

[12] 发明专利说明书

[21] ZL 专利号 97121209.0

[45]授权公告日 2002年4月10日

[11]授权公告号 CN 1082688C

[22]申请日 1997.10.28 [24]颁证日 2002.4.10

[21]申请号 97121209.0

[30]优先权

[32]1996.11.26 [33]US [31]031,494

[32]1997.3.13 [33]US [31]816,917

[73]专利权人 国际商业机器公司

地址 美国纽约

[72]发明人 弗兰克·诺曼·埃克森

迈克尔·亨利·索多尔·哈克

纳吉·哈里姆 理查德·帕尔文·金

[56]参考文献

EP 0136666 1985. 4. 10 G06F15/16

EP 0575067 1993. 12. 22 G06F12/16

US 5386552 1995. 1. 31 G06F12/00

审查员 朱世茵

[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

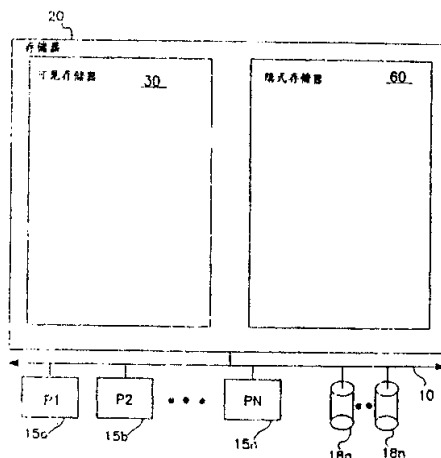
代理人 杨国旭

权利要求书 3 页 说明书 11 页 附图页数 6 页

[54]发明名称 保护存储器中的数据计算机系统和方法

[57]摘要

一个计算机系统的存储器包括可见部分和隐式部分。存储器的可见部分是可由处理器及在计算机系统内运行的操作系统寻址的。对于存储器的隐式部分来说,则排除了由处理器或操作系统寻址的可能性。存储器的隐式部分用来存储由处理器或操作系统传送的数据。位于存储器可见部分的通信区接收来自处理器或操作系统的存取请求。既定址存储器的可见部分,又定址存储器的隐式部分的隐式服务程序接收来自通信区的数据存取请求,并启动从隐式存储器的数据存取。



ISSN 1008-4274



权 利 要 求 书

1. 一种计算机系统，包括存储器，至少一个处理器和一个操作系统，它包括：

所述存储器的可见部分，可由所述的至少一个处理器和所述的操作系统寻址；

所述存储器的隐式部分，它排除了所述的至少一个处理器和所述的操作系统对其寻址的可能性，以及用来存储由所述的至少一个处理器和所述的操作系统两者之一传送的数据；

位于所述存储器的所述可见部分的通信区，用来接收来自所述的至少一个处理器以及所述的操作系统两者之一的数据存取请求；以及

隐式服务装置，用于既定址所述存储器的所述可见部分，又定址所述存储器的所述隐式部分，接收来自所述通信区的所述数据存取请求，以及启动从所述隐式存储器的所述数据存取。

2. 根据权利要求 1 的计算机系统，其特征在于：当所述处理器包括两个以上时，所述隐式服务装置由所述的至少一个处理器的第一个处理器来执行，而所述的操作系统由所述的至少一个处理器的第二个处理器来执行。

3. 如权利要求 1 中所述的计算机系统，其特征在于：进一步包括一个设备，它进行控制，使得当所述处理器处于第一种状态时，隐式存储器对所述处理器来说是不可见的，而处于第二种状态时，隐式存储器对所述处理器来说是可见的。所述的设备进一步还包括：

在所述第一种状态和所述第二种状态之间改变所述处理器的装置；

当所述的隐式服务装置处于所述的第二种状态下时，由所述隐式服务装置启动所需的操作的装置。

4. 如权利要求 1 中所述的计算机系统，其特征在于：所述的隐式服务装置是可重新装入和可重新启动的，所述存储器的所述可见部

分紧接着所述隐式服务装置再启动之前以及紧接着再启动之后都是相同的。

5. 如权利要求 1 中所述的计算机系统，其特征在于：进一步包括随机存取存储磁盘服务装置，用于在所述存储器的所述隐式部分存储数据，经受住故障，及重新启动操作系统，重新装入和重新启动隐式服务装置。

6. 如权利要求 1 中所述的计算机系统，其特征在于进一步包括磁盘超高速缓存服务装置，用于在所述存储器的所述隐式部分存储数据，经受住故障及重新启动所述的操作系统，及重新装入和重新启动隐式服务装置。

7. 一种保护存储在位于计算机系统内的存储器中的数据的方法，其中所述的计算机系统包括至少一个处理器和一个操作系统，所述的方法包括以下一些步骤：

分配可由所述的至少一个处理器和所述的操作系统寻址的所述存储器的可见部分；

分配所述的至少一个处理器和所述的操作系统不能寻址的、以及用来存储由所述的至少一个处理器和所述的操作系统传送的数据的所述存储器的隐式部分；

在位于所述可见存储器中的通信区内接收来自所述的至少一个处理器和所述的操作系统两者之中至少一个的请求；

由既定址所述存储器的所述可见部分、又定址所述存储器的所述隐式部分的隐式服务装置接收所述请求的每一个请求；以及

从所述的隐式存储器启动所述请求的每一个请求。

8. 根据权利要求 7 保护存储在存储器中的数据的方法，其特征在于：进一步包括以下步骤：

把所述处理器从第一种状态改变为第二种状态，在第一种状态下所述存储器的所述隐式部分对于所述的处理器来说是不可见的，在第二种状态下所述存储器的所述隐式部分对于所述处理器来说是可见的；

把所述处理器的控制权传送给所述的隐式服务装置；
 由所述的隐式服务装置接收执行所需操作的请求；
 由所述的隐式服务装置启动所述的所需操作；
 由所述的隐式服务装置终止所述处理器的控制权；以及
 把所述的处理器从所述的第二种状态改变为所述的第一种状态。

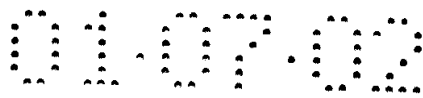
9. 根据权利要求 8 保护存储在存储器中的数据的方法，其特征在于：进一步包括下列步骤：

