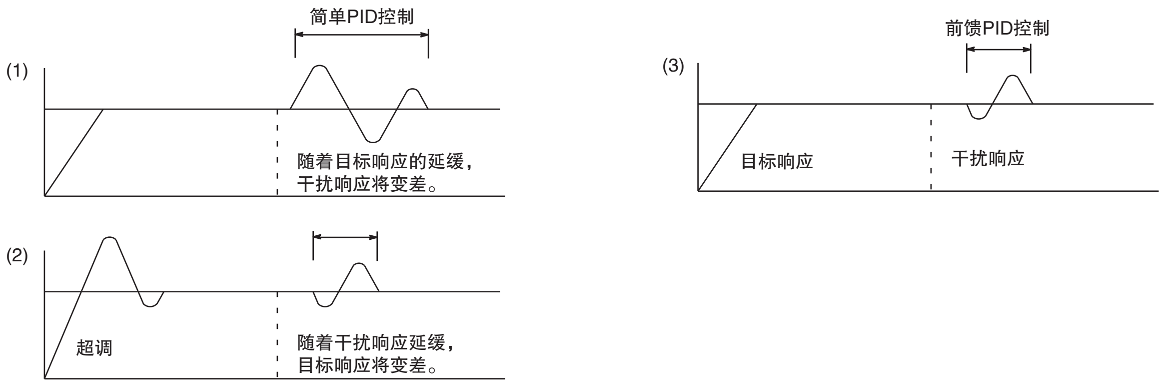
通过两个自由度的目标值过滤型控制来执行 PID 控制中的计算。

两个自由度的目标值 PID 方框图

当通过简单 PID 控制来防止超调时，干扰的稳定作用被延缓 (1)。而如果干扰的稳定作用被加速，则将发生超调，且对目标值的响应将延缓 (2)。

另一方面，当采用两个自由度的目标值 PID 控制方法时，不存在超调，且目标值的响应和干扰的稳定作用均可加速 (3)。





PID 参数设定

控制数据	项目	内容	设定范围	输入条件 ON 时的变化
C	设定值 (SV)	控制过程的目标值。	二进制数据 (与所指定的输入范围的位数相同)	允许
C+1	比例带	表示比例控制范围 / 总控制范围的 P 作用参数。	0001 ～ 270F Hex(1 ～ 9999)； (0.1% ～ 999.9%，以 0.1% 为单位)	若C+5的位1为1，则在输入条件 ON 时改变。
C+2	Tik 积分常数	表示积分作用强度的常数。积分强度随该值升高而降低。	0001 ～ 1FFF Hex(1 ～ 8191)； (9999= 不执行积分运算)(见注 1)	
C+3	Tdk 微分常数	表示微分作用强度的常数。微分强度随该值升高而降低。	0001 ～ 1FFF Hex(1 ～ 8191)； (0000= 不执行微分运算)(见注 1)	
C+4	采样周期 (τ)	设定执行 PID 作用的周期。	0001 ～ 270F Hex(1 ～ 9999)； (0.01 ～ 99.99s，以 10ms 为单位)	不允许
C+5 的位 04 ～ 15	2−PID 参数 (α)	输入滤波系数，通常为 0.65(即 000 设定)。滤波效率随滤波系数向 0 靠近而下降。	000 Hex：α=0.65 从 100 ～ 163 Hex 的设定表示将最右边两个数位的值设定为 α=0.00 ～ α=0.99。(见注 2)	
C+5 的位 03	被控变量输出指定	指定当 PV 等于 SV 时的被控变量输出。	0：输出 0% 1：输出 50%	
C+5 的位 01	PID 常数变化允许设定	允许改变用于 PID 计算的比例带 (P)、积分常数 (Tik) 和微分常数 (Tdk) 的时刻设定。	0：在 PID 指令开始执行时 1：在 PID 指令开始执行和各采样周期开始时	允许

控制数据	项目	内容	设定范围	输入条件 ON 时的变化
C+5 的位 00	PID 正向 / 反向指定	决定比例作用的方向。	0: 反向作用 1: 正向作用	不允许
C+6 的位 12	被控变量输出限位控制	决定是否对被控变量输出应用限位控制。	0: 禁止 (无限位控制) 1：允许 (限位控制)	
C+6 的位 08 ～ 11	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	
C+6 的位 04 ～ 07	积分和微分单位	决定用于表示积分和微分常数的单位。	1: 采样周期倍数 9: 时间 (单位：100ms)	
C+6 的位 00 ～ 03	输出范围	输出数据的位数。(输出的位数自动与输入的位数相同。)	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	
C+7	被控变量输出下限	被控变量输出限位有效时的下限。	0000 ～ FFFF(二进制) (见注 3)	
C+8	被控变量输出上限	被控变量输出限位有效时的上限。	0000 ～ FFFF(二进制) (见注 3)	
C+9 的位 15	自动整定命令位	该控制位启动自动整定。 · 将自动整定命令位设定为 1 将执行自动整定。(正在执行 PIDAT(191) 指令时，可启动自动整定。) · 当自动整定完成时，将自动对该位置 OFF。 自动整定执行时，如果将设定从 1 变为 0，则自动整定被中断，并根据自动整定执行前的PID参数开始PID计算。但是，如果自动整定执行后 PID 参数发生了改变，则以中断时所设定的值开始 PID 计算。	作为控制位： · 0 → 1: 执行自动整定。 · 1 → 0: 中断自动整定。 当自动整定完成时，PID(191) 指令将自动对该位置 OFF。 作为标志： 0：自动整定未执行。 1：自动整定正在执行。	允许
C+9 的位 00 ～ 11	自动整定计算增益	设定该参数，以调整 PID 计算结果对存储值的影响。 通常使该参数保持默认设定 (0000)。 · 要强调稳定性时，增大该值。 · 要强调响应时，减小该值。	0000 Hex：1.00(默认) 0001 ～ 03E8 Hex(1 ～ 1000) ; (0.01 ～ 10.00，以 0.01 为单位)	允许 (在自动整定开始时读入这些参数)
C+10	限位周期迟滞	当限位周期产生时设定迟滞。反向操作的默认值以 SV−20% 的迟滞对 MV 置 ON。 如果由于 PV 不稳定而导致无法产生合适的限位周期，则可提高该设定。但若限位周期迟滞高于所需值，则自动整定精度将下降。	0000 Hex：0.20%(默认) 0001 ～ 03E8 Hex： 0.01 ～ 10.00%，以 0.01% 为单位 FFFF Hex：0.00% **注** 百分比与输入范围相对应。	

注 1 当将单位指定为 1 时，范围为 1 ～ 8,191 倍周期。当将单位指定为 9 时，范围为 0.1 ～ 819.1s。指定为 9 时，将积分和微分次数设定到 1 ～ 8,191 倍采样周期的范围内。

2 将 2−PID 参数 (α) 设定为 000 时将产生 0.65，即标准值。

3 当被控变量输出限位控制有效 (即设定为 “1”) 时，请如下设定各值：
0000 ≤ MV 输出下限≤ MV 输出上限≤输出范围最大值

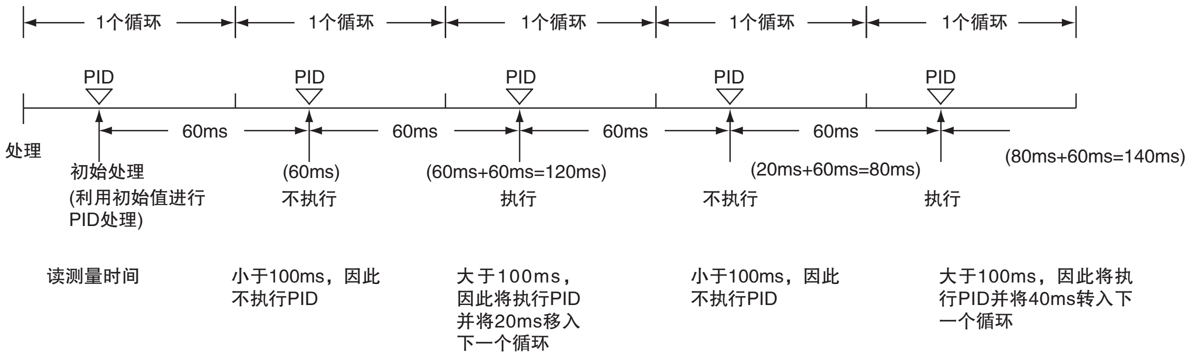
采样周期和循环时间

可以 10ms 为单位指定采样周期 (0.01 ～ 99.99s)，但实际的 PID 作用由采样周期和 PID 指令的执行时间 (每个循环) 共同决定。采样周期和循环时间之间的关系如下：

- 如果采样周期小于循环时间，则 PID 控制在每个循环而非每个采样周期中执行。
- 如果采样周期大于等于循环时间，则 PID(190) 不是在每个循环中执行，而是当循环时间 (PID 指令之间的时间) 的累计值大于等于采样周期时执行。累计值的剩余部分 (即循环时间的累计值减采样周期) 转入下一个累计值。

  例如，假设采样周期为 100ms 且循环时间恒定为 60ms。对于初始执行后的第一个循环，将不执行 PID(190)，因为 60ms 小于 100ms。对于第二个循环，60ms+60ms 大于 100ms，因此将执行 PID(190)。剩余的 20ms(即 120ms−100ms=20ms) 将转入下一个循环累计值。

  对于第三个循环，将余值 20ms 与 60ms 相加。由于和值 80ms 小于 100ms，因此将不执行 PID(190)。对于第四个循环，将 80ms 与 60ms 相加。由于和值 140ms 大于 100ms，则将执行 PID(190)，且将余值 40ms(即 140ms−100ms=40ms) 转入下一个循环累计值。在后续循环中将重复该步骤。



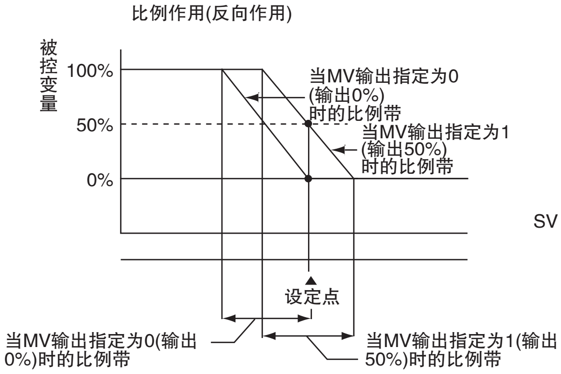
PID 控制

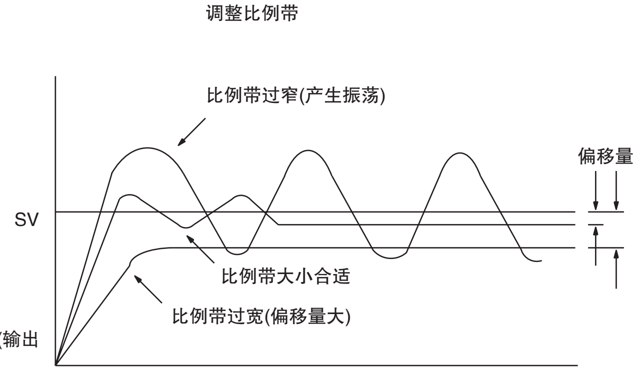
● 比例作用 (P)

比例作用是指建立在设定值 (SV) 上的比例带操作，在此带内被控变量 (MV) 与偏差成正比。下图所示为一个反向操作举例。

如果采用比例作用且当前值 (PV) 小于比例带，则被控变量 (MV) 为 100%(即最大值)。在比例带内，MV 与偏差 (SV 与 PV 之间的差) 成比例，并逐渐减小直到 SV 与 PV 相匹配 (即直到偏差为 0)，此时 MV将为最小值0%(或50%，取决于被控变量输出指定参数的设定)。当PV大于SV时，MV也将为0%。

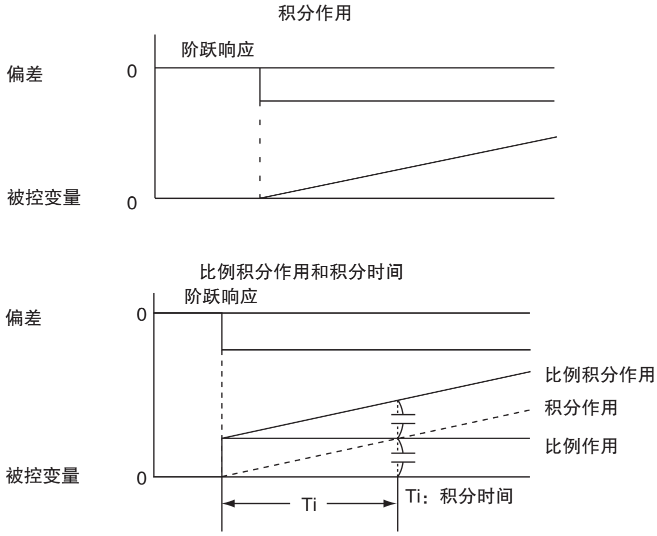
比例带用总输入范围的百分比来表示。比例带越小，则比例常数越大且校正作用越强。采用比例作用时，通常会产生一个偏移量 (剩余偏差)，但可通过减小比例带来减小该偏移量。但如果偏移量过小，将产生振荡。





● 积分作用 (I)

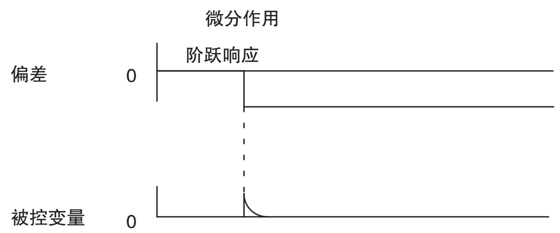
将积分作用和比例作用组合使用可使偏移量随时间流逝而减小，因此 PV 将与 SV 匹配。积分作用的强度通过积分时间（ 即对于阶跃偏差而言，积分作用的被控变量达到与比例作用的被控变量的相同水平所需的时间 ）来表示，如下图所示。积分时间越短，则积分作用的校正效果越强。如果积分时间过短，则校正作用将过强，从而导致振荡产生。

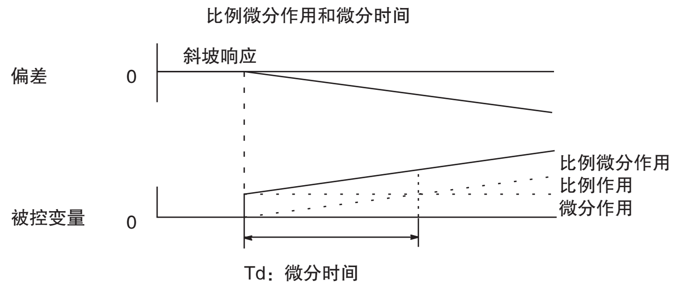


● 微分作用 (D)

比例作用和积分作用均通过控制结果进行校正，因此不可避免地存在响应延迟。微分作用可弥补这一缺陷。对突发干扰作出响应时，微分作用可产生一个很大的被控变量，然后迅速恢复原始状态。通过与偏差引起的斜坡 (微分系数) 成比例的被控变量来执行校正。

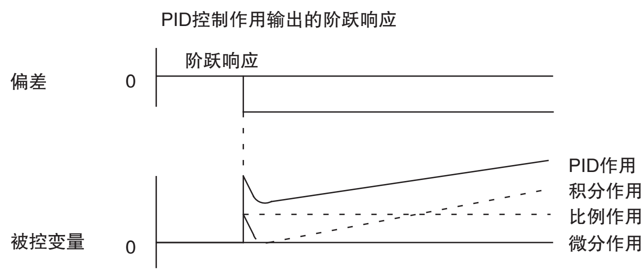
微分作用的强度通过微分时间（ 即对于阶跃偏差而言，微分作用的被控变量达到与比例作用的被控变量的相同水平所需的时间) 来表示，如下图所示。微分时间越长，则微分作用的校正效果越强。

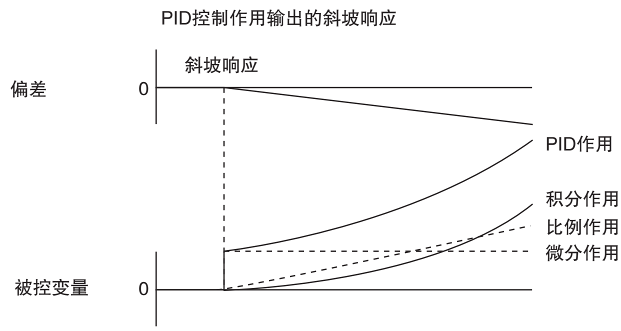




PID 作用

PID 作用由比例作用 (P)、积分作用 (I) 和微分作用 (D) 共同组成，即使对于有死区的控制对象也可产生极佳的控制结果。PID 作用中的比例作用可提供无振荡的平衡控制，积分作用可自动校正偏移，而微分作用则可加速对干扰的响应。

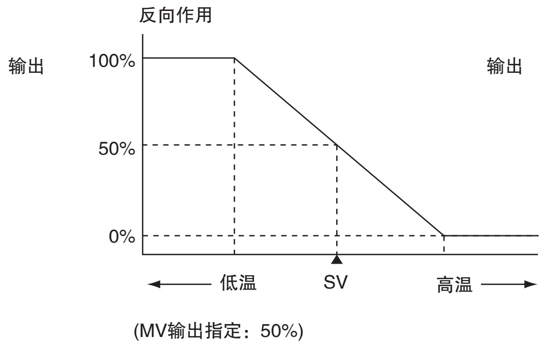




作用方向

采用 PID 控制时，请选择以下两种控制方向之一。无论选择何种方向，MV 均随 SV 和 PV 之差增大而增大。

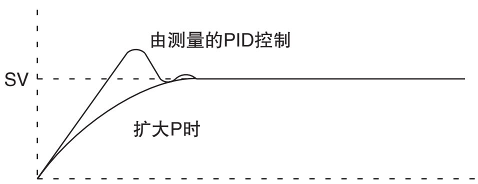
- 正向作用：当 PV 大于 SV 时 MV 增大。
- 反向作用：当 PV 小于 SV 时 MV 增大。



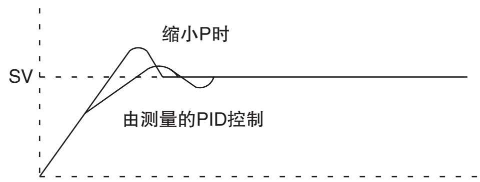
调整 PID 参数

PID 参数和控制状态之间的一般关系如下。

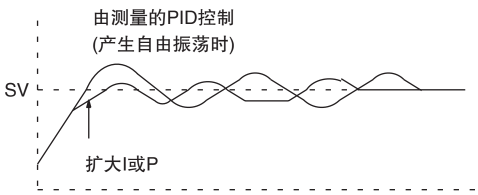
- 当达到稳定状态所需的时间 (稳定时间) 不构成问题、但不希望造成超调时，应扩大比例带。



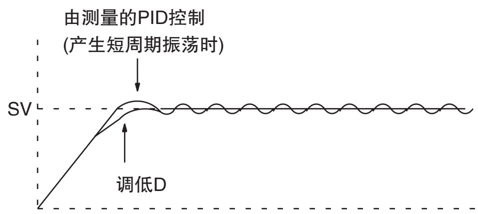
- 当超调不构成问题、但希望实现快速稳定控制时，应缩小比例带。但如果比例带过窄，则可能会产生振荡。



- 当存在宽幅振荡或者操作与超调和欠调密切相关时，则可能是因为积分作用太强所致。如果增加积分时间或扩大比例带，则将减轻振荡。

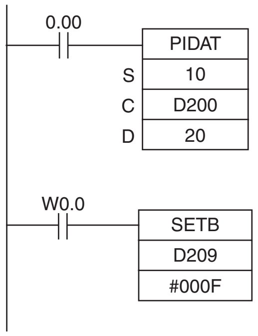


- 如果产生短周期振荡，则可能是因控制系统的响应太快和微分作用太强所致。此时可调弱微分作用。

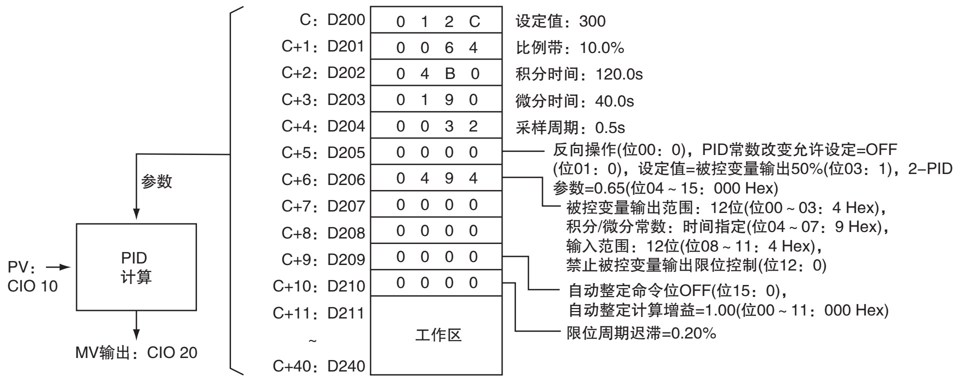


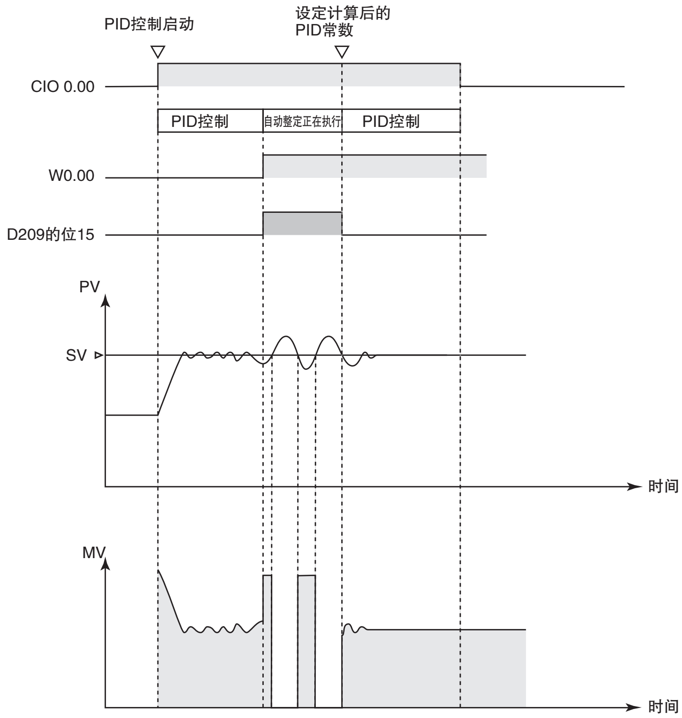
程序举例

● 中断 PID 控制以执行自动整定

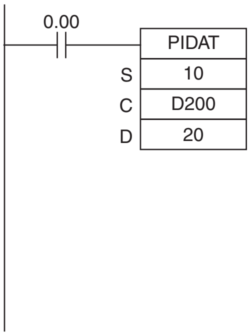


- 在 CIO 0.00(OFF → ON) 的上升沿，将根据 D200 ～ D208 中设定的参数 (如下所示) 对 D211 ～ D240 中的工作区进行初始化。对工作区进行初始化之后，将执行 PID 控制并将被控变量输出到 CIO 20。
- 当对 CIO 0.00 置 ON 之后，将根据 D200 ～ D210 中设定的参数，以采样周期间隔执行 PID 控制。被控变量将输出到 CIO 20。
- 即使比例带 (P)、积分常数 (Tik) 或微分常数在 CIO 0.00 变 ON 之后发生了改变，PID 计算中使用的 PID 常数也不会改变。
- 在 W 0.0(OFF → ON) 的上升沿，SETB(532) 指令对 D209(C+9) 的位 15 置 ON 并启动自动整定。当自动整定过程完成时，将 P、I 和 D 常数写入 C+1、C+2 和 C+3 中。然后使用新的 PID 常数重新启动 PID 控制。

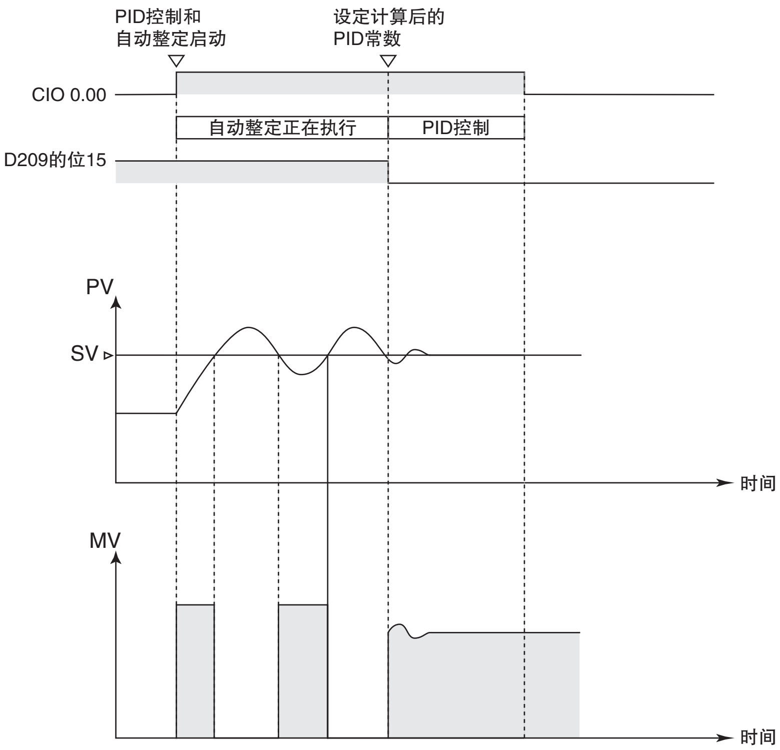




● 启动 PIDAT(191) 自动整定

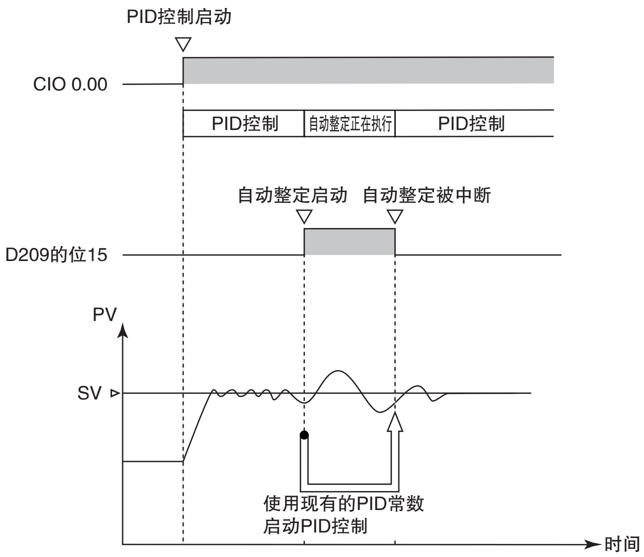


在 CIO 0.00(OFF → ON) 的上升沿，如果 D209(C+9) 的位 15 为 ON，将首先执行自动整定。当自动整定过程完成时，将 P、I 和 D 常数写入 C+1、C+2 和 C+3 中。然后使用计算后的 PID 常数启动 PID 控制。



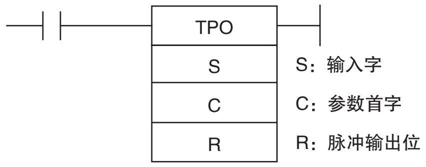
● 在自动整定完成之前使其中断

可通过将 D209(C+9) 的位 15 从 ON 置为 OFF 的方法来中断自动整定过程。PID 控制将使用在自动整定开始之前有效的 P、I 和 D 常数来重启。



TPO

指令	助记符	变化	功能代码	功能
时间比例输出	TPO	---	685	从指定字输入占空比或被控变量，根据指定参数将占空比转换成时间比例输出，然后从指定的输出参数输出结果。

	TPO
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	输入字	UINT	1
C	参数首字	WORD	7
R	脉冲输出位	BOOL	---

S：输入字

指定包含输入占空比或被控变量的输入字。

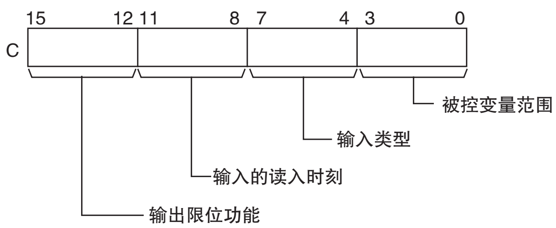
- 输入占空比：0000 ～ 2710 Hex(0.00% ～ 100.00%)
- 输入被控变量 (见注)：0000 ～ FFFF Hex(0 ～最大 65,535) (C 的位 00 ～ 03 指定被控变量的范围，即被控变量中的有效位数。指定与为 PIDAT(190) 指令中的输出范围设定所指定的位数相同的位数。)

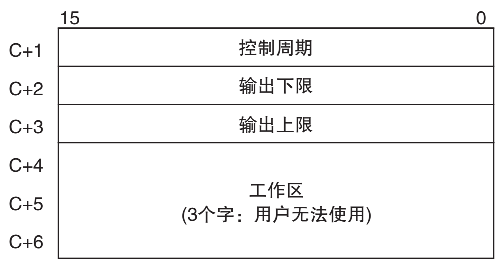
注 如果 S 为被控变量，则指定包含从 PIDAT(191) 指令输出的被控变量的字。

C：参数首字

C 的位 04 ～ 07 指定输入类型，即输入字是包含输入占空比还是被控变量。(将这些位置为 0 Hex 时指定输入占空比，置为 1 Hex 时指定被控变量。)

下图所示为参数数据的位置。





注 详情请参阅对各参数的描述。

R：脉冲输出位

指定脉冲输出的目的输出位。

通常指定一个分配到晶体管输出单元的输出位，并将固态继电器连接到晶体管输出单元。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
C	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
D	OK	OK	OK	OK	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	ER	· 当 S 中的输入数据超出范围时 ON。(输入数据设定范围取决于输入类型设定。) · 当 C 数据超出范围时 ON。(仅当将输入类型设定为被控变量时，被控变量范围将出错。) · 当 C+1 中的控制周期超出范围时 ON。 · 当输出限位功能有效但输出下限 (C+2) 或输出上限 (C+3) 超出范围时 ON。 · 当输出限位功能有效但输出下限 (C+2) 小于等于输出上限 (C+3) 时 ON。 · 其它情况下 OFF。

功能

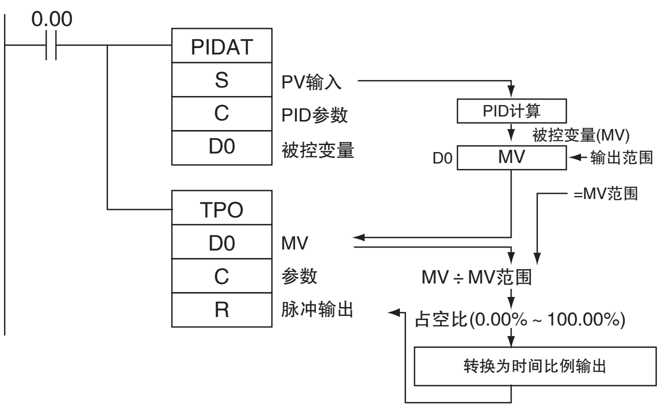
从由 S 所指定的字地址中接收占空比或被控变量输入，根据字 C ～ C+3 中所指定的参数将占空比转换成时间比例输出 (见注)，然后将脉冲输出输出到由 R 所指定的位。

注 时间比例输出根据输入字 S 中的 ON/OFF 比例来按比例改变。ON 和 OFF 状态发生改变的周期称为控制周期，在参数字 C+1 中进行设定。
例：当控制周期为 1s 且输入值为 50% 时，该位为 ON 状态的时间为 0.5s，为 OFF 状态的时间为 0.5s。当控制周期为 1s 且输入值为 80% 时，该位为 ON 状态的时间为 0.8s，为 OFF 状态的时间为 0.2s。

通常情况下，将 TPO(685) 指令与 PIDAT(191) 指令一起使用，且将 PID 指令的被控变量结果字 (D) 指定为 TPO(685) 指令的输入字 (S)。另外，通常将分配给晶体管输出单元的输出位指定为 R，并将固态继电器连接到晶体管输出单元，从而对加热器执行时间比例控制 (ON/OFF 比例控制)。

● 组合使用 TPO(685) 和 PID 控制指令

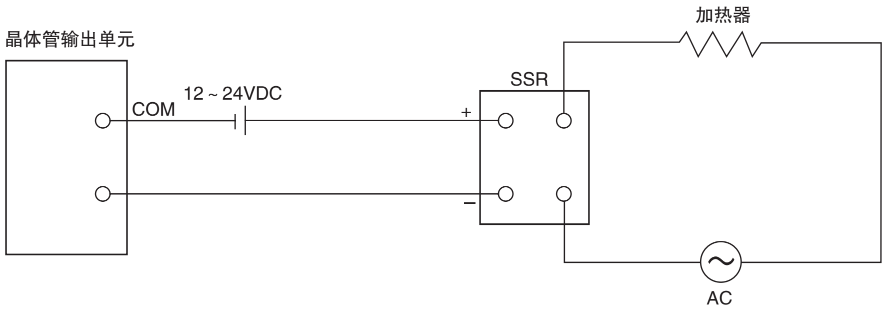
将 TPO(685) 指令与 PID 控制指令组合使用时，用被控变量输入除以被控变量范围以计算占空比，再将该占空比转换成时间比例输出并输出脉冲。



在这种情况下，请将 PID 控制指令的输出范围和 TPO(685) 指令的被控变量范围设定为相同值 例如，将 PID 控制指令的输出范围和 TPO(685) 指令的被控变量范围均设定为 12 个位 (0000 ~ 0FFF Hex) 时，用从 PID 控制指令输出的被控变量除以 0FFF Hex 即可计算得出占空比，而 TPO(685) 指令则将该占空比转换成时间比例输出。

● 外部配线举例

如下图所示，将晶体管输出单元连接到固态继电器 (SSR) 上。



参数设定

控制数据		项目	内容	设定范围	输入条件 ON 时的变化
字	位				
C	00 ~ 03	被控变量范围	指定输入数据的位数。	0 Hex：8 位　5 Hex：13 位 1 Hex：9 位　6 Hex：14 位 2 Hex：10 位　7 Hex：15 位 3 Hex：11 位　8 Hex：16 位 4 Hex：12 位	允许
	04 ~ 07	输入类型	指定 S 中包含占空比还是被控变量。	0 Hex：占空比 S 的设定范围: 0000 ~ 2710 Hex(0.00 ~ 100.00%) 1 Hex：被控变量 S 的设定范围：0000 ~ FFFF Hex(0 ~ 65,535) (最大设定取决于通过 C 的位 00 ~ 03 所设定的 MV 范围)	允许
	08 ~ 11	输入的读入时刻	指定输入的读入时刻	0 Hex：使用控制周期的起始值 1 Hex：使用较小值 2 Hex：使用较大值 3 Hex：连续调整	允许
	12 ~ 15	输出限位控制	指定输出限位功能是有效还是无效。	0 Hex：无效 1 Hex：有效 (见注)	允许
C+1	00 ~ 15	控制周期	控制周期 (ON/OFF 状态发生改变的时段)	0064 ~ 270F Hex(1.00 ~ 99.99s) **注** 例如，1.00s被设为0064 Hex而非0001 Hex。	允许
C+2	00 ~ 15	输出下限	指定输出限位有效时的下限。	0000 ~ 2710 Hex(0 ~ 100.00%)	允许
C+3	00 ~ 15	输出上限	指定输出限位有效时的上限。	0000 ~ 2710 Hex(0 ~ 100.00%)	允许
C+4	00 ~ 15	工作区	该工作区由系统使用，用户无法使用。	不可使用	---
C+5	00 ~ 15				
C+6	00 ~ 15				

注 当输出限位控制功能有效时，将下限和上限设定如下：
0000 Hex ≤下限≤上限≤ 2710 Hex.

执行

· 当输入条件为 ON 时执行该指令。

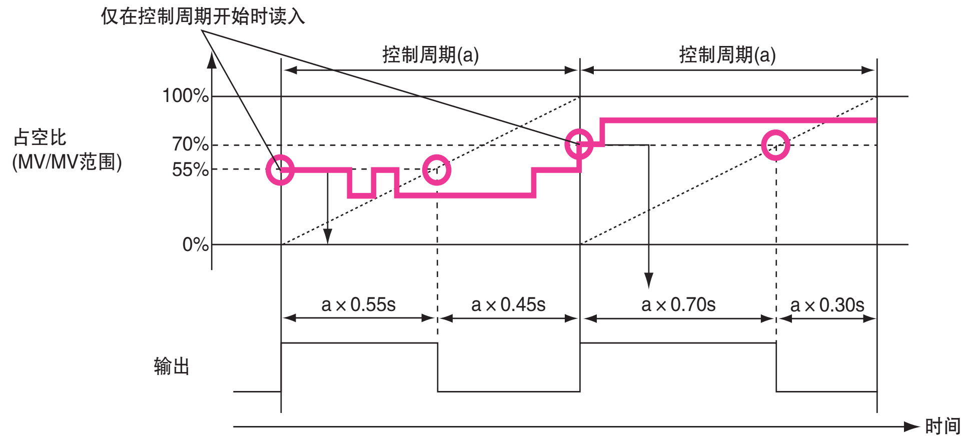
· 指令开始执行时，将根据占空比来对输出位 (R) 置 ON/OFF。

- 在每次执行指令时，均实时读入参数 (C ～ C+3)。当改变参数时，请同时改变所有参数，以免混淆不同的参数组。
- 执行指令时将对输出 (R) 置 ON/OFF，且输出的 ON/OFF 时刻的最高精度为 10ms。
- 当输入条件变 OFF 时，指令停止执行。此时，将对已经过的时间进行复位，并对控制周期进行初始化。
- 输入类型设定 (C 的位 04 ～ 07) 决定输入字 (S) 是包含占空比还是被控变量。当 S 包含被控变量时，用被控变量输入除以被控变量范围 (C 的位 00 ～ 03) 即可计算得出占空比。
- 输入的读入时刻设定 (C 的位 08 ～ 11) 指定何时读入输入字 (S)，如下表所示：

输入的读入时刻	描述
0:　使用控制周期的起始值	在控制周期开始时读入占空比输入，并且在控制周期期间不能改变该占空比。
1:　使用较小值	如果占空比输入降至控制周期开始时的占空比下方，则较小的占空比值优先级较高，并且将相应减少输出 ON 时间。
2:　使用较大值	如果占空比输入升至控制周期开始时的占空比上方，则较大的占空比值优先级较高，并且将相应增加输出 ON 时间。
3:　连续调整	每次执行指令时均实时读入占空比，并在控制周期内重复 ON/OFF 操作。

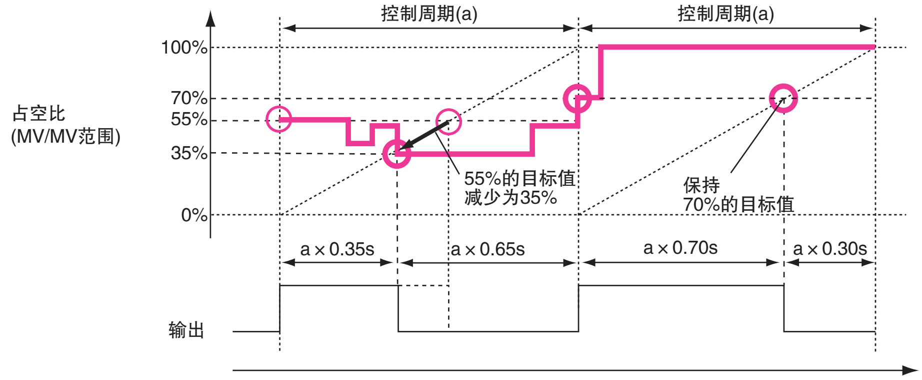
下图所示为各输入的读入时刻设定的操作。

- 输入时刻设定 =0(使用控制周期的起始值)



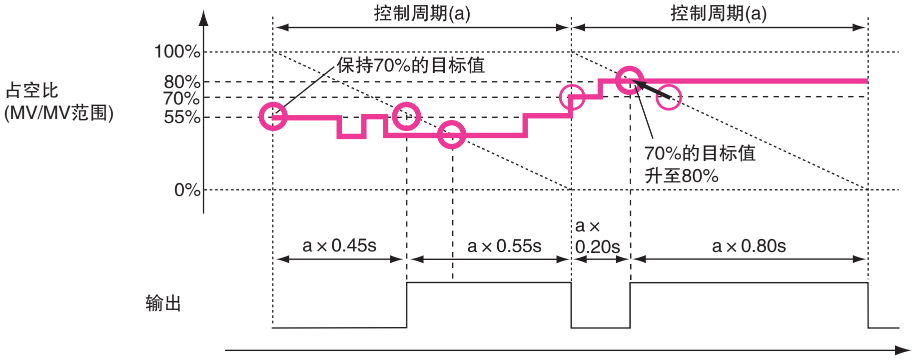
各控制周期的输出由该周期开始时的占空比决定。
该设定用于一般场合。

- 输入时刻设定 =1(使用较小值)



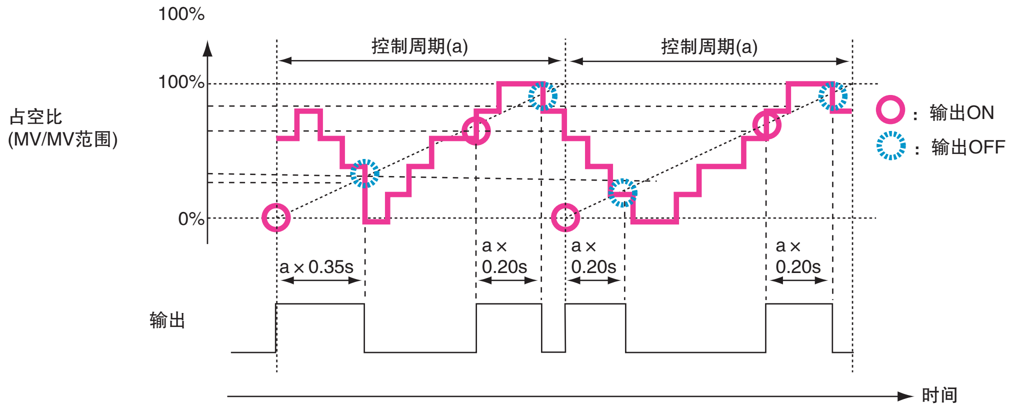
如果占空比在足够早的时间降至初始值的下方，则将对占空比进行调整，并且会在更早的时间对输出置OFF。
该设定可用于在使用时间比例控制来控制加热时想要避免超调的场合以及使用相对较长的周期控制的场合。

・ 输入时刻设定 =2(使用较大值)



如果占空比在足够早的时间升至初始值的上方，则将对占空比进行调整，并且会在更早的时间对输出置ON。(采用该设定时，输出的ON/OFF顺序将反转，且输出将从OFF变ON。)
该设定可用于在使用时间比例控制来控制冷却时想要避免欠调的场合以及使用相对较长的周期控制的场合。

・ 输入时刻设定 =3(连续调整)

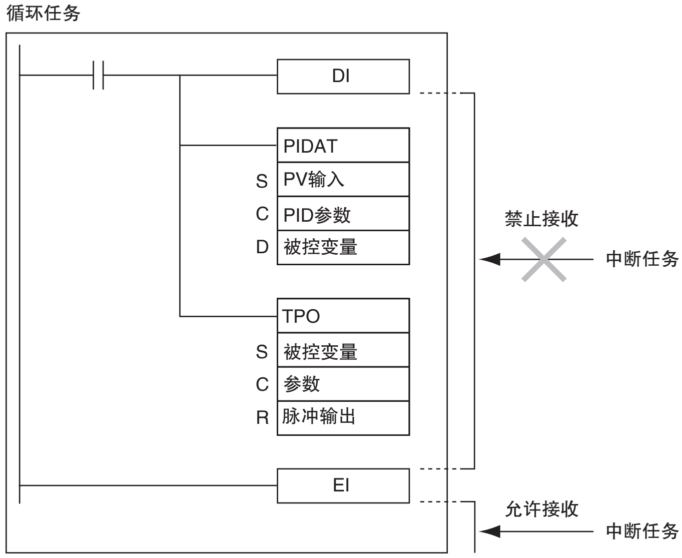


对占空比的变化进行实时监控。 如果占空比在足够早的时间降至初始值的下方，则将对占空比进行调整，并且会在更早的时间对输出置OFF。 如果此后占空比又再次上升，则将再次对占空比进行调整，并且对输出置ON。 持续重复该过程。该设定可用于在控制周期相对较长和占空比快速改变的场合改善响应性。 该设定还适合要求精准控制的照明或供电应用场合。

・ 可使输出限位功能 (C 的位 12 ～ 15) 有效，从而在输出超出输出限位的下限 (C+2) 和上限 (C+3) 之间的范围时对输出进行限制 (饱和)。

注意事项

当在某个循环任务中将 TPO(685) 指令与 PIDAT(191) 指令组合使用并且另外使用中断任务时，请在 PIDAT(191) 和 TPO(685) 指令之前执行 DI(693)(禁止中断) 指令来临时禁止中断。如果不禁止中断，并且在 PIDAT(191) 和 TPO(685) 指令之间发生了中断，则控制周期可能会移动。

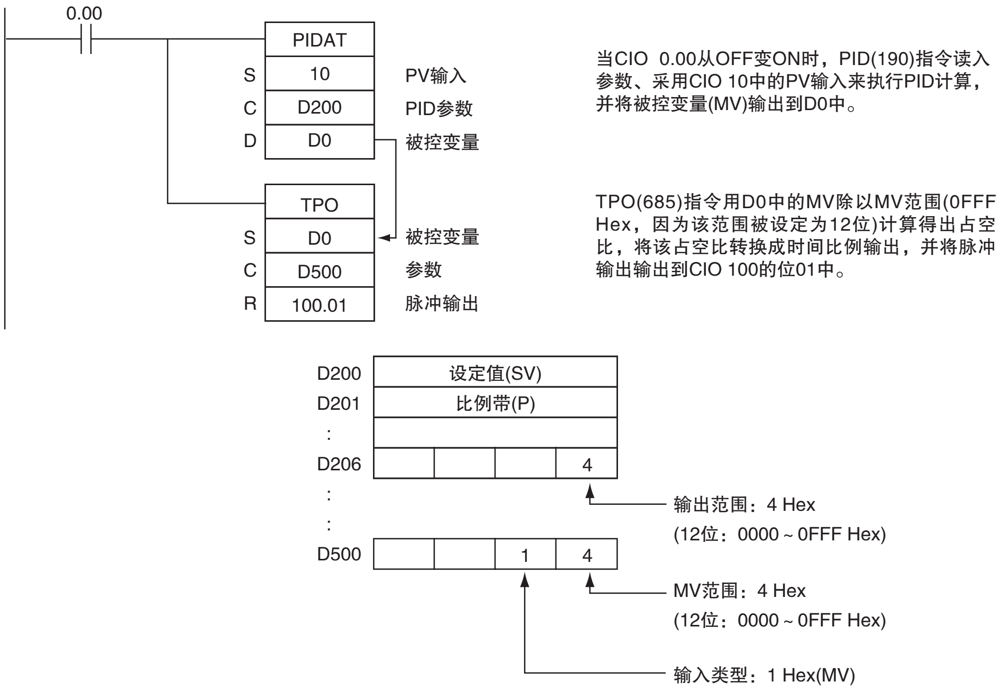


程序举例

● 组合使用 TPO(685) 和 PIDAT(191)

当 CIO 0.00 为 ON 时，TPO(685) 指令获得从 PIDAT(191) 指令输出的被控变量 (包含在 D0 中)，根据被控变量计算占空比 (占空比 = MV ÷ MV 范围)，然后将占空比转换成时间比例输出，并将脉冲输出到 CIO 100 的位 01 中。

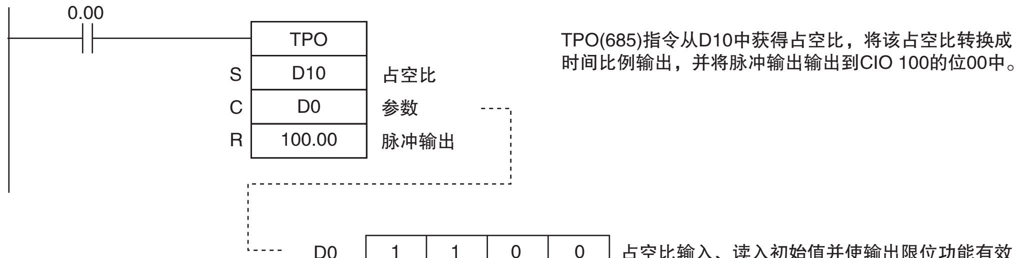
在这种情况下，CIO 100 被分配到晶体管输出单元，且位 CIO 100.01 被连接到用于控制加热器的固态继电器上。



● 单独使用 TPO(685) 指令

当 CIO 0.00 为 ON 时，TPO(685) 指令从 D10 中获得占空比，将该占空比转换成时间比例输出，并将脉冲输出到 CIO 100 的位 00 中。

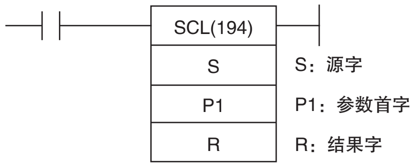
在这种情况下，控制周期为1s且输出限位功能有效，输出限位功能的下限为20.00%，上限为80.00%。



D0	1	1	0	0	占空比输入、读入初始值并使输出限位功能有效
D1	0	0	6	4	控制周期=1.00s
D2	0	7	D	0	输出下限=20.00%
D3	1	F	4	0	输出上限=80.00%
D4	不设定				
D5	不设定				
D6	不设定				
:					
:					
D10	0 ~ 2710 Hex				0 ~ 100.00%

SCL

指令	助记符	变化	功能代码	功能
定标	SCL	@SCL	194	根据指定的线性函数，将无符号二进制数据转换成无符号 BCD 数据。

	SCL
符号	

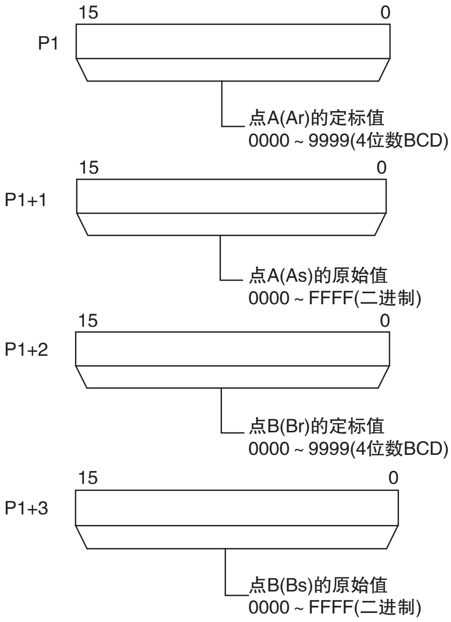
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	UINT	1
P1	参数首字	LWORD	4
R	结果字	WORD	1

P1：参数首字



注 P1 ~ P1+3 必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
P1	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・P1(Ar) 或 P1+1(Br) 的内容不是 BCD 时 ON。 ・P1+1(As) 和 P1+3(Bs) 的内容相等时 ON。 ・其它情况下 OFF。
等于标志	P_EQ	・结果为 0 时 ON。 ・其它情况下 OFF。

功能

SCL(194) 指令用于将源字 S 中包含的无符号二进制数据转换成无符号 BCD 数据，并根据由点 (As，Ar) 和 (Bs，Br) 定义的线性函数将结果放入结果字 R 中。包含坐标点 (As，Ar) 和 (Bs，Br) 的第一个字的地址由第一个参数字 P1 指定。这些点在定标前由 (As 和 Bs) 两值定义，在定标后由 (Ar 和 Br) 两值定义。

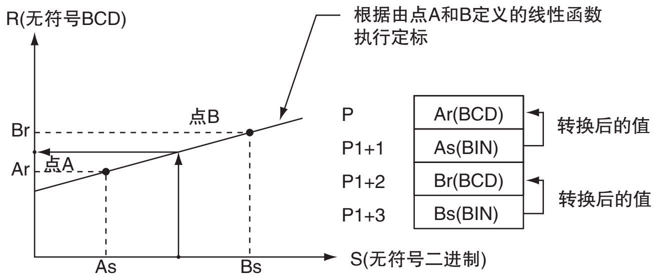
下列公式用于转换。

$$R=Bd-\frac{(Br-Ar)}{(Bs-As)\text{的BCD转换值}}\times(Bs-S)\text{的BCD转换值}$$

点 A 和 B 可定义一条正斜率或负斜率的直线。采用负斜率可实现反向定标。

・ 结果将四舍五入到最接近的整数。如果结果小于 0000，则将输出 0000 作为结果。

・ 如果结果大于 9999，则将输出 9999 作为结果。



提示

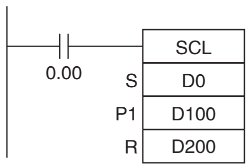
・ SCL(194) 指令可用于按照用户定义的定标参数，对来自模拟量输入单元的模拟信号转换值的结果进行定标。例如，如果有一个 1 ～ 5V 的模拟量输入单元的输入值以十六进制的 0000 ～ 0FA0 的形式输入到存储器中，则可使用 SCL(194) 指令将该值定标为 50↔200 ℃。

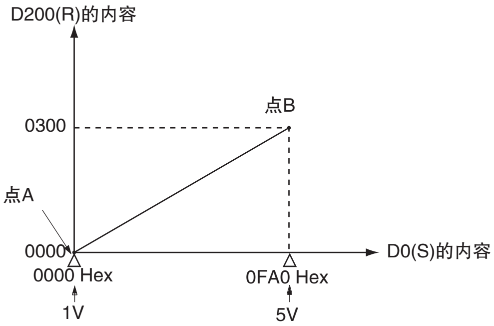
・ SCL(194) 指令将无符号二进制数转换成无符号 BCD 数据。若要转换负值，则在使用 SCL(194) 指令之前，必须先在程序中加最大的负值 (见下例)。
SCL(194) 指令无法将负值输出到结果字 R 中。如果结果为负值，则将 0000 输出到 R 中。

程序举例

下例中，假设要转换一个 1 ~ 5V 的模拟量信号，并作为十六进制数 0000 ~ 0FA0 输入到 D0 中。使用 SCL(194) 指令将 CIO 200 中的值转换 (定标) 为 0 ~ 300 BCD 之间的值。

当 CIO 0.00 为 ON 时，采用由点 A(0000，0000) 和点 B(0FA0，0300) 所定义的线性函数对 D0 的内容进行定标。这些点的坐标包含在 D100 ~ D103 中，结果将输出到 D200。

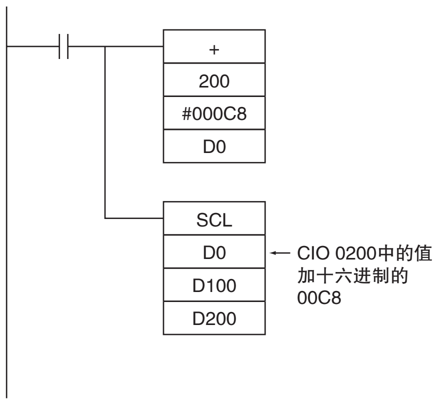


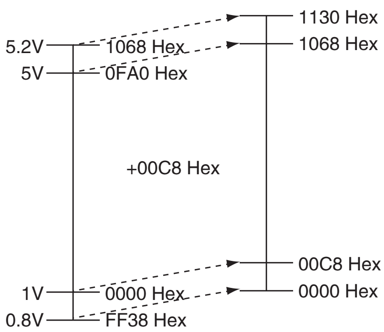


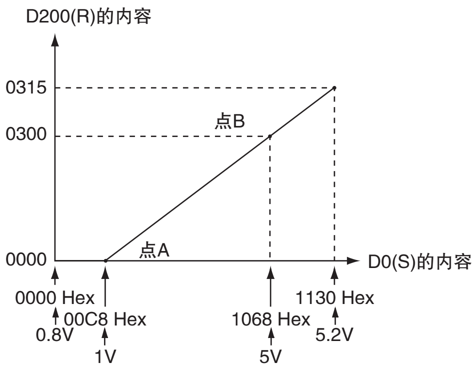
P1	: D100	0 0 0 0	Ar(BCD)
P1+1	: D101	0 0 0 0	As(BIN)
P1+2	: D102	0 3 0 0	Br(BCD)
P1+3	: D103	0 F A 0	Bs(BIN)

参考：

模拟量输入单元在输入 0.8 ~ 5.2V 的值时，实际上输入的是十六进制数 FF38 ~ 1068。但 SCL(194) 指令只能处理十六进制的 0000 ~ FFFF 之间的无符号二进制值，因而无法用 SCL(194) 指令来直接处理小于 1V(十六进制的 0000) 的带符号二进制值，即十六进制的 FF38 ~ FFFF。因此在实际应用中，必须在使用 SCL(194) 指令之前对所有的值加十六进制的 00C8，从而使得十六进制的 FF38 表示为十六进制的 0000，如下例所示。







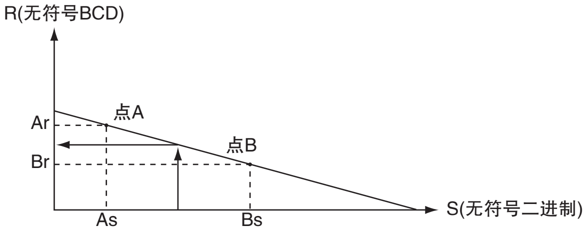
点A(00C8 Hex → 0000(BCD))
点B(1068 Hex → 0300(BCD))

P1	: D100	0 0 0 0	Ar(BCD)
P1+1	: D101	0 0 C 8	As(BIN)
P1+2	: D102	0 3 0 0	Br(BCD)
P1+3	: D103	1 0 6 8	Bs(BIN)

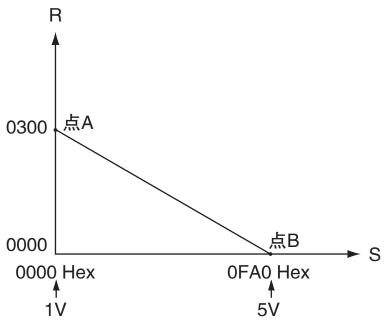
该例中，从十六进制的 0000 ~ 00C8 的值将被转换成负值。但 SCL(194) 指令只能输出从 0000 ~ 9999 的无符号 BCD 值，因此只要 D0.00 的内容在十六进制的 0000 ~ 00C8 之间，均会输出 0000 BCD。

反向定标

也可通过设定为 As<Bs 和 Ar>Br 来实现反向定标，该设定将产生如下关系。

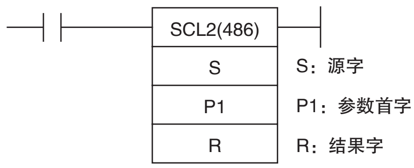


例如，可使用反向定标将 1 ~ 5V(十六进制的 0000 ~ 0FA0) 分别转换成 0300 ~ 0000，如下图所示。



SCL2

指令	助记符	变化	功能代码	功能
定标 2	SCL2	@SCL2	486	根据指定的线性函数，将带符号二进制数据转换成带符号 BCD 数据。定义线性函数时可输入一个偏移值。

	SCL2
符号	

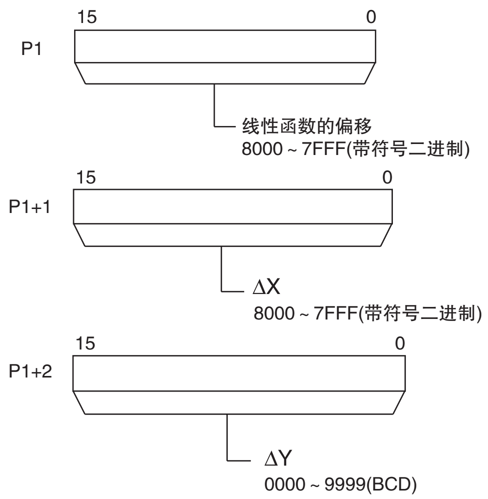
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	INT	1
P1	参数首字	WORD	3
R	结果字	WORD	1

P1：参数首字



注 P1 ~ P1+2 必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
P1											---					
R											OK					

条件标志

名称	标记	操作
出错标志	P_ER	· 当 C+1(ΔX) 的内容为 0000 时 ON。 · 当 C+2(ΔY) 的内容不是 BCD 时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
进位标志	P_CY	· 结果为负时 ON。 · 结果为 0 或正时 OFF。

功能

SCL2(486) 指令用于将源字 S 中包含的带符号二进制数据转换成带符号 BCD 数据 (该 BCD 数据包含绝对值，进位标志表示符号)，并根据由斜率 (ΔX， ΔY) 和偏移量所定义的线性函数将结果放入结果字 R 中。包含 ΔX，ΔY 和偏移量的第一个字的地址由第一个参数字 P1 指定。结果的符号由进位标志的状态来表示 (ON：负，OFF：正)。

下列公式用于转换。

$$R=\frac{\Delta Y}{\Delta X\text{的BCD转换值}}\times[(\text{S的BCD转换值})-(\text{偏移的BCD转换值})]$$

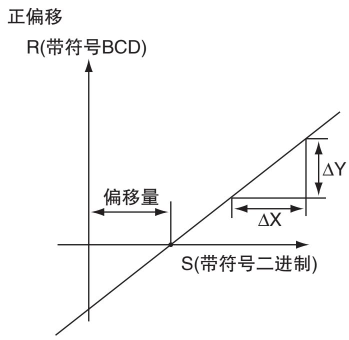
注 直线的斜率为ΔY/ΔX。

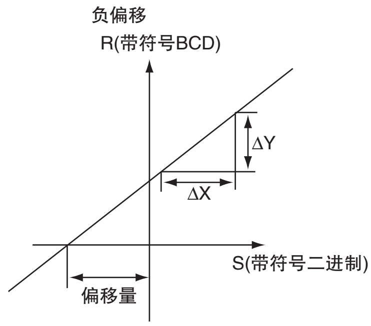
偏移和斜率可以是正值、0 或负值。采用负斜率可实现反向定标。

· 结果将四舍五入到最接近的整数。

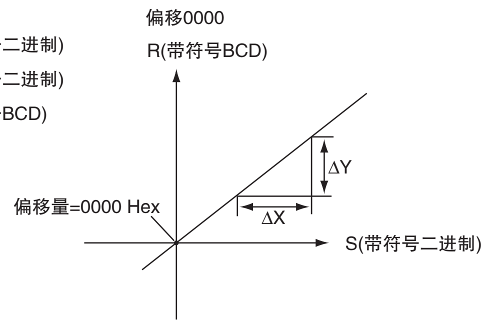
· R 中的结果将为转换后的 BCD 数的绝对值，符号则由进位标志来表示。因而结果可介于 –9999 ~ 9999 之间。

· 如果结果小于 –9999，则将输出 –9999 作为结果； 如果结果大于 9999，则将输出 9999 作为结果。





P1	偏移量	(带符号二进制)
P1+1	ΔY	(带符号二进制)
P1+2	ΔX	(带符号BCD)



提示

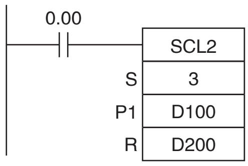
- SCL2(486) 指令可用于按照用户定义的定标参数，对来自模拟量输入单元的模拟信号转换值的结果进行定标。例如，如果有一个 1 ～ 5V 的模拟量输入单元的输入值以十六进制的 0000 ～ 0FA0 的形式输入到存储器中，则可使用 SCL2(486) 指令将该值定标为 –100↔200 ℃。
- SCL2(486) 指令将带符号二进制数转换成 BCD 数据，因此可直接处理 S 中的负值。另外，使用 R 中的定标结果和进位标志还可输出负的定标结果。

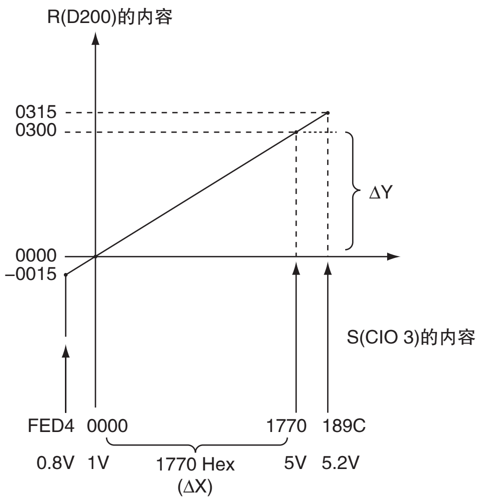
程序举例

● 将 1 ～ 5V 的模拟量输入定标为 0 ～ 300

下例中，假设要转换一个 1 ～ 5V 的模拟量信号，并作为十六进制数 0000 ～ 1770 输入到 CIO 3 中。使用 SCL2(486) 指令将 CIO 3 中的值转换 (定标) 为 0000 ～ 0300 BCD 之间的值。

当 CIO 0.00 为 ON 时，采用由 ΔX(1770)，ΔY(0300) 和偏移量 (0) 所定义的线性函数对 CIO 3 的内容进行定标。这些值包含在 D100 ～ D102 中，结果将输出到 D200。



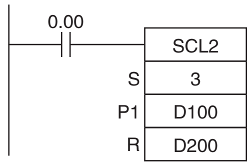


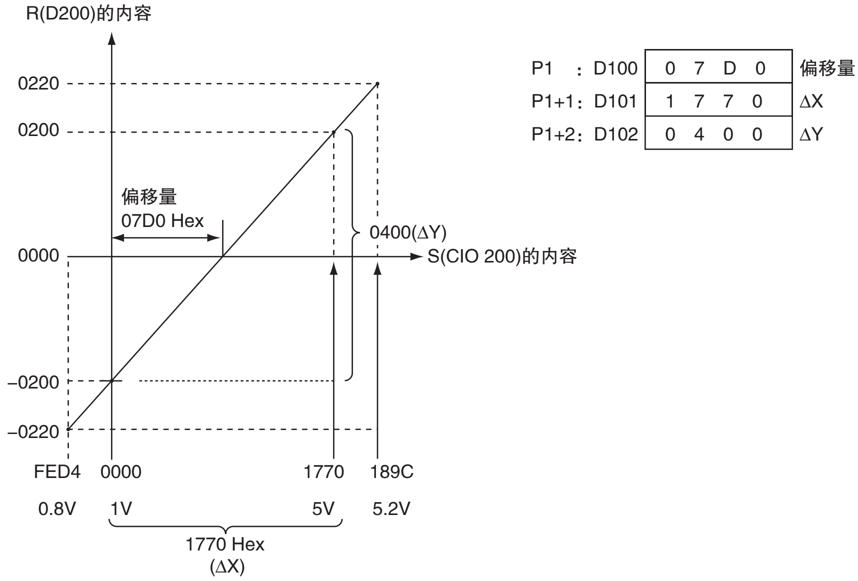
P1 ：D100	0 0 0 0	偏移量
P1+1：D101	1 7 7 0	ΔX
P1+2：D102	0 3 0 0	ΔY

● 将 1 ～ 5V 的模拟量输入定标为 –200 ～ 200

下例中，假设要转换一个 1 ～ 5V 的模拟量信号，并作为十六进制数 0000 ～ 1770 输入到 CIO 3 中。使用 SCL2(486) 指令将 CIO 3 中的值转换 (定标) 为 –0200 ～ 0200 BCD 之间的值。

