



## (12) 发明专利

(10) 授权公告号 CN 103562807 B

(45) 授权公告日 2016. 04. 27

(21) 申请号 201280026760. 9

(56) 对比文件

(22) 申请日 2012. 03. 12

JP 特开平 10-207724 A, 1998. 08. 07,  
US 2003/0084383 A1, 2003. 05. 01,  
CN 101526805 A, 2009. 09. 09,  
CN 201716564 A, 2011. 01. 19,  
JP 特开平 6-309010 A, 1994. 11. 04,

(30) 优先权数据

2011-156474 2011. 07. 15 JP

审查员 陈盈洁

(85) PCT国际申请进入国家阶段日

2013. 11. 29

(86) PCT国际申请的申请数据

PCT/JP2012/056243 2012. 03. 12

(87) PCT国际申请的公布数据

W02013/011713 JA 2013. 01. 24

(73) 专利权人 欧姆龙株式会社

地址 日本京都府京都市

(72) 发明人 西山佳秀 江口重行 滨崎治

(74) 专利代理机构 隆天知识产权代理有限公司

72003

代理人 金相允 向勇

(51) Int. Cl.

G05B 19/05(2006. 01)

G06F 9/48(2006. 01)

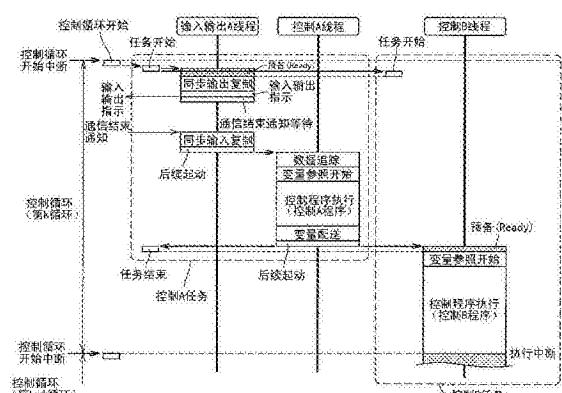
权利要求书4页 说明书36页 附图54页

(54) 发明名称

PLC 的 CPU 单元、PLC 系统、PLC 辅助装置

(57) 摘要

系统程序包含过程库 (procedure library), 该过程库中至少有用于对控制程序的执行进行控制的过程 (procedure) 和用于对输出数据的输出及输入数据的输入进行控制的过程。系统程序在包含收录在过程库中的过程的执行顺序的指定的排程构建数据保存在存储单元中时, 使微处理器按照包含在排程构建数据中的过程的执行顺序的指定来执行收录在过程库中的过程。



1. 一种 PLC 的 CPU 单元, 是用于对控制对象进行控制的 PLC(1) 的 CPU 单元 (13), 其特征在于,

具有:

微处理器 (100),

存储单元 (104、106),

通信电路 (120、140);

所述存储单元, 用于保存系统程序 (210)、控制程序 (230) 及排程构建数据 (224);

所述微处理器, 执行保存在所述存储单元中的所述系统程序及所述控制程序;

所述通信电路, 发送通过执行所述控制程序而生成的输出数据, 并且接收在执行所述控制程序时要使用的输入数据;

所述系统程序包含过程库 (214), 该过程库 (214) 中至少收录有用于对所述控制程序的执行进行控制的过程和用于对所述输出数据的输出及所述输入数据的输入进行控制的过程;

在所述存储单元中存储有包含用于对收录在所述过程库中的过程的执行顺序进行指定的指定信息的所述排程构建数据时, 所述系统程序使所述微处理器按照包含在所述排程构建数据中的针对过程的执行顺序的指定信息, 来执行收录在所述过程库中的过程。

2. 如权利要求 1 所述的 PLC 的 CPU 单元, 其特征在于,

所述排程构建数据, 包含用于利用收录在所述过程库中的过程来生成线程的指定信息;

所述系统程序, 包含按照包含在所述排程构建数据中的用于生成所述线程的指定信息来生成线程的处理, 以作为所述 PLC 的控制动作的执行准备处理。

3. 如权利要求 2 所述的 PLC 的 CPU 单元, 其特征在于,

所述排程构建数据, 包含用于在线程的执行过程中起动其他线程的指定信息, 以作为与所述线程相关的数据。

4. 如权利要求 2 所述的 PLC 的 CPU 单元, 其特征在于,

所述排程构建数据, 用于对成为反复执行的单位的任务进行设定的指定信息, 并且该任务包含一个或多个所述线程;

所述系统程序, 包含按照所述排程构建数据中的用于设定所述任务的指定信息来设定任务的处理, 以作为所述 PLC 的控制动作的执行准备处理。

5. 如权利要求 1 所述的 PLC 的 CPU 单元, 其特征在于,

所述 PLC 的 CPU 单元, 还具有用于获取所述排程构建数据的排程构建数据获取单元。

6. 如权利要求 1 所述的 PLC 的 CPU 单元, 其特征在于,

在所述存储单元中, 保存有包含用于反复执行如下这些处理的所述针对过程的执行顺序的指定信息, 这些处理包括:

发送所述输出数据,

接收所述输入数据,

通过执行所述控制程序, 从而使用所述输入数据生成所述输出数据。

7. 如权利要求 6 所述的 PLC 的 CPU 单元, 其特征在于,

所述 PLC 的 CPU 单元还具有用于设定控制循环周期的单元;

所述通信电路,按照每个所述控制循环,发送所述输出数据及接收所述输入数据;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行如下处理:在所述控制程序已执行结束的控制循环的下一个控制循环中,在由所述通信电路执行发送输出数据及接收输入数据之后,使所述控制程序开始执行;在所述控制程序未执行结束的控制循环的下一个控制循环中,执行所述控制程序中的未执行的部分。

8. 如权利要求 6 所述的 PLC 的 CPU 单元,其特征在于,

所述系统程序包含以下这些处理,以作为所述 PLC 的控制动作的执行准备处理,这些处理包括:

在所述存储单元中生成用于保存所接收的输入数据的接收缓存的处理,

针对由所述控制程序参照的每个输入数据,在所述存储单元中生成由该控制程序作为输入数据的参照对象的输入同步缓存的处理;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行以下处理:

输入复制处理,将所接收的所述输入数据从所述接收缓存复制到与该输入数据相对应的所述输入同步缓存中,

控制程序开始处理,使所述控制程序开始执行。

9. 如权利要求 8 所述的 PLC 的 CPU 单元,其特征在于,

所述输入数据的接收,是按照每个控制循环执行的;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行如下处理:按照所述控制循环的整数倍的每个执行循环,执行所述控制程序开始处理;将在所述控制程序的执行循环开始后的所述控制循环中接收的输入数据作为对象,执行所述输入复制处理。

10. 如权利要求 8 所述的 PLC 的 CPU 单元,其特征在于,

所述系统程序还包含以下这些处理,以作为所述 PLC 的控制动作的执行准备处理,这些处理包括:

在所述存储单元中生成用于保存应发送的输出数据的发送缓存的处理,

在所述存储单元中,针对每个所述输出数据,生成用于保存复制到所述发送缓存中的所述输出数据的输出同步缓存的处理;

所述排程构建数据还包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行输出复制处理,该输出复制处理用于将所述输出数据从所述输出同步缓存复制到所述发送缓存中。

11. 如权利要求 10 所述的 PLC 的 CPU 单元,其特征在于,

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行如下处理:针对控制循环的整数倍的每个执行循环,执行所述控制程序开始处理;在所述控制程序的执行循环的最初的控制循环中,在之前的执行循环中通过执行所述控制程序而生成的所述输出数据被从所述发送缓存发送的时刻,执行所述输出复制处理。

12. 如权利要求 6 所述的 PLC 的 CPU 单元,其特征在于,

所述存储单元,还用于保存与所述控制程序所使用的变量相关的属性数据;

在所述变量是由多个所述控制程序参照的全局变量的情况下,所述属性数据包含能够改写该变量的一个主侧控制程序的指定信息以及仅能够参照该变量的一个或多个参照侧控制程序的指定信息;

所述系统程序包含以下处理,以作为所述 PLC 的控制动作的执行准备处理,这些处理包括:

在所述存储单元中生成用于保存所述全局变量的全局变量区域的处理,

针对各参照侧控制程序,按照由该参照侧控制程序参照的每个全局变量,在由该参照侧控制程序参照全局变量时,取代所述全局变量区域而在所述存储单元中生成作为参照对象的变量同步缓存;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行以下这些处理,这些处理包括:

主侧开始处理,使主侧控制程序开始执行,

变量复制处理,如果主侧控制程序执行结束,则将由该主侧控制程序改写的全局变量从所述全局变量区域复制到与该全局变量相对应的所述变量同步缓存中,

参照侧开始处理,使参照侧控制程序开始执行。

13. 如权利要求 6 所述的 PLC 的 CPU 单元,其特征在于,

所述控制程序包括:

动态运算程序,计算用于对电机的动作进行控制的动态指令值,

用户程序,是根据用户的控制目的而生成的程序,包含对所述动态运算程序赋予其执行所需的指示;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述 PLC 的 CPU 单元反复依次执行以下处理:利用所述通信电路发送所述输出数据以及接收所述输入数据、执行所述用户程序、执行所述动态运算程序。

14. 如权利要求 6 所述的 PLC 的 CPU 单元,其特征在于,

所述通信电路,以控制循环周期,发送所述输出数据并接收所述输入数据;

所述控制程序包含第一控制程序及第二控制程序;

所述第一控制程序包含第一动态运算程序,该第一动态运算程序用于生成用于对电机的动作进行控制的第一动态指令值数据;

所述第二控制程序包含第二动态运算程序,该第二动态运算程序用于生成用于对电机的动作进行控制的第二动态指令值数据;

所述第一控制程序及所述第二控制程序中的至少一个控制程序包含用户程序,该用户程序是根据用户的控制目的而生成的,包含向所述第一动态运算程序及所述第二动态运算程序赋予其执行所需的指示的指令;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于使所述微处理器执行以下这些处理,这些处理包括:

按照与所述控制循环周期相同的周期的每个第一执行循环,使所述第一控制程序开始执行,按照所述控制循环周期的二以上的整数倍的周期的每个第二执行循环,使所述第二控制程序开始执行,

在所述第二执行循环开始的所述控制循环中,在所述第一控制程序的执行结束后使所述第二控制程序开始执行,如果到该控制循环结束为止,所述第二控制程序尚未结束,则在下一个控制循环中,所述第一控制程序执行结束之后,使所述第二控制程序中的未执行的部分的执行开始。

15. 如权利要求 6 所述的 PLC 的 CPU 单元,其特征在于,

所述微处理器至少包含第一核及第二核;

所述控制程序包含第一控制程序和第二控制程序;

所述排程构建数据包含如下的所述针对过程的执行顺序的指定信息,该过程用于在发送完所述输出数据及接收完所述输入数据之后,使所述第一核执行所述第一控制程序,并且使所述第二核与所述第一控制程序的执行并行地执行所述第二控制程序。

16. 一种 PLC 系统,包含如权利要求 5 所述的 PLC(1) 的 CPU 单元 (13) 以及用于对所述 PLC 的使用进行辅助的 PLC 辅助装置 (8),其特征在于,

所述 PLC 辅助装置具有存储部 (82、83、84) 和运算部 (81);

所述存储部用于保存 PLC 辅助程序 (320) 及所述排程构建数据;

所述 PLC 辅助程序,使所述运算部执行输出处理,在该输出处理中,以使所述 PLC 的 CPU 单元能够获取所述排程构建数据的方式输出该排程构建数据。

17. 一种 PLC 辅助装置 (8),用于辅助使用 PLC(1) 的 CPU 单元 (13),该 PLC(1) 用于对控制对象进行控制,该 PLC 辅助装置 (8) 的特征在于,

所述 PLC 的 CPU 单元具有:

过程库 (214),其至少收录有用于对控制程序的执行进行控制的过程和用于对通过执行所述控制程序而生成的输出数据的输出及执行所述控制程序时使用的输入数据的输入进行控制的过程,

排程构建数据获取单元,其用于获取排程构建数据,该排程构建数据包含收录在所述过程库中的过程的执行顺序的指定信息;

所述 PLC 辅助装置具有存储部 (82、83、84) 和运算部 (81);

所述存储部,用于保存 PLC 辅助程序及所述排程构建数据;

所述 PLC 辅助程序,使所述运算部执行输出处理,在该输出处理中,以使所述 PLC 的 CPU 单元能够获取所述排程构建数据的方式输出该排程构建数据。

## PLC 的 CPU 单元、PLC 系统、PLC 辅助装置

### 技术领域

[0001] 本发明涉及用于对机械及器材（设备）等的动作进行控制的 PLC (Programmable Logic Controller, 或者还称为可编程控制器) 的控制动作的排程 (schedule)。

### 背景技术

[0002] PLC 例如包括具有用于执行控制程序的微处理器的 CPU (Central Processing Unit) 单元、负责从外部的开关或传感器输入信号并向外部的继电器或触动器输出信号的 I/O (Input Output : 输入 / 输出) 单元等多个单元。CPU 单元通过反复向其他单元输出输出数据，从其他单元输入输入数据，执行使用输入数据生成输出数据的控制程序，从而对控制对象进行控制。控制程序包含按照用户的控制目的来生成的用户程序。控制程序也可以包含在用户程序中的指示执行的动态运算程序中。

[0003] PLC 能够如下面例示那样采用各种方式，对输入数据的输入、输出数据的输出、控制程序及 PLC 系统程序的执行，进行排程。

[0004] 如在专利文献 1 (日本特开 2000-105604 号公报) 中示出那样，在以往的一般的 PLC 中，在各控制循环中执行一个序列程序 (sequence program) (控制程序) 和一组输出刷新 (refresh) 及输入刷新。在该情况下，控制循环与控制程序的执行循环相等。

[0005] 还已知在 PLC 中分时执行多个控制程序。

[0006] 在专利文献 2 (日本特开 2007-140655 号公报) 中记载有如下技术，即，在用一个 CPU 对用于控制电机的动态控制功能和执行序列运算的 PLC 功能进行处理的装置中，在每个基本时钟的一个循环中，执行“恒定周期动态控制处理及各轴处理”和“高速序列处理”，进而在各基本时钟循环中的剩余的时间内，执行“低速序列处理”或“非恒定周期动态控制处理”。另外，还记载有如下技术，即，在低速序列处理在基本时钟循环内未能结束的情况下，停止相当于规定的基本时钟次数的时间之后执行剩余的处理(例如，第 0004 段)。

[0007] 在专利文献 3 (日本特开 2000-293210 号公报) 中关于控制装置的动作记载有如下技术，即，与周期任务(控制程序)相独立地分别准备多个进行输入处理及输出处理的刷新块，并在各控制循环中分别选择性地执行一些刷新块和一些周期任务。

[0008] 现有技术文献

[0009] 专利文献

[0010] 专利文献 1 : 日本特开 2000-105604 号公报

[0011] 专利文献 2 : 日本特开 2007-140655 号公报

[0012] 专利文献 3 : 日本特开 2000-293210 号公报

### 发明内容

[0013] 发明要解决的问题

[0014] 近年来，在信息技术领域，微处理器及通信网络的高速化取得了进展。因此，PLC 也能够使用这些技术，来进行多个控制程序的分时执行，并且能够采用高速的 PLC 系统总

线或现场网络或这两者。从而,对装置的处理速度方面的限制已减小,以用于提高对如下的控制动作的排程的自由度,该控制动作包括输入数据的输入、输出数据的输出、控制程序及PLC系统程序的执行。在此,对控制动作的排程的自由度还包括将在哪个时刻进行的输入和输出与哪个控制程序的执行建立对应关联关系的情况。另外,根据具体的控制目的,存在各种最佳的控制动作的排程,因此优选对PLC产品的控制动作的排程的自由度高的一方。

[0015] 然而,在实际的PLC产品中,实际情况是,控制动作的排程被PLC产品的系统程序决定,即使能够从一些排程模式中的选择一个排程模式以及能够调整执行参数,也不能由用户自由建立控制动作的排程。

[0016] 在此,从一些排程模式中选择排程模式是指,例如从控制程序的执行循环中,提供以“输出 - 输入 - 控制程序执行 - 其他系统处理”的顺序执行还是以“输入 - 控制程序执行 - 输出 - 其他系统处理”的顺序执行这样的选项的程度。另外,执行参数的调整是指,在分时执行多个控制程序的情况下,对赋予各控制程序的执行优先级进行调整的程度。

[0017] 若要大幅变更控制动作的排程模式或将新追加的程序模块(例如,温度控制用程序模块、图像处理用程序模块等)所承担的处理嵌入到控制动作的排程中,则需要更替PLC产品的系统程序的至少一部分。这样一来,PLC产品会变成别的机种,因而对PLC产品的厂商及用户的管理负担会增大。另外,在与用户的控制目的相对应的最佳的排程模式为特殊的排程模式的情况下,现实上用户难以得到安装有那样的排程模式的PLC产品。

[0018] 另外,例如将如下结构作为多机种的PLC产品所共通包含的平台,则能够提高PLC产品的厂商的开发效率并降低成本,这种结构是指,在将PLC产品提供给用户的阶段以后不能变更控制动作的排程模式的情况下,使用共通的系统程序也能够实现多种排程模式的结构。

[0019] 本发明的目的在于,不需更替系统程序,或即使使用共通(通用)的系统程序,也能够实现对PLC的控制动作的排程的高自由度。

[0020] 用于解决问题的手段

[0021] 根据本发明的一个技术方案,提供用于对控制对象进行控制的PLC的CPU单元。PLC的CPU单元包含微处理器、存储单元及通信电路。存储单元用于保存系统程序、控制程序及排程构建数据。微处理器执行保存在存储单元中的系统程序及控制程序。通信电路发送通过控制程序的执行来生成的输出数据,并且接收执行控制程序时使用的输入数据。系统程序包含程序库(procedure library),该程序库中至少收录有用于对控制程序的执行进行控制的过程(procedure)和用于对输出数据的输出及输入数据的输入进行控制的过程。系统程序,在包含有对收录在程序库中的过程的执行顺序的指定的排程构建数据保存在存储单元中时,使微处理器按照包含在排程构建数据中的对过程的执行顺序的指定,来执行收录在程序库中的过程。

[0022] 优选地,排程构建数据,包含用于利用收录在程序库中的过程来生成线程的指定信息。系统程序,包含按照包含在排程构建数据中的用于生成线程的指定信息来生成线程的处理,以作为对PLC的控制动作的执行准备处理。

[0023] 更优选地,排程构建数据,包含用于在线程的执行过程中起动其他线程的指定信息,以作为与线程相关的数据。

[0024] 或优选地,排程构建数据,用于对成为反复执行的单位的任务进行设定的指定信

息,并且该任务包含一个或多个线程;系统程序,包含按照排程构建数据中的用于设定任务的指定信息来设定任务的处理,以作为 PLC 的控制动作的执行准备处理。

[0025] 优选地,PLC 的 CPU 单元,还具有用于获取排程构建数据的排程构建数据获取单元。

[0026] 优选地,在存储单元中,保存有包含用于反复执行如下这些处理的针对过程的执行顺序的指定信息,这些处理包括:发送输出数据,接收输入数据,通过执行控制程序,从而使用输入数据生成输出数据。

[0027] 更优选地,PLC 的 CPU 单元还具有用于设定控制循环周期的单元;通信电路,按照每个控制循环,发送输出数据及接收输入数据;排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行如下处理:在控制程序已执行结束的控制循环的下一个控制循环中,在由通信电路执行发送输出数据及接收输入数据之后,使控制程序开始执行;在控制程序未执行结束的控制循环的下一个控制循环中,执行控制程序中的未执行的部分。

[0028] 或优选地,系统程序包含以下这些处理,以作为 PLC 的控制动作的执行准备处理,这些处理包括:在存储单元中生成用于保存所接收的输入数据的接收缓存的处理;针对由控制程序参照的每个输入数据,在存储单元中生成由该控制程序作为输入数据的参照对象的输入同步缓存的处理。排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行以下处理:输入复制处理,将所接收的输入数据从接收缓存复制到与该输入数据相对应的输入同步缓存中,控制程序开始处理,使控制程序开始执行。

[0029] 更优选地,排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行如下处理:按照控制循环的整数倍的每个执行循环,执行控制程序开始处理;将在控制程序的执行循环开始后的控制循环中接收的输入数据作为对象,执行输入复制处理。

[0030] 或更优选地,系统程序还包含以下这些处理,以作为 PLC 的控制动作的执行准备处理,这些处理包括:在存储单元中生成用于保存应发送的输出数据的发送缓存的处理;在存储单元中,针对每个输出数据,生成用于保存复制到发送缓存中的输出数据的输出同步缓存的处理。排程构建数据还包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行输出复制处理,该输出复制处理用于将输出数据从输出同步缓存复制到发送缓存中。

[0031] 更优选地,排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行如下处理:针对控制循环的整数倍的每个执行循环,执行控制程序开始处理;在控制程序的执行循环的最初的控制循环中,在之前的执行循环中通过执行控制程序而生成的输出数据被从发送缓存发送的时刻,执行输出复制处理。

[0032] 或优选地,存储单元,还用于保存与控制程序所使用的变量相关的属性数据;在变量是由多个控制程序参照的全局变量的情况下,属性数据包含能够改写该变量的一个主侧控制程序的指定信息以及仅能够参照该变量的一个或多个参照侧控制程序的指定信息。系统程序包含以下处理,以作为 PLC 的控制动作的执行准备处理,这些处理包括:在存储单元中生成用于保存全局变量的全局变量区域的处理;针对各参照侧控制程序,按照由该参照侧控制程序参照的每个全局变量,在由该参照侧控制程序参照全局变量时,取代全局变量

区域而在存储单元中生成作为参照对象的变量同步缓存。排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行以下这些处理,这些处理包括:主侧开始处理,使主侧控制程序开始执行;变量复制处理,如果主侧控制程序执行结束,则将由该主侧控制程序改写的全局变量从全局变量区域复制到与该全局变量相对应的变量同步缓存中;参照侧开始处理,使参照侧控制程序开始执行。

[0033] 或优选地,控制程序包括:动态运算程序,计算用于对电机的动作进行控制的动态指令值;用户程序,是根据用户的控制目的而生成的程序,包含对动态运算程序赋予其执行所需的指示。排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使PLC的CPU单元反复依次执行以下处理:利用通信电路发送输出数据以及接收输入数据、执行用户程序、执行动态运算程序。

[0034] 或优选地,通信电路,以控制循环周期,发送输出数据并接收输入数据;控制程序包含第一控制程序及第二控制程序;第一控制程序包含第一动态运算程序,该第一动态运算程序用于生成用于对电机的动作进行控制的第一动态指令值数据;第二控制程序包含第二动态运算程序,该第二动态运算程序用于生成用于对电机的动作进行控制的第二动态指令值数据;第一控制程序及第二控制程序中的至少一个控制程序包含用户程序,该用户程序是根据用户的控制目的而生成的,包含向第一动态运算程序及第二动态运算程序赋予其执行所需的指示的指令;排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于使微处理器执行以下这些处理,这些处理包括:按照与控制循环周期相同的周期的每个第一执行循环,使第一控制程序开始执行,按照控制循环周期的二以上的整数倍的周期的每个第二执行循环,使第二控制程序开始执行;在第二执行循环开始的控制循环中,在第一控制程序的执行结束后使第二控制程序开始执行,如果到该控制循环结束为止,第二控制程序尚未结束,则在下一个控制循环中,第一控制程序执行结束之后,使第二控制程序中的未执行的部分的执行开始。

[0035] 或优选地,微处理器至少包含第一核及第二核;控制程序包含第一控制程序和第二控制程序;排程构建数据包含如下的针对过程的执行顺序的指定信息,该过程用于在发送完输出数据及接收完输入数据之后,使第一核执行第一控制程序,并且使第二核与第一控制程序的执行并行地执行第二控制程序。

[0036] 根据本发明的另一技术方案,提供一种PLC用的系统程序,在具有微处理器、存储单元及通信电路的用于对控制对象进行控制的PLC的CPU单元中,该PLC用的系统程序被保存在存储单元中并且被微处理器执行。存储单元,用于保存系统程序、控制程序及排程构建数据。微处理器,执行保存在存储单元中的系统程序及控制程序;通信电路,发送通过执行控制程序而生成的输出数据,并且接收在执行控制程序时要使用的输入数据。系统程序包含程序库,该程序库中至少收录有用于对控制程序的执行进行控制的过程和用于对输出数据的输出及输入数据的输入进行控制的过程;在存储单元中存储有包含用于对收录在程序库中的过程的执行顺序进行指定的指定信息的排程构建数据时,系统程序使微处理器按照包含在排程构建数据中的针对过程的执行顺序的指定信息,来执行收录在程序库中的过程。

[0037] 优选地,排程构建数据,包含用于利用收录在程序库中的过程来生成线程的指定信息;系统程序,包含按照包含在排程构建数据中的用于生成线程的指定信息来生成线程

的处理,以作为 PLC 的控制动作的执行准备处理。

[0038] 更优选地,排程构建数据,包含用于在线程的执行过程中起动其他线程的指定信息,以作为与线程相关的数据。

[0039] 根据本发明的又一技术方案,提供一种保存有 PLC 用的系统程序的记录介质,用于保存 PLC 用的系统程序,在具有微处理器、存储单元及通信电路的用于对控制对象进行控制的 PLC 的 CPU 单元中,该 PLC 用的系统程序被保存在存储单元中并且被微处理器执行。存储单元,用于保存系统程序、控制程序及排程构建数据。微处理器,执行保存在存储单元中的系统程序及控制程序。通信电路,发送通过执行控制程序而生成的输出数据,并且接收在执行控制程序时要使用的输入数据。系统程序包含程序库,该程序库中至少收录有用于对控制程序的执行进行控制的过程和用于对输出数据的输出及输入数据的输入进行控制的过程。在存储单元中存储有包含用于对收录在程序库中的过程的执行顺序进行指定的指定信息的排程构建数据时,系统程序使微处理器按照包含在排程构建数据中的针对过程的执行顺序的指定信息,来执行收录在程序库中的过程。

[0040] 优选地,排程构建数据,包含用于利用收录在程序库中的过程来生成线程的指定信息;系统程序,包含按照包含在排程构建数据中的用于生成线程的指定信息来生成线程的处理,以作为 PLC 的控制动作的执行准备处理。

[0041] 更优选地,排程构建数据,包含用于在线程的执行过程中起动其他线程的指定信息,以作为与线程相关的数据。

[0042] 根据本发明的又一技术方案,提供一种 PLC 系统,该 PLC 系统包含上述的 PLC 的 CPU 单元和用于对 PLC 的使用进行辅助的 PLC 辅助装置。PLC 辅助装置具有存储部和运算部。存储部用于保存 PLC 辅助程序及排程构建数据。PLC 辅助程序,使运算部执行输出处理,在该输出处理中,以使 PLC 的 CPU 单元能够获取排程构建数据的方式输出该排程构建数据。

[0043] 根据本发明的又一技术方案,提供一种 PLC 辅助装置,该 PLC 辅助装置用于对控制控制对象的 PLC 的 CPU 单元的使用进行辅助。PLC 的 CPU 单元具有:程序库,其至少收录有用于对控制程序的执行进行控制的过程和用于对提供控制程序的执行来生成的输出数据的输出及执行控制程序时使用的输入数据的输入进行控制的过程;排程构建数据获取单元,其用于获取排程构建数据,该排程构建数据包含收录在程序库中的过程的执行顺序的指定信息。PLC 辅助装置具有存储部和运算部。存储部,用于保存 PLC 辅助程序及排程构建数据。PLC 辅助程序,使运算部执行输出处理,在该输出处理中,以使 PLC 的 CPU 单元能够获取排程构建数据的方式输出该排程构建数据。

[0044] 根据本发明的又一技术方案,提供一种 PLC 辅助程序,在用于对控制控制对象的 PLC 的 CPU 单元的使用进行辅助的 PLC 辅助装置中被执行。PLC 的 CPU 单元具有:程序库,其至少收录有用于对控制程序的执行进行控制的过程和用于对通过执行控制程序而生成的输出数据的输出及执行控制程序时使用的输入数据的输入进行控制的过程;排程构建数据获取单元,其用于获取排程构建数据,该排程构建数据包含收录在程序库中的过程的执行顺序的指定信息。PLC 辅助装置具有存储部和运算部。存储部,用于保存 PLC 辅助程序及排程构建数据;PLC 辅助程序,使运算部执行输出处理,在该输出处理中,以使 PLC 的 CPU 单元能够获取排程构建数据的方式输出该排程构建数据。

[0045] 根据本发明的又一技术方案，提供一种保存有 PLC 辅助程序的记录介质，该 PLC 辅助程序在用于对控制控制对象的 PLC 的 CPU 单元的使用进行辅助的 PLC 辅助装置中被执行。PLC 的 CPU 单元具有：程序库，其至少收录有用于对控制程序的执行进行控制的过程和用于对通过执行控制程序而生成的输出数据的输出及执行控制程序时使用的输入数据的输入进行控制的过程；排程构建数据获取单元，其用于获取排程构建数据，该排程构建数据包含收录在程序库中的过程的执行顺序的指定信息。PLC 辅助装置具有存储部和运算部；存储部，用于保存 PLC 辅助程序及排程构建数据；PLC 辅助程序，使运算部执行输出处理，在该输出处理中，以使 PLC 的 CPU 单元能够获取排程构建数据的方式输出该排程构建数据。

[0046] 发明效果

[0047] 根据本发明，不需更替系统程序，或者即使使用共通的系统程序，也能够实现 PLC 的控制动作的排程的高自由度，。

## 附图说明

[0048] 图 1 是示出了本发明的实施方式的 PLC 系统的概略结构的示意图。

[0049] 图 2 是示出了本发明的实施方式的 CPU 单元的硬件结构的示意图。

[0050] 图 3 是示出了由本发明的实施方式的 CPU 单元执行的软件结构的示意图。

[0051] 图 4 是示出了本发明的实施方式的 CPU 单元的主存储器的区域结构的示意图。

[0052] 图 5 是示出了连接到本发明的实施方式的 CPU 单元上使用的 PLC 辅助装置的硬件结构的示意图。

[0053] 图 6 是示出了连接到本发明的实施方式的 CPU 单元上使用的 PLC 辅助装置的软件结构的示意图。

[0054] 图 7 是示出了本发明的第一实施方式的 CPU 单元的系统程序的整体处理的流程图。

[0055] 图 8 是以表形式示出了本发明的第一实施方式中准备的任务的设定项目和对各设定项目的说明的图。

[0056] 图 9A 是以表形式示出了本发明的第一实施方式中准备的线程的设定项目和对各设定项目的说明的图。

[0057] 图 9B 是以表形式示出了本发明的第一实施方式中准备的线程的设定项目和对各设定项目的说明的图。

[0058] 图 10 是对于本发明的第一实施方式中准备的过程(procedure)之一的控制循环开始过程示出了其功能、设定项目及对设定项目的说明的图。

[0059] 图 11A 是对于本发明的第一实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。

[0060] 图 11B 是对于本发明的第一实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。

[0061] 图 12 是示出了与本发明的第一实施方式的排程例 1 相对应的排程构建数据 234 的图。

[0062] 图 13 是本发明的第一实施方式的排程例 1 的第 k 循环的时序图。

[0063] 图 14 是本发明的第一实施方式的排程例 1 的第 k+1 循环的时序图。

[0064] 图 15 是示出了在本发明的第一实施方式的排程例 1 中控制 A 线程的执行在第 k 循环内未结束的情况的动作的时序图。

[0065] 图 16 是示出了在本发明的第一实施方式的排程例 1 中控制 A 线程的执行在第 k 循环内未结束的情况的动作的时序图。

[0066] 图 17 是用于说明本发明的第一实施方式中准备的同步输入复制过程的动作的时序图。

[0067] 图 18 是用于说明本发明的第一实施方式中准备的同步输出复制过程的动作的时序图。

[0068] 图 19 是用于说明本发明的第一实施方式中准备的变量配送(发送)过程及开始参照变量过程的动作的时序图。

[0069] 图 20 是示出了与本发明的第一实施方式的排程例 2 相对应的排程构建数据 234 的图。

[0070] 图 21 是本发明的第一实施方式的排程例 2 的第 k 循环的时序图。

[0071] 图 22 是示出了与本发明的第一实施方式的排程例 3 相对应的排程构建数据的图。

[0072] 图 23 是本发明的第一实施方式的排程例 3 的第 k 循环的时序图。

[0073] 图 24 是本发明的第一实施方式的排程例 3 的第 k+1 循环的时序图。

[0074] 图 25 是示出了与本发明的第一实施方式的排程例 4 相对应的排程构建数据的图。

[0075] 图 26 是本发明的第一实施方式的排程例 4 的第 k 循环的时序图。

[0076] 图 27 是本发明的第一实施方式的排程例 4 的第 k+1 循环的时序图。

[0077] 图 28 是示出了与本发明的第一实施方式的排程例 5 相对应的排程构建数据的图。

[0078] 图 29 是本发明的第一实施方式的排程例 5 的第 k 循环的时序图。

[0079] 图 30 是与本发明的第一实施方式的排程例 6 相对应的时序图。

[0080] 图 31 是用于说明在本发明的第一实施方式的排程例 6 中的控制程序中的程序结构和指示的流程的示意图。

[0081] 图 32 是示出了与本发明的第一实施方式的排程例 7 相对应的排程构建数据的图。

[0082] 图 33 是本发明的第一实施方式的排程例 7 的第 k 循环的时序图。

[0083] 图 34 是本发明的第一实施方式的排程例 7 的第 k+1 循环的时序图。

[0084] 图 35 是示出了与本发明的第一实施方式的排程例 8 相对应的排程构建 数据的图。

[0085] 图 36 是本发明的第一实施方式的排程例 8 的第 k 循环及第 k+1 循环的时序图。

[0086] 图 37 是示出了与本发明的第一实施方式的排程例 9 相对应的排程构建数据的图。

[0087] 图 38 是本发明的第一实施方式的排程例 9 的第 k 循环及第 k+1 循环的时序图。

[0088] 图 39A 是以表形式示出了本发明的第二实施方式中准备的线程的设定项目和对各设定项目的说明的图。

[0089] 图 39B 是以表形式示出了本发明的第二实施方式中准备的线程的设定项目和对各设定项目的说明的图。

[0090] 图 40A 是示出了本发明的第二实施方式中准备的过程的名称及功能的图。

[0091] 图 40B 是示出了本发明的第二实施方式中准备的过程的名称及功能的图。

[0092] 图 41 是示出了与本发明的第二实施方式的排程例相对应的排程构建数据的图。

- [0093] 图 42 是本发明的第二实施方式的排程例的第 k 循环的时序图。
- [0094] 图 43 是本发明的第二实施方式的排程例的第 k+1 循环的时序图。
- [0095] 图 44 是示出了本发明的第三实施方式的各过程共通具备的执行控制功能所提供的处理的流程图。
- [0096] 图 45 是对于本发明的第三实施方式中准备的过程之一的控制循环开始过程示出了其功能、设定项目及对设定项目的说明的图。
- [0097] 图 46A 是对于本发明的第三实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。
- [0098] 图 46B 是对于本发明的第三实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。
- [0099] 图 47 是示出了与本发明的第三实施方式的排程例相对应的排程构建数据的图。
- [0100] 图 48 是与图 47 的排程构建数据相对应的排程例的第 k 循环的时序图。
- [0101] 图 49 是与图 47 的排程构建数据相对应的排程例的第 k+1 循环的时序图。

