

[12] 发 明 专 利 说 明 书

[21] ZL 专利号 94119324.1

[45]授权公告日 2002 年 12 月 11 日

[11]授权公告号 CN 1096018C

[22]申请日 1994. 12. 13 [21]申请号 94119324.1

[30]优先权

[32]1993. 12. 15 [33]US [31]169921

[73]专利权人 国际商业机器公司

地址 美国纽约

[72]发明人 张佑彰 李实恭

审查员 李韵美

[74]专利代理机构 中国专利代理(香港)有限公司

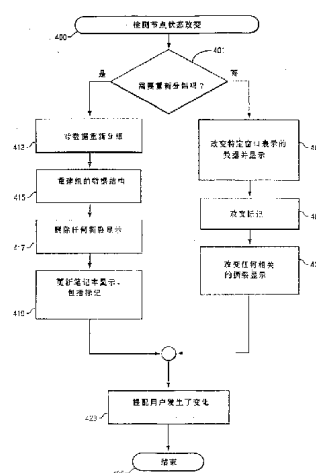
代理人 董江雄 马铁良

权利要求书 3 页 说明书 16 页 附图 14 页

[54]发明名称 作为笔记本的动态数据显示

[57]摘要

在显示器上显示动态数据的方法、系统和程序。动态数据作为许多组在笔记本中显示,每一组分配给笔记本中的一页。监测动态数据以检测动态数据状态的变化。根据检测到的变化,决定动态数据应根据状态变化重新分组。然后根据重新分组的动态数据显示笔记本。借助拖拉和放下操作,通过在笔记本以外附近的屏幕上复制窗口,可以建立一个子窗口。当根据用户的命令关闭笔记本时,对笔记本的关闭作出响应,自动关闭子窗口。



1. 一种在显示器上显示动态数据的方法，包括以下步骤：
将动态数据作为许多组显示，每一组分配给笔记本中的一页；
检测动态数据的状态变化；
根据状态变化确定动态数据应重新分组；以及
显示带有重新分组的动态数据的笔记本。
2. 权利要求1的方法，其特征还在于还包括根据属性类型和表示类型显示笔记本最上面的一页的步骤。
3. 权利要求1的方法，其特征还在于还包括根据状态变化显示笔记本最上面的一页的步骤。
4. 权利要求1的方法，其特征还在于动态数据代表计算机系统中的一组节点。
5. 一种在显示器上显示数据的方法，包括以下步骤：
将数据作为许多组显示，每一组分配给笔记本中的一页，最上面的一页具有一个窗口，被显示在笔记本的最上面；
通过复制窗口在笔记本以外建立一个子窗口。
6. 权利要求5的方法，其特征还在于通过拖拉和放下的操作建立子窗口。
7. 权利要求5的方法，其特征还在于进一步包括以下步骤：
根据用户命令关闭笔记本；以及
对关闭笔记本作出响应，自动关闭子窗口。
8. 一种在显示器上显示动态数据的方法，包括以下步骤：

将动态数据作为许多组显示，每一组分配给笔记本中的一页，每一页具有一个标记；

确定每一组动态数据的状态；以及

根据分配给那一页的组的状态显示该页的标记。

9. 权利要求8的方法，其特征在于还包括以下步骤：

检测至少第一组动态数据的状态变化；以及

根据状态的变化改变标记的显示，标记对应于第一组动态数据。

10. 权利要求8的方法，其特征在于动态数据代表计算机系统中的一组节点。

11. 权利要求9的方法，其特征在于还包括以下步骤：

根据状态变化确定动态数据应重新分组；以及

显示带有重新分组的动态数据的笔记本。

12. 权利要求8的方法，其特征在于还包括根据状态变化显示笔记本最上面的一页的步骤。

13. 一种在显示器上显示动态数据的系统，包括：

将动态数据作为许多组显示的装置，每一组分配给笔记本中的一页；

检测动态数据的状态变化的装置；

根据状态变化确定动态数据应重新分组的装置；以及

显示带有重新分组的动态数据的笔记本的装置。

14. 权利要求13的系统，其特征在于还包括根据属性类型和表示类型显示笔记本最上面的一页的装置。

15. 权利要求13的系统，其特征在于还包括根据状态变化显示笔记本最上面的一页的装置。

16. 权利要求13的系统，其特征在于动态数据代表计算机系统中的一组节点。

17. 一种在显示器上显示数据的系统，包括：

将数据作为许多组显示的装置，每一组分配给笔记本中的一页，最上面的一页具有一个窗口，被显示在笔记本的最上面；

通过复制窗口在笔记本以外建立一个子窗口的装置。

18. 权利要求17的系统，其特征在于通过拖拉和放下的操作建立子窗口。

19. 权利要求7的系统，其特征在于进一步包括：

根据用户命令关闭笔记本的装置；以及

对关闭笔记本作出响应，自动关闭子窗口的装置。

20. 一种在显示器上显示动态数据的系统，包括：

将动态数据作为许多组显示的装置，每一组分配给笔记本中的一页，每一页具有一个标记；

确定每一组动态数据的状态的装置；以及

根据分配给那一页的组的状态显示该页的标记的装置。

21. 权利要求20的系统，其特征在于还包括：

检测至少第一组动态数据的状态变化的装置；以及

根据状态的变化改变标记的显示的装置，标记对应于第一组动态数据。

22. 权利要求21的系统，其特征在于还包括：

根据状态变化确定动态数据应重新分组的装置；以及

显示带有重新分组的动态数据的笔记本的装置。

作为笔记本的动态数据显示

本发明总的来说涉及数据处理系统的用户界面。更具体地说，本发明涉及以笔记本的形式显示动态数据的问题。

已知的图形用户界面（GUI）是使用户能够控制计算机系统，并将系统运行的结果显示在显示器上。在图形用户界面中，应用程序和数据文件一般由多个图形目标如窗口或图符来代表。如果存在着大量的数据，特别是存在着如经常变化的状态信息那样的动态数据，那么要将所有的数据同时都显示在显示器上就非常困难。因此，可能不得不在许多窗口上顺序显示这些数据。例如，在设计和运行计算机网络时，出于各种理由要将网络分段，并且根据这种分段显示网络信息。根据通信模式，网络中的设备或“节点”被分成组或“群集”，因此在同一群集中的节点相互之间的通信要比与其它群集中的节点的通信更直接。也可以根据节点之间的传输媒体的不同、节点的位置或由于安全的原因来将节点分组。出于可管理性和可读性的考虑，组的规模不宜太大。于是在大网络中，很可能具有与如状态信息那样的动态数据有关的许多组，对用户来说，必须能够用图形用户界面的方法存取这些数据。

大家知道，当出现某种情况时，必须改变计算机显示器上的符号的外观，以便通知用户，对系统的状态引起警觉。节点组之间的网络事务处理的量或速率可以用这种方式进行监视。当群集中事务

处理的量或速率超过预定的界限时，代表那一组的符号就会提醒用户，调查该情况。然而，这一作法只允许有限的信息传递给用户，帮助他对系统进行调查。例如，对一般的系统来说，当符号表明一个群集中的事务处理量已经超过界限时，用户必须采取步骤，以便得到更多的关于那个群集的数据。通常这包括选择代表需要引起注意的群集的符号和改变计算机显示器，以便显示详细情况。这可能需要通过几个窗口的引导。如果存在大量的数据，就不可能同时显示所有的符号，因此用户便不能引起警觉。

对可管理的、适合于如网络状态信息那样的大量动态数据的GUI的需求与日聚增。由于分布式处理方式和并行处理方式的需要，寻求更大、更复杂的网络已成为一种趋势。此外，由于单个网络设备的功能变得更强、价格更便宜，所以用户倾向于更频繁地增添网络和重新配置网络。在这种情况下，监视计算机网络的状态将更重要、但也更困难。

图形用户界面中存在大量动态数据的其它例子还包括库存商品价格，它一直随着市场的波动而变化，以及试验过程例如航天飞机发射过程中的科学监测。

笔记本是一种图形控制系统，已经包括在IBM的OS/2操作系统中，它允许在GUI中简明地表示大量的信息。笔记本一般表示成一组窗口，只有最上面的窗口可以看到。从其它屏面页伸出一系列标记。然而，笔记本一般不表示动态数据，而是表示静态数据。此外，笔记本有许多限制，这使得监视如网络状态那样的动态数据非常困难。在今天的笔记本中，每一页与一组不变的数据相关联。没有办法来表示经常重新配置的数据组。除非用户正在看具体的一页，而

其中状态信息又出现在最上面一页的情况，即使笔记本适合于显示动态数据，也没有办法提醒用户状态已经发生了变化。与隐藏页相关的组的状态变化是无法看到的。

偶尔还需要比较不同组的动态数据。从逻辑上讲，笔记本的每一页将表示一组数据。而用笔记本一次只能看到一页。这样，当比较不同的组时，笔记本的一页不能提供足够的信息。用户不得不在两页之间来回翻动。这可能促使设计者把更多的信息放在一页上以减少总页数，不过这样做违背了使用笔记本的初衷，笔记本是要把数据分成组，并且每一页中给出一组数据。同时，用笔记本表示静态数据在一定程度上也存在这些问题，当出现动态数据，而用户不能信赖第一页上的数据在以后观看第二页时仍保持不变，这时问题将变得更为严重。

目前设计的笔记本动态连接库（DLL）的构造不支持动态信息，也不能一次看一页以上的数据。一旦开发者已经写了笔记本DLL，它也不支持对笔记本中的数据重新分组，而对某些应用程序来说这可能是需要的。

因此，本发明的一个目的是提供可管理形式的动态数据，同时还允许对数据进行全部存取。

本发明的另一个目的是提供作为动态数据的状态信息。

本发明的第三个目的是根据数据的状态变化对动态数据重新分组。

本发明的第四个目的是对一组动态数据的状态进行总结，并允许存取关于任何特定的动态数据组的详细信息。

本发明的第五个目的是在显示器上同时显示几组动态数据。

本发明的第六个目的是监视与计算机网络相连的节点的状态。

这些目的等是通过在显示器上显示动态数据的方法、系统和程序实现的。动态数据是作为许多组在笔记本中显示的，每一组被分配给笔记本中的一页。在一个最佳实施例中，监测动态数据以检测动态数据状态的变化。根据检测到的变化，决定动态数据应根据状态变化重新分组。然后根据重新分组的动态数据显示笔记本。根据属性类型和表示类型显示笔记本的每一页。本发明具有特殊的应用价值，可以将大量的动态数据，诸如一组节点的状态，在计算机系统中予以显示。

在本发明的第二实施例中，笔记本中的最上面的一页具有在其中显示的一个窗口。借助拖拉和放下操作，复制该窗口到笔记本以外的邻近显示区域中，可以建立一个子窗口。当根据用户的命令关闭笔记本时，对笔记本的关闭作出响应，自动关闭子窗口。

在本发明的第三实施例中，笔记本中的每一页具有一个标记，根据分配给那一页的组的状态显示该标记。根据检测的动态数据组中一组的状态变化，对应于那一组的标记的显示则根据状态变化而变化。