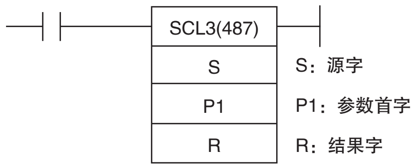
当 CIO 0.00 为 ON 时，采用由 ΔX(1770)，ΔY(0400) 和偏移量 (07D0) 所定义的线性函数对 CIO 3 的内容进行定标。这些值包含在 D100 ～ D102 中，结果将输出到 D200。





SCL3

指令	助记符	变化	功能代码	功能
定标 3	SCL3	@SCL3	487	根据指定的线性函数，将带符号 BCD 数据转换成带符号二进制数据。定义线性函数时可输入一个偏移值。

	SCL3
符号	

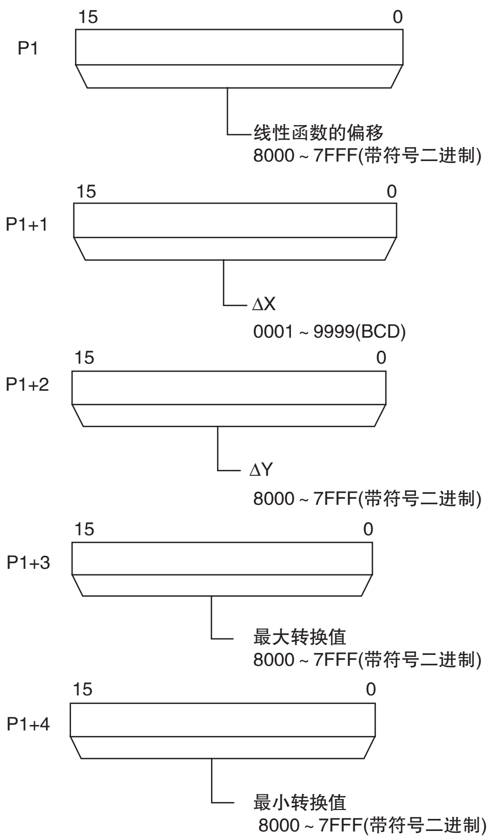
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	WORD	1
P1	参数首字	WORD	5
R	结果字	INT	1

P1：参数首字



注 P1 ～ P1+4 必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
P1	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· S 的内容不是 BCD 时 ON。 · 当 C+1(ΔX) 的内容不在 0001 ～ 9999 BCD 之间时 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 结果为 0 时 ON。 · 其它情况下 OFF。
负标志	P_N	· R(结果) 的 MSB 为 1 时 ON。 · 其它情况下 OFF。

功能

SCL3(487) 指令用于将源字 S 中包含的带符号 BCD 数据 (该 BCD 数据包含绝对值，进位标志表示符号) 转换成带符号二进制数据，并根据由斜率 (ΔX，ΔY) 和偏移量所定义的线性函数将结果放入结果字 R 中。同时还指定最大和最小转换值。包含 ΔX， ΔY、偏移量、最大转换值以及最小转换值的第一个字的地址由第一个参数字 P1 指定。

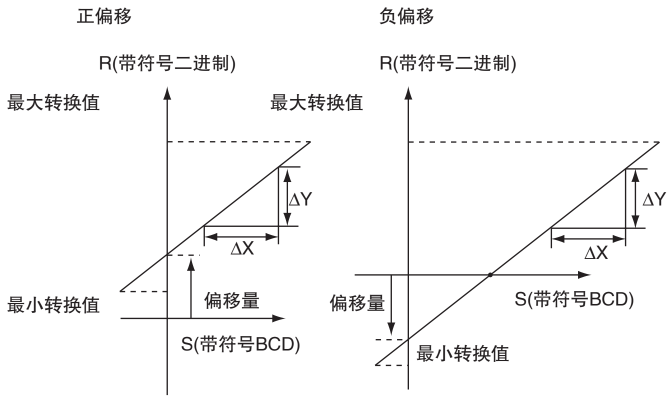
结果的符号由进位标志的状态来表示 (ON：负，OFF：正)。使用 STC(040) 和指令 CLC(041) 对进位标志置 ON 和 OFF。

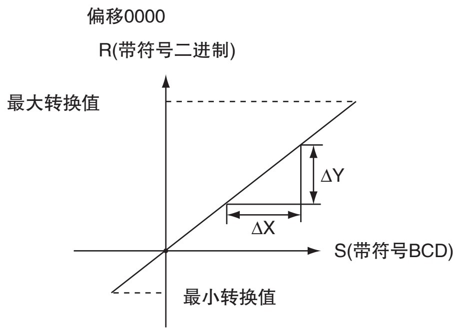
下列公式用于转换。

$$R=\frac{\Delta Y}{\Delta X\text{的二进制转换值}}\times((S\text{的二进制转换值})+(\text{偏移量}))$$

注 直线的斜率为ΔY/ΔX。

- 偏移和斜率可以是正值、0 或负值。采用负斜率可实现反向定标。
- 结果将四舍五入到最接近的整数。
- 将 S 中的源值作为 BCD 数的绝对值处理，符号则由进位标志来表示。因而源值可介于 –9999 ～ 9999 之间。
- 如果结果小于最小转换值，则将最小转换值作为结果输出。 如果结果大于最大转换值，则将最大转换值作为结果输出。



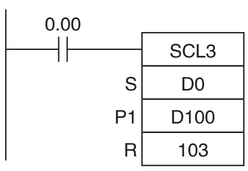


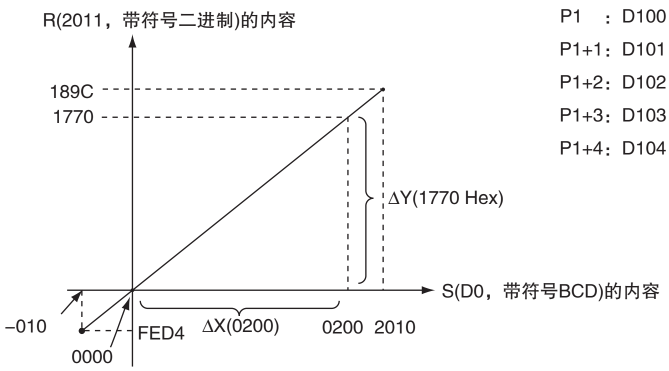
提示

SCL3(487) 指令用于将采用用户所定义标度的数据转换成用于模拟量输出单元的带符号二进制数据。例如，SCL3(487) 可将 0 ～ 200 ℃转换成 0000 ～ 1770(Hex)，并从模拟量输出单元输出模拟量输出信号 1 ～ 5V。

程序举例

当一个 0 ～ 200 的值被定标为模拟量信号 (例如 1 ～ 5V)，则将一个 0000 ～ 0200 的带符号 BCD 值转换 (定标) 成用于模拟量输出单元的 0000 ～ 1770 的带符号二进制值。下例中，当 CIO 0.00 变 ON 时，采用由 ΔX(0200)，ΔY(1770) 和偏移量 (0) 所定义的线性函数对 D0 的内容进行定标。这些值包含在 D100 ～ D102 中。D0 中的 BCD 值的符号由进位标志表示。结果输出到 CIO 103。

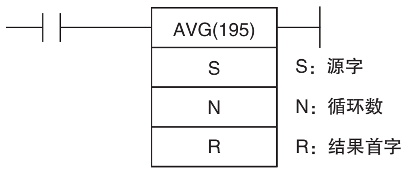




P1	：D100	0 0 0 0	偏移量
P1+1	：D101	0 2 0 0	ΔX
P1+2	：D102	1 7 7 0	ΔY
P1+3	：D103	1 8 9 C	最大转换值
P1+4	：D104	F E D 4	最小转换值

AVG

指令	助记符	变化	功能代码	功能
平均值	AVG	---	195	计算一个输入字在指定数目的循环中的平均值。

	AVG
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

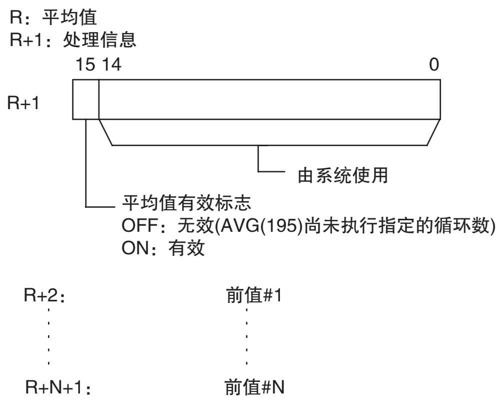
操作数

操作数	描述	数据类型	大小
S	源字	UINT	1
N	循环数	UINT	1
R	结果首字	UINT	可变

N：循环数

循环数必须介于十六进制的 0001 ～ 0040(1 ～ 64 个循环) 之间。

R：结果首字和 R+1：工作区首字



注 R ～ R+N+1 必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S，N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
R	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· N 的内容为 0 时 ON。 · 其它情况下 OFF。

功能

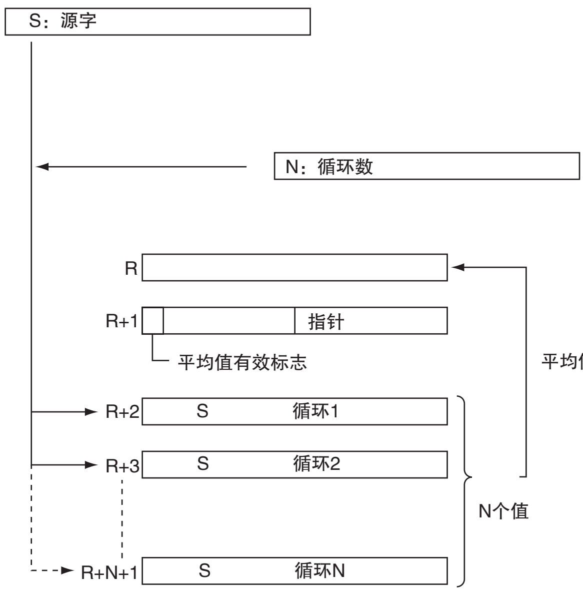
当执行条件为 ON 时，在前 N−1 个循环中，AVG(195) 指令将 S 的值按顺序写入从 R+2 开始的字中。每写入一个值时，前值指针 (R+1 的位 00 ~ 07) 均递增 1。将 S 的内容按原值输出到 R，且平均值标志 (R+1 的位 15) 将保持 OFF，直到写入第 N 个值。

将第 N 个值写入 R+N+1 之后，计算已写入的所有值的平均值，将平均值以无符号二进制值输出到 R，并对平均值标志 (R+1 的位 15) 置 ON。在后续所有循环中，R 的值按最新的 N 个 S 值更新。

N 的最大值为 64。如果指定了一个大于 64 的值，则将采用值 64 进行运算。

写入 N−1 个值之后，前值指针将复位到 0。

输出到 R 中的平均值将四舍五入到最接近的整数。



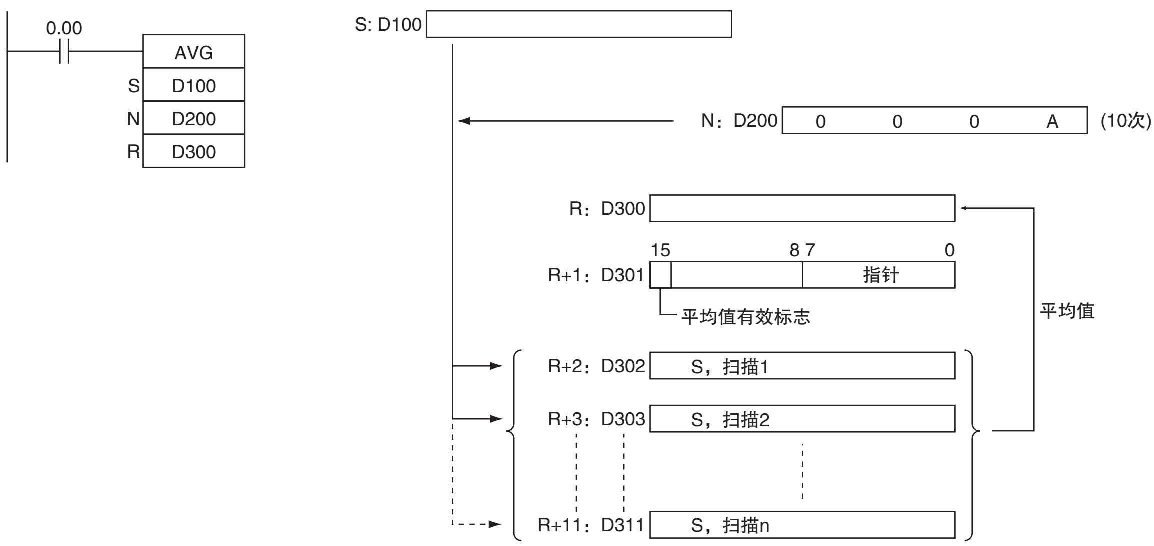
注意事项

每次当执行条件从 OFF 变为 ON 时，均会将处理信息 (R+1) 清为 0000。

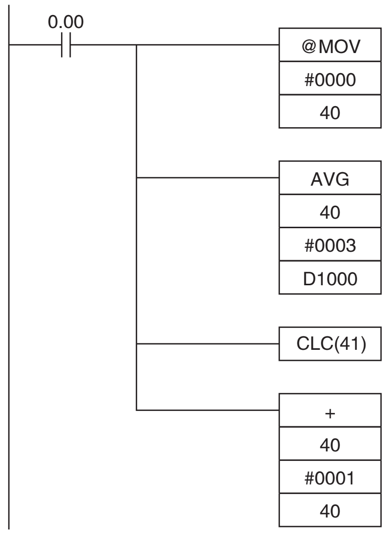
但在操作开始首次执行程序时，不会将处理信息 (R+1) 清为 0000。如果要在首次程序扫描时执行 AVG(195)，请清除程序的工作区首字。

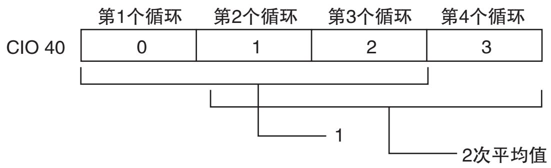
程序举例

下例中，当 CIO 0.00 为 ON 时，D100 中的内容将在每次扫描时存储一次，总扫描次数在 D200 中指定。内容将有序地存储在 D302 ~ D311 之间的 10 个字中。这十个字的内容的平均值将放入 D300 中，然后将 D301 的位 15 置 ON。



- 下例中，将 CIO 40 的内容设定为 #0000，并在每次循环时递增 1。
- 在前两个循环中，AVG(195) 将 CIO 40 的内容移入 D1002 和 D1003 中。D1001 的内容也将改变 (用于确认 AVG(195) 的结果已改变)。
- 在第三个及第三个以后的循环中，AVG(195) 计算 D1002 ～ D1004 的内容的平均值，并将该平均值写入 D1000 中。



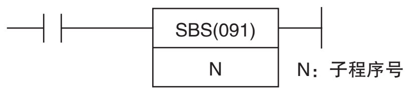


D1000	0	1	1	2	平均值
D1001	1	2	8000	8001	指针
D1002	0	0	0	3	IR 40的3个前值
D1003	---	1	1	1	
D1004	---	---	2	2	

子程序指令

SBS

指令	助记符	变化	功能代码	功能
子程序调用	SBS	@SBS	091	调用指定子程序号的子程序并执行该程序。

符号	SBS
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	子程序号	---	1

N：子程序号

指定介于十进制的 0 ～ 127 之间的子程序号。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---

组合使用的指令

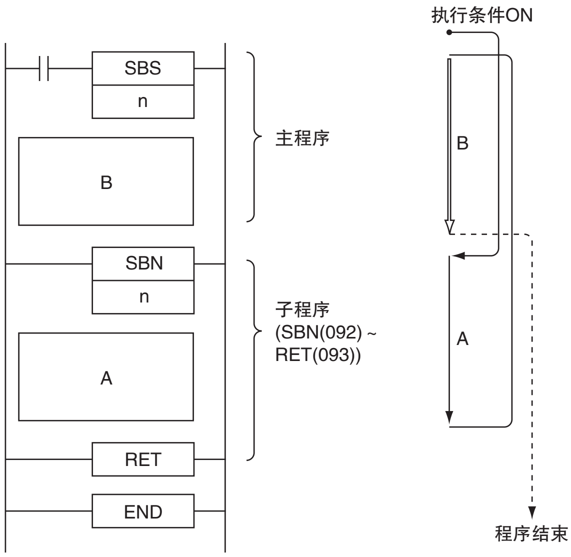
SBN(子程序入口) 指令和 RET(子程序返回) 指令

条件标志

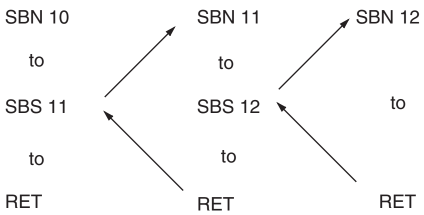
名称	标记	操作
出错标志	P_ER	· 当嵌套超过 16 层时 ON。 · 当指定的子程序号不存在时 ON。 · 当某个子程序调用自身时 ON。 · 当调用某个正在执行的子程序时 ON。 · 当指定的子程序在当前任务中未定义时为 ON。 · 其它情况下 OFF。

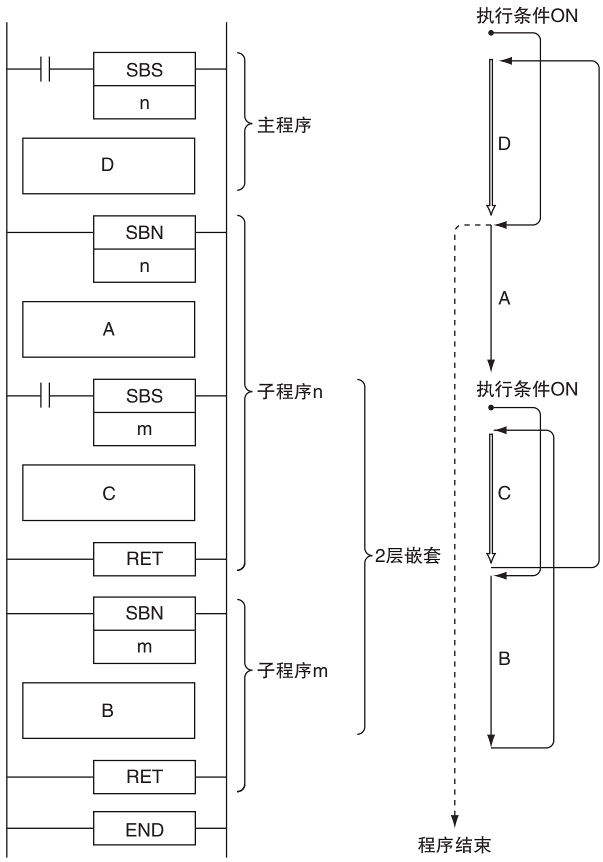
功能

SBS(091) 指令调用指定子程序号的子程序。子程序为 SBN(092) ～ RET(093) 之间的程序段。当该子程序结束时，程序从 SBS(091) 后的下一条指令开始继续执行。一个子程序在一个程序中可被调用多次。



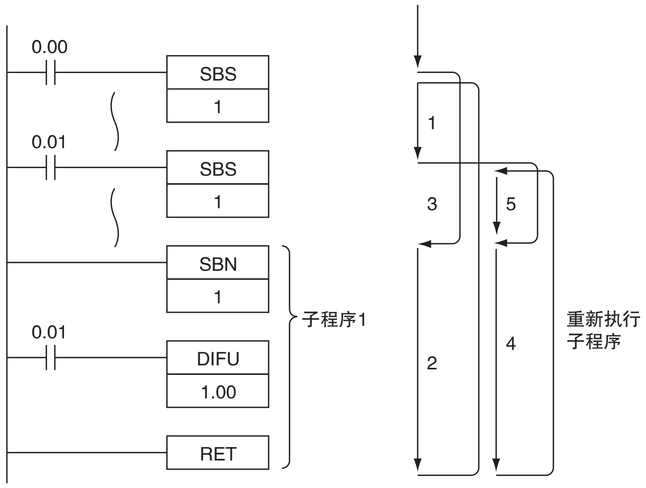
子程序最多可嵌套 16 层。嵌套是指在一个子程序内调用另一个子程序。如下例所示，子程序嵌套了 3 层。

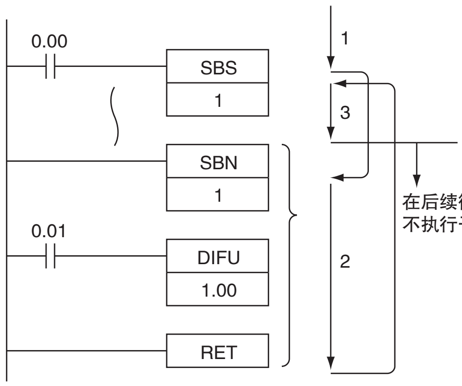




注意事项

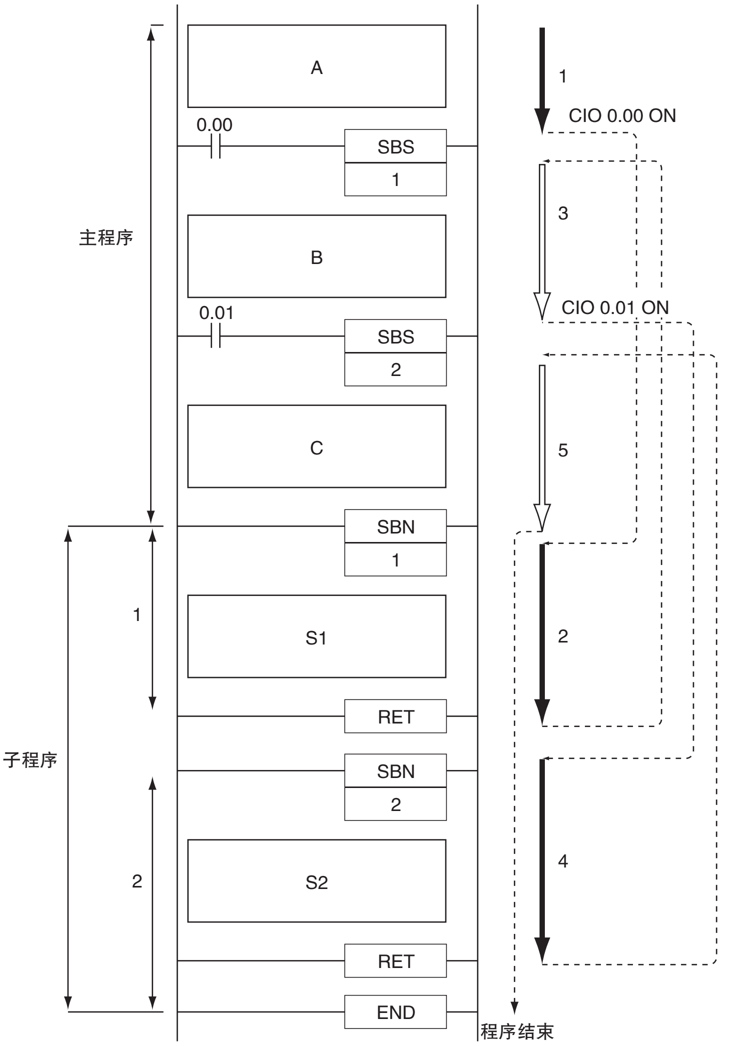
- 子程序号必须对各个子程序唯一。 同一个子程序号不能用于一个以上的子程序。
- 各个子程序的子程序号必须是独一无二的。请勿将同一个子程序号用于一个以上的子程序。
- 当在子程序中使用微分指令(DIFU(013)、DIFD(014) 或上升沿 / 下降沿微分指令 ）时，请遵守以下注意事项。
  - 如果一个子程序在同一个循环中执行的次数超过一次，则微分指令在子程序中的操作无法预测。下例中，当 CIO 0.00 为 ON 时执行子程序 1，而当 CIO 0.01 从 OFF → ON 时 CIO 1.00 将被 DIFU(013) 指令置 ON。如果在同一循环中 CIO 0.01 为 ON，则将再次执行子程序 1，但这次执行时，DIFU(013) 将不检查 CIO 0.01 的状态而对 CIO 1.00 置 OFF。
  - 相比之下，如果执行了微分指令 (UP、DOWN、DIFU(013) 或 DIFD(014)) 并对输出置ON,但不对同一个子程序调用第二次，则微分指令将使其保持 ON 状态。
  - 右例中，如果 CIO 0.00 为 ON，则执行子程序 1。当 CIO 0.01 从 OFF → ON 时，DIFU(013) 指令将对 CIO 1.00 置 ON。如果在后续循环中 CIO 0.00 为 OFF，则不会再次执行子程序 1，并且输出 CIO 1.00 将保持 ON。
  - SBS(091) 指令在由 IL(002) 和 ILC(003) 互锁的程序段中时将被作为 NOP(000) 处理。





程序举例

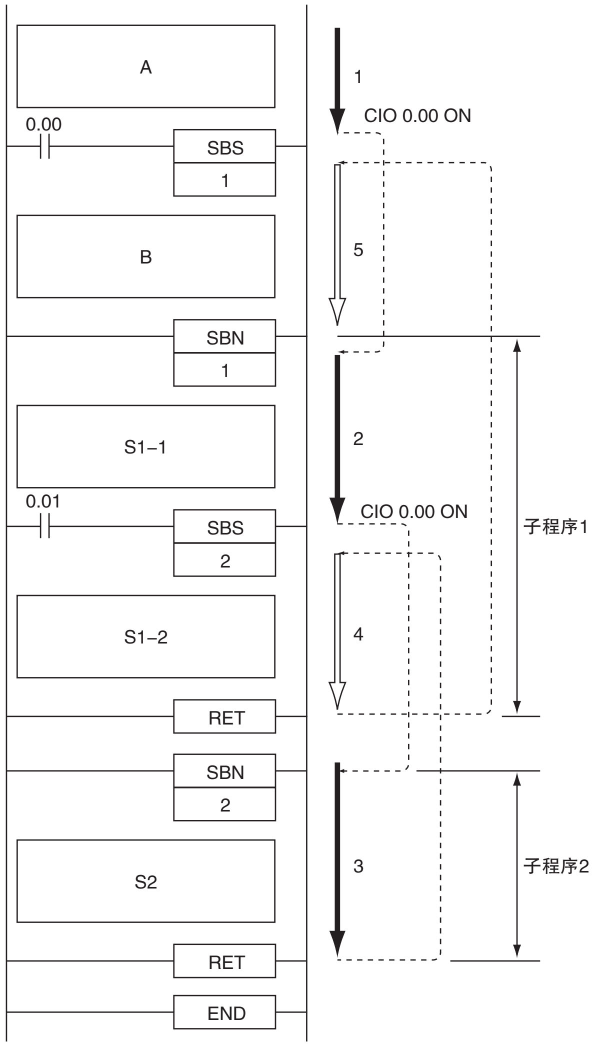
● 顺序（无嵌套）子程序



0.00	0.01	执行顺序
ON	ON	A→S1→B→S2→C
ON	OFF	A→S1→B→C
OFF	ON	A→B→S2→C
OFF	OFF	A→B→C

本例中，当 CIO 0.00 为 ON 时，将执行子程序 1，且程序的执行将返回 SBS(091) 1 之后的下一条指令。当 CIO 0.01 为 ON 时，将执行子程序 2，且程序的执行将返回 SBS(091) 2 之后的下一条指令。

● 嵌套子程序

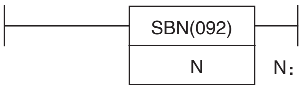


0.00	0.01	执行顺序
ON	ON	A→S1–1→S2→S1–2→B
ON	OFF	A→S1–1→S1–2→B
OFF	ON	A→B
OFF	OFF	A→B

本例中，当 CIO 0.00 为 ON 时，将执行子程序 1。当 CIO 0.01 为 ON 时，将从子程序 1 内部执行子程序 2，且在子程序 2 结束后，程序的执行将返回 SBS(091) 2 之后的下一条指令。继续执行子程序 1，且在子程序 1 结束后，程序的执行将返回 SBS(091) 1 之后的下一条指令。

SBN/RET

指令	助记符	变化	功能代码	功能
子程序入口	SBN	---	092	表示指定子程序号的子程序的开始。
子程序返回	RET	---	093	表示子程序的结束。

	SBN	RET
符号		

适用程序区

● SBN

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	---	---	OK

● RET

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	---	OK	OK

操作数

操作数	描述	数据类型	大小
		SBN	
N	子程序号	---	1

● SBN

N：子程序号

指定介于十进制的 0 ～ 127 之间的子程序号。

● 区域类型

区域		字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
		CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
SBN	N	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---

组合使用的指令

SBS(子程序调用) 指令

条件标志

● SBN/RET

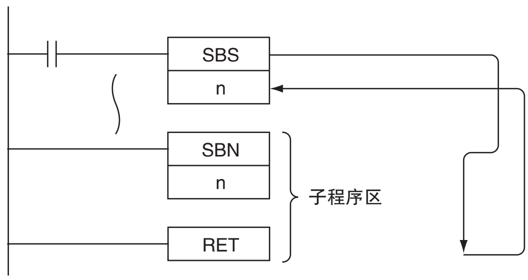
该指令不影响任何标志。

功能

● SBN

SBN(092)指令表示指定子程序号的子程序的开始。子程序的结束通过RET(093)指令来表示。

从第一条SBN(092)指令开始的程序区为子程序区。仅当通过SBS(091)指令调用某个子程序时，才会执行该子程序。

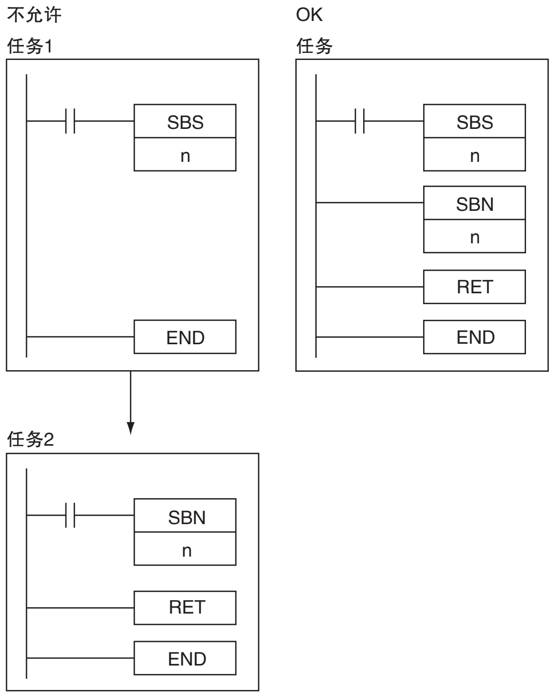


● RET

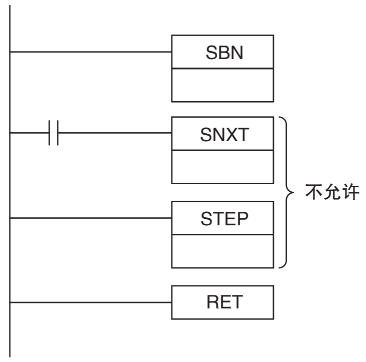
当程序执行到 RET(093) 指令时，将自动返回至调用子程序的 SBS(091) 指令的下一条指令。

注意事项

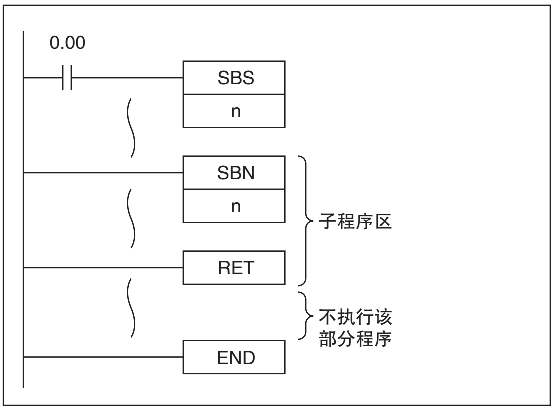
・ 将子程序的程序区 (SBN(092) ～ RET(093)) 放入与相同编号的 SBS(091) 指令相同的任务中。无法调用其它任务中的子程序。



・子程序中无法使用步指令 STEP(008) 和 SNXT(009)。



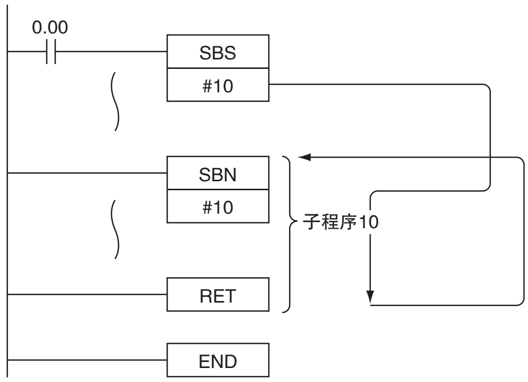
・应将子程序放在各任务的程序中的主程序之后和END(001)指令之前。如果有部分主程序放在子程序区之后，则该程序段将被忽略。



注 CX-Programmer的子程序号N的输入方法有所不同。
在 CX-Programmer 上应输入 #0 ～ #127。

程序举例

下例中，当 CIO 0.00 为 ON 时，将执行子程序10，然后程序的执行将返回至调用该子程序的SBS(091)或MCRO(099)之后的下一条指令。



中断控制指令

CP1E/CP2E CPU 单元支持下列类型的中断。

类型	执行条件	设定步骤
I/O 中断	来自 CPU 机架内上的内置输入单元的中断输入变为 ON/OFF。	使用 MSKS 指令来分配来自 CPU 机架上的中断输入单元的输入。
定时中断	定时执行 (以固定间隔执行)	使用 MSKS 指令来设定中断间隔。

中断控制指令概要

● 设置中断屏蔽：MSKS(690)

当 PLC 进入 RUN 模式时，对 I/O 中断任务和定时中断任务进行屏蔽 (禁止)。MSKS(690) 指令可用于对 I/O 中断进行去屏蔽或屏蔽，以及为定时中断设定时间间隔。

● 清除中断：CLI(691)

CLI(691) 指令可清除或保留已记录的中断输入 (针对 I/O 中断)，并设定距离第一次定时中断 (针对定时中断) 的时间。此外，还可清除或保留已记录的高速计数器中断。

● 禁止中断：DI(693)

DI(693) 指令禁止执行所有中断任务。

● 允许中断：EI(694)

EI(694) 指令允许执行所有中断任务。

使用中断任务的注意事项

● 所有中断的注意事项

· 当多个中断同时执行时，其执行顺序如下所示：
I/O 中断 > 定时中断

注 “A>B”表示 A 的优先级高于 B。当处于同一个中断级别时，编号小的任务的优先级高于编号大的任务。

● I/O 中断的注意事项

· 中断任务仅支持 CP1E/CP2E CPU 单元的内置输入。
· 请使用 CPU 机架上的中断输入 (位 0ch02 ～ 0ch09)。若使用其它输入，将不执行 I/O 中断。
· 清除中断屏蔽时，将清除已检测到的所有中断输入。
· 可记录的 I/O 中断输入数没有限制，但每个 I/O 中断号仅可对应一个中断。另外，已记录的中断将一直保留到相应的中断任务结束为止，因此如果在执行相应的中断任务时接收到新的中断输入，则新的中断输入将被忽略。

● 定时中断的注意事项

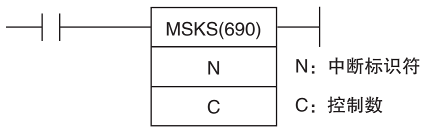
· 请确保设定时间间隔比执行定时中断任务所需的时间要长。
· 若要精确控制距离第一次定时中断的时间和中断间隔，可在紧靠 MSKS(690) 指令之前编写 CLI(691) 指令来设定距离第一次定时中断的时间。但如果使用 MSKS(690) 指令重启定时中断，则即使不使用 CLI(691) 指令，也可精确控制距离第一次定时中断的时间。
· 定时中断的时间单位始终为 0.1ms。

相关存储区字

名称	地址	操作
中断任务的最长处理时间	A440	中断任务的最长处理时间作为二进制数据以 0.1ms 为单位进行存储，并在操作开始时被清除。
处理时间最长的中断任务	A441	将处理时间最长的中断任务号作为二进制数据存储。此处，8000 ~ 800F Hex 与任务号 00 ~ 0F Hex 对应。 当操作开始后发生第一次中断时，A441.15 将变 ON。后续中断任务的最长处理时间将作为十六进制数据存储存储在最右边的两个数位中，并在操作开始时被清除。

MSKS

指令	助记符	变化	功能代码	功能
设置中断屏蔽	MSKS	@MSKS	690	控制是否执行 I/O 中断任务和定时中断任务。

符号	MSKS
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	中断标识符	---	1
C	控制数据	UINT	1

(1) I/O 中断任务

操作数	内容	
	允许 / 禁止中断输入设定时	为中断输入指定上升沿 / 下降沿微分时
	I/O 中断号	I/O 中断号
N	102：中断输入 2(中断任务 2) 103：中断输入 3(中断任务 3) 104：中断输入 4(中断任务 4) 105：中断输入 5(中断任务 5) 106：中断输入 6(中断任务 6) (无法用于 CP1E-E10D □ - □) 107：中断输入 7(中断任务 7) (无法用于 CP1E-E10D □ - □) 108：中断输入 8(中断任务 8) (仅可用于 CP2E-N20/30/40/60D □ - □) 109：中断输入 9(中断任务 9) (仅可用于 CP2E-N20/30/40/60D □ - □)	112：中断输入 2(中断任务 2) 113：中断输入 3(中断任务 3) 114：中断输入 4(中断任务 4) 115：中断输入 5(中断任务 5) 116：中断输入 6(中断任务 6) (无法用于 CP1E-E10D □ - □) 117：中断输入 7(中断任务 7) (无法用于 CP1E-E10D □ - □) 118：中断输入 8(中断任务 8) (仅可用于 CP2E-N20/30/40/60D □ - □) 119：中断输入 9(中断任务 9) (仅可用于 CP2E-N20/30/40/60D □ - □)
C	中断屏蔽	中断屏蔽
	0000 Hex：允许 (不屏蔽) 中断 (直接模式) 0001 Hex：禁止 (屏蔽) 中断 (直接模式)	0000 Hex：上升沿微分 (检测上升沿) 0001 Hex：下降沿微分 (检测下降沿)

注 改变上升沿 / 下降沿微分设定时，所有已检测到的中断输入均会被清除。

(2) 复位和启动定时中断

操作数	内容	
N	定时中断号	
	4 或 14：定时中断 0(中断任务 1)	
C	定时中断时间单位	定时中断的设定时间
	任何时间单位设定	十进制的 0(0000 Hex)： 禁止中断 (停止内部定时器)
	0.1ms	十进制的 10 ～ 9,999(000A ～ 270F Hex)： 允许中断 (使内部定时器的值复位，然后以 1.0 ～ 999.9ms 之间的某个中断间隔启动该定时器。) **注** 不能使用 0001 ～ 000A 的设定，否则将产生错误。

● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>N</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td rowspan="2">OK</td><td>---</td><td rowspan="2">---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>C</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	当 N 不在指定范围内时 ON。 指定 I/O 中断时的出错情况： ・若 C 不在指定的范围内，则出错标志将变 ON。 指定定时中断时的出错情况： ・若 C 不在十进制的 10 ～ 9,999(000A ～ 270F Hex) 范围内时，出错标志将变 ON。 其它情况下 OFF。

功能

当程序开始执行时，使产生 I/O 中断任务的中断输入被屏蔽 (禁止)，并使创建定时器中断 (该中断可产生定时中断任务) 的内部定时器停止计时。

请使用 MSKS(690) 指令以允许 I/O 中断和定时器中断，从而可执行相应的中断任务。

N 的值指定中断任务和将执行的处理的种类。

(1) N=102 ～ 109：允许 / 禁止 I/O 中断任务的中断输入

- 根据 C 位状态来允许或禁止由 N 指定的中断输入。使用该功能时，MSKS(690) 指令可控制是否执行各个任务。
- 允许中断输入时，到该点为止的所有已检测到的中断均会被清除。

(1) N=112 ～ 119：指定中断输入的微分设定

- 根据 C 中位的状态来指定由 N 所指定的中断输入是上升沿微分还是下降沿微分。
- 将微分指令规定与允许 / 禁止功能结合使用。如果不执行 MSKS(690) 指令来指定上升沿或下降沿微分，则中断输入为上升沿微分 (默认设定)。
- 当执行 MSKS(690) 指令来指定中断输入的上升沿或下降沿微分时，到该点为止的所有已检测到的中断都将被清除。

(1) N=4 或 14：复位和重启定时中断任务

- 为指定的定时中断任务 (由 N 指定) 设定时间间隔 (由 C 指定)，复位内部定时器的 PV，并启动内部定时器。由于内部定时器的 PV 被复位，该功能可从执行 MSKS(690) 指令到开始第一次中断之间保持适当的间隔。

提示

将最长的中断任务处理时间存储在 A440 中 (中断任务最长处理时间)。与此同时，将中断任务处理时间最长的中断任务的任务号存储在 A441 中 (处理时间最长的中断任务)。

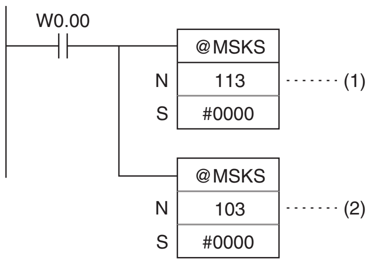
注意事项

- 请确保设定时间间隔比执行定时中断任务所需的时间要长。
- 若要精确控制距离第一次定时中断的时间和中断间隔，可在紧靠 MSKS(690) 指令之前编写 CLI(691) 指令来设定距离第一次定时中断的时间。但如果使用 MSKS(690) 指令重启定时中断，则即使不使用 CLI(691) 指令，也可精确控制距离第一次定时中断的时间。
- 在执行定时中断期间，无法改变定时中断的设定时间。请在使用 MSKS 指令设定为禁止中断 (停止内部定时器) 后再改变定时中断的设定时间。

程序举例

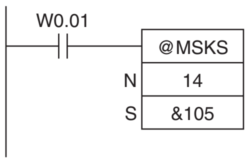
● 输入中断举例

下例中，当 W0.00 变 ON 时，第一条 MSKS(690)(1) 指定在中断输入变 ON 时产生输入中断 (针对中断输入 3)，而第二条 MSKS(690)(2) 则对中断进行去屏蔽。



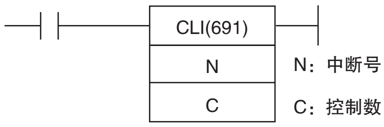
● 定时中断举例

下例中，当 W0.01 变 ON 时，MSKS(690) 指令将定时中断 0 的定时中断间隔设定为 10.5ms(假设在 PLC 设置的单位设定为 0.1ms 时)，并对内部定时器进行复位，然后再启动内部定时器。



CLI

指令	助记符	变化	功能代码	功能
清除中断	CLI	@CLI	691	清除 / 保留已记录的中断输入，为定时中断任务设定距离第一次定时中断的时间。

符号	CLI
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	中断号	---	1
C	控制数据	UINT	1

(1) 清除 / 保留 I/O 中断任务的已记录中断输入

操作数	内容
N	**中断输入号**
	102：中断输入 2(中断任务 2) 103：中断输入 3(中断任务 3) 104：中断输入 4(中断任务 4) 105：中断输入 5(中断任务 5) 106：中断输入 6(中断任务 6)(无法用于 CP1E-E10D □ - □) 107：中断输入 7(中断任务 7)(无法用于 CP1E-E10D □ - □) 108：中断输入 8(中断任务 8)(仅可用于 CP2E-N20/30/40/60D □ - □) 109：中断输入 9(中断任务 9)(仅可用于 CP2E-N20/30/40/60D □ - □)
C	**已记录的中断**
	0000 Hex：保留已记录的中断 0001 Hex：清除已记录的中断

(2) 设定距离第一次定时中断的时间

操作数	内容	
N	**定时中断号**	
	4：中断任务 0(中断任务 1)	
C	**定时中断时间单位 (在 PLC 设置中进行设定)**	**定时中断的设定时间**
	0.1ms	十进制的 10 ~ 9,999(000A ~ 270F Hex)： 将距离第一次中断的时间设定到 1.0 ~ 999.9ms 之间。

(3) 清除 / 保留高速计数器中断

操作数	内容
	高速计数器输入
N	10：高速计数器输入 0 11：高速计数器输入 1 12：高速计数器输入 2 13：高速计数器输入 3 14：高速计数器输入 4 15：高速计数器输入 5(无法用于 CP1E-E10D □ - □)
	已记录的中断
C	0000 Hex：保留已记录的中断 0001 Hex：清除已记录的中断

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
C	OK	OK	OK	OK	OK	OK	OK	OK	OK		OK		OK			

条件标志

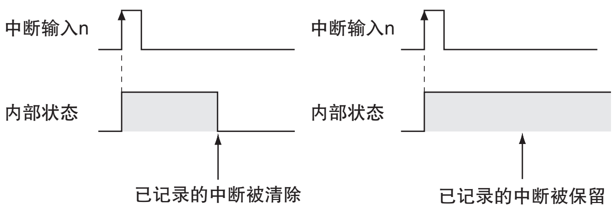
名称	标记	操作
出错标志	P_ER	· 当 N 不在指定范围内时 ON。 · 当 C 不为 0000 或 0001 Hex 时 O(对于 I/O 中断或高速计数器中断) · 对于定时中断，当 C 不在十进制的 10 ～ 9,999(000A ～ 270F Hex) 的范围内时 ON。 · 其它情况下 OFF。

功能

视 N 值而定，CLI(691) 指令将清除指定的已记录 I/O 中断 , 设定执行第一次定时中断之前的时间 , 或清除指定的已记录高速计数器中断。

(1) N=102 ～ 109：清除中断输入

CLI(691) 指令将在 C 的相应位为 ON 和 OFF 时，分别清除和保留由 N 指定的已记录的中断输入。

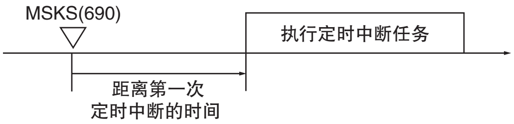


如果在执行某个 I/O 中断任务时接收到另一个中断号的中断输入，则将在内部记录该中断号。然后按照优先级高低 (从最小号到最大号) 执行已记录的 I/O 中断。

若要在执行中断任务时忽略中断输入，则需在执行中断任务前使用 CLI(691) 指令清除已记录的中断。

(2) N=4：设定距离第一次定时中断任务的时间

当 N 为 4 时，C 的内容指定距离第一次定时中断任务的时间间隔。



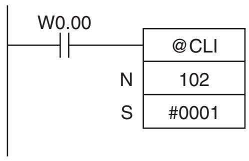
(3) N=10 或 15：清除高速计数器中断

当N为10或15时，CLI(691)指令清除或保留由N指定的已记录的高速计数器中断(目标值比较)。

程序举例

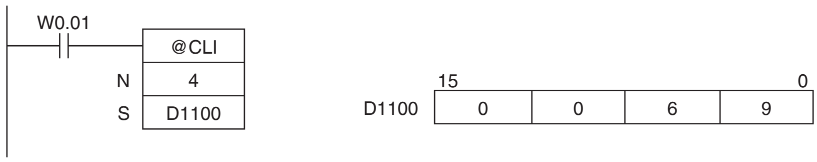
● 输入中断举例

下例中，当 W0.00 变为 ON 时，CLI(691) 指令清除保存在输入中断 2 中的所有中断。



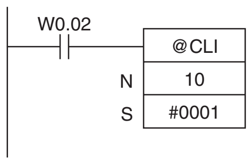
● 第一次定时中断举例

下例中，当 W0.01 变为 ON 时，CLI(691) 指令将距离第一次定时中断的时间设定为 10.5ms(十六进制的 0069= 十进制的 105)。



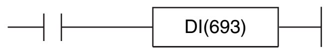
● 高速计数器中断举例

下例中，当 W0.02 变为 ON 时，CLI(691) 指令清除保存在高速计数器中断 0 中的所有中断。



DI

指令	助记符	变化	功能代码	功能
禁止中断	DI	@DI	693	禁止执行所有中断任务(除电源OFF中断除外)。

符号	DI
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

条件标志

名称	标记	操作
出错标志	P_ER	・当从某个中断任务执行 DI(693) 指令时 ON。 ・其它情况下 OFF。

功能

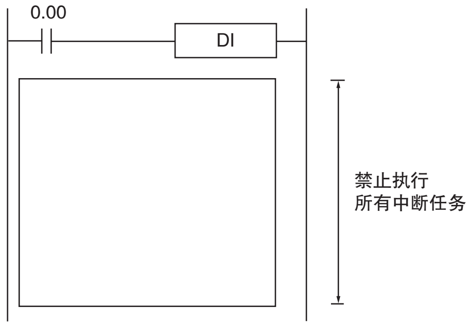
从主程序执行 DI(693) 指令，以临时禁止所有中断任务 (I/O 中断和定时中断)。

注意事项

所有中断任务均保持禁用状态，直到执行 EI(694) 指令为止。

无法从某个中断任务执行 DI(693) 指令。

程序举例



下例中，当 CIO 0.00 为 ON 时，DI(693) 指令将禁止所有中断任务。

EI

指令	助记符	变化	功能代码	功能
允许中断	EI	---	694	允许执行被 DI(693) 指令所禁止的所有中断任务。

符号	EI
	EI(694)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当从某个中断任务执行 EI(694)) 指令时 ON。 · 其它情况下 OFF。

功能

· 从主程序执行 EI(694) 指令，以临时允许所有被 DI(693) 指令所禁止的中断任务。DI(693) 禁止所有中断 (I/O 中断和定时中断)。

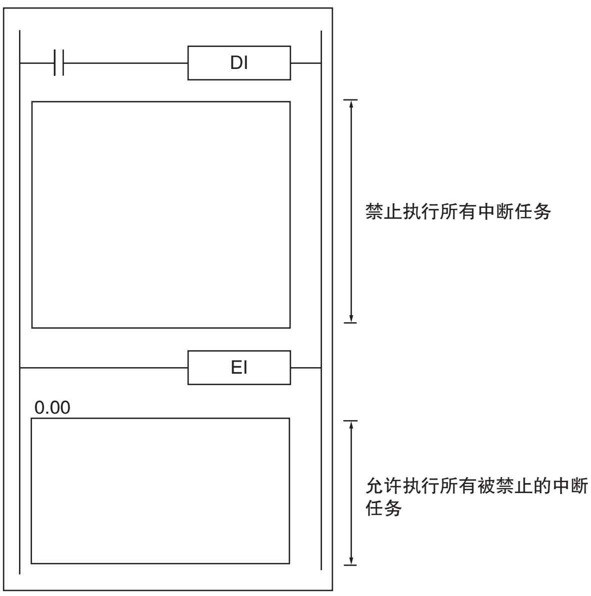
注意事项

· EI(694) 指令不需要执行条件。该指令将始终执行 (执行条件始终为 ON)。

· EI(694) 指令允许执行被 DI(693) 指令所禁止的中断任务。该指令无法对尚未被 MSKS(690) 指令去屏蔽的 I/O 中断去屏蔽，或者设定尚未由 MSKS(690) 指令所设定的定时中断。

· 无法从某个中断任务执行 EI(694) 指令。

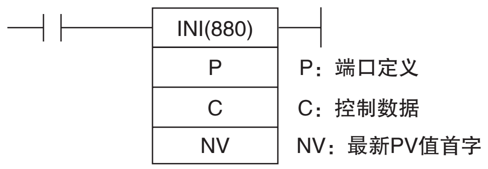
程序举例



高速计数器 / 脉冲输出指令

INI

指令	助记符	变化	功能代码	功能
模式控制	INI	@INI	880	INI(880) 可用于执行以下操作： ・启动与高速计数器比较表的比较 ・停止与高速计数器比较表的比较 ・改变高速计数器的 PV 值 ・改变计数器模式下中断输入的 PV 值 ・改变脉冲输出的 PV 值 (原始值固定为 0) ・停止脉冲输出 ・减速停止直线插补

	INI
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
P	端口定义	WORD	1
C	控制数据	UINT	1
NV	最新 PV 值首字	DWORD	2

P：端口定义

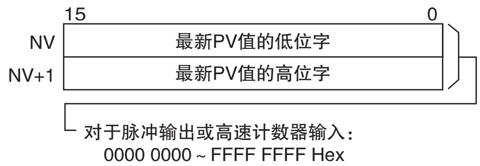
P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ – □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ – □)
0010 Hex	高速计数器 0
0011 Hex	高速计数器 1
0012 Hex	高速计数器 2
0013 Hex	高速计数器 3
0014 Hex	高速计数器 4
0015 Hex	高速计数器 5(无法用于 CP1E-E10D □ – □)
0030 Hex	直线插补 0(仅可用于 CP2E N □□型 CPU 单元)
0031 Hex	直线插补 1(仅可用于 CP2E-N30/40/60D □ – □)
1000 Hex	PWM(891) 输出 0

C：控制数据

C	INI(880) 功能
0000 Hex	启动比较
0001 Hex	停止比较
0002 Hex	改变 PV 值
0003 Hex	停止脉冲输出
0004 Hex	减速停止直线插补 (仅可用于 CP2E N □□型 CPU 单元)

NV：最新 PV 值首字

如果 C 为 0002 Hex(即改变 PV 值时)，NV 和 NV+1 包含最新的 PV 值。当 C 不等于 0002 Hex 时，NV 和 NV+1 中的任何值均被忽略。



● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>P，C</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>NV</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P、C 或 NV 的指定范围时 ON。 · 当 P 和 C 的组合不允许时 ON。 · 当未注册比较表但指定了启动比较时 ON。 · 当为某个当前正在输出脉冲的端口指定了一个新的 PV 值时 ON。 · 当为某个未被指定用于高速计数器的端口指定了改变高速计数器的 PV 值时 ON。 · 当在高速计数器的中断任务中执行了 INI(880) 指令并在执行 CTBL(882) 指令的过程中发生了中断时 ON。 · 其它情况下 OFF。

功能

INI(880) 对在 P 中指定的端口执行在 C 中指定的操作。各种操作和各个端口可能的组合如下表所示。

P：端口定义	C：控制数据				
	0000 Hex：启动比较	0001 Hex：停止比较	0002 Hex：改变 PV 值	0003 Hex：停止脉冲输出	0004 Hex：减速停止
0000 ~ 0003 Hex：脉冲输出	不允许	不允许	OK	OK	不允许
0010 ~ 0015 Hex：高速计数器输入	OK	OK	OK	不允许	不允许
0030 或 0031 Hex：直线插补	不允许	不允许	不允许	OK	OK
1000 Hex：PWM(891) 输出	不允许	不允许	不允许	OK	不允许

● 启动比较 (C=0000 Hex)

如果 C 等于 0000 Hex，则 INI(880) 指令将启动高速计数器的 PV 值与通过 CTBL(882) 指令所注册的比较表的比较。

注 必须事先通过 CTBL(882) 指令来注册目标值比较表。如果在未注册比较表的情况下执行 INI(880) 指令，则出错标志将置 ON。

- **停止比较 (C=0001 Hex)**

如果 C 等于 0001 Hex，则 INI(880) 指令将停止高速计数器的 PV 值与通过 CTBL(882) 指令所注册的比较表的比较。

- **改变 PV 值 (C=0002 Hex)**

<table>
<tr><th colspan="3">端口与模式</th><th>操作</th><th>设定范围</th></tr>
<tr><td colspan="3">脉冲输出 (P=0000 ~ 0003 Hex)</td><td>脉冲输出的当前值被更改。最新值在 NV 和 NV+1 中指定。
注 仅当停止了脉冲输出时才可执行该指令。如果在脉冲输出期间执行该指令，则将发生错误。</td><td>8000 0000 ~ 7FFF FFFF Hex
(−2,147,483,648 ~ 2,147,483,647)</td></tr>
<tr><td rowspan="3">高速计数器输入 (P=0010 ~ 0015 Hex)</td><td rowspan="2">线性模式</td><td>微分输入、增量 / 减量脉冲或脉冲 + 方向输入</td><td rowspan="3">高速计数器的当前值被更改。最新值在 NV 和 NV+1 中指定。
注 如果未为高速计数器设定指定的端口，则指令将发生错误。</td><td>8000 0000 ~ 7FFF FFFF Hex
(−2,147,483,648 ~ 2,147,483,647)</td></tr>
<tr><td>增量脉冲输入</td><td>0000 0000 ~ FFFF FFFF Hex
(0 ~ 4,294,967,295)</td></tr>
<tr><td colspan="2">环形模式</td><td>0000 0000 ~ FFFF FFFF Hex
(0 ~ 4,294,967,295)</td></tr>
</table>

- **停止脉冲输出 (P=0000 ~ 0003、0030、0031 或 1000 Hex 且 C=0003 Hex)**

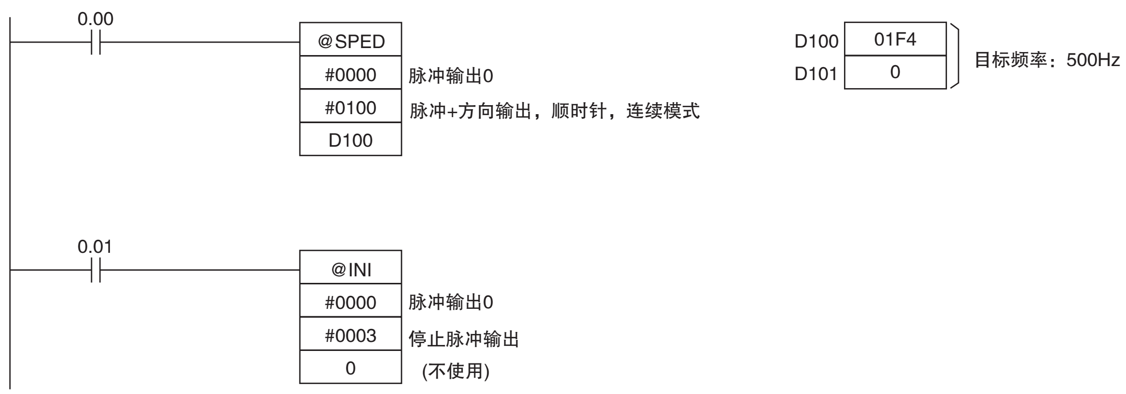
如果 C 等于 0003 Hex，则 INI(880) 指令将立即停止指定端口的脉冲输出或直线插补。如果在脉冲输入已停止时执行该指令，则脉冲量设定将被清除。

- **减速停止 (P=0030 或 0031 Hex 且 C=0004 Hex)**

如果 C 等于 0004 Hex，则 INI(880) 指令将减速停止指定端口的直线插补。减速比率取决于操作中的直线插补。

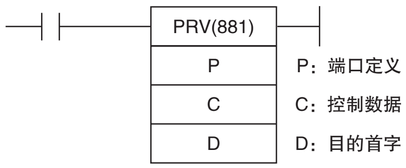
程序举例

下例中，当 CIO 0.00 变 ON 时，SPED(885) 指令在连续模式下从脉冲输出 0 启动 500Hz 的脉冲输出。当 CIO 0.01 变 ON 时，脉冲输出被 INI(880) 指令停止。



PRV

指令	助记符	变化	功能代码	功能
读高速 计数器的 PV 值	PRV	@PRV	881	PRV(881) 指令读入计数器模式下的高速计数器的 PV 值、脉冲输出的 PV 值以及中断输入的 PV 值。

符号	PRV
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
C	控制数据	---	1
D	目的首字	WORD	可变

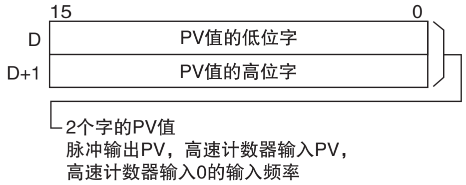
P：端口定义

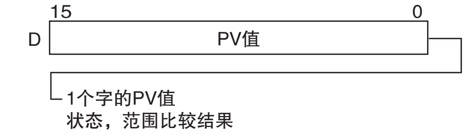
P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ - □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ - □)
0010 Hex	高速计数器 0
0011 Hex	高速计数器 1
0012 Hex	高速计数器 2
0013 Hex	高速计数器 3
0014 Hex	高速计数器 4
0015 Hex	高速计数器 5(无法用于 CP1E-E10D □ - □)
1000 Hex	PWM(891) 输出 0

C：控制数据

C	PRV(881) 的功能
0000 Hex	读入 PV
0001 Hex	读入状态
0002 Hex	读入范围比较结果
00 □ 3 Hex	P=0000 ～ 0003：读入脉冲输出 0 ～ 3 的输出频率 C=0003 Hex P=0010：读入高速计数器输入 0 的频率 C=0013 Hex：10ms 采样方法 C=0023 Hex：100ms 采样方法 C=0033 Hex：1s 采样方法

D：目的首字





● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，C	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---			OK			

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P 或 C 的指定范围时 ON。 · 当 P 和 C 的组合不允许时 ON。 · 当在未执行范围比较的情况下指定了读入范围比较结果时 ON。 · 当为除高速计数器 0 以外的其它参数指定了输出频率时 ON。 · 当指定了未被设定为高速计数器的端口时 ON。 · 当对直线插补执行中的脉冲输出执行 PRV(881) 指令时 ON。 · 其它情况下 OFF。

功能

PRV(881)为在P中指定的端口读入在C中指定的数据。各种数据和各个端口可能的组合如下表所示。

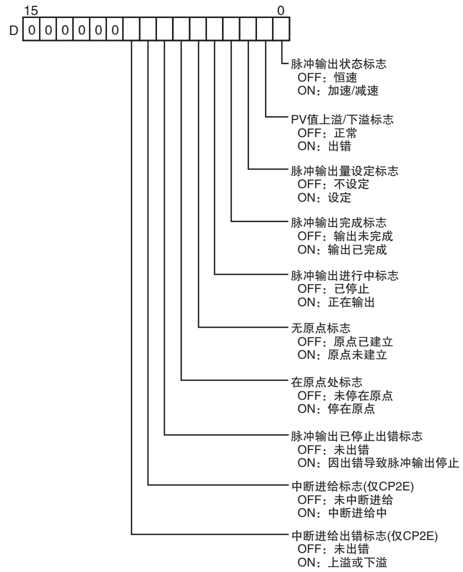
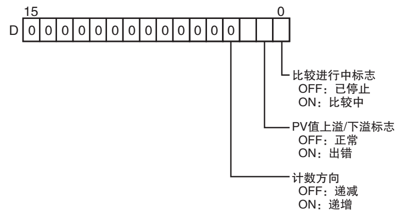
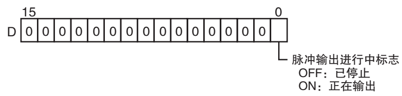
P：端口定义	C：控制数据		
	0000 Hex：读入 PV 值	0001 Hex：读入状态	0002 Hex：读入范围比较结果
0000 ～ 0003 Hex：脉冲输出	OK	OK	不允许
0010 ～ 0015 Hex：高速计数器输入	OK	OK	OK
1000 Hex：PWM(891) 输出	不允许	OK	不允许

P：端口定义	00□3 Hex：读入频率			
	0003 Hex：脉冲输出读入高速计数器频率	0013 Hex：10ms 采样方法	0023 Hex：100ms 采样方法	0033 Hex：1s 采样方法
0000 ～ 0003 Hex：脉冲输出	OK	不允许	不允许	不允许
0010 Hex：高速计数器输入	不允许	OK（仅限高速计数器 0)	OK（仅限高速计数器 0)	OK（仅限高速计数器 0)
1000 Hex：PWM(891) 输出	不允许	不允许	不允许	不允许

● 读入 PV 值 (C=0000 Hex)

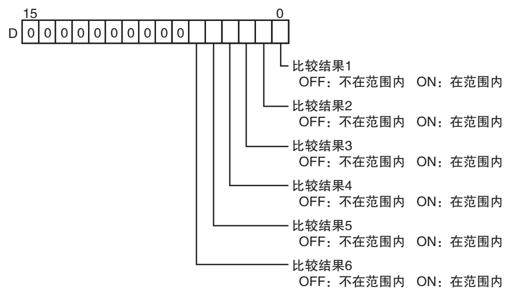
端口与模式		操作	设定范围
脉冲输出 (P=0000 ～ 0003 Hex)		将脉冲输出的当前值存储到 D 和 D+1 中。	8000 0000 ～ 7FFF FFFF Hex (–2,147,483,648 ～ 2,147,483,647)
高速计数器输入 (P=0010 ～ 0015 Hex)	线性模式	将高速计数器的当前值存储到 D 和 D+1 中。	8000 0000 ～ 7FFF FFFF Hex (–2,147,483,648 ～ 2,147,483,647)
	环形模式		0000 0000 ～ FFFF FFFF Hex (0 ～ 4,294,967,295)

● 读入状态 (C=0001 Hex)

端口与模式	操作	读入的结果
脉冲输出	将脉冲输出状态存储到 D 中。	
高速计数器输入	将高速计数器的状态存储到 D 中。	
PWM(891) 输出	将 PWM(891) 的输出存储到 D 中。	

● 读入范围比较的结果 (C=0002 Hex)

如果 C 等于 0002 Hex，则 PRV(881) 指令将读入范围比较的结果，并将该结果存储到 D 中，如下图所示。



- **读入脉冲输出或高速计数器频率 (C=00 □ 3 Hex)**

如果 C 等于 00 □ 3 Hex，则 PRV(881) 指令将读入正在从脉冲输出 0 ~ 3 输出的频率，或者正在输入到高速计数器 0 的频率，并将该频率存储到 D 和 D+1 中。

0000 ~ 0003 Hex(读入脉冲输出 0 ~ 3 的频率)

0000 0000 ~ 0001 86A0 Hex(0 ~ 100,000)

0010 Hex(读入高速计数器 0 的频率)

计数器输入方法：除 4× 微分相位模式以外的任何其它输入方法：

结果 =00000000 ~ 000186A0 Hex(0 ~ 100,000)

注 如果输入了高于 100kHz 的频率，则输出将保持最大值 000186A0 Hex。

计数器输入方法：4× 微分相位模式：

结果 =00000000 ~ 00030D40 Hex(0 ~ 200,000)

注 如果输入了高于 200kHz 的频率，则输出将保持最大值 00030D40 Hex。

- **脉冲频率计算方法**

该功能统计一个固定间隔 (采样时间) 内的脉冲数 , 并根据统计的数量来计算频率。通过设定 C 的最右边两个数位，可选择下述三个采样时间之一。

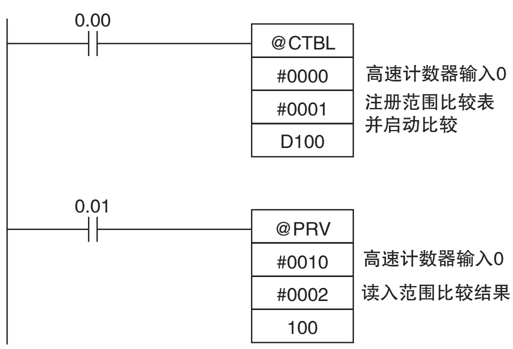
采样时间	C 的值	描述
10ms	0013 Hex	统计每 10ms 的脉冲数。1kHz 时的最大误差为 10%。
100ms	0023 Hex	统计每 100ms 的脉冲数。1kHz 时的最大误差为 1%。
1s	0033 Hex	统计每 1s 的脉冲数。1kHz 时的最大误差为 0.1%。

注意事项

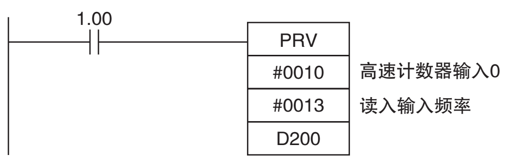
- 如果在 P 等于 0010 Hex(高速计数器 0) 且 C 等于 0013、0023 或 0033 Hex(采样方法) 时对计数器进行复位，则在计数器被复位后的采样时间期间读入的数据不可靠。
- 对直线插补执行中的脉冲输出，无法通过 PRV(881) 指令读入 PV 值和状态。执行 PRV(881) 指令时，P_ER 将变 ON。

