[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04L 12/54 (2006.01)

[12] 发明专利说明书

专利号 ZL 200410010495.2

[45] 授权公告日 2009 年 9 月 9 日

[11] 授权公告号 CN 100539538C

[22] 申请日 2004.9.29
[21] 申请号 200410010495.2
[30] 优先权
[32] 2003.10.31 [33] US [31] 10/699,315
[73] 专利权人 朗迅科技公司
地址 美国新泽西州
[72] 发明人 比德·J·兹维尔斯
[56] 参考文献
CN1400531A 2003.3.5
US5432908A 1995.7.11
US6363075B1 2002.3.26
审查员 林 姓

[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 李德山

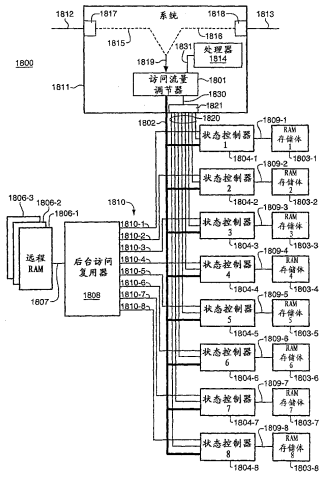
权利要求书 9 页 说明书 33 页 附图 14 页

[54] 发明名称

具有链表处理器的存储器管理系统

[57] 摘要

一种适合于处理链表数据文件的存储器管理系统。该系统具有多个低存储容量高速存储器和低速高存储容量的大容量存储器。一个访问流量调节器生成通过存储器读写链表文件的请求。链表的头部和尾部缓冲区以及任何中间缓冲区都被写入低存储容量高速存储器。中间缓冲区立即从低存储容量高速存储器被传输到上述大容量存储器，同时将链表的头部缓冲区和尾部缓冲区留在低存储容量高速存储器中。在读操作中，从低存储容量高速存储器中读出头部和尾部缓冲区。将中间缓冲区从大容量存储器传输到上述低存储容量高速存储器，并接着从该高速存储器中读出。



1. 一种操作存储器管理系统的方法，所述存储器管理系统适于处理链表数据文件；所述存储器管理系统包括多个低存储容量高速存储器，以及低速高存储容量的大容量存储器，所述低存储容量高速存储器拥有第一数据速率；所述大容量存储器拥有比所述第一数据速率低的第二数据速率，所述存储器管理系统进一步包括访问流量调节器，所述访问流量调节器用来生成通过所述存储器读和写链表的请求，所述方法包括以下步骤:

通过将写请求从所述访问流量调节器传送到所选的低存储容量高速存储器，启动在所述所选的低存储容量高速存储器中链表的写入;

将所述链表的头部缓冲区和尾部缓冲区以及至少一个中间缓冲区写入到所述所选的低存储容量高速存储器中；以及

将所述至少一个中间缓冲区从所述所选的低存储容量高速存储器传输到所述大容量存储器，同时将头部缓冲区和尾部缓冲区留在所述所选的低存储容量高速存储器中。

2. 如权利要求 1 所述的方法，进一步包括步骤:

操作所述存储器管理系统以针对来自所述访问流量调节器的多个请求并行处理链表;

操作所述存储器管理系统以处理存储在所述所选的低存储容量高速存储器的不同的低存储容量高速存储器中的新链表的缓冲区;

操作所述存储器管理系统以写入尾部缓冲区作为所述新链表的头部缓冲区；以及

读取所述新链表的头部缓冲区。

3. 如权利要求 1 所述的方法，进一步包括步骤:

在突发模式中，将链表的中间缓冲区从所述大容量存储器传输到所述所选的低存储容量高速存储器，其中所述突发模式具有基本上等于所述所选的低存储容量高速存储器的数据速率的数据速率;

将读出的缓冲区存储在所述所选的低存储容量高速存储器中;

在突发模式中将所述链表的缓冲区从所述大容量存储器读出到所述所选的低存储容量高速存储器，以传输到所述访问流量调节器；

通过在突发模式中从所述所选的低存储容量高速存储器向所述大容量存储器传输已有链表的已有尾部缓冲区，随后将缓冲区写入所述已有链表；以及

将新缓冲区作为所述已有链表的新尾部缓冲区写入所述所选的低存储容量高速存储器中。

4. 如权利要求 **1** 所述的方法，进一步包括步骤：

并行接收多个链表的缓冲区；

分离被导向所述访问流量调节器的多个链表的缓冲区；

将所述访问流量调节器接收到的多个缓冲区延伸至所选的低存储容量高速存储器；

响应来自所述访问流量调节器的多个接收到的请求中的每一个请求，以确定专用于状态控制器的所选的低存储容量高速存储器的当前占用率水平；

如果没有超过所述当前占用率水平，将所述接收到的缓冲区延伸至所述所选的低存储容量高速存储器；

如果超过所述所选的低存储容量高速存储器的所述当前占用率水平，则用信号通知所述访问流量调节器缓冲所述接收到的访问请求；

控制突发模式下从所述所选的低存储容量高速存储器到所述大容量存储器的缓冲区的传输；

随后控制从所述大容量存储器到所述所选的低存储容量高速存储器的缓冲区的传输；

在请求缓冲区的传输时，确定所述大容量存储器是否空闲；

如果空闲，则将所述缓冲区延伸至所述大容量存储器；以及

如果所述大容量存储器忙，则缓冲所述传输。

5. 如权利要求 **1** 所述的方法，其中所述传输步骤还包括步骤：

确定多个发出请求的低存储容量高速存储器中的哪一个将被准予访问所述大容量存储器；

缓冲其他低存储容量高速存储器的请求；

随后确定将把缓冲区从所述大容量存储器导向到的所选的低存储容量高速存储器的标识；以及

控制突发模式下从所述大容量存储器到所述标识的低存储容量高速存储器的所述缓冲区的传输。

6. 如权利要求1所述的方法，进一步包括以下步骤:

生成对每一个低存储容量高速存储器唯一的信号，该信号指示每一个所述低存储容量高速存储器的忙/空闲状态;

将每一个生成的信号延伸至所述访问流量调节器;

操作所述访问流量调节器以接收通过所述所选的低存储容量高速存储器进行链表的写或读的请求;

响应于请求的接收，操作所述访问流量调节器以读取由所述所选的低存储容量高速存储器生成的所述忙/空闲信号;

响应于所述读取，操作所述访问流量调节器以识别所述所选的低存储容量高速存储器中的空闲的低存储容量高速存储器；以及

操作所述访问流量调节器，以将用于读或写数据文件的请求延伸到所述空闲的所选的低存储容量高速存储器。

7. 一种操作存储器管理系统的方法，所述存储器管理系统适合于处理链表数据文件；所述存储器管理系统包括多个低存储容量高速存储器，以及低速高存储容量的大容量存储器，所述低存储容量高速存储器拥有第一数据速率，所述大容量存储器拥有比所述第一数据速率低的第二数据速率，所述存储器管理系统进一步包括访问流量调节器，所述访问流量调节器用来生成通过所述存储器读和写链表的请求，所述方法包括以下步骤:

将针对特定链表的读请求从所述访问流量调节器传送到所选的低存储容量高速存储器，所述所选的低存储容量高速存储器包括所述特定链表的缓冲区;

读取所述特定链表的头部缓冲区并将所读取的信息发送到所述访问流量调节器;

将所述特定链表的至少一个中间缓冲区从所述大容量存储器传输到所述所选的低存储容量高速存储器;

将传输到所述所选的低存储容量高速存储器的中间缓冲区指定为所述特定链表的替换头部缓冲区;

从所述所选的低存储容量高速存储器中读出所述特定链表的所述中间缓冲区;以及

将所述特定链表的所述读出的中间缓冲区传送到所述访问流量调节器。

8. 如权利要求7所述的方法,进一步包括步骤:

操作所述存储器管理系统以:

针对来自所述访问流量调节器的多个请求并行处理链表;

处理存储在多个所述所选的低存储容量高速存储器中的链表的缓冲区;

写入尾部缓冲区作为新链表的头部缓冲区;以及

读取所述新链表的头部缓冲区。

9. 如权利要求7所述的方法,其中所述存储器管理系统进一步包括多个状态控制器(1804),每一个状态控制器专用于所选的低存储容量高速存储器,所述存储器管理系统进一步包括将所述访问流量调节器连接到所述状态控制器的请求总线(1802);所述传送读请求的步骤包括以下步骤:

操作所述访问流量调节器以选择适于接收所述读请求的所选的低存储容量高速存储器;

将读请求经过所述请求总线从所述访问流量调节器传送到专用于所述所选的低存储容量高速存储器的状态控制器;

操作所述状态控制器以将所述读请求延伸至所述所选的低存储容量高速存储器;

如果所选的低存储容量高速存储器的当前占用率水平不超过预定水平,则操作所述状态控制器将所述读请求传送到所述所选的低存储容量高速存储器;

如果所述所选的低存储容量高速存储器的所述当前占用率水平超过了所述预定水平,则操作所述状态控制器请求到所述大容量存储器的连接;

所述存储器管理系统进一步包括后台访问复用器,以及连接所述状态控制器和所述后台访问复用器的访问总线,所述存储器管理系统进一步包括将所述

后台访问复用器连接到所述大容量存储器的总线，所述传送所述读请求的步骤进一步包括以下步骤:

操作所述后台访问复用器以:

从所述状态控制器接收连接到所述大容量存储器的请求;

确定多个发出请求的状态控制器中的哪一个将被准予访问所述大容量存储器;

将所述发出请求的状态控制器中的一个状态控制器连接到所述大容量存储器;

在从所述所选的低存储容量高速存储器到所述大容量存储器的数据传输过程中，控制所述大容量存储器的操作; 以及

通过所述访问总线将所述链表的缓冲区从所述大容量存储器经过所述状态控制器传输到所述后台访问复用器。

10. 如权利要求 7 所述的方法，其中所述将所述特定链表的至少一个中间缓冲区从所述大容量存储器传输到所述所选的低存储容量高速存储器的步骤包括以下步骤:

在突发模式中，将链表的中间缓冲区从所述大容量存储器传输到所选的低存储容量高速存储器，其中所述突发模式具有基本上等于所述所选的低存储容量高速存储器的数据速率的数据速率;

将所述从所述大容量存储器读出的缓冲区存储在所述所选的低存储容量高速存储器中;

随后从所述所选的低存储容量高速存储器中读出所述链表的缓冲区，以传输到所述访问流量调节器;

随后通过从所述所选的低存储容量高速存储器向所述大容量存储器传输已有链表的已有尾部缓冲区，将缓冲区写入所述已有链表; 以及

随后将新缓冲区作为所述链表的新尾部缓冲区写入所述所选的低存储容量高速存储器中。

11. 如权利要求 9 所述的方法，其中所述操作所述状态控制器的步骤进一步包括以下步骤:

并行接收多个链表的缓冲区;

分离被导向所述访问流量调节器的多个链表的缓冲区;

将所述访问流量调节器接收到的多个缓冲区延伸至所选的低存储容量高速存储器;

响应每一个接收到的访问请求，以确定专用于所述状态控制器的所选的低存储容量高速存储器的当前占用率水平;

如果所述所选的低存储容量高速存储器的所述当前占用率水平没有超过预定水平，将所述接收到的缓冲区延伸至所述所选的低存储容量高速存储器;

如果所述当前占用率水平超过所述预定水平，则用信号通知所述访问流量调节器缓冲所述请求;

控制在突发模式下从所述所选的低存储容量高速存储器到所述大容量存储器的缓冲区的传输，所述突发模式具有基本上等于所述所选的低存储容量高速存储器的数据速率的数据速率;

随后控制从所述大容量存储器到所述所选的低存储容量高速存储器的缓冲区的传输;

随后在请求从所述所选的低存储容量高速存储器到所述大容量存储器的缓冲区的传输时，确定所述大容量存储器是否空闲;

如果空闲，则将所述缓冲区延伸至所述大容量存储器；以及

如果所述大容量存储器忙，则缓冲所述传输。

12. 如权利要求9所述的方法，其中所述操作所述后台访问复用器的步骤进一步包括以下步骤:

确定多个发出请求的低存储容量高速存储器中的哪一个将被准予访问所述大容量存储器;

在所述所选的低存储容量高速存储器中缓冲其他缓冲区的请求;

确定将把缓冲区从所述大容量存储器导向到的所选的低存储容量高速存储器的标识；以及

控制在突发模式下从所述大容量存储器到所述标识的低存储容量高速存储器的所述缓冲区的传输。

13. 如权利要求7所述的方法，进一步包括以下步骤:

生成对每一个所选的低存储容量高速存储器唯一的信号，该信号指示每一个所述所选的低存储容量高速存储器的忙/空闲状态;

将每一个生成的信号延伸至所述访问流量调节器;

操作所述访问流量调节器以接收通过所述所选的低存储容量高速存储器进行链表的写或读的请求;

响应于请求的接收，操作所述访问流量调节器以读取由所述所选的低存储容量高速存储器生成的所述忙/空闲信号;

响应于所述读取，操作所述访问流量调节器以识别所述所选的低存储容量高速存储器中的空闲的所选的低存储容量高速存储器; 以及

操作所述访问流量调节器，以将用于读或写数据文件的请求延伸到所述空闲的所选的低存储容量高速存储器。

14. 一种存储器管理系统（1800），所述存储器管理系统适用于处理链表数据文件; 所述存储器管理系统包括:

多个所选的低存储容量高速存储器（1803-1，1803-2，1803-3，1803-4，1803-5，1803-6，1803-7，1803-8），以及低速高存储容量的大容量存储器（1806-1，1806-2，1806-3）;

所述所选的低存储容量高速存储器拥有第一数据速率，并且所述低速高存储容量的大容量存储器（1806-1，1806-2，1806-3）具有低于所述第一数据速率的第二数据速率;