通过以下结合附图对本发明的最佳实施例所作的详细描述，本发明将变得更易于理解，附图中：

图1表示由系统单元、键盘、鼠标器和显示器组成的计算机。

图2是图1所示计算机的结构框图。

图3表示一个计算机网络，其节点分布于不同大小的组中。

图4表示监视图3网络的本发明的笔记本。

图5A和5B表示本发明的笔记本，其中状态变化已经引起了对网

络中的节点重新分组。

图6A和6B表示网络最上面一页上的一组的状态变化，其中标记上的符号也变化了。

图7表示借助拖拉和放下操作，建立一个笔记本以外的子窗口。

图8表示根据本发明的笔记本的一种可能的数据结构。

图9是建立笔记本的过程的流程图。

图10是检测到节点中存在变化的流程图。

图11表示选择页标记的过程。

图12A—12C表示建立笔记本以外的一个子窗口的过程。

图13表示关闭笔记本的过程。

本发明可以在具有不同的操作系统的多台计算机上运行。网络中的计算机例如可以是个人计算机、小型计算机、大型计算机或在其它计算机的分布式网络中运行的计算机。虽然具体选择计算机仅仅受到所需磁盘和磁盘存储量的限制，但是IBM PS/2 (TM) 系列计算机都可以用于本发明。为了获得有关IBM PS/2 (TM) 系列计算机的更多的信息，读者可参阅“Technical Reference Manual Personal System/2 Model 50,60 system”，IBM公司，分册号68X 2224，序列号S68X- 2224，以及“Technical Reference Manual Personal System/2 (Model 80)”，IBM公司，分册号68X 2256，序列号S68X-2254。IBM PS/2个人计算机可以运行的一种操作系统是IBM的OS/2 2.0 (TM)。为了获得有关IBM OS/2 2.0 操作系统的更多信息，读者可参阅“OS/2 2.0 Technical Library Programming Guide”第1、2、3卷，2.00版，序列号10G6261、10G6495、10G6494。

另外，计算机系统也可以是IBM RISC System/6000 (TM) 系列的计算机，它们在AIX (TM) 操作系统下运行。不同型号的RISC System/6000 (TM) 在IBM公司的许多出版物中都有说明，例如，“RISC System / 6000, 7073 and 7016 POWERstation and POWERserver Hardware Technical”参考资料，序列号SA23-2644-00。在“General Concepts and Procedure-- AIX Version 3 for RISC System/6000”序列号SC23-2202-00以及IBM公司的其它出版物中都说明了AIX操作系统。

在图1中，计算机10包括系统单元11、键盘12、鼠标器13 和显示器14。显示装置14的屏幕16用来显示图形用户界面 (GUI)。图形用户界面由操作系统支持，它使用户能通过指示和滑动的输入方法执行用户命令或进行选择，即将鼠标指针15移动到屏幕16的特定位置上代表数据目标的图符处，并按动鼠标器的一个按钮。

图2表示图1所示的个人计算机各部分的框图。系统单元11包括一根系统总线或多根系统总线21，各部分与系统总线相连，并通过系统总线各部分之间实现通信。微处理器22与系统总线21相连，并由也与系统总线21相连的只读存储器 (ROM) 23 和随机存取存储器 (RAM) 24支持。IBM多媒体PS/2 系列计算机中的微处理器是包括386或486微处理器的Intel微处理器系列中的一种。然而，在具体的计算机中还可以采用下面所述的其它微处理器，但又不限于这些，比如：Motorola的微处理器系列如68000、68020或68030 微处理器和由IBM、Hewlett Packard、Sun、Intel、Motorola等制造的各种缩减指令集计算机 (RISC) 微处理器。

ROM 23其中包括基本输入/输出系统 (BIOS)，它控制基本硬

件操作,例如交互操作,以及磁盘驱动器和键盘。RAM 24是主存储器,其中装入操作系统和应用程序。存储器管理芯片25与系统总线21相连,控制直接存储器存取操作,该操作包括在RAM 24和硬盘驱动器26、软盘驱动器27之间传递数据。CD ROM 32也与系统总线21相连,用于存储大量的数据,例如多媒体程序或大的数据库。

各种I/O控制器也与系统总线21相连,它们是键盘控制器28、鼠标控制器29、视频控制器30和音频控制器31。如同所希望的那样,键盘控制器28为键盘12提供硬件接口,鼠标控制器29为鼠标13提供硬件接口,视频控制器30是显示器14的硬件接口,音频控制器31是扬声器15a和15b的硬件接口。扬声器15a和15b可以用来向用户提供声音。诸如令牌环形适配器(Token Ring Adapter)的I/O控制器40使得能够通过网络46与其它类似配置的数据处理系统联系。

本发明的一种最佳实施方案是以驻留在RAM 24中的代码模块结构的一组指令实现的。在计算机系统要用到这一组指令之前,它们可以存储在另外的计算机存储器中,例如硬盘驱动器26、最终用于CD ROM 32的光盘或最终用于软盘驱动器27的软盘中。如图所示,操作系统50和表示管理程序52驻留在RAM 24中。在本例中,本发明是通过附加在操作系统上的模块实现的。另外,图形用户界面可以包括在独立的应用程序54中,例如监视程序中。监视程序54可以监视网络中的所有节点,或仅仅监视作为并行数据库56一部分的节点的子集。

图3表示带有各种节点的计算机网络。网络100具有传输媒体101,各种设备可以通过传输媒体101相互联系。节点102a、102b等可以包括以下这些设备,如工作站103、文件服务器104、单元控制

器105和桥接器106等,所有这些设备可以分成大小不同的组,与网络100相连。

在图示的网络中,节点被分为八个组107a至107h。根据本发明的提法,组的大小被定义为组中的节点数 n 。从图中可以看出,本例的组的大小是很不相同的。

目前,IBM OS/2操作系统 2.1 版提供了一种控制设备,叫做“笔记本”,当显示许多屏面或页以供观看时,笔记本是非常有用的。如图4所示,根据本发明,笔记本110表示多组动态数据,例如如图3所示的网络的状态信息。最上面的一页111包括窗口113,它又包括矩阵115,矩阵115是一个表示计算机网络中的一组节点组1的CPU利用率的二维表。矩阵中的每个单元代表一个节点。第一列代表节点0—4,第二列代表节点5—9,第三列代表节点10—14,等等。单元中的数字代表CPU利用率的百分比。画阴影的单元代表网络中未被监视的节点,因为它们不是并行数据库的一部分。通过看图用户可以懂得,编号较大的节点在处理过程中恰巧也具有较高的CPU利用百分比,这意味着通过组1中的节点的装入是不平衡的。画阴影的单元代表在网络中、但不在并行数据库中的节点,所以监视程序不监视它们的状况。标记区121从页111突出,包含概括与那一页有关的数据状态的信息。其它的标记123、125、127、129和131与隐藏在最上面的一页111下的其它页关联。这些标记为用户提供了一种工具,使他能够通过将鼠标器指针移动到某一标记而跳到另一页即另一组。这些标记123、125、127、129和131中的每一个都有一个符号或字母数字信息124、126、128、130和132,它们代表包含在那一页上的动态数据的状态。

标记可以用符号、颜色表示不同的意义，例如标记可以具有明亮的红色火焰符号，以表示某个特定的节点组存在最紧急的情况。标记可以闪烁，以便出现紧急情况时吸引用户的注意力，或者改变页上的图符以反映页上的节点或动态数据的状态。如同下面将要讨论的那样，对窗口113中的信息的表示可以被指定为一种表示类型，例如矩阵、条形图，以及一种属性类型，用以描述包含于该表示中的数据的数据的类型。

图5A和5B表示本发明的笔记本，其中状态的变化已经引起了对网络中的节点重新分组。在这两幅图中，分组的标准是根据CPU的使用率，组1中CPU的使用率是0—10%，组2中CPU的使用率是10—30%，组3中CPU的使用率是30—50%，组4中CPU的使用率是50—75%，组5中CPU的使用率是75—85%，组6是最高的组，其CPU的使用率是85—100%。标记显示的是组中CPU的平均使用率。如图5A所示，最上面的一页151表示组6，它具有最高的使用率。标记153表示平均使用率为90%。窗口155将每个节点分别显示为长条152、154、156、158和160，每一长条分别表示特定节点的CPU的使用率。图中还有标记157、159、161、163和165，它们分别表示组1—5的CPU的平均使用率。

如图5B所示，不同节点CPU的使用率的关系已经改变了，因此对笔记本各页上的动态数据重新分组。节点S10、UXY和WM退出最高使用率组，而节点U1和A2进入最高使用率组。因此，当窗口155在笔记本页151上被刷新时，不仅显示节点S1和MT5的长条152、156，而且也显示新的节点U1和A2的新的长条162和164。还要注意，标记153、157、159、161、163和165也反映每一组节点的改变的CPU使

用率。

图6A和图6B表示在笔记本的最上面一页中的节点组的状态的改变情况，笔记本强制产生一个符号变化显示于相应的标记当中。在本例中，分组的标准是按照所处位置。笔记本200的最上面的一页201表示处于Austin的节点，窗口表示在每个节点进行事务处理的次数。标记205表示事务处理的平均数，由于它处于允许的限度内，因此以正常方式显示。另一方面，当事务处理的次数已经超过某一临界点时，标记207以相反的影像闪烁，以便唤起管理者的注意。另外，在标记中可以用绿色表示允许，用黄色表示边界，而用红色表示危险。

如图6B所示，在过去的一小时，由节点Lake处理的事务处理次数（如长条209所示）已经从略低于100次事务处理/秒跳到550次事务处理/秒。这使得与这一页有关的标记205变为以相反的影像闪烁，表示可能需要管理者引起注意。同时，第二标记207以正常方式显示，表示与隐藏的页相关的组中事务处理的次数已经下降到允许的程度。

本发明的笔记本的另一个特征是可以笔记本以外建立一个笔记本页的子窗口，因此可以同时比较两页。在图7中，已经建立了图4所示的笔记本110的最上面的一页111的子窗口211。这可以通过拖拉和放下操作实现。用户可将鼠标器指针放在标题条135上，按下鼠标器按钮并保持这一动作，拖拉窗口的图符图象到第二位置。在第二位置215释放鼠标器按钮。一旦释放鼠标器按钮，新的子窗口211便建立起来了。

可以存储在RAM中的GUI的一种数据结构示于图8中。图8中的

DLL数据结构表示本发明的设计基础。本发明建立的DLL将与笔记本DLL一起，用于产生本例中的状态监视程序的笔记本GUI。状态监视程序将用笔记本DLL建立笔记本，包括所需要的控制，例如标记、滚动条，并将本发明的DLL用于动态改变标记，建立被选择页的独立的窗口，以及本发明的其它特征。本领域的一般技术人员将认识到，也可以采用其它数据结构。数据结构300包括多个数据结构301、303，笔记本中的每一页用一个。每一页数据结构301、303包括信息域，如组号302、属性类型304、表示类型305，撕裂窗口306、如果有的话还有节点表307和下一组308的指针。属性类型描述包括在这一组表示中的属性的类型。属性类型的例子可以包括基于两个节点之间的通信量的连接数、每秒事务处理中的工作负荷、总的进出通信量或与另一节点的连接性。表示类型305描述表示的类型，例如矩阵、图形显示、文本、动画、图符，等等，如果这一组是笔记本中最上面的一页的话，这些都将显示出来。

撕裂域306是一个指向第一窗口文件309的指针，如果有的话，窗口文件309可以是一个笔记本的子窗口。至少，子窗口数据结构309将包括窗口标识域310和指向下一个子窗口数据域312（如果有的话）的指针311。如果没有更多的窗口，指针311将被设置。

节点表307是这一页中所有节点的表，也是指向第一个节点数据结构319的指针。节点数据结构319包含信息，如节点数320和指向下一个节点数据结构323的指针321。同样，在下一个数据结构323中，包括节点数域324和指向节点表325中的下一个节点的指针，如果这是表中的最后的节点，那么下一个域325中的指针将设为零。