程序举例

右例中，当 CIO 0.00 变 ON 时，CTBL(882) 指令注册一个范围比较表并对高速计数器 0 启动比较操作。当 CIO 0.01 变 ON 时，PRV(881) 指令读入当时的范围比较结果，并将结果存储到 CIO 0100 中。

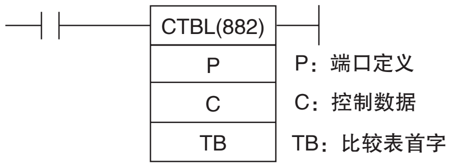


右例中，当 CIO 1.00 变 ON 时，PRV(881) 指令读入当时正在输入到高速计数器 0 的脉冲的频率，并将该频率作为十六进制值存储到 D200 和 D201 中。



CTBL

指令	助记符	变化	功能代码	功能
注册比较表	CTBL	@CTBL	882	CTBL(882)指令用于注册比较表并对高速计数器的PV值执行比较。

符号	CTBL
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
C	控制数据	---	1
TB	比较表首字	LWORD	可变

P：端口定义

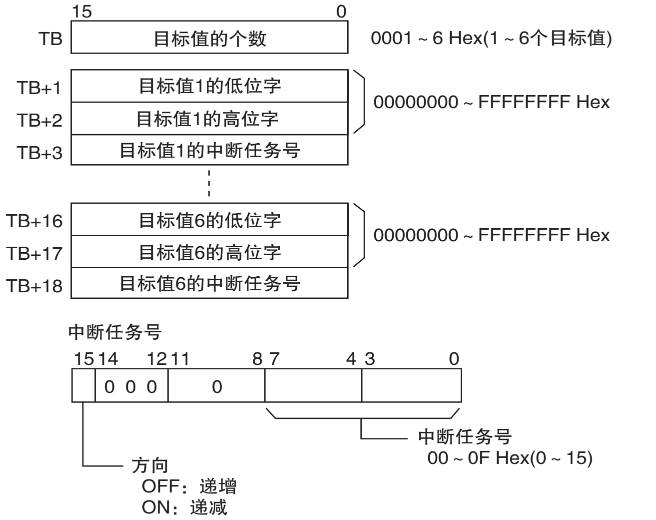
P	端口
0000 Hex	高速计数器 0
0001 Hex	高速计数器 1
0002 Hex	高速计数器 2
0003 Hex	高速计数器 3
0004 Hex	高速计数器 4
0005 Hex	高速计数器 5(无法用于 CP1E-E10D □ - □)

C：控制数据

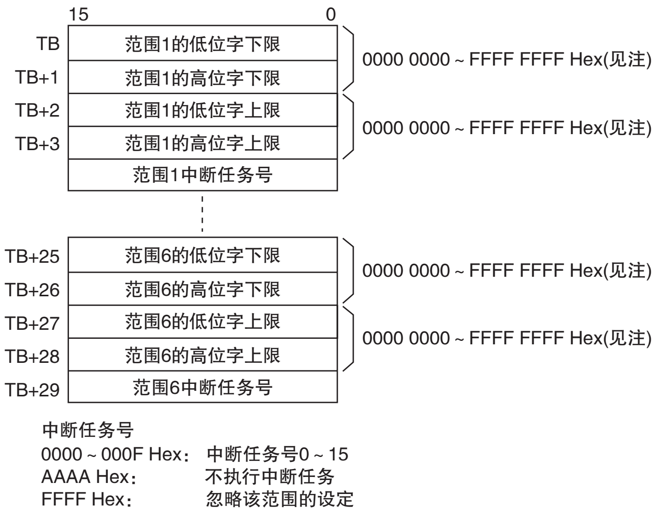
C	CTBL(882) 的功能
0000 Hex	注册一个目标值比较表并启动比较操作。
0001 Hex	注册一个范围比较表并执行一个比较操作。
0002 Hex	注册一个目标值比较表。比较操作通过 INI(880) 指令启动。
0003 Hex	注册一个范围比较表。比较操作通过 INI(880) 指令启动。

TB：比较表首字

- TB 是比较表的第一个字。比较表的结构取决于正在执行的比较的类型。
  对于目标值比较，比较表的长度由在 TB 中指定的目标值的个数来决定。表的长度可以介于 4 ~ 19 个字之间，如下所示。



- 对于范围比较，比较表始终包含 6 个范围。表的长度为 30 个字，如下所示。无需设定 6 个范围，只需对所有不使用的范围设定 FFFF Hex 的中断任务号。



注 对任一范围进行设定时，请务必使其上限大于等于其下限。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，C	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
TB	OK	OK	OK	OK	OK	OK	OK	OK	OK	---			OK			

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P 或 C 的指定范围时 ON。 · 当指定用于目标值比较的目标值的个数被设定为 0 时 ON。 · 当指定用于目标值比较的目标值的个数超过 6 时 ON。 · 当有任一范围的上限值小于下限值时 ON。 · 当在范围比较期间禁止了所有范围的设定值时 ON。 · 如果为增量脉冲模式设定了高速计数器，但在表中将比较方向设定为减量时 ON。 · 在将高速计数器设定为增量脉冲模式和线性模式的情况下，若同一个目标值在同一个方向的目标比较中的指定次数超过一次以上则为 ON。 · 当在将高速计数器设定为环形模式的情况下执行了指令且指定的值超过最大环形值时 ON。 · 当指定了未被设定为高速计数器的端口时 ON。 · 当比较操作正在执行时，采用另一种比较方法执行该指令时 ON。 · 其它情况下 OFF。

功能

CTBL(882) 指令注册一个比较表并对在 P 中指定的端口、以在 C 中指定的方法启动比较操作。比较表一经注册将一直有效，直到注册了另一个表或者 CPU 单元切换到 PROGRAM 模式为止。

每次执行 CTBL(882) 指令时，比较操作均在指定条件下启动。使用 CTBL(882) 指令来启动比较时，通常只需使用该指令的微分变化 (@CTBL(882)) 或只对一次扫描置 ON 的执行条件即可。

注 如果指定了尚未注册的中断任务，则在第一次生成该中断时将发生致命的程序错误。

● 注册比较表 (C=0002 或 0003 Hex)

如果将 C 设定为 0002 或 0003 Hex，则将注册一个比较表，但不会启动比较。比较操作通过 INI(880) 指令启动。

● 注册比较表并启动比较 (C=0000 或 0001 Hex)

如果将 C 设定为 0000 或 0001 Hex，则将注册一个比较表并启动比较。

● 停止比较

比较操作通过 INI(880) 指令停止。无论采用何种指令启动比较，均无区别。

● 目标值比较

当 PV 值与目标值匹配时，将调用并执行相应的中断任务。

· 可为一个以上的目标值指定同一个中断任务号。

· 还可设定方向，用于指定目标值是在 PV 递增时还是递减时有效。如果字中用于为范围指定中断任务号的位 15 为 OFF，则仅当 PV 值递增时将 PV 值与目标值进行比较；而如果位 00 为 ON，则仅当 PV 值递减时进行比较。

· 比较表最多可包含 6 个目标值，目标值的个数在 TB 中指定 (即比较表的长度取决于所指定的目标值的个数)。

· 比较操作将对在表中注册的所有目标值执行。

注 1 如果在同一个表中对同一个比较方向的同一个目标值注册了一次以上，将产生错误。
2 在为增量脉冲模式设定了高速计数器时，如果在表中将比较方向设定为减量，将产生错误。
3 如果在 PV 值等于必须朝与设定的比较方向相反的方向才能达到的目标值时计数方向发生改变，则将无法满足该目标值的比较条件。请勿将目标值设定为计数值的峰值和谷值。

● 范围比较

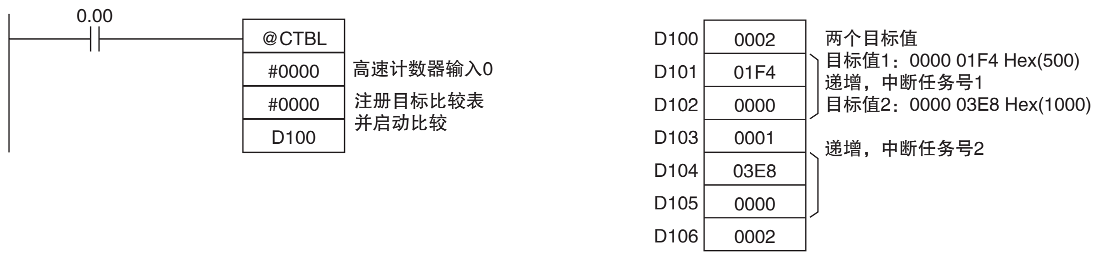
当 PV 值进入设定的范围时，将调用并执行相应的中断任务。

- 可为一个以上的目标值指定同一个中断任务号。
- 范围比较表包含 6 个范围，每个范围通过下限和上限来定义。如果不使用某个范围，可将中断任务号设定为 FFFF Hex 以禁止该范围。
- 当 PV 值进入范围时，仅调用并执行一次中断任务。
  如果在进行比较时 PV 值位于一个以上的范围内，则距离比较表的开头最近的范围的中断任务将取得优先权，而其它任务则将在后续循环中执行。
- 如果不需要执行某个中断任务，可将中断任务号指定为 AAAA Hex。可通过 PRV(881) 指令或范围比较进行中标志来读取范围比较的结果。

注 当有任一范围的上限值小于下限值时，将产生错误。

程序举例

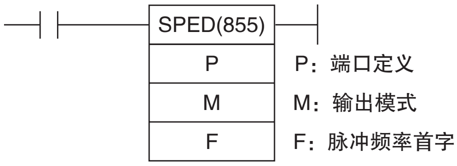
下例中，当 CIO 0.00 变 ON 时，CTBL(882) 指令注册一个目标值比较表，并对高速计数器 0 启动比较操作。高速计数器的 PV 值作增量计数，当 PV 值到达 500 时，即等于目标值 1，于是执行中断任务 1。当 PV 值增至 1000 时，即等于目标值 2，于是执行中断任务 2。



SPED

指令	助记符	变化	功能代码	功能
速度输出	SPED	@SPED	885	SPED(885) 指令用于为特定端口设定输出脉冲频率，并启动无加速或减速的脉冲输出。

	SPED
符号	



适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

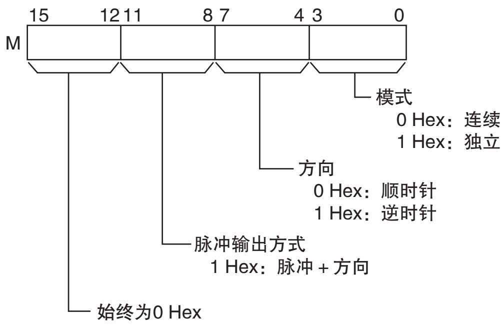
操作数

操作数	描述	数据类型	大小
P	端口定义	UINT	1
M	输出模式	WORD	1
F	脉冲频率首字	UDINT	2

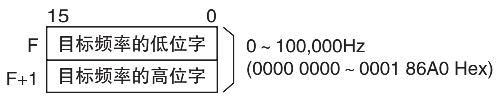
P：端口定义

P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ – □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ – □)

M：输出模式



F：脉冲频率首字



以 Hz 为单位指定脉冲频率

● 区域类型

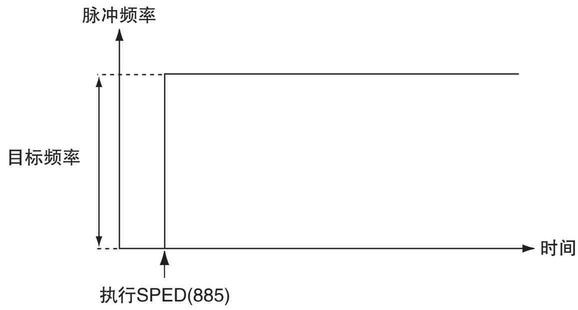
<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>P，M</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>F</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P、M 或 F 的指定范围时 ON。 · 当已经在执行 PLS2(887)、ORG(889)、IFEED(892) 或 ITPL(893) 指令来控制指定端口的脉冲输出时 ON。 · 当使用 SPED(885) 或 INI(880) 指令在脉冲输出期间使模式在连续和独立输出模式之间切换时 ON。 · 当在某个循环任务中正在执行一条控制脉冲输出的指令时，若在某个中断任务中执行 SPED(885) 指令则 ON。 · 当在独立模式下以脉冲数的绝对值执行 SPEC(885) 指令但原点尚未建立时 ON。 · 其它情况下 OFF。

功能

SPED(885) 指令采用在 M 中指定的方法，以在 F 中指定的频率对在 P 中指定的端口启动脉冲输出。每次执行 SPED(885) 指令时均会启动脉冲输出。因此，通常只需使用该指令的微分变化 (@SPED(885)) 或只对一次扫描置 ON 的执行条件即可。



在独立模式下，当已输出了事先通过 PULS(886) 指令所设定的数量的脉冲之后，脉冲输入将自动停止。在连续模式下，脉冲输出将一直持续到从程序使其停止为止。

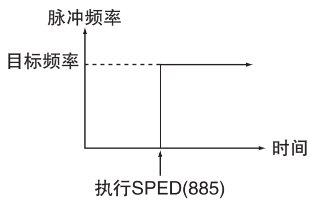
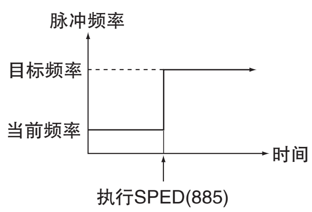
如果正在输出脉冲时，在独立模式和连续模式之间进行切换，则将产生错误。

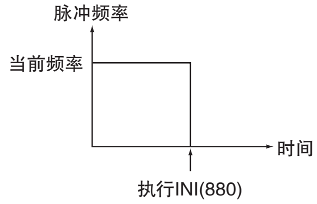
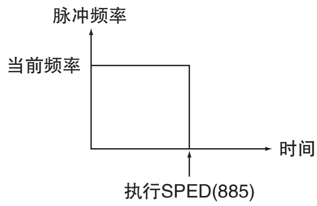
注 SPED 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

● 连续模式速度控制

当连续模式操作启动时，脉冲输出将一直持续到从程序使其停止为止。

注 如果将 CPU 单元改为 PROGRAM 模式，则脉冲输出将立即停止。

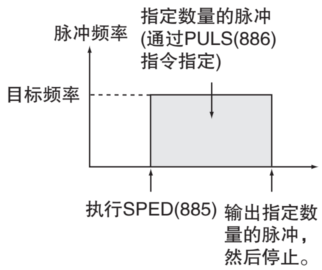
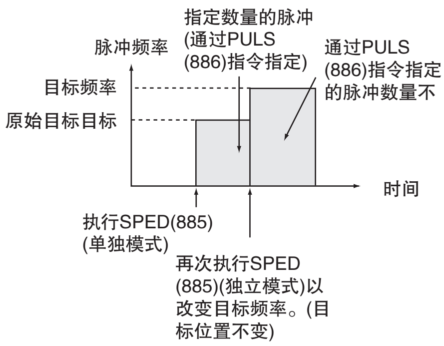
操作	目的	用途	频率改变	描述	步骤 / 指令
启动脉冲输出	以指定的速度进行输出	一步改变速度 (频率)		以指定的频率输出脉冲。	SPED(885)(连续)
改变设定	一步改变速度	操起期间改变速度		一步改变脉冲输出的频率 (升高或降低)。	SPED(885)(连续) ↓ SPED(885)(连续)

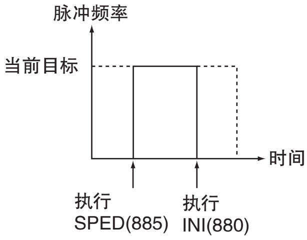
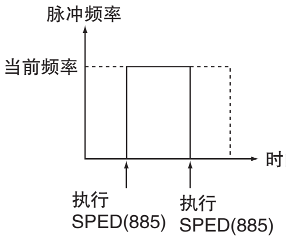
操作	目的	用途	频率改变	描述	步骤 / 指令
停止脉冲输出	停止脉冲输出	立即停止		立即停止脉冲输出	SPED(885)(连续) ↓ INI(880)
	停止脉冲输出	立即停止		立即停止脉冲输出	SPED(885)(连续) ↓ SPED(885)(连续，目标频率为 0Hz)

● 独立模式位置控制

当独立模式操作启动时，脉冲输出将一直持续到已输出指定数量的脉冲为止。

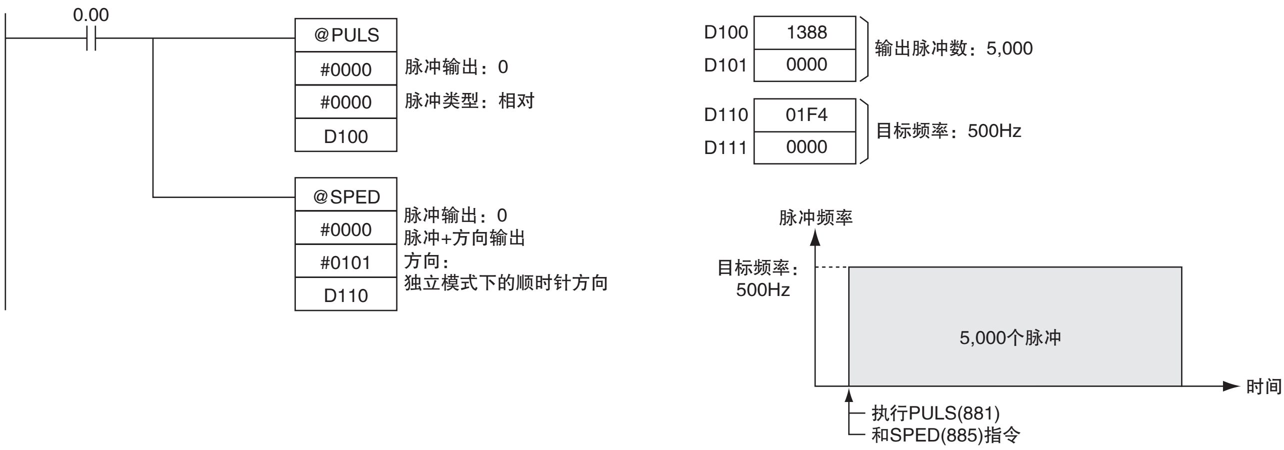
注 · 如果将 CPU 单元改为 PROGRAM 模式，则脉冲输出将立即停止。
- 每次重启输出时均必须设定输出脉冲数。
- 输出脉冲数必须事先用PULS(881)指令设定。若不首先执行PULS(881)指令，则SPED(885)指令将无法输出脉冲。
- 如果通过 PULS(881) 指令设定的脉冲数是一个绝对值，则在 SPED(885) 指令的操作数中设定的方向将被忽略。

操作	目的	用途	频率改变	描述	步骤 / 指令
启动脉冲输出	以指定的速度输出	无加减速的位置控制		以指定的频率启动脉冲输出，并在已输出指定数量的脉冲之后立即停止。 **注** 位置控制 (脉冲输出) 期间不能改变目标位置 (脉冲数)。	PULS(886) ↓ SPED(885) (独立)
改变设定	一步改变速度	操作期间一步改变速度		可在位置控制期间执行 SPED(885) 指令，从而一步改变 (升高或降低) 脉冲输出频率。 目标位置 (指定的脉冲数) 不变。	PULS(886) ↓ SPED(885) (独立) ↓ SPED(885) (独立)

操作	目的	用途	频率改变	描述	步骤 / 指令
停止脉冲输出	停止脉冲输出 (不保留脉冲数设定。)	立即停止		立即停止脉冲输出并清除输出脉冲数设定。	PULS(886) ↓ SPED(885) (独立) ↓ INI(880)
	停止脉冲输出 (不保留脉冲数设定。)	立即停止		立即停止脉冲输出并清除输出脉冲数设定。	PULS(886) ↓ SPED(885) (独立) ↓ SPED(885) (独立，目标频率为 0Hz)

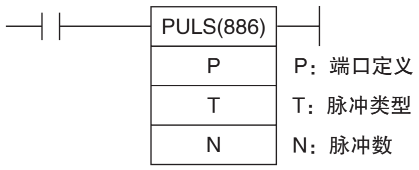
程序举例

下例中，当 CIO 0.00 变 ON 时，PULS(886) 指令设定脉冲输出 0 的输出脉冲数。该例中设定为 5,000 个脉冲的绝对值。随后执行 SPED(885) 指令，使用脉冲 + 方向方法以顺时针方向在独立模式下启动目标频率为 500Hz 的脉冲输出。



PULS

指令	助记符	变化	功能代码	功能
设置脉冲	PULS	@PULS	886	PULS(886) 用于设置脉冲输出量 (输出脉冲的数量)。

	PULS
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
T	脉冲类型	---	1
N	脉冲数	DINT	2

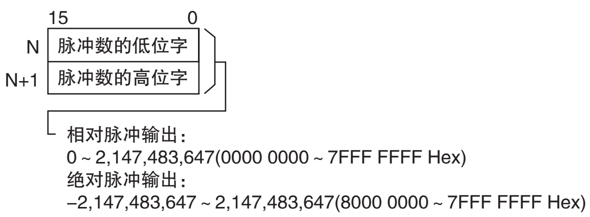
P：端口定义

P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ - □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ - □)

T：脉冲类型

T	脉冲类型
0000 Hex	相对
0001 Hex	绝对

N 和 N+1：脉冲数



- 将输出的移动脉冲的实际数量如下：
  对于相对值脉冲输出，移动脉冲的个数 = 脉冲的设定数量。
  对于绝对脉冲输出，移动脉冲的数量 = 脉冲的设定数量 –PV 值。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，T	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P、T 或 N 的指定范围时 ON。 · 当对某个正在输出脉冲的端口执行 PULS(886) 指令时 ON。 · 当在某个循环任务中正在执行一条控制脉冲输出的指令时，若在某个中断任务中执行 PULS(886) 指令则 ON。 · 其它情况下 OFF。

功能

PULS(886) 指令为在 P 中指定的端口设定脉冲类型和脉冲数 (在 T 和 N 中指定)。实际脉冲输出将在稍后通过 SPED(885) 或 ACC(888) 指令以独立模式启动。

注 · 正在输出脉冲时如果执行 PULS(886) 指令，则将发生错误。请使用该指令的微分变化 (@PULS(886)) 或只对一次扫描置 ON 的执行条件，以防止该错误发生。

· 即使使用了 INI(880) 指令来改变脉冲输出的 PV 值，PULS(886) 指令已计算出的脉冲输出数量也不会改变。

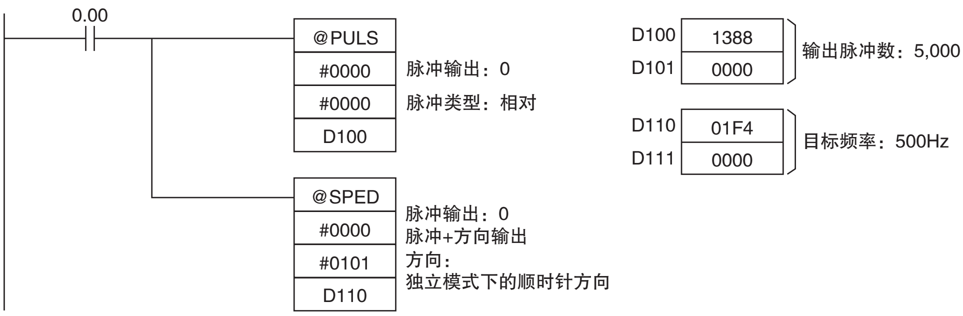
· 如果通过 PULS(881) 指令设定的脉冲数是一个绝对值，则为 SPED(885) 或 ACC(888) 指令所设定的方向将被忽略。

· 可以移出脉冲输出量 PV 值的范围之外 (–2,147,483,648 ～ 2,147,483,647)。

· PULS 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

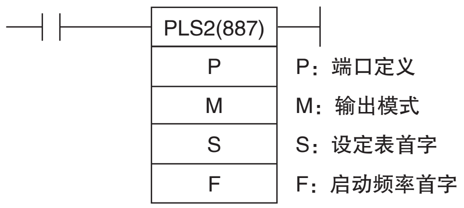
程序举例

下例中，当 CIO 0.00 变 ON 时，PULS(886) 指令设定脉冲输出 0 的输出脉冲数。该例中设定为 5,000 个脉冲的绝对值。随后执行 SPED(885) 指令，使用脉冲 + 方向方法以顺时针方向在独立模式下启动目标频率为 500Hz 的脉冲输出。



PLS2

指令	助记符	变化	功能代码	功能
脉冲输出	PLS2	@PLS2	887	PLS2(887) 指令将指定数量的脉冲输出到指定端口。脉冲输出以指定的启动频率启动、以指定的加速率加速至目标频率、以指定的减速率减速，然后在与启动频率大致相同的频率处停止。

	PLS2
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

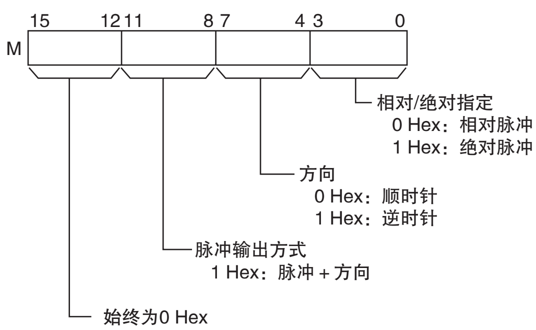
操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
M	输出模式	---	1
S	设定表首字	WORD	6
F	启动频率首字	UDINT	2

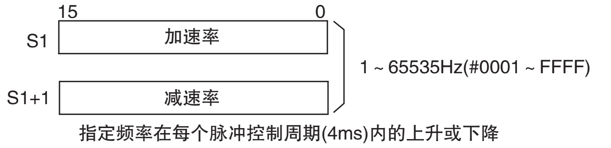
P：端口定义

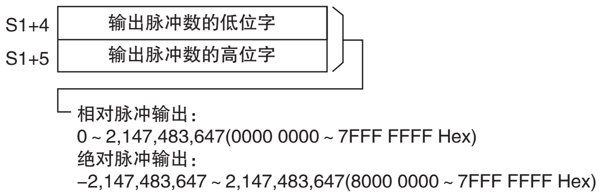
P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ - □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ - □)

M：输出模式



S：设定表首字



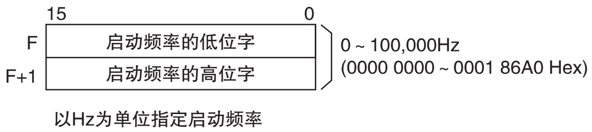


将输出的移动脉冲的实际数量如下：

- 对于相对脉冲输出，移动脉冲的数量 = 脉冲的设定数量。
- 对于绝对脉冲输出，移动脉冲的数量 = 脉冲的设定数量 –PV 值。

F：启动频率首字

启动频率在 F 和 F+1 中设定。



● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，M	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
F	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P、M、S 或 F 的指定范围时 ON。 · 当对某个正在通过 SPED(885)、ORG(889)、IFEED(892) 或 ITPL(893) 指令输出脉冲的端口执行 PLS2(887) 指令时 ON。 · 当在某个循环任务中正在执行一条控制脉冲输出的指令时，若在某个中断任务中执行 PLS2(887) 指令则 ON。 · 当执行 PLS2(887) 指令以获得绝对脉冲输出但原点尚未建立时 ON。 · 其它情况下 OFF。

功能

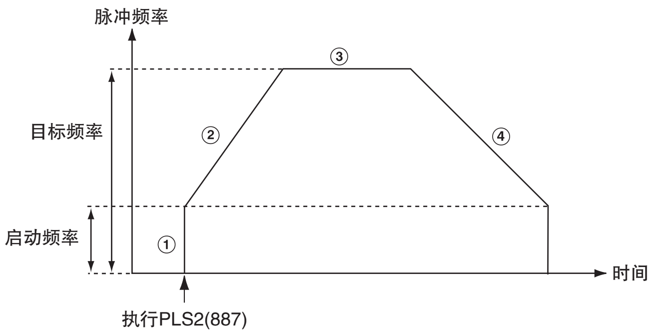
PLS2(887) 指令采用在 M 中指定的模式，以在 F 中指定的频率 (图中的 1) 对在 P 中指定的端口启动脉冲输出。

在每个脉冲控制周期 (4ms)，频率均以在 S 中指定的加速率 (图中的 2) 升高，直到达到在 S 中指定的目标频率为止。

当已达到目标频率时，将停止加速并以恒速 (图中的 3) 输出脉冲。

减速点通过输出脉冲数和在S中设定的减速率进行计算，当到达该点时，在每个脉冲控制周期(4ms)，频率均以在 S 中指定的减速率 (图中的 4) 减速，直到达到在 S 中指定的启动频率为止，并在该点停止脉冲输出。

每次执行 PLS2(887) 指令时均会启动脉冲输出。因此，通常只需使用该指令的微分变化 (@PLS2(887)) 或只对一次扫描置 ON 的执行条件即可。



PLS2(887) 指令只能用于位置控制。

对于 CJ1M CPU 单元，PLS2(887) 指令可在 ACC(888) 指令的脉冲输出期间以独立或连续模式执行，或者在加速、恒速或减速期间执行。(见注 1 和 2) ACC(888) 指令也可在 PLS2(887) 指令的脉冲输出期间、加速、恒速或减速期间执行。

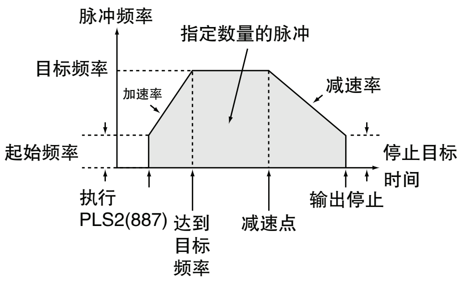
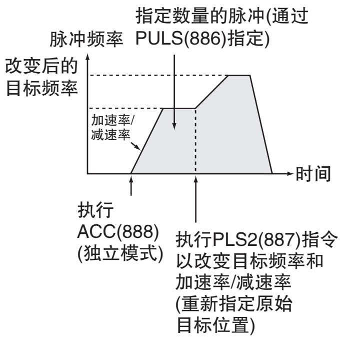
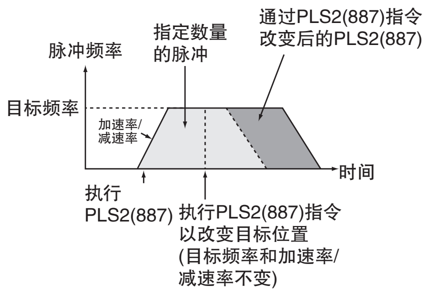
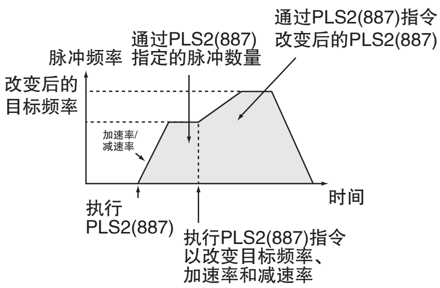
注 1 在通过 ACC(888) 指令 (连续模式) 控制速度期间以与 ACC(888) 指令相同的目标频率来执行 PLS2(887) 指令，即可获得定距离的中断馈给。在该应用中，将不会通过 PLS2(887) 指令来执行加速，但若将加速率设定为 0，则出错标志将置 ON，且不会执行 PLS2(887) 指令。因此请务必将加速率设定为除 0 以外的其它值。

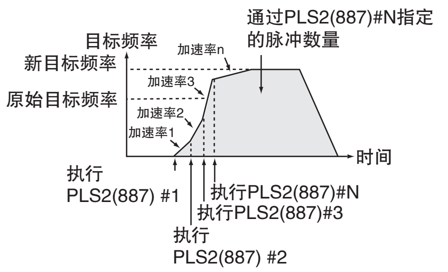
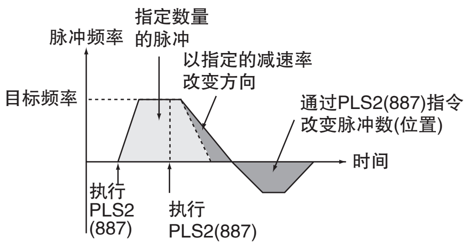
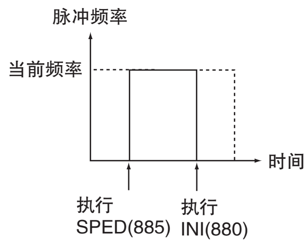
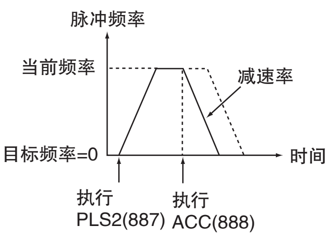
2 如果在停止脉冲输出到停止之后的一个循环时段内 (即当脉冲输出进行中标志为 ON 时) 执行 PLS2(887) 指令，则脉冲输出将在停止之后的下一个循环中重新启动。
但是，如果脉冲输出是因 INI(880) 而停止的，则脉冲输出指令将在停止输出的一个循环之内变为无效。此时，将一直执行指令直到脉冲输出进行中标志置 OFF 为止。

3 PLS2 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

● 独立模式位置控制

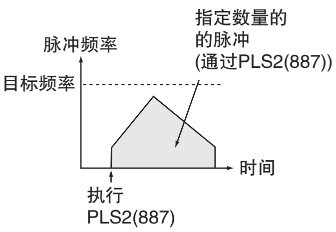
注 如果将 CPU 单元改为 PROGRAM 模式，则脉冲输出将立即停止。

操作	目的	用途	频率改变	描述	步骤 / 指令
启动脉冲输出	复杂梯形控制	通过梯形加速和减速进行位置控制(加速率和减速率采用不同的值，有启动速度) 位置控制期间可改变脉冲数。		以固定的加速率和减速率进行加速和减速。当已输出指定数量的脉冲时，脉冲输出将停止。(见注) **注** 位置控制期间可改变目标位置(指定的脉冲数)。	PLS2(887)
改变设定	平稳改变速度(采用不相等的加速率和减速率)	在位置控制期间改变目标速度(频率)(采用不相等的加速率和减速率)		可在位置控制过程中执行 PLS2(887) 以改变加速率、减速率和目标频率。 **注** 为防止故意改变目标位置，必须以绝对坐标来指定原始目标位置。	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (独立) ↓ PLS2(887)
	改变目标位置	位置控制期间改变目标位置(多路启动功能)		可在位置控制期间执行 PLS2(887) 指令，以改变目标位置(脉冲数)、加速率、减速率和目标频率。 **注** 如果在改变设定后无法保持恒速，则将发生错误且原始操作将继续，直到到达原始目标位置为止。	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (独立) ↓ PLS2(887)
	平稳改变目标位置和速度	在位置控制期间改变目标位置和目标速度(频率)(多路启动功能)		可在位置控制期间执行 PLS2(887) 指令，以改变目标位置(脉冲数)、加速率、减速率和目标频率。 **注** 如果在改变设定后无法保持恒速，则将发生错误且原始操作将继续，直到到达原始目标位置为止。	PULS(886) ↓ ACC(888) (独立) ↓ PLS2(887) PLS2(887) ↓ PLS2(887)

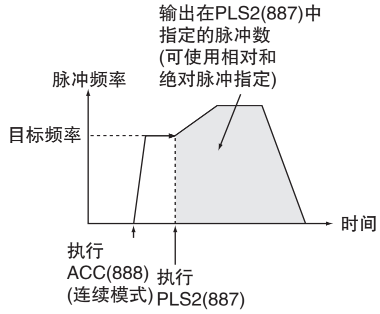
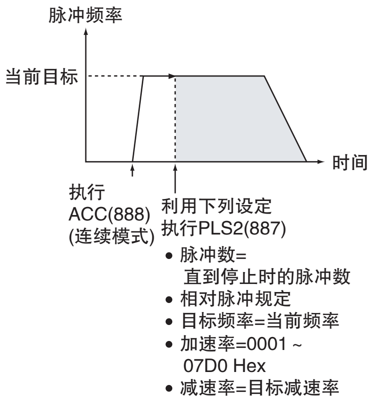
操作	目的	用途	频率改变	描述	步骤 / 指令
改变设定，继续	平稳改变目标位置和速度，继续	位置控制期间改变加速率和减速率(多路启动功能)		可在位置控制期间执行 PLS2(887) 以改变加速率或减速率。	PULS(886) ↓ ACC(888) (独立) ↓ PLS2(887) PLS2(887) ↓ PLS2(887)
	改变方向	位置控制期间改变方向		可在位置控制期间以绝对脉冲规定来执行 PLS2(887) 指令，以改变绝对脉冲和使方向反转。	PLS2(887) ↓ PLS2(887) PULS(886) ↓ ACC(888) (独立) ↓ PLS2(887)
停止脉冲输出	停止脉冲输出(不保留脉冲数设定)	立即停止		立即停止脉冲输出并清除输出脉冲数。	PLS2(887) ↓ INI(880)
	平稳停止脉冲输出(不保留脉冲数设定)	减速至停止		使脉冲输出减速直至停止。	PLS2(887) ↓ ACC(888) (独立，目标频率为 0Hz)

注 三角形控制

如果指定的脉冲数小于达到目标频率所需的脉冲数并返回 0，则该功能将自动缩短加速 / 减速时间并执行三角形控制 (仅限加速和减速)。将不会产生错误。

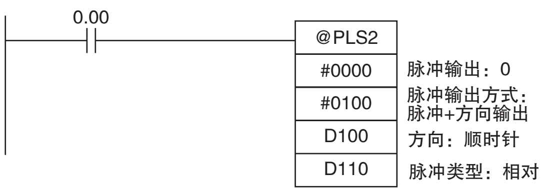


● 从连续模式速度控制切换到独立模式位置控制

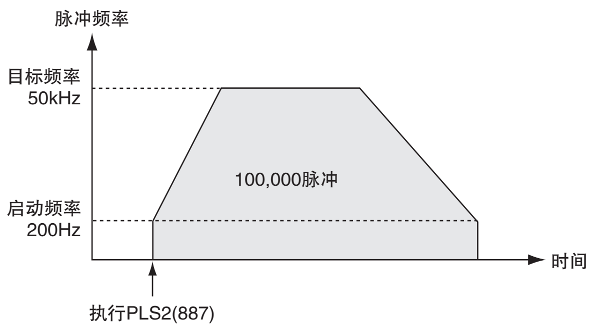
<table>
<tr><th>应用实例</th><th>频率改变</th><th>描述</th><th>步骤 / 指令</th></tr>
<tr><td>操作期间从速度控制改为定距离位置控制</td><td>
</td><td rowspan="2">在通过ACC(888)指令启动的速度控制操作期间可执行PLS2(887)指令，从而改为位置控制操作。</td><td rowspan="2">ACC(888)
(连续)
↓
PLS2(887)</td></tr>
<tr><td>定距离馈给中断</td><td>
</td></tr>
</table>

程序举例

下例中，当 CIO 0.00 变 ON 时，PLS2(887) 指令以 100,000 个脉冲的绝对脉冲规格启动从脉冲输出 0 的脉冲输出。脉冲输出从 200Hz 开始，以 500Hz/4ms 的加速率进行加速，直到达到 50kHz 的目标速度。从减速点开始，脉冲输出以 250Hz/4ms 的减速率进行减速，直到达到 200Hz 的启动速度为止，然后脉冲输出在该点停止。

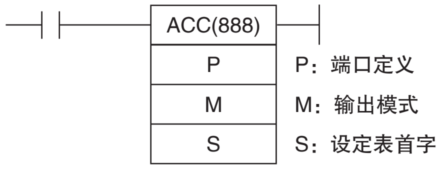


D100	01F4	加速率：500Hz/4ms
D101	00FA	减速率：250Hz/4ms
D102	C350	目标频率：50kHz
D103	0000	
D104	86A0	脉冲输出量：100,000个脉冲
D105	0001	
D110	00C8	启动频率：200Hz
D111	0000	



ACC

指令	助记符	变化	功能代码	功能
加速度控制	ACC	@ACC	888	ACC(888)指令将脉冲以指定的频率、指定的加速率和减速率输出到指定的输出端口。

	ACC
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

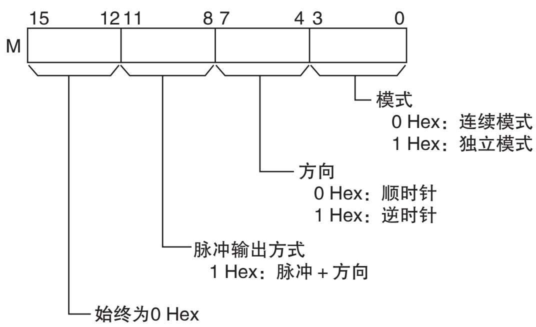
操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
M	输出模式	---	1
S	设定表首字	WORD	3

P：端口定义

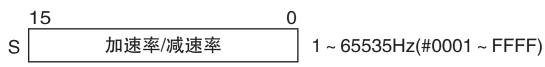
P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ - □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ - □)

M：输出模式

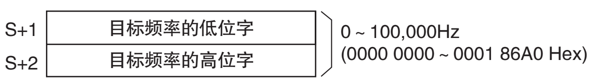


注 对脉冲输出 0 和 1 使用相同的脉冲输出方法。

S：设定表首字



指定频率在每个脉冲控制周期(4ms)内的上升或下降



以Hz为单位指定加速或减速之后的频率

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，M	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---			OK			

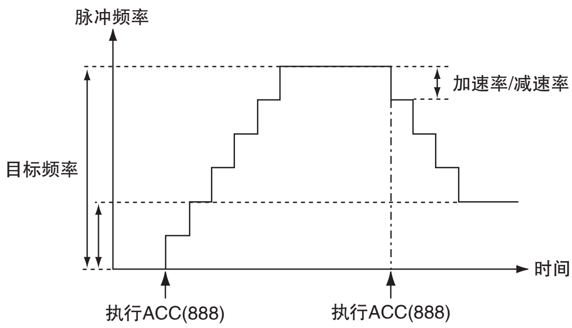
条件标志

名称	标记	操作
出错标志	P_ER	・当超出 P、M 或 S 的指定范围时 ON。 ・当正在使用 ORG(889)、IFEED(892) 或 ITPL(893) 指令为指定端口输出脉冲时 ON。 ・当对某个正在通过 SPED(885)、ACC(888) 或 PLS2(887) 指令输出脉冲的端口执行 ACC(888) 指令以在独立和连续模式之间进行切换时 ON。 ・当在某个循环任务中正在执行一条控制脉冲输出的指令时，若在某个中断任务中执行 ACC(888) 指令则 ON。 ・当执行 ACC(888) 指令以在独立模式下获得绝对脉冲输出但原点尚未建立时 ON。 ・其它情况下 OFF。

功能

ACC(888) 指令采用在 M 中指定的模式，以在 S 中指定的目标频率和加速率 / 减速率、对在 P 中指定的端口启动脉冲输出。在每个脉冲控制周期 (4ms)，频率均以在 S 中指定的加速率升高，直到达到在 S 中指定的目标频率为止。

每次执行 ACC(888) 指令时均会启动脉冲输出。因此通常只需使用该指令的微分变化 (@ACC(888)) 或只对一次扫描置 ON 的执行条件即可。



在独立模式下，当已输出指定数量的脉冲时，脉冲输出将自动停止。在连续模式下，脉冲输出将一直持续到从程序使其停止为止。

脉冲输出期间，如果试图在独立模式和连续模式之间进行切换，则将产生错误。

PLS2(887) 指令可在 ACC(888) 指令的脉冲输出期间以独立或连续模式执行，或者在加速、恒速或减速期间执行。(见注) ACC(888) 指令也可在 PLS2(887) 指令的脉冲输出期间、加速、恒速或减速期间执行。

如果在独立或连续模式下以 0Hz 的目标频率来执行 ACC(888) 指令，然后在脉冲输出停止之前执行 ACC(888) 或 PLS2(887) 指令，则目标频率不会改变且脉冲输出将停止。因此请在脉冲输出停止之后再执行 ACC(888) 或 PLS2(887) 指令。

注 1 在通过 ACC(888) 指令 (连续模式) 控制速度期间以与 ACC(888) 指令相同的目标频率来执行 PLS2(887) 指令，即可获得定距离的中断馈给。在该应用中，将不会通过 PLS2(887) 指令来执行加速，但若将加速率设定为 0，则出错标志将置 ON，且不会执行 PLS2(887) 指令。因此请务必将加速率设定为除 0 以外的其它值。

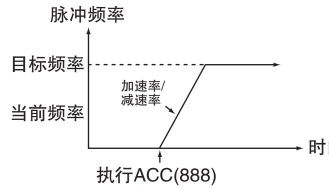
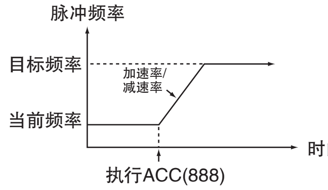
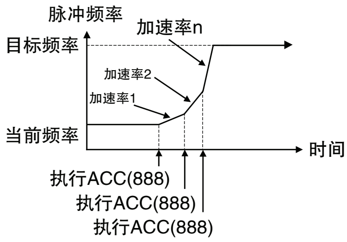
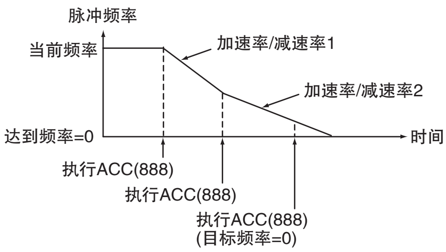
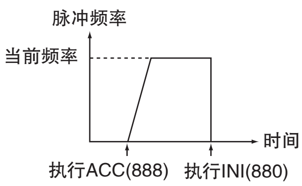
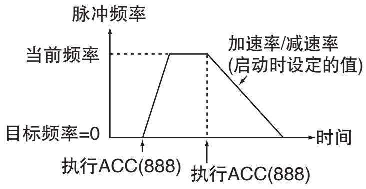
2 如果在停止脉冲输出到停止之后的一个循环时段内 (即当脉冲输出进行中标志为 ON 时) 执行 ACC(888) 或 PLS2(887)，则脉冲输出将在停止之后的下一个循环中重新启动。
但是，如果脉冲输出是因 INI(880) 而产生的，则脉冲输出指令在停止输出的一个循环之内将无效。此时，将一直执行指令直到脉冲输出进行中标志置 OFF 为止。

3 ACC 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

● 连续模式速度控制

脉冲输出将一直持续到从程序使其停止为止。

注 如果将 CPU 单元改为 PROGRAM 模式，则脉冲输出将立即停止。

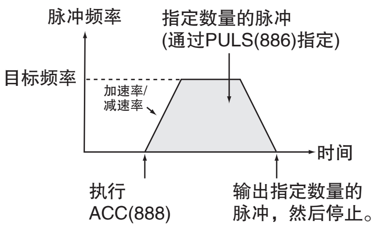
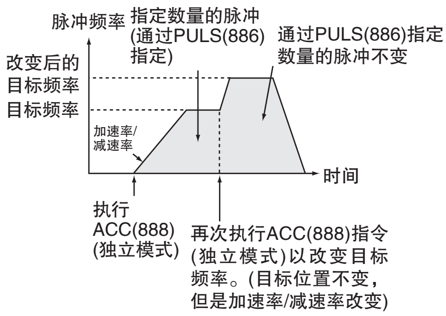
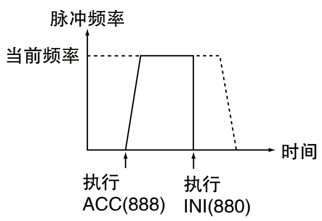
操作	目的	用途	频率改变	描述	步骤 / 指令
启动脉冲输出	以指定的加速率和速度进行输出	以固定的加速率进行加速 (升高频率)		输出脉冲并以固定的加速率改变频率。	ACC(888) (连续)
改变设定	平稳改变速度	操起期间平稳改变速度		从当前频率以固定的比率改变频率。可使频率加速或减速。	ACC(888) 或 SPED(885) (连续) ↓ ACC(888) (连续)
		操作期间以折线改变速度		在加速或减速期间改变加速率或减速率。	ACC(888) (连续) ↓ ACC(888) (连续)
		减速至停止		减速过程中减速率发生改变。 **注** 如果将目标频率设定为 0Hz，则将采用当前加速率。	ACC(888) (连续) ↓ ACC(888) (连续) ↓ ACC(888) (连续，目标频率为 0Hz)
停止脉冲输出	停止脉冲输出	立即停止		立即停止脉冲输出。	ACC(888) (连续) ↓ INI(880) (连续)
	平稳停止脉冲输出	减速至停止		使脉冲输出减速直至停止。 **注** 如果第 2 条 ACC (888)指令的目标频率为 0Hz，则将采用第 1 条 ACC(888) 指令处的减速率。	ACC(888) (连续) ↓ ACC(888) (连续，目标频率为 0Hz)

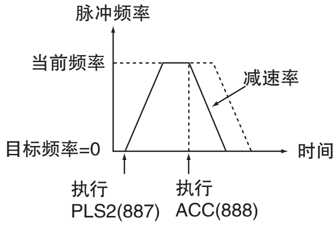
● 独立模式位置控制

当独立模式操作启动时，脉冲输出将一直持续到已输出指定数量的脉冲为止。

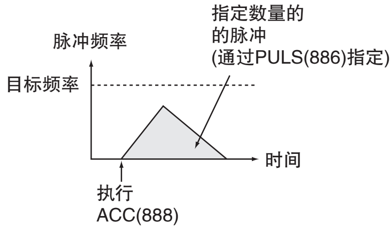
减速点通过输出脉冲数和在S中设定的减速率进行计算，当到达该点时，在每个脉冲控制周期(4ms)，频率均以在 S 中指定的减速率减速，直到已输出指定数量的脉冲为止，并在该点停止脉冲输出。

注 1 如果将 CPU 单元改为 PROGRAM 模式，则脉冲输出将立即停止。
2 每次重启输出时均必须设定输出脉冲数。
3 输出脉冲数必须事先用PULS(881)指令设定。若不首先执行PULS(881)指令，则ACC(888)指令将无法输出脉冲。
4 如果通过 PULS(881) 指令设定的脉冲数是一个绝对值，则在 ACC(888) 指令的操作数中设定的方向将被忽略。

操作	目的	用途	频率改变	描述	步骤 / 指令
启动脉冲输出	简单梯形控制	通过梯形加速和减速进行位置控制(加速率和减速率采用相同的值，无启动速度)位置控制期间不能改变脉冲数。		以相同的固定加速率和减速率进行加速和减速，并在已输出指定数量的脉冲之后立即停止。(见注) **注** 位置控制期间不能改变目标位置(指定的脉冲数)。	PULS(886) ↓ ACC(888) (独立)
改变设定	平稳改变速度(采用相同的加速率和减速率)	在位置控制期间改变目标速度(加速率＝减速率)		可在位置控制期间执行ACC(888)指令，以改变加速率 / 减速率和目标频率。 目标位置 (指定的脉冲数) 不变。	PULS(886) ↓ ACC(888) 或 SPED(885) (独立) ↓ ACC(888) (独立) PLS2(887) ↓ ACC(888) (独立)
停止脉冲输出	停止脉冲输出(不保留脉冲数设定)	立即停止		立即停止脉冲输出并清除剩余的输出脉冲数。	PULS(886) ↓ ACC(888) (独立) ↓ INI(880)

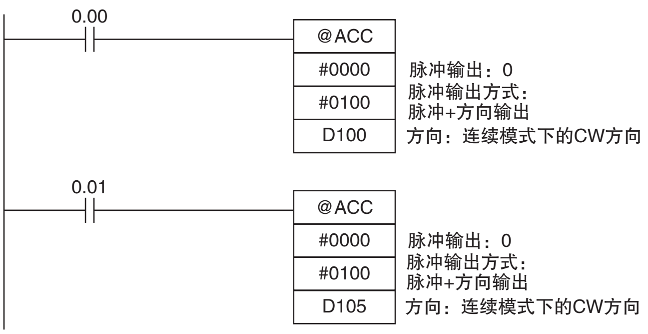
操作	目的	用途	频率改变	描述	步骤 / 指令
停止脉冲输出，继续	平稳停止脉冲输出。(不保留脉冲数设定。)	减速至停止		使脉冲输出减速直至停止。 **注** 如果通过ACC(888)指令启动操作，则原始的加速率 / 减速率将保持有效。 如果通过 SPED(885)指令启动操作，则加速率/减速率将无效且脉冲输出将立即停止。	PULS(886) ↓ ACC(888)或SPED(885) (独立) ↓ ACC(888) (独立，目标频率为 0Hz) ——— PLS2(887) ↓ ACC(888) (独立，目标频率为 0Hz)

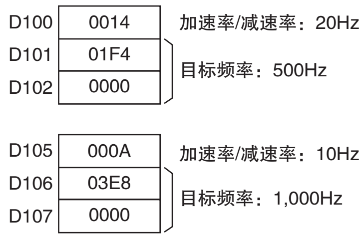
注 三角形控制
如果指定的脉冲数小于达到目标频率所需的脉冲数并返回 0，则该功能将自动缩短加速 / 减速时间并执行三角形控制 (仅限加速和减速)。将不会产生错误。

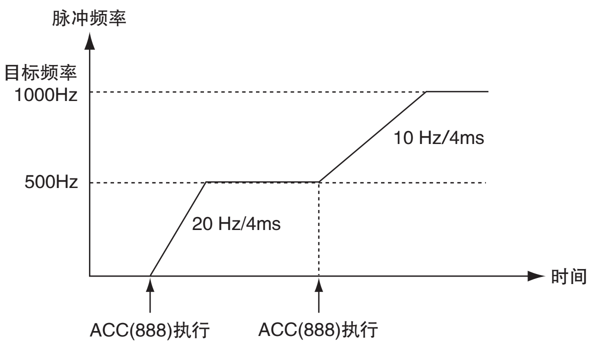


程序举例

下例中，当 CIO 0.00 变 ON 时，ACC(888) 指令在连续模式下采用脉冲 + 方向方法以顺时针方向启动从脉冲输出 0 的脉冲输出。脉冲输出以 20Hz/4ms 的加速率进行加速，直到达到 500Hz 的目标频率。当 CIO 0.01 变 ON 时，ACC(888) 指令使加速率变为 10Hz/4ms，直到达到 1,000Hz 的目标频率为止。

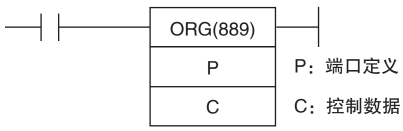






ORG

指令	助记符	变化	功能代码	功能
原点搜索	ORG	@ORG	889	ORG(889) 指令用于执行原点搜索和原点返回操作。

符号	ORG
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

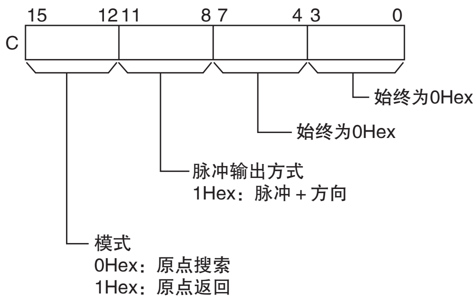
操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
C	控制数据	---	1

P：端口定义

P	端口
0000 Hex	脉冲输出 0
0001 Hex	脉冲输出 1
0002 Hex	脉冲输出 2(仅可用于 CP2E-N30/40/60D □ - □)
0003 Hex	脉冲输出 3(仅可用于 CP2E-N30/40/60D □ - □)

C：控制数据



● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P，C	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	· 当超出 P 或 C 的指定范围时 ON。 · 当在通过 SPED(885)、ACC(888)、PLS2(887)、IFEED(892) 或 ITPL(893) 指令输出脉冲期间指定了 ORG(889) 时 ON。 · 当在某个循环任务中正在执行一条控制脉冲输出的指令时，若在某个中断任务中执行 ORG(889) 指令则 ON。 · 当在 PLC 设置中所设定的原点搜索或原点返回参数不在范围内时 ON。 · 当原点搜索高速度小于等于原点搜索接近速度或者原点搜索接近速度小于等于原点搜索初始速度时 ON。 · 当在尚未建立原点的情况下试图进行原点返回操作时 ON。 · 其它情况下 OFF。

功能

ORG(889) 指令对在 P 中指定的端口、以在 C 中指定的方法执行原点搜索或原点返回操作。

执行 ORG(889) 指令之前，必须在 PLC 设置中设定下述参数。

原点搜索	原点返回
· 原点搜索功能允许 / 禁止 · 原点搜索操作模式 · 原点搜索操作设定 · 原点检测方法 · 原点搜索方向设定 · 原点搜索 / 返回初始速度 · 原点搜索高速度 · 原点搜索接近速度 · 原点补偿 · 原点搜索加速率 · 原点搜索减速率 · 限制输入信号类型 · 原点接近输入信号类型 · 原点输入信号类型 · 位置控制监控时间	· 原点搜索 / 返回初始速度 · 原点返回目标速度 · 原点返回加速率 · 原点返回减速率

每次执行 ORG(889) 指令时均会启动原点搜索或原点返回。因此通常只需使用该指令的微分变化 (@ORG(889)) 或只对一次扫描置 ON 的执行条件即可。

注 ORG 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

● 原点搜索 (C 的 12 ～ 15=0 Hex)

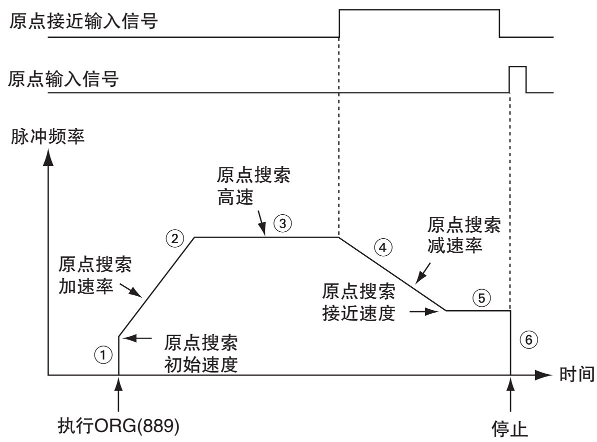
ORG(889) 指令采用指定的方法以原点搜索初始速度 (图中的 1) 来启动脉冲输出。

采用原点搜索加速率 (图中的 2) 将脉冲输出加速至原点搜索高速度。

然后脉冲输出以恒速 (图中的 3) 继续，直到原点接近输出信号变 ON，并从该点开始，脉冲输出采用原点搜索减速率 (图中的 4) 减速至原点搜索接近速度。

接着，脉冲再以恒速 (图中的 5) 输出，直到原点输入信号变 ON 为止。

当原点输入信号变 ON(图中的 6) 时，脉冲输出停止。



原点搜索操作完成时，出错计数器复位输出将变 ON。

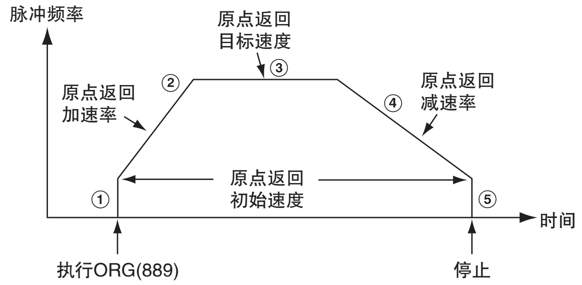
然而上述操作的具体情况取决于操作模式、原点检测方法和其它参数。

- **原点返回 (C 的 12 ～ 15=1 Hex)**

ORG(889) 指令采用指定的方法以原点返回初始速度 (图中的 1) 来启动脉冲输出。

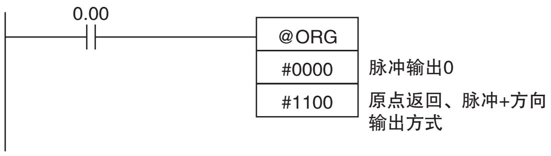
采用原点返回加速率 (图中的 2) 将脉冲输出加速至原点返回目标速度，然后脉冲输出以恒速 (图中的 3) 继续进行。

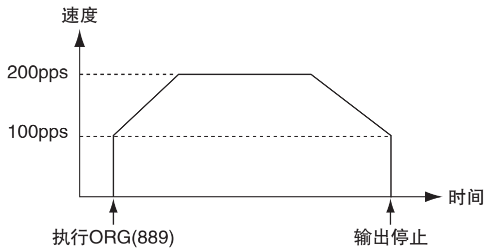
减速点通过距离原点剩余的脉冲数和减速率进行计算，当到达该点时，以原点返回减速率 (图中的 4) 使脉冲输出减速，直到达到原点返回启动速度为止，并在该点使脉冲输出停止于原点 (图中的 5)。



程序举例

下例中，当 CIO 0.00 变 ON 时，ORG(889) 指令采用脉冲 + 方向方法，通过输出脉冲来启动脉冲输出 0 的原点返回操作。根据 PLC 设置，初始速度为 100pps，目标速度为 200pps，加速率和减速率为 50Hz/4ms。



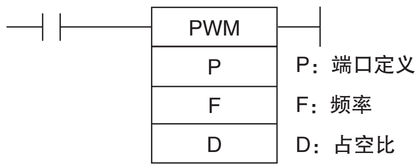


PLC 设置参数如下：

参数	设定
脉冲输出 0 原点搜索和原点返回的启动速度	0000 0064 Hex：100pps
脉冲输出 0 原点返回目标速度	0000 00C8 Hex：200pps
脉冲输出 0 原点返回加速率	0032 Hex：50Hex/4ms
脉冲输出 0 原点返回减速率	0032 Hex：50Hex/4ms

PWM

指令	助记符	变化	功能代码	功能
占空比可变脉冲	PWM	@PWM	891	PWM(891) 指令用于从指定端口输出指定占空比的脉冲。

	PWM
符号	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
P	端口定义	---	1
F	频率	---	1
D	占空比	---	1

P：端口定义

P	端口
1000 Hex	PWM 输出 0(占空比：以 1% 为增量；频率：以 0.1Hz 为增量)
1100 Hex	PWM 输出 0(占空比：以 1% 为增量；频率：以 1Hz 为增量)

F：频率

F 指定 PWM 的输出频率是介于 2.0 ～ 6,553.5Hz 之间 (0014 ～ FFFF Hex，以 0.1Hz 为增量单位) 还是 2 ～ 32,000Hz 之间 (0002 ～ 7D00 Hex，以 2Hz 为增量单位)。

D：占空比

- 0.0% ～ 100.0%(以 0.1% 为增量单位，0000 ～ 03E8 Hex)

D 指定 PWM 输出的占空比，即输出为 ON 的时间所占的百分比。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
P	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
F，D	OK	OK	OK	OK	OK	OK	OK	OK	OK		OK		OK			

条件标志

名称	标记	操作
出错标志	P_ER	・当超出 P、F 或 D 的指定范围时 ON。 ・当正在使用 PWM(891) 以外的指令为指定端口输出 PWM 时 ON。 ・当在某个循环任务中正在执行一条控制PWM输出的指令时，若在某个中断任务中执行PWM(891)指令则 ON。 ・其它情况下 OFF。

功能

PWM(891) 指令以在 D 中指定的占空比、从在 P 中指定的端口输出在 F 中指定的频率。可在占空比 PWM 输出期间执行 PWM(891) 指令，从而在不停止 PWM 输出的情况下改变占空比。试图改变频率的操作将被忽略。

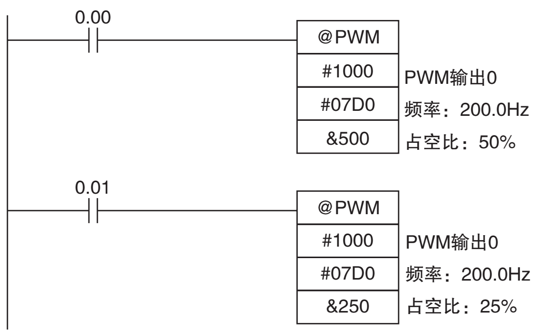
每次执行PWM(891)指令时均会启动PWM输出。因此，通常只需使用该指令的微分变化(@PWM(891))或只对一次扫描置 ON 的执行条件即可。

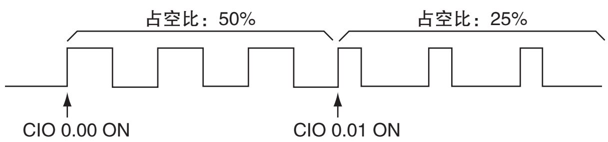
PWM 输出将一直持续到执行 INI(880) 指令使其停止 (C=0003 Hex: 停止 PWM 输出) 或者使 CPU 单元切换到 PROGRAM 模式为止。

注 PWM 指令只可用于晶体管输出型 CP1E N/NA □□型 CPU 单元和 CP2E N/S □□型 CPU 单元。
若为晶体管输出型 CP1E/CP2E E □□型 CPU 单元或继电器输出型，则适用 NOP 处理。

程序举例

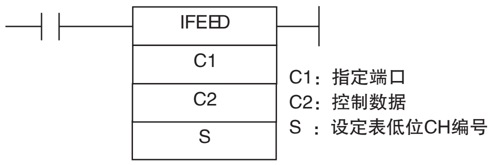
下例中，当 CIO 0.00 变 ON 时，PWM(891) 指令以 50% 的占空比从 PWM 输出 0 启动 200Hz 的 PWM 输出。当 CIO 0.01 变 ON 时，占空比变为 25%。





IFEED

指令	助记符	变化	功能代码	功能
中断进给	IFEED	@IFEED	892	将速度控制中的发生输入中断作为触发切换至位置控制，然后按指定脉冲量移动后减速停止。

符号	IFEED
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

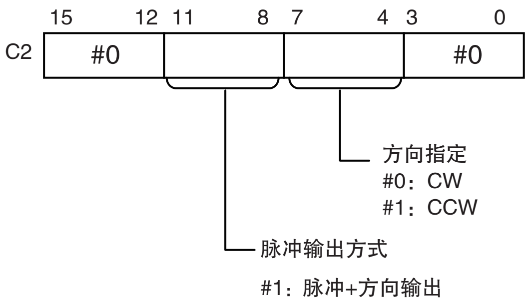
操作数	描述	数据类型	大小
C1	指定端口	–	1
C2	控制数据	–	1
S	设定表低位 CH 编号	WORD	6

C1：指定端口

指定端口 (16 进制)	
#0000	脉冲输出 0 – 输入中断 6
#0001	脉冲输出 1 – 输入中断 7
#0002	脉冲输出 2 – 输入中断 8
#0003	脉冲输出 3 – 输入中断 9

注 1 输入中断与脉冲输出的组合如上所述，为固定。无法变更。
2 CP1E 无法使用。
3 CP2E N □□型 14/20 点、E/S □□型无法使用脉冲输出 2、3。

C2 ：控制数据



S：设定表低位 CH 编号

S1	加速比率	1 ～ 65535Hz/4ms (16 进制 #0001 ～ FFFF)
S1+1	减速比率	1 ～ 65535Hz/4ms (16 进制 #0001 ～ FFFF)
S1+2	目标频率 (低位)	0 ～ 100000Hz(16 进制 #00000000 ～ 000186A0)
S1+3	目标频率 (高位)	
S1+4	脉冲输出量设定值 (低位)	脉冲输出量：0 ～ 2147483647(16 进制 #00000000 ～ 7FFFFFFF)
S1+5	脉冲输出量设定值 (高位)	

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C1，C2	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
S1	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK		OK			

