

OMRON

可 编 程 序 控 制 器

CQM1H 系列
CQM1H 内 插 板

编 程 手 册

SYSMAC CQM1H 系列

CQM1H – CPU□□可编程序控制器

CQM1H – □□□□□内插板




编程手册

1999年9月生产

注意:

OMRON 制造的产品由具有一定资格的人员按适当步骤使用,并只能按本手册描述的功能使用。

下列约定用于指明本手册中几类注意事项,必须按照这些要求做,否则可能引起严重的人身伤害或产品损坏。

-  **危险** 指示如果不按这些信息做,会引起严重的人身伤害。
-  **警告** 指示如果不按这些信息做,可能引起严重的人身伤害。
-  **注意** 指示如果不按这些信息做,可能造成一些伤害或产品损坏或操作失败。

OMRON 产品参考符号

所有 OMRON 产品在本手册中都用大写字母,“Unit”(单元)这个字为 OMRON 产品时也用大写字母,不管它是否是产品的合适名字。

缩写“Ch”(通道)出现在一些 OMRON 产品中表示“字”在意思,也可以缩写成“Wd”。

缩写“PC”只表示可编程控制器的意思,并且不表示其它任何的缩写。

辅助信息

手册中左列的下列首标用于帮助显示不同信息。

注 指示如何使产品更有效、方便地进行工作的信息。

1,2,3…… 1. 指示分类的列表,如步骤、检查表等。

© OMRON, 1999

版权所有。没有得到 OMRON 的书面许可,不允许将本出版物的任何部分以任何形式或任何方式(机械、电子、照相复制、记录等)进行翻版、存储或传输。

对使用本书包含的信息没有特许的责任,并且,由于 OMRON 一直致力于改进它的高质量的产品,本手册中所包含的信息进行修改时不作预告。虽然本手册已经作了各种预防措施,但不对于使用本出版物所含的信息所造成的损坏承担任何责任。

目 录

| | |
|----------------------------|------------|
| 注意事项 | xiii |
| 1. 预期的读者 | xiv |
| 2. 一般注意事项 | xiv |
| 3. 安全注意事项 | xiv |
| 4. 操作环境注意事项 | xv |
| 5. 应用注意事项 | xv |
| 6. 符合 EC 规范 | xvii |
| 第一章 | |
| PC 设置和其它特性 | 1 |
| 1-1 PC 设置 | 2 |
| 1-2 内板设定 | 8 |
| 1-3 基本 PC 操作和 I/O 处理 | 11 |
| 1-4 中断功能 | 17 |
| 1-5 脉冲输出功能 | 38 |
| 1-6 通信功能 | 41 |
| 1-7 计算有符号二进制数据 | 51 |
| 第二章 | |
| 内板 | 55 |
| 2-1 高速计数器板 | 56 |
| 2-2 脉冲 I/O 板 | 79 |
| 2-3 绝对编码器接口板 | 113 |
| 2-4 模拟设定板 | 126 |
| 2-5 模拟量 I/O 板 | 127 |
| 2-6 串行通信板 | 132 |
| 第三章 | |
| 内存区域 | 135 |
| 3-1 内存区结构 | 136 |
| 3-2 IR 区 | 138 |
| 3-3 SR 区 | 149 |
| 3-4 TR 区 | 152 |
| 3-5 HR 区 | 152 |
| 3-6 AR 区 | 152 |
| 3-7 LR 区 | 160 |
| 3-8 定时器/计数器区 | 160 |
| 3-9 DM 区 | 161 |
| 3-10 EM 区 | 162 |
| 3-11 存储器盒用法 | 162 |

第四章

| | |
|----------------------|-----|
| 梯形图编程 | 167 |
| 4-1 基本步骤 | 168 |
| 4-2 指令术语 | 168 |
| 4-3 基本梯形图 | 169 |
| 4-4 控制位状态 | 189 |
| 4-5 工作位(内部继电器) | 191 |
| 4-6 编程注意事项 | 193 |
| 4-7 程序执行 | 195 |

第五章

| | |
|---|-----|
| 指令集 | 197 |
| 5-1 注意事项 | 201 |
| 5-2 指令格式 | 201 |
| 5-3 数据区,定义符值和标志 | 201 |
| 5-4 微分指令 | 203 |
| 5-5 扩展指令 | 203 |
| 5-6 编码右侧指令 | 205 |
| 5-7 指令表 | 208 |
| 5-8 梯形图指令 | 213 |
| 5-9 位控制指令 | 214 |
| 5-10 空操作 - NOP(00) | 218 |
| 5-11 结束 - END(01) | 218 |
| 5-12 联锁和联锁清除 - IL(02)和 ILC(03) | 218 |
| 5-13 跳转和跳转结束 - JMP(04)和 JME(05) | 220 |
| 5-14 用户错误指令: 错误报警和复位 - FAL(06)和严重错误报警 - FALS(07) | 221 |
| 5-15 步指令: 步定义和步开始 - STEP(08)/SNXT(09) | 222 |
| 5-16 定时器和计数器指令 | 224 |
| 5-17 移位指令 | 249 |
| 5-18 数据传送指令 | 257 |
| 5-19 比较指令 | 268 |
| 5-20 转换指令 | 279 |
| 5-21 BCD 计算指令 | 305 |
| 5-22 二进制计算指令 | 316 |
| 5-23 专用数学指令 | 327 |
| 5-24 浮点算术指令 | 335 |
| 5-25 逻辑指令 | 360 |
| 5-26 递增/递减指令 | 363 |
| 5-27 子程序指令 | 365 |
| 5-28 特殊指令 | 367 |
| 5-29 网络指令 | 393 |

| | |
|----------------------------|------------|
| 5-30 通信指令 | 403 |
| 5-31 高级 I/O 指令 | 412 |
| 第六章 | |
| 上位机链接命令 | 425 |
| 6-1 上位机链接命令概述 | 426 |
| 6-2 结束代码 | 427 |
| 6-3 通信步骤 | 429 |
| 6-4 命令和响应格式 | 431 |
| 6-5 上位机链接命令 | 435 |
| 第七章 | |
| CPU 单元操作和处理时间 | 461 |
| 7-1 CPU 单元操作 | 462 |
| 7-2 电源中断 | 463 |
| 7-3 循环时间 | 466 |
| 第八章 | |
| 故障诊断 | 485 |
| 8-1 引言 | 486 |
| 8-2 编程器操作错误 | 486 |
| 8-3 编程错误 | 487 |
| 8-4 用户定义错误 | 488 |
| 8-5 操作错误 | 489 |
| 8-6 错误日志 | 491 |
| 8-7 故障诊断流程图 | 493 |
| 附录 | |
| A 编程指令 | 501 |
| B 错误和算术标志操作 | 507 |
| C 存储器区 | 511 |
| D 使用时钟 | 529 |
| E I/O 分配表 | 531 |
| F 程序代码表 | 533 |
| G FAL 号列表 | 537 |
| H 扩展 ASCII | 539 |
| 术语汇编 | 541 |
| 索引 | 561 |
| 修订历史 | 577 |


关于本手册：

本手册描述了 CQM1H 可编程序控制器编程,包括内存结构、内存内容、梯形图指令等,并包括了下述各章内容。硬件信息和编程器操作步骤参见 CQM1H 操作手册。

请仔细阅读本手册,在开始编程和操作 CQM1H 前确信已经理解本手册的内容。

- 第 1 章 解释 PC 设置和有关 PC 的功能,包括中断处理和通信。PC 设置能用于控制 PC 的工作参数。
- 第 2 章 描述可安装在 CPU 单元内扩展功能的内板,串行通信板的详细请参阅*串行通信板操作手册*(W365),第 2 章仅为这块板的概述。
- 第 3 章 描述 PC 内存区域的结构,解释如何使用它们。并且描述 CPU 单元和存储器盒之间传送数据中使用存储器盒的操作。
- 第 4 章 解释编写梯形图程序的基本步骤和基础,并介绍用于建立梯形图程序基本结构的指令及控制指令执行。
- 第 5 章 分别描述 CQM1H 可以使用的梯形图程序指令。
- 第 6 章 解释使用上位机链接命令的方法和步骤,这些命令通过 PC 端口能与上位机通信。
- 第 7 章 解释 PC 的内部处理,及处理和执行所需的时间。参见该章获得对 PC 工作的精确时序的理解。
- 第 8 章 描述如何诊断和改正 PC 工作中可能出现的硬件和软件错误。

提供下列附录:A 编程指令,B 错误和算术标志操作,C 内存区域,D 使用时钟,E I/O 分配表,F 程序代码表,G FAL 号列表,H 扩展 ASCII。

 **警告** 未阅读和理解本手册提供的信息可能会造成人身伤害、产品损坏或工作错误。在开始进行任何步骤或操作前,请完整阅读每一章并确信已理解每一章及相关章的信息。

注意事项

这一章提供使用 CQM1H 系列可编程控制器(PC)和相关设备的一般注意事项。

这一章包含的信息对安全和可靠应用可编程控制器是相当重要的，在试图设置和操作 PC 系统前必须阅读本章,并且理解包含的信息。

| | |
|------------------|-------|
| 1 预期的读者 | xiv |
| 2 一般注意事项 | xiv |
| 3 安全注意事项 | xiv |
| 4 操作环境注意事项 | xv |
| 5 应用注意事项 | xv |
| 6 符合 EC 规范 | xvii |
| 6-1 适用规范 | xvii |
| 6-2 内容 | xvii |
| 6-3 符合 EC 规范 | xviii |
| 6-4 减小继电器输出噪声的方法 | xviii |

1 预期的读者

本手册适用于具备必要电气系统知识的下列人员(电气工程师或相当人员)。

- FA 系统的安装人员。
- FA 系统的设计人员。
- FA 系统和设备的管理人员。


2 一般注意事项

用户必须按照操作手册描述的性能规格去使用产品。


当本产品的使用条件超出手册所述范围或用于核控制系统、铁路系统、航空系统、汽车燃烧系统、医疗设备、娱乐机器、安全装置和其它一些如果使用不当就会对生命财产造成严重影响的系统、机器和设备,在使用前请先与 OMRON 代理商联系。

确保产品的额定值、特性参数对于应用的系统、机器和设备是充足的,并且保证对它们提供双重的安全机理。


本手册提供了编程和使用此 PC 的信息,使用此 PC 前,必须阅读本手册,并在操作期间把手册放在手边随时参考。


 **警告** 此 PC 和所有 PC 单元用于特殊用途和特定条件下,尤其是用在能直接或间接影响人的生命安全情况下,这尤其重要。在把 PC 用于上述用途之前,必须和 OMRON 代理商联系。


3 安全注意事项


 **警告** 即使程序停止时(也就是在编程模式),CPU 单元也要刷新 I/O,在改变任何分配给 I/O 单元,专用 I/O 单元或内板的部分内存状态前,事先必须确认整个过程是安全的。分配给单元的数据任何改变,可能导致连接在单元上的负载意想不到的动作,下列操作可能导致内存状态的改变。

- 从编程装置传送 I/O 内存数据到 CPU 单元。
- 用编程装置修改当前内存中的值。
- 用编程装置强制置位/复位。
- 从上位计算机或网络上的其它 PC 传送 I/O 内存数据。

 **警告** 通电时,不要拆卸任何单元或触摸内部,否则可能遭受电击。

 **警告** 通电时,不要触摸任何接线端子或接线板,否则可能遭受电击。





 **警告** 不要拆卸、修理、改变任何单元,否则可能会导致故障、火灾、或电击。

 **警告** 为了确保由于 PC 故障或外部因素影响 PC 操作,导致不正常发生时系统的安全,必须在外部回路中(也就是不在可编程控制器中)提供安全措施,包括下列项目,否则可能导致严重事故。




- 紧急停止回路,互锁回路,限位回路,和相似安全措施必须在外部控制回路中提供。
- 当 PC 自诊断功能发现错误或严重失败,警告(FALS)指令执行,PC 会关掉

全部输出,为了应付这样的错误,外部安全措施必须保证系统的安全。

- 由于输出继电器的沉积物和烧毁以及输出晶体管的损坏,PC 输出可能保持 ON 或 OFF,作为这些问题的对策,外部安全措施必须提供以确保系统安全。
- 当 24VDC 输出(供给 PC 的电源)过载或短路,电压会掉下来,导致输出变 OFF,要对付这个问题,外部安全措施必须提供以确保系统安全。


-  **警告** 通电或电源刚关断时,不要去触摸电源单元,否则会受到电击。
-  **小心** 只有在确认由于延长了循环时间,不会产生不利影响后,才能执行在线编辑,否则输入信号会不可读。
-  **小心** 在把一个程序传送到另一个节点或修改 I/O 内存区的内容之前,先确认目的节点的安全性,否则可能引起伤害。
-  **小心** 按操作手册规定的转矩拧紧交流电源单元端子板螺丝,松动的螺丝可能引起烧损或误动作。

4 操作环境注意事项


-  **小心** 不要在下列地方操作控制系统:
 - 阳光直射的地方
 - 温度和湿度不符合规定指标的地方
 - 温度急剧变化的雾化状态下
 - 有腐蚀性或易燃气体的地方
 - 有粉末(特别是金属粉末)或含盐的地方
 - 暴露于水、油或化学物质的地方
 - 剧烈振动或冲击的地方
-  **小心** 当在下列地方安装系统时,必须采取适当及足够的措施:
 - 静电或其他形式的噪声
 - 强烈的电磁场
 - 暴露在放射线幅照下
 - 电源线附近
-  **小心** PC 系统的操作环境对系统的寿命和可靠性有很大的影响。不合适的操作环境可能对 PC 系统造成失灵、故障和其他一些无法预料的问题。在安装时确保操作环境在指定条件之内,并且在系统使用期间保存在指定的条件下。

5 应用注意事项

使用 PC 系统时,遵守以下注意事项。

-  **警告** 注意这些注意事项。不遵守以下注意事项可能导致严重或可能的致命伤害。

- 当安装此单元时,必须使系统接地为 100Ω 或更小。否则可导致电击。
- 在电源模块 GR 和 LG 端短接时,必须安装 100Ω 或更小的接地。
- 执行下列操作时,必须关断电源。否则可能导致误操作或电击。
 - 安装或拆卸 I/O 单元, CPU 单元, 内板, 或任何其他单元。
 - 装配单元。
 - 设定 DIP 开关或旋转开关。
 - 连接电缆或用电线连接系统。
 - 连接或断开连接器。

 **小心** 不遵守下列注意事项可能会引起 PC 或系统的错误操作, 或者会损坏 PC 或 PC 单元。必须注意这些注意事项。

- 在给控制系统上电前必须先给 PC 上电。如果在控制系统供电之后再给 PC 供电, 那么当对 PC 上电时, 因为直流输出单元和其他单元的输出端子瞬时为 ON, 会引起控制系统的信号出现瞬时错误。
- 用户必须采取安全措施, 以防在继电器、晶体管和其它元素内部电路出现故障而使输出单元保持 ON 状态时, 有安全保证。
- 用户必须采取安全措施, 以防在信号线断, 瞬时电源断或其它原因而出现错误, 丢失或非正常信号时, 有安全保证。
- 传送数据时不要关闭 PC 的电源。尤其是在读或写内存卡时不要关闭电源。还有, 当 BUSY 指示灯亮时, 也不要移去内存卡。若要移去一个内存卡, 先按内存卡的电源开关并等待 BUSY 指示灯熄灭, 再移去内存卡。
- 如果 I/O 保持位 (SR 25212) 为 ON, 当 PC 从运行或监视模式切换到编程模式时, PC 的输出将不变 OFF 并且维持其原先的状态。当发生上述情况时, 要确保外部负载不产生危险状况。(当有一个致命错误包括 FALS(007) 指令产生的错误而使运行停止时, 输出单元的所有输出将变 OFF 并且仅有内部输出状态维持。)
- 当用 200 ~ 240V 交流电供电时, 从电压选择端子处把金属跳接线去除。如果供 200 ~ 240V AC 电时, 金属跳接线仍接着, 那么将会损坏产品。
- 总是使用操作手册中指定的电源电压。不正确的电压可能引起失灵或燃烧。
- 采取适当措施以确保供给额定电压和频率的电源。不正确的供电可能会引起失灵。
- 在外接线安装外部电流断路器及其他安全措施以防止短路, 否则可能引起燃烧。
- 不要在输入单元上接超过额定输入电压的电压, 过电压可能引起燃烧。
- 不要在输出单元上接电压或负载超过最大开关容量。过电压或负载可能会引起燃烧。
- 当进行耐压试验时, 不要连接功能地端。否则可能引起燃烧。
- 按操作手册所指定的方法适当地安装单元。不适当安装单元可能引起失灵。
- 确保所有的固定螺丝、端子螺丝和电缆连接器螺丝均按有关手册所指定的扭矩扭紧。否则可能引起失灵。
- 接线时保留粘在单元上的标签。撕去标签后, 如有外物掉入单元会引起失灵。
- 为确保适当的热耗散, 完成接线后, 撕去标签。保留标签可能引起失灵。
- 接线时使用压接端子。不要把多股线直接连到端子上, 这样的连接可能会引起燃烧。
- 正确连接所有接线。
- 打开电源之前把所有接线和开关设定检查二次。不正确的接线可能引起燃烧。

- 在彻底检查端子块和连接器之后,固定单元。
- 在接触一个单元前,确保首先触摸一金属接地物以释放累积的静电。否则会引起失灵或损坏。
- 确保端子块、内存单元、扩展电缆和其他有固定装置的设备被正确固定好。否则会引起失灵。
- 开始运行前,检查开关设定,DM区的内容,及其他准备工作。没有合适的设定或数据就开始操作,可能引起一个意想不到的操作。
- 在单元中正式运行前检查用户程序,不检查程序可能引起意想不到的操作,试图以下操作时,确定系统中不会产生不利影响,不这样做可能引起意想不到的操作。
 - 修改 PC 的工作模式。
 - 强制置位/强制复位内存中的任何位。
 - 修改任何字的当前值或内存中的设定值。
- 在把 DM 区、HR 区的内容及其他恢复运行所需的数据转移到新的 CPU 单元之后再恢复运行。否则可能会引起一个意想不到的操作。
- 不要把电缆拉到或弯曲到他们所允许的范围之外,任何一种做法可能使电缆断掉。
- 不要把东西放在电缆或其他电线上。这样做可能使电缆断裂。
- 更换部件时,确保所更换的新部件是正确的。否则会引起失灵或燃烧。
- 在运输或储存电路板时,用防静电的物体把它们包装起来以防止静电,并且保持一适当的温度。
- 不要裸手触摸电路板或安置在其内的元件。如果处理不当,可能会碰到引起伤害的锋利的引线或板中的其他部件。
- 不要短路电池端或充电、拆卸、加热或烧毁电池,不要使电池受强烈冲击。所有这些可能引起泄漏、破裂、热反应、或电池燃烧,处置掉在地板上或其它形式过多冲击的电池,使用受冲击过的电池可能会泄漏。
- UL 标准要求仅由有经验技术人员调换电池,不能允许无资格人员调换电池。

6 符合 EC 规范

6-1 适用规范

- EMC 规范
- 低压规范

6-2 内容

EMC 规范

OMRON 装置符合 EC 规范和相关 EMC 标准,这样能更容易内置在其它装置和机器中,实际产品都必须通过符合 EMC 标准的检查(见以下注释),产品是否符合用户系统的标准,必须由用户检查。

符合 EC 规范的 OMRON 的装置的相关 EMC 性能根据内装 OMRON 装置的设备或控制板的配置、接线和其它条件而改变,用户必须进行最终检查确认装置和机器全部符合 EMC 标准。

注 下面是适用的 EMC(电磁兼容)标准:

EMS(电磁磁化系数): EN61131-2

EMS(电磁干扰): EN50081-2

(电磁发射: 10-m 规定)

低压规范

始终确认装置工作在电压 50 ~ 1000VAC 或 75 ~ 1500VDC。符合 PC 所需的安全标准(EN61131 - 2)。

6 - 3 符合 EC 规范

CQM1H 系列 PC 符合 EC 规范,为了确保安装 CQM1H 系列 PC 的机器或装置符合 EC 规范,PC 必须安装如下:

- 1,2,3...
1. PC 必须安装在控制板内。
 2. 通信和 I/O 电源所用的 DC 电源必须加强绝缘或双重绝缘。
 3. PC 符合 EC 规范,也符合一般辐射标准(EN50081 - 2),使用继电器输出开关装置产生的噪声可能使整台机器达不到标准,这时,PC 外部必须使用浪涌抑制器或其它措施。

以下方法是减少噪声的典型方法,并不是在所有场合都有效,采取的措施根据控制板连接的装置,系统的接线,配置和其它条件而不同。

6 - 4 减小继电器输出噪声的方法

CQM1H 系列 PC 符合 EMC 规范的一般辐射标准(EN50081 - 2),然而,继电器输出开关动作产生的噪声可能不满足这些标准,这时,负载侧必须连接噪声滤波器或在 PC 外部提供其它相似的措施。

满足标准而采取的措施根据负载侧的装置,机器的接线,配置等不同而不同。下面是减少噪声措施的几个例子。

措施

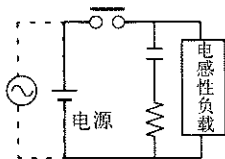
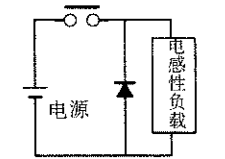
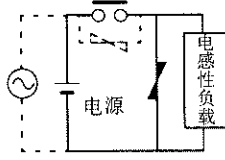
详细参阅 EN50081 - 2 标准。

如果整个系统包括 PC 负载切换的频率小于每分钟 5 次,不需要采取措施。

如果整个系统包括 PC 负载切换的频率每分钟 5 次或更多,需要采取措施。

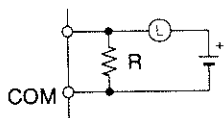
措施举例

切换电感负载时,连接浪涌保护器,二极管等,如下所示,并联在负载或线圈两端。

| 电路 | 电流 | | 特性 | 所需元件 |
|--|----|----|---|---|
| | AC | DC | | |
| 阻容方法  | 可以 | 可以 | 如果负载是继电器或螺旋线圈，在电路断开和负载复位时有一个时间延迟。 如果供电电压是24或48V，在负载两端并联浪涌保护器。 如果供电电压是100~200V，把浪涌保护器串联在接点间。 | 电容的值是每1A接点电流1~0.5μF，电阻的阻值每1V接点电压0.5~1Ω，当然这些值根据负载和继电器的特性而不同，通过测试决定这些值，然后进行补偿，电容抑制线圈断开时的放电火花，电阻限制电路再次合上流过负载的电流。 电容的耐压强度必须是200~300V，如果是AC电路，应使用无极性电容。 |
| 二极管法  | 不行 | 可以 | 并联在负载上的二极管把电路中线圈存储的能量变成电流，然后流过线圈，这样，通过电感性负载的阻值把电流转换成焦耳热能。 用这种方法引起的电路断开和负载复位之间的时间延迟比用阻容方法的要长。 | 二极管的反向耐压强度必须至少是电路最大电压值的10倍，二极管正向电流必须等于或大于负载电流。 如果浪涌保护器用在低压回路的电子线路中，二极管的反向耐压强度应是供电电压的2~3倍。 |
| 压敏电阻方法  | 可以 | 可以 | 压敏电阻法是利用压敏电阻恒定电压的特性来保护接点间高压。电路断开和负载复位间有时间延迟。 如果供电电压是24或48V，把压敏电阻并联在负载上。如果供电电压100~200V，串联在接点间。 | ... |

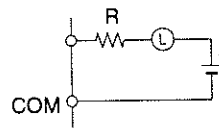
在开关如白炽灯有很大浪涌电流的负载时，要抑制浪涌电流，如下所示。

方法 1



分流大约流过白炽灯额定电流的三分之一电流。

方法 2



使用限流电阻

第 1 章

PC 设置和有关性能

本章阐述了 PC 设置和有关的 CQM1H 性能,包括中断处理和通信。PC 设置能用于控制 CQM1H 的工作参数。要改变 PC 设置,参见 CQM1H 操作手册的编程器操作步骤。CX-Programmer 步骤参见 CX-Programmer 操作手册。

如果对 OMRON PC 或梯形图程序还不太熟悉,可以先大致浏览一下 1-1 PC 设置中 CQM1H 可用的操作参数,然后阅读第 3 章 内存区域,第 4 章 梯形图编程,及第 5 章 指令集中的有关指令,再来阅读本章。

| | | |
|--------|--------------------|----|
| 1-1 | PC 设置 | 2 |
| 1-1-1 | 修改 PC 设置 | 2 |
| 1-1-2 | 串行通信板设定 | 3 |
| 1-1-3 | PC 设置设定 | 4 |
| 1-2 | 内板设定 | 8 |
| 1-2-1 | 串行通信板设定 | 8 |
| 1-2-2 | 高速计数器板设定 | 9 |
| 1-2-3 | 脉冲 I/O 板设定 | 10 |
| 1-2-4 | 绝对编码接口板设定 | 10 |
| 1-2-5 | 模拟量 I/O 板设定 | 11 |
| 1-3 | 基本 PC 操作和 I/O 处理 | 11 |
| 1-3-1 | 启动方式 | 11 |
| 1-3-2 | 保持位状态 | 12 |
| 1-3-3 | RS-232C 端口服务时间 | 12 |
| 1-3-4 | 外围端口服务时间 | 13 |
| 1-3-5 | 最小循环时间 | 13 |
| 1-3-6 | 输入时间常数 | 13 |
| 1-3-7 | 高速定时器 | 14 |
| 1-3-8 | DSW(87)输入数字和输出刷新方法 | 15 |
| 1-3-9 | 外围端口设定 | 15 |
| 1-3-10 | 错误日志设定 | 16 |
| 1-4 | 中断功能 | 17 |
| 1-4-1 | 中断类型 | 17 |
| 1-4-2 | 输入中断 | 19 |
| 1-4-3 | 屏蔽所有中断 | 26 |
| 1-4-4 | 间隔定时器中断 | 27 |
| 1-4-5 | 高速计数器 0 中断 | 29 |
| 1-4-6 | 高速计数器 0 上溢/下溢 | 37 |
| 1-5 | 脉冲输出功能 | 38 |
| 1-6 | 通信功能 | 41 |
| 1-6-1 | 上位机链接和无协议通信设置 | 42 |
| 1-6-2 | 上位机链接通信设置和步骤 | 44 |
| 1-6-3 | 无协议通信设置和步骤 | 46 |
| 1-6-4 | 一对一数据链接 | 49 |
| 1-6-5 | NT 链接 1:1 模式通信 | 51 |
| 1-6-6 | 端口接线 | 51 |
| 1-7 | 有符号二进制数计算 | 51 |
| 1-7-1 | 有符号二进制数定义 | 52 |
| 1-7-2 | 算术标志 | 53 |
| 1-7-3 | 用十进制值输入有符号二进制数 | 53 |
| 1-7-4 | 使用有符号二进制扩展指令 | 53 |
| 1-7-5 | 使用有符号二进制数的应用举例 | 54 |

1-1 PC 设置


PC 设置包括控制 CQM1H 工作的工作参数。在使用中断处理和通信功能时,为了使 CQM1H 功能得到充分的应用,可以根据工作条件对 PC 设置进行用户化。

一般 PC 设置的设定在 DM6600 到 DM6655 中,串行通信板设定在 DM6550 到 DM6559 中。严格地说,串行通信板设定是只读 DM 区的一部分,而不是 PC 设置的一部分,但它们包括在此是因为它们与 PC 设置的设定很相似。

PC 设置根据通常的情况设定了缺省值,因此,CQM1H 不用改变设定就能正常工作,但忠告用户在工作前检查缺省值。

缺省值

PC 设置的所有字的缺省值均为 0000。在任何时候将 SR 25210 置 ON 就能复位 DM6600 到 DM6655 的缺省值。

 小心 用编程设备清除数据存储器(DM)时,PC 设置的所有设定都被清 0。

1-1-1 修改 PC 设置

PC 设置的设定根据设定的不同所读的时间也不同,如下所示。

- DM 6550 ~ DM6559: 正常通电时读。
- DM 6600 ~ DM6614: 仅在 PC 刚通电时读。
- DM 6615 ~ DM6644: 仅在程序执行开始时读。
- DM 6645 ~ DM6655: 正常通电时读。

PC 设置的修改仅在上述时间变为有效。因此 CQM1H 必须重新启动以使 DM6600 ~ DM6614 中的修改有效,程序执行重新启动以使 DM6615 ~ DM6644 中的修改有效。

用编程设备进行修改

在用户程序中能够读取 PC 设置,但不能写入。写 PC 设置只能通过编程器或其它编程设备。

DM 6600 ~ DM 6644 只能在编程模式下设置或修改。

DM 6550 ~ DM 6559 和 DM 6645 ~ DM 6655 可以在编程模式或监视模式下设置或修改。

PC 设置写保护

PC 设置设定后,CPU 单元前面的 DIP 开关的 1 脚可置为 ON 以防止编程设备改写 PC 设置。1 脚置 ON 时,不能从编程设备上改写用户程序,只读 DM 区(DM6144 ~ DM6568)和 PC 设置(DM6600 ~ DM6655)。

PC 设置错误

如果存取了不正确的 PC 设置,将产生非致命错误(错误代码 9B),相应的错误标志将置 ON,并将使用缺省设置。

| 标志 | 功能 |
|-------------------|--|
| AR 2400 | DM 6600 ~ DM 6614中有错误时置ON(上电时读) |
| AR 2401 | DM 6615 ~ DM 6644中有错误时置ON(操作开始时读) |
| AR 2402 | DM 6645 ~ DM 6655中有错误时置ON(上电时读) |
| AR 0400 ~ AR 0407 | DM 6550 ~ DM 6559(上电时读)中有错误时,错误代码10写到此字节中。 |

1-1-2 串行通信板设置

下表显示了 DM 区中串行通信板设置。详细内容, 参见串行通信板操作手册。

| 字 | 位 | 功能 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------|--|----|----|----|----|----|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|
| 串行通信板设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 下列设置在传送到PC后有效。(端口2设置包含在DM6550 ~ 6554中, 端口1设置包含在DM6555 ~ DM6559中。) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6550 (端口2) | 00 ~ 03 | 端口设置 0: 标准(1个启动位, 7位数据, 偶校验, 2个停止位, 9600bps) 1: DM6551中设置(DM6556用于端口1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6555 (端口1) | 04 ~ 07 | CTS控制设置 0: 禁用; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 11 | 用于1:1数据链接的链接字(位12 ~ 15设为3时) 0: LR 00 ~ LR 63; 1: LR 00 ~ LR 31; 2: LR 00 ~ LR 15 最大可编程终端单元号(位12 ~ 15设为5时) 1 ~ 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 通信方式 0: 上位机链接; 1: 无协议; 2: 1:1数据链接从单元; 3: 1:1数据链接主单元; 4: 1:1方式的NT链接; 5: 1:N方式的NT链接; 6: 协议宏 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6551 (端口2) | 00 ~ 07 | 波特率 00: 1.2K, 01: 2.4K, 02: 4.8K, 03: 9.6K, 04: 19.2K | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6556 (端口1) | 08 ~ 15 | 帧格式 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="1"> <thead> <tr> <th></th> <th>起始</th> <th>长度</th> <th>停止</th> <th>校验</th> </tr> </thead> <tbody> <tr> <td>00:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>偶</td> </tr> <tr> <td>01:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>奇</td> </tr> <tr> <td>02:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>无</td> </tr> <tr> <td>03:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>偶</td> </tr> <tr> <td>04:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>奇</td> </tr> <tr> <td>05:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>无</td> </tr> <tr> <td>06:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>偶</td> </tr> <tr> <td>07:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>奇</td> </tr> <tr> <td>08:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>无</td> </tr> <tr> <td>09:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>偶</td> </tr> <tr> <td>10:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>奇</td> </tr> <tr> <td>11:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>无</td> </tr> </tbody> </table> | | 起始 | 长度 | 停止 | 校验 | 00: | 1位 | 7位 | 1位 | 偶 | 01: | 1位 | 7位 | 1位 | 奇 | 02: | 1位 | 7位 | 1位 | 无 | 03: | 1位 | 7位 | 2位 | 偶 | 04: | 1位 | 7位 | 2位 | 奇 | 05: | 1位 | 7位 | 2位 | 无 | 06: | 1位 | 8位 | 1位 | 偶 | 07: | 1位 | 8位 | 1位 | 奇 | 08: | 1位 | 8位 | 1位 | 无 | 09: | 1位 | 8位 | 2位 | 偶 | 10: | 1位 | 8位 | 2位 | 奇 | 11: | 1位 | 8位 | 2位 | 无 |
| | 起始 | 长度 | 停止 | 校验 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00: | 1位 | 7位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01: | 1位 | 7位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02: | 1位 | 7位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03: | 1位 | 7位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04: | 1位 | 7位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05: | 1位 | 7位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06: | 1位 | 8位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07: | 1位 | 8位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08: | 1位 | 8位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 09: | 1位 | 8位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10: | 1位 | 8位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11: | 1位 | 8位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6552 (端口2) | 00 ~ 15 | 传输延迟(上位机链接或无协议) 0000 ~ 9999(BCD): 以10ms为单位设置, 例如, 设置0001等于10ms | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6557 (端口1) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6553 (端口2) | 00 ~ 07 | 节点号(上位机链接) 00 ~ 31(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6558 (端口1) | 08 ~ 11 | 允许起始代码(无协议) 0: 禁用; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 允许结束代码(无协议) 0: 禁用(接收字节数) 1: 设置(指定结束代码) 2: CR, LF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6554 (端口2) | 00 ~ 07 | 起始代码(无协议) 00 ~ FF(十六进制) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6559 (端口1) | 08 ~ 15 | 当DM6553或DM6558的位12 ~ 15设为0时: 接收的字节数 00: 缺省设置(256字节) 01 ~ FF: 1 ~ 255字节 当DM6553或DM6558的位12 ~ 15设为1时: 结束代码(无协议) 00 ~ FF(十六进制) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

1-1-3 设定 PC 设置

下表按 DM 区中的顺序显示 PC 设置的设定。详细内容参见所示页数。

| 字 | 位 | 功能 | 页 |
|--|---------|---|----|
| 启动处理 (DM6600 ~ DM6614) 下列设定仅在传送到PC后,PC重新启动后有效。 | | | |
| DM 6600 | 00 ~ 07 | 启动模式 (位08 ~ 15设为02时有效) 00: 编程; 01: 监视 02: 运行 | 11 |
| | 08 ~ 15 | 启动模式定义 00: 根据CPU单元DIP开关的7脚和编程器开关设置 01: 继续使用上次断电前使用的方式 02: DM6600中的位00 ~ 07设定 | |
| DM 6601 | 00 ~ 07 | 未使用 | 12 |
| | 08 ~ 11 | I/O保持位状态 (SR 25212) 1: 复位; 1: 维持 | |
| | 12 ~ 15 | 强制状态保持位状态 (SR 25211) 0: 复位; 1: 维持 | |
| DM 6602 ~ DM 6603 | 00 ~ 15 | 内板槽1设定 (详细内容参见I-2 内板设定) | 8 |
| DM 6604 ~ DM 6610 | 00 ~ 15 | 未使用 | |
| DM 6611 ~ DM 6612 | 00 ~ 15 | 内板槽2设定 (详细内容参见I-2 内板设定) | 8 |
| DM 6613 | 00 ~ 15 | 串行通信板端口2服务时间设定 | 8 |
| DM 6614 | 00 ~ 15 | 串行通信板端口1服务时间设定 | |
| 脉冲输出和循环时间设定 (DM6615 ~ DM6619) 下列设定在传送到PC后的下次起动工作有效。 | | | |
| DM 6615 | 00 ~ 07 | 脉冲输出字 00: IR100; 01: IR101; 02: IR102... 15: IR115 设置用于从晶体管输出单元输出的脉冲输出的字。一次只能从一个输出中输出脉冲。 | 40 |
| | 08 ~ 15 | 未使用。设置为00。 | |
| DM 6616 | 00 ~ 07 | RS-232C 端口服务时间 (位 08 ~ 15 设为 01 时) 00 ~ 99 (BCD): 用于服务 RS-232C 端 循环时间的百分比。服务时间必须在 0.256ms 和 65.536ms 之间。 | 12 |
| | 08 ~ 15 | 允许RS-232C端口服务设定 00: 循环时间的5% 01: 使用在00 ~ 07中的时间 (当PC停止时,服务时间总是10ms) | |
| DM 6617 | 00 ~ 07 | 外围端口服务时间 (当位08 ~ 15设为01时) 00 ~ 99 (BCD): 用于服务外围端口的循环时间的百分比。服务时间必须在 0.256ms 和 65.536ms 之间。 | 13 |
| | 08 ~ 15 | 允许围端口服务设定 00: 循环时间的5% 01: 使用在00 ~ 07中设定的时间 (当PC停止时,服务时间总是10ms) | |
| DM 6618 | 00 ~ 07 | 循环监视时间 (当位08 ~ 15设为01,02或03) 00 ~ 99 (BCD) × 设定单位 (见位08 ~ 15) | 16 |
| | 08 ~ 15 | 允许循环监视 00: 120ms (禁用位00 ~ 07中的设定) 01: 设定单位: 10ms 02: 设定单位: 100ms 03: 设定单位: 1s | |
| DM 6619 | 00 ~ 15 | 循环时间 0000: 可变 (无最小值) 0001 ~ 9999 (BCD): ms 为单位的最小循环时间 | 13 |

| 字 | 位 | 功能 | 页 |
|------------------------|---------|---|----|
| 中断处理 (DM6620 ~ DM6639) | | | |
| 下列设定仅在传送到PC的下次启动工作有效。 | | | |
| DM 6620 | 00 ~ 03 | IR 00000 ~ IR 00007的输入时间常数 0: 8ms; 1: 1ms; 2: 2ms; 3: 4ms; 4: 8ms; 5: 16ms; 6: 32ms; 7: 64ms; 8: 128ms | 13 |
| | 04 ~ 07 | IR 00008 ~ IR 00015的输入时间常数(与位00 ~ 03设定相同) | |
| | 08 ~ 11 | IR 001的输入时间常数(与位00 ~ 03设定相同) | |
| | 12 ~ 15 | 未使用, 设为0。 | |
| DM 6621 | 00 ~ 07 | IR 002的输入常数 00: 8ms; 01: 1ms; 02: 2ms; 03: 4ms; 04: 8ms; 05: 16ms; 06: 32ms; 07: 64ms; 08: 128ms | 13 |
| | 08 ~ 15 | IR 003的输入常数(与IR 002的设定相同) | |
| DM 6622 | 00 ~ 07 | IR 004的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 005的输入常数(与IR 002的设定相同) | |
| DM 6623 | 00 ~ 07 | IR 006的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 007的输入常数(与IR 002的设定相同) | |
| DM 6624 | 00 ~ 07 | IR 008的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 009的输入常数(与IR 002的设定相同) | |
| DM6625 | 00 ~ 07 | IR 010的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 011的输入常数(与IR 002的设定相同) | |
| DM6626 | 00 ~ 07 | IR 012的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 013的输入常数(与IR 002的设定相同) | |
| DM6627 | 00 ~ 07 | IR 014的输入常数(与IR 002的设定相同) | |
| | 08 ~ 15 | IR 015的输入常数(与IR 002的设定相同) | |
| DM6628 | 00 ~ 03 | IR 00000中断允许 0: 标准输入; 1: 中断输入模式或计数器模式中的中断输入。 | 22 |
| | 04 ~ 07 | IR 00001中断允许 0: 标准输入; 1: 中断输入模式或计数器模式中的中断输入。 | |
| | 08 ~ 11 | IR 00002中断允许 0: 标准输入; 1: 中断输入模式或计数器模式中的中断输入。 | |
| | 12 ~ 15 | IR 00003中断允许 0: 标准输入; 1: 中断输入模式或计数器模式中的中断输入。 | |
| DM6629 | 00 ~ 07 | 通过中断刷新而刷新的TIMH(15)高速定时器数 00 ~ 15(BCD; 例, 设为3, 定时器00 ~ 02) | 14 |
| | 08 ~ 15 | 高速定时器中断刷新允许 00: 16个定时器(位00 ~ 07的设定禁用) 01: 使用00 ~ 07的设定 | |
| DM6630 | 00 ~ 07 | I/O中断0的第一个输入刷新字: 00 ~ 11(BCD) | 22 |
| | 08 ~ 15 | I/O中断0的输入刷新字数: 00 ~ 12(BCD) | |
| DM6631 | 00 ~ 07 | I/O中断1的第一个输入刷新字: 00 ~ 11(BCD) | |
| | 08 ~ 15 | I/O中断1的输入刷新字数: 00 ~ 12(BCD) | |
| DM6632 | 00 ~ 07 | I/O中断2的第一个输入刷新字: 00 ~ 11(BCD) | |
| | 08 ~ 15 | I/O中断2的输入刷新字数: 00 ~ 12(BCD) | |
| DM6633 | 00 ~ 07 | I/O中断3的第一个输入刷新字: 00 ~ 11(BCD) | |
| | 08 ~ 15 | I/O中断3的输入刷新字数: 00 ~ 12(BCD) | |
| DM6634 | 00 ~ 07 | 高速计数器1的第一个输入刷新字: 00 ~ 11(BCD) | 22 |
| | 08 ~ 15 | 高速计数器1的输入刷新字数: 00 ~ 12(BCD) | |
| DM6635 | 00 ~ 07 | 高速计数器2的第一个输入刷新字: 00 ~ 11(BCD) | 22 |
| | 08 ~ 15 | 高速计数器2的输入刷新字数: 00 ~ 12(BCD) | |

| 字 | 位 | 功能 | 页 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------|---|------------|----|----|----|----|----|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|
| DM 6636 | 00 ~ 07 | 间隔定时器0的第一个输入刷新字:00 ~ 15(BCD) | 27, 34 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 间隔定时器0的输入刷新字数:00 ~ 16(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6637 | 00 ~ 07 | 间隔定时器1的第一个输入刷新字:00 ~ 15(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 间隔定时器1的输入刷新字数:00 ~ 16(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6638 | 00 ~ 07 | 间隔定时器2或高速计数器0的第一个输入刷新字: 00 ~ 16(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 间隔定时器2或高速计数器0的输入刷新字数: 00 ~ 16(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6639 | 00 ~ 07 | 输出刷新方法 00: 循环; 01: 直接 | 15, 463 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 数字切换(DSW(87))指令的数字个数 00: 4个数字; 01: 8个数字 | 15, 415 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 高速计数器设定(DM6640 ~ DM6644) 下列设定在传送到PC, 下一次重新启动工作后有效。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6640 ~ DM 6641 | 00 ~ 15 | 内板槽1设定(详细内容参见I-2 内板设定) | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6642 | 00 ~ 03 | 高速计数器0输入模式 0: 相位差模式; 4: 递增模式 | 34 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 04 ~ 07 | 高速计数器0复位模式 0: Z相和软件复位; 1: 仅软件复位 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 高速计数器0允许 00: 不用高速计数器0; 01: 使用高速计数器0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6643 ~ DM 6644 | 00 ~ 15 | 内板槽2设定(详细内容参见I-2 内板设定) | 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RS-232C端口设定 下列设定在传送到PC后有效。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6645 | 00 ~ 03 | 端口设定(上位机链接或无协议模式) 0: 标准(1个起始位, 7位数据, 偶校验, 2个停止位, 9600bps) 1: 在DM6646中设定 | 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 04 ~ 07 | CTS控制设定(上位机链接或无协议模式) 0: 禁止; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 11 | 1: 1数据链接的链接字(1: 1数据链接主模式) 0: LR00 ~ LR 63; 1: LR00 ~ LR 31; 2: LR00 ~ LR 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 通信模式 0: 上位机链接; 1: 无协议; 2: 1: 1数据链接从; 3: 1: 1数据链接主; 4: 1: 1模式的NT链接 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6646 | 00 ~ 07 | 波特率 00: 1.2kbps, 01: 2.4kbps, 02: 4.8kbps, 03: 9.6kbps, 04: 19.2kbps | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 帧格式 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>起始</th> <th>长度</th> <th>停止</th> <th>校验</th> </tr> </thead> <tbody> <tr><td>00:</td><td>1位</td><td>7位</td><td>1位</td><td>偶</td></tr> <tr><td>01:</td><td>1位</td><td>7位</td><td>1位</td><td>奇</td></tr> <tr><td>02:</td><td>1位</td><td>7位</td><td>1位</td><td>无</td></tr> <tr><td>03:</td><td>1位</td><td>7位</td><td>2位</td><td>偶</td></tr> <tr><td>04:</td><td>1位</td><td>7位</td><td>2位</td><td>奇</td></tr> <tr><td>05:</td><td>1位</td><td>7位</td><td>2位</td><td>无</td></tr> <tr><td>06:</td><td>1位</td><td>8位</td><td>1位</td><td>偶</td></tr> <tr><td>07:</td><td>1位</td><td>8位</td><td>1位</td><td>奇</td></tr> <tr><td>08:</td><td>1位</td><td>8位</td><td>1位</td><td>无</td></tr> <tr><td>09:</td><td>1位</td><td>8位</td><td>2位</td><td>偶</td></tr> <tr><td>10:</td><td>1位</td><td>8位</td><td>2位</td><td>奇</td></tr> <tr><td>11:</td><td>1位</td><td>8位</td><td>2位</td><td>无</td></tr> </tbody> </table> | | | 起始 | 长度 | 停止 | 校验 | 00: | 1位 | 7位 | 1位 | 偶 | 01: | 1位 | 7位 | 1位 | 奇 | 02: | 1位 | 7位 | 1位 | 无 | 03: | 1位 | 7位 | 2位 | 偶 | 04: | 1位 | 7位 | 2位 | 奇 | 05: | 1位 | 7位 | 2位 | 无 | 06: | 1位 | 8位 | 1位 | 偶 | 07: | 1位 | 8位 | 1位 | 奇 | 08: | 1位 | 8位 | 1位 | 无 | 09: | 1位 | 8位 | 2位 | 偶 | 10: | 1位 | 8位 | 2位 | 奇 | 11: | 1位 | 8位 |
| | 起始 | 长度 | 停止 | 校验 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00: | 1位 | 7位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01: | 1位 | 7位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02: | 1位 | 7位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03: | 1位 | 7位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04: | 1位 | 7位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05: | 1位 | 7位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06: | 1位 | 8位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07: | 1位 | 8位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08: | 1位 | 8位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 09: | 1位 | 8位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10: | 1位 | 8位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11: | 1位 | 8位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6647 | 00 ~ 15 | 传输延迟(上位机链接或无协议) 0000 ~ 9999(BCD); 以10ms为单位设置, 例如, 设置0001等于10ms | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 字 | 位 | 功能 | 页 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------------------------|---------|--|--------|----|----|----|----|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|-----|----|----|----|---|--|
| DM 6648 | 00 ~ 07 | 节点号(上位机链接):00 ~ 31(BCD) | 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 11 | 起始代码允许(无协议) 0: 禁用; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 结束代码允许(无协议) 0: 禁用(接收的字节数) 1: 设置(指定结束代码) 2: CR, LF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6649 | 00 ~ 07 | 起始代码(无协议) 00 ~ FF(十六进制) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 当DM 6648的位12 ~ 15设为0时: 接收的字节数 00: 缺省设置(256字节) 01 ~ FF: 1 ~ 255字节 当DM 6648的位12 ~ 15设为1时: 结束代码(无协议) 00 ~ FF(十六进制) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 外围端口设定 下列设定在传送到PC后有效。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6650 | 00 ~ 03 | 端口设定(上位机链接或无协议模式) 0: 标准(1个起始位, 7位数据, 偶校验, 2个停止位, 9600bps) 1: 在DM 6651中设定 | 15, 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 04 ~ 07 | CTS控制设定(上位机链接或无协议模式) 0: 禁用; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 11 | 未使用 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 通信模式(当位00 ~ 03设为1时) 0: 上位机链接; 1: 无协议 当编程器连接到外围端口时, 把CPU单元的DIP开关的7脚置为OFF。(在此情况下, 5脚和PC设置设定禁用。) 当个人计算机连接到外围端口用作编程设备时, 把7脚置为ON, 并把通信模式设为“上位机链接”。进行这些设定, 并且个人计算机被设置为外围总线操作后, CPU单元的外围端口通信模式将自动切换到外围总线模式。 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6651 | 00 ~ 07 | 波特率(上位机链接, 外围总线, 或无协议模式) 00: 1.2kbps, 01: 2.4kbps, 02: 4.8kbps, 03: 9.6kbps, 04: 19.2kbps | 41 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 15 | 帧格式(上位机链接或无协议模式) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | <table border="0"> <thead> <tr> <th></th> <th>起始</th> <th>长度</th> <th>停止</th> <th>校验</th> </tr> </thead> <tbody> <tr> <td>00:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>偶</td> </tr> <tr> <td>01:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>奇</td> </tr> <tr> <td>02:</td> <td>1位</td> <td>7位</td> <td>1位</td> <td>无</td> </tr> <tr> <td>03:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>偶</td> </tr> <tr> <td>04:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>奇</td> </tr> <tr> <td>05:</td> <td>1位</td> <td>7位</td> <td>2位</td> <td>无</td> </tr> <tr> <td>06:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>偶</td> </tr> <tr> <td>07:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>奇</td> </tr> <tr> <td>08:</td> <td>1位</td> <td>8位</td> <td>1位</td> <td>无</td> </tr> <tr> <td>09:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>偶</td> </tr> <tr> <td>10:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>奇</td> </tr> <tr> <td>11:</td> <td>1位</td> <td>8位</td> <td>2位</td> <td>无</td> </tr> </tbody> </table> | | 起始 | 长度 | 停止 | 校验 | 00: | 1位 | 7位 | 1位 | 偶 | 01: | 1位 | 7位 | 1位 | 奇 | 02: | 1位 | 7位 | 1位 | 无 | 03: | 1位 | 7位 | 2位 | 偶 | 04: | 1位 | 7位 | 2位 | 奇 | 05: | 1位 | 7位 | 2位 | 无 | 06: | 1位 | 8位 | 1位 | 偶 | 07: | 1位 | 8位 | 1位 | 奇 | 08: | 1位 | 8位 | 1位 | 无 | 09: | 1位 | 8位 | 2位 | 偶 | 10: | 1位 | 8位 | 2位 | 奇 | 11: | 1位 | 8位 | 2位 | 无 | |
| | 起始 | 长度 | 停止 | 校验 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00: | 1位 | 7位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01: | 1位 | 7位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 02: | 1位 | 7位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 03: | 1位 | 7位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 04: | 1位 | 7位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 05: | 1位 | 7位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06: | 1位 | 8位 | 1位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 07: | 1位 | 8位 | 1位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 08: | 1位 | 8位 | 1位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 09: | 1位 | 8位 | 2位 | 偶 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10: | 1位 | 8位 | 2位 | 奇 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11: | 1位 | 8位 | 2位 | 无 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6652 | 00 ~ 15 | 传输延迟(仅无协议或从站启动上位机链接) 0000 ~ 9999(BCD); 以10ms为单位设置, 例如, 设置0001等于10ms | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DM 6653 | 00 ~ 07 | 节点号(上位机链接)00 ~ 31(BCD) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 08 ~ 11 | 起始代码允许(无协议) 0: 禁用; 1: 设置 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 ~ 15 | 结束代码允许(无协议) 0: 禁用(接收的字节数) 1: 设置(指定结束代码) 2: CR, LF | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| 字 | 位 | 功能 | 页 |
|-----------------------------------|---------|--|----|
| DM 6654 | 00 ~ 07 | 起始代码(无协议) 00 ~ FF(十六进制) | 41 |
| | 08 ~ 15 | 当DM 6653的位12 ~ 15设为0时: 接收的字节数 00: 缺省设定(256字节) 01 ~ FF: 1 ~ 255字节 当DM6653的位12 ~ 15设为1时: 结束代码(无协议) 00 ~ FF(十六进制) | |
| 错误日志设定(DM 6655) 下列设置在传送到PC后有效。 | | | |
| DM 6655 | 00 ~ 03 | 形式 0: 存贮10个记录后移位 1: 仅存贮开头10个记录(不移位) 2 ~ F: 不存贮记录 | 16 |
| | 04 ~ 07 | 不使用。设为0。 | |
| | 08 ~ 11 | 循环时间监视器允许 0: 检测长循环的非致命错误 1: 不检测长循环。 | |
| | 12 ~ 15 | 电池电压低错误允许 0: 检测电池电压低为非致命错误 1: 不检测电池电压低 | |

1-2 内板设定

本节描述了与安装在内板槽 1 和 2 上的内板有关的 PC 设置设定。

1-2-1 串行通信板设定

使用 DM 6613 和 DM 6614 中的设定以设置安装在内板槽 1 上的串行通信板的服务时间。(串行通信板不能安装在槽 2 中。)

| 字 | 位 | 功能 |
|---------|---------|---|
| DM 6613 | 00 ~ 07 | 串行通信板端口2的服务时间(通过位08 ~ 15允许) 00 ~ 99(BCD): 设置用于服务端口2的循环时间的百分比。服务时间必须在0.256ms和65.536ms之间 |
| | 08 ~ 15 | 串行通信板端口2服务设定 00: 定为循环时间的5% 01: 使用位00 ~ 07中的时间设定 (当PC停止时,服务时间总为10ms。) |
| DM 6614 | 00 ~ 07 | 串行通信板端口1服务时间(通过位08 ~ 15允许) 00 ~ 99(BCD): 设置用于服务端口1的循环时间的百分比。服务时间必须在0.256ms和65.536ms之间。 |
| | 08 ~ 15 | 串行通信板端口1服务设定 00: 定为循环时间的5% 01: 使用位00 ~ 07中的时间设定 (当PC停止时,服务时间总是10ms。) |

1-2-2 高速计数器板设定

DM 6602, DM 6640 和 DM 6641 中的设定决定了安装在内板槽 1 中的高速计数器板的操作。

DM 6611, DM 6643 和 DM 6644 中的设定决定了安装在内板槽 2 中的高速计数器板的操作。

| 字 | 位 | 功能 | 设定 |
|-----------------|---------|---------------------------|-----------------------|
| DM 6602 (槽1) | 00 ~ 03 | 高速计数器 1 ~ 4PV数据格式 | 0:8位数十六进制 0:8位数BCD |
| | DM 6611 | 未使用 | 设为0 |
| (槽2) | 08 ~ 11 | 外部输出1~4晶体管选择器 | 0:源 1:吸收 |
| | 12 ~ 15 | 未使用 | 设为0 |
| DM 6640 (槽1) | 00 ~ 03 | 高速计数器1输入模式 | 见注1 |
| | DM 6643 | 高速计数器1计数频率, 数值范围和计数器复位模式。 | 见注2 |
| (槽2) | 08 ~ 11 | 高速计数器2输入模式 | 见注1 |
| | 12 ~ 15 | 高速计数器2计数频率, 数值范围和计数器复位模式。 | 见注2 |
| DM 6641 (槽1) | 00 ~ 03 | 高速计数器3输入模式 | 见注1 |
| | DM 6644 | 高速计数器3计数频率, 数值范围和计数器复位模式。 | 见注2 |
| (槽2) | 08 ~ 11 | 高速计数器4输入模式 | 见注1 |
| | 12 ~ 15 | 高速计数器4计数频率, 数值范围和计数器复位模式。 | 见注2 |

注 1. 高速计数器输入模式设定如下:

| 设定 | 输入模式 |
|----|-----------|
| 0 | 相位差输入, 1× |
| 1 | 相位差输入, 2× |
| 2 | 相位差输入, 4× |
| 3 | 增/减输入 |
| 4 | 脉冲/方向输入 |

2. 高速计数器计数频率, 数值范围和计数器复位模式设定如下:

| 设定 | 计数频率 | 数值范围 | 复位模式 |
|----|---------|------|-----------|
| 0 | 50 kHz | 线性计数 | Z相 + 软件复位 |
| 1 | | | 只有软件复位 |
| 2 | | 环形计数 | Z相 + 软件复位 |
| 3 | | | 只有软件复位 |
| 4 | 500 kHz | 线性计数 | Z相 + 软件复位 |
| 5 | | | 只有软件复位 |
| 6 | | 环形计数 | Z相 + 软件复位 |
| 7 | | | 只有软件复位 |

1-2-3 脉冲 I/O 板设定

DM 6611, DM 6643 和 DM 6644 中的设定决定了安装在内板槽 2 中的脉冲 I/O 板操作。(脉冲 I/O 板不能安装在槽 1 中。)

| 字 | 位 | 功能 |
|---------|---------|---|
| DM 6611 | 00 ~ 15 | 槽 1 和 2 模式设定 0000: 高速计数器模式 0001: 简单定位模式 |
| DM 6643 | 00 ~ 03 | 端口 1 输入模式 0: 相位差模式 1: 脉冲/方向模式 2: 增/减模式 |
| | 04 ~ 07 | 端口 1 计数器复位方法 0: Z 相和软件复位; 1: 只有软件复位 |
| | 08 ~ 11 | 端口 1 数值范围 0: 线性计数; 1: 环形计数 |
| | 12 ~ 15 | 端口 1 脉冲输出占空率 0: 固定占空率; 1: 可变占空率 |
| DM 6644 | 00 ~ 03 | 端口 2 输入模式 0: 相位差模式 1: 脉冲/方向模式 2: 增/减模式 |
| | 04 ~ 07 | 端口 2 计数器复位方法 0: Z 相和软件复位; 1: 只有软件复位 |
| | 08 ~ 11 | 端口 2 数值范围 0: 线性计数; 1: 环形计数 |
| | 12 ~ 15 | 端口 2 脉冲输出占空率 0: 固定占空率; 1: 可变占空率 |

1-2-4 绝对编码器接口板设定

DM 6611, DM 6612, DM 6643 和 DM 6644 中的设定决定了安装在内板槽 2 中的绝对编码器接口板的操作。(绝对编码器接口板不能安装在槽 1 中。)

| 字 | 位 | 功能 |
|---------|---------|--|
| DM 6611 | 00 ~ 15 | 端口 1 原点补偿 (4 位数 BCD) 端口 1 原点补偿位 (SR25201) 置 ON 时将补偿原点。如果计数器设为 BCD 模式或 360° 模式, 补偿值将以 BCD 的形式在 0000 到 4095 之间记录。 |
| DM 6612 | 00 ~ 15 | 端口 2 原点补偿 (4 位数 BCD) 端口 2 原点补偿位 (SR25202) 置 ON 时将补偿原点。如果计数器设为 BCD 模式或 360° 模式, 补偿值将以 BCD 的形式在 0000 到 4095 之间记录。 |
| DM 6643 | 00 ~ 07 | 端口 1 输入分辨率 00: 8 位; 01: 10 位; 02: 12 位 |
| | 08 ~ 15 | 端口 1 操作模式 00: BCD 模式; 01: 360° 模式 |
| DM 6644 | 00 ~ 07 | 端口 2 输入分辨率 00: 8 位; 01: 10 位; 02: 12 位 |
| | 08 ~ 15 | 端口 2 操作模式 00: BCD 模式; 01: 360° 模式 |

1-2-5 模拟 I/O 板设定

DM 6611 中的设定决定了安装在内板槽 2 中的模拟 I/O 板的操作。(模拟 I/O 板不能安装在槽 1 中。)

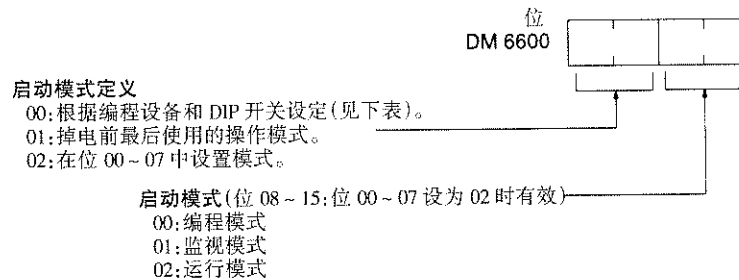
| 字 | 位 | 功能 | 设定 |
|---------|---------|-------------|--|
| DM 6611 | 00 ~ 01 | 模拟输入1输入信号范围 | 设置两个位的位状态如下： 00: -10 ~ +10V 01: 0 ~ 10V 10: 0 ~ 5V 或 0 ~ 20mA |
| | 02 ~ 03 | 模拟输入2输入信号范围 | |
| | 04 ~ 05 | 模拟输入3输入信号范围 | |
| | 06 ~ 07 | 模拟输入4输入信号范围 | |
| | 08 | 模拟输入1使用选择 | 0: 支持(使用)输入 1: 不支持输入 |
| | 09 | 模拟输入2使用选择 | |
| | 10 | 模拟输入3使用选择 | |
| | 11 | 模拟输入4使用选择 | |
| | 12 ~ 15 | 未使用 | 设为0 |

1-3 基本 PC 操作和 I/O 处理

本节描述了与基本操作和 I/O 处理有关的 PC 设置设定。

1-3-1 启动模式

上电时 PC 启动的工作模式可如下设置。



缺省: 操作模式由编程设备和 DIP 开关设定决定, 如下表所示。

| 启动时连接编程设备 | CPU单元的DIP开关7脚 | 启动模式 |
|-----------|---------------|------------------|
| 未连接 | OFF | 编程模式 |
| | ON | 运行模式 |
| 连接编程器 | OFF | 在编程器模式开关上设置的操作模式 |
| | ON | 编程模式(见注1) |
| 连接其它编程设备 | OFF | 编程模式(见注1) |
| | ON | 根据使用的连接电缆(见注2) |

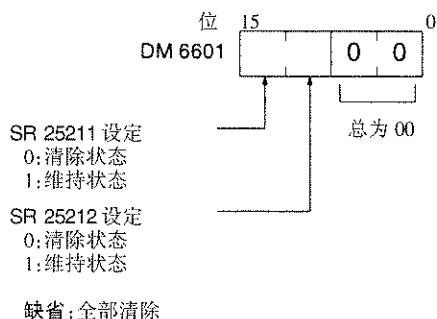
注 1. 在这些情况下, CQM1H 不能与所连接的编程设备通信。

2. 启动模式将根据使用的连接电缆为编程模式或运行模式。

| 连接电缆 | 启动模式 |
|-------------------------------------|------|
| CS1W - CN114 + CQM1 - CIF01/02 | 编程模式 |
| CS1W - CN118 + XW2Z - 200/500S(-V) | 编程模式 |
| CS1W - CN226/626 | 运行模式 |
| CS1W - CN118 + XW2Z - 200/500S - CV | 运行模式 |

1-3-2 保持位状态

进行如下设置以决定在电源通时,强制状态保持位(SR25211)和/或 I/O 保持位(SR25212)是维持上次关电时有效的状态,还是清除以前的状态。

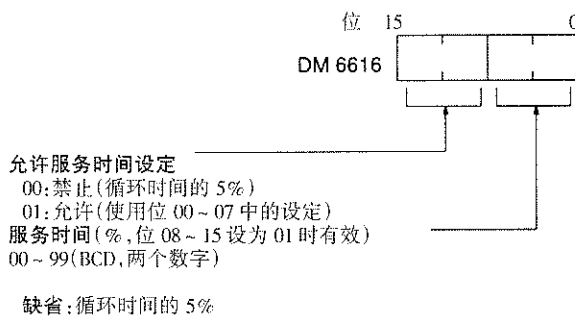


强制状态保持位(SR25211)决定了当从编程模式改变到监视模式时是否维持强制置位/复位状态。

I/O 保持位(SR25212)决定了启动和停止 PC 操作时是否维持 IR 位和 LR 位状态。

1-3-3 RS-232C 端口服务时间

下列设置用于决定服务于 RS-232C 端口的循环时间的百分比。



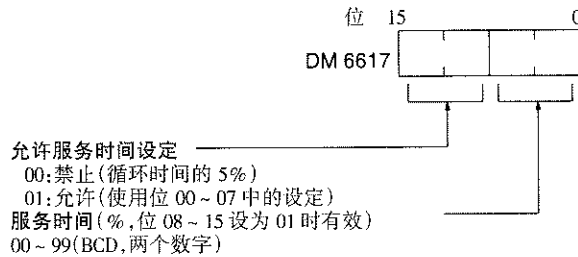
例:如果 DM 6616 设为 0110,服务于 RS-232C 端口将为循环时间的 10%。

最小服务时间是 0.256ms。

除非存在处理请求,否则整个服务时间都不使用。

1-3-4 外围端口服务时间

下列设定用于决定服务于外围端口的循环时间的百分比。



缺省: 循环时间的 5%

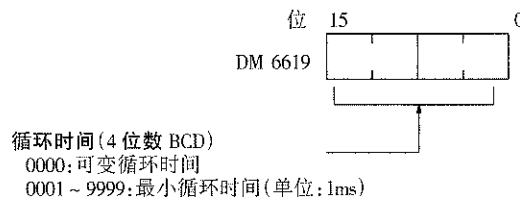
例: 如果 DM 6617 设为 0115, 服务于外围端口将为循环时间的 15%。

最小服务时间为 0.256ms。

除非存在处理请求, 否则整个服务时间将不使用。

1-3-5 最小循环时间

进行如下所示的设置使循环时间标准化, 并通过设置最小循环时间以消除 I/O 响应时间的变化。



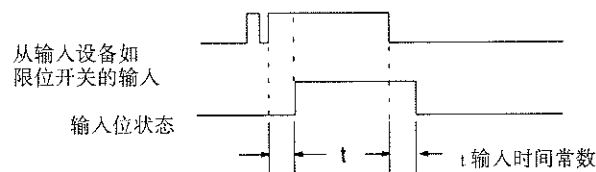
缺省: 可变循环时间

如果实际循环时间小于最小循环时间, 执行将等待, 直到达到最小时间。如果实际循环时间大于最小循环时间, 将根据实际循环时间执行操作。如果超过最小循环时间, AR2405 将置 ON。

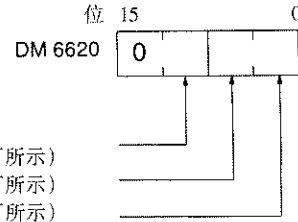
1-3-6 输入时间常数

进行如下所示设置以设置从 DC 输入单元的实际输入置为 ON 或 OFF, 直到相应输入位被刷新 (即, 直到其 ON/OFF 状态改变) 的时间。需要调整输入稳定的时间时, 进行这些设置。

增大输入时间常数可减小振动和外部噪声的影响。

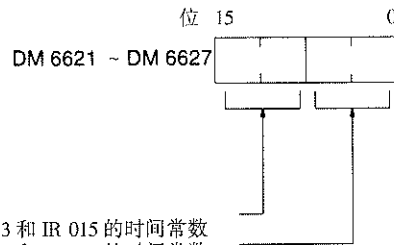


IR 000 和 IR 001 的输入时间常数



IR 00100 ~ IR 00115 的时间常数 (1 位数 BCD; 如下所示)
 IR 00008 ~ IR 00015 的时间常数 (1 位数 BCD; 如下所示)
 IR 00000 ~ IR 00007 的时间常数 (1 位数 BCD; 如下所示)
 缺省: 0000 (都是 8ms)

IR 002 ~ IR 015 的输入时间常数



DM 6621: IR 002 和 IR 003
 DM 6622: IR 004 和 IR 005
 DM 6623: IR 006 和 IR 007
 DM 6624: IR 008 和 IR 009
 DM 6625: IR 010 和 IR 011
 DM 6626: IR 012 和 IR 013
 DM 6627: IR 014 和 IR 015

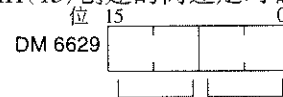
IR 003, IR 005, IR 007, IR 009, IR 011, IR 013 和 IR 015 的时间常数
 IR 002, IR 004, IR 006, IR 008, IR 010, IR 012 和 IR 014 的时间常数
 缺省: 0000 (都是 8ms)

下面列出输入时间常数的 9 个可能设置。仅设置 IR 000 的最右一位数。

- | | | | | |
|---------|---------|---------|----------|--------|
| 0: 8ms | 1: 1ms | 2: 2ms | 3: 4ms | 4: 8ms |
| 5: 16ms | 6: 32ms | 7: 64ms | 8: 128ms | |

1-3-7 高速定时器

进行如下设置以设置用使用中断处理的 TIMH(15) 创建的高速定时器个数。



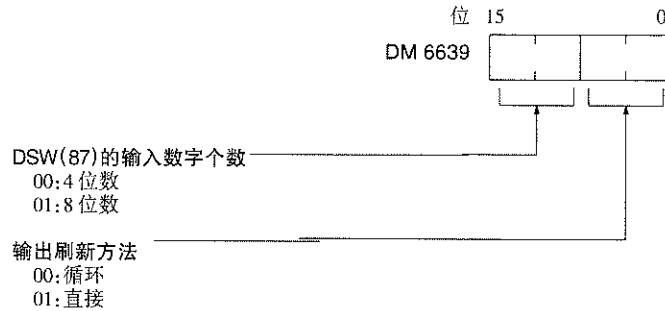
允许高速定时器中断设置
 00: 设置禁止 (TIM000 ~ TIM015 的高速定时器用中断处理)
 01: 允许 (使用位 00 ~ 07 中的设置)
 使用中断的高速定时器 (位 08 ~ 15 为 01 时有效)
 00 ~ 15 (2 位数 BCD)
 缺省: TIM000 ~ TIM015 高速定时器用中断处理

设定指明了从 TIM000 开始, 使用中断处理的定时器数。例如, 如果指定“0108”, 则 TIM000 ~ TIM007 的 8 个定时器将使用中断处理。

- 注
1. 如果没有中断处理, 高速定时器将不准确, 除非循环时间是 10ms 或更小。
 2. 如果使用 SPED(64) 指令, 并且以 500Hz 或以上频率输出脉冲, 则把使用中断处理的高速定时器数设为 4 或更小。详细内容参见 SPED(64) 指令信息。
 3. 在不需要高速定时器处理时, 如果中断处理设为 00, 将增加其它中断的中断反应时间。它包括循环时间小于 10ms 的任何时间。

1-3-8 DSW(87)输入数字和输出刷新方法

进行如下设定以设置 DSW(87) 指令的输入数字个数，并且设置输出刷新方法。



缺省:DSW(87)指令的输入数字个数设为“4”,并且输出刷新方法为循环。

DSW(87) 指令的详细内容参见 415 页, I/O 刷新方法的详细内容参见第 7 章 PC 操作和处理时间。

1-3-9 外围端口设定

外围端口的串行通信设定由 CPU 单元 DIP 开关的 5 和 7 脚, DM 6650 中的十六进制设定, 和连接到外围端口的设备决定。

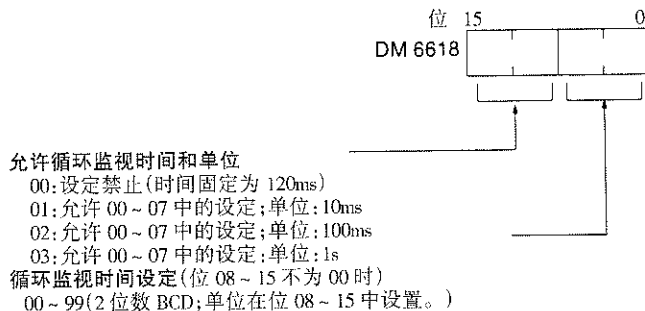
| DIP开关设定 | | DM 6650 设定 | 连接设备 | 串行通信模式 |
|---------|-----|------------|---------------------|---|
| 5脚 | 7脚 | | | |
| OFF | OFF | 忽略 | 编程器 | 编程器总线 |
| OFF | ON | 0000 | 编程设备而不是编程器(例如个人计算机) | 上位机链接, 标准设定外围总线模式(如果CX-Programmer设置为外围总线) |
| | | 0001 | | 上位机链接, 用户设定外围总线模式(如果CX-Programmer设置为外围总线) |
| | | 10□□ | | 无协议 |
| ON | OFF | 忽略 | 编程器 | 编程器总线 |
| ON | ON | 忽略 | 编程设备而不是编程器(例如个人计算机) | 上位机链接, 标准设定外围总线模式(如果CX-Programmer设置为外围总线) |

1-3-10 错误日志设定

进行下列设置以检测错误并存储错误日志。

循环监视时间
(DM 6618)

循环监视时间用于检测超长循环时间,它可在程序进入无限循环时发生。如果循环时间超过循环监视设定,将产生致命错误(FALS 9F)。



缺省: 120ms

注 1. 根据 DM 6618 中的循环监视时间的单位设置,用于最大和当前循环时间的单位记录在 AR 26 和 AR 27(4 位数 BCD)中,显示如下:

位 08~15 设为 01:0.1ms

位 08~15 设为 02:1ms

位 08~15 设为 03:10ms

2. 如果循环时间是 1s 或更长,从编程设备读出的循环时间将是 999.9ms。正确的最大和当前循环时间将记录在 AR 区中。

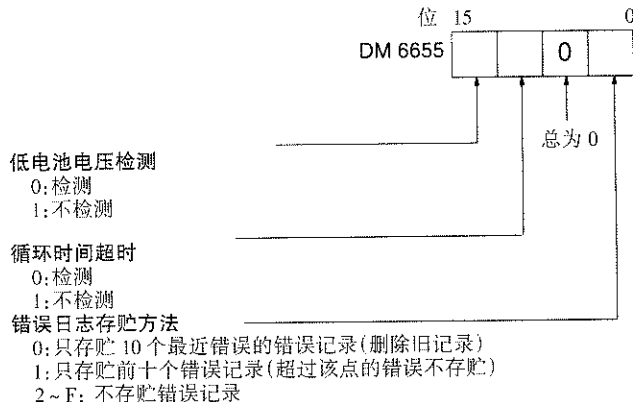
例

如果在 DM 6618 中设置了 0230,不会出现 FALS 9F 错误,直到循环时间超过 3s。如果实际循环时间是 2.59s,AR 区中存储的当前循环时间将为 2590(ms),但从编程设备中读出的循环时间将是 999.9ms。

当循环时间超过 100ms 时,将产生“循环时间超时”错误(非致命),除非使用 DM6655 中设定的长循环时间检测被禁止。

错误检测和错误日志操作 (DM6655)

进行如下设定以决定当循环时间超过 100ms 时,或当内置电池电压跌落时,是否产生非致命错误,并且设置出现错误时在错误日志里存储记录的方法。



缺省: 检测低电池电压和循环时间超时错误,存储最近 10 个错误的错误记录。

电池错误和循环时间超时运行错误是非致命错误。错误日志的详细内容参见第 8 章 排除故障。

1-4 中断功能

本节描述了 CQM1H 中断功能的设定和使用方法。

1-4-1 中断类型

CQM1H 具有四种类型中断,概述如下。

输入中断:

当从外部源到 CPU 单元位 IR 00000 ~ IR 00003 的某个输入置 ON 时执行中断处理。

间隔定时器中断:

通过精度为 0.1ms 的间隔定时器执行中断处理。

高速计数器中断:

根据内置高速计数器的当前值(PV)执行中断处理。CQM1H CPU 单元具有下列 3 种高速计数器中断。它们都可作为目标值中断或范围比较中断。(PV 与 SV 匹配时产生目标值中断,PV 处于预设置 SV 范围中时产生范围比较中断。)

1,2,3...

1. 高速计数器 0(在 CPU 单元中)

高速计数器 0 对输入到 CPU 单元输入 4~6 的脉冲计数,两相脉冲最多可计到 2.5KHz。

2. 高速计数器 1 和 2(脉冲 I/O 板)

高速计数器 1 和 2 计算输入到脉冲 I/O 板上的端口 1 和 2 的高速脉冲。两相脉冲最多可计到 25KHz。

3. 绝对高速计数器 1 和 2(绝对编码器接口板) 高速计数器 1 和 2 计算输入到绝对编码器接口板上的端口 1 和 2 的绝对旋转编码器代码。

注 高速计数器板上的高速计数器 1,2,3 和 4 不执行中断处理。高速计数器板可计算达到 50KHz 或 500KHz 的脉冲。高速计数器 PV 可根据目标值或 SV 范围检查,并且位型式可由内部或外部输出并不产生中断。

串行通信板中断:

当串行通信板接收所需信息时,CPU 单元请求中断处理。

产生中断时,执行指定的中断子程序。

定义子程序

如同普通子程序一样,在主程序结尾使用 SBN(92)和 RET(93)定义中断子程序。

执行中断子程序时,指定的输入位范围可以被刷新。

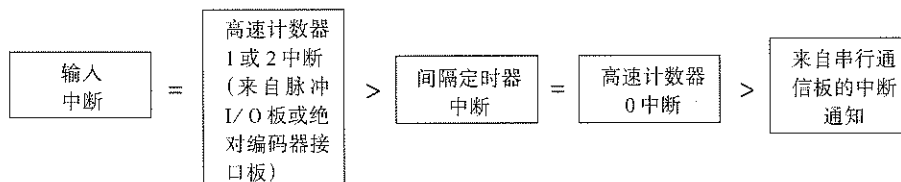
定义中断子程序后,进行程序检查时将产生“无 SBS 错误”,但处理会照常执行。如果出现这一错误,检查所有一般的子程序,确认在执行前 SBS(91)已经编入程序。

中断优先级

中断具有下列优先级顺序,输入中断和来自高速计数器 1 和 2 的中断具有最

中断处理

高的优先级,来自串行通信板的中断通知具有最低优先级。



在处理中断过程中接收到一个更高优先级的中断时,将停止当前处理并且将处理新接收的中断。在该程序完全执行后,前一个中断处理将继续。

在处理中断过程中接收到更低或相等优先级的中断时,当前处理的程序执行完成后就将处理新接收的中断。

如果两个相同优先级的中断同时出现,中断将以下列顺序执行:

- 1, 2, 3... 1. 输入中断 0 > 输入中断 1 > 输入中断 2 > 输入中断 3 > 高速计数器中断 1 > 高速计数器中断 2
2. 间隔定时器中断 0 > 间隔定时器中断 1 > 间隔定时器中断 2 (间隔定时器中断 2 是高速计数器中断 0。)

脉冲输出指令和中断

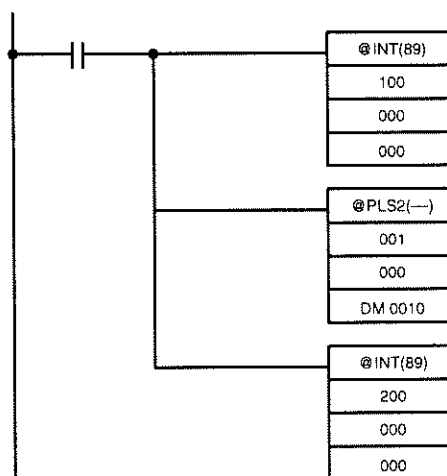
当在主程序中执行控制脉冲 I/O 或高速计数器的指令时,不能在中断子程序中执行下列指令:(SR25503 置 ON)

INI(89), PRV(62), CTBL(63), SPED(64), PULS(65), PWM(—), PLS2(—)和 ACC(—)

下列方法可用来绕过此限制:

方法 1

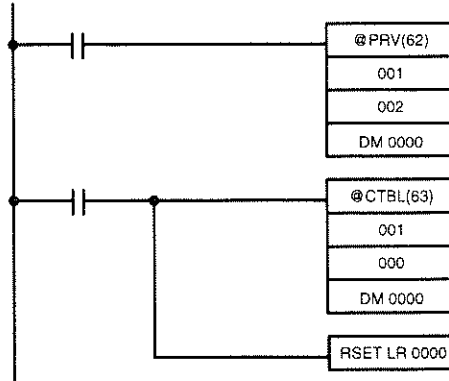
执行指令时可屏蔽所有中断处理。



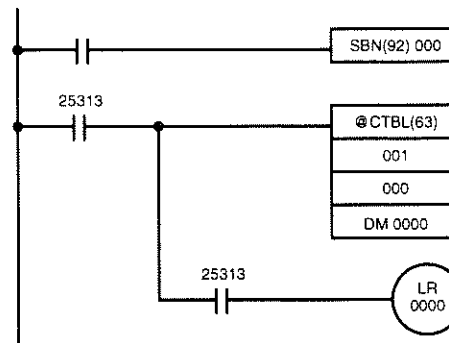
方法 2

在主程序中再次执行指令。

这是主程序的程序段：



这是中断子程序的程序段：



1-4-2 输入中断

CPU 单元的输入分配 IR 00000 ~ IR 00003 可用于外部源的中断。输入中断 0 ~ 3 分别与这些位对应,并且总是用于分别调用 000 ~ 003 的子程序。不使用输入中断时,子程序 000 ~ 003 可用于普通子程序。

处理

有两种处理输入中断的模式。第一种是输入中断模式,这种中断的进行响应外部输入。第二种是计数器模式,来自外部源的信号高速计数,每一定数目的信号都进行一次中断

INT(89)指令决定使用哪种模式。

输入中断模式中,可检测长度为 100 μ s 或更快的信号。在计数器模式中,可计算最高 1kHz 的信号。

过程

(输入中断模式)

用输入中断模式使用输入中断时,依照下列步骤。

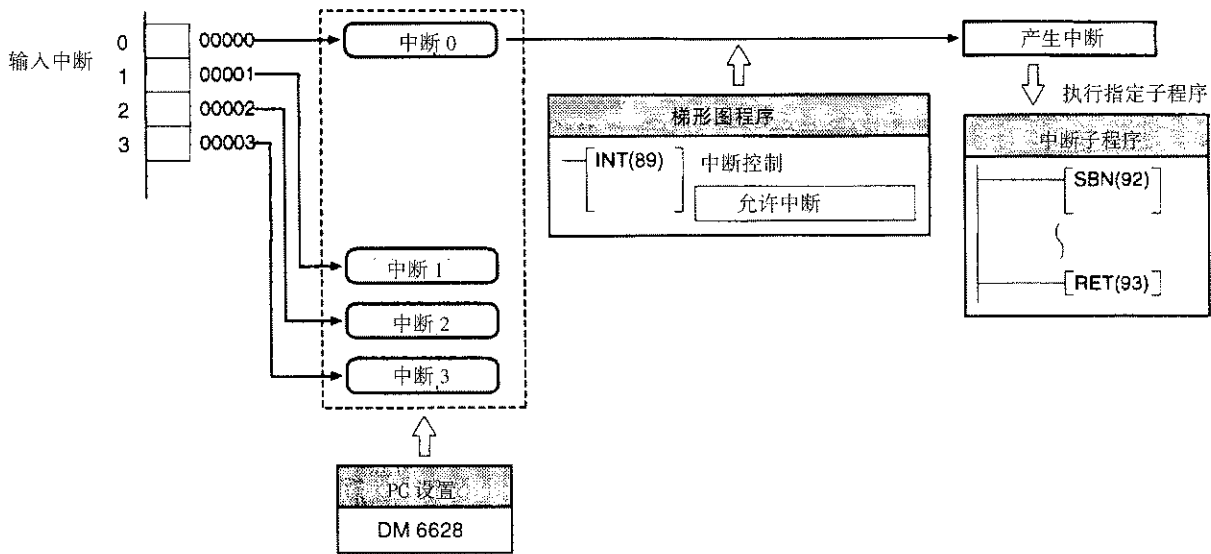
- 1,2,3... 1. 决定输入中断号。

| 端子 | | 相应位地址 | 子程序号 |
|----|-----|----------|------|
| B0 | IN0 | IR 00000 | 000 |
| A0 | IN1 | IR 00001 | 001 |
| B1 | IN2 | IR 00002 | 002 |
| A1 | IN3 | IR 00003 | 003 |

2. 输入接线。(详细内容见 21 页。)

3. 进行 PC 设置设定。(详细内容见 22 页。)

- a)在 DM 6628 相应数字位中写入 1 以表明此输入将用作输入中断(输入中断或计数器模式。)
 - b)DM 6630 ~ DM 6633 中的位置 ON,使得执行中断子程序前刷新输入。
4. 编写有关的程序段。
- a)使用 INT(89)对输入中断去屏蔽。(详细内容参见 23 页。)
 - b)在 SBN(92)和 RET(93)之间编写中断子程序。



过程
(计数器模式)

使用计数器模式的输入中断时,依照下列步骤。

- 1,2,3... 1. 决定输入中断号。

| 端子 | | 相应位地址 | 子程序号 |
|----|-----|----------|------|
| B0 | IN0 | IR 00000 | 000 |
| A0 | IN1 | IR 00001 | 001 |
| B1 | IN2 | IR 00002 | 002 |
| A1 | IN3 | IR 00003 | 003 |

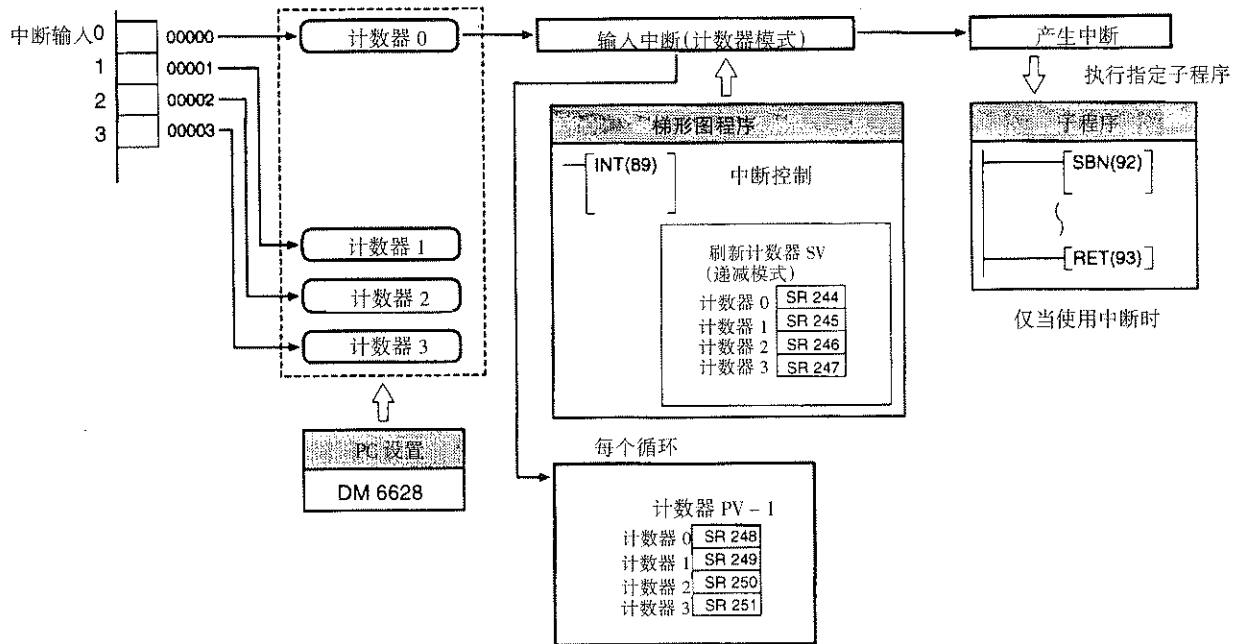
- 2. 决定初始计数 SV。
- 3. 输入接线(详细内容参见 21 页。)
- 4. 进行 PC 设置设定(详细内容参见 22 页。)

 - a)在 DM 6628 相应数字位中写入 1 以表明此输入将用作输入中断(输入中断或计数器模式。)
 - b)DM 6630 ~ DM 6633 的位置 ON,使得执行中断子程序前刷新输入。

- 5. 编写有关的程序段。

 - a)使用 INT(89)用计数器模式刷新计数器 SV(详细内容参见 24 页。)

b) 在 SBN(92) 和 RET(93) 之间编写中断子程序 (仅当使用计数到中断时。)

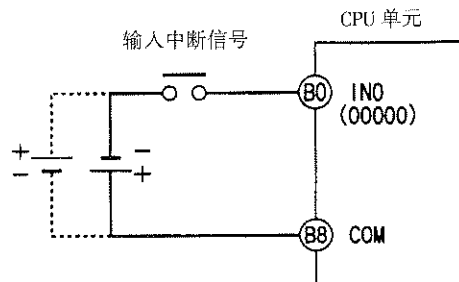


输入接线

使用输入中断前, 为 CPU 的输入端子接入输入中断信号或统计输入信号, 如下所示。

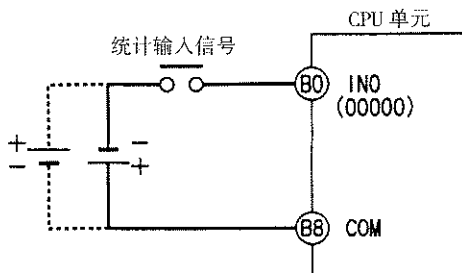
中断输入信号 (输入中断模式) 接线举例

| 端子 | 相应位地址 |
|---------|----------|
| B0(IN0) | IR 00000 |
| A0(IN1) | IR 00001 |
| B1(IN2) | IR 00002 |
| A1(IN3) | IR 00003 |



统计输入信号(计数器模式)接线举例

| 端子 | 相应位地址 | 递减模式 |
|---------|----------|--------------|
| B0(IN0) | IR 00000 | 脉冲输入(最多4个输入) |
| A0(IN1) | IR 00001 | |
| B1(IN2) | IR 00002 | |
| A1(IN3) | IR 00003 | |

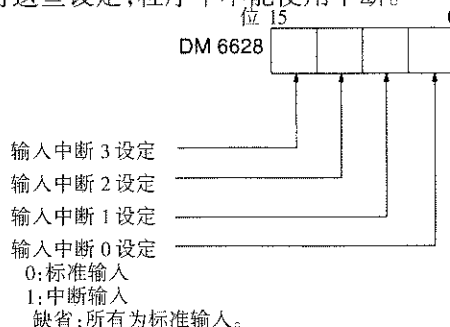


PC 设置参数

执行程序前,把 PC 设置中的下列设定设为编程模式。

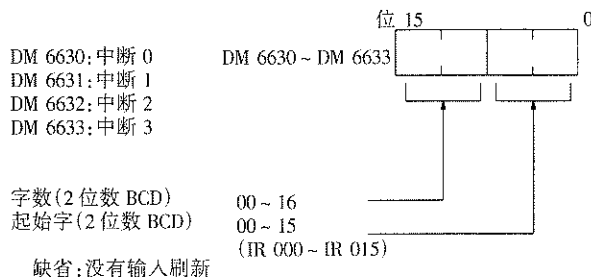
中断输入设定 (DM 6628)

如果不进行这些设定,程序中不能使用中断。



输入刷新字设定 (DM 6630 ~ DM 6633)

在需要为输入中断或计数器模式刷新输入时进行这些设定。



例

如果 DM 6630 设为 0100,在中断 0 接收到信号时 IR 000 将被刷新。

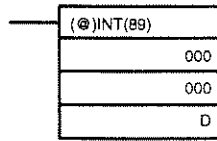
注 如果不用输入刷新,中断程序中的输入信号状态将不可靠,它甚至包括激活中断的中断输入位的状态。例如,在输入中断 0 的中断程序中,IR 00000 不会置 ON 除非它被刷新(在这种情况下,可使用常开标志 SR 25313 而不是 IR 00000)。

输入中断模式

使用下列指令以输入中断模式编写输入中断程序。

中断屏蔽

用 INT(89)指令, 按需要设置或清除输入中断屏蔽。



用对应于输入中断 0~3 的 D 位 0~3 进行设定。
0:清除屏蔽(允许输入中断)
1:设置屏蔽(不允许输入中断)

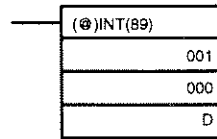
运行开始时,所有的输入中断都被屏蔽。在以输入中断模式使用输入中断前使用 INT(89)使输入中断去除屏蔽。

清除屏蔽的中断

如果在屏蔽时对应于输入中断的位置 ON, 则在内存中保存该输入中断并将在清除屏蔽时执行。为使清除屏蔽时不执行该输入中断,必须从内存中清除中断。

内存中每个中断号只能保存一个中断信号。

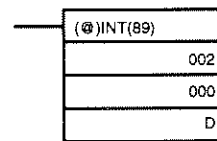
用 INT(89)指令,从内存中清除输入中断。



如果时应于输入中断 0~3 的 D 位 0~3 设为“1”,
则输入中断将从内存中清除。
0:保留输入中断
1:清除输入中断

读入屏蔽状态

用 INT(89)指令,读入输入中断屏蔽状态。



存储在字 D(位 0~3) 中的数据的最右数字位的状
态显示了屏蔽状态。
0:清除屏蔽(允许输入中断)
1:设置屏蔽(不允许输入中断)

计数器模式

使用下列步骤以计数器模式编写输入中断程序。

注 计数器模式中使用的 SR 字 (SR244~SR 251) 都包含二进制(十六进制)数据(不是 BCD)。

- 1,2,3... 1. 在 SR 字中写入相应于中断 0~3 的计数器操作设置值。写入在 0000 和 FFFF(0~65535)之间的设置值。值 0000 将禁止计数操作,除非设置新值并且重复下面的步骤 2。

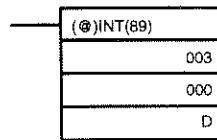
注 操作开始时这些 SR 位被清除,并且必须从程序中写入。

可计数的最快输入信号是 1kHz。

| 中断 | 包含计数器SV的字 |
|-------|-----------|
| 输入中断0 | SR 244 |
| 输入中断1 | SR 245 |
| 输入中断2 | SR 246 |
| 输入中断3 | SR 247 |

如果不使用计数器模式,这些 SR 位可用作工作位。

2. 用 INT(89)指令,刷新计数器模式设置值并且允许中断。



如果对应于输入中断 0~3 的 D 位 0~3 设为“0”,则设置值将被刷新并且允许中断。
 0:刷新计数器模式设置值并且清除屏蔽
 1:无操作(所有不改变的中断位设为 1)

刷新设置值的输入中断将在计数器模式中允许计数器达到设置值时,将出现中断,计数器将复位,并且计数/中断将继续直到计数器停止。

- 注**
1. 如果在计数时使用 INT(89)指令,当前值(PV)将返回设置值(SV),所以必须使用指令的微分形式,否则不会出现中断。
 2. 执行 INT(89)指令时设置值被设置,如果正在执行中断,则不改变设置值而仅改变 SR244~SR247 的内容。即,如果改变内容,必须通过再次执行 INT(89)指令刷新设置值。

可使用与输入中断模式相同的处理来屏蔽中断,但是如果用相同处理清除了屏蔽,则不保持计数器模式,相反使用输入中断模式。也可用与输入中断模式相同的处理清除屏蔽中断时接收的中断信号。

计数器模式的计数器 PV

如果输入中断使用计数器模式,计数器 PV 将存贮在对应于输入中断 0~3 的 SR 字中。值为 0000~FFFE(0~65534)并且等于计数器 PV 减 1。

| 中断 | 包含计数器PV-1的字 |
|-------|-------------|
| 输入中断0 | SR 248 |
| 输入中断1 | SR 249 |
| 输入中断2 | SR 250 |
| 输入中断3 | SR 251 |

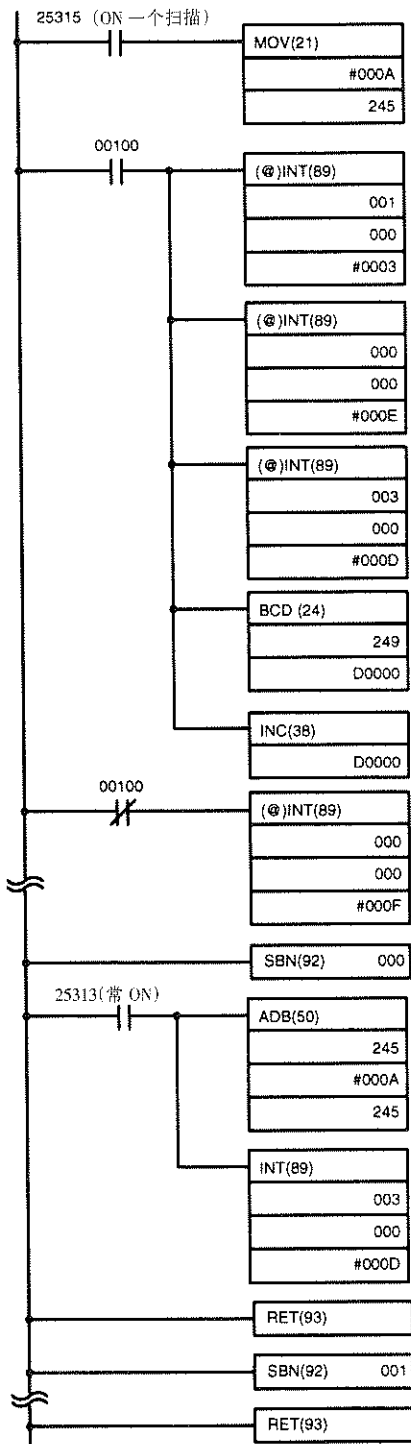
例 执行 INT(89)后立即把设定值为 000A 的中断的当前值记录为 0009。

注 即使输入中断不使用计数器模式,这些 SR 位也不能用作工作位。

应用例

在此例中,输入中断 0 使用输入中断模式,输入中断 1 使用计数器模式。执行程序前,检查确认 PC 设置正确。

PC 设置:DM 6628:0011(IR 00000 和 IR 00001 用于输入中断)缺省设置用于所有其它的 PC 设置参数。(在处理中断时不刷新输入。)



给输入中断 1 设为 10 作为计数器模式 SV。

IR 00100 置 ON 时:
清除输入中断 0 和 1 的屏蔽中断。

中断 0 允许输入中断模式的中断。

中断 1 允许计数器模式的中断。
(SV: 10)

SR 249 的内容(PV - 1)转换为 BCD 值并存储在 DM 0000 中。

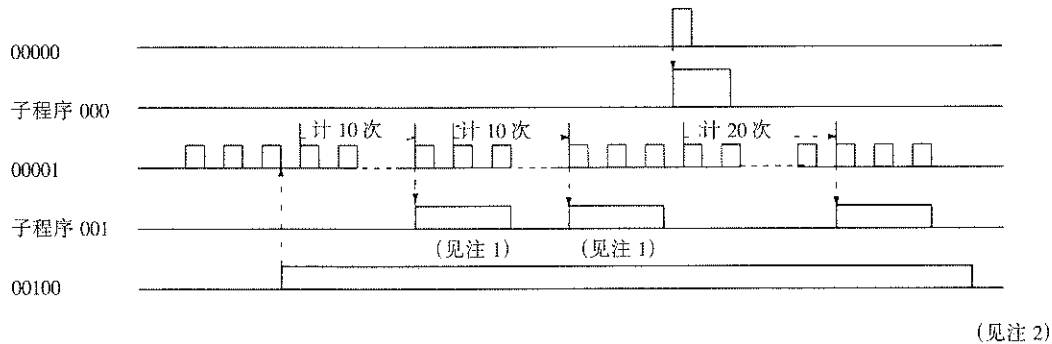
DM 0000 的内容加到 PC 中。

IR 00100 置 OFF 时,输入中断 0 和 1 被屏蔽并且中断被禁止。

当中断 0 执行输入中断时,调用子程序 000,并且使输入中断 1 的 SV 加 10 刷新计数器模式(SV = 20)

当计数达到输入中断 1 计数器,调用子程序 001 并且执行中断子程序。

程序执行时,操作如下图所示。



- 注 1. 在执行中断程序时计数器也能继续操作。
- 2. 输入中断保持被屏蔽。

1-4-3 屏蔽所有中断

INT(89) 指令可用于作为一组屏蔽和解除屏蔽所有中断,包括输入中断,间隔定时器中断和高速计数器中断。此屏蔽附加于单个中断类型的屏蔽。而且,清除所有中断的屏蔽不清除单个中断类型的屏蔽,只是把它们恢复到在执行 INT(89) 作为一组屏蔽之前存在的屏蔽条件。

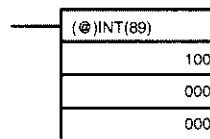
| 用INT(89)屏蔽/解除屏蔽中断 | 源单元或板 |
|-------------------|----------|
| 输入中断 | CPU单元 |
| 间隔定时器中断 | |
| 高速计数器0中断 | |
| 高速计数器1和2中断 | 脉冲I/O板 |
| 高速计数器1和2中断 | 绝对编码器接口板 |

不要使用 INT(89) 屏蔽中断,除非必须临时屏蔽所有中断,并且一定要成对使用 INT(89) 指令,使用第一个 INT(89) 指令屏蔽中断,第二个 INT(指令)解除屏蔽中断。

INT(89)不能用于从中断程序中屏蔽和解除屏蔽所有中断。

屏蔽中断

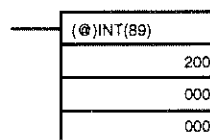
使用 INT(89)指令禁止所有中断。



如果在屏蔽中断时产生中断,不执行中断处理,但中断将被记录为输入、间隔定时器 and 高速计数器中断。中断解除屏蔽时就将服务中断。

解除屏蔽中断

使用 INT(89)指令如下解除屏蔽中断:



1-4-4 间隔定时器中断

使用间隔定时器可执行高速、高精度定时器中断处理。CQMIH 提供三个间隔定时器, 号码从 0 到 2。

注 1. 用 SPED(64) 指令把脉冲输出到晶体管输出单元时不能使用间隔定时器 0。

2. 间隔定时器 2 不能与高速计数器 0 同时使用。

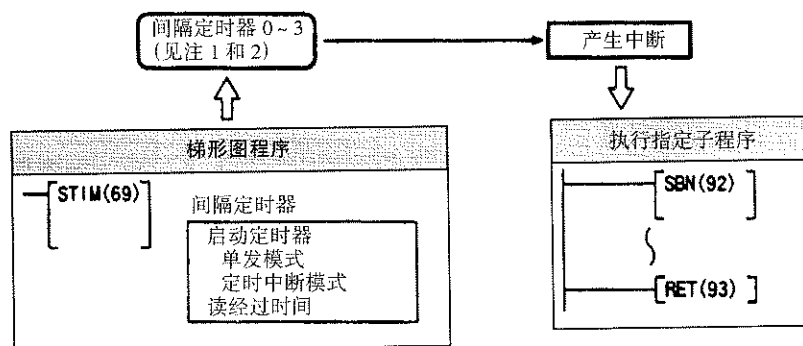
处理

间隔定时器有两种操作模式: 单发模式, 到时间时只执行一次中断; 定时中断模式, 以固定间隔重复中断。

步骤

使用间隔定时器中断时按照下列步骤。

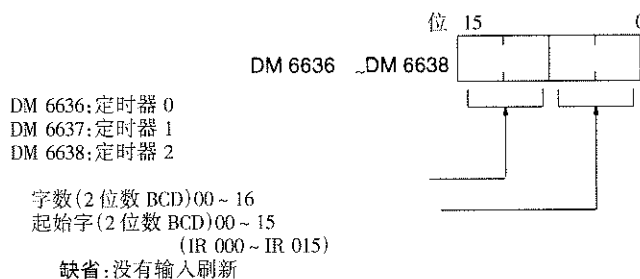
- 1,2,3... 1. 决定定时器以单发模式还是定时中断模式操作。
 2. 编写相关程序段。
 a) 使用 STIM(69) 设置定时器 SV, 并以单发或定时中断模式启动定时器。
 b) 在 SBN(92) 和 RET(93) 之间编写中断子程序。



- 注 1. 不能同时使用间隔定时器 2 和高速计数器 0。
 2. 用 SPED(64) 从晶体管输出单元输出脉冲时不能同时使用间隔定时器 0。
 使用间隔定时器中断时, 在执行程序前用编程模式在 PC 设置中进行如下设置。

PC 设置

输入刷新字设定 (DM 6636 ~ DM 6638)
 必须刷新输入时进行这些设置。



高速计数器设定 (DM 6642)

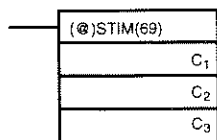
使用间隔定时器 2 时, 在开始操作前进行检查以确保高速计数器 (PC 设置: DM 6642) 设置为缺省设定 (0000: 不使用高速计数器)。

操作

使用下列指令激活并控制间隔定时器。

用单发模式启动

使用 STIM(69)指令以单发模式启动间隔定时器。



- C₁: 间隔定时器号
 间隔定时器 0: 000
 间隔定时器 1: 001
 间隔定时器 2: 002
- C₂: 定时器设定值(首字地址或常数)
- C₃: 子程序号(4 位数 BCD):0000 ~ 0255

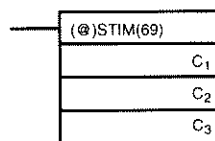
| 字 | 功能 |
|--------------------|---|
| C ₂ | 递减计数器设定值(4位数BCD):0000 ~ 9999 |
| C ₂ + 1 | 递减时间间隔(4位数BCD;单位:0.1ms):0005 ~ 0320(0.5ms ~ 32ms) 注 如果常数用于C ₂ ,递减时间间隔固定为0010或1ms,所以C ₂ 中的设定值以ms表示。 |

每次字 C₂ + 1 中指定的间隔结束时,递减计数器将把当前值递减 1。PV 达到 0 时,调用一次指定的子程序并且定时器停止。

当字地址用于 C₂,从 STIM(69)指令执行到时间用完的时间如下计算:(字 C₂ 的内容) × (字 C₂ + 1 的内容) × 0.1ms = (0.5 ~ 319968ms)

用定时中断模式启动

用 STIM(69)指令以定时中断模式启动间隔定时器。



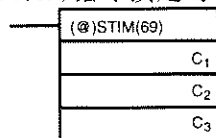
- C₁: 间隔定时器号 + 3
 间隔定时器 0: 003
 间隔定时器 1: 004
 间隔定时器 2: 005
- C₂: 定时器设定值(首字地址或常数)
- C₃: 子程序号(4 位数 BCD):0000 ~ 0255

| 字 | 功能 |
|--------------------|--|
| C ₂ | 递减计数器设定值(4位数BCD):0000 ~ 9999 |
| C ₂ + 1 | 递减时间间隔(4位数BCD;单位:0.1ms):0005 ~ 0320(0.5ms ~ 32ms) 注 如果C ₂ 用常数,递减时间间隔固定为0010或1ms,所以C ₂ 中的设定值以ms表示。 |

设定的意义与单发模式一样,但在定时中断模式中,定时器 PV 将复位为设定值,并且在调用子程序后重新开始递减。在定时中断模式中,连续以固定间隔重复中断,直到停止操作。

读定时器经过的时间

用 STIM(69)指令读定时器经过的时间。



- C₁: 间隔定时器号 + 6
 间隔定时器 0: 006
 间隔定时器 1: 007
 间隔定时器 2: 008
- C₂: 参数 1 的首字地址
- C₃: 参数 2

| 字 | 功能 |
|--------------------|---|
| C ₂ | 计数器已经递减的次数(4位数BCD) |
| C ₂ + 1 | 递减计数器时间间隔(4位数BCD;单位:0.1ms) |
| C ₃ | 上次递减后经过的时间(4位数BCD;单位:0.1ms) 注 此值小于递减计数器时间间隔。 |

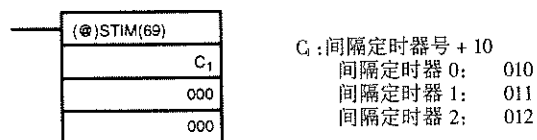
从间隔定时器启动到执行此指令的时间如下计算：

$$[(\text{字 } C_2 \text{ 的内容}) \times (\text{字 } C_2 + 1 \text{ 的内容}) + (\text{字 } C_3 \text{ 的内容})] \times 0.1\text{ms}.$$

如果指定的间隔定时器停止,则存贮“0000”。

停止定时器

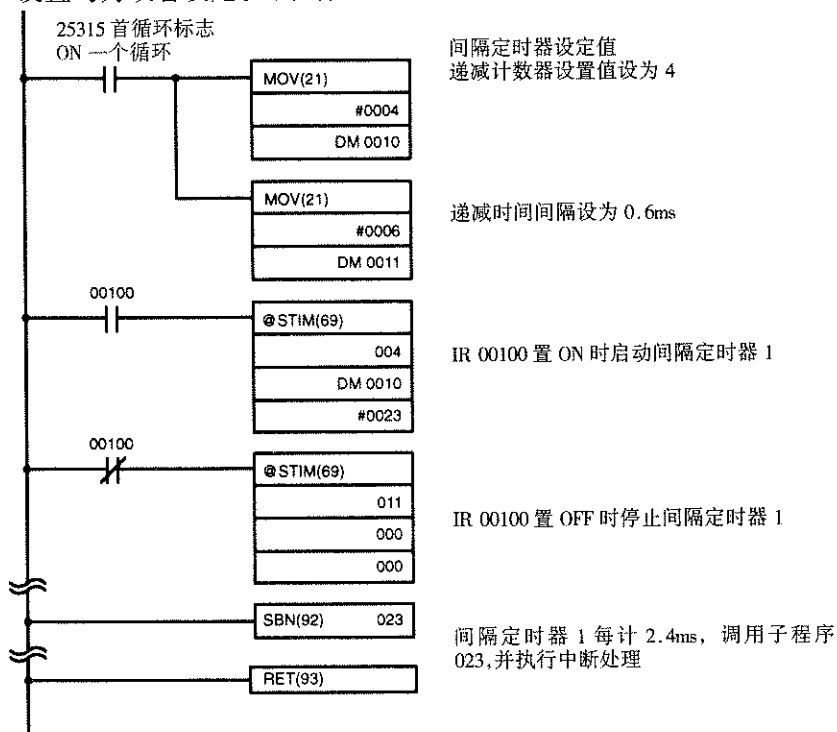
用 STIM(69)指令停止间隔定时器。



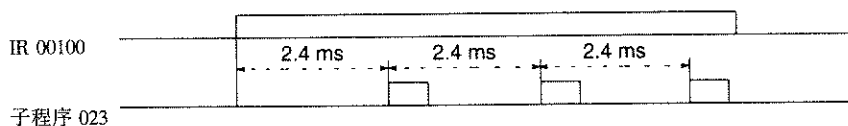
指定的间隔定时器将停止。

应用例

在此例中,用间隔定时器 1 每 2.4ms(0.6ms × 4) 执行一次中断。假定所有 PC 设置均为缺省设定。(中断处理不刷新输入。)



执行程序时,在 IR 00100 置 ON 时每 2.4ms 执行一次子程序 023。



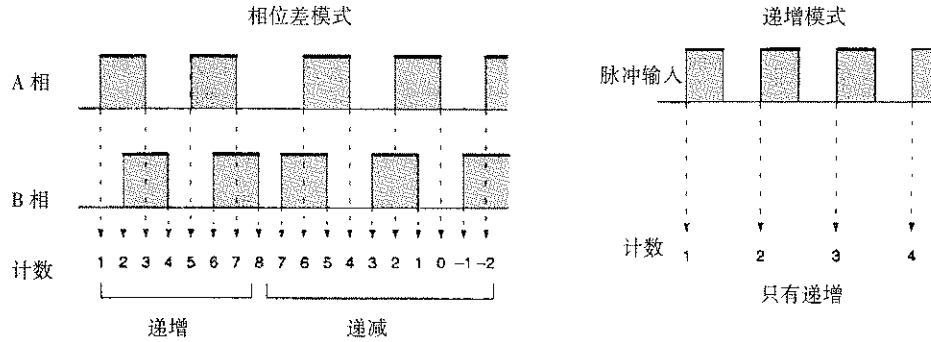
1-4-5 高速计数器 0 中断

脉冲编码器的脉冲信号到 CPU 的位 00004 ~ 00006 可使用高速计数器 0(内置高速计数器)高速计数,并可根据计数执行中断处理。

输入信号类型和输入模式

从脉冲编码器可输入两种信号。用于高速计数器 0 的输入模式由信号类型决定。

| 模式 | 操作 |
|-------|---|
| 相位差模式 | 相位差 $4 \times$ 两相信号(A相和B相)和Z相信号用于输入,根据2相信号的相差递增或递减计数。 |
| 递增模式 | 单相脉冲信号和计数复位信号用于输入。根据单相信号递增计数。 |



注 重新启动计数器时,下段中的一种方法可用于复位计数器,启动或停止程序执行时计数器将自动复位。

下列信号转换作为向前(递增)脉冲处理:A相上升沿到B相上升沿到A相下降沿到B相下降沿。下列信号转换作为反转(递减)脉冲处理:B相上升沿到A相上升沿到B相下降沿到A相下降沿。

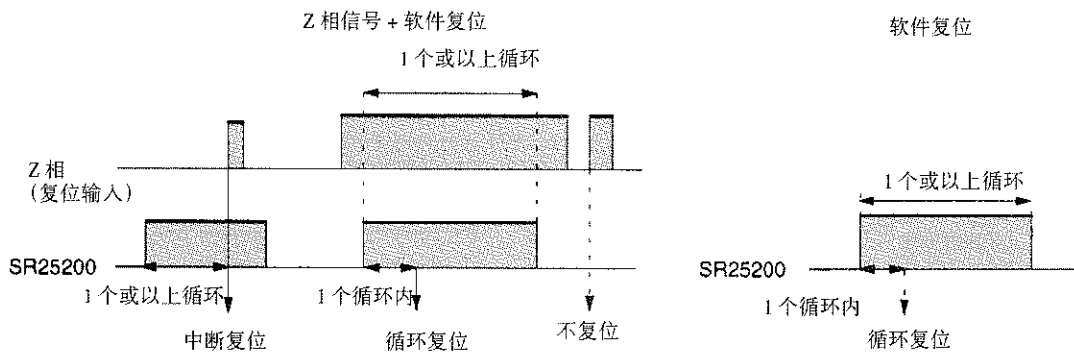
相位差模式的计数范围从-32767到32767,递增模式为0~65535。脉冲信号在相位差模式下最多可计到2.5kHz,在递增模式下最多5.0kHz。

相位差模式常使用 $4 \times$ 相位差的输入。每个编码器计数分辨率是计数器分辨率的4倍。根据可计数的范围选择编码器。

复位方法

下述两种方法都可选择用于复位计数的PV(即,把它设为0)。

| 方法 | 操作 |
|-------------|--|
| Z相信号 + 软件复位 | 高速计数器0复位位(SR 25200)置ON后,当Z相信号(复位输入)置ON时PV复位。 |
| 软件复位 | 高速计数器0复位位(SR 25200)置ON时PV复位。 |



注 高速计数器0复位位(SR 25200)每个循环刷新一次,因此为使它可靠读入,必须置ON至少一个循环。

“Z相”中的“Z”是“零”的缩写。它是表明编码器完成一个循环的信号。

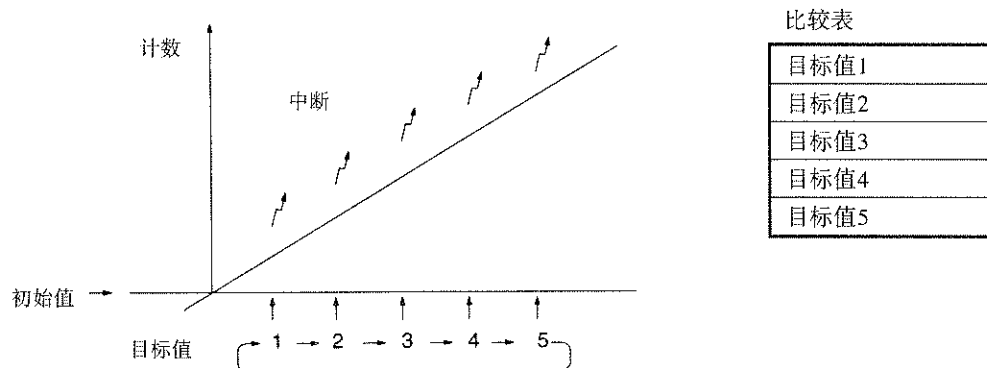
高速计数器 0 中断计数

对于高速计数器 0 中断,使用比较表而不是“计数到”。可用下述的两种方法之一进行计数检查。在比较表中,保存比较条件(与 PV 比较)和中断子程序组合。

| 方法 | 操作 |
|------|---|
| 目标值 | 比较表中最多可保存16个比较条件(目标值和计数方向)和中断子程序组合。当计数器PV和计数方向符合比较条件,则执行指定中断程序。 |
| 范围比较 | 比较表中保存了8个比较条件(上限和下限)和中断程序组合。当PV大于等于下限并小于等于上限时,执行指定中断子程序。 |

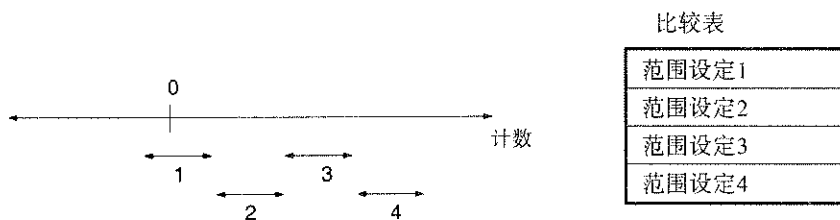
目标值比较

当前计数按照比较表中设置的目标值的顺序与目标值作比较,并且在计数等于每个目标值时产生中断。一旦计数已经等于表中所有的目标值,目标值设置为表中的第一个目标值,它再次与当前计数比较,直到两个值相等。



范围比较

当前计数同时以循环方式与所有范围作比较,并且根据比较结果产生中断。



注 进行目标值比较时,不要重复使用 INI 指令来改变计数当前值并启动比较操作。如果在程序中改变当前值后立即启动比较操作,中断操作将不能正常工作。(一旦最后一个目标值产生了中断,比较操作将自动返回第一个目标值。因此重复操作可能仅改变当前值。

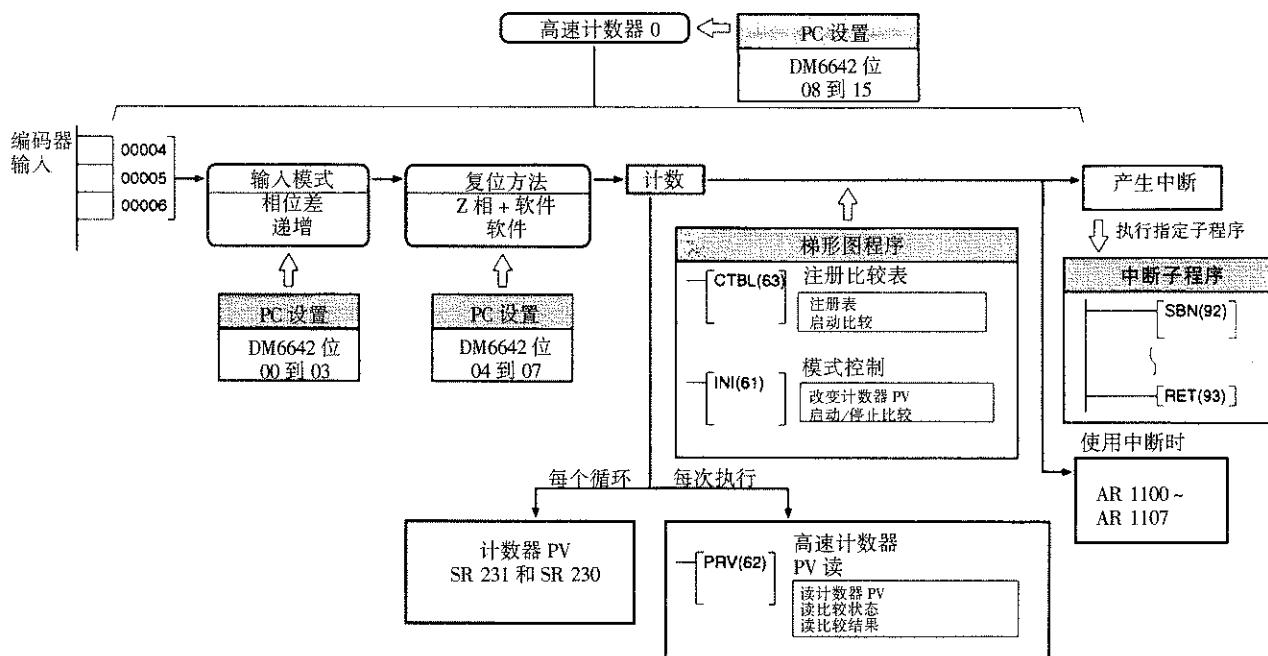
步骤

按下列步骤使用高速计数器 0(CPU 单元的内置高速计数器。)

- 1, 2, 3... 1. 决定使用的输入模式 (相位差模式或递增模式) 和复位方法 (Z 相信号 + 软件复位或软件复位)。
2. 决定中断规格。
 - a) 无中断 (读高速计数器 PV 或范围比较结果。)
 - b) 使用目标值中断或范围比较中断。
3. 输入接线 (详细内容参见 CQM1H 操作手册。)

| 端子 | | 相应位地址 |
|----|-----|----------|
| B2 | IN4 | IR 00004 |
| A2 | IN5 | IR 00005 |
| B3 | IN6 | IR 00006 |

4. 在 DM 6642 中进行 PC 设置设定 (详细内容参见 34 页。)
- a) 最左边的字节设为 01, 表示使用高速计数器 0。
- b) 设置输入模式 (相位差模式或递增模式。)
- c) 设置复位方法 (Z 相信号 + 软件复位, 或软件复位。)
- 注 使用间隔定时器 2 时不能使用高速计数器 0 (DM 6642 的最左边字节的设定决定使用高速计数器 0 或间隔定时器 2。)
5. 编写有关程序段。
 - a) 用 CTBL(63) 注册比较表并启动比较。
 - b) 用 INI(61) 改变高速计数器 PV 或启动比较。
 - c) 用 PRV(62) 读高速计数器 PV, 比较状态, 或比较结果。
 - d) 在 SBN(92) 和 RET(93) 之间写中断子程序 (仅当使用高速计数器 0 中断时。)



下列指令用于控制高速计数器操作。

| 指令 | 控制功能 |
|----------|-----------------------------|
| CTBL(63) | 注册目标值比较表并启动比较。 |
| | 注册范围比较表并启动比较。 |
| | 注册目标值比较表。 (用INI(61)启动比较) |
| | 注册范围比较表。 (用INI(61)启动比较) |
| INI(61) | 用注册比较表启动比较。 |
| | 停止比较。 |
| | 改变高速计数器PV。 |
| PRV(62) | 读高速计数器PV。 |
| | 读范围比较结果。 |

下列标志和控制位用于监视和控制高速计数器操作。

| 字 | 位 | 名称 | 功能 |
|--------|---------|----------------------|---|
| SR 230 | 00 ~ 15 | 高速计数器0PV (最右边4位数) | 包含高速计数器0的当前值 (CPU单元的内置高速计数器。) |
| SR 231 | 00 ~ 15 | 高速计数器0PV (最左边4位数) | |
| SR 252 | 00 | 高速计数器0复位位 | 复位高速计数器0的PV |
| AR 11 | 00 ~ 07 | 高速计数器0范围比较标志 | 表示高速计数器0的范围比较结果。 0: 不满足范围条件 1: 满足范围条件 |

接线

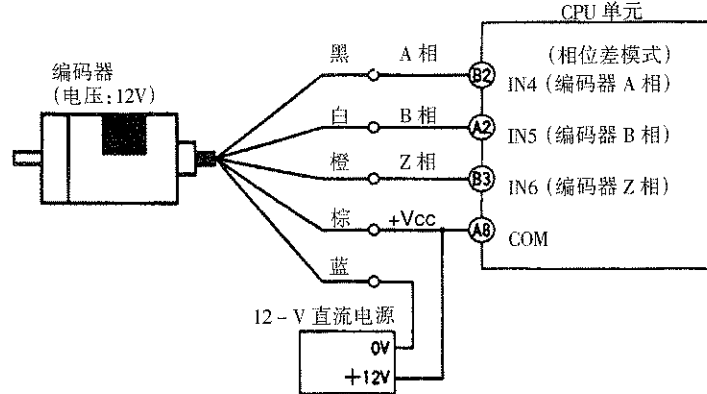
根据输入模式,从脉冲编码器到 CPU 单元输入端子的输入信号如下所示。

| 端子 | 分配的位地址 | 相位差模式 | 递增模式 |
|---------|--------|-------|--------|
| B2(IN4) | 00004 | 编码器A相 | 脉冲计数输入 |
| A2(IN5) | 00005 | 编码器B相 | ... |
| B3(IN6) | 00006 | 编码器Z相 | 复位输入 |

如果使用软件复位,IR00006 可用作普通输入。

- 注
1. 输入模式设为递增模式时,IR00005 可用作普通输入。
 2. 复位方法设为软件复位时,IR00006 可用作普通输入。

下图显示了用 E6B2 - CWZ6C NPN 开集输出的接线举例。

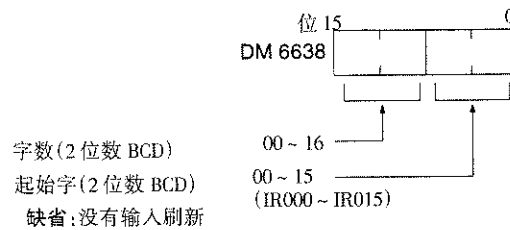


PC 设置

使用高速计数器 0 中断时,在执行程序前,在编程模式下进行如下设置。

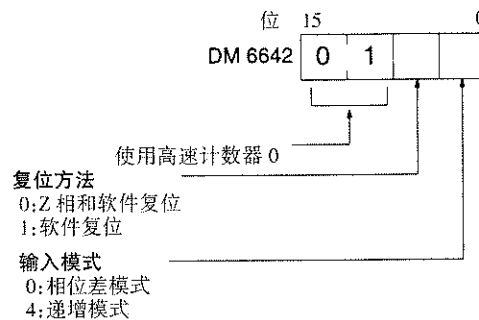
输入刷新字设置(DM 6638)

必须刷新输入时进行这些设置,这些设置与间隔定时器 2 的设置相同。



高速计数器 0 设置(DM 6642)

如果不进行这些设置,程序中不能使用高速计数器 0。



缺省:不使用高速计数器 0

DM 6642 中设置的修改仅在上电或启动 PC 程序执行时变成有效。

编程

使用下列步骤编写高速计数器 0 的程序。

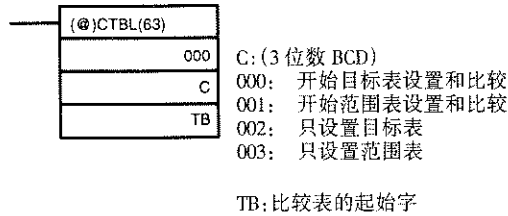
当进行了正确的 PC 设置后,高速计数器 0 开始计数操作,但是不会用比较表进行比较,并且不产生中断,除非执行 CTBL(63)指令。

电源上电并且开始操作时,高速计数器 0 复位为“0”。

高速计数器 0 的当前值保存在 SR 230 和 SR 231 中。

控制高速计数器 0 中断

- 1,2,3... 1. 使用 CTBL(63)指令在 CQM1H 中保存比较表并且开始比较。

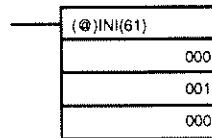


如果 C 设为 000, 则用目标匹配方法进行比较; 如果是 001, 则用范围比较方法进行。比较表被保存, 并且, 当保存操作完成时, 就开始比较。执行比较时, 根据比较表执行高速中断。所保存的比较表的详细内容, 参见第 5 章 指令集中的 CTBL(63)指令的解释。

注 执行范围比较时, 比较结果通常存贮在 AR 1100 ~ AR 1107 中。

如果 C 设为 002, 则用目标匹配方法进行比较; 如果是 003, 则用范围比较方法进行。对于这些设置中的任何一个, 比较表将被保存, 但是不开始比较, 必须用 INI(61)指令开始比较。

2. 执行如下所示的 INI(61)指令停止比较。



第 2 个操作数设为“000”(执行比较), 并且执行 INI(61)指令, 再次开始比较。

一旦保存了一个表, 只要未保存其它表, 它就会在操作时(例如, 程序执行时)保留在 CQM1H 中。

读 PV

有两种方法读 PV, 第一种是从 SR230 和 SR231 读, 第二种是使用 PRV(62)指令。

- 1,2,3... 1. 读 SR 230 和 SR 231

高速计数器 0 的 PV 存贮在 SR 230 和 SR 231 中, 如下所示。对于负值, 最左边数是 F。

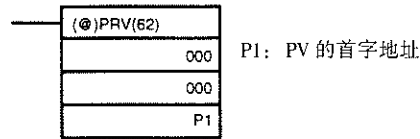
| 最左 4 位数 | 最右 4 位数 | 相位差模式 | 递增模式 |
|---------|---------|------------------------------------|---------------------|
| SR 231 | SR 230 | F0032768 ~ 00032767 (- 32, 768) | 00000000 ~ 00065535 |

注 这些字每周期刷新一次, 因此它们可能与实际 PV 有差别。

如果未使用高速计数器 0, 这些字中的位可用作工作位。

2. 使用 PRV(62)指令

用 PRV(62)指令读高速计数器 0 的 PV。



如下所示存贮高速计数器 0 的 PV,对于负值,最左边数为 F。

| 最左 4 位数 | 最右 4 位数 | 相位差模式 | 递增模式 |
|----------|---------|-----------------------------------|---------------------|
| [PI + 1] | [P1] | F0032768 ~ 00032767 (- 32,768) | 00000000 ~ 00065535 |

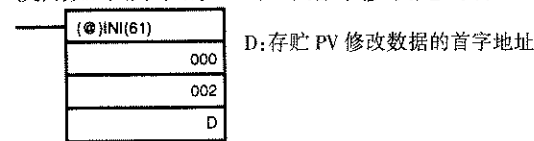
PRV(62)指令实际执行时读 PV。

修改 PV

有两种方法修改高速计数器 0 的 PV。第一种方法是用复位方法复位(在这种情况下,PV 复位为 0)。第二种方法是使用 INI(61)指令。

使用 INI(61)指令的方法解释如下。复位方法的解释,参见高速计数器 0 的开始描述部分。

使用如下所示的 INI(61)指令修改定时器 PV。



| 最左 4 位数 | 最右 4 位数 | 相位差模式 | 递增模式 |
|---------|---------|---------------------|---------------------|
| [D + 1] | [D] | F0032768 ~ 00032767 | 00000000 ~ 00065535 |

最左边数设为 F,指定一个负数。

操作例

此例显示了一个在递增模式下使用高速计数器 0,以目标匹配方法进行比较,并且根据计数器 PV 修改脉冲输出频率的程序。执行程序前,如下设置 PC 设置:

DM 6642: 0114(高速计数器 0 使用软件复位和递增模式),对于其它所有的 PC 设置,使用缺省设定。(中断处理时不刷新输入,并且 IR 100 执行脉冲输出。)

另外,下列数据存贮在比较表中:

DM 0000: 0002—比较条件数:2

DM 0001: 1000—目标值 1:1000

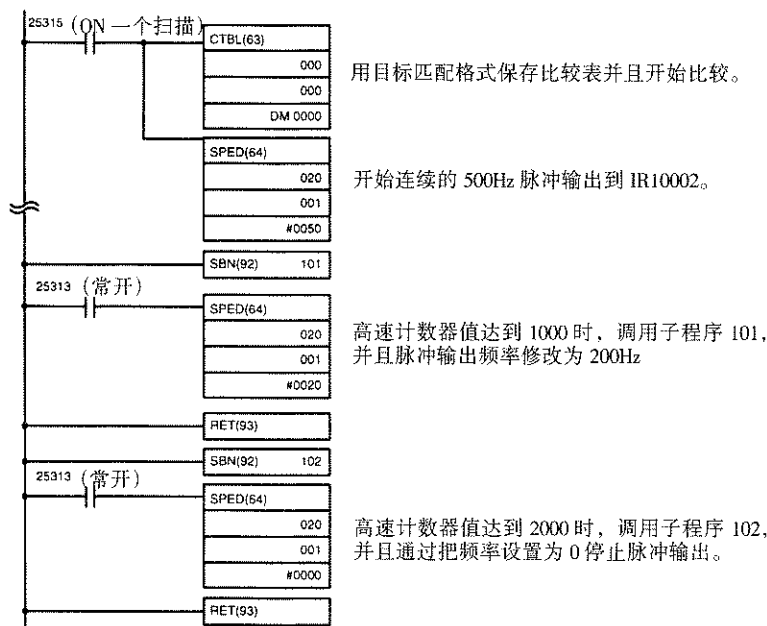
DM 0002: 0000

DM 0003: 0101—比较 1 中断子程序:101

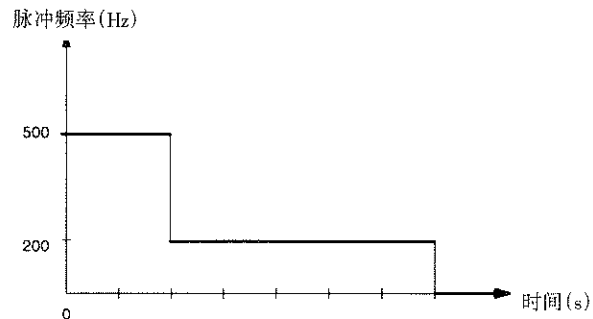
DM 0004: 2000—目标值 2:2000

DM 0005: 0000

DM 0006: 0102—比较 2 中断子程序:102



执行程序时,操作如下:



1-4-6 高速计数器 0 上溢/下溢

如果超过了高速计数器 0 的可计数范围,将出现上溢或下溢状态,并且计数器在上溢时将保持 PV 为 OFFFFFFFFF,在下溢时将保持 PV 为 FFFFFFFF,直到通过计数器复位清除了上溢/下溢状态。可计数范围如下:

相位差模式: F0032768 ~ 00032767

递增模式: 00000000 ~ 00065535

注 1. 上面给出的是理论值,并且假定为合理的短循环时间。这些值实际上在上溢/下溢存在时已存在了一个循环。

2. 高速计数器 0 的 PV 的第 6 和第 7 位数通常是 00,但是可通过检测超过可计数范围的值用作“上溢/下溢标志”。

高速计数器 0 可按前一节所述的复位或通过重启动程序执行自动复位。高速计数器 0 和相关操作在清除上溢/下溢状态前不能正常起作用。上溢/下溢状态中的操作如下。

- 比较表操作停止。
- 不清除比较表。
- 不执行高速计数器的中断程序。
- CTBL(63)只能用作注册比较表。如果试图进行启动比较表操作,则不启动操作,并且不注册比较表。
- INI(61)不能用于启动或停止比较表操作或修改当前值。
- PRV(62)只能读出 OFFFFFFFF 或 FFFFFFFF 作为当前值。

恢复

使用下列步骤从上溢/下溢状态恢复。

有注册的比较表

- 1,2,3... 1. 复位计数器。
2. 如果需要,用 PRV(62)设置 PV。
3. 如果需要,用 CTBL(63)设置比较表。
4. 用 INI(61)启动比较表操作。

没有注册的比较表

- 1,2,3... 1. 复位计数器。
2. 如果需要,用 PRV(62)设置 PV。
3. 设置比较表,并且用 CTBL(63)和 INI(61)启动操作。

注 AR11 中的范围比较结果在恢复后仍能保持。如果在上溢/下溢状态出现前中断条件已经满足,则在恢复后立即满足中断条件的中断程序不会执行。如果必须执行中断程序,在处理前清除 AR11。

复位操作

高速计数器 0 复位时, PV 将设为 0,从 0 开始计数,并将保持比较表、执行状态和执行结果。

启动计数器状态

启动高速计数器 0 时, PC 设置中的计数器模式被读出和使用, PV 将设为 0,清除上溢/下溢状态,清除比较表注册和执行状态,并且清除范围执行结果。(通常在开始操作或注册比较表时清除范围执行结果。)

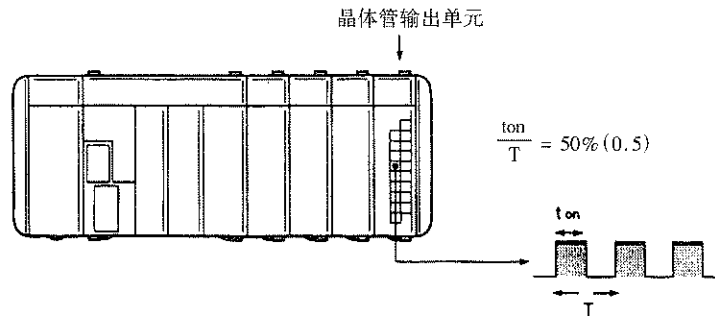
停止计数器状态

停止高速计数器 0 时, PV 将保持,清除比较表注册和执行状态,并且保持范围执行结果。

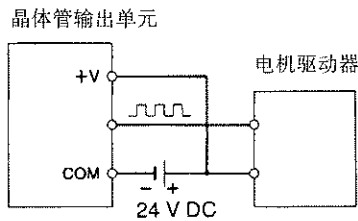
1-5 脉冲输出功能

本节解释使用 CQM1H 脉冲输出功能的设置和方法。输出点和端口的硬件连接的详细内容参见 CQM1H 操作手册。

标准脉冲可使用 SPED(64)从晶体管输出单元的输出点输出。脉冲一次只能从一位输出。脉冲输出的占空率是 50%,并且频率可设置为 20Hz ~ 1kHz。



| 项目 | 规格 |
|------|---|
| 适用单元 | 晶体管输出单元 |
| 脉冲输出 | 指定位脉冲输出 可指定IR 100 ~ IR 115的任意输出字，但是脉冲一次只能输出该字的一位。 |
| 特点 | 频率： 20Hz ~ 1 KHz 占空率： 50% 字规格： PC设置 (DM 6615) 位规格： 梯形图指令中 |
| 适用指令 | 设置脉冲数： PULS(65) 开始脉冲输出： SPED(64) 改变频率： SPED(64) 停止脉冲输出： SPED(64)或INI(61) |



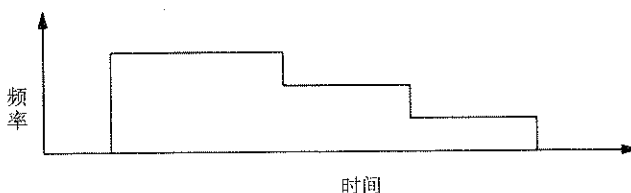
脉冲输出操作

下表显示可与 PULS(65), SPED(64) 和 INI(61) 组合一起进行的脉冲输出操作。

| 改变频率 | 指令 | 操作数设定 |
|------|----------|-----------------|
| | PULS(65) | 脉冲数 (只有独立模式) |
| | SPED(64) | 端口 模式 频率 |
| | SPED(64) | 端口 模式 频率 |
| | SPED(64) | 端口 频率 = 0 |
| | SPED(64) | 端口 频率 = 0 |
| | INI(61) | 控制字 = 003 |

注 此应用中必须使用晶体管输出单元。

从输出点输出脉冲时，可再次执行 SPED(64) 逐步把频率改变到不同的频率值，如下图所示。



脉冲可以连续模式或独立模式从输出点输出。

连续模式

脉冲连续输出直到用 SPED(64) 或 INI(61) 停止。

独立模式

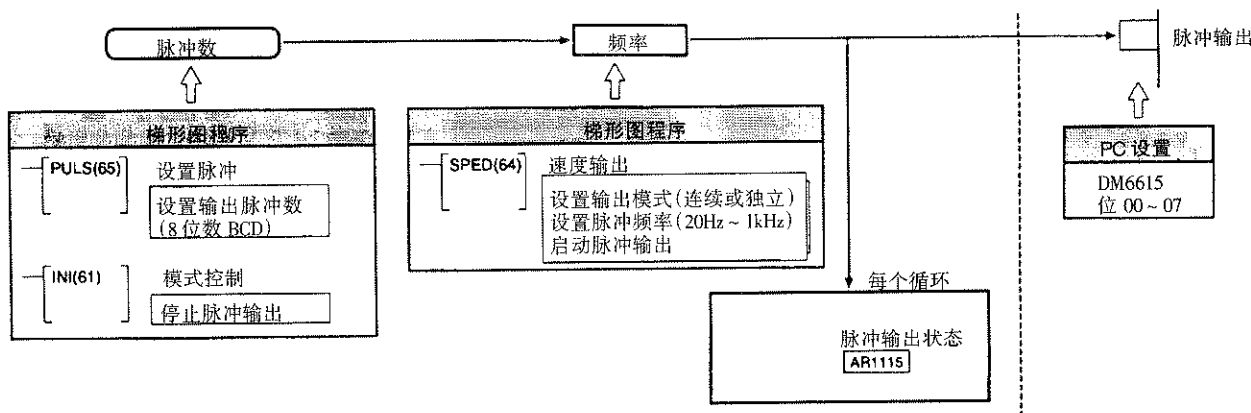
一旦输出了 SPED(64) 中指定数量的脉冲，脉冲输出自动停止。(用 SPED(64) 或 INI(61) 也能提前停止脉冲输出。)

步骤

从晶体管输出单元输出脉冲时按照下列步骤。脉冲一次只能从晶体管输出单元的一个端子输出。

- 1,2,3... 1. 决定脉冲输出要用的 IR 字(IR 100 ~ IR 115)。
2. 给晶体管输出单元接线。给所选字中实际使用的相应位的端子接线。
3. 在 PC 设置的 DM 6615 中设置所需的 IR 字地址。对应于 IR 100 ~ IR 115 设为 0000 ~ 0015BCD。(更多详细内容参见 40 页。)
4. 编写相关程序段。
 - a) PULS(65) 可用于设置输出脉冲个数。
 - b) SPED(64) 可用于控制脉冲输出(没有加速或减速的脉冲输出。)
 - c) INI(61) 可用于停止脉冲输出。

晶体管脉冲输出
(从分配 IR100 ~ IR115 之间的字的输出单元)

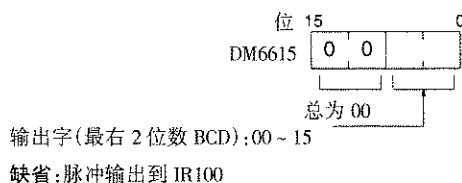


设定 PC 设置

执行 SPED(64) 从输出单元输出脉冲前，把 PC 设为编程模式，并且在 PC 设置中进行下列设定。

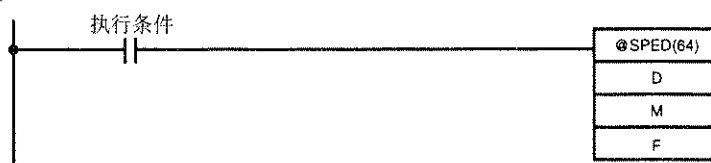
在 DM 6615 中，指定用于脉冲输出到输出单元的 SPED(64) 的输出字。(此位在 SPED(64) 的第一个操作数中指定。)

DM 6615 的内容(0000 ~ 0015)指定 IR 100 ~ IR 115 的输出字。例如,如果 DM 6615 设为 0002,脉冲将输出到 IR 102。



连续脉冲输出

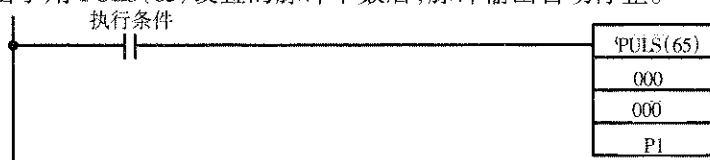
执行 SPED(64)时,脉冲将在指定的输出位开始输出。设置输出位 00 ~ 15(D = 000 ~ 150)和频率 20Hz ~ 1000Hz(F = 0002 ~ 0100)。模式设置为连续模式(M = 001)。



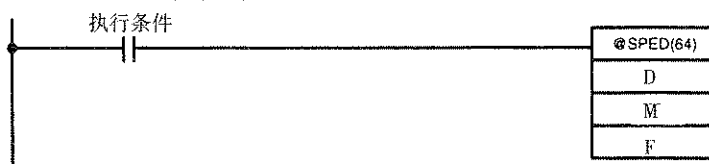
通过执行 INI(61),其中 C = 003 或再次执行 SPED(64),其中频率设为 0,可停止脉冲输出。可再次执行 SPED(64)用不同的频率设置改变频率。

设置脉冲个数

以独立模式执行 SPED(64)前,可用 PULS(65)设置要输出的总的脉冲个数。当输出了用 PULS(65)设置的脉冲个数后,脉冲输出自动停止。



PULS(65)设置 8 位数脉冲数 P1 + 1, P1。这些脉冲可从 00000001 到 16777215 设置。以独立模式执行 SPED(64)时,可获得用 PULS(65)设置的脉冲个数。(输出时不能改变脉冲个数。)



执行 SPED(64)时,脉冲以指定的输出位(D = 000 ~ 150; 位 00 ~ 15),以指定的频率(F = 0002 ~ 0100; 20Hz ~ 1000Hz)开始输出。设置独立模式(M = 000)输出用 PULS(65)设置的脉冲个数。再次执行 SPED(64),用不同的频率设置改变频率。

改变频率

再次执行 SPED(64),用不同的频率设置改变脉冲输出的频率。使用用于启动脉冲输出的相同的输出位(P)和模式(M)设置。新频率可为 20Hz ~ 1000Hz(F = 0002 ~ 0100)。

1-6 通信功能

下表显示了 CQM1H CPU 单元的通信端口支持的通信模式。(CQM1H - CPU11 CPU 单元没有 RS - 232C 端口。

这些通信模式的 PC 设置设定和通信步骤在本章后面介绍。

| 通信 | 用法 | 端口 | |
|------------|--------------------------------------|----|-----------|
| | | 外围 | RS-232C |
| 编程器总线 | 编程器连接 | 是 | 否 |
| 外围总线 | 用支持软件连续到个人计算机 | 是 | 否 |
| 上位机链接 | 上位机链接或可编程序终端连接 | 是 | 是 |
| 协议宏 | 使用任意协议的标准外部设备的数据传输 | 否 | 否 |
| 无协议 | 标准外部设备的无协议通信 | 是 | 是 |
| 1:1数据链接 | 用另一个CPU单元建立数据链接 | 否 | 是 |
| 1:1模式的NT链接 | 用可编程终端建立1:1数据链接 | 否 | 是 (见注) |
| 1:N模式的NT链接 | 用一个可编程终端建立1:1数据链接或两个或更多可编程终端建立1:N连接。 | 否 | 否 |

- 注
1. 可使用可编程终端的编程器功能,但是 DIP 开关上的 7 脚必须置 ON。。
 2. 任何设备使用外围端口(不包括编程器)时,CPU 单元的 DIP 开关的 7 脚置 ON。

1-6-1 上位机链接和无协议通信设定

本节介绍了上位机链接和无协议通信模式共用的 PC 设置设定。在建立上位机链接或无协议通信前进行所需的 PC 设置设定。

- 注 如果 CQM1H 的 DIP 开关的 5 脚置 ON, 则忽略 PC 设置通信参数并使用下列设定。

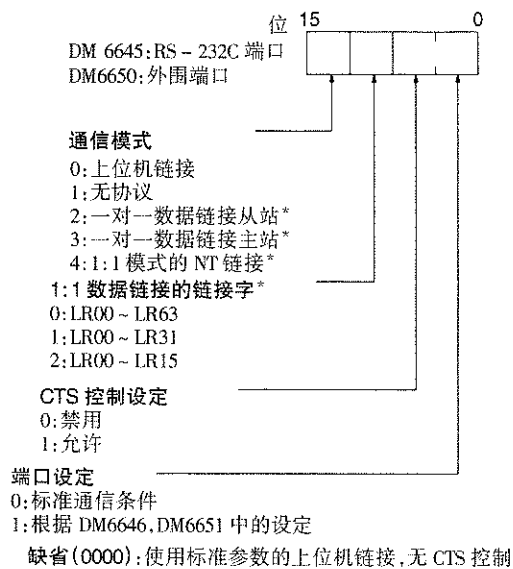
| 参数 | DIP开关的5脚置ON时的设定 |
|------|-----------------|
| 通信模式 | 上位机链接 |
| 节点号 | 00 |
| 起始位 | 1位 |
| 数据长度 | 7位 |
| 停止位 | 2位 |
| 校验 | 偶 |
| 波特率 | 9,600 bps |
| 传输延迟 | 无 |

DM 6645 ~ DM 6654 中的 PC 设置参数用于设置通信端口的参数。

通信设定

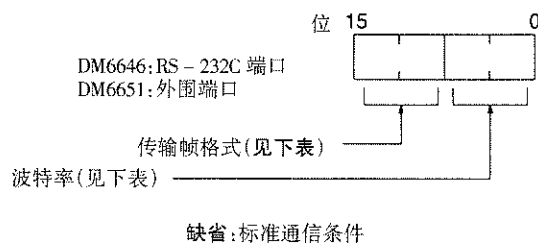
(DM 6645 和 DM 6650)

DM 6645 和 DM 6650 中的设定决定了主通信参数,如下图所示。



通信设定

当 CPU 单元的 DIP 开关的 5 脚置 OFF, 并且 DM 6646(或 DM 6651)中的设定在 (DM 6646 和 DM 6651) DM 6645(或 DM 6650)中允许, 则这些设定决定了传输的帧格式和波特率, 如下图所示。



传输帧格式

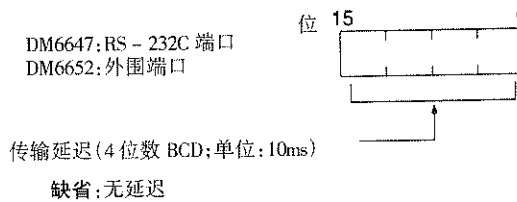
| 设定 | 停止位 | 数据长度 | 停止位 | 校验 |
|----|-----|------|-----|----|
| 00 | 1 | 7 | 1 | 偶 |
| 01 | 1 | 7 | 1 | 奇 |
| 02 | 1 | 7 | 1 | 无 |
| 03 | 1 | 7 | 2 | 偶 |
| 04 | 1 | 7 | 2 | 奇 |
| 05 | 1 | 7 | 2 | 无 |
| 06 | 1 | 8 | 1 | 偶 |
| 07 | 1 | 8 | 1 | 奇 |
| 08 | 1 | 8 | 1 | 无 |
| 09 | 1 | 8 | 2 | 偶 |
| 10 | 1 | 8 | 2 | 奇 |
| 11 | 1 | 8 | 2 | 无 |

波特率

| 设定 | 波特率 |
|----|------------|
| 00 | 1,200 bps |
| 01 | 2,400 bps |
| 02 | 4,800 bps |
| 03 | 9,600 bps |
| 04 | 19,200 bps |

传输延迟时间
(DM 6647 和 DM 6652)

根据连接到通信端口的设备,可能需要允许传输的时间。在这种情况下,设置传输延迟以调整允许的时间。

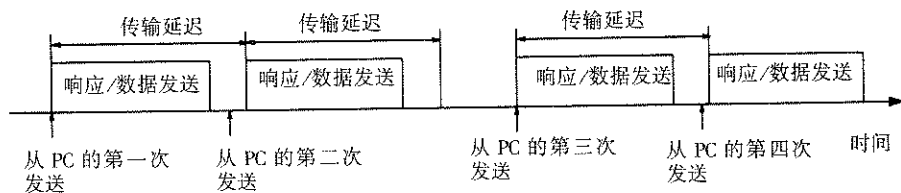


在 PC 设置中设定传输延迟,建立从 PC 发送的数据间的最小间隔。传输延迟用于下列串行通信模式。

| 串行通信模式 | 应用 |
|-------------------------|---|
| 上位机链接, 响应 | 一旦PC发送了对上位计算机的响应, 将不发送下一个响应直到设置的传输延迟时间经过。 |
| 上位机链接, PC初始的通信 无协议通信 | 一旦PC使用TXD(48)发送了数据, 将不会再次发送数据, 直到设置的传输延迟时间经过。 |

从 PC 第一次发送数据时不使用延迟。延迟对其它发送的影响仅在正常发送的时间在设置的传输延迟时间经过后才产生。

如果在下一次发送准备好时已经达到了延迟时间, 则立即发送数据。如果未达到延迟时间, 将延迟发送直到设置的传输延迟时间经过。



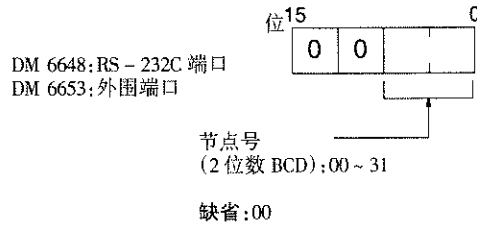
1-6-2 上位机链接通信设定和步骤

本节介绍了上位机链接通信需要的 PC 设置设定和步骤。

PC 设置设定

确认在 DM 6645(RS-232C 端口)或 DM 6650(外围端口)的最左位数写 00, 以指定上位机链接通信。其它上位机链接通信参数在 DM 6645/DM 6650 和 DM 6646/DM6651 的最右两位数设置。

当多节点参与通信时,必须为上位机链接通信设置节点号,用以区分不同的节点。只有上位机链接通信需要此设定。



上位机链接通信概述

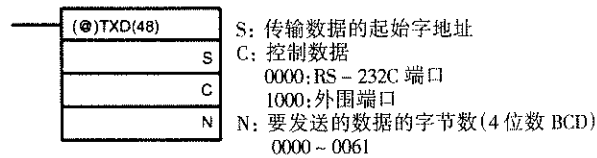
节点号通常设为 00,不需要其它设定,除非网络中连接了多节点。
上位机链接通信是 OMRON 开发的,用于通过 RS - 232C 电缆连接 PC 和一个或多个上位计算机,并且从上位计算机控制 PC 通信。通常上位计算机发布命令给 PC,PC 自动反馈响应。因此可以没有 PC 的有效参与而进行通信。PC 也有能力在必须直接参与时发起数据传输。

总的来说,有两种实现上位机链接通信的方法。一种是基于 C 模式命令,另一种是基于 FINS(CV 模式)命令。CQM1H 只支持 C 模式指令。上位机链接通信的详细内容,参见第 6 章 上位机链接命令。

通信步骤

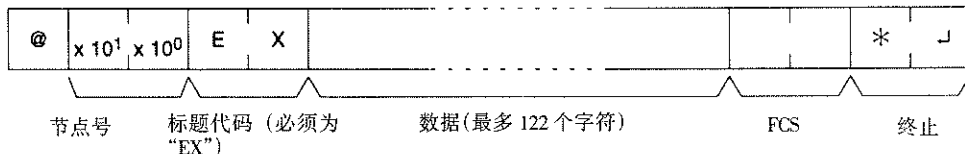
本节介绍了怎样使用上位机链接从 CQM1H 执行数据传输。使用此方法允许在数据改变时从 CQM1H 自动传输数据,因此可通过减少计算机经常监视的需要来简化通信处理。

- 1,2,3... 1. 检查确保 AR 0805(RS - 232C 端口传输允许标志)置 ON。
2. 使用 TXD(48)指令传输数据。



从执行指令到完成数据传输, AR 0805(或外围端口的 AR0813)将保持为 OFF。完成数据传输后,它将再次置 ON。TXD(48)指令不提供响应,因此为了接收确认计算机已接收了数据,必须写计算机的程序,使 CQM1H 写数据时给出确认通知。

用 TXD(48)指令以上位机链接模式用于数据传输的传输数据帧如下所示。

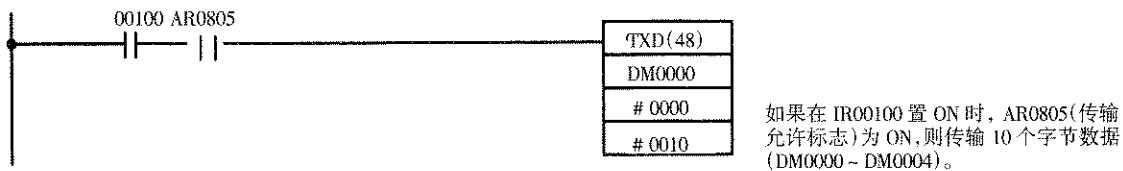


为复位 RS - 232C 端口(即恢复初始状态),把 SR25209 置 ON。为复位外围端口,把 SR 25208 置 ON。这些位在复位后将自动置 OFF。

如果在 CQM1H 响应计算机指令时执行 TXD(48) 指令,则在根据 TXD(48) 指令执行传输前首先完成响应传输。在所有其它情况下,基于 TXD(48)指令的数据传输具有最高优先级。

程序例

此例显示了以上位机链接模式用 RS-232C 端口向计算机传送 10 字节数据 (DM 0000 ~ DM 0004) 的程序。所有 PC 设置都假定为缺省值。(即以上位机链接模式使用 RS-232C 端口,节点号是 00,使用标准通信条件。)从 DM 0000 到 DM 0004 的每个字中存贮“1234”。对计算机,执行程序接收用标准通信条件的 CQM1H 数据。



必须在上位计算机中准备好下面这种程序以接收数据。此程序允许计算机在执行上位机链接读命令从 PC 读数据时,读并显示出从 PC 接收的数据。

```

10 'CQM1H SAMPLE PROGRAM FOR EXCEPTION
20 CLOSE 1
30 CLS
40 OPEN "COM:E73" AS #1
50 *KEYIN
60 INPUT "DATA -----",S$
70 IF S$=" " THEN GOTO 190
80 PRINT "SEND DATA = ";S$
90 ST$=S$
100 INPUT "SEND OK? Y or N?=",B$
110 IF B$="Y" THEN GOTO 130 ELSE GOTO *KEYIN
120 S$=ST$
130 PRINT #1,S$
140 INPUT #1,R$
150 PRINT "RECV DATA = ";R$
160 IF MID$(R$,4,2)="EX" THEN GOTO 210 '从 PC 识别命令
170 IF RIGHT$(R$,1)<>"*" THEN S$=" ":GOTO 130
180 GOTO *KEYIN
190 CLOSE 1
200 END
210 PRINT "EXCEPTION!! DATA"
220 GOTO 140

```

计算机接收的数据如下所示。(FCS 是“59”。)

“@00EX1234123412341234123459*CR”

1-6-3 无协议通信设定和步骤

本节介绍了无协议通信所需的 PC 设置设定和步骤。无协议通信允许与标准设备交换数据。例如,数据可输出到打印机或从条形码阅读器输入。

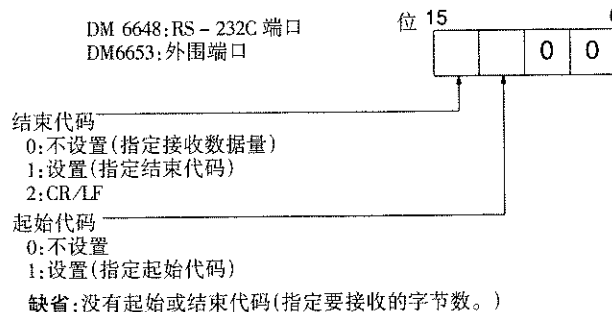
PC 设置设定

确认在 DM 6645(RS-232C 端口)或 DM 6650(外围端口)的最左位数写 10,以规定无协议通信。

其它通信参数在 DM 6645/ DM 6650 和 DM 6646/ DM 6651 的最右两位数设置。

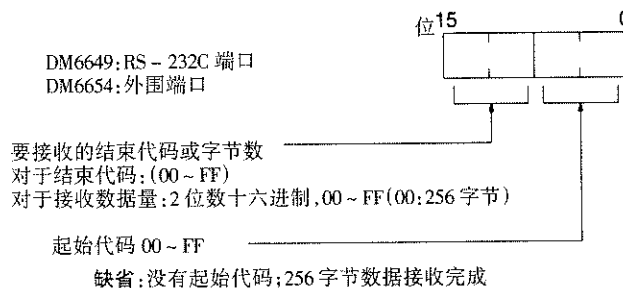
如果无协议通信需要,可如下图所示设置要接收的起始和结束代码或数据量。只有无协议通信需要设定。这些设定仅在 DIP 开关的 5 脚置 OFF 时有效。

允许起始和结束代码



指定数据开头是否设定起始代码,结尾是否设定结束代码。也可以不设定结束代码,指定在完成接收操作前要接收的字节数。要接收的代码和数据的字节数在 DM 6649 或 DM 6654 中设置。

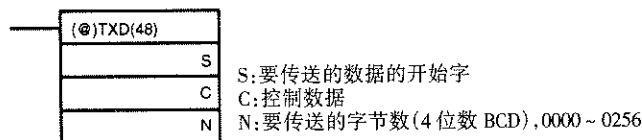
设定起始代码,结束代码和接收数据量。



通信步骤

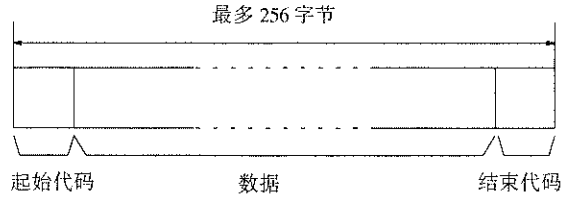
传输

- 1,2,3... 1. 检查确保 AR 0805(RS-232C 端口传输允许标志)置 ON。
2. 用 TXD(48)指令传送数据。



从执行该指令到完成数据传输,AR 0805(或外围端口的 AR 0813)将保持为 OFF。(在完成数据传输时将再次置 ON。)

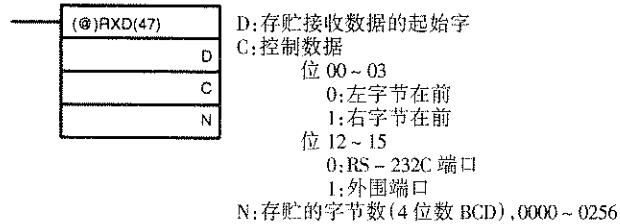
指定要传送的字节数时不包括起始和结束代码。可发送的带或不带起始和结束代码的最大传输是 256 字节。根据起始和结束代码的定义, N 最大在 254 和 256 之间。如果要发送的字节数设为 0000, 则只发送起始和结束代码。



要复位 RS-232C 端口(即恢复初始状态), 把 SR 25209 置 ON。要复位外围端口, 把 SR 25208 置 ON。复位后这些位将自动置 OFF。

接收

- 1, 2, 3... 1. 确认 AR 0806(RS-232C 接收完成标志) 或 AR 0814(外围接收完成标志) 置 ON。
2. 用 RXD(47) 指令接收数据。



3. 读接收数据的结果存贮在 AR 区中。检查确保操作正确完成。这些位的内容在每次执行 RXD(47) 时都将复位。

| RS-232C 端口 | 外围端口 | 错误 |
|-------------------|-------------------|--|
| AR 0800 ~ AR 0803 | AR 0808 ~ AR 0811 | RS-232C 端口错误代码(1位数BCD) 0: 正常完成 1: 校验错误 2: 帧错误 3: 过运行错误 |
| AR 0804 | AR 0812 | 通信错误 |
| AR 0807 | AR 0815 | 接收过运行标志(完成接收后, 用 RXD(47) 指令读数据前又接收到随后的数据。) |
| AR 09 | AR 10 | 接收的字节数(4位数BCD) |

为复位 RS-232C 端口(即恢复初始状态), 把 SR25209 置 ON。为复位外围端口, 把 SR25208 置 ON。复位后这些位将自动置 OFF。

起始代码和结束代码不包括在 AR09 或 AR10(接收的字节数)中。

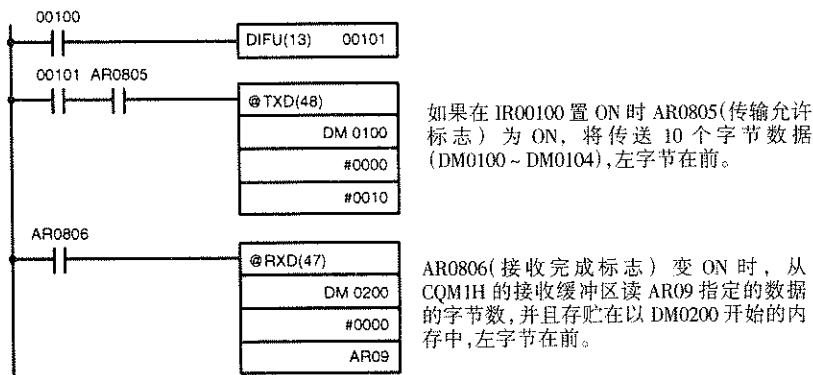
程序例

此例显示了以无协议模式使用 RS-232C 端口向计算机传送 10 字节数据(DM 0100 ~ DM 0104)的程序, 并且将从计算机接收的数据存贮到 DM 0200 开始的 DM 区。执行程序前, 必须设定下列 PC 设置。

DM 6645:1000(无协议模式的 RS-232C 端口;标准通信条件)

DM 6648:2000(没有起始代码;结束代码 CR/LF)

所有其它的 PC 设置设定都假定为缺省值。3132 存贮在从 DM 0100 到 DM 0104 的每一个字中。对计算机,执行程序接收用标准通信条件的 CQM1H 数据。



数据如下:“31323132313231323132CR LF”

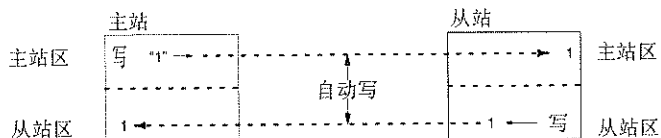
1-6-4 一对一数据链接

如果 CQM1H 通过 RS-232C 端口连接到另一个 CPU 单元, 实现一对一链接, 则它们共享公共 LR 区。一个 PC 用作主站, 另一个用作从站。CQM1H 可与以下任一种 PC 实现一对一链接: CQM1H, CQM1, C200HX/HG/HE, C200HS, CPM1, CPM1A, CPM2A, CPM2C 或 SRM1(-V2)。

注 外围端口不能用作 1:1 数据链接。使用 CPU 单元的内置 RS-232C 端口或串行通信板的 RS-232C 或 RS422A/485 端口。

一对一数据链接

1:1 数据链接允许两个 CQM1H 共享 LR 区的公共数据。如下图中所示, 当数据写入一个所链接单元的 LR 区中的一个字时, 它会自动按原样写到其它单元的同一个字中。每个 PC 具有可写的指定字和被其它 PC 写的指定字。每个被其它 PC 写的字都能读, 但不能写。

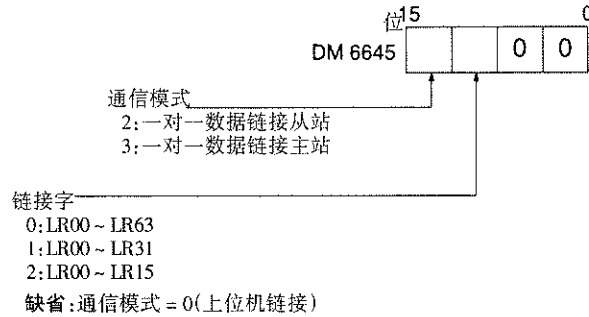


根据主、从和链接字的设定, 每个 PC 使用的字如下表所示。如果 CQM1H 与 CPM1, CPM1A, CPM2A 或 SRM1(-V2) PC 链接, 则设置链接区为 LR 00 ~ LR 15。

| DM 6645 设定 | 主站区 | 从站区 |
|---------------|---------------|---------------|
| LR 00 ~ LR 15 | LR 00 ~ LR 07 | LR 08 ~ LR 15 |
| LR 00 ~ LR 31 | LR 00 ~ LR 15 | LR 16 ~ LR 31 |
| LR 00 ~ LR 63 | LR 00 ~ LR 31 | LR 32 ~ LR 63 |

PC 设置设定

为使用 1:1 数据链接, 只需要设置通信模式和链接字。一个 PC 通信模式设置为 1:1 数据链接主站, 另一个 PC 设置为 1:1 数据链接从站, 并且在定义为主站的 PC 中设置链接字。



注 这些设置只在 CPU 单元的 DIP 开关的 5 脚置 OFF 时有效。位 08 ~ 11 只在 1:1 数据链接主站时有效。

通信步骤

如果进行了正确的主和从设置，只需把 CPU 单元都通电就能自动启动一对一数据链接，并且操作将独立于 CPU 单元的操作模式。

链接错误

如果在一秒钟内从站没有接收到来自主站的响应，则 1:1 数据链接错误标志 (AR 0802) 和通信错误标志 (AR 0804) 将置 ON。

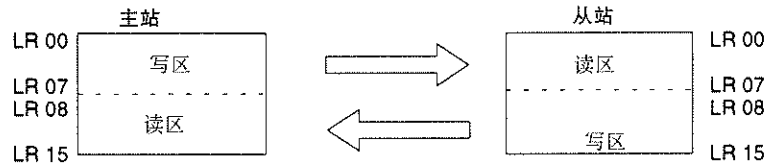
程序例

此例显示了检验用 RS - 232C 端口执行一对一数据链接情况的程序。执行程序前，设置下列 PC 设置参数。

主站: DM 6645:3200(1:1 数据链接主站;使用区域:LR 00 ~ LR 15)

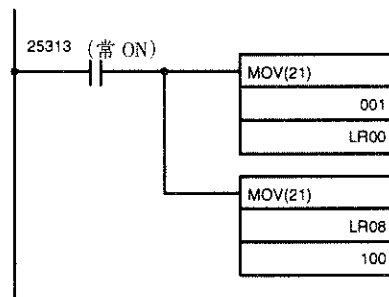
从站: DM 6645:2000(1:1 数据链接从站)

所有其它 PC 设置参数假定为缺省。一对一数据链接使用的字如下所示。

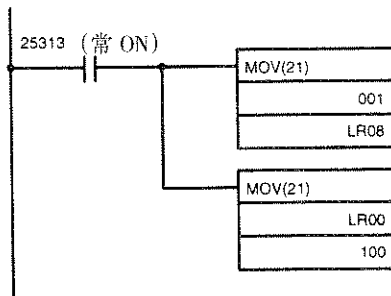


在主站和从站同时执行程序时，每个单元的 IR 001 的状态反映在其它单元的 IR 100 中。同样，其它单元的 IR 001 的状态反映在每个单元的 IR 100 中。IR 001 是输入字，IR 100 是输出字。

在主站中



在从站中



1-6-5 NT 链接 1:1 模式通信

本节介绍了通信模式设为 1:1 模式 NT 链接的可编程终端的通信，外围端口不能用作 NT 链接通信。

设定

通过把 DM 6645 设置为 4000,把通信模式设为 1:1 模式 NT 链接。确认 CPU 单元的 DIP 开关的 5 脚置为 OFF。

可编程终端设定的详细内容,参见可编程终端操作手册。

NT Link1:1 模式通信概述

NT 链接通信是 OMRON 开发的,在 PC 和可编程终端之间提供高速通信。有两种 NT 链接通信:单个可编程终端连接到 PC 的 1:1 模式和多个可编程终端连接到 PC 的 1:N 模式。CQM1H 的内置 RS-232C 端口只支持 1:1 模式通信,但是如果 PC 上安装了可选串行通信板,则 1:1 和 1:N 模式都能使用。

一些可编程终端具有编程器功能,它允许可编程终端对 CQM1H 进行编程和监视。如果编程器连接到 CQM1H 的外围端口,不能使用可编程终端的编程器功能。编程器功能的详细内容,参见可编程终端的操作手册。

通信步骤

对于 NT 链接通信,PC 自动响应可编程终端发出的指令,因此 CQM1H 不需要通信程序,不执行 NT 链接通信步骤。

1-6-6 端口接线

通信端口接线的信息参见 CQM1H 操作手册。

1-7 计算有符号二进制数据

CQM1H PC 允许有符号二进制数计算。下列指令控制有符号二进制数。用 2 的补码处理有符号数。

下列有符号二进制指令在 CQM1H PC 中有效:

单字指令

- 2 的补码—NEG(—)
- 二进制加—ADB(50)
- 二进制减—SBB(51)
- 有符号二进制乘—MBS(—)

- 有符号二进制除—DBS(—)
- 双字(长)指令
- 双字 2 的补码—NEGL(—)
- 双字二进制加—ADBL(—)
- 双字二进制减—SBBL(—)
- 双字有符号二进制乘—MBSL(—)
- 双字有符号二进制除—DBSL(—)

1-7-1 有符号二进制数据定义

CQMIH 提供对一字或两字数据的操作指令。用 2 的补码控制有符号二进制数,一字或两字数据的 MSB 用作符号位。可用一字或两字表示的数据范围如下:

- 一字数据:
 - 32,768 ~ 32,767(8000 ~ 7FFF 十六进制)
- 两字数据:
 - 2,147,483,648 ~ 2,147,483,647(8000 0000 ~ 7FFF FFFF 十六进制)

下表显示十进制数和十六进制数据之间的等值。

| 十进制 | 16位十六进制 | 32位十六进制 |
|-----------------|---------|-----------|
| 2,147,483,647 | —— | 7FFF FFFF |
| 2,147,483,646 | —— | 7FFF FFFE |
| . | . | . |
| . | . | . |
| . | . | . |
| 32,768 | —— | 0000 8000 |
| 32,767 | 7FFF | 0000 7FFF |
| 32,766 | 7FFE | 0000 7FFE |
| . | . | . |
| . | . | . |
| . | . | . |
| 2 | 0002 | 0000 0002 |
| 1 | 0001 | 0000 0001 |
| 0 | 0000 | 0000 0000 |
| -1 | FFFF | FFFF FFFF |
| -2 | FFFE | FFFF FFFE |
| . | . | . |
| . | . | . |
| . | . | . |
| - 32,767 | 8001 | FFFF8001 |
| - 32,768 | 8000 | FFFF8000 |
| - 32,769 | —— | FFFF7FFF |
| . | . | . |
| . | . | . |
| . | . | . |
| - 2,147,483,647 | —— | 8000 0001 |
| - 2,147,483,648 | —— | 8000 0000 |

1-7-2 算术标志

执行有符号二进制指令的结果反映在算术标志中。下表给出了标志和其置 ON 的条件,这些条件不满足时标志置 OFF。

| 标志 | 置 ON 条件 |
|-----------------|--|
| 进位标志 (SR 25504) | 加法的进位。 减法结果为负。 |
| 等于标志 (SR 25506) | 加法,减法, 乘法或除法的结果为 0。 2 的补码转换结果为 0。 |
| 上溢标志 (SR 25404) | 16 位加法或减法的结果超过 32767 (7FFF)。32 位加法或减法的结果超过 2147483647 (7FFF FFFF)。 |
| 下溢标志 (SR 25405) | 16 位加法或减法或 2 的补码的转换结果超过 -32,768 (8000)。 32 位加法或减法或 2 的补码的转换结果超过 -2147483648 (80000000)。 |

1-7-3 用十进制数输入有符号二进制数据

虽然有符号二进制数的计算使用十六进制表达,可使用十进制输入和指令的助记符从编程器或 CX-Programmer 输入。使用编程器以十进制数输入的步骤在 *CQM1H 操作手册* 中介绍。使用 CX-Programmer 时,详细内容参见 *CX-Programmer 操作手册: C 系列 PC*。

输入指令

下列指令只能输入 16 位操作数: NEG(—), ADB(50), SBB(51), MBS(—) 和 DBS(—)。从编程器输入指令的详细内容,参见 *CQM1H 操作手册*。

1-7-4 使用有符号二进制扩展指令

下列 CQM1H 指令在使用前必须在指令表中分配功能代码。

- 2 的补码—NEG(—)
- 双字 2 的补码—NEG(—)
- 双字二进制加—ADBL(—)
- 双字二进制减—SBBL(—)
- 有符号二进制乘—MBS(—)
- 双字有符号二进制乘—MBSL(—)
- 有符号二进制除—DBS(—)
- 双字有符号二进制除—DBSL(—)

分配功能代码

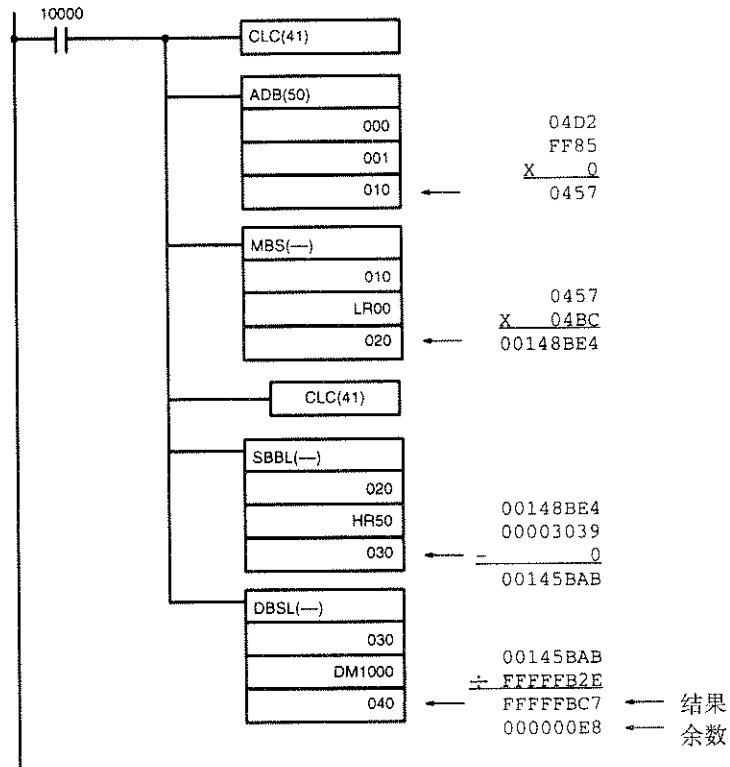
使用编程器分配功能代码的步骤在 *CQM1H 操作手册* 中介绍。在执行此操作前确认 CQM1H 的 DIP 开关的 4 脚置 ON,允许使用用户设置指令表。

1-7-5 使用有符号二进制数据的应用例

下面的程序可在 CQM1H 中进行计算：

$$((1234 + (-123)) \times 1212 - 12345) \div (-1234) = -1081, \text{余数 } 232$$

| | | | | |
|---------|---|------|---|-------|
| 000 | = | 04D2 | ← | 1234 |
| 001 | = | FF85 | ← | -123 |
| LR00 | = | 04BC | ← | 1212 |
| HR50 | = | 3039 | ← | 12345 |
| HR51 | = | 0000 | ← | |
| DM 1000 | = | FB2E | ← | -1234 |
| DM 1001 | = | FFFF | ← | |



第 2 章 内板

本章介绍了下列内板的软件应用信息:高速计数器板,脉冲 I/O 板,绝对值编码器接口板,模拟量设定板,模拟量 I/O 板和串行通信板。硬件信息参见 *CQMIH 操作手册*。

| | | |
|--------|----------------|-----|
| 2-1 | 高速计数器板 | 56 |
| 2-1-1 | 型号 | 56 |
| 2-1-2 | 功能 | 56 |
| 2-1-3 | 系统配置举例 | 56 |
| 2-1-4 | 适用的内板插槽 | 57 |
| 2-1-5 | 名称和功能 | 57 |
| 2-1-6 | 规格 | 58 |
| 2-1-7 | 高速计数器 1~4 | 61 |
| 2-2 | 脉冲 I/O 板 | 79 |
| 2-2-1 | 型号 | 79 |
| 2-2-2 | 功能 | 79 |
| 2-2-3 | 系统配置 | 80 |
| 2-2-4 | 适用的内板插槽 | 81 |
| 2-2-5 | 名称和功能 | 81 |
| 2-2-6 | 规格 | 82 |
| 2-2-7 | 高速计数器 1 和 2 | 88 |
| 2-2-8 | 功能 | 97 |
| 2-2-9 | 固定占空率脉冲输出 | 97 |
| 2-2-10 | 可变占空率脉冲输出 | 109 |
| 2-2-11 | 决定端口 1 和 2 的状态 | 112 |
| 2-2-12 | 使用脉冲输出功能时的注意事项 | 113 |
| 2-3 | 绝对值编码器接口板 | 113 |
| 2-3-1 | 型号 | 113 |
| 2-3-2 | 功能 | 114 |
| 2-3-3 | 系统配置 | 114 |
| 2-3-4 | 适用的内板插槽 | 114 |
| 2-3-5 | 名称和功能 | 115 |
| 2-3-6 | 绝对值编码器输入规格 | 115 |
| 2-3-7 | 高速计数器中断 | 117 |
| 2-4 | 模拟量设定板 | 126 |
| 2-4-1 | 型号 | 126 |
| 2-4-2 | 功能 | 126 |
| 2-4-3 | 适用的内板插槽 | 126 |
| 2-4-4 | 名称和功能 | 127 |
| 2-4-5 | 规格 | 127 |
| 2-5 | 模拟量 I/O 板 | 127 |
| 2-5-1 | 型号 | 127 |
| 2-5-2 | 功能 | 127 |
| 2-5-3 | 系统配置 | 128 |
| 2-5-4 | 适用的内板插槽 | 128 |
| 2-5-5 | 名称和功能 | 129 |
| 2-5-6 | 规格 | 130 |
| 2-5-7 | 应用步骤 | 132 |
| 2-6 | 串行通信板 | 132 |
| 2-6-1 | 型号 | 132 |
| 2-6-2 | 串行通信板 | 132 |
| 2-6-3 | 特征 | 132 |
| 2-6-4 | 系统配置 | 134 |

2-1 高速计数器板

2-1-1 型号

| 名称 | 型号 | 规格 |
|--------|-------------|-----------------------|
| 高速计数器板 | CQM1H-CTB41 | 4路脉冲输入 4个比较结果的外部输出 |

2-1-2 功能

高速计数器板是处理 4 路脉冲输入的内板。

高速计数器脉冲输入 1~4

高速计数器板计算从端口 1~4 输入的 50~500KHz 的高速脉冲,并且根据计算的脉冲数执行任务。

输入模式

下列三种输入模式有效:

- 相位差模式(1×/2×/4×)
- 增/减模式
- 脉冲/方向模式

比较操作

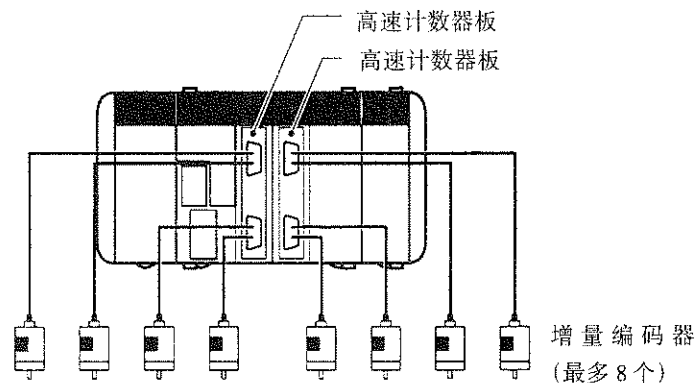
当高速计数器的 PV(当前值)等于指定的目标值或在指定范围内,比较表中指定的位型式存贮在内部输出位和外部输出位中。可为每个比较结果设置位型式,外部输出位可通过如下所述的外部输出端子输出。

外部输出

等于目标值或满足范围比较条件时,可产生最多 4 个外部输出。

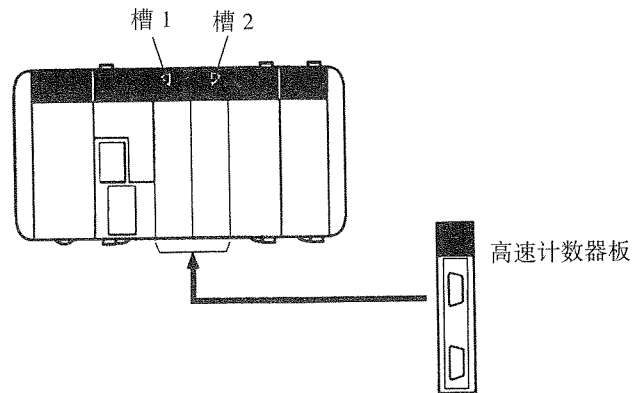
注 高速计数器板不提供高速计数器中断。它只是把 PV 与目标值或比较范围作比较,并产生内部和外部位输出。

2-1-3 系统配置举例



2-1-4 适用的内板插槽

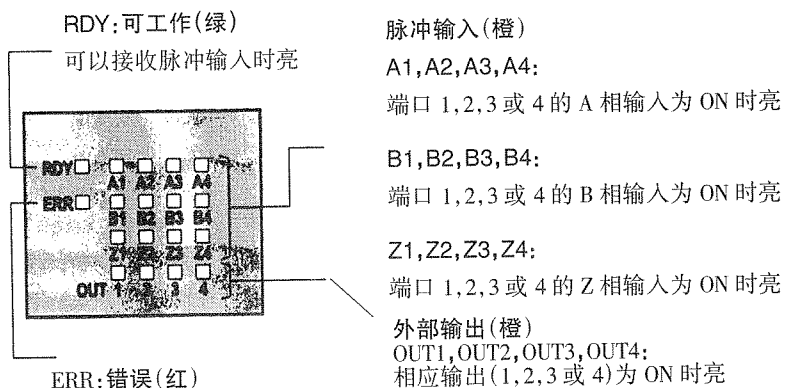
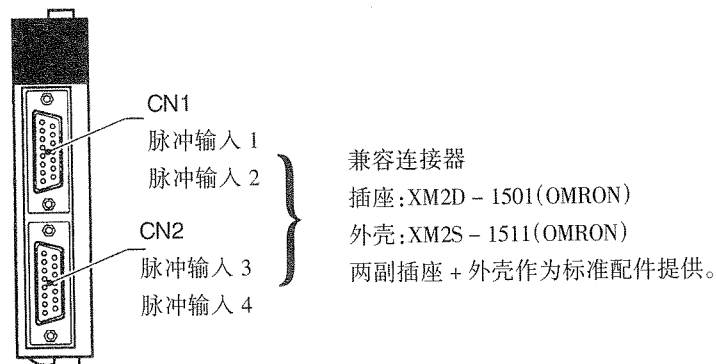
高速计数器板可安装在 CQM1H - CPU51/61 CPU 单元的槽 1(左槽)或槽 2(右槽)上。两个槽可同时使用。



2-1-5 名称和功能

一块高速计数器板提供两个接受高速脉冲输入的连接器和 CN1 用于输入 1 和 2, CN2 用于输入 3 和 4。

CQM1H - CTB41 高速计数器板



2-1-6 规格

指令

| 指令 | 意义 |
|----------|--|
| CTBL(63) | 用于注册目标或范围比较表或用于启动前一个注册比较表的比较。可注册表并且用单独指令或相同指令启动比较。 |
| INI(61) | 用于启动或停止使用注册比较表的比较,或用于改变高速计数器的PV。 |
| PRV(62) | 用于读高速计数器的PV或状态。 |

有关控制位,标志和状态信息

| 字 | | 位 | 名称 | | 功能 |
|--|--|---------|-------------------|-----------|--|
| 槽1 | 槽2 | | | | |
| IR 200 | IR 232 | 00 ~ 15 | 计数器1 | PV(最右4位数) | 每个高速计数器板端口的高速计数器的PV在每次循环后存贮。 数据存贮的格式(BCD或十六进制)可在PC设置(DM 6602和DM 6611)中指定。 |
| IR 201 | IR 233 | 00 ~ 15 | | PV(最左4位数) | |
| IR 202 | IR 234 | 00 ~ 15 | 计数器2 | PV(最右4位数) | |
| IR 203 | IR 235 | 00 ~ 15 | | PV(最左4位数) | |
| IR 204 | IR 236 | 00 ~ 15 | 计数器3 | PV(最右4位数) | |
| IR 205 | IR 237 | 00 ~ 15 | | PV(最左4位数) | |
| IR 206 | IR 238 | 00 ~ 15 | 计数器4 | PV(最右4位数) | |
| IR 207 | IR 239 | 00 ~ 15 | | PV(最左4位数) | |
| IR 208; 计数器1 IR 209; 计数器2 IR 210; 计数器3 IR 211; 计数器4 | IR 240; 计数器1 IR 241; 计数器2 IR 242; 计数器3 IR 243; 计数器4 | 00 ~ 07 | 比较结果:内部输出位00 ~ 07 | | 条件满足时包含CTBL(63)中的操作数指定的位型式。 |
| | | 08 ~ 11 | 比较结果:外部输出位1 ~ 4 | | 条件满足时包含CTBL(63)中的操作数指定的位型式。 |
| | | 12 | 计数器工作标志 | | 0:停止 1:工作 |
| | | 13 | 比较标志 | | 指示是否在进行比较。 0:停止 1:工作 |
| | | 14 | PV上溢/下溢标志 | | 指示是否出现上溢或下溢。 0:正常 1:出现上溢或下溢 |
| | | 15 | SV错误标志 | | 0:正常 1:设定错误 |

| 字 | | 位 | 名称 | 功能 | | |
|--------|-------------|---------|-------------|--|------------|-------------------------|
| 槽1 | 槽2 | | | | | |
| IR 212 | AR 05 | 00 | 高速计数器1复位位 | Z相和软件复位 0:计数器在Z相不复位 1:计数器在Z相复位 | | |
| | | 01 | 高速计数器2复位位 | | | |
| | | 02 | 高速计数器3复位位 | 只有软件复位 0: 计数器不复位 0→1: 计数器复位 | | |
| | | 03 | 高速计数器4复位位 | | | |
| | | 08 | 高速计数器1比较开始位 | | | |
| | | 09 | 高速计数器2比较开始位 | 0→1: 开始比较 1→0: 停止比较 | | |
| | | 10 | 高速计数器3比较开始位 | | | |
| | | 11 | 高速计数器4比较开始位 | | | |
| | | 12 | 高速计数器1停止位 | 0:继续操作 1:停止操作 | | |
| | | 13 | 高速计数器2停止位 | | | |
| | | 14 | 高速计数器3停止位 | | | |
| | | 15 | 高速计数器4停止位 | | | |
| | | IR 213 | AR 06 | 00 | 外部输出1强置置位位 | 0:对输出状态不起作用 1:强置输出ON |
| | | | | 01 | 外部输出2强置置位位 | |
| | | | | 02 | 外部输出3强置置位位 | |
| 03 | 外部输出4强置置位位 | | | | | |
| 04 | 外部输出强制置位允许位 | | | 0:禁止输出1~4强制置位 1:允许输出1~4强制置位 | | |
| SR 254 | | 15 | 内板错误标志 | 0:没有错误 1:错误 安装在槽1或槽2上的内板出错时置ON。 槽1的错误代码存贮在AR 0400 ~ AR 0407中, 槽2的错误代码存贮在AR 0408 ~ AR 0415中。 | | |
| AR 04 | | 00 ~ 07 | 槽1内板的错误代码 | 00 Hex: 正常 | | |
| | | 08 ~ 15 | 槽2内板的错误代码 | 01或02 Hex: 硬件错误 03 Hex: PC设置错误 | | |

有关的 PC 设置设定

| 字 | | 位 | 功能 | 读设定的时间 |
|---------|---------|---------|---|--------|
| 槽 1 | 槽 2 | | | |
| DM 6602 | DM 6611 | 00 ~ 03 | 高速计数器 1 ~ 4 的 PV 存贮的数据格式 0: 8 位数十六进制 (BIN) 1: 8 位数 BCD | 电源上电时 |
| | | 04 ~ 07 | 未使用 | |
| | | 08 ~ 11 | 外部输出 1 ~ 4 的源/吸收设定 0: 源 (PNP) 1: 吸收 (NPN) | |
| | | 12 ~ 15 | 未使用 | |
| DM 6640 | DM 6643 | 00 ~ 03 | 高速计数器 1 的输入模式 0 Hex: 1 × 相位差输入 1 Hex: 2 × 相位差输入 2 Hex: 4 × 相位差输入 3 Hex: 增/减脉冲输入 4 Hex: 脉冲/方向输入 | 开始工作时 |
| | | 04 ~ 07 | 高速计数器 1 的计数频率, 数值范围模式和计数器复位方法。参见下表。 | |
| | | 08 ~ 11 | 高速计数器 2 的输入模式 (参见上面的高速计数器 1 的介绍。) | |
| | | 12 ~ 15 | 高速计数器 2 的计数频率, 数值范围模式和计数器复位方法 (参见上面的高速计数器 1 的介绍。) | |
| DM 6641 | DM 6644 | 00 ~ 03 | 高速计数器 3 的输入模式 (参见上面的高速计数器 1 的介绍。) | |
| | | 04 ~ 07 | 高速计数器 3 的计数频率, 数值范围模式和计数器复位方法 (参见上面的高速计数器 1 的介绍。) | |
| | | 08 ~ 11 | 高速计数器 4 的输入模式 (参见上面的高速计数器 1 的介绍。) | |
| | | 12 ~ 15 | 高速计数器 4 的计数频率, 数值范围模式和计数器复位方法 (参见上面的高速计数器 1 的介绍。) | |

高速计数器的计数频率, 数值范围模式和计数器复位方法

| 数值 | 计数频率 | 数值范围模式 | 计数器复位方法 |
|-------|--------|--------|-----------|
| 0 Hex | 50 KHz | 线性模式 | Z相 + 软件复位 |
| 1 Hex | | | 只有软件复位 |
| 2 Hex | | 环形模式 | Z相 + 软件复位 |
| 3 Hex | | | 只有软件复位 |
| 4 Hex | 50 KHz | 线性模式 | Z相 + 软件复位 |
| 5 Hex | | | 只有软件复位 |
| 6 Hex | | 环形模式 | Z相 + 软件复位 |
| 7 Hex | | | 只有软件复位 |

2-1-7 高速计数器 1~4

高速计数器板通过端口 1~4 计算从旋转编码器输入的脉冲信号，根据计算的脉冲数来输出内部/外部输出的位型式。可独立使用 4 个端口。高速计数器 1~4 执行的处理概述如下。

处理概述

输入信号和输入模式

高速计数器 1~4 可根据信号输入的类型设为不同的输入模式。

相位差模式(计数速度:25KHz 或 250KHz)

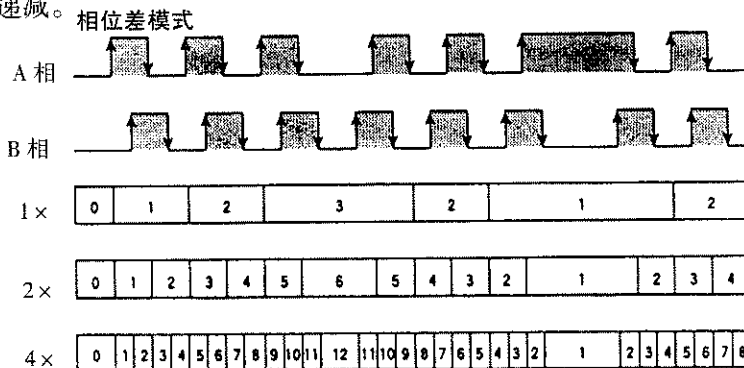
使用有相位差的两相(A相和B相)信号,并乘以 1x, 2x 或 4x 和一个 Z 相信号一起作为输入。根据两相信号的相位差递增或递减计数。

增/减模式(计数速度 50KHz 或 500KHz)

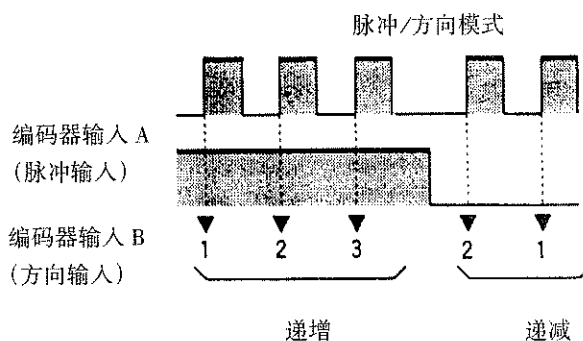
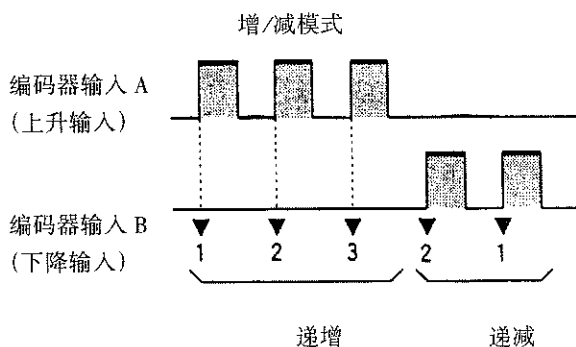
A 相是递增脉冲,B 相是递减脉冲。计数器根据检测到的脉冲递增或递减。

脉冲/方向模式(计数速度:50KHz 或 500KHz)

A 相是脉冲信号,B 相是方向信号。B 相信号置 ON 时计数器递增,置 OFF 时递减。



| A相 | B相 | 1x | 2x | 4x |
|----|----|-----|-----|----|
| ↑ | L | 递增 | 递增 | 递增 |
| H | ↑ | --- | --- | 递增 |
| ↓ | H | --- | 递增 | 递增 |
| L | ↓ | --- | --- | 递增 |
| L | ↑ | --- | --- | 递减 |
| ↑ | H | --- | 递减 | 递减 |
| H | ↓ | --- | --- | 递减 |
| ↓ | L | 递减 | 递减 | 递减 |



数值范围

高速计数器 1~4 计数的值可使用下列两种范围设定计数：

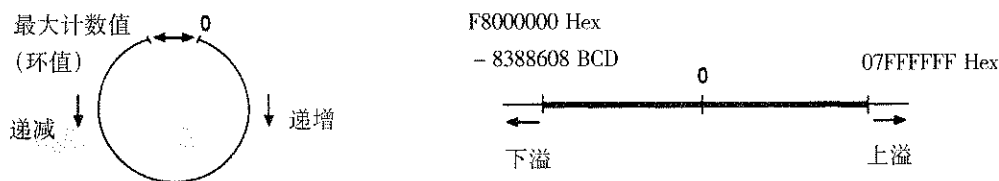
环形模式

在环形模式中,可使用 CTBL(63)设置数值范围的最大值,当递增计数超过最大值时,返回零,计数不会为负。同样,如果从 0 递减计数,返回最大值。通过设定最大值(即环值)为 1~8388607 BCD 或 1~7FFFFFFF Hex 决定环上的点数。最大值设为 8388607 时,范围为 0~8388607 BCD。

线性模式

在线性模式中,计数范围总是 -8388608~8388607 BCD 或 F8000000~07FFFFFFF Hex。如果递减计数小于 -8388608 BCD 或 F8000000 Hex,产生下溢;如果递增计数大于 8388607 BCD 或 07FFFFFFF Hex,产生上溢。

环形模式



出现上溢时,计数的 PV 保持为 08388607 BCD 或 07FFFFFFF Hex。出现下溢时,保持为 F8388608 BCD 或 F8000000 Hex。在两种情况下,停止计数和比较。但是比较表保存在内存中。下面所示的 PV 上溢/下溢标志置 ON 表示上溢或下溢。

| 计数器 | PV上溢/下溢标志 | |
|--------|-----------|----------|
| | 槽 1 | 槽 2 |
| 高速计数器1 | IR 20814 | IR 24014 |
| 高速计数器2 | IR 20914 | IR 24114 |
| 高速计数器3 | IR 21014 | IR 24214 |
| 高速计数器4 | IR 21114 | IR 24314 |

重新启动计数操作时,使用下面所给的复位方法复位高速计数器 1 和 2。(程序执行开始和完成时计数器将自动复位。)

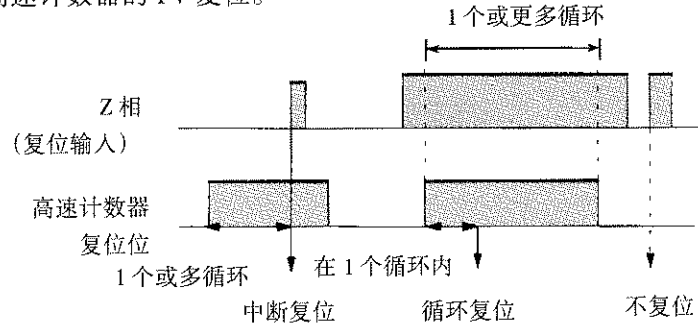
复位方法

可设置下列两种方法以决定计数器 PV 复位的时间(即设为 0):

- Z 相信号 + 软件复位
- 软件复位

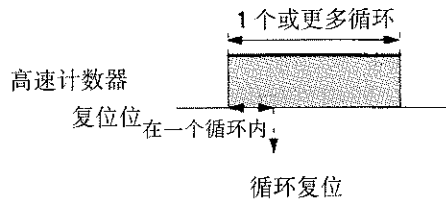
Z 相信号(复位输入) + 软件复位

在相应的高速计数器复位位 (见下图) 置 ON 后的第一个 Z 相信号的上升沿时, 高速计数器的 PV 复位。



软件复位

高速计数器复位位置 ON 时 PV 复位。高速计数器 1~4 的每一个都有单独的复位位。



下表给出高速计数器 1~4 复位位。

| 计数器 | 复位位 | |
|--------|----------|---------|
| | 槽1 | 槽2 |
| 高速计数器1 | IR 21200 | IR 0500 |
| 高速计数器2 | IR 21201 | IR 0501 |
| 高速计数器3 | IR 21202 | IR 0502 |
| 高速计数器4 | IR 21203 | IR 0503 |

高速计数器 1~4 的复位位每个循环只刷新一次。复位位要可靠地读入, 必须置 ON 至少一个周期。

注 PV 复位时不改变比较表注册和比较执行状态。如果在复位前正在执行比较, 将继续比较。

高速计数器中断的检查方法

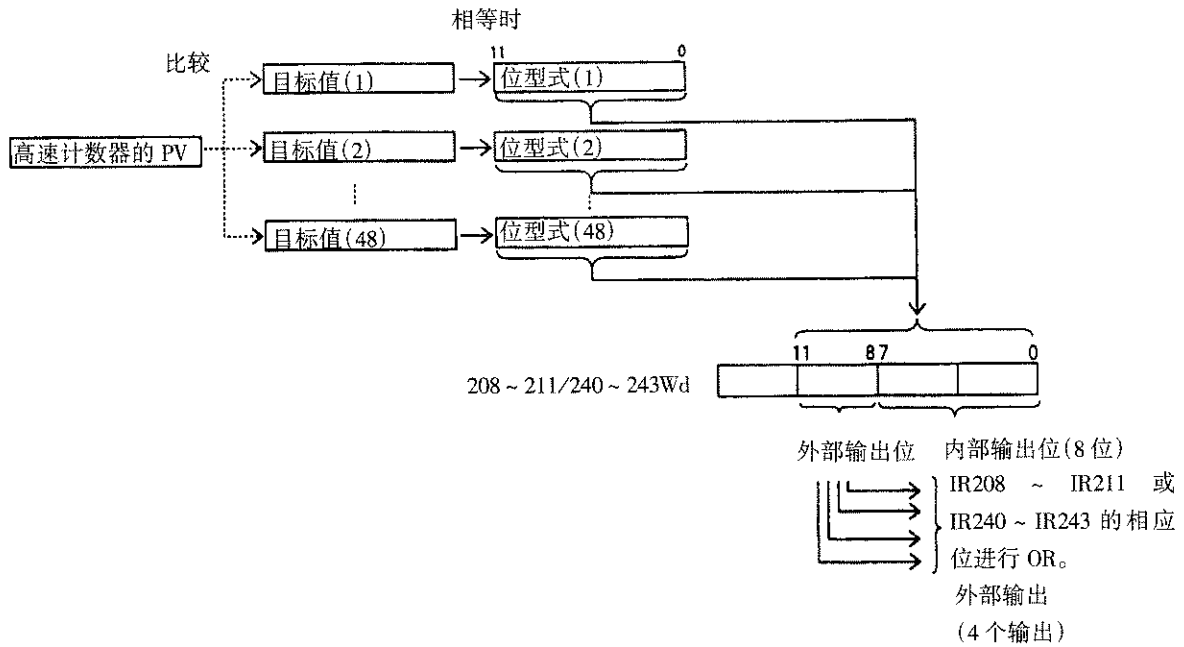
下列两种方法可用于检查高速计数器 1~4 的 PV。(与用于内置高速计数器 0 的方法相同。)

- 目标值方法
- 范围比较方法

每种方法的介绍参见 31 页。

对于目标值方法, 在比较表中最多可注册 48 个目标值。当计数器的 PV 等于

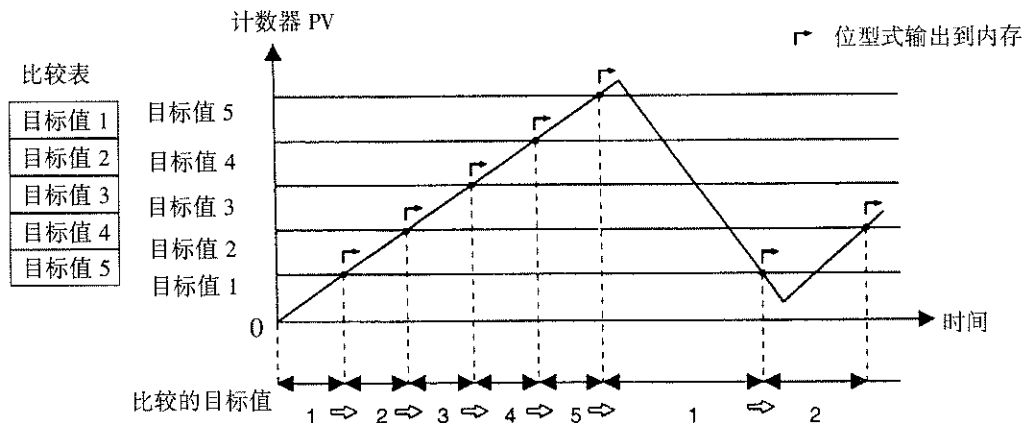
48 个注册目标值中的一个时, 相应的位型式 (1 ~ 48) 将输出到内存中的指定位。

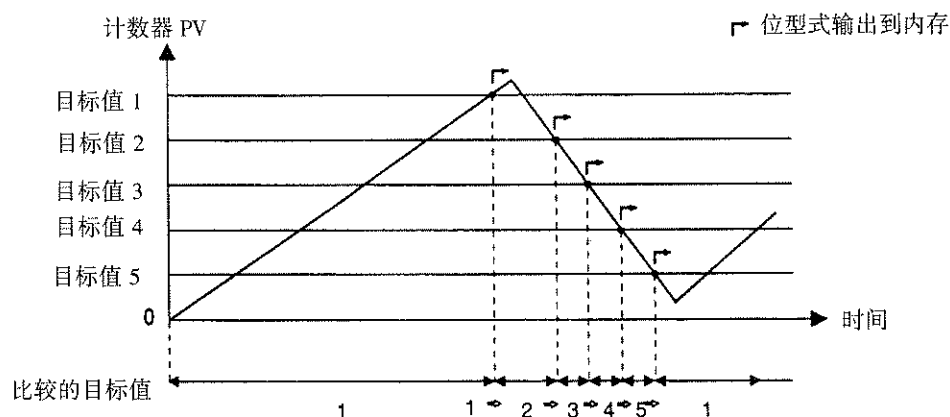


使用目标值时, 按照比较表的顺序与每个目标值进行比较, 直到满足所有数值, 然后比较返回表中的第一个数值。对于高速计数器板, 是递增还是递减 PV 达到目标值是没有区别的。

