

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
G06F 13/24 (2006.01)  
G06F 9/46 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200780001843.1

[43] 公开日 2009年2月11日

[11] 公开号 CN 101366012A

[22] 申请日 2007.1.4

[21] 申请号 200780001843.1

[30] 优先权

[32] 2006.1.4 [33] US [31] 60/756,424

[86] 国际申请 PCT/IB2007/050014 2007.1.4

[87] 国际公布 WO2007/077539 英 2007.7.12

[85] 进入国家阶段日期 2008.7.2

[71] 申请人 NXP 股份有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 米林德·马诺哈尔·库尔卡尼

[74] 专利代理机构 中科专利商标代理有限责任公司

代理人 朱进桂

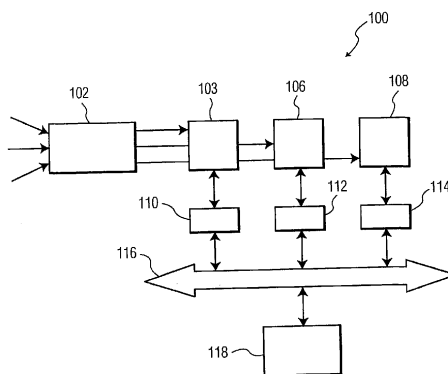
权利要求书 3 页 说明书 8 页 附图 4 页

## [54] 发明名称

用于多处理器系统中的中断分配的方法和系统

## [57] 摘要

一种在多处理器系统中给处理器分配中断负载的方法。该方法包括：利用多个处理器(104、106、108)执行当前事务，其中每个事务都与处理器之一相关联；产生中断请求；估计针对每个处理器的事务完成时间；以及将中断请求定向(102)到具有最小估计事务完成时间的处理器。估计事务完成时间周期性地发生，以存储并连续更新关于事务时间的信息。根据本发明的一个方面，当产生中断请求时，出现估计针对每个处理器的事务完成时间的步骤。根据本发明的另一方面，传送中断请求的步骤包括在估计事务完成时间之前将中断请求传送到中间处理器。



1、一种用于在多处理器系统中给处理器分配中断负载的方法，所述方法包括：

利用第一处理器执行当前事务；  
在当前事务的执行过程中接收中断请求；  
估计针对当前事务的第一事务完成时间；以及  
将中断请求重新定向到第二处理器。

2、根据权利要求1所述的方法，还包括：估计针对在第二处理器上执行的当前事务的第二事务完成时间，并且仅当第二事务完成时间小于第一事务完成时间时，将中断请求重新定向到第二处理器。

3、根据权利要求1所述的方法，还包括：估计针对多处理器系统中的每个处理器的事务完成时间，并将中断请求重新定向到具有最小估计事务完成时间的处理器。

4、根据权利要求3所述的方法，其中，所述处理器是对称处理器，以及所述方法还包括：在处理器之间进行负载平衡，以使得处理器操作于实质上均匀的负载。

5、一种用于在多处理器系统中给处理器分配中断负载的方法，包括：

利用第一处理器执行当前事务；  
在当前事务的执行期间产生中断请求，以及  
将中断请求定向到第一处理器；  
在产生中断请求时估计针对当前事务的第一事务完成时间；以及  
将中断请求重新定向到第二处理器。

6、根据权利要求5所述的方法，还包括：估计针对在第二处理器上执行的当前事务的第二事务完成时间，只有当第二事务完成时间小于第一事务完成时间时，将中断请求重新定向到第二处理器。

