

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

B65G 37/00 (2006.01)

B65G 43/10 (2006.01)



[12] 发明专利说明书

专利号 ZL 99816051.2

[45] 授权公告日 2006 年 3 月 29 日

[11] 授权公告号 CN 1247423C

[22] 申请日 1999.12.14 [21] 申请号 99816051.2

[30] 优先权

[32] 1998.12.14 [33] US [31] 09/212,002

[86] 国际申请 PCT/US1999/029692 1999.12.14

[87] 国际公布 WO2000/035782 英 2000.6.22

[85] 进入国家阶段日期 2001.8.6

[71] 专利权人 阿赛斯特技术公司

地址 美国加利福尼亚

[72] 发明人 布莱恩·维伦 克里福德·霍尔登

审查员 张京德

[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所

代理人 王以平

权利要求书 2 页 说明书 94 页 附图 50 页

[54] 发明名称

控制在材料输送系统中被输送的材料的运动的
分布式方法

[57] 摘要

本申请提供了一种用于控制在材料输送系统中被输送的材料的运动的分布式方法，包括规定一个包括构成所述材料输送系统的机电装置的邻接的子组的邻区；提供低级控制器用于控制机电装置子组，并接收来自机电装置的状态信息；以及构成每个区线程，用于当材料进入各个轨道区时确定使用所述状态信息；由被邻区更新的存储的信息确定上游区线程和材料进入各个轨道区的入口速度；向各个轨道区发出电动机控制指令以便按照由上游区线程输入的速度曲线信息和入口速度确定材料的速度；由存储的被相邻的下游区更新的信息确定材料进入相邻的下游区的速度；由可能的进入速度和材料的目的位置确定材料在一个或几个相邻的下游区的速度曲线；向一个或几个相邻的下游区发送速

度曲线信息以便执行所述速度曲线。

1. 一种用于控制在材料输送系统中被输送的材料的运动的分布式方法，包括：

规定一个包括构成所述材料输送系统的机电装置的邻接的子组的邻区，所述机电装置包括轨道区，其被构成用于控制材料的运动并报告区状态信息；

提供低级控制器用于控制机电装置子组，并接收来自机电装置的状态信息，所述低级控制器包括区控制器，每个区控制器被构成用于控制并接收来自各个轨道区的状态信息，并且接收来自邻区中其它区线程的信息，以及

构成每个区线程，用于：

当材料进入各个轨道区时确定使用所述状态信息；

由被邻区更新的存储的信息确定上游区线程和材料进入各个轨道区的入口速度；

向各个轨道区发出电动机控制指令，以便按照由上游区线程输入的速度曲线信息和入口速度确定材料的速度；

由存储的被相邻的下游区更新的信息确定材料进入相邻的下游区的速度；

由可能的进入速度和材料的目的地位置确定材料在一个或几个相邻的下游区的速度曲线；

向一个或几个相邻的下游区发送速度曲线信息，以便执行所述速度曲线。

2. 如权利要求 1 所述的分布式方法，还包括：

配置每个区线程，以便当材料离开各个区时向邻区发送离开信息。

3. 如权利要求 2 所述的分布式方法，其中所述被发送给离开信息的邻区包括这些邻区，其速度曲线的执行将受在所述邻区内的材料的位置的影响。

4. 如权利要求 1 所述的分布式方法，其中速度曲线信息包括：
开始速度索引，表示材料在开始执行速度曲线时的速度；以及
结束速度索引，表示在速度曲线的执行结束时材料的速度；
所述速度索引和轨道区使材料加速到达的实际电动机的速度有关，以便完成所需的速度曲线。

5. 如权利要求 1 所述的分布式方法，其中速度曲线从加速、减速或恒速速度曲线中选择。

6. 如权利要求 1 所述的分布式方法，其中存储的信息包括速度表，其中列有每个相邻下游区的特定的材料单元可以离开所述邻区的最大速度。

7. 如权利要求 1 所述的分布式方法，其中最大速度可被设置为表示区不可用于材料运动的第一个值，表示区被保留以便用于材料运动的第二个值，或者表示实际的材料速度的第三个值。

控制在材料输送系统中被输送的 材料的运动的分布式方法

技术领域

本发明一般涉及制造控制系统,尤其涉及用于材料输送系统的控制系统。

背景技术

自动输送系统的许多应用用于输送材料。一般利用用于控制材料流的自动设备把材料装在输送装置上。也使用自动设备在出口点除去材料,输送装置与/或搬移设备被设计使得允许几种物品积累在接触点附近,同时阻止相邻材料装置之间的冲突。在包括半导体处理在内的一些应用中,在沿着运送路径的一个或几个位置,必须暂时把材料从输送装置移动到工作台。然后材料被返回输送装置,输送装置然后把材料输送到下一个工作台或出口点。沿着路径在输送装置和工作台之间的移动材料的工作可能是复杂的,因为必须注意确保转移被完成,而又不使在输送装置上的材料流有大的中断。需要一种不影响输送装置的材料流而能够有效和方便地在输送装置系统和工作台之间传递材料的系统。

材料在中间位置从输送装置上被除去的领域的一个例子是半导体处理领域。在这个领域中,可以使用输送装置把半导体晶片或其它衬底输送给几个不同的处理机,或者把原版(reticle)从储料器输送到分档器。材料(即晶片或原版)必须被传递给机器进行处理,并且在处理完成之后,被返回输送装置,以便交付给下一个处理机。材料一般被保持在有保护设施的容器内,例如密封的容器,以便使得衬底最少地暴露在处理机外部的环境中,并保护材料不受微粒的污染。每个处理机的入口具有负载端口,其被设计用于在保护环境下从输送容器中自动地卸下材料。在设备的操作期间,必须频繁地

使材料在负载端口和输送装置之间运动。

一般地说，半导体制造设备具有多个舱室，每个舱室包括几个处理机。使用各种系统（被称为内部舱室输送系统）使材料在一个内部舱室的机器之间运动。例如，许多系统依靠人工使用小车把材料从一个端口传递到另一个端口。工人一般通过启动一种手动的机器人关节或其它的升高装置使材料向端口运动，并且在完成处理之后，把输送容器返回小车上。在下一个机器上重复所述的处理。内部舱室输送的另一种系统依靠自动导向的车辆（AGV），其在机器之间传递所述的容器，并使容器自动地向端口运动。小车和 AVG 没有和自动化的输送装置相关的优点，自动化的输送装置可以高效率地快速地沿着输送路径输送材料，并具有比小车和 AGV 高得多的容量。

半导体晶片是一种精密材料，特别是在后面几级的处理中，并且十分有价值。通过在半导体晶片或其它衬底上形成多层来制造集成电路。随着技术的发展，集成电路越来越复杂，并且一般包括多层复杂的布线。由于集成电路尺寸的减小，位于一个晶片上的集成电路的数量也增加。在未来的几年内，半导体晶片的标准尺寸将从 200mm 增加到 300mm，这将进一步增加可以在一个晶片上形成的集成电路的数量。随着集成电路的复杂性的增加和尺寸的减少，随着晶片通过各个处理阶段，半导体晶片的价值也增加。此外，300mm 晶片的容器使得重量增加，这在手动晶片处理中将产生人机工程学问题。因而，在处理半导体晶片时必须相当小心，尤其是在较后的处理阶段期间，因为引起晶片破坏将产生可观的经济损失。对于晶片的处理，要求清洁的室内环境，基本上没有微粒污染，对可用于输送材料的系统提出了进一步的限制。需要一种适用于在清洁的室内环境下操作的用于在输送装置和负载端口之间传递材料的系统。

需要一种用于在输送装置和负载端口或者工作台之间输送含有半导体晶片的容器或其它容器的传递系统，用于移动材料，例如半导体晶片。还需要一种可以用于半导体处理之外的领域的传递系

统，例如包括但不限于医药医疗系统，平面显示器和计算机硬件，例如盘驱动系统，调制解调器等。

在基于输送装置的传递系统中材料的运动通常由自动控制系统（ACS）管理。例如，在丹罗国际机场的行李处理系统中使用一种这样的系统。美国的邮政业务在 Carol Stream, IL 中的处理和分配中心使用另一个这种系统控制邮件的传送（关于更详细的信息可参阅“U.S.Postal Facility Improves Operation with Honeywell's Smart Distributed System” <http://www.honeywell.com/sensing/pressrel/9718.stm>）。在半导体制造操作中使用的基于至少一个输送装置的传递系统中已经使用过 ACS 管理晶片容器的运动。

和用于邮政以及处理行李的例子相比，在半导体制造操作中使用的基于输送装置的传递系统的 ACS 必须确保被传递的晶片容器不会碰撞，并且不会受到过大的加速度。此外，ACS 必须确保准时地把晶片容器从一个处理点交给另一个处理点。图 1 表示现有技术中的一种这样的系统，即“Asyst 自动控制系统”，其中使用一种在舱室之间或者在内部舱室传递晶片容器或开口的晶片盒的传递系统。这种传递系统包括：供材料在其上运动的轨道，导控器，这是一种机电装置，提供在轨道部分之间的转动，使之满足一定的角度，以及升降装置，这也是一种机电装置，用于使容器升降。所述轨道包括用于移动材料的若干个电动机和用于检测材料的位置的检测器。

参见图 1，其中所示的自动控制系统包括多个 PLC（可编程逻辑控制器），每个 PLC 按照系统目标控制传递系统的各个区域中的一个或几个容器的运动。每个 PLC 通过 ProfiBus 总线和构成轨道的各个区域的检测器以及电动机相连。所述 ProfiBus 是检测器总线，即只用于在灵巧控制器（PLC）和客户（电动机、检测器导向器、升降器等）传递信号而没有独立性。PLC 被相互连接，使得它们能够共享关于容器运动和位置的信息。可以由 PLC 控制的检测器和电动机的数量是有限的。这是因为 PLC 是一种以扫描方式工作的轮询装置。在每次扫描时，PLC 读取每个相关的检测器。因此，检测器越

多，扫描时间越长，每秒的扫描次数较少，因而使得系统的响应较慢。这种结构的另一个问题是，PLC 必须知道其每个相关装置的控制接口。结果，当新的检测器或电动机接口被加入传递系统中时，需要修改 PLC。另一个问题是，PLC 同时和高电平出口以及低电平出口有关，例如使容器向其目的地运动而不发生碰撞，以及使电动机加速。结果，PLC 的计算功率成为传递系统的性能的关键因素。这两个问题都妨碍传递系统的可量测性和重构。

因此，需要一种输送系统的 ACS，其是可升级的，能够高效地使用计算机资源，使得高级的和低级的控制操作不会冲突，并且容易支持新型的电动机、机电元件和检测器。

发明内容

本发明提供一种用于控制在材料输送系统中被输送的材料的运动的分布式方法，包括：规定一个包括构成所述材料输送系统的机电装置的邻接的子组的邻区，所述机电装置包括轨道区，其被构成用于控制材料的运动并报告区状态信息；提供低级控制器用于控制机电装置的子组，并接收来自机电装置的状态信息，所述低级控制器包括区控制器，每个区控制器被构成用于控制并接收来自各个轨道区的状态信息，并且接收来自邻区中其它区线程的信息，以及构成每个区线程，用于：当材料进入各个轨道区时确定使用所述状态信息；由被邻区更新的存储的信息确定上游区线程和材料进入各个轨道区的入口速度；向各个轨道区发出电动机控制指令，以便按照由上游区线程输入的速度曲线信息和入口速度确定材料的速度；由存储的被相邻的下游区更新的信息确定材料进入相邻的下游区的速度；由可能的进入速度和材料的目的地位置确定材料在一个或几个相邻的下游区的速度曲线；向一个或几个相邻的下游区发送速度曲线信息，以便执行所述速度曲线。

概括地说，本发明是一种控制系统体系结构和用于实现能够满足上述要求的材料输送系统的方法。本发明包括三级控制器。高级（输送）控制器和外部指令系统相连，所述指令系统用于向本控制系统发出表示材料要如何被移动的控制指令。例如，优选实施例中，该实施例实施的是一种用于半导体制造设备中的材料输送系统，这些控制指令包括把一个特定的材料容器从一个处理点移动到另一个处理点的指令。输送控制器（TC）通过排序执行控制指令的一系列基本操作执行控制指令。在一个实施例中，TC 通过把控制指令分解为

由至少一个第二级控制器（控制逻辑计算机，或 CLC）执行的一系列基本的获取、移动和停放指令实现上述功能。

在分布式控制系统中，TC 是知道整个材料输送系统的物理拓扑的唯一的系统入口。由 TC 存储的拓扑的一种表示被组织在一组所有可能的系统目的地和输送系统区的周围。每个目的地包括位置参考和与容器可以被预负载并离开所述目的地被发出的区有关的装置信息。TC 还使用由 CLC 返回的信息保持输送系统的状态信息。

每个控制逻辑计算机（CLC）提供高级的实时的控制，并通过对一组第三级控制器（智能驱动器）提供指令，协调物理输送系统的不同的区域，每个第三级控制器还负责在 CLC 的控制区域中的一个或几个机电装置的低级控制。例如，在优选实施例中，一个区域可以包括 64 个区，每个区包括一组检测器（例如 2 个），一定长度的轨道（例如 5m）和驱动电动机（例如 1 个）。根据来自检测器的信息、区域的拓扑信息和速度控制规则、路由以及避免碰撞，CLC 通过向下对其智能驱动器发出电动机控制指令执行基本指令。

在优选实施例中，具有不同类型的机电装置。每种类型由一种或几种智能驱动器控制。例如，在用于半导体制造设备的优选实施例中，机电装置可以包括区（一段输送机轨道以及与其相关的检测器和电动机，标签/条码阅读器，负载端口转移装置（LPTD），EMO（事故停车）检测器和导向器（一种能够转动的轨道装置）。因而，优选实施例包括以下形式的智能驱动器：

区控制器向 CLC 报告检测器数据，使电动机按照来自 CLC 的指令加速和减速；

标签控制器利用标签/条码阅读器提供双向通信；

轴控制器控制 LPTD 的任意轴，控制升降装置或导向器的转动；

EMO 控制器监视电源，并在电源故障时通知 CLC；以及

握手控制器执行多线程并联接口，用于同步单独控制的机电装置的操作。

每个区和相邻的 n 个上游区、 m 个下游区的邻域相关，在正常的输送系统操作期间，这些区能够发生相互作用。根据区在输送系统的拓扑内的位置，这些邻域被不同地定义。例如，在直的输送轨道内的区可以具有包括 3 个上游和 3 个下游节点的邻域，因为区和区之间的相互作用被限制在直的轨道内。在 3 个轨部分的交叉处的导向器将被多个区的邻域（例如 20 个区）覆盖，反映可以相互作用的区

的较宽的范围。CLC 把执行只影响特定邻域的指令的责任分配给负责控制和包括所述邻域的区有关的机电装置的 CLC 线程(或程序)。这些线程被配置用于在它们当中进行通信,以便协调它们的作用。把 CLC 的责任分配到区级使得 CLC 的任务能够在多个处理器的不同的处理器当中被分配。此外,全部 CLC 可以在一个处理器上运行,或者在分布在网络上的多个处理器上运行。

每个 CLC 从其相关的智能驱动器采集状态信息。CLC 向 TC 报告状态信息。CLC 还使用状态信息检测和处理影响其相关装置的报告的错误状态。这些错误状态包括:检测器故障,电动机故障,故障的负载端口,未期望的卸载等。每个控制层除去发出指令之外,还向下一个较高的控制层返回状态信息。下一个较高的层则负责所述状态信息形成策略。

附图说明

通过结合附图阅读下面的说明书和所附的权利要求书可以更清楚地理解本发明的其它目的和优点,其中:

图 1 是现有技术的用于半导体制造操作中用于在处理点之间输送晶片容器的传递系统的 ACS 的方块图;

图 2A 是输送系统的示意图,表示在处理工具和负载区域之间的关系;

图 2B 是输送系统的示意图,表示在具有两个负载点的处理工具和相关的负载区域之间的关系;

图 2C 是具有不和一个区域对准的负载/卸载点的输送系统的区域的示意图,表示和在输送器区中的检测器相关的负载区域的可能的部分;

图 3A - 3E 分别表示轨道区、拐弯处、导向器附近的轨道区,升降器和导向器的邻近;

图 4 是优选实施例的方块图,表示分级的 ACS 元件和本发明的 ACS 在常规的半导体制造控制系统中的位置;

图 5 是表示在按照本发明的用于半导体晶片传递系统中实施的 ACS 的元件之间的相互连接的方块图;

图 5B 是表示图 5 所示的机电装置、CAN 总线程和控制逻辑计算机的附加的细节的方块图;

图 6 是包括按照本发明实施的 ASC 的元件的半导体制造设备的示意图;

图 7 是具有包括按照本发明实施的 ASC 的元件的连接的舱室的半导体制造设备的示意图；

图 8 是具有按照本发明实施的 ASC 的元件的单独的舱室的半导体制造设备的示意图；

图 9A 是实施本发明的实施例的输送控制器功能的计算机的方块图；

图 9B 是实施本发明的实施例的控制逻辑计算机功能的方块图；

图 9C 是实施本发明的实施例的区控制器功能的方块图；

图 9D 是实施本发明的实施例的轴向控制器功能的方块图；

图 9E 是实施本发明的实施例的 ID 控制器功能的方块图；

图 9F 是实施本发明的实施例的紧急停车 (EMO) 控制器功能的方块图；

图 9G 是实施本发明的实施例的握手控制器机功能的方块图；

图 10 构成由图 9A 的控制器保持的内部拓扑的对象方块图；

图 11 是包括可以利用按照本发明的 ACS 的实施例控制的单向环和双向轨的晶片传递系统的示意图；

图 12A 是由相应于图 11 的晶片传递系统的图 9A 的输送控制器保持的内部拓扑的方块图；

图 12B 表示在图 12A 中使用的主要连接；

图 13 是和区线程 512 相关的方法和数据结构的方块图

图 14 是启动和协调运动操作的 TC 和 CLC 的软件成分的逻辑图；

图 15 是发出包括运动操作和其它的输送系统的操作的子指令的 CLC 和智能驱动器的软件成分的逻辑图；

图 16 是在本发明的自动控制系统的主要软件成分之间的通信路径图；

图 17 是从高级点观看的传递的基本操作的事件图；

图 18 - 19 是由不同的系统元件看的获取基本操作的事件图 (图 19 是针对特定传递装置的实施例, 和其它的机电系统不同)；

图 20, 21A - 21C, 22A - 22B 是从不同系统元件看的移动基本操作的事件图;

图 23A, 23B, 24A, 24B 是从不同系统元件看的停放基本操作的事件图;

图 25 是用于当材料到达不是负载区的轨道区 (轨道区可以是一个暂时的停止点) 时发出的信息的 UML 时序图;

图 26 是表示 300mm 晶片容器在区 Z2 附近向左轨运动时检测器信号转变的时序图 (200mm 的晶片容器的转变是不同的);

图 27 是表示当单个容器 X 从区 ZI 向区 ZA 左移时区和容器速度曲线之间发送的消息的时序图;

图 28 是表示当两个容器 X 和 Y 分别从区 ZE 和 ZJ 向左向区 ZZ 和 ZD 运动时在区之间发出的信息的时序图;

图 29 是表示当两个容器 X 和 Y 分别从区 ZI 和 ZJ 向左向区 ZZ 和 ZF 运动时在区之间发出的信息的时序图;

图 30 是说明在区线程中的区动态方式/状态机的状态图;

图 31 是包括导向器 Q、两个交叉的轨道以及两个分别需要通过导向器沿方向 R1、R2 运动的材料装置 P1、P2 的输送系统的区域的示意图;

图 32A, 32B 是表示对于图 31 所示的情况在区和导向器之间发出的信息的时序图;

图 33A, 33B 表示可能的导向器组配置;

图 33C - 33D 表示可以发生在图 33A, 33B 的导向器组配置中的可能的死锁情况;

图 34 是输送系统区域的示意图, 说明在导向器控制线程中应用的路径交付原理;

图 35 是输送系统区域的示意图, 说明考虑上游区的连通性的路径交付原理;

图 36 表示图 34 的物理布局, 具有附加的新的检测器 D9 和相关的轨道;

图 37 是输送系统的区域的示意图，说明考虑故障的输送系统节点的路径交付原理；以及

图 38 是输送系统的区域的示意图，说明考虑故障导向器的路径交付原理。

具体实施方式

下面参照几个特定的实施例说明本发明。本说明使用的术语的意义如下：

A. 术语表

术语	定义
----	----

AMHS	自动材料处理系统
------	----------

CAN	控制器区域网。包括装置在内的标准网络。
-----	---------------------

搬运器	容器的同义词
-----	--------

CIM	计算机集中制造。在本文中泛指所有的计算机系统，输送系统和这些系统之间进行通信。
-----	---

容器	一般用于指敞开的盒子、箱子或吊舱。容器是输送系统输送的对象。
----	--------------------------------

控制逻辑计算机	用于中间层次的软件成分的硬件平台。
---------	-------------------

CORBA	公共对象请求代理结构，由 OMG 研制的一种标准，作为分布式软件应用的一种方法在分级网络上共同操作。
-------	--

拐角	拐角是一种导向器，其唯一的功能是使搬运器拐弯（改变 90 度的方向）。拐角具有一个轴向控制器，并且是一种可配置用于左右转的软件。一个拐角可以具有多个输入（最多 3 个），但是只能具有一个输出。
----	--

导向器	导向器是一种用于使搬运器拐弯的机械装置，有两种类型的导向器，交叉导向器和拐角导向器。
-----	--

导向器控制器	一种在 CLC 上运行的软件应用程序，其负责控制包括导向器在内的元件。
--------	-------------------------------------

智能驱动器板	一种硬件平台，包括本地微控制器和用于监视和控制装置的网络支持（例如 CAN 总线支持）。
--------	--

E23 一种半导体设备和材料国际标准。其中规定了多导线的并联握手，用于在两个装置之间进行材料的转移。

E23 主动伙伴 主动伙伴是一种在材料传递中的装置，其具有使材料进行物理移动的元件，负载端口传递装置是主动的，其服务的负载端口是被动的。容器升降器是被动的，升降式起重机是主动的。

E23 被动伙伴 被动伙伴是一种请求材料负载或卸载的装置。其没有任何实现传递的机构。见 E23 主动伙伴关于输送系统主动元件和被动元件的说明。

E 停车 紧急停车。E 停车是一种使所有电动机立即停止的操作。E 停车在物理上不从任何计算装置除去功率。

升降器 一种升降器是一种用于升高或降低平台的轨道元件。升降器用于在位于不同楼层的输送器之间移动容器，或者用于把容器从上限高度移动到操作者的高度（在此情况下一般被称为起重机）。

升降器控制器 一种在 CLC 上运行的软件应用程序，其负责控制包括升降器在内的元件。

入口速度 当过来的容器的前沿接触皮带时区电动机必须运行的速度。如果区上游到另一个节点以加速或减速曲线运行，则入口速度是在区速度和出口速度中间的值。见出口速度和区速度。

出口速度 当出去的容器的后沿和区脱离接触时区电动机必须运行的速度。见入口速度和区速度。下游区的区出口速度和区速度相同。

FOUP 前方开口的整体容器，指的是含有 300mm 的晶片的容器并在前方开口。

门止动件 一种用于阻止搬运器在输送结束时运动的装置。门止动件可以用于升降器上，使得当升降器不和特定的轨道对准时将阻止搬运器从区的上游从升降器上下降。门止动件也用于缓

冲器上。门止动件由区软件控制。

握手控制器 一种在并行的 I/O 板上运行的软件应用程序。其和另一种装置实现 SEM E23 连接（例如 LPTD 和负载端口）。

中间台 在地板的中心向下运行的输送系统环，用于在不同的台之间移动材料。

交点导向器 其允许送入的材料选择多个输出方向中的一个方向。所述导向器具有两个或三个输出方向，和一个或两个输入方向。

内部台 该术语一般用于和中间台相连的并在一个台内提供输送服务的输送系统。

发送区 指的是容器开始在轨道上运动的区。和下游负载区同义。

负载区域 负载区域包括上游负载区域、下游负载区域和预负载区域。上游和下游区包括一个负载点（即可以骑垮两个区的负载点）。预负载区域是这样一个区域，其中容器被保持着直到其向着负载点运动。是上游负载区还是下游负载区取决于容器运动的方向（容器总是从上游区向下游区运动）。

图 2A - 2C 示意地表示包括多个区 42 的轨道 40 的不同的区域和至少一个工具 46，其中一个工具可以被负载。每个图都表示出了下游和上游负载区 54，52 和与每个工具 46 的负载点 44 相关的预负载区 50。

图 2A 表示的情况是工具 46 具有一个负载点 44，和相关的负载区域 48。因为容器从右向左运动，所以预负载区 50 和上游负载区 52 在下游负载区 54 的右方。

图 2B 表示的情况是工具 46 具有两个负载点 46A，46B。每个负载点具有各自的负载区 48A，48B。因为负载点 46A，46B 和区 42 相邻，负载区 48A，48B 重叠。例如，负载区 48A 的下游负载区 54A 和负载区 48B 的上游负载区 52B 相同。

图 2C 表示负载点相对于负载区的可能的位置，并对于左右运动

识别预负载区、上游和下游负载区。该图不用于表示实际的物理配置。

每个负载点 44 和负载端口传递装置 (LPTD) 相关, 所述传递装置把容器从轨道 (上游负载区 52) 移向工具 46 进行处理, 并在处理完成之后送回轨道。此外, LPTD 可用于多个负载点。

组 (lot) 被按照逻辑和 (选择地) 物理分组的一组晶片。一些制造装置 (fab) 含有用于一个盒的组, 而另一些则包括多个盒。输送系统不会通过很多。

组 ID 人/机可达到的用于识别组的选择器。

微控制器 一种包含在输送器内的计算机模块, 用于控制机电系统。

MMS 维护管理系统

维护管理系统 一种负责采集和保持其它系统的维护数据的系统。

材料 在本文中指半导体晶片或者可以在输送系统中运动的其它物品。

MCS 材料控制系统

MES 制造执行系统

MOVE 一个 AMHS 运动指令, 其可以被输送系统外部的元件启动。

运动方向 材料在轨道上可以运动的两个方向中的一个方向。行进方向被定义为面向轨道的观察者看的方向。即向观察者的左方的运动是左运动, 向观察者右方的运动是右运动。

邻域 邻域是在一个给定区周围的区的集合, 并且其确定一个可能的物理通路。一个邻域包括对于一个给定的区域的 n 个上游区和 m 个下游区 (一般 $n=m$)。每个区是多个邻域的成员 (一般为 $n+m$ 个邻域)。拐角、升降器和导向器都具有 2 个邻域。例如, 升降器的两个邻域分别是对于其向上的位置和向下的位置。在导向器附近的轨道区的邻域根据直接通过导向器的路径确定。图 3A-3E 示出

了对于不同的轨道配置的可能的邻域定义。在这些配置中， $n=m$ 。

图 3A 表示包括一个邻域的区 42，其在位于直的轨道 40 上的中心区 60 附近。所述的区包括编号为 1 的邻域。

图 3B 表示包括拐角 64 的两个邻域的区 42，其编号为 1 或 2，表示其所属的邻域。

图 3C 表示对于中心区 60 的包括一个邻域的区 42，其位于导向器 66 附近的直的轨道 40 上。在这种情况下，只有一个邻域，因为假定通过导向器的路径是直的。

图 3D 表示对于升降器 68 的包括两个邻域的区 42，所述升降器连接上轨 40U 和下轨 40U。区 42 编号为 1 或 2，表示其所属的邻域。

图 3E 表示对于交叉导向器 66 的包括两个邻域的区 42，区 42 编号为 1 或 2，表示其所属的邻域。

RF 标记 一种包括在容器中的射频应答器，其可以用电子方式读写，并且包括用于识别物理容器的足够的信息。

ORB 对象请求代理程序

并行 I/O 板 一种支持 8 位最好 16 位的数字 I/O 的硬件平台。用于实现握手控制器的 SEMI E23 接口。

预负载区 预负载区是一个目的区。在预负载区被交付给工具的材料在被运送到工具负载点之前将停止。

RAM 可靠性、可利用性和可维护性。一般指报告的 Semi E-10 组。

SEMI E-10 半导体设备制造者协会。规范 E-10 包括数据采集和编制可靠性报告。

SEMI E-23 见 E23。

灵巧标记 一种应答器产品，其可以被固定在容器上，用于提供可以用电子方式读写的数据存储。

灵巧标记控制器 一种在驱动器板上运行的应用程序，其负责和灵巧标记探针（一种标记控制器）的连接。

SMIF 容器 一种标准的机械接口容器。这种容器可以含有在

受控的小环境下的原版或 WIP。SMIF 容器是底部开口的，并且不用于 300mm 晶片（见 FOUP）。

速度曲线 其确定在区的转换期间的容器的速度。速度曲线由区速度和出口速度确定。例如，3-2 的曲线确定当容器完全在区域上时以速度 S3 运动，并被减速，使得当其脱离区的皮带时以速度 S2 运动。

只有在容器被完全包含在一个区域内时才启动速度曲线。注意当容器最初接触区的皮带时的速度取决于以前的曲线。

标志控制器 一种在驱动器板上运行的软件应用程序，其实现和标记读取器的接口。

输送控制器 一种软件系统，其负责高级的非实时的功能，包括外部连接和内部成分的协调。

输送系统 一般指在制造设备内把材料从一点运送到另一点的系统。

WIP 在制品，一般用于半导体晶片。

区 可以开始、停止和传递一个搬运器的轨道的一部分。一个区至少具有等于搬运器的长度。对于 300mm 的制造设备，区应当具有 500mm 的长度。

区地址 在一个节点内的区的网络地址。区地址具有 8 位的宽度。

区控制器 一种运行在驱动器板上的智能驱动应用软件，其负责一个电动机和相关的检测器的低级的实时的控制。

区最大速度 指的是在区的末端（即当搬运器完全占据区而不在其它区时）由检测器检测到的搬运器可以具有的最高速度。

区速度 指的是当搬运器完全在一个区内时的运动速度。见入口速度和出口速度。区的区速度和上游区的出口速度相同。

B. 系统描述

按照本发明实现的输送系统负责可靠地、准时地把材料从源装置的负载点移动到和目的装置相关的负载点。所述的源和目的可以

是存储系统，处理工具，晶片分类器，或者任何其它的制造系统，它们加工由输送系统运送的材料。

输送系统包括输送器硬件，控制计算机和在控制计算机中执行的软件。输送器硬件可以是单向或双向轨道，导向器，拐角，负载端口传递装置（LPTD）或升降器。除去 LPTD 之外，这些装置的每一个都由区构成。其中每个区是被单独控制的可以保持一个搬运器的输送器的物理区域。

在本说明所述的一个实施例中，在网络上具有一系列的区（以及控制计算机，被称为智能驱动器）。在所示的实施例中，所述网络是一种控制器区域网络（CAN），但是也可以使用任何高速的网络技术（例如 LonWork，火警线）。每个局域 CAN 网络和运行特定应用软件的控制逻辑计算机（CLC）相连。在利用输送器轨道的一个实施例中，CLC 区线程负责高级的控制，由区线程控制的区控制器（ZC）智能驱动器负责区的低级控制。每个区都属于多个“邻域”，每个邻域由特定区的 n 个上游区和特定区的 m 个下游区构成。在邻域中的所有的 CLC 线程共享实时信息，以确保材料的正确的运动和识别。根据物理系统的拓扑，邻域具有不同的尺寸。

输送系统需要和各个存储装置、处理工具以及在制造系统 LAN 上存在的其它元件通信。在一个实施例中，和负载端口（其又和处理工具通信）的通信由 SEMI E23 并行接口处理。在所述实施例中，输送系统的高级控制器（称为输送控制器 TC）和制造系统的 CIM 系统（例如 MCS）之间的通信通过符合 SENI 内部舱室特殊设备模式的 HSMS 连接进行，其被包括在此。此外，在 TC 和高级控制器之间的通信可以使用任何合适的通信技术和协议实现。这是因为本发明的操作独立于所述通信机构的配置。

安全系统和防火门互锁由直接和 CAN 网络相连的直接 I/O 控制器处理。这些控制器帮助确定系统操作方式的自动改变。这种控制器的版本被装入功率分配系统中，以便报警电源故障并互连故障停止装置。这种输送系统的其它细节下面参照图 4 说明。

参见图 4, 其中示出了包括本发明的输送系统 102 和常规的自动制造系统 110 的自动制造系统 100 的方块图。该图省略了输送硬件, 并突出控制系统的元件, 着重说明本发明。常规的制造系统 110 包括制造执行系统 (MES) 112 和材料控制系统 (MCS) 114。制造执行系统 112 跟踪制造系统内的组。材料控制系统 114 接收来自制造执行系统 112 的一个特定的组的下一个处理步骤的通知, 识别处理工具, 并命令输送系统 102 把组交给所述的工具。

输送系统 102 包括输送控制器 (TC) 104, 控制逻辑计算机 (CLC) 106, 和智能驱动器 108。输送控制器 (TC) 104 接收来自材料控制系统 (MCS) 114 的表示何处的材料要被移动/处理的指令。指令 115 包括把特定的晶片容器从一个处理工具传递到另一个处理工具的指令。TC 104 通过排序一系列执行所述指令的基本操作执行 MCS 指令 115。例如, TC 104 把传递指令分为一系列的获取、移动和停放指令 117, 它们被至少一个控制逻辑计算机 (CLC) 106 执行。MCS114 也可以发出这些基本操作。

TC 104 在拓扑数据库 105 内存储整个材料输送系统的物理拓扑。拓扑 105 的一种表示被组织在一组所有可能的制造系统目的地 (例如负载端口传递装置) 周围。每个目的地表示包括位置的参考和与区有关的装置信息, 由此容器可以被预负载并且从所述目的地发出。TC 104 还保持状态信息 107, 供输送系统 119 使用由 CLC106 返回的状态信息 119。

每个 CLC106 提供高级的实时控制并协调各组智能驱动器 108, 智能驱动器驱动构成物理输送系统的不同区的一组机电装置。根据来自检测器的信息 123, 对材料运动的动态限制, 邻域拓扑的图形和速度控制规则, 路由以及避免碰撞, CLC106 通过协调经低级指令 121 选择的智能驱动器 108 执行指令 115。CLC 106 只需要知道材料输送系统的其相关区域的物理拓扑, 将其存储在本地拓扑数据库 109 中。此外, 每个线程只知道相应于其邻域的拓扑。

在一个实施例中, 具有不同类型的机电装置。每种机电装置由

一个或几个智能驱动器 108 控制。例如，在一个实施例中，机电装置可以包括区（一段输送机轨道以及和其相关的检测器、电动机）、标记/条码读取器、负载端口传递装置（LPTD）、EMO（紧急停车）检测器、导向器（一种能转动的轨道装置）和升降器。因而，所述的实施例包括以下类型的智能驱动器：

区控制器	向 CLC 报告检测器数据，按照来自 CLC 的指令加速和减速电动机；
标记控制器	和标记/条码读取器进行双向通信；
轴控制器	控制 LPTD 或升降器的轴或导向器的转动；
EMO 控制器	监视电源，并在电源故障时通知 CLC；以及
握手控制器	执行多线并行连接，用于同步被单独控制的机电装置的操作。

参看图 5，其中示出了控制系统 100 的一个实施例 150 的展开的方块图。在实施例 150 中，具有一个 TC 104，其通过以太网 120 和多个 CLC 106 相连。每个 CLC 106 又通过一个或几个 CAN 总线和多个智能驱动器 124 相连。每个微处理器 124 操纵主动驱动器 108，其驱动一个构成输送机硬件的机电装置 160。例如，具有一个电动机 162 和两个检测器 164 的轨道区 160a 被区控制器（ZC）108a 控制。微处理器 124 是一般型的，其可以操纵不同类型的主动驱动器 108。

在实施例 150 中，每个 CAN 总线具有 64 个微处理器 124，由一个 CLC 106 控制 4 个 CAN 总线。根据可利用的计算机硬件和物理布局，每个 CAN 总线的微处理器 124 的数量、由一个 CLC 106 控制的 CAN 总线的数量、以及每个微处理器 124 控制的主动驱动器 108 的数量可以改变。输送器和智能驱动器的细节如图 5B 所示。

参见图 6-8，其中示出了不同的制造设备拓扑的示意图，其中包括现有技术的元件和本发明的输送系统。图 6 表示基本的拓扑 200，包括制造系统 LAN202，其和晶片存储系统 204 相连，处理工具 206，常规的制造控制系统 208，和按照本发明实现的输送控制器（TC）104。存储系统 204 和制造工具 206 通过轨道 160 或其它输送

系统 160 相连，其包括负载端口传递装置 160b-1,160b-2,用于分别负载和卸载存储系统 204 和处理工具 206。TC 104 监视舱室电源 210，并且如上所述，通过以太网控制 CLC 106。每个 CLC 106 通过一个或几个 CAN 总线 122 控制智能驱动器 108。智能驱动器 108 又控制输送系统 160 的各个元件。例如，智能驱动器 108 包括用于 LPTD160b-1,160b-2 的 LPTD 控制器 108b。其它类型的智能驱动器 108 包括区控制器（ZC）和 E23 接口（E23）。每个内部舱室和中间舱室被认为是一个完全独立的系统，其在物理上不和其它的内部舱室或中间舱室相互作用。

图 7 表示一种制造系统，其包括一系列的独立的舱室，每个舱室具有其自身的处理工具 206（为简单起见只示出了一个）和内部舱室输送系统 230。舱室通过储料器 204 连接，其和中间舱室输送系统 232 相连。每个内部舱室输送系统 230 和中间舱室输送系统 232 包括其自身的 TC 104，CLC 106，和智能驱动器 108。内部舱室输送系统 230 互不联系。

图 8 表示包括一系列的相连的舱室的制造系统。在这种结构中，内部舱室和中间舱室输送系统 230，232 作为一个互联的系统操作。结果，具有一个用于控制整个输送系统（即内部舱室和中间舱室输送系统）的 TC 104。每个内部舱室系统 230 和中间舱室系统 230 具有自身的 CLC 106，和相关的智能驱动器 108。TC 104，CLC 106，智能驱动器 108 的附加的细节下面参照图 9A - 9H 进行说明。

C. 系统描述和逻辑模块

参见图 9A，其中示出了输送控制器（TC）104 的方块图。TC 104 包括处理器 302，非易失存储器 304，例如硬盘和快闪存储器，快速主存储器 306，例如半导体随机存取存储器（RAM），以及，选择地，显示器 310 和用户输入装置，例如键盘 312 和鼠标 314。按照熟知的计算机原理，TC 104 在操作系统 320 的控制下执行存储器 306 中的程序，并通过显示器 310 和用户输入装置 312，314 执行用户交互功能。TC 104 通过在以太网上发出的指令控制其控制区域内的一

个或几个 CLC 106。如果用户不需要输入功能，显示器 310 和用户输入装置 312，314 是选择的。

非易失存储器 304 永久地存储着操作系统 320，TC 程序 322 和 TC 数据结构 340。TC 程序 322 使用 TC 类 324，其包括但不限于，运动调度类 326，运动类 328，容器定位类 330，和拓扑管理类 332。TC 类 324 确定 TC 的功能的关键部分，这在下面要详细说明。如果需要用户输入功能，TC 程序也可以包括选择的用户接口功能 334。TC 程序 322，类 326 - 332 和选择的 UI 功能 334 可以作为源码与/或可执行变量被存储。

TC 数据结构 340 包括被用于限定所述 TC 负责的输送系统的区的物理拓扑的一组数据结构。这些数据结构包括：

目的地表 342 目的地的标题

目的地 344 保持目的地的信息，包括目的地名和相应于其左右发送区和预负载区的位置的指针；

位置 346 确定输送系统中的可寻址的物理位置，包括左右相邻位置的指针，和表示是否运行从物理位置左右运动的布尔变量；

区 348 保持区的装置名，修改和区相关的位置；

导向器 35 保持导向器的信息，包括和导向器相关的上下相邻位置的指针以及是否运行从导向器上下运动的信息，修改和导向器相关的位置；

装置 352 保持用于机电装置的信息，包括 CAN 总线地址和相应于每个和位置相关的装置的装置名，其中一个或几个装置可以和每个位置相关；

参数表 354 一个变量表，其可以修改，并用于表征装置；

参数 356 具有唯一的名称、类型（例如整型，浮点型，长（整型）等等）和值的数据项。

操作系统 320，TC 程序 322，类事例 380（有时称为“对象”）和 TC 数据 390 被装入主存储器 306 中，供处理器 302 使用。类事例

380 包括运动调度 382, 运动对象 381, 容器位置 384 和拓扑管理器 386, 它们分别是类 326 - 332 的例子。TC 程序还包括 MCS (材料控制系统) 接口 388。TC 数据 390 包括拓扑数据库 392, 其包括链接的目的地表 392, 目的地 394, 位置 396, 区 398, 导向器 400, 装置 402, 参数表 404, 和参数 406, 它们分别是数据结构 342, 344, 346, 348, 350, 352, 354, 356 的例子。

输送控制器 104 使用 TC 程序 322 和 TC 数据 392 执行以下功能:
管理在其影响的范围内的所有节点的库存;
在其影响的范围内的其它控制器中下载和确认新的程序和参数;

对于在其影响的范围内的控制器保持所有数据和参数的备份;
接收来自制造控制系统的新的材料运动请求;
在材料交付处理期间确认工具的负载端口的可利用性;
获得作为材料获取处理的一部分的目的地信息;
对多个用户接口提供服务, 用于维护和显示当前操作状态。
保持在内部材料识别器和外部的制造系统范围内的材料识别器之间的映射;

保持在内部目的地地址和外部装置识别器之间的映射;
执行从外部系统接收的路由指令;
启动, 控制和监视材料的运动;
监视电源状态;
控制系统的操作状态。

下面说明 TC 104 的操作的其它细节。

图 10 示出了拓扑 392 的模式。图 10 中使用的注解和在对象模型中 (例如统一的模型化语言或 UML) 使用的相同。因此, 此处只作简要的说明。所示的每种对象都以方块表示, 其中列出了对象的属性, 以方块之间的线表示对象例之间的相关。每个相关的基数 (1 对 1, n 对 n, 1 对 n, 1 对无, 或 1 对 0..n) 被沿着每个连线示出。例如, 图 10 表示在目的地表 394 中可能的目的地 396 在 0 和 n 之间,

并且每个目的地 396 可能具有对 n 个相关的位置 398 的联系。空的箭头例如箭头 410, 表示属性的继承。例如, 箭头 410 表示虽然区和导向器数据例 400, 402 可以具有其自身的唯一的属性, 但是也继承了位置数据例 398 的所有的属性。

表中所示的每个例的属性用于一个实施例中, 并不用于限制本发明的范围。这些图表没有给出所有的属性, 只是给出了一些重要的属性。每种属性按以下形式规定: 属性名称: 数据类型。指针数据类型的属性用星号 (*) 表示。因而, 属性“左预负载区 412”是位置 398 的指针。可能需要解释的一种数据类型是 CORBA 对象参考。这个数据类型是一种在另一个地址空间 (或者同一个计算机, 或者不同的计算机) 中的软件对象的参考。例如, 属性“相关的负载区控制器 414”在用于管理相应的输送系统负载区域的 CLC 106 中把目的地数据例 396 变换为负载区域控制器 (LAC) 线程 (参照图 9B 所述)。注意本发明的实施不限于使用 CORBA 对象参考, 而是可以使用任何可以使对象参考在不同的地址和平台中间变换的中间产品。例如, 也可以使用 DCOM。

参见图 11 和 12, 其中示出了输送系统配置 420 和其作为拓扑 392 的表示的例子。输送系统 420 包括导向器 D1 - D5 和区 Z1 - Z19 的集合, 它们被连接成一个单向环, 和一段直的轨道。输送系统 420 包括在区 Z16、Z17 之间的目的地 DST1 和在区 Z10、Z11 之间的目的地 DST2。如上所述, 每个目的地 DST1、DST2 相应于 LPTD。描述输送系统 420 的信息被 MCS114 (图 4) 提供给 TC 104。TC 104 的拓扑管理器 386 由所述信息产生拓扑 392 (图 12)。