如上所述，属性表将由以下几项组成，如连接性、通信量和工

作负荷。节点号也与属性数据结构337关联，属性数据结构337包括属性表339上的信息。属性数据表339由多个数据结构341、345组成，它们包括表示属性的域342、346，表示属性值343、347的域，以及指向表344、348中的下一个属性的指针。

图9是表示图形用户界面如何初始化的流程图。过程开始于步骤350，进入步骤351，进行测试，确定图8所示的分组数据结构是否建立。如果没有建立，在步骤353建立分组数据结构。根据节点分组所依据的属性建立分组数据结构。例如，节点可以基于通信量分组。状态监视的设计者将决定组中允许的最大的节点数以便于表示。计算每个节点的当前的通信量，具有最大通信量的节点将放在最高的组中，一直这样做下去，直到所有的节点都分配给一组。

如果已经建立了分组数据结构，那么在步骤361访问这一数据结构，以为将来处理。在步骤363，根据数据结构中的组数确定笔记本表示所需的页数。在步骤364，组1被设为当前组。在步骤365检索组1的数据，该数据是基于这一特定的属性类型和表示类型进行表示所必需的。在步骤367，建立当前组即组1的窗口显示。在步骤369，进行测试，确定是否还有更多的组要建立窗口显示。如果存在更多的组，那么在步骤371设定下一组为当前组，并且过程返回步骤365，在那里过程等待用户进一步输入命令。

如果动态数据的状态发生变化，例如某些节点的状态改变了，那么遵循图10所示的过程。在步骤400，检测到节点状态改变。在步骤401进行测试，确定状态改变是否需要数据重新分组。如果不需要，那么改变最上面的窗口或最上面的一页表示的数据，并且刷新显示。接下来在步骤405，改变标记以反映动态数据中的变化。

这可以通过参考描述标记是如何显示的数据并给出动态数据的具体范围来实现。接下来在步骤407, 根据表示数据改变子窗口以反映新的动态数据。

如果状态的改变需要对动态数据重新分组, 那么在步骤413 对数据重新分组。在步骤415, 根据重新分组的动态数据和动态数据状态的改变, 重建组的数据结构。接下来在步骤417, 删除任何子窗口。这是假定由于重新分组, 用户将对笔记本中的不同页感兴趣。在步骤419, 改变笔记本显示即最上面的一页和标记, 以反映重新分组的数据。在步骤423, 通过嘟嘟的音频信号或上托信息框, 提醒用户发生了变化。

笔记本的一个特征是用户可以通过选择与页有关的标记迅速地从某一页变到另一页。这一过程如图11所示。在步骤450, 选择页标记。在步骤451进行测试, 确定所选择的标记是否与当前显示的页有关。如果是的话, 在步骤453当前显示的页用新的属性值更新。如果所选择的页标记与当前显示的页无关, 那么在步骤455 检索与所选择的标记有关的分组数据结构中的数据。然后在步骤459, 根据属性类型和表示类型在笔记本的最上面的一页显示该组动态数据。这一过程在步骤461结束。

本发明的另一个特征是不同于传统的笔记本控制, 本发明可以从笔记本内通过拖拉和放下操作选择一个窗口, 从而复制该窗口或在笔记本以外建立一个子窗口。如果用户需要同时观看多个窗口, 以便比较各组动态数据的状态, 例如网络中的各组节点, 那么此时将是非常有用的。拖拉和放下操作之后, 原始的窗口依然存在于笔记本中。

在笔记本以外建立子窗口的过程示于图12A—12C。过程开始于步骤500，系统检测到鼠标器的拖拉动作。在步骤501，进行检测，确定鼠标器是否在显示页的标题条或代表该页的图符上按动鼠标器，按动鼠标器便启动了鼠标器的拖拉动作。如果在步骤503 检测的结果是肯定的，那么在步骤505，Page-Drag变量设为“on”，并且Group-no变量设为被拖拉窗口所表示的组的编号。如果不在标题条或该页的图符上按动鼠标器，那么在步骤507进行其它的操作，在步骤509系统返回监视其它事件。

在图12B的步骤511，检测到鼠标器释放。在步骤513，检测Page-Drag变量，如果不为“on”，那么在步骤515进行其它操作。然而，如果Page-Drag变量为“on”，那么在步骤517进行测试，确定放下的位置是否在笔记本以外。如果放下的位置不在笔记本以外，那么在步骤521不采取行动，在步骤523过程返回监视其它事件。

然而，如果放下的位置在笔记本以外，那么在图12C的步骤525检查组的组数据结构，具有该组号的页已经被拖拉出窗口。在步骤527，在放下位置的附近建立一个窗口。这需要为所建窗口建立一个窗口标识和根据该窗口标识为撕裂链表建立一个入口。在步骤529，进行测试，确定撕裂域是否具有“空”值，如果不是，那么在步骤531建立的链入口与撕裂的链表的末尾相连。如果撕裂的域为“空”，那么在步骤533 将指向该入口的指针放在撕裂的域中。过程在步骤535返回。

由于新建立的笔记本以外的窗口定义为笔记本的子窗口，所以关闭笔记本也将自动地关闭所有的这种子窗口。在本发明的笔记本界面中，笔记本允许它的页被复制，并且被复制的窗口仍由笔记本

管理。图13表示关闭笔记本的过程。在步骤550，通过选择系统菜单中的关闭图符或从动作条的文件下拉菜单中选择“退出”动作，关闭笔记本的过程开始。在步骤551，检查每一组的数据结构，以便确定是否存在与笔记本相关的任何子窗口。如果存在的话，则将它们删除，因此当笔记本重新打开时这些子窗口不会自动出现。在步骤553进行检测，确定撕裂的域是否设为值“空”。如果不是，则在步骤555，根据撕裂表中的窗口标识从组数据结构中删除从笔记本建立的每个子窗口。删除撕裂链表，并且撕裂值复位为“空”。最后，在步骤557关闭笔记本本身。

笔记本界面适合于组织任何动态数据组。如以上实施例所示，虽然本发明可以用于表示任何种类的动态数据组，但它特别适合于组织大规模的节点服务器系统的状态数据等。状态监视器以组为单位显示状态数据。根据用户说明的群集规则，节点被分成组。根据该组的表示类型和属性类型，在窗口中显示组中的节点的状态数据。在笔记本中组织窗口，每个窗口一页。通过采用不同的颜色或形状，每一页的标记可以表示一组中的一个节点的总的状态。例如，红色的标记可以表示大部分节点处于危险状态。根据标记表示的信息，用户可以决定选择哪一页。此外，当笔记本中的所有的标记都一起出现在屏幕上时，这就提供了一种简单的方法来比较各组的状态。根据标记揭示的信息，可以选择一页做调查。于是，标记提供了组的基本的状态总结，并且还允许用户反复查看任何特定的组，以便获得进一步的信息。