重新装入所述的隐式服务装置；
 重新启动所述的隐式服务装置；以及
 保持所述存储器的所述可见部分紧接着所述隐式服务装置再启动之前以及紧接再启动之后是相同的。

10. 根据权利要求 7 保护存储在存储器中的数据的方法，其特征在于：进一步包括提供随机存取存储磁盘服务程序的步骤，该服务程序在所述存储器的所述隐式部分存储数据，能经受住故障及重新启动操作系统，以及重新装入和重新启动隐式服务装置。

11. 根据权利要求 7 保护存储在存储器中的数据的方法，其特征在于进一步包括提供磁盘超高速缓冲服务程序的步骤，该服务程序在所述存储器的所述隐式部分存储数据，能经受住故障及重新启动操作系统，以及重新装入和重新启动隐式服务装置。

12. 根据权利要求 7 保护存储在存储器中的数据的方法，其特征在于：当所述处理器包括两个以上时，所述的隐式服务装置由所述的至少一个处理器中的第一个处理器执行，而所述的操作系统则由所述的至少一个处理器中的第二个处理器执行。



说 明 书

保护存储器中的数据的计算机系统和方法

总的来说，本发明涉及计算机系统，更准确地说，它涉及对存储在位于计算机系统内的存储器部分中的数据的保护。特别是公开了在计算机系统和操作系统对其没有直接访问权的存储器的部分之间传送数据的机制。

当一个进程在计算机系统上进行时，该进程利用存储器的指定部分存储和检索数据。其它进程也可能访问存储器的那些指定部分。

当在计算机系统上运行的第一个进程被中断而随后又恢复时，在第一个进程被中断的期间，在计算机系统上运行的第二个进程可能会访问先前被第一个进程使用过的存储器的部分。当第一个进程被中断时，如果第二个进程重写由第一个进程存储在存储器中的数据值，那末，当第一个进程恢复时，在它被中断以前它曾存储在存储器中的数据值可能将不复存在，这就可能产生错误。

此外，如果几个进程在一个计算机系统内同时运行，一个进程可能把数据存储到另一进程正在使用的存储器的部分中去。这样，在一个进程检索其已存储的数据之前，该数据可能已被破坏。

一个计算机系统可能经历系统再启动，这也是典型的。这样的计算机系统可能有一个提供数据缓冲器的软件应用程序在运行，该缓冲器的内容不会因系统再启动而被擦除。例如，这种情况可能会在纽约州 Armonk 镇的 IBM 公司的一项产品，称为“事项处理设施(TPF)”的软件包中遇到。在 TPF 中，如果有系统再启动，某些数据缓冲器的内容不会被擦除。在再启动完成之后，当 TPF 继续在系统再启动之前它所处的点上运行时，未被擦除的数据缓冲器的内容仍保持原样是最理想的。但是，这些数据缓冲器是系统上运行的任何应用程序都可以访问的。因此，这些数据缓冲器可能会被操作系统中的或在操作系统上运行的应用程序中的错误所破坏。

一个计算机系统的存储器包括可见部分和隐式部分。存储器的可见部分可由处理器及在计算机系统内运行的操作系统访问。对于存储器的隐式部分来说，则排除了由处理器或操作系统访问的可能性。存储器的隐式部分用来存储由处理器或操作系统传送的数据。位于存储器可见部分的通信区接收来自处理器或操作系统的存取请求。既能访问存储器的可见部分，又能访问存储器的隐式部分的隐式服务程序接收来自通信区的数据存取请求，并启动从隐式存储器的数据存取。

图 1 是根据本发明的典型实施例说明存储器配置及其与多个处理器和多个输入/输出 (I/O) 装置关系的框图。

图 2 是说明图 1 所示的可见存储器的结构及图 1 所示的可见存储器与隐式存储器之间的通信路径的框图。

图 3 是说明图 1 所示的隐式存储器内部结构的框图。

图 4 是根据本发明的典型实施例说明操作系统放弃处理器对隐式服务程序的控制权的某些处理步骤的流程图。

图 5 是根据本发明的典型实施例说明图 3 的隐式服务程序所完成的步骤的流程图。

图 6 是根据本发明的典型实施例说明用来向隐式服务程序传送请求的队列结构的框图。

图 1 根据本发明的典型实施例说明一个计算机系统。如图所示，计算机硬件包括一个或多个处理器 15a - 15n，它们通过某个总线或其它互连机构 10 与存储器 20 连接。

可以使用任何计算机系统，但是，根据本发明的典型实施例，可以使用 1994 年 6 月第 24 届国际容错计算专题讨论会汇编第 414 - 423 页，由 Abbott, M., Har, D., Herger, L., kauffman, M., Mak, K., Murdock, J., Schulz, C., Smith, T.B., Tremaine, B., Yeh, D., Wong, L. 所著的“耐用存储器 RS/6000 系统设计”中所描述的“耐用存储器 RS/6000 (DM/6000)”。这个系统提供这样的硬件冗余，使任一硬件部件或电源的故障都不会使计算机停机。

存储器 20 的一个部分是可见存储器 30。在每个处理器 15a - 15n 上运行的程序有权访问可见存储器 30。

如图 2 所示，可见存储器 30 包括一个象 AIX（由纽约州 Armonk 镇的 IBM 公司生产）那样的通用操作系统 40 以及应用程序 5a-5n。存储器 20 的另一部分是隐式存储器 60。有几种方法可防止操作系统 40 及应用程序 5a-5n 访问这个隐式存储器。一种较可取的方法利用一种能力：通知 AIX，存储器 20 的某些部分是有故障的。当 AIX 被引导时，它期待着找到一个包括对有故障的存储器部分的描述 IPL 控制块。AIX 将决不访问或以任何方式改变被确定为有故障的任何存储器的内容，在其虚拟地址表中对这类存储器不作记录。因此，当 AIX 为其正在使用的无论那个处理器 15a-15n 一接通虚拟地址变换时，这个存储器就被有效地隐藏起来。这样，即使在它可被访问的短暂的初始化阶段，存储器 60 也决没有被修改的危险。其它操作系统在引导时能指定可用存储器。如果引导时指定的可用存储容量小于总的实际存储容量，可用存储器指定区域之外的存储器部分对于操作系统来说将是不可见的。