图 12 表示被表示为拓扑 392 的输送系统 420。所述表示的细节如图 12B 所示。参看图 12B, 每个导向器数据例 402 包括 4 个指针: 上连接 430, 左侧连接 432, 右侧连接 434, 和下连接 436, 它们相应于两维导向器的可能的物理连接。每个目的地数据例 396 具有 5 个指针: 左预负载区指针 438, 左发送区指针 440, 右预负载区指针 442, 右发送区指针 444, 和 LPTD 指针 445。每个区数据例 400 具

有两个指针：左侧连接 446 和右侧连接 448。每个 LPTD 例 401 具有指针 450，指向其服务的目的地表。TC 104 这样设置这些指针，使得完整地表示拓扑 392，如图 12A 所示。例如，目的地 DST2 396-2 具有指向左预负载区 Z9 的左预负载指针 438，指向左发送区 Z10 的左发送指针 440，和指向 LPTD 401-2 的 LPTD 指针 445-2，其和目的地 DST2 相关。目的地 DST2 具有右预负载和右发送指针 452，454，它们具有 0 值，因为容器只能在包括目的地 DST2 的环路中从左向右运动（即从目的地 DST2 的右方可能没有预负载或发送）。

参看图 9B，其中示出了 CLC 106 的方块图。CLC 106 包括处理器 452，非易失存储器 454，例如硬盘或快闪存储器，和快速主存储器 456，例如半导体随机存储器。按照熟知的计算机原理，CLC 106 在操作系统的控制下执行存储器 456 中的程序。CLC 106 通过经 CAN 总线连接发出的指令在其影响范围内控制一个或几个智能驱动器（IntDrv）108。

非易失存储器 454 永久性地存储操作系统 460，CLC 程序 462 和 CLC 数据结构 500。CLC 程序 462 使用 CLC 类 464，其包括但不限于，区类 466，负载区域控制器（LAC）类 472，健康监视器类 476，负载端口传递装置（LPTD）控制器类 480，和导向器控制器（DC）类 484。每个 CLC 类 464 包括各个方法（未示出）和数据结构 470，474，478，482，486。CLC 类 464 确定 CLC 的主要功能，下面还要详细说明。CLC 程序 462 还使用 CLC 状态机 488，其包括但不限于区状态机 490，LAC 状态机 492，HM 状态机 494，LPTD 状态机 496，和导向器状态机 498。CLC 类 464 可以包括用于控制输送系统的特殊元件的其它所需的类，例如，当系统包括升降器时，类 464 可以包括升降器（EC）类（未示出）。CLC 程序 462 和类 464 可作为源代码与/或可执行变量存储。

CLC 数据结构 500 包括一组数据结构 502，其用于规定 CLC 106 负责的输送系统的邻域的物理拓扑。这些数据结构包括和各个 TC 数据结构类似但形式未必相同的信息。

操作系统 460, CLC 程序 462, 类例 510 (有时被称为“对象”或“线程”)以及 CLC 数据 530 被装入主存储器 456, 供处理器 452 使用。类例 510 包括区线程 512, LAC 线程 514, HM 线程 516, LPTD 控制线程 518 和导向器控制线程 520。每个线程 510 是一个 CLC 类 464 的例子, 并实施由各个 CLC 状态机规定的行为。例如, 区线程 512 是区类 466 的例子, 并实施区状态机 490。类似地, LAC, HM, LPTD 控制器和 DC 线程 514, 516, 518, 520 从类/状态机 472/494, 476/494, 480/494, 484/496 得到。CLC 数据 530 包括拓扑数据库 532, 其包括拓扑结构 502 的例子, 其被连接以便表示 CLC 的本地拓扑和 CLC 状态 534。各种 CLC 线程 510 的高级说明如下。

在一个实施例中, 对于在输送系统上的每个区具有一个区线程 512。区线程控制各自的区控制器 (ZC) 108a (图 9C), 并且位于区控制器和 TC 104 之间的中间。区线程和其它的 CLC 线程 510 相同操作, 包括其它区线程 512, LAC 线程 514 和与公共邻域相关的导向器控制线程 520。具体地说, 每个区线程 512 完成以下功能:

- 保持关于在其邻域内的所有其它区的状态的信息。

- 和其区控制器通信, 以便进行电机控制。

- 和其区控制器通信, 以便获得检测器数据。

- 和其区控制器通信, 以便输入新的参数和程序数据。

- 以速度控制规则操纵材料间隔, 以便能够实现通过整个区的材料的安全高效的运动。

- 对区进行控制访问, 以便确保在任何给定的时间只有一个容器占据区。

下面参照图 13 说明程序的附加的细节以及和区线程相关的数据。

参看图 13, 其中示出了和用于控制区 Z4 的各个区线程 512-4 相关的程序和数据结构的方块图。如图 13 的上部所示, 包括区 Z4 的一个邻域包括区 Z1-Z3 和 Z5-Z7。为了便于说明, 假定有两个容器 C1 和 C2 在邻域中从左向右运动。区线程 512-4 (以及所有的

区线程)包括区状态机 620, 区线程程序 622 和区线程数据 624。具体地说, 区 Z4 的区线程数据包括邻区状态 626, 容器队 642, 最后的容器指针 648, 下游速度表 670, 上游速度指令(或曲线) 672, 最大速度 674, 以及速度表规则 676。

区线程程序 622 在区状态机 620 的控制下使用包含在区数据 626 中的信息实现上面参照图 9C 所述的功能。区线程程序 622 的操作下面参照图 17-37 详细说明。

邻区状态 626 分别给出邻区 Z1-Z3, Z5-Z7 的状态。其中在一个实施例中, 每个区状态 628-640 可以表示以下之一:

- 正在离开的搬运器(表示容器正在离开区);
- 搬运器已经离开(表示容器已经从区中离开);
- 搬运器停止(表示容器在区中被停止);
- 搬运器被除去(表示容器已经从区中被卸载, 例如通过 LPTD);
- 区可得到(表示容器可以进入区);
- 区保留(表示区已经被区线程保留, 以供容器运动)。

容器队 642 分别给出容器 C1, C2 的状态 644, 646。每个状态记录 644, 646 包括:

- 和容器相关的运动对象 381 的识别;
- 容器的位置;
- 容器的运动方向;
- 容器的目的地。

最后的容器指针 648 指向和最接近区 Z4 的容器相关的容器队 642 中的记录。

例如, 假定容器 C1, C2 的目的地分别是 Dest1、Dest2, 其移动对象是 381-1, 381-2, 则对于图 11 所示的情况的容器队 642 如下(最后的容器指针 648 将指向容器 C1 的记录 644):

容器队 642

容器	队数据
C1 644 (最近的容器)	运动ID = 381-1; 位置 = 退出 Z2; 运动方向 = 右; 目的地 = Dest1;
C2 646	运动ID = 381-2; 位置 = Z1; 运动方向 = 右; 目的地 = Dest2;

在所示的实施例中，下游速度表 670 含有通过区的容器的当前的和历史的的数据。在一个实施例中，速度表 670 包括在当前节点下游的每个区的速度数值（例如区 Z4 的速度表 670 将包括区 Z5, Z6, Z7 的当前的和历史的的速度值）。在当新的速度曲线不能执行需要返回旧的数据的情况下，提供历史数据。在另一个实施例中，不保持历史速度数据。每个区 512 根据从上游区 512 收到的信息更新其速度表。例如，区 Z4 512-4 根据从区 Z1-Z3 收到的信息更新其速度表 670。速度表将如下面详细说明。

速度表规则 676 说明区线程 Z4 如何使用速度表信息 670 确定下一个由特定容器执行的速度曲线。速度曲线规定容器在区 Z4 下游的邻域区的速度。其中，速度曲线可以规定在下游的邻域区内容器的速度被保持，减慢或增加。速度曲线也可以是三角形的，在这种情况下，材料增加到一定速度再减少到原来的速度。这种速度曲线可以被称为在一个区内执行的 0-0 曲线。在一个实施例中，速度表规则 676 确定如何根据容器以及在邻域内的其它容器的当前的和历史的的速度（在速度表 670 内可得到的信息）和其它因素例如邻域的物理配置导出容器的速度曲线。这些曲线被预先确定，以便阻止容器在靠近时碰撞，并确保容器平滑地被减速和加速。还具有速度表规则 676 的其它实施例，其中包括在运行期间分析的解析表示，专家系统型速度规则和上述任何规则的组合。

上游速度指令 672 是通过上游区线程输入到区 Z4 线程的指令，用于设置当容器在区 Z4 中时区 Z4 线程必须保持、加速或减速的速度。上游速度指令 672 从上游区和邻域区（例如区 Z3）的下游速度表导出。最大速度 674 是这样一个值，其可以被编程，或者由操作者或 TC 在飞行上修改，所述 TC 限制当容器通过区时的最高速度。

再次参看图 9B，在一个实施例中，输送系统的每个负载区域具有 LAC 线程 514。所述 LAC 线程控制握手控制器 108f（图 9H），并且，选择地，还控制标记控制器 108d（图 9F），并且在那些装置和 TC 104 之间的中介物。LAC 514 和区线程 512、健康监视线程 516 以及和给定的负载区域相关的 LPTD 控制线程 518 相同操作。具体地说，LAC 线程 514 完成以下功能：

在材料的获取和存放操作期间协调相关的负载端口传递装置（LPTD）并和所述装置通信。

在材料存放期间协调并排序材料从相关的预负载区向负载点的运动。

在材料获取期间协调并排序材料从相关的负载点向下游负载区的运动。

在一个实施例中，有一个在每个 CLC 106 上运行的健康监视（HM）线程 516。每个健康监视线程监视其相关的智能驱动器 108 的健康，其是在所述驱动器和 TC 104 中间的中介物，并且和其它的其操作和同一个智能驱动器 108 相关的 CLC 线程通信。具体地说，健康监视线程 516 具有以下功能：

连续监视特定的智能驱动应用程序，确保其处于操作状态。

通知相关的控制线程在区控制器操作状态中的任何变化。

在一个实施例中，具有用于每个传递机构的一个 LPTD 控制线程 518，所述传递机构用于使材料进入或离开输送系统的轨道。每个 LPTD 控制线程 518 控制其相关的传递机构的操作。每种类型的机构具有一个特定类型的 LPTD 控制线程。每个 LPTD 控制线程是一个特定的 LPTD 类的例子。LPTD 控制线程的功能是：

从负载端口获得要被移动到一个新的位置的材料。

选择地识别所述材料。

从轨道向负载端口传递材料。

在一个实施例中，对于输送系统中的每个导向器具有一个导向器控制线程 520。每个导向器控制线程 520 按照传递系统拓扑和在导向器附近所允许的运动方向控制通过其相关的导向器的一个或几个材料流。具体地说，导向器控制线程 520 具有以下功能：

使材料能够从多个输入位置中选择，然后到达多个输出位置的一个位置。

根据已知的材料的目的地确定使用哪一个输出位置和通过导向器到达所述目的地所可能的路径。

下面参照图 9C - 9H 说明由 CLC 线程 510 控制的智能驱动器。

图 9C 是计算机的方块图，其中实施作为本发明的实施例的区控制器功能。区控制器 108a 包括区控制器 (ZC) 程序 550 和 ZC 数据 552。在 ZC 程序 550 的引导下，区控制器 108a 控制和监视构成区的硬件。由 ZC 程序 550 所需的信息作为 ZC 数据 552 被存储，其包括：

最大的材料速度，

轮子每转行进的厘米数，

检测器防抖动计算，

区的长度，以及

装置的网络地址。

区控制器 108a 包括用于和特定类型的区相互作用的特定的硬件和软件，例如规定所述的区控制器 108a 和 ZC 程序 350，使得和包括两个检测器 554，556 以及电动机 448 的区相互作用。ZC 108a 的具体功能包括：

监视和去抖动检测器 554，556，

通知 CLC 106 中的相关的区线程 512 检测器状态的改变。

接收并执行电动机 558 的控制指令。

图 9D 是一种计算机的方块图，其中实施作为本发明的实施例的

轴控制器功能。所述轴控制器 108c 包括轴控制器 (AC) 程序 714 和 AC 数据 720。在 AC 程序 720 的引导下,轴控制器 108c 操纵输送机构 564 的一个轴 (例如 LPTD 轴) 的运动 (x,y,z 或 θ)。轴控制器 108c 接收沿轴在预定数量的位置之间运动的指令。位置的定位可以由检测器或者由离开已知位置的绝对偏移确定。AC 程序 714 所需的任何信息都作为 AC 数据被存储,其中包括:

从原始位置向拾取位置的微步进;

检测器的类型和数量,

检测器门限值,

轴的总行进距离, 以及

轴控制器网络地址。

轴控制器包括和特定机构 564 相互作用的特定硬件和软件 714。

图 9E 是一个计算机的方块图,其中实施作为本发明的实施例的 ID 控制器功能,所述 ID 控制器 108d 包括标记控制器 (TC) 程序 706 和 TC 数据 572。在 TC 程序 572 的引导下, ID 控制器 108d 从容器识别标记中读出容器识别数据。对于每种标记具有特定的 ID 控制器应用程序。例如,对于 Asyst RF 标记, ASYST IR 标记或条码标记具有单独的应用程序。TC 程序 706 所需的任何信息作为 TC 数据 572 被存储,其中包括:

标记类型,

读组 ID (Y/N),

读电子序列号 (ESN) (Y/N),

组 ID 长度,

ESN 长度,

标记中组 ID 的起始地址, 以及

ID 控制器网络地址。

图 9F 是计算机 108e 的方块图,其实施作为本发明的实施例的紧急停车 (EMO 控制器功能)。EMO 控制器 108e 包括 EMO 控制器 (EMOC) 程序 722 和 EMOC 数据 582。在 EMOC 程序的引导下

EMOC 108e 监视来自和输送系统电源相关的 EMO 检测器 584 的报警，并把所述报警通知各个健康监视线程 516。对于每种 EMO 检测器 584 具有特定的 EMOC 程序 722。由 EMOC 程序 722 所需的任何信息作为 EMO 数据 582 被存储，其中包括：

EMO 控制器网络地址，

EMO 源（例如火警，操作者按按钮等），以及

对每个 EMO 发出的信息。

图 9G 是计算机 108f 的方块图，其中实施作为本发明的实施例的握手控制功能。握手控制器 108f 包括握手控制器（HC）程序 590 和 HC 数据 592。在握手控制器程序 590 的指引下，HC 108f 在输送系统和负载端口之间提供 SEMI E-23 连接。握手控制器 108f 根据输送系统是在主动或被动方式下操作被提供给使其转向正确的握手协议的指令。HC 程序 590 所需的任何信息作为 HC 数据 592 被存储，其中包括：

HC 网络地址，

变换 E 23 控制线名为物理的 I/O 地址的表。

上面是针对一个特定的硬件实施例进行的说明，其中 TC，CLC，和智能驱动器用单独的计算机系统实现。不过，本发明可以用多种硬件配置来实现。例如，CLC 106 和 TC 104 可以用一个强大的计算机系统实现，或者 CLC 程序 462 可以被逐个地分配给较小的 CPU。对于这些实施，用于描述控制流程和系统软件对象之间的相互作用的本发明的基本逻辑模块是共用的。这些软件对象包括高级的成分（例如 TC 类例 380），中级的成分（例如 CLC 类例 510）和低级成分（例如智能驱动器 108）。下面参照图 14-16 说明系统的逻辑模块。

图 14 表示响应由 MCS 114 发出的指令启动和协调输送系统的操作的选择的高级（TC）和中级（CLC）的逻辑图。高级成分包括 MCS 通信接口 388，运动调度程序 382，运动对象 381 和拓扑管理器 386。中级成分包括 LAC 控制器线程 514，拓扑管理器 386 和容器定

位器。MCS 通信接口 388 接收来自 MCS 114 的远程指令 115(图 4), 并把和材料运动有关的 MCS 指令 389 传递给运动调度程序 382。

远程指令 115 由内部舱室 AMHS 输送专用设备模块确定, 即文件 2878, Rev A(1998 年 7 月 31 日)及其后续文件, 由 Semiconductor Equipment and Materials International, Mountain View CA 出版, 在此列为参考。远程指令 115(图 4)被归纳如表 1 所示, 表 1 的列包括: 远程指令名, 远程指令说明和指令参数名(有时被称为 cpname)。指令参数示于表 2。

表 1 MCS 指令 115

指令	说明	参数
撤销	由指令 ID 撤销未完成的指令	COMMANDID PRIORITY
识别	进行搬运器识别	COMMANDID PRIORITY VEHICLEID DESTPORT
安装	借助于在特定的搬运器位置对特定的搬运器识别附加特定的搬运器识别更新 TC 数据库	COMMANDID PRIORITY VEHICLEID CARRIERID CARRIERLOC
暂停	远方暂停 TC 的状态	None
继续	远程继续 TC 的状态	None
传递	进行一个搬运器从一个负载点到另一个负载点的整个传递操作	COMMANDID PRIORITY TRANSFERINFO

表 2 指令参数

CPNAME	说明	范围
ACQUIREINFO	一个记录具有两个字段 XFERPORT CARRIERID	
CARRIERID	被移动的搬运器的 ID	字母数字的
COMMANDID	远程指令 ID(以后可用于指远程指令)	
DEPOSITINFO	一个记录有两个字段: XFERPORT CARRIERID	
DESTPORT	目的地端口(负载点)唯一识别符	字母数字
PRIORITY	远程指令优先权	正常为高
SOURCEPORT	源端口(负载点)唯一识别符	字母数字
TRANSFERINFO	一个记录有 3 个字段: CARRIERID SOURCEPORT DESTPORT	
TRANSFERPORT	传递端口唯一识别符	字母数字

运动调度程序 382 产生运动对象 381, 其使用由容器定位器 384 和拓扑管理器 386 提供的信息 385, 387 协调执行特定的 MCS 指令 383 所需的输送系统的操作。拓扑管理器 386 为运动对象 381 提供输送系统布局信息 387, 以便证实在 MCS 规定的源和目的地位置之间有一工作路径。对于输送系统内的每个容器, 容器数据库 384a 的每个记录包括: 搬运器的 CARRIERID, 和在容器中的相关的负载运动。

如图 14 所示, 容器定位器 384 可以和所有的主动运动对象 384 通信。由参照图 10 所述的拓扑数据库 390 得到的信息 387 提供 MCS 指令 383 中涉及的轨道部分的信息。运动对象 381 通过向与所述指令相关的和源与/或目的地负载端口相关的负载区域控制器 (LAC) 线程 514 发出一系列运动指令 391 执行 MCS 指令 383。

例如, 假定运动对象 381 要执行以下的运动指令 383:

TRANSFER(COMMANDID=020, PRIORITY=HIGH,
SOURCEPORT=37, DESTPORT=272)

则运动对象 381 询问拓扑管理器 386 关于在位置“37”和与唯一的目的地识别符“272”相关的轨道位置之间的输送系统拓扑的信息, 容器要向着所述轨道运动。

一旦运动对象 381 证实对目的地具有至少一个操作路径 (DESTPORT = 272), 便向和位置“37”相关的 LAC 发出获取材料的指令。在上述的例子中, 这个运动指令被 020 的 COMMANDID 赋值, 其唯一地识别这个被发向执行的对象的指令。一旦 LAC 表示材料已经被获取, 则发出附加的指令, 以便使材料向目的地位置运动 (在另一个 LAC 的控制下), 并且然后把材料从搬运器上卸下, 放到目的地位置。这种情况假定被移动的容器在负载区域中出发。当要移动的容器起初不在负载区域中时 (由于电源故障), 则运动对象 381 直接和在移动容器的位置的其它类型的线程相互作用 (例如导向器控制线程或区线程)。下面参照图 15 说明在 LAC 线程和能够以分布方式执行运动指令 391 的其它 CLC 线程之间的连接。

图 15 是中级软件成分和执行构成运动指令 391 的子指令的智能驱动器的逻辑图。中级成分包括驱动器健康监视器 516, LAC 线程 514, LPTD 控制线程 518, 导向器控制线程 520 和区线程 512。智能驱动器包括区控制器 530, 轴控制器 526, 握手控制器 524, 标记控制器 530。

每个 LAC 线程 514 协调 LPTD 控制线程 518 和若干个区线程 512 (形成负载区域的那些区) 的操作。每个 LPTD 控制线程 518 可以

和作为一个负载端口的多个 LAC 线程 514 相互作用，一个负载端口传递装置可以服务于多个负载区域。每个区线程 512 是一个或多个其它区线程 512 的相邻线程，并且也是和在 0 和 2 个导向器控制线程 520（当区线程的相关的区处于导向器的邻域中时，区线程具有一个作为相邻线程的导向器控制线程）之间的相邻线程。包括区线程 512，导向器线程 520，LPTD 控制线程 518 的 CLC 线程和 LAC 线程 514 一般控制一个或几个智能驱动器 108。

如上所述，区线程 512 控制区控制器 530，导向器控制线程 520 控制区控制器 530 和轴控制器 526，LPTD 控制线程 518 控制 1-3 个轴控制器 526，LAC 控制器 514 控制握手控制器 524 和 ID/标记控制器 522。每个智能驱动器 108 还被 CLC 驱动器健康监视线程 516 监视，其向相关的控制对象 381 报告回各个驱动器 108 的健康情况（图 14）。

图 14 和图 15 的每种连接表示在系统对象之间的通信路径。这些通信路径的另一种表示如图 16 所示，该图表示主要的 CLC 软件成分和用圆圈表示的智能驱动器，并且每个对象间的通信路径用连接所述圆圈的若干个弧表示。TC 成分（例如图 14 的运动对象和输送调度控制器）用一个标号为“Transport Ctlr”的圆圈表示。在弧上的每个数字表示一个通信路径。具有多个标号的弧表示在对象类型的各个例子之间具有多个连接。例如，假定有 7 个区相邻，则区线程 512 和在其邻域中的每个其它的区线程 512 具有单独的通信路径 6, 25, 26, 27, 28, 29。

再次参看图 15，当其接收指令以便移动容器时，LAC 线程 514 通过向构成其负载区域的区线程发出容器移动子指令启动运动指令。然后这些区线程向其智能驱动器 108 和其邻域中的其它的 CLC 线程（例如导向器控制线程 520 和其它区线程 512）发出子指令，从而按照结合图 9A-9H 所述的线程的能力执行请求的操作。下面参照图 17-30 说明对象间通信的特定指令序列和用于执行传递指令 115 的活动以及一组基本操作或运动指令 391。

图 17-30 是说明 TC 和 CLC 软件成分以及智能驱动器的通信和操作的顺序图。顺序图是熟知的，因此这里只作简要说明。每个顺序图的顶部有一个涉及相关的指令顺序的软件成分表。在各个软件成分之间的通信/指令由发出所述指令的软件成分到接收器的直线表示；大部分直线标有相关的指令或信息名。指令从顺序图的顶部到底部按照其发出的顺序排列。由软件成分进行的关键操作用直接位于成分下方的具有标号的方块表示。操作一般由特定的成分响应输入的指令完成，并使特定成分发出返回指令。在说明和传递指令 115 有关的指令顺序和一些基本操作之前，说明用于启动运动指令的 3 种不同的方法。

基本操作可以由计算机集成制造（CIM）系统（相应于图 4 的制造执行系统 112）或者由具有或者没有 CIM 的自动材料处理系统（AMHS）（相应于图 4 的材料控制系统 114）启动。在 AMHS 启动的指令和在 CIM 启动的指令的情况的一个不同是，AMHS 启动是自底向上的启动，指的是其可以在没有来自 CIM 系统的指令的情况下启动容器的运动。这种自底向上的启动功能在以下情况下使用：

- a) CIM 系统已经关闭或脱机，
- b) AMHS 系统没有和 CIM 系统连接，或者
- c) CIM 系统支持自底向上的机构。下面说明这 3 种运动启动的方法。

1 CIM 系统运动启动：

当 CIM 系统启动运动时，CIM 的责任是和工具通信，使得当工具请求服务时其能够得知。在这种情况下，事件流如下：

（1）MCS 接收来自工具的请求服务的通知。MCS 作为响应向输送控制器的调度器产生主指令。在所示的实施例中，发出的指令必须是 TRANSFER（表 1）。

（2）TC 调度器产生新的 TC 运动对象，并向其输入主指令。

（3）TC 运动对象和 CLC 中的合适的负载区域控制器通信，从而实现所述指令。一旦完成，运动对象便向 MCS 回发送主指令完

成信息。

2 在 CIM 参与下 AMHS 运动启动:

在这种情况下, MCS 112 也参与运动, 其被 AMHS 工具启动。事件流如下:

(1) LAC 经 E23 接口从负载端口接收卸载请求, 并向输送控制器的运动调度器发出。

(2) 运动调度器向 MCS 输入运动请求。

(3) 作为响应, MCS 产生主指令, 向回输入给含有目的地的输送控制器调度器。

(4) 调度器产生新的运动对象, 并命令其执行规定的 TRANSFER 指令。

(5) 新产生的运动对象向在 CLC 中的源与目的地负载区域控制器发送一系列指令, 以便实现运动。

(6) 一旦操作完成, 运动向 MCS 发送主指令完成信息。

3 没有 CIM 参与的 AMHS 运动启动:

在这种情况下, MCS 不参与运动, 事件流如下:

(1) LAC 通过 E23 接收来自负载端口的卸载请求, 并向输送控制器运动调度器发出。

(2) 运动调度器产生新的运动对象, 并命令其执行 TRANSFER 操作。通过参考缺省的目的地表调度器获得目的地。所述的表含有对于每个源的缺省的目的地。

(3) 新产生的运动对象向在 CLC 中的源与目的地负载区域控制器发送一系列指令, 以便实现运动。

还有第 4 种运动启动的方法, 其中 TC 104 具有用户接口 (UI), 操作者从所述接口识别源和目的地, 并告知 TC 104 从源向目的地输送什么材料。用户可以从 UI 发出传递、获取、运动或放置指令。

4 系统操作

在讨论过 4 种运动启动方法之后, 下面说明和传递以及基本操作相关的指令序列的例子。这些说明将参引各种信息、指令和事件,

它们在执行这些指令序列的系统元件之间传输。在大多数情况下，对这些信息、指令和事件不作详细描述。关于至少一些这些指令和事件的详细信息在附录 A, B, C 中提供了，其中分别说明了外部事件（即在 CIM 和输送控制器 TC 之间报告的事件），内部 CLC 事件（即由一个 CLC 对象向另一个 CLC 对象报告的事件）和 CAN 总线信息（即智能驱动器输入和输出的信息，CLC 对象或智能驱动器输入和输出的信息）。下面按照以下顺序说明系统的操作：

- a. 传递材料；
- b. 获取材料；
- c. 移动材料；
- d. 存放材料。
- a. 传递操作：

在一个实施例中，CIM 系统 110（图 4）通过一个 CIM 系统指令 115 启动点对点的材料移动的传递操作。输送控制器 104（图 4）把传递指令分为获取、移动和存放操作，并按顺序执行这些指令。这个实施例通过把材料运动操作的责任移到输送控制器上使得 CIM 系统软件相对简单。

在另一个实施例中，CIM 系统 110 可以通过向 TC 104 发送若干个基本指令，例如获取、移动和存放实行点对点的材料移动，其中 TC 104 相应地执行所述指令。这个实施例要求 CIM 系统 110 监视每个基本指令的执行，因而，相应地简化了 TC 104 的操作。

参见图 17，其中示出了当 TC 执行 TRANSFER 主指令 700a 时在 CIM 系统 110 和 TC 104 之间交换的指令的顺序图。在启动 TRANSFER 指令之后，TC 返回“主指令启动”信息 700b。当 TC 排序执行 TRANSFER 所需的基本指令时，其返回表示构成 TRANSFER 指令的基本操作的进展的状态指令。当 TRANSFER 操作完成时，TC 返回“主指令完成”信息 700c。下面说明图 17 所示的各个基本操作和状态指令。

- b. 获取基本操作：

获取操作使输送系统从负载端口或上层起重机输送（OHT）系统获取材料。从输送控制器看来，对于这两种情况的操作是相同的。不过，CLC 按照源系统的不同的接口要求不同地处理每种情况。

在第一种情况下，输送系统使用负载端口传递装置从工具的负载端口或储料器获取材料。在这种情况下，被动的负载端口简单地发出要被卸载的信号。作为响应，主动的 LPTD 在 CLC 的控制下进行材料传递。在第二种情况下，输送系统从上层起重机输送（OHT）系统获取材料，所述主输送系统把材料放在容器升降器上。在这种情况下，容器升降器是表示要被负载的被动装置，OHF 是实现所述传递的主动装置。

图 17 表示获取操作的指令序列的高级图。如上所述，不管材料如何获取（即从 OHT 或从 LPTD），这个序列是相同的。这个序列涉及以下步骤：

- （1）TC 启动获取操作；
- （2）TC 向 CLC LAC 线程发出 ACQUIRE_MATERIAL 指令（702a）；
- （3）TC 返回表示操作状态是“获取”的事件报告发送信息（702b）；
- （4）CLC LAC 通过把材料从负载端口移动到轨道（704a）上执行指令 ACQUIRE_MATERIAL，然后向 TC 返回 MATERIAL_ACQUIRED 信息（702c）；
- （5）TC 返回表示材料已经被获取，并且容器已被安装在轨道上的事件报告发送信息（702d,702e）。

图 18 表示从使用 LPTD 进行的获取操作的 CLC 指令序列看的操作顺序图。OHT（上层起重机输送系统）将要求不同的指令序列，不过这个序列如此类似于本说明，以致在此不必进行说明。这个指令序列假定要从中获取容器的 LPTD 具有包括两个负载区的负载区域（负载区 1 和 2）。注意当材料从 CLC 不知道方向的负载端口获取时，材料将继续运动，因此其不知道两个负载区哪个是上游哪个

是下游。这个序列涉及以下步骤：

(1) CLC LAC 从 TC 接收指令 ACQUIRE_MATERIAL (702a) (见图 17)；

(2) LAC 接收来自握手控制器的卸载请求信息 (LP_UNLOAD_REQ) (714a)，该信息表示储料器或工具准备由 LPTD 卸载 - 该信息是上述的被动信息，其中负载端口简单地发出其需要被卸载的信号；

(3) LAC 向和区 1 相关的区线程发出信息 RESERVE_ZONE，从而使执行获取 ACQUIRE 操作的区 (714b) 保留 (即保证任何进入的材料在该区之前停止)；

(4) 和区 1 相关的区线程符合区 1 已经收到的 ZONE_RESERVED 信息 (714c)；

(5) LAC 向和区 2 相关的区线程发送信息 RESERVE_ZONE，以便使执行获取操作的区保留 (714d)；

(6) 和区 2 相关的区线程符合区 1 已经收到的信息 ZONE_RESERVED (714e)；

(7) LAC 向 LPTD 控制器发送信息 RESERVE_LPTD，以便保留 LPTD 执行获取操作 (714f) (确保另一个 LAC 不试图使用 LPTD)；

(8) LPTD 控制器符合 LPTD 被保留的信息 LPTD_RESERVE (714g)；

(9) 区 1 线程等待，直到邻域成为空的，或者最近的容器停止 (716a)，然后返回表示区 1 可为获取操作利用的 ZONE_IS_SAFE_NOTIFICATION (714h)；

(10) 区 2 线程等待，直到邻域成为空的，或者最近的容器停止 (716b)，然后返回表示区 1 可为获取操作利用的 ZONE_IS_SAFE_NOTIFICATION (714i)；

(11) 一旦证实区和 LPTD 是安全的，LAC 控制器就向握手控制器发出信息 INITIATE_HANDSHAKE (714j)，如果相关的负载

端口准备就绪，其则返回信息 **DEVICE_IS_READY (714k)**；

(12) 然后 LAC 控制器向握手控制器发出信息 **SET_BUSY (714l)**，其设置其具有负载端口 (LP) 的接口上的 **BUSY** 位，并且当 LP 响应时，发出信息 **BUSY_SET (714m)**；

(13) LAC 控制器然后向 LPTD 控制器发出信息 **ACQUIRE_MATERIAL (714)**，LPTD 从负载端口获取材料，并将其存放在轨道上 (716c)，然后向 LAC 线程返回信息 **MATERIAL_ACQUIRED**；

(14) LAC 线程和握手控制器交换信息 **COMPLETE_HANDSHAKE AND HANDSHAKE_COMPLETED (714q,714r)**，以便终止和负载端口的相互作用，并且 LAC 线程向 TC 返回 **MATERIAL_ACQUIRED** (图 17)。

图 19 表示从 LPTD 控制器看时使用规定类型的 LPTD 执行的获取操作的指令序列。不同的 LPTD 要求不同的指令序列，不过这些指令序列和本说明如此相似，以致在此不必说明。所示的序列假定 LPTD 控制器包括 X, Y, Z 轴的独立控制器 574x,574y,574z。假定 X 轴是轨道方向，Y 轴垂直于轨道并在轨道平面内，Z 轴垂直于轨道平面。

这个序列涉及以下步骤：

(1) 在接收指令 **RESERVE_LPTD** 之后 (714f)，LPTD 控制器通过不同的 LAC 证实其没有服务于另一个负载端口，然后响应保留请求 (720a)；

(2) 然后，LPTD 控制器返回信息 **LPTD_RESERVED (714g)**，并等待来自 LAC 的信息 **ACQUIRE_MATERIAL**；

(3) 在收到 **ACQUIRE_MATERIAL** 之后，LPTD 控制器向 X, Y, Z 轴控制器发出一系列信息 **MOVE_TO<position>**，从而使每个轴控制器沿轴各个轴把容器移动到指定的位置；

(4) X, Y 轴控制器通过执行所述移动并返回表示容器的过去的运动位置的信息 **MOVED_TO 714d** 响应指令 **MOVE_TO 714c**；

(5) Z 轴控制器通过把材料移动到材料拾取位置响应指令 **MOVE_TO 714c**。在检测到材料被拾取之后，轴控制器 (AC) 对容器检测器发出信息 **SENSOR_STATUS CHANGE**，然后返回表示容器过去运动位置的信息 **MOVED_TO 714d**；

(6) 一旦证实容器被拾取之后，LPTD 向轴控制器发出附加指令，使得容器被放置到轨道上。当容器不再被 LPTD 保持时，材料检测检测器将改变状态，引起一个 **SENSOR_STATUS_CHANGE**。一旦所有移动完成，LPTD 发出 “ ” 714o。

c.运动基本操作：

运动基本操作把在源负载位置上的容器移动到目的地的预负载区。从 TC, CLC 和区线程看来，和运动基本操作有关的指令序列如图 17 和图 20-22B 所示（对于加速和减速）。

图 17 表示运动操作的指令序列的高级图。这个序列涉及以下步骤：

(1) TC 向和材料源相关的 LAC 线程发出指令 **MOVE_MATERIAL (706a)**；

(2) TC 返回表示材料在运动的事件报告发送信息 (706b)；

(3) 当 MOVE 在进行中时，和材料源相关的 LAC 线程发出信息 **MOVE_STATUS_UPDATE (706c)**；

(3) 当 MOVE 完成时，和材料目的地相关的 LAC 线程向 TC 发出信息 **MOVE_COMPLETED (706d)**；

(4) TC 返回表示材料被运动的事件报告发送信息 (706e)。

图 20 表示从 CLC 子系统看来运动操作的指令序列。该序列涉及以下步骤：

(1) 在收到 **MOVE_MATERIAL** 指令之后 (706a)，源负载区域控制器向下游负载区线程发出指令 **SEND_MATERIAL 722b**；

(2) 下游负载区在上游负载区被其控制的情况下准备运行滑移曲线（滑移曲线使得容器完全在下游负载区内运动），这涉及下游负载向上游负载区发出指令 **ZONE_TASKING 722c**，其同步并返回

信息 **ZONE_TASKING_ACK 722d**;

(3) 一旦两个区准备就绪, 下游区便发出指令 **EXECUTE 722e**, 以便开始两个区的运动;

(4) 一旦容器离开上游负载区, 所述区就向其邻区发出指令 **CARRIER_EXITED 722f**, 并向 CLC 发送信息 **ZONE_AVAILABLE 722k**;

(5) 然后, 源负载区控制器向管理运动的运动对象返回表示运动的当前状态的信息 **MOVE_STATUS_UPDATE 706c**;

(6) 在 **MOVE** 被启动之后, 分别和区以及在源负载和目的地负载区相关的区线程和导向器控制线程在基本上不干扰运动对象的情况下处理材料运动操作;

(7) 当容器到达目的地时, 在目的地的预负载区向目的地 LAC 返回信息 **MATERIAL_ARRIVED 722j**, 然后其返回信息 **MOVE_COMPLETED 706d**, 对其进行参照图 17 所述的处理。

上面参照图 17 和 20 说明了 TC 和 CLC 子系统的高级操作。图 21A, 21B, 21C 和图 22A, 22B 表示在邻区中的区控制器如何相互作用从而使得容器加速或减速从而完成规定的运动操作。

参见图 21A, 21B 和 21C, 其中示出了开始在区线程 ZT (图的左侧) 和其各个区控制器 ZC (右侧) 之间和容器被向左加速跨过多个区的同一邻区的区线程当中的相互作用的指令序列。具体地说, 向左加速从由区控制器 ZC1 和区线程 ZT1 控制的区 Z1 开始, 直到由区控制器 ZC4 和区线程 ZT4 控制的区 Z4。被所示的 **MOVE** 操作影响的邻区包括 6 个区, Z1 - Z6。如上所述, 每个区具有两个检测器 S1、S2, 和一个电动机, 用于当容器位于该区时对其加速, 它们都被区控制器 ZC 控制。从图的顶部到底部按照时间顺序示出了 **MOVE** 操作的几个步骤。阴影矩形 P1 表示容器的当前位置。这个指令序列重复地使用几个低级的控制指令:

SET_PROFILE profile (on event, <seq #>) 730

由区线程 ZTi 向其相关的区控制器 ZCi 发出, 用于设置要由区

控制器 ZCi 实现的加速或减速曲线。在所示的实施例中，曲线被规定为固定速度的序列。例如，在所示的实施例中，曲线被规定为范围“0-1”（表示容器要从速度 0 加速到速度 1，其中速度数由区控制器翻译为相应的电动机速度）。在另一个实施例中，区线程 ZT 可以用许多方式命令其区控制器 ZC 在各个区内如何使容器加速或减速（例如，区线程 ZT 可以发出电动机的实际目标速度值）。可以在一个或多个线程上执行给定的曲线。

SET_PROFILE 指令的选择部分（在圆括号中表示）是“on event”条件，其使得区线程能够表示一些将来的事件，当这些事件发送时，则执行所述曲线。在所示的实施例中，将来的事件由序列号<seq#>表示。“on event”条件能够建立多个指令供将来执行，因此它们可以被不同的区控制器 ZC 串行地执行，所述执行的时间间隔太短，不能在合适的执行时刻单独地发出所述指令。例如，在所示的实施例中，事件是“on execute, <seq#>”，表示当由<seq#>识别的子指令被执行时执行所述曲线。<seq#>和 profile 由区线程 ZT1 产生。然后，具有相同的序列数的执行指令被同时向所有装置传送。具有那个序列数的那些装置则开始当前的执行。

ACK 732

由区控制器 ZCi 发出，用于通知相应的曲线指令 ZTi。

ZONE_TASKING profile:<seq#> 736

由区线程 ZTi 向邻区线程 ZTj 发出，用于使这些线程和在前面的 SET_PROFILE 指令中规定的运动曲线同步（即为了平滑地且协同地执行所述曲线）。在所示的实施例中，曲线被规定为索引，不过，如上所述，曲线可以用任何不同的方式规定。这个指令可以包括序列数<seq#>，其使这个指令和一组具有相同序列数的指令相关联。

ZONE_TASKING_ACK 736

由区控制器 ZTj 发出，用于通知相应的曲线指令 ZTi。

EXECUTE <seq#> 738

由区线程 ZTi 向多个区控制器 ZCj 同时发出, 用于启动和序列识别符 $seq\#$ 相关的曲线的执行。

在 EXECUTE 指令的右方和在上方标有“?”的右指向的箭头表示要通过广播发送给多个容器的信息。

SENSOR_STATUSright/left 740

由区控制器发出, 用于报告其左右检测器的状态。所述状态表示容器是否是清洁的或者是否和左右检测器一致。在所示的例子中, 检测器的转换由在指令右方的小的上下箭头表示。例如, 来自区控制器 $ZC1$ 的信息 624-1 表示容器刚刚离开区 $Z1$ 的右检测器 $S2$, 来自区控制器 $ZC2$ 的信息表示容器刚刚跨过区 $Z2$ 的右检测器 $S2$, 来自区控制器 $ZC1$ 的信息 624-3 表示容器刚刚离开区 $Z1$ 的左检测器 $S1$ 。

MOTOR_OFF 742

由区线程 ZTi 向其控制器 ZCi 发出, 用于使控制器关断其电动机。区线程 ZTi 发出这个指令用于当来自区控制器 ZCi 的信息 $SENSOR_STATUS:left$ 表示容器刚刚被传到其左检测器 $S1$ 的左侧时向左运动, 并且当来自区控制器 ZCi 的信息 $SENSOR_STATUS:right$ 表示容器刚刚被传到其右检测器 $S2$ 的右侧时向右运动。

CARRIER_EXITING 744

由区线程 ZTi 向其邻区线程发出, 表示容器正在传出各个区 Zi 。

CARRIER_EXITED 746

由区线程 ZTi 向其邻区线程发出, 表示容器已经传出各个区 Zi 。

因而, 参见图 21A, 21B, 21C, 区线程 $ZT1$ 的第一个作用是设置加速曲线“0-1”(610-1), 其由区控制器 $ZC1$ 通知(612-1)。这个加速曲线要在两个区 $Z1-Z2$ 上协同地执行(即容器要在区 $Z1-Z2$ 的空间内加速到其目标速度)。因而, 区线程 $ZT1$ 通过发送给区线程 $ZT2$ 对于曲线“0-1”的任务分配指令使其同步(614-1)。区线程 $ZT2$ 作为响应向其各个区控制器 $ZC2$ 发出指令

SET_PROFILE (730-2), 其以通知作为响应 (732-2)。在收到通知之后 (732-2), 区线程 ZT2 向区线程 ZT1 返回任务分配通知 (736-1)。然后, 区线程 ZT1 向区控制器 ZC1, ZC2 发出 EXECUTE 指令 738-1, 使得先前在指令 730-1, 730-2 中设置的曲线由区控制器协同地执行。当执行所述指令时, 区控制器返回表示容器行进的信息 SENSOR_STATUS 740。

在容器被加速到其目标速度之后, 由下游区线程设置的相继的曲线使其保持固定的速度 (例如 SET_PROFILE “2-2” 730-3)。区线程 ZTi 使用 CARRIER_EXITING 和 CARRIER_EXITED 信息 744, 746 通知邻区线程 ZTj 运动的进程。每当容器离开相应的区 Zi 时 (由信息 CARRIER_EXITED 746 表示), 区线程 ZTi 向其各个控制器 ZCi 发出信息 MOTOR_OFF 742。

参见图 22A 和 22B, 其中示出了开始在区线程 ZT (图的左侧) 和其各个区控制器 ZC (右侧) 之间和在相同邻区内的区线程当中相互作用的指令序列, 跨过多个区 Z1-Z4 的容器在接近其目的地 (在本例中区 Z4) 时在所述邻区被减速 (例如从速度 2 到 0)。线程 ZT1 根据其速度表 670 (图 13) 中的指示容器在其到达区 Z4 的时刻要被减速到速度 0 的信息开始减速。所述减速使用和上述相同的基本指令进行。图 22A 和 22B 表示一个指令序列, 借助于所述指令序列, 在区 Z1 中以速度 2 从右向左运动的容器在区 Z4 中被减速到速度 0。这个减速由区线程 ZT2, ZT3, ZT4 以及其各个区控制器 ZC2, ZC3, ZC4 协同地进行。图 22A 和 22B 此处不再详细说明, 因为其和图 21A, 21B, 21C 的说明相同。

d. 存放基本操作

存放操作使位于目的地的预负载区的容器向着负载点运动, 然后向负载点或者 OHT 卸载。现在参照图 17 说明存放操作的高级部分。

图 17 表示存放操作的指令序列的高级图, 其中材料被交付给负载点。负责 TRANSFER 的运动对象向 CLC 发送信息

DELIVER_MATERIAL 708a, 然后返回表示材料的存放状态的信息 **EventReportSend 708b**。作为响应, LAC 线程使用一系列规定将实现所述传递的特定的 LPTD 的指令协调材料从轨道向负载端口 710a 的运动。