条件标志

名称	标记	内容
错误标志	P_ER	・超出 C1、C2、S1 可指定的范围时 ON ・使用的输入中断的接点在 PLC 系统设定中未设定为输入中断时 ON ・周期执行任务在执行控制脉冲输出的指令过程中发生中断，在中断任务内执行了本指令时 ON ・除 IFEED 指令以外，对脉冲输出中的端口使用了本指令时 ON ・对基于 IFEED 指令的脉冲输出中的端口，通过本指令使用了目标频率 0 以外时 ON ・对非基于 IFEED 指令的脉冲输出中的端口，通过本指令使用了目标频率 0 时 ON ・使用的输入中断为允许中断状态时 ON ・S 指定的脉冲输出量小于通过指定动作可减速停止的脉冲数时 ON ・除此以外时，OFF

功能

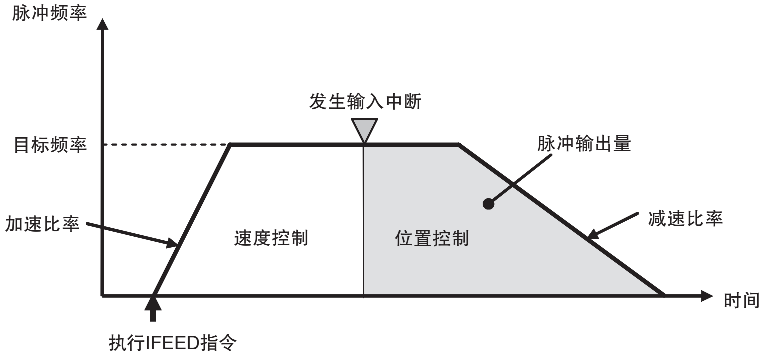
根据 C2 指定的方式，从 C1 指定的脉冲输出端口开始脉冲输出。并且，根据 S 指定的 “加速比率” 进行加速直至达到 “目标频率”，并使用连续模式执行速度控制。
然后，在发生 C1 指定的输入中断时 (注) 切换至位置控制，根据 S 指定的 “减速比率” 进行减速，并输出 S 指定的 “脉冲输出量设定值” 的脉冲量进行停止。

注　对中断输入接点的允许中断 (直接模式) 在本指令中执行，因此无需执行 MSKS 指令。此外，即使存在中断任务也不会启动。但需在中断输入接点的下降沿 (ON → OFF) 发生输入中断时，请在 IFEED 指令前执行 MSKS 指令 (中断输入的边缘检测方向指定)。不通过 MSKS 指令指定 “上升沿” 时，IFEED 指令将按照在中断输入接点的上升沿 (OFF → ON) 发生输入中断的设定进行动作。

本指令组合特定的脉冲输出和输入中断进行控制，不使用中断任务，使用 1 个指令执行中断标准移动的设定、启动。

与启动中断任务并从中执行 PLS2 指令相比，中断响应性更高。但脉冲输出和输入中断的组合为固定，无法变更。此外，基于本 IFEED 指令的脉冲输出开始后，将无法执行 INI 指令 (脉冲输出停止) 以外的脉冲输出指令。执行 INI 指令 (脉冲输出停止) 时，在中断脉冲输出的同时输入中断也将进入中断禁止状态。再次执行 IFEED 指令时，将从最初开始输出脉冲。

需任意组合脉冲输出和输入中断以及需在脉冲输出中变更设定时，使用 ACC 指令和 PLS2 指令。



注意事项

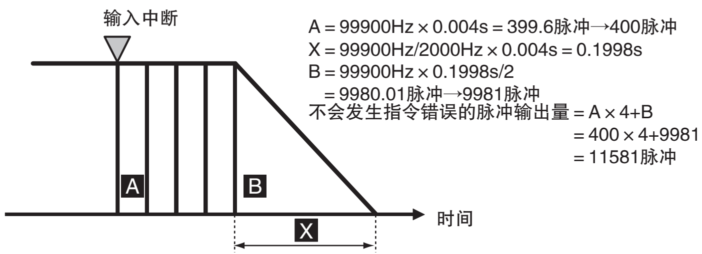
执行 IFEED 指令时，操作数 S 指定的脉冲输出量小于根据指定参数可减速停止的脉冲数时，将发生指令错误。

根据目标速度、减速比率，脉冲输出量请设定成大于以下计算公式求出的脉冲数。

不会发生指令错误的脉冲输出量 = 目标速度时 1 个脉冲控制周期中的脉冲数 * × 4
+ 从目标速度减速所需的脉冲数 *

* 小数点后进位

例：目标速度 99900Hz
减速比率 2000Hz/4ms
脉冲控制周期 4ms



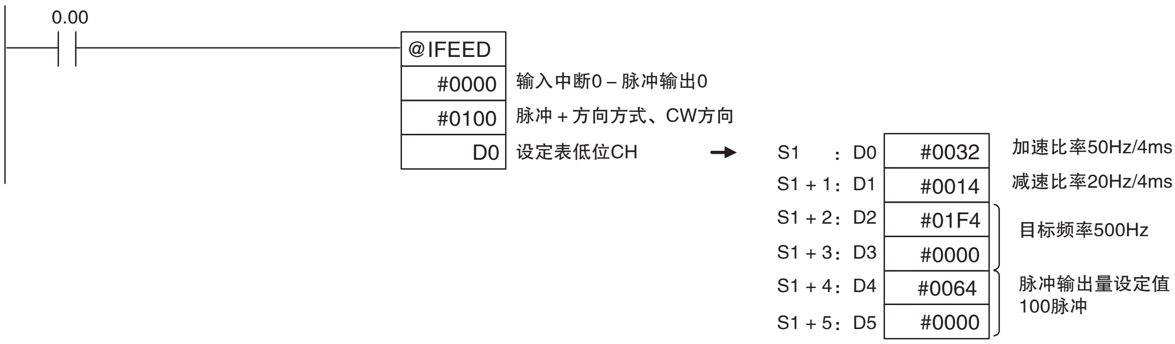
・执行 IFEED 指令时，在指定的中断输入任务解除的情况下，请事先使用 MSKS 指令将中断输入设定为禁止状态。在中断任务解除状态下执行 IFEED 指令时，将发生指令错误。

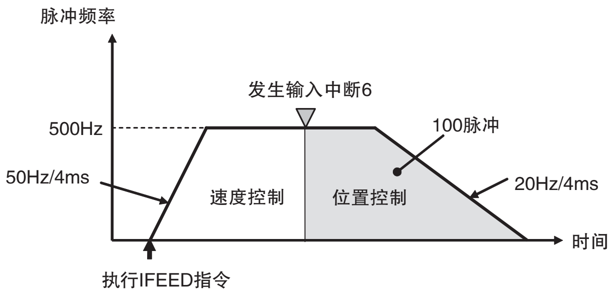
・IFEED 指令使用中断输入 6、7、8、9。中断输入 6、7、8、9 是与脉冲输出的原点输入兼用的端子，因此通过脉冲输出使用 IFEED 指令并使用中断标准移动时，无法使用原点搜索功能。

注 IFEED 指令仅 CP2E CPU 单元 N □□型晶体管输出型可用。
CP1E 无法使用。CP2E E/S □□型或继电器输出型时，为 NOP 处理。

程序举例

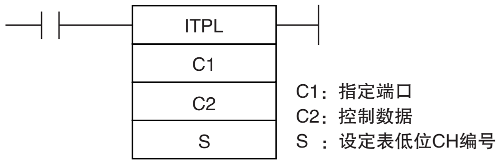
0.00 为 OFF → ON 时，从脉冲输出 0 开始输出。以 50Hz/4ms 加速，达到目标频率 500Hz 时，以连续模式的固定速度执行速度控制。发生输入中断 6 时，切换至位置控制，以 20Hz/4ms 开始减速，并根据脉冲输出量执行设定值的脉冲输出进行停止。





ITPL

指令	助记符	变化	功能代码	功能
直线插补	TPL	@ITPL	893	CP2E-N 型使用指定的脉冲输出端口执行 2 ～ 4 轴直线插补。

符号	ITPL
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

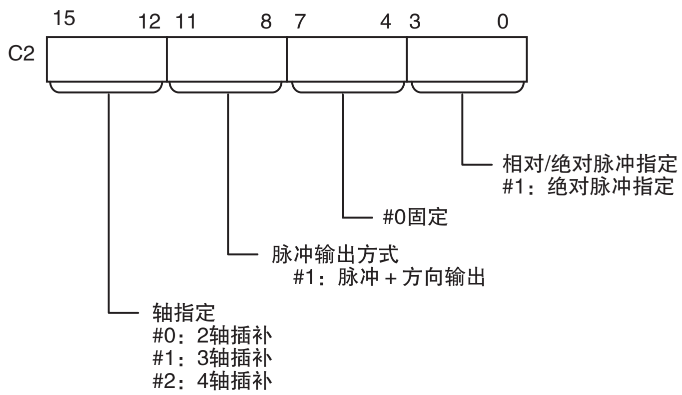
操作数

操作数	描述	数据类型	大小
C1	指定端口	–	1
C2	控制数据	–	1
S1	设定表低位 CH 编号	WORD	14

C1：指定端口

指定端口 (16 进制)	
#0030	直线插补 0
#0031	直线插补 1

C2 ：控制数据



注 1 2 轴插补在直线插补 0 中使用脉冲输出 0/1，在直线插补 1 中使用脉冲输出 2/3。
2 3 轴插补仅可在直线插补 0 中执行。使用脉冲输出 0/1/2。
3 4 轴插补仅可在直线插补 0 中执行。使用脉冲输出 0/1/2/3。
4 CP2E N □□型 14/20 点无法使用直线插补 0 的 3 轴插补 /4 轴插补和直线插补 1。
5 CP1E、CP2E E/S □□型无法使用。

S1：设定表低位 CH 编号

	15　　　　　　0	
S1	加速比率	1 ~ 65535Hz(16进制#0001 ~ FFFF)
S1+1	减速比率	

分别以1Hz为单位指定每4ms的频率增减量。

S1+2	目标频率(低位)	1 ~ 100000Hz (16进制#00000001 ~ 000186A0)
S1+3	目标频率(高位)	

以1Hz为单位指定加速后的频率。

	15　　　　　　0	
S1+4	启动频率(低位)	0 ~ 100000Hz (16进制#00000000 ~ 000186A0)
S1+5	启动频率(高位)	

以1Hz为单位指定启动时的频率。

S1+6	目标位置0(低位)	–2147483648 ~ +2147483647(16进制#80000000 ~ 7FFFFFFF)
S1+7	目标位置0(高位)	
S1+8	目标位置1(低位)	–2147483648 ~ +2147483647(16进制#80000000 ~ 7FFFFFFF)
S1+9	目标位置1(高位)	
S1+10	目标位置2(低位)	–2147483648 ~ +2147483647(16进制#80000000 ~ 7FFFFFFF)
S1+11	目标位置2(高位)	
S1+12	目标位置3(低位)	–2147483648 ~ +2147483647(16进制#80000000 ~ 7FFFFFFF)
S1+13	目标位置3(高位)	

以脉冲单位指定直线插补的目标位置。

- 使用直线插补端口 0 执行 2 轴插补时，请在目标位置 0 中设定脉冲输出 0 的位置，在目标位置 1 中设定脉冲输出 1 的位置。
- 使用直线插补端口 1 执行 2 轴插补时，请在目标位置 0 中设定脉冲输出 2 的位置，在目标位置 1 中设定脉冲输出 3 的位置。
- 3 轴插补时，请在目标位置 0 中设定脉冲输出 0 的位置，在目标位置 1 中设定脉冲输出 1 的位置，在目标位置 2 中设定脉冲输出 2 的位置。
- 4 轴插补时，请在目标位置 0 中设定脉冲输出 0 的位置，在目标位置 1 中设定脉冲输出 1 的位置，在目标位置 2 中设定脉冲输出 2 的位置，在目标位置 3 中设定脉冲输出 3 的位置。

● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>C1，C2</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>OK</td><td rowspan="2">---</td><td rowspan="2">---</td><td>---</td><td rowspan="2">---</td><td rowspan="2">---</td><td rowspan="2">---</td></tr>
<tr><td>S1</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td></tr>
</table>

条件标志

名称	标记	内容
错误标志	P_ER	· 超出 C1、C2、S 可指定的范围时 ON · 轴指定所指定的脉冲输出端口正在使用 SPED、ACC、PLS2、ORG、IFEED、ITPL 指令输出脉冲时 ON · 轴指定所指定的脉冲输出端口未确定原点时 ON · 周期执行任务在执行控制脉冲输出的指令过程中发生中断，在中断任务内执行了本指令时 ON · 合成移动量超出约 100000000 脉冲时 ON · 动作时间超出约 1000 秒时 ON · 除此以外时，OFF

功能

使用脉冲输出端口 0 ～ 3 执行 2 ～ 4 轴的直线插补动作。

最多可同时执行2个直线插补。使用直线插补端口以区分各自的动作。请使用C1指定直线插补端口。

构成直线的轴请使用 C2 的轴指定进行指定。直线插补端口与轴指定的关系如下表所示。

指定端口	2 轴插补	3 轴插补	4 轴插补
直线插补 0	脉冲输出端口 0/1	脉冲输出端口 0/1/2	脉冲输出端口 0/1/2/3
直线插补 1	脉冲输出端口 2/3	−	−

执行指令时，从当前位置至目标位置输出直线状动作的脉冲。各轴的目标位置请以前移方式指定至 S+6 ～ S+13。可使用 S 指定加速比率 (Hz/4ms)，使用 S+1 指定减速比率 (Hz/4ms)，使用 S+2 指定目标频率 (Hz)，使用 S+4 指定启动频率 (Hz)。这些参数全部请对当前位置至目标位置的直线进行指定。构成直线的各轴参数为自动计算。

参数有以下限制。

· 可通过直线插补移动的距离和动作时间存在上限。直线的距离请指定为 100000000 脉冲以下，动作时间请指定为 1000 秒以下。

· 请指定成启动频率≤目标频率。启动频率 > 目标频率时，按目标频率进行动作。

ITPL 指令执行 1 次时，按指定条件开始脉冲输出。因此，基本上按输入微分型 (带 @) 或 1 个周期 ON 的输入条件进行使用。

基于 ITPL 指令的直线插补仅可通过绝对脉冲指定进行定位。因此，在执行 ITPL 指令前请先确定原点。

执行 ITPL 指令中的停止可使用 INI 指令立即停止、减速停止。减速停止时将停止在直线插补的轨迹上。

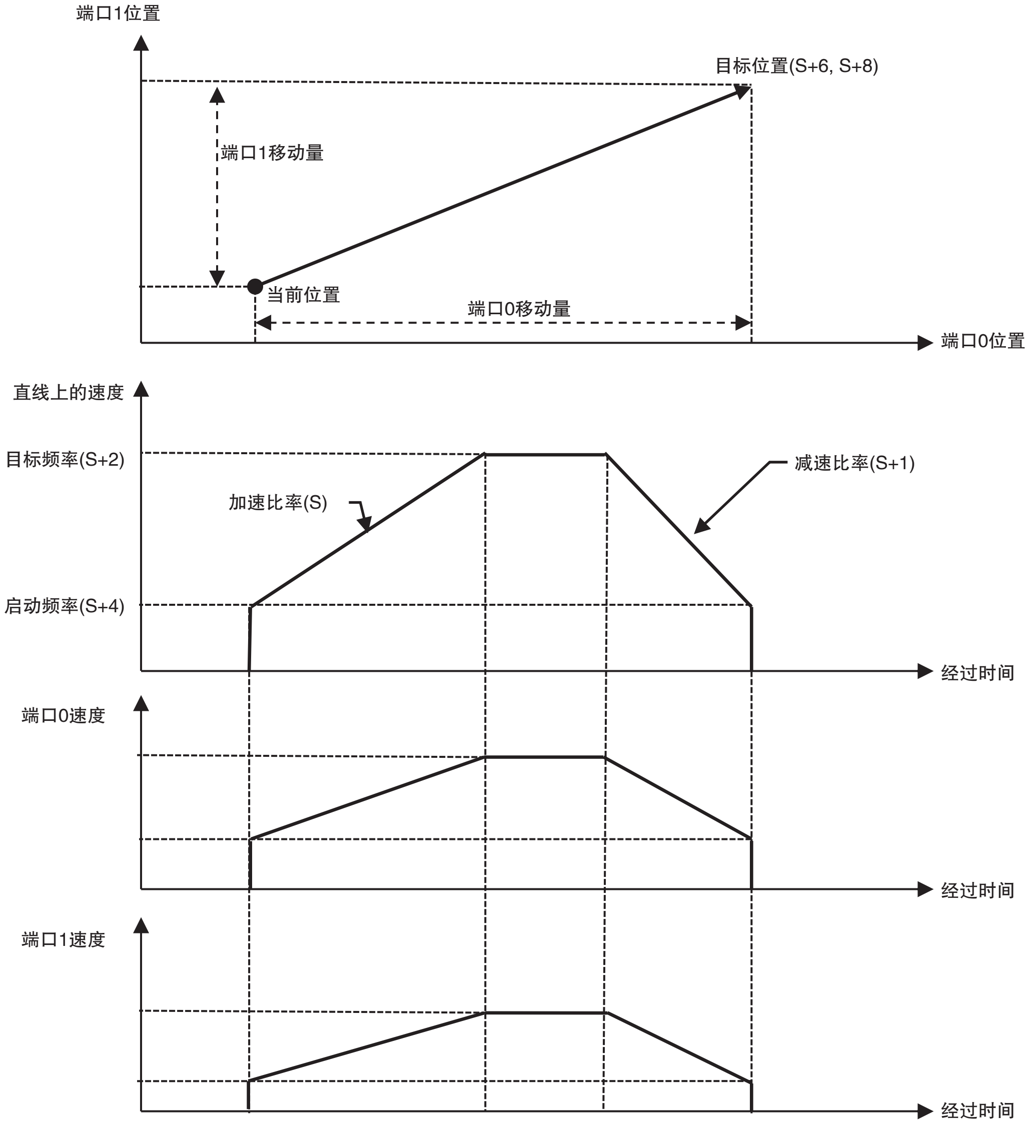
*1 ITPL 指令仅 CP2E CPU 单元 N □□型晶体管输出型可用。
CP1E 无法使用。CP2E E/S □□型或继电器输出型时，为 NOP 处理。

*2 执行 ITPL 指令中的脉冲输出端口无法执行其他脉冲输出指令 (SPED、ACC、PLS2、ORG、IFEED 指令)。

*3 执行 ITPL 指令的过程中，无法再次执行 ITPL 指令变更目标位置。
请执行减速停止、立即停止 (INI 指令)，在输出停止后再次执行 ITPL 指令。

*4 发生运行停止原因时，动作中的直线插补将立即停止。负载切断标志 (A500.15)ON 的过程中，也将立即停止。此外，在负载切断标志 ON 的过程中执行了指令时，将在脉冲不输出的状态下以 P_ER=OFF 结束。

通过端口 0 和端口 1 执行 2 轴直线插补的示例如下所示。



*：端口0、端口1的速度、移动量在CP2E内部自动计算。

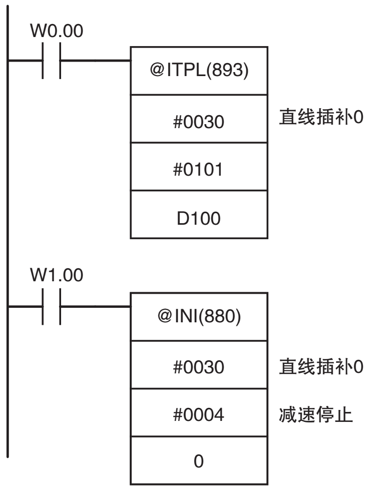
注意事项

输出速度为 125Hz 以下时，每个周期的脉冲数小于 1 脉冲，因此不在每个周期执行脉冲输出。这可能会导致脉冲的占空比非 50%。

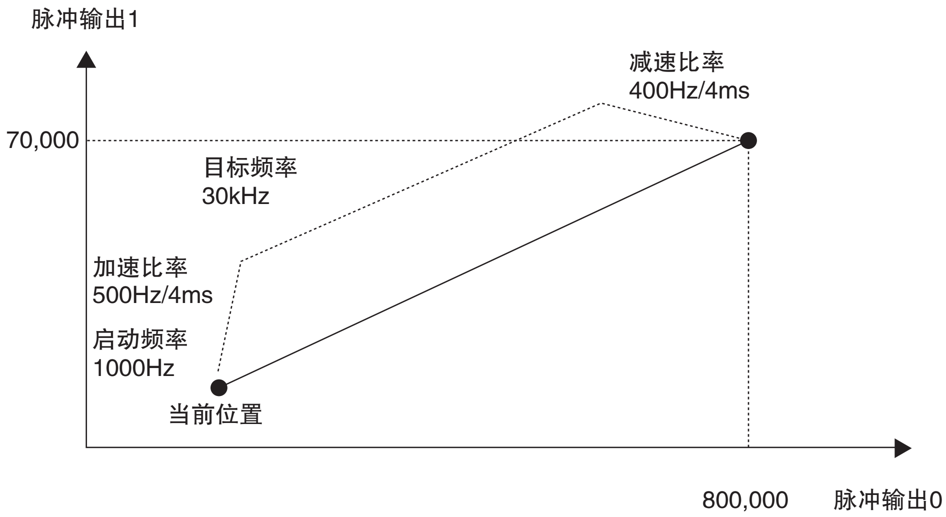
示例程序

W0.00 为 OFF → ON 时，使用 ITPL 指令，通过使用脉冲输出 0、脉冲输出 1 的直线插补进行定位。将 W1.00 设为 ON，对直线插补进行减速停止。

*：需在执行 ITPL 指令前确定原点。

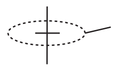
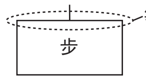


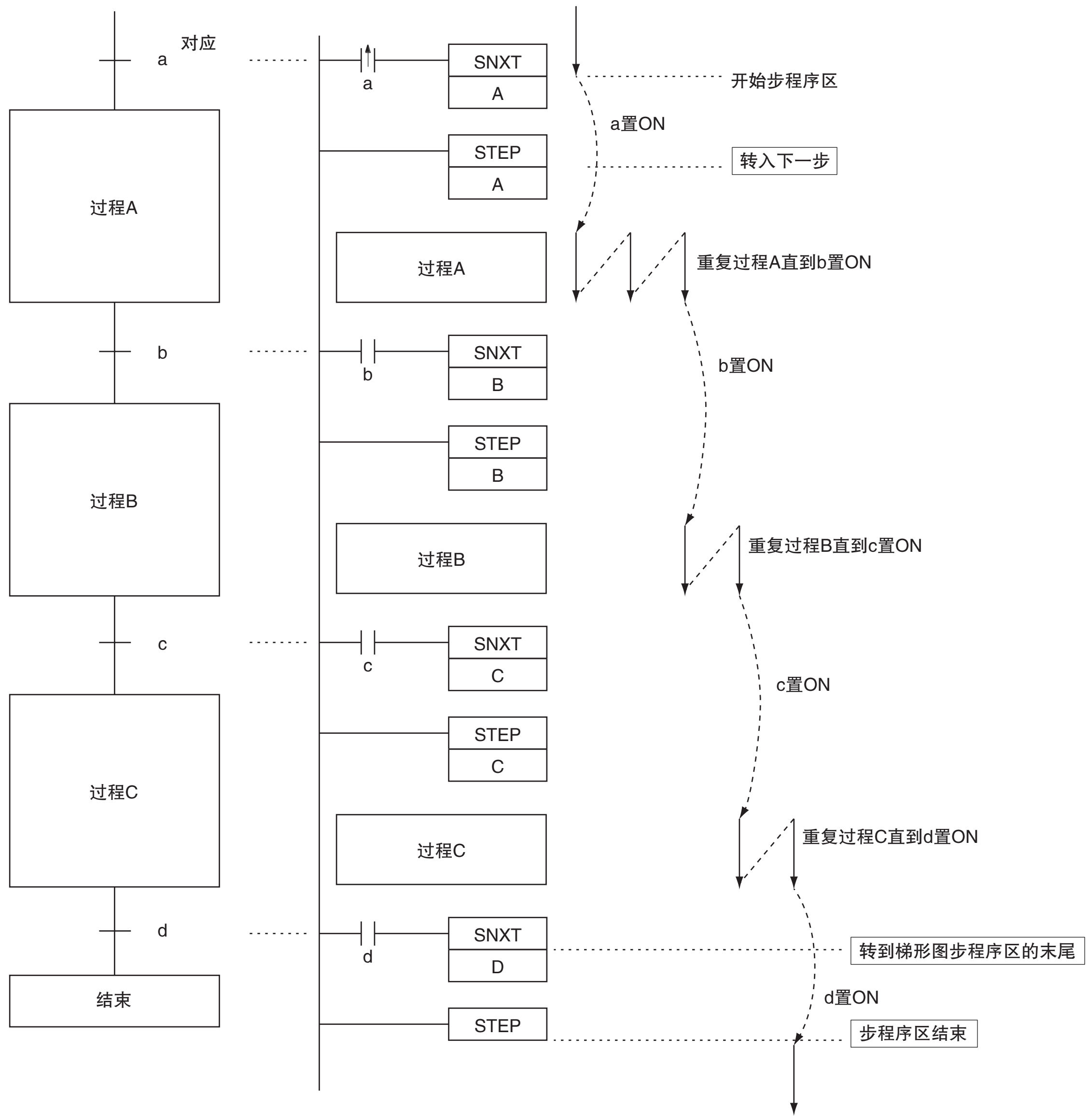
D100	01F4	加速比率500Hz/4ms
D101	0190	减速比率 400Hz/4ms
D102	7530	目标频率30kHz
D103	0000	
D104	03E8	启动频率 1000Hz
D105	0000	
D106	3500	目标位置 800,000
D107	000C	
D108	1170	目标位置 70,000
D109	0001	



步指令

在 CP1E/CP2E 系列 PLC 中，可一起使用 STEP(008)/SNXT(009) 指令来创建步程序。

指令	操作	图
SNXT(009)：步启动	控制进程转入下一步程序。	步梯形图区开始指令 
STEP(008)：步 定义	表示一个步的开始。重复同一步程序，直至转入下一步的进程条件成立为止。	步梯形图区开始指令 



注 将工作位用作 A、B、C 和 D 的控制位。

SNXT/STEP

指令	助记符	变化	功能代码	功能
步启动	SNXT	---	009	SNXT(009) 指令用于控制步程序区中步执行的进程。
步定义	STEP	---	008	STEP(008) 指令用于定义步程序区的开始和结束。

	SNXT	STEP
符号	SNXT(009) B B：位	定义步的开始时，需指定控制位，如下所示： STEP(008) B B: 位 定义步的结束时，不指定控制位，如下所示： STEP(008)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	OK	---	---

操作数

操作数	描述	数据类型	大小
B	位	---	1

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
B	---	OK	---	---	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

名称	标记	操作	
		SNXT	STEP
出错标志	P_ER	· 当指定位 B 不在 WR 区内时为 ON。 · 当在中断程序中使用SNXT(009)指令时ON。 · 其它情况下 OFF。	· 当指定位 B 不在 WR 区内时为 ON。 · 当在中断程序中使用STEP(008)指令时ON。 · 其它情况下 OFF。

功能

● SNXT(009)

SNXT(009) 指令用于以下 3 种情况：

1. 开始步程序的执行。
2. 进到下一个步控制位。
3. 结束步程序执行。

步程序区从第一条 STEP(008) 指令 (总是带一个控制位) 开始，到最后一条 STEP(008) 指令 (不带控制位) 结束。

启动步程序的执行

将 SNXT(009) 指令放在步程序区的起始处，用于启动步程序的执行。该指令将为下一条 STEP(008) 指令的步 B 指定的控制位置 ON，并进到步 B(STEP(008) B 指令之后的所有指令)。对于启动步程序区执行的 SNXT(009) 指令，必须使用指令的微分执行条件，否则步执行将只持续一个循环。

转入下一步

出现在步程序区中间的 SNXT(009) 指令用于转入下一步执行。该指令对先前的控制位置 OFF，并将对下一步 B 的下一个控制位置 ON，从而启动步 B 的执行 (STEP(008) B 指令之后的所有指令)。

结束步程序区

将 SNXT(009) 指令放在每一个步程序区的结尾处，用于结束步程序的执行并对先前的控制位置 OFF。为 B 指定的控制位是一个虚位，但这个位将被置 ON，因此请务必选择一个不会产生问题的位。

● STEP(008)

STEP(008) 指令视指令的位置和是否指定了控制位而定，有以下 2 个作用：

1. 开始指定步。
2. 结束步程序区 (例如步执行)。

启动一个步

将 STEP(008) 指令放在各个步的起始处，并带有一个操作数 B，用作该步的控制位。

控制位 B 将被 SNXT(009) 指令置 ON，且将从该步中紧随 STEP(008) 指令之后的指令开始执行。当开始步的执行时，A200.12(步标志) 将置 ON。

在第一个循环之后，步的执行将一直持续到改变步的条件成立为止，即直到 SNXT(009) 指令将下一条 STEP(008) 指令的控制位置 ON 为止。

当 SNXT(009) 指令对某个步的控制位置 ON 时，当前指令的控制位 B 将被复位 (置 OFF)，且由位 B 控制的步将被互锁。

某个步中的输出和指令的处理将根据控制位B的ON/OFF状态而改变。(控制位的状态受SNXT(009)指令控制。) 当控制位 B 被置 OFF 时，步中的指令将被复位和互锁。详情请参阅下表。

控制位状态	处理
ON	步中的指令将正常执行。
ON → OFF	步中的位和指令将被互锁，如下表所示。
OFF	步中的所有指令将作为 NOP 处理。

互锁状态 (IL)

指令输出		状态
为 OUT、OUT NOT 指定的位		全部 OFF
TIM、TIMX(551)、TIMH(015)、TIMHX(551)、TMHH(540)、TIMHHX(552)、TIML(542) 和 TIMLX(553)	PV	0000 Hex(复位)
	完成标志	OFF(复位)
为其它指令指定的位或字 (见注)		保持先前的状态 (但不执行指令)

注 表示所有其它指令，如 TTIM(087)、TTIMX(555)、SET、REST、CNT、CNTX(546)、CNTR(012)、CNTRX(548)、SFT(010) 和 KEEP(011)。

在各个步的起始位置必须放一条 STEP(008) 指令。在步程序区的起始位置放一条 STEP(008) 指令，用于定义步的开始。

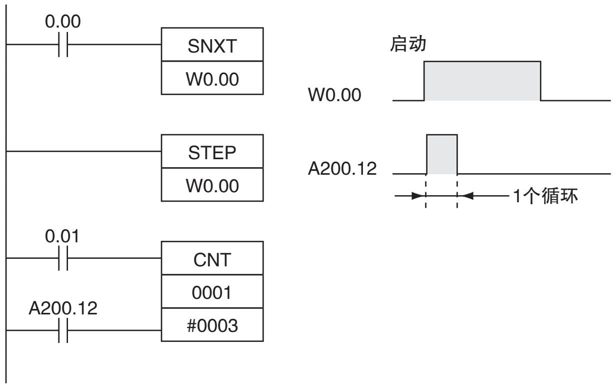
结束步程序区

在步程序区的末尾处放一条不带操作数的 STEP(008) 指令，用于定义步程序的结束。

当用于 SNXT(009) 指令之前的控制位被置 OFF 时，步执行将被 SNXT(009) 指令停止。

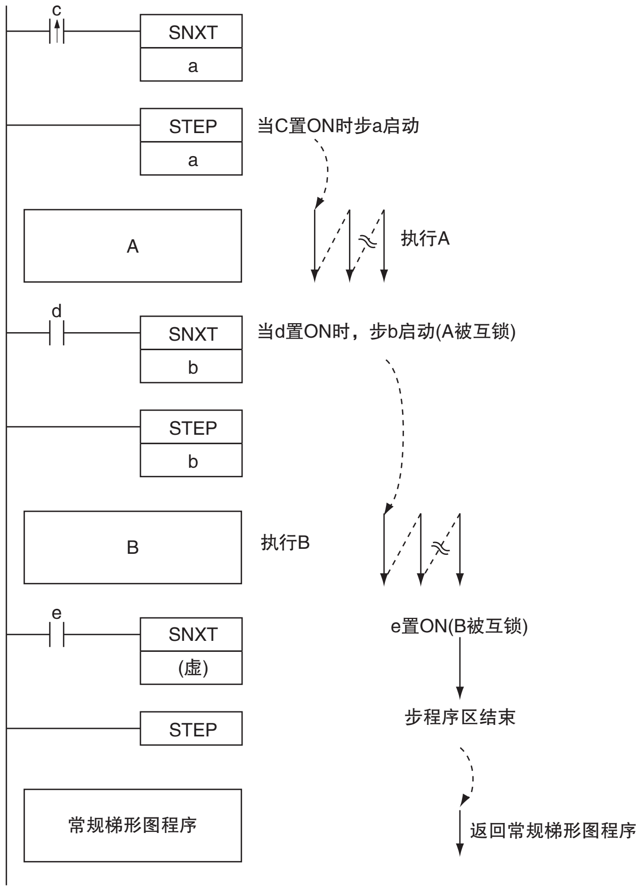
提示

执行 STEP(008) 指令时，A200.12(步标志) 将在一个循环中置 ON。该标志可用于在步执行开始时执行初始化。



相关位

名称	地址	详细说明
步标志	A200.12	当通过 STEP(008) 指令起动步程序时，对一个循环置 ON。可用于在启动一个新的步时复位定时器和执行其它处理。



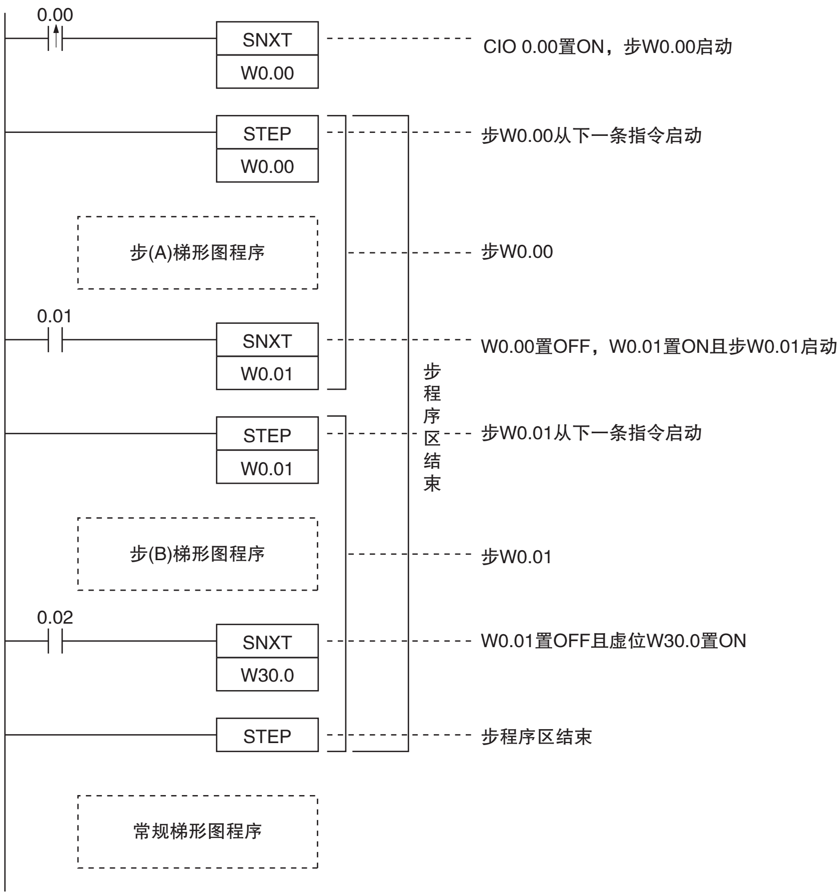
注意事项

- 控制位 B 必须位于 STEP(008)/SNXT(009) 指令的工作区内。
- STEP(008)/SNXT(009) 指令的控制位不能用在除梯形图以外的任何位置。如果同一个位使用了两次，将产生重复位错误。
- 如果使用 SBS(091) 指令从某个步内调用某个子程序，则当控制位被置 OFF 时，该子程序的输出和指令将不会被互锁。
- 执行条件 ON 时将执行 SNXT(009) 指令。

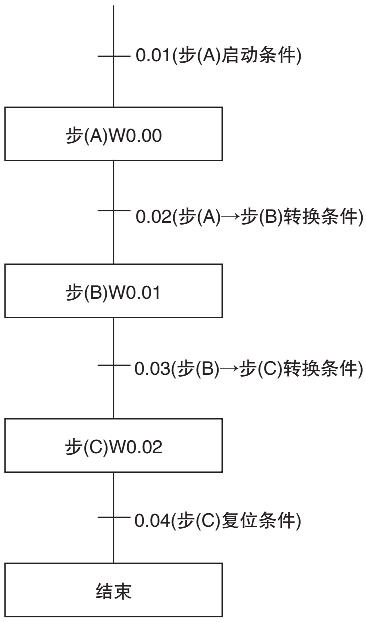
・在步程序区的末尾输入 SNXT(009) 指令，并确保控制位为工作区内的虚位。如果在步程序区的最后一条 SNXT(009) 指令中使用某个步的控制位，则在执行 SNXT(009) 指令时将启动相应的步。
・STEP(008) 和 SNXT(009) 指令不能在子程序、中断程序或块程序内部使用。
・请务必确保同一个循环期间不执行两个步。
・下表中列出了不能在步程序内部使用的指令。

功能	助记符	名称
顺序控制指令	END(001)	结束
	IL(002)	互锁
	ILC(003)	互锁清除
	JMP(004)	跳转
	JME(005)	跳转结束
	CJP(510)	条件跳转
子程序指令	SBN(092)	子程序入口
	RET(093)	子程序返回

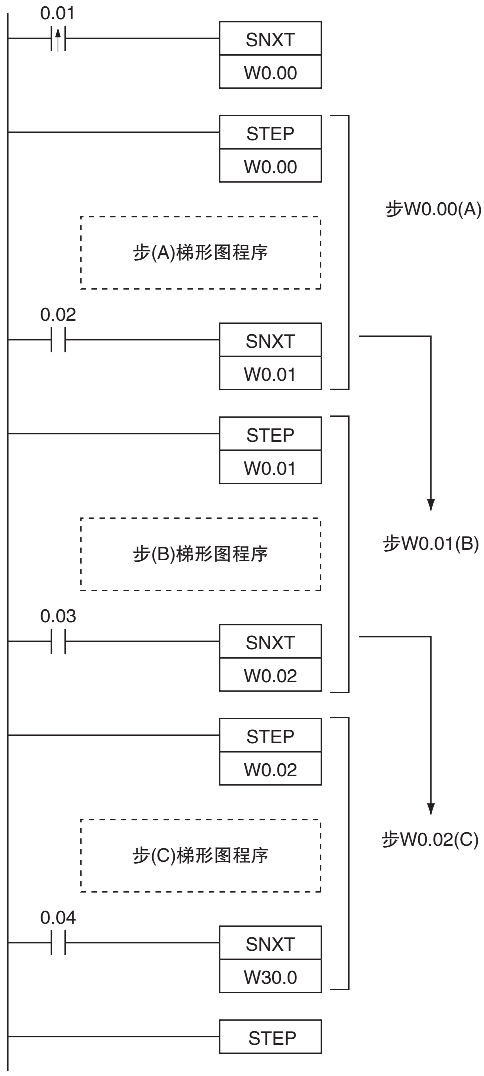
程序举例



(1) 顺序控制

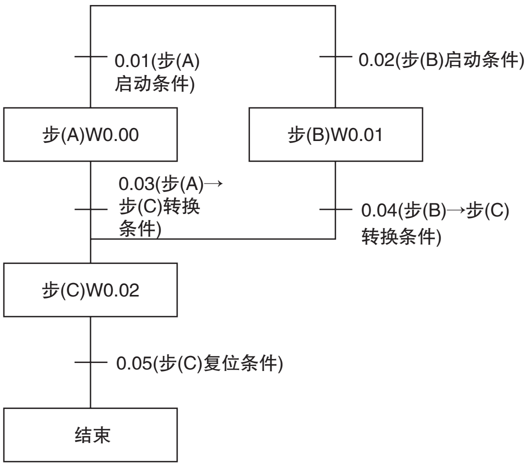


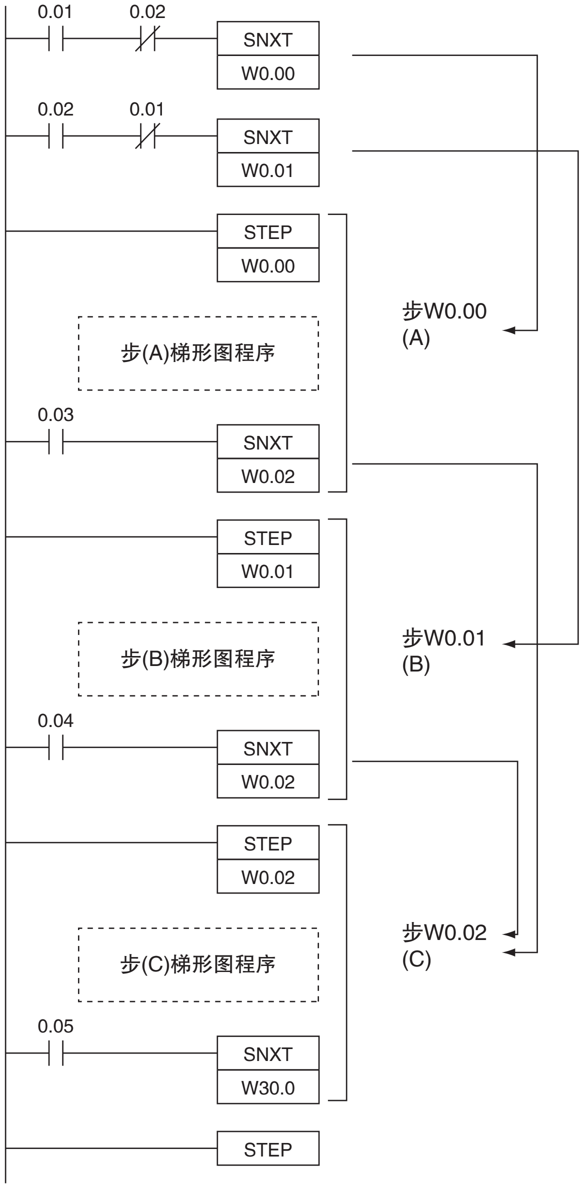
0.01(步(A)启动条件)
步(A)W0.00
0.02(步(A)→步(B)转换条件)
步(B)W0.01
0.03(步(B)→步(C)转换条件)
步(C)W0.02
0.04(步(C)复位条件)
结束



0.01
SNXT
W0.00
STEP
W0.00
步(A)梯形图程序
0.02
SNXT
W0.01
步W0.00(A)
STEP
W0.01
步(B)梯形图程序
0.03
SNXT
W0.02
步W0.01(B)
STEP
W0.02
步(C)梯形图程序
0.04
SNXT
W30.0
步W0.02(C)
STEP

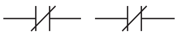
(2) 分支控制



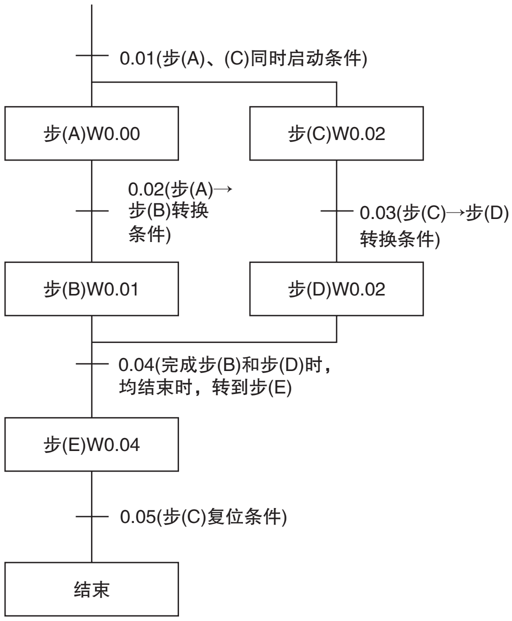


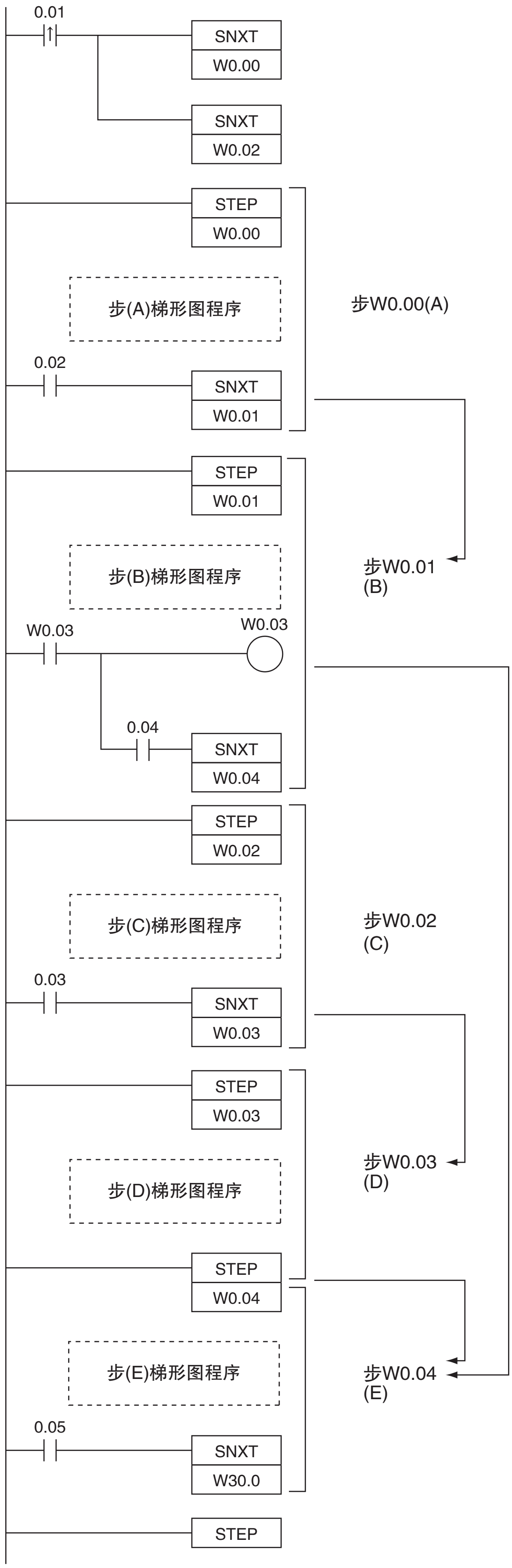
● 附加信息：

- 上例中，对 W0.02 执行 SNXT(009) 指令时，尽管同一个控制位使用了两次，分支仍移至下一步。使用 CX-Programmer 时，该现象不会作为程序错误进行检测。只有当步指令中的控制位同时也用于常规梯形图程序中时，步梯形图程序中才会发生重复位错误。
- 当步A和B不能同时执行时，可使用上述程序。若要同时执行A和B，请删除下图所示的执行条件。



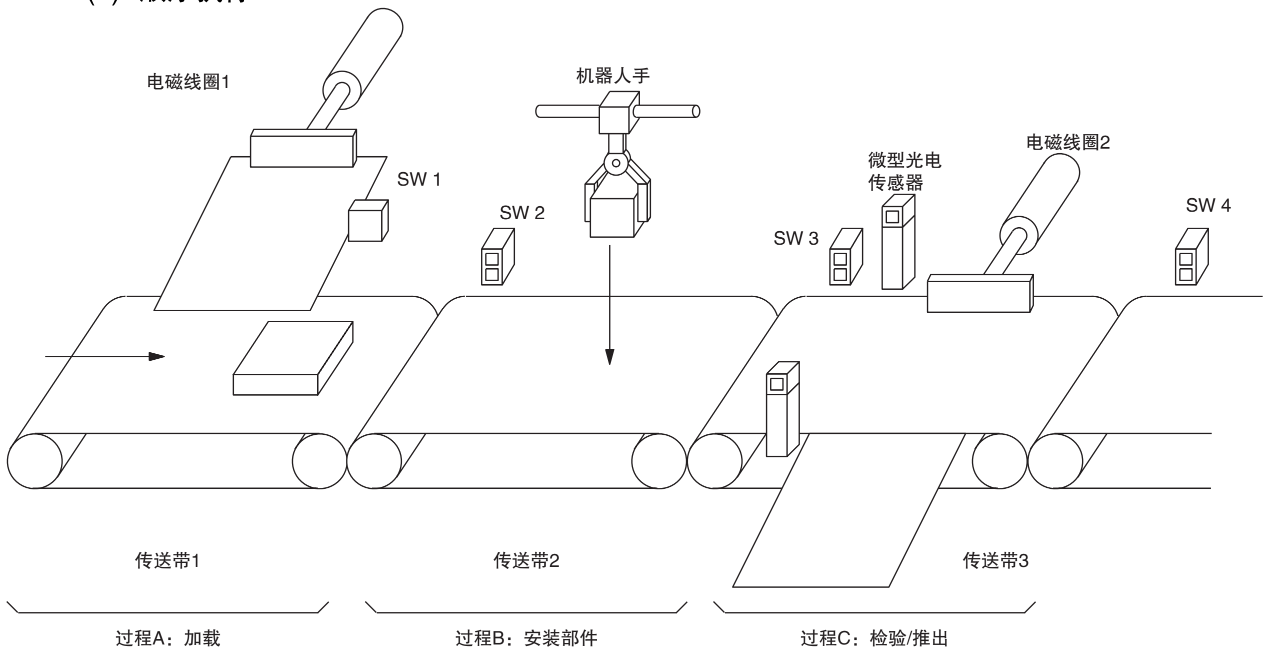
(3) 并行控制

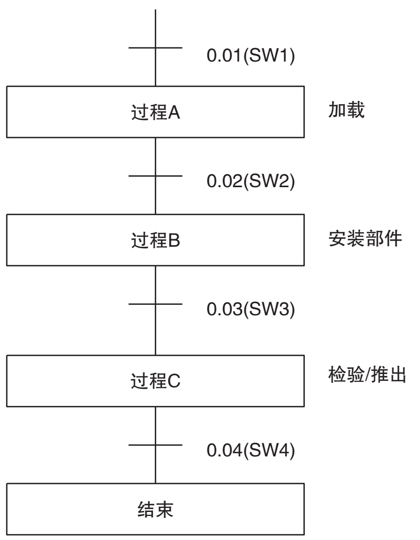

0.01(步(A)、(C)同时启动条件)
步(A)W0.00
步(C)W0.02
0.02(步(A)→步(B)转换条件)
0.03(步(C)→步(D)转换条件)
步(B)W0.01
步(D)W0.02
0.04(完成步(B)和步(D)时，均结束时，转到步(E)
步(E)W0.04
0.05(步(C)复位条件)
结束

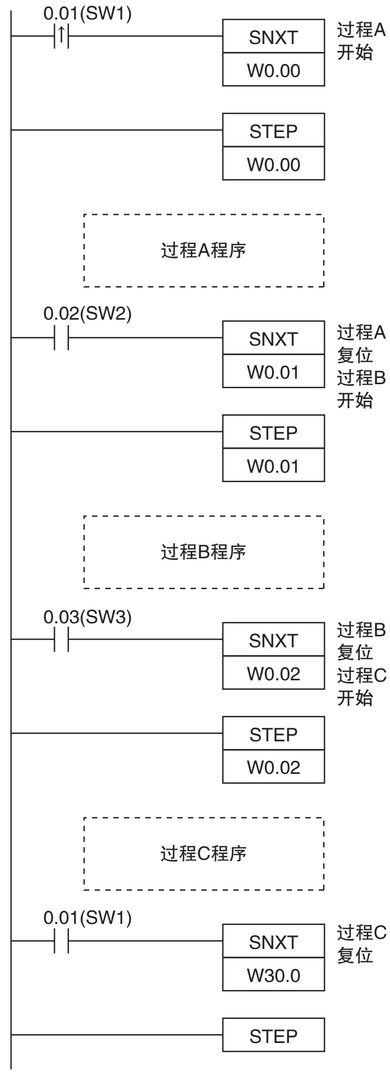

0.01
SNXT
W0.00
SNXT
W0.02
STEP
W0.00
步(A)梯形图程序
0.02
SNXT
W0.01
步W0.00(A)
STEP
W0.01
步(B)梯形图程序
步W0.01 (B)
W0.03
W0.03
0.04
SNXT
W0.04
STEP
W0.02
步(C)梯形图程序
步W0.02 (C)
0.03
SNXT
W0.03
STEP
W0.03
步(D)梯形图程序
步W0.03 (D)
STEP
W0.04
步(E)梯形图程序
步W0.04 (E)
0.05
SNXT
W30.0
STEP

应用举例

(1) 顺序执行







操作说明

(1) SW1 ON：
  电磁线圈1动作
  传送带1动作 } 过程A

(2) SW2 ON停止先前的过程
  机器人手动作
  传送带2动作 } 过程B

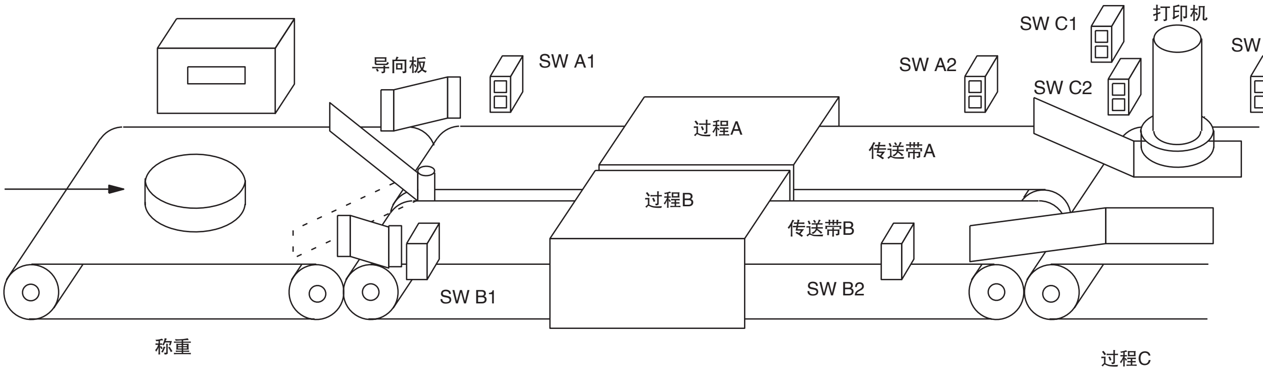
(3) SW3 ON停止先前的过程
  光电传感器动作(进行部件检查)
  传送带3动作
  电磁线圈2动作(推出有缺陷的项) } 过程C

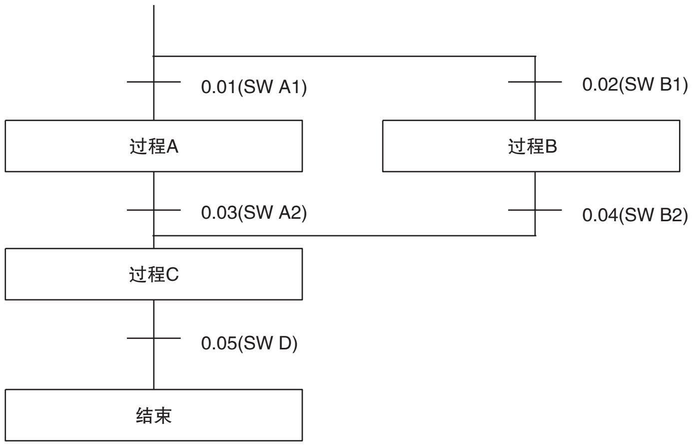
(4) SW4 ON停止先前的过程

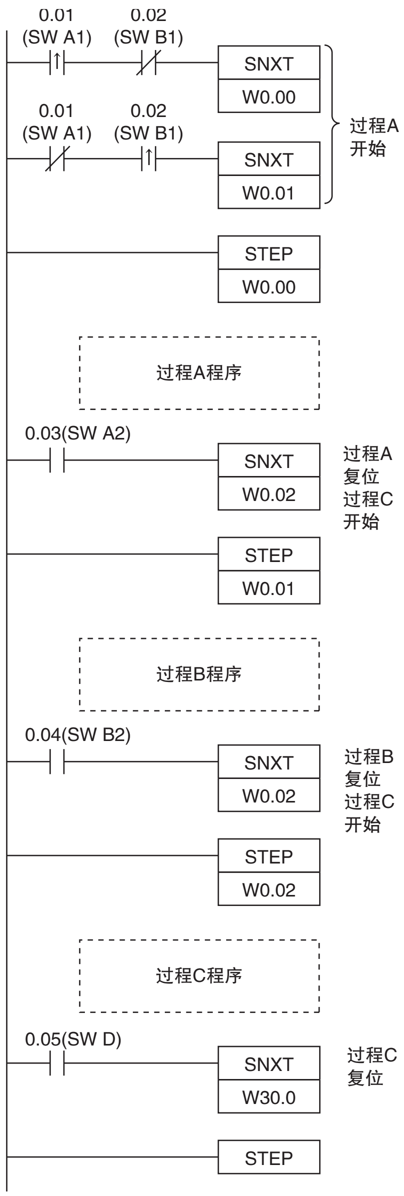
附加信息

如果从某个过程内部启动了另一个过程(当一条SNXT指令置ON时)，则当前过程的所有输出均在该循环中置OFF。

(2) 分支执行







操作说明

产品按照重量由导向板进行排序

(1) SWA1 ON：
传送带(A)动作
机器(A)动作 } 过程A

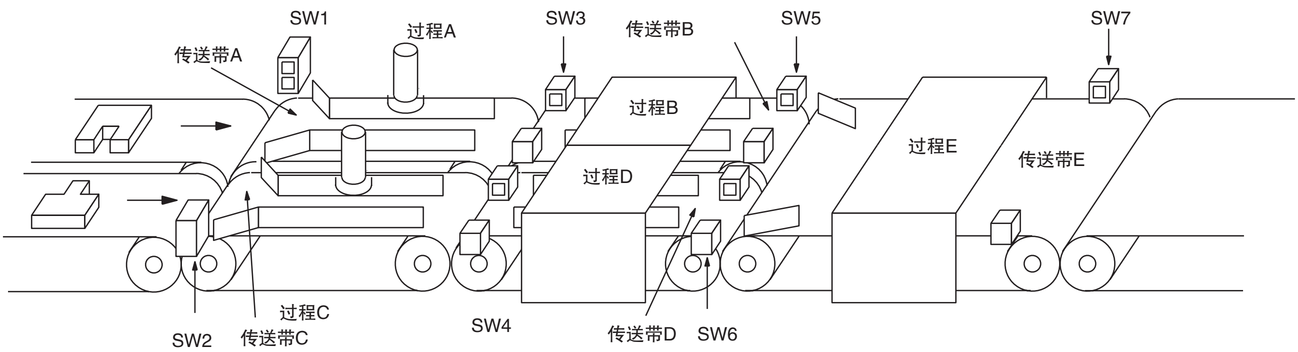
(2) SWB1 ON：
传送带(B)动作
机器(B)动作 } 过程B

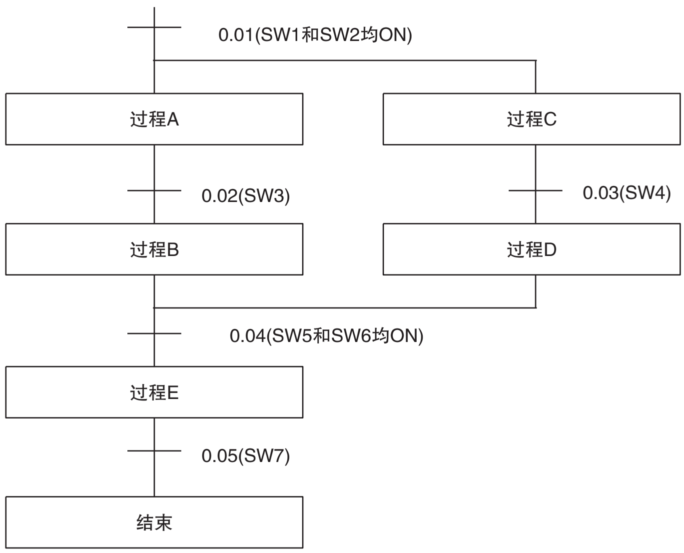
(3) SWA2 ON停止过程A
打印机动作(下降)
通过SWC2 ON上升 } 过程C

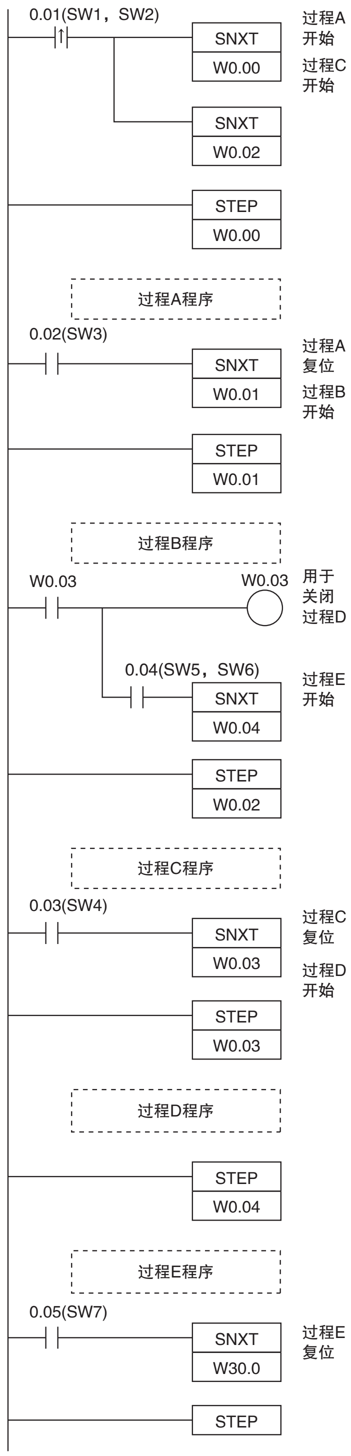
(4) SWB2 ON停止过程B
打印机动作(下降)
通过SWC2 ON上升 } 过程C

(5) SWD ON停止打印机

(3) 并行执行







操作说明

(1) SW1、SW2 ON：
- 传送带(A)动作 } 过程A
- 工件(A)动作
- 传送带(C)动作 } 过程C
- 工件(C)动作

(2) SW3 ON：
- 过程(A)停止 } 过程B
- 传送带(B)动作
- 工件(B)动作

(3) SW4 ON：
- 过程(C)停止 } 过程D
- 传送带(D)动作
- 工件(D)动作

(4) SW5、SW6 ON：
- 过程(B)停止 } 过程E
- 过程(D)停止
- 传送带(E)动作
- 工件(E)动作

(5) SW7 ON：
- 过程(E)停止

注： 当过程(B)和(D)正在执行且SW5和SW6置ON时，将判断为(B)和(D)已结束。
SNXT W0.04的执行将对过程(B)的输出置OFF，且W0.03置OFF。
STEP W0.03判断为ON→OFF，且对过程(D)的输出置OFF。

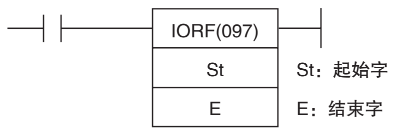
附加信息

当由ON→OFF时，STEP指令将使该过程中的所有输出置OFF。

基本 I/O 单元指令

IORF

指令	助记符	变化	功能代码	功能
I/O 刷新	IORF	@IORF	097	刷新指定的 I/O 字。

符号	IORF
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	---	OK	OK	OK

操作数

操作数	描述	数据类型	大小
St	起始字	---	可变
E	结束字	---	可变

St：起始字

CIO 001 ～ CIO 099、CIO 101 ～ CIO 199(CP1W 扩展 I/O 单元的 I/O 区)

E：结束字

CIO 001 ～ CIO 099、CIO 101 ～ CIO 199(CP1W 扩展 I/O 单元的 I/O 区)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
St，E	OK	---	---	---	---	---	---	---	---	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・St 大于 E 时 ON。 ・当 St 和 E 位于不同的存储区时 ON。 ・其它情况下 OFF。

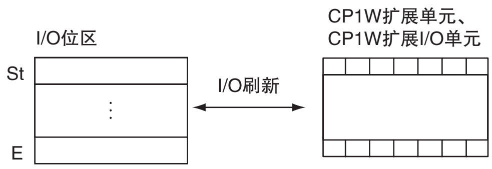
通过 IORF(097) 指令刷新单元

单元类型	CH	可通过 IORF(097) 进行刷新
30/40 点 I/O 型 CPU 单元	CP1E/CP2E CPU 单元内置 I/O：0CH、1CH、100CH 和 101CH	否
	CP1W 扩展 I/O 单元：2CH ~ 99CH、102CH ~ 199CH	是
	CP1W 扩展单元：2CH ~ 99CH、102CH ~ 199CH	是
60 点 I/O 型 CPU 单元	CP1E/CP2E CPU 单元内置 I/O：0CH、1CH、2CH、100CH、101CH 和 102CH	否
	CP1W 扩展 I/O 单元：3CH ~ 99CH、103CH ~ 199CH	是
	CP1W 扩展单元：3CH ~ 99CH、103CH ~ 199CH	是
NA 型 CPU 单元	CP1E CPU 单元内置 I/O：0CH 和 100CH	否
	CP1E CPU 单元内置模拟量：90CH、91CH 和 190CH	是
	CP1W 扩展 I/O 单元：1CH ~ 99CH、101CH ~ 199CH	是
	CP1W 扩展单元：1CH ~ 99CH、101CH ~ 199CH	是

注 CP1E/CP2E CPU 单元内置 I/O 区无法通过 IORF(097) 指令进行刷新。
CP1E/CP2E CPU 单元内置 I/O 区可通过即时刷新规定 (!) 进行刷新。

功能

IORF(097)指令刷新St和E之间(含)的I/O字。IORF(097)用于刷新分配给CP1W扩展(I/O)单元的字。对于30/40点I/O型而言，分配给扩展(I/O)单元的字介于CIO 002 ~ CIO 099、CIO 102 ~ CIO 199之间。对于60点I/O型而言，分配给扩展(I/O)单元的字介于CIO 003 ~ CIO 099、CIO 103 ~ CIO 199之间。对于NA20 I/O型而言，分配给扩展 (I/O) 单元的字介于 CIO 001 ~ CIO 099、CIO 101 ~ CIO 199 之间。



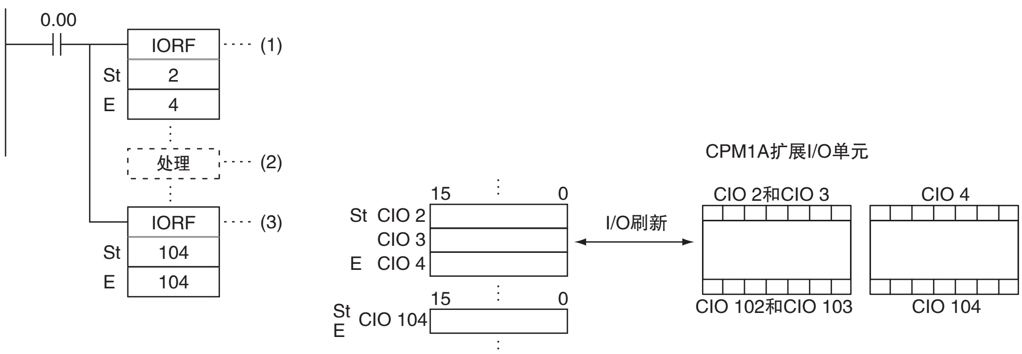
注意事项

- 可在中断任务中使用 IORF(097) 指令，从而通过中断对特定的 I/O 数据进行高速处理。如果在中断任务中使用 IORF(097) 指令，请务必对 PLC 设置中相应的高功能 I/O 单元循环刷新禁止位置 ON，从而禁止对指定的高功能 I/O 单元进行循环刷新。
- 如果 St ~ E 之间存在未安装单元的字，则不对这些字进行任何操作，而只对分配给单元的字进行刷新。
- 如果在通过 IORF(097) 指令启动的 I/O 刷新期间发生 I/O 总线错误，则该 I/O 刷新将被中途停止。

