



(12) 发明专利申请

(10) 申请公布号 CN 114488964 A

(43) 申请公布日 2022. 05. 13

(21) 申请号 202111335335.5

(22) 申请日 2021.11.11

(30) 优先权数据

2020-188984 2020.11.12 JP

(71) 申请人 大隈株式会社

地址 日本爱知县

(72) 发明人 领木正人 江场浩二 片冈章人

浅井孝俊

(74) 专利代理机构 华进联合专利商标代理有限公司

公司 44224

专利代理师 景怀宇

(51) Int.Cl.

G05B 19/418 (2006.01)

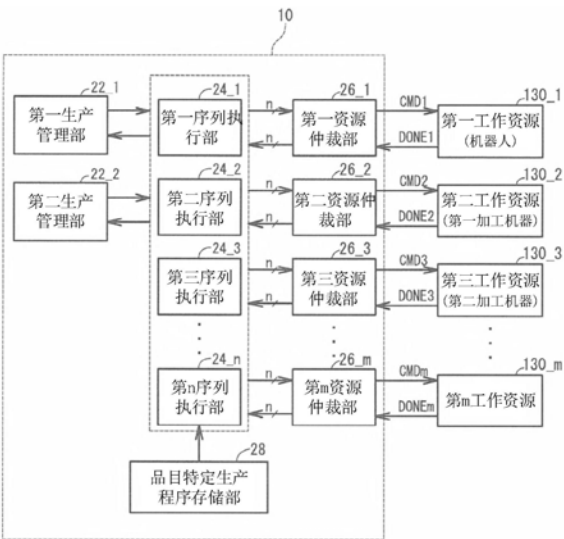
权利要求书2页 说明书19页 附图24页

(54) 发明名称

单元控制器

(57) 摘要

提供了单元控制器(10),该单元控制器(10)用于控制加工单元(100)的操作,该加工单元(100)包括作为工作资源(130)的两个以上加工机器(130_2、130_3)和一个以上机器人(130_1),该单元控制器(10)被配置成基于在对应于在加工单元(100)中生产的一个以上品目而准备的多个生产程序当中的记录有在加工单元(100)中仅生产对应品目的一个物品时要执行的一个以上工序的生产程序,控制加工单元(100)的操作。



1. 单元控制器,用于控制加工单元的操作,所述加工单元包括作为工作资源的两个以上加工机器以及一个以上机器人,

所述单元控制器配置为,基于在对应于在所述加工单元中生产的一个以上品目而准备的多个生产程序当中的记录有在所述加工单元中仅生产对应品目的一个物品时要执行的一个以上工序的生产程序,控制所述加工单元的所述操作。

2. 根据权利要求1所述的单元控制器,包括:

一个以上生产管理部,所述一个以上生产管理部是对应于在所述加工单元中生产的所述一个以上品目而设置的,并且配置为命令将新工件装载到所述加工单元中以生产对应品目的物品;

多个序列执行部,所述多个序列执行部配置为当接收到所述装载的命令时,顺序地执行所述生产程序中记录的所述一个以上工序;以及

多个资源仲裁部,所述多个资源仲裁部是与所述多个工作资源对应地设置的,其中:

所述序列执行部将所述资源仲裁部视为与所述资源仲裁部相关联的工作资源,并向所述资源仲裁部发送用于所述工作资源的命令,并且

所述资源仲裁部仲裁所述多个序列执行部中的每个序列执行部使用与所述资源仲裁部相关联的所述工作资源的时刻。

3. 根据权利要求2所述的单元控制器,其中:

在当前工序的执行完成之前,所述序列执行部向与所述工作资源相关联的所述资源仲裁部输出对下一工序中新需要的工作资源的使用请求。

4. 根据权利要求2或3所述的单元控制器,其中:

所述机器人包括一个以上安装部分,在所述安装部分上安装有手,

所述序列执行部当在下一工序中需要新机器人时,指定与当前工序对应的前一工件通过判断工序,并且在用于执行前一工件的生产工序的另一序列执行部开始执行所述前一工件通过判断工序或者发出对关于所述前一工件通过判断工序的机器人的使用请求之后,发出对关于所述下一工序的所述机器人的使用请求,并且

当在所述下一工序中使用的所述安装部分等同于在再次使用所述机器人的下一资源调用工序中使用的安装部分时,所述前一工件通过判断工序是所述下一资源调用工序,并且,当在所述下一工序中使用的所述安装部分不同于在所述下一资源调用工序中使用的所述安装部分时,或者当不存在所述下一资源调用工序时,所述前一工件通过判断工序是所述下一工序。

5. 根据权利要求2至4中任一项所述的单元控制器,还包括:

复合表存储部,其配置为存储复合表,所述复合表记录有构成所述生产程序的多个工序中的能够并行执行的两个以上工序,以及允许所述工作资源并行执行所述两个以上工序的复合命令,其中:

当从彼此不同的两个以上序列执行部中的每一个序列执行部接收到用于能够并行执行的两个以上工序的执行命令时,所述资源仲裁部基于所述复合表将用于所述两个以上工序的所述执行命令转换成所述复合命令,并将所述复合命令输出到所述工作资源。

6. 根据权利要求2至5中任一项所述的单元控制器,包括:

彼此独立地运行的所述两个以上生产管理部,其中:

所述单元控制器允许所述加工单元并行地生产两个以上品目的物品。

7. 根据权利要求2至6中任一项所述的单元控制器,还包括:

重试判断部,其配置为监视与工序执行相关的异常是否存在,并且当检测到所述异常时,返回到检测到所述异常的所述工序的开始或者返回到在所述工序的所述开始之前的设定的重试点,以向所述序列执行部发出用于重新执行包括所述工序的一个以上工序的重试命令。

8. 根据权利要求7所述的单元控制器,其中:

当为被所述重试命令指派为要重新执行的所述工序定义了替代工序时,所述序列执行部执行所述替代工序。

9. 根据权利要求7或8所述的单元控制器,其中:

所述重试判断部至少在当从所述工序的执行开始起所经过的时间超过预定的允许执行时间时或者在当所述工序的所述执行期间从所述工作资源输出错误时,判断发生了异常。

10. 根据权利要求7或8所述的单元控制器,还包括:

工件跳过执行部,其配置为当从所述重试判断部向一个序列执行部发出的针对一个工序的重试命令的连续发出次数超过预定的允许次数时,执行弃置当前工件的工件跳过工序。

单元控制器

[0001] 相关申请的交叉引用

[0002] 本申请要求于2020年11月12日提交的第2020-188984号日本专利申请的优先权，其包括说明书、权利要求书、附图和摘要的全部内容通过引用结合于此。

技术领域

[0003] 本公开公开了单元控制器，该单元控制器控制加工单元的操作，该加工单元包括两个以上加工机器以及一个以上例如机器人等的输送装置。

背景技术

[0004] 在相关技术中，已知包括作为工作资源的两个以上加工机器以及一个以上机器人的加工单元。可以使用这种加工单元并行地生产多个工件。加工单元由单元控制器 (cell controller) 控制。单元控制器根据要生产的物品的品目，向构成加工单元的多个工作资源输出命令。

[0005] 这里，根据相关技术的单元控制器通常包括市售的通用PLC。在这种情况下，已经使用PLC的编程语言开发了满足工厂生产线的加工单元的需求规范的PLC程序，以实现期望的生产工序。例如，根据相关技术的单元控制器指示每个工作资源根据加工进度执行用于该工作资源的程序 (例如，在加工机器的情况下的NC程序)。在根据相关技术的单元控制器中，需要创建PLC程序，例如描述了单元控制器的行为的阶梯程序 (ladder program)。基于例如要加工的品目的类型、要使用的工作资源的类型以及在加工过程中工件的位置等的信息，确定向任何工作资源输出任何程序的时刻。

[0006] 然而，在PLC程序中，要由单元控制器 (最终是PLC) 执行的功能被描述为来自所有输入信号的组合的操作序列。因此，加工单元的设备管理员或操作员难以自己创建或修改PLC程序，通常由具有专家知识的系统集成商创建和修改PLC程序。

[0007] 因此，在使用PLC的单元控制器的情况下，存在操作成本高的问题，因为需要外包给系统集成商。此外，存在改变生产工序需要时间的问题，因为每次改变生产过程时都需要外包给系统集成商。特别地，近来对多品种小批量 (high-mix low-volume) 生产的需求一直很高，并且生产工序经常改变。有一个很大的问题是需要外包，并且对于以这种方式频繁发生的生产工序的每次改变都需要时间。

[0008] JPH06-277990A公开了一种单元控制器，该单元控制器在使用加工单元生产物品时自动生成生产计划。然而，在JPH06-277990A中没有详细说明用于生成生产计划的具体步骤。特别地，根据JPH06-277990A，不清楚在并行生产多个工件时如何消除工件之间的工作资源争用。

[0009] 此外，JPH2004-185228A公开了一种系统，该系统允许每个工作资源预先存储用于执行一个以上工作部的操作程序，并且如果需要，允许单元控制器向每个工作资源输出生产命令，该生产命令包括分配有执行序列的一组工作部。在这种情况下，当从单元控制器输出生产命令时，工作资源按照由生产命令指令的顺序执行程序以执行工作。然而，根据

JPH2004-185228A的技术,需要在工作资源上存储特定的程序,这是麻烦的。此外,根据JPH2004-185228A,也不清楚在并行生产多个工件时如何消除工件之间的工作资源争用。

[0010] 因此,本公开公开了一种单元控制器,在该单元控制器中,即使由不熟悉加工单元的详细控制规范和PLC语言的操作员也能够容易地创建和编辑避免多个工件之间的工作资源争用的生产程序,并且由此能够进一步提高物品的生产效率。

发明内容

[0011] 本公开提供了单元控制器,所述单元控制器用于控制加工单元的操作,所述加工单元括作为工作资源的两个以上加工机器和一个以上机器人,所述单元控制器配置为基于在对应于在所述加工单元中生产的一个以上品目而准备的多个生产程序当中的记录有在所述加工单元中仅生产对应品目的一个物品时要执行的一个以上工序的生产程序,控制所述加工单元的操作。

[0012] 在这种情况下,所述单元控制器可以包括:一个以上生产管理部,所述一个以上生产管理部是对应于在所述加工单元中生产的所述一个以上品目而设置的,并且配置为命令将新工件装载到所述加工单元中以生产对应品目的物品;多个序列执行部,所述多个序列执行部配置为当接收到所述装载的命令时,顺序地执行所述生产程序中记录的所述一个以上工序;以及多个资源仲裁部,所述多个资源仲裁部是与所述多个工作资源对应地设置的,其中:所述序列执行部可以将所述资源仲裁部视为与所述资源仲裁部相关联的工作资源,并向所述资源仲裁部发送用于所述工作资源的命令,并且所述资源仲裁部可以仲裁所述多个序列执行部中的每个序列执行部使用与所述资源仲裁部相关联的所述工作资源的时刻。

[0013] 在当前工序的执行完成之前,所述序列执行部可以向与所述工作资源相关联的所述资源仲裁部输出对下一工序中新需要的工作资源的使用请求。

[0014] 所述机器人可以包括一个以上安装部分,在所述安装部分上安装有手,所述序列执行部可以当在下一工序中需要新机器人时,指定与当前工序对应的前一工件通过判断工序,并且在用于执行前一工件的生产工序的另一序列执行部开始执行所述前一工件通过判断工序或者发出对关于所述前一工件通过判断工序的机器人的使用请求之后,发出对关于所述下一工序的所述机器人的使用请求,并且当在所述下一工序中使用的所述安装部分等同于在再次使用所述机器人的下一资源调用工序中使用的安装部分时,所述前一工件通过判断工序可以是所述下一资源调用工序,并且,当在所述下一工序中使用的所述安装部分不同于在所述下一资源调用工序中使用的所述安装部分时,或者当不存在所述下一资源调用工序时,所述前一工件通过判断工序可以是所述下一工序。

[0015] 所述单元控制器还可以包括:复合表存储部,所述复合表存储部配置为存储复合表,所述复合表记录有构成所述生产程序的多个工序中的能够并行执行的两个以上工序、以及允许所述工作资源并行执行所述两个以上工序的复合命令,其中:当从彼此不同的两个以上序列执行部中的每一个序列执行部接收到用于能够并行执行的两个以上工序的执行命令时,所述资源仲裁部可以基于所述复合表将用于所述两个以上工序的所述执行命令转换成所述复合命令,并将所述复合命令输出到所述工作资源。

[0016] 所述单元控制器可以包括彼此独立地运行的所述两个以上生产管理部,其中:所述单元控制器可以允许所述加工单元并行地生产所述两个以上品目的物品。

[0017] 所述单元控制器还可以包括重试判断部,所述重试判断部配置为监视与工序执行相关的异常是否存在,并且当检测到所述异常时,返回到检测到所述异常的所述工序的开始或者返回到在所述工序的所述开始之前的设定的重试点,以向所述序列执行部发出用于重新执行包括所述工序的一个以上工序的重试命令。

[0018] 在这种情况下,当为被所述重试命令指派为要重新执行的所述工序定义了替代工序时,所述序列执行部可以执行所述替代工序。

[0019] 所述重试判断部可以在当从所述工序的执行开始起所经过的时间超过预定的允许执行时间时或者在当所述工序的所述执行期间从所述工作资源输出错误时,判断发生了异常。

[0020] 所述单元控制器可以包括工件跳过执行部,所述工件跳过执行部配置为当从所述重试判断部向所述一个序列执行部发出的针对所述一个工序的所述重试命令的连续发出次数超过预定的允许次数时,执行弃置当前工件的工件跳过工序。

[0021] 根据本公开的所述单元控制器,操作员可以创建记录有在加工单元中仅生产对应品目的一个物品时要执行的一个以上工序的生产程序。这样的生产程序能够仅通过操作员的知识来创建。结果,不需要请求外部专家来创建程序。结果,能够减少创建和编辑生产程序所需的时间,从而能够进一步提高物品的生产效率。另外,当单元控制器包括资源仲裁部时,即使当操作员没有创建复杂的生产程序时,也能够避免工件之间对工作资源的争用。

附图说明

[0022] 本公开的实施例将基于以下图进行描述,其中:

[0023] 图1是示出单元控制器以及要由单元控制器控制的加工单元的图;

[0024] 图2是单元控制器的功能框图;

[0025] 图3是提取了图2中的主要部件的图;

[0026] 图4是示出用于生产第一品目的物品的生产程序的示例的图;

[0027] 图5是示出用于生产第二品目的物品的生产程序的示例的图;

[0028] 图6是示出第一生产管理部的处理流程的流程图;

[0029] 图7是示出序列执行部的处理流程的前半部分的流程图;

[0030] 图8是示出序列执行部的处理流程的后半部分的流程图;

[0031] 图9是示出用于图7和图8中的机器人的处理的流程的流程图;

[0032] 图10是图示出前一工件通过判断工序计数的图;

[0033] 图11是示出资源仲裁部的处理流程的流程图;

[0034] 图12是示出图11中的选择处理的流程的流程图;

[0035] 图13是示出在另一具体示例中的序列执行部的处理流程的前半部分的流程图;

[0036] 图14是示出在另一具体示例中的序列执行部的处理流程的后半部分的流程图;

[0037] 图15是示出用于图13和图14中的机器人的处理的流程的流程图;

[0038] 图16是另一具体示例的单元控制器的功能框图;

[0039] 图17是示出复合表的示例的图;

[0040] 图18是示出包括复合操作判断部的资源仲裁部的处理流程的前半部分的流程图;

[0041] 图19是示出包括复合操作判断部的资源仲裁部的处理流程的后半部分的流程图;

- [0042] 图20是示出单元控制器的一仿真结果的图；
- [0043] 图21是示出单元控制器的另一仿真结果的图；
- [0044] 图22是示出单元控制器的再一仿真结果的图；
- [0045] 图23是示出另一具体示例的单元控制器的部分配置的图；
- [0046] 图24是示出添加了重试点的生产程序的示例的图；
- [0047] 图25是示出添加了替代工序的生产程序的示例的图；以及
- [0048] 图26是示出另一具体示例的单元控制器的部分配置的图。
- [0049] 附图标记说明
- [0050] 10:单元控制器;12:处理器;14:存储器;16:通信I/F;18:输入装置;20:输出装置;22:生产管理部;24:序列执行部;26:资源仲裁部;28:品目特定生产程序存储部;30:工序计数器;31:复合操作判断部;32:复合表存储部;34:重试判断部;36:执行时间收集部;38:工件跳过执行部;100:加工单元;102:第一加工机器;104:第二加工机器;106:主轴;108:副主轴;110:机器人;111:手;112:手存储器;114:钳存储器;116:工件清洁器;118:工件测量装置;120:临时放置台;122:搬入托盘;124:搬出托盘;130:工作资源。

具体实施方式

[0051] 下面将参照附图描述单元控制器10的配置。图1是示出单元控制器10以及要由单元控制器10控制的加工单元100的图。进一步地,图2是单元控制器10的功能框图,图3是提取了图2中的主要部件的图。