访问流量调节器（1801），用来生成通过所述存储器（1803，1806）进行链表的读和写的请求;

用于通过将写的请求从所述访问流量调节器发送到所述所选的低存储容量高速存储器（1803-1，1803-2，1803-3，1803-4，1803-5，1803-6，1803-7，1803-8）中的空闲的低存储容量高速存储器，启动所述存储器中链表的写入的装置（1819）;

用于将所述链表的头部缓冲区和尾部缓冲区以及至少一个中间缓冲区写入到所述所选的低存储容量高速存储器（1803-1，1803-2，1803-3，1803-4，1803-5，1803-6，1803-7，1803-8）中的装置（1819）;

用于将所述所选的低存储容量高速存储器（1803）中的特定链表中的所述至少一个中间缓冲区传输到所述低速高存储容量的大容量存储器（1806），同时将所述链表的头部缓冲区和尾部缓冲区留在所述所选的低存储容量高速存储器（1803）中，并将关于所述链表的信息发送到所述访问流量调节器（1801）的装置（1804，1808，1810）；

用于随后将读取所述链表的请求从所述访问流量调节器（1801）传送到所述所选的低存储容量高速存储器（1803）的装置（1802）；

用于在所述所选的低存储容量高速存储器之一中读取所述特定链表的头部缓冲区并将所读取的头部缓冲区发送到所述访问流量调节器的装置（1804）；

用于将所述链表的所述至少一个中间缓冲区从所述低速高存储容量的大容量存储器传输到所述所选的低存储容量高速存储器（1803）的装置（1808，1810，1804）；

用于将传输到所述所选的低存储容量高速存储器的中间缓冲区指定为新的头部缓冲区的装置（1810）；

用于随后从所述所选的低存储容量高速存储器（1803）中读出所述头部缓冲区和所述尾部缓冲区以及所述中间缓冲区的装置（1804）；以及

用于将所述链表的所述读出的缓冲区传送到所述访问流量调节器的装置（1802）。

15. 如权利要求14所述的存储器管理系统，进一步包括：

连接到所述存储器并用于生成对每一个所选的低存储容量高速存储器（1803）唯一的信号的装置（1804），该信号指示每一个所选的低存储容量高速存储器的当前忙/空闲状态；

用于将所述信号延伸至所述访问流量调节器的装置（1802）；

连接到所述访问流量调节器（1801）并用于接收通过所述所选的低存储容量高速存储器进行链表的写或读的请求的装置（1814）；

连接到所述访问流量调节器（1801）并用于响应于所述请求的接收而读取所述忙/空闲信号的装置（1821）；

连接到所述访问流量调节器（1801）并用于响应于所述读取而确定每一个所述所选的低存储容量高速存储器的当前忙/空闲状态的装置（1821）；以及

连接到所述访问流量调节器（1801）并用于响应于确定所述低存储容量高速存储器之一当前空闲而准予通过所述所选的低存储容量高速存储器进行链表的读或写的请求的装置（1821，1804）。

具有链表处理器的存储器管理系统

技术领域

本发明涉及用于通信网络的存储器管理设备，特别是通过在网络节点减少阻塞而用来优化网络通信量服务能力的设备。本发明进一步涉及存储器管理设备，它通过减少访问网络存储器单元的争用时间来改善传输流。本发明进一步涉及一个用于通过将信息的存储分布在小存储容量、高速度的存储器和低速高存储容量的大容量存储器之间而减少处理和争用时间的方案。

背景技术

主动地管理多节点通信网络以改善网络通信量服务能力，是众所周知的。网络被设计成在每一个节点上都具有足够的设备，以充分地服务预期的通信量。这包括提供用来服务正常通信量所需要的设备，以及用来在经济上可行的范围内服务偶尔出现的峰值通信量所需要的附加设备。通信网络通常并没有被设计成提供服务通信量峰值所需要的设备数量，其中该通信量峰值在理论上是可能的，但是如果有的话，也很少遇到。

多节点通信网络可能遇到通信量阻塞，即使网络在总体上被设计成可以服务足够的通信量水平。这个阻塞是因为通信量的不等分布，其中网络节点中的一些，但不是全部，因为过度的通信量而过载。如果一个网络节点是网络连接请求被导向的目标节点，它就可能会过载。如果一个网络节点通过一个链路连接到所请求的目标节点，并且接收来自目标节点、被导向上游节点的请求，那么它也可能会过载。众所周知，可以为网络配备通信量控制设备，以最小化由不等通信量分布所带来的节点过载。这些通信量控制节点监视每一个节点上的通信量，同时也监视由每个节点生成的连接请求。通过阻塞一些可因发送节点要访问已经过载的远端节点而产生的请求，避免了在远端节点上的信息拥挤问题。

多节点网络和它们的通信量控制设备允许网络以在符合要求的低阻塞水平服务正常的通信量水平。然而，管理和控制网络通信量所需的设备是复杂、昂贵的，并且由于所需的处理操作的复杂性，它们减少了网络的通信吞吐量。这

些设备包括，在节点的输入和输出处提供的由处理器控制的链表引擎，用以缓冲输入以及输出的通信量。链表引擎的操作要求有复杂的数据处理操作，它是使在链表引擎内争用问题最小化所需要的。这些争用问题的复杂性减小了整个网络的通信量服务能力。

当遇到繁重通信量的时候，每个节点处的链表缓冲区的耗尽会引起过载。这会导致数据包被丢弃，因此系统的性能会严重降低。缓冲过载是由于缓冲区的容量不足而引起，或是由于以速度不足的缓冲区处理输入通信量而引起的。系统设计者已经面对两种选择：使用低速但是大的缓冲区，或快速但是容量小的缓冲区。低速但大的缓冲区因引起丢失数据包而阻碍网络通信量。快速但小容量的缓冲区同样会由于在繁重的突发通信量期间缺乏可用缓冲区而引起缓冲区过载以及数据包的丢弃。

与两种类型的缓冲区相关的根本问题的原因是当遇到要使用相同设备的多个访问时而发生的争用问题。例如，当接收到要求读或写一个特定的存储体的多个访问时，争用就会发生。在这样的情况下，其中一个访问胜出，而其它访问等待所请求的存储体变为可用。与获胜访问相关的呼叫就被充分地服务；那些与延迟的访问相关的呼叫不是被丢弃，就是被不充分地服务。

对存储体的访问争用是因为使用的RAM存储体的数量不充足，以及/或是因为被提供用来服务于存储体的争用设备。有些争用设备依赖限制速率的算法和处理，其中在该速率下访问能够被服务。一个这样的现有技术装置利用一个算法，该算法在访问之间要求有大约 250 纳秒的最小时间延迟。这是一个重要的限制，因为它并没有规定要确定第二个 RAM 存储体是否可在访问被分配到第一个存储体之后接收该访问。因此，在服务各访问之间有 250 纳秒的时间间隔的情况下，系统的吞吐量被限制为一秒内最多服务 4，000，000 个访问，这里并不考虑 RAM 存储体的可用性。与已有争用装置有关的另一个问题是，它们中的许多都利用逻辑设备，而这些逻辑设备都很复杂、昂贵，并且不能充分地满足高通信量水平。

发明内容

本发明依照第一个示例性的实施例，克服了这些争用问题，该实施例提供了更多数目的 RAM 存储体。这本身就减少了争用的可能性。本发明提供的第二个特性是为每一个RAM存储体提供一个相关的控制单元，称作状态控制器。

该状态控制器是在它的 RAM 存储体和系统总线之间的一个接口，其中通过该系统总线而接收访问请求。所有的访问请求都由一个访问流量调节器提供到系统总线，其中访问流量调节器接收由一个节点生成的所有访问请求，判断一个RAM 存储体是否可用于满足访问请求，在指定 RAM 存储体当前为忙的情况下缓冲该访问请求，并且在空闲时间将访问请求提供到与该 RAM 存储体相关的状态控制器。如果 RAM 存储体为忙，则状态控制器提供一个信号给访问流量调节器，以指示它的 RAM 存储体当前正忙于为另一个访问服务，并且暂时不能为更多访问请求提供服务。

在写访问请求的情况下，一个访问流量调节器在企图将访问请求路由到 RAM 存储体时扫描所有的状态控制器。在进行这样的扫描时，它立即绕过正在为它们的相关 RAM 存储体(没有足够的存储空间)生成忙信号的状态控制器。访问流量调节器绕过忙或是完全消耗的 RAM 存储体以及它们的状态控制器，并将访问请求导向空闲的 RAM 存储体，其中上述空闲的存储体拥有可用于存储空间的缓冲区。

RAM 存储体的存储器是这样一种类型，它具有高速但相对低的存储容量。每一个 RAM 存储体都能快速地处理每一个导向到它的访问请求。在它的存取周期完成时，它的状态控制器将针对访问流量调节器、指示它的 RAM 的状态的忙信号移除。一旦忙信号被移除，访问流量调节器就知道 RAM 存储体现在可用来满足新的访问请求。

本发明的另一个可行的示例性的实施例是一个 RAM 存储体忙信号的使用操作，其中忙信号只在 RAM 存储体为忙的短暂时间间隔内持续。这个信号的提供构成了有利的争用方案，其能够以一定的速率服务于访问请求，其中该速率只受采用 RAM 存储体的 RAM 设备的速度的限制。这个争用方案是对现有技术方案的一大改善，其中能服务于访问的速率受强制时间间隔的限制，或受所提供的争用逻辑的复杂度的限制。

通过使用采用本发明的争用设备，动态 RAM 存储体设备可以操作的最大速率不是受争用方案所固有的任意限制的制约，而是只受所利用的 RAM 设备的速度的限制。采用本发明的动态高速 RAM 存储体设备可以以一种流水线方式来操作，以为以光纤传输设备的总线速率到达的数据包服务。本发明提供的争用方案提高了在高通信量水平的服务期间由处理器控制的链表引擎所能服务

于输入和输出通信量、同时争用最小的速率。

方面

该发明的一个方面是操作存储器管理系统的方法，所述管理系统适于处理链表数据文件；上述系统包括多个低存储容量高速存储器，以及低速高存储容量的大容量存储器，上述高速存储器拥有第一数据速率，上述大容量存储器拥有比上述第一数据速率低的第二数据速率，上述系统进一步包括访问流量调节器，用来生成通过上述存储器读、写链表的请求，上述方法包括以下步骤：

通过将写请求从上述访问流量调节器传送到上述高速存储器，启动在上述高速存储器中链表的写入；

将上述链表的头部缓冲区和尾部缓冲区，以及至少一个中间缓冲区写入上述高速存储器；以及

将上述至少一个中间缓冲区从上述高速存储器传输到上述大容量存储器，同时将头部缓冲区和尾部缓冲区留在上述高速存储器中。

本发明的另一个方面是操作一个存储器管理系统的方法，所述管理系统适合于处理链表数据文件；上述系统包括多个低存储容量高速存储器，以及低速高存储容量的大容量存储器，上述高速存储器拥有第一数据速率，上述大容量存储器拥有比上述第一数据速率低的第二数据速率，上述系统进一步包括访问流量调节器，用来生成通过上述存储器读、写链表的请求，上述方法包括以下步骤：

将针对特定链表的读请求从上述访问流量调节器传送到上述高速存储器，该存储器包括上述特定链表的缓冲区；

读取上述特定链表的头部缓冲区；

从上述大容量存储器将所述链表的所述至少一个中间缓冲区传送到上述高速存储器之一；

将传送到上述一个高速存储器的中间缓冲区指定为上述特定链表的替换头部缓冲区；

从上述一个高速存储器中读出上述特定链表的上述中间缓冲区；以及

将上述特定链表的上述读出缓冲区传送到上述访问流量调节器。

该方法进一步优选地包括以下步骤：

操作上述系统以针对来自上述访问流量调节器的多个请求并行处理链表；

操作上述系统以处理存储在上述高速存储器的不同存储器中的链表的缓冲区；

操作上述系统以将一个尾部缓冲区作为第一缓冲区写入新链表；以及

首先读取链表的头部缓冲区。

上述系统进一步优选地包括多个状态控制器，每一个状态控制器对于上述高速存储器中的相应一个存储器来说是专用的，上述系统进一步包括将上述访问流量调节器连接到上述状态控制器的请求总线；上述传送读请求的步骤包括以下步骤：

操作上述访问流量调节器以选择空闲高速存储器，它将来接收上述读请求；

将上述读请求经过上述请求总线从上述访问流量调节器传送到专用于所述选择的高速存储器的状态控制器；

操作上述状态控制器以确定所述选择的高速存储器的当前占用率水平；

如果上述当前占用率水平不超过预定水平，则将上述请求传送到上述高速存储器；以及

如果上述选择的高速存储器的上述当前占用率水平超过了上述预定水平，则请求到上述大容量存储器的连接；

上述系统进一步包括后台访问复用器，以及连接上述状态控制器和上述复用器的访问总线，上述系统进一步包括一总线，它将复用器连接到上述大容量存储器，上述方法包括操作上述复用器以执行以下操作的步骤：