当交付操作完成时, LAC 控制器线程返回信息 **MATERIAL_DELIVERED 708c**。接着, TC 向 CIM 系统返回表示存放操作完成 (708d) 和容器已经从轨道上被除去的信息 (708e)。

使用快速加载的 LPTD 存放材料的一个例子如图 23A, 23B, 24A, 24B 所示, 其是一种专用的 Palo Alto 技术产品。

图 23A, 23B 表示从 CLC 看的处理。图 24A, 24B 把所述处理改进为 LPTD 控制线程的处理。LAC 首先保留构成负载点的区。保留这些区使得确保没有其它的运动容器 (来自另外的方向) 试图进入这些区。保留通过发出 **RESERVE_ZONE_request** 请求完成。所述的区用 **ZONE_RESERVED** 响应。一旦确定任何过来的容器都在其之前停止, 便发出 **ZONE_IS_SAFE_NOTIFICATION**。此时, LAC 得知其能够唯一地控制和访问该区。以和参照图 18-19 所述的用于进行获取操作相同的方式, 在其保留负载区域的区之后, LAC 保留 LPTD。一旦 LPTD 被保留, LAC 便把容器移动到要由 LPTD 获取的位置, 此时其使用由上游和下游负载区实现的一系列 **MOVE** 指令。一旦容器就位, LAC 便利用负载端口启动 E23 握手程序。一旦负载端口表示其准备接收, LAC 便向 LPTD 控制器发出指令 **DELIVER_MATERIAL**, 使 LPTD 控制器把容器移开轨道向目的地工具或 OHT 运动 (724)。LPTD 向其实现交付的轴控制器发送一系列运动指令。在 LPTD 完成交付之后, LPTD 控制器向 LAC 返回指令 **MATERIAL_DELIVERED**。作为响应, LAC 向握手控制器发出信息 **COMPLETE_HANDSHAKE**, 从而使工具或 OHT 完成握手操作。一旦存放操作完成, LAC 控制器便释放 LPTD 和负载区域区。下面参照图 24A, 34B 说明从快负载 LPTD 控制器看来存放操作的附加细节。

参看图 24A, 24B, LPTD 控制线程执行以下的操作, 以便完成交付操作 724。在本说明中, LPTD 包括 X, Y, Z 轴控制器 574x,574y,574z, 按照参照图 19 所述进行操作(获取操作)。

选择地, 向标记读出点运动(如果读出点是轨道, 则不需要运动);

选择地, 读出识别数据(如果具有读取器);

选择地, 证实和输送控制器的一致性(如果具有 ID 读取器);

通过发出指向 X 轴控制器 574x 的 MOVE_TO<lp position>指令 750-1 向负载端口移动, 以便向负载端口位置运动从而获取容器, X 轴控制器在完成之后利用 MOVE_TO 信息 752-1 进行响应;

通过发出指向 Z 轴控制器 574z 的指令 MOVE_TO <pod acquire position>750-2, 从而从轨道上获取容器, 从搬运器/轨道获取材料, Z 轴控制器通过发出 SENSOR_STATUS_CHANGE 信息 754-1 进行响应, 表示其正在获取容器, 在完成获取之后, 发出信息 MOVED_TO 752-2;

存放材料, 其中通过(1)发出指向 Z 轴控制器 574z 的指令 MOVE_TO <full up>750-3, 以便移动到相应于要被加载的工具的负载高度的尽头的位置, 在完成之后, Z 轴控制器发出信息 MOVE_TO 752-3 进行响应;(2)发出指向 Y 轴控制器 574y 的 MOVE_TO <extended position>指令 750-4, 以便移动到延伸的位置, 把容器装在工具上, 在完成之后, Y 轴控制器发出信息 MOVED_TO 752-4 进行响应;(3)发出指向 Z 轴控制器 574z 的指令 MOVE_TO <lp deposit position>750-5, 以便存放材料, Z 轴控制器发出 SENSOR_STATUS_CHANGE 进行响应;容器未被检测到时信息 754-2 表示材料已经被存放, 并且发出信息 MOVED_TO 752-5。

通过向 Y、Z 轴控制器 574y,574z 发出各个 MOVE_TO <full retracted>和 MOVE_TO <full down position>信息, 将 LPTD 复位到其完全缩回和完全降低的位置, 其分别利用 MOVE_TO 信息 752-

6, 752 - 7 进行响应; 以及使用信息 708b 通知其它系统运动完成。

注意图 24A, 24B 所示的步骤假定没有 ID 读取器。

5 轨道到达情况:

至此讨论的是容器被移动到目的地 (例如 LPTD 的情况)。在其它情况下, 例如当容器必须在某个轨道位置上等待的情况下, 运动指令的末尾可以是轨道区代替负载区。这种情况被称为轨道到达情况。

在这种情况下, TC 把容器移动到目标轨道区。一旦位于目标轨道区, 容器便不从轨道被除去, 而是保留在轨道上直到对容器发出另一个运动指令。在正常情况下, TC 通过和目的地的负载区有关的 LAC 执行传递操作。不过, 在这种情况下, 无意把容器移动到目的地。因而, 即使当轨道区位于负载区域, TC 也直接和目标轨道区的区线程相互作用而不和相关的 LAC 线程相互作用。下面参照图 25 说明和轨道到达情况相关的指令序列。

图 25 表示和轨道到达情况相关的指令序列。在这种情况下, 作为运动的终点的区的区线程使容器在该区内减慢和停止。然后, 同一个区线程向其邻区线程发出信息 CARRIED_STOPPED (750a), 并向 TC 发出 MATERIAL_ARRIVED 信息 (750b)。

在说明了和可以由不同的实施例执行的几个基本操作相关的指令序列之后, 现在说明附加的细节, 说明区线程如何协调操作使得沿所需的方向移动一个或几个容器。帮助区线程协调其运动操作的重要因素是由每个区内的检测器提供的检测器信号。

图 26 是表示由 6 个检测器 S1 - S6 产生的信号的时序图, 从区 Z2' 的控制线程 (ZT2, 未示出) 看来所述的检测器和 3 个区 Z1 - Z3 相关。所示的信号被从区 Z3 到区 Z1 的用于 300mm 的晶片的容器的检测器产生 (用于 200mm 晶片的容器的定时将不同)。以虚线表示的信号不涉及 ZT2' 的运动控制方法。以粗体表示的信号可以直接地或者通过所示的 (意思是) 信息用于线程 ZT2。一般地说, 可以为任何线程使用的信号是和其自身的检测器以及邻区的出口检测器相

关的信号。因而，在区 Z2 的情况下，线程 ZT2 所需的信息包括来自检测器 Z5（经过信息）、其自身的检测器 Z4，Z3 以及检测器 S1（经过信息）的信号。在所示的实施例中，当容器通过所述检测器时确立每个检测器信号。因而，左检测器信号的上升沿向接收区线程 ZTi 表示容器开始离开相关区，而下降沿表示容器已经离开所述的区。

E 由区线程进行的速度控制

在讨论过相同操作之后，将参照图 27-29 详细说明一个实施例的软件对象和硬件控制器的各种指令序列的分布式执行，其中涉及区线程协调地确定和管理一个或几个材料单元（例如晶片的容器）通过输送系统的运动的方法。下面说明在 3 种运动中区线程的作用：

一个容器的向左运动；

占据同一邻区的两个容器的向左运动；

占据相同邻区的两个相邻容器的向左运动。

所述的情况是在区线程之间可能的宽范围的交互作用的例子，并不限制本发明的范围。

一般地说，当确定如何使材料加速和减速时，区线程 512 遵守一组被严格限定的速度控制规则 676（图 13）。下面根据一些性能假定说明性能/速度控制规则 676，所述性能假定能够给出精确的、有代表性的例子。这些假定包括：

1 从停止到停速占据 3 个整区。

2 从整速到停止占据 3 个整区。

这些假定是特定的例子，对于本发明的不同的实施例和不同的材料可以不同。

区线程 512 协同地实施速度控制方法，其中使用由区线程 512 在同一个邻区内交换的表示运动的材料的运动状态的信息。这些运动状态按照以下的区运动通信规则被交换，所述规则被包括在作为每个区线程 512 的一部分的区状态机 620（图 13）中（注意，在下面的说明中有时使用简称“区”指“区线程”，这个简称混淆了在

区和区线程之间的软件/硬件的区别，这只用于表征本发明的一个实施例而不是所有的实施例）：

1 当区开始使容器向区外运动时（由下游区检测器的上升沿检测到），区将发出信息 **CARRIER_EXITING**。下游区将使用这个信息得知进入的容器。

2 当区完成使容器离开区的运动时（由下游区检测器的下降沿检测到），区将发出信息 **CARRIER_EXITED**。上游区将使用这个信息确定容器继续运动，并且可以从区数据库中被除去。

3 如果区使容器在区内减速到停止，则区将发出信息 **CARRIER_STOPPED**。

4 在容器到达其目的地并被除去之后，将发出 **CARRIER_REMOVED** 或 **ZONE_AVAILABLE** 信息。

在给定上述的性能假定和区运动通信规则之后，下面说明区对区的速度控制规则，这些规则被包括在区线程系统 622 和速度表规则 676 中（无特殊说明均指图 13）：

1 不论由上游节点规定什么样的速度曲线，区必须接收容器。

2 每个区线程 512 在其邻区内保持容器队 642。每个区 512 具有指向最近的容器的指针 648。

3 每个区线程 512 含有其下游区的出口速度表 670。下面示出的在一个实施例中使用的速度值（注意实际的速度数（1..n）取决于要求把材料从停止加速到最大速度和使材料从最大速度减速到停止速度的区数）。

3 2 1 0 - 1（表示区是不能利用的）- 4（表示区被保留）。

4 当前区线程 512 使用其直接下游邻区的最大速度 630 以及在进入当前区时容器的速度确定被遵守的速度曲线。

当区检测到容器正在出上游区时（使用上游区的下游区检测器的上升沿检测），进行所示曲线的确定。

5 当前区的离开速度可以小于下游区的最大允许的进入速度 630（例如，如果容器处于目的地的邻区并过渡到停止）。

6 当收到用于区的信息 **CARRIER_EXITING** 时,接收的区线程 512 将设置速度表 670 的项,以便向区的下游邻区发送速度 - 1 (表示其不可利用性)。

7 当区线程 512 收到信息 **CARRIER_EXITED** 时,将设置速度表 670 的项,以便向区发送速度 0 (从 - 1 或 - 4)。所有较近的区将使其速度增加 1。即从发送区下一个区的上游将具有进入速度 2,然后,下一个为 3,并且任何随后的上游区也将是 3。从最远的到最近的将依次增加。如果碰到 - 1 或 - 4 的速度,将不增加,因而增加结束。

8 当区线程 512 收到信息 **RESERVE_ZONE** 时,如果该区尚未被保留,其将把其本身标记为保留区,并向其邻区发出 **ZONE_RESERVED** 多点传送信息。如果其已被标记为保留的,或者其是有故障的,则其向启动器发出 **ZONE_NOT_RESERVED** 单点传送信息。

9 当区线程 512 收到信息 **ZONE_RESERVED** 时,其将设置速度表 670 的项,以便使区成为 - 4。用于上游区的速度表 670 的项将分别被修改为 0, 1 和 2。接收区将产生前面的速度数据的备份。

10 当区线程 512 收到 **CARRIER_STOPPED** 信息时,其将设置速度表 670 的项使区成为 - 1。在同一个速度表 670 中的上游区速度将被修改为 0, 1 和 2。

11 当区线程 512 收到 **ZONE_AVAILABLE** 信息时,其将设置速度表 670 的项使区成为 3。用于上游区的速度表 670 的项也将被修改为 3。在区上的容器的记录将被删除。

12 当区线程 512 收到 **CARRIER_REMOVED** 信息时,规定的容器的记录将被从容器队 642 中除去。速度控制表 670 将不被更新,因此容器被除去的区将保持不可利用。当区线程 512 正在计算在下一个区中使用的曲线时,如果区确定其不能实行所述的曲线(例如在一个区中执行曲线 3-0),则区将扫描速度表寻找表示前面具有保留区的 - 4 速度。如果发现保留有下游区,则当前区得知其必须使容器通过所述的区,因为在当前区没有供停留的足够的空间。此时

当前区将使用前面的速度的备份继续进行。当在目的地的邻区内时，区线程 512 将按照下表 3 中的信息设置容器的速度曲线，其表示可允许的最大速度曲线。所述表中的信息被包括在速度表规则 676 中。例如，表 3 表示对于离开目的地两个区的区给定一个当前上游出口检测器速度数“2”，合适的速度曲线是“2→1”（表示容器的速度要被减 1）。不过，如果对于同一个检测器的出口检测器速度数是 3，则没有可以应用的速度曲线（N/A）因为任何曲线都要求容器每个区减速大于 1 的速度数。

表 3

离开目的地的距离 (区) ----- 出口检测器的 当前速度▲	1	2	3	4
0	0→0 (creep)	0→1	0→1	0→1
1	1→0	1→1	1→2	1→2
2	N/A	2→1	2→2	2→3
3	N/A	N/A	3→2	3→3

图 27-29 是指令序列定时图，表示由区发出的速度控制和其它信息，用于实现材料运动操作（图的下部）和所得的材料速度曲线（图的上部）。和上述的定时图一样，每个图都从上到下说明序列的进展。每个信息和标有标号的箭头相关，标号表示信息在指令序列中的顺序，并附有特定运动操作的一般说明。一些信息的说明参考速度表，所述速度表是由每个区线程存储的表，用于表示允许区在什么速度下加速材料以便进入相邻的区（见图 13 的速度表的说明）。随着运动操作的进行，材料的位置由有阴影的矩形表示。下面参照图 27-29 说明 3 个指令序列。

图 27 是表示在区和容器速度曲线之间发送的信息的定时图，用于材料 X 被存放在区 ZI，向区 ZA 运动的情况。假定加速到全速需

要3个附加的区(H, G和F), 因此容器的前沿达到区F的后沿时, 其达到了最大速度。在区I, H, G, F中的电动机被同步, 使得从区到区的过渡是平滑的。一些或所有这些区可以由一个微控制器控制, 但是, 这些区可以由两个节点控制, 所述节点的边界可以在任何区的边界。按两个阶段进行电动机的同步。第一, 是当前区的两个区的下游的区将从当前区收到信息 **CARRIER_EXITING**。其将使用在该信息中的曲线的最终速度启动其电动机在那个恒定的速度下运行。在第二个阶段, 是当前区的直接下游的区将收到信息 **CARRIER_EXITING**, 并改变其当前速度以便和信息中的速度匹配。信息序列如下:

(1) 区I发出信息 **I (ZONE_RESERVED)**, 以便获取负载, 用于新材料的装载。上游区将调整它们的入口速度表, 以便确保尚未在邻区中的进入的容器在区之前停止。

(2) 一旦区I确定邻区是安全的, (即没有容器入区), 区继续负载处理, 使容器X被存放。区将根据速度表启动容器的运动。区得知下游区H可以在直到速度3的任何速度下被进入。区还知道假定容器被停止, 唯一可利用的速度曲线是0-1。区将产生信息 **CARRIER_EXITING**, 其中含有0-1曲线的识别符。下游区将接收这个信息并和曲线同步。区I将更新区H的进入速度。

(3) 一旦容器离开区, 如检测器的后沿所示, 区I将发出信息 **CARRIER_EXITED**。上游节点将修改区I的入口速度值。

(4) 区H利用检测器的上升沿检测到容器正在离开区。此时, 容器达到速度1。通过检查其入口速度表, 其确定当前速度小于对于G的最大允许入口速度, 因此使用1-2曲线。其把区G的速度入口设置为-1。

(5) 在检测到检测器的后沿之后, 区H发出已离开信息。区I接收所述信息, 并修改其区H的速度入口。I不需要进一步处理, 因为H是其直接的邻区。

(6) 区G通过检测器的上升沿检测到容器正在离开。此时容器

已经达到速度 2。通过检查其入口速度表，其确定当前速度小于对于 F 的最大允许速度，因此使用 2-3 曲线。其设置区 F 的速度入口为 -1。区 H 和 I 也更新它们对于 F 的速度入口。

(7) 区 G 在看到检测器的后沿时发出已离开的信息。区 H 和 I 收到所述信息，修改其对于区 G 的速度入口。H 不需要进一步处理，因为 G 是其直接的邻区。I 需要进行附加的处理，以便根据对 G 的改变更新区 H 的速度值。

(8) 区 F 通过检测器的上升沿检测到容器正在离开。此时容器正在以速度 3 运动。通过检查其入口速度表，其确定当前速度等于对于 E 的最大允许速度，并通告改变为 3-3 曲线。其把区 E 的速度入口设置为 -1。区 G, H, I 也更新其对于 F 的速度入口。

(9) 区 F 在检测到检测器的后沿之后，发出已离开信息。区 G, H, I 收到所述信息，修改其区 F 的速度入口。G 不需要进一步处理，因为 F 是其直接的邻区。H 和 I 却需要进行附加的处理，以便更新从 F 起的区上游的速度值。

(10) 区 E 利用检测器的上升沿检测到容器正在离开。容器正在以速度 3 运动。通过检查其入口速度表，其确定当前速度等于对于 D 的最大允许入口速度，因此继续当前曲线。然后，其确定距离规定的目的地 3 个区，因此必须开始使容器减速。其发出具有曲线 3-2 的离开信息，并把对于区 D 的速度入口设置为 -1。区 F, G, H, I 也更新其对于 E 的速度入口。

(11) 区 E 在检测到检测器的后沿之后，发出已经离开信息。区 F, G, H, I 接收所述信息，并修改其对于区 E 的速度入口。F 不需要进一步处理，因为 E 是其直接的邻区。G, H, I 却需要进行附加的处理，以便更新从 E 起的区上游的速度值。

(12) 区 D 利用检测器的上升沿检测到容器正在离开。容器正在以速度 2 运动。D 知道距离规定的目的地区 B 2 个区，根据当前速度和 B 的位置，D 确定其需要转换到减速曲线 2-1，这包含在其离开的信息中。其把对于区 C 的速度入口设置为 -1。区 D, E, F,

G 也更新其对于 C 的速度入口。

(13) 在检测到检测器的后沿之后, 区 D 发出已经离开信息。区 E, F, G, H, 接收所述信息, 并修改其对于区 D 的速度入口。E 不需要进一步处理, 因为 D 是其直接的邻区。F, G, H 却需要进行附加的处理, 以便更新从 D 起的区上游的速度值。

(14) 区 C 利用检测器的上升沿检测到容器正在离开。其正在以速度 1 运动。C 知道距离规定的目的地区 B 1 个区, 根据当前速度和 B 的位置, 其把曲线设置为 1-0, 并把 B 区的速度入口设置为 -1。区 C, D, E, F 也更新其对于 B 的速度入口。

(15) 在检测到检测器的后沿之后, 区 C 发出已经离开信息。区 D, E, F, G 接收所述信息, 并修改其对于区 C 的速度入口。D 不需要进一步处理, 因为 C 是其直接的邻区。E, F, G 却需要进行附加的处理, 以便更新从 C 起的区上游的速度值。

(16) 区 B 检测到减速曲线的终点和区检测器的前沿。得知其是目的地, 区发出信息 CARRIER_ARRIVED。区 C, D, E, F 将其区 B 的入口速度设置为 -1, 并更新从 B 起的其上游区的速度。区 B 通知较高级的控制容器已经到达, 并等待被除去。

(17) 一旦区 B 检测到容器已被除去, 其便发出信息 ZONE_AVAILABLE。区 C, D, E, F 将其区 B 的入口速度从 -1 改变为 3, 并更新从 B 起上游区为速度 3。

在上述的 17 步中材料在区中的速度如表 4 所示。所述每个区的信息和被包括在各个区线程 ZT-B 到 ZT-I 的速度表 670 中的信息是等效的。

表 4

区/邻区		信息数																	
		init	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I	H	3	3	-1	-1	-1	0	0	1	1	2	2	3	3	3	3	3	3	3
	G	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2
	F	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0	0
	E	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0
H	G	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	3	3	3	3	3
	F	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	2	2
	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	1	1	1
	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0
G	F	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	3	3	3
	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2
	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	1
	C	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0
F	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	3
	D	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	3	3
	C	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	3	3
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	3
E	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	3
	C	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	3	3
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	3
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	C	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	3
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	3
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0
C	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	3
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
B	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	X	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

图 28 是表示在区之间发送的信息的定时图，针对的情况是，负载区 D 被保留用于在容器 Y 进入的同时接收来自一个装置的新的材料 (X)。如果容器正在全速运动，它必须在区 H 开始减速，以便在区 E 停止。区在允许新的容器被存放在轨道上之前必须确保所有进入的容器都已经停止或者通过。注意在这种情况下，进入的容器必须减速，使得其不会追上加速的容器。不过，根据计算，进入的容器不需要移动到一个停止点。如果如此，在两个容器之间的最终的空间将是 ± 6 个区，这不能提供好的利用。如果容器被减速到速度 1，然后被加速到 3，则最终的空间略大于 3 个区。区线程可以包括这个考虑或类似的考虑，以便改进轨道的最佳利用。信息序列如

下:

(1) 区 E 发出 `ZONE_RESERVED` 多传递信息。在 E 的邻近的所有区将收到所述信息, 并用 -4 更新其 E 区的速度表。上游区的速度被更新为 0, 1, 和 2。

(2) 一旦 E 确定没有不能停止的容器入区, 其便允许入区 X 被存放在轨道上。因为下游区 D 在其速度表入口中具有 3, E 可以自由地使容器加速。当容器被停止时, 唯一可利用的曲线是 0-1。发出具有这个曲线的离开信息。

(3) 几乎在容器 Y 以恒速 3 运动之后的瞬时, 容器释放区 J 中的下游检测器。J 检查其速度表, 并发现其可以用速度 3 进入 I, 发出具有曲线 3-3 的离开信息。所有的邻区都知道 I 是 J 的下游, 因而把 I 标记为不可利用的 (-1)。

(4) 容器 Y 离开区 J, 并发出已离开信息。J 的上游区将更新其速度表。

(5) 以速度 3 运动的容器 Y 释放在区 I 中的下游检测器。I 检查其速度表并发现对于 H 的速度是 2。I 转换到曲线 3-2, 并发出离开信息。下游区知道 H 是 I 的下游, 因而把 I 标记为不可利用的。

(6) 容器 Y 离开区 I, 并发出已离开信息。上游区将其 I 区的入口从 -1 改变为 0。

(7) 容器 Y 现在以速度 2 运动, 释放区 H 的下游检测器。H 检查其速度表并发现对于 G 的速度是 1。H 改变为曲线 2-1, 并在离开信息中将此发出。区知道 G 是 H 的下游, 将其标记为不可利用的。注意此时 X 大约进入 D 的途中。

(8) 容器 X 完成 E 区的过渡并发出已离开信息。所有具有标记为保留 (-4) 的 E 的区把这个字段更新为 0, 并递增每个上游区。

(9) 容器 X 现在以速度 1 跨过区 D 的下游检测器而运动。D 检查其 C 区的速度表入口, 并发现其可以使用直到 3 的任何速度。因为其当前速度是 1, 唯一的可利用的加速曲线是 1-2, 其在离开信息对此作了通知。邻区知道 D 区的下游是 C, 因而把 C 标记为不可

利用的。

(10) 现在以速度 1 运动的容器 Y 完成其 H 区的过渡, 并发出已离开信息。邻区把 H 的 -1 改变为 0, 并使上游区增加到最大值 3。

(11) 正在以速度 1 运动的容器 Y 碰到 G 的下游检测器, 并检查速度表中的 F 的入口。该表表示速度 1, 因此该区改变为曲线 1 - 1, 并在离开信息中通知。所有邻区都将其 F 的入口设置为 -1。

(12) 正在以速度 2 运动的容器 X 离开区 D 并发出已离开信息。上游区使其 D 的入口从 -1 变为 0, 并增加 D 的上游区的值。

(13) 容器 X 以速度 2 跨过下游检测器, 并检查其区 B 的速度表入口。因为值是 3, 区改变为曲线 2 - 3, 并在离开信息中将此发出。邻区把其对于 B 的速度表入口设置为 -1。

(14) 以速度 1 运动的容器 Y 完成其区 G 的过渡, 并发出已离开信息。邻区将其 G 的入口从 -1 变为 0, 并使上游区的入口增加。

(15) 以速度 1 运动的容器 Y 释放区 F 的下游检测器。该区检查其区 E 的速度表入口, 并发现速度设置为 1。所述区发出具有曲线 1 - 1 的离开信息。上游区将其区 E 的速度入口变为 -1。

(16) 容器 X 其区 C 的转换之后, 发出离开信息。上游区把 C 区的速度表入口从 -1 变为 0, 并增加 C 的上游区的速度。

(17) 现在以速度 3 运动的容器 X 释放区 B 的下游检测器, 并检查其区 A 的速度表入口。因为入口是 3, 并且容器以 3 运动, 其转换为恒定的 3 - 3 曲线, 并在离开信息中通知。上游区知道 A 是 B 的下游, 因而设置 A 的速度值为 -1。

(18) 容器 X 完成区 B 的转换, 并发出已离开信息。上游区设置 B 的速度为 0, 并增加上游区。

(19) 容器 X 释放 A 的下游检测器, 检查其速度表, 并发出对于曲线 3 - 3 的离开信息。上游区标记 Z 为不可利用的。

(20) 容器 Y 完成其区 F 的转换, 并发出离开信息。上游区将 F 的速度从 -1 改变为 0, 并增加其上游区。

(21) 以速度 1 行进的容器 Y 释放区 E 的下游检测器。E 检查

其 D 区的速度表，并发现为 2。E 发出具有曲线 1-2 的离开信息。上游区把区 D 设置为 -1。

(22) 容器 X 完成其区 A 的转换并发出已离开信息。上游区用速度 0 标记 A，并增加其上游区。

(23) 以速度 3 运动的容器 Y 释放区 Z 的下游检测器。Z 发出具有曲线 3-3 的离开信息。

(24) 以速度 2 行进的容器 Y 完成其 E 区的转换，并发出已离开信息。上游区以速度 0 标记 E 为可利用的，并增加其上游区。

(25) 以速度 2 行进的容器 Y 释放 D 的下游检测器，检查区 D 的速度表，发现值为 2，并且 Y 在其离开信息中通告 2-2 的曲线。

注意在本例中，速度 2 和 3 之间的差值足够小，使得 Y 需要许多区才能到达全速（X 和 Y 之间的间距非常慢地被减小）。

在上述的 25 步中区内的材料的速度如表 5 所示。所述每个区的信息和被包括在各个区线程 ZT-A 到 ZT-J 的速度表 670 中的信息是等效的。

表 5

		信息数																								
区/邻区	init	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
J	I	3	3	3	-1	-1	-1	0	0	0	0	1	1	3	3	3	3	3	2	2	3	3	3	3	3	3
	H	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2
	G	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	1	1	1	1	1	1
	F	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0
I	H	3	2	2	2	2	-1	-1	-1	-1	-1	0	0	0	0	1	1	1	1	1	2	2	2	2	3	3
	G	3	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	1	1	1	1	2	2
	F	3	0	0	0	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1
	E	3		-4	-4	-4	-4	-4	-4	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
H	G	3	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	1	1	1	1	2	2	2
	F	3	0	0	0	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1
	E	3	-4	-4	-4	-4	-4	-4	-4	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
	D	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1
G	F	3	0	0	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1
	E	3	-4	-4	-4	-4	-4	-4	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
	D	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1	-1	-1	-1	-1	-1	-1
	C	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	-1
F	E	3	-4	-4	-4	-4	-4	-4	0	0	0	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0
	D	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	1	1	1	2	2	2	-1	-1	-1	-1	-1
	C	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	1	1	1	1	1	1	-1
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	0	0	0	0
E	D	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	1	1	2	2	2	-1	-1	-1	-1	-1
	C	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	0	0	1	1	1	1	2	2	-1
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	1	1	1	1
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0
D	C	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	0	0	1	1	1	1	2	2	-1
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	1	1	1	1
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1
C	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	1	1	1	1
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1
B	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	3	3	3	3	3	3
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1
	X	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
A	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1
	W	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	X	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

图 29 是一个定时图，表示在区之间发送的信息，针对的情况是两个相邻的容器 X 和 Y 都开始运动，并且最终容器之间具有间隔。这种情况的初始条件是容器 Y 在区 J 上停止，因此容器 X 可以在区 I 上存放。一旦 X 被存放，其将开始向其目的地运动。这引起容器 Y 的运动。信息序列如下：

(1) 区 I 检查其速度表，发现其可以在直到最大速度的任何速度下进入区 H。当容器停止时，其必须使用 0-1 曲线，并利用离开

信息对此通告。邻区接收所述信息，并标记 H 区为不可利用的（-1）。

（2）一旦 X 已经离开区 I，I 将发出离开信息。上游邻区将在其速度表中标记 I 是可利用的。

（3）在更新其速度表之后，区 J 将确定其可以开始运动容器 Y。根据正在被停止的容器，和 I 的速度表入口是 0，J 可以以曲线 0-0（爬行速度）运动容器。

（4）区 H 通过下游检测器检测容器。容器 X 现在以速度 1 运动。H 为区 G 参考其速度表。H 决定使用曲线 1-2，并在离开信息中告知。上游邻区调整其对于 G 的速度表。

（5）H 检测到 X 已经完成其区的转换，并发出离开的信息。上游区把 H 标记为可利用的，并更新其 H 的上游区的入口。

（6）区 G 检测到 X 正在以速度 2 离开。在参考其速度表之后，G 转换为 2-3 曲线，并发出离开信息、上游区标记 F 为不可利用的。

（7）X 清除区 G。区 G 发出离开信息。上游邻区标记 G 为可利用的，并更新 G 的上游区的速度入口。

（8）区 F 检测到 X 正在以速度 3 离开。F 参考其速度表，发现 E 可以以 3 进入，因此转换为 3-3 曲线，并在离开信息中告知。上游邻区相应地调节其速度表入口。

（9）区 F 检测到出口检测器的下降沿，并发出离开信息。上游邻区调整其速度表，标记 F 为可利用的。

（10）区 E 检测到离开检测器的上升沿，并参考其速度表。E 确定其必须遵守 3-3 曲线，并发出离开信息。上游区标记区 D 为不可利用的。

（11）区 E 检测到出口检测器的下降沿并发出已离开信息。上游邻区调整其速度表，使得 E 区是可利用的。

（12）区 D 检测到离开的检测器的上升沿，并参考其速度表。D 确定其必须遵守 3-3 曲线，并发出离开信息。上游邻区标记区 C 是不可利用的。

(13) 区 D 检测到出口检测器的下降沿, 并发出离开的信息。上游邻区调整其速度表, 使得 D 区是可利用的。

(14) 区 C 检测到离开检测器的上升沿并参考其速度表。C 确定其必须遵守 3-3 曲线, 并发出离开信息。上游邻区标记区 B 为不可利用的。

(15) 区 I 检测器容器 Y 已经以爬行速度释放出口检测器。I 参考其速度表并发现可以以速度 3 进入。不过, 因为容器正在以速度 0-0 运动, 其必须使用曲线 0-1。I 发出具有曲线的离开信息。上游邻区标记 H 是不可利用的。

(16) C 检测到容器 X 已经离开区并发出离开信息。上游邻区标记 C 为可利用的, 并更新对于 C 的上游邻区的速度。

(17) 区 B 检测到离开检测器的上升沿并参考其速度表。B 确定其必须遵守曲线 3-3, 并发出离开信息。上游邻区标记 A 区是不可利用的。

(18) 区 B 检测到出口检测器的上升沿并发出离开信息。上游邻区调整其对于 B 的速度表并标记为可利用的。

(19) 区 A 检测到离开检测器的上升沿并参考其速度表。A 确定其必须遵守曲线 3-3, 并发出离开信息。上游邻区标记区 Z 是不可利用的。

(20) 区 A 检测到出口检测器的下降沿并发出离开信息。上游邻区调整其速度表使得 A 是可利用的。

(21) 区 Z 检测到离开检测器的上升沿并参考其速度表。Z 确定其必须遵守曲线 3-3, 并发出离开信息。上游邻区标记区 Y 不可利用的。

(22) 区 I 检测到出口检测器的下降沿并确定 Y 已离开该区。I 发出离开信息。上游邻区标记 Y 是可利用的, 并更新对于 Y 的上游邻区的速度入口。

(23) 区 H 检测到出口检测器的上升沿。容器 Y 现在以速度 I 运动。H 参考其速度表并决定转换到曲线 1-2。H 在离开信息中发

出这个信息。上游邻区标记 G 是不可利用的。

(24) 区 H 检测到出口检测到的下降沿, 并发出离开信息。上游邻区调整其速度表, 使得 H 是可利用的。

(25) 区 G 检测到其出口检测器的上升沿。Y 正在以速度 2 运动。G 确定其可以转换到 2-3 曲线, 并在离开信息中通告这个信息。上游区标记 F 是不可利用的。

(26) 区 G 检测到出口检测器的下降沿并确定 Y 已离开该区。G 发出离开信息。

(27) 区 F 检测到其速度表并决定使用 3-3 曲线。F 在离开信息中发出这个信息。

在上述的 27 步中区内的材料的速度如表 6 所示。所述每个区的信息和被包括在各个区线程 ZT-A 到 ZT-J 的速度表 670 中的信息是等效的。

表 6

		信息数																												
区/邻区		init	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
J	I	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1	2	2	
	H	3	-1	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1	
	G	3	3	3	3	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	-1	-1	0	0
	F	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	-1
I	H	3	-1	-1	-1	-1	0	0	1	1	2	2	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1
	G	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	-1	-1	-1	0	0
	F	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	-1	-1
	E	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1
H	G	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0
	F	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	-1	-1	-1
	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	F	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	3	3	3	3	3	3	3	3	3	3	-1	-1	-1
	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	-1
	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	C	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
F	E	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	1	1	1	2	2	3	3	3	3	3	3	3	3	3	3	-1
	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	1	1	2	2	2	2	2	2	2	2	2	2	2
	C	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1	1	1
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0
E	D	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	1	1	2	2	3	3	3	3	3	3	3	3	3
	C	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2	2	2
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0
D	C	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	2	2	2	2	2	2	2	2	2
	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
C	B	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	0	0	1	1	1	1	1	1	1	1	1
	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	0	0	0	0	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
B	A	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	0	0	0	0	0	0	0	0	0	0
	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	X	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1
A	Z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
	Y	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	-1
	W	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1
	X	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1

在另一个实施例中，速度表 670 只表示区是被占据还是被保留（即在表 670 中不包括速度数据）。按照速度表规则 676 的较简单的实施例，区线程 512（图 13）使用其下游区的占据或保留的状态和容器的进入速度计算容器的离开/最终速度数。在这个实施例中，区线程 512 设置最终速度为从 0, 1, 2, 3 中选择的速度数。具体地说，当一个或几个下游区被占据时，区线程 512 便按照上述对于的情况设置容器的最终速度数，其中相应的速度表值被设置为 -1。当

其一个或几个下游区被保留时，区线程 512 便按照上述的情况设置容器的最终速度数，其中相应的速度表值被设置为 -4。当没有下游区被占据或被保留时，区线程 512 便自由地设置容器的最终速度为任何约束容器运动的速度数。例如，当其下游区未被占据和保留时，如果中间的区是其目的地，则区线程 512 应当被限制于设置容器的离开/最终速度为 0。

F 区线程动态模型

每个区线程根据可能的区操作/状态的动态模型执行一个状态机 620（图 13）。在顶层，这些状态包括：

- 启动 当区启动材料时，
- 停止 当区停止材料时，
- 运输 当区沿着其当前路径运动材料时，
- 中断 当区阻止材料的运动时，
- 空闲 当区不参与材料运动时（区分配任务），
- 就绪 在以下条件下区处于就绪状态：

- 1 材料到达区中的停止点（从停止开始运输）；
- 2 材料已被放置在区中并等待开始运动。

下面参照图 30 说明这个动态模型。

图 30 使用 and UML（统一模型化语言）一致的状态图表示法表示本发明的一个实施例的动态模型。具体地说，每个区线程状态和其子状态，或者操作用具有标号的方块表示，例如在图 30 中标号为“停止”的方块代表区线程的停止状态并含有和其相应的区被停止在该区的容器占据的区线程相关的操作或子状态。每个状态或子状态相应于一系列执行所述状态或子状态的相关操作的指令。为了得到关于 UML 的详细信息，可参阅 Martin Fowler with Kendall Scott, UML Distilled, (1997), 列于此处作为参考。

按照 UML，每个信息/事件用在表示由该信息/事件引起的转变的箭头上的标号表示。例如，参看图 30，事件“CARRIER_EXITING: Upstream”（“Upstream”是包含在代表上游节点的信息中的数据）

引起从空载状态到运输状态的改变。在该图中所指的各个事件和信息在附录 A - C 中说明了。内部信息（即在状态对象内的信息，例如就绪对象 812）是包括在对象的方块的边界内的那些信息。

外部信息通过在对象方块的周边上的端口从对象引出。例如，**CARRIER_EXITING**：由空载状态/对象 816 发出的上游事件 817d 引起向输送状态/对象的转变。这些信息中的至少一些已经参照图 17 - 25 的指令序列进行了说明。和可能引起转变的事件相关的测试使用 C 语言表示，例如，表达式“**DetinationID=thisNode**” 817a 测试运动的目的地 ID 是否等于当前区（这个节点）的 ID。

图 30 是说明包括在区线程中的区动态模型/状态机的状态图。所述状态图表示在具有隐含的每个超状态的子状态信息的超状态之间的相互作用。所示的状态包括：停止 810，准备 812，启动 814，空载 816，输送 820 和中断 818，如上所述。**ResManager** 状态 822（保留的管理器的缩写）为需要从负载端口或储料器通过 LPTD 在轨道上被输送的容器保留相关的区。**ResManager** 822 确保没有容器试图通过 LPTD 将要存放材料的区。当容器到达并完全停止在该区内时，从停止状态到就绪状态的区线程转换 811a 发生。当由于新的运动指令或者由于下游阻断被除去而使得在该区中的容器开始运动时，从就绪状态到启动状态 814 的区线程转换 813a 发生。当容器被 LPTD 从区中除去时，从就绪状态到空载状态 814 的区线程转换 813b 发生。当静止的容器开始运动并离开区时，从启动状态到空载状态 814 的区线程转换 815 发生。当阻断从所述区中除去时，从阻断状态 818 到空载状态 814 的区线程转换 819 发生。当材料到达作为运动目的地（**DestinationID**）的区时，从空载状态 816 到停止状态 810 的区线程转换 817a 发生，信息 **MATERIAL_ARRIVED** 伴随着这个转换。当由表示区被保留的 **ZONE_IS_READY** 信息指定的事件发生时，从空载状态 816 到就绪状态 812 的区线程转换 817b 发生。当区的进入或离开检测器检测到表示区故障的 1 时，从空载状态 816 到阻断状态 818 的区线程转换 817c 发生，**ZONE_FAULT** 伴随着这个转换。

当表示容器正在离开区的直接上游区的上游信息 CARRIER_EXITING 被收到时,从空载状态 816 到输送状态 820 的区线程转换 817d 发生。

G 导向器操作

导向器确定在涉及轨道拐弯或连接的运动过程中材料所取的路径。与此相对,协调运动的运动对象只在启动运动之前确定在材料源和目的地之间是否具有某个路径,而实现所述运动的区线程中在材料的当前方向沿直线对材料加速或减速。除去提供重要的材料路径之外,导向器还能够根据区的故障或在其附近的负载端口动态地设置路径。

在一个实施例中,某个导向器包括路径选择表,其表示有效的路径和通过导向器的转动的角度。路径表在系统初始化时被建立,并且由导向器进行修改,修改的理由是发现一些初始路径由于机电转换元件或控制这些元件的智能驱动器的故障而成为不可利用的。下面参照图 34, 35 说明路径表及其产生。为了方便,只说明每个导向器使用路径表动态地确定通过它的每个材料单元的有效路径便足够了。

下面参照图 31, 32A, 32B 说明导向器的操作。这种情况涉及在导向器附近通过导向器选择路径的两个材料单元的同时到达。在所示的实施例中,导向器比常规的区具有较大的周围尺寸(例如在导向器的每侧上具有 4 个区,而不是 3 个区)。在附近增加的区使得导向器能够检测进入的材料单元,并使其减速到 0,在检测器的上游区,检测器将不能被利用(假定材料单元从全速减速到 0 需要 3 个区)。注意这种情况只是一个例子,并不限制本发明的范围。

图 3 是输送系统的区域的示意图,包括检测器 Q,两个交叉的轨道和两个材料单元 P1, P2,它们要沿着 R1, R2 所指的方向运动。检测器具有两个邻域: Z, Y, X, W, V, U, T, S 和 A, B, C, D, E, F, G, H。检测器同时种对一个邻域发出信息。这在图 32A, 32B 中通过在导向器的每一侧对信息名(例如 CARRIER_EXITED

Qvw)附加用于邻域的字母示出了,表示该信息被发送给邻域 Z, Y, X, W, V, U, T, S。

在这种情况下,材料 P1 在材料 P2 之前进入导向器的控制范围。P1 必须左转而 P2 需要沿直线行进。图 32A, 32B 所示的事件的顺序如下:

区 A 产生用于 P1 的信息 CARRIER_EXITING 830-1。导向器接收所述信息,确定在 P2 进入 832 之后其必须转动。

区 S 产生用于 P2 的信息 CARRIER_EXITING 830-2。导向器接收所述信息,并把 P2 置于其服务队 832 中。

导向器对区 B, C, D 发出信息 ZONE_RESERVED 834-1,表示容器需要减速,以便在 Q 停止。该信息使上游区 B, C, D 开始减速 P1。P1 被减速从而在导向器上停止。同时, P2 减速在区 V 停止(见信息 CARRIER_STOPPED 836-1)。(这在作为初始化的一部分检测器表示 Qvw 是不可利用的时发生。这使得沿着 R, S, T, U, V 进入的任何材料在检测器之前停止)。

当检测器确定 P1 停止时,其转动到使 P1 能够在区 W 上离开的位置 838。

当 P1 离开检测器时,发出信息 CARRIER_EXITING Qvw 840。一旦 P1 消除导向器,便发出信息 CARRIER_EXITED Qvw842。信息 CARRIER_EXITED 842 使得检测器对于从 V 到 W 的运动是可利用的。

一旦 P1 离开导向器,导向器便向区 V, Y, X 发出信息 ZONE_RESERVED 834-2,表示进入的容器 P2,并且只有 P2 可以通过导向器而不用停止(因为通过导向器的路径是直的)。一旦容器进入区 W,区 W 便发出信息 SAFE_TO_ROTATE 848。

一旦区 V 确定 P2 可以开始运动(根据上述的运动规则),P2 将开始向回加速到全速,从 V 到达 Q、W 等。一旦容器 P2 离开导向器,导向器便向区 W 发出信息 CARRIER_EXITING:Qvw 和 CARRIER_EXITED:Qvw。