[0052] 首先,将对要控制的加工单元100进行描述。加工单元100包括两个以上加工机器102、104,并且对工件执行各种类型的加工。本示例的加工单元100包括对金属材料执行加工的两个金属加工机器。第一加工机器102是对金属工件执行切割的切割机器;具体地为,车床(lathe)。第一加工机器102包括主轴106和副主轴108,并且能够自动地执行将工件从主轴106递送到副主轴108。因此,例如,可以分别使用主轴106和副主轴108连续地并且自动地加工一个工件的前表面和后表面。第二加工机器104具有与第一加工机器102相同的配置,并且包括主轴106和副主轴108。在加工机器102、104中的每一者的夹盘上附接有用以保持工件的钳,但该钳需要根据工件的形状而变更。因此,加工单元100设置有配置成储存多种类型的钳的钳存储器114。

[0053] 这里,第一加工机器102和第二加工机器104可以并行地或串行地使用。相应地,可以是由第一加工机器102加工后的工件未装载到第二加工机器104中的情况,也可以是由第二加工机器104加工后的工件未装载到第一加工机器102中的情况。例如,用于第一品目的工件可以由第一加工机器102加工,而用于第二品目的工件可以由第二加工机器104加工。此外,作为另一模式,可以将由第一加工机器102加工后的工件装载到第二加工机器104中,并且可以对其进行额外的加工。例如,用于第一品目的工件可以由第一加工机器102进行粗加工,并且可以由第二加工机器104进行精加工。

[0054] 加工单元100还设置有机人110。机器人110设置成对例如工件或加工机器的钳进行输送。所示示例中的机器人110是关节式串联机械手(articulated serial manipulator),该关节式串联机械手中多个连杆经由关节耦接。手111a和手111b可互换地附接到机器人110的末端以保持工件。在图1的示例中,机器人110是能够同时安装两只手

111a、111b的双手型机器人。在下面的描述中,当彼此没有区别时,两只手111a、111b仅被称为“手111”而没有下标。在本示例中,使用双手型机器人110,但是机器人110也可以是只能安装一只手111的单手型机器人。

[0055] 手111的类型根据要保持的物品的形状来选择。例如,可以取决于保持加工前的工件还是保持加工后的工件而使用不同类型的手。在这种情况下,在单手型机器人110中,当在加工前的工件被输送之后,加工后的工件被输送时,替换要安装的手111。在如图1中所示的双手型机器人110中,一个机器人110安装有用于加工前的工件的手111a和用于加工后的工件的手111b。在这种情况下,机器人110可以同时保持加工前的工件和加工后的工件,但不能同时保持两个加工前的工件或两个加工后的工件。安装到机器人110的手111保管在手存储器112中。

[0056] 加工单元100还设置有工件清洁器116和工件测量装置118。工件清洁器116是用于清洁加工后的工件的装置,并且例如是用于喷射例如空气或清洁液等的清洁介质的装置。可以将两个工件同时装载到工件清洁器116中。工件测量装置118测量加工前或加工后的工件的形状,工件测量装置118包括例如激光测量仪器。激光测量仪器不包括用于放置或保持工件的机构,而由工件测量装置118测量处于被保持在机器人110中的状态下的工件。

[0057] 在加工单元100中,作为工件的等待场所,还准备有搬入托盘122、搬出托盘124和临时放置台120。搬入托盘122是用于未加工工件的等待场所,搬出托盘124是用于已经生产并计划要运出的工件的等待场所。临时放置台120是在生产工序中工件暂时等待的场所。可以将两个工件同时装载到临时放置台120中。

[0058] 单元控制器10控制这样的加工单元100的操作。更具体地,单元控制器10向构成加工单元100的设备(例如,加工机器或机器人110,以及工件测量装置118)给出命令,以便仅产生所需品目的所需数量的物品。在下面的描述中,“工作资源130”表示在加工单元100的多个部件当中的从单元控制器10接收命令的部件。在图1的示例中,假设在多个部件当中,第一加工机器102、第二加工机器104和机器人110各自充当工作资源130。

[0059] 单元控制器10是物理上包括处理器12、存储器14、通信接口(在下文中称为“通信I/F16”)、输入装置18和输出装置20的计算机。“计算机”还包括微控制器,在该微控制器中,在一个集成电路中集成有计算机系统。另外,处理器12包括在广义上表示处理器的通用处理器(例如,CPU:Central Processing Unit,中央处理单元)或专用处理器(例如,GPU:Graphics Processing Unit,图形处理单元、ASIC:Application Specific Integrated Circuit,专用集成电路、FPGA:Field Programmable Gate Array,现场可编程门阵列、或者可编程逻辑器件)。将在下面描述的处理器12的操作不仅可以由一个处理器12执行,也可以由存在于物理上彼此远离的位置处的多个处理器12的协作执行。类似地,存储器14物理上不仅可以包括一个存储器,也可以包括存在于物理上彼此远离的位置处的多个存储器14。进一步地,存储器14可以包括半导体存储器(例如,RAM、ROM或固态驱动器)和磁盘(例如,硬盘驱动器)中的至少一者。

[0060] 通信I/F 16与单元控制器10的外部装置(例如,机器人110或加工机器的控制器(未图示),或另一计算机)交换数据。通信I/F 16可以以无线或有线方式与外部装置通信。输入装置18接收来自操作员的操作指令,并且包括例如键盘、鼠标、触控面板和麦克风。输出装置20向操作员呈现信息,并且包括例如监视器、扬声器和打印机。

[0061] 这样的单元控制器10具有如图2中所示的功能配置。在描述单元控制器10的功能配置之前,将简要描述单元控制器10。

[0062] 相关技术的单元控制器10在许多情况下包括市售的通用PLC。在这种情况下,为了实现期望的生产工序,已经使用PLC的编程语言根据工厂生产线的加工单元100的需求规范开发了PLC程序。例如,相关技术的单元控制器10指示每个工作资源130根据加工进度执行用于该工作资源130的程序(例如,在加工机器的情况下的NC程序)。在相关技术的单元控制器10中,需要创建PLC程序,例如描述了单元控制器10的行为的阶梯程序。基于例如要加工的品目的类型、要使用的工作资源130的类型以及在加工过程中工件的位置等的信息,确定向任何工作资源130输出任何程序的时刻。然而,这样的PLC程序将要由单元控制器10(以及因此PLC)执行的功能描述为来自所有输入信号的组合的操作序列。因此,加工单元100的设备管理员或操作员难以自己创建或修改PLC程序,通常由具有专家知识的系统集成商创建和修改PLC程序。

[0063] 因此,在使用PLC的单元控制器10的情况下,存在操作成本高的问题,因为需要外包给系统集成商。此外,存在改变生产工序需要时间的问题,因为每次改变生产过程时都需要外包给系统集成商。特别地,近来对多品种小批量(high-mix low-volume)生产的需求一直很高,并且生产工序经常改变。有一个很大的问题是需要外包,并且对于以这种方式频繁发生的生产工序的每次改变都需要时间。

[0064] 本示例的单元控制器10不需要这样的PLC程序,并且基于能够由操作员容易地创建的生产程序来控制加工单元100的操作。生产程序是记录有在加工单元100中仅生产一个物品时要执行的一个以上工序的程序。这样的生产程序是针对要在加工单元100中生产的品目的数量而准备的。

[0065] 图4是示出用于生产第一品目的物品的生产程序的示例的图,图5是示出用于生产第二品目的物品的生产程序的示例的图。在用于图4中所示的第一品目的生产程序中,针对为生产第一品目而执行的多个工序(在所示示例中为15个工序)中的每一个工序,彼此关联地记录工序编号、工序名称、要使用的工作资源130的类型、以及用于操作工作资源130的资源程序的标识信息。这样的生产程序存储在可以由操作员容易地编辑参数的格式(例如,以表格格式或文本格式)的文件中。在图4中,在工作资源130当中,第一加工机器102被写成“NC1”,第二加工机器104被写成“NC2”,机器人110被写成“R1”。进一步地,要由工作资源130读取和执行的程序的名称被作为标识信息记录在资源程序栏中。根据图4的示例,用于第一品目的工件由第一加工机器102切割,并且由机器人110输送。

[0066] 用于第二品目的生产程序也具有相同的配置,针对为了生产第二品目而执行的多个工序(在所示示例中为15个工序)中的每一个工序,彼此关联地记录工序编号、工序名称、要使用的工作资源130的类型、以及用于操作工作资源130的资源程序的标识信息。在图5的示例中,用于第二品目的工件由第二加工机器104切割,并且被输送至机器人110。从图4和图5中可以清楚看出,本示例的生产程序是按顺序记录单个工件的生产工序的程序,即使是由不熟悉用于PLC的编程语言的操作员也能对该生产程序容易地创建和修改。然后,由单元控制器10基于生产程序执行对加工单元100的控制,由此操作员能够快速响应生产工序的改变,并且能够更有效地生产物品。

[0067] 下面将参照图2和图3描述单元控制器10的功能配置。在功能上,如图2中所示,单

元控制器10包括一个以上生产管理部22_1和22_2、多个序列执行部24_1至24_n、多个资源仲裁部26_1至26_m、以及一个品目特定生产程序存储部28。在下面的描述中,当彼此没有区别时,第一生产管理部22_1和第二生产管理部22_2仅被称为“生产管理部22”,而没有“第一”或“第二”的符号和下标“_1”或“_2”。这同样适用于序列执行部和资源仲裁部。

[0068] 品目特定生产程序存储部28存储上述生产程序。对应于在加工单元100中生产的一个以上品目中的每一个品目来准备和存储生产程序。如有需要,操作员可以在特定品目生产程序存储部28中存储新的生产程序。另外,如有需要,操作员可以编辑在特定品目生产程序存储部28中存储的生产程序。

[0069] 生产管理部22是对应于在加工单元100中生产的每个品目而设置的,并且其命令加工单元100装载新的工件,以用于生产对应品目的目的。在本示例中,由于加工单元100生产第一品目和第二品目,因此单元控制器10包括两个生产管理部22;即,第一生产管理部22_1和第二生产管理部22_2。生产管理部22实际上是在计算机上启动的应用。因此,当处理器12执行在存储器14中存储的用于生产管理部22的应用程序时,计算机充当生产管理部22。在本实施例中,由于提供了两个生产管理部22,即,第一生产管理部22_1和第二生产管理部22_2,因此用于两个生产管理部22的应用程序彼此独立地启动。

[0070] 生产管理部22从外部计算机或操作员接收生产命令。生产命令包括关于要生产的品目的标识信息和要生产的批次数量。在接收到这样的生产命令时,生产管理部22命令序列执行部24(将在下面对其描述)执行对应品目的生产程序。通过这样的命令,新的工件被装载到加工单元100上。因此,由生产管理部22向序列执行部24做出的生产程序的执行命令充当将新工件装载到加工单元100的指令。重复对序列执行部24的生产程序的执行命令,直到被加工工件的数量达到指派的批次数量为止。进一步地,生产管理部22还管理当前装载到加工单元100中的工件的数量和被加工工件的数量。

[0071] 序列执行部24基于从生产管理部22发出的命令,顺序地执行在从生产管理部22指定的生产程序中记录的一个以上工序。与生产管理部22中一样,序列执行部24也是要在计算机上启动的应用。因此,当处理器12执行在存储器14中存储的用于序列执行部24的应用程序时,计算机充当序列执行部24。在本示例中,由于设置了多个序列执行部24,因此用于多个序列执行部24的应用彼此独立地启动。

[0072] 该多个序列执行部24彼此独立地运行。另外,多个序列执行部24中的每个序列执行部24将资源仲裁部26(将在以下对其进行描述)视作与该资源仲裁部26相关的工作资源130,并向资源仲裁部26发送有关于工作资源130的命令。具体地,如图3中所示,序列执行部24向资源仲裁部26发送工作资源130的使用请求REQ_{nm}和要向工作资源130发出的命令CMD_{nm},而资源仲裁部26向序列执行部24发送请求许可通知ACK_{nm}和表示命令执行完成的完成通知DONE_{nm}。在“REQ_{nm}”中,“n”表示作为请求源的序列执行部24_n的编号,“m”表示作为请求目的地的资源仲裁部26_m的编号。“CMD_{nm}”、“ACK_{nm}”和“DONE_{nm}”中的“n”和“m”的含义相同。

[0073] 第一生产管理部22_1以环形缓冲器的方式依次使用预定数量(例如,四个)的序列执行部24来执行第一品目的生产。例如,第一生产管理部22_1指示第一序列执行部24_1装载第一工件,指示第二序列执行部24_2装载第二工件,指示第三序列执行部24_3装载第三工件,以及指示第四序列执行部24_4装载第四工件。然后,当第一工件的生产完成时,第一

生产管理部22_1指示第一序列执行部24_1装载第五工件。类似地,第二生产管理部22_2也以环形缓冲器的方式依次使用预定数量(例如,四个)的序列执行部24,并且在加载工件加工命令的同时执行第二品目的生产。

[0074] 这里,这样的序列执行部24的数量没有特别限制,只要是两个以上即可。然而,为了减少工作资源130在不执行工作的情况下浪费地等待的时间并实现有效的生产,针对能够同时装载到的加工单元100中的工件的最大数量(在下文中称为最大可装载数量 W_{max})来设置序列执行部24。当并行生产第一品目和第二品目时,最大可装载数量 W_{max} 是能够同时装载到加工单元100中的用于第一品目的工件的最大数量(在下文中称为“第一可装载数量 W_{1max} ”)和能够同时装载到加工单元100中的用于第二品目的工件的最大数量(在下文中称为“第二可装载数 W_{2max} ”)的总和。换句话说,满足以下关系: $W_{max}=W_{1max}+W_{2max}$ 。

[0075] 当根据图4和图5中所示的生产程序并行地生产第一品目的物品和第二品目的物品时,用于第一品目的工件可以同时位于由机器人110、第一加工机器102、临时放置台120和工件清洁器116组成的四个场所处。因此,在这种情况下,第一可装载数量 W_{1max} 是“4”。机器人110可以暂时保持两个工件,但是在本示例中,并不在整个工序中维持两个工件被分配给一个工件资源130的状态。因此,可以将位于机器人110处的工件的数量计数为“1”。类似地,也可以将位于第一加工机器102处的工件的数量计数为“1”。

[0076] 在这种情况下,用于第二品目的工件可以同时位于由机器人110、第二加工机器104、临时放置台120和工件清洁器116组成的四个场所处。因此,第二可装载数量 W_{2max} 也是“4”。则,最大可装载数量 W_{max} 是8($W_{max}=4+4$)。当仅提供最大可装载数量 W_{max} (=8)时,序列执行部24可以并行地生产具有最大可装载数量 W_{max} 的工件,从而减少工作资源130浪费地等待的时间并实现有效生产。可以以大于最大可装载数量 W_{max} (=8)的量来设置序列执行部24,但是应当注意,要使用的计算机资源的量增加了,而生产效率却不能期望进一步提高。

[0077] 分别对应于多个工作资源130来设置资源仲裁部26,并且资源仲裁部26仲裁多个序列执行部24中的每一个序列执行部24使用对应的工作资源130的时刻。在图1的示例的情况下,由于以第一加工机器102、第二加工机器104和机器人110设置了三个工作资源130,因此设置三个资源仲裁部26。然后,例如,第一资源仲裁部26_1与机器人110(第一工作资源130_1)相关联,第二资源仲裁部26_2与第一加工机器102(第二工作资源130_2)相关联,第三资源仲裁部26_3与第二加工机器104(第三工作资源130_3)相关联。

[0078] 与在生产管理部22和序列执行部24中一样,资源仲裁部26也是要在计算机上启动的应用。因此,当处理器12执行在存储器14中存储的用于资源仲裁部26的应用程序时,计算机充当资源仲裁部26。在本示例中,由于设置了多个资源仲裁部26,因此用于多个资源仲裁部26的应用彼此独立地启动。

[0079] 如上所述,资源仲裁部26对多个序列执行部24中的每一个序列执行部24使用对应的工作资源130的时刻进行仲裁。对于这样的仲裁,资源仲裁部26随时从多个序列执行部24接收使用请求 REQ_{nm} 。然后,当释放对应的工作资源130时,基于预先在程序中定义的优先级顺序,从接收到的使用请求中选择一个使用请求 REQ_{nm} 。这里,可以根据装置的配置来适当地改变用于确定优先级顺序的算法。在本示例中,根据以下(1)至(3)来选择一个使用请求 REQ_{nm} :

[0080] (1) 在同一品目的使用请求REQnm中,以具有较高工序数量的请求为优先。

[0081] (2) 在比较不同品目的使用请求REQnm的情况下,以在发出使用请求REQnm之后具有长等待时间的请求为优先。

[0082] (3) 在不同品目的使用请求REQnm具有相同的等待时间的情况下,以第一品目的使用请求REQnm为优先。

[0083] 当可以选择一个使用请求REQnm时,资源仲裁部26向发出这样的使用请求REQnm的序列执行部24输出许可通知ACKnm。在接收到许可通知ACKnm时,序列执行部24向资源仲裁部26输出命令CMDnm,该命令CMDnm包括在生产程序中登记的资源程序的标识信息。资源仲裁部26将命令CMDnm输出到对应的工作资源130。当工作资源130根据发送的命令CMDnm运行时,执行工件的加工处理。当命令CMDnm的执行完成时,工作资源130向资源仲裁部26输出完成通知DOnem。资源仲裁部26向序列执行部24发送接收到的完成通知DOnem。在接收到完成通知DOnem时,序列执行部24将对工作资源130的释放或后续工序的命令CMDnm通知给资源仲裁部26。此后,所有资源仲裁部26重复相同的工序,由此生产出工件。