隐式服务程序 70 控制对隐式存储器 60 的存取。隐式服务程序 70 是装入隐式存储器 60 的一个程序。隐式服务程序 70 中的地址转换可以断开。隐式服务程序 70 及操作系统 40 和应用程序 5a-5n 共同使用位于可见存储器 30 内的通信区 80。这是隐式服务程序 70 和操作系统 40 及应用程序 5a-5n 后两者赖以通信的机构。把所有的通信机构都放置在可见存储器 30 之内，就排除了操作系统 40 和应用程序 5a-5n 损坏隐式存储器的可能性。

在最佳实施例中，通信区 80 包括队列 90 和请求块 100a-100n。队列 90 最好占有存储器的单独一页，用来形成队列的链式单元的所有指示字最好不是地址，而是那一页内的位移。这样，队列 90 能被在虚拟地址变换接通时工作的在操作系统 40 下运行的程序控制，以及能被在虚拟地址变换断开的情况下可以工作的隐式服务程序 70 所控制。队列 90 的条目有链式位移，以便连接队列条目及请求块 100a-100n 的实地址。通过利用队列的首部指示字的实地址，隐式服务程序 70 不使用 AIX 的虚拟地址变换机制，就能使用首部指示字。

如图 3 所示，隐式存储器 60 的一部分是稳态存储器 110。它包括一个稳态存储器描述符 120，隐式服务程序 70 利用预先安排的地址能确定

描述符 120 的位置，隐式服务程序 70 还利用描述符 120 来确定，稳态存储器 110 的那些部分正在使用。稳态存储器 110 的一小部分被隐式服务程序 70 用来保持其自己的记录，但是，稳态存储器 110 的大部分由专用服务程序 130a-130n 使用，这些服务程序都是隐式服务程序 70 可访问的程序。例如，最佳实施例包括一个 VDISK 服务程序 130a 及一个 PCACHE 服务程序 130b，以及其它服务程序。

VDISK 服务程序 130a 处理使用稳态存储器 110 的空间存储数据的命令，好象是在一个或多个磁盘上。以这种方式被使用的存储器的每个空间都称之为 VDISK。VDISK 服务程序 130a 最好至少能提供处理下述请求的能力，即分配稳态存储器 110 中被用来存储数据的空间的请求，把数据从可见存储器 30 复制到稳态存储器 110 的请求，以及把数据从稳态存储器 110 复制到可见存储器 30 的请求。PCACHE 服务程序 130b 超高速缓存准备在较晚的时候写到磁盘上去的数据。PCACHE 服务程序 130b 把数据的磁盘地址与数据本身一起存储起来，以便在操作系统 40 发生故障后再现超向速缓存的信息。

当装入并启动隐式服务程序 70 后，就要测试稳态存储器描述符 120 的有效性，如果它不是有效时，就使其初始化，以表明稳态存储器中没有一个空间在使用。对有效性的可能的测试有，例如，将检验和的值与对稳态存储器描述符 120 的内容重新计算所得的值进行比较，把存储器使用计数加起来查看它们是否与稳态存储器 110 的总数一致，以及在完成存储器的隐式服务程序 70 的初始化后对放在稳态存储器描述符 120 中的特征关键值进行比较。由隐式服务程序 70 的新近开始的版本使用先有的稳态存储器描述符 120 就使得有可能替换隐式服务程序 70 的代码而没有明显的信息损失。当然最为理想的是隐式服务程序 70 的新版本能解释旧版本的稳态存储器描述符 120。

隐式服务程序 70 利用稳态存储器描述符 120 的一部分来记录服务数据句柄(handle)115a-115n，每个句柄被分配给不同的服务程序 130a-130n。一个服务数据句柄指向一个数据描述符。一个数据描述符指出可供相应的服务程序使用的隐式存储器 60 的部分。

例如，VDISK 服务程序 130a 使用 VDISK 数据描述符 140a。每当

隐式服务程序 70 被启动时，VDISK 服务程序 130a 也就被启动（所有其它服务程序 130b-130n 也都如此），并被给与一个其服务数据句柄 115a 的复制品，后者或是零，或是指 VDISK 数据描述符 140a 的指示字。VDISK 服务程序 130a 确认其服务数据句柄 115a 指向 VDISK 数据描述符 140a 以及这个描述符 140a 的有效性，就象隐式服务程序 70 确认稳态存储器描述符 120 的存在及其有效性一样。如果 VDISK 服务程序 130a 发现 VDISK 数据描述符 140a 无效时，它就请求由隐式服务程序 70 分配一个并使其初始化，以表明还没有任何 VDISK 存在。VDISK 数据描述符 140a 代表其在操作系统 40 及应用程序 5a-5n 中的用户用来记录稳态存储器 110 的那些部分正被用来存储数据。

服务数据句柄 115a-115n 及数据描述符 140a-140n 的这种用途使服务程序 130a-130n 能被停止和再启动而无信息损失或明显的运行中断，就象隐式服务程序 70 能被停止，替换和再启动那样。正如隐式服务程序 70 的情况一样，由旧版本的服务程序 130a-130n 所产生的数据描述符 140a-140n 最好能被新版本的服务程序 130a-130n 所理解。

在最佳实施例中，对隐式服务程序 70 的请求包括服务程序 130a-130n 的名字，这也就是要处理该请求，从而隐式服务程序 70 能把该请求传送给的那个服务程序 130a-130n。任一服务程序可能需要的附加请求信息可在请求块 100a-100n 中找到。在 VDISK 服务程序 130a 的情况下，其包括表明是否要产生一个 VDISK 的操作码，象 VDISK 标识符那样用来指出应对那个 VDISK 进行操作的操作数的所需空间，以及象返回码和完成位那样的任一应答信息的所需空间，该返回码表示请求成功或失败，而完成位则用来指出在处理请求时什么时候完成 VDISK 服务程序 130a。

在最佳实施例中，VDISK 服务程序 130a 由安装到操作系统 40 中的 VDISK 设备驱动程序 160a 使用。VDISK 设备驱动程序 160a 就象安装在操作系统 40 中的任何磁盘驱动程序一样，提供同样的接口，不过它是利用 VDISK 服务程序 130a 而不是实磁盘设备来存储和检索数据。这就使操作系统 40 的其它部分及应用程序 5a-5n 能利用 VDISK，就象利用实磁盘一样。由于隐式存储器 60 决不会被在可见存储器 30 中运行的

任何程序的任何故障所破坏，由于稳态存储器 110 即使在隐式服务程序 70 及其有关的服务程序 130a-130n 再启动时也仍被保存，还由于 DM/6000 硬件的容错和电源保护性能，由 VDISK 服务程序 130a 存储在稳态存储器 110 中的数据可以被认为是象存储在实磁盘上一样安全。