## 具体实施方式

[0102] 参照附图,详细说明本发明的实施方式。此外,对图中的相同或等同的部分,标注相同的附图标记,不对其进行重复说明。

[0103] < A. 系统结构 >

[0104] 本实施方式的 PLC 对机械或器材(设备)等的控制对象进行控制。本实施方式的 PLC 包含作为其结构要素的 CPU 单元。CPU 单元包含微处理器、存储单元及通信电路。存储单元用于保存对程序的执行进行控制的系统程序、控制程序以及排程构建数据。排程构建数据包含用于以高自由度对本实施方式的 PLC 的控制动作进行排程的设定值等。在后面详细叙述该排程构建数据。

[0105] 微处理器执行保存在存储单元中的系统程序及控制程序。通信电路发送通过执行控制程序而生成的输出数据,并且接收执行控制程序时使用的输入数据。首先,参照图 1,说明本实施方式的 PLC1 的系统结构。

[0106] 图 1 是示出了本发明的实施方式的 PLC 系统的概略结构的示意图。参照图 1,PLC 系统 SYS 包括 PLC1、与 PLC1 经由现场网络 2 连接的伺服电机驱动器 3 及远程 I/O 终端 5、作为现场设备的检测开关 6 及继电器 7。另外,在 PLC1 上经由连接电缆 10 等连接有 PLC 辅助装置 8。

[0107] PLC1 包括主要执行运算处理的 CPU 单元 13、一个以上的 I/O 单元 14 以及特殊单元 15。这些单元能够经由 PLC 系统总线 11 相互交换数据。另外,电源单元 12 向这些单元供给适当电压的电源。此外,构成 PLC1 的各单元是由 PLC 厂商提供的,因而 PLC 系统总线 11 一般是每个 PLC 厂商独自开发并使用的。相对于此,如后述那样关于现场网络 2 则大多公开了其标准,以能够连接不同厂商的产品。

[0108] 参照图 2,在后面详细叙述 CPU 单元 13。

[0109] I/O 单元 14 是与一般的输入输出处理相关的单元,对开(ON)/闭(OFF)这样的二进制化的数据的输入输出进行控制。即, I/O 单元 14 收集这样的信息,该信息表示,处于由检测开关 6 等传感器检测出某种对象物的状态(开, ON)及未检测出任何对象物的状态(闭, OFF)

中的哪个状态这样的信息。另外，I0 单元 14 对继电器 7 或触动器这样的输出对象，输出用于激活的指令(开, ON)及用于取消激活的指令(闭, OFF)中的某个指令。

[0110] 特殊单元 15 具有不能在 I0 单元 14 中进行辅助的功能，这些功能是指模拟数据的输入输出、温度控制、通过特定的通信方式的通信这样的功能。

[0111] 现场网络 2 传送与 CPU 单元 13 交换的各种数据。作为现场网络 2，典型地能够利用各种工业以太网(注册商标)。作为工业以太网(注册商标)，例如已知有“EtherCAT”(注册商标)、“Profinet IRT”、“MECHATROLINK”(注册商标)-III、“Powerlink”，“SERCOS”(注册商标)-III、“CIP Motion”等，可以采用它们中的任何产品。进而，也可以使用工业以太网(注册商标)以外的现场网络。例如，在不进行动态控制的情况下，也可以使用“DeviceNet”、“CompoNet/IP”(注册商标)等。

[0112] 此外，在图 1 中，例示了具有 PLC 系统总线 11 及现场网络 2 这双方的 PLC 系统 SYS，但也可以采用安装有这双方中的一方的系统结构。例如，也可以用现场网络 2 连接全部的单元。或者，也可以不使用现场网络 2，而将伺服电机驱动器 3 直接连接到 PLC 系统总线 11 上。进而，也可以采用如下结构，即，将现场网络 2 的通信单元连接到 PLC 系统总线 11 上，CPU 单元 13 经由该通信单元与连接到现场网络 2 上的设备进行通信。

[0113] 此外，PLC1 也可以采用如下结构，即，使 CPU 单元 13 具备 I0 单元 14 的功能及伺服电机驱动器 3 的功能，由此不经由 I0 单元 14 及伺服电机驱动器 3 等，而由 CPU 单元 13 直接对控制对象进行控制。

[0114] 伺服电机驱动器 3 经由现场网络 2 与 CPU 单元 13 连接，并且按照来自 CPU 单元 13 的指令值来对伺服电机 4 进行驱动。更加具体地，伺服电机驱动器 3 从 PLC1 以规定周期接收位置指令值、速度指令值、扭矩指令值这样的指令值。另外，伺服电机驱动器 3 从与伺服电机 4 的轴相连接的位置传感器(旋转编码器)及扭矩传感器这样的检测器获取如位置、速度(典型地根据当前位置和前一次位置之间的差来计算)、扭矩这样的与伺服电机 4 的动作相关的实测值。然后，伺服电机驱动器 3 通过将来自 CPU 单元 13 的指令值设定为目标值，并将实测值作为反馈值，来进行反馈控制。即，伺服电机驱动器 3 对用于驱动伺服电机 4 的电流进行调整，以使实测值接近目标值。此外，还将伺服电机驱动器 3 称为伺服电机放大器。

[0115] 另外，在图 1 中示出了组合了伺服电机 4 和伺服电机驱动器 3 的系统的例子，但也可以采用组合了其他的结构例如脉冲电机和脉冲电机驱动器的系统。

[0116] 在图 1 所示的 PLC 系统 SYS 的现场网络 2 上还连接有远程 I0 终端(terminal)5。远程 I0 终端 5 基本上与 I0 单元 14 同样地进行与一般的输入输出处理相关的处理。更加具体地，远程 I0 终端 5 包括用于进行与现场网络 2 中的数据传送相关的处理的通信连接器 52 和一个以上的 I0 单元 53。这些单元能够经由远程 I0 终端总线 51 相互交换数据。

[0117] 在后面详细叙述 PLC 辅助装置 8。

[0118] < B. CPU 单元的硬件结构 >

[0119] 接着，参照图 2，说明 CPU 单元 13 的硬件结构。图 2 是示出了本发明的实施方式的 CPU 单元 13 的硬件结构的示意图。参照图 2，CPU 单元 13 包括微处理器 100、芯片集 102、主存储器 104、非易失性存储器 106、系统计时器 108、PLC 系统总线控制器 120、现场网络控制器 140 及 USB 连接器 110。在芯片集 102 与其他组件之间经由各种总线分别结合。

[0120] 微处理器 100 及芯片集 102 典型地具有通用的计算机体系结构。即，微处理器 100

对从芯片集 102 按照内部时钟依次供给来的指令码进行解读并执行。芯片集 102 与所连接的各种组件交换内部数据，并生成微处理器 100 所需的指令码。进而，芯片集 102 还具有对由微处理器 100 进行运算处理的结果所得到的数据等进行缓存的功能。

[0121] 在下面说明的第一实施方式至第三实施方式中，对于微处理器 100，未明确说明是多核的情况下，是单核(处理器)。多核表示核的个数为两个以上的情况(包括核的个数为两个的情况)。

[0122] CPU 单元 13 具有作为存储单元的主存储器 104 及非易失性存储器 106。

[0123] 主存储器 104 是易失性存储区域(RAM : 随机存取存储器)，向 CPU 单元 13 供电之后保持微处理器 100 应执行的各种程序。另外，主存储器 104 还作为由微处理器 100 执行各种程序时的工作用存储器来使用。作为这样的主存储器 104，使用 DRAM (Dynamic Random Access Memory : 动态随机存取存储器) 及 SRAM (Static Random Access Memory : 静态随机存取存储器) 这样的装置。

[0124] 另一方面，非易失性存储器 106 非易失性地保持实时 OS (Operating System : 操作系统)、PLC1 的系统程序、用户程序、动态运算程序、系统设定参数这样的数据。这些程序及数据根据需要被复制到主存储器 104 中，以使微处理器 100 能够访问这些程序及数据。作为这样的非易失性存储器 106，能够使用如闪存器那样的半导体存储器。或者，能够使用如硬盘驱动器那样的磁记录介质及如 DVD-RAM (Digital Versatile Disk Random Access Memory : 数字通用盘随机存取存储器) 那样的光记录介质等。

[0125] 系统计时器 108 以规定周期生成中断信号来提供给微处理器 100。典型地，根据硬件的产品规格来以不同的多个周期分别生成中断信号，但也能够设定为由 OS (Operating System : 操作系统) 或 BIOS (Basic Input Output System : 基本输入输出系统) 等以任意周期生成中断信号。利用该系统计时器 108 所生成的中断信号，来实现如后所述的对每个控制循环的控制动作。

[0126] CPU 单元 13 具有作为通信电路的 PLC 系统总线控制器 120 及现场网络控制器 140。这些通信电路进行输出数据的发送及输入数据的接收。

[0127] 此外，在 CPU 单元 13 本身具备 I/O 单元 14 及 / 或伺服电机驱动器 3 的功能的情况下，将承担 I/O 单元 14 及伺服电机驱动器 3 的功能的部分作为通信对象来在 CPU 单元 13 内部进行发送及接收，以此代替通过通信电路进行的输出数据的发送及输入数据的接收。

[0128] PLC 系统总线控制器 120 对经由 PLC 系统总线 11 的数据交换进行控制。更加具体地，PLC 系统总线控制器 120 包括 DMA (Dynamic Memory Access : 动态内存访问) 控制电路 122、PLC 系统总线控制电路 124 及缓冲存储器 126。此外，PLC 系统总线控制器 120 经由 PLC 系统总线连接器 130 与 PLC 系统总线 11 以内部方式连接。

[0129] 缓冲存储器 126 发挥经由 PLC 系统总线 11 向其他单元输出的数据(下面还称为“输出数据”)的发送缓存以及经由 PLC 系统总线 11 从其他单元输入的数据(下面还称为“输入数据”)的接收缓存的功能。此外，由微处理器 100 通过运算处理生成的输出数据，初步保存到主存储器 104 中。然后，从主存储器 104 读取应向特定单元传送的输出数据，将其暂时保持到缓冲存储器 126 中。另外，从其他单元传来的输入数据暂时保持到缓冲存储器 126 中后，再移送给主存储器 104。

[0130] DMA 控制电路 122 从主存储器 104 向缓冲存储器 126 传送输出数据，以及从缓冲存

储器 126 向主存储器 104 传送输入数据。

[0131] PLC 系统总线控制电路 124, 在与连接在 PLC 系统总线 11 上的其他单元之间, 进行发送缓冲存储器 126 的输出数据的处理以及接收输入数据并将其保存至缓冲存储器 126 的处理。典型地, PLC 系统总线控制电路 124 提供 PLC 系统总线 11 上的物理层及数据链路层的功能。

[0132] 现场网络控制器 140 对经由现场网络 2 的数据交换进行控制。即, 现场网络控制器 140 按照所使用的现场网络 2 的标准, 对输出数据的发送及输入数据的接收进行控制。例如, 在采用了“EtherCAT”(注册商标) 标准的现场网络 2 的情况下, 使用包含用于进行通常的以太网(注册商标) 通信的硬件的现场网络控制器 140。在“EtherCAT”(注册商标) 标准中, 能够使用按照通常的以太网(注册商标) 标准实现通信协议的一般的以太网(注册商标) 控制器。根据作为现场网络 2 采用的工业以太网(注册商标) 的类型, 使用与通常的通信协议不同的、与专用产品规格的通信协议相对应的特别产品规格的以太网(注册商标) 控制器。另外, 在采用了工业以太网(注册商标) 以外的现场网络的情况下, 使用与该标准相对应的专用的现场网络控制器。

[0133] 缓冲存储器 146 发挥经由现场网络 2 向其他装置等输出的数据(对于该数据, 下面也称为“输出数据”) 的发送缓存以及经由现场网络 2 从其他装置等输入的数据(对于该数据, 下面也称为“输入数据”) 的接收缓存的功能。如上所述, 由微处理器 100 通过运算处理生成的输出数据, 初步保存到主存储器 104 中。然后, 从主存储器 104 读取应向特定装置传送的输出数据, 并将其暂时保持到缓冲存储器 146 中。另外, 从其他装置传来的输入数据暂时保持到缓冲存储器 146 中后, 再被转移至主存储器 104。

[0134] DMA 控制电路 142 从主存储器 104 向缓冲存储器 146 传送输出数据, 以及从缓冲存储器 146 向主存储器 104 传送输入数据。

[0135] 现场网络控制电路 144, 在与连接在现场网络 2 上的其他装置之间, 进行发送缓冲存储器 146 的输出数据的处理以及接收输入数据并将其保存至缓冲存储器 146 的处理。典型地, 现场网络控制电路 144 提供现场网络 2 上的物理层及数据链路层的功能。

[0136] USB 连接器 110 是用于连接 PLC 辅助装置 8 和 CPU 单元 13 的接口。典型地, 从 PLC 辅助装置 8 传来的、能够在 CPU 单元 13 的微处理器 100 上执行的程序等, 经由 USB 连接器 110 导入到 PLC1 中。

[0137] < C. CPU 单元的软件结构 >

[0138] 接着, 参照图 3, 说明用于提供本实施方式的各种功能的软件组。包含在这些软件中的指令码, 在适当的时刻被读取并被 CPU 单元 13 的微处理器 100 执行。

[0139] 图 3 是示出了由本发明的实施方式的 CPU 单元 13 执行的软件结构的示意图。参照图 3, 作为由 CPU 单元 13 执行的软件, 有实时 OS200、系统程序 210、用户程序 236 这三个层次。另外, 在控制动作开始之前, 生成控制动作用线程及独立过程 222, 并在进行控制动作时执行这些控制动作用线程及独立过程 222。为了生成控制动作用线程及独立过程 222, 使用排程构建数据 234。在后面详细叙述该排程构建数据 234。

[0140] 实时 OS200 是根据 CPU 单元 13 的计算机体系结构来设计的, 提供用于由微处理器 100 执行系统程序 210 及用户程序 236 的基本的执行环境。典型地, 该实时 OS200 由 PLC 厂商或专业的软件公司等提供。

[0141] 系统程序 210 是用于提供 PLC1 的功能的软件组。具体地，系统程序 210 包含过程库 214、调度程序 212、序列指令运算程序 232 及动态运算程序 234。

[0142] 用户程序 236 是根据用户的控制目的来生成的。即，是根据用 PLC 系统 SYS 控制的目标线(objet line) (工序) 等而任意设计的程序。

[0143] 如后面叙述那样，用户程序 236 与序列指令运算程序 232 及动态运算程序 234 协同实现用户的控制目的。即，用户程序 236 使用由序列指令运算程序 232 及动态运算程序 234 提供的指令、函数、功能模块等，来实现程序化的动作。因此，有时将用户程序 236、序列指令运算程序 232 及动态运算程序 234 统称为控制程序 230。

[0144] 这样，CPU 单元 13 的微处理器 100 执行保存在存储单元中的系统程序 210 及用户程序 236。

[0145] 下面，详细说明各程序。

[0146] 如上所述，用户程序 236 是根据用户的控制目的(例如，目标线(objet line)或工序)来生成的。典型地，用户程序 236 成为能够由 CPU 单元 13 的微处理器 100 执行的目标程序形式。该用户程序是，在 PLC 辅助装置 8 等中对用梯式语言(Ladder language)等记述的源程序编译而生成的。然后，将生成的目标程序形式的用户程序从 PLC 辅助装置 8 经由连接电缆 10 传送至 CPU 单元 13，并将其保存至非易失性存储器 106 等中。

[0147] 过程库 214 是收录了各种过程的库，这些过程是控制动作用线程或独立过程的结构要素(组件)。过程是指，为了进行在程序中反复出现的处理，而将一系列的指令汇总为一个程序。作为实现例，收录在库中的状态的过程是对象及类，成为控制动作用线程的独立过程的结构要素的状态的过程是对象及实例。

[0148] 调度程序 212 按照包含在排程构建数据 224 中的指定(信息)，生成控制动作用线程及独立过程 222。

[0149] 排程构建数据 224 是从 PLC 辅助装置 8 输入的。排程构建数据 224 也可以在生产 CPU 单元 13 时保存到非易失性存储器 106 中。排程构建数据 224 包含用于指定执行收录在过程库 214 中的过程的执行顺序的信息。

[0150] 在控制动作用线程和独立过程 222 中，控制动作用线程包括用于对控制程序的执行进行控制的过程以及用于对输出数据的输出及输入数据的输入进行控制的过程，来作为结构要素。在控制动作用线程和独立过程 222 中，独立过程是不包含在线程中(不从属于线程)执行的过程。

[0151] 序列指令运算程序 232 是指，在执行由用户程序使用的某类序列指令时被调出并为了实现该指令的内容而被执行的程序。

[0152] 动态运算程序 234 是指，按照用户程序的指示来被执行，并且在每次执行时计算用于向伺服电机驱动器 3 或脉冲电机驱动器这样的电机驱动器输出的指令值的程序。

[0153] 实时 OS200 提供用于按照时间的经过切换执行多个程序的环境。

[0154] 过程库 214、调度程序 212、排程构建数据 224 及控制程序 230 保存在作为存储单元的主存储器 104 及非易失性存储器 106 中。

[0155] < D. 主存储器结构>

[0156] 接着，参照图 4，说明 CPU 单元 13 的主存储器 104 中的存储区域。

[0157] 图 4 是示出了本发明的实施方式的 CPU 单元 13 的主存储器 104 的区域结构的示

意图。参照图 4,在主存储器 104 中,形成有各种程序区域 1041、控制程序工作区域 1042、PLC 系统总线发送缓存 1043、PLC 系统总线接收缓存 1044、现场网络发送缓存 1045 及现场网络接收缓存 1046。

[0158] 在控制程序的工作区域 1042 中,通过系统程序 210 生成有各控制程序的本地变量区域,除此之外通过系统程序 210 还生成有全局变量区域 1042a 及变量同步缓冲区域 1042b。在控制程序的工作区域 1042 中,进而,通过系统程序 210 生成有如下区域:输出同步缓冲区域 1042c,其生成与各输出数据相对应的输出同步缓存;输入同步缓冲区域 1042d,其生成有与各输入数据相对应的输入同步缓存。

[0159] 另外,PLC 系统总线发送缓存 1043 及现场网络发送缓存 1045(下面,将这两者统称为“发送缓存”)、PLC 系统总线接收缓存 1044 及现场网络接收缓存 1046(下面,将这两者统称为“接收缓存”),也是通过系统程序 210 生成的。

[0160] PLC 系统总线控制器 120 的 DMA 控制电路 122,将保存在 PLC 系统总线发送缓存 1043 中的输出数据传送至 PLC 系统总线控制器 120 的缓冲存储器 126,并且将保存在缓冲存储器 126 中的输入数据传送至 PLC 系统总线接收缓存 1044。将传送的该输入数据保存至 PLC 系统总线接收缓存 1044。

[0161] 现场网络控制器 140 的 DMA 控制电路 142,将保存在现场网络发送缓存 1045 中的输出数据传送至现场网络控制器 140 的缓冲存储器 146,并且将保存在缓冲存储器 146 中的输入数据传送至现场网络接收缓存 1046。将传送的该输入数据保存至现场网络接收缓存 1046。

[0162] 控制程序的工作区域 1042、PLC 系统总线发送缓存 1043、PLC 系统总线接收缓存 1044、现场网络发送缓存 1045 及现场网络接收缓存 1046,能够相互独立地控制各自的访问。因此,例如能够并行执行下面示出的(1)~(3)这样的多个动作。

[0163] (1)由微处理器 100 进行的,与执行用户程序相伴地对控制程序的工作区域 1042 的访问;

[0164] (2)由 PLC 系统总线控制器 120 的 DMA 控制电路 122 进行的,为了在主存储器 104 中的 PLC 系统总线发送缓存 1043 或 PLC 系统总线接收缓存 1044 或这两者与 PLC 系统总线控制器 120 内的缓冲存储器 126 之间进行数据传送,而对主存储器 104 中的 PLC 系统总线发送缓存 1043 或 PLC 系统总线接收缓存 1044 或这两者的访问;

[0165] (3)由现场网络控制器 140 的 DMA 控制电路 142 进行的,为了在主存储器 104 中的现场网络发送缓存 1045 或现场网络接收缓存 1046 或这两者与现场网络控制器 140 内的缓冲存储器 146 之间进行数据传送,而对主存储器 104 中的现场网络发送缓存 1045 或现场网络接收缓存 1046 或这两者的访问。

[0166] < E. 辅助装置>

[0167] 接着,说明本实施方式的 PLC 辅助装置 8。PLC 辅助装置 8 辅助(辅助)PLC1 的 CPU 单元 13 的使用,具体来说,提供用于生成在 PLC1 中执行的程序以及对 PLC1 进行维护等的功能。

[0168] 图 5 是示出了连接到本发明的实施方式的 CPU 单元上使用的 PLC 辅助装置 8 的硬件结构的示意图。参照图 5,PLC 辅助装置 8 典型地由通用的计算机构成。此外,从维护性的角度出发,优选便携性优秀的笔记本类型的个人计算机。

[0169] 参照图 5,PLC 辅助装置 8 包括 :CPU81, 其执行包括 OS 的各种程序 ;ROM(Read Only Memory :只读存储器)82, 其保存 BIOS 及各种数据 ; 存储器 RAM83, 其提供用于保存由 CPU81 执行程序所需的数据的工作区域 ; 硬盘(HDD)84, 其非易失性地保存由 CPU81 执行的程序等。

[0170] PLC 辅助装置 8 还包括 : 键盘 85 及鼠标 86, 其用于接受来自用户的操作 ; 显示器 87, 其用于将信息提示给用户。进而,PLC 辅助装置 8 包括用于与 PLC1 (CPU 单元 13) 等进行通信的通信接口(IF)89。

[0171] 如后面叙述那样, 由 PLC 辅助装置 8 执行的各种程序保存到 CD-ROM9 中流通。保存在该 CD-ROM9 中的程序被 CD-ROM (Compact Disk-Read Only Memory :光盘只读存储器) 驱动器 88 读取并被保存至硬盘(HDD)84 等。或者, 也可以从上位的主机等经由网络下载程序。

[0172] 如上所述, PLC 辅助装置 8 用通用的计算机来实现, 因而不再进行详细说明。

[0173] 图 6 是示出了连接到本发明的实施方式的 CPU 单元上使用的 PLC 辅助装置 8 的软件结构的示意图。参照图 6, 在 PLC 辅助装置 8 中执行 OS310, 由此提供能够执行 PLC 辅助程序 320 所含的各种程序的环境。

[0174] PLC 辅助程序 320 包括编辑程序 321、编译程序 322、调试程序 323、排程构建数据模板 324、排程构建数据生成程序 325 及通信程序 326。包含在 PLC 辅助程序 320 中的各个程序, 典型地在保存在 CD-ROM9 中的状态流通, 而被安装到 PLC 辅助装置 8 上。

[0175] 编辑程序 321 提供用于生成用户程序的输入及编辑这样的功能。更加具体地, 编辑程序 321 除了提供由用户通过操作键盘 85 及鼠标 86 来生成用户程序的源程序 330 的功能之外, 还提供对所生成的源程序 330 进行保存的功能以及进行编辑的功能。另外, 编辑程序 321 输入来自外部的控制程序(其中, 特别是用户程序)的源程序, 并通过用户的操作来编辑现有的控制程序的源程序。

[0176] 编译程序 322 提供通过对控制程序的源程序进行编译来生成能够在 CPU 单元 13 的微处理器 100 中执行的目标程序形式的用户程序的功能。

[0177] 调试程序 323 提供用于对控制程序的源程序进行调试的功能。作为该调试的内容包括如下动作, 即, 在源程序中局部地执行用户所指定的范围、在执行源程序过程中追踪变量值的随时间的变化。

[0178] 排程构建数据模板 324 是收集如下的排程构建数据而成的, 该排程构建数据用于实现大多用户要使用的一些排程模式。还可以追加用户自己生成的排程构建数据作为模板。

[0179] 排程构建数据生成程序 325 通过以下处理中的某个处理, 来提供用于生成排程构建数据 234 的功能, 这些处理是指, 接受用户所决定的直接使用已经登录为排程构建数据模板 324 的某个排程构建数据的处理, 使用户对那样的登录的排程构建数据的一部分进行变更的处理, 使用户新输入排程构建数据的处理。在此, 直接使用登录为模板的排程构建数据的情况, 也包含在用于生成排程构建数据 234 的处理的概念中。

[0180] 通信程序 326 提供向 PLC1 的 CPU 单元 13 传送控制程序的 CPU 单元 13 用的目标程序(用户程序 236) 及排程构建数据 234 的功能。

[0181] PLC 辅助装置 8 也可以仅采用如下结构, 即, 不包含排程构建数据生成程序 325, 而

只保存一种排程构建数据并将该排程构建数据传送给 CPU 单元。

[0182] 即,PLC 辅助装置 8 包含存储部(图 5 的硬盘(HDD)84 等)和运算部(图 5 的 CPU81)。存储部用于保存 PLC 辅助程序 320 及排程构建数据 234。PLC 辅助程序 320 使运算部执行输出处理,该输出处理输出排程构建 数据,用于使 PLC1 的 CPU 单元 13 能够获取该排程构建数据。

[0183] 为了输出排程构建数据从而使 PLC1 能够获取该排程构建数据,可以经由图 1 所示的连接电缆 10 输出,除此之外,还可已经由未图示的网络直接输出,或者经由连接在未图示的网络上的服务器计算机来间接输出,而且还可以输出到 PLC1 能够读取的便携式记录介质中。

[0184] 在用于使 PLC1 的 CPU 单元 13 获取排程构建数据 234 的排程构建数据获取单元中,包括经由连接电缆 10 进行通信的通信电路、进行网络通信的通信电路以及用于从记录介质读取的读取电路,它们用于获取通过这些途径中的某个途径输出的排程构建数据 234。

[0185] 此外,PLC1 的 CPU 单元 13 也可以采用如下结构,即,在厂商的生产阶段将排程构建数据 234 保存到存储单元中,而从厂商出货后则不能变更排程构建数据 234。

[0186] 一般而言,安装在 PLC1 中的系统程序 210 在 CPU 单元 13 的生产阶段被保存到 CPU 单元 13 的非易失性存储器 106 中。但是,若将系统程序 210 保存到 CD-ROM9 中,则用户也能够将 CD-ROM9 的系统程序 210 复制到 PLC 辅助装置 8 中,并利用通信程序 326 所提供的功能来将所复制的系统程序 210 传送至 CPU 单元 13。进而,若将由 PLC1 的 CPU 单元 13 执行的实时 OS200 保存到 CD-ROM9 中,则实时 OS200 也能够通过用户操作来再次安装到 PLC1 中。

[0187] 下面,说明第一实施方式至第三实施方式,作为使用排程构建数据 234 对 PLC1 的控制动作进行排程的典型例。

[0188] < F. 第一实施方式>

[0189] 在下面说明的第一实施方式中,利用具有任务和线程这样的层次结构的排程构建机构。任务包含一个或多个线程,是重复执行的单位。

[0190] (f1 :整体处理)

[0191] 图 7 是示出了本发明的第一实施方式的 CPU 单元 13 的系统程序 210 的整体处理的流程图。如图 7 所示,首先,执行对 PLC1 的控制动作的执行准备处理(步骤 S1),接着执行对 PLC1 的控制动作的执行控制处理(步骤 S2)。

[0192] 在控制动作的执行准备处理(步骤 S1)中,包括生成全局变量区域 1042a 及变量同步缓存的处理、生成接收缓存(PLC 系统总线接收缓存 1044 或现场网络接收缓存 1046 或这两者)及输入同步缓存的处理、生成发送缓存(PLC 系统总线发送缓存 1043 或现场网络发送缓存 1045 或这两者)及输出同步缓存的处理、生成用于进行控制动作的线程及独立过程 222 的处理、生成用于保存各任务的执行循环数及执行次数和执行时间的各存储区域的处理、生成各任务的结束标识的处理。适当设计这些各处理的执行顺序。

[0193] (f2 :任务及线程)

[0194] 如上所述,在第一实施方式中,说明利用具有任务和线程这样的层次结构的排程构建的机构的处理例。在此,详细说明任务及线程。

[0195] 图 8 是以表形式示出了本发明的第一实施方式中准备的任务的设定项目和对各

设定项目的说明的图。图 9A 及图 9B 以表形式示出了本发明的第一实施方式中准备的线程的设定项目和对各设定项目的说明的图。在本实施方式的 PLC 辅助程序 320 中,能够对各任务设定记载在图 8 的“设定项目”栏中的内容。另外,能够对各线程设定记载在图 9A 及图 9B 的“设定项目”栏中的内容。

[0196] 在图 8、图 9A 及图 9B 中,对线程或过程使用的“起动”这样的术语,表示把成为对象的线程或过程追加到实时 OS200 的执行等待队列(queue)中。在登录到执行等待队列中的线程或过程中,实时 OS200 使微处理器 100 从执行优先级高的线程或过程开始依次执行。将处于登录到执行等待队列中等待执行的状态的线程或过程称为处于“Ready 状态”(预备状态)。将处于由微处理器 100 正在执行的状态的线程或过程称为处于“Run 状态”(运行状态)。将处于未登录到执行等待队列中的状态的线程或过程称为处于“Sleep 状态”(睡眠状态)。

[0197] 图 10 是对于本发明的第一实施方式中准备的过程之一的控制循环开始过程示出了其功能、设定项目及对设定项目的说明的图。图 11A 及图 11B 是对于本发明的第一实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。

[0198] 在图 10、图 11A 及图 11B 所示的过程中,控制循环开始过程(图 10)、任务开始过程(图 11A)及任务结束过程(图 11A)的实例,作为独立过程生成。

[0199] 下面,对本实施方式的排程例 1 ~ 10,与相对应的排程构建数据的例子等一起进行说明。

[0200] (f3 :排程例 1 :在执行循环周期内执行结束的情况)

[0201] 在本发明的第一实施方式的排程例 1 中,执行控制 A 任务和控制 B 任务。控制 A 任务包括输入输出 A 线程和控制 A 线程,控制 B 线程包括控制 B 线程。控制 A 任务的执行循环相当于一次控制循环,控制 B 任务的执行循环相当于两次控制循环。在第 k 循环中,开始控制 A 任务及控制 B 任务的一个执行循环。

[0202] 图 12 是示出了与本发明的第一实施方式的排程例 1 相对应的排程构建数据 234 的图。图 13 是本发明的第一实施方式的排程例 1 的第 k 循环的时序图。图 14 是本发明的第一实施方式的排程例 1 的第 k+1 循环的时序图。

[0203] 在图 12 中以表形式示意性示出了排程构建数据 234 的内容,但实际使用的排程构建数据 234 是以微处理器 100 能够处理的形式记述的,例如利用 XML (Extensible Markup Language :可扩充标记语言) 记述等。

[0204] 排程例 1 的排程构建数据 234 包括如图 12 的(1)部分示出的控制循环开始过程的设定内容、如图 12 的(2)部分和(5)部分示出的任务的设定内容、如图 12 的(3)部分和(4)部分和(6)部分示出的线程的设定内容。

[0205] 如图 12 的(1)部分所示,在第一实施方式的排程例 1 中,在控制循环开始过程中设定有作为控制循环开始条件的“控制循环开始中断”。从而,如在图 10 中说明的那样,控制循环开始过程在控制循环开始中断(在图 13 中,开始第 k 循环的控制循环开始中断)出现时被起动。然后,由于优先级被设定为“1”,因而控制循环开始过程比其他线程及过程优先被起动后立即被执行。另外,控制循环开始过程的功能是指,“起动在各任务的设定项目‘任务开始处理’中指定的过程”。具体而言,在图 12 的(2)部分及(5)部分各自的“任务开始处理”中指定有“任务开始过程”,因而控制循环开始过程起动控制 A 任务的任务开始过程

及控制 B 任务的任务开始过程，并结束自身的执行。

[0206] 若控制循环开始过程执行结束，则控制 A 任务的任务开始过程成为 Run 状态。任务开始过程执行在图 11A 中说明的功能。如在图 12 的(2)部分的“线程指定”中指定那样，在控制 A 任务中包含“输入输出 A 线程”和“控制 A 线程”。如图 12 的(3)部分示出那样，因为输入输出 A 线程的开始条件是“控制 A 任务”，所以输入输出 A 线程成为任务开始过程的后续起动对象。另一方面，因为控制 A 线程的开始条件不是“控制 A 任务”，所以控制 A 线程不是任务开始过程的后续起动对象。任务开始过程起动输入输出 A 线程，并结束自身的执行。

[0207] 若控制 A 任务的任务开始过程的执行结束，则控制 B 任务的任务开始过程成为 Run 状态。任务开始过程，在执行图 11A 中说明的功能之后，结束自身的执行。如在图 12 的(5)部分的“线程指定”中指定那样，在控制 B 任务中包含有“控制 B 线程”。参照图 12 的(6)部分，因为控制 B 线程的开始条件不是“控制 B 任务”，所以控制 B 线程不是任务开始过程的后续起动对象，控制 B 任务的任务开始过程不执行任何处理。