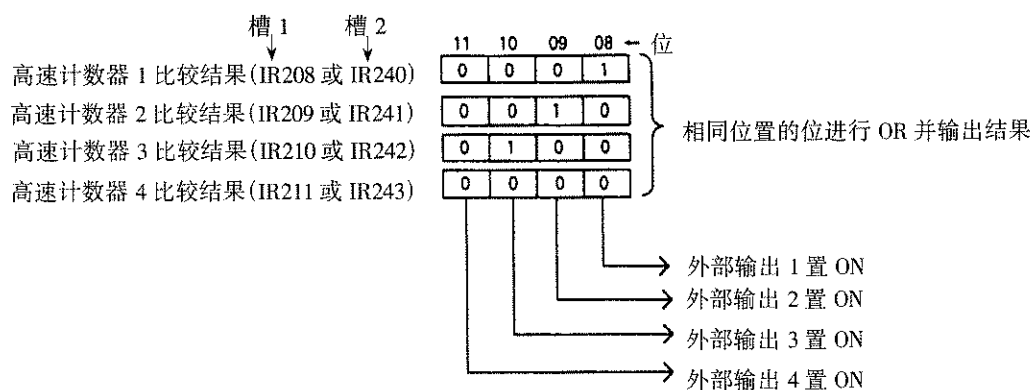
注 对于 CPU 单元中的高速计数器 0 或脉冲 I/O 板或绝对编码器接口板上的高速计数器 1 或 2, 比较表中包含子程序号的字的最左位决定目标值是递增 PV 还是递减 PV 时有效。

下图中显示比较表操作和位型式输出的例子。

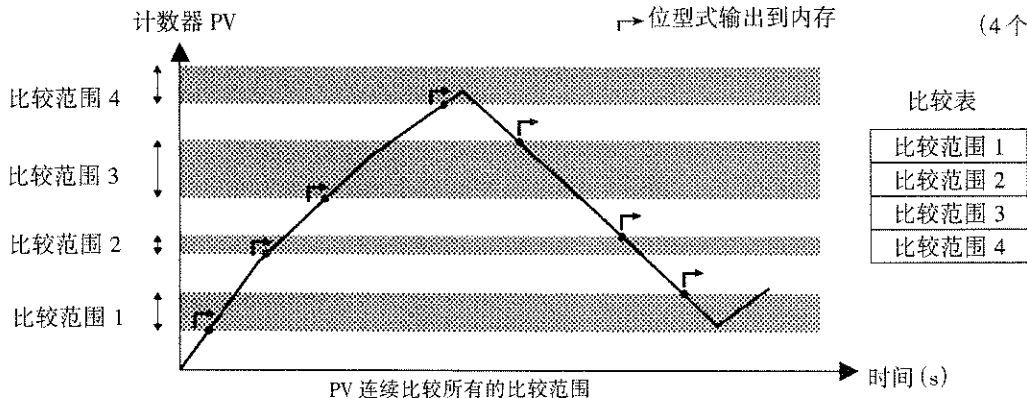
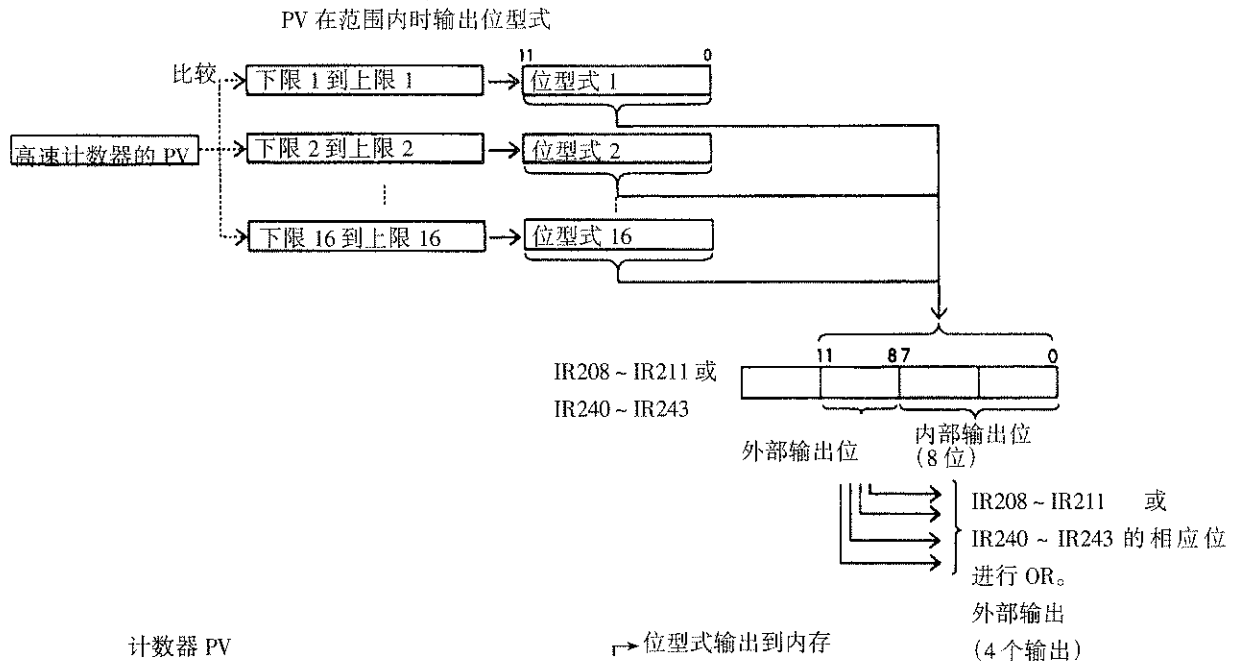




比较值 1~48 和位型式 1~48 在目标表中注册。对于每一个位型式的位 00~11, 位 0~7 作为内部输出位存贮, 位 08~11 作为外部输出位存贮。如下图所示, 外部位型式中的位用于高速计数器 1~4 相应位的 OR 操作, 结果就输出作为外部输出 1~4。

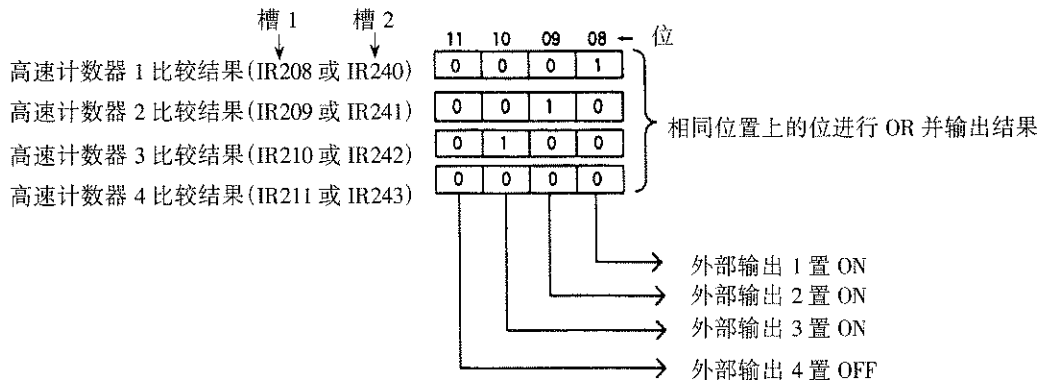


对于范围比较方法, 在比较表中注册 16 个比较范围。在计数器的 PV 第一次进入范围 1~16 中的一个上限和下限之间时, 相应的位型式 (1~16) 将输出一次到内存中的指定位。



范围 1~16 的下限和上限和位型式 1~16 在范围比较表中注册。每一个位型式的位 0~11 中,位 0~7 作为内部输出位存贮,位 8~11 作为外部输出位存贮。如下图所示,外部位型式中的位用于高速计数器 1~4 相应位的 OR 操作,其结果输出作为外部输出 1~4。

例子



外部输出 1~4 由高速计数器 1~4 的比较结果位 08~11 的相应位 (即有相同位号的位) 上进行 OR 控制的。用户必须决定对每一个可能的比较结果, 哪个输出置 ON, 并且设置位型式, 以使 OR 操作产生需要的结果。

注 内置高速计数器 (高速计数器 0) 和脉冲 I/O 板支持范围 1~8 的范围比较标志。但是, 高速计数器板不支持这些标志。内部位型式必须用于产生相同类型的输出结果。

读高速计数器状态

下列两种方法可用于读高速计数器 1~4 的状态。

- 使用 CPU 单元内存字
- 使用 PRV(62)

使用 CPU 单元内存字

在 CPU 单元中表示高速计数器 1~4 的状态的内存区字和位如下所示。

内板错误代码

| 字 | | 位 | 功能 | |
|-------|----|---------|----|-----------------------------------|
| 槽1 | 槽2 | | | |
| AR 04 | | 00 ~ 07 | 槽1 | 存贮下列2位数错误代码 00 Hex: 正常 |
| | | 08 ~ 15 | 槽2 | 01或02 Hex: 硬件错误 03 Hex: PC设置错误 |

操作状态字

| 高速计数器 | 字 | |
|--------|--------|--------|
| | 槽1 | 槽2 |
| 高速计数器1 | IR 208 | IR 240 |
| 高速计数器2 | IR 209 | IR 241 |
| 高速计数器3 | IR 210 | IR 242 |
| 高速计数器4 | IR 211 | IR 243 |

每个操作状态字中的位的功能如下:

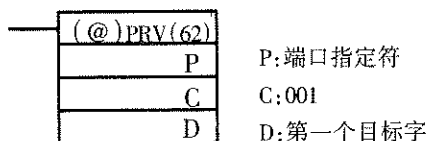
| 位 | 功能 |
|---------|--|
| 00 ~ 07 | 比较结果: 内部输出位 |
| 08 ~ 11 | 比较结果: 输出 1~4 的外部输出位 高速计数器 1~4 的所有相同位位置上的位的 OR 操作结果将被输出。(见注) |
| 12 | 计数器工作标志 (0: 停止; 1: 运行) |
| 13 | 比较标志 (0: 停止; 1: 运行) |
| 14 | PV 上溢/下溢标志 (0: 否; 1: 是) |
| 15 | SV 错误标志 (0: 正常; 1: 错误) |

注 下表显示了外部输出 1~4 之间的关系和比较结果外部输出位。

| 高速计数器 | 外部输出 | 槽1 | 槽2 |
|-------|-------|----------------------|----------------------|
| 计数器1 | 外部输出1 | IR208 ~ IR211的位08的OR | IR240 ~ IR243的位08的OR |
| 计数器2 | 外部输出2 | IR208 ~ IR211的位09的OR | IR240 ~ IR243的位09的OR |
| 计数器3 | 外部输出3 | IR208 ~ IR211的位10的OR | IR240 ~ IR243的位10的OR |
| 计数器4 | 外部输出4 | IR208 ~ IR211的位11的OR | IR240 ~ IR243的位11的OR |

使用 PRV(62)

高速计数器 1~4 的状态可用 PRV(62)以下列所示方式读入。



| 高速计数器 | P中指定的值 | |
|--------|--------|-----|
| | 槽 1 | 槽 2 |
| 高速计数器1 | 101 | 001 |
| 高速计数器2 | 102 | 002 |
| 高速计数器3 | 103 | 003 |
| 高速计数器4 | 104 | 004 |

存储高速计数器 1~4 的状态的 D 的单独位的意义如下表所示。

| 位 | 功能 |
|-------|---|
| 00~07 | 比较结果:内部输出位 |
| 08~11 | 比较结果;输出1~4的外部输出位 高速计数器1~4的所有相同位位置上的位的OR操作结果将被输出。(见注) |
| 12 | 计数器工作标志(0:停止;1:运行) |
| 13 | 比较标志(0:停止;1:运行) |
| 14 | PV上溢/下溢标志(0:否;1:是) |
| 15 | SV错误标志(0:正常;1:错误) |

使用高速计数器的步骤

决定计数速率，输入模式，复位方法，数值范围模式。高速计数器数据的 PV 存储的格式和外部输出方法。

计数速率:50KHz/500KHz
输入模式:
相位差模式;脉冲/方向模式;增/减模式
复位方法:Z 相 + 软件复位;软件复位
数值范围模式:环形模式或线性模式
高速计数器数据的 PV 存储的格式: 8 位数 BCD 或 8 位数十六进制
外部输出方法:
晶体管输出的源或吸收切换

设置输入电压
(板上开关)

安装板和输入接线。

PC 设置
(槽 1:DM6602,DM6640,DM6641
槽 2:DM6611,DM6643,DM6644)

计数速率:50KHz/500KHz
输入模式:
相位差模式;增/减模式;脉冲/方向模式
复位方法:
Z 相 + 软件复位;软件复位
数值范围:环形模式或线性模式
外部输出方法:
晶体管输出的源或吸收切换
高速计数器数据的 PV 存储的格式:
8 位数 BCD;8 位数十六进制

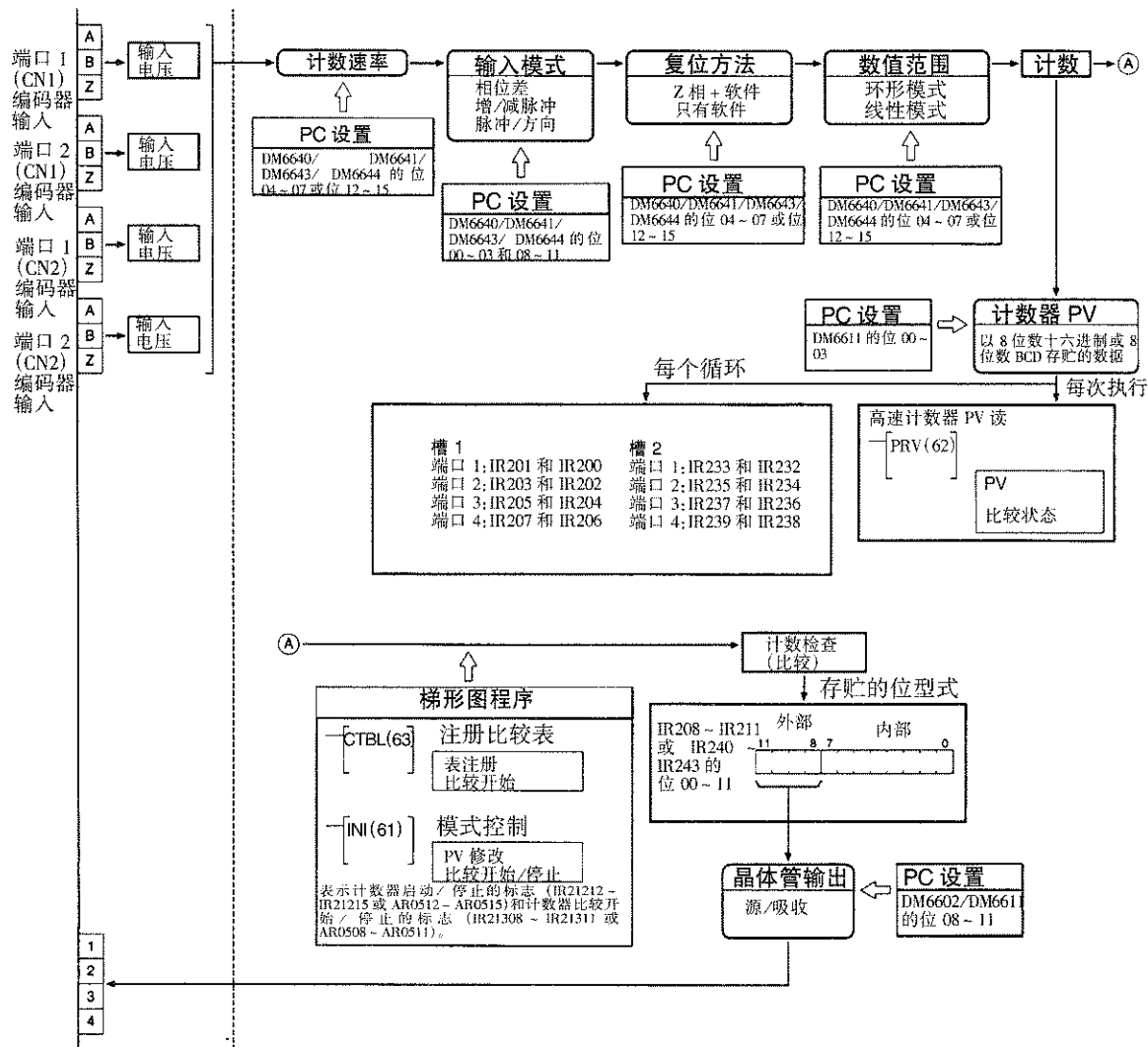
决定计数检查(比较)方法和内部/外部位型式。

计数检查方法:目标值或范围比较
满足条件时的输出位型式:内部和外部输出位

梯形图程序

注册比较表(CIBL(63)):
端口规定;比较表注册;比较开始
模式控制(INI(61)):
端口规定;PV 修改;比较开始
高速计数器 PV 读(PRV(62)):
读高速计数器 PV 和比较状态。

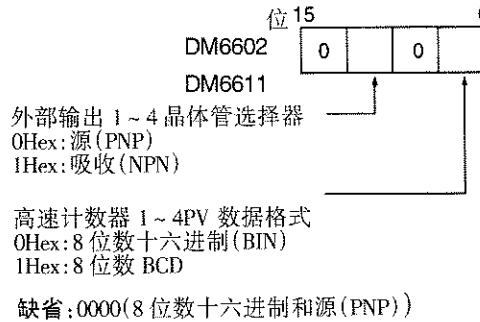
高速计数器功能



PC 设置预设定

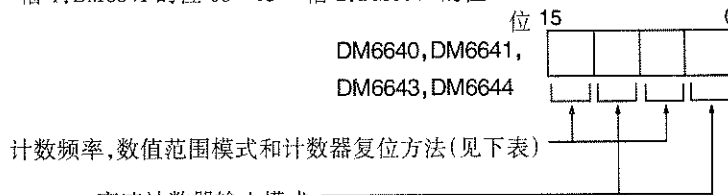
为使用高速计数器 1~4,在编程模式中进行下列设定。

外部输出的数据格式和源/吸收设定



输入模式,计数频率,数值范围模式和计数器复位方法

- 高速计数器 1
槽 1:DM6640 的位 00~07 槽 2:DM6643 的位 00~07
- 高速计数器 2
槽 1:DM6640 的位 08~15 槽 2:DM6643 的位 08~15
- 高速计数器 3
槽 1:DM6641 的位 00~07 槽 2:DM6644 的位 00~07
- 高速计数器 4
槽 1:DM6641 的位 08~15 槽 2:DM6644 的位 08~15



- 高速计数器输入模式
 0Hex:1x 相位差输入
 1Hex:2x 相位差输入
 2Hex:4x 相位差输入
 3Hex:增/减脉冲输入
 4Hex:脉冲/方向输入
- 缺省:0000(1x 相位差输入,50kHz,线性模式,Z 相 + 软件复位)

计数频率,数值范围模式和复位方法

| 值 | 计数频率 | 数值范围模式 | 计数器复位方法 |
|-------|---------|--------|-----------|
| 0 Hex | 50 KHz | 线性模式 | Z相 + 软件复位 |
| 1 Hex | | | 只有软件复位 |
| 2 Hex | | 环形模式 | Z相 + 软件复位 |
| 3 Hex | | | 只有软件复位 |
| 4 Hex | 500 KHz | 线性模式 | Z相 + 软件复位 |
| 5 Hex | | | 只有软件复位 |
| 6 Hex | | 环形模式 | Z相 + 软件复位 |
| 7 Hex | | | 只有软件复位 |

用法

- 高速计数器如下编程:
- 只要进行了有效设定就开始计数操作。

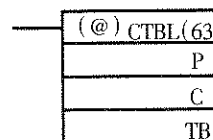
- 电源上电时和开始或停止执行程序时, PV 复位为 0。
- 单独计数操作不启动比较表的比较操作。
- 可使用下表所示的字监视 PV。

| 高速计数器 | 字 | |
|---------|----------------|----------------|
| | 槽 1 | 槽 2 |
| 高速计数器 1 | IR 200, IR 201 | IR 232, IR 233 |
| 高速计数器 2 | IR 202, IR 203 | IR 234, IR 235 |
| 高速计数器 3 | IR 204, IR 205 | IR 236, IR 237 |
| 高速计数器 4 | IR 206, IR 207 | IR 238, IR 239 |

启动比较操作

用 CTBL(63) 在 CQM1H 中注册比较表并启动比较。也可用有关控制位 (槽 1 的 IR 21208 ~ IR 21211, 槽 2 的 AR 0508 ~ AR 0511) 启动比较。

用 CTBL(63) 启动比较



- P: 端口
- C: 模式
- 000: 目标值表注册和比较启动
- 001: 范围比较表注册和比较启动
- 002: 只有目标值表注册
- 003: 只有范围比较表注册
- TB: 比较表的第一个字

| 高速计数器 | P 中指定的值 | |
|---------|---------|-----|
| | 槽 1 | 槽 2 |
| 高速计数器 1 | 101 | 001 |
| 高速计数器 2 | 102 | 002 |
| 高速计数器 3 | 103 | 003 |
| 高速计数器 4 | 104 | 004 |

C 值设为 000, 注册目标值比较表, 设为 001 注册范围比较表。完成注册后开始比较。执行比较时, 位型式存贮由比较表决定的内部输出位和外部输出位。比较表注册的详细内容参见 CTBL(63) 的说明。

注 C 值设为 002 注册目标值比较表, 设为 003 注册范围比较表, 但这些值的比较不会自动启动。控制位或 INI(61) 必须用于启动比较操作。

用控制位启动比较

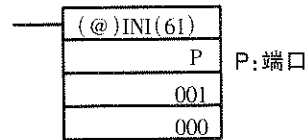
对应于槽 1 的 IR 21208 ~ IR 21211 或槽 2 的 AR 0508 ~ AR 0511 的高速计数器的位置 ON 时启动比较操作。必须预先注册比较表。不能在编程模式下执行比较。

注 高速计数器板以位型式把比较结果输出到内存中的指定位, 而且不执行中断子程序。位型式包括内部位和外部位, 外部位在外围输出 1 ~ 4 上输出。

停止比较操作

执行如下所示的 INI(61) 中止比较操作, 也可用控制位完成中止比较。

用 INI(61)停止比较



| 高速计数器 | P 中设定的值 | |
|---------|---------|-----|
| | 槽 1 | 槽 2 |
| 高速计数器 1 | 101 | 001 |
| 高速计数器 2 | 102 | 002 |
| 高速计数器 3 | 103 | 003 |
| 高速计数器 4 | 104 | 004 |

用控制位停止比较

对应于槽 1 的 IR 21208 ~ IR 21211 或槽 2 的 AR0508 ~ AR 0511 的高速计数器的位置 OFF 时停止比较操作。

- 注
1. 以端口号作第一操作数, 000(执行比较)作第二操作数执行 INI(61), 或把控制位的状态从 0 修改为 1, 重新启动比较。
 2. 一旦表被注册, 就在整个操作期间(即运行程序时)保存在 CQM1H 中, 直到注册了新表。

读 PC

下列两种方法可用于读高速计数器 1~4 的 PV:

- 读内存中的 PV 字
- 使用 PRV(62)

读内存中的 PV 字

高速计数器 1~4 的 PV 用下列方法存贮在内存中。PV 数据存贮的格式由槽 1 的 DM 6602 和槽 2 的 DM 6611 的位 00 ~ 03 的设定决定。缺省设定是 8 位数十六进制。

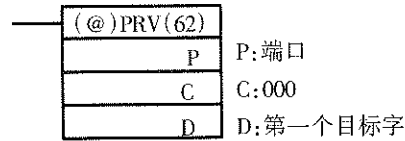
| 槽 1: | 最左 4 位数 | 最右 4 位数 | 线性模式 | 环形模式 |
|------|---------|---------|-----------------------------------|-------------------------|
| 端口 1 | IR201 | IR200 | 8 位数 Hex: F8000000 ~ 07FFFFFF Hex | 00000000 ~ 07FFFFFF Hex |
| 端口 2 | IR203 | IR202 | 8 位数 BCD: F8388608 ~ 08388607 | 00000000 ~ 08388607 |
| 端口 3 | IR205 | IR204 | (如果是负数, 最左的数字为 F。) | |
| 端口 4 | IR207 | IR206 | | |

| 槽 2: | 最左 4 位数 | 最右 4 位数 | 线性模式 | 环形模式 |
|------|---------|---------|-----------------------------------|-------------------------|
| 端口 1 | IR233 | IR232 | 8 位数 Hex: F8000000 ~ 07FFFFFF Hex | 00000000 ~ 07FFFFFF Hex |
| 端口 2 | IR235 | IR234 | 8 位数 BCD: F8388608 ~ 08388607 | 00000000 ~ 08388607 |
| 端口 3 | IR237 | IR236 | (如果是负数, 最左的数字为 F。) | |
| 端口 4 | IR239 | IR238 | | |

注 这些字每个循环只刷新一次, 因此所读的值可能与实际 PV 有些差别。

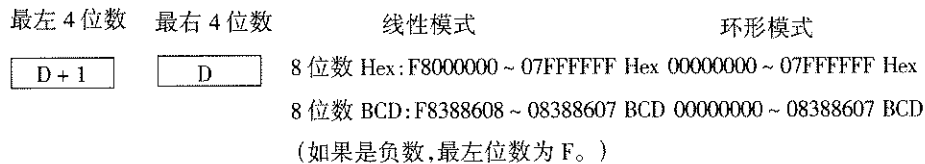
使用 PRV(62)

PRV(62)也可用于读高速计数器 1~4 的 PV。



| 高速计数器号 | P中指定的值 | |
|--------|--------|-----|
| | 槽1 | 槽2 |
| 高速计数器1 | 101 | 001 |
| 高速计数器2 | 102 | 002 |
| 高速计数器3 | 103 | 003 |
| 高速计数器4 | 104 | 004 |

高速计数器 1~4 的 PV 按下图所示存贮。



注 执行 PRV(62)时读当前 PV。

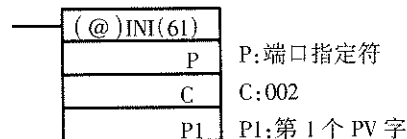
下列两种方法可用于修改高速计数器 1~4 的 PV。

- 用一种复位方法复位计数器(即计数器设为 0)
- 使用 INI(61)

下面是 INI(61)用法的介绍。复位方法用法介绍参见 63 页复位方法。

用 INI(61)修改 PV

INI(61)用于修改高速计数器 1~4 的 PV。



| 高速计数器号 | P中指定的值 | |
|--------|--------|-----|
| | 槽1 | 槽2 |
| 高速计数器1 | 101 | 001 |
| 高速计数器2 | 102 | 002 |
| 高速计数器3 | 103 | 003 |
| 高速计数器4 | 104 | 004 |



注 在目标值比较表中最终目标值匹配后,比较过程自动回到表中的第一个目标值。

修改 PV

停止并重启动计数操作

因此,比较顺序完成后,可通过初始化 PV 重复此处理。

能通过把控制位置 ON 停止高速计数器 1~4 中的一个计数操作,计数器的 PV 保持。

可通过把槽 1 的 IR 212 或槽 2 的 AR 05 的位 12~15 置 ON 来停止计数操作。这些位对应高速计数器 1~4,这些位置 OFF 重启动计数操作。高速计数器从它停止的值开始重启动。

注 计数器操作标志可用于决定计数操作运行还是停止(0:停止;1:操作)。

| 高速计数器 | 计数器操作标志 | |
|---------|----------|----------|
| | 槽 1 | 槽 2 |
| 高速计数器 1 | IR 20812 | IR 24012 |
| 高速计数器 2 | IR 20912 | IR 24112 |
| 高速计数器 3 | IR 21012 | IR 24212 |
| 高速计数器 4 | IR 21112 | IR 24312 |

例

下例示出安装在槽 2 上的高速计数器板上的高速计数器 1 的用法。执行目标值比较把存贮在内存中的内部/外部位型式中的位根据计数器的 PV 而置 ON。内部输出位状态用于控制接点脉冲输出频率。

程序中的复位位保持为 ON,使达到最后目标值后,计数器 PV 在 Z 相信号时复位。

运行程序前,如下设定 PC 设置,并重启动 CQM1H 以便允许 DM 6611 中进行新设定。

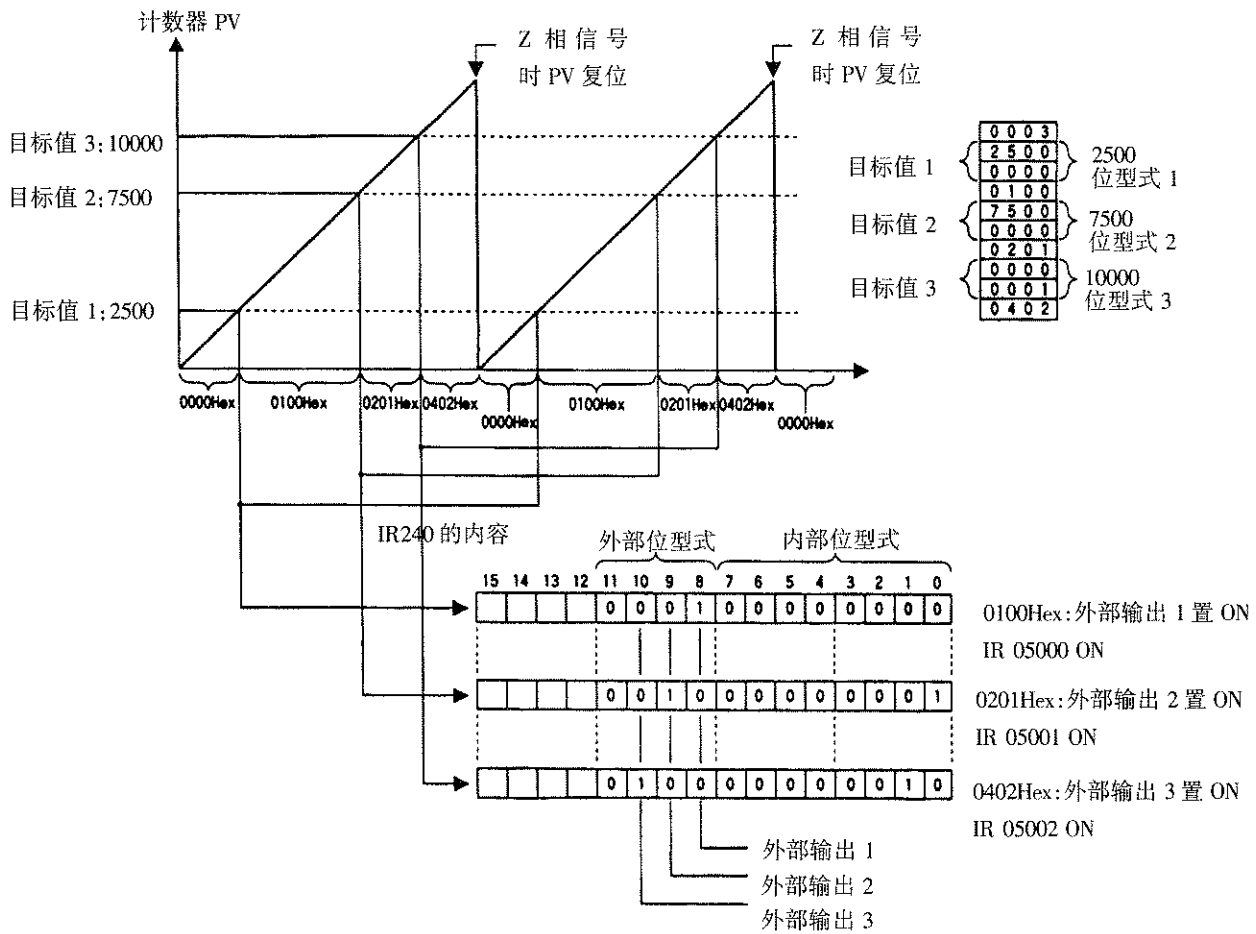
DM 6611: 0001(外部输出 1~4 源输出,高速计数器 1~4 PV 存贮为 8 位数 BCD)。

DM 6643: 0003(高速计数器 1:计数频率 50 kHz;线性模式;Z 相信号 + 软件复位;增/减模式)。

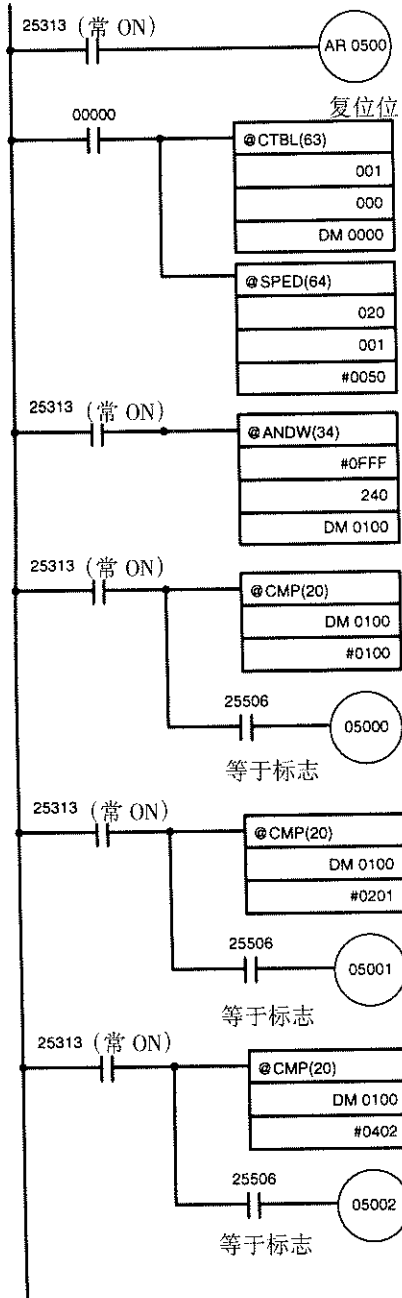
PV 达到 2500 时,IR 05000 将置 ON,而且外部输出 1 将置 ON。

PV 达到 7500 时,IR 05001 将置 ON,而且外部输出 2 将置 ON。

PV 达到 10000 时,IR 05002 将置 ON,而且外部输出 3 将置 ON。



如下面程序例子所示,接点脉冲输出频率变化过程为:执行 CTBL(63)时设为 500Hz,并在 IR 05000,IR 05001 和 IR 05002 置 ON 时,分别为 200Hz,100Hz 和 0Hz。



高速计数器的复位位保持为 ON。

指定槽 2 中的高速计数器 1 的目标比较,注册目标值比较表,并开始在 DM 0000 启动比较。

从输出位置 02 以 500Hz 设定连续的接点脉冲输出并启动脉冲输出。

IR 240 中存贮的位型式的内容与运算,并把结果存贮在 DM 0100 中。

DM 0100 与 # 0100 作比较。

如果 DM 0100 包含 # 0100 则把 IR 05000 置 ON。

DM 0100 与 # 0201 作比较。

如果 DM 0100 包含 # 0201,则把 IR 05001 置 ON。

DM 0100 与 # 0402 作比较。

如果 DM 0100 包含 # 0402,则把 IR 05002 置 ON。

DM 0000: 0003—3 个比较条件

DM 0001: 2500—目标值 1:2,500

DM 0002: 0000

DM 0003: 0100—位型式(1)

DM 0004: 7500—目标值 2:7,500

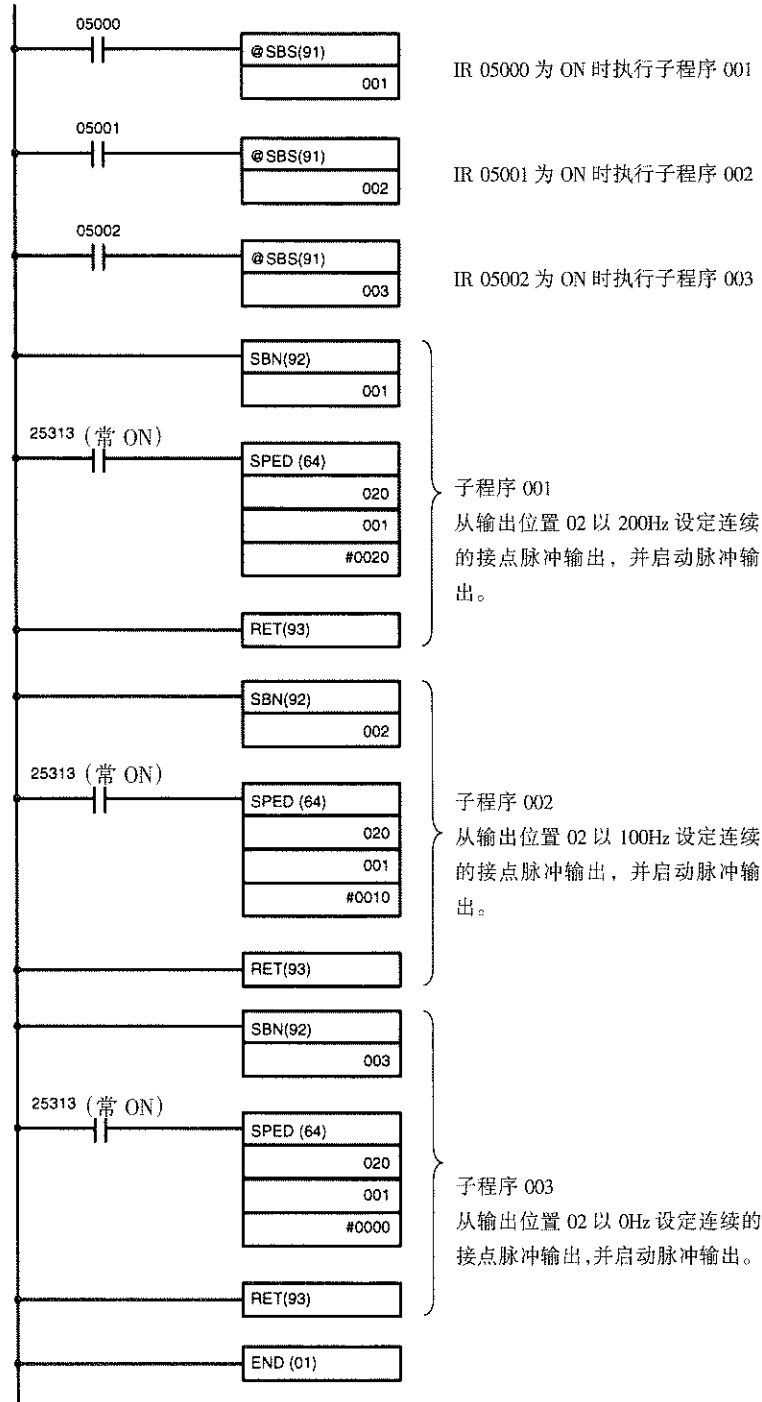
DM 0005: 0000

DM 0006: 0201—位型式(2)

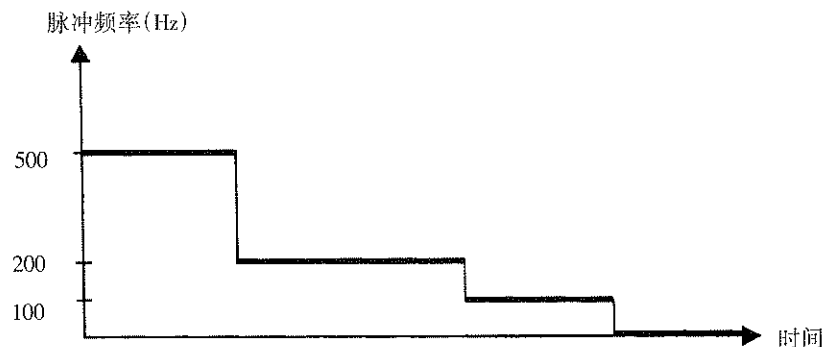
DM 0007: 0000—目标值 3:10,000

DM 0008: 0001

DM 0009: 0402—位型式(3)



执行程序时操作如下所示。



2-2 脉冲 I/O 板

2-2-1 型号

| 名称 | 型号 | 规格 |
|--------|-------------|-----------------|
| 脉冲I/O板 | CQM1H-PLB21 | 两个脉冲输入点和两个脉冲输出点 |

2-2-2 功能

脉冲输入 1 和 2

脉冲 I/O 板是支持两个脉冲输入和两个脉冲输出的内板。

脉冲输入 1 和脉冲输入 2 可用作高速计数器，对以 50KHz(相位信号) 或 25KHz(相位差) 的脉冲输入计数。可根据计数器的当前值 (PV) 执行中断处理。

输入模式

下列三种输入模式可用：

- 相位差模式 (4 ×)
- 脉冲/方向模式
- 增/减模式

中断

板可设定使之在高速计数器的值等于指定目标值时或 PV 处在指定比较范围内时执行中断子程序。

脉冲输出 1 和 2

从端口 1 和端口 2 可输出两个 10Hz ~ 50KHz 脉冲。可使用固定或可变占空率。

- 固定占空率可平滑地从 10Hz 到 50KHz 提高或降低输出频率。
- 可变占空率允许脉冲输出用范围从 1% 到 99% 的占空率执行。

注

脉冲输入和脉冲输出可以同时执行，但不能同时使用所有的高速计数器和脉冲输出功能。PC 设置 (DM 6611) 中的端口模式设定 (高速计数器模式/简单定位模式) 将决定哪个允许具有所有功能。

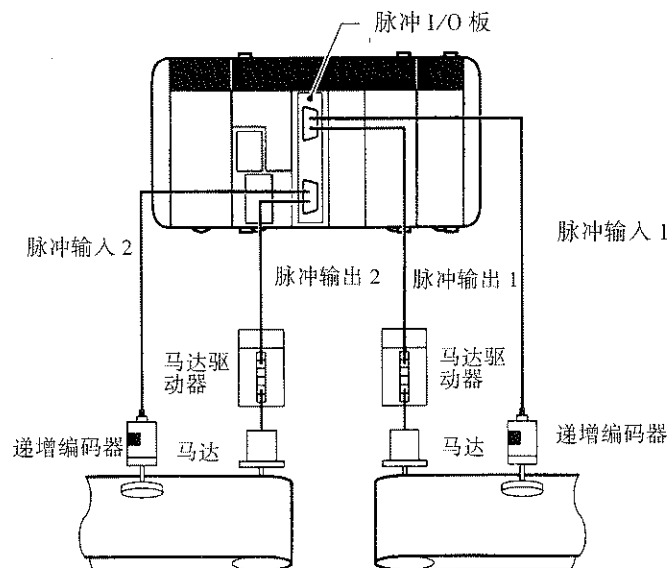
端口 1 和 2

两个脉冲输入 (高速计数器) 和两个脉冲输出可通过端口 1 和 2 同时使用。在 PC 设置 (DM 6611) 中必须输入合适的端口模式设定以决定哪个具有优先功能。

| 模式 | 内容 | 高速计数器功能 | | 脉冲输出功能 | | | DM 6611 设定 |
|---------|---|---------------|---------------------|----------------------|---------------------|---------------------|------------|
| | | 用 PRV(62) 读PV | 用 CTBL(63) 的高速计数器中断 | 无梯形加速/减速 (SPED (64)) | 相同加速度/减速度 (PLS2(-)) | 单独加速度/减速度 (ACC(-)) | |
| 高速计数器模式 | 高速计数器具有优先级。允许所有高速计数器功能。脉冲输出的梯形加速度/减速度有限制。 | 是 | 是 | 是 | | 模式0禁止(模式1~3允许) 见注1。 | 0000 Hex |
| 简单定位模式 | 脉冲输出具有优先级。允许所有脉冲输出功能。高速计数器中断禁止。 | 是 | 否 | 是 | 是 | 是 | 0001 Hex |

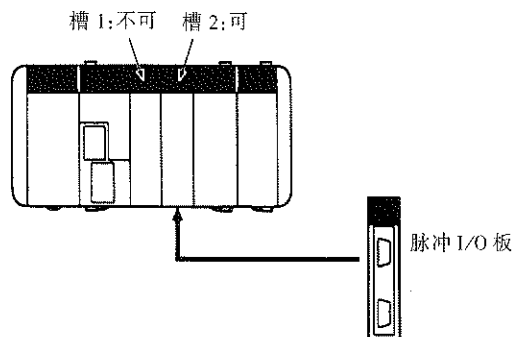
- 注 1. 模式 0: 加速 + 独立模式; 模式 1: 加速 + 连续模式; 模式 2: 减速 + 独立模式; 模式 3: 减速 + 连续模式。
2. 端口 1 和 2 的端口模式总是设为相同模式, 即高速计数器模式或简单定位模式。不能为每个端口单独设置模式。

2-2-3 系统配置



2-2-4 适用的内板槽

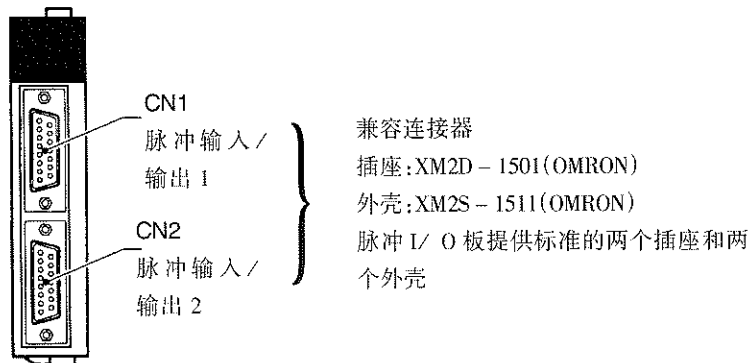
脉冲 I/O 板只能安装在 CQM1H CPU51/61 CPU 单元的槽 2(右槽)上。



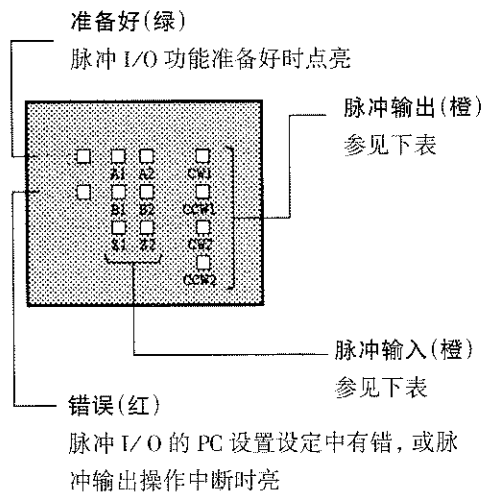
2-2-5 名称和功能

CQM1H-PLB21 脉冲 I/O 板具有脉冲输入 1 和脉冲输出 1 的 CN1 连接器,以及脉冲输入 2 和脉冲输出 2 的 CN2 连接器。

CQM1H-PLB21 脉冲 I/O 板



LED 指示灯



脉冲输出指示灯

| 指示灯 | 端口 | 功能 |
|------|------|------------------|
| CW1 | 端口 1 | CW 脉冲输出到端口 1 时亮 |
| CCW1 | | CCW 脉冲输出到端口 1 时亮 |
| CW2 | 端口 2 | CW 脉冲输出到端口 2 时亮 |
| CCW2 | | CCW 脉冲输出到端口 2 时亮 |

脉冲输入指示灯

| 端口1 | 端口2 | 功能 |
|-----|-----|------------------|
| A1 | A2 | 端口上的A相脉冲输入为ON时亮。 |
| B1 | B2 | 端口上的B相脉冲输入为ON时亮。 |
| Z1 | Z2 | 端口上的Z相脉冲输入为ON时亮。 |

2-2-6 规格

高速计数器规格

指令

| 指 令 | 控 制 | 意 义 |
|-------------|----------------|--------------------------------|
| (@)CTBL(63) | 范围比较表注册 + 比较开始 | 注册范围比较表并开始比较 |
| | 目标值表注册 + 比较开始 | 注册目标值表并开始比较 |
| | 范围比较表注册 | 注册范围比较表 |
| | 目标值表注册 | 注册目标值表 |
| (@)INI(61) | 比较开始 | 用注册的比较表启动比较 |
| | 比较停止 | 停止比较 |
| | 修改PV | 修改高速计数器的PV |
| (@)PRV(62) | 读PV | 读高速计数器的PV |
| | 读状态 | 读高速计数器的状态 |
| | 读范围比较结果 | 读范围比较结果 |
| (@)INT(89) | 屏蔽所有中断 | 屏蔽所有中断,例如输入中断,间隔定时器中断和高速计数器中断。 |
| | 清除中断屏蔽 | 清除中断屏蔽 |

脉冲输入的有关标志和控制位

使用脉冲 I/O 板时的内板槽 2 的位

| 字 | 位 | 名 称 | | 功 能 |
|-------|---------|------|------------|---------------------------------------|
| IR232 | 00 ~ 15 | 端口 1 | PV字(最右4位数) | 脉冲I/O板的每个端口的高速计数器的PV在每次循环后用8位数BCD值存贮。 |
| IR233 | 00 ~ 15 | | PV字(最左4位数) | |
| IR234 | 00 ~ 15 | 端口 2 | PV字(最右4位数) | |
| IR235 | 00 ~ 15 | | PV字(最左4位数) | |

SR 区位

| 字 | 位 | 名 称 | 功 能 |
|-------|----|----------------|--|
| SR252 | 01 | 高速计数器1复位位(端口1) | Z相和软件复位 0: 计数器在Z相不复位 1: 计数器在Z相复位 |
| | 02 | 高速计数器2复位位(端口2) | 仅软件复位 0: 计数器不复位 0→1: 计数器复位 |

AR 区标志

| 字 | 位 | 名称 | | 功能 | |
|-------|---------------|--|--------------|------------------------------|-------------------------------|
| AR 05 | 00 | 端口1 | 高速计数器1范围比较标志 | 满足第一个条件时ON | 高速计数器用于范围比较时,当满足相应条件时标志置ON。 |
| | 01 | | | 满足第二个条件时ON | |
| | 02 | | | 满足第三个条件时ON | |
| | 03 | | | 满足第四个条件时ON | |
| | 04 | | | 满足第五个条件时ON | |
| | 05 | | | 满足第六个条件时ON | |
| | 06 | | | 满足第七个条件时ON | |
| | 07 | | | 满足第八个条件时ON | |
| | 08 | | 高速计数器1比较标志 | 表示比较操作的状态。 0: 停止 1: 运行 | |
| 09 | 高速计数器1上溢/下溢标志 | 表示PV的上溢/下溢状态。 0: 正常(无上溢/下溢) 1: 出现上溢/下溢 | | | |
| AR 06 | 00 | 端口2 | 高速计数器2范围比较标志 | 满足第一个条件时ON | 高速计数器用于范围比较格式时,当满足相应条件时标志置ON。 |
| | 01 | | | 满足第二个条件时ON | |
| | 02 | | | 满足第三个条件时ON | |
| | 03 | | | 满足第四个条件时ON | |
| | 04 | | | 满足第五个条件时ON | |
| | 05 | | | 满足第六个条件时ON | |
| | 06 | | | 满足第七个条件时ON | |
| | 07 | | | 满足第八个条件时ON | |
| | 08 | | 高速计数器2比较标志 | 表示比较操作的状态。 0: 停止 1: 运行 | |
| 09 | 高速计数器2上溢/下溢标志 | 表示PV的上溢/下溢状态。 0: 正常(无上溢/下溢) 1: 出现上溢/下溢 | | | |

SR 区标志

| 字 | 位 | 功能 |
|--------|----|--------|
| SR 254 | 15 | 内板错误标志 |

AR 区标志

| 字 | 位 | 功能 |
|-------|---------|---|
| AR 04 | 08 ~ 15 | 槽2中内板的错误代码 00 Hex: 正常 01,02 Hex: 硬件错误 03 Hex: PC设置错误 |

有关的 PC 设置设定

| 字 | 位 | 功能 | | 起作用的时间 |
|---------|---------|--|---|--------|
| DM 6611 | 00 ~ 15 | 端口模式设定(对端口 1 和 2) 0000 Hex: 高速计数器模式 0001 Hex: 简单定位模式 | | 电源上电时 |
| DM 6643 | 00 ~ 03 | 端口 1 | 高速计数器输入模式 0 Hex: 相位差输入 1 Hex: 脉冲/方向输入 2 Hex: 增/减脉冲输入 | 开始操作时 |
| | 04 ~ 07 | | 高速计数器复位方法 0 Hex: Z 相信号 + 软件复位 1 Hex: 软件复位 | |
| | 08 ~ 11 | | 高速计数器数值范围 0 Hex: 线性模式 1 Hex: 环形模式 | |
| | 12 ~ 15 | | (脉冲输出使用设定) | |
| DM 6644 | 00 ~ 03 | | 高速计数器输入模式 0 Hex: 相位差输入 1 Hex: 脉冲/方向输入 2 Hex: 增/减脉冲输入 | |
| | 04 ~ 07 | | 高速计数器复位方法 0 Hex: Z 相信号 + 软件复位 1 Hex: 软件复位 | |
| | 08 ~ 11 | | 高速计数器数值范围 0 Hex: 线性模式 1 Hex: 环形模式 | |
| | 12 ~ 15 | | (脉冲输出设定) | |

脉冲输出规格

指令

用下表所示的 7 个指令控制脉冲输出。此表也显示了指令和脉冲输出类型之间的关系。

| 指令 | 控制提要 | 没有加速/ 减速的 单相脉冲输出 | 具有相同加速度/ 减速度的单相脉 冲输出 | 具有独立加速度/ 减速度的单相脉 冲输出 | 可变占 空率脉 冲输出 |
|---------------------------|----------------------------|---------------------|----------------------------|----------------------------|-------------------|
| PULS(65) (设置脉冲) | 设定输出脉冲数 | 是 (只有独立模式) | ... | 是 (只有独立模式) | ... |
| SPED(64) (速度输出) | 控制没有加速/ 减速的脉冲输出 | 是 | ... | ... | ... |
| PLS2(—) (脉冲输出) | 控制具有相同加速度/ 减速度的梯形加速/减速脉冲输出 | ... | 是 | ... | ... |
| ACC(—) (加速度控制) | 控制具有独立加速度/ 减速度的梯形加速/减速脉冲输出 | ... | ... | 是 | ... |
| PWM(—) (可变占空率 的脉冲) | 控制可变占空率脉冲输出 | ... | ... | ... | 是 |
| INI(61) (模式控制) | 停止脉冲输出 | 是 | 是 | 是 | 是 |
| PRV(62) (高速计数器 PV读) | 读脉冲输出状态 | 是 | 是 | 是 | 是 |

输出时适用的指令

一旦开始输出,与脉冲输出有关的某些指令不能改变。下表列出了可以和不能在另一个指令执行后(即当脉冲输出作为前一个指令的结果执行)执行以修改脉冲输出的指令。

| 启动脉冲输出的指令 | 用于修改脉冲输出的指令 | | | | | | | | | | |
|--------------------------|-------------|-----------|------------------|-----------------------|-------------------|------|-------------------|-------------------|-------------------|-------------------|-----|
| | SPED (单独) | SPED (连续) | PULS (0或1: 脉冲设定) | PULS (2或3: 脉冲加速/减速设定) | PULS (4或5: 无脉冲设定) | PLS2 | ACC 模式0 (加速 + 单独) | ACC 模式1 (加速 + 连续) | ACC 模式2 (减速 + 单独) | ACC 模式3 (减速 + 连续) | PWM |
| SPED(64) (单独模式) | 允许 | ... | ... | ... | ... | ... | 允许 | ... | 允许 | ... | ... |
| SPED(64) (连续模式) | 允许 | 允许 | 允许 | 允许 | ... | ... | ... | 允许 | ... | 允许 | ... |
| PULS(65) 0,1(脉冲设定) | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | ... | 允许 | 允许 | 允许 | ... |
| PULS(65) 2,3 (脉冲加速/减速设定) | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | 允许 | ... |
| PULS(65) 3,4 (无脉冲设定) | ... | 允许 | 允许 | 允许 | 允许 | 允许 | ... | 允许 | ... | 允许 | ... |
| PLS2(—) | ... | ... | ... | ... | ... | ... | ... | ... | 停止时允许 | ... | ... |
| ACC(—) 模式0 (加速 + 单独) | ... | ... | ... | ... | ... | ... | ... | ... | 允许 | ... | ... |
| ACC(—) 模式1 (加速 + 连续) | ... | 允许常速 | 允许 (见注) | 允许 (见注) | ... | ... | ... | 允许常速 | ... | 允许 | ... |
| ACC(—) 模式2 (减速 + 单独) | 允许常速 | ... | ... | ... | ... | ... | ... | ... | 允许 | ... | ... |
| ACC(—) 模式0 (减速 + 连续) | ... | 允许常速 | 允许 (见注) | 允许 (见注) | 允许 (见注) | ... | ... | 允许常速 | ... | 允许 | ... |
| PWM(—) | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | 允许 |

注 脉冲数可修改,但是方向不可修改。

有关标志和控制位
(对于脉冲输出)

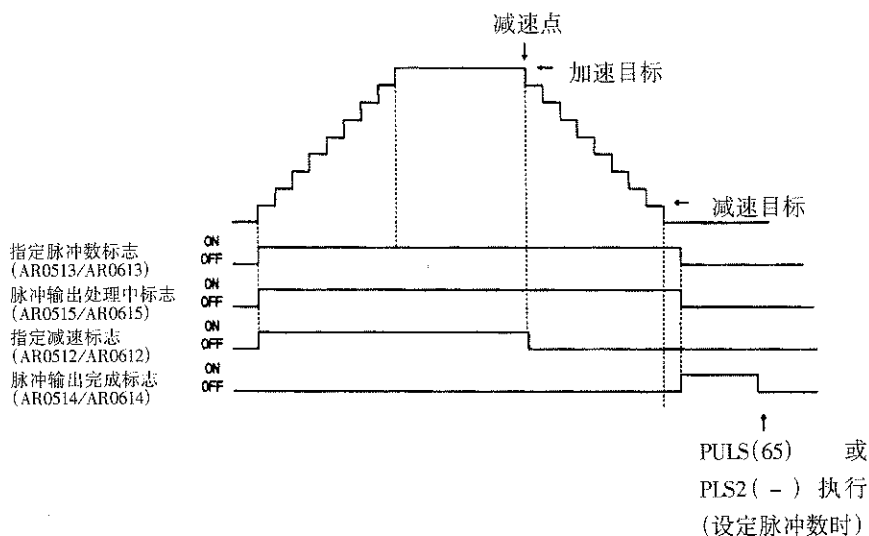
使用脉冲 I/O 板时的内板槽 2 的位

| 字 | 位 | 名称 | | 功能 |
|--------|---------|------|------------|--|
| IR 236 | 00 ~ 15 | 端口 1 | PV字(最右4位数) | 每个循环后,与脉冲I/O板的每个端口有关的脉冲输出的PV以8位数BCD存贮。在脉冲输出不使用时,这些位能作为内部辅助位使用。 |
| IR 237 | 00 ~ 15 | | PV字(最左4位数) | |
| IR 238 | 00 ~ 15 | 端口 2 | PV字(最右4位数) | |
| IR 239 | 00 ~ 15 | | PV字(最左4位数) | |

AR 区标志

| 字 | 位 | 名称 | 功能 |
|-------|----|-------------|---|
| AR 05 | 12 | 端口 1 脉冲输出标志 | 指定减速标志 表示指定减速时通过减速点 0: 未指定 1: 指定 |
| | 13 | | 指定脉冲数标志 表示脉冲数是否已用 PULS(65)进行了设置 0: 未指定 1: 指定 |
| | 14 | | 脉冲输出完成标志 表示用 SPED(64), PLS2(-)或 ACC(-)完成脉冲输出 0: 未完成 1: 完成 |
| | 15 | | 脉冲输出处理中标志 表示脉冲输出的执行状态 0: 无脉冲输出 1: 脉冲输出处理中 |
| AR 06 | 12 | 端口 2 脉冲输出标志 | 指定减速标志 表示指定减速时通过减速点 0: 未指定 1: 指定 |
| | 13 | | 指定脉冲数标志 表示脉冲数是否已用 PULS(65)进行了设置 0: 未指定 1: 指定 |
| | 14 | | 脉冲输出完成标志 表示用 SPED(64), PLS2(-),或 ACC(-)完成脉冲输出 0: 未完成 1: 完成 |
| | 15 | | 脉冲输出处理中标志 表示脉冲输出的执行状态 0: 无脉冲输出 1: 脉冲输出处理中 |

操作时序例



注 如上所示的 AR 区标志的状态可能根据输出频率与实际脉冲输出状态不同。

有关 PC 设置设定

| 字 | 位 | 功能 | 设定起作用的时间 |
|---------|---------|--|----------|
| DM 6611 | 00 ~ 15 | 端口模式设定(端口1和2) 0000 Hex: 高速计数器模式 0001 Hex: 简单定位模式 | 上电时 |
| DM 6643 | 00 ~ 11 | 端口1 (脉冲输入设定) | 操作开始时 |
| | 12 ~ 15 | 固定/可变脉冲输出占空率设定 0 Hex: 使用固定占空率脉冲输出(缺省) 1 Hex: 使用可变占空率脉冲输出 | |
| DM 6644 | 00 ~ 11 | 端口2 (脉冲输入设定) | |
| | 12 ~ 15 | 固定/可变脉冲输出占空率设定 0 Hex: 使用固定占空率脉冲输出(缺省) 1 Hex: 使用可变占空率脉冲输出 | |

2-2-7 高速计数器 1 和 2

从旋转编码器到脉冲 I/O 板的端口 1 和 2 的脉冲信号可高速计数, 并可根据计到的脉冲数执行中断处理。可单独使用两个端口, 用于端口 1 和 2 的计数器是高速计数器 1 和高速计数器 2。

本节介绍了怎样使用高速计数器 1 和 2。

注 可使用的指令被在 PC 设置的 DM 6611 中设定的板的端口模式设定所限制。

端口模式设定和适用的指令

在简单定位模式中, 不能使用 CTBL(63) (注册比较表), 不能执行高速计数器中断。只能执行读 PV。

| 指令 | CTBL(63) | INI(61) | PRV(62) |
|---------|---------------|-----------------|-------------------------|
| 功能 | 比较表注册 比较启动 | PV修改 比较启动/停止 | 读PV 读比较状态 读范围比较结果 |
| 高速计数器模式 | 允许 | 允许 | 允许 |
| 简单定位模式 | 禁止 | 允许(只有PV修改) | 允许 |

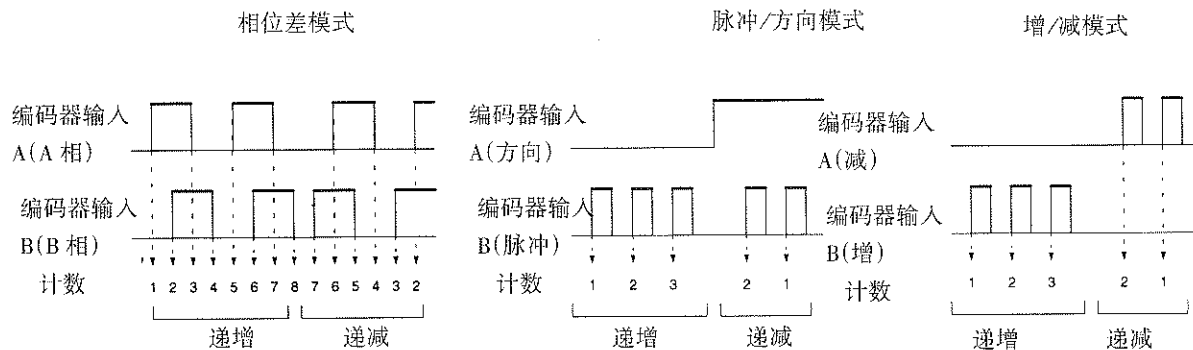
处理

输入信号和输入模式

信号类型决定了可用于高速计数器 1 和 2 的输入模式。

- 1, 2, 3...
1. 相位差模式(计数速率 = 25KHz):
两个相位差 4x 的信号(A 相和 B 相)和 Z 相信号用于输入。根据两相信号差递增或递减计数。
 2. 脉冲/方向模式(计数速率 = 50KHz):
A 相是方向信号, B 相是计数脉冲, A 相信号为 OFF 时计数器递增, 为 ON 时递减。
 3. 增/减模式(计数速率 = 50KHz):
A 相是递减信号, B 相是递增信号。

测到 A 相信号时计数器递减,测到 B 相信号时递增。



数值范围

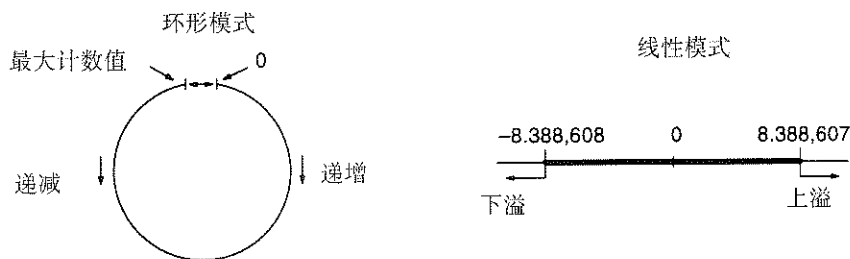
下列两种模式决定了高速计数器 1 和 2 计数的范围。

1,2,3... 1. 环形模式

在环形模式中,可和 CTBL(63)设定计数范围的最大值。递增时,计数器从最大计数值计到 0,递减时,计数器从 0 计到最大计数值;没有负值。最大计数值 + 1(即环值)作为设定输入。设定范围从 1 ~ 65000,计数范围从 0 ~ 64999。

2. 线性模式

线性模式中的计数范围固定为 - 8388608 ~ 8388607。如果计数降到下限以下,将产生下溢,如果超过上限,将产生上溢。时于上溢, PV 保持为 08388607,对于下溢, PV 保持为 F8388608,停止计数或比较(并保留比较表),并且 AR 0509(端口 1)或 AR 0609(端口 2)将置 ON。



重启动计数操作时,下段中的一种方法必须用于计数器复位。启动或停止执行程序时,计数器自动复位。

注 下列信号变换作为向前(递增)脉冲处理: A 相上升沿→B 相上升沿→A 相下降沿→B 相下降沿。

下列信号变换作为反向(递减)脉冲处理: B 相上升沿→A 相上升沿→B 相下降沿→A 相下降沿。

复位方法

下列两种方法可用于决定复位计数器 PV(即设置 0)的时序:

- Z 相信号 + 软件复位
- 软件复位

Z 相信号 + 软件复位或单独软件复位都可用于复位计数器的 PV。这些复位操作与高速计数器 0(内置高速计数器)复位方法相同。

详细内容参见 30 页。高速计数器 1 和 2 的复位位如下：

高速计数器 1 复位位：SR25201

高速计数器 2 复位位：SR25202

- 注
1. 由于高速计数器 1 和 2 的复位位 (SR25201 和 SR25202) 在每个循环刷新，标志必须至少在一个循环内置 ON 以可靠地读。
 2. 即使在复位后，比较表注册状态，比较执行状态和范围比较结果仍保持不变。(如果在复位前正在执行比较，比较将继续。)

高速计数器中断的
计数检查方法

与高速计数器 0 一样，下列两种计数检查方法可用于高速计数器 1 和 2：

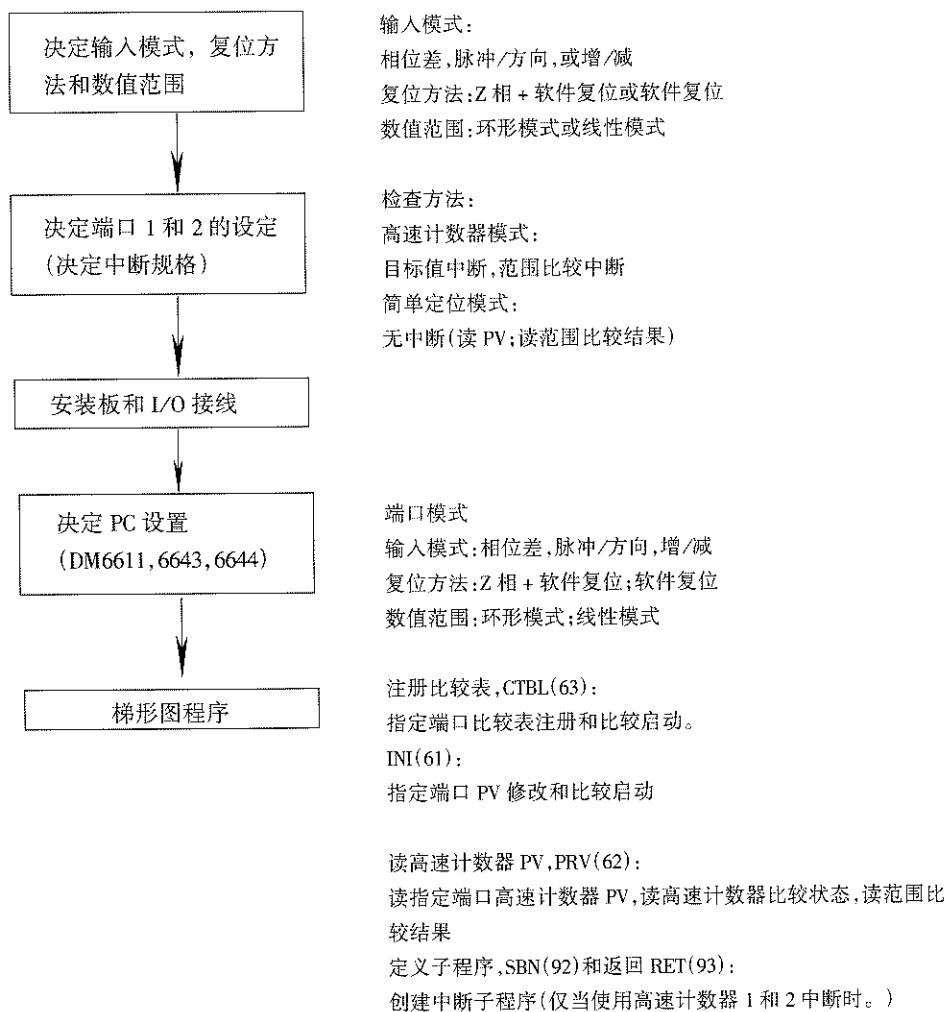
- 目标值方法
- 范围比较方法

每种方法的介绍参见 31 页。

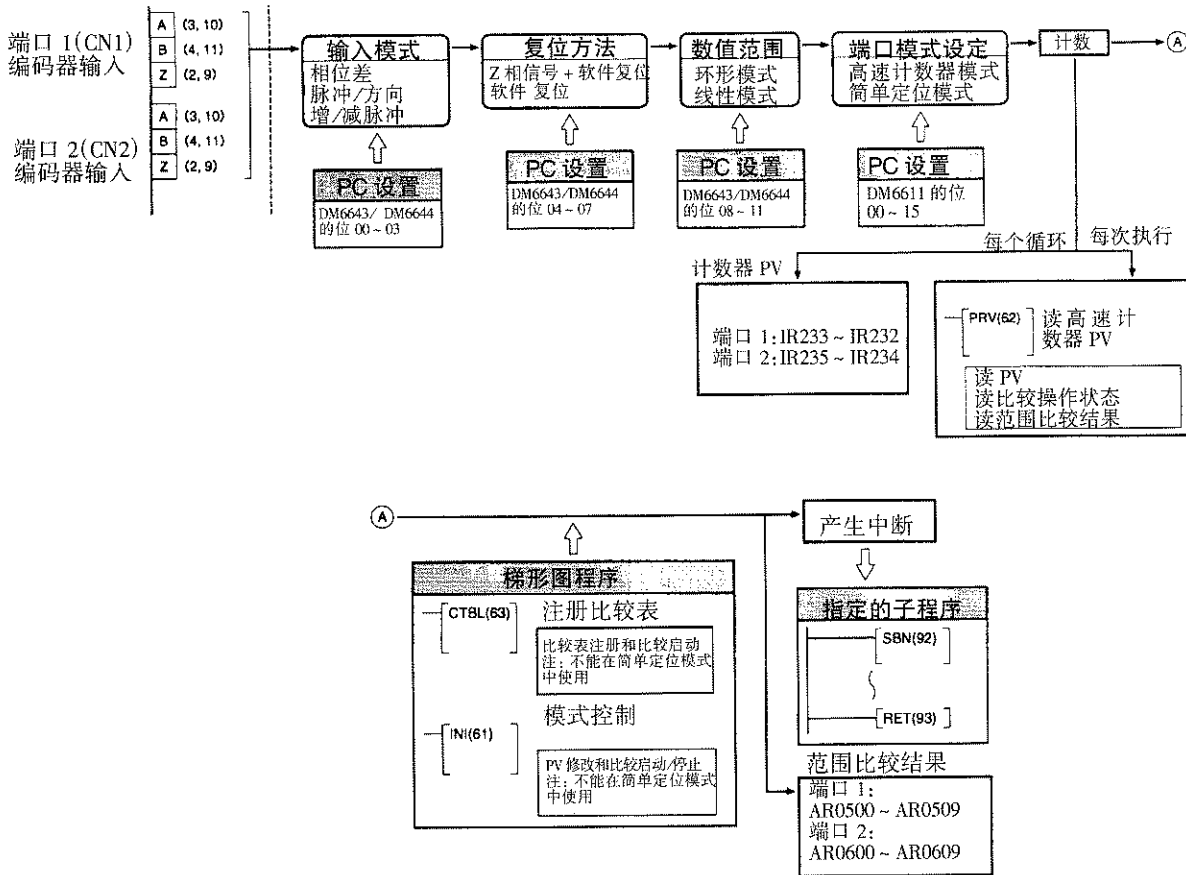
对于目标值方法，可在比较表中注册最多 48 个条件。当计数器的 PV 等于 48 个注册比较值中的一个时，执行相应的中断子程序。

对于范围比较方法，通常在比较表中注册 8 个比较条件。当计数器的 PV 处于范围 1~8 之一的上限与下限之间时，执行相应的中断子程序。

使用步骤



脉冲 I/O 板:高速计数器功能

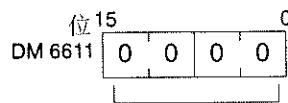


预置 PC 设置

使用高速计数器 1 和/或 2 前,在编程模式中输入下列设定。

端口模式设定 (DM 6611)

为端口 1 和 2 指定高速计数器模式。PC 变 ON 时读此设定。如果修改设定,必须重新启动 PC。



端口模式设定

0000Hex: 高速计数器模式

(使用高速计数器中断时必须设为高速计数器模式。)

0001Hex: 简单定位模式

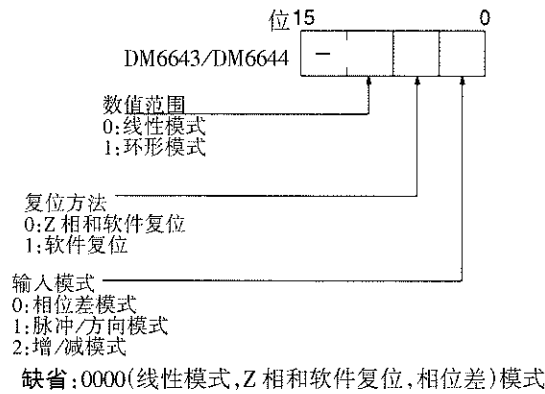
缺省: 0000(高速计数器模式)

- 注
1. 使用高速计数器 1 和 2 中断时,端口必须设为高速计数器模式。虽然高速计数器的 PV 可用简单定位模式读,但不能使用高速计数器 1 和 2 中断。
 2. 此设定仅在 CQM1H 启动时被识别。要修改设置,必须在执行程序前先关电再上电。

3. 如果用 DM 6611 把端口 1 和 2 设为简单定位模式,可以使用 BCMP(68)指令检查高速计数器 1 和 2 的 PV 字的内容 (IR 232~IR235),并且用此信息产生高速计数器 1 和 2 中断。然而,用此方法得到的 PV 可能与实际 PV 稍有不同。

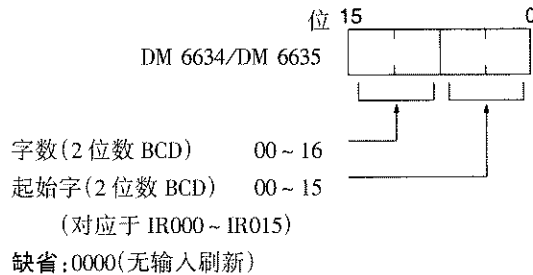
端口 1 和端口 2 操作设定

DM 6643 包含端口 1 的设定,DM 6644 包含端口 2 的设定。这些设定决定了这些高速计数器的操作参数。使用与每个端口的工作环境相符的设定。



输入刷新字设定

DM 6634 和 DM 6635 分别包含高速计数器 1 和 2 的输入刷新字设定。必须在执行中断前刷新输入时进行这些设定。



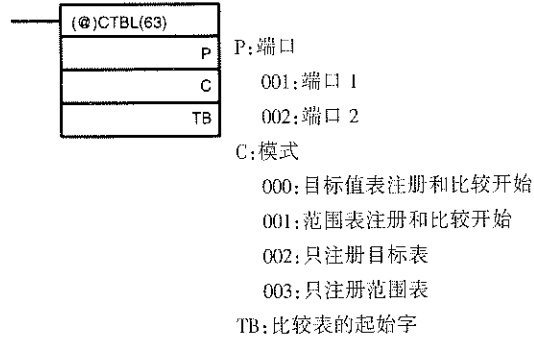
编程

使用下列步骤编写高速计数器 1 和 2 的程序。

- 注**
1. 进行合适的 PC 设置设定后,高速计数器 1 和 2 开始计数。
 2. 上电时,开始工作时,停止工作时,高速计数器 1 和 2 的 PV 复位为 0。
 3. 只用计数操作将不执行比较表比较和中断。
 4. 高速计数器 1 的 PV 存贮在 IR 232 和 IR 233 中,高速计数器 2 的 PV 存贮在 IR 234 和 IR 235 中。

启动和停止比较

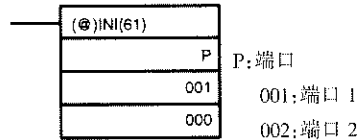
1,2,3... 1. 使用 CTBL(63)在 CQM1H 中保存比较表并开始比较。



如果 C 设为 000, 则用目标值方法进行比较; 如果为 001, 则用范围比较方法进行比较。两种情况下, 都在注册比较表后开始比较。执行比较时, 根据比较表执行高速计数器 1 和 2 中断。保存的比较表的详细内容参见第 5 章 指令集中的 CTBL(63)的介绍。

注 把 C 值设为 002 注册目标值比较表, 设为 003 注册范围比较表, 但不会自动启动比较。在这些情况下, INI(61)必须用于启动比较操作。

2. 执行如下所示的 INI(61)以停止比较。在 P 中指定端口 1 或 2(P = 001 或 002)。



- 注
1. 要启动比较, 必须把第 1 个操作数设为端口号, 第 2 个操作数设为“000”(执行比较), 并执行 INI(61)指令。
 2. 已注册的表在操作过程中(即程序执行时)保存在 CQM1H 中, 直到注册了新表。

读高速计数器 1 和 2 的 PV

下列两种方法可用于读高速计数器 1 和 2 的 PV:

- 从内存读 PV
- 使用 PRV(62)

从内存读 PV

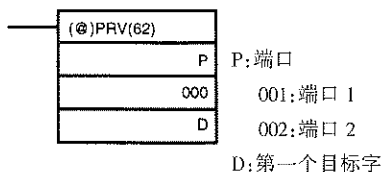
高速计数器 1 和 2 的 PV 以下列方法存贮在相应的数据区字中。

| | 最左 4 位数 | 最右 4 位数 | 线性模式 | 环形模式 |
|-------|---------|---------|--------------------------|---------------------|
| 端口 1: | IR233 | IR232 | F8388608 ~ 08388607 | 00000000 ~ 00064999 |
| 端口 2: | IR235 | IR234 | (-8,388,608 ~ 8,388,607) | |
| | | | (为负数时最左位数为 F) | |

注 这些字每个循环只刷新一次, 因此它们可能与实际 PV 不同。

使用 PRV(62)

PRV(62)用于读高速计数器 1 和 2 的 PV。在 P 中(P = 001 或 002)指定高速计数器 1 或 2。



每个高速计数器的 PV 如下所示存贮。在线性模式中,对于负数,最左位数为 F。

| 最左 4 位数 | 最右 4 位数 | 线性模式 | 环形模式 |
|---------|---------|--|---------------------|
| D + 1 | D | F8388608 ~ 08388607 (- 8, 388, 608 ~ 8, 388, 607) | 00000000 ~ 00064999 |

注 执行 PRV(62)时可准确地读 PV。

修改 PV

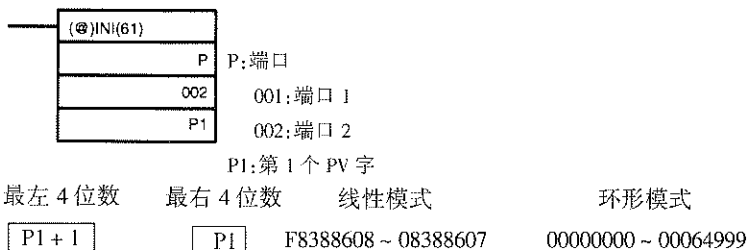
有两种方法修改高速计数器 1 和 2 的 PV。

- 用复位方法之一复位为 0
- 使用 INI(61)

使用 INI(61)的方法在此介绍。复位方法的用法介绍参见“复位方法”。(63 页)

用 INI(61)修改 PV

用 INI(61)修改高速计数器 1 和 2 的 PV,如下所示。



为在线性模式中指定负数,在最左位数中设定 F Hex。

读高速计数器 1 和 2 的状态

有两种方法读高速计数器 1 和 2 的状态。

- 读 CQM1H 的 AR 区中的有关标志
- 使用 PRV(62)

读有关的 AR 区标志

下面显示了与高速计数器 1 和 2 有关的 CQM1H 数据字。可以通过读这些字知道高速计数器 1 和 2 的状态。

- 内板错误代码

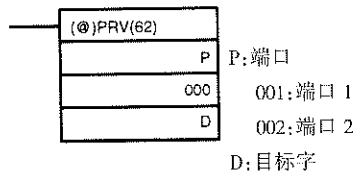
| 字 | 位 | 功能 | |
|-------|---------|-----|---|
| AR 04 | 08 ~ 15 | 槽 2 | 所存的错误代码如下: 00 Hex: 正常 01, 02 Hex: 硬件错误 03 Hex: PC 设置错误 |

· 工作状态

| 字 | | 位 | 名称 | 功能 | |
|-------|--------------|---|-------------|-----------------------------|---------------------------------|
| 计数器1 | 计数器2 | | | | |
| AR 05 | AR 06 | 00 | 高速计数器范围比较标志 | 满足第一个条件时ON | 当高速计数器使用范围比较格式时, 在满足相应条件时该位置ON。 |
| | | 01 | | 满足第二个条件时ON | |
| | | 02 | | 满足第三个条件时ON | |
| | | 03 | | 满足第四个条件时ON | |
| | | 04 | | 满足第五个条件时ON | |
| | | 05 | | 满足第六个条件时ON | |
| | | 06 | | 满足第七个条件时ON | |
| | | 07 | | 满足第八个条件时ON | |
| | | 08 | 高速计数器比较标志 | 表示比较操作的状态 0: 停止 1: 运行 | |
| 09 | 高速计数器上溢/下溢标志 | 表示PV的上溢/下溢状态 0: 正常(无上溢/下溢) 1: 出现上溢/下溢 | | | |

使用 PRV(62)

也可由 PRV(62)的执行决定高速计数器 1 和 2 的状态。指定高速计数器 1 或 2(P = 001 或 002)和目标字 D。状态信息将写到 D 的位 00 和 01 中。位 02 ~ 15 将设为 0。



指定高速计数器的状态存贮在 P1 的位 00 和 01 中,如下表所示。

| 位 | 功能 |
|----|------------------------------------|
| 00 | 比较操作标志(0:停止;1:运行) |
| 01 | 高速计数器1和2的PV上溢/下溢标志(0:正常;1:出现上溢或下溢) |

例

位 04 ~ 07 表示脉冲输出状态;所有其它位为 0。

此例显示了从端口 1 输出标准脉冲, 而用高速计数器 1 计数这些脉冲的程序。高速计数器在增/减模式下工作,用脉冲输出的 CW 脉冲递增计数(B 相输入), CCW 脉冲递减计数(A 相输入)。执行程序前,如下设定 PC 设置并且重新启动 PC,允许 DM 6611 设定。

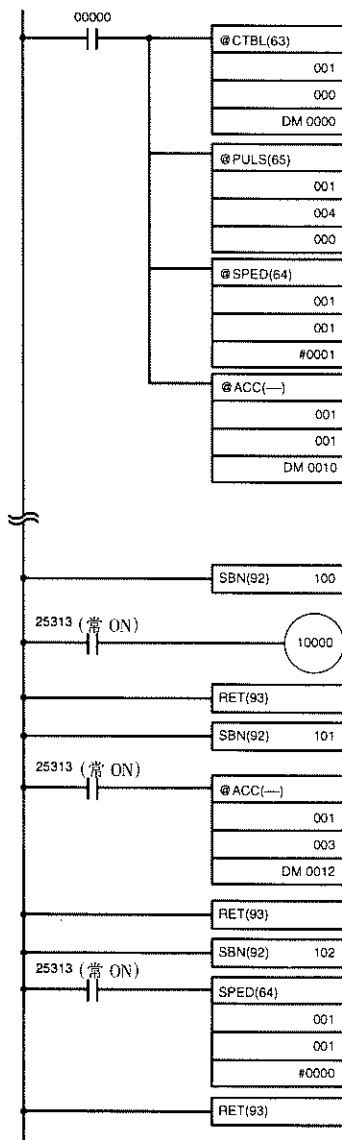
DM 6611: 0000(高速计数器模式)。

DM 6643: 0002(端口 1: 固定占空率脉冲输出, 线性模式, Z 相信号及软件复位和增/减模式)。

其它 PC 设置设定使用缺省设定。(处理中断前不刷新输入)。

此外,比较表中存贮了下列数据:

- DM 0000: 0003—目标值个数:3
- DM 0001: 2500—目标值 1:2500
- DM 0002: 0000
- DM 0003: 0100—比较 1 中断处理子程序号:100
- DM 0004: 7500—目标值 2:7500
- DM 0005: 0000
- DM 0006: 0101—比较 2 中断处理子程序号:101
- DM 0007: 0000—目标值 3:10000
- DM 0008: 0001
- DM 0009: 0102—比较 3 中断处理子程序号:102



指定端口 1,以目标值格式保存比较表,并开始比较。

为端口 1 设定 CW 脉冲。(未设定脉冲个数)

从端口 1 以 10Hz 为频率单位开始连续的脉冲输出。

ACC(-)模式 1 以大约 500Hz/4ms 把频率加速到 25KHz。
 DM 0010:0050 500Hz 加速 ./4ms
 DM 0011:2500 目标值 25KHz

IR 10000 置 ON

ACC(-)模式 3 以大约 500Hz/4ms 把频率减速到 500Hz
 DM 0012: 0050 500Hz 加速 ./4ms
 DM 0013: 0050 目标值:500Hz

用频率设为 0 来停止从端口 1 的脉冲输出。

2-2-8 功能

下表给出脉冲 I/O 板的脉冲输出功能。

| 分类 | 特征 | 使用的指令 |
|-----------------------|---|--|
| 端口 1 和 2 脉冲输出 (固定占空率) | 10Hz ~ 50(20)KHz 频率 固定占空率 双向输出 (CW 和 CCW) 频率可平滑地修改 | 设定脉冲数: PULS(65) 开始脉冲输出: SPED(64) 修改频率: SPED(64) 停止脉冲输出: SPED(64)/INI(61) 以相同速率加速/减速: PLS2(—) 以单独速率加速/减速: ACC(—) |
| 端口 1 和 2 脉冲输出 (可变占空率) | 91.6Hz, 1.5KHz, 或 5.9KHz 频率, 占空率在 1% 到 99% 可变, 仅单向输出。 | 开始脉冲输出: PWM(—) 停止脉冲输出: INI(61) |

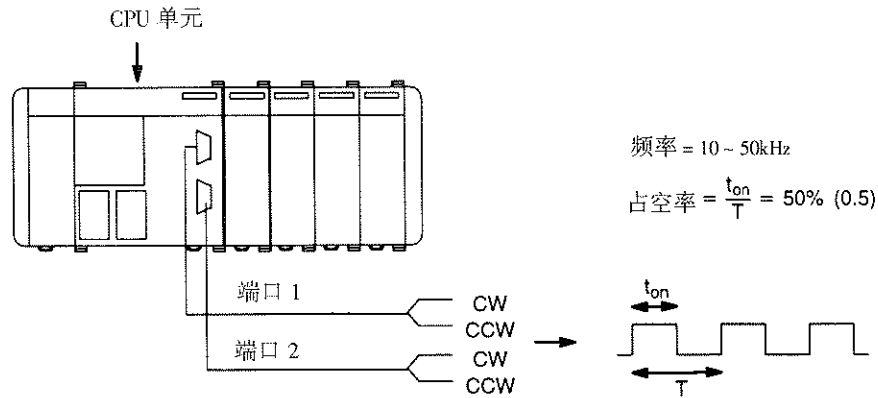
注 当步进马达连接到端口 1 或 2 的脉冲输出时, 使用不超过 20KHz 的最高频率。

2-2-9 固定占空率脉冲输出

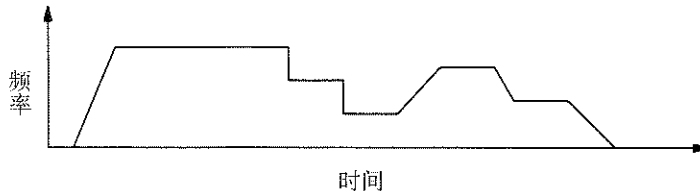
下面介绍用 50% 的占空率从端口 1 和 2 执行脉冲输出的步骤。

提要

如下图所示从端口 1 和 2 执行脉冲输出。端口 1 和 2 能同时使用, 每个端口的脉冲输出能切换为 CW(顺时针)或 CCW(逆时针)方向。



从端口 1 和 2 输出脉冲时, 可逐步或以指定速率修改频率, 如下图所示。



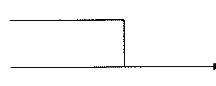
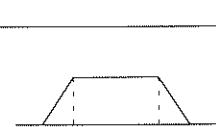
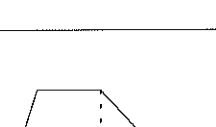

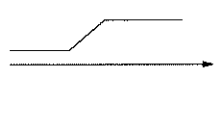
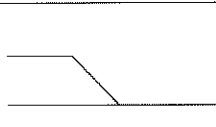


可以下列两种模式从端口 1 和 2 执行脉冲输出:

- 连续模式: 连续输出脉冲直到用 SPED(64) 指令或 INI(61) 指令停止。
- 独立模式: 输出了指定脉冲数时, 脉冲输出自动停止。也可用 SPED(64) 或 INI(61) 指令停止输出。

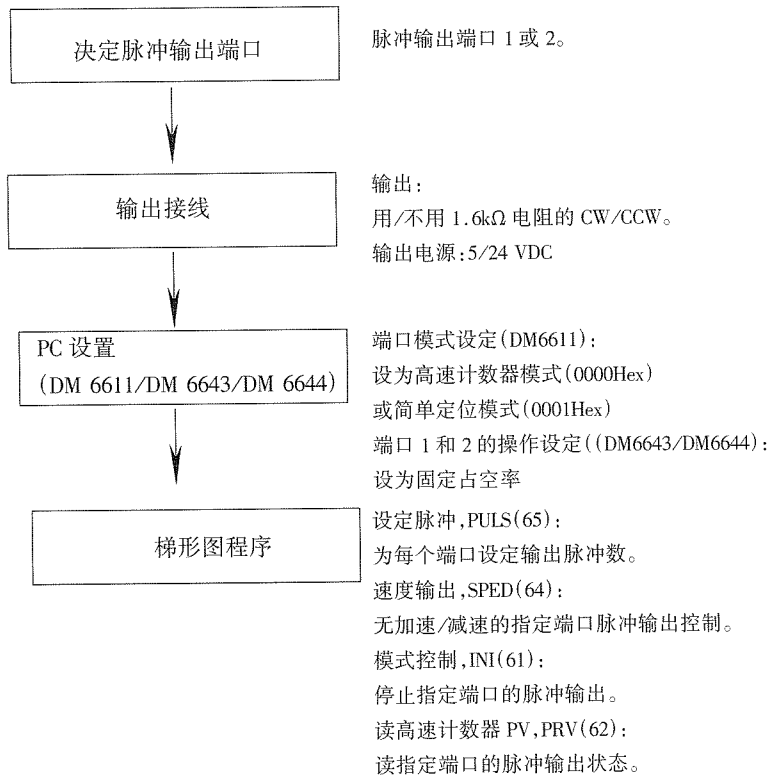
注 必须立即停止脉冲输出 (例如紧急停止) 时使用 INI(61)。即使 SPED(64), PLS2(—) 或 ACC(—) 信号把输入置 OFF, 不停止脉冲输出。

下表列出了可用 PULS(65), SPED(64), INI(61), PLS2(—), 和 ACC(—)的组合修改的频率类型。

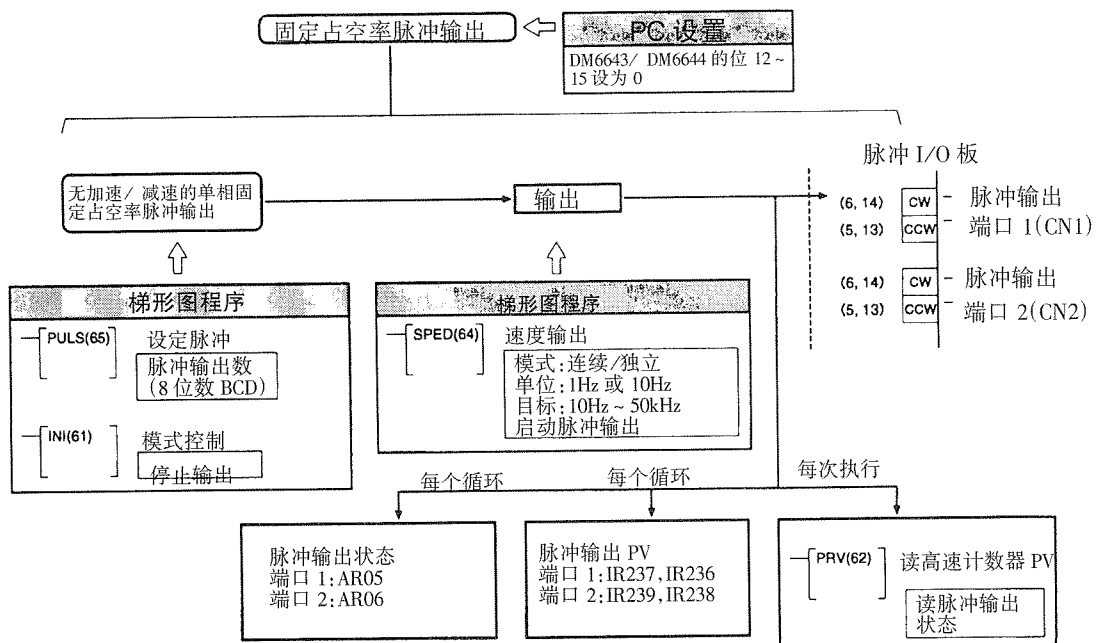
| 修改频率 | 指令 | 操作数设定 | 页 |
|--|-----------------|--|-----|
|  <p>以指定频率开始脉冲输出。 执行PULS(65),然后执行SPED(64)。</p> | PULS(65) | CW/CCW (脉冲数) | 103 |
| | SPED(64) | 端口 (连续/独立频率) | |
|  <p>脉冲输出时逐步修改频率。</p> | SPED(64) | 端口 (连续/独立频率) | |
|  <p>用指令停止脉冲输出。 执行SPED(64)或INI(61)。</p> | SPED(64) | 端口 频率 = 0 | 105 |
| | INI(61) | 设定控制数据停止 脉冲输出 | |
|  <p>输出指定数量脉冲,脉冲输出以指定速率加速到目标频率,以相同速率减速到停止。</p> | PLS2(—) | 端口 CW/CCW 加速度/减速度 目标频率 脉冲数 | 106 |
|  <p>输出指定数量脉冲。脉冲输出以指定加速度加速到目标频率,以另一个指定减速度减速到停止。 ACC(—)指令模式0:加速 + 独立模式。 执行PULS(65),接着执行ACC(—)。</p> | PULS(65) | CW/CCW 脉冲数 减速点 | 107 |
| | ACC(—) (模式0) | 端口 加速度 目标频率 1 减速度 目标频率 2 | |
|  <p>以指定加速度把脉冲输出从当前频率加速到目标频率。 继续脉冲输出。 执行PULS(65),接着执行ACC(—)。 ACC(—)指令模式1:加速 + 连续模式</p> | PULS(65) | CW/CCW | 107 |
| | ACC(—) (模式1) | 端口 加速度 目标频率 | |
|  <p>以指定减速度把脉冲输出从当前频率减速到目标频率。 输出指定数量脉冲时停止脉冲输出。 执行PULS(65),接着执行ACC(—)。 ACC(—)指令模式2:减速 + 独立模式。</p> | PULS(65) | CW/CCW 脉冲数 | 108 |
| | ACC(—) (模式2) | 端口 减速度 目标频率 | |
|  <p>以指定减速度把脉冲输出从当前频率减速到目标频率。 继续脉冲输出。 执行PULS(65),接着执行ACC(—)。 ACC(—)指令模式3:减速 + 连续模式</p> | PULS(65) | CW/CCW | 108 |
| | ACC(—) (模式3) | 端口 减速度 目标频率 | |

单相固定占空率脉冲输出

下面流程显示了使用 PULS(65) 和 SPED(64) 执行无加速或减速的单相固定占空率脉冲输出的步骤。

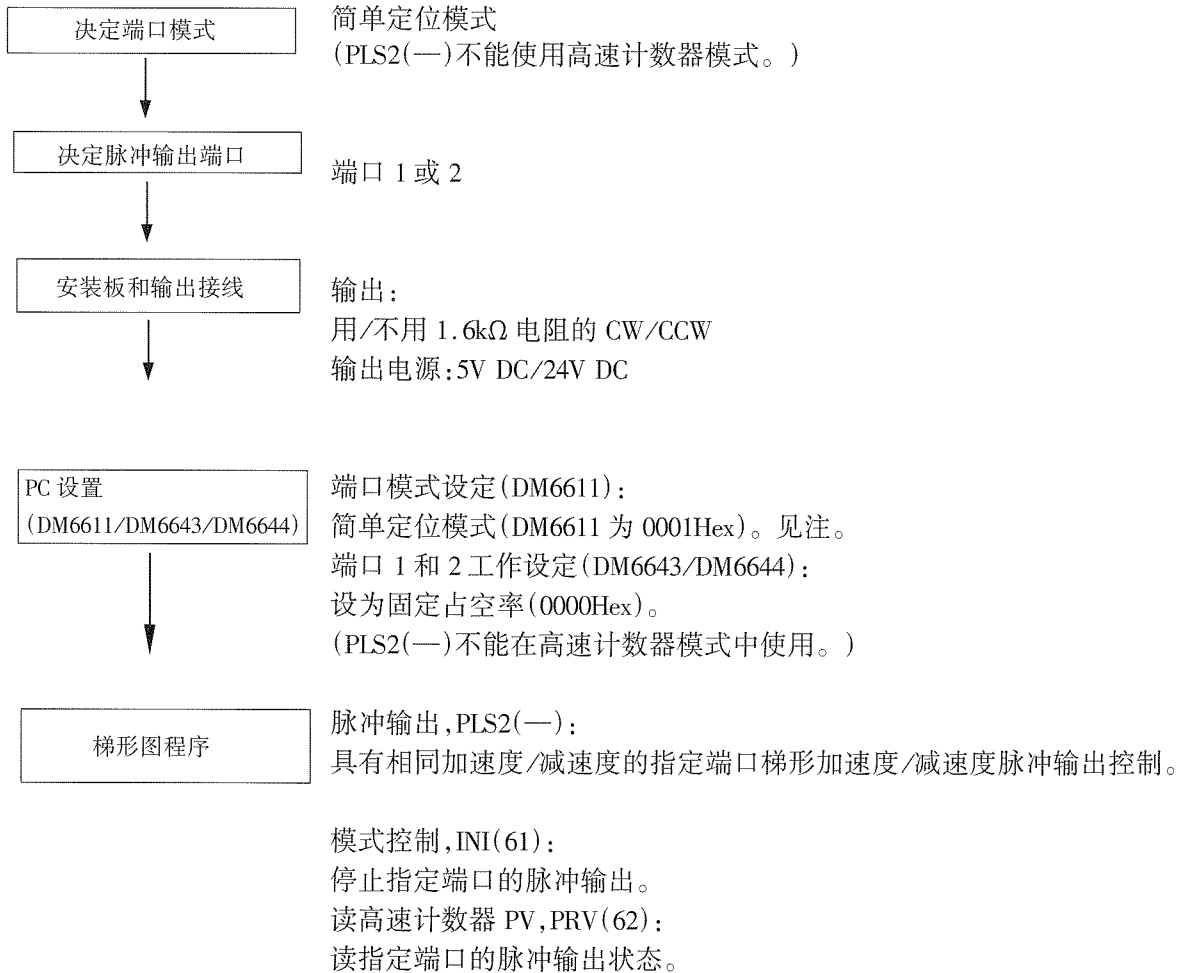


无加速/减速的单相固定占空率脉冲输出

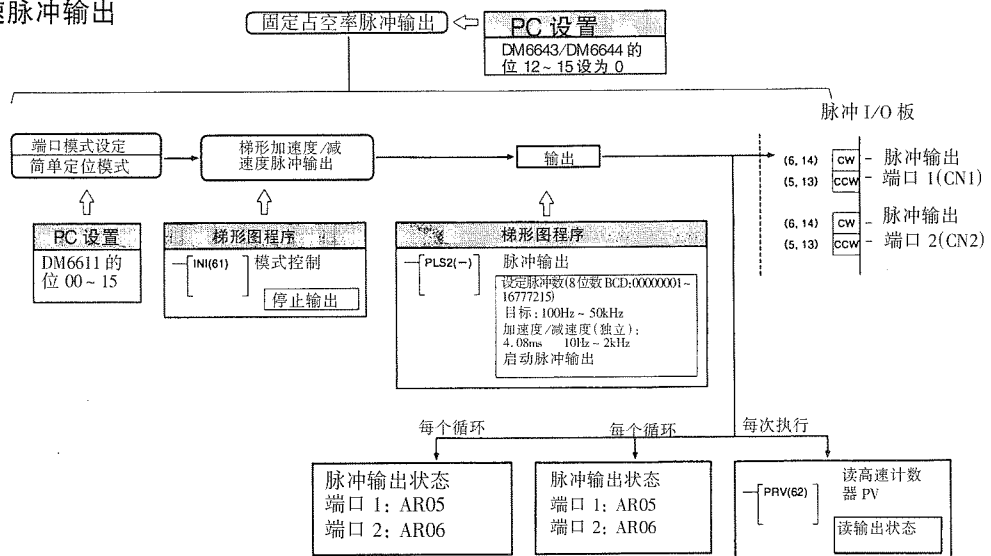


具有相同加速度/减速度的梯形脉冲输出

下面流程显示了使用 PLS2(一) 执行具有相同加速度/减速度的梯形脉冲输出的步骤。

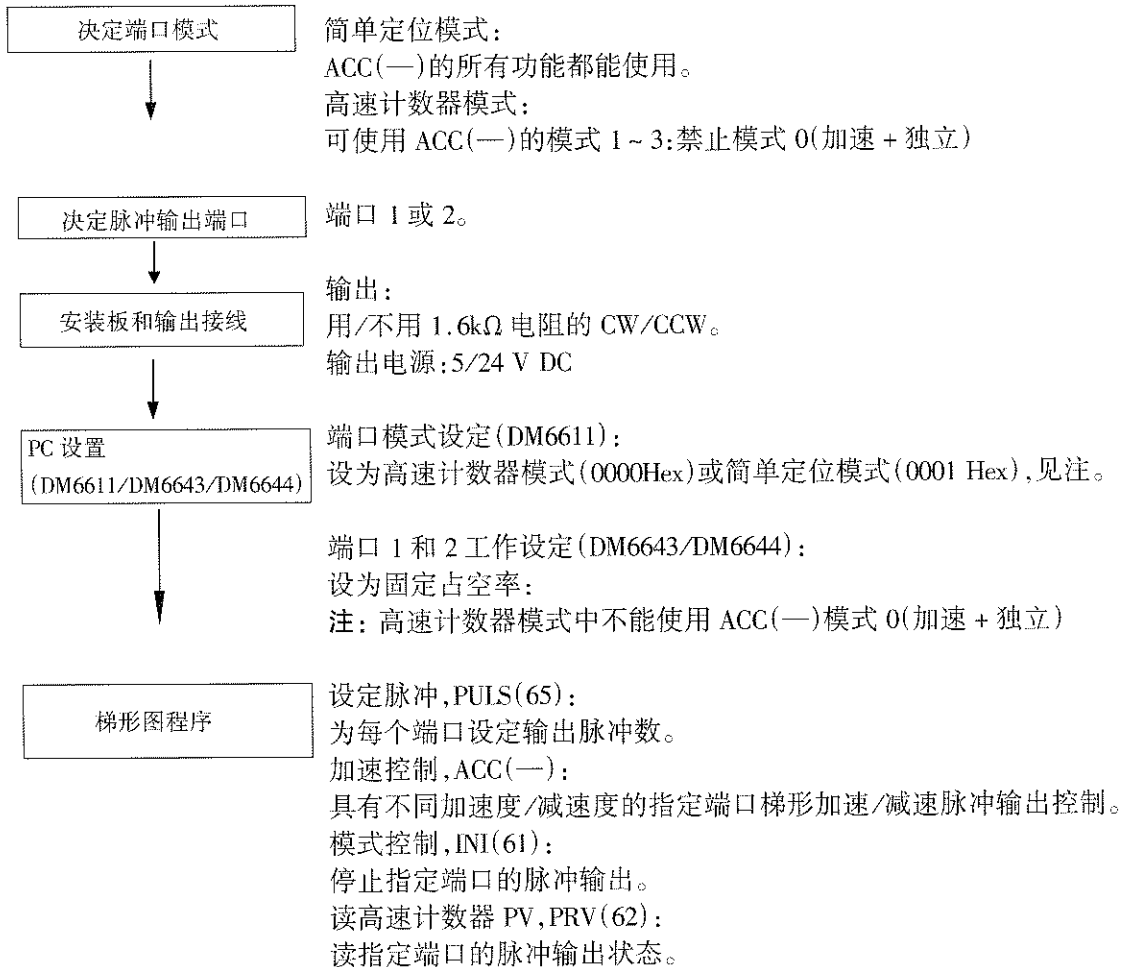


梯形加速/减速脉冲输出

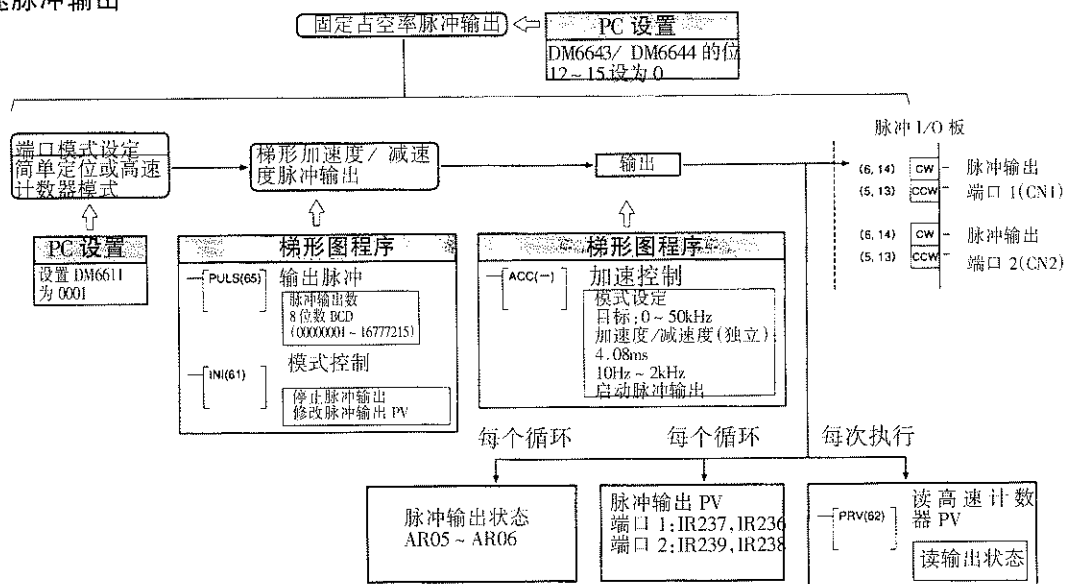


具有不同加速度/减速度的梯形脉冲输出

下面流程显示了使用 PULS(65) 和 ACC(-) 执行具有不同加速度/减速度的梯形脉冲输出的步骤。



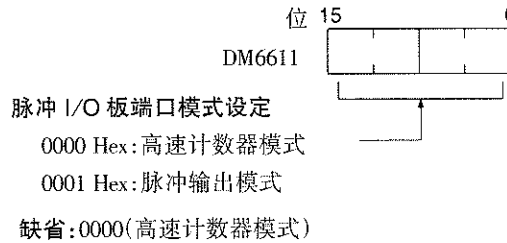
梯形加速/减速脉冲输出



PC 设置设定

从端口 1 或 2 输出脉冲前,把 PC 切换到编程模式,并在 PC 设置中输入下列设定。

端口模式设定(DM6611)



可使用的指令受到脉冲 I/O 板的端口 1 和 2 的端口模式设定的限制。端口模式在 PC 设置(DM6611)中指定。

端口模式设定和指令

下表显示了端口模式设定和可与不同的脉冲输出一起使用的指令。

不带梯形加速/减速的脉冲输出

不管端口模式设定是什么,所有指令都可使用。

| 指 令 | PULS(65) | SPED(64) | INI(61) | PRV(62) |
|---------|-----------------|----------|---------|---------|
| 功能 | 设定脉冲数 (组合使用) | 设定频率 | 停止脉冲输出 | 读脉冲输出状态 |
| 高速计数器模式 | 允许 | | | |
| 简单定位模式 | 允许 | | | |

具有梯形加速/减速和相同加速度/减速度的脉冲输出。

不能在高速计数器模式中使用 PLS2(一)(脉冲输出)。不能用相同的加速度/减速度执行梯形加速/减速脉冲输出。

| 指 令 | PLS2(一) | INI(61) | PRV(62) |
|---------|---------|---------|---------|
| 功能 | 设定脉冲数 | 停止脉冲输出 | 读脉冲输出状态 |
| 高速计数器模式 | 禁止 | 允许 | |
| 简单定位模式 | 允许 | | |

具有梯形加速/减速和独立加速度/减速度的脉冲输出

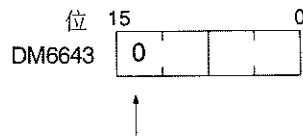
唯一的限制是在高速计数器模式中不能使用 ACC(一) (加速控制) 的模式 0 (加速 + 独立)。

| 指 令 | PULS(65) | ACC(一) | INI(61) | PRV(62) |
|-------------|----------|-------------------------------------|------------|-------------|
| 功能 | 设定脉冲数 | 加速度/减速度 (分别设定) 设定频率 开始脉冲输出 | 停止脉冲 输出 | 读脉冲输出 状态 |
| | (组合使用) | | | |
| 高速计数器 模式 | 允许 | 模式0(加速 + 独 立):禁止 模式3:允许 | 允许 | |
| 简单定位模 式 | 允许 | | | |

只在启动 CQM1H 时读 DM6611 中的设定。如果修改此设定, PC 必须先置 OFF,再置 ON,启用新值。

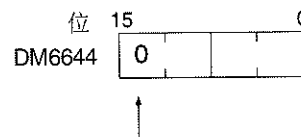
端口 1 和 2 工作设定(DM6643 和 DM6644)

下图显示了如何设定端口 1(DM6643) 和端口 2(DM6644) 以执行固定占空率脉冲输出(缺省脉冲输出格式)。端口 1 和 2 的设定可以不同。



端口 1 脉冲类型

- 0: 固定占空率脉冲输出执行标准脉冲输出时设为固定占空率。
- 1: 可变占空率脉冲输出
- 缺省: 0
(固定占空率脉冲输出)



端口 2 脉冲类型

- 0: 固定占空率脉冲输出执行标准脉冲输出时设为固定占空率。
- 1: 可变占空率脉冲输出
- 缺省: 0
(固定占空率脉冲输出)

如果设定为执行标准脉冲输出, 不能从端口输出可变占空率脉冲。

例

下例显示了控制端口 1 和 2 的脉冲输出的程序。运行程序前, 确认 PC 设置中的设定如下:

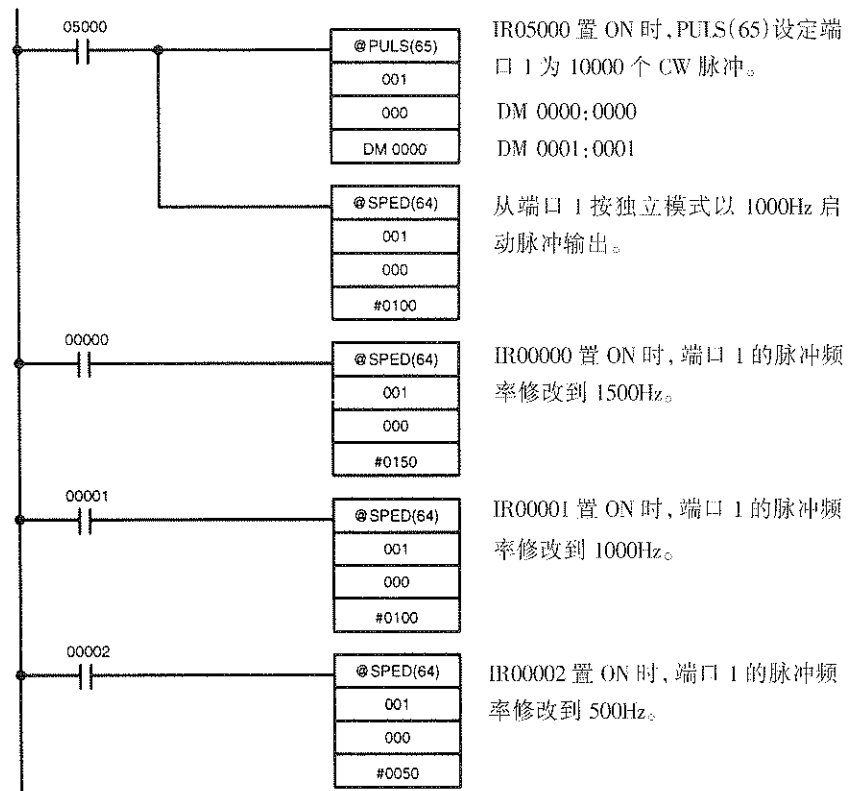
- DM6611: 0001(简单定位模式)
- DM6643: 0000(端口 1 的固定占空率脉冲输出)
- DM6644: 0000(端口 2 的固定占空率脉冲输出)

以指定频率启动脉冲输出

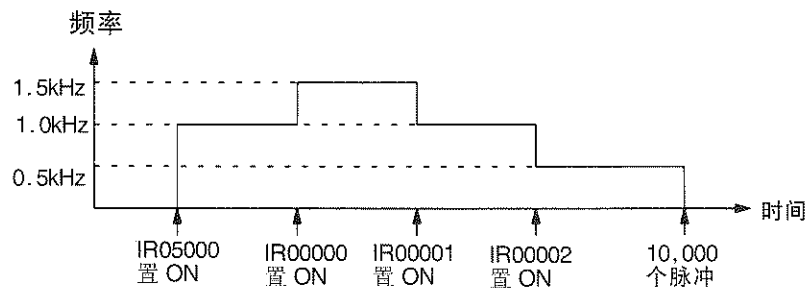
下例显示了用于控制端口 1 的脉冲输出的 PULS(65)和 SPED(64)。

例 1: 用 PULS(65)和 SPED(64) 启动脉冲输出

PULS(65) 中指定的脉冲数 (10000) 输出, 而用不同的频率设定执行 SPED (64), 改变频率。



下图显示程序执行时端口 1 的脉冲输出频率。

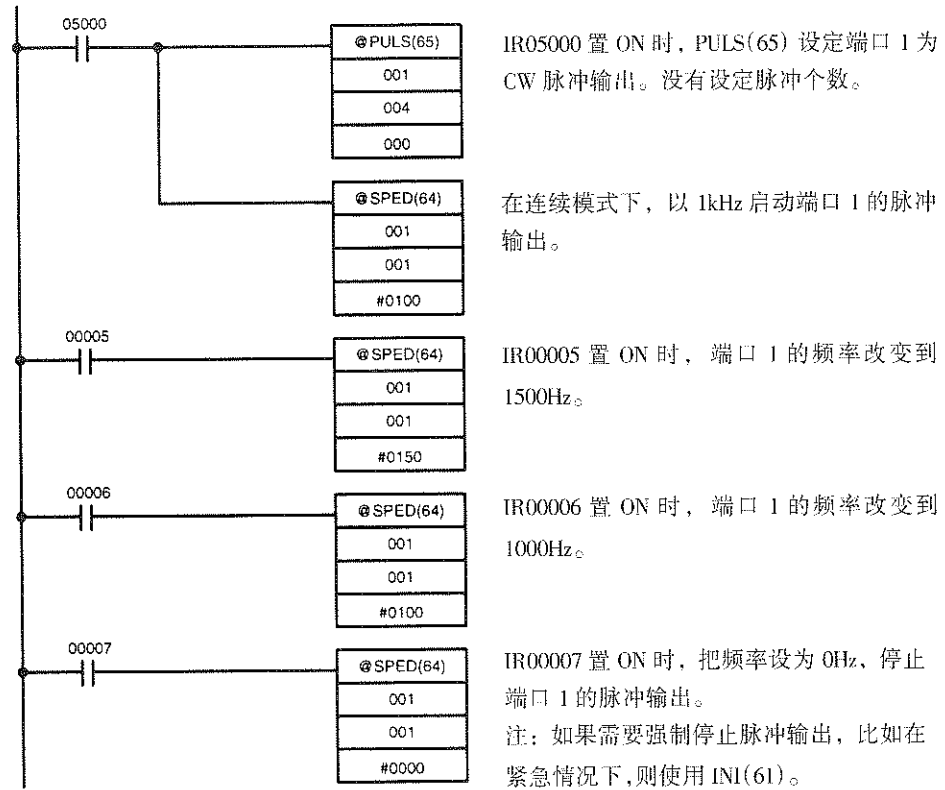


注意 确保在启动和停止马达时, 脉冲频率在马达的自启动频率范围内。

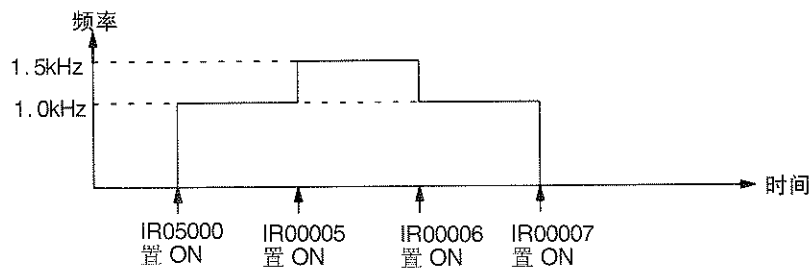
注 在频率改变作为输入中断处理过程中执行时, 速度控制定时很准确。

例 2: 用 SPED(64) 停止脉冲输出

下例显示用于控制端口 1 的脉冲输出的 PULS(65)和 SPED(64)。用不同的频率设定执行 SPED(64)修改频率,最后频率设为 0 停止。



下图显示程序执行时端口 1 的脉冲输出频率。



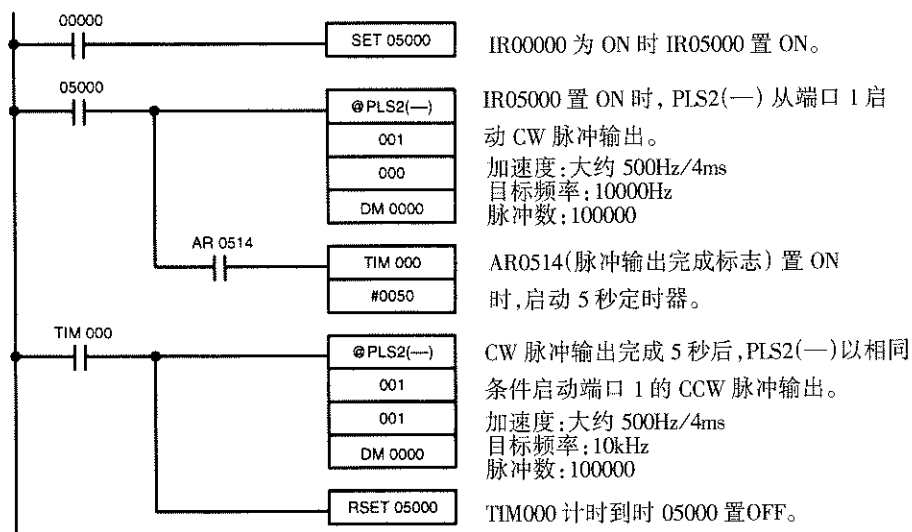
⚠ 注意 确保启动和停止马达时, 脉冲频率在马达的自启动频率范围内。

例 3: 使用 PLS2(—)
以相同的加速度/减
速度加速/减速频率

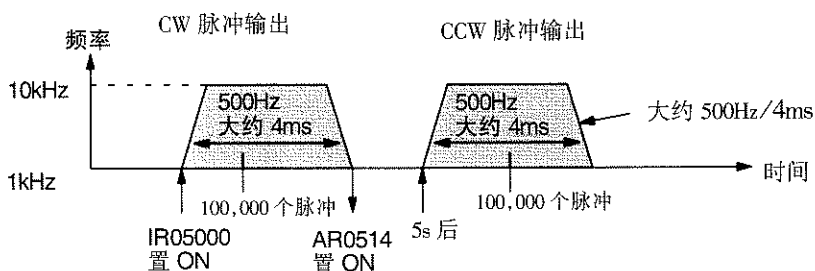
下例显示了用于从端口 1 输出 100000 个 CW 脉冲的 PLS2(—)。频率以大约 500Hz/4ms 加速到 10kHz,以相同减速度减速。

CW 脉冲输出 5 秒后,另一个 PLS2(—) 指令以相同设定输出 100000 个 CCW 脉冲。

| | |
|---------|------|
| DM 0000 | 0050 |
| DM 0001 | 1000 |
| DM 0002 | 0000 |
| DM 0003 | 0010 |



下图显示程序执行时端口 1 的脉冲输出频率。

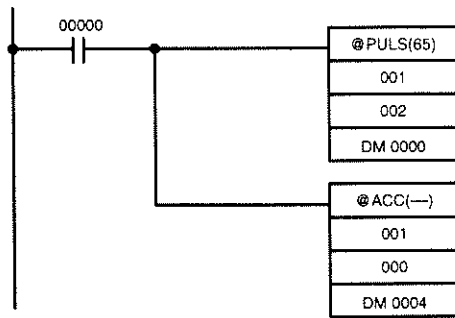


例 4: 用 ACC(—) 以不同的加速度加速/减速频率

下例显示了用于端口 1 输出 10000 个 CW 脉冲的 ACC(—) 的模式 0, 频率以大约 1kHz/4ms 加速到 10kHz, 以大约 250Hz/4ms 减速到 1kHz, 输出 9100 个脉冲后开始减速。

| | |
|---------|------|
| DM 0000 | 0000 |
| DM 0001 | 0001 |
| DM 0002 | 9100 |
| DM 0003 | 0000 |

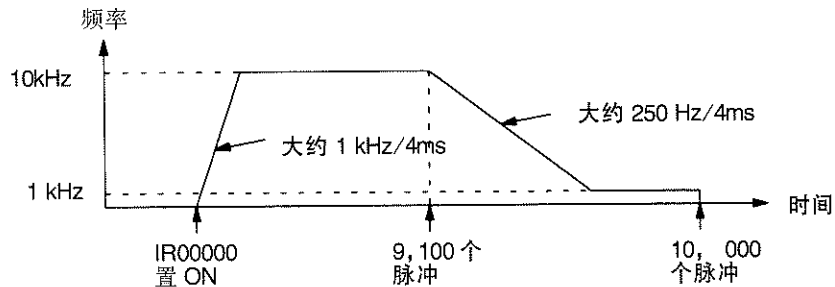
| | |
|---------|------|
| DM 0004 | 0100 |
| DM 0005 | 1000 |
| DM 0006 | 0025 |
| DM 0007 | 0050 |



IR00000 置 ON 时, PULS(65) 设定端口 1 用于 CW 脉冲输出。脉冲总数设为 10000, 减速点设为 9100 个脉冲。

启动端口 1 的 CW 脉冲输出。
 加速度: 大约 1000Hz/4ms。
 加速后的目标频率: 10000Hz。
 减速度: 大约 250Hz/4ms
 减速后的目标频率: 1kHz
 减速后从大约 500Hz/4ms 的目标频率开始输出脉冲。

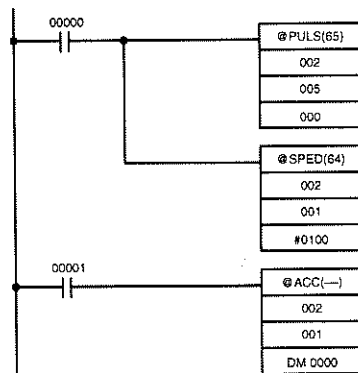
下图显示了程序执行时端口 1 的脉冲输出频率。



例 5: 用 ACC(—) 以指定的速度加速频率

下例显示了用于增加端口 1 的脉冲输出频率的 ACC(—) 的模式 1。频率以大约 500Hz/4ms 从 1kHz 加速到 20kHz。

| | |
|---------|------|
| DM 0000 | 0050 |
| DM 0001 | 2000 |

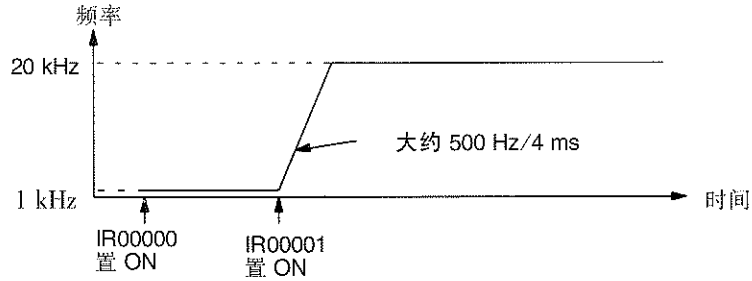


IR00000 置 ON 时, PULS(65) 设定端口 2 用于 CCW 脉冲输出, 脉冲数未设定。

以连续模式从端口 2 开始 1000Hz(1kHz) 脉冲输出。

IR00001 置 ON 时, ACC(—) 开始以大约 500Hz/4ms 加速端口 2 脉冲输出, 直到达到目标频率 20000Hz。

下图显示了程序执行时端口 2 的脉冲输出频率。



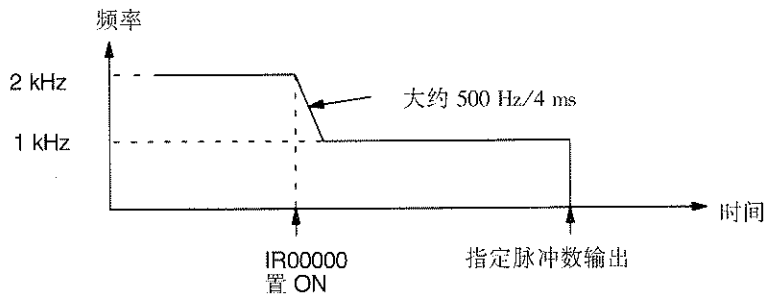
例 6: 用 ACC(—) 以指定速度减速频率和停止输出

下例显示了用于递减端口 1 的脉冲输出频率的 ACC(—) 的模式 2。独立模式的 2kHz 脉冲输出已在工作中, 当达到脉冲数时自动停止。

| | |
|---------|------|
| DM 0000 | 0050 |
| DM 0001 | 0001 |



下图显示了程序执行时端口 1 的脉冲输出频率。

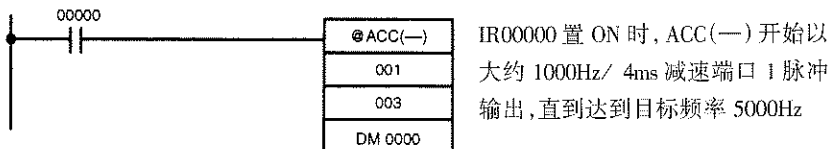


注 用目标频率为 0 执行 ACC(—) 模式 2 可停止脉冲输出, 然而, 因为脉冲输出不能在正确的脉冲数上停止, 不要使用此方法, 除非紧急停止。

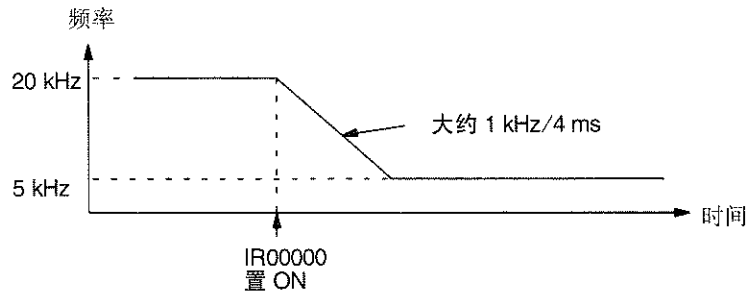
例 7: 使用 ACC(—) 以指定的速度减速频率

下例显示了用于递减端口 1 的脉冲输出频率的 ACC(—) 的模式 3。20kHz 脉冲输出正在以连续模式工作。

| | |
|---------|------|
| DM 0000 | 0100 |
| DM 0001 | 0500 |



下图显示了程序执行时端口 1 的脉冲输出频率。

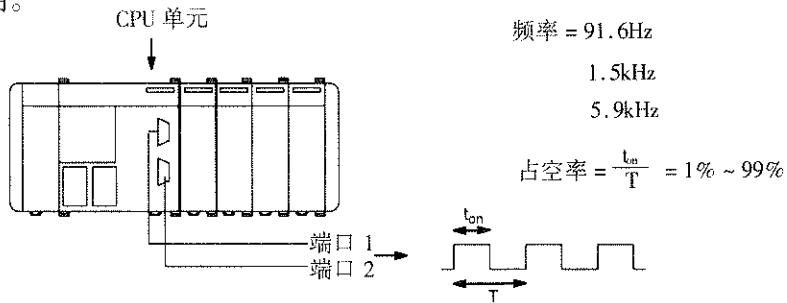


2-2-10 可变占空率脉冲输出

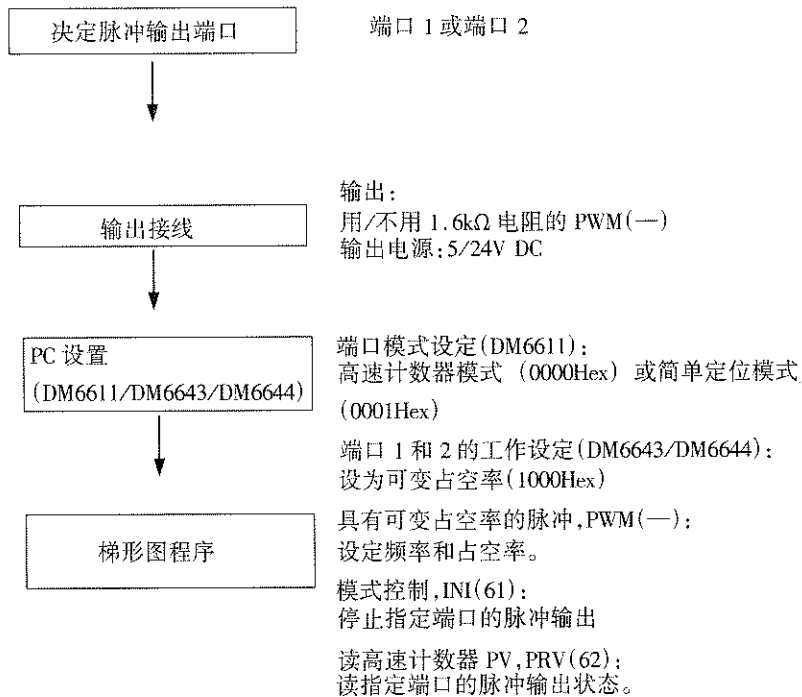
下面是从端口 1 和/或 2 输出具有可变占空率（即脉冲置 ON 的时间和脉冲周期的比例）的脉冲的步骤。此功能可用于不同种类的控制输出，例如光亮度输出或输出到变频器的速度控制。

提要

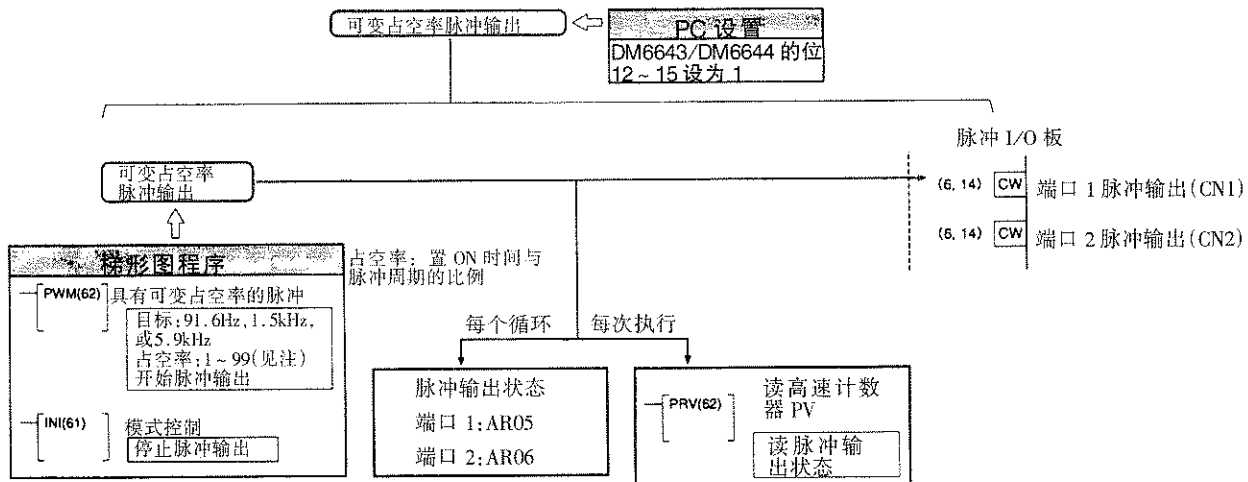
如下图所示执行从端口 1 和/或 2 的可变占空率脉冲输出，端口 1 和 2 可同时使用。



使用 PWM(-) 的可变占空率脉冲输出



可变占空率脉冲输出

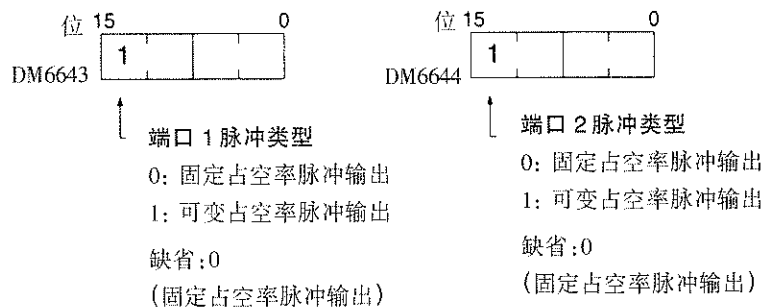


PC 设置设定

从端口 1 或 2 输出可变占空率脉冲之前,把 PC 切换到编程模式,并在 PC 设置中进行下列设定。

端口 1 和 2 的工作设定

进行下列设定,把端口 1(DM6643)或端口 2(DM6644)设为可变占空率脉冲输出模式。端口 1 和 2 可单独设定。

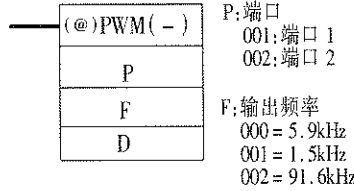


- 注
1. 端口设定为可变占空率脉冲输出时,不能输出固定占空率脉冲。
 2. 使用可变占空率脉冲输出时,所有指令都可使用,不管端口模式是什么。

| 指令 | PWM(-) | INI(61) | PRV(62) |
|---------|-------------------------|---------|---------|
| 功能 | 频率设定 占空率设定 开始脉冲输出 | 停止脉冲输出 | 读脉冲输出状态 |
| 高速计数器模式 | 允许 | | |
| 简单定位模式 | 允许 | | |

开始脉冲输出

PWM(—)用于指定端口号,脉冲频率,和占空率,并启动脉冲输出。

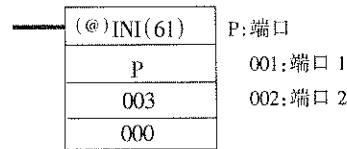


D: 占空率
指定一个 4 位数 BCD 常数或一个字地址, D 的值存贮为 4 位数 BCD, 表示百分比值。设定必须在 0001 和 0099 之间(即 1% ~ 99%)。

脉冲输出将用 PWM(—)指定的设定启动,并用这些设定继续输出,直到用不同的设定再次执行 PWM(—),或直到执行 INI(61)停止指定端口的脉冲输出。

停止脉冲输出

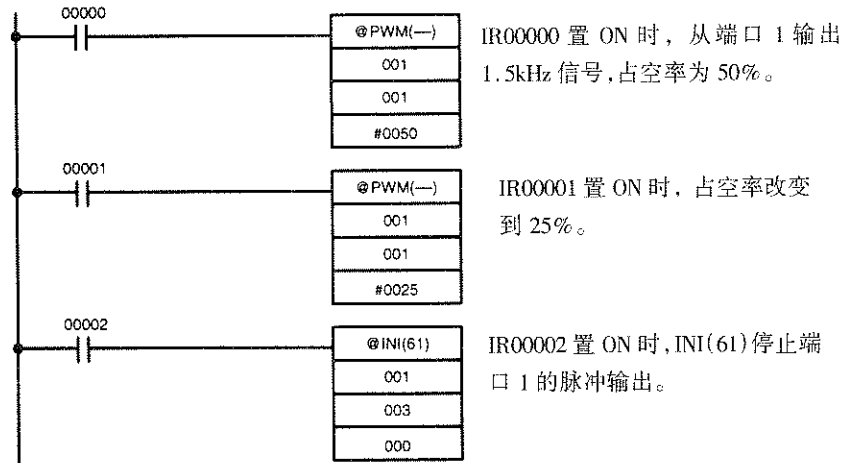
可通过执行 INI(61)(C=003)停止端口的脉冲输出。指定端口 1 或 2(P = 001 或 002)



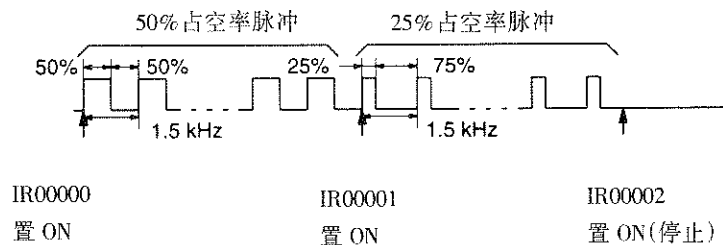
例:使用 PWM(—)

下例显示了用于从端口 1 启动 1.5kHz 脉冲输出,并把占空率从 50%改变到 25%的 PWM(—)。用 INI(61)停止脉冲输出,运行程序之前,确认 PC 设置中的设定如下:

DM6643:1000(端口 1 为可变占空率脉冲设定)



下图显示了程序执行时端口 1 的脉冲输出的占空率。



2-2-1 决定端口 1 和 2 的状态

端口 1 和 2 的脉冲输出(固定或可变占空率脉冲)状态可通过读 SR 和 AR 区中的相关标志状态或通过执行 PRV(62)决定。

读状态标志

下表显示了与端口 1 和 2 的脉冲输出状态有关的内存字。脉冲输出状态能通过读字的内容和这些字中显示的标志来决定。

· 内板错误代码

| 字 | 位 | 槽 | 功能 |
|-------|---------|-----|--|
| AR 04 | 08 ~ 15 | 槽 2 | 错误代码以 2 位数十六进制存储: 00Hex: 正常 01和02Hex: 硬件错误 02Hex: PC 设置错误 03Hex: 输出脉冲时 PC 停止 |

· 操作状态指示

| 字 | | 位 | 名称 | 功能 |
|------|------|----|----------|--|
| 端口 1 | 端口 2 | | | |
| AR05 | AR06 | 12 | 减速标志 | 指定减速时表示通过减速点 0: 未指定 1: 指定 |
| | | 13 | 脉冲数标志 | 存储是否指定了脉冲数 0: 未指定 1: 指定 |
| | | 14 | 脉冲输出完成标志 | 表示脉冲输出的完成状态。 0: 未完成 1: 完成 |
| | | 15 | 脉冲输出状态标志 | 表示脉冲输出的工作状态。 0: 停止脉冲输出 1: 正在输出脉冲 |

使用 PRV(62)

用 PRV(62)可决定脉冲输出状态。指定端口 1 或 2(P=001~002)和目标字 D。

| | |
|-------------|----------|
| (@) PRV(62) | |
| P | P: 端口指定符 |
| 001 | C: 001 |
| D | D: 第一目标字 |

存储在 D 中的包含脉冲输出状态信息的位具有下列意义:

| 位 | 功能 | 说明 |
|----|----------|-----------------------------------|
| 04 | 减速标志 | 表示减速 (0: 不减速, 1: 减速) |
| 05 | 脉冲数标志 | 表示是否指定了脉冲总数 (0: 未指定; 1: 指定) |
| 06 | 脉冲输出完成标志 | 表示脉冲输出是否完成 (0: 未完成; 1: 完成) |
| 07 | 脉冲输出状态标志 | 表示是否正在输出脉冲。 (0: 没有输出; 1: 正在输出) |

除以上之外, 位 0 和 1 存储关于高速计数器的状态信息。所有其它位为 0。

注 PRV(62)用于读端口状态时,读到最近的信息,与 PC 的循环时间无关。

2-2-12 使用脉冲输出功能的注意事项

脉冲 I/O 板把 500kHz 源时钟除以整数值,产生输出脉冲频率。因此,频率设定和产生的实际频率可能不同。参见下列公式计算实际频率。

设定频率:

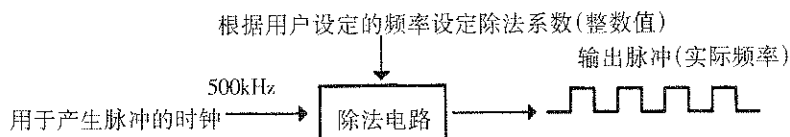
用户设定输出频率。

除法系数:

在除法电路中设定整数值,产生设定频率的输出脉冲。

实际频率:

实际输出脉冲频率由除法电路产生。



$$\text{实际频率(kHz)} = 500(\text{kHz}) / \text{INT}(500(\text{kHz}) / \text{设定频率(kHz)})$$

INT: 求出整数值的函数

INT(500/设定频率); 除法系数

设定频率和实际频率之间的差随着频率提高而增大,如下表所示。

| 设定频率(kHz) | 实际频率(kHz) |
|---------------|-----------|
| 45.46 ~ 50.00 | 50.00 |
| 41.67 ~ 45.45 | 45.45 |
| 38.47 ~ 41.66 | 41.67 |
| 31.26 ~ 33.33 | 33.33 |
| 29.42 ~ 31.25 | 31.25 |
| 27.78 ~ 29.41 | 29.41 |
| 20.01 ~ 20.83 | 20.83 |
| 19.24 ~ 20.00 | 20.00 |
| 18.52 ~ 19.23 | 19.23 |
| 10.01 ~ 10.20 | 10.20 |
| 9.81 ~ 10.00 | 10.00 |
| 9.62 ~ 9.80 | 9.80 |
| 5.01 ~ 5.05 | 5.05 |
| 4.96 ~ 5.00 | 5.00 |
| 4.90 ~ 4.95 | 4.95 |
| 3.02 ~ 3.03 | 3.03 |
| 3.00 ~ 3.01 | 3.01 |
| 2.98 ~ 2.99 | 2.99 |

2-3 绝对值编码器接口板

2-3-1 型号

| 名称 | 型号 | 规格 |
|-----------|-------------|------------|
| 绝对值编码器接口板 | CQM1H-ABB21 | 绝对值编码器2个输入 |

2-3-2 功能

具有中断功能的绝对值高速计数器

绝对值编码器接口板是对从绝对值 (ABS) 旋转编码器来的两个格雷二进制代码输入计数的内板。

绝对值编码器接口板读通过端口 1 和 2 从绝对值编码器以最大计数速率 4kHz 输入的二进制格雷代码 (反转二进制代码), 并根据输入值执行处理。

工作模式

BCD 模式和 360°模式

分辨率

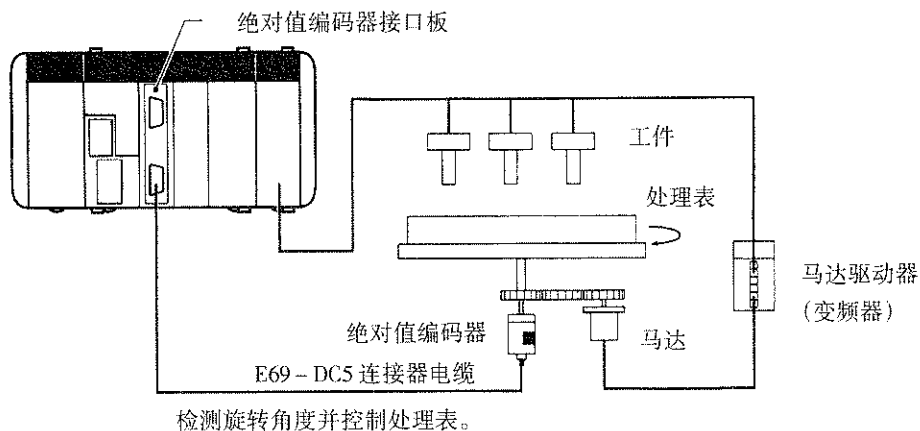
可设定下列之一: 8 位 (0~255), 10 位 (0~1023), 或 12 位 (0~4095)。分辨率应设定为与所连接的编码器的分辨率相同。

中断

绝对值高速计数器的 PV (当前值) 等于指定目标值或在指定的比较范围内时, 可执行中断子程序。

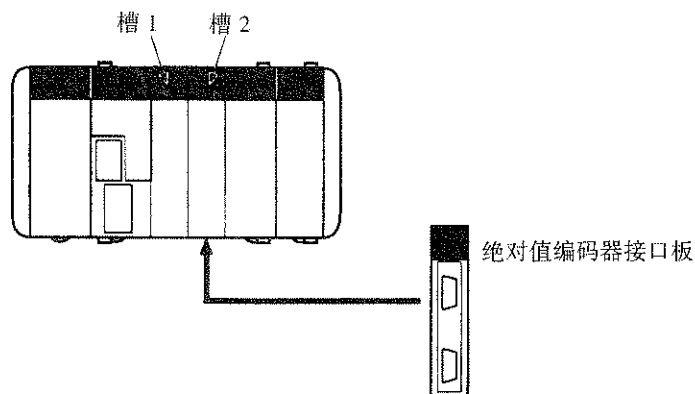
注 使用绝对值编码器意味着即使在电源中断时仍可保持定位数据, 重新上电时不需执行原点返回。此外, 原点补偿功能允许用户指定任意位置作为原点。

2-3-3 系统配置



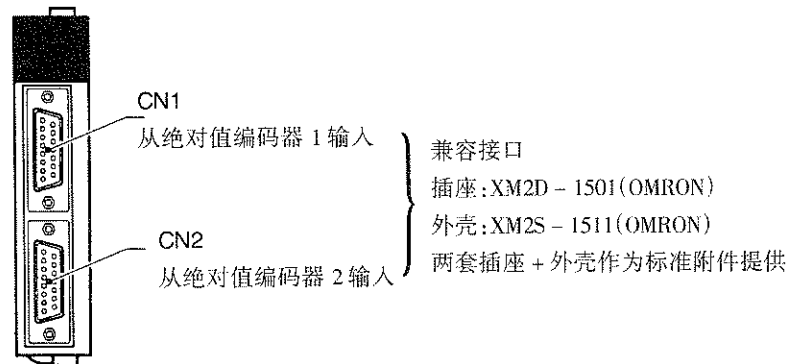
2-3-4 适用的内板槽

绝对值编码器接口板只能安装在 CQM1-CPU51/61 CPU 单元的槽 2 (右槽) 上。

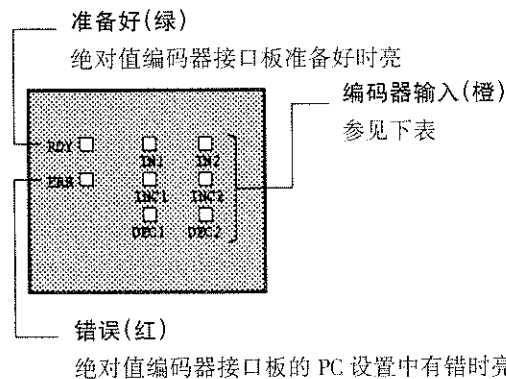


2-3-5 名称和功能

绝对值编码器接口板具有接收从绝对值旋转编码器输入的二进制格雷代码的端口 1 连接器 CN1 和端口 2 连接器 CN2。



LED 指示灯



| 编码器输入指示灯 | | 功 能 |
|----------|------|---------------|
| 端口 1 | 端口 2 | |
| IN1 | IN2 | 输入位 0 为 ON 时亮 |
| INC1 | INC2 | 输入值递增时亮 |
| DEC1 | DEC2 | 输入值递减时亮 |

2-3-6 绝对值编码器输入规格

| 指 令 | 意 义 |
|--------------|--|
| (@) CTBL(63) | 用于注册目标或范围比较表或启动前一个注册的比较表的比较。 |
| (@) INI(61) | 用于启动或停止注册比较表比较或修改高速计数器的 PV |
| (@) PRV(62) | 用于读高速计数器的 PV 或状态 |
| (@) INT(89) | 用于执行屏蔽所有中断, 例如输入中断, 间隔定时器中断, 和高速计数器中断。 |

相关标志和位

槽 2 中的绝对值编码器接口板的位

| 字 | 位 | 名 称 | | 功 能 |
|---------------|---------|-----|------------|---|
| IR232 | 00 ~ 15 | 端口1 | PV字(最右4位数) | 在每个循环后, 连接到绝对值编码器接口板端口1的绝对值高速计数器的PV以8位数BCD存贮。 |
| IR233 | 00 ~ 15 | | PV字(最左4位数) | |
| IR234 | 00 ~ 15 | 端口2 | PV字(最右4位数) | |
| IR235 | 00 ~ 15 | | PV字(最左4位数) | |
| IR236 ~ IR243 | 00 ~ 15 | | 未使用 | --- |

AR 标志

| 字 | 位 | 名 称 | | 功 能 | |
|------|----|-----------|--------------------------------|--------------------|-----------------------------------|
| AR05 | 00 | 端口1 | 高速计数器范围比较标志 | 计数器PV满足比较范围1条件时置ON | 在范围比较模式下使用高速计数器1时, 满足相应条件时每个位置ON。 |
| | 01 | | | 计数器PV满足比较范围2条件时置ON | |
| | 02 | | | 计数器PV满足比较范围3条件时置ON | |
| | 03 | | | 计数器PV满足比较范围4条件时置ON | |
| | 04 | | | 计数器PV满足比较范围5条件时置ON | |
| | 05 | | | 计数器PV满足比较范围6条件时置ON | |
| | 06 | | | 计数器PV满足比较范围7条件时置ON | |
| | 07 | | | 计数器PV满足比较范围8条件时置ON | |
| | 08 | 高速计数器比较标志 | 表示比较操作的状态 OFF: 停止 ON: 比较 | | |
| AR06 | 00 | 端口2 | 高速计数器范围比较标志 | 计数器PV满足比较范围1条件时置ON | 在范围比较模式下使用高速计数器2时, 满足相应条件时每个位置ON。 |
| | 01 | | | 计数器PV满足比较范围2条件时置ON | |
| | 02 | | | 计数器PV满足比较范围3条件时置ON | |
| | 03 | | | 计数器PV满足比较范围4条件时置ON | |
| | 04 | | | 计数器PV满足比较范围5条件时置ON | |
| | 05 | | | 计数器PV满足比较范围6条件时置ON | |
| | 06 | | | 计数器PV满足比较范围7条件时置ON | |
| | 07 | | | 计数器PV满足比较范围8条件时置ON | |
| | 08 | 高速计数器比较标志 | 表示比较操作的状态 OFF: 停止 ON: 比较 | | |

SR 标志

| 字 | 位 | 功 能 |
|-------|----|---------------------|
| IR252 | 01 | 绝对值高速计数器1原点补偿位(端口1) |
| | 02 | 绝对值高速计数器2原点补偿位(端口2) |
| IR254 | 15 | 内板错误标志 |

AR 区域

| 字 | 位 | 名称 | 功能 |
|-------|---------|-------------|--|
| AR 04 | 08 ~ 15 | 槽2中的内板的错误代码 | 00Hex: 没有错误 01或02Hex: 硬件错误 03Hex: PC设置错误 |

有关 PC 设置设定

| 字 | 位 | 功 能 | | 设定有效的时间 |
|--------|---------|-----------------|--|-----------------|
| DM6611 | 00 ~ 15 | 存贮端口1原点补偿值(BCD) | 0000 ~ 4095(4位数BCD) 原点补偿位(端口1的SR25201, 端口2的SR25202)置ON时补偿原点。补偿值以BCD模式或360°模式设定为在0000和4095之间的4位数BCD。 | 在编程模式下原点补偿位置ON时 |
| DM6612 | 00 ~ 15 | 存贮端口2原点补偿值(BCD) | | |
| DM6643 | 00 ~ 07 | 端口1 | 分辨率 00Hex: 8位 01Hex: 10位 02Hex: 12位 | 开始工作时 |
| | 00 ~ 15 | | 工作模式设定 00Hex: BCD模式 01Hex: 360°模式 | |
| DM6644 | 00 ~ 07 | 端口2 | 分辨率 00Hex: 8位 01Hex: 10位 02Hex: 12位 | |
| | 00 ~ 15 | | 工作模式设定 00Hex: BCD模式 01Hex: 360°模式 | |

2-3-7 高速计数器中断

绝对值编码器接口板连接一个绝对值编码器。根据从绝对值旋转编码器输入到端口1或2的二进制格雷码信号,可执行中断处理。

两个端口可独立运行。端口1计数器称为绝对值高速计数器1,端口2计数器称为绝对值高速计数器2。本节介绍了怎样使用绝对值高速计数器1和2,计数速率为4kHz。

处理

输入信号和工作模式

有两种工作模式可用于绝对值高速计数器1和2。

- 1,2,3... 1. BCD模式:
绝对值旋转编码器的二进制格雷码首先转换成标准二进制(十六进制)数据,然后转换成BCD。
2. 360°模式:
对于分辨的最大值取360°,绝对值旋转编码器的输入转换或0°和359°之

间的一个角度。CTBL(63)的设定以 5°为单位。

输入到端口 1 和 2 的二进制格雷码的分辨率必须是下表所列的三种分辨率之一。此表也显示了在每种工作模式下,与每种分辨率有关的数值范围。

| 分辨率 | 可能的PV | |
|-----|----------|---|
| | BCD模式 | 360°模式 |
| 8位 | 0 ~ 255 | PV输出:0° ~ 359°(1°为单位) 比较表设定:0° ~ 355°(5°为单位) |
| 10位 | 0 ~ 1023 | |
| 12位 | 0 ~ 4095 | |

在 360°模式下设定绝对值高速计数器

下表显示了怎样把以 5°为单位进行的设定根据分辨率转换成二进制格雷代码。

5° ~ 45°

| 分辨率 | 5° | 10° | 15° | 20° | 25° | 30° | 35° | 40° | 45° |
|-----|----|-----|-----|-----|-----|-----|-----|-----|-----|
| 8位 | 4 | 7 | 11 | 14 | 18 | 21 | 25 | 28 | 32 |
| 10位 | 14 | 28 | 43 | 57 | 71 | 85 | 100 | 114 | 128 |
| 12位 | 57 | 114 | 171 | 228 | 284 | 341 | 398 | 455 | 512 |

50° ~ 355°

根据上面给出的 5° ~ 45°范围内的转换,其它值的转换如下计算:

设定(°) ÷ 45° = A, 余数为 B(°)。

转换 = (45°转换) × A + (B 转换)

例如, 145°, 8 位分辨率

$145^\circ \div 45^\circ = 3$, 余数为 10°

所以, 转换值 = $32 \times 3 + 7 = 103$

对 10 和 12 位分辨率,即使在 PV 符合比较条件时,计算结果中的微小差别都可导致中断处理不执行。

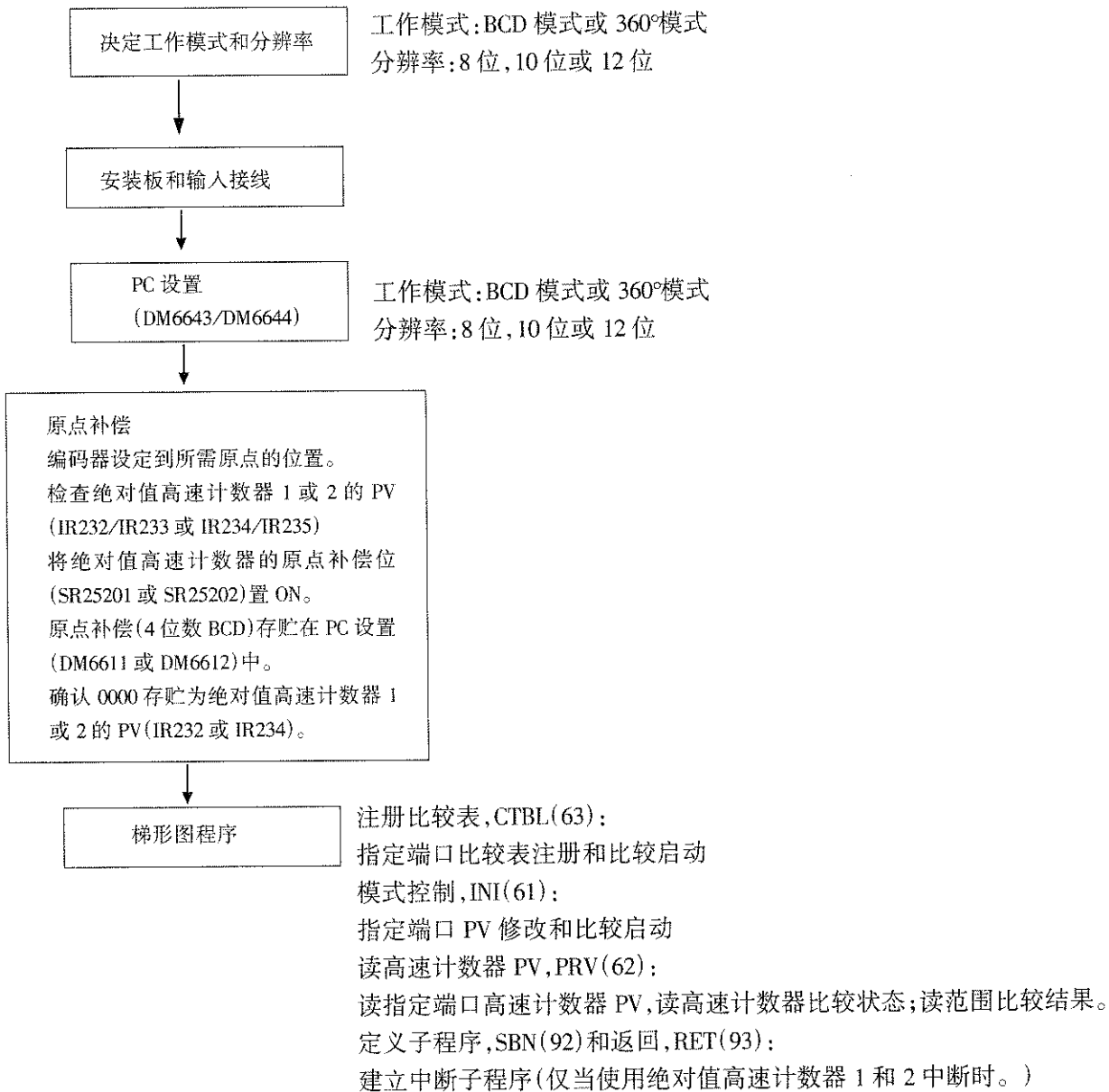
绝对值高速计数器中断计数

可用下列两种方法检查计数器的 PV。

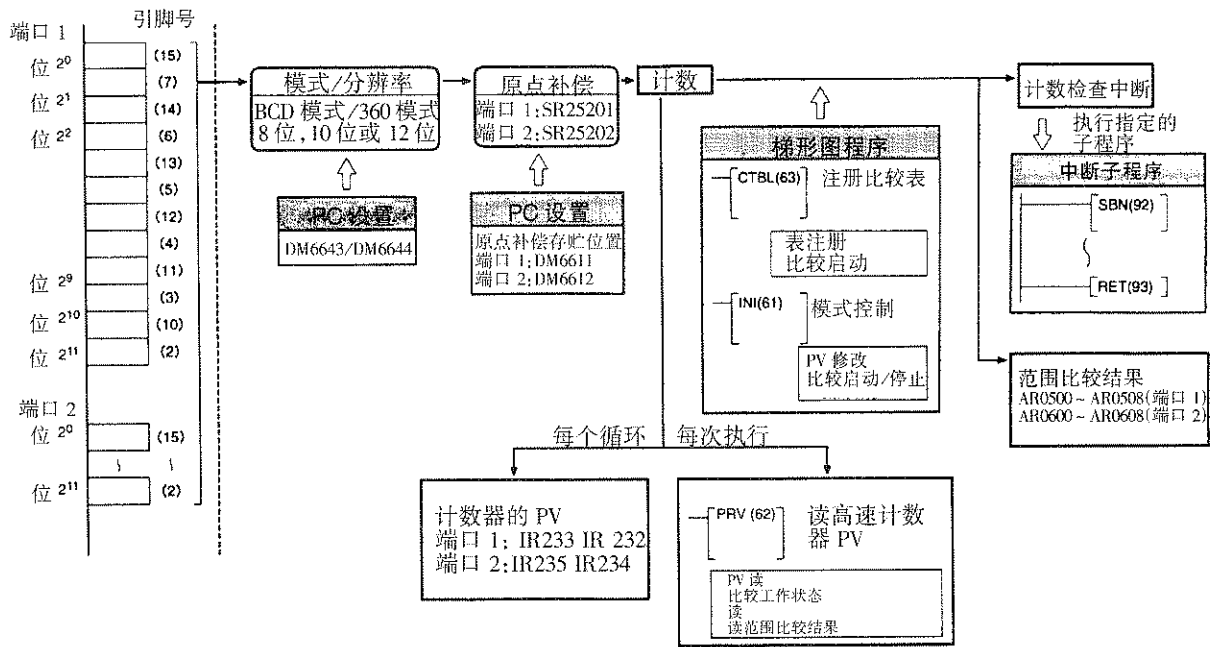
- 目标值方法
- 范围比较方法

每种方法的介绍参见 31 页。

使用绝对值高速计数器的步骤



高速计数器功能

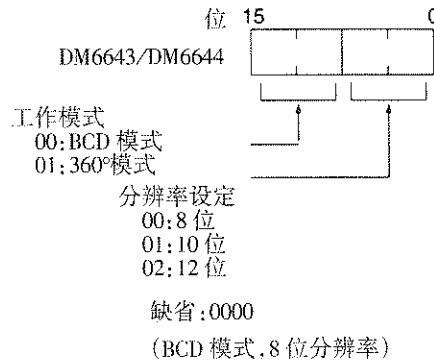


预置 PC 设置

在程序中使用绝对值高速计数器 1 或 2 的中断之前,要在编程模式下进行下列设定。

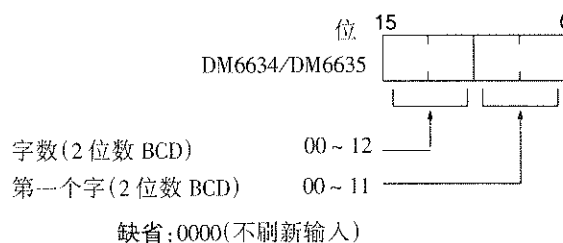
绝对值高速计数器设定

DM6643 包含绝对值高速计数器 1 的设定, DM6644 包含绝对值高速计数器 2 的设定,这些字决定了工作模式和分辨率设定。



输入刷新字设定

DM6634 包含绝对值高速计数器 1 的输入刷新字设定, DM6635 包含绝对值高速计数器 2 的设定。必须刷新输入时要进行这些设定。



原点补偿

可以补偿绝对值编码器原点 and 实际原点之间的差异。设定原点补偿后,在绝对值编码器的数据作为 PV 输出之前自动调整。一旦设定,原点补偿一直起作用,直到执行下一个原点补偿;即使在掉电后仍起作用。可分别为端口 1 和 2 设置原点补偿。

缺省设定为无原点补偿。

按下列步骤设定原点补偿。

- 1,2,3... 1. 设定绝对值编码器到所需原点位置。
2. 确认 CQM1H CPU 单元的 DIP 开关的 1 脚为 OFF(允许编程设备写 DM6144 到 DM6568),然后把 PC 切换到编程模式。
3. 在 DM6643 或 DM6644 中设定绝对分辨率。
4. 确认未出现致命错误或 FALS 9C 错误。
5. 在补偿原点前从 IR232 和 IR233(端口 1)或 IR234 和 IR235(端口 2)读绝对值高速计数器的 PV 以决定值。
6. 从编程设备把绝对值高速计数器 1 原点补偿位 (SR25201) 或绝对值高速计数器 2 原点补偿位 (SR25202) 置 ON。
补偿值写到 DM6611(端口 1) 或 DM6612(端口 2), 并且原点补偿位自动置 OFF。不论计数器设定为 BCD 模式还是 360°模式, 补偿值存贮为 0000 和 4095 之间的 4 位数 BCD。
7. 读高速计数器的 PV 字, 确保原点补偿已正常完成 (原点补偿后 PV 将为 0000)

补偿值一直起作用,直到被上面的步骤再次修改。

编程

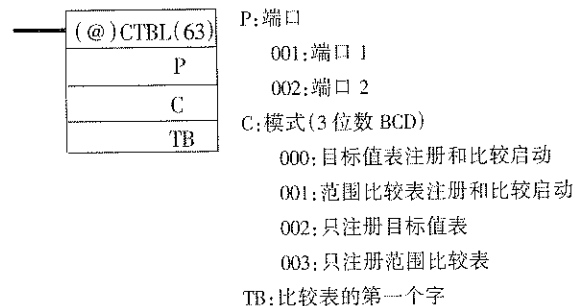
使用下列步骤编写绝对值高速计数器 1 和 2 的程序。

允许 PC 设置设定时,绝对值高速计数器 1 和 2 开始计数,但是比较表不进行比较,不产生中断,除非执行了 CTBL(63)指令。

绝对值高速计数器 1 的 PV 保存在 IR232 和 IR233 中,绝对值高速计数器 2 的 PV 保存在 IR234 和 IR235 中。

启动和停止比较

- 1,2,3... 1. 使用 CTBL(63)指令保存 CQM1H 中的比较表并开始比较。



P 指定端口。设定 P=001 指定绝对值高速计数器 1(即端口 1),或 P=002 指定绝对值高速计数器 2(端口 2)。

C 值设定为 000 注册目标值比较表,设为 001 注册范围比较表。完成注册后

开始比较。执行比较时,根据应用的比较表执行绝对值高速计数器中断。比较表注册的详细内容,参见 5-16-7,注册比较表 - CTBL(63)。

如果 C 设为 002,则用目标值方法进行比较;如果是 003,则用范围比较方法进行。在两种情况下都能保存比较表,但是并不开始实际比较,直到使用 INI(61)。

注 与其它高速计数器不同,比较表中注册的绝对值高速计数器 1 和 2 的中断,目标值和上下限都在一个字中设定。

2. 执行如下所示的 INI(61),以停止比较。在 P 中指定端口 1 或 2(P = 001 或 002)。

| | |
|------------|-----------|
| (@)INI(61) | |
| P | P: 端口 |
| 001 | 001: 端口 1 |
| 000 | 002: 端口 2 |

把第一个操作数设为端口号,第二个操作数设为 000(执行比较),并执行 INI(61),重新启动比较。

运行时(即程序执行时)已保存的表将保留在 CQM1H 中,直到保存新表。

读绝对值高速计数器 1 和 2 的 PV

下列两种方法可用于读绝对值高速计数器 1 和 2 的 PV。

- 从内存(IR232 或 IR234)读 PV
- 使用 PRV(62)

从内存读 PV

高速计数器 1 和 2 的 PV 以 8 位数 BCD 存贮在数据区字中,不论板处于 BCD 模式还是 360°模式。

| | 最左 4 位数 | 最右 4 位数 | BCD 模式 | 360°模式 |
|-------|------------|------------|-----------------------|-----------------------|
| 端口 1: | IR233 | IR232 | 0000 0000 ~ 0000 4095 | 0000 0000 ~ 0000 0359 |
| 端口 2: | IR235 | IR234 | | |

注 这些字每个循环只刷新一次,因此它们可能与实际 PV 不同。

使用 PRV(62)

PRV(62)用于读绝对值高速计数器 1 和 2 的 PV。在 P 中指定绝对值高速计数器 1 或 2(P = 001 或 002)。

| | |
|------------|-----------|
| (@)PRV(62) | |
| P | P: 端口 |
| 000 | 001: 端口 1 |
| D | 002: 端口 2 |
| | D: 第一个目标字 |

如下所示存贮指定绝对值高速计数器的 PV。不论板处于 BCD 模式还是 360°模式,PV 以 8 位数 BCD 存贮。

| 最左 4 位数 | 最右 4 位数 | BCD 模式 | 360°模式 |
|------------|------------|-----------------------|-----------------------|
| D+1 | D | 0000 0000 ~ 0000 4095 | 0000 0000 ~ 0000 0359 |

注 执行 PRV(62)时可准确地读 PV。

读绝对值高速计数器状态

有两种方法读高速计数器 1 和 2 的状态：

- 读 AR 区标志
- 使用 PRV(62)

读 AR 区标志

下面列出与绝对值高速计数器 1 和 2 有关的 CQMIH 字。通过读这些数据字，可以决定绝对值高速计数器 1 和 2 的状态。

· 内板错误代码

| 字 | 位 | 功 能 | |
|------|---------|-----|--|
| AR04 | 08 ~ 15 | 槽2 | 存贮的错误代码如下： 00Hex: 正常 01或02Hex: 硬件错误 03Hex: PC设置错误 |

· 表示运行状态的字

| 字 | | 位 | 名 称 | 功 能 | |
|------|------|----|-------------|----------------------------|-----------------------------|
| 计数器1 | 计数器2 | | | | |
| AR05 | AR06 | 00 | 高速计数器范围比较标志 | 满足第一个条件时置ON | 用范围比较格式使用高速计数器时，满足相应条件时置ON。 |
| | | 01 | | 满足第二个条件时置ON | |
| | | 02 | | 满足第三个条件时置ON | |
| | | 03 | | 满足第四个条件时置ON | |
| | | 04 | | 满足第五个条件时置ON | |
| | | 05 | | 满足第六个条件时置ON | |
| | | 06 | | 满足第七个条件时置ON | |
| | | 07 | | 满足第八个条件时置ON | |
| | | 08 | 高速计数器比较标志 | 表示比较操作的状态。 0:停止 1:运行 | |

使用 PRV(62)

也可由执行 PRV(62)决定绝对值高速计数器 1 和 2 的状态。指定高速计数器 1 或 2(P=001 或 002)和目标字 D。

| | |
|------------|-----------|
| (@)PRV(62) | |
| P | P: 端口 |
| 001 | 001: 端口 1 |
| D | 002: 端口 2 |
| | D: 第一个目标字 |

指定高速计数器的状态存贮在 D 的位 00 中，如下表所示。

| 位 | 功 能 |
|----|-------------------|
| 00 | 比较操作标志(0:停止;1:运行) |

位 01 ~ 15 置 0。

操作例

本例显示了在端口 1 接收绝对值旋转编码器的输入信号，并用此输入控制输出 IR10000 ~ IR10003 的程序。绝对值高速计数器 1 设定为 8 位分辨率和 360°模式，

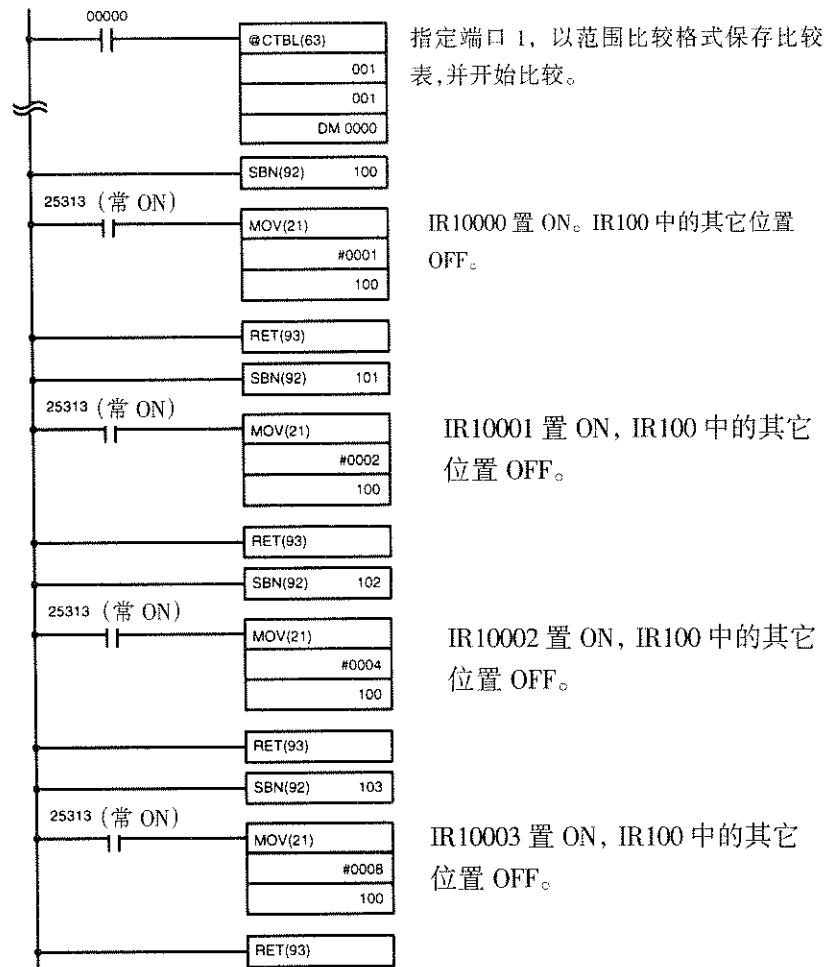
并执行范围比较。程序执行前,DM6643 设定为 0100(端口 1:360°模式,8 位分辨率)。

其它 PC 设置设定使用缺省设定。(中断处理时不刷新输入。)

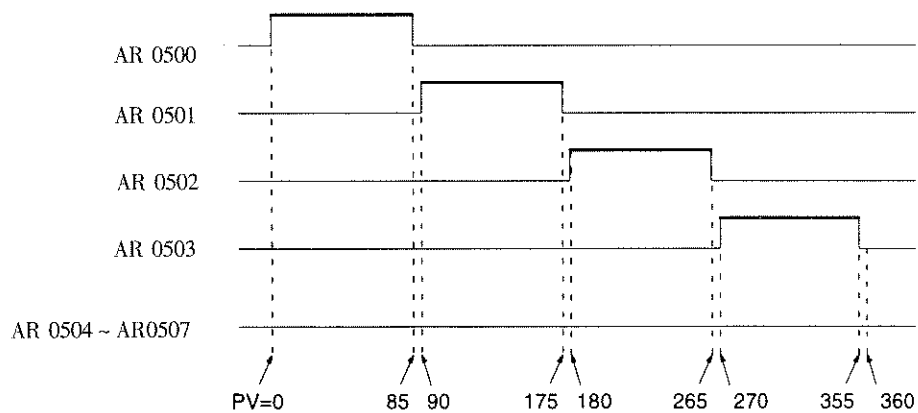
此外,在比较表中存贮下列数据。

| | | | |
|--------|------|--------------|---------------------------|
| DM0000 | 0000 | 下限 # 1(0°) | } → 第一个范围设定 (0° ~ 85°) |
| DM0001 | 0085 | 上限 # 1(85°) | |
| DM0002 | 0100 | 子程序号 100 | } → 第二个范围设定 (90° ~ 175°) |
| DM0003 | 0090 | 下限 # 2(90°) | |
| DM0004 | 0175 | 上限 # 2(175°) | } → 第三个范围设定 (180° ~ 265°) |
| DM0005 | 0101 | 子程序号 101 | |
| DM0006 | 0180 | 下限 # 3(180°) | } → 第四个范围设定 (270° ~ 355°) |
| DM0007 | 0265 | 上限 # 3(265°) | |
| DM0008 | 0102 | 子程序号 102 | } → 第五个范围设定 (未使用) |
| DM0009 | 0270 | 下限 # 4(270°) | |
| DM0010 | 0355 | 上限 # 4(355°) | } → 第六个范围设定 (未使用) |
| DM0011 | 0103 | 子程序号 103 | |
| DM0012 | 0000 | 下限 # 1(0°) | } → 第七个范围设定 (未使用) |
| DM0013 | 0000 | 下限 # 1(0°) | |
| DM0014 | FFFF | 无子程序号 | } → 第八个范围设定 (未使用) |
| DM0015 | 0000 | 下限 # 1(0°) | |
| DM0016 | 0000 | 上限 # 1(0°) | } → 第八个范围设定 (未使用) |
| DM0017 | FFFF | 无子程序号 | |
| DM0018 | 0000 | 下限 # 1(0°) | } → 第八个范围设定 (未使用) |
| DM0019 | 0000 | 上限 # 1(0°) | |
| DM0020 | FFFF | 无子程序号 | } → 第八个范围设定 (未使用) |
| DM0021 | 0000 | 下限 # 1(0°) | |
| DM0022 | 0000 | 上限 # 1(0°) | } → 第八个范围设定 (未使用) |
| DM0023 | FFFF | 无子程序号 | |

在 360°模式中,上下限以 5°为单位设定。



下图显示了执行上面的指令时, 绝对值高速计数器 1 的 PV 和范围比较结果标志 AR0500 ~ AR0507 之间的关系。



2-4 模拟设定板

2-4-1 型号

| 名称 | 型号 | 规格 |
|-------|-------------|----------|
| 模拟设定板 | CQM1H-AVB41 | 4个模拟设定螺丝 |

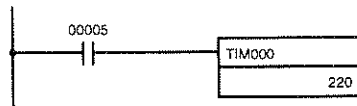
2-4-2 功能

用于模拟设定板前面的 4 个可变电阻设定的每一个值，以 0000 ~ 0200 之间的 4 位数 BCD 存贮在模拟设定字中 (IR220 ~ IR223)。

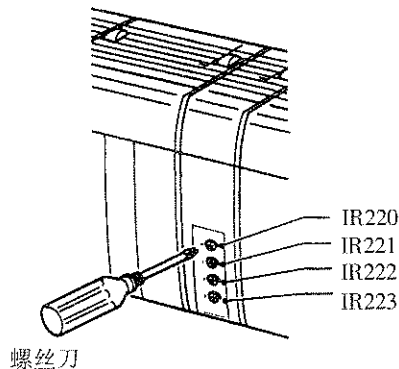
例如，使用模拟设定板，操作者可以用模拟设定 (IR220 ~ IR223) 设置定时器指令的值，就可以用螺丝刀调整控制传送带的升降速或定时，而不需用编程设备。

使用模拟定时器

以下显示了存贮在 IR220 ~ IR223 中，用作定时器设定的 4 位数 BCD 设定 (0000 ~ 0200) 的例子。

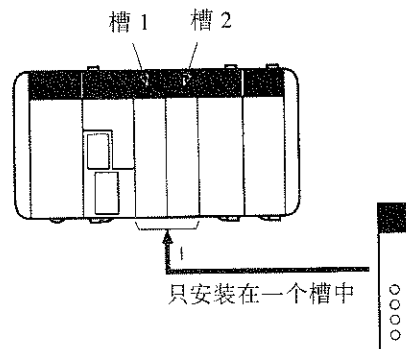


TIM000 的设定用 IR220 作为外部设置。
(用模拟控制 0 的设定执行定时器)



2-4-3 适用的内板插槽

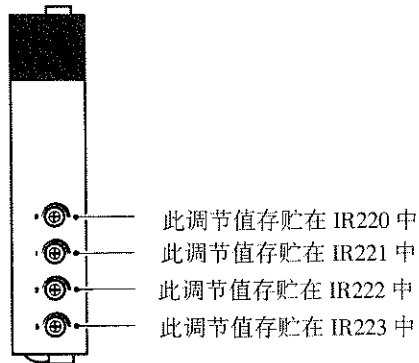
模拟设定板可安装在 CQM1H-CPU51/61 CPU 单元的槽 1(左槽) 或槽 2(右槽) 中。但是，两个槽不能同时使用。



2-4-4 名称和功能

模拟设定板的 4 个模拟控制位于前面板上。前面板上没有指示灯。设定值随着顺时针旋转调节而增大。使用小的螺丝刀。

指定 IR220 ~ IR223 作为 TIM 指令的设定值时，可以将此板用作模拟定时器。启动定时器时，模拟设定作为定时器设定值存贮。



注意 电源上电时，IR220 ~ IR223 的内容对应调节值一直在刷新。请确认这些字未从程序或编程设备写入。

2-4-5 规格

相关位

不论板安装在哪个槽中，模拟设定板的模拟调节值存贮在内板区的下列地址中。

| 字 | 位 | 名称 | 功能 |
|-------|---------|-------|--|
| IR220 | 00 ~ 15 | 模拟调节1 | 在每个循环，模拟调节 0 ~ 3 的值存贮为 0000 ~ 0200 之间的 4 位数 BCD 值。 |
| IR221 | 00 ~ 15 | 模拟调节2 | |
| IR222 | 00 ~ 15 | 模拟调节3 | |
| IR223 | 00 ~ 15 | 模拟调节4 | |

有关的 PC 设置设定 无

2-5 模拟量 I/O 板

2-5-1 型号

| 名称 | 型号 | 规格 |
|-----------|-------------|---|
| 模拟量 I/O 板 | CQM1H-MAB42 | 4 个模拟量输入 (-10 ~ +10V; 0 ~ 5V; 0 ~ 20mA, 每个点具有独立的信号范围) 2 个模拟量输出 (-10 ~ +10V; 0 ~ 20mA; 每个点具有独立的信号范围。 |

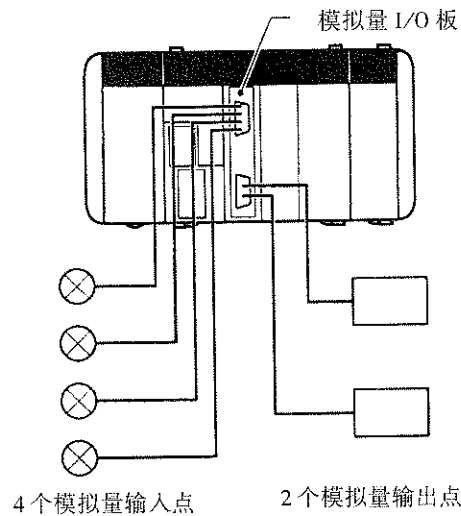
2-5-2 功能

模拟量 I/O 板是具有 4 个模拟量输入和 2 个模拟量输出的内板。

可用于每一个模拟量输入点的信号范围是 $-10 \sim +10V$, $0 \sim 5V$, 和 $0 \sim 20mA$ 。为每一点单独设定范围。DM6611 中的设定决定了信号范围。

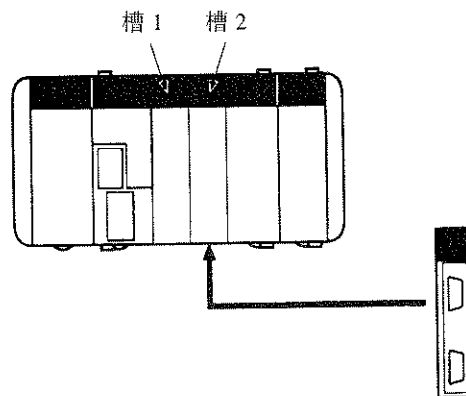
可用于每一个模拟量输出点的信号范围是 $-10 \sim +10V$ 和 $0 \sim 20mA$ 。可为每一点选择单独的信号范围。DM6611 中的设定决定信号范围。

2-5-3 系统配置



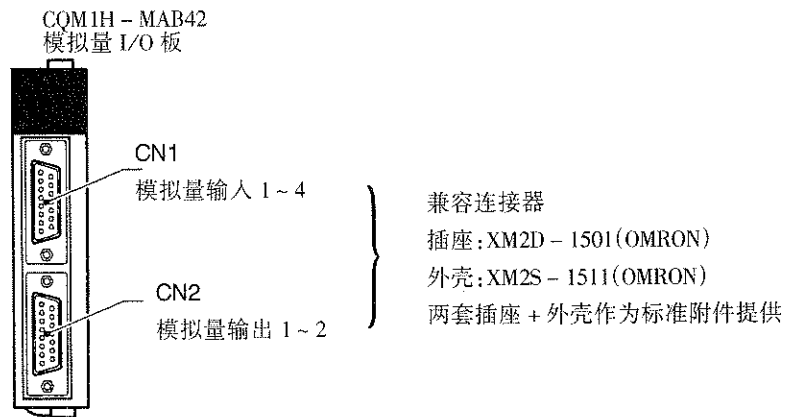
2-5-4 适用的内板槽

模拟量 I/O 板只能安装在 CQM1H - CPU 51/61 CPU 单元的槽 2(右槽)中。

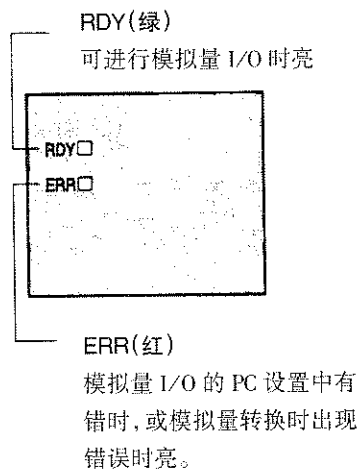


2-5-5 名称和功能

模拟量 I/O 板具有用于 4 个模拟量输入的 CN1 连接器和用于 2 个模拟量输出的 CN2 连接器。

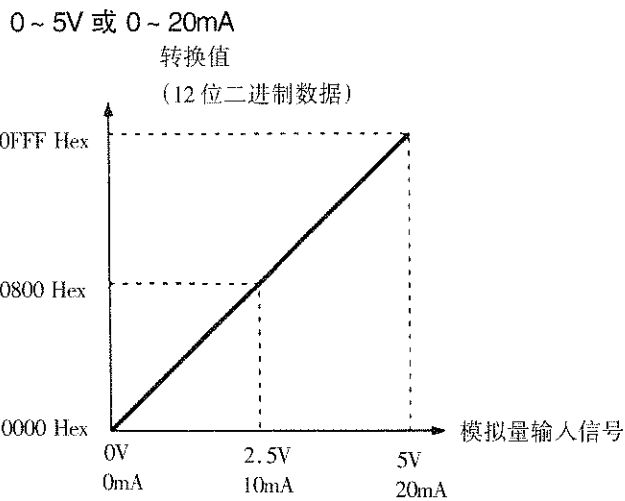
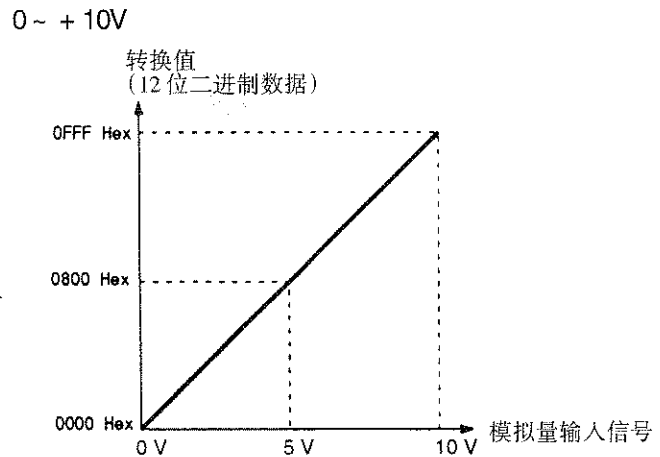
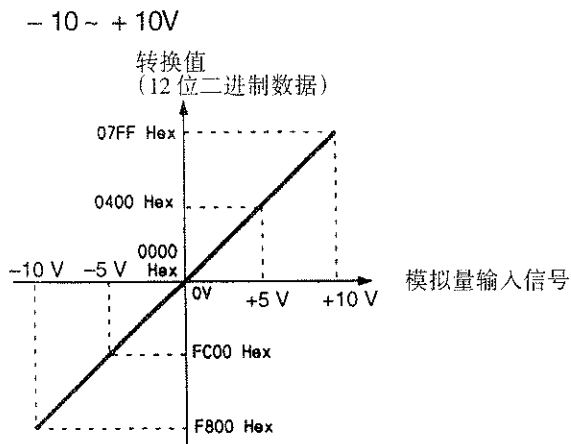


LED 指示灯

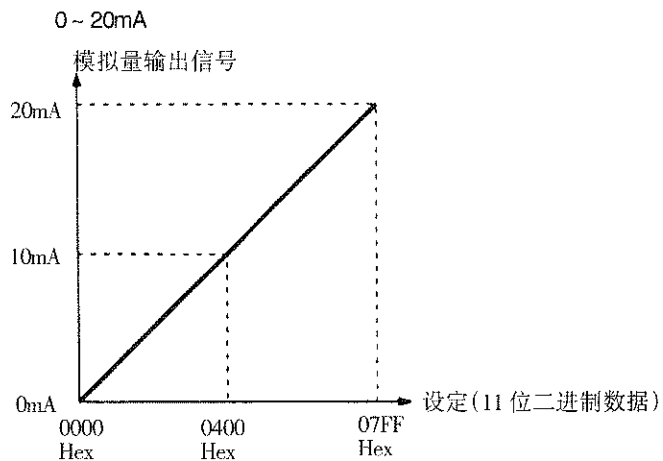
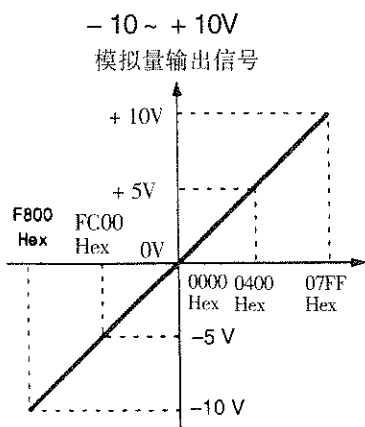


2-5-6 规格

模拟量输入:输入数据和转换值



模拟量输出:设定和输出数据



程序例

此板不使用特殊指令。MOV(21)用于读模拟量输入值和设定模拟量输出值。

相关位

槽 2 中内板使用的位

| 字 | 位 | 名称 | 功能 |
|-------|---------|-----------|--|
| IR232 | 00 ~ 15 | 模拟量输入1转换值 | 每个循环, 模拟量 I/O 板的每个输入的转换值存贮为4位数BCD。 - 10 ~ + 10V: F800 ~ 07FF Hex 0 ~ 10V: 0000 ~ 0FFF Hex 0 ~ 20mA: 0000 ~ 0FFF Hex |
| IR233 | 00 ~ 15 | 模拟量输入2转换值 | |
| IR234 | 00 ~ 15 | 模拟量输入3转换值 | |
| IR235 | 00 ~ 15 | 模拟量输入4转换值 | |
| IR236 | 00 ~ 15 | 模拟量输出1设定 | 模拟量 I/O 板的每个输出的设定存贮为4位数BCD。(每个循环读) - 10 ~ + 10V: F800 ~ 07FF Hex 0 ~ 20mA: 0000 ~ 07FF Hex |
| IR237 | 00 ~ 15 | 模拟量输出2设定 | |

SR 区标志

| 字 | 位 | 功能 |
|-------|----|--------|
| SR254 | 15 | 内板错误标志 |

AR 区标志

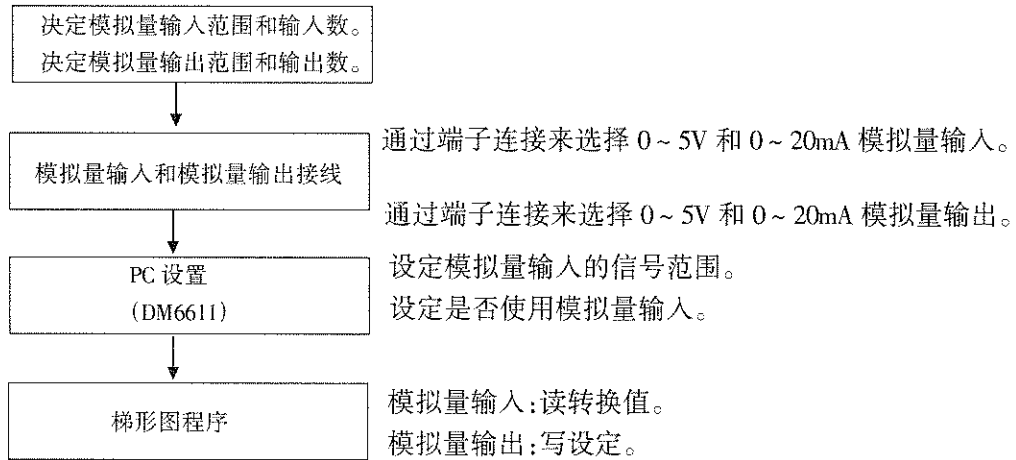
| 字 | 位 | 功能 |
|------|---------|--|
| AR04 | 08 ~ 15 | 槽2中的内板的错误代码 00Hex: 正常 01或02Hex: 硬件错误 03Hex: PC设置错误 04Hex: A/D或 D/A转换错误 |

有关的 PC 设置设定

| 字 | 位 | 功 | 能 | | |
|---------|------------|---------------------|--|------------|---|
| DM6611 | 00 ~ 07 | 00,01: 模拟量输入1输入信号范围 | 00: - 10 ~ + 10V 01: 0 ~ 10V 10: 0 ~ 5V 11: 未使用 (0 ~ 20mA由所连接的端子区别。) | | |
| | | 02,03: 模拟量输入2输入信号范围 | | | |
| | | 04,05: 模拟量输入3输入信号范围 | | | |
| | | 06,07: 模拟量输入4输入信号范围 | | | |
| | | 08 | | 模拟量输入1使用选择 | 指定每个端口使用或不使用A/ D转换。 0: 使用输入(转换) 1: 不使用输入(不转换) |
| | | 09 | | 模拟量输入2使用选择 | |
| 10 | 模拟量输入3使用选择 | | | | |
| 11 | 模拟量输入4使用选择 | | | | |
| 12 ~ 15 | 未使用(固定为0。) | | | | |

注 模拟量输出信号的电平由所连接的端子决定, 没有 PC 设置设定。

2-5-7 应用步骤



2-6 串行通信板

本节提供了串行通信板的介绍。详细的信息见串行通信板操作手册(W365)。

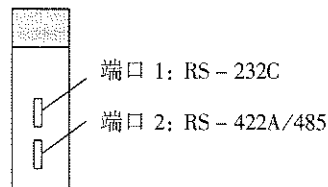
2-6-1 型号

| 名称 | 型号 | 规格 |
|-------|-------------|--------------------------------|
| 串行通信板 | CQM1H-SCB41 | 一个RS-232C端口 一个RS-422A/485端口 |

2-6-2 串行通信板

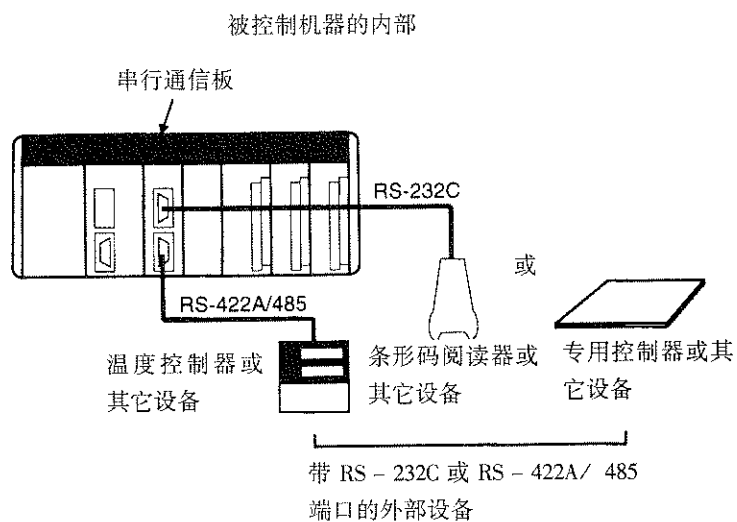
串行通信板是 CQM1H 系列 PC 的内板。CQM1H 系列 CPU 单元的内板槽 1 可安装一个板。此板不能安装在槽 2 中。

此板提供两个用于连接上位计算机,可编程终端(PT),通用外部设备,和编程设备(不包括编程器)的串行通信端口。这可以很容易地增加 CQM1H 系列 PC 的串行通信端口数。



2-6-3 性能

串行通信板是一种可选件,它可安装在 CPU 单元中,不使用 I/O 槽就增加了串口数。它支持协议宏(CPU 单元内置端口不支持协议宏),允许方便地连接到具有串口的通用设备。



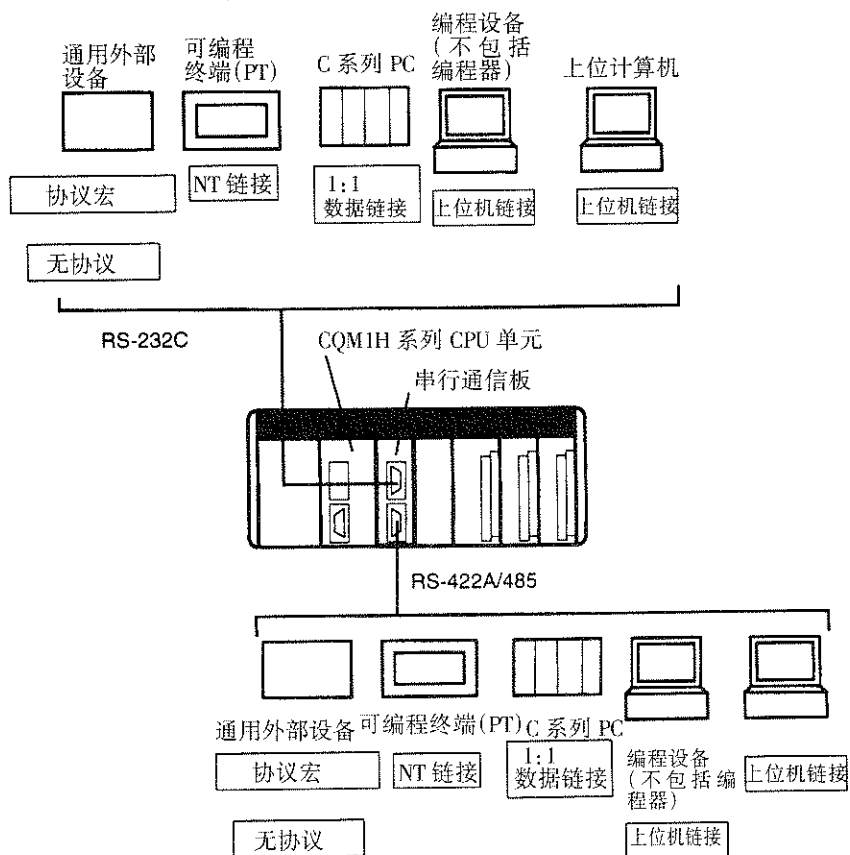
提供 RS - 232C 和 RS - 422A/485 端口。RS - 422A/485 端口允许不通过转换链接适配器与通用外部设备的 1:N 连接,1:N 连接可以在协议宏或 1:N 模式 NT 链接中使用。

2-6-4 系统配置

串行通信板支持下列串行通信模式：

上位机链接 (SYSMAC WAY), 协议宏, 无协议, 1:1 数据链接, 1:N 模式 NT 链接和 1:1 模式 NT 链接模式。下图所示的设备都可连接。

注 1:1 模式 NT 链接和 1:N 模式 NT 链接通信模式, 使用相互不兼容的不同协议。



注 NT-AL001-E 转换链接适配器可用于 RS-232C 和 RS-422A/485 之间的转换。此链接适配器需要 5V 电源。链接适配器连接到串行通信板上的 RS-232C 端口时, RS-232C 端口供电, 但是链接适配器连接到其它设备时必须单独供电。

第 3 章 内存区域

本章描述了 CQM1H PC 内存区的结构，并解释如何使用，同时描述了用于 CPU 单元和存储器盒之间传送数据的存储器盒操作。

| | | |
|--------|--------------------------------|-----|
| 3-1 | 内存区结构 | 136 |
| 3-2 | IR 区 | 138 |
| 3-2-1 | 输入和输出区 | 138 |
| 3-2-2 | 工作区 | 138 |
| 3-2-3 | I/O 分配 | 138 |
| 3-2-4 | 在插槽 1 上内板的标志/位 (IR200 ~ IR215) | 143 |
| 3-2-5 | 在插槽 2 上内板的标志/位 (IR232 ~ IR243) | 146 |
| 3-2-6 | 通信单元的标志/位 | 148 |
| 3-3 | SR 区 | 149 |
| 3-4 | TR 区 | 152 |
| 3-5 | HR 区 | 152 |
| 3-6 | AR 区 | 152 |
| 3-6-1 | 共用标志/位 (AR00 ~ AR04) | 153 |
| 3-6-2 | 内板的标志/位 (AR05 和 AR06) | 153 |
| 3-6-3 | 共用标志/位 (AR07 ~ AR27) | 155 |
| 3-6-4 | 使用时钟 | 159 |
| 3-7 | LR 区 | 160 |
| 3-8 | 定时器/计数器区 | 160 |
| 3-9 | DM 区 | 161 |
| 3-10 | EM 区 | 162 |
| 3-11 | 存储器盒用法 | 162 |
| 3-11-1 | 存储器盒和内容 | 162 |
| 3-11-2 | 存储器盒容量和程序大小 | 163 |
| 3-11-3 | 写入存储器盒 | 165 |
| 3-11-4 | 从存储器盒读 | 165 |
| 3-11-5 | 比较存储器盒内容 | 166 |

3-1 内存区结构

CQM1H 可使用下列内存区。

| 数据区 | | 大小 | 字 | 位 | 功能 |
|-------------------|-----|------------------|-------------------|-------------------|---|
| IR区(注1) | 输入区 | 256位 | IR000 ~ IR015 | IR00000 ~ IR01515 | 输入位可分配给输入单元或I/O单元, IR000中16位始终分配给内置在CPU单元上的输入点 |
| | 输出区 | 256位 | IR100 ~ IR115 | IR10000 ~ IR11515 | 输出位可分配给输出单元或I/O单元 |
| | 工作区 | 2,528位最小 (注2) | IR016 ~ IR089 | IR01600 ~ IR08915 | 工作位没有特别功能, 在程序中可随意使用。 |
| | | | IR116 ~ IR189 | IR11600 ~ IR18915 | |
| IR216 ~ IR219 | | | IR21600 ~ IR21915 | | |
| | | | IR224 ~ IR229 | IR22400 ~ IR22915 | |
| Controller Link位区 | | 96位 | IR090 ~ IR095 | IR09000 ~ IR09515 | 用来指示Controller Link数据链接状态的信息 (Controller Link单元没有安装时,可作为工作位使用) |
| | | 96位 | IR190 ~ IR195 | IR19000 ~ IR19515 | 用来指示Controller Link错误和网络参与信息 (Controller Link单元没有安装时,可作为工作位使用) |
| 宏操作数区(注1) | 输入区 | 64位 | IR096 ~ IR099 | IR09600 ~ IR09915 | 宏指令MCRO(99) 使用时使用; (宏指令不用时可作为工作位使用)。 |
| | 输出区 | 64位 | IR196 ~ IR199 | IR19600 ~ IR19915 | |
| 内板插槽1区 | | 256位 | IR200 ~ IR215 | IR20000 ~ IR21515 | 这些位分配给安装在CQM1H - CPU51/61插槽1上的内板 (当使用CQM1H - CPU11/CPU21或插槽1空时,可作为工作位使用)。 CQM1H - CTB41高速计数器板: IR200 ~ IR213(14字):由板使用 IR214和IR215(2字):不用 CQM1H - SCB41串行通信板: IR200 ~ IR207(8字):由板使用 IR208 ~ IR215(8字):不用 |
| 模拟设置区(注1) | | 64位 | IR220 ~ IR223 | IR22000 ~ IR22315 | CQM1H - AVB41模拟设定板安装时用于存放模拟设定 (模拟设定板不安装时,可作工作位使用)。 |
| 高速计数器0 PV(注1) | | 32位 | IR230 ~ IR231 | IR23000 ~ IR23115 | 用于存放内置高速计数器 (高速计数器0) 的当前值, (高速计数器0不用时可作工作位使用)。 |
| 内板插槽2区 | | 192位 | IR232 ~ IR243 | IR23200 ~ IR24315 | 这些位分配给安装在CQM1H - CPU51/61插槽2上的内板。(当使用CQM1H - CPU11/21或插模2空时,可作为工作位使用)。 CQM1H - CTB41高速计数器板: IR232 ~ IR243(12字):由板使用 CQM1H - PLB21脉冲I/O板 IR232 ~ IR239(8字)由板使用 IR240 ~ IR243(4字):不用 CQM1H - ABB21绝对型编码器接口板: IR232 ~ IR239(8字):由板使用 IR240 ~ IR243(4字):不用 CQM1H - MAB42模拟量I/O板: IR232 ~ IR239(8字):由板使用 IR240 ~ IR243(4字):不用 |

| 数据区 | | 大小 | 字 | 位 | 功能 |
|--------------|-----------|--------|--|-------------------|--|
| SR区 | | 184位 | SR244 ~ SR255 | SR24400 ~ SR25507 | 这些位具有特殊功能,如标志和控制位。 |
| HR区 | | 1,600位 | HR00 ~ HR99 | HR0000 ~ HR9915 | 这些位存储数据,并在掉电后保持它们的ON/OFF状态。 |
| AR区 | | 448位 | AR00 ~ AR27 | AR0000 ~ AR2715 | 这些位具有特殊功能,如标志和控制位。 |
| TR区 | | 8位 | --- | TR0 ~ TR7 | 这些位用于在程序分支中暂存ON/OFF状态。 |
| LR区(注1) | | 1,024位 | LR00 ~ LR63 | LR0000 ~ LR6315 | 通过RS232口或Controller Link单元的1:1数据链接时使用。 |
| 定时器/计数器区(注3) | | 512位 | TIM/CNT000 ~ TIM/CNT 511 (定时器/计数器号) | | 定时器和计数器使用相同的号,当使用TIMH(15)时,定时器号000~015可中断刷新,确保在长扫描周期中正确定时 |
| DM区 | 读/写 | 3,072字 | DM0000 ~ DM3071 | --- | DM区数据仅以字为单位存取,电源关掉后,数值保留。 |
| | | 3,072字 | DM3072 ~ DM6143 | --- | 仅在CQM1H-CPU 51/61 CPU单元中可用。 |
| | 只读(注4) | 425字 | DM6144 ~ DM6568 | --- | 不能用程序重写(仅可用编程装置)。 DM6400 ~ DM6409(10字): Controller Link DM参数区。 DM6450 ~ DM6499(50字) 路由表区。 DM6550 ~ DM6559(10字): 串行通信板设置。 |
| | 错误日志区(注4) | 31字 | DM6569 ~ DM6599 | --- | 用来存放错误产生时的错误发生时间和错误代码。 |
| | PC设置(注4) | 56字 | DM6600 ~ DM6655 | --- | 用来存放控制PC运行的各种参数。 |
| EM区 | | 6,144字 | EM0000 ~ EM6143 | --- | EM区数据仅以字为单位存取,电源关掉后数据保留,仅在CQM1H-CPU61 CPU单元可用。 |