传送到隐式服务程序 70 的所有地址都是可见存储器 30 中的实地址，它们是由在 AIX 下运行的 VDISK 设备驱动程序 160a 利用 AIX 的核心服务程序确定的，而请求块 100a-100n 合乎要求地被限制在存储器的单独一页，这样就避免了地址转换问题。

当隐式服务程序 70 在处理器 15a-15n 的子设备上运行，而该子设备又与操作系统 40 在其上运行的子设备分离时，隐式服务程序 70 对请求的处理是与操作系统 40 及应用程序 5a-5n 的处理并行的。当隐式服务程序 70 和操作系统 40 及应用程序 5a-5n 共用同一个处理器 15a-15n 时，最好利用传送程序 170 在两者之间来回传送控制权，传送程序 170 记录当前被安排为由操作系统 40 使用的处理器 15a-15n 的状态，然后改变处理器 15a-15n 的状态，以使隐式服务程序 70 能正常运行。被记录的状态除了别的之外，可能还包括寄存器的内容及虚拟地址变换装置的设置。

在最佳实施例中，传送程序 170 被放在存储器的第一个 64 千字节中。AIX 总是把其虚拟存储表设置成这样的，使存储器的这个部分有与其实地址相同的虚拟地址。因此，存储器这个区域内的程序能接通或断开地址转换，并仍能继续执行同一个程序的下一条指令。

虽然在操作系统 40 下运行的 VDISK 设备驱动程序 160a 及 VDISK 服务程序 130a 之间有发送信号的其它方法，例如利用中断，最佳实施例则利用请求块 100a-100n 的应答区内的完成标志。如果这个隐式服务程序 70 与操作系统 40 共用同一个处理器时，VDISK 设备驱动程序 160a 就在检验完成标志和把控制权给予隐式服务程序 70 之间同步循环。

从应用程序 5a-5n 请求 VDISK 中的数据直到把该数据传送回同一个应用程序的整个流程路径如下。请求首先进到确定设备就是 VDISK 设备的操作系统 40。因此，请求就被传送到 VDISK 设备驱动程序 160a，该驱动程序 160a 确定那一个 VDISK 装有数据，并在请求块 100a-100n 中建立请求，这个请求表明：这是对 VDISK 服务程序的一个请求，从那

个 VDISK 得到数据，在 VDISK 的什么地方开始读出，传送多少数据，以及把要读出的数据放在可见存储器 30 的什么地方。请注意，这个时候在请求块 100a-100n 中把完成标志设置成表明请求还没有结束。然后，VDISK 设备驱动程序 160a 在队列 90 中放置一个指向这个请求块 100a-100n 的队列条目，并开始等待完成标志来表明请求结束。在等待的同时，如果处理器 15a-15n 是由操作系统 40 及隐式服务程序 70 所共用时。VDISK 设备驱动程序 160a 将调用传送程序 170 把处理器 15a-15n 的控制权给予隐式服务程序 70。在任何情况下，当轮到隐式服务程序 70 时它将使这个队列条目出队并且确定，并且从请求块 100a-100n 确定由 VDISK 服务程序处理什么。利用其服务数据句柄 115a，VDISK 服务程序 130a 将确定其 VDISK 数据描述符 140a 的位置，并从而确定 VDISK 数据的位置，VDISK 服务程序 130a 随后将把 VDISK 数据复制到请求块 100a-100n 中所提到的可见存储器的部分中去，然后它将设置完成标志以表明操作结束。处理器 15a-15n 的控制权返回到隐式服务程序 70，如果它是由隐式服务程序 70 与操作系统 40 共用时，隐式服务程序 70 又将把把控制权返回到传送程序 170，并从而返回到 VDISK 设备驱动程序 130a。无论如何，VDISK 设备驱动程序 160a 将最终拥有处理器 15a-15n 的控制权并观察到完成标志已被设置，然后把控制权返回到操作系统 40，后者将把控制权返回到做出初始请求的应用程序 5a-5n。

PCACHE 设备驱动程序 160b 在下列方面与 VDISK 设备驱动程序 160a 非常相似：对于操作系统 40 和应用程序 5a-5n 来说，它具有实磁盘设备的外表，它还利用隐式服务程序 70 和 PCACHE 服务程序 130b 从稳态存储器存储和检索数据。区别在于，PCACHE 设备驱动程序 160b 只是利用 PCACHE 服务程序 130b 来保持一个新近从实磁盘 18a-18n 中读出的或写入其中的数据的超高速缓存，PCACHE 设备驱动程序 160b 利用供那个特定的磁盘设备 18a-18n 用的设备驱动程序 160a-160n 来访问实磁盘 18a-18n。这与 1994 年 11 月 22 日 IBM T.J.沃森研究中心，由 C.R.Attanasio, M.Butrico, C.A.Polyzois, S.E.Smith 所著的 RC - 19843 号报告可恢复虚拟共享磁盘的设计和实现中所描述的 VSD 设备驱动程序的工作方法大

致相同。

在操作系统 40 发生故障的情况下，可见存储器 30 的内容完全丢失。当操作系统 40 被再启动时，通过由 PCACHE 服务程序 130b 请求目前存在于稳态存储器 110 中的那些块的内部地址表，PCACHE 设备驱动程序 160b 重建其超高速缓存单元信息的目录。PCACHE 设备驱动程序 160b 然后就能重建它所使用的无论什么样的内部数据结构，以便监视超高速缓存的状态，例如散列表，因而在正常操作过程中就不再需要由 PCACHE 服务程序 130b 所存储的目录信息。

图 4 是操作系统 40，或者更准确地说，安装在操作系统内的设备驱动程序 160a-160n 及传送程序 170 为了完成控制权的传送所采取的步骤的逻辑流程图。首先，把准备由隐式服务程序 70 及专用服务程序 130a-130n 处理的任何请求放在通信区 80 的某个请求块 100a-100n 内（步骤 410）。然后，那个处理器 15a-15n 在执行这些步骤，就把该处理器的状态记录在可见存储器 30 的某个地方（步骤 420）。为了使隐式存储器 60 对这个处理器来说变为可见（步骤 430），在最佳实施例中，就把处理器的虚拟地址变换断开，不过其它方法也完全是可能的。现在，这个处理器就能访问那个隐式存储器 60，它能开始执行装在隐式服务程序 70 的程序中的指令（步骤 440）。当隐式服务程序 70 准备放弃处理器的控制权时，这个程序的下一步在传送程序 170 中再继续下去，其进行隐藏隐式存储器 60（步骤 450）。在最佳实施例中，这是通过把虚拟地址变换重新接通来实现的。然后，处理器状态的其余部分就能被恢复（步骤 460），而控制权则被返回到启动传送的设备驱动程序 160a-160n。于是就能检验放在通信区 80 或可见存储器 30 中某个地方的应答数据（步骤 470）。

