



# [12] 发明专利申请公开说明书

[21] 申请号 97114098.7

[43]公开日 1998年2月11日

[11] 公开号 CN 1172986A

[22]申请日 97.7.7

[30]优先权

[32]96.7.8 [33]JP[31]177898/96

[71]申请人 三菱电机株式会社

地址 日本东京都

[72]发明人 黑泽寿好

[74]专利代理机构 中国专利代理(香港)有限公司

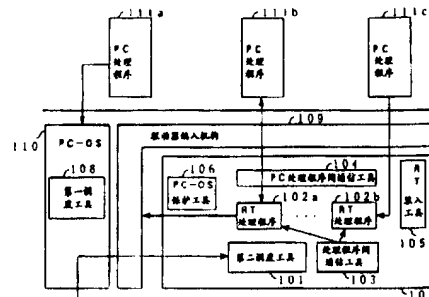
代理人 程天正 叶恺东

权利要求书 1 页 说明书 14 页 附图页数 16 页

[54]发明名称 实时控制系统

[57]摘要

在不保证实时性的个人计算机用的操作系统中(PC-OS)，通过在输入输出的设备驱动器的位置上提供实时处理，不改变PC-OS就可以实现实时应用程序和PC应用程序的共存。具有：PC-OS调出输入输出设备驱动器的机构；将获得的CPU使用权分配给各实时处理程序的第2调度工具；作为实时处理程序的安装工具和处理程序间通信机构。



## 权 利 要 求 书

1. 一种实时控制系统，其特征在于，在具有对应用处理程序进行 CPU 资源的分配和调度的第 1 调度工具以及用于安装对设备进行输入输出处理的设备驱动器的驱动器安装工具的操作系统中，

5 所述操作系统将要求实时响应设备来的中断的处理作为设备驱动器进行存取，构成实时控制机构，

该实时控制机构包括：与对设备进行处理对应构成的实时处理程度和

对实时处理程序进行所述 CPU 资源的分配和调度的第 2 调度工具。

10 2. 权利要求 1 记载的实时控制方式，其特征在于，所述实时控制机构具有在所述实时处理程序之间进行数据收发的实时处理程序间通信工具。

3. 权利要求 1 记载的实时控制方式，其特征在于，所述实时控制机构具有在所述操作系统上工作的应用处理程序和所述实时处理程序之间进行数据收发的处理程序间通信工具。

4. 权利要求 1 记载的实时控制方式，其特征在于，所述实时控制机构具有把在所述操作系统上工作的应用处理程序作为所述实时处理程序登录的实时处理程序装入机构。

5. 权利要求 1 记载的实时控制方式，其特征在于，所述实时控制机构设有操作系统保护工具，使所述实时处理程序可以利用所述操作系统对设备驱动器提供服务。

6. 权利要求 1 记载的实时控制方式，其特征在于，所述实时控制机构具有使用权监视工具，所述第 2 调度工具测量并保持操作系统占有 CPU 资源的时间，根据该测量结果强制性地使 CPU 资源返回操作系统。

25 7. 权利要求 1 记载的实时控制方式，其特征在于，当存在多个所述实时控制机构时，具有调整该多个实时控制机构之间的主从关系或优先顺序关系的竞争控制工具。

# 说明书

## 实时控制系统

5 本发明涉及一种 PC 实时控制机构，该 PC 实时控制机构的目的在于使要求实时性的处理程序和通用个人计算机上工作的处理程序共存。

近年来，伴随个人计算机（下称 PC）的迅速普及，即使在过去所谓封闭的实时系统中，也要求使用 PC 上流行的软件资源。

10 为了使用 PC 上流行的软件资源，必须使在实时系统上工作的实时（RT）处理程序和 PC 上工作的 PC 处理程序协调动作，已往，采取下面 2 种方法。

先有例 1

作为第 1 种方法，例如有在“测量和控制第 34 卷第 3 号”（1995 年 3 月发行）201 页上记载的方法。

15 该方法如图 15 所示，分别安装运行 PC - OS（110）和 CPU1（1301）和运行实时 OS（1304）的 CPU2（1302），RT 处理程序（102）在实时 OS（1302）的控制下工作，PC 处理程序（111）在 PC - OS（110）的控制下工作。

RT 处理程序和 PC 处理程序通过可以由 CPU1 和 CPU2 访问的公共存储器（1303）进行数据交换。

20 先有例 2

作为第 2 种方法，例如有在“接口技术 1996 年 6 月号”（CQ 出版）142 页上记载的方法。

该方法如图 16 所示，以实时 OS（1304）为核心，进而以仿真器的形式安装上 PC - OS（1401）。

25 RT 处理程序（102）直接利用实时 OS（1304）提供的服务，PC 处理程序（111）在 PC - OS 仿真器（1401）上工作。

因为先有的 PC 实时控制机构如上述那样构成，所以必须安装额外的硬件（如实时 OS 和 RT 处理程序工作时所用的 CPU 的公共存储器等）因而存在成本高和问题。

30 此外，当作为 PC - OS 仿真器构成时，虽然额外的硬件所带来的成本的增加能够得以抑制，但是由于需要跟踪 PC - OS 版本升级而导致的版本升级问题，所以有必要整个改变 PC - OS 仿真器，从而存在软件开



发费增加的问题。

本发明是鉴于上述课题而提出的，其目的在于，提供一种 PC 实时控制机构，在实现 RT 处理程序和 PC 处理程序共存的系统中，既可以降低硬件成本，同时对 PC - OS 的版本升级也能应付自如。

5 与本发明第一方面有关的实时控制系统，在具有对应用处理程序进行 CPU 资源的分配和调度的第一调度工具以及用于安装对设备进行输入输出处理的设备驱动器的驱动器安装工具的操作系统中，上述操作系统将要求实时响应设备来的中断的处理作为设备驱动器进行存取，构成实时控制机构，实时控制机构包括：与对设备进行处理对应构成的实时处  
10 理程序；对实时处理程序进行上述 CPU 资源的分配和调度的第二调度工具。

有关本发明的第二方面，在本发明第一方面的实时控制系统中，实时控制机构具有在实时处理程序间进行数据收发的实时处理程序间通信工具。

15 有关本发明的第三方面，在本发明第一方面的实时控制系统中，实时控制机构具有在操作系统上工作的应用处理程序和实时处理程序之间进行数据收发的处理程序间通信工具。

有关本发明的第四方面，在本发明第一方面的实时控制系统中，实时控制机构具有把在操作系统上工作的应用处理程序作为实时处理程序  
20 登录的实时处理程序装入机构。

有关本发明的第五方面，在本发明第一方面的实时控制系统中，实时控制机构设有操作系统保护工具，使实时处理程序可以利用操作系统对设备驱动器提供服务。

有关本发明的第六方面，在本发明第一方面的实时控制系统中，实  
25 时控制机构具有使用权监视工具，第 2 调度工具测量并保持操作系统占有 CPU 资源的时间，根据该测量结果强制性地使 CPU 资源返回操作系统。

有关本发明的第七方面，在本发明第一至第六方面的实时控制系统  
30 中，当存在多个实时控制机构时，具有调整多个实时控制机构之间的主从关系或优先顺序关系的竞争控制工具。

图 1 是表示本发明第 1 实施形态的系统构成图。

图 2 是本发明第 1 实施形态中的表示 PC - OS 的中断处理的流程

图。

图 3 是本发明第 1 实施形态中的表示 PC - OS 的设备驱动器输入输出请求处理的流程图。

5 图 4 是本发明第 1 实施形态中、当 RT 处理程序作为实时控制机构的程序已被安装时、表示第 2 调度工具的处理的流程图。

图 5 是本发明第 1 实施形态中、当 RT 处理程序具有独自的上下文时、表示第 2 调度工具的处理的流程图。

图 6 是本发明第 1 实施形态中、表示 RT 处理程序间通信工具的处理的流程图。

10 图 7 是本发明第 1 实施形态中、当 PC 处理程序接受数据时、表示 PC 处理程序间通信工具的处理的流程图。

图 8 是本发明第 1 实施形态中、当 PC 处理程序发送数据时、表示 PC 处理程序间通信工具的处理的流程图。

图 9 是表示本发明第 2 实施形态的系统构成图。

15 图 10 是在本发明第 2 实施形态中、表示使用权监视工具的处理流程图。

图 11 是在本发明第 2 实施形态中、表示第 2 调度工具的处理的流程图。

20 图 12 是在本发明第 2 实施形态中、表示使用权监视工具的另一处理例的流程图。

图 13 是本发明第 3 实施形态的系统构成图。

图 14 是在本发明第 3 实施形态中、表示竞争控制工具的处理的流程图。

图 15 表示先有的实时控制系统的系统构成图。

25 图 16 是表示先有的实时控制系统的另一个系统构成图。

实施形态 1

根据图 1 至图 8 说明本发明的第 1 实施形态。

图 1 是表示第 1 实施形态的系统构成的图。

30 在图 1 中， 101 是第 2 调度工具，是对来自 CPU 之外设备的中断、在 PC - OS 调出之前登录的中断服务程序，它向 RT 处理程序 102a 和 102b 分配 CPU 使用权，同时管理 RT 处理程序的执行状态。

103 是提供 RT 处理程序 102a 和 102b 之间的数据交换和同步处理的

RT 处理程序间通信工具，104 是提供 RT 处理程序和 PC 处理程序之间数据交换和同步处理功能的 PC 处理程序间通信工具。

105 是用于将制作和调试好的作为 PC 处理程序的程序作为 RT 处理程序来执行的 RT 安装工具。106 是用于使本来是 PC - OS 对设备驱动器提供的 PC - OS 功能可以由 RT 处理程序来使用的 PC - OS 保护工具。

107 表示 PC 实时控制机构，PC - OS 将其视为 1 个设备驱动器，内部具有 101 - 106 的各个工具和多个 RT 处理程序。108 是 PC - OS 内部具有的控制 PC 处理程序的执行顺序的第 1 调度工具，109 是提供用于将设备驱动器编入 PC - OS 的框架的驱动器编入机构，110 表示 PC - OS。

111a ~ c 是在 PC - OS 上工作的应用程序、即 PC 处理程序。

下面按照 (1) PC 实时控制机构、(2) 伴随 PC 处理程序的输入输出要求的驱动器的动作、(3) 将 RT 处理程序作为实时控制机构的 1 个程序进行控制时的第 2 调度工具、(4) 具有独立上下文的 RT 处理程序安装时的第 2 调度工具、(5) RT 处理程序间的通信处理、(6) PC 处理程序间的通信处理、(7) RT 安装工具、(8) PC - OS 保护工具的顺序说明其动作。

(1) 首先，说明 PC 实时控制机构是如何执行的。

20 从 PC - OS (110) 来看，PC 实时控制机构 107 是 1 个设备驱动器，设备驱动器的执行是由 CPU 之外的设备的中断和 PC 处理程序的输入输出请求启动的。

图 2 示出从通常 OS 中的中断发生开始到对应的设备驱动器的处理结束为止的处理流程。

25 当发生中断时，CPU 在 OS 初始化时启动对 CPU 登录由 PC - OS (110) 的程序。在已启动的 OS 程序中，首先在步骤 201 保存中断发生时的系统的状态。