7、根据权利要求6所述的方法，还包括：估计针对多处理器系统中的每个处理器的事务完成时间，并将中断请求重新定向到具有最小估计事务完成时间的处理器。

8、根据权利要求 7 所示的方法，其中，所述处理器是对称处理器，以及所述方法包括：在处理器之间进行负载平衡，以使得处理器操作于实质上均匀的负载。

9、一种用于在多处理器系统中给处理器分配中断负载的方法，包括：

利用多个处理器执行当前事务，其中每个事务与处理器之一相关联；

产生中断请求；

估计针对每个处理器的事务完成时间；以及

将中断请求定向到具有最小估计事务完成时间的处理器。

10、根据权利要求 9 所述的方法，其中，所述估计事务完成时间的步骤周期性地出现。

11、根据权利要求 9 所述的方法，其中，当产生中断请求时，估计针对每个处理器的事务完成时间的步骤出现。

12、根据权利要求 9 所述的方法，还包括步骤：在估计事务完成时间之前，将中断请求传送至中间处理器。

13、根据权利要求 9 所述的方法，其中，所述处理器是对称处理器，以及所述方法还包括：在处理器之间进行负载平衡，以使得处理器操作于实质上均匀的负载。

14、一种用于分配中断负载的系统，包括：

多个处理器，每个处理器能够执行事务；

中断分配器，与处理器通信，用于分配中断请求；以及

将所述中断分配器配置为估计针对每个处理器的事务完成时间，以使得当分配每个中断请求时，将中断请求指派至具有最小事务完成时间的处理器。

15、根据权利要求 14 所述的系统，其中，所述处理器是同步的，并操作于实质上恒定的频率。

16、根据权利要求 15 所述的系统，其中，所述处理器定义了处理器核的部分。

17、根据权利要求 15 所述的系统，其中，所述处理器是分立单元，

以及所述系统包括主存储器并将主存储器连接到处理器的总线。

18、根据权利要求 17 所述的系统，还包括：

高速缓冲缓存器，被关联以与每个处理器通信。

19、根据权利要求 14 所述的系统，其中，所述处理器是异步的，并且除了估计针对每个处理器的事务完成时间以外，所述中断分配器还对在每个可用处理器频率下的针对每个处理器的估计事务时间进行制表。

20、根据权利要求 14 所述的系统，其中，所述中断分配器与事务处理程序通信，所述事务处理程序与每个各自的处理器相关联，每个事务处理程序连续监控各自的处理器，以估计事务完成时间。

## 用于多处理器系统中的中断分配的方法和系统

### 技术领域

本发明涉及数据处理方法和系统，更具体地，涉及具有优化的中断处理的多处理器系统。

典型地，数据处理系统包括处理器、存储程序和数据的主存储器、以及暂时存储处理器指令的高速缓冲存储器。这种处理系统能够具有多种命令。可以给处理器或处理器组分派同时运行的多个应用。典型的系统具有调度处理器任务的任务调度器。当出现新任务并且需要重新调度任务时，向处理器传送中断请求。

一些系统采用硬中断调度策略，以使得某些高优先级应用或高优先级任务自动使处理器中断当前任务。尽管这对于单个处理器系统是有用的，但是多处理器系统的优化需要更精确的优先化策略。

一些多处理器使用更灵活的中断调度策略。在这样的一个系统中，将中断请求定向到第一处理器，并且在第一处理器目前操作于高负载的情况，该策略自动使第一处理器将中断请求重新定向到低使用率的第二处理器，从而确保了中断任务被更有效地执行。

中断请求的自动重新定向依赖于中断历史表和处理器历史表的使用，这二者可以一同用于更新中断调度。因此，可以基于处理器历史信息更新中断调度。该系统比较调度表信息、已知的预定标准值和处理器统计信息，以达到对中断调度表的最优修正。通过这种方法，提高了系统吞吐量。在 Nakagawa 的美国专利 No. 6,237,058 中描述了一个示例性系统。

令人遗憾的是，即使已知并利用处理器历史和中断历史，也存在其中可以利用干涉接近完成的任务的方式来调度中断请求的情况。中断接近完成的任务可能导致覆写高速缓冲缓存器的不理想效果，并且可能引入其它系统的等待时间。因此，希望开发修正中断调度和中断调度政策以提高整个系统的性能的更好的方法。

## 发明内容

为了提供基本的理解和高层次的概论，下面给出简化的总结。这个总结不是广泛的概述。它既不旨在识别主要或关键的要素，也不是描述范围。这个总结的唯一目的是给出一些简化形式的概念，作为后面给出的一些优选实施例的更详细描述的前言。另外，这里所使用的章节标题仅为了方便而提供，并不旨在也不应被视为任何形式的限制。

多处理器系统中的优化的系统性能依赖于处理中断请求的方式。优化的系统性能有时要求延迟执行中断请求或将中断请求的执行重新定向到另一个处理器，以完成当前的事务。

在多处理器系统中，通常将处理器任务共享并分成多个事务。根据处理器频率、总线速度和存储器可访问性和速度，事务花费一定的时间。理想情况下，即使当系统面临用于更高优先任务的中断请求时，仍可以完成许多事务以优化系统性能。因此，事务完成时间是一个重要的考虑，当仔细预测、估计或计算了事务完成时间时，可以将其用于产生优化的系统性能。当需要清除或重新分配各种级别的高速缓冲缓存器或主存储器以适应中断请求时，这尤其正确。