[0084] 以下将描述生产管理部22、序列执行部24和资源仲裁部26中的每一者的处理流程。图6是示出第一生产管理部22_1的处理流程的流程图。第一生产管理部22_1等待,直到从操作员或外部计算机接收到生产命令为止(S100)。生产命令包括关于要生产的品目的标识信息和要生产的品目的批次数量L1。当接收到生产命令时,第一生产管理部22_1将第一装载数量W1now和第一完成数量W1done复位为“0”(S102)。第一装载数量W1now是当前存在于加工单元100中的用于第一品目的工件的数量。另外,第一完成数量W1done是当前已经生产出的用于第一品目的工件的数量。

[0085] 接下来,第一生产管理部22_1将第一完成数量W1done与批次数量L1进行比较(S104)。当W1done<L1时,第一生产管理部22_1随后确认是否可以装载用于第一品目的新工件。具体地,第一生产管理部22_1将第一装载数量W1now与第一可装载数量W1max进行比较(S106),并使用工序计数CNT=0来判断是否存在前一工件(S108)。工序计数CNT是由每个序列执行部24当前正在执行的工序的数量的值,并且工序计数CNT的值由序列执行部24管理。

[0086] 当W1now≥W1max时,可以说加工单元100不能承受新装载用于第一品目的工件。此外,即使W1now<W1max,当存在具有CNT=0的前一工件时(也就是说,由于某些原因,其生产工序尚未从初始状态向前推进的前一工件),不能对装载的新工件向前推进生产进度。因此,当W1now≥W1max(S106中为“否”)时,或者当存在具有CNT=0的前一工件(S108中为“是”)时,第一生产管理部22_1不指示序列执行部24执行生产程序,并且处理前进至步骤S114。

[0087] 另一方面,当W1now<W1max(在S106中为“是”)且不存在具有CNT=0的前一工件(在S108中为“否”)时,第一生产管理部22_1指示不忙碌的序列执行部24执行生产程序(S110),然后将第一装载数量W1now递增(S112)。当第一生产管理部22_1将要执行的生产程序的标识信息(例如,名称和地址)作为自变量发送到序列执行部24时,执行指令可以被执行。

[0088] 在步骤S114中,第一生产管理部22_1监视是否有来自序列执行部24的加工完成的通知。当有来自序列执行部24的加工完成的通知时,第一装载数量W1now递减,并且第一完成数量递增(S116)。另一方面,当没有加工完成的通知时,工序返回至步骤S104而不改变这些参数。然后,重复步骤S104至S116,直到W1done≥L1(S104中为“否”)为止。这里,作为示例

描述了第一生产管理部22_1,但是这同样适用于第二生产管理部22_2的处理流程。

[0089] 下面将参照图7和图8来描述序列执行部24的处理流程。序列执行部24等待,直到由生产管理部22指示执行生产程序为止(S200)。当指示了执行生产程序时,序列执行部24将工序计数CNT复位为“0”(S201)。接着,序列执行部24从生产管理部22读出指派的生产程序,并指定用于执行初始工序的工作资源130(S202)。这可以通过参考生产程序的资源列(参见图4和图5)来指定。当机器人110用作工作资源130(在S204中为“是”)时,序列执行部24执行用于机器人的预定处理(S206),这将在下面描述。

[0090] 接着,当能够指定要使用的工作资源130时,序列执行部24发出指定的工作资源130的使用请求REQnm(S208)。这样的使用请求REQnm被输出到对应于工作资源130的资源仲裁部26。因此,当使用机器人110时,第n序列执行部24_n输出对对应于机器人110的第一资源仲裁部26_1的使用请求REQn1。另外,当机器人110和第一加工机器102都被使用时,第n序列执行部24_n输出对对应于机器人110的第一资源仲裁部26_1的使用请求REQn1和对对应于第一加工机器102的第二资源仲裁部26_2的使用请求REQn2两者。每个使用请求REQnm包括品目的标识信息、由序列执行部24管理的工序计数CNT的值。

[0091] 然后,序列执行部24等待,直到所有被请求要使用的工作资源130都被许可使用为止;即,直到接收到许可通知ACKnm为止(S210)。当所有的工作资源130都被许可使用时,序列执行部24将工序计数CNT递增(S212),然后将要发出给要使用的工作资源130的命令CMDnm输出到对应的资源仲裁部26(S214)。因此,命令CMDnm经由资源仲裁部26被输入到工作资源130,并且根据命令CMDnm的处理前进。序列执行部24等待,直到命令CMDnm的执行完成为止;即,直到接收到完成通知DONEnm为止(S216)。

[0092] 当命令CMDnm的执行完成(在S216中为“是”)时,序列执行部24将工序计数CNT与最大工序数量CNTmax进行比较(S218)。当 $CNT \geq CNT_{max}$ (S218中为“否”)时,这意味着生产程序的执行完成。在这种情况下,序列执行部24释放所有的工作资源;也即,将使用请求REQnm设定为关闭(OFF)(S220),然后通知生产管理部22加工已完成(S222)。当接收到通知时,生产管理部22将装载数量Wnow递减并将已完成数量Wdone递增。

[0093] 另一方面,当在步骤S218中 $CNT < CNT_{max}$ (S218中为“是”)时,这意味着存在未执行的工序。因此,在这种情况下,序列执行部24执行下一工序。具体地,序列执行部24在下一工序中释放不必要的工作资源130;也即,将使用请求REQnm设定为OFF(S224),然后指定在下一工序中要新使用的工作资源130(S226)。然后,当要新使用的工作资源130包括机器人110(S230中为“是”)时,序列执行部24执行对机器人的预定处理(S206),而当不包括机器人110(S230中为“否”)时,序列执行部24立刻向要使用的工作资源130发出使用请求REQnm(S232)。然后,当请求要使用的所有工作资源130都被许可使用(在S234中为“是”)时,工序前进至步骤S212。然后,重复相同的工序,直到 $CNT \geq CNT_{max}$ 为止。

[0094] 这里,将参照图9和图10来描述用于机器人的处理。图9是示出用于机器人的处理的流程的流程图。图10是图示出下面将描述的前一工件通过判断工序计数CNTs的图。在加工单元100中设置的工作资源130中,机器人110具有输送工件的功能。换句话说,机器人110充当用于将工件装载到加工机器中的装载装置,并且还充当将工件从加工机器中卸载的卸载装置。除非考虑到其它工件的进度而发出用于这样的机器人110的使用请求REQnm,否则会发生前一工件的工序进度与后一工件的工序进度相互干扰的死锁。

[0095] 例如,当机器人110是单手型机器人时,假定在从第一加工机器102拆卸下前一工件之前给予后一工件使用机器人110的许可。在这种情况下,前一工件保留在第一加工机器102中,使得后一工件不能附接到第一加工机器102,并且机器人110被后一工件占用,使得前一工件不能从第一加工机器102拆卸下来。结果,发生了将后一工件附接到第一加工机器102的工序和将前一工件从第一加工机器102拆卸的工序都不能进一步向前推进的死锁。

[0096] 当机器人110是双手型机器人时,可以同时保持两个工件,从而些许缓解了上述问题。然而,如上所述,在双手型的情况下,机器人110包括用于保持未加工工件的手和用于在切割之后保持工件的手。因此,机器人110在切割后只能保持一个工件。结果,当使用后切割手从工件清洁器116中取出前一工件时,发出使用请求REQnm,以使用后切割手将后一工件附接到工件清洁器116。当机器人110被分配给后一工件时,发生死锁。

[0097] 因此,在本示例中,当新请求使用机器人110时,执行用于机器人的处理以执行与前一工件的调整。在用于机器人的处理中,首先,指定对应于当前工序计数CNT的前一工件通过判断工序计数CNTs。前一工件通过判断工序计数CNTs是当后一工件发出机器人110的使用请求REQnm时前一工件应当已完成的工序计数。

[0098] 将详细描述前一工件通过判断工序计数CNTs。在前一工件通过判断工序计数CNTs的判断期间,下一资源调用工序和安装部分是重要的。下一资源调用工序是,对于当前工序随后的所有工序在首先停止使用机器人110之后再次使用机器人110的工序。另外,机器人110设置有被配置用于安装手的安装部分。安装部分的数量在单手型中为一个,在双手型中为两个。当在下一工序中使用的安装部分是在下一资源调用工序中使用的安装部分时,将下一资源调用工序的值设定为前一工件通过判断工序计数CNTs,并且当在下一工序中使用的安装部分不同于在下一资源调用工序中使用的安装部分时,或者当不存在下一资源调用工序时,将下一工序的值设定为前一工件通过判断工序计数CNTs。

[0099] 图10是示出工序计数CNT与前一工件通过判断工序计数CNTs之间的关系的示例的图。在图10的示例中,在当工序计数CNT(=0、7、11或13)完成的时刻,后一工件(序列执行部24)发出机器人110的使用请求REQn1。针对工序计数CNT(=0、7、11或13)设定前一工件通过判断工序计数CNTs。

[0100] 在单手型的情况下,在CNT=0之后的下一工序计数CNT(=0)中使用的安装部分是“h1”。此外,如从工序计数CNT(=0)看到的,下一资源调用工序是工序计数CNT(=8),在工序计数CNT(=8)中使用的安装部分是“h1”。因此,在单手型的情况下,对应于工序计数CNT(=0)的前一工件通过判断工序计数CNTs是“8”。类似地,对应于工序计数CNT(=7)的前一工件通过判断工序计数CNTs是“12”。关于工序计数CNT(=13),由于不存在下一资源调用工序,所以前一工件通过判断工序计数CNTs变成为下一工序的计数值;即“14”。

[0101] 另一方面,在双手型的情况下,在工序计数CNT(=0)之后的下一工序计数CNT(=1)中使用的安装部分是“h1”。如从工序计数CNT(=0)看到的,下一资源调用工序是工序计数CNT(=8),在工序计数CNT(=8)中使用的安装部分是“h2”。换句话说,在下一工序中使用的安装部分“h1”不同于在下一资源调用工序中使用的安装部分“h2”。因此,在这种情况下,对应于工序计数(=0)的前一工件通过判断工序计数CNTs是“1”。在工序计数CNT(=7)的情况下,在下一工序计数CNT(=8)中使用的安装部分“h2”等于在下一资源调用工序计数CNT=12中使用的安装部分“h2”。因此,对应于工序计数CNT(=7)的前一工件通过判断工序计

数CNTs是“12”。工序计数CNT和前一工件通过判断工序计数CNTs之间的关系可以通过利用序列执行部24对生产程序的解读或者通过由操作员预先准备的图10中所示的对应表来自自动确定。

[0102] 在用于机器人的处理中,首先,指定对应于当前工序计数CNT的前一工件通过判断工序计数CNTs (S240)。接下来,序列执行部24将前一工件的当前工序计数(在下文中称为“前一工序计数CNTp”)与指定的前一工件通过判断工序计数CNTs进行比较 (S242)。作为比较的结果,当 $CNTp < CNTs$ (S242中为“否”)时,序列执行部24不向机器人110发出使用请求REQnm,并按原样等待。另一方面,当 $CNTp \geq CNTs$ (在S242中为“是”)时,工序前进至步骤S208或S232,并且序列执行部24向机器人110发出使用请求REQnm。

[0103] 如上所述,对于引起死锁的工作资源130,可以通过在考虑到前一工件的进度的情况下发出使用请求REQnm来可靠地防止死锁。在本示例中,仅为机器人110考虑到前一工件的进度,但只要其他工件资源130引起死锁,则也可以为该其他工件资源130考虑前一工件的进度。在本示例中,序列执行部24掌握前一工件的进度,并调整使用请求REQnm的发出时刻。然而,作为序列执行部24的替代,资源仲裁部26也可以掌握前一工件的进度,并调整许可通知ACKnm的发出时刻。在图10的示例中,手一旦被安装到机器人110,则直到生产程序结束才被替换。然而,如果需要,可以替换安装到一个安装部分的手的类型。

[0104] 接下来,将描述资源仲裁部26的处理。图11是示出资源仲裁部26的处理流程的流程图。与图11的处理并行地,资源仲裁部26随时从多个序列执行部24接收对应的工作资源130的使用请求REQnm。在该状态下,当释放了对应的工作资源130 (在S300中为“是”)时,资源仲裁部26在此时从接收到的多个使用请求中选择一个使用请求REQnm (S302)。图12是示出选择处理的流程的流程图。

[0105] 如图12中所示,当选择请求时,首先,资源仲裁部26从与第一品目有关的使用请求中指定具有最大工序计数CNT的第一使用请求REQ1和该第一使用请求REQ1的等待时间Tw1 (S320)。随后,资源仲裁部26指定与第二品目有关的使用请求中具有最大工序计数CNT的第二使用请求REQ2和该第二使用请求REQ2的等待时间Tw2 (S322)。随后,资源仲裁部26将第一使用请求REQ1的等待时间Tw1与第二使用请求REQ2的等待时间Tw2进行比较 (S324)。

[0106] 作为比较的结果,当等待时间Tw1大于或等于等待时间Tw2 (S324中为“是”)时,资源仲裁部26选择第一使用请求REQ1 (S326)。另一方面,当等待时间Tw1短于等待时间Tw2 (S324中为“否”)时,资源仲裁部26选择第二使用请求REQ2 (S328)。在下面的描述中,将在步骤S326和S328中所选择的使用请求称为“使用请求REQnm”。

[0107] 如果可以根据图12的流程选择出一个使用请求REQnm,则资源仲裁部26随后向已经发出所选择的使用请求REQnm的第n序列执行部24_n发送许可通知ACKnm (S304)。由此,由第n序列执行部24_n确保与第m资源仲裁部26_m对应的第m工作资源130_m。

[0108] 在发出许可通知ACKnm之后,资源仲裁部26等待,直到从序列执行部24接收到命令CMDnm为止 (S306)。当接收到命令CMDnm时,资源仲裁部26将命令CMDnm发送到对应的工作资源130 (S308)。由此,工作资源130根据由命令CMDnm指示的资源程序而运行。当完成由资源程序指示的操作时,工作资源130向资源仲裁部26发送完成通知DOPENm。当接收到完成通知DOPENm (在S310中为“是”)时,资源仲裁部26向序列执行部24发送完成通知DOPENm (S312)。此后,当从序列执行部24接收到释放指令 (在S314中为“是”)时,资源仲裁部26释放对应的

工作资源130 (S316), 并且工序返回至步骤S300。另一方面, 当没有释放指令 (S314中为“否”) 时, 工序返回至步骤S306, 资源仲裁部26等待, 直到接收到下一命令CMDnm为止。

[0109] 如从以上描述清楚得知的, 根据本示例, 资源仲裁部26对从多个序列执行部24发出的使用请求REQnm之间的争用进行仲裁。因此, 操作员可以为单个工件生产创建程序; 即, 仅生产程序而无需知晓工件之间的争用。因此, 在物品的生产中不需要高度专业化的系统集成商, 从而可以降低成本。此外, 由于可以缩短创建和修改用于控制加工单元100的程序所需的时间, 因此可以进一步提高生产效率。

[0110] 接下来, 将描述另一具体示例。在该具体示例中, 序列执行部24在当前正在执行的工序完成之前 (即, 在接收完成通知DONEnm之前) 发出对执行下一工序所需的工作资源130的使用请求REQnm。由此, 能够缩短作业资源130的浪费的等待时间, 并且能够进一步提高生产效率。

[0111] 换句话说, 在上述描述中, 序列执行部24在直至当前正在执行的工序完成之前, 不会发出下一工序所需的使用请求REQnm。另一方面, 在接收到命令CMDnm时, 工作资源130首先执行预定的准备工作, 并且在准备工作完成之后开始对工件的工作。例如, 在机器人110的情况下, 当接收到命令CMDnm时, 首先执行例如手替换等的准备工作, 然后开始工件的输送。因此, 当在当前执行工序完成之后发出下一使用请求REQnm时, 存在工件可能浪费地等待, 直到在当前处理完成之后下一处理开始为止。例如, 考虑工件在由第一加工机器102切割之后被输送到工件清洁器116的情况。在这种情况下, 当在切割完成之后向机器人110发出使用请求REQnm时, 由于在切割完成之后机器人110开始例如手替换等的准备工作, 因此工件在第一加工机器102中浪费地等待这样的时段, 这导致较长的生产时间。