图 5 是隐式服务程序 70 为了处理请求所完成的步骤的逻辑流程图。第一步 510 是等待在通信区 80 内出现请求。当请求出现时，在第 520 步就完成与所做出的请求的性质相适应的某种处理。例如，请求数据将指示对该请求给出某个特定的服务程序并给出该由

那个服务程序完成的命令。这个动作的结果就是把应答信息放在通信区 80 或可见存储器 30 的任一单元（步骤 530）。例如，原始请求的内容能确定一个在可见存储器 30 之内但是在通信区之外的单元，特定请求的应答信息应放置到该单元里。然后，在第 540 步，就要进行一项测试，看看目前运行隐式服务程序 70 的程序的处理器是否也用来运行操作系统 40。这可用多种方法实现，如检验存在于通信区内的配置信息。如果处理器被共享时，隐式服务程序将在 540 这个点上将控制权传送回操作系统 40（步骤 550）。

图 6 是通信区 80 内的队列 90 在置入一系列请求之后看起来象什么的例子。A 条目是最老的，而 E 条目 610 则是最近增加到队列 90 的条目。为了避免条目入队和出队之间的干扰，需要一种同步方式。例如，尽管可以利用 IBM370 的 TEST（测试）和 SET（设置）指令序列来确保同步，但在最佳实施例中，却采用了 Power PC 体系结构的装入和保留指令以及条件存储指令。正如技术上众所周知的那样，装入和保留指令在存储器单元中建立保留，以供条件存储指令使用。条件存储指令首先确定，在试图存储的存储器单元中是否有保留。如果有保留，进行存储，而保留则被清除。如果没有保留时，指令结束而没有改变存储器。

为了使一个队列条目入队，就在含有队列中第一个条目的位移的队列首标的字上执行装入和保留。新条目的“下一个”位移就被设置成那个值。如果那个字上的队列首标的保留仍然存在的话，新条目的位移就被存储在队列首标。如果保留不存在时，就重复入队过程。

例如，通过装入首部指示符 600 并带着保留，增加了 E 条目 610，而在增加 E 条目 610 之前，首部指示符 600 曾指向 D 条目 620。这个指示字被放在 E 条目 610 的转栏 613。最后，E 条目 610 的地址被有条件地存放在首部指示符 600 中。如果存储成功，入队完成。否则，就重复该过程。

当然，这就产生一个以后进先出方式组织的队列。产生队列时，在出队过程中又产生先进先出的风格。为了使一个条目出队，

就要在队列首标中执行首部位移的装入和保留。然后，通过有条件地存储零来代替那个值，就从整个队列除去队列首标。因保有全部队列，服务程序就可以自由地取消最老的队列请求。换句话说，如果队列是非空的，出队就完成了。否则，保留的东西就被返回到队列首标。

当把保留的东西返回到队列首标时，如果与此同时有其它条目被装入队列，条目最好与被出队过程除去的一组条目合并起来。为了完成合并，首部位移（队列首标中指向队列中第一个条目的字）须经装入和保留。如果首部位移等于零，就没有新的条目。这样，目前位于被返回到队列的那些条目首部的条目的位移就被条件存储起来。不然，零就被条件存储起来，而装入的值则被用作要与老队列合并的新队列的开始。然后重复过程。而且，如果有存储故障，就重复整个过程。

为了使一个条目 610 - 650 出队，最老的条目是所希望的。在这个例子中，那就是 A 条目 630。为了达到这一点，就要通过反复装入首部指示符 600 并附带保留，并在其位置上条件存储零，来移动整个链。这样重复进行，直到存储成功。队列就被来回移动，直至得到一个其尾指示字非零的条目。在这个例子中，那将是 A 条目 630 的尾指示字 632。A 条目 630 之前的条目按后进先出次序排列，就象入队算法所确定的那样。A 条目 630 之后的条目按先进先出的次序排列，隐式服务程序 70 通过把条目重新排列成 B - C - D - E 的次序将保持这种先进先出的状态（A 条目 630 不在那里，因为它是要从队列 90 中被除去的条目）。B 条目 640 将使其尾指示字 642 指向 E 条目 610。这表明两件事情：B 条目 640 是先进先出条目中的第一个条目，以及按后进先出次序找到的新条目应被放在尾指示字 642 指向的那个条目之后。每当队列 90 需要重新排列时，这就避免了顺序查对整个队列 90。

把剩余的条目返回队列 90 与使一个条目入队非常相似。当某些其它条目已被放置到队列 90，而隐式服务程序 70 正在处理作为出队过程的部分的条目时，就能看到差别。在那种情况下，从首部

指示符 600 装入的值非零并指向新入队的条目。正确的应答是隐式服务程序 70 除去那些条目，就象正常出队一样，并把它们与它先前除去的那些条目结合起来，以形成一个单独的表，并返回去重复把这个表放回到队列 90 的过程。

实现了一种优化方法，以避免每次需要链上最老的条目时，都要跑到后进先出链的末端。这包括在把表返回到队列以前重新排列条目。重新安排的、现在按先进先出次序排列的条目，都含有表的尾部的位移。这就使按先进先出次序排列的条目有别于具有零值而不是尾位移的按后进先出次序排列的条目，它还使尾部在最后插入时能快速定位。当新条目入队时，将有一些后进先出条目，其后面是一些先进先出条目，但是，这可在下次隐式服务程序解除队列时由隐式服务程序予以纠正。

在我们描述我们的发明的最佳实施例的时候，不言而喻的是，在现在和将来，那些精通技术的人们可能会做出属于下列权利要求范围的各种改进和提高。这些权利要求应当被认为是对最先公开的发明的正当保护。