接着，在步骤 202 检查是否事先登录了与已发生的中断对应的处理程序 (以下称中断服务程序)。如果在步骤 202 没有登录中断服务程序则进到步骤 203，作为预想之外的中断发生而停止系统，或者什么都不做而进到步骤 205，恢复系统状态并结束处理。究竟进行上述哪一种处理因 PC - OS 的种类而异，因与本发明无直接关系，故在此省略其说明。

若在步骤 202 已登录了中断服务程序，则在步骤 204 调出上述中断服务程序，当中断服务程序处理完毕，进到步骤 205，恢复在步骤 201 保存的系统的状态，结束中断服务处理。

在步骤 204 调出的中断服务程序中，与已发生了中断的设备进行数据交换（当是输入数据的中断时，从设备取出数据，当是数据输出结束的中断时，将输出数据向设备送出），结果，如果存在有等待输入输出请求的 PC 处理程序，就解除该 PC 处理程序的等待状态，调出第 1 调度工具 108，进行 PC 处理程序的再调度。

再有，当在通常的 OS 下从中断服务程序直接调出调度工具时，系统处理出现矛盾，因此，具有当中断处理结束后能够调出第 1 调度工具的某种机构。

(2) 下面，根据图 3 的流程图说明伴随 PC 处理程序 111 的输入输出请求的设备驱动器的执行处理。

当从 PC 处理程序发出输入输出请求时，调出作为 PC - OS (110) 的一部分的驱动器编入机构。在驱动器编入机构中，启动与 PC 处理程序来输入输出请求的种类对应的设备驱动器的请求接收程序（步骤 301）。该请求接收程序是例如分为数据的读入、写出和设备的直接控制等由设备驱动器的初始化处理将其登录在驱动器编入机构中的程序。在设备驱动器的请求接收程序中，根据请求内容与设备进行数据交换，但是，因设备的速度与 CPU 相比非常低，所以，在驱动器编入机构中，使 PC 处理程序成为等待输入输出结束的状态（步骤 302）。

接着，在步骤 303 中，因在步骤 302 中 PC 处理程序处于等待状态，故为了选择下面要执行的 PC 处理程序而调出第 1 调度工具，从而结束输入输出请求的处理。

处于等待输入输出结束状态的 PC 处理程序由图 2 所示的中断服务程序来解除等待状态。

如上所述，PC - OS 具有的第 1 调度工具 108 不在设备驱动器处理中执行，结果，设备驱动器相对于 PC 处理程序 111a ~ c、进而相对于第 1 调度工具 108 具有优先处理权。

(3) 下面，根据图 4 说明控制 RT 处理程序执行的第 2 调度工具 101 的动作。

图 4 是表示当 RT 处理程序作为 PC 实时控制机构的 1 个程序已被安

装时的第 2 调度工具 101 的处理的流程图。

当在图 2 的步骤 204 调出第 2 调度工具 101 时，首先执行步骤 401，检查表示 PC 实时控制机构正在执行中的标志，若标志是 ON，则什么都不做而结束处理，并返回图 2 的步骤 205。

5 在步骤 401 中，若标志是 OFF，则在步骤 402 使标志变成 ON，再进到步骤 403，检查是否有可执行的 RT 处理程序。作为检查方法，例如，将 RT 处理程序的起始地址（在本实施形态中是例程序的起始地址）保存在表中，保存在表中的 RT 处理程序可以判定为全部处于可执行状态。此外，若设备来的中断是周期地发生的，只要在上述表中设置计数器，就可以以不同的周期时间启动每一个 RT 处理程序。

10 在步骤 403 中，若存在处于可执行状态的 RT 处理程序，则进到步骤 404，调出登录在表上的例程序。在调出的例程序（RT 处理程序）中，通过在进行 RT 处理程序特有的处理之后的返回指令，返回步骤 404，其后进到步骤 403。通过反复执行步骤 403 和步骤 404 对登录在表上的可执行的所有 RT 处理程序进行处理，然后进到步骤 405，将标志置 0 并结束处理。

这时，RT 处理程序如图所示通过从例程序返回而结束处理。

（4）下面，根据图 5 的流程图说明当 RT 处理程序不是作为 PC 实时控制的 1 个例程序、而是作为具有独立的上下文的程序安装时的第 2 调度工具 101。

20 这是，如图 5 所示，RT 处理程序成为反复处理处理程序本身的初始化处理和程序特有的处理状态。反复处理的结束通过利用等待调用自动放弃执行权来进行。

第 2 调度工具 101 一旦从图 2 的步骤 204 中调出，就首先执行步骤 25 401 和步骤 402。以上步骤与图 4 的相应的步骤相同。

接着，在步骤 403 检查可执行状态下的 RT 处理程序，检查方法与说明图 4 的流程图的情况一样。在步骤 403 中，若存在处于可执行状态的处理程序，则进到步骤 501，返回在步骤 403 所得到的处理程序的状态信息。RT 处理程序通过该处理再度从步骤 503 所示的地方开始执行，这 30 里，说明 RT 处理程序的处理。

当最初启动 RT 处理程序时，进行本身的初始化处理（步骤 502），进到步骤 503。RT 处理程序的最初启动可以从设备驱动器的 PC 实时控



制机构的初始化处理中调出，出可以从 PC 处理程序产生的设备启动请求中调出。

在步骤 503 中，在进行了 RT 处理程序的固有的处理之后，在步骤 504 中调用等待子程序。

5 等待调用是 PC 实时控制机构对 RT 处理程序提供的功能，是暂时中断 RT 处理程序的执行的功能。该功能在本发明中不是必须的功能，只是为了说明本实施形态的。将已调用的 RT 处理程序排除在可执行状态之外（步骤 505），保存该处理程序的状态信息（步骤 506），然后，再次调出第 2 调度工具 101（步骤 403）。

10 步骤 501 中 RT 处理程序的状态信息所返回的是在上述步骤 506 中已保存的状态信息。RT 处理程序从步骤 503 再开始执行。

反复执行以上的步骤 403、步骤 501 到步骤 506，当通过等待调用使所有处于可执行状态的 RT 处理程序排除在可执行状态之外时，在步骤 403 判定没有可执行的程序，在步骤 405，将标志变成 OFF 并返回到 PC - OS 的中断处理（图 2 流程图的步骤 204）。

（5）下面，说明 RT 处理程序间通信工具的处理。

在图 4 和图 5 的流程图中，已就 RT 处理程序各个独立的处理进行了说明，但一般的处理程序则在进行相互间的同步处理和数据交换的同时进行处理。

20 下面，以在 2 个 RT 处理程序 102a、102b 中进行最简单的数据交换的情况为例说明处理的流程。

图 6 是在 2 个 RT 处理程序 102a、102b 中 RT 处理程序 102a 进行数据的接收、RT 处理程序 102b 进行数据的发送的图。

25 首先，当 RT 处理程序 102a 在步骤 601 中进行数据的接收时，调出 RT 处理程序间通信工具的步骤 602，并检查有无数据。若存在数据，则将数据复制到调出 RT 处理的数据接收时所指定的存储区（步骤 603），数据接受处理结束，在 RT 处理程序的步骤 504a 调出等待，从可执行状态退出。

30 若在步骤 602 没有数据，则在步骤 604 将 RT 处理程序 102a 从可执行状态退出，在步骤 605 保持 RT 处理程序的状态信息，在步骤 606 调出第 2 调度工具 101。这里，调出的第 2 调度工具与上述在等待处理中说明过的情况一样，是图 5 的步骤 403。

第 2 调度工具 101 的处理结果是，当选择 RT 处理程序 102b 并返回状态信息时，RT 处理程序 102b 紧接在等待之后重新执行，进到步骤 607 并调用数据发送。当调出数据发送时，执行 RT 处理程序间通信工具的步骤 608，检查有无数据等待状态的处理。

5 若不存在数据等待状态下的 RT 处理程序，则在步骤 609 将数据复制到 RT 处理程序所具有的存储区，结束数据的发送，返回 RT 处理程序 102b，并利用等待从可执行状态退出。在本说明中，因 RT 处理程序 102a 在步骤 604 处于数据等待状态，故在步骤 608 判定为有数据等待处理程序，从而进到步骤 610。

10 在步骤 610 中，将数据复制到 RT 处理程序所指定的存储区，在步骤 611 中使 RT 处理程序成为可执行状态。由此结束数据发送处理，返回 RT 处理程序 102b，并在步骤 504b 调出等待，RT 处理程序 102b 从可执行状态退出。

此后，在图 5 的流程图的步骤 403，选择 RT 处理程序 102a 作为可执行状态的处理程序，在步骤 501 恢复状态时，返回到步骤 606。

因此，RT 处理程序 102a 的数据接收处理宣告结束，若在步骤 504a 调用等待，则从可执行状态退出，根据第 2 调度工具 101，因没有可执行状态下的 RT 处理程序，故返回 PC - OS 的中断处理。

(6) 下面，根据图 7、图 8 说明 PC 处理程序间通信工具的处理。

20 在 PC - OS 等通常的计算机系统中，在 PC 处理程序之间或 RT 处理程序之间可以相互进行同步处理或进行数据交换，但在 PC 处理程序和 RT 处理程序混合存在的情况下，也有必要在 PC 处理程序和 RT 处理程序之间进行数据的交换和同步处理。

下面，以在 PC 处理程序和 RT 处理程序中进行最简单的数据交换的情况为例分别示出处理的流程。图 7 是 PC 处理程序进行数据的接收、RT 处理程序进行数据的发送的图。此外，图 8 是 PC 处理程序进行数据的发送、RT 处理程序进行数据的接收的图。

实际上，在 PC 处理程序中，数据的发送与对相当于 PC 机实时控制机构的设备驱动器的数据输出请求对应，数据的接收与数据输入请求对应，可以以这种形式向 PC 处理程序提供功能，也可以向 PC 处理程序提供 1 个库，该库使数据输出请求与数据发送对应、数据输入请求与数据接收对应。

首先，参照图 7 说明 PC 处理程序进行数据的接收，RT 处理程序进行数据的发送情况下的动作。当利用第 1 调度工具 108 执行 PC 处理程序并在步骤 701 要求数据接收时，使用驱动器编入机构调出 PC 实时控制机构的 PC 处理程序间通信工具作为数据输入要求。

5        PC 处理程序间通信工具首先在步骤 702 检查有无数据。若存在数据则进到步骤 703，与通常的设备驱动器一样，作为数据输入结束向驱动器编入机构返回控制，结果，作为数据通信结束，PC 处理程序从步骤 701 接下来执行。

10        在步骤 702 中，若没有数据则进到步骤 704，将 PC 处理程序作为设备来的数据输入等待状态通知给 PC - OS。

接着，进到步骤 705，作为数据输入要求结束，向驱动器编入机构 109 返回控制。在驱动器编入机构中，因 PC 处理程序变成设备输入等待状态，故调出 PC - OS 的第 1 调度工具 108，成为不同于执行设备输入等待状态的 PC 处理程序的 PC 处理程序。

15        作为到 PC - OS 的设备中断调出第 2 调度工具，结果，当执行 RT 处理程序时，调出 PC 处理程序间通信工具的步骤 706，并检查有无等待数据接收的处理程序。