- 注
1. IR 和 LR 位没有被分配功能使用时可作工作位使用。
 2. 至少 2528 位可用作工作位,其它位在没有被分配功能使用时,可用作工作位。因此可用工作位总数根据 PC 的配置。
 3. 当访问 PV, TIM/CNT 号作字数据使用,访问结束标志时,作位数据使用。
 4. DM6144 ~ DM6655 中的数据不能用程序重写。

3-2 IR 区

下面解释 IR 区的功能。

3-2-1 输入和输出区域

IR 区位分配给 I/O 单元和专用 I/O 单元的端子,它们反映输入和输出信号的 ON/OFF 状态,输入位从 IR00000 开始,输出位从 IR10000 开始。对于 CQM1H, 仅 IR00000 ~ IR01515 可用作输入位,IR10000 ~ IR11515 用作输出位。

注 输入位不能用于输出指令,不要对同一输出位使用多于一次的 OUT 和/或 OUT NOT 指令,否则程序不能正确执行。

3-2-2 工作区

工作位在程序中自由使用,但仅可在程序中使用,不能直接对外部 I/O,当 CQM1H 断电或执行开始或停止时,工作位复位(即变 OFF),下面表格显示部分 IR 区留出作工作区使用。

| 字 | 位 |
|--------------------|---------------------------|
| IR016 ~ IR089(74字) | IR01600 ~ IR08915(1,184位) |
| IR116 ~ IR189(74字) | IR11600 ~ IR18915(1,184位) |
| IR216 ~ IR219(4字) | IR21600 ~ IR21915(64位) |
| IR224 ~ IR229(6字) | IR22400 ~ IR22915(96位) |

下面所示范围内的位具有特殊功能,但在不使用它们的特殊功能时仍然用作工作位。

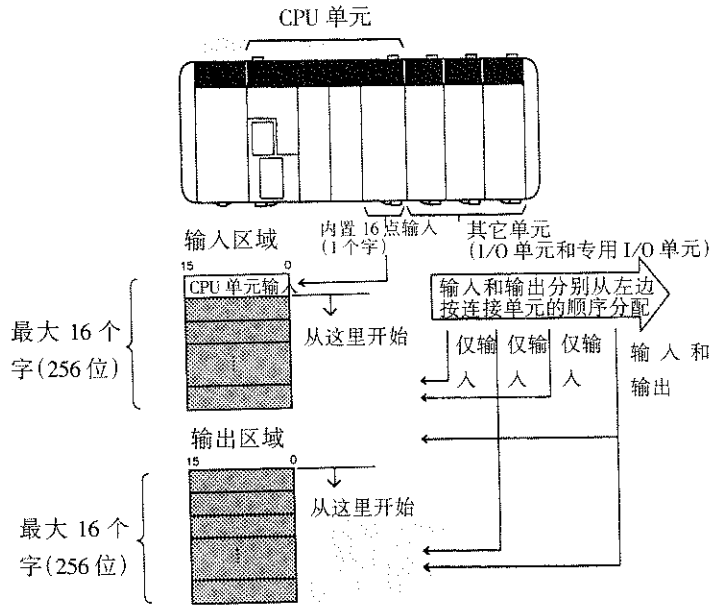
| 范 围 | 功 能 |
|---------------|--|
| IR001 ~ IR015 | 分配给输入单元时,这些位用作输入位。 |
| IR090 ~ IR095 | PC上安装Controller Link单元时,这些位指示数据链接的状态。 |
| IR096 ~ IR099 | 使用宏指令时,这些位用作操作数输入位。 |
| IR100 ~ IR115 | 分配给输出单元时,这些位用作输出位。 |
| IR190 ~ IR195 | PC上安装Controller Link单元时,这些位显示出错信息和网络节点号。 |
| IR196 ~ IR199 | 使用宏指令时,这些位用作操作数输出位。 |
| IR200 ~ IR215 | 这些位由安装在插槽1上的内板使用。 |
| IR220 ~ IR223 | 安装模拟设定板时,这些位用来存放模拟设定。 |
| IR230 ~ IR231 | 使用高速计数器0时,这些位用于存放当前值。 |
| IR232 ~ IR243 | 这些位由安装在插槽2上的内板使用。 |

3-2-3 I/O 分配

I/O 字从左边按次序分配给 I/O 单元和专用 I/O 单元,输入从 IR001 开始,输出从 IR100 开始,CPU 单元的 16 点输入分配给 IR000,I/O 位以一个字单元分配,即使 I/O 单元仅有 8 个位,也是这样。

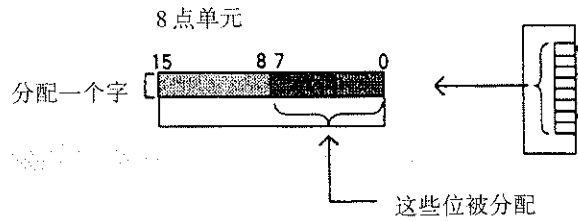
注 输入和输出位不能分配给内板或通信单元。

CQM1H PC 中没有 I/O 登记表, 因此不必从编程装置登记 I/O 表, 只要在 PC 上安装所用的单元, I/O 自动分配。



8 点 I/O 单元

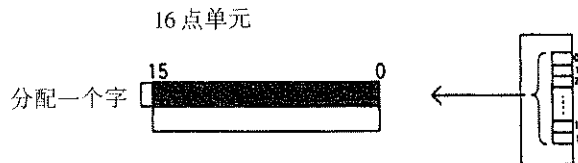
I/O 位以一个字为单位分配, 即使 I/O 单元只需要 8 个点。



没有使用的输入位 (08 ~ 15) 不能作为工作位使用, 但没有使用的输出位 (08 ~ 15) 可作为工作位使用。

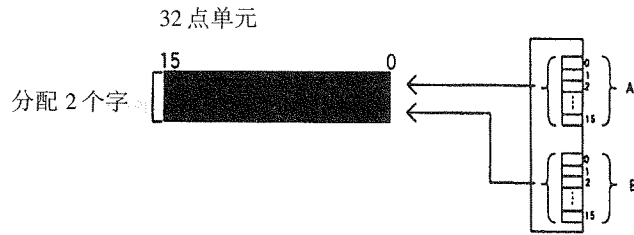
16 点 I/O 单元

每 16 点输入单元分配一个输入字, 每 16 点输出单元分配一个输出字, 输入和输出点 0 ~ 15 对应于分配字的 00 ~ 15 位。



32 点 I/O 单元

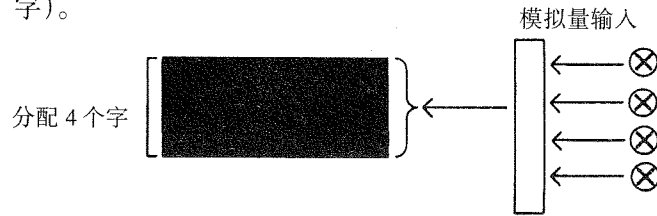
每个 32 点输入单元分配 2 个输入字, 每个 32 点输出单元分配 2 个输出字, 连接器 A 的 I/O 点 0 ~ 15 对应于第一个分配字 (n) 的 00 ~ 15 位, 连接器 B 的 I/O 点 0 ~ 15 对应于下一个分配字 (n+1) 的 00 ~ 15 位。



专用 I/O 单元

专用 I/O 单元需要预定输入位,输出位,或输入输出位的数量。在有些专用 I/O 单元中,所需字数根据单元的 DIP 开关设置。

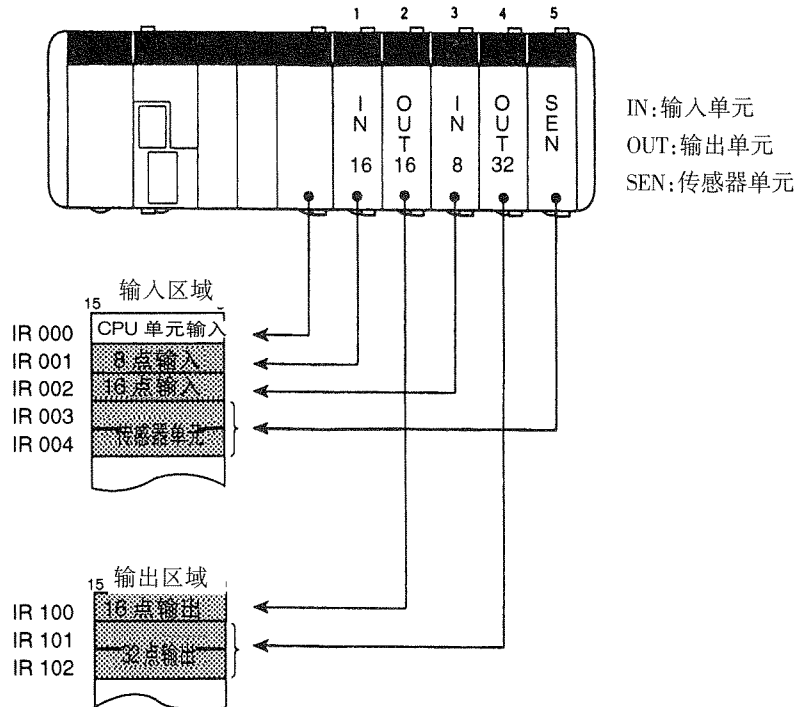
例如, CQM1-AD041 模拟量输入单元需要 4 个输入字或 2 个输入字。(模拟量输入单元在 4 路模拟量输入时需要 4 个输入字, 2 路输入时需要 2 个输入字)。



没有分配给单元的输入字和输出字可用作工作字。

I/O 分配举例

这个例子显示了一套 PC 的 I/O 分配, 有 2 块 DC 输入单元, 2 块晶体管输出单元, 和一块传感器单元。



| PC中的次序 | 单 元 | 规 格 | 字 数 | 分配的字 |
|--------|------------|---------|-------|-------------|
| 1 | CQM1-ID111 | 16点输入 | 1个输入字 | IR001 |
| 2 | CQM1-OD212 | 16点输出 | 1个输出字 | IR100 |
| 3 | CQM1-ID211 | 8点输入 | 1个输入字 | IR002 |
| 4 | CQM1-OD213 | 32点输出 | 2个输出字 | IR101和IR102 |
| 5 | CQM1-SEN01 | 1路传感器输入 | 2个输入字 | IR003和IR004 |

可分配的 I/O 点数要根据使用的 CQM1H CPU 单元,如下面表格所示,确认自动分配给 CPU 单元输入的一个输入字 (IR000) 包括在总数内,如果可分配的字数超出 CPU 单元的容量,会产生致命的 I/O 单元超出错误(错误代码 E1)。

| CPU单元 | 最大I/O位数 | 除CPU单元外可用的I/O字数 |
|-------------|--|--------------------|
| CQM1H-CPU61 | 512位 (256点输入, 256点输出), (32个字; 16个输入字和16个输出字) | 31(15个输入字, 16个输出字) |
| CQM1H-CPU51 | | |
| CQM1H-CPU21 | 256位 | 15 |
| CQM1H-CPU11 | | |

参考第 142 页表格显示的每个 I/O 单元占用多少个 I/O 字,第 143 页表格显示的每个专用 I/O 单元占用多少个 I/O 字。

AR22 显示已分配的输入字数和输出字数,如下所示。

| 字 | 位 | 功 能 | 数据范围 |
|------|---------|----------|-----------------|
| AR22 | 00 ~ 07 | 已分配的输入字数 | 00 ~ 16(2位数BCD) |
| | 08 ~ 15 | 已分配的输出字数 | 00 ~ 16(2位数BCD) |

CQM1H 没有底板,因此,分配 I/O 字时,不必处理空槽,自动分配最低可用的 I/O 字地址。

不管安装的输入单元和输出单元的次序如何,输入字自动分配给输入点,输出字自动分配给输出点。虽然 I/O 分配没什么影响,但建议输入单元安装在一起,输出单元也安装在一起,以便更容易理解分配字和消除噪声问题。

I/O 单元所需的 I/O 字数

| 名 称 | I/O点 | 型 号 | 输入字数(从IR001开始) | 输出字数(从IR100开始) |
|---------|------|------------|----------------|----------------|
| DC 输入单元 | 8 | CQM1-ID211 | 1 | --- |
| | 16 | CQM1-ID111 | 1 | |
| | | CQM1-ID212 | $\frac{1}{1}$ | |
| | 32 | CQM1-ID112 | 2 | |
| | | CQM1-ID213 | 2 | |
| | | CQM1-ID214 | 2 | |
| AC 输入单元 | 8 | CQM1-IA121 | 1 | |
| | | CQM1-IA221 | 1 | |
| 继电器输出单元 | 8 | CQM1-OC221 | --- | 1 |
| | 16 | CQM1-OC222 | | 1 |
| | | CQM1-OC224 | | 1 |
| 晶体管输出单元 | 8 | CQM1-OD211 | --- | 1 |
| | 16 | CQM1-OD212 | | 1 |
| | 32 | CQM1-OD213 | | 2 |
| | | CQM1-OD216 | | 2 |
| | 16 | CQM1-OD214 | | 1 |
| | 8 | CQM1-OD215 | | 1 |
| AC 输出单元 | 8 | CQM1-OA221 | --- | 1 |
| | 16 | CQM1-OA222 | | 1 |

专用 I/O 单元所需的 I/O 字数

| 名称 | 型 号 | 输入字数(从IR001开始) | 输出字数(从IR100开始) |
|--------------------|---------------|----------------|----------------|
| 模拟量输入单元 | CQM1-AD041 | 2或4 | --- |
| 模拟量输出单元 | CQM1-DA021 | --- | 2 |
| 电源供给单元 | CQM1-IPS01 | --- | --- |
| | CQM1-IPS02 | --- | --- |
| B7A接口单元 | CQM1-B7A02 | --- | 1 |
| | CQM1-B7A12 | 1 | --- |
| | CQM1-B7A03 | --- | 2 |
| | CQM1-B7A13 | 2 | --- |
| | CQM1-B7A21 | 1 | 1 |
| G730接口单元 | CQM1-G7M21 | 2或1 | 2或1 |
| | CQM1-G7N11 | 2或1 | --- |
| | CQM1-G7N01 | --- | 2或1 |
| I/O链接单元 | CQM1-LK501 | 2 | 2 |
| 传感器单元 | CQM1-SEN01 | 1(见注) | --- |
| 光纤光电模块 | E3X-MA11 | 1 | --- |
| 带独立放大器光电模块 | E3C-MA11 | 1 | |
| 带独立放大器接近模块 | E2C-MA11 | 1 | |
| 虚拟模块 | E39-M11 | 1 | |
| 远程控制器 | CQM1-TU001 | --- | |
| 温度控制单元 | CQM1-TC001 | 2或1 | 2或1 |
| | CQM1-TC002 | | |
| | CQM1-TC101 | | |
| | CQM1-TC102 | | |
| 线性传感器接口单元 | CQM1-LSE01 | 1 | 1 |
| | CQM1-LSE02 | 1 | 1 |
| CompoBus/s主单元 | CQM1-SRM21-V1 | 4,2或1 | 4,2或1 |
| CompoBus/D I/O链接单元 | CQM1-DRT21 | 1 | 1 |

注 当安装下面 4 个模块 (E3X-MA11, E3C-MA11, E2C-MA11 和 E39-M11) 时,总共需要 5 个字。

3-2-4 插槽 1 上内板的标志/位(IR200 到 IR215)

串行通信板标志/位

| 字 | 位 | 功 能 | 通信模式 |
|-------|-------|----------------|------|
| IR200 | 00 | 串行通信板硬件错误标志 | 所有模式 |
| | 01 | 端口识别错误标志(硬件错误) | |
| | 02 | 协议数据错误标志 | |
| | 03~10 | 未使用 | 协议宏 |
| | 11 | 端口2协议宏执行错误标志 | |
| | 12 | 端口1协议宏执行错误标志 | 所有模式 |
| | 13 | 端口2PC设置错误标志 | |
| | 14 | 端口1PC设置错误标志 | |
| | 15 | PC设置错误标志 | |

| 字 | 位 | 功 能 | | 通信模式 |
|-------|---------|------|---|-------------|
| IR201 | 00 ~ 03 | 端口 1 | 错误代码 0: 正常执行 1: 校验位错误 2: 帧错误 3: 过运行错误 4: FCS错误 5: 超时错误 6: 校验和错误 7: 命令错误 | 所有模式 |
| | 04 | | 通信错误标志 | |
| | 05 | | 传输允许标志 | |
| | 06 | | 接收完成标志 | |
| | 07 | | 接收溢出标志 | |
| | 08 ~ 11 | 端口 2 | 错误代码 0: 正常工作 1: 校验位错误 2: 帧错误 3: 过运行错误 4: FCS错误 5: 超时错误 6: 校验和错误 7: 命令错误 | 所有模式 |
| | 12 | | 通信错误标志 | |
| | 13 | | 传输允许标志 | |
| | 14 | | 接收完成标志 | |
| | 15 | | 接收溢出标志 | |
| IR202 | 00 ~ 07 | 端口 1 | 与PT通信标志(位00 ~ 07 = PT 0 ~ 7) | NT链接(1:N模式) |
| | 00 ~ 15 | | 重复计数器PV(十六进制00 ~ FF) | 协议宏 |
| | | | 接收计数器(4位数BCD) | 无协议 |
| IR203 | 00 ~ 07 | 端口 2 | 与PT通信标志(位00 ~ 07 = PT 0 ~ 7) | NT链接(1:N模式) |
| | 00 ~ 15 | | 重复计数器PV(十六进制00 ~ FF) | 协议宏 |
| | | | 接收计数器(4位数BCD) | 无协议 |
| IR204 | 00 | 端口 1 | 跟踪标志 | 协议宏 |
| | 01 | 端口 2 | | |
| | 02 ~ 07 | 未使用 | | |
| | 08 ~ 11 | 端口 1 | 协议宏错误代码 0: 正常工作 1: 无协议宏功能 2: 顺序号错误 3: 接收数据/写区域溢出 | |
| | 12 ~ 15 | 端口 2 | 4: 协议数据语法错误 5: 端口初始化时执行协议宏 | |
| IR205 | 00 ~ 03 | 端口 1 | 完成接收实例号 | 协议宏 |
| | 04 ~ 07 | | 完成步号 | |
| | 08 ~ 14 | | 未使用 | |
| | 15 | | IR20408 ~ IR20411数据存储标志 0: 无数据存储, 1: 数据存储 | |
| IR206 | 00 ~ 03 | 端口 2 | 完成接收实例号 | 协议宏 |
| | 04 ~ 07 | | 完成步号 | |
| | 08 ~ 14 | | 未使用 | |
| | 15 | | IR20412 ~ IR20415数据存储标志 0: 无数据存储, 1: 数据存储 | |

| 字 | 位 | 功 能 | | 通信模式 |
|---------------------|-------|-----|------------|---------|
| IR207 | 00 | 端口1 | 串行通信口重启动位 | 所有模式 |
| | 01 | 端口2 | | |
| | 02 | 端口1 | 连续跟踪开始/停止位 | 协议宏 |
| | 03 | 端口2 | | |
| | 04 | 端口1 | | |
| | 05 | 端口2 | 断点跟踪开始/停止位 | |
| | 06~07 | 未使用 | | |
| | 08 | 端口1 | 协议宏执行标志 | |
| | 09 | | 步错误进程标志 | |
| | 10 | | 顺序结束完成标志 | 协议宏 |
| | 11 | | 强制退出位 | |
| | 12 | 端口2 | 协议宏执行标志 | 无协议或协议宏 |
| | 13 | | 步错误进程标志 | |
| | 14 | | 顺序结束完成标志 | 协议宏 |
| | 15 | | 强制退出位 | |
| IR208 ~ IR215 | 00~15 | 未使用 | | --- |

高速计数器板标志/位

| 字 | 位 | 名 称 | | 功 能 |
|----------|-------|------------------|-----------|--|
| IR200 | 00~15 | 高速计数器1 | PV(最右4位数) | 包含每个高速计数器板端口的高速计数器PV值。 注 PV数据格式(BCD或十六进制)可在PC设置(DM6602)中设置。 |
| IR201 | 00~15 | | PV(最左4位数) | |
| IR202 | 00~15 | 高速计数器2 | PV(最右4位数) | |
| IR203 | 00~15 | | PV(最左4位数) | |
| IR204 | 00~15 | 高速计数器3 | PV(最右4位数) | |
| IR205 | 00~15 | | PV(最左4位数) | |
| IR206 | 00~15 | 高速计数器4 | PV(最右4位数) | |
| IR207 | 00~15 | | PV(最左4位数) | |
| IR208 | 00~07 | 比较结果:内部输出位 | | 条件满足时包含CTBL(一)中的操作数指定的位型式。 |
| (高速计数器1) | 08~11 | 比较结果:输出1~4的外部输出位 | | 条件满足时包含CTBL(一)中的操作数指定的位型式。 |
| IR209 | 12 | 计数器工作标志 | | 0:停止 1:工作 |
| (高速计数器2) | 13 | 比较标志 | | 显示比较是否正在进行。 0:停止,1:工作 |
| IR210 | 14 | PV上溢/下溢标志 | | 0:正常 1:上溢或下溢产生 |
| (高速计数器3) | 15 | SV错误标志 | | 0:正常 1:SV错误产生 |
| IR211 | | | | |
| (高速计数器4) | | | | |

| 字 | 位 | 名称 | 功能 | |
|-------|-------|-------------|--------------------------------------|------------|
| IR212 | 00 | 高速计数器1复位位 | Z相和软件复位 0: Z相不复位计数器 1: Z相复位计数器 | |
| | 01 | 高速计数器2复位位 | | |
| | 02 | 高速计数器3复位位 | 仅软件复位 | |
| | 03 | 高速计数器4复位位 | 0: 计数器不复位 0→1: 计数器复位 | |
| | 04~07 | 未使用 | | |
| | 08 | 高速计数器1比较停止位 | 0→1: 开始比较 | |
| | 09 | 高速计数器2比较停止位 | 1→0: 停止比较 | |
| | 10 | 高速计数器3比较停止位 | | |
| | 11 | 高速计数器4比较停止位 | | |
| | 12 | 高速计数器1停止位 | 0: 继续工作 | |
| | 13 | 高速计数器2停止位 | 1: 停止工作 | |
| | 14 | 高速计数器3停止位 | | |
| | 15 | 高速计数器4停止位 | | |
| | IR213 | 00 | 外部输出1强制置位位 | 0: 不影响输出状态 |
| | | 01 | 外部输出2强制置位位 | 1: 强制输出ON |
| 02 | | 外部输出3强制置位位 | | |
| 03 | | 外部输出4强制置位位 | | |
| 04 | | 外部输出强制置位允许位 | 1: 输出1~4强制置位允许 0: 输出1~4强制置位禁止 | |
| 05~15 | | 未使用 | | |

模拟设定板(插槽 1 和 2)标志/位

| 字 | 位 | 功能 |
|-------|-------|--------------------------|
| IR220 | 00~15 | 模拟SV1: 0000~0200(4位数BCD) |
| IR221 | 00~15 | 模拟SV2: 0000~0200(4位数BCD) |
| IR222 | 00~15 | 模拟SV3: 0000~0200(4位数BCD) |
| IR223 | 00~15 | 模拟SV4: 0000~0200(4位数BCD) |

3-2-5 插槽 2 上内板的标志/位(IR232~IR243)

高速计数器板标志/位

| 字 | 位 | 名称 | 功能 |
|--|-------|-------------------|---|
| IR232 | 00~15 | 高速计数器1 | 包括每个高速计数器板端口的高速计数器PV。 注 PV数据格式(BCD或十六进制)可在PC设置(DM6602)中设置。 |
| IR233 | | PV(最右4位数) | |
| IR234 | 00~15 | 高速计数器2 | |
| IR235 | | PV(最左4位数) | |
| IR236 | 00~15 | 高速计数器3 | |
| IR237 | | PV(最右4位数) | |
| IR238 | 00~15 | 高速计数器4 | |
| IR239 | | PV(最左4位数) | |
| IR240 (高速计数器1) IR241 (高速计数器2) IR242 (高速计数器3) IR243 (高速计数器4) | 00~07 | 比较结果: 内部输出位 | 条件满足时包含CTBL(一)中的操作数指定的位型式 |
| | 08~11 | 比较结果: 输出1~4的外部输出位 | 条件满足时包含CTBL(一)中的操作数指定的位型式 |
| | 12 | 计数器工作标志 | 0: 停止 1: 工作 |
| | 13 | 比较标志 | 指示比较是否正在进行 0: 停止, 1: 工作 |
| | 14 | PV上溢/下溢标志 | 0: 正常 1: 上溢或下溢产生 |
| | 15 | SV错误标志 | 0: 正常 1: SV错误产生 |

| 字 | 位 | 名 称 | 功 能 | |
|-------|-------|-------------|--------------------------------|-----------------------|
| AR05 | 00 | 高速计数器1复位位 | Z相和软件复位 0:Z相复位禁止 | |
| | 01 | 高速计数器2复位位 | 1:Z相复位允许 | |
| | 02 | 高速计数器3复位位 | 仅软件复位 | |
| | 03 | 高速计数器4复位位 | 0:软件复位禁止 0→1:执行软件复位 | |
| | 04~07 | 未使用 | | |
| | 08 | 高速计数器1比较停止位 | 0→1:开始比较 1→0:停止比较 | |
| | 09 | 高速计数器2比较停止位 | | |
| | 10 | 高速计数器3比较停止位 | | |
| | 11 | 高速计数器4比较停止位 | | |
| | 12 | 高速计数器1停止位 | 0:继续工作 1:停止工作 | |
| | 13 | 高速计数器2停止位 | | |
| | 14 | 高速计数器3停止位 | | |
| | 15 | 高速计数器4停止位 | | |
| | AR06 | 00 | 外部输出1强制置位位 | 0:不影响输出状态 1:强制输出ON |
| | | 01 | 外部输出2强制置位位 | |
| 02 | | 外部输出3强制置位位 | | |
| 03 | | 外部输出4强制置位位 | | |
| 04 | | 外部输出强制置位允许位 | 1:输出1~4强制置位允许 0:输出1~4强制置位禁止 | |
| 05~15 | | 未使用 | | |

脉冲 I/O 板标志/位

| 字 | 位 | 功 能 |
|-------------|-------|------------------|
| IR232 | 00~15 | 高速计数器1 PV(最右4位数) |
| IR233 | 00~15 | 高速计数器1 PV(最左4位数) |
| IR234 | 00~15 | 高速计数器2 PV(最右4位数) |
| IR235 | 00~15 | 高速计数器2 PV(最左4位数) |
| IR236 | 00~15 | 端口1脉冲输出PV(最右4位数) |
| IR237 | 00~15 | 端口1脉冲输出PV(最左4位数) |
| IR238 | 00~15 | 端口2脉冲输出PV(最右4位数) |
| IR239 | 00~15 | 端口2脉冲输出PV(最左4位数) |
| IR240~IR243 | 00~15 | 未使用 |

绝对值编码器接口板标志/位

| 字 | 位 | 功 能 |
|-------------|-------|------------------------|
| IR232 | 00~15 | 绝对值编码器高速计数器1 PV(最右4位数) |
| IR233 | 00~15 | 绝对值编码器高速计数器1 PV(最左4位数) |
| IR234 | 00~15 | 绝对值编码器高速计数器2 PV(最右4位数) |
| IR235 | 00~15 | 绝对值编码器高速计数器2 PV(最左4位数) |
| IR236~IR243 | 00~15 | 未使用 |

模拟量 I/O 板标志/位

| 字 | 位 | 功 能 |
|-------------|-------|-----------|
| IR232 | 00~15 | 模拟量输入1转换值 |
| IR233 | 00~15 | 模拟量输入2转换值 |
| IR234 | 00~15 | 模拟量输入3转换值 |
| IR235 | 00~15 | 模拟量输入4转换值 |
| IR236 | 00~15 | 模拟量输出1 SV |
| IR237 | 00~15 | 模拟量输出2 SV |
| IR238~IR243 | 00~15 | 未使用 |

模拟设置板(插槽 1 和 2)标志/位

| 字 | 位 | 功 能 |
|-------|---------|---------------------------|
| IR220 | 00 ~ 15 | 模拟SV1:0000 ~ 0200(4位数BCD) |
| IR221 | 00 ~ 15 | 模拟SV2:0000 ~ 0200(4位数BCD) |
| IR222 | 00 ~ 15 | 模拟SV3:0000 ~ 0200(4位数BCD) |
| IR223 | 00 ~ 15 | 模拟SV4:0000 ~ 0200(4位数BCD) |

3-2-6 通信单元的标志/位

Controller Link 状态区

域 1(IR090 ~ IR095)

| 字 | 位 | 功 能 |
|-------|---------|---|
| IR090 | 00 ~ 14 | 始终为0 |
| | 15 | 本地节点数据链接参与状态 0:本地节点不在数据链接上或数据链接已停止 1:本地节点参与数据链接 |
| IR091 | 00 ~ 07 | 数据链接状态:节点1 |
| | 08 ~ 15 | 数据链接状态:节点2 |
| IR092 | 00 ~ 07 | 数据链接状态:节点3 |
| | 08 ~ 15 | 数据链接状态:节点4 |
| IR093 | 00 ~ 07 | 数据链接状态:节点5 |
| | 08 ~ 15 | 数据链接状态:节点6 |
| IR094 | 00 ~ 15 | 未使用 |
| IR095 | 00 ~ 10 | 始终为0 |
| | 11 | 终端状态 0:终端电阻开关OFF 1:终端电阻开关ON |
| | 12 ~ 15 | 始终为0 |

Controller Link 状态区

域 2(IR190 ~ IR195)

| 字 | 位 | 功 能 |
|-------------|---------|--------------------------------|
| IR190 | 00 | 网络参数错误标志 1:发生错误,0:没有错误 |
| | 01 | 数据链接表错误标志 1:发生错误,0:没有错误 |
| | 02 | 路由表错误标志 1:发生错误,0:没有错误 |
| | 03 ~ 06 | 始终为0 |
| | 07 | EEPROM写错误标志 1:发生错误,0:没有错误 |
| | 08 | 始终为0 |
| | 09 | 节点号重复错误标志 1:发生错误,0:没有错误 |
| | 10 | 网络参数不匹配错误标志 1:发生错误,0:没有错误 |
| | 11 | 通信控制器发送器错误标志 1:发生错误,0:没有错误 |
| | 12 | 通信控制器硬件错误标志 1:发生错误,0:没有错误 |
| | 13和14 | 始终为0 |
| | 15 | 错误日志标志 1:错误记录已记录,0:无错误记录被记录 |
| IR191 | 00 ~ 07 | 轮询节点的节点号 |
| | 08 ~ 15 | 开始节点的节点号 |
| IR192和IR193 | 00 ~ 15 | 网络参与状态 1:参与网络,0:不参与网络 |
| IR194和IR195 | 00 ~ 15 | 未使用 |

3-3 SR 区域

这些位主要用于有关 CQM1H 工作的标志, 下面表格提供了各个位功能的详细情况。

SR244 ~ SR247 在计数器模式中不使用输入中断时, 可作工作位使用。

| 字 | 位 | 功能 | 页数 |
|-------|---------|--|----|
| SR244 | 00 ~ 15 | 输入中断0计数器模式SV 输入中断0用于计数器模式时的SV(4位数十六进制, 0000 ~ FFFF)。(输入中断0未在计数器模式下使用时, 可作为工作位使用) | 24 |
| SR245 | 00 ~ 15 | 输入中断1计数器模式SV 输入中断1用于计数器模式时的SV(4位数十六进制, 0000 ~ FFFF)。(输入中断1未在计数器模式下使用时, 可作为工作位使用)。 | |
| SR246 | 00 ~ 15 | 输入中断2计数器模式SV 输入中断2用于计数器模式时的SV(4位数十六进制, 0000 ~ FFFF)。(输入中断2未在计数器模式下使用时, 可作为工作位使用)。 | |
| SR247 | 00 ~ 15 | 输入中断3计数器模式SV 输入中断3用于计数器模式时的SV(4位数十六进制, 0000 ~ FFFF)。(输入中断3未在计数器模式下使用时, 可作为工作位使用)。 | |
| SR248 | 00 ~ 15 | 输入中断0计数器模式PV减1 输入中断0用于计数器模式时的计数器PV - 1(4位数十六进制) | 24 |
| SR249 | 00 ~ 15 | 输入中断1计数器模式PV减1 输入中断1用于计数器模式时的计数器PV - 1(4位数十六进制) | |
| SR250 | 00 ~ 15 | 输入中断2计数器模式PV减1 输入中断2用于计数器模式时的计数器PV - 1(4位数十六进制) | |
| SR251 | 00 ~ 15 | 输入中断3计数器模式PV减1 输入中断3用于计数器模式时的计数器PV - 1(4位数十六进制) | |

| 字 | 位 | 功 能 | 页数 | |
|--------|--------|---|--|-----|
| SR 252 | 00 | 高速计数器0复位位 | 30 | |
| | 01 | 插槽 2 内板控制位 脉冲/I/O板:高速计数器1复位位 置ON复位高速计数器1(端口1)的PV。 绝对编码器接口板: 绝对高速计数器1原点补偿位 置ON设置绝对高速计数器1(端口1)原点补偿,补偿值设置在DM6611后,自动变OFF。 | 137 | |
| | 02 | 插槽 2 内板控制位 脉冲I/O板:高速计数器2复位位 置ON复位高速计数器2(端口2)的PV。 绝对编码器接口板: 绝对高速计数器2原点补偿位 置ON设置绝对高速计数器2(端口2)原点补偿,补偿值设置在DM6612后自动变OFF。 | 137 | |
| | 03~07 | 未使用。 | | |
| | 08 | 外围端口复位位 置ON复位外围端口,(连接编程装置时不可用)复位完成后自动变OFF。 | 45 | |
| | 09 | RS-232C口复位位 置ON复位RS-232C口,复位结束后自动变OFF。 | | |
| | 10 | PC设置复位位 置ON初始化PC设置(DM 6600~DM 6655)。复位结束后自动变OFF,仅在PC是编程模式下有效。 | 2 | |
| | 11 | 强制状态保持位 OFF:被强制置位/复位的位在从编程模式切换到监视模式时被清除。 ON:被强制置位/复位的位在从编程模式切换到监视模式时保持。 | 12 | |
| | 12 | I/O保持位 OFF:开始或停止运行时,IR和LR位复位。 ON:开始或停止运行时,IR和LR位状态保持。 | 12 | |
| | 13 | 未使用。 | | |
| | 14 | 错误日志复位位 置ON清除错误日志,操作完成后自动变OFF。 | 493 | |
| | 15 | 输出OFF位 OFF:正常输出状态。 ON:全部输出变OFF。 | 152 | |
| | SR 253 | 00~07 | FAL错误代码 当错误产生时,错误代码(2位数)存放在这里,执行FAL(06)或FALS(07)时,FAL号存放在这里,执行FAL00指令或用编程装置清除错误去复位这个字节(变成00)。 | 221 |
| | | 08 | 电池电压低标志 CPU单元电池电压低时变ON。 | 490 |
| | | 09 | 循环时间溢出标志 循环时间溢出产生时(也就是循环时间超出100ms)变ON。 | 490 |
| 10~12 | | 未使用。 | | |
| 13 | | 始终ON标志 | ... | |
| 14 | | 始终OFF标志 | ... | |
| 15 | | 第一次循环标志 运行开始第一个循环内变ON。 | ... | |

| 字 | 位 | 功 能 | 页数 |
|--------|--------|--|----------------------------|
| SR 254 | 00 | 一分钟时钟脉冲(30秒ON;30秒OFF) | ... |
| | 01 | 0.02秒时钟脉冲(0.01秒ON;0.01秒OFF) | ... |
| | 02~03 | 未使用。 | |
| | 04 | 上溢(OV)标志 计算结果超出有符号二进制数据上限时变ON。 | 316 |
| | 05 | 下溢(UF)标志 计算结果低于有符号二进制数据下限时变ON。 | 316 |
| | 06 | 微分监视器结束标志 微分监视结束后变ON。 | 137 |
| | 07 | STEP(08)执行标志 STEP(08)进程开始一个循环内变ON。 | 222 |
| | 08 | HKY(-)执行标志 执行HKY(-)时变ON。 | 419 |
| | 09 | 7SEG(88)执行标志 执行7SEG(88)时变ON。 | 412 |
| | 10 | DSW(87)执行标志 执行DSW(87)时变ON。 | 415 |
| | 11~12 | 未使用。 | |
| | 13 | 通信单元错误标志 通信单元产生错误时变ON,这个标志反映了通信单元错误标志(AR 0011)的操作。 | 415 |
| | 14 | 未使用。 | |
| | 15 | 内板错误标志 安装在插槽1或2的内板产生错误时变ON,插槽1的错误代码存放在AR 0400~AR 0407,插槽2的错误代码存放在AR 0408~AR 0415。 | ... |
| | SR 255 | 00 | 0.1秒时钟脉冲(0.05秒ON,0.05秒OFF) |
| 01 | | 0.2秒时钟脉冲(0.1秒ON,0.1秒OFF) | ... |
| 02 | | 1秒时钟脉冲(0.5秒ON,0.5秒OFF) | ... |
| 03 | | 指令执行错误(ER)标志 执行一条指令产生错误时变ON。 | ... |
| 04 | | 进位(CY)标志 指令执行结果有进位时变ON。 | ... |
| 05 | | 大于(GR)标志 比较结果大于时变ON。 | ... |
| 06 | | 等于(EQ)标志 比较结果等于或指令执行结果为0时变ON。 | ... |
| 07 | | 小于(LE)标志 比较结果小于时变ON。 | ... |

SR 25211
(强制状态保持位)

强制置位/复位状态清除后,被强制的位会如下变 ON 或 OFF:

强制置位清除后:位变 ON

强制复位清除后:位变 OFF

在 PC 切换到运行模式时,所有强制置位或强制复位位被清除,除非 PC 设置中 DM6601 设置为在电源上电后保持强制状态保持位的当前状态,这个设置能用于防止强制状态在上电时被清除。

用编程装置使该位变 ON 和 OFF。

SR 25212
(I/O 保持位)

该位 ON 时,IR 和 LR 区域位的状态在 PC 从编程模式切换到运行或监视模式时保持。(如果 I/O 保持位 OFF,所有 IR 和 LR 位在 PC 开始运行时被复位。)

用编程装置使该位变 ON 和 OFF。

在 PC 设置中 DM6601 可设置为在电源上电时保持 I/O 保持位的当前状态, 当进行这个设置和 I/O 保持位为 ON 时, IR 和 LR 区域中的位状态在电源上电时不会被清除。

SR 25215
(输出 OFF 位)

该位变 ON 时, 所有输出变 OFF, CPU 单元的 INH 指示灯会亮, 只要输出 OFF 位为 ON, 即使输出位因程序会变 ON, 输出也会保持 OFF。

从晶体管输出单元和脉冲 I/O 板的脉冲输出只要输出 OFF 位为 ON 时, 会保持 OFF, 如果安装了高速计数器板, 只要输出 OFF 位为 ON, 板的外部输出 (1~4) 会保持 OFF。

输出 OFF 位正常时为 OFF, 通过程序正常时使该位变 OFF, 如果不通过程序使输出 OFF 位变 OFF, 在电源关掉后, 该位 ON/OFF 状态会保持。(如果后备电池失效, 该位状态可能不保持。)

SR 25308
(电池电压低标志)

在 PC 设置 (DM6655) 中可以设置以便这些错误不会产生。

SR 25309
(循环时间溢出标志)

可以在 PC 设置 (DM6655) 中加以设置以便这些错误不会发生。

3—4 TR 区


当复杂的梯形图不能在助记符方式下编程时, 这些位可以临时存放程序分支的 ON/OFF 执行条件, 仅可以在助记符方式下使用, 直接用梯形图编程时, TR 位会自动处理。

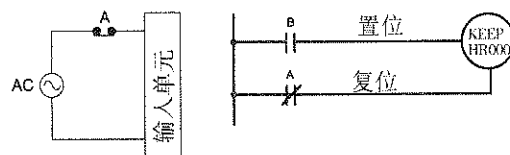
在同一程序块中, 相同的 TR 位不能多次使用, 但在不同程序块中, 可再次使用, TR 位的 ON/OFF 状态不能用编程装置监视。

在第 185 页上举例 TR 位在程序中的用法。

3—5 HR 区

这些位在 CQMIH 电源关掉后或运行开始或停止仍然保持自己的 ON/OFF 状态, 用法和工作位一样。

 **注意** 输入装置使用 AC 电源供电时 (如下图), KEEP (I1) 的复位 (R) 不要使用 NC 条件的输入位, PC 的 DC 电源关断相对输入设备的 AC 电源有一定的延迟, 会引起 KEEP (I1) 指定位被复位。



3—6 AR 区

这些位主要用于有关 CQMIH 工作的标志, AR05 和 AR06 中有关内板操作和功能的标志因每块内板而不同, 下面表格列出了共用标志的功能 (AR00~AR04 和 AR07~AR27) 和特定内板唯一的标志 (AR05 和 AR06)

AR23 (掉电计数器) 是例外, AR 字和位的状态每次循环刷新一次。(AR23 只有电源中断时才刷新。)

3-6-1 共用标志/位 (AR00 ~ AR04)

| 字 | 位 | 功 能 |
|------|---------|--|
| AR00 | 00 ~ 10 | 未使用 |
| | 11 | 通信单元错误标志 通信单元中产生错误时变ON。 |
| | 12 ~ 15 | 未使用 |
| AR01 | 00 ~ 10 | 未使用 |
| | 11 | 通信单元重启动位 使该位ON然后OFF,以重新启动通信单元。 |
| | 12 ~ 15 | 未使用 |
| AR02 | 00 ~ 07 | 网络指令完成代码 包含网络指令 (SEND(90), RECV(98), 或 CMND(—)) 的完成代码。 |
| | 08 | 网络指令 (SEND(90), RECV(98), 或 CMND(—)) 错误标志 在执行网络指令 (SEND(90), RECV(98), 或 CMND(—)) 产生错误时变ON。 |
| | 09 | 网络指令 (SEND(90), RECV(98), 或 CMND(—)) 允许标志 网络指令 (SEND(90), RECV(98), 或 CMND(—)) 可以执行时变ON。 |
| | 10 ~ 14 | 未使用 |
| | 15 | 通信单元连接标志 通信单元安装在PC上时变ON。 |
| AR03 | 00 ~ 15 | 通信单元服务时间 显示最近一次循环的服务时间,以0.1ms为单位(4位数BCD)。 |
| AR04 | 00 ~ 07 | 插槽1内板错误代码(Hex). 00:正常 01,02:硬件错误 04:串行通信板错误 |
| | 08 ~ 15 | 插槽2内板错误代码(Hex). 00:正常 01,02:硬件错误 03:PC设置错误 04:脉冲输出时PC停止或A/D(D/A)转换错误 |

3-6-2 内板的标志/位 (AR05 和 AR06)

插槽 2 高速计数器板的标志/位 (AR05 ~ AR06)

| 字 | 位 | 功 能 | 操 作 |
|------|---------|-------------|---------------------------------|
| AR05 | 00 | 高速计数器1复位位 | Z相和软件复位 0:Z相复位禁止 1:Z相复位允许 |
| | 01 | 高速计数器2复位位 | 仅软件复位 0:软件复位禁止 0→1:执行软件复位 |
| | 02 | 高速计数器3复位位 | --- |
| | 03 | 高速计数器4复位位 | --- |
| | 04 ~ 07 | 未使用 | --- |
| | 08 | 高速计数器1比较停止位 | 0→1:开始比较 1→0:停止比较 |
| | 09 | 高速计数器2比较停止位 | |
| | 10 | 高速计数器3比较停止位 | |
| | 11 | 高速计数器4比较停止位 | |
| | 12 | 高速计数器1停止位 | 0:继续执行 1:停止执行 |
| | 13 | 高速计数器2停止位 | |
| | 14 | 高速计数器3停止位 | |
| | 15 | 高速计数器4停止位 | |

| 字 | 位 | 功 能 | 操 作 |
|------|-------|-------------|--------------------------------|
| AR06 | 00 | 外部输出1强制置位位 | 0:无效 1:强制ON |
| | 01 | 外部输出2强制置位位 | |
| | 02 | 外部输出3强制置位位 | |
| | 03 | 外部输出4强制置位位 | |
| | 04 | 外部输出强制置位允许位 | 0:输出1~4强制置位禁止 1:输出1~4强制置位允许 |
| | 05~15 | 未使用 | - - - |

插槽 2 脉冲 I/O 板标志/位 (AR05 ~ AR06)

| 字 | 位 | 功 能 |
|------|---------|--|
| AR05 | 00 ~ 07 | 高速计数器1范围比较标志 位00 ON:计数器PV满足比较范围1的条件 位01 ON:计数器PV满足比较范围2的条件 位02 ON:计数器PV满足比较范围3的条件 位03 ON:计数器PV满足比较范围4的条件 位04 ON:计数器PV满足比较范围5的条件 位05 ON:计数器PV满足比较范围6的条件 位06 ON:计数器PV满足比较范围7的条件 位07 ON:计数器PV满足比较范围8的条件 |
| | 08 | 高速计数器1比较标志 OFF:停止 ON:比较 |
| | 09 | 高速计数器1上溢/下溢标志 OFF:正常 ON:上溢或下溢产生 |
| | 10 ~ 11 | 未使用 |
| | 12 ~ 15 | 端口1脉冲输出标志 位12 ON:减速指定(OFF:未指定) 位13 ON:脉冲数指定(OFF:未指定) 位14 ON:脉冲输出结束(OFF:未结束) 位15 ON:脉冲输出正在进行(OFF:无脉冲输出) |
| AR06 | 00 ~ 07 | 高速计数器2范围比较标志 位00 ON:计数器PV满足比较范围1的条件 位01 ON:计数器PV满足比较范围2的条件 位02 ON:计数器PV满足比较范围3的条件 位03 ON:计数器PV满足比较范围4的条件 位04 ON:计数器PV满足比较范围5的条件 位05 ON:计数器PV满足比较范围6的条件 位06 ON:计数器PV满足比较范围7的条件 位07 ON:计数器PV满足比较范围8的条件 |
| | 08 | 高速计数器2比较标志 OFF:停止 ON:比较 |
| | 09 | 高速计数器2上溢/下溢标志 OFF:正常 ON:上溢或下溢产生 |
| | 10 ~ 11 | 未使用 |
| | 12 ~ 15 | 端口2脉冲输出标志 位12 ON:减速指定(OFF:未指定) 位13 ON:脉冲数指定(OFF:未指定) 位14 ON:脉冲输出传来(OFF:未结束) 位15 ON:脉冲输出区在进行(OFF:无脉冲输出) |

绝对值编码器接口板标志/位 (AR05 ~ AR06)

| 字 | 位 | 功 能 |
|------|---------|--|
| AR05 | 00 ~ 07 | 高速计数器1范围比较标志 位00 ON: 计数器PV满足比较范围1的条件 位01 ON: 计数器PV满足比较范围2的条件 位02 ON: 计数器PV满足比较范围3的条件 位03 ON: 计数器PV满足比较范围4的条件 位04 ON: 计数器PV满足比较范围5的条件 位05 ON: 计数器PV满足比较范围6的条件 位06 ON: 计数器PV满足比较范围7的条件 位07 ON: 计数器PV满足比较范围8的条件 |
| | 08 | 高速计数器1比较标志 OFF: 停止 ON: 比较 |
| | 09 ~ 15 | 未使用 |
| AR06 | 00 ~ 07 | 高速计数器2范围比较标志 位00 ON: 计数器PV满足比较范围1的条件 位01 ON: 计数器PV满足比较范围2的条件 位02 ON: 计数器PV满足比较范围3的条件 位03 ON: 计数器PV满足比较范围4的条件 位04 ON: 计数器PV满足比较范围5的条件 位05 ON: 计数器PV满足比较范围6的条件 位06 ON: 计数器PV满足比较范围7的条件 位07 ON: 计数器PV满足比较范围8的条件 |
| | 08 | 高速计数器2比较标志 OFF: 停止 ON: 比较 |
| | 09 ~ 15 | 未使用 |

3-6-3 共用标志/位 (AR07 ~ AR27)

| 字 | 位 | 功 能 |
|------|---------|---|
| AR07 | 00 | Controller Link数据链接启动位 OFF→ON: 开始(电源上电时, 该位ON) ON→OFF: 停止 |
| | 01 ~ 11 | 未使用 |
| | 12 | DIP开关第6脚标志 OFF: CPU单元DIP开关第6脚OFF ON: CPU单元DIP开关第6脚ON |
| | 13 ~ 15 | 未使用 |

| 字 | 位 | 功 能 |
|-------|---------|--|
| AR 08 | 00 ~ 03 | RS - 232C口错误代码(1位数) 0:正常结束;1:校验错误;2:帧错误;3:过运行错误 |
| | 04 | RS - 232C口错误标志 CPU单元内置RS - 232C口产生通信错误时变ON |
| | 05 | RS - 232C口传送允许标志 仅当CPU单元内置RS - 232C口使用上位机链接或RS - 232C通信时有效 |
| | 06 | RS - 232C口接收完成标志 仅当CPU单元内置RS - 232C口使用RS - 232C通信时有效 |
| | 07 | RS - 232C口接收溢出标志 仅当CPU单元内置RS - 232C口使用上位机链接或RS - 232C通信时有效 |
| | 08 ~ 11 | 外围口错误代码(1位数) 0:正常结束;1:检验错误;2:帧错误;3:过运行错误 |
| | 12 | 外围口错误标志 外围口通信错误发生时变ON。 |
| | 13 | 外围口传送允许标志 仅当使用上位机链接或RS - 232C时有效 |
| | 14 | 外围口接收完成标志 仅使用RS - 232C通信时有效 |
| | 15 | 外围口接收溢出标志 仅使用上位机链接或RS - 232C通信时有效 |
| AR 09 | 00 ~ 15 | RS - 232C口接收计数器 4位数BCD,仅当RS - 232C通信使用时有效 |
| AR 10 | 00 ~ 15 | 外围口接收计数器 4位数BCD,仅当RS - 232C通信使用时有效 |
| AR 11 | 00 ~ 07 | 高速计数器0范围比较标志 位00 ON:计数器PV满足比较范围1的条件 位01 ON:计数器PV满足比较范围2的条件 位02 ON:计数器PV满足比较范围3的条件 位03 ON:计数器PV满足比较范围4的条件 位04 ON:计数器PV满足比较范围5的条件 位05 ON:计数器PV满足比较范围6的条件 位06 ON:计数器PV满足比较范围7的条件 位07 ON:计数器PV满足比较范围8的条件 |
| | 08 ~ 14 | 未使用 |
| | 15 | 脉冲输出位定义的脉冲输出状态 0:停止;1:输出 |
| AR 12 | 00 ~ 15 | 未使用 |
| AR 13 | 00 | 存储器盒安装标志 在电源上电时,存储器盒已安装时变ON |
| | 01 | 时钟可用标志 配备时钟的存储器盒安装后变ON |
| | 02 | 存储器盒写保护标志 当EEPROM或快闪内存存储器盒安装并写保护或安装EPROM存储器盒时ON |
| | 03 | 未使用 |
| | 04 ~ 07 | 存储器盒代码(1位数字) 0:未安装存储器盒 1:EEPROM,安装了4K字存储器盒 2:EEPROM,安装了8K字存储器盒 3:快闪内存,安装了16K字存储器盒 4:安装了EPROM类型存储器盒 |
| | 08 ~ 15 | 未使用 |

| 字 | 位 | 功 能 |
|------|-------|---|
| AR14 | 00 | CPU单元到存储器盒传送位 从CPU单元传送给存储器盒时变ON,操作结束后自动变OFF |
| | 01 | 存储器盒到CPU单元传送位 从存储器盒传送到CPU单元时变ON,操作结束后自动变OFF |
| | 02 | 存储器盒比较位 比较PC和存储器盒的内容时变ON,操作结束后自动变OFF |
| | 03 | 存储器比较结果标志 ON:发现差别或不能比较 OFF:内容比较后一致 |
| | 04~11 | 未使用 |
| | 12 | 编程模式传送错误标志 由于在编程模式下无法进行传送时变ON |
| | 13 | 写保护错误标志 由于写保护,传送无法执行时变ON |
| | 14 | 容量不足标志 由于传送目标容量不足,传送无法执行时变ON |
| | 15 | 无程序标志 由存储器盒中没有程序,传送无法执行时变ON |
| AR15 | 00~07 | 存储器盒程序代码 代码(2位数字)显示存储器盒中程序的大小 00: 没有程序或没有安装存储器盒 04: 程序小于3.2K字 08: 程序小于7.2K字 12: 程序小于11.2K字 16: 程序小于15.2K字 |
| | 08~15 | CPU单元程序代码 代码(2位数字)显示CPU单元中程序的大小 04: 程序小于3.2K字 08: 程序小于7.2K字 12: 程序小于11.2K字 16: 程序小于15.2K字 |
| AR16 | 00~10 | 未使用 |
| | 11 | PC设置初始化标志 在PC设置区检测错误发生,并且全部设置初始化回缺省设置时变ON |
| | 12 | 程序无效标志 在UM区(用户程序)产生检测错误或执行非法指令时变ON |
| | 13 | 指令表初始化标志 在指令表发生检测错误,并且全部设置初始化回缺省设置时变ON |
| | 14 | 存储器盒加装标志 电源供电时安装存储器盒时变ON |
| | 15 | 存储器盒传送错误标志 当DIP开关第2脚设置为ON时(也就是设置为刚上电时存储器盒内容自动传送)传送不能正常执行时变ON |
| AR17 | 00~07 | 当前时间的“分钟”部分,2位数BCD (仅配置时钟的存储器盒安装后有效,详细情况见159页) |
| | 08~15 | 当前时间的“小时”部分,2位数BCD (仅配置时钟的存储器盒安装后有效,详细情况见159页) |
| AR18 | 00~07 | 当前时间的“秒”部分,2位数BCD (仅配置时钟的存储器盒安装后有效,详细情况见159页) |
| | 08~15 | 当前时间的“分钟”部分,2位数BCD (仅配置时钟的存储器盒安装后有效,详细情况见159页) |

| 字 | 位 | 功 能 |
|------|---------|---|
| AR19 | 00 ~ 07 | 当前时间的“小时”部分,2位数BCD(仅安装了配置时钟的存储器盒时有效,详细情况见159页) |
| | 08 ~ 15 | 当前时间的“日期”部分,2位数BCD(仅安装了配置时钟的存储器盒时有效,详细情况见159页) |
| AR20 | 00 ~ 07 | 当前时间的“月份”部分,2位数BCD(仅安装了配置时钟的存储器盒时有效,详细情况见159页) |
| | 08 ~ 15 | 当前时间的“年份”部分,2位数BCD(仅安装了配置时钟的存储器盒时有效,详细情况见159页) |
| AR21 | 00 ~ 07 | 当前时间的“星期”部分,2位数BCD(00:星期天~06:星期六) (仅安装了配置时钟的存储器盒时有效,详细情况见159页) |
| | 08 ~ 12 | 未使用 |
| | 13 | 30秒校准位 仅安装了配置时钟的存储器盒时有效,详细内容见159页 |
| | 14 | 时钟停止位 仅安装了配置时钟的存储器盒时有效,详细内容见159页 |
| | 15 | 时钟设定位 仅安装了配置时钟的存储器盒时有效,详细内容见159页 |
| AR22 | 00 ~ 07 | 输入字 分配给输入位字的数目(2位数BCD)(仅存放可识别的数值,如果发生I/O单元超出错误时存放00) |
| | 08 ~ 15 | 输出字 分配给输出位字的数目(2位数BCD)(仅存放可识别的数值,如果发生I/O单元超出错误时存放00) |
| AR23 | 00 ~ 15 | 掉电计数器(4位数BCD) 这是一个电源关掉次数的计数,要清除计数,用编程装置写入“0000” |
| AR24 | 00 | 上电PC设置错误标志 DM6600 ~ DM6614中有错误时变ON(这部分PC设置在上电时读取) |
| | 01 | 启动PC设置错误标志 DM6615 ~ DM6644中有错误时变ON(这部分PC设置在执行开始时读取) |
| | 02 | 运行PC设置错误标志 DM6645 ~ DM6655中有错误时ON(这部分PC设置始终读取) |
| | 03 | CPU单元外围口设置改变标志 |
| | 04 | CPU单元RS-232C口设置改变标志 |
| | 05 | 循环时间长标志 当实际循环时间大于DM6619中设置的循环时间时ON |
| | 06,07 | 未使用 |
| | 08 ~ 15 | 显示检测I/O总线错误字编号的代码(2位数十六进制) 00 ~ 15(BCD):对应输入字000 ~ 015 80 ~ 95(BCD):对应输出字100 ~ 115 FO(十六进制):不能识别安装在插槽1上的内板 F1(十六进制):不能识别安装在插槽2上的内板 FF(十六进制):不能识别端盖 |
| AR25 | 00 ~ 07 | 未使用 |
| | 08 | FPD(一)教学位 |
| | 09 ~ 11 | 未使用 |
| | 12 | 跟踪完成标志 |
| | 13 | 跟踪标志 |
| | 14 | 跟踪触发位 |
| | 15 | 采样启动位(不要用程序改写此位) |

| 字 | 位 | 功能 |
|------|---------|---|
| AR26 | 00 ~ 15 | 最大循环时间(4位数BCD) 存放工作开始以来的最长循环时间,在工作开始时清除,而不是在结束 根据9F监视时间(DM6618)的设置,单位为下面几种,缺省:0.1ms;“10ms”设置:0.1ms; “100ms”设置:1ms;“1s”设置:10ms |
| AR27 | 00 ~ 15 | 当前循环时间(4位数BCD) 存放工作中的最近循环时间,工作停止时,当前循环时间不会清除 根据9F监视时间(DM6618)的设置,单位为下面几种,缺省:0.1ms;“10ms”设置:0.1ms; “100ms”设置:1ms;“1s”设置:10ms |

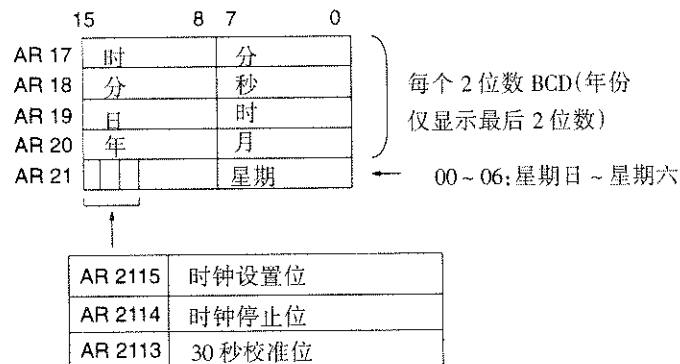
3-6-4 使用时钟

CQM1H PC 可通过安装带时钟的存储器盒而装备时钟,本节解释如何使用时钟。型号末尾有“R”的存储器盒内置时钟,例如,CQM1-ME04R 存储器盒有内置时钟,R来自于“实时时钟”。

注 如果存储器盒从 CPU 单元中移去,时钟会停止,并且当前日期和时间时钟数据会丢失。

包含日期和
时间的字

下面图例示意使用时钟时字的配置(AR17~AR21),这些字根据需要可以读和使用(提供 AR17 以便小时和分钟能快速读取)



设置时间

如下面使用编程装置设置时间

注 使用编程装置(如编程器)的菜单操作可很方便地设置时间,编程器步骤参考 CQM1H 操作手册。

设置各个内容

按下面步骤设置时间和日期:

- 1,2,3... 1. 使 AR2114(时钟停止位)变 ON 停止时钟,并允许 AR18~AR21 被覆写。
2. 使用编程装置,设置 AR18~AR20(分钟/秒,日期/小时,和年/月)和 AR2100~AR2107,(星期)
3. 到时间设置第 2 步时,使 AR2115(时钟设置位)变 ON,时钟会从设置的时间开始运行,时钟停止位和时钟设置位会自动变 OFF。

仅设置秒

使用 AR2113,可以不通过复杂的步骤,而简单地设置秒为“00”,当 AR2113 变 ON 后,时钟的时间如下面改变:

如果秒的设置是 00~29,秒会复位到“00”,并且分钟设置保留不变。
如果秒的设置是 30~59,秒会复位到“00”,并且分钟设置增加 1。
时间设置完成后,AR2113 会自动变 OFF。

3-7 LR 区

这些位用来在 1:1 数据链接 (CQM1H 和其它 PC 间) 或 Controller Link 数据链接时共享数据,这两个功能不能同时使用相同的 LR 位。

不用 1:1 数据链接时 LR 可用作工作位。

1 对 1 数据链接

两个 CPU 单元可连接建立 1:1 数据链接,共享两个 PC LR 区的数据,CQM1H 可 1 对 1 链接,下列 PC: CQM1H, CQM1, C200HX/HG/HE, C200HS, CPM1, CPM1A, CPM2A, CPM2C, 或 SRM1(-V2), 详情请参阅 1-6-4 1 对 1 链接通信。

注 由于 CPM1, CPM1A, CPM2A, 和 SRM1(-V2) PC LR 区比较小,当 CQM1H 和这些 PC 1:1 链接时,链接区域设置 (DM6645) 必须是 LR00~LR15。

Controller Link 数据链接

安装 Controller Link 单元,通过自动或手动设置,可建立 Controller Link 数据链接,详细内容参阅 Controller Link 单元的操作手册。

3-8 定时器/计数器区

该区域用来管理由 TIM, TIMH(15), CNT, CNTR(12) 和 TTIM(一) 建立的定时器和计数器,定时器和计数器使用相同的号,每个号在用户程序中只能使用一次,在不同的指令中也不要两次使用同一个 TIM/CNT 号。

TIM/CNT 号用于建立定时器和计数器,及访问完成标志和当前值 (PV),如果 TIM/CNT 号指定为字数据,将访问当前值 (PV); 如果用于位数据,将访问定时器/计数器的完成标志。

在使用的定时器/计数器的 PV 值变 0 时,完成标志变为 ON。

定时器和计数器的详情参见 224 页开始的指令。

确保 TIMH(15)精度

在循环时间超过 10ms 时,TIMH(15)应该使用 TIM/CNT 号 000~015 和中断处理,用其它定时器/计数器号或不用中断处理,会导致高速定时器不精确,中断处理在 PC 设置的 DM6629 中设定。

TIM 和 TIMH(15) PV 复位条件

当程序开始执行,指令输入条件变 OFF,或在互锁程序段 (IL—ILC) 中互锁条件变 OFF 时,PV 会复位到 SV。

TTIM(一)PV 复位条件

定时器复位输入变 ON 时 PV 会复位到 0000。

当程序开始执行,指令输入条件变 OFF,或在互锁程序段 (IL—ILC) 中,互锁条件变 OFF 时,PV 会保持。

CNT 和 CNTR (12)PV 复位条件

计数器复位输入变 ON 时,PV 会复位到 SV。

当程序开始执行,指令输入条件变 OFF,或在互锁程序段 (IL—ILC) 中,互锁条件变 OFF 时,PV 会保持。

3-9 DM 区

数据以字为单位访问,如下所示,DM 区读/写部分可用程序随意读取和写入,DM 区其余部分已预先指定给特殊功能。

| 名 称 | | 范 围 |
|-----------------|-----------------------|-----------------|
| 读/写 | 所有CQM1H CPU单元 | DM0000 ~ DM3071 |
| | 仅CQM1H CPU51/61 | DM3072 ~ DM6143 |
| 只读区 (见注1和注2) | 整个只读区域 | DM6144 ~ DM6568 |
| | Controller Link DM参数区 | DM6400 ~ DM6409 |
| | 路由表区 | DM6450 ~ DM6499 |
| | 串行通信板设置 | DM6550 ~ DM6559 |
| 错误日志区 | | DM6569 ~ DM6599 |
| PC设置(见注2) | | DM6600 ~ DM6655 |

- 注
1. 只读区域范围为 DM6144 ~ DM6568。
 2. 只读区域, PC 设置, 程序, 和扩展指令分配可作为一个单一数据块传送给或从存储器盒传送过来, 详细请见 3-11 使用存储器盒。

读/写 DM 区

读/写区域没有分配特殊功能,可任意使用,通过程序或编程装置可读出和写入,读/写区的大小根据 CPU 单元的型号,如下表所示。

| CPU单元 | 范 围 | 用指令访问 | | 用编程装置访问 | |
|---------------|--------------------|-------|---|---------|---|
| | | 读 | 写 | 读 | 写 |
| CQM1H - CPU11 | DM0000 ~ DM3071 | 是 | 是 | 是 | 是 |
| CQM1H - CPU21 | | | | | |
| CQM1H - CPU51 | DM0000 ~ DM6143 | | | | |
| CQM1H - CPU61 | | | | | |

只读区

(DM6144 ~ DM6568)

DM 地址从 DM6144 ~ DM6568 组成只读区域, 读区中的数据可通过指令读取(不能改写),用编程装置可以读取和改写,如果你不希望被程序修改,可以使用只读区存放数据。

为了防止使用编程装置改写数据,把 CPU 单元前面 DIP 开关 1 脚置 ON。

当使用 Controller Link 单元或串行通信板时, 部分只读区作为 Controller Link 参数/路由表或串行通信板设置,如下面表格所示。

| 名 称 | 范 围 | 用指令访问 | | 用编程装置访问 | |
|--------------------------|-----------------|-------|---|---------|-----------|
| | | 读 | 写 | 读 | 写 |
| Controller Link DM参数区 | DM6400 ~ DM6409 | 是 | 否 | 是 | 是 (见注) |
| 路由表区 | DM6450 ~ DM6499 | | | | |
| 串行通信板设置 | DM6550 ~ DM6559 | | | | |

- 注 当 CPU 单元前面 DIP 开关 1 脚 ON 时,数据不能用编程装置改写。

错误日志区
(DM6569 ~ DM6599)

CPU 单元在错误日志区自动记录最多 10 个错误 (致命和非致命) 的错误代码和日期/时间。

| 用指令访问 | | 用编程装置访问 | |
|-------|---|---------|---|
| 读 | 写 | 读 | 写 |
| 是 | 否 | 是 | 否 |

PC 设置
(DM6600 ~ DM6655)

PC 设置包括了所有 PC 设置值, 除了串行通信板设置值 (存放在 DM6550 ~ DM6559)。用编程装置设置 PC 设置值。

| 用指令访问 | | 用编程装置访问 | |
|-------|---|---------|---|
| 读 | 写 | 读 | 写 |
| 是 | 否 | 是 | 否 |

3-10 EM 区

EM 区仅可在 CQM1H—CPU61 CPU 单元中使用, EM 数据以字为单位存取, 由于仅有一个 EM 区可用, 区的定义就不需要了。

EM 区地址范围从 EM0000 ~ EM6143, 这个区域没有分配特殊功能, 可任意使用, 能通过程序或编程装置读取和写入。

3-11 存储器盒用法

这一节提供了存储器盒规格的一般信息, 并解释如何读出、写入和比较存储器盒中内容。安装存储器盒, 写保护快闪内存或 EEPROM 存储器盒, 调换 EPROM 芯片和修改 EPROM 版本开关设置, 详细参考 *CQM1H 操作手册*。

可选择存储器盒存放程序, 只读 DM(DM6114 ~ DM6568)、PC 设置 (DM6600 ~ DM6655), 和扩展指令分配, 用存储器盒记录这些数据, 防止程序和重要设置被意外修改, 另外, 不同控制过程所需的设置和程序通过简单地调换存储器盒就很容易地改变。

程序可写入 CPU 单元内部 RAM, 在没有存储器盒情况下运行 CQM1H。但是使用存储器盒时, 其内容在启动时传送给 CPU 单元, 这样, 即使 CPU 单元电池失效, CQM1H 同样能运行。

时钟功能

CQM1H PC 可通过安装带时钟的存储器盒配置时钟, 内置时钟的存储器盒型号末尾有字母“R”, 详细见 3-6-4 *使用时钟*。

不同 CPU 单元间的兼容性

用 CQM1H CPU 单元写入存储器盒的数据不能被 CQM1 CPU 单元读取, 但用 CQM1 CPU 单元写入的数据可被 CQM1H CPU 单元读取。

用 CQM1H—CPU61 写入存储器盒的数据可被 CQM1H—CPU51, CQM1H—CPU21, 和 CQM1H—CPU11 单元读取, 但如果使用了 EM 区地址, 程序运行会不正确。

3-11-1 存储器盒和内容

可用存储器盒

下面存储器盒是可用的。

| 内存 | 型号 | 规格 |
|-----------------|---------------|--------------------|
| EEPROM (见注2) | CQM1 - ME04K | 4K字不带时钟 |
| | CQM1 - ME04R | 4K字带时钟 |
| | CQM1 - ME08K | 8K字不带时钟 |
| | CQM1 - ME08R | 8K字带时钟 |
| 快闪内存(见注1和注2) | CQM1H - ME16K | 16K字不带时钟 |
| | CQM1H - ME16R | 16K字带时钟 |
| EPROM (见注2) | CQM1 - MP08K | 8K, 16K, 32K字不带时钟 |
| | CQM1 - MP08R | 8K, 16K, 32K, 字带时钟 |

- 注
1. 用编程装置可对 EEPROM 存储器盒读出和写入数据。
 2. 用编程装置可从 EPROM 存储器盒中读出数据, 但必须用 EPROM 写入器写入, 存储器盒中可安装 8K, 16K, 32K 字的 EPROM 芯片。
 3. CQM1H—ME16K 和 CQM1H—ME16R 不能用在 CQM1 PC 中

下面 EPROM 芯片(另外出售)用在 EPROM 存储器盒中。

| 型号 | ROM版本 | 容量 | 存取速度 |
|--------------|----------|------|-------|
| ROM - ID - B | 27128或相当 | 8K字 | 150ns |
| ROM - JD - B | 27256或相当 | 16K字 | 150ns |
| ROM - KD - B | 27512或相当 | 32K字 | 150ns |

调换 EPROM 芯片和改变存储器盒 EPROM 版本开关设置的详细情况参考 CQM1H 操作手册。

内容

存储在存储器盒中的数据主要是 CPU 单元的只读 DM、PC 设置和程序, 如下面表格所示, 所有这些数据作为一个单元处理; 4 个区域不能单独读取、写入、或比较。

| 信息 | | 内容 |
|------|--------|--|
| DM区 | 只读区 | 只读DM不能用程序写入, 范围是DM6144 ~ DM6568, 这些字可任意使用。 |
| | PC设置 | PC设置区设置CQM1H的运行参数并存放在DM6600 ~ DM6655。 |
| | 扩展指令分配 | 这些设置指示已分配功能代码的扩展指令。 |
| 用户程序 | | 全部用户程序。 |

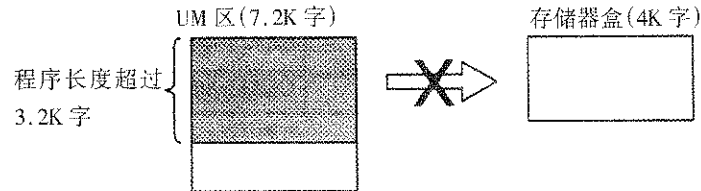
3-11-2 存储器盒容量和程序大小

下面表格显示每种大小的存储器盒可存放的最大程序。

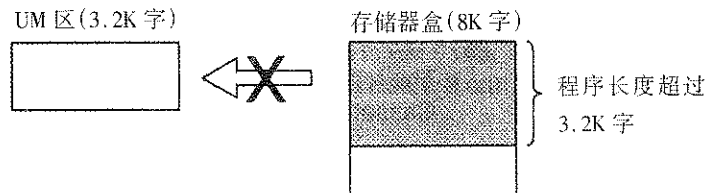
| 存储器盒大小 | 最大程序容量 |
|--------|--------|
| 4K字 | 3.2K字 |
| 8K字 | 7.2K字 |
| 16K字 | 15.2K字 |

如果试图在存储器盒中存放的程序太大或从 CPU 单元读取的程序太大，一个非致命错误会产生，传送不执行，下面是二个例子。

- 1,2,3... 1. 当一个 4K 字的 EEPROM 存储器盒安装在具有 7.2K 字 UM 区的 CPU 单元中时，程序最多能写 3.2K 字到存储器盒。如果试图写超过 3.2K 字的程序到存储器盒，就会出现非致命错误。



2. 当一个 8K 字或更大的存储器盒安装在具有 3.2K 字 UM(用户程序)区的 CPU 单元中时，程序最多为 3.2K 字时能从存储器盒中读出。如果试图从存储器盒中读超过 3.2K 字的程序，就会出现非致命错误。



注 上述 2 种传送在程序为 3.2K 字或更小时会正常结束。UM(用户程序)区和存储器盒中程序的大致的大小能由 AR15 中的内容决定，如下表所示。

| 程序位置 | 位 | 内容 | 意义 |
|------|---------------|----|---------------------------------------|
| 存储器盒 | AR1500~AR1507 | 00 | 没有安装存储器盒或存储器盒中没有保存程序。 |
| | | 04 | 程序少于3.2K字,任何CQM1H CPU单元都能读。 |
| | | 08 | 程序少于7.2K字,只能由CQM1H CPU - 51/61 CPU单元读 |
| | | 12 | 程序少于11.2K字长时,只能由CQM1H - CPU61 CPU单元读 |
| | | 16 | 程序少于15.2K字长时,只能由CQPM1H - CPU61 CPU单元读 |
| UM区 | AR1508~AR1515 | 04 | 程序少于3.2K字长时,可写入任何快闪内存或EEPROM存储器盒。 |
| | | 08 | 程序少于7.2K字长时,可写入8K字快闪内存存储器盒。 |
| | | 12 | 程序少于11.2K字长时,仅可写入16K字快闪内存存储器盒。 |
| | | 16 | 程序少于15.2K字长时,仅可写入16K字快闪内存存储器盒。 |

在 CQM1H—CPU11/21 CPU 中单元中，AR1508 ~ AR1515 的内容一般为 04，在安装 4K 字存储器盒时，AR1500 ~ AR1507 的内容一般为 04。

AR15 中指示的程序大小不包括 END(01) 指令后的 NOP(00) 指令,但包括其它任何不是 NOP(00)的指令。确认清除 END(01)后无用的指令,以得到程序大小的精确测量。

3-11-3 写到存储器盒

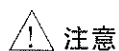
这一节解释如何把 CPU 单元的数据写入快闪内存或 EEPROM 存储器盒。

注 把数据写入 EPROM 存储器盒时需要 PROM 写入器和支持软件,详细参考支持软件的操作手册。

步骤

下面的步骤描述了写入快闪内存或 EEPROM 存储器盒。

- 1, 2, 3... 1. 检查存储器盒的写保护开关为 OFF(即允许写),如果允许写,存储器盒写保护标志 (AR1302)变 OFF。如果开关为 ON(即不允许写),则关断 CQM1H 电源,取出存储器盒并改变开关。
2. 检查 CQM1H 在编程模式,如果在运行或监视模式下,使用编程装置改变模式。
3. 用编程装置使 AR1400 变 ON,信息从 CQM1H 写入存储器盒,操作完成后,AR1400 自动会变 OFF。



注意

如果出现内存错误,数据不能写入存储器盒。

注 如果在传输时出现错误,就会产生一个非致命错误(FAL 9D),相应的 AR 位(从 AR1412 ~ AR1415)变 ON/OFF。如果出现,见第 8 章 故障诊断并进行必要的改正。

3-11-4 从存储器盒读

有 2 种方法可从存储器盒读,用编程装置使存储器盒到 CPU 单元传送位 (AR1401) 变 ON 或 CPU 单元 DIP 开关 2 脚置 ON,在启动时自动从存储器盒读数据。

如果存储器盒中的程序有功能代码和缺省设定不同的扩展指令,必须确认 CPU 单元的 DIP 开关 4 脚为 ON(指示用户分配功能代码)。

存储器盒中的内容不能用程序读取。

对任何类型存储器盒都能执行从存储器盒中读取。

如果传输数据时出现错误,将产生非致命错误 (FAL 9D),相应 AR 位(从 AR1412 ~ AR1415)变 ON/OFF。(如果出现,见排除故障章并进行必要的改正。


编程装置步骤

用编程装置从存储器盒读的步骤概述如下。

- 1, 2, 3... 1. 检查 CQM1H 为编程模式,如果为运行或监视模式,用编程装置改变模式。
2. 使用编程装置使 AR1401 变 ON,信息从存储器盒读到 CQM1H,读操作完成后,AR1401 自动变为 OFF。

启动时自动传送

如果 CPU 单元的 DIP 开关 2 脚为 ON, CQM1H 电源上电时, 数据从存储器盒自动读。如果在存储器盒和 CQM1H 内存间传送数据时出现错误, 就会出现内存错误并且不能工作。

 **注意** 改变 CQM1H 的 DIP 开关设定时, 一定要确认电源已关断。

3-11-5 比较存储器盒内容

存储器盒的内容能与 CQM1H 内存中的内容进行比较, 看它们是否相同。能对任何类型存储器盒进行比较。

步骤

使用如下步骤。

- 1, 2, 3... 1. 检查 CQM1H 为编程模式, 如果为运行或监视模式, 用编程设备将其改变为编程模式。
2. 用编程设备使 AR1402 为 ON, 存储器盒的内容将和 CQM1H 内存的内容比较。在比较操作结束后, AR1402 自动变为 OFF。
3. 检查 AR1403 的状态来查看比较结果。如果内容不相同或因为 CQM1H 不在编程模式而使比较不能进行, 则 AR1403 为 ON。如果 AR1403 为 OFF, 则比较成功且内容相同。

AR1403 不能由程序控制或编程设备控制, 只能由比较结果控制。

如果 CQM1H 不在编程模式时进行比较, 会出现非致命错误 (FAL 9D) 且 AR1412 变为 ON, 尽管 AR1403 也为 ON, 比较并没有进行。如果存储器盒没有安装在 CQM1H 上而试图比较, AR1403 也会变为 ON。

第 4 章 梯形图编程

本章解释了编写一个基本梯形图程序的基本步骤和概念。它介绍了用于构造梯形图的基本结构和控制其执行的指令。编程中使用的整个指令集在第五章 指令集中描述。

| | | |
|-------|---------------------|-----|
| 4—1 | 基本步骤 | 168 |
| 4—2 | 指令术语 | 168 |
| 4—3 | 基本梯形图 | 169 |
| 4—3—1 | 基本术语 | 169 |
| 4—3—2 | 助记符代码 | 170 |
| 4—3—3 | 梯形图指令 | 171 |
| 4—3—4 | OUTPUT 和 OUTPUT NOT | 174 |
| 4—3—5 | END 指令 | 175 |
| 4—3—6 | 逻辑块指令 | 175 |
| 4—3—7 | 编码多右侧指令 | 184 |
| 4—3—8 | 分支指令行 | 184 |
| 4—3—9 | 跳转 | 188 |
| 4—4 | 控制位状态 | 189 |
| 4—4—1 | 置位和复位 | 189 |
| 4—4—2 | 微分上升和微分下降 | 190 |
| 4—4—3 | 保持 | 190 |
| 4—4—4 | 自保持位(封口) | 191 |
| 4—5 | 工作位(内部继电器) | 191 |
| 4—6 | 编程注意事项 | 193 |
| 4—7 | 程序执行 | 195 |

4-1 基本步骤

编写程序有几个基本步骤。可以复制所提供的附录 E I/O 分配表和附录 F 程序代码表以帮助编程。

- 1, 2, 3... 1. 应取得一份所有 I/O 设备及分配给它们的 I/O 点的清单。并准备一份表示分配到各 I/O 设备的 I/O 位的表格。
2. 如果使用 LR 位链接两台 PC, 准备一份表示这些位的用法表。
3. 要确定那些字可用于工作位, 并准备一份表格, 以方便对这些工作的分配以及使用。
4. 同时准备 TC(定时器和计数器) 号和跳转号的表格, 这样, 使用它们时就可分配它们。要记住, 在程序中, 一个 TC 号的功能只可以定义一次; 跳转号 01 至 99 每个也只能使用一次。(TC 号在 5-16 定时器和计数器指令中说明; 跳转号在本节的后面说明。)
5. 画梯形图。
6. 将程序输入到 CPU。当使用编程器时, 这个步骤将包括将程序转换为助记符形式。
7. 查出程序的语法错误并改正之。
8. 执行程序以检查是否有执行错误并改正。
9. 当全部控制系统安装好以后并可投入使用时, 执行程序, 如果需要的话, 进一步细调。

梯形图编程及转换到助记符代码的基础知识在 4-3 基本梯形图中描述, 准备和输入程序, 用编程器的方法在 *CQM1H 操作手册* 中说明, 用 *CX - Programmer* 的方法在 *CX - Programmer 用户手册* 中说明。

第 4 章的其余部分包括高级编程, 编程预备知识和程序执行, 所有的特殊应用指令在第 5 章 指令集中说明, 调试在 *CQM1H 操作手册* 和 *CX - Programmer 用户手册* 中说明, 第 8 章 故障诊断中也提供了所需的调试信息。

4-2 指令术语

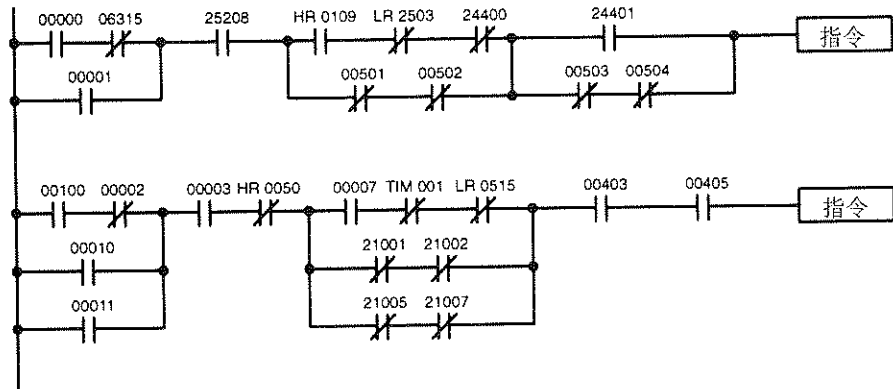
梯形图编程有两种基本指令: 1. 对应梯形图上的条件的指令, 仅当程序转换为助记符代码时才使用指令形式; 2. 是用在梯形图右侧的指令, 并根据它们前面的指令行的条件而执行。

大多数指令相应地至少有一个或更多的操作数。操作数指示或提供指令执行的数据。这些操作数有时作为实际数值输入, 但通常是包含要使用的数据的数据区位或字的地址。例如 MOVE 指令指定 IR000 作为源操作数, 因此将把 IR000 中的内容移到其它某个位置。这个位置也被指定为一个操作数。地址被指定为操作数的位称为操作数位。地址被指定为操作数的字称为操作数字。如果实际数值是作为一个常数输入的, 它要加上前缀 # 以标明它不是地址。

描述指令所用的其它术语在第五章 指令集中介绍。

4-3 基本梯形图

梯形图由一条位于左侧的竖直向下的线和一些向右的分支线组成。左侧的线称为母线；分支线称为指令行或梯级。沿着指令行，布置着指向右侧的其它指令的条件。这些条件的逻辑组合决定了右侧的指令何时、如何执行。一个梯形图如下所示。



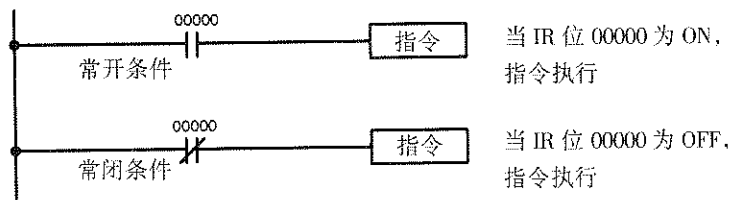
如上面梯形图所示，指令行可以分支开来，也可以合并回去。竖对线称为条件。没有斜线与它们相交的条件称为常开条件，对应于 LOAD, AND 或 OR 指令。有斜线与它们相交的条件称为常闭条件，对应于 LOAD NOT, AND NOT 或 OR NOT 指令。各个条件上面的数字表明指令的操作数位。各个条件的相应位的状态决定了下面指令的执行条件。对应于条件的各个指令的操作方法在下面说明。在考虑这些之前，有些基本术语必须解释。

注 当用 CX - Programmer 显示梯形图时，在梯形图右侧显示第二条母线，并连接到右侧的所有指令。但这不会改变梯形图程序的功能。在右侧的指令和右侧的母线之间不能放置条件，即右侧指令必须直接与右侧母线相连接。详细请参照 CX - Programmer 用户手册。

4-3-1 基本术语

常开和常闭条件

在梯形图中每个条件可以为 ON 或 OFF，这取决于已分配给它的操作数位的状态。如果操作数位为 ON；常开条件为 ON，如果操作数位为 OFF，常开条件即为 OFF。对常闭条件来说，如果操作数位为 OFF 它为 ON；如果操作数位为 ON 它为 OFF。一般来说，希望某位为 ON 时发生动作，则使用常开条件；希望某位 OFF 时发生动作，则使用常闭条件。



| | |
|------|---|
| 执行条件 | 在梯形图编程中，一条指令之前的 ON 和 OFF 条件的逻辑组合决定指令执行的综合条件。此条件，非 ON 即 OFF，称为指令的执行条件。除了 LOAD 指令，所有的指令具有执行条件。 |
| 操作数位 | IR, SR, HR, AR, LR 或 TC 区域的任何位，都可指定为任何梯形图指令的操作数。这意味着梯形图中的条件可由 I/O 位，标志位，工作位，定时器/计数器等决定。LOAD 和 OUTPUT 指令也可以使用 TR 区的位，但是它们只在特殊应用中才这样使用。详细请参照 4-3-8 分支指令行。 |
| 逻辑块 | 条件与指令的对应关系，取决于在指令行内部各条件之间的相互关系。任何一组结合起来，产生逻辑结果的条件称为逻辑块，虽然编写梯形图时不需要实际分析单个逻辑块，但是理解逻辑块对于有效的编程是必须的；当以助记符代码输入程序时，逻辑块是基础。 |
| 指令块 | 一个指令块由跨接梯形图的所有相互连接的指令组成。因此一个指令块由这些指令组成，这些指令在你可以不与任何竖直线相交而画一条水平线和下一个这样的地方之间。 |

4-3-2 助记符代码

梯形图不能通过编程器直接输入到 PC；而需要 CX - Programmer。要从编程器中输入，必须把梯形图转换为助记符代码。助记符代码能提供与梯形图完全一样的信息，但是是一种可以直接键入 PC 的形式。实际上可以直接用助记符代码编程，虽然对于初学者或对于编写复杂程序不提倡这种做法。同时，不管使用什么编程器，程序都是以助记符形式存储在存储器中的。因此，理解助记符代码很重要。

由于编程器作为外围设备的重要性和助记符代码对完全理解程序的重要性，我们将随同梯形图一起引入和介绍助记符代码。请记住，如果是通过 CX - Programmer 输入程序的，将不需要使用助记符代码（虽然也可以在用 CX - Programmer 时使用它，如果愿意的话）。

程序存储器结构

程序是输入到程序存储器地址中去的。程序存储器的地址同其它存储区的地址有略微的差别，因为每个地址不必包含相同数目的数据。相反，每个地址包含一条指令和此指令所需的定义和操作数（后面将详细介绍）。因为有些指令不需要操作数，而有些最多需要三个操作数，所以程序存储器地址可以为一到四字长。

程序存储器地址起始于 00000，一直到程序存储器容量耗尽。各个地址的第一个字定义指令。指令使用的定义符也包含在第一个字中。同时，如果指令只需要一个位操作数（没有定义符），此位操作数也作为指令编到同一行中。指令需要的其它字包括指定使用什么数据的操作数。当转换为助记符时，除了梯形图指令外，所用的其它（指令）都写成同一形式，一个字一行，如同它们以梯形图符号出现的一样。下面显示一个助记符代码的例子。其中使用的指令在手册的后面介绍。

| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | HR 0001 |
| 00001 | AND | 00001 |
| 00002 | OR | 00002 |
| 00003 | LD NOT | 00100 |
| 00004 | AND | 00101 |
| 00005 | AND LD | |
| 00006 | MOV(21) | |
| | | 000 |
| | | DM 0000 |
| 00007 | CMP(20) | |
| | | DM 0000 |
| | | HR 00 |
| 00008 | AND | 25505 |
| 00009 | OUT | 10000 |
| 00010 | MOV(21) | |
| | | DM 0000 |
| | | DM 0500 |
| 00011 | LD | 00502 |
| 00012 | AND | 00005 |
| 00013 | OUT | 10003 |

助记符代码表的地址和指令列只用指令字填入。对于所有其它行,左边两侧留为空白。如果指令不需要定义符或位操作数,第一行的操作数列留为空白。在空白的数据列空间(用于不需要数据的所有指令字)上划交叉线是个好办法,因为这样可以迅速地扫描到数据列,以确定是否有(程序存储)地址留下。

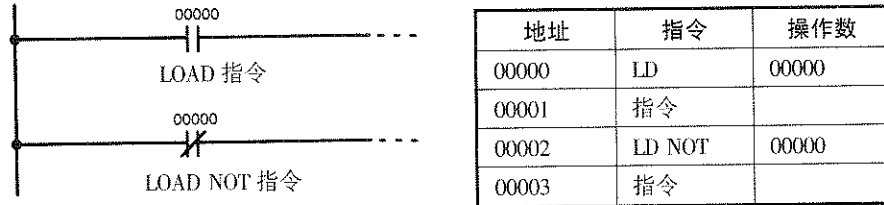
编程时,会自动地显示地址,并不需要输入,除非为了某些原因,希望为指令指定另外的位置。当转换为助记符代码时,最好从程序存储地址 00000 起始,除非有特定的原因要求程序起始于其它的地方。

4-3-3 梯形图指令

梯形图指令是与梯形图上的条件相适应的指令。梯形图指令可以单独地与下面将要介绍的逻辑块相结合构成执行条件,它是所有其它指令的执行的基础。

LOAD 和 LOAD NOT

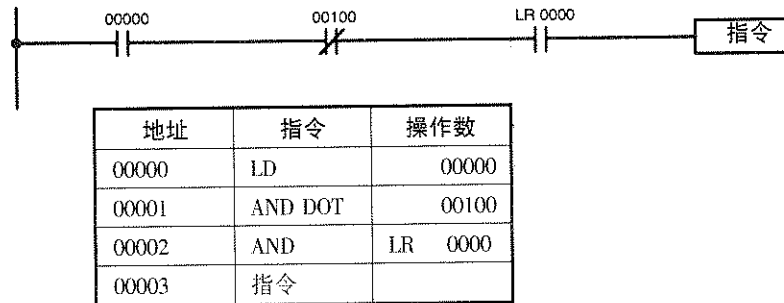
在梯形图中起始任一逻辑块的第一个条件为 LOAD 和 LOAD NOT 指令。每个指令需要一行助记符代码。在下面的例子中,用“指令”作为一个虚拟指令,可以是手册后面将要介绍的任一种右侧指令。



当这是指令行的唯一条件时,当条件为 ON 时,指令的执行条件就为 ON。对于 LOAD 指令(即,常开条件),当 IR00000 为 ON 时,执行条件为 ON;对于 LOAD NOT 指令(即,常闭条件)当 00000 为 OFF 时,执行条件为 ON。

AND 和 AND NOT

当同一指令行上串联两个或更多条件时,第一个条件对应为 LOAD 或 LOAD NOT 指令;其余的条件对应为 AND 或 AND NOT 指令。下面的例子表示三个条件,左起顺序相应地为 LOAD,AND NOT 和 AND 指令。同样,每个指令需要一行助记符。



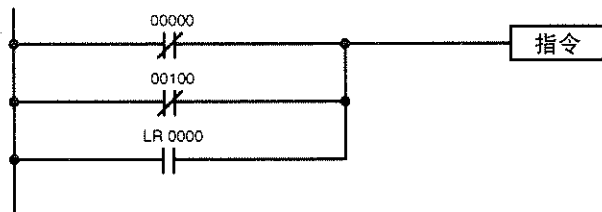
仅当所有三个条件都为 ON 时,指令的执行条件才为 ON。即,IR00000 为 ON,IR00100 为 OFF,LR0000 为 ON。

串联的 AND 指令可以分别地考虑,每个指令将执行条件(即,到达那一点上条件的总和)与 AND 指令的操作位状态进行逻辑与运算。如果这两个都是 ON,将使下一个指令的执行条件为 ON。如果有一个为 OFF,结果也为 OFF。串联的第一个 AND 指令的执行条件是这个指令行上的第一个条件。

串联的每个 AND NOT 指令,将其执行条件与其操作数位的反码进行逻辑与运算。

OR 和 OR NOT

当同一指令行上并联两个或更多条件时，第一个条件对应为 LOAD 或 LOAD NOT 指令；其余的条件对应为 OR 或 OR NOT 指令。下面的例子示出了三个条件，上起顺序相应地为 LOAD NOT, OR NOT 和 OR 指令。同样，每个指令需要一行助记符。



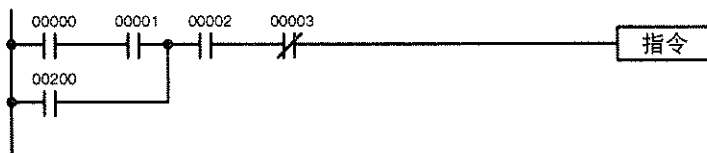
| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD NOT | 00000 |
| 00001 | OR NOT | 00100 |
| 00002 | OR | LR 0000 |
| 00003 | 指令 | |

当三个条件之中任一个为 ON 时，指令的执行条件即为 ON。即，IR00000 为 OFF, 或 IR00100 为 OFF, 或 LR0000 为 ON。

OR 和 OR NOT 指令可以单独地考虑，每个(指令)将其执行条件与 OR 指令的操作数位状态进行逻辑或运算。如果两个之中有一个为 ON, 将使下一个指令的执行条件为 ON。

组合 AND 和 OR 指令

当 AND 和 OR 指令组合更复杂的梯形图时，它们有时可以分别单独考虑，每个指令将其执行条件与其操作数位状态进行逻辑运算。下面是一个例子。仔细研究这个例子，直到你认识到助记符代码同梯形图一样遵循同样的逻辑流。



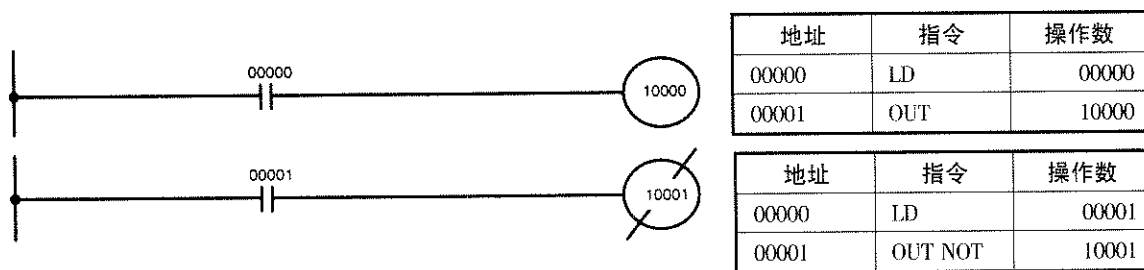
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND | 00001 |
| 00002 | OR | 00200 |
| 00003 | AND | 00002 |
| 00004 | AND NOT | 00003 |
| 00005 | 指令 | |

这里，将 IR00000 的状态和 IR00001 的状态进行与运算，以决定与 IR00200 进行 OR 运算的执行条件。这个操作的结果决定与 IR00002 的状态进行 AND 的执行条件，此执行条件又决定与 IR00003 的状态的取反进行 AND(即, AND NOT)的执行条件。

然而,在更复杂的梯形图中,在能够决定最终指令的执行条件之前,必须考虑逻辑块。那正是使用 AND LOAD 和 OR LOAD 指令的地方。但在我们考虑更复杂的梯形图之前,我们来看看完成一个简单的“输入—输出”程序所需要的指令。

4—3—4 OUTPUT 和 OUTPUT NOT

输出组合执行条件的结果的最简单的方法是直接用 OUTPUT 和 OUTPUT NOT 指令。这些指令用按相应的执行条件控制指定操作数位的状态。用 OUTPUT 指令时,当执行条件为 ON 时,操作数位为 ON;当执行条件为 OFF 时,操作数位为 OFF。用 OUTPUT NOT 指令,当执行条件为 OFF 时,操作数位为 ON;当执行条件为 ON 时,操作数位为 OFF。如下所示。用助记符代码表示,每个指令需要一行。

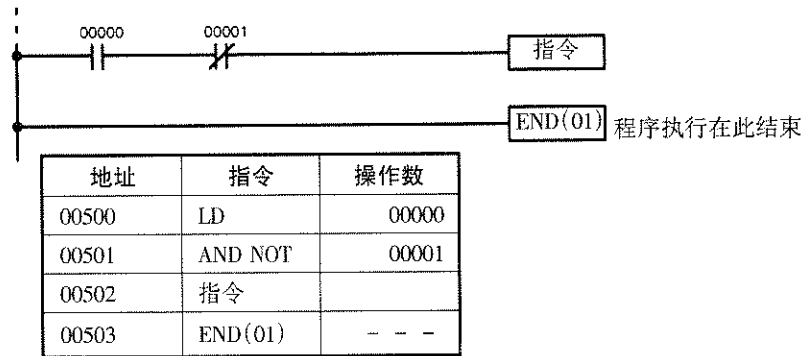


在上面的例子中,当 IR00000 为 ON 时,IR10000 将为 ON;当 IR00001 为 ON 时,IR10001 将为 OFF。这里,IR00000 和 IR00001 为分配给由 PC 控制的单元的输入位,IR10000 和 IR10001 为分配给由 PC 控制的单元的输出位,即进入输入点的信号分配给 IR00000 和 IR00001,控制输出点分配给 IR10000 和 IR10001。

一个位为 ON 或 OFF 的时间长度,可由 OUTPUT 或 OUTPUT NOT 指令与计时器指令组合来进行控制。详细请参照 5—16—1 定时器—TIM 中的例子。

4-3-5 END 指令

完成一个简单程序所需的最后一条指令为 END 指令。当 CPU 单元扫描程序时,它执行所有的指令,直到第一个 END 指令,然后返回到程序的起始处再次开始执行程序。虽然 END 指令可以放到程序的任一点(当调试时有时会这样做),但是 END 指令后面的指令将不会执行,直到 END 指令移去。助记符中,END 指令后面的数字是它的功能代码。大多数指令输入到 PC 中去时使用它。这将在后面说明。END 指令不需要操作数;不能将条件与它放到同一个指令行上。



如果程序中没有 END 指令,程序将不会执行。

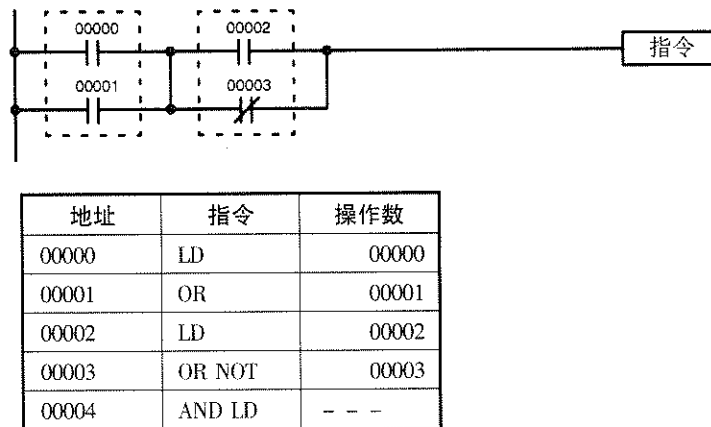
现在你已经知道了编写简单输入—输出程序的所有指令了,在结束梯形图基础、开始输入程序到 PC 之前,再来看看逻辑块指令 (AND LOAD 和 OR LOAD),它即使在简单的梯形图中有时也需要。

4-3-6 逻辑块指令

逻辑块指令并不对应于梯形图上的特定的条件;相反,它们描述了逻辑块之间的关系。AND LOAD 指令将两个逻辑块产生的执行条件进行逻辑与运算。OR LOAD 指令将两个逻辑块产生的执行条件进行逻辑或运算。

AND LOAD

下面的梯形图虽然看起来简单,却需要一个 AND LOAD 指令。



两个逻辑块用虚线标注。研究这个例子，会发现：当左边的逻辑块的条件为 ON(即，IR00000 或 IR00001 为 ON)，同时右边的逻辑块的条件为 ON(即，IR00002 为 ON 或 IR00003 为 OFF)时，则执行条件为 ON。

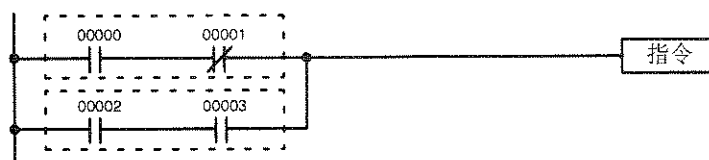
然而，上面的梯形图仅用 AND 和 OR 指令不能转换为助记符代码。如果试图在 IR00000 和 IR00001 相或的结果与 IR00002 之间用 AND，则 IR00002 和 IR00003 之间的 OR NOT 会丢失，并且 OR NOT 变成 IR00003 和 IR00002 与第一个 OR 之间的 AND 运算结果的 OR NOT 运算。因此需要先独立完成 OR (NOT)，然后组合这两个结果的办法。

为此，可以在指令行的中间使用 LOAD 或 LOAD NOT 指令。当用这种方法执行 LOAD 或 LOAD NOT 时，当前的执行条件保存在特殊的缓冲区中，并重新开始进行逻辑处理。为将当前的执行条件的结果与先前“未使用”的执行条件组合起来，要使用 AND LOAD 或 OR LOAD 指令。这里，“LOAD”指的是装载上一次未使用的执行条件。利用 LOAD 或 LOAD NOT 指令，可以为指令行中除了第一个条件的其它任何条件产生未使用执行条件。

从助记符上分析上面的梯形图，IR00000 的条件是 LOAD 指令，在它下面是 IR00000 的状态和 IR00001 的状态之间的 OR 指令。IR00002 的条件是另一个 LOAD 指令，它下面的条件是 OR NOT 指令，即，IR00002 的状态和 IR00003 的状态的取反再 OR。为得到右边指令的执行条件，必须将从这两个逻辑块中产生的结果进行逻辑与运算。AND LOAD 做了此工作。梯形图对应的助记符代码显示在下面。AND LOAD 指令本身不需要操作数，因为它对前面已经确定的执行条件进行操作。这里，用断线表明不需要指定或输入操作数。

OR LOAD

下面的梯形图在上层的逻辑块和下层的逻辑块之间需要一个 OR LOAD 指令。当 IR00000 为 ON，同时 IR00001 为 OFF；或者 IR00002 和 IR00003 同时为 ON 时，右边的指令的执行条件将为 ON。除了当前的执行条件要同上一个未使用的执行条件进行 OR 运算外，OR LOAD 指令的操作和助记符代码与 AND LOAD 指令的操作和助记符完全相同。



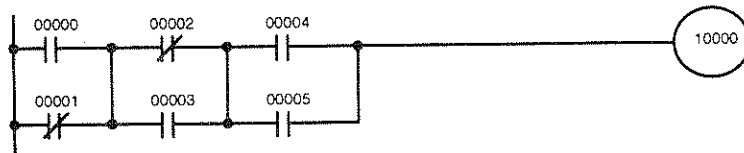
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND DOT | 00001 |
| 00002 | LD | 00002 |
| 00003 | AND | 00003 |
| 00004 | OR LD | - - - |

自然地，有些梯形图既需要 AND LOAD 指令又需要 OR LOAD 指令。

串联的逻辑块指令

为将包含串联的逻辑块的梯形图转化为代码，必须将梯形图分解为逻辑块。各块使用 LOAD 指令将第一个条件转化为代码，然后用 AND LOAD 或 OR LOAD 将各块进行逻辑组合。对于 AND LOAD 和 OR LOAD，有两种方法可以作到这一点。一种是在起始两个逻辑块后就对逻辑块指令进行编码，然后逐个对附加块进行编码。另一种是先对所有要组合的块进行编码，各个块用 LOAD 或 LOAD NOT 作为开头。然后对逻辑块进行组合编码。在此情况下，最后一对块的指令应该首先组合，然后同前面的各块组合，一直到第一个块。虽然这两种方法产生的结果完全一样，然后，第二种将所有的逻辑块指令一起进行编码的方法，只有在八个或更少的块进行组合的情况下才可以使用，即，如果需要七个或更少的逻辑块指令的情况。

下面的梯形图需要用 AND LOAD 指令以转换为助记符形式，因为有三对并联的条件相串联。在下面表示两种程序编码方法。

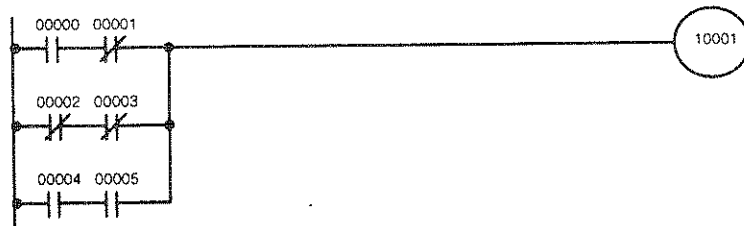


| 地址 | 指令 | 操作数 |
|-------|--------|-------|
| 00000 | LD | 00000 |
| 00001 | OR DOT | 00001 |
| 00002 | LD NOT | 00002 |
| 00003 | OR | 00003 |
| 00004 | AND LD | - - - |
| 00005 | LD | 00004 |
| 00006 | OR | 00005 |
| 00007 | AND LD | - |
| 00008 | OUT | 10000 |

| 地址 | 指令 | 操作数 |
|-------|--------|-------|
| 00000 | LD | 00000 |
| 00001 | OR DOT | 00001 |
| 00002 | LD NOT | 00002 |
| 00003 | OR | 00003 |
| 00004 | LD | 00004 |
| 00005 | OR | 00005 |
| 00006 | AND LD | - |
| 00007 | AND LD | - |
| 00008 | OUT | 10000 |

用右边的方法，可以最多组合八个块。用第一种方法，可以组合的快的数目没有限制。

下面的梯形图需要用 OR LOAD 指令以转换为助记符形式，因为有三对串联的条件相互并联。



第一对条件转换为带指定位操作数的 LOAD，然后与其它条件进行逻辑与。可以先对起始两块进行编码，后面接着 OR LOAD、最后一块和另一个 OR LOAD；或者可以首先进行三块的编码，后面接着两个 OR LOAD。两种方法的助记符代码表示在下面。

| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND DOT | 00001 |
| 00002 | LD NOT | 00002 |
| 00003 | AND NOT | 00003 |
| 00004 | OR LD | - |
| 00005 | LD | 00004 |
| 00006 | AND | 00005 |
| 00007 | OR LD | - |
| 00008 | OUT | 10001 |

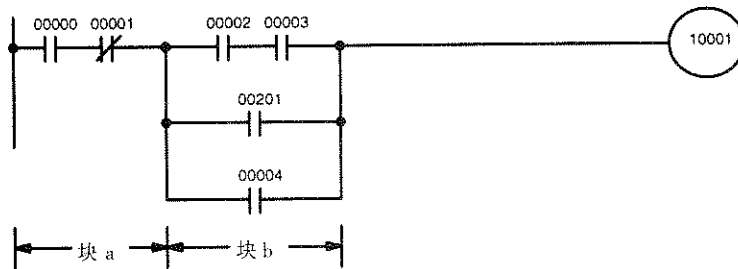
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND DOT | 00001 |
| 00002 | LD NOT | 00002 |
| 00003 | AND NOT | 00003 |
| 00004 | LD | 00004 |
| 00005 | AND | 00005 |
| 00006 | OR LD | - |
| 00007 | OR LD | - |
| 00008 | OUT | 10000 |

同样地，用右边的方法，可以最多组合八个块。用第一种方法，可以组合的块的数目没有限制。

组合 AND LOAD 和 OR LOAD

当要使用 AND LOAD 和 OR LOAD 时，上面介绍的两种方法都可以使用，只要组合的块的数目不超过八个。

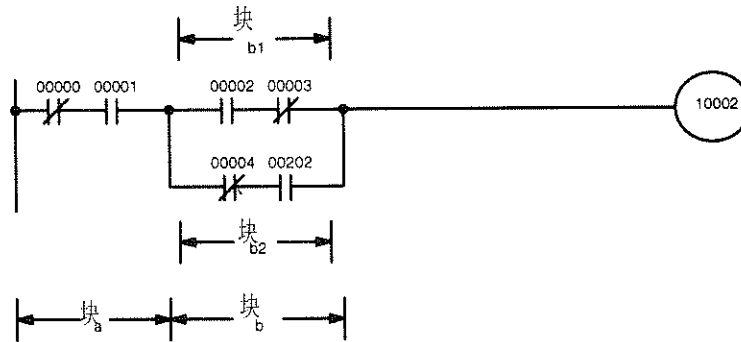
下面所示的梯形图只包含两个逻辑块。不必要进一步分割块 b，因为它可以只用 AND 和 OR 进行编码。



| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND DOT | 00001 |
| 00002 | LD | 00002 |
| 00003 | AND | 00003 |
| 00004 | OR | 00201 |
| 00005 | OR | 00004 |
| 00006 | AND LD | - |
| 00007 | OUT | 10001 |

虽然下面的图与上面的一个相似，如果不把图中的块 b 分割成用 OR LOAD 组合起来的两块的话，就不能进行编码。在此例中，三个块首先进行编码，然后用 OR LOAD 组合最后两个块，接着用 AND LOAD 将 OR LOAD 产生的执行条件与块 a 的执行条件进行组合。

当在已编码的逻辑块的末尾将逻辑块指令组合在一起时，它们必须如下所示，按相反的方向进行编码，即，最后两块的逻辑块指令首先进行组合编码，接着，将第一个组合逻辑块指令产生的执行条件与倒数第三个逻辑块的执行条件组合，一直到要组合的第一个逻辑块。



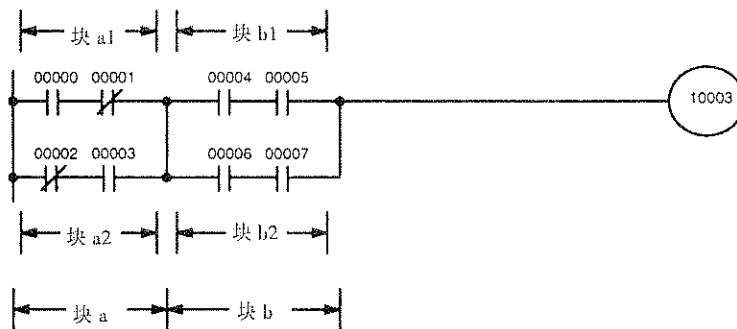
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD NOT | 00000 |
| 00001 | AND | 00001 |
| 00002 | LD | 00002 |
| 00003 | AND NOT | 00003 |
| 00004 | LD NOT | 00004 |
| 00005 | AND | 00202 |
| 00006 | OR LD | - |
| 00007 | AND LD | - |
| 00008 | OUT | 10002 |

复杂的梯形图

当决定需要什么逻辑块指令来对梯形图进行编码时，有时需要将梯形图分割成大的块，然后再不断地分割大的块，直到形成的逻辑块可以不需要逻辑块指令进行编码。然后对这些块进行编码，首先组合小的块，然后组合大的块。可以用 AND LOAD 或 OR LOAD 组合这些块，即 AND LOAD 或 OR LOAD 总是组合已存在的最后两个执行条件，不管执行条件是从单独的条件得到还是从逻辑块，或是从先前的逻辑块指令产生的。

当遇到复杂的梯形图时，从左上解开始编码，先向下再横向。这意味着，只要可能，先对 OR LOAD 编码，然后再对 AND LOAD 编码。

下面的梯形图必须分割成两块，然后各个块再分割成两块，这样才可以对它进行编码。如下所示，块 a 和块 b 之间需要一个 AND LOAD。然而，在可以使用 AND LOAD 之前，必须使用 OR LOAD 组合两侧的上层块和下层块，即，组合 a_1 和 a_2 ; b_1 和 b_2 。



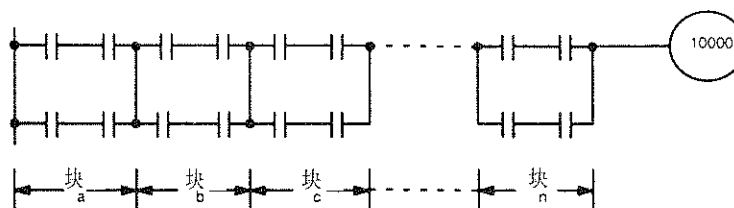
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND NOT | 00001 |
| 00002 | LD NOT | 00002 |
| 00003 | AND | 00003 |
| 00004 | OR LD | - |
| 00005 | LD | 00004 |
| 00006 | AND | 00005 |
| 00007 | LD | 00006 |
| 00008 | AND | 00007 |
| 00009 | OR LD | - |
| 00010 | AND LD | - |
| 00011 | OUT | 10003 |

块 a1 与 a2

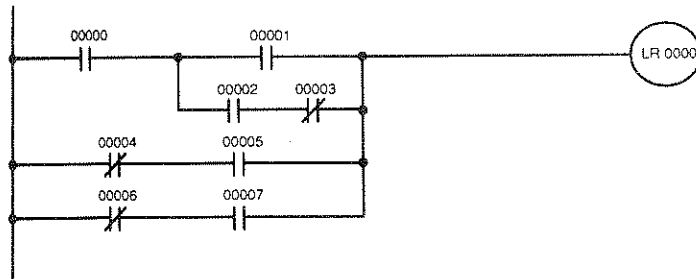
块 b1 与 b2

块 a 与 b

下面一种梯形图如果各块按如下顺序可以容易地编码：先从上到下，然后从左到右。在下面的梯形图中，块 a 和块 b 可以如上所示使用 AND LOAD 组合，然后对块 c 进行编码，再使用第二个 AND LOAD 对第一个 AND LOAD 产生的执行条件与它进行组合。然后对块 d 进行编码，再使用第三个 AND LOAD 对第二个 AND LOAD 产生的执行条件与它进行组合。如此进行，直到块 n。

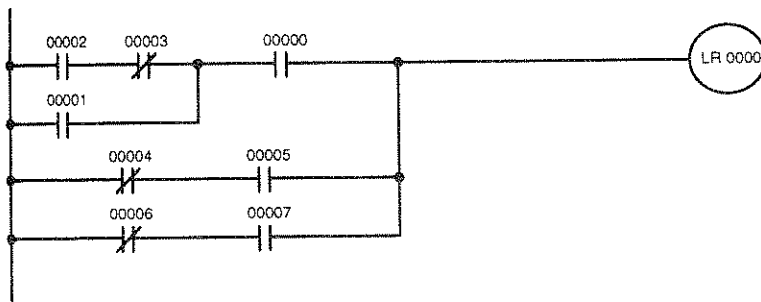


下面的梯形图需要 OR LOAD 接着一个 AND LOAD, 以对上面的三块进行编码, 然后再用两个 OR LOAD 以完成助记符代码。



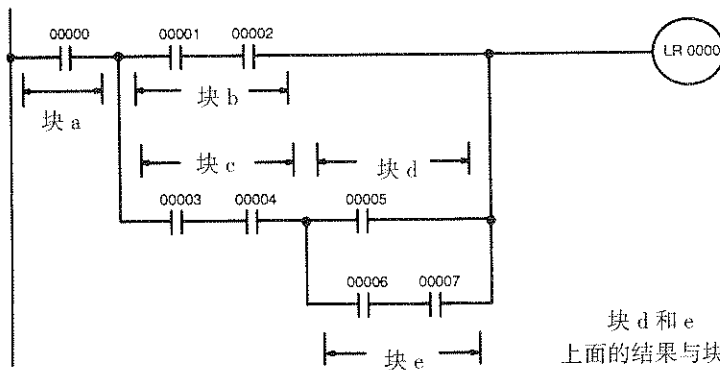
| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | LD | 00001 |
| 00002 | LD | 00002 |
| 00003 | AND NOT | 00003 |
| 00004 | OR LD | -- |
| 00005 | AND LD | -- |
| 00006 | LD NOT | 00004 |
| 00007 | AND | 00005 |
| 00008 | OR LD | -- |
| 00009 | LD NOT | 00006 |
| 00010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

虽然程序将按所编写的执行, 但是此梯形图可以写成如下形式, 以省略第一个 OR LOAD 和 AND LOAD, 并简化程序, 节省存储空间。



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

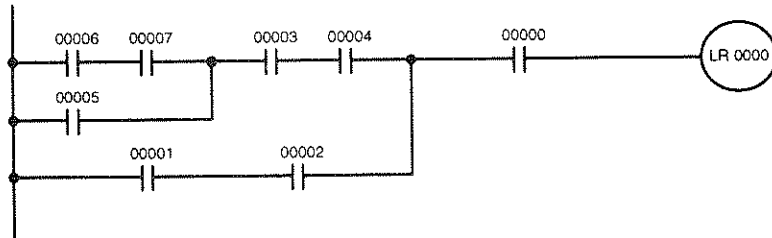
下面的梯形图需要五个块, 首先对它们编码, 然后使用 OR LOAD 和 AND LOAD, 从最后两块开始, 对它们进行组合然后向右。程序地址为 00008 和 OR LOAD 对块 d 和 e 进行组合, 下面的 AND LOAD 对所产生的执行条件与块 c 的执行条件进行组合, 等等。



块 d 和 e
上面的结果与块 c
上面的结果与块 b
上面的结果与块 a

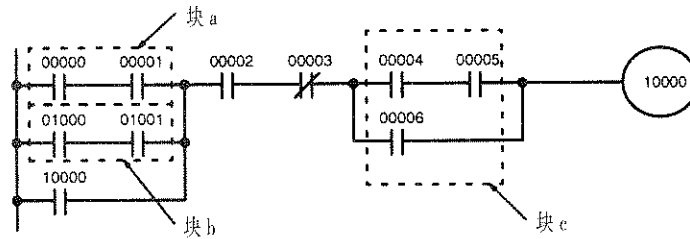
| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD | 00000 |
| 00001 | LD | 00001 |
| 00002 | AND | 00002 |
| 00003 | LD | 00003 |
| 00004 | AND | 00004 |
| 00005 | LD | 00005 |
| 00006 | LD | 00006 |
| 00007 | AND | 00007 |
| 00008 | OR LD | -- |
| 00009 | AND LD | -- |
| 00010 | OR LD | --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

同样地,这个梯形图可以画成如下形式,以简化程序结构和代码,并节省存储空间。

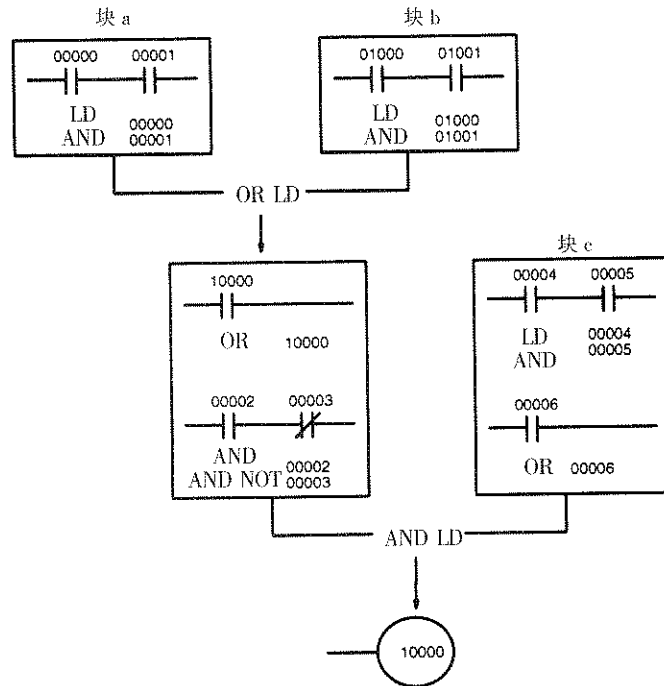


| 地址 | 指令 | 操作数 |
|-------|-------|---------|
| 00000 | LD | 00006 |
| 00001 | AND | 00007 |
| 00002 | OR | 00005 |
| 00003 | AND | 00003 |
| 00004 | AND | 00004 |
| 00005 | LD | 00001 |
| 00006 | AND | 00002 |
| 00007 | OR LD | -- |
| 00008 | AND | 00000 |
| 00009 | OUT | LR 0000 |

下面最后一个例子初看起来可能很复杂,但是它只要使用两个逻辑块指令就可以编码。梯形图如下所示。



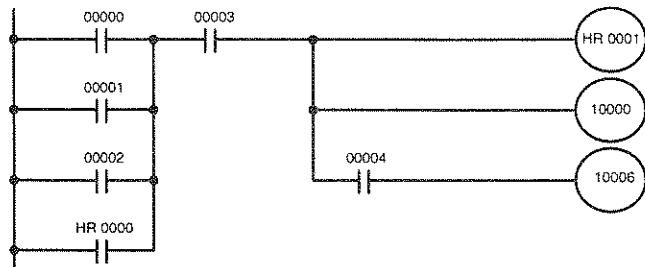
第一个逻辑块指令用来组合块 a 和 b 的执行条件, 第二个用来将块 c 的执行条件同常闭条件 IR00003 的执行条件相结合。梯形图的其余部分可以用 OR, AND 和 AND NOT 编码, 这个过程的逻辑流程图和结果代码显示如下。



| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND | 00001 |
| 00002 | LD | 01000 |
| 00003 | AND | 01001 |
| 00004 | OR LD | -- |
| 00005 | OR | 10000 |
| 00006 | AND | 00002 |
| 00007 | AND NOT | 00003 |
| 00008 | LD | 00004 |
| 00009 | AND | 00005 |
| 00010 | OR | 00006 |
| 00011 | AND LD | -- |
| 00012 | OUT | 10000 |

4-3-7 多右侧指令编码

如果有一条以上以同样的执行条件执行的右侧指令,它们要在指令行上的最后的条件之后连续地编码。在下面的例子中,最后一条指令行还另外包含一个与 IR00004 做 AND 操作相对应的条件。



| 地址 | 指令 | 操作数 |
|-------|-----|---------|
| 00000 | LD | 00000 |
| 00001 | OR | 00001 |
| 00002 | OR | 00002 |
| 00003 | OR | HR 0000 |
| 00004 | AND | 00003 |
| 00005 | OUT | HR 0001 |
| 00006 | OUT | 10000 |
| 00007 | AND | 00004 |
| 00008 | OUT | 10006 |

4-3-8 分支指令行

当一条指令行分支成两行或更多行时,有时需要使用联锁或 TR 位以保持分支点上已存在的执行条件。这是因为指令行一直执行到右侧指令,然后再能返回到分支点,执行分支行上的指令。如果一个条件存在于分支点后面的指令行上,在此期间执行条件可能会改变,从而不能正确地执行。下图说明了这一点。在两个图中,都是先执行指令 1,然后返回到分支点,移动到指向指令 2 的分支行上。

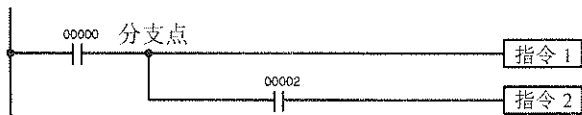


图 A: 正确操作

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

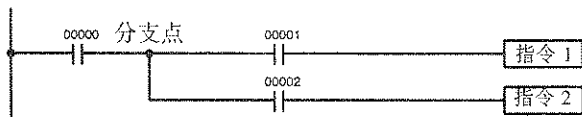


图 B: 不正确操作

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

如果,如图 A 所示,在返回到分支行之前,存在于分支点上的执行条件不会改变(最右边的指令不改变执行条件),那么,分支行将会正确地执行,并不需要特殊的编程手段。

如果,如图 B 所示,在分支点和上面的最后一个指令之间存在一个条件的话,分支点上的执行条件和完成上面的指令行之后的执行条件有时会不同,从而不能保证分支行的正确执行。

有两种编写分支程序的手段,用来保存执行条件。一个是使用 TR 位,另一个是使用联锁(IL(02)/ILC(03))。

TR 位

TR 区域提供 TR0 到 TR7 八个位,用来暂时保存执行条件。如果把 TR 位放到分支点上,当前的执行条件将存储到指定的 TR 位。当返回分支点时,TR 位恢复在程序执行中第一次到达的分支点时执行。

上面的图 B 可以写成下面的形式,以保证正确执行。在助记符代码中,使用 TR 位作为 OUTPUT 指令的操作数,保存分支点的执行条件。当使用同一 TR 位作为 LOAD 指令的操作数,执行完右侧指令后,恢复此执行条件。

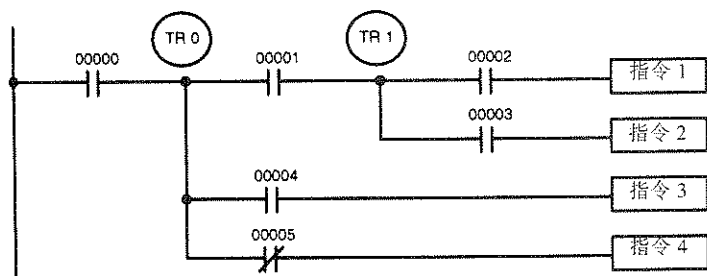


图 B: 用 TR 位改正

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

以实际指令来说,上图是这样的:装载 IR00000 的状态 (LOAD 指令) 以建立初始执行条件。然后用 OUTPUT 指令将此执行条件输出到 TR0, 以保存分支点的执行条件。然后此执行条件与 IR00001 的状态相与, 并相应地执行指令 1。然后重新装载在分支点保存的执行条件 (以 TR0 作为操作数的 LOAD 指令), 它与 IR00002 的状态相与, 并相应地执行指令 2。

下面的例子表示使用两个 TR 位的用法。



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

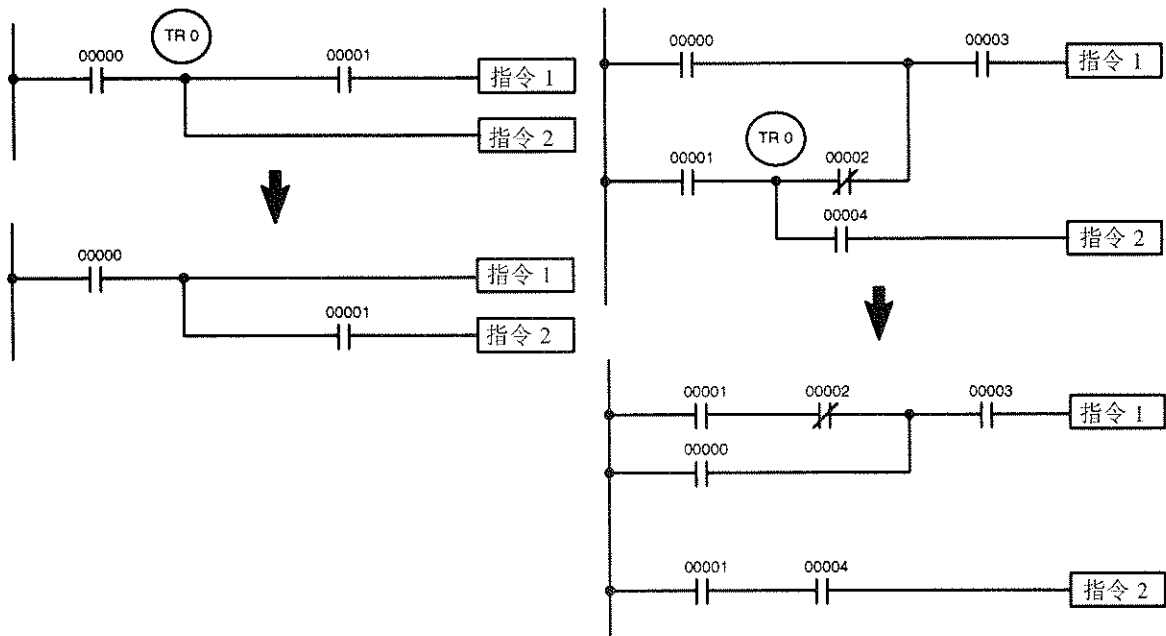
在这个例子中,TR0 和 TR1 用来保存分支点的执行条件。在执行指令 1 之后,保存在 TR1 中的执行条件被装载, 然后与 IR00003 的状态相与。保存在 TR0 中的执行条件被装载两次, 第一次与 IR00004 的状态相与, 第二次与 IR00005 的状态取反相与。

同一 TR 位可以按需要使用任意次,只在要同一指令块中不重复使用同一 TR 位,每当执行返回到母线时,就开始一个新的指令块。

当在一个单独的指令块中,如果有八个以上的分支点需要保存执行条件,则必须使用联锁(后面将介绍)。

当画梯形图时,注意:除非必须,应尽量避免使用 TR 位。通过重新编排梯形图常常可以减少程序所用的指令数,使程序变得容易理解和省去原来需要的 TR 位。在下面一对图中,下面的一种需要较少的指令,并且不需要 TR 位。在第一个例子中,这是通过重新组织指令块部分来达到的;下面的图是通过第二个 OUTPUT 指令分开,使用另一个 LOAD 指令,以产生相应于它的执行条件。

注 虽然简化程序始终是我们关心的问题,但有时指令的执行顺序也很重要。例如,在执行二进制加指令之前,可能需要一个传送指令以将相应的数据放到要用的操作数字中。在重新组织程序进行简化之前,要保证已经考虑过程序的执行顺序。



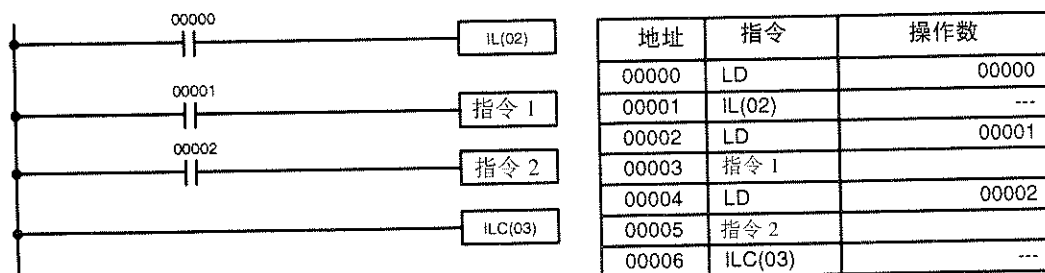
注 只有当使用助记符编程时,才需要由用户输入 TR 位。当直接使用梯形图输入时,则不需要 TR 位,因为它会自动地进行处理。上述关于需要 TR 位的分支点的数目和限制,以及减少编程指令数目的方法的考虑仍然是需要的。

联锁

保存分支点的执行条件的问题也可以通过使用联锁(IL (02))和清联锁(ILC(03))指令来解决。它通过一个特定的执行条件来控制一组指令,从而完全消除分支点的问题。联锁和清联锁指令总是成对使用。

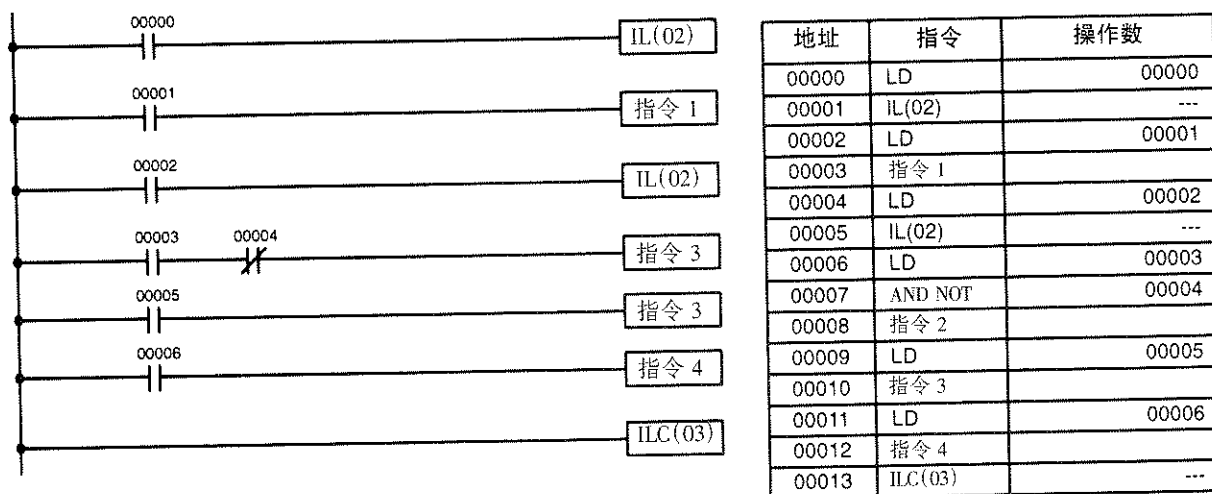
当联锁指令放在一段梯形图程序之前，联锁指令的执行条件将控制清联锁指令出现之前的所有指令。如果联锁指令的执行条件为 OFF，直到下一个清联锁指令出现之前的所有右侧指令都将以 OFF 执行条件来执行，从而复位整个一段梯形图。它对于特殊指令的影响在节 5—12 联锁和清联锁—IL(02) 和 ILC(03) 中说明。

梯形图 B 也可以使用联锁来改正。这里，指向分支点的条件放到联锁指令行上，所有从分支点引出的指令行都作为独立的指令行来编写，并加上另外一条清联锁指令的指令行。清联锁的指令行不允许有条件。注意联锁和清联锁都不需要操作数。



在上面修改过的 B 图中，如果 IR00000 为 ON，IR00001 和 IR00002 的状态将分别决定指令 1 和指令 2 的执行条件。因为 IR00000 为 ON，它与这些位的状态相与将产生相同的结果。如果 IR00000 为 OFF，联锁指令将为指令 1 和指令 2 产生 OFF 执行条件，然后将执行清联锁指令之后的指令行。

如下图所示，一个指令块中可以使用一个以上的联锁指令；各个指令直到下一个清联锁指令之前有效。



上图中,如果 IR00000 为 OFF(即,如果第一个联锁指令的执行条件为 OFF),指令 1 到指令 4 将以 OFF 执行条件执行,并接着执行清联锁指令后面的指令。如果 IR00000 为 ON,IR00001 的状态将作为指令 1 的执行条件,然后 IR00002 的状态将形成第二个联锁指令的执行条件。如果 IR00002 为 OFF,指令 2 到指令 4 将以 OFF 执行条件执行,如果 IR00002 为 ON,IR00003,IR00005 和 IR00006 将决定新的指令行的第一执行条件。

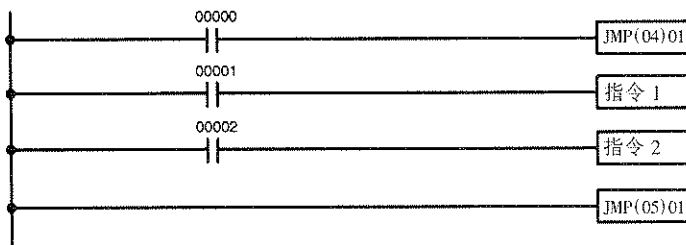
4-3-9 跳转

可以按照一个指定的执行条件,跳过某一指定的程序段。虽然这与当联锁指令的执行条件为 OFF 时发生的情况类似,但是使用跳转指令时,所有指令的操作数都可以保持它的状态。因此跳转可以用来控制需要保持输出的设备,例如,气动装置和液压传动设备;而联锁可以用来控制不需要保持输出的设备,例如,电子仪器。

通过使用跳转(JMP(04))和跳转结束(JME(05))指令来产生跳转。如果跳转指令的执行条件为 ON,程序将正常执行,如同跳转不存在一样。如果跳转指令的执行条件为 OFF,程序立即跳转到跳转结束指令执行,而不改变跳转和跳转结束指令之间的任何状态。

所有的跳转和跳转结束指令都分配一个 00~99 之间的编号。有两种类型的跳转。所使用的跳转号决定跳转的类型。

一个跳转仅可以使用一次 01~99 之间的跳转号,即,一个跳转号只能在跳转指令中使用一次和在跳转结束指令中使用一次。当被分配某一个跳转号的跳转指令执行时,指令的执行立刻跳转到具有相同跳转号的跳转结束指令,如同它们之间的所有指令不存在一样。用 TR 位和联锁中的图 B 可以使用跳转指令来重画成如下形式。虽然 01 已经作为一个跳转号使用,01~99 之间的任何数只要它没有在程序的其它部分使用都可以使用,跳转和跳转结束不需要其它操作数,指向跳转结束的指令行上不能有任何条件。



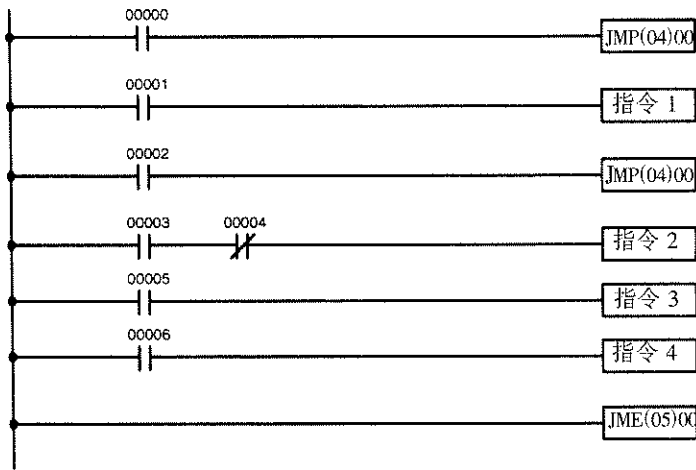
| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | JMP(04) | 01 |
| 00002 | LD | 00001 |
| 00003 | 指令 1 | |
| 00004 | LD | 00002 |
| 00005 | 指令 2 | |
| 00006 | JMP(05) | 01 |

图 B:用 JUMP 指令更正

当 IR00000 为 OFF 时,图 B 的这种形式较其它的两种需要的执行时间较短。

另外一种类型的跳转用跳转号 00 生成。跳转号 00 可以生成多次跳转, 跳转号为 00 的跳转指令可以连续多次使用, 几个跳转之间不需要使用跳转号 00 的跳转结束指令。甚至所有的跳转 00 指令都可以将程序的执行转移到同一跳转结束 00, 即对程序中的所有跳转 00 指令只需要一条跳转结束 00 指令。当用 00 作为跳转指令的跳转编号时, 程序跳转到下一个使用跳转号为 00 的跳转结束指令后面的一条指令。虽然, 如在所有的跳转中一样, 在跳转 00 和跳转结束 00 之间的所有状态不改变, 所有指令不执行, 但是程序必须搜寻下一个跳转结束 00 指令, 这会略微增加执行时间。

含有多条跳转 00 指令和一条跳转结束 00 指令的程序的执行与用联锁指令的一段程序的执行相似。下图与上面联锁中使用的例子相同, 除了它是使用跳转来重画的。这个图的执行与上面介绍的图执行不同。(例, 前面的图中, 联锁将使联锁所包含的某些部分复位, 然而, 跳转不影响跳转和跳转结束指令之间的任何位的状态)。



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

4-4 控制位的状态

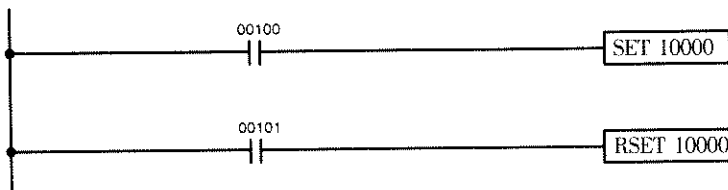
通常, 有七条基本指令可以用来控制个别的位状态。它们是输出, 输出非, 置位, 复位, 微分上升, 微分下降和保持指令。所有这些指令在指令行中作为最后一条指令出现, 并有一个位地址操作数。虽然细节将在 5-9 位控制指令中给出, 由于它们在大多数程序中的重要性, 这些指标(除了已经介绍过的输出和输出非)在这里给予介绍。虽然这些指令用来将 IR 区的输出位变为 ON 或 OFF(即向外部设备发送或停止输出信号), 它们也用来控制 IR 区或其它区的其它位的状态。

4-4-1 SET(置位)和 RESET(复位)

置位和复位指令与输出和输出非指令很相似, 除了它们只在 ON 执行条件下改变它们的操作数位状态。当执行条件为 OFF 时, 这两条指令都不影响其操作数位的状态。

当执行条件变为 ON 时, 置位将把操作数位变 ON, 但是不象输出指令, 当执行条件变为 OFF 时, 置位并不把操作数位变为 OFF。当执行条件变为 ON 时, 复位将把操作数位变为 OFF, 但是不象输出非指令, 当执行条件变为 OFF 时复位并不把操作数位变 ON。

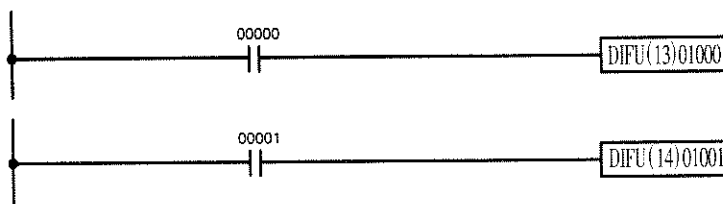
在下面的例子中, 当 IR00100 变为 ON 时, IR10000 将变为 ON, 并保持为 ON 直到 IR00101 变为 ON, 而不管 IR00100 的状态, 当 IR00101 变为 ON 时, 复位将把 IR10000 变为 OFF。



| 地址 | 指令 | 操作数 |
|-------|------|-------|
| 00000 | LD | 00100 |
| 00001 | SET | 10000 |
| 00002 | LD | 00101 |
| 00003 | RSET | 10000 |

4-4-2 微分上升和微分下降

微分上升和微分下降指令用来使操作数位每次在一个循环内为 ON。微分上升指令在其执行条件从 OFF 变为 ON 后使操作数位在一个循环内为 ON; 微分下降指令在其执行条件从 ON 变为 OFF 后使操作数位在一个循环内为 ON。这两条指令都只需要一行助记符代码。



| 地址 | 指令 | 操作数 |
|-------|----------|-------|
| 00000 | LD | 00000 |
| 00001 | DIFU(13) | 01000 |

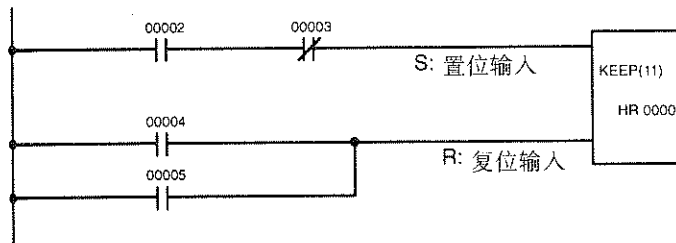
| 地址 | 指令 | 操作数 |
|-------|----------|-------|
| 00000 | LD | 00001 |
| 00001 | DIFD(14) | 01001 |

这里, 在 IR00000 变为 ON 后, IR01000 将在一个循环内为 ON。下一次执行 DIFU(13)01000 时, 不管 IR00000 的状态如何, IR01000 将变为 OFF。对微分下降指令, 当 IR00001 变为 OFF (IR01001 保持为 OFF, 直到以后) 后, IR01001 将在一个循环内为 ON, 当下一次执行 DIFD(14) 时, IR01001 变为 OFF。

4-4-3 KEEP(保持)

保持指令用以根据二个执行条件保持操作数位的状态。为此保持指令与两个指令行连接。当第一个指令行的末尾的执行条件为 ON 时, 保持指令的操作数位变为 ON。当第二条指令行的末尾的执行条件为 ON 时, 保持指令的操作数位变为 OFF。即使它位于梯形图的联锁段内, KEEP 指令的操作数位将保持其 ON 或 OFF 状态。

在下面的例子中,当 IR00002 为 ON, IR00003 为 OFF 时, HR0000 将变为 ON。HR0000 将保持为 ON 直到 IR00004 或 IR00005 变为 ON。使用保持指令,如同所有需要一条以上指令行的指令,指令行在它们控制的指令之前编码。



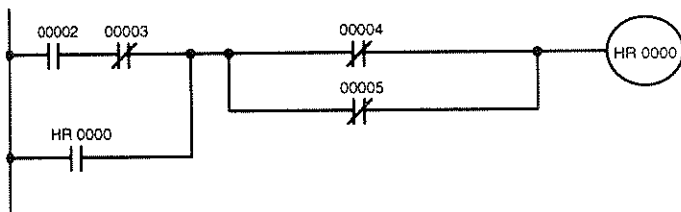
| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00002 |
| 00001 | AND NOT | 00003 |
| 00002 | LD | 00004 |
| 00003 | OR | 00005 |
| 00004 | KEEP(11) | HR 0000 |

4-4-4 自保持位(封口)

虽然保持指令可用来产生自保持位,有时需要以另一种方式来产生自保持位,从而使它们在程序的联锁段中时可以变为 OFF。

为了产生自保持位,用一个输出指令的操作数位作为同一输出指令的一个或置位条件。这样此输出指令的操作数位可以保持 ON 或 OFF,直到其他位出现变化。在输出指令前至少要用一个其他条件作为复位,没有这个复位位就无法控制此输出指令的操作数位。

上图的 KEEP 指令可以改写如下。这两个梯形图只是当处在联锁程序段中,并且联锁的执行条件为 ON 时有一点差别。这里如同使用保持指令时相同。使用两个复位位,即在 IR00004 或 IR00005 为 ON 时, IR0000 能够复位。



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

4-5 工作位(内部继电器)

在编程中,组合条件以直接产生执行条件常常是非常困难的。然而,通过特定的位间接地触发其它指令,很容易地克服这些困难。使用工作位很容易实现这种编程方法。有时,为某种目的需要整个字。这些字称为工作字。

PC 不传送也不接受工作位。如上所述人它们为方便编程由程序员选择。I/O 位及其它专用位不能用作工作位。IR 区中未被分配作 I/O 的所有的位。以及 AR 区一些特定的未使用的位,可以用作工作位。注意记录如何和在何处使用了工作位,这可以帮助进行程序设计和编写,还可以帮助调试工作。

工作位的应用

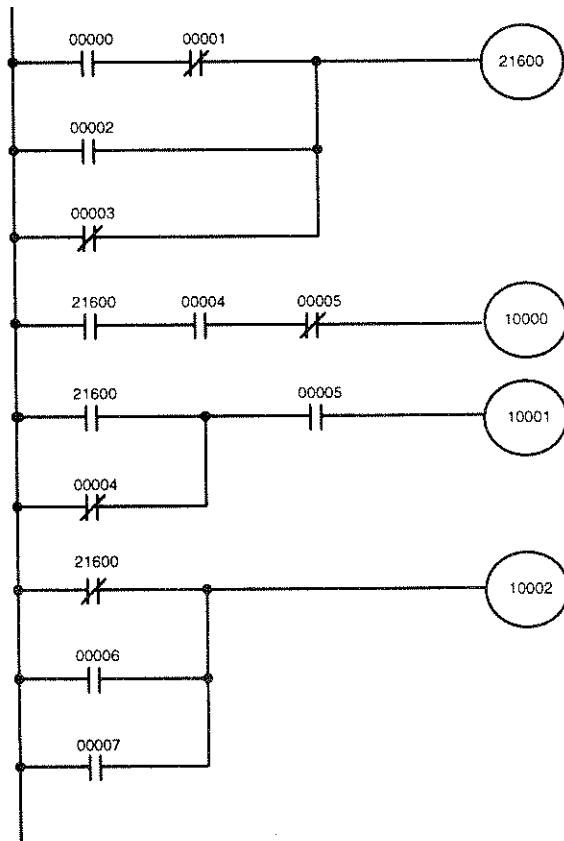
本节后面给出的例子表示了使用工作位的两种最常用的方法。这些例子是使用工作位无数种方法中的范例。当在编写控制动作遇到困难时,应考虑到工作位以及怎样使用它们以简化编程。

工作位常和输出、输出非、微分上升、微分下降、保持指令一起使用。工作位首先作为这些指令的操作数使用。这样,以后它可以用来作为决定其它指令如何执行的条件。工作位还可以同其它指令,如移位寄存器指令(SFT(10))一起使用。使用移位寄存器指令的工作字或工作位的例子在 5—16 移位寄存器—SFT(10) 中给出。

虽然不总是被特定称为工作位,第五章 指令集的例子中使用的许多位作为工作位。理解这些位的用法对于有效地编程是必要的。

减少复杂的条件

当一定的组合条件被重复地使用以同其它条件组合时,可以使用工作位简化程序。在下面的例子中,IR00000, IR00001, IR00002 和 IR00003 组成一个逻辑块,在 IR21600 的状态中保存了执行条件的结果。然后 IR21600 同各种其它条件组合以决定 IR10000, IR10001 和 IR10002 的输出条件,即使分配给这些位的输出变 ON 或 OFF。

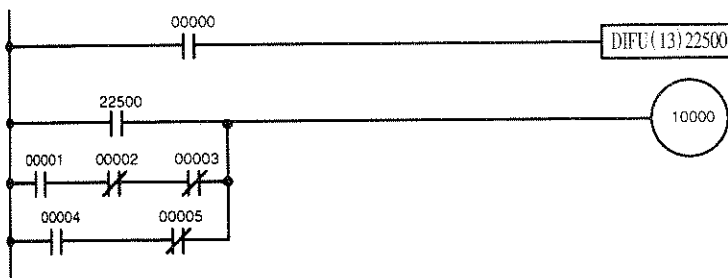


| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | AND NOT | 00001 |
| 00002 | OR | 00002 |
| 00003 | OR NOT | 00003 |
| 00004 | OUT | 21600 |
| 00005 | LD | 21600 |
| 00006 | AND | 00004 |
| 00007 | AND NOT | 00005 |
| 00008 | OUT | 10000 |
| 00009 | LD | 21600 |
| 00010 | OR NOT | 00004 |
| 00011 | AND | 00005 |
| 00012 | OUT | 10001 |
| 00013 | LD NOT | 21600 |
| 00014 | OR | 00006 |
| 00015 | OR | 00007 |
| 00016 | OUT | 10002 |

微分条件

如果一条指令的部分执行条件需要(不是全部)进行微分处理,也可以使用工作位。在本例子中,只要 IR00101 为 ON,同时 IR00002 和 IR00003 为 OFF,或者 IR00004 为 ON,同时 IR00005 为 OFF,IR10000 必然持续地保持为 ON。每当 IR00000 变为 ON,它会在一个循环内保持 ON(除非上述条件之一使之连续地保持 ON)。

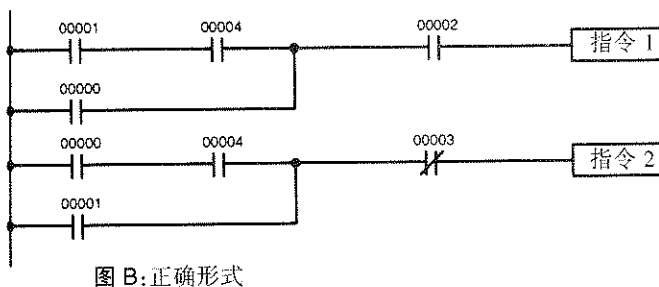
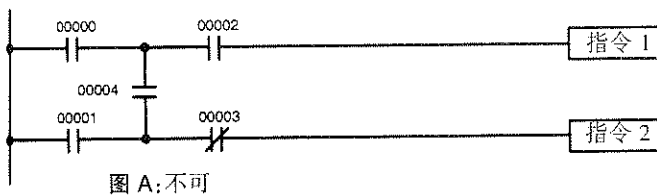
通过使用 IR22500 为工作位作为微分上升指令(DIFU(13))的操作数编程。可以容易地实现上述功能。当 IR00000 变为 ON,IR22500 将在一个循环内为 ON,然后在下一个循环中,DIFU(13)使之变为 OFF。假设控制 IR10000 的其它条件不将它保持为 ON,IR22500 工作位将只在一个循环内使 IR10000 为 ON。



| 地址 | 指令 | 操作数 |
|-------|----------|-------|
| 00000 | LD | 00000 |
| 00001 | DIFU(13) | 22500 |
| 00002 | LD | 22500 |
| 00003 | LD | 00001 |
| 00004 | AND NOT | 00002 |
| 00005 | AND NOT | 00003 |
| 00006 | OR LD | --- |
| 00007 | LD | 00004 |
| 00008 | AND NOT | 00005 |
| 00009 | OR LD | --- |
| 00010 | OUT | 10000 |

4—6 编程注意事项

条件的串联和并联的数只要不超出 PC 的存储容量,可以不受限制。因此按需要使用条件来画一张清楚的梯形图。虽然在一个指令行内可以画很复杂的梯形图,但是在两个指令行之间的竖线上不能有任何条件。例如下图 A 是不允许的,应画成图 B。助记符只用于图 B。对图 A 进行助记符编码是不可能的。



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

任何一个特殊位分配给条件的次数不受限制,因此在程序中按需要使用它们以简化程序。复杂的程序常常是由于试图减少位的使用次数而造成的。

除了那些不允许有条件的指令(如,清联锁和跳转结束,见下述),每条指令行必须具有至少一个条件以决定右边的指令的执行条件。同样地,下面的图 A 必须画成图 B。如果一条指令必须连续地执行(如果程序执行时,一个输出必须总要保持 ON),可以使用 SR 区的常接通标志(SR25313)。

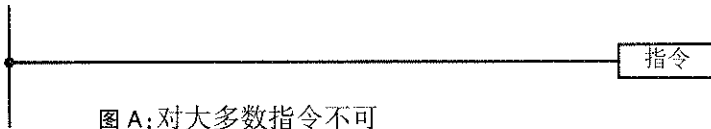


图 A:对大多数指令不可



图 B:正确形式

| 地址 | 指令 | 操作数 |
|-------|----|-------|
| 00000 | LD | 25313 |
| 00001 | 指令 | |

这条规则有几个例外,包括清联锁、跳转结束和步指令。每个这样的指令都是指令对的第二个指令,由第一个指令的执行条件控制,指向这些指令的指令行上不能放置条件。详情请参照第五章 指令集。

当画梯形图时,记住指令的编号是很重要的,因需要将它们输入。在下面的图 A 中,上面和下面的指令行需要用 OR LOAD 指令来组合。这可以通过重画成图 B 以避免使用 AND LOAD 或 OR LOAD。详情请参照 5-7-2 AND LOAD 和 OR LOAD。

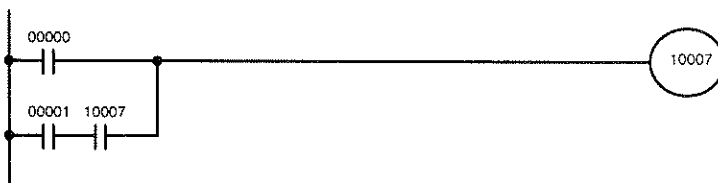


图 A

| 地址 | 指令 | 操作数 |
|-------|-------|-------|
| 00000 | LD | 00000 |
| 00001 | LD | 00001 |
| 00002 | AND | 10007 |
| 00003 | OR LD | --- |
| 00004 | OUT | 10007 |



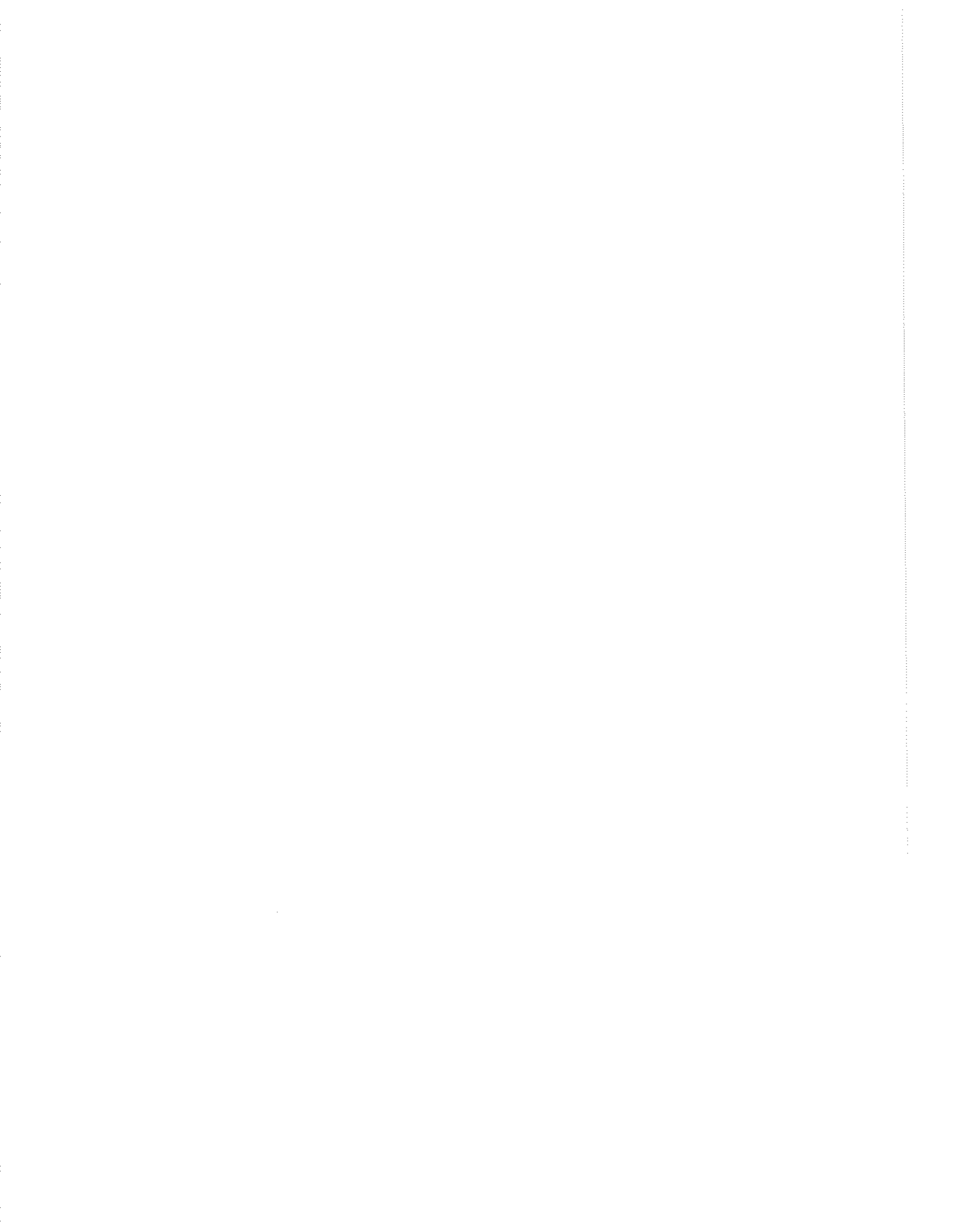
图 B

| 地址 | 指令 | 操作数 |
|-------|-----|-------|
| 00000 | LD | 00001 |
| 00001 | AND | 10007 |
| 00002 | OR | 00000 |
| 00003 | OUT | 10007 |

4—7 程序执行

当启动程序执行时,CPU 从上到下扫描程序,检查所有的条件,并执行移到母线的指令。按正确顺序放置指令很重要,例如,在一个字作为一条指令的操作数之前,要把需要的数据移到这个字中。记住,一条指令行执行右边的最终指令完毕后,才执行从第一条指令行分支、到其它右侧最终指令的指令行。

程序执行只是 CPU 一个循环时间内完成的任务之一。详情请参照第七章 *CPU 工作和处理时间*。



第 5 章 指令集

CQM1H 拥有丰富的编程指令系统,可以方便地处理复杂的控制过程的编程。本章分别解释了各指令,并给出了梯形图符号、数据区和各指令所使用的标志。

这些 PC 提供的大量指令按指令组分成下列章节。这些组包括梯形图指令,具有固定的功能代码的指令以及设置指令。

有些指令,如定时器和计数器指令,用来控制其它指令的执行。例如,当定时器所设定的时间到达后,定时器的完成标志可以用来置某位为 ON。虽然其它指令常常用来通过输出指令控制输出位,它们也可以用来控制其它指令的执行。因此本手册的例子中的输出指令通常可以用其它指令来取代,以实现特定的应用程序,而不是直接控制输出位。

| | | |
|--------|--|-----|
| 5-1 | 注意事项 | 201 |
| 5-2 | 指令格式 | 201 |
| 5-3 | 数据区、定义符值和标志 | 201 |
| 5-4 | 微分指令 | 203 |
| 5-5 | 扩展指令 | 203 |
| 5-6 | 编码右侧指令 | 205 |
| 5-7 | 指令表 | 208 |
| 5-7-1 | 用固定功能代码的指令 | 208 |
| 5-7-2 | 扩展指令 | 209 |
| 5-7-3 | 按字母顺序排列的助记符表 | 209 |
| 5-8 | 梯形图指令 | 213 |
| 5-8-1 | LOAD, LOAD NOT, AND, AND NOT, OR 和 OR NOT | 213 |
| 5-8-2 | AND LOAD 和 OR LOAD | 214 |
| 5-9 | 位控制指令 | 214 |
| 5-9-1 | 输出和输出非——OUT 和 OUT NOT | 214 |
| 5-9-2 | 置位和复位——SET 和 RESET | 215 |
| 5-9-3 | 保持——KEEP(11) | 216 |
| 5-9-4 | 微分上升和微分下降——DIFU(13)和 DIFD(14) | 217 |
| 5-10 | 空操作——NOP(00) | 218 |
| 5-11 | 结束——END(01) | 218 |
| 5-12 | 联锁和联锁清除——IL(02)和 ILC(03) | 218 |
| 5-13 | 跳转和跳转结束——JMP(04)和 JME(05) | 220 |
| 5-14 | 用户错误指令: 错误报警与复位——FAL(06)和严重错误报警——FALS(07) | 221 |
| 5-15 | 步指令: 步定义和步启动——STEP(08)/SNXT(09) | 222 |
| 5-16 | 定时器和计数器指令 | 224 |
| 5-16-1 | 定时器——TIM | 225 |
| 5-16-2 | 计数器——CNT | 226 |
| 5-16-3 | 可逆计数器——CNTR(12) | 227 |
| 5-16-4 | 高速定时器——TIMH(15) | 228 |
| 5-16-5 | 累加定时器——TTIM(—) | 229 |
| 5-16-6 | 间隔定时器——STIM(69) | 231 |
| 5-16-7 | 寄存器比较表——CTBL(63) | 232 |
| 5-16-8 | 模式控制——INI(61) | 244 |
| 5-16-9 | 读高速计数器 PV——PRV(62) | 246 |
| 5-17 | 移位指令 | 249 |
| 5-17-1 | 移位寄存器——SFT(10) | 249 |
| 5-17-2 | 字移位——WSFT(16) | 250 |
| 5-17-3 | 算术左移位——ASL(25) | 251 |
| 5-17-4 | 算术右移位——ASR(26) | 251 |
| 5-17-5 | 循环左移——ROL(27) | 252 |
| 5-17-6 | 循环右移——ROR(28) | 252 |
| 5-17-7 | 一位数字左移——SLD(74) | 253 |

| | | |
|---------|-------------------------|-----|
| 5-17-8 | 一位数字右移——SRD(75) | 254 |
| 5-17-9 | 可逆移位寄存器——SFTR(84) | 254 |
| 5-17-10 | 异步移位寄存器——ASFT(17) | 256 |
| 5-18 | 数据传送指令 | 257 |
| 5-18-1 | 传送——MOV(21) | 257 |
| 5-18-2 | 传送非——MVN(22) | 258 |
| 5-18-3 | 块传送——XFER(70) | 259 |
| 5-18-4 | 块设定——BSET(71) | 260 |
| 5-18-5 | 数据交换——XCHG(73) | 261 |
| 5-18-6 | 单字分配——DIST(80) | 261 |
| 5-18-7 | 数据收集——COLL(81) | 263 |
| 5-18-8 | 位传送——MOVB(82) | 265 |
| 5-18-9 | 数字传送——MOVD(83) | 266 |
| 5-18-10 | 传送位——XFRB(—) | 267 |
| 5-19 | 比较指令 | 268 |
| 5-19-1 | 比较——CMP(20) | 268 |
| 5-19-2 | 表比较——TCMP(85) | 269 |
| 5-19-3 | 块比较——BCMP(68) | 270 |
| 5-19-4 | 双字长比较——CMPL(60) | 272 |
| 5-19-5 | 多字比较——MCMP(19) | 273 |
| 5-19-6 | 有符号二进制比较——CPS(—) | 274 |
| 5-19-7 | 双字有符号二进制比较——CPSL(—) | 275 |
| 5-19-8 | 区域范围比较——ZCP(—) | 277 |
| 5-19-9 | 双字区域范围比较——ZCPL(—) | 278 |
| 5-20 | 转换指令 | 279 |
| 5-20-1 | BCD转换为二进制——BIN(23) | 279 |
| 5-20-2 | 二进制转换为BCD——BCD(24) | 280 |
| 5-20-3 | 双字BCD转换为双字二进制——BINL(58) | 280 |
| 5-20-4 | 双字二进制转换为双字BCD——BCDL(59) | 281 |
| 5-20-5 | 4—16译码——MLPX(76) | 282 |
| 5-20-6 | 16—4编码——DMPX(77) | 284 |
| 5-20-7 | 7段译码——SDEC(78) | 286 |
| 5-20-8 | ASCII码转换——ASC(86) | 289 |
| 5-20-9 | ASCII转换到十六进制——HEX(—) | 290 |
| 5-20-10 | 比例——SCL(66) | 293 |
| 5-20-11 | 有符号二进制到BCD标度——SCL2(—) | 295 |
| 5-20-12 | BCD到有符号二进制标度——SCL3(—) | 296 |
| 5-20-13 | 时转换为秒——SEC(—) | 298 |
| 5-20-14 | 秒转换为时——HMS(—) | 299 |
| 5-20-15 | 列到行转换——LINE(—) | 300 |
| 5-20-16 | 行到列转换——COLM(—) | 301 |
| 5-20-17 | 2的补码——NEG(—) | 302 |
| 5-20-18 | 双字2的补码——NEGL(—) | 303 |
| 5-21 | BCD计算指令 | 305 |
| 5-21-1 | 置进位——STC(40) | 305 |
| 5-21-2 | 清除进位——CLC(41) | 305 |
| 5-21-3 | BCD加——ADD(30) | 305 |
| 5-21-4 | BCD减——SUB(31) | 306 |
| 5-21-5 | BCD乘——MUL(32) | 308 |
| 5-21-6 | BCD除——DIV(33) | 309 |
| 5-21-7 | 双字BCD加——ADDL(54) | 310 |
| 5-21-8 | 双字BCD减——SUBL(55) | 312 |
| 5-21-9 | 双字BCD乘——MULL(56) | 313 |
| 5-21-10 | 双字BCD除——DIVL(57) | 314 |
| 5-21-11 | 平方根——ROOT(72) | 315 |
| 5-22 | 二进制计算指令 | 316 |
| 5-22-1 | 二进制加——ADB(50) | 316 |
| 5-22-2 | 二进制减——SBB(51) | 317 |

| | | |
|---------|---------------------------|-----|
| 5-22-3 | 二进制乘——MLB(52) | 318 |
| 5-22-4 | 二进制除——DVB(53) | 319 |
| 5-22-5 | 双字二进制加——ADLB(—) | 320 |
| 5-22-6 | 双字二进制减——SBBL(—) | 321 |
| 5-22-7 | 有符号二进制乘——MBS(—) | 323 |
| 5-22-8 | 双字有符号二进制乘——MBSL(—) | 324 |
| 5-22-9 | 有符号二进制除——DBS(—) | 325 |
| 5-22-10 | 双字有符号二进制除——DBSL(—) | 326 |
| 5-23 | 专用数学指令 | 327 |
| 5-23-1 | 求最大值——MAX(—) | 327 |
| 5-23-2 | 求最小值——MIN(—) | 328 |
| 5-23-3 | 求平均值——AVG(—) | 229 |
| 5-23-4 | 求和——SUM(—) | 330 |
| 5-23-5 | 算术处理——APR(—) | 332 |
| 5-24 | 浮点算术指令 | 335 |
| 5-24-1 | 浮点到16位:FIX(—) | 340 |
| 5-24-2 | 浮点到32位:FIXL(—) | 341 |
| 5-24-3 | 16位到浮点:FLT(—) | 342 |
| 5-24-4 | 32位到浮点:FLTL(—) | 343 |
| 5-24-5 | 浮点加: + F(—) | 343 |
| 5-24-6 | 浮点减: - F(—) | 345 |
| 5-24-7 | 浮点乘: * F(—) | 346 |
| 5-24-8 | 浮点除: / F(—) | 347 |
| 5-24-9 | 角度到弧度:RAD(—) | 348 |
| 5-24-10 | 弧度到角度:DEG(—) | 349 |
| 5-24-11 | 正弦:SIN(—) | 350 |
| 5-24-12 | 余弦: COS(—) | 351 |
| 5-24-13 | 正切:TAN(—) | 352 |
| 5-24-14 | 反正弦: ASIN(—) | 353 |
| 5-24-15 | 反余弦: ACOS(—) | 354 |
| 5-24-16 | 反正切: ATAN(—) | 355 |
| 5-24-17 | 平方根: SQRT(—) | 356 |
| 5-24-18 | 指数: EXP(—) | 357 |
| 5-24-19 | 对数: LOG(—) | 359 |
| 5-25 | 逻辑指令 | 360 |
| 5-25-1 | 补码——COM(29) | 360 |
| 5-25-2 | 逻辑与——ANDW(34) | 360 |
| 5-25-3 | 逻辑或——ORW(35) | 361 |
| 5-25-4 | 异或——XORW(36) | 362 |
| 5-25-5 | 异或非——XNRW(37) | 362 |
| 5-26 | 递增/递减指令 | 363 |
| 5-26-1 | BCD值递增——INC(38) | 363 |
| 5-26-2 | BCD值递减——DEC(39) | 363 |
| 5-27 | 子程序指令 | 365 |
| 5-27-1 | 进入子程序——SBS(91) | 365 |
| 5-27-2 | 子程序定义及返回——SBN(92)/RET(93) | 367 |
| 5-28 | 特殊指令 | 367 |
| 5-28-1 | 跟踪内存采样——TRSM(45) | 367 |
| 5-28-2 | 信息显示——MSG(46) | 369 |
| 5-28-3 | I/O刷新——IORF(97) | 370 |
| 5-28-4 | 宏——MCRO(99) | 371 |
| 5-28-5 | 位计数器——BCNT(—) | 373 |
| 5-28-6 | 帧校验和——FCS(—) | 373 |
| 5-28-7 | 故障点检测——FPD(—) | 375 |
| 5-28-8 | 中断控制——INT(89) | 379 |
| 5-28-9 | 设置脉冲——PULS(65) | 380 |
| 5-28-10 | 速度输出——SPED(64) | 382 |
| 5-28-11 | 脉冲输出——PLS2(—) | 385 |
| 5-28-12 | 加速控制——ACC(—) | 387 |
| 5-28-13 | 可变占空率脉冲——PWM(—) | 389 |

| | | | |
|------|-----------|------------------|-----|
| | 5—28—14 | 数据搜索—SRCH(—) | 390 |
| | 5—28—15 | PID 控制—PID(—) | 392 |
| 5—29 | 网络指令 | | 393 |
| | 5—29—1 | 网络发送—SEND(90) | 393 |
| | 5—29—2 | 网络接收—RECV(98) | 397 |
| | 5—29—3 | 发布命令—CMND(—) | 400 |
| 5—30 | 通信指令 | | 403 |
| | 5—30—1 | 接收—RXD(47) | 403 |
| | 5—30—2 | 传送—TXD(48) | 405 |
| | 5—30—3 | 改变串行口设置—STUP(—) | 407 |
| | 5—30—4 | 协议宏—PMCR(—) | 410 |
| 5—31 | 高级 I/O 指令 | | 412 |
| | 5—31—1 | 7 段显示输出—7SEG(88) | 412 |
| | 5—31—2 | 数字开关输入—DSW(87) | 415 |
| | 5—31—3 | 16 键输入—HKY(—) | 419 |
| | 5—31—4 | 10 键输入—TKY(18) | 422 |

5—1 注意事项

在本手册的其余部分,所有的指令都用它们的助记符来表示。例如,输出指令将被称为 OUT; AND LOAD 指令称为 AND LD。如果你不清楚指令的助记符,请参照附录 A 编程指令。

如果一条指令分配了一个功能代码,它将在助记符后面的括号里给出。这些 2 位十进制数的功能代码,用于大多数指令输入 CPU。附录 A 编程指令中也给出了按功能代码排列的指令表。指令清单在 5—7 指令表中给出。

助记符前面的 @ 表示那条指令是微分型指令。微分指令在 5—4 节中解释。

5—2 指令格式

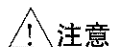
大多数指令相应地有一个或多个操作数。操作数指示或提供指令进行操作的数据。它们有时是作为实际数值(即,作为常数)输入的,但通常是作为包含要使用的数据的字或位的地址。其地址指定为操作数的位称为操作数位;其地址作为操作数的字称为操作数字。在一些指令中,指令中指定的字地址是包含所需数据的多个字的首地址。

每条指令在程序存储器中需要一个或多个字。第一个字是指令字,它指定了指令,并包含指令所需的定义符(下面介绍)或操作数位。指令所需的其它操作数包含在下面的字中,每个操作数为一个字。有的指令最多需要四个字。定义符是与指令相应的一个操作数,包含在与指令本身相同的字中。这些操作数定义了指令,而不是告诉使用什么数据。定义符的一个例子是 TIM/CNT 的编号,它用来在定时器和计数器指令中产生定时器和计数器;同样的还有跳转号(用来定义哪个跳转指令与哪个跳转结束指令相对)。位操作数也包含在与指令本身相同的字中,虽然它们不被认为是定义符。

5—3 数据区、定义符值和标志

在本节中,每个指令的描述包括其梯形图符号,可由其操作数使用的数据区以及可以作为定义符使用的值。数据区的详细情况同样由操作数名称和各个操作数所需的数据类型来规定(即,字或位,以及对于字来说,十六进制数或 BCD)。

规定的数据区内并不是必须所有的地址都可以作为操作数,例如,如果一个操作数需要两个字,数据区的最后一个字则不能作为操作数的第一个字,因为一个操作数的所有字必须在同一个数据区内。其它的特定限制在限制段落中给出。寻址规定、标志和控制位的地址请参照第三章 内存区。



注意

IR 区和 SR 区被认为是独立的数据区。如果一个操作数访问了其中的一个区域,并不意味着同一操作数将访问另一个区域。然而 IR 区和 SR 区之间的边界可以跨过一个单独的操作数,即,IR 区的最后一位可以作为需要一个以上字的操作数,只要 SR 区域可以作为那个操作数。

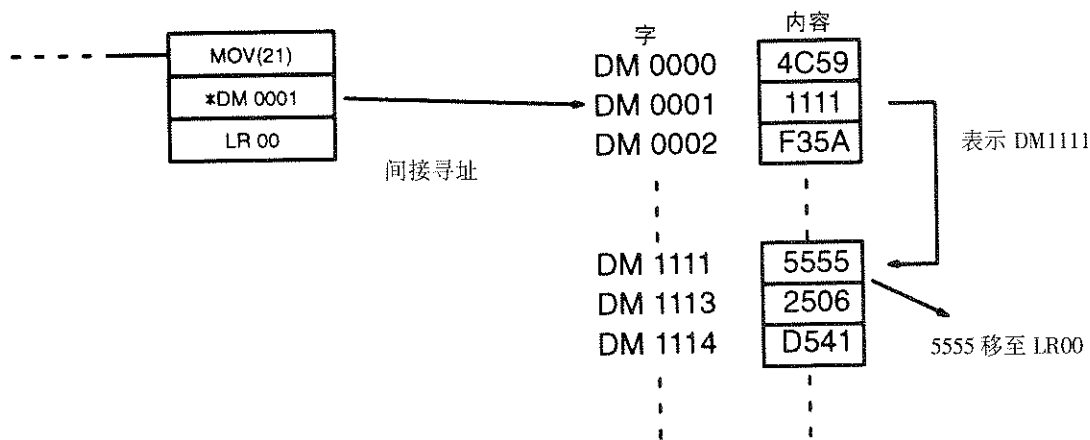
标志项列出了受指令执行影响的标志。这些标志包括下列 SR 区的标志。

| 缩 写 | 名 称 | 位 |
|-----|----------|-------|
| ER | 指令执行错误标志 | 25503 |
| CY | 进位标志 | 25504 |
| GR | 大于标志 | 25505 |
| EQ | 等于标志 | 25506 |
| LE | 小于标志 | 25507 |

ER 是最常用的监视指令执行标志。当 ER 变为 ON 时，它表示执行当前的指令时发生错误。各指令的标志项列出了 ER 变为 ON 的可能原因。如果操作数没有正确地输入，ER 将会变成 ON。当 ER 为 ON 时，指令不执行。附录 B 错误和算术标志操作中给出了指令和标志及其影响的表。

间接寻址

当 DM 区指定为操作数时，可以使用间接寻址。通过在 DM 前加一个星号：*DM 指定间接 DM 寻址。当指定一个间接 DM 寻址时，指定的 DM 字将包含作为指令的操作数使用的数据的 DM 字的地址。例如，如果 *DM0001 被指定为 MOV(21) 的第一个操作数，而 LR00 作为第二个操作数，DM0001 的内容为 1111，DM1111 包含的内容为 5555，值 5555 将被传送到 LR00。



当使用间接寻址时，所需字的地址必须是 BCD 码，它必须指定 DM 区内的一个字。在上面的例子中，*DM0000 的内容必须是 0000~1999 之间的 BCD 码。

定义常数

尽管操作数常常为数据区地址，许多操作数和所有的定义符以常数形式输入。任何一个定义符或操作数的取值范围取决于使用的指令，常数输入也必须是指令所要求的形式，即 BCD 码或十六进制数。

5-4 微分指令

大多数指令都有微分形式或非微分形式。微分指令在指令助记符前有@作为标记。

非微分指令只要它的执行条件为 ON, 每次扫描它的时候都执行。微分指令只在其执行条件从 OFF 变为 ON 时才执行一次。如果执行条件未改变, 或者上次扫描指令时从 ON 变为 OFF, 指令将不会执行。下面的两个例子说明了 MOV(21) 和 @MOV(21) 是如何工作的。这两条指令都是用来将第一个操作数指定的地址中的数据移到第二个操作数指定的地址中。

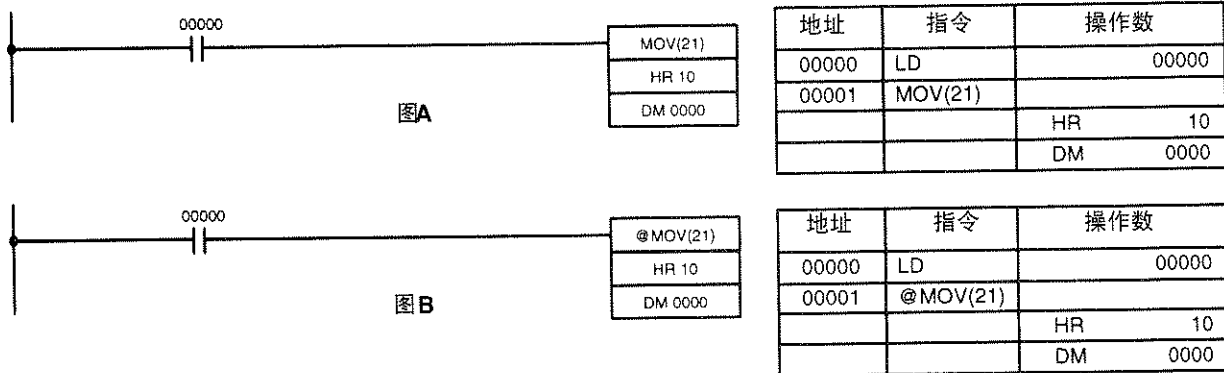


图 A 中, 非微分指令 MOV(21) 根据每次扫描 00000 的条件将 HR10 的内容移到 DM0000 中去。如果扫描时间为 80ms, 00000 保持 ON2.0 秒, 那么数据传送将进行 25 次, 只有最后一次传送到 DM0000 的数据保存在那里。

图 B 中, 微分指令 @MOV(21) 只在 00000 状态变为 ON 后将 HR10 的内容到 DM0000 传送一次。即使扫描时间为 80ms, 00000 保持 ON2.0 秒, 传送操作只在 00000 从 OFF 变为 ON 的第一个扫描周期内执行一次。因为 00000 为 ON 时, HR10 的内容在 2 秒内很可能会变化, 所以 2 秒后 DM0000 的最终内容会由于使用 MOV(21) 或 @MOV(21) 而可能不同。

无论是采用微分指令还是非微分指令, 指令的操作数, 梯形图符号和它的参数都是相同的。当输入时, 使用同样的功能代码, 但是对于微分型指令, 要在功能代码后面输入 NOT。大多数(但不是全部)指令都具有微分形式。

联锁对于微分指令的影响请参照 5-12 联锁和清联锁—IL(O2) 和 IL(O3)。

CQM1H 也提供了微分指令: DIFU(13) 和 DIFD(14)。DIFU(13) 的操作与微分指令一样, 但是它用来使一位在一个循环内为 ON。DIFD(14) 也使一位在一个循环内为 ON, 但是它是在执行条件从 ON 变为 OFF 时执行。详情请参照 5-9-4 微分上升与下降—DIFU(13) 和 DIFD(14)。

5-5 扩展指令

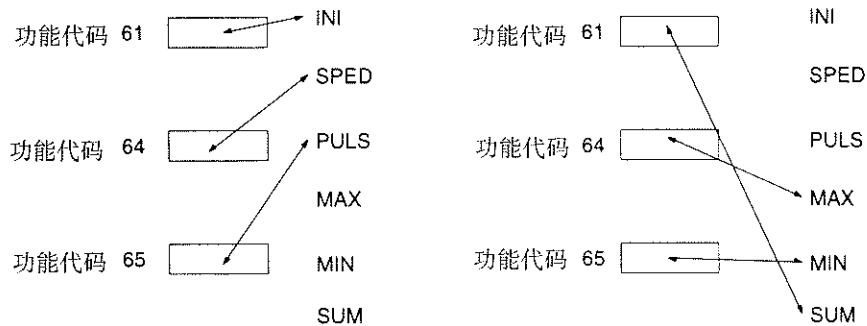
扩展指令集在特殊编程要求中作为辅助功能。功能代码可以分配到 18 个扩展指令, 使之在程序中使用。这就允许用户只取每个程序中需要的指令, 更有效地使用所需输入指令的函数代码。

扩展指令的助记符后面紧跟“—”作为功能代码,以表示编程前必须由用户在指令表中指定功能代码(除非在缺省设定下使用)。

任何一个指定功能代码的指令在用于编程前,需要由编程设备在 CQNIH 中指定功能代码。修改指定给扩展指令的功能代码将改变指令和操作数的含义,因此必须确保在编程前指定了功能代码并且在程序执行前已把扩展指令设置传送给 CQNIH。

例

下例显示如何修改缺省功能代码的设定。



出厂时,功能代码如上所示(在此例中,所有指令与脉冲输出有关)

如果未使用脉冲输出,并且需要最大值,最小值及求和,那么设置指令操作如上所示,以重新指定指令表中的指令。

扩展指令的功能代码

以下 18 个功能代码可用于扩展指令:

17, 18, 19, 47, 48, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 87, 88, 和 89

可使用的 74 个扩展指令以及 CQMIH 出厂时就指定好的缺省功能代码如下表所示。

| 助记符 | 代码 | 助记符 | 代码 | 助记符 | 代码 | 助记符 | 代码 |
|------|----|------|-----|------|-----|------|-----|
| ASFT | 17 | ACC | --- | FIXL | --- | RAD | --- |
| TKY | 18 | ACOS | --- | FLT | --- | SBBL | --- |
| MCMP | 19 | ADBL | --- | FLTL | --- | SCL2 | --- |
| RXD | 47 | APR | --- | FPD | --- | SCL3 | --- |
| TXD | 48 | ASIN | --- | HEX | --- | SEC | --- |
| CMPL | 60 | ATAN | --- | HKY | --- | SIN | --- |
| INI | 61 | AVG | --- | HMS | --- | SQRT | --- |
| PRV | 62 | CMND | --- | LINE | --- | SRCH | --- |
| CTBL | 63 | COLM | --- | LOG | --- | STUP | --- |
| SPED | 64 | COS | --- | MAX | --- | SUM | --- |
| PULS | 65 | CPS | --- | MBS | --- | TAN | --- |
| SCL | 66 | CPSL | --- | MBSL | --- | TTIM | --- |
| BCNT | 67 | DBS | --- | MIN | --- | XFRB | --- |
| BCMP | 68 | DBSL | --- | NEG | --- | ZCP | --- |
| STIM | 69 | DEG | --- | NEGL | --- | ZCPL | --- |
| DSW | 87 | EXP | --- | PID | --- | +F | --- |
| 7SEG | 88 | FCS | --- | PLS2 | --- | -F | --- |
| INT | 89 | FIX | --- | PMCR | --- | *F | --- |
| | | | | PWM | --- | /F | --- |

在使用时,扩展指令的指定可存贮在内存盒中。在使用一个已被其它 CQM1H 使用过的内存盒时要小心,确认使用了适当的扩展指令分配。



注意

如果 CQM1H 的 DIP 开关的 4 脚为 OFF,那么将使用缺省的功能代码,并且用户设定扩展指令的指定将被忽略,当电源通电时,缺省功能代码的指定也将被设定,删除任何以前的设定。

当从内存盒中读一段程序(有用户设定扩展指令)时,要确保 CPU 单元 DIP 开关的 4 脚为 ON。如果 4 脚为 OFF,那么缺省功能代码的指定将用作从内存盒中读取的程序的扩展指令。(在此情形中,把内存盒中读出的程序和内存盒中的程序相比较时,它们并不相同。)

5-6 编码右侧指令

已在第四章 梯形图编程中介绍了编写梯形图的助记符。如下所述,所有指令的梯形图符号的转换信息具有同样的形式,故不针对每条指令一一说明。

任何指令的第一个字用来定义指令和给出定义符。如果指令仅需要一个不带定义符的信号位操作数,此位操作数也作为助记符放在同一行上。所有其它操作数放到指令行后面的行上,每行一个操作数,并按指令的梯形图符号出现的顺序放置。

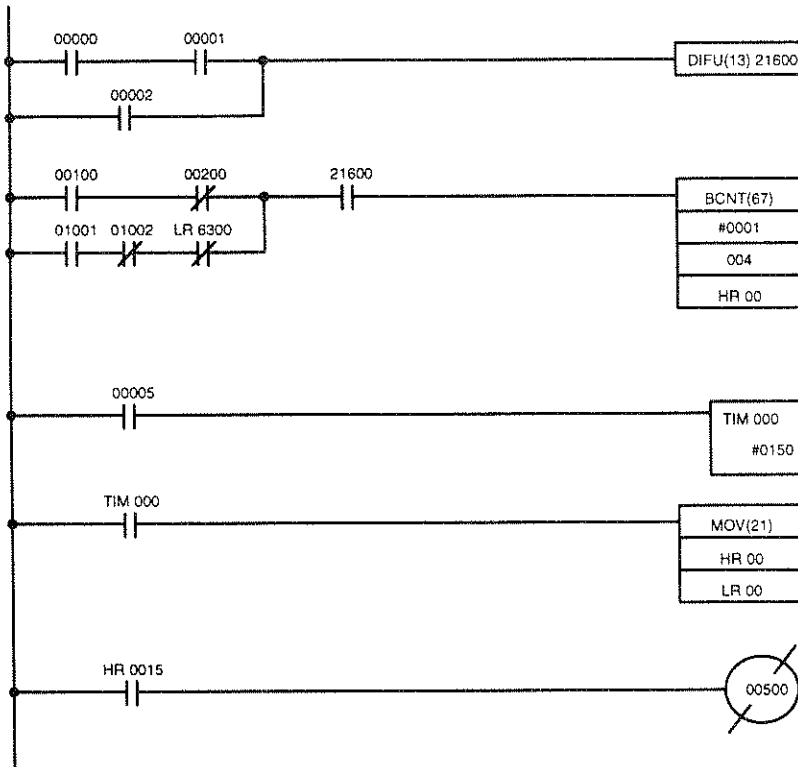
助记符代码表中的地址和指令列只填入指令字。对于所有其它行,左边的两列留空。如果指令不需要定义符或位操作数,第一行的数据列留空。在空白的数据列空间(用于不需要数据的所有指令字)上划交叉线是个好办法,因为这样可以迅速地扫描到数据列,以确定是否有地址留出。

如果在数据列中使用 IR 或 SR 地址,左边一列将留空。如果使用其它数据区,数据区缩写放到左边,地址放到右边。如果输入常数,在数据列左边放一个数字(#),输入的数字放到右边。被作为指令字的定义符输入的数不需要在其左边放置数字符。TIM/CNT 位如果被定义为定时器或计数器,需要加前缀 TIM(定时器)或 CNT(计数器)。

当对具有功能代码的指令进行编码时,注意写进功能代码,在通过编程器输入指令时这是必须的。同样要注意用@符号标注微分指令。

注 扩展指令的助记符后面加“(—)”作为功能代码,表明可以在用于编程之前,必须由用户在指令表中分配一个功能代码。详情请参照 17 页。

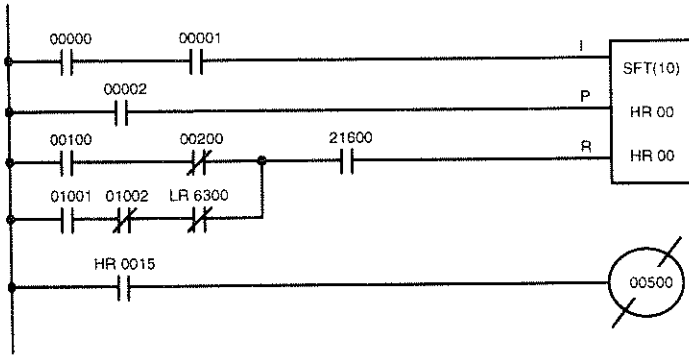
下面的图及相应的助记符说明了上面所介绍的内容。



| 地址 | 指令 | 操作数 |
|-------|----------|-----------|
| 00000 | LD | : 00000 |
| 00001 | AND | : 00001 |
| 00002 | OR | : 00002 |
| 00003 | DIFU(13) | : 21600 |
| 00004 | LD | : 00100 |
| 00005 | AND NOT | : 00200 |
| 00006 | LD | : 01001 |
| 00007 | AND NOT | : 01002 |
| 00008 | AND NOT | LR : 6300 |
| 00009 | OR LD | : — |
| 00010 | AND | : 21600 |
| 00011 | BCNT(67) | : — |
| | | # : 0001 |
| | | : 004 |
| | | HR : 00 |
| 00012 | LD | : 00005 |
| 00013 | TIM | : 000 |
| | | # : 0150 |
| 00014 | LD | TIM : 000 |
| 00015 | MOV(21) | : — |
| | | HR : 00 |
| | | LR : 00 |
| 00016 | LD | HR : 0015 |
| 00017 | OUT NOT | : 00500 |

多指令行

如果右侧指令需要多个指令行（如 KEEP(11)），指令所有行必须在右侧指令之前输入。编码各指令行，都用 LD 或 LD NOT 开始，以组成与右侧指令相结合的‘逻辑块’。下例用 SFT(10)加以说明。



| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | AND | 00001 |
| 00002 | LD | 00002 |
| 00003 | LD | 00100 |
| 00004 | AND NOT | 00200 |
| 00005 | LD | 01001 |
| 00006 | AND NOT | 01002 |
| 00007 | AND NOT | LR 6300 |
| 00008 | OR LD | — |
| 00009 | AND | 21600 |
| 00010 | SFT(10) | HR 00 |
| | | HR 00 |
| 00011 | LD | HR 0015 |
| 00012 | OUT NOT | 00500 |

5-7 指令表

本节给出 CQM1H 中可用的指令表。第一张表用于按功能代码查找指令。第二张表用于按助记符查找指令。

5-7-1 具有固定功能代码的指令

下表列出了具有固定的功能代码的指令。每个指令按助记符和指令名列出。用最左边一栏的数作为功能代码左边的数字,用栏头的数作为功能代码右边的数字。@符号代表微分型的指令。

扩展指令没有缺省功能代码,必须分配功能代码使之允许使用。即使是缺省功能代码的扩展指令,下表也省略为空格,可在使用时填入一个。扩展指令的详情参见下一页。

| 左位数 | 右位数 | | | | | | | | | |
|-----|-----------------|-----------------|----------------|-------------------|--------------------|-------------------|-------------------|-------------------|------------------------|------------------------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | NOP 空操作 | END 结束 | IL 联锁 | ILC 联锁清除 | JMP 跳转 | JME 跳转结束 | (@)FAL 错误报警及复位 | FALS 严重错误报警 | STEP 步定义 | SNXT 步开始 |
| 1 | SFT 移位寄存器 | KEEP 保持 | CNTR 可逆计数器 | DIFU 微分上升 | DIFD 微分下降 | TIMH 高速定时器 | (@)WSFT 字移动 | (扩展指令) | (扩展指令) | (扩展指令) |
| 2 | CMP 比较 | (@)MOV 传送 | (@)MVN 传送反相 | (@)BIN BCD-二进制 | (@)BCD 二进制-BCD | (@)ASL 左移 | (@)ASR 右移 | (@)ROL 循环左移 | (@)ROR 循环右移 | (@)COM 取补 |
| 3 | (@)ADD BCD加 | (@)SUB BCD减 | (@)MUL BCD乘 | (@)DIV BCD除 | (@)ANDW 逻辑与 | (@)ORW 逻辑或 | (@)XORW 异或 | (@)XNRW 异或非 | (@)INC 递增 | (@)DEC 递减 |
| 4 | (@)STC 设置进位 | (@)CLC 清除进位 | - | - | - | TRSM 跟踪存储器采样 | (@)MSG 信息显示 | (扩展指令) | (扩展指令) | - |
| 5 | (@)ADB 二进制加 | (@)SBB 二进制减 | (@)MLB 二进制乘 | (@)DVB 二进制除 | (@)ADDL 双字BCD加 | (@)SUBL 双字BCD减 | (@)MULL 双字BCD乘 | (@)DIVL 双字BCD除 | (@)BINL 双字BCD-双字二进制 | (@)BCDL 双字二进制-双字BCD |
| 6 | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) | (扩展指令) |
| 7 | (@)XFER 块传送 | (@)BSET 块设置 | (@)ROOT 平方根 | (@)XCHG 数据交换 | (@)SLD 左移1位数字 | (@)SRD 右移1位数字 | (@)MLPX 4-16译码 | (@)DMPX 16-4编码 | (@)SDEC 7段译码 | - |
| 8 | (@)DIST 单字分配 | (@)COLL 数据收集 | (@)MOVB 传送位 | (@)MOVD 传送数字 | (@)SFTR 可逆移位寄存器 | (@)TCMP 表比较 | (@)ASC ASCII转换 | (扩展指令) | (扩展指令) | (扩展指令) |
| 9 | (@)SEND 网络发送 | (@)SBS 子程序入口 | SBN 子程序定义 | RET 子程序返回 | - | - | - | (@)IORF I/O刷新 | (@)RECV 网络接收 | (@)MCRO 宏 |

5-7-2 扩展指令

下表列出了可使用的 74 条扩展指令以及 CQM1H 出厂时分配的缺省功能代码,详情参考 1-4 扩展指令。

| 助记符 | 代码 | 助记符 | 代码 | 助记符 | 代码 | 助记符 | 代码 |
|------|----|------|-----|------|-----|------|-----|
| ASFT | 17 | ACC | --- | FIXL | --- | RAD | --- |
| TKY | 18 | ACOS | --- | FLT | --- | SBBL | --- |
| MCMP | 19 | ADBL | --- | FLTL | --- | SCL2 | --- |
| RXD | 47 | APR | --- | FPD | --- | SCL3 | --- |
| TXD | 48 | ASIN | --- | HEX | --- | SEC | --- |
| CMPL | 60 | ATAN | --- | HKY | --- | SIN | --- |
| INI | 61 | AVG | --- | HMS | --- | SQRT | --- |
| PRV | 62 | CMND | --- | LINE | --- | SRCH | --- |
| CTBL | 63 | COLM | --- | LOG | --- | STUP | --- |
| SPED | 64 | COS | --- | MAX | --- | SUM | --- |
| PULS | 65 | CPS | --- | MBS | --- | TAN | --- |
| SCL | 66 | CPSL | --- | MBSL | --- | TTIM | --- |
| BCNT | 67 | DBS | --- | MIN | --- | XFRB | --- |
| BCMP | 68 | DBSL | --- | NEG | --- | ZCP | --- |
| STIM | 69 | DEG | --- | NEGL | --- | ZCPL | --- |
| DSW | 87 | EXP | --- | PID | --- | /F | --- |
| 7SEG | 88 | FCS | --- | PLS2 | --- | +F | --- |
| INT | 89 | FIX | --- | PMCR | --- | -F | --- |
| | | | | PWM | --- | *F | --- |

5-7-3 按助记符字母顺序排列的指令表

代码这一列内的“—”代表没有固定功能代码的扩展指令。“无”则表示该指令不使用功能代码。@符号表示微分形式的指令。

| 助记符 | 代码 | 字数 | 名称 | 页 |
|----------|----|----|----------|-----|
| 7SEG | 88 | 4 | 7 段显示输出 | 412 |
| ACC (@) | — | 4 | 加速控制 | 387 |
| ACOS (@) | — | 3 | 反正弦 | 354 |
| ADB (@) | 50 | 4 | 二进制加 | 316 |
| ADBL (@) | — | 4 | 双字二进制加 | 320 |
| ADD (@) | 30 | 4 | BCD 加 | 305 |
| ADDL (@) | 54 | 4 | 双字 BCD 加 | 311 |
| AND | 无 | 1 | 与 | 213 |
| AND LD | 无 | 1 | 与装载 | 214 |
| AND NOT | 无 | 1 | 与非 | 213 |
| ANDW (@) | 34 | 4 | 逻辑与 | 360 |
| APR (@) | — | 4 | 算术处理 | 332 |
| ASC (@) | 86 | 4 | ASCII 转换 | 289 |
| ASFT (@) | 17 | 4 | 异步移位寄存器 | 256 |
| ASIN (@) | — | 3 | 反正弦 | 353 |
| ASL (@) | 25 | 2 | 算术左移 | 251 |
| ASR (@) | 26 | 2 | 算术右移 | 251 |
| ATAN (@) | — | 3 | 反正切 | 355 |

| 助记符 | 代码 | 字数 | 名称 | 页 |
|---------|----|----|-------------|-----|
| AVG | — | 4 | 平均值 | 329 |
| BCD(@) | 24 | 3 | 二进制—BCD | 280 |
| BCDL(@) | 59 | 3 | 双字二进制—双字BCD | 281 |
| BCMP(@) | 68 | 4 | 块比较 | 270 |
| BCNT(@) | 67 | 4 | 位计数器 | 373 |
| BIN(@) | 23 | 3 | BCD—二进制 | 279 |
| BINL(@) | 58 | 3 | 双字BCD—双字二进制 | 280 |
| BSET(@) | 71 | 4 | 块设置 | 260 |
| CLC(@) | 41 | 1 | 清除进位 | 305 |
| CMND(@) | — | 4 | 发布命令 | 400 |
| CMP | 20 | 3 | 比较 | 268 |
| CMPL | 60 | 4 | 双字比较 | 272 |
| CNT | 无 | 2 | 计数器 | 226 |
| CNTR | 12 | 3 | 可逆计数器 | 227 |
| COLL(@) | 81 | 4 | 数据收集 | 263 |
| COLM(@) | — | 4 | 行—列 | 301 |
| COM(@) | 29 | 2 | 取补 | 360 |
| COS(@) | — | 3 | 余弦 | 351 |
| CPS | — | 4 | 有符号二进制比较 | 274 |
| CPSL | — | 4 | 双字有符号二进制比较 | 275 |
| CTBL(@) | 63 | 4 | 装载比较表 | 232 |
| DBS(@) | — | 4 | 有符号二进制除法 | 325 |
| DBSL(@) | — | 4 | 双字有符号二进制除法 | 326 |
| DEC(@) | 39 | 2 | BCD递减 | 363 |
| DEG(@) | — | 3 | 弧度—度 | 349 |
| DIFD | 14 | 2 | 微分下降 | 217 |
| DIFU | 13 | 2 | 微分上升 | 217 |
| DIST(@) | 80 | 4 | 单字分配 | 261 |
| DIV(@) | 33 | 4 | BCD除法 | 309 |
| DIVL(@) | 57 | 4 | 双字BCD除法 | 314 |
| DMPX(@) | 77 | 4 | 16—4编码 | 284 |
| DSW | 87 | 4 | 数字切换 | 415 |
| DVB(@) | 53 | 4 | 二进制除法 | 319 |
| END | 01 | 1 | 结束 | 218 |
| EXP(@) | — | 4 | 指数 | 357 |
| FAL(@) | 06 | 2 | 错误报警与复位 | 221 |
| FALS | 07 | 2 | 严重错误报警 | 221 |
| FCS(@) | — | 4 | FCS计算 | 373 |
| FIX(@) | — | 3 | 浮点—16位 | 340 |
| FIXL(@) | — | 3 | 浮点—32位 | 341 |
| FLT(@) | — | 3 | 16位—浮点 | 342 |
| FLTL(@) | — | 3 | 32位—浮点 | 343 |
| FPD | — | 4 | 故障点检测 | 375 |
| HEX(@) | — | 4 | ASCII到十六进制 | 290 |
| HKY | — | 4 | 16键输入 | 419 |
| HMS | — | 4 | 秒—时 | 299 |
| IL | 02 | 1 | 联锁 | 218 |
| ILC | 03 | 1 | 联锁清除 | 218 |
| INC(@) | 38 | 2 | 递增 | 363 |