导向器接收来自下游区的表示容器已经完全离开的信息 `SAFE_TO_ROTATE`。导向器检查其服务队，发现没有其它的材料，便转动回到原始位置。

一旦转动完成，便向邻区 A、B、C、D、E、F、G、H 发出信息 `ZONE_AVAILABLE 846`，通知可以从 D 通过 Q 并到达 E 的任何材料。

除去上述的导向器的情况之外，还可能有其它的导向器情况。

2 导向器组

在一些情况下，多个导向器可以直接连接在一起，构成导向器组。导向器组需要附加的通信，以便确保容器能够通过所有所需的导向器而不会发生死锁的情况。

图 33A - 33B 表示可能的导向器组结构，图 33C，33D 表示可能的死锁的情况。在图 33A，33B 中，导向器具有标号 A，B，C，D，利用阴影箭头表示通过每个导向器的可能的进入和离开的方向，利用细的实线箭头表示材料通过导向器组的可能的路径。

注意在这种类型的导向器组中，当材料单元需要转过 U 形的弯时，涉及到 4 个方向而不是两个。如果 U 形弯只需要两个导向器，则由于只有两个材料单元而可能产生死锁。借助于要求使用所有的 4 个导向器，使得只有在 4 个导向器上都具有材料时才发生死锁。图 33C，33D 表示可能的死锁条件。

在图 33C 中死锁是当材料从相对方向同时到达发生的，并且每个材料单元需要使用一个以上的导向器以便继续运动（例如两个材料单元想要沿直线行进）。在图 33D 中，死锁发生在两个向南运动的容器进入导向器组，其中第一个朝西转向的情况下。虽然这些情况是极少的，但是导向器控制器应当被这样构成，使得这些情况根本就不会发生。导向器组使用的用于避免死锁的方法如下：

（1）当导向器检测到接近的容器时，其将确定要采用的输出方向。

（2）如果所述输出方向和另一个导向器相关，则第一导向器必

须向第二导向器发送请求执行进入的信息。

(3) 如果第二导向器不能立即同意执行进入, 则第一导向器将不允许容器进入。

(4) 如果第二导向器也确定容器的输出方向是进入另一个导向器的方向, 其在回答第一导向器之前将请求执行从第三导向器执行进入。

(5) 第三导向器可能想要请求执行对第四导向器的进入。

(6) 一旦第一导向器收到来自第二导向器的进入允许, 将保证除去第二个之外任何其它的导向器都允许进入。

F 容器路径

导向器沿着多个可能的路径支持容器选择路径。下面说明的用于选择路径的机构被设计用于满足下面的要求:

(1) 允许区被任意地标号。Asyst 自动输送系统要求一种既复杂又难于维持的寻址方案。这种寻址方法反映物理布局并且当输送器的结构改变时需要更新。下面的机构能消除这些限制。

(2) 规定自动发现到目的地的路径。在 Asyst 自动输送系统中, 到任何目的地的路径都隐含在目的地地址中。只有决定点必须进行简单的数值比较, 从而选择输送容器的方向。不过, 这要求在增加目的地时用手更新控制系统内的数据库。下面的算法使得系统可以自动地更新它们的数据库。

(3) 提供一种机构, 使得当新的目的地区或导向器被安装时, 它们能够自动地把自身结合在系统的其余部分中。

建立路径长度

在本系统中的执行选择路径功能的导向器可以具有一个以上的到达目的地的路径。在具有多个离开方向的情况下, 导向器必须具有一些量度, 借助于所述量度可以选择路径。

为了获得这个距离数据, 导向器向其所有离开方向的直接下游邻区发出信息 PATH_LENGTH。当正常的区收到所述信息时, 其增加区计数字段, 并把信息输入到其直接相邻的下游邻区。最后, 信

息到达下游导向器，其增加计数并向回给始发导向器发送信息。所述信息借助于始发导向器被加到其选择路径的表中。

当收到来自监控程序的信息 `ROUTE_CONT` 时，或者当应用程序启动并发现没有到其下游导向器的距离信息时，也执行这种处理。直到导向器收到对其下游导向器的距离信息，它才产生信息。

路径的发现

按照本发明实现的输送系统能够在系统中发现从加载/卸载区到任何其它加载/卸载区的路径。当加载/卸载区有效时（例如相关的节点第一次被加电），通过 `DEST_ANNOUNCE` 信息所述区通知上游节点其是目的地。如果上游节点不是导向器，其便向上游节点发送信息 `DEST_ANNOUNCE`，直到达到导向器。每当所述信息向后发送时，区计数器便增加。因而区计数器的值表示从导向器到负载区的距离。

在收到信息 `DEST_ANNOUNCE` 时，导向器更新其路径选择表，从而包括进所述目的地。导向器然后向其上游导向器发出宣布所述目的地的信息。最后，这个数据通过所有的导向器传输并返回到始发导向器。始发导向器不发送所述信息。通过使区信息在节点之间传送直到到达导向器，每个从目的地看来是上游的节点便能够确定哪一个目的地最近。因而，任何区可以通知其上游导向器由于故障哪一个目的地已经变成不可利用的。（这个操作将在下面详细说明）。

当导向器成为有效时，其确定到其下游导向器的路径长度，然后采集并传输路径信息。现在参照图 34-36 说明如何执行这个操作的几个例子。然后参照图 37-38 说明在故障的系统元件面前导向器如何修改其路径信息。

这些图都表示一般的输送系统的物理布局，其中：

拐弯用标有“C”的圆圈表示；

导向器用标有“Di”的圆圈表示，其中“i”是索引；

区用空白方块表示，或者用标有 Zi 的方块表示，其中“i”是索

引；

目的地加载/卸载区标有“L/U”；以及
允许的行进方向用细的实线箭头表示。

具体地说，图 34 表示具有一个加载/卸载区 L/U、两个上游区 Z1, Z2、和几个导向器 D1, D2, D4, D6, D8 的物理布局。在发现路径中使用的信息一般在上游流动。为了较好地理解这些信息流，图 35 表示从区 L/U 看来在图 34 的元件之间连接的上游。给定图 34 所示的布局，假定区 L/U 被第一次加电。这个区检查其配置信息，并确定其是加载/卸载目的地。因为这是其第一次被加电，在其非易失存储器中的标记被清除，表示区 L/U 尚未作为目的地登记。作为标记测试的结果，区 L/U 宣告其自身作为目的地。将引起以下的信息：

(1) L/U 对其紧接的具有距离 1 的上游区发出信息。

(2) Z1 记录 L/U 的地址和距离。Z1 向其紧接的上游区 Z2 发送信息，这次具有距离 2。

(3) Z2 记录 L/U 的地址和距离。Z2 向其紧接的上游邻区 D8 发送信息 DEST_ANNOUNCE。

(4) D8 对其路径表增加 L/U 作为直接路径。

(5) D8 向 L/U 返回信息 DEST_REGISTERED。L/U 接收这个信息，并修改其目的地标记，使得在下一次程序启动时将不再发出信息 DEST_ANNOUNCE。如果在预定的时间间隔之后没有收到信息，则节点等待一个随机的时间间隔，并发出另一个 DEST_ANNOUNCE。这一直继续到接收到一个响应。

(6) 一旦 D8 确定对 D2, D4, D6 的距离，其向它们发出信息 DEST_ANNOUNCE。所述信息含有从直接路径加上到每个导向器的距离的总和。D2, D4, D6 更新其路径表，以便包括 L/U 作为间接路径或“通路”。根据导向器得知的对 D8 连接的物理方位，离开方向被设置。

(7) 然后 D6 向上游导向器 D8 发送信息 DEST_ANNOUNCE。

因为 D8 具有到 L/U 的直接路径，其将放弃限定一个通过路径的信息。

(8) D4 向具有到 L/U 的总距离的导向器 D2 发出信息。D2 确定经过 D4 的路径取和 D8 的路径不同的离开方向，并且在表中存储新的路径。

(9) 作为从 D8 接收信息的结果，D2 向 D2 发送信息，并作为接收 D2 的信息的结果，发送另一个信息。首先发出的一个是不确定的。在接收到第一个信息时，D1 增加两个路径（一个用于对 D2 的每个离开方向），其路径长度由所述信息和已知到 D2 的距离计算。当第二信息到达时，D1 确定路径已经存在，但是重新计算所述距离，并且如果新的距离较小，将修改表。没有信息因设置的第二信息而被发出。

(10) D1 对 D4, D6 发出信息 DEST_ANNOUNCE。D4, D6 识别新的路径，并在其表中存储所述路径。

(11) D4 向 D2 发出路径信息。D2 确定其已经具有所述路径，重新计算距离，并在需要时修改其路径表。没有信息因设置的第二信息而被发出。

(12) 由于通过 D1 的路径，D6 也向 D8 发出信息。D8 放弃通过路径的信息，因为其具有直接路径。

由于这些操作，导向器 D1, D2, D4, D6, D8 的路径表被更新，如表 9 所示。在这种情况下，“目的地”栏表示可以通过在“导向器”栏列出的导向器达到的一个路径的目的地的 ID，“路径类型”或者是“经过”（如果目的地通过另一个导向器达到），或者是“直接”（如果目的地可以不通过另一个导向器达到），“导向器出口方向”栏对每个“经过”路径给出一个导向器和出口方向，路径必须经过所述方向，并且每个“直接”路径，只有一个出口方向。

表 9

路由信息			
导向器	目的地	路径类型	导向器出口方向
D1	L/U	经过	D2: 90o
	L/U	经过	D2: 180o
D2	L/U	经过	D4: 0o
	L/U	经过	D8: 90o
D4	L/U	经过	D8: 0o
	L/U	经过	D1: 270o
D6	L/U	经过	D8: 90o
	L/U	经过	D1: 180o
D8	L/U	直接	0o

注意，对于任何给定的目的地，导向器必须导向到所述目的地的路径数等于导向器具有的出口路径数。

当新的导向器被加入系统中时，出现第二种情况。这可以由更新故障的导向器引起，或者由增加新的轨道部分引起。这两种情况的处理有些不同，因此将分别讨论。当更新现有的导向器时，导向器需要：

求出路径长度。

使下游目的地对自身通告。这需要一个节点一个节点地传送信息，以便避免在系统中的每个目的地重复通告。

加载来自下游导向器的所有其它的路径。

当新的导向器加入时，除去以上述的方式求出部分长度之外，可能具有新的轨道部分，至少被现有的导向器存储的路径信息的子组需要被更新。现在参照图 36 说明在这种情况下重新选择路径的例子。

图 36 表示图 34 的物理布局，其中增加了新的导向器 D9 和相关的轨道。因为增加了导向器 D9，导向器 D1 不再沿着 180o 路径和导向器 D2 相连，即其路径表不再有效，上游导向器的 D2 的表也不正确。最后，D9 具有空的路径表。现在说明处理这种改变的方法。

注意，为了在机械上和逻辑上增加新的轨道部分，导向器 D1 以前需要由其监督程序通过信息告知，以便中断其 180o 路径的使用。一旦新的轨道部分被加电，导向器 D9 便检查其路径表，并且发现是空的。结果，导向器 D9 首先确定通过 0o 和 90o 出口到导向器 D2 的路径长度。然后，导向器 D9 向其下游导向器 D2 发出信息 ROUTE_TABLE_REQ。如果当导向器 D2 收到来自 D9 的信息时由于某些原因导向器 D2 没有路径表，导向器 D2 将不响应。然后导向器 D9 将空出时间，其再次发出信息。这继续到导向器 D2 向导向器 D9 发出路径表为止。（注意如果整个系统是第一次启动，则从 D2 接收的路径表可能是不完整的，这是因为所有的目的地信息尚未通过网络传输。其余的信息作为通过网络传送的信息最后到达 D9）。连接的导向器的导向器 D1 的表也必须被更新，以便包括 D9。D1 也需要更新路径表，以便包括其 180o 出口。

路径表

如上所述，在所示的实施例 中，每个导向器保持一个路径表，由此确定容器离开的方向。路径表含有所有本地目的地（即不通过另一个导向器可以直接到达的那些区）和远方目的地（即必须通过其它的导向器到达的区）的入口。在一个实施例中，路径表含有以下信息：

目的地。

出口方向。

路径是直接的或者是经过的。

路径长度（即从当前的导向器到最终的目的地的小区数）。路径长度更一般地被称为“品质因数”，其表示两个路径对一个特定的目的地的相对的“品质”。最简单的品质因数是距离，但是也可以使用其它的算法，例如通信条件（即相对速度）等。

路径状态

路径数据被这样组织，使得对于给定目的地的所有路径都是邻接的。在一种可能的组织中，直接路径是被经过路径跟随着的第一

个路径，经过路径按照增加的距离排序。

路径的改变

除去上述增加路径的情况之外，还有由于现有路径成为不可利用的而发生的路径的改变。路径成为不可利用的原因如下：

- (F1) 目的地区检测到机械故障。
- (F2) 目的地区的节点故障。
- (F3) 上游导向器和目的地之间的区检测到机械故障。
- (F4) 上游导向器和目的地的节点之间的节点故障。
- (F5) 目的地上游的导向器检测到机械故障。
- (F6) 目的地上游的导向器故障。
- (F7) 监控程序发出废除路径的指令。

情况 F7: 监控程序废除路径

在情况(F7)中，负责导向器的监控程序通知导向器一组特定的路径要被废除。监控程序在信息 PATH_DISCON 中规定要被废除的导向器的输出方向。此时，所有和所述方向相关的路径成为不可利用的。这种类型的废除可以在选择不用的轨道部分时进行，也可以在维修或轨道改变时进行。

情况 F4: 总的节点故障

在情况 F4 中，当节点故障时（例如由于断电或 CPU 故障），所述节点将不再能够运送容器，甚至不能将其状态通知其上游的导向器。这种状态需要由另一个节点检测。为了使节点能够检测其下游节点的总体故障，每个节点定期地向其直接相邻的下游节点发出信息 NODE_PING。如果接收节点没有接收到对于脉冲信号的响应，则发送节点就假定下游节点发生了故障。然后发送节点向其上游相邻节点发送含有故障节点的地址和其最近的目的地信息 NODE_FAULT（这用于通知导向器哪一个目的地不能到达）。所述信息被向回传送到导向器。导向器向监控程序发出信息 NODE_FAULT，使用出口方向定位所有发生故障的目的地，并且解除对这些目的地的路径。一旦节点检测到下游节点发生故障，便继

续尝试对该节点发送脉冲。对于每个不连续的脉冲，节点向回对上游发送另一个 **NODE_FAULT** 信息。

最后，由于修复或者由于更换，下游节点将响应脉冲信息。然后，发出脉冲的节点向回向上游发送信息 **NODE_RESTORED**，该信息将到达导向器。如果节点被修复，则其以上述的方式告知，但是下游目的地仍然需要被加入。如果节点被修复，其不在此告知。

图 37 表示一种示例的物理布局，用于说明系统处理程序对故障节点（标记“X”）进行的处理。在节点故障之前的这种布局的初始路径信息示于表 10。除去“最后的导向器”一栏之外，该表中的栏和表 9 相同，上述的一栏表示下游导向器是否是在目的地之前的最近的导向器。

表 10

初始路径信息				
导向器	目的地	路径类型	导向器 出口方向	最终的导向器
D1	L/U1	经过	D2: 90o	F
	L/U1	经过	D2: 180o	T
	L/U2	经过	D2: 90o	T
	L/U2	经过	D2: 180o	T
	L/U3	经过	D2: 90o	T
	L/U3	经过	D2: 180o	T
D2	L/U2	直接	90o	-
	L/U3	直接	90o	-
	L/U1	经过	D4: 0o	F
	L/U1	经过	D8: 90o	T
D4	L/U1	经过	D8: 0o	T
	L/U1	经过	D1: 270o	F
	L/U2	经过	D8: 0o	F
	L/U2	经过	D1: 270o	F
	L/U3	经过	D8: 0o	F
	L/U3	经过	D1: 270o	F
D6	L/U1	经过	D8: 90o	T
	L/U1	经过	D1: 180o	F
	L/U2	经过	D1: 180o	F
	L/U2	经过	D8: 90o	F
	L/U3	经过	D1: 180o	F
	L/U3	经过	D8: 90o	F
D8	L/U1	直接	0o	-
	L/U2	经过	D6: 0o	F
	L/U2	经过	D6: 270o	F
	L/U3	经过	D6: 0o	F
	L/U3	经过	D6: 270o	F

假定标记为 X 的节点发生故障。最终的处理如下：

(1) 节点 DN 向下游节点发送 NODE_PING, 没有收到回答。

(2) 节点 DN 向其上游节点发送信息 NODE_FAILED。该信息包括故障节点的地址和 L/U3 的地址(当目的地宣告自身时得知的信息)，对节点 DN 的最近的目的地。

(3) 上游节点继续发送信息，直到由导向器 D2 收到。D2 在路径表中发现 L/U3，并检测路径和所有更远的并通过 90o 出口进入的其它直接目的地(即 L/U2)。D2 将删除在 90o 通过 D8 的 L/U1 的路径。当 L/U2 和 L/U3 被从路径表中除去时将对 D1 发送 ROUTE_DISCON 信息。对于到 L/U1 的通过路径将不发送信息 ROUTE_DISCON，因为到 L/U1 的路径仍然通过 0o 出口出去。(在故障节点的情况下，不使用最后导向器标记)。

(4) D1 接收间断路径 L/U2 的信息。其将从其路径表中除去 L/U2 的两个入口，并向 D4 和 D6 发送 L/U2 的信息 ROUTE_DISCON。类似地，对于 L/U3 的路径将被删除，并对 D3 发送 ROUTE_DISCON 信息。

(5) D5 和 D6 将接收对于 L/U2 和 L/U3 的信息 ROUTE_DISCON，并从其路径表中删除入口。

(6) D4 将向 D6, D8 发送信息 ROUTE_DISCON。D6 将使该信息无效，因为没有在其表中发现路径(已被删除)。D6 将不发送任何信息。D8 将从其路径表中删除这些路径，并向 D6、D4 发送信息。两者都将丢弃这些信息。

(7) D6 也将向 D8 发送信息 ROUTE_DISCON。D8 将丢弃这些信息，因为不再存在路径。

在这种处理之后，由导向器存储的路径信息如表 11 所示。表中的栏和表 9 的相同。

表 11

最后的路径信息			
导向器	目的地	路径类型	导向器 出口方向
D1	L/U1	经过	D2: 90o
	L/U1	经过	D2: 180o
D2	L/U1	经过	D4: 0o
D4	L/U1	经过	D8: 0o
	L/U1	经过	D1: 270o
D6	L/U1	经过	D8: 90o
	L/U1	经过	D1: 180o
D8	L/U1	直接	0o

情况 F2: 区机械故障

如果区碰到限制容器运动的机械故障，则区利用信息 **ZONE_FAULT** 把所述故障和最接近的下游目的地地址通知上游节点。所述信息被向回发送给上游导向器。导向器使用 信息中的地址标记对所述目的地的路径和所有目的地的路径作为不可利用的。然后，导向器向其已经成为不可利用的每个路径的上游导向器发送信息 **ROUTE_DISCON**。

情况 F5/F6: 导向器故障的处理

下面的例子参照图 38 说明当检测器故障时进行的处理。在这个例子中，导向器 D8 发生了故障。这种布局在故障之前的初始路径和表 5 所示的相同。在这种情况下，系统进行如下的处理：

- (1) 当导向器 D8 故障时，其不再响应信息。
- (2) 信息 **NODE_FAILED** 将从上游拐角区被传递到 D6, D4。这个信息还将通过节点沿着其 90o 路径向回传递到 D2。
- (3) 当导向器 D2 被通知时，其将检索其路径表寻找通过 D8 的路径，并发现到达 A 和 L/U1 的路径。经过 D8 到达 A 的路径将从其路径表中删除。存在另一个到达 A 的路径，其将不发送信息 **ROUTE_DISCON**。其将检查通过 D8 的 L/U1 的路径。因为所述路径具有有标记的最后的导向器，D8 的故障有效地终止了所有的路径。D2 将删除到 L/U1 的两个路径，并向上游导向器发出信息

ROUTE_DISCON.

(4) D1 将接收对于 L/U1 的信息 ROUTE_DISCON, 并删除在其路径表中的两个路径。因为所有的路径都被删除, 其将向上游导向器 D4, D6 发送信息 ROUTE_DISCON。

(5) D6 将接收来自其上游节点的信息 NODE_FAILED 和来自 D1 的信息 ROUTE_DISCON。如果其首先接收 NODE_FAILED 信息, 其将删除两个路径(因为 D8 的路径设置有最终的导向器标记), 并且不发出任何信息 ROUTE_DISCON(因为上游导向器是一个故障的导向器)。如果其首先接收信息 ROUTE_DISCON, 其将删除两个路径(不检查最后的导向器标记), 并确定其不需要发送信息 ROUTE_DISCON, 因为上游导向器是一个故障的导向器。当第二信息进入时, 路径已经被删除, 因而没有处理发生。

(6) D4 也将接收来自其上游节点的信息 NODE_FAULT。其将通过 D8 定位到 L/U1 的路径, 并检查最后的导向器标记。根据所述的标记, 节点将删除到 L/U1 的路径, 并向上游导向器 D2 发送信息。其将删除经过 D8 到 A 的路径。

(7) 如果 D2 已经删除其到 L/U1 的路径, 则来自 D4 的 ROUTE-DISCON 信息将使它们被删除。

虽然上面参照特定的实施例说明了本发明, 上述的说明并不限制本发明的范围。在所附权利要求限定的范围和构思内, 本领域技术人员可以作出各种改变和改型。

附录 A: 外部事件

输送系统外部事件			
事件	源	目的地	说明
CARRIER_INSTALLED	传输控制器	CIM	当材料由本地端口被传递到轨道上时由输送控制器发送。 由内部舱室 SEM 规定。
CARRIER_REMOVED	传输控制器	CIM	当材料由轨道被传递到负载端口上时由输送控制器发送。 由内部舱室 SEM 规定。
HOST_COMMAND	CIM	传输控制器	该事件从 CIM 系统发出，用于启动材料运动指令可能的指令值是： TRANSFER
HOST_COMMAND_INITIATED	传输控制器	CIM	响应信息 HOST COMMAND， 表示输送控制器已经收到并执行所述指令
REMOTE_COMMAND_SEND	传输控制器	CIM	内部舱室确定的信息，这些信息在运动状态机的状态转变时发出。
HOST_COMMAND_COMPLETE	传输控制器	CIM	当先前接收的完成时发出
事件报告发送	传输控制器	CIM	由输送控制器发出，表示基本操作的启动和完成。可能是事件包括： 正在获取，已经获取 正在移动，已被移动 存放，存放完成， 容器安装，容器被除去。

附录 B 内部 CLC 事件

Intra-CLC Events				
事件	源	目的地	多传递或 点传递	说明
ACQUIRE_MATERIAL	LAC 线程	LPTD 线程	点	该事件由 LAC 线程发出，用于命令 LPTD 线程开始把材料从规定的负载端口传递到规定的负载点。在这种信息中的信息是： 源 LAC 地址 负载端口 ID (或 LPTD 位置 ¹⁾) 负载点 ID (或 LPTD 位置 ²)
ALARM			多	产生报警事件把故障通知另一个系统 报警事件包括以下信息： 报警示例 主要报警代码 次要报警代码 响应选择： 终止 重试 忽略 无响应 报警特殊信息

¹ 该位置可以是一个 LPTD 可以理解的简单的数值，用于指示物理位置。用这种方式可以用整数规定两个负载端口和轨道位置。
² 同上

事件	源	目的地	多传递或点传递	说明
ALARM_RESPONSE				该事件要求响应的报警而产生。 其包含以下信息： 报警示例选择： 响终止试重忽略 运动方向
CARRIER_EXITED	区线程	邻区线程， LAC 线程	多	事件告知其它区容器已经离开 当前区，现在应当在下游区。 消息源地址 容器识别符
CARRIER_EXITING	区线程	邻区线程	多	事件告知当前区中的容器开始 离开当前区，预期合适的曲线。 消息源地址 容器内部材料 ID 速度
CARRIER_REMOVED	区线程	邻区线程	多	该事件当材料被从轨道上除去时被 当前区发出。这告知邻区从其路程表 中除去容器更新中的信息。 速度在消息地址 ID 内部材料最大速度 注意除去的同时，区仍然是不可利用的。

事件	源	目的地	多传递或 点传递	说明
CARRIER_STOPPED	区线程	邻区线程	多	该事件表示材料被减速以便停止在发出信息的区，所述信息可用于停止任何区区间定时或更新速度表，在消息中的信息： 源区地址 内部材料 ID
DELIVER_MATERIAL	LAC	LPTD		该事件对 LPTD 控制线程产生，命令其把容器从轨道移动到负载端口。该事件将含有以下信息： 内部运动 ID 负载端口识别符 材料位置识别符
DEPOSIT_MATERIAL	R.C.	S.C.		产生该事件用于命令输送机将材料放置到轨道上。
DEST_ANNOUNCE	目的地区	任一区		该信息被作为目的地的区产生。该信息被发送给直接上游区，并被逐个区地传送，直到其达到导向器。该信息必须包括发出该信息的目的地区的地址，并且每个区的计数段可以增加。 如果在预定的时间内区没有接收到响应，它将等待一个随机的时间间隔，然后再发送信息。
DEST_REGISTERED	导向器	目的地区		这个信息当导向器收到目的地区通告信息时由导向器发出。该信息告知目的地区其已被登记在导向器的路径表中。
FREE_LPTD	LAC	LPTD		一旦涉及 LPTD 的传递完成便产生该事件，其通知 LPTD 控制线程所述机构现在可以使用。
FREE_ZONE	LAC	区线程	点	发出该事件命令区对其自身自由标记。

事件	源	目的地	多传递或 点传递	说明
LPTD_NOT_RESERVED	LPTD 线程	LAC 线程	点	该事件由 LPTD 线程产生, 用于表示 LPTD 不能被请求者接收。
LPTD_RESERVED				
MATERIAL_ACQUIRED	LPTD 线程	LAC 线程	点	一旦 LPTD 线程完成材料向轨道上的转移, 便发出该事件。
MATERIAL_ARRIVED	区线程或 LAC 线程	TC 或 RC		发出该事件, 用于通知输送控制器材料已到达指定的预加载区。在事件中的信息: 内部材料 ID 目的地地址 该事件由区线程产生, 用于轨道到达, 并由 LAC 线程产生, 用于交付。
MATERIAL_DELIVERED	LPTD	LAC		一旦 LPTD 把材料成功地交付给负载端口便产生该事件。
MATERIAL_NOT_DELIVERED	LPTD	LAC		发出该事件用于指示指令未被成功地完成。
MATERIAL_MOVED	S.C.	R.C.		在成功地完成运动时发生该事件。 对于材料获取, 该事件表示材料在轨道上。
MATERIAL_NOT_SENT	负载区 控制器线程	S.C.	点	该事件由故障的轨道控制器发出, 表示材料从当前区向目的地的运动。事件包含: 源区地址, 内部材料识别 错误码
MATERIAL_SENT	负载区 控制器线程	S.C.	点	该事件由轨道控制器产生, 表示已经成功地材料送向目的地。该事件在容器行进到轨道检测器的末端时被发出。 事件包括: 源区地址 内部材料 ID

事件	源	目的地	多传递或 点传递	说明
MOVE_STATUS_UPDATE	S.C. LAC 线程	T.C.		该事件发送给 T.C., 用于提供特定运动的状态更新。 该事件包含: 发送器地址 内部材料识别符 状态
NODE_FAULT	任一节点	紧邻上游 节点		节点向其直接上游节点发送该信息, 直到被导向器 收到, 用于表示节点已经故障。该信息包含故障节 点的地址和对报告故障的节点最近的下游目的地。
PATH_LENGTH	任一	任一		导向器将其直接的下游邻区发送这个信息。该 信息被逐区传递, 直到其到达下一个导向器。当 每个区传递该信息时将增加路径长度。接收的导 向器也增加计数并向回导向器发送该信息。
POSITION_TO	LAC 线程	上游和下游 负载区线程	点	LAC 发送该信息命令两个区把材料从负载点移动 到下游负载区, 使得两个检测器检测到容器。
POSITIONED_TO	区线程	LAC 线程	点	当区完成定位时发出这个事件。
RESERVE_LPTD	LAC 线程	LPTD 线程	点	这个事件由 LAC 发送, 试图保留用于材料传递的 LPTD。该事件被包括在 LPTD 可以服务于多个负载 端口的情况下, 因此可能被一个以上的 LPTD 请求。
RESERVE_ZONE	LAC 线程	区线程		该事件被轨道控制器产生, 用于使区被保留。该事件 由负载区控制器产生。如果区在作为 LAC 的同 一个 RC 的控制下, 则通信是中间处理。如果在作为 LAC 的不同的 RC 的控制下, 则通信是中间处理器。

事件	源	目的地	多传递或点传递	说明
ROUTE_ANNOUNCE	导向器	导向器		在导向器收到新的目的地通知并更新其自身的路径表之后, 其将向其上游导向器发送这个信息。如果路径已经在路径表中存在, 则不发送信息。
ROUTE_CONT	传输控制器	导向器		这个信息就发送用于标记导向器的先前无效的出口路径有效。
ROUTE_DISCON	传输控制器	导向器		该信息被从传输控制器发送给导向器, 用于标记导向器的出口路径是不可利用的。
ROUTE_TABLE	导向器	导向器		该信息从导向器向在信息 ROUTE_TABLE_REQ 中识别的上游导向器发送。该信息包括发送导向器的完整的路径表。
ROUTE_TABLE_REQ	导向器	导向器		该信息由导向器向其下游导向器被发送, 用于请求那些导向器发送其路径表。如果在预定的时间间隔没有收到对该信息的响应, 则导向器应当等待一个随机的时间间隔并重新发送所述信息。
SEND_MATERIAL	LAC	Zone	点	该事件被发送并命令所述的区使容器开始运动。 该事件包括: 内部运动 ID 目的地地址 运动方向 (左右)
SET_PARAMETER	传输控制器	控制器		由传输控制器向控制器发送, 用于用新的值代替控制器的当前设置参数。 将增加不存在于控制器中的参数名。存在的参数名将改变其值。从控制器中除去具有 0 的新值的现有的参数名。
SHUTDOWN_CMD	传输控制器	控制器		事件告知所有的控制器立即把所有的容器减速到 0。
SHUTDOWN_STATUS	控制器	传输控制器		事件提供关于控制器停止处理的状态信息。

事件	源	目的地	多传递或 点传递	说明
ZONE_AVAILABLE	区线程	邻区线程, LAC 线程	多	区线程发出这个信息告知邻区其可利用性和最大进入速度。如果区中具有容器, 这事件命令接收器从其数据库中删除所述容器。
ZONE_FAULT	任一区线程	紧邻的 上游节点		在检测到节点的故障状态时节点将发送向其上游节点发送这个信息。该信息将被发送给导向器。该信息包括故障代码, 故障节点的地址, 和所述节点的最接近的下目的地。(该信息可以具有和信息 NODE FAULT 相同的形式)。
ZONE_IS_SAFE_NOTIFICATION	区线程	LAC 线程	点	一旦区确定其是安全时该事件便响应 NOTIFY_WHEN_SAFE_SEQ 事件被发出。
ZONE_RESERVED	目前区线程	邻区线程, LAC 线程	多	该事件被区发出, 用于通告其已被保留。 该事件将包含: 源区地址

附录 C: 本地 CAN 总线信息

在下面的说明中使用以下的缩写代表本地 CAN 总线信息/事件:

ZC 区控制器

AC 轴控制器

ICID 控制器

HC (E23) 握手控制器

XC SECS 控制器

TC 输送控制器

这些说明被分成 4 种类型的信息/事件: 一般信息/事件, 区控制器信息/事件, 握手控制器信息/事件, 和轴控制器信息/事件

1 一般的信息/事件

1.1 ACK

ACK 信息由智能驱动器板响应要求通知的指令即包含一个执行 ID 的指令发出。

1.2 ALARM

报警信息包含主代码, 次代码和选择的解释性的数据字段。主代码是一个系统的宽度, 16 位的值。次代码在主代码内是唯一的, 因此可以用 8 位表示。选择的数据字段将根据特定的报警而改变。

1.3 CONTROLLER_STATUS_REQUEST

CONTROLLER_STATUS_REQUEST 发出指令用于询问控制器当前的状态。控制器利用在每个控制器中的规定的状态信息进行响应, 例如区控制器利用信息 ZONE_STATUS 响应。

1.4 FORCE_RESET

FORCE_RESET 发出指令用于强迫控制器再启动。属于该指令的数据确定控制器是否使用缺省值即被存储在非易失存储器中的控制参数尤其是其 CAN 地址被复位。

1.5 GET_PARAMETER

GET_PARAMETER 信息由 CLC 发出用于读取由智能控制器存

储的各个控制参数，例如驱动器板的 CAN 地址。利用这个指令读取的参数和控制板的类型有关。要被读取的参数由唯一的 2 字节上规定。该数由在前两位数据字节中给出，其相应于 SET_PARAMETER 指令使用的值。智能驱动器检查参数 ID 的有效性，如果 ID 值对控制器不合适则发出报警。

1.6 I_AM_ALIVE

I_AM_ALIVE 信息在自举操作之后，当驱动器板完成其初始化时或者响应 PING 或信息被发出。自举可能由于电源复位，手动复位，监视器计数器复位，或者被指令强迫的复位。每个区的控制器板的监视器时限是 630 毫秒。

1.7 PARAMETER_RESPONSE

PARAMETER_RESPONSE 信息由控制器响应先前的来自 CLC 的指令 GET_PARAMETER 被发出。被返回的参数由唯一的在前 2 个数据字节字节给出的 2 字节数规定。参数的当前值在随后的 6 个字节中给出。小于 6 字节的值以最小有效字节被返回。ASCII 数据可以在这种信息中被返回。可以返回多达 6 个 ASCII 字符。如果小于 6 个字符，则字符串以 0 结束。如果返回 6 个字符，则调用程序负责 0 结束的字符串。参数的 ID 值和 SET_PARAMETER 指令中给出的相同。

1.8 PING

PING 信息由 CLC 发出，用于快速询问具有给定的网络地址的智能驱动器板的存在。如果驱动器板存在并正在工作，其将利用信息 I_AM_ALIVE 响应。

1.9 SELF_TEST

SELF_TEST 信息由 CLC 发出，用于强制驱动器板执行其内装的诊断功能。在维修期间，作为系统安装处理的一部分，或者在系统脱机排除故障的期间内执行自测试。这些测试一般需要大量的时间（从计算机的观点看），其时间长度根据控制板的类型而不同。信息 SELF_TEST 包括 6 个参数，其意义根据不同的测试而不同。

不用的参数也被发出，因而该信息总是具有 8 个字节的数据。

1.10 SELF_TEST_RESULTS

SELF_TEST_RESULTS 信息由智能驱动器板响应信息发出。

1.11 SENSOR_STATUE

SENSOR_STATUE 信息由控制器响应指令 SENSOR_STATUS_REQUEST 发出。此外，大多数控制器当其检测器改变状态时自动地发出这些信息。例如区控制器当其左、右检测器改变状态时产生所述信息。

1.12 SENSOR_STATUS_REQUEST

发出这个指令意义询问控制器的一个检测器的当前状态。控制器以 SENSOR_STATUS 信息进行响应。

1.13 SET_PARAMETER

SET_PARAMETER 信息由 CLC 发出，用于设置或复位智能驱动器的各个控制参数，例如驱动器板的 CAN 地址。可以利用这个指令设置的参数因控制板的类型而不同。大多数参数被存储在控制器的非易失存储器中。要被设置的参数由唯一的 2 字节数识别，其在前 2 个数据字节中给出，在随后的 6 个字节给出要被设置的值。智能驱动器检查参数的有效性，并且当 a)如果参数的 ID 值对控制器不合适，或 b)如果参数超出范围，则发出 ALARM 信息。

2. 区控制器信息

2.1 EXECUTE

EXECUTE 信息告知区控制器执行两个可能的电动机曲线。所有存储的曲线都在区控制器队中出现。该指令用于同步多个区控制器，使得它们在同一个速度下运行同一个曲线。

EXECUTE 指令对一组区控制器编址，其中包括在指令之前发出的执行 ID 数，其位于信息 EXECUTE 的头部的目的地址部分。

2.2 EMERGENCY_DECEL

EMERGENCY_DECEL 信息用于尽可能快地使电动机减速，从当前速度到平滑停止。该信息使电动机在 3 倍于其正常速度下减速。

2.3 FLUSH_QUEUE

FLUSH_QUEUE 指令用于从区控制器的指令队中删除所有现有的指令 **SET_PROFILE**（包括等待 **EXECUTE** 指令的那些指令）。当前运行的曲线将继续运行，只有尚未运行的曲线被删除。**CLC** 不能使用这些指令停止电动机。如果没有曲线等待执行，则该指令无效。

2.4 MOTOR_IS_OFF

当电动机停止或断电时（自由转动，无维持电流）**MOTOR_IS_OFF** 信息被发送给 **CLC**。该指令一般表示指令 **MOTOR_OFF** 完成。该信息在以下条件下被发出：

响应 **MOTOR_OFF** 指令。

电动机被减速到 0。

由于电动机停止曲线而使电动机停止。

2.5 MOTOR_OFF

MOTOR_OFF 指令停止当前运行的曲线，刷新任何排队的曲线，使电动机凭惯性转动（无电流）。该指令被收到时尽快地被执行。如果该指令应当在完成现有的曲线之后再执行，则应当发出规定电动机停止曲线的指令 **SET_PROFILE**。

2.6 MOVE_N

MOVE_N 信息指示区控制器板使电动机转动一个十分之几毫米的数量。该信息用于使容器精确地定位，通常将其置于通过两个区定位的负载端口的前方。

2.7 MOVE_N_NODE

MOVE_N_NODE 信息当区控制器完成 **MOVE_N** 操作时由区控制器发出。

2.8 POSITION_REQUEST

POSITION_REQUEST 指令用于相对于任何一个区检测器的最后的转变使在区中的容器到达当前位置。控制器利用信息 **POSITION_RESPONSE** 进行响应。

2.9 POSITION_RESPONSE

POSITION_RESPONSE 信息由于先前的指令 **POSITION_REQUEST** 而被发出。在响应中的数据字段给出容器相对于最近改变的检测器（左、右）的最后转换的当前位置。所述位置以十分之几毫米的整数值给出。如果容器在区中向左运动，则所述的值是负。

2.10 ROTATION_UPDATE

ROTATION_UPDATE 指令当控制器转过由先前的指令 **SET_PARAMETER** 规定的数量时被发出。区控制器具有相应于每隔 5 个厘米一个的缺省值。如果电动机停止，这些信息也停止。

注意如果轮子停止转动或者不能以区控制器指令的预定的速度转动区控制器则发出 **ALARM** 信息。

2.11 SET_CONSTANT_SPEED

SET_CONSTANT_SPEED 指令通知区控制器在给定的每秒毫米的速度下使电动机转动，例如对于 56.7 cps 的速度，其值应当是 567。这个指令的执行可以用和指令 **SET_PROFILE** 相同的方式被触发，即“执行触发器”字段可以具有和在 **SET_PROFILE** 的说明中给定的相同的值。

2.12 SET_PROFILE_PARAM

SET_PROFILE_PARAM 信息复位当前的曲线持续时间，a0 和 a1 参数为由 CAN 信息规定的值。

2.13 GET_PROFILE_PARAM

GET_PROFILE_PARAM 信息要求区控制器规定其自身的有效参数值。

2.14 PROFILE_PARAM_RESP

PROFILE_PARAM_RESP 在 **GET_PROFILE_PARAM** 之后被发出。该信息含有要求的曲线的有效值。

2.15 SET_PROFILE

信息 **SET_PROFILE** 通知区控制器执行哪一个电动机曲线，并

在所述曲线开始执行时进行选择。

2.16 ZONE_STATUS

信息 ZONE_STATUS 由区控制器响应指令 CONTROL_STATUS_REQUEST 被发出。

3.1 握手控制器

3.1.1 BUSY_SET

这是一个对于 CLC 的响应信息，表示 E23 忙被设置。

3.1.2 COMPLETE_HANDSHAKE

这个来自 CLC 的指令命令控制器完成当前进行的握手。

3.1.3 HANDSHAKE_COMPLETE

当控制器成功地完成握手之后这信息由控制器发出。

3.1.4 INITIATE_HANDSHAKE

这指令由 CLC 发出，命令控制器和其它的负载端口和连接装置建立握手。

3.1.5 LP_INIT_LOAD_REQ

该指令由控制器在检测到负载端口或其它连接装置开始请求其加载新的材料时向 CLC 发出。

3.1.6 LP_UNLOAD_REQ

该信息由控制器在检测到负载端口或其它连接装置开始请求其下载材料时向 CLC 发出。

3.1.7 SET_BUSY

该指令由 CLC 向控制器发出，命令控制器设置忙状态，表示操作正在进行。

3.2 轴控制器信息/事件

3.2.1 AXIS_STATUS

轴控制器发出的状态信息。

3.2.1.1 DEFINE_LOCATION

该指令由 CLC 向轴控制器发出，用于规定轴的当前位置。接着可以发出指令 GOTO，以便移动到这个位置。

3.2.2 FIND_HOME

发出该 FIND_HOME 指令使转动器达到其原始位置。

3.2.3 FIND_POSITION

当轴的当前位置未知时由 CLC 发出该 FIND_POSITION 指令。这可以由于在轴的转动期间发生电源故障引起。CLC 必须规定电动机的方向。可以规定总的距离。FIND_POSITION 指令将搜索先前规定的位置。在执行该指令之前轴必须在原始位置。