虽然以上针对具体的实施例描述了本发明，但是本领域的一般技术人员应懂得，在不偏离本发明的精神和范围的前提下可以做各

种改进。这些实施例只是为了说明而举的例子，并不对本发明的范围进行限制。

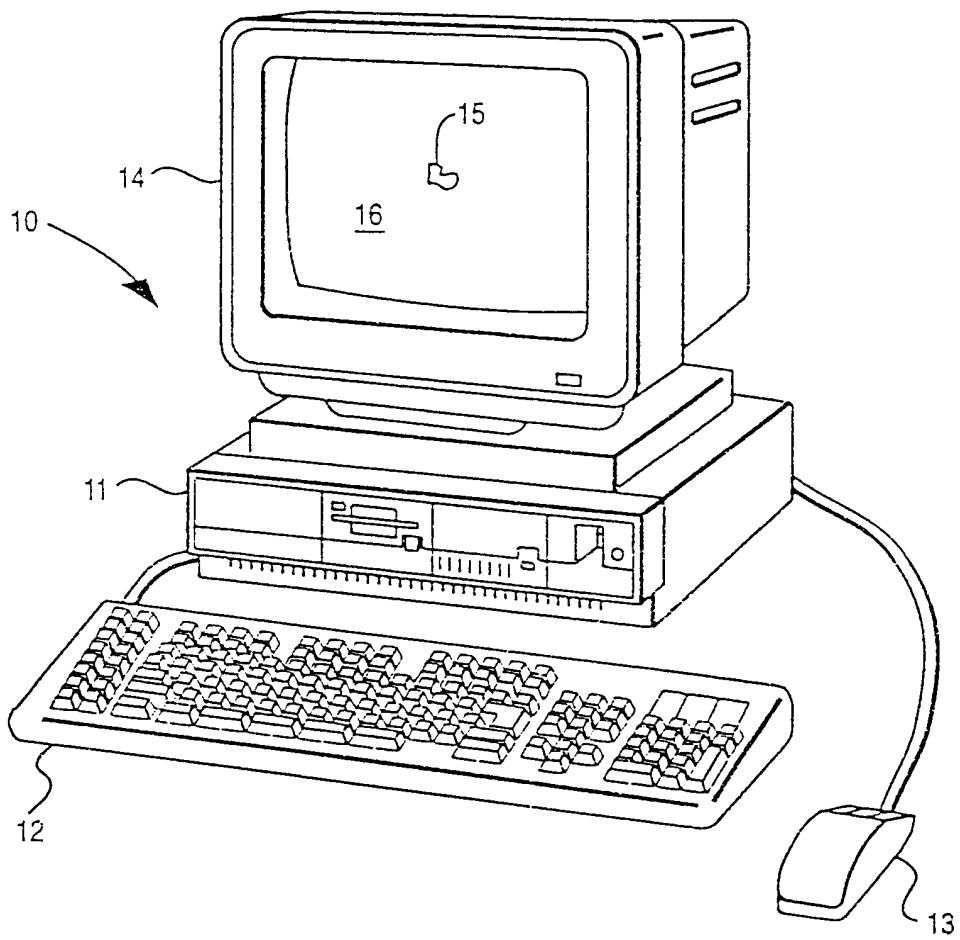


图 1

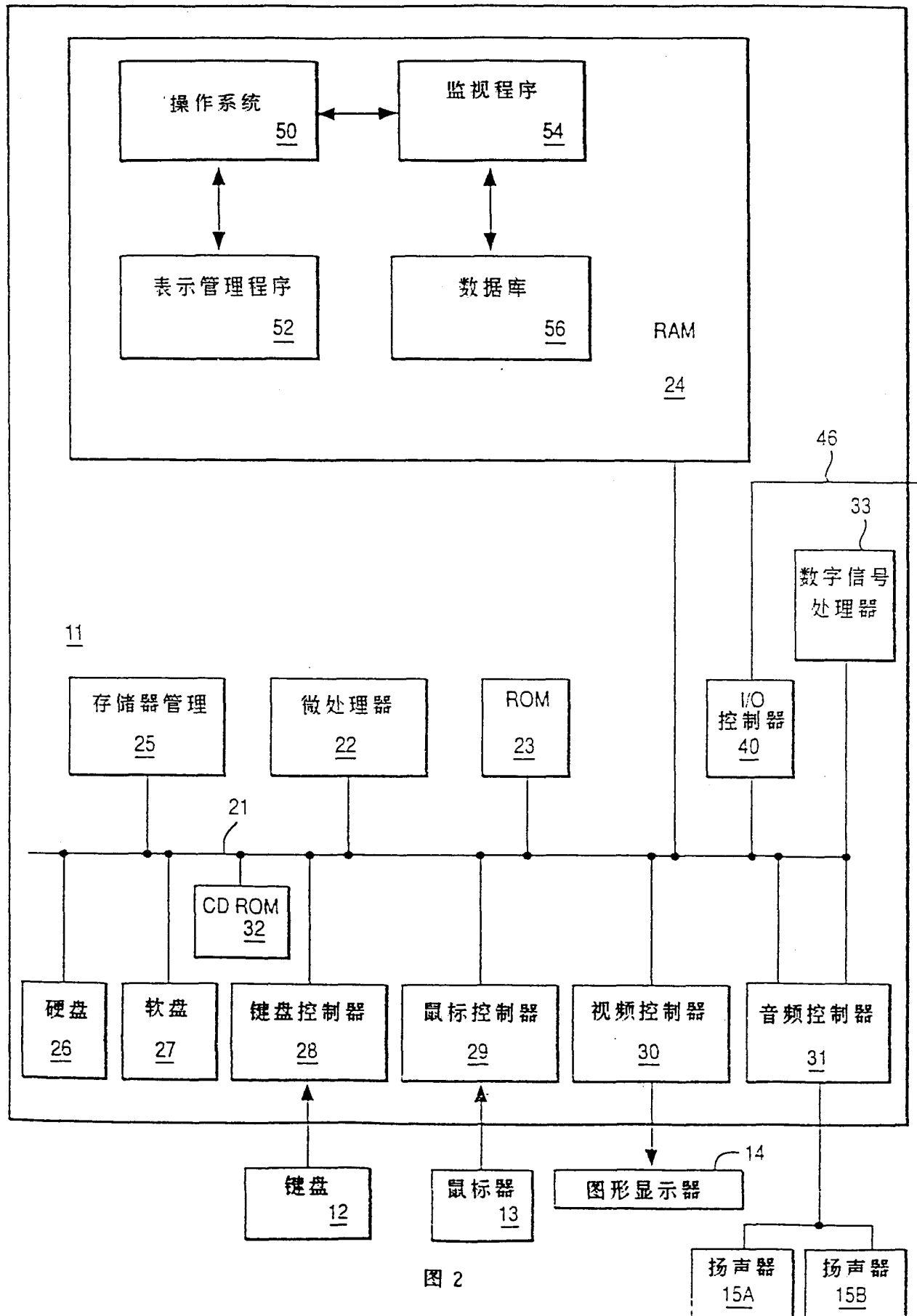
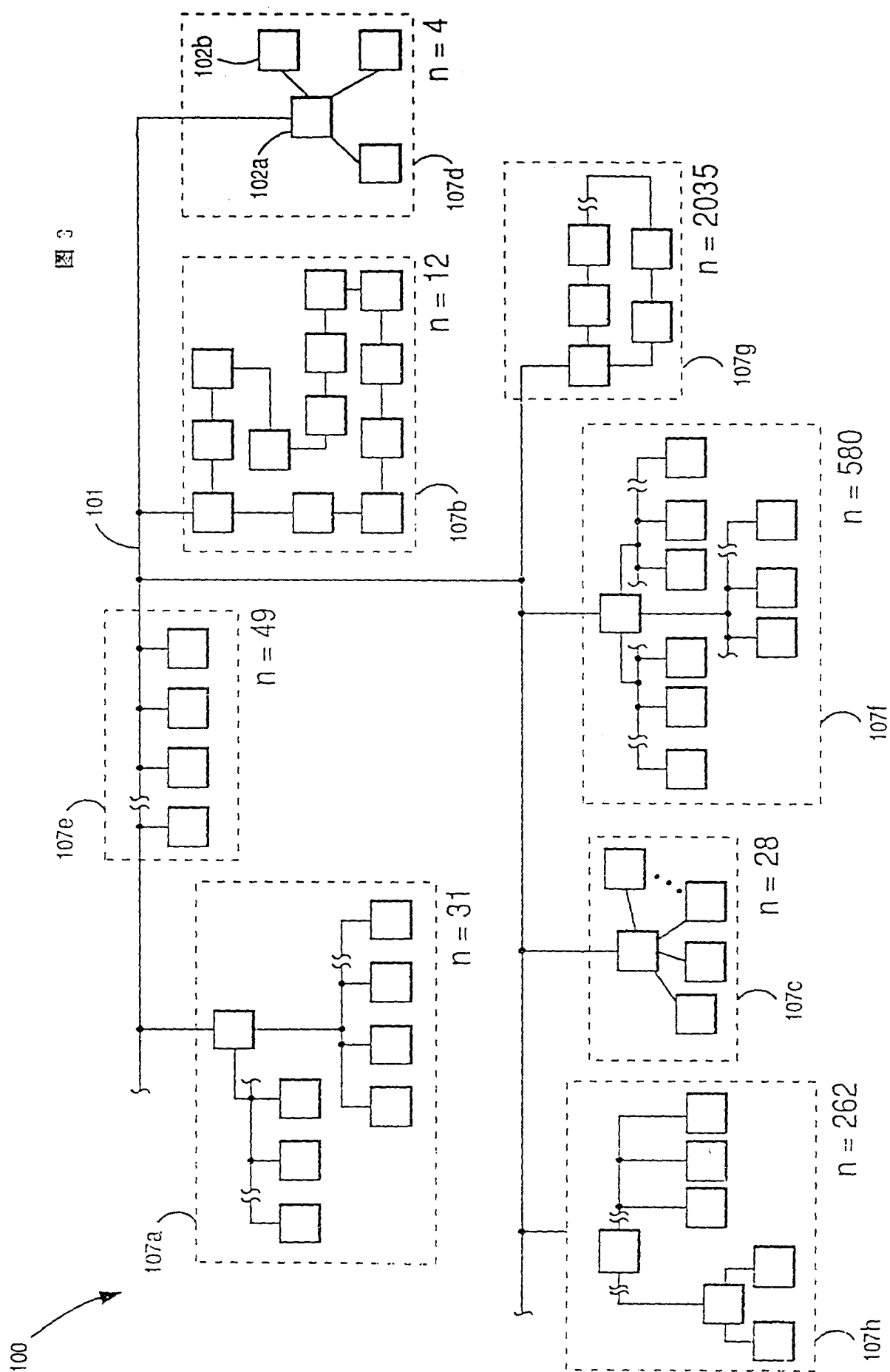


图 2

67
[Image]