一种根据本发明的用于在多处理器中给处理器分配中断负载的方法，包括：利用第一处理器执行当前事务，产生中断请求，以及将中断请求重新定向到第一处理器。该方法包括估计针对当前事务的第一事务完成时间。

根据本发明的一个方面，与每个系统处理器相关联的事务处理程序(handler)估计事务完成时间。事务处理程序连续监控各自的处理器，以最准确且最快地估计事务完成时间。根据一个备选方法，事务处理程序或其它装置在中断请求产生时估计事务完成时间，并将中断请求重新定向到第二处理器。

该方法还包括估计针对在第二处理器上执行的当前事务的第二事务完成时间。如果第二事务完成时间小于第一事务完成时间，则立即将中断请求重新定向到第二处理器。

根据本发明的一个方法，当每个处理器处理事务时，针对每个处

理器，估计事务完成时间。根据这个方法，将中断请求重新定向到具有最小估计事务完成时间的处理器。

尽管本发明的方法特别适用于具有对称处理器的系统，其中该对称处理器利用处理器之间的负载平衡，以使得处理器操作于实质上均匀的负载，但是本发明也可以应用于具有非对称处理器的系统。此外，本发明的以上总结并不旨在代表每个公开的实施例，或者本发明的每个方面。在下面的图示和详细描述中，提供了其它的方面和示例实施例。

考虑到下面结合附图对本发明的各个实施例的详细描述，可以更完全地理解本发明。

## 附图说明

图 1 是根据本发明的系统；

图 2 是根据本发明的系统；

图 3 是示出了根据本发明的方法的流程图；

图 4 是示出了根据本发明的方法的流程图。

## 具体实施方式

尽管本发明经受得起各种修改和备选形式，在附图中以示例的方式示出了其特定实施例，并将加以详细描述。然而，应当理解，本发明并没有将本发明局限于所描述的特定实施例。相反，本发明将覆盖落入所附权利要求所限定的本发明的精神和范围内的所有修改、等同物和备选。

图 1 示出了根据本发明的系统 100。系统 100 包括中断分配器 102、处理器 104、处理器 106、处理器 108、高速缓冲缓存器 110、高速缓冲缓存器 112、高速缓冲缓存器 114、系统总线 116 和主存储器 108。

每个处理器 102、104 和 106 被关联为分别直接与特定的高速存储器 110、112 和 114 通信。系统总线 116 使得处理器 102、104 和 106 能够与系统 100 以及与主存储器 108 通信。

系统 100 用于在多个处理器之间分配中断负载。每个处理器能够

执行事务和任务，采用各自的高速缓冲存储器或主存储器以促进事务执行。中断分配器 102 与处理器 104、106 和 108 通信，以便给处理器分配中断请求。将中断分配器 103 配置为估计每个处理器 102、104 和 106 的事务完成时间，以便在分配每个中断请求时，将中断请求指派到具有最小事务完成时间的处理器。应当意识到，尽管本发明的这个示例性实施例中示出了三个处理器，本发明对于采用多个额外处理器的系统同样有效。

根据本发明的一个方面，处理器 104、106 和 108 是同步的，并被设计为操作于实质上恒定的频率和电压。

根据本发明的备选方面，处理器 104、106 和 108 是异步的，并操作于不同的或可变的电压。除了估计针对每个处理器 104、106 和 108 的事务完成时间以外，中断分配器 102 还对在每个可用处理器频率下的针对每个处理器的估计事务时间进行制表。

根据本发明的另一个方面，处理器 104、106 和 108 定义了处理器核的部分，而且将系统 100 配置在单个装置上。

可以意识到，本发明也可以在多个分立的单元上实现。在这种实现中，处理器是分立的单元，系统包括主存储器和将主存储器连接到处理器的总线。因此，分别将高速缓冲缓存器 110、112 和 114 唯一地指派到处理器 104、106 和 108，并被关联以唯一地与各自的处理器通信。

图 2 示出了根据本发明的系统 200。系统 200 包括处理器 202、204、206 和 208，以及与每个处理器分别关联的事务处理程序 210、212、214 和 216。系统 200 包括第一仲裁器 218、第二仲裁器 222、第三仲裁器 226 和存储器资源 220、224、228、230、232 和中断分配器 234。