[0112] 因此, 在本示例中, 序列执行部24在当前执行的工序完成之前发出对下一工序的执行所需的工作资源130d的使用请求REQnm。这里, 发出下一工序的使用请求REQnm的时刻 (在下文中称为“预发出时刻”) 没有特别限定, 只要其在当前工序完成之前即可。然而, 当预发出时刻太晚时, 工件的浪费的等待时间增加。另一方面, 当预发出时刻太早时, 一个序列执行部24占用工作资源130的时间变得不必要地长, 这导致其他序列执行部24的等待时间的增加。因此, 可以基于下一工序的准备时间来确定预发出时刻。例如, 通过从当前工序的执行时间中减去下一工序的准备时间来获得等待时间, 并且当当前工序发出命令CMDnm之后经过的时间达到等待时间时的时刻可以被视为预发出时刻。每个工序的执行时间和准备时间可以从过去的实际结果或模拟中预先获得。另外, 每个工序的执行时间和准备时间可以被记录在行程程序中, 或者可以被记录在独立于生产程序的单独文件中。在任何情况下, 序列执行部24基于在存储器14中存储的每个工序的执行时间和准备时间来指定每个工序的预发出时刻。

[0113] 图13和14是示出本具体示例中的序列执行部24的处理流程的流程图。在图13中, 步骤S200至S214的工序与图7中步骤S200至S214的工序几乎相同。换句话说, 当序列执行部24指定所需的工作资源130、向工作资源130发出使用请求REQnm、以及从所有工作资源130接收许可通知ACKnm时, 序列执行部24向相关工作资源130发出命令CMDnm。本示例中的用于机器人的处理 (S206*) 的内容与图7中的用于机器人的处理 (S206) 的内容稍有不同, 将在以下进行描述。

[0114] 当发出命令CMDnm时, 序列执行部24等待直到达到定义的预发出时刻 (S246)。在预

发出时刻之时,序列执行部24指定下一工序中新需要的工作资源130 (S226)。然后,序列执行部24向指定的工作资源130发出使用请求REQnm (S232*)。使用请求REQnm除了包括品目的标识信息和当前工序计数CNT的值之外,还包括开始下一工序所需的工作资源130的准备操作的内容。当下一工序中没有新需要的工作资源130时,序列执行部24跳过步骤S230、S206*、和S232*。

[0115] 当不必要发出使用请求REQnm时,或者当可以发出使用请求REQnm时,序列执行部24等待直到当前工序完成,即,接收到完成通知DONEnm为止 (S216)。当接收到完成通知DONEnm时,序列执行部24将工序计数CNT与工序最大数量CNTmax进行比较 (S218)。当 $CNT \geq CNTmax$ 时,序列执行部24释放所有的工作资源130 (S220),然后通知生产管理部22加工完成 (S222)。

[0116] 另一方面,当 $CNT < CNTmax$ 时,序列执行部24释放在下一工序中不再需要的工作资源130 (S224)。然后,由于对在下一工序中所需的工作资源130的使用请求REQnm已经被发出,所以序列执行部24在该时间之后等待,直到获得了对所有工作资源130的许可通知ACKnm为止 (S234)。当获得了对所有工作资源130的许可通知ACKnm时,工序返回至步骤S212,并且此后重复相同的处理。

[0117] 下面将参照图15描述本具体示例中的用于机器人的处理 (S206*)。图15中的处理与图9中的处理不同的是具有步骤S244。即,在图9中的用于机器人的处理中,指定了与后一工件的当前工序计数CNT对应的前一工件通过判断工序计数CNTs (S240),并且当前一工件的工序计数(即,前一工序计数CNTp)达到前一工件通过判断工序计数CNTs (S242中为“是”)时,许可发出后一工件的使用请求REQnm。

[0118] 在本示例中,如上所述,每个序列执行部24在当前工序完成之前发出对下一工序的请求。因此,即使当前一工件未达到前一工件通过判断工序计数CNTs时,也有可能发出针对前一工件通过判断工序计数CNTs的使用请求REQnm。因此,在本示例中,当如图15中所示的 $CNTp < CNTs$ (步骤S242中为“否”)时,前一工件的工序计数达到 $(CNTs - 1)$,确认是否已经发出针对CNTs的使用请求 (S244)。当已经发出针对CNTs的使用请求时,工序前进至步骤S208或S232*,并且许可发出使用请求REQnm。由此,可以更早地发出使用请求REQnm,并且可以减少工件的浪费的等待时间。结果,可以进一步提高物品的生产效率。

[0119] 将在以下描述另一具体示例。图16是另一具体示例的单元控制器10的功能框图。本具体示例的单元控制器10与上述单元控制器10的不同之处在于,从构成生产程序的多个工序中指定能够并行执行的两个工序,并且当时刻正确时并行执行这两个工序。

[0120] 具体地,双手型机器人110可以同时保持两个工件。因此,一只手可以在切割之前保持后一工件,另一只手可以在切割之后保持前一工件。然后,使用这样的双手型机器人110,可以并行地执行将前一工件从加工机器拆卸的工序和将后一工件附接到加工机器的工序。加工单元100包括例如能够同时分配两个以上工件的双手型机器人110等的元件,并且通过使用这样的元件,可以与用于前一工件的预定工序并行地执行用于后一工件的另一工序。当存在可以以这种方式并行执行的工序时,可以通过并行执行这些工序来缩短时间。

[0121] 因此,在本示例中,指定了能够并行执行的两个工序,并且当前一工件或后一工件请求执行两个工序中的一个工序时,则在当后一工件或前一工件请求执行两个工序中的一个工序的时刻并行执行这两个工序。为了实现这样的操作,如图16中所示,本具体示例的单

元控制器10包括复合表存储部32,并且每个资源仲裁部26包括复合操作判断部31。

[0122] 复合表存储部32存储复合表。复合表是记录有一对可并行执行的两个不同工序以及当这两个工序并行执行时的资源程序(在下文中称为“复合程序”)的表。复合程序充当允许工作资源130执行两个以上工序的复合命令。图17是示出复合表的示例的图;在图17的示例中,将后一工件的工序的标识信息、能够与后一工件的工序并行执行的前一工件的工序的标识信息、以及对应于这两个工序的复合程序的标识信息彼此关联地存储。例如,复合表的第一行记录有后一工件的工序编号“3”和前一工件的工序编号“8”。此外,第一行还记录有并行执行工序编号“3”和工序编号“8”所需的资源程序的名称“RPR003C-W01234”。

[0123] 复合表可以由操作员预先创建的状态被记录在单元控制器10中。替代地,单元控制器10可以自动生成复合表。在这种情况下,操作员预先创建加工单元100的详细配置(例如,附接到机器人110的手的数量和类型)、用于在使用机器人110的工序中保持工件的手的标识信息、以及在工件的输送工序中工件的输送目的地或输送源的标识信息,并将它们作为初步信息登记在加工单元100中。这样的初步信息可以记录在独立于生产程序的文件中,或者可以合并入生产程序中。单元控制器10可以基于这样的初步信息和生产程序而自动生成复合表。替代地,可以通过单元控制器10与操作员之间的对话来收集上述初步信息,并且单元控制器10可以生成复合表。

[0124] 如图16中所示,每个资源仲裁部26包括复合操作判断部31。复合操作判断部31参照复合表,判断工序是否被整合,并且,在当工序被整合时,指令对应的工作资源130执行整合有两个工序的复合程序。图18和图19是示出包括复合操作判断部31的资源仲裁部26的处理流程的流程图。

[0125] 类似于图11和图12的处理流程,资源仲裁部26首先等待,直到对应的工作资源130变成为空闲为止(S300),并且,在空闲状态的情况下,从接收到的多个使用请求中选择一个使用请求REQ_{nm}(S302)。然后,资源仲裁部26向发出使用请求REQ_{nm}的第n序列执行部24_n发送许可通知ACK_{nm}(S304)。当从已经接收到许可通知ACK_{nm}的序列执行部24发送了命令CMD_{nm}(在S306中为“是”)时,资源仲裁部26指定与由命令CMD_{nm}指示的工序并行的工序编号,并从接收到的多个使用请求中搜索与该工序编号相对应的使用请求REQ_{km}(S330)。作为搜索的结果,在接收到关于能够并行执行的工序的使用请求REQ_{km}(在S332中为“是”)时,资源仲裁部26前进至步骤S334。另一方面,在没有接收到使用请求REQ_{km}时,资源仲裁部26执行步骤S308至S314。步骤S308至S316的内容与图11和图12中步骤S308至S316的内容相同,因此在此不再描述。

[0126] 在步骤S334中,资源仲裁部26向第k序列执行部24_k发送许可通知ACK_{km}。因此,此时,第n序列执行部24_n和第k序列执行部24_k两者都被许可使用对应的工作资源130。同时,此时,接收到第n序列执行部24_n的命令CMD_{nm},但未接收到第k序列执行部24_k的命令CMD_{km},因此资源仲裁部26等待,直到命令CMD_{km}从第k序列执行部24_k到来为止(S336)。

[0127] 当获得了命令CMD_{nm}和命令CMD_{km}两者(在S336中为“是”)时,资源仲裁部26参照复合表,并将这两个命令CMD_{nm}和CMD_{km}转换成复合命令CMD1(S338)。复合命令CMD1包括复合程序的标识信息。资源仲裁部26将复合命令CMD1发送到对应的工作资源130(S340)。此后,工作资源130并行地执行第n序列执行部24_n所请求的工序和第k序列执行部24_k所请求的工序。

[0128] 资源仲裁部26等待,直到从工作资源130发送出完成通知DONE1为止(S342)。当接收到完成通知DONE1时,资源仲裁部26向第n序列执行部24_n发送完成通知DONE_nm,并向第k序列执行部24_k发送完成通知DONE_km。

[0129] 当一个工序的执行完成时,资源仲裁部26然后确认是否存在释放工作资源130的指令(S346至S350)。然后,当第n序列执行部24_n和第k序列执行部24_k都没有给出释放工作资源130的指令(S346中为“否”且S348中为“否”)时,资源仲裁部26返回至步骤S336并重复后续处理。另一方面,当不存在来自第n序列执行部24_n的释放指令,但存在来自第k序列执行部24_k的释放指令(在S346中为“否”且在S348中为“是”)时,资源仲裁部26返回至步骤S306,并重复后续处理。此外,当第n序列执行部24_n和第k序列执行部24_k都给出释放指令(在S346中为“是”且在S350中为“是”)时,资源仲裁部26前进至步骤S316,并释放工作资源130。此外,当存在来自第n序列执行部24_n的释放指令,但不存在来自第k序列执行部24_k的释放指令(在S346中为“是”且在S350中为“否”)时,资源仲裁部26前进至步骤S306。在步骤S306中,描述了“命令CMD_nm”,但是在此情况下,“m”替换为“n”,并且处理向前推进。

[0130] 从上面的描述可以清楚地看出,根据本示例,只要时刻正确,就并行地执行两个工序。结果,能够进一步缩短生产物品所需的时间,并且能够进一步提高生产效率。在这种情况下,序列执行部24可以执行图7至图9中所示的处理,或者可以执行图13至图15中所示的处理。当序列执行部24执行图13至图15中所示的处理时,在资源仲裁部26向前一工件发出许可通知ACK_nm(S304)之后,直到接收到命令CMD_nm(S306)为止的时间变得更长。因此,同时增加了接收后一工件的使用请求REQ_km的可能性。结果,当序列执行部24执行图13至图15中所示的处理时,增加了能够并行执行两个工序的可能性,从而能够更可靠地提高生产效率。

[0131] 图20至图22是示出通过仿真本公开中公开的单元控制器10的算法而获得的结果的图。在该仿真中,图13至图15的算法被采用作为序列执行部24,图18至图19的算法被采用作为资源仲裁部26。图20至图22示出了随着每个工件的时间的流逝的工序的转变,假设工序向下行进并且时间向右行进。“标准执行时间”指示当电子数据表的一行被视为一个单位时间时每个工序的标准执行时间。因此,当工件位于一个工序的行中的时间(单元数量)超过标准执行时间时,可以说该工件正在等待在下一工序中要使用的工作资源变为可用。在下面的描述中,第i品目的第j工件称为“工件W_ij”。因此,第一品目的第二工件称为“工件W12”,第二品目的第三工件称为“工件W23”。

[0132] 图20示出了四个第一品目由第一加工机器102加工,并且四个第二品目由第二加工机器104加工的状态。在该示例中,按照工件W11、工件W21、工件W12和工件W22的顺序向机器人110发出使用请求。结果,如图20中所示,首先生产工件W11,并由机器人110将工件W11附接到第一加工机器102。然后,在当机器人110变成空闲时的时刻7,机器人110被分配给工件W21,并且工件W21的生产开始。类似地,机器人110在时刻12被分配给工件W12,并且在时刻27被分配给工件W22,由此每个工件的生产开始。然后,各工件的执行顺序由资源仲裁部26仲裁,以使得各工件不相互竞用。

[0133] 例如,因为在时刻14工序“3”完成,但是在下一工序“4”中使用的第一加工机器102被前一工件W11占用,所以工件W12连续等待直到时刻23为止。另外,在时刻24,工件W12的工序“4”和工件W11的工序“9”根据复合表被整合在一起,并且被并行执行。

[0134] 图21示出了当工序“13:工件清洁”的标准执行时间比图20中的更长时的仿真结

果。在这种情况下,很可能在工件清洁器116的前面发生工件的拥塞。结果,在图20的示例中要同时装载到加工单元100中的最大装载数量 W_{now} 是“4”,但是在图21的示例中增加到“5”。

[0135] 图22示出了当使用第一加工机器102的工序和使用第二加工机器104的工序串行连接时的仿真结果。换句话说,图22示出了当一个工件经受第一加工机器102的切割和第二加工机器104的切割两者时的仿真结果。即使在这种情况下,也可以看出,由资源仲裁部26对工件之间的争用进行仲裁以有效地生产多个工件。

[0136] 接下来,将进一步描述另一具体示例。图23是示出另一具体示例的单元控制器10的部分配置的图。该单元控制器10与图2中的单元控制器10不同的方面在于,包括重试判断部34。重试判断部34监视由每个序列执行部24执行的工序的状态,并且在当判断出工序中发生异常时,输出指令序列执行部24重新执行该工序的重试命令 RTR_n 。异常的存在与否可以基于从每个工序的开始起所经过的时间来判断,或者可以基于来自工作资源130的误差信号 ERR_m 来判断。

[0137] 当基于所经过的时间判断是否存在异常时,单元控制器10还包括执行时间收集部36。执行时间收集部36从多个序列执行部24收集工序的执行状态,并基于收集到的数据生成记录有每个工序的标准执行时间的标准执行时间表。具体地,多个序列执行部24中的每一个序列执行部24设置有对当前工序编号进行计数的工序计数器30。当向上计数时,工序计数器30将计数数据 $COUNT_n$ 输出到执行时间收集部36。计数数据 $COUNT_n$ 包括输出计数数据 $COUNT_n$ 和当前工序计数 CNT 的序列执行部24的标识信息。执行时间收集部36测量每个工序的向上计数间隔时间,并将其存储为该工序的执行时间。在本示例中,多个序列执行部24在不同时间执行相同的生产程序。因此,针对一个工序可以获得多个执行时间。执行时间收集部36计算通过对针对一个工序获得的多个执行时间执行统计处理而获得的代表值,例如平均值或中间值等,作为该工序的标准执行时间。然后,执行时间收集部36生成标准执行时间表,该标准执行时间表彼此关联地记录有工序和与工序相对应的标准执行时间,并将标准执行时间表输出到重试判断部34。

[0138] 重试判断部34监视从多个工序中的每一个工序的开始起所经过的时间,并且当所经过的时间超过基于标准执行时间确定的允许执行时间时,判断发生异常。计数数据 $COUNT_n$ 也被输入到重试判断部34以测量从工序开始起所经过的时间。重试判断部34测量在用内部计时器向上计数之后所经过的时间。此外,允许执行时间没有特别限制,只要其比标准执行时间更长即可。例如,允许执行时间是标准执行时间的两倍以上。当一个工序的所经过的时间超过这样的工序的允许执行时间时,重试判断部34判断在这样的工序的执行中发生了一些异常。

[0139] 在上述示例中,获取工序计数器30的向上计数间隔时间作为工序执行时间。然而,向上计数间隔时间不仅包括当工作资源130实际执行该工序时的工作时间(即,从发出 CMD_{nm} 直到接收到 $DONE_{nm}$ 的时间),还包括在发出用以执行下一工序的使用请求 REQ_{nm} 之后直到获得许可通知 ACK_{nm} 为止的请求等待时间。当这样的请求等待时间包括在行程执行时间中时,在某些情况下可以准确地判断工序异常的存在与否。因此,代替向上计数间隔时间,可以将从发出 CMD_{nm} 直到接收到 $DONE_{nm}$ 为止的工作时间视为工序执行时间。在这种情况下,序列执行部24在接收完成通知 $DONE_{nm}$ 的时刻向执行时间收集部36和重试判断部34发送在工序计数器30的向上计数的时刻的计数数据 $COUNT_n$ 和完成通知 $DONE_{nm}$ 。执行时间收集部