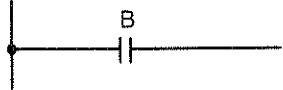
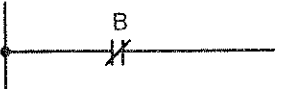
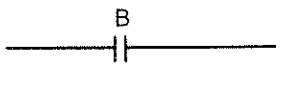
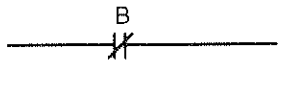


| 助记符 | 代码 | 字数 | 名称 | 页 |
|---------|----|----|------------|-----|
| INI(@) | 61 | 4 | 模式控制 | 244 |
| INT(@) | 89 | 4 | 中断控制 | 379 |
| IORF(@) | 97 | 3 | I/O刷新 | 370 |
| JME | 05 | 2 | 跳转结束 | 220 |
| JMP | 04 | 2 | 跳转 | 220 |
| KEEP | 11 | 2 | 保持 | 216 |
| LD | 无 | 1 | 装载 | 213 |
| LD NOT | 无 | 1 | 装载非 | 213 |
| LINE | — | 4 | 行 | 300 |
| LOG(@) | — | 3 | 对数 | 359 |
| MAX(@) | — | 4 | 求最大值 | 327 |
| MBS(@) | — | 4 | 有符号二进制乘法 | 323 |
| MBSL(@) | — | 4 | 双字有符号二进制乘法 | 324 |
| MCMP(@) | 19 | 4 | 多字比较 | 328 |
| MCRO(@) | 99 | 4 | 宏 | 371 |
| MIN(@) | — | 4 | 求最小值 | 328 |
| MLB@) | 52 | 4 | 二进制乘法 | 318 |
| MLPX(@) | 76 | 4 | 4-16译码 | 282 |
| MOV(@) | 21 | 3 | 传送 | 257 |
| MOVB(@) | 82 | 4 | 传送位 | 265 |
| MOVD(@) | 83 | 4 | 传送数字 | 266 |
| MSG(@) | 46 | 2 | 信息 | 369 |
| MUL(@) | 32 | 4 | BCD乘法 | 308 |
| MULL(@) | 56 | 4 | 双字BCD乘 | 313 |
| MVN(@) | 22 | 3 | 传送非 | 258 |
| NEG(@) | — | 4 | 二进制补码 | 302 |
| NEGL(@) | — | 4 | 双字二进制补码 | 303 |
| NOP | 00 | 1 | 空操作 | 218 |
| OR | 无 | 1 | 或 | 213 |
| OR LD | 无 | 1 | 或装载 | 214 |
| OR NOT | 无 | 1 | 或非 | 213 |
| ORW(@) | 35 | 4 | 逻辑或 | 361 |
| OUT | 无 | 2 | 输出 | 214 |
| OUT NOT | 无 | 2 | 输出非 | 214 |
| PID | — | 4 | PID控制 | 392 |
| PLS2(@) | — | 4 | 脉冲输出 | 385 |
| PMCR(@) | — | 4 | 协议宏 | 410 |
| PRV(@) | 62 | 4 | 读高速计数器当前值 | 246 |
| PULS(@) | 65 | 4 | 设置脉冲 | 380 |
| PWM(@) | — | 4 | 可变占空比脉冲 | 389 |
| RAD(@) | — | 3 | 度数—弧度 | 348 |
| RECV(@) | 98 | 4 | 网络接收 | 397 |
| RET(@) | 93 | 1 | 子程序返回 | 367 |
| ROL(@) | 27 | 2 | 循环左移 | 252 |
| ROOT(@) | 72 | 3 | 平方根 | 315 |
| ROR(@) | 28 | 2 | 循环右移 | 252 |
| RSET | 无 | 2 | 复位 | 215 |
| RXD(@) | 47 | 4 | 接收 | 403 |
| SBB(@) | 51 | 4 | 二进制减法 | 317 |

| 助记符 | 代码 | 字数 | 名称 | 页 |
|---------|----|----|--------------|-----|
| SBBL(@) | — | 4 | 双字二进制减法 | 321 |
| SBN | 92 | 2 | 子程序定义 | 367 |
| SBS(@) | 91 | 2 | 子程序入口 | 365 |
| SCL(@) | 66 | 4 | 标度 | 293 |
| SCL2(@) | — | 4 | 有符号二进制到BCD标度 | 295 |
| SCL3(@) | — | 4 | BCD到有符号二进制标度 | 296 |
| SDEC(@) | 78 | 4 | 7段译码 | 286 |
| SEC | — | 4 | 时一秒 | 298 |
| SEND(@) | 90 | 4 | 网络发送 | 393 |
| SET | 无 | 2 | 设置 | 215 |
| SFT | 10 | 3 | 移位寄存器 | 249 |
| SFTR(@) | 84 | 4 | 可递移位寄存器 | 254 |
| SIN(@) | — | 4 | 正弦 | 350 |
| SLD(@) | 74 | 3 | 1位数字左移 | 253 |
| SNXT | 09 | 2 | 步开始 | 222 |
| SPED(@) | 64 | 4 | 速度输出 | 382 |
| SQRT(@) | — | 3 | 平方根 | 356 |
| SRCH(@) | — | 4 | 数据搜索 | 390 |
| SRD(@) | 75 | 3 | 1位数字右移 | 254 |
| STC(@) | 40 | 1 | 设置进位 | 305 |
| STEP | 08 | 2 | 步定义 | 222 |
| STIM(@) | 69 | 4 | 间隔定时器 | 231 |
| STUP(@) | — | 4 | 修改串口设置 | 407 |
| SUB | 31 | 4 | BCD减法 | 306 |
| SUBL(@) | 55 | 4 | 双字BCD减法 | 312 |
| SUM(@) | — | 4 | 求和 | 330 |
| TAN(@) | — | 3 | 正切 | 352 |
| TCMP(@) | 85 | 4 | 表比较 | 269 |
| TIM | 无 | 2 | 定时器 | 225 |
| TMH | 15 | 3 | 高速定时器 | 228 |
| TKY(@) | 18 | 4 | 十键输入 | 422 |
| TRSM | 45 | 1 | 跟踪内存采样 | 367 |
| TTIM | — | 4 | 累加定时器 | 229 |
| TXD(@) | 48 | 4 | 传送 | 405 |
| WSFT(@) | 16 | 3 | 字移位 | 250 |
| XCHG(@) | 73 | 3 | 数据交换 | 261 |
| XFER(@) | 70 | 4 | 块传送 | 259 |
| XFRB(@) | — | 4 | 位传送 | 267 |
| XNRW(@) | 37 | 4 | 异或非 | 362 |
| XORW(@) | 36 | 4 | 异或 | 362 |
| ZCP | — | 4 | 区域范围比较 | 277 |
| ZCPL | — | 4 | 双字区域范围比较 | 278 |
| +F(@) | — | 4 | 浮点加法 | 343 |
| -F(@) | — | 4 | 浮点减法 | 345 |
| *F(@) | — | 4 | 浮点乘法 | 346 |
| /F(@) | — | 4 | 浮点除法 | 347 |

5-8 梯形图指令

梯形图指令包括梯级指令,逻辑块指令和梯形图上相应的条件。逻辑块指令用于相对复杂的部分。

5-8-1 LOAD,LOAD NOT,AND ,AND NOT,OR 和 OR NOT

| | 梯形图符号 | 操作数数据区 | | |
|---------------------------------|---|---|------|---------------------------------|
| LOAD - LD |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR, TR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR, TR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR, TR | | | | |
| LOAD NOT - LD NOT |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR | | | | |
| AND - AND |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR | | | | |
| AND NOT - AND NOT |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR | | | | |
| OR - OR |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR | | | | |
| OR NOT - OR NOT |  | <table border="1"><tr><td>B: 位</td></tr><tr><td>IR, SR, AR, HR, TIM/CNT, LR</td></tr></table> | B: 位 | IR, SR, AR, HR, TIM/CNT, LR |
| B: 位 | | | | |
| IR, SR, AR, HR, TIM/CNT, LR | | | | |

限制

只要不超过 PC 的存储容量,这些指令使用次数和使用顺序不受限制。

说明

这六条基本指令对应于梯形图上的条件部分。如第四章 梯形图编程所述,分配到各个指令的位的状态决定所有其它指令的执行条件,每条指令和每个位地址可以按需要使用多次。每一个可根据需要使用多个指令。

分配到 LD 或 LD NOT 的位操作数(B)的状态决定第一个执行条件。AND 将执行条件与它的位操作数状态进行逻辑与; AND NOT 将执行条件与其位操作数状态取反再进行逻辑与, OR 将执行条件与其位操作数状态进行逻辑或; OR NOT 将执行条件与其位操作数状态取反再进行逻辑或。

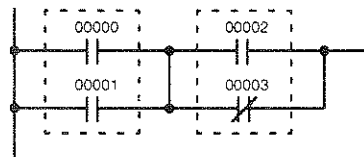
标志

这些指令不影响标志。

5-8-2 AND LOAD 和 OR LOAD

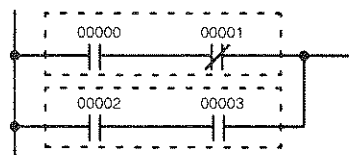
AND LOAD - AND LD

梯形图符号



OR LOAD - OR LD

梯形图符号



说明

当指令不能仅仅使用 OR 和 AND 操作组合进逻辑块时, 要使用 AND LD 和 OR LD。AND 和 OR 操作是将一个位状态和执行条件进行逻辑组合, 而 AND LD 和 OR LD 将两个执行条件(当前的和上次未使用的)进行逻辑组合。

在画梯形图时, 并不需要使用 AND LD 和 OR LD 指令, 因为可以从 CX-Programmer 直接输入梯形图, 所以也不需要它们。然而当转换程序到助记符形式, 或以助记符形式输入时就需要这些指令。

为了减少需要的编程指令数, 需要对逻辑块指令有基本的理解。逻辑块的介绍请参照 4-3-6 逻辑块指令。

标志

这些指令不影响标志。

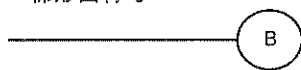
5-9 位控制指令

有七条指令通常可以用来控制独立的位状态。它们是 OUT, OUT NOT, DIFU (13), DIFD (14), SET, RSET 和 KEEP (11)。这些指令使用不同的方法将位变为 ON 和 OFF。

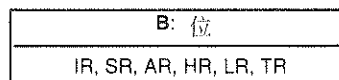
5-9-1 输出和输出非——OUT 和 OUT NOT

输出 - OUT

梯形图符号

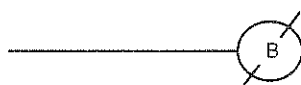


操作数数据区

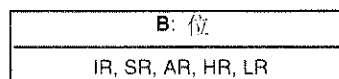


输出非 - OUT NOT

梯形图符号



操作数数据区



限制

通常, 任何输出位只能由一条指令控制其状态。

说明

OUT 和 OUT NOT 可以用于按照执行条件,控制指定位的状态。OUT 在 ON 执行条件下使指定位为 ON, 在 OFF 执行条件下使指定位为 OFF。使用 TR 位, OUT 出现在分支点上,而不是在指令行的末端。详情请参照 4-3-8 分支指令行。

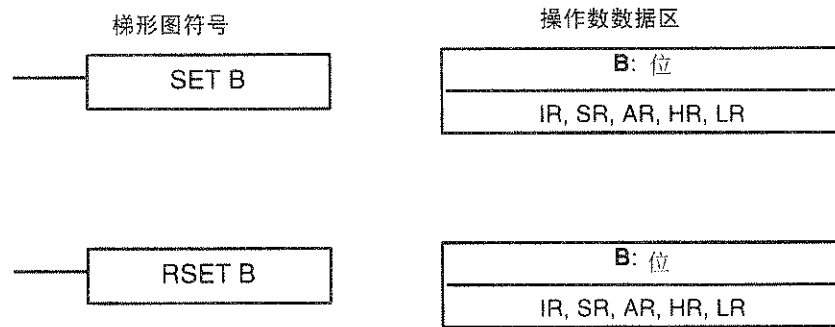
OUT NOT 在 OFF 执行条件下使指定位为 ON, 在 ON 执行条件下使指定位为 OFF。

OUT 和 OUT NOT 通过使分配到梯形图上的条件的位为 ON 或 OFF 来控制执行,从而决定其它指令的执行条件。这一点特别有用;使它可以由一组复杂的条件来控制单个工作位的状态,然后那个工作位可以用来控制其它指令。通过 OUT 或 OUT NOT 与 TIM 组合可以控制一个位保持 ON 或 OFF 的时间的长短。详情请参照 5-16-1 定时器—TIM 中的例子。

标志

这些指令不影响标志。

5-9-2 置位和复位——SET 和 RSET



说明

当执行条件为 ON 时,SET 使操作数位为 ON,并且当执行条件为 OFF 时,不影响操作数位的状态。当执行条件为 ON 时,RSET 使操作数位为 OFF,并且当执行条件为 OFF 时,不影响操作数位的状态。

SET 的操作与 OUT 的操作不同,因为对于 OUT 指令来说,当执行条件为 OFF 时,操作数位为 OFF。同样地,RSET 的操作与 OUT NOT 的操作不同,因为对于 OUT NOT 指令来说,当执行条件为 OFF 时,操作数位为 ON。

注意

当遇到联锁或跳转条件时(即,当 IL(02)或 JMP(04)以 OFF 执行条件执行时),在 IL(02)和 ILC(03)之间,或 JMP(04)和 JME(05)之间的 SET 和 RSET 的操作数位状态不会改变。

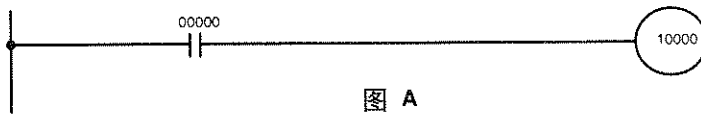
标志

这些指令不影响标志。

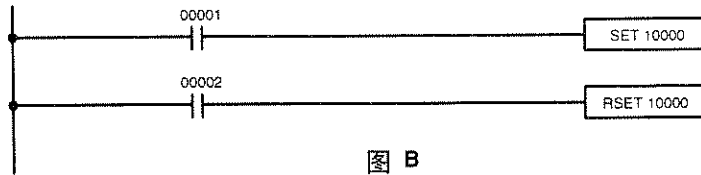
例子

下面的例子表示了 OUT 和 SET/RSET 之间的差别。在第一个例子中(图 A),当 IR0000 为 ON 或 OFF 时,IR1000 将置为 ON 或 OFF。

在第二个例子中 (图 B), 当 IR00001 为 ON 时, IR10000 将置为 ON 并保持为 ON (即使 IR00001 变为 OFF), 直到 IR00002 为 ON。

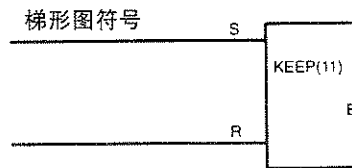


| 地址 | 指令 | 操作数 |
|-------|-----|-------|
| 00000 | LD | 00000 |
| 00001 | OUT | 10000 |



| 地址 | 指令 | 操作数 |
|-------|------|-------|
| 00000 | LD | 00001 |
| 00001 | SET | 10000 |
| 00002 | LD | 00002 |
| 00003 | RSET | 10000 |

5-9-3 保持—KEEP(11)



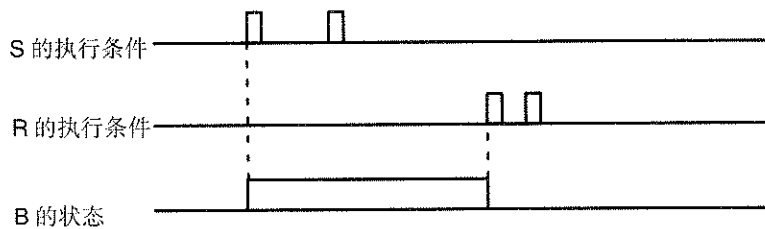
| 操作数数据区 | |
|--------------------|--|
| B: 位 | |
| IR, SR, AR, HR, LR | |

限制说明

任何输出位一般只能在控制其状态的一条指令中使用。

KEEP(11) 用来基于两个执行条件保持指定位的状态。这两个执行条件被称为 S 和 R。S 为置位输入; R 为复位输入。KEEP (11) 的动作就象一个由 S 置位、由 R 复位的锁存继电器。

当 S 转为 ON 时, 指定位将置位 ON, 并保持为 ON 直到复位, 而不管 S 保持为 ON 还是变为 OFF。当 R 为 ON, 指定位将置为 OFF, 并保持 OFF 直到置位, 而不管 R 保持为 ON 还是变为 OFF。执行条件之间的关系和 KEEP(11) 的位状态如下所示。

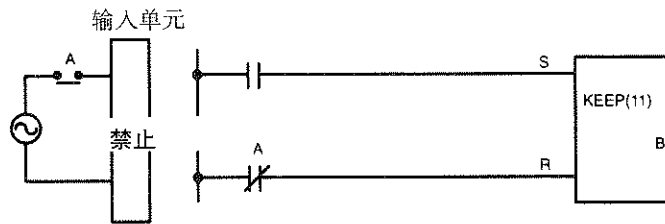


标志

这些指令不影响标志。

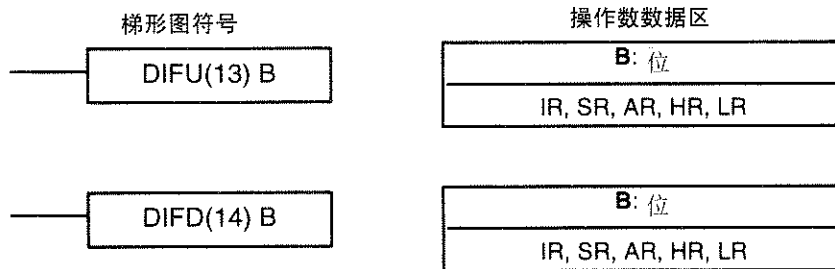
注意

当使用由一个常闭外设控制复位行的 KEEP 指令时,要千万小心。当输入外设使用交流电源时,禁止用输入位取反条件作为 KEEP(11)的复位输入(R)。PC 直流电源断电的延迟(相对输入设备的交流电源)可能导致 KEEP(11)的指定位复位。这种情况如下所示。



KEEP 中使用的位在联锁中不复位。详情请参照 5-12 联锁和联锁清除—IL(02)和 ILC(03)。

5-8-4 上升沿和下降沿微分—DIFU(13)和 DIFD(14)



限制说明

任何输出位一般只能在控制其状态的一条指令中使用。

DIFU(13)和 DIFD(14)用于将指定位在一个循环内转为 ON。执行时,DIFU(13)将当前执行条件与以前的执行条件进行比较。如果以前的执行条件为 OFF,当前为 ON,DIFU(13)把指定位置为 ON。如果以前的执行条件为 ON,不管当前为 ON 或还是 OFF,DIFU(13)均将把指定位置为 OFF 或保持 OFF(即,如果指定位已为 OFF)。因此,指定位绝不会转为 ON 超过一个循环,假如它每个循环都执行的话(见下面的注意)。

执行时,DIFD(14)将当前执行条件与以前的执行条件进行比较。如果以前的执行条件为 ON,当前为 OFF,DIFD(14)把指定位置为 ON。如果以前的执行条件为 OFF,不管当前为 ON 还是 OFF,DIFD(14)将把指定位置为 OFF 或使它保持为 OFF。因此,指定位绝不会转为 ON 超过一个循环,假如它每个循环都执行的话(见下面的注意)。

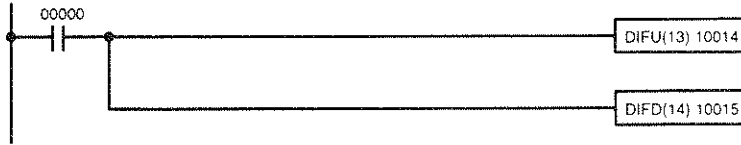
当不能用微分指令(即,使用@作前缀的指令)但需要一条特定的指令执行一个周期时可使用这些指令。在使用它们能简化程序时,也可以用于具有微分形式指令的非微分形式。这样的例子在下面给出。

标志

这些指令不影响标志。

注意

当指令在 IL 和 ILC 之间, JMP 和 JME 之间或在子程序中时, DIFU(13) 和 DIFD(14) 操作可能不确定。请参考 5—12 联锁和联锁清除—IL(02) 和 ILC(03), 5—13 跳转和跳转结束—JMP(04) 和 JME(05), 5—27 子程序指令和 5—28—8 中断控制—INT(89)。



| | | |
|-------|----------|-------|
| 00000 | LD | 00000 |
| 00001 | DIFU(13) | 10014 |
| 00002 | DIFD(14) | 10015 |

5—10 空操作—NOP(00)

说明

编程中一般不需要 NOP(00), 它也没有梯形图符号。在程序中发现 NOP(00) 时, 什么也不执行, 程序转移到下一个指令执行。当在编程前清除存储器时, 所有的地址都写为 NOP(00), NOP(00) 可以通过 00 功能代码输入。

标志

NOP(00) 不影响标志。

5—11 结束—END(01)



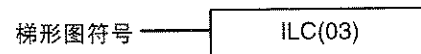
说明

在任何程序中, 都需要 END(01) 作为最后一条指令。如果有子程序, END(01) 放到最后一个子程序后面。写在 END(01) 后面的程序将不会执行。END(01) 可以放到程序的任何地方, 程序执行到此点结束。在调试时有时需要这样做, 但之后必须要将它移到最后, 以便执行其余的程序。如果程序中没有 END(01), 将不会执行任何指令, 并出现错误信息 “NO END INST”。

标志

END(01) 将 ER, CY, GR, EQ, LE, OF 和 UF 标志置为 OFF。

5—12 联锁和联锁清除—IL(02) 和 ILC(03)



说明

IL(02) 总是与 ILC(03) 联合使用以产生联锁。联锁用来产生与使用 TR 位得到的一样的分支。但是, 当 IL(02) 的执行条件为 OFF 时, 在 IL(02) 和 ILC(03) 之间的指令的处理与使用 TR 位时不一样。如果 IL(02) 的执行条件为 ON, 程序将如编写的那样执行, 用 ON 执行条件从 IL(02) 所位于的点启动各指令行, 直到下一个 ILC(03)。两种方法的基本说明请参照 4—3—8 分支指令行。

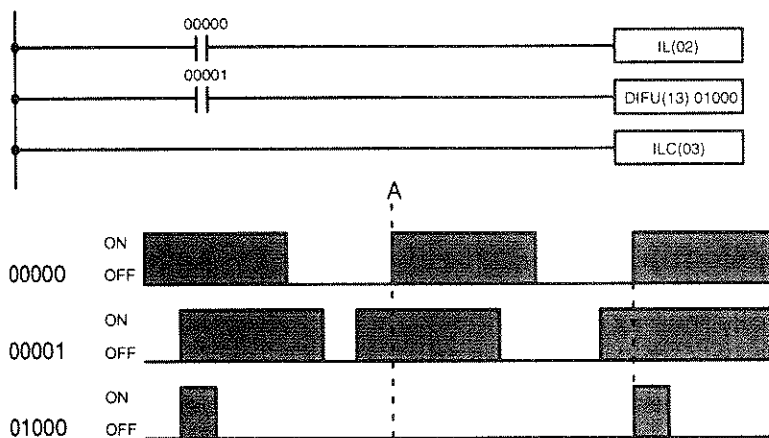
如果 IL(02) 的执行条件为 OFF, IL(02) 和 ILC(03) 之间的联锁段将按下表操作。

| 指令 | 处 理 |
|--------------------|---|
| OUT 和 OUT NOT | 指定位转为 OFF |
| TIM 和 TIMH(15) | 复位 |
| CNT, CNTR(12) | 保持当前值 |
| KEEP(11) | 位状态保持 |
| DIFU(13)和 DIFD(14) | 不执行(见下面) |
| 所有其它指令 | 指令不执行, 所有作为操作数写进指令的 IR, AR, LR, HR 和 SR 位和字置为 OFF |

IL(02)和 ILC(03)不是必须成对使用。IL(02)可以在一行中多次使用,每个 IL(02)到下一个 ILC(03)产生联锁段。只有在 ILC(03)和前面的 ILC(03)之间至少有一个 IL(02)时,才可以使用 ILC(03)。

在联锁中 DIFU(13) 和 DIFD(14)

如果 DIFU(13)和 DIFD(14)在联锁段中,并且 IL(02)的执行条件为 OFF, DIFU(13)或 DIFD(14)的执行条件的变化将不能记录下来。因此当在 IL(02)的执行条件转为 ON 后要执行联锁段中的 DIFU(13)或 DIFD(14)时, DIFU(13)或 DIFD(14)的执行条件将同联锁变为有效之前(即,在 IL(02)的执行条件变为 OFF 之前)的执行条件相比较。对此,梯形图和位状态变化表示如下。当 00000 为 OFF 时,联锁有效。注意在 A 点,即使 00001 已经为 OFF,然后又为 ON, 01000 不转为 ON。



| 地址 | 指令 | 操作数 |
|-------|----------|-------|
| 00000 | LD | 00000 |
| 00001 | IL(02) | |
| 00002 | LD | 00001 |
| 00003 | DIFU(13) | 01000 |
| 00004 | ILC(03) | |

注意

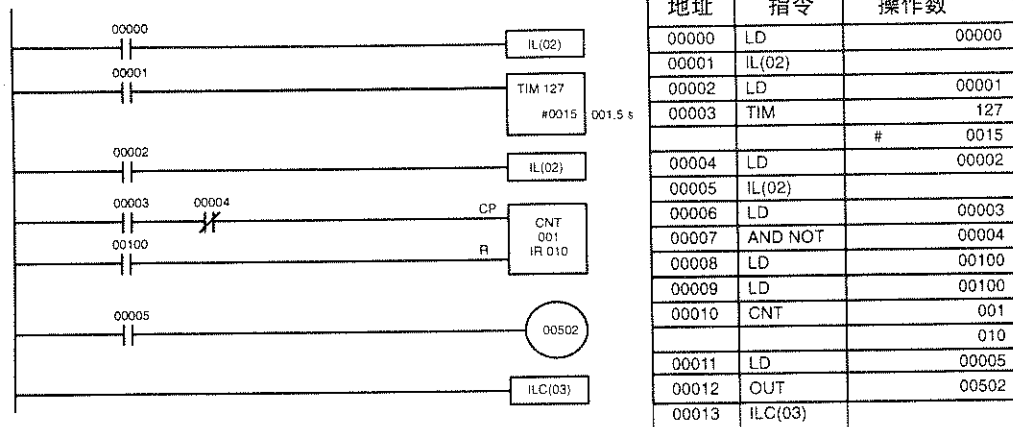
在一个或多个 IL(02)之后必须有一个 ILC(03)。虽然一个 ILC(03) 可以按需要用于对应多次 IL(02), 但是如果其中没有至少一个 IL(02), 将不能使用 ILC(03) 指令(即不可以嵌套)。当 ILC(03) 执行时, 有效的 ILC(03)和前面的 ILC(03)之间的所有联锁都将被清除。当一个以上的 IL(02)与同一个 ILC(03)一起使用时, 进行程序检查时, 将会出现错误信息, 但程序能正常执行。

标志

这些指令不影响标志。

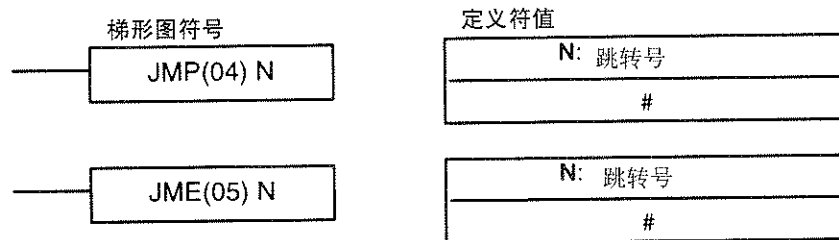
例

下图示出了使用两次 IL(02)与一条 ILC(03)对应。



当第一个 IL(02) 的执行条件为 OFF, TIM127 将被复位为 1.5S, CNT001 不改变, 00502 被置为 OFF。当第一个 IL(02)的执行条件为 ON,且第二个 IL(02)的执行条件为 OFF, TIM127 将根据 00001 的状态执行。CNT001 不改变, 00502 置为 OFF。当两个 IL(02)的执行条件为 ON,程序将按照所编写的执行。

5—13 跳转和跳转结束—JMP(04)和 JME(05)



限制

跳转号在 01 ~ 99 之间只能在 JMP(04) 中可使用一次, 在 JME(05) 中使用一次。即每个跳转号只能用来定义一次跳转。但跳转号 00 可以按照需要多次使用。

跳转号从 00 ~ 99。

说明

JMP(04) 总是与 JME(05) 联合使用以产生跳转, 即从梯形图的一点跳到另一点。JMP(04) 定义跳转的源点, JME(05) 定义跳转的目的地。当 JMP(04) 的执行条件为 ON 时, 不产生跳转, 程序如所编写的那样执行。当 JMP(04) 的执行条件为 OFF 时, 将跳转到具有同样跳转号的 JME(05), 并接着执行 JME(05) 后面的指令。

如果 JMP(04) 的跳转号在 01 ~ 99 之间, 当产生跳转时, 将直接跳转到具有相同跳转号的 JME(05), 不执行其中的任何指令。定时器、计数器的状态, OUT 使用的位, OUT NOT 使用的位以及所有其它由 JMP(04) 和 JME(05) 之间的指令控制的状态位都不改变。这些跳转号只可以用来定义一个跳转。因为在 JMP(04) 和 JME(05) 之间的所有指令都被跳过, 跳转号 01 ~ 99 可以用来减少循环时间。

跳转号 00

如果 JMP(04)的跳转号为 00,CUP 将寻找下一个具有 00 跳转号的 JME(05)。为此,它必须在整个程序中查找。这使得它比其它跳转所用的循环时间较长(当执行条件为 OFF)。定时器、计数器的状态,OUT 使用的位,OUT NOT 使用的位以及所有其它由 JMP(04) 00 和 JME(05) 00 之间的指令控制状态的位的状态都不改变。跳转号 00 可以按需要使用多次。从 JMP(04)00 出发的跳转总是跳转到程序中的下一个 JME(05) 00 处。因此可以连续地使用 JMP(04) 00,而用同一个 JME(05)00 与它们相对应。然而连续使用 JME(05)00 是没有意义的,因为产生的跳转将在第一个 JME(05)00 处结束。

跳转中的 DIFU
(13) 和 DIFD
(14)

虽然 DIFU(13)和 DIFD(14)用来将指定位在一个循环中置为 ON,但是当它们在 JMP(04)和 JME(05)之间时就不需要这样做。因为一旦 DIFU(13)或 DIFD(14)将某位置为 ON,它将保持 ON 直到下一次再执行 DIFU(13)或 DIFD(14)指令时才结束。在一般的编程中,这意味着下一个循环。但在跳转程序中,这意味着下一次将不发生从 JMP(04)至 JME(05)的跳转,即如果由于 DIFU(13)或 DIFD(14)将某位转为 ON,然后下一个循环中将产生跳转,使得 DIFU(13)或 DIFD(14)被跳过,使指定位将保持 ON,直到下一次控制跳转的 JMP(04)的执行条件为 ON 为止。

注意

当 JMP(04)或 JME(05)不是成对使用时,在检查程序时,将会出现错误信息。虽然当 JMP(04)00 和 JME(05)00 不成对使用时,也会出现这条错误信息,程序将按照所编写的执行。

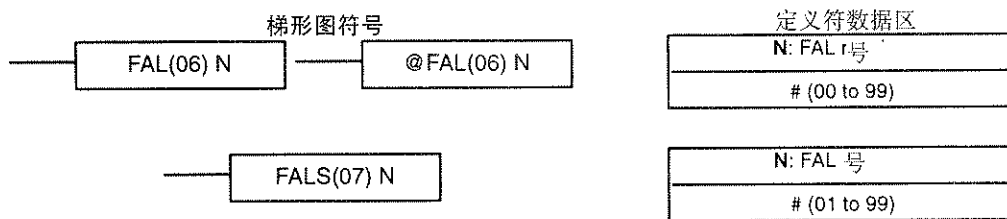
标志

这些指令不影响标志。

例子

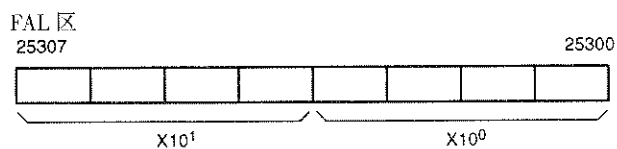
跳转程序的例子在 4—3—9 跳转中给出。

5—14 用户错误指令:错误报警与复位—FAL(06)及严重错误报警—FALS(07)



说明

FAL(06)和 FALS(07)用来使编程者可以在运行、维护和调试程序时输出错误编写。当以 ON 条件执行时,这两条指令都可以输出一个 FAL 编号到 SR253 的 00 位 ~ 07 位。输出的 FAL 编号可以在 01 ~ 99 之间,它是作为 FAL(06)或 FALS(07)的定义符输入的。定义符为 00 的 FAL(06)用来复位此区域(见下图)。



FAL(06)产生非严重错误,FALS(07)产生严重错误。当 FAL(06)以 ON 执行条件执行时,CPU 前面的报警/错误指示灯将闪烁,但 PC 将继续运行。当 FALS(07)以 ON 执行条件执行时,报警/错误指示灯将变亮,PC 停止运行。系统在 FAL 区生成错误代码。

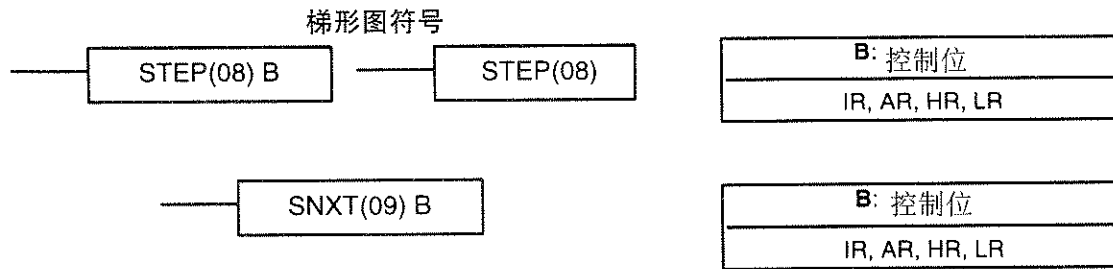
复位错误

在存储器中可保存 FAL 错误代码,而且其中只有一个可以保留在 FAL 区中。为了访问其它 FAL 代码,要执行 FAL(06) 00 以复位 FAL 区,每次执行一次 FAL(06) 00,另一个 FAL 错误代码将移到 FAL 区,并清除原先在那里的 FAL 错误代码。FAL 错误代码以数字次序记录。

FAL(06)00 也用来清除用指令 MSG(46)编程的信息。

如果 FAL 区不能被清除,通常这是当在 FALS(07)执行的时候,此时首先要消除错误的起因,然后用编程器或 CX—Programmer 清除 FAL 区。

5—15 步指令:步定义和步启动——STEP(08)/SNXT(09)



限制说明

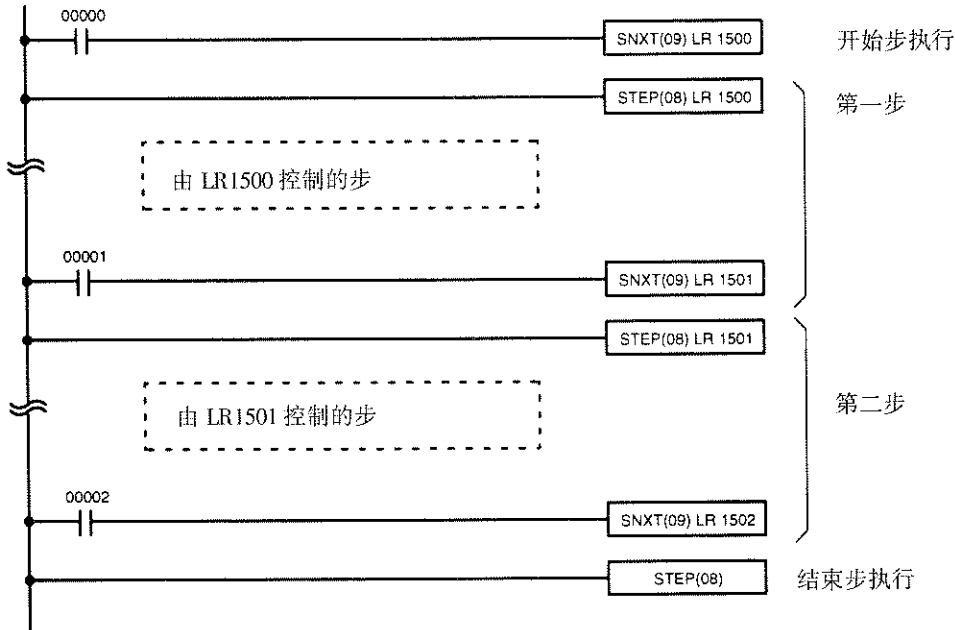
所有的控制位必须在同一个字中且必须连续。

步指令 STEP(08) 和 SNXT(09) 一起使用,以在大块程序的段间设立断点,从而使各程序段能作为一个单元来执行并在完毕后复位。通常按照应用中的实际过程来定义一个程序段(参照本节后面的应用例子)。一个步就象一个普通的编程代码,除一些特定的指令,(即 END(01), IL(02)/ILC(03), JMP(04)/(JME(05)和 SBN(92))不能包含在内。

STEP(08) 使用 IR 或 HR 区中的一个控制位来定义称为一个步的程序段的起始。STEP(08)不需要执行条件,即它的执行通过控制位来控制。为启动一个步的执行,要使用与 STEP(08)使用相同控制位的 SNXT(09)。如果 SNXT(09)以 ON 执行条件执行,具有同一控制位的步将执行。如果执行条件为 OFF,这一步就不执行。SNXT(09)指令必须写进程序中,以使它在程序到达它所启动的步之前能执行。它可以按照两个不同的执行条件来控制的步(见下面的例 2)之前用在不同的地方,程序中任何不用 SNXT(09)启动的步将不执行。

一旦程序中使用了 SNXT(09),步的执行要持续到 STEP(08)用没有控制位执行。没有控制位的 STEP(08)必须由带名义控制位的 SNXT(09)引导。名义控制位可以是任何未使用的 IR 或 HR 位。它不能是 STEP(08)中已使用的控制位。

一个步的执行通过执行下一个 SNXT(09) 或由步控制位转为 OFF 来完成 (见下面例 3)。当步完成时,此步中所有的 IR 和 HR 位都置为 OFF,所有定时器复位到它们的设定值。计数器,移位寄存器及 KEEP(11)中使用的位都保持其状态。两个简单的步如下所示。



| 地址 | 指令 | 操作数 |
|---------------|----------|---------|
| 00000 | LD | 00000 |
| 00001 | SNXT(09) | LR 1500 |
| 00002 | STEP(08) | LR 1500 |
| 由 LR1500 控制的步 | | |
| 00100 | LD | 00001 |
| 00101 | SNXT(09) | LR 1501 |

| 地址 | 指令 | 操作数 |
|---------------|----------|---------|
| 00102 | STEP(08) | LR 1501 |
| 由 LR1500 控制的步 | | |
| 00200 | LD | 00002 |
| 00201 | SNXT(09) | LR 1502 |
| 00202 | STEP(08) | --- |

可以连续地编写步程序,每个步必须以 STEP(08) 起始,通常以 SNXT(09) 结束 (例外情况见下面的例 3)。当串联形式编写步程序时,可能有三种执行方式:顺序、分支或平行方式。SNXT(09) 的执行条件和位置决定程序步如何执行。下面给出的三个例子说明了步程序的三种执行方式。

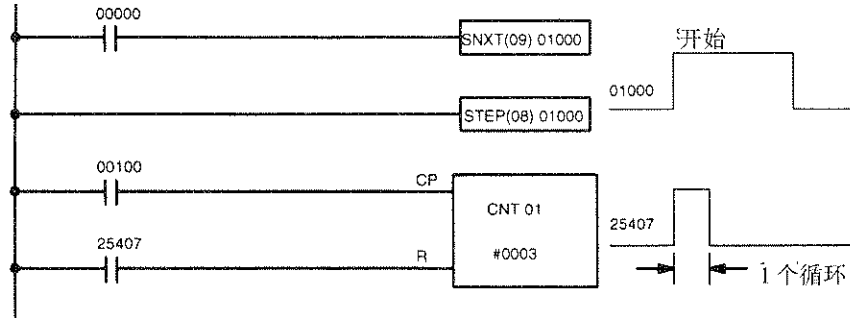
注意

在步程序内不能使用联锁、跳转、SBN(92)和 END(01)指令。

作为控制位使用的位不能在程序的其它地方使用,除非它们作为控制步操作的位使用 (见下面的例 3)。所有的控制位必须在同一字中且必须连续。如果 IR 或 LR 位用作控制位,在电源中断期间它们的状态将会丢失。如果必须保持状态,从同一步恢复执行,那么必须使用 HR 位。

标志

25407: 步起始标志; 当 STEP(08) 执行时置为 ON 一个循环, 如果需要, 可以如下所示用来在步内复位计数器。



| 地址 | 指令 | 操作数 |
|-------|----------|-------|
| 00000 | LD | 00000 |
| 00001 | SNXT(09) | 01000 |
| 00002 | STEP(08) | 01000 |
| 00003 | LD | 00100 |

| 地址 | 指令 | 操作数 |
|-------|-----|--------|
| 00004 | LD | 25407 |
| 00005 | CNT | 01 |
| | | # 0003 |

5—16 定时器和计数器指令

TIM 和 TIMH(15)是需要一个 TIM/CNT 编号和一个设定值(SV)的递减式接通延时定时器指令。STIM(69)用于控制激活中断子程序的间隔定时器。

CNT 是递减式计数器指令, CNTR(12)是可逆式计数器指令。它们都需要一个 TIM/ CNT 编号和一个设定值。它们也都与作为输入信号和复位信号的多个指令行相连。CTBL(63)、INT(89)和 PRV(62)用来管理高速计数器。INT(89)也用来停止脉冲输出。

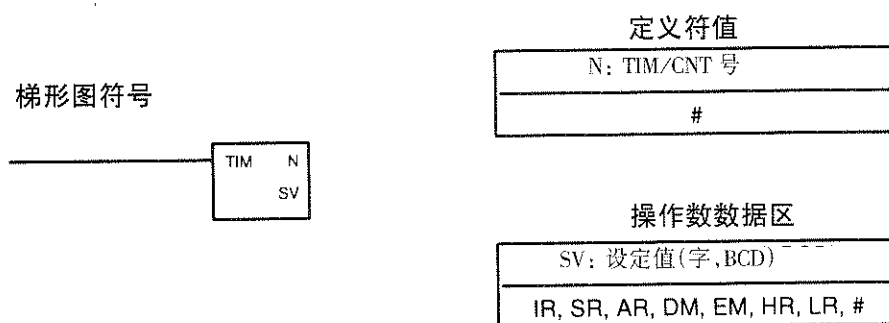
任何一个 TIM/ CNT 编号不能定义两次, 即一旦已经作为任一定时器或计数器指令的定义符使用过, 就不能被再次定义。一旦被定义, 只要需要 TIM/CNT 编号作为指令操作数而不是定时器或计数器指令, 可以使用无数次。

TIM/CNT 编号范围从 000 ~ 511。当用 TIM/CNT 编号作为定时器或计数器指令的定义符时, 不需要加前缀。一旦定义为定时器, 作为特定指令的操作数, TIM/ CNT 编号可以加前缀 TIM。尽管这个定时器指令曾用来定义这个定时器, 仍可用此 TIM 前缀, 一旦定义为计数器, TIM/ CNT 编号可以加前缀 CNT 作为特定指令的操作数。尽管这个计数器指令曾用来定义计数器, 而仍可使用这个 CNT。

TIM/ CNT 编号可以指定作为需要位或字数据的操作数。当指定作为需要位数据的操作数时, TIM/CNT 编号访问表示定时/计数已经达到的“完成标志”位, 即常开接点, 当到达设定值(SV)时为 ON。当指定作为需要字数据的操作数时, 此 TIM/CNT 编号访问保存定时或计数器当前值(PV)的存储位置。因此, 定时器或计数器的当前值可以作为比较指令 CMP(20)的操作数, 也可以作为 TIM/ CNT 区允许的其它指令的操作数。这通过指定用来定义定时器或计数器的 TIM/CNT 编号来访问保存 PV 值的存储器位置。

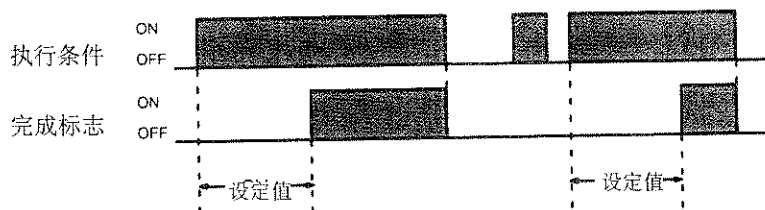
注意“TIM 000”用来指定一个用 TIM/CNT 编号 000 定义的定时器指令,指定此定时器的完成标志,并指定此定时器的 PV。本文中这个术语的意思应该是清楚的,即首先它总是一条指令,其次它总是一个位操作数,第三它也是一个字操作数。对于所有其它用 TIM 或 CNT 作前缀的 TIM/CNT 号都是一样的。设定值可以以常数或数据区的字地址输入,如果分配给输入单元的 IR 区的字被指定为字地址,输入单元的 IR 区的字被指定为字地址,输入单元可以通过适当的接线使能经过外部的拨盘开关或类似器件来设置 SV 值。用这种方法接线的定时器或计数器只能在运行或监视状态下,从外部设定,所有的设定值,包括从外部设定的,必须为 BCD 码形式。

5—16—1 定时器——TIM



限制

SV 在 000.0—999.9 之间。小数点不必输入。
 EM 区仅在 CQM1H—CPU61 CPU 单元中有效。
 TIM/CNT 编号只能在一个定时器或计数器指令中作为定义符使用。
 在 TIM 中不应使用 TIM/CNT000~TIM/CNT015, 如果它们在 TIMH(15) 中需要, 详情参阅 5—16—4 高速定时器—TIMH(15)。
 当它的执行条件为 ON 时, 定时器被启动; 当执行条件为 OFF 时, 它被复位(到设定值)。一旦被启动, 定时器从设定值以 0.1 秒为单位计数。
 如果 TIM 的执行条件保持为 ON 有足够时间, 直到时间值降到零。所使用的 TIM/CNT 编号的完成标志将转为 ON, 并保持 ON 直到 TIM 复位(即直到它的执行条件转为 OFF)。
 下图表示 TIM 的执行条件和分配给它的完成标志之间的关系。

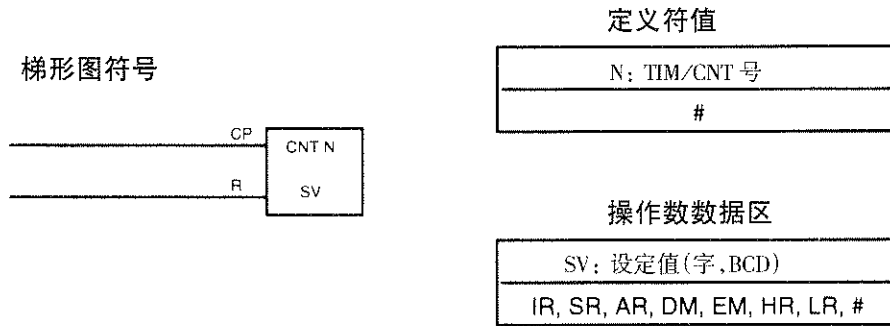


注意

当 IL(02)的执行条件为 OFF 时,在联锁程序段中的定时器将复位。电源中断也将使定时器复位。如果需要定时器在这些条件下不复位, 可以用 SR 区的时钟脉冲位对 CNT 进行计数来产生定时器。详情请参考 5—16—2 计数器—CNT。

标志 ER: 设定值不是 BCD 码形式。
 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)

5-16-2 计数器——CNT

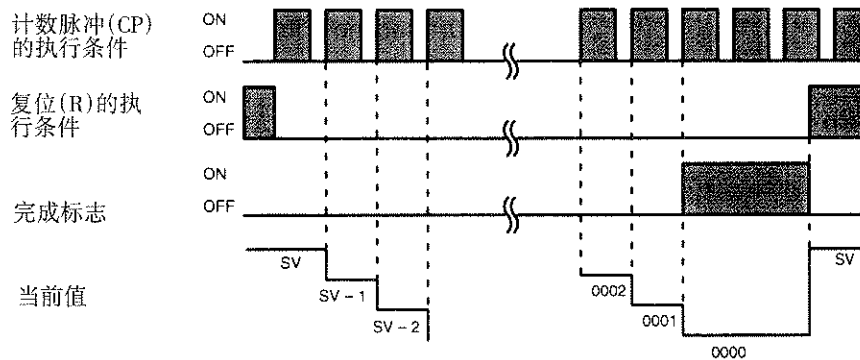


限制 每个 TIM/CNT 编号只能在一个定时器或计数器指令中作为定义符使用。
 EM 区仅在 CQM1H—CPU 6I CPU 单元中有效。

说明 当计数脉冲 (CP) 的执行条件从 OFF 变为 ON 时, 计数器从设定值 SV 减计数, 即当 CNT 的执行条件 CP 为 ON, 而上一次为 OFF 时, 当前值减 1。如果执行条件没有改变, 或从 ON 变为 OFF, 计数器的当前值不改变。在当前值到达零时, 计数器的完成标志置为 ON, 并保持为 ON 直到计数器复位。

计数器用复位输入 R 复位。当 R 从 OFF 变为 ON, 当前值复位到设定值。当 R 为 ON 时, 当前值不递减。当 R 为 OFF 时, CNT 的 PV 值又从设定值向下计数。在连锁程序段中或电源中断时, 计数器的当前值不复位。

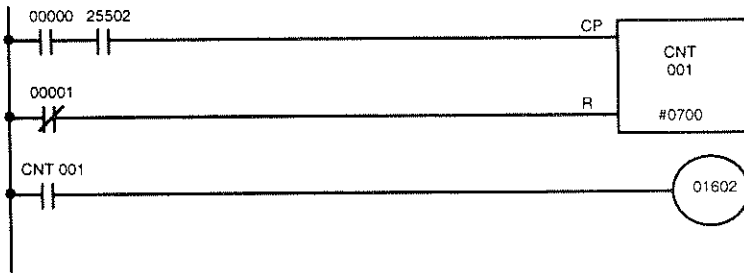
执行条件、完成标志和当前值 (PV) 的变化如下图所示。PV 线高度只是表示当前值的变化。



注意 即使设定值未使用 BCD 码, 程序也将继续执行, 但是设定值 SV 将不正确。
 标志 ER: 设定值 (SV) 不是 BCD 码形式。
 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 码形式, 或者超出了 EM/DM 区的边界)。

例

在下面的例子中,使用 CNT 用计数 SR 区时钟脉冲位以产生扩展定时器。
 CNT001 计数 1 秒时钟脉冲位(SR25502)从 OFF 变为 ON 的次数,同样地,当计数器在运行时,使用 IR00000 来控制次数。
 因为本例中 CNT001 的设定值为 700, 当过了 1 秒 × 700 次或 11 分 40 秒后, CNT002 的完成标志置为 ON。这将导致 IR01602 置为 ON。

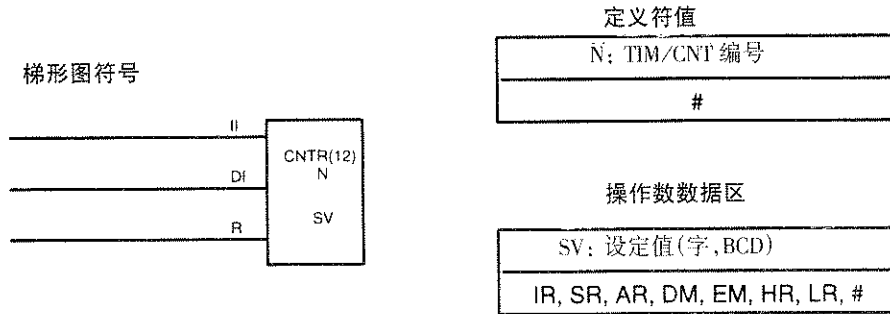


| 地址 | 指令 | 操作数 |
|-------|--------|---------------|
| 00000 | LD | 00000 |
| 00001 | AND | 25502 |
| 00002 | LD NOT | 00001 |
| 00003 | CNT | 001 # 0700 |
| 00004 | LD | CNT 001 |
| 00005 | OUT | 01602 |

注意

较短的时钟脉冲不一定会产生精确的记时,因为在较长循环时,可能不能精确地读出时间较短的 ON 的次数。特别是 CNT 指令不能使用 0.02 秒和 0.1 秒时钟脉冲产生定时器。

5—16—3 可逆计数器—CNTR(12)



限制

每个 TIM/CNT 编号只能在一个定时器或计数器指令中作为定义符使用。

EM 区仅在 CQM1H—CPU61 CPU 单元中有效。

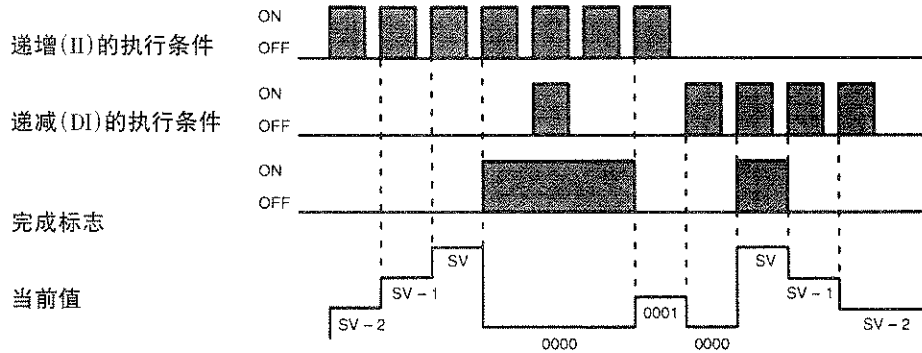
说明

CNTR(12) 是一个可逆的、加/减循环的计数器,即它可按照递增输入(II)和递减输入(DI)的执行条件的变化在 0 和设定值之间计数。

当 CNTR(12) 的 II 执行条件为 ON 而上一次的执行条件为 OFF 时,当前值(PV)加 1。当 CNTR(12)的 DI 执行条件为 ON 而上一次的执行条件为 OFF 时,当前值(PV)减 1。如果在 II 和 DI 同时出现 OFF 到 ON 的改变,PV 将不改变。如果 II 和 DI 的执行条件不改变或从 ON 变为 OFF,计数器的当前值不改变。当从 0000 往下减时,当前值将设置为设定值,并且完成标志置为 ON,直到当前值再递减为止。当递增到超过设定值时,当前值设置为 0000,完成标志置为 ON,直到当前值重新递增为止。

CNTR(12)用复位输入 R 复位。当 R 从 OFF 变为 ON,当前值复位到 0。当 R 为 ON 时,当前值不递增又不递减。当 R 为 OFF 时,计数将重新开始。在联锁程序段中或受电源中断影响 CNTR(12)的当前值不复位。

II 和 DI 的执行条件,完成标志和当前值的变化如下图所示,它是 CNTR(12)的一部分(即,当复位时,从零计数)。PV 线高度只是表示当前值的变化。



注意

即使设定值未使用 BCD 码,程序也将继续执行,只是设定值将不正确。

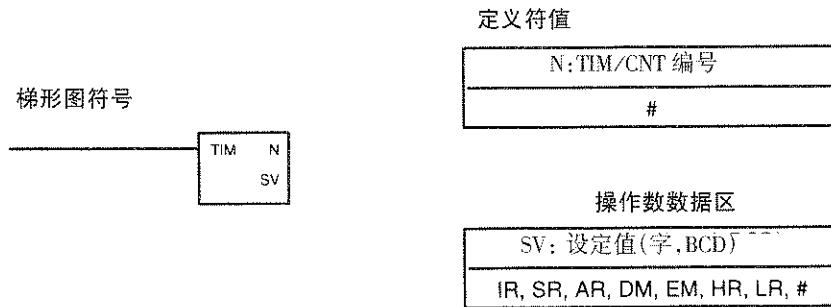
标志

ER: 设定值不是 BCD 码形式。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 形式,或者超出 EM/DM 区边界)

5-16-4 高速定时器—TIMH(15)



限制

设定值的范围在 00.00 和 99.99 之间。(虽然可以设置为 00.00 和 00.01,但是 00.00 会使定时器不工作,它会立刻将完成标志置为 ON;而 00.01 也不能可靠地被扫描。)小数点不必输入。

每个 TIM/ CNT 编号只能在一个定时器或计数器指令中作为定义符使用。使用 TIM/ CNT 号 000 ~ 015。如果循环时间超过 10ms,高速定时器不要使用定时器号 TIM/ CNT016 ~ TIM/ CNT511。

说明

除了 TIMH 以 0.01 秒为单位计数外, TIMH(15) 与 TIM 运行方式相同,操作详情请参考 5-16-1 定时器—TIM。

注意

当 IL(02)的执行条件为 OFF 时,联锁程序段中的定时器复位。电源中断也使定时器复位。如果要求定时器在这些条件下不复位,可以用 SR 区的时钟脉冲位来计数以产生使用计数器的定时器。详情请参考 5-16-2 计数器—CNT。

当 JMP(04) 的执行条件为 OFF 时, 跳转程序段中的定时器不复位, 但是如果使用跳转号 00, 定时器将停止计时。如果使用 01 ~ 99 之间的跳转号, 定时器将继续计时。

使用 TIM/CNT000~TIM/CNT015 定时器编号的高速定时器, 当 PC 设置(DM6629) 为这些定时器执行中断处理时, 它们能精确计时。

当循环时间超过 10ms 时, 使用 TIM/CNT016~TIM/CNT511 定时号的高速定时器将不能精确计时。如果循环时间大于 10ms, 要使用 TIM/CNT000~TIM/CNT015 并设置 DM6629 为所使用的定时器号进行中断处理。

标志

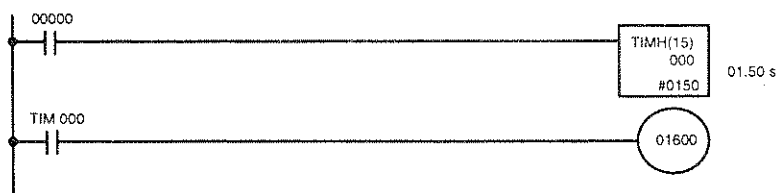
ER: 设定值不是 BCD 码形式。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 形式, 或者已超出 EM/DM 区边界)。

例

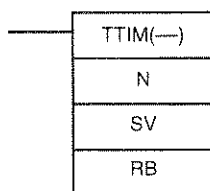
下面的例子表示一个用常数设置的定时器。00000 为 ON 且至少保持 1.5 秒后, 01600 将置为 ON。当 00000 为 OFF 时, 定时器将复位, 并且 01600 也将被置为 OFF。



| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00000 |
| 00001 | TIMH(15) | 000 |
| | | # 0150 |
| 00002 | LD | TIM 000 |
| 00003 | OUT | 01600 |

5-16-5 累加定时器—TTIM(—)

梯形图符号



定义符值

| |
|----------------|
| IN: TIM/CNT 编号 |
| # (000 ~ 511) |

操作数数据区

| |
|------------------------|
| SV: 设定值(字, BCD) |
| IR, AR, DM, EM, HR, LR |

RB: 复位位

| |
|--------------------|
| IR, SR, AR, HR, LR |
|--------------------|

限制

设定值在 0000 ~ 9999 之间并且是 BCD 形式。小数点不输入。

EM 区仅在 CQM1H—CPU61 CPU 单元有效。

说明

每个 TIM/CNT 编号只能在一个定时器或一个计数器指令中作为定义符使用。TTIM(—) 用于产生一个每 0.1S 递增当前值的定时器。定时时间在 0.1S ~ 999.9S 之间。TTIM(—) 以 0.1S 为单位从零向上加。TTIM(—) 的精确度为 + 0.0/ - 0.1S。只要执行条件为 ON, TTIM(—) 定时器就开始定时直到它达到 PV 或者直到 RB 变 ON 以复位定时器。TTIM(—) 定时器只在每个循环执行时

计时,即在联锁程序段中或当程序跳过时他们并不计时,但仍保持当前的PV。

注 减操作定时器(如 TIM)的当前值表示直到定时器定时到的剩余时间,但是 TTIM(—) 定时器的当前值则表示已流逝的时间。TTIM(—) 的当前值在计算和显示中可以“就这样”使用来表示流逝的时间。

注意

当电源断开或者 PC 由“编程”模式切换到“监控”或“运行”模式(或相反)时,当前值将复位到 000,并且完成标志将变为 OFF。

当 IL(O2) 执行条件为 OFF 时,联锁程序段中的 TTIM(—) 的当前值保持原值。不同于定时器和高速定时器继续计时,当前值在跳转程序段中也保持原值。

如果循环时间超过 0.1S,那 TTIM(—) 将不可能很好地运行,因为仅当执行 TTIM(—) 时,才刷新当前值,并且当前值以 0.1S 单位增加。

在定时器计时到后,完成标志变 ON 可能有一个循环的延迟,因为完成标志仅在 TTIM(—) 执行时刷新。

定时结束后,TTIM(—) 将不再重新启动,除非当前值变为小于 SV 的值或者复位输入变 ON。

标志

ER:N 不是 TIM 号。

SV 不是 BCD。

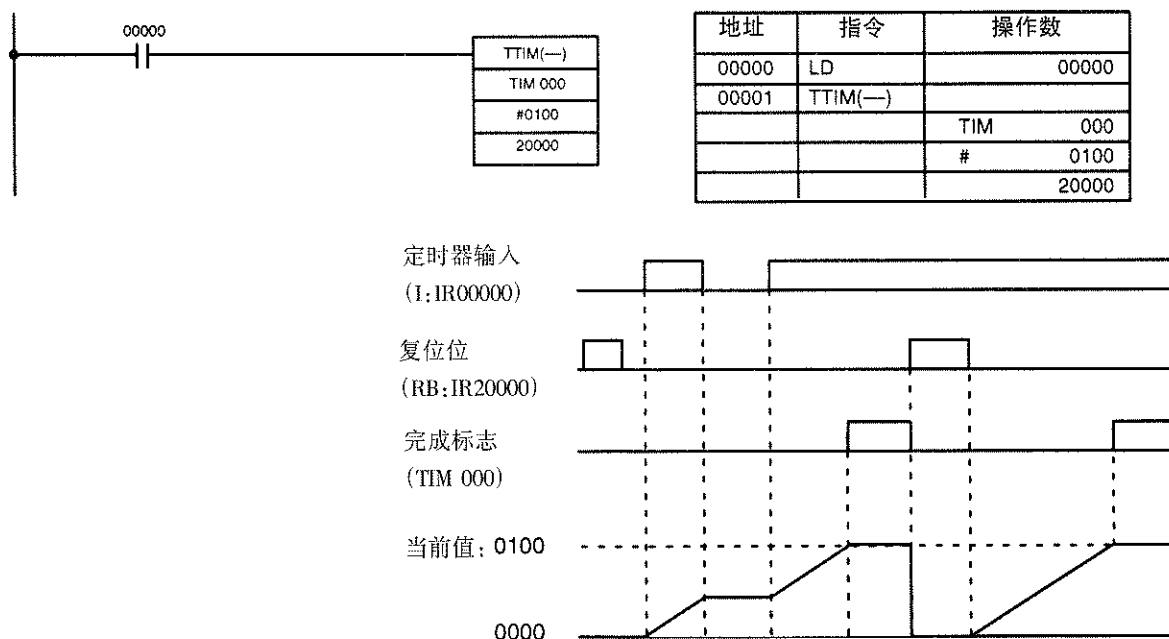
RB 不是有效的位地址。

间接寻址的 EM/DM 字不存在。

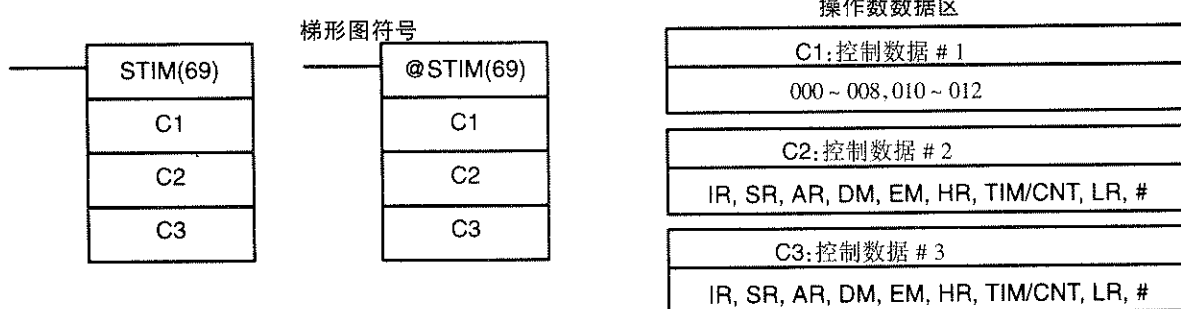
(* EM/* DM 字不是 BCD,或已超出 EM/DM 区边界。)

例

下图示出了累加定时器(设定值 2S) 执行条件、当前值、和完成标志之间的关系。



5-16-6 间隔定时器—STIM(69)



限制

C1 必须在 000 ~ 008 或者 010 ~ 012 之间。

如果 C1 在 000 ~ 005 之间,那么 C3 不能使用大于 0255 的常数。
 如果 C1 在 006 ~ 008 之间,C2 或 C3 不能使用常数和 DM6143 ~ DM6655, 如果 C1 在 010 ~ 012 之间,那么 C2 和 C3 都设为 000。

说明

STIM(69) 通过执行四个基本功能控制间隔定时器: 启动单发中断定时器、启动定时中断定时器、停止定时器和读定时器的当前值。设定 C1 的值以确定执行哪个功能, 三个间隔定时器哪一个执行, 如下表所示。使用间隔定时器中断的详情请参照 1-4-4 间隔定时器中断。表后也详细地介绍了 STIM(69)。

| 功 能 | 定 时 器 | C1值 |
|--------|-------|-----|
| 启动定时器 | 0 | 000 |
| | 1 | 001 |
| | 2 | 002 |
| 启动定时中断 | 0 | 003 |
| | 1 | 004 |
| | 2 | 005 |
| 读定时器PV | 0 | 006 |
| | 1 | 007 |
| | 2 | 008 |
| 停止定时器 | 0 | 010 |
| | 1 | 011 |
| | 2 | 012 |

注

1. 当脉冲输出是通过 SPED(64)指令输出时,不能使用间隔定时器 0。
2. 当 PC 设置的 DM6642 中已允许高速计数器 0 工作时, 不能使用间隔定时器 2。

启动中断

设置 C1 = 000 ~ 002 以启动定时器 0 ~ 2,从而激活单发中断。设置 C1 = 003 ~ 005 以启动定时中断,使用定时器 0 ~ 2。

C2 指定定时器的设定值,可以是一个常数,或是包含设定值的两个字中的第一个字。对于所使用的不同方法,设定有略微的不同。

如果 C2 是一个常数,它指定递减计数器的初始值 (BCD, 0000 ~ 9999)。递减时间间隔为 1ms。

如果 C2 是一个字地址, C2 指定递减计数器的初始值 (BCD, 0000 ~ 9999), C2 + 1 指定以 0.1ms 为单位的递减时间间隔 (BCD, 0005 ~ 0320)。因此递减时间间隔可以为 0.5 ~ 32ms。

C3 指定子程序编号为 0000 ~ 0255。

注 从间隔定时器启动到结束所需要的时间为： $(C2 \text{ 的内容}) \times (C2 + 1 \text{ 的内容}) \times 0.1\text{ms}$ 。

读定时器 PV

设置 C1 = 006 ~ 008 读定时器 0 ~ 2 的当前值。

C2 指定接收定时器的当前值的两个目的字的第一个字。C2 接收递减计数器递减的次数 (BCD, 0000 ~ 9999), C2 + 1 接收递减时间间隔 (BCD, 以 0.1ms 为单位)。

C3 指定接收从上次定时器递减后所经过的时间的目的字 (BCD, 以 0.1ms 为单位)。

(必须等于或小于 C2 + 1 内设置的递减时间间隔)

注 从启动定时器开始所经过的时间可以作如下计算：
 $[(C2 \text{ 的内容}) \times (C2 + 1 \text{ 的内容}) + (C3 \text{ 的内容})] \times 0.1\text{ms}$

停止定时器

设置 C1 = 010 ~ 012 以停止定时器 0 ~ 2

C2 和 C3 没有意义, 都应设为 000。

标志

ER: 当脉冲输出运行时, 启动了间隔定时器 0。

(只对于 C1 = 000)

当允许高速计数器 0 时, 启动了间隔定时器 2。

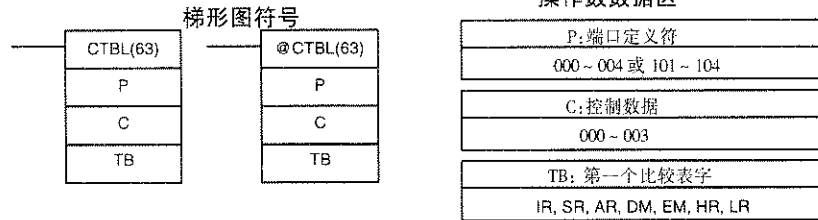
(只对于 C1 = 002)

间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)

一个数据区超出边界范围。

5-16-7 寄存器比较表—CTBL(63)



限制

第一个和最后一个比较表字必须在同一数据区内。(比较表的长度因设置而不同)。

如果 PC 设置 (DM6611) 被设置为脉冲输出模式, 那就不能使用 CTBL(63)。

说明

CTBL(63) 用于寄存器比较表并启动高速计数器的比较。下表显示 CTBL(63) 的功能。

| 单元/板 | 功 能 |
|----------|-----------------------------|
| CPU 单元 | 高速计数器 0 (IR00004 ~ IR00006) |
| 脉冲 I/O 板 | 高速计数器 1 和 2 |
| 绝对编码器接口板 | 绝对高速计数器 1 和 2 |
| 高速计数器板 | 高速计数器 1 ~ 4 |

CTBL(63) 操作说明分为二部分。对 CPU 单元, 脉冲 I/O 板, 及绝对编码器接口板的操作描述见 233 页。

CPU 单元, 脉冲 I/O 板, 及绝对编码器接口板

详细情况参考 238 页用高速计数器板的 CTBL(63)操作。

当执行条件为 OFF 时, CTBL(63)不执行。当执行条件为 ON 时, CTBL(63)登录一个用于高速计数器当前值的比较表。根据 C 的值, 与高速计数器的当前值进行比较可以立即启动或者可以用 INI(61)独立启动。

端口定义符(P)指定将用于比较的高速计数器。

| 单元/板 | 功能 | 端口定义符(P) |
|--------------------|------------|----------|
| CPU单元 | 高速计数器0(内置) | 000 |
| 脉冲I/O板 (见注1和注2) | 高速计数器1 | 001 |
| | 高速计数器2 | 002 |
| 绝对编码器接口板 (见注1) | 高速计数器1 | 001 |
| | 高速计数器2 | 002 |

- 注
1. 脉冲 I/O 板和绝对编码器接口板必须安装在槽 2 中。
 2. 当使用一个脉冲 I/O 板时, 在 PC 设置的 DM6611 中, 口 1 和口 2 的模式必须被设置成高速计数器模式。如果把模式设置为简单定位控制模式, 那么就不能使用 CTBL(63)。

CTBL(63)的功能由控制数据 C 决定, 如下表所示。这些功能在表后说明。

| C | CTBL(63)功能 |
|-----|--------------------------|
| 000 | 登录一个目标值比较表并开始比较。 |
| 001 | 登录一个范围比较表并开始比较。 |
| 002 | 登录一个目标值比较表。用INI(61)开始比较。 |
| 003 | 登录一个范围比较表。用INI(61)开始比较。 |

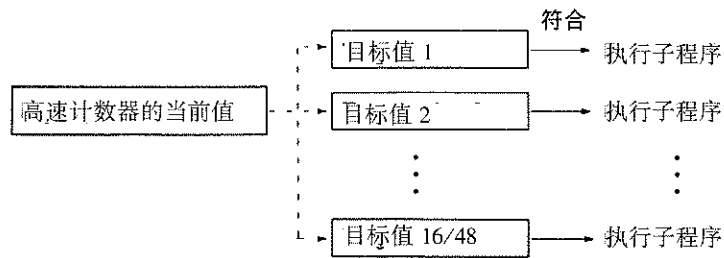
在当前值与目标值相一致或落在指定的范围内, 那么指定的子程序被调用和执行。有关比较表的详细情况参考 1-4-5 高速计数器 0 中断。

如果在 PC 设置 (DM6642) 中允许高速计数器, 当 CQM1H 投入运行时它将从零开始计数。当前值将不与比较表进行比较, 直到表被登录并且由 INI(61)或 CTBL(63)开始比较。INI(61)可以停止比较和起启动比较, 或者可复位当前值。一旦比较表已登录, 它就是有效的, 直到 CQM1H 暂停或在试图登录一张新表时发生错误。当要减少循环时间时, 建议使用 CTBL(63)的微分形式。

一旦比较表已登录, 它就是有效的, 直到 CQM1H 暂停或在试图登录一张新表时发生错误。当要减少循环时间时, 建议使用 CTBL(63)的微分形式。

目标值比较

对于 CPU 单元中的高速计数器 0, 最多可登录 16 个目标值。对应于各个目标值, 也可登录一个子程序编号 (1 ~ 16)。对于脉冲 I/O 板或绝对编码器接口板中的高速计数器 1 和 2, 最多可登录 48 个目标值。对应于各个目标值也可登录一个子程序编号 (1 ~ 48), 在任一种情况时, 在当前值与目标值相等时调用相应的子程序并执行子程序。(当不需要中断处理时, 可以输入未定义的子程序号)。



按照比较表的次序目标值比较一次完成一个项目。在当前值达到表中第一个目标值时,执行中断子程序并且比较继续到表中的下一个值。当完成了表中的最后一个目标值时,比较返回到表中第一个值并且重复此过程。

下图显示使用 CPU 单元内置高速计数器 0 或用于线性计数的脉冲 I/O 板高速计数器 1 或 2 的目标值比较表的结构。目标值编号可以是 0001 ~ 0048。

| | | |
|--------|-----------------------|-------------|
| TB | 目标值编号(BCD) | } → 一个目标值设定 |
| TB + 1 | 目标值 # 1, 低 4 位数字(BCD) | |
| TB + 2 | 目标值 # 1, 高 4 位数字(BCD) | |
| TB + 3 | 子程序号(见注1) | |
| | | |
| | | |
| | | |
| | | |

下图显示在使用脉冲 I/O 板的高速计数器 1 和 2 的环形计数时目标值比较表的结构。按升序或降序输入目标值。

环值指定环中的点数和最大的计数值(环值 = 最大计数值 + 1)。环值可为 0 ~ 65,000。当比较正在进行时,不要改变环值。

| | | |
|--------|-----------------------|-------------|
| TB | 环值, 低4位数字(BCD) | } → 环值设定 |
| TB + 1 | 环值, 高4位数字(BCD) | |
| TB + 2 | 目标值数(BCD) | } → 一个目标值设定 |
| TB + 3 | 目标值 # 1, 低 4 位数字(BCD) | |
| TB + 4 | 目标值 # 1, 高 4 位数字(BCD) | |
| TB + 5 | 子程序号(见注1) | |
| | | |
| | | |
| | | |
| | | |

下图显示使用绝对编码器接口板的高速计数器 1 和 2 的目标值比较表的结构。按上升次序或下降次序输入目标值。目标值数可以是 0001 ~ 0048。

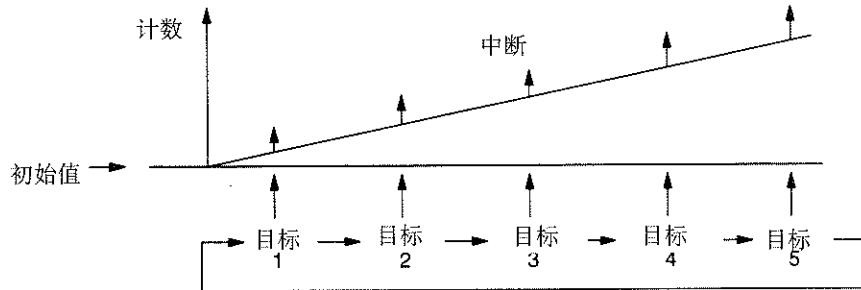
| | | |
|--------|--------------|-------------|
| TB | 目标值数(BCD) | } → 一个目标值设定 |
| TB + 1 | 目标值 # 1(BCD) | |
| TB + 2 | 子程序号(见注1) | |
| | | |
| | | |
| | | |
| | | |

注 1. 当递减时,可以用子程序号 F000 ~ F255 启动子程序。当递增时,可以用子程序号 0000 ~ 0255 启动子程序。

2. 当为高速计数器 1 和 2 设定目标值时, 允许中断处理的时间间隔至少为 0.2ms。

目标值比较操作

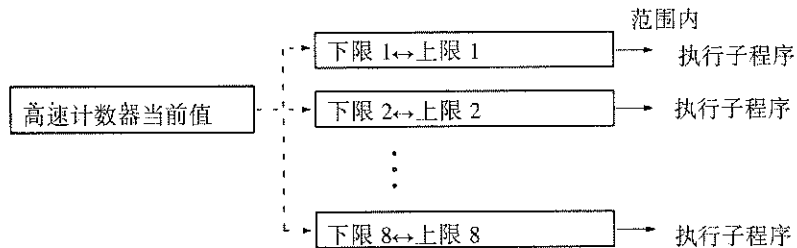
下图显示在比较表中连续地设定 1~5 目标值的目标值比较操作。



如上图所示, 当前的计数与登录在目标值比较表中的每个目标值按次序比较, 当计数与当前目标值相同时产生一个中断, 并且开始与下一个目标值的比较。当比较表中所有目标值都已经相符并产生了中断, 目标值返回到表中第一个目标值并且重复此运行。

范围比较

范围比较表包括 8 个范围, 它们由 8 个下限数字和 8 个上限数字, 以及与它们对应的子程序编号。当当前值落在给定的范围内时, 相应的子程序被调用并且执行。(当不需要中断处理时, 可以输入一个未定义的子程序编号。)



总是设置 8 个范围。如果需要 8 个以下的范围, 那么设置其余的子程序号为 FFFF。如果需要 8 个以上的范围, 可以使用另一个比较指令如 BCMP(68) 以将范围和 IR230~IR235 中的高速计数器当前值进行比较。记住: 这些字每个循环刷新一次。

当高速计数器的当前值落在八个范围中的一个或多个范围中时, AR 区中的标志会标明。当当前值落在相应的范围时标志变 ON。

| 计数器 | AR区标志 |
|--------|-------------------------|
| 高速计数器0 | AR1100 ~ AR1107对应范围1~8。 |
| 高速计数器1 | AR0500 ~ AR0507对应范围1~8。 |
| 高速计数器2 | AR0600 ~ AR0607对应范围1~8。 |

下图显示使用 CPU 单元内置高速计数器 0 或用于线性计数的脉冲 I/O 板高速计数器 1 或 2 的范围比较表的结构。

| | | |
|-------|-----------------------|-----------|
| TB | 下限 # 1, 低 4 位数字 (BCD) | } 第一个范围设定 |
| TB+1 | 下限 # 1, 高 4 位数字 (BCD) | |
| TB+2 | 上限 # 1, 低 4 位数字 (BCD) | |
| TB+3 | 上限 # 1, 高 4 位数字 (BCD) | |
| TB+4 | 子程序号 (见注 1) | |
| ⋮ | ⋮ | |
| TB+35 | 下限 # 8, 低 4 位数字 (BCD) | } 第八个范围设定 |
| TB+36 | 下限 # 8, 高 4 位数字 (BCD) | |
| TB+37 | 上限 # 8, 低 4 位数字 (BCD) | |
| TB+38 | 上限 # 8, 高 4 位数字 (BCD) | |
| TB+39 | 子程序号 (见注 1) | |

下图显示使用脉冲 I/O 板高速计数器 1 或 2 设置为环形计数的范围比较表的结构。环值指定环中的点数和最大计数值 (环值 = 最大计数值 + 1); 环值的设定范围是 0 ~ 65,000, 当比较正在进行时, 不要改变环值。

| | | |
|-------|-----------------------|-----------|
| TB | 环值, 低 4 位数字 (BCD) | } 环值设定 |
| TB+1 | 环值, 高 4 位数字 (BCD) | |
| TB+3 | 下限 # 1, 低 4 位数字 (BCD) | } 第一个范围设定 |
| TB+4 | 下限 # 1, 高 4 位数字 (BCD) | |
| TB+5 | 上限 # 1, 低 4 位数字 (BCD) | |
| TB+6 | 上限 # 1, 高 4 位数字 (BCD) | |
| TB+7 | 子程序号 (见注 1) | |
| ⋮ | ⋮ | |
| TB+37 | 下限 # 8, 低 4 位数字 (BCD) | } 第八个范围设定 |
| TB+38 | 下限 # 8, 高 4 位数字 (BCD) | |
| TB+39 | 上限 # 8, 低 4 位数字 (BCD) | |
| TB+40 | 上限 # 8, 高 4 位数字 (BCD) | |
| TB+41 | 子程序号 (见注 1) | |

下图显示使用绝对编码器接口板高速计数器 1 和 2 的范围比较表的结构。

| | | |
|-------|--------------|-----------|
| TB | 下限 # 1 (BCD) | } 第一个范围设定 |
| TB+2 | 上限 # 1 (BCD) | |
| TB+4 | 子程序号 (见注 1) | |
| ⋮ | ⋮ | |
| TB+21 | 下限 # 8 (BCD) | } 第八个范围设定 |
| TB+22 | 上限 # 8 (BCD) | |
| TB+23 | 子程序号 (见注 1) | |

- 注
1. 子程序号可以是 0000 ~ 0255 并且只要计数器的当前值在指定的范围内, 子程序就会执行。FFFF 值表示无子程序要执行。
 2. 对高速计数器 1 和 2 进行范围比较时, 在下限和上限之间允许至少有 2ms 的时间间隔 (上限 - 下限) $0.002 \times$ 输入脉冲频率)。
- 下表显示目标值、下限值和上限值的可能值。最高位的十六进制数 F 表明此值为负。

| 计数器 | 可能值 |
|--------------------------|---|
| 高速计数器0(CPU 单元) | 相位差模式: F003 2768 ~ 0003 2767 递增模式: 0000 0000 ~ 0006 5535 |
| 高速计数器1和2(脉冲I/O板) | 线性计数:F838 8607 ~ 0838 8608 环形计数:0000 0000 ~ 0006 4999 |
| 绝对高速计数器1和2 (绝对编码器接口板) | BCD模式:0000 ~ 4095 360°模式:0000 ~ 0355(单位为5°) |

在 360°模式下,高速计数器的角度值在内部转换为二进制值。转换后的二进制值取决于 PC 设置中所选择的分辨率(DM6643 和/或 DM6644)。下表示出了 5°—45°之间转换后的值。

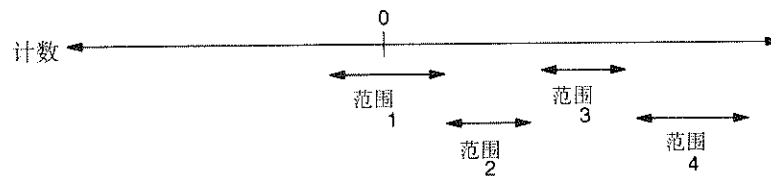
| 分辨率 | 转换后的值 | | | | | | | | |
|-------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| | 5° | 10° | 15° | 20° | 25° | 30° | 35° | 40° | 45° |
| 8位(0~255) | 4 | 7 | 11 | 14 | 18 | 21 | 25 | 28 | 32 |
| 10位(0~1023) | 14 | 28 | 43 | 57 | 71 | 85 | 100 | 114 | 128 |
| 12位(0~4095) | 57 | 114 | 171 | 228 | 284 | 341 | 398 | 455 | 512 |

对于更高的数值,从表中找出最接近 45°的转换值并加上余值。例如,将 145°转换为 8 位分辨率: 32×3 (对 135°) + 7(对 10°) = 103

注意 对于 10 位和 12 位分辨率,当角度值符合比较值时,因为转换后的值不能准确地匹配,可能不会触发中断处理。

范围比较操作

下图显示在比较表中连续设定 1~4 范围设定的范围比较的操作



如上图所示,当前计数同时与所有比较范围进行比较,并且输出每个范围的结果。

AR 区标志

下列 AR 区标志表明 CPU 单元中的高速计数器 0 及脉冲 I/O 板或绝对编码器接口板内的高速计数器 1 和 2 的比较运行的状态。

| 字 | 位 | 运行 |
|------|-------|---|
| AR05 | 00~07 | 高速计数器1(口1)范围比较标志 当计数器当前值落在相应范围(1~8)内时,位00~07变ON。 |
| | 08 | 高速计数器1(口1)比较标志 比较当前值时,此标志变ON。 |
| | 09 | 高速计数器1(口1)上溢/下溢标志 当出现上溢或下溢时,此标志为ON。 |

| | | |
|------|---------|---|
| AR06 | 00 ~ 07 | 高速计数器2(口2)范围比较标志 当计数器当前值在相应范围(1~8)内时,位00~07将变ON。 |
| | 08 | 高速计数器2(口2)比较标志 当前值比较期间,此标志变ON。 |
| | 09 | 高速计数器2(口2)上溢/下溢标志 当出现上溢或下溢时,此标志将变ON。 |
| AR11 | 00 ~ 07 | 高速计数器0范围比较标志 当计数器当前值落在相应范围(1~8)内时,位00~07将变ON。 |

高速计数器板的工作

当执行条件为 OFF 时, CTBL(63) 不执行。当执行条件为 ON 时, CTBL(63) 登录一个用于高速计数器当前值的比较表。根据 C 的值, 与高速计数器当前值进行比较可以立即启动或者可以用 INI(61) 独立启动。

端口定义符(P)指定高数计数器板的哪一个高速计数器在比较中使用。

| 功能 | 端口定义符(P) | |
|--------|----------|-------|
| | 槽1中的板 | 槽2中的板 |
| 高速计数器1 | 101 | 001 |
| 高速计数器2 | 102 | 002 |
| 高速计数器3 | 103 | 003 |
| 高速计数器4 | 104 | 004 |

CTBL(63)的功能由控制数据 C 所决定,如下表所示。这些功能在表后说明。

| C | CTBL(63)功能 |
|-----|-------------------------|
| 000 | 登录一个目标值比较表并开始比较。 |
| 001 | 登录一个范围比较表并开始比较。 |
| 002 | 登录一个目标值比较表。用INI(61)开始比较 |
| 003 | 登录一个范围比较表。用INI(61)开始比较。 |

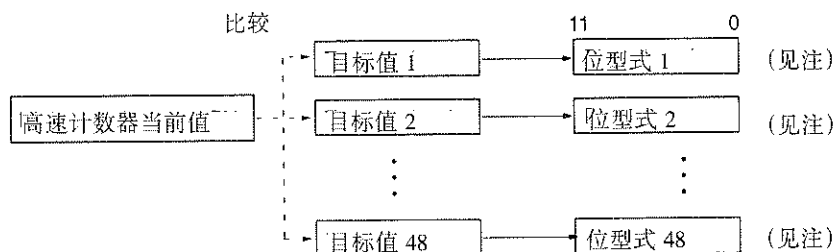
在当前值与目标值一致或落在指定范围内时, 位型式输出到分配的 IR 字中。有关表比较 的详细情况参考 1-4-5 高速计数器 0 中断。

如果在 PC 设置 (DM6642) 中允许高速计数器, 那么当 CQM1H 投入运行时, 它将从零开始计数。当前值将不与比较表进行比较, 直到表已登录, 并且用 INI(61) 或 CTBL(63) 启动比较。用 INI(61) 可以停止比较和起启动比较, 或者可复位当前值。

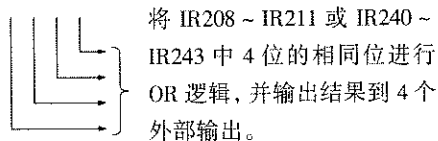
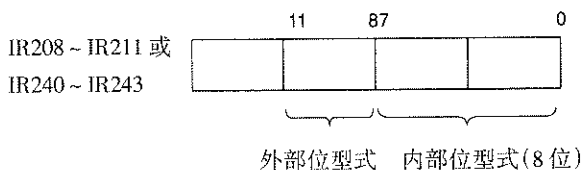
一旦比较表已登录, 它就是有效的, 直到 CQM1H 暂停或在试图登录一个新表时发生错误。当要减少循环时间时, 建议使用 CTBL(63) 的微分形式。

目标值比较

最多可登录 48 个目标值。对每个目标值还登录一个位型式。在当前值符合目标值时, 已登录的位型式输出到分配的 IR 字中。高速计数器板不产生中断, 登录的位型式反映到分配的 IR 字和外部输出中。



注 位型式 1 ~ 48 如下图示：



按照比较表的次序目标值比较一次进行一个项目。当当前值达到表中第一个目标值时, 位型式输出到分配的 IR 字中, 并且比较继续到表中的下一个值。当完成了表中的最后一个目标值时, 比较返回到表中第一个值并且重复此过程。

下图显示使用高速计数器 1 ~ 4 线性计数时的目标值比较表的结构。

| | | |
|------|-----------------------|-----------|
| TB | 目标值数(BCD) | } 一个目标值设定 |
| TB+1 | 目标值 # 1, 低 4 位数字(BCD) | |
| TB+2 | 目标值 # 1, 高 4 位数字(BCD) | |
| TB+3 | 位型式 # 1 | |
| ⋮ | ⋮ | |

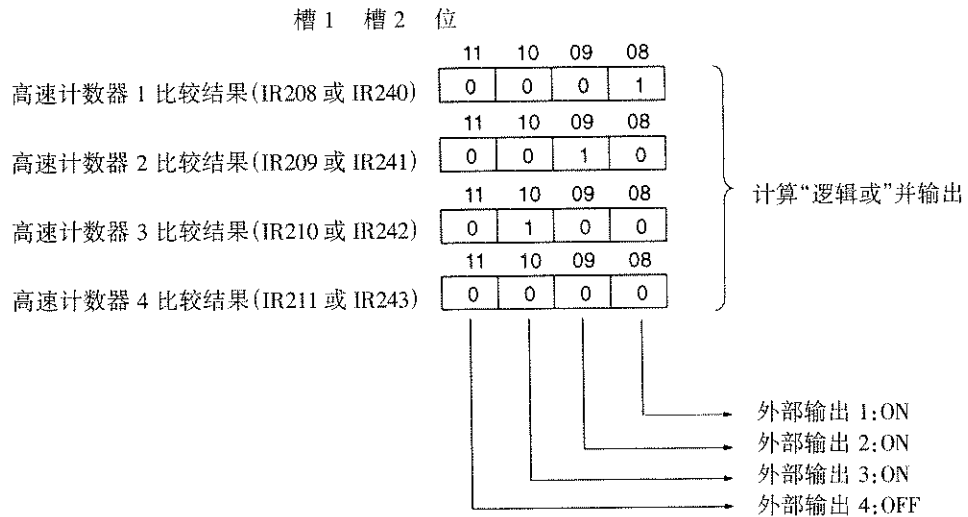
下图显示使用高速计数器 1 ~ 4 环形计数时的目标值比较表的结构。按升序或降序输入目标值。

环形值指定环中的点数和最大的计数值(环值 = 最大计数值 + 1)。当比较正在进行时, 不要改变环值。

| | | |
|------|-----------------------|-----------|
| TB | 环值, 低 4 位数字(BCD) | } 环值设定 |
| TB+1 | 环值, 高 4 位数字(BCD) | |
| TB+2 | 目标值数(BCD) | } 一个目标值设定 |
| TB+3 | 目标值 # 1, 低 4 位数字(BCD) | |
| TB+4 | 目标值 # 1, 高 4 位数字(BCD) | |
| TB+5 | 位型式 # 1 | |
| ⋮ | ⋮ | |

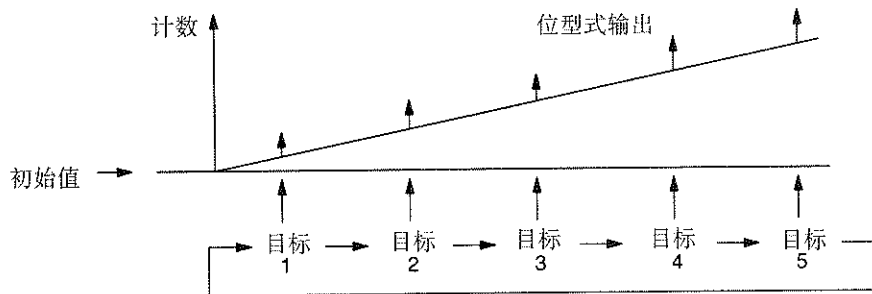
目标值 1~48 和位型式 1~48 存储在比较表中。位型式的位 0~7 存储内部位型式。而位型式的位 8~11 则存储外部位型式,对四个高速计数器,这些位进行逻辑或的计算,并把结果输出至外部输出 1~4。

下图显示高速计数器 1~4 的位型式如何进行“或”,而产生外部输出的输出结果。



目标值比较操作

下图显示在比较表中连续设定 1~5 个目标值的目标值比较操作。

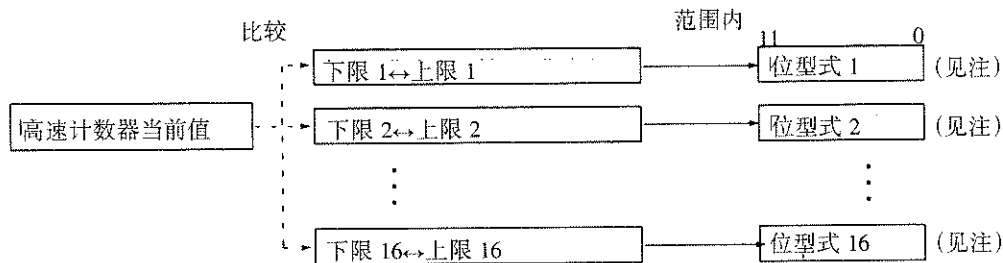


如上图所示,按登录在目标值比较表中的次序,把当前的计数与每个目标值相比较。当计数与当前目标值相同时,把登录的位型式输出到分配 IR 字中,并且开始与下一个目标值比较。当表中所有目标值都已经相同并且它们的位型式已经输出,目标值返回到表中第一个目标值并且重复此操作。

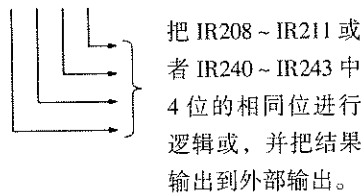
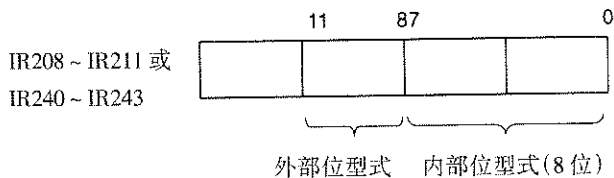
范围比较

一个范围比较表包含 8 个范围,它们由 8 个数字的下限和 8 个数字的上限以及位型式所定义。在当前值落在给定范围内时,登录的位型式输出到分配的 IR 字中。

高速计数器板不产生中断；登录的位型式反映在分配的 IR 字和外部输出上。

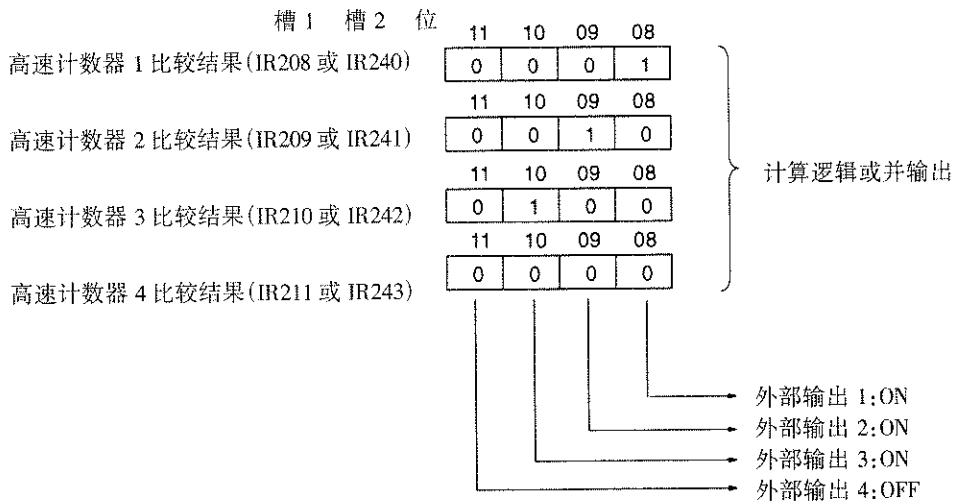


注：位型式 1~16 如下图示：



为范围比较表中的每个范围(1~16)登录一个下限,上限,和位型式。位型式的位 0~7 存储内部位型式,位 8~11 则存储外部位型式,四个高速计数器的这些位进行逻辑“或”计算,并把结果输出到外部输出 1~4。

下图显示如何对高速计数器 1~4 的位型式进行“或”而产生外部输出的输出结果。



下图显示使用高速计数器 1—4 设置为线性计数时的范围比较表的结构。

| | | |
|-------|------------------------|------------|
| TB | 下限 # 1, 低 4 位数字 (BCD) | } 第一个范围设定 |
| TB+1 | 下限 # 1, 高 4 位数字 (BCD) | |
| TB+2 | 上限 # 1, 低 4 位数字 (BCD) | |
| TB+3 | 上限 # 1, 高 4 位数字 (BCD) | |
| TB+4 | 位型式 # 1 | } 第十六个范围设定 |
| ⋮ | ⋮ | |
| TB+75 | 下限 # 16, 低 4 位数字 (BCD) | |
| TB+76 | 下限 # 16, 高 4 位数字 (BCD) | |
| TB+77 | 上限 # 16, 低 4 位数字 (BCD) | |
| TB+78 | 上限 # 16, 高 4 位数字 (BCD) | |
| TB+79 | 位型式 # 16 | |

下图显示使用高速计数器 1~4 设置为环形计数时的范围比较表的结构。环值指定环中的点数和最大计数值 (环值 = 最大计数值 + 1)。当比较正在进行时, 不要改变环值。

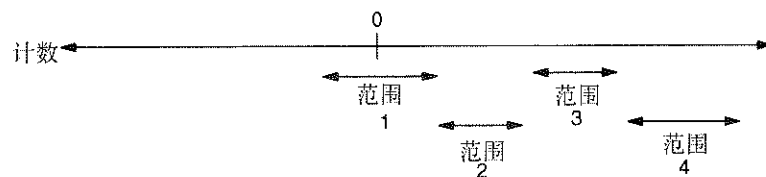
| | | |
|-------|------------------------|------------|
| TB | 环值, 低 4 位数字 (BCD) | } 环值设定 |
| TB+1 | 环值, 高 4 位数字 (BCD) | |
| TB+2 | 下限 # 1, 低 4 位数字 (BCD) | } 第一个范围设定 |
| TB+3 | 下限 # 1, 高 4 位数字 (BCD) | |
| TB+4 | 上限 # 1, 低 4 位数字 (BCD) | |
| TB+5 | 上限 # 1, 高 4 位数字 (BCD) | |
| TB+6 | 位型式 # 1 | } 第十六个范围设定 |
| ⋮ | ⋮ | |
| TB+77 | 下限 # 16, 低 4 位数字 (BCD) | |
| TB+78 | 下限 # 16, 高 4 位数字 (BCD) | |
| TB+79 | 上限 # 16, 低 4 位数字 (BCD) | |
| TB+80 | 上限 # 16, 高 4 位数字 (BCD) | |
| TB+81 | 位型式 # 16 | |

下表显示能设置为高速计数器 1~4 目标值、下限值和上限值的可能值。最高位数的十六进制 F 表示该值是负数 (负的 7 位数值)

| 数据格式 | 可能值 | |
|-------|-----------------------|-----------------------|
| | 线性计数 | 环形计数 |
| BCD | F838 8608 ~ 0838 8607 | 0000 0001 ~ 0838 8607 |
| 十六进制数 | F800 0000 ~ 07FF FFFF | 0000 0001 ~ 07FF FFFF |

范围比较操作

下图显示在比较表中连续设定的范围设定 1~4 的范围比较操作。



如上图示, 当前的计数同时与所有比较范围进行比较, 并输出每个范围的结果。

当把高速计数器板安装在槽 1 中时,位型式就被输出到 IR208 ~ IR211。当把高速计数器板安装在槽 2 中时,位型式就被输出到 IR240 ~ IR243。

| 计数器号 | 分配的IR字 | |
|--------|--------|-------|
| | 槽1中的板 | 槽2中的板 |
| 高速计数器1 | IR208 | IR240 |
| 高速计数器2 | IR209 | IR241 |
| 高速计数器3 | IR210 | IR242 |
| 高速计数器4 | IR211 | IR243 |

下表显示分配的 IR 字中的位功能。

| 位 | 功 能 |
|---------|----------------------------|
| 00 ~ 07 | 包括内部位型式 |
| 08 ~ 11 | 包括外部位型式 |
| 12 | 计数器运行标志(0:停止;1:运行) |
| 13 | 比较标志(0:停止;1:运行) |
| 14 | 当前值上溢/下溢标志(0:正常;1:出现上溢/下溢) |
| 15 | SV错误标志(0:正常;1:出现SV错误)。 |

- 注
1. 当使用高速计数器 1~4 的目标比较时,设定目标值以便使位型式以 0.2ms 或更长的间隔输出。
 2. 当使用高速计数器 1~4 的范围比较时,要设定适当极限以使计数器的当前值在上、下限之间至少有 0.5ms 间隔。(上限—下限)0.0005×输入频率)
 3. 当使用高速计数器 1~4 的目标比较时,通过上升还是下降达到目标值无关紧要。对高速计数器板的目标值比较也是如此,但是它却不同于脉冲 I/O 板中环形模式下的高速计数器 1 和 2 或者绝对编码器接口板中的高速计数器 1 和 2。

当 CQM1H 程序开始运行时,高速计数器 1~4 从 0 开始计数,但是将不输出位型式直到开始比较。用 INT(61)停止比较。

用 CTBL(63)登录的比较表有效,直到 CQM1H 程序结束运行或登录一个不同的比较表。需要时,可以通过执行 CTBL(63)的微分变化形式来减少循环时间。

标志

ER:指定口和功能不相兼容。

CTBL(63)指令调用的子程序中有另一个用不同比较方法的 CTBL(63)指令。

比较期间,有一个用不同比较方法的 CTBL(63)指令执行。

间接寻址的 EM/DM 字不存在。

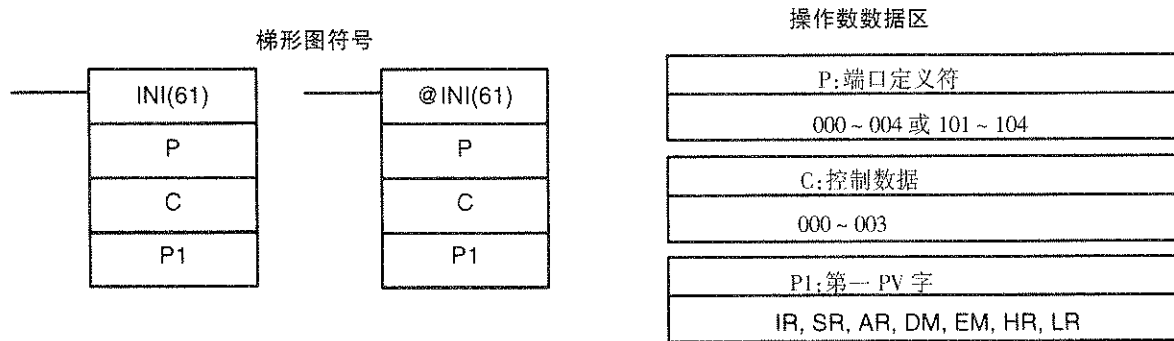
(*EM/*DM 字的内容不是 BCD,或者已超过 EM/DM 区边界。)

比较表超出了数据区边界,或者在比较表的设置中出现错误。

当主程序中执行脉冲 I/O 或高速计数器指令时,中断子程序中执行了 CTBL(63)。

当执行条件第一次满足时,仅执行一次子程序或位型式输出。AR 状态每个循环仅刷新一次。如果表中超过一项的条件同时满足,表中的第一项具有优先。

5—16—8 模式控制—INI(61)



限制

除 C 为 002 外, P1 必须为 000。
P1 和 P1 + 1 必须在同一数据区内。
DM6143 ~ DM6655 不能用于 P1。

说明

INI(61)可用于下表所列的功能。

| 单元/板 | 功能 |
|-----------|---------------------------|
| CPU单元 | 高速计数器 (IR00004 ~ IR00006) |
| 晶体管输出单元 | 脉冲输出 |
| 脉冲I/O板 | 高速计数器1和2 脉冲输出1和2 |
| 绝对值编码器接口板 | 绝对值高速计数器1和2 |
| 高速计数器板 | 高速计数器1 ~ 4 |

当执行条件为 OFF 时, INI(61)不执行。当执行条件为 ON 时, INI(61)用来控制高速计数器的操作并停止脉冲输出。

端口指定符(P)指定要控制的高速计数器或脉冲输出。

| 单元 | 功能 | 端口定义符(P) |
|---------|-------|----------|
| CPU单元 | 高速计数器 | 000 |
| 晶体管输出单元 | 脉冲输出 | 000 |

| 内板 | 功能 | 端口定义符(P) | |
|-----------|--------------|----------|-----|
| | | 槽1 | 槽2 |
| 脉冲I/O板 | 高速计数器1或脉冲输出1 | --- | 001 |
| | 高速计数器2或脉冲输出2 | --- | 002 |
| 绝对值编码器接口板 | 绝对值高速计数器1 | --- | 001 |
| | 绝对值高速计数器2 | --- | 002 |
| 高速计数器板 | 高速计数器1 | 101 | 001 |
| | 高速计数器2 | 102 | 002 |
| | 高速计数器3 | 103 | 003 |
| | 高速计数器4 | 104 | 004 |

INI(61)的功能由控制数据 C 所决定。(当改变当前值时,P1 和 P1 + 1 包括新的高速计数器当前值)。

| C | P1 | INI(61)功能 |
|-----|------------|---------------|
| 000 | 000 | 开始CTBL(63)表比较 |
| 001 | 000 | 停止CTBL(63)表比较 |
| 002 | 新的高速计数器当前值 | 改变高速计数器当前值 |
| 003 | 000 | 停止脉冲输出 |

下表显示对每个功能 C 可以使用的值。

| 单元/板 | 功能 | C值 | | | |
|-----------|-------------|-----|-----|-----|-----|
| | | 000 | 001 | 002 | 003 |
| CPU单元 | 高速计数器0 | 是 | 是 | 是 | --- |
| 晶体管输出单元 | 脉冲输出 | --- | --- | --- | 是 |
| 脉冲I/O板 | 高速计数器1和2 | 是 | 是 | 是 | --- |
| | 脉冲输出1和2 | --- | --- | --- | 是 |
| 绝对值编码器接口板 | 绝对值高速计数器1和2 | 是 | 是 | --- | --- |
| 高速计数器板 | 高速计数器1~4 | 是 | 是 | 是 | --- |

CTBL(63)表比较

如果 C 为 000 或 001,INI(61)停止或启动用 CTBL(63)登录的比较表与高速计数器当前值的比较。有关比较表的详情参考 1-4-5 高速计数器0 中断。

PV 改变

如果 C 为 002,那么 INI(61)把高速计数器的当前值变成 P1 和 P1 + 1 中的 8 位数值。左边的 4 位数字存于 P1 + 1 中,右四位数字存于 P1 中。当前值最高位的十六进制值 F 表示当前值为负。

CPU 单元:内置的高速计数器 0

下表显示高速计数器 0 当前值可能的 8 位数字 BCD 值。

| 模式 | 可能值 |
|-------|-----------------------|
| 相位差模式 | F003 2768 ~ 0003 2767 |
| 增量模式 | 0000 0000 ~ 0006 5535 |

脉冲 I/O 板:高速计数器 1 和 2

下表显示脉冲 I/O 板上的高速计数器 1 和 2 当前值可能的 8 个数字 BCD 值。

| 数字范围 | 可能值 |
|------|-----------------------|
| 线性计数 | F838 8608 ~ 0838 8607 |
| 环形计数 | 0000 0000 ~ 0006 4999 |

绝对编码器接口板:高速计数器 1 和 2

不能改变绝对值高速计数器 1 和 2 的当前值。

高速计数器板:高速计数器 1~4

下表显示高速计数器板上高速计数器 1~4 当前值可能的 8 个数字值 (BCD 或十六进制)。

| 数字范围 | 可能值 | |
|------|-----------------------|-----------------------|
| | BCD格式 | 十六进制格式 |
| 线性计数 | F838 8608 ~ 0838 8607 | F800 0000 ~ 07FF FFFF |
| 环形计数 | 0000 0000 ~ 0838 8607 | 0000 0000 ~ 07FF FFFF |

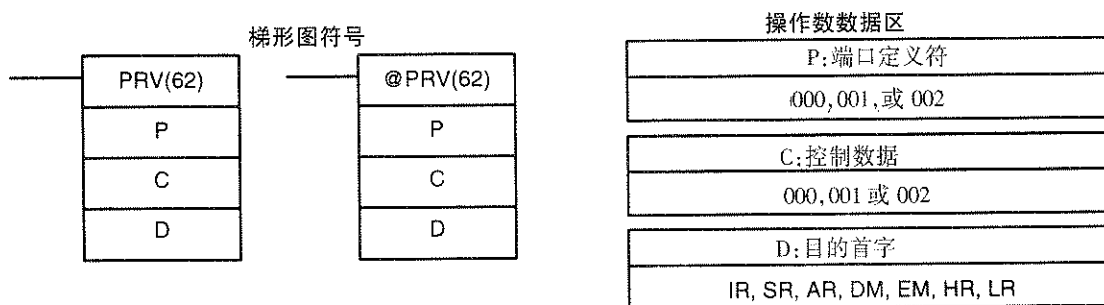
停止脉冲输出

如果 C 为 003,那么 INI(61)停止脉冲输出。有关停止脉冲 I/O 板脉冲输出 1 和 2 的详情参考 1-5 脉冲输出功能。

标志

ER: 指定的端口与功能不相兼容。
 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 或已超出 EM/DM 区边界。)
 P1 + 1 超过数据区边界。(C = 002)
 操作数设置错误。
 当主程序中执行脉冲 I/O 或高速计数器指令时, 中断子程序中执行 INI (61)指令。

5—16—9 读高速计数器当前值—PRV(62)



限制

D 和 D + 1 必须在同一数据区
 DM6143 ~ DM6655 不能用于 D。

说明

PRV(62)可以用于下表所列的功能。

| 单元/板 | 功能 |
|-----------|---------------------------|
| CPU单元 | 高速计数器0(IR00004 ~ IR00006) |
| 脉冲I/O板 | 高速计数器1和2 脉冲输出1和2 |
| 绝对值编码器接口板 | 绝对值高速计数器1和2 |
| 高速计数器板 | 高速计数器1~4 |

当执行条件为 OFF 时, PRV(62)不执行。当执行条件为 ON 时, PRV(62)读根据 P 和 C 指定的数据, 并把它写到 D 或 D + 1 中。

端口定义符(P)指定要控制的高速计数器或脉冲输出。

| 单元 | 功能 | 端口定义符(P) |
|---------|--------|----------|
| CPU单元 | 高速计数器0 | 000 |
| 晶体管输出单元 | 脉冲输出 | 000 |

| 内板 | 功能 | 端口定义符(P) | |
|-----------|--------------|----------|-----|
| | | 槽1 | 槽2 |
| 脉冲I/O板 | 高速计数器1或脉冲输出1 | --- | 001 |
| | 高速计数器2或脉冲输出2 | --- | 002 |
| 绝对值编码器接口板 | 绝对值高速计数器1 | --- | 001 |
| | 绝对值高速计数器2 | --- | 002 |
| 高速计数器板 | 高速计数器1 | 101 | 001 |
| | 高速计数器2 | 102 | 002 |
| | 高速计数器3 | 103 | 003 |
| | 高速计数器4 | 104 | 004 |

控制数据 C 决定要访问的数据类型。

| C | 数据 | 目的字 |
|-----|---------------|---------|
| 000 | 高速计数器当前值 | D和D + 1 |
| 001 | 高速计数器或脉冲输出的状态 | D |
| 002 | 范围比较结果 | D |

下表显示对每个功能 C 可以使用的值

| 单元/板 | 功能 | C值 | | |
|-----------|-------------|-----|-----|-----|
| | | 000 | 001 | 002 |
| CPU单元 | 高速计数器0 | 是 | --- | 是 |
| 晶体管输出单元 | 脉冲输出 | --- | --- | --- |
| 脉冲I/O板 | 高速计数器1和2 | 是 | 是 | 是 |
| | 脉冲输出1和2 | --- | 是 | --- |
| 绝对值编码器接口板 | 绝对值高速计数器1和2 | 是 | 是 | 是 |
| 高速计数器板 | 高速计数器1~4 | 是 | 是 | --- |

**高速计数器当前值
(C = 000)**

如果 C 为 000, 那么 PRV(62) 读指定的高速计数器的当前值并将八位数字写入 D 和 D + 1 中。左边 4 位数存贮于 D + 1 中, 右 4 位数存贮于 D 中。当前值最高位的十六进制值 F 表示当前值为负。

PRV(62) 读的高速计数器当前值的信息与该目的分配的存贮于 IR 字中的相同 (对高速计数器 0, 为 IR230 和 IR231, 对于高速计数器 1~4 则为 IR200~IR207 或者 IR232~IR239), 但是, 分配的 IR 字在每个循环中只刷新一次, 而 PRV(62) 读最新的数值。

CPU 单元: 内置高速计数器 0

下表显示高速计数器 0 的当前值可能的 8 位数 BCD 值。

| 模式 | 可能值 |
|-------|-----------------------|
| 相位差模式 | F003 2768 ~ 0003 2767 |
| 递增模式 | 0000 0000 ~ 0006 5535 |

脉冲 I/O 板: 高速计数器 1 和 2

下表显示脉冲 I/O 板上高速计数器 1 和 2 当前值可能的 8 个数字 BCD 值。

| 数字范围 | 可能值 |
|------|-----------------------|
| 线性计数 | F838 8608 ~ 0838 8607 |
| 环形计数 | 0000 0000 ~ 0006 4999 |

绝对值编码器接口板: 高速计数器 1 和 2

下表显示绝对值高速计数器 1 和 2 当前值的可能值。

| 模式 | 可能值 |
|--------|-----------------------|
| BCD模式 | 0000 0000 ~ 0000 4095 |
| 360°模式 | 0000 0000 ~ 0000 0359 |

高速计数器板: 高速计数器 1~4

下表显示高速计数器板上高速计数器 1~4 当前值的可能的 8 位数值 (BCD 或十六进制)。

| 数字范围 | 可能值 | |
|------|-----------------------|-----------------------|
| | BCD格式 | 十六进制格式 |
| 线性计数 | F838 8608 ~ 0838 8607 | F800 0000 ~ 07FF FFFF |
| 环形计数 | 0000 0000 ~ 0838 8607 | 0000 0000 ~ 07FF FFFF |

高速计数器或脉冲输出状态(C = 001)

如果 C 为 001, 那么 PRV(62) 读指定的高速计数器或脉冲输出的操作状态, 并将数据写入 D。

PRV(62)读的信息与该目的分配的存贮于 AR 和 IR 字中的相同(对于脉冲 I/O 板或绝对值编码器接口板, 为 AR05 和 AR06, 对高速计数器板则为 IR208 ~ IR211 或 IR240 ~ IR243), 但是, 这些分配的 AR 和 IR 字在每个循环中只刷新一次, 而 PRV(62)读最新的值。

脉冲 I/O 板

下表显示用于高速计数器 1 和 2, 或脉冲 I/O 板上的端口 1 和 2 输出脉冲的 D 中的位的功能。表中未列出的位不使用, 并且总为 0。

| 位 | 功能 |
|----|----------------------------|
| 00 | 高速计数器比较状态(0:已停止;1:正在比较) |
| 01 | 高速计数器下溢/上溢(0:正常;1:发生下溢/上溢) |
| 04 | 脉冲频率减速(0:未指定;1:指定) |
| 05 | 指定脉冲的总数(0:未指定;1:指定) |
| 06 | 脉冲输出完毕(0:未完毕;1:完毕) |
| 07 | 脉冲输出状态(0:已停止;1:正在输出) |

绝对值编码器接口板

对于绝对值高速计数器 1 和 2, D 的位 00 表示比较状态(0:已停止;1:正在比较)。D 的其它位(01 ~ 15)未使用并且总为 0。

高速计数器板

下表显示对于高速计数器板上 的高速计数器 1 ~ 4 的 D 中位的功能。

| 位 | 功能 |
|---------|----------------------------|
| 00 ~ 07 | 包括内部位型式 |
| 08 ~ 11 | 包括外部位型式 |
| 12 | 计数器运行标志(0:已停止;1:正在运行) |
| 13 | 比较标志(0:已停止;1:正在运行) |
| 14 | 当前值上溢/下溢标志(0:正常;1:发生上溢/下溢) |
| 15 | SV错误标志(0:正常;1:发生SV错误) |

范围比较结果(C = 002)

如果 C 为 002, PRV(62) 读内置高速计数器 0, 脉冲 I/O 板高速计数器 1 和 2 或者绝对值编码器接口板绝对高速计数器 1 和 2 的范围比较结果的结果。

D 的 00 ~ 07 位包含范围 1 ~ 8 的比较结果标志(0:不在范围内;1:在范围内)。PRV(62) 读的信息与该目的分配的存贮于 AR 字中的相同(对脉冲 I/O 板或绝对值编码器接口板为 AR05 和 AR06, 对内置高速计数器 0 则为 AR11), 但是分配字在每个循环中只刷新一次, 而 PRV(62)读最新的数值。

标志

ER:指定的端口与功能不兼容。

间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或者已超出 EM/DM 区边界)。

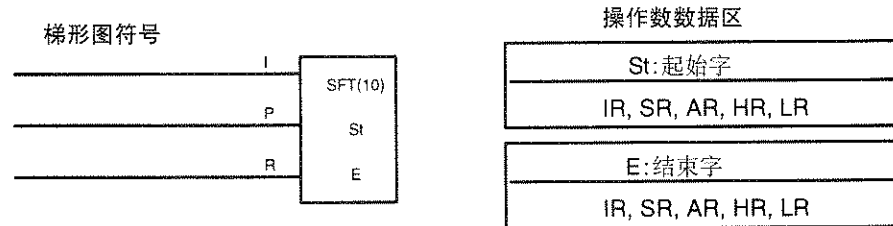
D + 1 超出数据区边界(C = 000)

操作设定时有错误。

当在主程序中执行脉冲 I/O 或高速计数器的指令时, 中断子程序执行 PRV(62)。

5—17 移位指令

5—17—1 移位寄存器——SFT(10)



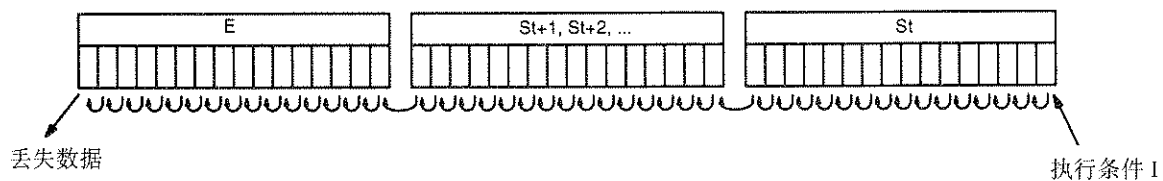
限制

E 必须大于或等于 St, St 和 E 必须在同一数据区。

如果用于移位寄存器的某一字的位地址同时用在单独的位状态控制指令 (如, OUT, KEEP(11)) 中, 当用编程器或其它编程设备进行程序语法检查时, 将会产生一个错误 (“COIL/OUT DUPL”: 线圈/输出重复)。但是, 仍将按照编写的程序执行。见例 2: 移位寄存器的控制位编程举例。

说明

SFT(10) 由三个执行条件 I, P 和 R 控制。如果执行 SFT(10), 并且 1) 执行条件 P 为 ON 且上次为 OFF, 2) R 为 OFF, 那么, 执行条件 I 移到 St 和 E 之间的移位寄存器的最右面位。即如果 I 为 ON, 1 移进寄存器; 如果 I 为 OFF, 移进 0。当 I 移进寄存器时, 寄存器中以前的所有位向左移, 寄存器的最左位丢失。



执行条件 P 的功能就象一个微分指令, 即只有当 P 为 ON 且上次执行 SFT(10) 时 P 为 OFF 时, I 才移到寄存器中。如果 P 的执行条件没有改变, 或从 ON 变为 OFF, 移位寄存器将保持不变。

St 指定移位寄存器的最右边的字; E 指定最左边的字。移位寄存器包括这两个字和这两个字之间的所有字。St 和 E 可以指定为同一个字, 这时则产生一个 16 位 (即 1 字) 移位寄存器。

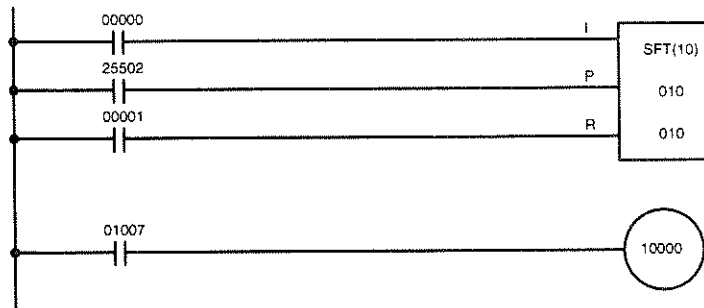
当 R 执行条件为 ON 时, 移位寄存器的所有位将置为 OFF (即置为 0), 移位寄存器将不动作, 直到 R 又变为 OFF。

标志

SFT(10) 不影响标志。

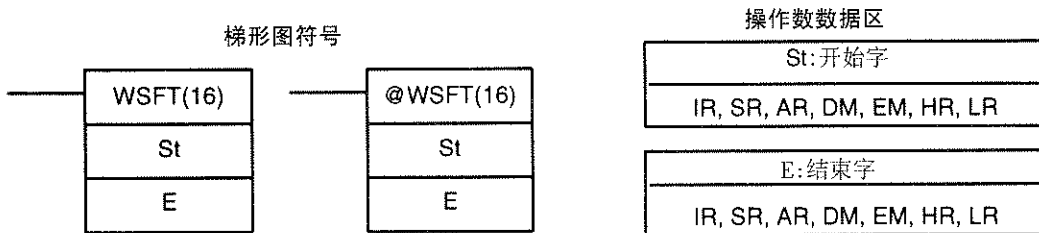
例

下面的例子使用 1 秒时钟脉冲位 (25502), 因此由 00000 所产生的执行条件每秒移进 IR010 一次。当 1 移进 01007 时, 输出 10000 置为 ON。



| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | LD | 25502 |
| 00002 | LD | 00001 |
| 00003 | SFT(10) | 010 |
| | | 010 |
| 00004 | LD | 01007 |
| 00005 | OUT | 10000 |

5-17-2 字移位—WSFT(16)

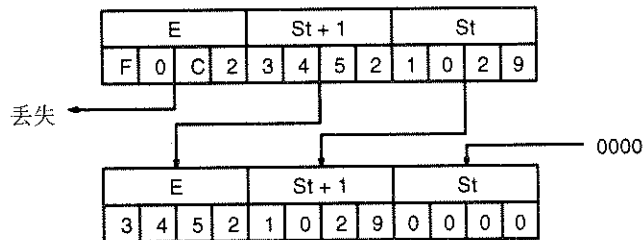


限制

St 和 E 必须在同一数据区中, E 必须大于或等于 St。
St 或 E 不能使用 DM6144 ~ DM6655

说明

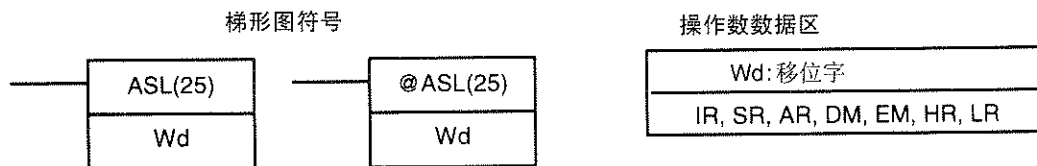
当执行条件为 OFF 时, WSFT(16) 不执行。当执行条件为 ON 时, WSFT(16) 将 St 和 E 之间的数据以字为单位移动。零移进 St, E 的内容丢失。



标志

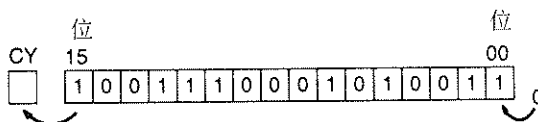
ER: St 和 E 的字在不同的区域, 或 St 大于 E。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD 形式, 或者超过 EM/DM 区边界)。

5-17-3 算术左移——ASL(25)



限制 DM6144 ~ DM6655 不能用于 Wd。

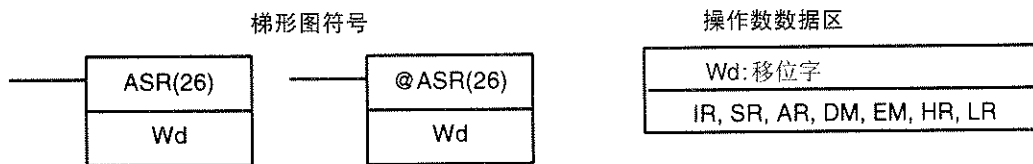
说明 当执行条件为 OFF 时, ASL(25) 不执行。当执行条件为 ON 时, ASL(25) 将 0 移进 Wd 的 00 位, Wd 中的位依次左移一位, 第 15 位的状态移进 CY。



注意 如果使用 ASL(25) 的非微分形式, 每个循环都要移进位 00 一个 0。使用微分形式 (@ASL(25)), 或者将 ASL(25) 同 DIFU(13) 或 DIFD(14) 组合使用, 使得只移位一次。

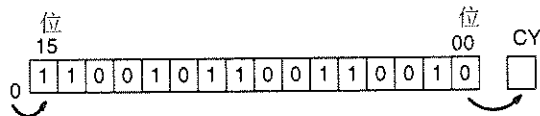
标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/*DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)
 CY: 接收位 15 的状态。
 EQ: 当 Wd 的内容为零时为 ON; 否则为 OFF。

5-17-4 算术右移——ASR(26)



限制 DM6144 ~ DM6655 不能用于 Wd。

说明 当执行条件为 OFF 时, ASR(26) 不执行。当执行条件为 ON 时, ASR(26) 将 0 移进 Wd 的 15 位, Wd 中的位依次右移一位, 00 位的状态移进 CY。



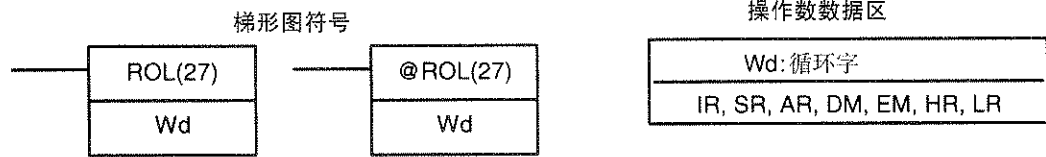
注意 如果使用 ASR(26) 的非微分形式, 每个循环都要移进位 15 一个 0。使用微分形式 (@ASR(26)), 或者将 ASR(26) 同 DIFU(13) 或 DIFD(14) 组合使用, 使得只移位一次。

标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/*DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)

CY:接收位 00 的数据。

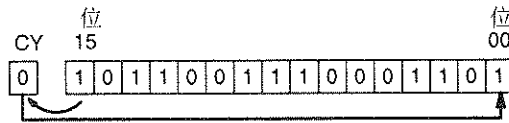
EQ:当 Wd 的内容为零时为 ON;反之为 OFF。

5-17-5 循环左移—ROL(27)



限制 DM6144 ~ DM6655 不能用于 Wd。

说明 当执行条件为 OFF 时,ROL(27)不执行。当执行条件为 ON 时,ROL(27)将 Wd 中所有位依次左移一次,CY 移进 Wd 的位 00,Wd 的 15 位移进 CY。



注意 在执行循环操作之前,使用 STC(41) 设置 CY 的状态,或者用 CLC(41) 清除 CY 的状态,以确保在执行 ROL(27)之前,CY 具有正确的状态。

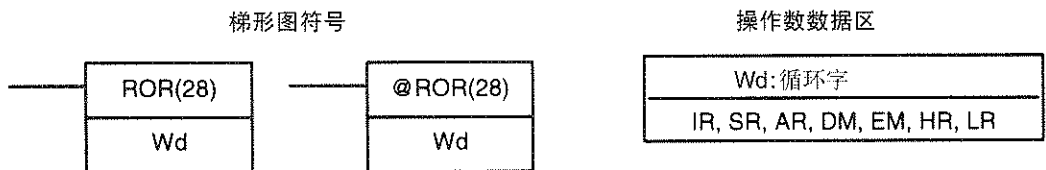
如果使用 ROL(27)的非微分形式,每个循环都要将 CY 移进位 00。使用微分形式(@ROL(27)),或者将 ROL(27)同 DIFU(13)或 DIFD(14)组合使用,使得只移位一次。

标志 ER:间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD 形式,或者超出 EM/DM 区边界。)

CY:接收位 15 的状态。

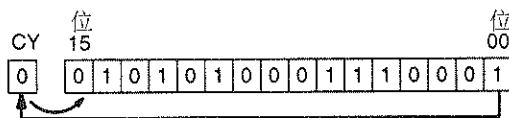
EQ:当 Wd 的内容为零时 ON;否则为 OFF。

5-17-6 循环右移—ROR(28)



限制 DM6144 ~ DM6655 不能用于 Wd。

说明 当执行条件为 OFF 时,ROR(28)不执行。当执行条件为 ON 时,ROR(28)将 Wd 中所有位依次右移一次,CY 移进 Wd 的位 15 并且 Wd 的位 00 移进 CY。

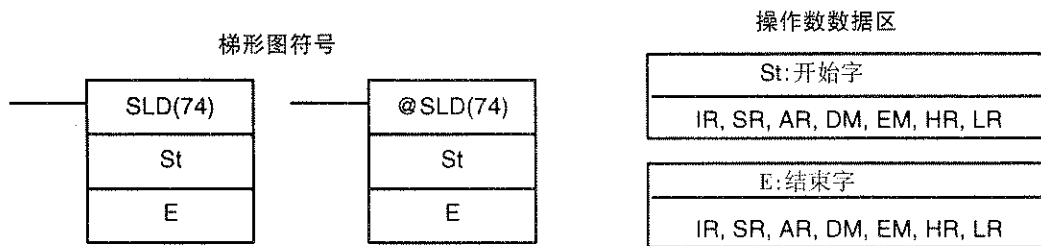


注意 在进行循环操作之前,使用 STC(41) 设置 CY 的状态,或者用 CLC(41) 清除 CY 的状态,以确保在执行 ROR(28)之前,CY 具有正确的状态。

如果使用 ROR(28)的非微分形式,每个循环 CY 都要移进位 15。使用微分形式(@ROR(28)),或者将 ROR(28)同 DIFU(13)或 DIFD(14)组合使用,使得只移位一次。

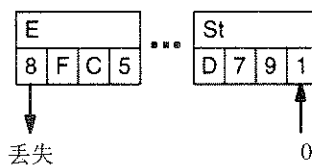
标志 ER:间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 形式,或者超出 EM/DM 区边界。)
 CY:接收位 00 的状态。
 EQ:当 Wd 的内容为零时为 ON;反之为 OFF。

5-17-7 一位数字左移—CLD(74)



限制 St 和 E 必须在同一数据区中,而且 E 必须大于或等于 St。
 DM6144 ~ DM6655 不能用于 St 或 E。

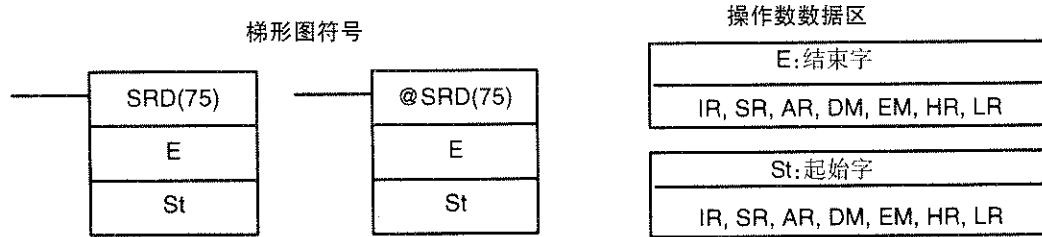
说明 当执行条件为 OFF 时,SLD(74) 不执行。当执行条件为 ON 时,SLD(74) 将 St 和 E 之间的 (包括本身) 数据左移一个数字 (4 位)。St 的最右边的数字写进 O,E 的最左边的数字的内容丢失。



注意 当跨越 50 个字的移位操作执行时发生电源故障,移位操作可能不能完成。如果使用 SLD(74) 的非微分形式,每个循环都要将一个 0 移进 St 的最低位数。使用微分形式(@SLD(74)),或者将 SLD(74)同 DIFU(13)或 DIFD(14)组合使用,使得只移位一次。

标志 ER:St 和 E 不在同一区域,或者 St 大于 E。
 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 形式,或者已超出 EM/DM 区边界。)

5—17—8 一位数字右移—SRD(75)

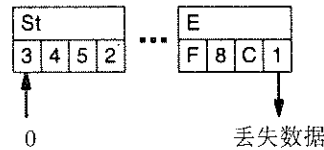


限制

St 和 E 必须在同一数据区中, E 必须小于或等于 St。
DM6144 ~ D6655 不能用于 St 或 E。

说明

当执行条件为 OFF 时, SRD(75) 不执行。当执行条件为 ON 时, SRD(75) 将 St 和 E 之间的数据(包括自身)右移一个数字(4 位)。St 的最左边的数字写进 O, E 的最右边数字的内容丢失。



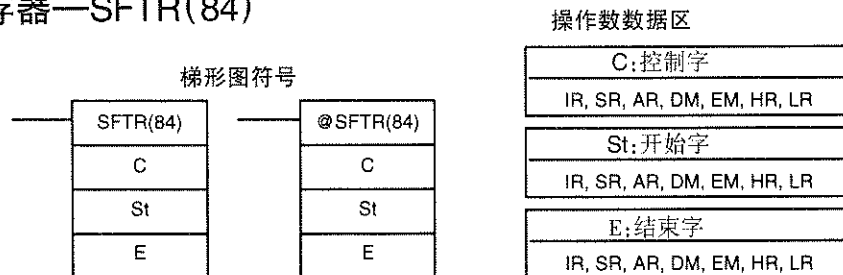
注意

当跨越 50 个字的移位操作执行时, 如果发生电源故障, 移位操作可能不能完成。
如果使用 SRD(75) 的非微分形式, 每个循环都要将一个 0 移进 St 的最高位数。
使用微分形式(@SRD(75)), 或者将 SRD(75) 同 DIFU(13) 或 DIFD(14) 组合使用, 使得移位只发生一次。

标志

ER: St 和 E 在不同的区域, 或者 St 小于 E。
间接寻址 EM/DM 字不存在。
(*EM/*DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)

5—17—9 可逆移位寄存器—SFTR(84)



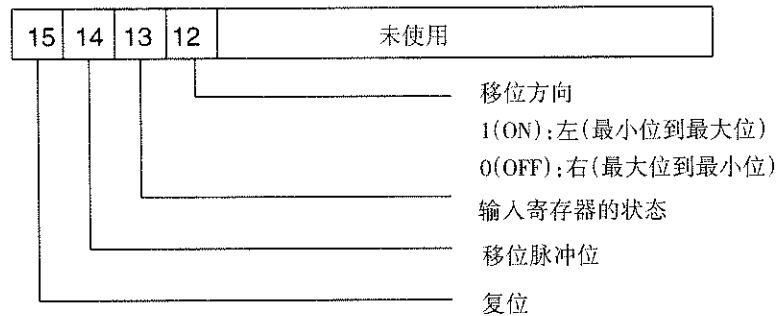
限制

St 和 E 必须在同一数据区中, St 必须小于或等于 E。
DM6144 ~ DM6655 不能用于 C、St 或 E。

说明

SFTR(84) 用来产生可以左移或右移数据的单字或多字移位寄存器。
要产生一个单字寄存器, St 和 E 指字为同一个字。控制字提供移动方向、将

要移进寄存器的状态、移位脉冲以及复位输入。控制字的分配如下：



当 SFTR(84) 的执行条件为 ON, 复位位为 OFF, 14 位为 ON, 移位寄存器将按位 12 指定的方向移动一位, 移出一位到 CY, 13 位的状态移到另一端。如果 SFTR(84) 的执行条件为 OFF, 或以位 14 为 OFF 条件执行, 移位寄存器保持不变。如果 SFTR(84) 的执行条件为 ON, 复位位(15 位)为 OFF, 整个移位寄存器和 CY 将置为零。

标志

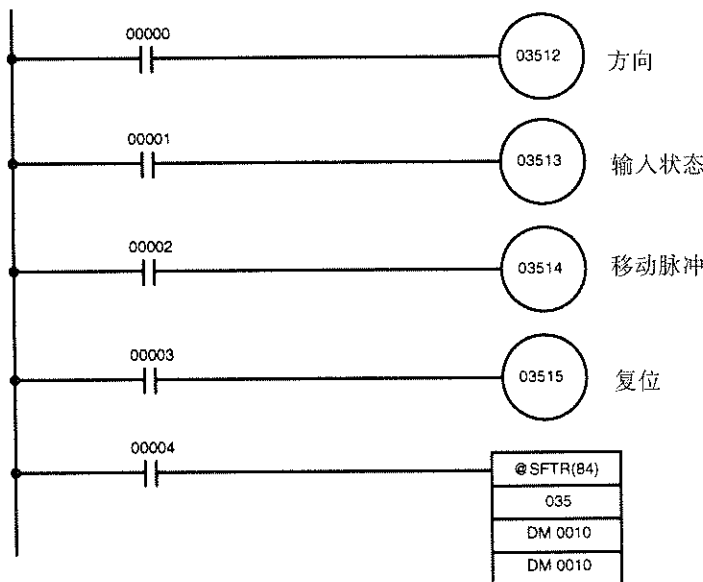
ER: St 和 E 在不同的区域或者 St 大于 E。
间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 形式, 或者超出 EM/DM 区边界)。

CY: 接收 St 的 00 位或 E 的 15 位的状态(取决于移位方向)。

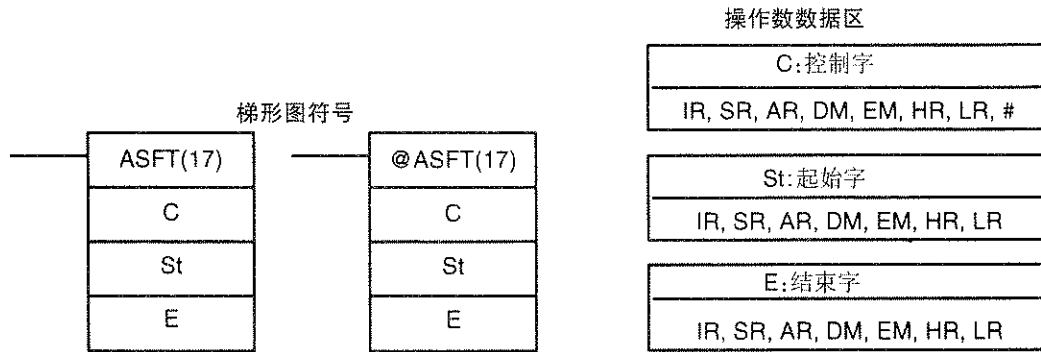
例子

在下面的例子中, IR00000, IR00001, IR00002 和 IR00003 用以控制 @SFTR(84) 使用中 C 的位, 移位寄存器在 DM0010 中, 它通过 IR00004 控制。



| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00000 |
| 00001 | OUT | 03512 |
| 00002 | LD | 00001 |
| 00003 | OUT | 03513 |
| 00004 | LD | 00002 |
| 00005 | OUT | 03514 |
| 00006 | LD | 00003 |
| 00007 | OUT | 03515 |
| 00008 | LD | 00004 |
| 00009 | @SFT(10) | |
| | | 035 |
| | | DM 0010 |
| | | DM 0010 |

5-17-10 异步移位寄存器—ASFT(17)



限制

St 和 E 必须在同一数据区中,并且 E 必须大于或等于 St。
DM6144 ~ DM6655 不能用于 St 或 E。

说明

当执行条件为 OFF, ASFT(17) 不作任何事, 程序转移到下一个指令。当执行条件为 ON, ASFT(17) 用于产生和控制 St 和 E 之间的可逆异步字移位寄存器。当寄存器中的下一个字为零时, 这个寄存器才移动字。如果寄存器中不包含零的字, 将不发生移动。并且, 只有寄存器中内容为零的字只移动一个字。当一个字的内容移到下一个字。原来的字的内容置为 0。实质上, 当寄存器移位时, 寄存器中的每个为零的字与下一个字交换(见下例)。

移动方向(即:“下一个字”是高一位字还是低一位字)由 C 决定。C 也用来复位寄存器, 寄存器的任何部分可以用 St 和 E 指定所要求的部分来复位。

控制字

C 的 00 位 ~ 12 位没有使用。13 位为移动方向: 13 位置 ON 为向下移动(向低地址字)。置 OFF 为向上移动(向高地址字)。14 位是移动允许位: 14 位置 ON 允许移位寄存器按照 13 位(指定的方向)操作, 置 OFF 不允许移位寄存器操作。15 位是复位位: 当 15 位置 ON 时执行 ASFT(17), 寄存器在 St 和 E 之间复位(置为零), 15 位置 OFF 为正常操作。

注

如果使用 ASFT(17) 的非微分形式, 当执行条件为 ON 时, 每个循环数据移位。使用微分形式可以防止这样。

标志

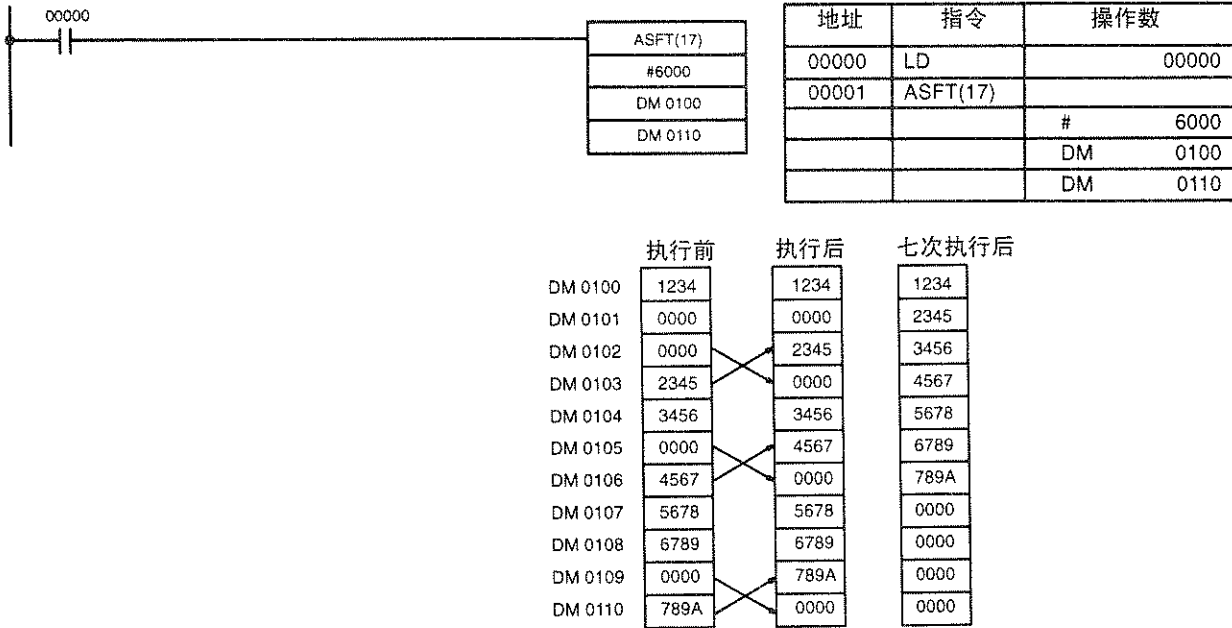
ER: St 和 E 在不同的区, 或 St 大于 E。

间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD 形式或者超出 EM/DM 区边界。)

例

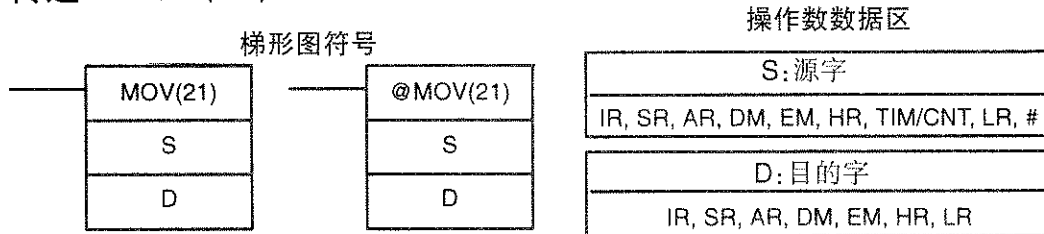
下面的例子示出了使用 SAFT(17)以 C = # 6000 移动在 DM0100 和 DM0110 之间的 11 字移位寄存器中的字。非零数据移向 St。(DM0110)。



注 如果 C = 4000, 零向上移动, 如果 C = 8000, 整个移位寄存器置为零。

5-18 数据传送指令

5-18-1 传送 - MOV(21)



限制
说明

D 不能使用 DM6144 ~ DM6655
当执行条件为 OFF 时, MOV(21) 不执行, 当执行条件为 ON, MOV(21) 将 S 的内容复制到 D。



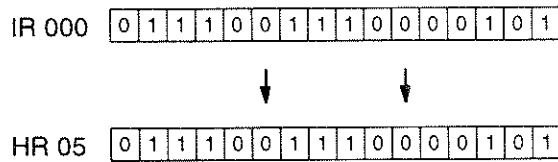
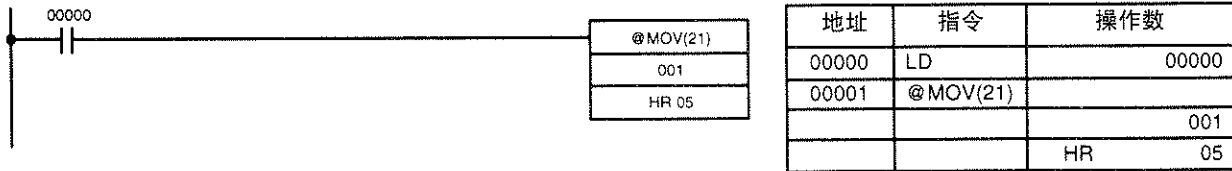
注意
标志

TIM/CNT 号不能指定作为 D 以改变定时器或计数器的当前值。
但是, 可以用 BSET(71) 容易地改变定时器或计数器的当前值。
ER: 间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD 形式或者超出 EM/DM 区边界。)

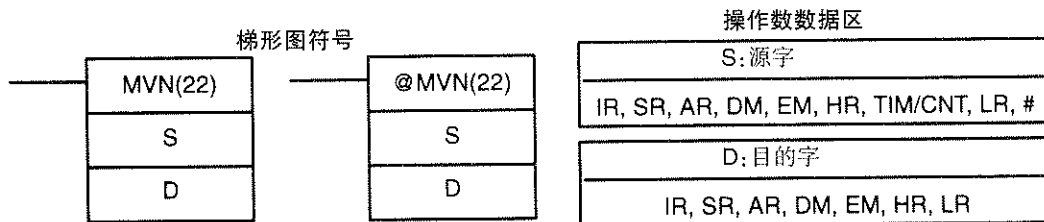
EQ: 当全零传送到 D 时, 为 ON。

例

下例示出了使用 @MOV(21), 当 IR00000 从 OFF 变为 ON 时, 将 IR001 的内容复制到 HR05。



5-18-2 传送非——MVN(22)



限制

D 不能使用 DM6144 ~ DM6655。

说明

当执行条件为 OFF 时, MVN(22) 不执行。当执行条件为 ON, MVN(22) 将 S (指定字或四位十六进制常数) 的内容取反传送到 D。即对应于 S 中的各 ON 位, D 中的相应位为 OFF, 对于 S 中的各 OFF 位, D 中的相应位为 ON。



注意

TIM/ CNT 不能指定作为 D, 以改变定时器或计数器的当前值。但是, 利用 BSET(71) 可以很容易地改变。

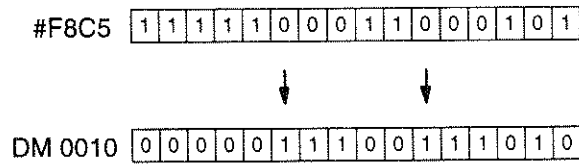
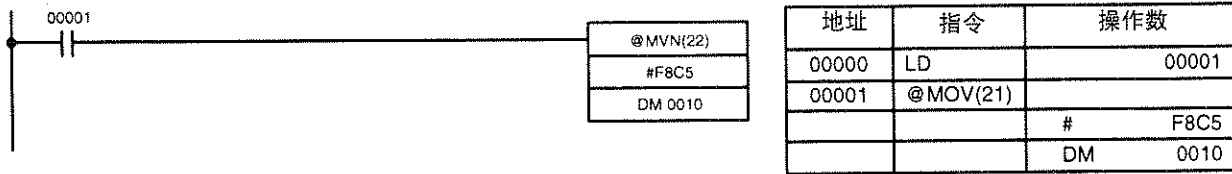
标志

ER: 间接寻址的 EM/DM 字不存在。

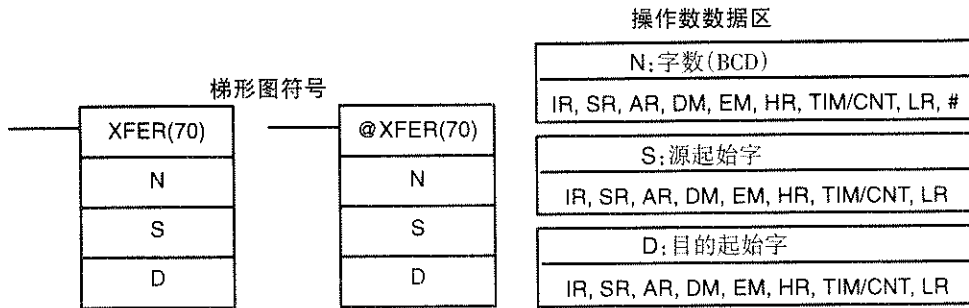
(* EM/* DM 的内容不是 BCD 形式, 或者超出 EM/DM 区边界。)

EQ: 当全零传送到 D 时为 ON。

例 下例示出了使用@MVN(22),当 IR00001 从 OFF 变为 ON 时,将 #F8C5 的补码复制到 DM0010 中。

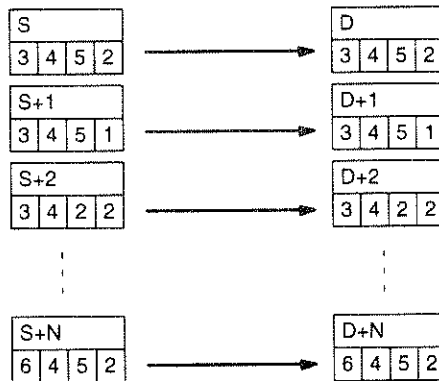


5-18-3 块传送—XFER(70)



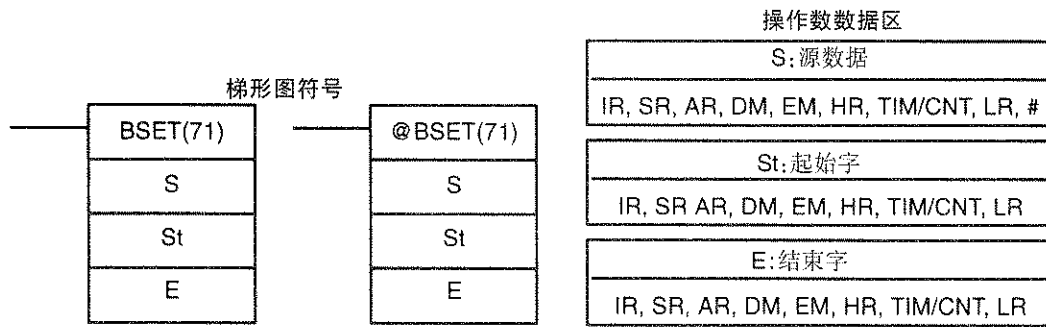
限制 S 和 S+N, D 和 D+N 必须在同一数据区中。
D 不能使用 DM6144 ~ DM6655。

说明 当执行条件为 OFF 时, XFER(70) 不执行, 当执行条件为 ON 时, XFER(70) 将 S, S+1, ..., S+N 的内容复制到 D, D+1, ..., D+N。



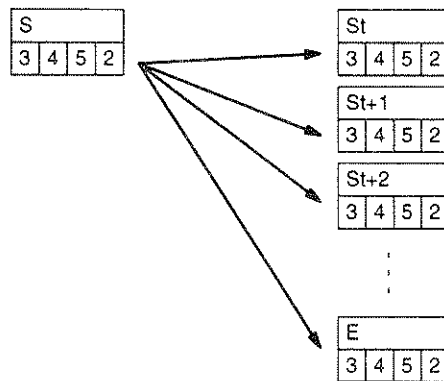
标志 ER:N 不是 BCD 码。
S 与 S+N 或者 D 与 D+N 不在同一数据区。
间接寻址的 EM/DM 字不存在。
(* EM/*DM 的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

5-18-4 块设置—BSET(71)



限制 St 必须小于或等于 E,且 St 和 E 必须在同一数据区。
St 或 E 不能使用 DM6144 ~ DM6655。

说明 当执行条件为 OFF 时,BSET(71)不执行。当执行条件为 ON 时,BSET(71)将 S 的内容复制到 St ~ E 的所有字。



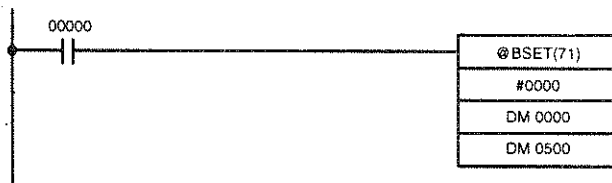
可以用 BSET(71) 修改定时器/计数器当前值(不能用 MOV(21) 或 MVN(22))。也可以用 BSET(71) 清除一段数据区,即 DM 区,以准备执行其它指令。它还可以通过传送全零清除字。

标志 ER:St 与 E 不在同一数据区,或者 St 大于 E。

间接寻址的 EM/DM 字不存在。

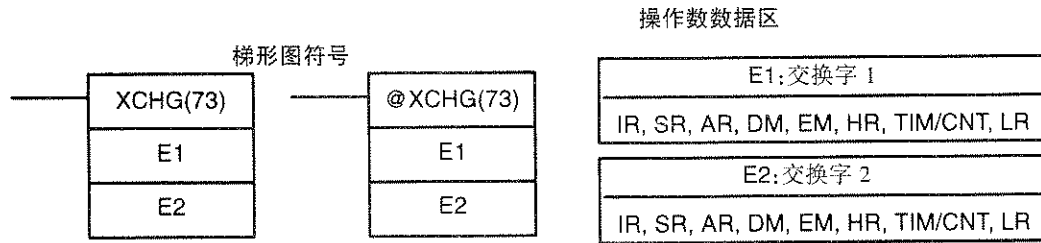
(* EM/* DM 字的内容不是 BCD 码,或者超出 EM/DM 区的边界)。

例 下例示出当 IR00000 为 ON 时,如何使用 BSET(71),将一常数 (# 0000)复制到一块 DM 区 (DM0000 ~ DM0500)。



| 地址 | 指令 | 操作数 |
|-------|-----------|---------|
| 00000 | LD | 00000 |
| 00001 | @BSET(71) | |
| | | # 0000 |
| | | DM 0000 |
| | | DM 0500 |

5—18—5 数据交换—XCHG(73)



限制
说明

E1 或 E2 不能使用 DM6144 ~ DM6655。
当执行条件为 OFF 时, XCHG(73) 不执行, 当执行条件为 ON 时, XCHG(73) 交换 E1 和 E2 的内容。

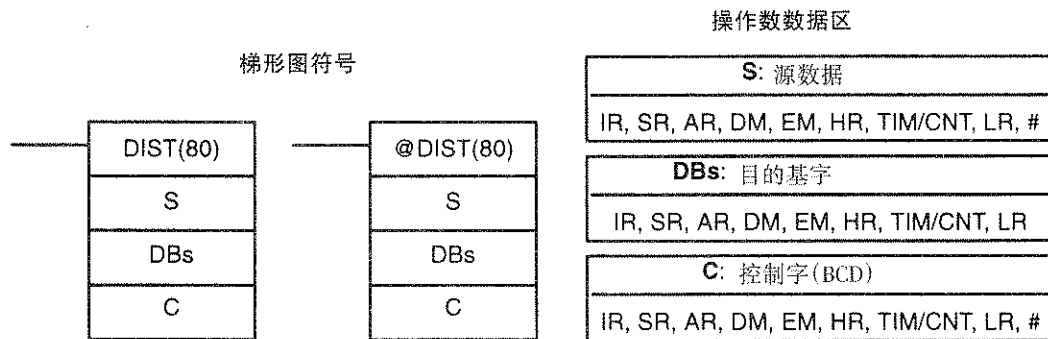


如果要交换长度大于 1 个字的块的内容, 用工作字作为中间缓冲区以保存一个块, 并使用三次 XFER(70)。

标志

ER: 间接寻址的 EM/DM 字不存在。
(*EM/*DM 字的内容不是 BCD 码, 或超出 EM/DM 区边界)。

5—17—6 单字分配—DIST(80)



限制

C 必须是 BCD 码。
DBs 或 C 不能使用 DM6144 ~ DM6655。

说明

单字分配

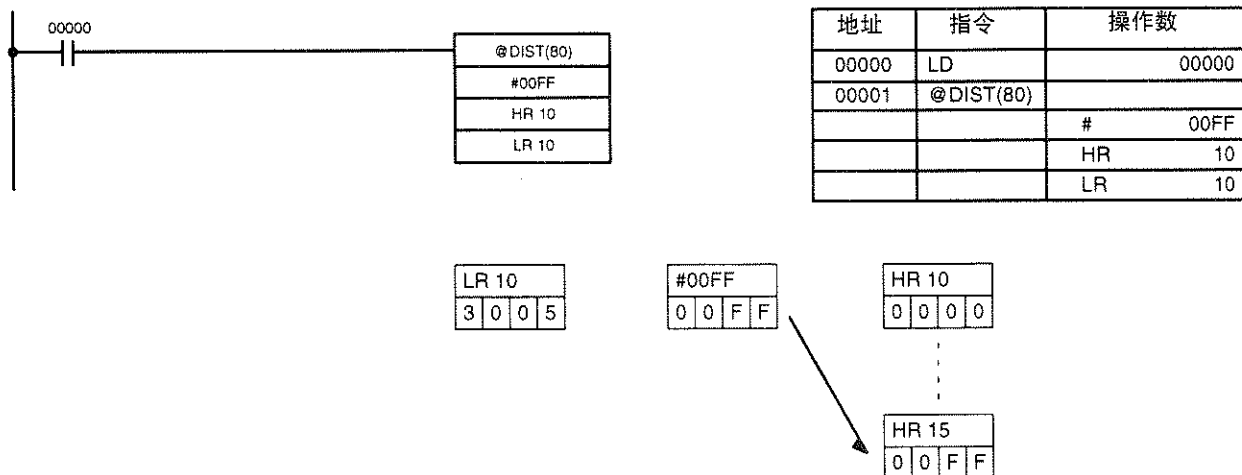
DIST(80) 根据控制字 C 的内容可用于单字分配或堆栈操作。
当 C 的 12 位 ~ 15 位 = 0 ~ 8, DIST(80) 可以用于单字分配操作。C 的整个内容指定一个偏移 Of。
当执行条件为 OFF 时, DIST(80) 不执行, 当执行条件为 ON 时, DIST(80) 将 S 的内容复制到 DBs + Of 中, 即 Of 加到 DBs 上决定目的字。

注 DBs 与 DBs + Of 必须在同一数据区, 且不能在 DM6144 与 DM6655 之间。

例

下例示出如何使用 DIST(80) 将 # 00FF 复制到 HR10 + Of。

LR10 的内容是 # 3005, 因此当 IR00000 为 ON 时, # 00FF 复制到 HR15(HR10 + 5)。



堆栈操作

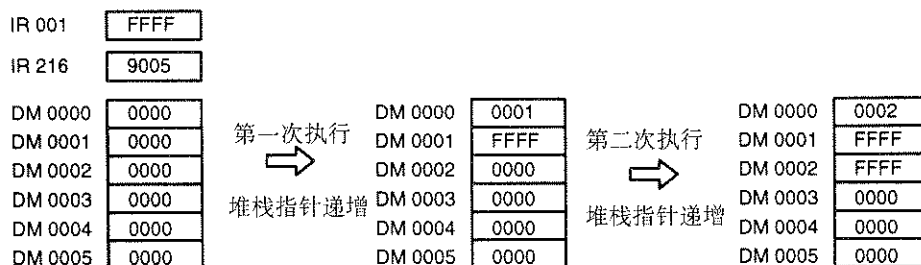
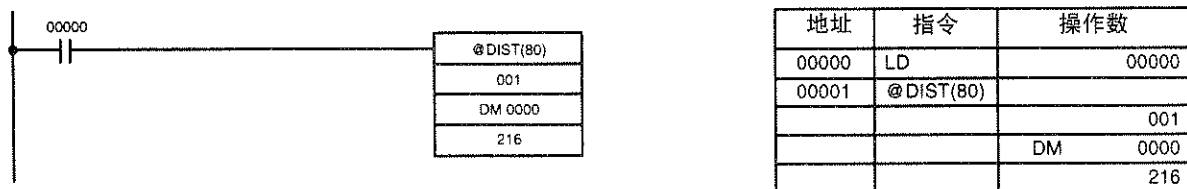
当 C 的 12 位 ~ 15 位 = 9, DIST(80) 可以用于堆栈操作, C 的其它 3 个数字指定堆栈(000 ~ 999)中的字数, DBs 的内容为堆栈指针。

当执行条件为 OFF 时, DIST(80) 不执行, 当执行条件为 ON 时, DIST(80) 将 S 的内容复制到 DBs + 1 + DBs 的内容。换句话说, 1 和 DBs 的内容加上 DBs 决定目的字。然后 DBs 的内容增 1。

- 注
1. DIST(80) 每个循环都执行, 除非使用微分形式 (@DIST(80)) 或者 DIST(80) 同 DIFU(13) 或 DIFD(14) 一起使用。
 2. 在使用 DIST(80) 进行堆栈操作之前, 务必初始化堆栈指针。

例

下例示出如何使用 DIST(80) 在 DM0001 ~ DM0005 之间产生一个堆栈, DM0000 作为堆栈指针。



标志

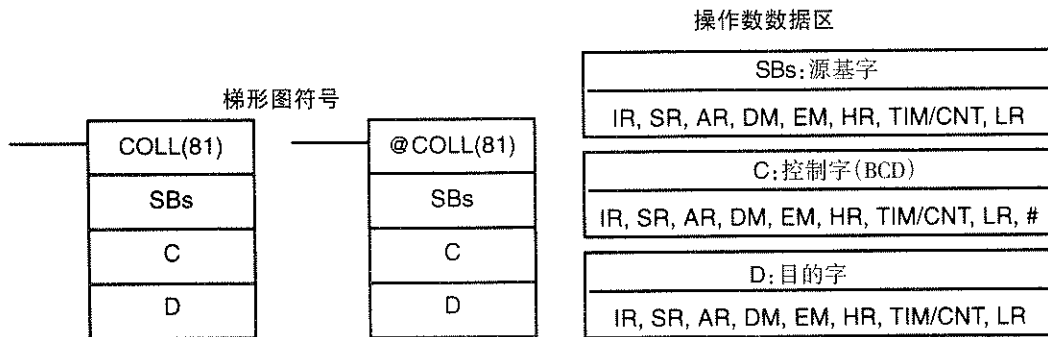
ER: 控制字中的偏移或堆栈深度不是 BCD 码。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界)。

堆栈操作中,堆栈指针 + 1 的值超出堆栈深度。
EQ: 当 S 的内容为零时为 ON, 否则为 OFF。

5—18—8 数据收集—COLL(81)



限制

C 必须为 BCD 码。
D 不能使用 DM6144 ~ DM6655。

说明

根据控制字 C 的内容, COLL(81) 可以用于数据采集, 可以是 FIFO 堆栈操作, 或者 LIFO 堆栈操作。

数据采集

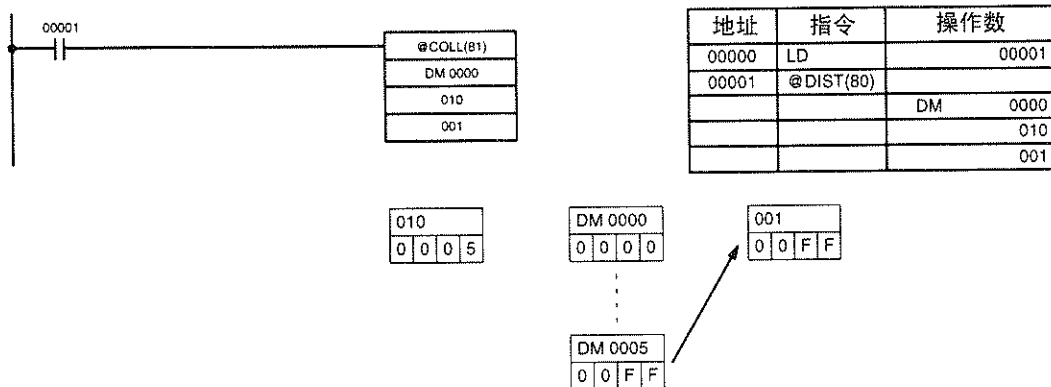
当 C 的 12 位 ~ 15 位为 0 ~ 7, COLL(81) 用于数据采集, C 的整个内容指定了偏移 Of。

当执行条件为 OFF 时, COLL(81) 不执行, 当执行条件为 ON 时, COLL(81) 将 SBs + Of 的内容复制到 D, 即, Of 加上 SBs 决定源字。

注 SBs 与 SBs + Of 必须在同一数据区中。

例

下例示出了如何使用 COLL(81) 将 DM0000 + Of 的内容复制到 IR001。010 的内容为 # 0005, 所以当 IR00001 为 ON 时, DM0005 (DM0000 + 5) 的内容复制到 IR001。



FIFO 堆栈操作

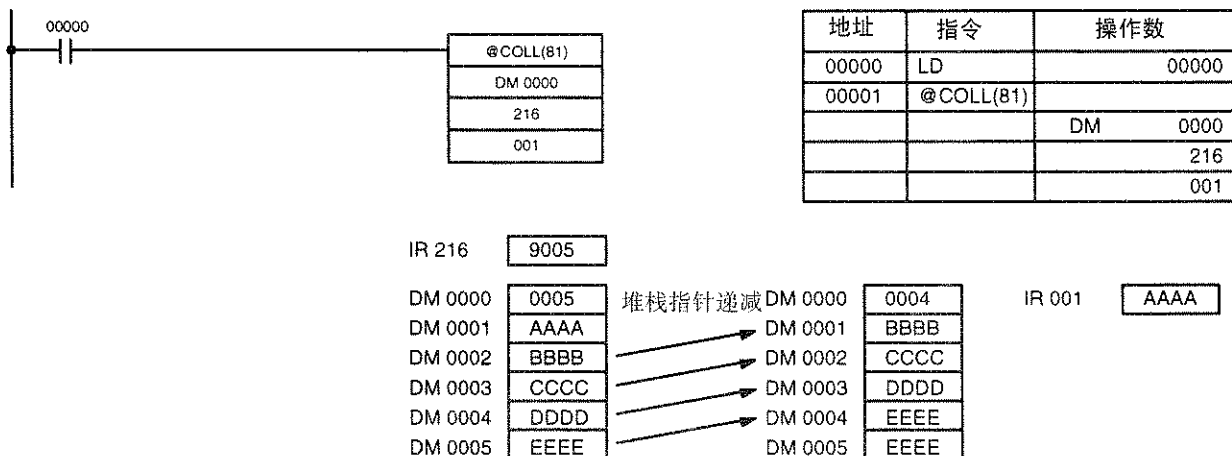
当 C 的 12 位 ~ 15 位 = 9, COLL(81) 可以用于 FIFO 堆栈操作。C 的其它三个数字指定堆栈中的字数 (000 ~ 999), SBs 中的内容为堆栈指针。
当执行条件为 ON 时, COLL(81) 把堆栈中的各个字的内容向下移动一个地址, 最后把 SBs + 1 (写进堆栈的第一个值) 的数据转移到目的字 (D), 然后堆栈指针的内容 (SBs) 递减 1。

注 COLL(81) 每个循环都执行,除非使用微分形式 (@COLL(81)) 或者 COLL(81) 同 DIFU(13) 或 DIFD(14) 一起组合使用。

例

下例示出了如何使用 COLL(81) 在 DM0001 和 DM0005 之间产生一个堆栈。DM0000 作为堆栈指针。

当 IR00000 从 OFF 变成 ON, COLL(81) 把 DM0002 ~ DM0005 的内容向下移动一个地址, 将 DM0001 的数据移到 IR001。然后堆栈指针的内容 (DM0000) 减 1。



LIFO 堆栈操作

当 C 的 12 位 ~ 15 位 = 8, COLL(81) 可以用于 LIFO 堆栈操作。C 的其它三个数字指定堆栈的字数 (000 ~ 999)。SBs 的内容是堆栈指针。

当执行条件为 ON 时, COLL(81) 将由堆栈指针 (SBs + SBs 的内容) 指示的字的的数据复制到目的字 (D) 中。然后堆栈指针 (SBs) 的内容减 1。

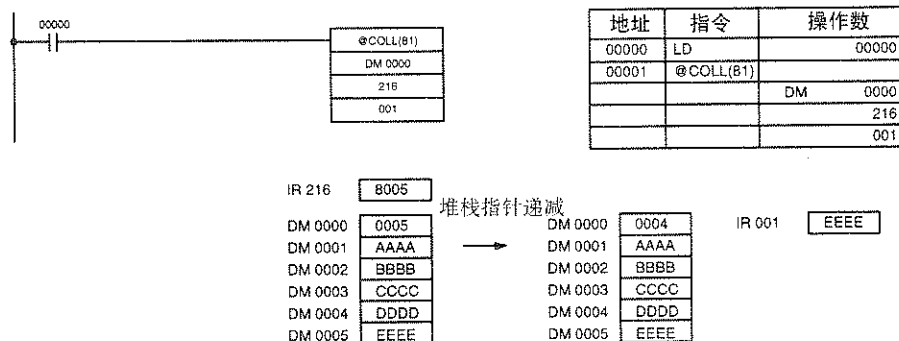
堆栈指针是堆栈中唯一要修改的字。

注 COLL(81) 每个循环都执行,除非使用微分形式 (@COLL(81)) 或 COLL(81) 同 DIFU(13) 或 DIFD(14) 组合使用。

例

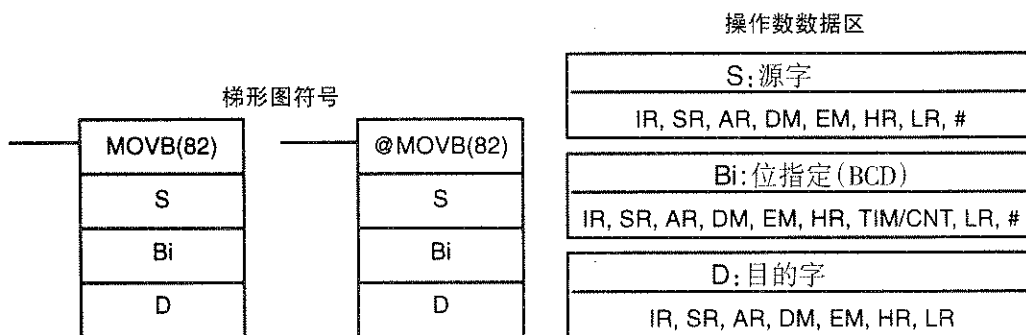
下例示出了如何 COLL(81) 以在 DM0001 与 DM0005 之间产生一个堆栈, DM0000 是堆栈指针。

当 IR00000 从 OFF 变为 ON 时, COLL(81) 将 DM0005 (DM0000 + 5) 的内容复制到 IR001, 然后堆栈指针 (DM0000) 的内容减 1。



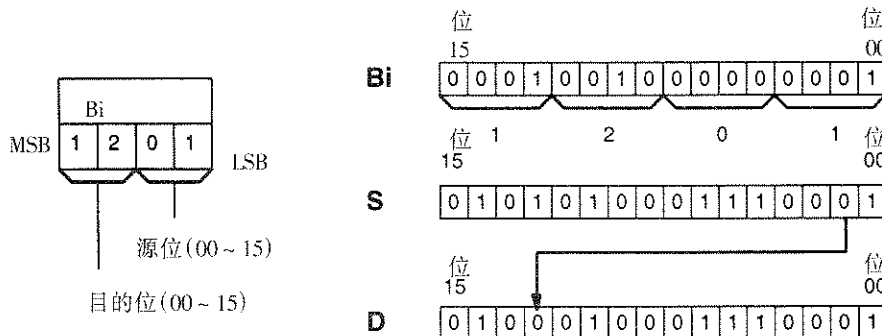
标志 ER: 控制字中的偏移或堆栈深度不是 BCD 码。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)
 堆栈操作中, 堆栈指针的内容超过堆栈深度, 试图写到超出堆栈底部的地方。
 EQ: 当 S 的内容为 0 时置 ON; 否则为 OFF

5-18-8 位传递—MOVB(82)



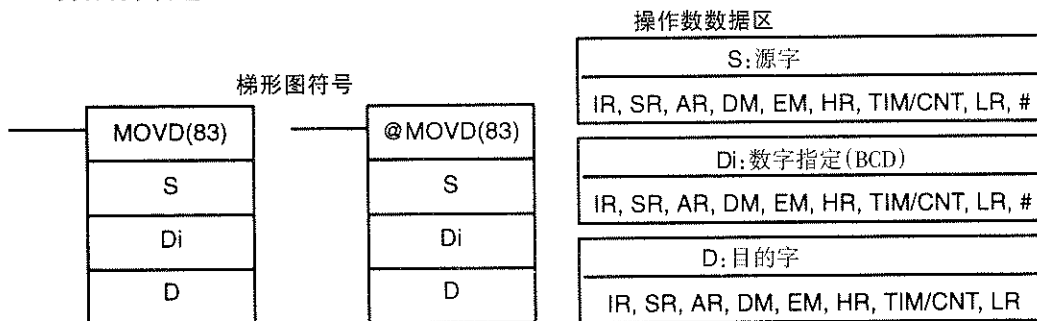
限制 Bi 的最右边两个数字和最左边的两个数字都必须在 00 ~ 15 之间。
 Bi 或 D 不能使用 DM6144 ~ DM6655。

说明 当执行条件为 OFF 时, MOVC(82) 不执行, 执行条件为 ON 时, MOVB(82) 将 S 的指定位复制到 D 的指定位, S 和 D 中的位由 Bi 指定。Bi 的最右边的两个数字指定源位; 最左边的两个数字指定目的位。



标志 ER: Bi 不是 BCD 码, 或者它指定的位不存在 (即位必须指定在 00 ~ 15 之间)。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD 码, 或者超 EM/DM 区的边界。)

5—18—9 数据传送—MOVD(83)

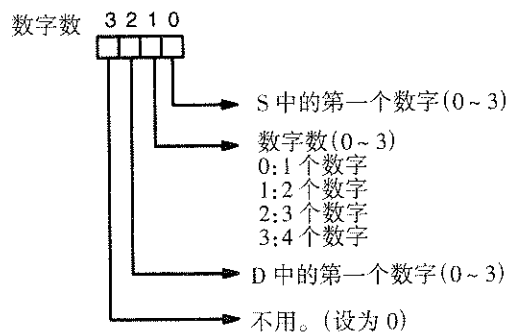


限制

Di 最右边的三个数字都必须在 0~3 之间。
Di 或 D 不能使用 DM6144 ~ DM6655。

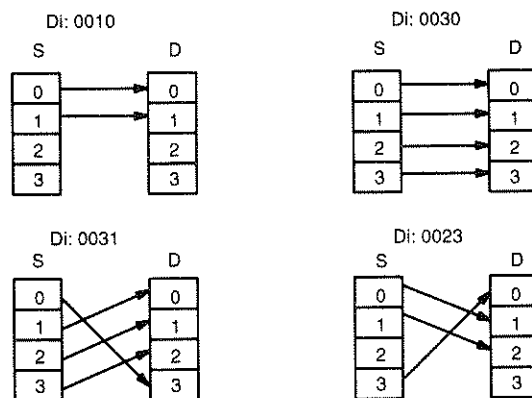
说明

当执行条件为 OFF 时, MOVD(83) 不执行, 当执行条件为 ON 时, MOVD(83) 将 S 中指定数字的内容复制到 D 中指定数字, 一次最多可以传送 4 个数字。要复制的第一个数字, 要复制的数字数以及第一个要接收复制的数字都由 Di 指定, 如下所示。S 中的数字将被复制到 D 中的连续的数字中, 复制从指定的第一个数字开始, 连续至指定的数字数为止。如果到达 S 或 D 中最后一个数字, 则回到数字 0 开始用下面的数字。



数字指定

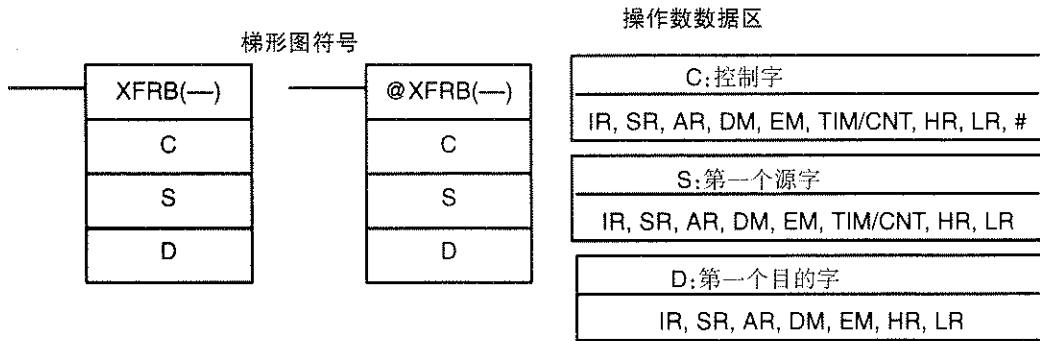
下面示出了对于不同的 Di 值, 数据传送的例子。



标志

ER: Di 的最右边三个数字中, 至少有一个不在 0 和 3 之间。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

5-18-10 多位传送—XFRB(—)

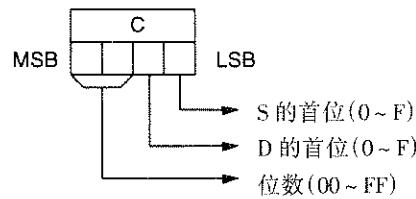


限制

指定的源位必须在同一数据区中。
指定的目的位必须在同一数据区中。
D 不能使用 DM6144 ~ DM6655。

说明

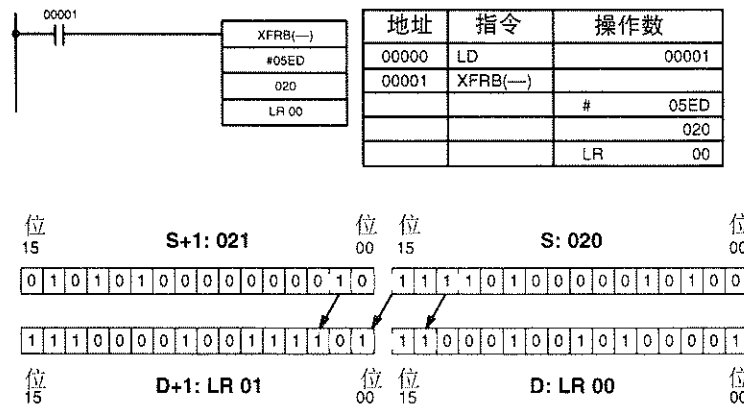
当执行条件为 OFF 时, XFRB(—) 不执行, 当执行条件为 ON 时, XFRB(—) 把指定源位复制到指定目的位。C 的最右边的两个数字指定 S 和 D 中的起始位, 最左边的两个数字表示要复制的位数。



注 一次最多可以复制 255(FF)位。

例

在下面的例子中, XFRB(—) 把从 IR020 和 IR021 的 5 个位传送到 LR00 和 LR01。IR020 中的起始位是 D(13), LR00 中的起始位是 E(14), 所以 IR02013 ~ IR02101 复制到 LR0014 ~ LR0102。



标志

ER: 指定的源位不全在同一数据区中。
指定的目的位不全在同一数据区中。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

5—19 比较指令

5—19-1 比较 - CMP(20)



限制 当把一个数值与定时器或计数器的当前值进行比较时，此数值必须是 BCD 码。

说明 当执行条件为 OFF 时，CMP(20) 不执行，当执行条件为 ON 时，CMP(20) 比较 CP1 和 CP2，输出结果到 SR 区的 GR、EQ 和 LE 标志中。

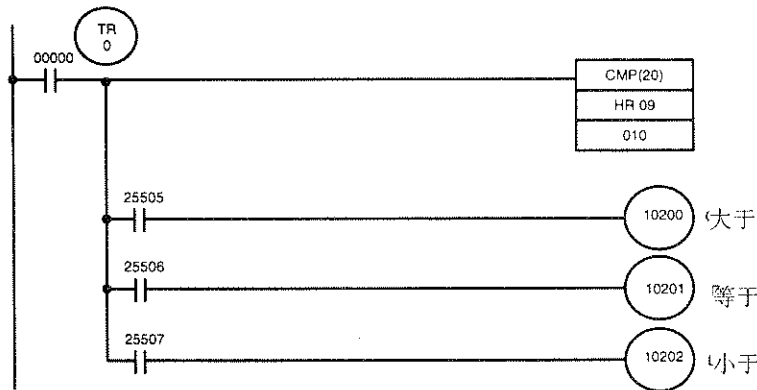
注意 在 CMP(20) 和访问 EQ、LE 和 GR 标示的操作之间放上其它指令，可能会改变这些标志的状态。注意在所要得到的状态改变之前访问它们。

标志 ER: 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)
 EQ: 如果 CP1 等于 CP2, 置 ON。
 LE: 如果 CP1 小于 CP2, 置 ON。
 GR: 如果 CP1 大于 CP2, 置 ON。

| 标志 | 地址 | C1 < C2 | C1 = C2 | C1 > C2 |
|----|-------|---------|---------|---------|
| GR | 25505 | OFF | OFF | ON |
| EQ | 25506 | OFF | ON | OFF |
| LE | 25507 | ON | OFF | OFF |

例
保存 CMP(20) 结果

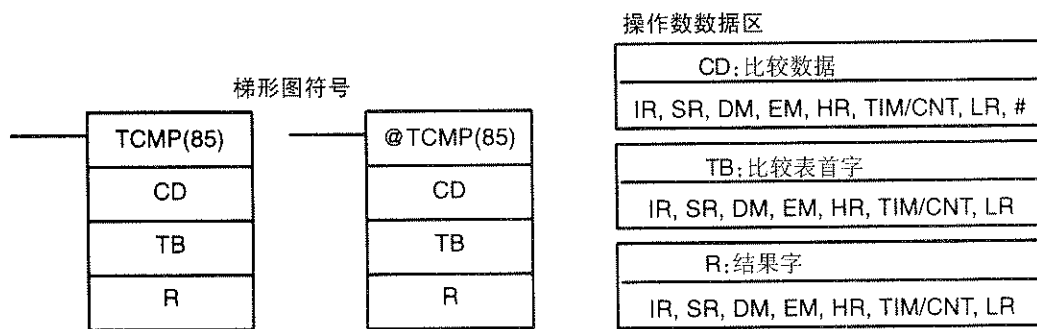
下例示出了如何立刻保存比较结果。如果 HR09 的内容大于 010 的内容, 10200 置为 ON; 如果两个内容相等, 10201 置为 ON; 如果 HR09 的内容比 010 的内容小, 10202 置为 ON。在一些应用中, 三个输出只有一个是有必要的, 这样就不必使用 TR0。这样编程, 只有执行 CMP(20) 时, 10200, 10201 和 10202 才改变。



| 地址 | 指令 | 操作数 |
|-------|---------|--------------|
| 00000 | LD | 00000 |
| 00001 | OUT | TR 0 |
| 00002 | CMP(20) | HR 09 010 |
| 00003 | AND | 25505 |
| 00004 | OUT | 10200 |

| 地址 | 指令 | 操作数 |
|-------|-----|-------|
| 00005 | LD | TR 0 |
| 00006 | AND | 25506 |
| 00007 | OUT | 10201 |
| 00008 | LD | TR 0 |
| 00009 | AND | 25507 |
| 00010 | OUT | 10202 |

5-19-2 表比较——TCMP(85)



限制

R 不能使用 DM6144 ~ DM6655。

说明

当执行条件为 OFF 时, TCMP(85) 不执行。当执行条件为 ON 时, TCMP(85) 将 CD 与 TB, TB + 1, TB + 2, ..., TB + 15 的内容进行比较。如果 CD 等于这些字中的任一个字的内容, R 中的相应位置位。例如如果 CD 等于 TB 的内容, 00 位置 ON, 如果它等于 TB + 1 的内容, 01 位置 ON, 等等。R 中的其它位置为 OFF。

标志

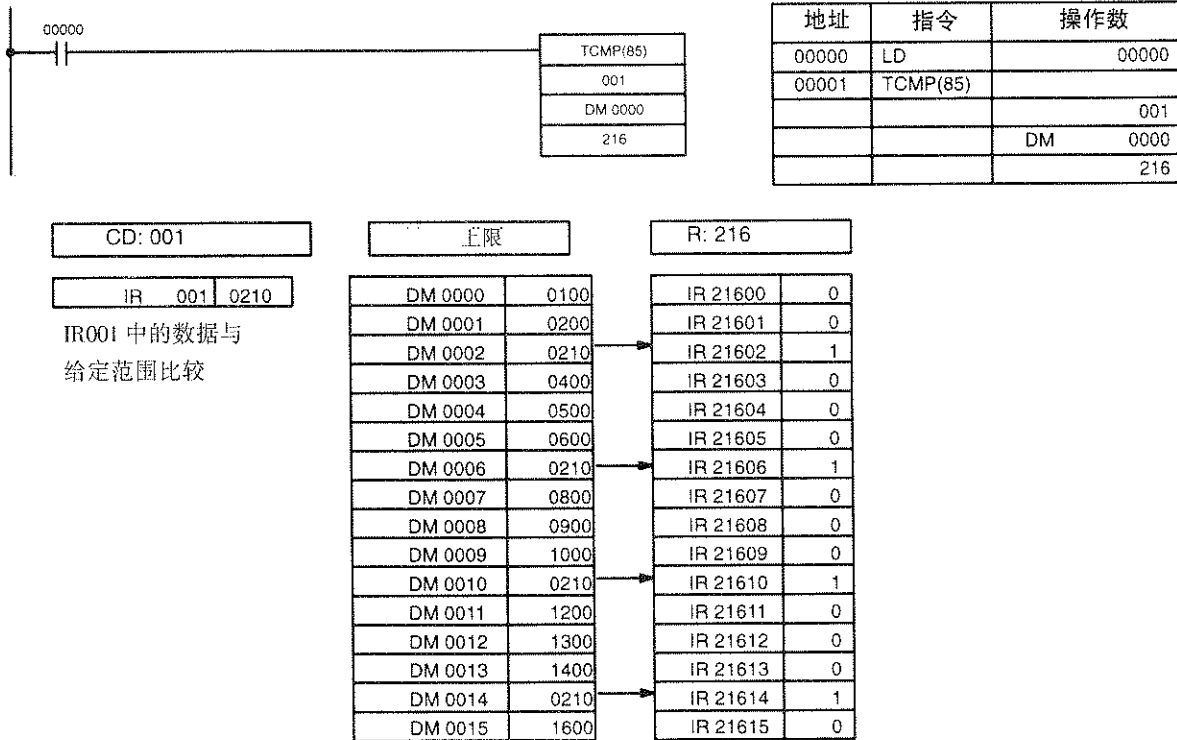
ER: 比较表 (即, TB ~ TB + 15) 超出数据区。

间接寻址 EM/DM 字不存在。

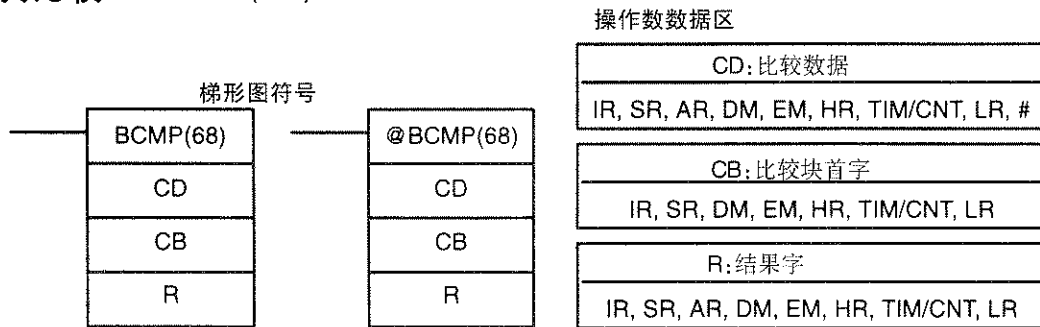
(*EM/*DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

例

下面的例子示出用 TCMP(85) 进行比较和所得结果。这里,当 IR0000 为 ON, 每个循环都进行比较。



5-19-3 块比较—BCMP(68)



限制

比较块中的各个下限必须小于或等于上限。
R 不能使用 DM6144 ~ DM6655。

说明

当执行条件为 OFF 时, BCMP(68) 不执行, 当执行条件为 ON 时, BCMP(68) 将 CD 与由 CB, CB + 1, CB + 2, …… , CB + 31 组成的块所定义的范围进行比较, 每个范围由两个字定义, 第一个字是下限, 第二个字是上限。如果 CD 处在这些范围之内(包括上下限), 在 R 中的相应的位被置位。所进行的比较和对应于每个比较 R 中相应位在每个比较为真时置位的情况下如下所示。R 中的其余位置为 OFF。

| | |
|--------------------------------|------|
| $CB \leq CD \leq CB + 1$ | 位 00 |
| $CB + 2 \leq CD \leq CB + 3$ | 位 01 |
| $CB + 4 \leq CD \leq CB + 5$ | 位 02 |
| $CB + 6 \leq CD \leq CB + 7$ | 位 03 |
| $CB + 8 \leq CD \leq CB + 9$ | 位 04 |
| $CB + 10 \leq CD \leq CB + 11$ | 位 05 |
| $CB + 12 \leq CD \leq CB + 13$ | 位 06 |
| $CB + 14 \leq CD \leq CB + 15$ | 位 07 |
| $CB + 16 \leq CD \leq CB + 17$ | 位 08 |
| $CB + 18 \leq CD \leq CB + 19$ | 位 09 |
| $CB + 20 \leq CD \leq CB + 21$ | 位 10 |
| $CB + 22 \leq CD \leq CB + 23$ | 位 11 |
| $CB + 24 \leq CD \leq CB + 25$ | 位 12 |
| $CB + 26 \leq CD \leq CB + 27$ | 位 13 |
| $CB + 28 \leq CD \leq CB + 29$ | 位 14 |
| $CB + 30 \leq CD \leq CB + 31$ | 位 15 |

标志

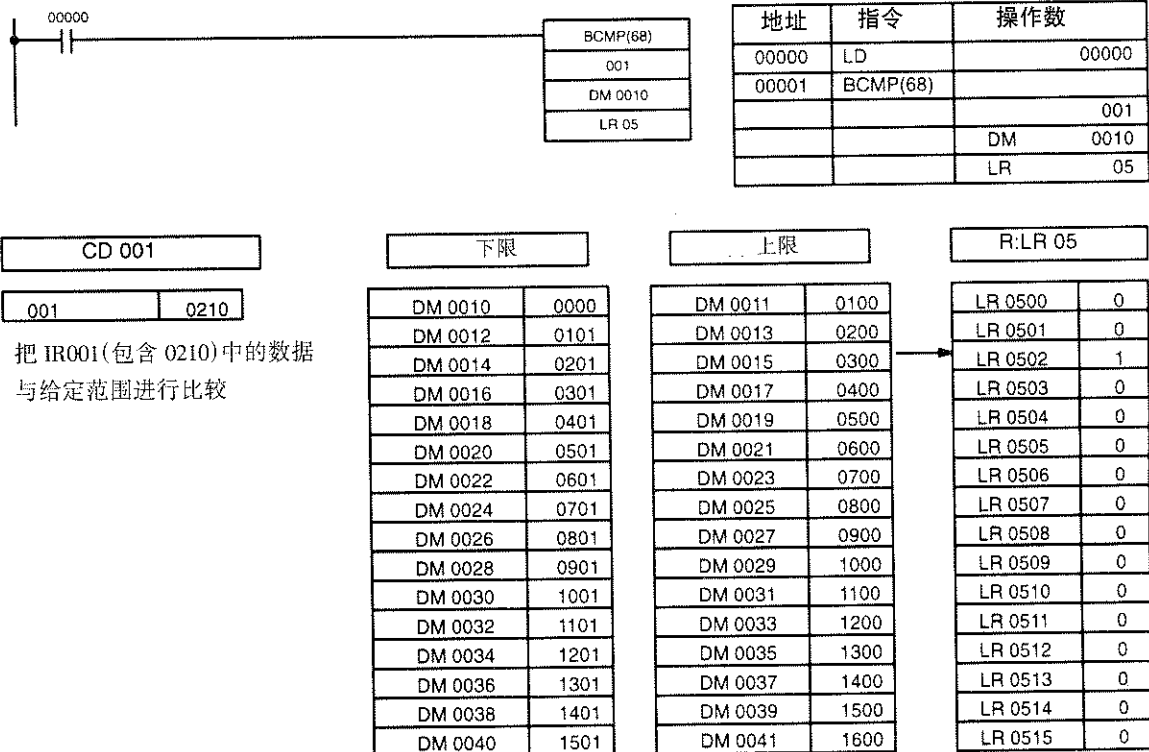
ER: 比较块(即 CB ~ CB + 31)超出数据区。

间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD 码, 或者超出 EM/DM 边界。)

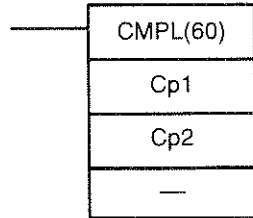
例

下例示出了 BCMP(68)进行的比较与所得到的结果。当 IR00000 为 ON 时, 每个循环都进行比较。



5-19-4 双字比较 - CMPL(60)

梯形图符号



操作数数据区

| |
|-------------------------------------|
| Cp1: 第一对比较字的首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| Cp2: 第二对比较字的首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |

限制

CP1 和 CP1 + 1 必须在同一个数据区。
 CP2 和 CP2 + 1 必须在同一数据区。
 把第三个操作数设为 000。

说明

当执行条件为 OFF 时, CMPL(60) 不执行, 当执行条件为 ON 时, CMPL(60) 将 CP1 + 1 的 4 位十六进制数与 CP1 的结合起来, 将 CP2 + 1 的与 CP2 的结合起来, 产生两个 8 位十六进制数 CP1 + 1, CP1 与 CP2 + 1, CP2, 然后比较这两个 8 位数, 结果输出到 SR 区中的 GR、EQ 和 LE 标志。

注意

在 CMPL(60) 和访问 EQ, LE 和 GR 标志的操作之间放上其它指令, 可能会改变这些标志的状态, 确保在所想要的状态改变之前访问它们。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

GR: 置 ON, 如果 CP1 + 1, CP1 大于 CP2 + 1, CP2。

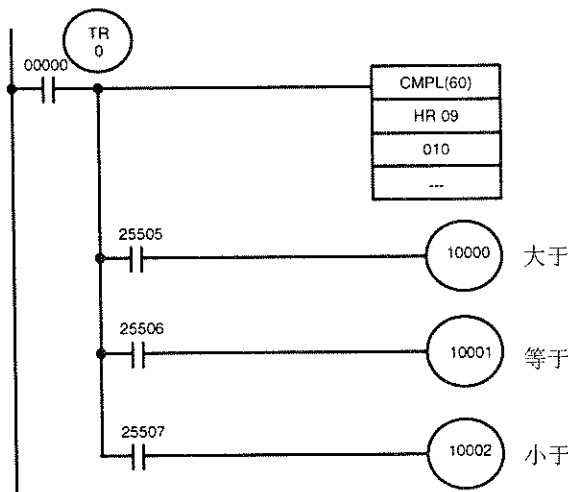
EQ: 置 ON, 如果 CP1 + 1, CP1 等于 CP2 + 1, CP2。

LE: 置 ON, 如果 CP1 + 1, CP1 小于 CP2 + 1, CP2。

例

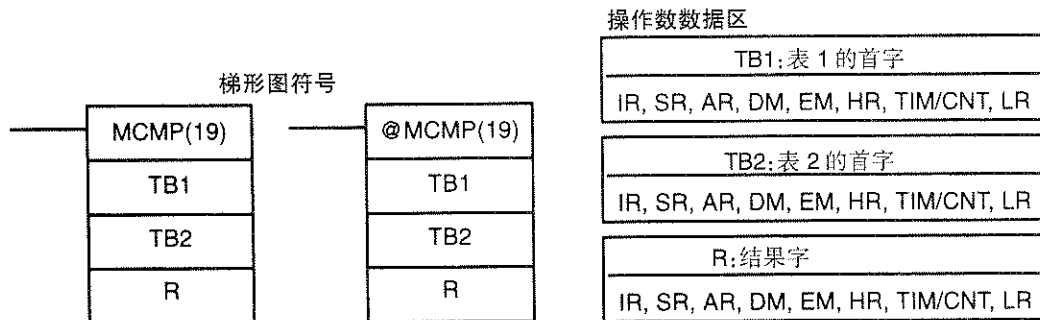
保存 CMPL(60) 结果

下例示出如何立即保存比较结果, 如果 HR10, HR09 的内容大于 011, 010 的内容, 10000 置 ON, 如果两个内容相等, 10001 置 ON, 如果 HR10, HR09 的内容小于 011, 010 的内容, 10002 置 ON。在一些应用中, 三个输出只有一个是必要的, 这样就不必使用 TR0 编程。在这些类型编程中, 只有当 CMPL(60) 执行时, 10000、100001 和 10002 才改变。



| 地址 | 指令 | 操作数 |
|-------|----------|--------------|
| 00000 | LD | 00000 |
| 00001 | OUT | TR 0 |
| 00002 | CMPL(60) | HR 09 010 |
| 00003 | AND | 25505 |
| 00004 | OUT | 10000 |
| 00005 | LD | TR 0 |
| 00006 | AND | 25506 |
| 00007 | OUT | 10001 |
| 00008 | LD | TR 0 |
| 00009 | AND | 25507 |
| 00010 | OUT | 10002 |

5—19—5 多字比较 - MCMP(19)



限制

TB1 与 TB1 + 15 必须在同一数据区。
 TB2 与 TB2 + 15 必须在同一数据区。
 R 不能使用 DM6144 ~ DM6655。

说明

当执行条件为 OFF 时,MCMP(19)不执行。当执行条件为 ON 时,MCMP(19)比较 TB1 与 TB2, TB1 + 1 与 TB2 + 1, TB1 + 2 与 TB2 + 2, ..., TB1 + 15 与 TB2 + 15 的内容。如果第一对相等,R 中的第一位置 OFF,等等,即,如果 TB1 的内容等于 TB2 的内容,位 00 置 OFF,如果 TB1 + 1 的内容等于 TB2 + 1 的内容,位 01 置 OFF,等等,R 中的其它位置 ON。

标志

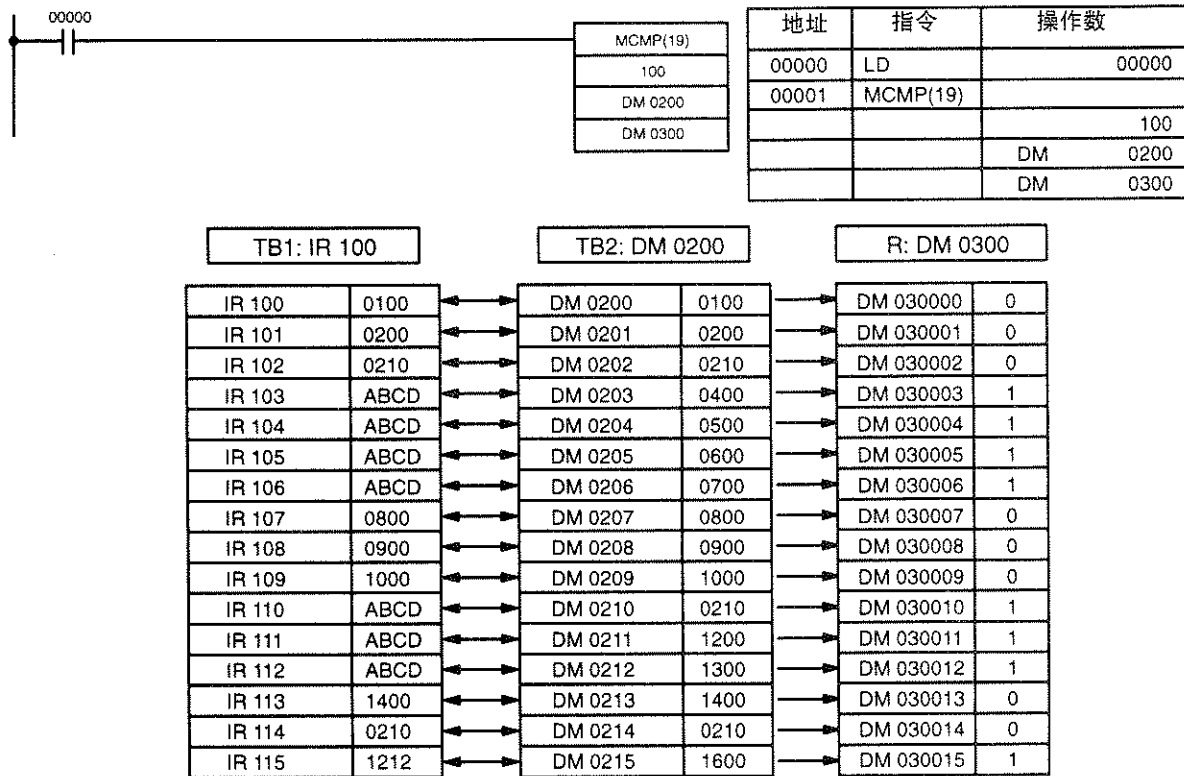
ER: 有一个表(即 TB1 ~ TB1 + 15,或 TB2 ~ TB2 + 15)超出数据区。
 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD 码,或者超出 EM/DM 区边界。)

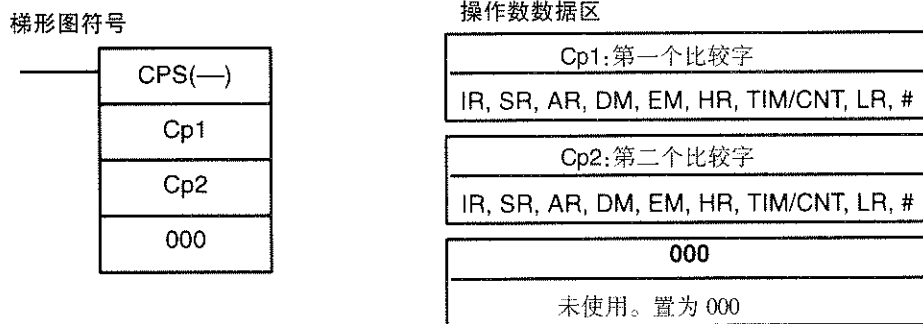
EQ: 如果两个表的全部内容相等且 R = 0000 时置 ON

例

下例示出了用 MCMP(19)进行的比较与所得到的结果。当 00000 为 ON 时, 每个循环都进行比较。



5—19—6 有符号二进制比较—CPS(—)



说明

当执行条件为 OFF 时, CPS(—) 不执行, 当执行条件为 ON 时, CPS(—) 比较 CP1 与 CP2 中的 16 位 (4 个数字) 有符号二进制数, 并输出结果到 SR 区中的 GR、EQ 和 LE 标志。

注意

在 CPS(—) 和访问 EQ、LE 和 GR 标志的操作之间放上其它指令, 可能会改变这些标志的状态, 确保在所要得到的状态改变之前访问它们。

标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

EQ: 如果 CP1 等于 CP2, 则置 ON。

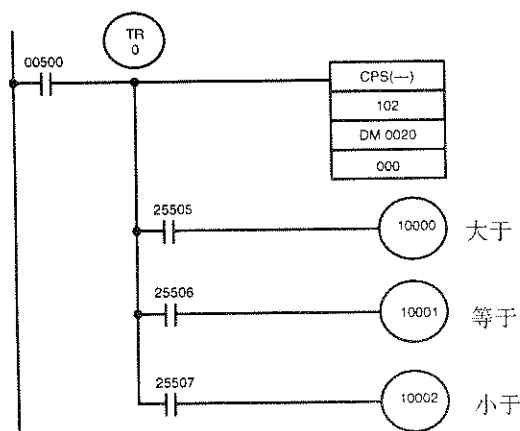
LE: 如果 CP1 小于 CP2, 则置 ON。

GR: 如果 CP1 大于 CP2, 则置 ON。

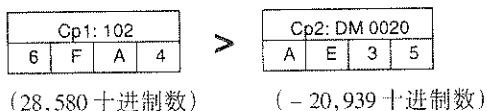
| 比较结果 | 标志状态 | | |
|-----------|---------------|---------------|---------------|
| | GR (SR 25505) | EQ (SR 25506) | LE (SR 25507) |
| Cp1 < Cp2 | 0 | 0 | 1 |
| Cp1 = Cp2 | 0 | 1 | 0 |
| Cp1 > Cp2 | 1 | 0 | 0 |

例

在下面的例子中, 102 的内容大于 DM0020 的内容。因此, 10000 置为 ON, 其它位, 10001 和 10002 置为 OFF。

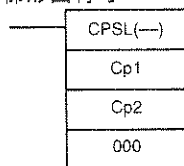


| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD | 00500 |
| 00001 | OUT | TR 0 |
| 00002 | CPS(-) | |
| | | 102 |
| | | DM 0020 |
| | | 000 |
| 00003 | AND | 25505 |
| 00004 | OUT | 10000 |
| 00005 | LD | TR 0 |
| 00006 | AND | 25506 |
| 00007 | OUT | 10001 |
| 00008 | LD | TR 0 |
| 00009 | AND | 25507 |
| 00010 | OUT | 10002 |



5-19-7 双字有符号二进制比较—CPLS(-)

梯形图符号



操作数数据区

| |
|--|
| Cp1: 第一个比较字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Cp2: 第二个比较字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| 000 |
| 未使用。置为 000 |

说明

当执行条件为 OFF 时, CPLS(-) 不执行, 当执行条件为 ON 时, CPLS(-) 将 CP1 + 1, CP1 与 CP2 + 1, CP2 中的 32 位 (8 位数字) 有符号二进制数进行比较, 并将结果输出到 SR 区的 GR、EQ 和 LE 标志。