事务处理程序 210、212、214 和 216 通过第一仲裁器 218 与存储器资源 220 通信。存储器资源 220 通过第二仲裁器 222 与存储器资源 224 通信。存储器资源 224 与作为主存储模块的存储器资源 228、238 和 232 通信。

系统 200 具有分级存储结构以及包括存储器和总线的多个共享资源。处理器竞争共享资源。处理器 202、204、206 和 208 发出由各个

事务处理程序 210、212、214 和 216 接收到的读和写事务，各个事务处理程序 210、212、214 和 216 被关联以与每个处理器 202、204、206 和 208 通信。事务处理程序 210、212、214 和 216 包括保存各个处理器的事务状态的存储器。根据本发明的一个方面，事务处理程序 210、212、214 和 216 周期性地轮询各自处理器 202、204、206 和 208 的事务状态。

可以通过多种方式确定当前事务完成时间。考虑本示例。根据图 2 中的系统，存储器资源 220 和 224 是高速缓冲缓存器。存储器资源 228 和 230 是 SRAM 或 DRAM 存储器。存储器资源 232 是磁或光驱动。

处理器 202 发出包括要对在 SRAM/DRAM 存储器 228 中的某个地址上执行的读和写指令的事务。与处理器 202 相关联的事务处理程序 216 开始处理事务。

多个处理器 204、206 和 208 共享二级高速缓冲缓存器资源 220。事务处理程序 216 在仲裁总线 218 上进行仲裁，并在特定时段  $T_1$  后访问表示 L2 高速缓存的共享的高速缓冲缓存器资源 220。如果共享的 L2 高速缓冲缓存器资源 220 可以服务于事务，则由于该事务导致高速缓存命中 (cache hit)，事务在具有  $\Delta T_1$  周期的持续时间的额外数目的处理器周期内完成。然而，如果共享的 L2 高速缓冲缓存器资源 220 导致高速缓存失效 (cache miss)，则较低级的存储器必须为事务提供服务。具体地，事务必须由存储必需指令的存储器资源提供服务。在这种情况下，L3/L4 存储器资源 224、228、存储器资源 230、或主存储器 232 必须服务于该事务。如果将事务转发至 L3 存储器 224，则需要具有用于仲裁的  $T_2$  持续时间的额外的处理器周期。因此，L3 高速缓存命中将引起可预计数目的具有  $T_2$  持续时间的处理器周期，而 L3 高速缓存失效将导致必须由 L4 存储器服务于事务的高度不确定性。访问 L4 存储器要求特定数目的具有典型  $T_3$  持续时间的处理器周期。

可以意识到， $T_1$ 、 $T_2$  和  $T_3$  具有统计上可预测的持续时间。根据本发明的一个方面，基于平均的高速缓存访问时间(说明特定的缓存级)、高速缓冲缓存器的速度和事务长度来预测这些时间。根据本发明的另

一方面，可以利用平均时间。根据本发明的另一方面，在每个实例中计算实际的时间。

事务处理程序 210、212、214 和 216 为各自的处理器 202、204、206 和 208 记录事务时间，并且事务处理程序知道事务在哪一阶段被处理。根据本发明的一个方面，事务处理程序轮询每个处理器，并对每个处理器事务的过程进行制表。根据本发明的另一方面，事务处理程序周期性地轮询各自的处理器。具体地，每个事务处理程序都具有计时器，以估计在事务的每个阶段关于每个事务的平均、最小和最大完成时间的信息。基于上述信息，可以估计任何事务的完成时间。

例如，可能存在由第一处理器 208 和第二处理器 210 发出的两个事务。来自第一处理器 208 的第一事务在等待访问 L2 高速缓存，而来自第二处理器 210 的第二事务在等待访问 L3 高速缓存。事务的完成依赖于在高速缓存中是否存在命中。如果在第一和第二事务中都存在命中，那么针对这些事务的完成时间可能相等（假设用于 L2 和 L3 高速缓存的平均仲裁时间相等）。然而，如果两个事务都导致失效，那么如果这二者都由 L4 高速缓存提供服务，则更可能出现第二事务在第一事务之前完成的情况。然而，如果缓存命中信息对于两个事务都是有效的，那么容易将中断指派到导致缓存命中的处理器。