从上述状态控制器接收针对到上述大容量存储器的连接的请求；

确定多个发出请求的状态控制器中的哪一个将被准予访问上述大容量存储器；

将一个上述发出请求的状态控制器连接到上述大容量存储器；

在从上述一个高速存储器到上述大容量存储器的数据传输过程中，控制上述大容量存储器的操作；以及

通过上述访问总线将上述链表的缓冲区从上述状态控制器传输到上述复用器。

优选地，从上述大容量存储器传送上述缓冲区的上述步骤包括以下步骤：

在突发模式中，将链表的中间缓冲区从上述大容量存储器传送到上述高速存储器，其中突发模式具有基本上等于上述高速存储器的数据速率的数据速率；

将上述读出的缓冲区存储在上述高速存储器中；以及

随后从上述高速存储器中读出上述链表的缓冲区，以传送到上述访问流量调节器；

通过从上述高速存储器向上述大容量存储器传送已有链表的已有尾部，将缓冲区写入所述已有链表；以及

将新缓冲区作为上述已有链表的新尾部缓冲区写入上述高速存储器。

优选地，操作状态控制器的上述步骤进一步包括以下步骤：

并行接收多个链表的缓冲区；

分离被导向上述访问流量调节器的多个链表的缓冲区；以及

将由上述访问流量调节器接收的多个访问延伸至上述高速存储器；

响应每一个从上述访问流量调节器接收的请求，以确定专用于上述状态控制器的高速存储器的当前占用水平；

如果没有超过上述当前占用水平，将上述访问延伸至上述相关的高速存储器；

如果超过当前占用水平，则用信号通知上述访问流量调节器缓冲上述请求；

控制突发模式下从上述高速存储器到上述大容量存储器的缓冲区的传输；

控制从上述大容量存储器到上述高速存储器的缓冲区的传输；

在请求一个传输时，确定上述大容量存储器是否空闲；

如果空闲，则将上述缓冲区延伸至上述大容量存储器；以及

如果上述大容量存储器忙，则缓冲上述传输。

优选地，操作上述复用器的上述步骤进一步包括以下步骤：

确定多个发出请求的高速存储器中的哪一个将被准予访问上述大容量存储器；

在上述一个高速存储器中将其他缓冲区的请求缓冲；

确定将把一个缓冲区从上述大容量存储器传输到的高速存储器的标识；以及

控制突发模式下从大容量存储器到上述标识的高速存储器的上述缓冲区的传输。

优选地，上述方法进一步包括下列步骤：

生成对每一个高速存储器唯一的信号，该信号指示每一个上述高速存储器

的忙/空闲状态；

将每一个生成的信号延伸至上述访问流量调节器；

操作上述访问流量调节器以接收通过上述高速存储器进行链表写或读的请求；

响应请求的接收，操作上述访问流量调节器以读取由上述高速存储器生成的上述忙/空闲信号；

响应上述读操作，操作上述访问流量调节器以识别上述高速存储器中的空闲存储器；以及

操作上述访问流量调节器，以将用于读、写数据文件的请求延伸到上述一个空闲的高速存储器。

该发明的另一个方面包括一个存储器管理系统，该系统适用于处理链表数据文件；上述系统包括：

多个高速低存储容量存储器，以及低速高存储容量的大容量存储器，上述高速存储器拥有第一数据速率，上述大容量存储器具有低于上述第一数据速率的第二数据速率；

访问流量调节器，用来生成通过上述存储器进行链表读、写的请求；

用于通过发送读写请求到上述高速存储器中的空闲存储器，从而启动上述存储器中链表的写入的装置；

用于将上述链表的头部缓冲区和尾部缓冲区，以及至少一个中间缓冲区写入上述高速存储器的装置；

用于将上述链表的上述至少一个中间缓冲区从上述高速存储器传输到上述大容量存储器，同时将上述链表的头部缓冲区和尾部缓冲区留在上述高速存储器中的装置；

用于随后将读取上述链表的请求从上述访问流量调节器发送到上述高速存储器的装置；

用于在上述高速存储器之一中读取上述链表的头部缓冲区的装置；

用于将上述链表的上述至少一个中间缓冲区从上述大容量存储器传输到上述高速存储器的装置；

用于在上述高速存储器中将所传输的缓冲区指定为新的头部缓冲区的装置；

用于随后从上述高速存储器中读出上述头部缓冲区和上述尾部缓冲区，以及上述中间缓冲区的装置；以及

用于将上述链表的上述读出缓冲区发送到上述访问流量调节器的装置。

优选地，上述存储器管理系统进一步包括：

包括上述高速存储器、用于生成对每一个高速存储器唯一的信号的装置，该信号指示每一个高速存储器的当前忙/空闲状态；

用于将所述信号延伸至上述访问流量调节器的装置；

包括上述访问流量调节器、用于接收通过上述高速存储器进行链表写或读的请求的装置；

包括上述访问流量调节器、用于响应上述请求的接收而读取上述忙/空闲信号的装置；

包括上述访问流量调节器、用于响应上述读取而确定每一个上述高速存储器的当前忙/空闲状态的装置；以及

包括上述访问流量调节器、用于响应确定上述存储器之一当前空闲而准予通过上述高速存储器进行链表的读或写的请求的装置。

附图说明

通过结合图例来阅读详细的说明，可以更好的理解该发明的这些以及其他方面，其中：

图 1 公开了一个多节点网络。

图 2 公开了组成一个节点的硬件单元。

图 3 通过分块公开了图 1 的多节点网络。

图 4 到 6 公开了链表缓冲区的结构。

图 7 公开了在图 3 的节点上的假设通信量情况。

图 8 以及图 9 公开了由多个 RAM 存储体所服务的网络对访问请求的处理。

图 10 公开了连接到一个控制总线和一个数据总线的四个 RAM 存储体。

图 11 的时序图说明了图 10 的 RAM 存储体服务于如图 8 和 9 所示的访问请求的过程。

图 12 公开了由链表处理器控制的存储器系统，它有四个 RAM 存储体。

图 13 的时序图示出了图 12 的系统的操作。

图 14 的另一个时序图示出了图 12 的系统的可选操作。

图15公开了由处理器控制的链表处理系统，它有八个RAM存储体。

图16的时序图示出了图15的系统的操作。

图17公开了图18的状态控制器1804的单元。

图18公开了处理器控制的RAM存储器系统，它采用本发明。

图19公开了根据本发明的链表缓冲区的结构。

图20公开了一个读操作，用于说明作为链表的新头部的缓冲区的创建。

图21的时序图示出了图18的系统的操作。

图22到25是示出了本发明操作的流程图。

具体实施方式

图1的说明

本发明包括一个增强的存储器接口，用来提高多节点通信网络的通信吞吐率。这个增强的存储器接口采用通信量控制单元，它控制打包信息到如图1所示的通信网络的释放。图1的网络具有互连交换单元，术语叫节点，它们相互通信。将节点指定为A、B、C、D、E、F和G，并且由单独的链路连接，其中链路被指定为链路1到链路8。图1的节点定义了一个网络，它将通信量从进站端口分布到出站端口。

图2和3的说明

图2公开了实施图1节点的设备。图1公开了一个互连节点A和节点B的单路径（链路1）。图1中的每一个节点都与另一节点通过输入链路和输出链路连接。节点A通过自身的输入链路接收来自节点B的通信量，以及通过节点A的输出链路向节点B传送通信量。

图2公开了每一个节点的细节，包括它的输入链路和输出链路。路径218是其节点的输出链路；路径223是其节点的输入链路。图2显示了定义节点的设备，并且在图2的左半部，该节点包括输入链路223、分路器222和路径221，路径221将每一个分路器与多个端口201-205的每一个互连。端口201在它的右侧连接到一个出站链路218，其延伸至另一个节点。图2左侧所示的设备包括出站端口201到205，它们允许节点连接到链路218，而链路218则通过五个出站端口201到205延伸至五个不同的节点。

图2的右侧更详细地说明了实施出站端口202的设备。端口202包括多个链表队列215，专用于每一个队列的控制逻辑1800。随后在图18中详细阐述控

制逻辑，它包含处理链表信息并且将它延伸至复用器 213 所需要的设备。五个队列 215 分别由控制逻辑单元 1800 中的相应单元服务，其中控制逻辑单元 1800 通过路径 231 连接到复用器 213。复用器 213 将路径 231 连接到输出链路 218，它会对应于图 1 的链路 1 到 8 的任意一个。图 2 的节点具有使性能最优化的输出排队。对于在任意时刻的每一个出站端口 201 到 205，复用器 213 选择队列 215 中的一个，并且来自队列 215 的数据包被发送至出站链路 218。针对给定端口的队列 1800 的选择取决于网络使用的通信量控制算法。假设图 1 的若干节点想要与例如节点 A 的一个节点所服务的小集团通信。当通信业务通过到节点 A 的链路争用节点 A 的资源时，过载就会发生。包被缓冲，以在起作用的链路得到服务时完成传输。

这种高通信量的情况如图 3 详细所示，其中节点 A 为所需要的节点，并且变暗的链路 1、2、3、4、5 以及 6 代表服务于到所需要的节点 A 的可能电路路径的链路。

节点 B 负担四个节点的通信量：它自己，以及三个节点（C，E 以及 F），它们提供遍历节点 B，和或节点 C 以到达节点 A 的通信量。节点 B 只有一个链路（链路 1）到节点 A。节点 G 和 D 可以以满载链路容量提供通信量。因此链路 1 过载。在高争用的时间内，节点 B 必须缓冲去往节点 A 的通信量。随着争用逐步变弱，因而发送通信量。

如果高争用的间隔持续足够长的时间，节点 B 的缓冲区会溢出。为了防止它发生，通信量控制算法调整通信量到链路的释放，以允许整个网络的缓冲区用尽，并且准备好吸收去往所需要的节点的更多通信量。例如，节点 E 和 F 可能不将通信量分别释放到链路 5 和 6，即使它们可能有足够的包来填充这些链路。会花较长的时间来发送这些包，但是在此过程中节点 B 会避免过载，并且网络会丢弃较少的包。通信量控制可以看作前摄(pro-active)流控。

通信量控制要求高性能的缓冲才能成功。这种缓冲区依赖硬件链表处理器。硬件链表处理器可以有效地利用存储器，动态地分配缓冲区给输入的包，以及在缓冲区所保持的数据成功传输到出站链路之后，回收缓冲区。

图 4 到 6 的说明

根据本发明的链表缓冲区被用于缓冲信息。在一个已初始化的系统中，所有的存储器都被划分成一般缓冲区(generic buffer)。每一个缓冲区都有用于该缓

冲区存储的内容401的空间，以及指向下一个缓冲区的指针402。如图4所示。

在初始化时，通过设置前一个缓冲区的指针字段指向下一个缓冲区的地址，所有的缓冲区都被链路在一起。这叫做自由表(free list)，如图5所示。

通信系统将信息逐个填入这些一般缓冲区，然后将填充的缓冲区链接到一个队列，该队列存储了用于某种特定功能的信息。在系统初始化之后，所有的队列为空。它们的队列长度为零，以及它们的头部和尾部指向 NULL。当信息到达图 6 的特定队列时，一个一般缓冲区从该自由表中取出，被填入信息，然后加入到队列列表中。尾指针所指向的地址改变到队列中所增加单元的地址，然后递增队列长度计数。当要从一个队列中读出信息时，读取该队列的列表的头部的缓冲区内容，并且该列表的头部被移动到该列表中的下一个缓冲区。队列长度也被递减。

图6的队列A有一个带有内容Q的头部缓冲区以及一个带有内容Z的尾部缓冲区，队列B有一个带有内容N的头部缓冲区以及一个带有内容G的尾部缓冲区，而队列C有一个缓冲区，它的头部缓冲区和尾部缓冲区都带有内容HH。这个系统有十一个缓冲区以及三个队列。链表缓冲系统的一个关键特征是缓冲区分配是完全动态的。只要自由表没有被清空，则任何到队列的缓冲区分布都是允许的。例如，队列 A 可以拥有所有十一个缓冲区，过一段时间后，队列A拥有四个缓冲区，队列B拥有四个缓冲区，并且队列C拥有三个缓冲区。一段时间后，所有的队列都可能为空，并且所有的缓冲区可能再次在自由表中。

链表缓冲特别适合通信应用，因为一个给定的设备在满载运行时，到达的信息以恒定的速率消耗缓冲区。但是，该设备必须经常复用与多于一个的流动(flow)或流(stream)相关的信息。链表是一个有效的办法，来解复用并且组织进站的信息，以将它们进行处理并随后复用到出站传输介质上。在一个经过处理的缓冲区传送之后，它可以通过自由表而循环使用。

假设集聚的进站至图2的节点B的通信量每秒消耗十个缓冲区，但是节点B 也每秒传送十个缓冲区。这个节点就达到平衡，因为缓冲区清空速度与填充速度一样。从缓冲的立场看，信息流是进入还是离开都没有关系。例如，如果从节点C到节点B的进站通信量是去往节点G或节点A，这并没有什么关系。在一个瞬时，将链路1馈送到节点A的队列可以被填充，并且将该链路馈送到节点G的队列可以为空，然后一段时间之后，第一组可以为空而第二组可以填

满。只要整体的流和缓冲区容量相匹配，就保持系统的完整性。然而，虽然整体通信量以一个固定的最大速率到达和离开介质，然而在支持的通信介质中，可以复用很多流。缓冲区的灵活管理减轻了瞬时设备状况。

在通信量控制情况中，链表缓冲区是很有用的。每一个节点都支持某个整体吞吐率。通信量控制的实现逐步而巧妙地在节点中进行缓冲，因为在传输之前，必须在一个时间间隔内缓冲去往某些出站链路的进站通信量，以满足控制分布(shaping profile)。在无连接的面向包的网络中，一个进站包的目的地在该包到达之前是未知的。虽然一个特定的流或流动可以有一个整体的最大速率，然而一个给定的信息流可以将信息突发进一个节点。因为在该突发期间，信息流突发能消耗全部的链路容量，根据定义，由该链路所支持的去往另外的节点的包流是空闲的。

作为一个例子，再次考虑图3的网络。节点B连接到三个链路1，4和7。链路4上的通信量可能去往节点A或节点G。去往节点A的一系列包构成从节点B到节点A的流或流动。去往节点G的一系列包构成从节点B到节点G的流或流动。