程序举例

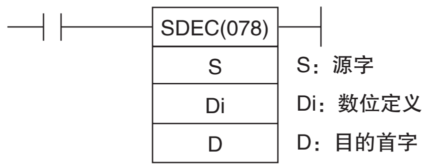
刷新 I/O 位区中的字

下例中，当 CIO 0.00 为 ON 时，首先刷新 CIO 2 ~ CIO 4(36 点输入)(1)，然后按要求进行相关处理 (2)，处理完成后接着刷新 CIO 104(8 点输出)(3)。



SDEC

指令	助记符	变化	功能代码	功能
7 段译码	SDEC	@SDEC	078	将指定数位的十六进制内容转换成 8 位、7 段显示代码，并将其放入指定目的字的高 8 位或低 8 位中。

	SDEC
符号	

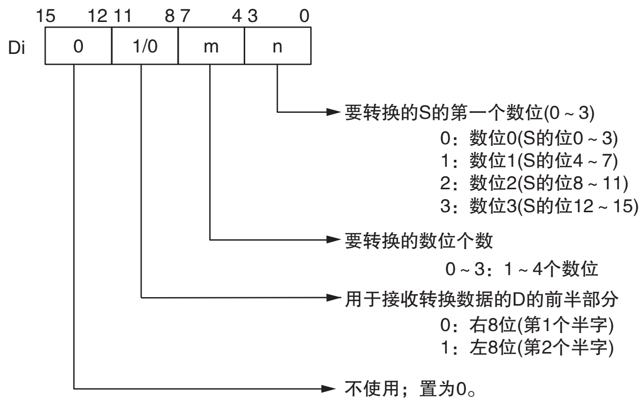
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源字	UINT	1
Di	数位定义	UINT	1
D	目的首字	UINT	可变

Di：数位定义



● 区域类型

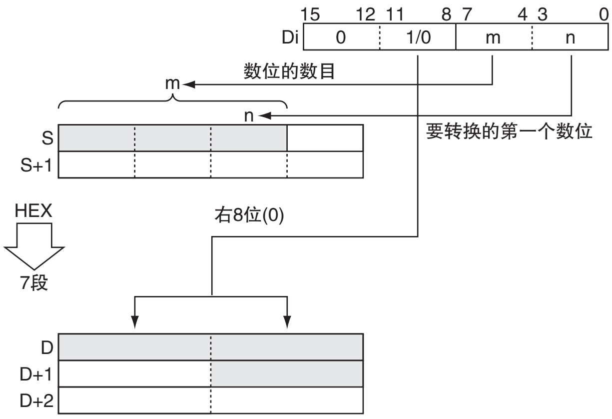
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	OK	---	---	---
Di	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・当 Di 中的设定不在指定范围内时 ON。 ・其它情况下 OFF。

功能

SDEC(078) 将 S 指定的数据当做 4 个数位的十六进制数据，将 S 中由 Di 指定的数位 (第一个数位和数位的个数) 转换成 7 段数据并将结果输出到由 Di 指定的 D 的位中。

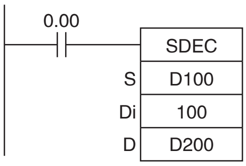


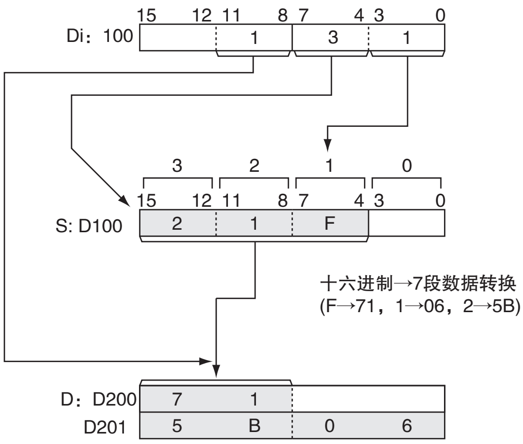
注意事项

- 如果在 Di 中指定了转换一个以上的数位，则数位将按从最低位到最高位的顺序转换。数位 0 是数位 3 之后的下一个数位。
- 结果按照从指定位置到高位字的顺序存储在 D 中。 如果目的字中仅有一个字节接收转换后的数据，则其它字节不变。

程序举例

下例中，当 CIO 0.00 变 ON 时，将 D100 中从数位 1 开始的 3 个数位的内容从十六进制数据转换成 7 段数据，并将结果输出到 D200 的高字节和 D201 中的两个字节。要转换的字节和输出字节位置的规定在 CIO 100 设定。

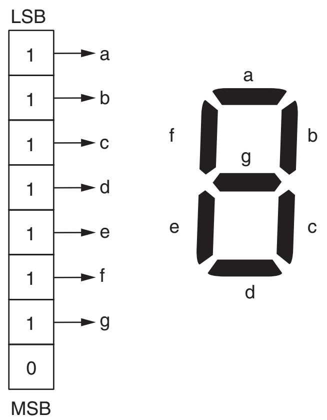




● 7 段数据

下表所示为从一个十六进制数位 (4 位) 转换成 7 段代码 (8 位) 的数据转换说明。

原始数据					转换码 (段)									显示原始数据
数位	位				–	g	f	e	d	c	b	a	Hex	
0	0	0	0	0	0	0	1	1	1	1	1	1	3F	
1	0	0	0	1	0	0	0	0	0	1	1	0	06	
2	0	0	1	0	0	1	0	1	1	0	1	1	5B	
3	0	0	1	1	0	1	0	0	1	1	1	1	4F	
4	0	1	0	0	0	1	1	0	0	1	1	0	66	
5	0	1	0	1	0	1	1	0	1	1	0	1	6D	
6	0	1	1	0	0	1	1	1	1	1	0	1	7D	
7	0	1	1	1	0	0	1	0	0	1	1	1	27	
8	1	0	0	0	0	1	1	1	1	1	1	1	7F	
9	1	0	0	1	0	1	1	0	1	1	1	1	6F	
A	1	0	1	0	0	1	1	1	0	1	1	1	77	
B	1	0	1	1	0	1	1	1	1	1	0	0	7C	
C	1	1	0	0	0	0	1	1	1	0	0	1	39	
D	1	1	0	1	0	1	0	1	1	1	1	0	5E	
E	1	1	1	0	0	1	1	1	1	0	0	1	79	
F	1	1	1	1	0	1	1	1	0	0	0	1	71	

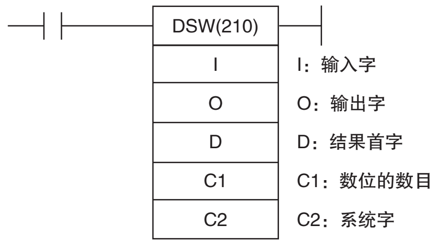


DSW

指令	助记符	变化	功能代码	功能
数字开关输入	DSW	---	210	读取设置在与 I/O 单元相连的外部数字开关 (或指轮开关) 上的值，并将 4 位数或 8 位数的值存储到指定的字中。

符号

DSW



适用程序区

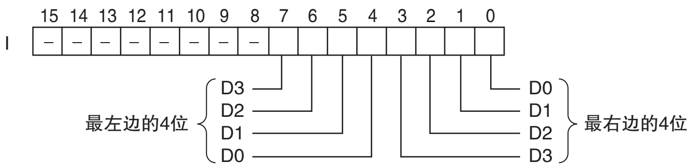
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型	大小
I	输入字	UINT	1
O	输出字	UINT	1
D	结果首字	WORD	可变
C1	数位的数目	UINT	1
C2	系统字	WORD	1

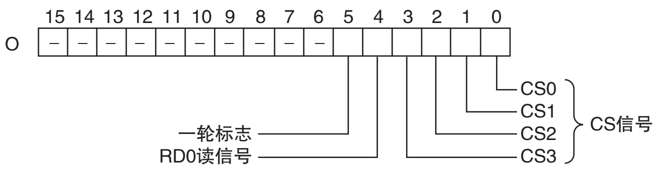
I：输入字 (数据线 D0 ～ D3 输入)

指定分配到输入单元的输入字，并将数字开关的 D0 ～ D3 数据线连接到输入单元，如下图所示。



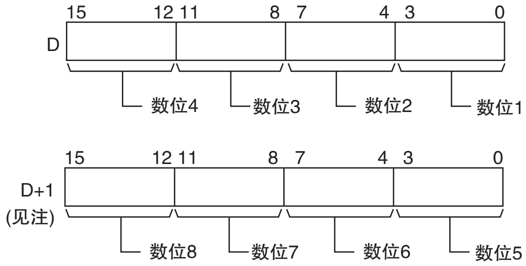
O：输出字 (CS/RD 控制信号输出)

指定分配到输出单元的输出字，并将数字开关的控制信号 (CS 和 RD 信号) 连接到输出单元，如下图所示。



D：结果首字

指定用于存储外部数字开关的设定值的起始字的地址。



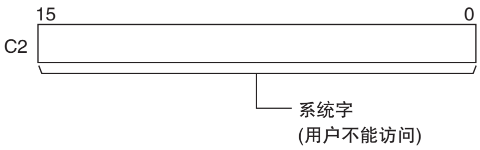
注：仅当C1=0001 Hex时读入8个数位。

C1：数位的数目

指定将从外部数字开关读入的数位的数目。将 C1 设定为 0000 Hex 可读入 4 个数位，设定为 0001 Hex 可读入 8 个数位。

C2：系统字

指定指令使用的工作字。该字不能用于其它场合。



● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>I，O</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td>OK</td><td rowspan="4">---</td><td rowspan="2">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>D</td><td rowspan="2">---</td></tr>
<tr><td>C1</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td rowspan="2">---</td><td>OK</td><td>---</td></tr>
<tr><td>C2</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	OFF

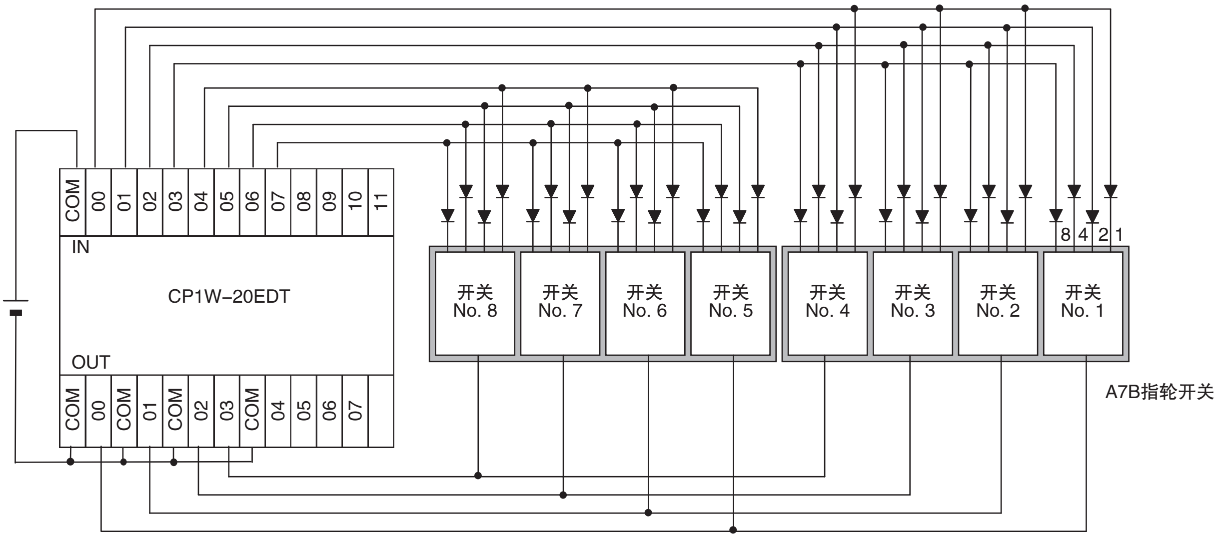
功能

DSW(210) 指令将控制信号输出到 O 的位 00 ～ 04，从 I 中读入数字开关数据线数据的指定数目的数位 (4 个数位或 8 个数位，在 C1 中指定)，并将结果存储到 D 和 D+1 中。(如果读入 4 个数位，结果将存储到 D 中；如果读入 8 个数位，结果将存储到 D 和 D+1 中。)

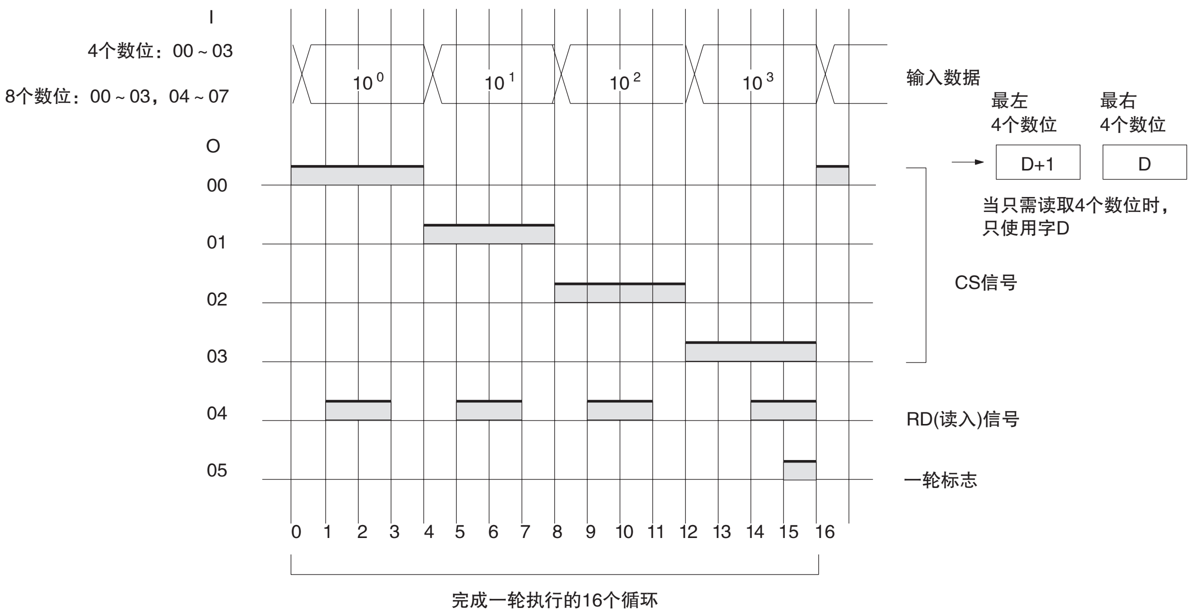
DSW(210) 指令每 16 个循环读入一次 4 位数或 8 位数的开关数据，执行相应的操作，然后继续读入数据。在 16 个 CPU 单元循环的每次循环中，一轮标志 (O 的位 05) 均置 ON 一次。

● 外部连接

将数字开关或指轮开关连接到输入单元触点 0 ~ 7 和输出单元触点 0 ~ 4，如下图所示。下图为 A7B 指轮开关的连接举例。



● 时序图



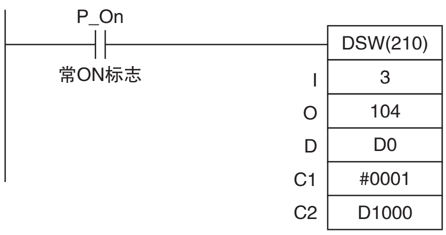
注意事项

- 请勿通过任何其它指令读 / 写系统字 (C2)。如果通过其它指令访问系统字，则 DSW(210) 指令将无法正确工作。当程序执行开始时，在第一个循环中 DSW(210) 指令不会对系统字进行初始化。如果从第一个循环开始使用 DSW(210) 指令，则请在程序中对系统字清零。
- 如果在执行 DSW(210) 指令后未在输入单元和输出单元连接至数字开关或指轮开关的情况下进行 I/O 刷新，则 DSW(210) 指令将无法正确工作。因此，请将输入单元的输入时间常数 (数据线输入字用) 设定为一个比循环时间更短的值。
- DSW(210) 指令在 16 个循环中读入一次 4 位数或 8 位数的数据并执行相应的操作，然后在下一轮 16 个循环中继续读入数据。
- 执行时，无论上一条指令停在什么位置，DSW(210) 指令均从 16 个循环中的第一个循环开始读入开关数据。

程序举例

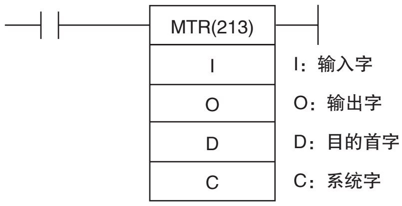
该例中，DSW(210) 指令用于从数字开关读入一个 8 位数并将结果值恒定输出到 D0 和 D3 中。数字开关通过 CIO 3 和 CIO 104 进行连接。

D1000 用作系统字。



MTR

指令	助记符	变化	功能代码	功能
矩阵输入	MTR	---	213	从与输入单元和输出单元 (使用 8 点输入和 8 点输出) 相连的 8 × 8 矩阵输入最多 64 个信号，并将该 64 位数据储存到 4 个目的字中。

符号	MTR
	

适用程序区

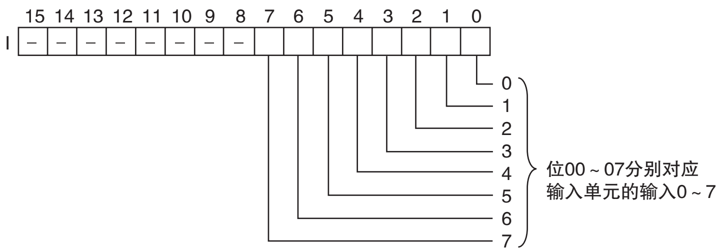
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

操作数

操作数	描述	数据类型	大小
I	输入字	UINT	1
O	输出字	UINT	1
D	目的首字	ULINT	4
C	系统字	WORD	1

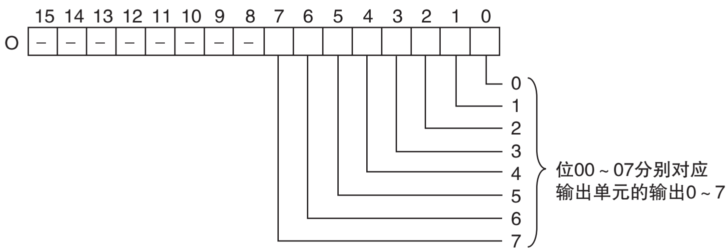
I：输入字

指定分配到输入单元的输入字，并将 8 根输入信号线连接到输入单元，如下图所示。



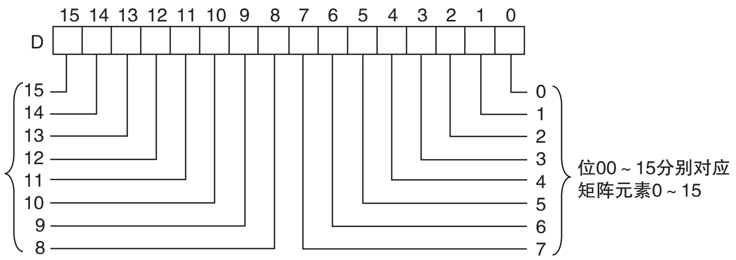
O：输出字 (选择信号输出)

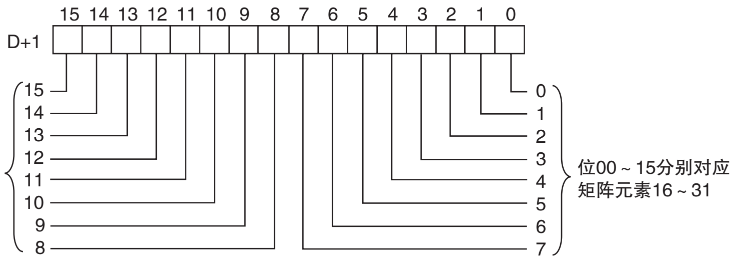
指定分配到输出单元的输出字，并将 8 个选择信号连接到输出单元，如下图所示。

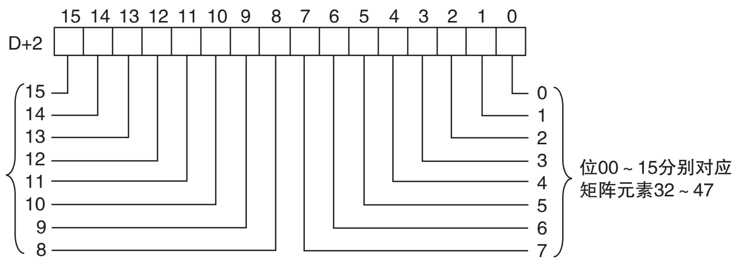


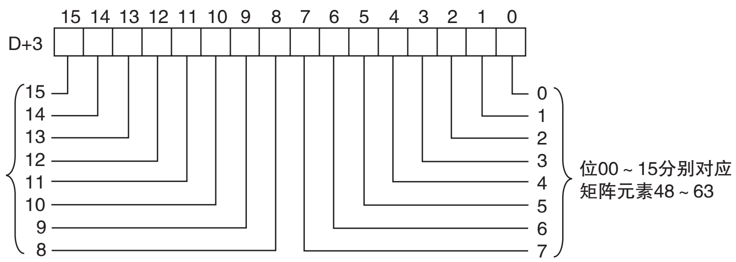
D：寄存器首字

指定包含 8 × 8 矩阵中数据的 4 个字的起始字地址。



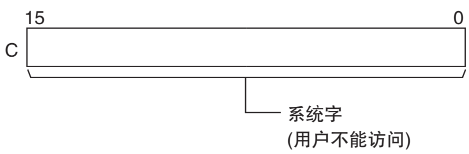






C：系统字

指定指令使用的工作字。该字不能用于其它场合。



- **区域类型**

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
I，O											OK					
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
C											OK					

条件标志

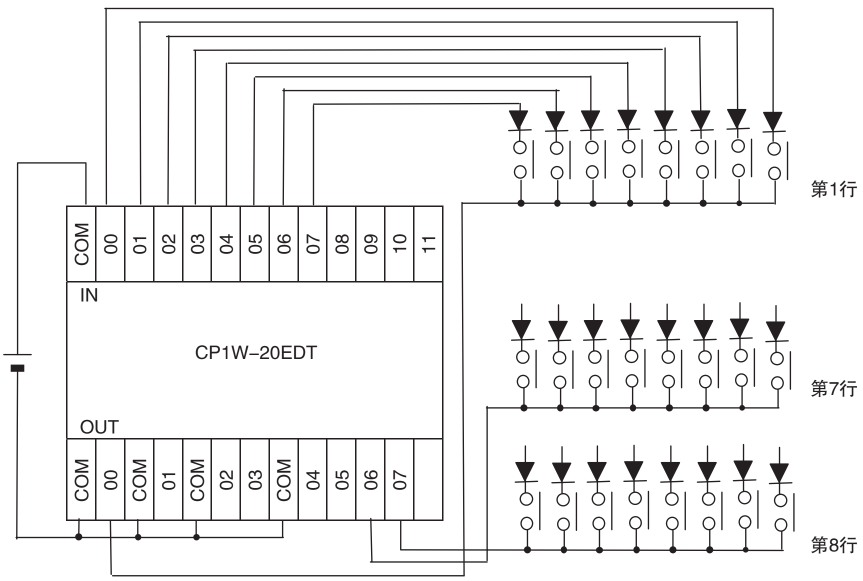
名称	标记	操作
出错标志	P_ER	OFF

功能

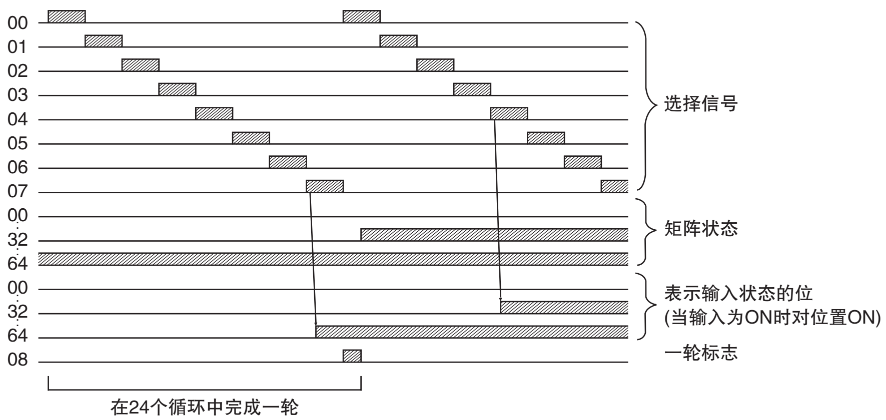
MTR(213) 指令将选择信号输出到 O 的位 00 ~ 07，按照 I 的位 00 ~ 07 顺序读入数据，并将 64 位的数据存储到 D ~ D+3 的 4 个字中。MTR(213) 指令每 24 个 CPU 单元循环读入一次 64 位矩阵的状态。在各选择信号置 ON 之后，一轮标志 (O 的位 08) 对每 24 个循环中的一个循环置 ON。

● 外部连接

将十六进制键盘连接到输入单元触点 0 ~ 7 和输出单元触点 0 ~ 7，如下图所示。



● 时序图

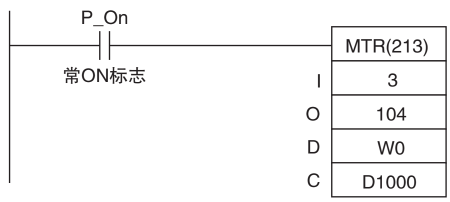


注意事项

- 请勿通过任何其它指令读 / 写系统字 (C)。如果通过其它指令访问系统字，则 MTR(213) 指令将无法正确工作。当程序执行开始时，在第一个循环中 MTR(213) 指令不会对系统字进行初始化。如果从第一个循环开始使用 MTR(213) 指令，则请在程序中对系统字清零。
- 如果在执行MTR(213)指令后未在输入单元和输出单元连接到外部矩阵后进行I/O刷新，则MTR(213)将无法正确工作。因此，请将输入单元的输入时间常数 (数据线输入字用) 设定为一个比循环时间更短的值。
- 执行时，无论上一条指令停在什么位置，MTR(213) 指令均从矩阵的起始位置开始读入矩阵状态。

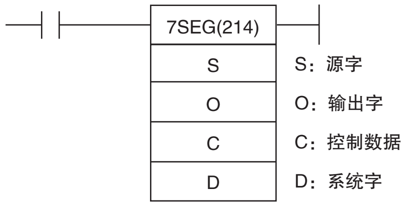
程序举例

该例中，MTR(213) 指令从 8 × 8 矩阵读入 64 位数据，并将该数据存储到 W0 ～ W3 中。8 × 8 矩阵通过 CIO 3 和 CIO 104 进行连接。D1000 用作系统字。



7SEG

指令	助记符	变化	功能代码	功能
7 段显示输出	7SEG	---	214	将源数据 (4 位数或 8 位数 BCD) 转换成 7 段显示数据，并将该数据输出至指定的输出字中。

	7SEG
符号	

适用程序区

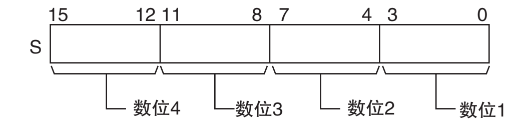
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	---

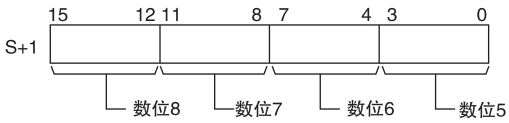
操作数

操作数	描述	数据类型	大小
S	源字	WORD	可变
O	输出字	UINT	1
C	控制字	#+10 仅十进制	1
D	系统字	WORD	1

S：源字

指定包含要转换成 7 段显示数据的源首字。

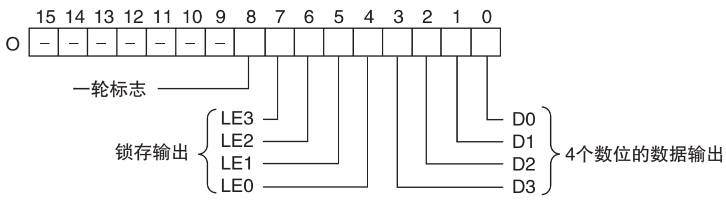




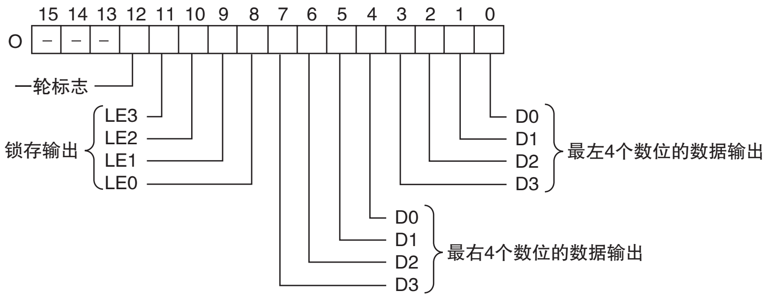
O：输出字 (数据和锁存输出)

指定分配到输出单元的输出字，并将 7 段显示器连接到输出单元，如下图所示。

- 转换 4 个数位



・转换 8 个数位



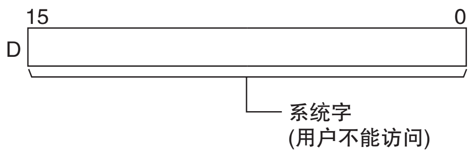
C：控制数据

C 的值表示源数据的数位个数和输入 / 输出单元的逻辑，如下表所示。(逻辑是指晶体管输出的 NPN 或 PNP 逻辑。)

<table>
<tr><th>源数据</th><th>显示的数据输入逻辑</th><th>显示的锁存输入逻辑</th><th>C</th></tr>
<tr><td rowspan="4">4 个数位 (S)</td><td rowspan="2">与输出单元相同</td><td>与输出单元相同</td><td>0000</td></tr>
<tr><td>与输出单元不同</td><td>0001</td></tr>
<tr><td rowspan="2">与输出单元不同</td><td>与输出单元相同</td><td>0002</td></tr>
<tr><td>与输出单元不同</td><td>0003</td></tr>
<tr><td rowspan="4">8 个数位 (S，S+1)</td><td rowspan="2">与输出单元相同</td><td>与输出单元相同</td><td>0004</td></tr>
<tr><td>与输出单元不同</td><td>0005</td></tr>
<tr><td rowspan="2">与输出单元不同</td><td>与输出单元相同</td><td>0006</td></tr>
<tr><td>与输出单元不同</td><td>0007</td></tr>
</table>

D：系统字

指定指令使用的工作字。该字不能用于其它场合。



● 区域类型

<table>
<tr><th rowspan="2">区域</th><th colspan="7">字地址</th><th colspan="2">DM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>S</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">OK</td><td rowspan="2">---</td><td>---</td><td rowspan="4">---</td><td rowspan="2">OK</td><td rowspan="4">---</td><td rowspan="4">---</td><td rowspan="4">---</td></tr>
<tr><td>O</td><td>OK</td></tr>
<tr><td>C</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>---</td><td>OK</td><td>---</td><td>---</td></tr>
<tr><td>D</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>OK</td><td>---</td><td>OK</td><td>OK</td></tr>
</table>

条件标志

名称	标记	操作
出错标志	P_ER	OFF

功能

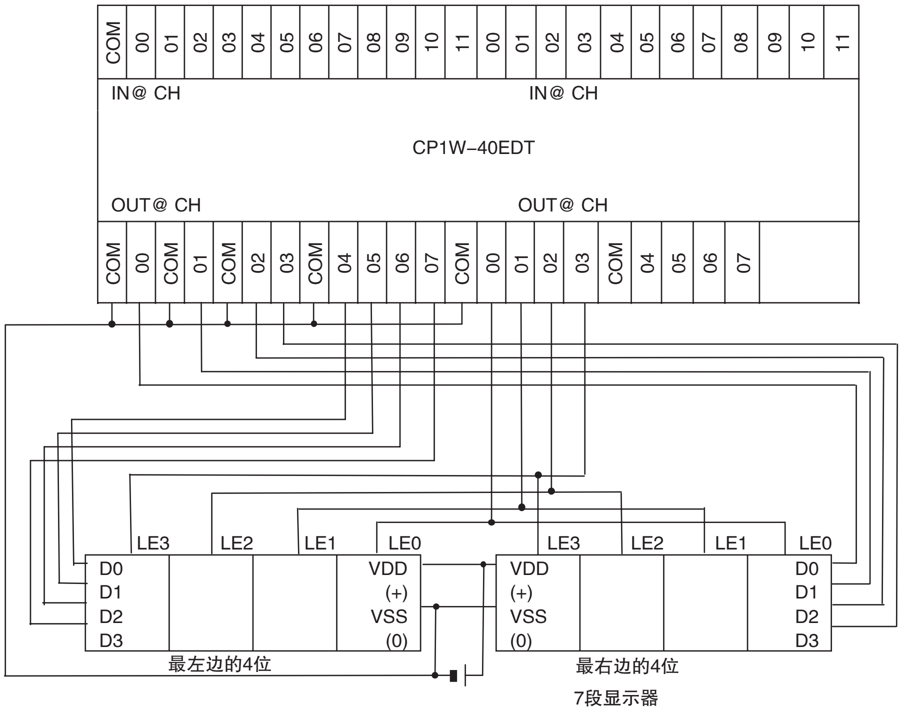
7SEG(214) 指令读入源数据，将其转换成 7 段显示数据，然后将数据 (最左 4 位 D0 ～ D3，最右 4 位 D0 ～ D3，锁存输出信号 LE0 ～ LE3) 输出到与通过 O 指定的输出相连的 7 段显示器上。C 的值表示源数据的数位个数 (4 个数位或 8 个数位) 以及输入和输出单元的逻辑。

7SEG(214) 指令显示 12 个循环中的 4 位数或 8 位数数据，执行相应的操作，然后显示读入数据。

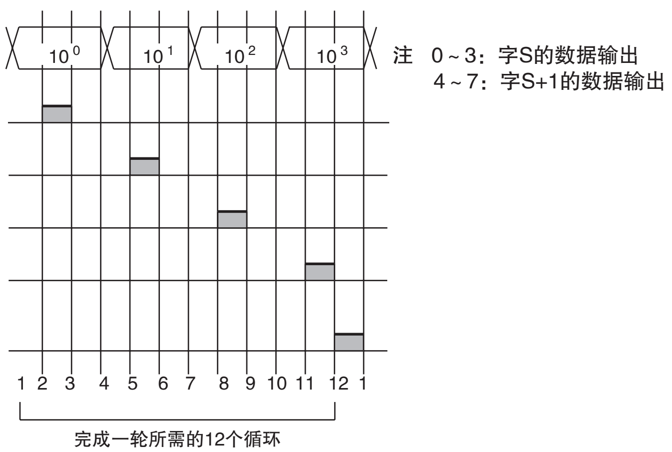
在 7SEG(214) 指令对各锁存输出信号置 ON 之后，一轮标志 (转换 4 位数据时为 O 的位 08，转换 8 位数据时为 O 的位 12) 对每 12 个循环中的一个循环置 ON。

● 外部连接

将 7 段显示器连接到输出单元，如下图所示。该例所示为 8 数位显示器。采用 4 数位显示器时，数据输出 (D0 ～ D3) 将连接到输出 0 ～ 3，且锁存输出 (LE0 ～ LE3) 将连接到输出 4 ～ 7。当已输出一轮数据，但除非应用有要求，否则不必连接输出点 12(对于 8 数位显示器) 或输出点 8(对于 4 数位显示器) 时，输出点 12 或输出点 8 将被置 ON。



● 时序图

功能	O 中的位 (4 个数位，1 个块)	O 中的位 (4 个数位，2 个块)	输出状态 (数据和锁存逻辑取决于 C)
数据输出	00 ～ 03	00 ～ 03 04 ～ 07	
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	
一轮标志	08	12	

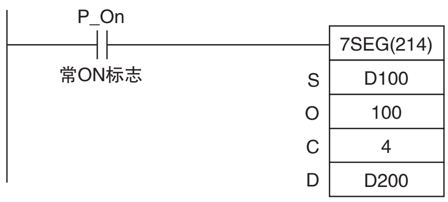
注意事项

- 请勿通过任何其它指令读 / 写系统字 (D)。如果通过其它指令访问系统字，则 7SEG(214) 指令将无法正确工作。当程序执行开始时，在第一个循环中 7SEG(214) 指令不会对系统字进行初始化。如果从第一个循环开始使用 7SEG(214) 指令，则请在程序中对系统字清零。
- 在输出 12 个循环中的 7 段数据之后，7SEG(214) 指令执行相应的操作，然后在下一轮 12 个循环中转换源字中的当前内容。
- 执行时，无论上一条指令停在什么位置，7SEG(214) 指令均从一轮的起始位置开始锁存输出 0。
- 即使相连的 7 段显示器中的显示数位少于 4 个或 8 个，7SEG(214) 指令也仍将输出数据的 4 个数位或 8 个数位。

程序举例

该例中，7SEG(214) 指令转换 D100 和 D101 中的 8 数位 BCD 数据，并将转换后的数据通过 CIO 100 进行输出。

将输出 8 个数位的数据，且 7 段显示器的逻辑与输出单元的逻辑相同，因此控制数据 (C) 被设定为 4。D200 用作系统字 D。



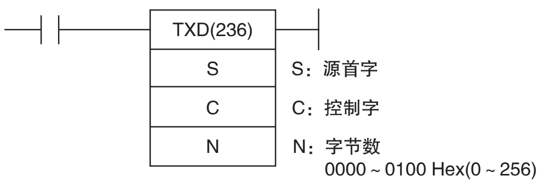
串行通信指令

TXD

指令	助记符	变化	功能代码	功能
发送	TXD	@TXD	236	从 CPU 单元的内置 RS-232C 端、内置 RS-485 端口或串行选件板端口输出指定字节数的数据。

符号

TXD



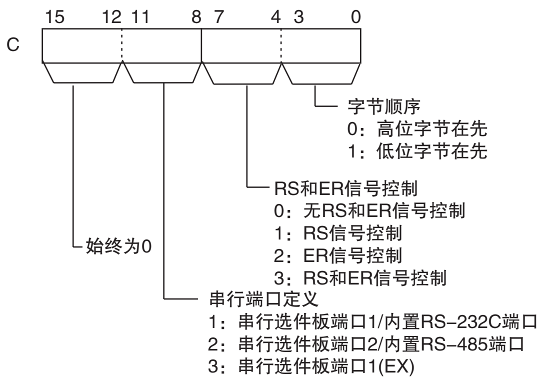
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源首字	UINT	可变
C	控制字	UINT	1
N	字节数 0000 ~ 0100 Hex(0 ~ 256)	UINT	1

C：控制字



注 1 串行选件板端口 1(EX) 仅可用于带 2 个端口的选件板 CP2W-CIFD1/D2/D3。

2 RS、ER 信号仅可用于支持的通信端口。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
C，N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・未在 PLC 设置中设定无协议模式时 ON。 ・当 C 的值不在范围内时 ON。 ・当 N 的值不在 0000 ～ 0100 Hex 之间时 ON。 ・当在发送就绪标志为 OFF 的情况下试图发送时 ON。(发送就绪标志为 A392.05、A392.13 或 A50.05。) ・其它情况下 OFF。

辅助区中的相关字和位

● 内置 RS-232C 端口 / 串行选件板端口 1

名称	地址	内容
内置 RS-232C 端口 / 串行选件板端口 1 发送就绪标志	A392.05	可以在无协议模式下发送数据时 ON。

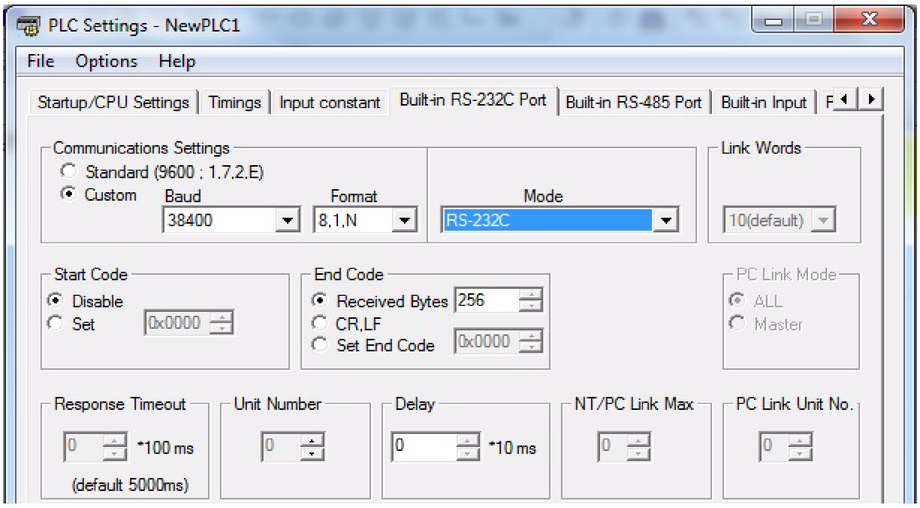
● 内置 RS-485 端口 / 串行选件板端口 2

名称	地址	内容
内置 RS-485 端口 / 串行选件板端口 2 发送就绪标志	A392.13	可以在无协议模式下发送数据时 ON。

● 串行选件板端口 1(EX)

名称	地址	内容
串行选件板端口 1(EX) 发送就绪标志	A50.05	可以在无协议模式下发送数据时 ON。

相关的 PLC 设置设定



功能

・TXD(236) 指令读取 S ～ S+(N ÷ 2)−1 中的 N 个字节的数据，并将这些未经处理的数据以无协议模式从 CPU 单元的内置 RS-232C 端口、内置 RS-485 端口或串行选件板端口输出。(输出端口通过 C 的位 8 ～ 11 来指定。)

・可在 PLC 设置中设定发送报文帧格式。

1) 起始码：无或 00 ～ FF Hex。

2) 结束码：无、CR+LF 或 00 ～ FF Hex。

数据将带在 PLC 设置中所指定的起始码和 / 或结束码一起发送。如果指定了起始码和结束码，则这些码将添加到发送数据 (N) 上。在这种情况下，可为 N 指定的最大字节数为 256 个字节。

- 数据按在 C0 ～ C3 中指定的顺序发送。
- 在 C4 ～ C7 中指定的 RS 和 ER 信号的控制规定的作用如下：

  1) 如果在 C 中指定了 RS 信号控制，则 S 的位 15 将被用作 RS 信号。

  2) 如果在 C 中指定了 ER 信号控制，则 S 的位 15 将被用作 ER 信号。

  3) 如果在C中指定了RS和ER信号控制，则S的位15将被用作RS信号，S的位14将被用作ER信号。

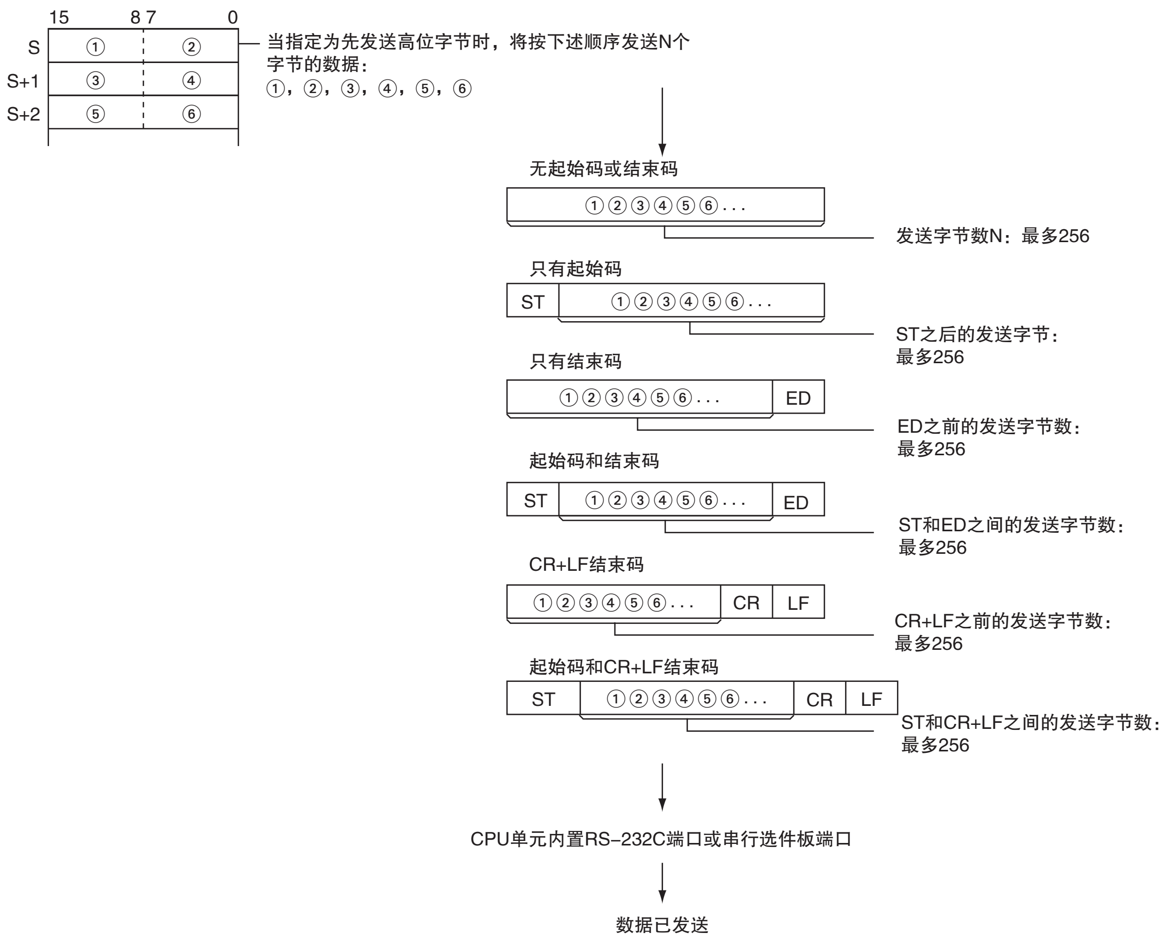
  如果在 C 中将 RS 和 ER 信号控制指定为 1、2 或 3 Hex，则无论发送就绪标志 (A392.05、A392.13 或 A50.05，取决于使用的端口) 的状态如何，均执行 TXD(236) 指令。

注 1 CP1E N □□ S(1) 型 CPU 单元和 CP2E E/S □□型 CPU 单元的内置 RS-232C 端口不支持 ER 信号，因此不能进行 ER 信号控制。请勿将 C 的位 4~7 设定为 2 或 3 Hex。

2 CP1E N □□ S1 型 CPU 单元的内置 RS-485 端口和串行选件板 CP1W-CIF11/12-V1、CP2W-CIFD1/D2/D3 不支持 RS、ER 信号，因此不能进行 RS、ER 信号控制。请勿将 C 的位 4~7 设定为 1、2 或 3 Hex。

- 最多可发送 259 个字节，其中包括发送数据 (N= 最多 256 个字节)、起始码和结束码。
- 用 N 指定发送数据的大小，其中不包括起始码和结束码。

● 起始码 / 结束码设定和发送数据



提示

- 当通过 TXD 指令将数据发送至另一设备时，该设备可能会要求数据以特定间隔发送。在这种情况下，可设定一个发送延时，以调整发送间隔。

注意事项

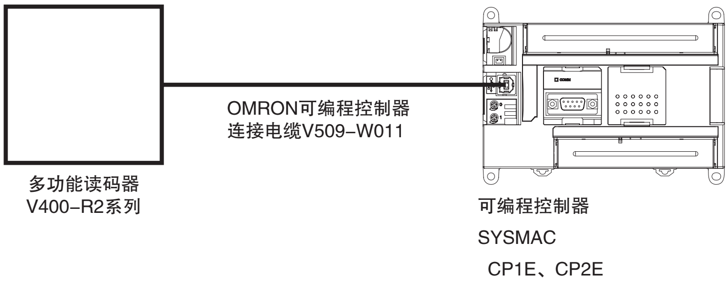
- TXD(236) 指令仅可用于 CPU 单元的内置 RS-232C 端口、内置 RS-485 端口或串行选件板端口。另外，端口必须设定为无协议模式。
- 仅当端口的发送就绪标志为 ON 时才可发送数据。(发送就绪标志为 A392.05、A392.13 或 A50.05。)
- 如果为 N 指定 0，则将不发送任何数据。

程序举例

● 将数据发送到读码器

下面以将数据发送到 V400-R2 系列多功能读码器的过程为例，说明如何与外部设备进行通信。

硬件配置



该例中，外部设备连接至内置到 CPU 单元中的 RS-232C 端口上。

首先为读码器设定读入条件。

通信设定

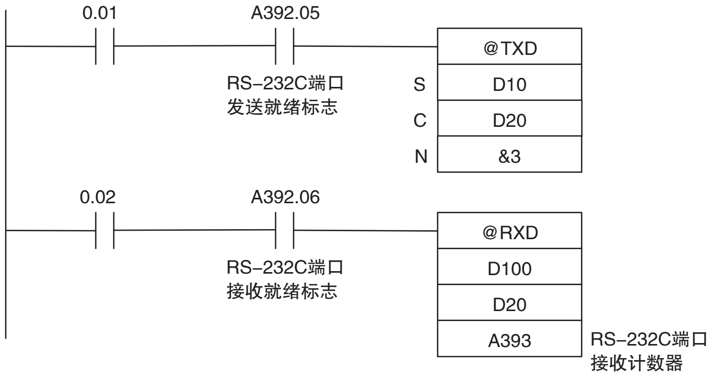
读码器的通信设定如下表所示。表中为默认设定。

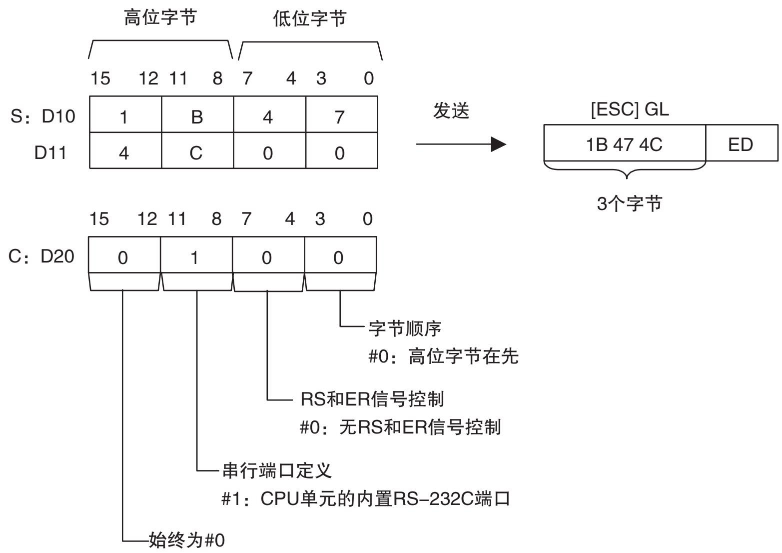
项目	设定
通信模式	无协议
波特率	9,600bps
数据位长	8 个位
校验	无校验
停止位	1
起始码	无校验
结束码	CR(0x000D)

将 PLC 通信设定设为与 PLC 设置中相同的值。只需设定结束码。

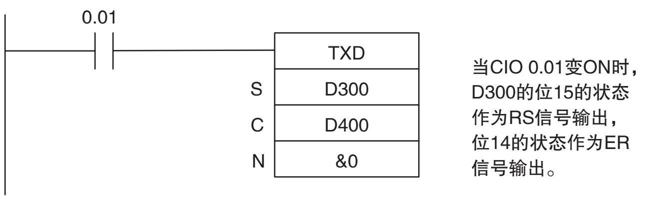
编程举例

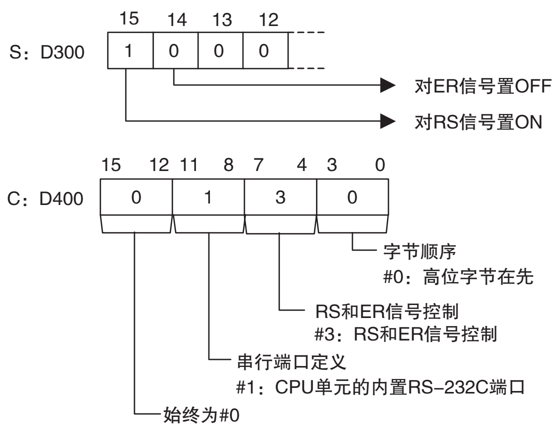
在 RS-232C 端口发送就绪标志 (A392.05) 为 ON 的情况下，如果 CIO 0.01 变 ON，则将从 D10 的高位字节开始的 3 个字节的数据不经转换直接发送到与 CPU 单元的内置 RS-232C 端口相连的读码器。这 3 个字节中包含 “[ESC] GL [CR]”，即用作从 RS-232C 到读码器的触发器输入的读入触发器指令。





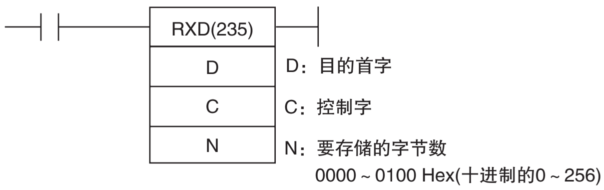
● 控制信号





RXD

指令	助记符	变化	功能代码	功能
接收	RXD	@RXD	235	从 CPU 单元的内置 RS-232C 端口、内置 RS-485 端口或串行选件板端口读取指定字节数的数据。

符号	RXD
	

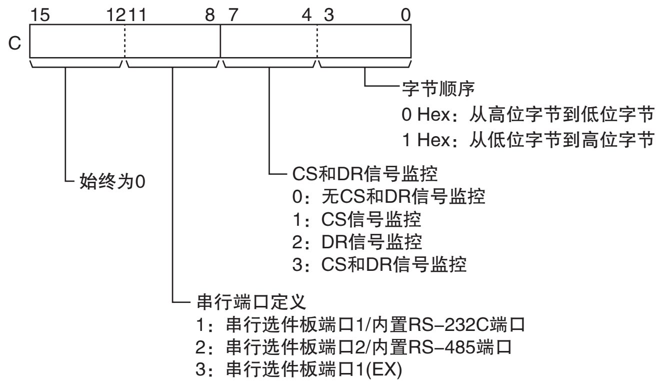
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
D	目的首字	UINT	可变
C	控制字	UINT	1
N	要存储的字节数 0000 ~ 0100 Hex(十进制的 0 ~ 256)	UINT	1

C：控制字



注 1 串行选件板端口 1(EX) 仅可用于带 2 个端口的选件板 CP2W-CIFD1/D2/D3。
2 CS、DR 信号仅可用于支持的通信端口。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
C，N	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	OK	---	---	---

条件标志

名称	标记	操作
出错标志	P_ER	・未在 PLC 设置中设定无协议模式时 ON。 ・当 C 的值不在范围内时 ON。 ・当 N 的值不在 0000 ~ 0100 Hex 之间时 ON。 ・其它情况下 OFF。

辅助区中的相关字和位

● 内置 RS-232C 端口 / 串行选件板 1 的辅助区标志

名称	地址	内容
内置 RS-232C 端口 / 串行选件板 1 接收完成标志	A392.06	当无协议接收完成时 ON。 指定了接收字节数：当已接收到指定的字节数时，该标志将置 ON。 指定了结束码：当接收到结束码或者当已接收到 256 个字节时，该标志将置 ON。
内置 RS-232C 端口 / 串行选件板 1 接收溢出标志	A392.07	在无协议模式下接收到的字节数超过预期的字节数时 ON。 指定了接收字节数：若在接收完成之后但尚未通过 RXD(235) 指令从缓冲区中读入已接收到的数据之前又接收到数据时，该标志将置 ON。 指定了结束码：当接收到 257 个或以上的字节的数据而无结束码时，该标志将置 ON。
内置 RS-232C 端口 / 串行选件板 1 接收计数器	A393.00 ~ A393.15	以十六进制统计以无协议模式接收到的字节数 (十进制的 0 ~ 256)。

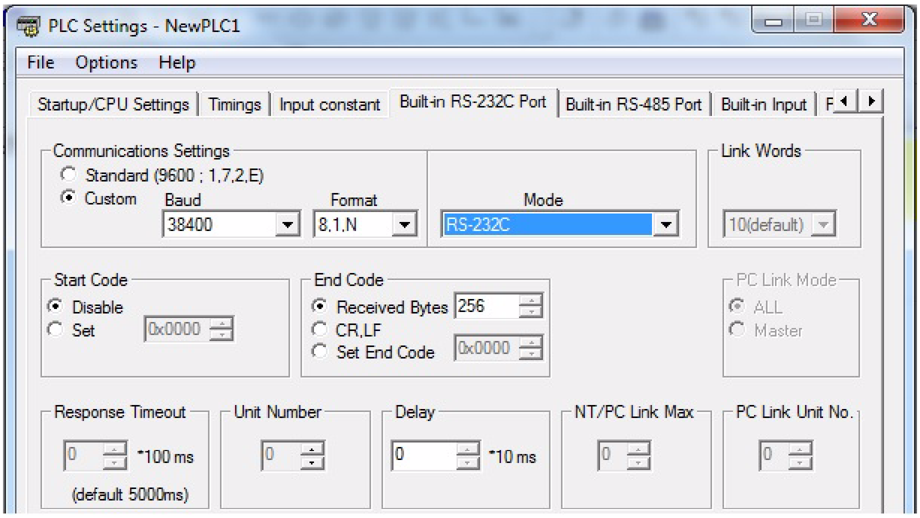
● 内置 RS-485 端口 / 串行选件板端口 2 的辅助区标志

名称	地址	内容
内置 RS-485 端口 / 串行选件板端口 2 接收完成标志	A392.14	当无协议接收完成时 ON。 指定了接收字节数：当已接收到指定的字节数时，该标志将置 ON。 指定了结束码：当接收到结束码或者当已接收到 256 个字节时，该标志将置 ON。
内置 RS-485 端口 / 串行选件板端口 2 接收溢出标志	A392.15	在无协议模式下接收到的字节数超过预期的字节数时 ON。 指定了接收字节数：若在接收完成之后但尚未通过 RXD(235) 指令从缓冲区中读入已接收到的数据之前又接收到数据时，该标志将置 ON。 指定了结束码：当接收到 257 个或以上的字节的数据而无结束码时，该标志将置 ON。
内置 RS-485 端口 / 串行选件板端口 2 接收计数器	A394.00 ~ A394.15	以十六进制统计以无协议模式接收到的字节数 (十进制的 0 ~ 256)。

● 串行选件板端口 1(EX) 的辅助区标志

名称	地址	内容
串行选件板端口 1(EX) 接收完成标志	A50.06	当无协议接收完成时 ON。 指定了接收字节数：当已接收到指定的字节数时，该标志将置 ON。 指定了结束码：当接收到结束码或者当已接收到 256 个字节时，该标志将置 ON。
串行选件板端口 1(EX) 接收溢出标志	A50.07	在无协议模式下接收到的字节数超过预期的字节数时 ON。 指定了接收字节数：若在接收完成之后但尚未通过 RXD(235) 指令从缓冲区中读入已接收到的数据之前又接收到数据时，该标志将置 ON。 指定了结束码：当接收到 257 个或以上的字节的数据而无结束码时，该标志将置 ON。
串行选件板端口 1(EX) 接收计数器	A51.00 ~ A51.15	以十六进制统计以无协议模式接收到的字节数 (十进制的 0 ~ 256)。

相关的 PLC 设置设定



功能

- RXD(235) 指令读取在无协议模式下从 CPU 单元的内置 RS-232C 端口、内置 RS-485 端口或串行选件板端口 (端口通过 C 的位 8 ～ 11 指定) 接收到的数据，并将 N 个字节的数据存储到字 D ～ D + (N ÷ 2)-1 中。如果从端口接收到的数据未达到 N 个字节，则只存储已接收到的数据。
- 可在 PLC 设置中设定下列接收报文帧格式。

  1) 起始码：无或 00 ～ FF Hex。

  2) 结束码：无、CR+LF 或 00 ～ FF Hex。如果未指定结束码，则要接收的字节数设定为 00 ～ FF Hex(十进制的 1 ～ 256 ； 00 指定 256 个字节)。
- 数据将按在 C0 ～ C3 中指定的顺序存储到存储器中。
- 接收完成标志置 ON 的情况

  当已接收到在 PLC 设置中指定的字节数的数据时，接收完成标志 (注 (a)) 将置 ON。当接收完成标志变 ON 后，接收计数器 (注 (b)) 中的字节数将与在 PLC 设置区中指定的接收字节数相同。
  如果在 PLC 设置区中指定了结束码，则当接收到结束码时或者已接收到 256 个字节的数据时，接收完成标志 (注 (a)) 将置 ON。如果接收到的字节数超过指定的字节数，则接收溢出标志 (注 (c)) 将置 ON。
- 执行 RXD(235) 指令时，数据存储到从 D 开始的存储器中，接收完成标志 (注 (a)) 将置 OFF(即使接收溢出标志 (注 (c)) 为 ON 也如此)，且接收计数器 (注 (b)) 将被清零。
- 如果 RS-232C 端口重启位 (注 (d)) 被置 ON，则接收完成标志 (注 (a)) 将被置 OFF(即使接收溢出标志为 ON 也如此)，且接收计数器 (注 (b)) 将被清零。
- 在 C4 ～ C7 中指定的 CS 和 DR 信号的监控规定的作用如下：

  1) 如果在 C 中指定了 CS 信号监控，则将 CS 信号的状态存储到 D 的位 15 中。

  2) 如果在 C 中指定了 DR 信号监控，则将 DR 信号的状态存储到 D 的位 15 中。

  3) 如果在 C 中指定了 CS 和 DR 信号监控，则将 CS 信号的状态存储到 D 的位 15 中，将 DR 信号的状态存储到 D 的位 14 中。

  如果在 C 中为 CS 和 DR 信号控制指定 1、2 或 3 Hex，则无论接收完成标志 (注 (a)) 的状态如何，均执行 RXD(235) 指令。

注 1 CP1E N □□ S(1) 型 CPU 单元和 CP2E E/S □□型 CPU 单元的内置 RS-232C 端口不支持 CS 信号，因此不能进行 CS 信号监控。请勿将 C 的位 4~7 设定为 2 或 3 Hex。

2 CP1E N □□ S1 型 CPU 单元的内置 RS-485 端口和串行选件板 CP1W-CIF11/12-V1、CP2W-CIFD1/D2/D3 不支持 CS、DR 信号，因此不能进行 CS、DR 信号监控。请勿将 C 的位 4~7 设定为 1、2 或 3 Hex。

- 如果指定 CS 或 DR 信号监控，则将不存储接收到的数据。
- 最多可接收 259 个字节，其中包括接收数据 (N= 最多 256 个字节)、起始码和结束码。
- 用 N 指定接收数据的大小，其中不包括起始码和结束码。

注 相关辅助区和 CIO 区地址

(a) **接收完成标志**

内置 RS−232C 端口 / 串行选件板端口 1	A392.06
内置 RS−485 端口 / 串行选件板端口 2	A392.14
串行选件板端口 1(EX)	A50.06

(b) **接收计数器**

内置 RS−232C 端口 / 串行选件板端口 1	A393
内置 RS−485 端口 / 串行选件板端口 2	A394
串行选件板端口 1(EX)	A51

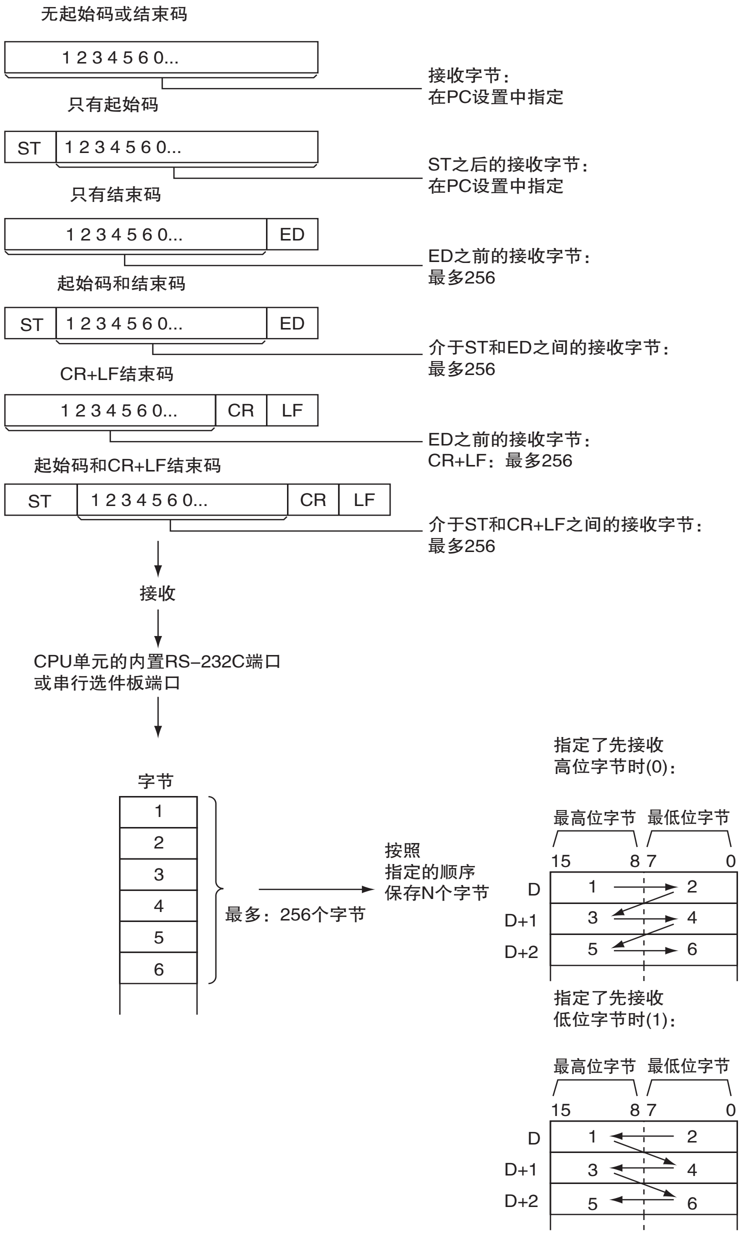
(c) **接收溢出标志**

内置 RS−232C 端口 / 串行选件板端口 1	A392.07
内置 RS−485 端口 / 串行选件板端口 2	A392.15
串行选件板端口 1(EX)	A50.07

(d) **RS−232C 端口重启位**

内置 RS−232C 端口 / 串行选件板端口 1	A526.00
内置 RS−485 端口 / 串行选件板端口 2	A526.01
串行选件板端口 1(EX)	A526.02

● 起始码 / 结束码设定和接收数据



提示

· 当使用 RXD(235) 指令来读入从某个串行选件板端口接收到的数据时，该端口的接收缓冲区将在执行 RXD(235) 指令之后被清除。因而无法重复执行 RXD(235) 指令来分批读入一个区块的数据。