3.2.4 GOTO

该指令引起控制器转动到规定的位置、这将计算所需的微步数。

3.2.5 MOTOR_IS_OFF

该信息在电动机停止时被发出。

3.2.6 MOTOR_OFF

该信息被发出用于立即切断电动机的电源。

3.2.7 MOTOR_N

该信息使检测器运动一个规定的距离。该格式与区控制器 MOVE_N 的相同（数据有不同含义）。

3.2.8 MOVE_N_DONE

在完成 MOVE_N 信息时发出该信息。格式和区控制器的相同。

3.2.9 POSITION_REQUEST

该信息请求被规定为位置检测器的检测器被读出并被向回通知。

3.2.10 POSITION_RESPONSE

该信息响应 POSITION_REQUEST 被发出。位置检测器的状态在位变换字段中表示（以便使得在故障情况下能够读出多个检测器）。

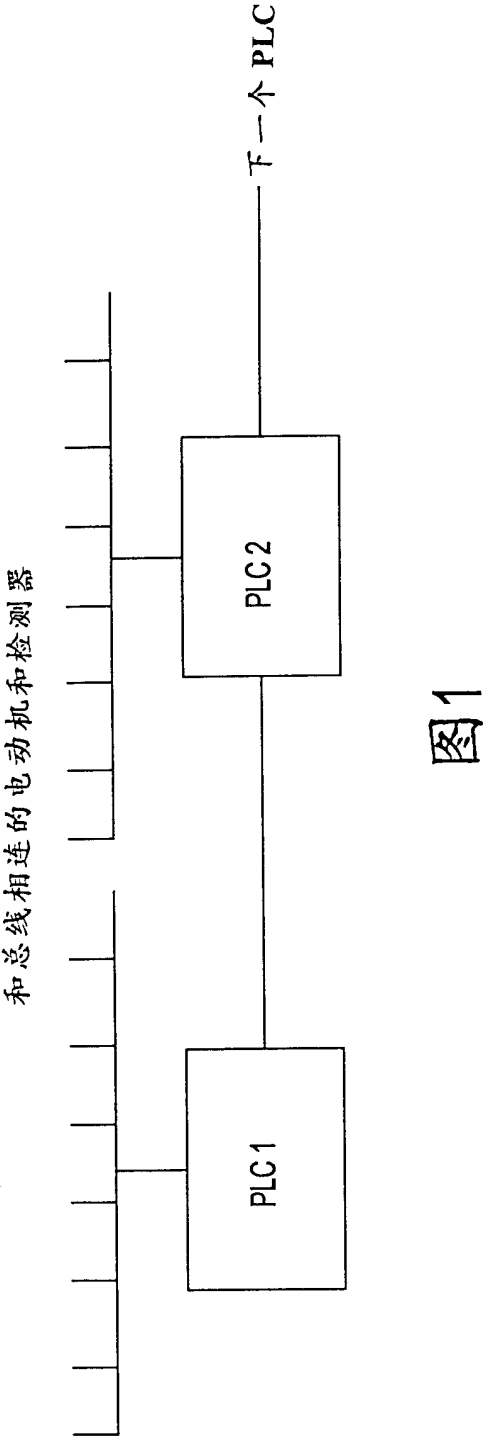


图1

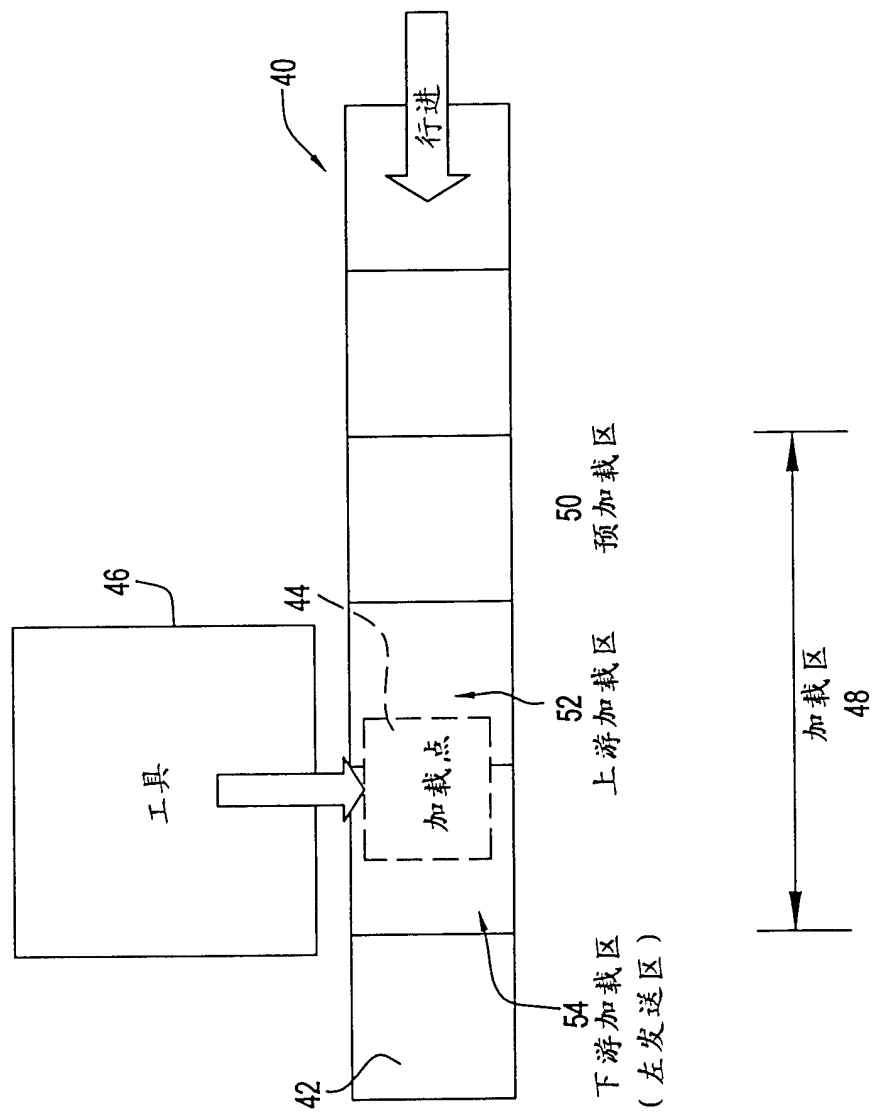


图 2A

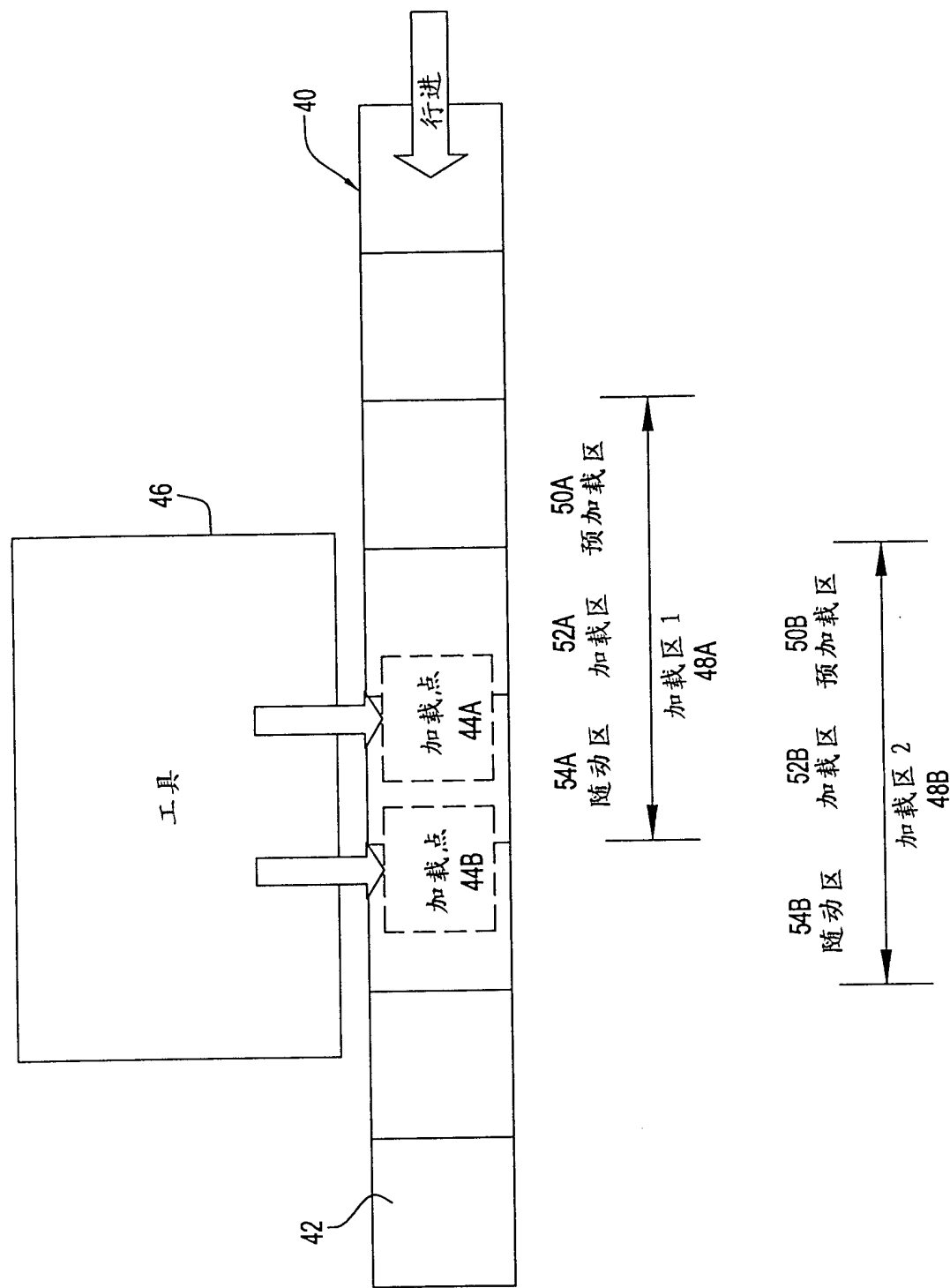


图2B

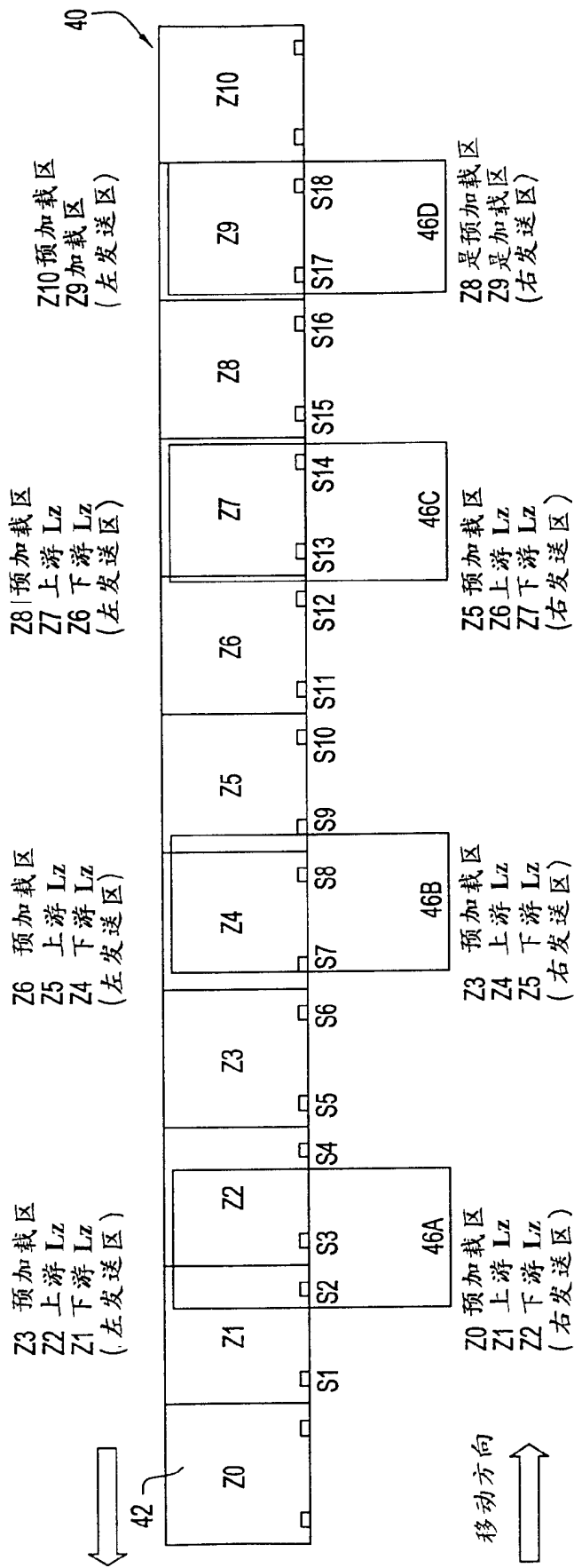


图2C

轨道区的邻域

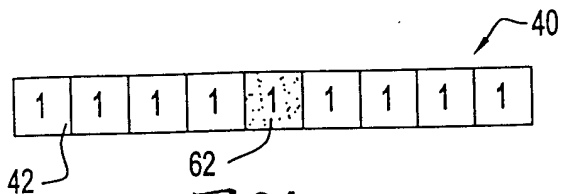
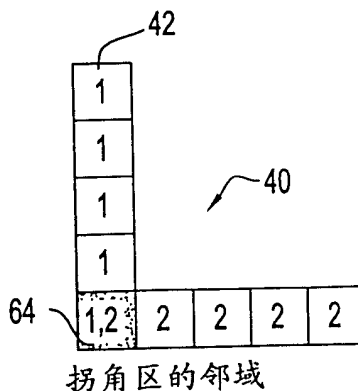
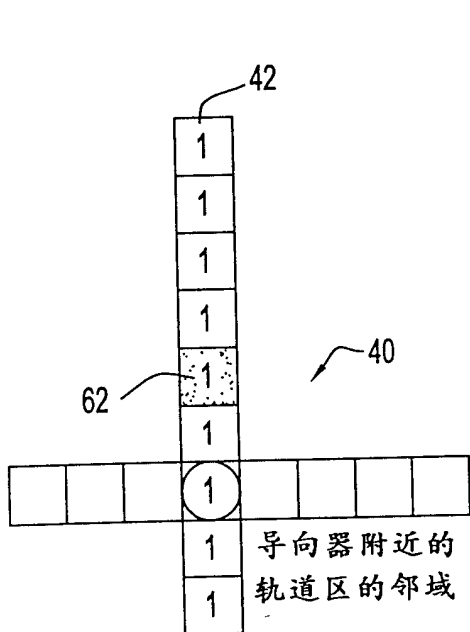


图3A



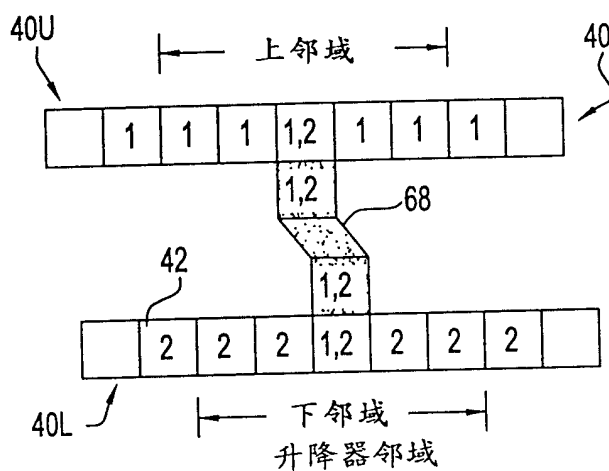
拐角区的邻域

图3B



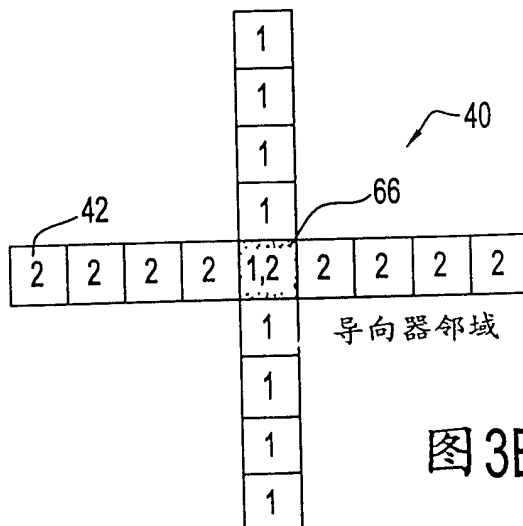
导向器附近的
轨道区的邻域

图3C



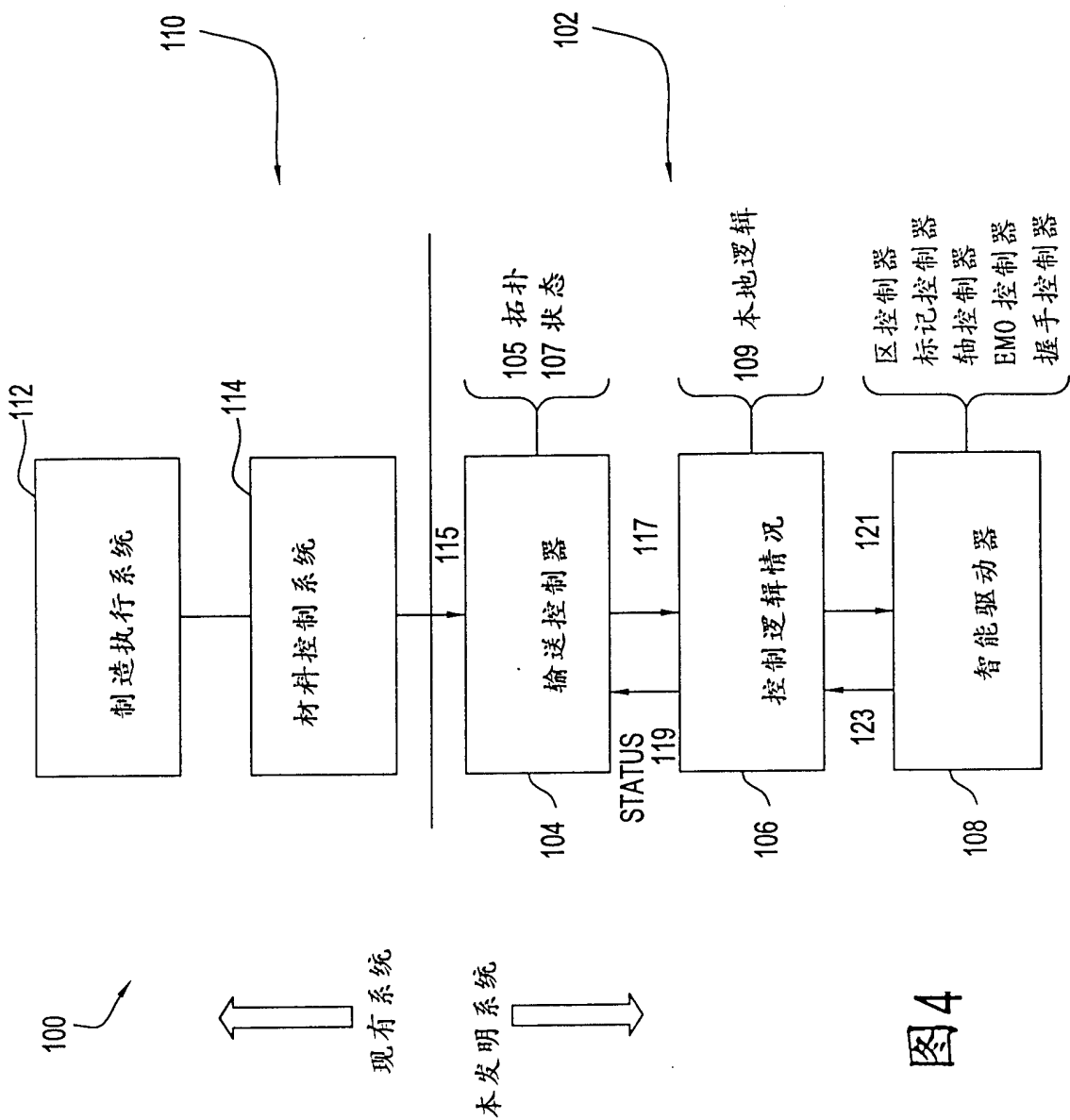
升降器邻域

图3D



导向器邻域

图3E



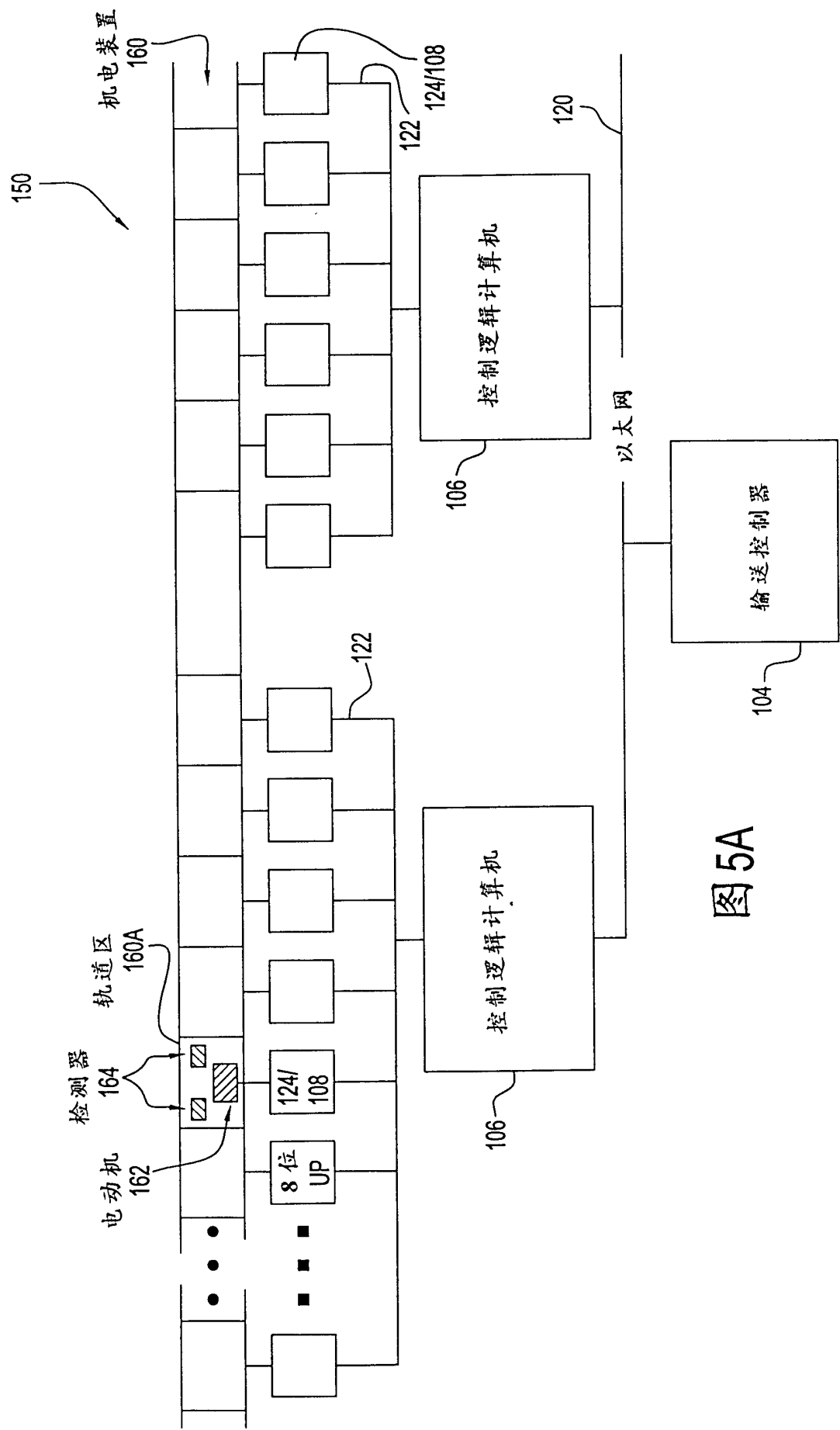


图 5A

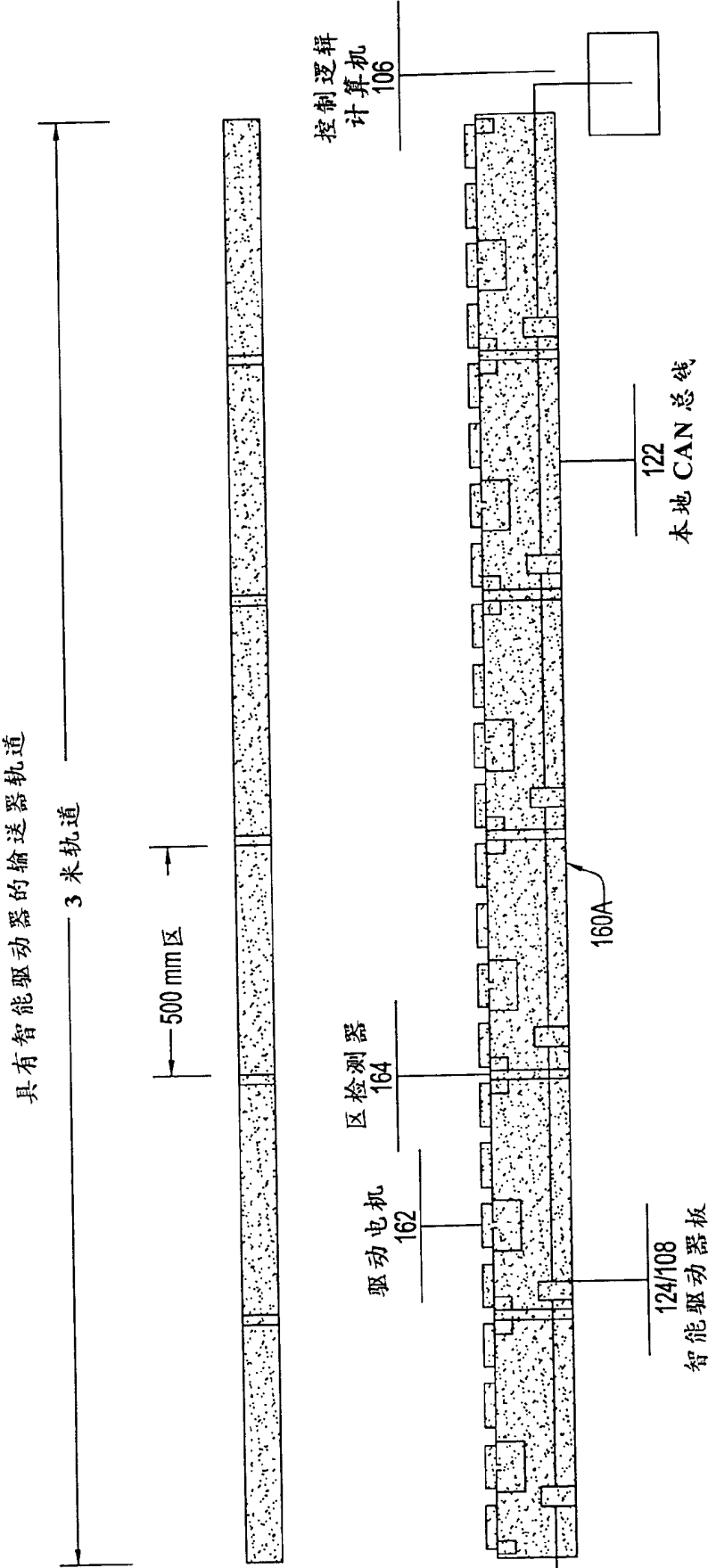


图 5B

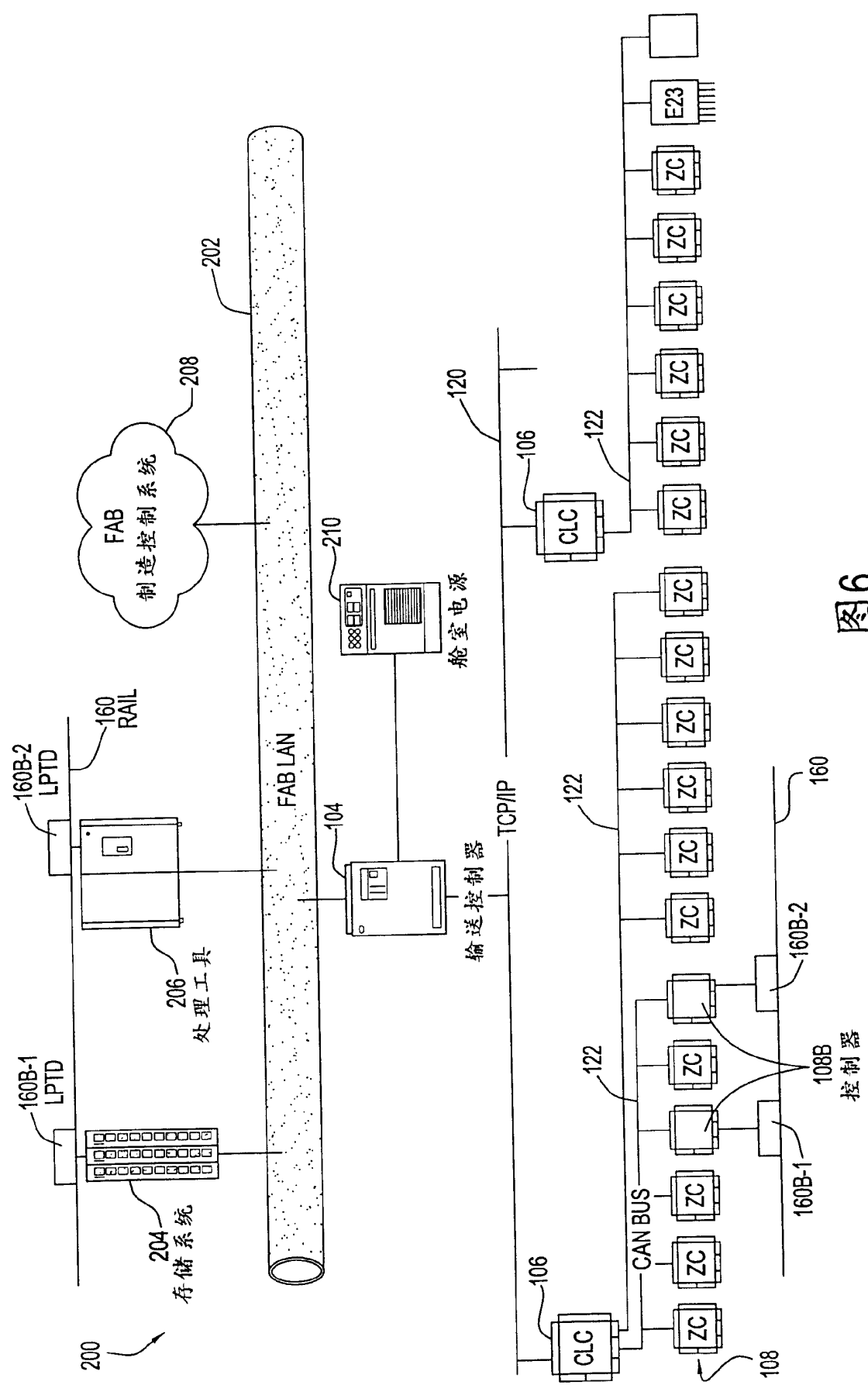


图6

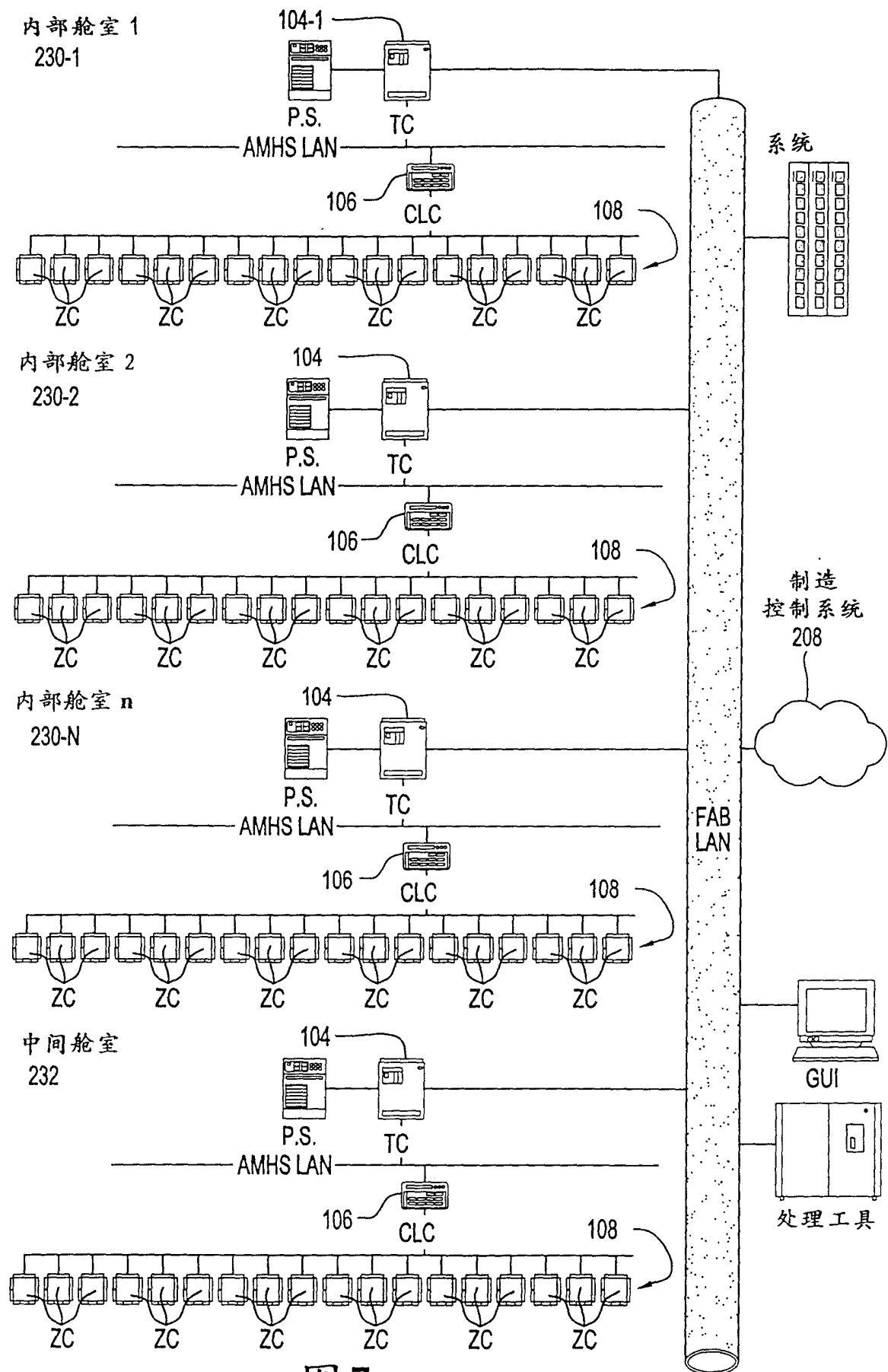


图 7

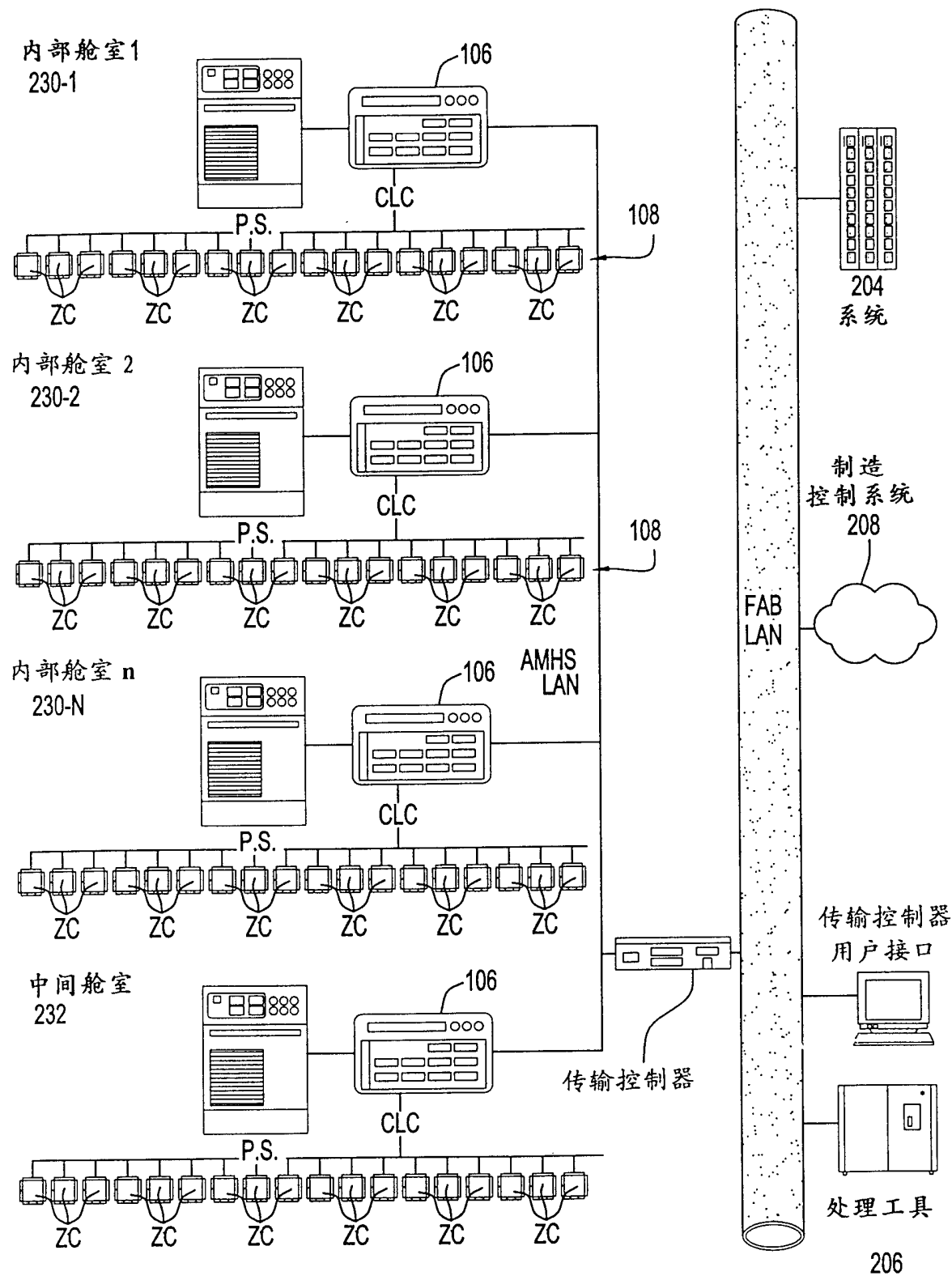


图8

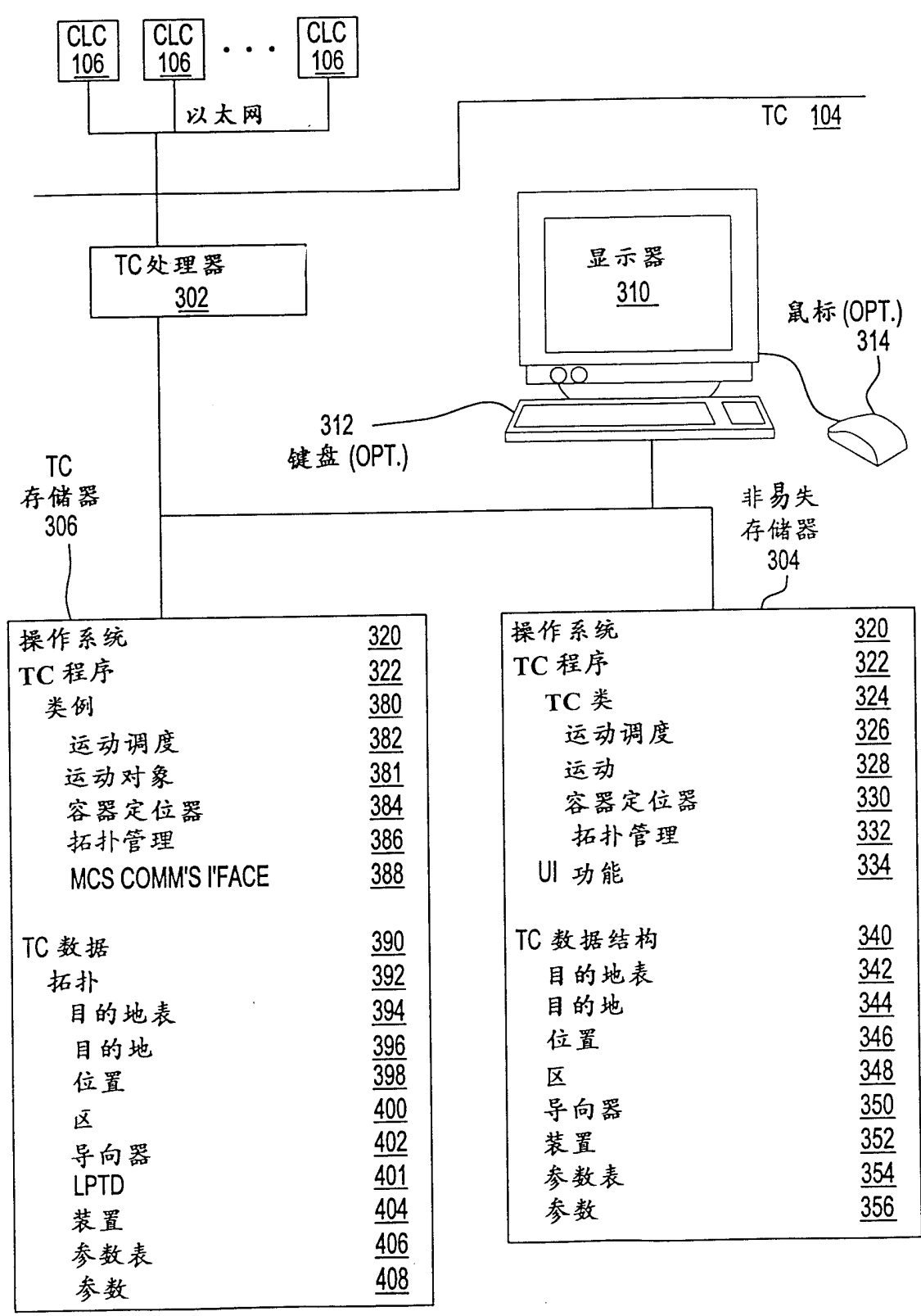


图9A

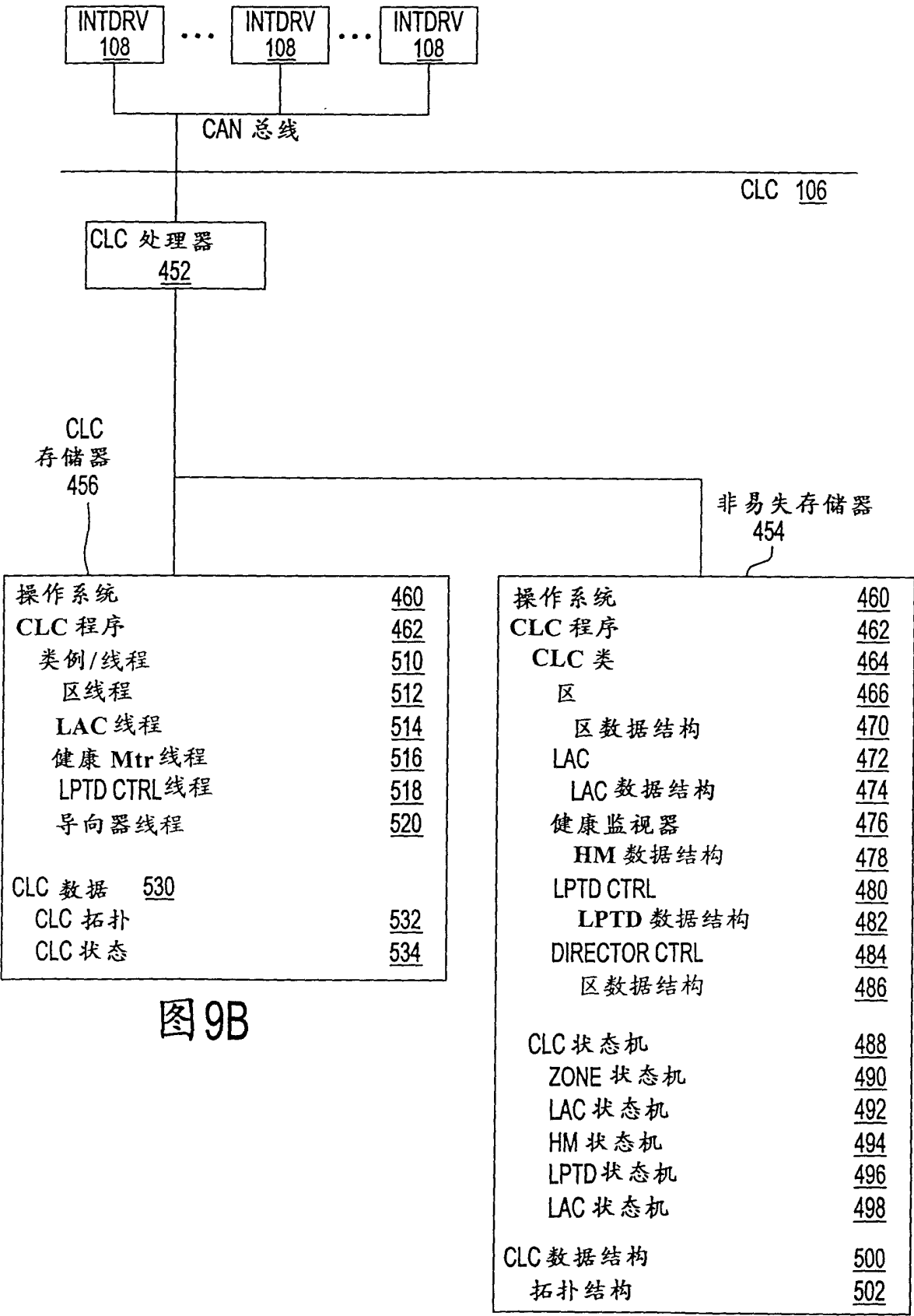
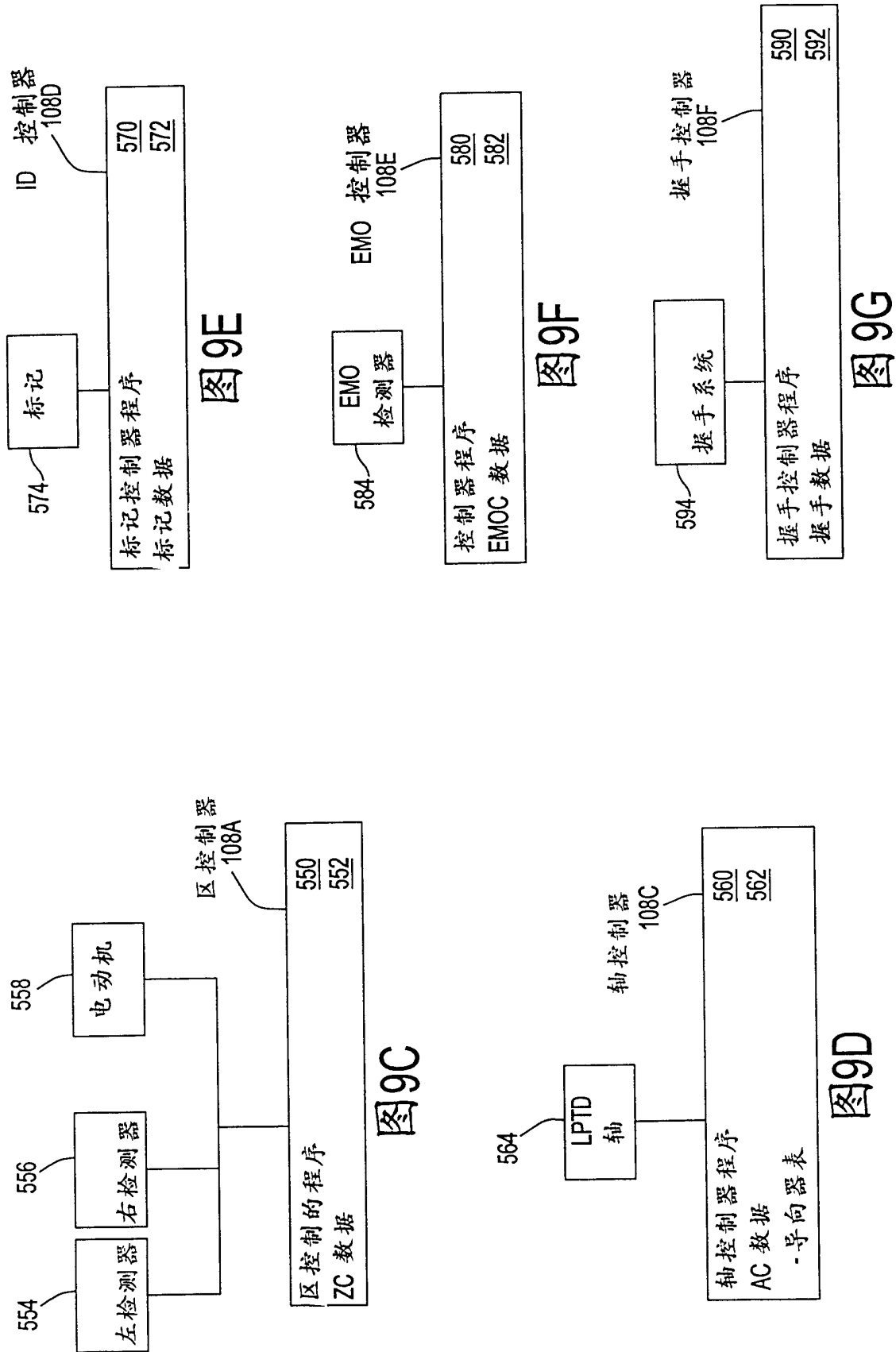