图7的说明

图7的时序图说明了在链路4上的进站包的一个可能的顺序。

在四个去往节点A的包的突发期间，离开自由表的缓冲区全部加入到支持链路1的队列中。如果这个突发延伸，那么从自由表中取出的缓冲区就流入这个队列中。但是，在任何时刻，下一个进站包可能去往节点A或节点G。下一个进站包的目的地在被看到之前是未知的。因此，灵活的缓冲区动态分配对于有效包交换设备的操作是必要的。在四个包的这个突发期间，支持到节点G的链路7的队列没有接收额外的缓冲区，但是因为突发专用于去往节点A的通信量，这些队列没有任何进站通信量，因为它们不再需要缓冲区。因此，灵活的缓冲区动态分配对于有效包交换设备的操作是足够的。

链表的操作

链表引擎采用半导体存储器作为RAM存储器。操作链表以增加或移除缓冲区的机构涉及一系列对缓冲区存储器、以及相关链表表格存储器的读或写。链表表格存储器是静态结构，它包含查找表格，用于针对每一个由链表处理器所支持的链表的头部和尾部。例如，当一个流具有读取或写入的通信量时，链

表处理器首先使用流号来查找想要的列表的头和尾部的地址。知道了这些想要链表的地址之后，接着处理器就能在该列表上执行指定的操作。当要把一个缓冲区加入到链表中时，一个空缓冲区从图 6 的自由表的头部取出，然后会重写该自由表的头部到该列表中的下一个空缓冲区。自由表的新头部的地址包含在刚刚为了填充而取出的缓冲区的链接中。该缓冲区的内容被填入，然后在链表尾部的缓冲区的链接字段被写入刚刚填充的缓冲区的地址。接着，将新尾部地址写入表格存储器。在将新缓冲区写入链表的处理中，表格存储器支持一个读和一个写操作，缓冲区存储器支持两个写操作。在从链表读取缓冲区的处理中，表格存储器支持一个读和一个写操作，缓冲区存储器支持一个读和一个写操作。由于空缓冲区必须重新链接到自由表，发生对缓冲区存储器的写入。

链表处理器随机访问缓冲区存储器

这个处理的一个重要的特征是访问存储器的随机性。两个因素有助于随机化。当一个链表缓冲来自一个通信设备的通信量时，访问的顺序完全取决于由该设备所传递的通信量。在一个无连接网络中，如 Ethernet 中，到达一个设备的包当在其行程中被路由时，可能去往若干队列中的任何队列。总的来说无法预测将来的包的目标队列。到达的随机性使一个给定队列中的地址不规则。虽然出站包的传送在控制之中，然而网络状况再次有助于随机化。例如，假设出站设备上要复用若干队列。有时，所有这些队列都可能起作用，有时是一些，有时是一个或者没有。信息拥挤问题可能由出站设备的远端流量控制所引起，或者由承载去往出站设备的交通量的若干进站设备所引起。

第二个有助于随机化的重要因素是自由表。对自由表的贡献完全取决于到出站设备的缓冲区传输次序。但是，出站传输服从于不能预测的状况。因此，通信量状况使在自由表中空缓冲区的地址顺序随机化。

一旦使用一个链表处理器用于缓冲区管理目的一个典型系统在显著负载状况下运行一秒或两秒，对缓冲区存储器的访问完全缺乏任何相关性。

缓冲区存储器访问参数和机构

因为链表处理包括一系列的预先编导的对存储器的访问，链表处理器的性能很大程度上取决于这些访问是怎样管理的。但是，无止境的缓冲需求，如存储器组件可用性和访问管理，都约束了为链表处理器设计的存储器系统。

当链表处理器被用在中继交换应用时，附连设备的长度和它的容量对存储

器系统设计产生影响。例如，在 SONET 网络中，标准的电缆截断报告间隔是60 毫秒。然而，对于通常出现的网络电缆行程(其中报告间隔更通常地为 200 到300 毫秒)，这个标准是限制性的。这对应于数千公里的电缆行程。在突发传输最小容量的包时，一个单 OC-768 光纤在 300 毫秒内传送超过 2 千 3 百万个包。一个光纤电缆可以有一百或更多不同并且单独的承载通信量的光纤的线束(strand)。因此，连通一个这样的电缆的系统将需要缓冲上亿个包的量级，以便能够从一个电缆截断中无缝恢复。

硬件链表引擎的基本问题

支持硬件链表处理的存储器子系统必须大、快速、支持许多队列，并且能够在任何时钟边沿操作任何队列中的存储器，以适应当前可用通信量的瞬时变化。随机访问存储器(能合适地操作链表处理器的存储器)不能足够快地循环，或不够大，以致不能缓冲最大容量的设备。

这些两种类型的存储器，即通常可用的同步动态存储器（SDRAM）的包最高包含一兆位的存储，但是它们的随机读写周期时间大约为 60 纳秒。由于一个 OC-768 中继线需要兆位量级的存储，SDRAM 的大小是合适的，但是对于一个满载的、突发传输最小大小的包的 OC-768 中继线来说，每 40 纳秒就到达一个包。因此，市场上可得到的 SDRAM 对于服务一个 OC-768 设备而言速度太慢。

通常可用的同步静态存储器（SSRAM）以大约 2 纳秒来循环进行随机读写访问，等待时间为 4 纳秒。这用来处理少数 OC-768 设备以及控制开销是足够快的。但是，SSRAM 在容量大于 16 兆位的情况下不可用。需要大约 90 个 SSRAM 设备来充分地缓冲一个 OC-768 设备。这些 SSRAM 设备生成的热量将会成为一个问题。

总的来说，可由硬件链表处理器组成的存储器的一个基本问题是，它们或者大但是慢（SDRAM），或者快但是小（SSRAM）。这里没有有效的折衷办法来使硬件链表处理器既大又快。

改进的链表处理器设计

本发明的改进链表处理器采用一个解决方案，来解决如何得到大容量、高密度缓冲区，以及快速、高性能的缓冲区的问题。使用一种崭新的方法来做到这一点，该方法依赖于到存储器的访问的流的随机性。

至今，使用SDRAM的存储器子系统的争用问题导致系统等待每一个连续存储器周期完成，而不是通过使用可用SDRAM的流水线特性来重叠访问。这允许系统以快得多的总线周期速率来执行。通过允许RAM存储器以它的端口速度，即数百兆赫来操作，而不是以它固有的随机读写访问速度，即十几兆赫来操作，本发明解决了争用问题。该发明的另一个特征是，在存在到联组工作的RAM存储器的随机访问流的情况下，更多的存储体被使用，因为随着存储体数量的增加，下一个访问被导向已经忙的存储体的可能性就会下降。

图8和9的说明

动态RAM在一个单独封装内与多个存储体一起封装。存储体是可单独寻址的存储单元。由于存储体共享输入/输出资源，例如物理封装上的引脚，多个存储器可以并行处理访问请求。可以服务的待完成请求的最大数量取决于时钟速度和同步SDRAM设备的构造。作为一个近似，如果对SDRAM中存储体的访问需要四个时钟周期来完成，以及在封装内有四个或更多的存储体，那么能被并行处理四个访问。在完成第一个访问所涉及的四个时钟的每个上升沿使能新的访问。

图8显示了在周期803中的四个存储体810，811，812和813。这代表在一个可能的SDRAM内的四个存储体。漏斗802和喷管804代表共享的控制和数据总线资源，用来访问在SDRAM内的部分。四个访问请求A，B，C，D（801）显示为正在进入SDRAM 803的漏斗802。

一次只能有一个访问请求A、B、C或D能进入漏斗802。每一个存储体801-813用与其它存储体同样的时间处理一个访问。如果一个存储体开始处理请求A，然后过一会儿另一个存储体开始处理请求B，请求A的结果会在请求B的结果出现之前从喷管804出现。但在一段时间内，一个存储体将处理请求A，并且另一个存储体将并行处理请求B。

可以通过任意存储体的组合来为访问服务。这些访问可以由相同的存储体来服务，或者可以由不同的存储体来服务。某些访问将有自己的存储体，但是其它的访问将共享存储体。对于特定的访问组，存储体可能根本不在使用中。例如，同一存储体810可能服务访问A和B，存储体812服务访问C，并且存储体813服务访问D，如图9所示。将一组访问分布到可用存储体的过程叫做分区。从存储体的角度观察，访问的计数是非常重要的信息部分，因为所有的

访问都有一致的特征。因此，分区是针对存储体的访问的核计。例如，{4，0，0，0}表明四个访问占用一个单独存储体，另外三个存储体没有被占用。图9中，分区为{2，1，1，0}。

图10-13的说明

图10的同步动态RAM（SDRAM）是一种四独立存储体存储器结构。对每一个存储体，操作涉及访问等待时间、对于写操作的从引脚到存储器组的信息传递、或对于读操作的从存储器组到引脚的信息传递。这里还需要一个预充电间隔，以允许存储器内的传感放大器准备下一个读或写周期。四个存储体中的每一个都可用。四个存储体中的每一个都有它自己的传感放大器，因此访问只是争用SDRAM的控制和数据端口。

图10和11表明控制总线801提供导向SDRAM的存储体1和2的活动。在图11的访问命令“A”和其相关的读命令“R”之间有等待时间。同样在读命令和数据有效之间也有等待时间。另外，在访问命令A1和A2之间有等待时间。在图11中，为了方便，使用10纳秒的时钟周期。总体周期时间为80纳秒。在这个期间，可以访问SDRAM的所有四个可用存储体，但是在这个间隔内，任何存储体只能被访问一次。对于这些等待时间，我们说SDRAM具有四个级段的流水线深度。冲突情况的消除需要存储器的额外80纳秒周期。冲突情况被定义为访问流量调节器到达忙的SDRAM存储体。例如，如果存储体1的访问请求在图11的时钟周期2到达，它不会被提供到存储器，直到时钟周期9，因为对存储体1的请求A在时钟周期1到达。

图12显示一个硬件链表处理器1201，它使用SDRAM作为它的缓冲区存储器，以及一个较小的同步静态RAM 1207，用于表格存储，它保存每一个支持的流的头部，尾部以及计数。SDRAM 1203有四个SDRAM 1201、1211、1213以及1214。假设链表处理器1201在时钟1、3、5、11和15遇到如图13所示的5个访问请求的流。头三个访问A1、A2和A3能被流水线化，因为它们针对不同的存储体，但是必须延迟在时钟11的第四个访问A2，因为它针对正服务于时钟3的服务A2的忙存储体2。

总的来说，对于一个四存储体的SDRAM，在存在随机访问流的情况下遇到的争用的平均量可以作为一个度量。这样，可以更简单地计算对四存储体SDRAM的一组访问所需的加权平均访问时间，因为存储体只共享输入/输出设

备。例如，考虑分区{3，1，0，0}。如果访问被标记为（A，B，C，D），则可行的分组为{（A，B，C），（D）}、{（A，B，D），（C）}、{（A，C，D），（B）}以及{（B，C，D），（A）}。所选择的两个存储体可以是{1，2}，{1，3}，{1，4}、{2，3}、{2，4}或{3，4}。

有两种办法来映射三个访问的分区以及一个访问的分区到两个存储体。例如，如果我们试着映射分区{（A，B，C），（D）}到{1，2}，它可以是（A，B，C）对应存储体1以及（D）对应存储体2，或是（D）对应存储体1以及（A，B，C）对应存储体1。因为4*6*2=48，以及有256种方法来放置四个访问，那么出现分区{3，1，0，0}的概率为0.1875。在两个冲突访问中，杂散访问可以被流水线化，因此完成所需的时钟数为8*3=24。

分区	概率	完成所需的时钟	完成访问所需的加权平均时间
4，0，0，0	0.0156	32	0.5
3，1，0，0	0.1875	24	4.5
2，2，0，0	0.1406	19	2.6714
2，1，1，0	0.5625	16	9.0
1，1，1，1	0.0938	14	1.3125

表1：四个事务到四个存储器的安排

表1阐述了涉及对四个存储体的访问的各种可能分布的数据，以及针对每一个可能分布示出的相关数据。表1的第一列列出了对存储体的访问能够对应的各种可能分区。表1的首行指示第一个存储体得到四个访问，而剩下的存储体没有得到任何访问。第一列第二行的分布是3，1，0，0。第一列最下一行指示所有四个访问被平均分布到每一个存储体。第二列指示这一行的每一个分布的概率。第三列指示完成功能所需的时钟周期数。第四列指示完成访问所需要的加权平均时间。对于第一行，所有四个访问针对第一个存储体的分布的概率为0.0156。这需要32个时钟周期来完成，其中完成需要的加权平均时间为0.5。分布为1,1,1,1的最下一行的概率为0.0938，需要14个时钟周期来完成，加权平均时间为1.3125。从这个表可以看出，最有可能的分布是2,1,1,0，它的概率是0.5625。在最下一行示出最优的可能分布，有着0.0938的概率。

对于表1的四个存储体，完成四个访问的总加权平均时间为17.98个时钟。这个数值是完成对四存储体存储器的所有可能访问组合所需的加权时间的和。

假设两个 10 纳秒系统时钟的流水线时钟和 16 比特位宽的总线，一个四存储体 SDRAM 的平均可维持吞吐率为 1.24 吉位每秒，因为每一个存储器事务涉及 16 比特每两个时钟。争用开销为 28.5%。这个值是这样算出的：将完成一个访问所需的平均时间和完成一个访问所需的最短时间之间的差值除以完成一个访问所需的最短可能时间。例如，假设完成一个访问的平均时间是 50 纳秒，并且完成一个访问最短可能时间为 40 纳秒。那么开销是 25%。

图 13 和 14 示出了争用的和不争用的访问之间的差别。

图 14 的说明

频繁的争用影响设计决定。考虑图 14 的时序图。这里，针对存储体的访问总线被划分时隙。每一个访问都有一个读间隔以及一个写间隔。必须将这些访问尽可能紧密地打包在一起，以最小化存储器的控制总线上的空闲时间。