注意事项

· RXD(235) 指令仅可用于 CPU 单元的内置 RS-232C 端口、内置 RS-485 端口或串行选件板端口。另外，端口必须设定为无协议模式。

· 请在接收完成标志(A392.06、A392.14或A50.06)为1(ON)时执行该指令以(从接收缓冲区)接收数据。

· 接收到数据时，必须通过 RXD 指令来读入该数据，否则无法接收下一条数据。当接收完成标志变 ON 时，在下次接收之前，应通过 RXD 指令读入接收到的数据。

· 用 N 指定接收数据的大小，其中不包括起始码和结束码。

· 如果为 N 指定 0，则将接收完成标志和接收溢出标志 (注 (a)) 置 OFF，接收计数器 (注 (b)) 将被清零，且不会将任何数据存储到存储器中。

程序举例

● 接收数据

下面以从 V400-R2 系列多功能读码器接收数据的过程为例，说明如何与外部设备进行通信。

硬件配置

该例中，外部设备连接至内置到 CPU 单元中的 RS-232C 端口上。

首先为读码器设定读入条件。

通信设定

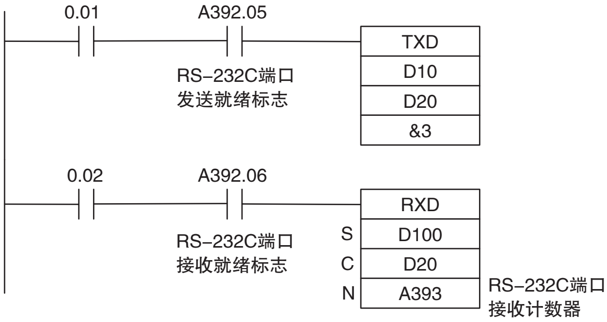
读码器的通信设定如下表所示。表中为默认设定。

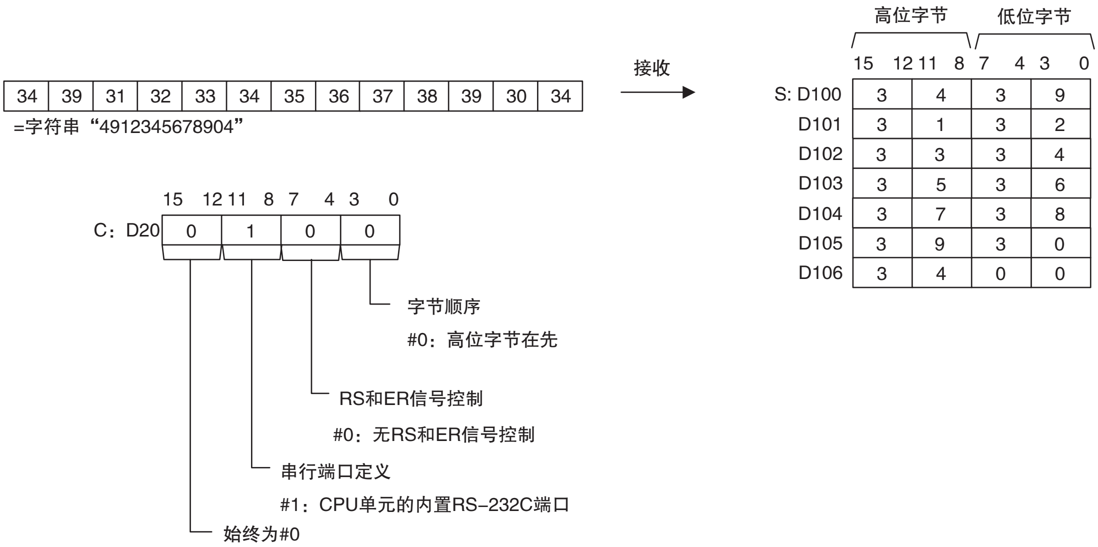
项目	设定
通信模式	无协议
波特率	9,600bps
数据位长	8 个位
校验	无校验
停止位	1
起始码	无校验
结束码	CR(0x000D)

将 PLC 通信设定设为与 PLC 设置中相同的值。只需设定结束码。

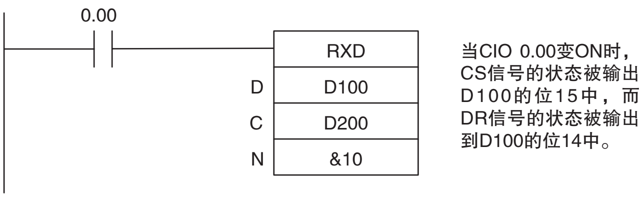
编程举例

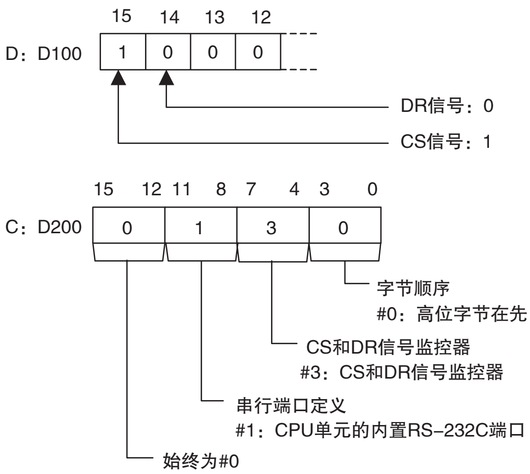
在 RS-232C 端口接收完成标志 (A392.06) 为 ON 的情况下，如果 CIO 0.02 变 ON，则将从与 CPU 单元的内置 RS-232C 端口相连的读码器读入在 RS-232C 端口接收计数器 (A393) 中指定的读取结果的字节数的数据，并存储在从 D100 的高位字节开始的位置。





控制信号





网络通信指令

关于网络通信指令的概要

必要时 (条件成立时) 对网络上或 CPU 装置上的各种装置 (CPU 单元、计算机等) 进行数据收发及 (模式变更等) 控制用的指令。

CP2E 可执行使用 Ethernet 端口的数据收发。

*CP1E 无法使用网络通信指令。

网络通信指令有以下 2 种。

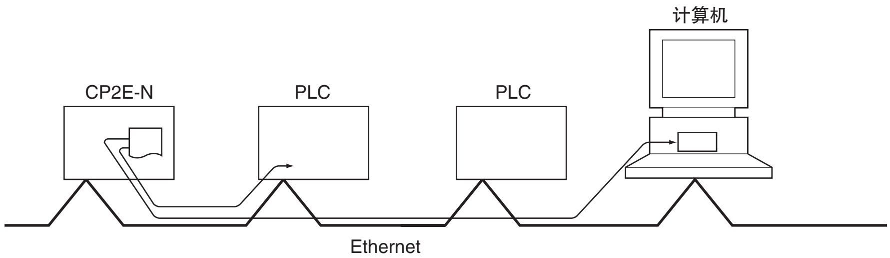
信息内容	网络通信指令	内 容
发送数据发送／接收用指令 (FINS指令)	SEND/RECV指令	CPU单元 SEND或RECV指令 → 数据发送 → 各种装置 CPU单元或计算机；← 数据接收
发送任意指令 (FINS指令)	CMND指令	CPU单元 CMND指令 → 发送指令 → 各种装置 CPU单元或计算机；← 响应

通过上述网络通信指令发送的指令是称为 “FINS 指令”的 FA 控制装置用通信指令。“FINS 指令”的详请请参阅 “SYSMAC CS/CJ 系列 通信指令参考手册”(手册编号：W342)。

网络通信的模式

● 对于网络上的装置

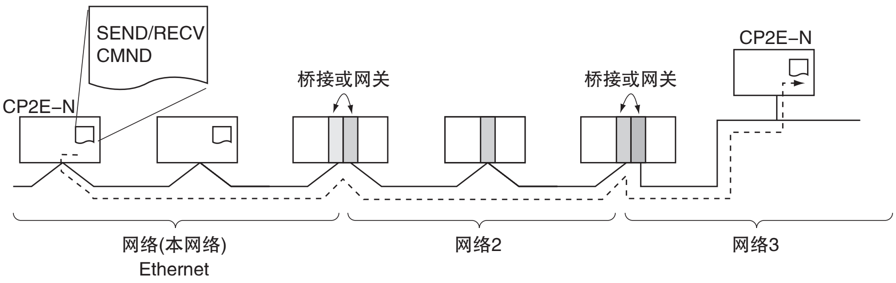
对本网络上的 PLC、计算机发送指令。



详情请参阅各网络 (Ethernet) 用单元的用户手册。

● **跨网络**

包括本网络 (发送数据的网络) 在内，最多可跨越 3 层网络。



注 CP2E CPU 单元不支持桥接或网关功能。

执行跨网络通信时，需将表示到达目标网络上节点用线路的表格登录至各 PLC(CPU 单元)。该表格称为 “路由表”。路由表由以下 2 种表格构成。

・本网络表　：　本 PLC 安装的单元编号和网络地址的对应表
・中继网络表　：　未连接本 PLC 的目标网络 (最终网络)，以及到达该位置的线路首个中继点 (首先应去的位置) 的网络地址和节点地址的对应表

网络通信用指令

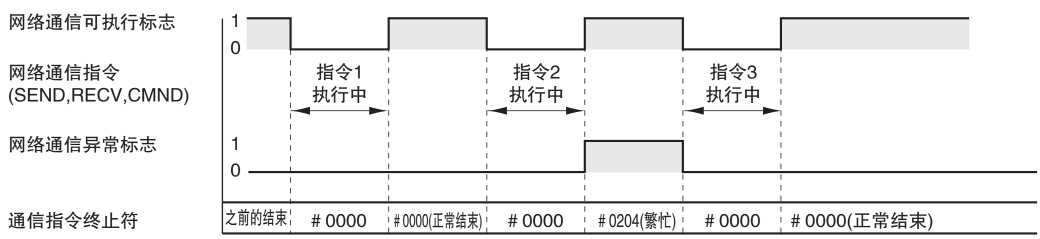
CP2E N □□型最多可同时执行基于 8 个指令的网络通信。

	助记符	指令名称	特征
网络通信用指令	SEND	网络发送	最多可同时使用 8 个。 通信的完成、错误的检出使用 AR(特殊辅助继电器)。 需指定内部逻辑端口或使用内部逻辑端口自动分配功能。
	RECV	网络接收	
	CMND	指令发送	

通信标志的动作

・ 网络通信可执行标志在收发执行中变为 0(OFF)，在执行结束 (无论正常还是异常) 后变为 1(ON)。
・ 网络通信异常标志在下一次收发执行前将保持状态。
・ 网络通信异常标志即使异常结束，在下一次通信指令执行时仍会变为 0(OFF)。

(例)

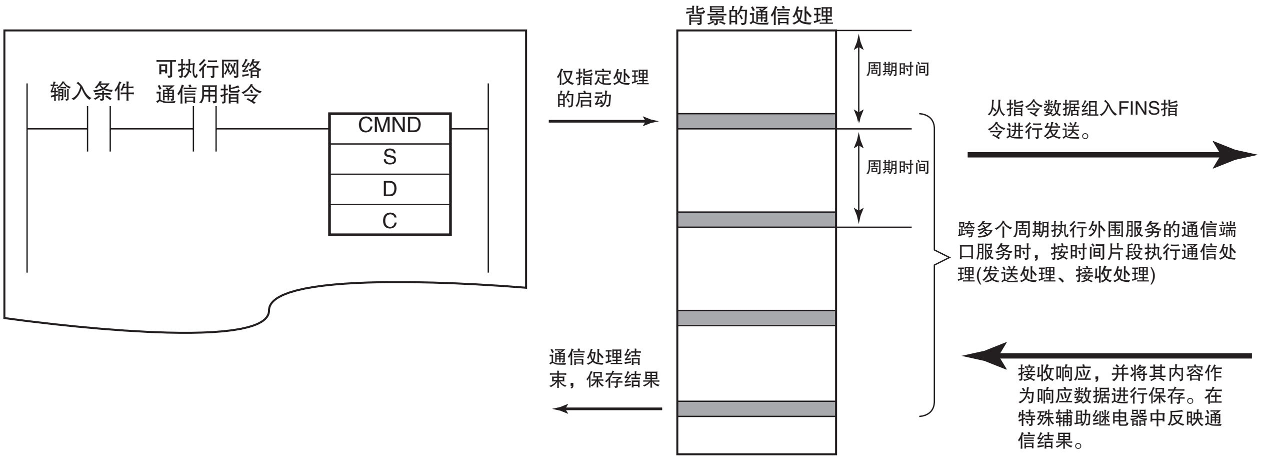


地址	位	名称	内容
A202	00 ～ 07	网络通信指令可执行标志	通信指令可执行时变为 1(ON)。各位 00 ～ 07 对应内部逻辑端口 No.0 ～ 7。通信指令执行后根据本标志 OFF → ON，可获知网络通信完成。通信指令执行时变为 0(OFF)。
	15	网络通信可自动分配标志	可通过内部逻辑端口自动分配执行通信指令时，即可用于自动分配的内部逻辑端口有空闲时变为 1(ON)。 参考：同时使用 9 个以上的通信指令时，根据该标志确认自动分配可使用的内部逻辑端口是否有空闲后执行通信指令。

地址	位	名称	内容
A203 ~ A210CH	–	网络通信响应代码	设置执行了网络通信时的响应代码。
A214CH	00 ~ 07	网络通信完成时 1 个周期 ON 标志	通信完成时仅 1 个周期变为 1(ON)。各位 00 ~ 07 对应内部逻辑端口 No.0 ~ 7。 注：本标志在执行了通信指令的下一个周期起生效，因此需在梯形图程序中待机 1 个周期后再查看，避免在执行了通信指令的周期中查看。参考：根据保存在 A218CH(使用内部逻辑端口 No.) 中的使用内部逻辑端口 No.，指定位 00 ~ 07 中的任意一个进行查看。
	08 ~ 15	禁用	–
A215CH	00 ~ 07	网络通信完成时 1 个周期 ON 错误标志	若在通信完成时发生错误，仅 1 个周期变为 1(ON)。各位 00 ~ 07 对应内部逻辑端口 No.0 ~ 7。本标志变为 1(ON) 时，请查看网络通信响应代码 (A203 ~ A210CH)，确定错误原因。 注：本标志在执行了通信指令的下一个周期起生效，因此需在梯形图程序中待机 1 个周期后再查看，避免在执行了通信指令的周期中查看。 参考：根据保存在 A218CH(使用内部逻辑端口 No.) 中的使用内部逻辑端口 No.，指定位 00 ~ 07 中的任意一个进行查看。
	08 ~ 15	禁用	–
A216 ~ A217CH	–	网络通信响应代码保存地址	执行通信指令后，通信响应代码将自动保存至特殊辅助继电器的任意通道地址。这里保存该任意通道地址的 I/O 存储器有效地址。 参考：将这里保存的 I/O 存储器有效地址传送至索引寄存器 IR，根据索引寄存器间接指定，可读取任意通道地址中保存的通信响应代码。
A218CH	–	使用内部逻辑端口 No.	执行通信指令后，这里保存使用了的内部逻辑端口 No.。 0 ~ 7：内部逻辑端口 No.0 ~ 7
A219CH	00 ~ 07	网络通信执行错误标志	网络通信执行中发生错误时变为 1(ON)。本标志变为 1(ON) 时，请查看网络通信响应代码 (A203 ~ A210CH)，确定错误原因。各位 00 ~ 07 对应内部逻辑端口 No.0 ~ 7。通信指令执行时变为 0(OFF)。

网络通信用指令的执行时间

网络通信用指令在输入条件成立时，仅启动通信处理，实际的通信处理在背景中通过外围服务处理执行。

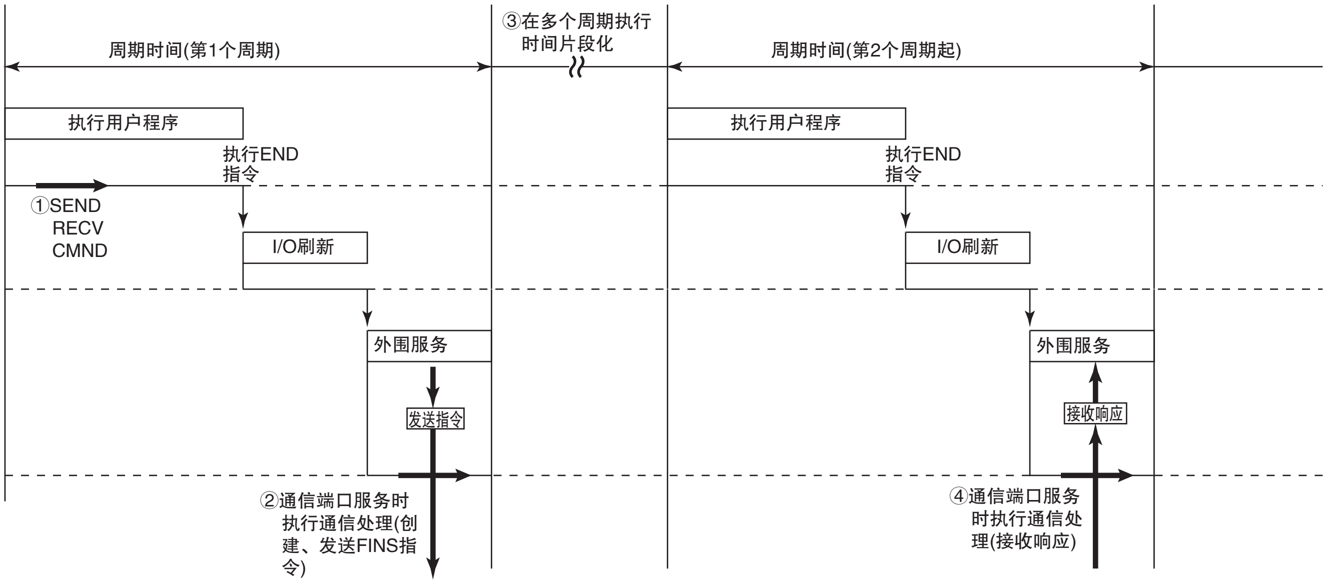


进行以下动作。

①输入条件成立时，若此时的网络通信指令可执行标志 (A202.00 ~ A202.07) 为 1(ON)，则各指令将网络指令可执行标志 (A202.00 ~ A202.07) 设为 0(OFF)，将网络通信执行错误标志 (A219.00 ~ A219.07) 设为 0(OFF)，将网络通信响应代码 (A203 ~ A210CH) 设为 #0000，读取 C，并启动通信处理 (FINS 指令发送／响应接收处理)。

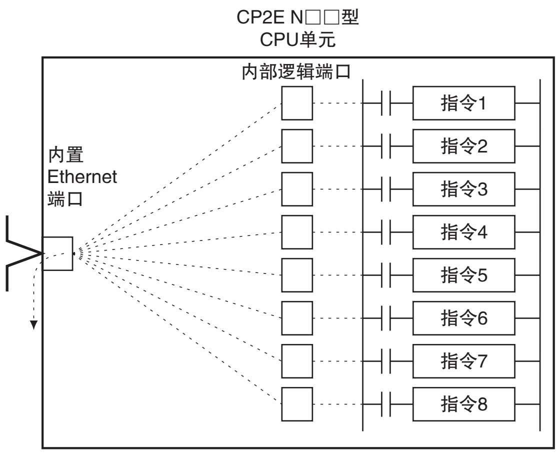
②通过外围服务处理执行通信指令处理。该处理跨多个周期执行。

③返回响应时，通信端口服务内将更新操作数指定的响应数据。此时，将特殊辅助继电器的网络通信指令可执行标志 (A202.00 ~ A202.07) 设为 1(ON)，并更新网络通信执行错误标志 (A219.00 ~ A219.07)、网络通信响应代码 (A203 ~ A210CH)。



网络通信用指令
关于内部逻辑端口

备有 8 个作为逻辑上通信端口的内部逻辑端口。可同时执行 8 个通信指令。1 个内部逻辑端口 1 次只能执行 1 个指令。执行 9 个以上的通信指令时，需编入排他性控制。

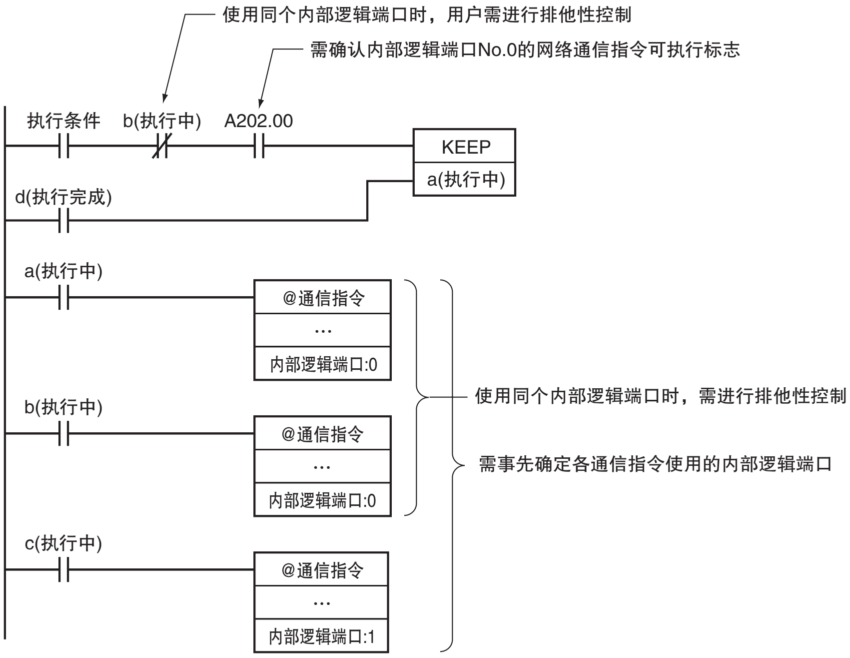


网络通信用指令分为指定内部逻辑端口时和 CPU 单元自动分配时的两种。

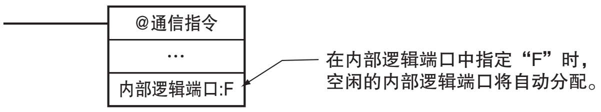
各内部逻辑端口无法同时使用，因此指定内部逻辑端口时，需执行以下任一处理或同时执行两个处理。

・编程时，在确认空闲的内部逻辑端口 No. 的同时，在指令的操作数中指定使用内部逻辑端口 No.
・梯形图程序中，在指令执行前添加确认该内部逻辑端口 No. 是否空闲的处理

未使用自动分配功能时 (例)



内部逻辑端口 No. 指定 “F”而非 “0” ~ “7”，可自动检出此时空闲的内部逻辑端口 No. 进行分配使用。



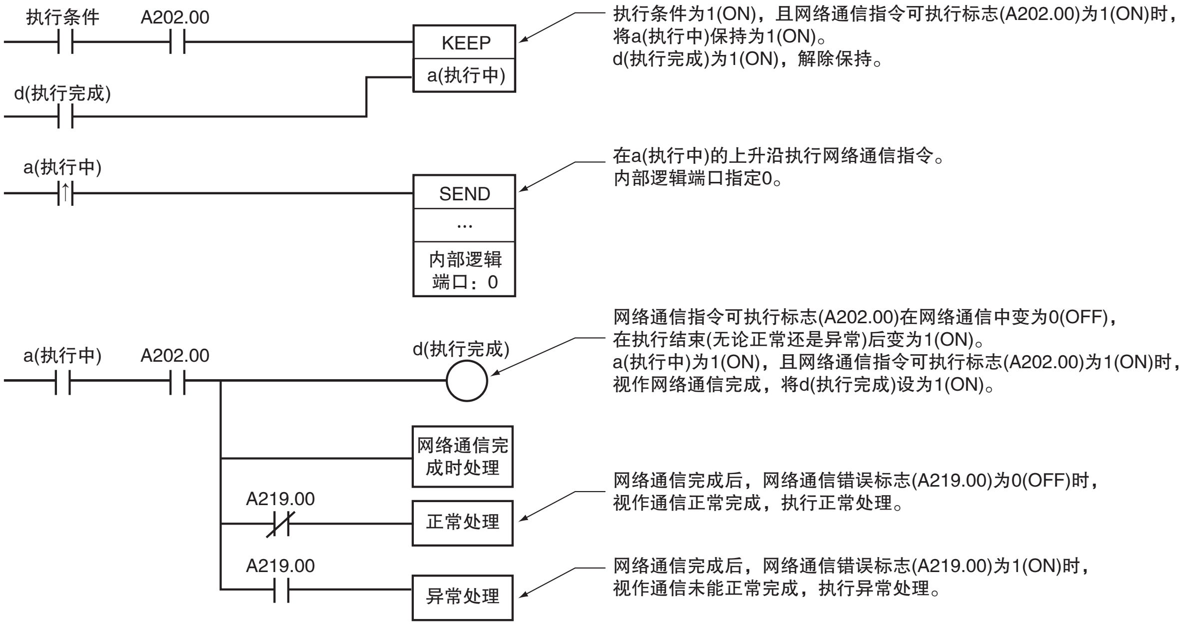
使用自动分配功能，编程时则无需在意内部逻辑端口。

区别如下所示。

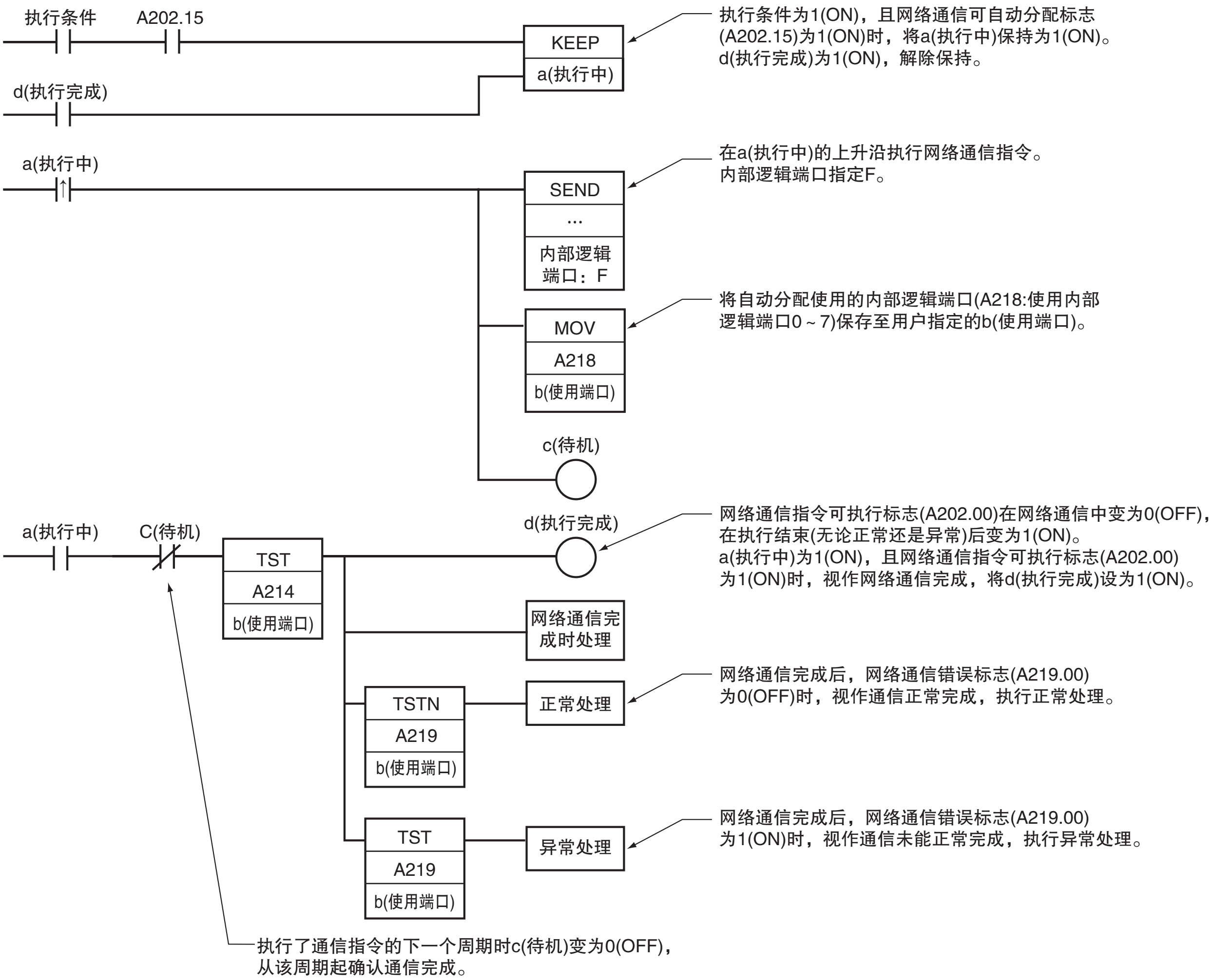
项目	内部逻辑端口用户指定时	内部逻辑端口自动分配时
控制数据内 “内部逻辑端口 No.” 指定	“0” ~ “7”	“F”
排他性控制	需要	不需要 但同时可能会有 9 个以上的通信指令动作时，需要排他性控制。
各种标志的使用方法	使用 LD/LDNOT 指令判别对应指定内部逻辑端口的标志。	使用 “使用内部逻辑端口 No.”(A218CH) 和 TST/TSTN 指令进行判别。
网络通信响应代码	查看对应用户指定的内部逻辑端口的响应代码。	使用 “网络通信响应代码保存地址” (在 A216CH/217CH 中保存 I/O 存储器有效地址) 和索引寄存器间接指定查看响应代码。

梯形图程序示例

● 指定内部逻辑端口时



● 将内部逻辑端口设为自动分配时



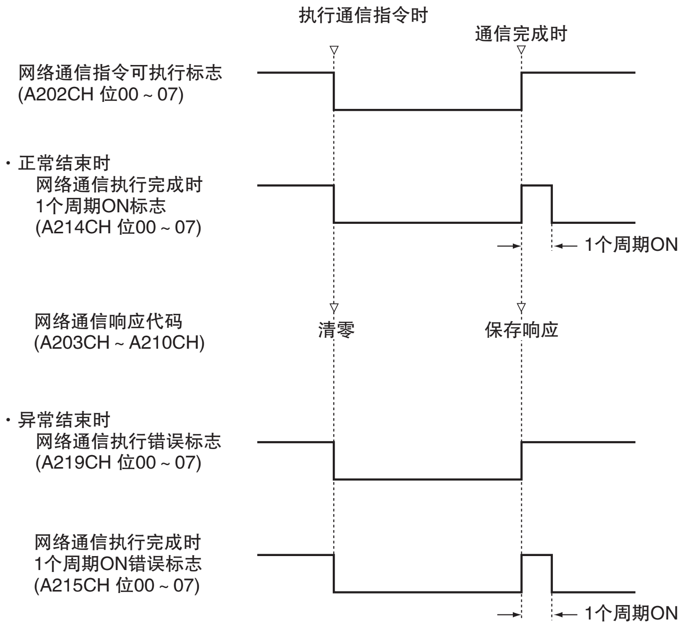
用户指定内部逻辑端口时

指定网络通信用指令的控制数据使用的内部逻辑端口。
指令处理期间，其它指令无法使用同一内部逻辑端口。
需要排他性控制。

● 内部逻辑端口用户指定使用的特殊辅助继电器

地址	位	名称	内容
A202	00 ～ 07	网络通信指令可执行标志	通信指令可执行时变为 1(ON)。各位 00 ～ 07 对应内部逻辑端口 No.0 ～ 7。通信指令执行后根据本标志 OFF → ON，可获知网络通信完成。通信指令执行时变为 0(OFF)。
A203 ～ A210CH	–	网络通信响应代码	设置执行了网络通信时的响应代码。
A219CH	00 ～ 07	网络通信执行错误标志	网络通信执行中发生错误时变为 1(ON)。本标志变为 1(ON) 时，请查看网络通信响应代码 (A203 ～ A210CH)，确定错误原因。各位 00 ～ 07 对应内部逻辑端口 No.0 ～ 7。通信指令执行时变为 0(OFF)。

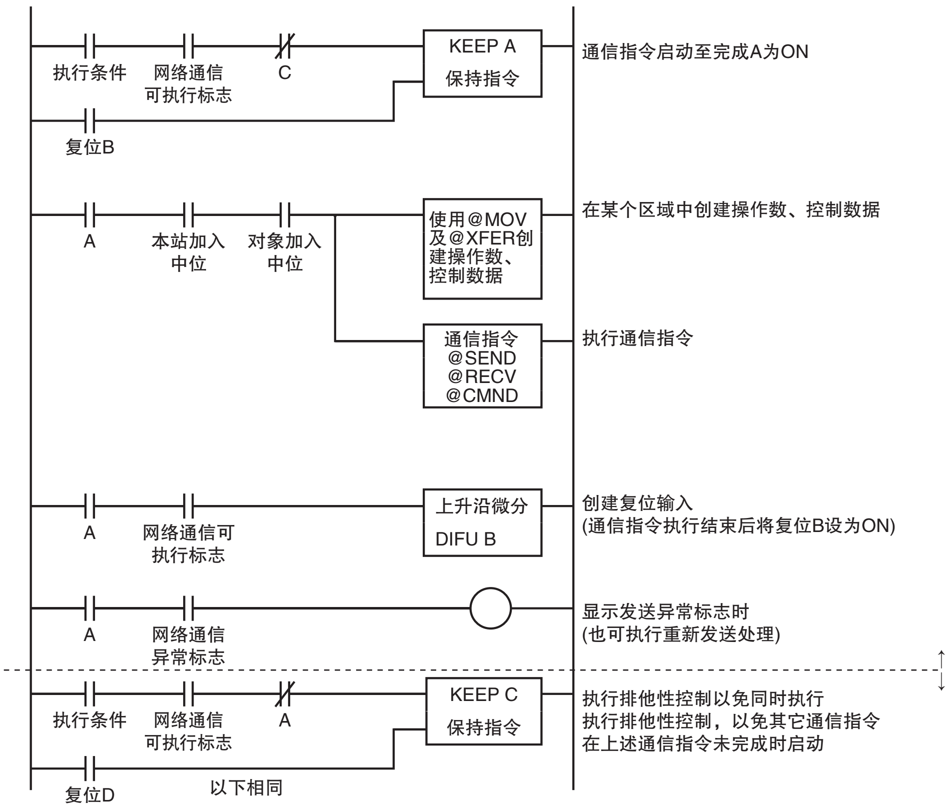
● 各种标志的动作



使用方法

1 在指令的操作数中将 16 进制 #0 ～ 7 指定为内部逻辑端口。

2 请使用网络通信可执行标志 (A202.00 ～ A202.07) 编入排他性控制进行使用。



内部逻辑端口的自动分配功能

● 概要

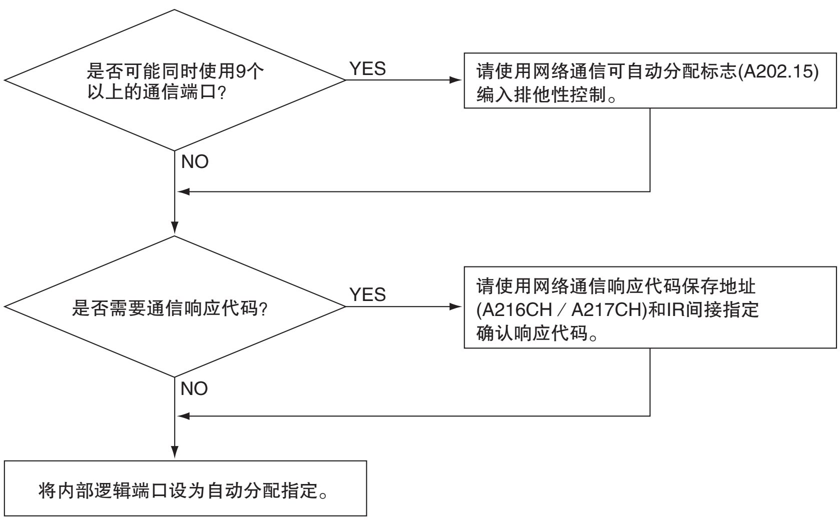
以下指令使用内部逻辑 No.0 ～ 7 中的 1 个执行网络通信。

・ 网络通信用指令 (SEND、RECV、CMND)

● 内部逻辑端口自动分配使用的特殊辅助继电器

地址	位	名称	内容
A202CH	15	网络通信可自动分配标志	可通过内部逻辑端口自动分配执行通信指令时，即可用于自动分配的内部逻辑端口有空闲时变为 1(ON)。 参考：同时使用 9 个以上的通信指令时，根据该标志确认自动分配可使用的内部逻辑端口是否有空闲后执行通信指令。
A214CH	00 ～ 07	网络通信完成时 1 个周期 ON 标志	通信完成时仅 1 个周期变为 1(ON)。各位 00 ～ 07 对应内部逻辑端口 No.0 ～ 7。 注：本标志在执行了通信指令的下一个周期起生效，因此需在梯形图程序中待机 1 个周期后再查看，避免在执行了通信指令的周期中查看。参考：根据保存在 A218CH(使用内部逻辑端口 No.) 中的使用内部逻辑端口 No.，指定位 00 ～ 07 中的任意一个进行查看。
	08 ～ 15	禁用	–
A215CH	00 ～ 07	网络通信完成时 1 个周期 ON 错误标志	若在通信完成时发生错误，仅 1 个周期变为 1(ON)。各位 00 ～ 07 对应内部逻辑端口 No.0 ～ 7。本标志变为 1(ON) 时，请查看网络通信响应代码 (A203 ～ A210CH)，确定错误原因。 注：本标志在执行了通信指令的下一个周期起生效，因此需在梯形图程序中待机 1 个周期后再查看，避免在执行了通信指令的周期中查看。 参考：根据保存在 A218CH(使用内部逻辑端口 No.) 中的使用内部逻辑端口 No.，指定位 00 ～ 07 中的任意一个进行查看。
	08 ～ 15	禁用	–
A216 ～ A217CH	–	网络通信响应代码保存地址	执行通信指令后，通信响应代码将自动保存至特殊辅助继电器的任意通道地址。这里保存该任意通道地址的 I/O 存储器有效地址。 参考：将这里保存的 I/O 存储器有效地址传送至索引寄存器 IR，根据索引寄存器间接指定，可读取任意通道地址中保存的通信响应代码。
A218CH	–	使用内部逻辑端口 No.	执行通信指令后，这里保存使用了的内部逻辑端口 No.。 0 ～ 7：内部逻辑端口 No.0 ～ 7

是否使用网络通信可自动分配标志 (A202.15)、网络通信响应代码保存地址 (A216CH/A217CH) 的判断如下所示。

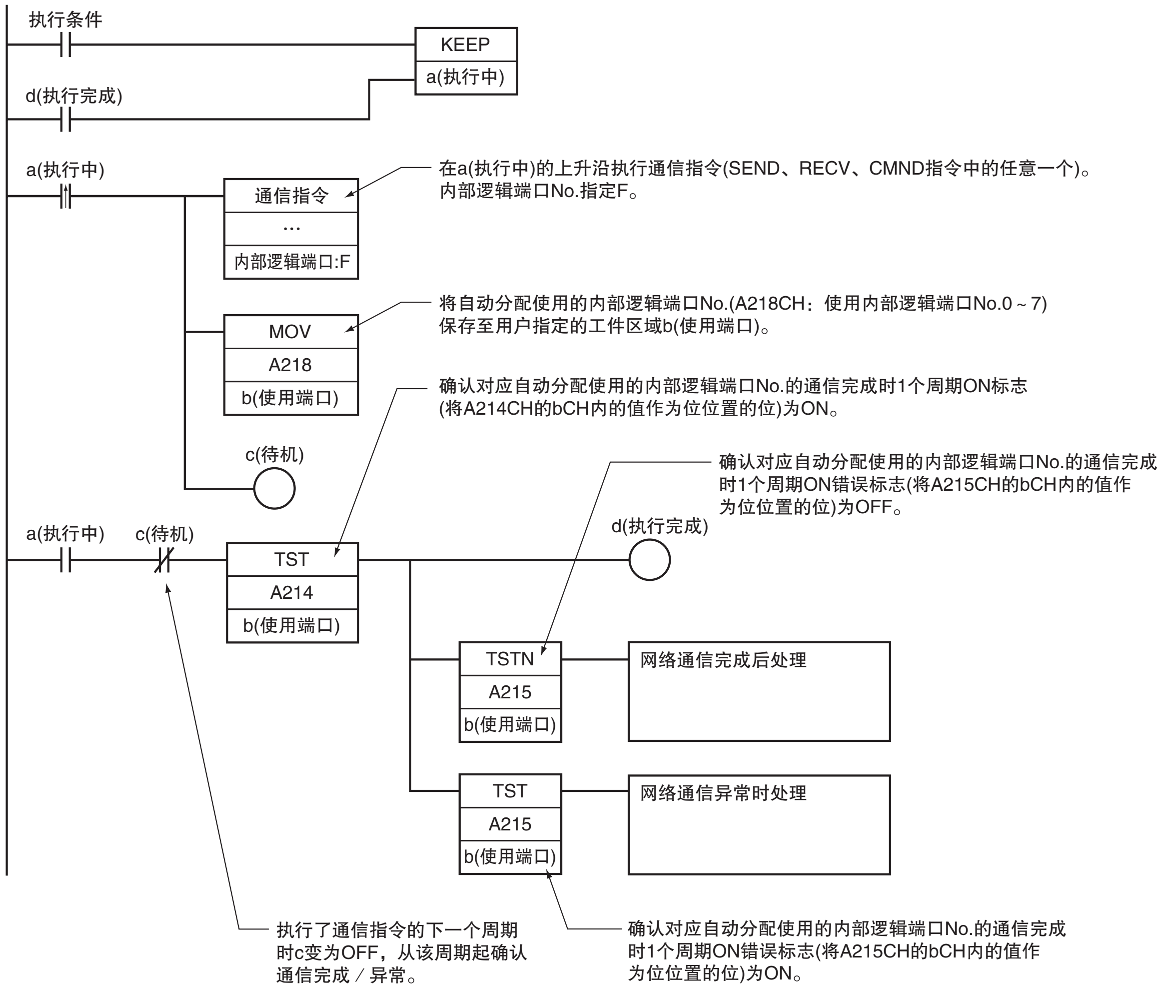


使用方法

1 在指令的操作数中将 16 进制 #F 指定为内部逻辑端口 No.。

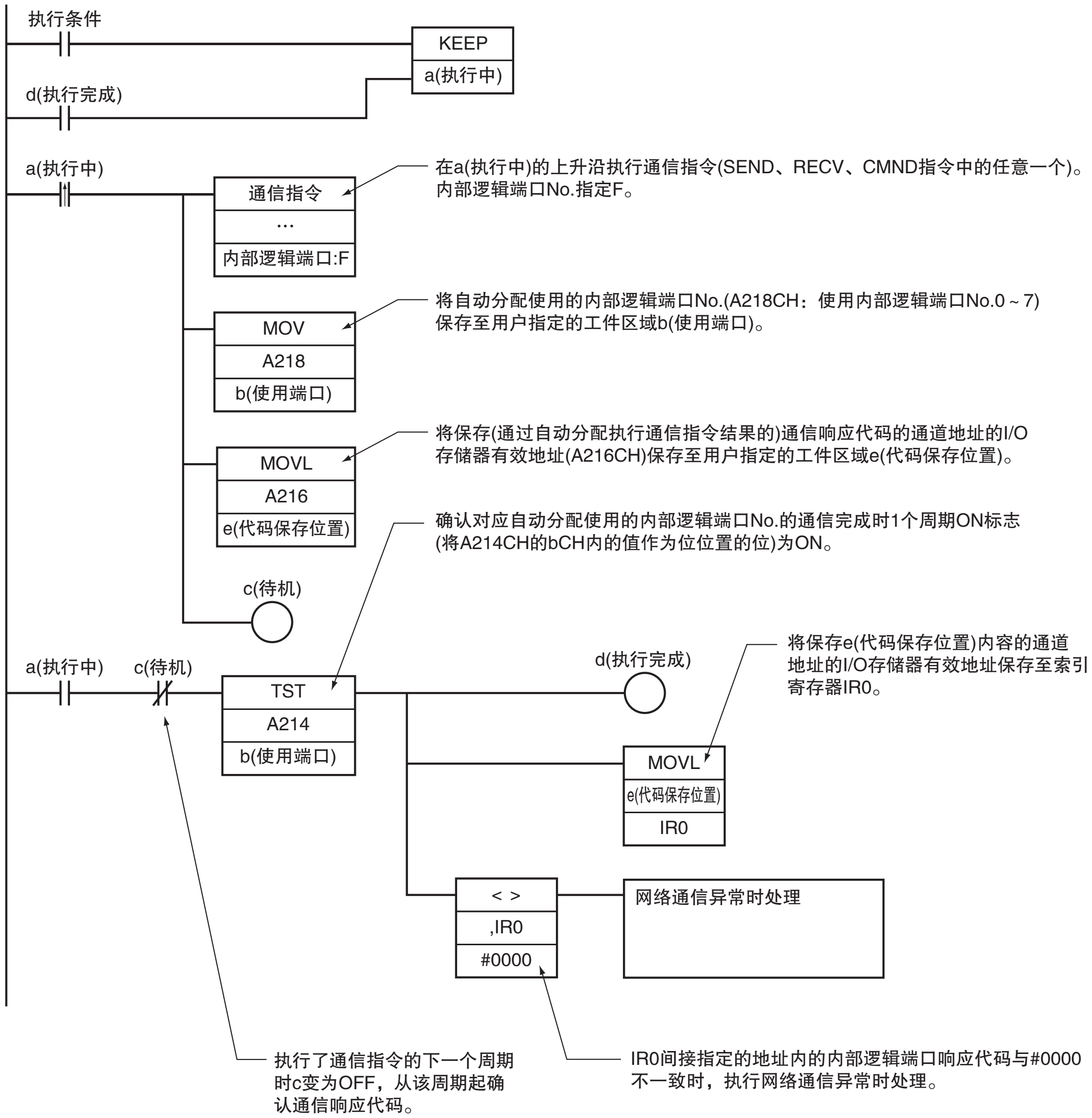
2 请务必编入以下梯形图程序进行使用。

● 执行通信指令后网络通信完成 / 执行异常时处理时

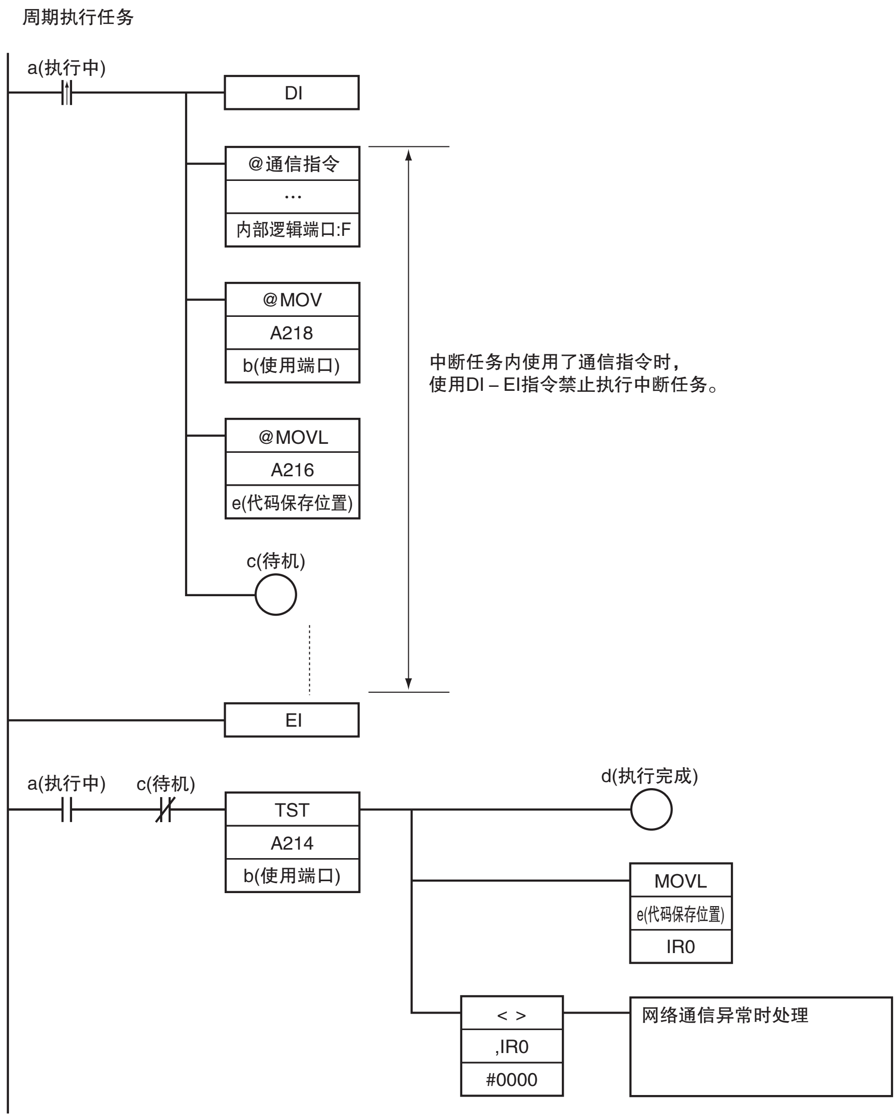


● 执行通信指令后查看网络通信响应代码时

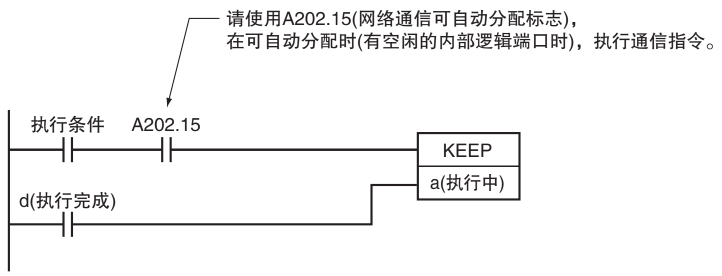
网络通信响应代码主要用于判断发生错误时的原因，也可如下所述，检查是否为 #0000，确认是否正常结束。



注 1 中断任务内也使用通信指令时（内部逻辑端口自动分配或内部逻辑端口用户指定均可），在执行周期执行任务内使用内部逻辑端口自动分配的通信指令前后，如下所述，请使用 DI 指令和 EI 指令禁止执行中断任务。

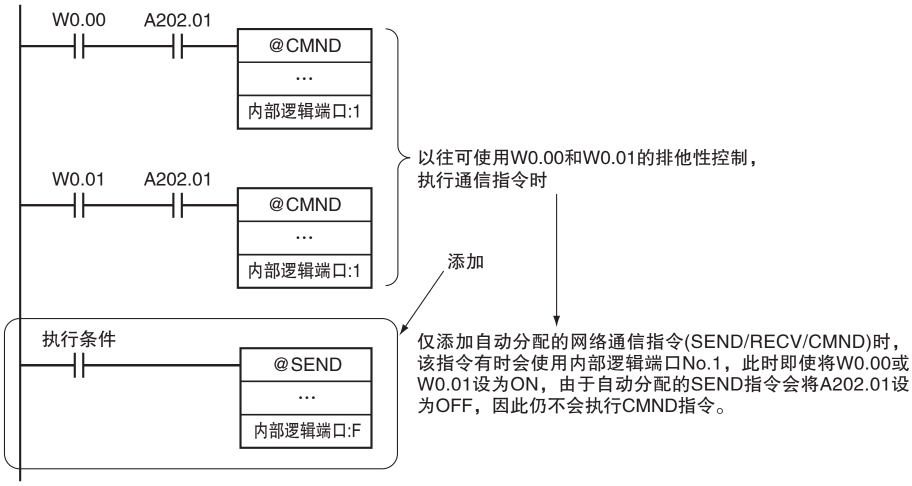


2 可能会同时执行 9 个以上的通信指令时，即使执行自动分配，如下所述，也请在执行前编入确认是否有空闲的内部逻辑端口的梯形图程序。



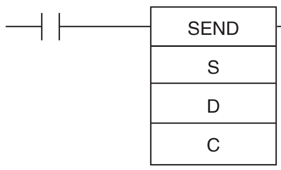
3 通过通信指令，可混用内部逻辑端口用户指定和内部逻辑端口自动分配。但根据各通信指令的执行条件，内部逻辑端口自动分配的通信指令可能会使用用户指定的内部逻辑端口。因此，尤其如下所示，在现有的梯形图程序中添加自动分配的通信指令时需加以注意。

例)



SEND

指令	助记符	变化	功能代码	功能
网络发送	SEND	@SEND	090	对网络上的节点发送数据

符号	SEND
	 S：发送源(本机节点)发送开始CH编号 D：发送目标(对象节点)接收开始CH编号 C：控制数据低位CH编号

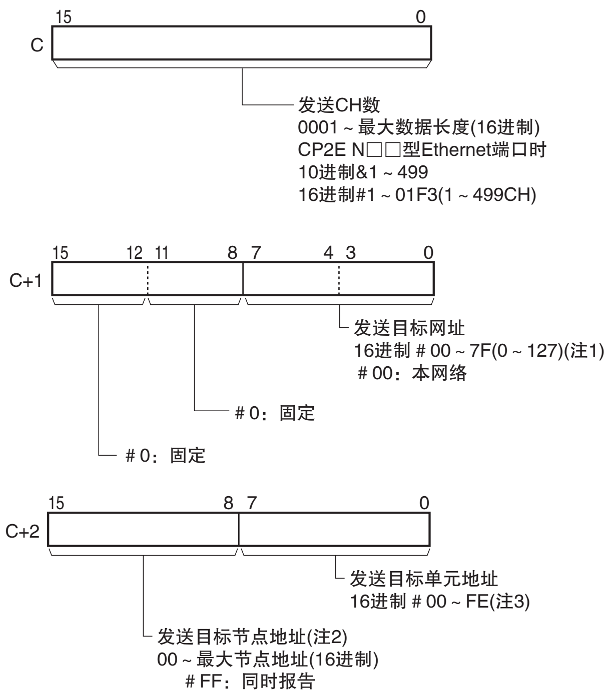
适用程序区

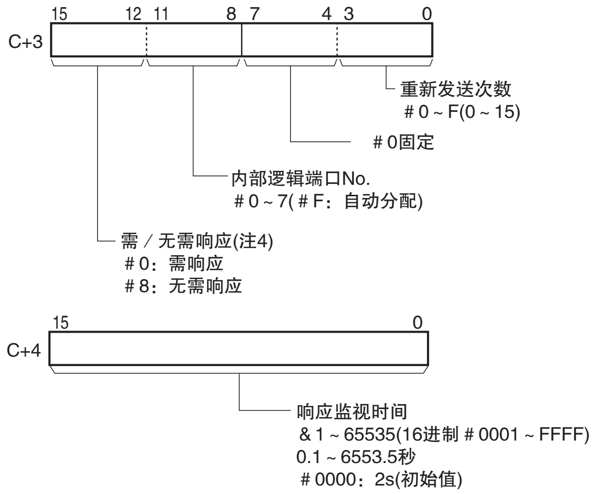
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	发送源 (本机节点) 发送开始 CH 编号	UINT	可变
D	发送目标 (对象节点) 接收开始 CH 编号	UINT	可变
C	控制数据低位 CH 编号	WORD	5

C：控制数据低位 CH 编号





注 1 将发送目标网络地址设为 #00 时，将发送至与 CP2E N □□型的内置 Ethernet 端口相同的网络中。

2 同时报告发送时，设定为 16 进制 #FF。
无法发送至本单元 (发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00)。

3 单元地址：

・ CPU 单元：16 进制 #00

・ 计算机：16 进制 #01

* 也可将 CS/CJ 系列的 CPU 高功能单元及高功能 I/O 单元、串行端口指定为地址。单元地址的设定请参阅 “SYSMAC CS/CJ 系列 通信指令基准手册”(手册编号：W342)。

4 执行同时报告发送时，即使设定为需响应也不会返回响应。

● 区域种类

内容	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S D C	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

名称	标记	内容
错误标志	P_ER	・对应 C 指定的内部逻辑端口 No. 的网络指令可执行标志为 OFF 时 ON ・发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00 时 ON ・除此以外时，OFF

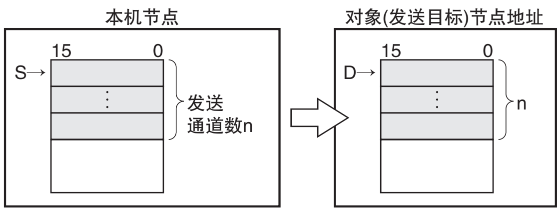
相关特殊辅助继电器

名称	地址	内容
网络通信指令可执行标志	A202.00 ~ A202.07	可执行网络通信时变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ~ 07：内部逻辑端口 No.00 ~ 07 网络通信执行中变为 0(OFF)，在执行结束 (无论正常还是异常) 后变为 1(ON)。
网络通信执行错误标志	A219.00 ~ A219.07	执行网络通信时发生错误 (异常) 的情况下变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ~ 07：内部逻辑端口 No.00 ~ 07 在下一次网络通信执行前将保持状态。即使异常结束，在下一次通信指令执行时仍会变为 0(OFF)。
网络通信响应代码	A203CH ~ A210CH	执行网络通信后，将保存响应代码 (终止符)。 各 CH 对应内部逻辑端口 No.。 A203 ~ 210CH：内部逻辑端口 No. 00 ~ 07 通信指令执行中变为 #0000，通信指令执行结束时得到反映。指令执行时清除。

注 关于网络通信响应代码，请参阅 “CS/CJ 系列　通信指令参考手册” (手册编号：W342) 的 FINS 指令终止符一览。

功能

・从 S 指定的发送源 (本机节点) 发送开始 CH 编号将发送 CH 数的数据通过 CPU 总线或网络进行发送，经由 C 指定的发送目标网络地址的发送目标节点地址，从指定发送目标单元地址 (装置) 的 D 指定的接收开始 CH 进行写入。



・C 指定的内部逻辑端口 No. 为可执行 (网络通信指令可执行标志为 ON) 时，将指定内部逻辑端口 No. 的网络通信执行错误标志 (内部逻辑端口 No.00 ~ 07：A219.00 ~ A219.07) 和网络通信可执行标志 (内部逻辑端口 No.00 ~ 07：A202.00 ~ A202.07) 设为 OFF，在网络通信响应代码中写入 #0000。然后，对网络上的节点发送数据。

・将数据发送至本网络以外 (跨网络) 时，需将路由表 (登录了至连接通信对象 PLC 的网络的通信线路的表格) 登录至各网络上的 PLC(CPU 单元)。

・发送目标节点地址为 #FF 时，发送至指定网络地址的所有节点。(同时报告)

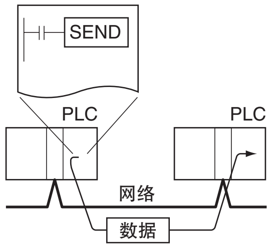
・设定为需响应时，若响应监视时间内无响应，则在有响应前按指定重新发送次数重新发送数据。

・指定同时报告时，无响应，也不重新发送。

可对网络上的 PLC 或计算机发送数据。

● 对网络上的 PLC(CPU 单元) 或计算机发送数据

对网络上的 PLC(CPU 单元) 或计算机执行了本 SEND 指令时，可将指定 I/O 存储器区域的数据发送至对象的指定 I/O 存储器区域。



注意事项

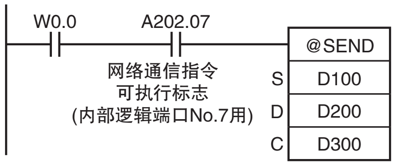
· 对 1 个内部逻辑端口 1 次只能执行 1 个指令。因此，对应各端口的网络通信指令可执行标志为 0(OFF) 时，请将网络通信指令可执行标志 (A202.00 ~ A202.07) 设为 a 接点的输入条件，以免对该端口执行本指令。

· 关于通过自动分配 (“F”) 使用内部逻辑端口 No. 时，请参阅 “第 2 章　各指令的说明” 中 “网络通信用指令” 的 “内部逻辑端口的自动分配功能” 的说明。

· 考虑干扰等导致发送信息及响应在通信途中消失等情况，执行 SEND 指令时，请将重新发送次数设定为 0 以外。

· 仅可使用 CP2E N □□型的 Ethernet 端口。
无法用于 CP1E、CP2E 的串行通信端口。

程序举例

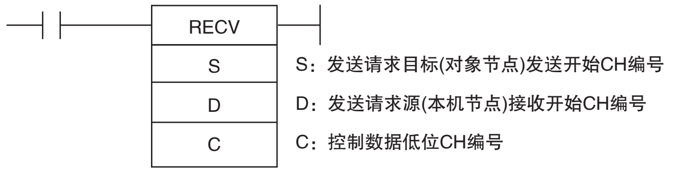


W0.0及A202.07(内部逻辑端口No.7用网络通信指令可执行标志)为ON时，将从D100起10CH(D100 ~ D109)的数据发送至本网络上节点地址3的CPU单元从D200起的10CH(D200 ~ D209)。响应监视时间10秒后未返回响应时，最多执行3次重新发送处理

C：D300	0 0 0 A	发送CH＝10CH
C+1：D301	0 0 0 0	对于本网络上的装置本身
C+2：D302	0 3 0 0	发送目标节点地址＝＃03、单元地址＝＃00(CPU单元)
C+3：D303	0 7 0 3	需响应、内部逻辑端口7、重新发送次数＝＃03
C+4：D304	0 0 6 4	响应监视时间＝＃0064(10秒)

RECV

指令	助记符	变化	功能代码	功能
网络接收	RECV	@RECV	098	对网络上的节点请求发送，接收数据。

	RECV
符号	

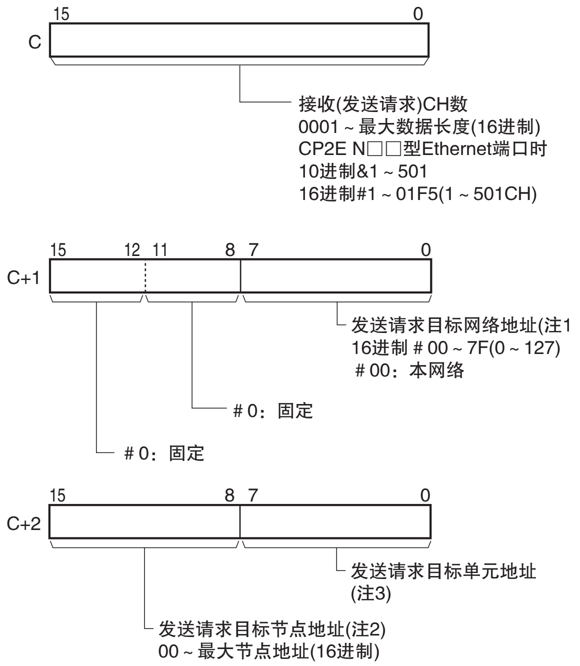
适用程序区

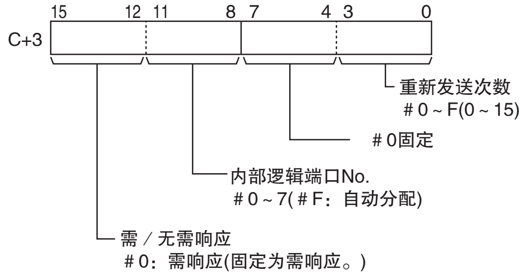
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

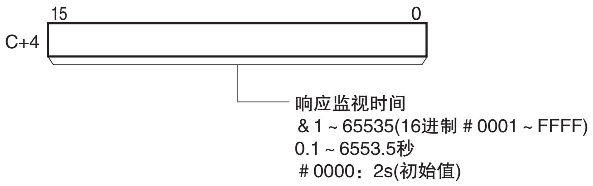
操作数

操作数	描述	数据类型	大小
S	发送请求目标 (对象节点) 发送开始 CH 编号	UINT	可变
D	发送请求源 (本机节点) 接收开始 CH 编号	UINT	可变
C	控制数据低位 CH 编号	WORD	5

C：控制数据低位 CH 编号







注 1 将发送请求目标网络地址设为 #00 时，对与 CP2E N □□型的内置 Ethernet 端口相同的网络执行发送请求。

2 RECV 指令无法进行同时报告发送。
无法发送至本单元 (发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00)。

3 单元地址

・ CPU 单元：16 进制 #00

・ 计算机：16 进制 #01

* 也可将 CS/CJ 系列的 CPU 高功能单元及高功能 I/O 单元、串行端口指定为地址。单元地址的设定请参阅“SYSMAC CS/CJ 系列 通信指令参考手册”(手册编号：W342)。

● 区域种类

内容	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S																
D	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---
C																

条件标志

名称	标记	内容
错误标志	P_ER	・对应 C 指定的内部逻辑端口 No. 的网络指令可执行标志为 OFF 时 ON ・发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00 时 ON ・除此以外时，OFF

相关特殊辅助继电器

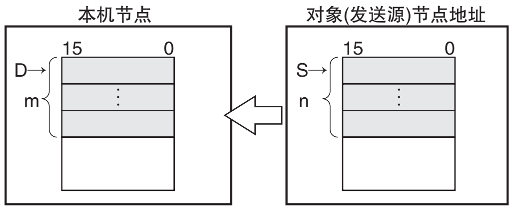
名称	地址	内容
网络通信指令可执行标志	A202.00 ～ A202.07	可执行网络通信时变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ～ 07：内部逻辑端口 No.00 ～ 07 网络通信执行中变为 0(OFF)，在执行结束 (无论正常还是异常) 后变为 1(ON)。
网络通信执行错误标志	A219.00 ～ A219.07	执行网络通信时发生错误 (异常) 的情况下变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ～ 07：内部逻辑端口 No.00 ～ 07 在下一次网络通信执行前将保持状态。即使异常结束，在下一次通信指令执行时仍会变为 0(OFF)。

名称	地址	内容
网络通信响应代码	A203CH ~ A210CH	执行网络通信后，将保存响应代码 (终止符)。 各 CH 对应内部逻辑端口 No.。 A203 ~ 210CH：内部逻辑端口 No.00 ~ 07 通信指令执行中变为 #0000，通信指令执行结束时得到反映。 运行开始时清除。

注 关于网络通信响应代码，请参阅 “CS/CJ 系列　通信指令参考手册”(手册编号：W342) 的 FINS 指令终止符一览。

功能

在 C 指定的网络地址的发送请求目标 (对象) 节点地址经由发送请求目标单元地址 (装置) 中，从 S 指定的发送开始 CH 通过 CPU 总线或网络接收 C 指定的接收 CH 数的数据，并从 D 指定的发送请求源 (本机节点) 接收开始 CH 进行写入。

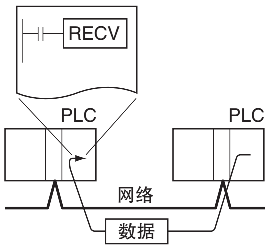


- C 指定的内部逻辑端口 No. 为可执行 (网络通信指令可执行标志为 ON) 时，将指定内部逻辑端口 No. 的网络通信执行错误标志 (内部逻辑端口 No.00 ~ 07：A219.00 ~ A219.07) 和网络通信可执行标志 (内部逻辑端口 No.00 ~ 07：A202.00 ~ A202.07) 设为 OFF，在网络通信响应代码中写入 #0000。然后，接收网络上节点的数据。
- 接收网络以外的数据 (跨网络) 时，需将路由表 (登录了至连接通信对象 PLC 的网络的通信线路的表格) 登录至各网络上的 PLC(CPU 单元)。
- RECV 指令作为响应需返回数据，因此设定为需响应。
  设定为需响应时，若响应监视时间内无响应，则在有响应前按指定重新发送次数执行发送请求。