尽管本发明可用于相对于图 2 所描述的结构，然而可以设计其它可能的系统构型来使用本发明。例如，本发明可以被配置为采用不需要高速缓冲缓存器的结构。例如在 IBM Cell 结构中可以观察到这种配置。在这个场景中，甚至可以更准确地预测在每一级别访问存储器的平均时间。

图 3 示出了在多处理器系统中用于给处理器分配中断负载的方法 300。方法 300 包括以下步骤：在第一处理器上执行 302 当前事务，接收 304 中断请求，估计 306 针对当前事务的事务完成时间，将中断请求重新定向到 308 第二处理器，以及处理 310 中断请求。处理 310 中断请求使得接收 304 中断请求能够重复。

执行 302 当前事务包括利用第一处理器处理事务的特定部分。事务具有普遍可定义的平均持续时间。执行 302 包括处理事务的开始，

也可以包括多个处理器周期，但是事务还没有完全完成。

中断分配器 102 在完成事务之前产生中断请求。将中断请求传送到第一处理器，或中间处理器。第一处理器、或中间处理器接收 304 该中断请求，然后估计用于执行 302 第一处理器的当前事务的完成时间。

可以意识到，尽管第一处理器在接收中断请求之后估计 306 完成时间，估计 306 完成时间备选地可以由中间的 i..，由中断分配器、事务处理程序或其它装置实现。步骤 306 的时序可以发生在产生中断请求时、在接收到 304 中断请求时、或在接收到 304 中断请求之后。

根据本发明的一个方面，估计 306 由中断分配器 102 在产生中断请求之前完成。采用预测分析和补充处理器，以使得中断分配器 102 能够预先对针对每个事务的估计的事务完成时间进行制表。

接收 304 中断请求发生在第一处理器当前事务的执行 302 期间，并在当前事务剩下预定或估计数目的周期用于完成，以及在第二处理器具有比这个剩下的用于第二处理器当前事务的预定或估计数目的周期小的周期时，最佳地出现将中断请求重新定向到 308 第二处理器的步骤。

在对称的处理系统中，周期具有固定的持续时间，并且处理器之间出现负载平衡，以使得处理器操作于普遍地、或实质上均匀的负载。估计第二事务完成时间的步骤可以针对在第二处理器上执行的当前事务而发生，并且只有当第二事务完成时间小于第一事务完成时间时，将中断请求重新定向到 308 第二处理器。最佳地，将采用具有多于两个处理器的系统。因此，重新定向 308 将中断请求发送到系统中具有最小估计事务完成时间的可用处理器。

图 4 示出了用于在多处理器系统中给处理器分配中断负载的方法 400。方法 400 包括以下步骤：执行 402 与多个处理器的当前事务，轮询 404 处理器以估计事务完成时间，产生 406 中断请求，将中断请求定向到 408 具有最小估计事务时间的处理器，以及处理 410 中断请求以重复产生 406 中断请求的步骤。

通过多个处理器执行 402 当前事务依赖于将事务预先指派给特定

的处理器。在对称的多处理器系统中，这典型地通过负载平衡技术来实现，在非对称多处理器系统中通过其它协议实现。轮询 404 处理器以估计事务完成时间，消除了任何使用预测分析的需要。轮询可以周期性地实现，响应于中断请求的产生 406、或者对中断请求的预期而实现。

利用中断表跟踪过去的中断请求、并且推断以某种确定性预测中断请求将在何时产生，这些可以完成预期的中断请求。当存在将产生中断请求的确定性的选择度时，初始化轮询 404。理想情况下，通过系统约束调节确定性的选择度，以适应地优化系统性能。

根据本发明的另外一个备选实施例，轮询不是周期性的，而是响应于处理器所接收的中断请求而发生的。在任何情况下，都将中断定向到具有最小估计事务时间的处理器，以优化系统性能。