[0208] 若控制 B 任务的任务开始过程的执行结束，则输入输出 A 线程成为 Run 状态。在图 12 的(3)部分的“过程指定”中例举的过程以例举的顺序被执行。

[0209] 若输入输出 A 线程成为 Run 状态，则最初执行同步输出复制过程。就同步输出复制过程的功能而言，如图 11A 中说明的那样，因为控制 A 任务及控制 B 任务各自的当前的执行循环数均为“1”，所以对控制 A 任务及控制 B 任务这双方，将预先指定的输出数据从输出同步缓存复制到发送缓存中。参照图 18，在后面详细叙述同步输出复制过程的功能。

[0210] 接着，执行输入输出指示过程。即，通过对通信电路发送输入输出指示信号，通信电路发送保存在发送缓存中的输出数据，将所接收的输入数据保存到接收缓存中，并将通信结束通知信号发送给微处理器 100。例如，在利用作为现场网络 2 的“EtherCAT”（注册商标）的情况下，从现场网络控制器 140 发送来的帧(frame)，在与从属设备进行数据交换的状态下，遍历从属设备后返回。现场网络控制器 140 自动接收返回来的帧。从而，输入输出指示信号虽然直接意味着发送指示信号，但还间接地指示进行接收。

[0211] 执行输入输出指示过程之后，接着执行通信结束通知等待过程。即，通信结束通知等待过程被设定为在通信结束通知出现时再次起动输入输出 A 线程，并使输入输出 A 线程成为 Sleep 状态。

[0212] 在输入输出 A 线程处于 Sleep 状态期间，执行登录在执行等待队列中的 比从属于控制任务的线程及过程的优先级低的各种系统处理。

[0213] 若通信结束通知出现，则执行同步输入复制过程。在同步输入复制过程的功能中，如图 11B 中说明的那样，因为控制 A 任务及控制 B 任务各自的当前的执行循环计数器均为“1”，所以对控制 A 任务及控制 B 任务这双方，将预先指定的输入数据从接收缓存复制到输入同步缓存中。参照图 17，在后面详细叙述同步输入复制过程的功能。

[0214] 对成为输出复制的对象的输出数据以及成为输入复制的对象的输入数据的预先指定，是由 PLC 辅助装置 8 作为对控制程序 230 的设定信息来指定的。

[0215] 接着，执行后续起动过程。如图 12 的(4)部分所示，因为在控制 A 线程的“开始条件”中指定有输入输出 A 线程，所以后续起动过程起动控制 A 线程。

[0216] 若输入输出 A 线程的执行结束，控制 A 线程成为 Run 状态。在图 12 的(4)部分的

“过程指定”中例举的过程以例举的顺序被执行。

[0217] 若控制 A 线程成为 Run 状态,最初执行数据追踪过程。如图 11B 中说明的那样,数据追踪过程将预先指定的变量值记录到日志文件中。对该变量的预先指定,也是由 PLC 辅助装置 8 作为对控制程序 230 的设定信息来指定的。

[0218] 接着,执行开始参照变量过程。即,进行用于使变量同步缓存不能改写如下变量的处理,该变量是指,(在后述的“控制程序指定”中指定)由控制 A 程序作为参照侧控制程序 (referrer-side control program) 来参照的变量。参照图 19,在后面详细叙述这样的由开始参照变量过程及变量配送过程(控制 B 线程)实现的变量同步功能。

[0219] 接着,执行控制程序执行过程。如图 12 的(4)部分所示,因为在控制 A 线程的“控制程序指定”中指定有“控制 A 程序”,所以执行控制 A 程序。控制 A 程序可以是按照用户的控制目的来生成的序列程序,并且,在那样的序列程序中含有动态指令的情况下,该控制 A 程序也可以是序列程序及在该序列程序之后接着执行的动态运算程序。

[0220] 若控制 A 程序的执行结束,执行变量配送过程。即,对于由控制 A 程序作为主侧控制程序 (owner-side control program) 来进行更新并且由控制 B 程序作为参照侧控制程序来参照的全局变量,由变量配送过程将该变量的内容复制到控制 B 程序的变量同步缓存中。此外,参照图 19,在后面详细叙述主侧控制程序及参照侧控制程序。

[0221] 接着,执行后续起动过程。如图 12 的(2)部分所示,在控制 A 任务的“任务结束条件”中指定有控制 A 线程。另外,如图 12 的(6)部分所示,在控制 B 线程的“开始条件”中指定有控制 A 线程,而且,当前的控制 B 任务的任务结束标识为关(OFF)。从而,后续起动过程起动控制 A 任务的任务结束过程和控制 B 线程。

[0222] 若控制 A 线程的执行结束,控制 A 任务的任务结束过程成为 Run 状态。任务结束过程执行在图 11A 中说明的功能。

[0223] 若控制 A 任务的任务结束过程的执行结束,控制 B 线程成为 Run 状态。在图 12 的(6)部分的“过程指定”中例举的过程以例举的顺序被执行。

[0224] 若控制 B 线程成为 Run 状态,最初执行开始参照变量过程。即,进行用于使变量同步缓存不能改写如下变量的处理,该变量是指,(在后述的“控制程序指定”中被指定)由控制 B 程序作为参照侧控制程序来参照的变量。

[0225] 接着,执行控制程序执行过程。如图 12 的(6)部分所示,因为在控制 B 线程的“控制程序指定”中指定有“控制 B 程序”,所以执行控制 B 程序。控制 B 程序可以是根据用户的控制目的来生成的序列程序,并且,在那样的序列程序包含动态指令的情况下,该控制 B 程序也可以是序列程序及在该序列程序之后接着执行的动态运算程序。

[0226] 在图 13 中示出了在控制 B 程序的执行过程中包括用于开始第 k+1 循环的控制循环开始中断的情况。控制循环开始中断使控制程序 B 执行中断。

[0227] 如图 14 所示,在第 k+1 循环中,与第 k 循环的情况同样地,执行控制 A 任务的处理及控制 B 任务的任务开始过程。

[0228] 在输入输出 A 线程的通信结束通知等待时间内,由于与在第 k 循环中执行的系统处理相比,控制 B 线程一侧的优先级更高,因而再次开始控制程序 B 的执行,若通信结束通知出现,则控制程序 B 再次被执行中断。

[0229] 若控制 A 任务的任务结束过程的执行结束,再次开始控制程序 B 的执行。

[0230] 若控制 B 程序的执行结束, 执行变量配送过程。即, 对于由控制 B 程序作为主侧控制程序来进行更新并且由控制 A 程序作为参照侧控制程序来参 照的全局变量, 将该变量的内容复制到控制 A 程序的变量同步缓存中。

[0231] 接着, 执行后续起动过程。如图 12 的(5)部分所示, 由于在控制 B 任务的“任务结束条件”中指定有“控制 B 线程”, 因而后续起动过程起动控制 B 任务的任务结束过程。

[0232] 若控制 B 线程的执行结束, 控制 B 任务的任务结束过程成为 Run 状态。任务结束过程执行图 11A 中说明的功能。

[0233] 若控制 B 任务的任务结束过程的执行结束, 在接收到下一个控制循环开始中断为止的期间内, 执行各种系统处理。

[0234] 如在上面说明的那样, 本发明的第一实施方式的 PLC1 的 CPU 单元 13 的系统程序 210 具有如下特征。即, 系统程序 210 包括过程库(过程库 214), 该过程库(过程库 214)至少收录有用于对控制程序的执行进行控制的过程和用于对输出数据的输出及输入数据的输入进行控制的过程。然后, 系统程序 210, 在包含特定信息的排程构建数据 234 被保存在存储单元(主存储器 104 或非易失性存储器 106)中时, 使微处理器 100 按照该特定信息来执行收录在过程库中的过程, 该特定信息是指, 对收录在过程库中的过程的执行顺序进行指定的信息。此外, 后述的第二实施方式及第三实施方式也共通地具有该特征。

[0235] 在此, 用于对控制程序 230 的执行进行控制的过程的例子, 是开始参照变量过程、控制程序执行过程及变量配送过程。用于对输出数据的输出及输入数据的输入进行控制的过程的例子, 是同步输出复制过程、输入输出指示过程、通信结束通知等待过程及同步输入复制过程。

[0236] 具体地, 上述特征在第一实施方式中按如下方式实现的。即, 排程构建数据 234 包括用于使用收录在过程库(过程库 214)中的过程来生成线程的指定(图 12 的(3)部分、(4)部分及(6)部分)(信息)。系统程序 210 以下处理来作为对 PLC 的控制动作的执行准备处理, 在该处理中, 按照包含在排程构建数据 234 中的用于生成线程的指定(信息)来生成线程(图 7 的步骤 S1)。由此, 能够利用实时 OS200 的执行等待队列和执行优先级的机构, 进行对 PLC1 的控制动作的执行控制。此外, 后述的第二实施方式也共通地具有该特征。

[0237] 另外, 在第一实施方式中, 排程构建数据 234 包含用于在执行线程的过 程中起动其他线程的指定(在图 12 等中示出的对线程的“开始条件”的指定)(信息), 来作为与线程相关的数据。由此, 能够以高自由度来进行对 PLC1 的控制动作的执行控制, 而这种高自由度是仅使用优先级机构(机制)无法实现的。此外, 在后述的第二实施方式中也共通地具有该特征。

[0238] 另外, 在第一实施方式中, 排程构建数据 234 包含用于对成为重复(反复)执行的单位的任务进行设定的指定(信息), 该指定(信息)包含一个或多个线程。系统程序 210 包含按照用于对包含在排程构建数据 234 指定任务进行设定的指定(信息)来设定任务的处理(典型地, 图 7 的步骤 S1 中的生成独立过程的处理), 以作为对 PLC1 的控制动作的执行准备处理。由此, 反复执行的单位变得更加易懂。另外, 若将执行循环的周期设定到任务中而不是设定到各线程中, 则对执行循环的设定变得更加简单。另外, 对反复执行的单位(任务)进行执行次数的计数或执行时间的计测的处理也变得容易。

[0239] 另外, CPU 单元 13 在如下情况下能够执行控制动作, 该情况是指, 在存储单元(主

存储器 104 或非易失性存储器 106) 中保存有包含对如下过程的执行顺序的指定(信息)的排程构建数据 234, 其中, 该过程的执行顺序, 用于反复进行输出数据的发送、输入数据的接收以及使用输入数据来生成输出数据的控制程序的执行。

[0240] (f4 :排程例 1 :在执行循环周期内未执行结束的情况)

[0241] 接着, 说明任务在对该任务设定的执行循环周期内未执行结束的情况, 即执行循环超过(超限)时的动作。

[0242] 图 15 及图 16 是示出了在本发明的第一实施方式的排程例 1 中控制 A 线程的执行在第 k 循环内未结束的情况的动作的时序图。

[0243] 如上述的图 12 的(2)部分及(5)部分所示, 由于对控制 A 任务及控制 B 任务的“循环超过容许次数”分别指定有“1”, 因而对控制 A 任务及控制 B 任务容许延长一次控制循环的执行循环。

[0244] 如图 15 所示, 由于在用于开始第 k+1 循环的控制循环开始中断发生之前, 控制 A 程序尚未执行结束, 因而该控制 A 程序被实时 OS200 执行中断。

[0245] 如图 16 所示, 在第 k+1 循环中, 控制循环开始过程与第 k 循环同样地执行。在第 k+1 循环开始时, 在第 k 循环中执行开始的控制 A 任务的执行未结束, 但由于属于图 11A 所记载的任务开始过程的“递增计数后的执行循环数超过了执行循环周期(n)并且任务结束标识为闭(OFF)”条件, 因而控制循环开始过程起动输入输出 A 线程。这样, 就输入输出 A 线程而言, 除了执行循环超过的错误处理的情况及前一次控制循环的输入输出 A 线程未结束的情况之外, 在所有的控制循环中都被执行。在第 k+1 循环中的输入输出 A 线程的通信结束通知等待时间内, 控制 A 程序从执行中断的位置开始执行。进而, 在输入输出 A 线程的执行结束后, 控制 A 程序再次执行开始。就输入输出 A 线程的后续起动过程而言, 由于本来是起动控制 A 线程但控制 A 线程已被起动, 因而不能重复起动控制 A 线程。

[0246] 在第 k+1 循环中, 若控制 A 任务的任务结束过程的执行结束, 控制 B 线程成为 Run 状态。控制 B 任务的执行循环从第 k 循环开始, 但在第 k 循环中没有执行控制 B 线程的时间, 因而控制 B 线程从第 k+1 循环开始执行。

[0247] 就控制 B 任务的执行循环而言, 本来在第 k+1 循环内结束, 但在第 k+1 循环内控制 B 线程未执行结束, 因而控制 B 任务也发生执行循环超过。

[0248] 如在图 11A 中说明的那样, 若在执行循环周期中任务在加上了循环超过容许次数的控制循环次数内执行未结束, 则执行执行循环超过的错误处理(出错处理)。例如, 中止控制动作, 并将发生执行循环超过的情况(的信息)经由未图示的网络报知给上位计算机。

[0249] 在第一实施方式的 PLC1 的 CPU 单元 13 中, 设有用于设定控制循环周期的单元。作为设定控制循环周期的单元, 具有为了设定控制循环周期而使用的以下的要素, 即, 为了获取用于指定控制循环周期的信息而使用的与 PLC 辅助装置 8 进行通信的通信单元、包含在系统程序 210 中的用于设定控制循环周期的程序、能够对控制循环开始中断信号的周期任意进行设定的系统计时器的结构。也可以将用于接受用于指定控制循环周期的输入的操作开关等输入装置安装到 CPU 单元 13 本身上, 以代替从 PLC 辅助装置 8 获取用于指定控制循环周期的信息。

[0250] 第一实施方式的 PLC1 的 CPU 单元 13 的通信电路(PLC 系统总线控制器 120 或现场网络控制器 140 或这两者), 在每个控制循环中进行输出数据的发送及输入数据的接收。

更加具体地，通信电路接受通过输入输出指示过程的执行来生成的输入输出指示信号，由此进行发送及接收。代替上述处理，也可以使通信电路通过被由系统计时器生成的规定周期的信号触发来进行每个控制循环中的发送及接收。另外，也可以通过在通信电路本身设置计时器电路，来按照该计时器电路生成的规定周期的信号来进行每个控制循环中的发送及接收。

[0251] 如在上面说明的那样，第一实施方式的 PLC1 的 CPU 单元 13 具有用于设定控制循环周期的单元。通信电路在每个控制循环中进行输出数据的发送及输入数据的接收。而且，排程构建数据 234 包含对如下过程的执行顺序的指定(信息)，该过程是指，对于与控制循环周期相等的执行循环的周期内执行的控制程序(控制 A 程序)，在控制程序的执行结束的控制循环的下一个控制循环内，使微处理器 100 在通信电路进行输出数据的发送及输入数据的接收之后，开始执行控制程序，而在控制程序的执行未结束的控制循环的下一个控制循环内，使微处理器 100 执行控制程序中的未执行的部分。根据这样的 CPU 单元 13，PLC1 的用户能够在控制程序的执行时间零星地变长的状况下，将由 PLC1 的 CPU 单元 13 为了进行控制数据的输出及输入而与控制循环同步进行的通信的周期，设定为比控制程序的最大执行时间更短的规定时间。

[0252] 这种状况的一个例子是控制程序 230 包含动态运算程序的情况，该动态运算程序在每次被执行时都要计算用于输出给电机驱动器的指令值。动态运算程序存在如下情况，即，该动态运算程序在被通过按照用户的控制目的来生成的用户程序 236 (序列程序) 起动而初次执行时，因执行开始计算指令值所需的初始处理而需要比后续的执行所需的时间更长的执行时间。在这样的状况下，也能够将由 CPU 单元 13 为了执行控制数据的输出及输入而与控制循环同步进行的通信的周期，设定为比控制程序 230 的最大执行时间更短的规定时间。

[0253] (f5 : 同步输入复制过程)

[0254] 接着，详细说明同步输入复制过程的功能。图 17 是用于说明本发明的第一实施方式中准备的同步输入复制过程的动作的时序图。

[0255] 在图 17 所示的排程例中，示出了执行了执行循环为一个控制循环的“输入输出 C 线程”、执行循环为两个控制循环的“控制 C 线程”以及执行循环为四个控制循环的“控制 D 线程”这样的合计三个线程的情况。

[0256] 输入输出 C 线程的结构与图 13 及图 14 所示的输入输出 A 线程相同。控制 C 线程及控制 D 线程的结构分别与图 13 及图 14 所示的控制 A 线程及控制 B 线程相同，但执行循环的周期与控制 A 线程及控制 B 线程不同。另外，在控制 C 线程中执行控制 C 程序，在控制 D 线程中执行控制 D 程序。图 17 的各线程分别属于不同的任务中。

[0257] 图 17 的接收缓存是图 4 的 PLC 系统总线接收缓存 1044 及现场网络接收缓存 1046。图 17 的输入同步缓存 C 是针对由在控制 C 线程中执行的控制 C 程序进行参照的每个输入数据来设定的。同样地，图 17 的输入同步缓存 D 是针对由在控制 D 线程中执行的控制 D 程序进行参照的每个输入数据来设定的。输入同步缓存 C 及输入同步缓存 D 设在图 4 的输入同步缓冲区域 1042d。此外，也可以将由控制程序使用的变量区域按照原样直接作为输入同步缓存来利用。

[0258] 参照图 17，在每次执行输入输出 C 线程时，将 PLC 系统总线控制器 120 或现场网

络控制器 140 或这两者所接收的新的输入数据覆写(覆盖)保存到接收缓存中。同步输入复制过程将保存在接收缓存中的输入数据,复制到使用该输入数据的控制程序的输入同步缓存(输入同步缓存 C 及输入同步缓存 D)中。此时,以适于控制程序使用的形式再次配置输入数据。

[0259] 如在图 11B 中说明的那样,同步输入复制过程仅限于任务的执行循环数为“1”的情况,即,仅限于当前的控制循环在任务的执行循环中处于最初的控制循环的情况,具有将对该任务预先指定的输入数据从接收缓存复制到输入同步缓存中的功能。从而,如图 17 所示,在控制循环 1 ~ 控制循环 4 的范围内,从接收缓存向输入同步缓存 C 的复制仅在控制循环 1 及控制循环 3 中进行,从接收缓存向输入同步缓存 D 的复制仅在控制循环 1 中进行。这是由于控制 C 线程的执行循环设定为两个控制循环,控制 D 线程的执行循环为四个控制循环。

[0260] 于是,保存在输入同步缓存中的输入数据,在执行参照该输入数据的控制程序的期间内,不会被新的输入数据覆盖,因而即使控制程序参照几次输入数据都能够得到相同的值,从而能够进行具有一致性的运算。进而,由于仅在控制线程(控制程序)的执行循环中的最初的控制循环中进行输入复制,因而能够使得使用控制程序的输入数据的更新周期恒定。

[0261] 在图 17 中,针对每个控制程序以及针对每个输入数据设定一个输入同步缓存,但也可以针对每个控制程序以及针对每个输入数据设定两个或三个输入同步缓存,由此设计一边更替各输入同步缓存的作用(状态)一边运用的过程。例如,设计为使一个缓存处于由控制程序参照的状态(参照状态)而使另一个缓存处于能够写入新的输入数据的状态(写入状态)。然后,在开始执行控制程序之前,若在能够写入的缓存中保存有新的输入数据,则能够更替两个缓存的作用。这样,由于不会存在不能进行输入复制的期间,因而能够防止输入数据消失。其中,在图 17 的排程例中,各线程的执行循环的周期为整数倍的关系,因而控制程序应使用的输入数据的原本就不会消失。也可以取代直接更替输入同步缓存的参照状态和写入状态,采取如下方法,即,通过利用三个缓存,来使保存了新的输入数据的缓存从写入状态经由第三状态(待机状态)转移至参照状态。

[0262] 此外,如在图 11B 中说明的那样,还准备有如下的输入复制过程,该输入复制过程将预先指定的输入数据从接收缓存复制到输入同步缓存中的功能,但不对各任务的执行循环数进行判断(过程本身不具有与其他任务的执行循环采取同步的功能)。在后述的排程例 3 中说明这样的输入复制过程的使用例。

[0263] 对于利用接收缓存及输入同步缓存的 CPU 单元 13 或利用输入复制过程或同步输入复制过程的输入复制处理的 CPU 单元 13 的特征,能够总结如下。

[0264] 系统程序 210 包括:作为对 PLC1 的控制动作的执行准备处理,在存储单元(主存储器 104 等)中生成用于保存所接收的输入数据的接收缓存的处理(图 7 的步骤 S1);针对由控制程序 230 参照的每个输入数据,在存储单元(主存储器 104 等)中生成由该控制程序 230 作为输入数据的参照对象的输入同步缓存的处理(图 7 的步骤 S1)。

[0265] 排程构建数据 234 包括对如下过程的执行顺序的指定,该过程使微处理器 100 执行以下处理,即,将所接收的输入数据从接收缓存复制到与该输入数据相对应的输入同步缓存中的输入复制处理、使控制程序 230 执行开始的控制程序开始处理。于是,保存在输入

同步缓存中的输入数据，在执行参照该输入数据的控制程序的期间内，不会被新的输入数据覆盖，因而控制程序 230 即使参照几次输入数据都能够得到相同的值，从而能够进行具有一致性的运算。

[0266] 进而，对利用同步输入复制过程的同步输入复制处理的 CPU 单元 13 的特征，能够总结如下。

[0267] 在每个控制循环中执行输入数据的接收。排程构建数据 234 包括对如下过程的执行顺序的指定，该过程使微处理器 100 执行以下处理，即，在控制循环的整数倍的每个执行循环中执行控制程序开始处理，并且在控制程序的执行循环开始的控制循环中将所接收的输入数据作为对象，执行输入复制处理。于是，能够使控制程序 230 所使用的输入数据的更新周期恒定。

[0268] (f6：同步输出复制过程)

[0269] 接着，详细说明同步输出复制过程的功能。图 18 是用于说明本发明的第一实施方式中准备的同步输出复制过程的动作的时序图。

[0270] 在图 18 所示的排程例中，示出了执行输入输出 C 线程、控制 C 线程及控制 D 线程这合计三个线程的情况。这些线程的排程例与图 17 的排程例相同。

[0271] 图 18 的发送缓存是图 4 的 PLC 系统总线发送缓存 1043 及现场网络发送缓存 1045。图 18 的输出同步缓存 C 针对通过在控制 C 线程中执行的控制 C 程序的执行来生成的每个输出数据而设定的。同样地，输出同步缓存 D 针对通过在控制 D 线程中执行的控制 D 程序的执行来生成的每个输出数据而设定的。输出同步缓存 C 及输出同步缓存 D 设在图 4 的输出同步缓冲区域 1042c。此外，也可以将由控制程序使用的变量区域按照原样直接作为输出同步缓存来使用。

[0272] 参照图 18，在每次执行包含在输入输出 C 线程中的输入输出指示过程时，PLC 系统总线控制器 120 或现场网络控制器 140 或这两者，发送保存在发送缓存中的输出数据。同步输出复制过程在执行输入输出指示过程之前被执行，将保存在输出同步缓存（输出同步缓存 C 及输出同步缓存 D）中的输出数据复制至发送缓存。此时，在 PLC 系统总线发送缓存 1043 中，以针对每个发送对象的单元集中配置的形式，再次配置输出数据，另外，在现场网络发送缓存 1045 中，能够以串行帧发送的形式，再次配置输出数据。

[0273] 如在图 11A 中说明的那样，同步输出复制过程仅限于任务的执行循环数为“1”的情况，即，仅限于当前的控制循环在任务的执行循环中处于最初 的控制循环的情况，具有将对该任务预先指定的输出数据从输出同步缓存复制到发送缓存中的功能。从而，如图 18 所述，在控制循环 1 ~ 控制循环 4 的范围内，从输出同步缓存 C 向发送缓存的复制仅在控制循环 1 及控制循环 3 中进行，从输出同步缓存 D 向发送缓存的复制仅在控制循环 1 中进行。这是由于控制 C 线程的执行循环设定为两个控制循环，控制 D 线程的执行循环设定为四个控制循环。

[0274] 于是，保存在发送缓存中的输出数据，在刚要在各控制程序的执行循环中的最初的控制循环中发送之前，从输出同步缓存复制至该发送缓存。该输出数据是在之前的执行循环中生成的，在输出数据被复制到发送缓存的时间点，之前的执行循环中的控制程序的执行已结束。从而，不会发送在控制程序的执行过程中的还有可能被更新的输出数据。进而，由于仅在控制线程（控制程序）的执行循环中的最初的控制循环中进行输出复制，因而

能够使所发送的输出数据的更新周期恒定。

[0275] 此外,如在图 11A 中说明的那样,还准备有如下的输出复制过程,该输出复制过程具有将预先指定的输出数据从输出同步缓存复制到发送缓存中的功能,但不对各任务的执行循环数进行判断(过程本身不具有与其他任务的执行循环采取同步的功能)。在后述的排程例 3 中说明这样的输出复制过程的使用例。

[0276] 对于使用输出同步缓存及发送缓存的 CPU 单元 13 或使用输出复制过程或同步输出复制过程的输出复制处理的 CPU 单元 13 的特征,能够总结如下。

[0277] 作为对PLC1 的控制动作的执行准备处理,系统程序 210 包括 :在存储单元(主存储器 104 等)中生成用于保存应发送的输出数据的发送缓存的处理(图 7 的步骤 S1);对于保存有复制在发送缓存中的输出数据的输出同步缓存,针对每个输出数据,在存储单元(主存储器 104 等)中生成该输出同步缓存的处理(图 7 的步骤 S1)。

[0278] 排程构建数据 234 包括对如下过程的执行顺序的指定,该过程使微处理器 100 执行将输出数据从输出同步缓存复制到发送缓存中的输出复制处理。这样一来,在控制程序的执行过程中还有可能被覆盖更新的输出数据不会被发送。

[0279] 进而,对于使用同步输出复制过程的同步输出复制处理的 CPU 单元 13 的特征,能够总结如下。

[0280] 排程构建数据 234 包括对如下过程的执行顺序的指定,该过程使微处理器 100 执行以下处理,即,在控制循环的整数倍的每个执行循环中执行控制程序开始处理,并且在控制程序 230 的执行循环的最初的控制循环中,在由之前的执行循环中的控制程序 230 的执行而生成的输出数据从发送缓存被发送的时刻,执行输出复制处理。于是,能够使所发送的输出数据的更新周期恒定。

[0281] (f7 :变量配送过程及开始参照变量过程)

[0282] 接着,详细说明变量配送过程及开始参照变量过程的功能(变量同步功能)。图 19 是用于说明本发明的第一实施方式中准备的变量配送过程及开始参照变量过程的动作的时序图。

[0283] 图 19 所示的排程例,与上述的图 13 及图 14 所示的排程例 1 相同,仅示出了与变量同步功能关联的控制 A 线程及控制 B 线程的部分。

[0284] 图 19 的全局变量区域 A 是用于保存如下变量的主存储器的区域,该变量即由控制 A 线程的控制程序 A 来使用,也由其他线程的控制程序(具体地,控制 B 线程的控制 B 程序)来参照。全局变量区域 A 设在图 4 的全局变量区域 1042a。此外,也可以将由控制程序 230 使用的变量区域按照原样直接作为全局变量区域来使用。

[0285] 将容许对全局变量区域 A 进行改写的控制程序(在图 19 的例子中是控制程序 A)称为“主侧控制程序”,将仅容许参照的控制程序(在图 19 的例子中是控制程序 B)称为“参照侧控制程序”。对于一个全局变量,也可以有一个主侧控制程序但有多个参照侧控制程序。

[0286] 针对由控制 B 线程的控制 B 程序作为参照侧控制程序来参照的每个全局变量,在图 4 的变量同步缓冲区域 1042b 设有变量同步缓存 B。此外,也可以将由控制程序 230 使用的变量区域按照原样直接作为变量同步缓存来利用。

[0287] 哪个控制程序 230 是主侧控制程序或参照侧控制程序,是针对每个全局变量作为变量的属性来指定的。包含该指定的变量的属性数据,是在 PLC 辅助装置 8 中通过用户的

输入操作来生成的,将所生成的变量的属性数据从 PLC 辅助装置 8 发送至 CPU 单元 13,并将其保存至 CPU 单元 13 的非易失性存储器 106。

[0288] 虽未图示,但还存在控制 B 程序是主侧控制程序而控制 A 程序是参照侧控制程序的全局变量,针对该全局变量设有与控制 B 程序相对应关联的全局变量区域 B 和与控制 A 程序相对应关联的变量同步缓存 A。

[0289] 变量配送过程,将在自身所属的线程中执行的控制程序 230 是主侧的全局变量的值,从全局变量区域,复制到与作为该全局变量的参照侧的控制程序相对应关联的变量同步缓存中。在图 19 中示出了由控制 A 线程的变量配送过程将全局变量区域 A 的内容(数据)复制到变量同步缓存 B 中的情况。

[0290] 开始参照变量过程进行用于使由在自身所属的线程中执行的控制程序参照的变量同步缓存的内容不被更新的处理。在图 19 中,控制 B 线程的开始参照变量过程使变量同步缓存 B 处于写入禁止状态。从而,在控制循环 1 中,控制 A 线程的变量配送过程能够向变量同步缓存 B 进行复制,但在控制 B 线程的开始参照变量过程被执行之后即在控制循环 2 中,控制 A 线程的变量配送过程不能向变量同步缓存 B 进行复制。可以使任务结束过程具有用于解除该变量同步缓存的写入禁止状态的功能,也可以准备具有解除写入禁止状态的变量参照结束过程。

[0291] 于是,即使在从控制 B 程序的执行开始起到执行结束为止的期间由控制程序 A 对全局变量区域 A 进行了改写,也由于由控制程序 B 参照的变量同步缓存 B 的内容不会被改写,因而即使控制程序 B 参照几次该全局变量都能够得到相同的值,从而能够进行具有一致性的运算。

[0292] 在图 19 中,针对每个控制程序以及针对由该控制程序作为参照侧来参照的每个全局变量设定了变量同步缓存,但也可以针对每个控制程序以及针对由该控制程序作为参照侧来参照的每个全局变量设定两个或三个变量同步缓存,由此设计能够一边更替各变量同步缓存的作用(状态)一边运用的过程。例如,设计为使一个缓存处于由参照侧控制程序参照的状态(参照状态)而使另一个缓存处于能够写入新数据的状态(写入状态)。然后,在开始执行参照侧控制程序之前,若在能够写入的缓存中保存有新的输入数据,则更替两个缓存的作用。这样,由于不会存在不能进行变量配送的期间,因而能够防止由主侧控制程序生成的最新全局变量值的配送遗漏。在图 19 的情况下,对于在控制循环 2 及控制循环 4 中由控制 A 程序生成的全局变量的值,由于在控制 B 程序的各执行开始时间点不是最新的,因而即使不配送也没问题,但在利用多核微处理器来以不同的核来并行执行控制 A 程序和控制 B 程序的情况下,若变量同步缓存是一个则有可能发生配送遗漏。这样的情况下,通过如上述那样用两个变量同步缓存一边更替一组变量同步缓存的参照状态和写入状态一边运用,能够防止配送遗漏。也可以取代直接更替变量同步缓存的参照状态和写入状态,采取如下方法,即,通过使用三个缓存,来使保存有新配送来的全局变量的值的缓存从写入状态经由第三状态(待机状态)转移至参照状态。

[0293] 对于使用全局变量区域及变量同步缓存的 CPU 单元 13 或使用变量配送过程的变量配送处理的 CPU 单元 13 的特征,能够总结如下。

[0294] 存储单元(非易失性存储器 106 等)用于保存对由控制程序 230 使用的变量的属性数据。就属性数据而言,在变量是由多个控制程序 230 参照的全局变量的情况下,能够包含

对能够对该变量进行改写的一个主侧控制程序的指定以及对仅能够参照该变量的一个或多个参照侧控制程序的指定(信息)。

[0295] 作为对PLC1的控制动作的执行准备处理,系统程序210包括:在存储单元(主存储器104等)中生成用于保存全局变量的全局变量区域1042a的处理(图7的步骤S1);对于各参照侧控制程序,针对由该参照侧控制程序参照的每个全局变量,在由该参照侧控制程序参照全局变量时,取代全局变量区域而将作为参照对象的变量同步缓存生成到存储单元(主存储器104等)中的处理(图7的步骤S1)。

[0296] 排程构建数据234包括对如下过程的执行顺序的指定,该过程使微处理器100执行以下处理,即,使主侧控制程序执行开始的主侧开始处理;若主侧控制程序的执行结束,则将由该主侧控制程序改写的全局变量从全局变量区域复制到与该全局变量相对应的变量同步缓存中的复制处理;使参照侧控制程序执行开始的参照侧开始处理。

[0297] 这样,即使在从参照侧控制程序执行开始起到执行结束为止的期间由主侧控制程序对全局变量区域进行了改写,也由于由参照侧控制程序参照的变量同步缓存的内容不会被改写,因而即使参照侧控制程序参照几次该全局变量都能够得到相同的值,从而能够进行具有一致性的运算。

[0298] (f8 :排程例 2)

[0299] 接着,说明本发明的第一实施方式的排程例2。

[0300] 图20是示出了与本发明的第一实施方式的排程例2相对应的排程构建数据234的图。图21是本发明的第一实施方式的排程例2的第k循环的时序图。对于根据图20的排程构建数据234来生成图21的序列的处理,若考虑排程例1则会不言自明,因而不详细说明图20的排程构建数据234和图21的序列之间的关系。

[0301] 就排程例2而言,与上述的排程例1相比,不同点在于,输入输出线程(排程例1的输入输出A线程)分为输出线程(输出E线程)和输入线程(输入E线程)。另外,与排程例1的控制B线程的情况相比,不同点在于对控制F线程的开始条件指定有控制F任务,因而利用控制F任务的任务开始过程来起动控制F线程。在控制F线程不使用输入数据的情况下,通过不这样等待输入E线程的结束,而在输入E线程的通信结束通知的等待时间使控制F线程的执行开始,能够提早结束控制F线程。