20        在步骤 706 中，若没有等待数据接收的处理程序，则在步骤 707 保持数据并返回 RT 处理程序。本实施形态因为 PC 处理程序是等待数据接收，故进到步骤 708，PC 处理程序把数据复制到数据输入请求时指定的 PC 处理程序的存储区，在步骤 709 中把已解除 PC 处理程序的等待数据输入状态通知给 PC - OS。

25        在 PC - OS 中，使 PC 处理程序从输入等待状态变成可执行状态，但如图 2 的流程图所说明的那样，因为 PC - OS 知道正在进行设备中断处理，故不调出第 1 调度工具 108 而在步骤 709 返回控制。由此结束 PC 处理程序间通信工具的数据发送处理，向 RT 处理程序返回控制。

30        如果所有可执行的 RT 处理程序均被处理的话，则从第 2 调度工具 101 返回到 PC - OS 的中断处理，PC - OS 如图 2 的流程图所说明的那样，因为在中断处理内存在已解除了等待设备输入的 PC 处理程序，故调出第 1 调度工具 108，（依情况而定）以数据接收结束的形式再次执行 PC 处理程序。

下面，根据图 8 说明 PC 处理程序进行数据的发送、RT 处理程序进

行数据的接收的情况。

从来自设备的中断调出第 2 调度工具 101，结果就执行 RT 处理程序，如果调用数据接收，就调用 PC 处理程序间通信工具。

5 在 PC 处理程序间通信工具中，首先在步骤 801 检查有无数据，若有数据则将数据复制到 RT 处理程序指定的存储区（步骤 802），结束数据接收并向 RT 处理程序返回控制。

10 在步骤 801 中，若没有数据，则在步骤 803 使 RT 处理程序作为数据接收等待状态从可执行状态退出，在保持了状态信息之后（步骤 804），调出第 2 调度工具 101（步骤 805）。这时，调出的第 2 调度工具与说明了的等待的情况一样，是图 5 的步骤 403。

当等待数据接收的 RT 处理程序之外的所有可执行状态的 RT 处理程序都被处理时，从第 2 调度工具 101 返回 PC - OS 的中断处理，结束设备中断处理。结果，调用 PC - OS 的第 1 调度工具 108。

15 再有，根据操作系统，当设备中断处理结束时，（在设备驱动器处理中设备输入等待的处理程序连 1 个等待状态都没有被解除的情况）有时返回到中断发生的地方。虽然多数情况是不管正在进行哪个 PC 处理程序，伴随中断处理的结束都可以再次开始处理，但是，因为 PC - OS 对各个 PC 处理程序平等地分配 CPU 资源，所以无论那种情况都是由第 1 调度工具 108 来执行 PC 处理程序。

20 在 PC 处理程序中，当在步骤 806 请求数据发送时，由驱动器编入机构调出作为数据输出请求的 PC 实时控制机构的 PC 处理程序间通信工具。在 PC 处理程序间通信工具中，首先在步骤 807 检查有无数据等待状态的 RT 处理程序。若存在数据等待状态的 RT 处理程序，则在步骤 808 把数据复制到 PC 处理程序间通信工具所有的存储区。

25 接着，在步骤 809 向驱动器编入机构通知数据输出要求结束并结束处理。结果，作为数据通信结束再次执行 PC 处理程序。本实施形态因为 RT 处理程序处于数据等待状态，故从步骤 807 进到 810，将数据复制到 RT 处理指定的存储区。

30 接着，在步骤 811 使 RT 处理程序成为可执行状态，进到步骤 809，向驱动器编入机构通知数据输出请求结束并结束处理。在步骤 811，处于可执行状态的 RT 处理程序根据下 1 个设备驱动器的中断、由第 2 调度工具 101 来执行，结束 RT 处理程序的数据接收处理。

再有，在步骤 811，在 RT 处理程序成为可执行状态之后通知数据输出请求结束，但是，为了提高对 RT 处理程序的响应速度，也可以在通知数据输出要求结束之前调出第 1 调度工具 101 来执行 RT 处理程序、这时，在通知数据输出要求结束之后，必须由 RT 处理程序所调出的等待来结束处理。

(7) 下面，说明 RT 安装工具 105 的处理。

在一般的计算机系统中，设备驱动器的开发比应用程序开发更要小心，更加困难。在本发明的 PC 实时控制机构中，把 RT 处理程序作为设备驱动器的内部程序来执行，所以，与先有的作为 RT - OS 的应用程序而开发的 RT 处理程序相比，要困难一些。

因此，RT 安装工具 105 将作为 PC 处理程序 111c 开发调试的应用作为设备驱动器，即 PC 实时控制机构的 RT 处理程序 102b 编入。

RT 安装工具 105 在很大程度上依赖于将 RT 处理程序作为 PC 处理程序制作并调试时的限制。例如，在要求通过 RT 安装工具把作为 PC 处理程序的 1 个程序作成的 RT 处理程序作为 RT 处理程序登录时，RT 安装工具只将从指定的程序开始的地址开始的程序大小复制到设备驱动器、即 PC 实时控制机构内部的存储区内，并将程序的开始地址登录在已用图 4 的流程图说明过的表中。

在将作为 PC 处理程序而作成并调试好的 PC 处理程序整个作为 RT 处理程序登录时，RT 安装工具例如可以把存储在外存储器上的 PC 处理程序读入到 PC 实时控制机构内的存储区内，并将其开始地址登录在已用图 4 的流程图说明过的表中。

在以上的 RT 安装工具中，在利用把作为 RT 处理程序作成并调试好的程序或 PC 处理程序提供给 PC - OS 的功能（系统调用等）时，当然必须在 PC 实时控制机构内部提供相同的功能。此外，因为这些功能只是作为子程序 PC 处理程序中移植过来的，所以在 PC 实时控制机构内当然也有必要在安装或执行 RT 处理程序的同时解决这些地址。

作为该地址的解决方法，例如考虑在 PC 处理程序调试结束后，在地址未解决的状态下只对子程序再进行编译，在复制程序的同时，用 RT 安装工具将未解决的地址替换成 PC 实时控制机构内部的对子程序。

此外尚有几个众所周知的方法，在本发明中可以采用任何一种方法。此外，作为在作成作为 RT 处理程序的 PC 处理程序时的使用限制，

也可以禁止使用 PC - OS 提供的功能。

(8) 下面, 说明 PC - OS 保护工具 106。

具有驱动器编入机构的 PC - OS 从设备驱动器提供可使用的 PC - OS 的功能。这些功能与 PC - OS 向 PC 处理程序提供的功能通常是不同的。此外, 设备驱动器多数是作为 PC - OS 的一部分而工作, 从 RT 处理程序随便使用向驱动器提供的功能可能会引起系统停机的重大问题。

PC - OS 保护工具是一种转换程序, 用于将 PC - OS 向驱动器提供的功能较安全地向 RT 处理程序提供, 进行严密的出错检查和访问地址的检查, 并对引起 PC - OS 操作出现矛盾的处理进行排除等。

10 实施形态 2

下面, 根据图 9 至 12 说明本发明的第 2 实施形态。

901 是在 PC 实时控制机构中监视占有 CPU 的时间的使用权监视工具。因 PC 实时控制机构作为设备驱动器工作, 故不成为第 1 调度工具的调度形象。因此, 当多个 RT 处理程序工作时, 控制并不转移到第 1 调度工具, 从而发生 PC 处理程序不能长期工作的状态。

使用权监视工具监视占有 CPU 的时间, 进行计数, 在适当的条件下强制性地向 CPU 返回控制, 因而可以在 PC 实时控制机构和 PC - OS 之间适当地分配 CPU。

再有, 图中 101 ~ 111 与图 1 所示的相同。

20 下面, 说明其动作。

使用权监视工具作为从图 2 的步骤 204 调出的中断服务程序工作。

首先, 在步骤 1001 判别 PC 实时控制机构是否正在工作。判别方法与图 4 的步骤 401 - 样。

若正在工作, 在步骤 1002 对 PC 实时控制机构工作的时间进行计数。计数方法可以只是计数, 也可以是工作时间的累计。

接着, 在步骤 1003, 判定在步骤 1002 已计数的信息是否满足预定的条件。所谓预定的条件例如可以是 PC 实时控制机构连续占有 CPU 的上限值, 也可以是一定时间内 CPU 占有率的上限值。

在步骤 1003, 若判断满足条件, 即处于强制性地将 CPU 的使用权从 PC 实时控制机构返回 PC - OS 的状态, 则进到步骤 1004, 使强制结束标志置位并结束。

该强制结束标志由第 2 调度工具进行检查, 图 11 示出追加了该检查

处理的第 2 调度工具的处理流程。

图 11 与图 4 只有如下不同：添加了步骤 404 的下 1 个步骤 1101 的强制结束标志。

5 当在步骤 1101 强制结束标志置位时，例如即使存在可执行状态的 RT 处理程序，也停止向 RT 处理程序分配 CPU，将控制返回 PC - OS。

接着，在步骤 1003，若不满足条件，就直接结束使用权监视工具的处理（因 PC 实时控制机构正在工作），RT 处理程序从中断发生的地点继续执行。

10 在步骤 1001 中，若判定 PC 实时控制机构不是正在执行中，则进到步骤 1005，与步骤 1003 一样，根据过去计数的信息检查是否满足条件。该步骤是当一定时间内的 CPU 占有率在上限值之上时才有效地处理。

在步骤 1005，若判断满足条件，就结束使用权监视工具，返回 PC - OS。

15 在步骤 1005，当判断不满足条件、即 PC 实时控制机构可以工作时，则进到步骤 1006，调出第 2 调度工具。当所有的可执行的 RT 处理程序的处理结束时，控制从第 2 调度工具转移到步骤 1007，在进行了 PC 实时控制机构的工作时间的计数之后，向 PC - OS 返回控制。

下面，根据图 12 说明强制性地 CPU 使用权返回 PC - OS 的另一个处理例。

20 图中，步骤 1001 ~ 1007 是与图 10 的流程相同的处理。

在步骤 1003，当判断满足条件时，则进到步骤 1201，将当前处于可执行状态的 RT 处理程序全部从可执行状态退出，结束使用权监视工具的处理。

25 此外，当在步骤 1005 判断不满足条件时，在步骤 1203，将步骤 1201 保持的 RT 处理程序全部返回可执行状态后，在步骤 1006 调出第 2 调度工具。

### 实施形态 3

下面，根据图 13、图 14 说明本发明的第 3 实施形态。

30 在图 13 中，1301a、1301b 是竞争控制工具，当 RT 处理程序跨过 PC 实时控制机构间进行数据交换和同步处理时，执行排他控制。

其它的构成要素和图 1 所示的相同。

下面，根据图 14 的流程图说明动作。

竞争控制工具在本实施形态中是用于向其他 PC 实时控制机构提供 RT 处理程序间的通信和同步等功能的访问指针。作为 PC 实时控制机构, 提供处理程序间通信和同步等功能是自由的, 因与本发明无直接关系, 故在此省略其说明。