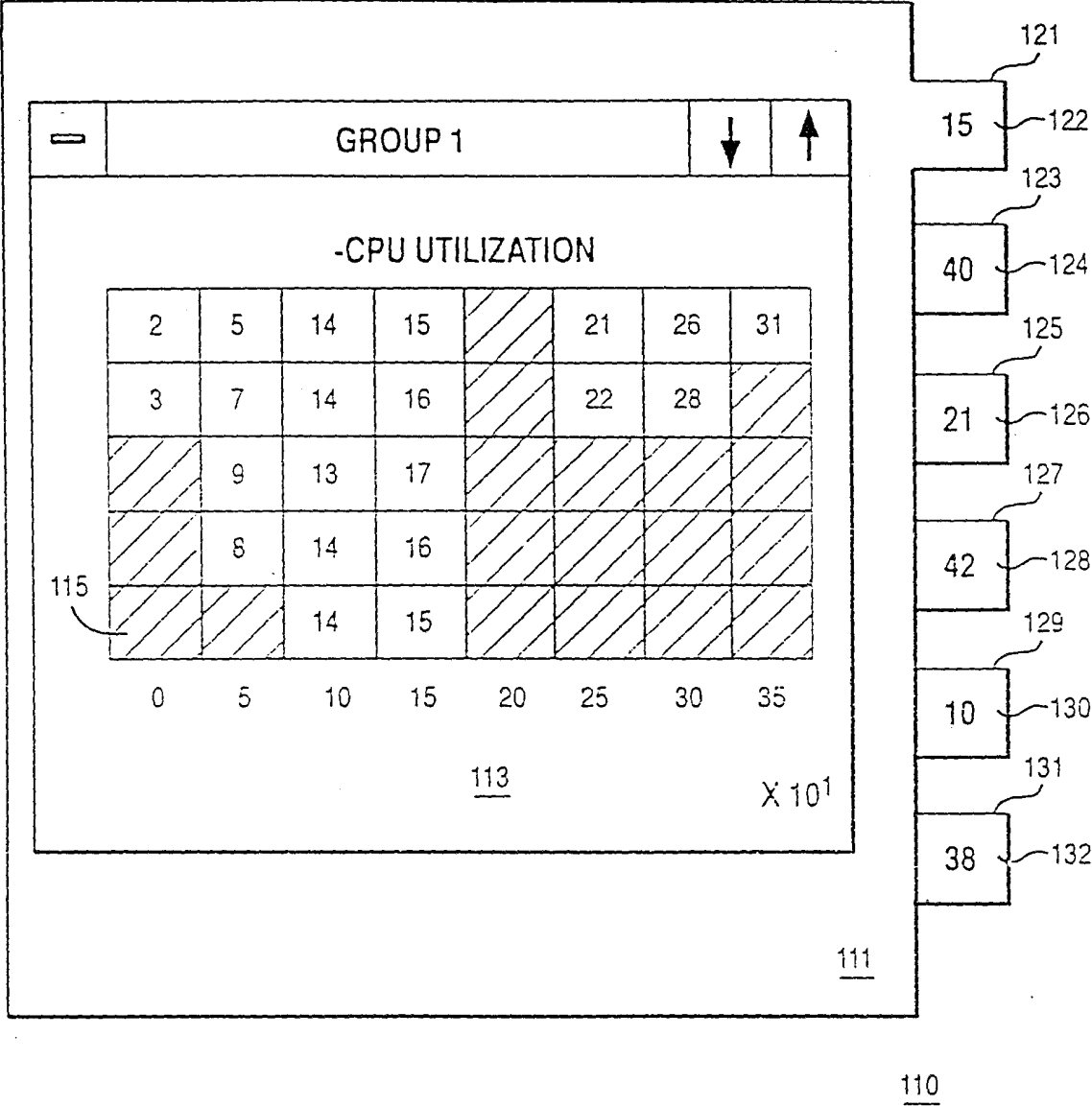


图 4

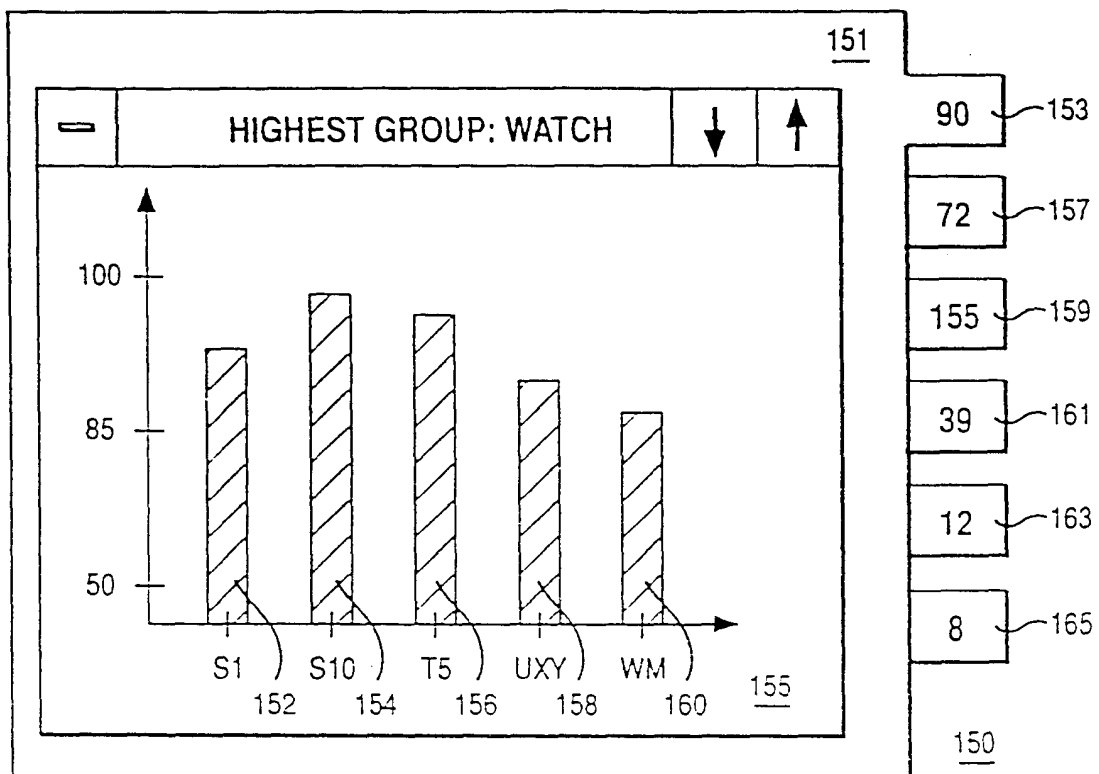


图 5 A

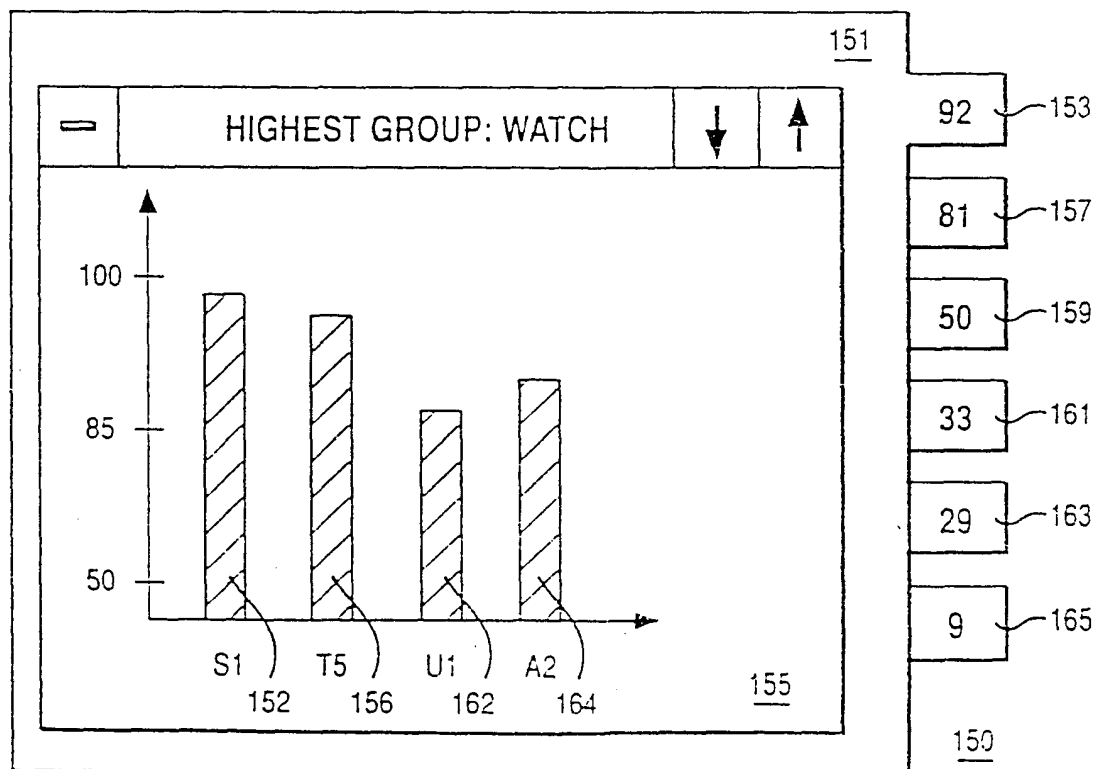
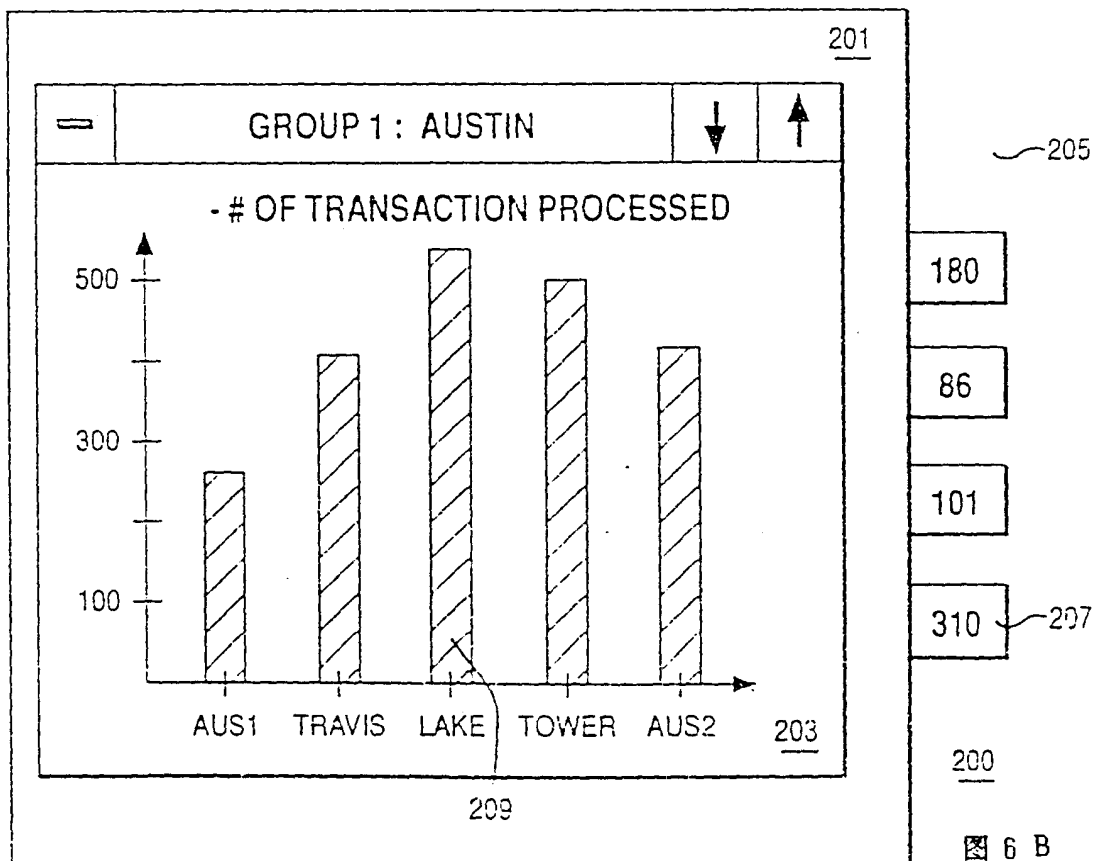
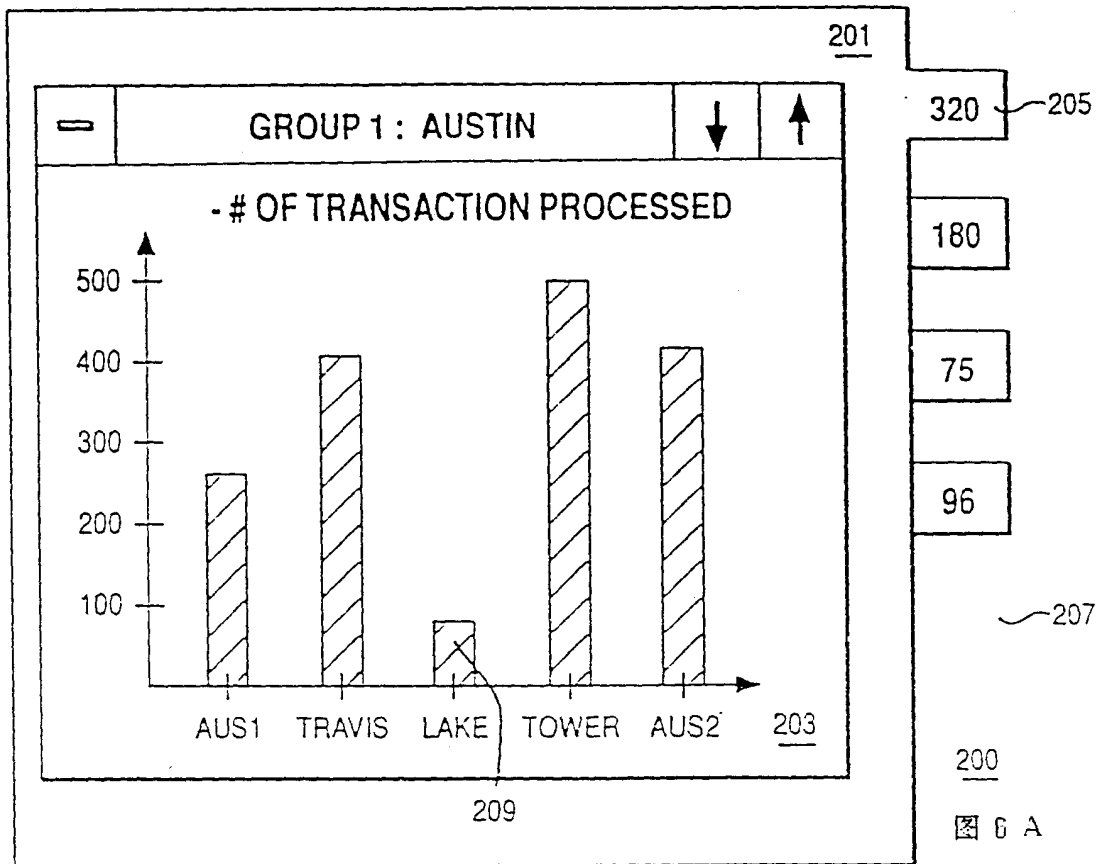