[0302] 在排程例2中,说明分别独立地设定输出E线程和输入E线程的情况。考虑作为现场网络使用“EtherCAT”(注册商标)的情况,例如考虑如下情况,即,在第k循环中发送来的帧因噪声的影响等而在网络传送过程中丢失,导致在网络中巡回一圈的帧未到达CPU单元13的情况。在排程例1及排程例2中,不生成通信结束通知,导致产生控制A任务或控制E任务在第k循环中未结束这样的循环超过的现象。

[0303] 在这样的情况下,如排程例1那样,若输入输出线程是一个线程,则在第k+1循环中再次等待通信结束通知,但若在网络中帧本身丢失,则发生最终超过循环超过容许次数的错误。对此,在排程例2中,第k+1循环属于在图11A中关于任务开始过程进行说明的“在递增计数后执行循环数超过执行循环周期(n)而且任务结束标识为闭(OFF)”这样的条件。因此,第k+1循环的任务开始过程起动设定为后续起动对象的输出E线程。从而,发送新的帧,在该帧在网络中巡回一圈后回来时,能够继续进行如图21所示的输入E线程的通信结束通知以后的处理。

[0304] 这样,在采用了帧在网络中巡回一圈的类型的现场网络的情况下,如本排程例那样,通过分开输出线程和输入线程,即使在网络中帧丢失的情况下,也会使继续进行规定周期的网络通信的处理变得容易。

[0305] (f9 :排程例 3)

[0306] 接着,说明本发明的第一实施方式的排程例 3。

[0307] 图 22 是示出了与本发明的第一实施方式的排程例 3 相对应的排程构建数据 234 的图。图 23 是本发明的第一实施方式的排程例 3 的第 k 循环的时序图。图 24 是本发明的第一实施方式的排程例 3 的第 k+1 循环的时序图。对于根据图 22 的排程构建数据 234 来生成图 23 及图 24 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 22 的排程构建数据 234 和图 23 及图 24 的序列之间的关系。

[0308] 就排程例 3 而言,与上述的排程例 1 相比,不同点在于,执行循环的周期较长,因而该周期中包含的作为控制线程的优先级较低的任务的控制任务 H (相当于排程例 1 的控制任务 B) 中也包含有输入输出线程(输入输出 H 线程)。

[0309] 输入输出 G 线程例如通过现场网络来进行控制 G 线程的控制 A 程序所需的所有的输入输出和控制 H 线程的控制程序 B 所需的一部分的输入输出。输入输出 H 线程例如通过 PLC 系统总线 11 来进行控制 H 线程的控制 B 程序所需的剩余的输入输出。

[0310] 在输入输出 G 线程中,利用同步输出复制过程及同步输入复制过程,相对于此,在输入输出 H 线程中,仅进行从属于相同任务的用于进行控制 H 线程的输入输出,因而利用对各任务的执行循环数不进行判断的(过程本身不具有与他的任务的执行循环采取同步的功能)输出复制过程及输入复制过程。

[0311] 这样,也可以在执行循环的周期较长的任务中进行输入输出处理。另外,也可以设定仅进行输入输出处理的任务。以与每个输入数据及输出数据相对应地以执行控制程序 230 所需的周期进行输入输出处理的方式,进行排程构建,由此能够避免白白地以高的频率进行大量的数据量的通信而浪费通信时间的情况。

[0312] (f10 :排程例 4)

[0313] 接着,说明本发明的第一实施方式的排程例 4。

[0314] 图 25 是示出了与本发明的第一实施方式的排程例 4 相对应的排程构建数据 234 的图。图 26 是本发明的第一实施方式的排程例 4 的第 k 循环的时序图。图 27 是本发明的第一实施方式的排程例 4 的第 k+1 循环的时序图。对于根据图 25 的排程构建数据 234 来生成图 26 及图 27 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 25 的排程构建数据 234 和图 26 及图 27 的序列之间的关系。

[0315] 就排程例 4 而言,与上述的排程例 1 相比,不同点在于,在控制 J 线程中执行的控制程序是动态运算程序而不包含序列程序,而且,仅在控制 K 线程中执行根据用户的控制目的来生成的序列程序。其中,与排程例 1 相比,不同点还有:序列程序仅在控制 K 线程中执行,伴随与此,数据追踪过程在控制 K 线程中执行,另外,由于不需要在多个序列程序之间进行全局变量的同步,因而不使用变量配送过程及开始参照变量过程。

[0316] 在控制 J 线程中执行的控制程序仅有动态运算程序,是由于在图 25 的(4)部分的“控制程序指定”中指定为“动态运算程序”。在图 25 的(6)部分的“控制程序指定”中指定有在控制 K 线程中执行的控制程序为“序列 K 程序”,在“动态线程指定”中指定在“控制 J

线程”中执行包含在序列 K 程序中的指定动态指令。

[0317] 例如,若在序列 K 程序中执行“请从坐标 X1 至 X2 为止以速度 V 移动”这样的动态指令,通过与此相对应地执行其后的最初的控制循环中的控制 J 线程的动态运算程序,动态运算程序执行用于对此后向伺服电机驱动器发送的位置及速度的指令值进行反复计算的初始运算以及第一次的指令值计算。此后,到由动态指令指示的动作结束为止,即使没有来自序列 K 程序的干预,也可以通过控制 J 线程的动态运算程序的执行来针对每个控制循环计算指令值。计算并输出指令值的周期较短的一方能够使电机的动作的误差变小。若假定包含在执行循环为一个控制循环的任务中的控制程序仅有动态运算程序,则容易将控制循环周期设定得短。从该角度,优选排程例 4。

[0318] (f11 :排程例 5)

[0319] 接着,说明本发明的第一实施方式的排程例 5。

[0320] 图 28 是示出了与本发明的第一实施方式的排程例 5 相对应的排程构建数据 234 的图。图 29 是本发明的第一实施方式的排程例 5 的第 k 循环的时序图。对于根据图 28 的排程构建数据 234 来生成图 29 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 28 的排程构建数据 234 和图 29 的序列之间的关系。

[0321] 就排程例 5 而言,与上述的排程例 1 相比,不同点在于,在执行循环周期为一个控制循环周期的控制 L 任务中,执行输入输出 L 线程、控制 L1 线程及控制 L2 线程。在控制 L1 线程中执行的控制程序是序列 L 程序,在控制 L1 线程中不执行动态运算程序。另一方面,在控制 L2 线程中执行的控制程序仅有按照包含在序列 L 程序中的动态指令来执行的动态运算程序。

[0322] 排程例 5 的特征在于,输入输出处理、包含动态指令的序列程序(用户程序)以及动态运算程序,以该顺序被反复执行。就动态运算程序而言,在按照动态指令开始反复执行时,在到动态指令所指示的动作结束为止的期间,能够在序列程序不干预的状态下被执行。然而,存在如下情况,即,在到动态指令所指示的动作结束为止的期间,因使用了新的输入数据的序列程序的运算结果而导致向动态运算程序发送的指示内容被变更。在这样的情况下,若采用上述特征的排程,则通过输入数据的输入,经由利用了该输入数据的序列运算而将对动作的变更指示发送至动态运算程序,能够缩短到输出反映了变更指示的动态指令值为止的期间的时间。另外,通过输入使动态指令的执行条件成立的输入数据,还能够缩短到按照该动态指令输出最初的动力指令值为止的期间的时间。

[0323] 在排程例 1 中,在控制程序 A 包含序列程序和在该序列程序之后接着执行的动态运算程序的情况下,也能够得到同样的效果。

[0324] 对于采用如上述那样的处理顺序的 CPU 单元 13 的特征,能够总结如下。

[0325] 控制程序 230 包括:动态运算程序 234,其计算用于对电机的动作进行控制的动态指令值;用户程序 236,其是根据用户的控制目的而生成的,包括对动态运算程序 234 发送用于执行该动态运算程序 234 所需的指示的处理。

[0326] 排程构建数据 234 包含对如下过程的行顺序的指定,该过程使 PLC1 的 CPU 单元 13 依次反复执行如下处理,即,通过通信电路(PLC 系统总线控制器 120 或现场网络控制器 140 或这两者)进行的输出数据的发送及输入数据的接收、用户程序 236 的执行、动态运算程序 234 的执行。但是,与输出数据的发送和输入数据的接收之间的执行顺序无关。

[0327] (f12 :排程例 6)

[0328] 接着,说明本发明的第一实施方式的排程例 6。在排程例 6 中,示出了执行输入输出 P 线程、控制 P 线程及控制 Q 线程这合计三个线程的情况。

[0329] 图 30 是与本发明的第一实施方式的排程例 6 相对应的时序图。参照图 30,输入输出 P 线程及控制 P 线程包含在执行循环周期为一个控制循环周期的任务中,控制 Q 线程包含在执行循环周期为两个控制循环周期的任务中。而且,在控制 P 线程中执行的第一控制程序 230-1 和在控制 Q 线程中执行的第二控制程序 230-2,均包含动态运算程序 234。另外,第一控制程序 230-1 及第二控制程序 230-2 中的至少一个程序包括了含有动态指令的用户程序 236。

[0330] 在图 30 中,从输入输出 P 线程向控制 P 线程及控制 Q 线程分别延伸的箭头,表示输入数据的流向。具体地,相当于由同步输入复制过程执行的输入数据的复制。存在如下情况,即,在第一控制程序 230-1 或第二控制程序 230-2 或这两者包含用户程序 236 的情况下,在该输入数据中,除了包含由该用户程序 236 使用的输入数据之外,还包含由动态运算程序 234 使用的输入数据(此外,还存在动态运算程序 234 不使用输入数据而执行运算的情况)。

[0331] 同样地,在图 30 中,从控制 P 线程及控制 Q 线程向输入输出 P 线程分别延伸的箭头,表示输出数据的流向。具体地,相当于由同步输出复制过程执行的输出数据的复制。在第一控制程序 230-1 或第二控制程序 230-2 或这两者包含用户程序 236 的情况下,该输出数据除了包含由该用户程序 236 生成的输出数据之外,在动态运算程序 234 处于激活状态的(即,在每次执行时计算指令值)期间内,还包含由该动态运算程序 234 生成的动态指令值数据。

[0332] 图 31 是用于说明在本发明的第一实施方式的排程例 6 中的控制程序中的程序结构和指示的流程的示意图。图 31 中示出了能够包含在第一控制程序 230-1 及第二控制程序 230-2 中的用户程序 236 的变化(variation)的例子。即,图 31 的(1)部分示出了仅有第一控制程序 230-1 包含用户程序 236 的例子,图 31 的(2)部分示出了仅有第二控制程序 230-2 包含用户程序 236 的例子,图 31 的(3)部分示出了第一控制程序 230-1 及第二控制程序 230-2 这双方包含用户程序 236 的例子。

[0333] 在图 31 中,从用户程序 236(在图 31 的(3)部分中是第一用户程序 236-1 或第二用户程序 236-2)向动态运算程序 234(第一动态运算程序 234-1 或第二动态运算程序 234-2)延伸的箭头,表示执行动态运算程序 234 所需的指示的流动。该指示例如在用户程序 236 中记述为用于向定义动态控制内容的动态功能块(动态指令)输入的输入参数(输入定数或输入变量)。而且,在随着用户程序 236 的执行而动态控制开始的条件成立从而执行动态功能块时,将该输入参数赋予动态运算程序 234。

[0334] 就在控制程序包含用户程序 236 及动态运算程序 234 时执行这两个程序的执行顺序而言,先执行哪个程序都可以。在先执行用户程序 236 的情况下,优选能够将用户程序 236 的执行结果立即反映至动态运算程序 234 的执行中。另一方面,在先执行动态运算程序 234 的情况下,将用户程序 236 的执行结果在如下的执行循环中反映至动态运算程序 234 的执行中。

[0335] 参照图 31 的(1)部分,说明仅在第一控制程序 230-1 中包含用户程序 236 的方

式。在能够在所希望的控制循环时间内执行向第一动态运算程序 234-1 及第二动态运算程序 234-2 这双方发送指示的用户程序 236 和第一动态运算程序 234-1 的情况下,能够采用图 31 的(1)部分所示的例子。

[0336] 即,在图 31 的(1)部分所示的例子中,第一控制程序 230-1 包含如下的用户程序 236,该用户程序 236 包含用于向第一动态运算程序 234-1 发送执行该程序所需的指示的指令以及用于向第二动态运算程序 234-2 发送执行该程序所需的指示的指令。

[0337] 根据图 31 的(1)部分所示的例子,在第一动态运算程序 234-1 的任意执行循环中,都能够从用户程序 236 接收新的指示。从而,能够将用于变更执行过程中的电机动作的指示等迅速地反映至动态控制中。此外,在第二动态运算程序 234-2 的执行循环的途中发送到了第二动态运算程序 234-2 的指示,在第二动态运算程序 234-2 的下一个执行循环中被反映。进而,根据图 31 的(1)部分所示的例子,能够将用户程序 236 概括为一个,因而其设计变得容易。

[0338] 参照图 31 的(2)部分,说明仅在第二控制程序 230-2 中包含用户程序 236 的方式。在从所希望的控制循环时间中减去第一动态运算程序 234-1 的执行时间后得到的能够对用户程序 236 的执行分配的时间较少的情况下,图 31 的(2)部分所示的例子是最佳的方式。

[0339] 即,在图 31 的(2)部分所示的例子中,第二控制程序 230-2 包含如下的用户程序 236,该用户程序 236 包含用于向第一动态运算程序 234-1 发送执行该程序所需的指示的指令以及用于向第二动态运算程序 234-2 发送执行该程序所需的指示的指令。

[0340] 根据图 31 的(2)部分所示的例子,能够将控制循环时间缩短至足以执行第一动态运算程序 234-1 的时间。但是,从用户程序 236 向第一动态运算程序 234-1 发送的指示,能够仅在第二控制程序 230-2 的执行循环的周期内发送。然而,在动态运算程序 234 从用户程序 236 例如接收到“请从坐标 X1 向坐标 X2 为止以速度 V 移动”这样的指示而处于激活状态时,在到该指示的执行结束为止的期间内,能够不接收来自用户程序 236 的指示,而是针对每个执行循环计算动态指令值数据。从而,在这样的指示的执行过程中,在不需发送用于使电机的动作中止的指示或向其他动作变更的指示而使其迅速反映的情况下,即使采用图 31 的(2)部分所示的方式也没问题。进而,根据图 31 的(2)部分所示的例子,能够将用户程序 236 概括为一个,因而其设计变得容易。

[0341] 参照图 31 的(3)部分,说明在第一控制程序 230-1 中包含第一用户程序 236-1 而且在第二控制程序 230-2 中包含第二用户程序 236-2 的方式。

[0342] 即,在图 31 的(3)部分所示的例子中,用户程序 236 由第一用户程序 236-1 和第二用户程序 236-2 构成,其中,该第一用户程序 236-1 包含用于向第一动态运算程序 234-1 发送执行该程序所需的指示的指令,该第二用户程序 236-2 包含用于向第二动态运算程序 234-2 发送执行该程序所需的指示的指令。

[0343] 根据图 31 的(3)部分所示的例子,包含在第一控制程序 230-1 中的第一用户程序 236-1,不需进行与相第二动态运算程序 234-2 发送指示的处理相关的处理,因而能够相对缩短执行时间。从而,能够相对缩短作为能够执行第一用户程序 236-1 和第一动态运算程序 234-1 的时间的控制循环时间。进而,在第一动态运算程序 234-1 及第二动态运算程序 234-2 中的任意程序的执行循环中,都能够从第一用户程序 236-1 及第二用户程序 236-2 分

别接收新的指示。

[0344] 如上述的排程例 6 那样,对于采用在多个控制程序中包含动态运算程序 234 的排程的 CPU 单元 13 的特征,能够总结如下。

[0345] 通信电路(PLC 系统总线控制器 120 或现场网络控制器 140 或这两者)在控制循环周期内发送输出数据并接收输入数据。控制程序 230 包含第一控制程序 230-1 及第二控制程序 230-2。

[0346] 第一控制程序 230-1 包含第一动态运算程序 234-1,该第一动态运算程序 234-1 生成用于对电机的动作进行控制的第一动态指令值数据。第二控制程序 230-2 包含第二动态运算程序 234-1,该第二动态运算程序 234-1 生成用于对电机的动作进行控制的第二动态指令值数据。而且,第一控制程序 230-1 及第二控制程序 230-2 中的至少一个程序包含如下的用户程序 236(第一用户程序 236-1 或第二用户程序 236-2),该用户程序 236 是根据用户的控制目的来生成的,包含用于向第一动态运算程序 234-1 及第二动态运算程序 234-1 发送执行这些程序所需的指示的指令。

[0347] 排程构建数据 234 使微处理器 100 针对与控制循环周期相同的周期的每个第一执行循环开始执行第一控制程序 230-1,并且针对控制循环周期的二倍以上的整数倍的周期的每个第二执行循环开始执行第二控制程序 230-2。进而,排程构建数据 234 包含对如下的过程的执行顺序的指定,该过程,在第二执行循环开始的控制循环中在第一控制程序 230-1 的执行结束之后开始执行第二控制程序 230-2,并且在到该控制循环结束为止第二控制程序 230-2 未结束时,在下一个控制循环中在第一控制程序 230-1 的执行结束之后开始执行第二控制程序 230-2 的未执行的部分。

[0348] 于是,在具有动态控制功能的 PLC1 中,能够为了需要高速处理的一部分动态控制处理而确保较短的周期的执行循环,并且还能够以规定周期执行这以外的动态控制处理。

[0349] 例如,在控制对对象物进行切削加工的加工设备的情况下,能够以短的周期执行如与刀具动作的控制相关的高速且高精度的控制所需的动态控制处理,并且还能够以规定周期执行如向加工设备搬入对象物或从加工设备搬出对象物这样的相对地不需高速性的动态控制处理。

[0350] (f13 :排程例 7)

[0351] 接着,说明本发明的第一实施方式的排程例 7。

[0352] 图 32 是示出了与本发明的第一实施方式的排程例 7 相对应的排程构建数据 234 的图。图 33 是本发明的第一实施方式的排程例 7 的第 k 循环的时序图。图 34 是本发明的第一实施方式的排程例 7 的第 k+1 循环的时序图。对于根据图 32 的排程构建数据 234 来生成图 33 及图 34 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 32 的排程构建数据 234 和图 33 及图 34 的序列之间的关系。

[0353] 适用排程例 7 的 CPU 单元 13 的微处理器 100 具有包含第一核及第二核的多个核。而且,以第一核执行控制循环开始过程及控制 R 任务,以第二核执行控制 S 任务。

[0354] 关于控制 R 任务的执行,除了以下点以外,与排程例 1 的控制 A 任务相同,该点是指,在控制 R 任务的任务开始过程的执行之后,不等待控制 S 任务的任务开始过程的执行结束而输入输出 R 线程成为 Run 状态。

[0355] 就控制 S 线程而言,与排程例 1 的控制 B 线程不同地,被输入输出 R 线程的后续起

动过程起动时立即成为 Run 状态,与控制 R 线程并行地被执行。另外,控制 S 线程不会被使第 k+1 循环开始的控制循环开始中断执行中断,而是继续执行。进而,控制 S 线程还与第 k+1 循环中的输入输出 R 线程并行执行。

[0356] 控制 S 线程在利用第二核的资源的基础上,还能够与第 k 循环的输入输出 R 线程并行执行,但就在第 k 循环中执行开始的控制 S 线程而言,由于使用通过执行第 k 循环中的输入输出 R 线程来输入的输入数据,因而被输入输出 R 线程的后续起动过程起动。

[0357] 对于上面说明的采用在具有多个核的微处理器 100 中执行的排程的 CPU 单元 13 的特征,能够总结如下。

[0358] 微处理器 100 至少包含第一核及第二核。控制程序 230 包含第一控制程序和第二控制程序。排程构建数据 234 包含对如下的过程的执行顺序的指定,该过程,在输出数据的发送及输入数据的接收结束之后,使第一核执行第一控制程序,并且使第二核与第一控制程序的执行并行地执行第二控制程序。

[0359] 这样一来,能够使 PLC1 采用特征性的输入输出处理和控制程序执行之间的周期性(cyclic)处理,还能够通过使具有多个核的微处理器 100 并行执行程序来实现处理的高速化。

[0360] (f14 :排程例 8 及排程例 9)

[0361] 根据本实施方式,能够建立与以往的典型的 PLC1 相同的排程。在下面 说明的排程例 8 及排程例 9 就是这样的例子。

[0362] 图 35 是示出了与本发明的第一实施方式的排程例 8 相对应的排程构建数据 234 的图。图 36 是本发明的第一实施方式的排程例 8 的第 k 循环及第 k+1 循环的时序图。对于根据图 35 的排程构建数据 234 来生成图 36 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 35 的排程构建数据 234 和图 36 的序列之间的关系。

[0363] 在排程例 8 中,在一个线程(控制 T 线程)中,进行控制程序的执行以及输入输出处理。另外,不利用控制循环开始中断,而是在一个控制循环的任务执行结束时立即开始下一个控制循环。这是由于,在图 35 的(1)部分的“控制循环开始条件”中指定有“控制 T 任务”,而不是指定有“控制循环开始中断”。在由系统程序 210 执行的处理中,除了图 36 所示的处理以外的包含与 PLC 辅助装置 8 之间的通信处理等的其他处理,在通信结束通知等待时间内执行。将由该系统程序 210 执行的其他的处理还称为外围处理。

[0364] 作为排程例 8 的变形例,也可以通过将由系统程序 210 执行的上述其他处理(外围处理)嵌入到控制 T 线程中,来例如在执行各控制循环的输入复制过程之后执行上述其他处理。

[0365] 图 37 是示出了与本发明的第一实施方式的排程例 9 相对应的排程构建数据 234 的图。图 38 是本发明的第一实施方式的排程例 9 的第 k 循环及第 k+1 循环的时序图。对于根据图 37 的排程构建数据 234 来生成图 38 的序列的处理,若考虑排程例 1 则会不言自明,因而不详细说明图 35 的排程构建数据 234 和图 36 的序列之间的关系。

[0366] 在排程例 8 中,在执行控制程序之后进行输入输出处理,但在排程例 9 中,依次执行输入处理、控制程序的执行以及输出处理。随之在排程例 9 中,不仅使用输入输出指示过程,还使用输入指示过程和输出指示过程。

[0367] 进而,在排程例 9 中,在进行了输出的通信结束通知之后,执行循环计时过程。如

在图 11B 中说明的那样,循环计时过程具有如下功能,即,设定处理器内置的计时器,从而获取从控制循环开始经过的经过时间,并计算到控制循环时间结束为止的剩余时间,如果经过了所计算出的剩余时间则进入计时中断,设定为在计时中断出现时使本线程再次起动,并具有使本线程中止(使其成为 Sleep 状态)的功能。此外,若在执行循环计时过程的时间点 已经过了控制循环时间,则循环计时过程不设定计时中断,而是随即执行后续起动过程。与利用控制循环开始中断的情况不同地,不会因控制循环时间的经过而强制中断线程的执行。通过使用循环计时过程,能够在不使用控制循环开始中断的状态下,使控制循环周期几乎为恒定时间。在输入及输出的通信结束通知等待时间内以及循环计时过程所设定的计时中断的等待时间内,执行由系统程序 210 执行的上述其他处理(外围处理)。

[0368] <G. 第二实施方式>

[0369] 在上述的第一实施方式中,说明了使用具有由任务和线程这样的层次结构的排程构建的机构的方法。相对于此,在第二实施方式中,说明建立排程时使用线程但不使用任务的方法。

[0370] 在第二实施方式中,在由系统程序 210 执行的对 PLC1 的控制动作的执行准备处理(图 7 的步骤 S1)中,与各线程相对应地,在存储单元(主存储器 104 等)中设定有用于保存线程的执行循环数的存储区域、线程结束标识的存储区域、用于保存线程的执行次数的存储区域以及用于保存线程的执行时间的存储区域。

[0371] 图 39A 及图 39B 时以表形式示出了本发明的第二实施方式中准备的线程的设定项目和对各设定项目的说明的图。在图 8 所示的第一实施方式的任务的设定项目中,对作为任务开始条件的设定项目来表示的周期的设定,在图 39A 及图 39B 所示的第二实施方式中移转到了对线程的开始条件的设定项目。另外,在第一实施方式中作为任务的设定项目的循环超过容许次数,在第二实施方式中也移转到了线程的设定项目。

[0372] 图 40A 及图 40B 是示出了本发明的第二实施方式中准备的过程的名称及功能的图。关于控制循环开始过程,在图 40A 及图 40B 中还示出了设定项目及对设定项目的说明。在第二实施方式中,也准备了图 11A 的后续起动过程以后的各过程,但由于其内容与图 11A 及图 11B 相同而未重复图示。

[0373] 在第二实施方式中,第一实施方式中的任务开始过程的功能实质上移转到了控制循环开始过程中,任务结束过程的功能实质上移转到了线程结束过程中。

[0374] 图 41 是与本发明的第二实施方式的排程例相对应的排程构建数据 234 的图。图 42 是本发明的第二实施方式的排程例的第 k 循环的时序图。图 43 是本发明的第二实施方式的排程例的第 k+1 循环的时序图。对于根据图 41 的排程构建数据 234 来生成图 42 及图 43 的序列的处理,若参照图 39A、图 39B、图 40A 及图 40B 考虑第一实施方式的排程例 1 则会不言自明,因而不详细说明图 41 的排程构建数据 234 和图 42 及图 43 的序列之间的关系。

[0375] 第二实施方式的该排程例利用第二实施方式的机构来实现了与第一实施方式的排程例 1 几乎相同的排程。即,建立排程时利用线程但不用任务。因此,在第一实施方式中将任务作为对象进行的执行次数的计数以及执行时间的计测,在第二实施方式中将线程作为对象来进行。对于第一实施方式的其他排程例,也能够同样地利用第二实施方式的机构来实现几乎相同的排程。

[0376] <H. 第三实施方式>

[0377] 在上述的第一实施方式中,说明了利用具有由任务和线程这样的层次结构的排程构建的机构的方法。相对于此,在第三实施方式中,说明不使用任务及线程,而是通过对过程赋予执行编号来按照执行编号的顺序执行过程由此建立排程的方法。

[0378] 第三实施方式的排程构建数据包含对要执行的过程的指定、对所指定的各过程的执行编号和执行循环周期的指定。在通过对相同名称的过程指定多次来分别指定了不同的执行编号的情况下,针对每个执行编号生成过程的目标及实例。在以后的说明中,被指定的执行编号不同的相同名称的过程是相互区分的不同的过程。

[0379] 在第三实施方式中,在系统程序 210 对 PLC1 的控制动作的执行准备处理(图 7 的步骤 S1)中,与各过程相对应地,在存储单元(主存储器 104 等)中设定用于保存执行循环数的存储区域和已执行标识的存储区域。

[0380] 另外,第三实施方式的过程除了具有各过程固有的功能之外,还与任意过程共通地具有对过程的执行进行控制的功能。

[0381] 图 44 是示出了本发明的第三实施方式的各过程共通具备的执行控制功能所提供的处理的流程图。参照图 44,若过程的执行开始,则判断本过程的已执行标识是否是关(OFF)(步骤 S100)。在第三实施方式中,对于各过程的执行是否结束,利用相对应的已执行标识的状态(关(OFF)或开(ON))来进行判断。

[0382] 若本过程的已执行标识不是关(OFF)(在步骤 S100 中为“否”),则处理进入步骤 S106。

[0383] 若本过程的已执行标识为关(OFF)(在步骤 S100 中为“是”),则执行本过程的固有功能的处理(步骤 S102)。此外,在本过程通过之前的处理而处于执行中断过程中的情况下,执行未执行的部分。而且,若本过程的固有功能的处理的执行结束,则将本过程的已执行标识设定为开(ON)(步骤 S104)。

[0384] 在步骤 S106 中,判断是否存在比本过程的执行编号大的执行编号的其他过程。若比本过程的执行编号大的执行编号的其他过程不存在(在步骤 S106 中为“否”),则本过程结束。

[0385] 另一方面,若比本过程的执行编号大的执行编号的其他过程存在(在步骤 S106 中为“是”),则在本过程后接着起动执行编号大的过程(步骤 S108)。然后,本过程的执行结束。

[0386] 图 45 是对于本发明的第三实施方式中准备的过程之一的控制循环开始过程示出了其功能、设定项目及对设定项目的说明的图。图 46A 及图 46B 是对于本发明的第三实施方式中准备的控制循环开始过程以外的过程示出了各过程的名称及功能的图。

[0387] 图 47 是示出了与本发明的第三实施方式的排程例相对应的排程构建数据 234 的图。在图 47 的(1)部分中,对控制循环开始过程设定有作为控制循环开始条件的“控制循环开始中断”。在图 47 的(2)部分中,如前面的说明那样,指定要执行的过程,并且针对所指定的每个过程指定执行编号及执行循环周期。进而,对于控制程序执行过程,还指定由该过程执行的对象控制程序。在此,示出设定这样的与第一实施方式的排程例 1 类似的排程的例子,例如,设定为以执行循环周期“1”执行控制 A 程序,以执行循环周期“2”执行控制 B 程序。

[0388] 图 48 是与图 47 的排程构建数据相对应的排程例的第 k 循环的时序图。图 49 是与图 47 的排程构建数据相对应的排程例的第 k+1 循环的时序图。

[0389] 参照图 47 及图 48，在第 k 循环中，因控制循环开始中断出现而执行控制循环开始过程。如在图 45 中说明的那样，控制循环开始过程执行对各过程的执行循环数的递增计数以及其他处理。控制循环开始过程，在第 k 循环 中对全部的过程将执行循环数设定为“1”，并且将已执行标识设定为关(OFF)。进而，控制循环开始过程首先起动执行编号最小的过程(在图 47 的例子中是同步输出复制过程)。此后，按照各过程共通具备的执行控制功能的处理步骤(图 44 所示的流程图)，从执行编号小的过程开始依次执行各过程。各过程在执行固有功能的处理之后已执行标识被设定为开(ON)。在图 47 的例子中，在控制 B 程序的执行过程中第 k+1 循环的控制循环开始中断出现，因而使控制 B 程序的执行中断。

[0390] 参照图 47 及图 49，在第 k+1 循环中，因控制循环开始中断出现而执行控制循环开始过程。控制循环开始过程对各过程的执行循环数进行递增计数之后，各过程的执行循环数成为“2”。对于从执行编号为“1”到“8”为止的过程，执行循环周期为“1”，所以在该时间点超过了执行循环数所指定的执行循环周期。因此，控制循环开始过程将这些过程的执行循环数设定为“1”，并且将相对应的已执行标识设定为关(OFF)。另一方面，对于从执行编号为“9”到“11”为止的过程，执行循环周期为“2”，所以在该时间点未超过执行循环数所指定的执行循环周期。因此，控制循环开始过程使这些过程的执行循环数及已执行标识保持现状。此后，从执行编号最小的过程(在图 47 的例子中是同步输出过程)开始执行。

[0391] 执行循环周期为“1”的执行编号为“1”到“8”为止的过程，被与第 k 循环中的情况同样地执行。就执行编号“9”的开始参照变量过程而言，由于在第 k 循环中已执行标识被设定为开(ON)而保持原状，因而按照图 44 的流程，不执行其固有功能，而启动执行编号“10”的控制程序执行过程(参照图 44 的步骤 S100)。

[0392] 就利用执行编号“10”的控制程序执行过程来执行的控制程序 B 而言，在第 k 循环中处于执行中断过程中，因而执行该控制程序 B 的未执行的部分(参照图 44 的步骤 S102)。

[0393] 接着，通过执行编号“11”的变量配送过程的执行结束，第 k+1 循环中的过程的执行结束。

[0394] 接着，到第 k+2 循环的控制循环开始中断出现为止的时间，用来执行系统程序的其他处理(外围处理)。

[0395] 上述排程例的执行内容与上述的第一实施方式的排程例 1 类似。这样，在本发明的第三实施方式中，针对每个过程，指定用于决定执行顺序的行编号和执行循环周期，因而只要关注在各控制循环内执行的过程的执行顺序，就能够构建可得到与使用执行循环周期不同的多个线程的情况相同的结果的排程。

[0396] 在上述的第三实施方式的例子中不包含与执行循环超过相对应的功能，但也可以模仿第一实施方式及第二实施方式，来追加用于容许所设定的控制循环数为止的过程的执行循环超过的功能。

[0397] 也可以取代如上述第三实施方式那样使过程本身具备执行控制功能，来使系统程序 210 承担与图 44 相当的执行控制功能，并使系统程序 210 基于如图 47 所示的排程构建数据来逐个依次调出过程并执行，也能够实现与第三实施方式相同的排程。

[0398] 应当认为本公开的实施方式是在全部点的例示而非限制。本发明的范围并不由上述说明来表示，而是由权利要求书来表示，意在包括在与权利要求书均等的意思和范围内的全部变更。

[0399] 附图标记的说明

[0400] 1PLC, 2 现场网络, 3 伺服电机驱动器, 4 伺服电机, 5 终端, 6 检测开关, 7 继电器, 8PLC 辅助装置, 9CD-ROM, 10 连接电缆, 11PLC 系统总线, 12 电源单元, 13CPU 单元, 14、53I0 单元, 15 特殊单元, 51 终端总线, 52 通信连接器(coupler), 81CPU, 83RAM, 85 键盘, 86 鼠标, 87 显示器, 88CD-ROM 驱动器, 100 微处理器, 102 芯片集, 104 主存储器, 106 非易失性存储器, 108 系统计时器, 110USB 连接器, 120PLC 系统总线控制器, 122DMA 控制电路, 124PLC 系统总线控制电路, 126、146 缓冲存储器, 130 连接器, 140 现场网络控制器, 144 现场网络控制电路, 200 实时 OS, 210 系统程序, 212 调度程序, 214 过程库, 222 控制动作用线程及独立过程, 224、234 排程构建数据, 230 控制程序, 230-1 第一控制程序, 230-2 第二控制程序, 232 序列指令运算程序, 234 动态运算程序, 234-1 第一动态运算程序, 234-2 第二动态运算程序, 236 用户程序, 236-1 第一用户程序, 236-2 第二用户程序, 3100S, 320 辅助程序, 321 编辑程序, 322 编译程序, 323 调试程序, 324 排程构建数据模板, 325 排程构建数据生成程序, 326 通信程序, 330 源程序, 1041 程序区域, 1042 控制程序的工作区域, 1042a 全局变量区域, 1042b 变量同步缓冲区域, 1042c 输出同步缓冲区域, 1042d 输入同步缓冲区域, 1043PLC 系统总线发送缓存, 1044PLC 系统总线接收缓存, 1045 现场网络发送缓存, 1046 现场网络接收缓存, SYS 系统。