注意

在 CPSL(—)和访问 EQ、LE 和 GR 标志的操作之间放上其它指令,可能会改变这些标志的状态。要确保在所想要的状态改变之前访问它们。

标志

ER:间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 码,或者超出 EM/DM 区边界)。

EQ:置 ON,如果 CP1 + 1,CP1 等于 CP2 + 1,CP2。

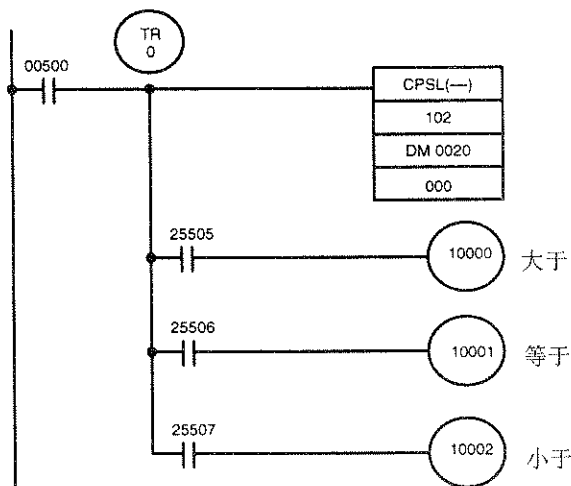
LE:置 ON,如果 CP1 + 1,CP1 小于 CP2 + 1,CP2。

GR:置 ON,如果 CP1 + 1,CP1 大于 CP2 + 1,CP2。

| 比较结果 | 标志状态 | | |
|-----------------------------|--------------|--------------|--------------|
| | GR(SR 25505) | EQ(SR 25506) | LE(SR 25507) |
| Cp1 + 1, Cp1 < Cp2 + 1, Cp2 | 0 | 0 | 1 |
| Cp1 + 1, Cp1 = Cp2 + 1, Cp2 | 0 | 1 | 0 |
| Cp1 + 1, Cp1 > Cp2 + 1, Cp2 | 1 | 0 | 0 |

例

在下例中,103,102 的内容小于 DM0021,DM0020 的内容,因此 10002 置为 ON,其它,10000 和 10001 置为 OFF。



| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00500 |
| 00001 | OUT | TR 0 |
| 00002 | CPSL(—) | |
| | | 102 |
| | | DM 0020 |
| | | 000 |
| 00003 | AND | 25505 |
| 00004 | OUT | 10000 |
| 00005 | LD | TR 0 |
| 00006 | AND | 25506 |
| 00007 | OUT | 10001 |
| 00008 | LD | TR 0 |
| 00009 | AND | 25507 |
| 00010 | OUT | 10002 |

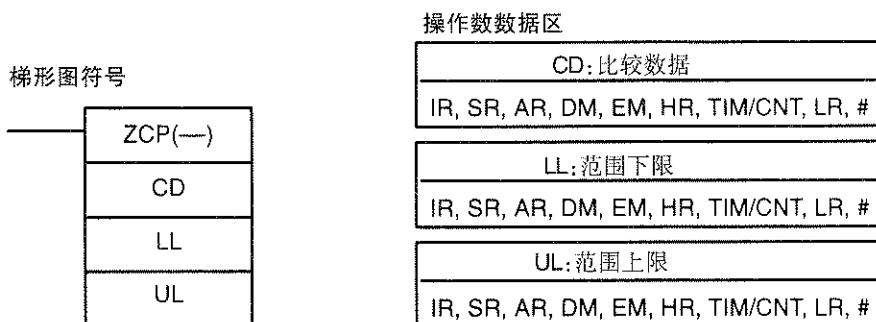
| Cp1+1: 103 | | | | Cp1: 102 | | | |
|------------|---|---|---|----------|---|---|---|
| 8 | 2 | B | 6 | F | 5 | 7 | B |

(-2,101,938,823 十进制数)

| Cp2+1: DM 0021 | | | | Cp2: DM 0020 | | | |
|----------------|---|---|---|--------------|---|---|---|
| 0 | 5 | 6 | A | 9 | 9 | D | B |

(90,872,283 十进制数)

5-19-8 区域范围比较 - ZCP(一)



限制

LL 必须小于或等 UL。

说明

当执行条件为 OFF 时, ZCP(一) 不执行。当执行条件为 ON 时, ZCP(一) 将 CD 与由下限 LL 和上限 UL 定义的范围进行比较, 并且把结果输出到 SR 区中的 GR, EQ 和 LE 标志。结果标志状态如下所示。

| 比较结果 | 标志状态 | | |
|--------------|--------------|--------------|--------------|
| | GR(SR 25505) | EQ(SR 25506) | LE(SR 25507) |
| CD < LL | 0 | 0 | 1 |
| LL ≤ CD ≤ UL | 0 | 1 | 0 |
| UL < CD | 1 | 0 | 0 |

注意

在 ZCP(一) 和访问 EQ, LE 和 GR 标志的操作之间放上其它指令, 可能会改变这些标志的状态。要确保在所要得到的状态改变之前访问它们。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界。)

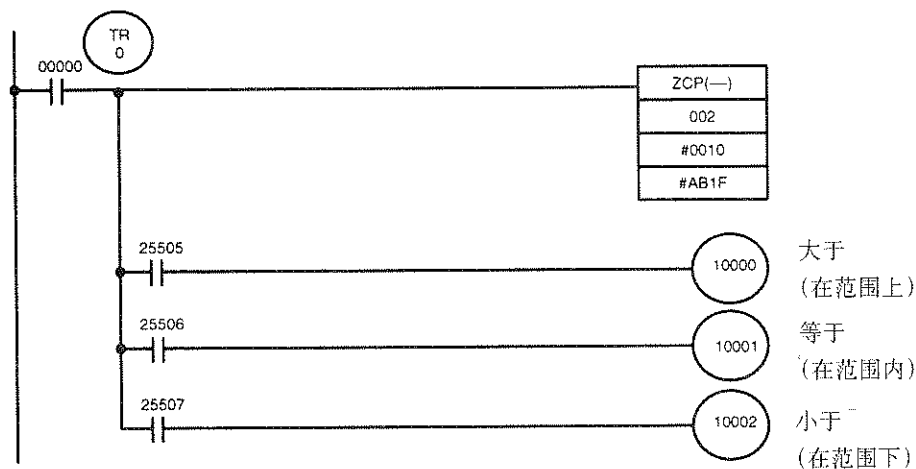
EQ: 置 ON, 如果 LL ≤ CD ≤ UL

LE: 置 ON, 如果 CD < LL

GR: 置 ON, 如果 CD > UL

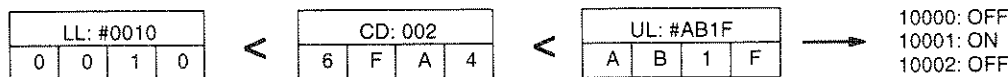
例

在下例中, IR002 的内容 (# 6FA4) 与范围 # 0010 ~ # AB1F 进行比较, 因为 # 0010 ≤ # 6FA4 ≤ # AB1F, EQ 标志与 IR10001 置 ON。

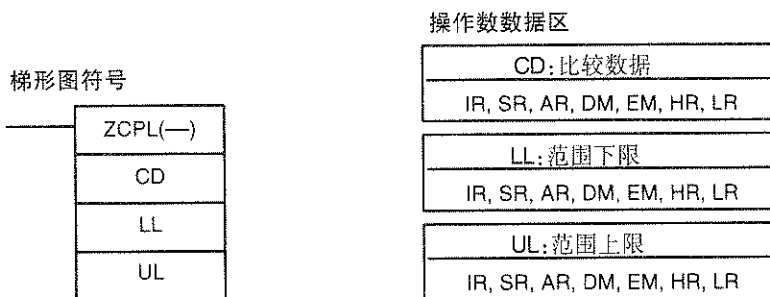


| 地址 | 指令 | 操作数 |
|-------|--------|--------|
| 00000 | LD | 00000 |
| 00001 | OUT | TR 0 |
| 00002 | ZCP(-) | |
| | | 002 |
| | | # 0010 |
| | | # AB1F |
| 00003 | AND | 25505 |

| 地址 | 指令 | 操作数 |
|-------|-----|-------|
| 00004 | OUT | 10000 |
| 00005 | LD | TR 0 |
| 00006 | AND | 25506 |
| 00007 | OUT | 10001 |
| 00008 | LD | TR 0 |
| 00009 | AND | 25507 |
| 00010 | OUT | 10002 |



5-19-9 双字区域范围比较 - ZCPL(-)



限制

LL + 1, LL 中的 8 位数字数值必须小于或等于 UL + 1, UL。

说明

当执行条件为 OFF 时, ZCPL(-) 不执行。当执行条件为 ON 时, ZCPL(-) 将 CD, CD + 1 中的 8 位数字数值与由下限 LL + 1, LL 和上限 UL + 1, UL 定义的范围进行比较, 并将结果输出到 SR 区中的 GR, EQ 和 LE 标志, 结果标志状态如下表所示。

| 比较结果 | 标志状态 | | |
|--------------------------------------|------------------|------------------|------------------|
| | GR (SR 25505) | EQ (SR 25506) | LE (SR 25507) |
| CD, CD + 1 < LL + 1, LL | 0 | 0 | 1 |
| LL + 1, LL ≤ CD, CD + 1 ≤ UL + 1, UL | 0 | 1 | 0 |
| UL + 1, UL < CD, CD + 1 | 1 | 0 | 0 |

注意 在 ZCPL(—) 和访问 EQ, LE 和 GR 标志的操作之间放上其它指令, 可能会改变这些标志的状态。要确保在所要得到的状态改变之前访问它们。

标志 ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD 码, 或者超出 EM/DM 区边界)。

LL + 1, LL 大于 UL + 1, UL。

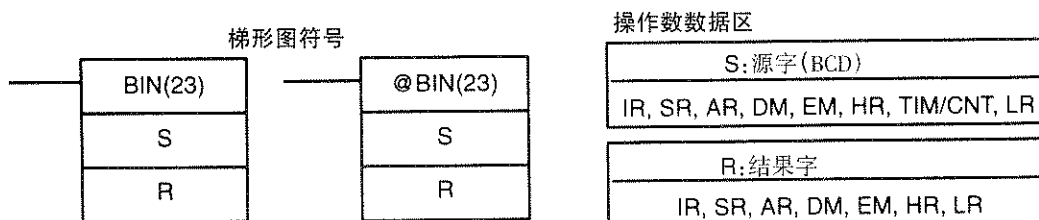
EQ: 置 ON, 如果 LL + 1, LL ≤ CD, CD + 1 ≤ UL + 1, UL。

LE: 置 ON, 如果 CD, CD + 1 < LL + 1, LL。

GR: 置 ON, 如果 CD, CD + 1 > UL + 1, UL。

5-20 转换指令

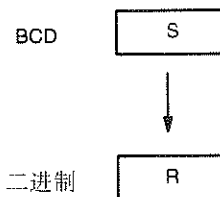
5-20-1 BCD 转换为二进制 - BIN(23)



限制说明

DM6144 ~ DM6655 不能用作 R。

当执行条件为 OFF, BIN(23) 不执行。当执行条件为 ON, BIN(23) 将 S 的 BCD 内容转换为等值的二进制数, 并将二进制值输出到 R。仅仅 R 的内容被改变; S 的内容保留不变。



BIN(23) 用来将 BCD 转换为二进制, 以便编程器或任何其它编程设备的显示表示为 16 进制而不是 10 进制, 它还可用来转换到二进制以执行二进制算术运算而不是执行 BCD 算术运算, 例如, 当 BCD 值与二进制值要相加时。

标志

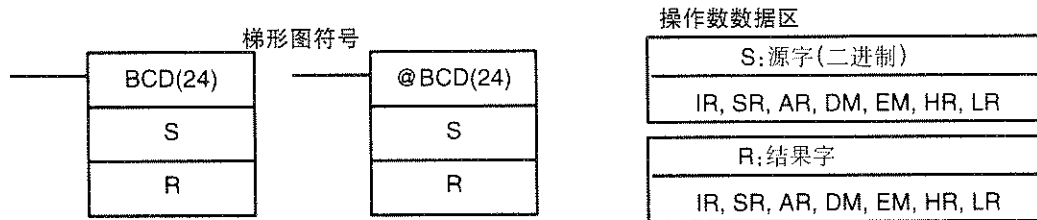
ER: S 的内容不是 BCD

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

EQ: 当结果为零时为 ON。

5-20-2 二进制转换为 BCD—BCD(24)



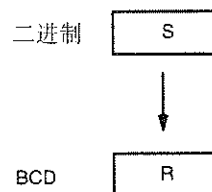
限制

若 S 的内容不大于 270F, 转换结果将超过 9999, BCD(24) 将不执行。当该指令不执行, R 的内容保持不变。

DM6144 ~ DM6655 不能用于 R。

说明

BCD(24) 将 S 中的二进制 (16 进制) 内容转换为等值的 BCD 值, 并将 BCD 值输出到 R。仅仅 R 的内容被改变; S 的内容保持不变。



BCD(24) 可用来将二进制值转换为 BCD 值, 以便在编程器上或任何其它编程设备上的显示表示为 10 进制而不是 16 进制。它还可以用来转换到 BCD 以进行 BCD 算术运算而不是执行二进制算术运算, 例如, 当 BCD 值和二进制要相加时。

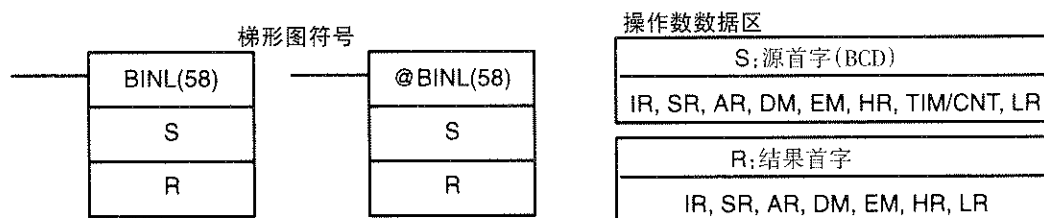
标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

EQ: 当结果为零时为 ON。

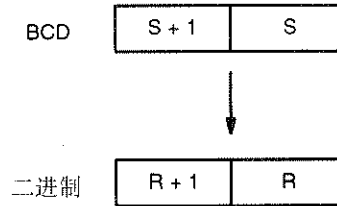
5-20-3 双字 BCD 转换为双字二进制 - BINL(58)



限制

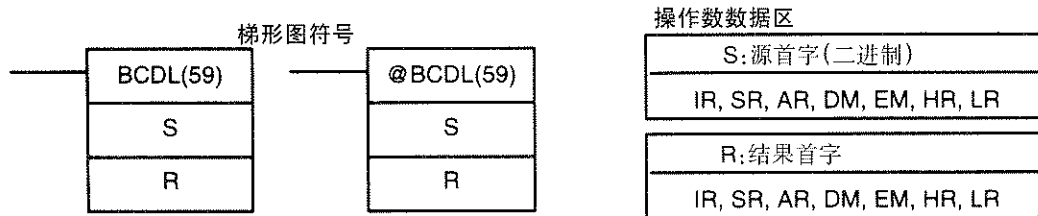
DM6143 - DM6655 不能用于 R。

说明 当执行条件为 OFF 时, BINL(58) 不执行, 当执行条件为 ON 时, BINL(58) 将 S 和 S + 1 中的 8 位数转换为 32 位的二进制值, 并将转换后的数据输出到 R 和 R + 1。



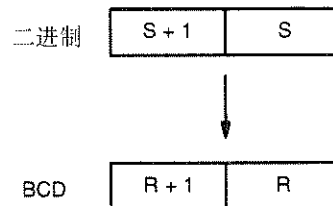
标志 ER: S 和/或 S + 1 字的内容不是 BCD。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 当结果为零时为 ON。

5-20-4 双字二进制转换为双字 BCD - BCDL(59)



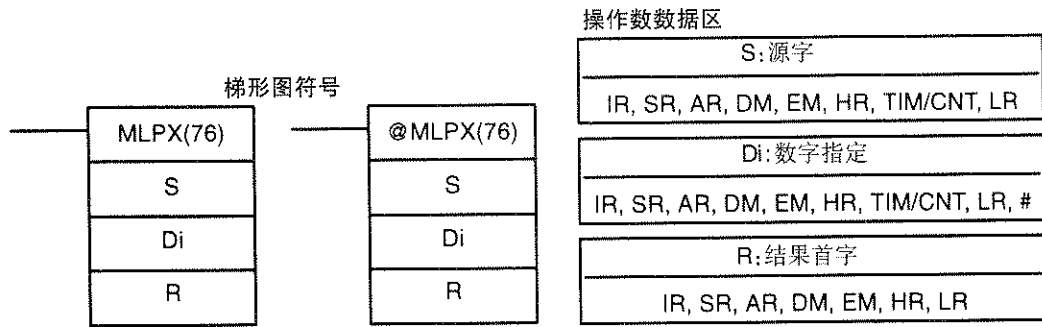
限制 若 S 的内容大于 05F5E0FF, 转换结果将超过 99999999, BCDL(59) 不执行。当该指令不执行时, R 和 R + 1 的内容保持不变。

DM6143 ~ DM6655 不能用于 R。
说明 BCDL(59) 将 S 和 S + 1 中的 32 位二进制值转换为 8 位数的 BCD 值, 并将转换数据输出到 R 和 R + 1。



标志 ER: R 和 R + 1 内容超过 99999999。
 间接寻址的 EM/DM 不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 当结果为零时为 ON。

5-20-5 4-16 译码—MLPX(76)



限制

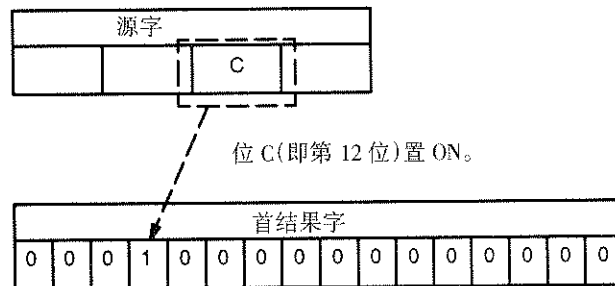
Di 最右边的两个数字必须各为 0~3 之间的值。全部结果字必须在相同数据区。

DM6144 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时,MLPX(76)不执行,当执行条件为 ON 时,MLPX(76)将 S 中最多 4 个 4 位的 16 进制数转换为 0 至 15 间的 10 进制值,每个 10 进制值用于指示位的位置。在结果字中,对应被转换值的位被置为 ON。若指定了要转换多于一个数字,从 R 开始连续的若干个字将有一个位被置 ON。(见下例)

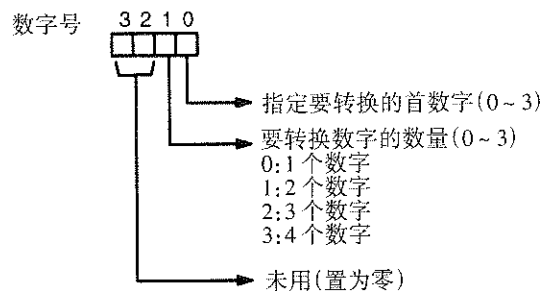
下面的例子是从 S 的一号数字进行一个数的译码操作,这里 Di 应为 0001。



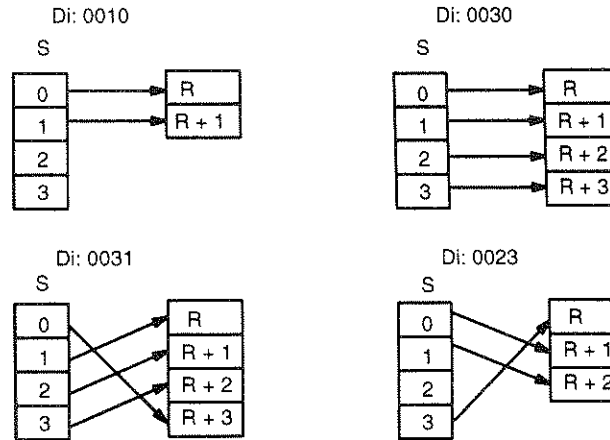
首数字和要转换数字的数量由 Di 指定,若指定了比在 S 中剩下的还要多的数字(从指定的首数字算起),余下的数字将从 S 的开始处重新开始计,要求用来存储转换结果的最后一个字(R 加上要转换的数字数量)必须与 R 同在一个数据区。例如,若要转换两个数字,数据区中最后一个地址不能指定;如果转换 3 个数字,数据区中最后 2 个字不能指定。

数字指定

Di 中的数字如下所示设定。



一些 Di 值及它们产生的数字到字之间的转换例如下所示。



标志

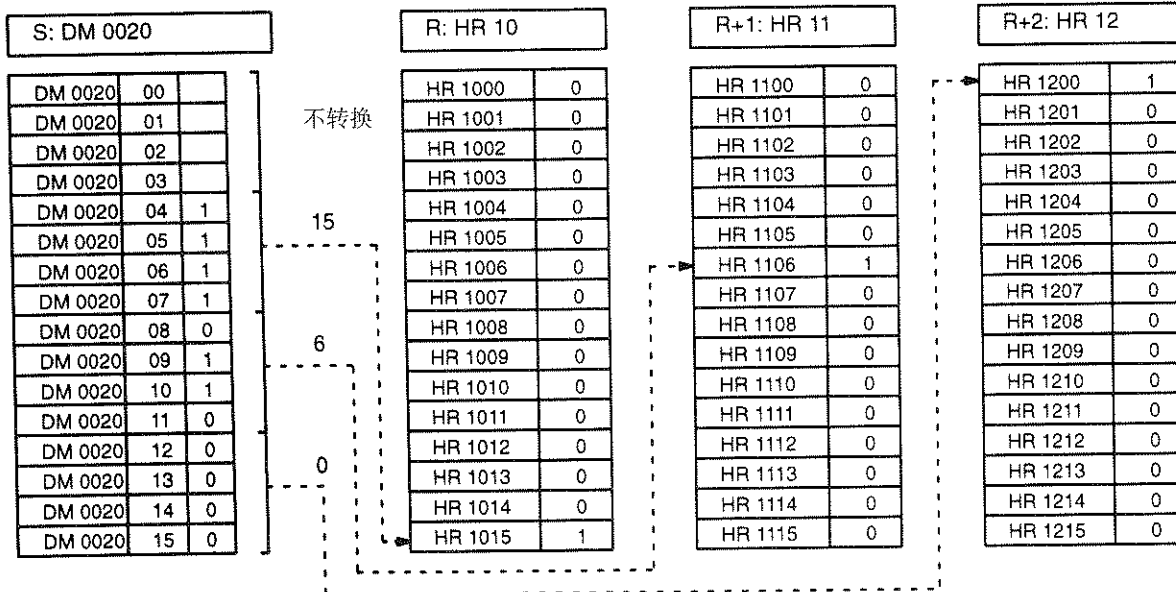
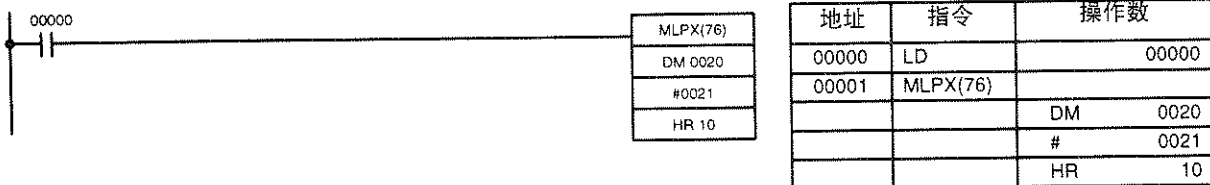
ER: 未定义数字指定符, 或 R 加数字量超出一个数据区。

间接寻址的 EM/DM 字不存在。

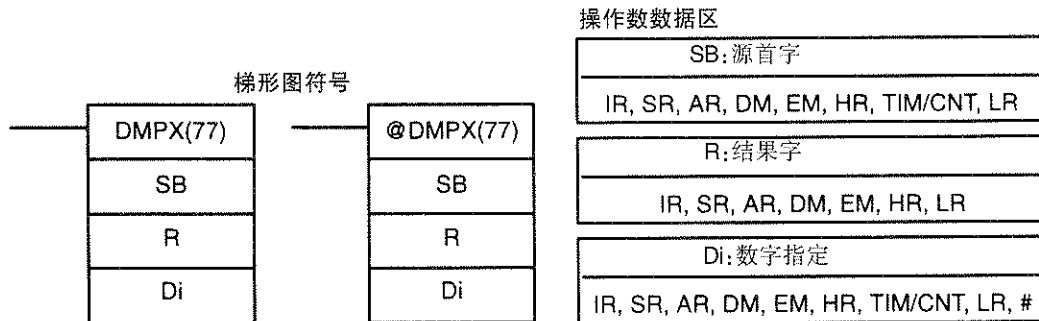
(* EM/* DM 字的内容不是 BCD 或已超出 EM/DM 区边界。)

例

下述程序将 DM0020 数据的 1~3 的数字转换为位的位置, 并且将从 HR10 开始的连续 3 个字的对应位置 ON。数字 0 不转换。



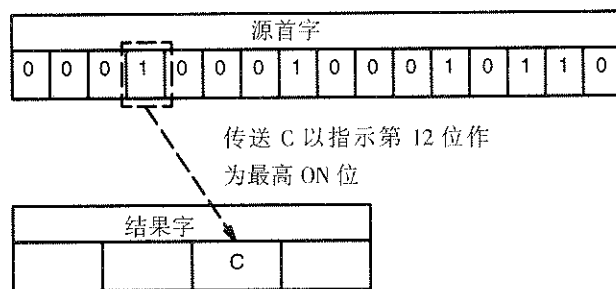
5-20-6 16-4 编码 ——DMPX(77)



Di 的最右边两个数字必须各为 0~3 之间的值。
 所有源字必须在同一数据区。
 DM6144~DM6655 不能用于 SB, R 或 Di。

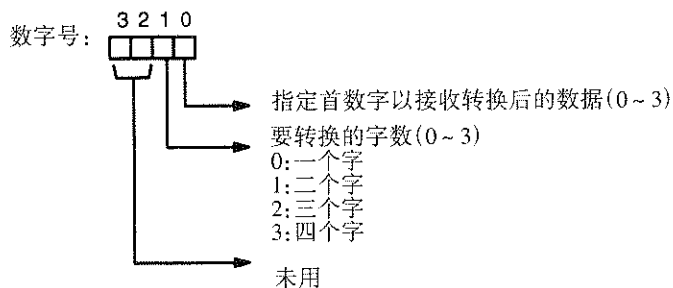
说明

当执行条件为 OFF 时, DMPX(77) 不执行。当执行条件为 ON 时, DMPX(77) 确定 S 中的最高的 ON 位位置, 将这最高 ON 位的位序号对应值编码为单个的 16 进制数字, 然后将其输出到 R 中的指定数字。接收结果的数字由 Di 指定, Di 还指定要编码数字的数量。下面是一个例子, 一个数字编码到 R 中序号为 1 的数字。这时 Di 应为 0001。

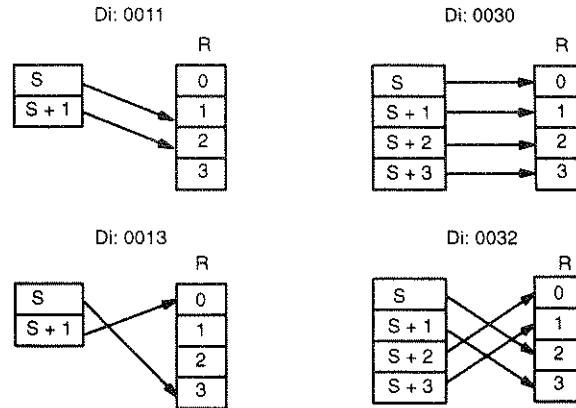


最多能有从 S 开始的连续的四个源字能被编码, 并且数字从指定的首数字按顺序写到 R, 若指定的数字多于 R 中剩余的数字 (从指定首数字开始计), 剩余的数字将从 R 的开始处重新放置。
 被转换的最后字 (S 加上要转换数字的数量) 必须与 SB 处于同一数据区。
 Di 的数字如下设置

数字标志符



一些 Di 值及由它所产生的字数字转换的例子如下所示。



标志

ER: 未定义数字, 标志符, 或 S 加数字数超出一个数据区。

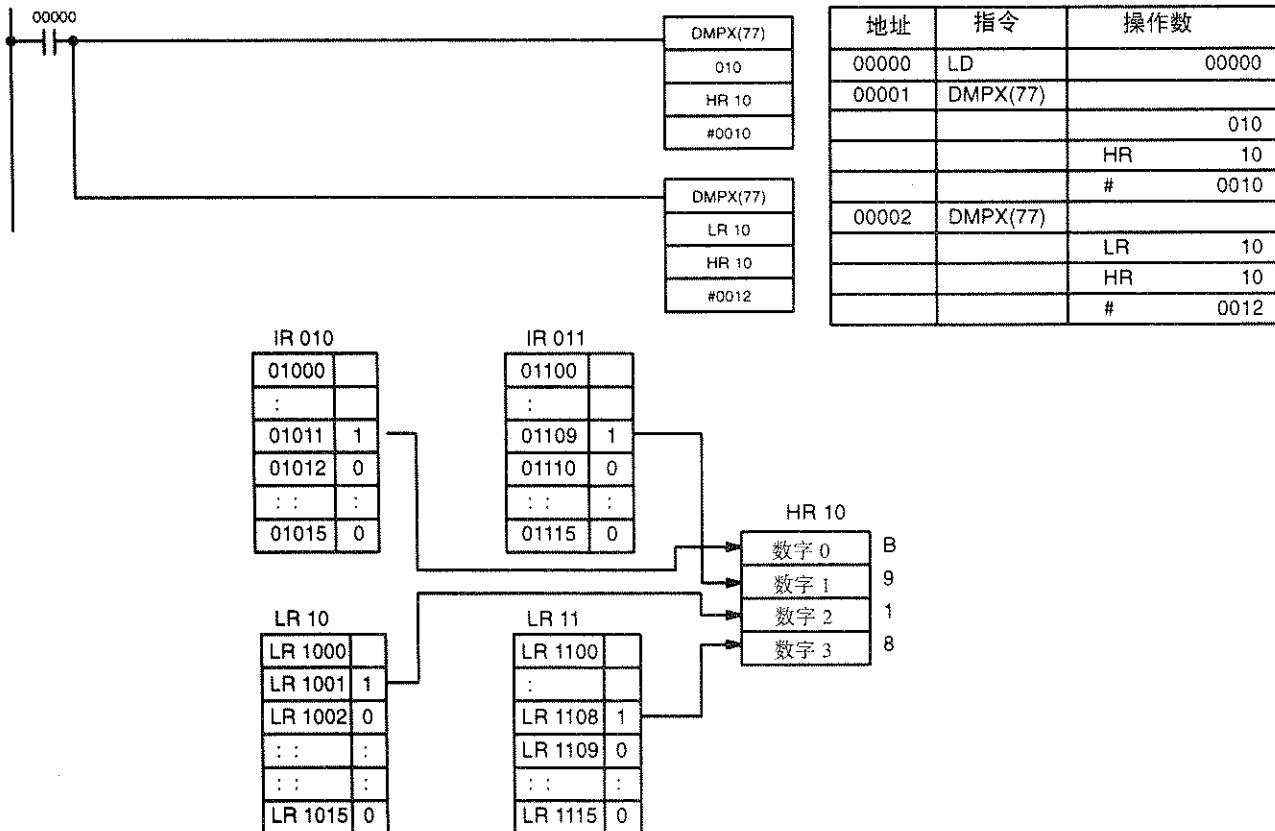
源字的内容为零。

间接寻址的 EM/DM 字不存在。

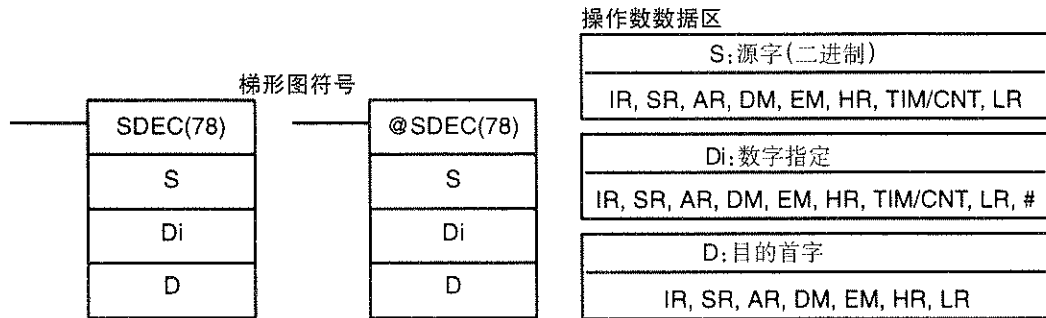
(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区的边界。)

例

当 00000 是 ON 时, 下面的程序将 IR 字 010 与 011 编码至 HR10 的头两个数字, 然后将 LR10 与 LR11 编码至 HR10 的后两个数字。虽然源字的每个位状态没显示出来, 但假定显示的位的状态 1(ON)是该字中的最高 ON 位。



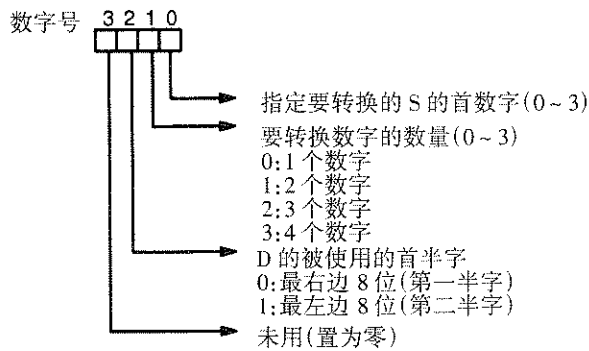
5-20-7 七段译码—SDEC(78)



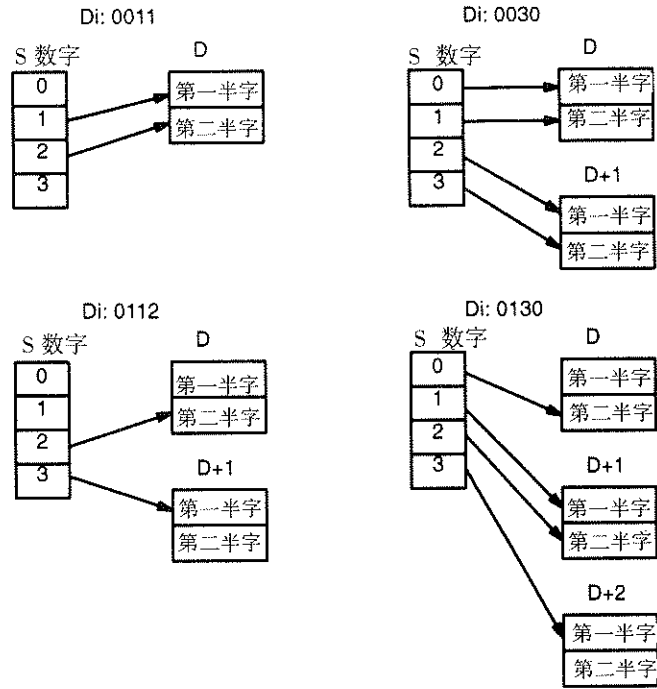
限制 Di 必须在以下给出的数值中。
全部目的字必须在同一数据区。
DM6144 ~ DM6655 不能用于 D。

说明 当执行条件为 OFF 时,SDEC(78)不执行。当执行条件是 ON 时,SDEC(78)将 S 中指定的数字转换为等值的 8 位、7 段显示码,并将其置于以 D 开头的目的字。任何或全部 S 中的数字可以从指定的首数字开始顺序转换。首数字、要转换数字的数目及 D 的哪一半用来接收首个 7 段显示码,(最右边或最左边 8 位)将在 Di 中指定。若指定了多个数字,它们将从指定的 D 的半字开始顺序放置,每个要求两个数字。若指定的数字多于 S 中剩余的数字(从指定首数字算起),多下来的数字将使用 S 的起始处。

指定 Di 的数字设置如下。

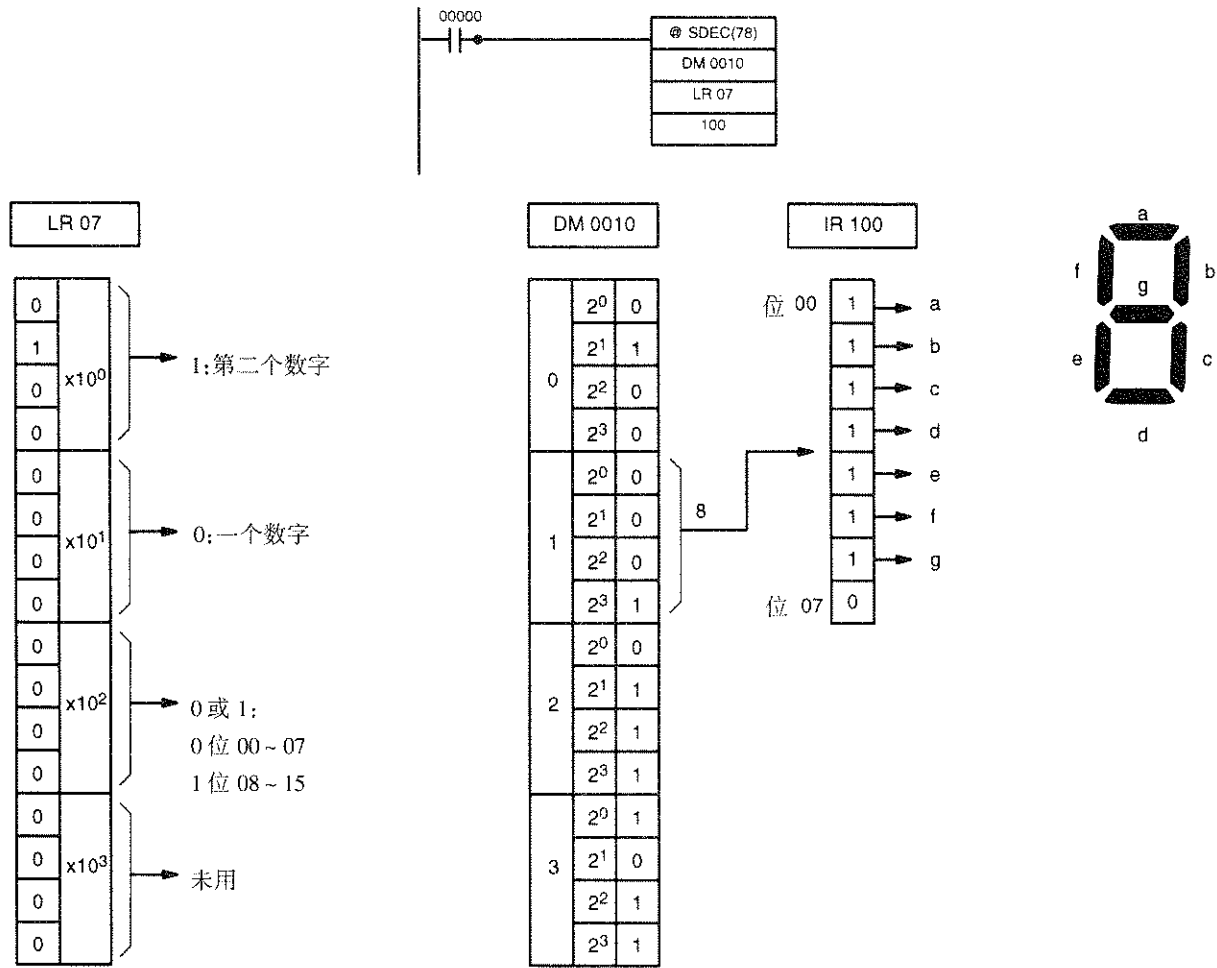


D_i 值及产生的 4 位二进制转换为 7 段显示的一些例子显示如下：



例

下述例子显示产生一个 8 的数据，小写字母表示哪一位对应 7 段显示的哪一段，接着的表格说明了所有 16 进制数字的源数据及转换码。

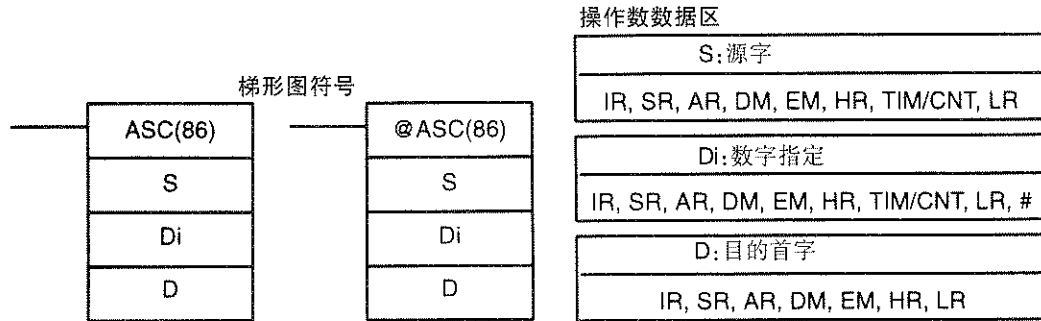


| 数字 | 源数据 | | | | 转换码(段) | | | | | | | | 显示 |
|----|-----|---|---|---|--------|---|---|---|---|---|---|---|----|
| | 位 | | | | - | g | f | e | d | c | b | a | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| A | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| B | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| C | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| D | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| E | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| F | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

标志

ER: 不正确的数字指定, 或目的数据区超出。
 · 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 数据区边界。)

5-20-8 ASCII 转换——ASC(86)



限制

Di 必须在以下指定的值范围内。
 所有目的字必须在同一数据区。
 DM6144 ~ DM6655 不能用于 D。

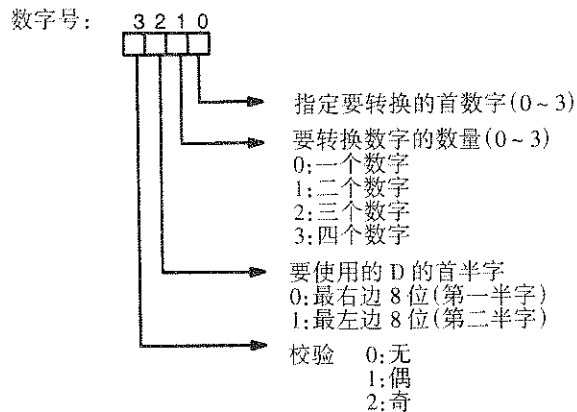
说明

当执行条件为 OFF 时, ASC(86) 不执行。当执行条件为 ON 时, ASC(86) 将 S 中指定的数字转换为等值的 8 位 ASCII 码, 并将其置于以 D 开始的字中。
 在 S 中的任何或所有数字可从指定的首数字起顺序转换。首数字、要转换数字的数量及接收首个 ASCII 码的 D 的半字(最右或最左 8 位)在 Di 中指定。
 若指定了多个数字, 它们将从指定的 D 的半字开始顺序放置, 每个字放两个数字。若指定的数字多于 S 中的剩余数字(从指定的首数字算起), 多下来的数字将使用 S 的起始处。

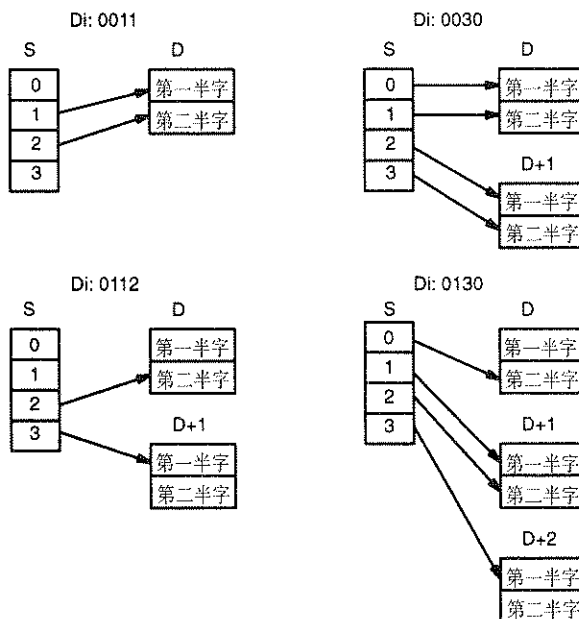
注

参考附录 H ASCII 字符表。
 Di 的数字设置如下所示。

数字指定



Di 值及其生成的 4 位二进制到 8 位 ASCII 转换的一些例子如下所示：



校验

各 ASCII 字符 (2 个数字) 的最左边位可以自动调整作为奇校验或偶校验, 若不指定校验, 最左位将总为零。

当指定为偶校验, 最左位将调整为使得所有 ON 位数量是偶数。例如, 当设定为偶校验, ASCII“31”(00110001) 将为“B1”(10110001; 校验位置 ON 以生成 ON 位数量的偶值); ASCII“36”(00110110) 将为“36”(00110110; 校验位置 OFF, 因为 ON 位数量值已经是偶数), 校验位的状态不影响 ASCII 码的意义。

当指定为奇校验, 各 ASCII 字符最左位将调整为使得 ON 位数量值为奇数。

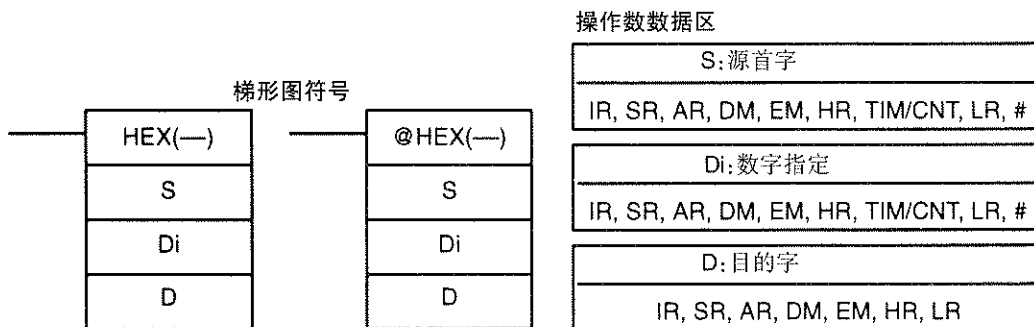
标志

ER: 不正确的数字指定, 或已超出目标数据区。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

5-20-9 ASCII 转换为十六进制——HEX(—)



限制

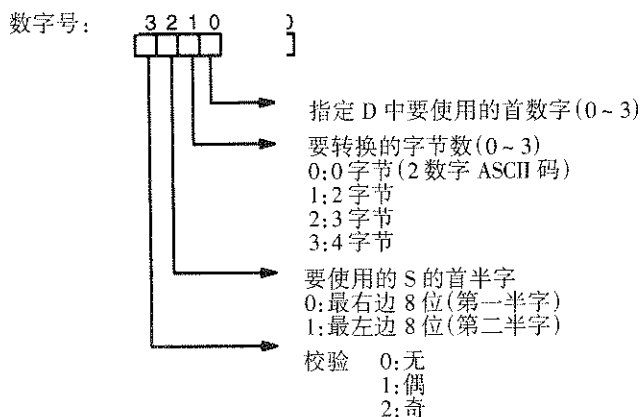
Di 必须在如下指定的范围。
 所有源字必须在同一数据区。
 源字的字节中必须包含等于 16 进制值的 ASCII 码, 例如, 30~39(0~9), 41~46(A~F)。
 DM6144~DM6655 不能用于 D。

说明

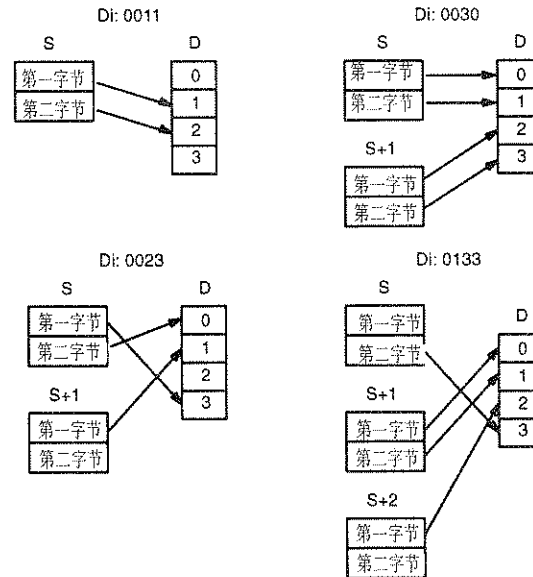
当执行条件为 OFF 时, HEX(一) 不执行。当执行条件为 ON 时, HEX(一) 将指定源字中 ASCII 码的字节转换为等值的 16 进制值并将其置于 D。可进行以指定的 S 的首字节开始最多 4 个 ASCII 码的转换, 转换后的 16 进制值按指定的数字顺序放置到 D。首字节(最右或最左 8 位), 要转换的字节数及接收 16 进制数首值的 D 的数字将在 Di 中指定。若指定了多个字节, 它们将由指定的 S 的半字开始顺序转换, 若需要, 还可连续到 S+1 和 S+2。
 若指定了多于 D 中剩余的数字(从指定首数字算起), 这多下来的数字将使用 D 的起始处, D 中不接收转换值的数字将不改变。

数字指定

Di 的数字按如下设置。



Di 值及其生成的 8 位 ASCII 到 4 位 16 进制转换的一些例子如下所示：



ASCII 码表

下面显示了转换前的 ASCII 码及转换后的 16 进制值。参考附录 H ASCII 字符表。

| ASCII 码 | 源 数 据 | | | | | | | | 转 换 后 数 据 | | | | |
|---------|-------------|---|---|---|---|---|---|---|-----------|---|---|---|---|
| | 位 状 态 (见 注) | | | | | | | | 数 字 | 位 | | | |
| 30 | * | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 31 | * | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 32 | * | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 0 |
| 33 | * | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 1 | 1 |
| 34 | * | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 4 | 0 | 1 | 0 | 0 |
| 35 | * | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 5 | 0 | 1 | 0 | 1 |
| 36 | * | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 6 | 0 | 1 | 1 | 0 |
| 37 | * | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 7 | 0 | 1 | 1 | 1 |
| 38 | * | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 8 | 1 | 0 | 0 | 0 |
| 39 | * | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 9 | 1 | 0 | 0 | 1 |
| 41 | * | 1 | 0 | 1 | 0 | 0 | 0 | 1 | A | 1 | 0 | 1 | 0 |
| 42 | * | 1 | 0 | 1 | 0 | 0 | 1 | 0 | B | 1 | 0 | 1 | 1 |
| 43 | * | 1 | 0 | 1 | 0 | 0 | 1 | 1 | C | 1 | 1 | 0 | 0 |
| 44 | * | 1 | 0 | 1 | 0 | 1 | 0 | 0 | D | 1 | 1 | 0 | 1 |
| 45 | * | 1 | 0 | 1 | 0 | 1 | 0 | 1 | E | 1 | 1 | 1 | 0 |
| 46 | * | 1 | 0 | 1 | 0 | 1 | 1 | 0 | F | 1 | 1 | 1 | 1 |

注 各个 ASCII 代码最左边位用于奇偶校验。

校验

各个 ASCII 字符(2 个数字)的最左位用于奇偶校验自动调整。

若无检验,最左边位总是为零。若为奇或偶检验,每个 ASCII 字符的最左位将调整以使得所有 ON 位之和为奇数或偶数。

若 S 中的 ASCII 码的校验与 Di 指定的校验不符,ER 标志(SR25503)将置 ON 并且该指令将不执行。

标志

ER: 不正确的数字指定, 或超出目的数据区。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界)。

例

在下述例子中, LR10 的第二字节和 LR11 的第一字节转换为 16 进制值, 这些值写到 IRO10 的第一和第二字节。



| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00000 |
| 00001 | @HEX(—) | |
| | | LR 10 |
| | | HR 10 |
| | | 010 |

HR 10

| | | | |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
|---|---|---|---|

LR 12

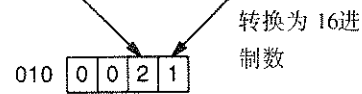
| | | | |
|---|---|---|---|
| 3 | 5 | 3 | 4 |
|---|---|---|---|

LR 11

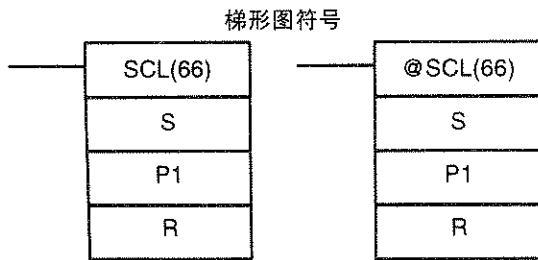
| | | | |
|---|---|---|---|
| 4 | 2 | 3 | 2 |
|---|---|---|---|

LR 10

| | | | |
|---|---|---|---|
| 3 | 1 | 3 | 0 |
|---|---|---|---|



5-20-10 标度 - SCL(66)



操作数数据区

| |
|--|
| S: 源字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| P1: 首参数字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

S 必须是 BCD。

P1 ~ P1 + 3 必须在同一数据区。

DM6144 ~ DM6655 不能用于 P1 ~ P1 + 3 或 R。

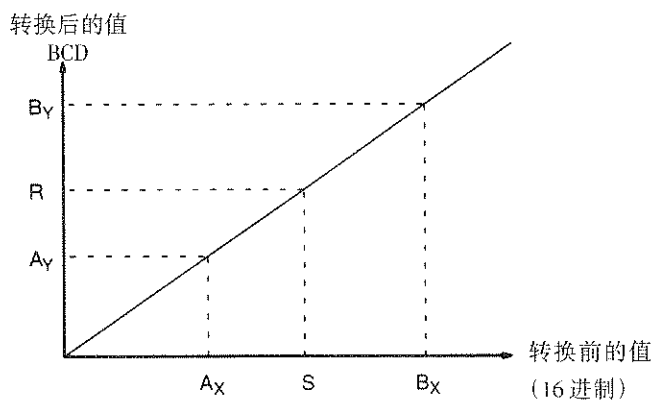
说明

SCL(66) 用来将 4 位数 16 进制值线性地转换为 4 位数 BCD 值, 不同于 BCD (24) 的将 4 位数 16 进制等值地转换为 BCD ($S_{hex} \rightarrow S_{bcd}$), SCL(66) 能按指定的线性关系转换 16 进制值。转换线由在参数字 P1 ~ P1 + 3 中指定的两个点来定义。当执行条件为 OFF, SCL(66) 不执行。当执行条件为 ON, SCL(66) 将 S 中的 4 位数 16 进制值, 按点 (P1, P1 + 1) 和 (P1 + 2, P1 + 3) 定义的线性关系转换为 4 位数 BCD 值, 并将结果置于 R。结果舍入到最相近的整数。若结果小于 0000, 则将 0000 写到 R, 若结果大于 9999, 则将 9999 写到 R。

下表说明参数数字的功能和范围:

| 参数 | 功能 | 范围 | 注释 |
|--------|-----------------------------|-------------|----------------------|
| P1 | BCD 点 # 1(A _Y) | 0000 ~ 9999 | - - - |
| P1 + 1 | 16 进制点 # 1(A _X) | 0000 ~ FFFF | 不要设成 P1 + 1 = P1 + 3 |
| P1 + 2 | BCD 点 # 2(B _Y) | 0000 ~ 9999 | - - - |
| P1 + 3 | 16 进制点 # 2(B _X) | 0000 ~ FFFF | 不要设成 P1 + 3 = P1 + 1 |

下图说明根据由点(A_Y, A_X)和(B_Y, B_X)定义的线将源字 S 转换到 D。



首先把所有的值转换为 BCD, 然后利用以下的公式计算结果:

$$\text{结果} = B_Y - [(B_Y - A_Y) / (B_X - A_X) \times (B_X - S)]$$

标志

ER: P1 + 1 的值等于 P1 + 3 的值。

间接寻址的 EM/DM 字不存在。

(* EM / * DM 字的内容不是 BCD, 或者超出 EM/DM 区边界。)

P1 和 P1 + 3 不在同一数据区, 或其它设置出错。

EQ: 当结果 R 为 0000 时置 ON。

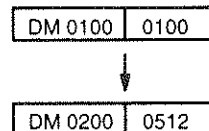
例

在下面的例子中, 当 00000 置 ON, 在 DM0100(# 0100) 中的 BCD 源数据, 依据在 DM0150 ~ DM0153 中的参数转换为 16 进制值, 其结果(# 0512)被写到 DM0200。

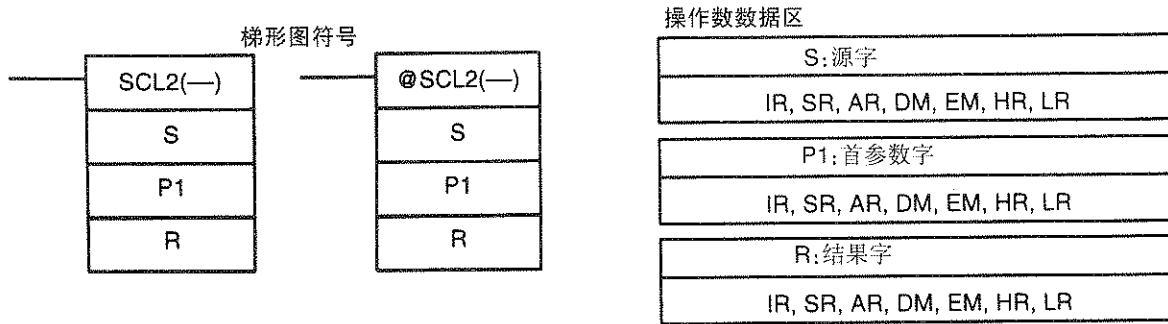


| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00000 |
| 00001 | @SCL(66) | |
| | | DM 0100 |
| | | DM 0150 |
| | | DM 0200 |

| | |
|---------|------|
| DM 0150 | 0010 |
| DM 0151 | 0005 |
| DM 0152 | 0050 |
| DM 0153 | 0019 |



5-20-11 有符号二进制到 BCD 标度 - SCL2(—)



限制

S 必须是 BCD。
 P1 ~ P1 + 2 必须在同一数据区。
 DM6144 ~ DM6655 不能用于 R。

说明

SCL2(—) 用来将 4 位数有符号 16 进制值线性地转换为 4 位数 BCD 值, 不同于 BCD(24) 等值地将 4 位数 16 进制转换为等值的 4 位数 BCD($S_{hex} \sim S_{BCD}$), SCL2(—) 可根据指定的线性关系转换有符号 16 进制值。转换线由参数字 P1 至 P1 + 2 中规定的 X 轴截距和线斜率来定义。

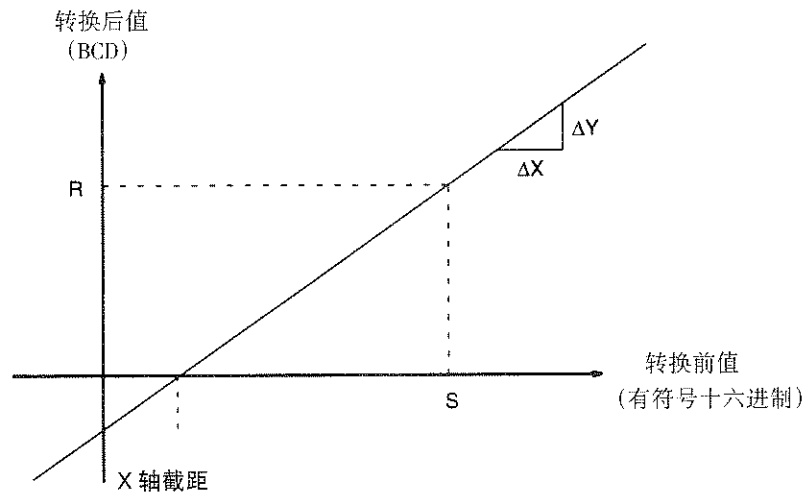
当执行条件为 OFF 时, SCL2(—) 不执行, 当执行条件为 ON 时, SCL2(—) 将 S 中的有符号 4 位数 16 进制值转换为 4 位数 BCD 值, 转换依据 X 轴截距 (P1, 0) 和斜率 (P1 + 2 ÷ P1 + 1) 定义的线进行, 并将结果置于 R。结果舍入为最接近的整数。

若结果为负数, 置 CY 为 1。若结果小于 -9999, 将 -9999 写到 R。若结果大于 9999, 将 9999 写到 R。

下表说明参数字的功能和范围。

| 参数 | 功能 | 范围 |
|--------|----------------------|--------------------------------|
| P1 | X轴截距 (有符号16进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |
| P1 + 1 | ΔX (有符号16进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |
| P1 + 2 | ΔY (BCD) | 0000 ~ 9999 |

下图显示源字 S 根据点 (P1, 0) 和斜率 $\Delta Y/\Delta X$ 定义的线转换到 R。



结果的获得,首先将所有有符号 16 进制值转换为 BCD 值,然后,依据下式算出。

$$R = \frac{\Delta Y}{\Delta X} \times (S - P1)$$

标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD,或已超出 EM/DM 区边界)。

P1 和 P1+2 不在同一数据区,或其他设置错误。

CY: 当结果 R 为负值时置 ON。

EQ: 当结果 R 为 0000 时置 ON。

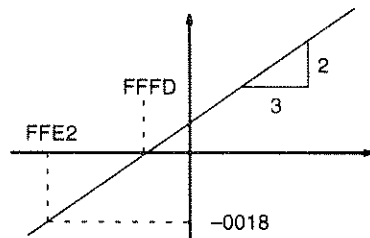
例

下面的例子中,当 05000 为 ON 时,在 001(# FFE2)中的有符号二进制源数据按在 DM0000 ~ DM0002 中的参数转换为 BCD, 其结果 (# 0018) 随后写入 LR00,并且由于结果是负值 CY 置 ON。



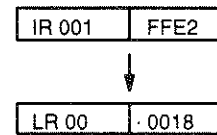
| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 05000 |
| 00001 | @SCL2(-) | |
| | | 001 |
| | | DM 0000 |
| | | LR 00 |

| | |
|---------|------|
| DM 0000 | FFFD |
| DM 0001 | 0003 |
| DM 0002 | 0002 |



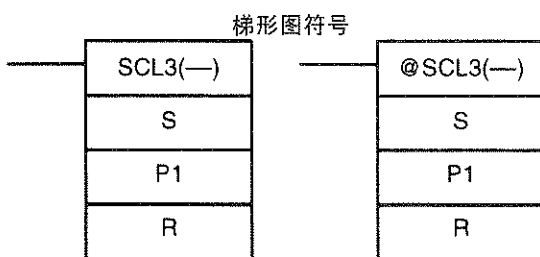
$$R = \frac{0002}{0003} \times (FFE2 - FFFD)$$

$$= \frac{2}{3} \times (-1B) = -18$$



CY=1
↑
CY 标志变 ON, 因为转换结果是负值。

5-20-12 BCD 到有符号二进制标度 - SCL3(-)



| 操作数数据区 | |
|----------------------------|--|
| S: 源字 | |
| IR, SR, AR, DM, EM, HR, LR | |
| P1: 首参数字 | |
| IR, SR, AR, DM, EM, HR, LR | |
| R: 结果字 | |
| IR, SR, AR, DM, EM, HR, LR | |

限制

P1+1 必须是 BCD。

P1 ~ P1+4 必须在同一数据区。

DM6144 ~ DM6655 不能用于 R。

说明

SCL3(一)用来将 4 位数 BCD 值转换为 4 位数有符号的 16 进制值。SCL3(一)依据指定的线性关系转换 BCD 值,转换线由指定在参数字 P1~P1+2 中的 Y 轴截距和线斜率来定义。

当执行条件为 OFF 时,SCL3(一)不执行。当执行条件为 ON 时,SCL3(一)依据由 Y 轴截距(0,P1)和斜率(P1+2÷P1+1)定义的线将 S 中的 4 位数 BCD 值转换为 4 位数有符号 16 进制值,并将结果置于 R。结果将舍入到最相近的整数。

S 的内容可为 0000~9999,若 CY=1 则将 S 视为负数,所以 S 的范围实际上是 -9999~9999。一定要使用 STC(40) 或 CLC(41) 将 CY 设置为所需要的符号。

参数字 P1+3 和 P1+4 规定结果的上限和下限。若结果大于 P1+3 中的上限,上限被写到 R。若结果小于 P1+4 中的下限,下限被写到 R。

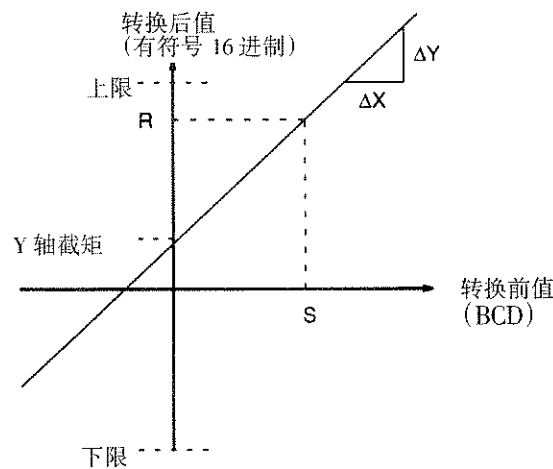
注 对 12 位模拟量输入单元,上下限分别为 07FF 和 F800。

下表说明参数字的功能和范围。

| 参 数 | 功 能 | 范 围 |
|------|------------------|--------------------------------|
| P1 | X 轴截距(有符号 16 进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |
| P1+1 | ΔX (BCD) | 0001 ~ 9999 |
| P1+2 | ΔY (有符号 16 进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |
| P1+3 | 上限(有符号 16 进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |
| P1+4 | 下限(有符号 16 进制) | 8000 ~ 7FFF (-32,768 ~ 32,767) |

注 不要将第二个字(P1+1)的 ΔX 设置为 0000(4 位数 BCD),P1+1 用作除数,被 0000 除就不能得到正确的转换,如果使用十六进制值也不能得到正确的结果。P1+1 必须为 0001 到 9999 的 BCD 数据。

下图说明源字 S 根据由点(0,P1)和斜率 ΔY/ΔX 定义的线转换到 R。



该结果的计算是:首先将所有 BCD 值转换为有符号的二进制值,然后利用以下算式

$$R = \left(\frac{\Delta Y}{\Delta X} \times S\right) + P1$$

标志

ER:间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD,或已超出 EM/DM 区边界)。

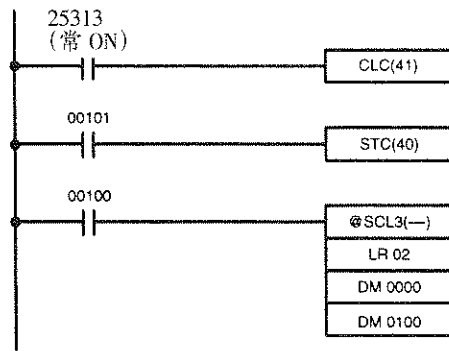
S 的内容不是 BCD。

CY: SCL3(—)不改变 CY。(CY 代表转换前 S 的符号。)

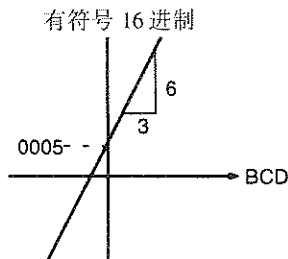
EQ: 结果 R 为 0000 时置 ON。

例

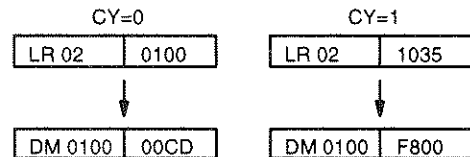
在下例中,00101 的状态决定了源字中 BCD 值的符号。如果 00101 为 ON,那么源字为负值。当 00100 为 ON 时,在 LR02 中的 BCD 源数据根据在 DM0000~DM0004 中的参数转换为有符号的二进制值。结果被写入到 DM0100(在第二个转换中,有符号二进制值为 -1035,小于在 DM0004 中规定的下限,所以下限写到 DM0100)。



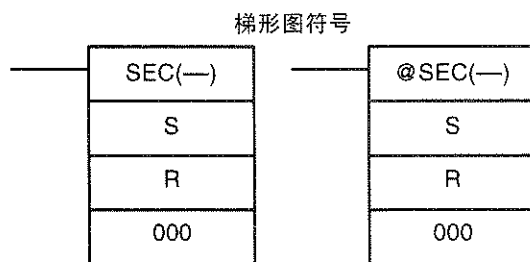
| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 25313 |
| 00001 | CLC(41) | |
| 00002 | LD | 00101 |
| 00101 | STC(40) | |
| 00004 | LD | 00100 |
| 00005 | SCL3(—) | |
| | | LR 02 |
| | | DM 0000 |
| | | DM 0100 |



| | |
|---------|------|
| DM 0000 | 0005 |
| DM 0001 | 0003 |
| DM 0002 | 0006 |
| DM 0003 | 07FF |
| DM 0004 | F800 |



5-20-13 时转换为秒—SEC(—)



操作数数据区

| |
|-------------------------------------|
| S:源首字(BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R:结果首字(BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| 000:不起作用 |
| 000 |

限制

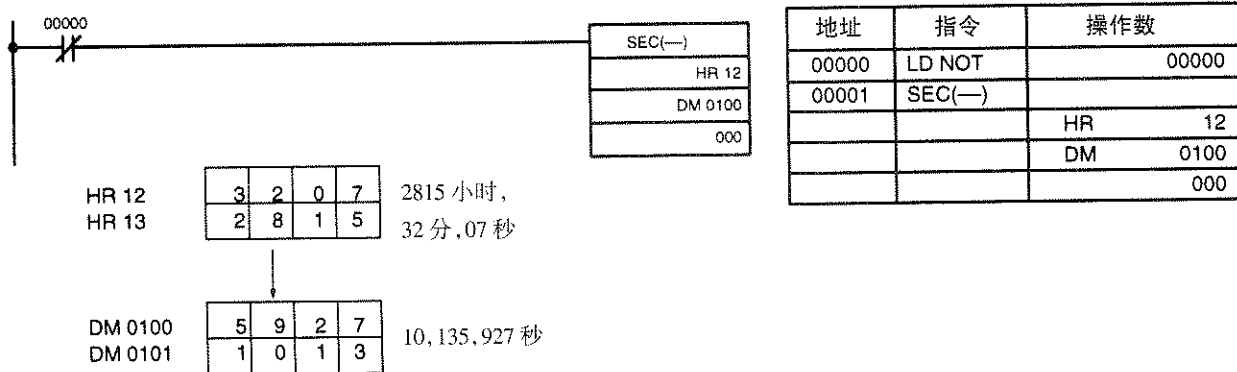
S 和 S + 1 必须在同一数据区,R 和 R + 1 必须在同一数据区。S 和 S + 1 必须是 BCD 并且必须是适当的时/分/秒格式。

DM6143 ~ DM6655 不能用于 R。

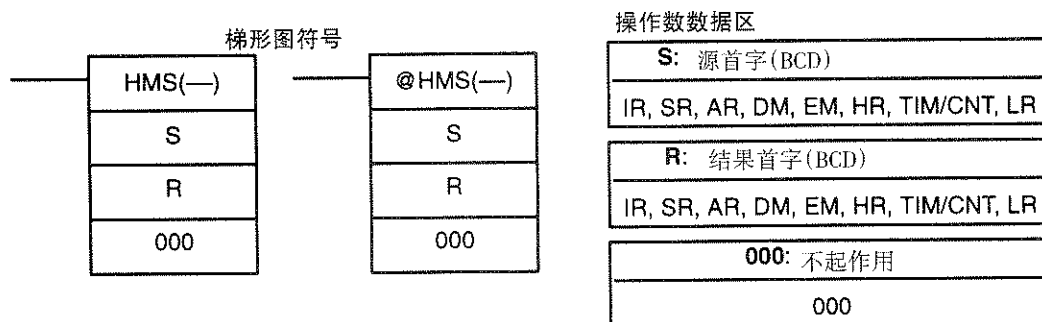
说明 SEC(—)用来把以时/分/秒表示的时间转换为等同的秒值。
 在源数据中,秒由 S 中的位 00~07 指定,分由 S 的位 08~15 指定。小时由 S+1 指定。所以最大值为 9999 小时,59 分,59 秒。
 结果输出到 R 和 R+1,所能获得的最大值是 35,999,999 秒

标志 ER:S 和 S+1 或 R 和 R+1 不在同一数据区。
 S 和/或 S+1 不是 BCD。
 秒和/或分的值超出 59。
 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD,或已超出 EM/DM 区边界)。
 EQ:结果为零时置 ON

例 当 00000 为 OFF(即当执行条件为 ON), 下面的指令将 HR12 和 HR13 给出的时,分,秒转换为秒,并将结果存到 DM0100 和 DM0101。



5-20-14 秒转换为时 - HMS(—)



限制 S 和 S+1 必须在同一数据区,R 和 R+1 必须在同一数据区。S 和 S+1 必须是 BCD 并且必须为 0~35,999,999 秒。
 DM6143~DM6655 不能用于 R。

说明 HMS(—)用来将以秒表示的时间转换为相等的以时/分/秒表示的时间。
 在 S 和 S+1 中指定的秒值转换为时/分/秒并置于 R 和 R+1。

在结果中,秒值置于 K 的位 00—07,分值置于 R 的位 08—15,时值置于 R + 1 中。最大值是 9,999 小时 59 分,59 秒。

标志

ER: S 和 S + 1 或 R 和 R + 1 不在同一数据区。

S 和/或 S + 1 不是 BCD 或者超出 36,000,000 秒。

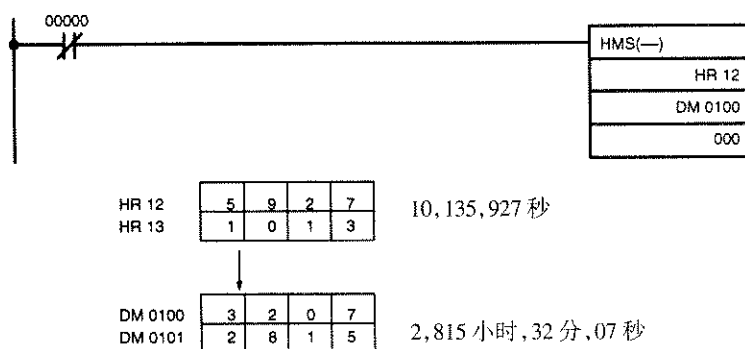
间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)

EQ: 当结果为零时置 N。

例

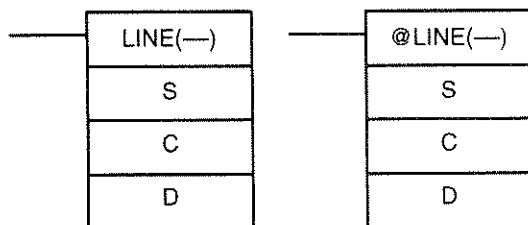
当 00000 为 OFF(即执行条件为 ON), 下述指令将在 HR12 和 HR13 中指定的秒,转换为时、分、秒,并将结果存在 DM0100 和 DM0101。



| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD NOT | 00000 |
| 00001 | HMS(-) | |
| | | HR 12 |
| | | DM 0100 |
| | | 0000 |

5-20-15 列-行转换 - LINE(-)

梯形图符号



操作数数据区

| |
|--|
| S: 16 个源字组的首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| C: 列位指定 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| D: 目的字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |

限制

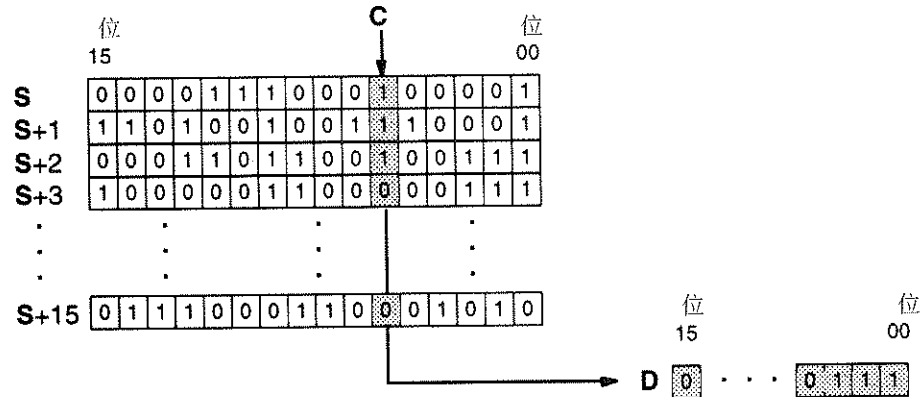
S 和 S + 15 必须在同一数据区。

C 必须是 # 0000 和 # 0015 之间的 BCD。

DM6144—DM6655 不能用于 D。

说明

当执行条件为 OFF 时,LINE(—)不执行。当执行条件为 ON 时,LINE(—)将十六字组(S~S+15)中列的位 C 复制到字 D 的 16 个位中(00—15)。



标志

ER: 列位指定 C 不是 BCD 值,或指定了不存在的位。(即,指定位必须在 00~15 之间)

间接寻址的 EM/DM 不存在。

(*EM/*DM 字内容不是 BCD,或超出 EM/DM 区边界。)

S 和 S+15 不在同一数据区。

EQ: 当 D 为零时置 ON,否则置 OFF。

例

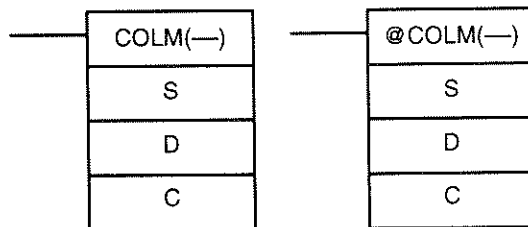
下例表明如何使用 LINE(—)将字组(IR100~IR115)的位列 07 移至 DM0100。



| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | LINE(—) | |
| | | 100 |
| | | # 0007 |
| | | DM 0100 |

5-20-16 行列转换—COLM(—)

梯形图符号



操作数数据区

| |
|--------------------------------|
| S:源字 |
| IR,SR,AR,DM,EM,HR,TIM/CNT,LR |
| D:目的字组首字 |
| IR,SR,AR,DM,EM,HR,TIM/CNT,LR |
| C:列位指定(BCD) |
| IR,SR,AR,DM,EM,HR,TIM/CNT,LR,# |

限制

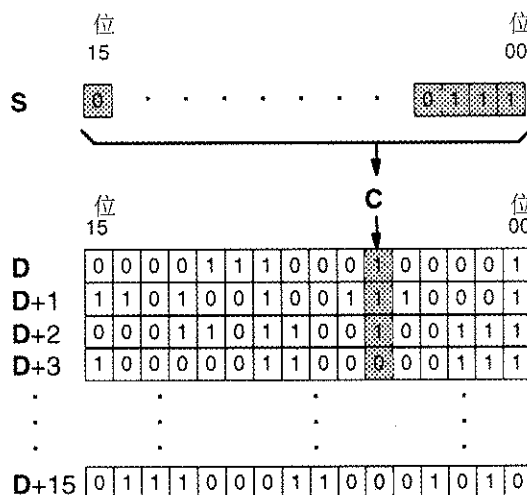
D 和 D+15 必须在同一数据区。

DM6129~DM6655 不能用于 D。

C 必须是 #0000 和 #0015 之间的 BCD 值。

说明

当执行条件为 OFF 时, COLM(—) 不执行。当执行条件为 ON 时, COLM(—) 将字 S 的 16 个位(00~15)复制到 16 字组(D~D+15)中 C 列的位中。



标志

ER: 位指定 C 不是 BCD 值, 或指定了不存在的位 (指定位必须是 00~15 之间的值)。

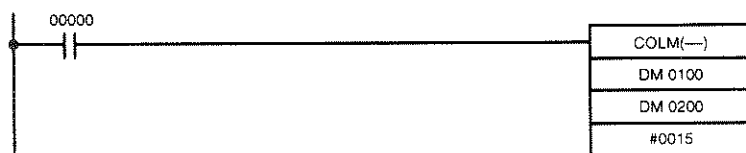
间接寻址的 EM/DM 字不存在。(* EM/ * DM 字内容不是 BCD, 或超出 EM/DM 区边界。)

D 和 D+15 不在同一数据区。

EQ: 若 S 为零则置 ON, 否则置 OFF。

例

下例说明如何使用 COLM(—) 将字 DM0100(00~15) 的内容移到字组(DM0200~DM0215)的位列 15。

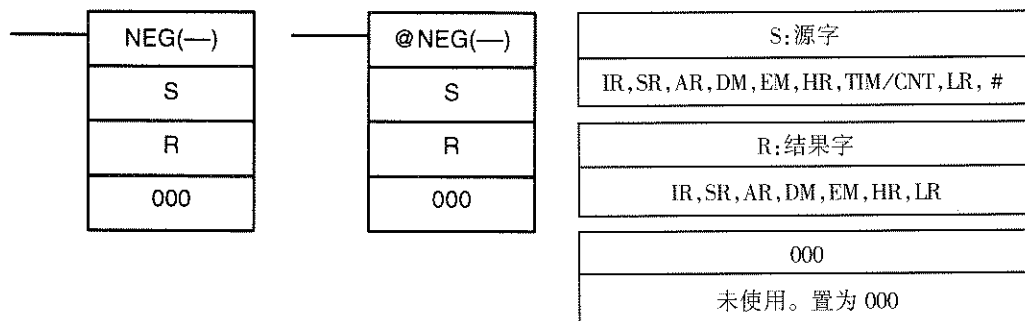


| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | COLM(—) | |
| | | DM 0100 |
| | | DM 0200 |
| | | # 0015 |

5-20-17 求 2 进制补码 NEG(—)

梯形图符号

操作数数据区



限制

DM6144 ~ DM6655 不能用于 R。

说明 将源字(S)的 4 位数 16 进制内容转换为它的 2 进制补码,并将结果输出到结果字(R)。

该运算的效果就象从 0000 减去 S 并输出结果到 R; 对于负的有符号二进制数,该运算为计算其绝对值。

若 S 的内容是 0000,指令执行后 R 的内容还是 0000,并且 EQ 标志(SR25506)将置 ON。

若 S 的内容是 8000,指令执行后 R 的内容亦是 8000,并且 UF 标志(SR25405)将置 ON。

注: 更详细的情况请参考 1-7 用有符号二进制数据计算。

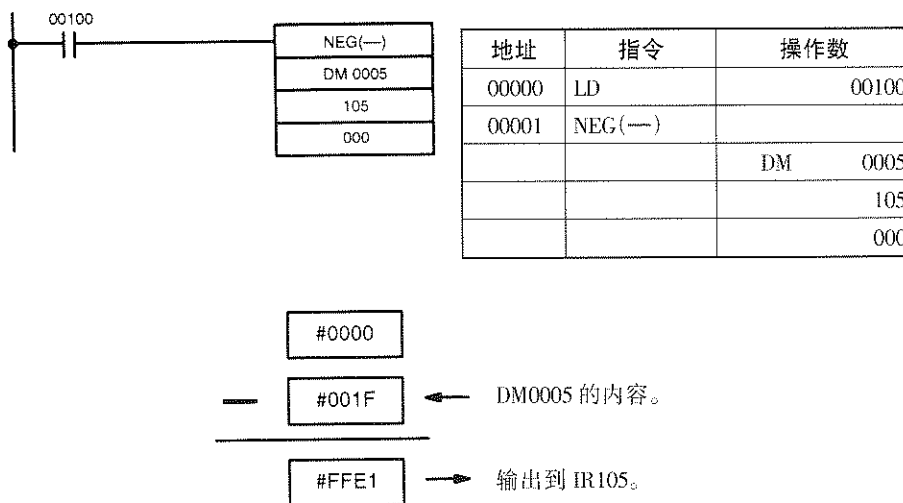
标志 ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)

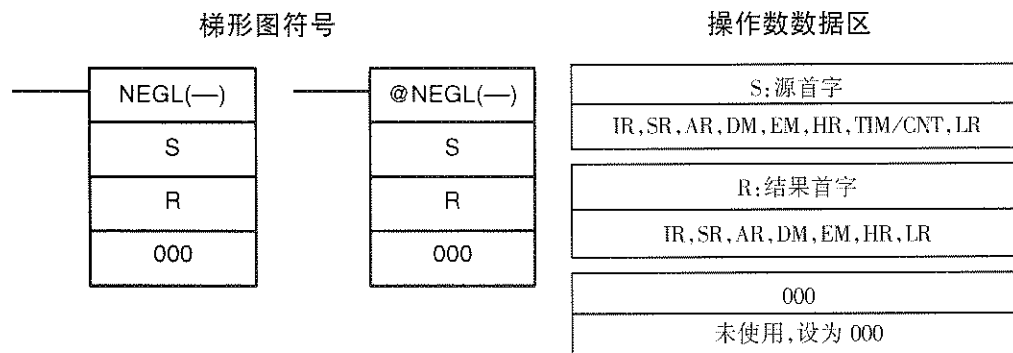
EQ: 执行后当 R 的内容为 0 置 ON, 否则置 OFF。

UF: 执行后当 S 的值为 8000 时置 ON; 否则置 OFF。

例 下例说明如何使用 NEG(—) 找出 DM0005 内容的 2 进制补码, 并将结果送 IR105。



5-20-18 求双字 2 进制补码 - NEGL(—)



限制

DM6143 ~ DM6655 不能用于 R。

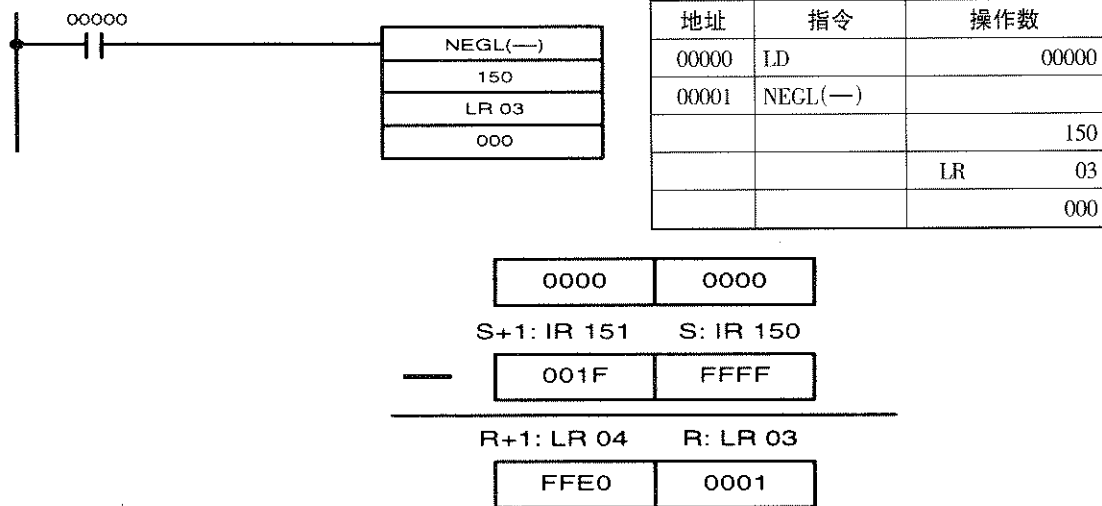
S 和 S + 1 必须在同一数据区, 同样地 R 和 R + 1 也必须在同一数据区。

说明 将源字 (S 和 S + 1) 的 8 位数 16 进制内容转换为它的 2 进制补码, 并将结果输出到结果字 (R 和 R + 1)。该运算的效果就象从 00000000 减去 S 和 S + 1 中的 8 位数内容, 并将结果输出到 R 和 R + 1; 若数据为负的有符号二进制数, 为计算其绝对值。
 若 S 的内容是 00000000, 指令执行后 R 的内容还将是 00000000, 并且 EQ (SR25506) 将置 ON。
 若 S 的内容是 80000000, 那么执行后 R 的内容还是 80000000, 并且 UF (SR25405) 将置 ON。

注 详细情况参考 1—7 用有符号二进制数据计算。

标志 ER: 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD, 或者超出 EM/DM 区边界。)
 EQ: 若执行后, R + 1, R 的内容为零则置 ON, 否则置 OFF。
 UF: 若 S + 1, S 的内容是 80000000 则置 ON, 否则置 OFF。

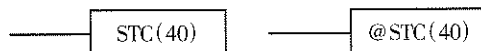
例 下例说明如何使用 NEGL(—) 找出在 IR151, IR150 中的值 (001FFFF) 的二进制补码, 并输出结果到 HR04, HR03。



5-21 BCD 计算指令

5-21-1 设置进位 - STC(40)

梯形图符号



当执行条件为 OFF 时, STC(40) 不执行。当执行条件为 ON 时, STC(40) 将 CY (SR25504) 置 ON。

注: 关于影响 CY 的指令表参考附录 B 错误和算术标志操作。

5-21-2 清除进位 - CLC(41)

梯形图符号



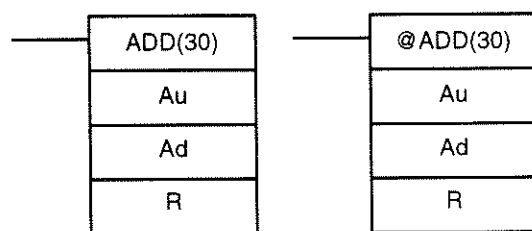
当执行条件为 OFF 时, CLC(41) 不执行。当执行条件为 ON 时, CLC(41) 将 CY (SR25504) 置为 OFF。

清除进位是用来把 CY (SR25504) 复位(置 OFF)为“0”。

注 关于影响 CY 的指令表参考附录 B 错误和算术标志操作。

5-21-3 BCD 加 - ADD(30)

梯形图符号



操作数数据区

| |
|--|
| Au: 被加字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Ad: 加字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 结果字 |
| IR, SR, AR, DM, EM, HR, LR |

限制
说明

DM6144 ~ DM6655 不能用于 R。

当执行条件为 OFF 时, ADD(30) 不执行。当执行条件为 ON 时, ADD(30) 将 Au 和 Ad 以及 CY 的内容相加, 并将结果置于 R。若结果大于 9999 将把 CY 置 ON。

$$\boxed{\text{Au}} + \boxed{\text{Ad}} + \boxed{\text{CY}} \rightarrow \boxed{\text{CY}} \quad \boxed{\text{R}}$$

标志

ER: Au 和 Ad 不是 BCD 值。

间接寻址的 EM/DM 字不存在。

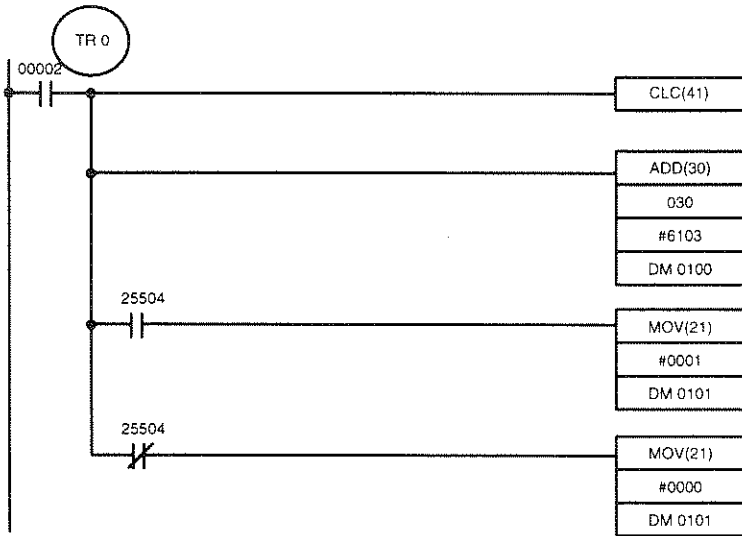
(*EM/*DM 字的内容不是 BCD, 或者超出 EM/DM 区边界。)

CY: 当结果有进位时置 ON。

EQ: 结果为 0 时置为 ON。

例

在下面的程序中,若 00002 为 ON, CLC(41) 将清除 CY, 将 IR030 的内容与一个常数 (6103) 相加, 结果置于 DM0100, 然后根据 CY(25504) 的状态将全 0 或 0001 移到 DM0101。这就保证了来自最后一个数字的任何进位可保存在 R + 1 中, 以便整个结果在以后的处理中可以作为一个 8 位数的数据处理。

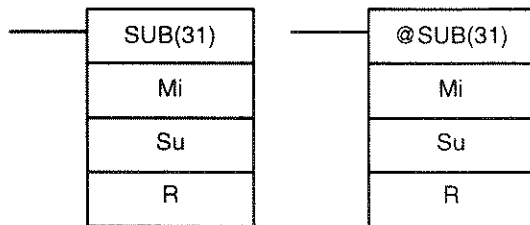


| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00002 |
| 00001 | OUT | TR 0 |
| 00002 | CLC(41) | |
| 00003 | AND(30) | |
| | | 030 |
| | | # 6103 |
| | | DM 0100 |
| 00004 | AND | 25504 |
| 00005 | MOV(21) | |
| | | # 0001 |
| | | DM 0101 |
| 00006 | LD | TR 0 |
| 00007 | AND NOT | 25504 |
| 00008 | MOV(21) | |
| | | # 0000 |
| | | DM 0101 |

虽然可以同时使用二条 ADD(30) 来实现 8 位数 BCD 的相加, 但 ADDL(54) 是专门设计用来实现这个目的。

5-21-4 BCD 减 - SUB(31)

梯形图符号



操作数数据区

| |
|--|
| Mi: 被减字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR # |
| Su: 减字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 结果字 |
| IR, SR, AR, DM, EM, HR, LR |

限制说明

DM6144 ~ DM6655 不能用于 R。

当执行条件为 OFF 时, SUB(31) 不执行。当执行条件为 ON 时, SUB(31) 从 Mi 中减去 Su 和 CY 的内容, 并将结果置于 R。若结果为负, 置位 CY 并将实际结果取 10 进制补码置于 R。要将 10 进制补码转换为实际结果。可用零减去 R 的内容(见下例)。

$$\boxed{Mi} - \boxed{Su} - \boxed{CY} \rightarrow \boxed{CY} \quad \boxed{R}$$

标志

ER: Mi 和/或 Su 不是 BCD。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD, 或超出 DM/EM 区边界。)

CY: 当结果为负数时(即当 M_i 小于 S_u 加 CY)置 ON。

EQ: 结果为零时置 ON。

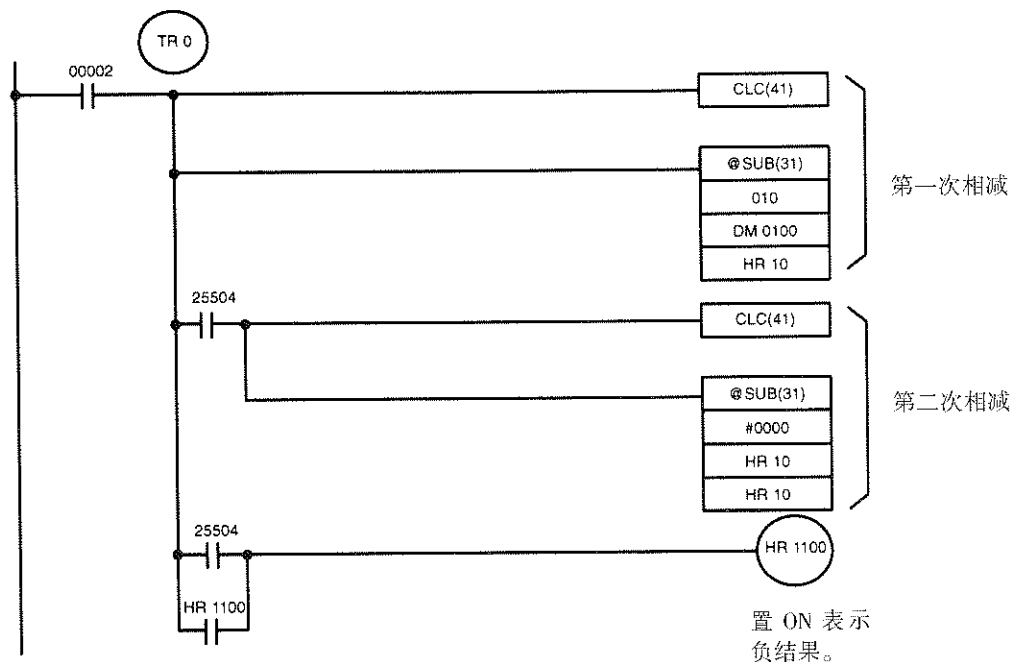
注意 若不需要先前的进位标志,应在执行 SUB(31)以前用 CLC(41)清除。要在执行 SUB(31)之后检查 CY 的状态。若执行 SUB(31)后 CY 为 ON(结果为负数),输出结果为实际答案的 10 进制补码,若要将输出结果转换为实际值,可用 0 减去 R 的值。

例

当 00002 为 ON,下面梯形图程序清除 CY,从 010 的内容减去 DM0100 的内容和 CY,并将结果置 HR10。若 CY 由于执行 SUB(31)而置位,HR10 中的结果被从零减去(注意再次需要 CLC(41)以得到真正的结果),结果重新置回 HR10,HR1100 置 ON 以标志一个负数结果。

若 CY 没被执行 SUB(31)置位,则结果是正数,第二步相减就不必执行,并且 HR1100 不置 ON。HR1100 被编程为自保持位,以便当程序重新扫描时 CY 的状态不会将其置 OFF。

本例中使用了 SUB(31) 的微分形式以使得每当 0002 置 ON 时减运算只执行一次。若需要再次减运算,需要至少在一个循环中将 00002 置 OFF(复位 HR1100),然后再将其置 ON。



| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00002 |
| 00001 | OUT | TR 0 |
| 00002 | CLC(41) | |
| 00003 | @SUB(31) | |
| | | 010 |
| | | DM 0100 |
| | | HR 10 |
| 00004 | AND | 25504 |
| 00005 | CLC(41) | |
| 00006 | @SUB(31) | |
| | | # 0000 |
| | | HR 10 |
| | | HR 10 |
| 00007 | LD | TR 0 |
| 00008 | LD | 25504 |
| 00009 | OR | HR 1100 |
| 00010 | AND LD | |
| 00011 | OUT | HR 1100 |

本例的第一第二次相减将在下面用示例数据 010 和 DM0100 说明。

注 实际的 SUB(31)运算包括从 10,000 与 Mi 之和减去 Su 和 CY。对于正数结果，最左位数被截去，对于负数结果，得到的是 10 进制补码，建立正确答案的过程由如下给出。

第一次相减

```

IR 010   1029
DM0100  - 3452
CY       - 0
-----
HR 10   7577 (1029 + (10000 - 3452))
CY      1   (负数结果)
    
```

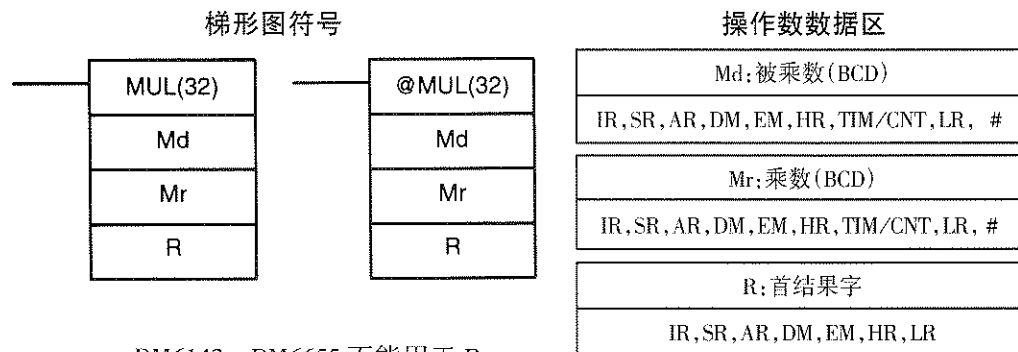
第二次相减

```

          0000
HR 10   - 7577
CY       - 0
-----
HR 10   2423 (0000 + (10000 - 7577))
CY      1   (负数结果)
    
```

在上面的情形中，程序将把 HR1100 置 ON 以表示保持在 HR10 中的值是负数。

5-21-5 BCD 乘 - MUL(32)

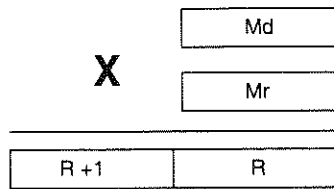


限制

DM6143 ~ DM6655 不能用于 R。

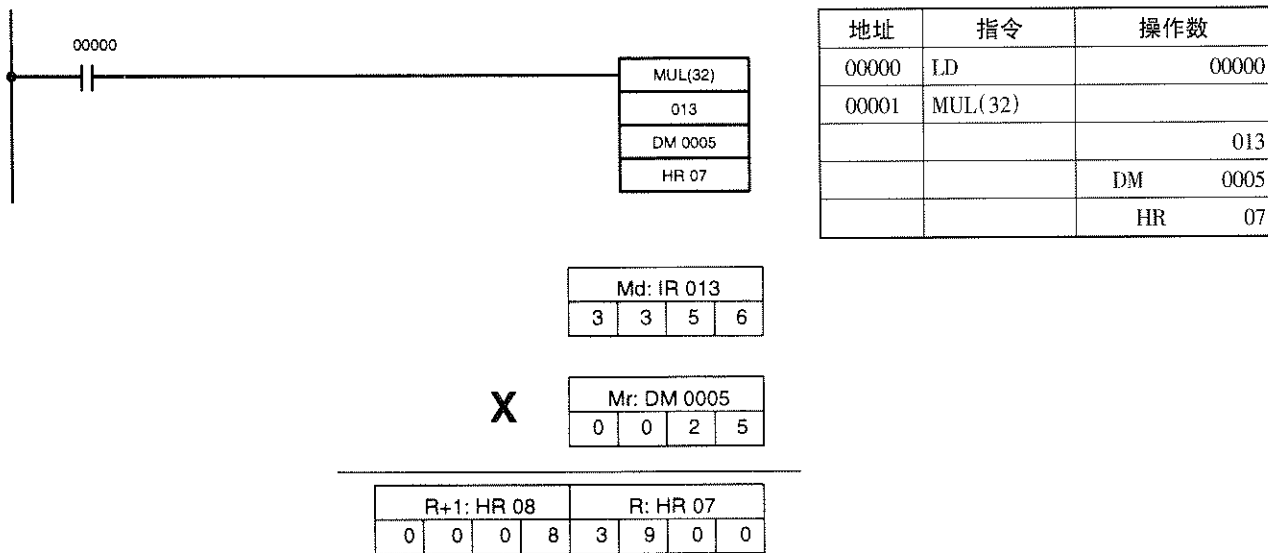
说明

当执行条件为 OFF 时, MUL(32) 不执行, 当执行条件为 ON 时, MUL(32) 将 Md 乘以 Mr 的内容, 并将结果置于 R 和 R + 1。



例

下例中, 当 IR00000 为 ON, IR013 和 DM0005 的内容相乘, 结果被置于 HR07 和 HR08。示例数据及计算过程在下图中显示。



标志

ER: Md 和/或 Mr 不是 BCD。

间接寻址的 EM/DM 字不存在。

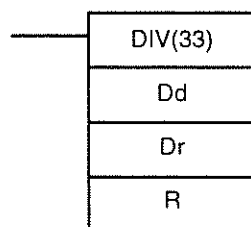
(* EM/*DM 字的内容不是 BCD, 或者超出 EM/DM 区边界。)

CY: 结果有进位时置 ON。

EQ: 结果为零时置 ON。

5-21-6 BCD 除 - DIV(33)

梯形图符号



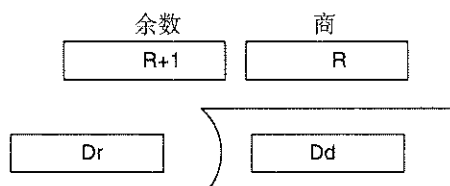
操作数数据区

| |
|--|
| Dd: 被除数字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Dr: 除数字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 首结果字 (BCD) |
| IR, SR, AR, DM, EM, HR, LR |

限制

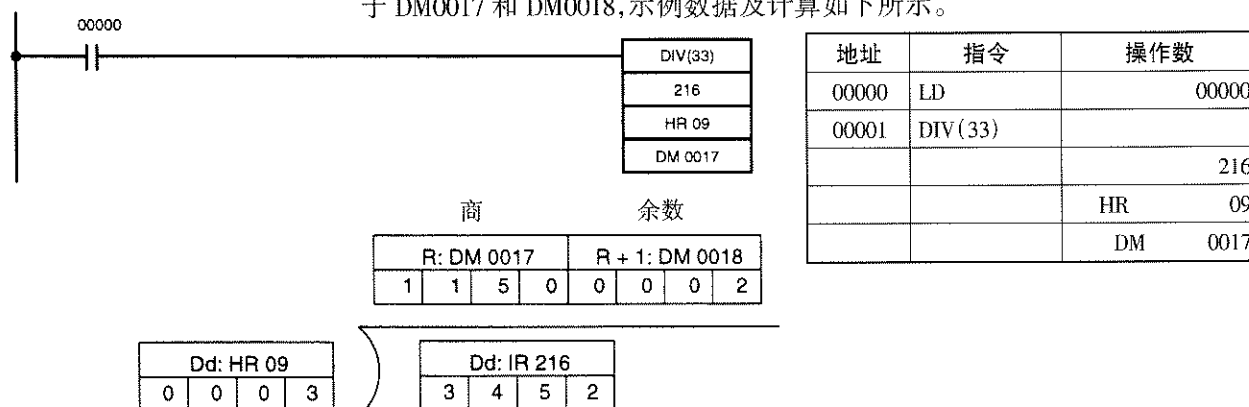
R 和 R + 1 必须在同一数据区。DM6143 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, DIV(33) 不执行, 并且程序移到下一指令。当执行条件为 ON 时, Dd 和 Dr 除并且结果被置于 R 和 R + 1; 商在 R 中, 余数在 R + 1 中。



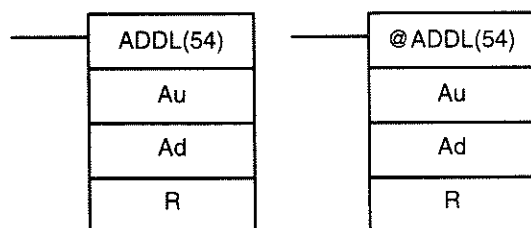
标志 ER: Dd 或 Dr 不是 BCD。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 结果为零时置 ON。

例 在下例中, 当 IR00000 为 ON 时, IR216 和内容被 HR09 的内容除, 并且结果置于 DM0017 和 DM0018, 示例数据及计算如下所示。



5-21-7 双字 BCD 加 - ADDL(54)

梯形图符号



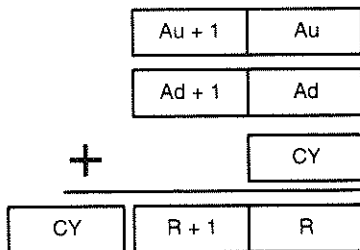
操作数数据区

| |
|-------------------------------------|
| Au: 被加数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| Ad: 加数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制 DM6143 ~ DM6655 不能作于 R。

说明

当执行条件为 OFF 时, ADDL(54) 不执行。当执行条件为 ON 时, ADDL(54) 将 CY 内容加到 8 位数值的 A_u 和 $A_u + 1$, 再加到 8 位数值的 A_d 和 $A_d + 1$, 并把结果置于 R 和 $R + 1$ 。若结果大于 99999999, 那么 CY 被置位。



标志

ER: A_u 和/或 A_d 不是 BCD。

间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

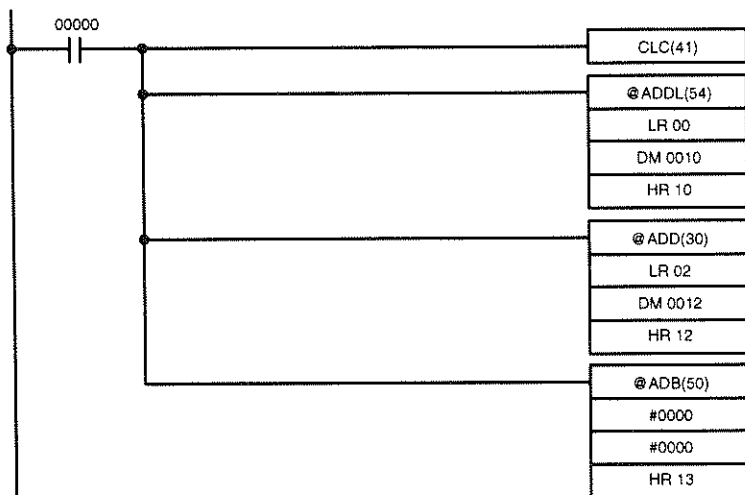
CY: 当结果有进位时置 ON。

EQ: 当结果为 0 时置 ON。

例

下述程序段中, 当 00000 为 ON 时, 将二个 12 位数值相加, 第一个数值存于 LR00 ~ LR02 之中, 第二个数值存于 DM0010 ~ DM0012 中。结果置于 HR10 ~ HR13。

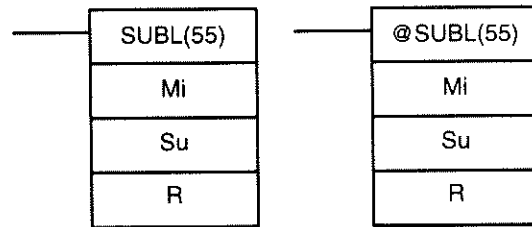
两个数据的最右 8 位数用 ADDL(54) 相加, 即 LR00 和 LR01 的内容加 DM0010 和 DM0011, 并将结果置于 HR10 ~ HR11。第二部分加法用 ADD(30) 将各数值的最左 4 位数相加, 还包括从第一部分加法得到的任何进位。最后的指令 ADB(50) (见 5-22-1 二进制加—ADB(50)) 相加两个全零常数以将第二部分加法得到的进位置于 HR13。



| 地址 | 指令 | 操作数 |
|-------|-----------|---------|
| 00000 | LD | 00000 |
| 00001 | CLC(41) | |
| 00002 | @ADDL(54) | |
| | | LR 00 |
| | | DM 0010 |
| | | HR 10 |
| 00003 | @ADD(30) | |
| | | LR 02 |
| | | DM 0012 |
| | | HR 12 |
| 00004 | @ADB(50) | |
| | | # 0000 |
| | | # 0000 |
| | | HR 13 |

5-21-8 双字 BCD 减 - SUBL(55)

梯形图符号



操作数数据区

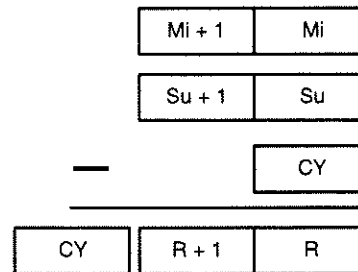
| |
|--|
| Mi: 被减首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Su: 减首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, SUBL(55) 不执行。当执行条件为 ON 时, SUBL(55) 从 Mi 和 Mi + 1 中的 8 位数值中减去 CY 及 Su 和 Su + 1 中的 8 位数值, 并将结果置于 R 和 R + 1。若结果为负数, CY 被置位, 并将实际结果的 10 进制补码置于 R。要将 10 进制补码转换为实际结果, 用零减去 R 的内容。因为 8 位数常数不能直接输入, 使用 BSET(71) 指令 (见 5-18-4 块设置——BSET(71)), 以生成一个 8 位数常数。



标志

ER: Mi, Mi + 1, Su, 或 Su + 1 不是 BCD。

间接寻址的 EM/DM 不存在。

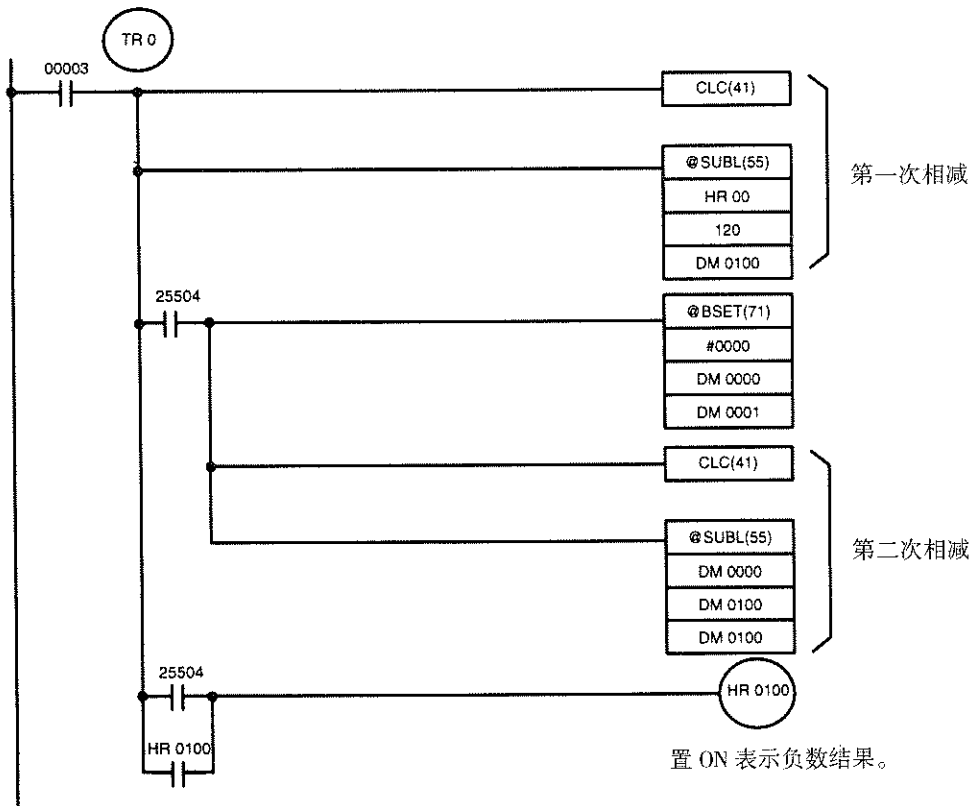
(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

CY: 当结果为负数 (即: Mi 小于 Su) 时, 置 ON。

EQ: 结果为 0 时置 ON。

例

下面的例子与单字减的例子非常相似。然而在本例中, 需要用 BSET(71) 来清除 DM0000 和 DM0001 的内容, 以使得负数结果能从 0 减去 (不可能直接输入一个 8 位数常数)。

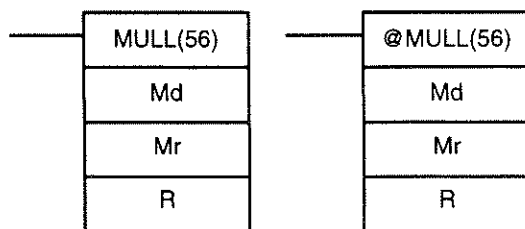


| 地址 | 指令 | 操作数 |
|-------|-----------|---------|
| 00000 | LD | 00003 |
| 00001 | OUT | TR 0 |
| 00002 | CLC(41) | |
| 00003 | @SUBL(55) | |
| | | HR 00 |
| | | 120 |
| | | DM 0100 |
| 00004 | AND | 25504 |
| 00005 | @BSET(71) | |
| | | # 0000 |
| | | DM 0000 |
| | | DM 0001 |

| 地址 | 指令 | 操作数 |
|-------|-----------|---------|
| 00006 | CLC(41) | |
| 00007 | @SUBL(55) | |
| | | DM 0000 |
| | | DM 0100 |
| | | DM 0100 |
| 00008 | LD | TR 0 |
| 00009 | LD | 25504 |
| 00010 | OR | HR 0100 |
| 00011 | AND LD | |
| 00012 | OUT | HR 0100 |

5-21-9 双字 BCD 乘 - MULL(56)

梯形图符号



操作数数据区

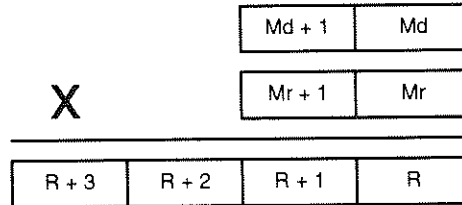
| |
|-------------------------------------|
| Md: 被乘数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| Mr: 乘数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

DM6141 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, MULL(56) 不执行。当执行条件为 ON 时, MULL(56) 把 Md 和 Md + 1 中的 8 位数值与 Mr 和 Mr + 1 中的 8 位数值相乘, 并把结果置于 R ~ R + 3。

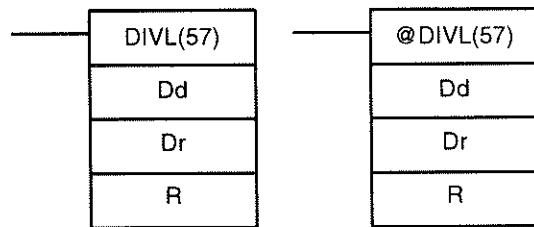


标志

ER: Md, Md + 1, Mr, 或 Mr + 1 不是 BCD。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 CY: 当结果中有进位时置 ON。
 EQ: 当结果为 0 时置 ON。

5-21-10 双字 BCD 除 - DIVL(57)

梯形图符号



操作数数据区

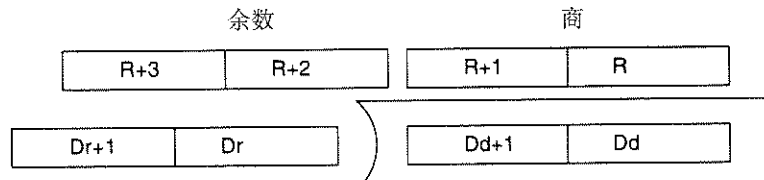
| |
|-------------------------------------|
| Dd: 被除数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| Dr: 除数首字 (BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

DM6141 ~ DM6655 不能用于 R。

说明

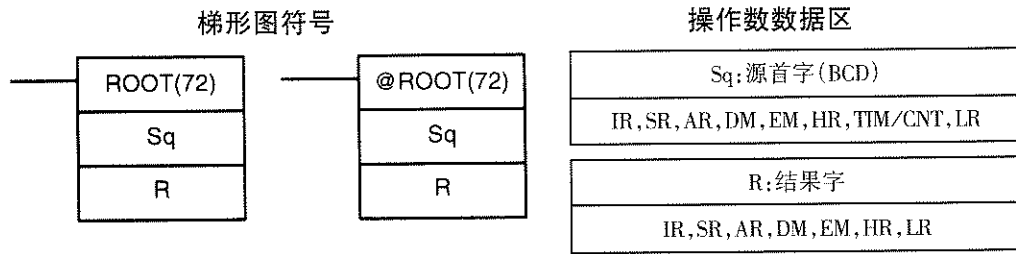
当执行条件为 OFF 时, DIVL(57) 不执行。当执行条件为 ON 时, DIVL(57) 将 Dd 和 Dd + 1 中的 8 位数除以在 Dr 和 Dr + 1 中的 8 位数, 结果置于 R ~ R + 3, 商存在 R 和 R + 1, 余数存在 R + 2 和 R + 3。



标志

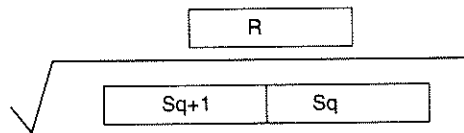
ER: Dr 和 Dr + 1 包含 0。
 Dd, Dd + 1, Dr, 或 Dr + 1 不是 BCD。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)
 EQ: 结果为 0 时置 ON。

5-21-11 平方根 - ROOT(72)



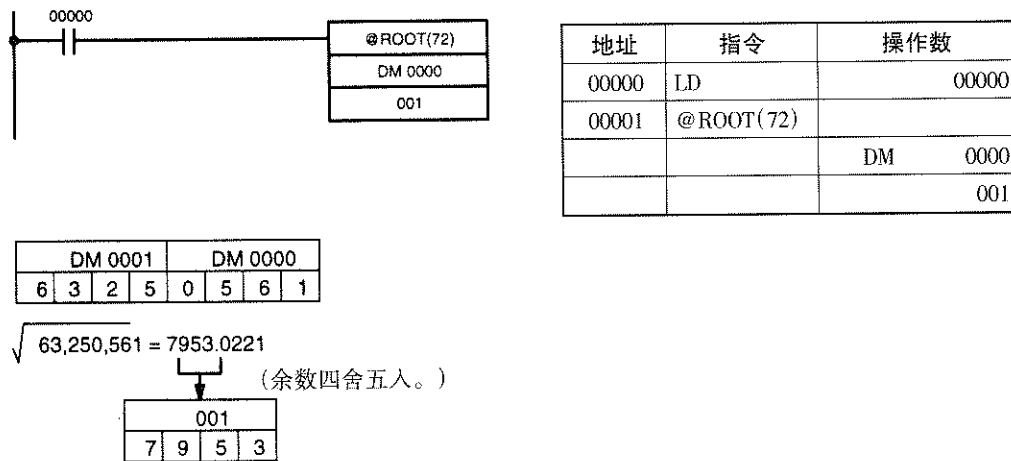
限制 DM6144 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时,ROOT(72)不执行。当执行条件为 ON 时,ROOT(72)计算 Sq 和 Sq + 1 中 8 位数的平方根,并将结果置于 R。小数部分则被舍掉。



标志 ER: Sq 不是 BCD。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)
 EQ: 当结果为 0 时置 ON。

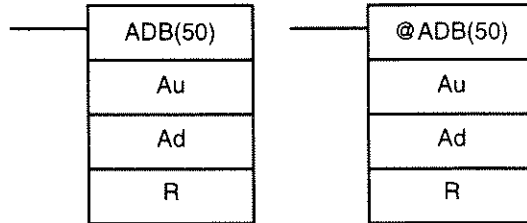
例 下例说明如何求出一个 8 位数的平方根。余数四舍五入后结果为 4 位数。
 在此例中, $\sqrt{63250561} = 7953.0221\dots$, 四舍五入后得 7953。



5-22 二进制计算指令

5-22-1 二进制加 - ADB(50)

梯形图符号



操作数数据区

| |
|--|
| Au: 被加字(二进制) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Ad: 加字(二进制) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 结果字 |
| IR, SR, AR, DM, EM, HR, LR |

限制
说明

DM6144 ~ DM6655 不能用于 R。
当执行条件为 OFF 时, ADB(50) 不执行。当执行条件为 ON 时, ADB(50) 把 Au、Ad 和 CY 的内容相加, 并把结果置于 R。如果结果大于 FFFF, 那么 CY 将被置位。

$$\boxed{\text{Au}} + \boxed{\text{Ad}} + \boxed{\text{CY}} \rightarrow \boxed{\text{CY}} \boxed{\text{R}}$$

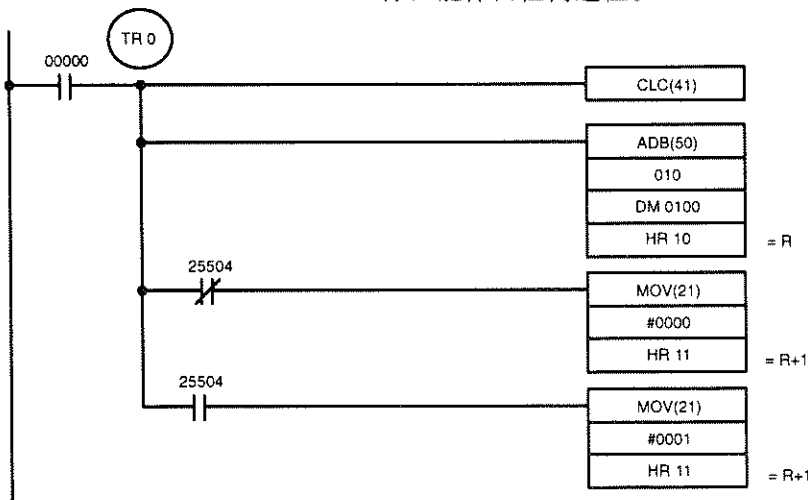
标志

ADB(50) 也可用于有符号二进制数据相加。下溢和上溢标志 (SR25404 和 SR25405) 表明结果是否超出 16 位有符号二进制数据范围的下限和上限。

- ER: 间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
- CY: 当结果大于 FFFF 时置 ON。
- EQ: 当结果为 0 时置 ON。
- OF: 当结果超出 + 32,767(7FFF) 时置 ON。
- UF: 当结果低于 - 32,768(8000) 时置 ON。

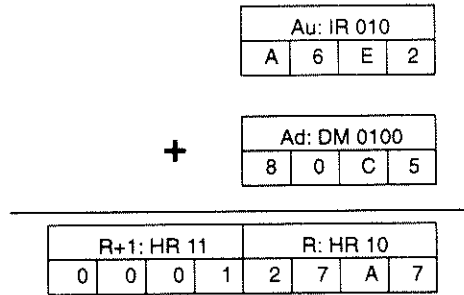
例

下例说明带 CY 的 4 位数相加, CY 用于将 # 0000 或 # 0001 置于 R + 1, 这样保证能保留任何进位。



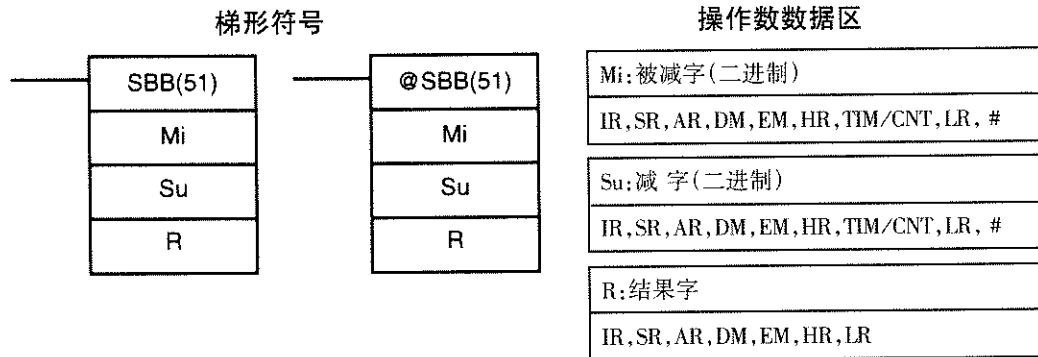
| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | OUT | TR 0 |
| 00002 | CLC(41) | |
| 00003 | ADB(50) | |
| | | 010 |
| | | DM 0100 |
| | | HR 10 |
| | | = R |
| 00004 | AND NOT | 25504 |
| 00005 | MOV(21) | # 0000 |
| | | HR 11 |
| | | = R+1 |
| 00006 | LD | TR 0 |
| 00007 | AND | 25504 |
| 00008 | MOV(21) | # 0001 |
| | | HR 11 |
| | | = R+1 |

在下述情况中, $A6E2 + 80C5 = 127A7$ 。结果是一个 5 位数, 因此 $CY(SR25504) = 1$, 并且 $R + 1$ 的内容变为 # 0001。



注 对于有符号二进制计算, UF 和 OF 标志的状态表明结果是否超出有符号二进制数据的范围 ($-32,768(8000) - +32,767(7FFF)$)。

5-22-2 二进制减 - SBB(51)



限制 DM6144 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, SBB(51) 不执行。当执行条件为 ON 时, SBB(51) 从 Mi 减 Su 和 CY 的内容, 并将结果置于 R。若结果为负数, CY 置位并将实际结果的 2 进制补码置于 R。

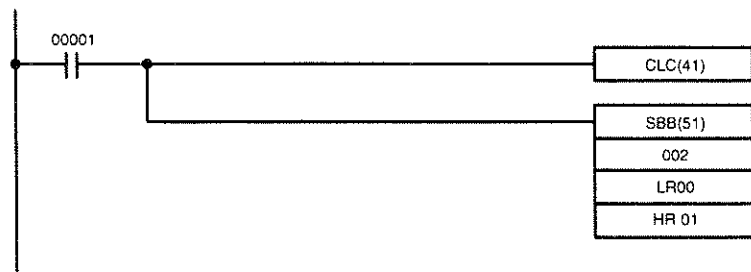


标志 SBB(51) 还可用于有符号二进制数据相减。下溢和上溢标志 (SR25404 和 SR25405) 表明结果是否已超出 16 位有符号二进制数据范围的下限和上限。

- ER: 间接寻址 EM/DM 字不存在。
- (* EM/ * DM 字内容不是 BCD, 或超出 EM/DM 区边界。)
- CY: 当结果是负数置 ON, 即 Mi 小于 Su 加 CY。
- EQ: 当结果为 0 置 ON。
- OF: 当结果大于 + 32,767(7FFF) 时置 ON。
- UF: 当结果小于 - 32,768(8000) 时置 ON。

例 下例为 4 位数相减。当 IR00001 为 ON 时, IR002 的内容减去 LR00 和 CY 的内容, 并将结果写到 HR01。

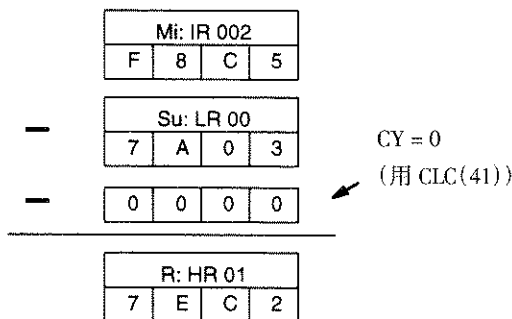
如果结果为负数,那么 CY 变 ON。如果使用正常数值,负数结果(有符号二进制)必须用 NEG(-)来转换成正常数据。详情参见 5-20-17 二进制补码—NEG(-)。



| 地址 | 指令 | 操作数 |
|-------|---------|-------|
| 00000 | LD | 00001 |
| 00001 | OUT | TR 1 |
| 00002 | CLC(41) | |
| 00003 | SBB(51) | |
| | | 002 |
| | | LR 00 |
| | | HR 01 |

在下面的情况下,IR002(# F8C5)的内容减去 LR00(# 7A03)和 CY 的内容,由于结果为正数,所以 CY 为 0。

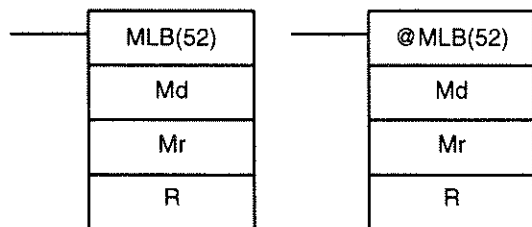
如果结果为负数,那么 CY 将被置为 1。对于正常(无符号)数据而言,结果应转换为它的二进制补码。



注 对有符号二进制计算,UF 和 OF 标志状态指示结果是否超出有符号二进制数据范围(- 32,768(8000) ~ + 32,767(7FFF))。

5-22-3 二进制乘—MLB(52)

梯形图符号



操作数数据区

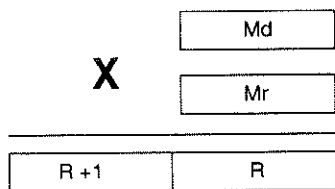
| |
|--|
| Md:被乘字(二进制) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| Mr: 乘字(二进制) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

DM6143 ~ DM6655 不能用于 R。

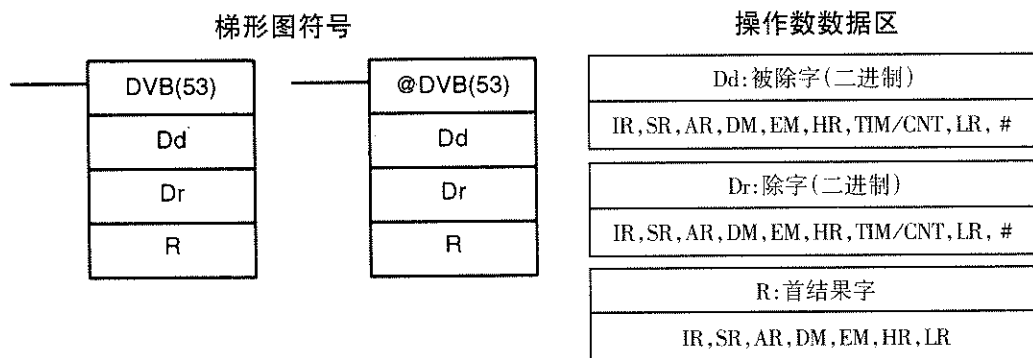
MLB(52)不能用于有符号二进制相乘,但可使用 MBS(-)。详情参见 5-22-7 有符号二进制乘——MBS(-)。

说明 当执行条件为 OFF 时, MLB(52) 不执行。当执行条件为 ON 时, MLB(52) 将 Md 内容乘以 Mr 内容, 并把结果的最右四位数字置于 R, 而最左四位数字置于 R + 1。



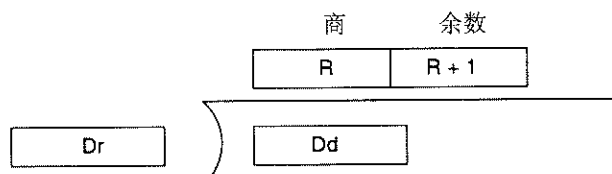
标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字内容不是 BCD, 或超出 EM/DM 区边界。)
 EQ: 当结果为 0 时置 ON。

5-22-4 二进制除 - DVB(53)



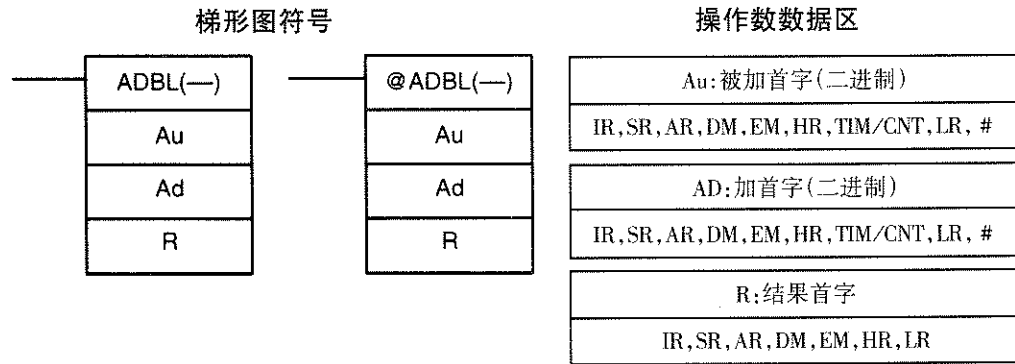
限制 DM6143 ~ DM6655 不能用于 R。
 DVB(53) 不能用作有符号二进制数据相除, 而要用 DBS(—)。详情参见 5-22-9 有符号二进制除 - DBS(—)。

说明 当执行条件为 OFF 时, DVB(53) 不执行。当执行条件为 ON 时, DVB(53) 将 Dd 内容除以 Dr 内容, 并置结果在 R 和 R + 1, 商在 R, 余数在 R + 1。



标志 ER: Dr 为 0。
 间接寻址的 EM/DM 字不存在。
 (* EM/DM 字内容不是 BCD, 或超出 EM/DM 区边界。)
 ER: 结果为零置 ON。

5-22-5 双字二进制加——ADBL(—)

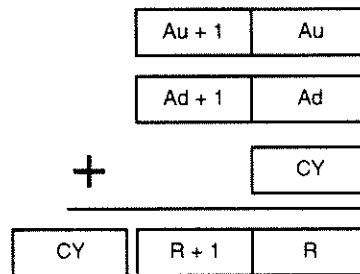


限制

AU 和 AU + 1 必须在同一数据区, AD 和 AD + 1, R 和 R + 1 也必须在同一数据区。
DM6142 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时,ADBL(—)不执行。当执行条件为 ON 时,ADBL(—)将 AU + 1 和 AU 的 8 位数内容与 AD + 1 和 AD 的 8 位数内容及 CY 相加,置结果于 R。若结果大于 FFFFFFFF,CY 被置位。



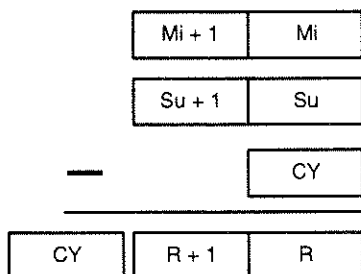
ADBL(—) 也可用于有符号二进制数据相加。下溢和上溢标志 (SR25404 和 SR25405) 指示结果是否超出 32 位有符号二进制数据的下限或上限。

标志

- ER: 间接寻址的 EM/DM 字不存在。
(* EM/* DM 字内容不是 BCD,或超出 EM/DM 区边界。)
- CY: 当结果大于 FFFFFFFF 时,置 ON。
- EQ: 当结果为 0 时置 ON。
- UF: 当结果低于 - 2,147,483,648(80000000) 时置 ON。
- OF: 当结果超过 + 2,147,483,647(7FFFFFFF) 时置 ON。

说明

当执行条件为 OFF 时,SBBL(—)不执行。当执行条件为 ON 时,SBBL(—)从 Mi 和 Mi + 1 的 8 位数值减去在 Su 和 Su + 1 中的 8 位数值和 CY, 并置结果于 R 和 R + 1。若结果为负数,CY 被置位并且实际结果的 2 进制补码被置于 R + 1 和 R。使用 NEGL(—)将 2 进制补码转换为实际结果。



标志

SBBL(—)还可用于有符号二进制减。下溢和上溢标志 (SR25404 和 SR25405) 指示结果是否超出 32 位有符号二进制数据的下限和上限。

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD,或超出 EM/DM 区边界)。

CY: 当结果为负数置 ON,即 Mi 小于 Su 加 CY。

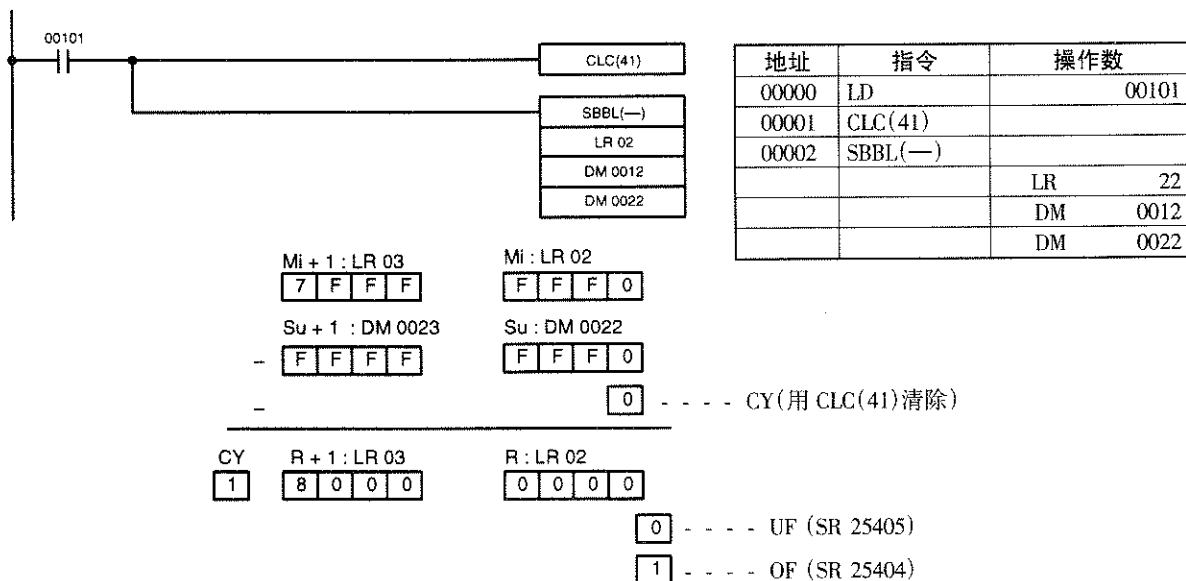
EQ: 当结果为 0 置 ON。

OF: 当结果超过 + 2, 147, 483, 647(7FFFFFFF)置 ON。

UF: 当结果小于 - 2, 147, 483, 648(80000000)置 ON。

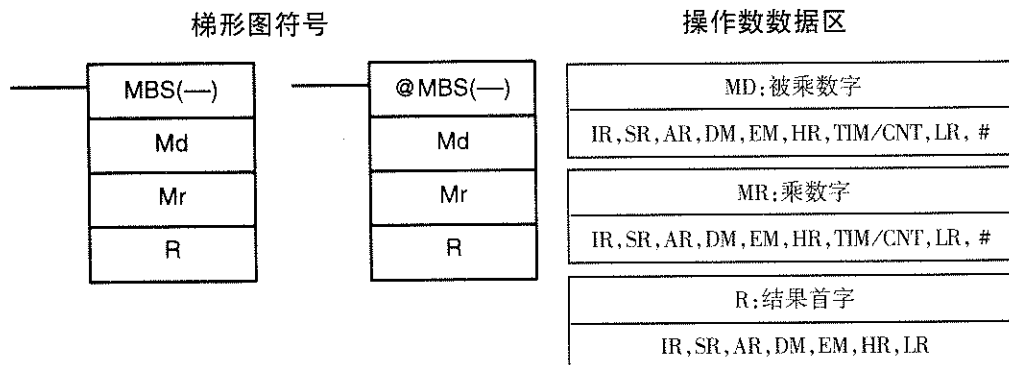
例

下例说明 8 位数相减,CY(SR25504)用来指示负数结果(对无符号数据)。UF 和 OF 标志状态指示结果是否超出有符号二进制数据的范围 (- 2, 147, 483, 648(80000000) ~ + 2, 147, 483, 647(7FFFFFFF))。



- 注 1. 对无符号二进制数据,CY 指示结果为负数,用 NEGL(—)求 2 进制补码以得到实际结果的绝对值(UF 和 OF 可忽略)。
2. 对有符号二进制数据,OF 标志指示结果超出 + 2, 147, 483, 647(7FFFFFFF)。(CY 可忽略)。

5-22-7 有符号二进制乘——MBS(—)



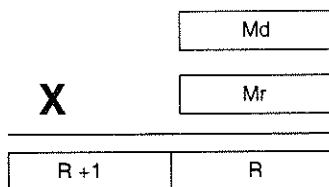
限制

DM6143 ~ DM6655 不能用于 R。

说明

MBS(—)将两个字有符号二进制内容相乘,并将 8 位数有符号二进制结果输出到 R+1 和 R。最右 4 位数置于 R,最左 4 位数置于 R+1。

注 详细情况参考 1-7 计算有符号二进制数据。



标志

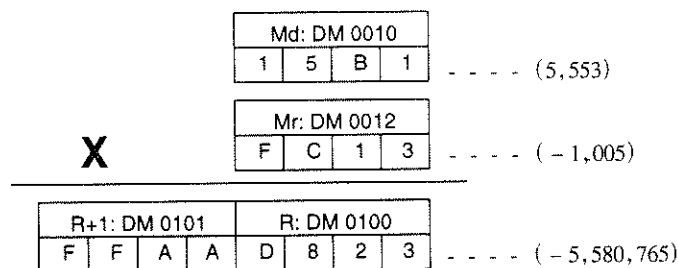
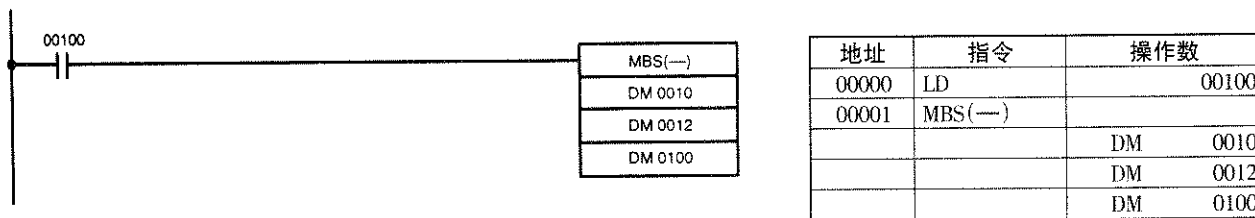
ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字内容不是 BCD,或超出 EM/DM 区边界)。

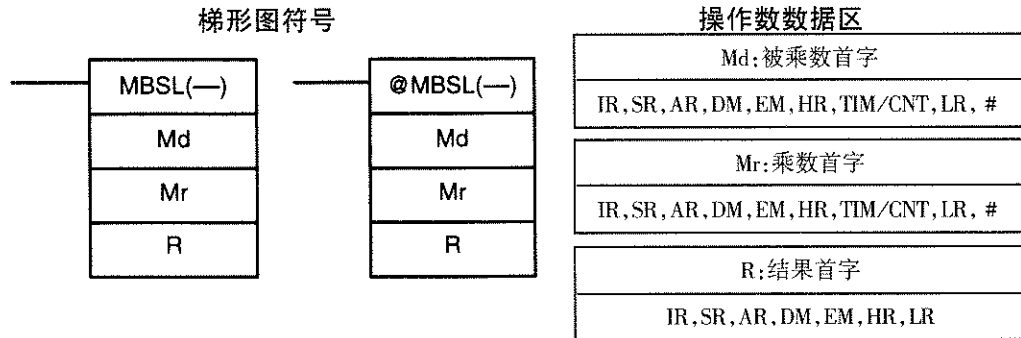
EQ: 若结果为 00000000 置 ON,否则置 OFF。

例

在下例中,MBS(—)用来将 DM0010 的有符号二进制内容乘以 DM0012 中的有符号二进制内容,并将结果输出到 DM0100 和 DM0101。



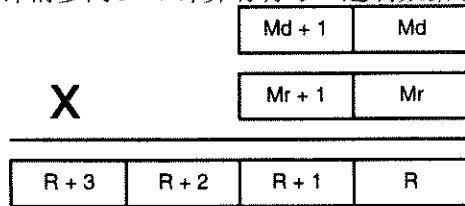
5-22-8 双字有符号二进制乘—MBSL(—)



限制 Md 和 Md + 1 必须在同一数据区, Mr 和 Mr + 1 也必须在同一数据区。
R 和 R + 3 必须在同一数据区。
DM6143 ~ DM6655 不能用于 R。

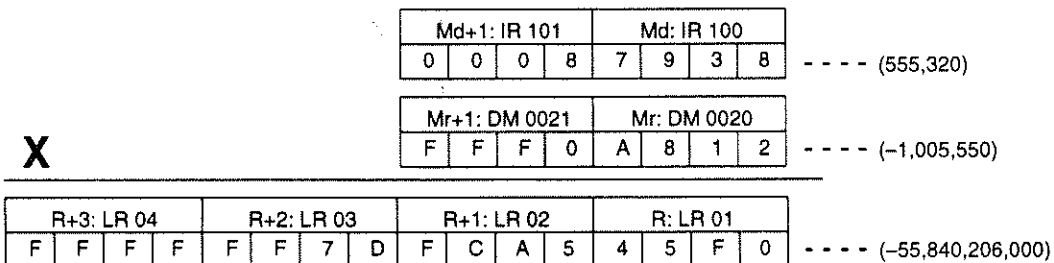
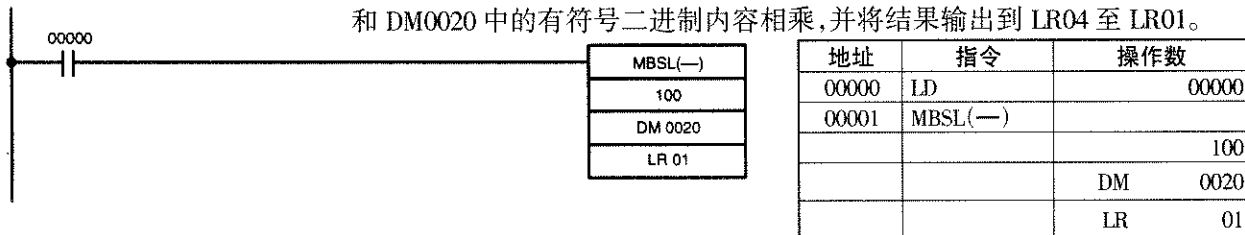
说明 MBSL(—) 将 Md + 1 和 Md 中的 32 位 (8 位数字) 的有符号二进制数据乘以 Mr + 1 和 Mr 中的 32 位有符号二进制数据, 并输出 16 位数字的有符号二进制结果到 R + 3 ~ R。

注 详情参阅 1-7 计算有符号二进制数据。

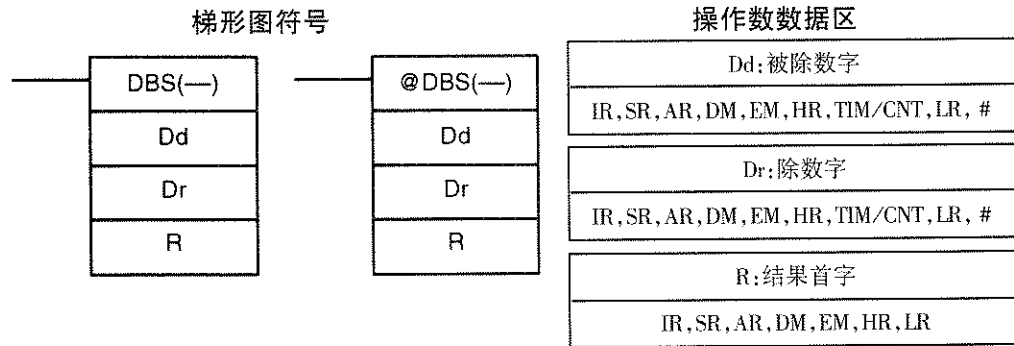


标志 ER: 间接寻址 EM/DM 字不存在。
(*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界)。
EQ: 当结果为 0(R + 3 ~ R 的内容全为 0) 置 ON, 否则置 OFF。

例 在下例中, MBSL(—) 用来将 IR101 和 IR100 的有符号二进制内容与 DM0021 和 DM0020 中的有符号二进制内容相乘, 并将结果输出到 LR04 至 LR01。



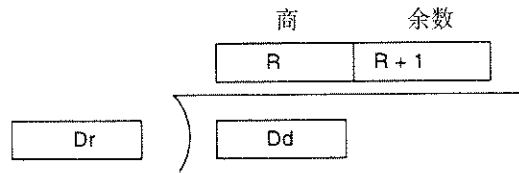
5-22-9 有符号二进制除——DBS(—)



限制 DM6143 ~ DM6655 不能用于 R。

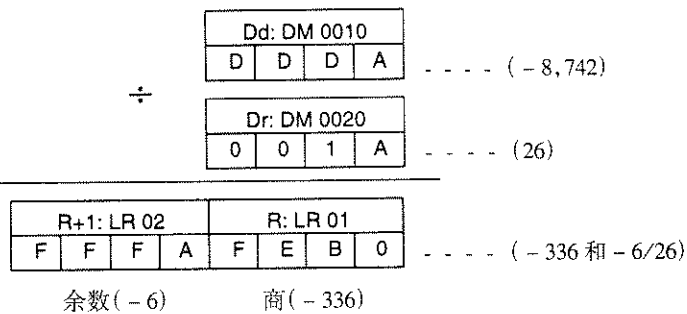
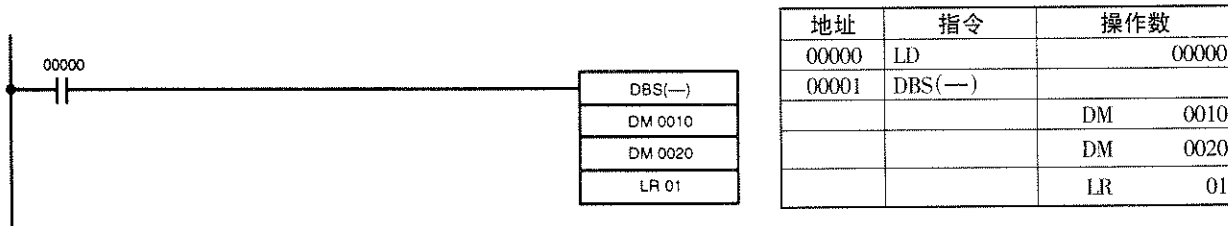
说明 DBS(—) 将 Dd 中的有符号二进制内容除以 Dr 中的有符号二进制内容, 并输出 8 位数字有符号二进制结果到 R + 1 和 R。商在 R, 余数在 R + 1。

注 详情请参考 1-7 计算有符号二进制数据。

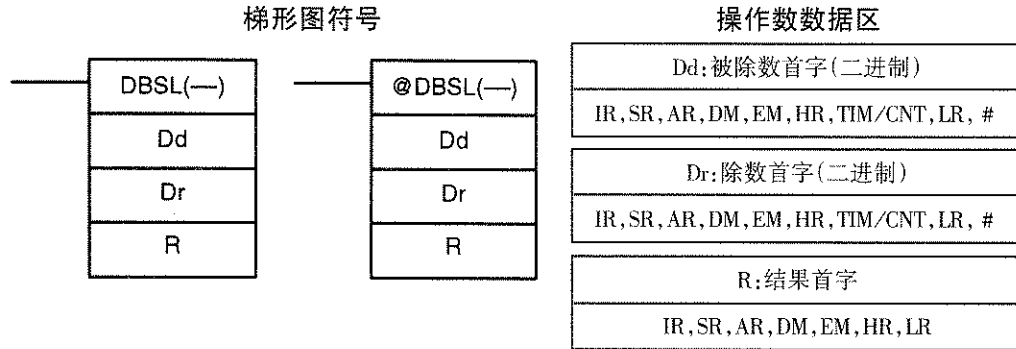


标志 ER: DR 为 0。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 字内容不是 BCD, 或超出 EM/DM 区边界)。

例 在下例, DBS(—) 用来将 DM0010 的有符号二进制内容除以 DM0020 中的有符号二进制内容, 并将结果输出到 LR01 和 LR02。



5-22-10 双字有符号二进制除——DBSL(—)

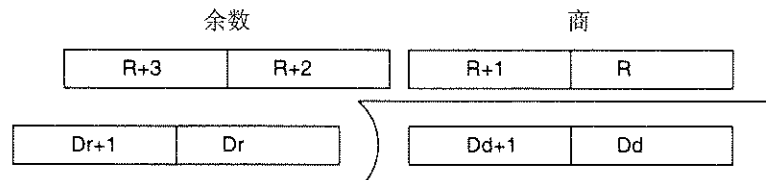


限制 Dd 和 Dd + 1 必须在同一数据区, Dr 和 Dr + 1, 及 R 和 R + 3 也必须在同一数据区。

DM6143 ~ DM6655 不能用于 R。

说明 DBSL(—) 使 Dd + 1 和 Dd 中的 32 位 (8 位数字) 有符号二进制数据被 Dr + 1 和 Dr 中的 32 位有符号二进制数据除, 并输出 16 位数有符号二进制结果到 R + 3 至 R。商在 R + 1 和 R, 余数在 R + 3 和 R + 2。

注 详情参考 1-7 计算有符号二进制数据。



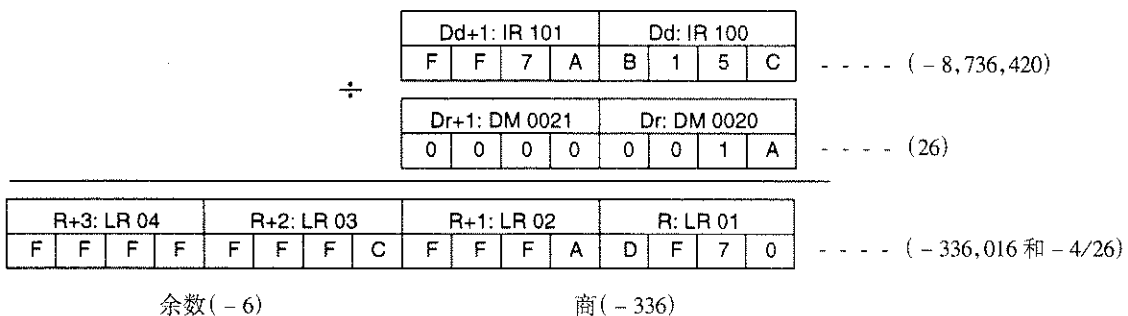
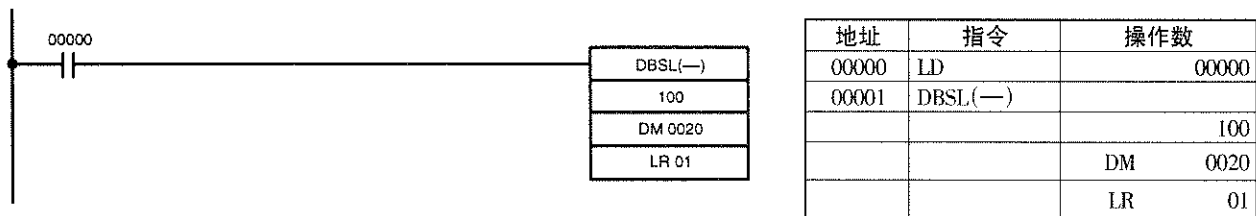
标志 ER: Dr + 1 和 Dr 值为 0。

间接寻址的 EM/DM 字不存在。

(* EM/* DM 字内容不是 BCD, 或超出 EM/DM 区边界。)

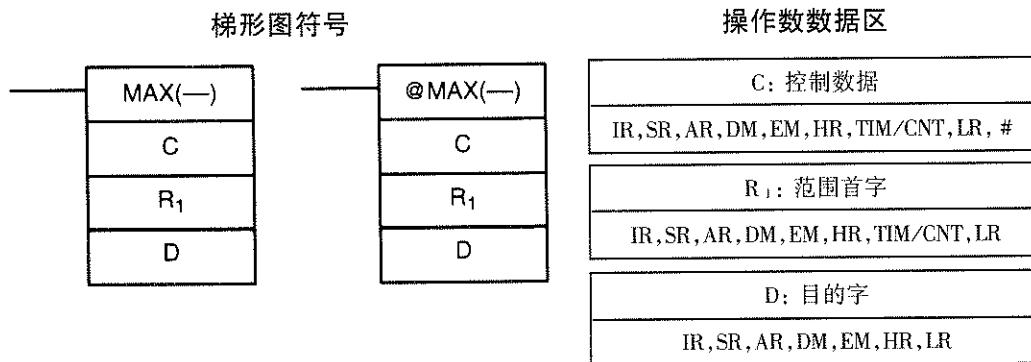
EQ: R + 1 和 R(商)的内容为 0 置 ON, 否则置 OFF。

例 在下例中, DBSL(—) 用来将 IR101 和 IR100 中的有符号二进制内容被 DM0021 和 DM0020 中的有符号二进制内容除, 并将结果输出到 LR04 至 LR01。



5-23 专用数学指令

5-23-1 找最大值 - MAX(-)



限制

N 必须是 0001 ~ 9999 之间的 BCD 值。

R₁ 和 R₁ + N - 1 必须在同一数据区。

DM6144 ~ DM6655 不能用于 D。

说明

当执行条件为 OFF 时, MAX(-) 不执行。当执行条件为 ON 时, MAX(-) 搜索 R₁ ~ R₁ + N - 1 的存储器范围, 找出含有最大值的地址, 并将最大值输出到目的字(D)。

若 C 的第 15 位为 ON, MAX(-) 将包含最大值的地址标记在 D + 1, 对 DM 区, 地址标记不一样。

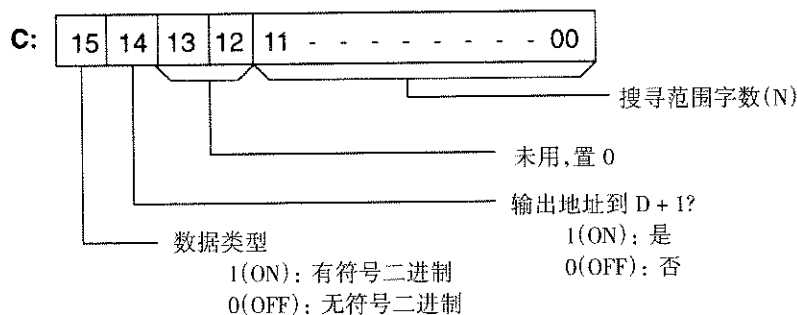
1, 2, 3, ... 1. 对于 DM 区地址, 将字地址写到 D + 1。例如, 若包含最大值的地址是 DM0114, 则把 # 0114 写到 D + 1。

2. 对于其它数据区地址, 将从搜索开始处的地址数写到 D + 1, 例如, 若包含最大值的地址是 IR114, 而搜索范围的首字是 IR014, 则把 # 0100 写到 D + 1。

若 C 的 14 位为 ON, 并且有多于一个的地址含有相同的最大值, 则地址中的最低位置将输出到 D + 1。若数据在 DM 区, 则用 DM 地址输出, 若为其它数据区, 则取对搜寻范围第一个地址的偏移量。

搜寻范围字数包含在 C 的最右 3 个数字中, 并且必须是 001 - 999 间的 BCD 值。

当 C 的第 15 位为 OFF, 在搜寻范围中的数据将作为无符号二进制值处理, 当该位为 ON, 数据作为有符号二进制值处理。

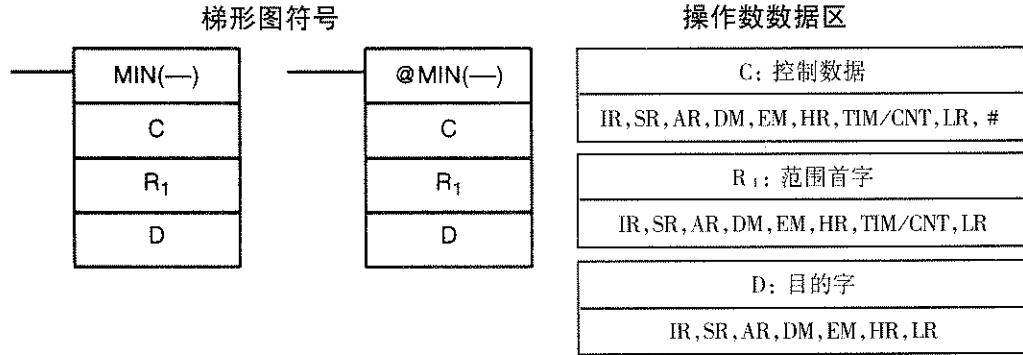


注意 若 C 的第 14 位为 ON, 在 # 8000 以上的值将被视为负数, 所以结果会随数据类型的指定而变。要保证指定正确的数据类型。

标志

ER: 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界)。
 R_i 和 R_i + N - 1 不在同一数据区。
 EQ: 若最大值为 # 0000 置 ON。

5-23-2 找最小值 - MIN(—)



限制

N 必须是 0001 ~ 9999 之间的 BCD 值。
 R_i 和 R_i + N - 1 必须在同一数据区。

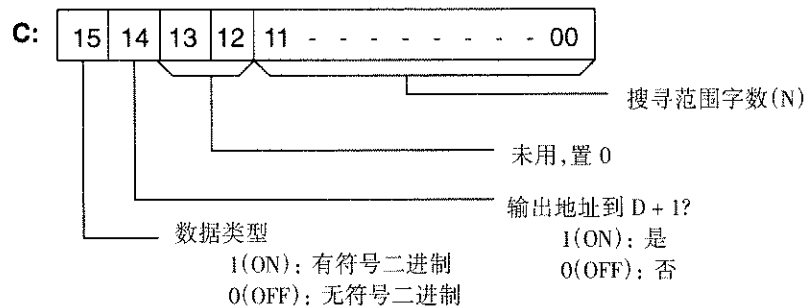
说明

DM6144 ~ DM6655 不能用于 D。
 当执行条件为 OFF 时, MIN(—) 不执行。当执行条件为 ON 时, MIN(—) 搜索 R_i ~ R_i + N - 1 存储器范围, 找出包含最小值的地址, 并将最小值输出到目的字(D)。
 若 C 的第 15 位为 ON, MIN(—) 将包含最小值的字地址标记在 D + 1, 对 DM 区, 地址标记不一样。

- 1, 2, 3... 1. 对于 DM 区地址, 将字地址写到 D + 1。例如, 若包含最小值的地址是 DM0114, 则把 # 0114 写到 D + 1。
2. 对于其它数据区地址, 将从搜索开始处的地址数写到 D + 1, 例如, 若包含最小值的地址是 IR114, 而搜索范围的起始字是 IR014, 则把 # 0100 写到 D + 1。

若 C 的第 14 位为 ON, 并且多于一个的地址包含相同的最小值, 这些地址中的最低一个将写到 D + 1。若是 DM 区数据, 则写 DM 地址, 若为其它数据区, 则写搜寻范围首地址的偏移量。

搜寻范围字数 (N) 包含在 C 最右 3 位数字, N 必须是 001 ~ 999 之间的 BCD 值。当 C 的第 15 位为 OFF, 范围区内的数据按无符号二进制值处理, 若为 ON, 数据按有符号二进制值处理。

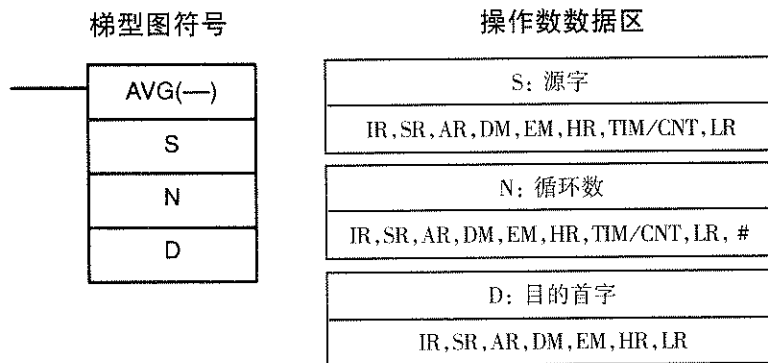


注意 若 C 的第 14 位为 ON, 在 # 8000 以上的值按负数处理, 所以结果将按数据类型的指定而变。确保指定正确的数据类型。

标志

ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)
 R_i 和 R_i + N - 1 不在同一数据区。
 EQ: 当最小值为 # 0000 置 ON。

5-23-3 求平均值 - AVG(-)



限制

S 必须是 16 进制值。
 N 必须是从 # 0001 ~ # 0064 之间的 BCD 值。
 D 和 D + N + 1 必须在同一数据区。
 DM6144 ~ DM6655 不能用于 S, N 或 D ~ D + N + 1。

说明

AVG 用来计算 S 在 N 个循环中的平均值。
 当执行条件为 OFF 时, AVG(-) 不执行。
 每次执行 AVG(-), S 的内容被存储在字 D + 2 ~ D + N + 1。在第一次执行时, AVG(-) 将 S 内容写到 D + 2; 在第二次执行, 它将 S 的内容写到 D + 3, 等等。在第 N 次执行, AVG(-) 将 S 的内容写到 D + N + 1, 并计算存在 D + 2 ~ D + N + 1 中的数值的平均值, 再将这平均值写到 D。
 下图表说明字 D ~ D + N + 1 的功能。

| | |
|-----------|------------------------|
| D | 平均值 (在 N 次或更多次执行) |
| D + 1 | 系统使用 |
| D + 2 | 第一次执行 AVG(-) 时 S 的内容 |
| D + 3 | 第二次执行 AVG(-) 时 S 的内容 |
| ⋮ | ⋮ ⋮ ⋮ |
| D + N + 1 | 第 N 次执行 AVG(-) 时 S 的内容 |

注意

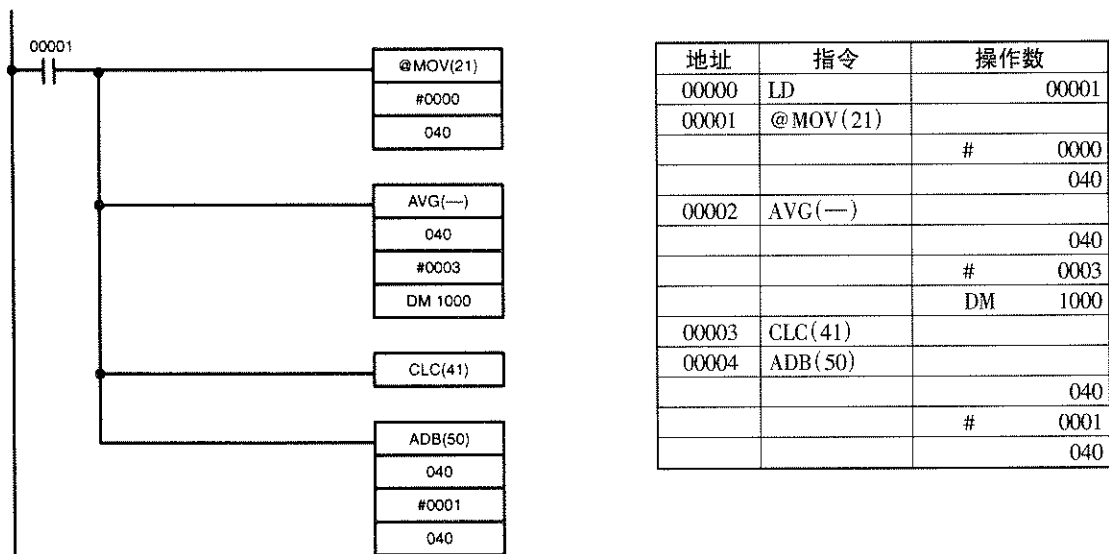
平均值用二进制计算。应保证 S 的内容为二进制。
 N 必须为 # 0001 ~ # 0064 之间的 BCD 值。若 N ≥ # 0065, AVG(-) 将按 N = 64 运算。平均值将四舍五入到最接近的整数。(0.5 被四舍五入为 1。)
 AVG(-) 在执行第一次后 D + 1 的内容设置为 # 0000。

标志

ER: 间接寻址的 EM/DM 字不存在。
 (* EM/*DM 字内容不是 BCD,或超出 EM/DM 区边界。)
 一个或多个操作数设置不正确。
 D 和 D + N + 1 不在同一数据区。

例

在下例中,IR040 内容被设为 # 0000,然后每一循环增加 1。在头两个循环中,AVG(—) 将 IR040 内容移至 DM1002 和 DM1003。在第三和再后一个循环,AVG(—)计算 DM1002 ~ 1004 中内容的平均值,将并结果写到 DM1000。



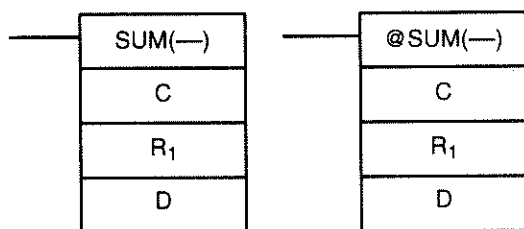
| | 第一个循环 | 第二个循环 | 第三个循环 | 第四个循环 |
|--------|-------|-------|-------|-------|
| IR 040 | 0000 | 0001 | 0002 | 0003 |

| | 第一个循环 | 第二个循环 | 第三个循环 | 第四个循环 |
|---------|-------|-------|-------|-------|
| DM 1000 | 0000 | 0001 | 0001 | 0002 |
| DM 1001 | | | | |
| DM 1002 | 0000 | 0000 | 0000 | 0003 |
| DM 1003 | ... | 0001 | 0001 | 0001 |
| DM 1004 | ... | ... | 0002 | 0002 |

平均值由系统使用。
IR40 先前值

5-23-4 求和 - SUM(—)

梯形用符号



操作数数据区

| |
|-------------------------------------|
| C: 控制数据 |
| IR, SR, AR, DM, EM, HR, LR, # |
| R ₁ : 范围首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| D: 目的首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

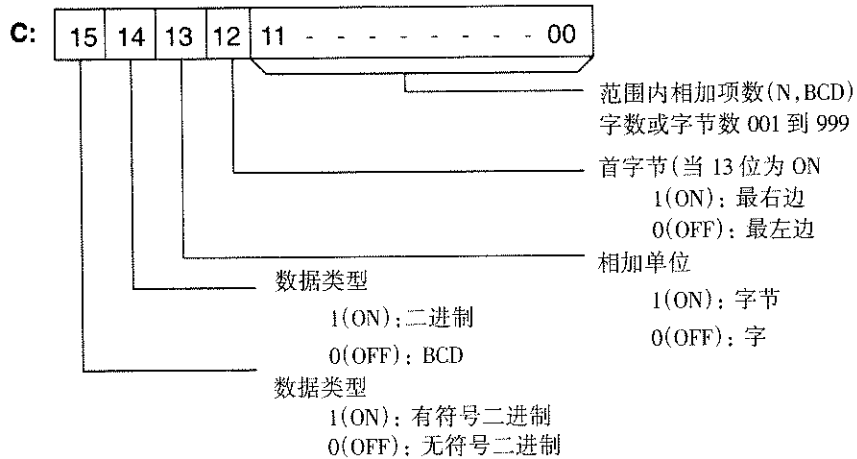
C 的最右 3 个数字必须是 001 ~ 999 之间的 BCD 值。

DM6143 ~ DM6655 不能用于 D。

若 C 的 14 位为 OFF(设置为 BCD 数据), 在范围 $R_1 \sim R_1 + N - 1$ 中的所有数据必须是 BCD 值。

说明

当执行条件为 OFF 时, SUM(一)不执行。当执行条件为 ON 时, SUM(一)将 $R_1 \sim R_1 + N - 1$ 字中的内容或 $R_1 \sim R_1 + N/2 - 1$ 字中的字节相加, 并输出该值至目的字(D 和 D + 1)。该数据能以二进制或 BCD 值求和, 并以相同格式输出。二进制数据可以是有符号的或无符号的。C 中位的功能在下图中说明, 并在以下作更详细的解释。



范围内项数

范围内项数(N)包含在 C 的最右 3 位数字, 它必须是 001 ~ 999 间的 BCD 值, 该数据依据求和项表示字数或字节数。

相加单位

若第 13 位为 OFF, 相加的是字, 若为 ON, 相加的是字节。

若指定了字节, 相加范围可由 R_1 的最左字节或最右字节起始, 若 C 的第 12 位为 ON, R_1 的最左字节不相加。

| | MSB | LSB |
|-----------|-----|-----|
| R_1 | 1 | 1 |
| $R_1 + 1$ | 3 | 4 |
| $R_1 + 2$ | 5 | 6 |
| $R_1 + 3$ | 7 | 8 |
| ⋮ | ⋮ | ⋮ |

若第 12 位为 OFF, 字节相加将按此次序: 1 + 2 + 3 + 4 + ...

若第 12 位为 ON, 字节相加将按此次序: 2 + 3 + 4

数据类型

若 C 的第 14 位为 ON, 第 15 位为 OFF, 范围内数据按无符号二进制处理, 若 C 的第 14 位和第 15 位都是 ON, 数据按有符号二进制处理。若 C 的第 14 位为 OFF, 不管第 15 位的状态如何, 范围内数据按 BCD 处理。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字内容不是 BCD, 或超出 EM/DM 区边界。)

R_1 和 $R_1 + N - 1$ 不在同一数据区。

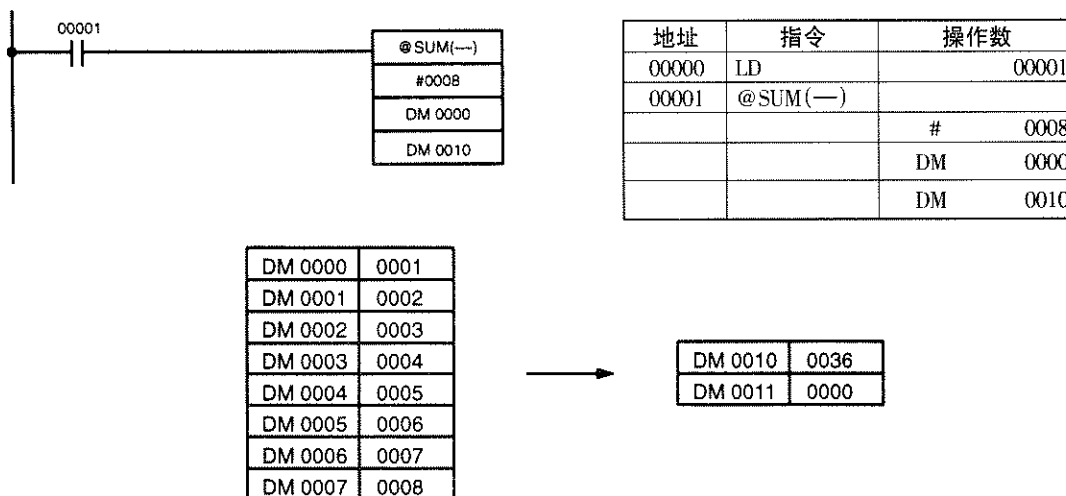
C 中的项数不是 001 ~ 999 间的 BCD 值。

当指定 BCD 值时, 求和数据不是 BCD 值。

EQ: 结果为 0 时置 ON。

例

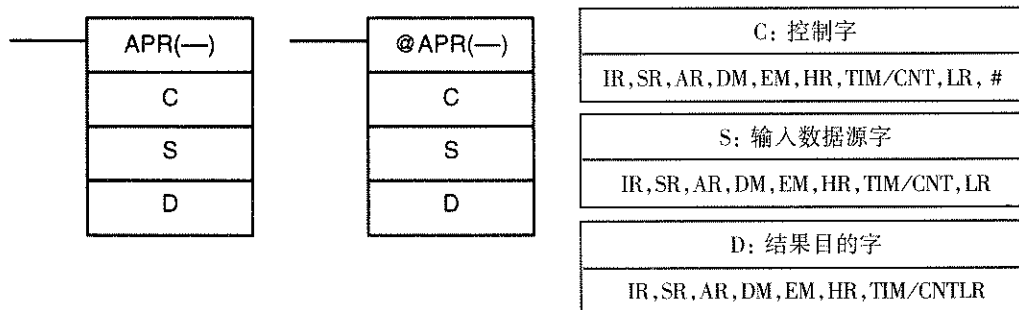
在下例中, 当 IR0001 为 ON, DM0000 ~ DM0007 中 8 个字的 BCD 内容相加, 结果被写到 DM0010 和 DM0011。



5-23-5 算术处理 - APR(—)

梯形图符号

操作数数据区



限制

对于三角函数, S 必须是 0000 ~ 0900 ($0^\circ \leq \theta \leq 90^\circ$) 之间的 BCD 值。

DM6144 ~ DM6655 不能用于 D。

说明

当执行条件为 OFF 时, APR(—) 不执行。当执行条件为 ON 时, APR(—) 的运算依据控制字 C。

若 C 是 # 0000 或 # 0001, APR(—) 计算 $\sin\theta$ 或 $\cos\theta^*$ 。S 中的 BCD 值是以 0.1 度为单位的 θ 。

若 C 是一个地址, APR(—) 计算 $f(x)$, 这是事先输入的, 以 C 开始存储的函数。这是由编程者确定的一系列线段(它可近似一条曲线)。S 的 BCD 值或 16 进制值表示 x 。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字不是 BCD, 或超出 EM/DM 区边界。)

对于三角函数, $x > 0900$ 。(x 为 S 的内容)。

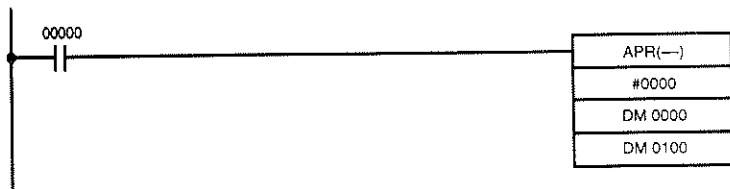
给 C 设定的常数是 # 0000 或 # 0001 以外的值。

线性逼近数据是不可读的。

EQ: 结果为 0000。

例
sin 函数

下例说明使用 APR(—)sin 函数计算 30° 的 sin 值,当 C 为 # 0000 时指定为 sin 函数。



| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD | 00000 |
| 00001 | APR(—) | |
| | | # 0000 |
| | | DM 0000 |
| | | DM 0100 |

输入数据, x

| S: DM 0000 | | | |
|------------|-----------------|-----------------|------------------|
| 0 | 10 ¹ | 10 ⁰ | 10 ⁻¹ |
| 0 | 3 | 0 | 0 |

输入数据为 BCD 值
并且不超过 # 0900

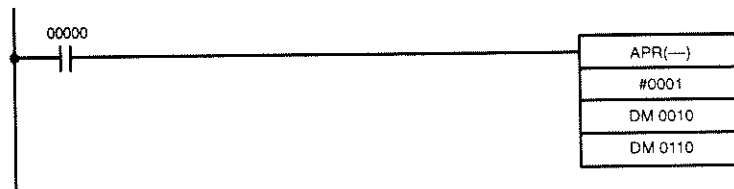
结果数据

| D: DM 0100 | | | |
|------------------|------------------|------------------|------------------|
| 10 ⁻¹ | 10 ⁻² | 10 ⁻³ | 10 ⁻⁴ |
| 5 | 0 | 0 | 0 |

结果数据有 4 位有效数字,
第五和更多的数字忽略。
sin(90)的结果是 0.9999 而不是 1。

cos 函数

下例说明用 APR(—)cos 函数计算 cos30°。当 C 为 # 0001 指定为 cos 函数。



| 地址 | 指令 | 操作数 |
|-------|--------|---------|
| 00000 | LD | 00000 |
| 00001 | APR(—) | |
| | | # 0001 |
| | | DM 0010 |
| | | DM 0110 |

输入数据, x

| S: DM 0010 | | | |
|------------|-----------------|-----------------|------------------|
| 0 | 10 ¹ | 10 ⁰ | 10 ⁻¹ |
| 0 | 3 | 0 | 0 |

输入数据为 BCD 值
并且不超过 # 0900

结果数据

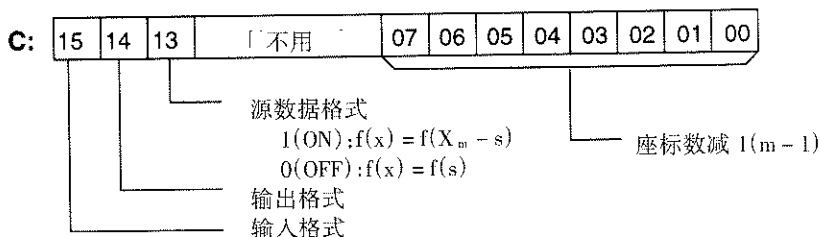
| D: DM 0110 | | | |
|------------------|------------------|------------------|------------------|
| 10 ⁻¹ | 10 ⁻² | 10 ⁻³ | 10 ⁻⁴ |
| 8 | 6 | 6 | 0 |

结果数据有 4 位有效数字,
第五和更多的数字忽略。
cos(0)的结果是 0.9999 而不是 1。

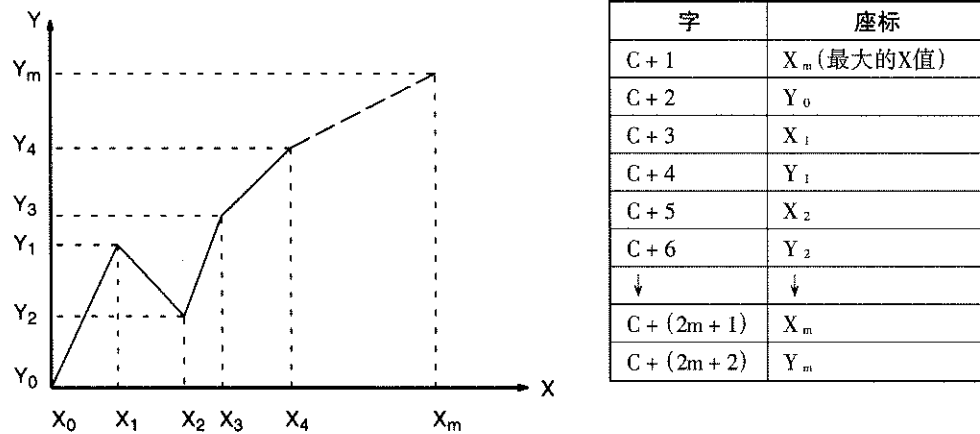
线性逼近

为 C 为存储器地址时指定 APR(—)线性近似, C 是包含线性逼近数据的存储器连续块的首字。

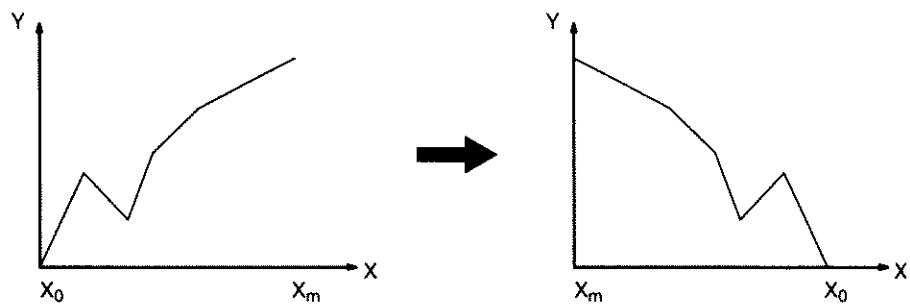
C 的内容规定了逼近中的线段数,及输入和输出是 BCD 还是二进制格式。位 00~07 包含了线段数减 1(m-1)的二进制数据,位 14 和 15 分别确定输出和输入格式;0 规定 BCD,1 规定二进制。



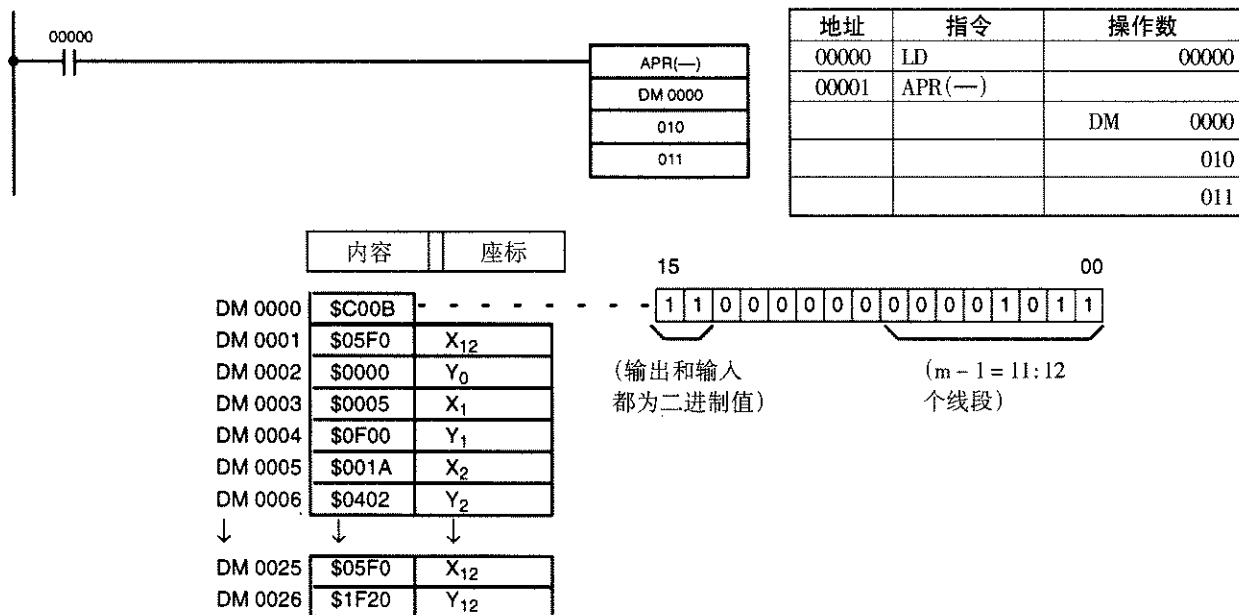
输入 $m+1$ 个端点的坐标,它们定义了 m 个线段,如下表所示。所有坐标用二进制格式输入。输入座位总是从 X 的最小值(X_0)到最大值(X_m) X_0 是 0000,不需要输入。



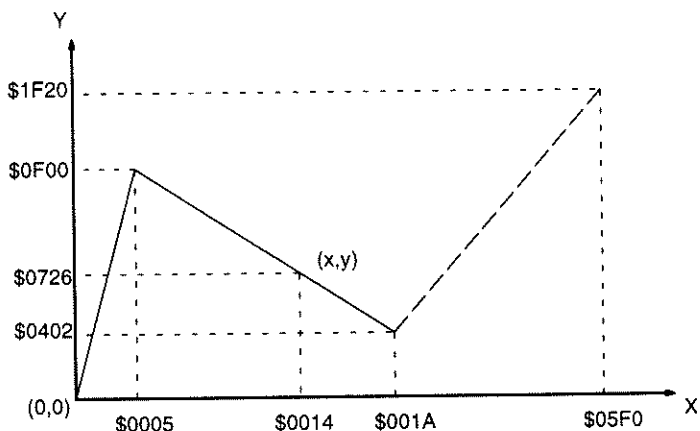
假若 C 的 13 位设为 1,那么函数图象将从左映射到右,如下图所示。



下例说明具有 12 个线段的线性逼近结构。数据块必须是连续的,从 DM0000 ~ DM0026 ($C \sim C + (2 \times 12 + 2)$)。输入数据从 IR010 获得,结果输出到 IR011。



在本例中,输入数据字 IR010 包含了 # 0014, $f(0014) = \# 0726$ 输出到 R (IR011)。



5 - 24 浮点算术指令

浮点算术指令转换数据并执行浮点算术运算。CQM1H - 系列 CPU 支持下述指令。

| 指令 | 助记符 | 功能代码 | 页数 |
|--------|------|------|-----|
| 浮点到16位 | FIX | — | 340 |
| 浮点到32位 | FIXL | — | 341 |
| 16位到浮点 | FLT | — | 342 |
| 32位到浮点 | FLTL | — | 343 |
| 浮点加法 | + F | — | 343 |
| 浮点减法 | - F | — | 345 |
| 浮点乘法 | * F | — | 346 |
| 浮点除法 | /F | — | 347 |
| 度到弧度 | RAD | — | 348 |
| 弧度到度 | DEG | — | 349 |
| 正弦 | SIN | — | 350 |
| 余弦 | COS | — | 351 |
| 正切 | TAN | — | 352 |
| 反正弦 | ASIN | — | 353 |
| 反余弦 | ACOS | — | 354 |
| 反正切 | ATAN | — | 355 |
| 平方根 | SQRT | — | 356 |
| 指数 | EXP | — | 357 |
| 对数 | LOG | — | 359 |

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,用下列计算式。

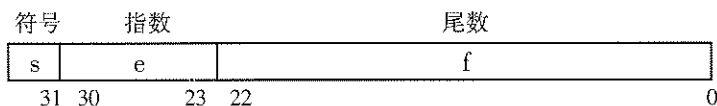
$$\text{实数} = (-1)^s 2^{e-127} (1.f)$$

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:

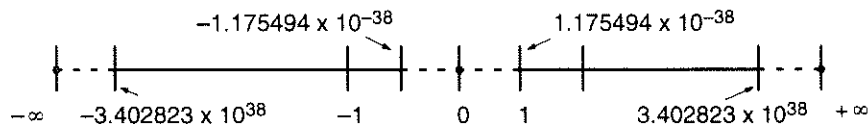


| 数据 | 位数 | 内容 |
|-------|----|---|
| s: 符号 | 1 | 0: 正; 1: 负 |
| e: 指数 | 8 | 指数 (e) 值从 0 ~ 255。实际指数是 e 减去 127 后的余数, 结果从 -127 ~ 128。“e = 0” 和 “e = 255” 表示特殊数字。 |
| f: 尾数 | 23 | 二进制浮点数据的尾数部分满足 $2.0 > 1.f \geq 1.0$ |

数字个数
浮点数据

浮点数据的有效数字个数为 24 位二进制数 (约七个数字十进制数)。
下列数据可以用浮点数据表示:

- $-\infty$
- $-3.402823 \times 10^{38} \leq \text{值} \leq -1.175494 \times 10^{-38}$
- 0
- $1.175494 \times 10^{-38} \leq \text{值} \leq 3.402823 \times 10^{38}$
- $+\infty$
- 不是一个数字 (NaN)



特殊数字

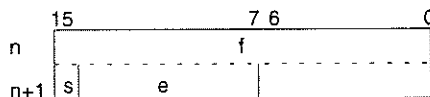
NaN, $\pm \infty$, 和 0 的格式如下:

- NaN*: e = 255, f ≠ 0
- $+\infty$: e = 255, f = 0, s = 0
- $-\infty$: e = 255, f = 0, s = 1
- 0: e = 0

*NaN (不是一个数字) 不是一个有效的浮点数。执行浮点计算指令不会导致 NaN。

写浮点数据

当在 CX-Programmer 中把 I/O 内存编辑显示器里的数据格式指定为浮点时, 显示器里输入的标准十进制数字自动转换为上面 (IEEE754 格式) 所示的浮点格式, 并且写到 I/O 内存。当在显示器上监控时, 用 IEEE754 格式写的数字自动转换为标准十进制格式。



读和写浮点数据时, 用户不需要知道 IEEE754 数据格式。只需记住每个浮点数占有两个字。

下列类型浮点数可使用。

| 尾数 (f) | 指数 (e) | | |
|--------|--------|-----------|-----------|
| | 0 | 非 0 和非全 1 | 全 1 (255) |
| 0 | 0 | 标准数字 | 无穷大 |
| 非 0 | 非标准数字 | | NaN |

注 非标准数字是绝对值太小而不能表示为标准数字的数字。非标准数字具有较少的有效数字。如果计算结果是非标准数字。(包括中间结果),有效数字的个数将减少。

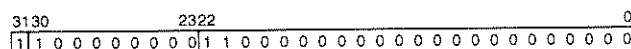
标准数字

标准数字表示实数。符号位为 0 表示正数,1 表示负数。
 指数(e)将从 1 到 254 表示,实际指数将小于 127,即 -126 到 127。
 尾数(f)将从 0 到 $2^{23} - 1$ 表示,它假设,在实际尾数中,位 2^{23} 为 1,并且二进制小数点立即跟在它后面。

标准数字如下表示:

$$(-1)^{(\text{符号 } s)} \times 2^{(\text{指数 } e) - 127} \times (1 + \text{尾数} \times 2^{-23})$$

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: $-1.75 \times 2^1 = -3.5$

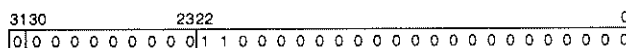
非标准数字

非标准数字表示绝对值很小的实数。
 符号为 0 表示正数,为 1 则表示负数。
 指数(e)为 0,且实际指数将为 -126。
 尾数(f)从 $1 \sim 2^{23} - 1$ 表示,并假设在实际尾数中,位 2^{23} 为 0,并且二进制小数点立即跟在它后面。

非标准数字表示如下:

$$(-1)^{(\text{符号 } s)} \times 2^{-126} \times (\text{尾数} \times 2^{-23})$$

例



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}

零

+0.0 和 -0.0 值可通过设置符号来表示,0 表示正数,1 表示负数。指数和尾数都为 0。+0.0 和 -0.0 都等于 0.0。由 0.0 的符号产生的差别,参考下面的浮点算术结果。

无穷大

$+\infty$ 和 $-\infty$ 值可通过设置符号来表示,0 表示正数,1 表示负数。指数为 255 ($2^8 - 1$),尾数为 0。

NaN

NaN(不是一个数字)在计算如 $0.0/0.0$, ∞/∞ , 或 $\infty - \infty$ 结果,不等于一个数字或无穷大时产生。指数为 255 ($2^8 - 1$),尾数为非 0。

注 没有 NaN 符号或尾数区域值(不是非 0)的定义。

浮点算术结果

四舍五入结果

当浮点算术精确结果中的数字个数超过内部处理表达式中的有效数字时,以下方法将用于结果的四舍五入。

如果结果近似于两个内部浮点表达式中的一个,使用较接近的值。如果结果介于两个内部浮点表达式的中间,结果将四舍五入,以便使尾数的最后一个数字为 0。

上溢出,下溢出和非法计算

上溢出将根据结果符号以正无穷或负无穷输出。下溢出将根据结果符号以正 0 或负 0 输出。

非法计算将导致 NaN,非法计算包括给一个数字加上带相反符号的无穷大,从一个数字减去带相反符号的无穷大,0 和无穷大相乘,0 和 0 相除,或无穷大与无穷大相除。

如果在浮点数转换为整数时出现上溢出,那么结果值可能不正确。

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。
- 如果任何操作数是 NaN,结果将为 NaN。
- 正 0 和负 0 在比较时以相等处理。
- 对一个或更多的 NaN 进行比较或相等测试时,!= 总为真,其它指令总为假。

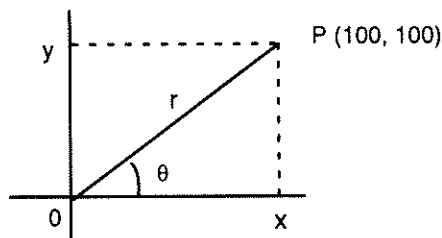
浮点计算结果

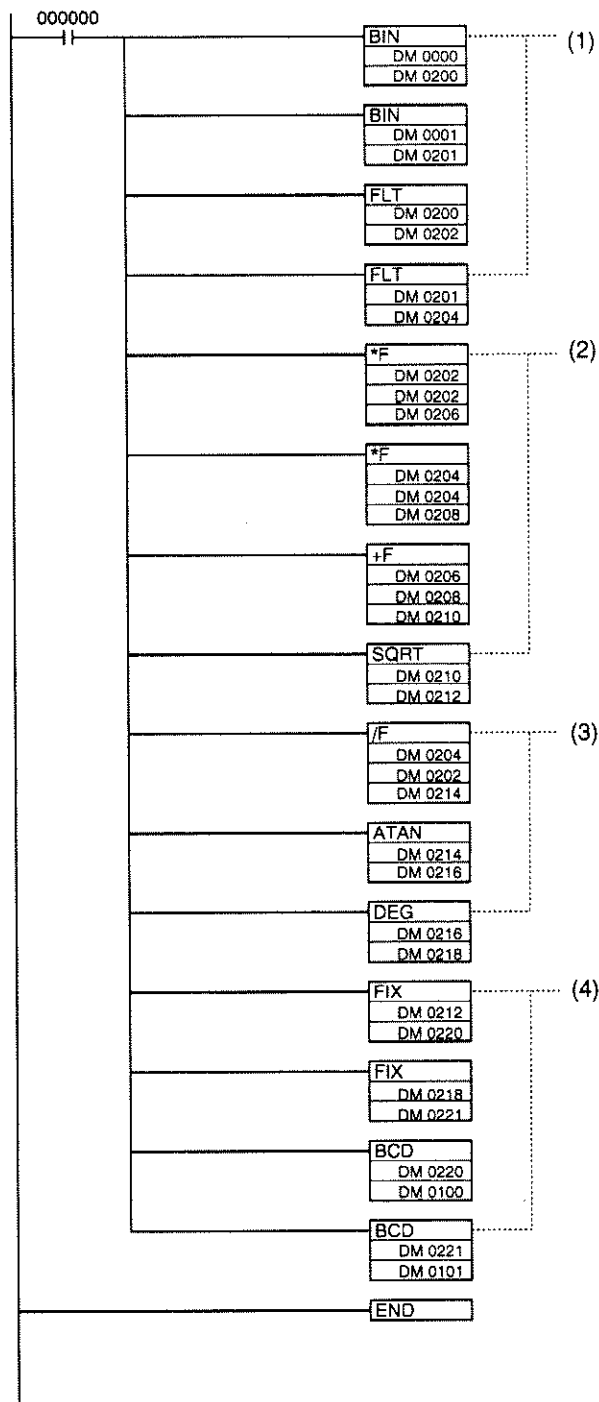
当结果的绝对值大于浮点数据能够表达的最大值,上溢出标志 (SR25404) 将变 ON,结果输出 $\pm \infty$ 。如果结果为正,输出 $+\infty$;如果为负,输出为 $-\infty$ 。

计算后指数(e)和尾数(f)都为 0 时,等于标志变 ON。当结果的绝对值小于浮点数据能够表达的最小值,计算结果也输出 0。这时下溢出标志 (SR25405) 变 ON。

例

在此程序例中,X 轴和 Y 轴坐标 (X, Y) 由 DM0000 和 DM0001 的 4 位 BCD 数的内容表示。从原点的距离 (r) 和角度 (θ , 用角度) 算出,并输出到 DM0100 和 DM0101。结果中,小数点后面的数被截去。

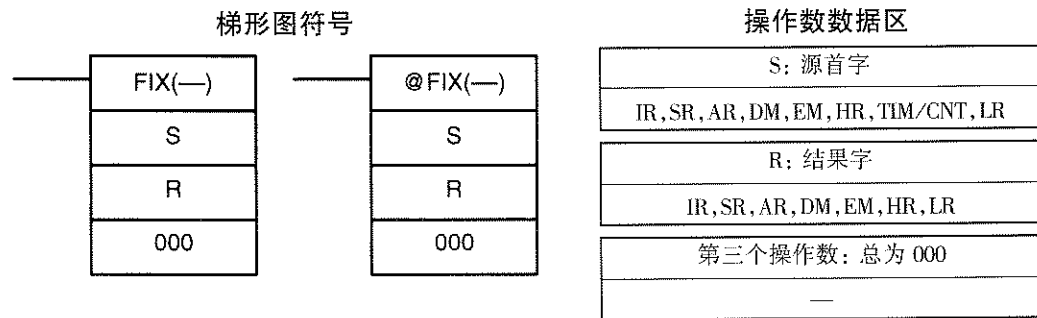




| | |
|--|---|
| 计算 | 例 |
| 距离 $r = \sqrt{x^2 + y^2}$ | 距离 $r = \sqrt{100^2 + 100^2} = 141.4214$ |
| 角度 $\theta = \tan^{-1}(\frac{y}{x})$ | 角度 $\theta = \tan^{-1}(\frac{100}{100}) = 45.0$ |
| DM 内容 | |
| DM 0000 0100 (BCD) x | DM 0100 0141 r |
| DM 0001 0100 (BCD) y | DM 0101 0045 θ |

1. 这部分程序把数据从 BCD 码转换成浮点数。
 - a) 从 DM0200 向上的数据区用作工作区。
 - b) 第 1 个 BIN(23) 用于暂时把 BCD 数据转换成二进制数据, 而 FLT(一) 用于把二进制数据转换成浮点数据。
 - c) 已经转换成浮点数据的 X 值输出到 DM0203 和 DM0202。
 - d) 已经转换成浮点数据的 Y 值输出到 DM0205 和 DM0204。
2. 为了找到距离 r, 浮点数学指令用于计算 $X^2 + Y^2$ 的平方根。结果作为浮点数据输出到 DM0213 和 DM0212 中。
3. 为了找到角度 θ , 浮点数学指令用于计算 $\tan^{-1}(Y/X)$, ATAN(一) 输出结果以弧度表示, 因此 DEG(一) 用于转换为度数。结果作为浮点数据输出到 DM0219 和 DM0218。
4. 数据从浮点数转换回 BCD。
 - a) 第 1 个 FIX(一) 用于暂时把浮点数据转换成二进制数据, 而 BCD(24) 用于把二进制数据转换为 BCD 数据。
 - b) 距离 r 输出到 DM0100。
 - c) 角度 θ 输出到 DM0101。

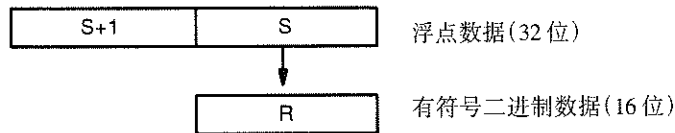
5-24-1 浮点到 16 位: FIX(一)



限制 S + 1 和 S 的内容必须是浮点数据并且整数部分必须在 - 32, 768 ~ 32, 767 的范围内。

DM6144 ~ DM6655 不能用于 R。

说明 当执行为 OFF 时, FIX(一) 不执行。当执行条件为 ON 时, FIX(一) 把 S + 1 和 S 中的 32 位浮点数 (IEEE754 格式) 的整数部分转换为 16 位有符号的二进制数据, 并把结果置于 R 中。



只转换了浮点数据的整数部分,小数部分被截去。浮点数据的整数部分必须在 - 32,768 ~ 32,767 之间。

转换例:

浮点值 3.5 转换为 3。

浮点值 - 3.5 转换为 - 3。

标志

RE: 间接寻址的 EM/DM 字不存在。

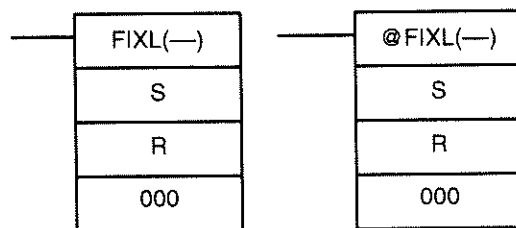
(* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界。)

如果 S + 1 和 S 的整数部分不在 - 32,768 ~ 32,767 范围内则置 ON。

EQ: 结果为 0000 时置 ON。

5-24-2 浮点数到 32 位: FIXL(—)

梯形图符号



操作数数据区

| |
|-------------------------------------|
| S: 源首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |
| 第三个操作数: 总为 000 |
| — |

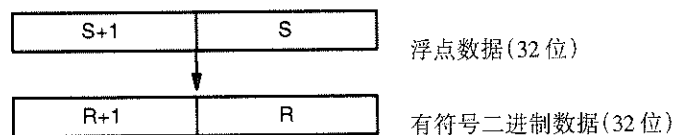
限制

S + 1 和 S 的内容必须是浮点数据,并且整数部分必须在 - 2,147,483,648 ~ 2,147,483,647 之间。

DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, FIXL(—) 不执行。当执行条件为 ON 时, FIXL(—) 把 S + 1 和 S 中的 32 位浮点数 (IEEE754 格式) 的整数部分转换成 32 位有符号的二进制数据,并将结果置于 R + 1 和 R 中。



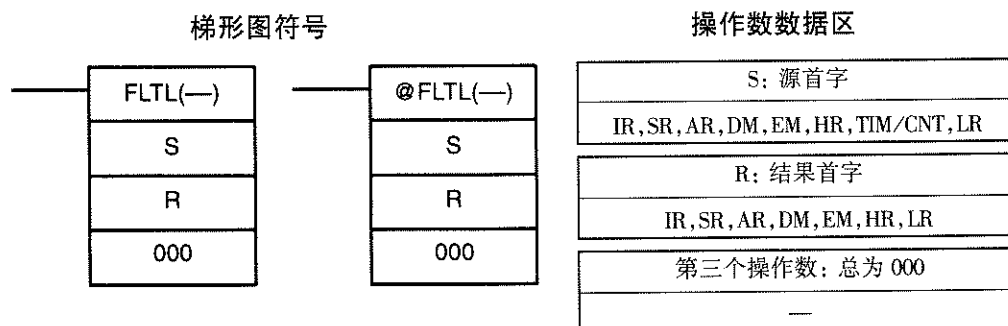
被转换的仅仅是浮点数据的整数部分,而小数部分则被舍去。(浮点数据的整数部分必须在 - 2,147,483,648—2,147,483,647 范围之内。)

转换例:

2,147,483,640.5 的浮点数转换为 2,147,483,640。

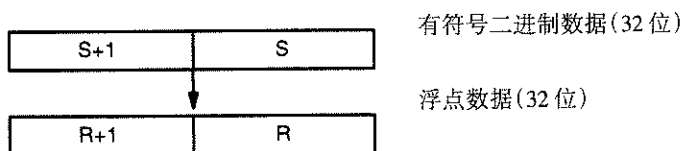
- 2,147,483,640.5 的浮点数转换为 - 2,147,483,640。

5-24-4 32 位到浮点: FLTL(—)



限制 如果一个绝对值大于 16,777,215(可用 24 位表示的最大值)的数被转换,那么结果将不准确。
DM6143 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, FLTL(—) 不执行。当执行条件为 ON 时, FLTL(—) 把 S+1 和 S 中的 32 位有符号二进制值转换为 32 位浮点数据 (IEEE754 格式), 并把结果置于 R+1 和 R 中。浮点结果的小数点后面加了一个 0。