36和重试判断部34将从接收计数数据COUNT_n到接收完成通知DONE_n的时间视为行程执行时间。

[0140] 此外,附加或者替代于在从工序开始起所经过的时间超过允许执行时间的情况下,即使当从工作资源130输出了错误信号ERR_n时,重试判断部34也可以判断在工序的执行中发生了异常。然后,当确定在工序的执行中发生了异常时,重试判断部34向执行该工序的序列执行部24输出重试命令RTR_n。

[0141] 已经接收到重试命令RTR_n的序列执行部24取消当前正在输出的用于工作资源130的命令CMD_n,然后重新执行一些工序。这里,要重新执行的工序可以仅仅是其中发生了异常的工序,或者可以从发生了异常的工序之前的工序处重新执行。例如,当在工序“2”执行期间发生异常并且接收到重试命令RTR_n时,序列执行部24取消与工序“2”相关的命令CMD_n,然后可以再次将用于执行工序“2”的命令CMD_n从头输出到资源仲裁部26。

[0142] 替代地,可以为每个工序设定重试点以在异常发生时返回,并且当异常发生时,可以重新执行由重试点表示的工序。图24是示出添加了重试点的生产程序的示例的图。在图24中,“重试点”栏中的“1”表示充当重试点的工序。当接收重试命令RTR_n时,序列执行部24追溯到紧邻重试点之前,并重新执行该工序。

[0143] 例如,当在工序“5”的执行期间接收到重试命令RTR_n时,序列执行部24首先取消用于工序“5”的命令CMD_n。随后,序列执行部24参照生产程序,并确认工序“5”的“重试点”栏。在图24的示例中,工序“5”的“重试点”栏未记录有“1”。因此,在这种情况下,序列执行部24追溯到前一工序“4”,并确认其“重试点”。由于工序“4”的“重试点”记录有“1”,所以序列执行部24判断工序“4”是重试点,并从工序“4”重新执行该工序。换句话说,序列执行部24发出重新执行工序“4”所需的命令CMD_n和使用请求REQ_n。此外,当以这种方式追溯工序时,序列执行部24将工序计数CNT的值递减所回溯的工序数量。

[0144] 如上所述,本示例具有在工序的执行中监视异常的存在与否的配置,并且当异常发生时,重新执行包括发生异常的工序的一个以上工序。通过这种配置,能够在不改变单元控制器10的控制程序的情况下容易地处理异常。

[0145] 然而,取决于工序的内容,除非工作资源130执行特殊的返回操作,否则可以不重新执行工序。因此,代替在没有异常发生时执行的正常工序,预先准备了在重新执行工序时要执行的替代工序,并且可以在接收到重试命令RTR_n时执行该替代工序。图25是示出添加了替代工序的生产程序的示例的图。在图25的示例中,在生产程序中增加了“替代工序”栏。在“替代工序”栏中,描述了在重新执行时要执行的替代工序的编号。此外,在生产程序中增加了替代工序行。替代工序行记录有在替代工序中要执行的替代资源程序。

[0146] 当接收到重试命令RTR_n时,序列执行部24确认对应于当前正在执行的工序的“替代工序编号”,并且当没有“替代工序编号”时,按原样重新执行当前正在执行的工序。另一方面,当定义了“替代工序编号”时,序列执行部24执行在替代工序编号行中描述的替代资源程序。

[0147] 在图25的示例中,在工序“7”中没有定义替代工序编号。因此,当在执行工序“7”时接收到重试命令RTR_n时,序列执行部24按原样重新执行工序“7”。另一方面,在图25的示例中,“100800”被定义为工序“8”中的替代工序编号。因此,当在执行工序“8”时接收到重试命令RTR_n时,序列执行部24不重新执行工序“8”,而是执行替代工序“100800”。换句话说,序列

执行部24将在替代工序“100800”行中定义的替代资源程序作为命令CMDnm输出到资源仲裁部26。以这种方式,能够通过预先设定替代工序来更可靠地处理异常。

[0148] 以下将参考图26描述另一具体示例。图26中的单元控制器10与图23中的单元控制器10不同的方面在于,还包括工件跳过执行部38。工件跳过执行部38在当判断难以消除异常时,执行定义的工件跳过工序并弃置该工件。

[0149] 换句话说,当检测到异常发生时,重试判断部34指示序列执行部24执行用于执行正常工序或替代工序的重试。然而,当工件存在任何问题时,即使在重复了这样的重试时,也可能不会消除异常。当这样情况仍然存在时,后一工件不能向前推进该工序,导致拥塞。

[0150] 因此,工件跳过执行部38对用于该同一工序的重试命令RTRn被连续输出到同一序列执行部24的次数进行计数,作为连续重试次数。然后,当连续重试次数达到预定允许次数时,执行所定义的工件跳过工序。

[0151] 在工件跳过工序中,工件跳过执行部38向与工件资源130对应的资源仲裁部26发出用于将发生了异常的工件作为NG产品输送到弃置站点所需的强制命令CMDsm。这里,当需要多个工作资源130来输送到弃置站点时,工件跳过执行部向多个资源仲裁部26发出命令CMDsm。当接收到命令CMDsm时,资源仲裁部26将该命令发送到对应的工作资源130。然后,当工作资源130根据命令CMDsm进行操作时,发生异常的工件被输送到弃置站点。当弃置操作完成时;即,当从资源仲裁部26返回完成通知DONEsm到工件跳过执行部38时,工件跳过执行部38向管理被弃置的工件的生产的序列执行部24发出复位命令RSTn。在接收到复位命令RSTn时,序列执行部24复位工序计数器30的值,并通知生产管理部22被弃置的工件已经产生。此后,序列执行部24等待,直到接收到新的工件装载命令为止。另一方面,在接收到工件的弃置通知时,生产管理部22递减由自身管理的工件的装载数量Wnow的值。

[0152] 如上所述,在本具体示例中,当连续重试次数达到允许次数时,发生异常的工件被强制弃置,而未发生异常的其他工件照常地以生产程序被连续执行。换句话说,根据本具体示例,不需要为未发生异常的其他工件设计恢复工序,能够显著减少用于工件跳过的设计工时,并且能够防止该工序的设计错误。

[0153] 上述的配置均为示例,并且其他配置可以被适当地改变,只要单元控制器10基于记录有在加工单元100中仅生产一个目标品目所需的一个以上工序的生产程序来控制加工单元100的操作即可。

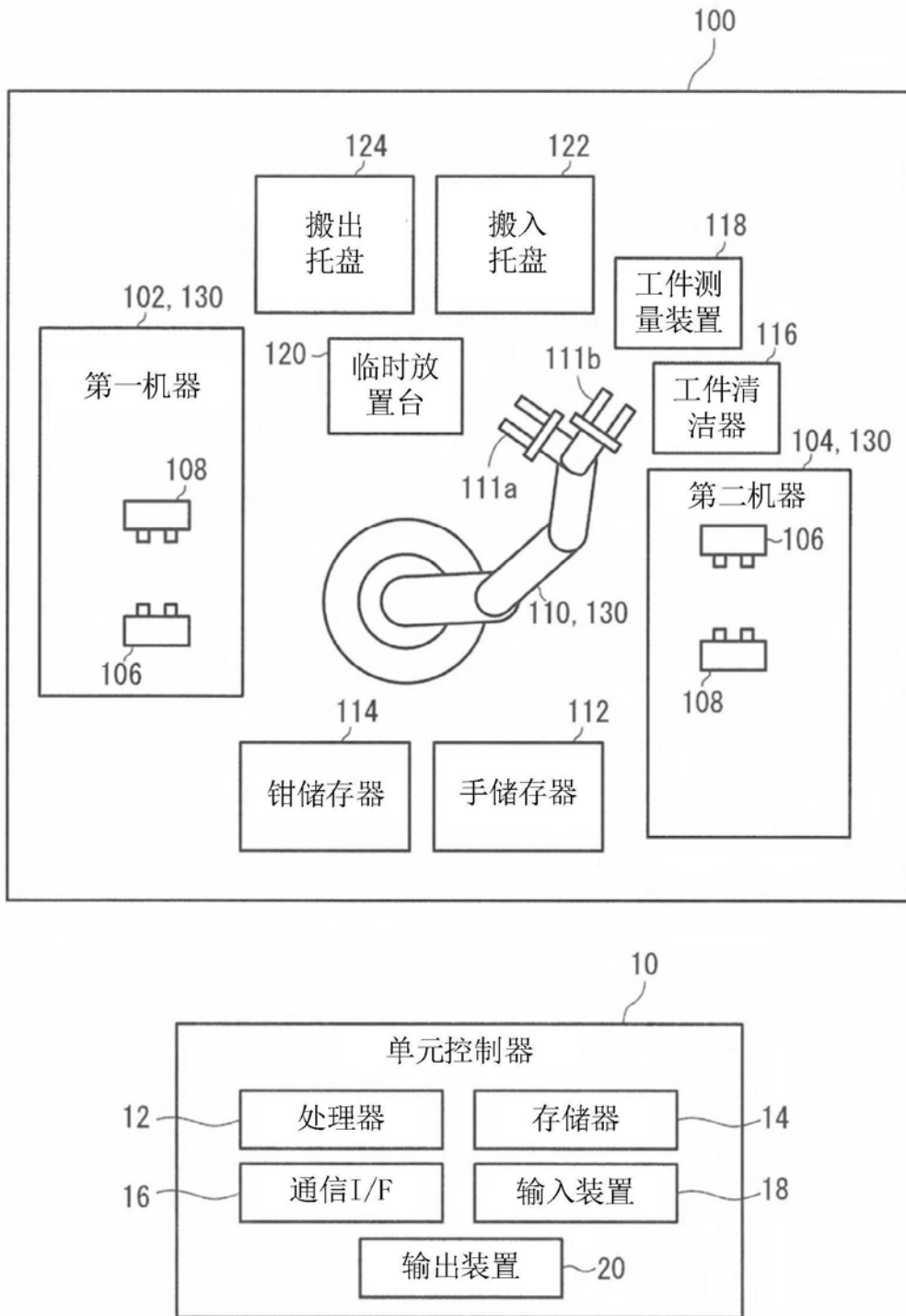


图1

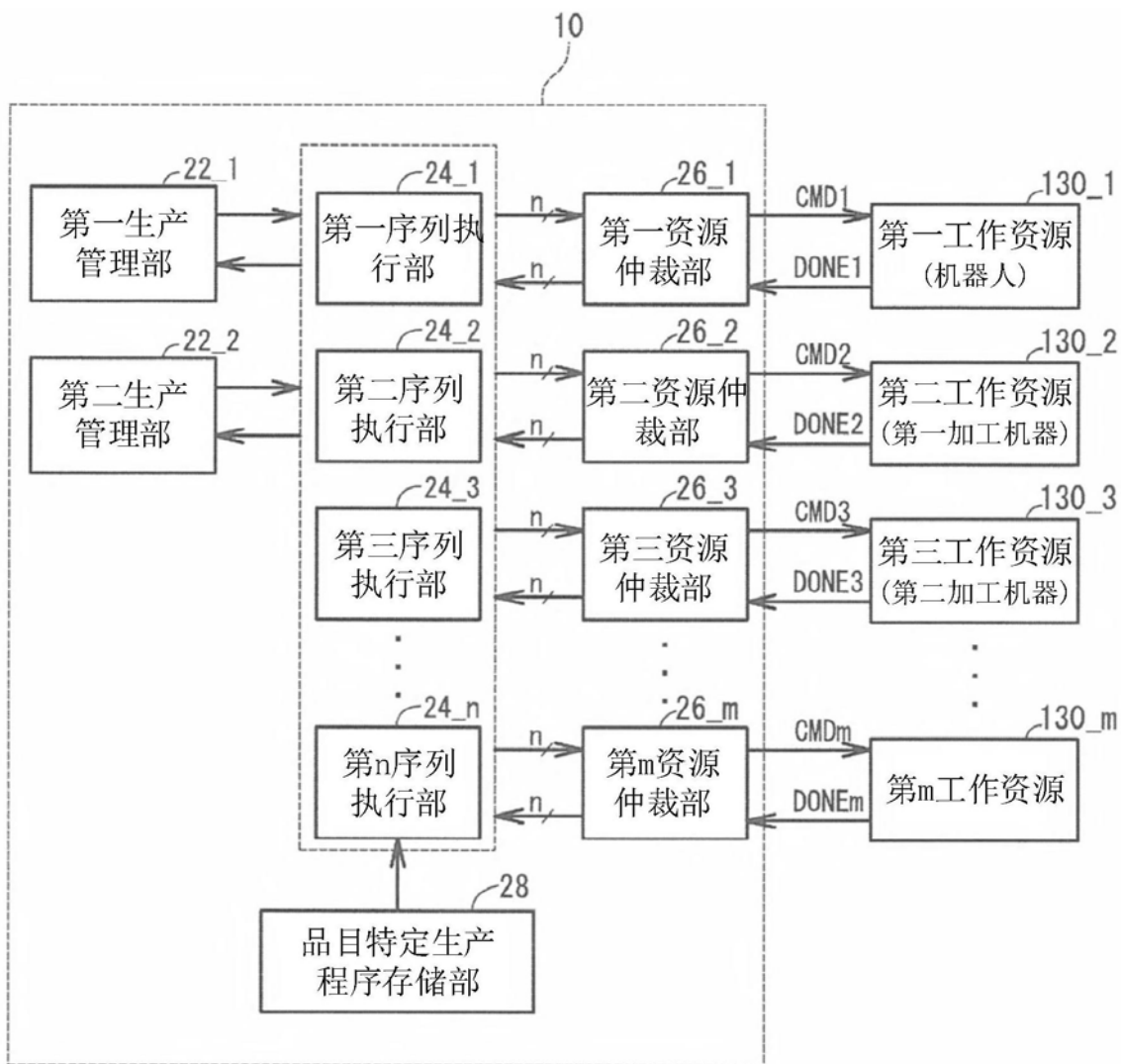


图2

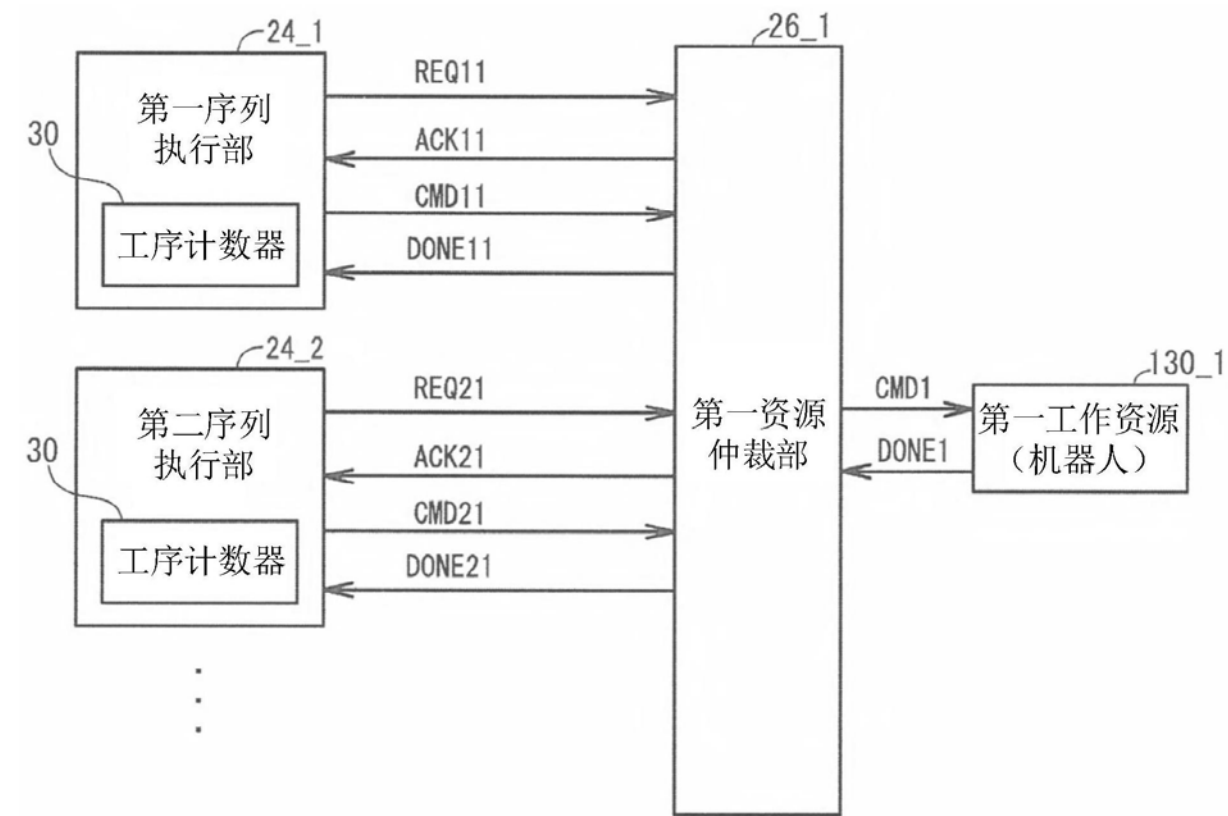


图3

进程		资源		资源程序	
工序编号	工序名称	NC	机器人	NC	机器人
0	在搬入托盘上等待				
1	材料的卸载		R1		RPR001-W01
2	材料尺寸的确认		R1		RPR002-W01
3	NC1安装准备		R1		RPR003-W01
4	NC1安装	NC1	R1	NPR004-W01	RPR004-W01
5	NC1的主轴加工	NC1		NPR005-W01	
6	在NC1的轴之间的递送	NC1		NPR006-W01	
7	NC1的副主轴加工	NC1		NPR007-W01	
8	NC1移除准备	NC1	R1	NPR008-W01	RPR008-W01
9	NC1移除	NC1	R1	NPR009-W01	RPR009-W01
10	装载到临时放置台上		R1		RPR010-W01
11	在临时放置台处等待				
12	附接至工件清洁器		R1		RPR012-W01
13	工件清洁				
14	装载搬出托盘		R1		RPR014-W01
15	在搬出托盘上等待				