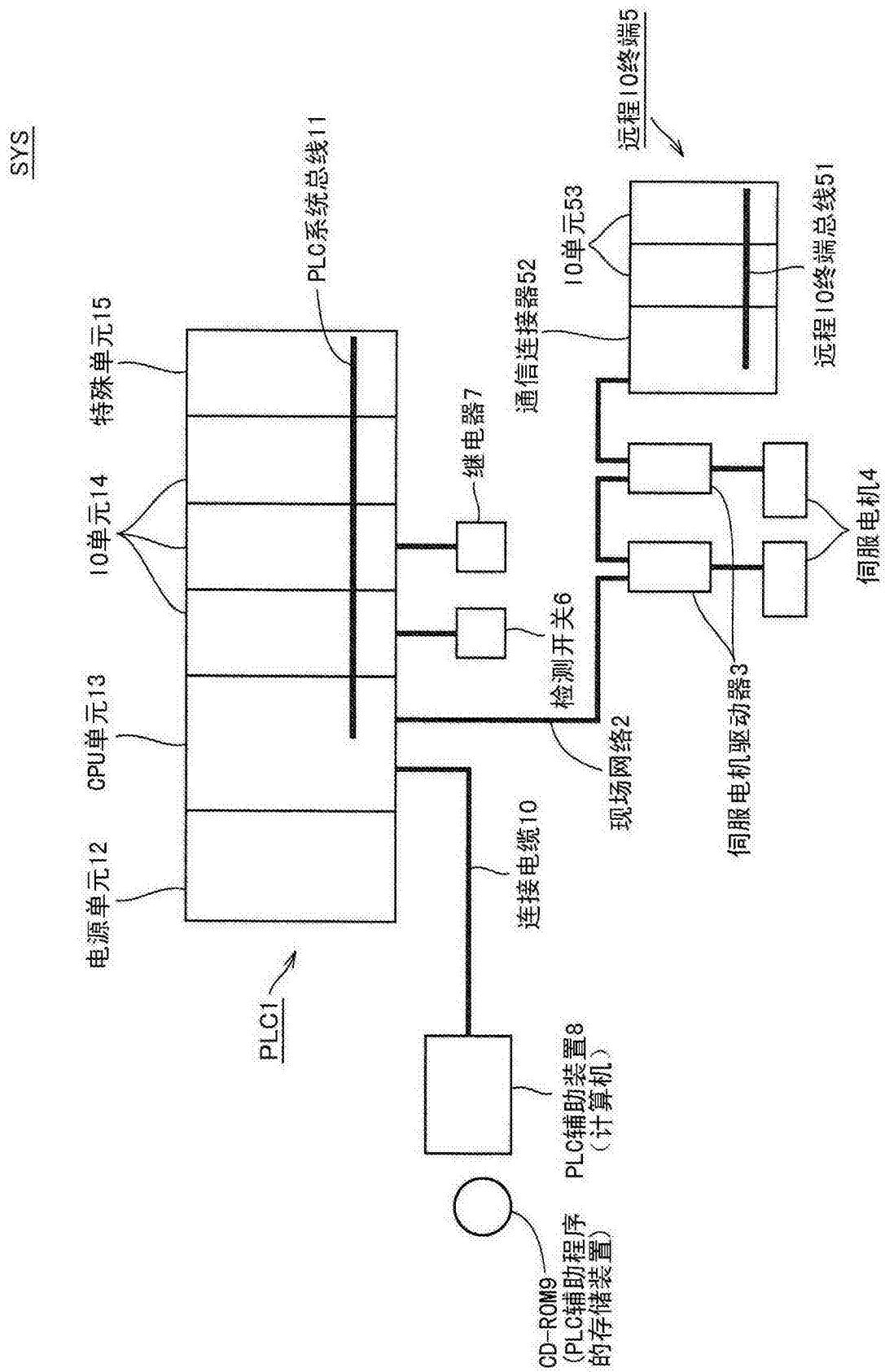


图 1

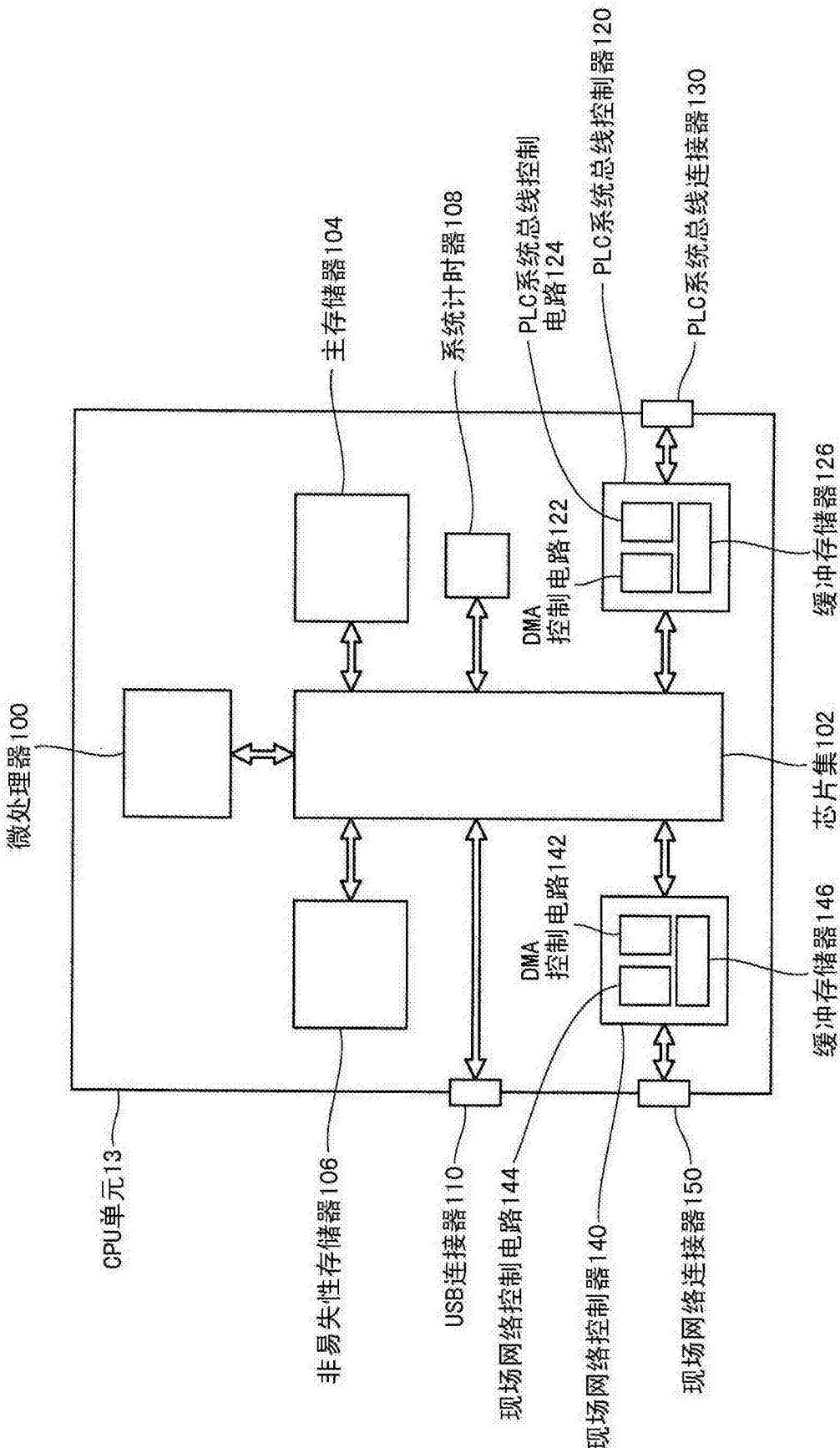


图 2

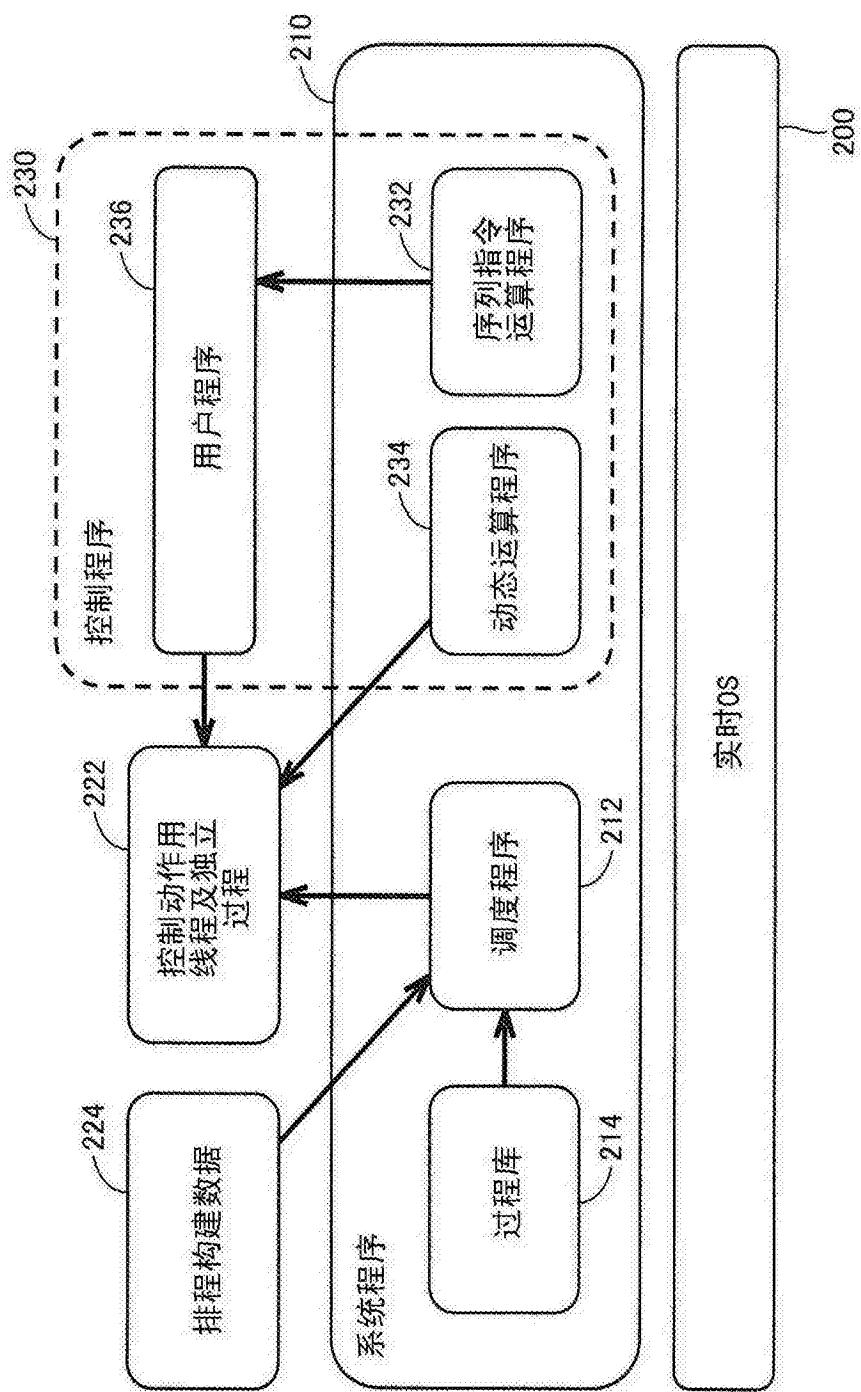


图 3

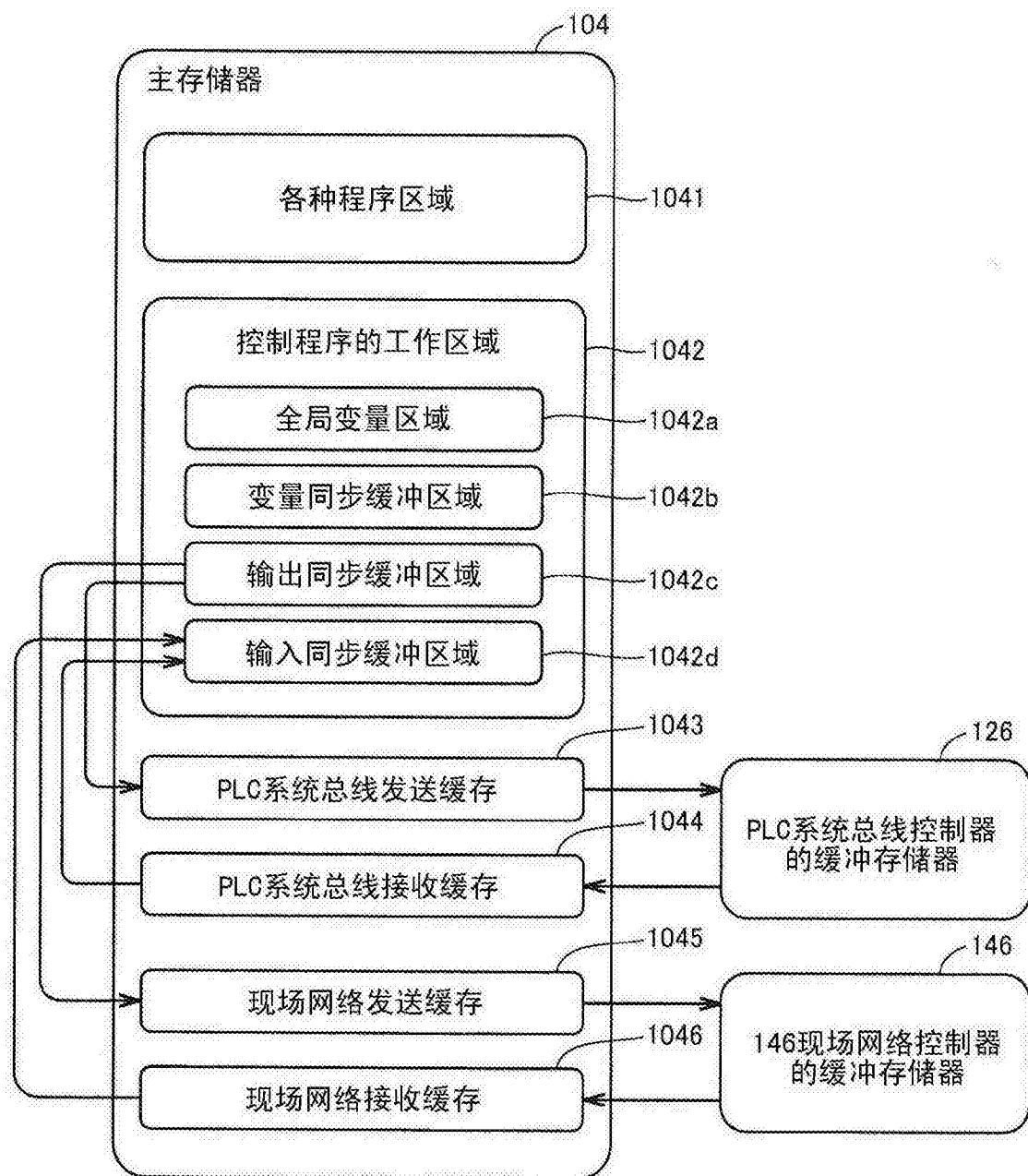


图 4

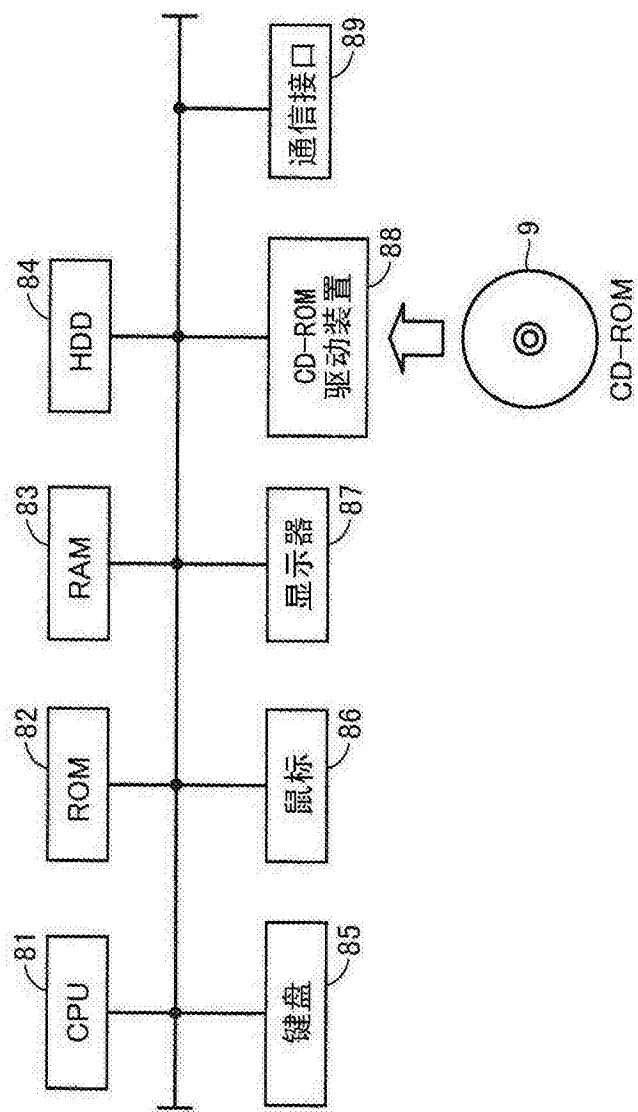
8

图 5

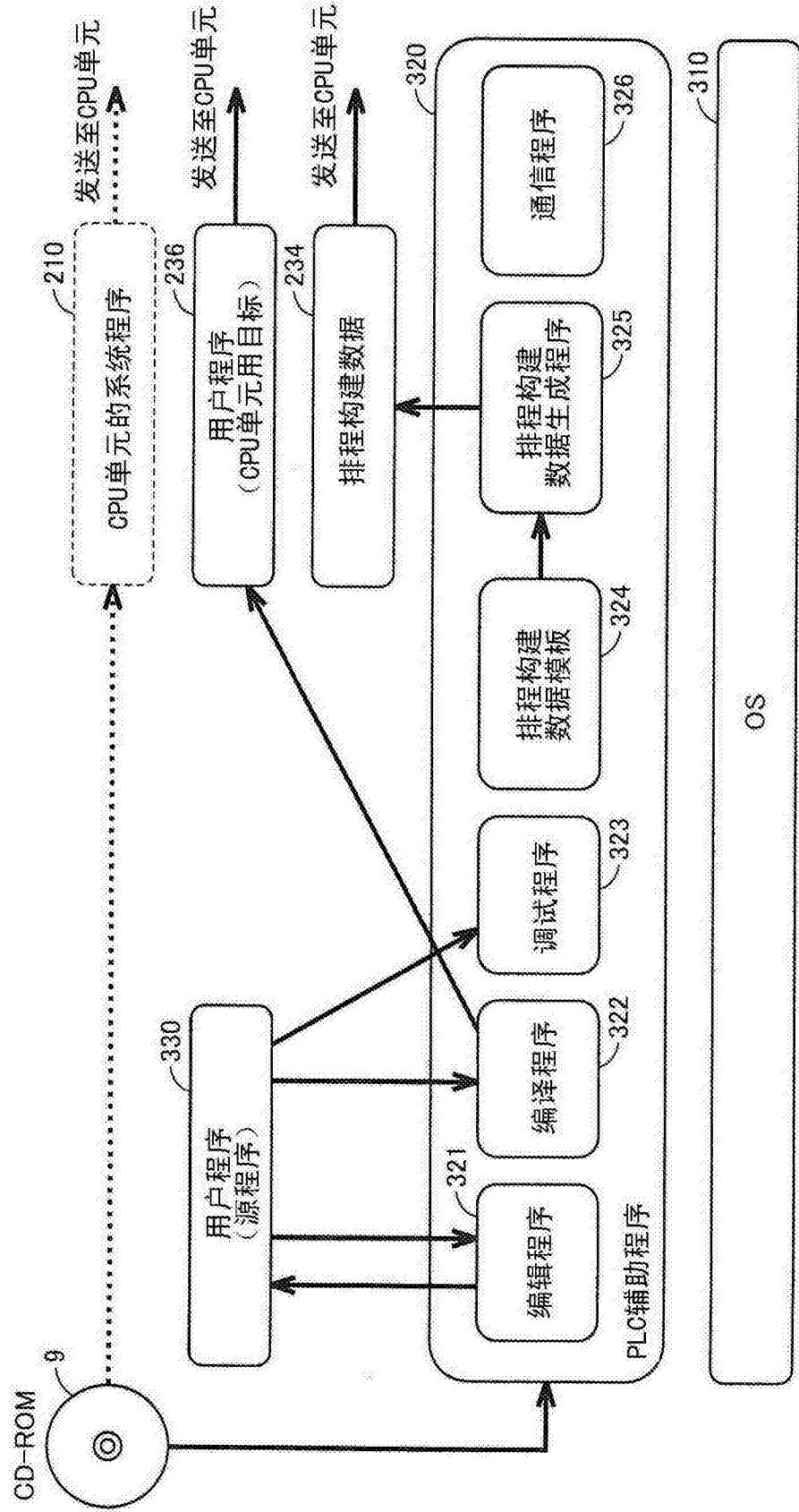


图 6

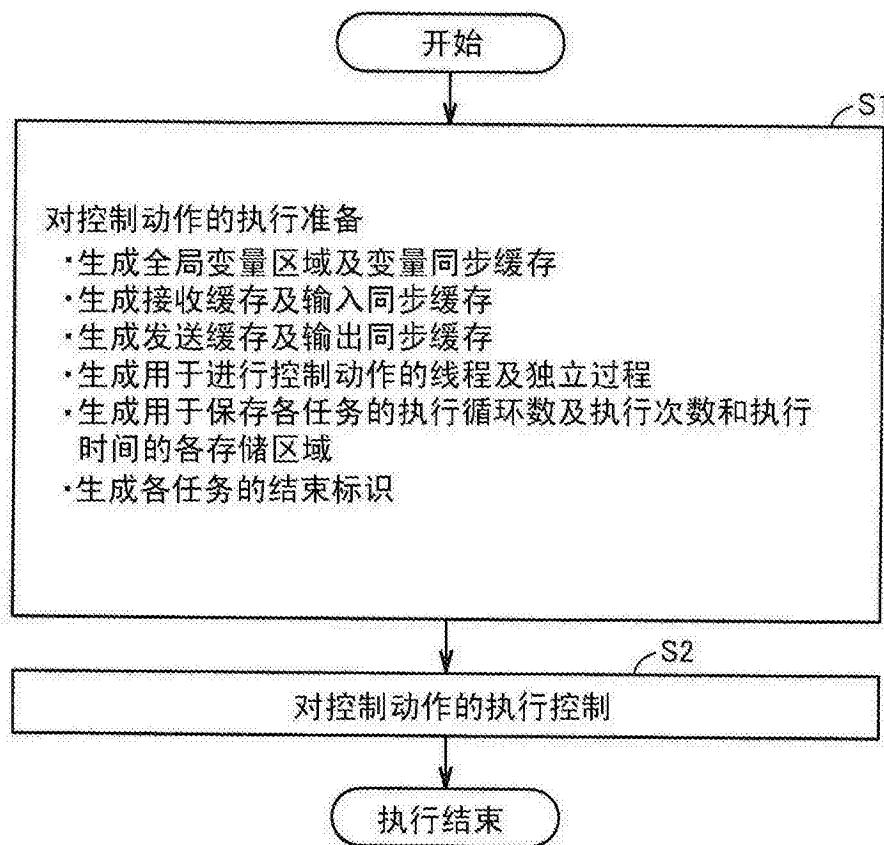


图 7

## 对任务的设定项目的说明

设定项目	说明
任务名称	指定任务的名称
任务开始条件	是任务的开始条件。能够如下指定。 “执行循环周期(n)”：每次执行循环，开始执行任务。 “事件(eee)”：eee是表示事件的名称的字符串。事件是指特定的中断等。若所指定的事件发生，则使处理器在出现空闲时间时执行“任务开始处理”中指定的过程。
任务开始处理	指定在任务开始时执行的过程的名称及优先级。优先级的初始值为2。
线程指定	指定要执行的线程的名称。能够指定多个线程。
任务结束条件	指定线程的名称。在设定项目“任务结束处理”中指定的过程，成为在本项目中指定的线程的后续起动对象。
任务结束处理	指定在任务结束时执行的过程的名称及优先级。优先级的初始值为3。
循环超过容许次数	以整数指定循环超过容许次数。在超过循环超过容许次数而发生超过循环时进行错误处理。 在指令值为“0”的情况下，不容许发生周期超过而在发生一次周期超过时进行错误处理。

对线程的设定项目的说明(1/2)	
设定项目	说明
线程名称	指定线程的名称。
优先级	<p>指定线程的执行优先级(一般指定为4以上)。以整数指定。值越小优先级越高。</p> <p>执行等待队列中的优先级最高的线程。</p>
开始条件	<p>指定线程的开始条件。能够如下指定。</p> <p>“ttt”或“sss”： ttt是表示任务的名称的任务的后续起动对象。若指定任务的名称，则本线程成为所指定的任务的后续起动对象。</p> <p>“事件(eee)”： eee是表示事件的名称的字符串，事件是特定的中断等。若所指定的事件发生，则启动本线程。</p>

图 9A

## 对线程的设定项目的说明 (2/2)

设定项目	说明
过程指定	指定要执行的过程的名称。能够指定多个过程。以例举的顺序执行各过程。
控制程序指定	指定要执行的用户程序的名称或保留字“动态运算程序”。能够指定多个用户程序的名称。若指定了多个用户程序的名称，则以例举的顺序执行多个用户程序。若指定了“动态运算程序”，则为其他线程的用户程序和保留字“动态运算程序”这双方。
动态线程指定	指定如下线程的名称，该线程是指，在设定项目“控制程序指定”中指定有用户程序并且在所指定的用户程序中没有指定的情况下，在执行“动态运算程序而执行”中指定的用户程序之后，接着在本线程中执行动态运算程序。
执行核指定	在用多核芯片执行的情况下，指定用于执行线程的核。

图 9B

名称	功能
控制循环开始指令	起动在各任务的设定项目“任务开始处理”中指定的过程
设定项目	对设定项目的说明
控制循环开始条件	<ul style="list-style-type: none"><li>•“控制循环开始中断”： 在控制循环开始中断时执行。 (若优先级为1，则立即执行开始)</li><li>•“tttt”： tttt是表示任务的名称的字符串。 成为所指定的任务的结束处理的后续起动对象。</li></ul>
优先级	初始值为1。能够变更。

图 10

名称	功能
任务开始过程	<ul style="list-style-type: none"> <li>·对执行循环数进行计数。</li> <li>·在计数之后执行循环数超过了“执行循环周期 (n) + 循环超过容许次数”并且任务结束标识为关 (OFF) 时，进行执行循环超过的错误处理。</li> <li>·在计数之后执行循环数超过了执行循环周期 (n) 并且任务结束标识为开始 (ON) 时，执行以下处理。           <ul style="list-style-type: none"> <li>·将任务的执行循环数设定为“1”<sup>a</sup>。</li> <li>·将任务结束标识设定为关 (OFF)。</li> <li>·使用于计测任务执行时间的计时器开始计时。</li> <li>·起动作为后续起动对象的线程。</li> <li>·在计数之后执行循环数超过了执行循环周期 (n) 并且任务结束标识为关 (OFF) 时，起动作为后续起动对象的线程。 (但是，在作为后继起动对象的线程已被起动或者处于事件(通信结束通知等)等待的暂停状态时，不执行任何处理)。</li> </ul> </li> </ul>
任务结束过程	<ul style="list-style-type: none"> <li>·将任务结束标识设定为开 (ON)。</li> <li>·对任务的执行次数进行计数。</li> <li>·对任务的执行时间进行计测。</li> </ul>
后续起动过程	<ul style="list-style-type: none"> <li>·起动被设定为后续执行对象的线程及过程，此后使本线程的执行结束(设定为 Sleep 状态)。</li> <li>·但不起动任务结束标记为开 (ON) 的任务的线程。</li> </ul>
输出复制过程	将预先指定的输出数据从输出同步缓存复制到发送缓存中。
同步输出复制过程	仅限于执行循环数为“1”的情况，将对该任务预先指定的输出数据从输出同步缓存复制到发送缓存中。 必须在控制循环中在执行相关的控制程序之前执行。

图 11A

名称	功能
输入输出指示过程	向通信电路指示将预先指定的输出数据及输入数据作为对象的输出及输入。
输出指示过程	向通信电路指示将预先指定的输出数据作为对象的输出。
输入指示过程	向通信电路指示将预先指定的输入数据作为对象的输入。
通信结束通知等待过程	在接收到通信结束通知时进行设定以使本线程能够再次起动，使本线程暂停(使得处于Sleep状态)。
输入复制过程	将预先指定的输入数据从接收缓存复制到输入同步缓存中。
同步输入复制过程	仅限于执行循环数为“1”的情况，将对该任务预先指定的输入数据从接收缓存复制到输入同步缓存中。 必须在控制循环中在执行相关的控制程序之前执行。
变量参照开始过程	由参照侧控制程序参照的变量同步缓存不能被改写。
控制程序执行过程	使控制程序开始执行。
变量配送过程	将参照侧控制程序所具有的全局变量的内容复制到参照侧控制程序的同步缓存中。
循环计时过程	获取从控制循环开始起的经过时间，计算出到控制循环时间结束为止的剩余时间，设定处理器内置的计时器以在所计算出的剩余时间经过时使计时中断出现，进行设定以在计时中断时使本线程再次起动，并使本线程暂停(使得处于Sleep状态)。
数据追踪过程	将预先指定的变量的值记录到日志文件中。

图 11B

(1) 控制循环开始条件 优先级	控制循环开始中断 1				
(2) 任务名称 任务开始条件 任务开始处理 线程指定	控制A任务 执行循环周期(1) 任务开始过程、优先级(2) 输入输出A线程 控制A线程	(3) 线程名称 优先级 开始条件 过程指定	输入输出A线程 4 控制A任务 同步输出复制过程 输入输出指示等待过程 通信步输入复制过程 同续起动过程 后		
任务结束条件 任务结束处理 循环超过容许次数	控制A线程 任务结束过程、优先级(3) 1	(4) 线程名称 优先级 开始条件 过程指定	控制A线程 5 输入输出A线程 数据追踪过程 变量参照程序执行过程 控制量配送过程 变量后 续起动过程		
		(5) 线程名称 优先级 开始条件 过程指定	控制程序指定 控制A程序		
		(6) 线程名称 优先级 开始条件 过程指定	控制B线程 6 控制A线程 变量参照开始过程 控制量配送过程 变量后 续起动过程		
		任务结束条件 任务结束处理 循环超过容许次数	控制程序指定 控制B程序 1		