5 首先, 在步骤 1401, 检查已交给竞争控制工具的访问指针的处理申请是从自己的 PC 实时控制机构向别的 PC 实时控制机构的处理申请, 还是从别的 PC 实时控制机构来的处理申请。为了进行该检查, 例如, 可以在处理申请的数据中输入能够区别上述情况的代码, 也可以对 PC 实时控制机构指定 1 个特征序号, 并将该序号嵌入处理申请数据中的特定位置。

10 在步骤 1401 中, 若不是来自别的 PC 实时控制机构的处理申请, 则进到步骤 1402, 向别的 PC 实时控制机构的竞争控制工具 (图 13 的 1301a) 送出处理申请数据。

在步骤 1401, 若是来自别的 PC 机实时控制机构的处理申请, 则进到步骤 1403, 调出执行申请的处理。

15 如上所述, 若按照本发明, 以在 1 个 CPU 上工作的操作系统为基础, 使应用处理程序和实时处理程序工作, 所以, 可以容易地跟踪版本的升级, 可以实现硬件成本降低了的处理程序共存系统。

若按照本发明, 在实时处理程序间或在实时处理程序和应用处理程序程序之间进行数据交换, 所以, 可以实现实时处理程序和应用处理程序程序的协调动作。

20 若按照本发明, 将实时处理程序作为应用处理程序程序开发之后, 再安装实时处理程序并执行, 所以, 可以容易地将应实现的功能作为实时处理程序进行编程。

25 若按照本发明, 把操作系统本来提供给输入输出装置的设备驱动器用的服务也向实时处理程序提供, 所以, 即使有作为实时处理程序和应用处理程序程序开发的程序也能安全地工作。