以这种方法将访问结构化是有意义的，因为争用基本上增加了存储器的平均读写周期时间。由于平均读写周期时间具有最大读写周期时间的量级，针对存储器的最大读和写周期时间设计系统会更有效。使用更短读写周期在效率上以及操作速度上的提高不值得付出管理争用所需的硬件的成本。

存储器可以以它的总线速度所限定的速率来接受访问请求，因为它的内部流水线设计允许它这样做。但是，对于 28.5%的争用率，与更少争用的系统相比，在这个系统中的排队显著增加，因为争用率与队列深度成指数关系。用来支持可变访问时间的排队和状态控制设备比期望恒定、最大访问时间的设备来更复杂。例如，一个具有恒定、最大访问时间的设计不需要队列，并且状态控制吞吐非常简单。一个具有可变访问时间的设计在正使用存储器的机器中经常需要不止一个的队列，而且需要更复杂的状态逻辑以允许应用程序在争用发生时停止和起动这个机器。

图 15 和 16 的说明

假设 SDRAM 有八个存储体，如图 15 所示。在这种情况下，存储器被分区成图 13 所示的 SDRAM 一半大小的块。例如，前往图 13 的存储体 1 的访问现在或者到图 16 的存储体 1，或者到图 16 的存储体 2。前往图 13 的存储体 2 的访问被分给图 16 的存储体 3 和存储体 4，等等。在图 13 中，到存储体 2 的第二个访问与第一个访问争用存储体 2，因而引起延时。在图 16 中，这些访问中的第一个前往存储体 3，但是这些访问中的第二个前往存储体 4。

图 13 和 16 的比较表明，在图 13 中，争用的减轻消除了时钟 13（A4）和时钟 16（R4）之间的空闲间隙。在图 16，从时钟 7 到时钟 16，数据总线被连续占用。对于指定的随机进站访问流，争用会以不同组合发生。八个存储体并没有消灭争用，但是随着存储体数量的增加，争用发生的可能性减少。

存储体	完成访问所需的加权平均时间	争用开销	最大可维持吞吐率（吉位/秒）
4	17.98	28.5%	1.24
5	17.06	21.9%	1.31
6	16.47	17.6%	1.36
7	16.06	14.7%	1.39
8	15.77	12.6%	1.42
9	15.55	11.1%	1.44
10	15.38	9.9%	1.45
11	15.24	8.9%	1.47
12	15.12	8.0%	1.48
13	15.03	7.4%	1.49
14	14.95	6.8%	1.5
15	14.88	6.3%	1.5
16	14.82	5.9%	1.51

表 2：完成对大量存储体的四个访问所需要的加权平均时间

表 2 显示了随着存储体数量的增加，完成访问所需要的加权平均时间是怎样减少的。最大可维持吞吐率再次假定 10 纳秒系统时钟以及 16 比特位宽的数据总线。

流水线深度也会影响性能。例如，如果有八个存储器存储体，但是在流水线中只有两个级阶，那么完成两个访问所需的加权平均时间是 6.25 个时钟。这个情况的开销为 4.2%，并且最大可维持吞吐率是 1.51 吉位每秒。假设争用率降到 5%。那么平均访问时间是 3.5 纳秒。这非常接近总线周期时间。通过在图 14 和 16 中所示的访问控制方案的比较可以看出结果。可以看出，图 14 中控制总线活动密度更大，意味着与图 14 中的情况相比，每时钟有更多的随机读写访问。

图 17 和 18 的说明

依照本发明，独立的协作状态控制器 1804 被分配到 RAM 封装的每一个存储体 1803。允许每一个存储体独立循环，并且通过它的状态控制器 1804，以和其它 RAM 存储体 1803 协作的方式无缝地形成结果。另外，状态控制器 1804 对访问流量调节器 1801 队列进行流控，该队列在偶然争用的情况下保存访问请求。这就防止了访问请求的丢弃。状态控制器 1804 也独立地管理内务管理功能，如存储体 1803 刷新。状态控制器 1804 是独立的。状态控制器 1804 有助于与其它 RAM 存储体 1803 中的前台访问活动并行地针对其 RAM 存储体 1803 进行后台突发传输。这允许在远离 RAM 存储体 1803 的 RAM 1806 中存储链表的中间部分。这样就只剩下链表的头和尾部留在 RAM 存储体 1803 中。例如，参考图 6 的队列 506，缓冲区 Q 和 Z 位于 RAM 存储体 1803 的某处，但是缓冲区 D 和 R 存储在远程 RAM 1806 中。远程存储链表的中间部分的能力允许该公开的系统使用市场上可得到的封装 RAM 支持任意大小的列表。如果链表的大量内容可以远程地存储在远程 RAM 1806 中，那么嵌入在 FPGA 中的 RAM 1803 可以用于头和尾部。状态控制器 1804 与持有头部和尾部的 RAM 1803 结合。这个设计比驻留在不同于 RAM 1803 的封装中的状态控制器更有效。RAM 1803 和状态控制器的共同定位为存储列表的头和尾部提供了技术选择。这些选择是用于少量链表队列的板载寄存器，用于中等数量链表队列的静态 RAM 1803，或是用于大量链表队列的动态 RAM 1806。

图 17 显示了状态控制器 1804 的模块图。

状态控制器 1804 由仲裁与定序逻辑所控制，该逻辑从前台端口选通输入信息流，并在 RAM 存储体 1803 忙于前台或后台传输时防止它接受新的进入活动。另外，状态控制器监视 RAM 存储体 1803 的状况，并确定与远程 RAM 1806 的交互何时会发生。状态控制器 1804 与后台访问复用器 1808，远程 RAM 1806 以及访问流量调节器一起配合成系统，如图 18 所示。

图 17 的状态控制器 1804 充当它的相关 RAM 存储体 1803，访问流量调节器 1801 以及后台访问复用器 1808 之间的接口。图 17 详细显示了到这些单元的连接。状态控制器 1804 包括复用器 1702 以及仲裁与定序逻辑单元 1703。在它的下侧，复用器 1702 连接到路径 1710 和 1711，路径 1710 和 1711 成为图 18 的总线 1809-1 到 1809-8 的一部分。在这条路径上，复用器经由路径 1710 与其相

关 RAM 存储体通过读、写操作交换数据。路径 1711 是一条双向控制路径，它允许状态控制器 1804 通过复用器 1702 来控制它的相关 RAM 存储体 1803 的操作。RAM 数据路径 1710 可以通过复用器连接到数据路径 1704，或延伸到后台访问复用器 1808 的数据路径，或是通过数据路径 1705 和总线 1802 连接到访问流量调节器 1801。这些数据路径可以在读写操作中使用。

在复用器 1702 底端的 RAM 控制路径 1711 通过路径 1712 和仲裁与控制逻辑单元 1703 连接到路径 1707 和 1706。复用器的路径 1711 一次只能连接到路径 1707 和 1706 中的一个。当连接到路径 1706 时，它进一步通过路径 1810 延伸以在读写操作中控制后台访问复用器 1808 和它的相关远程 RAM 1806 的操作。当路径 1711 通过单元 1703 连接到路径 1707 时，它进一步通过总线 1802 延伸到访问流量调节器 1801。仲裁与定序逻辑单元 1703 包含在通过读和写操作与状态控制器 1804 交换数据时控制访问流量调节器 1801 所需的智能和逻辑。仲裁与定序逻辑单元 1703 也通过总线 1706 和 1810 与后台访问复用器 1808 进行通信，以控制其在远程 RAM 1806 接收来自 RAM 存储体 1803 的数据时的操作，以及其中远程 RAM 1806 向状态控制器 1804 发送数据以输入到与状态控制器相关的 RAM 存储体的操作。

状态控制器 1804 在其通过后台访问复用器 1808 与其相关 RAM 存储体 1803、访问流量调节器 1801 以及远程 RAM 1806 进行控制与数据交换时有四个高层功能。接着说明这四个高层功能。

由图 17 的状态控制器 1804 执行的第一个功能是在读或写请求时启动并控制进站访问序列，该序列与来自访问流量调节器 1801 的信息传输相关，并且在这样做时，控制其相关 RAM 存储体 1803 在写请求时将数据写入到 RAM 存储体 1803，并在来自于访问流量调节器 1801 的读请求时，从 RAM 存储体 1803 读取数据。

由图 17 的状态控制器所执行的第二个功能是响应以触发信号，该信号表明在其相关 RAM 存储体 1803 内检测到一个缓冲区填充水平。这个触发指示在其相关 RAM 存储体内的缓冲区已经充分消耗或已经耗尽。当其相关 RAM 存储体内的缓冲区被充分消耗时，触发对远程 RAM 1806 的写入。当其相关 RAM 存储体内的缓冲区被完全耗尽时，触发从远程 RAM 1806 的读取。

由状态控制器 1804 所执行的第三个功能是启动并管理从其相关 RAM 存储

体 1803 到远程 RAM 1806 的传输，以及管理相反方向上从远程 RAM 1806 到 RAM 存储体 1803 的数据传输。

由状态控制器 1804 所执行的第四个功能是等待来自复用器 1702 的信号，并在收到来自复用器 1702 的该信号时，启动针对远程 RAM 1806 的传输。

由复用器 1702 所执行的另一个功能是在多个 RAM 存储体 1803 同时请求访问远程 RAM 1806 的情况下，选择将访问远程 RAM 1806 的发出请求的 RAM 存储体 1803。复用器 1702 所执行的另一个功能是在上述传输之间存在相关的情况下，启动与远程 RAM 1806 和发出请求的 RAM 存储体 1803 之间的操作相关的传输及调度功能。所述传输之间的相关性源自进入或离开存储器系统的流式访问。

由复用器 1702 所执行的另一个功能是控制 RAM 存储体 1803，以指引来自远程 RAM 1806 的写输入。复用器 1702 准予对远程 RAM 1806 的访问，以及在远程 RAM 1806 和 RAM 存储体 1803 之间路由信息。

表 2 显示 RAM 存储体 1803 争用会限制基于传统 SDRAM 的系统的性能。图 14 的讨论说明了这个限制会非常严重，以致于已经围绕 RAM 存储体争用而设计了系统。与基于预期存储体可用性的争用相反，通过使满 RAM 存储器周期时间的时间间隔的访问结构化，图 14 的设计避免了在每一个周期上的争用。这意味着通过设计，RAM 存储体运行得比最佳速度慢。图 12 和 13 描述了一个预期存储体可用性的系统，但是在争用的情况下，必须有基本的额外逻辑。该实现的问题是争用影响性能，并且所需要的额外硬件不提供足够的性能增益。

图 15 和 16 通过增加 RAM 存储体数量以减少争用量，提供了改进的操作。这需要更多的硬件逻辑。性能改进和硬件增加之间的关系是，可接受数量的额外硬件能产生明显的性能差异。为了协同对许多 RAM 存储体的访问，需要专门的状态逻辑，如图 17 所示的，它消耗硬件资源。这些资源可以在市场上可得到的 FPGA 中找到。为了使性能最大化，一个宽的，可屏蔽的数据总线是最好的。它应该是宽的，以平稳的吸收突发数据，它也应该是可屏蔽的，以助于最小数据片的存储。在市场上可得到的 FPGA 中，可以得到可通过这种方式配置的存储器。但是，该存储器只可少量使用，因为它们对根本的缓冲任务来说是不够的，例如对于缓冲前面讨论的 OC-768 光纤的 300 毫秒。另外，在集成电路中只有有限的可用空间量。用于 RAM 存储器的就不能用于状态控制逻辑，反

过来也是一样。但是，更多的 RAM 存储体意味着更高的性能，并且每一个存储体必须具有它自己的状态控制器。因此，在满足缓冲需求和系统性能之间就有了冲突。

这个冲突的一个解决方案是限制在 FPGA RAM 存储体 1803 上的存储器数量。只访问链表的头和尾，并且链表的中间单元始终为空闲，直到它们移动到列表的头部的时候。因此，将链表的中间单元从 FPGA RAM 存储体 1803 移出到远程 RAM 1806，允许有更高的性能。

图 17 和 18 显示了实现这种折衷方案的一个系统。

就经济原因而言，这个解决方案也是明智的。RAM 存储器的可配置存储体就每个比特而言比市场上可得到的 SDRAM 贵得多。如果保存链表中间单元的远程 RAM 1806 没有正被访问，那么不会付出与该存储器相关的周期时间惩罚代价。最终，不得不访问远程 RAM 1806 以存储和取得列表中间的单元。但是，FPGA 上存储器的许多存储体的如上所述的可配置性质允许设计与 SDRAM RAM 存储体 1803 的突发模式兼容。可以使存储器的许多存储体的系统的吞吐率与 SDRAM 匹配，从而实现平衡的设计。通过允许 SDRAM 1803 以突发模式工作，最小化了 SDRAM 周期时间的成本，因此周期时间可以与在数据总线上按时钟同步提供的数据流水线化。这将存储器返回到其所设计针对的操作模式。

通过后台访问复用器 1808，链表缓冲区被存放到远程 RAM 1806，或从远程 RAM 1806 取回。这允许前台访问在没有额外传输的情况下继续进行。这非常重要，因为由表 2 阐述的该几率模型依靠如图 18 所示的可用前台总线。前台总线 1810 在针对远程 RAM 1806 的后台传输进程中的阻塞使用来产生表 2 的模型大大地复杂化。这会降低性能。这里显示了后台访问总线 1810。

图 18 公开了一个实施本发明的链表引擎 1800。所示的链表引擎连接到一个通信系统 1811，通信系统 1811 具有连接到端口 1817 和 1818 的输入和输出路径 1812 和 1813。该系统包括一个处理器 1814，并进一步包括路径 1815 和 1816，路径 1815 和 1816 通过路径 1819 延伸到访问流量调节器 1801。在其操作中，系统 1811 使用链表引擎 1800 的存储器执行读和写操作，以存储端口 1817 和 1818 在其操作中所需要的数据。