图 12

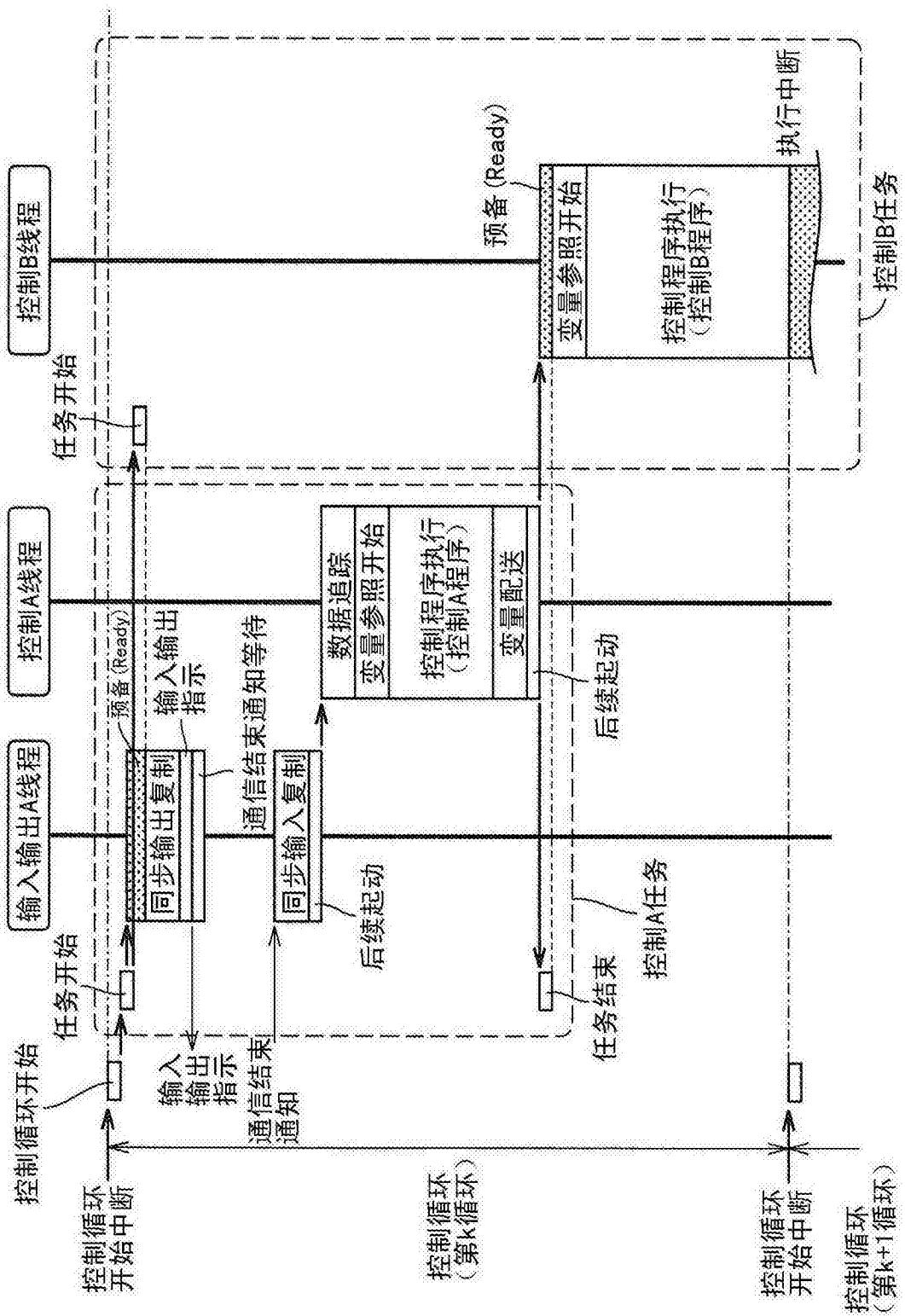


图 13

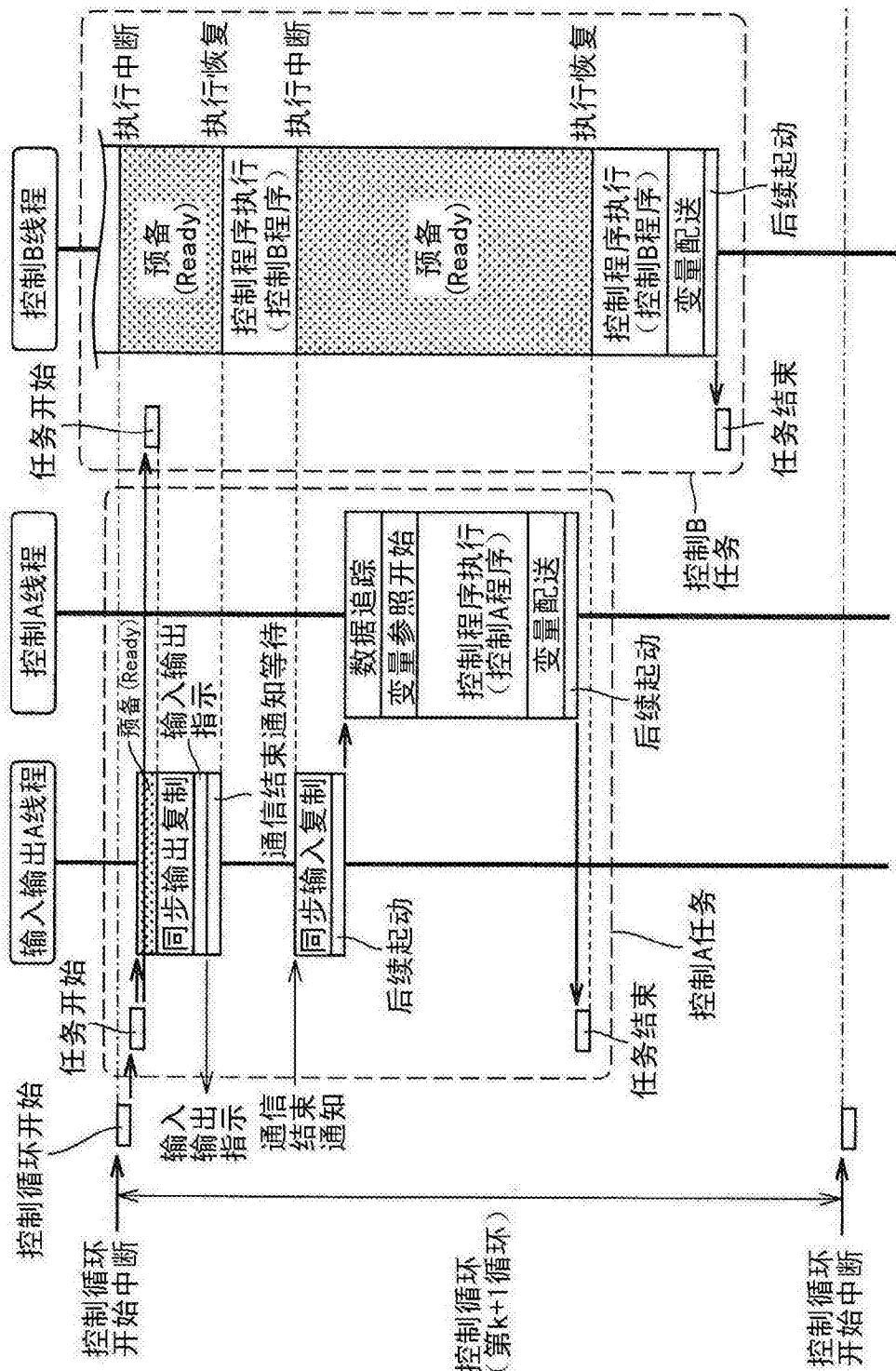


图 14

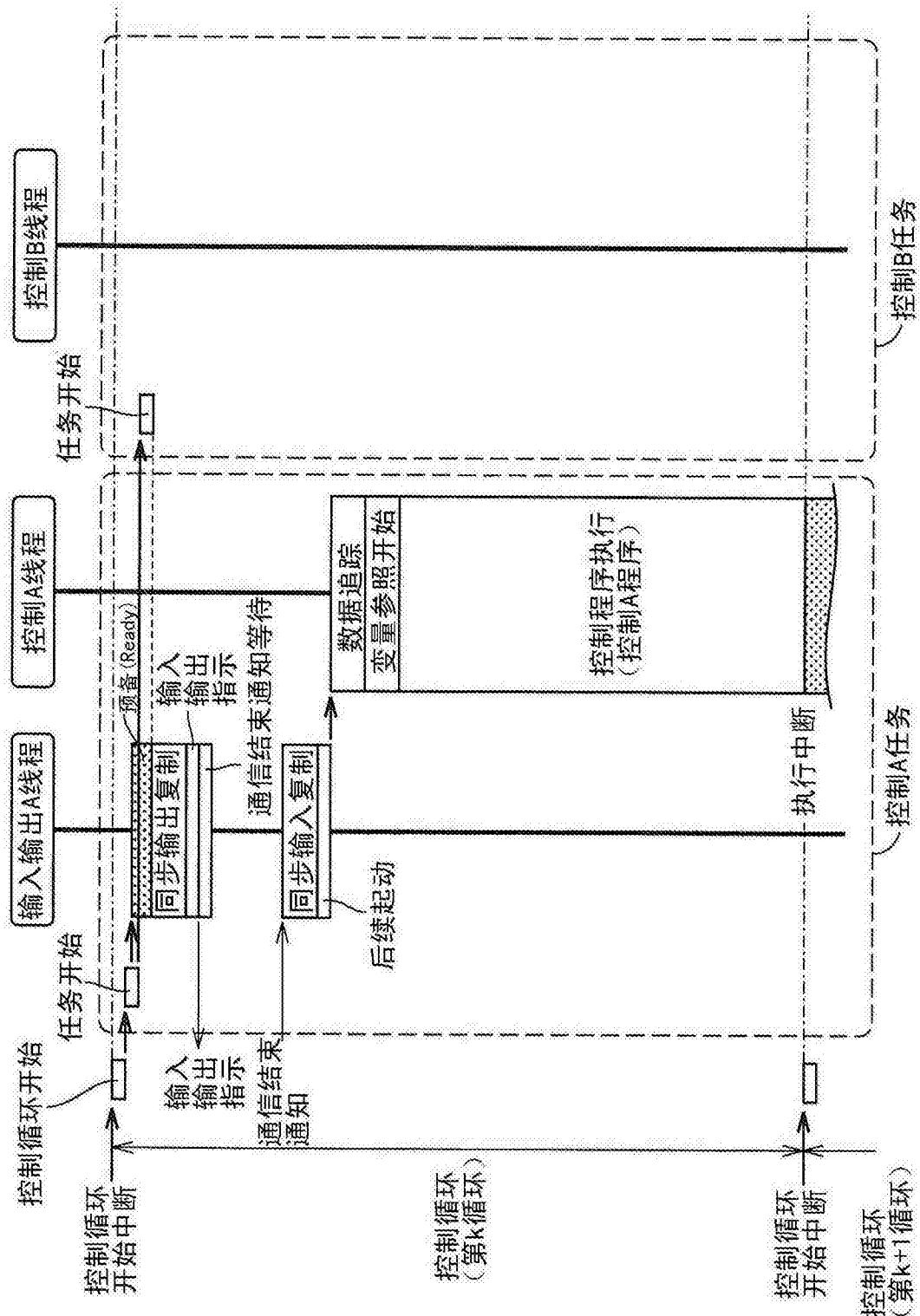


图 15

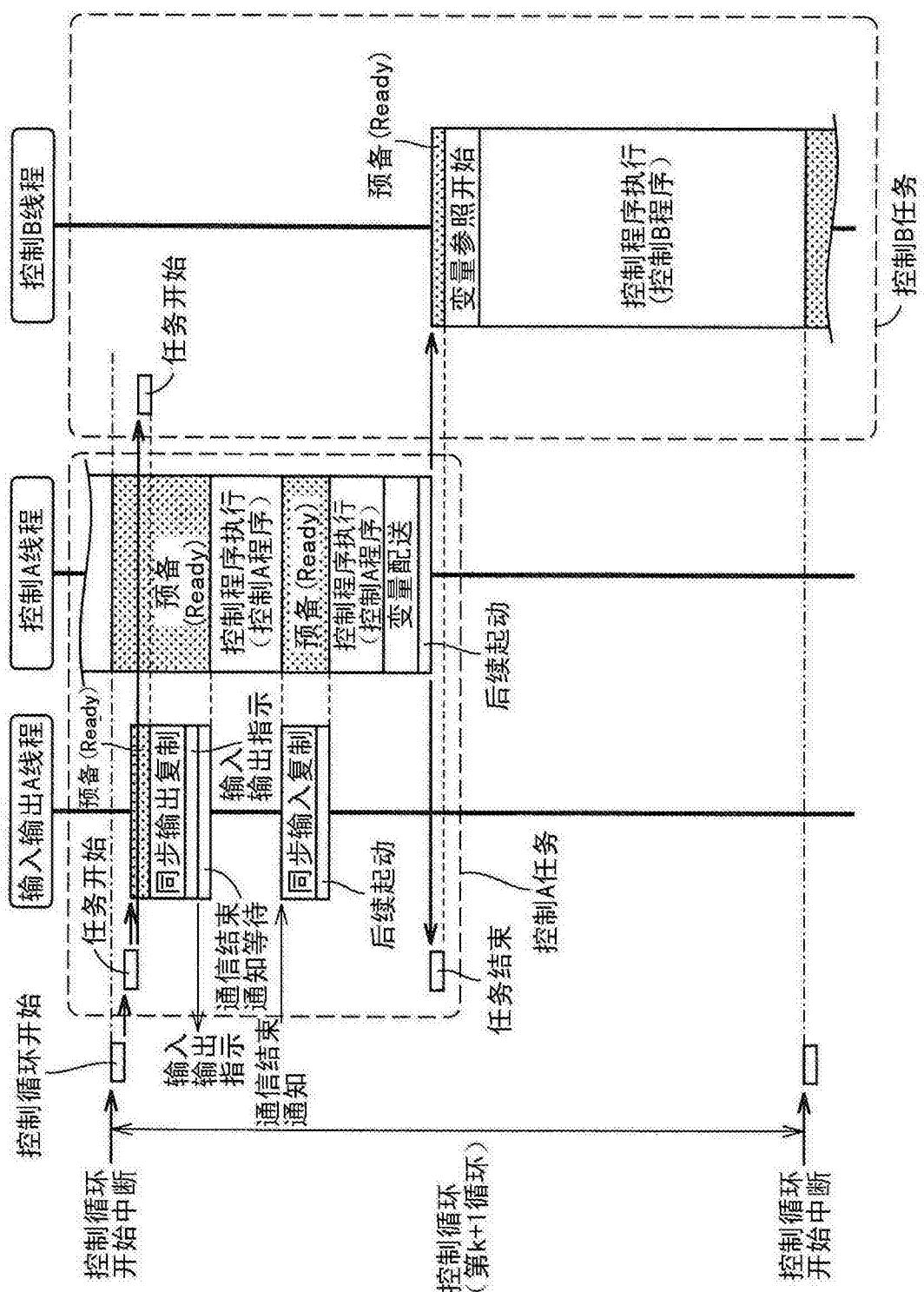


图 16

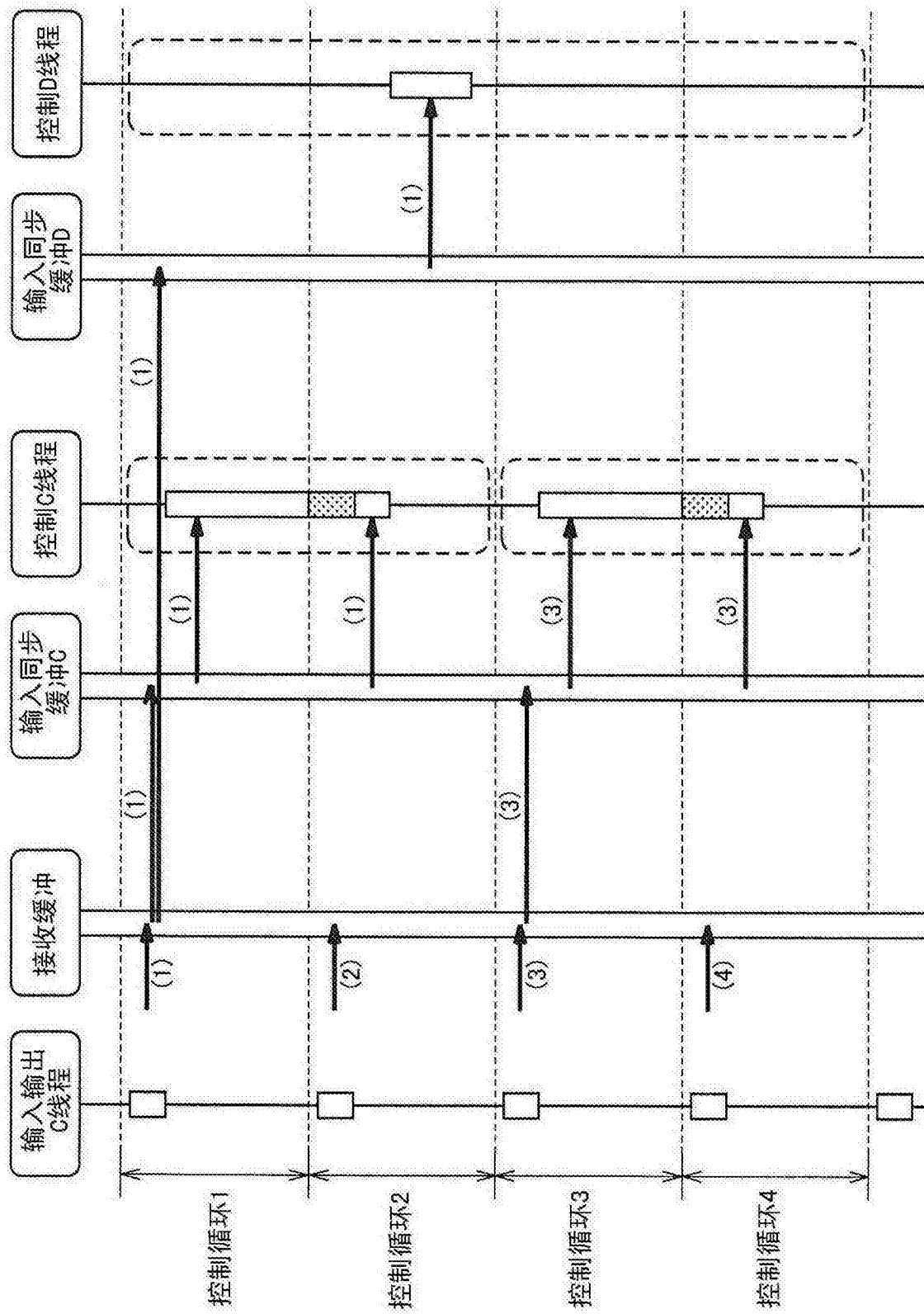


图 17

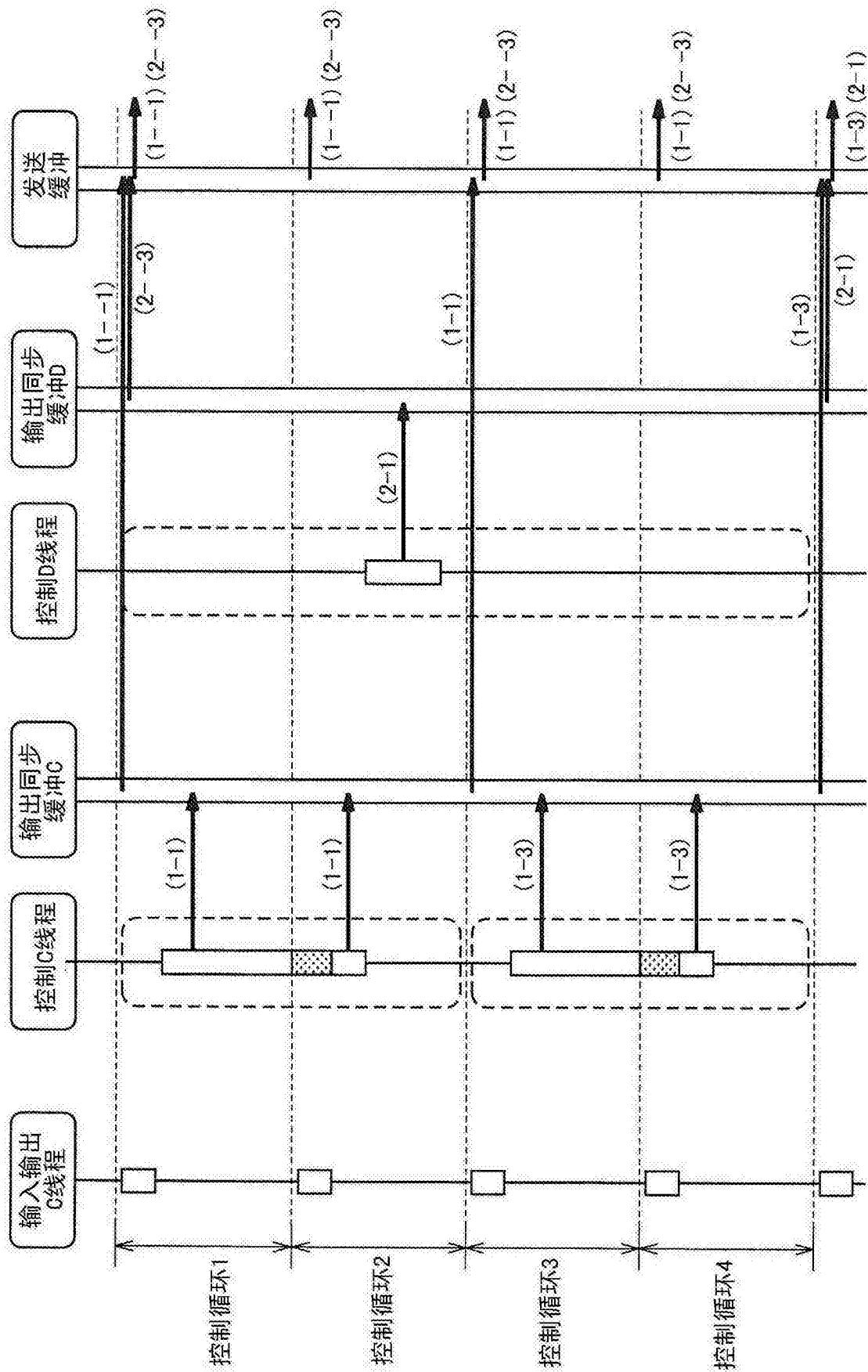


图 18

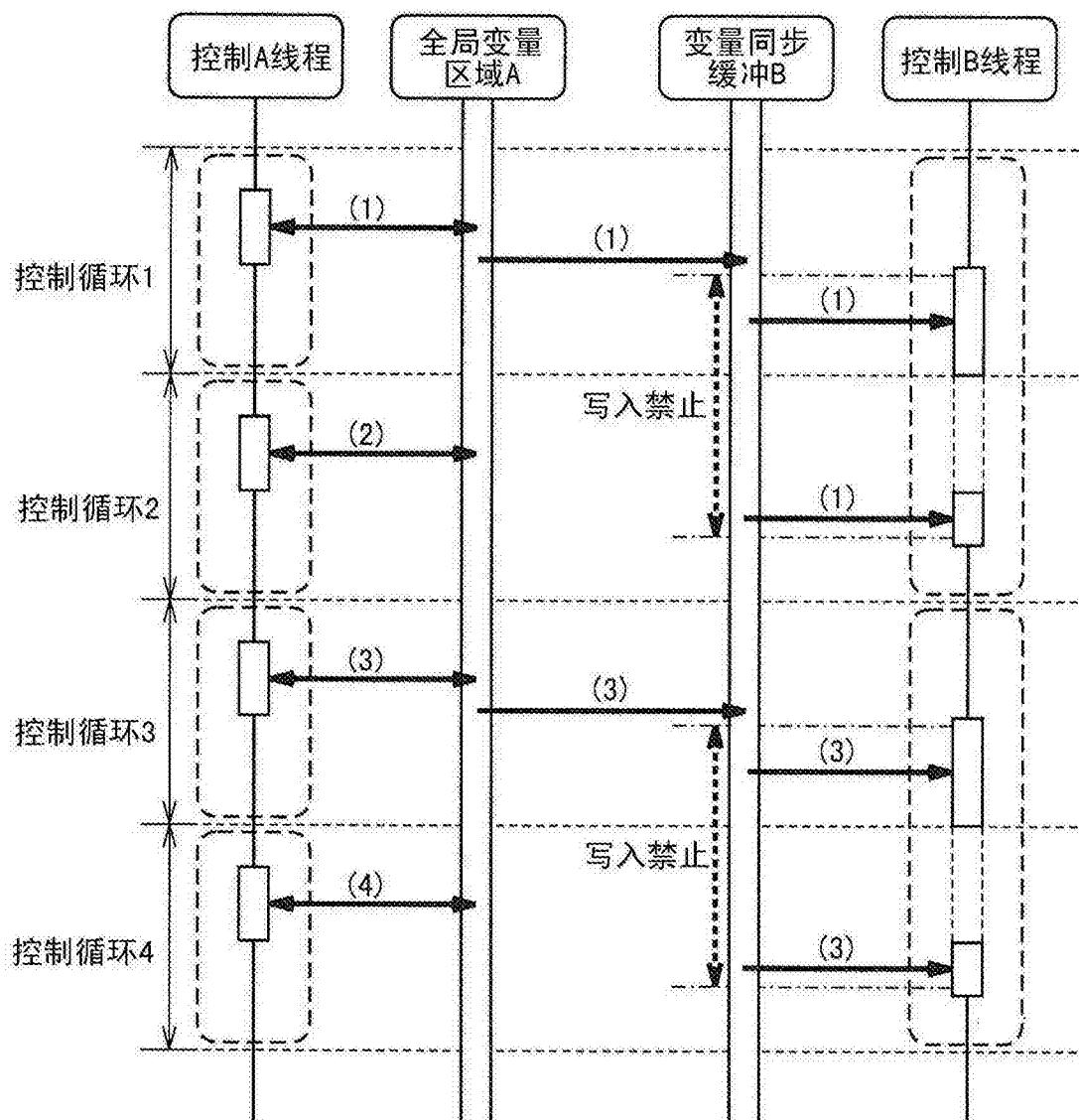


图 19

(1)	控制循环开始条件 优先级	1	控制循环开始中断		
(2)	任务名称 任务开始处理 任务开始处理 线程指定	控制E任务 执行循环周期(1) 任务开始过程、优先级(2) 输出E线程 输入E线程 控制E线程	(3)	线程名称 优先级 开始条件 过程指定	输出E线程 4 控制E任务 同步输出复制过程 输出指示过程 后续起动过程
	任务结束条件 任务结束处理 循环超过容许次数	控制E线程 任务结束过程 1	(4)	线程名称 优先级 开始条件 过程指定	控制E线程 5 输入E线程 通信结束通知等待过程 同步输入复制过程 后续起动过程
			(5)	线程名称 优先级 开始条件 过程指定	控制E线程 6 输入E线程 数据追踪开始过程 数据参照执行过程 变量参考序执行过程 变量配送过程 变量配送起动过程 后续起动过程
				指定控制程序	控制E线程
(6)	任务名称 任务开始条件 任务开始处理 线程指定	控制F任务 执行循环周期(2) 任务开始过程、优先级(2) 控制F线程	(6)	线程名称 优先级 开始条件 过程指定	控制F线程 7 控制F任务 变量参考开始过程 变量参照执行过程 控制程序配送过程 变量配送起动过程
	任务结束条件 任务结束处理 循环超过容许次数	控制F线程 任务结束过程 1		控制程序指定	控制F程序

图 20

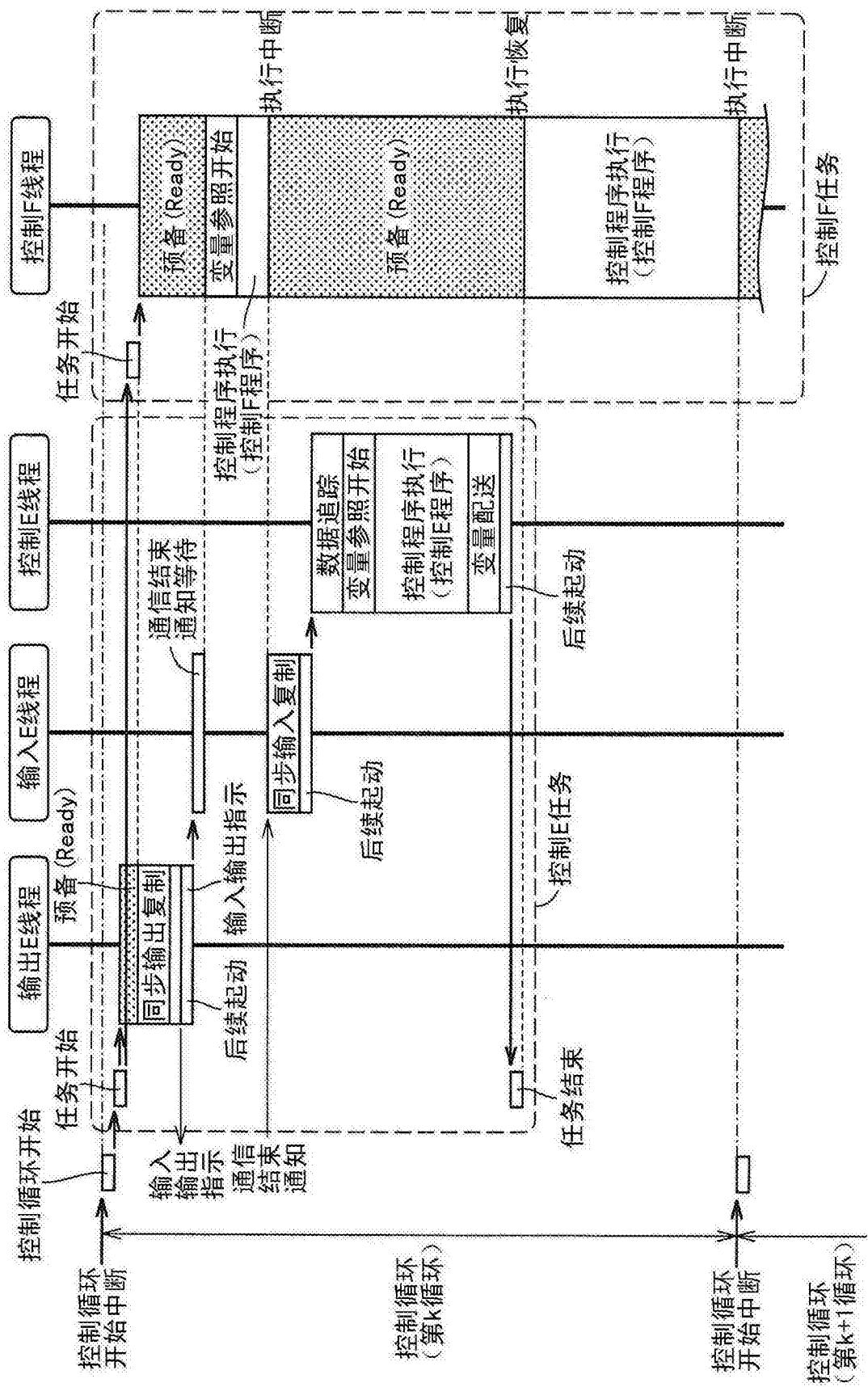


图 21

(1)	控制循环开始条件 优先级	控制循环开始中断 1	(3) 线程名称 优先级	输出G线程 4	(3) 线程名称 优先级	输出G线程 4
(2)	任务名称 任务开始条件 任务开始处理 线程指定	控制G任务 执行循环周期(1) 任务开始过程、优先级(2) 输入输出G线程 控制G线程	开始条件 过程指定	同步输入输出 通信输入 同步输入 信号输入 通知启动过程 后续起动过程	控制G任务 输出指示 复制过程 等待过程	同步输入输出 通信输入 同步输入 信号输入 通知启动过程 后续起动过程
(4)	任务名称 任务结束条件 任务结束处理 循环超过次数	控制G线程 任务结束过程、优先级(3) 1	开始条件 过程指定	6	控制G线程 输入输出G线程 数据追踪过程 参照开始过程 控制程序 变量配送过程 变量执行过程 变量发送过程 变量接收过程	控制G线程 输入输出G线程 数据追踪过程 参照开始过程 控制程序 变量配送过程 变量执行过程 变量发送过程 变量接收过程
(5)	任务名称 任务开始条件 任务开始处理 线程指定	控制H任务 执行循环周期(2) 任务开始过程、优先级(2) 输入输出H线程 控制H线程	开始条件 过程指定	5	控制程序指定 控制A程序	控制程序指定 控制A程序
(6)	任务名称 任务结束条件 任务结束处理 循环超过次数	控制H线程 任务结束过程、优先级(3) 1	开始条件 过程指定	6	控制H线程 输入输出H线程 输入输出 通信输入 通知启动过程 后续起动过程	控制H线程 输入输出H线程 输入输出 通信输入 通知启动过程 后续起动过程
(7)	任务名称 优先级 开始条件 过程指定 指定控制程序	控制H线程 7	开始条件 过程指定 指定控制程序	7	控制H线程 输入输出H线程 输入输出 通信输入 通知启动过程 后续起动过程 控制B程序	控制H线程 输入输出H线程 输入输出 通信输入 通知启动过程 后续起动过程 控制B程序