若按照本发明, 监视实时处理程序占有 CPU 资源的时间, 不使其超过一定的时间, 所以, 对应用处理程序程序的使用者不损害响应性能, 又可以执行实时处理程序。

30 若按照本发明, 在多个实时控制机构间实现了存取竞争控制机构, 所以, 可以在 1 个操作系统下使具有多种特性的实时处理程序同时工作。



# 说明书附图

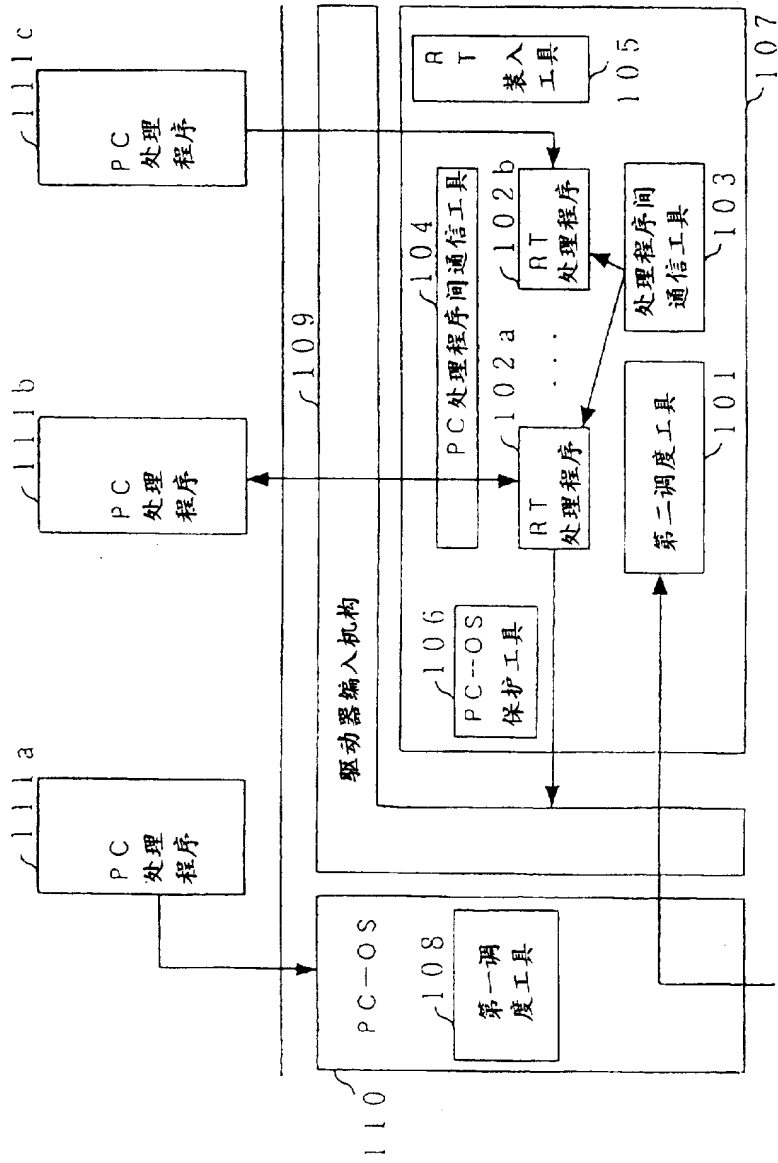


图 1

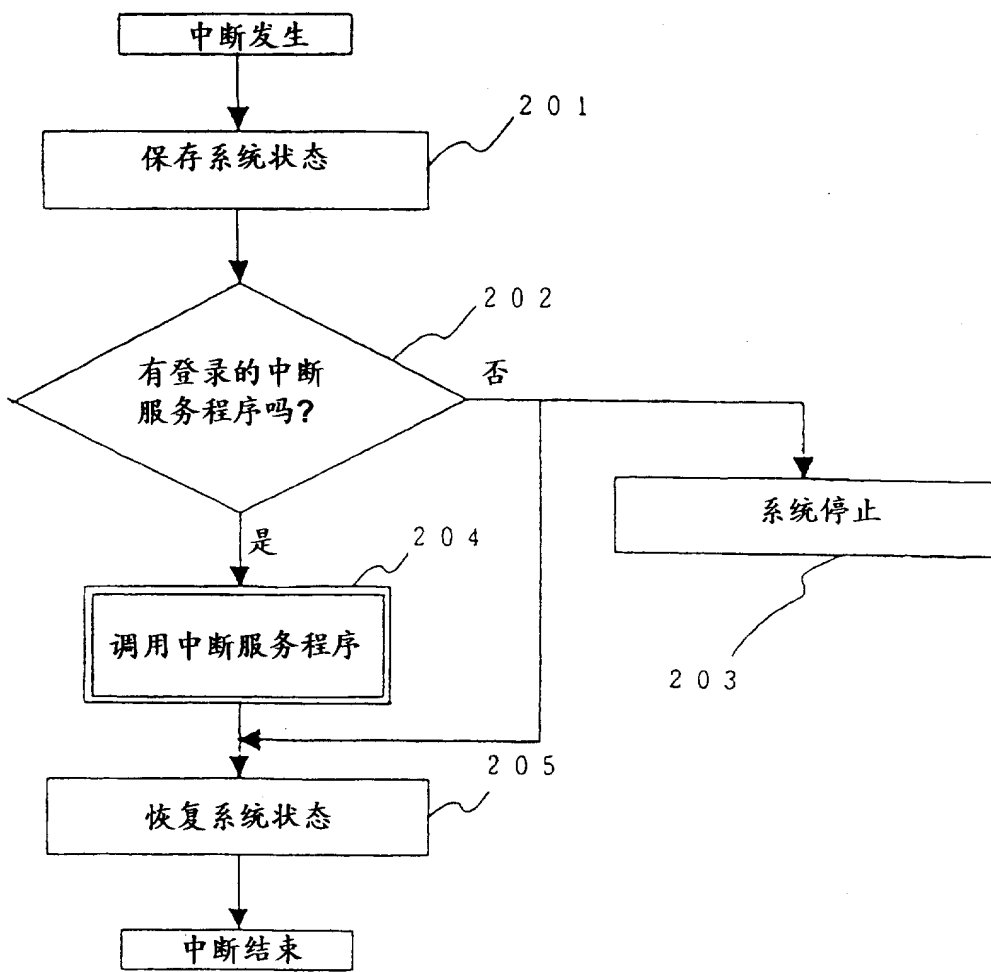