在范围 -2,147,483,648 ~ 2,147,483,647 之内的有符号二进制数据可指定在 S+1 和 S 中。浮点值有 24 个有效的二进制数字(或位)。如果一个大于 16,777,215 的数(可用 24 位表示的最大值)用 FLTL(—) 转换,那么结果将不准确。

转换例:

有符号二进制值 16,777,215 转换为 16,777,215.0。

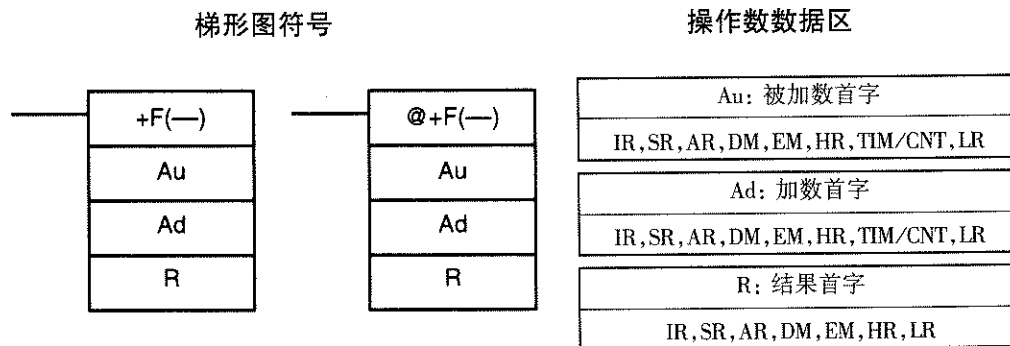
有符号二进制值 -16,777,215 转换为 -16,777,215.0。

标志 ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

EQ: 结果的指数和尾数为 0 时置 ON。

5-24-5 浮点加: +F(—)

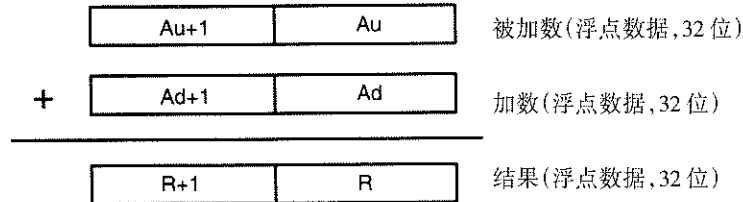


限制 被加数 (Au+1 和 Au) 和加数 (Ad+1 和 Ad) 数据必须是 IEEE754 浮点数据格式。

DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, +F(—)不执行。当执行条件为 ON 时, +F(—)把 Ad + 1 和 Ad 中的 32 位浮点数与 Au + 1 和 Au 中的 32 位浮点数相加, 并把结果置于 R + 1 和 R 中。(浮点数据必须是 IEEE754 格式。)



如果结果的绝对值大于可用浮点数表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果将以 $\pm \infty$ 输出。

如果结果的绝对值小于可用浮点数表示的最小值, 那么下溢标志 (SR25405) 将变 ON, 并且结果将以 0 输出。

被加数和加数的不同组合将产生如下表所示的结果。

| 加数 | 被加数 | | | | NaN |
|-----------|-----------|-----------|-----------|-----------|------|
| | 0 | 数字 | $+\infty$ | $-\infty$ | |
| 0 | 0 | 数字 | $+\infty$ | $-\infty$ | 见注2。 |
| 数字 | 数字 | 见注1。 | $+\infty$ | $-\infty$ | |
| $+\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | 见注2。 | |
| $-\infty$ | $-\infty$ | $-\infty$ | 见注2。 | $-\infty$ | |
| NaN | | | | | |

- 注 1. 结果可能为 0(包括下溢出), 一个数字, $+\infty$ 或 $-\infty$ 。
 2. 错误标志将变 ON, 并且指令将不执行。

标志

RE: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

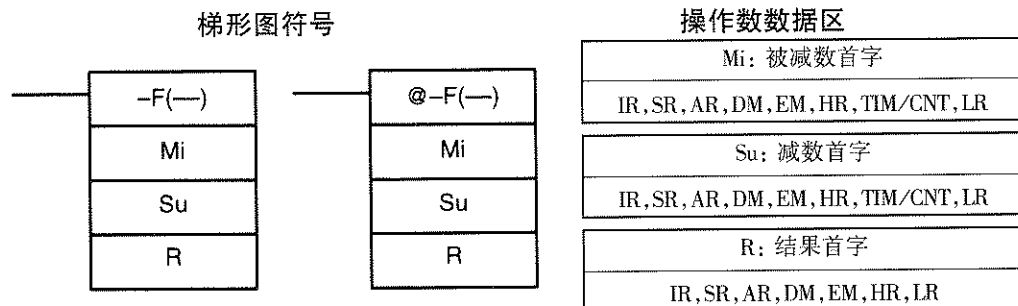
如果被加数或 加数不是浮点数据, 则置 ON。

EQ: 如果结果的指数和尾数都为 0, 则置 ON。

OF: 如果结果的绝对值太大而不能表示为 32 位浮点数则置 ON。(结果将以 $\pm \infty$ 输出)

UF: 如果结果的绝对值太小而不能表示为 32 位浮点数则置 ON。(结果将以 0 输出。)

5-24-6 浮点减: -F(—)

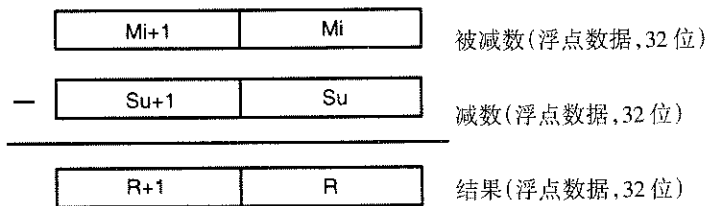


限制

被减数(Mi + 1 和 Mi)和减数(Su + 1 和 Su)必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, -F(—)不执行。当执行条件为 ON 时, -F(—)用 Mi + 1 和 Mi 中的 32 位浮点数减去 Su + 1 和 Su 中的 32 位浮点数, 并且把结果置于 R + 1 和 R 中。(浮点数必须是 IEEE754 格式。)



如果结果的绝对值大于可用浮点数表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果将以 $\pm \infty$ 输出。

如果结果的绝对值小于可用浮点数表示的最小值, 那么下溢标志 (SR25405) 将变 ON, 并且结果将以 0 输出。

被减数和减数的不同组合将产生如下表所示的结果。

| 减数 | 被减数 | | | | NaN |
|-----------|-----------|-----------|-----------|-----------|------|
| | 0 | 数字 | $+\infty$ | $-\infty$ | |
| 0 | 0 | 数字 | $+\infty$ | $-\infty$ | 见注2。 |
| 数字 | 数字 | 见注1。 | $+\infty$ | $-\infty$ | |
| $+\infty$ | $-\infty$ | $-\infty$ | 见注2。 | $-\infty$ | |
| $-\infty$ | $+\infty$ | $+\infty$ | $+\infty$ | 见注2。 | |
| NaN | | | | | |

- 注 1. 结果可能为 0(包括下溢出), 一个数字, $+\infty$ 或 $-\infty$ 。
2. 错误标志变 ON, 并且指令将不执行。

标志

RE: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

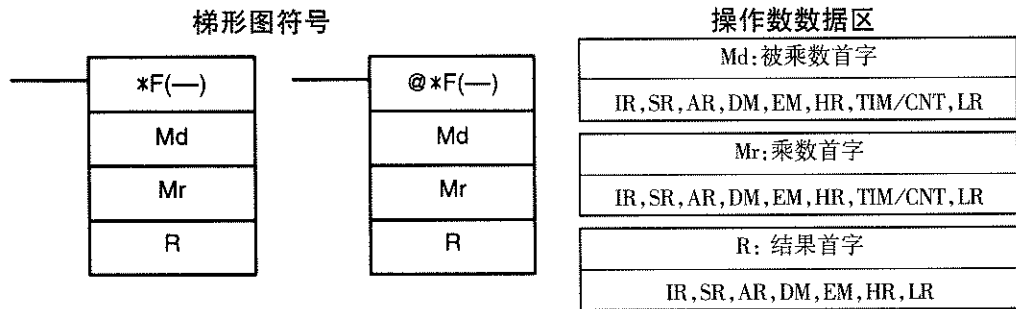
如果被减数或减数不是浮点数据, 则置 ON。

EQ: 如果结果的指数和尾数都为 0, 则置 ON。

OF: 如果结果的绝对值太大而不能表示为 32 位浮点数,则置 ON。(结果将以 $\pm \infty$ 输出。)

UR: 如果结果的绝对值太小而不能表示为 32 位浮点数,则置 ON。(结果将以 0 输出。)

5-24-7 浮点乘: *F(—)

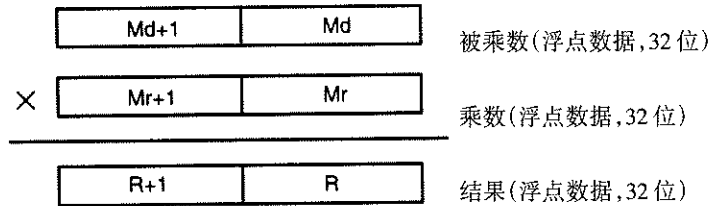


限制

被乘数(Md + 1 和 Md)和乘数(Mr + 1 和 Mr)必须是 IEEE754 浮点数据格式。DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, *F(—) 不执行。当执行条件为 ON 时, *F(—) 把 Md + 1 和 Md 中的 32 位浮点数乘以 Mr + 1 和 Mr 中的 32 位浮点数,并把结果置于 R + 1 和 R 中。(浮点数据必须是 IEEE754 格式。)



如果结果的绝对值大于可用浮点数表示的最大值,那么上溢标志 (SR25404) 将变 ON,并且结果将以 $\pm \infty$ 输出。

如果结果的绝对值小于可用浮点数表示的最小值,那么下溢标志 (SR25405) 将变 ON,并且结果将以 0 输出。

被乘数和乘数的不同组合将产生如下表所示的结果。

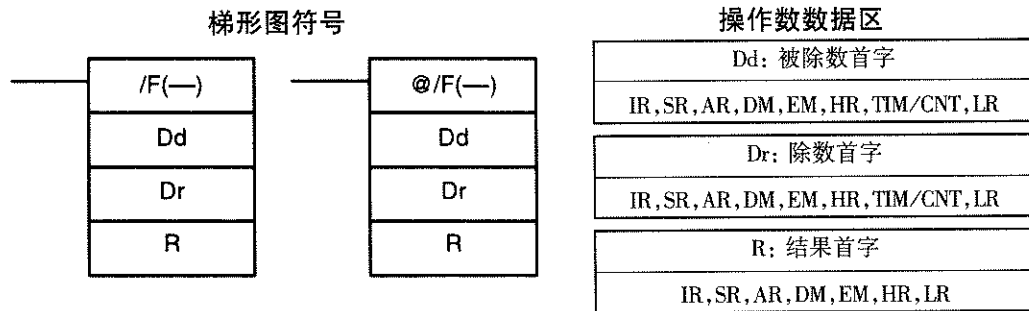
| 乘数 | 被乘数 | | | | NaN |
|-----------|------|-------------|-------------|-------------|------|
| | 0 | 数字 | $+\infty$ | $-\infty$ | |
| 0 | 0 | 0 | 见注2。 | 见注2。 | 见注2。 |
| 数字 | 0 | 见注1。 | $+/-\infty$ | $+/-\infty$ | |
| $+\infty$ | 见注2。 | $+/-\infty$ | $+\infty$ | $-\infty$ | |
| $-\infty$ | 见注2。 | $+/-\infty$ | $-\infty$ | $+\infty$ | |
| NaN | | | | | |

- 注 1. 结果可能为 0(包括下溢出),一个数字, $+\infty$ 或 $-\infty$ 。
- 2. 错误标志将变 ON,并且指令将不执行。

标志

- ER: 间接寻址的 EM/DM 字不存在。
(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
如果被乘或乘数不是浮点数据, 则置 ON。
- EQ: 如果结果的指数和尾数为 0, 则置 ON。
- OF: 如果结果的绝对值太大而不能表示为 32 位浮点数则置 ON。(结果将以 $\pm \infty$ 输出。)
- UF: 如果结果的绝对值太小而不能表示为 32 位浮点数则置 ON。(结果将以 0 输出。)

5-24-8 浮点除: /F(—)

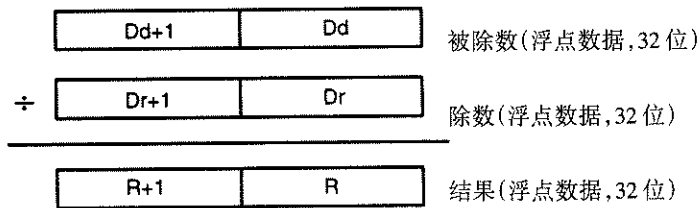


限制

被除数 (Dd + 1 和 Dd) 和除数 (Dr + 1 和 Dr) 必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, /F(—) 不执行。当执行条件为 ON 时, /F(—) 用 Dd + 1 和 Dd 中的 32 位浮点数除以 Dr + 1 和 Dr 中的 32 位浮点数, 并把结果置于 R + 1 和 R 中。(浮点数据必须是 IEEE754 格式。)



如果结果的绝对值大于可用浮点数表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果将以 $\pm \infty$ 输出。

如果结果的绝对值小于可用浮点数表示的最小值, 那么下溢标志 (SR25405) 将变 ON, 并且结果将以 0 输出。

被除数和除数的不同组合将产生如下表所示的结果。

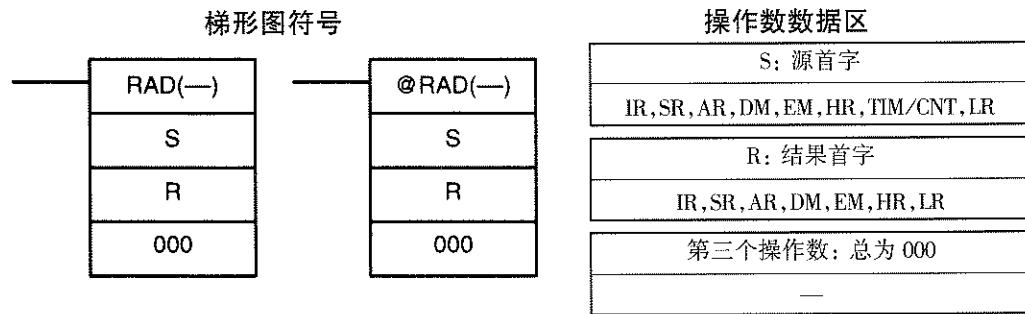
| 除数 | 被除数 | | | | NaN |
|-----|------|---------|---------|---------|------|
| | 0 | 数字 | + ∞ | - ∞ | |
| 0 | 见注3。 | + / - ∞ | + ∞ | - ∞ | 见注3。 |
| 数字 | 0 | 见注1。 | + / - ∞ | + / - ∞ | |
| + ∞ | 0 | 见注2。 | 见注3。 | 见注3。 | |
| - ∞ | 0 | 见注2。 | 见注3。 | 见注3。 | |
| NaN | | | | | |

- 注 1. 结果可能为 0(包括下溢出), 一个数字, $+\infty$ 或 $-\infty$
 2. 对下溢出, 结果将为 0。
 3. 错误标志为 ON 并且指令不执行。

标志

ER: 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 如果被除或除数不是浮点数据, 则置 ON。
 EQ: 如果结果的指数和尾数为 0, 则置 ON。
 OF: 如果结果的绝对值太大而不能表示为 32 位浮点数则置 ON。(结果将以 $\pm\infty$ 输出。)
 UF: 如果结果的绝对值太小而不能表示为 32 位浮点数则置 ON。(结果将以 0 输出。)

5-24-9 度到弧度: RAD(—)

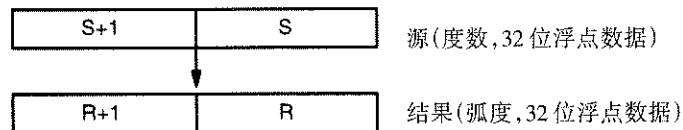


限制

S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
 DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, RAD(—) 不执行。当执行条件为 ON 时, RAD(—) 把 S+1 和 S 中的 32 位浮点数从角度转换为弧度, 并且把结果置于 R+1 和 R 中。
 (浮点数据必须是 IEEE754 格式。)



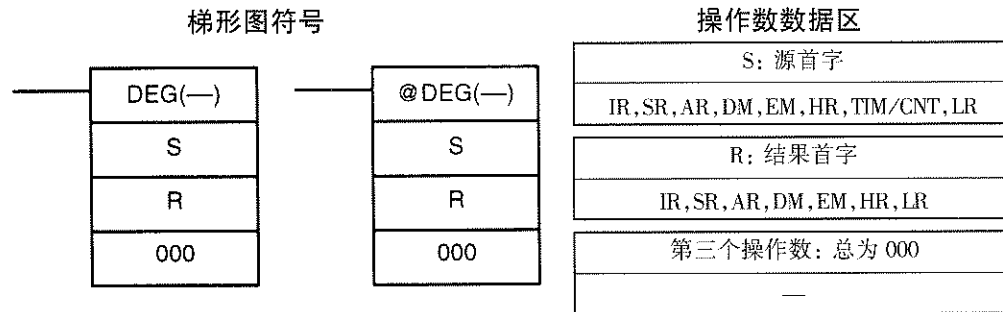
用以下公式把度数转换为弧度: $度 \times \pi / 180 = 弧度$
 如果结果的绝对值大于可用浮点数表示的最大值, 那么上溢出标志 (SR25404) 将变 ON, 并且结果以 $\pm\infty$ 输出。
 如果结果的绝对值小于可用浮点数表示的最小值, 那么下溢出标志 (SR25405) 将变 ON, 并且结果将以 0 输出。

标志

ER: 间接寻址的 EM/DM 字不存在。
 (*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 如果源数据不是浮点数则置 ON。

- EQ: 如果结果的指数和尾数都为 0,则置 ON。
- OF: 如果结果的绝对值太大而不能表示为 32 位浮点数则置 ON。(结果将以 $\pm \infty$ 输出。)
- UF: 如果结果的绝对值太小而不能表示为 32 位浮点数则置 ON。(结果将以 0 输出。)

5-24-10 弧度到度: DEG(—)

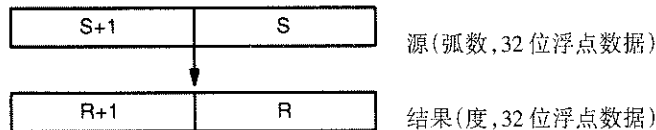


限制

S + 1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, DEG(—) 不执行。当执行条件为 ON 时, DEG(—) 把 S + 1 和 S 中的 32 位浮点数从弧度转换为度, 并且把结果置于 R + 1 和 R 中。(浮点数据必须是 IEEE754 格式。)



用以下公式把弧度转换为度:

$$\text{弧度} \times 180/\pi = \text{度}$$

如果结果的绝对值大于可用浮点数表示的最大值, 那么上溢出标志 (SR25404) 将变 ON, 并且结果以 $\pm \infty$ 输出。

如果结果的绝对值小于可用浮点数表示的最小值, 那么下溢出标志 (SR25405) 将变 ON, 并且结果将以 0 输出。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

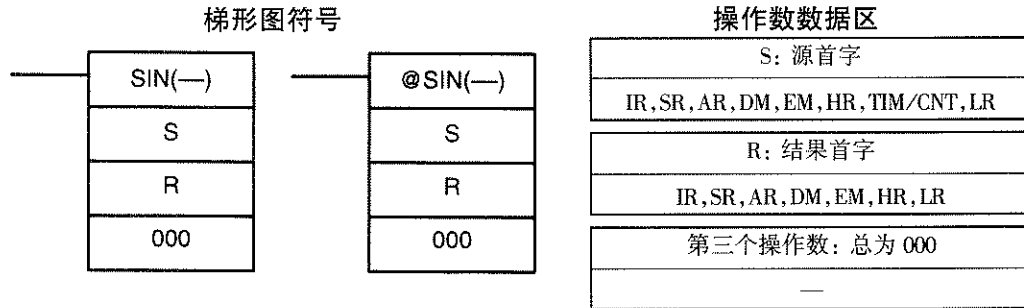
如果源数据不是浮点数则置 ON。

EQ: 如果结果的指数和尾数都为 0, 则置 ON。

OF: 如果结果的绝对值太大而不能表示为 32 位浮点数则置 ON。(结果将以 $\pm \infty$ 输出。)

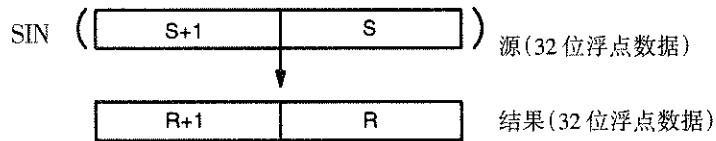
UF: 如果结果的绝对值太小而不能表示为 32 位浮点数则置 ON。(结果将以 0 输出。)

5-24-11 正弦: SIN(—)



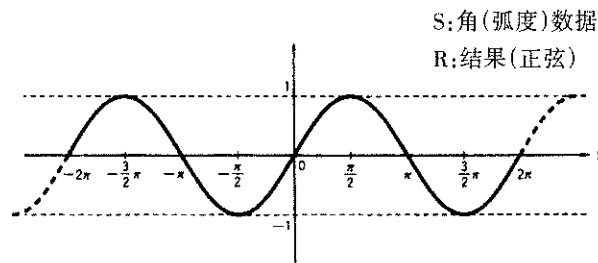
限制 S + 1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, SIN(—) 不执行。当执行条件为 ON 时, SIN(—) 计算 S + 1 和 S 中用 32 位浮点数表示的角(弧度)的正弦, 并把结果置于 R + 1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



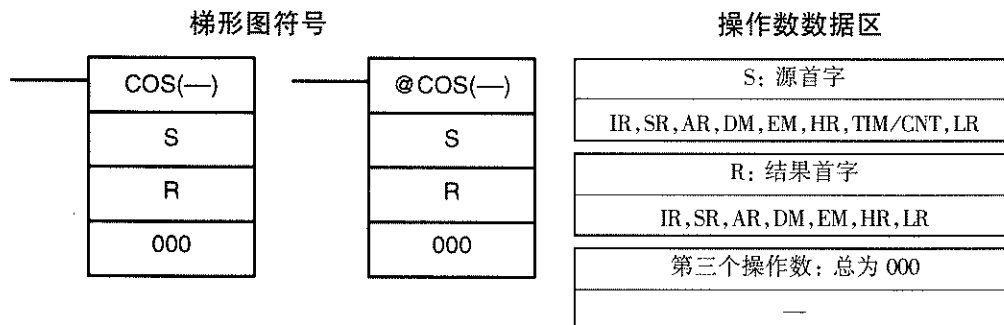
在 S + 1 和 S 中用弧度指定所求的角 (- 65, 535 ~ 65, 535)。如果角的绝对值超过 65, 535, 将会出现错误并且指令也将不执行。有关把度转换为弧度的信息, 参见 5-24-9 度到弧度: RAD(—)。

下图显示角和结果之间的关系。



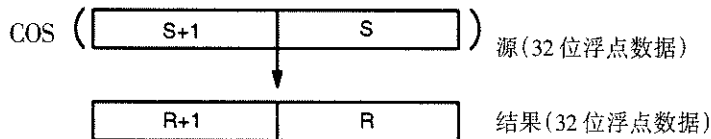
标志 ER: 间接寻址的 EM/DM 字不存在。
(* EM / * DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
如果源数据不是浮点数据则置 ON。
如果源数据的绝对值超过 65, 535 则置 ON。
EQ: 如果结果的指数和尾数都为 0, 则置 ON。

5-24-12 余弦: COS(—)



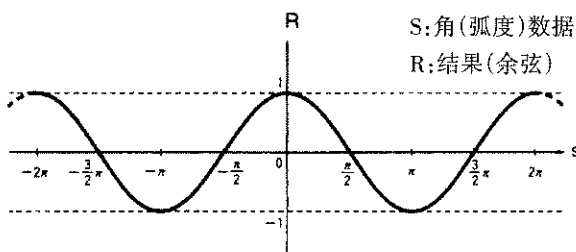
限制 S + 1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, COS(—) 不执行。当执行条件为 ON 时, COS(—) 计算 S + 1 和 S 中用 32 位浮点数表示的角(弧度)的余弦, 并把结果置于 R + 1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



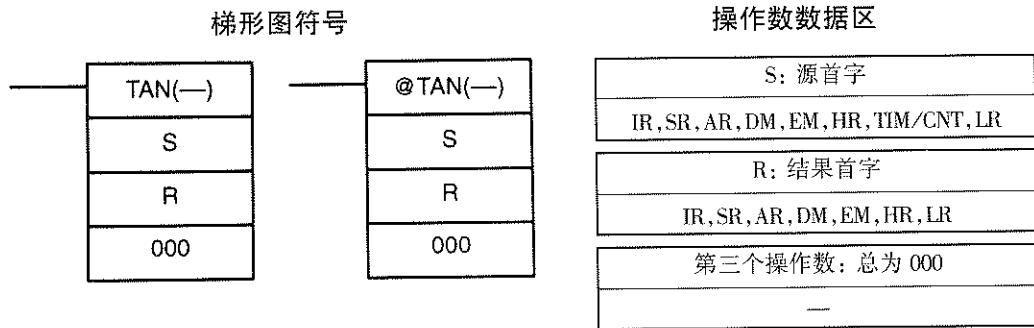
在 S + 1 和 S 中用弧度指定所求的角 (- 65, 535 ~ 65, 535)。如果角的绝对值超过 65, 535, 将会出现错误并且指令也将不执行。有关把度转换为弧度的信息, 参见 5-24-9 度到弧度: RAD(—)。

下图显示角和结果之间的关系。



标志 ER: 间接寻址的 EM/DM 字不存在。
(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
如果源数据不是浮点数据则置 ON。
如果源数据的绝对值超过 65, 535 则置 ON。
EQ: 如果结果的指数和尾数都为 0, 则置 ON。

5-24-12 正切: TAN(—)

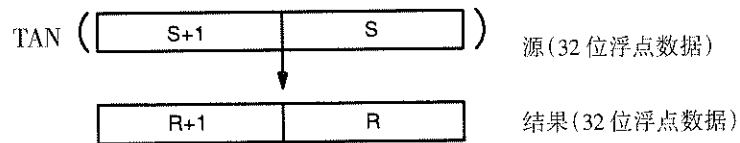


限制

S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

说明

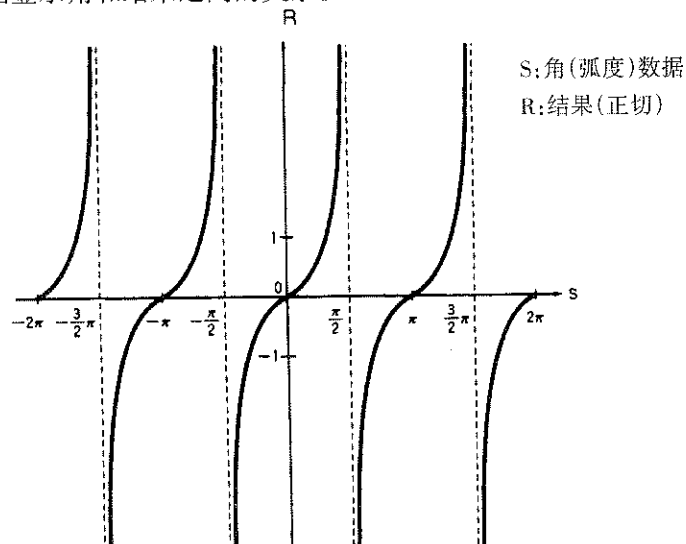
当执行条件为 OFF 时, TAN(—) 不执行。当执行条件为 ON 时, TAN(—) 计算 S+1 和 S 中用 32 位浮点数表示的角 (弧度) 的正切, 并把结果置于 R+1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



在 S+1 和 S 中用弧度指定所求的角 (- 65, 535 ~ 65, 535)。如果角的绝对值超过 65, 535, 那么将会出现错误, 并且指令也将不执行。有关把度转换为弧度的信息, 参见 5-24-9 度到弧度: RAD(—)。

如果结果的绝对值大于可用浮点数据表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果将以 ± ∞ 输出。

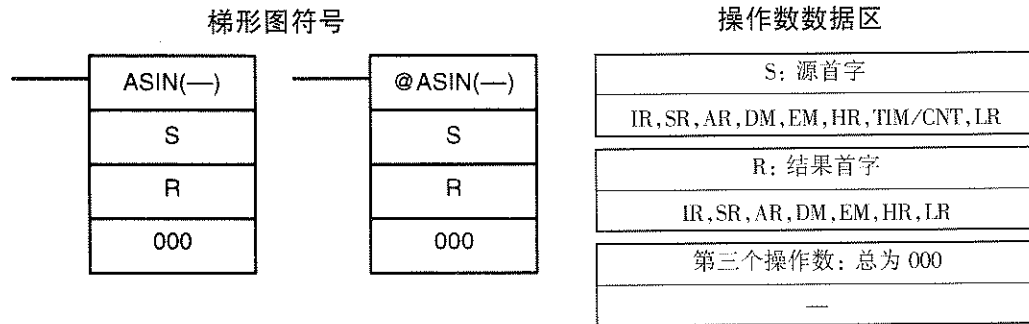
下图显示角和结果之间的关系。



标志

ER:间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)
 如果源数据不是浮点数据则置 ON。
 如果源数据的绝对值超过 65,535 则置 ON。
 EQ: 如果结果的指数和尾数都为 0,则置 ON。

5-24-14 反正弦: ASIN(—)

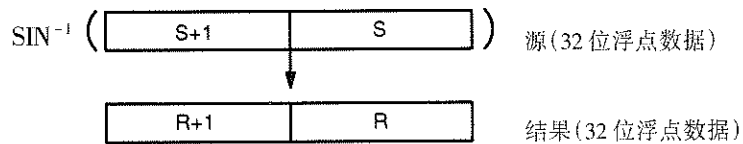


限制

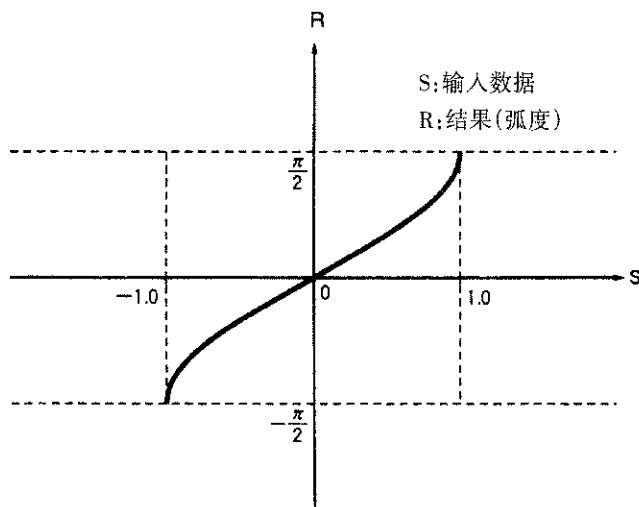
S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
 DM6143 ~ DM6655 不能用于 R。

说明

ASIN(—) 计算一个 32 位浮点数的反正弦,并把结果置于指定的结果字中。
 (反正弦函数是正弦函数的反函数;它返回产生所给的正弦值为 -1 到 1 之间的角度。)
 当执行条件为 OFF 时,ASIN(—)不执行。当执行条件为 ON 时,ASIN(—)计算 S+1 和 S 中的用 32 位浮点数表示正弦值的角(弧度),并把结果置于 R+1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



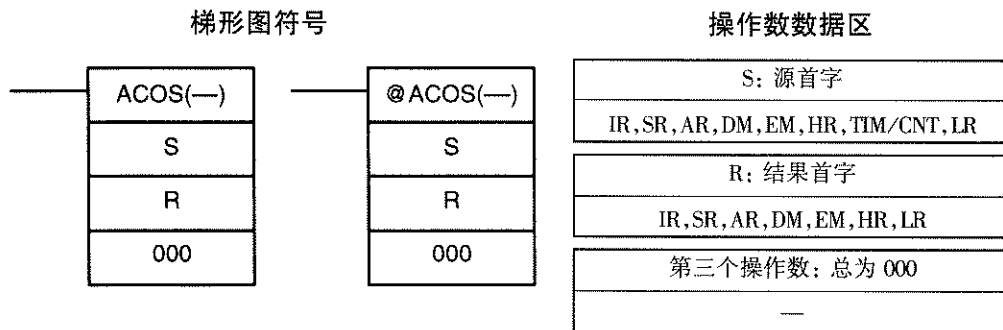
源数据必须在 -1.0 和 1.0 之间。如果源数据的绝对值超过 1.0,那么将会产生错误,并且指令不执行。
 结果以 $-\pi/2$ 到 $\pi/2$ 之间的角(弧度)形式输出到字 R+1 和 R。
 下图显示输入数据和结果之间的关系。



标志

ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 如果源数据不是浮点数据则置 ON。
 如果源数据的绝对值超过 1.0 则置 ON。
 EQ: 如果结果的指数和尾数都为 0, 则置 ON。

5-24-15 反余弦: ACOS(—)

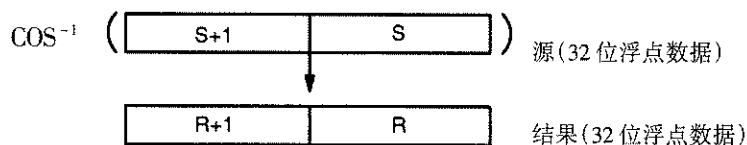


限制

S + 1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
 DM6143 ~ DM6655 不能用于 R。

说明

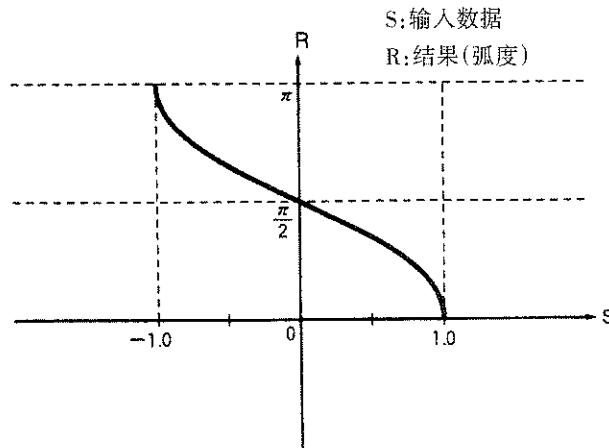
ACOS(—) 计算一个 32 位浮点数的反余弦, 并把结果置于指定的结果字中。
 (反余弦函数是余弦函数的反函数; 它返回产生所给的余弦值为 -1 到 1 之间的角度。)
 当执行条件为 OFF 时, ACOS(—) 不执行。当执行条件为 ON 时, ACOS(—) 计算 S + 1 和 S 中用 32 位浮点数表示的余弦值的角(弧度), 并把结果置于 R + 1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



源数据必须在 -1.0 和 1.0 之间。如果源数据的绝对值超过 1.0,那么将会产生一个错误,并且指令不执行。

结果以 $0 \sim \pi$ 之间的角(弧度)形式输出到字 R+1 和 R。

下图显示输入数据和结果之间的关系。



标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)

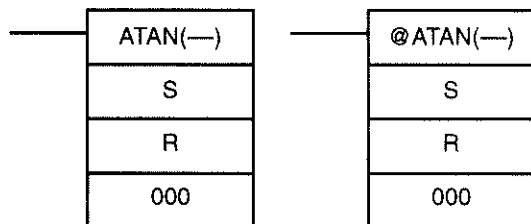
如果源数据不是浮点数据则置 ON。

如果源数据的绝对值超过 1.0 则置 ON。

EQ: 如果结果的指数和尾数都为 0,则置 ON。

5-24-16 反正切: ATAN(—)

梯形图符号



操作数数据区

| |
|-------------------------------------|
| S: 源首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |
| 第三个操作数: 总为 000 |
| — |

限制

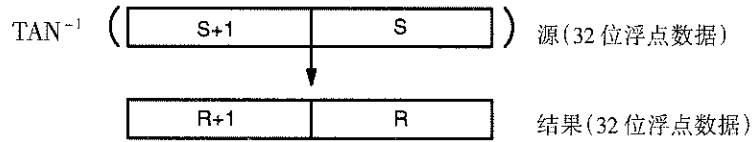
S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。

DM6143 ~ DM6655 不能用于 R。

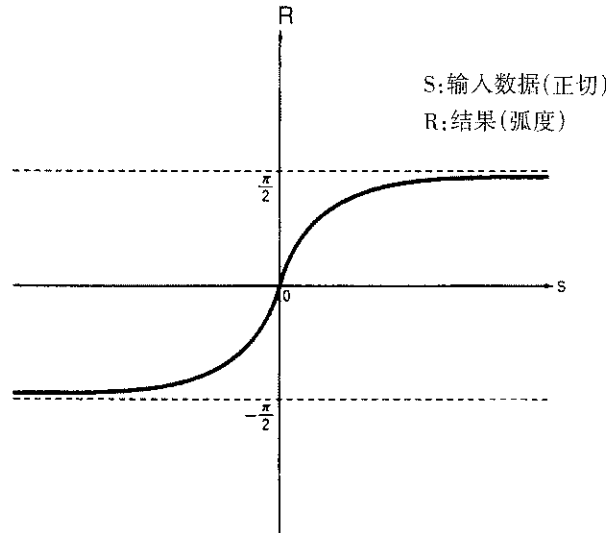
说明

ATAN(—)计算一个 32 位浮点数的反正切,并把结果置于指定的结果字中。(反正切函数是正切函数的反函数;它返回产生所给的正切值的角。)

当执行条件为 OFF 时,ATAN(—)不执行。当执行条件为 ON 时,ATAN(—)计算 S+1 和 S 中用 32 位浮点数表示的正切值的角(弧度),并把结果置于 R+1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



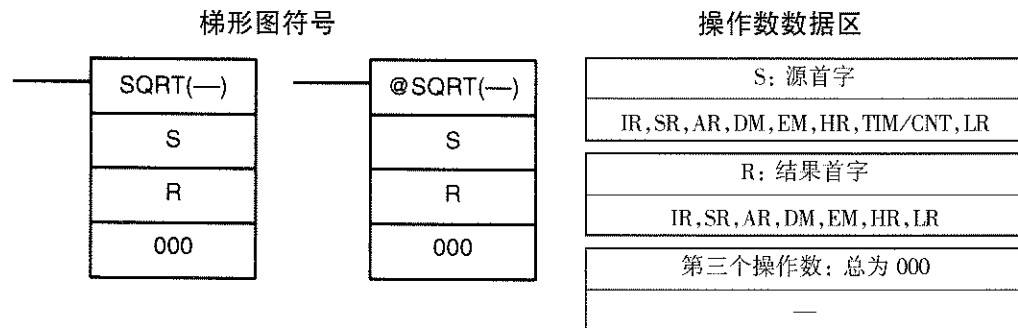
结果以 $-\pi/2 \sim \pi/2$ 之间的角(弧度)形式输出到字 R+1 和 R。
 下图显示输入数据和结果之间的关系。



标志

- ER: 间接寻址的 EM/DM 字不存在。
 (* EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
- 如果源数据不是浮点数据则置 ON。
- EQ: 如果结果的指数和尾数都为 0, 则置 ON。

5-24-17 平方根: SQRT(—)

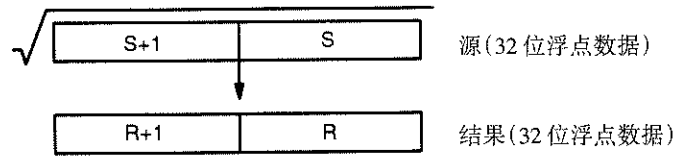


限制

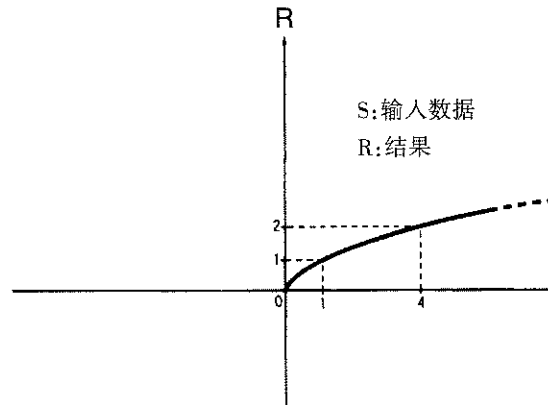
- S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
- DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, SQRT(—) 不执行。当执行条件为 ON 时, SQRT(—) 计算 S+1 和 S 中的 32 位浮点数的平方根, 并把结果置于 R+1 和 R 中。(浮点源数据必须是 IEEE754 格式。)



源数据必须是正值;如果是负的,那么将会产生错误,并且指令不执行。
 如果结果的绝对值大于可用浮点数据表示的最大值,那么上溢标志 (SR25404)将变 ON,并且结果以 $\pm \infty$ 输出。
 下图显示输入数据和结果之间的关系。

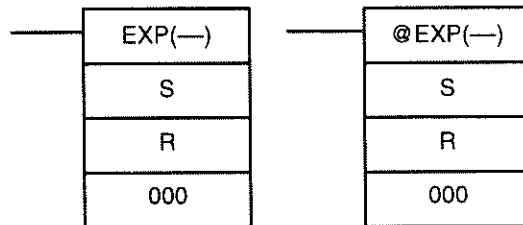


标志

- ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或已超出 EM/DM 区边界。)
 如果源数据不是浮点数据则置 ON。
 如果源数据为负数则置 ON。
- EQ: 如果结果的指数和尾数都为 0,则置 ON。
- OF: 如果结果的绝对值太大而不能表示为 32 位浮点值时置 ON(结果将以 $+\infty$ 输出。)

5-24-18 指数: EXP(—)

梯形图符号



操作数数据区

| |
|-------------------------------------|
| S: 源首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, LR |
| 第三个操作数: 总为 000 |
| — |

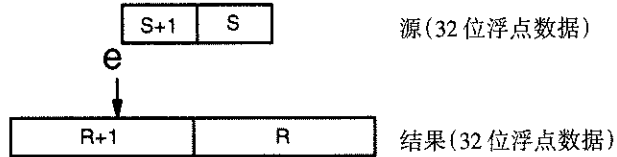
限制

S + 1 和 S 中的源数据必须是 IEEE754 浮点数据格式。

DM6143 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时, EXP(—) 不执行。当执行条件为 ON 时, EXP(—) 计算 S+1 和 S 中 32 位浮点数的自然指数(底为 e), 并把结果置于 R+1 和 R 中。换句话说, EXP(—) 计算 e^x (x = 源数据), 并且将结果置于 R+1 和 R。

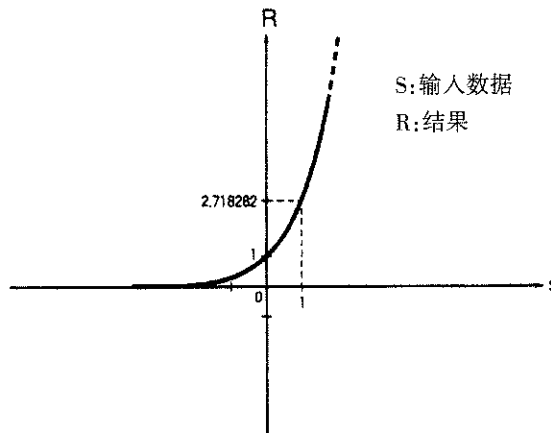


如果结果的绝对值大于可用浮点数据表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果以 $+\infty$ 输出。

如果结果的绝对值小于可用浮点数据表示的最小值, 那么下溢标志 (SR25405) 将变 ON, 并且结果以 0 输出。

注 常数 e 为 2.718282

下图显示输入数据和结果之间的关系。



标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

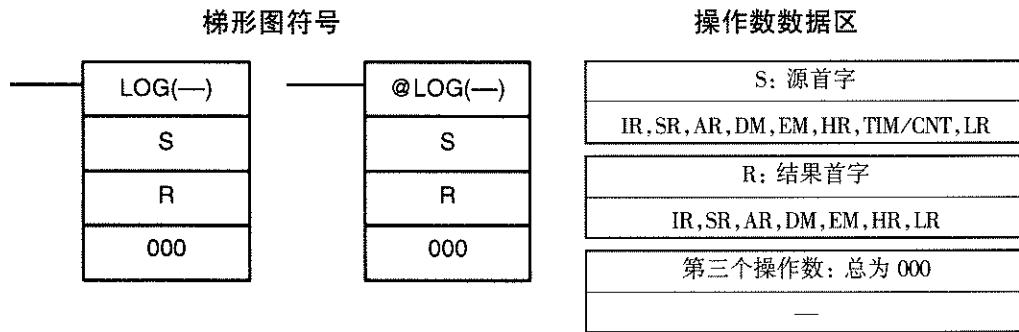
如果源数据不是浮点数据则置 ON。

EQ: 如果结果的指数和尾数都为 0, 则置 ON。

OF: 如果结果的绝对值太大而不能表示为 32 位浮点值时置 ON (结果将以 $+\infty$ 输出。)

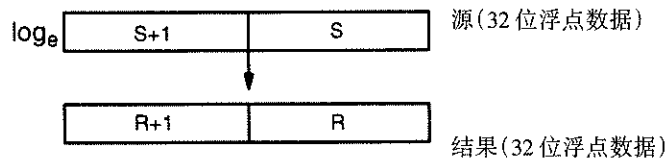
UF: 如果结果的绝对值太小而不能表示为 32 位浮点值时置 ON (结果将以 0 输出。)

5-24-19 对数: LOG(—)



限制 S+1 和 S 中的源数据必须是 IEEE754 浮点数据格式。
DM6143 ~ DM6655 不能用于 R。

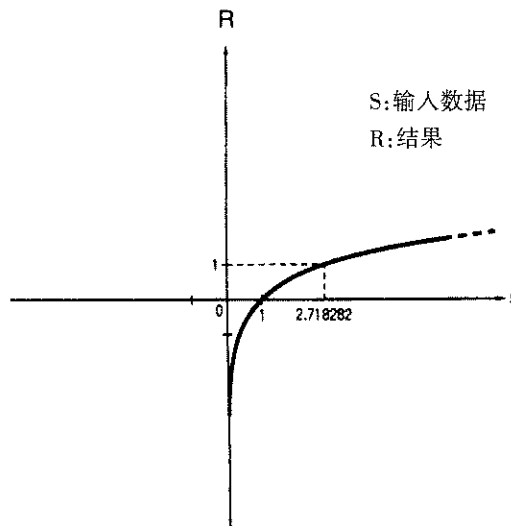
说明 当执行条件为 OFF 时, LOG(—) 不执行。当执行条件为 ON 时, LOG(—) 计算 S+1 和 S 中 32 位浮点数的自然对数(底为 e), 并把结果置于 R+1 和 R 中。



源数据必须是正值; 如果为负, 则会出现错误, 并且指令不执行。
如果结果的绝对值大于可用浮点数据表示的最大值, 那么上溢标志 (SR25404) 将变 ON, 并且结果以 $\pm \infty$ 输出。

注 常数 e 为 2.718282

下图显示输入数据和结果之间的关系。



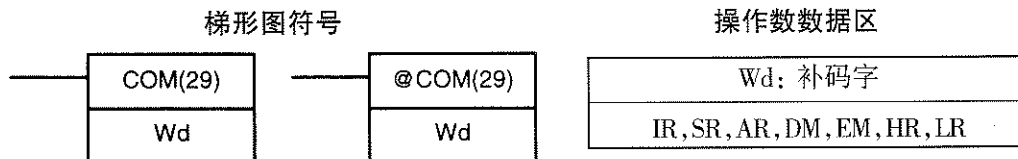
标志 ER: 间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
如果源数据不是浮点数据则置 ON。

EQ: 如果结果的指数和尾数都为 0, 则置 ON。

OF: 如果结果的绝对值太大而不能表示为 32 位浮点值时置 ON (结果将以 $+\infty$ 输出。)

5-25 逻辑指令

5-25-1 补码—COM(29)



限制
说明

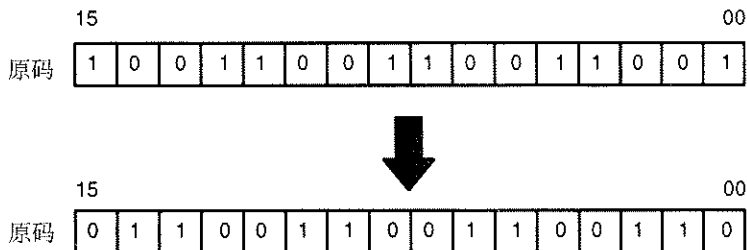
DM6144 ~ DM6655 不能用于 Wd。

当执行条件为 OFF 时, COM(29) 不执行, 当执行条件为 ON 时, COM(29) 清除 Wd 中所有的 ON 位, 并置位所有 OFF 位。

注意

若使用 COM(29) 的非微分格式, 则每循环都将对 Wd 求补码。使用微分格式 (@COM(29)) 或将 COM(29) 与 DIFU(13) 或 DIFD(14) 组合只作一次求补运算。

例



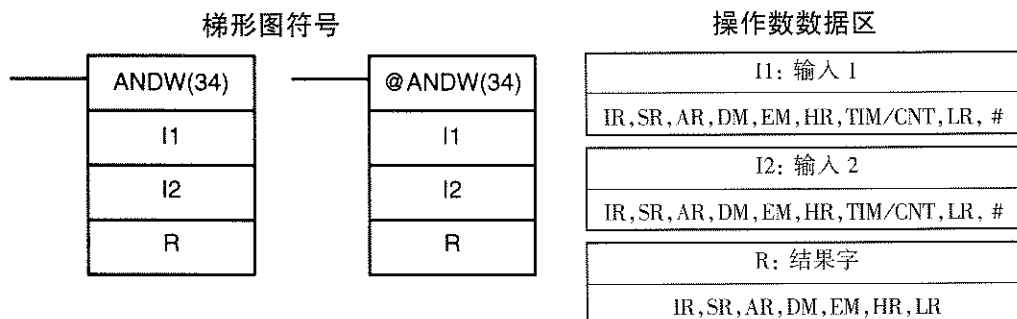
标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

EQ: 当结果为零时置 ON。

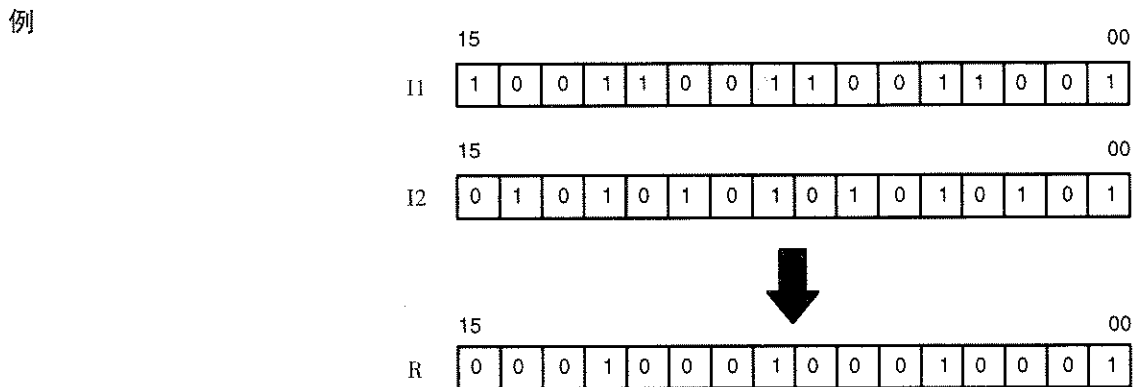
5-25-2 逻辑与—ANDW(34)



限制

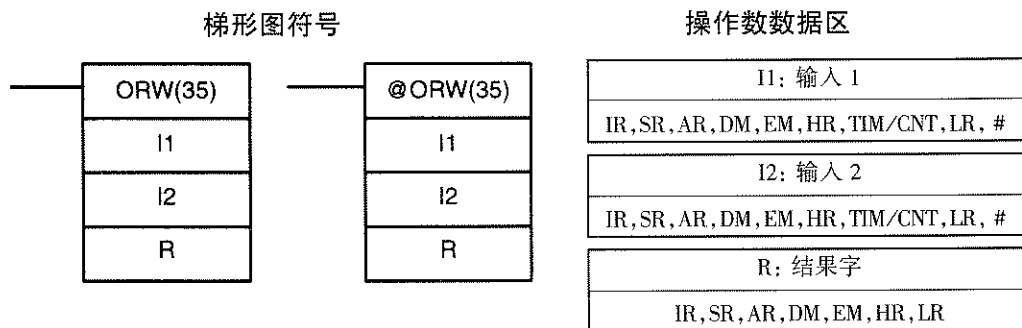
DM6144 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, ANDW(34) 不执行。当执行条件为 ON 时, ANDW(34) 将 I1 和 I2 的内容按位求逻辑与, 并将结果置于 R。

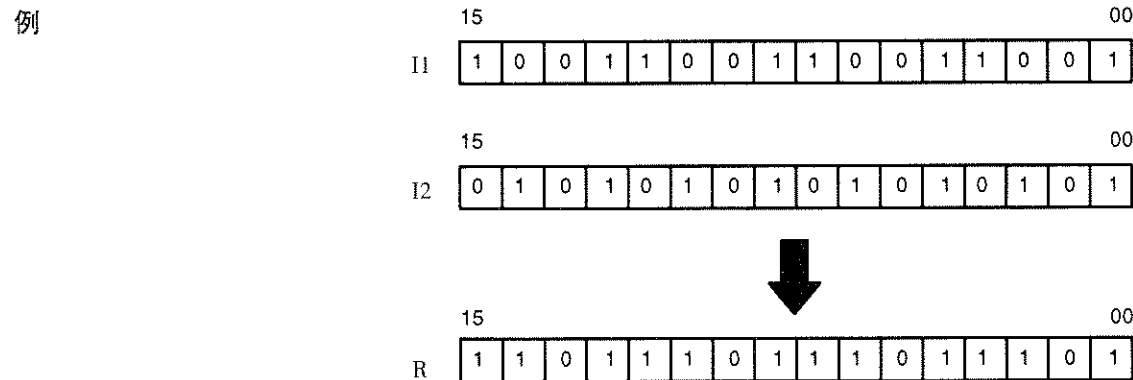


标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 当结果为零时置 ON。

5-25-3 逻辑或—ORW(35)



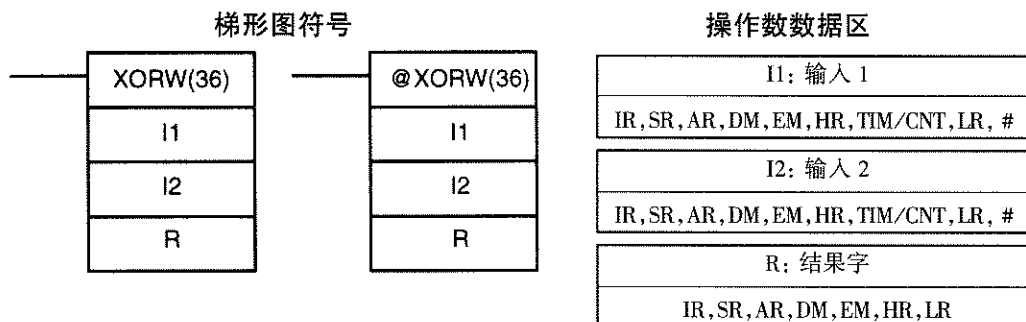
限制 DM6144 ~ DM6655 不能用于 R。
说明 当执行条件为 OFF 时, ORW(35) 不执行。当执行条件为 ON 时, ORW(35) 将 I1 和 I2 的内容按位求逻辑或, 并将结果置于 R。



标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)

EQ: 当结果为 0 时置 ON。

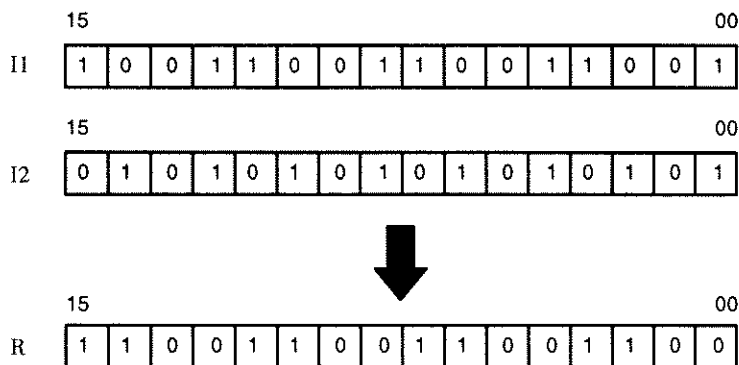
5-25-4 异或—XORW(36)



限制
说明

DM6144 ~ DM6655 不能用于 R。
当执行条件为 OFF 时, XORW(36) 不执行。当执行条件为 ON 时, XORW(36) 将 I1 和 I2 的内容按位求异或, 并将结果置于 R。

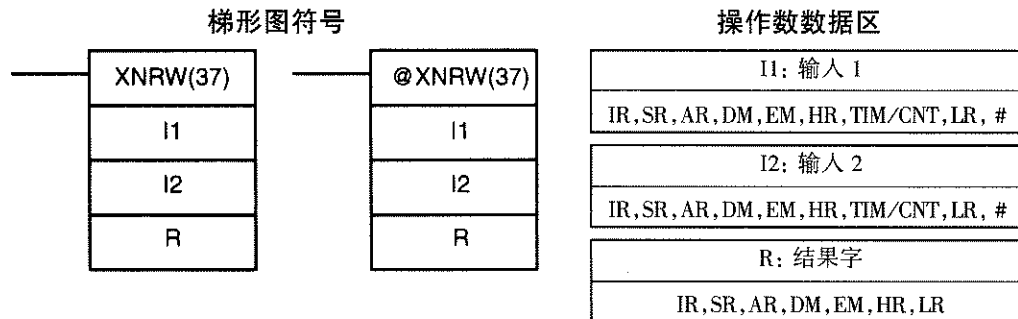
例



标志

ER: 间接寻址的 EM/DM 字不存在。
(*EM/*DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
EQ: 当结果为 0 时置 ON。

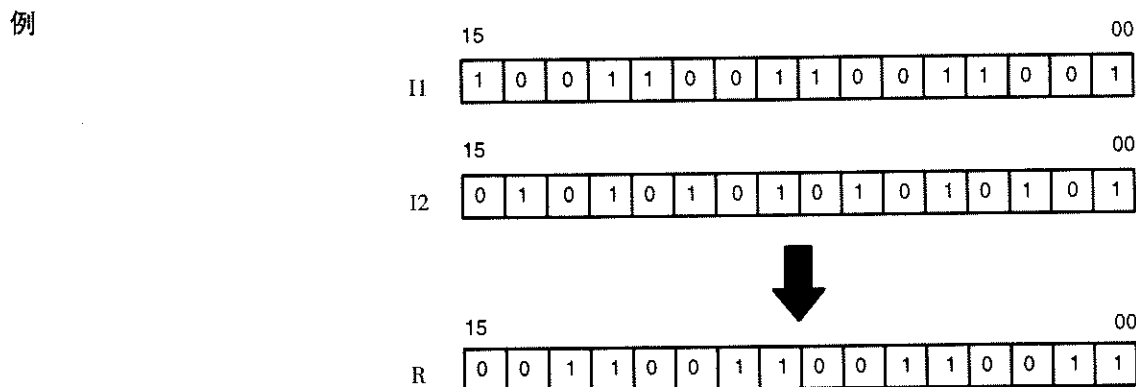
5-25-5 异或非—XNRW(37)



限制

DM6144 ~ DM6655 不能用于 R。

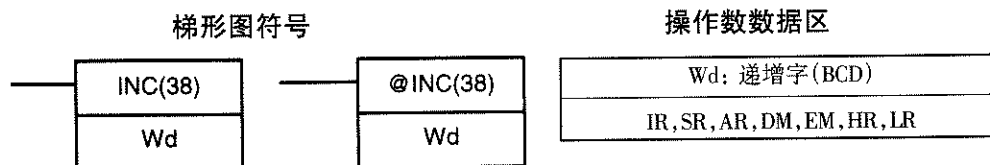
说明 当执行条件为 OFF 时, XNRW(37) 不执行。当执行条件为 ON 时, XNRW(37) 将 I1 和 I2 的内容按位求异或非, 并将结果置于 R。



标志 ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 当结果为 0 时置 ON。

5-26 递增/递减指令

5-26-1 BCD 递增—INC(38)



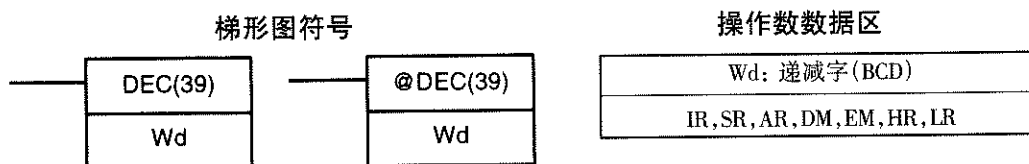
限制 DM6144 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, INC(38) 不执行, 当执行条件为 ON 时, INC(38) 递增 Wd, 不影响进位(CY)。

注意 若使用了非微分格式的 INC(38), Wd 的内容在每个循环将递增, 使用微分格式(@INC(38))或将 INC(38)与 DIFU(13)或 DIFD(14)组合, Wd 只递增一次。

标志 ER: Wd 不是 BCD 值。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或已超出 EM/DM 区边界。)
 EQ: 当递增结果为零时置 ON。

5-26-2 BCD 递减—DEC(39)



限制 DM6144 ~ DM6655 不能用于 Wd。

| | |
|----|--|
| 说明 | 当执行条件为 OFF 时, DEC(39) 不执行, 当执行条件为 ON 时。DEC(39) 递减 Wd, 不影响 CY。除了其递减数值而不是递增外 DEC(39) 与 INC(38) 以一样的方式工作。 |
| 注意 | 若使用了 DEC(39) 的非微分格式, Wd 的内容将在每循环递减。使用微分格式 (@DEC(39)) 或将 DEC(39) 与 DIFU(13) 或 DIFU(14) 组合, Wd 只递减一次。 |
| 标志 | ER: Wd 不是 BCD 值。 间接寻址的 EM/DM 字不存在。 (*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界。) EQ: 当递减 结果为 0 时置 ON。 |

5-27 子程序指令

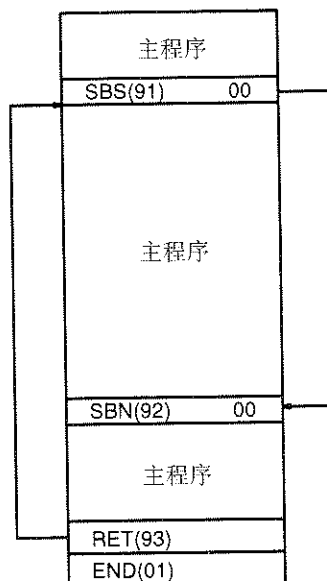
子程序将大的控制任务分成较小控制任务，使你能重复使用一给定的指令组，当主程序调用子程序时，控制被传送至子程序并且子程序指令被执行。子程序中的指令按主程序代码的同样方式编写。当子程序中所有指令执行完毕，控制返回主程序中刚才进入子程序的后一点（除非在子程序中有其它的规定）。

5-27-1 进入子程序—SBS(91)



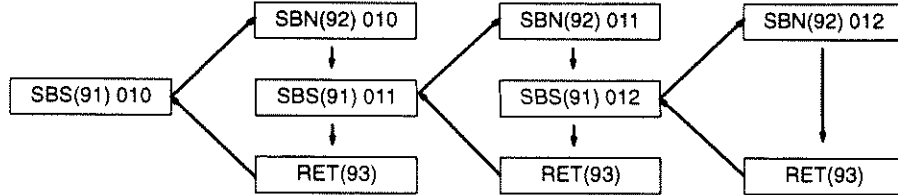
说明

若希望主程序中某个点执行一个子程序，则将 SBS(91) 置于该点，在 SBS(91) 中使用的子程序编号表示所希望的子程序。当 SBS(91) 被执行（即当其执行条件为 ON 时），在执行返回到作出调用的 SBS(91) 后面的指令前，执行具有相同子程序编号的 SBN(92) 与第一个 RET(93) 之间的指令。

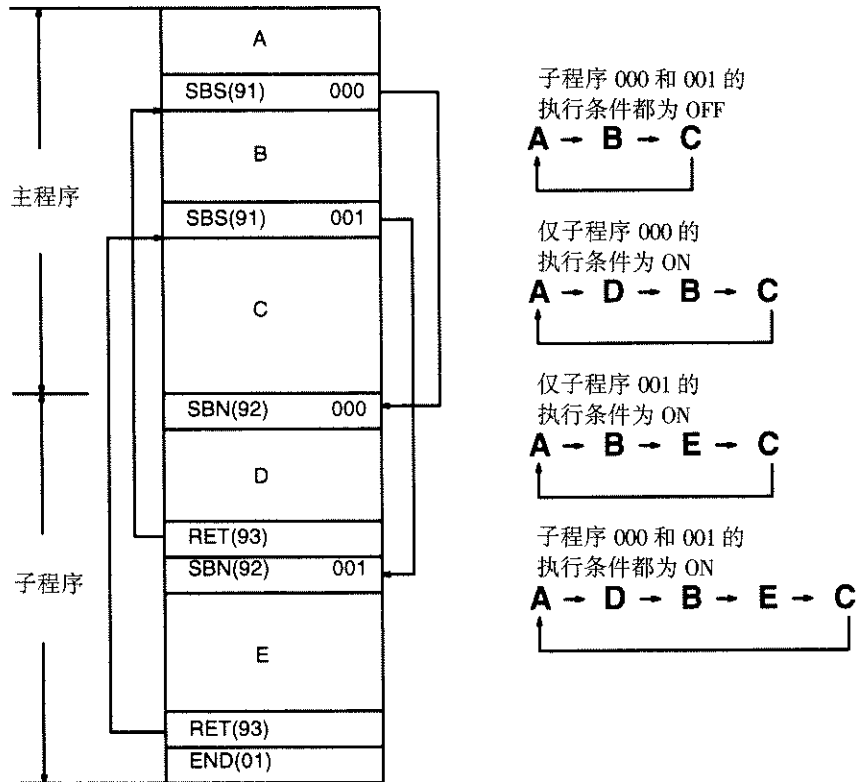


SBS(91) 可以在主程序中想用多少次就用多少次，即相同的子程序可以在程序中的不同的地方调用。

SBS(91)还可以置于子程序中,以将程序执行从一个子程序传递到另一个,即子程序可以嵌套。当第二个子程序已经完成(即到达了 RET(93)),程序执行返回到源子程序,执行完毕后才返回主程序。嵌套可以多达 16 层。子程序不能调用自己(例如,在用 SBN(92) 000 定义的子程序中不能编程 SBS(91) 000)。下图示出了二层嵌套。



下图说明了 2 个 SBS(91)在各种执行条件下的程序执行流程。

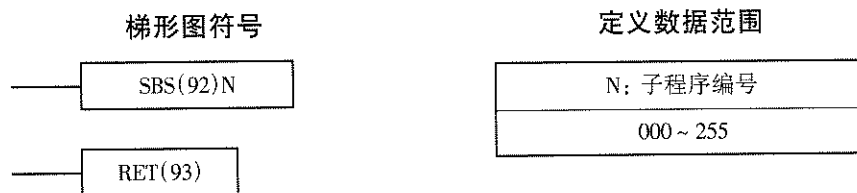


标志

- ER: 指定子程序编号的子程序不存在。
- 子程序调用自己。
- 激活的子程序被调用。

! 注意 当 ER 为 ON, SBS(91)将不执行,并不能调用子程序。

5-27-2 子程序定义和返回—SBN(92)/RET(93)



限制
说明

每一个子程序号只能被 SBN(92)使用一次。
SBN(92)用来标记子程序的起始;RET(93)则用来标记其结束,各子程序用子程序编号 N 来标记, N 编程为 SBN(92) 的定义符,此同一编号可由任何调用该子程序的 SBS(91)使用(见 5—27—1,进入子程序—SBS(91))。RET(93)不需要子程序编号。

所有子程序必须在主程序的结束处编程, 当有一个或更多的子程序被编程时, 主程序将执行到第一个 SBN(92), 然后返回到下一循环的起始地址 00000。子程序只有在用 SBS(91)调用时才执行。

END(01) 必须置于最后一个子程序后面, 即在最后一个 RET(93) 之后。在程序的任何其它点不要求 END(01)。

注意

若错误地将 SBN(92)放在主程序中, 这样会阻止程序超过该点执行, 也就是, 一遇到 SBN(92)程序执行将返回到起始点。

若将 DIFU(13)或 DIFD(14)置于一个子程序中, 在下次再执行子程序之前, 操作数位将不会返回 OFF, 即操作数位可能停留在 ON 状态超过一个循环。

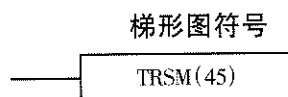
标志

没有直接受这些指令影响的标志位。

5-28 特殊指令

5-28-1 跟踪内存采样—TRSM(45)

数据跟踪的使用有利于调试程序。要设置和使用数据跟踪, 必须有运行 SSS 的上位计算机; 用编程器不可能进行数据跟踪, 数据跟踪在 SSS 操作手册: C 系列 PC 中有详细说明, 本节说明 TRSM(45)的梯形图符号及给出编程例子。



说明

在程序中 TRSM(45) 用来标记指定数据存储跟踪在跟踪内存中的位置, 可以指定最多 12 位和最多 3 个字作跟踪(详见 CX—Programmer 操作手册)。

TRSM(45) 不受执行条件控制, 而是受 AR 区的两个位 AR2515 和 AR2514 控制。AR2515 是采样起动位。该位置 ON 时启动采样过程以作跟踪。采样起动位不能由程序置 ON, 即必须由外部设备置 ON, AR2514 是跟踪起动位, 当它被置位, 指定数据被存储在跟踪内存。跟踪起动位可由程序或编程设备置位, 可设置正的或负的延迟, 以便更实际跟踪开始点。

数据可由三种方法中的任一种来记录, TRSM(45) 可置于程序中的一个或多个地点以指示指定数据何处被跟踪。若 TRSM(45) 未被使用, 则在 END(01) 被执行时跟踪指定数据。第三种方法用到由外部设备设置一个时间间隔, 以便指定数据能按独立于循环时间的规则的间隔被跟踪 (参见 SSS 操作手册: C 系列 PC)

TRSM(45) 可插入于程序的任何位置, 任意次。在跟踪内存中的数据可以由编程器, 上位计算机等等来监视。

AR 控制位和标志

下列控制位和标志在数据跟踪时使用, 跟踪运行时跟踪标志将置 ON。当足够的数据被跟踪并存满跟踪内存时, 跟踪完成标志置 ON。

| 标志 | 功能 |
|--------|--------|
| AR2515 | 采样起动位* |
| AR2514 | 跟踪起动位 |
| AR2513 | 跟踪标志 |
| AR2512 | 跟踪完成标志 |

注 *不要在程序中改变 AR2515 的状态。

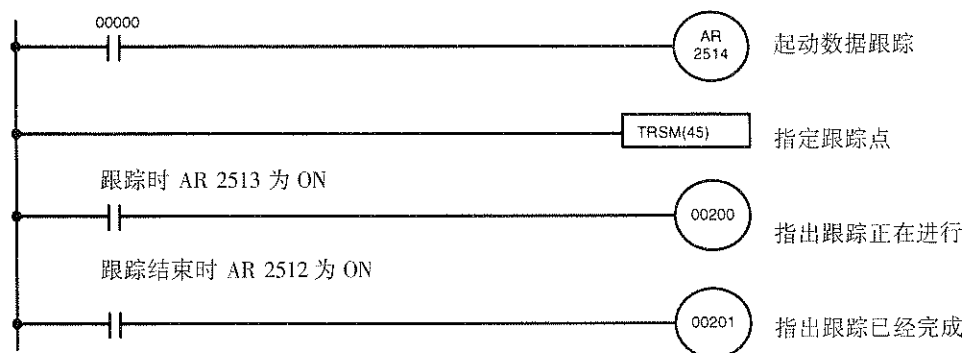
注意事项

若 TRSM(45) 存在, 当跳转条件 OFF 时, 在 JMP(08) — JME(09) 块中的 TRSM(45) 将不执行。

例

下例说明用于数据跟踪的基本程序及运行, 强制置位采样起动位 (AR2515) 以启动采样。采样起动位不能由程序置位, 数据被读出并存到跟踪内存。当 IR00000 为 ON 时, 跟踪起动位 (AR2514) 也被置 ON, 于是 CPU 检查延迟, 从而标记跟踪内存, 这意味着已经采样的一些数据将作为跟踪内存记录 (负延迟), 或在作记录前采样更多的数据 (正延迟)。

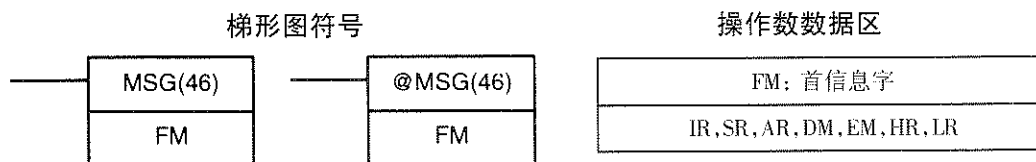
采样数据被写到跟踪内存,一旦到达内存区的结束处,就跳转到内存区的开始处继续写,直至开始标记,这意味着先前记录的数据(即本次采样置于开始标记前的数据)被覆盖(对正的延迟尤其是这样),对于负的延迟不能如此,因为要求的数据在采样开始之前。



| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 0000 |
| 00001 | OUT | AR 2514 |
| 00002 | TRSM(45) | |
| 00003 | LD | AR 2513 |

| 地址 | 指令 | 操作数 |
|-------|-----|---------|
| 00004 | OUT | 00200 |
| 00005 | LD | AR 2512 |
| 00006 | OUT | 00201 |

5-28-2 信息显示—MSG(46)



限制 DM6649 ~ DM6655 不能用于 FM。

说明 当以 ON 执行条件执行时, MSG(46) 从 FM ~ FM + 7 中读出 8 个扩展 ASCII 码,并将其显示在编程器。显示信息最多可达 16 个字符,即每个 ASCII 字符码需要 8 位(2 个数字)。关于 ASCII 码参见附录 H。日文片假名字符也包括在里面。
若信息不要求全部 8 个字,可输入“OD”,将其停在任一点。在信息中遇到 OD 时,就不再读字,通常用于信息的字就可用于别的目的。

信息缓冲及优先权 最多能有三组信息缓冲在内存,一旦存于缓冲器,它们基本按先进先出显示。因为在单一循环中可能有多于三个 MSG(46) 被执行,于是就有一个优先权安排,它是基于信息所存储的数据区来选择被缓冲的信息。
对于信息显示数据区优先权排列如下:

LR > IR > HR > AR > TIM/CNT > DM

处理同一数据区信息时,具有最低地址值的信息有更高的优先权。

处理间接寻址信息 (例如 *DM) 时, 具有最低最终 DM 地址的信息具有更高优先权。

清除信息

为了清除信息, 执行 FAL(06)00, 或通过编程器或 SSS 清除之。

若在信息显示时信息数据改变, 则显示也跟着改变。

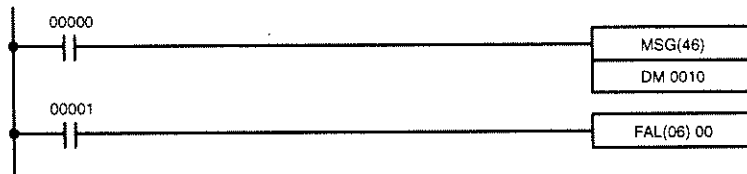
标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界)。

例

下例说明该指令将产生的显示及当 00000 为 ON 时给出的数据, 若 00001 为 ON, 信息将被清除。



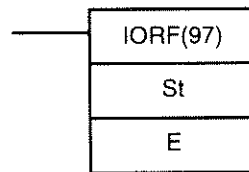
| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 00000 |
| 00001 | MSG(46) | |
| | | DM 0010 |
| 00002 | LD | 00001 |
| 00003 | FAL(06) | 00 |

| DM 内容 | | | | | 等效 ASCII 码 | |
|---------|---|---|---|---|------------|---|
| DM 0010 | 4 | 1 | 4 | 2 | A | B |
| DM 0011 | 4 | 3 | 4 | 4 | C | D |
| DM 0012 | 4 | 5 | 4 | 6 | E | F |
| DM 0013 | 4 | 7 | 4 | 8 | G | H |
| DM 0014 | 4 | 9 | 4 | A | I | J |
| DM 0015 | 4 | B | 4 | C | K | L |
| DM 0016 | 4 | D | 4 | E | M | N |
| DM 0017 | 4 | F | 5 | 0 | O | P |

MSG
ABCDEFGHIJKLMNPO

5-28-3 I/O 刷新—IORF(97)

梯形图符号



操作数数据区

| |
|---------------|
| St: 开始字 |
| IR000 ~ IR115 |
| E: 结束字 |
| IR000 ~ IR115 |

限制

St 必须小于或等于 E。

说明

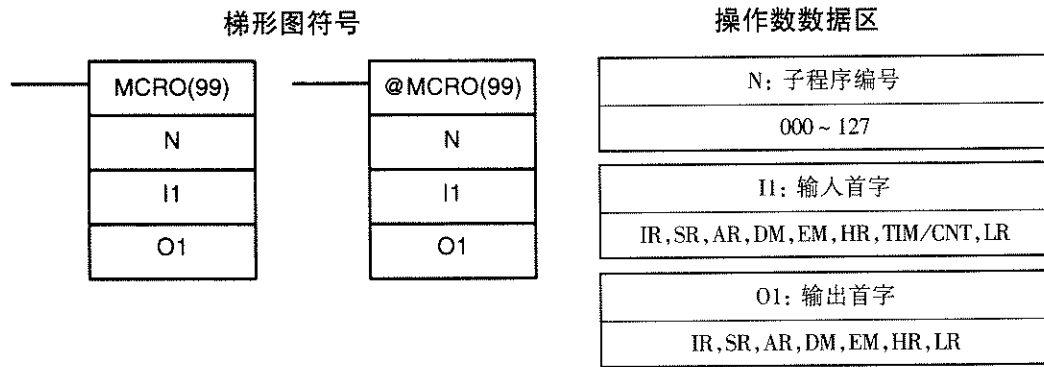
为了刷新 I/O 字, 仅仅指出要刷新的第一个 (St) 和最后一个 (E) I/O 字。当 IORE(97) 的执行条件为 ON, 在 St 和 E 之间的全部字被刷新, 这是在 CPU 的循环中在正常 I/O 刷新之外完成的刷新。

注 此指令不会影响没用于 I/O 的字。

标志

没有标志受此指令影响。

5-28-4 宏—MCRO(99)



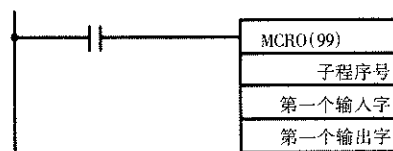
限制 DM6144 ~ DM6655 不能用于 O1。

说明 宏指令允许用一个单一子程序代替几个具有相同结构但不同操作数的子程序。有 4 个输入字 IR096 ~ IR099, 4 个输出字 IR196 ~ IR199, 分配给 MCRO(99)。并当子程序被执行时, 这 8 个字用于子程序从 I1 至 I1 + 3 和 O1 至 O1 + 3 取代其内容。

当执行条件为 OFF 时, MCRO(99) 不执行。当执行条件为 ON 时, MCRO(99) 将 I1 至 I1 + 3 的内容拷贝到 IR096 至 IR099, 将 O1 至 O1 + 3 的内容拷贝到 IR196 至 IR199, 然后调用并执行 N 指定的子程序。当子程序完成时, IR196 至 IR199 的内容传送回 O1 至 O1 + 3, 然后才结束 MCRO(99)。

宏功能允许通过只改变 I/O 字来使用一个子程序(编程形式), 一系列相似的程序能够只用一个子程序来处理, 从而大大缩减程序的步数, 并使得程序更易理解。

使用宏 要想使用宏, 必须用宏指令调用子程序, 如下图所示, 用 MCRO(99) 代替 SBS(91)(子程序进入)。



程序按以下步骤执行 MCRO(99):

1. 第一个输入字起始的四个字的内容传送到 IR096 至 IR099, 第一个输出字起始的四个字的内容传送到 IR196 至 IR199。
2. 将执行指定的子程序, 直到执行 RET(93)(子程序返回) 为止。
3. IR196 至 IR199 的内容传送到第一个输出字起始的四个字中。
4. MCRO(99) 执行完毕。

当执行 MCRO(99) 时, 可按需要使用相同的指令型式, 只要改变第一个输入字和第一个输出字。

- 每次执行宏指令时, 能使用的字仅第一个输入字起始的四个连续字(对输入)和第一个输出字起始的四个连续字(对输出)。

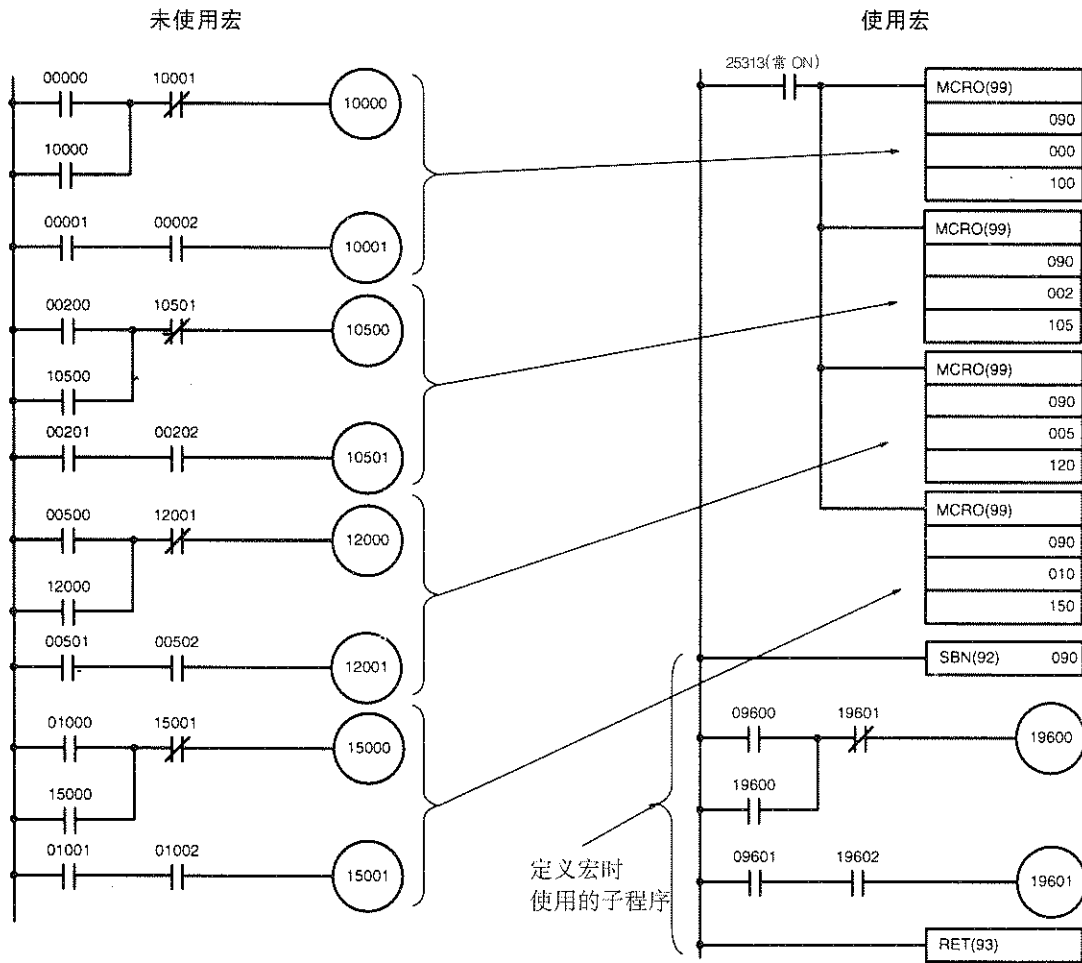
- 指定的输入和输出,必须正确地与子程序中的字相对应。
- 即使输出使用了直接输出方式,只有当子程序执行完毕(上面步骤 3)时,子程序的结果才实际映射到指定的输出字中。

注 IR096 至 IR099 和 IR196 至 IR199 在不使用 MCRO(99)时,可以用于工作位。第一个输入字和第一个输出字可以不必指定为 I/O 位,也可以是其它位(例如 HR 位,工作位等)或 DM 字。

MCRO(99) 调用的子程序同一般的子程序一样,由 SBN(92) 和 RET(93) 定义。

应用例

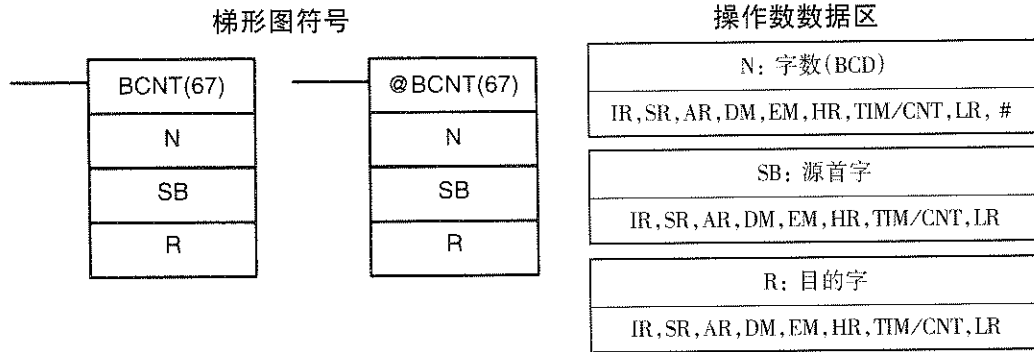
使用宏时,程序可简化成以下形式。



标志

- ER: 指定子程序编号的子程序不存在。
- 操作数超出数据区边界。
- 间接寻址的 EM/DM 字不存在。
- (* EM/* DM 字内容不是 BCD,或超出 EM/DM 区边界。)
- 子程序调用自己。
- 调用一已激活的子程序。

5-28-5 位计数器—BCNT(67)

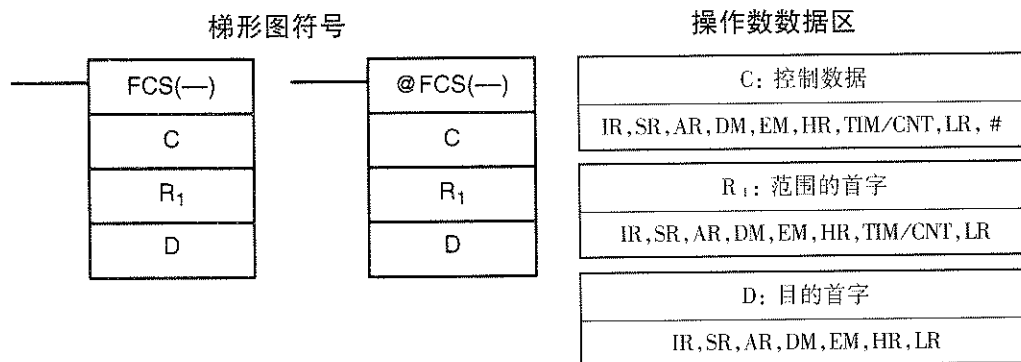


限制 N 不能是 0。
DM6144 ~ DM6655 不能用于 R。

说明 当执行条件为 OFF 时, BCNT(67) 不执行。当执行条件为 ON 时, BCNT(67) 计算在 SB 和 SB + (N - 1) 之间的所有字中的 ON 位的总数, 并将结果置于 R。

标志 ER: N 不是 BCD 值或 N 为 0; SB 和 SB + (N - 1) 不在同一数据区。
结果的计数值超过 9999。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 字内容不是 BCD 或超出 EM/DM 区边界。)
EQ: 当结果为 0 置 ON。

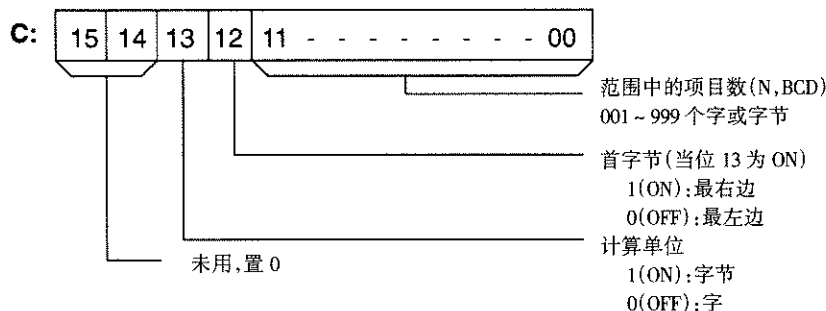
5-28-6 帧校验和—FCS(—)



限制 C 的最右边 3 个数字必须是 001 至 999 间的 BCD 值。
DM6143 ~ DM6655 不能用于 D。

说明 当经过通讯口传送数据时, FCS(—) 可用来检查错误。
当执行条件为 OFF 时, FCS(—) 不执行, 当执行条件为 ON 时, FCS(—) 通过对 R₁ 至 R₁ + N - 1 的字内容或 R₁ 至 R₁ + N - 1 的字节作异或, 对指定范围计算帧校验和, 帧校验和值 (16 进制) 被转换成 ASCII 并输出至目的字 (D 和 D + 1)

C 中各位的功能在下图中示出,在下面做更详细的解释。



范围内的项目数

在范围的项目数(N)包含在 C 的最右边 3 个数字中,它必须是 001 至 999 之间的 BCD 值。

计算单位

若位 13 为 OFF,则对字计算帧校验和,如位 13 为 ON,则对字节计算帧校验和。

若指定为字节,范围可由 R_1 的最左边或最右边的字节开始,若位 12 为 ON,则 R_1 的最左边字节不包括在内。

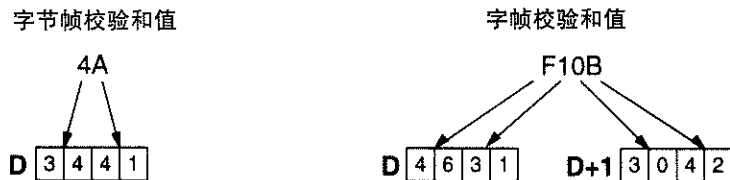
| | MSB | LSB |
|---------|-----|-----|
| R_1 | 1 | 2 |
| R_1+1 | 3 | 4 |
| R_1+2 | 5 | 6 |
| R_1+3 | 7 | 8 |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |

若位 12 为 OFF,字节异或按这样的顺序进行:1,2,3,4,...

若位 12 为 ON,字节异或按这样的顺序进行:2,3,4,5,...

转换到 ASCII

字节帧校验和计算给出 2 位数的 16 进制值,它转换为等值的 4 位数 ASCII 码。字的帧校验和计算给出 4 位数 16 进制值,它转换为 8 位数的等值的 ASCII 码,如下所示。



标志

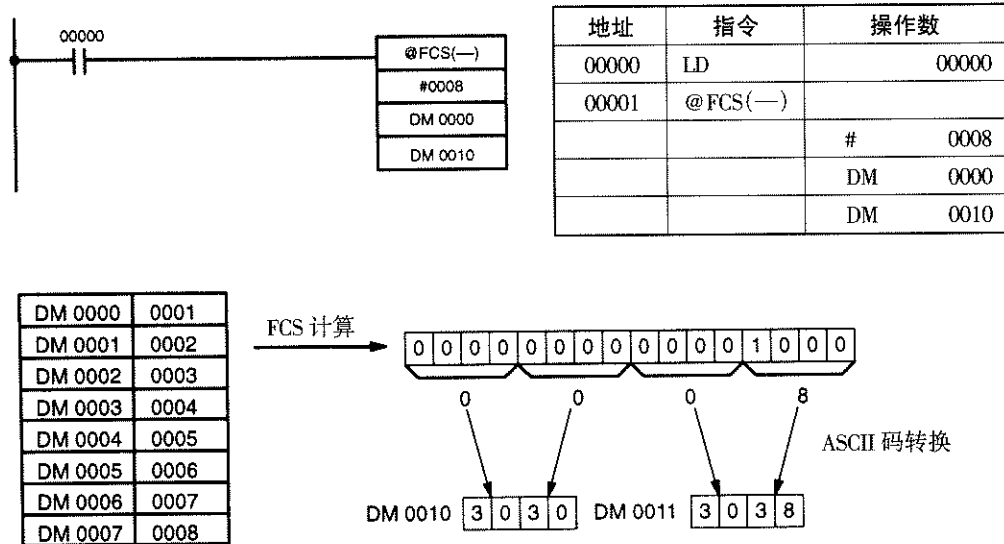
ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字内容不是 BCD,或超出 EM/DM 区边界)。

项目数不是 001 ~ 999 之间的 BCD 值。

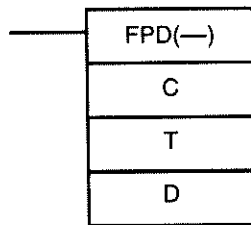
例

下例中,当 IR00000 为 ON 时,帧校验和(0008)计算从 DM0000 至 DM0007 中 8 个字,而等值 ASCII(30303038)被写到 DM0010 和 DM0011。



5-28-7 错误点检出—FPD(—)

梯形图符号



操作数数据区

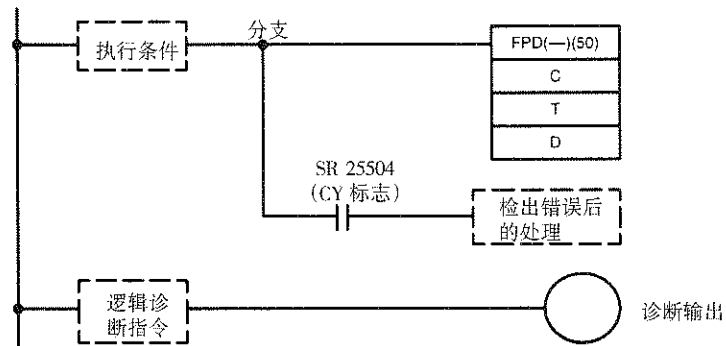
| |
|--|
| C: 控制数据 |
| # |
| T: 监视时间(BCD) |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| D: 登录首字 |
| IR, SR, AR, DM, EM, HR, LR |

限制

当 C 的 15 位为 ON 时, D 和 D + 8 必须在同一数据区。
DM6144 ~ DM6655 不能用于 T 或 D。
C 必须作为常数输入。

说明

FPD(—)可在程序中使用任意多次,但每次必须使用不同的 D。FPD(—)监视从其被执行开始至执行诊断输出之间的时间,若该时间超出 T,将产生一个在 C 中指定 FAL 编号的 FAL(06)非致命错误。
在下面的梯形图中,用短划线标示的程序段可根据专门的程序应用编写。由 CY 触发的处理编程段是可选的并可为除 LD 和 LD NOT 之外的任何指令使用。逻辑诊断指令和执行条件可根据需要由 NC 或 NO 条件的任一组合构成。

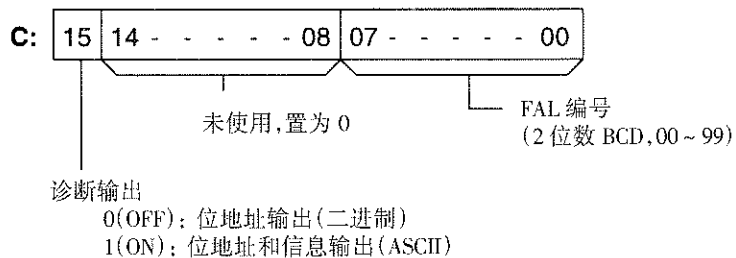


当执行条件为 OFF 时, FPD(-) 不执行。当执行条件为 ON 时, FPD(-) 监视直到逻辑诊断条件变为 ON, 并将诊断输出置 ON 的这段时间, 若这时间超过 T, 将出现下列情况:

- 1, 2, 3...
1. 发生一个 FAL(06) 错误, 其 FAL 编号由 C 的头二个数字指定, 若指定的编号为 00, 不会产生错误。
 2. 搜索第一个 OFF 输入条件逻辑诊断指令, 而此条件的位地址输出到以 D 开始的目标字。
 3. CY 标志 (SR25504) 置 ON。若需要可用 CY 标志执行错误处理程序段。
 4. 若 C 的 15 位为 ON, 最多 8 个 ASCII 字符的当前信息和第 2 步提到的位地址将显示在外部设备上。

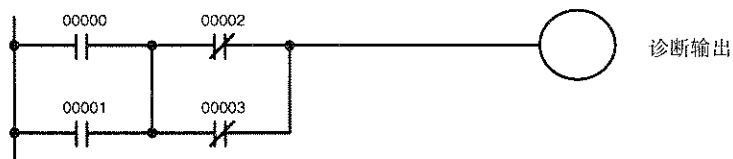
控制数据

控制数据 C 各位的功能在下图示出。



逻辑诊断指令

若直至逻辑诊断条件变为 ON 的这段时间超过 T, 逻辑诊断指令搜索 OFF 输入条件, 若多于一个的输入条件为 OFF, 则选择位于最高指令行及最靠近左边总线列的输入条件。

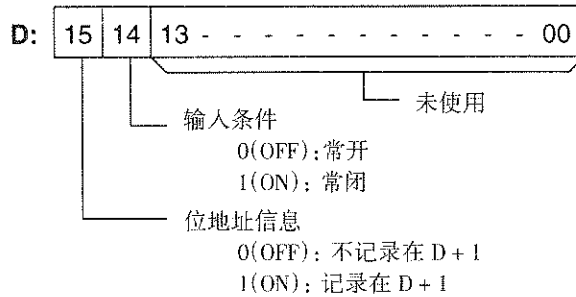


当 IR00000 至 IR00003 为 ON, 则常闭条件 IF00002 被找到, 作为引起诊断输出的条件没有置 ON 的原因。

诊断输出

有两种方式来输出在逻辑诊断条件中检出的 OFF 条件的位地址。

- 1, 2, 3... 1. 位地址输出(当 C 的 15 位为 OFF 时使用)。
 D 的 15 位指示位地址信息是否存储在 D + 1。若是的,第 14 位指示输入条件是常开还是常闭。



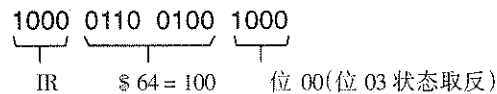
如下所示, D + 1 包含了输入条件的位地址码。字地址, 位编号及 TIM/CNT 编号都是二进制值。

| 数据区 | D + 1 位状态 | | | | | | | | | | | | | | | | | |
|----------|-----------|----|----|----|-----|-----|-----|-----------|----|----|----|-----|-----|-----|----|----|--|--|
| | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 | | |
| IR, SR | 1 | 0 | 0 | 0 | 字地址 | | | | | | | 位编号 | | | | | | |
| HR | 1 | 0 | 0 | 1 | 1 | 字地址 | | | | | | | 位编号 | | | | | |
| LR | 1 | 0 | 0 | 1 | 0 | 0 | 字地址 | | | | | | | 位编号 | | | | |
| TIM/CNT* | 1 | 0 | 0 | 1 | 0 | 1 | * | 定时器或计数器编号 | | | | | | | | | | |

注 a) *对于 TIM/CNT 区, D + 1 的位 09 表示编号是定时器的还是计数器的。0 表示定时器, 1 表示计数器。

b) 位编号最左位(位 03)是取反的。

例 若 D + 1 包含 1000 0110 0100 1000, IR10000 指示如下:



2. 位地址和信息输出(当 C 的 15 位为 ON 时选择)。
 D 的 15 位表示位地址信息是否存储在 D + 1 ~ D + 3。若是, D 的 14 位表示输入条件是常开还是常闭, 参见下表。

当 FPD(一) 执行时, 字 D + 5 至 D + 8 包含了显示于外部设备的 ASCII 信息及位地址。字 D + 5 至 D + 8 包含由用户预置的信息, 如下表所示。

| 字 | 位 15 ~ 08 | 位 07 ~ 00 |
|-------|--------------|--------------------|
| D + 1 | 20 = 空格 | 第一个 ASCII 字符 |
| D + 2 | 第二个 ASCII 字符 | 第三个 ASCII 字符 |
| D + 3 | 第四个 ASCII 字符 | 第五个 ASCII 字符 |
| D + 4 | 2D = " - " | "0" = 常开, "1" = 常闭 |
| D + 5 | 第一个 ASCII 字符 | 第二个 ASCII 字符 |
| D + 6 | 第三个 ASCII 字符 | 第四个 ASCII 字符 |
| D + 7 | 第五个 ASCII 字符 | 第六个 ASCII 字符 |
| D + 8 | 第七个 ASCII 字符 | 第八个 ASCII 字符 |

注 若信息不需要 8 个字符,在最后一个字符后输入“OD”。

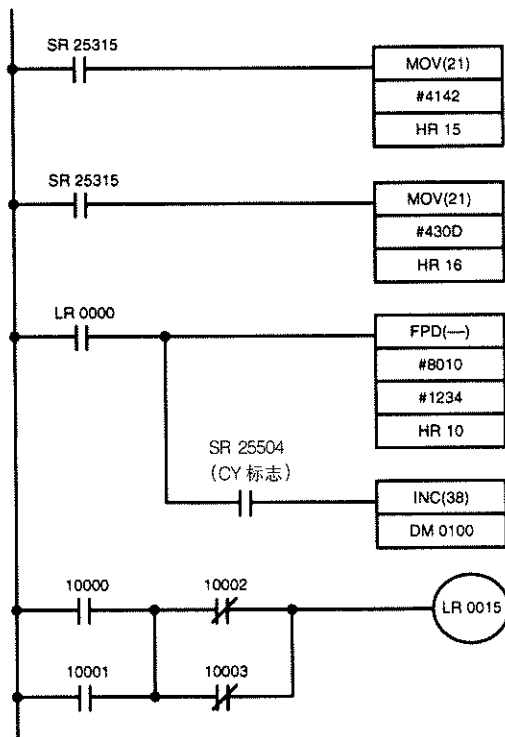
决定监视时间

下述步骤当指定一个字操作数作为 T 时可在真实操作条件下,自动设置监视时间 T。若设置常数作为 T 则不能使用此操作。

- 1,2,3... 1. 将 CQM1H 切换为监视模式。
2. 连接外部设备,如编程器。
3. 用外部设备将控制位 AR2508 置 ON。
4. AR2508 置 ON 时执行程序,若 T 的当前监视时间被超过,真实监视时间的 1.5 倍将被存到 T 中。在 AR2508 置 ON 期间 FAL(06)错误不会发生。
5. 若一个可接受的值已存在 T 中,AR2508 置 OFF。

例

在下例中,FPD(—) 被设置为当监视时间 123.4 秒被超过时,就显示位地址和信息(“ABC”)。



| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00000 | LD | 25315 |
| 00001 | MOV(21) | |
| | | # 4142 |
| | | HR 15 |
| 00002 | LD | 25315 |
| 00003 | MOV(21) | |
| | | # 430D |
| | | HR 16 |
| 00004 | LD | LR 0000 |
| 00005 | FPD(—) | |
| | | # 0010 |
| | | # 1234 |
| | | HR 10 |
| 00006 | AND | 25504 |
| 00007 | INC(38) | |
| | | DM 0100 |
| 00008 | LD | 10000 |
| 00009 | OR | 10001 |
| 00010 | LD NOT | 10002 |
| 00011 | OR NOT | 10003 |
| 00012 | AND LD | |
| 00013 | OUT | LR 0015 |

当 LR0000 为 ON,FPD(—)被执行并开始监视。若 LR0015 在 123.4 秒内不为 ON,而 IR10000 至 IR10003 全为 ON,IR10002 将被选出为出错的原因。并产生 FAL 编号为 10 的 FAL(06)错误,位地址和预置信息(“10002—1ABC”)将显示在外部设备上。

| | |
|-------|------|
| HR 10 | 0000 |
| HR 11 | 0000 |
| HR 12 | 0000 |
| HR 13 | 0000 |
| HR 14 | 0000 |
| HR 15 | 4142 |
| HR 16 | 430D |
| HR 17 | 0000 |
| HR 18 | 0000 |



| | |
|-------|------|
| HR 10 | C000 |
| HR 11 | 2031 |
| HR 12 | 3030 |
| HR 13 | 3032 |
| HR 14 | 2D31 |
| HR 15 | 4142 |
| HR 16 | 430D |
| HR 17 | 0000 |
| HR 18 | 0000 |

表示信息,常闭条件
“1”
“00”
“02”
“- 1”
“AB”
“C”,和 CR 码
最后两个字无作用
(显示为空格)

标志

ER: T 不是 BCD 值。

C 不是常数或不是 00 至 99 间的 BCD 值。

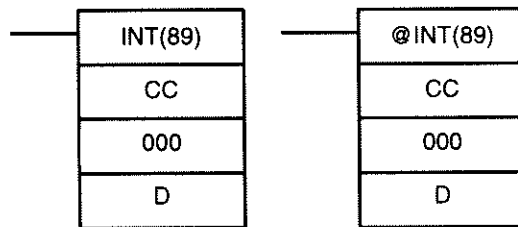
间接寻址的 EM/DM 字不存在。

(*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界)。

CY: 在执行 FPD(一)和执行诊断输出之间的时间超过 T 时置 ON。

5-28-8 中断控制 - INT(89)

梯形图符号



操作数数据区

| |
|--|
| C: 控制数据 |
| # (000 到 003, 100, 或 200) |
| 000: 无功能 |
| # (000) |
| D: 控制数据 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, TR, # |

限制
说明

当 CC = 002 时, DM6644 ~ DM6655 不能用于 D。

当执行条件为 OFF 时, INT(89) 不执行。当执行条件为 ON 时, INT(89) 用来控制中断, 并根据 CC 的值完成下表所示 6 种功能中的一种。

| INT(89) 功能 | CC |
|------------|-----|
| 屏蔽/不屏蔽输入中断 | 000 |
| 清除输入中断 | 001 |
| 读当前屏蔽状态 | 002 |
| 更新计数器设定值 | 003 |
| 屏蔽所有中断 | 100 |
| 去除屏蔽所有中断 | 200 |

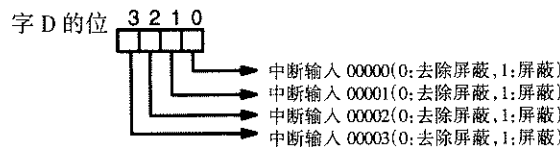
在下面对 6 种功能作更详细说明, 关于这些功能的更多信息参见 38 页。

屏蔽/去除屏

蔽 I/O 中断 (CC = 000)

此功能用来对 I/O 中断输入 00000 至 00003 进行屏蔽或去除屏蔽。屏蔽的输入被记录, 但被忽略。当一个输入被屏蔽时, 一旦屏蔽被解除, 对应的中断程序马上被执行 (除非预先执行 CC = 001 的 INT(89) 来清除之)。

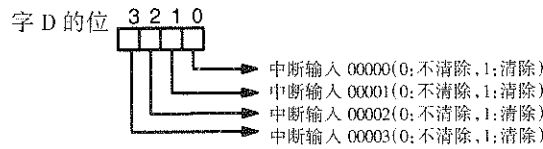
设置 D 中的对应位为 0 或 1 来去除所蔽或屏蔽 I/O 中断输入。位 00 至 03 对应 00000 至 00003。位 04 至 15 应设为 0。



清 I/O 中断 (CC = 001)

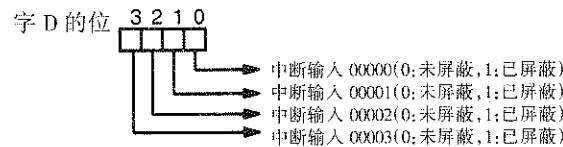
此功能用来清除 I/O 中断输入 00000 至 00003。因为中断输入被记录时, 一旦把被屏蔽的中断去除屏蔽, 对应的中断将运行, 除非在这之前把它们清除。

设置 D 中的对应位为 1 以清除 I/O 中断,位 00 至 03 对应 00000 至 00003。位 04 至 15 应设为 0。



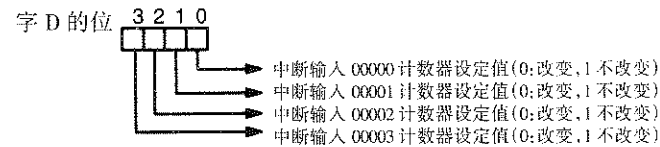
读当前屏蔽状态(CC = 002)

此功能用来将 I/O 中断输入 00000 ~ 00003 的当前屏蔽状态写至字 D,若输入被屏蔽则对应位为 0N(位 00 至 03 对应 00000 至 00003)。



更新计数器设定值(CC = 003)

此功能用来将 I/O 中断输入 00000 至 00003 的计数器设定值更新,将 D 中的对应位置 1 以将输入的计数器设定值更新。(位 00 至 03 对应 00000 至 00003)。



屏蔽/去除屏蔽所有中断(CC = 100/200)

此功能用来屏蔽或去除屏蔽所有中断处理。被屏蔽的输入被记录,但不处理,详见 26 页。

此功能不使用控制数据 D。将 D 设为 # 0000。

标志

ER: 计数器设定值不正确(仅当 CC = 003 时)。

间接寻址 EM/DM 字不存在。

(*EM/*DM 字内容不是 BCD,或超出 EM/DM 区边界)。

当中断程序正在执行时 CC = 100 或 200。

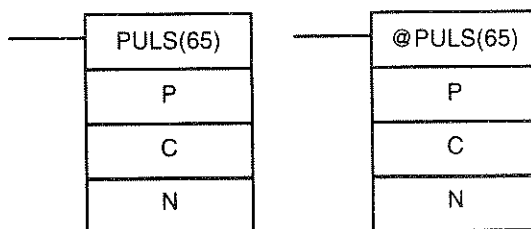
当输入已被屏蔽时 CC = 100。

当所有输入已去除屏蔽时 CC = 200。

CC 和/或 D 不在指定值范围。

5-28-9 设置脉冲 - PULS(65)

梯形图符号



操作数数据区

| |
|----------------------|
| P: 端口指定 |
| 000,001 |
| C: 控制数据 |
| 000 到 005 |
| N: 脉冲数 |
| IR,SR,AR,DM,EM,HR,LR |

限制 N 和 N + 1 必须在同一数据区。

DM6143 ~ DM6655 不能用于 N。

说明 PULS(65)可用于下表列出的功能。

| 单元/板 | 功能 |
|---------|---------|
| 晶体管输出单元 | 脉冲输出 |
| 脉冲I/O板 | 脉冲输出1和2 |

PULS(65) 用来设置脉冲输出的参数，脉冲输出将在以后在程序中用 SPED (64)或 ACC(—)启动。可设置的参数为,以独立模式输出的脉冲数,端口 1 和 2 的脉冲输出的方向,和由 ACC(—)模式 0 控制的脉冲输出的减速点。

因为 PULS(65) 具有相当长的执行时间,所以只在需要时才通过此指令的微分形式 (@PULS(65))的执行以减少循环时间。

注 详见 1—5 脉冲输出功能。

端口指定 (P) 端口指定表示脉冲输出的位置,在 C 和 N 中设置的参数将应用于下一个 SPED(64)或 ACC(—)指令,在这些指令中指定相同的端口输出位置。

| 脉冲输出位置 | P |
|----------------|-----|
| 输出位00 ~ 15(见注) | 000 |
| 端口1 | 001 |
| 端口2 | 002 |

注 以接点脉冲输出的 00 和 15 间的位由 SPED(64)中的 P 操作数指定。

控制数据 (C) 该控制数据确定端口 1 和 2 的脉冲输出的方向,和指出是否在 N 至 N + 3 中规定脉冲数和/或减速点。在 P 中指定输出位时 (P = □□0),这操作数应置为 000。

| C | 方向 | 脉冲数 | 减速点 |
|-----|-----|-------------|------------------|
| 000 | CW | 在N和N + 1中设定 | 不设定 |
| 001 | CCW | 在N和N + 1中设定 | 不设定 |
| 002 | CW | 在N和N + 1中设定 | 在N + 2和N + 3中设定 |
| 003 | CCW | 在N和N + 1中设定 | 在N + 2和 N + 3中设定 |
| 004 | CW | 不设定 | 不设定 |
| 005 | CCW | 不设定 | 不设定 |

方向的设置在程序执行停止或 PULS(—)重新执行之前都是有效的。

脉冲数 (C = 000 或 C = 001)

当 C = 000 或 001, N + 1, N 包含独立模式脉冲输出的 8 位数脉冲设定数。N + 1, N 从 00000001 至 16777215。SPED(64)或 ACC(—)起动的脉冲输出在输出指定数目的脉冲后自动停止。

左边 4 位数 右边 4 位数 允许范围
 脉冲数: N + 1 N 0000 0001 至 1677 7215

脉冲数和减速点 (C = 002 或 C = 003)

当 C = 002 或 003, N + 1, N 包含独立模式脉冲输出的 8 位数脉冲设定数。N + 1, N 可从 00000001 至 16777215。由 ACC(—)起动的脉冲输出在输出指定数目的脉冲后自动停止。

脉冲数:

| | |
|---------|-------|
| 左边 4 位数 | N + 1 |
|---------|-------|

| | |
|---------|---|
| 右边 4 位数 | N |
|---------|---|

 允许范围 0000 0001 至 1677 7215

N + 3, N + 2 包含 ACC(一) 模式 O 中使用的 8 位数减速点脉冲设定, N + 3, N + 2 从 00000001 至 16777215。ACC(一) 起动的脉冲输出在这些脉冲输出后开始减速。

脉冲数:

| | |
|---------|-------|
| 左边 4 位数 | N + 1 |
|---------|-------|

| | |
|---------|---|
| 右边 4 位数 | N |
|---------|---|

 允许范围 0000 0001 至 1677 7215

改变输出目的(C = 004 或 C = 005)

当 C = 004 或 005, 脉冲数和减速点都不设置, N 设置为 000。使用这些设置来改变 1 口或 2 口连续模式脉冲输出的输出目的。

频率改变

即使在运行过程中用 SPED(64) 改变了脉冲频率, 仍使用设置的输出脉冲数。(运行中不能改变脉冲数)

例如, 若脉冲数设为 2100 而频率从 1kHz 改变为 100Hz, 脉冲输出将停止在: 12 秒, 若脉冲频率在 1kHz 于 1 秒后改变。

3 秒, 若脉冲频率在 1kHz 于 2 秒后改变。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM / * DM 字内容不是 BCD, 或超出 EM/DM 区边界)。

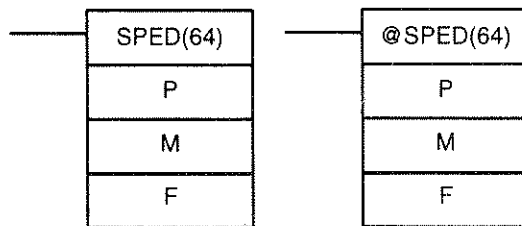
数据区超出边界。

指令设置有错误。

在主程序中脉冲 I/O 或高速计数器指令正在执行时, 在中断子程序执行 PULS(65)。

5-28-10 速度输出—SPED(64)

梯形图符号



操作数数据区

| |
|-------------------------------|
| P: 端口指定 |
| 000, 001 或 010 ~ 150 |
| M: 输出方式 |
| 000 或 001 |
| F: 脉冲频率 |
| IR, SR, AR, DM, EM, HR, LR, # |

限制

当指定端口时, F 必须是 BCD 值 # 0000 至 # 5000, 若规定输出位时, 为 # 0000 或 # 0002 ~ # 0100。

DM6144 ~ DM6655 不能用于 F。

说明

SPED(64) 可以用于下表所列各功能。

| 单元/板 | 功能 |
|----------|------------|
| 晶体管输出单元 | 脉冲输出 |
| 脉冲 I/O 板 | 脉冲输出 1 和 2 |

SPED(64)用来为指定的端口或输出位设置、改变或停止脉冲输出。当执行条件为 OFF 时,SPED(64)不执行。当执行条件为 ON 时,SPED(64)为 P 所规定的端口或输出位设置脉冲频率 F。M 决定输出模式。

因为 SPED(64) 有相当长的执行时间,所以仅当需要时执行该指令的微分格式(@SPED(64))以减小循环时间。

注 详见 1 - 5 脉冲输出功能。

端口指定(P)

端口指定规定了脉冲输出的端口或输出位。

| P | 脉冲输出位置 |
|-----------|--|
| 001 | 端口 1 |
| 002 | 端口 2 |
| 000 ~ 150 | 输出位 IR10000 ~ IR10015。 P 的头 2 个数字指定 IR100 的那个位为输出位, P 的第三个数字总是设为 0, 例如 P = 000 指定 IR10000, P = 010 指定 IR10001, ... 和 P = 150 指定位 IR10015。 |

输出模式(M)

M 值决定了输出模式。

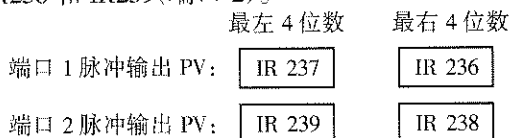
| M | 输出模式 |
|-----|---------------------|
| 000 | 独立模式, 频率以单位 10Hz 设置 |
| 001 | 连续模式, 频率以单位 10Hz 设置 |
| 002 | 独立模式, 频率以单位 1Hz 设置 |
| 003 | 连续模式, 频率以单位 1Hz 设置。 |

注 设定 002 和 003 只用于指定脉冲 I/O 板的 1 口和 2 口(P = 001 或 P = 002)。独立模式下, 脉冲输出是连续的, 直到下述事件之一出现:

- 1, 2, 3, ...
1. 到达由 PULS(65) 指令指定的脉冲数 (当指定独立模式, 在执行 SPED(64) 前执行 PULS(65))。
 2. C = 003 的 INI(61) 指令被执行。
 3. 输出频率 F 设为 000 的 SPED(64) 再被执行。

当输出脉冲是独立模式, 事先用执行 PULS(65) 指定脉冲数。若从端口 1 或 2 输出, 还要指定方向(CW 或 CCW)。

在独立模式中, 已经输出到端口 1 和 2 的脉冲数包含在 IR236 和 237(端口 1) 以及 IR238 和 IR239(端口 2)。



在连续模式, 脉冲一直输出直到执行了 C = 003 的 INI(61) 指令或重新执行 F = 0000 的 SPED(64) 指令。若是从端口 1 或 2 输出时未指定方向 (CW 或 CCW), 则脉冲为 CW。

脉冲频率(F)

F 值设置脉冲频率, 如下所示, 设置 F 为 0000 将停止在指定位置的脉冲输出。

| 输出 | 单位 | F 的可能值 |
|----------|-------|--|
| 输出位 | 10 Hz | 0000(停止输出。)或 0002 至 0100(20 Hz 至 1 kHz) |
| 端口 1 或 2 | 10 Hz | 0000(停止输出。)或 0001 至 5000(10 Hz 至 50 kHz) |
| | 1 Hz | 0000(停止输出。)或 0010 至 9999(10 Hz 至 9,999 Hz) |

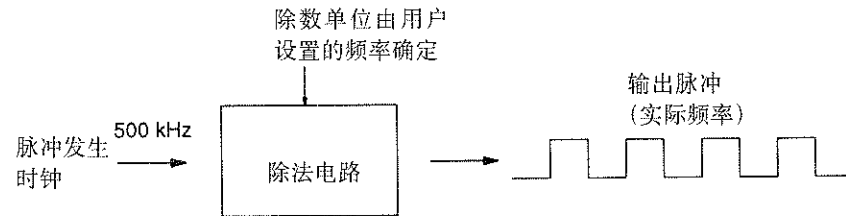
关于脉冲输出的注意

从 CQM1H-PLB21 脉冲 I/O 板上输出的脉冲频率是通过用整数值去除 500 kHz 基本时钟脉冲而产生的, 因此将导致设置频率和实际频率存在差距。参照以下公式来计算实际频率。

设置频率: 用户在指令中设置的输出频率

除数单位: 在除法电路中设置的一个整数, 以产生一个设置频率的输出脉冲

实际频率: 从除法电路中实际输出的输出脉冲频率



公式:

$$\text{实际频率 (KHz)} = 500(\text{KHz}) / \text{INT}(500(\text{KHz}) / \text{设置频率}(\text{KHz}))$$

INT: 取整函数

INT(500/设置频率): 除数单位

当频率变高时, 设置频率与实际频率的差距变大。

例

| 设置频率 (kHz) | 实际频率 (kHz) |
|---------------|------------|
| 45.46 ~ 50.00 | 50.00 |
| 41.67 ~ 45.45 | 45.45 |
| 38.47 ~ 41.66 | 41.67 |
| : | : |
| 31.26 ~ 33.33 | 33.33 |
| 29.42 ~ 31.25 | 31.25 |
| 27.78 ~ 29.41 | 29.41 |
| : | : |
| 20.01 ~ 20.83 | 20.83 |
| 19.24 ~ 20.00 | 20.00 |
| 18.52 ~ 19.23 | 19.23 |
| : | : |
| 10.01 ~ 10.20 | 10.20 |
| 9.81 ~ 10.00 | 10.00 |
| 9.62 ~ 9.80 | 9.80 |
| : | : |
| 5.01 ~ 5.05 | 5.05 |
| 4.96 ~ 5.00 | 5.00 |
| 4.90 ~ 4.95 | 4.95 |
| : | : |
| 3.02 ~ 3.03 | 3.03 |
| 3.00 ~ 3.01 | 3.01 |
| 2.98 ~ 2.99 | 2.99 |
| : | : |

注意

当间隔定时器 O 正在运行时, 不能使用脉冲输出。

当脉冲输出位输出了频率为 500Hz 或以上的脉冲时,对 TIMH(15) 设置中断处理,TIM/CNT 号为 000 至 003,(PC 设置的 DM6629 设置为 # 0104。
一次只能有一个输出位有脉冲输出。

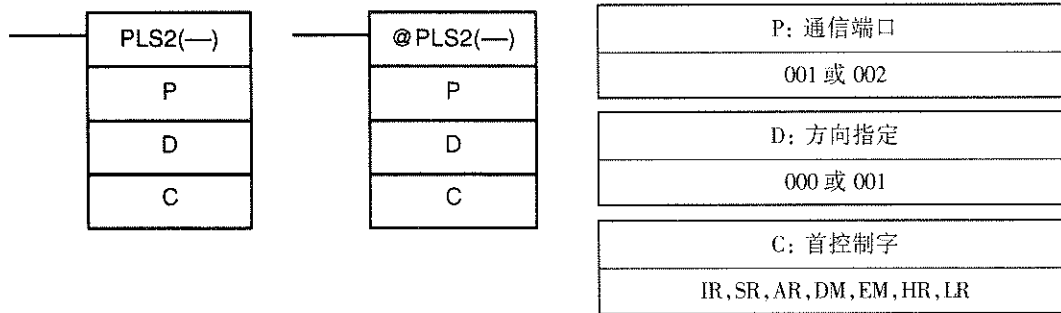
标志

ER: 间隔定时器 O 运行时执行 SPED(64)。
间接寻址的 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界。)
指令设置中有错误。
SPED(64) 在中断程序中执行,而在程序中正在执行脉冲 I/O 或高速计数器指令。

5-28-11 脉冲输出 - PLS2(—)

梯形图符号

操作数数据区



限制

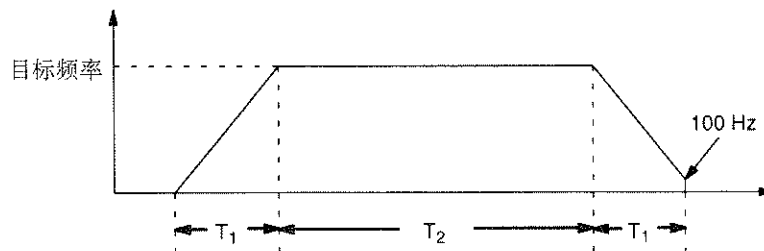
若 PC 设置(DM6611)设为高速计数器模式,则不能使用 PLS2(—)。
P 必须是 001 或 002,D 必须是 000 或 001。
C ~ C + 3 必须在同一数据区。

说明

PLS2(—)能使用下表所列的功能。

| 单元/板 | 功能 |
|--------|---|
| 脉冲I/O板 | 脉冲输出1和2 (端口1和端口2必须在DM6611中将其模式设置为简单定位模式。如果设置成高速计数器模式,则PLS2(—)不可用。) |

PLS2(—) 用来从端口 1 或 2 输出指定数量的 CW 或 CCW 脉冲。脉冲输出按指定的速率加速到目标频率,并以相同的速率减速。(脉冲输出在 100Hz 时停止)。



下式说明如何计算加速/减速时间 T1 和运行时间 T2 的近似值, 时间单位为秒。

$$T_1 \cong 0.004 \times \frac{\text{目标频率}}{\text{加速/减速速率}}$$

$$T_2 \cong \frac{\text{脉冲数} - (T_1 \times \text{目标频率})}{\text{目标频率}}$$

- 注 1. 虽然由于运行条件影响, T1 和 T2 会有非常小的变化, 而脉冲数输出是精确的。
 2. 若指定的端口正在输出脉冲, PLS2(一)将不运行。执行 PLS2(一)之前检查脉冲输出标志 (AR0515 对应端口 1, AR0615 对应端口 2)。
 3. 详见 1—5 脉冲输出功能。

操作数设置

P 指定脉冲输出的端口。P = 001 脉冲从端口 1 输出, P = 002 脉冲从端口 2 输出。

D 指定输出信号是顺时针方向 (CW) 还是反时针方向 (CCW)。D = 000 输出 CW, D = 001 为 CCW。

C 的内容决定加速/减速速率, 在加速或减速期间, 每 4.08ms 输出频率以设置在 C 中的数值加速或减速。C 必须是 0001 至 0200 之间的 BCD 值 (10Hz 至 2KHz)。

C + 1 内容指定目标频率。C + 1 必须是 0010 至 5000 间的 BCD 值 (100Hz 至 50KHz)。

C + 3, C + 2 中的 8 位数内容决定了将要输出的脉冲数, C + 3, C + 2 必须是 00000001 至 16777215 之间的 BCD 值。

标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字内容不是 BCD, 或超出 EM/DM 区边界)。

操作数设置有错。

没有安装脉冲 I/O 板时执行 PLS2(一)。

PC 设置没设脉冲输出。

目标频率, 加速/减速速率, 脉冲数不正确。

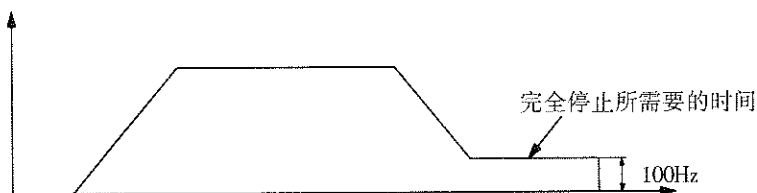
(脉冲数 < T₁ × 目标频率)。

在主程序正在执行脉冲 I/O 或高速计数器指令时中断子程序执行 PLS2(一)。

AR0515: 端口 1 输出标志, 当端口 1 正在输出脉冲时为 ON。

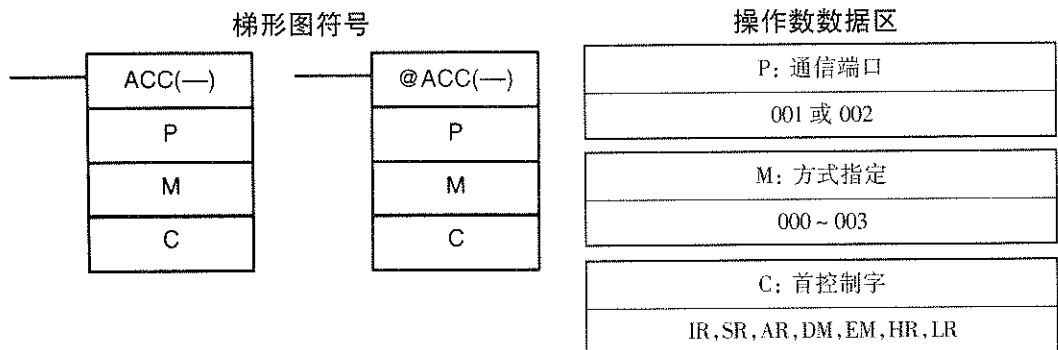
AR0615: 端口 2 输出标志, 当端口 2 正在输出脉冲时为 ON。

注意 使用 PLS2(一) 时, 加/减速和目标速度的条件可导致低速脉冲输出 (100Hz) 持续一定时间, 即使如此, 将输出正确的脉冲数。



停止时,通过校正加/减速和/或目标速度,或通过使用 ACC(—) 指令(模式 0)以增加速度(减速目标频率)来修正系统。

5-28-12 加速控制—ACC(—)



限制

若 PC 设置(DM6611)设为高速计数器模式,就不能使用 ACC(—)模式 0。
P 必须是 001 或 002, M 必须是 000 至 003。

说明

C 至 C + 3 必须在同一数据区。
ACC(—)可使用下表所列的功能。

| 单元/板 | 功能 |
|--------|---|
| 脉冲I/O板 | 脉冲输出1和2 (为了使用ACC(—)模式0,端口1和2必须设置简单定位模式(PC设置的DM6611)。如果模式设置为高速计数器模式,则不能使用ACC(—)。) |

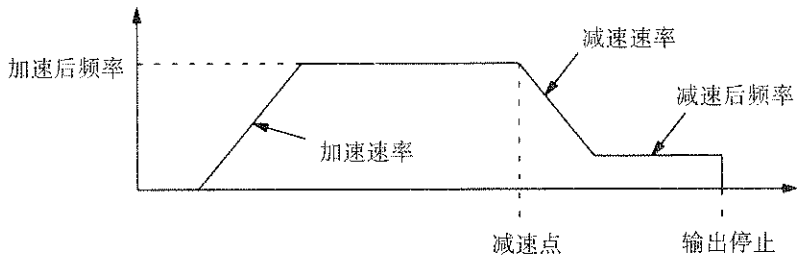
ACC(—) 与 PULS(65) 一起用来控制从端口 1 或 2 输出脉冲的加速和/或减速,可用的 4 种模式简述如下。

控制字功能在 4 种模式中是不同的,但 P 总是指定输出脉冲的端口, M 总是指定模式。设置 P = 001 或 002 指定端口 1 或 2, M = 000 至 003 指示模式 0 至 3。

注 详见 1—5 脉冲输出功能。

模式 0(M = 000)

模式 0 用来从端口 1 或 2 输出指定数量的 CW 或 CCW 脉冲,加速速率,加速后频率,减速点,减速速率,减速后的频率都可以控制。



PULS(65) 操作数设置

PULS(65) 必须在 ACC(—) 之前执行,以指定方向,输出脉冲总数及减速点。
PULS(65)操作数的功能说明如下,详见 5—28—9 设置脉冲 PULS(65)。

1, 2, 3...

1. PULS(65) 的第一个操作数指定输出端口,若 P = 001, 脉冲从端口 1 输出,若 P = 002 则从端口 2 输出。

2. 第二个操作数指定方向, 若 C = 002, 输出为顺时针方向 (CW), 若 C = 003 为逆时针 (CCW)。
3. 第三个操作数指定 4 个控制字的首字。
 - a) N + 1, N 的 8 位数字值 (00000001 至 16777215) 决定了将要输出的脉冲总数。
 - b) N + 3, N + 2 的 8 位数字值 (00000001 至 16777215) 决定了减速点。

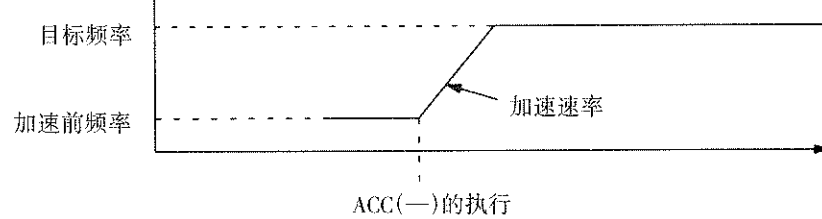
ACC(一)控制字

这 4 个控制字表示加速速率, 加速后频率, 减速速率, 减速后频率。

- 1, 2, 3... 1. C 的内容决定加速速率。加速期间, 输出频率每 4.08ms 一次增加设在 C 中的数值, C 必须是 0001 至 0200 间的 BCD 值 (10Hz 至 2KHz)。
2. C + 1 的内容指定加速后频率, C + 1 必须是从 0000 至 5000 间的 BCD 值 (0Hz 至 50KHz)。
3. C + 2 的内容决定减速速率, 减速期间, 输出频率在每 4.08ms 减少设在 C + 2 中的数值。C 必须是 0001 至 0200 间的 BCD 值 (10Hz 至 2KHz)。
4. C + 3 的内容指定减速后的频率, C + 3 必须是从 0000 至 5000 间的 BCD 值 (0Hz 至 50KHz)。

模式 1 (M = 001)

模式 1 用来将正在输出的脉冲按指定的速率增加频率至目标频率, 脉冲输出继续到停止

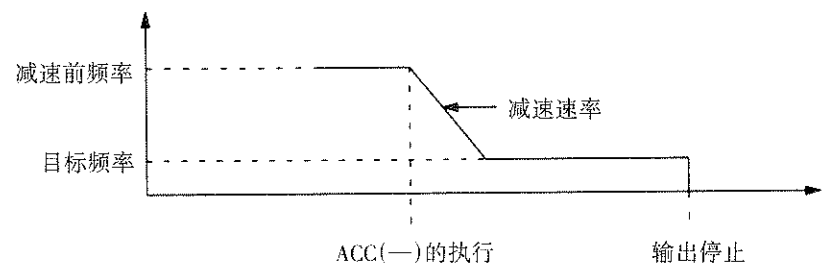


用 2 个控制字表示加速速率和目标频率。

- 1, 2, 3... 1. C 的内容决定加速速率, 加速期间, 每 4.08ms 一次根据设在 C 中的值增加输出频率。C 必须是 0001 ~ 0200 间的 BCD 值 (10Hz 至 2KHz)。
2. C + 1 的内容规定目标频率。C + 1 必须是 0000 至 5000 间的 BCD 值 (0Hz 至 50KHz)。

模式 2 (M = 002)

模式 2 用来将正在输出的脉冲的频率按指定的速率减少至目标频率, 当输出脉冲总数达到 PULS(65) 中指定的值时输出停止。



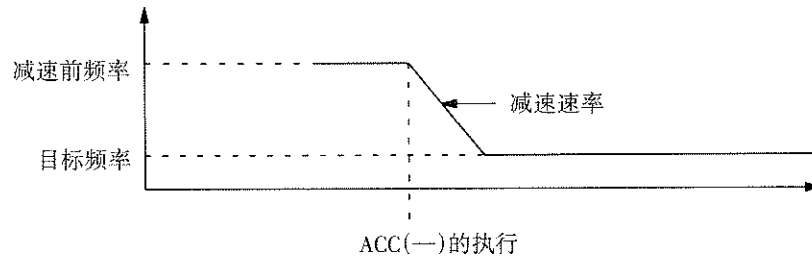
用 2 个控制字表示减速速率和目标频率。

- 1, 2, 3... 1. C 的内容决定减速速率, 在减速期间, 每 4.08ms 一次输出频率减少设在 C 中的值。C 必须是 0001 至 0200 间的 BCD 值 (10Hz 至 2KHz)。

模式 3(M = 003)

2. C + 1 的内容指定目标频率,C + 1 必须是 0000 至 5000 间的 BCD 值(0Hz 至 50KHz)。

模式 3 用来将正在输出的脉冲的频率按指定的速率减少至目标频率。脉冲输出继续到停止。



用 2 个控制字表示减速速率和目标频率。

1, 2, 3...

1. C 的内容决定减速速率,在减速期间,每 4.08ms 一次输出频率减少设在 C 中的值。C 必须是 0001 至 0200 间的 BCD 值(10Hz 至 2KHz)。

2. C + 1 的内容指定目标频率.C + 1 必须是 0000 至 5000 间的 BCD 值(0Hz 至 50KHz)。

标志

ER:间接寻址的 EM/DM 字不存在。

(* EM/* DM 字内容不是 BCD,或超出 EM/DM 区边界。)

操作数设置有错。

没有安装脉冲 I/O 板时执行 ACC(-)。

PC 设置没有设置脉冲输出。

指定输出端口正在使用而 ACC(-)以 M = 000 执行。

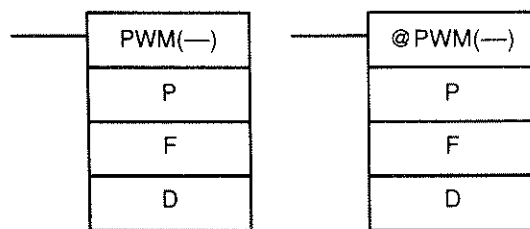
主程序正在执行脉冲 I/O 或高速计数器指令时,中断子程序执行了 ACC(-)。

AR0515: 端口 1 输出标志。当脉冲正从端口 1 输出时置 ON。

AR0615: 端口 2 输出标志。当脉冲正从端口 2 输出时置 ON。

5-28-13 可变占空率脉冲—PWM(-)

梯形图符号



操作数数据区

| |
|--|
| P: 通信端口 |
| 001 或 002 |
| F: 频率 |
| 000, 001, 或 002 |
| D: 占空率 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |

限制

除非 PC 设置 (DM6643 或 DM6644) 被设置为有可变占空率脉冲输出, 否则不能使用 PWM(-)。

P 必须是 001 或 002, F 必须是 000, 001 或 002。

D 必须是 0001 至 0099 间的 BCD 值。

说明

PWM(一)可用于下表所列的功能。

| 单元/板 | 功能 |
|--------|---------|
| 脉冲I/O板 | 脉冲输出1和2 |

PWM(一)用来从端口 1 或 2 输出指定占空率的脉冲。输出频率可设为三种中的一种: 5.9KHz, 1.5KHz, 或 91.6Hz。脉冲连续输出直到执行 INI(一) 来停止。

PC 设置必须将可变占空率脉冲输出设置给指定端口, 以便 PWM(一) 能执行。将 DM6643 最左位数设为 1, 使得端口 1 能输出可变占空率脉冲, 将 DM6644 最左位数设为 1, 使得端口 2 能输出可变占空率脉冲。设置为可变占空率的端口不能输出常规的脉冲。

注 详见 1—5 脉冲输出功能。

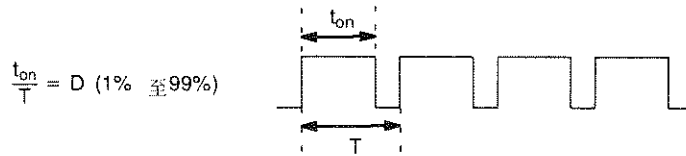
操作数设置

P 指定要输出脉冲的端口, 当 P = 001, 脉冲从端口 1 输出, 当 P = 002, 脉冲从端口 2 输出。

F 指定脉冲输出的频率, 如下表所示。

| F | 频率 |
|-----|---------|
| 000 | 5.9 kHz |
| 001 | 1.5 kHz |
| 002 | 91.6 Hz |

D 指定输出脉冲的占空率, 即输出为 ON 时间的百分比。D 必须是 0001 至 0099 之间的 BCD 值(1% 至 99%)。下图中的占空率为 75%。



标志

ER: 操作数设置有错。

没有安装脉冲 I/O 板时执行 PWM(一)。

PC 设置没有设置为可变占空率脉冲输出。

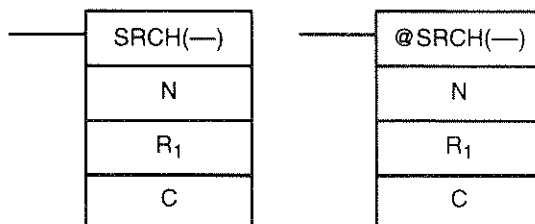
主程序正在执行脉冲 I/O 或高速计数器指令时, 中断子程序执行了 PWM(一)。

间接寻址的 EM/DM 字不存在。

(* EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

5-28-14 数据搜索—SRCH(一)

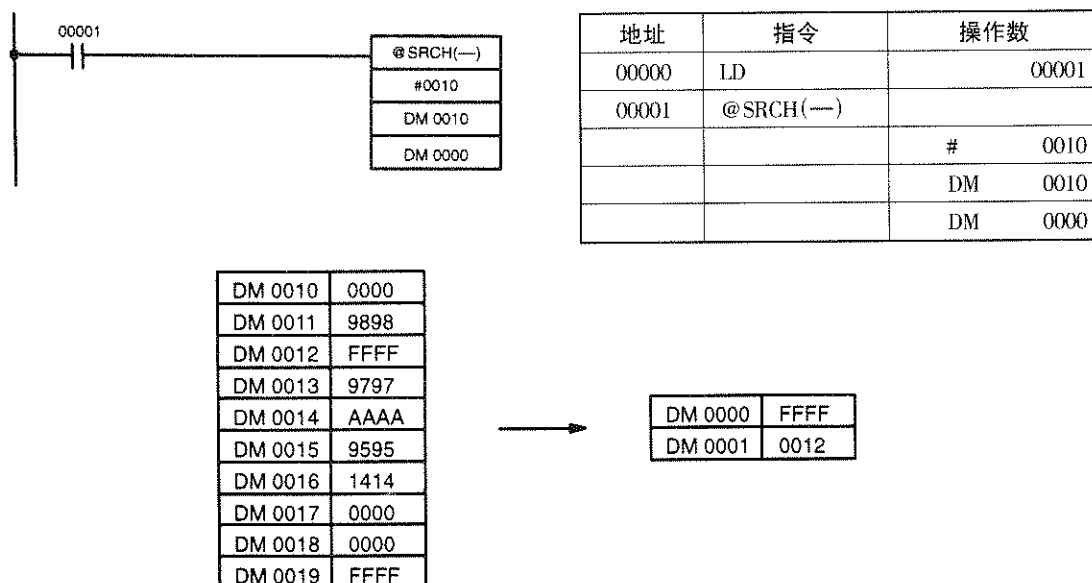
梯形图符号



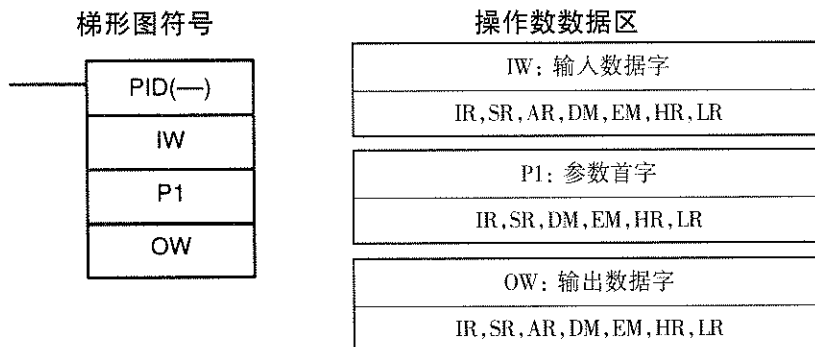
操作数数据区

| |
|--|
| N: 字数 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| R ₁ : 范围的首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| C: 比较数据, 结果字 |
| IR, SR, AR, DM, EM, HR, LR |

- 限制** N 必须是 0001 至 9999 间的 BCD 值。
 R_i 和 $R_i + N - 1$ 必须在同一数据区。
 DM6143 ~ DM6655 不能用于 C。
- 说明** 当执行条件为 OFF 时,SRCH(—)不执行。当执行条件为 ON 时,SRCH(—)搜索从 $R_i \sim R_i + N - 1$ 范围的内存,寻找包含有 C 中的比较数据的地址。若多于一个的地址含有比较数据,则 EQ 标志 (SR25506) 置 ON, 包含有比较数据的最低地址在 C + 1 中标记,地址标记对 DM 区不同。
- 1,2,3... 1. 若地址在 DM 区,该字地址写到 C + 1。例如,若包含比较数据的最低地址是 DM0114,则 # 0114 写到 C + 1。
 2. 在其它数据区的地址,从搜索开始处的地址偏移量被写到 C + 1,例如,若包含比较数据的最低地址是 IR114,而搜索范围的首字是 IR014,则把 # 0100 写到 C + 1。
 若范围中没有包含比较数据的地址,EQ 标志 (SR25506) 置 OFF,C + 1 保持不变。
- 标志** ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界。)
 N 不是 0001 ~ 9999 间的 BCD 值。
 EQ: 当在搜索范围中有与比较数据匹配的值时置 ON。
- 例** 在下列中,从 DM0010 至 DM0019 的 10 个字范围中,搜索包含与 DM0000 中有相同数据 (# FFFF)的地址。因为 DM0012 包含了该数据。EQ 标志 (SR25506) 置 ON,# 0012 被写到 DM0001。



5-28-15 PID 控制—PID(—)



限制

DM6144 ~ DM6655 不能用于 IW, P1 ~ P1 + 32, 或 OW。

P1 ~ P1 + 32 必须在同一数据区。

注意 必须提供 PID(—) 33 个从 P1 起始的连续字以正确工作。在下列情况下, PID(—) 不能工作: 在中断程序中, 子程序中, IL(02) 和 ILC(03) 之间, JMP(04) 和 JME(05) 之间, 在步程序 (STEP(08)/SNXT(09)) 中。在这些情况下, 不要用 PID(—)。

说明

PID(—) 根据在 P1 至 P1 + 6 中指定的参数完成 PID 控制, 在 IW 中的数据用来计算写到 OW 的输出数据, 下表说明参数字的功能。

| 字 | 位 | 参数名 | 功能/设置范围 |
|------------------|---------|---------|--|
| P1 | 00 ~ 15 | 设定值(SV) | 这是PID控制的目标值, 可按输入范围参数设定的位数设为任意的二进制值。 |
| P1 + 1 | 00 ~ 15 | 比例带宽 | 该参数规定比例带宽与输入范围之比, 从0.1% ~ 999.9%。它必须是0001至9999的BCD值。 |
| P1 + 2 | 00 ~ 15 | 积分时间 | 设置积分时间与采样周期的比值, 用于积分控制, 它必须是0001至8191间的BCD值, 或9999(9999取消积分控制)。 |
| P1 + 3 | 00 ~ 15 | 微分时间 | 设置微分时间/采样周期之比, 用在微分控制, 它必须是0001到8191之间的BCD值, 或0000。 |
| P1 + 4 | 00 ~ 15 | 采样周期 | 设置两次输入数据采样的时间间隔, 为0.1至102.3秒间的值, 它必须为0001至1023之间的BCD值。 |
| P1 + 5 | 00 ~ 03 | 运算指定 | 设置倒相或正常运算, 设为0规定倒相运算, 为1规定正常运算。 |
| | 04 ~ 15 | 输入滤波系数 | 决定输入滤波的强度, 系数越小, 滤波越弱。它必须是100至199, 或000的BCD值, 000设置为缺省值(0.65)。设为100至199对应系数为0.00至0.99。 |
| P1 + 6 | 00 ~ 07 | 输出范围 | 决定输出数据的位数, 它必须是00至08间的值, 对应设置输出范围是8至16位。 |
| | 08 ~ 15 | 输入范围 | 决定输入数据的位数, 它必须是00至08间的值, 对应设置输入范围是8至16位。 |
| P1 + 7 ~ P1 + 32 | 00 ~ 15 | 工作区 | 未使用(由系统使用) |

当执行条件为 OFF 时, PID(—) 不执行, 指令的数据保持不变, 在执行条件为 OFF 期间, 可以手动地将所需输出数据直接写到 OW。

当执行条件首次从 OFF 变为 ON 时, PID(—) 读参数并初始化工作区, 在启动后有一个内部功能连续地改变输出数据, 因为输出数据的突然变化会对受控系统产生不利影响。

注意 对参数的改变直到 PID(一)的执行条件从 OFF 变 ON 时才有效。

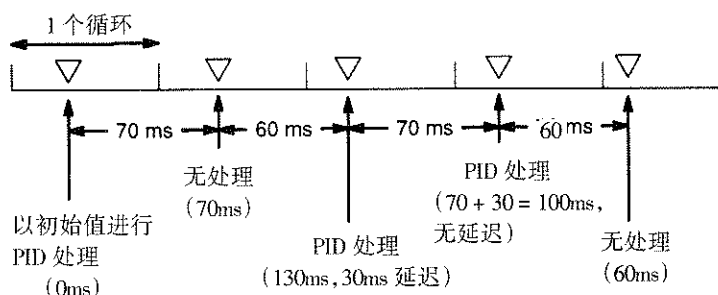
注 下列情况下不使用 PID(一),因为它可能不能正确执行。

- 在中断程序中
- 在子程序中
- 在互锁程序部分(IL 和 ILC 之间)
- 在跳转程序部分(JMP 和 JME 之间)
- 在步程序部分(由 STEP 创建)

当执行条件为 ON 时, PID(一) 在采样周期过去之后进行对输入数据的 PID 计算。采样周期是为处理读输入数据之前必须经过的时间。

下图说明采样周期与 PID 处理的关系。

仅在采样周期(此处为 100ms)已经到时进行 PID 处理。



标志

ER: 参数设置有错误。

循环时间长于 2 倍的采样周期, 所以 PID(一) 不能精确地执行, 在这种情况下 PID(一) 仍将执行。

间接寻址的 EM/DM 字不存在。

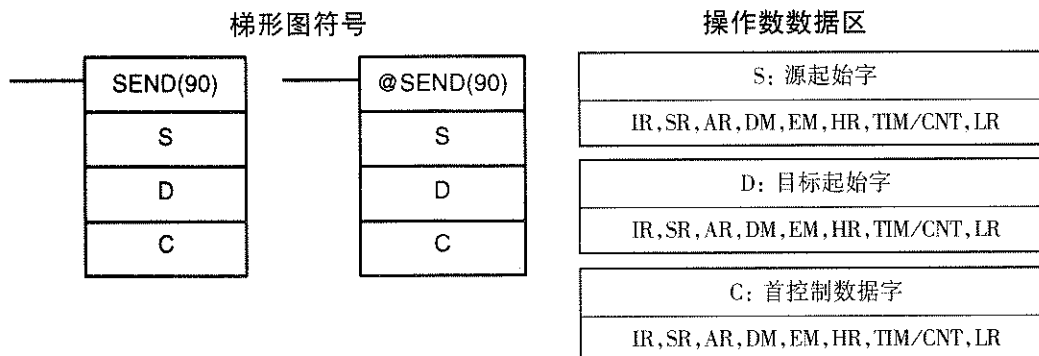
(* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

CY: 当 PID 处理完成时置 ON。(当采样周期没到时置 OFF。)

5 - 29 网络指令

网络指令用于与通过 Controller Link 系统链接的其它 PC、上位机的通信。

5 - 29 - 1 网络发送—SEND(90)



限制

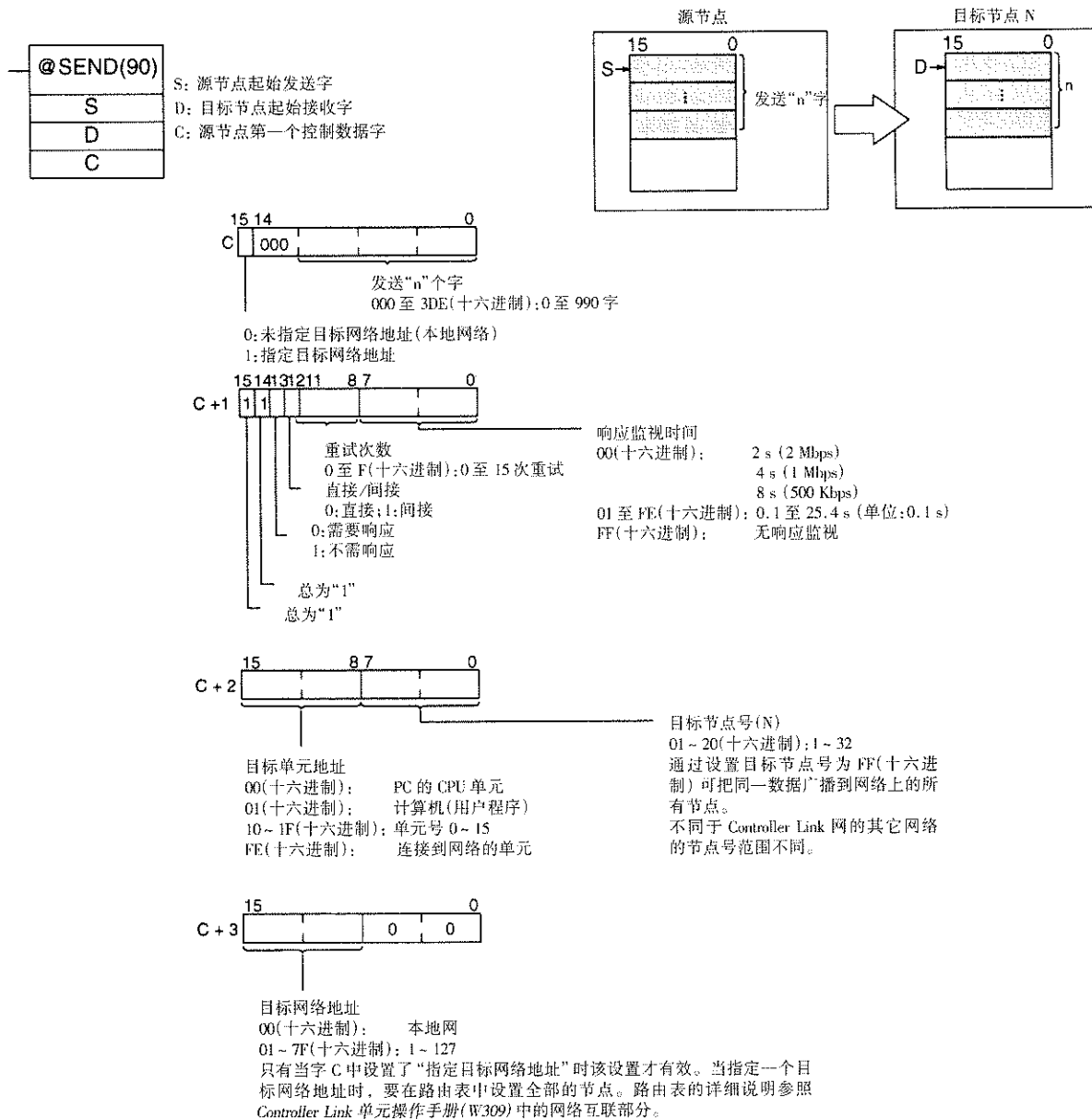
C 至 C + 2 必须在同一数据区, 并且必须在指定的数值范围内, 要使用 SEND (90), 系统必须安装 Controller Link 单元。

说明

当执行条件为 OFF 时, SEND(90) 不执行。当执行条件为 ON 时, SEND(90) 把以 S 为起始字的数据传送到由 D 指定的 Controller Link 系统指定节点的地址中。以 C 起始的控制字, 指定发送的字数, 目标节点和其它参数。

控制字

SEND(90) 传送以 S 起始的“n”个字(源节点中用于数据传输的源起始字)到以 D 起始的“n”个字中(目标节点 N 中用于数据接收的目标起始字)。



执行 SEND(90) 的同时也就启动了通信单元的数据传输。为了检查是否实际完成了传输, 验证网络指令允许标志(AR0209)已从 OFF 变 ON 以及网络指令错误标志(AR0208)为 OFF。当执行到 END(01) 时, 传输过程即结束。

如果要求应答却在应答监视时间内未收到应答信号,将再次进行数据传输,直到收到应答信号或到达指定的重试次数为止(最多 15 次)。

当目标节点号设置为 FF,则同一数据广播到指定网络的所有节点上。当指定为广播传输时,将不返回应答,并不重新传输。

在执行 SEND(90)时,如果网络指令允许标志(AR0209)为 OFF,则指令等同于 NOP(00),并不执行,这时将发生错误,错误标志变 ON。

在执行 SEND(90)时,如果网络指令允许标志(AR0209)为 ON,网络指令错误标志(AR0208)和网络指令允许标志(SR0209)将变 OFF,网络指令完成代码将置为 00,数据将发送到网络的节点上。

当在目标起始字(D)中指定了一个当前区的 EM 地址时,传输数据将写入目标节点的当前 EM 区,当数据传送到比 CQM1H 数据区更大的 PC 中时,例如 CS1 系列或 CV 系列的 PC,目标起始字(D)中可使用间接寻址。间接寻址也可用于改变目标起始字。

如果数据要传送到其它网络的节点中,每一个网络中在 PC(CPU 单元)中都必须登记路由表。(路由表指明目标节点所连的其它网络的路由)。

一次只能执行一个网络指令。为了保证第一个指令执行后再执行第二个网络指令,程序中把网络指令允许标志(AR0209)作为一个常开条件。

当数据正在传输时以及网络指令允许标志是 OFF 时,不要改变控制数据(C 至 C+3)。

噪音和其它因素能导致传输或应答被破坏或丢失,所以建议设置重试次数为非 0 的数值,这样当在应答监视时间内未收到应答时,将再次执行 SEND(90)指令。

间接指定目标起始字

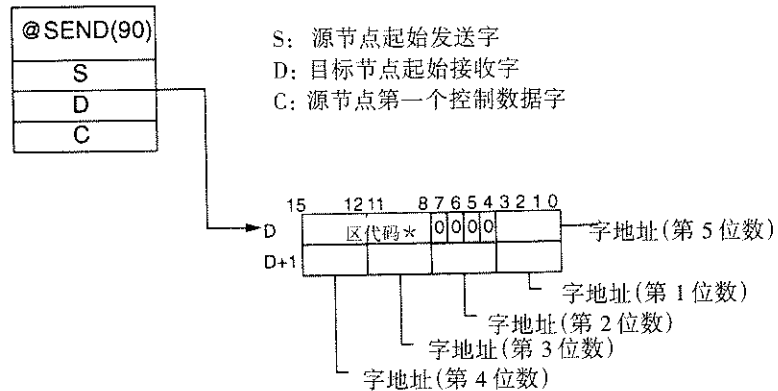
当使用间接指定时 D 按以下所示指定目标起始字:

| 字 | 位12~15 | 位08~11 | 位04~07 | 位00~03 |
|-----|---------------|---------------|---------------|---------------|
| D | 区域类型 | | 0 | 字地址 (第5位数) |
| D+1 | 字地址 (第4位数) | 字地址 (第3位数) | 字地址 (第2位数) | 字地址 (第1位数) |

CS1 系列 PC 和 CV 系列 PC 的数据区比 CQM1H 大,因此目标节点发送和接收的起始字不全能用 SEND(90)和 RECV(98)操作数来直接指定。另外,根据情况也可能需要改变目标节点的起始字。

该情况下,“直接/间接”控制数据设置成“1”(间接)并按以下所述指定发送的起始字。

起始接收字由目标节点 D 和 D + 1 字中的内容所决定。



S: 源节点起始发送字
D: 目标节点起始接收字
C: 源节点第一个控制数据字

注 根据下表指定区代码。

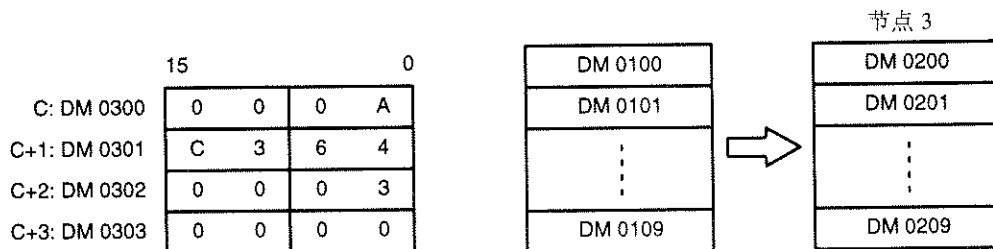
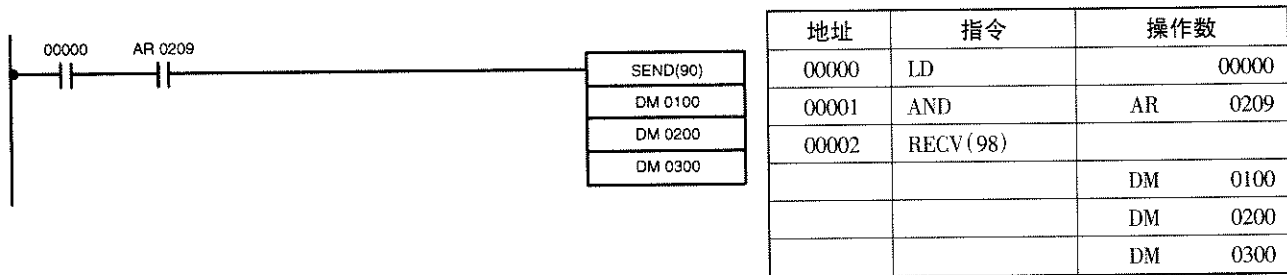
| 目标节点: CS1系列PC | | 目标节点: CQM1H, C200HX/HG/HE PC | | 目标节点: CV系列PC | |
|------------------|-------|---------------------------------|----------------------|----------------------|----|
| 区域 | 代码 | 区域 | 代码 | 区域 | 代码 |
| CIO | 00 | IR | 00 | CIO | 00 |
| 定时器(见注1) | 03 | LR | 06 | CPU总线链接 | 01 |
| 计数器(见注2) | 04 | HR | 07 | 辅助区 | 02 |
| DM | 05 | AR | 08 | 定时器 | 03 |
| EM | 0~7区 | 定时器/计数器 | 03 | 计数器 | 04 |
| | 8~15区 | DM | 05 | DM | 05 |
| | 当前区 | EM | 0~7区 8~15区 当前区 | 10~17 28~2F 18 | EM |

注 1. IR 区中 0~2555 字能发送和接收数据。

2. 0~2047 号定时器/计数器能发送和接收数据。

例

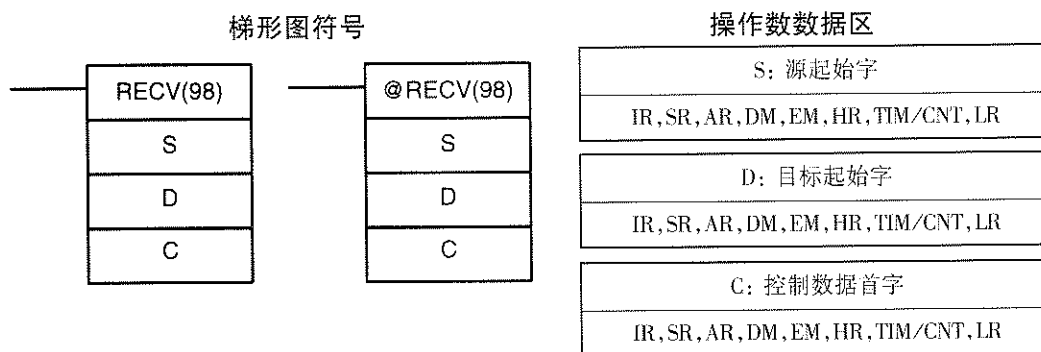
下例中当 IR00000 和 AR0209(网络指令允许标志)为 ON, DM 0100 至 DM0109 中的 10 个字被传送到本地网 3 号节点中从 DM0200 至 DM0209 的 10 个字中。如果 10 秒之内未接收到应答,数据将传送多达 3 次。



标志

ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)
 Controller Link 单元的发送字数超出 990 字。
 未安装 Controller Link 单元。
 源字超出数据区边界。

5-29-2 网络接收—RECV(98)



限制

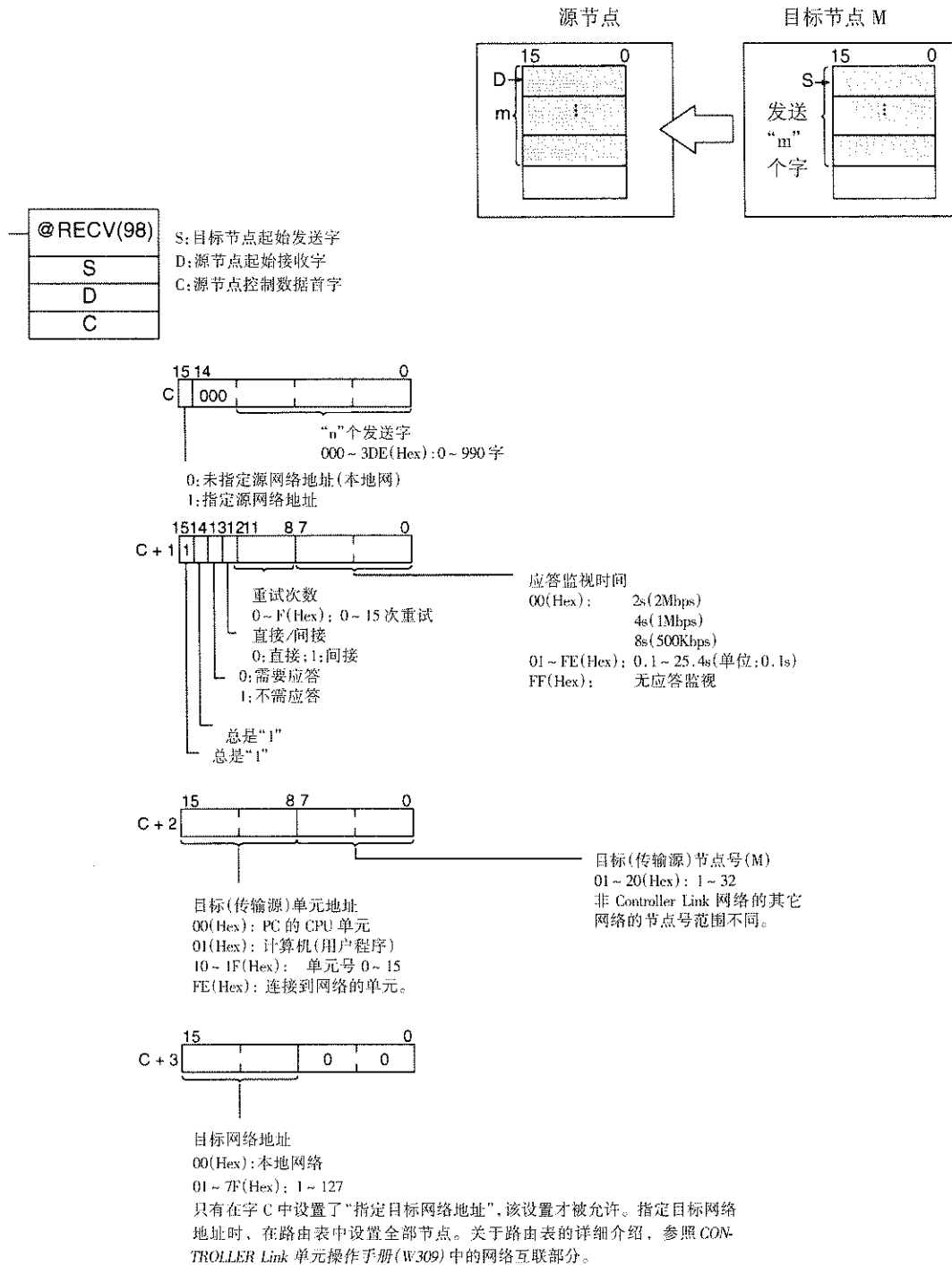
C 至 C + 2 必须在同一数据区，并且必须在指定数值范围之内。使用 RECV(98)之前必须安装 Controller Link 单元。

说明

在执行条件为 OFF 时，RECV(98) 不执行，执行条件为 ON 时，RECV(98) 把 Controller Link 系统的节点起始字 S 的数据传输到起始字 D 中。以 C 起始的控制字提供接收的字数，源节点以及其它传输参数。

控制字

RECV(98) 接收以 S 起始的“m”字(目标节点 M 用于数据传输的起始字)到 D 字(源节点用于接收数据的起始字)中。



执行 RECV(98) 就是起动通信单元的数据接收。要检查是否执行了接收, 检查网络指令允许标志 (AR0209) 是否从 OFF 变 ON, 并且网络指令错误标志 (AR0208) 为 OFF。接收过程在执行到 END(01) 时结束。

RECV(98) 需要应答, 因为应答中包含接收的数据, 因此把 C + 1 的 13 位设置为 "0" 来表示需要应答。如果在 C + 4 中设置的应答监视时间之内未收到响应, 则将会重新请求传输数据, 直到收到响应为止或达到重试次数 (最多 15 次) 为止。

如果网络指令允许标志 (AR0209) 为 OFF 时执行 RECV(98), 指令将作为 NOP (00) 看待而不被执行。将会出现错误且错误标志变 ON。

如果网络指令允许标志 (AR0209) 为 ON 时执行 RECV(98), 网络指令错误标志 (AR0208) 和网络指令允许标志 (AR0209) 将变 OFF, 网络指令完成代码将设置为 00, 将从其它节点接收数据。

一次只能执行一个网络指令。为了确保第一个指令执行完之后再执行第二个网络指令, 程序中将网络指令允许标志 (AR0209) 做为一个常开条件。

当数据正在被接收时, 并且网络指令允许标志为 OFF 时, 不要改变控制数据 (C 至 C + 3)。

噪音和其它因素能导致传输或响应遭到破坏或被丢失, 所以建议设置重试次数为非 0 的数值, 这样如果在响应监视时间内未收到响应时, 将再次执行 RECV(98) 指令。

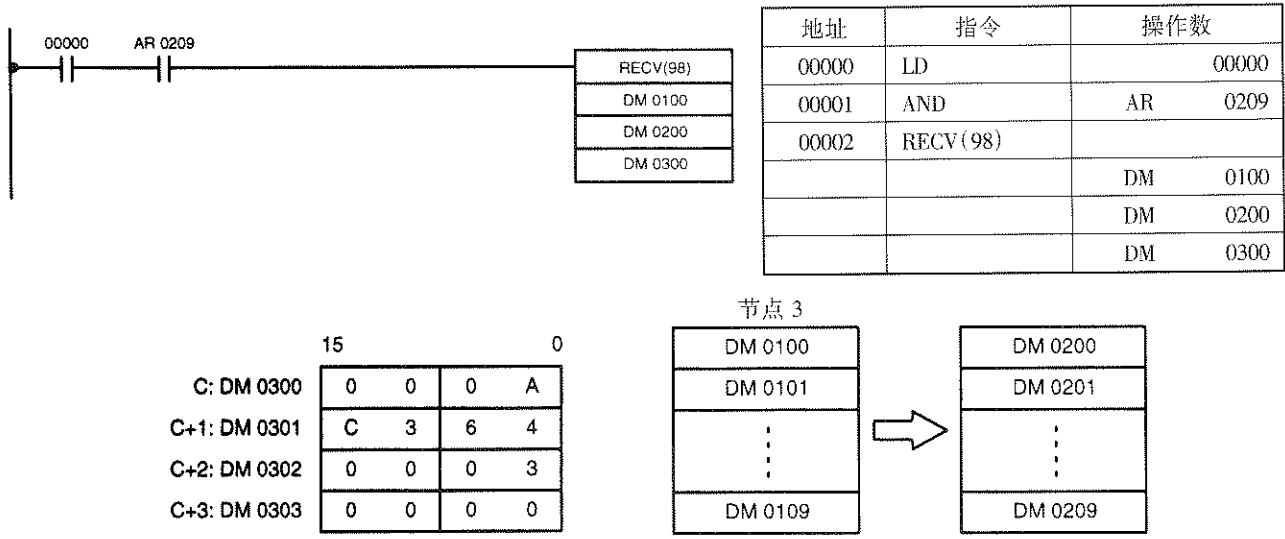
当 PC (例如 CS1 系列或 CV 系列) 的数据区比 CQM1H 大时, 接收来自该 PC 的数据时, 源起始字 (S) 可以使用间接寻址方式。间接寻址还可以用来改变源起始字以适应各种状况。

间接指定源起始字

当需要间接指定时, S 用于指定源起始字。使用的指定方法与指定 SEND(90) 的目标起始字相同。

例

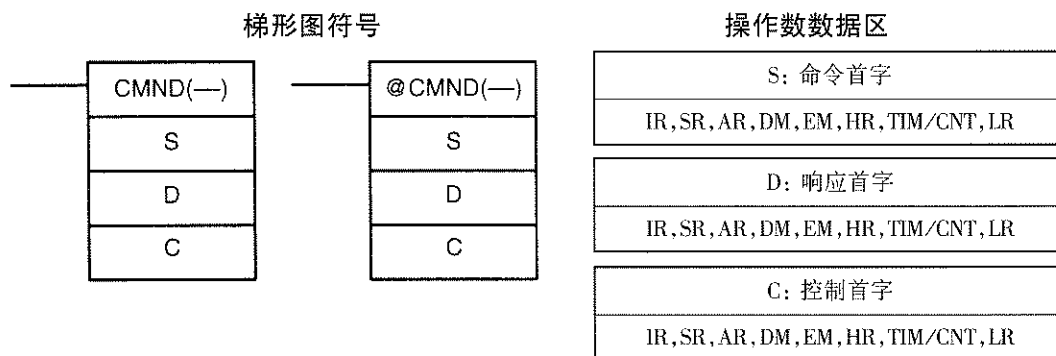
下例中当 IR00000 和 AR0209 (网络指令允许标志) 为 ON 时, 本地网 3 号节点中从 DM0100 到 DM0109 的 10 个字中的数据被接收并写入到从 DM0200 至 DM0209 的 10 个字中。如果 10 秒之内未接收到响应, 数据传输请求将传送达 3 次。



标志

ER: 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界。)
 Controller Link 单元的发送字数超过 990 个字。
 未安装 Controller Link 单元。
 接收数据超出数据区边界。

5-29-3 发布命令:CMND(—)

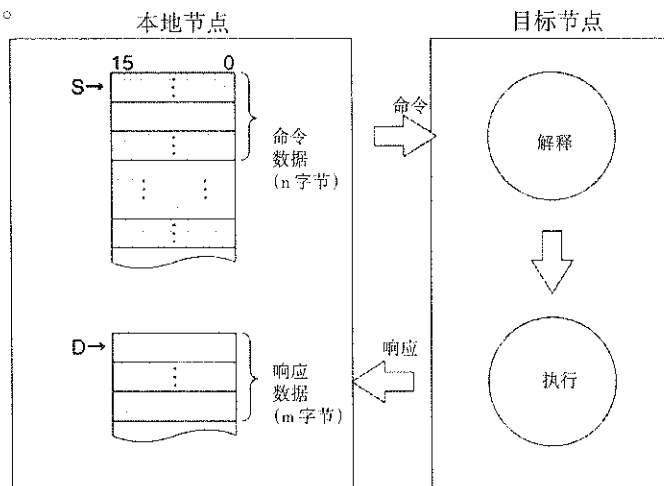


限制

C 至 C + 5 必须在同一数据区, 并且必须在下面指定的数值范围内。使用 CMND(—)之前必须安装 Controller Link 单元。

说明

CMND(—)在执行条件为 OFF 时不执行。当执行条件为 ON 时,CMND(—)把 S 字起始的 FINS 命令传送到 Controller Link 系统的指定节点中, 并接收响应。



控制字

C 至 C + 5 的六个控制字指定了命令数据和响应数据的字节数, 目标, 其它设置, 如下表所示。

| 字 | 位00 ~ 07 | 位08 ~ 15 |
|-------|--|----------|
| C | 命令数据的字节: 0000 ~ 07C6十六进制(0 ~ 1990字节) | |
| C + 1 | 响应数据的字节: 0000 ~ 07C6十六进制(0 ~ 1990字节) | |
| C + 2 | 目标网络地址 00: 本地网络 01 ~ 7F: 网络1 ~ 127 | 总是00 |

| 字 | 位00 ~ 07 | 位08 ~ 15 |
|-------|--|--|
| C + 3 | 目标单元地址 00: CPU单元 01: 计算机(用户程序) 10 ~ 1F: 单元0 ~ 15 E1: 内板 FE: 连接到网络的单元 | 目标节点号 01 ~ 20: 1 ~ 32(见注1) EF: 广播(见注2) |
| C + 4 | 重试次数: 00 ~ 0F(0 ~ 15) | 响应设置 00: 需要响应 80: 不需要响应 |
| C + 5 | 响应监视时间 0000: 2s当2Mbps, 4s当1Mbps或8s当500Kbps 0001 ~ FFFF: 0.1 ~ 6,553.5秒(单位0.1s) | |

- 注 1. Controller Link 的允许范围是十六进制 01 ~ 20(1 ~ 32), 但最大节点号不同于其它网络。
2. 目标节点号设置成 FF, 则命令被广播到网络的所有节点上。

执行 CMND(一) 时就起动通信单元的 FINS 命令的传输。为了检查是否实际完成了传输, 只要检查网络指令允许标志(AR0209)已从 OFF 变 ON 且网络指令错误标志(AR0208)为 OFF。当 END(01)执行时命令传送过程结束。

如果需要应答但在响应监视时间内未收到应答, 命令将重复发布, 直到收到响应或达到指定重试次数(达 15 次)为止。在发布不产生响应的命令时, 确认指出不需要响应。

当目标节点号设置成 FF 时, 同一个命令广播到指定网络的所有节点上, 当指定广播传送时, 将不返回响应, 传送也不会重新发生。

如果响应数据的数量超出 C + 1 中设置的响应数据的字节数时, 将产生错误。

如果 CMND(一) 执行时网络指令允许标志(AR0209)为 OFF, 指令将等同于 NOP(00), 且不被执行。将产生错误且错误标志将变 ON。

如果 CMND(一) 执行时网络指令允许标志(AR0209)为 ON, 网络指令错误标志(AR0208)和网络指令允许标志(AR0209)将变 OFF, 网络指令完成代码设置成 00, FINS 命令发布到网络节点中。

目标节点通过网络 PC 中注册的路由表定位。(路由表表明到达目标节点所连接的其它网络的路径。)

一次只能执行一个网络指令, 为了保证不同时执行两条网络指令, 程序中使网络指令允许标志(AR0209)做为常开条件。

当 FINS 命令正在处理以及网络指令允许标志为 OFF 时, 不要改变控制数据(C 至 C + 5)。

噪声和其它因素能导致发送或响应破坏或丢失, 因此建议设置重试次数为非 0 的数值。使 CMND(一)在响应监视时间内未收到响应时能再次执行。

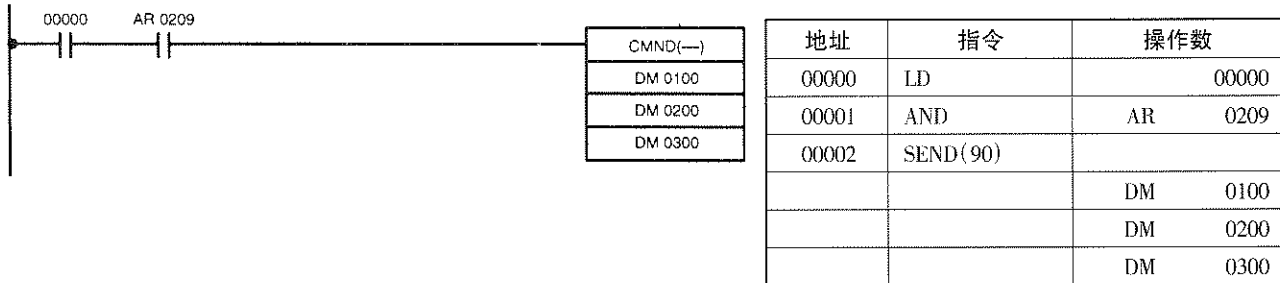
如果 FINS 命令代码为 0102(内存区写), 则 CMND(一)操作如同 SEND(90)命令, 如果代码为 0101(内存区读)则如同 RECV(098)。

例

下例中当 IR00000 和 AR0209(网络指令允许标志) 为 ON, CMND 发布 FINS 命令 0101(内存区读)到本地网络的节点 3 中。

内存区读命令读 10 字 (从 DM0010 ~ DM0019), 响应包含 2 字节命令代码 (0101), 2 字节完成代码, 和 10 字数据, 总共 12 个字或 24 字节。

如果 10 秒之内未接收到响应, 该命令将再次发送 3 次。



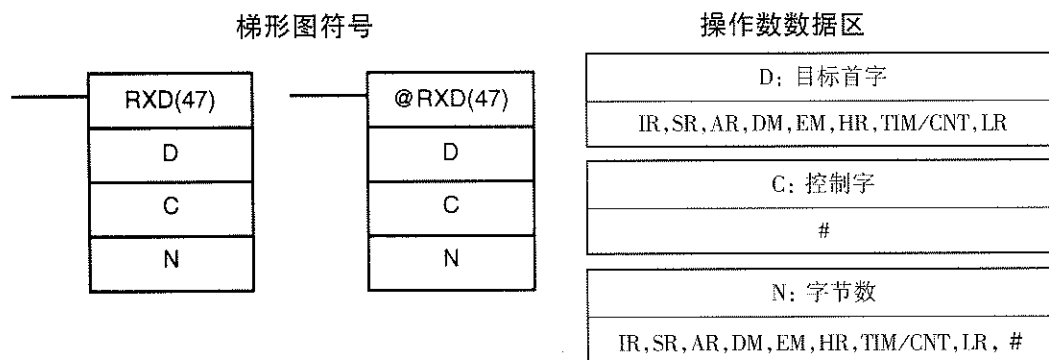
标志

ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

5 - 30 通信指令

5 - 30 - 1 接收—RXD(47)



限制

D 和 $D + (N \div 2) - 1$ 必须在同一数据区。

DM6144 ~ DM6655 不能用于 D 或 N。

N 必须是 # 0000 ~ # 0256 间的 BCD 值。

说明

当执行条件为 OFF 时, RXD(47) 不执行。当执行条件为 ON 时, RXD(47) 从控制字指定的端口读出接收数据的 N 个字节, 然后将其写至字 D ~ $D + (N \div 2) - 1$ 。

一次最多能读 256 个字节的数据。

若接收了少于 N 个的字节, 则将读到已接收的数量。

注 关于 RXD(47) 指令的使用, 在 PC 设置中设置通信协议等等, 详见 4-6 通信功能。如果没有用 RXD(47) 读出已接收的数据, CQM1H 在接收到 256 字节后就不能接收更多的数据。在接受完成标志置 ON 后尽快读入数据。下表列举了各端口的接收完成标志。

| 端口 | | 接收完成标志 |
|---------------------|-----|----------|
| CPU单元的内置RS - 232C端口 | | AR 0806 |
| 外围端口 | | AR 0814 |
| 串行通信板 | 端口1 | IR 20106 |
| | 端口2 | IR 20114 |

RXD(47)的 N 设置为 0000 时可以清除通信标志和计数器。

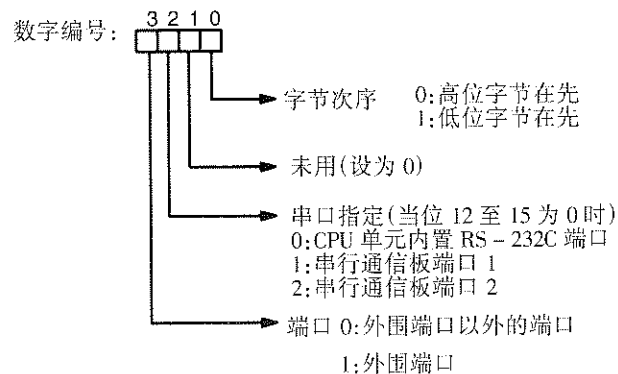
相关标志和控制位

下表列举了使用 RXD(47)接收数据时使用的各种标志,控制位和字

| 端口 | 标志 | 功能 |
|--------------------|----------|--|
| CPU单元的内置 RS-232C端口 | AR 0806 | 接收完成标志在接收结束时变ON,用RXD(47)读取数据时变OFF。 |
| | AR 09 | 包含以4位BCD表示的接收字节数, RXD(47)读取数据时该字被清为0000。 |
| | SR 25209 | 使RS-232C端口复位位变ON以使RS-232C端口复位。 |
| 外围端口 | AR 0814 | 接收完成标志在接收结束时变ON,用RXD(47)读取数据时变OFF。 |
| | AR 10 | 包含以4位BCD表示的接收字节数, RXD(47)读取数据时该字被清为0000。 |
| | SR 25208 | 使外围端口复位位变ON以使外围端口复位。 |
| 串行通信板端口1 | IR 20106 | 接收完成标志在接收结束时变ON,用RXD(47)读取数据时变OFF。 |
| | IR 202 | 包含以4位BCD表示的接收字节数。用RXD(47)读取数据时该字被清为0000。 |
| | IR 20700 | 使端口1重启动位变ON,以复位端口1。 |
| 串行通信板端口2 | IR 20114 | 接收完成标志在接收结束时变ON,用RXD(47)读取数据时变OFF。 |
| | IR 203 | 包含以4位BCD表示的接收字节数。用RXD(47)读取数据时该字被清为0000。 |
| | IR 20701 | 使端口2重启动位变ON,以复位端口2。 |

控制字(C)

控制字的值决定读数据的端口以及将数据写入内存的次序。



把数据写到内存的次序依据 C 中数字 0 的值。8 字节的数据 12345678... 将按以下方式写入。

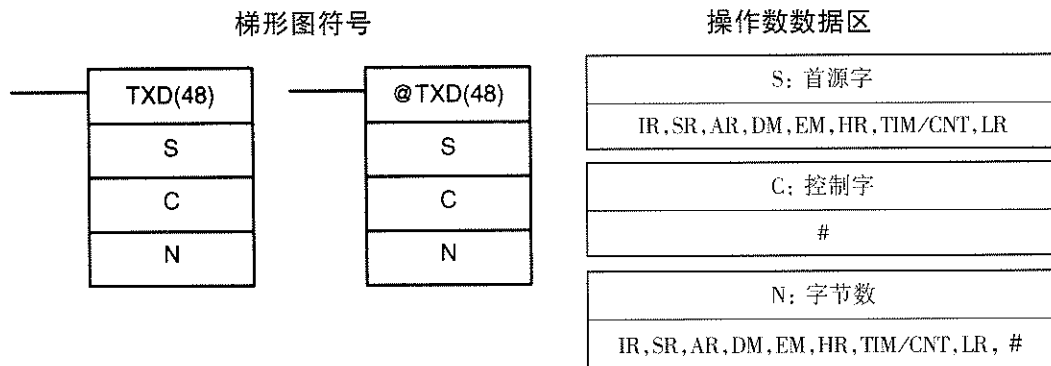
| | MSB | LSB |
|-----|-----|-----|
| D | 1 | 2 |
| D+1 | 3 | 4 |
| D+2 | 5 | 6 |
| D+3 | 7 | 8 |
| ⋮ | ⋮ | ⋮ |

| | MSB | LSB |
|-----|-----|-----|
| D | 2 | 1 |
| D+1 | 4 | 3 |
| D+2 | 6 | 5 |
| D+3 | 8 | 7 |
| ⋮ | ⋮ | ⋮ |

标志

ER: 指定串行通信板的端口,但串行通信板没安装。
 通信设置(PC 设置)或操作数设置有错误。
 间接寻址的 EM/DM 字不存在。
 (* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界)。
 目标字(D 至 D + (N ÷ 2) - 1)超出数据区。

5-30-2 传送—TXD(48)



限制

S 和 S + (N ÷ 2) - 1 必须在同一数据区。
 DM6144 ~ DM6655 不能用于 S 或 N。
 N 必须是 # 0000 ~ # 0256 间的 BCD 值。(上位机链接模式为 # 0000 至 # 0061)

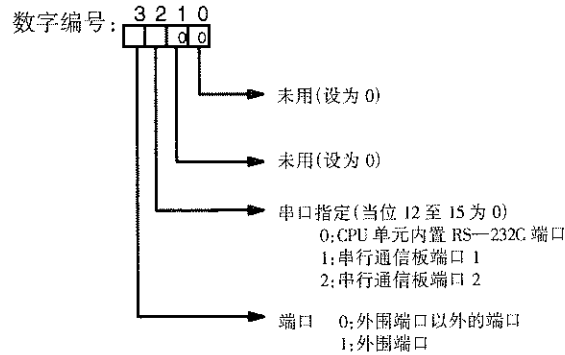
说明

当执行条件为 OFF 时, TXD(48) 不执行, 当执行条件为 ON 时, TXD(48) 从字 S 至 S + (N ÷ 2) - 1 中读出 N 个字节数据, 将其转换为 ASCII, 然后输出数据到指定端口, TXD(48) 在上位机链接模式和无协议模式中的运行是不一样的, 所以这些模式分别说明。

注 关于 TXD(48) 指令的使用, 在 PC 设置中设置通信协议, 等等, 详见 1-6 通信功能。

上位机链接模式

N 必须是 # 0000 至 # 0061 间的 BCD 值 (即最多有 122 个字节的 ASCII 字符)。控制字 (C) 的值决定了数据输出的端口,如下所示。



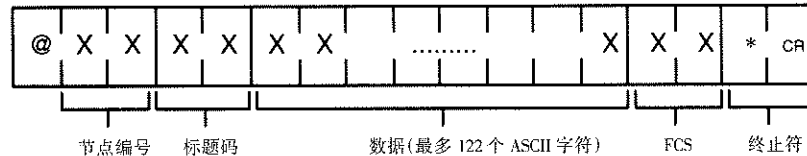
指定数量的字节将从 S 至 S(N/2) - 1 中读出,转换为 ASCII,通过指定端口传送,如下所示的源数据字节将按这样的次序传送:12345678...

| | MSB | LSB |
|-----|-----|-----|
| S | 1 | 2 |
| S+1 | 3 | 4 |
| S+2 | 5 | 6 |
| S+3 | 7 | 8 |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |

下表列出了各端口的传输允许标志。当 CQM1H 能通过某端口传送数据时,相应端口的传输允许标志将变 ON。

| 端口 | | 传输允许标志 |
|------------------|------|----------|
| CPU单元内置RS-232C端口 | | AR 0805 |
| 外围端口 | | AR 0813 |
| 串行通信板 | 端口 1 | IR 20105 |
| | 端口 2 | IR 20113 |

下图说明了上位机链接命令 (TXD) 从 CQM1H 发送的格式, CQM1H 自动加前缀和后缀,如节点编号,标题和 FCS。

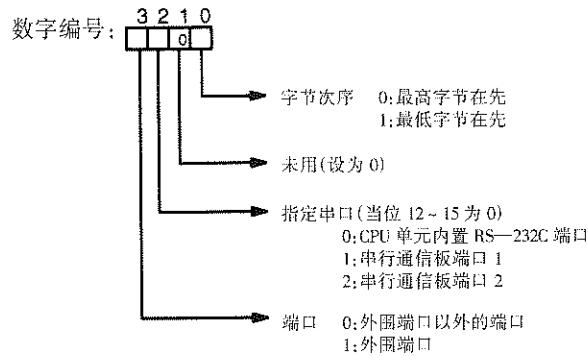


无协议模式

N 必须是 # 0000 至 # 0256 之间的 BCD 值,控制字的值决定数据从哪一端口输出,并以何种次序将数据写入内存中。

控制字(C)

控制字的值决定数据从何端口中被读取,并以何种次序写入到内存中。



指定的字节数将从 S 至 S + (N ÷ 2) - 1 中读取并从指定的端口传送。

| | MSB | LSB |
|-----|-----|-----|
| S | 1 | 2 |
| S+1 | 3 | 4 |
| S+2 | 5 | 6 |
| S+3 | 7 | 8 |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |

当 C 的数字 0 为 0 时,上面所示的源数据将按该次序传送:12345678...

当 C 的数字 0 为 1 时,上面所示的源数据将按该次序传送:21436587...