图4

工序		资源		资源程序	
工序编号	工序名称	NC	机器人	NC	机器人
0	在搬入托盘上等待				
1	材料的卸载		R1		RPR001-W02
2	材料尺寸的确认		R1		RPR002-W02
3	NC2安装准备		R1		RPR003-W02
4	NC2安装	NC2	R1	NPR004-W02	RPR004-W02
5	NC1的主轴加工	NC2		NPR005-W02	
6	在NC2的轴之间的递送	NC2		NPR006-W02	
7	NC2的副主轴加工	NC2		NPR007-W02	
8	NC2移除准备	NC2	R1	NPR008-W02	RPR008-W02
9	NC2移除	NC2	R1	NPR009-W02	RPR009-W02
10	装载到临时放置台上		R1		RPR010-W02
11	在临时放置台处等待				
12	附接至工件清洁器		R1		RPR012-W02
13	工件清洁				
14	装载搬出托盘		R1		RPR014-W02
15	在搬出托盘上等待				

图5

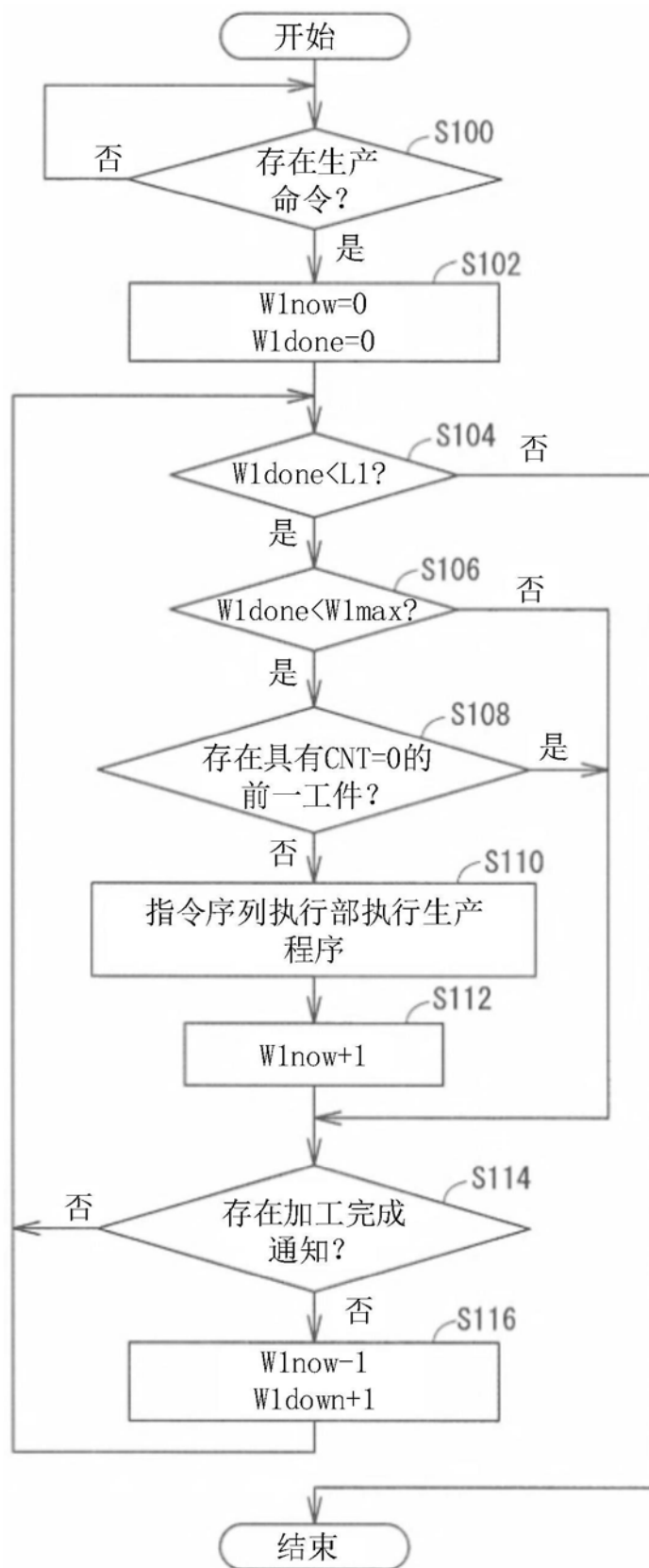


图6

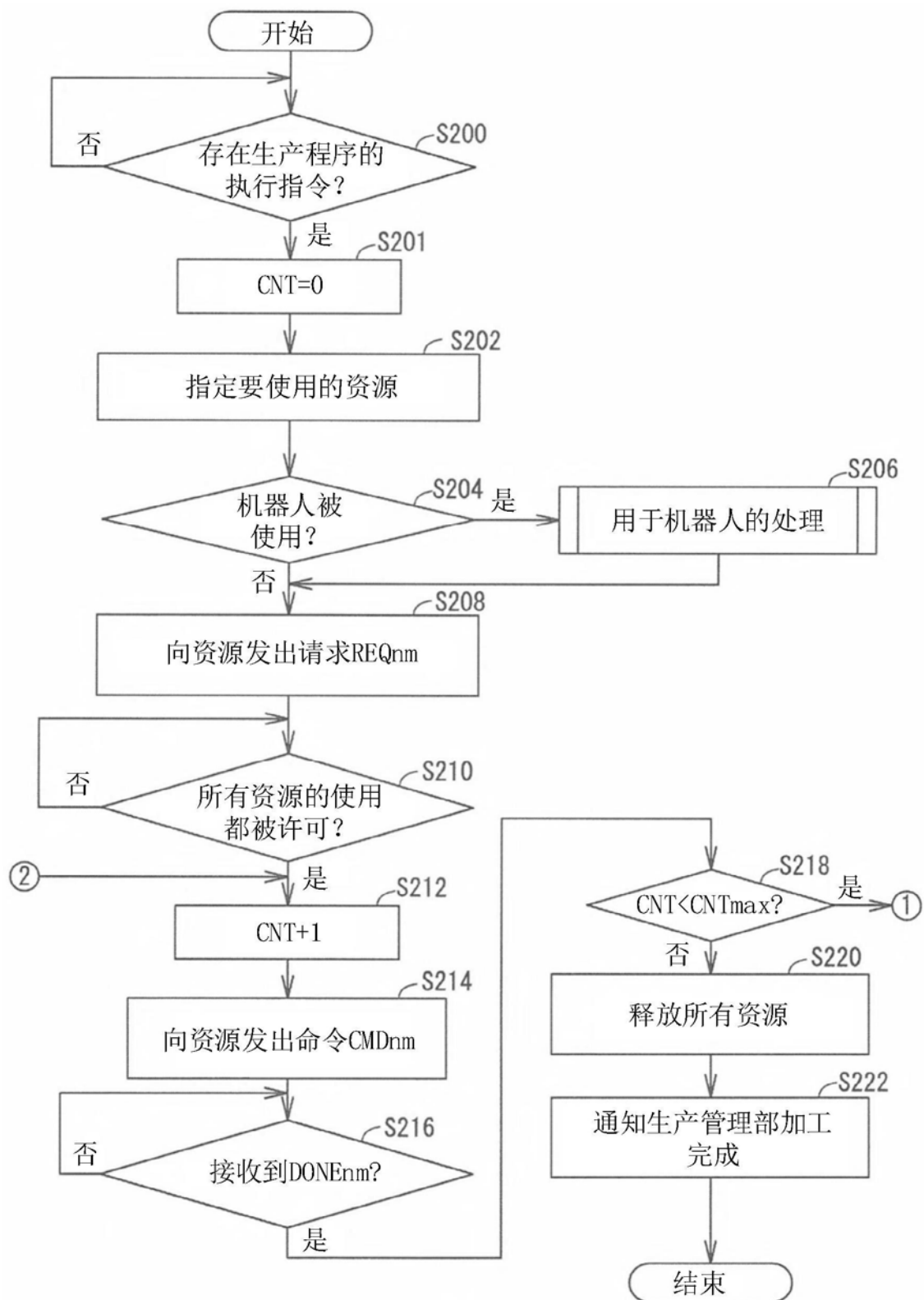


图7

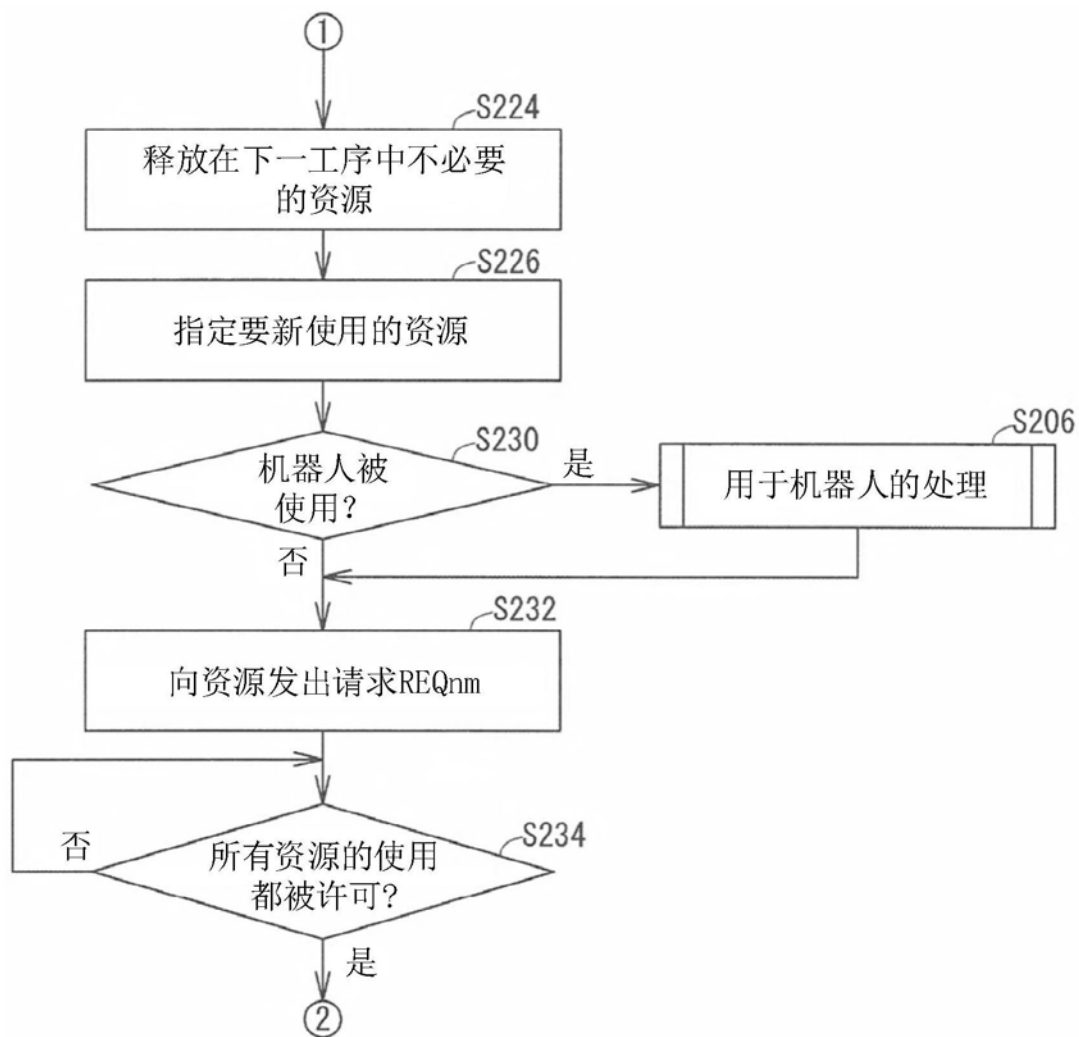


图8

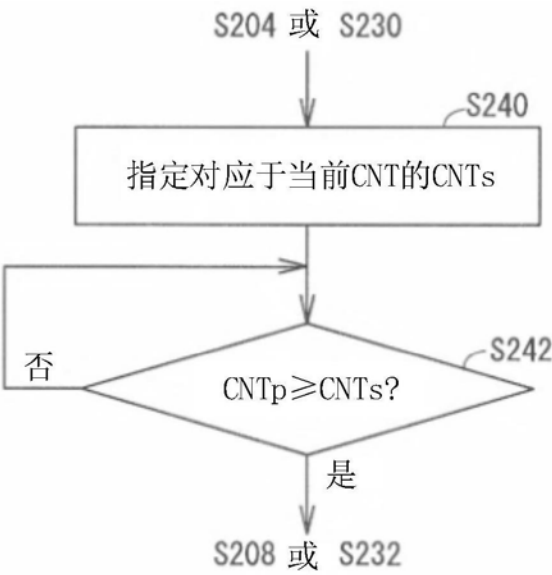


图9

工序		单		双	
工序 编号	工序名称	安装 部分	CNT s	安装 部分	CNT s
0	在搬入托盘上等待		8		1
1	材料的卸载	h1		h1	
2	材料尺寸的确认	h1		h1	
3	NC1安装准备	h1		h1	
4	NC1安装	h1		h1	
5	NC1的主轴加工				
6	在NC1的轴之间的递送				
7	NC1的副主轴加工		12		12
8	NC1移除准备	h1		h2	
9	NC1移除	h1		h2	
10	装载到临时放置台上	h1		h2	
11	在临时放置台处等待		14		14
12	附接至工件清洁器	h1		h2	
13	工件清洁		14		14
14	装载搬出托盘	h1		h2	
15	在搬出托盘上等待				