图 2

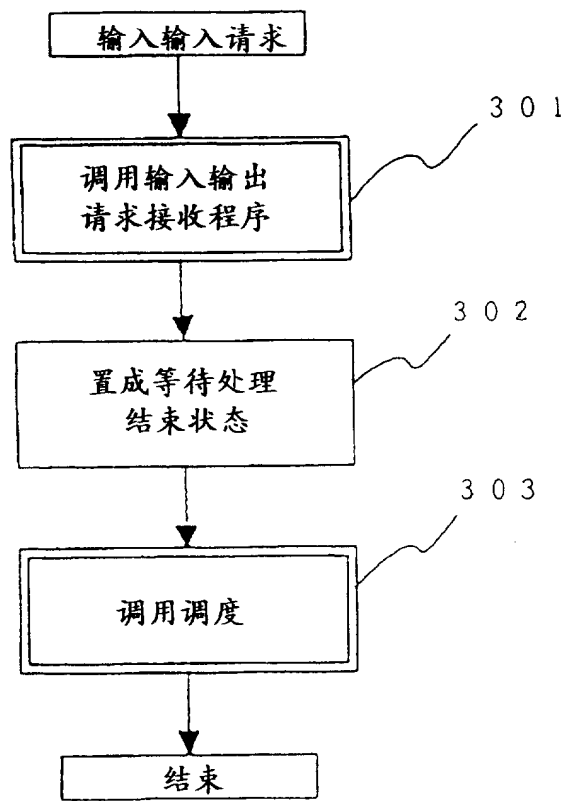


图 3

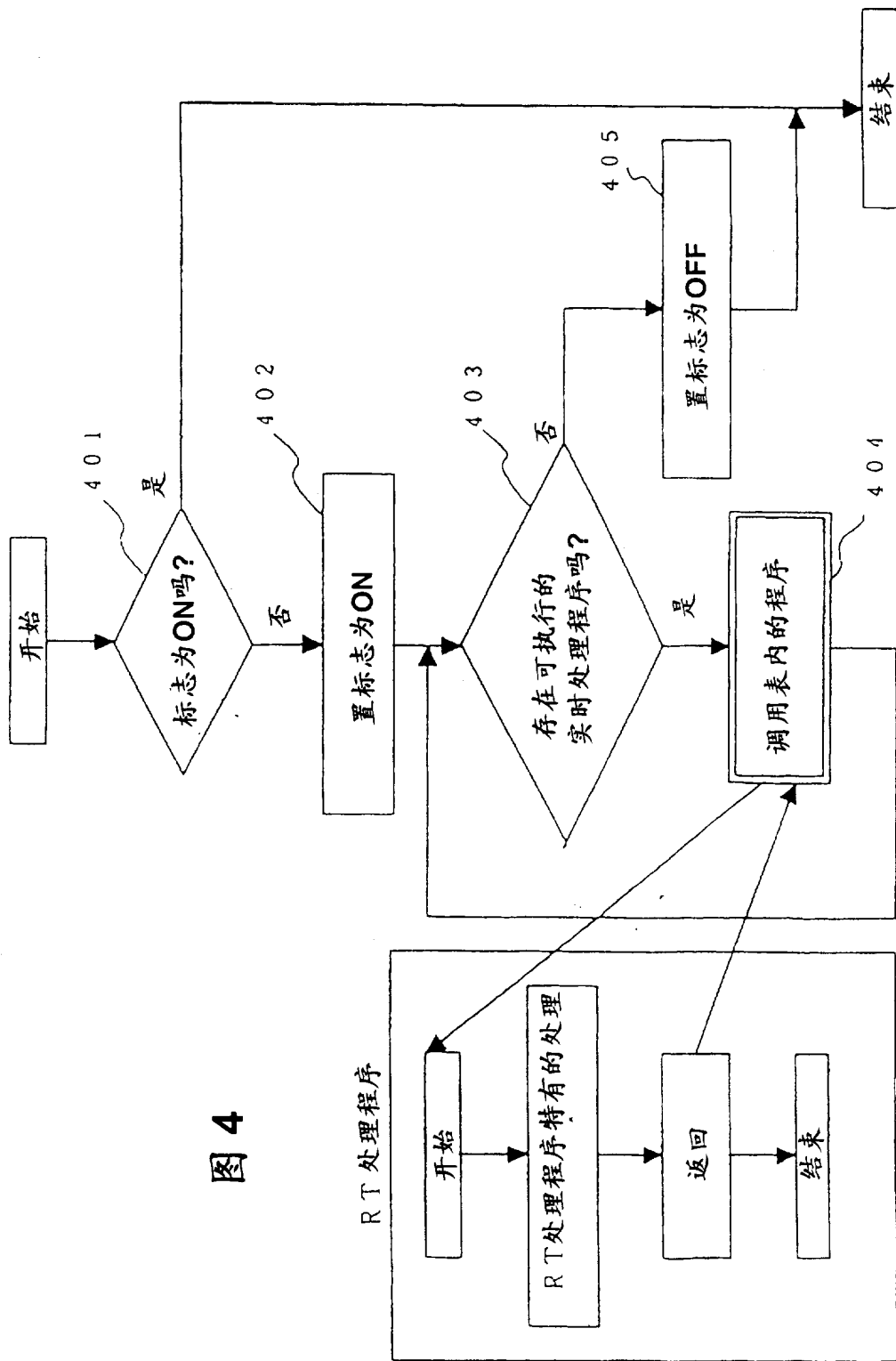


图 4

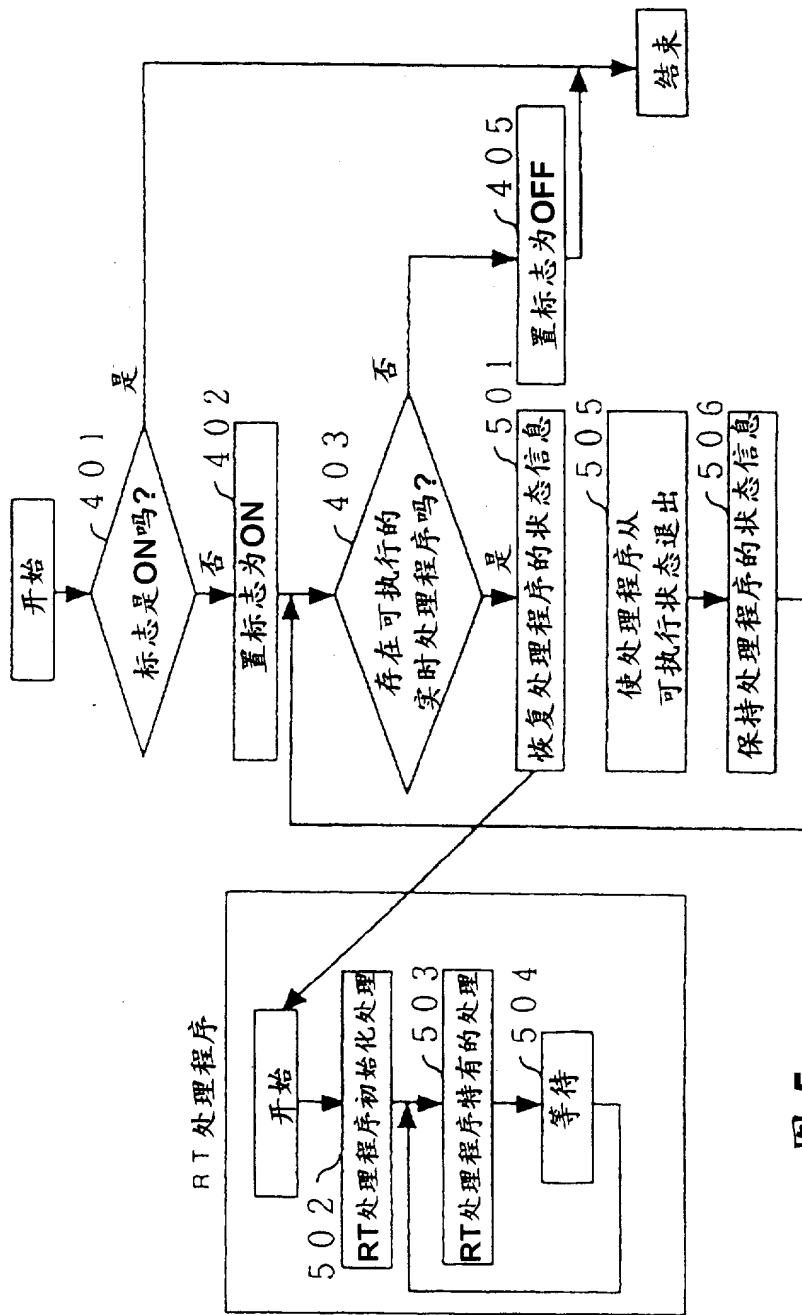


图 5

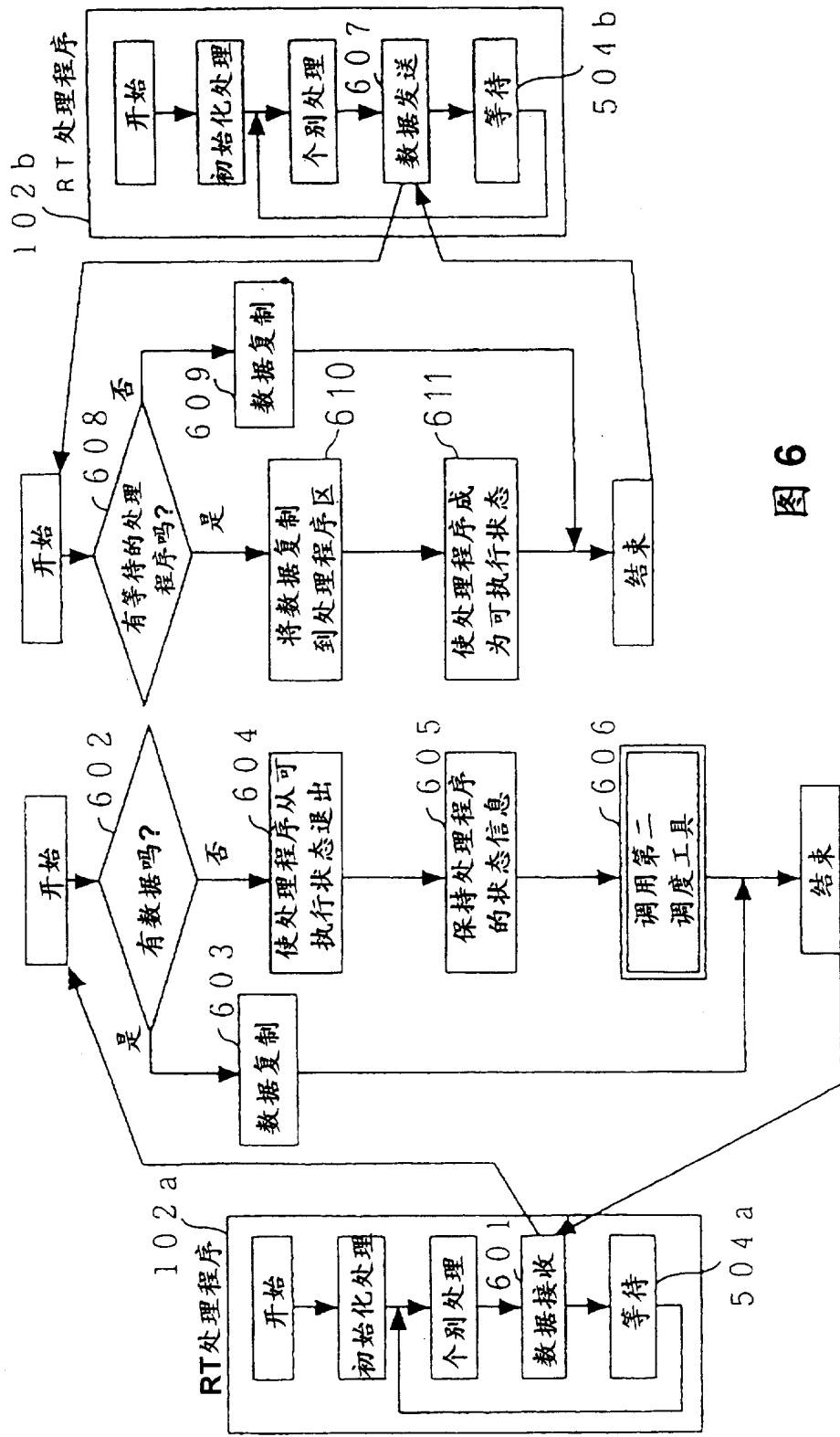


图 6