说明书附图

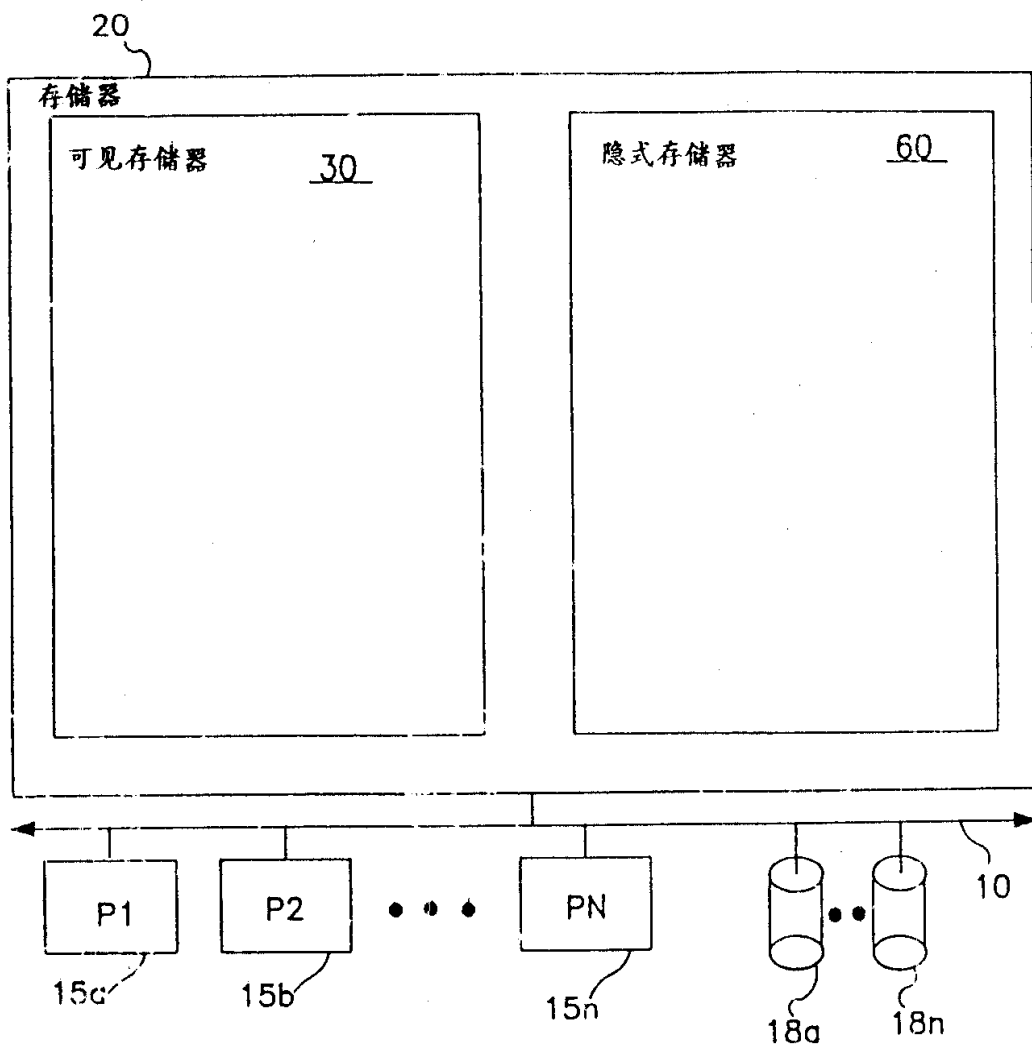


图 1

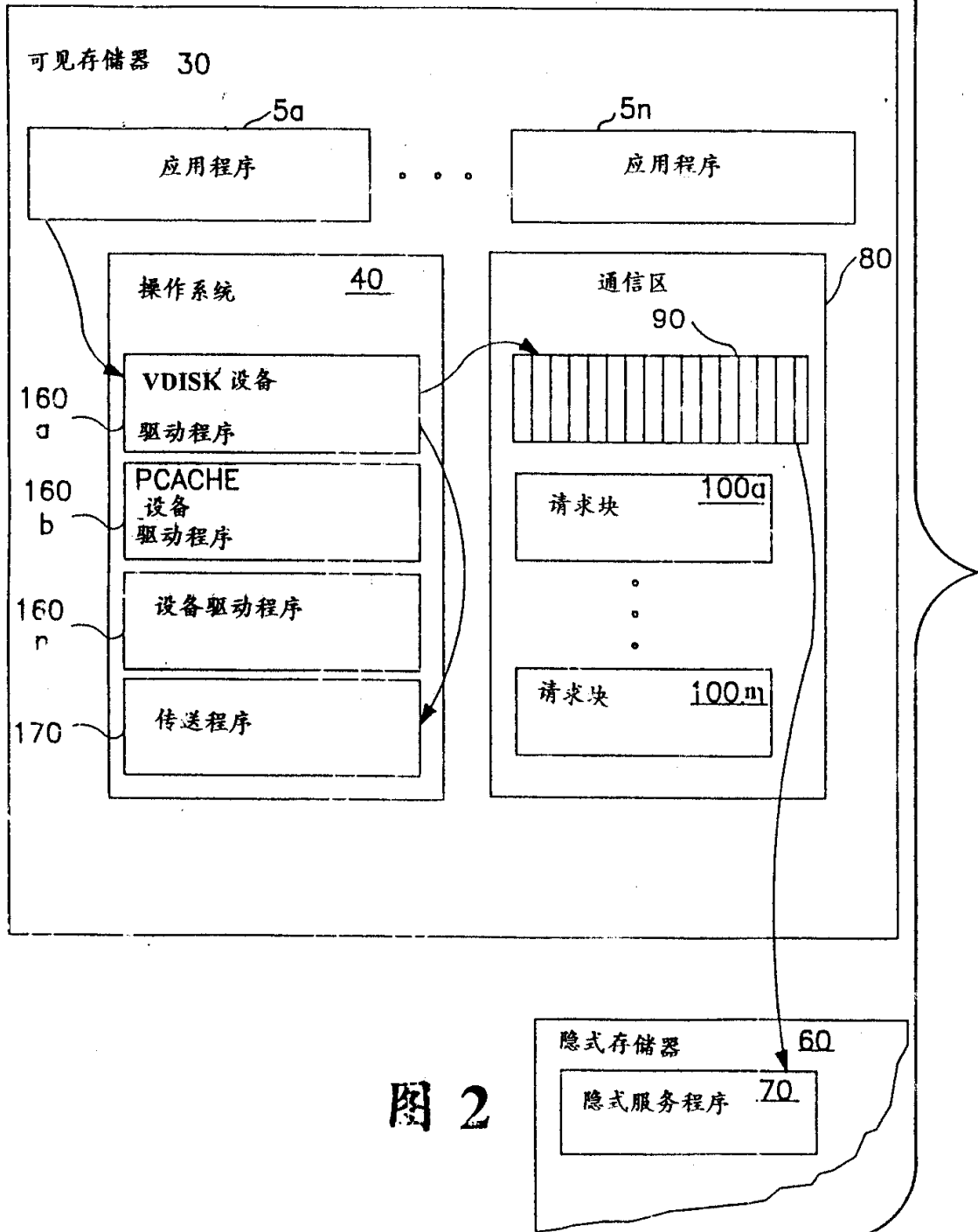