尽管已经参考数个特定示例实施例描述了本发明，本领域的技术人员将认识到，可以在不背离权利要求中所阐述的本发明的精神和范围的前提下，进行多处修改。

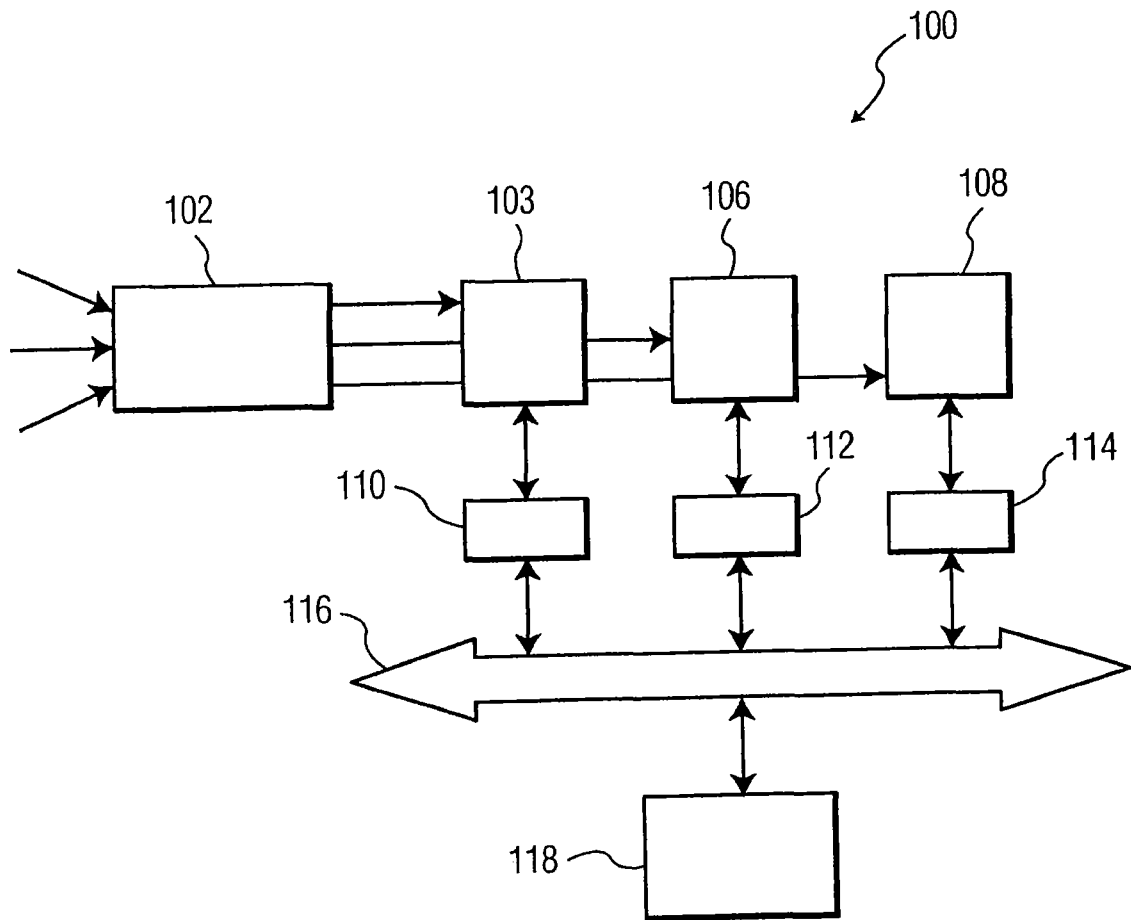


图 1

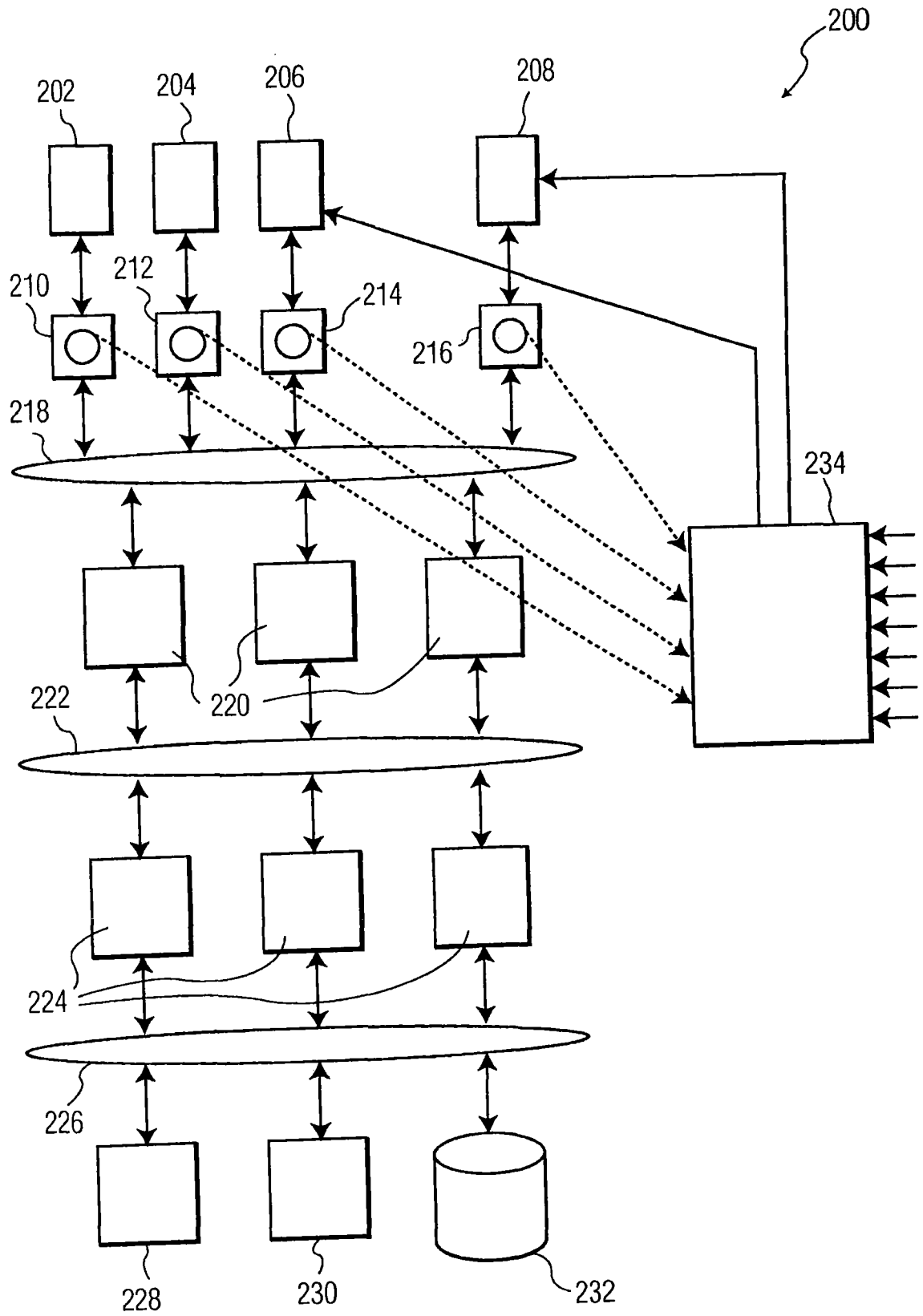


图 2