图 5 B



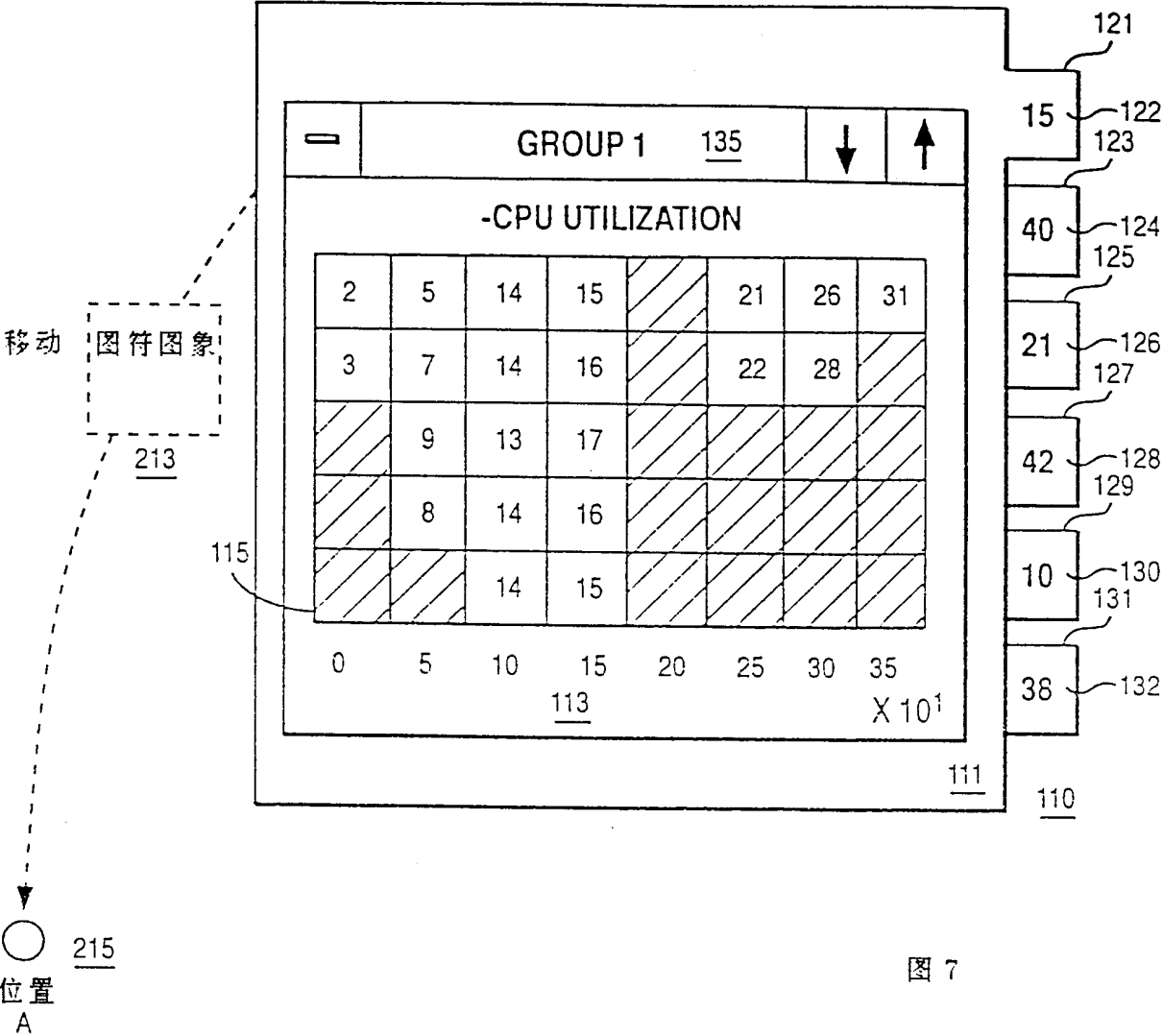
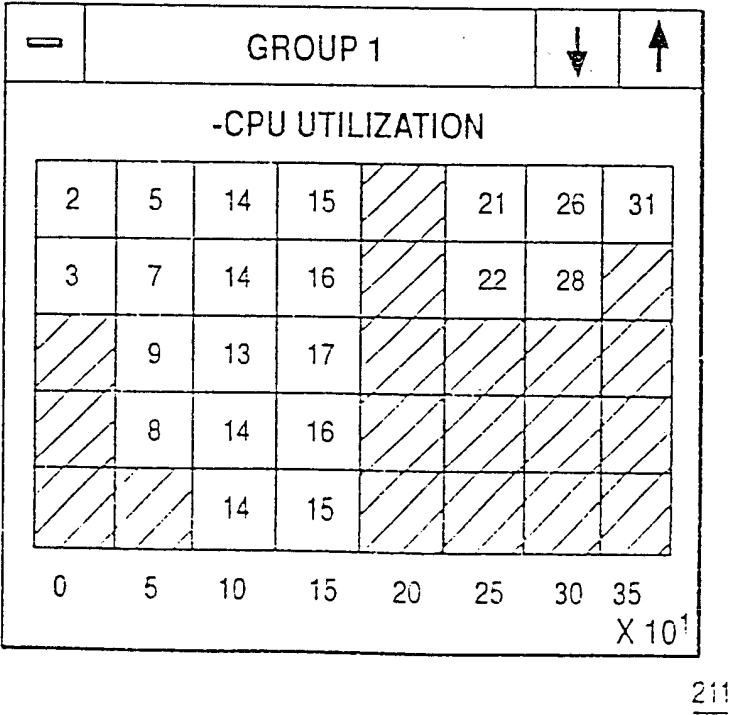


图 7



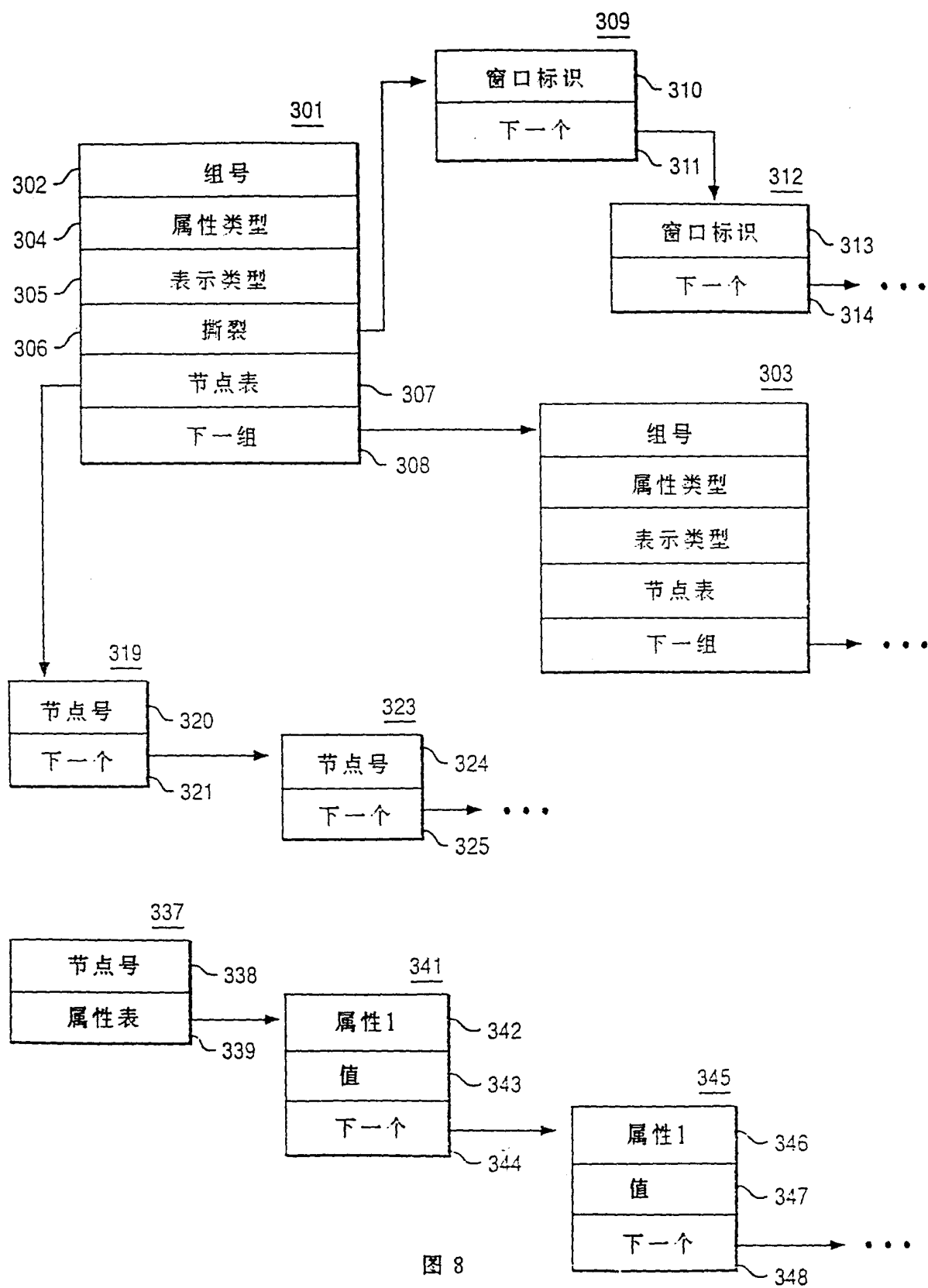
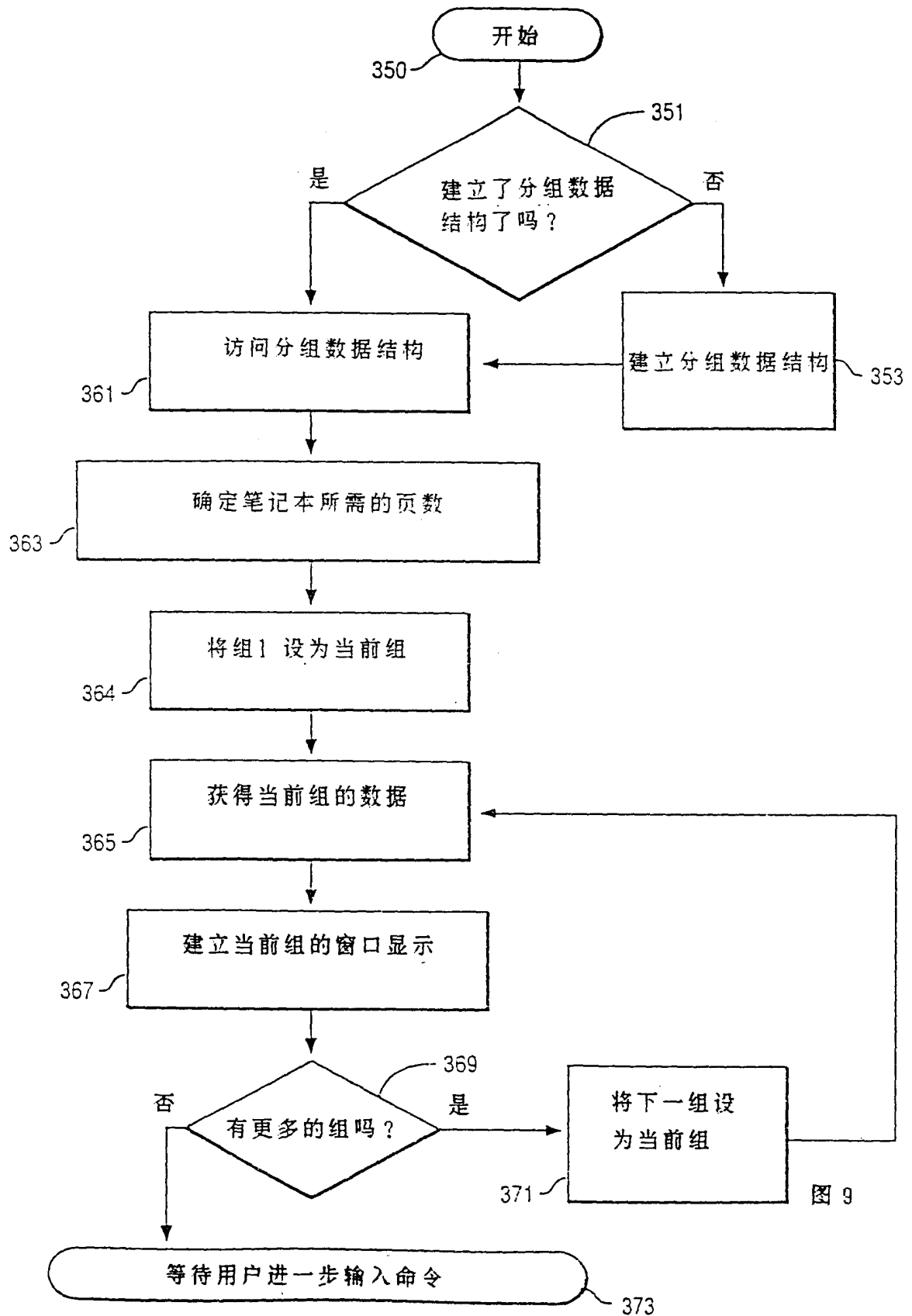