可从网络上的 PLC 或计算机接收数据。

● 从网络上的 PLC(CPU 单元) 或计算机接收数据

对网络上的 PLC(CPU 单元) 或计算机执行了本 RECV 指令时，可将对象的指定 I/O 存储器区域的数据接收至自身的指定 I/O 存储器区域。

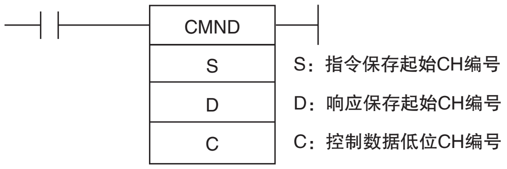


注意事项

- 对 1 个内部逻辑端口 1 次只能执行 1 个指令。因此，对应各端口的网络通信指令可执行标志为 0(OFF) 时，请将网络通信指令可执行标志 (A202.00 ~ A202.07) 设为 a 接点的输入条件，以免对该端口执行本指令。
- 关于通过自动分配 (“F”) 使用内部逻辑端口 No. 时，请参阅 “第 2 章　各指令的说明” 中 “网络通信用指令” 的 “内部逻辑端口的自动分配功能” 的说明。
- 考虑干扰等导致发送请求信息及响应在通信途中消失等情况，执行 RECV 指令时，请将重新发送次数设定为 0 以外。
- 仅可使用 CP2E N □□型的 Ethernet 端口。
  无法用于 CP1E、CP2E 的串行通信端口。

CMND

指令	助记符	变化	功能代码	功能
指令发送	CMND	@CMND	490	对网络上的节点发送任意 FINS 指令，并接收响应。

符号	CMND
	

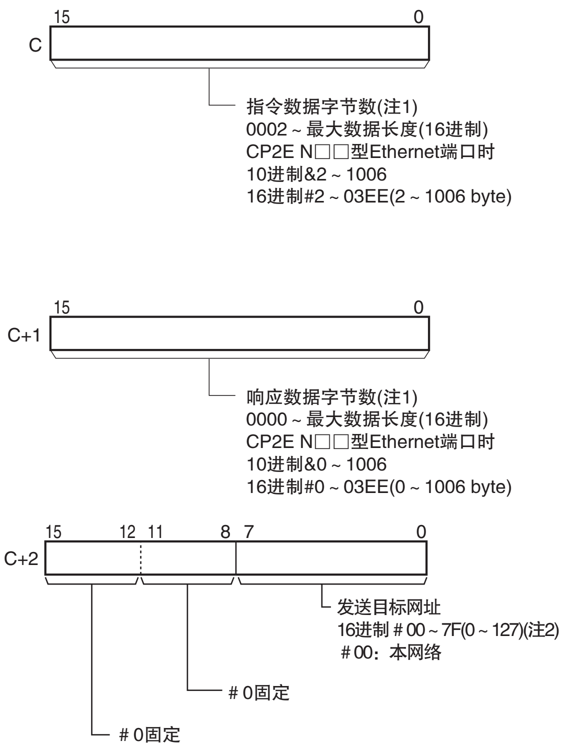
适用程序区

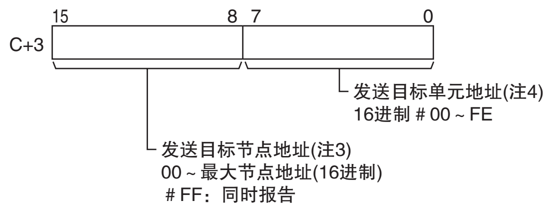
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

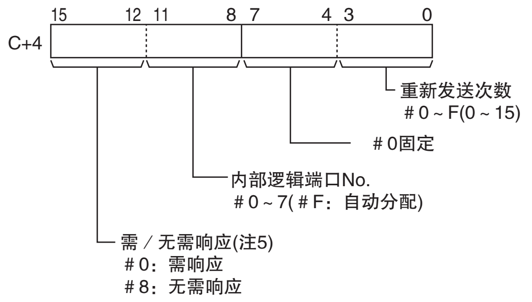
操作数

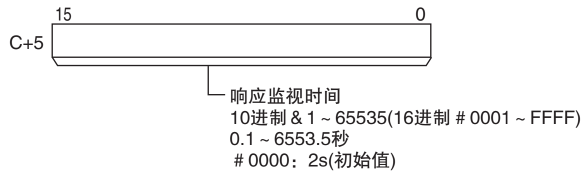
操作数	描述	数据类型	大小
S	指令保存起始 CH 编号	UINT	可变
D	响应保存起始 CH 编号	UINT	可变
C	控制数据低位 CH 编号	WORD	6

C：控制数据低位 CH 编号









注 1 指令数据字节数及响应数据字节数的最大数据长度请参阅所用网络的单元手册。
指令数据字节数及响应数据字节数的最大数据长度取决于 FINS 指令线路的网络中数据长度最小的网络。
返回比响应数据字节数长的响应数据时，不会保存响应数据。返回比响应数据字节数短的响应数据时，将接收响应数据，剩余区域保持之前的状态。

2 将发送目标网络地址设为 #00 时，将发送至与 CP2E N □□型的内置 Ethernet 端口相同的网络中。

3 同时报告发送时，设定 #FF。
无法发送至本单元 (发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00)。

4 单元地址
- CPU 单元：#00(16 进制)
- 计算机：#01(16 进制)

* 也可将 CS/CJ 系列的 CPU 高功能单元及高功能 I/O 单元、串行端口指定为地址。单元地址的设定请参阅“SYSMAC CS/CJ 系列 通信指令参考手册”(手册编号：W342)。

5 执行同时报告发送时，即使设定为需响应也不会返回响应。

● 区域种类

<table>
<tr><th rowspan="2">内容</th><th colspan="7">字地址</th><th colspan="2">DM/EM 间接地址</th><th rowspan="2">常数</th><th colspan="3">寄存器</th><th rowspan="2">CF</th><th rowspan="2">脉冲位</th><th rowspan="2">TR 位</th></tr>
<tr><th>CIO</th><th>WR</th><th>HR</th><th>AR</th><th>T</th><th>C</th><th>DM</th><th>@DM</th><th>*DM</th><th>DR</th><th>IR 直接</th><th>IR 间接</th></tr>
<tr><td>S</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">OK</td><td rowspan="3">---</td><td rowspan="3">---</td><td rowspan="3">---</td><td rowspan="3">OK</td><td rowspan="3">---</td><td rowspan="3">---</td><td rowspan="3">---</td></tr>
<tr><td>D</td></tr>
<tr><td>C</td></tr>
</table>

条件标志

名称	标记	内容
错误标志	P_ER	・对应 C 指定的内部逻辑端口 No. 的网络指令可执行标志为 OFF 时 ON ・发送目标网络地址 #00、发送目标节点地址 #00、发送目标单元地址 #00 时 ON ・除此以外时，OFF

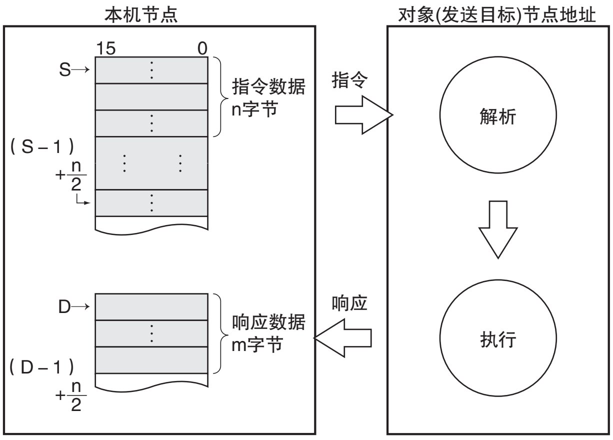
相关特殊辅助继电器

名称	地址	内容
网络通信指令可执行标志	A202.00 ～ A202.07	可执行网络通信时变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ～ 07：内部逻辑端口 No.00 ～ 07 网络通信执行中变为 0(OFF)，在执行结束 (无论正常还是异常) 后变为 1(ON)。
网络通信执行错误标志	A219.00 ～ A219.07	执行网络通信时发生错误 (异常) 的情况下变为 1(ON)。 各位对应内部逻辑端口 No.。 位 00 ～ 07：内部逻辑端口 No.00 ～ 07 在下一次网络通信执行前将保持状态。即使异常结束，在下一次通信指令执行时仍会变为 0(OFF)。
网络通信响应代码	A203CH ～ A210CH	执行网络通信后，将保存响应代码 (终止符)。 各 CH 对应内部逻辑端口 No.。 A203 ～ 210CH：内部逻辑端口 No.00 ～ 07 通信指令执行中变为 #0000，通信指令执行结束时得到反映。 运行开始时清除。

注 关于网络通信响应代码，请参阅 “CS/CJ 系列　通信指令参考手册”(手册编号：W342) 的 FINS 指令终止符一览。

功能

从 S 指定的指令保存起始 CH 编号，通过 CPU 总线或网络经由 C 指定的网络地址的节点地址将指令数据字节数的任意 FINS 指令数据发送至指定单元地址 (装置)。以 D 为开头按响应数据字节数保存该响应。

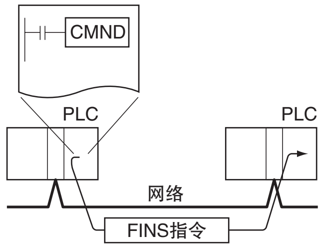


- C 指定的内部逻辑端口 No. 为可执行 (网络通信指令可执行标志为 ON) 时，将指定内部逻辑端口 No. 的网络通信执行错误标志 (端口 No.00 ～ 07：A219.00 ～ A219.07) 和网络通信可执行标志 (内部逻辑端口 No.00 ～ 07：A202.00 ～ A202.07) 设为 OFF，在网络通信响应代码中写入 #0000。然后，对网络上的节点发送 FINS 指令。
- 发送目标的节点通过路由表 (登录了至连接通信对象 PLC 的网络的通信线路的表格) 进行查找。
- 除了对装置外，也可对装置的串行端口执行本指令。
- 发送目标节点地址为 #FF 时，对指定网络地址的所有节点发送同一指令。(同时报告)
- 设定为需响应时，若响应监视时间内无响应，则在有响应前按指定重新发送次数发送指令。请对不存在响应的指令设定无需响应。
- 指定同时报告时，无响应，也不重新发送。
- 比设定的响应数据字节数多的数据作为响应返回时，将发生错误。

可对网络上的 PLC(CPU 单元或 CPU 高功能单元) 或计算机发送任意 FINS 指令。

● 对网络上的 PLC(CPU 单元或 CPU 高功能单元) 或计算机发送任意 FINS 指令

对网络上的 PLC(CPU 单元或 CPU 高功能单元) 或计算机执行了 CMND 指令时，可对对象发送任意 FINS 指令。



提示

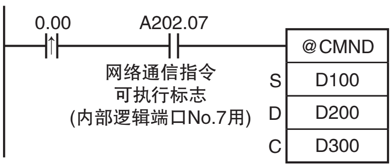
- 通过本指令发送 FINS 指令的 I/O 存储器区域的写入指令 (指令代码：#0102) 与 SEND 指令等同。
  通过本指令发送 FINS 指令的 I/O 存储器区域的读取指令 (指令代码：#0101) 与 RECV 指令等同。

注意事项

- 对 1 个内部逻辑端口 1 次只能执行 1 个指令。因此，对应各端口的网络通信指令可执行标志为 0(OFF) 时，请将网络通信指令可执行标志 (A202.00 ~ A202.07) 设为 a 接点的输入条件，以免对该端口执行本指令。
- 关于通过自动分配 (“F”) 使用内部逻辑端口 No. 时，请参阅 “第 2 章　各指令的说明” 中 “网络通信用指令” 的 “内部逻辑端口的自动分配功能” 的说明。
- 考虑干扰等导致 FINS 指令信息及响应在通信途中消失等情况，执行 CMND 指令时，请将重新发送次数设定为 0 以外。
- 仅可使用 CP2E N □□型的 Ethernet 端口。
  无法用于 CP1E、CP2E 的串行通信端口。

程序举例

● 对网络上的其它 CPU 单元发送 FINS 指令



0.00及A202.07(内部逻辑端口No.7用网络通信指令可执行标志)为ON时，对本网络上节点地址3的CPU单元发送FINS指令＃0101(I/O存储器区域的读取指令)，并将响应保存至D200～D211。
(FINS指令仅读取D10～D19的10CH。
响应在指令代码＃0101的2字节和终止符的2字节后返回10CH的数据：共14CH＝28字节)

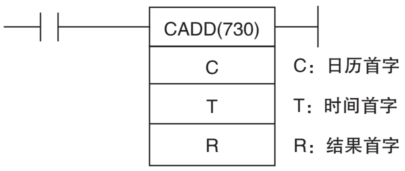
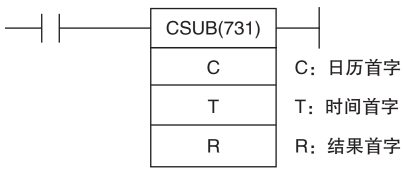
	15		87	0	
S：D100	0	1	0	1	指令代码＃0101＝I/O存储器区域的读取
S+1：D101	8	2	0	0	D10(I/O存储器种类＝＃82、地址＝＃000A00)
S+2：D102	0	A	0	0	
S+3：D103	0	0	0	A	读取元素数＝＃000A(10)

	15		87	0	
C：D300	0	0	0	8	指令数据字节数＝＃0008(8)
C+1：D301	0	0	1	8	响应数据字节数＝＃0018 24)
C+2：D302	0	0	0	0	对于本网络上的装置本身
C+3：D303	0	3	0	0	发送目标节点地址＝＃03、单元地址＝＃00(CPU单元)
C+4：D304	0	7	0	3	需响应、内部逻辑端口7、重新发送次数＝＃03
C+5：D305	0	0	6	4	响应监视时间＝＃0064(10秒)

时钟指令

CADD/CSUB

指令	助记符	变化	功能代码	功能
日历加	CADD	@CADD	730	将指定字中的日历数据和时间相加。
日历减	CSUB	@CSUB	731	从指定字中的日历数据减去时间。

	CADD	CSUB
符号		

适用程序区

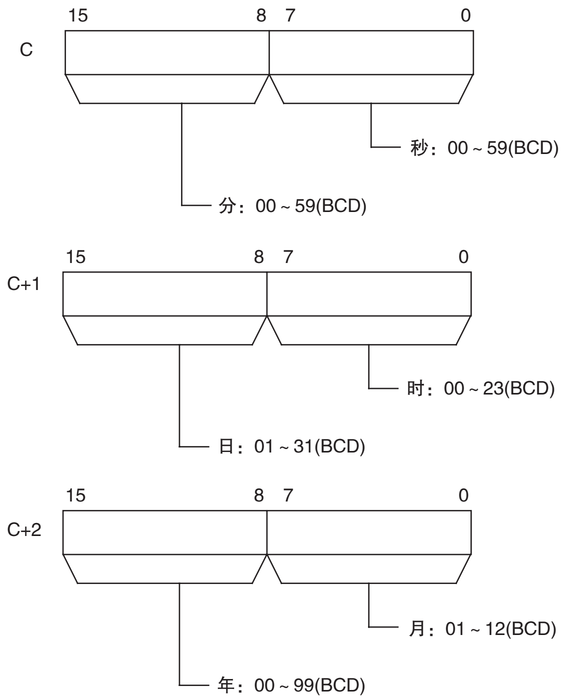
区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

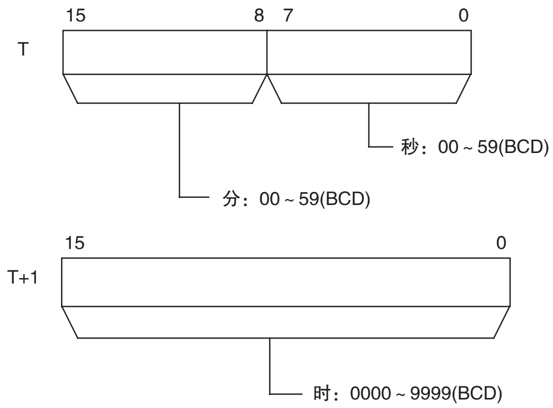
操作数	描述	数据类型	大小
C	日历首字	WORD	3
T	时间首字	DWORD	2
R	结果首字	WORD	3

- CADD

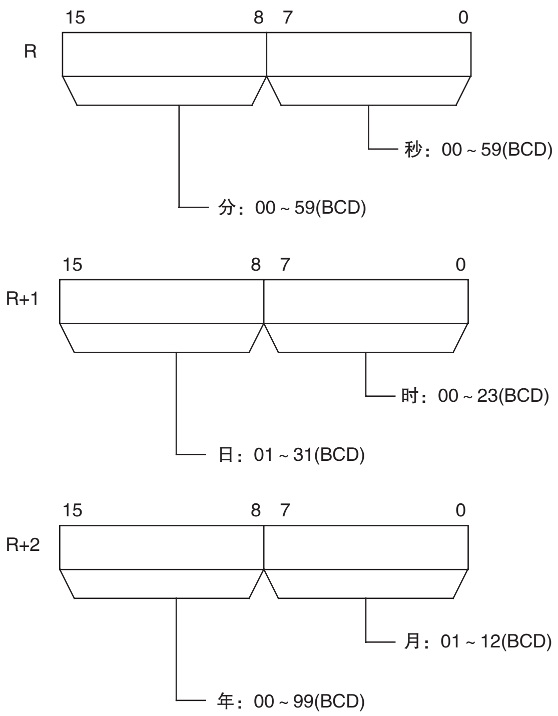
C ～ C+2：日历数据



T 和 T+1：时间数据

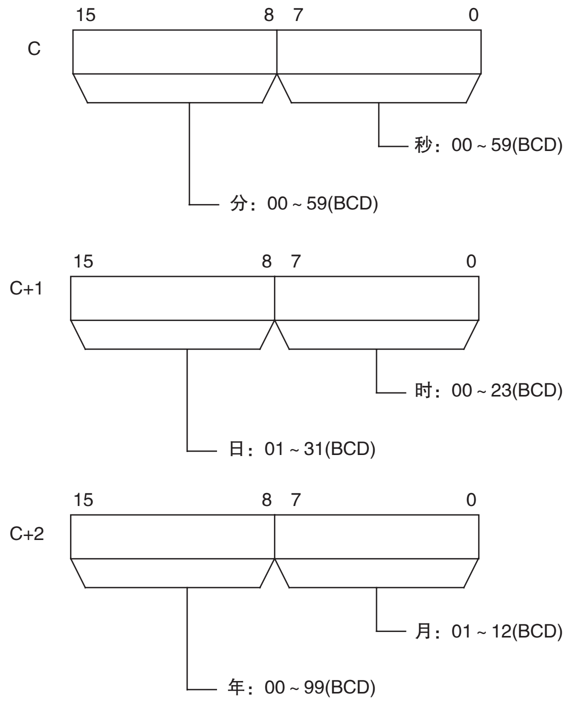


R ～ R+2：结果数据

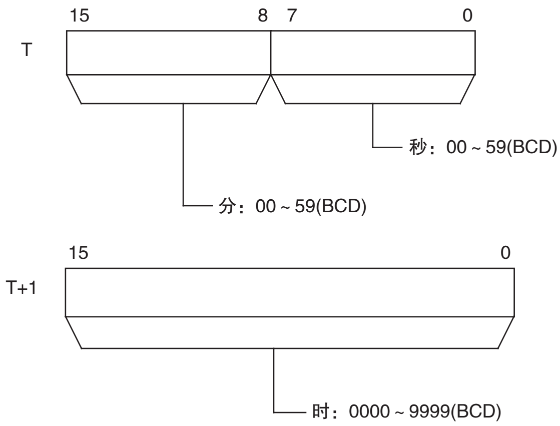


● CSUB

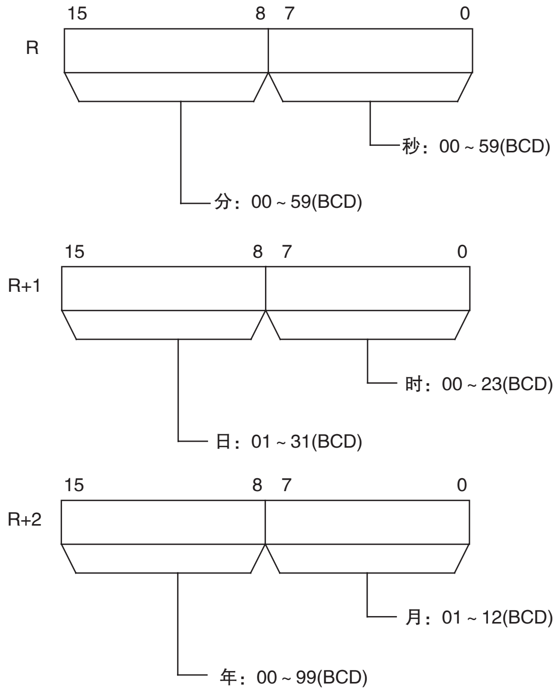
C ~ C+2：日历数据



T 和 T+1：时间数据



R ~ R+2：结果数据



● 区域类型

内容	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
C										---						
T	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	OK	---	---	---
R										---						

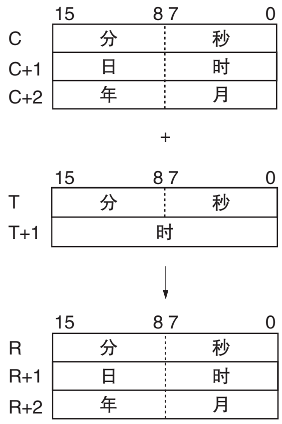
条件标志

名称	标记	操作
出错标志	P_ER	· C ～ C+2 中的日历数据不在指定范围内时为 ON。 · T 和 T+1 中的时间数据不在指定范围内时为 ON。 · 其它情况下 OFF。
等于标志	P_EQ	· 当 CSUB 指令的结果为 0 时 ON。 · 其它情况下 OFF。

功能

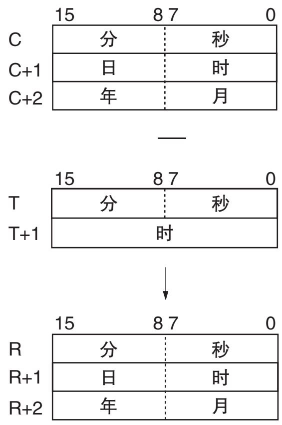
● CADD

CADD(730) 将日历数据 (字 C ～ C+2) 与时间数据 (字 T 和 T+1) 相加，并将日历数据结果输出到 R ～ R+2 中。



● CSUB

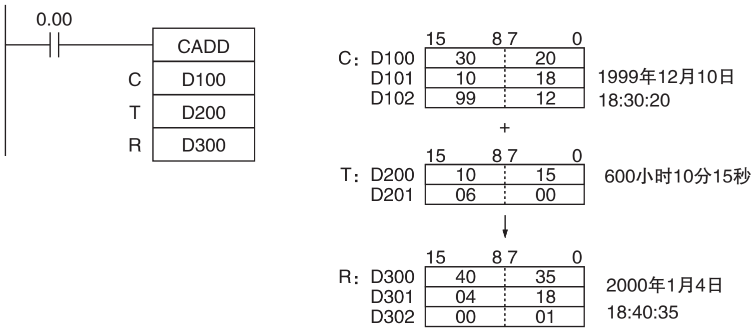
CSUB(731) 从日历数据 (字 C ～ C+2) 中减去时间数据 (字 T 和 T+1)，并将日历数据结果输出到 R ～ R+2 中。



程序举例

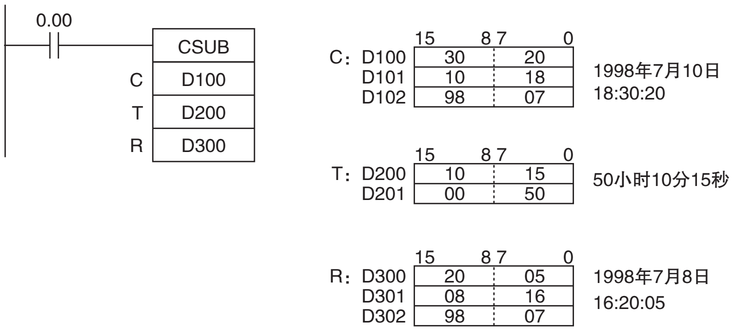
● CADD

下例中，当 CIO 0.00 变为 ON 时，D100 ～ D102(年、月、日、时、分、秒) 中的日历数据与 D200 和 D201(时、分、秒) 中的时间数据相加，并将结果输出到 D300 ～ D302。



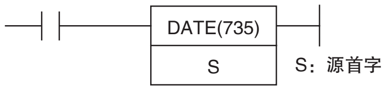
● CSUB

下例中，当 CIO 0.00 变为 ON 时，D100 ～ D102(年、月、日、时、分、秒) 中的日历数据减去 D200 和 D201(时、分、秒) 中的时间数据，并将结果输出到 D300 ～ D302。



DATE

指令	助记符	变化	功能代码	功能
时钟调整	DATE	@DATE	735	将内部时钟设定改为指定源字中的设定。

符号	DATE
	

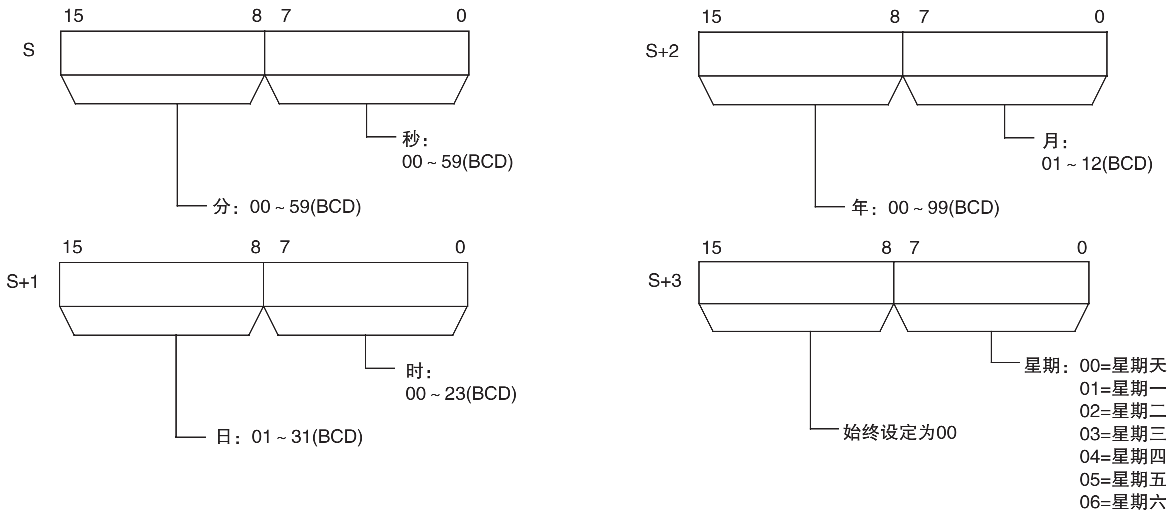
适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
S	源首字	LWORD	4

S ～ S+3：新时钟设定



注 S ～ S+3 必须位于同一个数据区。

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
S	OK	OK	OK	OK	OK	OK	OK	OK	OK	---	---	---	OK	---	---	---

条件标志

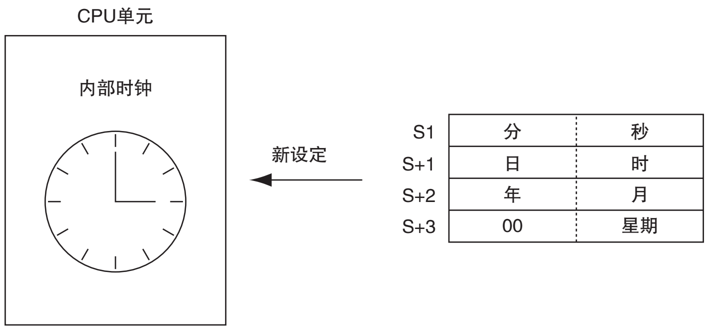
名称	标记	操作
出错标志	P_ER	· 当 S ～ S+3 中的新时钟设定不在指定范围内时 ON。 · 对 CP1E-E □□□□ – □执行 DATE 指令时 ON。 · 时钟初始化中执行 DATE 指令时 ON。 · 其它情况下 OFF。

辅助区中的相关字和位

名称	地址	操作
时钟数据	A351 ~ A354	A351.00 ~ A351.07：秒 (00 ~ 59)(BCD) A351.08 ~ A351.15：分 (00 ~ 59)(BCD) A352.00 ~ A352.07：时 (00 ~ 23)(BCD) A352.08 ~ A352.15：日 (01 ~ 31)(BCD) A353.00 ~ A353.07：月 (01 ~ 12)(BCD) A353.08 ~ A353.15：年 (00 ~ 99)(BCD) A354.00 ~ A354.07：星期 (00 ~ 06)(BCD) 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

功能

DATE(735) 根据 4 个源字中的时钟数据改变内部时钟设定。新的内部时钟设定会立即反映在日历 / 时钟区 (A351 ~ A354) 中。



提示

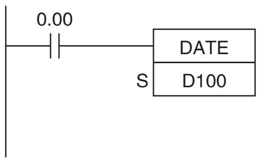
也可通过外围设备或 CLOCK WRITE FINS 命令 (0702) 改变内部时钟设定。

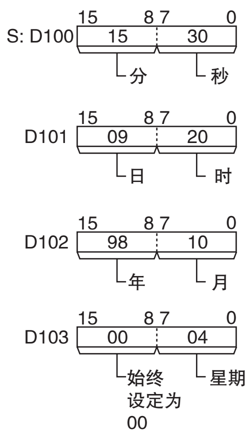
注意事项

- 即使内部时钟被设定为一个不存在的日期 (如 11 月 31 日)，也不会产生错误。
- 若对 E 型 CPU 单元执行该指令，则出错标志将置 ON 且指令无法执行。对于 E 型 CPU 单元，A351 ~ A354 始终为 “01-01-01 01:01:01 Sunday”形式。

程序举例

下例中，当 CIO 0.00 变为 ON 时，内部时钟将被设定为 1998 年 10 月 9 日，星期四，20:15:30。





故障诊断指令

FAL

指令	助记符	变化	功能代码	功能
故障报警	FAL	@FAL	006	产生或清除用户自定义的非致命错误。非致命错误不会使 PLC 停止运行。

	FAL	
符号	产生或清除用户自定义的非致命错误	产生非致命系统错误
	FAL(006) N　N：FAL号 S　S：信息首字或常数(0000 ~ FFFF)	FAL(006) N　N：FAL号(A529中的值) S　S：包含出错代码和错误详情的首字

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	FAL 号	仅限常数	1
S	包含错误代码和错误详情的信息首字或常数 / 首字	WORD	可变

● 产生或清除用户自定义的非致命错误。

注 操作数 N 的值必须与 A529(系统产生的 FAL/FALS 号) 的内容不同。

	产生非致命错误	清除所有非致命错误
N	1 ~ 511(这些 FAL 号与 FALS 号共用)	0
S	字地址： 产生带对应 FAL 号的非致命错误 编程设备将显示 S ~ S+7 中包含的 16 字符 ASCII 信息 #0000 ~ #FFFF： 产生带对应 FAL 号的非致命错误 (无信息)	#FFFF：清除所有非致命错误 #0001 ~ #01FF：清除带对应 FAL 号的非致命错误 其它：清除严重程度最高的非致命错误

● 产生非致命系统错误

注 操作数 N 的值必须与 A529(系统产生的 FAL/FALS 号) 的内容相同。

	产生非致命错误
N	1 ~ 511(这些 FAL 号与 FALS 号共用)
S	将会产生的错误代码 (参见下文的 “功能”)
S+1	将会产生的错误详情代码 (参见下文的 “功能”)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK				OK			

条件标志

名称	标记	操作
出错标志	ER	· 当 N 不在十进制 0 ～ 511 的指定范围内时为 ON。 · 当产生非致命系统错误但指定的错误代码或错误详情代码不正确时为 ON。 · 其它情况下 OFF。

辅助区中的相关字和位

● 仅用于用户自定义错误的辅助区字 / 标志

名称	地址	操作
FAL 错误标志	A402.15	当由 FAL(006) 产生错误时为 ON。
已执行 FAL 号标志	A360.01 ～ A391.15	由 FAL(006) 产生错误时，对应的标志将变为 ON。标志 A360.01 ～ A391.15 对应 FAL 号 1 ～ 511(十进制)。

● 仅用于系统错误的辅助区字 / 标志

名称	地址	操作
系统产生的 FAL/FALS 号	A529	由 FAL(006) 产生系统错误时，将使用一个虚拟的 FAL/FALS 号。请在该字内设置相同的虚拟 FAL/FALS 号 (十六进制数 0001 ～ 01FF，十进制数 1 ～ 511)。

● 可用于用户自定义错误和系统错误的辅助区字 / 标志

名称	地址	操作
出错日志区	A100 ~ A199	出错日志区中包含了错误代码和错误发生时间 / 日期 (最近 20 条错误)，其中包括 FAL(006) 产生的错误。
错误代码	A400	发生错误时，其对应的错误代码被保存在 A400 中。FAL 号 0001 ～ 01FF 对应的错误代码分别为 4101 ～ 42FF。 如果同时发生两个或两个以上的错误，严重程度最高的错误的代码将被保存在 A400 中。

无需编程设备清除非致命错误

● 清除用户自定义的非致命错误

若在 N 设定为 0 的情况下执行 FAL(006)，则可清除非致命错误。如下表所示，处理过程将根据 S 的值来确定。

S	处理过程
&1 ～ &511(0001 ～ 01FF Hex)	清除指定号的 FAL 错误。
FFFF Hex	清除所有的非致命错误 (包括系统错误)。
0200 ～ FFFE Hex 或指定字	产生的严重程度最高的非致命错误 (甚至是非致命系统错误)。当产生多个 FAL 错误时，将清除 FAL 号最靠前的 FAL 错误。

● 清除非致命系统错误

有两种方法可用于清除由 FAL(006) 产生的非致命系统错误。

· 将 PLC 断电后再上电。

· 当 PLC 保持上电状态时，必须清除指定错误 (假设已经发生了指定的错误)。

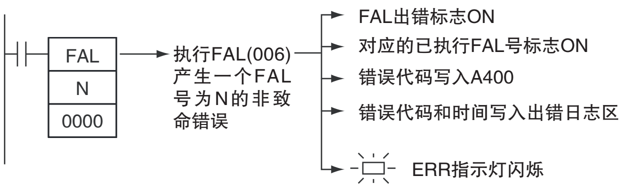
功能

● 产生用户自定义的非致命系统错误

下表所示为 FAL(006) 对应的错误代码和 FAL 错误标志。

FAL 号	十进制 1 ～ 511
FAL 错误代码	4101 ～ 42FF
已执行 FAL 号标志	A360.01 ～ A391.15

当执行 FAL(006) 且 N 设定了一个与 A529(系统产生的 FAL/FALS 号) 内容不相同的 FAL 号 (&1 ～ &511)，将由该 FAL 号产生一个非致命错误，并执行以下过程：



1. FAL 错误标志 (A402.15) 将变为 ON。(PLC 将继续运行。)
2. 与 FAL 号对应的已执行 FAL 号标志将变为 ON。标志 A360.01 ～ A391.15 对应 FAL 号 0001 ～ 01FF(1 ～ 511)。
3. 错误代码将被写入 A400。错误代码 4101 ～ 42FF 对应 FAL 号 0001 ～ 01FF(1 ～ 511)。
4. 错误代码和错误发生时间将被写入出错日志区 (A100 ～ A199)。

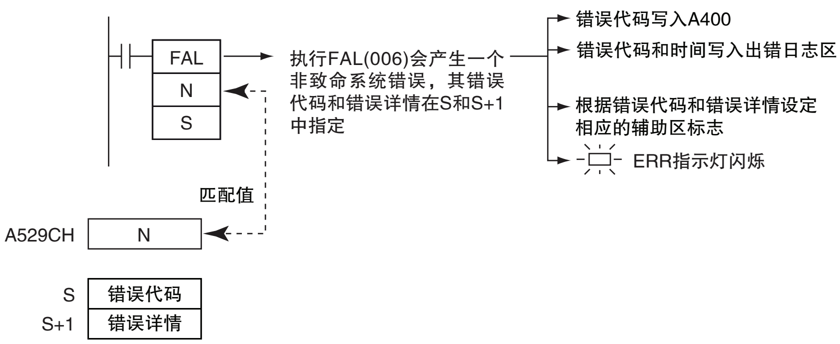
注 如果在 PLC 设置中选择了 “Don't register FAL to error log”选项，出错记录将不会被写入出错日志区。

5. CPU 单元上的 ERR 指示灯将会闪烁。
6. 如果在 S 中指定了一个字地址，则从 S 开始的信息将被注册 (显示在编程设备上)。

注 执行 FAL(006) 指令时，如果同时发生致命错误或严重程度更高的非致命错误，则后者的错误代码会被写入 A400。

● 产生非致命系统错误

当执行 FAL(006) 且 N 设定了一个与 A529(系统产生的 FAL/FALS 号) 内容相同的 FAL 号 (&1 ～ &511)，将产生一个非致命错误，其错误代码和错误详情在 S 和 S+1 中指定，同时执行以下过程：



1. 指定的错误代码将被写入 A400。

2. 错误代码和错误发生时间将被写入出错日志区 (A100 ~ A199)。

3. 根据错误代码和错误详情设定相应的辅助区标志。

4. CPU 单元上的 ERR 指示灯将会闪烁且 PLC 继续运行。

注 1 调试程序时，FAL(006) 可产生系统的非致命错误。例如，可特意产生一个系统错误以检查错误信息是否能正确地显示在可编程终端 (PT) 等接口设备上。

2 A529(系统产生的 FAL/FLAS 号) 中的值是由系统特意产生一个非致命错误时使用的一个虚拟 FAL 号 (FAL、FALS 号共用)，由于该号是一个虚拟 FAL 号，因此不会改变已执行 FAL 号标志 (A360.01 ~ A391.15) 或错误代码的状态。
若需要产生两个或两个以上的系统错误 (致命和 / 或非致命错误)，可在 A529 和 N 中的值相同及 S 和 S+1 中的值不同的情况下多次执行 FAL/FALS 指令，从而产生不同的错误。

3 如果在执行 FAL(006) 指令的同时发生了一个更严重的错误 (系统产生的致命错误或 FALS(007))，其错误代码会被写入 A400。

4 若要清除 FAL(006) 产生的系统错误，请将 PLC 断电后再上电。此时，PLC 可保持上电状态。但如果确实发生了指定错误，则需通过相同的处理方式来清除错误。
请参阅 *CP1E CPU 单元硬件操作手册*或 *CP1E CPU 单元软件操作手册*。

下表所示为如何在 S 和 S+1 中指定错误代码和错误详情。

CP1E

错误名称	S	S+1
PLC 设置错误	009B Hex	PLC 设置错误位置 0000 ~ FFFF Hex
内置模拟量错误	008A Hex	---(不定)
选件板错误	00D1 Hex	选件板插槽号 0001 Hex
电池错误	00F7 Hex	---(不定)

CP2E

错误名称	S	S+1
备份存储器错误	00F1 Hex	---
PLC 设置错误	009B Hex	PLC 设置错误位置 0000 ~ FFFF Hex
电池错误	007F Hex	0000 Hex
选件板错误 (仅CP2E N□□型CPU单元)	00D1 Hex	选件板插槽号 0001 Hex
	00D2 Hex	选件板插槽号 0002 Hex
Ethernet 设定表错误 (仅CP2E N□□型CPU单元)	021A Hex	---

错误名称	S	S+1
FINS/TCP 连接设定错误 (仅CP2E N□□型CPU单元)	03C0 Hex	最左边字节：FINS/TCP 连接 No. 01：FINS/TCP 连接 No.1 02：FINS/TCP 连接 No.2 03：FINS/TCP 连接 No.3 最右边字节：错误原因 01：自动分配 FINS 结点地址错误 02：目标 IP 地址错误 03：目标 FINS/TCP 端口编号错误
Ethernet 服务器设定错误 (仅CP2E N□□型CPU单元)	03C1 Hex	最左边字节：服务器类型 00：DNS 服务器 01：SNTP 服务器 最右边字节：错误原因 01：IP 地址错误 02：主机名称异常 03：端口编号错误 04：其他设定错误
Ethernet 服务器连接错误 (仅CP2E N□□型CPU单元)	03C4 Hex	最左边字节：服务器类型 00：DNS 服务器 01：SNTP 服务器 最右边字节：错误原因 01：无指定主机 02：指定主机无服务 03：超时错误 04：主机侧关闭 05：因账户信息不匹配而无法连接 06：主机名称解决错误 07：发送错误 08：接收错误 09：其他错误

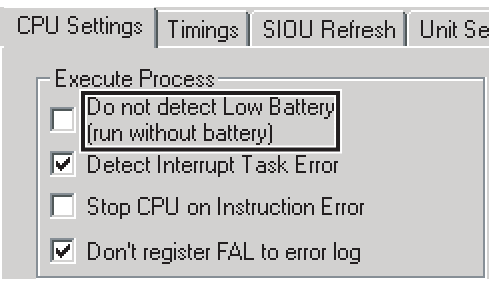
● 禁止将用户自定义错误记录到出错日志

一般当 FAL(006) 产生用户自定义错误时，错误代码和错误发生时间将被写入出错日志区 (A100 ~ A199)。可对 PLC 设置进行设定，防止 FAL(006) 产生的用户自定义错误被记录到出错日志中。

注 即使错误没有被记录到出错日志中，FAL 错误标志 (402.15) 和对应的已执行 FAL 号标志 (A360.01 ~ A391.15) 仍将变为 ON，且错误代码被写入 A400。

若仅需记录系统产生的错误，则可禁止将用户自定义的 FAL(006) 错误记录到出错日志中。例如，在调试过程中，如果 FAL(006) 指令用于多个场合，且错误日志中充满了用户自定义的 FAL(006) 错误，可使用该功能进行处理。

・以下截图所示为 CX-Programmer 中的 PLC 设置设定。



注 即使 PLC 设置中的字 129 的位 15 设定为 1(请勿将 FAL 错误记录至出错日志中)，也将记录下列错误：

- FALS(007) 产生的致命错误
- 系统产生的非致命错误
- 系统产生的致命错误
- 通过 FAL(006) 由系统特意产生的非致命错误
- 通过 FALS(007) 由系统特意产生的致命错误

● 显示用户自定义的非致命错误信息

- 如果 S 是一个字地址且其中已经保存了 ASCII 信息，当执行 FAL(006) 时，外围设备将显示该信息。(如果无需显示信息，可将 S 设定为常数。)
- 当执行 FAL(006) 时，从 S 处开始的信息将被记录。信息一旦被记录后，将会加以显示。
- S ～ S+7 中最多可以保存长达 16 字符的 ASCII 信息。每个字中最左边的字节将被优先显示。
- 信息的结束代码为空字符 (00 Hex)。
- 如果省略了空字符，S ～ S+7 中的 16 个字符都将被显示。
- 如果执行 FAL(006) 之后，包含信息的字的内容发生了改变，信息也将进行相应的改变。

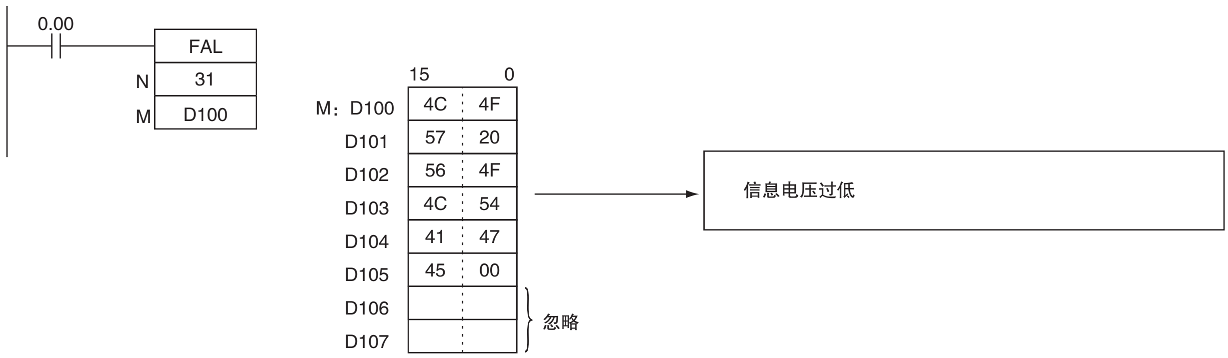
程序举例

● 产生非致命错误

下例中，当CIO 0.00为ON时，FAL(006)将会产生一个FAL号为31的非致命错误，同时执行以下过程。

1. FAL 错误标志 (A402.15) 将变为 ON。
2. 相应的已执行 FAL 号标志 (A361.15) 将变为 ON。
3. 相应的错误代码 (411F) 被写入 A400。
4. 错误代码和错误发生时间 / 日期被写入出错日志区 (A100 ～ A199)。
5. CPU 单元上的 ERR 指示灯将会闪烁。
6. 外围设备将显示 D100 ～ D107 中的 ASCII 信息。

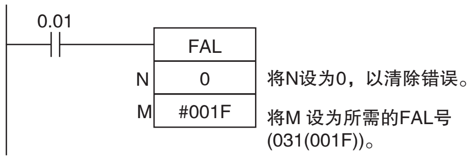
注 (如果无需显示信息，可将 S 设定为常数。)



注 如果同时发生两个或两个以上的错误，严重程度最高的错误的代码将被保存在 A400 中。

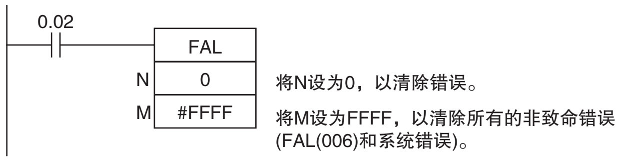
● 清除特定的非致命错误

下例中，当 CIO 0.01 为 ON 时，FAL(006) 将清除 FAL 号为 31 的非致命错误，且将对应的已执行 FAL 号标志 (A361.15) 和 FAL 错误标志 (A402.15) 置 OFF。



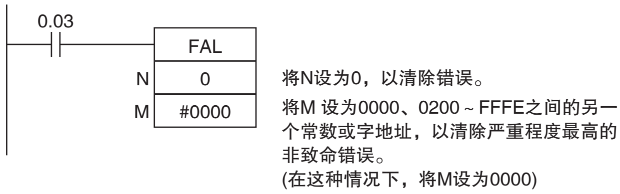
● 清除所有非致命错误

下例中，当 CIO 0.02 变为 ON 时，FAL(006) 将清除所有非致命错误，且将对应的已执行 FAL 号标志 (A360.01 ～ A391.15) 和 FAL 错误标志 (A402.15) 置 OFF。



● 清除严重程度最高的非致命错误

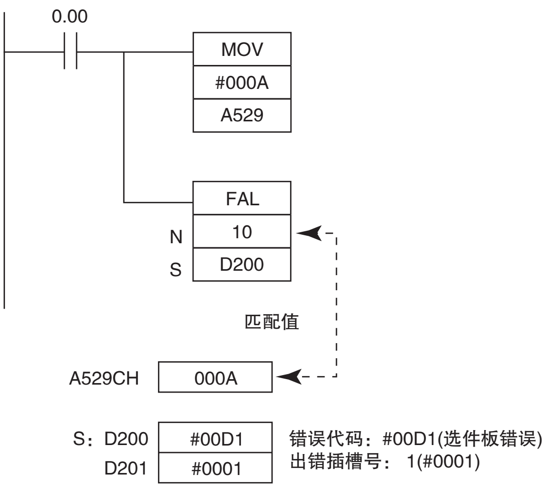
下例中，当 CIO 0.03 变为 ON 时，FAL(006) 将清除严重程度最高的非致命错误，并重设 A400 中的错误代码。如果被清除的错误最初是由 FAL(006) 产生的，对应的已执行 FAL 号标志和 FAL 错误标志 (A402.15) 将被置 OFF。



● 产生非致命系统错误

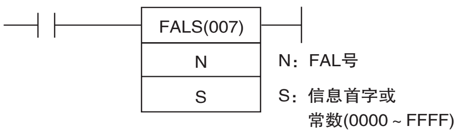
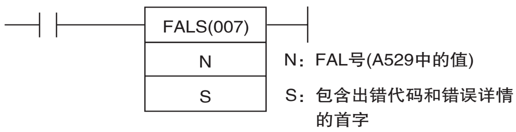
下例中，当 CIO 0.00 为 ON 时，FAL(006) 将产生选件板错误。在这种情况下，将使用虚拟 FAL 号 10，并将相应的值 (000A Hex) 保存在 A529 中。

1. 若属于严重程度最高的错误，其对应的错误代码 (00D1) 将被写入 A400。
2. 错误代码和错误发生时间 / 日期被写入出错日志区 (A100 ～ A199)。
3. 选件板出错标志 (A315.13) 将被置 ON。
4. CPU 单元的 ERR 指示灯将会闪烁。
5. 发生选件板错误。



FALS

指令	助记符	变化	功能代码	功能
严重故障报警	FALS	---	007	产生用户自定义的致命错误。致命错误将使 PLC 停止运行。

	FALS	
符号	产生用户自定义的致命错误 	产生致命系统错误 

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
N	FAL 号	仅限常数	1
S	包含错误代码和错误详情的信息首字或常数 / 首字	WORD	可变

● 产生用户自定义的致命错误

注 操作数 N 的值必须与 A529(系统产生的 FAL/FALS 号) 的内容不同。

	产生非致命错误
N	1 ～ 511(这些 FALS 号与 FAL 号共用)
S	指定 8 个字中的首字，这 8 个字中包含显示在编程设备上的 ASCII 信息。 如果无需显示信息，可指定一个常数 (0000 ～ FFFF)。

● 由系统产生的致命错误

下表所示为操作数的功能。

注 操作数 N 的值必须与 A529(系统产生的 FAL/FALS 号) 的内容相同。

	产生非致命错误
N	1 ～ 511(这些 FALS 号与 FAL 号共用)
S	将会产生的错误代码 (参见下文的 “说明”)
S+1	将会产生的错误详情代码 (参见下文的 “说明”)

● 区域类型

区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
N	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---
S	OK	OK	OK	OK	OK	OK	OK	OK	OK				OK			

条件标志

名称	标记	操作
出错标志	P_ER	· 当 N 不在 0001 ～ 01FF(十进制数 0 ～ 511) 的指定范围内时为 ON。 · 当产生致命系统错误，但指定的错误代码或错误详情代码不正确时为 ON。 · 其它情况下 OFF。

辅助区中的相关字和位

● 仅用于用户自定义错误的辅助区字 / 标志

名称	地址	操作
FALS 错误标志	A401.06	当由 FALS(007) 产生错误时为 ON。

● 仅用于系统错误的辅助区字 / 标志

名称	地址	操作
系统产生的 FAL/FALS 号	A529	由 FALS(007) 产生系统错误时，将使用一个虚拟的 FAL/FALS 号。请在该字内设置相同的虚拟 FAL/FALS 号 (十六进制数 0001 ～ 01FF，十进制数 1 ～ 511)。

● 可用于用户自定义错误和系统错误的辅助区字 / 标志

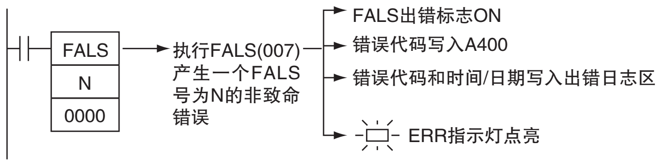
名称	地址	操作
出错日志区	A100 ~ A199	出错日志区中包含了错误代码和错误发生时间 / 日期 (最近 20 条错误)，其中包括 FALS(007) 产生的错误。
错误代码	A400	发生错误时，其对应的错误代码被保存在 A400 中。FALS 号 0001 ～ 01FF(十进制数 1 ～ 511) 对应的错误代码分别为 C101 ～ C2FF。 注 如果同时发生两个或两个以上的错误，严重程度最高的错误的代码将被保存在 A400 中。

功能

● 产生用户自定义的错误

FALS 号	1 ～ 511
FALS 错误代码	C101 ～ C2FF

当执行 FALS(007) 且 N 设定了一个与 A529(系统产生的 FAL/FALS 号) 内容不相同的 FALS 号 (1 ～ 511)，将由该 FALS 号产生一个致命错误，并执行以下过程：



1. FALS 错误标志 (A401.06) 将变为 ON。(PLC 将停止运行。)

2. 错误代码将被写入 A400。错误代码 C101 ～ C2FF 对应 FALS 号 0001 ～ 01FF(1 ～ 511)。

注 如果发生了严重程度高于 FALS(007) 指令的错误 (具备更高的错误代码等级)，其错误代码将被写入 A400。

3. 错误代码和错误发生时间 / 日期被写入出错日志区 (A100 ～ A199)。

4. CPU 单元上的 ERR 指示灯将会点亮。

5. 如果在 S 中指定了一个字地址，则从 S 开始的 ASCII 信息将被注册 (显示在外围设备上)。

注 1 如果同时发生由该指令注册的错误和严重程度更高的错误 (包括致命系统错误)，则后者对应的错误代码将被写入 A400。

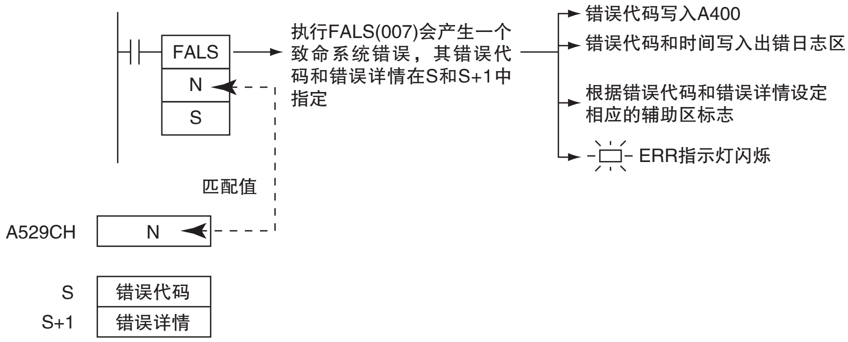
2 信息的结束代码为空字符 (00 Hex)。如果省略了空字符，S ～ S+7 中的 16 个字符都将被显示。

3 N 必须介于 0001 和 01FF 之间。如果 N 在指定范围以外，将会导致错误发生，并使错误标志变为 ON。

4 如果注册了用户自定义的致命错误，输出单元的 I/O 存储器状态和输出状态如下表所示。

		I/O 存储器	输出单元的输出状态
IOM 保持位 (A500.12)	ON	保持	OFF
	OFF	保持	OFF

● 产生非致命系统错误



当执行 FALS(007) 且 N 设定了一个与 A529(系统产生的 FAL/FALS 号) 内容相同的 FAL 号 (1 ～ 511)，将产生一个致命错误，其错误代码和错误详情在 S 和 S+1 中指定，同时执行以下过程：

1. 指定的错误代码将被写入 A400。
2. 错误代码和错误发生时间将被写入出错日志区 (A100 ～ A199)。
3. 根据错误代码和错误详情设定相应的辅助区标志。
4. CPU 单元上的 ERR 指示灯将会点亮且 PLC 停止运行。

注 1 A529(系统产生的 FAL/FLAS 号) 中的值是由系统特意产生一个非致命错误时使用的一个虚拟 FAL 号 (FAL、FALS 号共用)，由于该号是一个虚拟 FAL 号，因此不会改变错误代码的状态。若需要产生两个或两个以上的系统错误，可在 A529 和 N 中的值相同及 S 和 S+1 中的值不同的情况下多次执行 FAL/FALS 指令，从而产生不同的错误。

2 如果在执行 FALS(007) 指令的同时发生了一个严重程度更高的错误 (系统产生的致命错误或其它 FALS(007) 错误)，其错误代码会被写入 A400。

3 若要清除 FALS(007) 产生的系统错误，请将 PLC 断电后再上电。此时，PLC 可保持上电状态。但如果确实发生了指定错误，则需通过相同的处理方式来清除错误。有关详情，请参阅 *CP1E CPU 单元硬件操作手册或 CP1E CPU 单元软件操作手册*。

4 下表显示了 FALS(007) 产生系统致命错误后，I/O 保持位是如何影响 I/O 存储器状态和输出单元的输出状态的。

		I/O 存储器状态	输出单元的输出状态
IOM 保持位 (A500.12)	ON	保持	OFF
	OFF	清除	OFF

下表所示为如何在 S 和 S+1 中指定错误代码和错误详情。

CP1E

错误名称	S	S+1
	错误代码	错误详情
存储器错误	80F1 Hex	位 00 ～ 09：存储器错误位置 位 00：用户程序 位 01：I/O 存储器 位 04：PLC 设置 位 2、3、5 ～ 15：无效
I/O 总线错误	80CA Hex	CP1W 扩展 I/O 单元、扩展单元 #0A0A Hex
I/O 点数过多错误	80E1 Hex	位 13 ～ 15：错误代码 位 00 ～ 12：详情 ・CP1W 扩展 I/O 单元通道数过多。 位 13 ～ 15：001 位 00 ～ 12：全为 0
程序错误	80F0 Hex	位 08 ～ 15：错误原因 位 15：UM 溢出错误 位 14：非法指令错误 位 13：微分溢出错误 位 12：任务错误 位 11：无 END 指令错误 位 10：非法访问错误 位 09：间接 DM BCD 错误 位 08：指令错误 位 00 ～ 07：无效
循环时间超时错误	809F Hex	#0000 Hex

CP2E

错误名称	S	S+1
	错误代码	错误详情
存储器错误	80F1 Hex	位 00 ～ 09：存储器错误位置 (如下) 位 00：用户程序 位 04：PLC 设置 位 07：路由表 (仅 CP2E N □□型 CPU 单元) 位 11：IP 地址表 (仅 CP2E N □□型 CPU 单元) 位 12：IP 路由表 (仅 CP2E N □□型 CPU 单元) 位 14：I/O 存储器
I/O 总线错误	80CA Hex	#0A0A Hex
I/O 点数过多错误	80E1 Hex	#4000 Hex
程序错误	80F0 Hex	位 08 ～ 15：错误原因 位 15：UM 溢出错误 位 14：非法指令错误 位 13：微分溢出错误 位 12：任务错误 位 11：无 END 指令错误 位 10：非法访问错误 位 09：间接 DM BCD 错误 位 08：指令错误 位 00 ～ 07：无效
循环时间超时错误	809F Hex	#0000 Hex
内置 Ethernet 停止错误 (仅CP2E N□□型CPU单元)	80F0 Hex	--- (不定)

● 显示用户自定义的致命错误信息

- 如果 S 为字地址，则在执行 FALS(007) 时，将由编程设备显示从 S 开始的 ASCII 信息。(如果无需显示信息，可将 S 设定为常数。)
- 当执行 FALS(007) 时，从 S 处开始的信息将被记录。信息一旦被记录后，将会加以显示。
- S ～ S+7 中最多可以保存长达 16 字符的 ASCII 信息。每个字中最左边的字节将被优先显示。
- 信息的结束代码为空字符 (00 Hex)。
- 如果省略了空字符，S ～ S+7 中的 16 个字符都将被显示。
- 如果执行 FALS(007) 之后，包含信息的字的内容发生了改变，信息也将进行相应的改变。

● 清除 FALS(007) 致命系统错误

有两种方法可用于清除由 FALS(007) 产生的致命系统错误。

1. 将 PLC 断电后再上电。
2. 当 PLC 保持上电状态时，必须清除指定错误 (假设已经发生了指定的错误)。

● 清除用户自定义的 FALS(007) 致命错误

若要清除 FALS(007) 产生的错误，应首先排除错误原因，然后通过编程设备清除错误，或者将 PLC 断电后再上电。

注意事项

在已登记致命系统错误的情况下，若 IOM 保持位为 OFF，则 I/O 存储器将被清除。

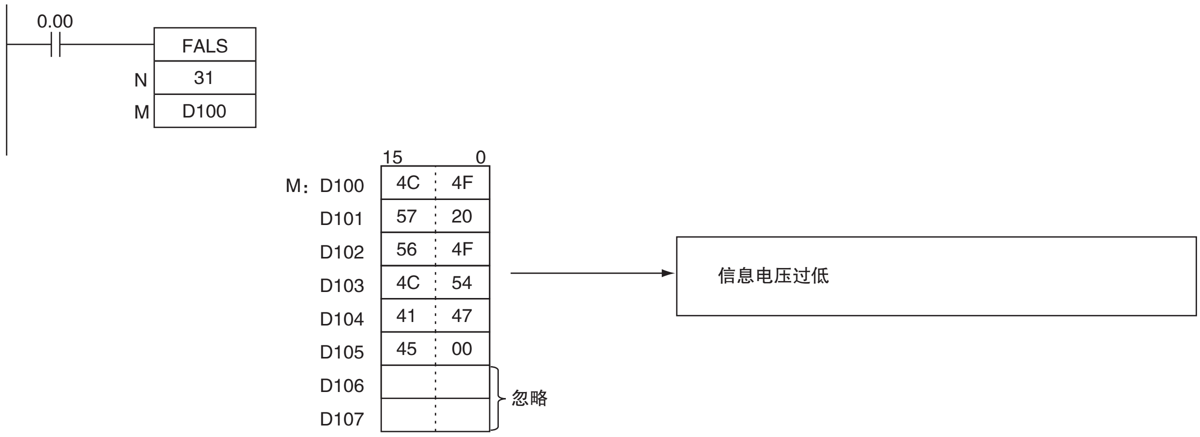
程序举例

● 产生用户自定义的错误

下例中，当 CIO 0.00 为 ON 时，FALS(007) 将会产生一个 FAL 号为 31 的致命错误，同时执行以下过程。

1. FALS 错误标志 (A401.06) 将变为 ON。
2. 相应的错误代码 (C11F) 被写入 A400。
3. 错误代码和错误发生时间 / 日期被写入出错日志区 (A100 ～ A199)。
4. CPU 单元上的 ERR 指示灯将会点亮。
5. 外围设备将显示 D100 ～ D107 中的 ASCII 信息。

注 (如果无需显示信息，可将 S 设定为常数。)

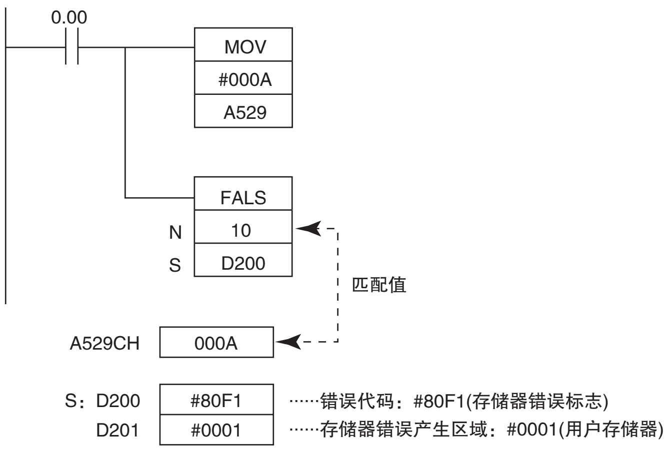


注 A400 中将会包含所有严重程度最高的错误对应的错误代码，其中包括非致命和致命系统错误以及由 FAL(006) 和 FAL(007) 产生的错误。

● 产生非致命系统错误

下例中，当 CIO 0.00 为 ON 时，FALS(007) 将产生存储器错误 (用户程序错误)。在这种情况下，将使用虚拟 FAL 号 10，并将相应的值 (80F1 Hex) 保存在 A529 中。

1. 若属于严重程度最高的错误，其对应的错误代码 (80F1) 将被写入 A400。
2. 错误代码和错误发生时间 / 日期被写入出错日志区 (A100 ～ A199)。
3. 存储器错误标志 (A401.15) 将变为 ON。
4. CPU 单元的 ERR 指示灯将会点亮，且 PLC 停止运行。
5. 发生存储器错误。



其它指令

STC/CLC

指令	助记符	变化	功能代码	功能
置进位	STC	@STC	040	置进位标志 (CY)。
清除进位	CLC	@CLC	041	使进位标志 (CY) 变为 OFF。

符号	STC	CLC
	STC(040)	CLC(041)

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

条件标志

操作数	描述	数据类型	
		STC	CLC
进位标志	P_CY	ON	OFF

功能

● STC

当执行条件为 ON 时，ST(040) 可使进位标志 (CY) 变为 ON。但通过执行能影响进位标记的顺序指令，也可改变进位标记的 ON/OFF 状态。

ROL(027) 和 ROR(028) 在循环移位操作中使用进位标记。

● CLC

当执行条件为 ON 时，CLC(040) 可使进位标志 (CY) 变为 OFF。但通过执行能影响进位标记的顺序指令，也可改变进位标记的 ON/OFF 状态。

+C(402)、+CL(403)、+BC(406)、+BCL(407)、−C(412)、−CL(413)、−BC(416) 和 −BCL(417) 在加操作中使用进位标记。为防止其它先前的指令的影响，使用这些指令前先要使用 CLC(041)。

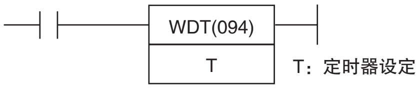
ROL(027) 和 ROR(028) 在循环移位操作中使用进位标记。

提示

+(400)、+L(401)、+B(404)、+BL(405)、−(410)、−L(411)、−B(414) 和 −BL(415) 指令在加减操作中不包括进位标记。通常情况下，执行加 / 减操作时需使用这些指令。

WDT

指令	助记符	变化	功能代码	功能
延长最大循环时间	WDT	@WDT	094	延长最大循环时间，但仅对于执行该指令的循环有效。当特殊处理暂时需要一个较长的循环时间时，WDT(094) 能防止循环时间过长而导致错误发生。

符号	WDT
	

适用程序区

区域	功能块定义内	步程序区	子程序区	中断任务程序区
能否使用	OK	OK	OK	OK

操作数

操作数	描述	数据类型	大小
T	定时器设定	仅限常数	1

T：定时器设定

指定十六进制数 0000 ~ 0064 或十进制数 &0000 ~ &0100 之间的看门狗定时器设定。

● 区域类型

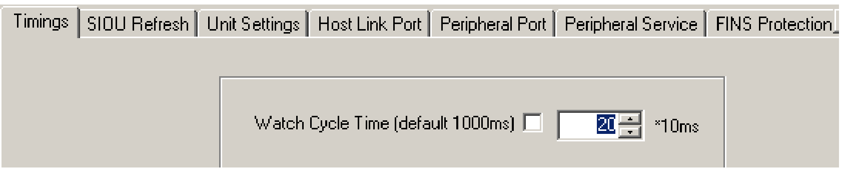
区域	字地址							DM 间接地址		常数	寄存器			CF	脉冲位	TR 位
	CIO	WR	HR	AR	T	C	DM	@DM	*DM		DR	IR 直接	IR 间接			
T	---	---	---	---	---	---	---	---	---	OK	---	---	---	---	---	---

条件标志

操作数	描述	数据类型
出错标志	P_ER	· 当看门狗定时器设定值超过 1 秒时为 ON。 · 其它情况下 OFF。

相关的 PLC 设置设定

● CX-Programmer 设定



● PLC 设置中的设定

名称	功能	各种设定
监视循环时间	如果循环时间超过最大设定，将记录循环时间超长错误(致命错误)。	0：默认设定 (1,000ms) 1：用户时间设定
	设定最大循环时间。 (仅当第一项设置设定为 1 时，该设置才有效)	0001 ～ 0FA0 (10 ～ 1,000ms，以 10ms 为单位)

注 · 最大循环时间的默认值为 1,000ms，可在 10 ～ 1,000ms 范围内以 10ms 为单位任意设定。
· WDT(094) 可在一个循环内使用多次。在这种情况下，延长的循环时间可以进行累加，但总和不得超过 1,000ms。如果循环时间已经延长至 1,000ms，将无法再次执行 WDT(094)。

辅助区中的相关字和位

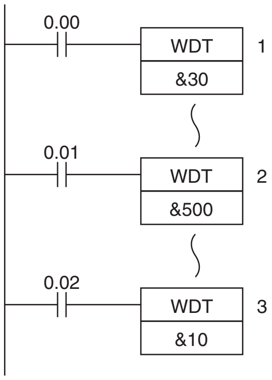
名称	地址	操作
循环时间超长标志	A401.08	当前循环时间超过 PLC 设置中设定的最大循环时间 (监视循环时间) 时为 ON。这属于致命错误，可导致程序停止执行。
最大循环时间	A262 和 A263	这些字中包含 32 位二进制的最大循环时间。这些数值在每次循环时都会刷新。
当前循环时间	A264 和 A265	这些字中包含 32 位二进制的当前循环时间。这些数值在每次循环时都会刷新。