访问流量调节器 1801 和总线 1802 被连接到多个状态控制器 1804，其中每一个状态控制器 1804 都与 RAM 存储体 1803 中的一个相关。访问流量调节器

1801 接收来自系统 1811 的写请求，其请求存储信息到一个 RAM 存储体 1803 中。访问流量调节器 1801 接收和存储这些访问请求，然后选择性地将它们分配到各个状态控制器 1804，以将数据输入到相关的 RAM 存储体 1803。当访问流量调节器 1801 接收到来自系统 1811 的读请求时，该处理以相反的方向执行存储器读操作，并且通过状态控制器 1804，使得包含所请求数据的 RAM 存储体 1803 被读取并通过状态控制器 1804 和总线 1802 提供给访问流量调节器 1801，访问流量调节器 1801 则发送给系统 1811。

RAM 存储体设备 1803 是具有一个相对较小、由远程 RAM 1806 来扩充的存储器存储容量的高速单元，当 RAM 存储体不立即需要信息时，远程 RAM 1806 可以从 RAM 存储体 1803 接收信息，并将它保存。后台访问复用器 1808 和总线路径 1810 协助远程 RAM 1806 的操作，每一个总线路径 1810 都延伸到一个唯一的状态控制器 1804，以及它相关的 RAM 存储体 1803。通过这种方法，一个正变满或变空的 RAM 存储体 1803 可以将此情况通知其相关状态控制器 1804，状态控制器 1804 则通过总线路径 810 与后台访问复用器 1808 通信。复用器 1808 协助状态控制器 1804 从其 RAM 存储体 1803 读取信息，以及将它传输给远程 RAM 1806 进行临时存储，直到其所来自的 RAM 存储体 1803 再次需要该信息。在这个时候，RAM 存储体 1803 发信号给它的状态控制器 1804，告诉它远程 RAM 1806 包含将要被 RAM 存储体需要的信息。接着后台访问复用器 1808 以及状态控制器 1804 共同使得远程 RAM 1806 的适当部分被进行信息读出操作，其中信息被传输到其所来自的 RAM 存储体 1803。远程 RAM 1806 是相对慢速的大容量存储器单元，它可以有效存储从 RAM 存储体 1803 溢出的信息，或是供给信息给下溢的 RAM 存储体 1803。

本发明的一个特征包括图 18 的方法和设备，其中对 RAM 存储体 1803 的写操作通过以下步骤执行：将链表信息经由其状态控制器 1804 写入 RAM 存储体 1803；继续对 RAM 存储体 1803 的写操作，直到它快要接近满状态；继续将来自访问流量调节器 1801 的额外信息写入 RAM 存储体，同时并行地经过它的状态控制器 1804 和后台复用器 1808 从 RAM 存储体 1803 读出某些新接收的信息并写入远程 RAM 1806。这个信息一直保存在远程 RAM 1806 中，直到随后所来自的 RAM 存储体 1803 需要它。

图 18 的系统通过以下步骤执行从 RAM 存储体 1803 读出数据的操作：给

与包含所请求数据的 RAM 存储体 1803 相关的状态控制器 1804 发信号；通过状态控制器 1804 和总线 1802 启动所选择数据的读出，并返回到访问流量调节器 1801；继续所选择 RAM 存储体 1803 的读出，并并行确定这些读操作已经耗尽具有想要数据的所选择 RAM 存储体 1803，并且随后的读操作所需的一些想要的数据当前存储在远程 RAM 1806；启动远程 RAM 1806 预取读出，以在其被请求之前将信息传输回其所来自的 RAM 存储体 1803；并且继续从所选择的 RAM 存储体 1803 读出数据，并且如果需要，继续从远程 RAM 1806 读出数据并返回到正被读出的 RAM 存储体 1803。这个操作一直继续，直到在读操作中由访问流量调节器 1801 请求的信息全部完成。

RAM 存储体 1803 的高速存储设备以及远程 RAM 1806 的低速大容量存储器协同工作以存储超过高速 RAM 存储体 1803 存储容量的数据。远程 RAM 1806 在写操作时接收来自正被写入的 RAM 存储体的这个数据，同时允许 RAM 存储体 1803 继续进一步从访问流量调节器 1801 高速接收数据。当 RAM 存储体 1803 最初以高速读出时，处理以相反方向执行读操作；在预取方案中，当 RAM 存储体 1803 中所要的信息耗尽时，RAM 存储体所需并且存储在远程 RAM 1806 内的数据就传输回 RAM 存储体 1803。在远程 RAM 1806 中，读操作可以继续将所有有关的信息传输回高速 RAM 存储体 1803，其中高速 RAM 存储体 1803 继续以高数据率被读出，直到访问流量调节器 1801 所请求的信息完全通过状态控制器 1804 和总线 1802 由高速 RAM 存储体 1803 传输回访问流量调节器 1801。或者，状态控制器 1804 和 RAM 存储体 1803 可以继续与上一次读操作分离的写操作，其中读操作触发了后台取回。因为预取是在触发时自动执行的，当一个特定的状态控制器 1804 和 RAM 存储体 1803 被后台传输所占用时，其它的状态控制器 1804 和 RAM 存储体 1803 有空执行与这个后台操作独立的操作。

本发明的一个进一步的特征是使用状态控制器 1804，通过路径 1820 来提供信号到访问流量调节器 1801，以指示与每一个状态控制器 1804 相关的 RAM 存储体 1803 当前是否忙于已有的读或写操作，或是可用于接收的针对新的读或写操作的请求。因为 RAM 存储体设备 1803 的存储器单元以光纤总线的高数据率操作，因此 RAM 存储体 1803 可以以适合于光纤总线的速度执行读或写操作。从而，通过路径 1820，由状态控制器提供忙信号给访问流量调节器 1801，以向其相关 RAM 存储体指示其可用或不足。这个信号只持续 RAM 存储体 1803 执

行读或写操作所需的几纳秒。因此，通过状态控制器 1804 向路径 1820 提供这些忙/空闲信号包括一个争用设备，它允许访问流量调节器 1801 和它的单元 1821 监视 RAM 存储体 1803 的忙/空闲状态。这消灭了任何由于现有技术复杂逻辑电路单元或争用装置导致的争用延迟，所述争用装置在对 RAM 存储体 1803 的读或写之间带来任意预定最小时间间隔。通过这个方法，访问流量调节器 1801，路径 1820 和状态控制器 1804 提供了一个有效并且是高速的争用设备，它以光纤总线的纳秒速率操作。该增强的高速争用方案允许一个由系统 1811 所服务的更大的数据吞吐率，因为它的与端口 1817 和 1818 相关的输入和输出队列可以以一个更快速的速率来进行处理数据交换，因为它们由 RAM 存储体 1803 的高速单元所服务。因此，图 18 的链表引擎 1800 可以以光纤链路相当的速度，执行由端口 1817 和 1818 所需要的数据队列读和写的操作。

图 19 的说明

图 19 公开了一个典型的链表，包括五个缓冲区 1 到 5。五个缓冲区链表并不存储在图 18 的同一个 RAM 存储体 1803 中。五个缓冲区随机存储在五个分别的存储体 1803 中。每一个缓冲区都有一个第一部分，它包含要由系统存储和处理的物理信息或数据。每一个缓冲区的下部包含一个针对 RAM 存储体的链路字段地址，其中该 RAM 存储体存储了链表的下一个缓冲区。图 19 的缓冲区 1 在它的上部存储物理信息，并在它的下部存储 0100 的链路字段地址。存储缓冲区 1 的 RAM 存储体的地址是 000/00，如缓冲区 1 的右边所示。

链表的缓冲区 2 存储在 RAM 地址 010/00(如缓冲区 1 的链路字段所指定)。缓冲区 2 的头部包含物理信息（数据）。下部包含链路字段地址 01001，指定存储链表的缓冲区 3 的 RAM 存储体标识和位置。

缓冲区 3 存储在 RAM 存储体地址 010/01(如缓冲区 2 链路字段所指定)。缓冲区 3 的链路字段包含地址 11010，指定链表的第四个缓冲区的位置。

第四个缓冲区存储在 RAM 存储体 110/10(如缓冲区 3 的链路字段所指定)。相似地，链表的第五个，即最后的缓冲区存储在 RAM 存储体 101/11(如缓冲区 4 的链路字段所指定)。缓冲区 5 的头部指示：这是可用的空闲列表的头缓冲区。

链表的缓冲区随机地存储在图 18 的分立的 RAM 存储体中。该随机性对于系统的高效数据处理和控制操作而言期望而且必需的。该随机性对于实现本发明的争用设备的所需操作特别需要而且有用。

图20和21的说明

图20显示了RAM存储体1803中链表队列的头部如何被耗尽，然后需要大容量RAM存储器1806的访问。所示的链表队列按它们各自的顺序缓冲有内容Q，F，R和Z。在图21中，时钟周期A1的第一个访问从在图21的时钟周期7和8保存内容Q的RAM 1803地址读取。但是，访问A1从高速存储器1803取得队列的头部以使用，并将下一个单元F存储在远程RAM 1806。因此，访问A1在时钟周期2通过涉及访问A1的RAM存储体1803的状态控制器1804触发后台请求Rq1。这个远程存储器1806访问请求由后台访问复用器1808处理，接着在图21的时钟周期3由远程存储器访问A1B返回。注意，远程RAM 1806访问与针对RAM存储体1803的访问流并行发生。数据D1B在时钟周期9和10内从远程RAM 1806取出。这个数据是链表单元F。该单元F被写入一个从空闲列表中取出的空单元，其地址在RAM存储体5中，并且通过在时钟周期11开始的访问A5取出。一直保持着连接。因此，拥有单元F的队列的头部再一次驻留在高性能RAM存储体1803-1805中。持有单元Q的RAM存储体1803-1805以及先前持有内容F的远程存储器单元被返回到它们各自的空闲列表。

对高性能RAM1803的访问的无关联性被保留，以保持导致表2的模型的完好，同时维持更有效利用的前景，如图20中通过访问控制中间隙的缺乏所演示的。在较早前的讨论中，间隙是由争用所引起的。相关的通信量(是顺序的对相同存储体的访问)引起了争用。

图22的说明

图22公开了本发明的处理，它执行由访问流量调节器1801启动的读请求，其中访问流量调节器1801请求读取一个链表，该链表的缓冲区随机地存储在图18的各个RAM存储体1803-1到1803-8中（下文的RAM存储体1803）。当访问流量调节器1801接收到来自处理器1814的请求读操作的指令时，处理在步骤2201开始。链表的单独的缓冲区按顺序一次读取一个，然后随机存储在各个RAM存储体1803中。每一个缓冲区的读取需要访问流量调节器1801的单独读请求。

第一个读请求由步骤2202所接收，并且延伸到步骤2203，该步骤确定从RAM存储体1803读出的单元的数量是否超出了门限值。如前所述，链表的读

取需要从 RAM 存储体 1803 读取列表的初始缓冲区（头部），其中头缓冲区存储在RAM 存储体 1803 中。链表的剩余执行需要读取链表的中间缓冲区，中间缓冲区存储在远程 RAM 1806 中并且必须取得及返回到 RAM 存储体 1803。从远程 RAM 1806 回到 RAM 存储体 1803 的有效缓冲区传输需要将多个这样的请求提供给后台访问复用器 1808，并且接着提供给远程 RAM 1806 以提高步骤执行效率。正是这个原因，才提供了门限值检测单元 2203，使得多个这样的请求同一时间被延伸到远程 RAM 1806，而不是单独地一次一个。

最初假定单元 2203 确定没有超过门限值。在这个情况下，并不马上访问远程 RAM 1806，并且处理继续到步骤 2204，该步骤 2204 读取由针对链表的第一个缓冲区（头部）的读请求所标识的 RAM 存储体。读取并暂时存储这个缓冲区位置，然后处理继续到了步骤 2205，该步骤 2205 将读信息返回到访问流量调节器 1801。接着处理延伸到步骤 2206，指示 RAM 存储体 1803 已经准备好接收来自访问流量调节器 1801 的下一个访问请求。尾部缓冲区由相同的方法读取，如果链表只由一个缓冲区组成的话。

接着假定单元 2203 确定步骤 2202 的新到的读请求导致用于读操作的在 RAM 存储体 1803 中可得到的缓冲区的数量超过门限值。在这种情况下，步骤 2211 读取头缓冲区并且继续到步骤 2220，其中步骤 2220 将步骤 2211 的读信息传送到访问流量调节器 1801。接着步骤 2221 请求读取中间缓冲区，中间缓冲区靠近在被读取的 RAM 1806 中的列表的头部。这包括针对列表的头部的请求。这样，步骤 2221 将针对列表的新头缓冲区的读请求放置在后台访问总线 1810 上，该后台访问总线 1810 服务于特定的 RAM 存储体 1806。接着，步骤 2222 从远程 RAM 1806 取得列表的多个中间缓冲区。接着处理继续到步骤 2223，步骤 2223 指示远程 RAM 1806 已经为另一个访问而准备好。处理也继续到步骤 2212，步骤 2212 将在步骤 2222 中从远程 RAM 1806 读出的信息写入特定的 RAM 存储体 1803。该信息包括特定 RAM 存储体 1803 中列表的新头缓冲区的信息。接着处理继续到步骤 2205，步骤 2205 将信息延伸到访问流量调节器 1801。接着处理延伸到步骤 2206，该步骤 2206 表示读请求的完成，并指示 RAM 存储体 1803 已经为下一个访问请求而准备好。