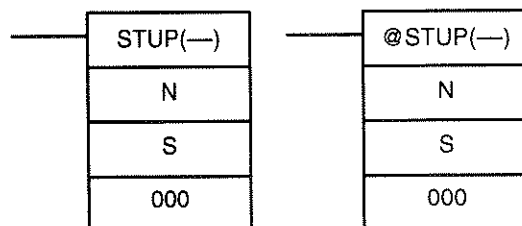
注 当指定了起始和结束码时,数据总长最大为 256 字节,包括起始和结束码(当同时指定起始码和结束码时,数据最大长度为 254 字节)。

标志

- ER: 指定了串行通信板上的端口,但串行通信板未安装。
- 通信设置(PC 设置)或操作数设置中有错误。
- 间接寻址的 EM/DM 字不存在。
- (*EM/*DM 字的内容不是 BCD,或超出 EM/DM 区边界)
- 源字(S ~ S + (N ÷ 2) - 1)超出数据区。

5 - 30 - 3 改变串行口设置 - STUP(—)

梯形图符号



操作数数据区

| |
|--|
| N: 端口指定 |
| IR000, IR001, IR002, 或 IR003 |
| S: 源首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR, # |
| 第三操作数: 置为 000 |
| — |

限制

N 必须是 IR000, IR001, IR002 或 IR003。
 S 和 S + 4 必须在同一数据区。
 (S 设置成 # 0000 时, RS-232C 为缺省设置)
 如果 DIP 开关的 5 脚为 ON 时, STUP(-) 不能用于 CPU 单元的内置 RS-232C 端口。
 STUP(-) 不能在中断程序中执行。

说明

STUP(-) 在执行条件为 OFF 时不执行。当执行条件为 ON 时, STUP(-) 改变 N 指定的端口的 PC 设置的设置。
 N 决定 RS-232C 设置的哪一部分被改变。

| N | 指定端口 |
|--------|--|
| IR 000 | 内置 RS-232C 端口 (PC 设置: DM 6645 ~ DM 6649) |
| IR 001 | 串行通信板端口 1 (PC 设置: DM 6555 ~ DM 6559) |
| IR 002 | 串行通信板端口 2 (PC 设置: DM 6550 ~ DM 6554) |
| IR 003 | 外围端口 (PC 设置: DM6650 ~ D M6654) |

如果 S 是一个字地址, S 至 S + 4 的内容被复制到 PC 设置的 5 个字中, 该设置包含 N 指定端口的设置。
 如果 S 输入常数 # 0000, 指定端口恢复为缺省设置。

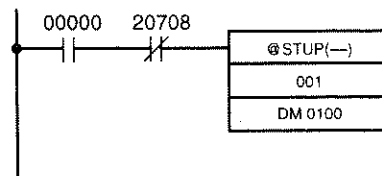
| S | 功能 |
|-------------|---|
| 字地址 | S 至 S + 4 的内容被复制到 PC 设置中 N 所指定端口部分所包含的设定。 |
| 常数 (# 0000) | N 指定端口的设定恢复为缺省设置。 |

下表列出了各端口的设置改变标志或协议宏执行标志。相应的标志当 STUP(-) 执行时保持为 ON, 当改变完成 时该标志变 OFF。

| 端口 | 标志名 | 标志地址 |
|---------------|-------------------------|----------|
| 内置 RS-232C 端口 | CPU 单元 RS-232C 端口设置改变标志 | AR 2402 |
| 外围端口 | CPU 单元外围端口设置改变标志 | AR 2403 |
| 串行通信板 | 端口 1 | 协议宏执行标志 |
| | 端口 2 | 协议宏执行标志 |
| | | IR 20708 |
| | | IR 20712 |

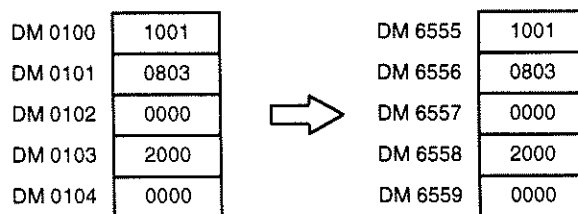
应用例

下例程序当 IR00000 为 ON 并且 IR20708 为 OFF 时, 把 DM0100 至 DM0104 中的内容传送到串行通信板端口 1 的 PC 设置区中。(DM6555 至 DM6559)



| 地址 | 指令 | 操作数 |
|-------|----------|---------|
| 00000 | LD | 00000 |
| 00001 | AND NOT | 20708 |
| 00002 | @STUP(-) | |
| | | 001 |
| | | DM 0100 |

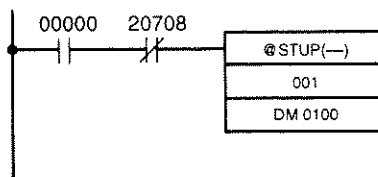
该设置的传送如下所示。当传送完成时,端口 1 的协议宏执行标志(IR20708)将变 OFF。



下表显示被传送的设置数据的功能。

| 字 | 内容 | 功能 |
|---------|------|--|
| DM 0100 | 1001 | 允许DM0101中的通信设置,并将通信模式设置成RS-232C。 |
| DM 0101 | 0803 | 按以下通信设置进行设置: 9, 600bps, 1起动位, 8位数据, 1位停止位, 无校验 |
| DM 0102 | 0000 | 无传输延迟(0 ms) |
| DM 0103 | 2000 | 允许结束代码CR, LF。 |
| DM 0104 | 0000 | ... |

注 如果 STUP(一) 执行时,某端口的设置改变标志或协议宏执行标志为 ON,将发生错误,因此将该标志的常闭条件作为执行条件。



使用 STUP(一)在工作时改变通信模式等设置。例如 在协议宏模式下可以执行通信序列以通过调制解调器连接来交换数据。当需要不停机而监视/编程 PC 时,通信模式可以转化成上位机链接模式。

标志

ER: 间接寻址的 EM/DM 字不存在。

(* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界)

端口指定(N)不是 IR000, IR001, IR002 或 IR003。

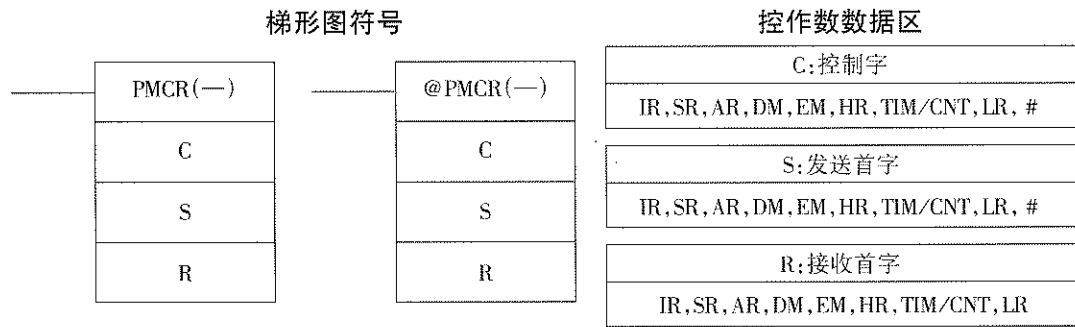
指定的源字超出数据区。

已指定了内置 RS-232C 端口或外围端口,但 DIP 开关的 5 脚却为 ON。

指定了串行通信板的端口,但却未安装串行通信板。

当指定端口的设置改变标志 (RS-232C 端口的 AR2404 或外围端口的 AR2403) 或协议宏执行标志 (端口 1 为 IR20708 或端口 2 为 IR20712) 为 ON 时,STUP(一)却被执行。

5-30-4 协议宏 PMCR(—)



限制

C 必须是 # 1000 ~ # 2999 之间的 BCD 值。

DM6144 ~ DM6655 不能用于 R。

说明

当执行条件为 OFF 时,PMCR(—)不执行,当执行条件为 ON 时,PMCR(—)调用并执行指定的通信序列(协议数据)该序已经登记到 PC 上安装的串行通信板上。

C 的位 00 ~ 11 指定通信序列号, C 的位 12 ~ 15 指定序列是从端口 1 还是端口 2 被执行。

当发送消息的变量中指定一个操作数时, S 中的内容(0001 ~ 0129 BCD)指定了包括 S 在内的发送区的字数。(发送数据从 S + 1 开始,所以实际发送字数是 0 ~ 128 字)。

当 S 和 R 未指定 DM 时, 串行通信板上注册的通信序列的发送/接收消息必须被设置成直接读或写字数据。如果无发送数据, S 中输入常数 # 0000, 输入任何其它常数或地址都会导致错误发生。

当通信序列不需要接收字时, 任意指定一个字地址, 数据不会贮存到指定的字中, 该字的内容将保持不变, 当通信序列需要接收字时, 指定程序中未使用的字。

发送和接收字(S 和 R)也可以在串行通信板上注册的通信序列中设置。

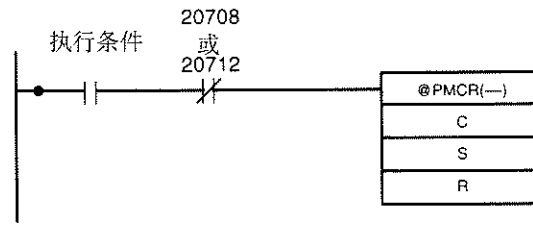
注 串行通信板的详细介绍参照串行通信板操作手册, 通信序列的详细介绍参考协议软件操作手册。

发送消息变量中的符号读选项(R())控制指定发送区中发送数据的传输。同样接收消息变量中的符号写选项(W())控制接收数据区中数据的接收。在消息中指定 R()和 W()选项的详细介绍参照 CX—Protocol 操作手册。

协议宏执行标志

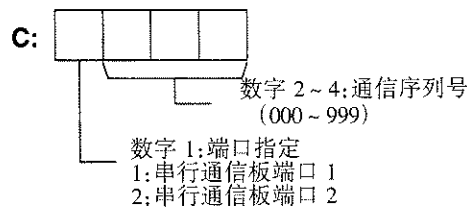
当 PMCR(—)执行时, 端口的协议宏执行标志(端口 1 的 IR20708 或端口 2 的 IR20712) 将变 ON。当通信序列执行完并且接收数据已贮存到指定的接收字中时, 该标志会变 OFF。

每个端口只能执行一个通信序列, 当端口的协议宏执行标志为 ON 时执行 PMCR(—), 会发生错误。因此把该标志的常闭条件作为一个执行条件, 以防止两条通信序列同时执行。



控制字(C)

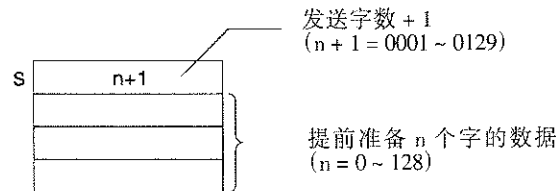
控制字的第一个数字 (1 或 2) 指定串行通信板的端口, 后 3 个数字指定通信序列 (000 至 999), 如下图所示:



发送首字(S)

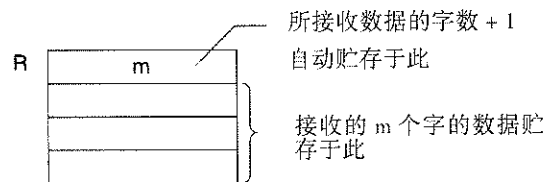
指定所需发送数据字的首字, S 包含发送的字数 + 1 (即包括 S 字), 并从 S + 1 开始发送数据。能发送 0 和 0128 之间的字数。

如果无发送数据, S 设置成常数 0000。如果设置任何其它常数或字地址, 将出现错误, 错误标志变 ON, PMCR(-) 不执行。



接收首字(R)。

该字包含接收的数据。无论有无数据被接收, 都要为 R 指定一个字地址。如果 R 设置为常数, 将发生错误, 错误标志将变 ON, 并且不执行 PMCR(-)。



标志

ER: 间接寻址 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

R 不是 BCD 或者 DM6144 ~ DM6655 用于 R。

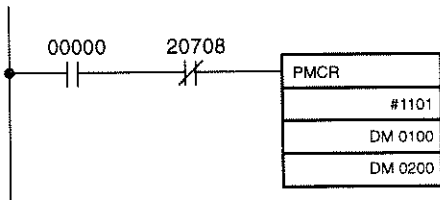
当 PMCR(-) 指令执行时, 另一 MPCR(-) 指令已经在执行; 同时协议宏执行标志已为 ON。

端口指定非 1 或 2。

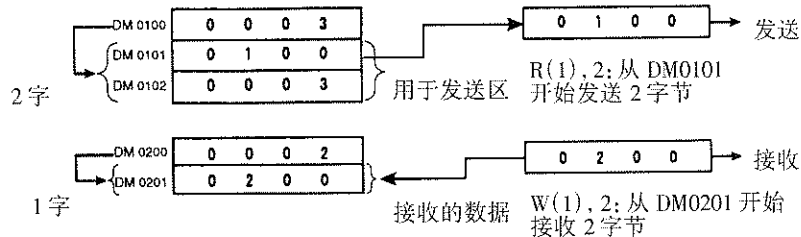
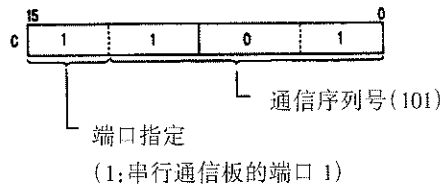
例

当 IR00000 为 ON 且 SR20708(端口 1 协议宏执行标志) 为 OFF 时, PMCR(—) 执行通信序列 101, DM0100 包含 0003, 所以后面 2 个字 (DM0101 和 DM0102) 用于发送数据。接收数据从 DM0201 开始贮存, 接收的字数自动写入 DM0200 (第一个接收字)。

注 发送消息中的符号读选项 R() 或符号写选项 W() 实际发送/接收数据。



| 地址 | 指令 | 操作数 |
|-------|---------|---------|
| 00200 | LD | 00000 |
| 00201 | AND NOT | 20708 |
| 00202 | PMCR(—) | |
| | | # 1101 |
| | | DM 0100 |
| | | DM 0200 |



5—31 高级 I/O 指令

5—31—1 7 段显示输出—7SEG(88)

梯形图符号

| |
|----------|
| 7SEG(88) |
| S |
| O |
| C |

操作数数据区

| |
|-------------------------------------|
| S: 首源字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| O: 输出字 |
| IR, SR, AR, HR, LR, TIM/CNT, DM, EM |
| C: 控制字 |
| 000 ~ 007 |

限制
说明

程序中使用 7SEG(88) 不能多于 2 次。
当执行条件为 OFF 时, 7SEG(88) 不执行。当执行条件为 ON 时, 7SEG(88) 读源数据 (4 或 8 位数), 将其转换为 7 段显示数据, 并输出数据到与 O 指示的输出相连接的 7 段显示。

C 的值表示源数据的数字位数和输入输出单元的逻辑,如下表所示。

| 源数据 | 显示的数据输入逻辑 | 显示的锁存输入逻辑 | C |
|------------------|-----------|-----------|------|
| 4 位数字(S) | 同输出单元 | 同输出单元 | 0000 |
| | | 不同于输出单元 | 0001 |
| | 不同于输出单元 | 同输出单元 | 0002 |
| | | 不同于输出单元 | 0003 |
| 8 位数字 (S,S+1) | 同输出单元 | 同输出单元 | 0004 |
| | | 不同于输出单元 | 0005 |
| | 不同于输出单元 | 同输出单元 | 0006 |
| | | 不同于输出单元 | 0007 |

若有 8 位数的源数据,它们被置于 S 和 S + 1,高位数字置于 S + 1。若源数据是 4 位数字,它们被置于 S。

7SEG(88) 在 12 个循环内显示 4 或 8 位数字,然后重新开始并继续显示数据。

关于 7SEG(88)和它的应用详见 412 页。

标志

ER: S 和 S + 1 不在同一数据区。(当设置显示 8 位数字的数据时。)

间接寻址 EM/DM 字不存在。

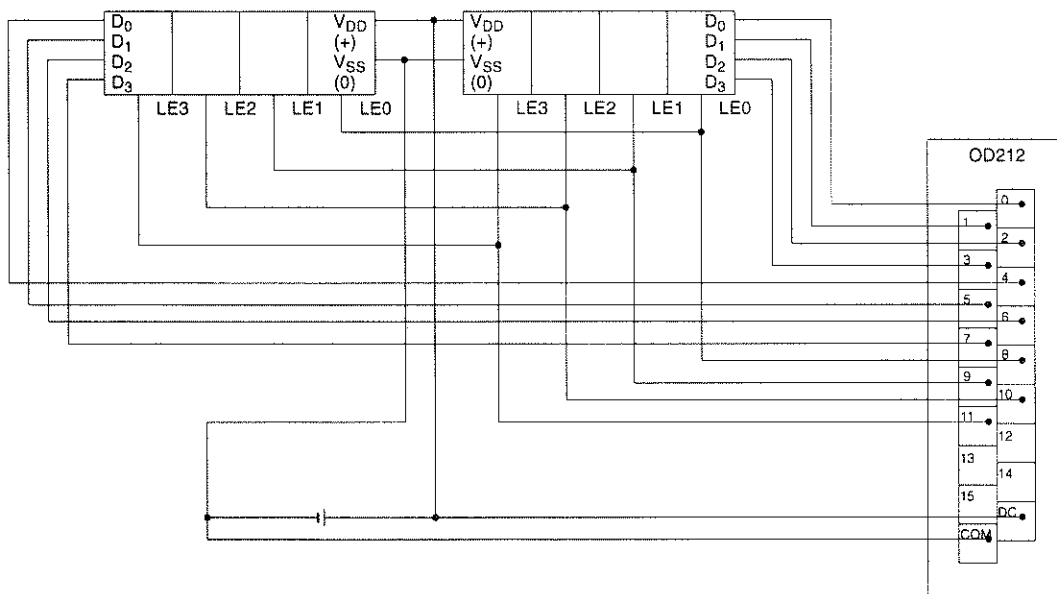
(* EM/* DM 字的内容不是 BCD,或超出 EM/DM 区边界。)

操作数设置有错误。

硬件

SR25409:当 7SEG(88)执行时为 ON。

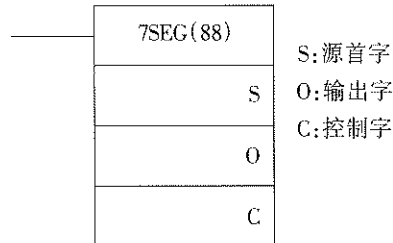
如下图显示,7 段显示连接到一个输出单元。4 位数显示时,数据输出(D0 到 D3)连接到输出点 0 至 3,锁存输出(CS0 到 SC3)连接到输出点 4 至 7。当一个轮回的数据被显示时,输出点 12(8 位数显示)或输出点 8(4 位数显示)将变 ON,但不必连接它们,除非应用需要。



输出可连接的晶体管输出单元为：对 4 位数，8 或以上输出点；对 8 位数，16 或以上输出点。

- 注
- 1、输出单元经常采取负逻辑(仅 PND 输出单元采用正逻辑)。
 - 2、7 段显示可能需要正或负逻辑,依据型号而定。

使用指令



如果在 S 中指定了保存显示数据的首字,在 O 中指定了输出字,在 C 中指定了表中获取的 SV,那么程序执行将按下列所示进行处理。

数据贮存格式

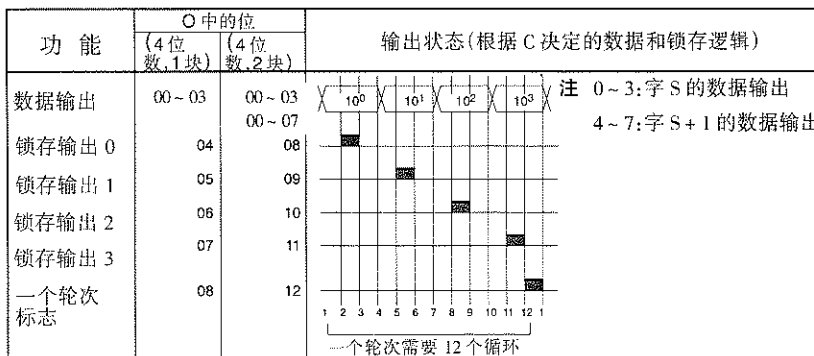


如果只显示 4 位数,则只使用字 S。

选择逻辑和数字数量的设置值(C)

| 显示数字数量 | 显示单元数据输入和输出单元逻辑 | 显示单元锁存输入和输出单元逻辑 | C 设置数据 |
|----------------|-----------------|-----------------|--------|
| 4 位数(4 位数,1 块) | 相同 | 相同 | 000 |
| | | 不同 | 001 |
| | 不同 | 相同 | 002 |
| | | 不同 | 003 |
| 8 位数(4 位数,2 块) | 相同 | 相同 | 004 |
| | | 不同 | 005 |
| | 不同 | 相同 | 006 |
| | | 不同 | 007 |

注 C 不要设置成 000~007 以外的值

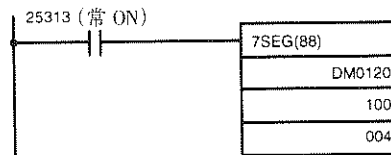


当执行 7SEG(88)时,SR25409 将变 ON。

- 注
- 1、在同一程序中不要多次使用 7SEG(88)。
 - 2、设计系统时考虑循环时间和 7 段显示的特性。
 - 3、未使用到的输出位可用于一般输出位。
- 使用该指令时, 4 位数或 8 位数在 12 个循环内被显示。
操作从首次执行开始, 不考虑以前执行的状态。

应用例

本例示出了在 7 段 LED 上显示 CQM1 的 8 位数 BCD 码的程序。假设 7 段显示连接在输出字 IR100 上, 并且输出单元使用负逻辑, 7 段显示逻辑的数据信号和锁存信号也为负逻辑。



使用 7SEG(88) 一直显示 DM0120(右边 4 位数) 和 DM0121(左边 4 位数) 的 8 位数 BCD 数据。当 DM0120 和 DM0121 中的内容改变时, 显示也随之改变。

5—31—2 数字开关输入—DSW(87)

梯形图符号

| |
|---------|
| DSW(87) |
| IW |
| OW |
| R |

操作数数据区

| |
|-------------------------------------|
| IW: 输入字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| OW: 输出字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| R: 结果首字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |

限制说明

DM6144 ~ DM6655 不能用于 R。
DSW(87)用来读出在与 I/O 单元连接的数字开关上的设定值, 当执行条件为 OFF 时, DSW(87)不执行, 当执行条件为 ON 时, DSW(87)从 IW 上读出在数字开关上设定的值并将结果置于 R。
若数值为 8 位数字, 把它置于 R 和 R + 1, 高位数字置于 R + 1。数字位数在 PC 设置的 DM6639 中设定。

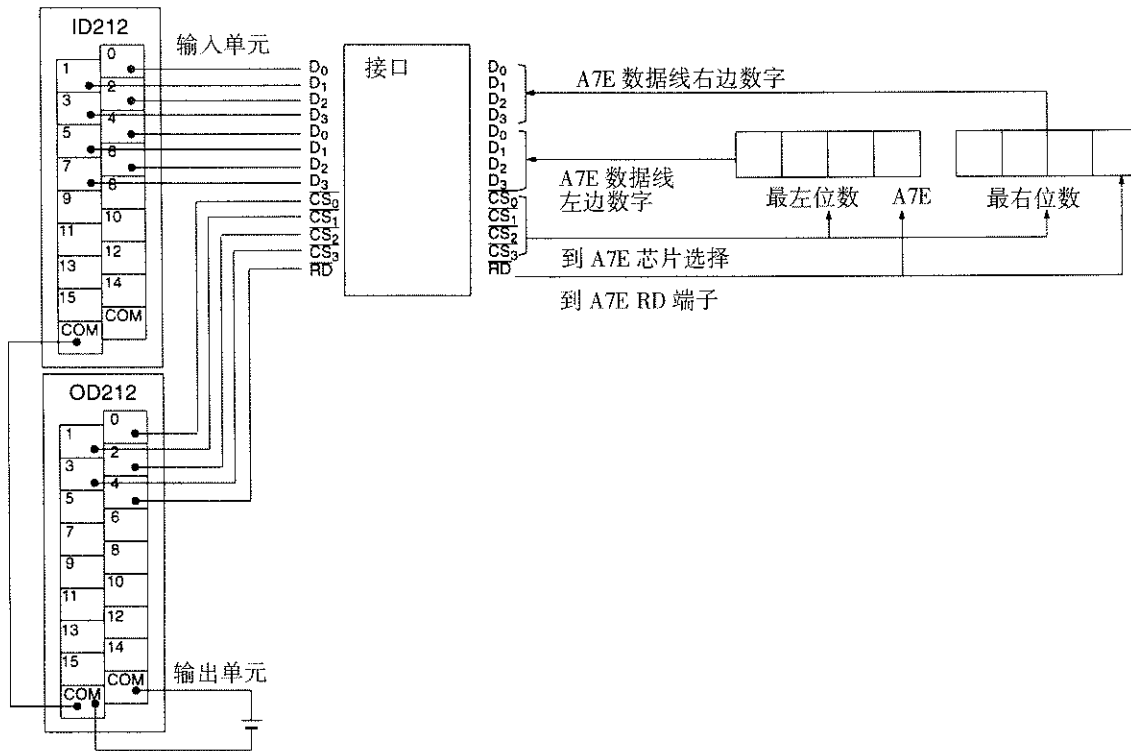
DSW(87)在 12 个循环中读 4 或 8 位数字值, 然后重新开始继续读数据。
关于 DSW(87)及其应用详见 415 页。

标志

ER: IW 和/或 OW 没有分配正确的 I/O 单元。
间接寻址 EM/DM 字不存在。
(* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)
R 和 R + 1 不在同一数据区。(当 CQM1H 设置为接收 8 位数字值)
SR25410: 当 DSW(87)执行时为 ON。

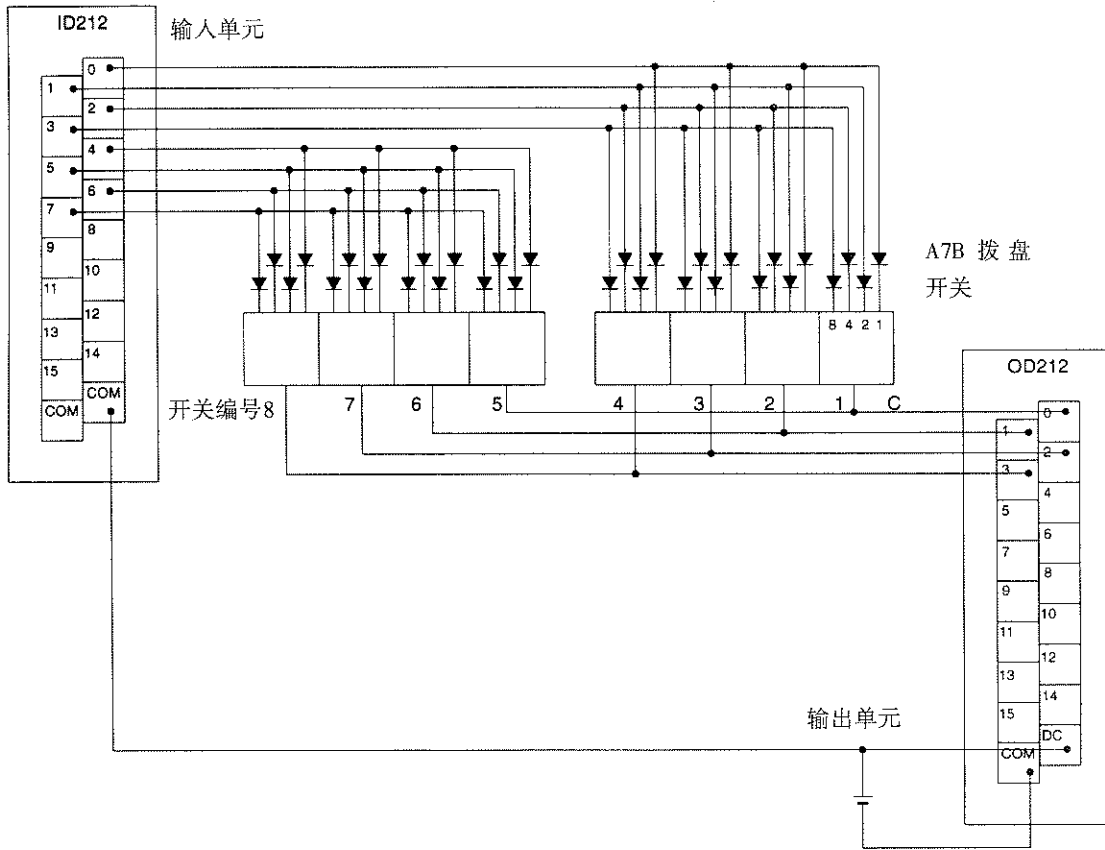
硬件

按下图所示连接数字开关和输入输出单元。图形中显示了一个 8 位数输入。当使用 4 位数输入时, 数字开关 D0 ~ D3 连接到输入点 0 ~ 3 上。不论哪种情况, 当一个轮次的数据被读取时, 输出点 5 将变 ON, 但却不需要连接输出点 5, 除非应用需要。



注 连接 A7E 数字开关时需要一个接口,用于把 5V 转化成 24V。

下例显示了一个 A7B 拨盘开关的连接

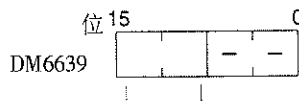


注 该例中不需要数据读信号。

输入可以连接到 CPU 单元的输入端或 8 点以上的 DC 输入单元。输出可以连接到 8 点或以上的晶体管输出单元。

注意

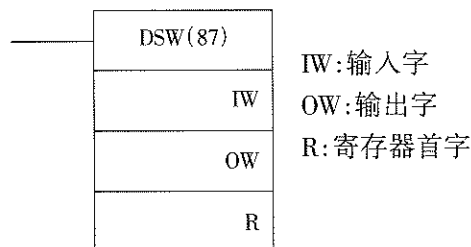
当使用 DSW(87)时,在执行程序前,在编程模式中对 PC 设置进行以下设置。
数字开关设置(PC 设置)



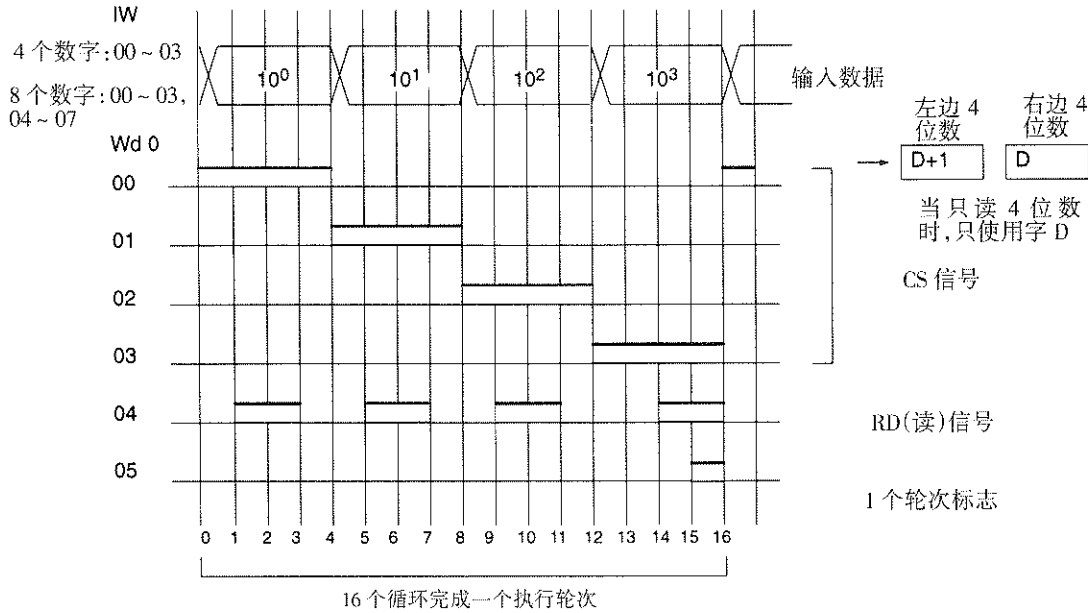
要读的位数
00:4 位数
01:8 位数
缺省:4 位数

不要改变位 0~7,它们与 DSW(87)无关。

使用指令



如果连接数字开关的输入字在 IW 中指定,输出字在 OW 中指定,那么程序执行过程如下。



当 DSW(87) 执行时 IR25410 将变 ON。

- 注**
- 1、同一程序中不要多次使用 DSW(87)。
 - 2、使用 DSW(87) 时,相关输入字的输入常数的设置要小于循环时间(输入常数可以预先在 DM6620 中修改),在程序和系统设计中也要考虑数字开关的特性。
 - 3、未使用的输入和输出位可以用做一般的输入和输出位。

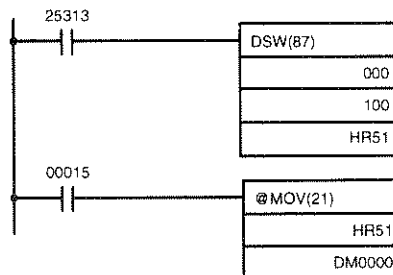
使用该指令,4 位数或 8 位数设定值可以在 16 个循环中被读取。

应用例

该例显示了一个从数字开关中读取 4 位数 BCD 值的程序。假定数字开关连接到 IR000(输入)和 IR100(输出)中,并假定 PC 设置全部为缺省状态(读 4 位数)。

从数据开关设定的数据通过 DSW(87) 存贮在 HR51 中。

当 IR00015 变 ON 时,存贮在 HR51 中的数值转移到 DM0001 中。

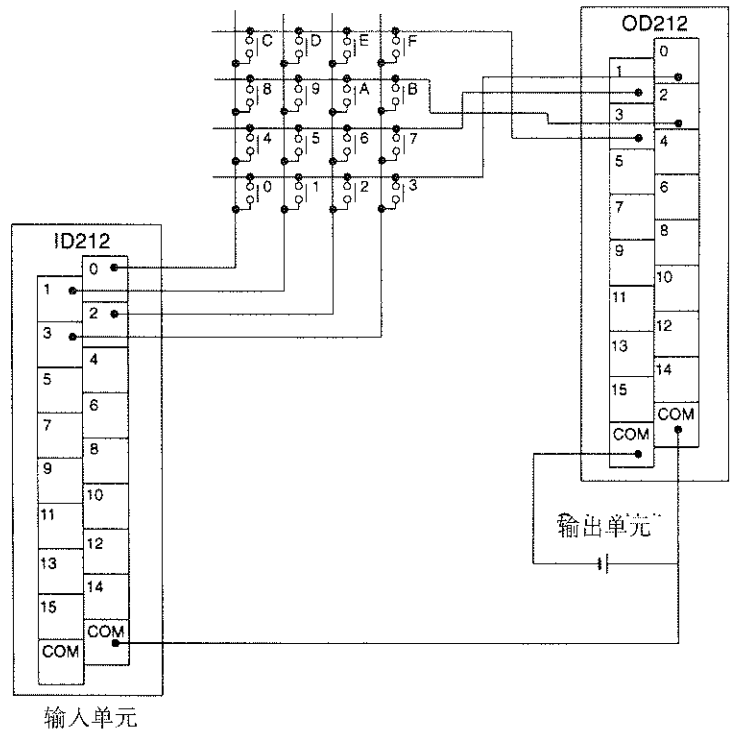


- 注**
- 当读取一个轮次的数据时,输出点 5(此处为 IR10005)变 ON,当 DSW(87)用于将数据输到不同的存贮区时,该输出点可以用于定时切换数据存贮区和门信号(CS 信号)。

5-31-3 16 键输入—HKY(—)

| 梯形图符号 | 操作数数据区 |
|--------|---|
| HKY(—) | IW:输入字 IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| IW | OW:控制信号输出字 IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| OW | D:寄存首字 IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| D | |

| | |
|----|--|
| 限制 | D 和 D + 2 必须在同一数据区。 程序中不能使用 HKY(—)两次以上。 DM6144 ~ DM6655 不能用于 D。 |
| 说明 | 当执行条件为 OFF 时, HKY(—)不执行。当执行条件为 ON 时, HKY(—)从与 IW 指示的输入相连的 16 进制键盘输入数据, 数据输入有两种方法: 1, 2, 3... 1. 在 D 和 D + 1 生成一个 8 位数字移位寄存器, 当在 16 进制键盘中按下一个键, 对应的 16 进制值被移进 D 的最低位数字。D 和 D + 1 的其它数字则左移, D + 1 的最高位数字丢失。 2. D + 2 的位和 OW 的第 4 位表示键输入。当键盘中一个键(0 ~ F)被按下, 在 D + 2 中的对应位(00 至 15)和 OW 的第 4 位被置 ON。 注 当键盘中一个键被按下, 从其它的键输入是不可能的。 HKY(—)在 3 至 12 个循环中输入各个数字, 然后开始重复并继续输入, 详见 419 页的 HKY(—)。 |
| 标志 | ER:间接寻址 EM/DM 字不存在。 (* EM/* DM 字的内容不是 BCD, 或超出 EM/DM 区边界。) D 和 D + 2 不在同一数据区。 |
| 硬件 | SR25408:当 HKY(—)执行时置 ON。 准备十六进制键盘, 按下图所示将 0 ~ F 数字键开关连接到输入点 0 ~ 3 和输出点 0 ~ 3。当按下任意键时, 输出点 4 将变 ON, 但这里却不必连接该点。 |



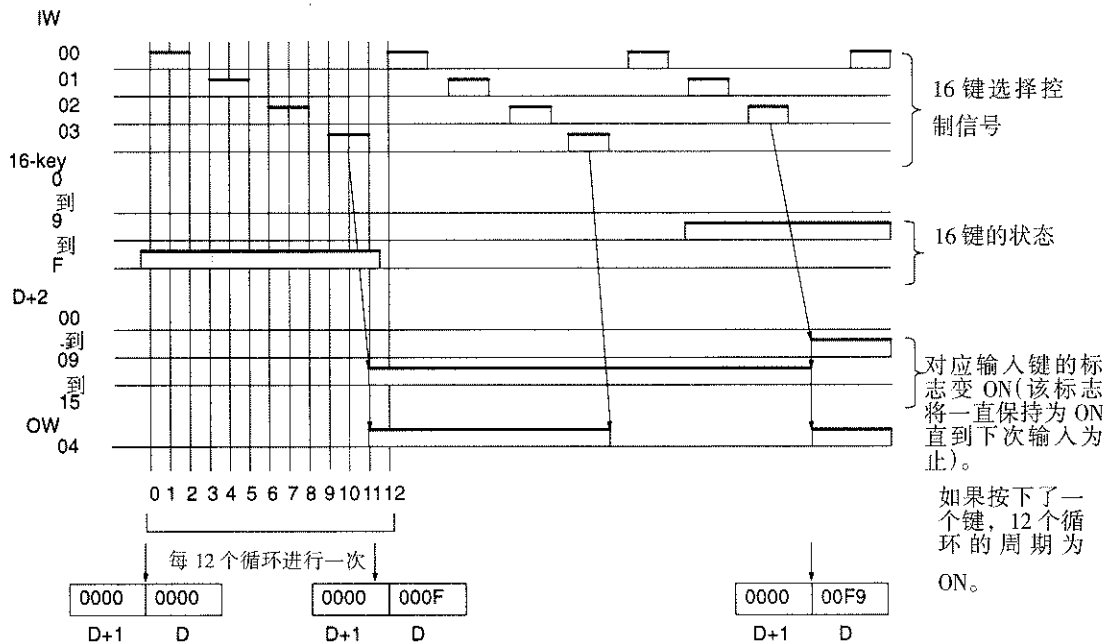
输入可以连接到 CPU 单元输入或 8 点及 8 点以上的 DC 输入单元的端子上，
输出可连接到 8 点或 8 点以上的晶体管输出单元上。

使用指令

| | |
|--|-----|
| | HKY |
| | IW |
| | OW |
| | D |

IW: 输入字
OW: 控制信号输出字
D: 寄存器首字

如果在 IW 中指定了连接十六进制键盘的输入字,在 OW 中指定了输出字,程序运行过程将如下图所示。



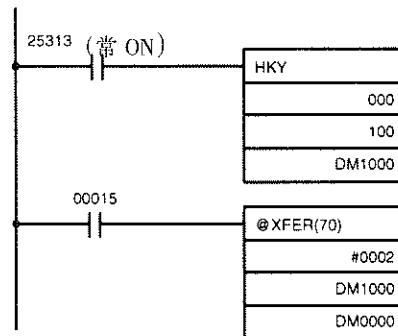
当执行 HKY(—)时 SR25408 将变 ON。

- 注
1. 同一程序中不要多次使用 HKY(—)。
 2. 使用 HKY(—)时,相关输入字的输入常数的设定要小于循环时间(输入常数可以在 DM6620 中预先修改)。
 3. 当按下某一键时,将不接收来自其它键的输入。
 4. 如果输入超出 8 个数字,将从最左边数开始删除数字。
 5. 不使用的输入和输出位可以用于一般的输入和输出位。

使用该指令,一个键输入在 3 至 12 个循环中被读取。需要多个循环是因为 ON 键只能在输出变 ON 时才能确定。

该例显示了一个从十六进制键盘输入数字的程序,假定十六进制键盘连接到 IR000(输入)和 IR100(输出)中。

应用例



HKY(一) 输入到 IR000 中的十六进制键信息被转化成十六进制数中并存贮到 DM1000 和 DM1001 字中。

IR00015 用做“回车键”，当 IR00015 变 ON，贮存在 DM1000 和 DM1001 中的数字传送到 DM0000 和 DM0001 中。

5—31—4 10 键输入—TKY(18)

梯形图符号

| |
|----------------|
| YKY(18) |
| IW |
| D ₁ |
| D ₂ |

操作数数据区

| |
|-------------------------------------|
| IW:输入字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| D ₁ :首寄存字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |
| D ₂ :键输入字 |
| IR, SR, AR, DM, EM, HR, TIM/CNT, LR |

限制

D₁ 和 D₁ + 1 必须在同一数据区。

DM6143 ~ DM6655 不能用于 D₁。

说明

当执行条件为 OFF 时, TKY(18) 不执行, 当执行条件为 ON 时, TKY(18) 从与 IW 指出的输入连接的十键键盘输入数据。数据输入有两种方法:

- 1, 2, 3... 1. D₁ 和 D₁ + 1 生成一个 8 位数字移位寄存器。当十键键盘的一个键被按下, 对应的 BCD 数字被移进 D₁ 的最低位数。D₁, D₁ + 1 的其它位都左移。D₁ + 1 的最高位数丢失。
2. D₂ 的头 10 位表示键输入。当键盘的一个键(0 至 9)被按下, D₂ 的对应位(00 至 09)置 ON。

注 当键盘中的一个键被按下, 不能从其它键输入。

TKY(18) 可依靠改变输入字 IW 在程序的多个位置使用, 详见 422 页的 TKY(18)。

标志

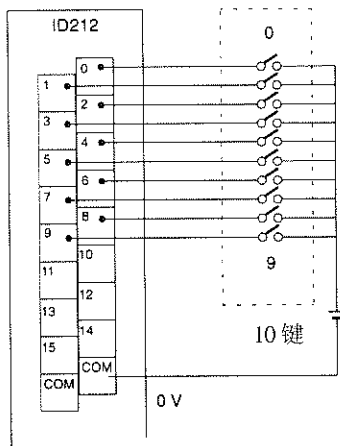
ER: 间接寻址的 EM/DM 字不存在。

(*EM/*DM 字的内容不是 BCD, 或超出 EM/DM 区边界。)

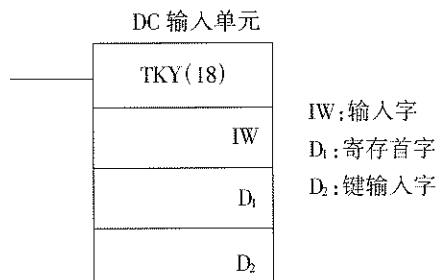
D₁ 和 D₁ + 1 不在同一数据区。

硬件

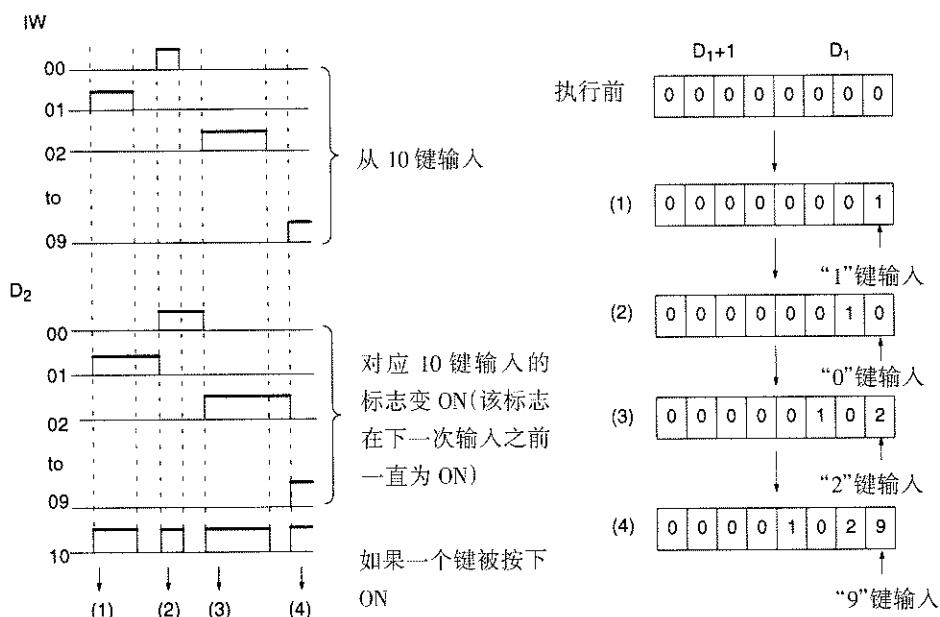
准备一个 10 键键盘, 连接该键盘使得数字键 0 至 9 被输入到点 0 至 9 中, 如下图所示。可以使用 CPU 单元的输入端或 16 位及以上输入点的 DC 输入单元。



使用指令



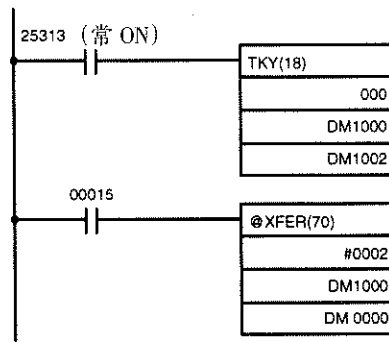
如果连接 10 键键盘的输入字已在 IW 中指定, 则当程序执行时操作将如下进行。



- 注
1. 当按下一个键时, 来自其它键的输入将不被接收。
 2. 如果输入多于 8 位数, 从最左位数开始删除数字。
 3. 未使用的输入位可用于一般的输入位。

应用例

该例显示了用 10 键输入数字的程序。假设 10 键与 IR000 相连。



使用 TKY(18) 将 10 键的信息输入到 IR000，转换为 BCD 格式并贮存到 DM1000 和 DM1001 中。键信息贮存到 DM1002 中。

IR00015 用作为“ENTER 键”，当 IR00015 变 ON，贮存在 DM1000 和 DM1001 中的数据将传送到 DM0000 和 DM0001 中。

第 6 章 上位机链接命令

本章介绍上位机链接命令的使用方法与步骤,这些命令可用于 COM1H 端口的上位机链接通信。

| | | |
|--------|--------------|-----|
| 6-1 | 上位机链接命令概述 | 426 |
| 6-2 | 结束代码 | 427 |
| 6-2-1 | 代码 | 427 |
| 6-2-2 | 代码及可使用的命令 | 429 |
| 6-3 | 通信步骤 | 429 |
| 6-4 | 命令和响应格式 | 431 |
| 6-4-1 | 上位计算机的命令 | 431 |
| 6-4-2 | PC 的命令 | 434 |
| 6-5 | 上位机链接命令 | 435 |
| 6-5-1 | 读 IR/SR 区—RR | 435 |
| 6-5-2 | 读 LR 区—RL | 435 |
| 6-5-3 | 读 HR 区—RH | 436 |
| 6-5-4 | 读 PV—RC | 436 |
| 6-5-5 | 读 TC 状态—RG | 436 |
| 6-5-6 | 读 DM 区—RD | 437 |
| 6-5-7 | 读 EM 区—RE | 437 |
| 6-5-8 | 读 AR 区—RJ | 438 |
| 6-5-9 | 写 IR/SR 区—WR | 438 |
| 6-5-10 | 写 LR 区—WL | 439 |
| 6-5-11 | 写 HR 区—WH | 439 |
| 6-5-12 | 写 PV 区—WC | 440 |
| 6-5-13 | 写 TC 状态—WG | 440 |
| 6-5-14 | 写 DM 区—WD | 441 |
| 6-5-15 | 写 EM 区—WE | 442 |
| 6-5-16 | 写 AR 区—WJ | 442 |
| 6-5-17 | SV 读取 1—R # | 443 |
| 6-5-18 | SV 读取 2—R \$ | 444 |
| 6-5-19 | SV 读取 3—R % | 445 |
| 6-5-20 | SV 改变 1—W # | 446 |
| 6-5-21 | SV 改变 2—W \$ | 446 |
| 6-5-22 | SV 改变 3—W % | 447 |
| 6-5-23 | 读状态—MS | 448 |
| 6-5-24 | 写状态—SC | 449 |
| 6-5-25 | 读错误—MF | 450 |
| 6-5-26 | 强制置位—KS | 451 |
| 6-5-27 | 强制复位—KR | 452 |
| 6-5-28 | 多点强制置位/复位—FK | 453 |
| 6-5-29 | 取消强制置位/复位—KC | 454 |
| 6-5-30 | 读 PC 型号—MM | 455 |
| 6-5-31 | 测试—TS | 455 |
| 6-5-32 | 读程序—RP | 456 |
| 6-5-33 | 写程序—WP | 456 |
| 6-5-34 | 复合命令—QQ | 456 |
| 6-5-35 | 终止—XZ | 458 |
| 6-5-36 | 初始化—** | 458 |
| 6-5-37 | TXD 响应—EX | 459 |
| 6-5-38 | 未定义的命令—IC | 459 |

6-1 上位机链接命令概述

下表所列上位机链接命令可用上位机链接通信发送到 CQM1H。

| 标头代码 | PC 模式 | | | 名称 | 页 |
|------|-------|-----|-----|-------------|-----|
| | 运行 | 监视 | 编程 | | |
| RR | 有效 | 有效 | 有效 | 读 IR/SR 区 | 435 |
| RL | 有效 | 有效 | 有效 | 读 LR 区 | 435 |
| RH | 有效 | 有效 | 有效 | 读 HR 区 | 436 |
| RC | 有效 | 有效 | 有效 | 读 PV | 436 |
| RG | 有效 | 有效 | 有效 | 读 TC 状态 | 436 |
| RD | 有效 | 有效 | 有效 | 读 DM 区 | 437 |
| RE | 有效 | 有效 | 有效 | 读 EM 区 | 437 |
| RJ | 有效 | 有效 | 有效 | 读 AR 区 | 438 |
| WR | 无效 | 有效 | 有效 | 写 IR/SR 区 | 438 |
| WL | 无效 | 有效 | 有效 | 写 LR 区 | 439 |
| WH | 无效 | 有效 | 有效 | 写 HR 区 | 439 |
| WC | 无效 | 有效 | 有效 | 写 PV | 440 |
| WG | 无效 | 有效 | 有效 | 写 TC 状态 | 440 |
| WD | 无效 | 有效 | 有效 | 写 DM 区 | 441 |
| WE | 无效 | 有效 | 有效 | 写 EM 区 | 442 |
| WJ | 无效 | 有效 | 有效 | 写 AR 区 | 442 |
| R # | 有效 | 有效 | 有效 | SV 读取 1 | 443 |
| R \$ | 有效 | 有效 | 有效 | SV 读取 2 | 444 |
| R % | 有效 | 有效 | 有效 | SV 读取 3 | 445 |
| W # | 无效 | 有效 | 有效 | SV 改变 1 | 446 |
| W \$ | 无效 | 有效 | 有效 | SV 改变 2 | 446 |
| W % | 无效 | 有效 | 有效 | SV 改变 3 | 447 |
| MS | 有效 | 有效 | 有效 | 读状态 | 448 |
| SC | 有效 | 有效 | 有效 | 写状态 | 449 |
| MF | 有效 | 有效 | 有效 | 读错误 | 450 |
| KS | 无效 | 有效 | 有效 | 强制置位 | 451 |
| KR | 无效 | 有效 | 有效 | 强制复位 | 452 |
| FK | 无效 | 有效 | 有效 | 多点强制置位/复位 | 453 |
| KC | 无效 | 有效 | 有效 | 强制置位/复位取消 | 454 |
| MM | 有效 | 有效 | 有效 | 读 PC 型号 | 455 |
| TS | 有效 | 有效 | 有效 | 测试 | 455 |
| RP | 有效 | 有效 | 有效 | 读程序 | 456 |
| WP | 无效 | 无效 | 有效 | 写程序 | 456 |
| QQ | 有效 | 有效 | 有效 | 复合命令 | 456 |
| XZ | 有效 | 有效 | 有效 | 终止(仅命令) | 458 |
| * * | 有效 | 有效 | 有效 | 初始化(仅命令) | 458 |
| EX | 有效 | 有效 | 无效 | TXD 响应(仅响应) | 459 |
| IC | ... | ... | ... | 未定义命令(仅响应) | 459 |

6—2 结束代码

6—2—1 代码

下表所列响应(结束)代码是对上位机链接命令的响应帧返回。当发生 2 个及以下的错误时,先返回第一个错误的结束代码。

| 结束码 | 内容 | 可能的原因 | 修正措施 |
|-----|----------------------|--|-------------------------------|
| 00 | 正常结束 | 不存在问题。 | …… |
| 01 | 运行模式下不能执行 | 当 PC 为运行模式时发送命令不可执行。 | 检查命令和 PC 模式的关系。 |
| 02 | 监视模式下不能执行 | 当 PC 为监视模式时发送命令不可执行。 | |
| 03 | UM 写保护 | PC 的 UM 为写保护。 | CPU 单元 DIP 开关(SW1)的 1 脚置 OFF。 |
| 04 | 地址溢出 | 读或写命令中设置的程序地址高于最大程序地址。 | 检查程序。 |
| 13 | FCS 错误 | FCS 错误。 | 检查 FCS 计算方法,如果有噪声影响,命令再传送一次。 |
| 14 | 格式错误 | 命令格式错误或不能划分的命令被划分,或帧长度小于应用命令的最小长度。 | 检查格式并再次传送命令。 |
| 15 | 入口号数据错误 | 数据超出指定范围或太长。未指定十六进制的数据。 | 修正数据并再次传送命令。 |
| 16 | 命令不被支持 | 在程序中不存在 SV 读或 SV 改变命令中指定的操作数。 | 检查查找数据或查找起始点。 |
| 18 | 帧长度错误 | 超出最大帧长度 132 字节。如果帧超出 280 字节,接收溢出标志会变 ON,并且无响应。 | 检查命令,如有必要,将其分隔成多个帧。 |
| 19 | 不可执行 | 读 SV 超出 9,999,或者当要读的项目未登记时却执行了 L/O 内存批读命令。 | 在试图批读取之前登记要读的项目。 |
| 23 | 用户内存被保护 | UM 被写保护。 | 取消写保护。 |
| A3 | 由于传输数据中的 FCS 错误而异常终止 | 第二个或以后的帧发生了 FCS 错误,或在多重写时,中间或最后帧中存在 2 个字节或更少的数据。 | 终止命令数据并再次传送命令。 |
| A4 | 因传输数据的格式错误而异常终止 | 命令格式与第二个或以后的帧的字节数不匹配。 | |
| A5 | 因传输数据的入口号数据错误而异常终止 | 第二个或以后的帧中有入口号数据错误,数据长度错误或数据未设置为十六进制。 | |
| A8 | 因传输数据的帧长度错误而异常终止 | 第二个及以后的帧的长度超出 128 字节的最大值。 | |

对某些错误,不管命令是什么,响应将收不到。这些错误列表如下。

| 错 误 | PC 操作 |
|-------------------------------------|---|
| 接收命令时校验,过运行或帧错误。 (甚至对其它单元的命令地址)。 | 通信错误标志将 ON, 错误代码将被记录, 接收将被复位。 (通信正常重启时该错误自动被消除。) 通信错误标志如下: 外围端口: AR0812 内置 RS-232C 端口: AR0804 串行通信板端口 1: IR20104 串行通信板端口 2: IR20104 |
| 接收命令的首帧起始字符不为@。 | 命令将被忽略。 |
| 不正确的节点号(不是本地单元或超出 30)。 | 命令将被忽略。 |
| 多帧写时中间或最后帧中的数据超出 2 字节。 | 将发生 FCS 错误。 |

6—2—2 代码和可用的命令

下表列出各命令可返回的结束码。

| 标头代码 | 可能的结束码 | | | | | | | | 说明 | |
|------|--------|----|----|----|----|----|----------|----|-------------|-----|
| RR | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RL | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RH | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RC | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RG | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RD | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RE | 00 | | 13 | 14 | 15 | 18 | | A3 | A8 | ... |
| RJ | 00 | | 13 | 14 | 15 | 18 | | | | ... |
| WR | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WL | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WH | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WC | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WG | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WD | 00 | 01 | | 13 | 14 | 15 | 18 | 23 | A3 A4 A5 A8 | ... |
| WE | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| WJ | 00 | 01 | | 13 | 14 | 15 | 18 | | A3 A4 A5 A8 | ... |
| R# | 00 | | | 13 | 14 | 15 | 16 18 | 23 | | ... |
| R\$ | 00 | | 04 | 13 | 14 | 15 | 16 18 | 23 | | ... |
| R% | 00 | | 04 | 13 | 14 | 15 | 16 18 | 23 | | ... |
| W# | 00 | 01 | | 13 | 14 | 15 | 16 18 | 23 | | ... |
| W\$ | 00 | 01 | 04 | 13 | 14 | 15 | 16 18 | 23 | | ... |
| W% | 00 | 01 | 04 | 13 | 14 | 15 | 16 18 | 23 | | ... |
| MS | 00 | | | 13 | 14 | | 18 | | | ... |
| SC | 00 | | | 13 | 14 | 15 | 18 19 | | | ... |
| MF | 00 | | | 13 | 14 | 15 | 18 | | | ... |
| KS | 00 | 01 | | 13 | 14 | 15 | 18 | | | ... |
| KR | 00 | 01 | | 13 | 14 | 15 | 18 | | | ... |
| FK | 00 | 01 | | 13 | 14 | 15 | 18 | | | ... |
| KC | 00 | 01 | | 13 | 14 | | 18 | | | ... |
| MM | 00 | | | 13 | 14 | | 18 | | | ... |
| TS | | | | 13 | 14 | | 18 | | | ... |
| RP | 00 | | | 13 | 14 | | 18 | 23 | A3 A8 | ... |
| WP | 00 | 01 | 02 | | 13 | 14 | 15 18 19 | 23 | A3 A4 A5 A8 | ... |
| QQ | 00 | | | 13 | 14 | 15 | 18 19 | | A3 A4 A5 A8 | ... |
| XZ | | | | | | | ... | | | 无响应 |
| * * | | | | | | | ... | | | 无响应 |
| IC | | | | | | | ... | | | 无响应 |
| EX | | | | | | | ... | | | 无响应 |

6—3 通信步骤

依靠上位计算机和 PC 之间交换命令和响应实施上位机链接通信。对 CQM1H, 有两种通信方法可以使用。一种是通常使用的方法, 从上位机发出

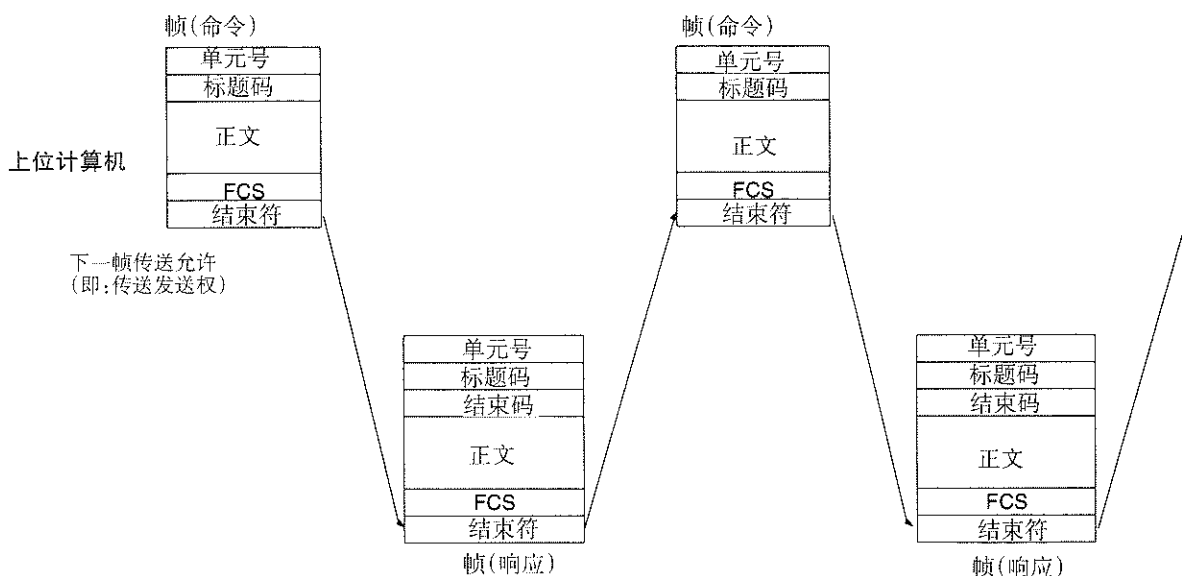
帧发送与接收

命令给 PC,另一种方法允许从 PC 发出命令给上位计算机。
命令和响应如下图所示顺序进行交换。在一个发送中传送的数据块称为“帧”。一个帧最多由 131 个字符的数据组成。

发送一个帧的权称为“发送权”。有发送权的单元是在任意给定时间发出一帧的单元,每次发送完一帧,发送权就在上位计算机和 PC 之间轮换。当收到结束符(标志一个命令或响应结束的代码)或分界符(分开帧的代码)时发送权从发送单元传给接收单元。

上位计算机的命令

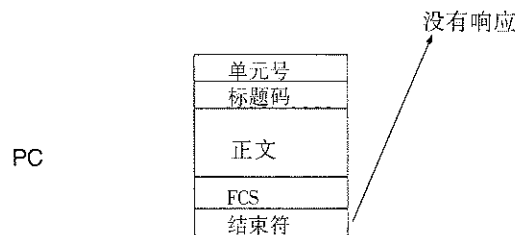
在上位机链接通信中,上位计算机一般具有先传送权并启动通信,PC 然后自动发出一个响应。



PC 的命令

对于 COMIH PC 来说,在上位机链接通信中 PC 发送命令给上位计算机也是可能的。此时 PC 具有传送权并启动通信。

上位计算机



当命令发至上位计算机,数据从 PC 单向传送至上位计算机。如果一个命令需要相应的响应,使用上位机链接通信命令将上位计算机的响应写到 PC。

6—4 命令和响应格式

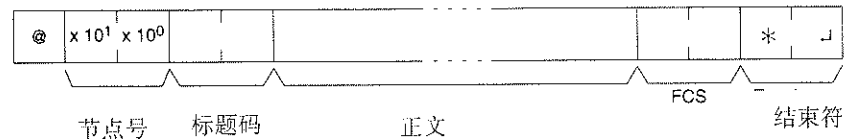
本节解释了在上位机链接通信中交换的命令和响应格式。

6—4—1 上位计算机的命令

当从上位计算机发出一个命令时,命令和响应格式如下所示。

命令格式

当从上位计算机发送一个命令时,用如下所示格式准备命令数据。



@

开始位必须放“@”符号。

节点号

标识和上位计算机通信的 PC。

在“PC 设置”中设定 PC 的节点号 (DM6648 和 DM6653 用于 CPU 单元, DM6553 和 DM6558 用于串行通信板)。

标题码

设置 2 字符的命令代码。

正文

设置命令参数。

FCS

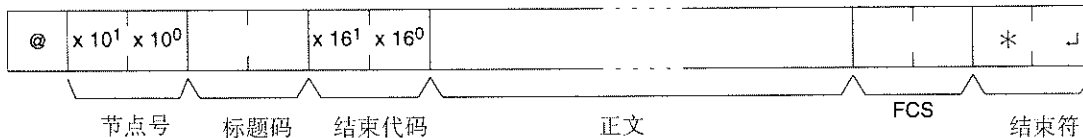
设置 2 字符的帧检查顺序代码。见 433 页。

结束符

设置 2 个字符,“*”和回车 (CHR \$(13)) 以表示命令的结束。

响应格式

从 PC 返回的响应格式如下所示。要准备一个程序以解释和处理响应数据。



@, 节点号, 标题码

返回内容与命令中的相同。

结束代码

返回命令的完成状态 (例如: 是否发生错误)。

正文

当有数据时如读数据才返回正文。

FCS, 结束符

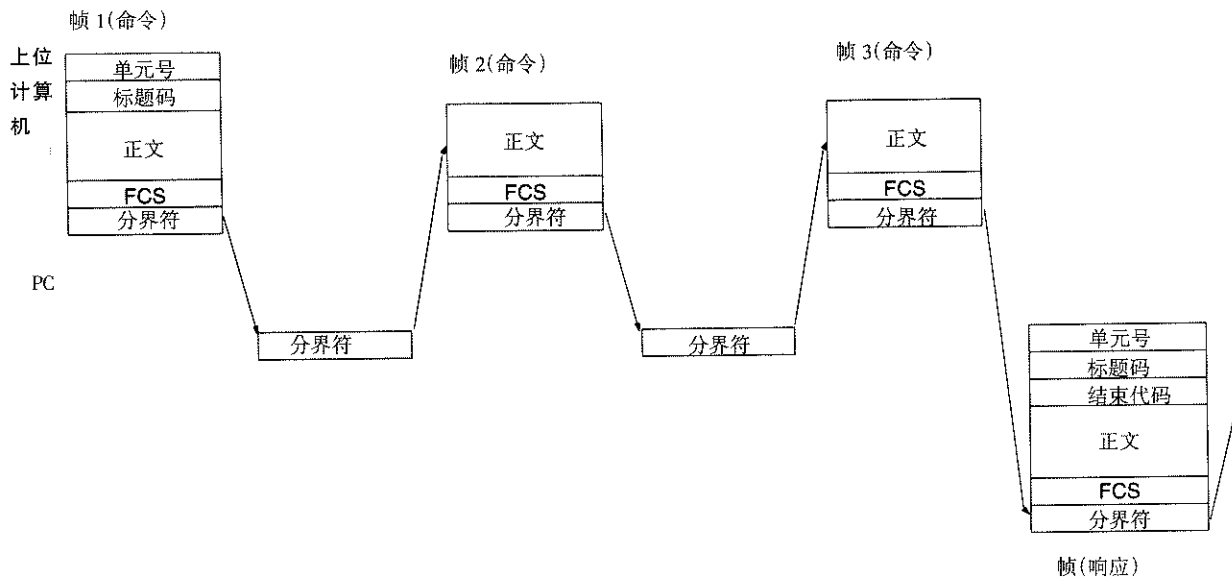
参照“命令格式”项下对应的解释。

长发送

可作为一个单个帧发送的最大的数据块为 131 个字符。因此一个 132 个或更多字符的命令和响应在发送前要分成一个以上帧。当一个传送要分解时,第一帧和中间各帧的末端用分界符代替结束符作为区别标志。

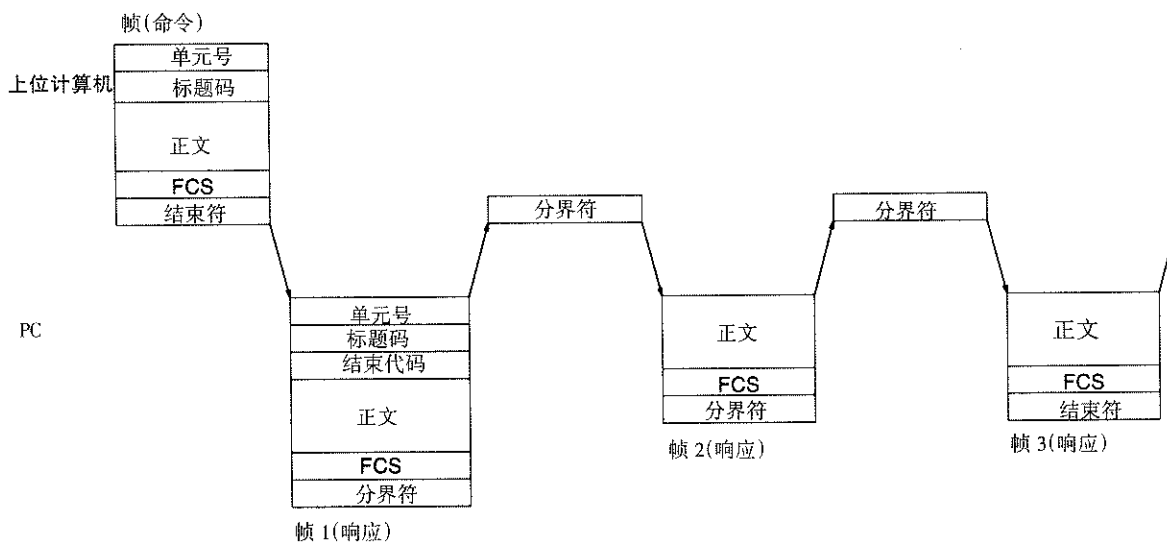
分解命令(上位计算机→PC)

上位计算机发送每帧后,计算机等待 PC 发送的分界符。在分界符发送之后,才能传送下一帧。重复这个过程直到整个命令发送完。



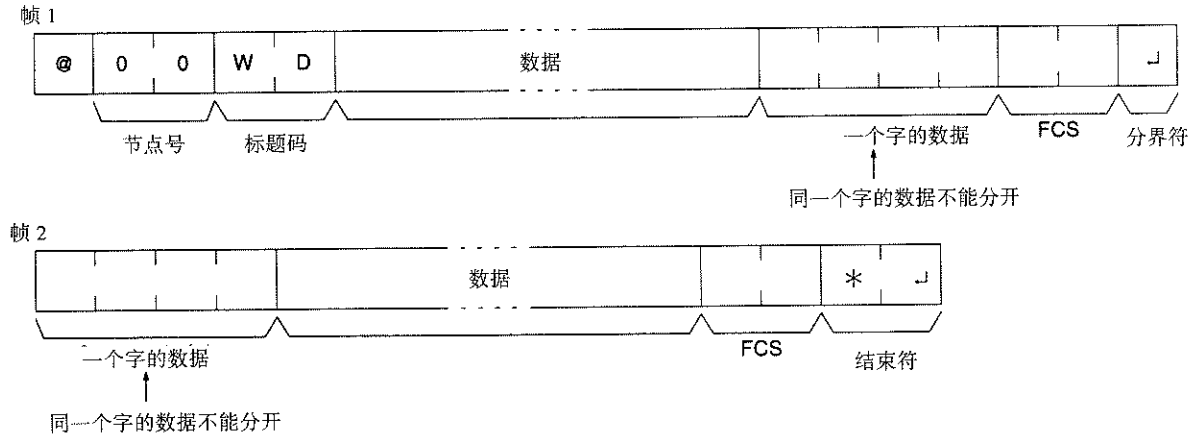
分解响应(PC→上位计算机)

当上位计算机接收到每一个帧时,一个分界符传送到 PC,分界符传送之后,PC 将传送下一个帧,该过程重复执行直到全部响应传送完毕为止。



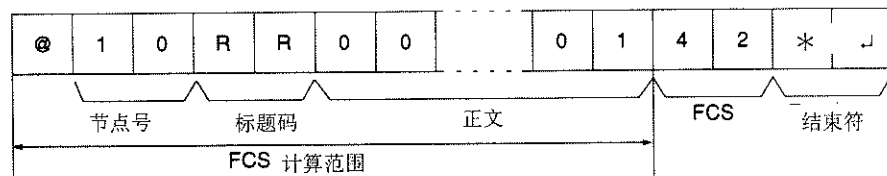
长传送时注意事项

当分解执行写操作的命令时,例如 WR,WL,WC 或 WD,注意不要将写入的单个字分割到不同帧数据。如下图所示,要确保分解帧使其与各字间的分割相符合。



FCS(帧检查顺序)

帧发送时,FCS 置于分界符或结束符前以检查是否发生任何数据错误。FCS 是转换成 2 个 ASCII 字符的 8 位数据。这个 8 位数据是对帧开始数据直到此帧正文结束(即就在 FCS 前)的数据进行“异或”运算的结果。每次接收一帧时计算 FCS 并将结果与包含在帧中的 FCS 比较使之能检查帧中的数据错误。



| ASCII 代码 | | | | |
|----------|----|------|------|---------------|
| @ | 40 | 0100 | 0000 | |
| | | XOR | | |
| 1 | 31 | 0011 | 0001 | |
| | | XOR | | |
| 0 | 30 | 0011 | 0000 | |
| | | XOR | | |
| R | 52 | 0101 | 0010 | |
| | | ⋮ | | |
| 1 | 31 | 0011 | 0001 | |
| 计算结果 | | 0100 | 0010 | |
| | | ↓ | ↓ | 转换成十六进制。 |
| | | 4 | 2 | 作为 ASCII 字符处理 |

FCS 的程序实例

此例示出了对上位计算机接收的一帧数据进行 FCS 检查的 BASIC 子程序。

```

400 *FCSCHECK
410 L = LEN(RESPONSE $)'.....数据发送与接收
420 Q = 0: FCCK $ = " "
430 A $ = RIGHT $ (RESPONSE $ , 1)
440 PRINT RESPONSE $ , AS, L
450 IF A $ = "*" THEN LENG = LEN(RESPONSE $) - 3
      ELSE LENG = LEN(RESPONSE $) - 2
460 FCSP $ = MID(RESPONSE $ , LENG + 1, 2)'.....接收的 FCS 数据
470 FOR I = 1 TO LENG'.....在 FCS 中的字符数目
480 Q = ASC(MID $ (RESPONSE $ , I, 1))XOR Q
490 NEXT I
500 FCSD $ = HEX $ (Q)
510 IF LEN(FCSD $) = 1, THEN FCSD $ = "0" + FCSD $ 'FCS 结果
520 IF FCSD $ < > FCSP $ , THEN FCCK $ = "ERR"
530 PRINT"FCSD $ = ";FCSD $ , "FCSP $ = ";FCSP $ , "FCCK $ = ";FCCK $
540 RETURN

```

- 注
1. 正常的接收数据包括 FCS, 分界符或结束符等等。当传送中发生错误时, 可能会丢失 FCS 或其他数据, 所以对系统编程以克服这些问题。
 2. 在本例中 CR 代码 (CHR \$ (13)) 没有送入 RESPONSE \$。当包含 CR 代码时, 对行 430 和行 450 进行改变。

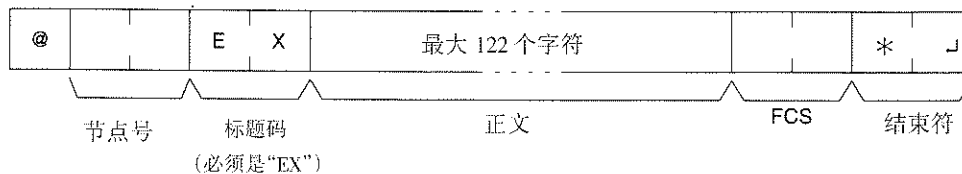
6-4-2 PC 命令

在上位机链接通信中, 命令一般从上位计算机发至 PC, 不过也有可能从 PC 发至上位计算机。在上位机链接模式时, 任何数据都能从 PC 发送至上位计算机。要向上位机发送一个命令时, 在上位机链接模式下, 使用 PC 程序的发送指令 (TXD(48))。

TXD(48) 从指定端口 (RS-232C 端口或外围端口) 输出数据, 关于使用 TXD(48) 的详情参阅 405 页。

接收格式

执行 TXD(48) 时, 存贮在从第一个发送字开始的字中的数据转换成 ASCII 码, 并以如下格式作为一个上位机链接命令输出到上位计算机。在发送发出时自动加入符号 @、节点号、标题码、FCS 和分界符。在上位计算机必须预先准备一段程序用于解释和处理这种格式。



一个字节的的数据 (2 个十六进制数字) 转换成两个 ASCII 字符用于传送, 传送的数据数目两倍于 TXD(48) 指定的字数。传送字符的最大数目为 122, TXD(48) 能指定的最大字节的数目是它的二分之一或 61。

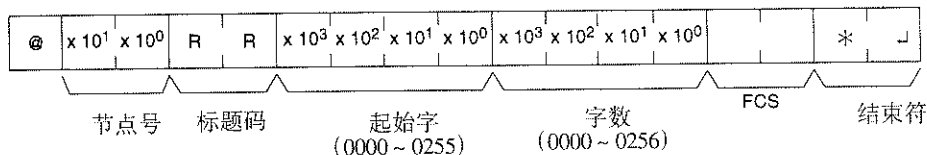
6-5 上位机链接命令

本节解释能从上位计算机发至 PC 的命令。

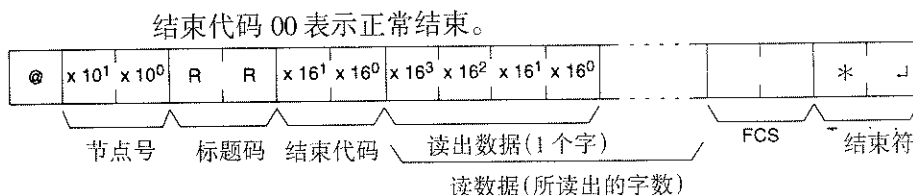
6-5-1 读 IR/SR 区 - RR

从指定字开始读 IR 和 SR 字中指定数目的内容。

命令格式



响应格式



注 当读的字数超出 30 字时,响应将被分割。

参数

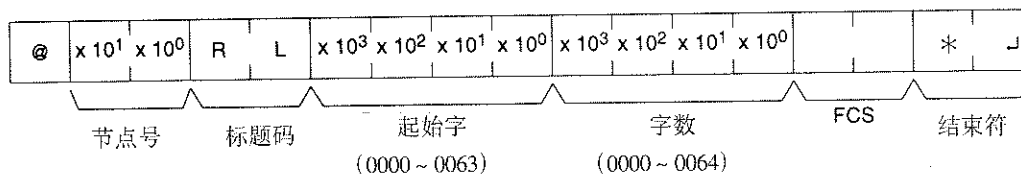
读数据(响应)

命令指定的字数的内容以十六进制数返回作为响应。这些字按序从指定的起始字开始返回。

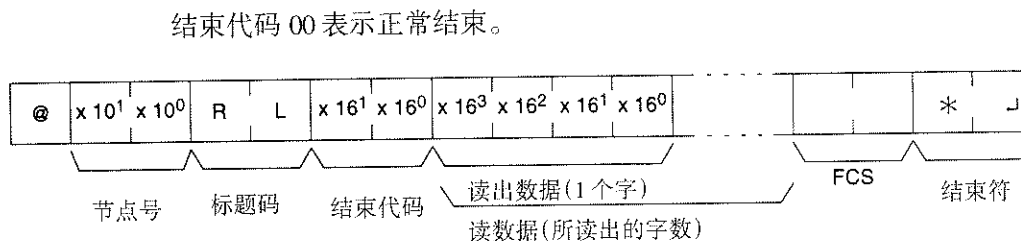
6-5-2 读 LR 区 - RL

从指定字开始,读 LR 字指定数目的内容。

命令格式



响应格式



参数

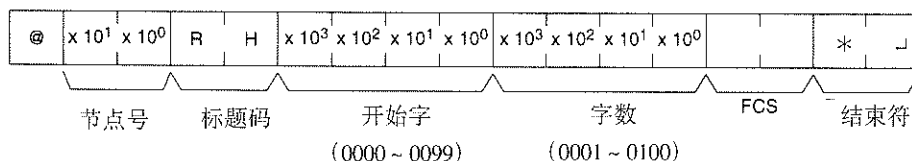
读数据(响应)

命令指定的字数的内容以十六进制返回作为响应。这些字按序从指定的起始字开始返回。

6-5-3 读 HR 区—RH

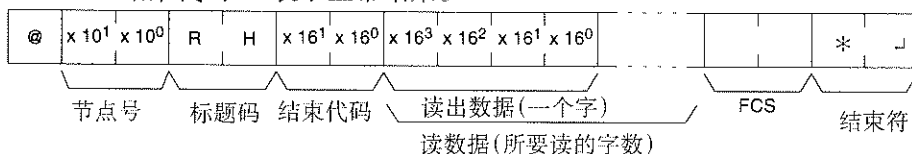
命令格式

从指定字开始,读指定数目的 HR 字的内容。



响应格式

结束代码 00 表示正常结束。



参数

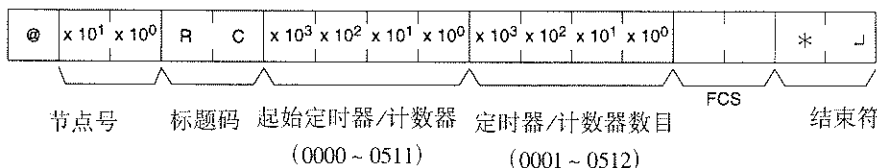
读数据(响应)

命令指定的字数的内容作为响应以十六进制数返回。这些字从指定首字开始按序返回。

6-5-4 读 PV-RC

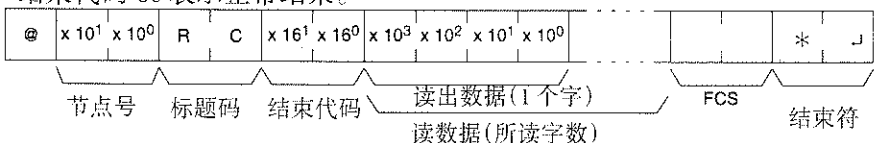
从指定的定时器/计数器开始,读指定数目的定时器/计数器的 PV(当前值)的内容。

命令格式



响应格式

结束代码 00 表示正常结束。



当读的字数超出 30 字时,响应将被分割。

参数

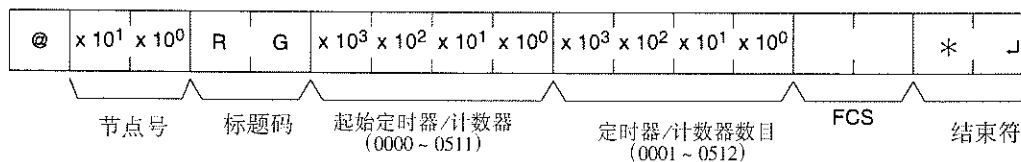
读数据(响应)

命令指定数目的当前值作为响应以十六进制数返回。这些 PV 从指定的起始定时器/计数器起,按顺序返回。

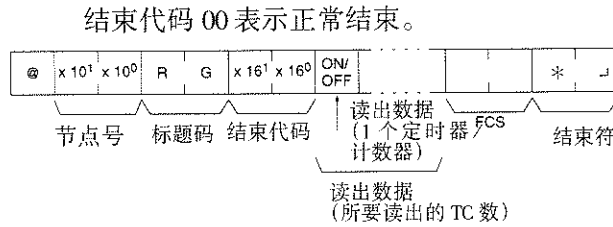
6-5-5 读 TC 状态-RG

从指定的定时器/计数器起,读指定数目的定时器/计数器的完成标志的状态。

命令格式



响应格式



当读的状态超过 123 个定时器/计数器时,响应被分割。

参数

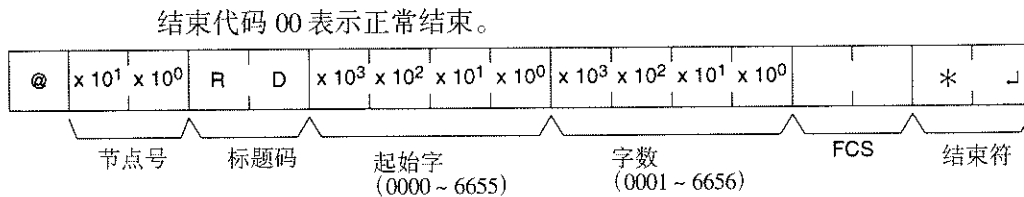
读数据(响应)

命令指定数目的“完成标志”的状态作为响应返回,“1”表示完成标志为 ON。

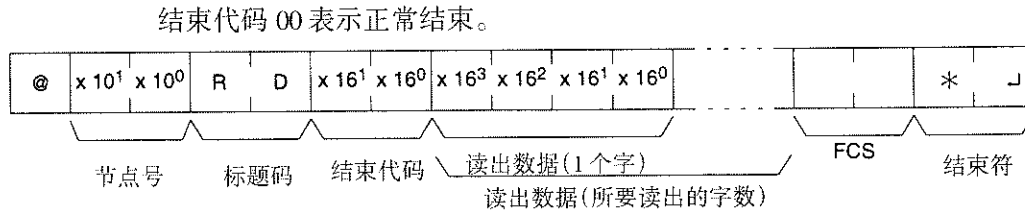
6-5-6 读 DM 区 - RD

从指定字开始,读指定数目的 DM 字的内容。

命令格式



响应格式



参数

读数据(响应)

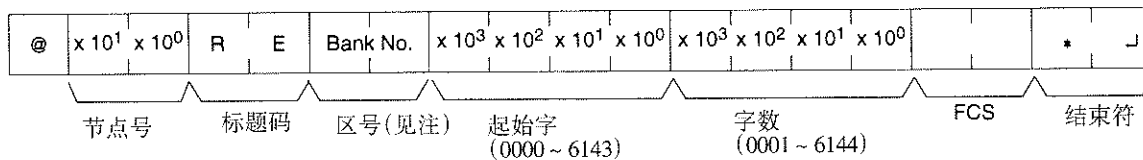
命令指定数目的字的内容作为响应以十六进制数返回。从指定起始字开始按顺序返回。

注 注意 DM 区的组成,因为它随 CPU 单元的型号而变化。

6-5-7 读 EM 区 - RE

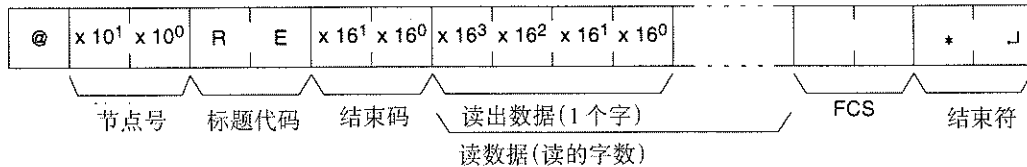
命令格式

从 EM 区指定字开始,读指定数目的 EM 字的内容。



注 输入十六进制 00 以指定区号 0, 或输入 2 个空格以指定当前区号。只有 CQM1H - CPU61 CPU 单元具有 EM 区,并且只有一个区,即 0 区。

响应格式



参数

读数据(响应)

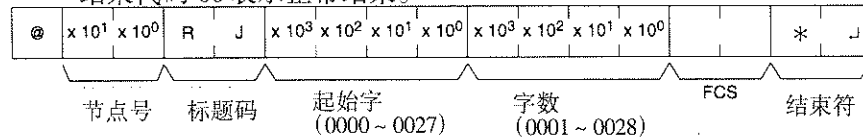
命令指定的字数中的内容作为响应以十六进制数返回。该字从指定的起始字开始按顺序返回。

注 注意 EM 区的结构,因为它根据 CPU 单元型号而变化。

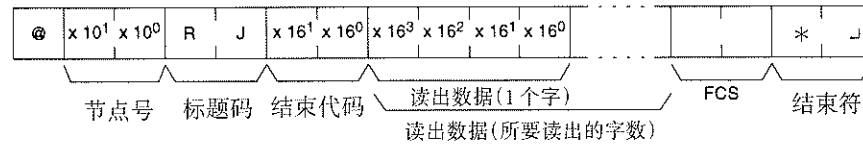
6-5-8 读 AR 区 - RJ

从指定字开始,读指定数目的 AR 字的内容。
结束代码 00 表示正常结束。

命令格式



响应格式



参数

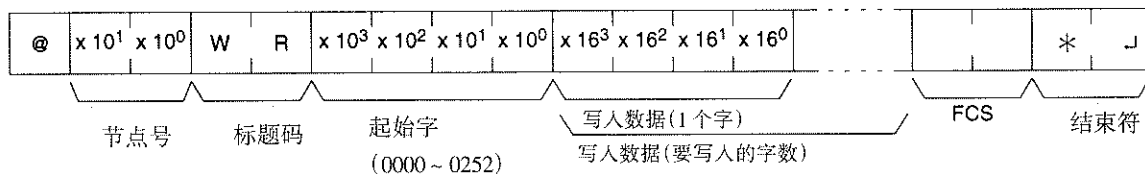
读数据(响应)

命令指定数目的字的内容作为响应以十六进制数返回。从指定的起始字开始按顺序返回。

6-5-9 写 IR/SR - WR

将数据写到 IR/SR 区,从指定字开始,写是逐字完成的。

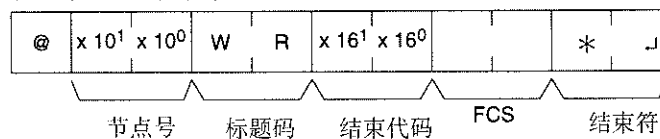
命令格式



注 当写入数据超出 30 字时分割命令。

响应格式

结束代码 00 表示正常结束。



参数

写数据(命令)

从指定字开始按顺序以十六进制指定要写入 IR 或 SR 区的字数的内容。

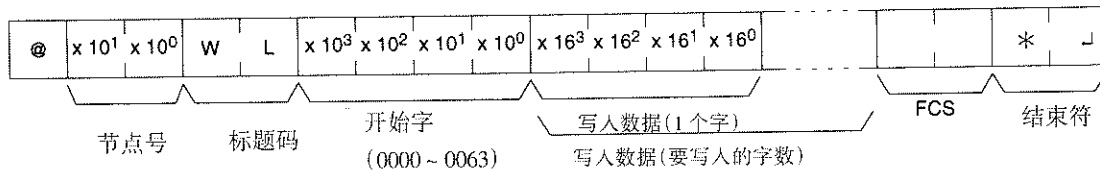
注 根据第一个写入字结果如下。

| 设 置 | 结 果 |
|------------------------------|-------------------------------|
| 第一个写入字 ≤ 252 | 数据写入直到字 252, 但不写入其它字, 返回正常响应。 |
| $253 \leq$ 第一个写入字 ≤ 255 | 无数据写入并返回正常响应。 |
| $255 <$ 第一个写入字 | 无数据写入并且发生错误。 |

6-5-10 写 LR 区—WL

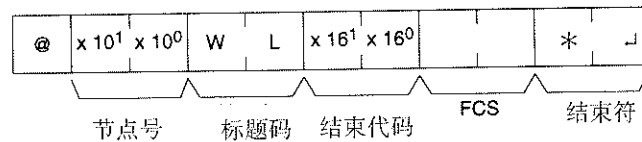
从指定字开始, 将数据写入 LR 区。字是逐字写入的。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

写数据(命令)

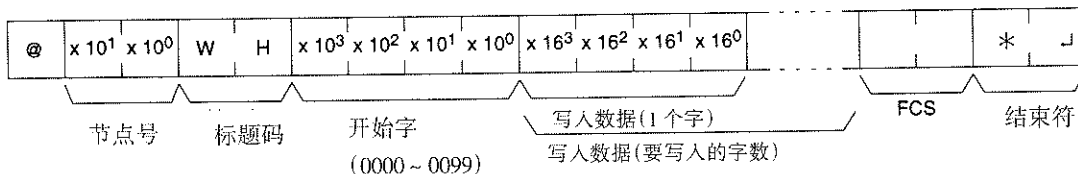
从指定起始字开始, 以十六进制按顺序指定要写入 LR 区的字数的内容。

注 如果指定要写的的数据超出允许范围, 会产生一个错误并且写操作将不执行。
例如: 如果 60 是指定要写的开始字并且指定写 5 个字, 那么 64 就是写数据的最后一个字, 由于 LR64 超出了 LR 区界限, 所以命令将不执行。

6-5-11 写 HR 区—WH

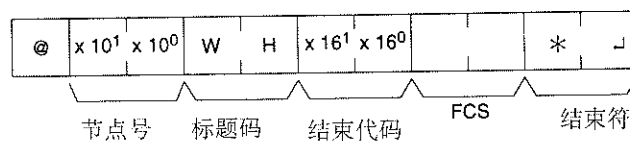
从指定字开始, 将数据写入 HR 区。写入是逐字完成的。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

写数据(命令)

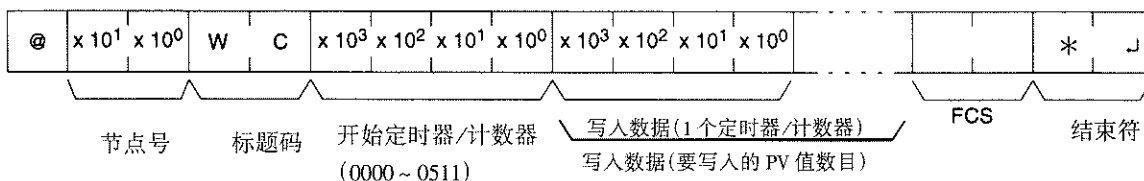
从指定起始字开始,按顺序以十六进制指定要写入 HR 区的字数的内容。

注 如果指定要写的的数据超出允许范围,会产生一个错误并且写操作将不执行。
 例如:如 98 是指定要写的开始字,并且指定写 3 个字数据,那么 100 就是要写数据的最后一个字,因为 HR100 超出了 HR 区界限,所以命令将不执行。

6-5-12 写 PV-WC

命令格式

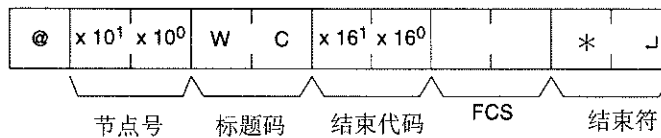
从指定的定时器/计数器开始,写入定时器/计数器的 PV 值(当前值)。



注 写超过 29 字数据时分割命令。

响应格式

结束代码 00 表示正常结束。



参数

写入数据(命令)

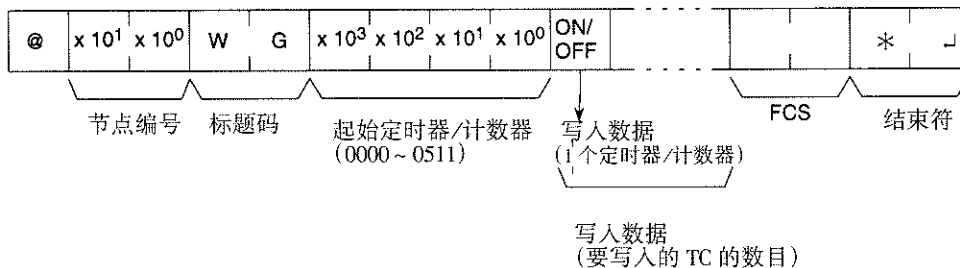
从起始定时器/计数器起,以十进制数(BCD)指定要写的定时器/计数器数的当前值。

注 1. 当此命令用于将数据写入 PV 区时,被写的定时器/计数器的完成标志将转为 OFF。
 2. 如果指定要写入的数据超过允许的范围,将产生一个错误,并且写操作不会执行。例如,如果指定写入起始字为 510,指定写入 3 个字,那么要写入数据的最末一个字是 512,因为 TC 512 已在这个区边界以外,所以这个命令不会执行。

6-5-13 写 TC 状态-WG

命令格式

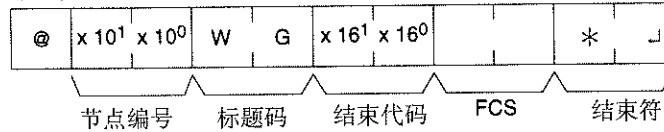
从指定的定时器/计数器(编号)起,在 TC 区写入定时器和计数器的完成标志的状态。按编号一个个写。



注 写的状态超过 118 个定时器/计数器时,分割命令。

响应格式

结束代码 00 表示正常结束。



参数

写入数据(命令)

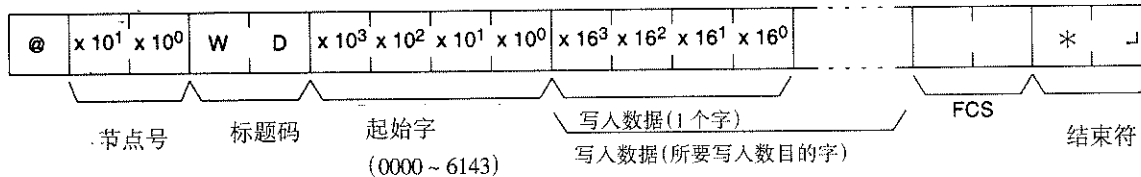
指定要写入数目的定时器/计数器的完成标志的状态,从起始字起按顺序如 ON(即“1”)或 OFF(即“0”)写入,如果完成标志为 ON,则表示定时或计数已到。

注 如果指定要写入的数据超出允许的范围。将出现错误信息并且写操作将不执行。例如指定起始字为 510,要写入 3 个字,那么要写入数据的最后一个字就成为 512,由于 TC512 是在此区之外,这条命令就不能执行。

6-5-14 写 DM 区 - WD

从指定字开始将数据写入 DM 区,逐字写入。

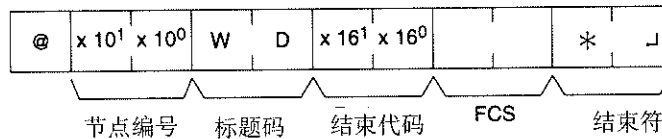
命令格式



注 当写入数据超出 29 字时,分割命令。

响应格式

结束代码 00 表示正常结束。



参数

写入数据(命令)

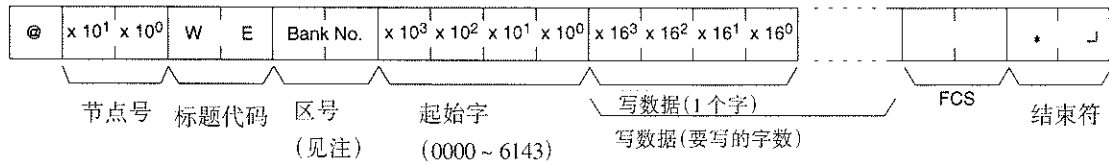
以十六进制数按顺序指定要写入 DM 区(从指定字开始)的字的的内容。

- 注
1. 如果指定要写入的数据超出允许的范围,将产生一个错误,并且写操作不执行。例如:指定要写入的起始字 6142,并指定要写入 3 个字,那末 6144 成为要写入数据的最后一个字,由于 DM6144 在能写入范围以外,此命令就不能执行。
 2. 要注意 DM 的构成,对不同型号的 CPU 单元,DM 区不一样。

6-5-15 写 EM 区 - WE

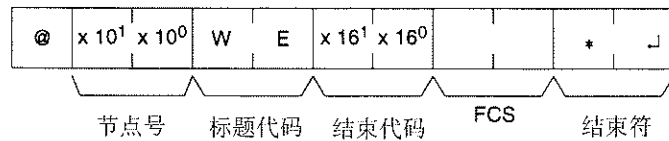
将数据写入指定的 EM 区中,从指定字开始。写是逐字进行的。

命令格式



注 输入十六进制数值 00 指定区号 0 或输入两个空格以指定当前区, 只有 CQM1H - CPU61 单元有 EM 区, 并且只有一个区, 即 0 区。

响应格式



参数

写入数据(命令)

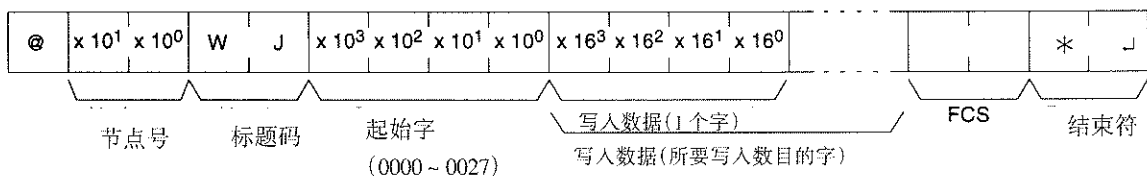
从指定的起始字开始,依次指定要写入 EM 区的字数的内容。

- 注
1. 如果指定的数据超出允许范围,将产生错误并且写操作不执行。例如如果 6142 被指定为要写的起始字,并指定数据为 3 个字,则 6144 为最后一个写入字,由于 6144 超出可写范围,命令将不被执行。
 2. 注意在 EM 区的结构,因为它随 CPU 单元型号的不同而有所不同。

6-5-16 写 AR 区 - WJ

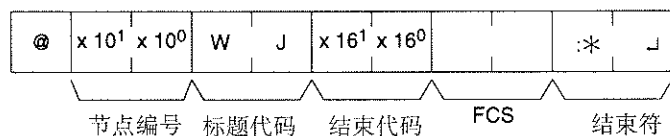
从指定字开始逐字将数据写入 AR 区。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

写入数据(命令)

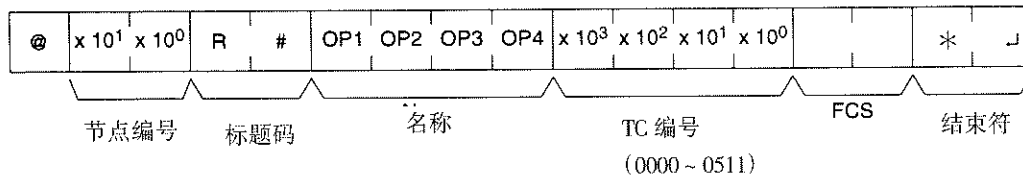
从指定起始字起以十六进制数顺序指定要写入 AR 区的字数的内容。

注 如果指明要写入的数据超出允许的范围,会产生错误,并且不执行此写操作。例如:指定要写入的起始字为 26,并且要写入 3 个字数据,那么 28 是要写入的最后一个字,由于 AR28 不在可写入范围,所以这个命令不能执行。

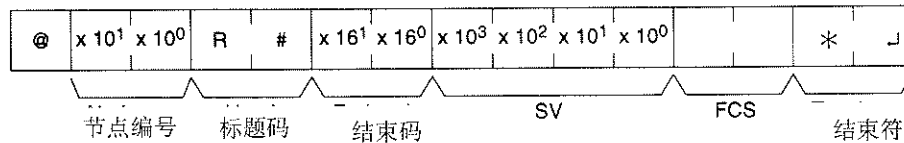
6-5-17 SV 读取 1—R #

按指定的 TC 编号在用户程序中寻找第一个 TIM, TIMH(15), TTIM, CNT 和 CNTR(12)指令,并读出 PV 值(假定是以常数设置)。所读出的 SV 是 4 位十进制数(BCD)。从起始开始搜索程序,因此约需要 10 秒才能得到响应。

命令格式



响应格式



参数

名称, TC 编号(命令)

在“名称”中,指定读 SV 的指令。用四个字符设定,在“TC 编号”上指定指令所使用的定时器/计数器的编号。

| 指 令 名 称 | | | | 分 类 |
|---------|-----|-----|-----|-------|
| OP1 | OP2 | OP3 | OP4 | |
| T | I | M | (空) | 定时器 |
| T | I | M | H | 高速定时器 |
| T | T | I | M | 累积定时器 |
| C | N | T | (空) | 计数器 |
| C | N | T | R | 可逆计数器 |

SV(响应)

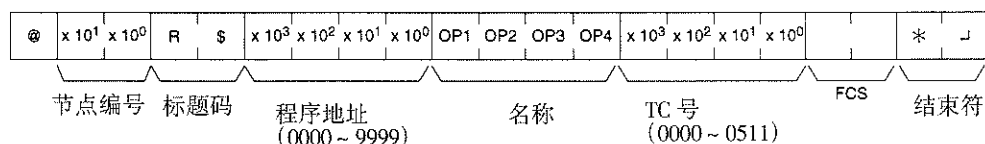
返回 SV 常数。

- 注
1. 在“名称”上指定的指令必须是四字符。
 2. 如果同一指令在程序中使用多次,那么只读出第一个。
 3. 只在明确常数 SV 已被设置时,才能使用这条指令。
 4. 如果 SV 不是常数,则响应结束代码将指示出错(16)。

6-5-18 SV 读取 2-R\$

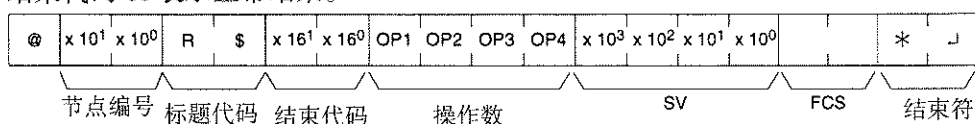
读常数 SV 或贮存 SV 的字地址,所读出的 SV 是一个 4 位十进制数(BCD),它作为 TIM, TIMH(15), TTIM, CNT 或 CNTR(12)指令的第二操作数,在用户程序的指定程序地址,只有当程序小于 10000 时才能使用这个指令。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称, TC 编号(命令)

在“名称”上指定读 SV 指定的名称,用 4 个字符设置。在“TC 号”上,指定指令所用的定时器/计数器号。

| 指 令 名 | | | | 分 类 |
|-------|-----|-----|-----|-------|
| OP1 | OP2 | OP3 | OP4 | |
| T | I | M | (空) | 定时器 |
| T | I | M | H | 高速定时器 |
| T | T | I | M | 累积定时器 |
| C | N | T | (空) | 计数器 |
| C | N | T | R | 可逆计数器 |

操作数, SV(响应)

“操作数”上返回指示 SV 分类的名字,“SV”上返回 SV 所贮存的字地址或常数 SV。

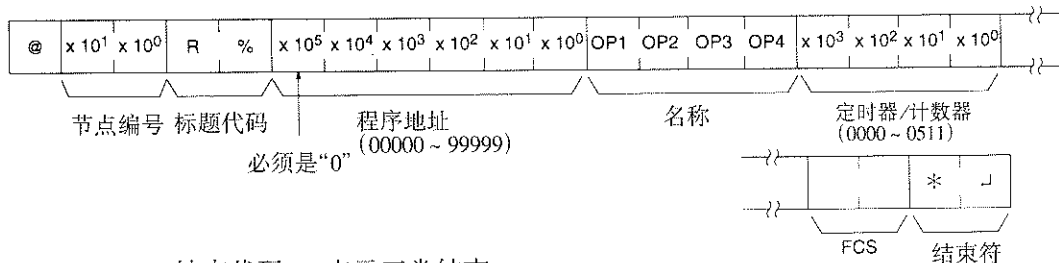
| 操 作 数 | | | | 分 类 | 常数或字地址 |
|-------|-----|-----|-----|---------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0255 |
| L | R | (空) | (空) | LR | 0000 ~ 0063 |
| H | R | (空) | (空) | HR | 0000 ~ 0099 |
| A | R | (空) | (空) | AR | 0000 ~ 0027 |
| D | M | (空) | (空) | DM | 0000 ~ 6655 |
| D | M | * | (空) | DM(间接) | 0000 ~ 6655 |
| E | M | (空) | (空) | EM | 0000 ~ 6143 |
| E | M | * | (空) | EM(间接) | 0000 ~ 6143 |
| C | O | N | (空) | 常数 | 0000 ~ 9999 |

- 注
1. 指令名和操作数区指示符必须为四个字符,用空格填入来代替空字符,使总数为 4 个字符。
 2. 只有 CQM1H-CPU61 CPU 单元具有 EM 区。

6-5-19 SV 读取 3-R%

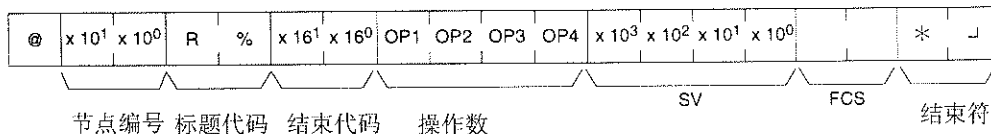
读常数 SV 或存储 SV 的字地址。所读出的 SV 是一个 4 位十进制数 (BCD)，它作为 TIM, TTMH(15), TTIM, CNT 或 CNTR(12) 指令的第二操作数, 在用户程序的指定程序地址, 此命令中, 程序地址可以指定的程序达 99999 步。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称, TC 编号 (命令)

在“名称”上, 指定要读出 SV 的指令的名称, 用四位字符设定, 在“TC 编号”上, 指定指令所用的定时器/计数器号。

| 指令名 | | | | 分类 | TC 号范围 |
|-----|-----|-----|-----|-------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| T | I | M | (空) | 定时器 | 0000 ~ 0511 |
| T | I | M | H | 高速定时器 | |
| T | T | I | M | 累积定时器 | |
| C | N | T | (空) | 计数器 | |
| C | N | T | R | 可逆计数器 | |

操作数, SV (响应)

“操作数”上返回指示 SV 类别的名称, “SV”上返回存储 SV 的字地址或常数 SV。

| 操作数 | | | | 分类 | 常数或字地址 |
|-----|-----|-----|-----|---------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| C | I | O | (空) | IR 和 SR | 0000 ~ 0255 |
| L | R | (空) | (空) | LR | 0000 ~ 0063 |
| H | R | (空) | (空) | HR | 0000 ~ 0099 |
| A | R | (空) | (空) | AR | 0000 ~ 0027 |
| D | M | (空) | (空) | DM | 0000 ~ 6655 |
| D | M | * | (空) | DM(间接) | 0000 ~ 6655 |
| E | M | (空) | (空) | EM | 0000 ~ 6143 |
| E | M | * | (空) | EM(间接) | 0000 ~ 6143 |
| C | O | N | (空) | 常数 | 0000 ~ 9999 |

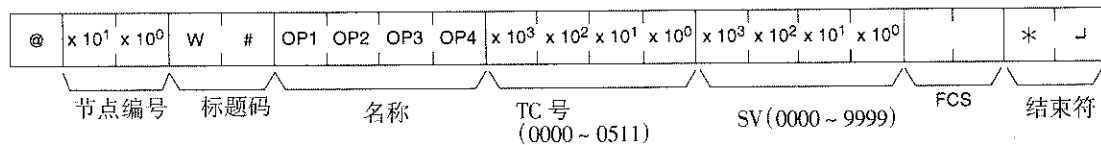
注

1. 指令名和操作数区指示符必须是四个字符。用空格来代替空字符, 使总数为 4 个字符。
2. 只有 CQM1H - CPU61 CPU 单元具有 EM 区。

6-5-20 SV 修改 1-W#

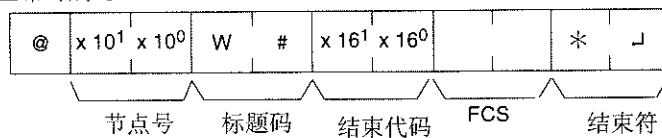
在用户程序中搜寻所指定的 TIM, TIMH(15), TTIM, CNT 或 CNTR(12) 指令的第一个实例, 并将此指令的第二个字用新的常数 SV 值来修改 SV。将从头开始搜索程序, 因此可能需要约 10 秒才能产生一个响应。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称, TC 编号(命令)

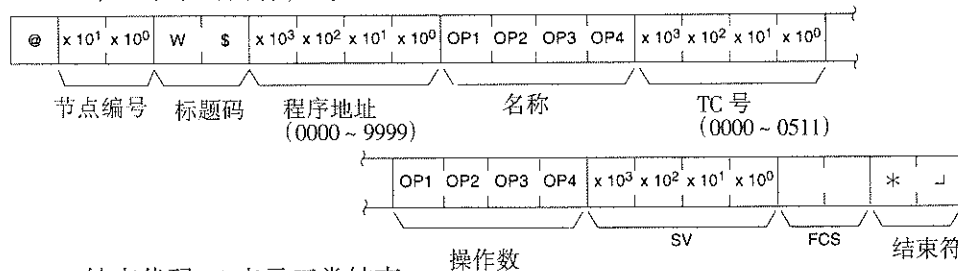
在“名称”上指定要修改 SV 的指令名称, 为四字符, 在“TC 编号”上指定指令所使用的定时器/计数器编号。

| 指令名称 | | | | 分类 |
|------|-----|-----|-----|-------|
| OP1 | OP2 | OP3 | OP4 | |
| T | I | M | (空) | 定时器 |
| T | I | M | H | 高速定时器 |
| T | T | I | M | 累积定时器 |
| C | N | T | (空) | 计数器 |
| C | N | T | R | 可逆计数器 |

6-5-21 SV 修改 2-W\$

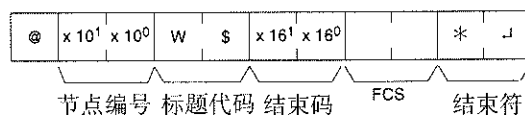
在用户程序指定的程度地址, 修改 TIM, TIMH(15), TTIM, CNT 或 CNTR(12) 的第二个字的内容, 此修改只能到程序的 9999 步。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称, TC 编号(命令)

在“名称”上指定修改 SV 的指令名称, 为四字符。在“TC 编号”上指定指令所使用的定时器/计数器编号。

| 指 令 名 | | | | 分 类 |
|-------|-----|-----|-----|-------|
| OP1 | OP2 | OP3 | OP4 | |
| T | I | M | (空) | 定时器 |
| T | I | M | H | 高速定时器 |
| T | T | I | M | 累积定时器 |
| C | N | T | (空) | 计数器 |
| C | N | T | R | 可逆计数器 |

操作数, SV(响应)

在“操作数”上指定 SV 分类名称, 用四个字符表示。在“SV”上指定贮存 SV 值的字地址或 SV 常数。

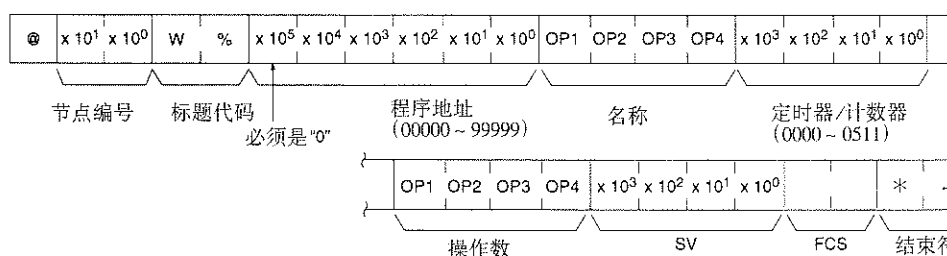
| 操 作 数 | | | | 分 类 | 常数或字地址 |
|-------|-----|-----|-----|---------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0255 |
| L | R | (空) | (空) | LR | 0000 ~ 0063 |
| H | R | (空) | (空) | HR | 0000 ~ 0099 |
| A | R | (空) | (空) | AR | 0000 ~ 0027 |
| D | M | (空) | (空) | DM | 0000 ~ 6655 |
| D | M | * | (空) | DM(间接) | 0000 ~ 6655 |
| E | M | (空) | (空) | EM | 0000 ~ 6143 |
| E | M | * | (空) | EM(间接) | 0000 ~ 6143 |
| C | O | N | (空) | 常数 | 0000 ~ 9999 |

注 只有 CQM1H-CPU61 CPU 单元有 EM 区。

6-5-22 SV 修改 3-W%

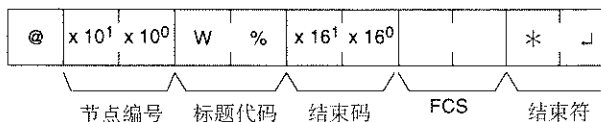
在用户程序的指定程序地址修改 TIM, TIMH(15), TTIM, CNT 或 CNTR(12) 的第二个字的内容。用此命令, 能指定程序地址的程序达 99999 步。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称,TC 编号(命令)

在“命令”上指定修改 SV 的指令的名称,名称为四字符。在“TC 编号”上指定指令所用的定时器/计数器的编号。

| 指 令 名 | | | | 分 类 | TC 号范围 |
|-------|-----|-----|-----|-------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| T | I | M | (空) | 定时器 | 0000 ~ 0511 |
| T | I | M | H | 高速定时器 | |
| T | T | I | M | 累积定时器 | |
| C | N | T | (空) | 计数器 | |
| C | N | T | R | 可逆计数器 | |

操作数,SV(响应)

在“操作数”上指定 SV 分类的名称,四个字符表示,在“SV”上指定贮存 SV 值的字地址或 SV 常数。

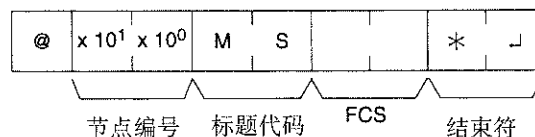
| 操 作 数 | | | | 分 类 | 常数或字地址 |
|-------|-----|-----|-----|---------|-------------|
| OP1 | OP2 | OP3 | OP4 | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0252 |
| L | R | (空) | (空) | LR | 0000 ~ 0063 |
| H | R | (空) | (空) | HR | 0000 ~ 0099 |
| A | R | (空) | (空) | AR | 0000 ~ 0027 |
| D | M | (空) | (空) | DM | 0000 ~ 6655 |
| D | M | * | (空) | DM(间接) | 0000 ~ 6655 |
| E | M | (空) | (空) | EM | 0000 ~ 6143 |
| E | M | * | (空) | EM(间接) | 0000 ~ 6143 |
| C | O | N | (空) | 常数 | 0000 ~ 9999 |

注 仅 CQM1H - CPU61 CPU 单元有 EM 区。

6-5-23 读状态 - MS

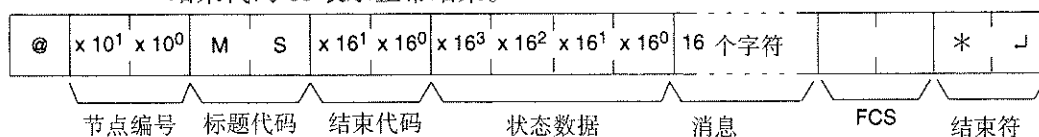
读 PC 操作条件。

命令格式



响应格式

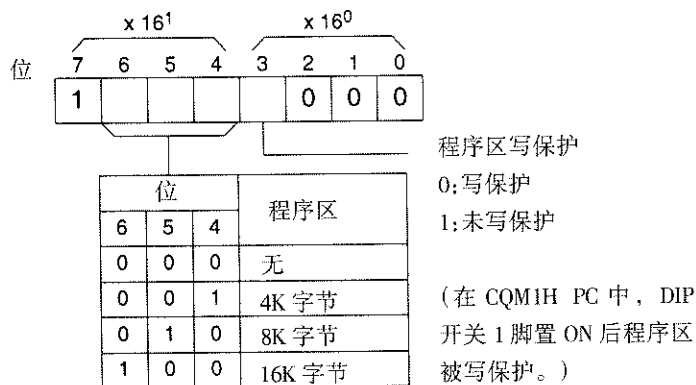
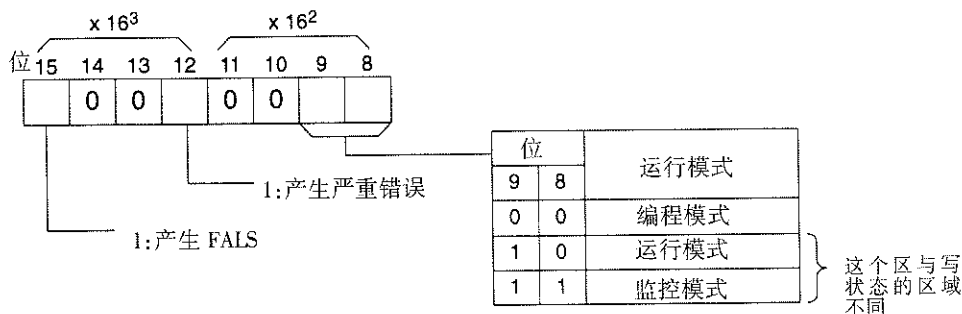
结束代码 00 表示正常结束。



参数

状态数据,消息(响应)

“状态数据”由四位十六进制数组成(二字节),左面字节表示 CPU 单元的运行模式,右面字节表示程序区的大小。

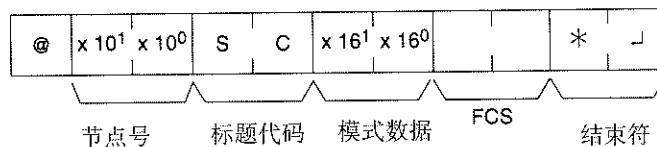


“消息”参数是在命令执行时所存在的 FAL/FALS 号,当无消息时,该参数被忽略。

6-5-24 写状态 - SC

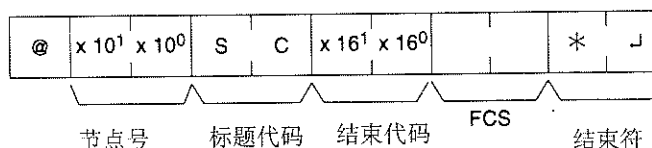
修改 PC 操作模式。

命令格式



响应格式

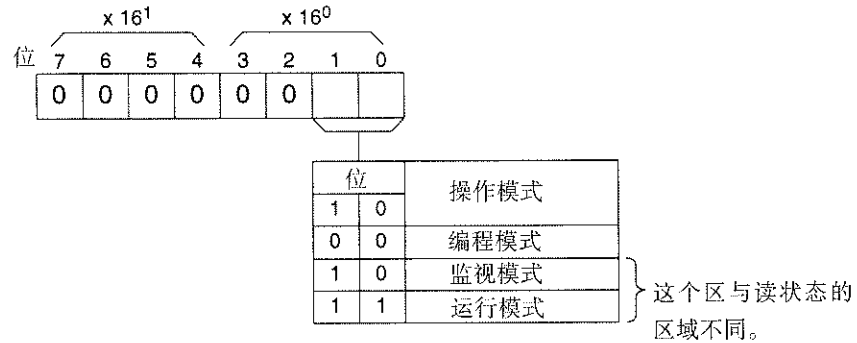
结束代码 00 表示正常结束。



参数

模式数据(命令)

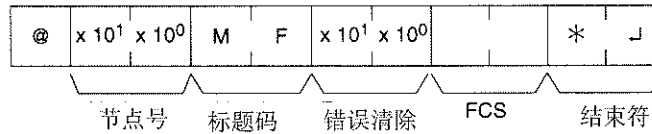
“模式数据”由二位十六进制数(一个字节)组成。最右边二位标明 PC 操作模式。其他位全部置“0”。



6-5-25 错误读—MF

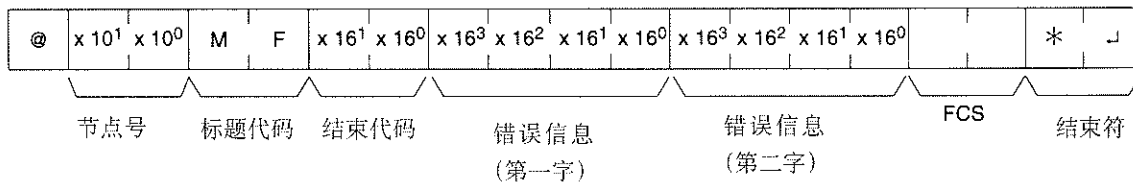
读取并清除 PC 中的错误。也检查以前的错误是否已清除。

命令格式



响应格式

结束代码 00 表示正常结束。



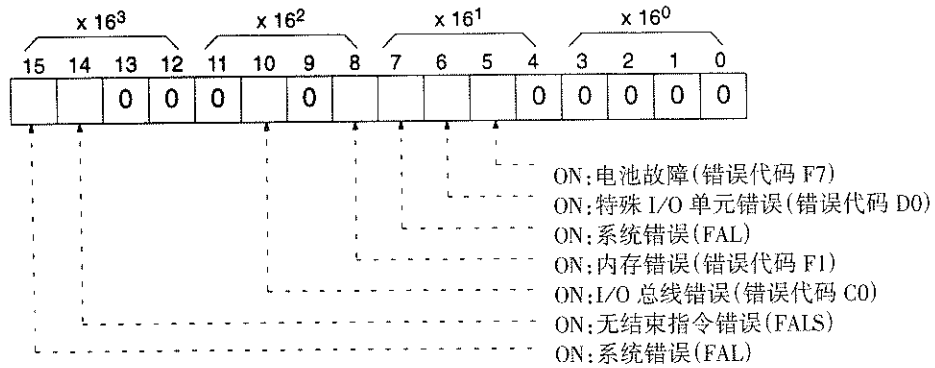
参数

错误清除(命令)

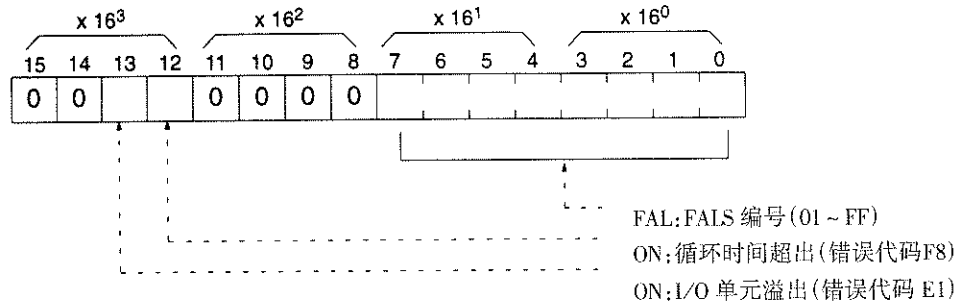
01 表示清除错误, 00 表示不清除错误 (BCD), 仅当 PC 处于编程模式时才能清除致命错误。

错误信息 (响应)
 错误信息含有二个字。

第一个字



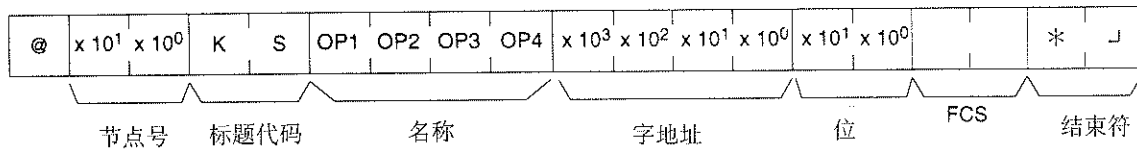
第二个字



6-5-26 强制置位 - KS

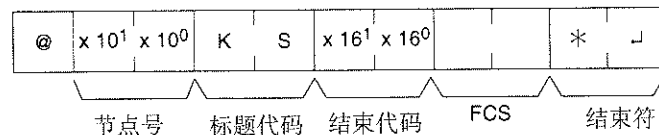
对 IR,SR,LR,HR,AR 或 TC 区中的某一位进行强制置位。一次只能强制一个位。
 一旦某个位已经强制置位或复位, 它的状态将保持不变直到强制置位/复位
 取消 (KC) 命令或者下一个强制置位/复位命令被传送为止。

命令格式



响应格式

结束代码 00 表示正常结束。



参数 名称,字地址,位(命令)
 在“名称”上指定要强制置位的区域(即 IR,SR,LR,HR,AR 或 TC)。名称用四个字符表示,在“字地址”上指定字地址,在“位”上指定要强制置位的位的编号。

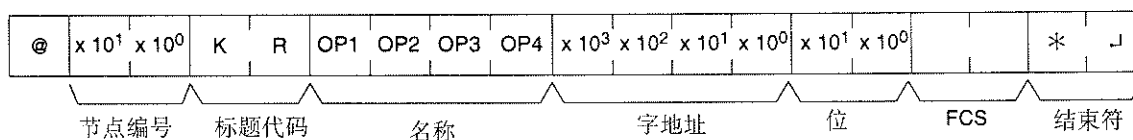
| 名 称 | | | | 分 类 | 字地址设置范围 | 位 |
|-----|-----|-----|-----|-------------|-------------|--------------|
| OP1 | OP2 | OP3 | OP4 | | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0252 | 00 ~ 15(十进制) |
| L | R | (空) | (空) | LR | 0000 ~ 0063 | |
| H | R | (空) | (空) | HR | 0000 ~ 0099 | |
| A | R | (空) | (空) | AR | 0000 ~ 0027 | |
| T | I | M | (空) | 完成标志(定时器) | 0000 ~ 0511 | 总是 00 |
| T | I | M | H | 完成标志(高速定时器) | | |
| T | T | I | M | 完成标志(累积定时器) | | |
| C | N | T | (空) | 完成标志(计数器) | | |
| C | N | T | R | 完成标志(可逆计数器) | | |

注 在“名称”上指定的区名必须是四个字符。任一空白处填以空格使成为完整的 4 字符。

6-5-27 强制复位 - KR

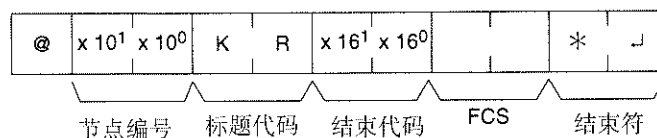
在 IR,SR,LR,HR,AR 或 TC 区中强制复位一个位。一次只能复位一个位。一个位一旦被强制置位或复位,它的状态将保持不变直到强制置位/复位取消(KC)命令或强制置位/复位命令被传送。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称,字地址,位(命令)
 在“名称”上指定要强制复位的区域(即 IR,SR,LR,HR,AR 或 TC),名称用四个字符表示,在“字地址”上指定字的地址,在“位”上指定要强制复位的位的编号。

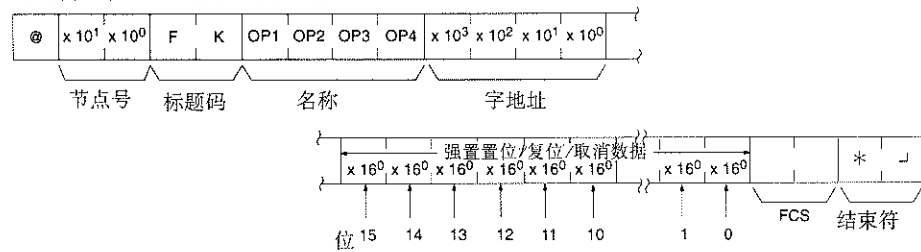
| 名 称 | | | | 分 类 | 字地址设置范围 | 位 |
|-----|-----|-----|-----|-------------|-------------|--------------|
| OP1 | OP2 | OP3 | OP4 | | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0252 | 00 ~ 15(十进制) |
| L | R | (空) | (空) | LR | 0000 ~ 0063 | |
| H | R | (空) | (空) | HR | 0000 ~ 0099 | |
| A | R | (空) | (空) | AR | 0000 ~ 0027 | |
| T | I | M | (空) | 完成标志(定时器) | 0000 ~ 0511 | 总是 00 |
| T | I | M | H | 完成标志(高速定时器) | | |
| T | T | I | M | 完成标志(累积定时器) | | |
| C | N | T | (空) | 完成标志(计数器) | | |
| C | N | T | R | 完成标志(可逆计数器) | | |

注 在“名称”上指定的区名必须是四个字符,如果少于 4 个字符,在数据区名称之后加入空格。

6-5-28 多位强制置位/复位 - FK

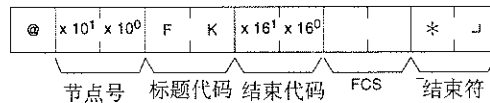
强制置位、强置复位或取消 IR,SR,LR,HR,AR 或 TC 区中某字各个位的状态。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

名称,字地址(命令)

在“名称”上指定要强制置位/复位的数据区(即 IR,SR,LR,HR,AR 或 TC)。名称用 4 个字符表示 0 在“字地址”上指定要强制置位/复位的字地址。

| 名 称 | | | | 分 类 | 字地址设置范围 | 位 |
|-----|-----|-----|-----|-------------|-------------|--------------|
| OP1 | OP2 | OP3 | OP4 | | | |
| C | I | O | (空) | IR 或 SR | 0000 ~ 0252 | 00 ~ 15(十进制) |
| L | R | (空) | (空) | LR | 0000 ~ 0063 | |
| H | R | (空) | (空) | HR | 0000 ~ 0099 | |
| A | R | (空) | (空) | AR | 0000 ~ 0027 | |
| T | I | M | (空) | 完成标志(定时器) | 0000 ~ 0511 | 总是 00 |
| T | I | M | H | 完成标志(高速定时器) | | |
| T | T | I | M | 完成标志(累积定时器) | | |
| C | N | T | (空) | 完成标志(计数器) | | |
| C | N | T | R | 完成标志(可逆计数器) | | |

强制置位/复位/取消数据(命令)

如果指定了一个定时器或计数器的完成标志, 仅位 15 是有效的, 其它位将被忽略。对于定时器/计数器仅可用强制置位和强制复位。

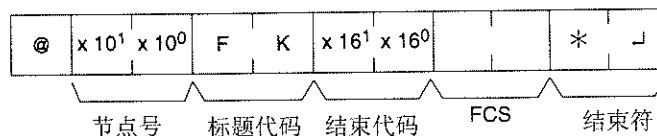
如果指定了一个字地址, 字的内容指明指定字中所需要处理的每一位, 如下表所示。

| BCD 设置 | 处 理 |
|--------|-------------|
| 0 | 无动作(位状态不改变) |
| 2 | 复位 |
| 3 | 置位 |
| 4 | 强制复位 |
| 5 | 强制置位 |
| 8 | 强制置位/复位状态取消 |

仅仅置位或复位的位, 在程序下次执行时就可改变状态, 但强制置位或强制复位将一直保持强制状态, 直到该强制被清除为止。

结束代码 00 表示正常结束。

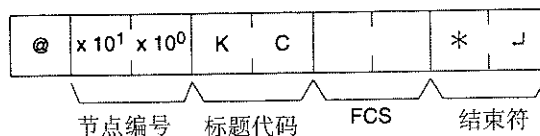
响应格式



6-5-29 取消强制置位/复位 - KC

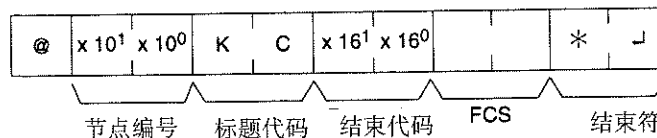
取消所有强制置位和强制复位位 (包括用强制置位、强制复位和多点强制置位/复位命令做的那些)。如已置位了多个位, 那么强制状态的取消是对所有位的。不能用 KC 来逐位取消强制。

命令格式



响应格式

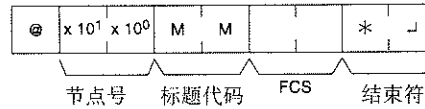
结束代码 00 表示正常结束。



6-5-30 读 PC 型号—MM

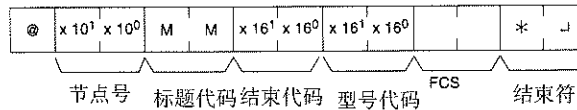
读 PC 型号。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

型号代码

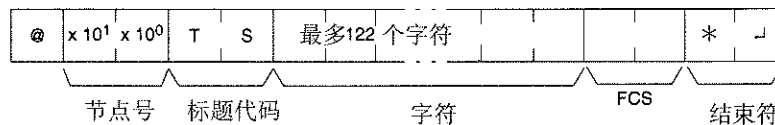
“型号代码”用 2 位十进制数表示 PC 型号。

| 型号代码 | 型 号 |
|------|---|
| 01 | C250 |
| 02 | C500 |
| 03 | C120 |
| 0E | C2000 |
| 10 | C1000H |
| 11 | CQM1H/C2000H/CQM1/CPM1/CMP1A/CPM2A/CPM2C/SRM1 |
| 12 | C20H/C28H/C40H/C200H/C200HS |
| 20 | CV500 |
| 21 | CV1000 |
| 22 | CV2000 |
| 40 | CVM1 - CPU01 - E |
| 41 | CVM1 - CPU11 - E |
| 42 | CVM1 - CPU21 - E |

6-5-31 测试 - TS

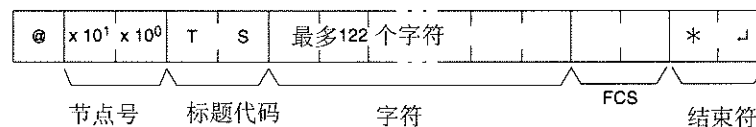
无更改地返回由上位计算机发送的一个数据块。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

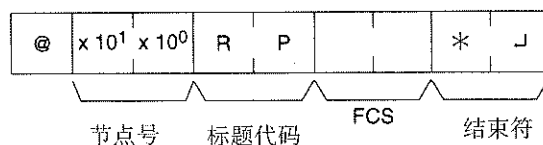
字符(命令,响应)

对于命令,此设置指定任何不是回车(CHR \$(13))的字符。对于响应,如果测试成功,将返回与命令指定相同的字符。

6-5-32 读程序—RP

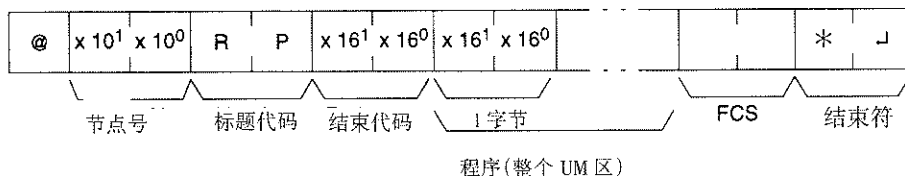
读取 PC 用户程序区用机器语言(目标码)表示的内容。该内容从开始到结束作为一个块来读取。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

程序(响应)

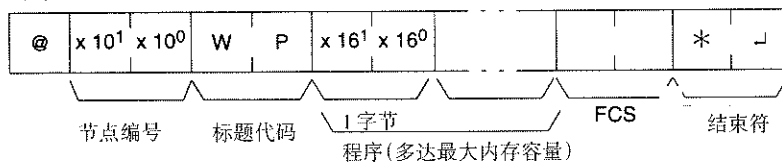
从整个程序区读出程序。

注 执行中止(XZ)命令以停止该操作。

6-5-33 写入程序—WP

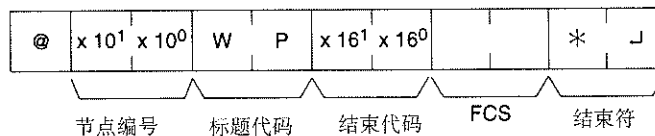
将上位计算机传送来的机器语言(目标码)程序,写入 PC 用户程序区。其内容作为一个块。从头开始写入。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

程序(命令)

程序数据多达最大的内存容量。

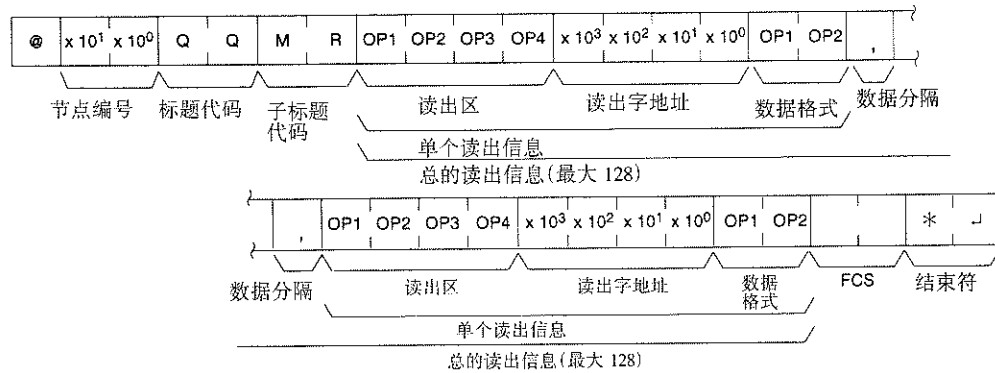
6-5-34 复合命令——QQ

在 PC 中先登录所要读出的位,字和定时器/计数器,然后整批读出所有这些状态。

登录读出信息

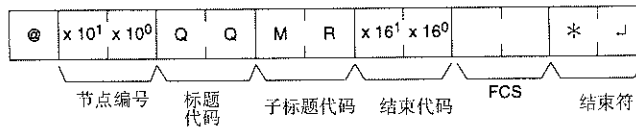
登录所有要读出的位,字和定时器/计数器的信息。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

读出区 (命令)

用四个字符的代码指定要读的区域,可指定的代码如下表所示。

读字地址,数据格式 (命令)

按照要读的数据区和数据类型,要读的信息如下表所示。“读数据”以四位数字 BCD 来表示,数据格式以二位数 BCD 表示。

| 区域 | 读数据 | 读区 | 读字 | 数据格式 |
|---------|-------|----------|-------------|--------------|
| IR 或 SR | 位 | CIO(S) | 0000 ~ 0255 | 00 ~ 15(十进制) |
| | 字 | | | “CH” |
| LR | 位 | LR(S)(S) | 0000 ~ 0063 | 00 ~ 15(十进制) |
| | 字 | | | “CH” |
| HR | 位 | HR(S)(S) | 0000 ~ 0099 | 00 ~ 15(十进制) |
| | 字 | | | “CH” |
| AR | 位 | AR(S)(S) | 0000 ~ 0027 | 00 ~ 15(十进制) |
| | 字 | | | “CH” |
| 定时器 | 完成标志 | TIM(S) | 0000 ~ 0511 | 除“CH”外的 2 字符 |
| | PV | | | “CH” |
| 高速定时器 | 完成标志 | TIMH | 0000 ~ 0511 | 除“CH”外的 2 字符 |
| | PV | | | “CH” |
| 累积定时器 | 完成标志 | TTIM | 0000 ~ 0511 | 除“CH”外的 2 字符 |
| | PV | | | “CH” |
| 计数器 | 完成标志 | CNT(S) | 0000 ~ 0511 | 除“CH”外的 2 字符 |
| | PV | | | “CH” |
| 可逆计数器 | 完成标志 | CNTR | 0000 ~ 0511 | 除“CH”外的 2 字符 |
| | PV | | | “CH” |
| DM | 字 | DM(S)(S) | 0000 ~ 6655 | 任意 2 字符 |
| EM | 当前区的字 | EM(S)(S) | 0000 ~ 6143 | 任意 2 字符 |
| | 指定区的字 | EM00 | | |

注 仅 CQM1H - CPU61 CPU 单元有 EM 区。

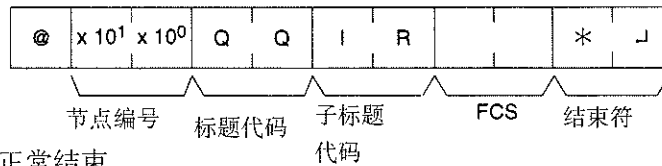
数据间隔(命令)

所指定的读出信息每一项之间用一分隔码(,)分开。能指定的最多项目数是 128。(但当指定定时器/计数器的 PV 值时,完成标志的状态也返回,因此必须作为二项计算)。

批读出

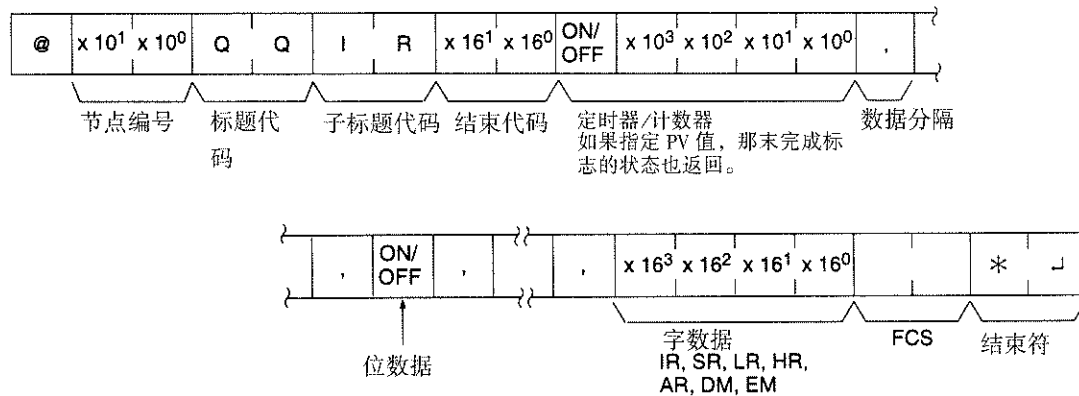
按照用 QQ 登记的读出信息,将位、字和定时器/计数器的状态作为一批读出。

命令格式



响应格式

结束代码 00 表示正常结束。



参数

读出数据(响应)

按照使用 QQ 登录读出信息的数据格式和次序返回读出数据。如果指定“完成标志”,就返回位数据(ON 或 OFF)。如果指定“字”就返回字数据。如果指定定时器/计数器的“PV”值,那么在返回 PV 值时还要跟随返回完成标志。

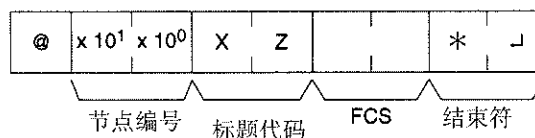
数据分隔(响应)

返回被读各分段之间的分隔代码。

6-5-35 中止 - XZ

中止当前正在进行的上位机链接操作,以便接受下一个命令,中止命令不接收响应。

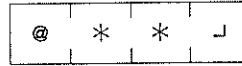
命令格式



6-5-36 初始化—* *

初始化所有与上位计算机连接的 PC 的传送控制步骤。初始化命令不使用节点编号或 FCS,并且不接收响应。

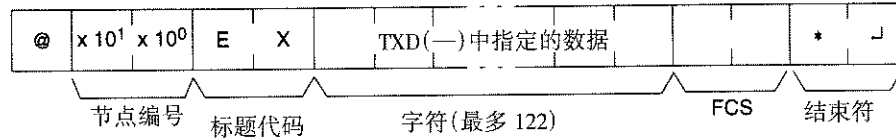
命令格式



6-5-30 TXD 响应 - EX

该响应格式在以上位机链接模式执行 PC 的 TXD(一)指令时使用。(TXD(一)把指定的数据转化成 ASCII 代码,并用该格式把代码传输到上位计算机。)

响应格式



参数

字符(响应)

该帧能容纳 122 个字符。TXD(48)不支持多帧。

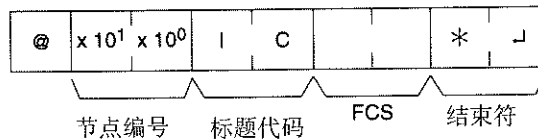
结束代码

该指令无结束代码。

6-5-38 未定义的命令—IC

如果不能译出一条命令的标题代码,就返回此响应。检查标题代码。

命令格式



第 7 章 CPU 操作和处理时间

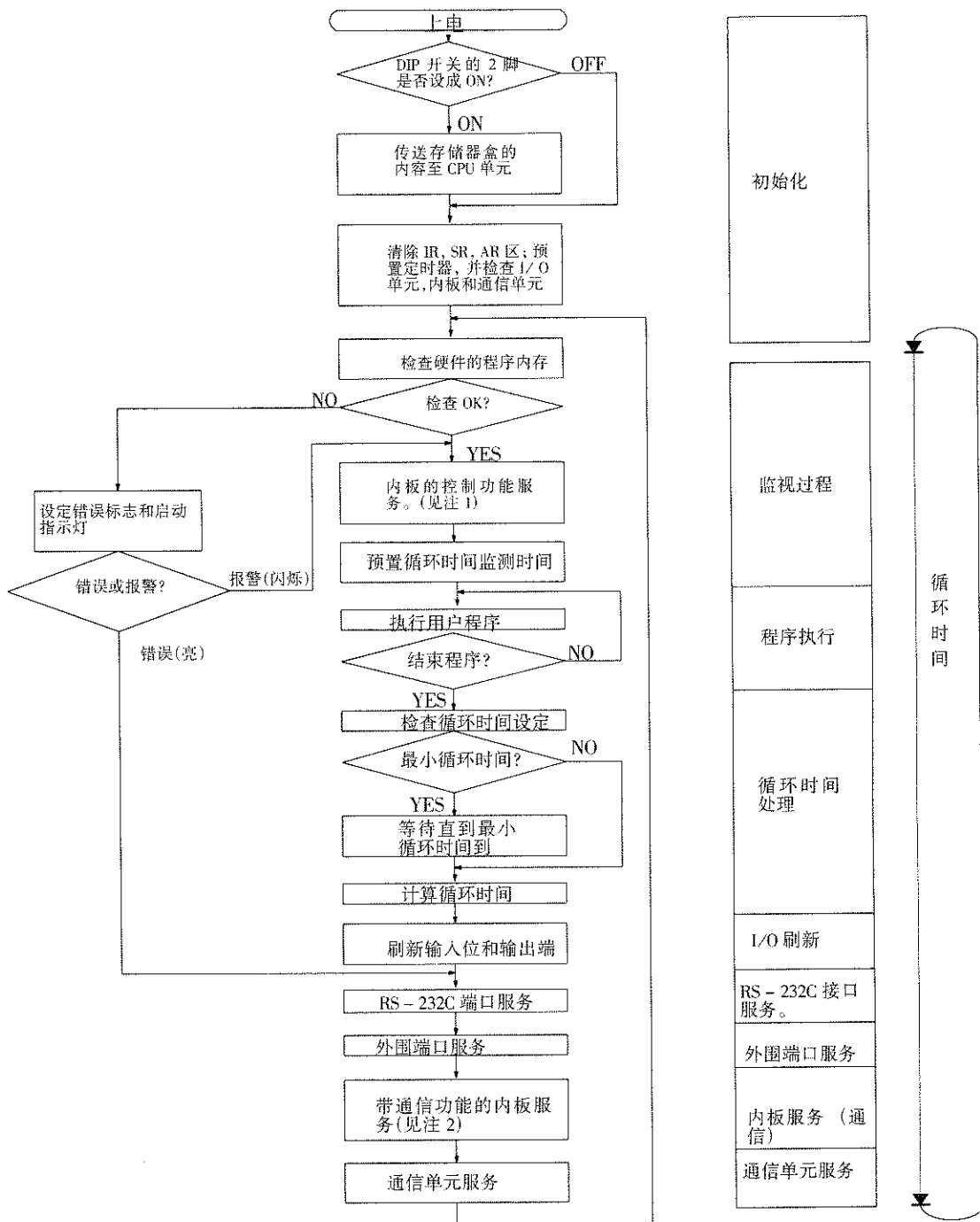
本章介绍了 CQM1H CPU 单元的内部处理及处理和执行所需要的时间。熟悉本章可以理解 CQM1H 操作的精确时序。

| | | |
|-------|----------------------|-----|
| 7-1 | CUP 单元操作 | 462 |
| 7-2 | 电源中断 | 463 |
| 7-2-1 | 电源中断操作 | 463 |
| 7-2-2 | 电源中断后的启动操作 | 464 |
| 7-3 | 循环时间 | 466 |
| 7-3-1 | 概述 | 466 |
| 7-3-2 | 指令执行时间 | 467 |
| 7-3-3 | I/O 响应时间 | 479 |
| 7-3-4 | 一对一链接 I/O 响应时间 | 480 |
| 7-3-5 | 中断处理时间 | 482 |

7-1 CPU 单元操作

操作流程图

CQM1H 操作的完整流程如下面流程图所示。执行一个 CPU 单元操作循环所需的时间称为循环时间。



- 注 1. 对包括 CPU 单元和高速计算器、脉冲 I/O、绝对值译码器接口、模拟设定、模拟量 I/O 板之间传送数据的控制功能的内板服务。
 2. 具有通信功能的内板服务,包括在 CPU 单元和串行通信板之间传送数据。

I/O 刷新方法

CQM1H I/O 刷新操作一般分成两类。第一类是输入刷新,包括将输入点的 ON/OFF 状态读到输入位。第二类是输出刷新,包括程序执行后将 ON/OFF 状态写到输出点,CQM1H I/O 刷新方法如下表所示:

| 输入/输出 | I/O 刷新方法 | 功 能 |
|-------|----------|--|
| 输 入 | 循环刷新 | 每循环在时间执行输入刷新一次。 |
| | 中断输入刷新 | 无论是输入中断、间隔定时器中断或高速计数器中断发生,在执行中断处理子程序前要执行输入刷新(循环刷新也执行)。 |
| 输 出 | 循环刷新 | 按设定时间每循环执行输出刷新一次。 |
| | 直接刷新 | 当用户程序有一个输出时,输出点立即被刷新(循环刷新也执行)。 |

I/O 刷新的缺省设定如下:

输入:仅执行循环刷新

输出:仅执行循环刷新

输入和输出必须执行循环刷新。如果在中断时间要执行输入刷新,那么在 PC 设置中要设定输入刷新范围(DM6630 至 DM6638)。在 PC 设置的 DM6639 中可以设定允许直接刷新。

除了上述方法,也可用 IORF(97)在程序中执行 I/O 刷新。

7-2 电源中断

7-2-1 电源中断操作

如果 CPU 单元断电执行。下述操作当 CPU 单元处于运行或监视模式时,若电源电压低于额定电压的 85%时执行如下操作。

1,2,3... 1. CPU 单元停止。

2. 输出单元的所有输出置 OFF。

注 所有输出置 OFF,与 I/O 保持位的状态或 PC 设置中的 I/O 保持位状态设置无关。

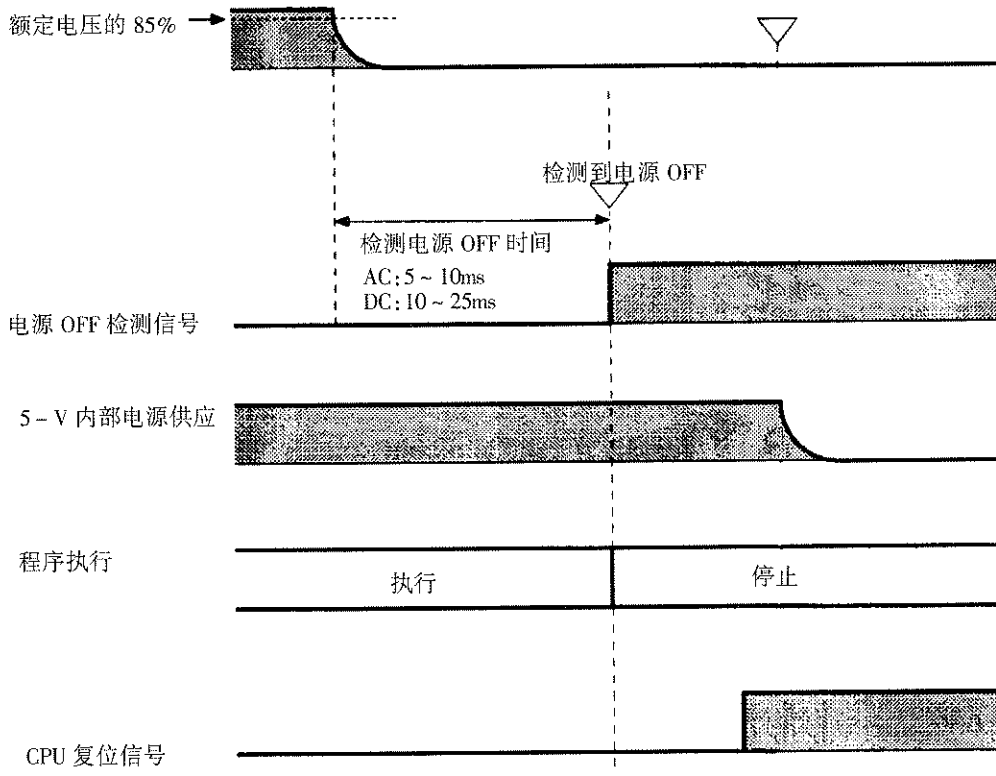
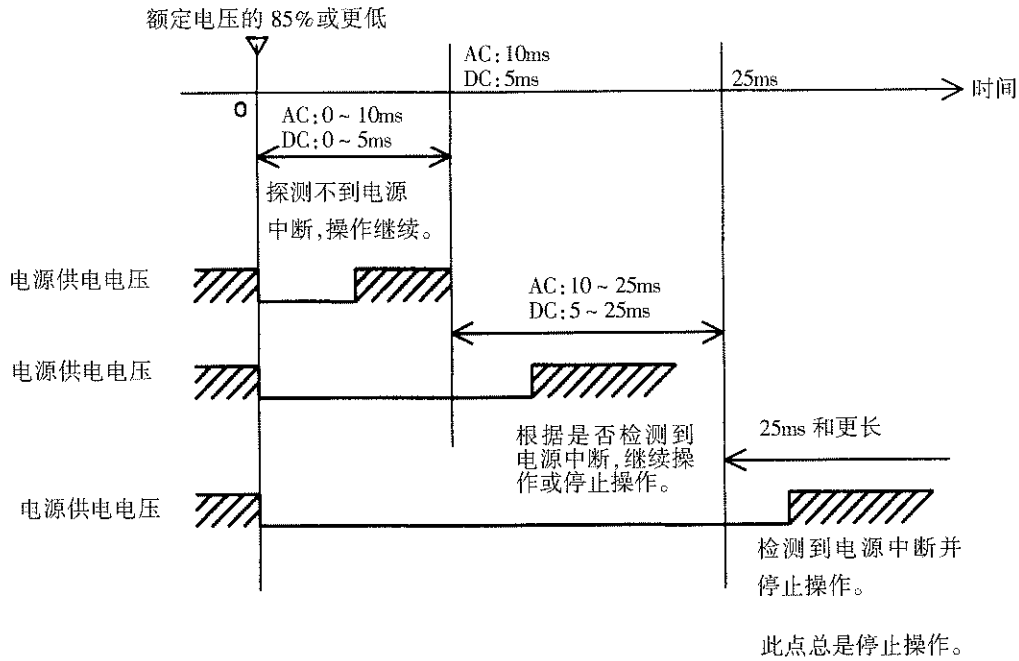
额定电压的 85%:

AC 电源:100 VAC 系统的 85V 和 200 VAC 系统的 170V

DC 电源:19.2VDC

在瞬时断电时执行如下操作。

- 1, 2, 3... 1. 如果断电持续时间在 AC 电源低于 10ms, DC 电源低于 5ms(即电压低于额定电压的 85%的时段)时系统将继续无条件运行。
2. 持续 10ms 以上 25ms 以下的 AC 电源,中断或 5ms 以上 25ms 以下的 DC 电源,中断可能被探测到也可能被忽略,即系统可能继续运行也可能停止。
3. 无论是 AC 或 DC 电源,只要电源中断超过 25ms 系统将无条件停止。



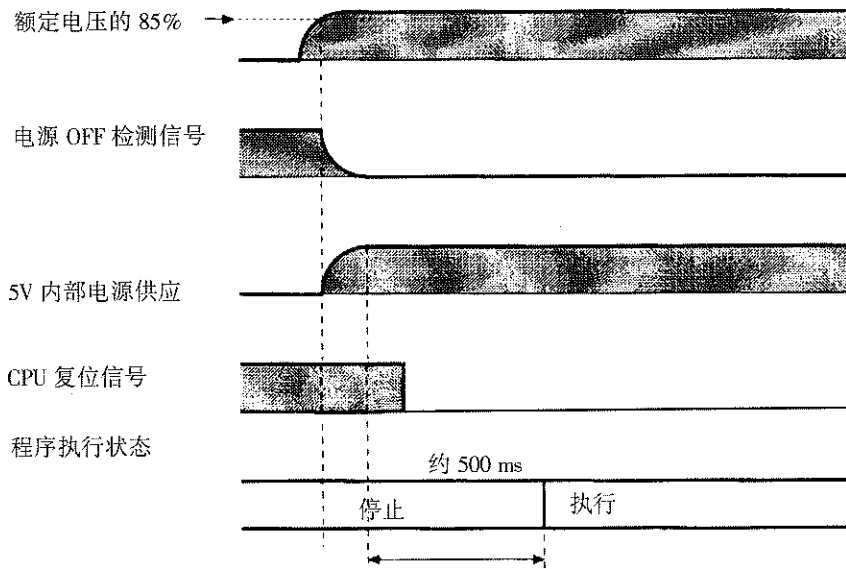
7-2-2 电源中断后的启动操作

电源中断后重新供电时, CPU 单元将按以下方式操作。重新供电后恢复操作所需的时间取决于电源供电电压、配置、周围温度、程序内容及其它条件。在下述任一情况下, CPU 单元将按运行或监视模式启动操作。

- DM6600(启动模式)为默认设定,未与外围端口相连,CPU 单元上的 DIP 开关的 7 脚为 ON。
- DM6600(启动模式)被设置为 0202Hex(运行模式)或 0201Hex(监视模式)。
- 与编程器相连且其模式选择设定为运行或监视模式。(DM6600 必须是缺省设定。)

此时操作如下:(启动操作模式详情参阅 *CQMIH 操作手册*):

- 当(AC 或 DC)电源恢复时(即电压超出额定电压的 85%),在 5V 内部电源恢复后约 500ms,CPU 单元启动操作。时间图如下所示。



7-3 循环时间

7-3-1 概述

下表所示为执行一次循环包括的过程,并说明了它们各自的处理时间。

| 过程 | 内 容 | 需 要 时 间 |
|--------------------|---|--|
| 监视 | 设置循环监视定时器, I/O 总线检查, UM 检查, 刷新时钟, 刷新 SR 区和 AR 区中的位, 具控制功能的内板服务(仅 CQM1H—CPU 61, 见注 1)等等。 | 0.7 ms(当安装了一个带时钟的存储器盒时需 0.1 ms)。每个内板(不包括串行通信板)增加 0.1 ms。若无内板, 则无需附加时间。 |
| 程序执行 | 执行用户程序。 | 执行所有指令的整个时间(根据用户程序的内容不同而不同)。 |
| 循环时间计算 | 当在 PC 设置的 DM6619 中设置了最小循环时间, 等待直到设定时间。 循环时间的计算。 | 除等待处理, 否则几乎是瞬时的。 |
| I/O 刷新 | 将输入单元的输入信息读到输入位。输出信息(执行程序的结果)写到输出单元的输出位。 | 输入字数目 × 0.01 ms 输出字数目 × 0.005 ms |
| RS—232C 端口服务 | 连接到 RS—232C 端口的设备服务。(CQM1H—CPU 11 除外。) | 循环时间的 5% 或更少(见注 3) |
| 外围端口服务 | 连接到外围端口的设备的服务。 | 循环时间的 5% 或更少(见注 3) |
| 具有通信功能的内板服务(见注 2)。 | 安装了串行通信板时, 处理板发出的命令(仅用于 CQM1H—CPU 51/61)。 | 0.4 ms + 每个端口的处理时间 每个端口的处理时间最小值为 0.256 或 0.05 × 循环时间。 若未安装串行通信板, 时间为 0 ms。 |
| 通信单元服务 | 安装了 Controller Link 单元时, 处理板发出的命令(仅用于 CQM1H—CPU 51/61)。 | 对于 CQM1H—CLK21, 最大为 4 ms。若未安装通信单元, 时间为 0 ms。 |

- 注 1. 包括在 CPU 单元和高速计数器, 脉冲 I/O, 绝对值编码器接口, 模拟设定和模拟量 I/O 板之间传送数据的内板控制功能服务。
2. 具有通信功能的内板服务, 包括在 CPU 单元和串行通信板之间传送数据。
3. 可在 PC 设置中改变百分比 (DM6616: 端口 RS—232C 的服务时间; DM6617: 外围端口的服务时间)。使用 RS—232C 端口、外围端口或串行通信板端口 1 或 2 时, 每个端口的最小时间为 0.256ms。

循环时间和操作

循环时间对 CPU 单元操作的影响如下所示。

| 循环时间 | 操 作 条 件 |
|-----------|--|
| 10 ms或更长 | 使用 TC016 ~ TC511 时, TIMH(15)可能不精确(对于 TC000 ~ TC015 操作正常)(见注 1)。 |
| 20 ms或更长 | 编程时使用 0.02 秒时钟位(SR25401)可能不精确。 |
| 100 ms或更长 | 编程时使用的 0.1 秒时钟位(SR25500)可能不精确。产生循环时间超出错误(SR25309 会置 ON)(见注 2)。定时器(TIM)和累计定时器(TTIM)指令可能不精确。 |
| 120 ms或更长 | FALS 9F 监测时间 SV 超过。产生系统错误(FALS 9F)并且操作停止(见注 3)。 |
| 200 ms或更长 | 编程时使用的 0.2 秒时钟位(SR25501)可能不精确。 |

- 注 1. 在 PC 设置的 DM6629 中可以设定使用中断处理的定时器数目。缺省设置

为 TC000 ~ TC015。

2. PC 设置(DM6655)可以用于禁止检测循环时间超出错误。

3. 用 PC 设置(DM6618)可以改变 FALS 9F 循环监测时间。

循环时间例

本例为计算带 80 点 I/O 的 CQM1H 的循环时间。I/O 如下配置：

DC 输入： 48 点(3 个字)

位输出： 32 点(2 个字)

其他操作条件假定如下：

用户程序 2000 条指令(由 LD 和 OUT 指令组成)

内板 串行通信板和高速计数器板。

通信单元 无 Controller Link 单元

时钟： 没有

RS-232 端口：使用

循环时间： 可变的(无最小设定)

注 在用户程序中的单个指令的平均处理时间假定为 $0.625\mu\text{s}$ 。

循环时间如下表所示。

| 过程 | 计算方法 | 带外围设备时需要的时间 | 不带外围设备时需要的时间 |
|--------------|--|-------------|--------------|
| 监视 | 固定的 | 0.8 ms | 0.8 ms |
| 程序执行 | $0.625 \times 2000 \mu\text{s}$ | 1.25 ms | 1.25 ms |
| 循环时间计算 | 可忽略 | 0 ms | 0 ms |
| I/O 刷新 | $0.01 \times 3 + 0.0005 \times 2(\mu\text{s})$ | 0.04 ms | 0.04 ms |
| RS-232C 端口服务 | | 0 ms | 0 ms |
| 外围端口服务 | 最小时间 | 0.34 ms | 0 ms |
| 串行通信板服务 | $0.4 + 0.26(\text{ms})$ | 0.66 ms | 0.66 ms |
| 通信单元服务 | 0 ms | 0 ms | 0 ms |
| 循环时间 | (1) + (2) + (3) + (4) + (5) + (6) | 3.27 ms | 3.01 ms |

- 注
1. 通过外围设备可以自动从 PC 读取循环时间。
 2. 最大的和当前的循环时间存贮在 AR26 和 AR27。
 3. 循环时间随实际的操作条件而变化,也不一定与计算值完全一致。
 4. RS-232 和外围端口服务时间最小 0.256ms,最大为 65.536ms。

7-3-2 指令执行时间

下表列出了 CQM1H 指令的执行时间。最大和最小的执行时间及引起它们的条件在相关处给出。条件栏列中有关的“字”指除了间接寻址 DM 字外任何字的内容。使用时需要较长执行时间的间接寻址 DM 字表示为“*DM”。

大多数指令的执行时间取决于它们执行的是 ON 执行条件还是 OFF 执行条件。例外的是梯形图指令 OUT 和 OUT NOT,不管什么操作条件都需要相同的时间。一个指令的 OFF 执行时间也可根据情况而变化,即:该指令是否在一个互锁程序部分并且 IL 的执行条件为 OFF,该指令是否在 JMP(04) 和 JMP(05)之间并且 JMP(04)执行条件为 OFF,或者该指令是否由 OFF 执行条件来复位。

“REST”、“IL”和“JMP”是用来表示这三种时间。

基本指令

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) | | |
|-----|---------|-------------|---|--------------|-------|-------|
| | | | | RSET | IL | JMP |
| ... | LD | 0.375 | 任何条件 | ... | | |
| ... | LD NOT | | | | | |
| ... | AND | | | | | |
| ... | AND NOT | | | | | |
| ... | OR | | | | | |
| ... | OR NOT | | | | | |
| ... | AND LD | | | | | |
| ... | OR LD | | | | | |
| ... | OUT | 0.563 | 没有直接输出,或用直接输出时 操作数不是IR10000 ~ IR11515 的其他数 | ... | | |
| ... | OUT NOT | | | | | |
| ... | SET | 0.938 | 直接输出 | ... | | |
| ... | RSET | | | | | |
| ... | TIM | 1.125 | SV 为常数 | 1.125 | 1.125 | 1.125 |
| | | | SV 为 * DM | 40.8 | 1.125 | 1.125 |
| ... | CNT | 1.125 | SV 为常数 | 1.125 | 1.125 | 1.125 |
| | | | SV 为 * DM | 38.7 | 1.125 | 1.125 |

特殊指令

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) | | |
|----|------|-------------|---|--------------|------|------|
| 00 | NOP | 0.375 | 任何条件 | ... | | |
| 01 | END | 28.0 | | ... | | |
| 02 | IL | 9.3 | | 8.2 | | |
| 03 | ILC | 8.5 | | 8.5 | | |
| 04 | JMP | 13.8 | | 8.9 | | |
| 05 | JME | 8.3 | | 8.3 | | |
| 06 | FAL | 42.6 | | 1.125 | | |
| 07 | FALS | 3.0 | | 1.125 | | |
| 08 | STEP | 43.7 | | 1.125 | | |
| 09 | SNXT | 18.8 | | 1.125 | | |
| 10 | SFT | 33.2 | 1 字移位寄存器 | 复位 | IL | JMP |
| | | 58.3 | 10 字移位寄存器 | 32.4 | 11.5 | 11.5 |
| | | 311.4 | 100 字移位寄存器 | 52.0 | 11.5 | 11.5 |
| 11 | KEEP | 0.563 | 没有直接输出,或用直接输出时 操作数不是IR10000 ~ IR11515。 | ... | | |
| | | 0.938 | 使用IR10000 ~ IR11515 的直接输出 | ... | | |
| 12 | CNTR | 39.8 | SV 为常数 | 复位 | IL | JMP |
| | | 59.7 | SV 为 * DM | 0.25 | 15.5 | 15.5 |
| 13 | DIFU | 16.2 | 任何条件 | 正常 | IL | JMP |
| | | | | 15.8 | 15.5 | 13.4 |
| 14 | DIFU | 15.6 | 任何条件 | 正常 | IL | JMP |
| | | | | 15.6 | 15.5 | 13.3 |

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) | | |
|----|------|-------------|-----------------------|--------------|------|------|
| | | | | 复位 | IL | JMP |
| 15 | TIMH | 27.4 | SV 为常数 | 41.2 | 40.0 | 20.8 |
| | | 27.4 | SV 为 * DM | 60.6 | 59.4 | 20.8 |
| | | | | | | |
| 16 | WSFT | 33.6 | 1 字移位寄存器 | 1.5 | | |
| | | 57.8 | 10 字移位寄存器 | | | |
| | | 1.7ms | 使用 * DM 的 1024 字移位寄存器 | | | |
| | | 9.8ms | 使用 * DM 的 6144 字移位寄存器 | | | |
| 20 | CMP | 20.1 | 常数和字比较时 | 1.5 | | |
| | | 22.2 | 两个字比较时 | | | |
| | | 58.0 | 两个 * DM 比较时 | | | |
| 21 | MOV | 17.7 | 常数传送到字时 | 1.5 | | |
| | | 19.8 | 一字传送到另一字时 | | | |
| | | 54.6 | * DM 到 * DM 传送时 | | | |
| 22 | MVN | 17.8 | 常数传送到字时 | 1.5 | | |
| | | 19.9 | 一字传送到另一字时 | | | |
| | | 54.5 | * DM 到 * DM 传送时 | | | |
| 23 | BIN | 37.8 | 字变换到字时 | 1.5 | | |
| | | 72.0 | * DM 变换到 * DM 时 | | | |
| 24 | BCD | 35.8 | 字变换到字时 | 1.5 | | |
| | | 70.0 | * DM 变换到 * DM 时 | | | |
| 25 | ASL | 18.0 | 移位字时 | 1.125 | | |
| | | 34.4 | 移位 * DM 时 | | | |
| 26 | ASR | 18.0 | 移位字时 | 1.125 | | |
| | | 34.4 | 移位 * DM 时 | | | |
| 27 | ROL | 18.6 | 旋转移位字时 | 1.125 | | |
| | | 35.0 | 旋转位 * DM 时 | | | |
| 28 | ROR | 18.6 | 旋转移位字时 | 1.125 | | |
| | | 35.0 | 旋转移位 * DM 时 | | | |
| 29 | COM | 19.5 | 字取反时 | 1.125 | | |
| | | 36.3 | * DM 取反时 | | | |
| 30 | ADD | 37.5 | 常数 + 字 → 字 | 1.875 | | |
| | | 39.9 | 字 + 字 → 字 | | | |
| | | 91.6 | * DM + * DM → * DM | | | |
| 31 | SUB | 37.5 | 常数 - 字 → 字 | 1.875 | | |
| | | 39.8 | 字 - 字 → 字 | | | |
| | | 91.6 | * DM - * DM → * DM | | | |
| 32 | MUL | 55.3 | 常数 × 字 → 字 | 1.875 | | |
| | | 57.8 | 字 × 字 → 字 | | | |
| | | 108.4 | * DM × * DM → * DM | | | |
| 33 | DIV | 54.2 | 字 ÷ 常数 → 字 | 1.875 | | |
| | | 56.6 | 字 ÷ 字 → 字 | | | |
| | | 107.3 | * DM ÷ * DM → * DM | | | |
| 34 | ANDW | 31.5 | 常数 ∧ 字 → 字 | 1.875 | | |
| | | 33.9 | 字 ∧ 字 → 字 | | | |
| | | 85.6 | * DM ∧ * DM → * DM | | | |
| 35 | ORW | 31.5 | 常数 ∨ 字 → 字 | 1.875 | | |
| | | 33.9 | 字 ∨ 字 → 字 | | | |
| | | 85.6 | * DM ∨ * DM → * DM | | | |

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) |
|----|------|-------------|--------------------|--------------|
| 36 | XORW | 31.5 | 常数∨字→字 | 1.875 |
| | | 33.9 | 字∨字→字 | |
| | | 85.6 | * DM ∨ * DM → * DM | |
| 37 | XNRW | 31.5 | 常数∩字→字 | 1.875 |
| | | 33.9 | 字∩字→字 | |
| | | 85.6 | * DM ∩ * DM → * DM | |
| 38 | INC | 20.9 | 字递增时 | 1.125 |
| | | 37.6 | * DM 递增时 | |
| 39 | DEC | 21.3 | 字递减时 | 1.125 |
| | | 38.1 | * DM 递增时 | |
| 40 | STC | 9.0 | 任何条件 | 0.75 |
| 41 | CLC | 9.0 | | 0.75 |
| 45 | TRSM | 21.6 | | 0.75 |
| 46 | MSC | 18.5 | 用字信息时 | 1.125 |
| | | 36.3 | 用 * DM 信息时 | |
| 50 | ADB | 40.1 | 常数+字→字 | 1.875 |
| | | 42.5 | 字+字→字 | |
| | | 94.2 | * DM + * DM → * DM | |
| 51 | SBB | 40.1 | 常数-字→字 | 1.875 |
| | | 42.5 | 字-字→字 | |
| | | 94.2 | * DM - * DM → * DM | |
| 52 | MLB | 34.3 | 常数×字→字 | 1.875 |
| | | 36.7 | 字×字→字 | |
| | | 87.3 | * DM × * DM → * DM | |
| 53 | DVB | 35.1 | 字÷常数→字 | 1.875 |
| | | 37.5 | 字÷字→字 | |
| | | 88.1 | * DM ÷ * DM → * DM | |
| 54 | ADDL | 44.5 | 字+字→字 | 1.875 |
| | | 96.7 | * DM + * DM → * DM | |
| 55 | SUBL | 44.5 | 字-字→字 | 1.875 |
| | | 96.7 | * DM - * DM → * DM | |
| 56 | MULL | 153.4 | 字×字→字 | 1.875 |
| | | 203.4 | * DM × * DM → * DM | |
| 57 | DIVL | 154.5 | 字÷字→字 | 1.875 |
| | | 204.5 | * DM ÷ * DM → * DM | |
| 58 | BINL | 57.0 | 字→字 | 1.5 |
| | | 90.5 | * DM → * DM | |
| 59 | BCDL | 45.7 | 字→字 | 1.5 |
| | | 79.2 | * DM → * DM | |
| 70 | XFER | 54.7 | 常数传送到字时 | 1.875 |
| | | 57.1 | 字传送到字时 | |
| | | 2.2ms | 用 * DM 传送 1,024 字时 | |
| | | 12.5ms | 用 * DM 传送 6,144 字时 | |
| 71 | BSET | 34.2 | 1 字设成常数时 | 1.875 |
| | | 58.5 | 10 字设成常数时 | |
| | | 1.47ms | 1024 字设成 * DM 数时 | |
| | | 8.22ms | 6144 字设成 * DM 数时 | |
| 72 | ROOT | 48.0 | 字运算→字 | 1.5 |
| | | 83.1 | * DM 运算→ * DM | |
| 73 | XCHC | 30.7 | 字→字 | 1.5 |
| | | 64.2 | * DM → * DM | |
| 74 | SLD | 30.9 | 移位 1 个字 | 1.5 |
| | | 76.5 | 移位 10 个字 | |
| | | 4.12ms | 使用 * DM 移位 1024 字 | |
| | | 24.44ms | 使用 * DM 移位 6144 字 | |

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) |
|----|------|-------------|-----------------------------------|--------------|
| 75 | SRD | 30.9 | 移位 1 个字 | 1.5 |
| | | 76.5 | 移位 10 个字 | |
| | | 4.12ms | 使用 * DM 移位 1,024 字 | |
| | | 24.44ms | 使用 * DM 移位 6,144 字 | |
| 76 | MLPX | 44.4 | 字到字译码时 | 1.875 |
| | | 102.3 | * DM 到 * DM 译码时 | |
| 77 | DMPX | 33.9 | 字到字编码时 | 1.875 |
| | | 90.5 | * DM 到 * DM 编码时 | |
| 78 | SDEC | 45.5 | 字到字译码时 | 1.875 |
| | | 103.9 | * DM 到 * DM 译码时 | |
| 80 | DIST | 49.5 | 把一个常数设到字 + 字时 | 1.875 |
| | | 52.0 | 把一个字设到字 + 字时 | |
| | | 108.3 | 把 * DM 设到 * DM + * DM 时 | |
| | | 75.8 | 把一个常数设到堆栈时 | |
| | | 78.3 | 把一个字设到堆栈时 | |
| | | 133.4 | 通过 * DM 把 * DM 设到堆栈时 | |
| 81 | COLL | 48.9 | 把一个常数 + 字设到字时 | 1.875 |
| | | 51.3 | 把字 + 字设到字时 | |
| | | 105.1 | 把 * DM + * DM 设到 * DM 时 | |
| | | 45.9 | 把字 + 常数设到 FIFO 堆栈时 | |
| | | 48.3 | 把字 + 字设到 FIFO 堆栈时 | |
| | | 103.2 | 通过 * DM 把 * DM + * DM 设到 FIFO 堆栈时 | |
| | | 45.3 | 把字 + 常数设到 LIFO 堆栈 | |
| | | 47.7 | 把字 + 字设到 LIFO 堆栈时 | |
| 82 | MOVB | 34.8 | 移常数到字 | 1.875 |
| | | 41.2 | 移字到字时 | |
| | | 93.9 | 移 * DM 到 * DM 时 | |
| 83 | MOVD | 30.6 | 移常数到字时 | 1.875 |
| | | 36.9 | 移字到字时 | |
| | | 89.6 | 移 * DM 到 * DM 时 | |
| 84 | SFTR | 43.1 | 移位 1 个字 | 1.875 |
| | | 73.8 | 移位 10 个字 | |
| | | 1.7ms | 使用 * DM 移位 1024 字 | |
| | | 9.68ms | 使用 * DM 移位 6144 字 | |
| 85 | TCMP | 71.9 | 常数和字设置表比较 | 1.875 |
| | | 74.1 | 字和字设置表比较 | |
| | | 126.8 | * DM 与 * DM 设置表比较 | |
| 86 | ASC | 46.9 | 字 → 字 | 1.875 |
| | | 108.3 | * DM → * DM | |
| 90 | SEND | 65.6 | 字 | 1.875 |
| | | 121.4 | * DM | |
| 91 | SBS | 31.1 | 任何条件 | 1.125 |
| 92 | SBN | ... | | ... |
| 93 | RET | 29.3 | | 1.125 |

| 代码 | 助记符 | ON 执行时间(μs) | 条件(上:最小;下:最大) | OFF 执行时间(μs) |
|----|------|-------------|-------------------|--------------|
| 97 | IORF | 29.1 | 刷新 IR000 | 1.5 |
| | | 35.0 | 刷新一个输入字 | |
| | | 39.0 | 刷新一个输出字 | |
| | | 93.3 | 刷新 8 个 I/O 字 | |
| 98 | RECV | 78.4 | 字 | 1.875 |
| | | 132.4 | * DM | |
| 99 | MCRO | 105.2 | 用字设置 I/O 操作数 | 1.875 |
| | | 141.1 | 用 * DM 设置 I/O 操作数 | |

扩展指令

| 代 码 | 助 记 符 | ON执行时间(μs) | 条 件 | OFF 执行时间(μs) |
|-----|-------|------------|---------------------------|--------------|
| 17 | ASFT | 47.1 | 移位1字 | 1.875 |
| | | 72.6 | 移位10字 | |
| | | 1.85ms | 通过 * DM移位1,024字 | |
| | | 12.3ms | 通过 * DM移位6,144字 | |
| 18 | TKY | 60.9 | 字到字 | 1.875 |
| | | 99.0 | * DM到 * DM | |
| 19 | MCMP | 93.0 | 比较字 | 1.875 |
| | | 146.5 | 比较 * DM | |
| 47 | RXD | 92.4 | 通过字输入1字节 | 1.875 |
| | | 635.5 | 通过 * DM输入256字节 | |
| 48 | TXD | 78.9 | 通过字输出1字节(RS - 232C) | 1.875 |
| | | 624.3 | 通过 * DM输出256字节(RS - 232C) | |
| | | 64.7 | 通过字输出1字节(上位机链接) | |
| | | 106.4 | 通过 * DM输出256字节(上位机链接) | |
| 60 | CMPL | 38.2 | 比较字 | 1.875 |
| | | 75.8 | 比较 * DM | |

| 代码 | 助记符 | ON执行时间(μs) | 条件 | OFF执行时间(μs) |
|----|-----|------------------------------|-------------|-------------|
| 61 | INI | 内置高速计数器0或输出位的脉冲输出: | | 1.875 |
| | | 81.6 | 通过字启动比较 | |
| | | 103.0 | 通过*DM启动比较 | |
| | | 64.9 | 通过字停止比较 | |
| | | 74.7 | 通过*DM停止比较 | |
| | | 147.3 | 通过字改变PV | |
| | | 164.0 | 通过*DM改变PV | |
| | | 50.8 | 通过字停止脉冲输出 | |
| | | 72.2 | 通过*DM停止脉冲输出 | |
| | | 高速计数器板的1~4高速计数器 | | |
| | | 94.0 | 通过字启动比较 | |
| | | 112.0 | 通过*DM启动比较 | |
| | | 94.0 | 通过字停止比较 | |
| | | 112.0 | 通过*DM停止比较 | |
| | | 136.0 | 通过字改变PV | |
| | | 154.0 | 通过*DM改变PV | |
| | | 脉冲I/O板上高速计数器端口1和2或输出脉冲的端口1和2 | | |
| | | 267.2 | 通过字启动比较 | |
| | | 291.9 | 通过*DM启动比较 | |
| | | 186.6 | 通过字停止比较 | |
| | | 209.6 | 通过*DM停止比较 | |
| | | 421.5 | 通过字改变PV | |
| | | 439.1 | 通过*DM改变PV | |
| | | 223.9 | 通过字停止脉冲输出 | |
| | | 242.9 | 通过*DM停止脉冲输出 | |
| | | 绝对值编码器接口板上的高速计数器1和2。 | | |
| | | 266.7 | 通过字启动比较 | |
| | | 285.1 | 通过*DM启动比较 | |
| | | 182.1 | 通过字停止比较 | |
| | | 203.7 | 通过*DM停止比较 | |

| 代码 | 助记符 | ON 执行时间(μs) | 条 件 | OFF 执行时间(μs) | | | |
|------------------|---------------------|------------------------------|-----------------------|--------------|----------------------|---------------|-------|
| 62 | PRV | 内置高速计数器 0 或输出位的脉冲输出: | | 1.875 | | | |
| | | 82.4 | 通过字指定输出 | | | | |
| | | 105.7 | 通过 * DM 指定输出 | | | | |
| | | 高速计数器板上高速计数器 1~4。 | | | | | |
| | | 115.0 | 通过字指定输出(读状态) | | | | |
| | | 132.0 | 通过 * DM 指定输出(读状态) | | | | |
| | | 124.0 | 通过字指定输出(读 PV) | | | | |
| | | 142.0 | 通过 * DM 指定输出(读 PV) | | | | |
| | | 脉冲 I/O 板上的高速计数器脉冲输出的端口 1 和 2 | | | | | |
| | | 206.4 | 通过字指定输出(读状态) | | | | |
| | | 224.4 | 通过 * DM 指定输出(读状态) | | | | |
| | | 206.9 | 通过字指定输出(读区域比较结果) | | | | |
| | | 230.7 | 通过 * DM 指定输出(读区域比较结果) | | | | |
| | | 绝对编码器接口板的高速计数器 1 和 2: | | | | | |
| | | 203.7 | 通过字指定输出(读状态) | | | | |
| | | 228.0 | 通过 * DM 指定输出(读状态) | | | | |
| | | 205.0 | 通过字指定输出(读区域比较结果) | | | | |
| | | 228.0 | 通过 * DM 指定输出(读区域比较结果) | | | | |
| | | 63 | CTBL | | 内置高速计数器 0 或输出位的脉冲输出: | | 1.875 |
| | | | | | 189.3 | 1 个字目标的目标表并启动 | |
| 210.5 | 1 个 * DM 目标的目标表并启动 | | | | | | |
| 1.18ms | 16 个字目标的目标表并启动 | | | | | | |
| 1.20ms | 16 个 * DM 目标的目标表并启动 | | | | | | |
| 1.13ms | 字区域表并启动 | | | | | | |
| 1.14ms | * DM 区域表并启动 | | | | | | |
| 153.8 | 1 个字目标的目标表 | | | | | | |
| 174.9 | 1 个 * DM 目标的目标表 | | | | | | |
| 1.14ms | 16 字目标的目标表 | | | | | | |
| 1.18ms | 16 个字 * DM 目标的目标表 | | | | | | |
| 981.0 | 字区域表 | | | | | | |
| 999.0 | * DM 区域表 | | | | | | |
| 高速计数器板的高速计数器 1~4 | | | | | | | |
| 152.0 | 1 个字目标的目标表并启动 | | | | | | |
| 168.0 | 1 个 * DM 目标的目标表并启动 | | | | | | |
| 1.05ms | 48 个字目标的目标表并启动 | | | | | | |
| 1.07ms | 48 个 * DM 目标的目标表并启动 | | | | | | |
| 718.0 | 字区域表并启动 | | | | | | |
| 735.0 | * DM 区域表并启动 | | | | | | |
| 152.0 | 1 个字目标的目标表 | | | | | | |
| 168.0 | 1 个 * DM 目标的目标表 | | | | | | |
| 1.05ms | 48 个字目标的目标表 | | | | | | |
| 1.07ms | 16/48 个 * DM 目标的目标表 | | | | | | |
| 718.0 | 字区域表 | | | | | | |
| 735.0 | * DM 区域表 | | | | | | |

| 代码 | 助记符 | ON 执行时间(μs) | 条 件 | OFF 执行时间(μs) | | |
|----|------|--------------------------------------|------------------------|--------------|--|--|
| | CTBL | 脉冲 I/O 板的高速计数器 1 和 2 或脉冲输出的端口 1 和 2: | | 1.875 | | |
| | | 623.6 | 1 个字目标的目标表并启动 | | | |
| | | 649.3 | 1 个 * DM 目标的目标表并启动 | | | |
| | | 7.06/7.84ms | 16/48 个字目标的目标表并启动 | | | |
| | | 7.07ms | 16/48 个 * DM 目标的目标表并启动 | | | |
| | | 2.03ms | 字区域表并启动 | | | |
| | | 2.05ms | * DM 区域表并启动 | | | |
| | | 440.0 | 1 个字目标的目标表 | | | |
| | | 466.1 | 1 个 * DM 目标的目标表 | | | |
| | | 6.90ms | 16/48 个字目标的目标表 | | | |
| | | 6.95ms | 16/48 个 * DM 目标的目标表 | | | |
| | | 1.98ms | 字区域表 | | | |
| | | 1.99ms | * DM 区域表 | | | |
| | | 绝对值编码器接口板上的高速计数器 1 和 2: | | | | |
| | | 540.8 | 1 个字目标的目标表并启动 | | | |
| | | 562.4 | 1 个 * DM 目标的目标表并启动 | | | |
| | | 5.84ms | 48 个字目标的目标表并启动 | | | |
| | | 5.92ms | 48 个 * DM 目标的目标表并启动 | | | |
| | | 1.32ms | 字区域表并启动 | | | |
| | | 1.35ms | * DM 区域表并启动 | | | |
| | | 414.8 | 1 个字目标的目标表 | | | |
| | | 436.4 | 1 个 * DM 目标的目标表 | | | |
| | | 5.40ms | 48 个字目标的目标表 | | | |
| | | 5.42ms | 48 个 * DM 目标的目标表 | | | |
| | | 1.31ms | 字区域表 | | | |
| | | 1.33ms | * DM 区域表 | | | |
| 64 | SPED | CPU 单元的输出位的脉冲输出: | | 1.875 | | |
| | | 106.6 | 由常数确定频率 | | | |
| | | 110.9 | 由字确定频率 | | | |
| | | 132.2 | 由 * DM 确定频率 | | | |
| | | 脉冲 I/O 板的端口 1 和 2 的脉冲输出: | | | | |
| | | 272.1 | 由常数确定频率 | | | |
| | | 279.3 | 由字确定频率 | | | |
| | | 288.3 | 由 * DM 确定频率 | | | |
| 65 | PULS | CPU 单元的输出位和脉冲输出: | | 1.875 | | |
| | | 98.1 | 由字确定脉冲数目 | | | |
| | | 124.1 | 由 * DM 确定脉冲数目 | | | |
| | | 脉冲 I/O 板的端口 1 和 2 的脉冲输出: | | | | |
| | | 303.6 | 由字确定脉冲数目 | | | |
| | | 324.3 | 由 * DM 确定脉冲数目 | | | |
| 66 | SCL | 79.4 | 字指定 | 1.875 | | |
| | | 135.4 | * DM 指定 | | | |
| 67 | BCNT | 66.3 | 计数一个字 | 1.875 | | |
| | | 36.99ms | 通过 * DM 计数 6,656 字 | | | |
| 68 | BCMP | 105.0 | 比较常数,结果放入字 | 1.875 | | |
| | | 107.3 | 比较字,结果放入字 | | | |
| | | 146.1 | 比较 * DM,结果放入 * DM | | | |

| 代码 | 助记符 | ON 执行时间(μs) | 条 件 | OFF 执行时间(μs) |
|------|------------------|--------------------------|--------------------|---------------------------|
| 69 | STIM | 27.6 | 字设定单发中断启动 | 1.875 |
| | | 55.4 | * DM 设定单发中断启动 | |
| | | 28.0 | 字设定定时中断启动 | |
| | | 55.8 | * DM 设定定时中断启动 | |
| | | 49.8 | 字设定定时器读 | |
| | | 85.2 | * DM 设定定时器读 | |
| | | 26.5 | 字设定定时器停止 | |
| | | 26.7 | * DM 设定定时器停止 | |
| 87 | DSW | 52.8 | 字设定 4 位数 CS 输出 | 1.875 |
| | | 52.8 | 字设定 4 位数 RD 输出 | |
| | | 66.9 | 字设定 4 位数数据输入 | |
| | | 69.9 | * DM 设定 4 位数 CS 输出 | |
| | | 69.9 | * DM 设定 4 位数 RD 输出 | |
| | | 82.8 | * DM 设定 4 位数数据输入 | |
| | | 56.1 | 字设定 8 位数 CS 输出 | |
| | | 56.4 | 字设定 8 位数 RD 输出 | |
| | | 79.2 | 字设定 8 位数数据输入 | |
| | | 77.7 | * DM 设定 8 位数 CS 输出 | |
| | | 78.0 | * DM 设定 8 位数 RD 输出 | |
| 98.7 | * DM 设定 8 位数数据输入 | | | |
| 88 | 7SEG | 59.1 | 4 位数, 字指定 | 1.875 |
| | | 77.0 | 4 位数, * DM 指定 | |
| | | 69.1 | 8 位数, 字指定 | |
| | | 87.9 | 8 位数, * DM 指定 | |
| 89 | INT | 39.8 | 通过字设定屏蔽 | 1.875 |
| | | 60.6 | 通过 * DM 设定屏蔽 | |
| | | 37.5 | 通过字清除中断 | |
| | | 54.9 | 通过 * DM 清除中断 | |
| | | 38.1 | 通过字读屏蔽状态 | |
| | | 54.0 | 通过 * DM 读屏蔽状态 | |
| | | 48.6 | 通过字改变计数器 SV | |
| | | 66.1 | 通过 * DM 改变计数器 SV | |
| | | 20.7 | 通过字屏蔽所有中断 | |
| | | 20.7 | 通过 * DM 屏蔽所有中断 | |
| | | 21.4 | 通过字清除所有中断 | |
| | | 21.4 | 通过 * DM 清除所有中断 | |
| — | ACC | 413.2 | 模式 0: 字作为控制字 | 1.875 |
| | | 435.5 | 模式 0: * DM 作为控制字 | |
| | | 297.3 | 模式 1: 字作为控制字 | |
| | | 320.7 | 模式 1: * DM 作为控制字 | |
| | | 306.3 | 模式 2: 字作为控制字 | |
| | | 325.5 | 模式 2: * DM 作为控制字 | |
| | | 197.8 | 模式 3: 字作为控制字 | |
| | | 316.5 | 模式 3: * DM 作为控制字 | |

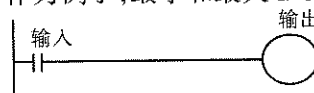
| 代码 | 助记符 | ON 执行时间(μs) | 条 件 | OFF 执行时间(μs) |
|----|------|-------------|--------------------------|--------------|
| — | ACOS | 1.15ms | 字→字 | 1.875 |
| | | 1.18ms | * DM→* DM | |
| — | ADBL | 59.3 | 字+字→字 | 1.875 |
| | | 116.7 | * DM+* DM→* DM | |
| — | APR | 45.8 | 计算正弦 | 1.875 |
| | | 348.0 | 通过* DM 指定的 256 项表的线性逼近 | |
| — | ASIN | 1.10ms | 字→字 | 1.875 |
| | | 1.13ms | * DM→* DM | |
| — | ATAN | 536.0 | 字→字 | 1.875 |
| | | 572.0 | * DM→* DM | |
| — | AVG | 58.0 | 字的一次循环平均数 | 1.875 |
| | | 214.6 | 通过* DM 的 64 次循环平均数 | |
| — | CMND | 74.2 | 字 | 1.875 |
| | | 128.4 | * DM | |
| — | COLM | 89.1 | 字→字 | 1.875 |
| | | 140.1 | * DM→* DM | |
| — | COS | 7660 | 字→字 | 1.875 |
| | | 800.0 | * DM→* DM | |
| — | CPS | 26.0 | 比较常数和字 | 1.875 |
| | | 28.0 | 比较字 | |
| | | 64.5 | 比较* DM | |
| — | CPSL | 41.2 | 比较字 | 1.875 |
| | | 79.7 | 比较* DM | |
| — | DBS | 24.0 | 常数÷字→字 | 1.875 |
| | | 49.5 | 字÷字→字 | |
| | | 105.0 | * DM÷* DM→* DM | |
| — | DBLS | 67.5 | 字÷字→字 | 1.875 |
| | | 123.0 | * DM÷* DM→* DM | |
| — | DEG | 105.2 | 字→字 | 1.875 |
| | | 140.0 | * DM→* DM | |
| — | EXP | 1.08ms | 字→字 | 1.875 |
| | | 1.12ms | * DM→* DM | |
| — | FCS | 57.9 | 计算一个字,结果放入字 | 1.875 |
| | | 1.75ms | 通过* DM 计算 999 字,结果放入* DM | |
| — | FIX | 65.2 | 字→字 | 1.875 |
| | | 99.6 | * DM→* DM | |
| — | FIXL | 99.6 | 字→字 | 1.875 |
| | | 134.4 | * DM→* DM | |
| — | FLT | 56.0 | 字→字 | 1.875 |
| | | 91.2 | * DM→* DM | |
| — | FLTL | 93.6 | 字→字 | 1.875 |
| | | 128.4 | * DM→* DM | |
| — | FPD | 131.4 | 字指定,无信息,执行 | 1.875 |
| | | 212.4 | * DM 指定,信息,执行 | |
| | | 156.4 | 字指定,无信息,初始 | |
| | | 236.7 | * DM 指定,信息,初始 | |

| 代码 | 助记符 | ON 执行时间(μ s) | 条 件 | OFF 执行时间(μ s) |
|----|------|-------------------|----------------------------------|--------------------|
| — | HEX | 64.5 | 字→字 | 1.875 |
| | | 118.5 | * DM→* DM | |
| — | HKY | 56.4 | 输出字→字 | 1.875 |
| | | 78.0 | 输出 * DM→* DM | |
| | | 63.9 | 输入字→字 | |
| | | 84.9 | 输入 * DM→* DM | |
| — | HMS | 73.9 | 字→字 | 1.875 |
| | | 114.3 | * DM→* DM | |
| — | LINE | 72.8 | 字→字 | 1.875 |
| | | 127.6 | * DM→* DM | |
| — | LOG | 552.0 | 字→字 | 1.875 |
| | | 586.0 | * DM→* DM | |
| — | MAX | 44.8 | 搜索字,结果放入字 | 1.875 |
| | | 1.93ms | 通过 * DM 搜索 999 字,结果放入 * DM | |
| — | MBS | 46.2 | 常数×字→字 | 1.875 |
| | | 48.6 | 字×字→字 | |
| | | 104.0 | * DM×* DM→* DM | |
| — | MBSL | 73.2 | 字×字→字 | 1.875 |
| | | 128.4 | * DM×* DM→* DM | |
| — | MIN | 44.8 | 搜索字,结果放入字 | 1.875 |
| | | 1.33ms | 通过 * DM 搜索 999 字,结果放入 * DM | |
| — | NEG | 33.7 | 变换常数→字 | 1.875 |
| | | 36.1 | 变换字→字 | |
| | | 72.3 | 变换 * DM→* DM | |
| — | NEGL | 41.1 | 变换常数→字 | 1.875 |
| | | 80.1 | 变换 * DM→* DM | |
| — | PID | 1.59ms | 字→字(初始执行) | 1.875 |
| | | 1.73ms | * DM→* DM(初始执行) | |
| | | 458.5 | 字→字(采样时) | |
| | | 673.0 | * DM→* DM(采样时) | |
| — | PLS2 | 619.0 | 字作为控制字 | 1.875 |
| | | 639.8 | * DM 作为控制字 | |
| — | PMCR | 182.0 | 常数作为端口/序列号,DM 作为 I/O 字 | 1.875 |
| | | 728.0 | * DM 作为端口/序列号,* DM 作为 I/O 字 | |
| | | 772.0 | * DM/* EM 作为端口/序列号,* DM,作为 I/O 字 | |
| — | PWM | 202.8 | 由常数确定占空率 | 1.875 |
| | | 207.4 | 由字确定占空率 | |
| | | 223.1 | 由 * DM 确定占空率 | |
| — | RAD | 106.0 | 字→字 | 1.875 |
| | | 140.4 | * DM→* DM | |
| — | SBBL | 59.3 | 字 - 字→字 | 1.875 |
| | | 116.7 | * DM - * DM→* DM | |
| — | SCL2 | 81.5 | 字→字变换,字作为参数字 | 1.875 |
| | | 137.6 | * DM→* DM 变换,* DM 作为参数字 | |

| 代码 | 助记符 | ON 执行时间(μs) | 条 件 | OFF 执行时间(μs) |
|----|------|-------------|-------------------------------|--------------------------|
| — | SCL3 | 86.7 | 字→字变换,字作为参数字 | 1.875 |
| | | 142.8 | * DM→* DM 变换,* DM 作为参数字 | |
| — | SEC | 72.4 | 字→字 | 1.875 |
| | | 112.4 | * DM→* DM | |
| — | SIN | 716.0 | 字→字 | 1.875 |
| | | 750.0 | * DM→* DM | |
| — | SQRT | 206.0 | 字→字 | 1.875 |
| | | | * DM→* DM | |
| — | SRCH | 49.5 | 搜索字,结果放入字 | 1.875 |
| | | 1.99ms | 通过 * DM 搜索 1024 字,结果放入 * DM | |
| | | 11.34ms | 通过 * DM 搜索 6144 字,结果放入 * DM | |
| — | STUP | 160.8 | 内置 RS-232C 端口,字指定 | 1.875 |
| | | 177.0 | 内置 RS-232C 端口,* DM 指定 | |
| | | 160.8 | 外围端口,字指定 | |
| | | 177.0 | 外围端口,* DM 指定 | |
| | | 300.0 | 串行通信板端口 1 和 2,字指定 | |
| | | 317.0 | 串行通信板端口 1 和 2,* DM 指定 | |
| — | TAN | 1.10ms | 字→字 | 1.875 |
| | | 1.14ms | * DM→* DM | |
| — | TTIM | 41.8 | 在字中指定设置值 | 复位 40.0 IL 39.4 JMP 21.0 |
| | | 63.2 | 在 * DM 中指定设置值 | 复位 59.4 IL 6061 JMP 34.0 |
| — | SUM | 57.4 | 加一字,结果放入字 | 1.875 |
| | | 5.15ms | 通过 * DM 加 999 字,结果放入 * DM | |
| — | XFRB | 29.2 | 由常数作为控制数据在字间传送 1 位 | 1.875 |
| | | 45.3 | 由字作为控制数据在字间传送 1 位 | |
| | | 226.5 | 由 * DM 作为控制数据在 * DM 间传送 255 位 | |
| — | ZCP | 31.4 | 比较常数和字域 | 1.875 |
| | | 36.3 | 比较字和字域 | |
| | | 88.7 | 比较 * DM 和 * DM 域 | |
| — | ZCPL | 61.0 | 比较字和字域 | 1.875 |
| | | 116.3 | 比较 * DM 和 * DM 域 | |
| — | + F | 110.4 | 字 + 字→字 | 1.875 |
| | | 162.4 | * DM + * DM→* DM | |
| — | - F | 122.0 | 字 - 字→字 | 1.875 |
| | | 173.8 | * DM - * DM→* DM | |
| — | * F | 120.0 | 字 × 字→字 | 1.875 |
| | | 172.0 | * DM × * DM→* DM | |
| — | /F | 135.6 | 字 ÷ 字→字 | 1.875 |
| | | 187.0 | * DM ÷ * DM→* DM | |

7-3-3 I/O 响应时间

I/O 响应时间是指 PC 接收到输入信号后 (即输入位变 ON 后), 检查和处理信息并输出一个控制信号 (即输出处理结果到输出位) 所需要的时间。I/O 响应时间随时序和处理条件而变化。用下面的程序作为例子, 最小和最大 I/O 响应时间如下所示。



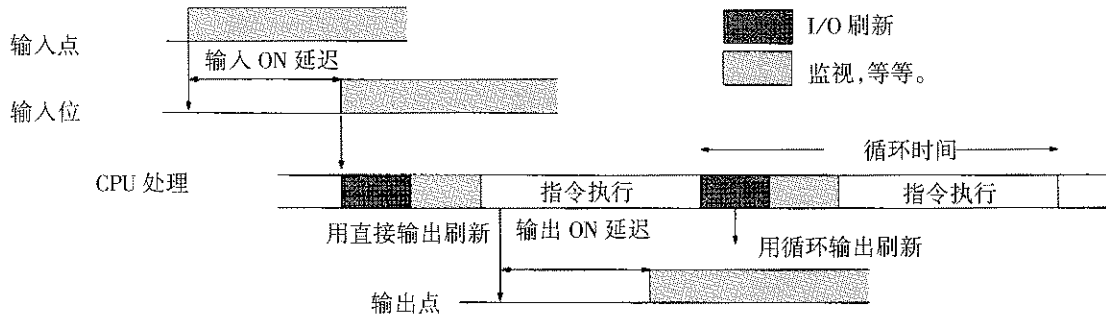
取下列条件作为计算 I/O 响应时间的例子。

- 输入 ON 延迟: 8ms
- 监视时间: 1ms
- 指令执行时间: 14ms
- 输出 ON 延迟: 10ms
- 输出指令位置: 程序开始
- 通信口: 不用

注 DC 输入单元的输入 ON 延迟可以在 PC 设置中设置。

最短 I/O 响应时间

如下图所示。当 CQM1H 接收到一输入信号刚好在输入刷新循环阶段之前其响应最快。如下图所示。



使用循环输出刷新时:

$$\text{最短 I/O 响应时间} = 8 + 15 + 10 = 33\text{ms}$$

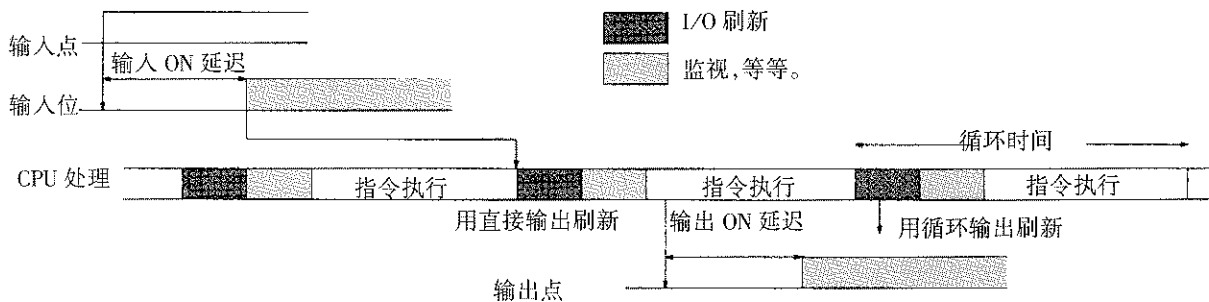
使用直接输出刷新时:

$$\text{最短 I/O 响应时间} = 8 + 1 + 10 = 19\text{ms}$$

注 可以通过使用输入中断和直接输出刷新得到最快响应时间 (标准 100μs)。

最长 I/O 响应时间

如下图所示, 当 CQM1H 接收到一输入信号刚好在输入刷新循环阶段之后其响应时间最长。此时, 会产生约一个循环的延迟。



使用循环输出刷新时:

$$\text{最短 I/O 响应时间} = 8 + 15 \times 2 + 10 = 48\text{ms}$$

使用直接输出刷新时:

$$\text{最短 I/O 响应时间} = 8 + 15 + 10 = 33\text{ms}$$

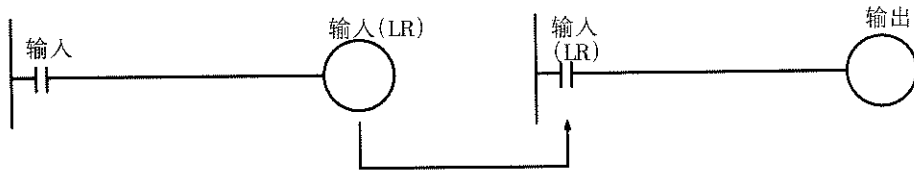
7-3-4 一对一链接 I/O 响应时间

当两个 CQM1H 链接成一对一时, I/O 响应时间是通过一对一链接通信在一个 CQM1H 的输入执行到输出至另一个 CQM1H 所需的时间。

一对一链接通信在主站和从站之间交互进行。根据使用的 LR 字数, 各传送时间如下所示。

| 使用的字数 | 传送时间 |
|-------------------|------|
| 64 字(LR00 ~ LR63) | 39ms |
| 32 字(LR00 ~ LR31) | 20ms |
| 16 字(LR00 ~ LR15) | 10ms |

用如下图解作为主站和从站间执行指令的例子以说明最短和最长 I/O 响应时间,在此例中,通信从主站传送到从站。



作为计算 I/O 响应时间的例子的条件如下:

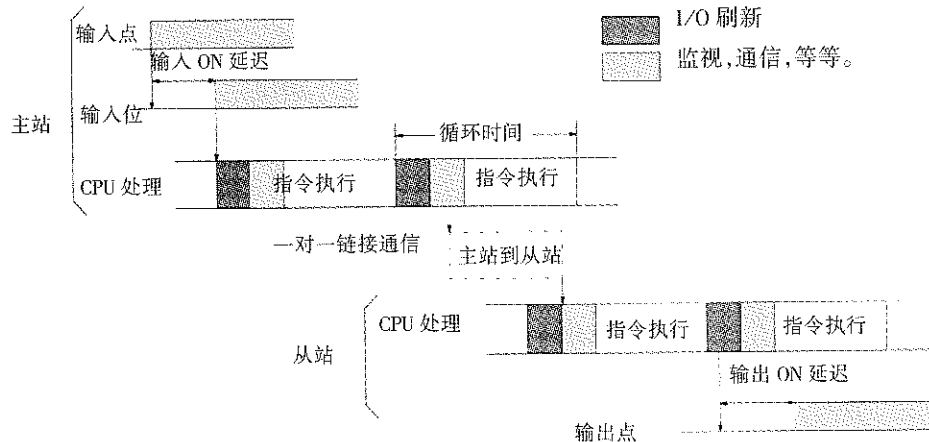
- 输入 ON 延迟: 8ms
- 主站循环时间: 10ms
- 从站循环时间: 15ms
- 输出 ON 延迟: 10ms
- 直接输出: 不用
- LR 字数目: 64

注 DC 输入单元的输入 ON 延迟可在 PC 设置中设定。

在如下情况时 COM1H 响应最快。

最短 I/O 响应时间

- 1,2,3, ... 1. COM1H 接收到输入信号刚好在输入刷新循环阶段之前。
2. 主站到从站传送立即开始。
3. 通信完毕后从站立即执行通信服务。



最短 I/O 响应时间如下:

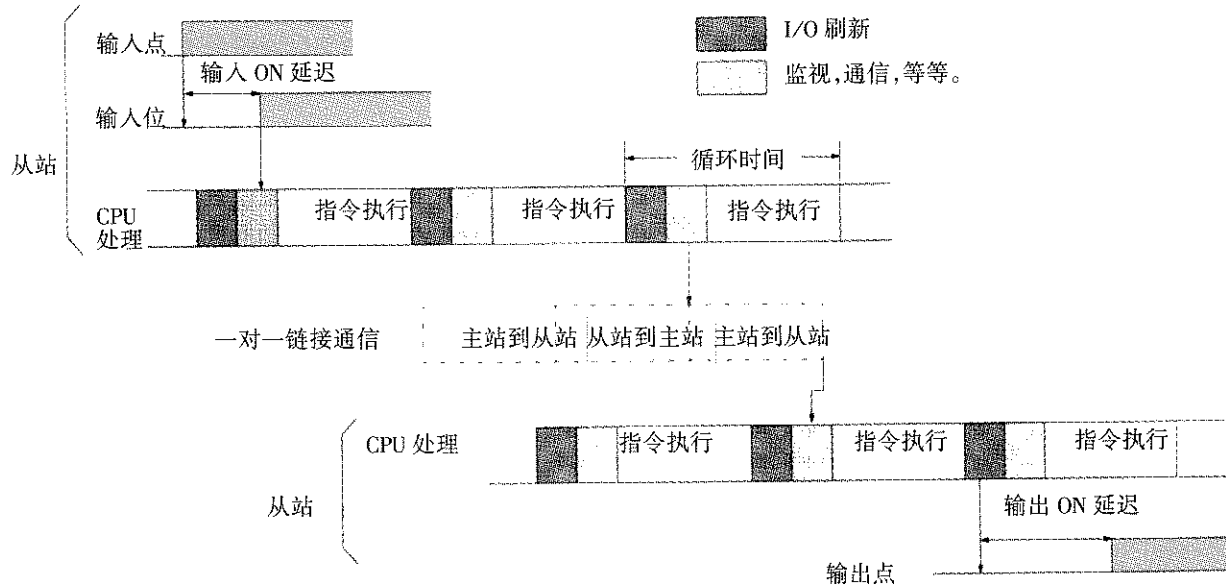
- 输入 ON 延迟: 8ms
- 主站循环时间: 10ms
- 传送时间: 39ms
- 从站循环时间: 15ms
- + 输出 ON 延迟: 10ms

- 最短 I/O 响应时间: 82ms

最长 I/O 响应时间

如下情况时 CQM1H 需要最长响应时间。

- 1.2,3... 1. CQM1H 接收一输入信号刚好在输入刷新循环阶段之后。
2. 主站到从站传送开始不及时。
3. 通信正好在从站执行通信服务后才完成。



最长 I/O 响应时间如下:

| | |
|--------------|----------|
| 输入 ON 延迟: | 8ms |
| 主站循环时间: | 10ms × 2 |
| 传送时间: | 39ms × 3 |
| 从站循环时间: | 15ms × 2 |
| + 输出 ON 延迟: | 10ms |
| <hr/> | |
| 最长 I/O 响应时间: | 185ms |

7-3-5 中断处理时间

本节介绍的处理时间包括从执行中断直到中断处理子程序被调用的时间,和从中断处理子程序完成直到回到原来位置的时间。该解释适用于下列三种类型的中断:输入中断,间隔定时器中断和高速计数器中断。

处理时间

下列说明的时间包括从中断信号发生直到调用中断处理子程序,和从中断处理子程序完成时直到返回到原来位置的时间。

| 项 目 | 内 容 | 时 间 |
|----------------|---|--|
| 中断输入 ON 延迟 | 这是从中断输入位置 ON 一直到执行中断的延迟时间。它与其它中断无关。 | 50μs |
| ↓ (实现中断条件)(见注) | | |
| 等待直到中断屏蔽处理完成 | 这是中断等待直到处理完成的时间。当执行屏蔽处理时这种情况发生。下面有更详细的解释。 | 见下面 |
| ↓ | | |
| 切换至中断处理 | 这将是处理切换到中断所需的时间。 | 输入中断间隔定时器中断 或高速计数器中断:30μs 串行通信板中断:55μs |
| ↓ | | |
| 中断时输入刷新 | 这是设定在调用中断处理子程序之时执行输入刷新时所需要的输入刷新时间(在 PC 设置DM6630 ~ DM6638 中设定) | 每字 10us |
| ↓ (执行中断处理子程序) | | |
| 返回 | 从执行 RET(93)到返回到被中断的处理所化的时间。 | 30us |

- 注 1. 当高速计数器 0 使用区域比较表时,循环时间会影响中断处理时间。
 2. 当脉冲 I/O 板或绝对值编码器接口板的高速计数器 1 和 2 使用范围比较表时(对 CQM1H—51/61 CPU 单元),中断处理时间会延迟达 1ms。

屏蔽处理

在如下所述操作时中断被屏蔽。在指示时间内任何中断保持屏蔽直到处理完成。

高速定时器: 根据 (a) TIMH(15) 使用的定时器数目和 (b) 此时活动的高速定时器数目,需要如下的时间。(在 PC 设置 DM6629 中设定高速定时器数目, 缺省设置为 16)。

$$0 \leq \text{等待时间} \leq 40 + 3 \times (a + b) \mu\text{s}$$

即使在不用高速定时器时需要高达 40μs。

非致命错误的发生和清除: 当一非致命的错误发生而错误内容登录到 CQM1H 时, 或当错误正被清除时, 中断会被屏蔽长达 75μs 直到处理完成。

在线编辑: 当操作时执行在线编辑, 中断可被屏蔽最长达 250ms。

基于 SPED(64) 的脉冲输出还可能受到中断处理影响, 因此会导致输出时序的变化。

计算例

本例说明了当如下条件时使用输入中断的中断响应时间 (即, 从中断输入置 ON, 直到中断处理子程序启动之间的时间)。

高速定时器数目: 0(没有高速定时器启动)

在线编辑: 不用

中断时输入刷新: 没有

最短响应时间

| | |
|-------------|------------|
| 中断输入 ON 延迟: | 50 μ s |
| 中断屏蔽等待时间: | 0 μ s |
| + 切换至中断处理: | 30 μ s |

最短响应时间: 80 μ s

最长响应时间

| | |
|-------------|------------|
| 中断输入 ON 延迟: | 50 μ s |
| 中断屏蔽等待时间: | 40 μ s |
| + 切换至中断处理: | 30 μ s |

最长响应时间: 120 μ s

除了上述响应时间外,还有中断处理子程序本身执行需要的时间以及还必须考虑返回被中断的处理时的 30 μ s 返回时间。

在程序中使用中断时务必允许中断处理时间。

如果使用直接输出,从中断子程序的输出就可以立即输出。在主程序和中断子程序中同时使用直接输出。不可分开设定。

第 8 章

故障诊断

本章概述如何诊断和改正在 PC 操作时可能发生的硬件错误和软件错误。

| | |
|-------------|-----|
| 8-1 引言 | 486 |
| 8-2 编程器操作错误 | 486 |
| 8-3 编程错误 | 487 |
| 8-4 用户定义错误 | 488 |
| 8-5 操作错误 | 489 |
| 8-5-1 非致命错误 | 489 |
| 8-5-2 致命错误 | 491 |
| 8-6 错误日志 | 491 |
| 8-7 故障诊断流程图 | 493 |

8—1 引言

PC 错误一般可以分为下列四类：

- 1,2,3... 1. 程序输入错误
这些错误在输入程序时或在着手为 PC 运行作准备的操作时发生。
2. 编程错误
这些错误会在用程序检查操作对程序进行检查时发生。
3. 用户定义错误
用户可用来自定义其自身错误或信息的指令有三条。在操作时这些指令会在发生特定条件(由用户定义)时执行。
4. 操作错误
这些错误在程序开始执行后发生。
 - a)非致命操作错误
PC 操作和程序执行在发生一个或几个这种错误后仍继续进行。
 - b)致命操作错误
任一个这种错误发生时, PC 操作和程序执行会停止进行, 且 PC 的所有输出都会变为 OFF。

PC 发生错误时 PC 指示灯会显示, 如果连接有编程器或上位计算机, 则错误信息或代码会显示在编程器或上位计算机上。SR25300 ~ SR25307 内也含有错误代码。

对于大多数的最近错误, PC 的错误日志区会记录错误的类型和发生的时间。从第 491 页开始提供详细介绍。

SR 和 AR 区提供的各标志和其它信息可在故障诊断中使用。关于这些表请参阅第三章 存储器区。

注 除上述的各类错误外, 当 PC 是上位链接系统的组成部分时, 还可能发生通信错误。详细介绍请参阅第六章 上位机链接命令。

8—2 编程器操作错误

在编程器上执行操作时可能出现下列错误信息。按指示改正错误, 而后继续操作。关于在操作 SSS 或数据存取控制器时可能出现的错误请参阅梯形图支持软件操作手册、SYSMAC 支持软件操作手册: C - 系列 PC 或数据存取控制器操作手册。

| 信 息 | 意义和适当反应 |
|--------------|--|
| REPL ROM | 试图对写保护存储器进行写入。将写保护开关(CPU 单元 DIP 开关的脚 1)置于 OFF。 |
| PROG OVER | 存储器最末地址处的指令不是 NOP(00)。擦除程序结束处的所有不必要的指令。 |
| ADDR OVER | 设置了比程序存储器内最高存储器地址高的地址。输入较低地址。 |
| SET DATA ERR | FAL 00 已被输入, 而“00”不能输入。重新输入新据。 |
| I/O NO. ERR | 已指定的数据区地址超出了数据区的范围, 例如, 一地址太大。确认指令的要求, 而后重新输入地址。 |

8—3 编程错误

在使用程序检查操作对程序进行检查时,这些程序语法错误会被检测出来。有三级程序检验可供使用。要检出被检测的错误类型,必须指定所要求的级别。下表给出了所有语法错误的错误类型、显示和说明。检验级 0 可检验 A 类、B 类和 C 类错误;检验级 1 可检验 A 类和 B 类错误;而检验级 2 仅检验 A 类错误。

A 级错误


| 信 息 | 意义和适当应答 |
|--------------|---|
| ????? | 程序已被破坏,生成一个不存在的功能代码。重新输入程序。 |
| CIRCUIT ERR | 逻辑块数和逻辑块指令不一致,即,LD 或 LD NOT 已被用来启动一个其执行条件还未被另一指令使用的逻辑块,或已被使用的逻辑块指令并不具有所需的逻辑块数。检验程序。 |
| OPERAND ERR | 给指令输入的常数不在所要求的值内,更改常数使之在正确范围内。 |
| NO END INSTR | 程序中没有 END(01)。在程序的最末地址写入 END(01)。 |
| LOCN ERR | 一指令在程序中位置错了。检验指令要求并改正程序。 |
| JME UNDEFD | 对 JMP(05)指令,缺少 JME(04)指令。改正跳转号或插入正确的 JME(04)指令。 |
| DUPL | 同一跳转号或子程序号已被使用两次,更正程序以使同一个号每个只使用一次。 |
| SBN UNDEFD | 编程 SBS(91)指令有并不存在的子程序号,更正子程序号或编制所要求的子程序。 |
| STEP ERR | 没有正确地使用有段号的 STEP(08)和无段号的 STEP(08)指令。检验 STEP(08)编程要求,改正程序。 |