图 9B



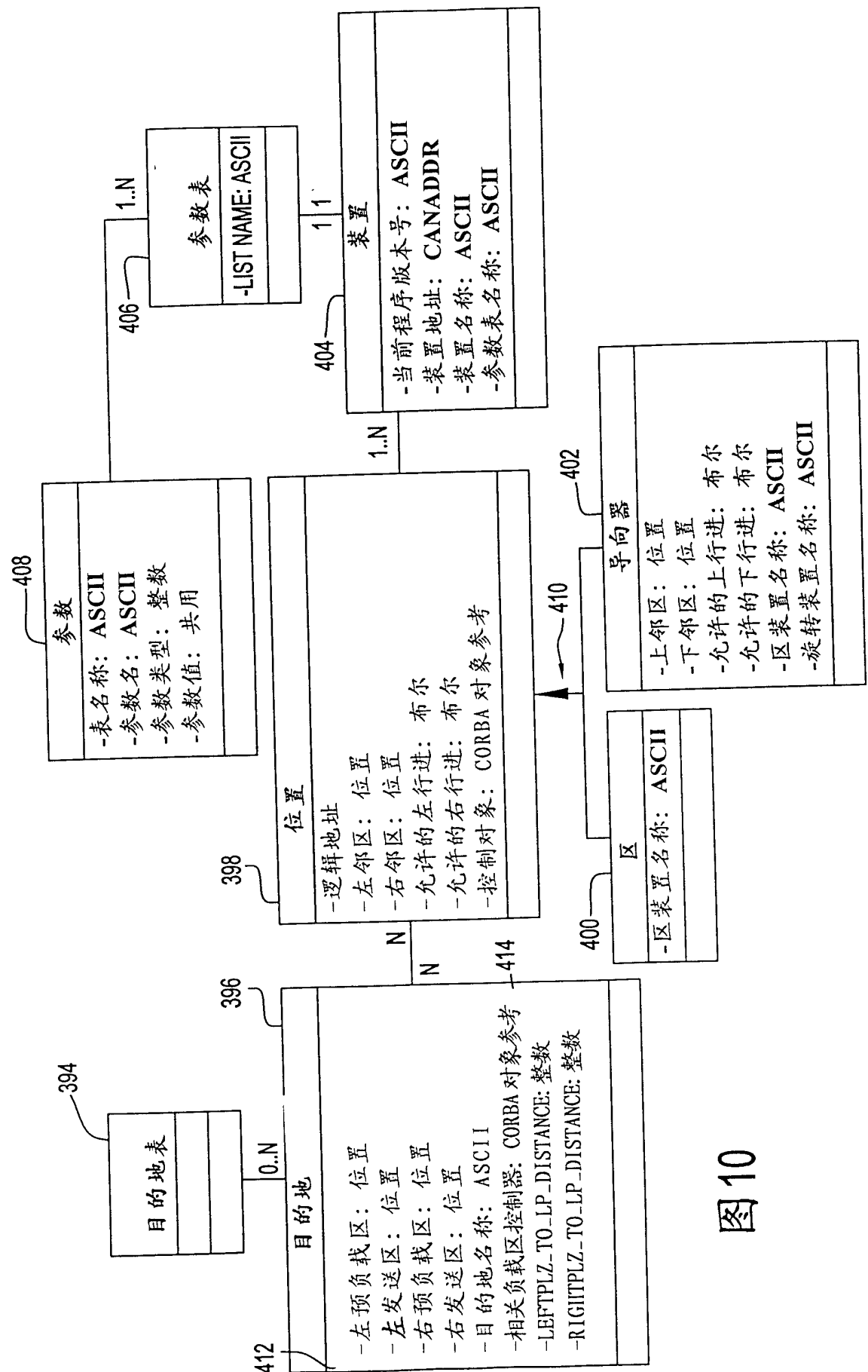
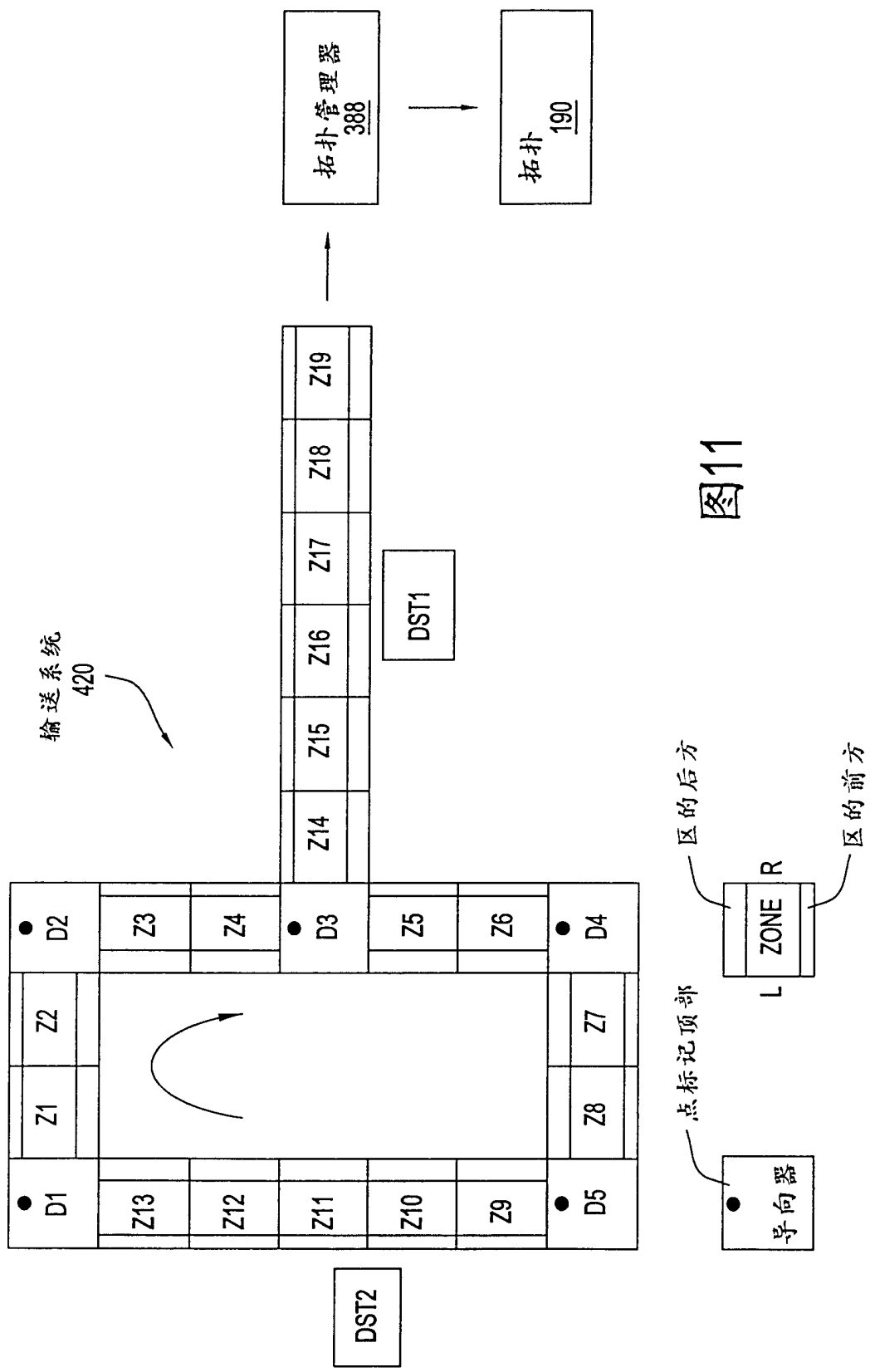


图10



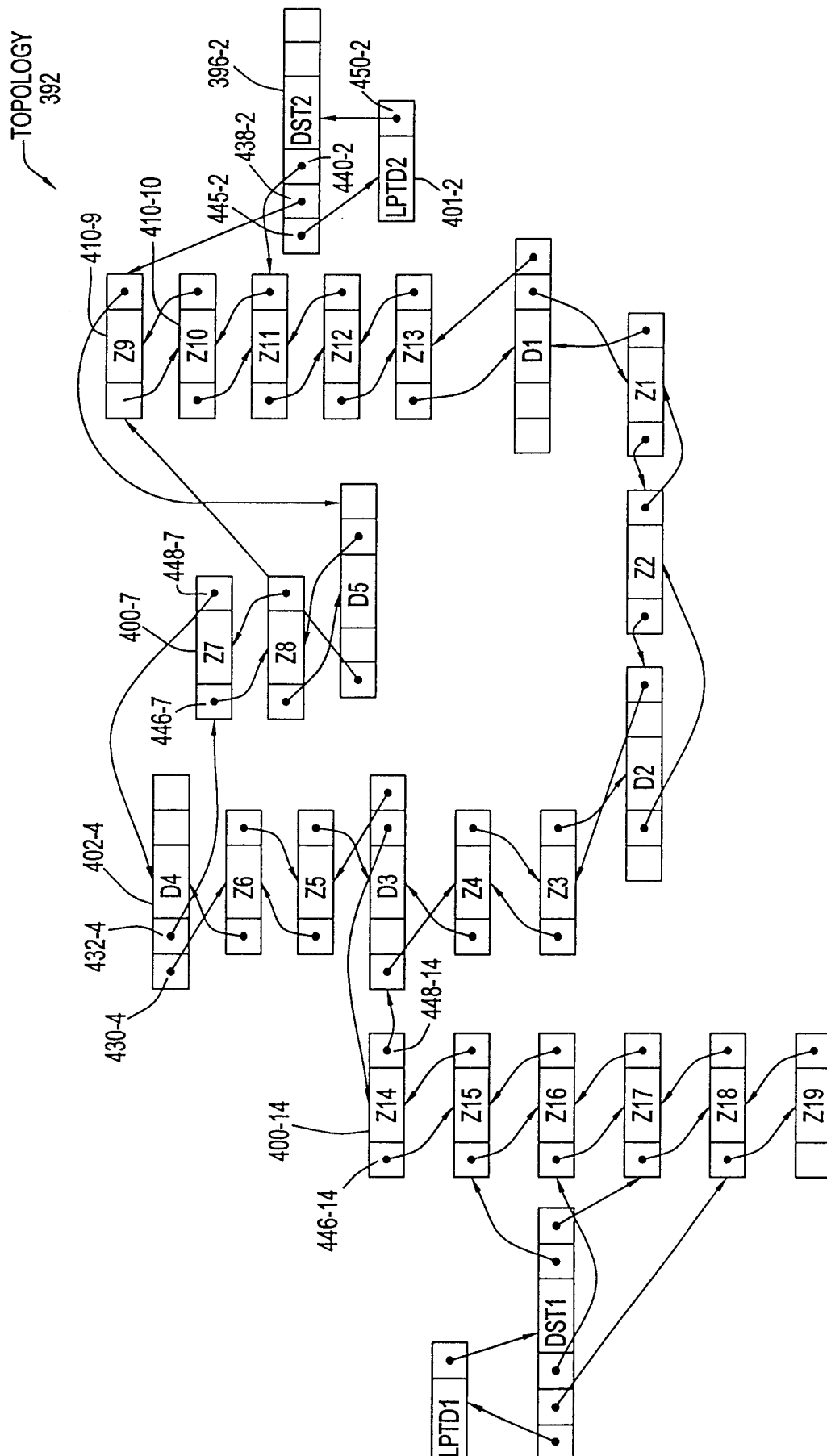


图12A

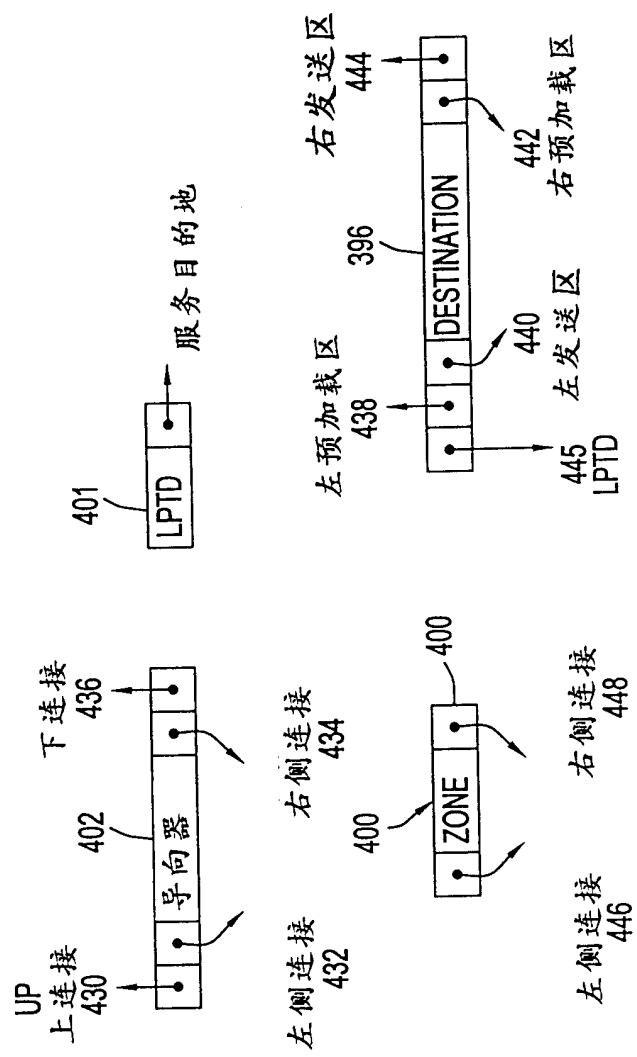


图12B

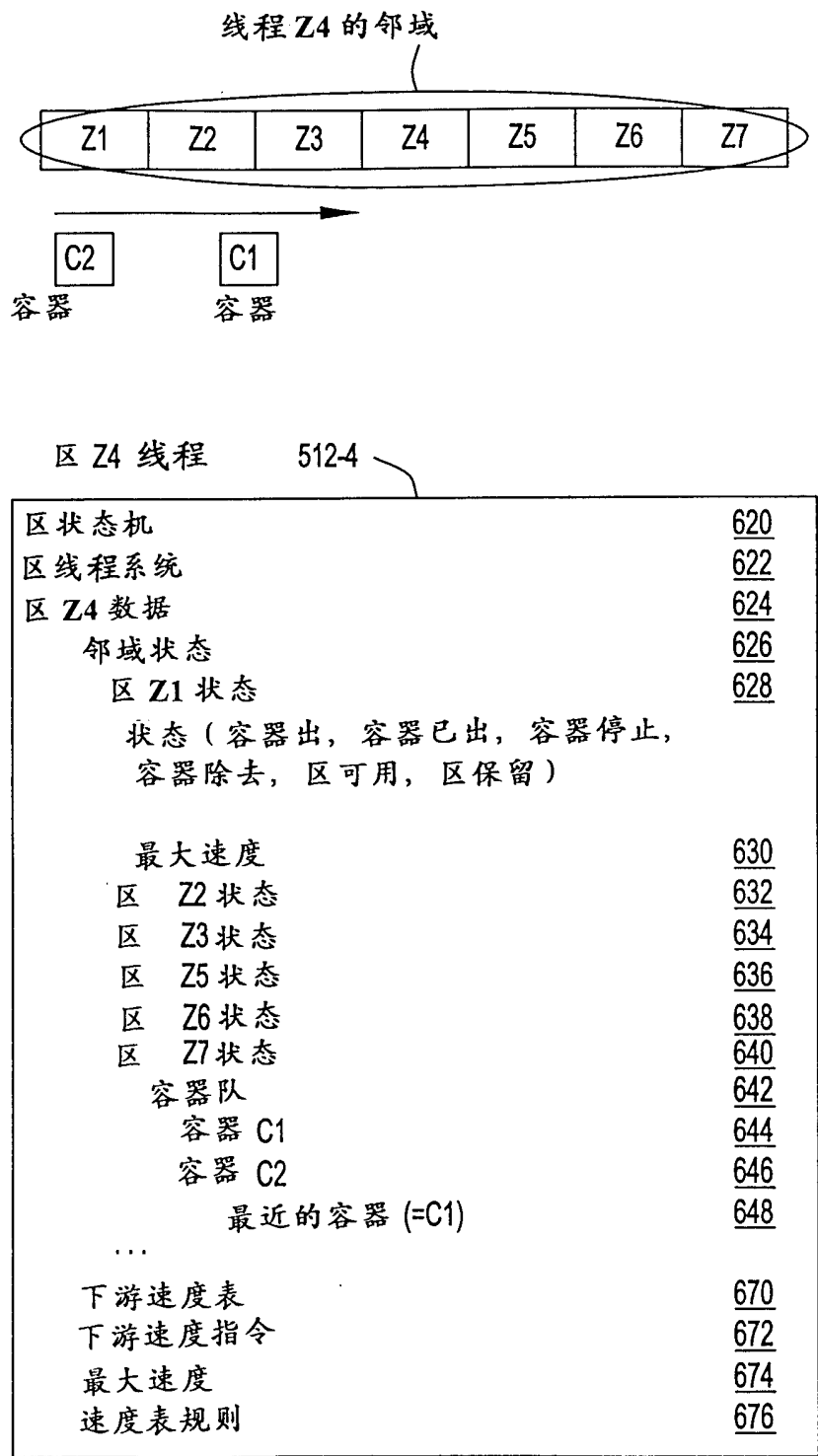


图13

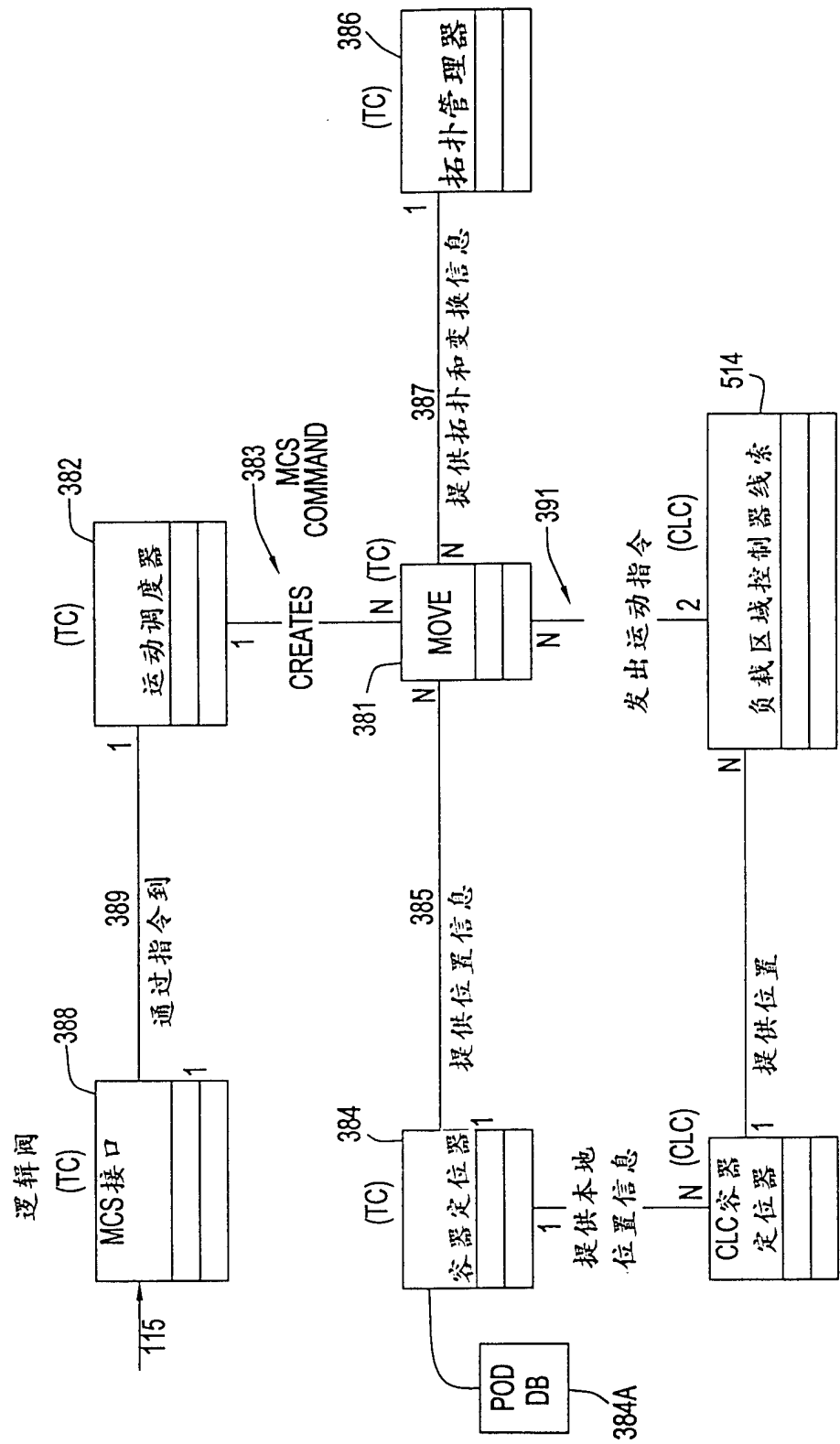
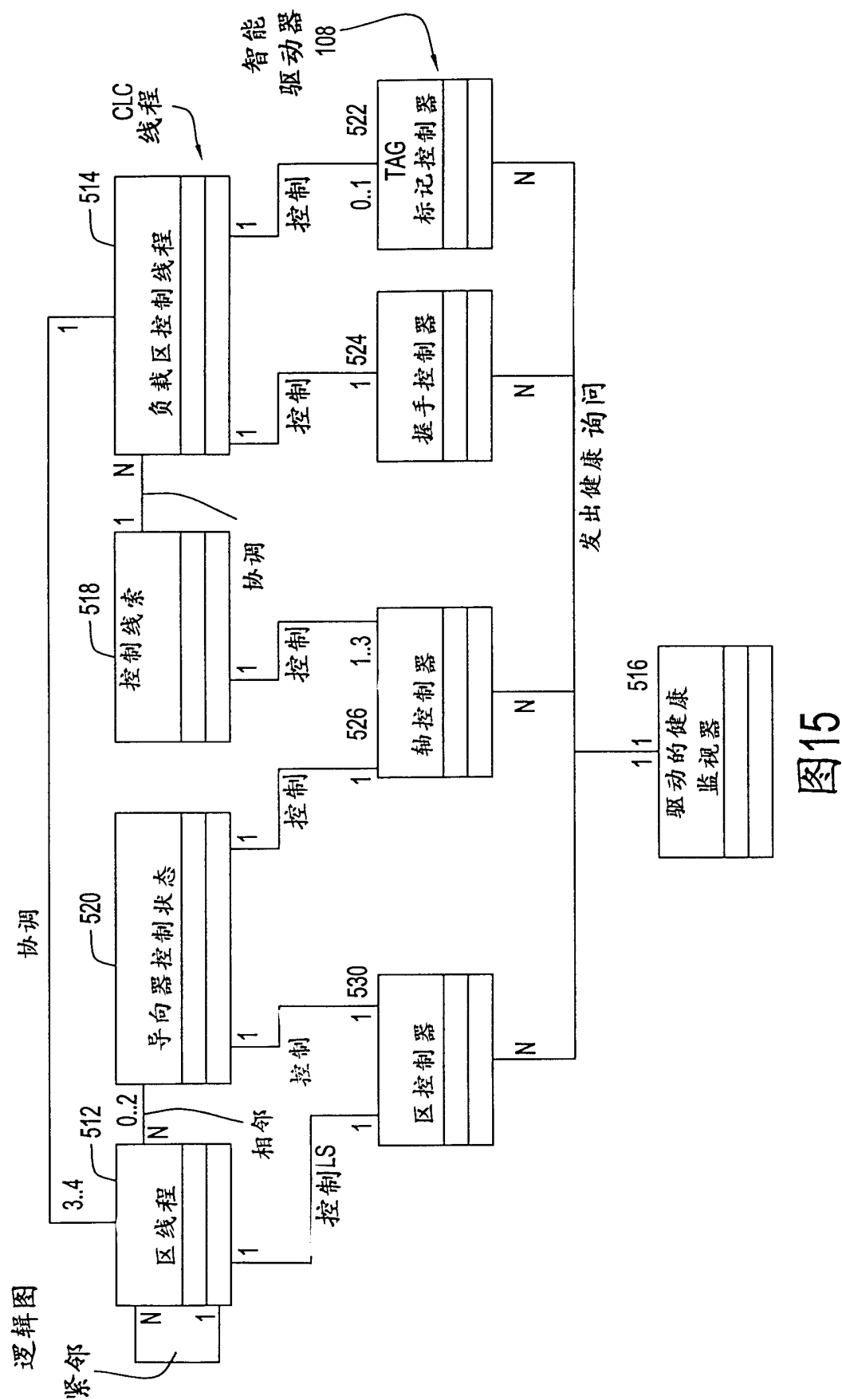
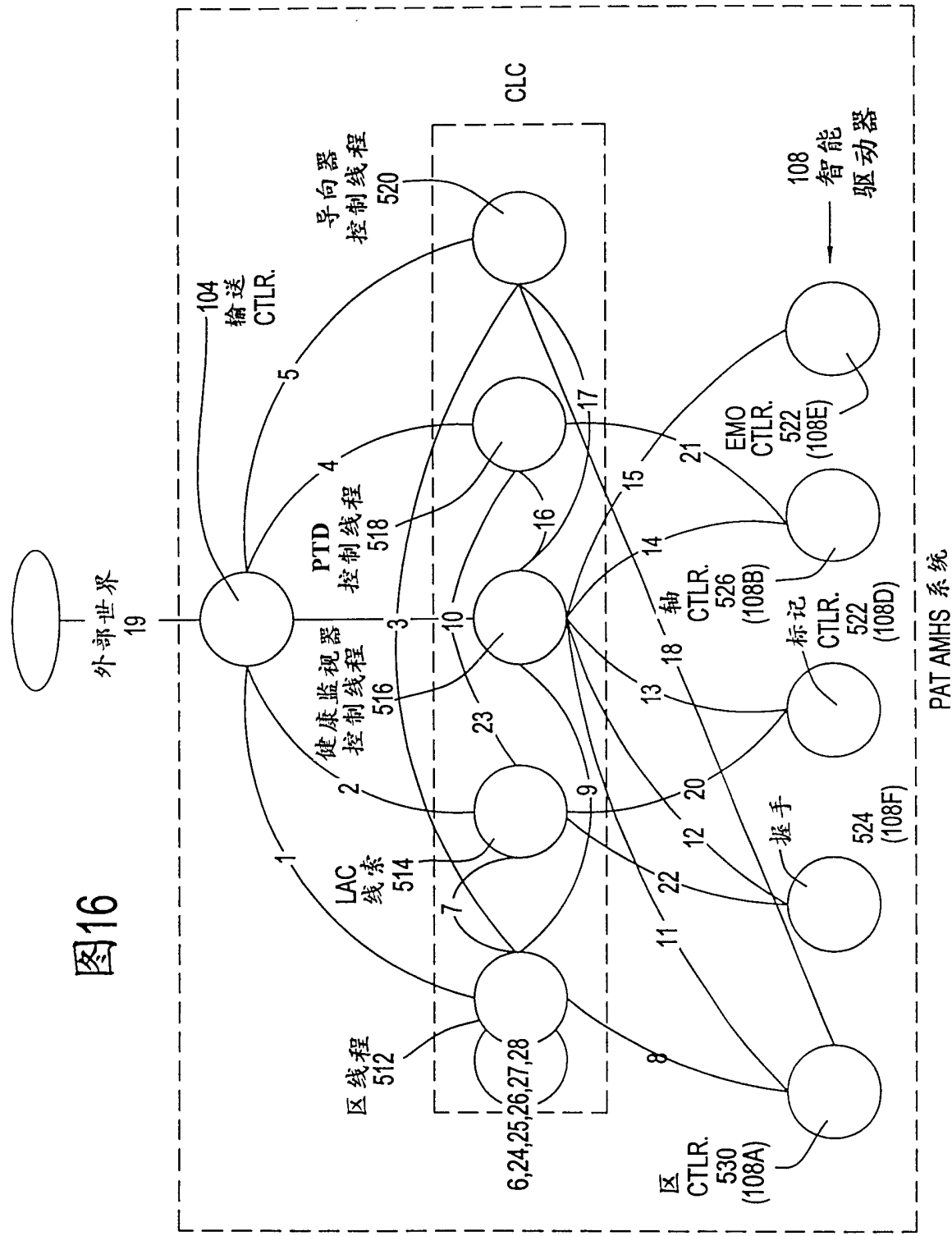


图14





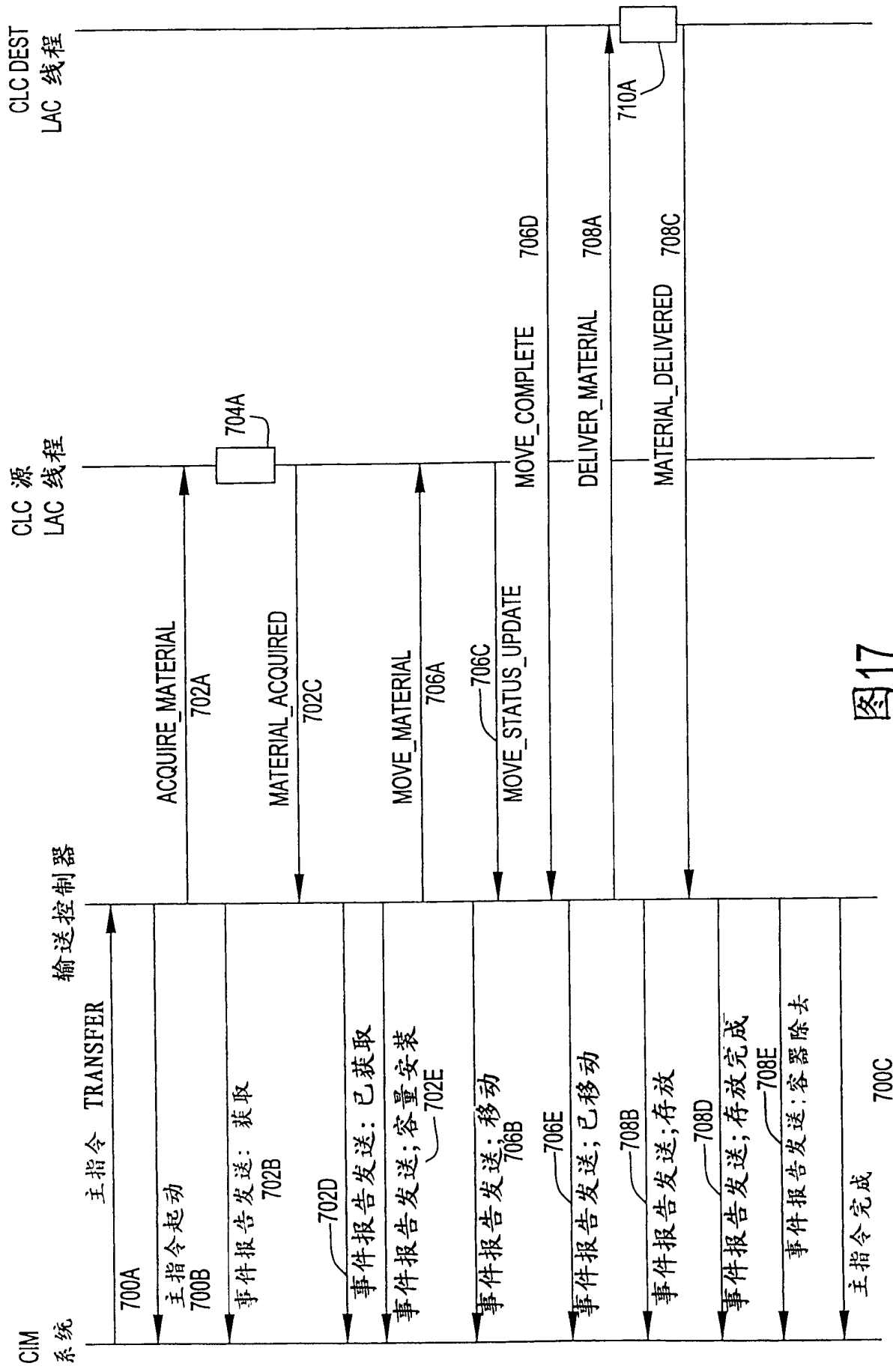


图17

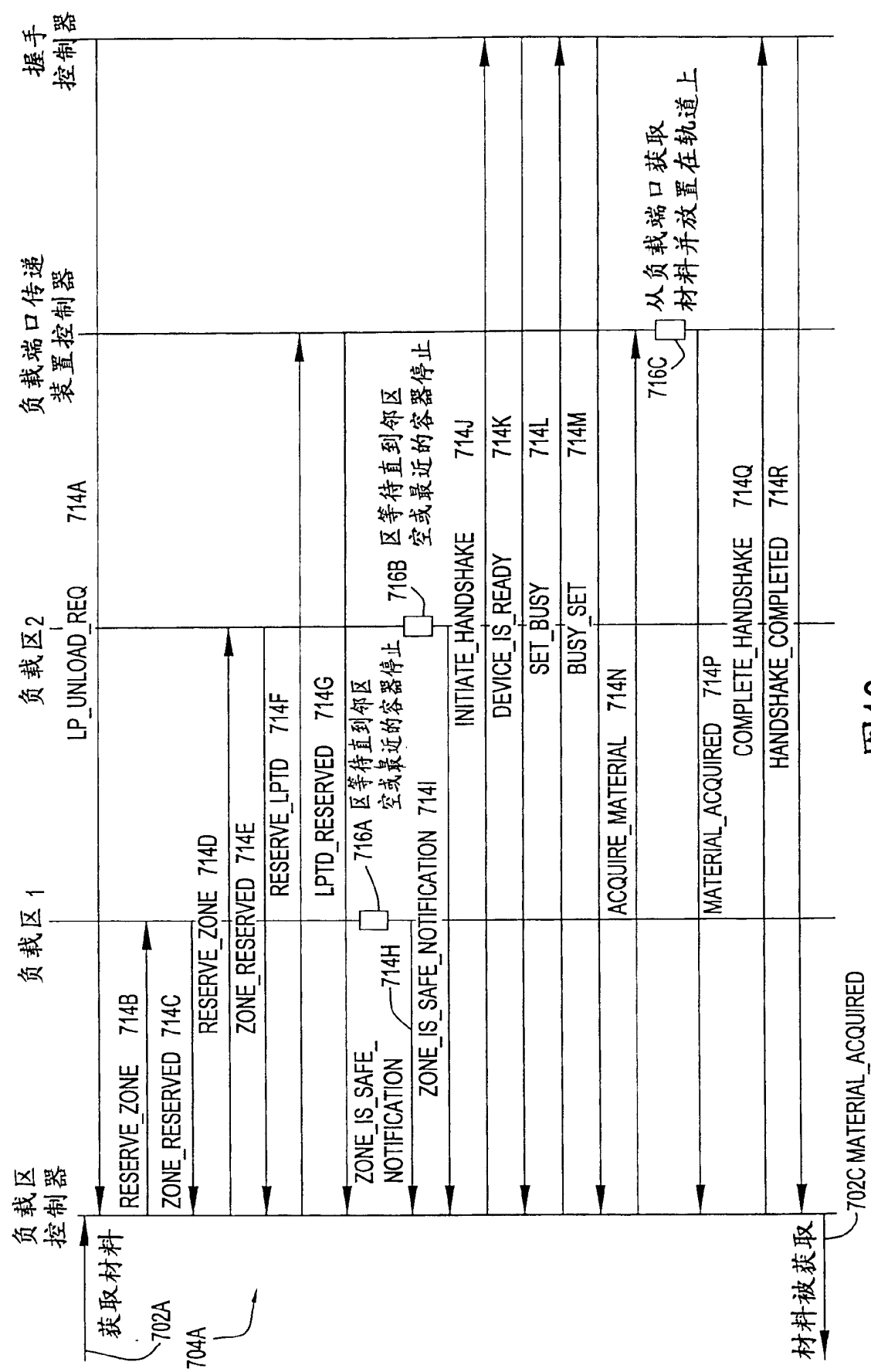
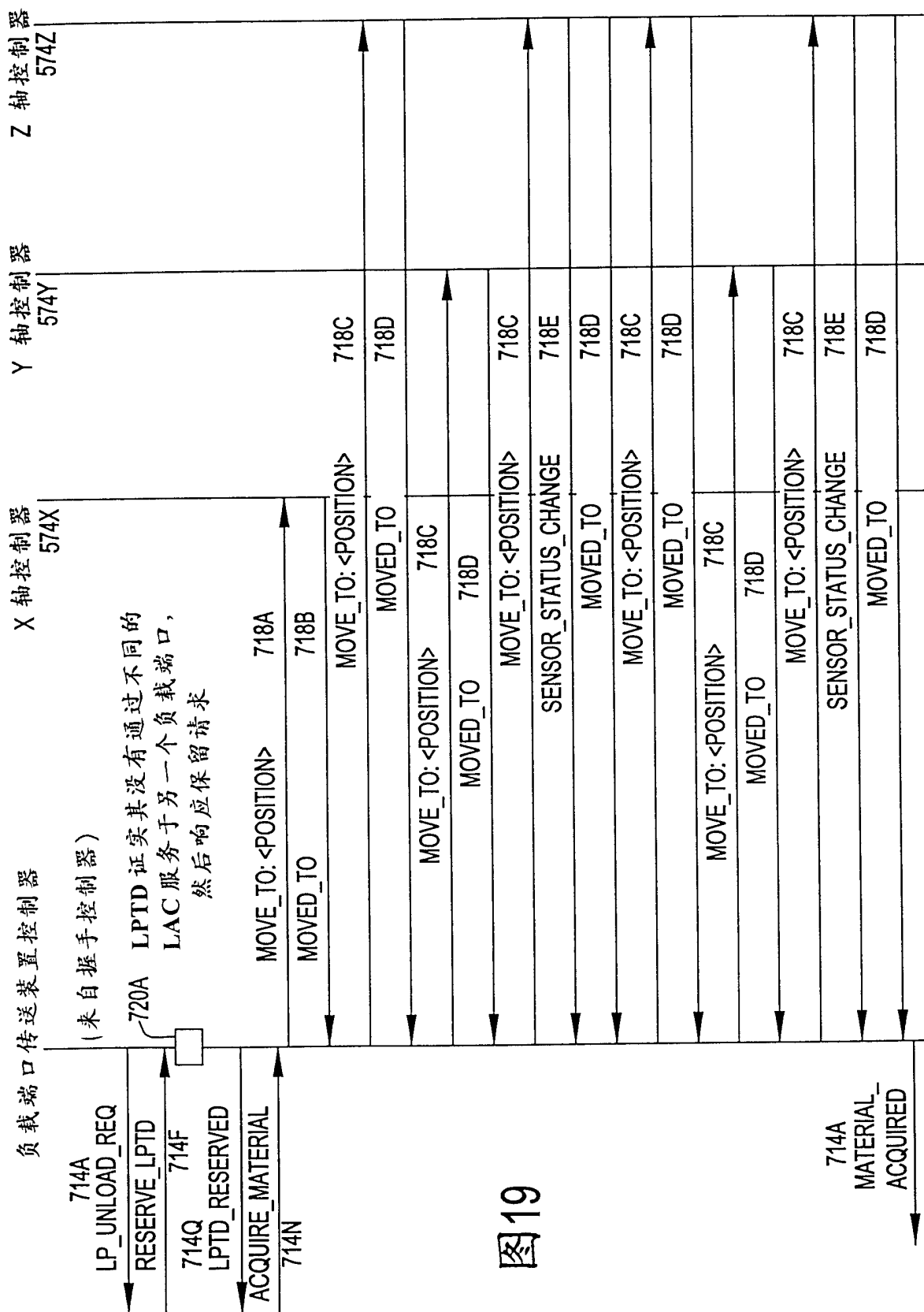