图 23 的说明

图 23 公开了执行一个接收自访问流量调节器 1801 的写请求所需的步骤。

处理从步骤2301开始，其中访问流量调节器1801将写请求提供给总线1802。步骤2302让写请求进入到图18的状态控制器1804。步骤2203确定靠近列表尾部的被写入单元是否超过门限值。如果没有，（这可能是一个新列表的情况），该列表的最后缓冲区（尾部）被写入RAM存储体1803。处理接着进行到步骤2305，步骤2305向访问流量调节器1801指示对远程RAM 1806的请求是不必要的。接着处理进行到步骤2306，步骤2306指示RAM存储体1803已经为下一个访问而准备好。

假设步骤2303确定用于写请求的门限值被超过。在这个情况下，处理进行到步骤2311，步骤2311取得RAM存储体1803所保存的列表的任何尾部。接着处理进行到步骤2321，步骤2321将在步骤2311中取得的尾缓冲区放置在后台访问总线1810上，以存放到远程RAM 1806。接着，步骤2322更新从远程RAM 1806中倒数第二个最后缓冲区到在步骤2321中所写的缓冲区的位置的链接字段，因为通过步骤2311和2321，在倒数第二个缓冲区的链路字段中指示的缓冲区的位置已经从RAM存储体1803改变到远程RAM 1806。步骤2323指示刚写入的远程RAM 1803已经准备好被访问。

与步骤2321并行地，在步骤2312中，一个空缓冲区被写入RAM存储体1803的链表的末端。指向在步骤2321中写入的缓冲区的指针被写入到空缓冲区的链接字段，其中空缓冲区是在步骤2312被写入的。接着处理延伸到步骤2306，步骤2306表示写操作完成，并指示RAM存储体1803已经为下一个访问请求做好准备。

图24的说明

存储器管理设施被说明为用来处理链表文件。依照本发明的另一个可能的实施例，公开的存储器管理设备也可以使用高速RAM存储体1803以及远程RAM 1806，以处理数据文件，其中的数据文件不是链表类型。这在图24和25的处理步骤中说明。

接下来说明图24的处理步骤，其中图18的读操作，将连同RAM存储体1803和远程RAM 1806来说明，其中，RAM存储体1803和远程RAM 1806被构造成以这样一种方式来操作：高速低容量RAM存储体1803存储接收自访问流量调节器1801的信息。远程RAM 1806被用作存储大容量文件信息的溢流口。处理在步骤2401开始，然后进行到步骤2402，其中访问流量调节器1801将一

个读请求发送到总线1802，以请求存储在RAM存储体1803中的信息。

步骤2403确定将要取得的文件大小是否超过当前RAM 1803存储器的容量。如果没有超过门限值，那么处理进行到步骤2404，其中RAM存储体1803读取所请求的文件以传送回访问流量调节器1801。接着处理进行到步骤2405，步骤2405将从RAM存储体1803读出的所请求信息由状态控制器1804经过总线1802返回到访问流量调节器1801。访问流量调节器1801接收信息，并将它传递给处理器1814以供在功能与所请求的信息相关的控制器中使用。处理接着继续到步骤2046，其中步骤2046通知访问流量调节器1801：RAM存储体1803已经为接收其它访问请求作好准备。

如果单元2403确定要读的所请求的文件大小超过了门限值，处理移动到步骤2410，其中步骤2410读取可能在RAM存储体1803中的文件部分。处理继续到步骤2411，其中在步骤2411中，读取的信息被返回到访问流量调节器1801。处理继续到步骤2412，其中步骤2412将一个来自远程RAM 1806的读请求放置到后台访问总线1810上，它将该请求通过后台访问复用器1808延伸到所请求的远程RAM 1806。

接着处理移动到步骤2413，它从远程RAM 1806取得所请求的信息。接着步骤2415将在步骤2413中取自远程RAM 1806的所请求信息，传送至存储它的RAM存储体1803。

图25的说明

图25说明了对一个已经存在于RAM存储体1803中的文件进行附加的写请求。RAM存储体1803和远程RAM 1806协作以通过与图24所描述的读操作相似的方式存储接收自访问流量调节器1801的大容量数据。RAM存储体1803存储一个文件的已有的选定数量的数据，而大容量文件的剩余部分溢出，并写入远程RAM 1806。

该处理从步骤2501开始，并继续到单元2502，其中单元2502分析接收自访问流量调节器1801的写请求，并确定要写的所请求文件的大小是否超过能被存储进RAM存储体1803的容量。如果没有超过门限值，那么单元2502导致步骤2504将额外数据写入已经存在于所选择的RAM存储体1803中的相关文件。处理接着继续到单元2505，该单元2505将一个确认信息发送回访问流量调节器1801，通知它已经将该请求的文件写入RAM存储体1803，并且没有必要

写入远程RAM 1803。

如果单元 2502 确定超过了门限值，那么处理就继续到步骤 2511，该步骤导致读取已经存储在 RAM 存储体 1803 中的文件部分。处理接着继续到步骤 2522，它开始一个操作，该操作是导致在步骤2511 中取得的这个文件部分被写入远程RAM 存储体 1806 所需的。步骤 2523 指示远程 RAM 1806 已经为访问作好准备。步骤2512 导致写指针被写进RAM 存储体 1803，以允许其针包含文件剩余部分的远程 RAM 1806 地址关联，其中该文件的其它部分被写入 RAM 1803。处理接着继续到步骤 2506，在步骤 2506 中指示 RAM 存储体 1803 已经为接收另外访问请求而作好准备。

后记

网络传输控制需要灵活的缓冲。高优先权流比其它的低优先权流要优先。例如，实时传输的要求高于普通数据传输，例如文件传输。但是，传输的瞬时特性是随机的。下一个进站包可能被导向任何流。例如，再一次考虑图 3。在链路 1 上进入节点 A 的包流可出站到链路 2 或链路 3。在链路 1 上进站的传输可携带导向链路 2 的数千个连续的包，并且携带一个导向链路 3 的单独包，其中这个单独包跟在导向链路 2 的数千个连续包之后。除了这个需求，即不丢弃包，以及支持从灾难性的事件，如电缆被截断中恢复之外，它必须支持可变延迟调度的出站通信量控制算法。因此，输入和输出通信量的性质是复杂的，并且需要灵活的缓冲。

最有效的平衡进站流和出站流的处理方法是硬件链表引擎。但是，当前硬件链表引擎的实现，大而且慢，或是小但是快。本发明的增强链表引擎比先前的列表引擎要高级，因为它提供不贵的千兆字节的缓冲区存储器，并且它以可用半导体存储器的最大吞吐率运行。因此，本发明该增强的链表引擎如当前技术状态般大，并且相当快。额外的速度，对通信量控制应用程序来说是有吸引力的，因为该增强的链表引擎支持更大容量的（例如，OC-768）光缆，假定相同数量的硬件资源，即 FPGA 和 SDRAM。该增强的链表引擎允许支持更高容量的线路，因为它的缓冲区的深度高达千兆字节。

这个发明支持由市场上可得到的动态存储器形成的突发。这个支持可以由本发明的两个方面来表明。首先，当对市场上可得到的动态 RAM 的访问是连续的和相邻的时，这个发明占用市场上可得到的动态存储器的数据引脚，读取

所取得的信息或写入要存储的信息，同时启动下一个相邻的访问。接着，本发明高速缓冲要写入到市场上可得到的动态 RAM 中的缓冲区，以及以逐个上下文的方式缓冲从市场上可得到的动态 RAM 读取的缓冲区。

通过并行处理当前访问请求的数据引脚和启动下一个访问请求，本发明利用市场上可得到的动态 RAM 上的数据总线的锁存特性。这个特性允许动态存储器的一整行，以数据总线的速度(通常是 166 兆赫兹或更高)读出或写入。这个行在长度上可以是 128 比特，但是在动态存储器上可用的数据总线引脚数可能只是 8。因此，存储器在内部锁住要读取的存储器行，并将这个信息以每次 8 字节的方式发送给引脚，或者在以每次 8 字节的方式将整行写入它的内部存储器组前锁住进站数据。通过以这种方法重叠访问，本发明以这样的方法维持在数据总线上的数据：从市场上可得到的动态存储器中的连续读取以与本发明的 RAM 存储体(如图 18 的 RAM 存储体 1803)兼容的速率进行。当一个长度为许多缓冲区的列表被连续访问时，这是值得的。在这样的情况下，信息必须从市场上可得到的动态存储器中流出。为了在这种对市场上可得到的动态存储器的扩展访问系列中维持性能，到市场上可得到的动态存储器的接口的稳定状态性能必须与其它任何存储器一致。

将要写的缓冲区和已经读取的缓冲区进行高速缓冲，允许有效率地使用市场上可得到的动态存储器。通过行来访问市场上可得到的动态存储器会更有效。但是，行通常很大，差不多 128 比特或更高。要存储的或取得的单个缓冲区可能只包含 16 比特。因此，如果高速缓冲缓冲区，直到收集了足够的信息而构成一个整行，接着到市场上可得到的动态存储器的一个写访问将一个整行传输，这是最有效的。相似地，如果只依据读操作而高速缓冲整行的话，每行只有一个访问需要由市场上可得到存储器形成，它是最有效的。

因此，将读和写操作高速缓冲导致根本上的性能提升，因为只需要少量的对市场上可得到的动态存储器的访问，并且对市场上可得到的动态存储器来说，减小了访问量和争用。

本发明支持许多并行链表。到许多并行链表中的一个任意链表的交互完全独立于许多并行链表的其余部分。

下面的权利要求将 RAM 存储体 1803 表征为高速存储器，以及将远程存储器 1806 表征为大容量存储器。

上述的说明公开了可能的示例性的本发明的实施例。可以预料本领域的技术人员能够而且会设计替换实施例，从而在如下面的权利要求在字面上陈述的通过同等的表述所陈述的本发明的范围内。

图1

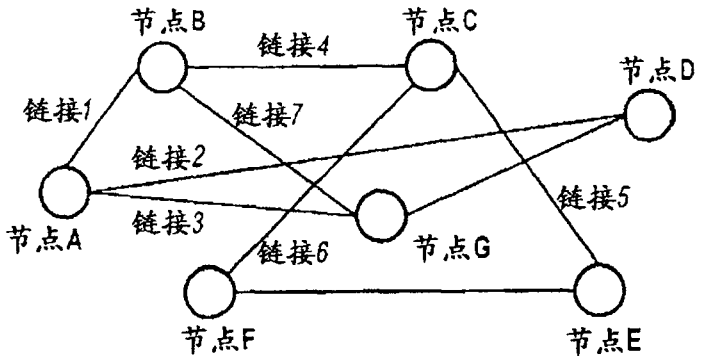

节点B
链接4
节点C
节点D
链接1
链接7
链接2
链接5
节点A
链接3
节点G
链接6
节点F
节点E

图2

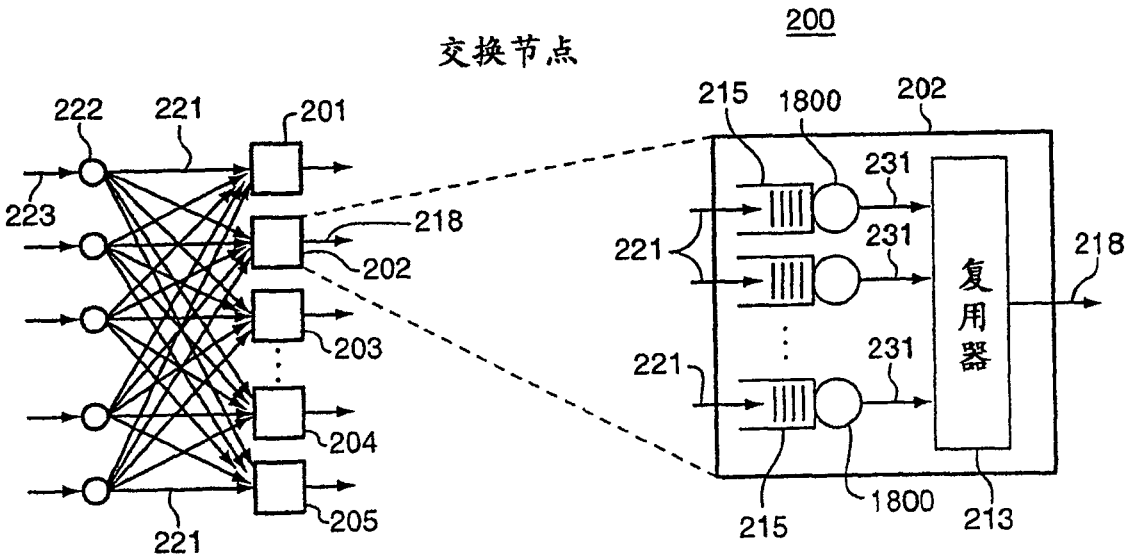

200
交换节点
222
221
201
223
218
202
203
204
205
221
215
1800
202
231
231
221
218
复用器
231
221
215
1800
213