B 级错误

| 信 息 | 意义和适当应答 |
|-------------|--|
| IL—ILC ERR | IL(02)和 ILC(03)未被成对使用。更正程序,以使每个 IL(02)有唯一的 ILC(03)。虽然如果多于一个 IL(02)使用同样的 ILC(03)会出这个错误信息,但程序会如所编写的执行。在处理前,确认程序是按要求而编写的。 |
| JMP—JME ERR | JMP(04)和 JME(05)未被成对使用。在处理前,确认程序是按要求而编写的。 |
| SBN—RET ERR | 如果显示的是 SBN(92)的地址,则两个不同的子程序已被定义有相同子程序号。改变子程序号中的一个或删除子程序中的一个。如果显示的是 RET(93)的地址,则 RET(93)未被正确使用。检验 RET(93)的各要求,并更正程序。 |

C 级错误

| 信 息 | 意义和适当应答 |
|------------|--|
| COIL DUPL | 同一个位正受到一个以上指令(如, OUT, OUT NOT, DIFU(13)、DIFD(14)、KEEP(11)、SFT(10))控制(即变为 ON 和/或 OFF)。虽然对某些指令这是允许的,但还是要检验指令要求,以确认程序是正确的,或重新编写程序,以使每位仅受一条指令控制。 |
| JMP UNDEFD | JME(05)没有与具有相同跳转号的 JMP(04)一起使用。加上具有相同号的 JMP(04),或删除未被使用的 JME(05)。 |
| SBS UNDEFD | 存在不被 SBS(91)调用的子程序。在正确位置编写一个子程序调用,或如果不需要就删除该子程序。 |

 **小心** 扩展指令(被赋予功能代码 17、18、19、47、48、60~69、87、88 和 89 的那些指令)不受程序检验。程序检验也不包含 DM3070~DM6143,某些 PC 并不支持 DM 区的这部分(如, CQM1H—CPU11 和 CQM1H—CPU21)。即使指定这些区,数据也不会被写入,从这些区读数据将总是未定义的。

8-4 用户定义的错误

有四条指令可供用户用来定义错误或信息。这些指令被用来生成警告(ERR/ALM 灯闪烁的非致命错误)或错误(ERR/ALM 灯亮的致命错误),并在编程器上显示消息。

消息—MSG(46)

MSG(46)用来在编程器上显示消息。当该指令条件为 ON 时,消息(多达 16 个字符长)就被显示。详情请参阅第 369 页。

故障报警—FAL(06)

FAL(06)是一条引起非致命错误的指令。详情请参阅第 221 页。当 FAL(06)指令被执行时会发生下列情况:

- 1, 2, 3, ... 1. CPU 的 ERR/ALM 指示灯会闪烁。PC 操作会继续。
2. 指令的 2 位数 BCD FAL 号(01~99)会被写入 SR25300~SR25307。
3. FAL 号会被记录在 PC 错误日志区。如果使用带有时钟(RTC)的存储器盒,发生时间也会记录。

FAL 号可以任意设置以表示特定条件。同一个号不能用作 FAL 号和 FALS 号。为了清除 FAL 错误,排除错误的起因,执行 FAL00,而后用编程器清除错误。

详情请参阅第 221 页。

严重故障报警—FALS(07)

FALS(07)是一条引起致命错误的指令,详情请参阅第 221 页。当 FALS(07)指令被执行时会发生下列情况:

- 1, 2, 3, ... 1. 程序执行会被停止且输出会变为 OFF。
2. CPU 的 ERR/ALM 指示灯会发亮。
3. 指令的 2 位数 BCD FALS 号(01~99)会被写入 SR25300~SR25307。
4. FALS 号会被记录在 PC 错误日志区。如果使用带有时钟(RTC)的存储器盒,则发生时间也会记录。


FALS 号可以任意设置以表示特定条件。同一个号不能用作 FAL 号和 FALS 号。为了清除 FALS 错误,将 PC 切换到编程模式,排除错误的起因,而后再用编程器清除错误。

故障点检测—FPD(一)

使用 FPD(一)也能产生非致命错误和错误消息。详情请参阅第 375 页。

8-5 操作错误


有两种操作错误:非致命错误和致命错误。在非致命错误发生后 PC 操作会继续进行,但如果发生致命错误操作就会停止。

 **小心** 查清所有错误,不管严重与否,尽快排除错误的起因,而后再起 PC。关于有关错误的硬件信息和编程器操作请参阅 *CQM1H 操作手册*。

8-5-1 非致命错误

在这些错误中的一个或几个发生后,PC 操作和程序执行会继续进行。虽然 PC 操作会继续,但应排除错误的起因并尽快清除错误。

当这些错误中的一个发生时,POWER 和 RUN 指示灯会保持发亮,而 ERR/ALM 指示灯会闪烁。

 **小心** 虽然产生非致命错误时 PC 操作可继续进行,但仍应尽快查出错误原因并采取相应措施。排除故障原因后,即可再次将 PC 置为 OFF 和 ON,亦可通过编程器清除错误。编程器步骤参阅 *CQM1H 操作手册*。

| 消息 | FAL 号 | 意义和适应应答 |
|--------------------------|---------|--|
| SYS FAIL FAL * * (见注) | 01 ~ 99 | 程序的 FAL(06)指令已被执行。检查 FAL 号以确定会引起执行的条件,改正起因,并清除错误。 |
| | 9D | <p>在 CPU 和存储器盒之间传输数据时发生了错误。检验标志 AR1412 ~ AR1415 的状态,并如指出的改正之。</p> <p>AR1412 ON: 切换到编程模式,清除错误,再次传送。</p> <p>AR1413 ON: 传送目的地被写保护。</p> <p>如果 PC 是目的地,断开 PC 的电源,确保 CPU 单元的 DIP 开关 1 为 OFF。清除错误,再次传送。</p> <p>如果 EEPROM 存储器盒是目的地,检验电源是否是 ON。清除错误,再次传送。</p> <p>如果 ERPOM 存储器盒是目的地,改为可写存储器盒。</p> <p>AR1414 ON: 目的地容量不足。检验 AR15 中的源程序容量,并考虑使用不同的 CPU 或存储器盒。</p> <p>AR1415 ON: 存储器盒内无程序或程序有错误,检验存储器盒。</p> |

| 消息 | FAL 号 | 意义和适应应答 |
|--------------------------|-------|--|
| | 9C | <p>检验 AR0400 ~ AR0407(插在槽 1 中的内板错误代码)的内容(2 位数 BCD),并如指出的改正之。</p> <p>01、02Hex:硬件发生错误。断开电源,然后再上电。如果错误继续存在,更换内板。</p> <p>03 Hex:PC 设置(DM6650 ~ DM6559、DM6613、DM6614、DM6602、DM6603、DM6640、DM6641)设定不正确,更正设定。</p> <p>10 Hex:串行通信板发生错误。检验内存中的串行通信板的标志和状态信息,并依此更正。</p> <p>槽 1 或槽 2 中的内板发生错误。</p> <p>检查 AR0408 ~ AR0415(插在槽 2 中的内板错误代码)的内容(2 位数 BCD),并如指出的改正之。</p> <p>01、02 Hex:硬件发生错误。断开电源,然后再上电。如果错误继续存在,更换内板。</p> <p>03 Hex:PC 设置(DM6611、DM6612、DM6643、DM6644)设定不正确更正设定。</p> <p>04 Hex:CQM1H 操作在脉冲输出时被中断(安装了 CQM1H—PLB21)。</p> <p>检测一下设备,对输出脉冲的接收是否受到影响。</p> |
| SYS FAIL FAL * * (见注) | 9B | <p>在 PC 设置中检测出错误。检验标志 AR2400 ~ AR2402,如指出的更正之。</p> <p>AR2400 ON:当电源被接通时,在 PC 设置中(DM6600 ~ DM6614)检测出错误设定。在编程模式中改正设定,并再次接通电源。</p> <p>AR2401 ON:当切换到运行模式时,在 PC 设置中(DM6615 ~ DM6644)检测出不正确设定。在编程模式中改正设定,并再次切换到运行模式。</p> <p>AR2402 ON:操作时在 PC 设置中(DM6645 ~ DM6655)检测出不正确设定。更正设定并清除错误。</p> |
| SCAN TIME OVER | F8 | <p>监视定时器已超过 100ms。(SR25309 会为 ON)。</p> <p>这表示程序循环时间大于推荐的时间。如果可能就减少循环时间。</p> |
| BATT LOW | F7 | <p>后备电池丢失或其电压跌落。(SR25308 会为 ON)。</p> <p>检验电池,如果需要就更换之。检验 PC 设置(DM6655)看看是否检测电池低压。</p> |
| SIOU__ERR | D0 | <p>CPU 单元与 Controller Link 单元或 Controller Link 单元本身在传送数据时产生错误(SR25413 和 AR0011 为 ON)。关闭电源,然后再接通。若错误依然存在,更换 Controller Link 单元。</p> |

注 ** 为 01 ~ 99,9D,9C 或 9B。

通信错误

如果通过外围端口或内置 RS—232C 端口的通信发生错误,则对应的指示灯(PRPHL 或 COMM)会停止闪烁。检验连接电缆以及 PC 和上位计算机的程序。

用端口复位位、SR25208 和 SR25209 使通信端口复位。

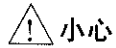
输出禁止

当 OUT INH 指示灯发亮时,输出 OFF 位(SR25215)为 ON,CPU 单元的所有输出会变为 OFF。如果不必使所有输出都为 OFF,则将 SR25215 变为 OFF。

8-5-2 严重错误

当这些错误中的任一个发生时, PC 操作和程序执行会停止, PC 的所有输出会变为 OFF。

所有 CPU 单元指示灯会因为电源中断错误而为 OFF。对于所有其它致命操作错误, POWER 和 ERR/ALM 指示灯将会发亮, RUN 指示灯将会 OFF。



小心

查出错误原因并尽快采取适当措施。清除错误起因后, 即可再启动 PC, 亦可运行错误清除操作。有关编程器步骤请参阅 *CQM1H 操作手册*。

| 消息 | FALS 号 | 意义和适应应答 |
|---------------------------|---------|--|
| 电源中断(无信息) | 无 | 电源已中断至少 10 ms。检查电源电压和电源线, 再次试上电。 |
| MEMORY ERR | F1 | AR1611 ON: 检验和错误已在 PC 设置中(DM6600 ~ DM6655)发生。初始化所有 PC 设置, 而后重新输入。 |
| | | AR1612 ON: 检验和错误已在程序中发生, 同时指出不正确的指令。检查程序并更正检测出的任何错误。 |
| | | AR1613 ON: 检验和错误已在扩展指令数据中发生。初始化所有扩展指令设定, 而后重新输入。 |
| | | AR1614 ON: 通电时安装或取下存储器盒。断开电源, 安装存储器盒, 而后再接通电源。 |
| | | AR1615 ON: 起动时读不出存储器盒内容。检验标志 AR1412 ~ AR1415, 以确定问题所在, 改正之, 而后再次接通电源。 |
| NO END INST | F0 | END(01)没有被写入程序的任何地方。在程序的最末地址处写入 END(01)。 |
| L/O BUS ERR | C0 | 在 CPU 和 L/O 单元之间传送数据时发生了错误。使用标志 AR2408 ~ AR2415 确定问题的单元, 将电源断开, 检查松动的 L/O 单元或端盖, 再次接通电源。 |
| L/O UNIT OVER | E1 | 安装的 L/O 单元上的 L/O 字数超过了最大值。断开电源重新安排系统, 以减少 L/O 字数, 再接通电源。 |
| SYS FAIL FALS * * (见注) | 01 ~ 99 | FALS(07)指令已在程序中执行。检查 FALS 号以确定会引起执行的条件, 排除起因, 清除错误。 |
| | 9F | 循环时间已超过 FALS 9F 循环时间监视时间(DM6618)。检查循环时间, 如果需要就调整循环时间监视时间。 |

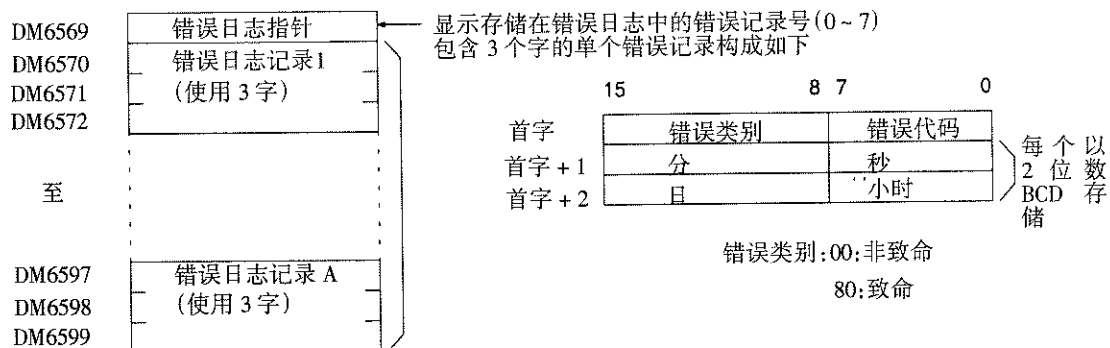
注 * * 为 01 ~ 99 或 9F

8-6 错误日志

错误日志寄存发生在 PC 上的任何致命或非致命错误的错误代码, 以及其发生的日期和时间。请参阅第 489 页的错误代码。

错误日志区

错误日志存储在 DM6569 ~ DM6599, 如下所示。



即使 CPU 单元上的 DIP 开关的脚 1 被置 ON 以保护 DM6144 ~ DM6655, 错误记录也会被存储。

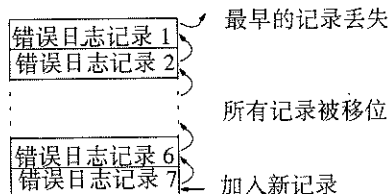
有关错误代码详情参见 8-5 操作错误。

若在 PC 设置 (DM6655, 位 00 ~ 03) 设定为禁止保存记录到错误历史 (2 ~ F Hex); DM6569 ~ DM6599 可作为一般的只读 DM 字使用。

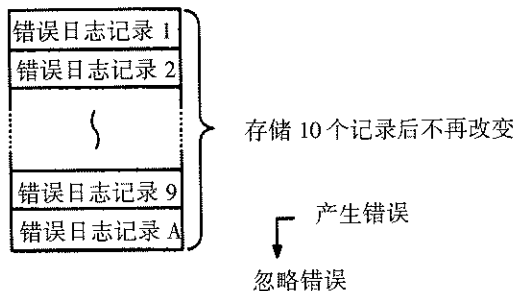
错误日志存储方法

错误日志存储方法是设置在 PC 设置 (DM6655 位 00 至 03) 内, 设置下列中的任一方法。

1. 0 Hex: 可以存储最近的 10 个错误日志记录并放弃较早的记录。如下所示, 这是通过移动记录来完成的, 致使每当产生一个新记录时最早的记录 (记录 0) 消失。



2. 1 Hex: 可以只存储最先 10 个错误日志记录, 而忽略 10 个以外的任何后继错误。



3. 2 ~ F Hex: 可以禁止日志, 以致无记录存储。

缺省设定是第一方法。关于错误日志的 PC 设置的详情请参阅第 16 页的错误日志设定。

- 注
1. 如果安装无时钟 (RTC) 的存储器盒, 则产生错误的日期和时间会全为“0000”。
 2. 即使 CPU 单元前侧的 DIP 开关的脚 1 被变为 ON, 使 DM6144 ~ DM6655 为写保护, 错误也会被存储在错误日志中。

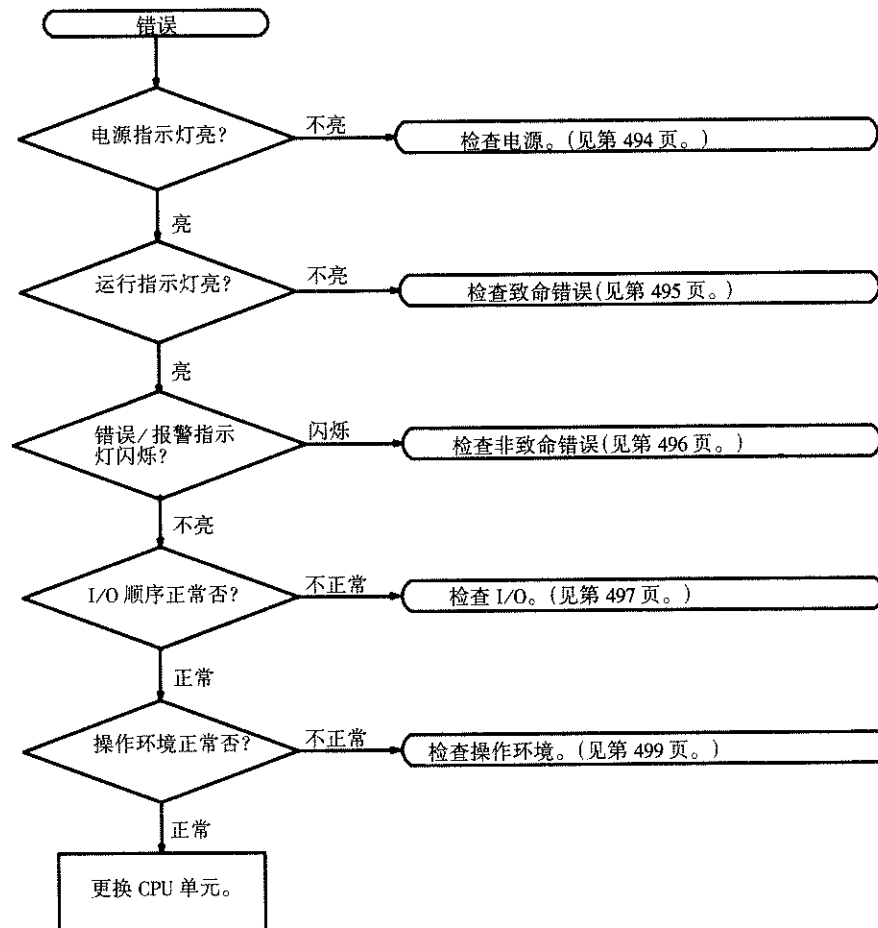
清除错误日志

为清除全部错误日志,可由编程设备或用指令使 SR25214 变为 ON。(在错误日志被清除后,SR25214 会再次自动变为 OFF。)

8-7 故障诊断流程图

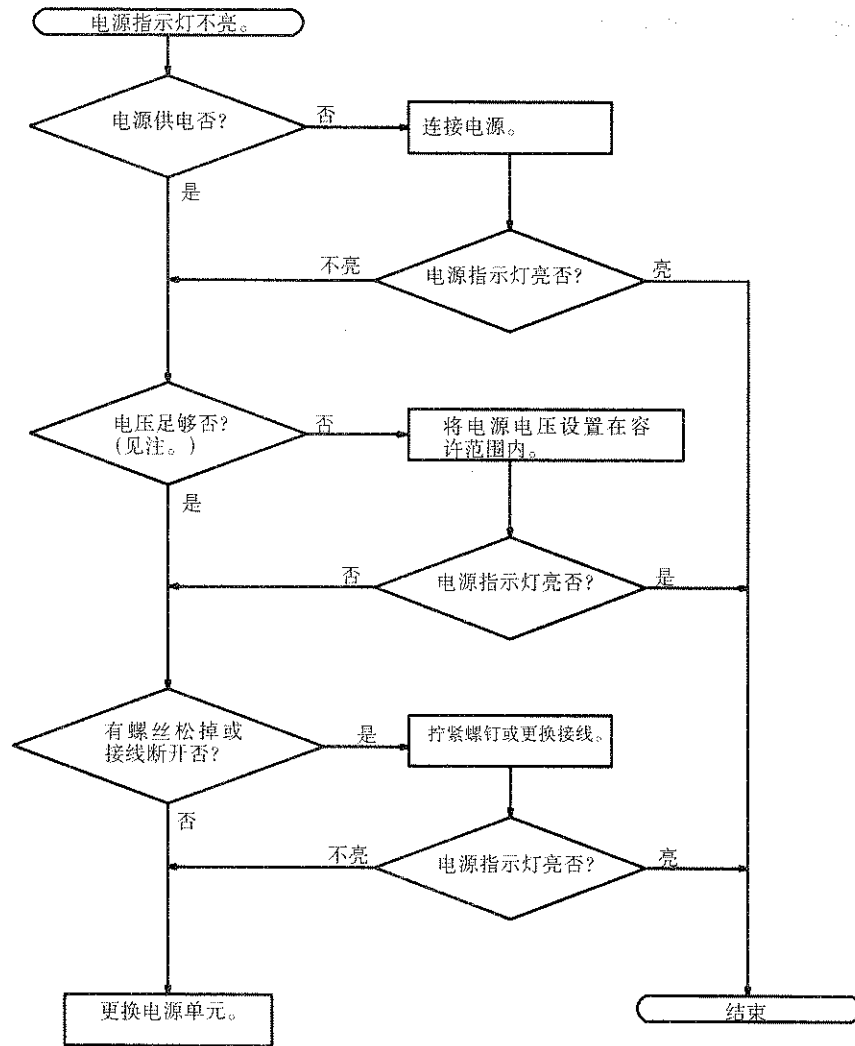
使用下列流程图诊断在操作时发生的错误。

总检查



注 在更换各单元、电池、接线、或电缆等之前总要将 PC 的电源断开。

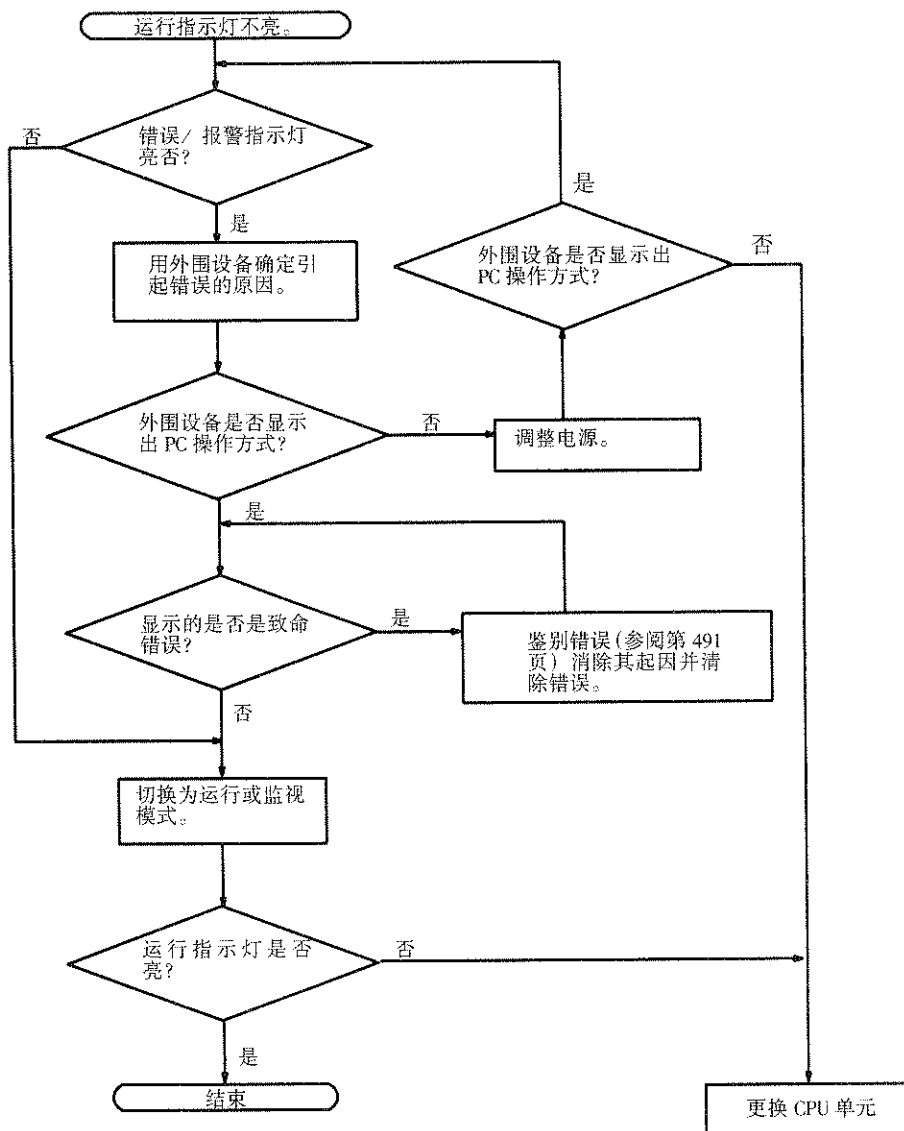
电源检查



注 CQM1H 的容许电压范围参阅 CQM1H 操作手册。

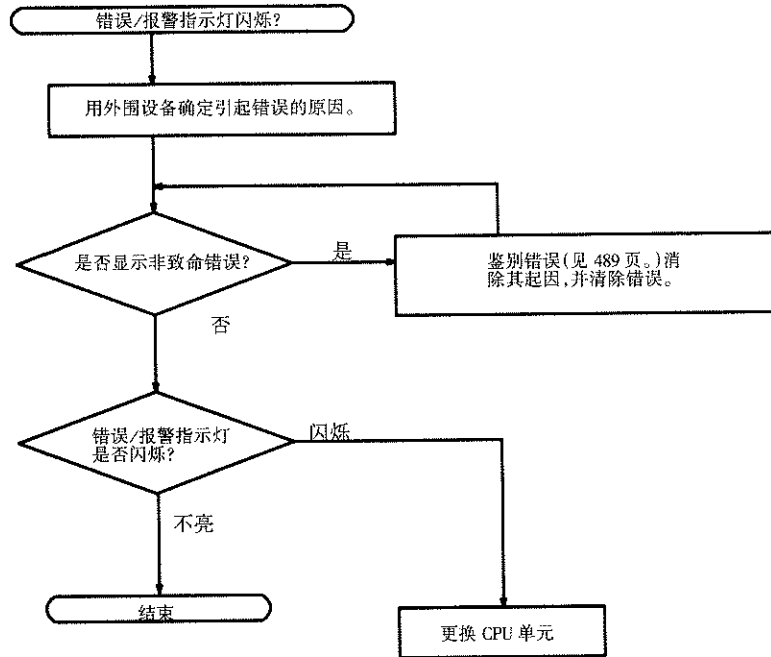
致命错误检查

电源指示灯亮着时发生的致命错误可用下列流程图来查找。



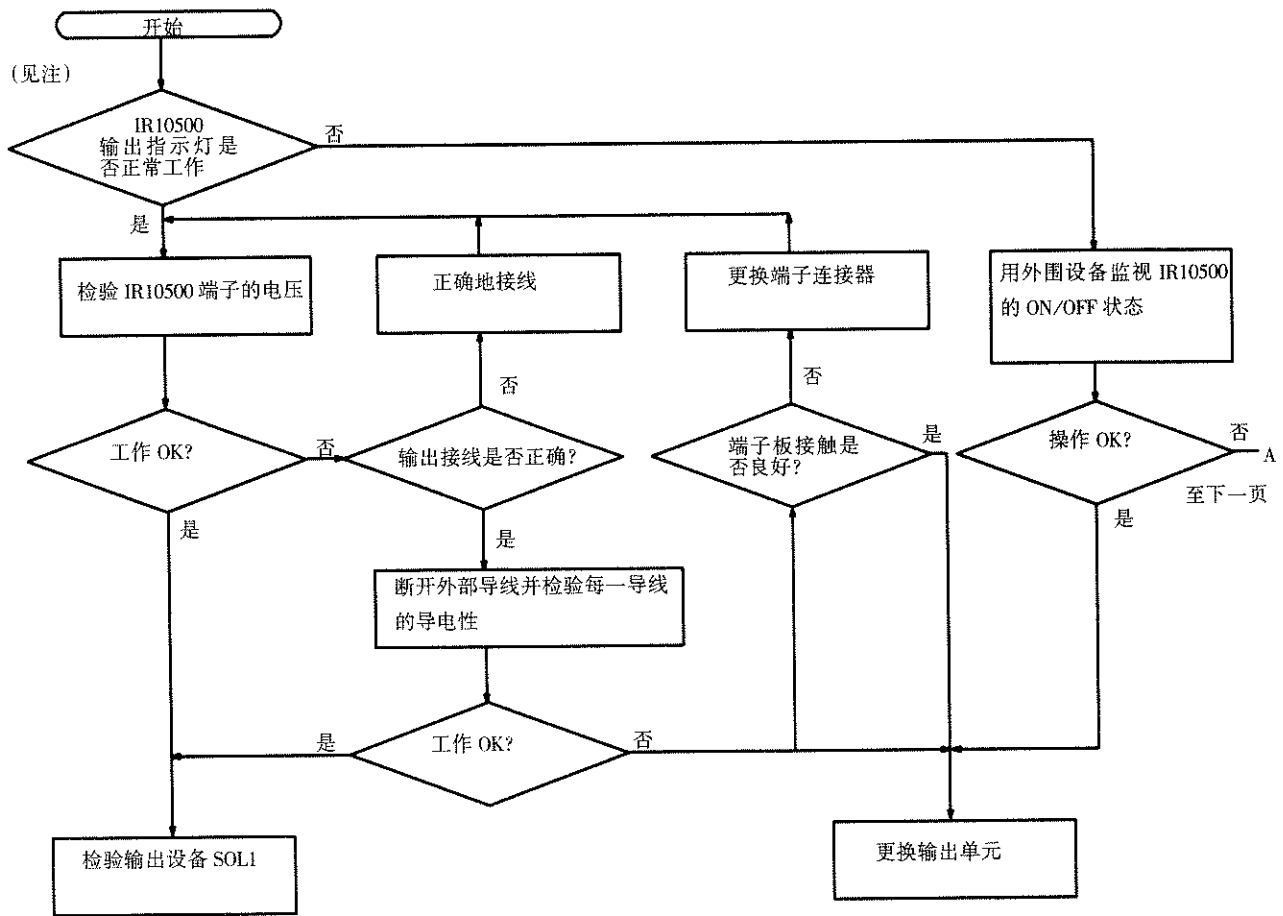
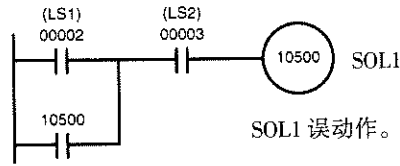
非致命错误检查

虽然 PC 在非致命错误时会继续操作,但应确定错误的起因,尽快消除错误以确保正常操作。为消除某些非致命错误也许需要停止 PC 操作。

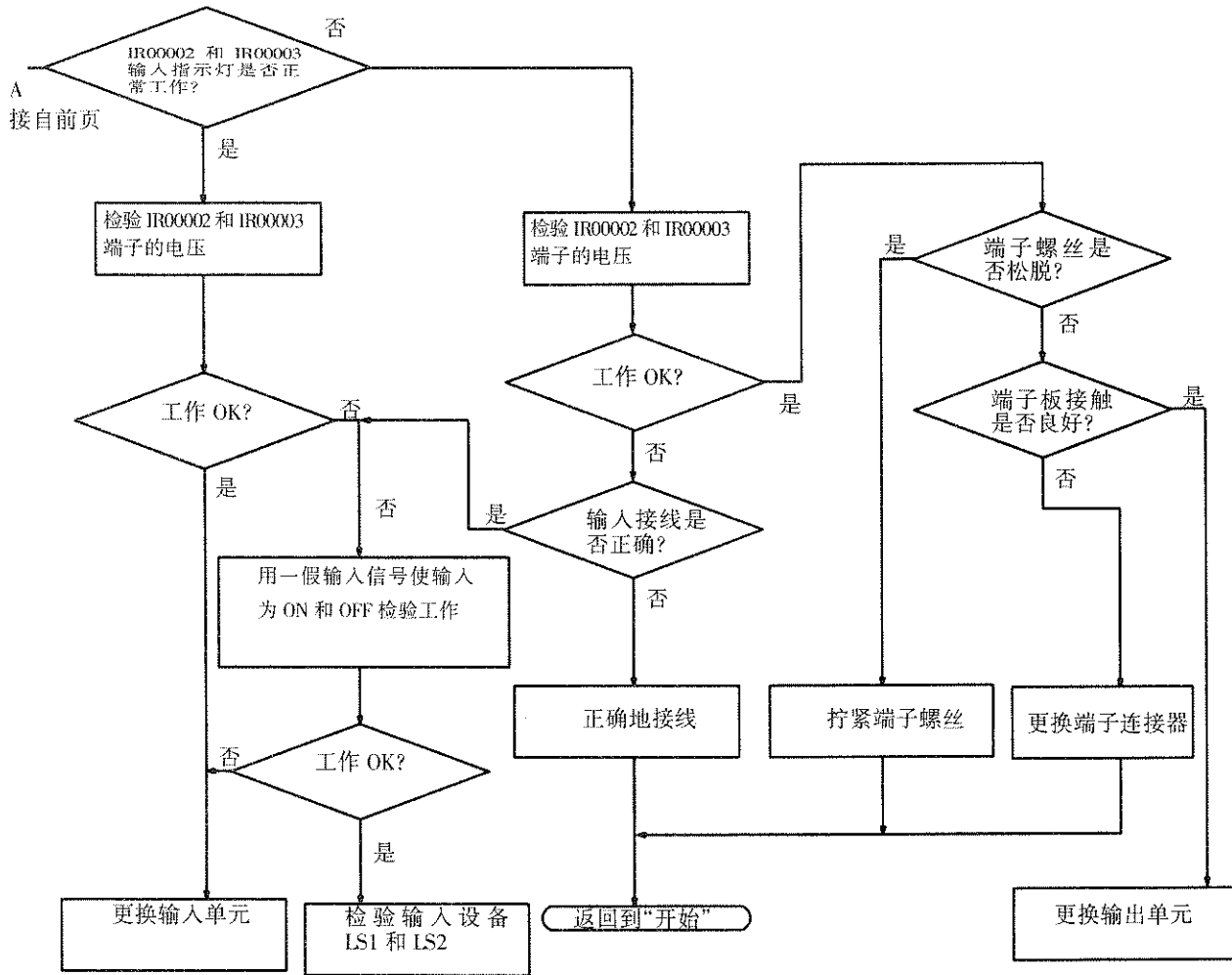


I/O 检验

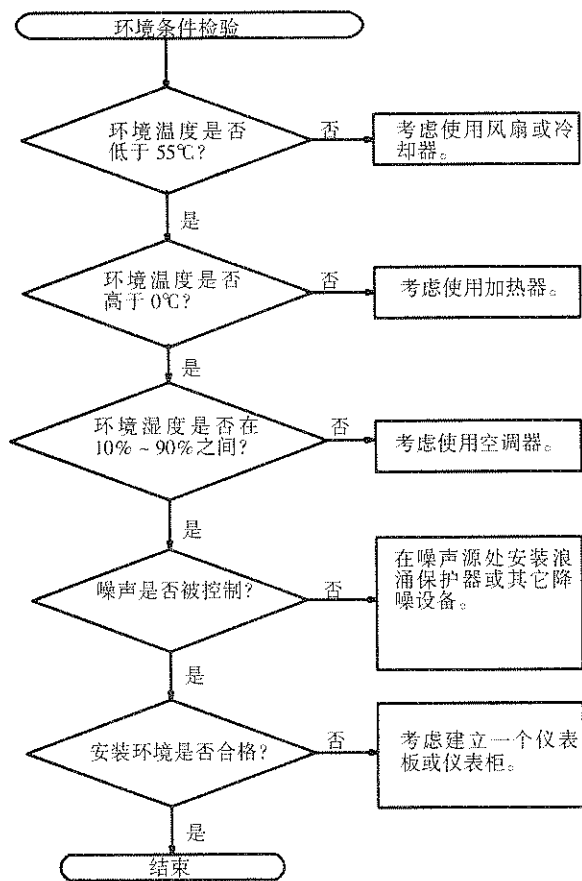
I/O 检验流程图是根据下列梯形图部分。



导致错误的原因可能是保险丝熔断或输出晶体管误操作。



环境条件检验





附录 A 编程指令

PC 指令是通过按相应的编程器键(例如, LD、AND、OR、NOT)或用功能代码来输入的。要输入一条指令及其功能代码,按 FUN、功能代码,而后按 WRITE。详情请参阅列举编程和指令的各页。

| 代码 | 助记符 | 名称 | 功 能 | 页 |
|----|---------|---------|--|-----|
| - | AND | 与 | 将指定位状态与执行条件进行逻辑与。 | 213 |
| - | AND LD | 与装载 | 将先前各块的结果进行逻辑与。 | 214 |
| - | AND NOT | 与非 | 将指定位的反相与执行条件进行逻辑与。 | 213 |
| - | CNT | 计数器 | 减量计数器。 | 226 |
| - | LD | 装载 | 用于起动具有指定位状态的指令行,或定义 AND LD 和 OR LD 用的逻辑块。 | 213 |
| - | LD NOT | 装载非 | 用于起动具有指定位的反相的指令行。 | 213 |
| - | OR | 或 | 将指定位的状态与执行条件进行逻辑或。 | 213 |
| - | OR LD | 或装载 | 将先前各块的结果进行逻辑或。 | 214 |
| - | OR NOT | 或非 | 将指定位的反相与执行条件进行逻辑或。 | 213 |
| - | OUT | 输出 | 将执行条件为 ON 的操作位置 ON。 将执行条件为 OFF 的操作位置 OFF。 | 214 |
| - | OUT NOT | 输出非 | 将执行条件为 ON 的操作位置 OFF。 将执行条件为 OFF 的操作位置 ON(即,反相操作)。 | 214 |
| - | RSET | 复位 | 在执行条件为 ON 时将操作位置 OFF,执行条件为 OFF 时不影响操作位的状态。 | 215 |
| - | SET | 置位 | 在执行条件为 ON 时,将操作位置 ON,执行条件为 OFF 时不影响操作位的状态。 | 215 |
| - | TIM | 定时器 | ON 延迟(减量)定时器工作 | 225 |
| 00 | NOP | 不操作 | 不执行任何操作,程序转移到下一个指令。 | 218 |
| 01 | END | 结束 | 程序结束处所需要的。 | 218 |
| 02 | IL | 联锁 | 如果联锁条件为 OFF,则在 IL(02)和下一个 ILC(03)之间所有输出都变为 OFF,而所有定时器 PV 复位,其它指令被处理为 NOP;计数器 PV 保持不变 | 218 |
| 03 | ILC | 联锁清除 | | 218 |
| 04 | JMP | 跳转 | 如果跳转条件为 OFF,则 JMP(04)和相应的 JME(05)之间的所有指令都被忽略。 | 220 |
| 05 | JME | 跳转结束 | | 220 |
| 06 | @FAL | 故障报警和复位 | 产生一非致命错误,并将指定的 FAL 号输出给编程器。 | 221 |
| 07 | FALS | 严重故障报警 | 产生一致命错误,并将指定的 FAL 号输出给编程器。 | 221 |
| 08 | STEP | 步定义 | 当与控制位一起使用时,定义一新步的起动,并使先前的步复位。当使用时若无 N,定义步执行结束。 | 222 |
| 09 | SNXT | 步启动 | 与控制位一起使用,以指出步的结束,复位步并起动下一步。 | 222 |
| 10 | SFT | 移位寄存器 | 生成移位寄存器。 | 249 |
| 11 | KEEP | 保持 | 将位定义为由置位和复位输入控制的锁存器。 | 216 |
| 12 | CNTR | 可逆计数器 | 每当增量输入信号或减量输入信号分别从 OFF 变为 ON 时,PV 加 1 或减 1。 | 227 |
| 13 | DIFU | 上升微分 | 在输入信号的上升沿时,指定位在一个循环内置为 ON。 | 217 |

| 代码 | 助记符 | 名称 | 功 能 | 页 |
|---------|---------|---------|---|-----|
| 14 | DIFD | 下降微分 | 在信号后沿时,位变为 ON 一个循环。 | 217 |
| 15 | TIMH | 高速定时器 | 高速,ON 延迟(递减)计时器。 | 228 |
| 16 | (@)WSFT | 字移位 | 将起始字和结束字之间的数据以字为单位移位,零写入起始字。 | 250 |
| 17~19 | 扩展指令用。 | | | 203 |
| 20 | CMP | 比较 | 比较两字的内容,结果输出到 GR 标志、EQ 标志和 LE 标志 | 268 |
| 21 | (@)MOV | 传送 | 将源数据(字或常数)复制到目的字。 | 257 |
| 22 | (@)MVN | 传送非 | 将源数据(字或常数)反相,而后将其复制到目的字。 | 258 |
| 23 | (@)BIN | BCD→二进制 | 将源字的 4 位数 BCD 数据转换为 16 位二进制数据,而后将转换的数据输出到结果字。 | 279 |
| 24 | (@)BCD | 二进制→BCD | 将源字的二进制数据转换为 BCD,而后将转换的数据输出到结果字。 | 280 |
| 25 | (@)ASL | 算术左移 | 将数据中的单字的每一位左移 1 位,带进位。 | 251 |
| 26 | (@)ASR | 算术右移 | 将数据中的单字的每一位右移 1 位,带进位。 | 251 |
| 27 | (@)ROL | 循环左移 | 将数据中的单字的各位左转 1 位,带进位。 | 252 |
| 28 | (@)ROR | 循环右移 | 将数据中的单字的各位右转 1 位,带进位。 | 252 |
| 29 | (@)COM | 求补 | 将数据中的一字的各位状态反相。 | 360 |
| 30 | (@)ADD | BCD 加 | 将一个 4 位数 BCD 值和进位内容相加,而后将结果输出到指定的结果字。 | 305 |
| 31 | (@)SUB | BCD 减 | 将两个 4 位数 BCD 值和进位内容从另一个 4 位数 BCD 值减去,而后将结果输出到结果字。 | 306 |
| 32 | (@)MUL | BCD 乘 | 将两个 4 位数 BCD 值相乘,而后将结果输出到指定的结果字。 | 308 |
| 33 | (@)DIV | BCD 除 | 将 4 位数 BCD 被除数除以 4 位数 BCD 除数,而后将结果输出到指定的结果字。 | 309 |
| 34 | (@)ANDW | 逻辑与 | 将两个 16 位输入字进行逻辑与,如果输入字的各对应位两者都为 ON,则结果字的对应位置位。 | 360 |
| 35 | (@)ORW | 逻辑或 | 将两个 16 位输入字进行逻辑或,如果输入数据的各对应位中的一个或两个为 ON,则结果字的对应位置位。 | 361 |
| 36 | (@)XORW | 异或 | 将两个 16 位输入字进行异或,当输入字的对应位的状态不同时结果字的位置位。 | 362 |
| 37 | (@)XNRW | 异或非 | 将两个 16 位输入字进行异或非,当输入字的对应位的状态相同时结果字的位置位。 | 362 |
| 38 | (@)INC | BCD 递增 | 4 位数 BCD 字加一。 | 363 |
| 39 | (@)DEC | BCD 递减 | 4 位数 BCD 字减一。 | 363 |
| 40 | (@)STC | 设置进位 | 进位标志置位(即 CY 变为 ON)。 | 305 |
| 41 | (@)CLC | 清除进位 | 进位标志清零(即 CY 变为 OFF)。 | 305 |
| 45 | TRSM | 跟踪存储器采样 | 启动数据跟踪。 | 367 |
| 46 | (@)MSG | 消息 | 在编程器显示器上显示出 16 字符的消息。 | 369 |
| 47 和 48 | 扩展指令用。 | | | 203 |
| 50 | (@)ADB | 二进制加 | 将两个 4 位数十六进制值和进位的内容相加,而后将结果输出到指定的结果字。 | 316 |
| 51 | (@)SBB | 二进制减 | 将一个 4 位数十六进制和进位从另一个 4 位数十六进制值减去,而后将结果输出到结果字。 | 317 |
| 52 | (@)MLB | 二进制乘 | 将两个 4 位数十六进制值相乘,而后将结果输出到指定的结果字。 | 318 |

| 代码 | 助记符 | 名称 | 功 能 | 页 |
|-------|----------|--------------|--|-----|
| 53 | (@)DVB | 二进制除 | 将 4 位数十六进制被除数除以 4 位数十六进制除数,而后将结果输出到指定的结果字。 | 319 |
| 54 | (@)ADDL | 双字 BCD 加 | 将两个 8 位数值(每个 2 个字)和进位内容相加,而后将结果输出到指定的结果字。 | 311 |
| 55 | (@)SUBL | 双字 BCD 减 | 将一个 8 位数 BCD 值和进位从另一个 8 位数 BCD 值减去,结果输出到指定的结果字。 | 312 |
| 56 | (@)MULL | 双字 BCD 乘 | 两个 8 数 BCD 值相乘,结果输出到指定的结果字。 | 313 |
| 57 | (@)DIVL | 双字 BCD 除 | 8 位数 BCD 被除数除以 8 位数 BCD 除数,结果输出到指定的结果字。 | 314 |
| 58 | (@)BINL | 双字 BCD→双字二进制 | 两个连续源字的 BCD 值转换为二进制,转换后的数据输出到两个连续结果字。 | 280 |
| 59 | (@)BCDL | 双字二进制→双字 BCD | 两个连续源字的二进制值转换为 BCD,转换后的数据输出到两个连续结果字。 | 281 |
| 60-69 | 扩展指令用。 | | | 203 |
| 70 | (@)XFER | 块传送 | 若干连续源字的内容传送到连续的目的字。 | 259 |
| 71 | (@)BSET | 块设置 | 将一个字的内容或常数复制到若干连续字。 | 260 |
| 72 | (@)ROOT | 平方根 | 计算 8 位数 BCD 值的平方根,而后将舍位的 4 位数整数结果输出到指定结果字。 | 315 |
| 73 | (@)XCHG | 数据交换 | 将两个不同字的内容进行交换。 | 261 |
| 74 | (@)SLD | 左移 1 位数字 | 将起始字和结束字之间的数据左移 1 位数字(4 位)。 | 253 |
| 75 | (@)SRD | 右移 1 位数字 | 将起始字和结束字之间的数据右移 1 位数字(4 位)。 | 254 |
| 76 | (@)MLPX | 4-16 译码器 | 将源字的多达 4 个十六进制数字转换为 0~15 的十进制值,在结果字中,其位置对应于转换值的位变为 ON。 | 282 |
| 77 | (@)DMPX | 16-4 编码器 | 确定源字中最高 ON 位的位置,并将结果字中的对应位变为 ON。 | 284 |
| 78 | (@)SDEC | 七段译码器 | 将源字的十六进制值转换为七段显示的数据。 | 286 |
| 80 | (@)DIST | 单字分布 | 将源数据中的一个字传送到目的字,其地址为目的基字加偏移。 | 261 |
| 81 | (@)COLL | 数据收集 | 从源字摘录数据,而后将其写到目的字。 | 263 |
| 82 | (@)MOVVB | 传送位 | 将源字或常数中的指定位传送到目的字的指定位。 | 265 |
| 83 | (@)MOVD | 传送数字 | 将指定的 4 位源数字中的十六进制内容传送到最多达 4 个数字的指定目的数字。 | 266 |
| 84 | (@)SFTR | 可逆移位寄存器 | 将指定字或系列字的数据左移或右移。 | 254 |
| 85 | (@)TCMP | 表比较 | 将 4 位数十六进制值与由 16 个字构成的表中的值比较。 | 269 |
| 86 | (@)ASC | ASCII 转换 | 将源字的十六进制值转换为以指定起始字左半或右半处开始的 8 位 ASCII 码。 | 289 |
| 87-89 | 扩展指令用。 | | | 203 |
| 90 | (@)SEND | 网络发送。 | 在网络中向另一个节点发送数据。 | 393 |
| 91 | (@)SBS | 子程序入口 | 调用和执行子程序 N。 | 365 |
| 92 | SBN | 子程序定义 | 标志子程序 N 的起始。 | 367 |
| 93 | RET | 返回 | 标志子程序的结束,控制返回到主程序。 | 367 |
| 97 | (@)IORF | I/O 刷新 | 刷新在开始字和结束字之间的所有 I/O 字,不能在 SRM1 使用。 | 370 |
| 98 | (@)RECV | 网络接收 | 要求网络中另一节点传送数据。 | 397 |
| 99 | (@)MCRO | 宏 | 调用和执行取代 I/O 字的子程序。 | 371 |

扩展指令

下表示出了能作为扩展指令处理的指令。缺省功能代码赋予在缺省时分配代码的指令。

| 代码 | 助记符 | 名称 | 功 能 | 页 |
|-----|---------|-----------|--|-----|
| 17 | (@)ASFT | 异步移位寄存器 | 生成一个当相邻字的一个为零而另一个不为零时交换内容的移位寄存器。 | 256 |
| 18 | TKY | 十键输入 | 由 10 键键盘输入 8 位数 BCD 数据。 | 422 |
| 19 | (@)MCMP | 多字比较 | 将一块 16 个连续字与另一块 16 个连续字比较。 | 273 |
| 47 | (@)RXD | 接收 | 接收经由通信端口的数据。 | 403 |
| 48 | (@)TXD | 发送 | 发送经由通信端口的数据。 | 405 |
| 60 | CMPL | 双字比较 | 比较两个 8 位数十六进制值。 | 272 |
| 61 | (@)INI | 模式控制 | 起动和停止计数器操作,比较和改变计数器 PV 以及停止脉冲输出。 | 244 |
| 62 | (@)PRV | 读高速计数器 PV | 读计数器 PV 和高速计数器的状态数据。 | 246 |
| 63 | (@)CTBL | 比较表装载 | 登录比较表并启动高速计数器的比较。 | 232 |
| 64 | (@)SPED | 速度输出 | 以指定频率(10HZ ~ 50KHZ,以 10HZ 为单位)输出脉冲,在脉冲输出的同时可以改变输出频率。 | 382 |
| 65 | (@)PULS | 设置脉冲 | 输出指定频率的指定的脉冲数。脉冲输出直到已输出指定的脉冲数据停止。 | 380 |
| 66 | (@)SCL | 标度 | 对计算值执行标度转换。 | 293 |
| 67 | (@)BCNT | 位计数器 | 计算指定字块中为 ON 的位的总数。 | 373 |
| 68 | (@)BCMP | 块比较 | 判断一字的值是否在 16 范围(由下限和上限定义)内。 | 270 |
| 69 | (@)STIM | 间隔定时器 | 控制为实现定时中断用的间隔定时器。 | 231 |
| 87 | DSW | 数字开关输入 | 从数字开关输入 4 位数或 8 位数 BCD 数据。 | 415 |
| 88 | 7SEG | 七段显示输出 | 将 4 位或 8 位数 BCD 数据转换为 7 段显示格式,而后输出转换后的数据。 | 412 |
| 89 | (@)INT | 中断控制 | 执行中断控制,诸如屏蔽 I/O 中断的中断位或解除屏蔽。 | 379 |
| ... | (@)ACC | 加速控制 | PULS(—),ACC(—)一起控制端口 1 和 2 脉冲输出的加速和/或减速。 | 387 |
| ... | (@)ACOS | 反余弦 | 计算 32 位浮点数的反余弦。 | 354 |
| ... | (@)ADBL | 双字二进制加 | 将两个 8 位数二进制值(常规的或有符号的数据)相加,而后将结果输出到 R 和 R+1。 | 320 |
| ... | (@)APR | 算术处理 | 执行正弦、余弦、或线性近似计算。 | 332 |
| ... | (@)ASIN | 反正弦 | 计算 32 位浮点数的反正弦。 | 353 |
| ... | (@)ATAN | 反正切 | 计算 32 位浮点数的反正切。 | 355 |
| ... | AVG | 平均值 | 将指定数量的十六进制字相加,而后计算平均值,小数点后舍去,成为 4 位数。 | 329 |
| ... | (@)CMND | 发布命令 | 在网络中传送 FINS 命令到特定节点,并根据需要接收响应。 | 400 |
| ... | (@)COLM | 行→列 | 将 16 位从指定字复制到 16 连续字的位列中。 | 301 |

| 代码 | 助记符 | 名称 | 功能 | 页 |
|-----|---------|------------|---|-----|
| - | (@)COS | 余弦 | 计算角度(弧度)的余弦值并表示为 32 位浮点数。 | 351 |
| - | CPS | 有符号二进制比较 | 比较两个 16 位(4 位数)有符号二进制值,而后将结果输出到 GR、EQ 和 LE 标志。 | 274 |
| - | CPSL | 双字有符号二进制比较 | 比较两个 32 位(8 位数)有符号二进制值,结果输出到 GR、EQ 和 LE 标志。 | 275 |
| - | (@)DBS | 有符号二进制除 | 将一个 16 位有符号二进制值除以另一个值,32 位有符号二进制结果输出到 R+1 和 R。 | 325 |
| - | (@)DBSL | 双字有符号二进制除 | 将一个 32 位有符号二进制值除以另一个值,64 位有符号二进制结果输出到 R+3~R。 | 326 |
| - | (@)DEG | 弧度到角度 | 将 32 位浮点数从弧度转换为角度。 | 349 |
| - | (@)EXP | 指数 | 计算一个 32 位浮点数的自然指数(底为 e)。 | 357 |
| - | (@)FCS | FCS 计算 | 检验由上位机链接命令发送的数据中的错误。 | 373 |
| - | (@)FIX | 浮点到 16 位 | 将一个 32 位浮点数的整数部分转换成 16 位有符号二进制数据。 | 340 |
| - | (@)FIXL | 浮点到 32 位 | 将一个 32 位浮点数的整数部分转换成 32 位有符号二进制数据。 | 341 |
| - | (@)FLT | 16 位到浮点 | 将一个 16 位有符号二进制数据转换为 32 位浮点数。 | 342 |
| - | (@)FLTL | 32 位到浮点 | 将一个 32 位有符号二进制数据转换为 32 位浮点数。 | 343 |
| - | FPD | 故障点检测 | 查找指令块内的错误。 | 375 |
| - | (@)HEX | ASCH 到十六进制 | 将 ASCII 数据转换为十六进制数据。 | 290 |
| - | HKY | 十六进制键输入 | 由 16 键键盘输入多达 8 位数十六进制数据。 | 419 |
| - | (@)HMS | 秒到小时 | 将秒数据转换为小时和分数数据。 | 299 |
| - | (@)LINE | 行 | 将位列从 16 连续字复制到指定字。 | 300 |
| - | (@)LOG | 对数 | 计算一个 32 位浮点数的自然对数(底为 e)。 | 359 |
| - | (@)MAX | 查找最大值 | 查找指定数据区的最大值,并将该值输出到另一个字。 | 327 |
| - | (@)MBS | 有符号二进制乘 | 将两个字的有符号二进制内容相乘,而后 8 位数有符号二进制结果输出到 R+1 和 R。 | 323 |
| - | (@)MBSL | 双字有符号二进制乘 | 两个 32 位(8 位数)有符号二进制值相乘。而后 16 位数有符号二进制结果输出到 R+3~R。 | 324 |
| - | (@)MIN | 查找最小值 | 查找指定数据区的最小值,并将该值输出到另一个字。 | 328 |
| - | (@)NEG | 二进制补码 | 将源字的 4 位数十六进制内容转换为二进制补码,而结果输出到 R。 | 302 |
| - | (@)NEGL | 双字二进制补码 | 将源字的 8 位数十六进制内容转换为二进制补码,结果输出到 R 和 R+1。 | 303 |
| - | PID | PID 控制 | 根据指定参数执行 PID 控制。 | 392 |
| - | (@)PLS2 | 脉冲输出 | 以指定速度从 0 到目标频率加速脉冲输出和以同一速度减速。 | 385 |
| - | (@)PMCR | 协议宏 | 执行登录在串行通信板上的指定通信序列(协议数据) | 410 |
| ... | (@)PWM | 可变占空率脉冲 | 从端口 1 或 2 输出具有指定占空率(0%~99%)脉冲。 | 389 |
| - | (@)RAD | 角度到弧度 | 将一个 32 位浮点数从角度转换为弧度。 | 348 |
| - | (@)SBBL | 双字二进制减 | 将一个 8 位数二进制值(正常的或有符号的数据)从另一个减去,结果输出到 R 和 R+1。 | 321 |

| 代码 | 助记符 | 名称 | 功 能 | 页 |
|-----|---------|----------------|--|-----|
| ... | (@)SCL2 | 有符号二进制到 BCD 标度 | 将 4 位数有符号十六进制值线性转换为 4 位数 BCD 值。 | 295 |
| ... | (@)SCL3 | BCD 到有符号二进制标度 | 将 4 位数 BCD 值线性转换为 4 位数有符号十六进制值。 | 296 |
| ... | (@)SEC | 小时到秒 | 将小时和分数据转换为秒数据。 | 298 |
| ... | (@)SIN | 正弦 | 计算一个角(弧度)的正弦并表示为一个 32 位浮点数。 | 350 |
| ... | (@)SQRT | 平方根 | 计算一个 32 位浮点数的平方根。 | 356 |
| ... | (@)SRCH | 数据查找 | 在存储器中的指定范围内查找指定数据。输出含有该数据的字范围中的字地址。 | 390 |
| ... | (@)STUP | 更改串行端口设置 | 改变指定端口在 PC 设置中的通信参数。 | 407 |
| ... | (@)SUM | 求和计算 | 计算存储器中指定范围内各字内容之和。 | 330 |
| ... | (@)TAN | 正切 | 计算一个角(弧度)的正切值并表示为 32 位浮点数。 | 352 |
| ... | (@)TTIM | 累计定时器 | 创建一个在 0.1 和 999.9s 之间的以 0.1s 为单位 PV 递增的定时器。 | 229 |
| ... | (@)XFRB | 传送位 | 将多达 255 指定源位的状态复制到指定目的位。 | 267 |
| ... | ZCP | 区域范围比较 | 将一字与由下限和上限定义的一范围比较,结果输出到 GR、EQ 和 LE 标志。 | 277 |
| ... | ZCPL | 双字区域范围比较 | 将一 8 位数值与由下限和上限定义的一范围比较,结果输出到 GR、EQ 和 LE 标志。 | 278 |
| ... | (@)+F | 浮点数加 | 将两个 32 位浮点数相加。 | 343 |
| ... | (@)-F | 浮点数减 | 从一个 32 位浮点数减去另一个。 | 345 |
| ... | (@)*F | 浮点数乘 | 两个 32 位浮点数相乘。 | 346 |
| ... | (@)/F | 浮点数除 | 一个 32 位浮点数除以另一个。 | 347 |

附录 B

错误和算术标志操作

下表显示会对 OF、UF、ER、CY、GR、LE 和 EQ 标志起作用的指令。

通常，OF 表示一个 16 位的计算结果大于 32,767(7FFF) 或一个 32 位的计算结果大于 2,147,483,647(7FFF FFFF)。UF 表示一个 16 位的计算结果小于 -32,768(8000) 或一个 32 位的计算结果小于 -2,147,483,648(8000 0000)。详情参见第五章 指令集。

ER 表示操作数数据不在所需要之内。CY 表示算术或数据移位结果。GR 表示比较值是大于某标准值，LE 表示是小于，而 EQ 表示是相同。EQ 还表示算术运算的结果是零。详情请参阅第五章 指令集。

表中的垂直箭头表示会根据指令的结果变为 ON 和 OFF 的标志。虽然梯形图指令 TIM 和 CNT 会在 ER 为 ON 时执行，但在 ER 栏中具有垂直箭头的其它指令在 ER 为 ON 时不会执行。下表中的所有其它标志在 ER 为 ON 时也不会工作。

未示出的指令不会对表中的任何标志起作用。虽然仅仅示出了各指令的非微分形式，微分指令对标志所起的作用实际上是一样的。

当 END(01) 执行时，7 个标志全为 OFF，所以它们的状态不能用编程器监视。

| 助记符 | 25503(ER) | 25504(CY) | 25505(GR) | 25506(EQ) | 25507(LE) | 25404(OF) | 25405(UF) | 页 |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| TIM | ↓ | ... | ... | ... | ... | ... | ... | 225 |
| CNT | ↓ | ... | ... | ... | ... | ... | ... | 226 |
| END(01) | OFF | OFF | OFF | OFF | OFF | OFF | OFF | 218 |
| CNTR(12) | ↓ | ... | ... | ... | ... | ... | ... | 227 |
| TIMH(15) | ↓ | ... | ... | ... | ... | ... | ... | 228 |
| WSFT(16) | ↓ | ... | ... | ... | ... | ... | ... | 250 |
| CMP(20) | ↓ | ... | ↓ | ↓ | ↓ | ... | ... | 268 |
| MOV(21) | ↓ | ... | ... | ↓ | ... | ... | ... | 257 |
| MVN(22) | ↓ | ... | ... | ↓ | ... | ... | ... | 258 |
| BIN(23) | ↓ | ... | ... | ↓ | ... | ... | ... | 279 |
| BCD(24) | ↓ | ... | ... | ↓ | ... | ... | ... | 280 |
| ASL(25) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 251 |
| ASR(26) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 251 |
| ROL(27) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 252 |
| ROR(28) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 252 |
| COM(29) | ↓ | ... | ... | ↓ | ... | ... | ... | 360 |
| ADD(30) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 305 |
| SUB(31) | ↓ | ↓ | ... | ↓ | ... | ... | ... | 306 |
| MUL(32) | ↓ | ... | ... | ↓ | ... | ... | ... | 308 |
| DIV(33) | ↓ | ... | ... | ↓ | ... | ... | ... | 309 |
| ANDW(34) | ↓ | ... | ... | ↓ | ... | ... | ... | 360 |
| ORW(35) | ↓ | ... | ... | ↓ | ... | ... | ... | 361 |
| XORW(36) | ↓ | ... | ... | ↓ | ... | ... | ... | 362 |
| XNRW(37) | ↓ | ... | ... | ↓ | ... | ... | ... | 362 |
| INC(38) | ↓ | ... | ... | ↓ | ... | ... | ... | 363 |
| DEC(39) | ↓ | ... | ... | ↓ | ... | ... | ... | 363 |

| 助记符 | 25503(ER) | 25504(CY) | 25505(GR) | 25506(EQ) | 25507(LE) | 25404(OF) | 25405(UF) | 页 |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| STC(40) | ... | ON | ... | ... | ... | ... | ... | 305 |
| CLD(41) | ... | ... | ... | ... | ... | ... | ... | 305 |
| MSG(46) | ↑ | ... | ... | ... | ... | ... | ... | 369 |
| ADB(50) | ↑ | ↓ | ... | ↑ | ... | ↑ | ↓ | 316 |
| SBB(51) | ↑ | ↑ | ... | ↑ | ... | ↑ | ↑ | 317 |
| MLB(52) | ↑ | ... | ... | ↑ | ... | ... | ... | 318 |
| DVB(53) | ↑ | ... | ... | ↑ | ... | ... | ... | 319 |
| ADDL(54) | ↑ | ↑ | ... | ↑ | ... | ... | ... | 311 |
| SUBL(55) | ↑ | ↓ | ... | ↑ | ... | ... | ... | 312 |
| MULL(56) | ↑ | ... | ... | ↑ | ... | ... | ... | 313 |
| DIVL(57) | ↑ | ... | ... | ↑ | ... | ... | ... | 314 |
| BINL(58) | ↑ | ... | ... | ↑ | ... | ... | ... | 280 |
| BCDL(59) | ↑ | ... | ... | ↑ | ... | ... | ... | 281 |
| XFER(70) | ↑ | ... | ... | ... | ... | ... | ... | 259 |
| BSET(71) | ↑ | ... | ... | ... | ... | ... | ... | 260 |
| ROOT(72) | ↑ | ... | ... | ↑ | ... | ... | ... | 315 |
| XCHG(73) | ↑ | ... | ... | ... | ... | ... | ... | 261 |
| SLD(74) | ↑ | ... | ... | ... | ... | ... | ... | 253 |
| SRD(75) | ↑ | ... | ... | ... | ... | ... | ... | 254 |
| MLPX(76) | ↑ | ... | ... | ... | ... | ... | ... | 282 |
| DMPX(77) | ↑ | ... | ... | ... | ... | ... | ... | 284 |
| SDEC(78) | ↑ | ... | ... | ... | ... | ... | ... | 286 |
| DIST(80) | ↑ | ... | ... | ... | ↑ | ... | ... | 261 |
| COLL(81) | ↑ | ... | ... | ... | ↑ | ... | ... | 263 |
| MOVB(82) | ↑ | ... | ... | ... | ... | ... | ... | 265 |
| MOVD(83) | ↑ | ... | ... | ... | ... | ... | ... | 266 |
| SFTR(84) | ↑ | ↑ | ... | ... | ... | ... | ... | 254 |
| TCMP(85) | ↑ | ... | ... | ↑ | ... | ... | ... | 269 |
| ASC(86) | ↑ | ... | ... | ... | ... | ... | ... | 289 |
| SEND(90) | ↑ | ... | ... | ... | ... | ... | ... | 393 |
| SBS(91) | ↑ | ... | ... | ... | ... | ... | ... | 365 |
| SBN(92) | OFF | OFF | OFF | OFF | OFF | OFF | OFF | 367 |
| RECV(98) | ↑ | ... | ... | ... | ... | ... | ... | 397 |
| MCRO(99) | ↑ | ... | ... | ... | ... | ... | ... | 371 |

扩展指令

缺省功能代码显示在有缺省功能代码的指令上。

| 助记符 | 25503(ER) | 25504(CY) | 25505(GR) | 25506(EQ) | 25507(LE) | 25404(OF) | 25405(UF) | 页 |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| 7SGE(88) | ↑ | ... | ... | ... | ... | ... | ... | 412 |
| ACC(—) | ↑ | ... | ... | ... | ... | ... | ... | 387 |
| ACOS(—) | ↑ | ... | ... | ↑ | ... | OFF | OFF | 354 |
| ADBL(—) | ↑ | ↑ | ... | ↑ | ... | ↑ | ↑ | 320 |

| 助记符 | 25503(ER) | 25504(CY) | 25505(GR) | 25506(EQ) | 25507(LE) | 25404(OF) | 25405(UF) | 页 |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| APR(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 332 |
| ASFT(17) | ↑ | ... | ... | ... | ... | ... | ... | 256 |
| ASIN(—) | ↑ | ... | ... | ↑ | ... | OFF | OFF | 353 |
| ATAN(—) | ↑ | ... | ... | ↑ | ... | OFF | OFF | 355 |
| AVG(—) | ↑ | ... | ... | ... | ... | ... | ... | 329 |
| BCMP(68) | ↑ | ... | ... | ... | ... | ... | ... | 270 |
| BCNT(67) | ↑ | ... | ... | ↑ | ... | ... | ... | 373 |
| CMND(—) | ↑ | ... | ... | ... | ... | ... | ... | 400 |
| CMPL(60) | ↑ | ... | ↑ | ↑ | ↑ | ... | ... | 272 |
| COLM(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 301 |
| COS(—) | ↑ | ... | ... | ↑ | ... | OFF | OFF | 351 |
| CPS(—) | ↑ | ... | ↑ | ↑ | ↑ | ... | ... | 274 |
| CPSL(—) | ↑ | ... | ↑ | ↑ | ↑ | ... | ... | 275 |
| CTBL(63) | ↑ | ... | ... | ... | ... | ... | ... | 232 |
| DBS(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 325 |
| DBSL(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 326 |
| DEG(—) | ↑ | ... | ... | ↑ | ... | ↑ | ↑ | 349 |
| EXP(—) | ↑ | ... | ... | ↑ | ... | ↑ | ↑ | 357 |
| DSW(87) | ↑ | ... | ... | ... | ... | ... | ... | 415 |
| FCS(—) | ↑ | ... | ... | ... | ... | ... | ... | 373 |
| FIX(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 340 |
| FIXL(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 341 |
| FLT(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 342 |
| FLTL(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 343 |
| FPD(—) | ↑ | ↑ | ... | ... | ... | ... | ... | 375 |
| HEX(—) | ↑ | ... | ... | ... | ... | ... | ... | 290 |
| HKY(—) | ↑ | ... | ... | ... | ... | ... | ... | 419 |
| HMS(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 299 |
| INI(61) | ↑ | ... | ... | ... | ... | ... | ... | 244 |
| INT(89) | ↑ | ... | ... | ... | ... | ... | ... | 379 |
| LINE(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 300 |
| LOG(—) | ↑ | ... | ... | ↑ | ... | ↑ | OFF | 359 |
| MAX(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 327 |
| MBS(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 323 |
| MBSL(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 324 |
| MCMP(19) | ↑ | ... | ... | ↑ | ... | ... | ... | 328 |
| MIN(—) | ↑ | ... | ... | ↑ | ... | ... | ... | 328 |
| NEG(—) | ↑ | ... | ... | ↑ | ... | ... | ↑ | 302 |
| NEGL(—) | ↑ | ... | ... | ↑ | ... | ... | ↑ | 303 |
| PID(—) | ↑ | ↑ | ... | ... | ... | ... | ... | 309 |
| PLS2(—) | ↑ | ... | ... | ... | ... | ... | ... | 385 |
| PMCR(—) | ↑ | ... | ... | ... | ... | ... | ... | 410 |
| PRV(62) | ↑ | ... | ... | ... | ... | ... | ... | 246 |
| PULS(65) | ↑ | ... | ... | ... | ... | ... | ... | 380 |

| 助记符 | 25503(ER) | 25504(CY) | 25505(GR) | 25506(EQ) | 25507(LE) | 25404(OF) | 25405(UF) | 页 |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----|
| PWM(—) | ‡ | ... | ... | ... | ... | ... | ... | 389 |
| RAD(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 348 |
| RXD(47) | ‡ | ... | ... | ... | ... | ... | ... | 403 |
| SBBL(—) | ‡ | ‡ | ... | ‡ | ... | ‡ | ‡ | 321 |
| SCL(66) | ‡ | ... | ... | ‡ | ... | ... | ... | 293 |
| SCL2(—) | ‡ | ‡ | ... | ‡ | ... | ... | ... | 295 |
| SCL3(—) | ‡ | ... | ... | ‡ | ... | ... | ... | 296 |
| SEC(—) | ‡ | ... | ... | ‡ | ... | ... | ... | 298 |
| SIN(—) | ‡ | ... | ... | ‡ | ... | OFF | OFF | 350 |
| SPEED(64) | ‡ | ... | ... | ... | ... | ... | ... | 382 |
| SQRT(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 356 |
| SRCH(—) | ‡ | ... | ... | ‡ | ... | ... | ... | 390 |
| STIM(69) | ‡ | ... | ... | ... | ... | ... | ... | 231 |
| STUP(—) | ‡ | ... | ... | ... | ... | ... | ... | 407 |
| SUM(—) | ‡ | ... | ... | ‡ | ... | ... | ... | 330 |
| TAN(—) | ‡ | ... | ... | ‡ | ... | OFF | OFF | 352 |
| TKY(18) | ‡ | ... | ... | ... | ... | ... | ... | 422 |
| TTIM(—) | ‡ | ... | ... | ... | ... | ... | ... | 229 |
| TXD(48) | ‡ | ... | ... | ... | ... | ... | ... | 405 |
| XFRB(—) | ‡ | ... | ... | ... | ... | ... | ... | 267 |
| ZCP(—) | ‡ | ... | ‡ | ‡ | ‡ | ... | ... | 277 |
| ZCPL(—) | ‡ | ... | ‡ | ‡ | ‡ | ... | ... | 278 |
| + F(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 343 |
| - F(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 345 |
| * F(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 346 |
| /F(—) | ‡ | ... | ... | ‡ | ... | ‡ | ‡ | 347 |

附录 C

存储器区

存储器区结构

下表示出了 CQM1H 可使用的存储器区。

| 数据区 | | 长度 | 字 | 位 | 功 能 |
|---------------------|------|-----------------------|----------------------|--|--|
| IR 区(注 1) | 输入区 | 256 位 | IR000 ~ IR015 | IR00000 ~ IR01515 | 输入位可分配给输入单元或 I/O 单元。IR000 中的 16 位总是分配给 CPU 单元的内部输入。 |
| | 输出区 | 256 位 | IR100 ~ IR115 | IR10000 ~ IR11515 | 输出位可分配给输出单元或 I/O 单元。 |
| | 工作区 | 最少 2528 位 (注 2) | IR016 ~ IR089 | IR01600 ~ IR08915 | 工作位不具有任何特定功能。它们能在程序中自由使用。 |
| | | | IR116 ~ IR189 | IR11600 ~ IR18915 | |
| IR216 ~ IR219 | | | IR21600 ~ IR21915 | | |
| Controller Link 状态区 | 96 位 | IR090 ~ IR095 | IR09000 ~ IR09615 | 用于指示 Controller Link 数据链接状态信息。(未安装 Controller Link 单元时可用作工作位。) | |
| | 96 位 | IR190 ~ IR195 | IR19000 ~ IR19615 | 用于指示 Controller Link 错误和网络参与信息(未安装 Controller Link 时可用作工作位。) | |
| 宏操作数区 (注 1) | 输入区 | 64 位 | IR096 ~ IR099 | IR09600 ~ IR09915 | 在使用宏指令 MCRO(99)时使用。(在未使用宏指令时可用作工作位。) |
| | 输出区 | 64 位 | IR196 ~ IR199 | IR19600 ~ IR19915 | |
| 内板槽 1 区 | | 256 位 | IR200 ~ IR215 | IR20000 ~ IR21515 | 这些位可分配给安装在 CQM1H—CPU51/61 的槽 1 上的内板。(使用 CQM1H—CPU11/CPU21 或槽 1 为空时可用作工作位。) CQM1H—CTB41 高速计数器板: IR200 ~ IR213(14 字):被板使用 IR214 和 IR215(2 字):未用。 CQM1H—SCB41 串行通信板: IR200 ~ IR207(8 字):被板使用 IR208 ~ IR215(8 字):未用。 |
| 模拟设定区(注 1) | | 64 位 | IR220 ~ IR223 | IR22000 ~ IR22315 | 安装了 CQM1H—AVB41 模拟设定板时用于存储模拟设定。(未安装模拟设定板时可用作工作位。) |
| 高速计数器 0 PV(注 1) | | 32 位 | IR320 ~ IR231 | IR23000 ~ IR23115 | 用于存储内置高速计数器(高速计数器 0)的当前值。(未使用高速计数器 0 时可用作工作位。) |

| 数据区 | 长度 | 字 | 位 | 功能 | |
|---------------|------------|------------------------------------|-------------------|---|---|
| 内板槽 2 区 | 192 位 | IR232 ~ IR243 | IR23200 ~ IR24315 | 这些位可分配给安装在 CQM1H—CPU51/61 的槽 2 上的内板。(使用了 CQM1H—CPU11/21 或槽 2 为空时可用作工作位。) CQM1H—CTB41 高速计数器板: IR232 ~ IR243(12 字):被板使用 CQM1H—PLB 21 脉冲 I/O 板: IRD232 ~ IR239(8 字):被板使用 IR240 ~ IR243(4 字):未用 CQM1H—ABB21:绝对编码器接 0 板: IR232 ~ IR239(8 字):被板使用 IR240 ~ IR243(4 字):未用 CQM1H—MAB42 模拟量 I/O 板: IR232 ~ IR239(8 字):被板使用 IR240 ~ IR243(4 字):未用 | |
| SR 区 | 184 位 | SR244 ~ SR255 | SR24400 ~ SR25507 | 这些位用于专用功能,例如标志和控制位。 | |
| HR 区 | 1,600 位 | HR00 ~ HR99 | HR0000 ~ HR9915 | 这些位在电源开时存储数据并保持它们的 ON/OFF 状态。 | |
| AR 区 | 448 位 | AR00 ~ AR27 | AR0000 ~ AR2715 | 这些位用于专用功能,例如标志和控制位。 | |
| TR 区 | 8 位 | ... | TR0 ~ TR7 | 这些位用来在程序分支时暂时存储 ON/OFF 状态。 | |
| LR 区(注 1) | 1,024 位 | LR00 ~ LR63 | LR0000 ~ LR6315 | 用于经过 RS—232 端口或 Controller Link 单元的 1:1 数据链路。 | |
| 定时器/计数器区(注 3) | 512 位 | TIM/CNT000 ~ TIM/CNT511 (定时器/计数器号) | | 定时器和计数器所用的号相同。在使用 TIMH(15)时,定时器号 000 ~ 015 可以中断刷新以保证在长循环中的正确计时。 | |
| DM 区 | 读/写 | 3,072 字 | DM0000 ~ DM3071 | ... | DM 区数据只能以字为单位存取。当电源被断开时字值保持。 |
| | | 3,072 字 | DM3072 ~ DM6143 | ... | 只存在于 CQM1H—CPU51/61 CPU 单元中。 |
| | 只读(注 4) | 425 字 | DM6144 ~ DM6568 | ... | 不能由程序重写(只有编程设备)。 DM 6400 ~ DM 6409(10 字): Controller Link DM 参数区 DM6450 ~ DM6499(50 字): 路由表区 DM 6550 ~ DM 6559(10 字): 串行通信板设定 |
| | 错误日志区(注 4) | 31 字 | DM6569 ~ DM6599 | ... | 用于存储发生错误的发生时间和错误代码。 |
| | PC 设置(注 4) | 56 字 | DM6600 ~ DM6655 | ... | 用于存储对 PC 操作进行控制和各种参数。 |
| EM 区 | 6,144 字 | EM0000 ~ EM6143 | ... | EM 区数据只能以字为单位存取。当电源被断开时字值保持。 仅存在于 CQM1H—CPU61 CPU 单元中。 | |

- 注
1. 未被用于其分配功能的 IR 位和 LR 位可以用作工作区。
 2. 至少 2,528 位能用作工作位。因为未被用于分配功能的其它位可用作工作位,所以能用的位的总数取决于 PC 系统的配置。
 3. 在访问 PV 时, TIM/CNT 号被用作字数据,在访问完成标志时,它们被用作位数据。
 4. DM6144 ~ DM6655 中的数据不能由程序重写。

IR 区

用于槽 1 内板的标志位(IR200 ~ IR215)

串行通信板标志/位

| 字 | 位 | 功 能 | 通信模式 | |
|-------|-----------|--|----------------|-----|
| IR200 | 00 | 串行通信板硬件错误标志 | 所有模式 | |
| | 01 | 端口标识错误标志(硬件错误) | | |
| | 02 | 协议数据错误标志 | 协议宏 | |
| | 03 ~ 10 | 未用 | | |
| | 11 | 端口 2 协议宏执行错误标志 | | |
| | 12 | 端口 1 协议宏执行错误标志 | 所有模式 | |
| | 13 | 端口 2 PC 设置错误标志 | | |
| | 14 | 端口 1 PC 设置错误标志 | | |
| 15 | PC 设置错误标志 | | | |
| IR201 | 00 ~ 03 | 端口 1 错误代码 0:操作正常; 1:校验错误; 2:帧错误; 3:过运行错误; 4:FCS 错误; 5:超时错误; 6:校验和错误; 7:命令错误 | 所有模式 | |
| | 04 | 通信错误标志 | | |
| | 05 | 传输允许标志 | 上位机链接或无协议 | |
| | 06 | 接收完成标志 | | |
| | 07 | 接收溢出标志 | | |
| | | | 序列中止完成标志 | 协议宏 |
| | 08 ~ 11 | 端口 2 错误代码 0:操作正常; 1:校验错误; 2:帧错误; 3:过运行错误; 4:FCS 错误; 5:超时错误; 6:校验和错误; 7:命令错误 | 所有模式 | |
| | 12 | 通信错误标志 | | |
| | 13 | 传输允许标志 | 上位机链接或无协议 | |
| | 14 | 接收完成标志 | | |
| | 15 | 接收溢出标志 | | |
| | | | 序列中止完成标志 | 协议宏 |
| IR202 | 00 ~ 07 | 端口 1 有 PT 标志的通信(位 00 ~ 07 = PT0 ~ 7) 重复计数器 PV(00 ~ FF 十六进制) | 1:N 模式中的 NT 链接 | |
| | 00 ~ 15 | 接收计数器(4 位数 BCD) | 协议宏 | |
| IR203 | 00 ~ 07 | 端口 2 有 PT 标志的通信(位 00 ~ 07 = PT0 ~ 7) 重复计数器 PV(00 ~ FF 十六进制) | 1:N 模式中的 NT 链接 | |
| | 00 ~ 15 | 接收计数器(4 位数 BCD) | 协议宏 | |
| IR204 | 00 | 端口 1 跟踪标志 | 协议宏 | |
| | 01 | 端口 2 | | |
| | 02 ~ 07 | 未用 | | |
| | 08 ~ 11 | 端口 1 协议宏错误代码 0:正常操作 1:无协议宏函数 2:序列号错误 3:接收数据写数据溢出 4:协议数据语法错误 5:端口初始化时执行协议宏 | | |
| | 12 ~ 15 | 端口 2 | | |
| IR205 | 00 ~ 03 | 端口 1 完成接收事件号 | 协议宏 | |
| | 04 ~ 07 | 完成步号 | | |
| | 08 ~ 14 | 未用 | | |
| | 15 | IR20408 ~ IR20411 数据存储标志 0:无数据存储; 1:数据存储 | | |

| 字 | 位 | 功 能 | | 通信模式 |
|--------------------|---------|------|---|---------|
| IR206 | 00 ~ 03 | 端口 2 | 完成接收事件号 | 协议宏 |
| | 04 ~ 07 | | 完成步号 | |
| | 08 ~ 14 | | 未用 | |
| | 15 | | IR20412 IR20415 数据存储标志 0: 无数据存储; 1: 数据存储 | |
| IR207 | 00 | 端口 1 | 串行通信端口重启动位 | 所有模式 |
| | 01 | 端口 2 | | |
| | 02 | 端口 1 | 持续跟踪开始/停止位 | 协议宏 |
| | 03 | 端口 2 | | |
| | 04 | 端口 1 | 短跟踪开始/停止位 | 协议宏 |
| | 05 | 端口 2 | | |
| | 06 ~ 07 | 未用 | | 无协议或协议宏 |
| | 08 | 端口 1 | 协议宏执行标志 | |
| | 09 | 端口 1 | 步错误过程标志 | 协议宏 |
| | 10 | | 序列结束完成标志 | |
| | 11 | | 强制中止位 | |
| | 12 | 端口 2 | 协议宏执行标志 | 无协议或协议宏 |
| | 13 | | 步错误过程标志 | |
| | 14 | | 序列结束完成标志 | |
| | 15 | | 强制中止位 | |
| IR 208 ~ IR 215 | 00 ~ 15 | 未用 | | ... |

高速计数器板标志/位

| 字 | 位 | 名 称 | | 功 能 |
|----------------|---------|---------------------|-------------|--|
| IR200 | 00 ~ 15 | 高速计数器 1 | PV(最右 4 位数) | 包含每个高速计数器板的端口的高速计数器 PV。 注 可在 PC 设置(DM6602)中设定 PV 数据格式(BCD 或十六进制)。 |
| IR201 | | | PV(最左 4 位数) | |
| IR202 | 00 ~ 15 | 高速计数器 2 | PV(最右 4 位数) | |
| IR203 | | | PV(最左 4 位数) | |
| IR204 | 00 ~ 15 | 高速计数器 3 | PV(最右 4 位数) | |
| IR205 | | | PV(最左 4 位数) | |
| IR206 | 00 ~ 15 | 高速计数器 4 | PV(最右 4 位数) | |
| IR207 | | | PV(最左 4 位数) | |
| IR208(高速计数器 1) | 00 ~ 07 | 比较结果: 内部输出位 | | 满足条件时通过 CTBL(一)中的操作数确定包含的位型式。 |
| | 08 ~ 11 | 比较结果: 输出 1~4 的外部输出位 | | 满足条件时通过 CTBL(一)中的操作数确定包含的位型式。 |
| IR209(最速计数器 2) | 12 | 计数器操作标志 | | 0: 停止 1: 工作 |
| IR210(高速计数器 3) | 13 | 比较标志 | | 指明是否在比较 0: 停止; 1: 工作 |
| IR211(高速计数器 4) | 14 | PV 下溢/下溢标志 | | 0: 正常 1: 出现上溢或下溢 |
| | 15 | SV 错误标志 | | 0: 正常 1: 出现 SV 错误 |

| 字 | 位 | 名称 | 功能 | |
|-------|-------|---------------|------------------------------------|------------|
| IR212 | 00 | 高速计数器 1 复位位 | Z 相和软件复位 | |
| | 01 | 高速计数器 2 复位位 | 0:计数器在 Z 相不复位 1:计数器在 Z 相复位 | |
| | 02 | 高速计数器 3 复位位 | 仅软件复位 | |
| | 03 | 高速计数器 4 复位位 | 0:计数器不复位 0→1:计数器复位 | |
| | 04~07 | 未用 | | |
| | 08 | 高速计数器 1 比较停止位 | 0→1:启动比较 | |
| | 09 | 高速计数器 2 比较停止位 | 1→0:停止比较 | |
| | 10 | 高速计数器 3 比较停止位 | | |
| | 11 | 高速计数器 4 比较停止位 | | |
| | 12 | 高速计数器 1 停止位 | 0:持续工作 | |
| | 13 | 高速计数器 2 停止位 | 1:停止工作 | |
| | 14 | 高速计数器 3 停止位 | | |
| | 15 | 高速计数器 4 停止位 | | |
| | IR213 | 00 | 外部输出 1 强制置位位 | 0:对输出状态无影响 |
| | | 01 | 外部输出 2 强制置位位 | 1:强制输出 ON |
| 02 | | 外部输出 3 强制置位位 | | |
| 03 | | 外部输出 4 强制置位位 | | |
| 04 | | 外部输出强制置位允许位 | 1:输出 1~4 强制置位允许 0:输出 1~4 强制置位禁止 | |
| 05~15 | | 未用 | | |

模拟设定板(槽 1 和 2)标志/位

| 字 | 位 | 功能 |
|--------|-------|-----------------------------|
| IR 220 | 00~15 | 模拟 SV 1:0000~0200(4 位数 BCD) |
| IR 221 | 00~15 | 模拟 SV 2:0000~0200(4 位数 BCD) |
| IR 222 | 00~15 | 模拟 SV 3:0000~0200(4 位数 BCD) |
| IR 223 | 00~15 | 模拟 SV 4:0000~0200(4 位数 BCD) |

槽 2 中的内板标志/位(IR232~IR243)

高速计数器板标志/位

| 字 | 位 | 名称 | 功能 |
|----------------|-------|------------------------|--|
| IR232 | 00~15 | 高速计数器 1 PV(最右 4 位数) | 包含每个高速计数器板的端口的高速计数器 PV。 注 可在 PC 设置(DM6602)中设定 PV 数据格式(BCD 或十六进制)。 |
| IR233 | 00~15 | | |
| IR234 | 00~15 | 高速计数器 2 PV(最右 4 位数) | |
| IR235 | 00~15 | | |
| IR236 | 00~15 | 高速计数器 3 PV(最右 4 位数) | |
| IR237 | 00~15 | | |
| IR238 | 00~15 | 高速计数器 4 PV(最右 4 位数) | |
| IR239 | 00~15 | | |
| IR240(高速计数器 1) | 00~07 | 比较结果:内部输出位 | 满足条件时通过 CTBL(一)中的操作数确定包含的位型式。 |
| IR241(高速计数器 2) | 08~11 | 比较结果:输出 1~4 的外部输出位 | 满足条件时通过 CTBL(一)中的操作数确定包含的位型式。 |
| IR242(高速计数器 3) | 12 | 计数器操作标志 | 0:停止 1:工作 |
| IR243(高速计数器 4) | 13 | 比较标志 | 指明是否在进行比较 0:停止; 1:工作 |
| | 14 | PV 上溢/下溢标志 | 0:正常 1:出现上溢或下溢 |
| | 15 | SV 错误标志 | 0:正常 1:出现 SV 错误 |

| 字 | 位 | 名称 | 功能 |
|------|-------|---------------|------------------------------------|
| AR05 | 00 | 高速计数器 1 复位位 | Z 相和软件复位 |
| | 01 | 高速计数器 2 复位位 | 0:Z 相复位禁止 1:Z 相复位允许 |
| | 02 | 高速计数器 3 复位位 | 仅软件复位 |
| | 03 | 高速计数器 4 复位位 | 0:软件复位禁止 0→1:执行软件复位 |
| | 04~07 | 未用 | |
| | 08 | 高速计数器 1 比较停止位 | 0→1:启动比较 |
| | 09 | 高速计数器 2 比较停止位 | 1→0:停止比较 |
| | 10 | 高速计数器 3 比较停止位 | |
| | 11 | 高速计数器 4 比较停止位 | |
| | 12 | 高速计数器 1 停止位 | 0:持续工作 |
| | 13 | 高速计数器 2 停止位 | 1:停止工作 |
| | 14 | 高速计数器 3 停止位 | |
| | 15 | 高速计数器 4 停止位 | |
| AR06 | 00 | 外部输出 1 强制置位位 | 0:对输出状态无影响 |
| | 01 | 外部输出 2 强制置位位 | 1:强制输出 ON |
| | 02 | 外部输出 3 强制置位位 | |
| | 03 | 外部输出 4 强制置位位 | |
| | 04 | 外部输出强制置位允许位 | 1:输出 1~4 强制置位允许 0:输出 1~4 强制置位禁止 |
| | 05~15 | 未用 | |

脉冲 I/O 板标志/位

| 字 | 位 | 功能 |
|-------------|-------|-----------------------|
| IR232 | 00~15 | 高速计数器 1PV(最右 4 位数) |
| IR233 | 00~15 | 高速计数器 1PV(最左 4 位数) |
| IR234 | 00~15 | 高速计数器 2PV(最右 4 位数) |
| IR235 | 00~15 | 高速计数器 2PV(最左 4 位数) |
| IR236 | 00~15 | 端口 1 脉冲输出 PV(最右 4 位数) |
| IR237 | 00~15 | 端口 1 脉冲输出 PV(最左 4 位数) |
| IR238 | 00~15 | 端口 2 脉冲输出 PV(最右 4 位数) |
| IR239 | 00~15 | 端口 2 脉冲输出 PV(最左 4 位数) |
| IR240~IR243 | 00~15 | 未用 |

绝对编码器接口板标志/位

| 字 | 位 | 功能 |
|---------------|-------|--------------------------|
| IR232 | 00~15 | 绝对编码器高速计数器 1 PV(最右 4 位数) |
| IR233 | 00~15 | 绝对编码器高速计数器 1 PV(最左 4 位数) |
| IR234 | 00~15 | 绝对编码器高速计数器 2 PV(最右 4 位数) |
| IR235 | 00~15 | 绝对编码器高速计数器 2 PV(最左 4 位数) |
| IR 236~IR 243 | 00~15 | 未用 |

模拟量 I/O 板标志/位

| 字 | 位 | 功能 |
|-------------|-------|-------------|
| IR232 | 00~15 | 模拟量输入 1 转换值 |
| IR233 | 00~15 | 模拟量输入 2 转换值 |
| IR234 | 00~15 | 模拟量输入 3 转换值 |
| IR235 | 00~15 | 模拟量输入 4 转换值 |
| IR236 | 00~15 | 模拟量输出 1SV |
| IR237 | 00~15 | 模拟量输出 2SV |
| IR236~IR243 | 00~15 | 未用 |

模拟设定板(槽 1 和 2)标志/位

| 字 | 位 | 功 能 |
|--------|---------|------------------------------|
| IR 220 | 00 ~ 15 | 模拟 SV1:0000 ~ 0200(4 位数 BCD) |
| IR 221 | 00 ~ 15 | 模拟 SV2:0000 ~ 0200(4 位数 BCD) |
| IR 222 | 00 ~ 15 | 模拟 SV3:0000 ~ 0200(4 位数 BCD) |
| IR 223 | 00 ~ 15 | 模拟 SV4:0000 ~ 0200(4 位数 BCD) |

通信单元标志/位

Controller Link 状态区 1(IR090 ~ IR095)

| 字 | 位 | 功 能 |
|--------|---------|--|
| IR 090 | 00 ~ 14 | 总为 0 |
| | 15 | 本地节点的数据链接参与状态 0:本地节点不在数据链接中或数据链接停止。 1:本地节点参加了数据链接。 |
| IR 091 | 00 ~ 07 | 数据链接状态:节点 1 |
| | 08 ~ 15 | 数据链接状态:节点 2 |
| IR 092 | 00 ~ 07 | 数据链接状态:节点 3 |
| | 08 ~ 15 | 数据链接状态:节点 4 |
| IR 093 | 00 ~ 07 | 数据链接状态:节点 5 |
| | 08 ~ 15 | 数据链接状态:节点 6 |
| IR 094 | 00 ~ 15 | 未用 |
| IR 095 | 00 ~ 10 | 总为 0 |
| | 11 | 终端状态 0:终端电阻切至 OFF 1:终端电阻切至 ON |
| | 12 ~ 15 | 总为 0 |

Controller Link 状态区 2
(IR190 ~ IR195)

| 字 | 位 | 功 能 | |
|----------------|----------------|------------------------------|--------------------------|
| IR 190 | 00 | 网络参数错误标志 1:产生错误;0:无错误 | |
| | 01 | 数据链接表错误标志 1:产生错误;0:无错误 | |
| | 02 | 路由表错误标志 1:产生错误;0:无错误 | |
| | 03 ~ 06 | 总为 0 | |
| | 07 | EEPROM 写错误标志 1:产生错误;0:无错误 | |
| | 08 | 总为 0 | |
| | 09 | 节点号重复错误标志 1:产生错误;0:无错误 | |
| | 10 | 网络参数不匹配错误标志 1:产生错误;0:无错误 | |
| | 11 | 通信控制器传送器错误标志 1:产生错误;0:无错误 | |
| | 12 | 通信控制器硬件错误标志 1:产生错误;0:无错误 | |
| | 13 和 14 | 总为 0 | |
| | 15 | 错误日志标志 1:记录错误记录;0:未记录错误记录 | |
| | IR191 | 00 ~ 07 | 轮询节点的节点号 |
| | | 08 ~ 15 | 启动节点的节点号 |
| | IR192 和 IR 193 | 00 ~ 15 | 网络参与状态 1:参加网络;0:未参加网络 |
| IR 194 和 IR195 | 00 ~ 15 | 未用 | |

SR 区

这些位主要用作与 CQM1H 操作相关的标志。下表为各位的位功能。当输入中断未在计数器模式中使用时，SR244 ~ SR247 也可用作工作位。

| 字 | 位 | 功 能 | |
|-------|---------|---|----|
| SR244 | 00 ~ 15 | 输入中断 0 计数器模式 SV 当输入中断 0 用于计数器模式时为 SV(4 位数十六进制, 0000 ~ FFFF)。(当输入中断不用于计数器模式时不可用作工作位)。 | 24 |
| SR245 | 00 ~ 15 | 输入中断 1 计数器模式 SV 当输入中断 1 用于计数器模式时为 SV(4 位数十六进制, 0000 ~ FFFF)。(当输入中断不用于计数器模式时不可用作工作位)。 | |
| SR246 | 00 ~ 15 | 输入中断 2 计数器模式 SV 当输入中断 2 用于计数器模式时为 SV(4 位数十六进制, 0000 ~ FFFF)。(当输入中断不用于计数器模式时不可用作工作位)。 | |
| SR247 | 00 ~ 15 | 输入中断 3 计数器模式 SV 当输入中断 3 用于计数器模式时为 SV(4 位数十六进制, 0000 ~ FFFF)。(当输入中断不用于计数器模式时不可用作工作位)。 | |
| SR248 | 00 ~ 15 | 输入中断 0 计数器模式 PV 减 1 当输入中断 0 用于计数器模式时为计数器 PV - 1(4 位数十六进制)。 | 24 |
| SR249 | 00 ~ 15 | 输入中断 1 计数器模式 PV 减 1 当输入中断 1 用于计数器模式时为计数器 PV - 1(4 位数十六进制)。 | |
| SR250 | 00 ~ 15 | 输入中断 2 计数器模式 PV 减 1 当输入中断 2 用于计数器模式时为计数器 PV - 1(4 位数十六进制)。 | |
| SR251 | 00 ~ 15 | 输入中断 3 计数器模式 PV 减 1 当输入中断 3 用于计数器模式时为计数器 PV - 1(4 位数十六进制)。 | |

| 字 | 位 | 功 能 | | |
|---------|---------|--|---|-----|
| SR252 | 00 | 高速计数器 0 复位位 | 30 | |
| | 01 | 槽 2 中的内板控制位 脉冲 I/O 板:高速计数器 1 复位位 置 ON 使高速计数器 1(端口 1)的 PV 复位。 绝对编码器接口板: 绝对高速计数器 1 原点补偿位 置 ON 设置绝对高速计数器 1(端口 1)的原点补偿。补偿值置于 DM6611,自动变为 OFF。 | 137 | |
| | 02 | 槽 2 中的内板控制位 脉冲 I/O 板:高速计数器 2 复位位 置 ON 使高速计数器 2(端口 2)的 PV 复位。 绝对编码器接口板: 绝对高速计数器 2 原点补偿位 置 ON 设置绝对高速计数器 2(端口 2)的原点补偿。补偿值置于 DM6612,自动变为 OFF。 | 137 | |
| | 03 ~ 07 | 不用。 | | |
| | 08 | 外围端口复位位 置 ON 使外围端口复位。(当与外围设备相连时无效)。在复位完成后,自动变为 OFF。 | 45 | |
| | 09 | RS-232C 复位位 置 ON 使 RS-232C 端口复位,复位完成后自动变为 OFF。 | | |
| | 10 | PC 设置复位位 置 ON 初始化 PC 设置(DM6600 ~ DM6655)。在复位完成后自动变为 OFF。仅当 PC 在编程模式下才有效。 | 2 | |
| | 11 | 强制状态保持位 OFF:当从编程模式切换到监视模式时,强制使置位/复位位清除。 ON:当从编程模式切换到监视模式时,强制使置位/复位位的状态保持。 | 12 | |
| | 12 | L/O 保持位 OFF:当起动操作或停止操作时,IR 位和 LR 位都复位。 ON:当起动操作或停止操作时,IR 位状态和 LR 位状态保持。 | 12 | |
| | 13 | 不用 | | |
| | 14 | 错误日志复位位 为 ON 时清除错误日志。本操作完成后自动变 OFF。 | 493 | |
| | 15 | 输出 OFF 位 OFF:正常输出状态。 ON:所有输出变为 OFF。 | 152 | |
| | SR253 | 00 ~ 07 | FAL 错误代码 当错误发生时,错误代码(2 位数)存储在此处。当 FAL(06)或 FALS(07)被执行时,FAL 号存储在此处。执行 FAL 00 指令或由编程设备清除错误,此字节就被复位(为 00)。 | 221 |
| | | 08 | 低电池标志 当 CPU 电池电压跌落时变为 ON。 | 490 |
| | | 09 | 循环时间超过标志 当循环时间过运行发生时(即当循环时间超过 100ms)变为 ON。 | 490 |
| 10 ~ 12 | | 不用。 | | |
| 13 | | 总是 ON 标志 | ... | |
| 14 | | 总是 OFF 标志 | ... | |
| 15 | | 首循环标志 操作开始时的一个循环置 ON。 | ... | |

| 字 | 位 | 功 能 | 页 |
|--------|--------|---|--------------------------------|
| SR 254 | 00 | 1分时钟脉冲 (30秒 ON; 30秒 OFF) | ... |
| | 01 | 0.02秒时钟脉冲 (0.01秒 ON; 0.01秒 OFF) | ... |
| | 02~ 03 | 不用 | |
| | 04 | 上溢 (OF 标志 当计算结果高于有符号二进制数据的上限时变为 ON。 | 316 |
| | 05 | 下溢 (UF 标志 当计算结果低于有符号二进制数据的下限时变为 ON。 | 316 |
| | 06 | 微分监视完成标志 当微分监视完成时变为 ON。 | 137 |
| | 07 | STEP(08)执行标志 只在基于 STEP(08)的过程开始时的一个循环置 ON。 | 222 |
| | 08 | HKY(—) 执行标志 在执行 HKY(—) 时变为 ON。 | 419 |
| | 09 | 7SEG(88)执行标志 在执行 7SEG(88)时变为 ON。 | 412 |
| | 10 | DSW(87)执行标志 在执行 DSW(87)时变为 ON。 | 415 |
| | 11~ 12 | 不用。 | |
| | 13 | 通信单元错误标志 在一个通信单元中产生错误时置 ON。该标志反映通信单元操作错误标志 (AR0011)。 | 415 |
| | 14 | 不用 | |
| | 15 | 内板错误标志 安装在槽 1或槽 2上的内板出错时置 ON。槽 1的错误代码存储在 AR0400~AR0407中,槽 2的错误代码存放在 AR0408~ AR0415中。 | ... |
| | SR 255 | 00 | 0.1秒时钟脉冲 (0.05秒 ON; 0.05秒 OFF) |
| 01 | | 0.2秒时钟脉冲 (0.1秒 ON; 0.1秒 OFF) | ... |
| 02 | | 1.0秒时钟脉冲 (0.5秒 ON; 0.5秒 OFF) | ... |
| 03 | | 指令执行错误 (ER) 标志 当在执行指令时发生错误时变为 ON。 | ... |
| 04 | | 进位 (CY) 标志 当指令执行的结果有进位时变为 ON。 | ... |
| 05 | | 大于 (GR) 标志 当比较操作的结果是“大于”时变为 ON。 | ... |
| 06 | | 等于 (EQ) 标志 当比较操作的结果是“等于”,或当指令执行的结果是 0时,变为 ON。 | ... |
| 07 | | 小于 (LE) 标志 当比较操作的结果是“小于”时,变为 ON。 | ... |

注 下列各字不可写:SR248 ~ SR251 和 SR253 ~ SR255。

SR 位的解释

SR25211(强制状态保持位)

当强制置位/复位状态被清除时,被强制的各位会置为 ON 或 OFF 如下:

强制置位被清除:位置为 ON

强制复位被清除:位置为 OFF

除非通上电源时 PC 设置中的 DM6601 被设定为保持强制状态保持位的当前状态,否则当 PC 切换到运行模

式时所有的强制置位或强制复位位会被清除。此设定能用于防止通上电源时清除强制的状态。

SR25212(I/O 保持位)

这个位为 ON 时,当 PC 从编程模式换到运行或监视模式时,保持 IR 和 LR 区中的位状态。(若 I/O 保持位为 OFF,PC 启动时,所有的 IR 和 LR 位都会复位。)

这个位由编程设备置为 ON 和 OFF。

PC 设置中的 DM6601 可设置为在电源变为 ON 时保持 I/O 保持位先前的状态。作出设定和 I/O 保持位为 ON 时,在上电时不清除 IR 和 LR 区中的位的状态。

SR25215(输出 OFF 位)

当这个位变为 ON,所有输出都会变为 OFF,且 CPU 单元的 INH 指示灯发亮。只要输出 OFF 位为 ON,甚至在程序置输出位为 ON 时,输出仍保持 OFF。

只要输出 OFF 位为 ON,晶体管输出单元和脉冲 I/O 板的脉冲输出会保持 OFF。如果安装了高速计数器板,只要输出 OFF 位为 ON,板的外部输出(1~4)会保持 OFF。

输出 OFF 位正常为 OFF,由程序正常时置 OFF。若输出 OFF 位未由程序置 OFF,电源断电时它的 ON/OFF 状态将保持。(尽管当备份电池失败时会不能保持。)

SR 25308(电池电压低标志)和 SR25309(循环时间超出标志)

可在 PV 设置(DM6655)做出设定以避免发生这些错误。

AR 区

这些位主要用作 CQM1H 操作的相关标志。AR05 和 AR06 中的标志与内板操作相关联,且每种内板功能各不相同。下表分别显示了共享标志(AR00~AR04 和 AR07~AR27)和内板专用标志(AR05 和 AR06)。

除 AR23(断电计数器)外,AR 字和位的状态每循环就被刷新。(AR23 仅在电源中断时被刷新。)

共享标志/位(AR00~AR04)

| 字 | 位 | 功 能 |
|------|-------|---|
| AR00 | 00~10 | 不用 |
| | 11 | 通信单元错误标志 在通信单元中出错时置 ON。 |
| | 12~15 | 不用 |
| AR01 | 00~10 | 不用 |
| | 11 | 通信单元重启动位 先置位 ON 然后 OFF,以重启通信单元。 |
| | 12~15 | 不用 |
| AR02 | 00~07 | 网络指令完成代码 包含网络指令(SEND(90)、RECV(98)或 CMND(—))的完成代码。 |
| | 08 | 网络指令(SEND(90)、RECV(98)或 CMND(—))错误标志 网络指令(SEND(90)、RECV(98)或 CMND(—))执行出错时置 ON。 |
| | 09 | 网络指令(SEND(90)、RECV(98)或 CMND(—))允许标志 置 ON 时可执行网络指令(SEND(90)、RECV(98)或 CMND(—))。 |
| | 10~14 | 不用 |
| | 15 | 通信单元连接标志 通信单元安装在 PC 上时置 ON。 |
| AR03 | 00~15 | 通信单元服务时间 以 0.1ms 为单位指示最后一次循环的服务时间(4 位数 BCD)。 |
| AR04 | 00~07 | 槽 1 内板错误代码(16 进制) 00:正常 01、02:硬件错误 04:串行通信板错误 |
| | 08~15 | 槽 2 内板错误代码(16 进制) 00:正常 01、02:硬件错误; 03:PC 设置错误; 04:脉冲输出时 PC 停止或 A/D(D/A)转换错误。 |

内板标志/位 (AR05 和 AR06)

高速计数器板槽 2 标志/位 (AR05 ~ AR06)

| 字 | 位 | 功 能 | 操 作 |
|-------|---------|---------------|--|
| AR 05 | 00 | 高速计数器 1 复位位 | Z 相和软件复位 0: Z 相复位禁止 1: Z 相复位允许 软件复位 0: 软件复位禁止 0 → 1: 执行软件复位 |
| | 01 | 高速计数器 2 复位位 | |
| | 02 | 高速计数器 3 复位位 | |
| | 03 | 高速计数器 4 复位位 | |
| | 04 ~ 07 | 不用 | ... |
| | 08 | 高速计数器 1 比较停止位 | 0 → 1: 启动比较 1 → 0: 停止比较 |
| | 09 | 高速计数器 2 比较停止位 | |
| | 10 | 高速计数器 3 比较停止位 | |
| | 11 | 高速计数器 4 比较停止位 | 0: 持续工作 1: 停止工作 |
| | 12 | 高速计数器 1 停止位 | |
| | 13 | 高速计数器 2 停止位 | |
| | 14 | 高速计数器 3 停止位 | |
| | 15 | 高速计数器 4 停止位 | |
| AR 06 | 00 | 外部输出 1 强制置位位 | 0: 无效 1: 强制 ON |
| | 01 | 外部输出 2 强制置位位 | |
| | 02 | 外部输出 3 强制置位位 | |
| | 03 | 外部输出 4 强制置位位 | |
| | 04 | 外部输出强制置位允许位 | 0: 输出 1 ~ 4 强制置位禁止 1: 输出 1 ~ 4 强制置位允许 |
| | 05 ~ 15 | 不用 | ... |

脉冲 I/O 板槽 2 标志/位 (AR05 ~ AR06)

| 字 | 位 | 操 作 |
|------|---------|--|
| AR05 | 00 ~ 07 | 高速计数器 1 范围比较标志 位 00 ON: 计数器 PV 在比较范围 1 内 位 01 ON: 计数器 PV 在比较范围 2 内 位 02 ON: 计数器 PV 在比较范围 3 内 位 03 ON: 计数器 PV 在比较范围 4 内 位 04 ON: 计数器 PV 在比较范围 5 内 位 05 ON: 计数器 PV 在比较范围 6 内 位 06 ON: 计数器 PV 在比较范围 7 内 位 07 ON: 计数器 PV 在比较范围 8 内 |
| | 08 | 高速计数器 1 比较标志 OFF: 停止 ON: 比较 |
| | 09 | 高速计数器 1 上溢/下溢标志 OFF: 正常 ON: 发生上溢或下溢。 |
| | 10 ~ 11 | 不用。 |
| | 12 ~ 15 | 端口 1 脉冲输出标志 位 12 ON: 减速指定 (OFF: 未指定) 位 13 ON: 脉冲数指定 (OFF: 未指定) 位 14 ON: 脉冲输出完成 (OFF: 未完成) 位 15 ON: 脉冲输出在进行 (OFF: 无脉冲输出) |

| 字 | 位 | 操 作 |
|-------|---------|--|
| AR 06 | 00 ~ 07 | 高速计数器 2 范围比较标志 位 00 ON: 计数器 PV 在比较范围 1 内 位 01 ON: 计数器 PV 在比较范围 2 内 位 02 ON: 计数器 PV 在比较范围 3 内 位 03 ON: 计数器 PV 在比较范围 4 内 位 04 ON: 计数器 PV 在比较范围 5 内 位 05 ON: 计数器 PV 在比较范围 6 内 位 06 ON: 计数器 PV 在比较范围 7 内 位 07 ON: 计数器 PV 在比较范围 8 内 |
| | 08 | 高速计数器 2 比较标志 OFF: 停止 ON: 比较 |
| | 09 | 高速计数器 2 上溢/下溢标志 OFF: 正常 ON: 上溢或下溢发生 |
| | 10 ~ 11 | 不用 |
| | 12 ~ 15 | 端口 2 脉冲输出标志 位 12 ON: 减速指定(OFF: 未指定) 位 13 ON: 脉冲数指定(OFF: 未指定) 位 14 ON: 脉冲输出完成(OFF: 未完成) 位 15 ON: 脉冲输出正在进行(OFF: 无脉冲输出) |

绝对译码器接口板标志/位 (AR05 ~ AR06)

| 字 | 位 | 操 作 |
|-------|---------|--|
| AR 05 | 00 ~ 07 | 高速计数器 1 范围比较标志 位 00 ON: 计数器 PV 在比较范围 1 内 位 01 ON: 计数器 PV 在比较范围 2 内 位 02 ON: 计数器 PV 在比较范围 3 内 位 03 ON: 计数器 PV 在比较范围 4 内 位 04 ON: 计数器 PV 在比较范围 5 内 位 05 ON: 计数器 PV 在比较范围 6 内 位 06 ON: 计数器 PV 在比较范围 7 内 位 07 ON: 计数器 PV 在比较范围 8 内 |
| | 08 | 高速计数器 1 比较标志 OFF: 停止 ON: 比较 |
| | 09 ~ 15 | 不用 |
| AR06 | 00 ~ 07 | 高速计数器 2 范围比较标志 位 00 ON: 计数器 PV 在比较范围 1 内 位 01 ON: 计数器 PV 在比较范围 2 内 位 02 ON: 计数器 PV 在比较范围 3 内 位 03 ON: 计数器 PV 在比较范围 4 内 位 04 ON: 计数器 PV 在比较范围 5 内 位 05 ON: 计数器 PV 在比较范围 6 内 位 06 ON: 计数器 PV 在比较范围 7 内 位 07 ON: 计数器 PV 在比较范围 8 内 |
| | 08 | 高速计数器 2 比较标志 OFF: 停止 ON: 比较 |
| | 09 ~ 15 | 不用 |

共享标志/位 (AR07 ~ AR27)

| 字 | 位 | 功 能 |
|-------|---------|--|
| AR 07 | 00 | Controller Link 数据链接启动位 OFF→ON: 启动(上电时该位为 ON) ON→OFF: 停止 |
| | 01 ~ 11 | 不用。 |
| | 12 | DIP 开关 6 脚标志 OFF: CPU DIP 开关 6 脚是 OFF。 ON: CPU DIP 开关 6 脚是 ON。 |
| | 13 ~ 15 | 不用 |

| 字 | 位 | 功 能 |
|-------|---------|--|
| AR 08 | 00 ~ 03 | RS-232C 端口错误代码(1 位数) 0:正常完成;1. 校验错误;2. 帧错误;3. 过运行错误 |
| | 04 | RS-232C 端口错误标志 CPU 单元中的内置 RS-232C 端口发生通信错误时置 ON。 |
| | 05 | RS-232C 端口传输允许标志 只有在 CPU 单元内置的 RS-232C 端口使用上位机链接或 RS-232C 通信时有效。 |
| | 06 | RS-232C 端口接收完成标志 只有在 CPU 单元内置的 RS-232C 端口使用 RS-232C 通信时有效。 |
| | 07 | RS-232C 端口接收溢出标志 只有在 CPU 单元内置的 RS-232C 端口使用上位机链接或 RS-232C 通信时有效。 |
| | 08 ~ 11 | 外围端口错误代码(1 位数) 0:正常完成;1. 校验错误;2. 帧错误;3. 过运行错误 |
| | 12 | 外围端口错误标志 当外围端口通信错误出现时置 ON。 |
| | 13 | 外围端口传输允许标志 仅使用上位机链接或 RS-232C 通信时有效。 |
| | 14 | 外围端口接收完成标志 仅使用 RS-232C 通信时有效。 |
| | 15 | 外围端口接收溢出标志 仅使用上位机链接或 RS-232C 通信时有效。 |
| AR 09 | 00 ~ 15 | RS-232C 端口接收计数器 4 位数 BCD;仅使用 RS-232C 通信时有效。 |
| AR 10 | 00 ~ 15 | 外围口接收计数器。 4 位数 BCD;仅使用 RS-232C 通信时有效。 |
| AR 11 | 00 ~ 07 | 高速计数器 0 范围比较标志 位 00 ON:计数器 PV 在比较范围 1 内 位 01 ON:计算机 PV 在比较范围 2 内 位 02 ON:计算机 PV 在比较范围 3 内 位 03 ON:计算机 PV 在比较范围 4 内 位 04 ON:计算机 PV 在比较范围 5 内 位 05 ON:计算机 PV 在比较范围 6 内 位 06 ON:计算机 PV 在比较范围 7 内 位 07 ON:计算机 PV 在比较范围 8 内 |
| | 08 ~ 14 | 未用 |
| | 15 | 指定脉冲输出位的脉冲输出状态 0:停止;1:输出 |
| AR 12 | 00 ~ 15 | 不用。 |
| AR 13 | 00 | 存储器盒安装标志 如存储器盒在上电时安装则置 ON。 |
| | 01 | 时钟可用标志 如果安装有时钟的存储器盒则变为 ON。 |
| | 02 | 存储器盒写保护标志 如安装 EEPROM 或闪烁内存存储器盒并写保护,或安装 EPROM 时则置 ON。 |
| | 03 | 不用 |
| | 04 ~ 07 | 存储器盒代码(1 位数) 0:未安装存储器盒。 1:安装了 4K 字 EEPROM 存储器盒。 2:安装了 8K 字 EEPROM 存储器盒。 3:安装了 16K 字闪烁内存存储器盒。 4:安装了 EPROM 型存储器盒。 |
| | 08 ~ 15 | 不用 |

| 字 | 位 | 功 能 |
|-------|---------|---|
| AR 14 | 00 | CPU 单元→存储器盒传送位 从 CPU 单元向存储器盒传送时变为 ON, 在操作完成后再自动地变为 OFF。 |
| | 01 | 存储器盒→CPU 单元传送位 从存储器盒向 CPU 单元传送时变为 ON, 在操作完成后再自动地变为 OFF。 |
| | 02 | 存储器盒比较位 变 ON 去比较 PC 的内容与存储器盒的内容, 在比较完成后自动地变为 OFF。 |
| | 03 | 存储器盒比较结果标志 ON: 发现差别或不可比较。 OFF: 内容已比较完, 并发现是相同的。 |
| | 04 ~ 11 | 不用 |
| | 12 | 编程模式传送错误标志 当由于正处于编程模式而不能执行传送时变为 ON。 |
| | 13 | 写保护错误标志 当由于写保持而不能执行传送时变为 ON。 |
| | 14 | 容量不足标志 当由于传送目的地容量不足而不能执行传送时变为 ON。 |
| AR 15 | 15 | 无程序标志 当由于存储器盒内无程序而不能执行传送时变为 ON。 |
| | 00 ~ 07 | 存储器盒程序代码 代码(2 位数)表明存储在存储器盒内的程序的长度。 00: 无程序, 或未安装存储器盒。 04: 程序小于 3.2K 字。 08: 程序小于 7.2K 字。 12: 程序小于 11.2K 字。 16: 程序小于 15.2K 字。 |
| AR 16 | 08 ~ 15 | CPU 单元程序代码 代码(2 位数)表明存储在 CPU 单元内的程序的长度。 04: 程序小于 3.2K 字。 08: 程序小于 7.2K 字。 12: 程序小于 11.2K 字。 16: 程序小于 15.2K 字。 |
| | 00 ~ 10 | 不用 |
| | 11 | PC 设置初始化标志 当在 PC 设置区发生检验和错误时置为 ON, 所有设定被初始化回缺省设定。 |
| | 12 | 程序无效标志 当在 UM(用户程序)区发生检验和错误时, 或执行不正确的指令时置为 ON。 |
| | 13 | 指令表初始化标志 当在指令表发生检验和错误时置为 ON, 所有设定被初始化回缺省设置。 |
| | 14 | 存储器盒加入标志 如上电时安装存储器盒则置 ON。 |
| AR 17 | 15 | 存储器盒传送错误标志 如果在 DIP 开关 2 脚设置为 ON 时(即: 设置为在上电时自动传送存储器盒的内容。)传送不能成功地被执行则变 ON。 |
| | 00 ~ 07 | 现在时间的“分”部分, 2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |
| AR 18 | 08 ~ 15 | 现在时间的“小时”部分, 2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |
| | 00 ~ 07 | 现在时间的“秒”部分, 2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |
| AR 18 | 08 ~ 15 | 现在时间的“分”部分, 2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |

| 字 | 位 | 功 能 |
|-------|---------|--|
| AR 19 | 00 ~ 07 | 现在时间的“小时”部分。2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |
| | 08 ~ 15 | 现在时间的“日”部分,2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页。) |
| AR 20 | 00 ~ 07 | 现在时间的“月”部分。2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页) |
| | 08 ~ 15 | 现在时间的“年”部分,2 位数 BCD(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页。) |
| AR 21 | 00 ~ 07 | 现在时间的“星期”部分,2 位数 BCD(00:星期日 ~ 06:星期六)(仅在安装具有时钟功能的存储器盒时有效。详情参见第 159 页。) |
| | 08 ~ 12 | 不用 |
| | 13 | 30 秒调整位 仅在安装具有时钟功能的存储器盒时有效。详情参阅第 159 页。 |
| | 14 | 时钟停止位 仅在安装具有时钟功能的存储器盒时有效。详情参阅第 159 页。 |
| | 15 | 时钟设置位 仅在安装具有时钟功能的存储器盒时有效。详情参阅第 159 页。 |
| AR 22 | 00 ~ 07 | 输入字 分配给输入位的字数(2 位数 BCD)(只存贮识别的值。若发生 I/O 单元溢出错误则存贮 00 值。) |
| | 08 ~ 15 | 输出字 分配给输出位的字数(2 位数 BCD)(只存贮识别的值。若发生 I/O 单元溢出错误则存贮 00 值。) |
| AR 23 | 00 ~ 15 | 断电计数器(4 位数 BCD) 这是电源被断开的次数的计数。要清除该计数,从编程设备写入“0000”。 |
| AR 24 | 00 | 上电 PC 设置错误标志 在 DM6600 ~ DM6614 内(在操作开始时被读的 PC 设置区部分)有错误时变为 ON。 |
| | 01 | 起动 PC 设置错误标志 在 DM6615 ~ DM6614 内(在上电时被读的 PC 设置区部分)有错误时变为 ON。 |
| | 02 | 运行 PC 设置错误标志 在 DM6645 ~ DM6655 内(一直被读的 PC 设置区部分)有错误时变为 ON。 |
| | 03 | CPU 单元外围端口设定改变标志 |
| | 04 | CPU 单元 RS—232C 端口设定改变标志 |
| | 05 | 长循环时间标志 如果实际循环时间长于 DM6619 中设置的循环时间则变为 ON。 |
| | 06 ~ 07 | 不用 |
| | 08 ~ 15 | 显示检出的 I/O 总线错误的字号的代码(2 位数 BCD) 00 ~ 15(BCD):对应于输入字 000 ~ 015。 80 ~ 95(BCD):对应于输出字 100 ~ 115。 FO(十六进制):安装在槽 1 的内板不能确认。 FI(十六进制):安装在槽 2 的内板不能确认。 FF(十六进制):端板不能确认。 |
| AR 25 | 00 ~ 07 | 不用。 |
| | 08 | FPD(一)教学位 |
| | 09 ~ 11 | 不用。 |
| | 12 | 跟踪完成标志 |
| | 13 | 跟踪标志 |
| | 14 | 跟踪触发标志 |
| | 15 | 采样开始位(不能由程序重写此位。) |

| 字 | 位 | 功 能 |
|-------|---------|--|
| AR 26 | 00 ~ 15 | 最长循环时间(4 位数 BCD) 存储操作开始后最长循环时间。它在操作开始时,而不是在结束时被清除。 单位可以是下列中的任一个,这依据 9F 监视时间(DM6618)的设定而定。 缺省:0.1ms;“10ms”设定:0.1ms;“100ms”设定:1ms;“1s”设定:10ms |
| AR27 | 00 ~ 15 | 当前循环时间(4 位数 BCD) 存储操作时最近的循环时间。操作停止时,当前循环时间不被清除。 单位可以是下列中的任一个,这取决于 9F 监视时间(DM6618)的设定。缺省:0.1ms; “10ms”设定:0.1ms;“100ms”设定:1ms;“1s”设定:10ms |

附录 D

使用时钟

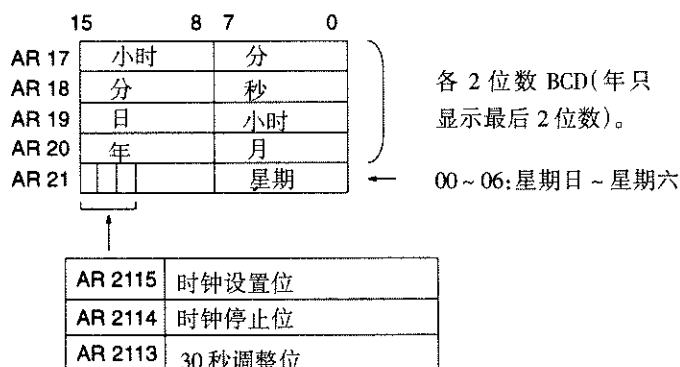
CQM1H PC 通过安装具有时钟的存储器盒装备时钟。本附录说明如何使用时钟。

带有内置时钟的存储器盒的型号的末尾有一个“R”字。例如 CQM1—ME04R 型存储器盒带有一个内置时钟。3—11 使用存储器盒介绍一系列存储器盒的使用。

注 若将存储器盒从 CPU 单元上取下,则时钟将停止,且当前日期和时间数据会丢失。

包含日期和时间的字

下图示出了使用时钟的字(AR17~AR21)的配置。当需要时,这些字可以被读出和使用。(设定 AR17,以使可以迅速调用小时和分。)



设定时间

设定时间时:使用编程设备如下:

注 使用编程设备的菜单操作(如编程器)可以方便地设置时间。关于编程器操作步骤参见 CQM1H 操作手册。

全部设定

通过下述过程设定时间和日期:

- 1, 2, 3... 1. 使 AR2114(时钟停止位)变为 ON,以使时钟停止,并允许 AR18~AR21 被改写。
2. 使用编程设备,设置 AR18~AR20(分/秒,日期/小时,年/月)和 AR2100~AR2107(星期)。
3. 到达第二步设置的时间时,使 AR2115 变为 ON(时钟设置位)。时钟从设置时间开始工作,而时钟停止位和时钟设置位将自动变为 OFF。

仅设定秒

还可能使用 AR2113 仅仅设置秒为“00”而不通过复杂的步骤。当 AR2113 变为 ON 时,时钟时间会改变如下:

如果秒设定是 00~29,则秒会复位到“00”,而分设定会保持不变。

如果秒设定是 30~59,则秒会复位到“00”,而分设定会加 1。

当时间设定完成时,AR2113 会自动变为 OFF。

附录 E

I/O 分配表

| | | | | |
|-------|----|----|----|----|
| 系统名称 | | 制表 | 检验 | 审定 |
| PC 型号 | 表号 | | | |

| IR | 单元号: | 型号: | IR | 单元号: | 型号: |
|----|------|-----|----|------|-----|
| 00 | | | 00 | | |
| 01 | | | 01 | | |
| 02 | | | 02 | | |
| 03 | | | 03 | | |
| 04 | | | 04 | | |
| 05 | | | 05 | | |
| 06 | | | 06 | | |
| 07 | | | 07 | | |
| 08 | | | 08 | | |
| 09 | | | 09 | | |
| 10 | | | 10 | | |
| 11 | | | 11 | | |
| 12 | | | 12 | | |
| 13 | | | 13 | | |
| 14 | | | 14 | | |
| 15 | | | 15 | | |
| IR | 单元号: | 型号: | IR | 单元号: | 型号: |
| 00 | | | 00 | | |
| 01 | | | 01 | | |
| 02 | | | 02 | | |
| 03 | | | 03 | | |
| 04 | | | 04 | | |
| 05 | | | 05 | | |
| 06 | | | 06 | | |
| 07 | | | 07 | | |
| 08 | | | 08 | | |
| 09 | | | 09 | | |
| 10 | | | 10 | | |
| 11 | | | 11 | | |
| 12 | | | 12 | | |
| 13 | | | 13 | | |
| 14 | | | 14 | | |
| 15 | | | 15 | | |

附录 F

程序代码表

| | | | | |
|------|----|----|----|----|
| 系统名称 | | 制表 | 检验 | 审定 |
| PC | 表号 | | | |

| 地址 | | 指令 | 功能代码 | 操 作 |
|----|---|----|------|-----|
| | 0 | 0 | | |
| | 0 | 1 | | |
| | 0 | 2 | | |
| | 0 | 3 | | |
| | 0 | 4 | | |
| | 0 | 5 | | |
| | 0 | 6 | | |
| | 0 | 7 | | |
| | 0 | 8 | | |
| | 0 | 9 | | |
| | 1 | 0 | | |
| | 1 | 1 | | |
| | 1 | 2 | | |
| | 1 | 3 | | |
| | 1 | 4 | | |
| | 1 | 5 | | |
| | 1 | 6 | | |
| | 1 | 7 | | |
| | 1 | 8 | | |
| | 1 | 9 | | |
| | 2 | 0 | | |
| | 2 | 1 | | |
| | 2 | 2 | | |
| | 2 | 3 | | |
| | 2 | 4 | | |
| | 2 | 5 | | |
| | 2 | 6 | | |
| | 2 | 7 | | |
| | 2 | 8 | | |
| | 2 | 9 | | |
| | 3 | 0 | | |
| | 3 | 1 | | |

| 地址 | | | | 指令 | 功能代码 | 操 作 |
|----|--|--|---|----|------|-----|
| | | | 3 | 2 | | |
| | | | 3 | 3 | | |
| | | | 3 | 4 | | |
| | | | 3 | 5 | | |
| | | | 3 | 6 | | |
| | | | 3 | 7 | | |
| | | | 3 | 8 | | |
| | | | 3 | 9 | | |
| | | | 4 | 0 | | |
| | | | 4 | 1 | | |
| | | | 4 | 2 | | |
| | | | 4 | 3 | | |
| | | | 4 | 4 | | |
| | | | 4 | 5 | | |
| | | | 4 | 6 | | |
| | | | 4 | 7 | | |
| | | | 4 | 8 | | |
| | | | 4 | 9 | | |
| | | | 5 | 0 | | |
| | | | 5 | 1 | | |
| | | | 5 | 2 | | |
| | | | 5 | 3 | | |
| | | | 5 | 4 | | |
| | | | 5 | 5 | | |
| | | | 5 | 6 | | |
| | | | 5 | 7 | | |
| | | | 5 | 8 | | |
| | | | 5 | 9 | | |
| | | | 6 | 0 | | |
| | | | 6 | 1 | | |
| | | | 6 | 2 | | |
| | | | 6 | 3 | | |
| | | | 6 | 4 | | |
| | | | 6 | 5 | | |
| | | | 6 | 6 | | |
| | | | 6 | 7 | | |

| 地址 | | | | 指令 | 功能代码 | 操 作 |
|----|--|--|-----|----|------|-----|
| | | | 6 8 | | | |
| | | | 6 9 | | | |
| | | | 7 0 | | | |
| | | | 7 1 | | | |
| | | | 7 2 | | | |
| | | | 7 3 | | | |
| | | | 7 4 | | | |
| | | | 7 5 | | | |
| | | | 7 6 | | | |
| | | | 7 7 | | | |
| | | | 7 8 | | | |
| | | | 7 9 | | | |
| | | | 8 0 | | | |
| | | | 8 1 | | | |
| | | | 8 2 | | | |
| | | | 8 3 | | | |
| | | | 8 4 | | | |
| | | | 8 5 | | | |
| | | | 8 6 | | | |
| | | | 8 7 | | | |
| | | | 8 8 | | | |
| | | | 8 9 | | | |
| | | | 9 0 | | | |
| | | | 9 1 | | | |
| | | | 9 2 | | | |
| | | | 9 3 | | | |
| | | | 9 4 | | | |
| | | | 9 5 | | | |
| | | | 9 6 | | | |
| | | | 9 7 | | | |
| | | | 9 8 | | | |
| | | | 9 9 | | | |

附录 G

FAL 号列表

| | | | | |
|------|----|----|----|----|
| 系统名称 | | 制表 | 检验 | 审定 |
| PC型号 | 表号 | | | |

| FAL号 | FAL内容 | 校正措施 | FAL号 | FAL内容 | 校正措施 |
|------|-------|------|------|-------|------|
| 00 | | | 32 | | |
| 01 | | | 33 | | |
| 02 | | | 34 | | |
| 03 | | | 35 | | |
| 04 | | | 36 | | |
| 05 | | | 37 | | |
| 06 | | | 38 | | |
| 07 | | | 39 | | |
| 08 | | | 40 | | |
| 09 | | | 41 | | |
| 10 | | | 42 | | |
| 11 | | | 43 | | |
| 12 | | | 44 | | |
| 13 | | | 45 | | |
| 14 | | | 46 | | |
| 15 | | | 47 | | |
| 16 | | | 48 | | |
| 17 | | | 49 | | |
| 18 | | | 50 | | |
| 19 | | | 51 | | |
| 20 | | | 52 | | |
| 21 | | | 53 | | |
| 22 | | | 54 | | |
| 23 | | | 55 | | |
| 24 | | | 56 | | |
| 25 | | | 57 | | |
| 26 | | | 58 | | |
| 27 | | | 59 | | |
| 28 | | | 60 | | |
| 29 | | | 61 | | |
| 30 | | | 62 | | |
| 31 | | | 63 | | |

| FAL 号 | FAL 内容 | 校正措施 | FAL 号 | FAL 内容 | 校正措施 |
|-------|--------|------|-------|--------|------|
| 64 | | | 82 | | |
| 65 | | | 83 | | |
| 66 | | | 84 | | |
| 67 | | | 85 | | |
| 68 | | | 86 | | |
| 69 | | | 87 | | |
| 70 | | | 88 | | |
| 71 | | | 89 | | |
| 72 | | | 90 | | |
| 73 | | | 91 | | |
| 74 | | | 92 | | |
| 75 | | | 93 | | |
| 76 | | | 94 | | |
| 77 | | | 95 | | |
| 78 | | | 96 | | |
| 79 | | | 97 | | |
| 80 | | | 98 | | |
| 81 | | | 99 | | |

附录 H

扩展 ASCII

下面的代码用于在使用 MSG(46) 或 FPD(一) 时向编程器或数据访问器输出字符。详情请参阅第 369 页和第 375 页。

| 右位数 | 左 位 数 | | | | | | | | | | | | |
|-----|---------|----|---|---|---|---|---|----|---|---|---|---|---|
| | 0、1、8、9 | 2 | 3 | 4 | 5 | 6 | 7 | A | B | C | D | E | F |
| 0 | | | 0 | @ | P | \ | P | | - | @ | P | \ | P |
| 1 | | ! | 1 | A | Q | a | q | ! | 1 | A | Q | a | q |
| 2 | | " | 2 | B | R | b | r | " | 2 | B | R | b | r |
| 3 | | # | 3 | C | S | c | s | # | 3 | C | S | c | s |
| 4 | | \$ | 4 | D | T | d | t | \$ | 4 | D | T | d | t |
| 5 | | % | 5 | E | U | e | u | % | 5 | E | U | e | u |
| 6 | | & | 6 | F | V | f | v | & | 6 | F | V | f | v |
| 7 | | ' | 7 | G | W | g | w | ' | 7 | G | W | g | w |
| 8 | | (| 8 | H | X | h | x | (| 8 | H | X | h | x |
| 9 | |) | 9 | I | Y | i | y |) | 9 | I | Y | i | y |
| A | | * | : | J | Z | j | z | * | : | J | Z | j | z |
| B | | + | ; | K | [| k | { | + | ; | K | [| k | { |
| C | | , | < | L | ≠ | l | ! | , | < | L | ≠ | l | ! |
| D | | - | = | M |] | m | } | - | = | M |] | m | } |
| E | | . | > | N | ^ | n | → | . | > | N | ^ | n | |
| F | | / | ? | O | - | o | ← | / | ? | O | - | o | → |

术语汇编

| | |
|-------------------|--|
| * DM | 间接寻址的 DM 区,见间接地址和 DM 区。 |
| 1:1 链接 | 在两台 PC 之间建立的一种链接,以使在它们的 LR 区建立公共数据。 |
| 1:1 link | |
| ACP | 见加计数输入。 |
| 加计数输入 | 用于当信号由 OFF 转为 ON 时使计数器递增计数的一个输入信号。 |
| add count input | |
| 地址 | 用来识别数据或程序指令在内存中所处位置的一个数字。 |
| address | |
| 与 | 一种逻辑运算,只有在二个前提都是“真”时,结果才是“真”。在梯形图编程中,前提通常是位的开/关状态,或称为执行条件的状态的逻辑组合。 |
| AND | |
| 区域 | 见数据区和内存区。 |
| area | |
| 区域前缀 | 用于标明 PC 中一个内存区的一个或二个字母前缀。除 IR 区和 SR 区外,所有内存区都需要用前缀来标明内存中的地址。 |
| area prefix | |
| 算术移位 | 一种包括进位位的移位操作。 |
| arithmetic shift | |
| ASCII | 美国国家信息交换标准代码的缩写,是用于向打印机和其他外围器件输出字符时的编码。 |
| AR 区 | 分配给 标志和控制位的一个 PC 数据区。 |
| AR area | |
| AUTOEXEC.BAT | 一个包含了在启动时能自动执行命令的 MS DOS 文件。 |
| 备份 | 从存在数据得到一个拷贝,以保证即使原始数据受到损坏或被清除时不会失去数据。 |
| back - up | |
| 基本指令 | 用于梯形图的基本指令。 |
| basic instruction | |
| 波特率 | 一个系统内二个设备间数据传输速度的计量单位(Bit/秒)。 |
| baud rate | |
| BCD | 见二进制编码的十进制。 |
| BCD 计算 | 用二进制编码的十进制表示数字的算术运算。 |
| 二进制 | 所有数以 2 为基数表示的数制,即各数仅用“0”和“1”来书写。每四个二进制位的一组等于一个十六进制数,为了方便,在存储器中二进制数据经常用十六进制数表示。 |
| binary | |

| | |
|----------------------------------|---|
| 二进制运算 | 用二进制数表示的算术运算。 |
| binary calculation | |
| 二进制编码的十进制 | 一种数字表示制,每四个二进制位等于一个十进制数。 |
| binary - coded decimal | |
| 位 | 是计算机用来表示信息的最小单位。一个位有 0 或 1 二个值,对应于电信号的 ON 和 OFF。一个位表示一个二进制数。有些特定地址的位分配给一些特殊用途。如,来自外部设备的输入状态的保持,而其它位供编程使用。 |
| bit | |
| 位地址 | 在内存中一个位的信息所贮存的地方。一个位地址要标明数据区和字的地址以及这个位在字中所处位置的编号。 |
| bit address | |
| 位标记 | 用于标明指令所要使用的一个字中某个位或几个位的一个操作数。 |
| bit designator | |
| 位号 | 表明位在一个字中所处位置的数字。位 00 是最右(最低)位,位 15 是最左(最高)位。 |
| bit number | |
| 位控制指令 | 用于控制某一个单独位的状态的指令,与它相对是整个字的状态。 |
| bit - control instruction | |
| 块 | 见逻辑块和指令块。 |
| block | |
| 积木式 PC | PC 是由单独部件或“积木块”构成的。在积木块 PC 中没有一个单元可以独立地称为一个 PC。PC 象是各单元的功能组合。 |
| building block PC | |
| 总线 | 在任何与之连接的单元之间用于传递数据的通路。 |
| bus | |
| 母线 | 梯形图的左边起始线(有时在梯形图的右边也有)。指令的执行沿这条线逐条下向,它是所有指令行的起始点。 |
| bus bar | |
| 字节 | 数据单位,相当于 8 位即半个字。 |
| byte | |
| 调用 | 指令从主程序移到一个子程序执行的过程。可以通过一条指令或一个中断来调用子程序。 |
| call | |
| 进位标志 | 在算术运算中使用的一个标志,用于保持在加法运算或乘法运算中出现的进位,或在减法运算中标志结果是负值。进位标志也用于某种移位操作。 |
| Carry Flag | |
| 中央处理单元 | 一个能存贮程序和数据,并能执行程序中的指令的器件。在一个 PC 系统中,中央处理单元执行程序、处理 I/O 信号、与外部设备通讯等。 |
| central processing unit | |
| CH | 见字。 |
| 通道 | 见字。 |
| channel | |
| 字符代码 | 用于表示一个字母数字字符的数字代码(通常为二进制)。 |
| character code | |

| | |
|--------------------------------|--|
| 检验和 checksum | 在一个通信数据块中所传输数据的总和。根据接收的数据重新计算一个检验和,由此可以检验在传输过程中是否出错。 |
| 时钟脉冲 clock pulse | 在内存特定位置所激活的一个脉冲,可用于定时操作。各种时钟脉冲的脉冲宽度不同,因此频率不同。 |
| 时钟脉冲位 clock pulse bit | 内存中的一个位,它能提供用于定时操作的脉冲。有各种脉冲宽度、各种不同频率的时钟脉冲位可供使用。 |
| 公共数据 common data | 贮存在一台 PC 内存中的数据,这些数据为同一系统其他 PC 共享。每台 PC 在这个区域中分得一个指定的部分。每台 PC 将信息写入分配给它的部分,并从分配给其它 PC 的区中读数据,从而分享公共数据。 |
| 通信电缆 communication cable | 用于在一个控制系统的各部件之间传送数据并符合 RS-232C 或 RS-422 标准的电缆。 |
| 比较指令 comparison instruction | 用于比较内存中不同位置的信息,以确定这些数据之间的相对关系的指令。 |
| 完成标志 Completion Flag | 用于定时器和计数器的一个标志位,在定时器计时到达或计数器达到设定值时转为 ON。 |
| 条件 condition | 在指令行中的一个符号,表示控制终端指令执行条件的一条指令,每个条件分配有内存中的一个位,这位决定了它的状态,赋值给每个条件的位的状态,它决定了下一个执行条件。条件对应于 LOAD、LOAD NOT、AND、AND NOT、OR 或 OR NOT 指令。 |
| CONFIG.SYS | 包含一台个人计算机的环境设置的一个 MS DOS 文件。 |
| 常数 constant | 一个实际数值操作数的输入。可以在某些操作数的内存区地址的地方输入常数,某些操作数必须输入常数。 |
| 控制位 control bit | 内存区中一个位,可通过程序设置或通过编程器件来设置,以达到某种目的。如,用重启动位 ON 和 OFF 来重新启动一个单元。 |
| 控制数据 control data | 标明怎样执行一条指令的一个操作数。控制数据可以标明一个字的一部分作为操作数,也可能标明一条数据传送指令的目的,也可标明在一条指令中所用的数据表的大小,等等。 |
| 控制信号 control signal | 由 PC 发出去影响被控系统运行的信号。 |
| 控制系统 Control System | 用于控制其他设备的所有硬件和软件的总和,一个控制系统包括 PC 系统、PC 程序和用于控制和从被控系统得到反馈的所有 I/O 设备。 |
| 受控系统 controlled system | 由一个 PC 系统控制的设备。 |

| | |
|-------------------------------------|--|
| 计数脉冲 count pulse | 由一个计数器计数的信号。 |
| 计数器 counter | 在内存中用于计数一个指定过程发生次数的一组专用数或字,或在内存中可以通过 TIM/CNT 位存取的内存单元,用于记录一个位或一个执行条件从 OFF 转为 ON 的变化次数。 |
| CPU | 见中央处理单元。 |
| CTS | 是 clear-to-send(清除发送)的缩写,电子设备间通信用的一個信号,表示接收端已准备好接受数据。 |
| CY | 见进位标志。 |
| 循环 cycle | 由 CPU 完成的一个工作单位,包括执行梯形图程序、外围设备服务、I/O 刷新等。 |
| 循环时间 cycle time | 完成一个 CPU 处理周期所需的时间。 |
| 循环中断 cyclic interrupt | 见定时中断。 |
| 数据区 data area | 在 PC 内存中保存指定类型数据的区域。 |
| 数据区边界 data area Boundary | 在数据区中的最高地址,当指定一个多字操作数时,必须要确定它没有超过数据区的最高地址。 |
| 数据磁盘 data disk | 用于贮存程序、DM 区内容、注释和其它用户数据的磁盘。 |
| 数据长度 data length | 在数据通讯中,作为一个单元处理的传输数据的位数。 |
| 数据链接 data link | 一种自动的数据传送操作,使 PC 或 PC 内的单元能经过公共数据区交换数据。 |
| 数据链接区 data link area | 一个通过数据链接建立的公共数据区。 |
| 数据移动指令 data movement instruction | 用于将内存中某一位置的数据移到另一位置的指令,在原来位置的数据仍保留不改变。 |
| 数据共享 data sharing | 在二台或多台 PC 之间生成公共数据区或公共数据字的过程。 |
| 数据跟踪 data trace | 在程序执行过程中指定内存位置的所存内容改变时,都要被记录下来,这种过程称为数据跟踪。 |
| 数据传送 data transfer | 在同一器件中,或在经由通讯线或网络连接的不同器件间,将数据从一个存储位置传送到另一位置的过程。 |

| | |
|--------------------------------------|---|
| 调试 debug | 使编写的程序初稿修改至达到予期要求的过程。调试包括消除语法错误,也包括时间或控制条件协调的精确调整。 |
| 十进制 decimal | 一种以 10 为基数的数字表示数制。在 PC 中所有数据最后都以二进制形式贮存,通常用四个二进制位表示一个十进制数,称为二进制编码的十进制。 |
| 递减 decrement | 递减一个数值,通常为减 1。 |
| 缺省 default | 当用户不特别设置另一值时,PC 自动设置的一个值。许多器件都是在上电时执行缺省条件。 |
| 定义符 definer | 一个用作指令的操作数的数字,但是这个数是定义指令本身,而不是这条指令要操作的对象,定义符包括跳转编号、子程序编号等。 |
| 目的地 destination | 指令存放其操作数据的地方叫目的地,对应于从其中取出数据供指令用的则叫作源。 |
| 微分指令 differentiated instruction | 在执行条件由 OFF 到 ON 转变时仅仅执行一次的指令。而非微分指令,只要执行条件是 ON 时,每次扫描都执行。 |
| 微分法指令 differentiation instruction | 用于保证操作数位在上升沿指令执行条件由 OFF 到 ON 转换,或下降沿指令执行条件由 ON 到 OFF 后其接通不会超过一个扫描周期。 |
| 数字 digit | 由四个位组成的内存存贮单位。 |
| 数字指定 digit designator | 一个操作数,用于定义一条指令所使用的一个数字或几个数字。 |
| DIN 导轨 DIN track | 用于将各种器件嵌在槽内的一种导轨,能将器件快速和方便地装在它上面。 |
| DIP 开关 DIP switch | 以双列封装,安装在线路板上组成一个信号阵列,用于设置工作参数的开关组。 |
| 直接输出 direct output | 能使程序执行结果立即输出,以减小循环时间影响的一种方法。 |
| 分布控制 distributed control | 一种自动化的概念。按这种概念,一个自动化系统的每一部分的控制设备都放在实际要控制对象的附近,即控制是分散地分布在整个系统中。分布控制是 PC 系统的基本概念。 |
| DM 区 DM area | 仅用于保持以字为数据的数据区,在 DM 区中的字不能逐位存取。 |
| DM 字 DM word | 在 DM 区中的一个字。 |
| 下载 downloading | 从高一级或上位计算机向低一级或从站计算机传送程序或数据的过程。如果有编程设备,则编程设备看作是上位机。 |

| | |
|------------------------------------|--|
| EEPROM | 电可擦除的可编程只读存储器,是一种 ROM,其内部贮存的信息可以擦去和重新编程。只要使用连接到 EEPROM 芯片引脚上的专门控制线,不必将 EEPROM 拔下就可擦除和重新编程。 |
| 电气噪声 electrical noise | 一个或多个电气特性如电压、电流和数据发生的随机变化,这种变化会干扰一个设备的正常运行。 |
| EPROM | 可擦除的可编程只读存储器,是一种 ROM,其内部存贮的数据可通过紫外线或其他方法擦去,并可重新编程。 |
| 错误代码 error code | 所产生的数码用于指示出现了一个错误,和关于错误性质的一些提示。某些错误代码由系统发出,有一些是由操作人员在程序中定义。 |
| 错误日志区 Error Log Area | 用于存储表示系统中发生错误的的时间和性质的记录的一个区域。 |
| 偶校验 even parity | 一种通信设定,它调整 ON 位的数目,使它始终是偶数,见校验。 |
| 事件处理 event processing | 在应答一个事件,如一个中断信号时所进行的处理。 |
| 异或非 exclusive NOR | 一种逻辑运算,它在二个前提均为“真”或均为“假”时,结果是“真”。在梯形图编程时,前提通常是位的 ON/OFF 状态,或者是这些状态的逻辑组合,称为执行条件。 |
| 异或 exclusive OR | 一种逻辑运算,如果一个前提是“真”,而且仅仅是一个前提为“真”时结果为“真”。在梯形图编程时,前提通常是位的 ON/OFF 状态,或者是这些状态的逻辑组合,称为执行条件。 |
| 执行条件 execution condition | 是 ON 或 OFF 状态,根据这个状态决定指令是否执行。执行条件是由同一指令行中条件的逻辑组合来决定的,直到当前正在执行的指令。 |
| 执行循环 execution cycle | 用于执行 CPU 所要求的所有处理的循环,包括执行程序、I/O 刷新、外围服务等。 |
| 执行时间 execution time | CPU 执行某一条指令或整个程序所需的时间。 |
| 扩展计数器 extended counter | 连续使用二条或多条计数指令在程序中生成的一个计数器。这种计数器的计数能力高于任何单指令提供的标准计数器。 |
| 扩展定时器 extended timer | 连续使用二条或多条定时指令在程序中生成的一个定时器。这种定时器的时间范围长于任何单指令提供的标准定时器。 |
| FA | 工厂自动化。 |
| 工厂计算机 factory computer | 一种通用计算机,通常与事务处理计算机非常相似,但用于工厂自动控制。 |
| FAL 错误 FAL error | 由于在用户程序中执行了 FAL(O6)指令而产生的错误。 |

| | |
|---------------------------------|--|
| FALS 错误 | 由于在用户程序中执行了 FALS(07)指令而产生的错误或由系统产生的错误。 |
| FALS error | |
| 致命错误 | 致使 PC 停止运行,并且要排除以后才能继续运行的错误。 |
| fatal error | |
| FCS | 见帧校验和。 |
| 标志 | 在内存中的一个专用位,由系统设置以指示某种类型的运行状态,有些标志如 |
| flag | 进位标志,也可由操作员或通过程序来置位。 |
| 闪烁位 | 一个由程序控制的以规定频率改变 ON 和 OFF 的位。 |
| flicker bit | |
| 浮点十进制 | 用一个数(尾数)乘以 10 的幂表示的十进制数如 0.538×10^{-5} 。 |
| floating - point decimal | |
| 强制复位 | 通过编程器将一个位强制变 OFF 的过程。通常一个位的复位是程序执行的结果。 |
| force reset | |
| 强制置位 | 通过编程器将一个位强制变 ON 的过程。通常一个位的置位是程序执行的结果。 |
| force set | |
| 强制状态 | 被强制复位或强制置位的一个位的状态。 |
| forced status | |
| 帧校验和 | 在一个规定的计算范围内所有数据异或的结果。帧校验和对传送的数据在发送和接收时都进行计算,以判定数据传输的正确性。 |
| frame checksum | |
| 功能代码 | 用于将指令输入到 PC 的两位数字。 |
| function code | |
| 硬件错误 | 源于 PC 硬件结构(电子元件)的一个故障,与源于软件(即程序)的软件故障相反。 |
| hardware error | |
| 标题代码 | 在一个指令中指定指令要做什么的代码。 |
| header code | |
| 十六进制 | 以十六为基数的数制,在 PC 机中所有数据以二进制形式存贮,但在编程器上显示和输入时为了方便,经常表示为十六进制。每四位二进制位组成一组,在数值上与一个十六进制数等效。 |
| hexadecimal | |
| 上位计算机 | 上位机链接系统中用于向 PC 传送或接收 PC 数据的一台计算机。上位机用作数据管理和总系统控制,一般是小型个人计算机或商务计算机。 |
| host computer | |
| 上位机接口 | 能与上位机通信的接口。 |
| host interface | |
| 上位机链接 | PC 与上位计算机链接的接口,使上位机能进行监视或程序控制。 |
| host link | |
| HR 区 | 在电源中断时保持位状态的内存区,在编程时用作工作位。 |
| HR area | |

| | |
|---|--|
| I/O 位 I/O bit | 内存中用于保持 I/O 状态的一个位。输入位反映输入端子的状态,输出位保持用于输出端子的状态。 |
| I/O 容量 I/O capacity | PC 能够处理的输入和输出点数。它的范围从小型 PC 的 100 点至大型机的 2000 点。 |
| I/O 延时 I/O delay | 从信号送到一个输出到输出状态受到实际影响之间的延时,或从一个输入状态的变化到接收到的状态指示出这个变化之间的延时。 |
| I/O 设备 I/O device | 与 I/O 单元上的 I/O 端子相连接的设备。它可能是控制系统的一部分,如果它的功能是帮助控制其他设备的,可能是受控系统的一部分。 |
| I/O 中断 I/O interrupt | 由 I/O 信号引发的一个中断。 |
| I/O 点 I/O point | 是输入信号接入 PC 系统或从 PC 系统中引出输出信号的地方。从物理角度讲,I/O 点是对应于单元的接线端或连接器的引脚,从编程角度讲,I/O 点对应 IR 区中的 I/O 位。 |
| I/O 刷新 I/O refreshing | 更新送到外部器件的输出状态,使它与保存在内存中输出位的状态一致,以及更新内存中的输入位,使它与输入端的外部器件的状态一致的过程。 |
| I/O 响应时间 I/O response time | 从接收外部器件的输入信号到将响应这个输入信号的输出信号送出所需的时间。 |
| I/O 单元 I/O Unit | PC 中与 I/O 设备相连接以输入和输出信号的单元。I/O 单元包括输入单元和输出单元。每种单元各有一定范围的各种规格。 |
| I/O 字 I/O word | 是 IR 区的一个字,它分配给 PC 系统中的一个单元,它用于保持该单元 I/O 的状态。 |
| IBM PC/AT 或兼容机 IBM PC/AT or compatible | 结构相似,逻辑上兼容,能运行为 IBM PC/AT 计算机设计的软件的计算机。 |
| 递增 increment | 数值递增,通常为 1。 |
| 间接地址 indirect address | 地址内容指示另一个地址,这第二个地址的内容是实际的操作数。 |
| 初始化错误 initialization error | 在 PC 系统启动(即初始化)时所出现的硬件或软件方面的错误。 |
| 初始化 initialize | 启动过程的一部分,这时某些内存区被清除,检查系统设置和设置缺省值。 |
| 输入 input | 从外部器件进入 PC 的信号。术语输入在理论上常用于进入的信号。 |

| | |
|---|--|
| 输入位 input bit | 在 IR 区中的一个位,分配用于保持一个输入状态。 |
| 输入设备 input device | 将信号送入 PC 系统的一个外部设备。 |
| 输入点 input point | 输入信号进入 PC 系统的地方,输入点在物理上对应于一个连接端子或一个连接器的引脚。 |
| 输入信号 input signal | 进入 PC 的一个连接状态的变化。例如,在一个连接点上电压由低到高变化或者由断开到接通状态,则说明该输入信号存在。 |
| 安装 install | 在计算机上使用一个程序或软件包如 LSS,SSS 必须做的准备。 |
| 指令 instruction | 在程序中给出的指导,告诉 PC 所要执行动作和执行动作所需要使用的数据。指令可以是仅仅将一位转为 ON 或 OFF,或完成许多复杂得多的动作,如,转换和/或传送一个大的数据块。 |
| 指令块 instruction block | 在梯形图程序中一组相互间有逻辑关系的指令。一个逻辑块包括从连接到左边总线一行或多行指令到连接到右边总线的一行或多行指令间互相连接的所有指令行。 |
| 指令执行时间 instruction execution time | 执行一条指令所需的时间。对任一条指令其执行时间随其执行条件和所用的操作数不同而变化。 |
| 指令行 instruction line | 处于梯形图同一水平行中的一组条件。指令行可以分支或汇合而构成指令块。也称为梯级。 |
| 接口 interface | 接口是各系统或器件之间的边界,通常包括通信数据表示方法的转换。接口设备要完成代码、格式或数据速率的转换。 |
| 连锁 interlock | 用于处理作为一组的数条指令的指令组编程方法,不需要逐条执行,它们可以一起复位。通常当执行条件为 ON 时,执行这个连锁程序段,在条件为 OFF 时部分复位。 |
| 中断(信号) interrupt(signal) | 使正常程序停止执行,引发一个子程序的投运或发生其他处理的信号。 |
| 中断程序 interrupt program | 响应中断后执行的程序。 |
| 非条件 inverse condition | 见常闭条件。 |
| JIS | 是日本工业标准(Japanese Industrial Standards)的缩写。 |
| 跳转 jump | 一种编程方式,这里使程序的执行直接从程序中某一点跳到另一点,这二点之间的指令不再逐条执行。 |

| | |
|--|---|
| 跳转号 jump number | 跳转命令使用的一个定义符,它定义跳转起点和跳转点。 |
| 梯形图(程序) ladder diagram(program) | 一种程序的形式。它源于继电器控制系统,它使用电路图形式来表示编程指令的逻辑流程。程序的样子象梯子,故有此名。 |
| 梯形图符号 ladder diagram symbol | 用于绘制梯形图程序的符号。 |
| 梯形图指令 ladder instruction | 在梯形图程序中表示条件的指令,其它在梯形图中的位于右边的指令称为终端指令。 |
| 梯形图支持软件 Ladder Support Software | 安装在 IBM PC/AT 机或兼容机上,使它成为编程设备的一个软件包。 |
| 最低有效(位/字) least - significant (bit/ word) | 见最右位/字。 |
| LED | 是发光二极管(light - emitting diode)的缩写,用于指示或显示的器件。 |
| 最左(位/字) leftmost(bit/word) | 一组数位中的最高位,一般是一个整字,或一组字中的最高数值字。这些位/字常称为最高有效或位最高有效字。 |
| 链接 link | 建立在二个单元之间的硬件或软件连接,“链接”可以是二单元之间物理连接的一部分,或是与存在另一处的数据的一个软件连接(即数据链接) |
| 装载 load | 从外部器件或存贮区向系统的一个工作部分如显示缓冲器复制数据的过程称装载,另外与 PC 连接的一个输出器件也称为装载。 |
| 逻辑块 logic block | 在梯形图程序中,相互有逻辑关系,并且需要逻辑块指令使之与其他指令或逻辑块建立关系的一组指令。 |
| 逻辑块指令 logic block instruction | 用于将一个逻辑块引出的执行条件与当前执行条件相组合的指令。当前执行条件可能是一个单条件的结果,或是其他逻辑块的结果。AND LOAD 和 OR LOAD 是这二种逻辑块指令。 |
| 逻辑指令 logic instruction | 用于二个字内容的逻辑组合并且将结果输出到指定结果字中的指令。逻辑指令组合了二个字中所有相同编号的位,并且将结果输出到指定结果字的相同的位中。 |
| LR 区 LR area | 用于数据链接的一个数据区。 |
| LSS | 见梯形图支持软件。 |
| 主程序 main program | 除子程序和中断程序以外的所有程序。 |
| 标记跟踪 mark trace | 在程序执行时,记录指定内存位置的存贮内容的修改,此过程称标记跟踪。 |

| | |
|------------------------------------|--|
| 被屏蔽位 | 状态暂时变得无效的位。 |
| masked bit | |
| 屏蔽 | “覆盖”一个中断信号使它不起作用,直至此屏蔽取消。 |
| masking | |
| 兆字节 | 存储量单位,等于一百万字节。 |
| megabyte | |
| 内存区 | 在 PC 中用于保持数据或程序的任一区域。 |
| memory area | |
| 消息号 | 分配给由消息指令产生的一个消息的编号。 |
| message number | |
| 助记符代码 | 由顺序的指令清单组成(不用梯形图)的一种梯形图程序形式。 |
| mnemonic code | |
| 监视模式 | 一种 PC 运行模式,此时能执行正常程序并能修改内存中的数据,用于 PC 监控或程序调试。 |
| MONITOR mode | |
| 最高有效位/字 | 见最左位/字 |
| most - significant (bit/ word) | |
| NC 输入 | 常闭输入,即当接到输入的回路开路时认为有输入信号。 |
| NC input | |
| 负延时 | 为数据跟踪设置的一种延时,要在跟踪信号达到指定量之前记录数据。 |
| negative delay | |
| 嵌套 | 编程使一个回路在另一个回路内,在一个子程序内调用另一个子程序,或在一个跳转内进行另一个跳转。 |
| nesting | |
| NO 输入 | 常开输入,即当连接到输入端的回路闭合时认为有输入信号。 |
| NO input | |
| 噪声干扰 | 由电噪声引起的信号中的干扰。 |
| noise interference | |
| 非致命错误 | 引起报警,但 PC 不停止运行的硬件或软件错误。 |
| nonfatal error | |
| 正常条件 | 见常开条件。 |
| normal condition | |
| 常闭条件 | 当所分配的位是 OFF 时产生 ON 执行条件,而当所分配的位是 ON 时产生 OFF 的执行条件。 |
| normally closed condition | |
| 常开条件 | 当所分配的位是 ON 时产生 ON 执行条件,而当所分配的位是 OFF 时产生 OFF 执行条件。 |
| normally open condition | |
| NOT | 对操作数的状态取反的逻辑操作。如 AND NOT 表示取与操作数位实际状态的反码操作。 |

| | |
|---|---|
| OFF | 信号不存在时输入或输出的状态。一般用低电压或不导通表示 OFF 状态,但也可作相反的定义。 |
| OFF 延时 OFF delay | 在信号被切换为 OFF(如由一个输入器件或 PC)到信号到达可以由接收部分(如输出器件或 PC)读出为 OFF 信号(即作为无信号)的时刻之间的时间间隔。 |
| 偏置 offset | 在基值上如地址上附加一个正值或负值,以指向所要的地址。 |
| ON | 表示一个信号为存在时输入或输出的状态。一般用高电压或导通表示 ON 状态,但也可作相反的定义。 |
| ON 延迟 ON delay | 从 ON 信号被启动(如,由输入器件或 PC 发出)到信号到达由接收部分(如,输出器件或 PC)可以作为 ON 信号读出时刻的时间。 |
| 单发位 one - shot bit | 转为 ON 或 OFF 持续大于一个扫描周期间隔时间的位。 |
| 1 对 1 链接 one - to - one link | 见 1:1 链接 |
| 在线编辑 online edit | 由编程设备直接修改 PC 内程序的过程。PC 必须处于编程或监视模式下才可进行在线编辑。在监视模式下意味着是对实际执行的程序进行修改。 |
| 操作数 operand | 这个值是作为指令的数据使用。操作数可以作为一个常数输入,表示所用的实际数值,或作为一个地址输入,表示所用数据在内存中的地址。 |
| 操作数位 operand bit | 一个指定的位,作为一条指令的操作数。 |
| 操作数字 operand word | 一个指定的字,作为一条指令的操作数。 |
| 操作模式 operating modes | 三种 PC 运行模式:编程模式、监视模式和运行模式。 |
| 操作错误 operating error | PC 在实际运行中出现的错误。与此相对的是初始化错误,它是在 PC 能进入实际运行前出现的错误。 |
| 或 OR | 一种逻辑运算,如果二个前提有一个是真或二者都是真时,结果就是真。在梯形图编程中,前提通常是位的 ON/OFF 状态,或是这些称为执行条件的状态的逻辑组合。 |
| 输出 output | 信号由 PC 送往外部设备。术语输出在理论上经常用于表示输出信号。 |
| 输出位 output bit | 在 IR 区中的一个位,用于保持送到输出设备的状态。 |
| 输出设备 output device | 接收来自 PC 系统信号的一个外部设备。 |
| 输出点 output point | PC 系统送出输出的点。在物理上输出点对应于端子或连接器的引脚。 |

| | |
|--------------------------------|---|
| 输出信号 output signal | 一个要发送到外部器件去的信号。一般来说,称存在输出信号,是在连接点能从低电压转入高电压,或由断开到导通状态。 |
| 溢出 overflow | 数据存贮器单元容量超出的状态。 |
| 监视 overseeing | CPU 完成的一部分处理,包括 PC 运行所需的通用任务。 |
| 改写 overwrite | 改写内存某单元的内容,因此先前的内容消失。 |
| 校验 parity | 调整在一个字或其他数据单元中 ON 位的数目,以使总数始终是偶数或奇数。校验一般用于在数据被传送后,通过检查 ON 位的数目是否仍为偶数或奇数来确定数据是否正确。 |
| 奇偶校验 parity check | 通过校验奇偶性以保证传送的数据的正确性。 |
| PC | 见可编程控制器。 |
| PC 配置 PC configuration | 单元之间的按排和互相连接,从而形成功能正常的 PC。 |
| PC 系统 PC System | 包括积木式 PC,所有与 I/O 设备连接的独立单元,但不包括 I/O 器件。PC 系统的边界上端是 PC 和在 CPU 中的程序,各 I/O 单元在它的下端。 |
| PCB | 见印刷电路板。 |
| PC 设置 PC Setup | 通过编程设备在 PC 内设置控制 PC 运行的参数。 |
| 外围设备 Peripheral Device | 与 PC 系统连接,参与系统运行的设备。外围设备包括打印机、编程器、外部存贮介质等。 |
| 外围服务 peripheral servicing | 处理与外围设备交换的信号,包括刷新、通信处理、中断等。 |
| 端口 port | 在 PC 或计算机上的一个插口,用于连接外部设备。 |
| 正延时 positive delay | 以数据跟踪而设置的延时,它要求在跟踪信号开始后,经过指定值的延时开始记录数据。 |
| 电源单元 Power Supply Unit | 与一台 PC 连接向其他单元提供所需电压的电源。 |
| 现在值 present value | 在设备运行时随时登录在其内的当前值。现在值缩写为 PV。这个术语一般限于用于定时器和计数器。 |
| 印刷电路板 printed circuit board | 在上面印刷有电子线路,用于装入计算机或电子设备的板。 |

| | |
|--|--|
| 编程模式 PROGRAM mode | PC 的一种运行模式,这时可以输入和调试要执行的程序,但不允许正常执行程序。 |
| 可编程序控制器 Programmable Controller | 一个计算机化的装置,它按保存在内存中的程序,能从外部输入信号,并向外部设备发出输出作用。可编程序控制器用于对外部设备进行自动控制。PC 可以是一单元的,也可以由各种部件构成的模块式的 PC,所以只有当这些足够的分离的部件组装成一个功能组件后,才形成一个 PC。 |
| 程控报警 programmed alarm | 在程序中设计为产生报警的指令,由于其执行的结果给出的报警,称程序报警,与此相对的是系统给出的报警。 |
| 程控错误 programmed error | 由于执行了用来产生错误指示的程序指令的结果而产生的错误,与此相对的是系统产生的错误。 |
| 程控消息 programmed message | 在程序中设计为产生消息的指令执行后给出的消息,与此相对的是系统产生的消息。 |
| 编程器 Programming Console | 用于 PC 的手握式编程设备。 |
| 编程设备 Programming Device | 用于将程序输入 PC,或改变和监控已保存在 PC 中的程序的外部设备。有专用的编程器件,如编程器,或不是专用的,如上位计算机。 |
| PROM | 可编程的只读存储器,是一种 ROM。它是在生产后由用户写入程序或数据,但从此后便固定下来。 |
| 提示 prompt | 显示在显示屏的消息或符号,请求操作员输入。 |
| 协议 protocol | 使二个设备能进行通讯或使编程员或操作员能与一个器件通讯的标准化的参数和步骤。 |
| PV | 见现在值。 |
| RAM | 随机存取存储器,一种数据存贮介质,它所保存的数据在停电时不能保持。 |
| RAS | 是可靠、保险和安全的首字母缩写。 |
| 只读区 read - only area | 一个存贮区,用户只能读出它的状态而不能写入信息。 |
| 刷新 refresh | 更新送到外部设备的输出状态使它与保存在内存中的输出值的状态一致,同时更新内存中输入位的状态使它与来自外部设备的输入状态一致的过程。 |
| 基于继电器控制 relay - based control | PC 的前身是继电器控制中一组继电器互相连接构成控制电路。在 PC 中这些已被可编程电路代替。 |
| 保留位 reserved bit | 用户应用不能使用的位。 |

| | |
|--------------------------------------|--|
| 保留字 reserved word | 在内存中的一个字,保留用于某种专门用途,用户不能访问的字。 |
| 复位 reset | 将一个位或信号转为 OFF,或将一个定时器或计数器的现行值改为设定值或零的过程。 |
| 响应代码 response code | 为响应一个数据传送而发送的代码,它表示传送的数据如何被处理。 |
| 响应格式 response format | 标明响应一个数据传送所要求的数据格式。 |
| 响应监视时间 response monitoring time | 一个器件在发送数据后等待响应信息的时间。如果经此时间仍未有回答可确定为出错。 |
| 重启动位 Restart Bit | 用于重新启动 PC 部分的一个位。 |
| 结果字 result word | 用于保持一条指令执行结果的字。 |
| 恢复 retrieve | 将外部设备或存贮区的数据复制到系统的工作区,如显示缓冲区。但将数据复制到与 PC 连接的输出设备称装载。 |
| 重试 retry | 由于前一次发送的数据在接收器中产生了出错信息,在发送端重新发送一次的过程。 |
| 返回 return | 指令的执行由一个子程序返回主程序的过程(通常返回调用此子程序的点)。 |
| 可逆计数器 reversible counter | 根据规定的条件可以是加计数和减计数的计数器。 |
| 可逆移位寄存器 reversible shift register | 根据指定条件可以双向移动数据的移位寄存器。 |
| 右侧指令 right - hand instruction | 见终端指令。 |
| 最右位/字 rightmost (bit/word) | 一组位中的最低位,通常是整个字中的最低位,或一组字中的最低的字,这个位/字常称为最低有效位/字。 |
| 上升沿 rising edge | 一个信号从 OFF 到 ON 状态变化的时刻。 |
| ROM | 只读存贮器,一种不能写入的数字存贮器。ROM 芯片的程序和数据是在制造时就存入的,是不可改变的,但是其程序或数据可以读出无论多少遍。 |
| 循环移位寄存器 rotate register | 一种移位寄存器。在这种寄存器中,从一端移出的数据又从另一端移入。 |
| RS - 232C 接口 RS - 232C interface | 一种串行通信工业标准。 |
| 运行模式 RUN mode | PC 在正常工作时所使用的操作模式。 |

| | |
|------------------------|---|
| 梯级 | 见指令行。 |
| rung | |
| 扫描 | 用于执行梯形图程序的过程。程序从开始到结束,每条指令按照执行条件依次执行。 |
| scan | |
| 扫描时间 | 见循环时间。 |
| scan time | |
| 定时中断 | 一种中断方式。它是由系统按指定时间自动发出的,或由操作员指定的程序单元发出的。定时中断致使指定子程序的执行,它可用于必须按规定的时间内间隔重复执行的指令。 |
| scheduled interrupt | |
| SCP | 见减计数输入。 |
| 封口 | 见自保持位。 |
| seal | |
| 自诊断 | 系统检查自己的工作,如果发现不正常就发出报警或错误信号。 |
| self diagnosis | |
| 自保持位 | 一个编程的位,保持它在 OFF 或 ON 状态,直到按指定的条件将它置位或复位。 |
| self - maintaining bit | |
| 串联 | 一种接线方法。这种方法将单元连续地联成一串。 |
| series | |
| 服务 | PC 检查一个连接器或单元以确定它们是否需要专门处理的过程。 |
| servicing | |
| 置位 | 将一个位或某一信号转为 ON 状态的过程。 |
| set | |
| 设定值 | 递减计数器从此值开始递减计数,或递增计数加到此目标值(即最大计数值),或定时器从此值或以此值开始计时。SV 是设定值的缩写。 |
| set value | |
| 移位输入信号 | 一种输入信号,当它由 OFF 变为 ON 时,使数据移动一位。 |
| shift input signal | |
| 移位寄存器 | 一个或多个字的寄存器,数据在其中可以向左或向右移动指定的单位(位、数字、或字为单位)。在循环移位寄存器中,数据从一端移出同时从另一端移入。在其它移位寄存器中,新的数据(也可是指定的数据,或 0,或 1)从一端移入,在另一端移出并丢失。 |
| shift register | |
| 有符号二进制数 | 存在内存中的一个二进制数附带一个符号位,它表示这个值是正数还是负数。 |
| signed binary | |
| 软件错误 | 源于软件程序的错误。 |
| software error | |
| 软件保护 | 通过软件防止数据被改写,而不是通过物理开关或其他硬件设置来达到该目的。 |
| software protect | |

| | |
|--|--|
| 源(字) source(word) | 取出提供指令使用的数据地址。与此相反的是 指令运算结果写入的地方,称为目的地址。 |
| 特殊指令 special instruction | 一种带功能代码的指令输入,它能在梯形图中完成数据处理操作,与此相对的是基本指令,它完成梯形图中的基本操作部分。 |
| SR 区 SR area | 包含标志和其他专用功能位/字的一个内存区。 |
| SSS | 见 SYSMAC 支持软件。 |
| 存储 store | 将写入显示缓冲区的一个程序永久地记录在存储器的过程。 |
| 子程序 subroutine | 与主程序分开存放的一组指令。只有在主程序调用它或由中断调用时才执行。 |
| 子程序号 subroutine number | 用于区别子程序调用或中断激活子程序的定义符。 |
| 减计数输入 subtract count input | 当信号从 OFF 转 ON 时,使计数器做减法计算的一个输入信号。 |
| SV | 见设定值。 |
| 开关能力 switching capacity | 继电器能够安全切换的最大电压/电流。 |
| 同步执行 synchronous execution | 在程序执行和服务操作中,程序执行和服务操作是同步的,也就是在每次执行程序时所有服务操作也同时实施。 |
| 句法 syntax | 程序语句的格式(与此相对的是其意义)。 |
| 语法错误 syntax error | 一种程序编写方式上的错误,语法错误也包括“拼写”错误(即出现无法实施的功能代码)、在可接受的参数范围内指定操作数的错误(如将只读位指定为一目标位)以及指令的实际应用方面的错误(如调用一个不存在的子程序)。 |
| SYSMAC 支持软件 SYSMAC Support Software | 安装在 IBM PC/AT 或兼容机上的用作编程设备的软件包。 |
| 系统配置 system configuration | 单元在系统连接中的安排。这个术语指的是概念上的安排和将系统中所需要的所有器件连接在一起。 |
| 系统错误 system error | 由系统产生的错误,与此对应的是由于执行了为产生错误指示的指令而产生的错误。 |
| 系统错误消息 system error message | 由系统产生的错误消息。与此相对的是由于执行了为产生错误消息号的指令而产生的消息。 |
| 系统设置 system setup | 编程设备的操作环境设定,如 LSS、SSS。 |

| | |
|-------------------------------|--|
| 终端指令 terminal instruction | 在一个梯形图右边,使用指令行中最终执行条件的一条指令。 |
| 定时器 timer | 在内存中的一存储单元,可以通过一个 TIM/CNT 位访问,并从定时器的设定值随时间递减。按它的执行条件可将定时器转为 ON 和复位。 |
| TR 区 TR area | 用于存贮执行条件的一个数据区,因此可以重新装载供其他指令使用。 |
| TR 位 TR bit | TR 区中一个位。 |
| 跟踪 trace | 一种操作,在跟踪时每一步程序执行的结果都存贮起来供一步一步分析和调试。 |
| 跟踪内存 trace memory | 用于存贮跟踪操作结果的一个内存区。 |
| 传送 transfer | 将数据从 PC 内的一个存储单元传送到另一个存储单元,或在 PC 和各外部设备之间传送的过程。数据传送时,通常数据的副本被传送到目的地,即传送源的内容不变。 |
| 传送距离 transmission distance | 信号能够传送的距离。 |
| 触发器 trigger | 用于激活某个过程的信号,如跟踪操作的执行。 |
| 触发地址 trigger address | 在程序中的一个地址,它定义跟踪的起点。通过定义一个正延时或负延时还可以修改实际触发点。 |
| UM 区 UM area | 用于保存现行即现时正在执行的程序的内存区。 |
| 单元 Unit | 在 OMRON PC 术语中,单元(Unit)这个字采用大写是表示作为 PC 系统出售的任一产品。这些产品的名称大部分以“单元”结尾。 |
| 单元号 Unit number | 分配给某些单元一个编号,以便在分配字或其他工作参数时容易辨认。 |
| 未屏蔽位 unmasked bit | 一个位它的状态是有效的。见屏蔽位。 |
| 无符号二进制 unsigned binary | 存贮在内存中的一个二进制数值,它没有正负值指示。 |
| 上载 uploading | 从低级或从站计算机向高级或上位计算机传送程序或数据,如果包含一个编程设备,则编程设备也看作上位计算机。 |
| 监视定时器 watchdog timer | 系统内部的一个定时器,用它保证扫描时间在指定的时间限制以内。当达到限定时间时,根据达到的时间限度,或发出报警或停止 PC 的运行。 |

| | |
|---|---|
| WDT | 见监视定时器。 |
| 字 word | 存在内存中由 16 位组成的一个数据单位,所有数据区都由字组成。某些数据区只能以字为单位存取,其它的或用字或用位存取。 |
| 字地址 word address | 一个字的数据在内存中的贮存位置,一个字地址必须标明数据区(有时这可以缺省)和作为寻址根据的字和编号。 |
| 工作区 work area | 包含工作字/位的一部分内存。 |
| 工作位 work bit | 工作字中的一个位。 |
| 工作字 work word | 能用于数据计算或在编程中的其他管理的一个字,即内存中的一个“工作空间”。IR 区中大部分留作工作字用,其他区的未派专门用途的也可作为工作字用。 |
| 写保护开关 write protect switch | 用于存贮设备内容的写保护的开关。例如,对于软盘,如果软盘左上角的孔是打开的,那么这个软盘上的信息就不能改写。 |
| 写保护 write - protect | 一种状态,在此状态下一个存贮设备中的内容只能读出而不能改写。 |

索引

数字

- 1:1 Data Link, 134
- 1:1 数据链接, 134
- 1:1 NT Link, 134
- 1:1 NT 链接, 134
- 1:N NT Link, 134
- 1:N NT 链接, 134

A

- absolute encoder inputs, specifications, 115
- 绝对值编码器输入, 规格, 115
- Absolute Encoder Interface Board, 113 – 125
- 绝对值编码器接口板, 113 – 125
 - components, 115
 - 元件, 115
 - configuration, 114
 - 配置, 114
 - flags and bits, 116, 147, 155, 516, 523
 - 标志和位, 116, 147, 155, 516, 523
 - functions, 114
 - 功能, 114
 - high – speed counter interrupts, 117
 - 高速计数器中断, 117
 - installation, 114
 - 安装, 114
 - settings, 10
 - 设置, 10
- absolute high – speed counter, reading status, 123
- 绝对值高速计数器, 读状态, 123
- ACC(—), 98, 107
- ADBL(—), 53
- address tracing. See tracing, data tracing.
- 地址跟踪。参见跟踪, 数据跟踪
- Analog I/O Board, 127 – 132
- 模拟量 I/O 板, 127 – 132
 - components, 129
 - 元件, 129
 - flag and bits, 147, 516
 - 标志和位, 147, 516
 - installation, 128

- 安装, 128
- settings, 11
- 设置, 11
- specifications, 130
- 配置, 130
- Analog Setting Board, 126 – 127
- 模拟设置板, 126 – 127
 - components, 127
 - 元件, 127
 - flags and bits, 146, 148, 515, 517
 - 标志和位, 146, 148, 515, 517
 - functions, 126
 - 功能, 126
 - installation, 126
 - 安装, 126
 - specifications, 127
 - 规格, 127
- applications, precautions, xvi
- 应用程序, 预防, xvi
- AR area, 152
- AR 区, 152
- arithmetic flags, 53, 202
- 算术标志, 53, 202
- ASCII, converting data, 289, 290
- ASCII, 转换数据, 289, 290

B – C

- bits, controlling, 214
- 位, 控制, 214
- check levels, program checks, 487
- 检验级, 程序检验
- checksum, calculating frame checksum, 373
- 检验和, 计算帧检验和, 373
- clock
- 时钟
 - reading the clock, 159, 529
 - 读时钟, 159, 529
 - setting the clock, 159, 529
 - 时钟设置, 159, 529
- communication errors, 490
- 通信错误, 490

communications

通信

Host Link, 44

上位机链接, 44

node number, 45

节点号, 45

link

链接

NT Link, 51

NT 链接, 51

one - to - one, 49

一对一, 49

no - protocol, 46

无协议, 46

one - to - one, 49

一对一, 49

PC Setup, 42

PC 设置, 42

wiring, 51

配线, 51

communications functions, 41

通信功能, 41

Communications Units, flags and Bits, 148, 517

通信单元, 标志和位, 148, 517

comparison, starting comparison operation, 72, 121

比较, 启动比较操作, 72, 121

compensation value, 121

补偿值, 121

constants, operands, 202

常数, 操作数, 202

Controller Link System, instructions, 393

Controller Link 系统, 指令, 393

converting. See data, converting

转换。参见数据, 转换

counters

计数器

conditions when reset, 226, 228

复位时条件, 226, 228

creating extended timers, 227

创建扩展定时器, 227

reversible counters, 227

可逆计数器, 227

cycle monitor time, PC Setup settings, 16

循环监视时间, PC 设置设定, 16

cycle time

循环时间

calculating, 466

计算, 466

effects on operations, 466

对操作数的影响, 466

processes, 466

过程, 466

cycle time (minimum). PC Setup settings, 13

循环时间(最小), PC 设置设定, 13

D

data

数据

converting, radians and degrees, 348, 349

转换, 弧度和角度, 348, 349

decrementing, 363

递减, 363

incrementing, 363

递增, 363

data tracing, 367 - 393

数据跟踪, 367 - 393

DBS(—), 53

DBSL(—), 53

decrementing. See data

递减, 见数据

definers, definition, 20

定义符, 定义, 20

degrees, converting degrees to radians, 348

角度, 角度转换到弧度, 348

differentiated instructions, 203

微分指令, 203

function codes, 201

功能代码, 201

DM area, 161

DM 区, 161

duty factor

占空因数

fixed, 97

固定, 97

pulse with variable duty factor, 389

可变占空率的脉冲, 389

variable, 109

可变, 109

E

EC, Directives, xix

EC, 规范, xix

EM area, 162

EM 区, 162

end codes, 427 - 429

结束代码

EPROM ICs. See Memory Cassettes

EPROM IC, 参见存储器盒

error codes, programming, 221

错误代码, 编程, 221

error log, 488

错误日志, 488

PC Setup settings, 16

PC 设置设定

error messages, programming, 369

错误消息, 编程, 369

errors

错误

1:1 links, 50

1:1 链接, 50

communications, 490

通信, 490

fatal, 491

致命, 491

general, 486

一般的, 486

non - fatal, 489

非致命, 489

programming, 487

编程, 487

Programming Console operations, 486

编程器操作, 486

programming messages, 369

编程消息, 369

resetting, 222

复位, 222

types, 486

类型, 486

user - defined errors, 488

用户定义错误, 488

execution condition, definition, 170

执行条件, 定义, 170

expansion instructions, 504

扩展指令, 504

function codes, 203

功能代码, 203

exponents, 357

指数, 357

F

FAL area, 221

FAL 区, 221

FAL(06), 488

FALS(07), 488

flags

标志

arithmetic, programming example, 269, 272

算术, 编程例, 269, 272

CY

clearing, 305

清除, 305

setting, 305

置位, 305

error and arithmetic, 507

错误和算术, 507

signed binary arithmetic, 507

有符号二进制计算, 507

floating - point data, 336

浮点数据, 336

exponents, 357

指数, 357

floating - point math instructions, 335 - 360

浮点算术指令, 335 - 360

logarithms, 359

对数, 359

square roots, 356

平方根, 356

Frame Check sequence. See frames. FCS

帧校验序列, 参见帧, FCS

frame checksum, calculating with FCS(-), 373

帧校验和,用 FCS(-)计算

frames

帧

dividing

除

See also host link

又见上位机链接

precautions, 433

注意事项, 433

FCS, 433

function codes, 201

功能代码, 201

expansion instructions, 203, 204

扩展指令, 203, 204

H

high - speed timers, PC Setup settings, 14

高速定时器, PC 设置设定, 14

High - speed Counter Board, 56 - 79

高速计数器板, 56 - 79

changing PV, 74

改变 PV, 74

checking PV, 63

检验 PV, 63

components, 57

元件, 57

configuration, 56

配置, 56

count frequencies, 60 - 63, 71

计数频率, 60 - 63, 71

count modes, 56, 61

计数模式, 56, 61

counting modes, 60, 62, 71

计数模式, 60, 62, 71

flags and bits, 145, 146, 153, 514, 515, 522

标志和位, 145, 146, 153, 514, 515, 522

functions, 56

功能, 56

installation, 57

安装, 57

instructions, 58

指令, 58

numeric ranges, 62

数字范围, 62

range comparison method, 63 - 64

区域比较方法, 63 - 64

reading counter status, 67

读计数器状态, 67

reading PV, 73

读 PV, 73

related bits and flags, 58 - 59

相关位和标志, 58 - 59

reset bits, 63

复位位, 63

reset methods, 71

复位方法, 71

resetting counters, 60, 63, 89

复位计数器, 60, 63, 89

settings, 9

设定, 9

PC Setup, 60

PC 设置, 60

specifications, 58 - 63

规格, 58 - 63

target value method, 63 - 64

目标值方法, 63 - 64

high - speed counters

高速计数器

specifications, 82

规格, 82

stopping and restarting operation, 75

停止和再起动手操作, 75

high - speed counters 1 and 2, counting modes
(numeric ranges)

高速计数器 1 和 2, 计数模式(数字范围)

hold bit status, PC Setup settings, 12

保持位状态, PC 设置设定, 12

Host Link, 44, 134

上位机链接, 44, 134

command and response formats, 431

命令和响应格式, 431

communications

通信

- See also Host Link commands
及见上位机链接命令
- methods, 429
方法, 429
- PC transmission, 434
PC 传输, 434
- Procedures, 45, 429
过程, 45, 429
- data transfer, 430
数据传送, 430
- dividing frames, 432
分割帧, 432
- frame
帧
- definition, 430
定义, 430
 - maximum size, 430
最大值, 430
- node number, 45
节点号, 45
- setting parameters, start and end codes, 47
设定参数, 启动和结束代码, 47
- Host Link commands
上位机链接命令
- ** , 458
 - EX, 459
 - FK, 453
 - IC, 459
 - KC, 454
 - KR, 452
 - KS, 451
 - MF, 450
 - MM, 455
 - MS, 448
 - QQ, 456
 - R # , 443
 - R \$, 444
 - R % , 445
 - RC, 436
 - RD, 437
 - RE, 437
 - RG, 436
 - RH, 436
 - RJ, 438
 - RL, 435
 - RP, 456
 - RR, 435
 - SC, 449
 - TS, 455
 - W # , 446
 - R \$, 446
 - W % , 447
 - WC, 440
 - WD, 441
 - WE, 442
 - WG, 440
 - WH, 439
 - WJ, 442
 - WL, 439
 - WP, 456
 - WR, 438
 - XZ, 458
- HR area, 152
HR 区, 152
- |
- I/O bits, 138
I/O 位, 138
- I/O points, refreshing, 370
I/O 点, 刷新
- I/O refresh operations, types, 463
I/O 刷新操作, 类型, 463
- I/O response time
I/O 响应时间
- See also timing
又见时序
 - one - to - one link communications, 480
一对一链接通信, 480
- I/O words, allocating, 138
I/O 字, 定位, 138
- incrementing, 363
递增, 363
- indirect addressing, 202
间接地址, 202

- INI(61), 39, 98
 Inner Board, PC Setup settings, 8
 内板, PC 设置设定, 8
 input time constants, PC Setup settings, 13
 输入时间常数, PC 设置设定, 13
 installation, precautions, xvi
 安装, 注意事项, xvi
 instruction set, 501
 指令集, 501
 + F(—), 343
 - F(—), 345
 * F(—), 346
 /F(—), 347
 7SEG(88), 412
 ACC(—), 98, 107, 387
 ACOS(—), 354
 ADB(50), 316
 ADBL(—), 53, 320
 ADD(30), 305
 ADDL(54), 310
 AND, 172, 213
 combining with OR, 173
 与 OR 组合, 173
 AND LD, 175, 214
 combining with OR LD, 178
 与 OR LD 组合, 178
 use in logic blocks, 177
 在逻辑块中使用, 177
 AND NOT, 172, 213
 ANDW(34), 360
 ASC(86), 289
 ASFT(17), 256
 ASIN(—), 353
 ASL(25), 251
 ASR(26), 251
 ATAN(—), 355
 AVG(—), 329
 BCD(24), 280
 BCDL(59), 281
 BCMP(68), 270
 BCNT(67), 373
 BIN(23), 279
 BINL(58), 280
 BSET(71), 260
 CLC(41), 305
 CMND(—), 400
 CMP(20), 268
 CMPL(60), 272
 CNT, 226
 CNTR(12), 227
 COLL(81), 263
 COM(29), 360
 COS(—), 351
 CPS(—), 274
 CPSL(—), 275
 CTBL(63), 232
 CTW(—), 300
 DBS(—), 53, 325
 DBSL(—), 53, 326
 DEC(39), 363
 DEG(—), 349
 DIFD(14), 190, 217 - 225
 using in interlocks, 219
 在联锁时使用, 219
 using in jumps, 221
 在跳转中使用, 221
 DIFU(13), 190, 217 - 225
 using in interlocks, 219
 在联锁时使用
 using in jumps, 221
 在跳转中使用
 DIST(80), 261
 DIV(33), 309
 DIVL(57), 314
 DMPX(77), 284
 DSW(87), 15, 415
 DVB(53), 319
 END(01), 175, 218
 EXP(—), 357
 FAL(06), 221
 FALS(07), 221
 FCS(—), 373
 FIX(—), 340
 FIXL(—), 341
 FLT(—), 342
 FLTL(—), 343
 FPD(—), 375
 HEX(—), 290

- HKY(—), 419
 HTS(65), 298
 IL(02), 186, 218 – 220
 ILC(03), 186, 218 – 220
 INC(38), 363
 INI(61), 39, 98, 244
 INT(89), 26, 379
 IORF(97), 370
 JME(05), 220
 JMP(04), 220
 JMP(04), 和 JME(05), 188
 KEEP(11), 216
 in controlling bit status, 190
 用于控制位状态, 190
 ladder instructions, 171
 梯形指令, 171
 LD, 172, 213
 LD NOT, 172, 213
 LOG(—), 359
 MAX(—), 327
 MBS(—), 53, 323
 MBSL(—), 53, 324
 MCMP(19), 273
 MCRO(99), 371
 MIN(—), 328
 MLB(52), 318
 MLPX(76), 282
 MOV(21), 257
 MOVB(82), 265
 MOVD(83), 266
 MSG(46), 369
 MUL(32), 308
 MULL(56), 313
 MVN(22), 258
 NEG(—), 53, 302
 NEGL(—), 53, 303
 NOP(00), 218
 NOT, 169
 operands, 168
 操作数, 168
 OR, 173, 213
 与 AND 组合
 OR LD, 176, 214
 combining with AND, 178
 与 AND LD 组合, 178
 use in logic blocks, 177
 在逻辑块中使用, 177
 OR NOT, 173, 213
 ORW(35), 361
 OUT, 174, 214
 OUT NOT, 174, 214
 PID(—), 392
 PLS2(—), 98, 106, 385
 PMCR(—), 410
 PMW(—), 109
 PRV(62), 112, 122, 246
 PULS(65), 98, 103, 380
 PWM(—), 389
 RAD(—), 348
 RECV(98), 397
 RESET, 189
 RET(93), 17, 367
 ROL(27), 252
 ROOT(72), 315
 ROR(28), 252
 RSET, 215 – 216
 RXD(47), 403
 SBB(51), 317
 SBBL(—), 53, 321
 SBN(92), 367
 SBS(91), 365
 SCL(66), 293
 SCL2(—), 295
 SCL3(—), 296
 SDEC(78), 286
 SEND(90), 393
 SET, 189, 215 – 216
 SFT(10), 249
 SFTR(84), 254
 SIN(—), 350
 SLD(74), 253
 SNXT(09), 222
 SPED(64), 39, 98, 103, 105, 382

- SQRT(—), 356
 SRCH(—), 390
 SRD(75), 254
 STC(40), 305
 STEP(08), 222
 STH(—), 299
 STIM(69), 28, 231
 STUP(—), 407
 SUB(31), 306
 SUBL(55), 312
 SUM(—), 330
 TAN(—), 352
 TCMP(85), 269
 terminology, 168
 术语, 168
 TIM, 225
 TIMH(15), 228
 TKY(18), 422
 TRSM(45), 367
 TTIM(—), 229
 TXD(48), 405
 VCAL(—), 332
 WSFT(16), 250
 WTC(—), 301
 XCHG(73), 261
 XFER(70), 259
 XFRB(—), 267
 XNRW(37), 362
 XORW(36), 362
 ZCP(—), 277
 ZCPL(—), 278
- instructions
- 指令
- execution times, 467
 - 执行时间, 467
 - expansion, 203
 - 扩展, 203
 - floating - point math instructions, 335 - 360
 - 浮点数数学指令, 335 - 360
 - instruction set lists, 208
 - 指令集列表, 208
 - mnemonics list, ladder, 209
 - 助记符列表, 梯形, 209
 - right - hand instructions, coding multiple, 184
 - 右侧指令, 复合代码, 184
 - subroutines, 365
 - 子程序, 365
- tables
- 表
- default, 205
 - 缺省, 205
 - user - set, 205
 - 用户设定, 205
- INT(89), 26
- interlocks, 218 - 220
- 互锁, 218 - 220
- using self - maintaining bits, 191
 - 使用自保持位, 191
- interrupt functions, 17
- 中断功能, 17
- interrupt processing
- 中断过程
- calculating response time, 483
 - 计算响应时间, 483
 - masking, 483
 - 屏蔽, 483
 - timing, 482
 - 时间, 482
- interrupts
- 中断
- absolute high - speed counters, programming, 121
 - 绝对高速计数器, 编程, 121
 - control, 379
 - 控制, 379
 - counter mode, 23
 - 计数器模式, 23
 - high - speed counter
 - 高速计数器
 - overflows and overflows, 38
 - 上溢和下溢, 38
 - programming, 34
 - 编程, 34
 - high - speed counter 0, 29
 - 高速计数器 0, 29

overflows and overflows, 37

上溢和下溢, 37

high-speed counters 1 and 2, 88

高速计数器 1 和 2, 88

input, parameters, 19

输入, 参数, 19

input interrupts, 19

输入中断, 19

interval timers, 27

间隔定时器, 27

scheduled interrupt mode, 28

定时中断模式, 28

masking, 26

屏蔽, 26

setting modes, 23

设定模式, 23

types, 17

类型, 17

unmasking, 26

去除屏蔽, 26

J - L

jump numbers, 220

跳转号, 220

jumps, 220 - 221

跳转, 220 - 221

ladder diagram

梯形图

branching, 184

分支, 184

IL(02) and ILC(03), 186

IL(02) 和 ILC(03), 186

using TR bits, 185

使用 TR 位, 185

combining logic blocks, 179

组合逻辑块, 179

controlling bit status

控制位状态

using DIFU(13) and DIFD(14), 190, 217 - 225

使用 DIFU(13) 和 DIFD(14), 190, 217 - 225

using KEEP(11), 216 - 217

使用 KEEP(11), 216 - 217

using OUT and OUT NOT, 174

使用 OUT 和 OUT NOT, 174

using SET and RESET, 189

使用 SET 和 RESET, 189

using SET and RSET, 215 - 216

使用 SET 和 RSET, 215 - 216

converting to mnemonic code, 170 - 189

转换成助记符代码

display via CX - Programmer, 169

通过 CX - Programmer 显示, 169

instructions

指令

combining, AND LD and OR LD, 178

组合 AND LD 和 OR LD, 178

controlling bit status

控制位状态

using KEEP(11), 190

使用 KEEP(11), 190

using OUT and OUT NOT, 214

使用 OUT 和 OUT NOT, 214

format, 201

格式, 201

notation, 201

进制, 201

structure, 169

结构, 169

using logic blocks, 175

使用逻辑块, 175

ladder diagram instructions, 213 - 214

梯形图指令, 213 - 214

logarithm, 359

对数, 359

logic block instructions, converting to mnemonic code, 175 - 183

逻辑块指令, 转换成助记符代码, 175 - 183

logic blocks. See ladder diagram

逻辑块, 见梯形图

M

macro function, subroutines. See programming

宏函数子程序, 见编程

masking, interrupt processes, 483

屏蔽, 中断过程, 483

mathematics

数学

See also trigonometric functions

又见三角函数

exponents, 357

指数, 357

floating - point addition, 343

浮点加, 343

floating - point division, 347

浮点除, 347

floating - point math instructions, 335 - 360

浮点数学指令, 335 - 360

floating - point multiplication, 346

浮点乘, 346

floating - point subtraction, 345

浮点减, 345

logarithm, 359

对数, 359

square root, 356

平方根, 356

MBS(—), 53

MBSL(—), 53

memory areas

存储器区

AR area bits, 152, 521

AR 区位, 152, 521

DM area, 161

DM 区, 161

EM area, 162

EM 区, 162

flags, 149

标志, 149

flags and bits(SR area), 518

标志和位(SR 区), 518

HR area, 152

HR 区, 152

IR area bits, 138

IR 区位, 138

link bits, 160

链接位, 160

structure, 136, 511

结构, 136, 511

timer and counter bits, 160

定时器和计数器位, 160

TR bits, 152

TR 位, 152

work bits, 138

工作位, 138

Memory Cassettes, 161

存储器盒, 161

and program size, 163

和程序大小, 163

automatic transfer at startup, 166

在启动时自动传送, 166

comparing contents, 166

比较内容, 166

contents, 163

内容, 163

reading data, 165

读数据, 165

reading with peripherals, 165

用外围读, 165

required EPROMs, 163

要求 EPROM, 163

storing DM and UM data, 162

存储 DM 和 UM 数据, 162

types, 162

类型, 162

writing data, 165

写数据, 165

messages, programming, 369

消息, 编程, 369

minimum cycle time, PC Setup settings, 13

最小循环时间, PC 设置设定, 13

mnemonic code, converting, 170 - 189

助记符代码, 转换, 170 - 189

momentary power interruption, 463

电源暂时中断, 463

MSG(46), 488

N

NEG(—), 53

NEGL(—), 53

nesting, subroutines, 366
 嵌套, 子程序, 366

no - protocol communications, 134
 无协议通信, 134

- receiving, 48
接收, 48
- transmitting, 47
传送, 47

normally open/closed condition, definition, 169
 常开/闭条件, 定义, 169

NOT, definition, 169
 NOT, 定义, 169

NT Link, 51
 NT 链接, 51

O

one - to - one link communications
 一对一链接通信

- I/O response timing, 480
I/O 响应时间, 480
- link errors, 50
链接错误, 50

one - to - one link, 49
 一对一链接, 49

operand bit, 170
 操作数位, 170

operands, 201
 操作数, 201

- allowable designations, 201
允许指定, 201
- requirements, 201
要求, 201

operating environment, precautions, xvi
 操作环境, 注意事项, xvi

operations
 操作

- effects on cycle time, 466
影响循环时间, 466
- internal processing, flowchart, 462
内部过程, 流程图, 462

origin compensation, 121
 原点补偿, 121

output bit
 输出位

- controlling ON/OFF time, 215
控制 ON/OFF 时间, 215
- controlling status, 189, 191
控制状态

output refresh method, PC Setup settings, 15
 输出刷新方法, PC 设置设定, 15

outputs, turning OFF, 490
 输出, 置 OFF, 490

P

PC Setup, See settings
 PC 设置, 见设定

peripheral port, servicing time, 13
 外围端口, 服务时间, 13

peripheral port servicing time, PC Setup settings, 13
 外围端口服务时间, PC 设置设定, 13

PLS2(—), 98, 106

PMW(—), 109

power interruptions
 电源中断

- momentary interruptions, 463
暂时中断, 463
- Programmable Controller, 463
可编程控制器, 463

power OFF processing, 463 - 465
 断电过程, 463 - 465

precautions, xiii
 注意事项, xiii

- applications, xvi
应用, xvi
- general, xiv
一般, xiv
- operating environment, xvi
操作环境, xvi
- safety, xiv
安全, xiv

Program Memory, structure, 170
 程序存储器, 结构, 170

programming
 编程

- absolute high – speed counters, 121
 - 绝对高速计数器, 121
 - errors, 487
 - 错误, 487
 - high – speed counter 0, 34
 - 高速计数器 0, 34
 - high – speed counters 1 and 2, 92
 - 高速计数器 1 和 2, 92
 - instructions, 501
 - 指令, 501
 - interrupts, 34, 92, 121
 - 中断, 34, 92, 121
 - jumps, 188
 - 跳转, 188
 - macro function, subroutines, 371
 - 宏功能, 子程序, 371
 - precautions, 193
 - 注意事项, 193
 - preparing data in data areas, 260
 - 数据区中准备数据, 260
 - special features, 203
 - 特性, 203
 - writing, 168
 - 写, 168
 - programs
 - 程序
 - checking, check levels, 487
 - 校验, 校验级, 487
 - executing, 195
 - 执行, 195
 - PROTOCOL MACRO instruction, 410
 - 协议宏指令, 410
 - protocol macros, 134
 - 协议宏, 134
 - PRV(62), 112, 122
 - PULS(65), 98, 103
 - Pulse I/O Board, 79 – 113
 - 脉冲 I/O 板, 79 – 113
 - configuration, 80
 - 配置, 80
 - count modes, 79
 - 计数模式, 79
 - flags and bits, 147, 154, 516, 522
 - 标志和位, 147, 154, 516, 522
 - indicators
 - 指示灯
 - pulse inputs, 82
 - 脉冲输入, 82
 - pulse outputs, 81
 - 脉冲输出, 81
 - installation, 81
 - 安装, 81
 - interrupts, 79 – 113
 - 中断, 79 – 113
 - pulse input indicators, 82
 - 脉冲输入指示灯, 82
 - pulse output indicators, 81
 - 脉冲输出指示灯, 81
 - settings, 10
 - 设定, 10
 - pulse inputs, flags and control bits, 82
 - 脉冲输入, 标志和控制位, 82
 - pulse outputs
 - 脉冲输出
 - determining status of ports 1 and 2, 112
 - 判断端口 1 和 2 的状态, 112
 - fixed – duty – factor, 97
 - 固定占空率, 97
 - flags and control bits, 86
 - 标志和控制位, 86
 - from ports 1 and 2, 97
 - 从端口 1 和 2, 97
 - functions, 97
 - 功能, 97
 - variable – duty – factor, 109
 - 可变占空率, 109
 - PV
 - CNTR(12), 227
 - timers and counters, 225
 - 定时器和计数器, 225
- ## R
- radians, converting radians to degrees, 349
 - 弧度, 转换弧度为角度, 349

response codes, 427 - 429

响应代码, 427 - 429

RET(93), 17

right - hand instructions, coding, See instructions

右侧指令, 代码。见指令

RS - 232C Port

RS - 232C 端口

connecting Units, 49

连接单元, 49

control bits, 48

控制位, 48

one - to - one link, 49

一对一链接, 49

servicing time, 12

服务时间, 12

RS - 232C port servicing time, PC Setup settings, 12

RS - 232C 端口服务时间, PC 设置设定, 12

S

safety precautions, See precautions

安全注意, 见注意事项

SBBL(一), 53

Serial Communications Board, 132 - 134

串行通信板, 132 - 134

flags and bits, 143, 513

标志和位, 143, 513

settings, 3, 8

设定, 3, 8

serial communications functions, 41

串行通信功能, 41

serial communications modes

串行通信模式

1:1 Data Link, 134

1:1 数据链接, 134

1:1 NT Link, 134

1:1 NT 链接, 134

1:N NT Link, 134

1:N NT 链接, 134

Host Link, 134

上位机链接, 134

no - protocol, 132, 134

无协议, 132, 134

protocol macro, 134

协议宏, 134

settings

设定

basic operations

基本操作

error log, 16

错误日志, 16

high - speed timers, 14

高速定时器, 14

hold bit status, 12

保持位状态, 12

input digits number, 15

输入数字号, 15

startup mode, 11

启动模式, 11

changing, 2

改变, 2

communications, 41

通信, 41

Host Link, 44

上位机链接, 44

no - protocol, 46

无协议, 46

PC Setup, 42

PC 设置, 42

defaults, 2

缺省, 2

expansion instructions, 205

扩展指令, 205

high - speed counters 1 and 2, 92

高速计数器 1 和 2, 92

I/O operations, 11

I/O 操作, 11

port servicing scan time, 12, 13

端口服务扫描时间, 12, 13

pulse output word, 40, 102, 110

脉冲输出字, 40, 102, 110

Inner Board, 8

内板, 8

interrupts, 17

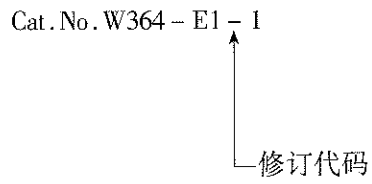
中断, 17

- external sources, 19
 - 外部源, 19
 - parameters, 22
 - 参数, 22
 - PC Setup settings, 4 - 8
 - PC 设置设定, 4 - 8
 - pulse outputs, 38
 - 脉冲输出, 38
 - seven - - segment displays, converting data, 286
 - 7 段显示, 转换数据, 286
 - signed binary arithmetic flags, 507
 - 有符号二进制算术标志, 507
 - signed binary data, 51
 - 有符号二进制数据, 51
 - specifications
 - 规格
 - absolute encoder inputs, 115
 - 绝对编码器输入, 115
 - Analog I/O Board, 130
 - 模拟量 I/O 板, 130
 - Analog Setting Board, 127
 - 模拟设定板, 127
 - High - speed Counter Board, 58
 - 高速计数器板, 58
 - high - speed counters, 82
 - 高速计数器, 82
 - pulse outputs, 84
 - 脉冲输出, 84
 - SPED (64), 39, 98, 103, 105
 - square root, floating - point data, 356
 - 平方根, 浮点数据, 356
 - SR area, 149
 - SR 区, 149
 - srartup mode, PC Setup settings, 11
 - 启动模式, PC 设置设定, 11
 - STIM (69), 28
 - subroutine number, 367
 - 子程序号, 367
 - SV
 - CNTR (12), 227
 - timers and counters, 225
 - 定时器和计数器, 225
 - SYSMAC WAY, See Host Link
 - SYSMAC WAY 见上位机链接
- ## T
- TIM/CNT, timer/counter area, 160
 - TIM/CNT 定时器/计数器区, 160
 - TIM/CNT numbers, 224
 - TIM/CNT 号, 224
 - time
 - 时间
 - reading the time, 159, 529
 - 读时间, 159, 529
 - setting the time, 159, 529
 - 设定时间, 159, 529
 - timer/counter area, 160
 - 定时器/计时器区, 160
 - timers, conditions when reset, 225, 228
 - 定时器, 复位时条件, 225, 228
 - TTIM(一), 230
 - timing
 - 时序
 - basic instructions, 468
 - 基本指令, 468
 - cycle time, 466
 - 循环时间, 466
 - I/O response time, 479
 - I/O 响应时间, 479
 - instruction execution, See instruction
 - 指令执行, 见指令
 - interrupt processing, 482
 - 中断过程, 482
 - special instructions, 468
 - 特殊指令, 468
 - TR area, 152
 - TR 区, 152
 - TR bits, use in branching, 185
 - TR 位, 在分支中使用, 185
 - tracing, See See data tracing and address tracing
 - 跟踪, 见数据跟踪和地址跟踪
 - trigonometric functions
 - 三角函数

- arc cosine, 354
- 反余弦, 354
- arc sine, 353
- 反正弦, 353
- arc tangent, 355
- 反正切, 355
- converting degrees to radians, 348
- 将角度转换到弧度, 348
- converting radians to degrees, 349
- 将弧度转换到角度, 349
- cosine, 351
- 余弦, 351
- sine, 350
- 正弦, 350
- tangent, 352
- 正切, 352
- troubleshooting, 485
- 故障诊断, 485
 - flowcharts, 493
 - 流程图, 493

修订历史

手册封面上商品目录号的后缀为手册修订代码。



下表概括了各修订版的手册所做的修改。页号参照以前版本。

| 修订代码 | 日期 | 修订内容 |
|------|---------|------|
| 1 | 1999年9月 | 原版 |



**Innovation
in the Solution Age**

OMRON

欧姆龙(中国)有限公司
欧姆龙亚洲有限公司

| | | | |
|-------|----------------|-------|----------------|
| 上海办事处 | 021 - 50372222 | 广州办事处 | 020 - 87320508 |
| 南京办事处 | 025 - 4726876 | 厦门办事处 | 0592 - 5117709 |
| 武汉办事处 | 027 - 65776566 | 西安办事处 | 029 - 5381152 |
| 苏州办事处 | 0512 - 8669277 | 成都办事处 | 028 - 6765345 |
| 北京办事处 | 010 - 83913005 | 重庆办事处 | 023 - 63803720 |
| 山东办事处 | 0531 - 2929795 | 昆明办事处 | 0871 - 5366019 |
| 辽宁办事处 | 024 - 22566105 | | |

授权经销商

技术 咨 询

电子邮件: omron@omron.com.cn
网 址: <http://www.omron.com>
800 免费技术咨询电话: 800-820-4535

规格可能改变,恕不另行通知。

样本编号 OEZ - ZDP00101A

欧姆龙(中国)有限公司 版权所有

上海印刷
200207 S 03