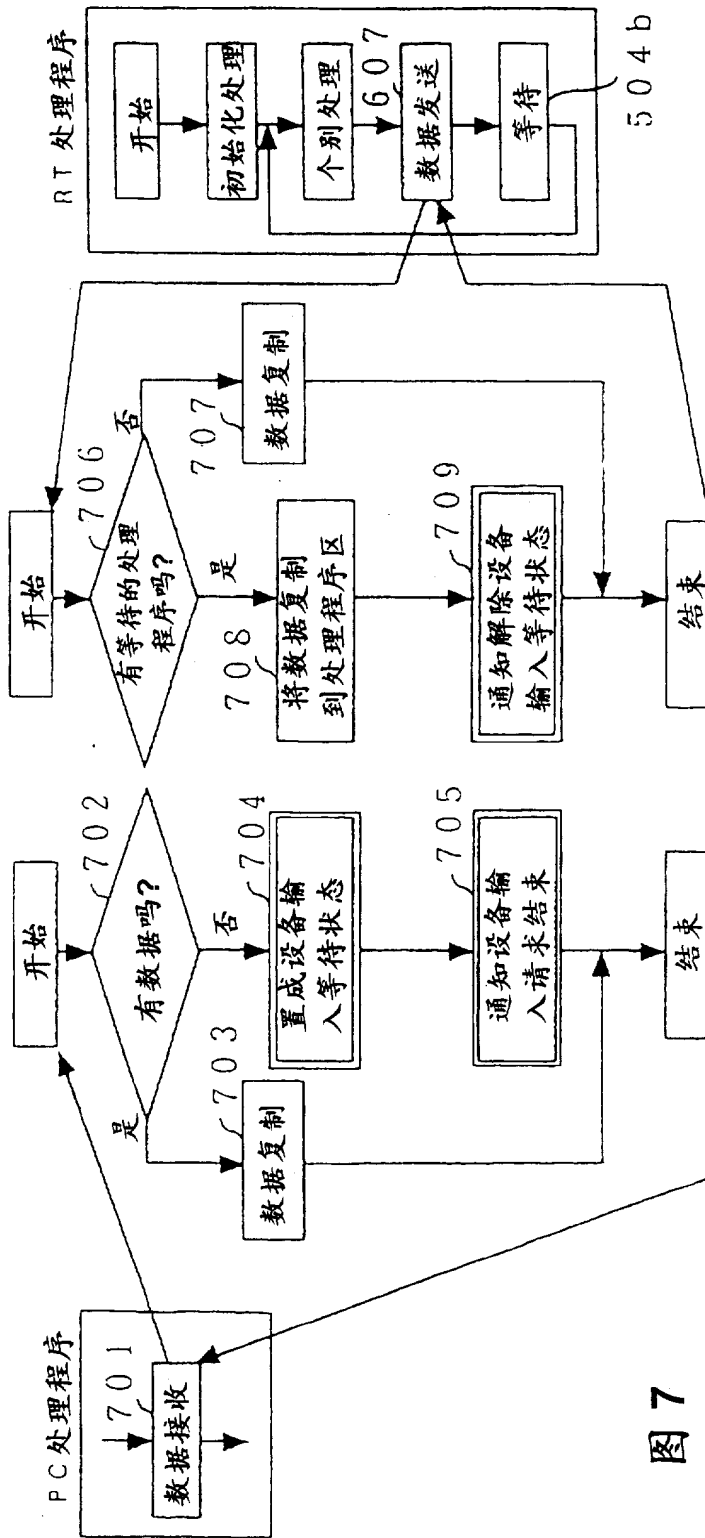


图 7

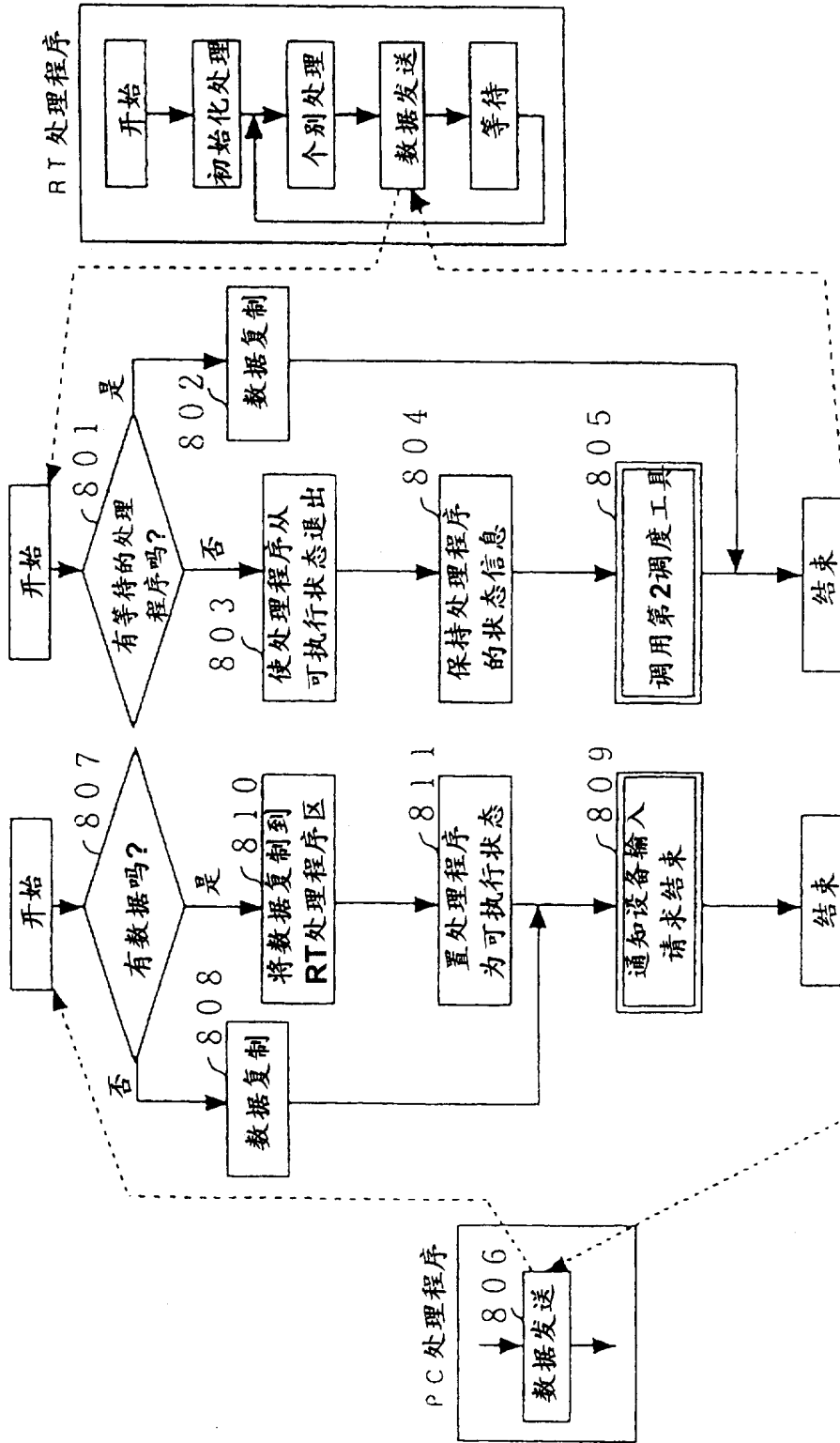


图 8



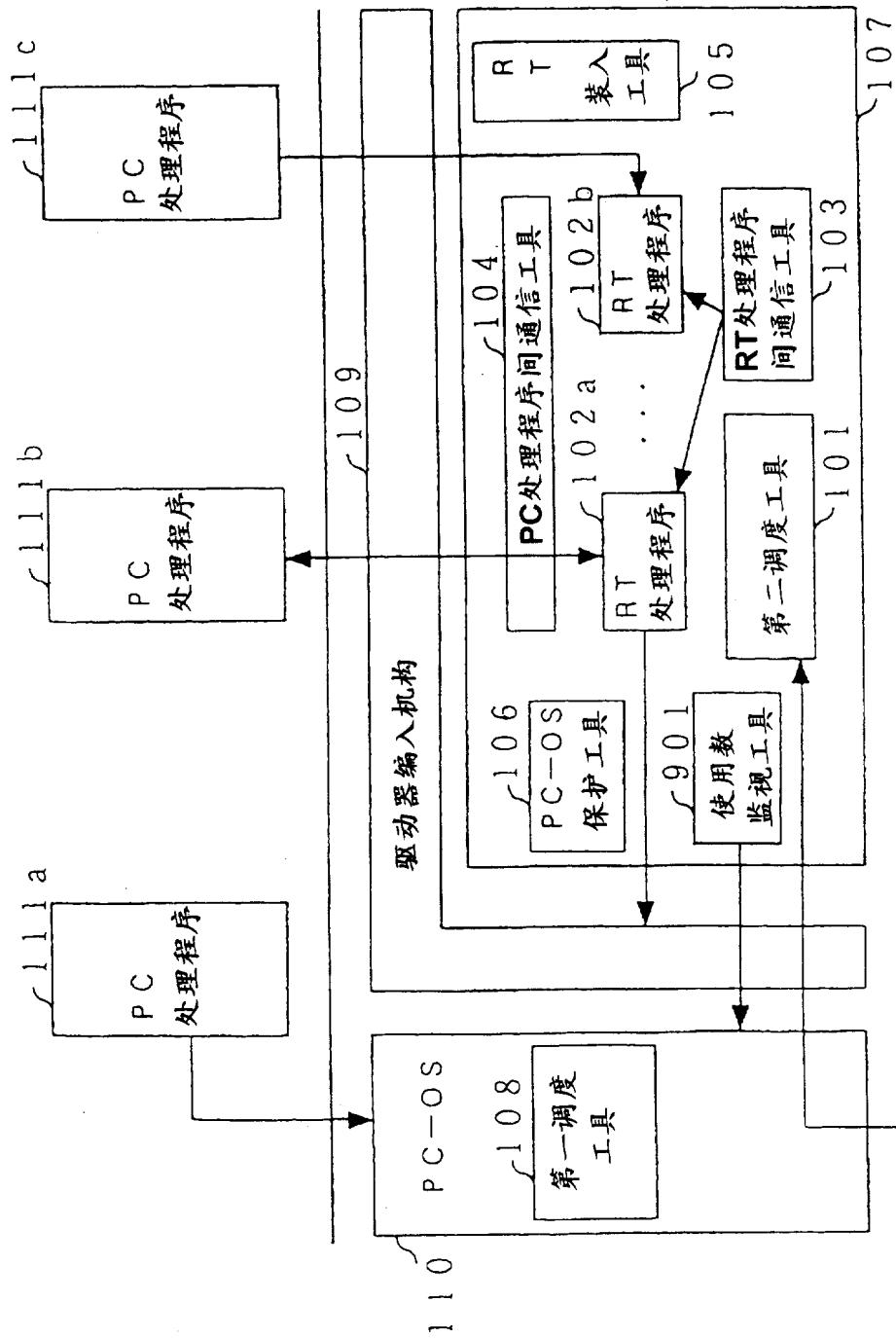


图 9

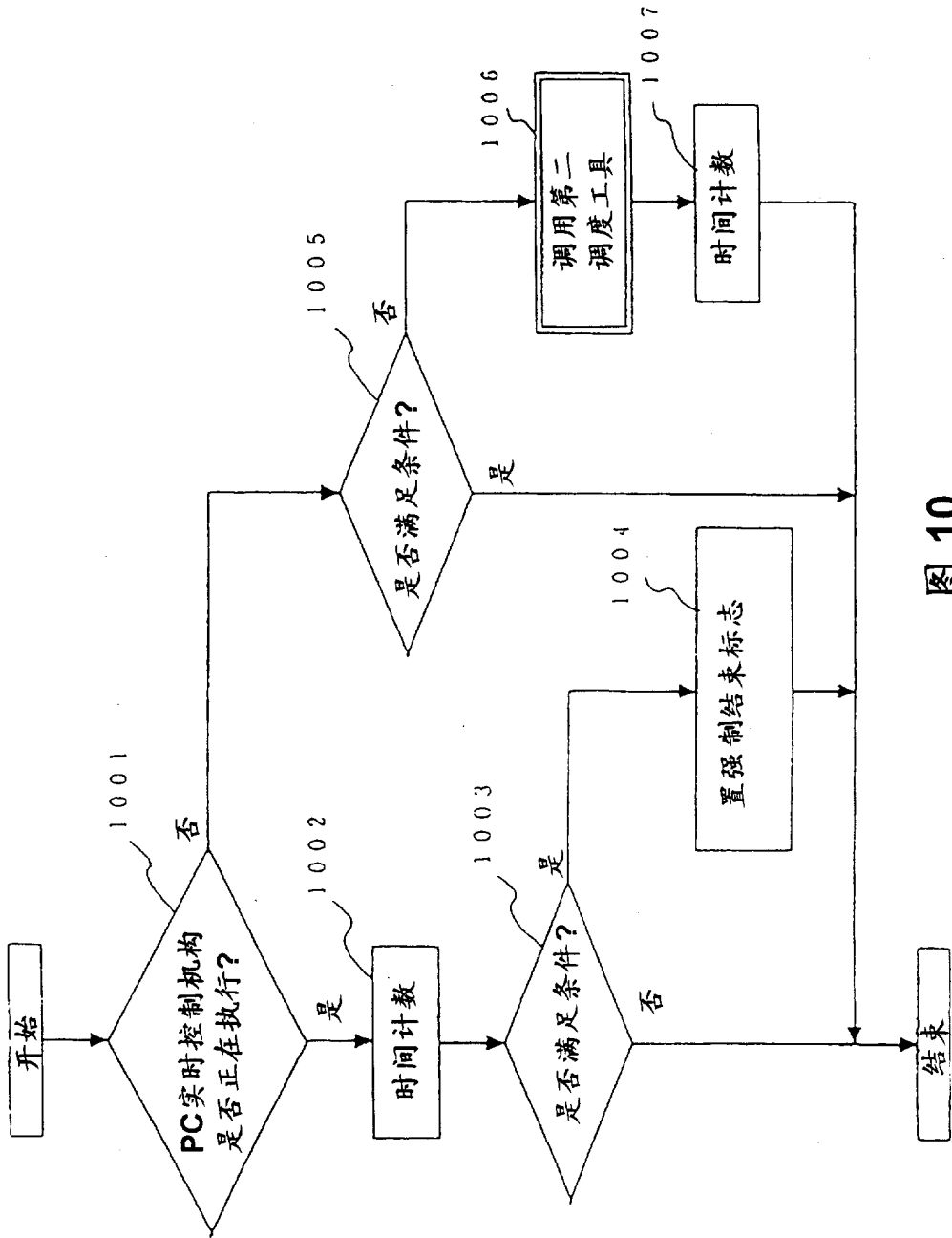


图 10

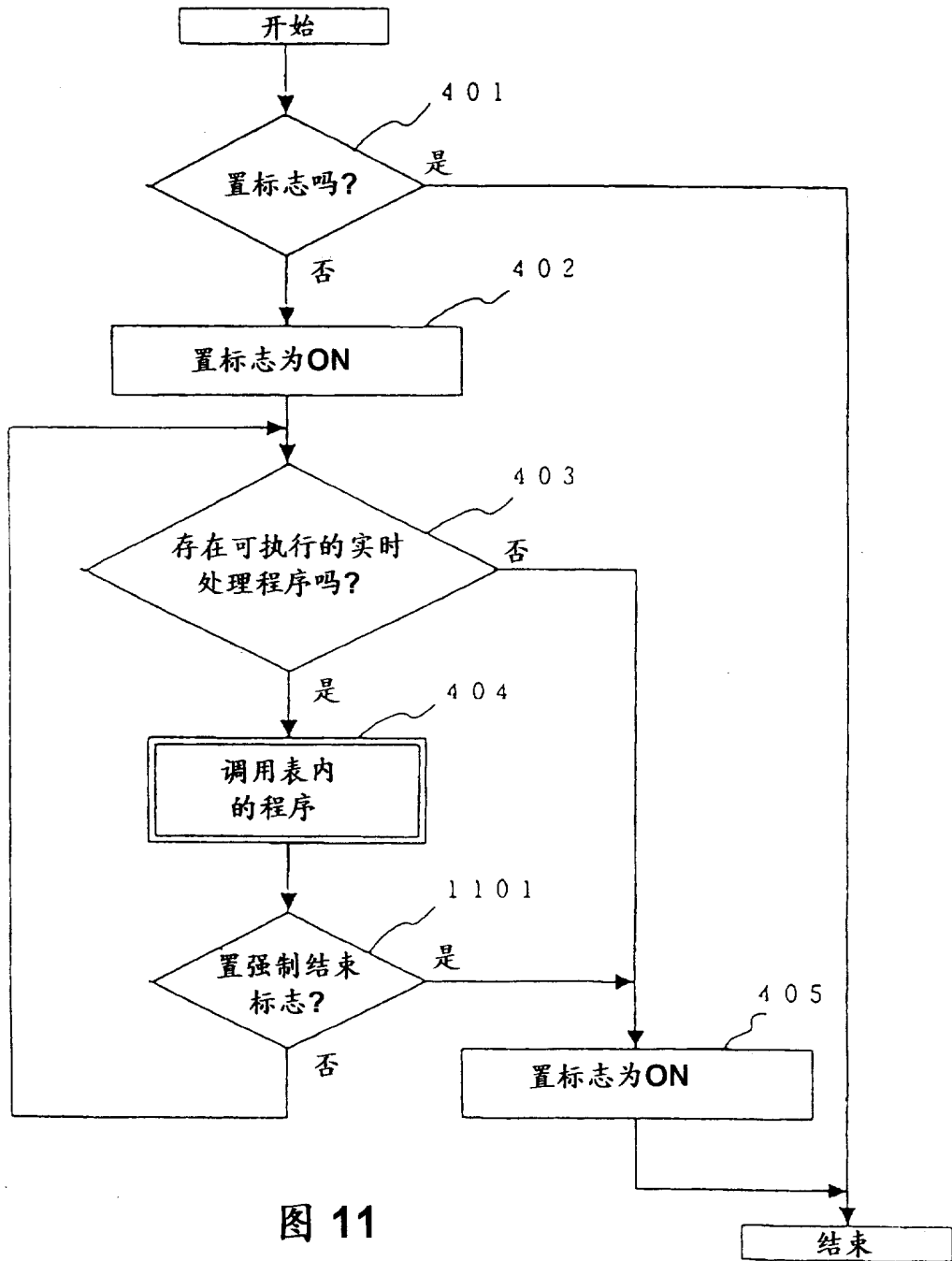


图 11