图18



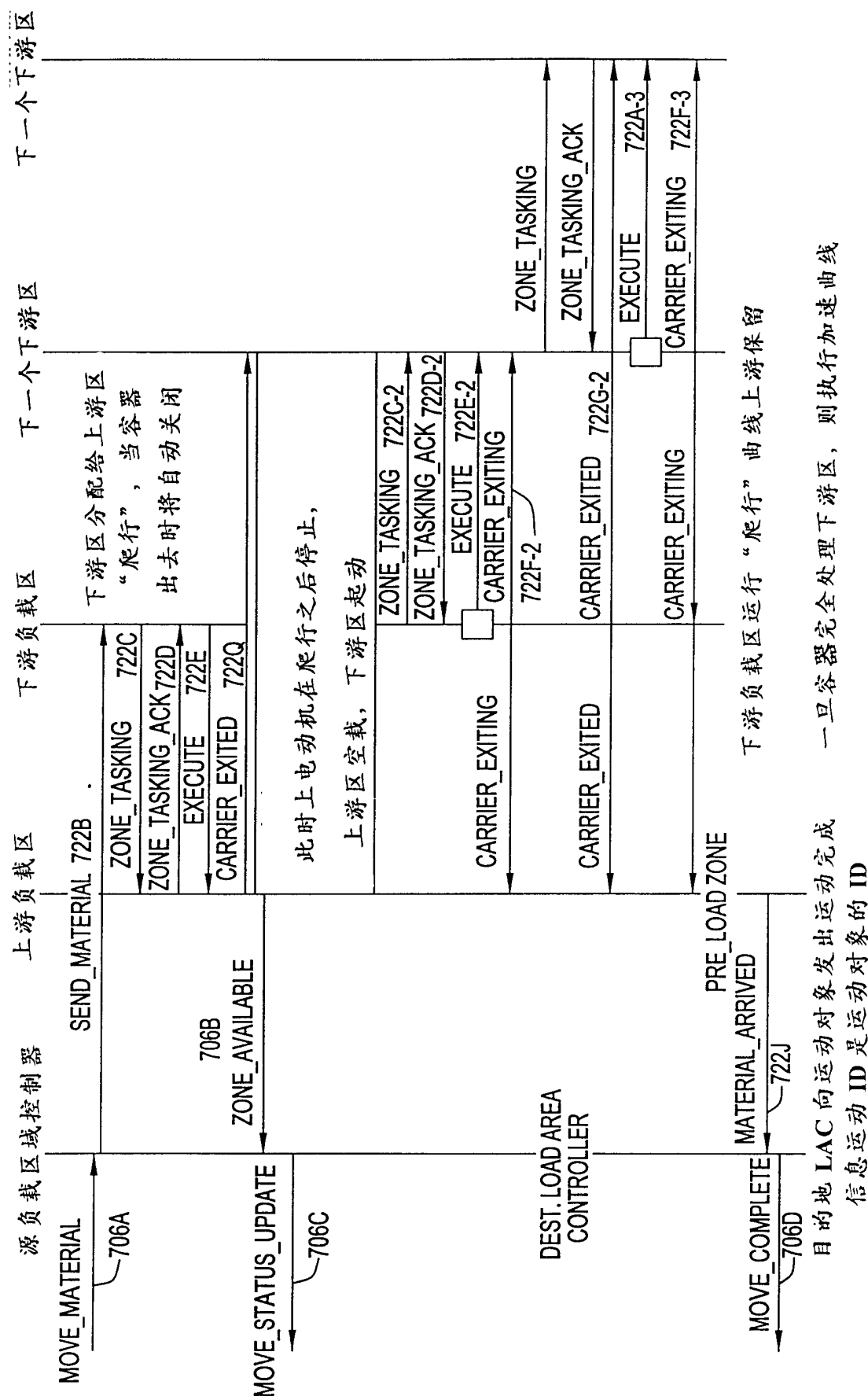
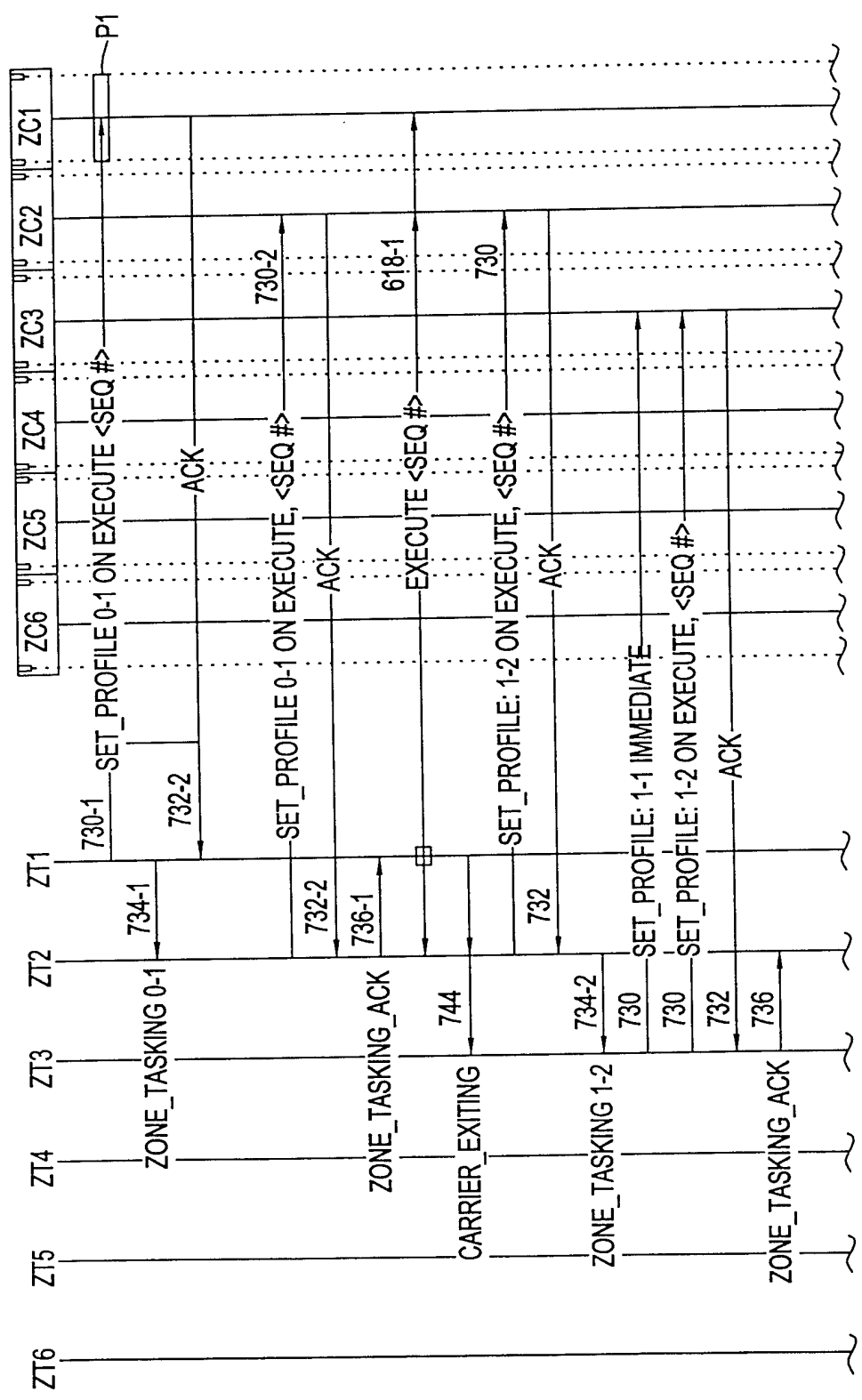
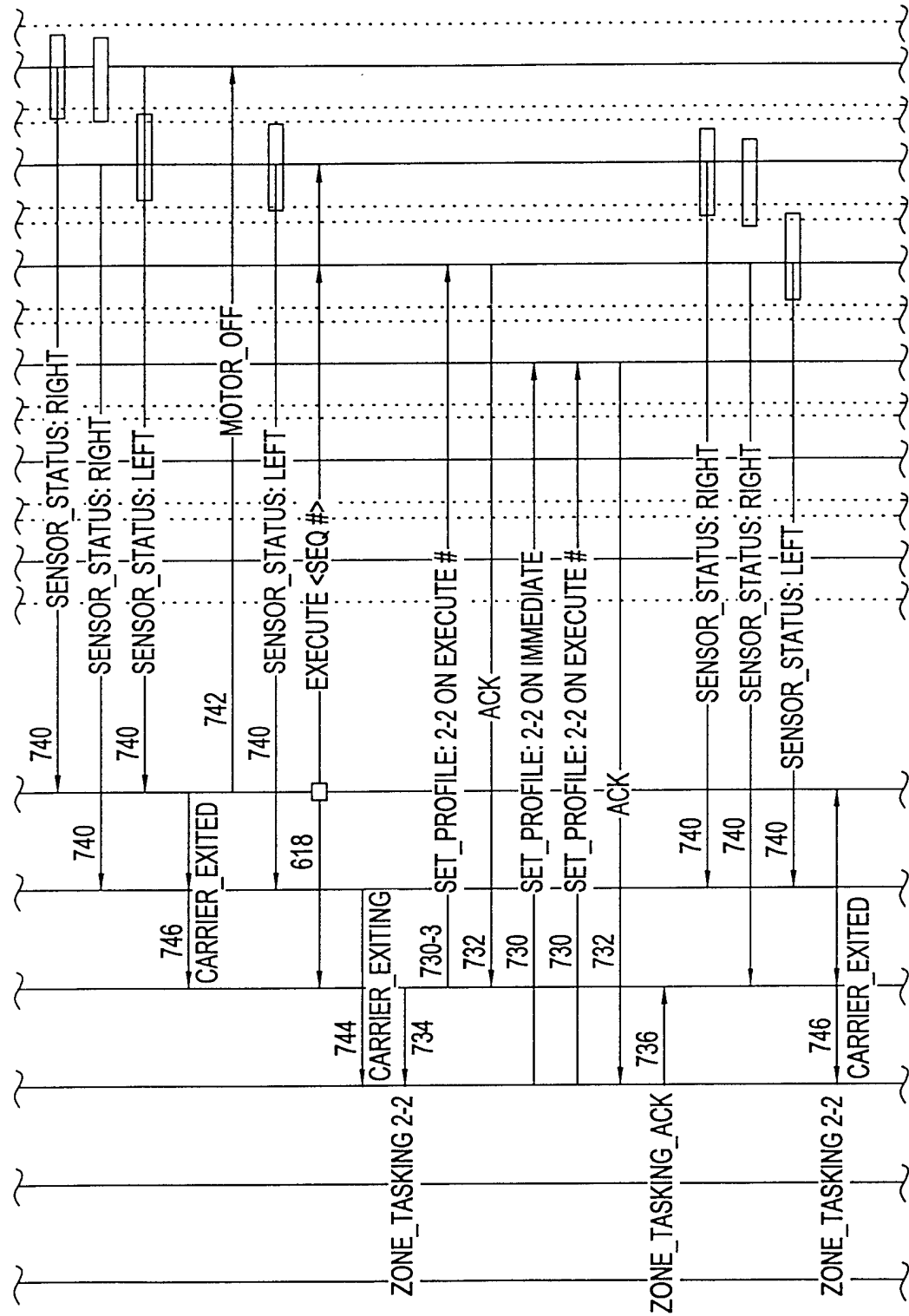


图 20



继续到图21B

图21A



连接到图 21C

图21B

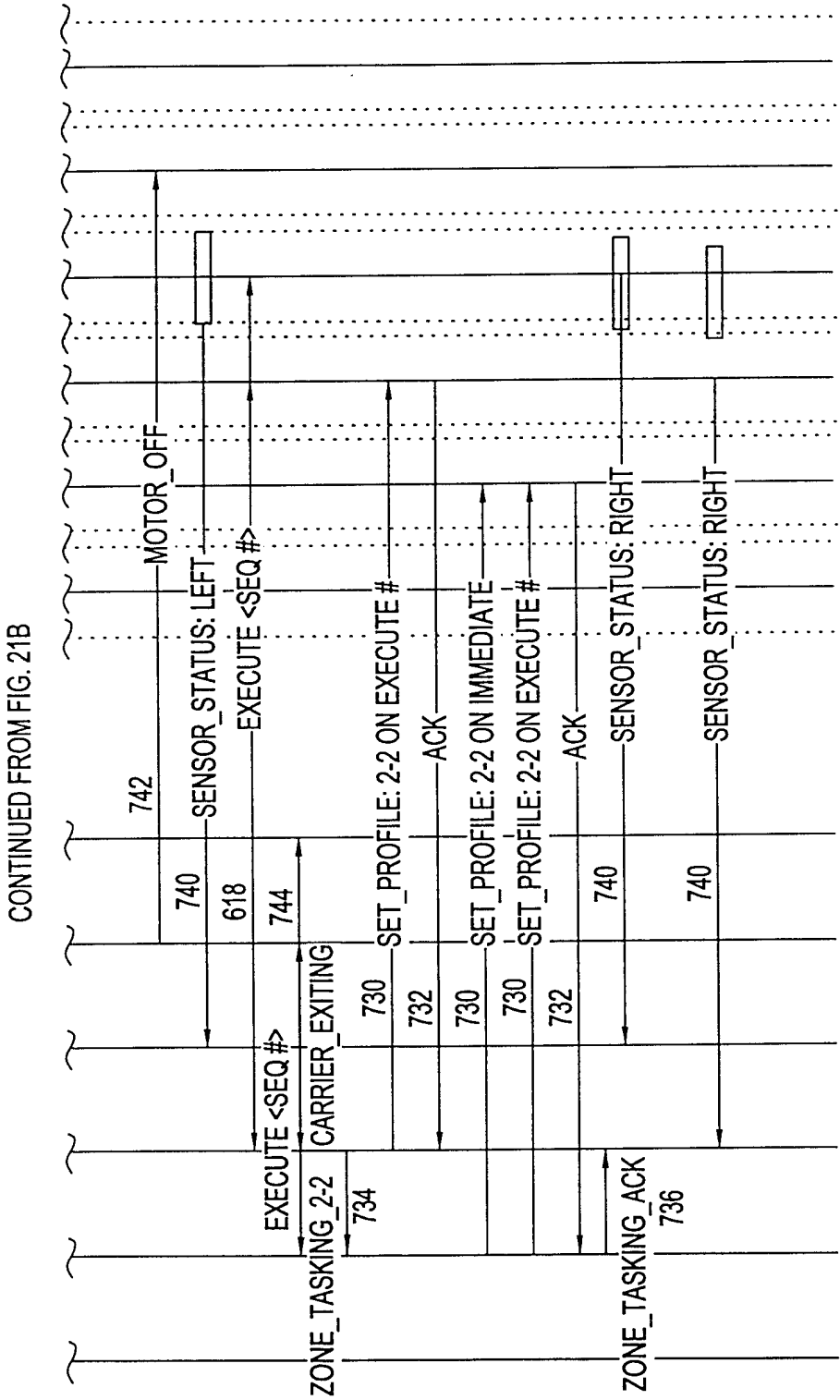
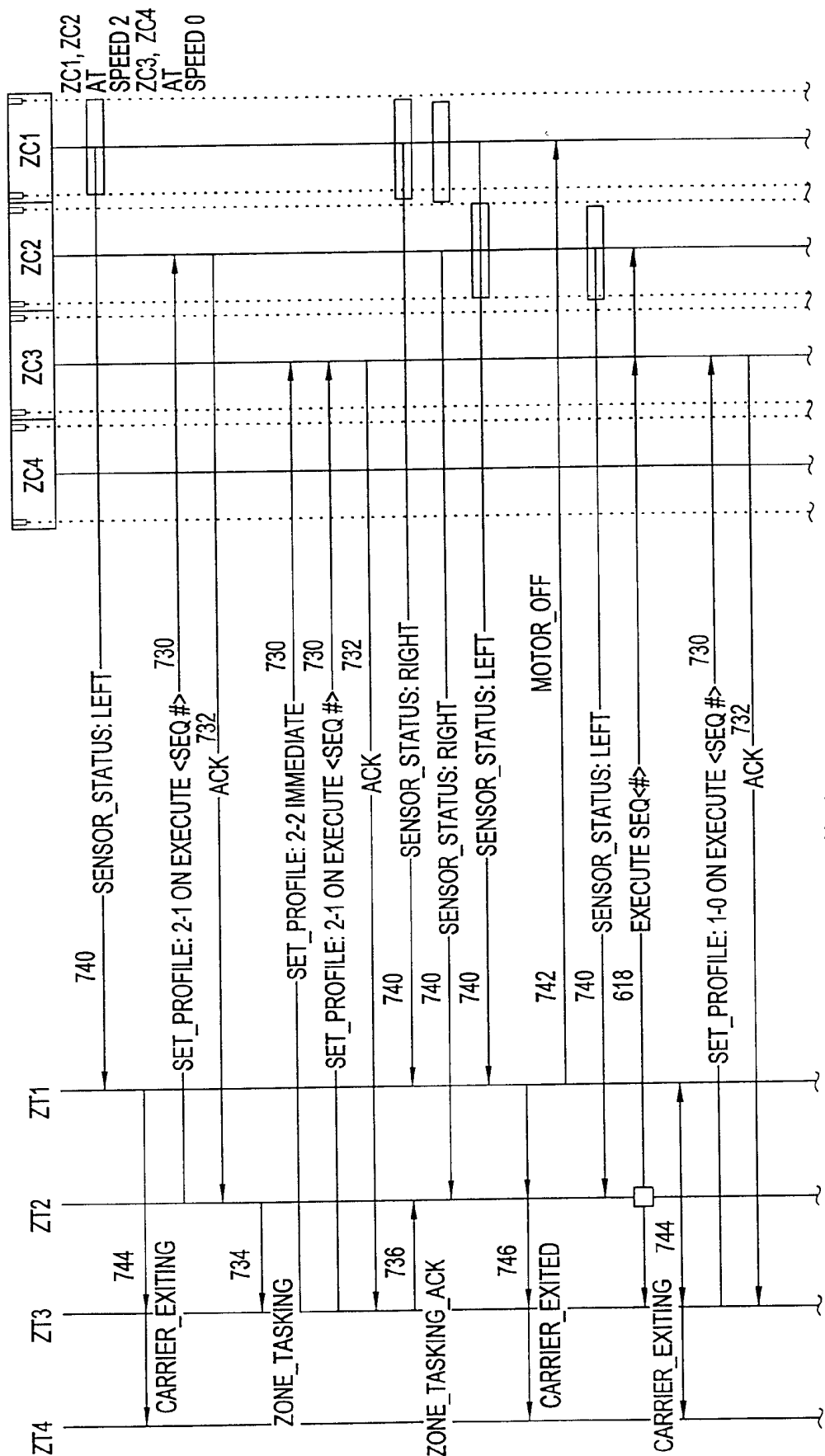
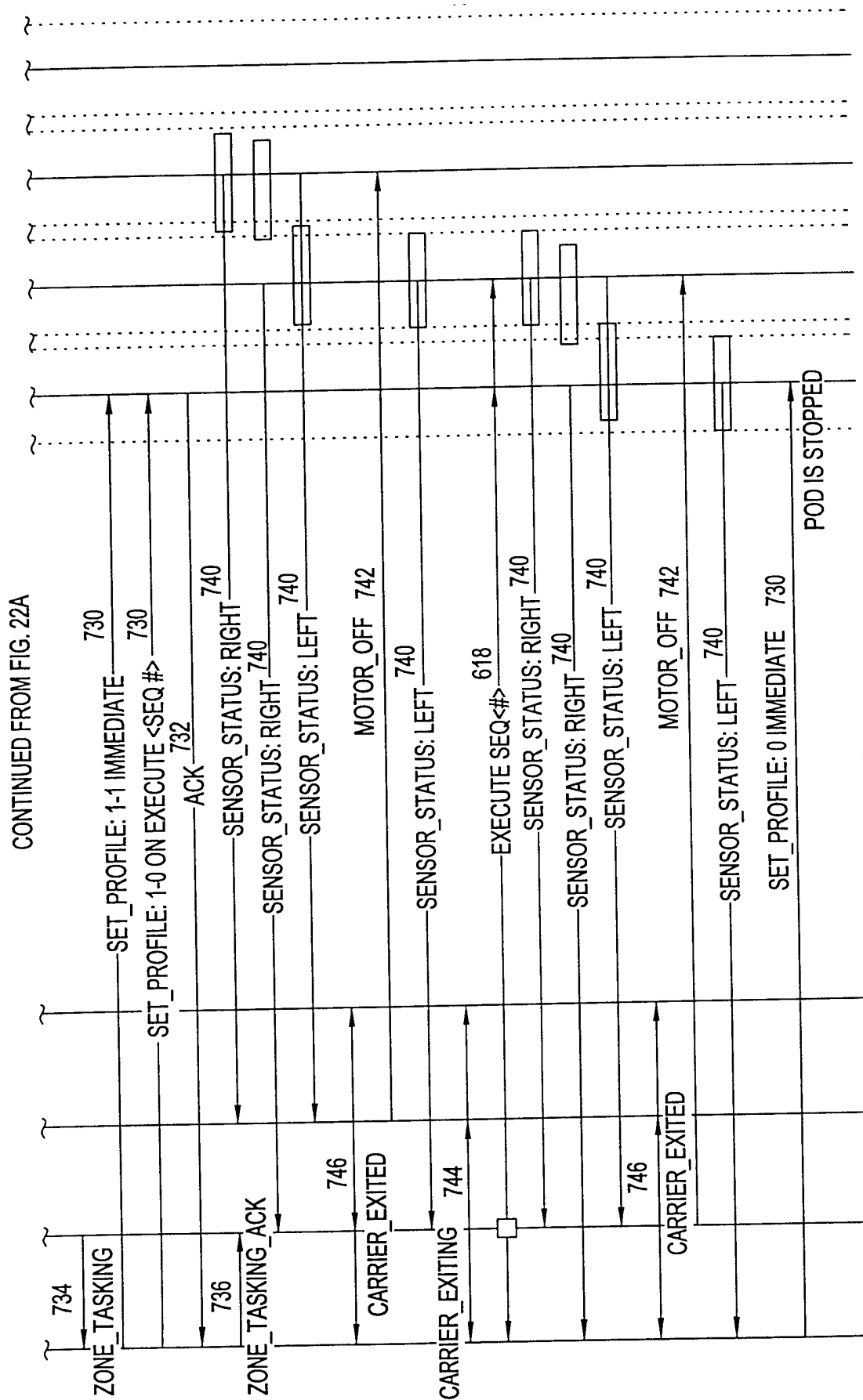


图21C

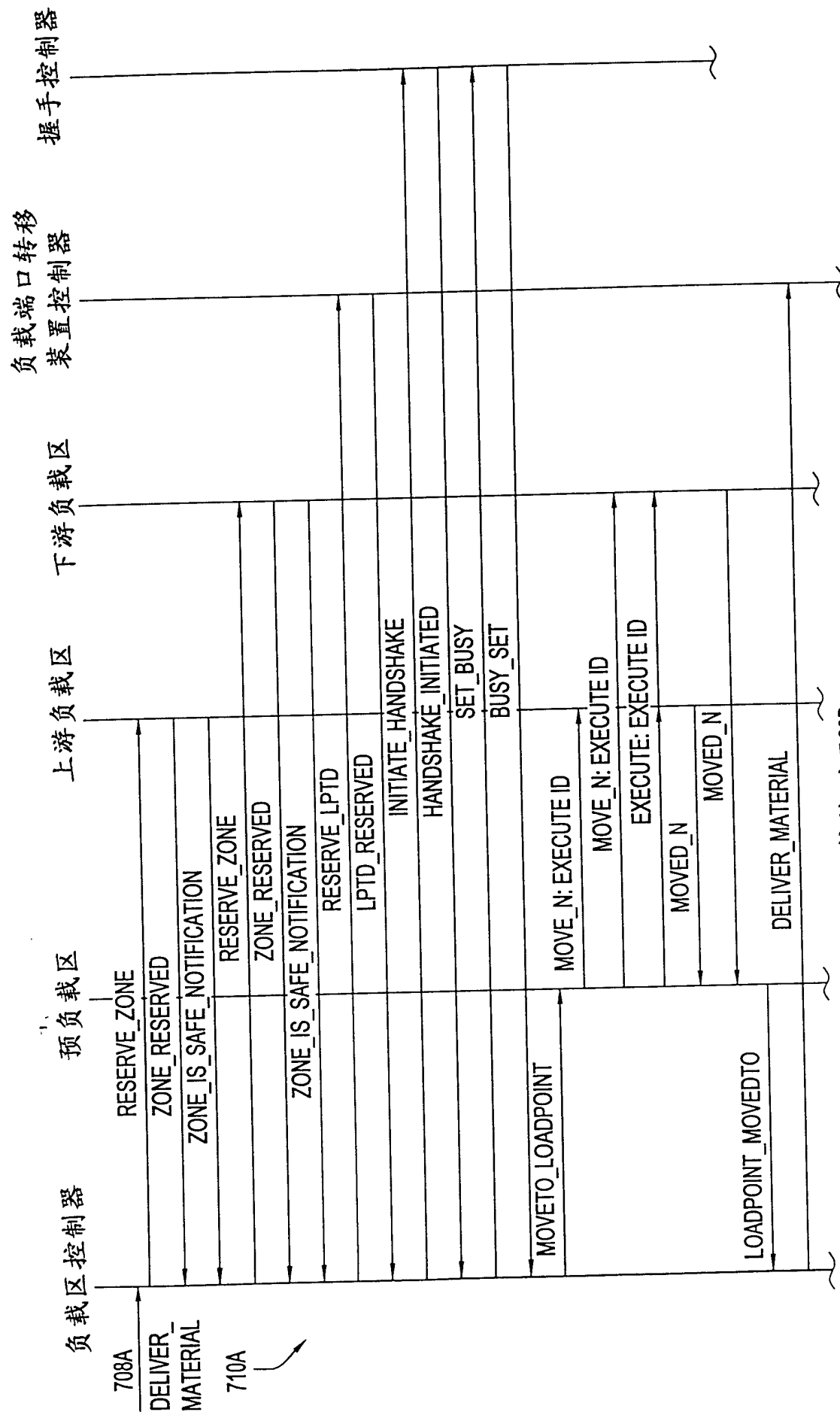


连接到图 22B

图 22A



22B



继续到图23B

图 23A

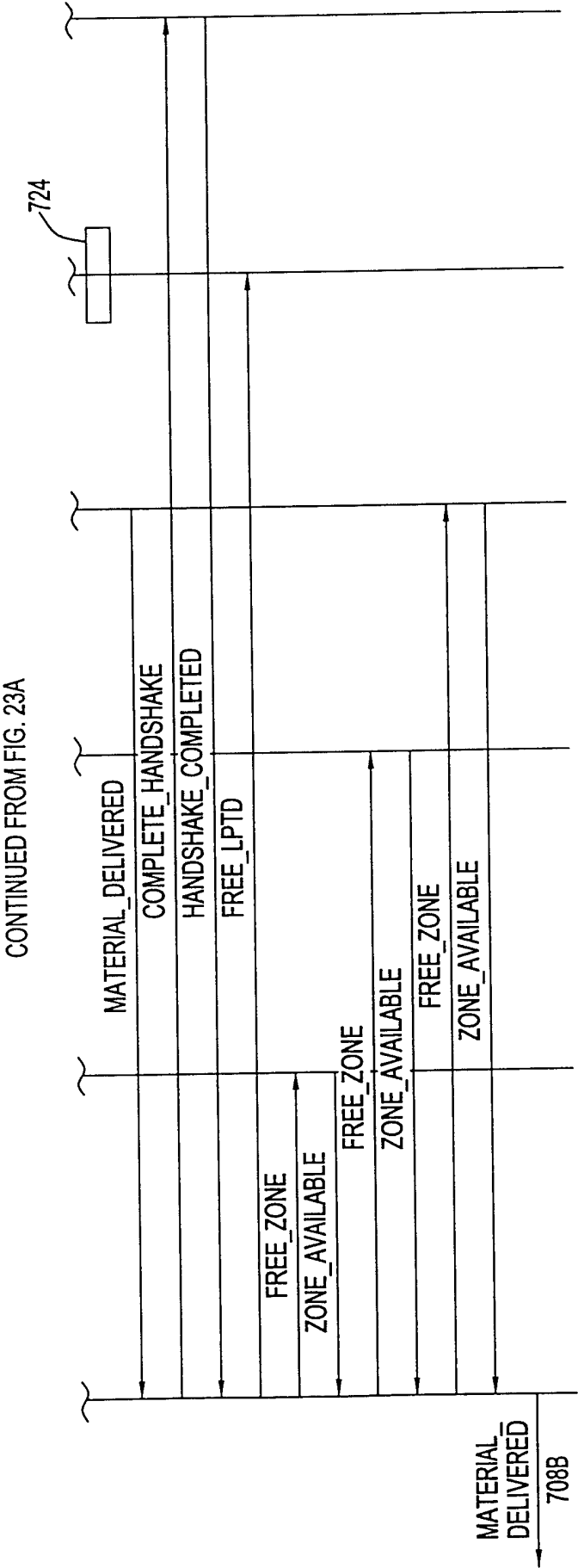


图 23B

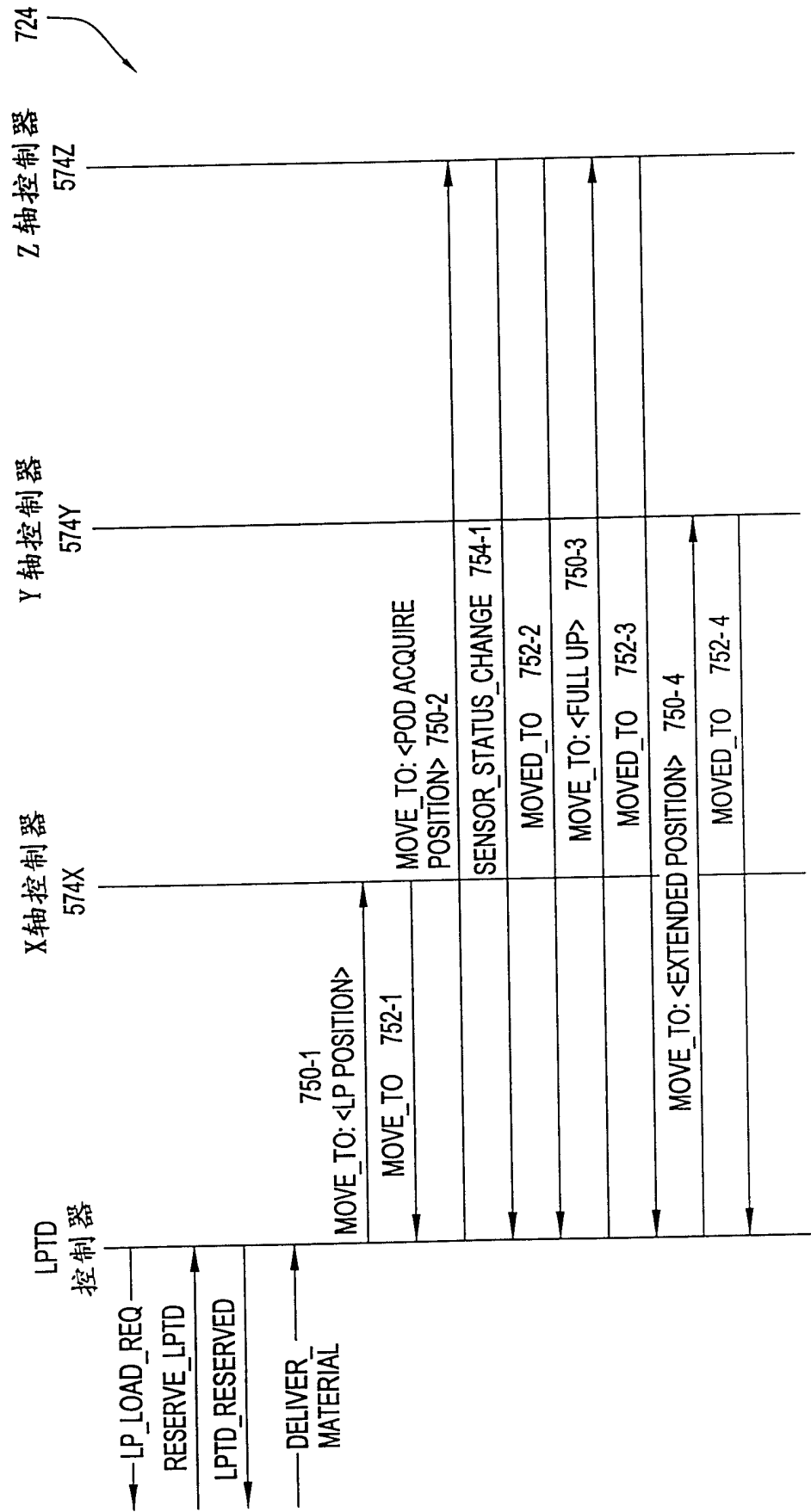


图24A

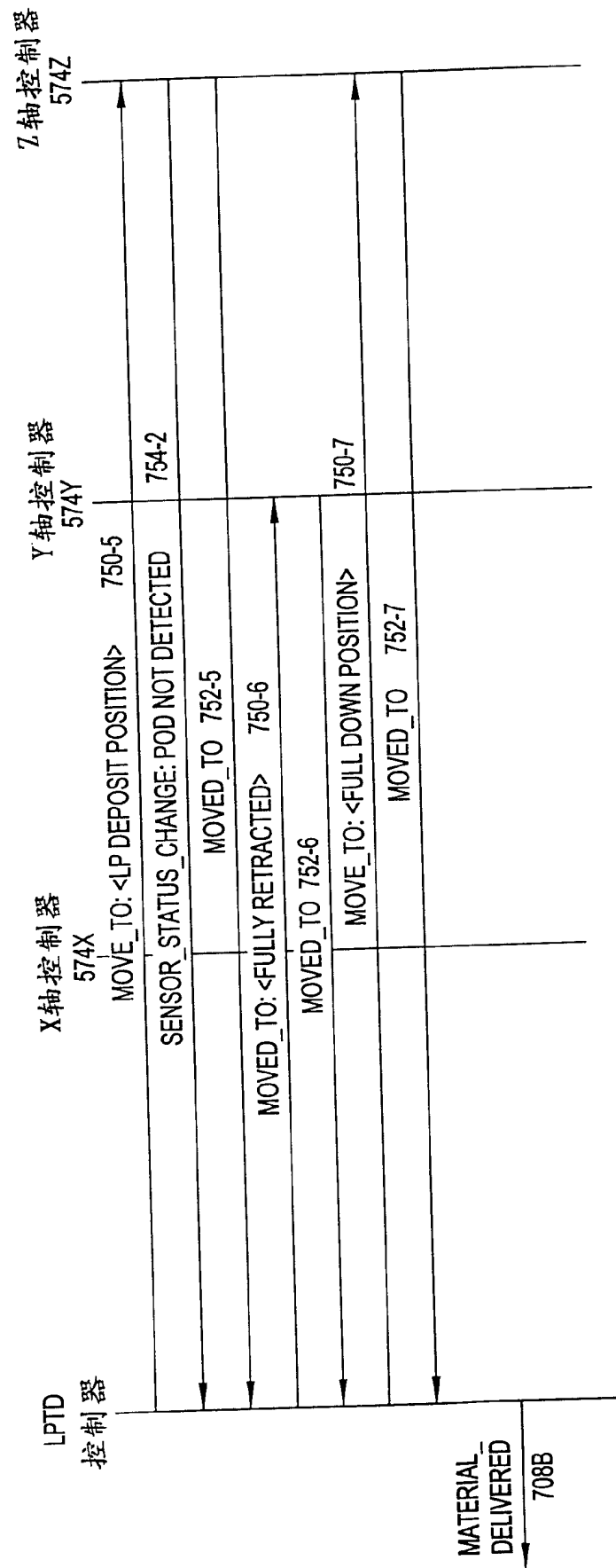


图24B

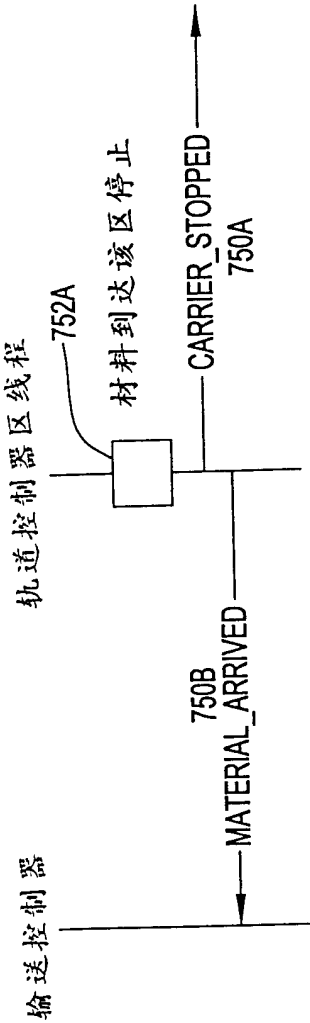
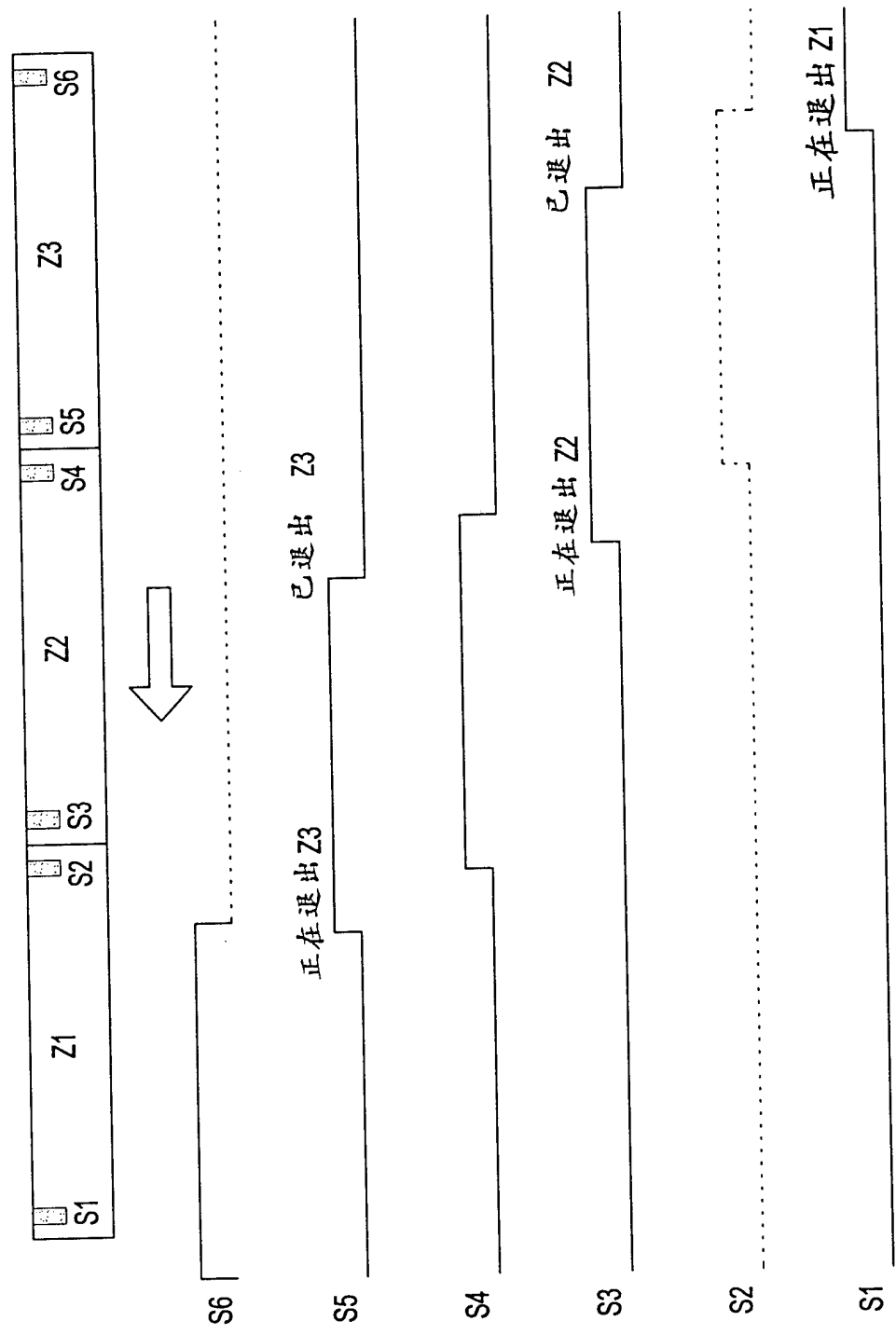