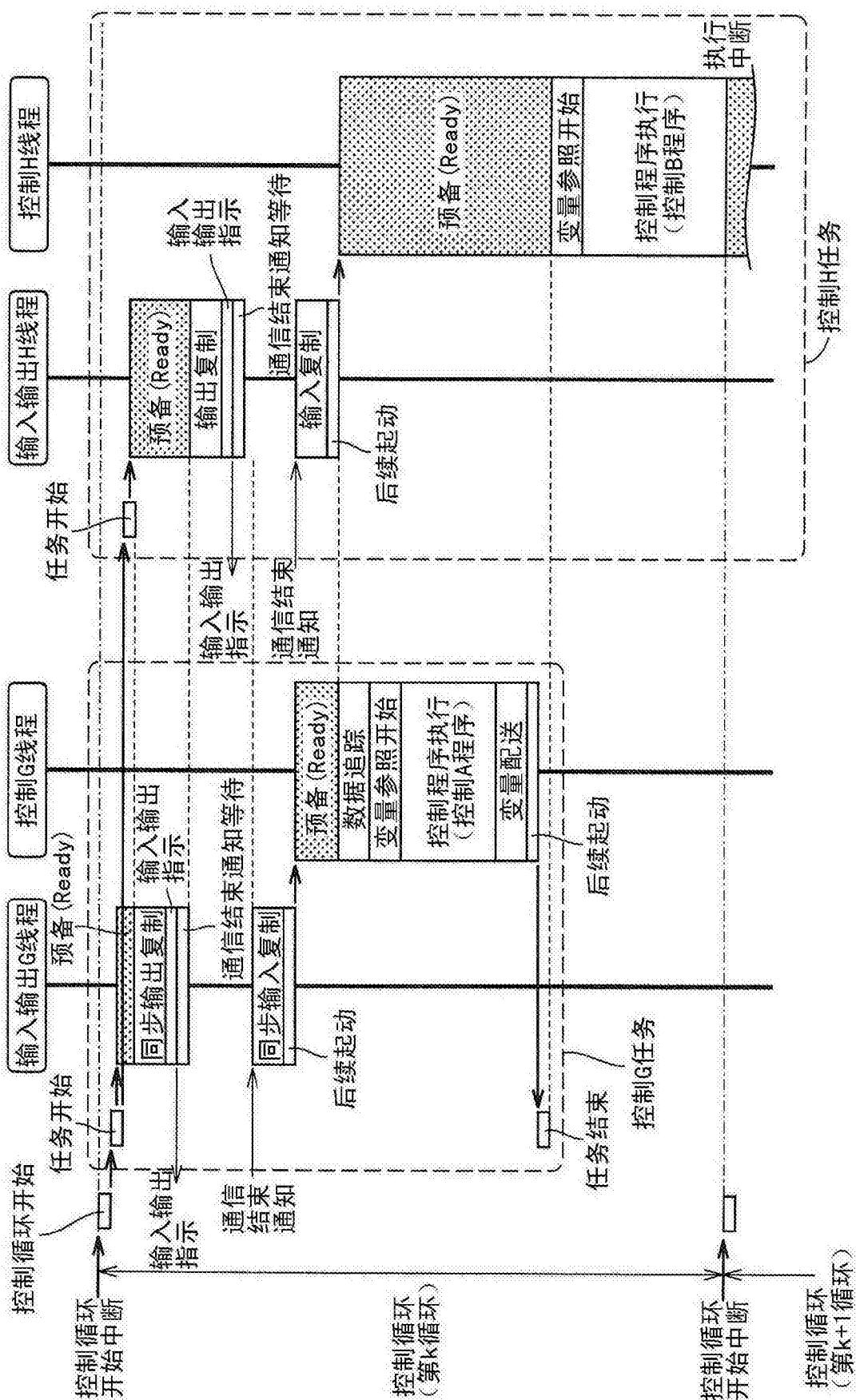


图 23

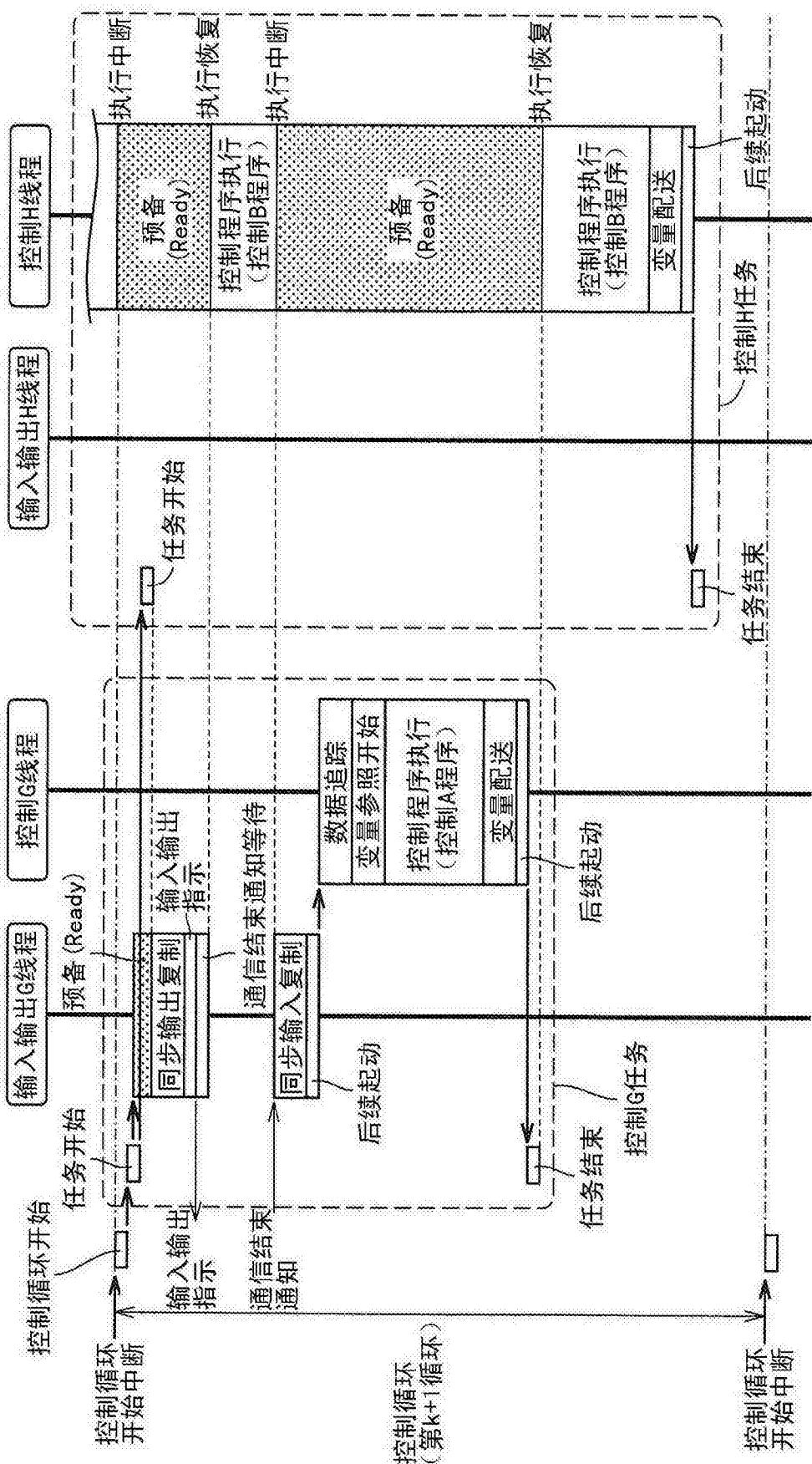


图 24

(1)	控制循环开始条件	控制循环开始中断
优先级	1	
(2)	任务名称	控制J任务
任务开始条件	执行循环周期(1)	
任务开始处理	任务开始过程、优先级(2)	
线程指定	输入输出J线程	
任务结束条件	控制J线程	
任务结束处理	任务结束过程、优先级(3)	
循环超过容许次数	1	
(3)	线程名称	输入输出J线程
优先级	4	
开始条件	控制J任务	
过程指定	同步输出复制过程 输出指示过程 输出结束输入过程 输出信令过程 同步输入过程 后续输入过程 后续起动过程	
(4)	线程名称	控制J线程
优先级	5	
开始条件	输入输出J线程	
过程指定	控制程序执行过程 后续起动过程	
控制程序指定	动态运算程序	
(5)	线程名称	控制K线程
优先级	6	
开始条件	控制J线程	
过程指定	数据追踪过程 控制程序执行过程 后续起动过程	
任务结束条件	控制K线程	
任务结束处理	任务结束过程、优先级(3)	
循环超过容许次数	1	
(6)	线程名称	控制K线程
优先级	6	
开始条件	控制J线程	
过程指定	控制程序指定 序列K程序	
控制程序指定	动态线程指定	

图 25

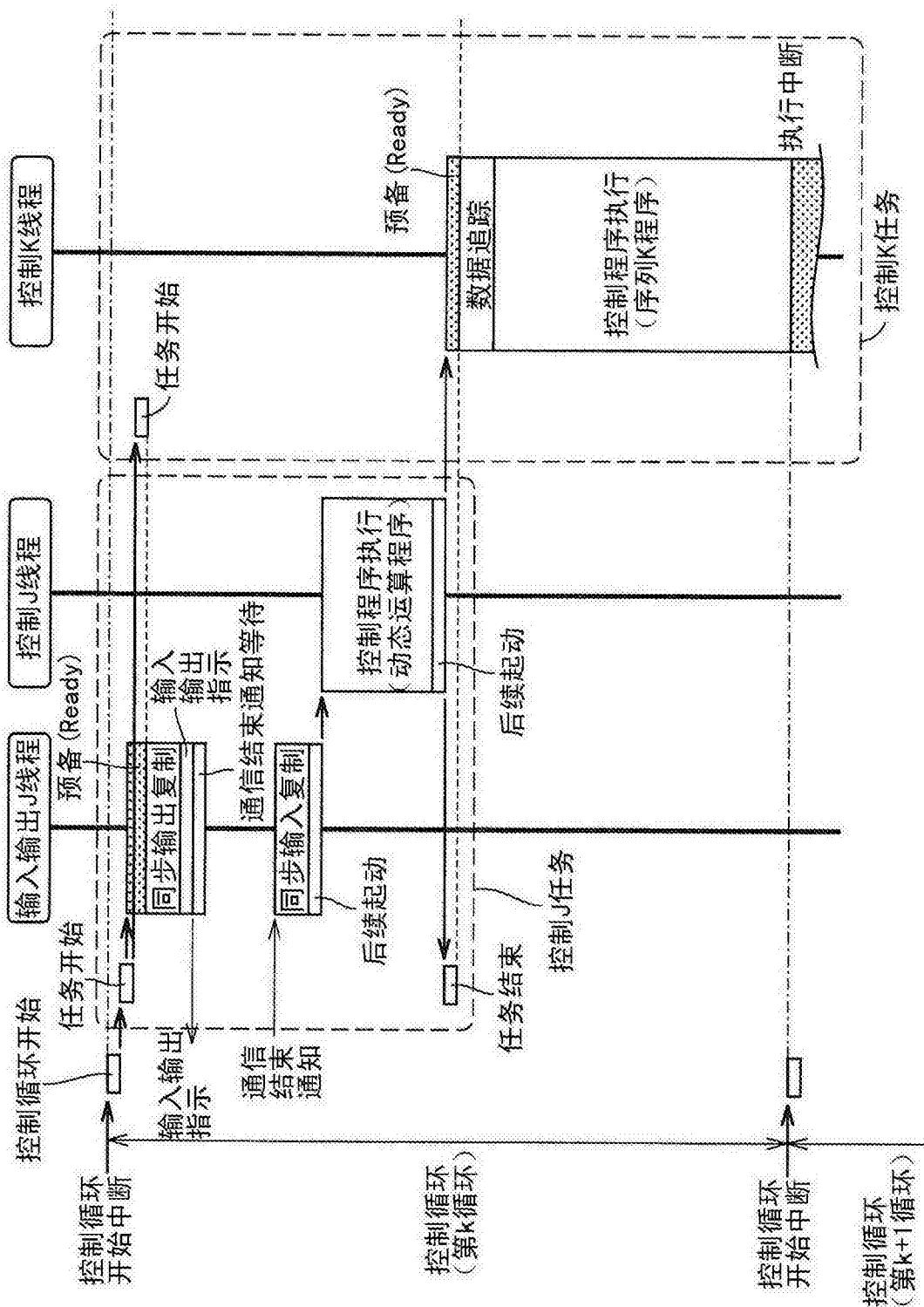


图 26

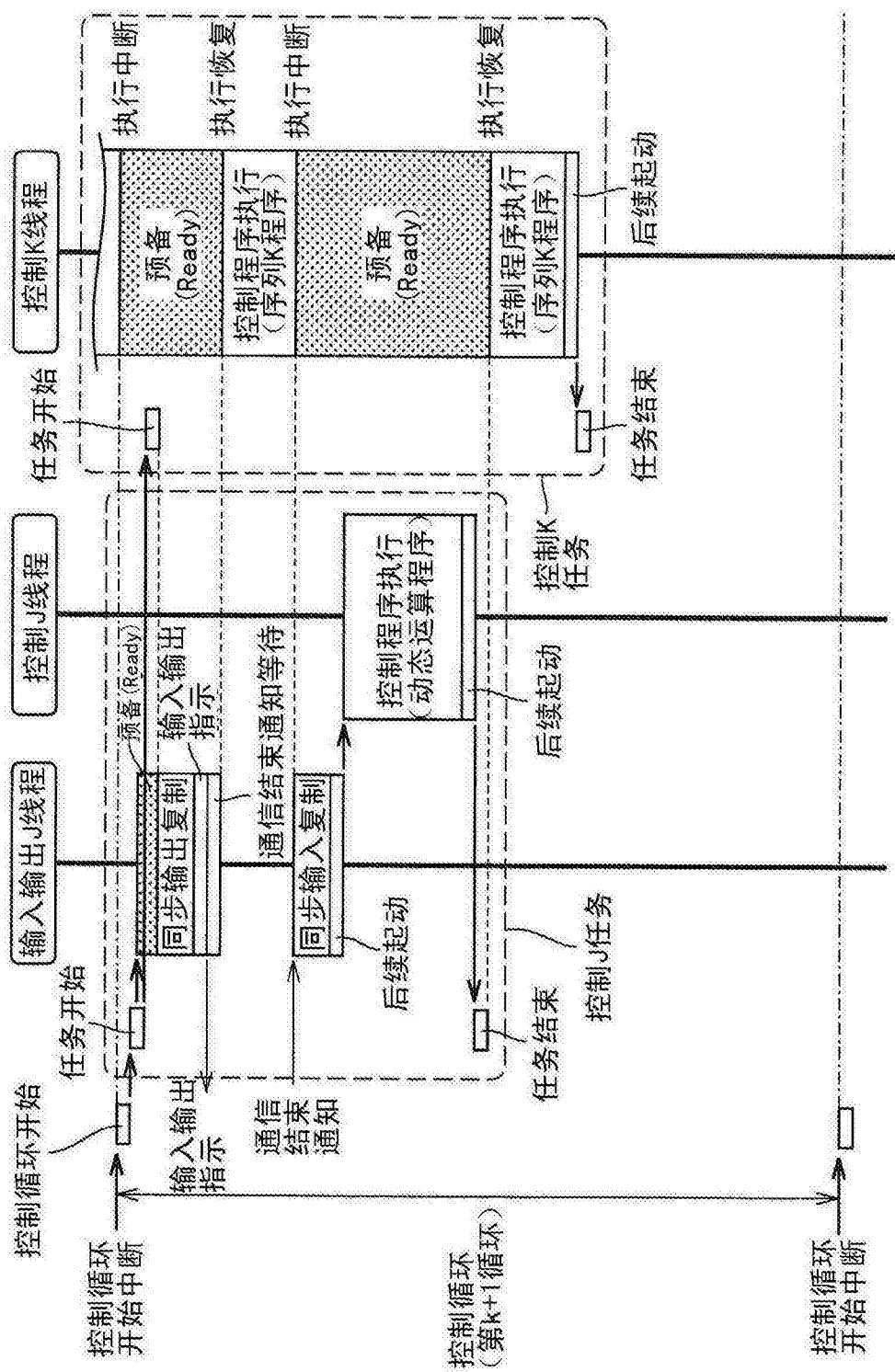


图 27

(1)	控制循环开始条件 优先级	控制循环开始中断 1	
(2)	任务名称 任务开始条件 任务开始处理 线程指定	控制L1任务 执行循环周期(1) 任务开始过程、优先级(2) 输入输出L1线程 控制L1线程 控制L2线程	
(3)	任务名称 优先级 开始条件 过程指定	输入输出L1线程 4 控制L1任务 输出复制过程 输出输出结束通知等待过程 输入信号复制过程 输入后续起动过程	
(4)	任务名称 优先级 开始条件 过程指定 控制程序指定 动态线程指定	控制L1线程 5 输入输出L1线程 数据追踪过程 控制程序执行过程 后续起动过程 序列L1程序 控制L2线程	
(5)	任务名称 优先级 开始条件 过程指定 控制程序指定	控制L2线程 6 控制L1线程 控制程序执行过程 后续起动过程 动态运算程序	

图 28

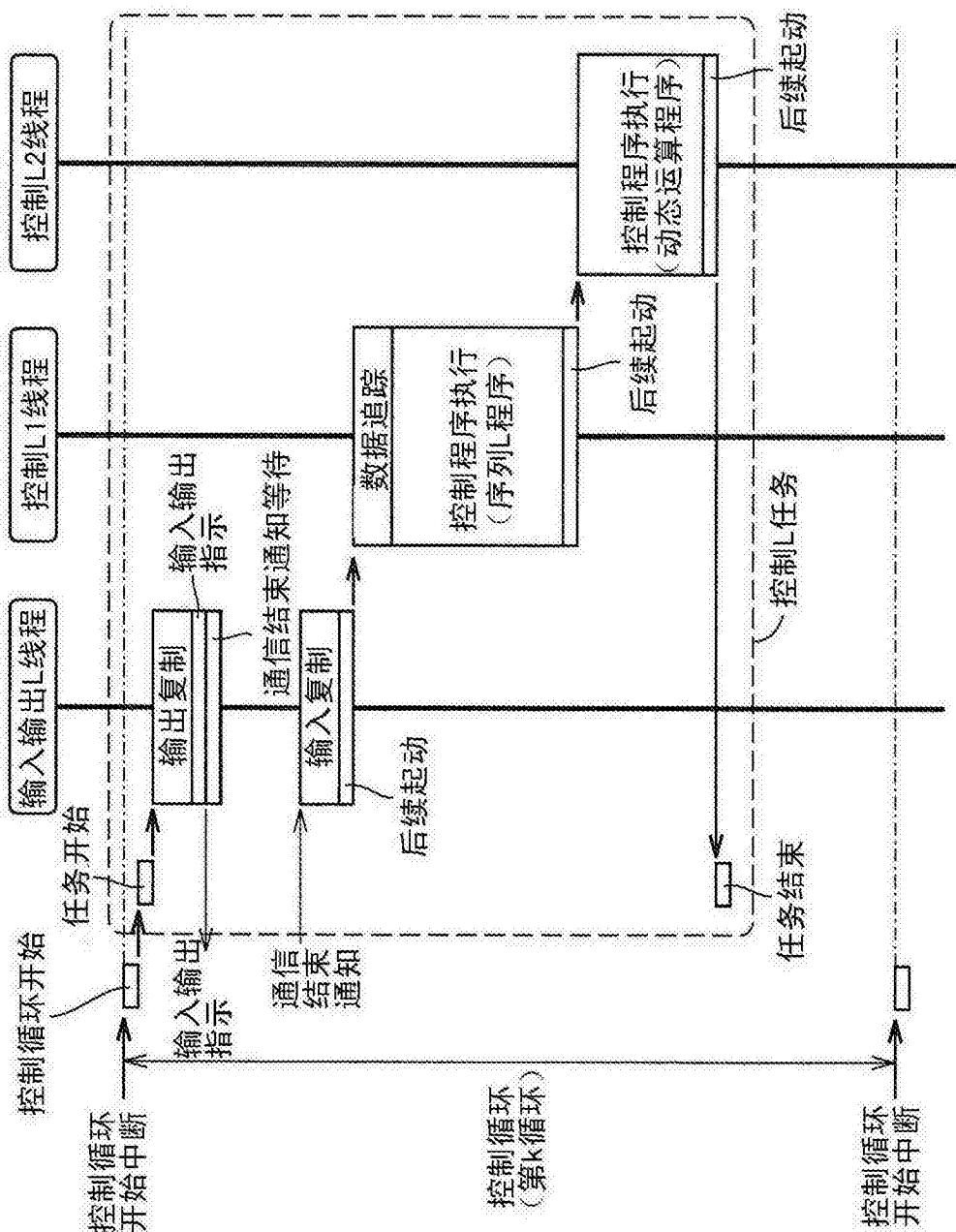


图 29

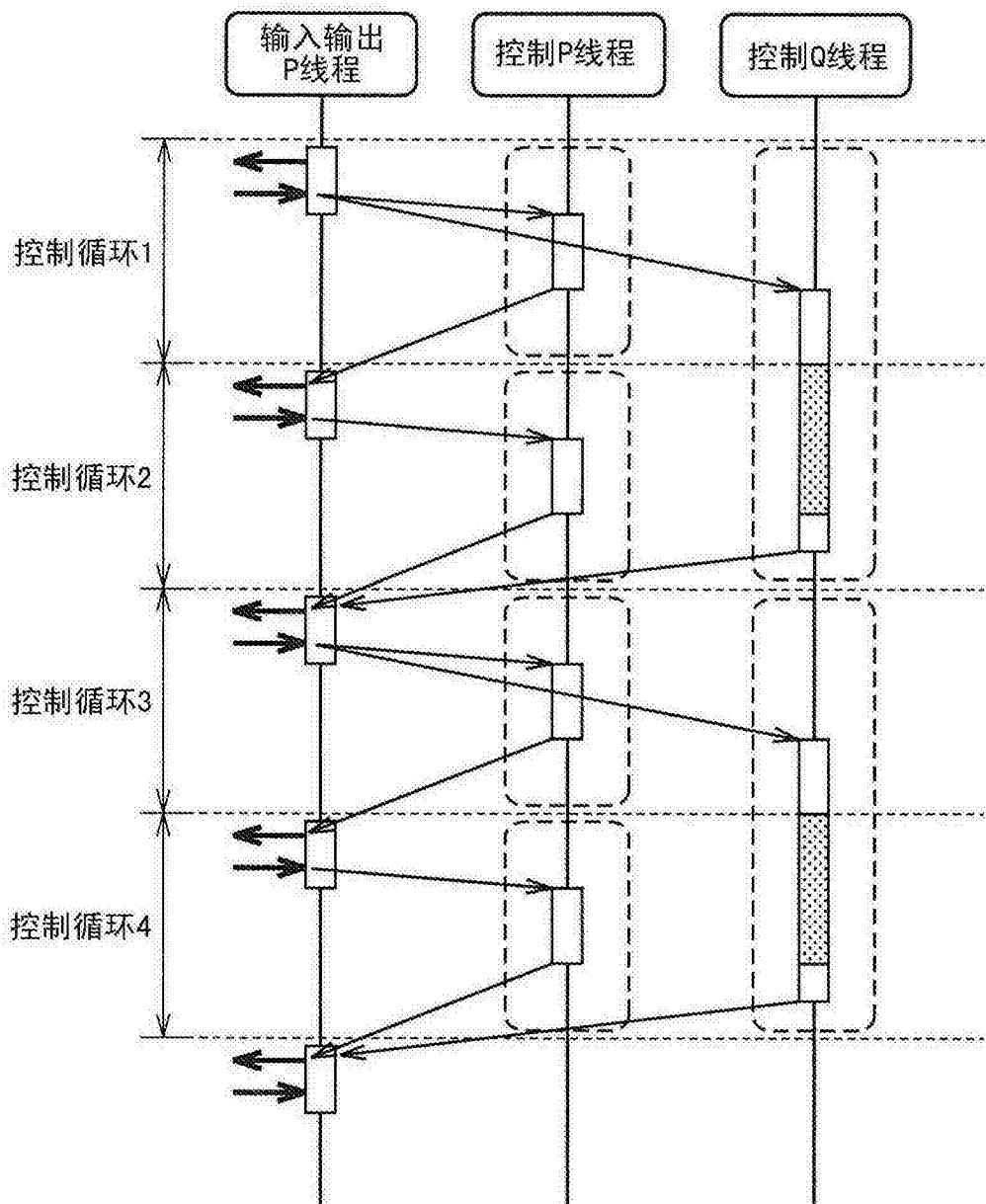


图 30

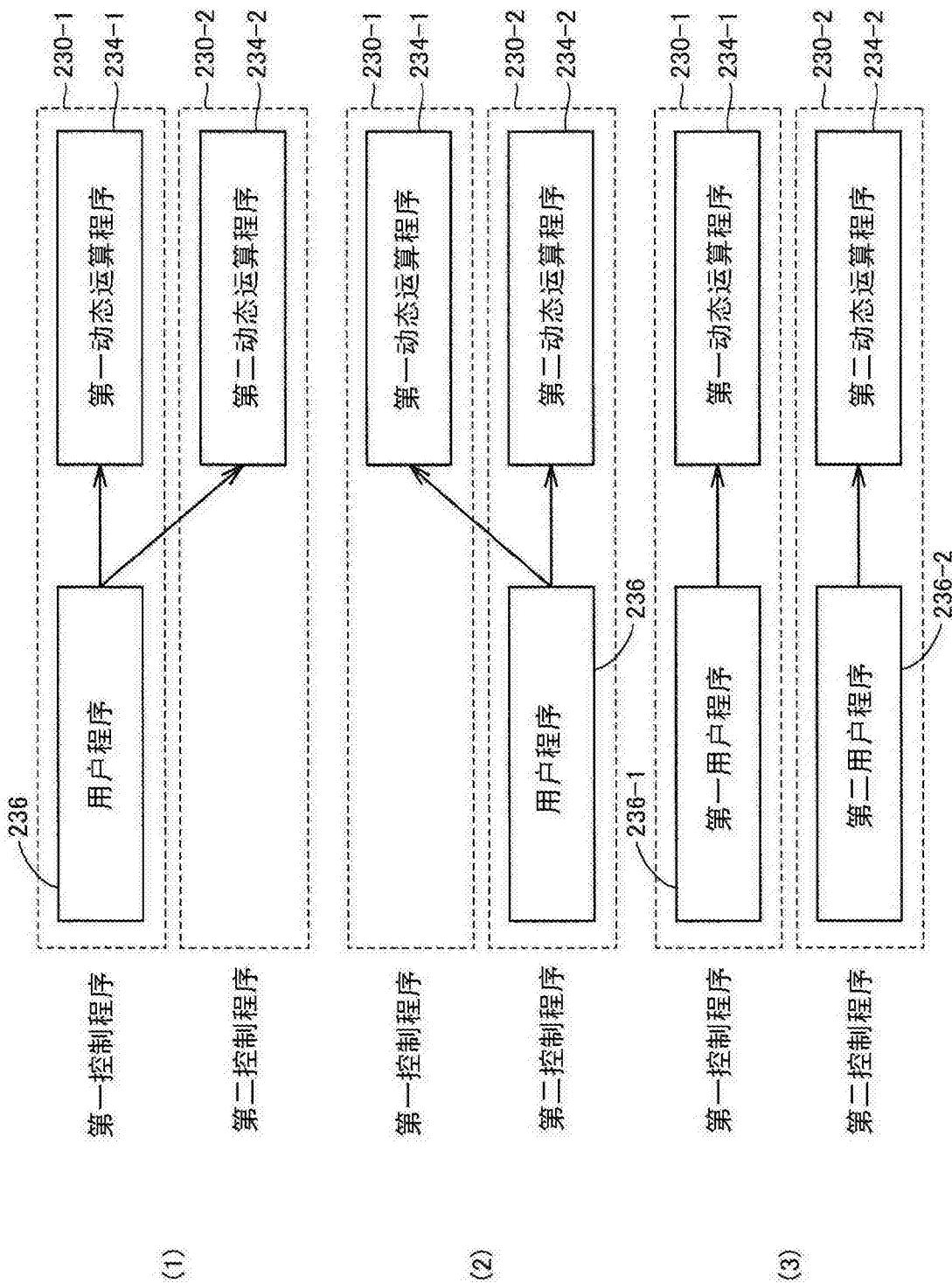


图 31

(1) 控制循环开始条件 优先级		控制循环开始中断 1		(3) 线程名称 优先级		输入输出R线程 4	
(2) 任务名称	控制R任务	开始条件	过程指定	控制R任务	输出复制过程 输出指示过程 输入输出通知和等待过程 输入输出命令 输入输出命令 输入输出命令	同步输出 输入输出命令 输入输出命令 输入输出命令 输入输出命令 输入输出命令	控制R任务
任务开始条件	执行循环周期(1)						
任务开始处理	任务开始过程、优先级(2)						
线程指定	输入输出R线程						
任务结束条件	控制R线程	执行核指定	第一核				
任务结束处理	任务结束过程、优先级(3)						
循环超过容许次数	1						
(4) 线程名称 优先级		控制R线程 5		输入输出R线程 5		数据追踪过程 数据参照开始过程 变量参照开始过程 变量参照开始过程 变量参照开始过程 变量参照开始过程	
任务名称	控制R任务	开始条件	过程指定	控制R任务	输出复制过程 输出命令 输入输出命令 输入输出命令 输入输出命令 输入输出命令	变量程序送过程 变量配对过程 变量配对过程 变量配对过程 变量配对过程 变量配对过程	控制R任务
任务开始条件	执行循环周期(2)						
任务开始处理	任务开始过程、优先级(2)						
线程指定	控制S线程						
任务结束条件	控制S线程						
任务结束处理	任务结束过程、优先级(3)	控制程序指定	第一核				
循环超过容许次数	1	执行核指定	第一核				
(5) 线程名称 优先级		控制S任务 6		控制S线程 6		输入输出R线程 6	
任务名称	控制S任务	开始条件	过程指定	控制S任务	输出开始过程 输入输出命令 输入输出命令 输入输出命令 输入输出命令 输入输出命令	变量参照开始过程 变量程序送过程 变量配对过程 变量配对过程 变量配对过程 变量配对过程	控制S任务
任务开始条件	执行循环周期(2)						
任务开始处理	任务开始过程、优先级(2)						
线程指定	控制S线程						
任务结束条件	控制S线程						
任务结束处理	任务结束过程、优先级(3)	控制程序指定	第二核				
循环超过容许次数	1	执行核指定	第二核				

图 32

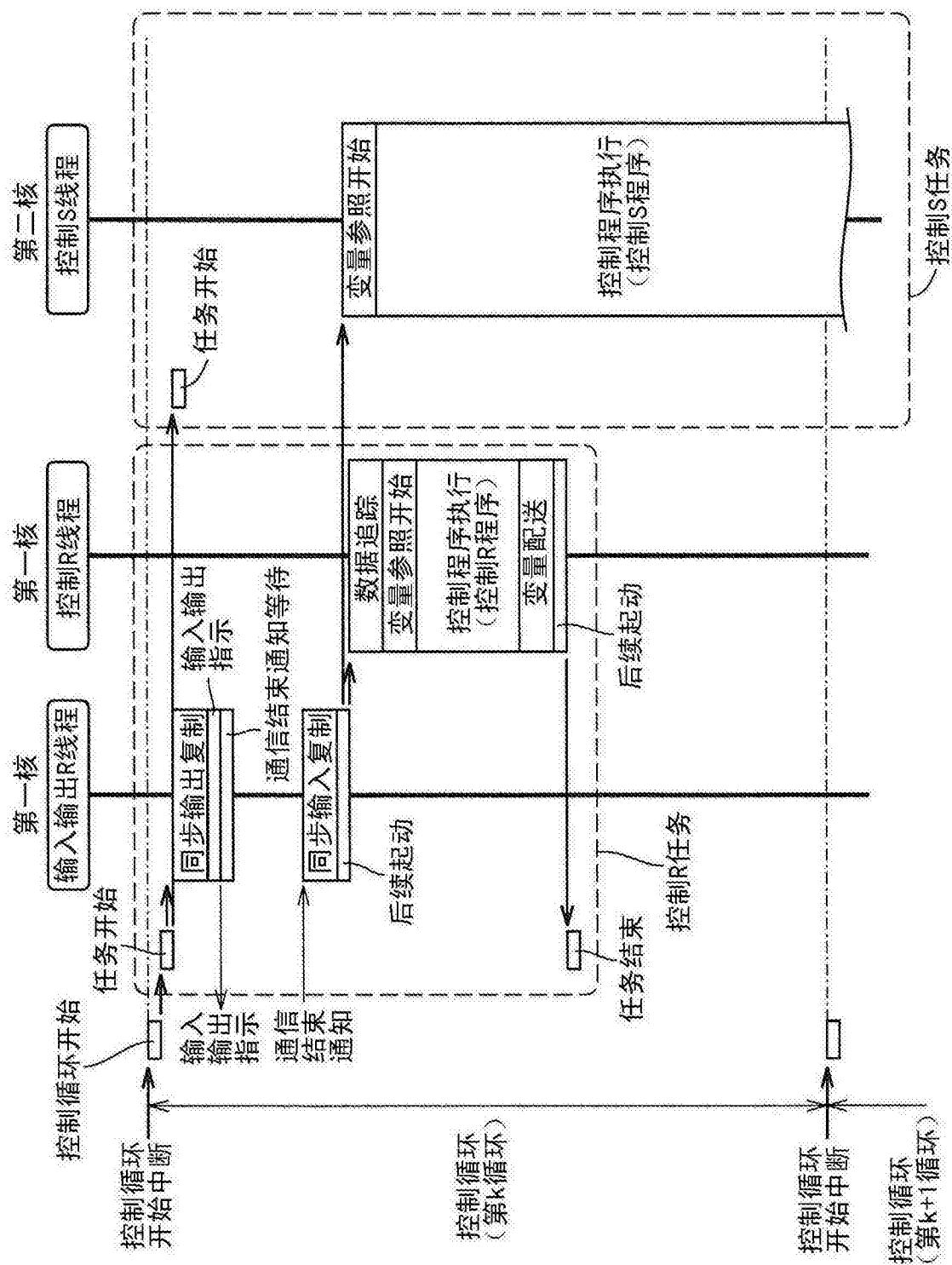


图 33

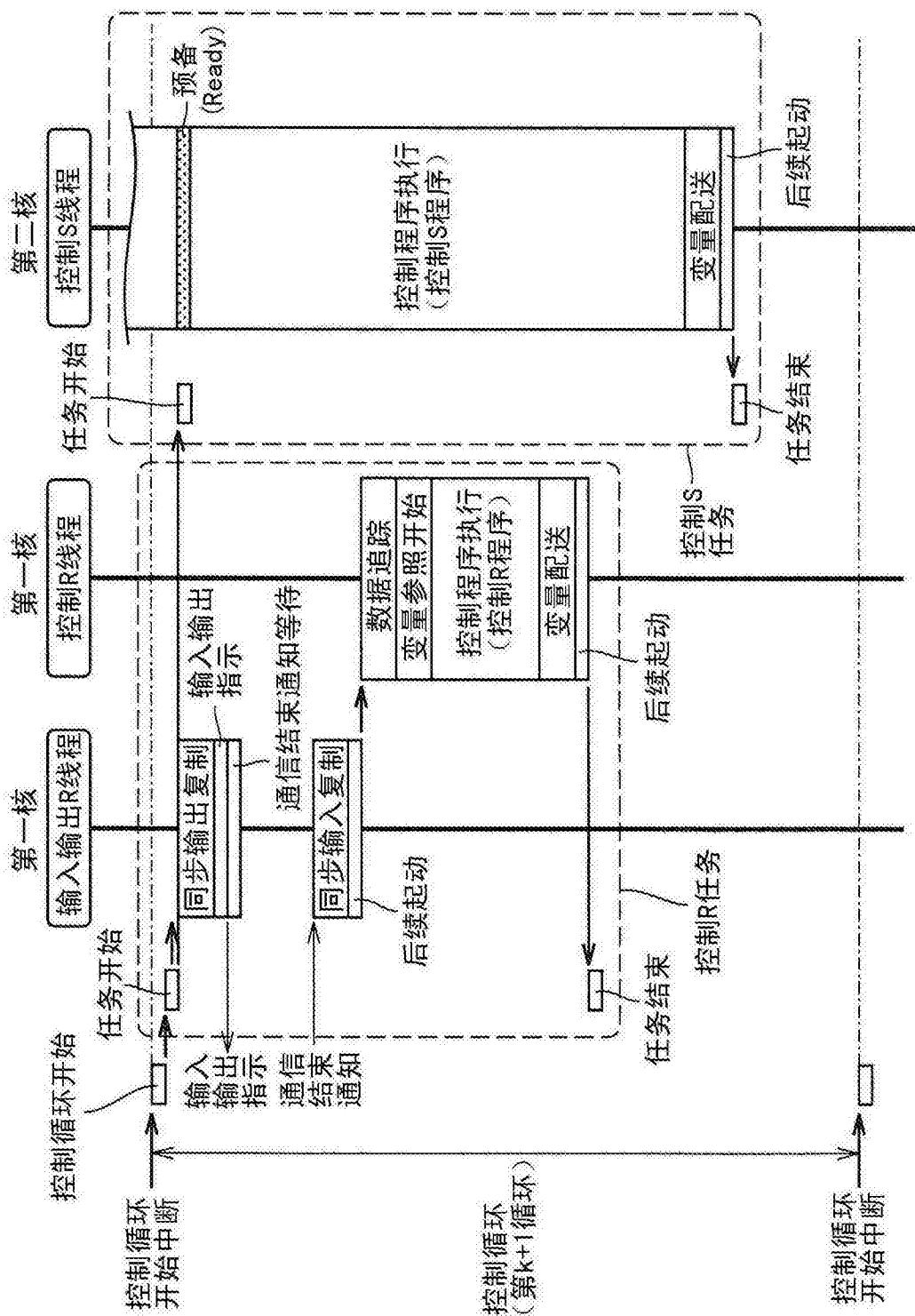


图 34

(1) 控制循环开始条件		控制任务	
任务名称	控制任务	线程名称	控制任务
任务开始条件	执行循环周期(1)	优先级	4
任务开始处理	任务开始过程、优先级(2)	开始条件	控制任务
线程指定	控制任务	过程指定	数据追踪过程 控制程序执行过程
任务结束条件	控制任务		控制输出复制过程 输出结束指示过程
任务结束处理	任务结束过程、优先级(3)		输入通信过程 输入复制过程
循环超过容许次数	1		后续启动过程 启动后等待过程
		控制程序指定	控制任务