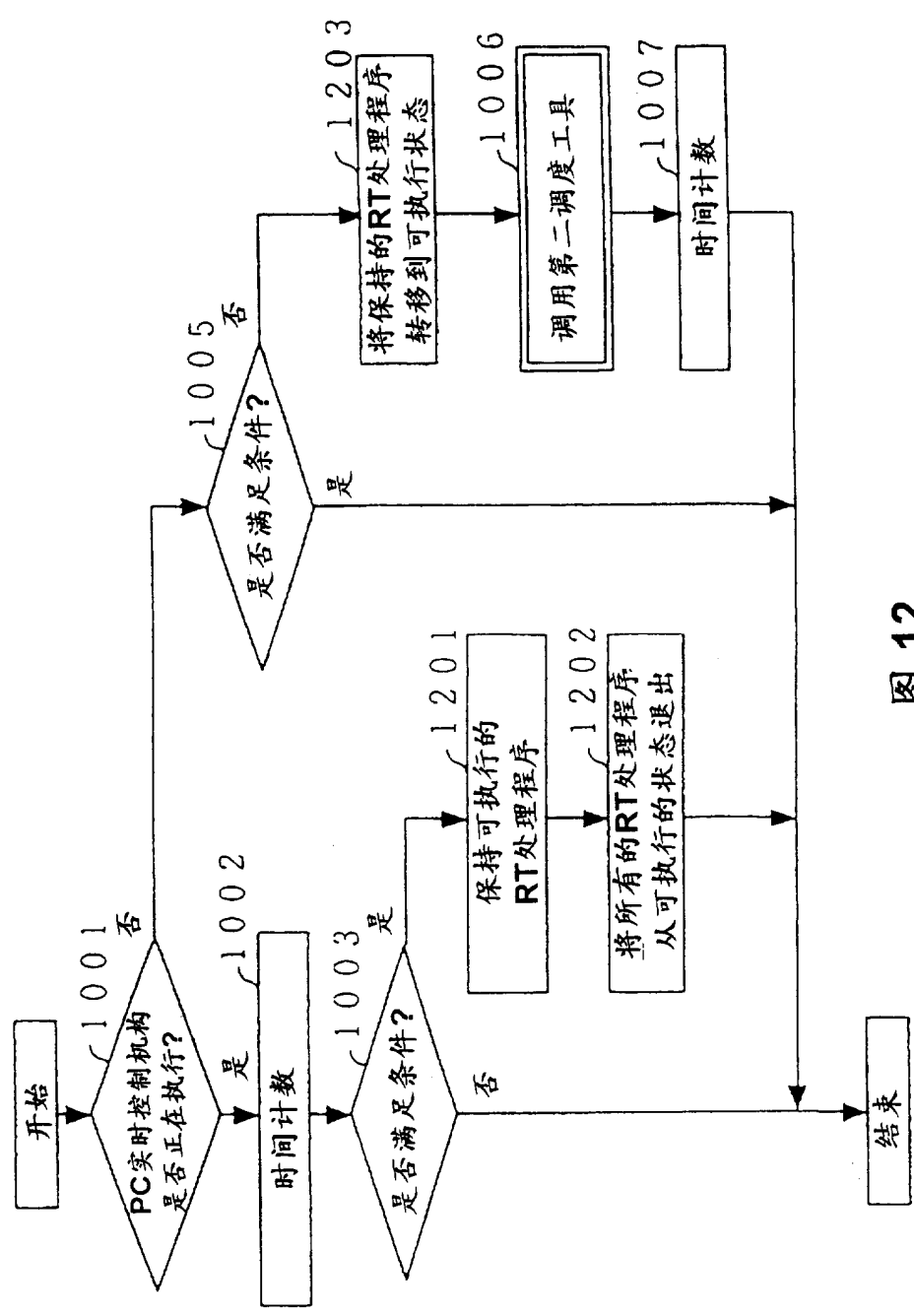


图 12

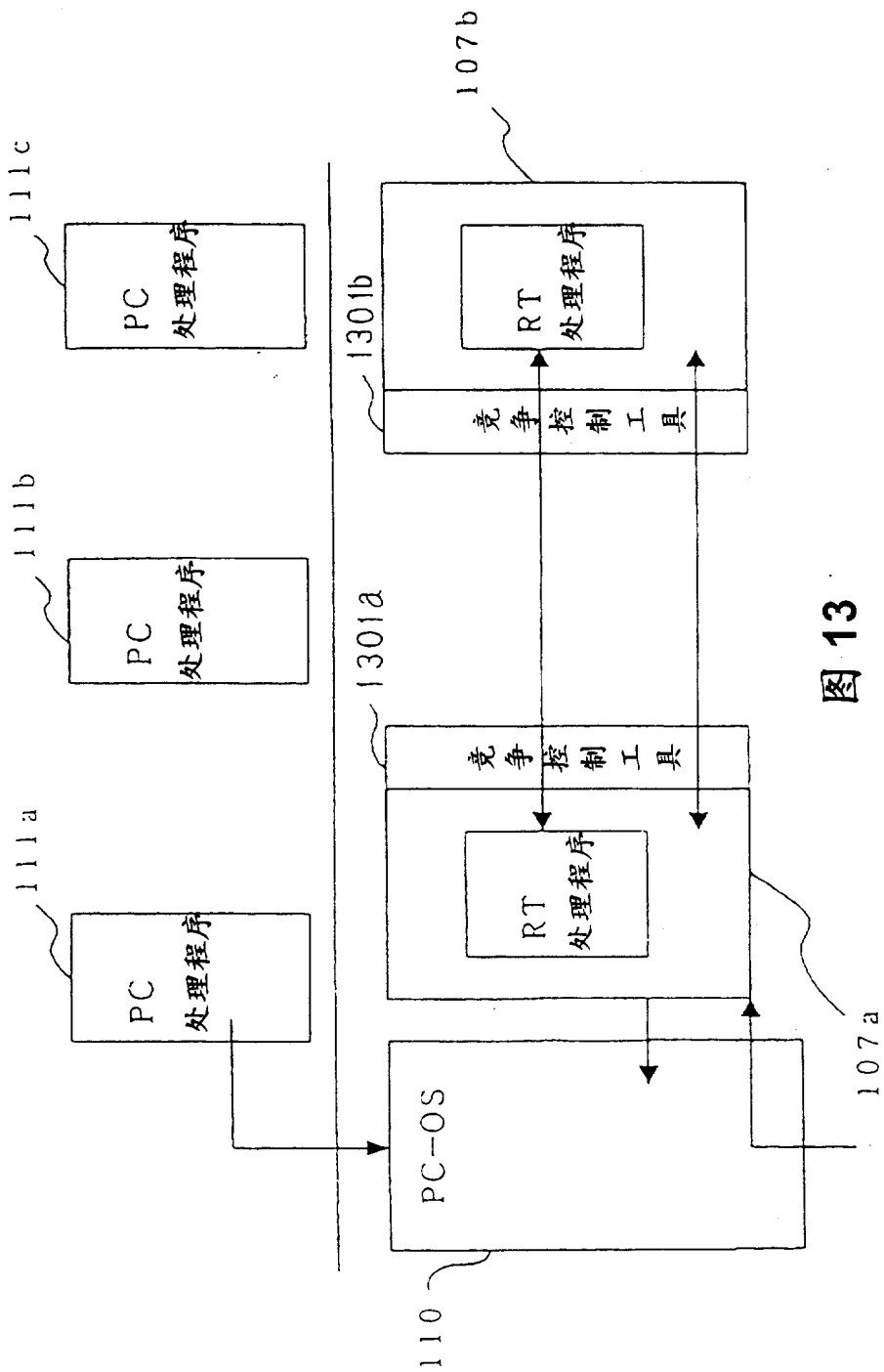


图 13

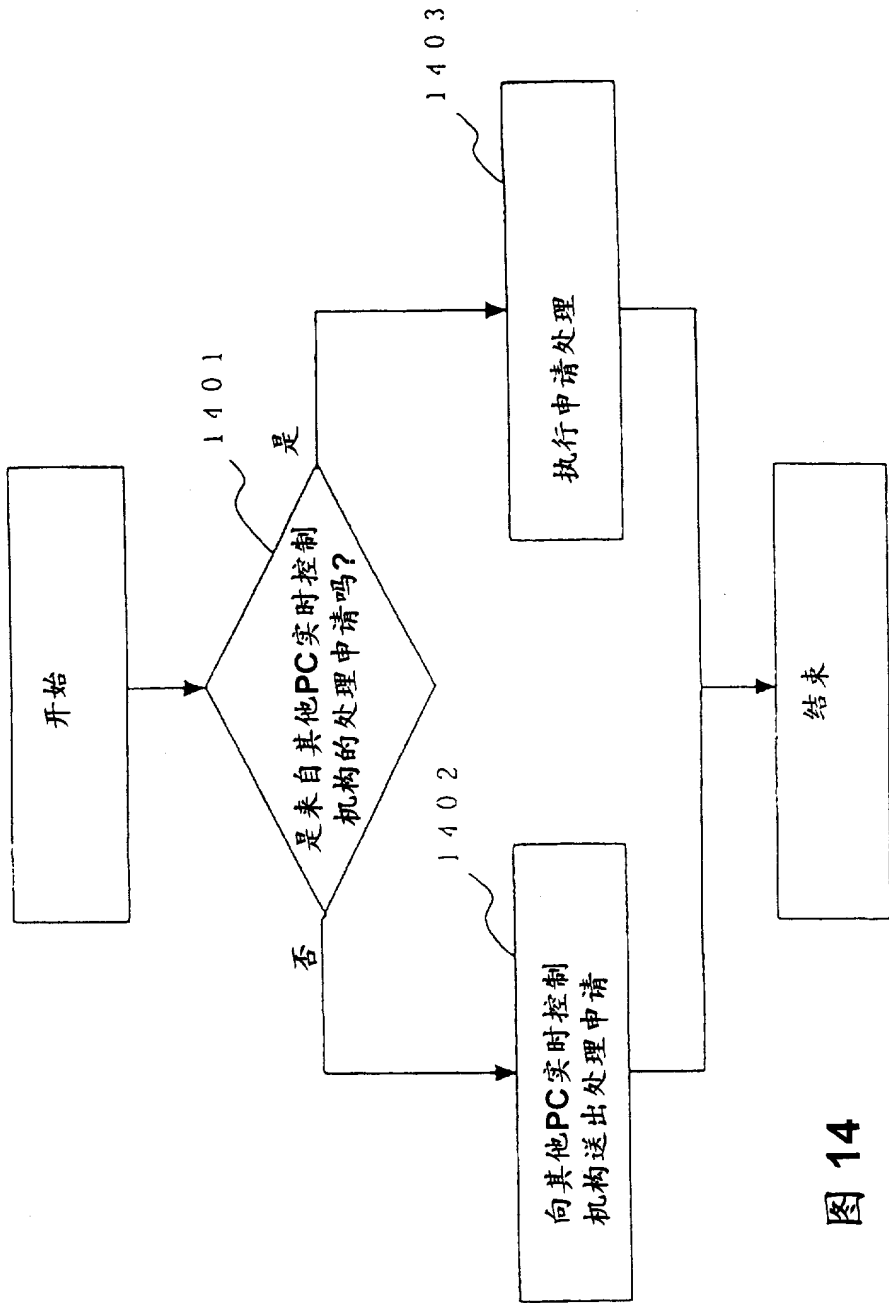


图 14

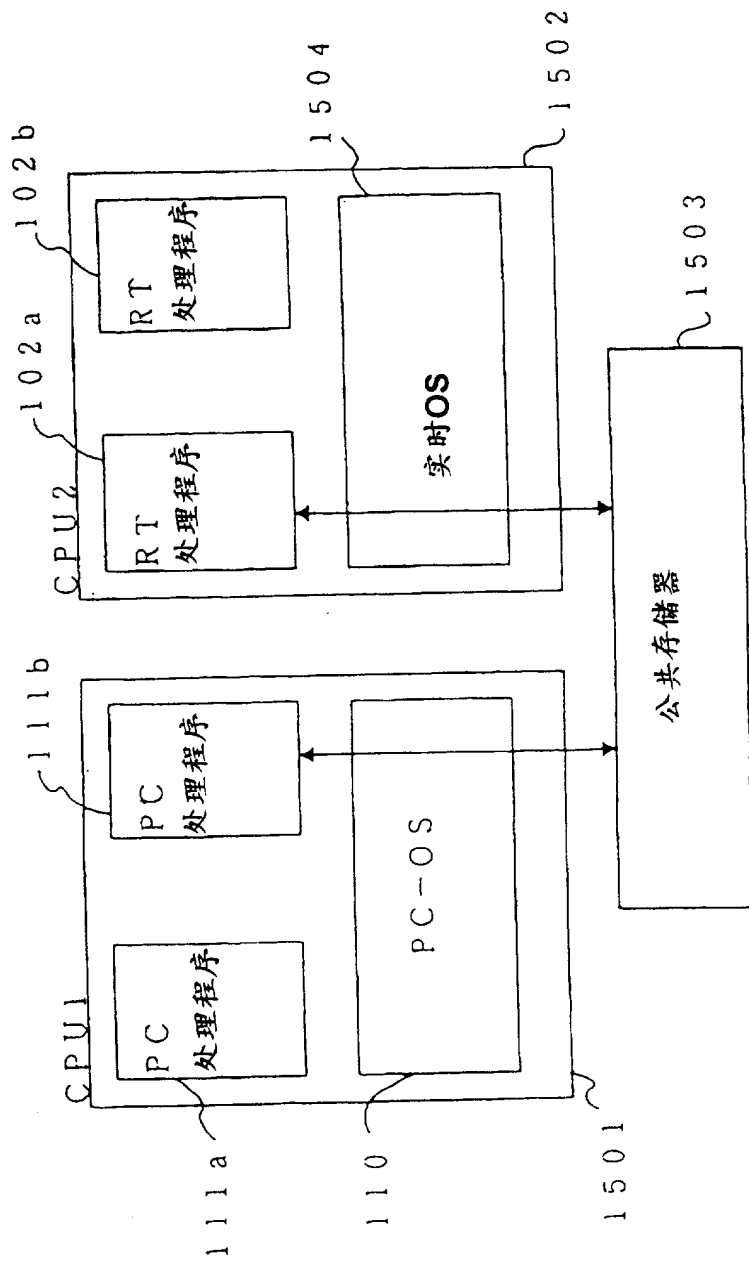


图 15

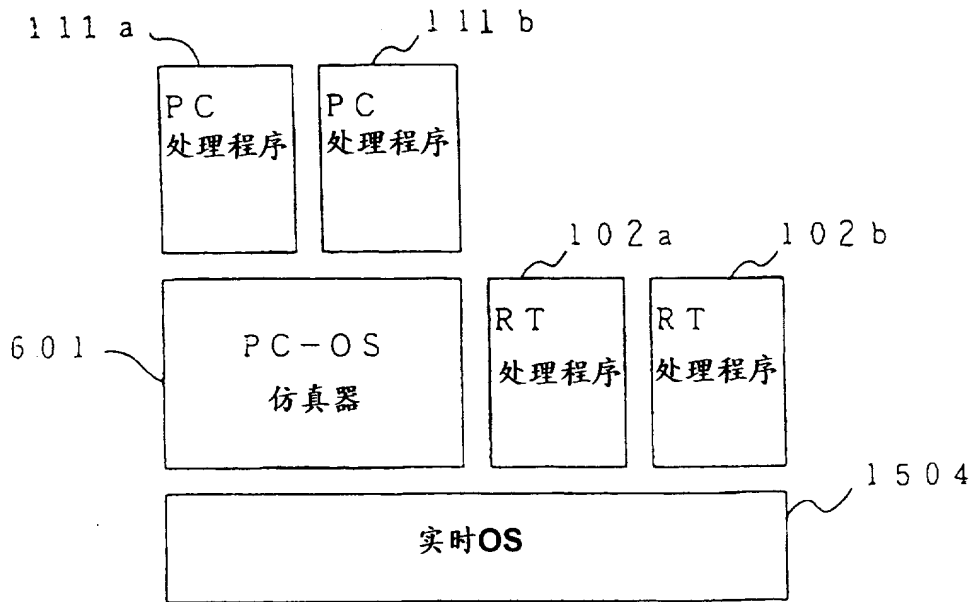


图 16