图 2

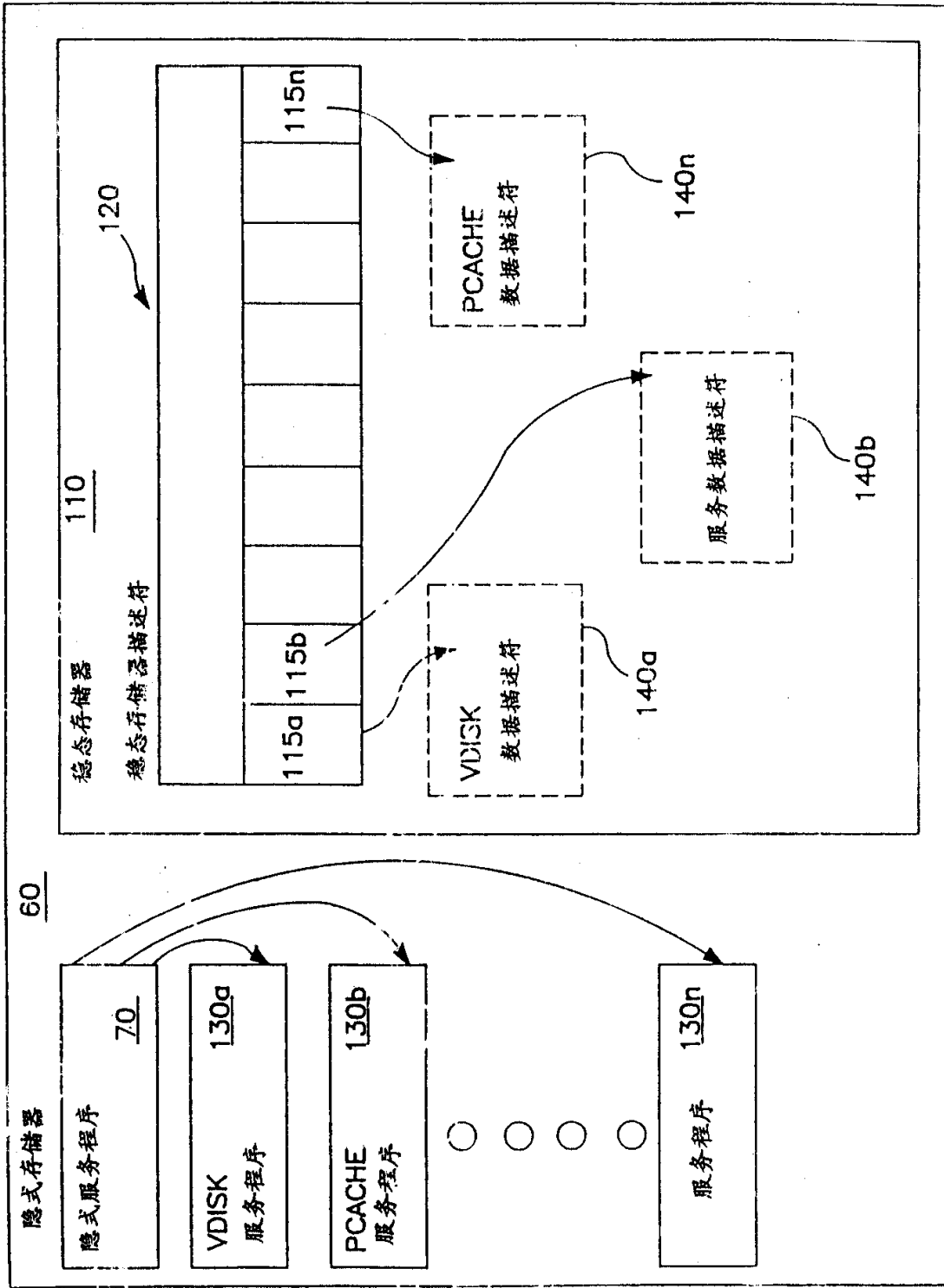


图 3

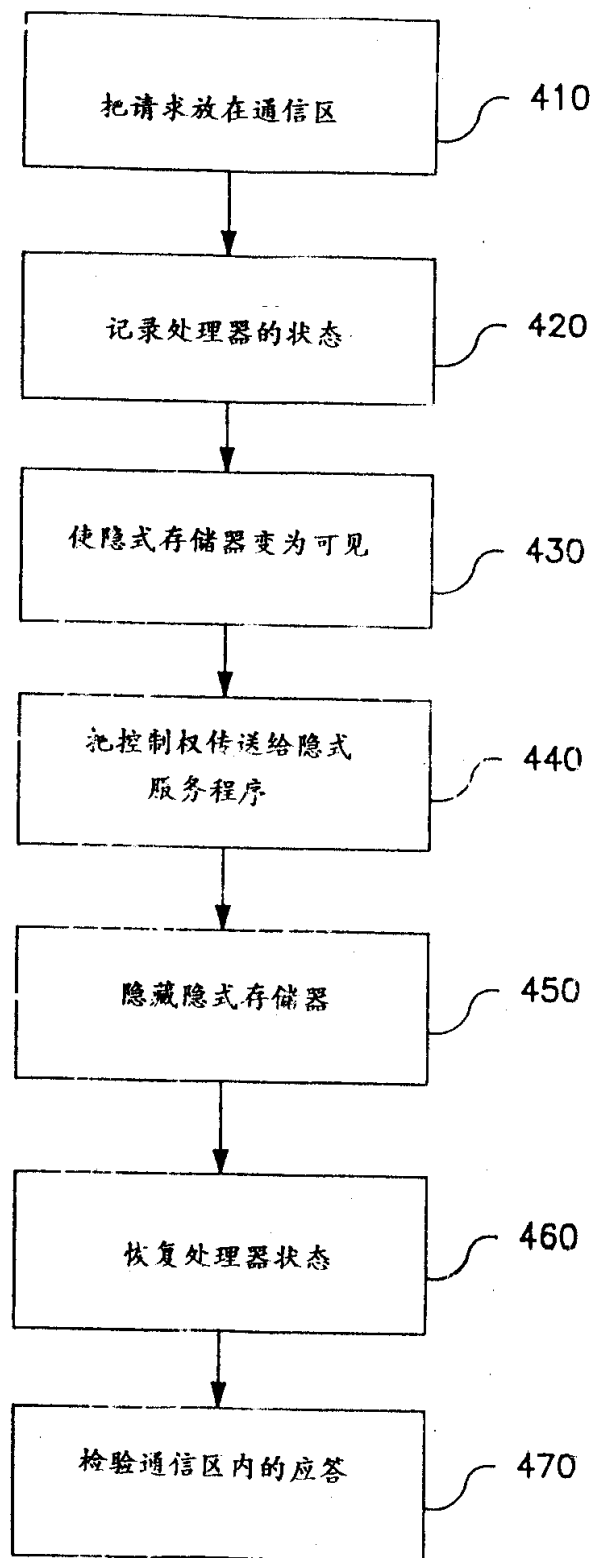