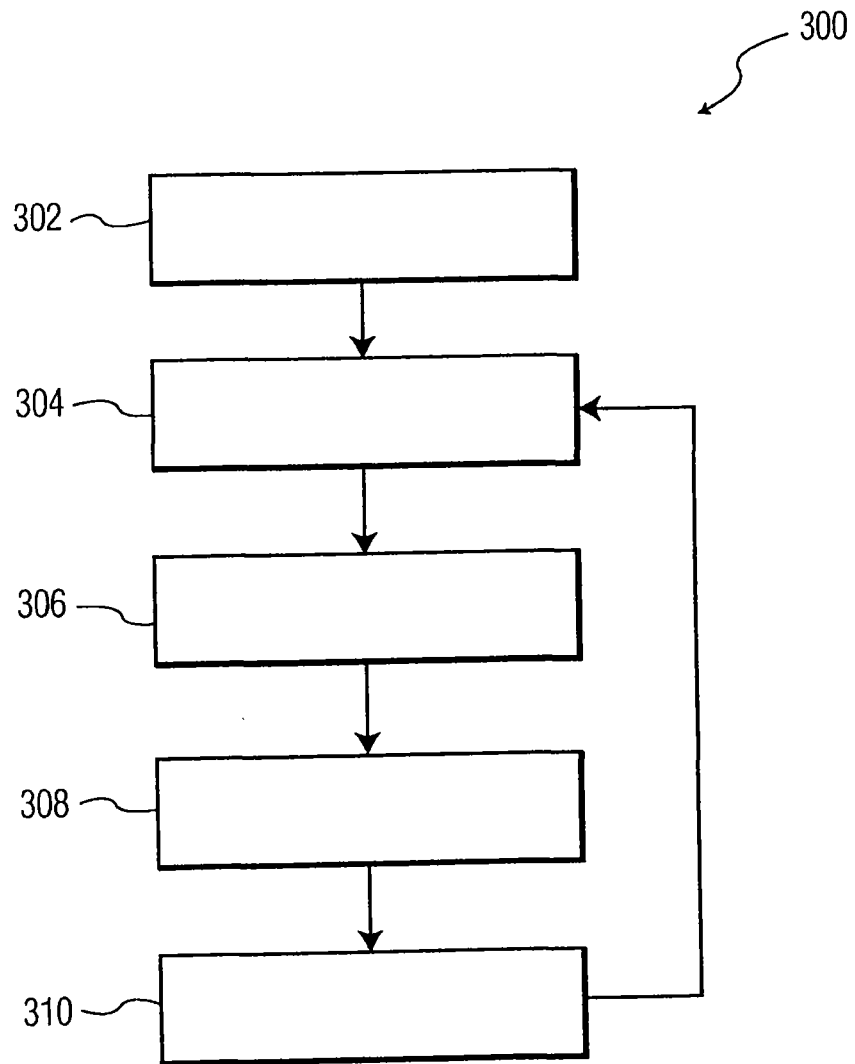


图 3

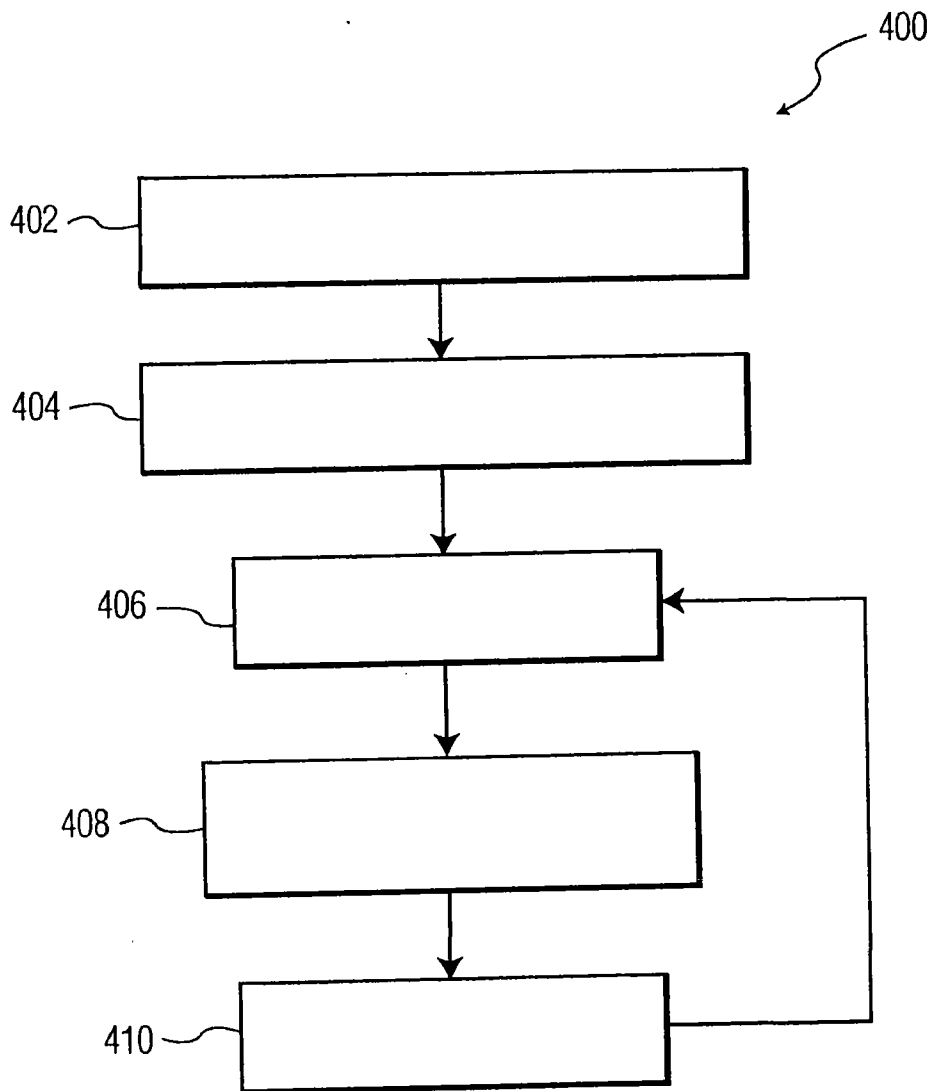


图 4