图10

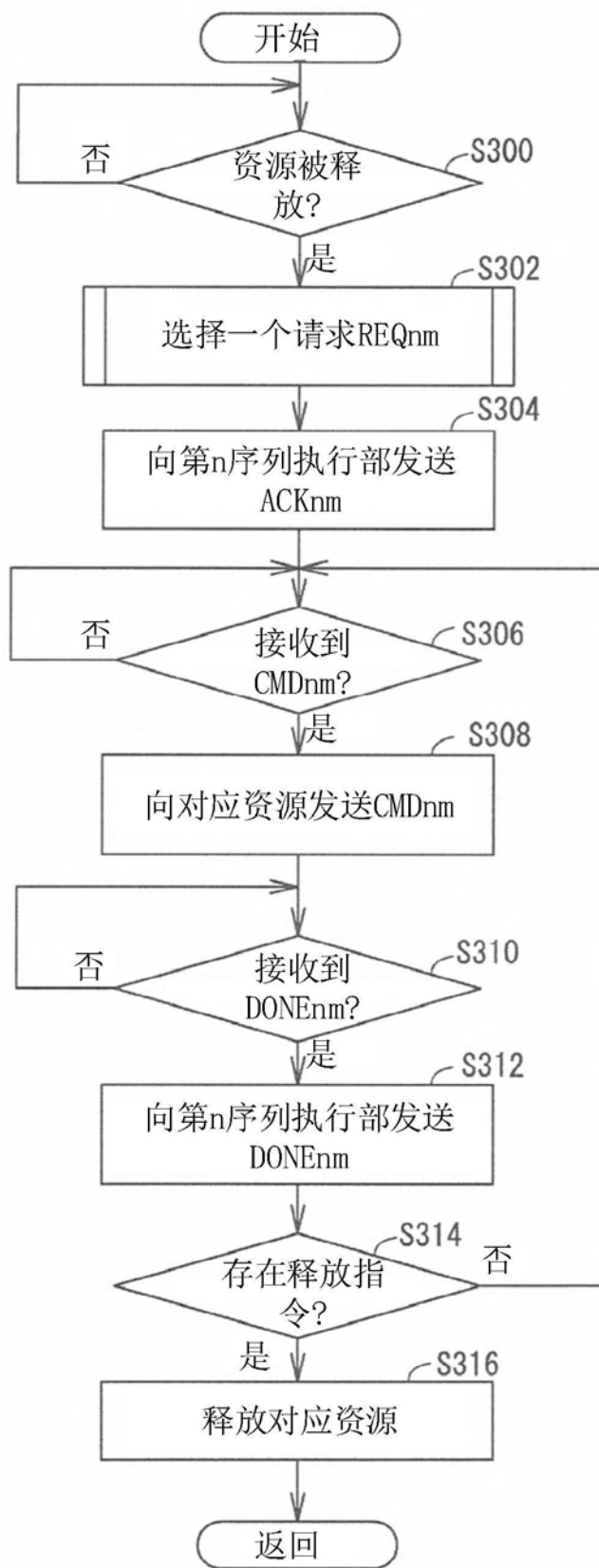


图11

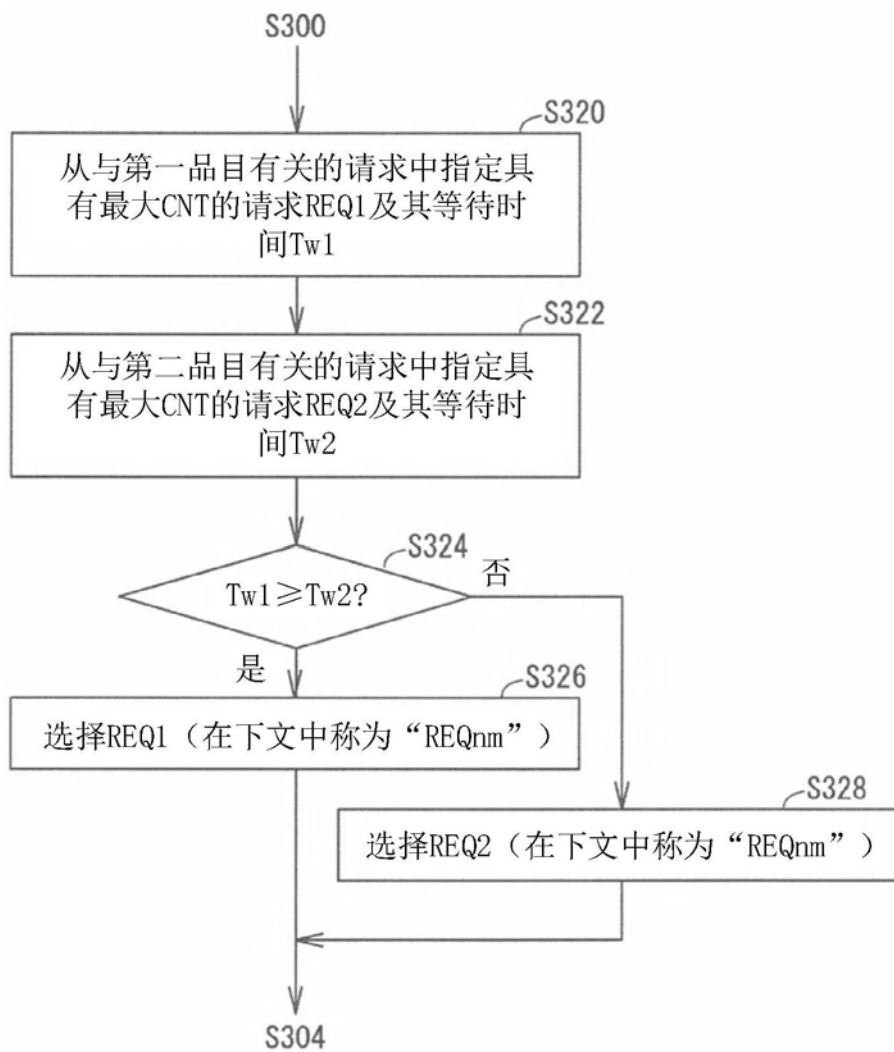


图12

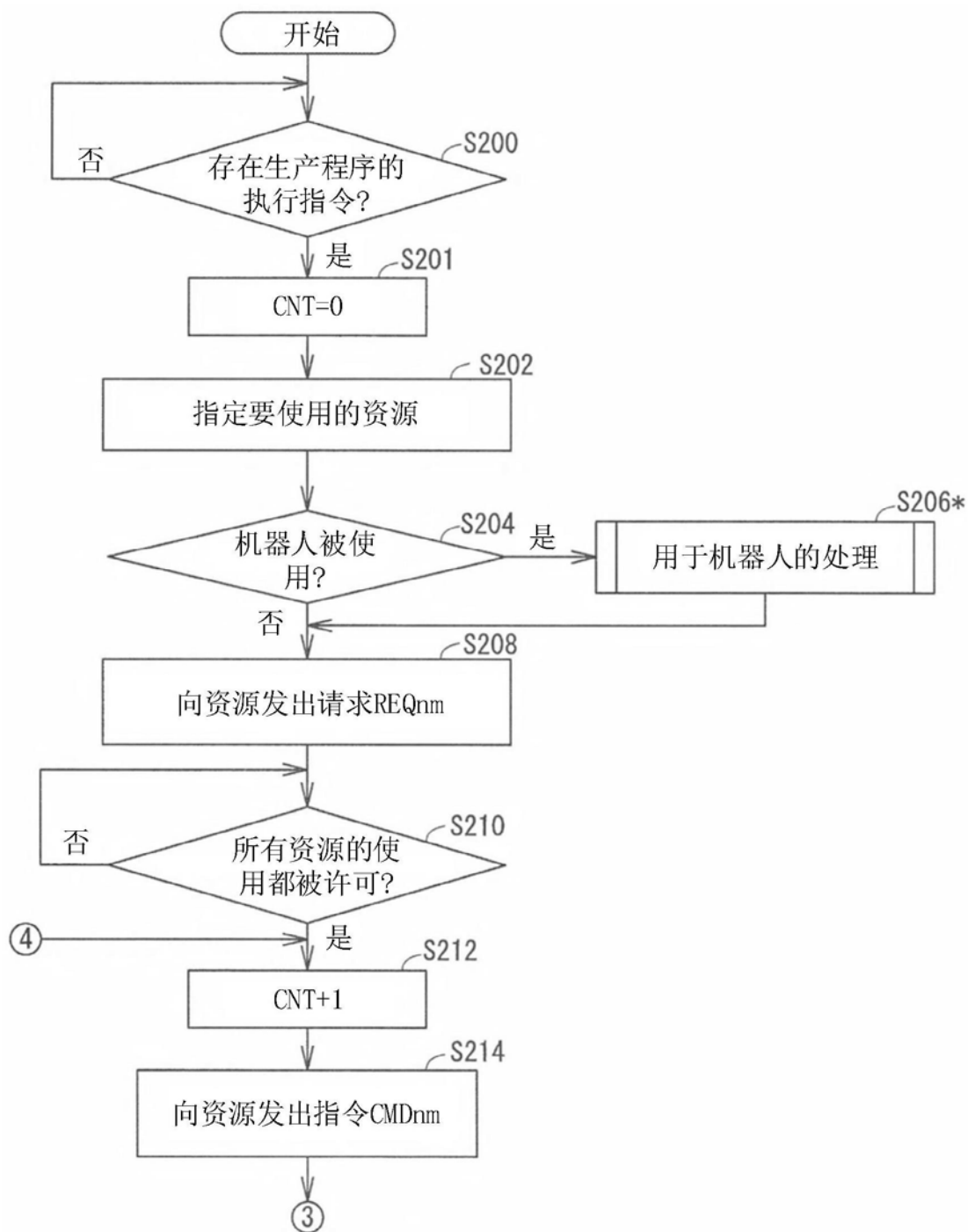


图13

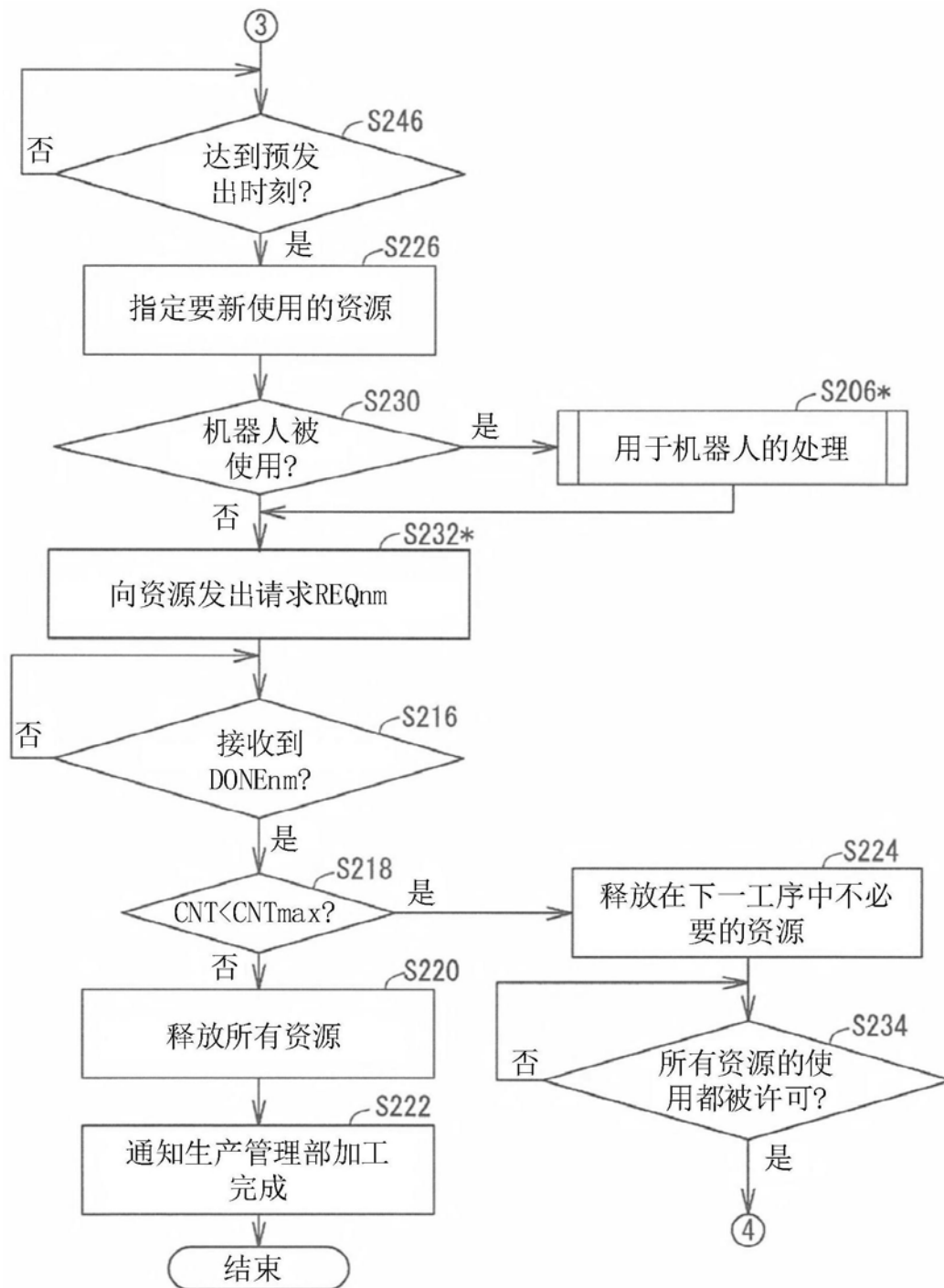


图14

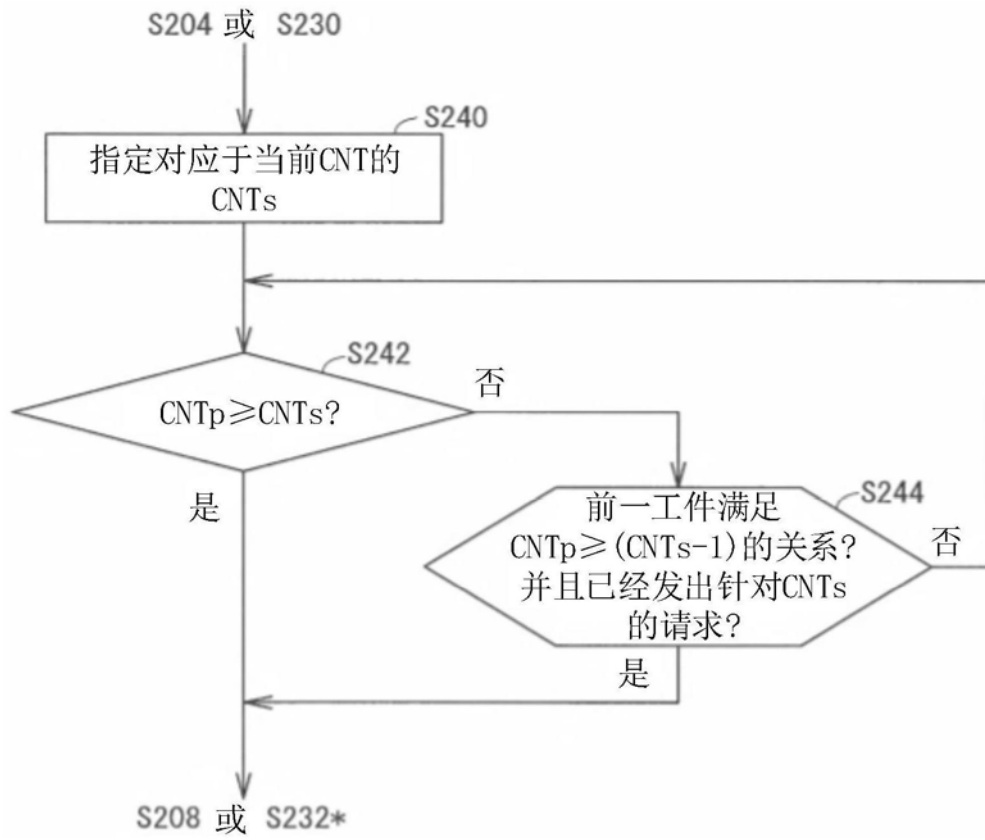


图15

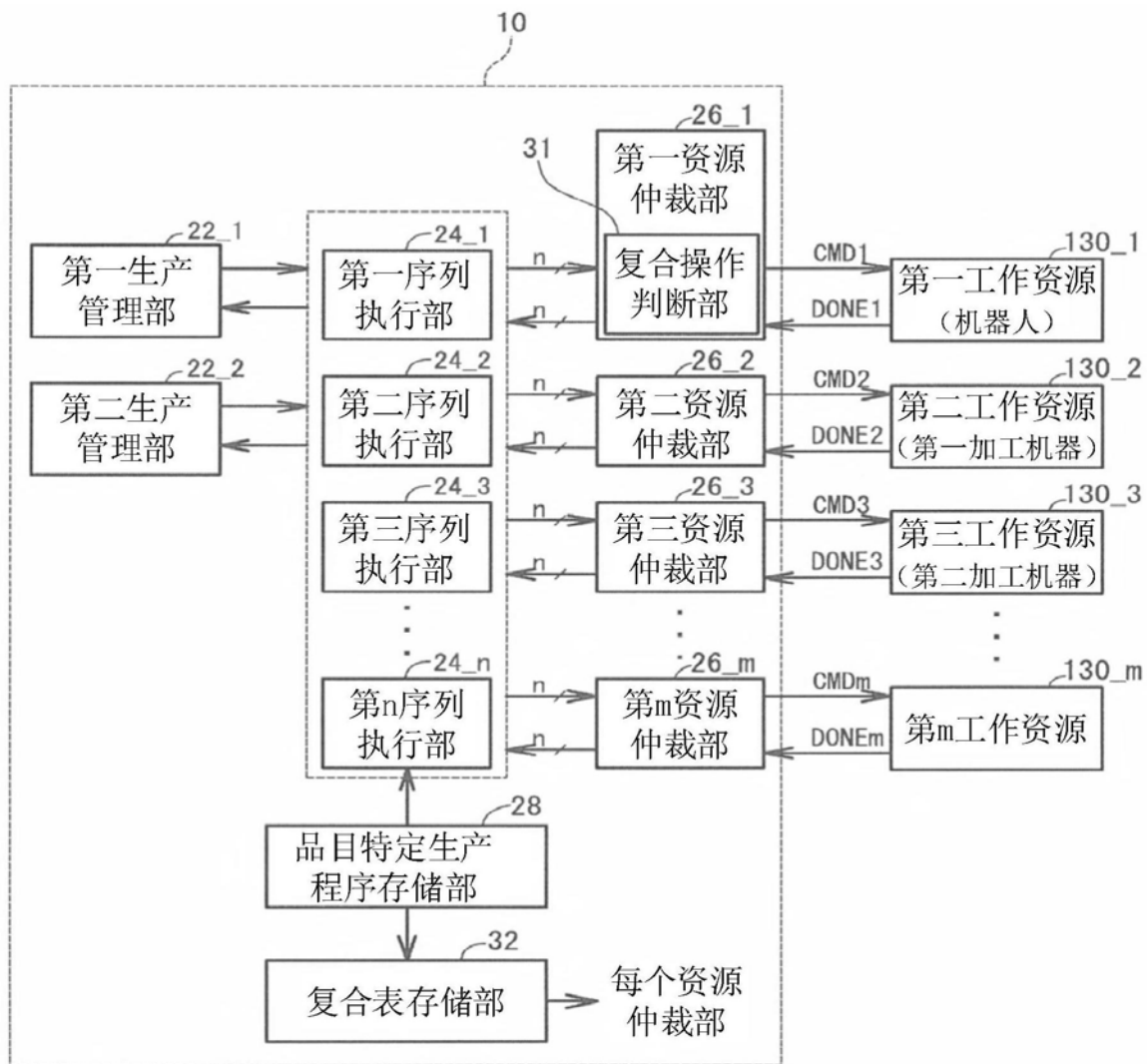


图16

后一工件		前一工件		资源		复合程序	
工序 编号	工序名称	工序 编号	工序名称	NC	机器人	NC	机器人
3	NC1安装准备	8	NC1移除准备		R1		RPR003C-W01234
4	NC1安装	9	NC1移除	NC1	R1	NPR004C-W01234	RPR004C-W01234
10	装载到临时放置台上	12	附接至工件清洁器		R1		RPR010C-W01234
12	附接至工件清洁器	14	装载搬出托盘		R1		RPR012C-W01234
3	NC2安装准备	8	NC2移除准备		R1		RPR003C-W01235
4	NC2安装	9	NC2移除	NC2	R1	NPR004C-W01235	RPR004C-W01235
10	装载到临时放置台上	12	附接至工件清洁器		R1		RPR010C-W01235
12	附接至工件清洁器	14	装载搬出托盘		R1		RPR012C-W01235