图 25



和区 2 相关的信息控制线程序
虚线表示探测器转换不被区 Z2 利用

图 26

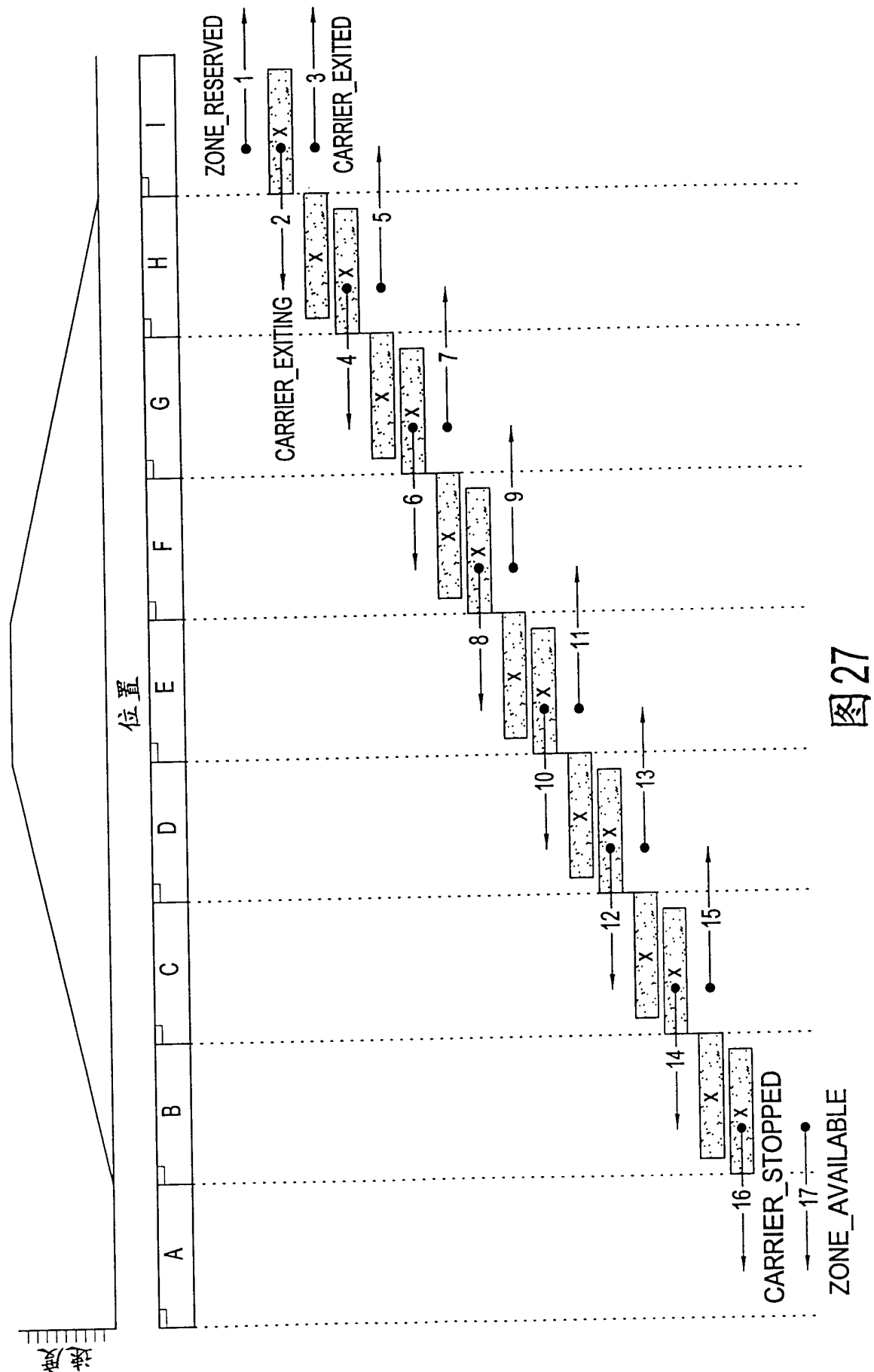


图27

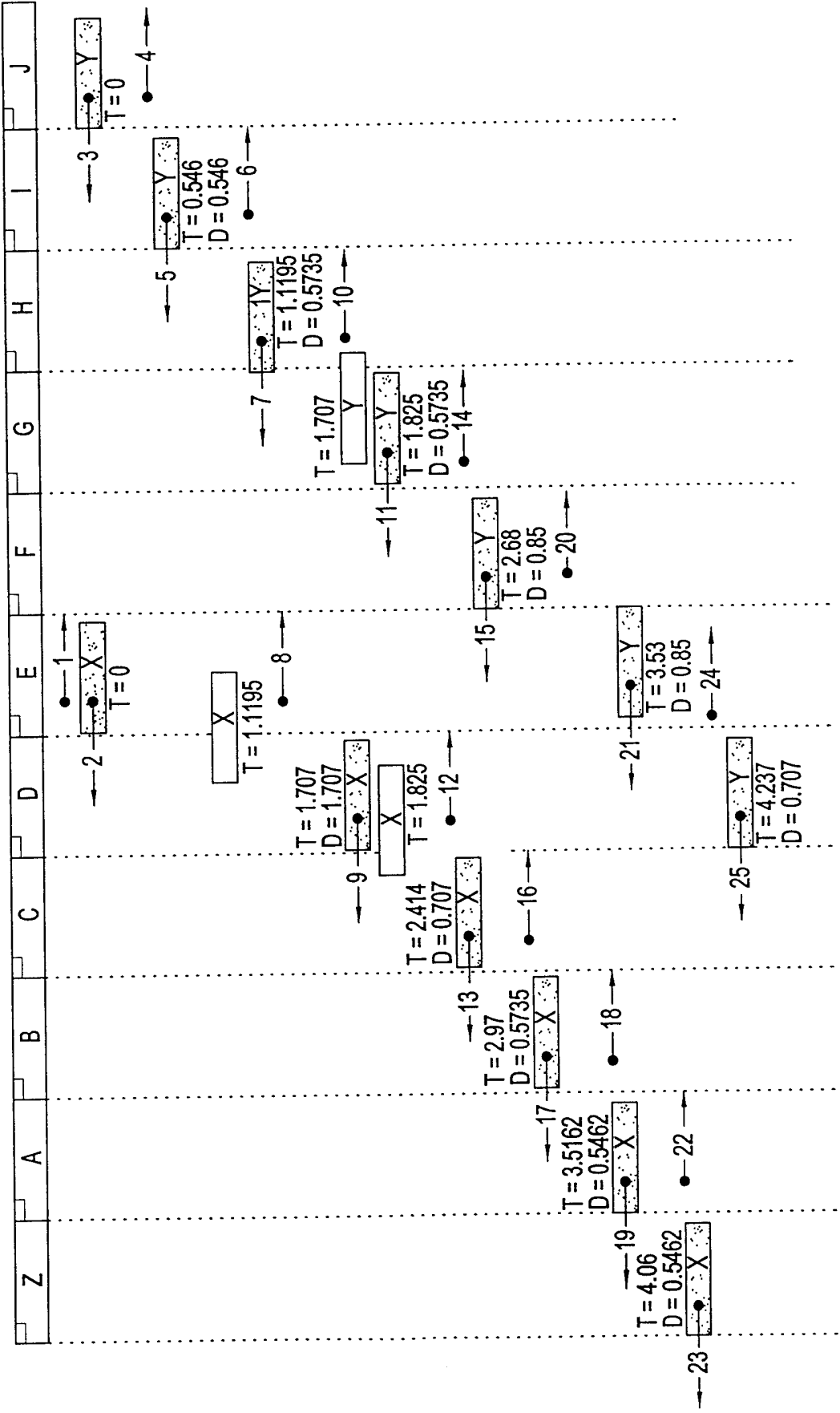


图28

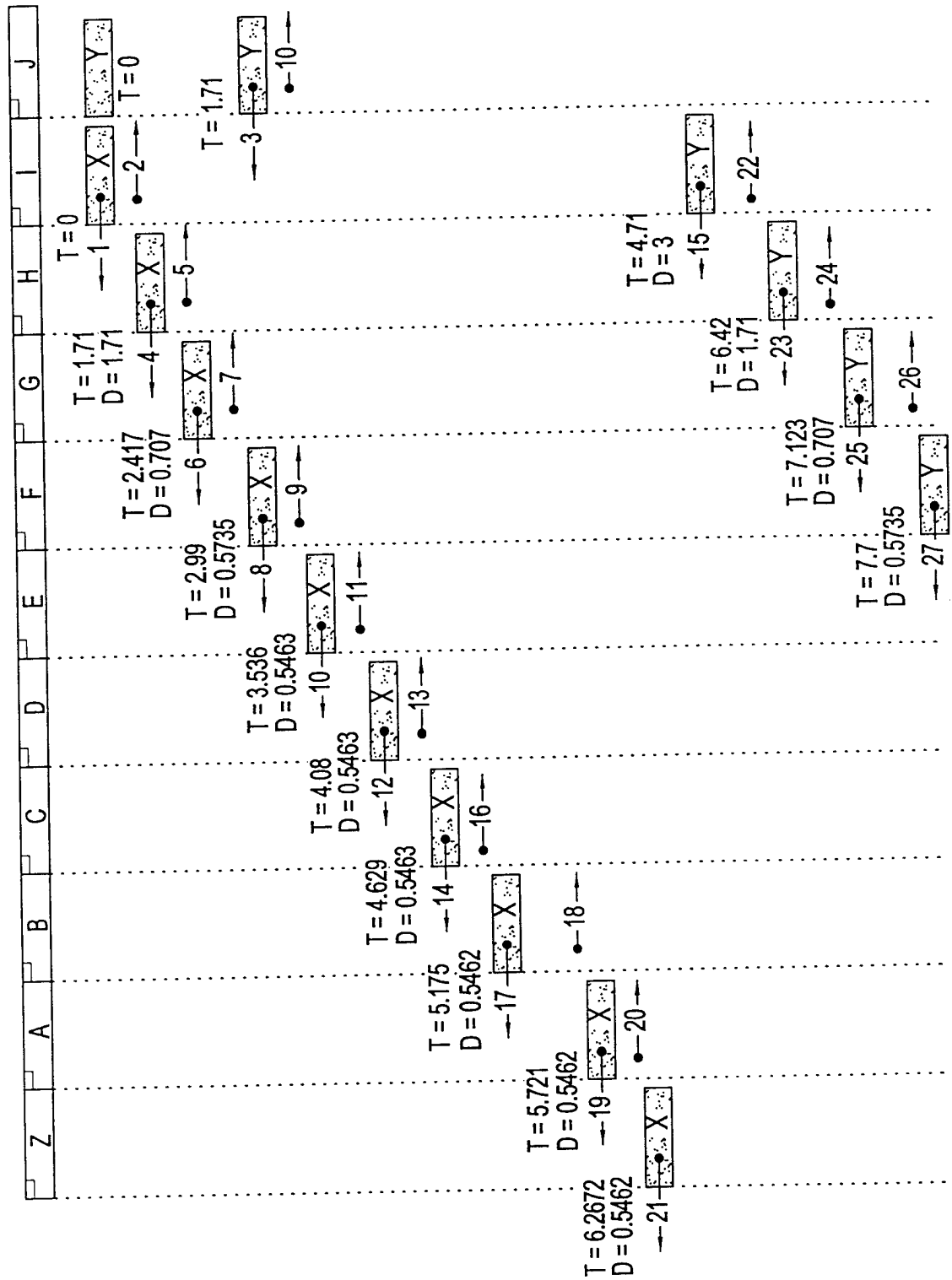


图29

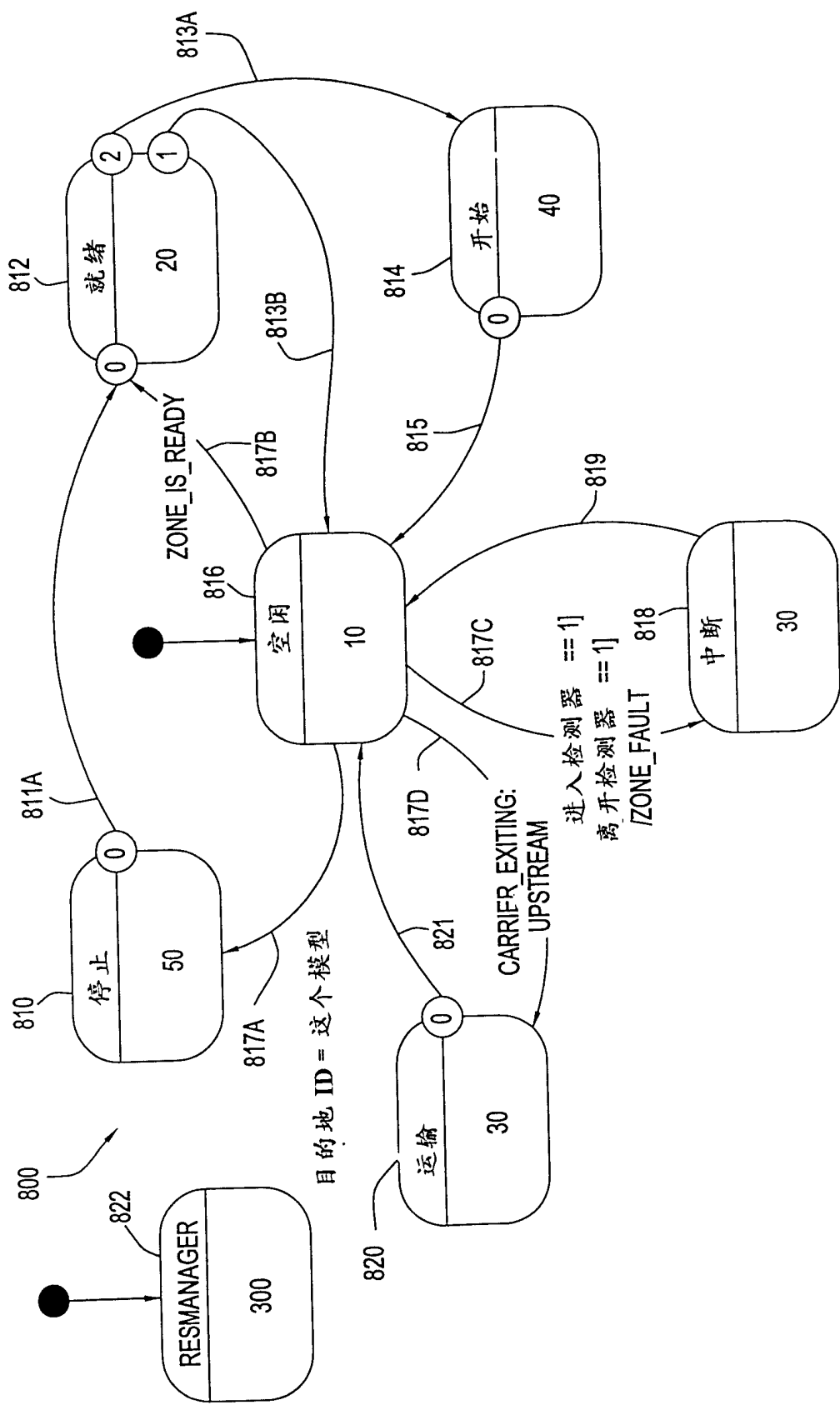


图30

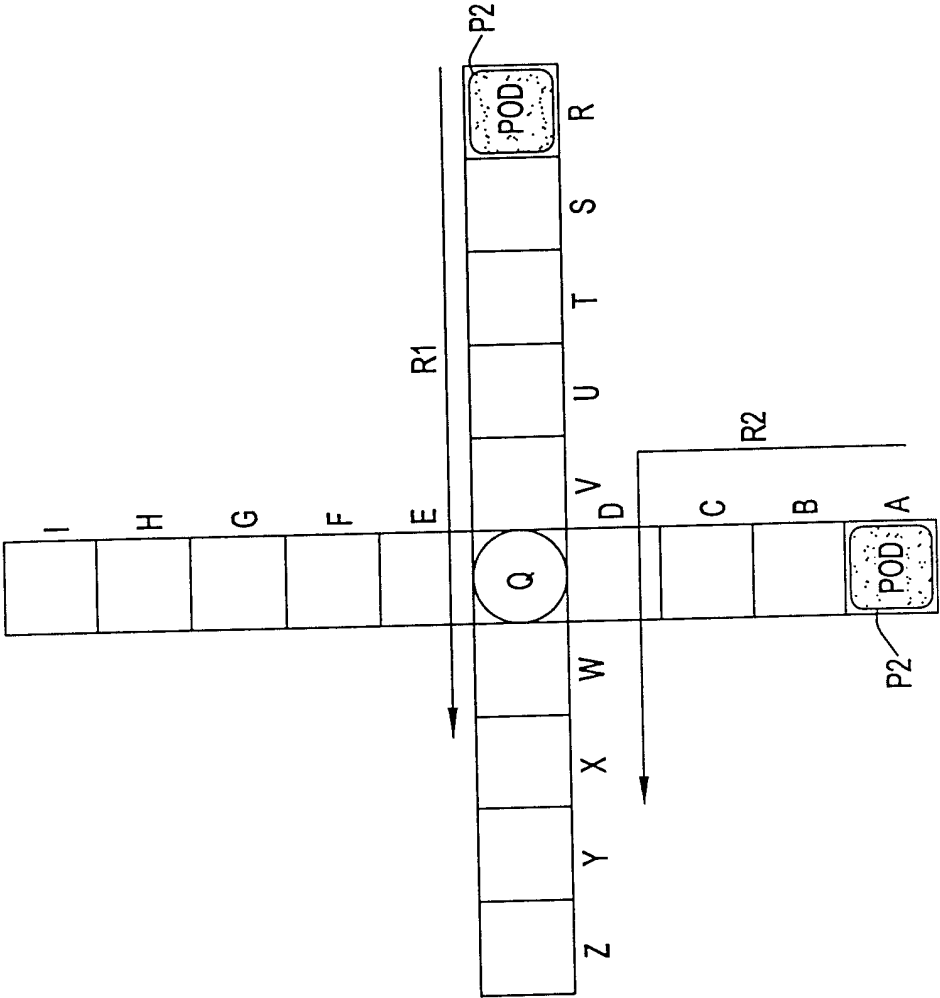
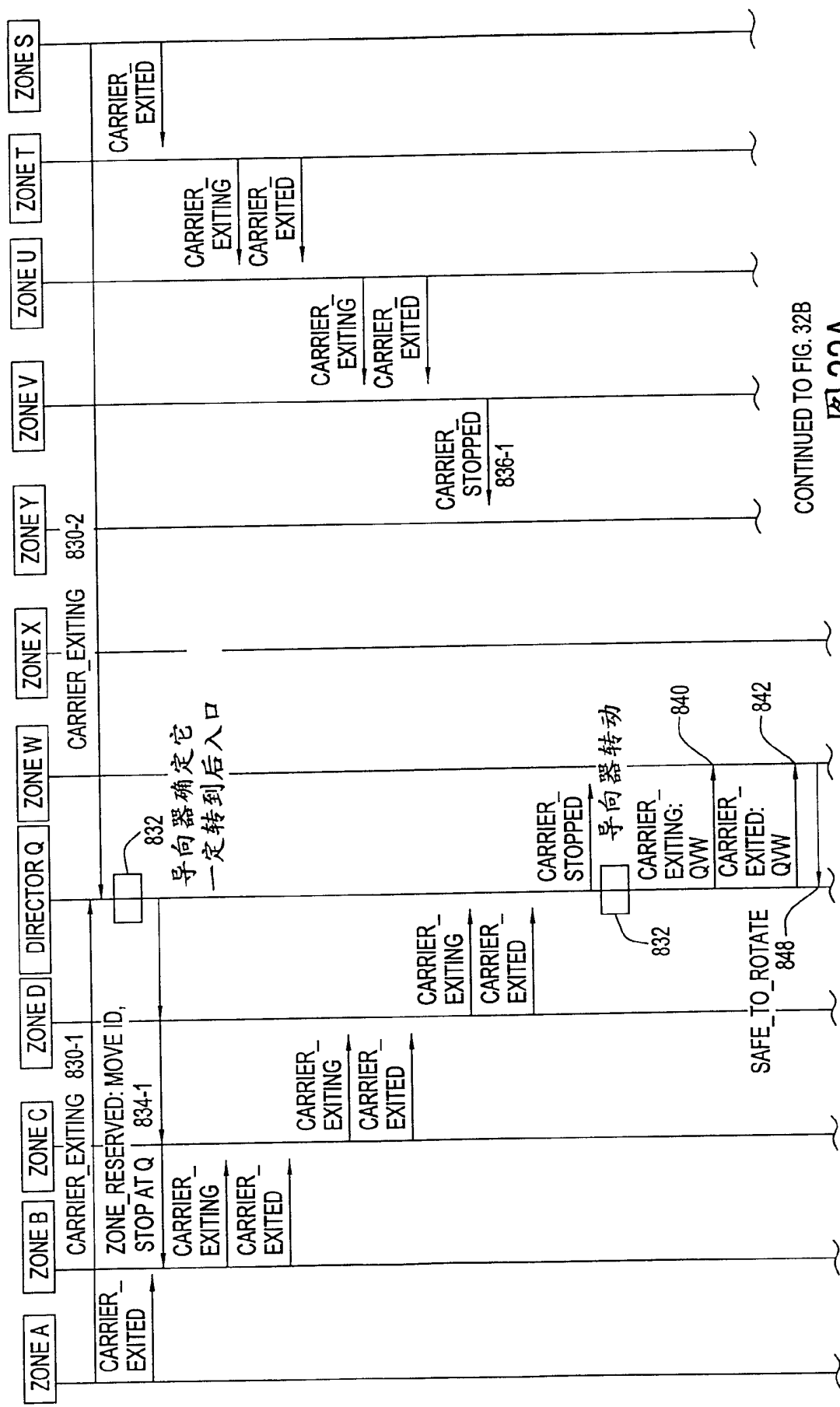
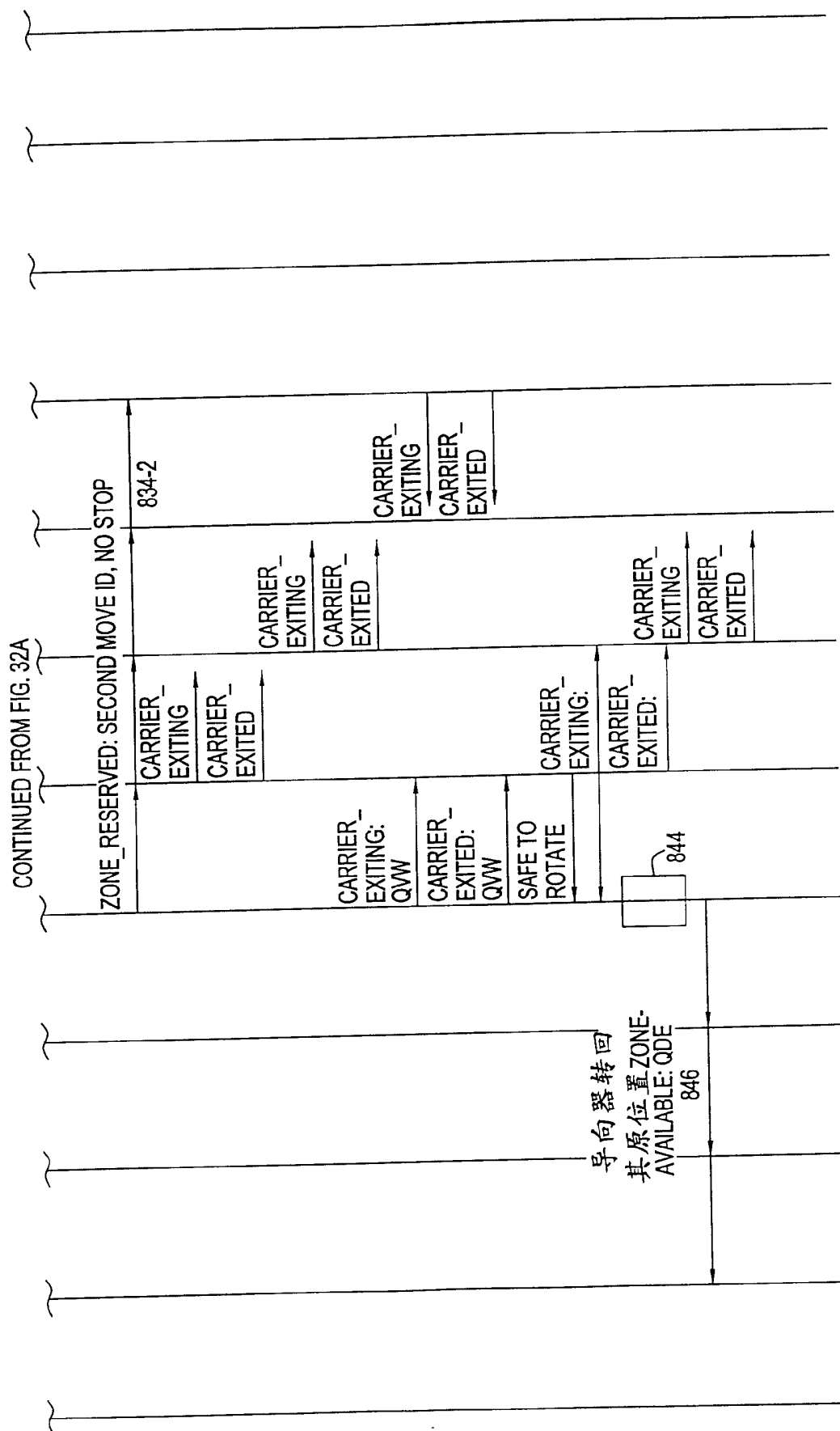


图 31



CONTINUED TO FIG. 32B

图 32A



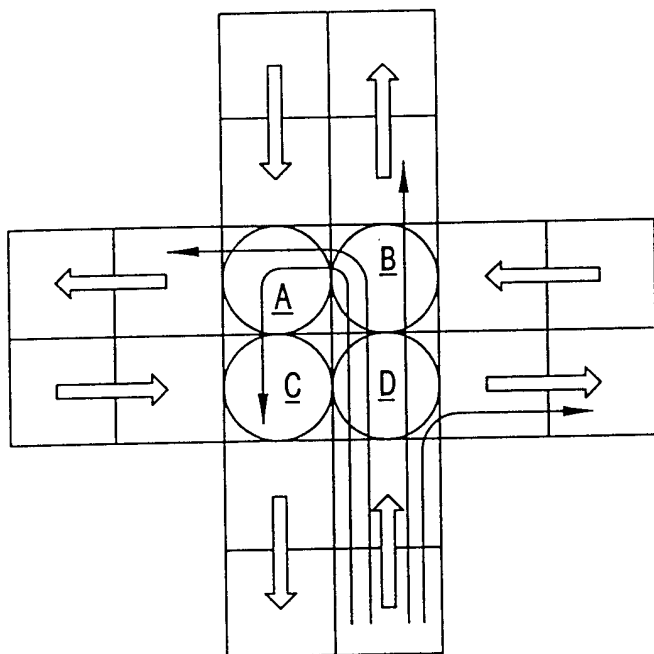


图33A

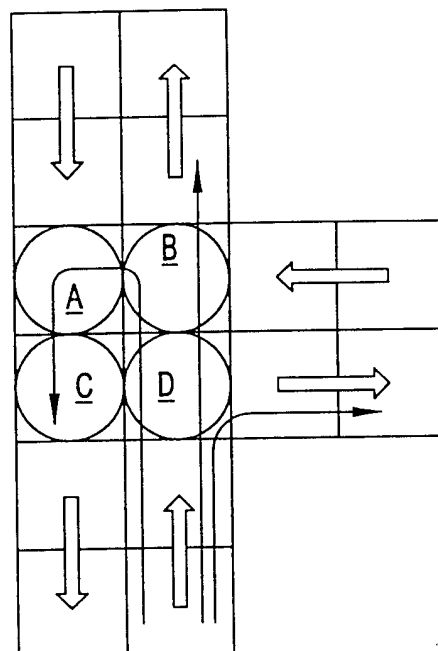


图33B

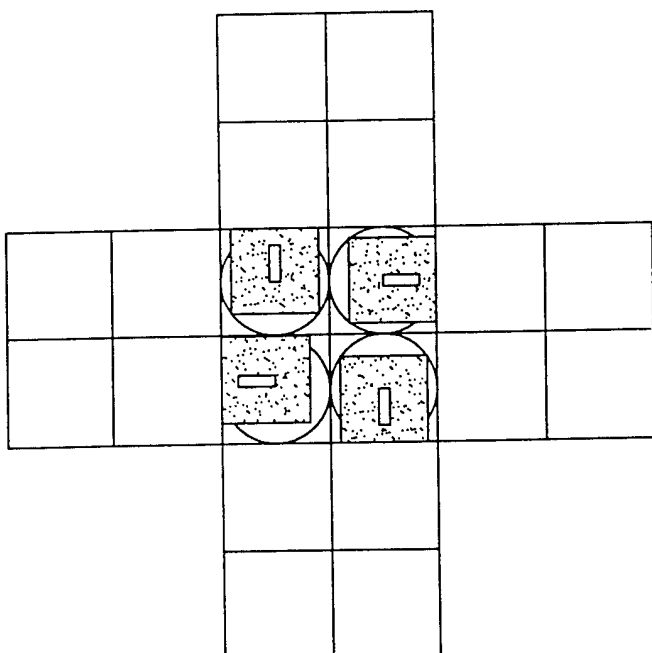


图33C

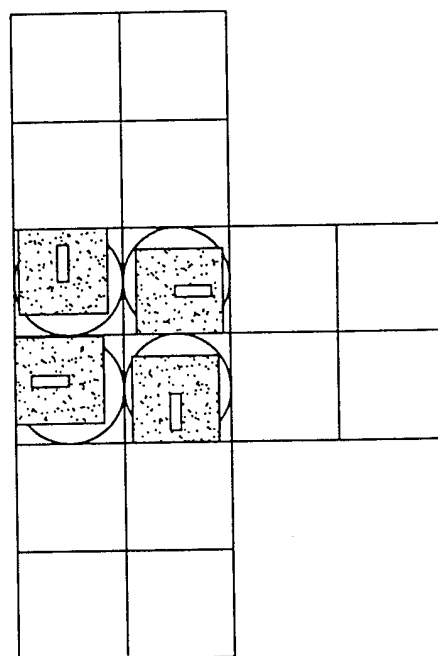


图33D

路径发现实例：
物理区布局

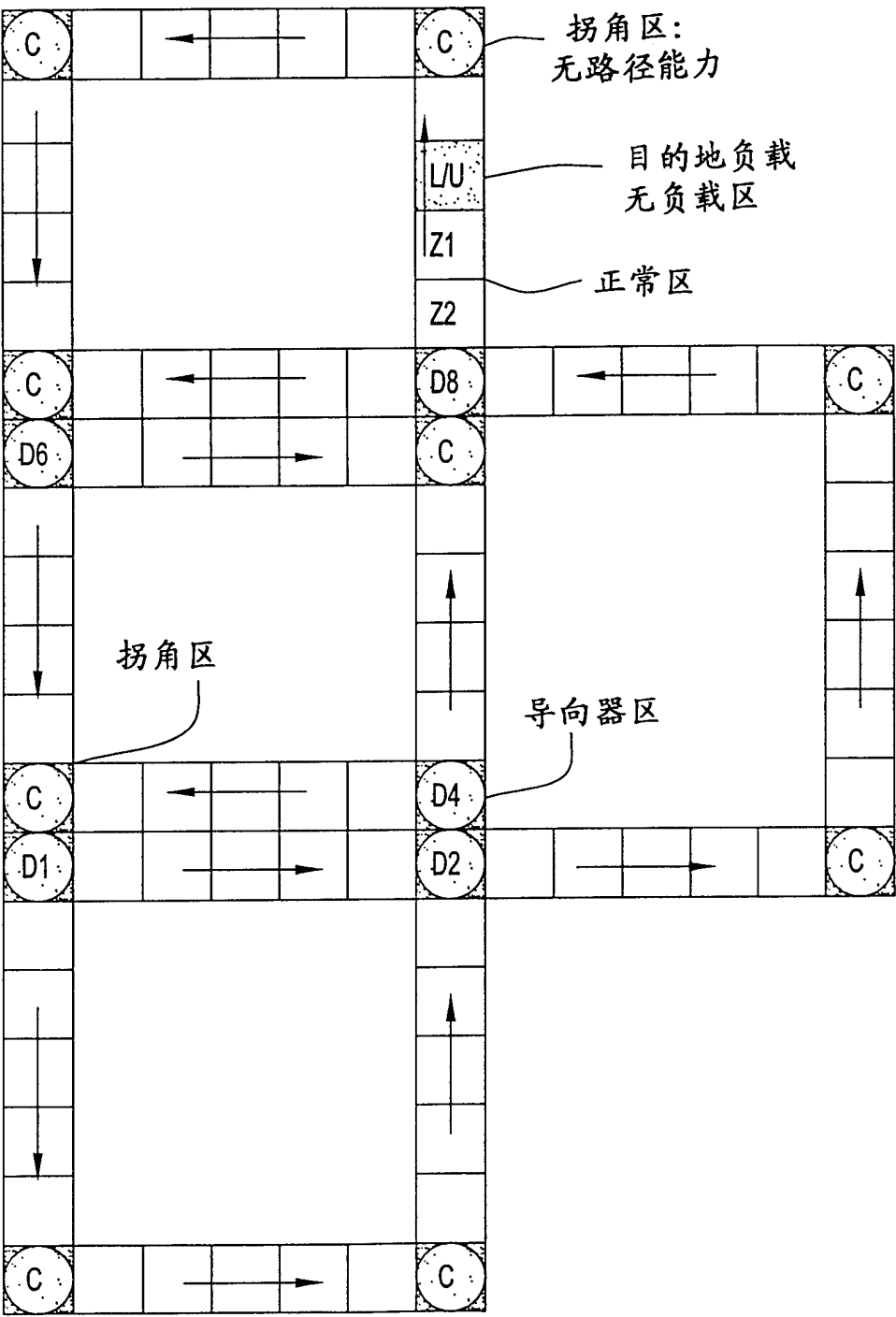


图 34

路径发现实例：上游区连接性

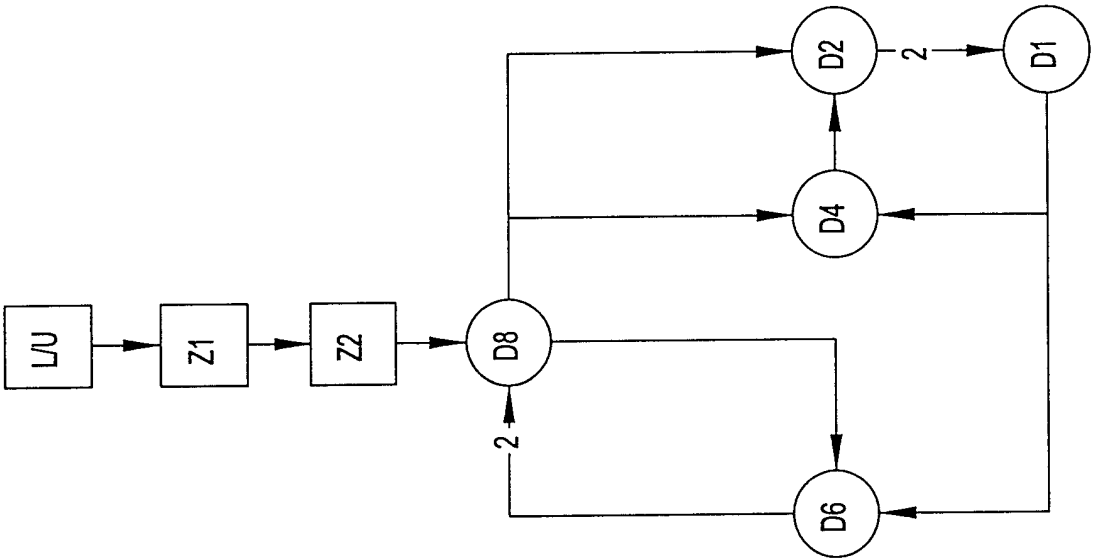


图35

路径发现实例：
物理区布局

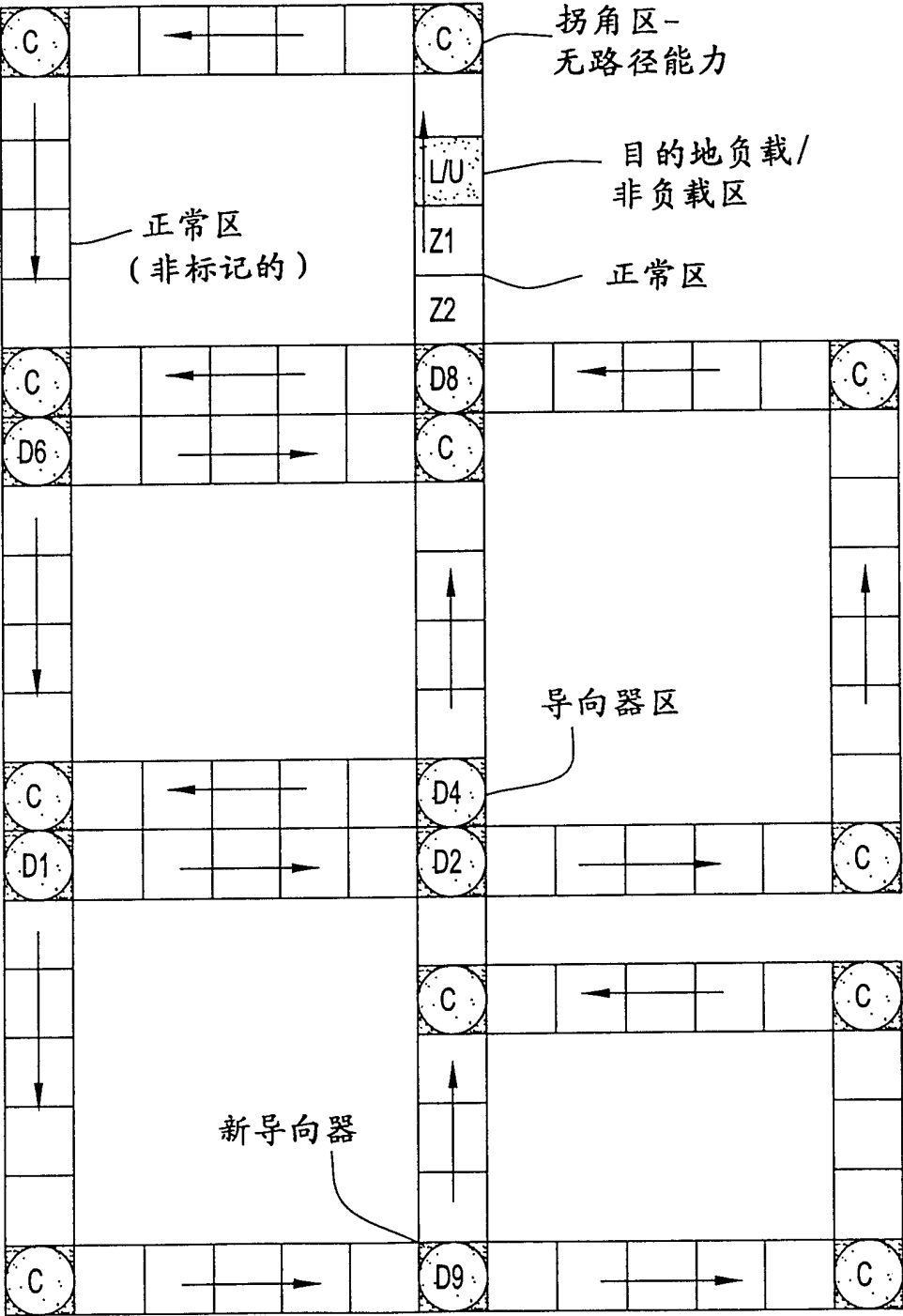


图 36

失败节点：
实例布局

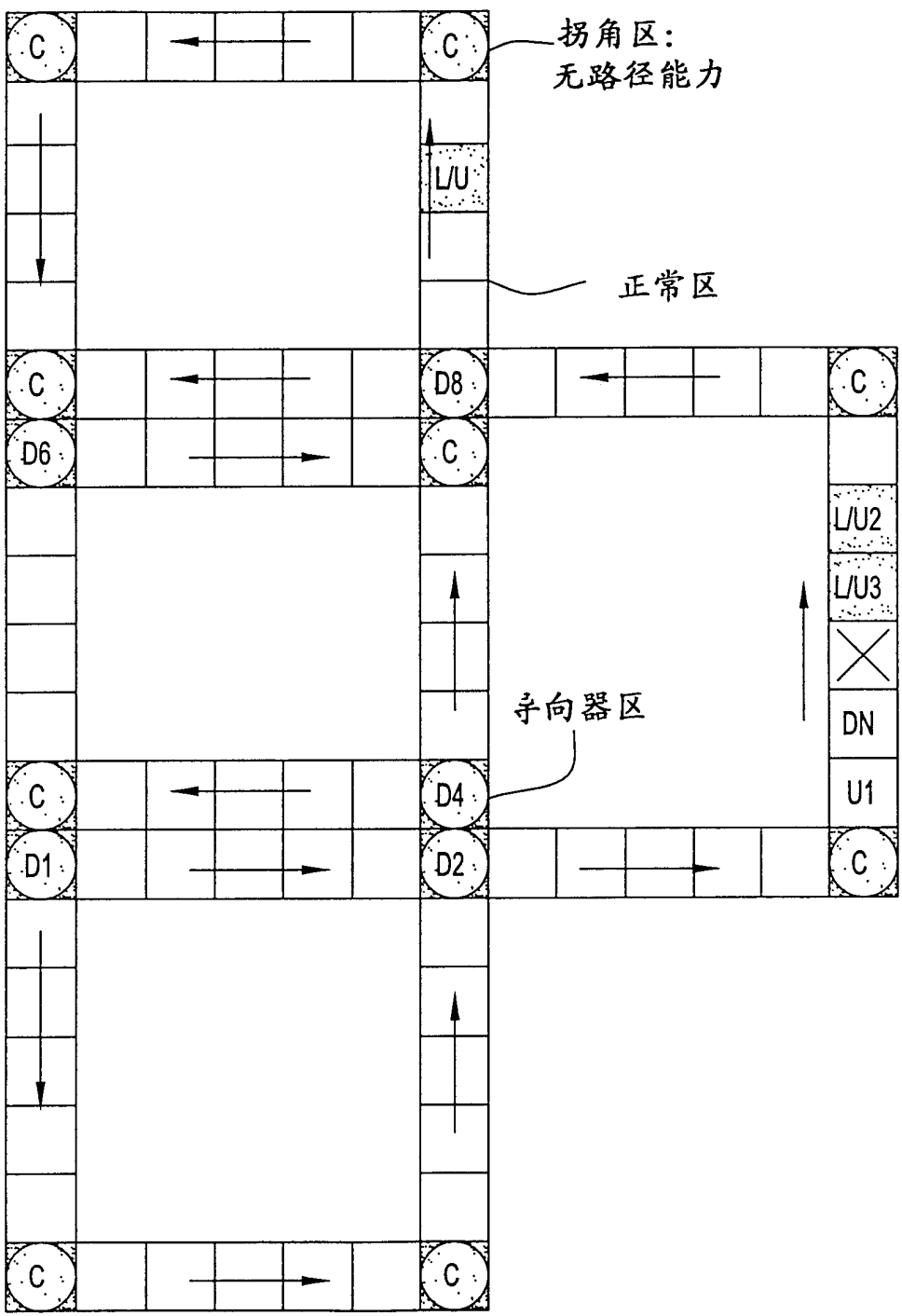


图 37

失败导向器：
实例布局

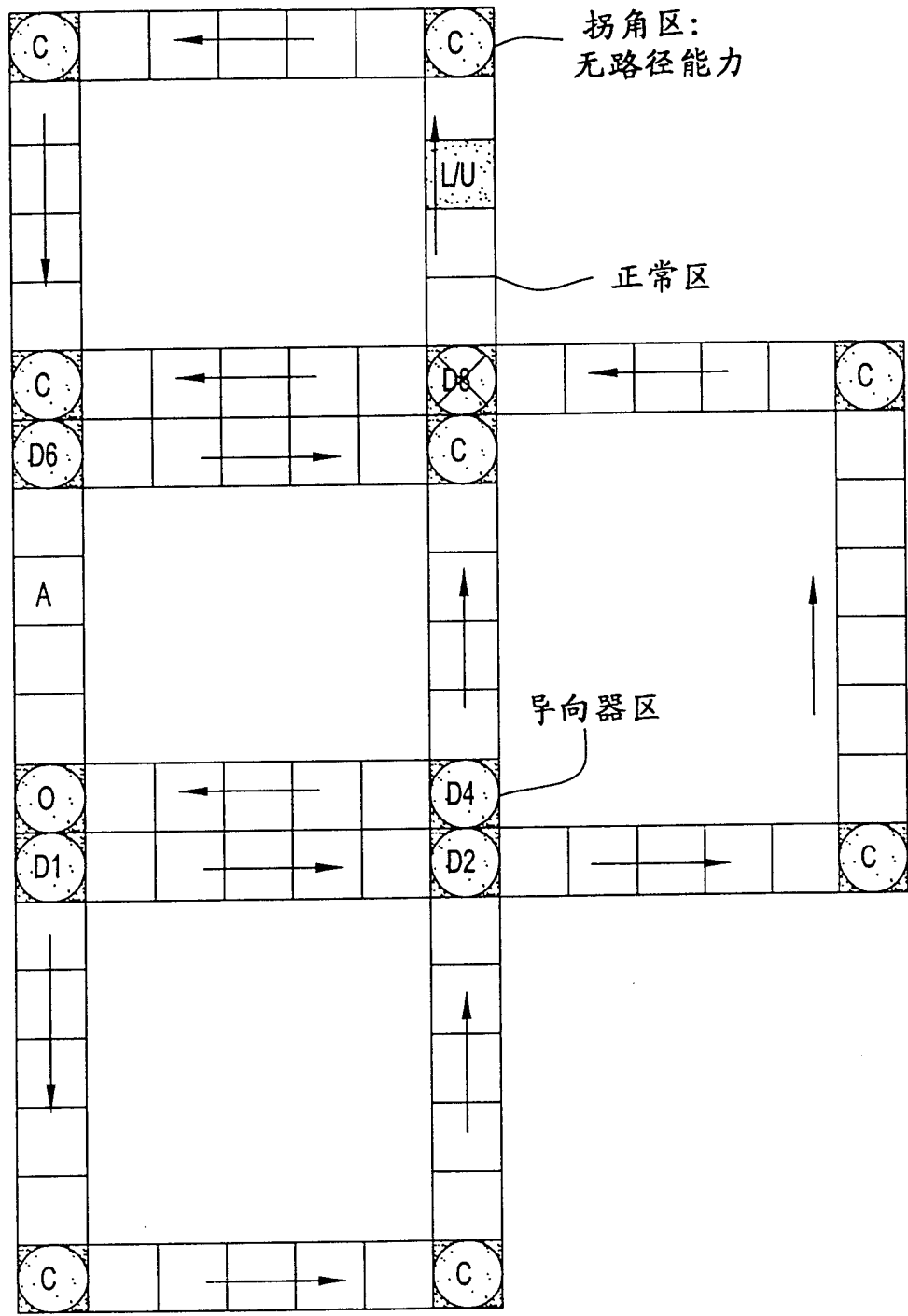


图 38