图 8



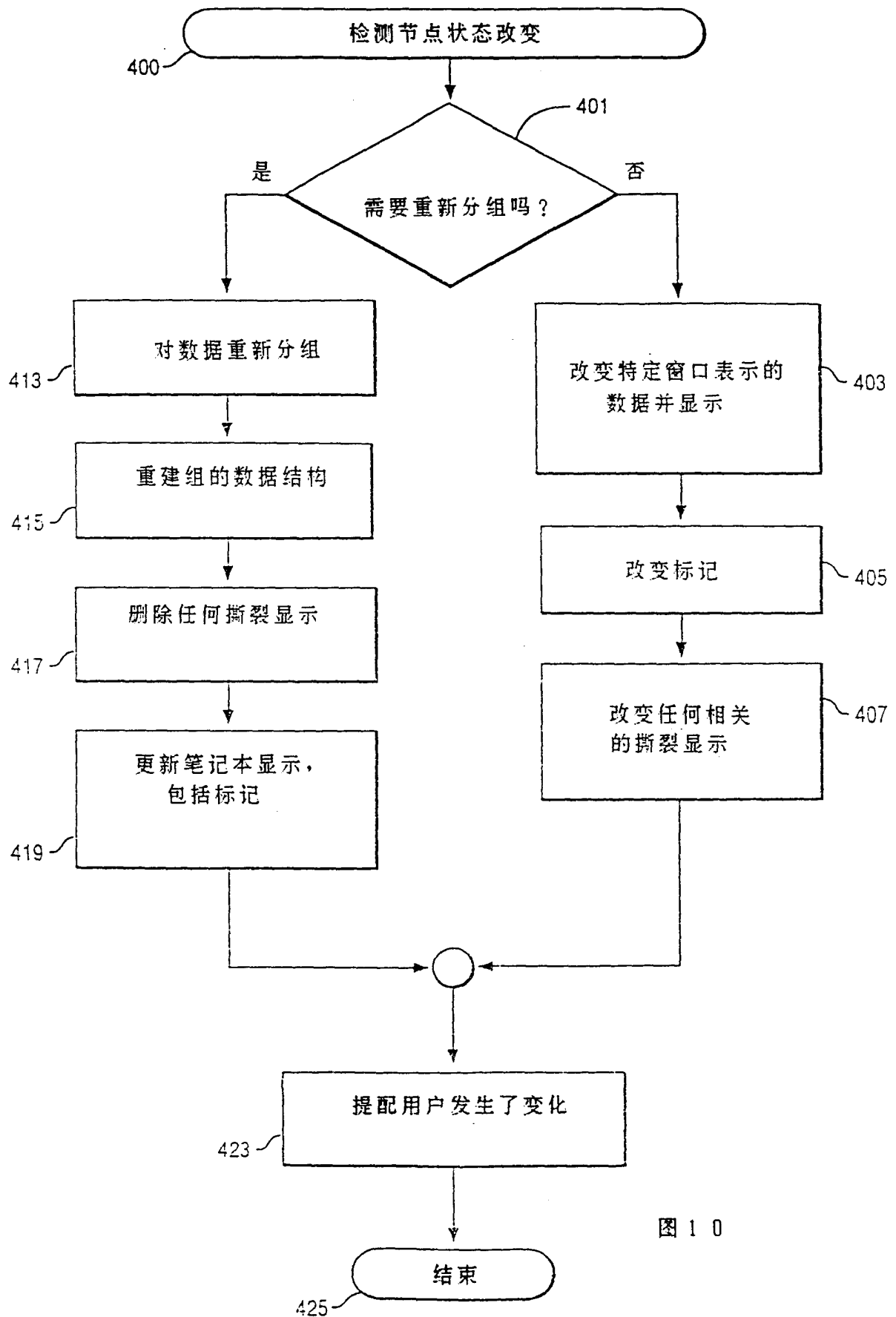
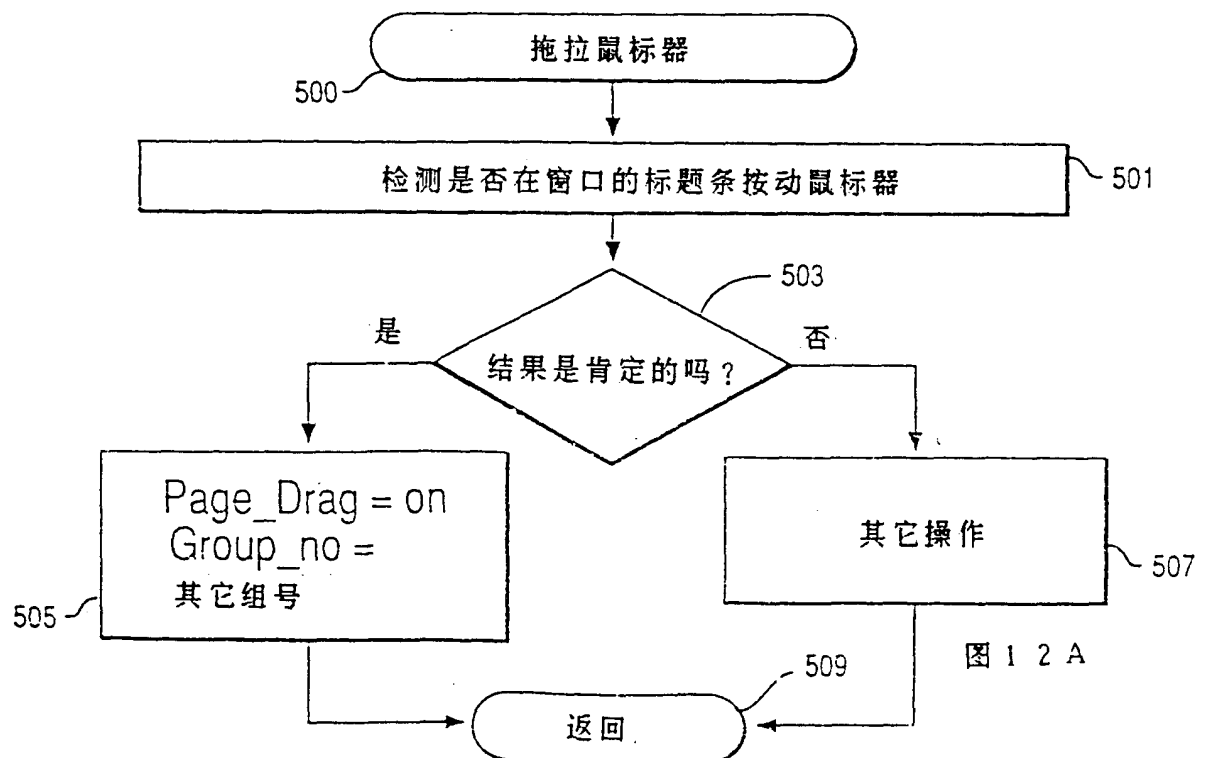
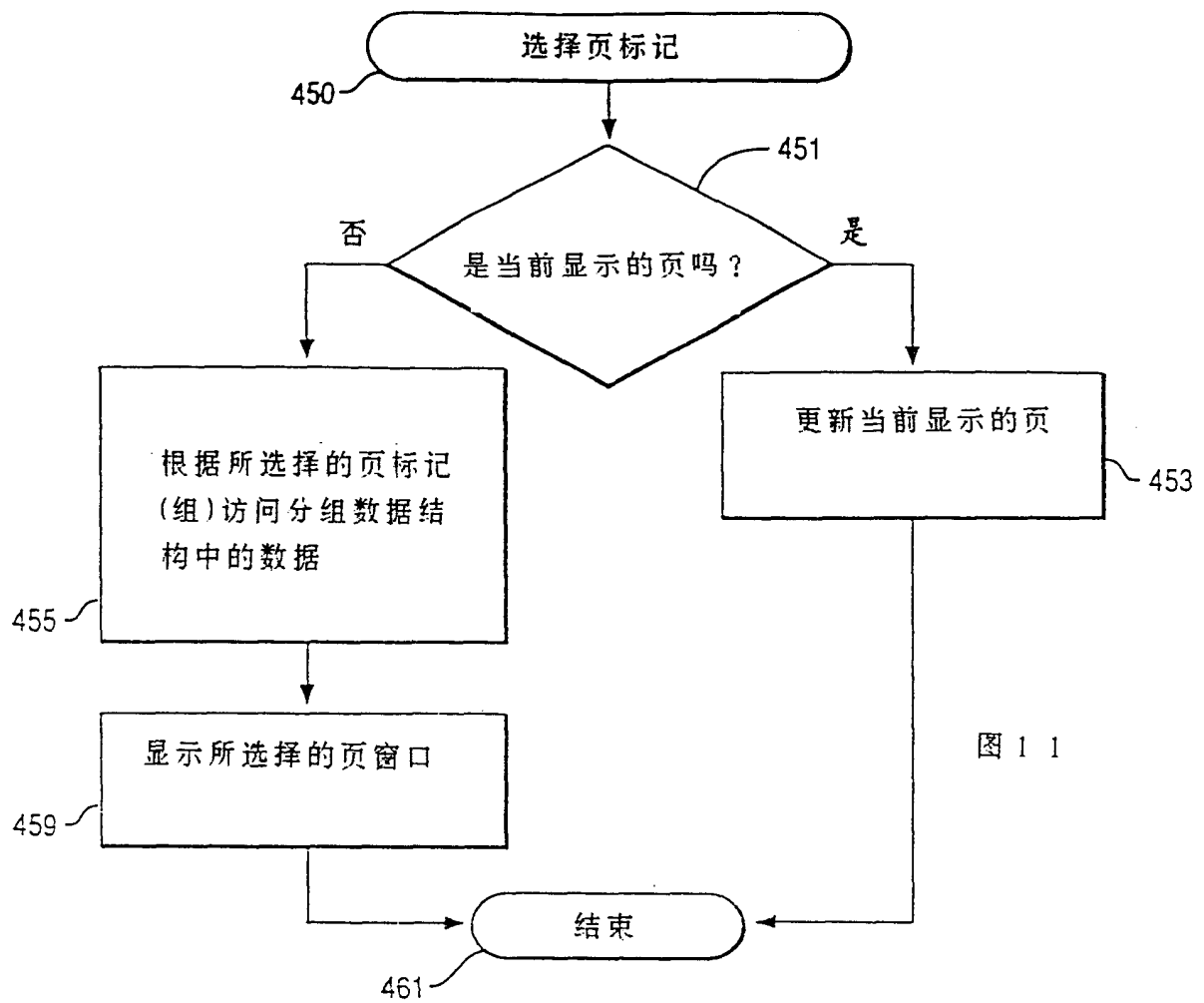