图17

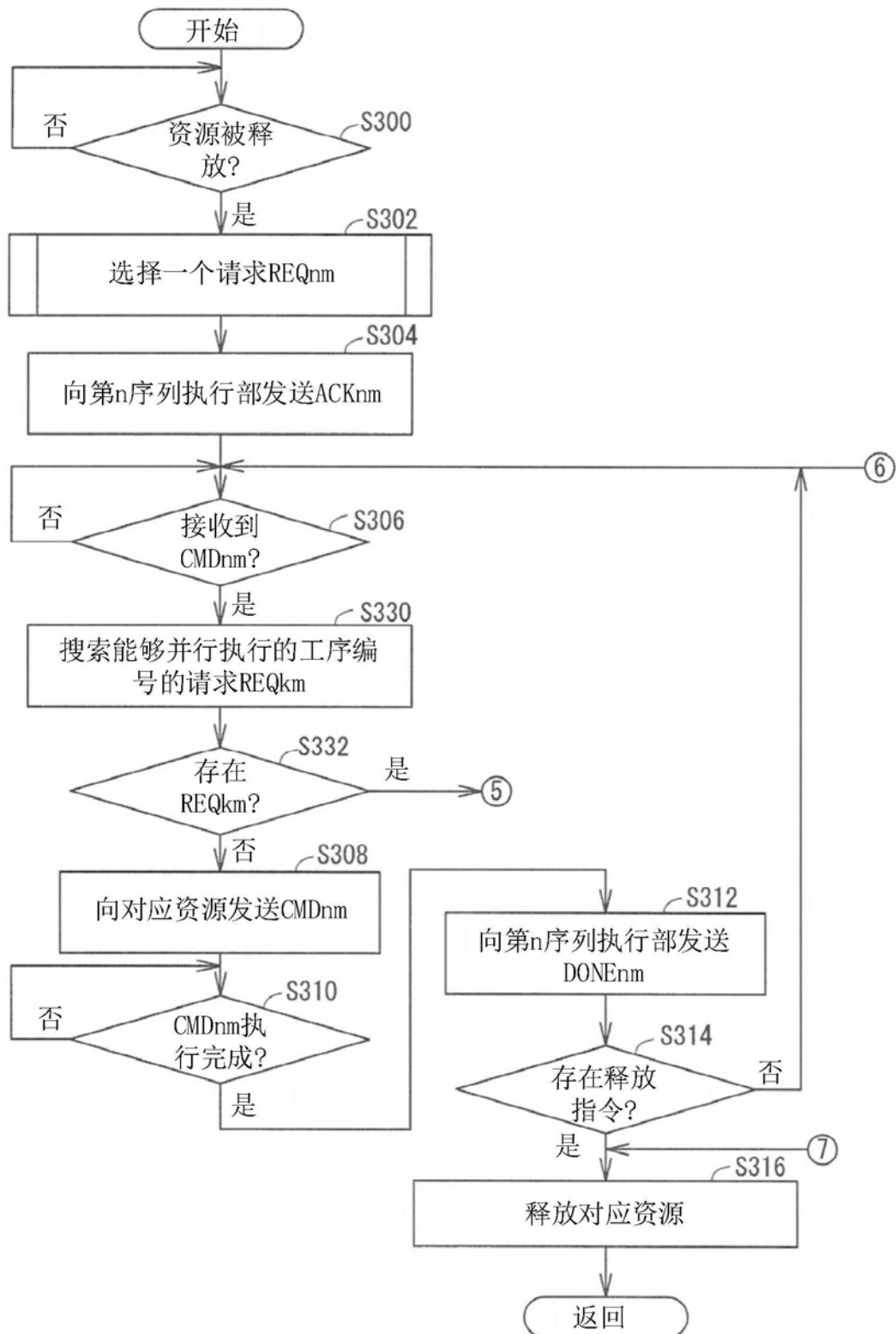


图18

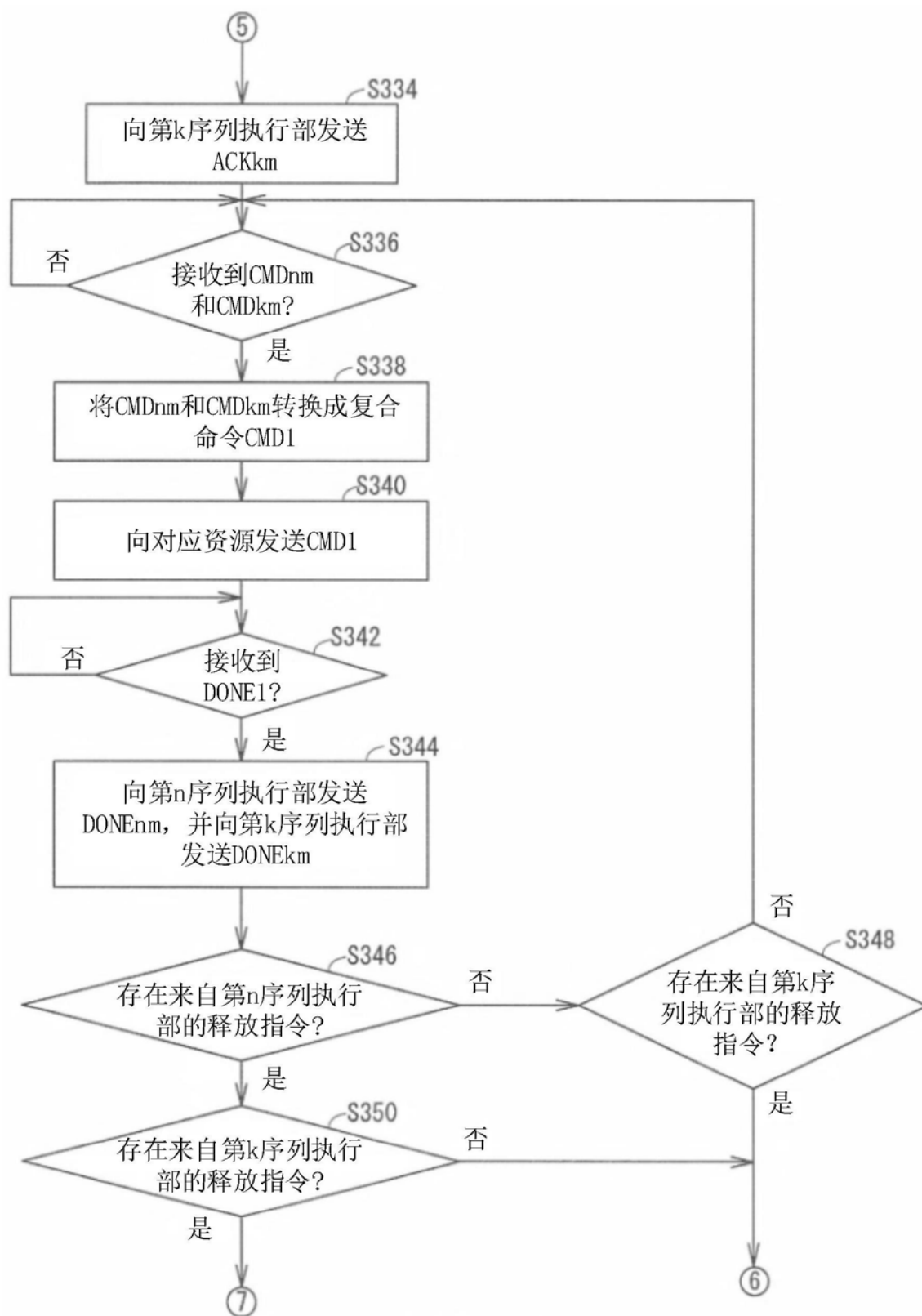


图19

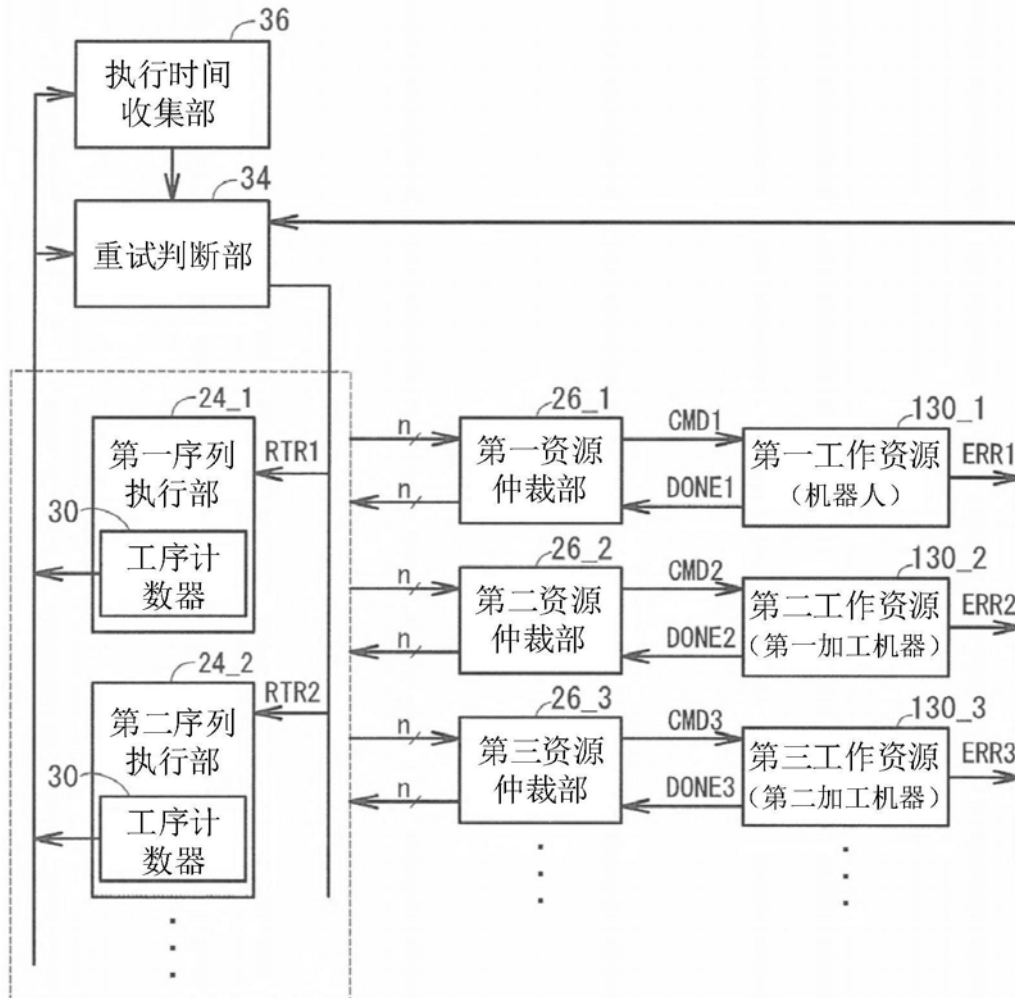


图23

工序		资源		资源程序		重试点
工序编号	工序名称	NC	机器人	NC	机器人	
0	在搬入托盘上等待					
1	材料的卸载		R1		RPR001-W01234	1
2	材料尺寸的确认		R1		RPR002-W01234	1
3	NC1安装准备		R1		RPR003-W01234	
4	NC1安装	NC1	R1	NPR004-W01234	RPR004-W01234	1
5	NC1的主轴加工	NC1		NPR005-W01234		
6	在NC1的轴之间的递送	NC1		NPR006-W01234		1
7	NC1的副主轴加工	NC1		NPR007-W01234		
8	NC1移除准备	NC1	R1	NPR008-W01234	RPR008-W01234	1

图24

工序		资源		资源程序		替代进程
工序编号	工序名称	NC	机器人	NC	机器人	
0004	NC1安装	NC1	R1	NPR004-W01234	RPR004-W01234	100400
0005	NC1的主轴加工	NC1		NPR005-W01234		
0006	在NC1的轴之间的递送	NC1		NPR006-W01234		100600
0007	NC1的副主轴加工	NC1		NPR007-W01234		
0008	NC1移除准备	NC1	R1	NPR008-W01234	RPR008-W01234	100800
100800	NC1移除诊断操作	NC1	R1	NRT100800	RRT100800	
100810	情况1 机器内清洁, 返回	NC1	R1	NRT100810	RRT100810	
100820	情况2 迅速移除, 工件排空	NC1	R1	NRT100820	PRT100820	
100821	手替换, 迅速移除	NC1	R1	NRT100821	PRT100821	
100822	手/工件的返回	NC1	R1	NRT100822	PRT100822	

图25

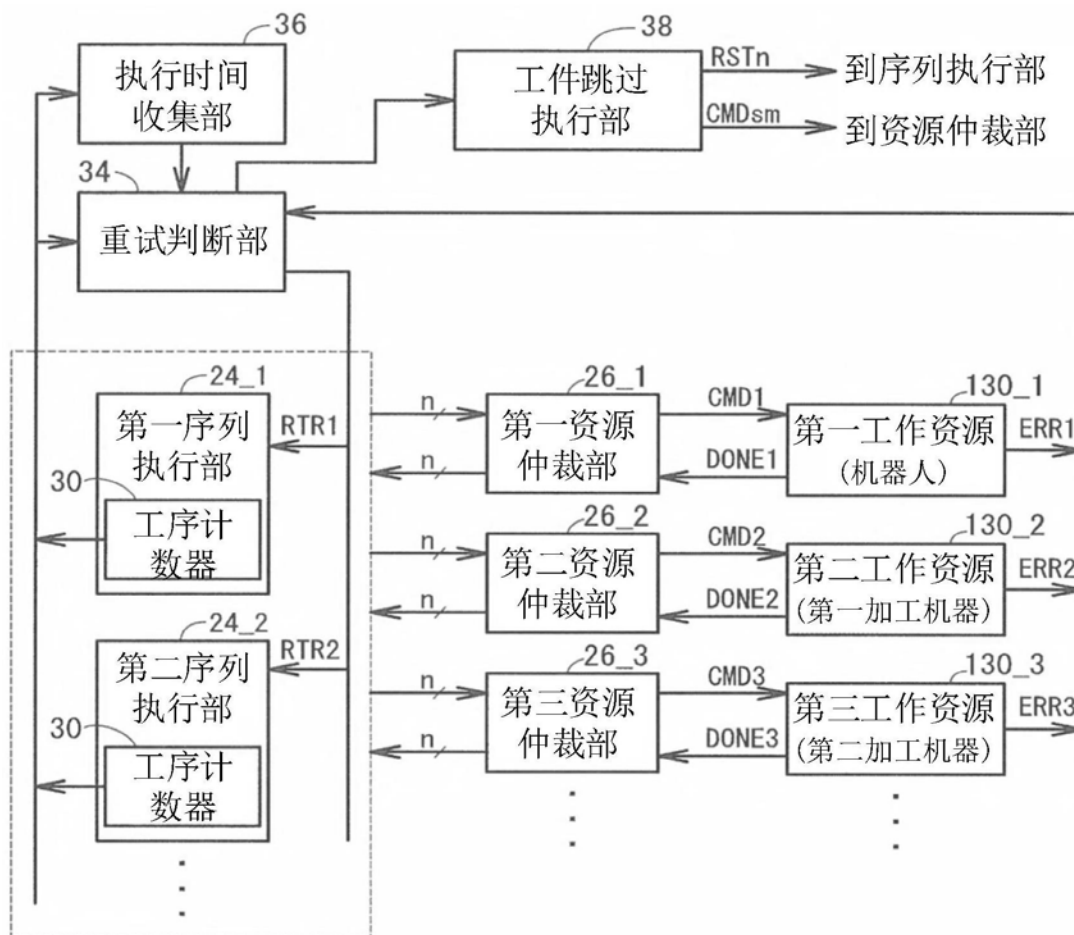


图26