图 35

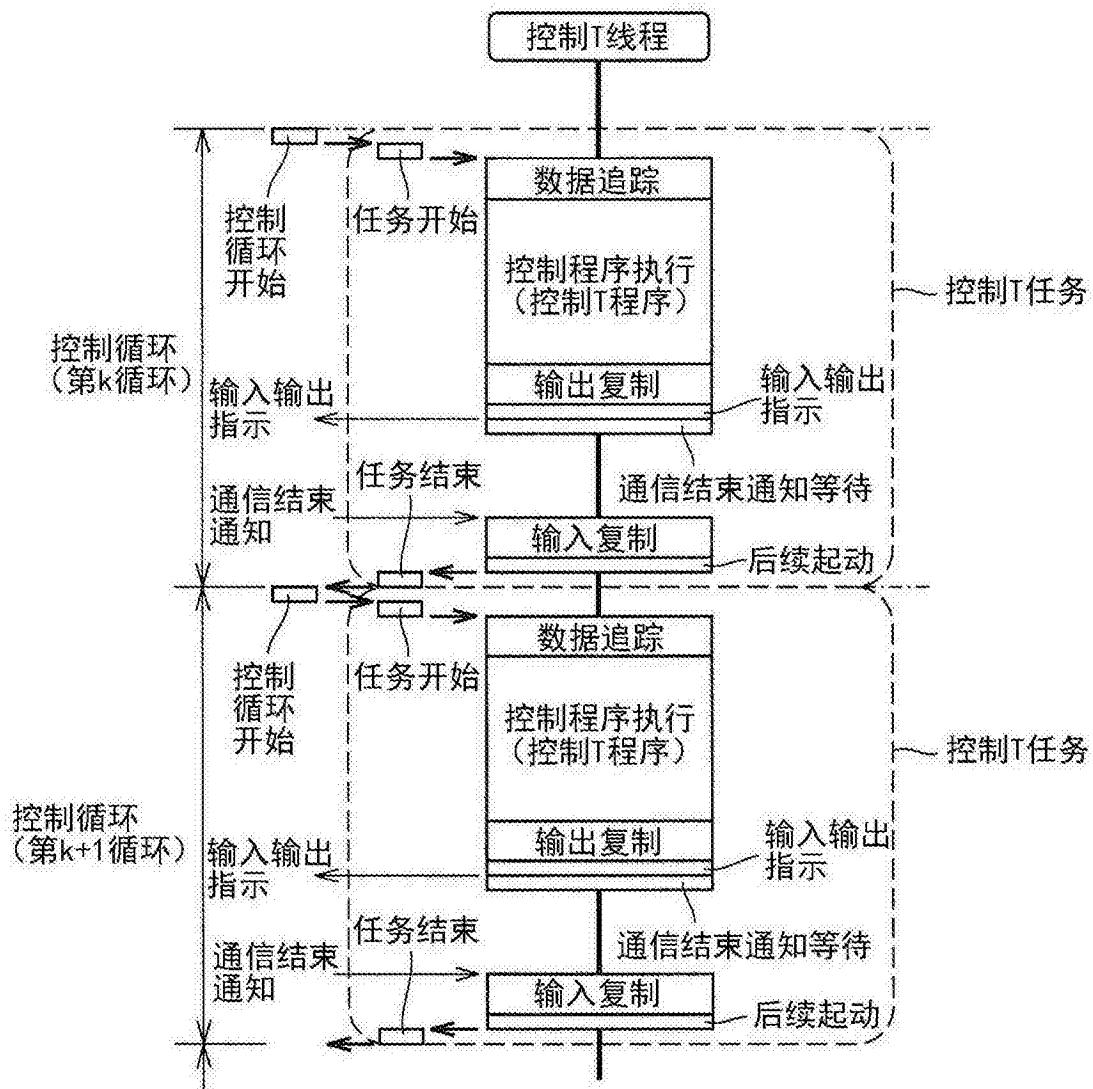


图 36

(1) 控制循环开始条件		控制U任务	
优先级	1	线程名称	控制U线程
(2) 任务名称	控制U任务	优先级	4
任务开始条件	执行循环周期(1)	开始条件	控制U任务
任务开始处理	任务开始过程、优先级(2)	过程指定	输入指示过程 通信结束通知过程 输入结束通知过程 输入数据复制过程 输出数据复制过程 输出结束通知过程 通信环计时过程 后续起动过程
线程指定	控制U线程		
任务结束条件	控制U线程		
任务结束处理	任务结束过程、优先级(3)		
循环超过容许次数	1	控制程序指定	控制U程序

图 37

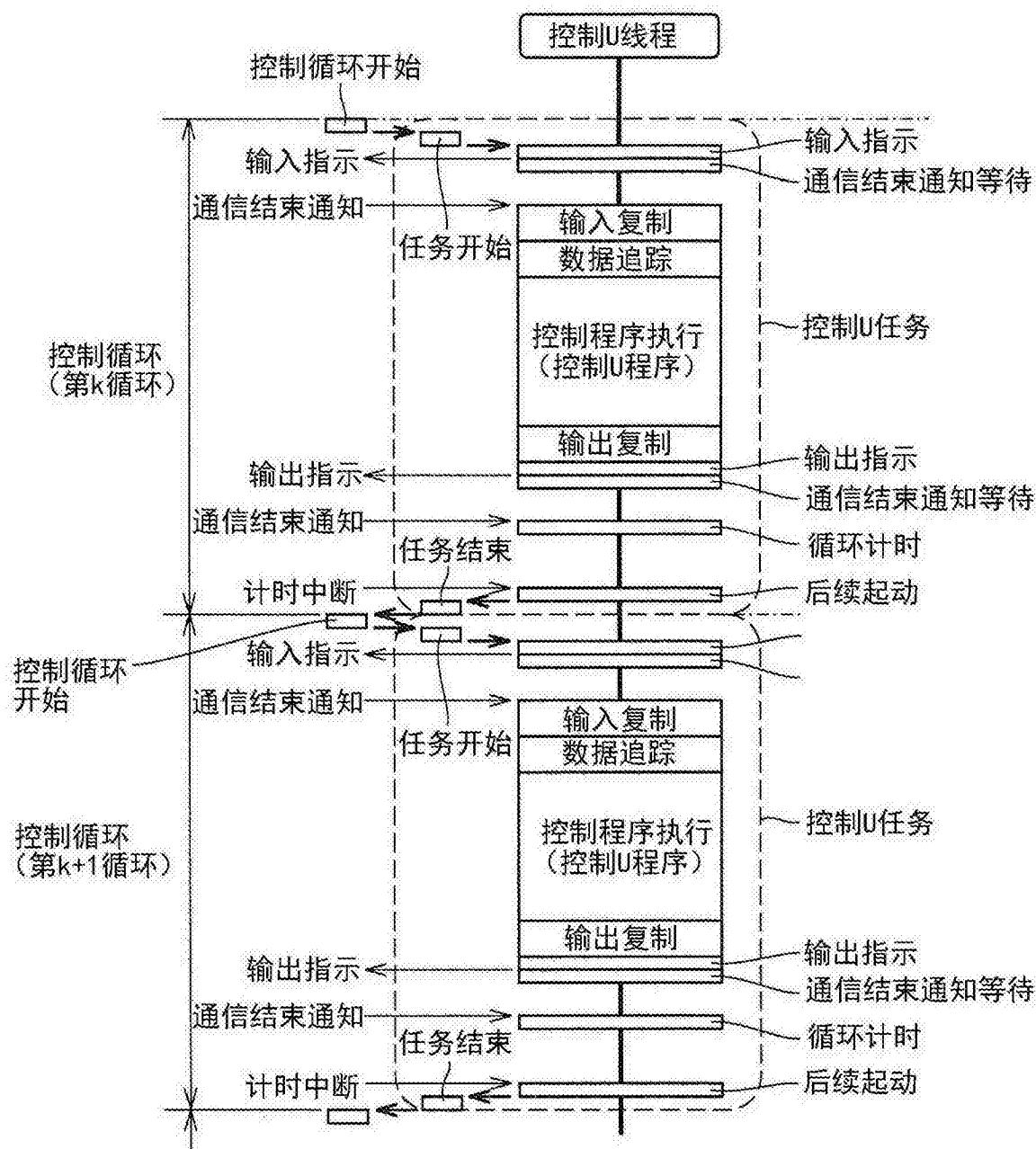


图 38

对线程的设定项目的说明(1/2)

设定项目	说明
线程名称	指定线程的名称。
优先级	指定线程的执行优先级(一般指定为4以上)。以整数指定。值越小优先级越高。执行等待队列中的优先级最高的线程。
开始条件	<p>指定线程的开始条件。能够如下指定。          “执行循环周期(n)”：在第n次执行循环中开始执行任务。          “控制循环开始过程”：成为“控制循环开始过程”的后续起动的对象。          “ssss”：          ssss是表示线程的名称的字符串。成为所指定的线程的后续起动对象。          “事件(eeee)”：          eeee是表示事件的名称的字符串。事件是指特定的中断等。          若所指定的事件发生，则起动线程。</p>
过程指定	指定要执行的过程的名称。能够指定多个过程。以列举的顺序执行各过程。

图 39A

## 对线程的设定项目的说明 (2/2)

设定项目	说明
控制程序指定	指定要执行的用户程序的名称或保留字“动态运算程序”。能够指定多个用户程序的名称。若指定了多个用户程序的名称，则以例举的顺序执行多个用户程序。若指定了保留字“动态运算程序”，则为其他线程的用户程序名称和保留字“动态运算程序”这双方。
动态线程指定	指定如下线程的名称，该线程是指，在设定项目“控制程序指定”中指定有用户程序并且在所指定的用户程序中包含动态指令的情况下，在执行所指定的用户程序之后，在该指定的项目中没有指定的线程情况下，在执行设定项目“控制程序指定”中指定有用户程序之后，接着在本线程中执行动态运算程序。
循环超过容许次数	以整数值来指定循环超过容许次数。“0”不容许发生周期超过而在发生一次周期超过时进行错误处理。
执行核指定	在用多核芯片执行的情况下，指定用于执行线程的核。

图 39B

名称	功能
控制循环开始过程	<ul style="list-style-type: none"> <li>• 针对各线程，执行以下处理。</li> <li>• 对线程的执行循环数进行计数。“执行循环周期 (n) + 循环超过容许次数”并且线程结束标识为关 (OFF) 时，进行执行循环超过的错误处理。</li> <li>• 在计数之后执行循环数超过了执行循环周期 (n) 并且线程结束标识为开 (ON) 时，执行以下处理。           <ul style="list-style-type: none"> <li>• 将线程的执行循环数设定为“1”。</li> <li>• 将线程结束标识设定为关 (OFF)。</li> <li>• 使用于计测线程执行对象时间的计时器开始计时。</li> <li>• 使用于作为后续起动对象的线程。</li> <li>• 在计数之后执行循环数超过了执行循环周期 (n) 并且线程结束标识为关 (OFF) 时，起动“在作为后续起动对象的线程已被起动或者处于事件（通信结束通知等）等待的暂停状态时，不执行任何处理”。</li> </ul> </li> </ul>
设定项目	对设定项目的说明
控制循环开始条件	<p>“控制循环开始中断”； “在控制循环开始中断出现时执行。 (若优先级为1，则立即执行开始)”</p> <p>“ssss”； ssss 是表示程序的名称的结束处理的字符串。</p> <p>成为所指定的线程的结束处理的后续起动对象。</p>
优先级	初始值为1。能够变更。

图 40A

名称	功能
线程结束过程	<ul style="list-style-type: none"><li>•将线程结束标识设为开(ON)。</li><li>•对线程的执行次数进行计数。</li><li>•对线程的执行时间进行计测。</li></ul>

后续启动过程”以后与图11A及图11B相同。

图 40B

(1)	控制循环开始条件	控制循环开始中断
	优先级	1
(2)	线程名称	输入输出V线程
	优先级	2
	开始条件	执行循环周期(1) 控制循环开始
	过程指定	同步输出复制过程 输入输出指示过程 通信结束通知等待过程 同步输入复制过程 后续起动过程
	循环超过容许次数	0
(3)	线程名称	控制V线程
	优先级	3
	开始条件	执行循环周期(1) 输入输出V线程
	过程指定	数据追踪过程 变量参照开始过程 控制程序执行过程 变量配送过程 后续起动过程
	控制程序指定	控制A程序
	循环超过容许次数	1
(4)	线程名称	控制W线程
	优先级	4
	开始条件	控制V线程
	过程指定	变量参照开始过程 控制程序执行过程 变量配送过程 后续起动过程
	控制程序指定	控制B程序
	循环超过容许次数	1

图 41

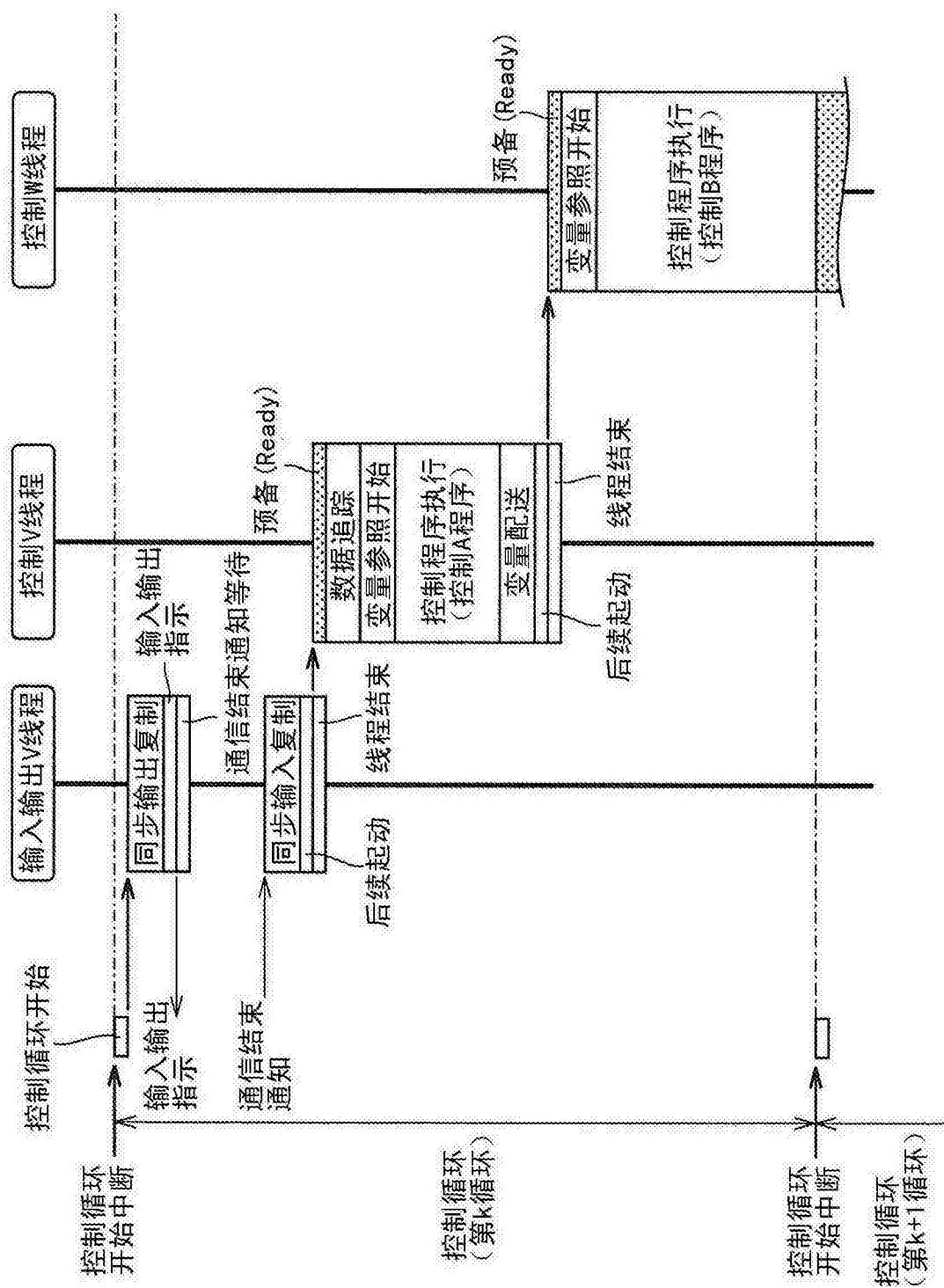


图 42

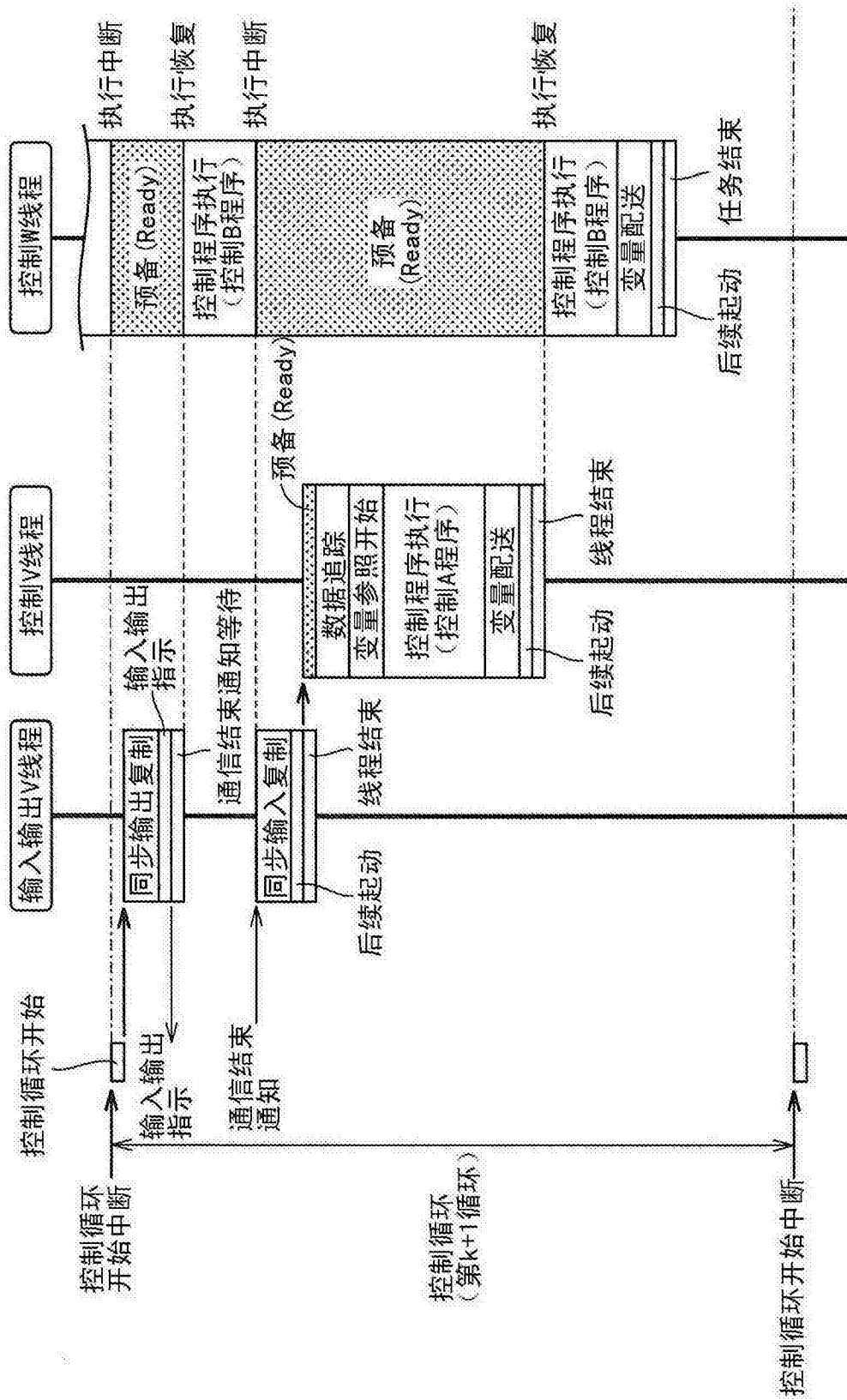


图 43

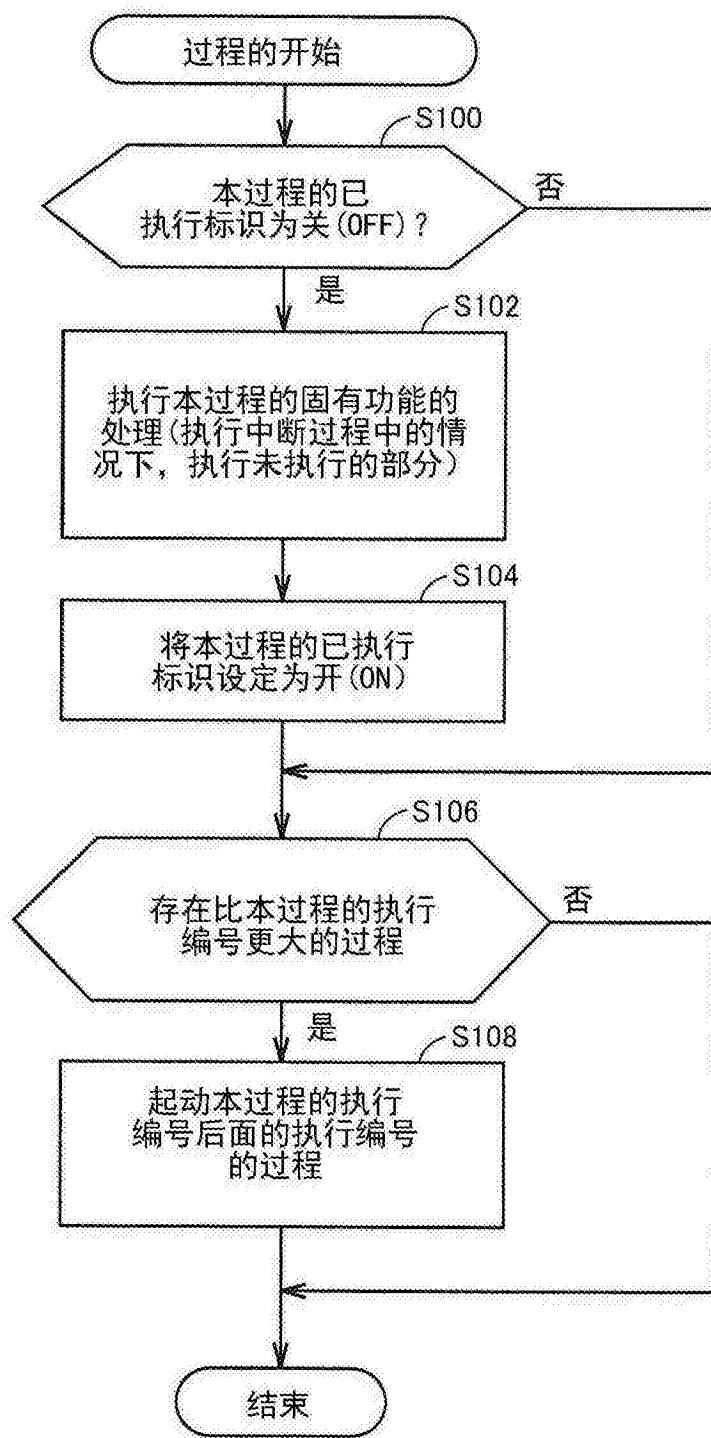


图 44

名称	功能
<b>控制循环开始过程</b> <ul style="list-style-type: none"> <li>• 针对各过程，执行以下处理。           <ul style="list-style-type: none"> <li>• 对过程的执行循环数进行计数。</li> <li>• 在计数之后执行循环数超过了执行循环周期 (n) 并且已执行标识为关 (OFF) 时，进行执行循环超过的错误处理。</li> <li>• 在计数之后执行循环数超过了执行循环周期 (n) 并且已执行标识为开 (ON) 时，执行以下处理。               <ul style="list-style-type: none"> <li>• 将线程的执行循环数设定为 “1”。</li> <li>• 将已执行标识设定为关 (OFF)。</li> </ul> </li> <li>• 起动执行编号最小的过程</li> </ul> </li> </ul>	
设定项目	对设定项目的说明
<b>控制循环开始条件</b>	<ul style="list-style-type: none"> <li>• “控制循环开始中断”：           <ul style="list-style-type: none"> <li>• 在控制循环开始中断出现时起动。(若优先级为初始值的1，则立即执行开始)。</li> </ul> </li> <li>• “循环”：           <ul style="list-style-type: none"> <li>• 在执行编号最大的过程执行结束时，起动。</li> </ul> </li> </ul>

图 45

名称	功能
输出复制过程	将预先指定的输出数据从输出同步缓存复制到发送缓存中。
同步输出复制过程	在执行循环同步模式的情况下，仅限于过程的执行循环数为“1”的情况，将预先指定的输出数据从输出同步缓存复制到发送缓存中。
输入输出指示过程	向通信电路指示将预先指定的输出数据及输入数据作为对象的输出及输入。
输出指示过程	向通信电路指示将预先指定的输出数据作为对象的输出。
输入指示过程	向通信电路指示将预先指定的输入数据作为对象的输入。
通信结束通知等待过程	等待通信结束通知的中断。
输入复制过程	将预先指定的输入数据从接收缓存复制到输入同步缓存中。
同步输入复制过程	在执行循环同步模式的情况下，仅限于过程的执行循环数为“1”的情况，将预先指定的输入数据从接收缓存复制到输入同步缓存中。

图 46A

名称	功能
变量参照开始过程	使由参照侧控制程序参照的变量同步缓存不能被改写。
控制程序执行过程	使控制程序开始执行。
变量配送过程	将参照侧控制程序所具有的全局变量的内容复制到参照侧控制程序的同步缓存中。
循环计时过程	读取从控制循环开始起的经过时间；计算出到控制循环时间结束为止的剩余时间，设定处理器内置的计时器以在所计算出的剩余时间经过时使计时中断出现，进行设定以在计时中断出现时执行控制循环开始过程。
数据追迹过程	将预先指定的变量的值记录到日志文件中。

图 46B

控制循环开始条件	控制循环开始中断
----------	----------

(2)	过程	执行编号	执行循环周期	对象控制程序
	同步输出复制过程	1	1	
	输入输出指示过程	2	1	
	通信结束通知等待过程	3	1	
	同步输入复制过程	4	1	
	数据追溯过程	5	1	
	变量参照开始过程	6	1	
	控制程序执行过程	7	1	控制A程序
	变量配送过程	8	1	
	变量参照开始过程	9	2	
	控制程序执行过程	10	2	控制B程序
	变量配送过程	11	2	

图 47

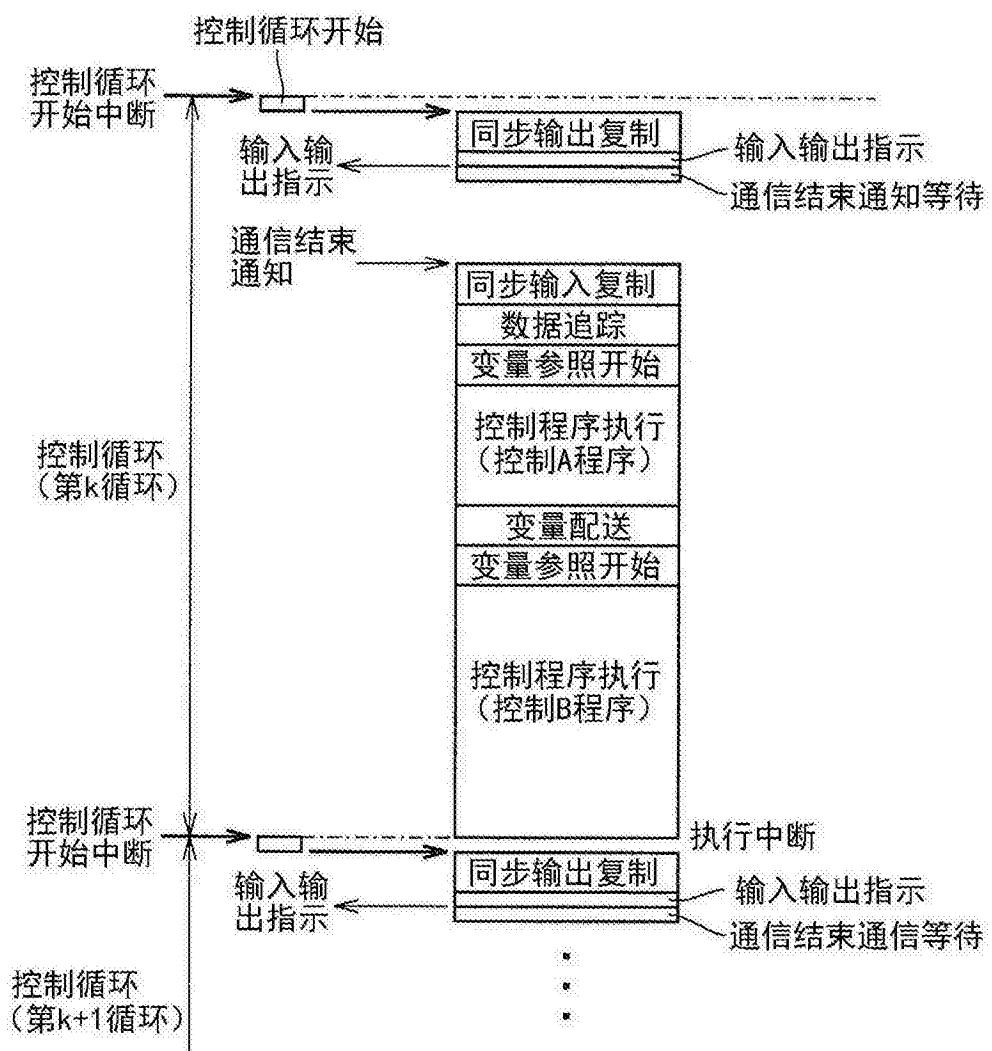


图 48

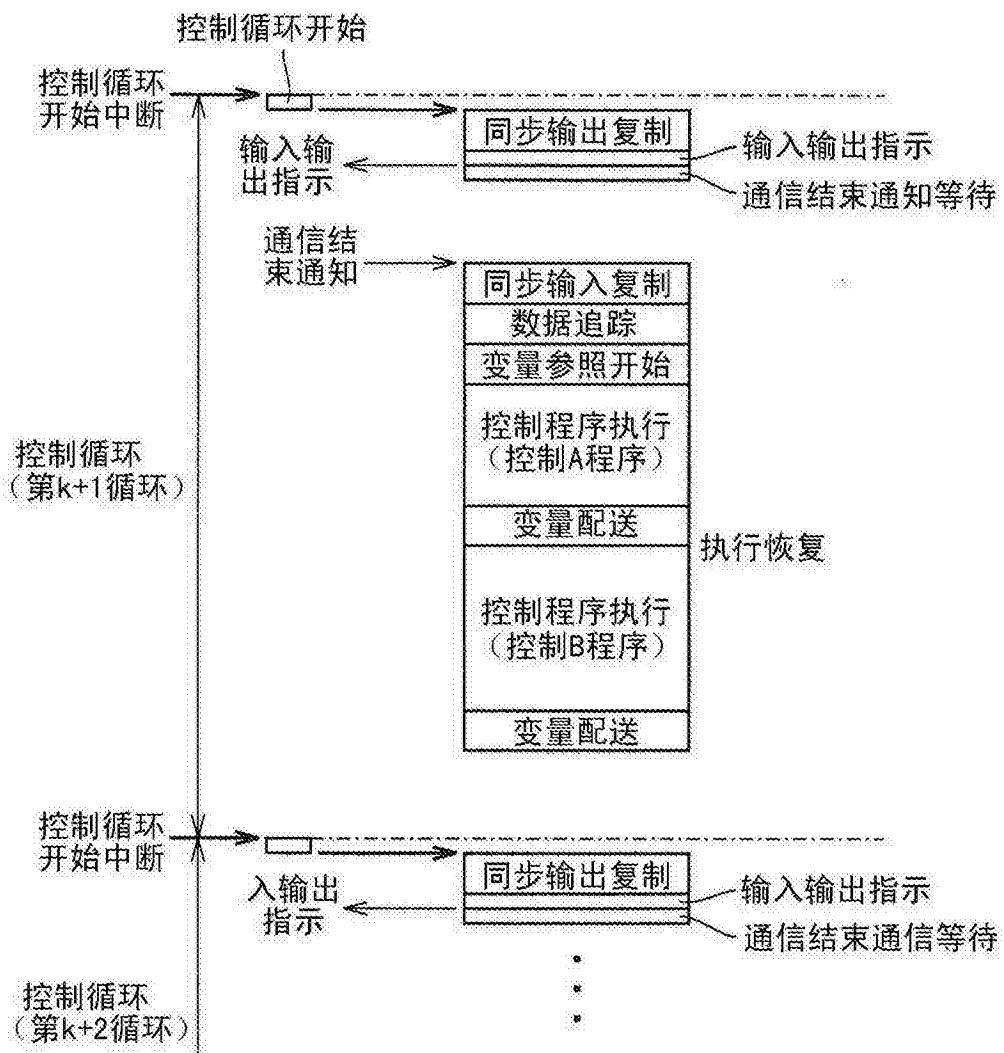


图 49