功能

WDT(094) 可延长最大循环时间，但仅对于执行该指令的循环有效。PLC 设置中的看门狗定时器设定可延长 T × 10ms(0 ～ 1,000ms)。

当循环时间由于数据处理量的暂时增加而延长时，该指令可防止出现循环时间错误。

程序举例



WDT(094) 的运行

本例中，看门狗定时器设定被设为 500ms。

· 当 CIO 0.00 变为 ON 时，第一条 WDT(094) 指令可将循环时间延长 300ms(30 × 10ms)。因此，该点处的总循环时间为 800ms。

· 当 CIO 0.01 变为 ON 时，第二条 WDT(094) 指令尝试将循环时间再延长 500ms。由于总循环时间 (1,300ms) 超过了 1,000ms 的上限值，超出的 300ms 将被忽略。因此，第二条 WDT(094) 指令实际将总循环时间延长了 200ms。

· 当 CIO 0.02 变为 ON 时，第三条 WDT(094) 指令尝试将循环时间再延长 10ms。由于总循环时间已经达到 1,000ms 的上限值，第三条 WDT(094) 指令将不被执行。

3

指令执行时间和步数

本章节介绍了 CP1E/CP2E CPU 单元支持的所有指令的执行时间。

3-1 指令执行时间和步数

下表列出了 CPU 单元所支持的所有指令的执行时间。

一个用户程序中指令的总执行时间为计算循环时间时的程序执行的处理时间 (见注)。

注 将可在循环任务中和满足中断条件的中断任务中执行的任务分配给用户程序。

大多数指令的执行时间因所使用的 CPU 单元和执行指令时的条件而异。

当执行条件为 OFF 时，执行时间也会有所不同。

下表还在长度 (步数) 栏中列出了每条指令的长度。各指令在用户程序区中所需的步数取决于指令和指令中所使用的操作数。

程序中的步数不等于指令的条数。

注 1 支持大多数指令的微分形式 (用↑，↓，@ 和 % 表示)。指定为微分形式将会使执行时间增加下表中列出的量。

(单位：μs)

符号	CP1E CPU 单元	CP2E CPU 单元
↑或↓	+4.0	+0.6
@ 或 %	+2.5	+0.4

2 未执行指令时，请采用下列时间作为参考依据。

(单位：μs)

CP1E CPU 单元	CP2E CPU 单元
1.4	0.19

顺序输入指令

指令	助记符	功能代码	长度（步数）	执行时间 (µs) CP1E	执行时间 (µs) CP2E	条件
载入	LD	---	1	1.19	0.23	---
	!LD	---	2	10.26	2.57	---
载入非	LD NOT	---	1	1.19	0.23	---
	!LD NOT	---	2	10.26	2.57	---
与	AND	---	1	1.19	0.23	---
	!AND	---	2	10.26	2.55	---
与非	AND NOT	---	1	1.19	0.23	---
	!AND NOT	---	2	10.26	2.57	---
或	OR	---	1	1.29	0.26	---
	!OR	---	2	10.36	2.62	---
或非	OR NOT	---	1	1.29	0.26	---
	!OR NOT	---	2	10.36	2.62	---
逻辑块与	AND LD	---	1	0.60	0.16	---
逻辑块或	OR LD	---	1	0.60	0.16	---
非	NOT	520	1	0.80	0.16	---
条件 ON	UP	521	3	4.92	0.97	---
条件 OFF	DOWN	522	4	5.69	1.14	---
LD 型・位测试	LD TST	350	3	---	1.85	---
LD 型・非位测试	LD TSTN	351	3	---	1.85	---
AND 型・位测试	AND TST	350	3	---	1.85	---
AND 型・非位测试	AND TSTN	351	3	---	1.85	---
OR 型・位测试	OR TST	350	3	---	1.85	---
OR 型・非位测试	OR TSTN	351	3	---	1.85	---

顺序输出指令

指令	助记符	功能代码	长度（步数）	执行时间 (µs) CP1E	执行时间 (µs) CP2E	条件
输出	OUT	---	1	1.61	0.30	---
	!OUT	---	2	38.06	11.39	---
反相输出	OUT NOT	---	1	1.61	0.34	---
	!OUT NOT	---	2	38.06	11.46	---
保持	KEEP	011	1	4.72	0.34	---
上升沿微分	DIFU	013	2	4.12	0.94	---
下降沿微分	DIFD	014	2	4.19	0.93	---
置位	SET	---	1	2.69	0.41	---
	!SET	---	2	39.12	6.22	---
复位	RSET	---	1	2.69	0.41	指定了字
	!RSET	---	2	39.12	6.23	---
多个位置位	SETA	530	4	17.60	5.53	对 1 个位置位
				253.5	49.37	对 1,000 个位置位
多个位复位	RSTA	531	4	17.60	5.55	对 1 个位复位
				249.5	48.31	对 1,000 个位复位
单个位置位	SETB	532	2	16.60	3.98	---
	!SETB		3	54.60	9.67	---
单个位输出	RSTB	534	2	16.60	3.96	---
	!RSTB		3	54.60	9.67	---

顺序控制指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
结束	END	001	1	4.6	1.27	---
空操作	NOP	000	1	1.2	0.18	---
互锁	IL	002	1	4.3	0.90	---
互锁清除	ILC	003	1	4.3	0.90	---
多路互锁微分保持	MILH	517	3	19.4	4.23	处于互锁期间
				19.4	4.23	未处于互锁期间且互锁未被置位
				21.5	5.02	未处于互锁期间但互锁被置位
多路互锁微分释放	MILR	518	3	19.4	4.21	处于互锁期间
				19.4	4.21	未处于互锁期间且互锁未被置位
				21.5	5.02	未处于互锁期间但互锁被置位
多路互锁清除	MILC	519	2	8.9	2.32	互锁未被清除
				8.9	2.32	互锁已被清除
跳转	JMP	004	2	6.1	2.78	---
跳转结束	JME	005	2	6.2	1.48	---
条件跳转	CJP	510	2	10.1	2.93	满足跳转条件时
FOR 循环	FOR	512	2	9.7	2.85	指定一个常数
循环中断	BREAK	514	1	4.1	1.08	---
NEXT 循环	NEXT	513	1	5.8	1.30	继续循环时
				5.7	1.83	结束循环时

定时器和计数器指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
定时器	TIM	---	3	11.6	2.49	---
	TIMX	550			2.48	继续循环时
计数器	CNT	---	3	11.5	3.28	---
	CNTX	546			3.29	继续循环时
高速计数器	TIMH	015	3	11.6	2.52	---
	TIMHX	551			2.37	继续循环时
1ms 定时器	TMHH	540	3	10.8	2.43	---
	TMHHX	552			2.43	---
累加定时器	TTIM	087	3	22.7	6.42	---
				17.4	4.91	复位时
				15.0	5.40	互锁时
	TTIMX	555	3	22.2	6.26	---
				17.4	4.90	复位时
				15.2	5.40	互锁时
长定时器	TIML	542	4	24.3	5.80	---
				20.4	5.33	互锁时
	TIMLX	553	4	24.5	5.78	---
				22.2	5.20	互锁时
可逆计数器	CNTR	012	3	26.2	7.40	---
	CNTRX	548		25.4	7.30	---
复位定时器 / 计数器	CNR	545	3	19.0	4.45	复位 1 个字时
				659.0	187.23	复位 256 个字时
	CNRX	547	3	19.0	4.45	复位 1 个字时
				659.0	187.23	复位 256 个字时

比较指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
输入比较指令（无符号）	LD,AND,OR+=	300	4	9.3	1.61	---
	LD,AND,OR+<>	305			1.61	
	LD,AND,OR+<	310			1.61	
	LD,AND,OR+<=	315			1.61	
	LD,AND,OR+>	320			1.61	
	LD,AND,OR+>=	325			1.61	
输入比较指令（双字、无符号）	LD,AND,OR+=+L	301	4	10.8	1.92	---
	LD,AND,OR+<>+L	306			1.92	
	LD,AND,OR+<+L	311			1.92	
	LD,AND,OR+<=+L	316			1.92	
	LD,AND,OR+>+L	321			1.92	
	LD,AND,OR+>=+L	326			1.92	
输入比较指令（带符号）	LD,AND,OR+=+S	302	4	9.4	2.04	---
	LD,AND,OR+<>+S	307			1.64	
	LD,AND,OR+<+S	312			1.64	
	LD,AND,OR+<=+S	317			1.64	
	LD,AND,OR+>+S	322			2.04	
	LD,AND,OR+>=+S	327			2.04	
输入比较指令（双字、带符号）	LD,AND,OR+=+SL	303	4	10.9	2.36	---
	LD,AND,OR+<>+SL	308			2.31	
	LD,AND,OR+<+SL	313			2.31	
	LD,AND,OR+<=+SL	318			2.31	
	LD,AND,OR+>+SL	323			2.36	
	LD,AND,OR+>=+SL	328			2.36	
时间比较指令	=DT	341	4	14.5	3.15	---
	<>DT	342	4	14.5	3.18	---
	<DT	343	4	14.4	3.19	---
	<=DT	344	4	14.4	3.19	---
	>DT	345	4	14.6	3.18	---
	>=DT	346	4	14.6	3.19	---
比较	CMP	020	3	8.1	1.43	---
	!CMP	020	7	49.1	6.06	---
双字比较	CMPL	060	3	9.5	1.75	---
带符号二进制比较	CPS	114	3	8.1	1.95	---
	!CPS	114	7	49.1	6.13	---
带符号双字二进制比较	CPSL	115	3	9.5	2.29	---
表比较	TCMP	085	4	61.1	14.27	---
无符号块比较	BCMP	068	4	107.6	23.55	---
区域范围比较	ZCP	088	3	17.8	4.10	---
双字区域范围比较	ZCPL	116	3	20.1	4.70	---

数据传送指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
传送	MOV	021	3	8.0	1.76	---
	!MOV	021	7	57.7	8.63	---
双字传送	MOVL	498	3	8.9	2.10	---
传送反	MVN	022	3	13.7	3.27	---
位传送	MOVB	082	4	21.4	4.97	---
数位传送	MOVD	083	4	22.4	5.21	---
多位传送	XFRB	062	4	26.4	6.04	传送 1 个字
				137.3	29.83	传送 255 位
块传送	XFER	070	4	24.2	5.74	传送 1 个字
				3747.7	737.62	传送 1,000 个字
块设置	BSET	071	4	21.3	5.00	设置 1 个字
				2074.4	385.06	设置 1,000 个字
数据交换	XCHG	073	3	19.2	4.35	---
单字分配	DIST	080	4	20.8	4.65	---
数据收集	COLL	081	4	20.6	4.65	---
索引寄存器设定	MOVR	560	3	---	3.27	---
	MOVRW	561	3	---	3.27	---

数据移位指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
移位寄存器	SFT	010	3	14.1	3.40	使 1 个字移位
				1076.0	29.50	使 290 个字移位
可逆移位寄存器	SFTR	084	4	18.0	6.30	使 1 个字移位
				3784.4	802.37	使 1,000 个字移位
字移位	WSFT	016	4	25.8	5.86	使 1 个字移位
				3783.9	785.80	使 1,000 个字移位
算术左移	ASL	025	2	13.0	3.57	---
双字左移	ASLL	570	2	---	3.57	---
算术右移	ASR	026	2	13.0	3.57	---
双字右移	ASRL	571	2	---	3.54	---
循环左移	ROL	027	2	13.3	3.63	---
双字循环左移	ROLL	572	2	---	3.75	---
循环右移	ROR	028	2	13.5	3.57	---
双字循环右移	RORL	573	2	---	3.77	---
一个数位左移	SLD	074	3	21.8	4.97	使 1 个字移位
				3778.3	801.47	使 1,000 个字移位
一个数位右移	SRD	075	3	22.2	5.05	使 1 个字移位
				3778.6	798.54	使 1,000 个字移位
左移 N 位	NASL	580	3	19.5	5.05	---
双字左移 N 位	NSLL	582	3	20.8	5.34	---
右移 N 位	NASR	581	3	19.6	5.05	---
双字右移 N 位	NSRL	583	3	21.0	5.35	---

递增 / 递减指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
二进制递增	++	590	2	12.3	3.23	---
双字二进制递增	++L	591	2	13.5	3.56	---
二进制递减	--	592	2	12.3	3.24	---
双字二进制递减	--L	593	2	13.6	3.59	---
BCD 递增	++B	594	2	13.2	3.42	---
双字 BCD 递增	++BL	595	2	14.4	3.79	---
BCD 递减	--B	596	2	13.2	3.42	---
双字 BCD 递减	--BL	597	2	14.5	3.76	---

四则运算指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
无进位带符号二进制加	+	400	4	11.5	2.55	---
无进位带符号双字二进制加	+L	401	4	13.0	2.92	---
有进位带符号二进制加	+C	402	4	11.7	2.61	---
有进位带符号双字二进制加	+CL	403	4	13.2	2.96	---
无进位 BCD 加	+B	404	4	20.6	4.61	---
无进位双字 BCD 加	+BL	405	4	22.9	5.14	---
有进位 BCD 加	+BC	406	4	20.8	4.63	---
有进位双字 BCD 加	+BCL	407	4	23.1	5.19	---
无进位带符号二进制减	–	410	4	11.6	2.61	---
无进位带符号双字二进制减	–L	411	4	13.2	2.98	---
有进位带符号二进制减	–C	412	4	11.7	2.64	---
有进位带符号双字二进制减	–CL	413	4	13.3	3.01	---
无进位 BCD 减	–B	414	4	20.3	4.60	---
无进位双字 BCD 减	–BL	415	4	23.6	5.19	---
有进位 BCD 减	–BC	416	4	20.5	4.65	---
有进位双字 BCD 减	–BCL	417	4	23.8	5.27	---
带符号二进制乘	*	420	4	18.4	4.19	---
带符号双字二进制乘	*L	421	4	23.9	5.53	---
无符号二进制乘	*U	422	4	---	4.30	---
无符号双字二进制乘	*UL	423	4	---	6.25	---
BCD 乘	*B	424	4	22.0	4.79	---
双字 BCD 乘	*BL	425	4	33.2	7.01	---
带符号二进制除	/	430	4	19.8	4.47	---
带符号双字二进制除	/L	431	4	25.8	5.67	---
无符号二进制除	/U	432	4	---	4.49	---
无符号双字二进制除	/UL	433	4	---	5.69	---
BCD 除	/B	434	4	23.2	4.97	---
双字 BCD 除	/BL	435	4	33.0	6.71	---

转换指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
BCD →二进制	BIN	023	3	15.1	3.63	---
双字 BCD →双字二进制	BINL	058	3	16.7	3.91	---
二进制→ BCD	BCD	024	3	15.1	3.59	---
双字二进制→双字 BCD	BCDL	059	3	17.3	3.93	---
二进制求补	NEG	160	3	14.3	3.35	---
数据译码	MLPX	076	4	19.6	4.68	对 1 个数位译码 (4 ~ 16)
				31.0	7.12	对 4 个数位译码 (4 ~ 16)
				79.4	16.49	对 1 个数位译码 (8 ~ 256)
				138.2	28.84	对 2 个数位译码 (8 ~ 256)
数据编码	DMPX	077	4	32.5	7.57	对 1 个数位编码 (16 ~ 4)
				63.0	14.58	对 4 个数位编码 (16 ~ 4)
				68.0	8.54	对 1 个数位编码 (256 ~ 8)
				112.3	10.73	对 2 个数位编码 (256 ~ 8)
ASCII 转换	ASC	086	4	22.8	5.48	将 1 个数位转换成 ASCII 码
				24.7	6.15	将 4 个数位转换成 ASCII 码
ASCII →十六进制	HEX	162	4	18.4	4.53	转换 1 个数位

逻辑指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
逻辑与	ANDW	034	4	18.6	4.26	---
双字逻辑与	ANDL	610	4	20.4	4.68	---
逻辑或	ORW	035	4	18.6	4.26	---
双字逻辑或	ORWL	611	4	20.4	4.67	---
异或	XORW	036	4	18.6	4.26	---
双字异或	XORL	612	4	20.4	4.70	---
求补	COM	029	2	12.4	3.24	---
双字求补	COML	614	2	13.6	3.56	---

特殊算术指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
算术处理	APR	069	4	34.2	4.88	指定 SIN 和 COS
				25.9	7.56	指定线段接近
位计数器	BCNT	067	4	19.5	4.44	对 1 个字计数

浮点算术运算指令

指令	助记符	功能代码	长度（步数）	执行时间(μs)		条件
				CP1E	CP2E	
浮点数→16 位	FIX	450	3	15.9	3.87	---
浮点数→32 位	FIXL	451	3	16.2	3.75	---
16 位→浮点数	FLT	452	3	16.2	3.42	---
32 位→浮点数	FLTL	453	3	17.1	3.59	---
浮点数加	+F	454	4	24.1	5.04	---
浮点数减	–F	455	4	25.2	5.05	---
浮点数除	/F	457	4	25.0	5.21	---
浮点数乘	*F	456	4	24.4	5.21	---
浮点符号比较	LD,AND,OR+=F	329	3	11.6	2.34	---
	LD,AND,OR+<>F	330			1.98	
	LD,AND,OR+<F	331			1.98	
	LD,AND,OR+<=F	332			1.98	
	LD,AND,OR+>F	333			1.99	
	LD,AND,OR+>=F	334			1.99	
浮点数→ ASCII	FSTR	448	4	56.8	14.81	---
ASCII →浮点数	FVAL	449	3	42.9	9.95	---

表数据处理指令

指令	助记符	功能代码	长度（步数）	执行时间(μs)		条件
				CP1E	CP2E	
交换字节	SWAP	637	3	16.8	4.01	交换 1 个字
				6250.0	1255.28	交换 1,000 个字
最大值检索	MAX	182	4	---	5.97	比较 1 个字
					641.20	比较 1,000 个字
最小值检索	MIN	183	4	---	5.99	比较 1 个字
					641.99	比较 1,000 个字
帧校验和	FCS	180	4	24.1	5.62	对于 1 个字的表长度
				2710.0	622.57	对于 1,000 个字的表长度

数据控制指令

指令	助记符	功能代码	长度（步数）	执行时间(μs)		条件
				CP1E	CP2E	
带自动整定功能的 PID 控制	PIDAT	191	4	316.0	76.42	首次执行 PID 处理
				270.0	48.24	采样时执行 PID 处理
				228.0	47.59	不采样时执行 PID 处理
				275.5	53.81	初次执行自动整定
				276.0	50.04	采样时执行自动整定
时间比例输出	TPO	685	4	5.8	3.52	OFF 执行时间
				40.8	12.36	占空比指定或显示的输出限位有效时的 ON 执行时间
				43.4	8.26	被控变量指定和输出限位有效时的 ON 执行时间
定标	SCL	194	4	24.8	7.61	---
定标 2	SCL2	486	4	20.2	6.46	---
定标 3	SCL3	487	4	26.4	7.89	---
平均值	AVG	195	4	24.2	8.23	1 次运算的平均值
				225.5	47.15	64 次运算的平均值

子程序指令

指令	助记符	功能代码	长度（步数）	执行时间 (µs)		条件
				CP1E	CP2E	
子程序调用	SBS	091	2	6.6	4.38	---
子程序入口	SBN	092	2	2.6	1.10	---
子程序返回	RET	093	1	3.1	1.20	---

中断控制指令

指令	助记符	功能代码	长度（步数）	执行时间 (µs)		条件
				CP1E	CP2E	
设置中断屏蔽	MSKS	690	3	15.1	5.19	设定
				15.1	4.23	复位
清除中断	CLI	691	3	14.9	3.22	设定
				18.0	3.22	复位
禁止中断	DI	693	1	8.5	0.90	---
允许中断	EI	694	1	8.9	0.88	---

高速计数器和脉冲输出指令

指令	助记符	功能代码	长度（步数）	执行时间 (µs)		条件
				CP1E	CP2E	
模式控制	INI	880	4	46.0	4.53	启动高速计数器比较
				31.8	4.53	停止高速计数器比较
				48.7	10.80	改变脉冲输出 PV
				35.2	8.30	改变高速计数器 PV
				27.2	10.67	停止脉冲输出
				13.0	4.67	停止 PWM(891) 输出
读高速计数器的 PV 值	PRV	881	4	40.0	9.23	读脉冲输出 PV
				35.0	8.11	读高速计数器 PV
				37.2	8.28	读脉冲输出状态
				32.6	7.17	读高速计数器状态
				24.5	5.22	读 PWM(891) 状态
				36.5	3.68	读高速计数器的范围比较结果
				29.1	4.23	读高速计数器 0 的频率
比较表载入	CTBL	882	4	69.3	16.61	注册目标值表并启动1个目标值的比较
				116.3	26.94	注册目标值表并启动6个目标值的比较
				126.6	10.89	注册范围表并启动比较
				46.3	10.73	只为 1 个目标值注册目标值表
				93.3	21.06	只为 6 个目标值注册目标值表
				122.5	26.89	只注册范围表
速度输出	SPED	885	4	69.2	19.50	连续模式
				74.0	21.23	单独模式
设置脉冲	PULS	886	4	44.1	10.50	---
脉冲输出	PLS2	887	5	97.6	32.62	---
加速度控制	ACC	888	4	75.6	29.62	连续模式
				82.8	35.37	单独模式

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
原点搜索	ORG	889	3	52.2	15.89	原点搜索
				126.8	20.08	原点返回
占空比可变脉冲	PWM	891	4	28.9	9.44	---
中断进给	IFEED	892	4	---	31.50	---
直线插补	ITPL	893	4	---	89.1	2 轴插补
					106.64	3 轴插补
					122.7	4 轴插补

步指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
步定义	STEP	008	2	10.5	3.13	步控制位 ON
				10.4	3.13	步控制位 OFF
步启动	SNXT	009	2	9.6	2.82	---

I/O 单元指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
I/O 刷新	IORF	097	3	170.7	114.11	刷新 CP1W 扩展单元的 1 个输入字
				146.6	123.25	刷新 CP1W 扩展单元的 1 个输出字
				1725.8	1245.43	刷新 CP1W 扩展单元的 12 个输入字
				1359.9	1354.20	刷新 CP1W 扩展单元的 12 个输出字
7 段译码	SDEC	078	4	21.9	5.11	---
矩阵输入	MTR	213	5	31.6	7.42	数据输入值：00
				31.6	7.42	数据输入值：FF
7 段显示输出	7SEG	214	5	27.1	8.28	4 个数位
				30.8	9.32	8 个数位

串行通信指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
发送	TXD	236	4	25.0	7.65	发送 1 个字节
				25.0	7.65	发送 256 个字节
接收	RXD	235	4	99.6	12.90	存储 1 个字节
				1113.3	339.00	存储 256 个字节

网络通信指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
网络发送	SEND	090	4	---	8.60	发送 499 个字
网络接收	RECV	098	4	---	8.70	接收 (发送请求)501 个字
指令发送	CMND	490	4	---	9.32	指令数据 1006 个字节

时钟指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
日历加	CADD	730	4	56.6	14.71	---
日历减	CSUB	731	4	55.1	14.02	---
时钟调整	DATE	735	2	29.9	8.22	---

故障诊断指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
故障报警	FAL	006	3	55.6	32.21	记录错误
				79.6	14.23	删除错误 (按优先级的顺序)
				61.6	15.54	删除错误 (所有错误)
				60.0	19.01	删除错误 (单个删除)
严重故障报警	FALS	007	3	---	---	---

其它指令

指令	助记符	功能代码	长度（步数）	执行时间 (μs)		条件
				CP1E	CP2E	
置进位	STC	040	1	32.6	0.67	---
清除进位	CLC	041	1	3.9	0.67	---
延长最大循环时间	WDT	094	2	11.7	2.47	---

3-2 关于使用功能块时的实例执行时间

创建功能块定义，在用户程序中粘贴实例后，基于实例执行的周期时间的影响时间请以如下时间为大致标准。

基于实例的周期时间的影响时间 = 启动时间 A + 输入输出参数传送处理时间 B + 功能块定义内指令执行时间 C

各 A、B、C 的时间请以如下时间为大致标准。

内容			CP2E CPU 单元
A	启动时间	除输入输出参数传送以外的启动时间	54.37 μs
B	输入输出参数传送处理时间 ( ) 内为数据类型	1 位 (BOOL) 的输入变量、输出变量时	9.74 μs
		1 字 (INT、UINT、WORD) 的输入变量、输出变量时	5.4 μs
		2 字 (DINT、UDINT、DWORD、REAL) 的输入变量、输出变量时	6.31 μs
		4 字 (LINT、ULINT、LWORD、LREAL) 的输入变量、输出变量时	20.9 μs
C	功能块定义内指令执行时间	合计指令处理时间 (与通常的用户程序相同)	

例)

数据类型为 1 字 (INT) 的输入变量 = 3 个、

数据类型为 1 字 (INT) 的输出变量 = 2 个、

功能块定义部分的合计指令处理时间为 10 μs 时：

1 个实例执行时间 = 54.37 μs + (3+2) × 5.4 μs+ 10 μs = 91.37 μs。

注 粘贴多个相同的功能块定义时 (多个实例时)，执行时间将根据实例数而增加。

附加信息

关于使用功能块时的程序步数

创建功能块定义，在用户程序中粘贴实例时的程序步数请以如下步数为大致标准。

步数 = 实例数 × (调用部容量 m + 输入输出参数传送部容量 n × 参数数) + 功能块定义内指令步数 p(注)

注 粘贴多个相同的功能块定义时 (多个实例时)，第 2 个以后的实例不会消耗功能块定义内指令步数 (p)。因此，上述计算公式未对功能块定义内指令步数 (p) 乘以实例数。

内容			CP2E CPU 单元
m	调用部		57 步
n	输入输出参数传送部 ( ) 内为数据类型	1 位 (BOOL) 的输入变量、输出变量时	6 步
		1 字 (INT、UINT、WORD) 的输入变量、输出变量时	6 步
		2 字 (DINT、UDINT、DWORD、REAL) 的输入变量、输出变量时	6 步
		4 字 (LINT、ULINT、LWORD、LREAL) 的输入变量、输出变量时	12 步
p	功能块定义内指令步数	合计指令步数 (与通常的用户程序相同) <u>加上 27 步后</u>的步数	

例)

数据类型为 1 字 (INT) 的输入变量 = 5 个、

数据类型为 1 字 (INT) 的输出变量 = 5 个、

功能块定义部分的步数为 100 步时：

1 个实例的步数 = 57 + (5+5) × 6 步 +100 步 + 27 步 = 244 步。

4

循环时间的监控和计算

本章节介绍了如何监控和计算可在程序中使用的 CP1E/CP2E CPU 单元的循环时间。

4-1 循环时间的监控

4-1-1 循环时间的监控

当 CX-Programmer 在线连接至 CPU 单元时，可对循环时间的平均值、最大值、最小值进行监控。

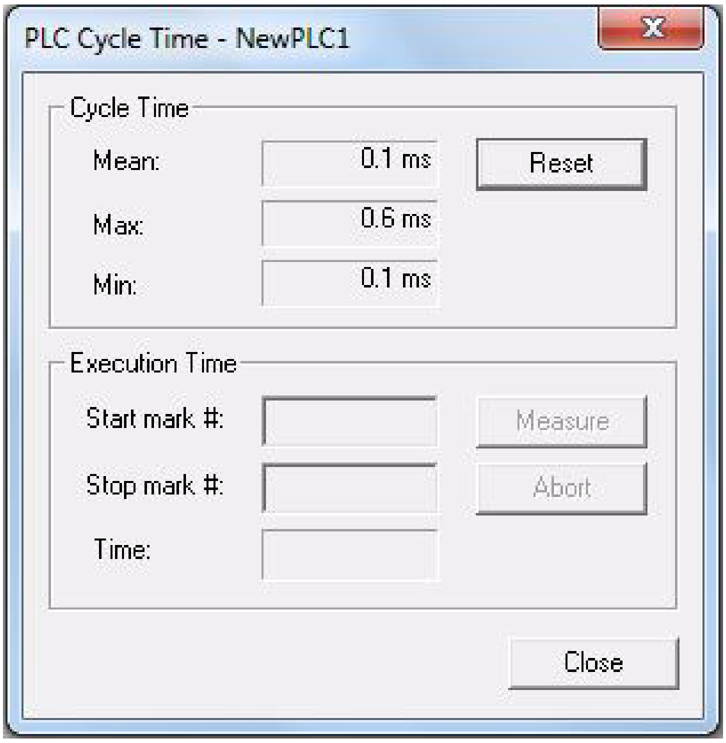
平均值的监控

在在线连接到 PLC 上的情况下，当 CPU 单元处于 PROGRAM 以外的任何模式时，均会在状态栏显示平均循环时间。

最大值和最小值的监控

在 PLC 菜单中选择 ***"PLC Setting–PLC Information–Cycle Time"*** 选项。

随即便显示 "PLC Cycle Time" 对话框。



循环时间的平均值、最大值和最小值将自上而下依次显示。

若要重新计算和显示循环时间，请点击 **"Reset"** 按钮。

附加信息

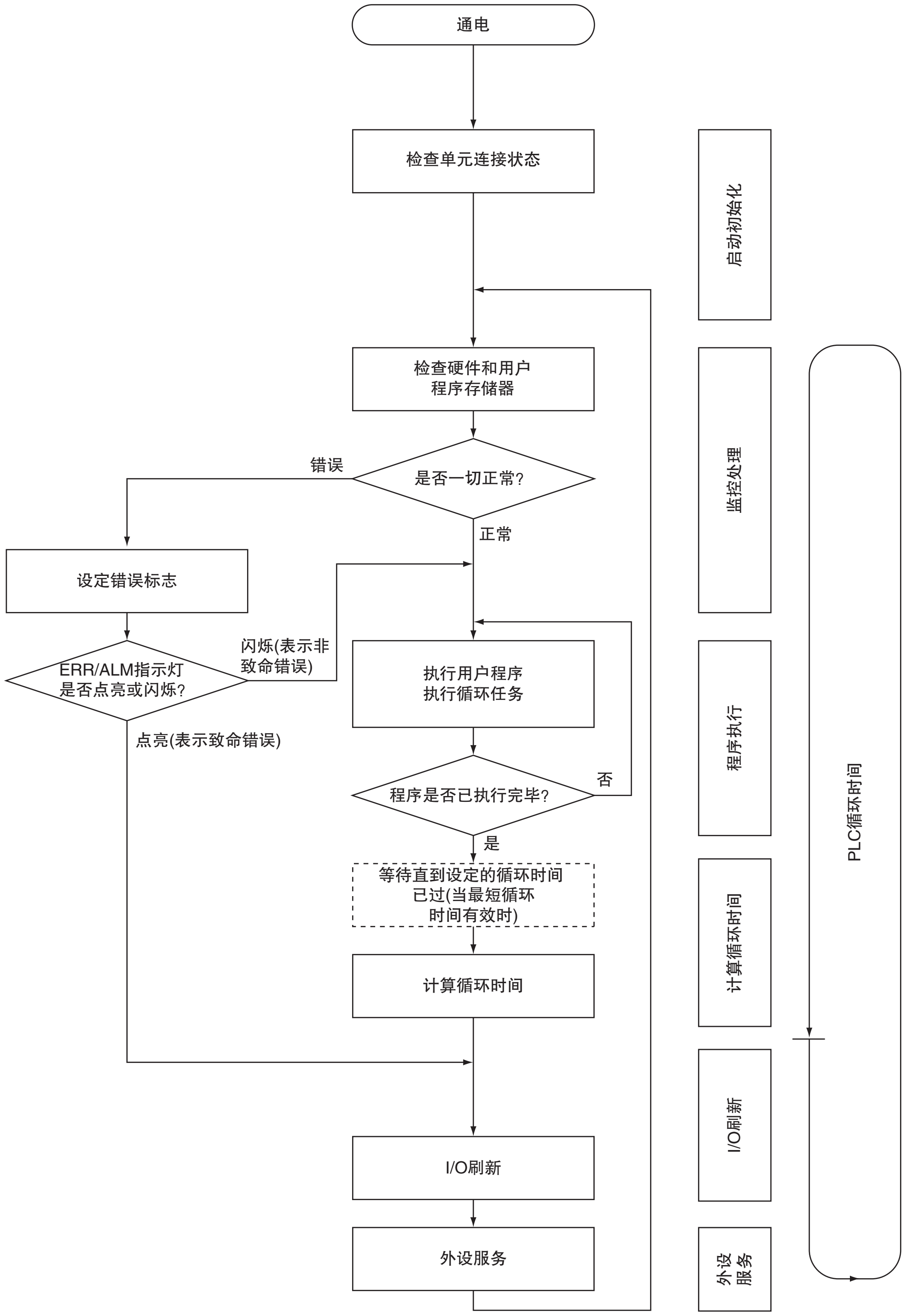
循环时间当前值和最大值存储在下列辅助区字中：

- 循环时间当前值 (以 0.1ms 为增量单位)：A264(低位字节) 和 A265(高位字节)
- 循环时间最大值 (以 0.1ms 为增量单位)：A262(低位字节) 和 A263(高位字节)

4-2 循环时间的计算

4-2-1 CPU 单元运行流程图

CPU 单元的数据处理过程如下图所示，即按照从监控处理到外设服务的顺序不断重复循环。



4-2-2 循环时间概述

循环时间取决于下列因素。

- 用户程序中的指令类型和数量 (满足执行条件的循环任务和所有中断任务)
- CP 系列扩展单元和扩展 I/O 单元的类型和数量
- PLC 设置中的最小 (常数) 循环时间设定
- 外设 USB、Ethernet 和串行端口的使用情况

正确使用注意事项

当从 MONITOR 模式切换至 RUN 模式时，循环时间可能会延长 10ms(但这并不会造成循环时间超长错误)。

循环时间为 PLC 运行所需要的总时间，具体如下表所示。

循环时间 =(1)+(2)+(3)+(4)+(5)

(1) 监控

操作	处理时间和波动原因
检查 I/O 总线和用户存储器以及是否存在电池错误等。	0.4ms 以上

(2) 执行程序

操作	处理时间和波动原因
执行用户程序中的各条指令。所需时间为执行所有指令的总时间。	执行指令的总时间

(3) 计算循环时间 (最小循环时间)

操作	处理时间和波动原因
当已在 PLC 设置中设定了最小 (常数) 循环时间时，请等待直到指定的循环时间已过。 计算循环时间。	若未设定最小循环时间，则第 3 步所需的时间约为 0。 若设定了最小循环时间，则第 3 步所需时间为预设固定循环时间与实际循环时间 ((1)+(2)+(4)+(5)) 的差值。

(4) 执行 I/O 刷新

操作		处理时间和波动原因
CPU 单元内置 I/O、CPU 单元内置模拟量 I/O(仅 NA 型)、CP 系列扩展单元和扩展 I/O 单元	首先对从 CPU 单元输出到实际输出的输出进行刷新 (针对各个单元)，然后再进行输入。	各个单元的 I/O 刷新时间乘以所用单元的数量。

(5) 外设服务

操作	处理时间和波动原因
提供外设 USB 端口相关服务	在该项服务中，将为外设服务留出一定的时间，即上一个循环时间的 8%(步骤 (3) 中的计算值)。
提供串行端口相关服务 (内置 RS-232C 端口、内置 RS-485 端口、串行选件板)	
提供 Ethernet 端口相关服务	

4-2-3 PLC 单元的 I/O 刷新时间

● 内置模拟量 I/O 的 I/O 刷新时间 (仅 NA 型 CPU 单元)

单元名称		型号	各单元的 I/O 刷新时间
NA 型 CPU 单元	20 点 I/O 型 + 模拟量 I/O 型 CPU 单元	CP1E-NA20D □ - □	0.5ms

注 无论是否使用模拟量 I/O 功能，I/O 刷新时间均相同。

● CP 系列扩展单元和扩展 I/O 单元的 I/O 刷新时间

单元名称		型号	各单元的 I/O 刷新时间
扩展 I/O 单元	8 点输入单元	CP1W-8ED	0.14ms
	8 点输出单元	CP1W-8ER	0.06ms
		CP1W-8ET	
		CP1W-8ET1	
	16 点输出单元	CP1W-16ER	0.17ms
		CP1W-16ET	
		CP1W-16ET1	
	20 点 I/O 单元	CP1W-20EDR1	0.20ms
		CP1W-20EDT	
		CP1W-20EDT1	
	32 点输出单元	CP1W-32ER	0.33ms
		CP1W-32ET	
		CP1W-32ET1	
	40 点 I/O 单元	CP1W-40EDR	0.45ms
		CP1W-40EDT	
		CP1W-40EDT1	
扩展单元	模拟量输入单元	CP1W-AD041	0.72ms
		CP1W-AD042	0.87ms
	模拟量输出单元	CP1W-DA021	0.33ms
		CP1W-DA041	
		CP1W-DA042	0.40ms
	模拟量 I/O 单元	CP1W-MAD11	0.36ms
		CP1W-MAD42	0.87ms
		CP1W-MAD44	0.97ms
	温度传感器单元	CP1W-TS001	0.30ms
		CP1W-TS002	0.57ms
		CP1W-TS003	0.67ms
		CP1W-TS004	0.47ms
		CP1W-TS101	0.30ms
		CP1W-TS102	0.57ms
	CompoBus/S I/O 链接单元	CP1W-SRT21	0.20ms

附加信息

CPU 单元内置 I/O 的 I/O 刷新时间包含在监控处理时间内。

4-2-4 循环时间计算示例

下列示例介绍了在扩展 I/O 单元连接至 CP1E CPU 单元上的情况下，用于计算循环时间的方法。

● 条件

项目	描述	
CP1E CPU 单元	40 点 I/O 单元 CP1W-40EDR	1 个
梯形图	5K 步	LD 指令：2.5K 步 OUT 指令：2.5K 步
外设 USB 端口连接	是或否	
最小循环时间处理	无	
串行端口连接	无	
其它外设服务	无	

● 计算示例

过程名称	公式	处理时间	
		连接外设 USB 端口时	未连接外设 USB 端口时
(1) 监控	−	0.4ms	0.4ms
(2) 执行程序	1.19μs × 2,500+1.61μs × 2,500	7.0ms	7.0ms
(3) 计算循环时间	(未设定最小循环时间)	0ms	0ms
(4) I/O 刷新	0.45ms	0.45ms	0.45ms
(5) 外设服务	(仅连接外设 USB 端口时)	0.2ms	0ms
循环时间	(1)+(2)+(3)+(4)+(5)	8.15ms	7.85ms

4-2-5 延长在线编辑的循环时间

CPU 单元处于 MONITOR 运行模式下时，若通过 CX-Programmer 执行在线编辑以修改程序，则 CPU 单元将在程序改变后暂时停止运行。而循环时间则会因为写入 CPU 单元程序而得以延长。而且，在 CPU 单元暂停运行时将不执行定时任务。

循环时间延长情况如下所示：

CPU 单元	延长在线编辑的循环时间
CP1E CPU 单元	最长：16ms ；正常：6ms(适用于 8K 步大小的程序)
CP2E CPU 单元	最长：3.5ms ；正常：2.3ms(适用于 10K 步大小的程序)

进行在线编辑时，循环时间将得以延长，且定时任务的执行将视编辑的执行情况出现延迟或发生异常。

App

附录

App

按助记符首字母顺序编排的指令列表

A

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
ACC	加速控制	888	@ACC	---	---	2-352
AND	和	---	@AND	%AND	!AND	2-9
AND LD	逻辑块与	---	---	---	---	2-13
AND NOT	与非	---	---	---	!AND NOT	2-9
AND TST	AND 型・位测试	350	---	---	---	2-20
AND TSTN	AND 型・非位测试	351	---	---	---	2-20
AND<	与小于	310	---	---	---	2-94
AND<=	与小于等于	315	---	---	---	2-94
AND<= F	与浮点数小于等于	332	---	---	---	2-258
AND<= DT	与时间小于等于	344	---	---	---	2-97
AND<= L	与双字小于等于	316	---	---	---	2-94
AND<= S	与带符号小于等于	317	---	---	---	2-94
AND<= SL	与双字带符号小于等于	318	---	---	---	2-94
AND<>	与不等于	305	---	---	---	2-94
AND<> DT	与时间不等于	342	---	---	---	2-97
AND<> F	与浮点数不等于	330	---	---	---	2-258
AND<> L	与双字不等于	306	---	---	---	2-94
AND<> S	与带符号不等于	307	---	---	---	2-94
AND<> SL	与双字带符号不等于	308	---	---	---	2-94
AND< DT	与时间小于	343	---	---	---	2-97
AND<F	与浮点数小于	331	---	---	---	2-258
AND<L	与双字小于	311	---	---	---	2-94
AND<S	与带符号小于	312	---	---	---	2-94
AND< SL	与双字带符号小于	313	---	---	---	2-94
AND=	与等于	300	---	---	---	2-94
AND= DT	与时间等于	341	---	---	---	2-97
AND=F	与浮点数等于	329	---	---	---	2-258
AND=L	与双字等于	301	---	---	---	2-94
AND=S	与带符号等于	302	---	---	---	2-94
AND= SL	与双字带符号等于	303	---	---	---	2-94
AND>	与大于	320	---	---	---	2-94
AND>=	与大于等于	325	---	---	---	2-94
AND>= DT	与时间大于等于	346	---	---	---	2-97
AND>= F	与浮点数大于等于	334	---	---	---	2-258
AND>= L	与双字大于等于	326	---	---	---	2-94
AND>= S	与带符号大于等于	327	---	---	---	2-94
AND>= SL	与双字带符号大于等于	328	---	---	---	2-94
AND> DT	与时间大于	345	---	---	---	2-97
AND>F	与浮点数大于	333	---	---	---	2-258
AND>L	与双字大于	321	---	---	---	2-94
AND>S	与带符号大于	322	---	---	---	2-94
AND> SL	与双字带符号大于	323	---	---	---	2-94
ANDL	双字逻辑与	610	@ANDL	---	---	2-227
ANDW	逻辑与	034	@ANDW	---	---	2-227
APR	算术处理	069	@APR	---	---	2-235
ASC	ASCII 转换	086	@ASC	---	---	2-218
ASL	算术左移	025	@ASL	---	---	2-143
ASLL	双字左移	570	@ASLL	---	---	2-143

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
ASR	算术右移	026	@ASR	---	---	2-145
ASRL	双字右移	571	@ASRL	---	---	2-145
AVG	平均值	195	---	---	---	2-308

B

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
BCD	二进制→BCD	024	@BCD	---	---	2-204
BCDL	双字二进制→双字BCD	059	@BCDL	---	---	2-204
BCMP	块比较	068	@BCMP	---	---	2-109
BCNT	位计数器	067	@BCNT	---	---	2-244
BIN	BCD →二进制	023	@BIN	---	---	2-202
BINL	双字 BCD→双字二进制	058	@BINL	---	---	2-202
BREAK	循环中断	514	---	---	---	2-65
BSET	块设置	071	@BSET	---	---	2-126

C

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
CADD	日历加	730	@CADD	---	---	2-435
CJP	条件跳转	510	---	---	---	2-59
CLC	清除进位	041	@CLC	---	---	2-455
CLI	清除中断	691	@CLI	---	---	2-324
CMND	指令发送	490	@CMND	---	---	2-431
CMP	比较	020	---	---	!CMP	2-101
CMPL	双字比较	060	---	---	---	2-101
CNR	复位定时器 / 计数器	545	@CNR	---	---	2-92
CNRX	复位定时器 / 计数器	547	@CNRX	---	---	2-92
CNT	计数器	---	---	---	---	2-86
CNTR	可逆计数器	012	---	---	---	2-89
CNTRX	可逆计数器	548	---	---	---	2-89
CNTX	计数器	546	---	---	---	2-86
COLL	数据收集	081	@COLL	---	---	2-132
COM	求补	029	@COM	---	---	2-233
COML	双字求补	614	@COML	---	---	2-233
CPS	带符号二进制比较	114	---	---	!CPS	2-104
CPSL	带符号双字二进制比较	115	---	---	---	2-104
CSUB	日历减	731	@CSUB	---	---	2-435
CTBL	比较表载入	882	@CTBL	---	---	2-336

D

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
DATE	时钟调整	735	@DATE	---	---	2-440
DI	禁止中断	693	@DI	---	---	2-327
DIFD	下降沿微分	014	---	---	!DIFD	2-33
DIFU	上升沿微分	013	---	---	!DIFU	2-31
DIST	单字分配	080	@DIST	---	---	2-130
DMPX	数据编码	077	@DMPX	---	---	2-213
DOWN	条件 OFF	522	---	---	---	2-17
DSW	数字开关输入	210	---	---	---	2-387

E

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
EI	允许中断	694	---	---	---	2-328
END	结束	001	---	---	---	2-44

F

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
FAL	故障报警	006	@FAL	---	---	2-442
FALS	严重故障报警	007	---	---	---	2-449
FCS	帧校验和	180	@FCS	---	---	2-276
FIX	浮点数→16 位	450	@FIX	---	---	2-250
FIXL	浮点数→32 位	451	@FIXL	---	---	2-250
FLT	16 位→浮点数	452	@FLT	---	---	2-252

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
FLTL	32 位→浮点数	453	@FLTL	---	---	2-252
FOR	---	512	---	---	---	2-62
FSTR	浮点数→ASCII	448	@FSTR	---	---	2-261
FVAL	ASCII →浮点数	449	@FVAL	---	---	2-266

H

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
HEX	ASCII →十六进制	162	@HEX	---	---	2-222

I

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
IFEED	中断进给	892	@IFEED	---	---	2-362
IL	互锁	002	---	---	---	2-46
ILC	互锁清除	003	---	---	---	2-46
INI	模式控制	880	@INI	---	---	2-329
IORF	I/O 刷新	097	@IORF	---	---	2-382
ITPL	直线插补	893	@ITPL	---	---	2-366

J

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
JME	跳转结束	005	---	---	---	2-59
JMP	跳转	004	---	---	---	2-59

K

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
KEEP	保持	011	---	---	!KEEP	2-27

L

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
LD	载入	---	@LD	%LD	!LD	2-7
LD NOT	载入非	---	---	---	!LD NOT	2-7

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
LD TST	LD 型・位测试	350	---	---	---	2-18
LD TSTN	LD 型・非位测试	351	---	---	---	2-18
LD<	载入小于	310	---	---	---	2-94
LD<=	载入小于等于	315	---	---	---	2-94
LD<= DT	载入时间小于等于	344	---	---	---	2-97
LD<=F	载入浮点数小于等于	332	---	---	---	2-258
LD<=L	载入双字小于等于	316	---	---	---	2-94
LD<=S	载入带符号小于等于	317	---	---	---	2-94
LD<= SL	载入双字带符号小于等于	318	---	---	---	2-94
LD<>	载入不等于	305	---	---	---	2-94
LD<> DT	载入时间不等于	342	---	---	---	2-97
LD<>F	载入浮点数不等于	330	---	---	---	2-258
LD<>L	载入双字不等于	306	---	---	---	2-94
LD<>S	载入带符号不等于	307	---	---	---	2-94
LD<> SL	载入双字带符号不等于	308	---	---	---	2-94
LD<DT	载入时间小于	343	---	---	---	2-97
LD<F	载入浮点数小于	331	---	---	---	2-258
LD<L	载入双字小于	311	---	---	---	2-94
LD<S	载入带符号小于	312	---	---	---	2-94
LD<SL	载入双字带符号小于	313	---	---	---	2-94
LD=	载入等于	300	---	---	---	2-94
LD=DT	载入时间等于	341	---	---	---	2-97
LD=F	载入浮点数不等于	329	---	---	---	2-258
LD=L	载入双字等于	301	---	---	---	2-94

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
LD=S	载入带符号等于	302	---	---	---	2-94
LD=SL	载入双字带符号等于	303	---	---	---	2-94
LD>	载入大于	320	---	---	---	2-94
LD>=	载入大于等于	325	---	---	---	2-94
LD>=DT	载入时间大于等于	346	---	---	---	2-97
LD>=F	载入浮点数大于等于	334	---	---	---	2-258
LD>=L	载入双字大于等于	326	---	---	---	2-94
LD>=S	载入带符号大于等于	327	---	---	---	2-94
LD>=SL	载入双字带符号大于等于	328	---	---	---	2-94
LD>DT	载入时间大于	345	---	---	---	2-97
LD>F	载入浮点数大于	333	---	---	---	2-258
LD>L	载入双字大于	321	---	---	---	2-94
LD>S	载入带符号大于	322	---	---	---	2-94
LD>SL	载入双字带符号大于	323	---	---	---	2-94

M

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
MAX	最大值检索	182	@MAX	---	---	2-272
MILC	多路互锁清除	519	---	---	---	2-50
MILH	多路互锁微分保持	517	---	---	---	2-50
MILR	多路互锁微分释放	518	---	---	---	2-50
MIN	最小值检索	183	@MIN	---	---	2-272
MLPX	数据译码	076	@MLPX	---	---	2-208
MOV	传送	021	@MOV	---	!MOV	2-115
MOVB	位传送	082	@MOVB	---	---	2-118
MOVD	数位传送	083	@MOVD	---	---	2-120
MOVL	双字传送	498	@MOVL	---	---	2-115
MOVR	索引寄存器设定	560	@MOVR	---	---	2-134
MOVRW	索引寄存器设定	561	@MOVRW	---	---	2-134
MSKS	设置中断屏蔽	690	@MSKS	---	---	2-321
MTR	矩阵输入	213	---	---	---	2-391
MVN	传送反	022	@MVN	---	---	2-115

N

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
NASL	左移 N 位	580	@NASL	---	---	2-153
NASR	右移 N 位	581	@NASR	---	---	2-156
NEG	二进制求补	160	@NEG	---	---	2-206
NEXT	---	513	---	---	---	2-62
NOP	空操作	000	---	---	---	2-45
NOT	非	520	---	---	---	2-16
NSLL	双字左移 N 位	582	@NSLL	---	---	2-153
NSRL	双字右移 N 位	583	@NSRL	---	---	2-156

O

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
OR	或	---	@OR	%OR	!OR	2-11
ORG	原点搜索	889	@ORG	---	---	2-357
OR LD	逻辑块或	---	---	---	---	2-13
OR NOT	或非	---	---	---	!OR NOT	2-11
OR TST	OR 型・位测试	350	---	---	---	2-22
OR TSTN	OR 型・非位测试	351	---	---	---	2-22
OR<	或小于	310	---	---	---	2-94
OR<=	或小于等于	315	---	---	---	2-94
OR<=DT	或时间小于等于	344	---	---	---	2-97
OR<=F	或浮点数小于等于	332	---	---	---	2-258

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
OR<=L	或双字小于等于	316	---	---	---	2-94
OR<=S	或带符号小于等于	317	---	---	---	2-94
OR<=SL	或双字带符号小于等于	318	---	---	---	2-94
OR<>	或不等于	305	---	---	---	2-94
OR<>DT	或时间不等于	342	---	---	---	2-97
OR<>F	或浮点数不等于	330	---	---	---	2-258
OR<>L	或双字不等于	306	---	---	---	2-94
OR<>S	或带符号不等于	307	---	---	---	2-94
OR<>SL	或双字带符号不等于	308	---	---	---	2-94
OR<DT	或时间小于	343	---	---	---	2-97
OR<F	或浮点数小于	331	---	---	---	2-258
OR<L	或双字小于	311	---	---	---	2-94
OR<S	或带符号小于	312	---	---	---	2-94
OR<SL	或双字带符号小于	313	---	---	---	2-94
OR=	或等于	300	---	---	---	2-94
OR=DT	或时间等于	341	---	---	---	2-97
OR=F	或浮点数等于	329	---	---	---	2-258
OR=L	或双字等于	301	---	---	---	2-94
OR=S	或带符号等于	302	---	---	---	2-94
OR=SL	或双字带符号等于	303	---	---	---	2-94
OR>	或大于	320	---	---	---	2-94
OR>=	或大于等于	325	---	---	---	2-94
OR>=DT	或时间大于等于	346	---	---	---	2-97
OR>=F	或浮点数大于等于	334	---	---	---	2-258
OR>=L	或双字大于等于	326	---	---	---	2-94
OR>=S	或带符号大于等于	327	---	---	---	2-94
OR>=SL	或双字带符号大于等于	328	---	---	---	2-94
OR>DT	或时间大于	345	---	---	---	2-97
OR>F	或浮点数大于	333	---	---	---	2-258
OR>L	或双字大于	321	---	---	---	2-94
OR>S	或带符号大于	322	---	---	---	2-94
OR>SL	或双字带符号大于	323	---	---	---	2-94
ORW	逻辑或	035	@ORW	---	---	2-229
ORWL	双字逻辑或	611	@ORWL	---	---	2-229
OUT	输出	---	---	---	!OUT	2-24
OUT NOT	反相输出	---	---	---	!OUT NOT	2-24

P

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
PIDAT	带自动整定的 PID 控制	191	---	---	---	2-278
PLS2	脉冲输出	887	@PLS2	---	---	2-346
PRV	读高速计数器的 PV 值	881	@PRV	---	---	2-332
PULS	设置脉冲	886	@PULS	---	---	2-344
PWM	占空比可变脉冲	891	@PWM	---	---	2-360

R

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
RECV	网络接收	098	@RECV	---	---	2-428
RET	子程序返回	093	---	---	---	2-316
ROL	循环左移	027	@ROL	---	---	2-147
ROLL	双字循环左移	572	@ROLL	---	---	2-147
ROR	循环右移	028	@ROR	---	---	2-149

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
RORL	双字循环右移	573	@RORL	---	---	2-149
RSET	复位	---	@RSET	%RSET	!RSET	2-35
RSTA	多个位复位	531	@RSTA	---	---	2-37
RSTB	单个位复位	533	@RSTB	---	!RSTB	2-39
RXD	接收	235	@RXD	---	---	2-404

S

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
SBN	子程序入口	092	---	---	---	2-316
SBS	子程序调用	091	@SBS	---	---	2-311
SCL	定标	194	@SCL	---	---	2-297
SCL2	定标 2	486	@SCL2	---	---	2-301
SCL3	定标 3	487	@SCL3	---	---	2-305
SDEC	7 段译码	078	@SDEC	---	---	2-384
SEND	网络发送	090	@SEND	---	---	2-424
SET	置位	---	@SET	%SET	!SET	2-35
SETA	多个位置位	530	@SETA	---	---	2-37
SETB	单个位置位	532	@SETB	---	!SETB	2-39
SFT	移位寄存器	010	---	---	---	2-137
SFTR	可逆移位寄存器	084	@SFTR	---	---	2-139
SLD	一个数位左移	074	@SLD	---	---	2-151
SNXT	步启动	009	---	---	---	2-372
SPED	速度输出	885	@SPED	---	---	2-340
SRD	一个数位右移	075	@SRD	---	---	2-151
STC	置进位	040	@STC	---	---	2-455
STEP	步定义	008	---	---	---	2-372
SWAP	交换字节	637	@SWAP	---	---	2-270

T

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
TCMP	表比较	085	@TCMP	---	---	2-107
TIM	100ms 定时器	---	---	---	---	2-72
TIMH	10ms 定时器	015	---	---	---	2-75
TIMHX	10ms 定时器	551	---	---	---	2-75
TIML	长定时器	542	---	---	---	2-83
TIMLX	长定时器	553	---	---	---	2-83
TIMX	100ms 定时器	550	---	---	---	2-72
TMHH	1ms 定时器	540	---	---	---	2-78
TMHHX	1ms 定时器	552	---	---	---	2-78
TPO	时间比例输出	685	---	---	---	2-290
TR	TR 位	---	---	---	---	2-26
TTIM	累加定时器	087	---	---	---	2-80
TTIMX	累加定时器	555	---	---	---	2-80
TXD	发送	236	@TXD	---	---	2-399

U

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
UP	条件 ON	521	---	---	---	2-17

W

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
WDT	延长最大循环时间	094	@WDT	---	---	2-456
WSFT	字移位	016	@WSFT	---	---	2-141

X

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
XCHG	数据交换	073	@XCHG	---	---	2-128
XFER	块传送	070	@XFER	---	---	2-124
XFRB	多位传送	062	@XFRB	---	---	2-122
XORL	双字异或	612	@XORL	---	---	2-231
XORW	异或	036	@XORW	---	---	2-231

Z

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
ZCP	区域范围比较	088	---	---	---	2-111
ZCPL	双字区域范围比较	116	---	---	---	2-111

符号

助记符	指令	功能代码	上升沿微分	下降沿微分	即时刷新指定	页码
7SEG	7 段显示输出	214	---	---	---	2-395
+	无进位带符号二进制加	400	@+	---	---	2-170
++	二进制递增	590	@++	---	---	2-159
++B	BCD 递增	594	@++B	---	---	2-165
++BL	双字 BCD 递增	595	@++BL	---	---	2-165
++L	双字二进制递增	591	@++L	---	---	2-159
+B	无进位 BCD 加	404	@+B	---	---	2-174
+BC	有进位 BCD 加	406	@+BC	---	---	2-176
+BCL	有进位双字 BCD 加	407	@+BCL	---	---	2-176
+BL	无进位双字 BCD 加	405	@+BL	---	---	2-174
+C	有进位带符号二进制加	402	@+C	---	---	2-172
+CL	有进位带符号双字二进制加	403	@+CL	---	---	2-172
+F	浮点数加	454	@+F	---	---	2-254
+L	无进位带符号双字二进制加	401	@+L	---	---	2-170
-	无进位带符号二进制减	410	@-	---	---	2-178
--	二进制递减	592	@--	---	---	2-162
--B	BCD 递减	596	@--B	---	---	2-168
--BL	双字 BCD 递减	597	@--BL	---	---	2-168
--L	二进制递减	593	@--L	---	---	2-162
-B	无进位 BCD 减	414	@-B	---	---	2-185
-BC	有进位 BCD 减	416	@-BC	---	---	2-188
-BCL	有进位双字 BCD 减	417	@-BCL	---	---	2-188
-BL	无进位双字 BCD 减	415	@-BL	---	---	2-185
-C	有进位带符号二进制减	412	@-C	---	---	2-182
-CL	有进位带符号双字二进制减	413	@-CL	---	---	2-182
-F	浮点数减	455	@-F	---	---	2-254
-L	无进位带符号双字二进制减	411	@-L	---	---	2-178
*	带符号二进制乘	420	@*	---	---	2-190
*B	BCD 乘	424	@*B	---	---	2-194
*BL	双字 BCD 乘	425	@*BL	---	---	2-194
*F	浮点数乘	456	@*F	---	---	2-254
*L	带符号双字二进制乘	421	@*L	---	---	2-190
*U	无符号二进制乘	422	@*U	---	---	2-192
*UL	无符号双字二进制乘	423	@*UL	---	---	2-192
/	带符号二进制除	430	@/	---	---	2-196
/B	BCD 除	434	@/B	---	---	2-200
/BL	双字 BCD 除	435	@/BL	---	---	2-200
/F	浮点数除	457	@/F	---	---	2-254
/L	带符号双字二进制除	431	@/L	---	---	2-196
/U	无符号二进制除	432	@/U	---	---	2-198
/UL	无符号双字二进制除	433	@/UL	---	---	2-198

ASCII 码表

		最左边的4位															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
最右边的4位	0			Sp	0	@	P	‘	p				ー	タ	ミ		
	1			!	1	A	Q	a	q			。	ア	チ	ム		
	2			”	2	B	R	b	r			「	イ	ツ	メ		
	3			#	3	C	S	c	s			」	ウ	テ	モ		
	4			$	4	D	T	d	t			、	エ	ト	ヤ		
	5			%	5	E	U	e	u			・	オ	ナ	ユ		
	6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
	7			’	7	G	W	g	w			ァ	キ	ヌ	ラ		
	8			(	8	H	X	h	x			ィ	ク	ネ	リ		
	9			)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
	A			*	:	J	Z	j	z			ェ	コ	ハ	レ		
	B			+	;	K	[	k	{			ォ	サ	ヒ	ロ		
	C			,	<	L	¥	l	|			ャ	シ	フ	ワ		
	D			−	=	M	]	m	}			ュ	ス	ヘ	ン		
	E			.	>	N	^	n	~			ョ	セ	ホ	゛		
	F			/	?	O	_	o				ッ	ソ	マ	゜		

修订记录

手册封面上的手册编号的后缀部分即修订号。

Cat. No. W483-CN5-07

修订号

修订号	日期	修订内容
01	2009 年 3 月	首次出版
02	2009 年 6 月	勘误
03	2010 年 1 月	E10/14、N14/60 和 NA20 CPU 单元新增信息
04	2010 年 6 月	CP 系列扩展单元新增型号 CP1W-DA021
05	2012 年 11 月	勘误。
06	2014 年 11 月	CP 系列扩展单元中增加 CP1W-AD042 模拟量输入单元，CP1W-DA042 模拟量输出单元，CP1W-MAD42/MAD44 模拟量 I/O 单元和 CP1W-TS003/TS004 温度传感器单元。
07	2019 年 10 月	新增 CP2E CPU 单元 新增以下指令 ・顺序输入指令：LD TST/LD TSTN, AND TST/AND TSTN, OR TST/OR TSTN ・数据传送指令：MOVR/MOVRW ・数据移位指令：ASLL, ASRL, ROLL, RORL ・四则运算指令：/U, /UL, *U, *UL ・表数据处理指令：MAX/MIN ・高速计数器和脉冲输出指令：IFEED, ITPL ・网络通信指令：SEND, RECV, CMND

购买欧姆龙产品的客户须知

承诺事项

承蒙对欧姆龙株式会社（以下简称“本公司”）产品的一贯厚爱和支持，藉此机会再次深表谢意。
如果未特别约定，无论贵司从何处购买的产品，都将适用本承诺事项中记载的事项。
请在充分了解这些注意事项基础上订购。

1. 定义

本承诺事项中的术语定义如下。

(1)“本公司产品”：是指“本公司”的 FA 系统机器、通用控制器、传感器、电子 / 结构部件。
(2)“产品目录等”：是指与“本公司产品”有关的欧姆龙综合产品目录、FA 系统设备综合产品目录、安全组件综合产品目录、电子 / 机构部件综合产品目录以及其他产品目录、规格书、使用说明书、操作指南等，包括以电子数据方式提供的资料。
(3)“使用条件等”：是指在“产品目录等”资料中记载的“本公司产品”的使用条件、额定值、性能、运行环境、操作使用方法、使用时的注意事项、禁止事项以及其他事项。
(4)“客户用途”：是指客户使用“本公司产品”的方法，包括将“本公司产品”组装或运用到客户生产的部件、电子电路板、机器、设备或系统等产品中。
(5)“适用性等”：是指在“客户用途”中“本公司产品”的 (a) 适用性、(b) 动作、(c) 不侵害第三方知识产权、(d) 法规法令的遵守以及 (e) 满足各种规格标准。

2. 关于记载事项的注意事项

对“产品目录等”中的记载内容，请理解如下要点。

(1) 额定值及性能值是在单项试验中分别在各条件下获得的值，并不构成对各额定值及性能值的综合条件下获得值的承诺。
(2) 提供的参考数据仅作为参考，并非可在该范围内一直正常运行的保证。
(3) 应用示例仅作参考，不构成对“适用性等”的保证。
(4) 如果因技术改进等原因，“本公司”可能会停止“本公司产品”的生产或变更“本公司产品”的规格。

3. 使用时的注意事项

选用及使用本公司产品时请理解如下要点。

(1) 除了额定值、性能指标外，使用时还必须遵守“使用条件等”。
(2) 客户应事先确认“适用性等”，进而再判断是否选用“本公司产品”。“本公司”对“适用性等”不做任何保证。
(3) 对于“本公司产品”在客户的整个系统中的设计用途，客户应负责事先确认是否已进行了适当配电、安装等事项。
(4) 使用“本公司产品”时，客户必须采取如下措施：(i) 相对额定值及性能指标，必须在留有余量的前提下使用“本公司产品”，并采用冗余设计等安全设计 (ii) 所采用的安全设计必须确保即使“本公司产品”发生故障时也可将“客户用途”中的危险降到最小程度、(iii) 构建随时提示使用者危险的完整安全体系、(iv) 针对“本公司产品”及“客户用途”定期实施各项维护保养。
(5) 因 DDoS 攻击（分布式 DoS 攻击）、计算机病毒以及其他技术性有害程序、非法侵入，即使导致“本公司产品”、所安装软件、或者所有的计算机器材、计算机程序、网络、数据库受到感染，对于由此而引起的直接或间接损失、损害以及其他费用，“本公司”将不承担任何责任。
对于 (i) 杀毒保护、(ii) 数据输入输出、(iii) 丢失数据的恢复、(iv) 防止“本公司产品”或者所安装软件感染计算机病毒、(v) 防止对“本公司产品”的非法侵入，请客户自行负责采取充分措施。
(6)“本公司产品”是作为应用于一般工业产品的通用产品而设计生产的。除“本公司”已表明可用于特殊用途的，或已经与客户有特殊约定的情形外，若客户将“本公司产品”直接用于以下用途的，“本公司”无法作出保证。
(a) 必须具备很高安全性的用途（例：核能控制设备、燃烧设备、航空 / 宇宙设备、铁路设备、升降设备、娱乐设备、医疗设备、安全装置、其他可能危及生命及人身安全的用途）
(b) 必须具备很高可靠性的用途（例：燃气、自来水、电力等供应系统、24 小时连续运行系统、结算系统、以及其他处理权利、财产的用途等）
(c) 具有苛刻条件或严酷环境的用途（例：安装在室外的设备、会受到化学污染的设备、会受到电磁波影响的设备、会受到振动或冲击的设备等）
(d)“产品目录等”资料中未记载的条件或环境下的用途
(7) 除了不适用于上述 3. (6) (a) 至 (d) 中记载的用途外，“本产品目录等资料中记载的产品”也不适用于汽车（含二轮车，以下同）。请勿配置到汽车上使用。关于汽车配置用产品，请咨询本公司销售人员。

4. 保修条件

“本公司产品”的保修条件如下。

(1) 保修期限　自购买之日起 1 年。（但是，“产品目录等”资料中有明确说明时除外。）
(2) 保修内容　对于发生故障的“本公司产品”，由“本公司”判断并可选择以下其中之一方式进行保修。
(a) 在本公司的维修保养服务点对发生故障的“本公司产品”进行免费修理（但是对于电子、结构部件不提供修理服务。）
(b) 对发生故障的“本公司产品”免费提供同等数量的替代品
(3) 当故障因以下任何一种情形引起时，不属于保修的范围。
(a) 将“本公司产品”用于原本设计用途以外的用途
(b) 超过“使用条件等”范围的使用
(c) 违反本注意事项“3. 使用时的注意事项”的使用
(d) 非因“本公司”进行的改装、修理导致故障时
(e) 非因“本公司”出品的软件导致故障时
(f)“本公司”生产时的科学、技术水平无法预见的原因
(g) 除上述情形外的其它原因，如“本公司”或“本公司产品”以外的原因（包括天灾等不可抗力）

5. 责任限制

本承诺事项中记载的保修是关于“本公司产品”的全部保证。对于因“本公司产品”而发生的其他损害，“本公司”及“本公司产品”的经销商不负任何责任。

6. 出口管理

客户若将“本公司产品”或技术资料出口或向境外提供时，请遵守中国及各国关于安全保障进出口管理方面的法律、法规。否则，“本公司”有权不予提供“本公司产品”或技术资料。

IC321GC-zh
201911

注：规格如有变更，恕不另行通知。请以最新产品说明书为准。

欧姆龙自动化（中国）有限公司

http://www.fa.omron.com.cn　　咨询热线：400-820-4535

Cat. No. W483-CN5-07