图 10



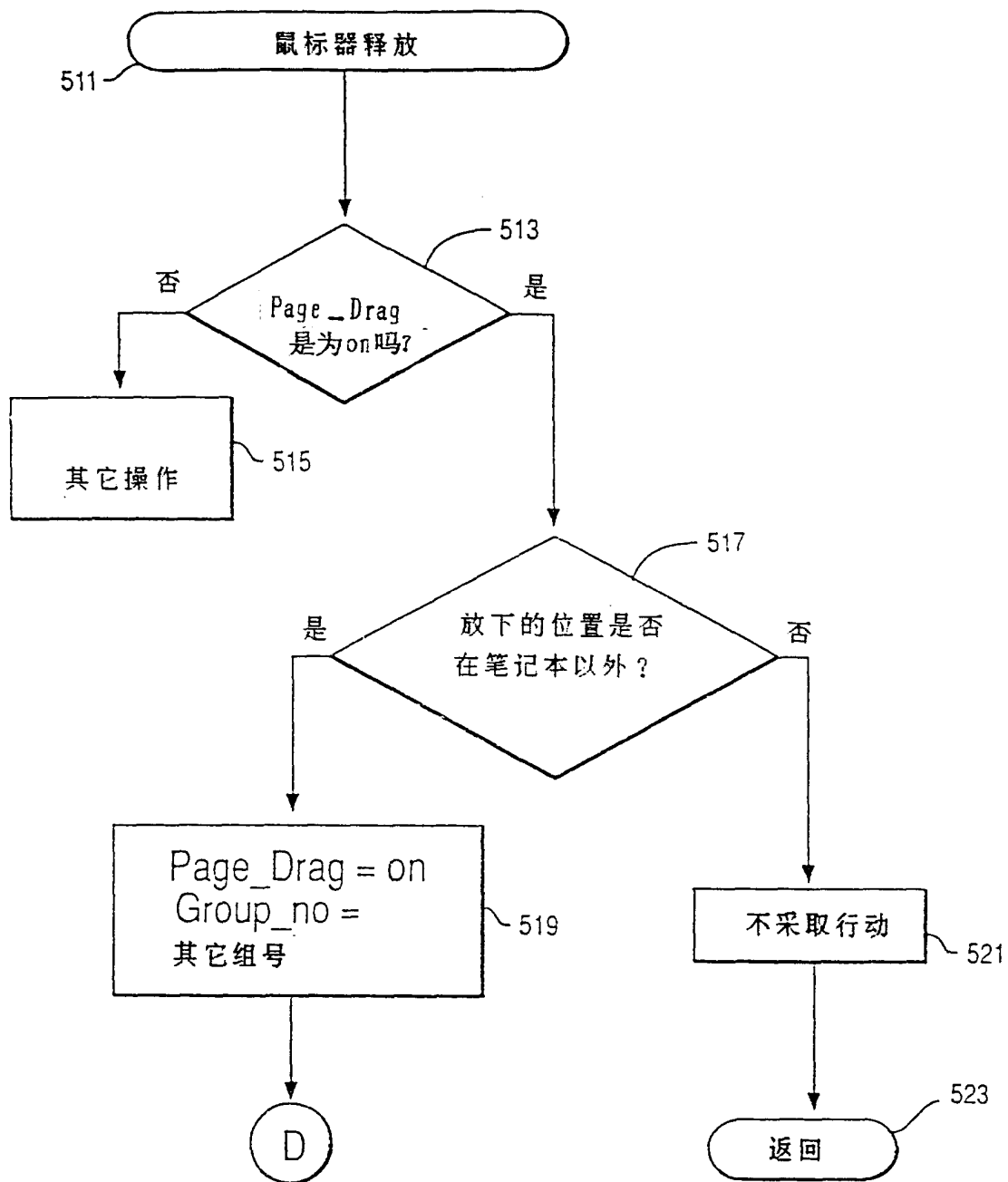
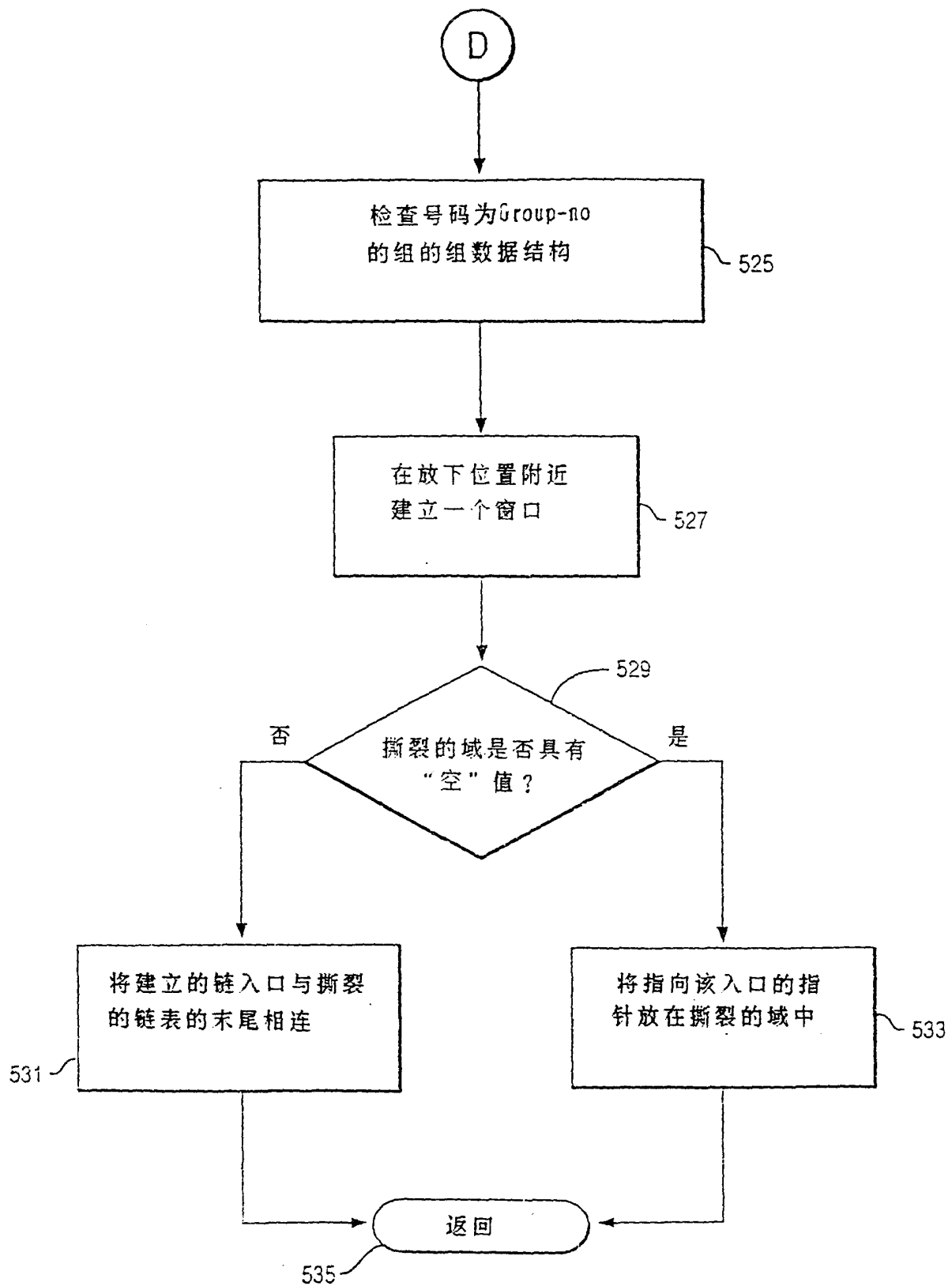


图 1 2 B



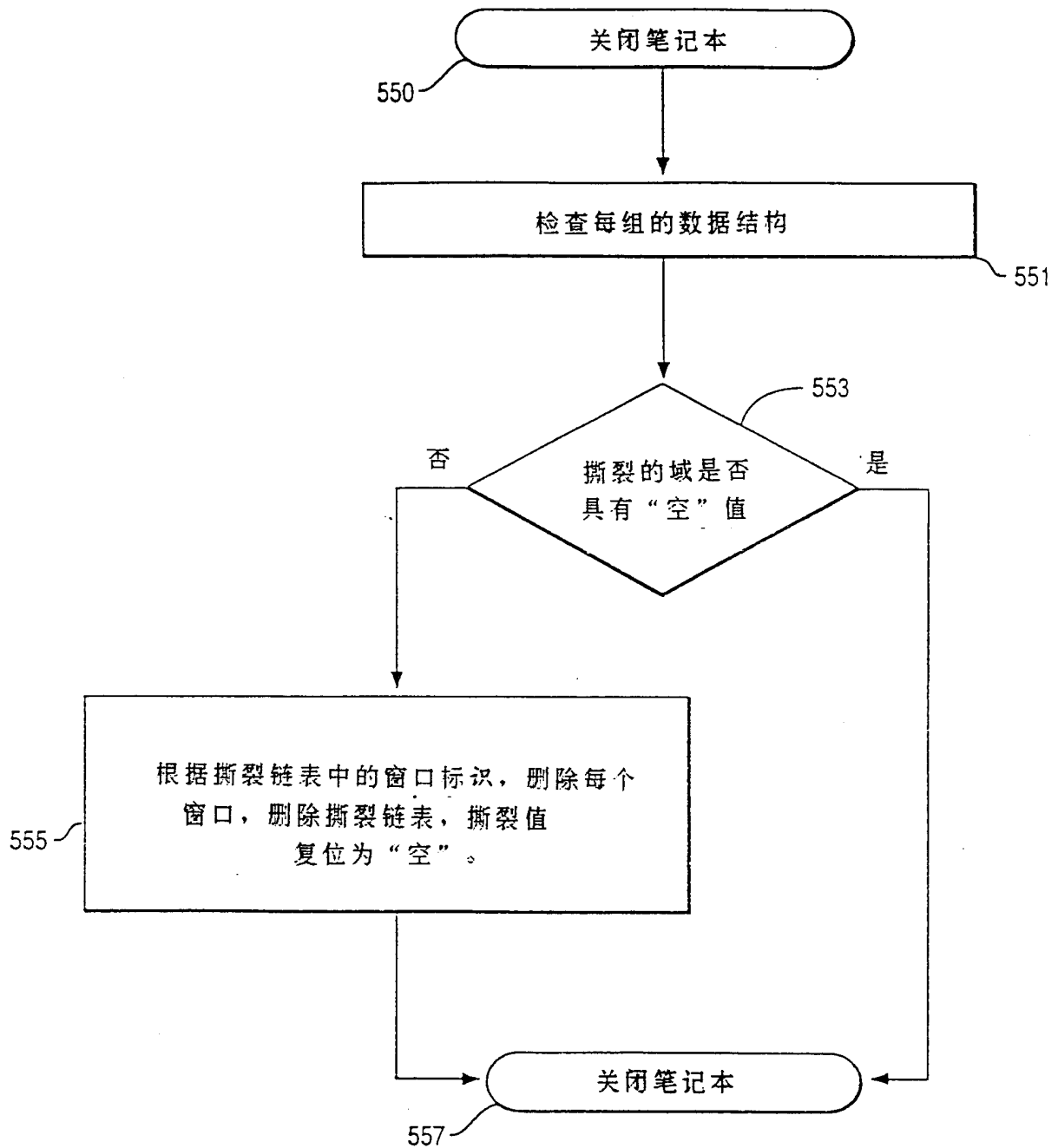


图 1 3