图 4

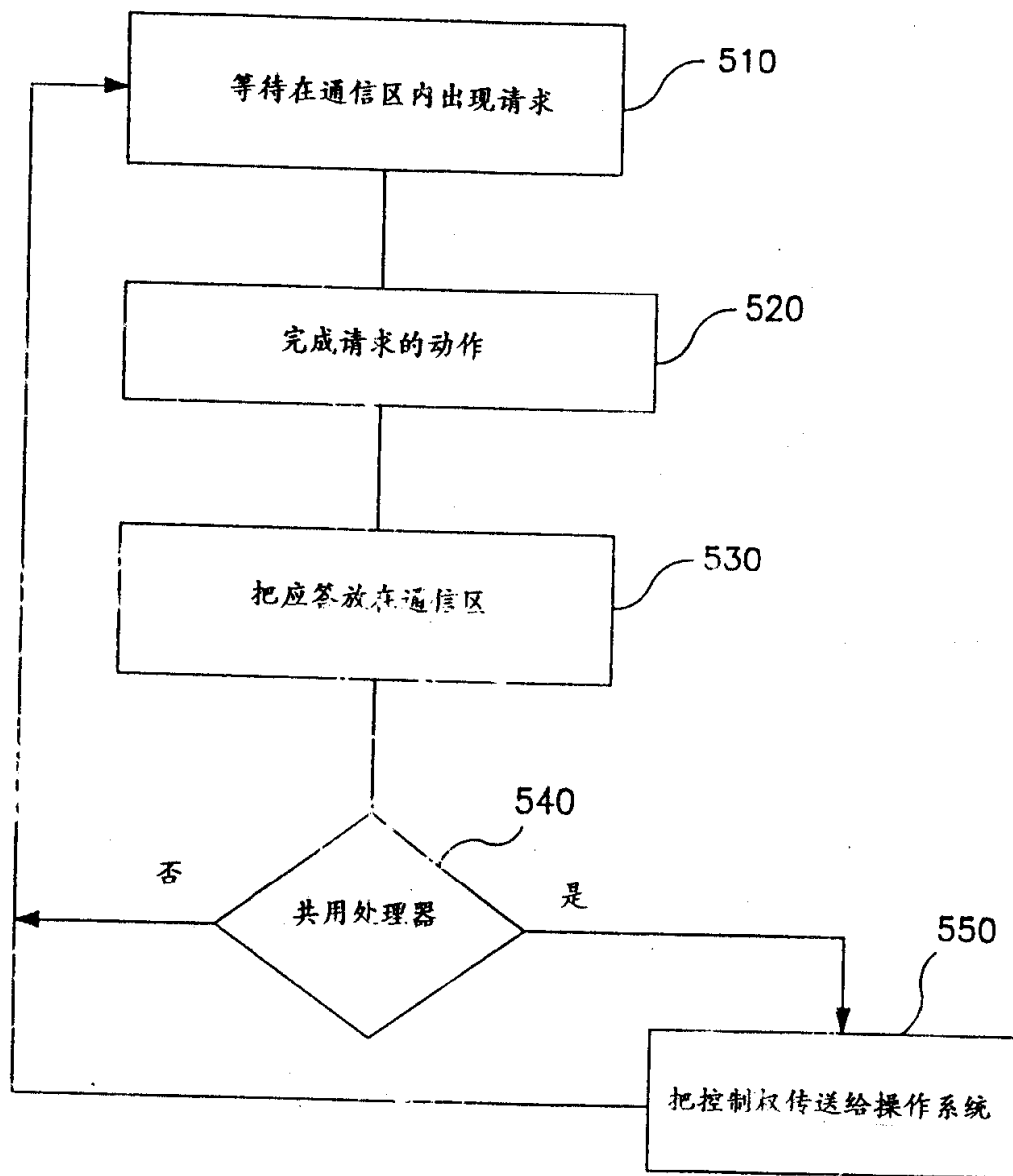


图 5

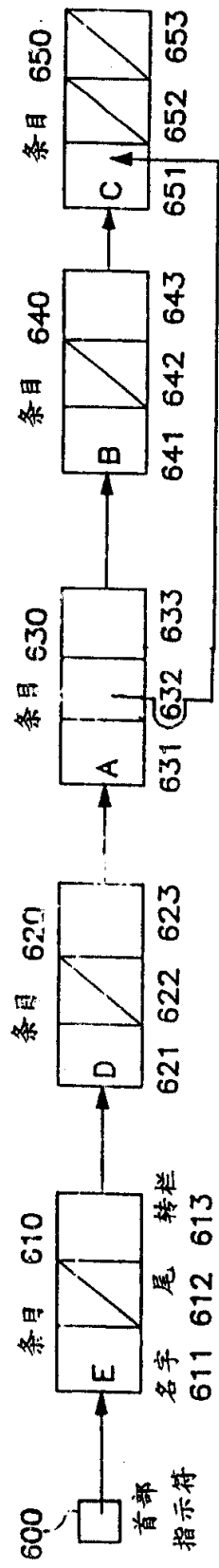


图 6