图3

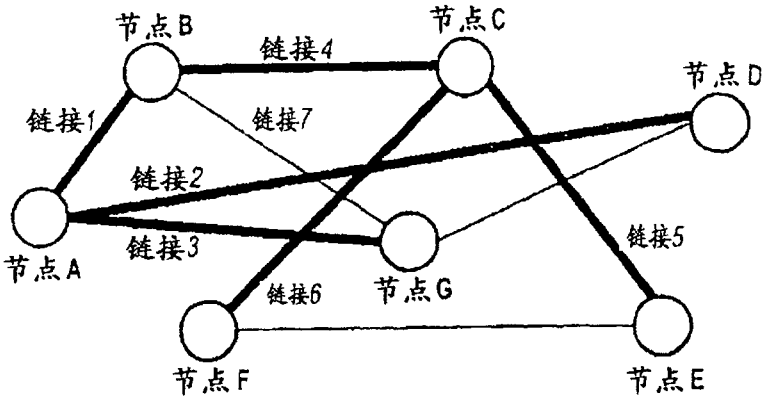

节点B
节点C
链接4
节点D
链接1
链接7
链接2
链接3
节点A
链接5
链接6
节点G
节点F
节点E

图4

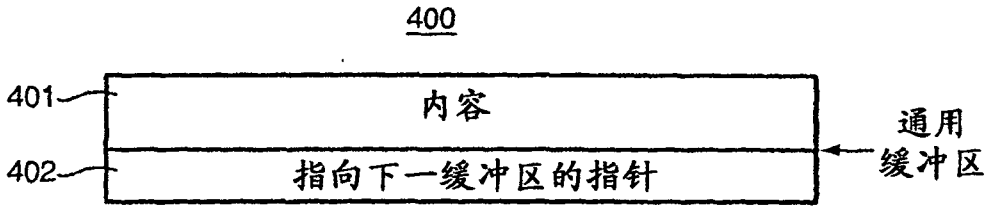

400
401
内容
402
指向下一缓冲区的指针
通用
缓冲区

图5

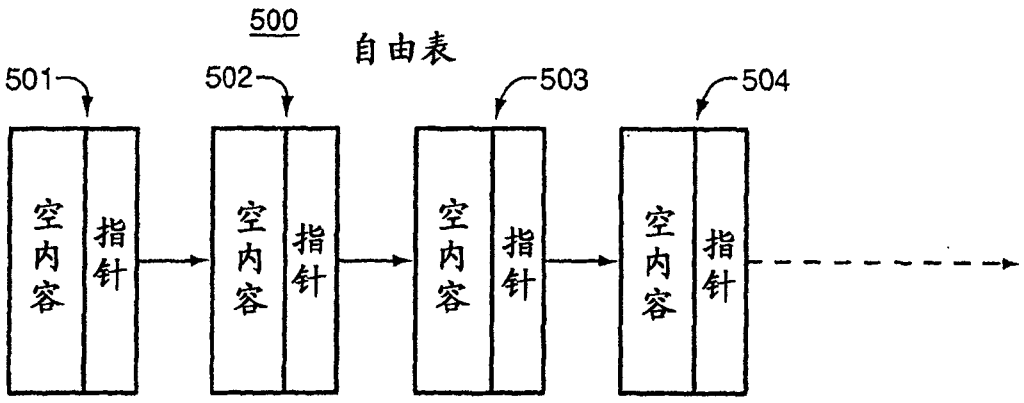

500
自由表
501
502
503
504
空内容
指针
空内容
指针
空内容
指针
空内容
指针

图6

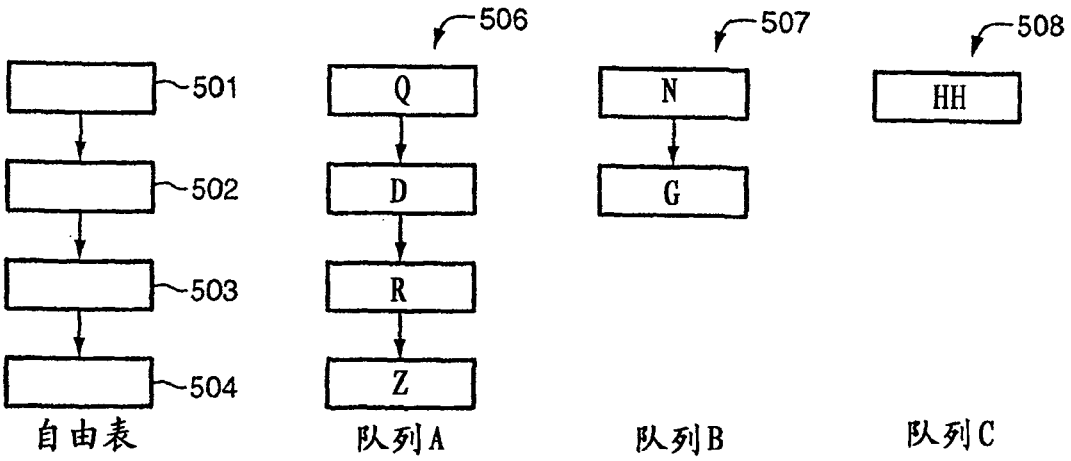

501
502
503
504
506
Q
D
R
Z
507
N
G
508
HH
自由表
队列A
队列B
队列C

图7

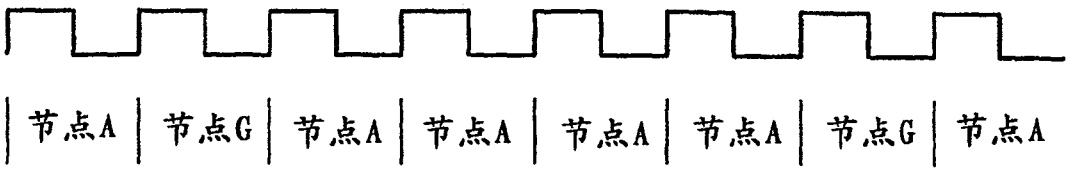

节点A
节点G
节点A
节点A
节点A
节点A
节点G
节点A

图8

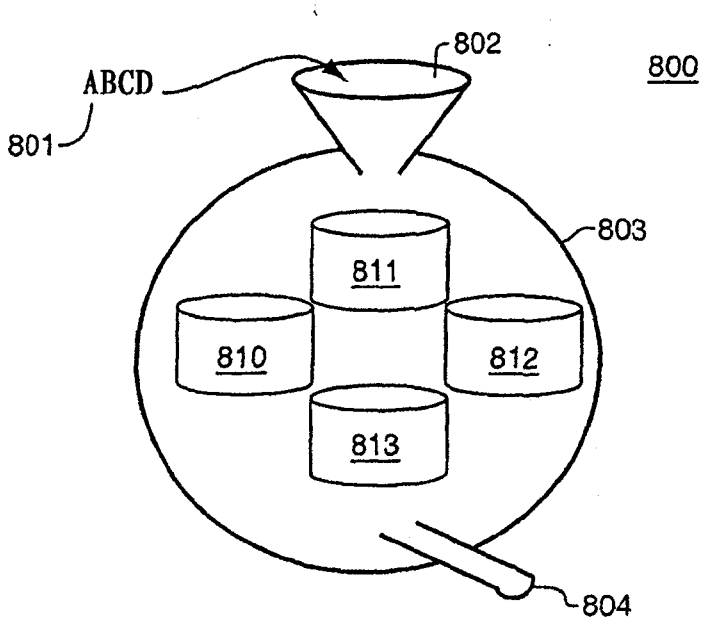

ABCD
801
802
800
803
811
810
812
813
804

图9

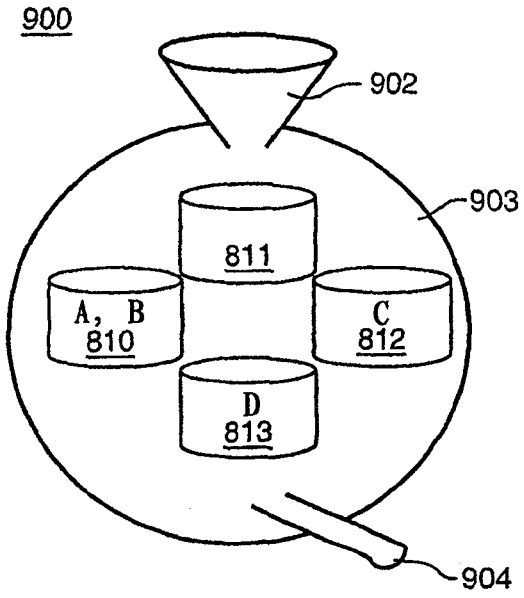

900
902
903
811
A, B
810
C
812
D
813
904

图10

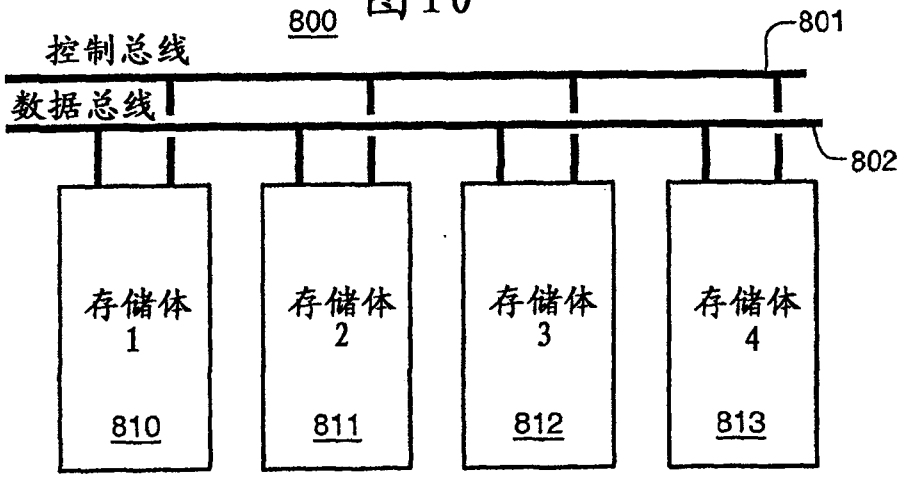

800
801
控制总线
数据总线
802
存储体
1
810
存储体
2
811
存储体
3
812
存储体
4
813

图11

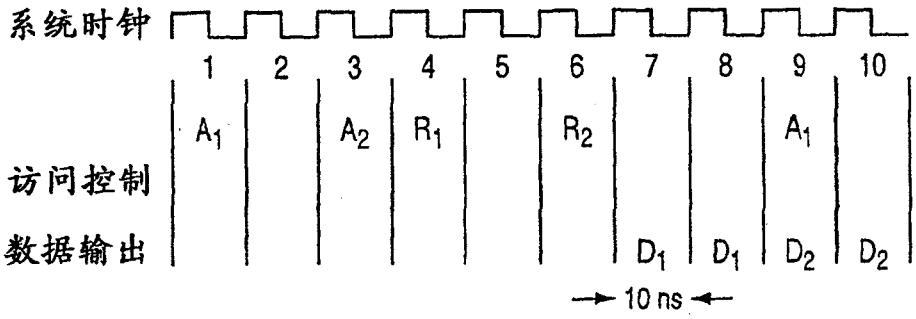

系统时钟
1
2
3
4
5
6
7
8
9
10
访问控制
数据输出
A1
A2
R1
R2
A1
D1
D1
D2
D2
10 ns

图12

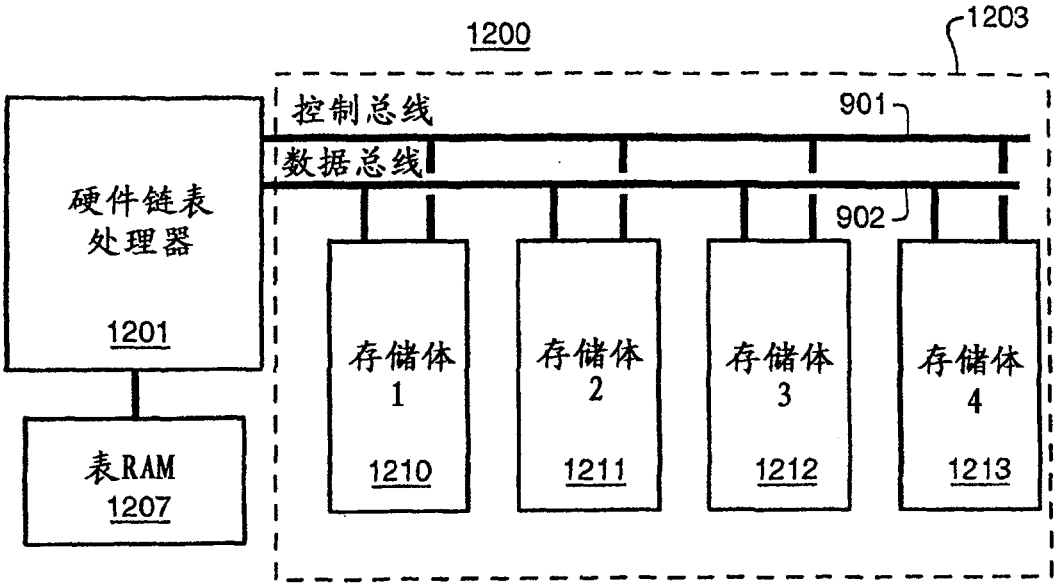

1200
1203
控制总线
901
数据总线
902
硬件链表处理器
1201
表RAM
1207
存储体1
1210
存储体2
1211
存储体3
1212
存储体4
1213

图13

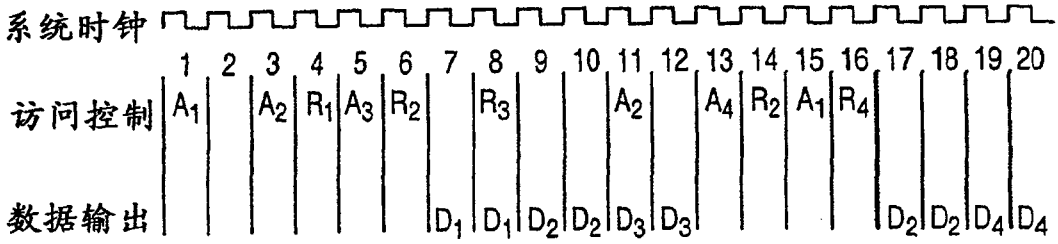

系统时钟
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
访问控制 A1 A2 R1 A3 R2 R3 A2 A4 R2 A1 R4
数据输出 D1 D1 D2 D2 D3 D3 D2 D2 D4 D4

图14

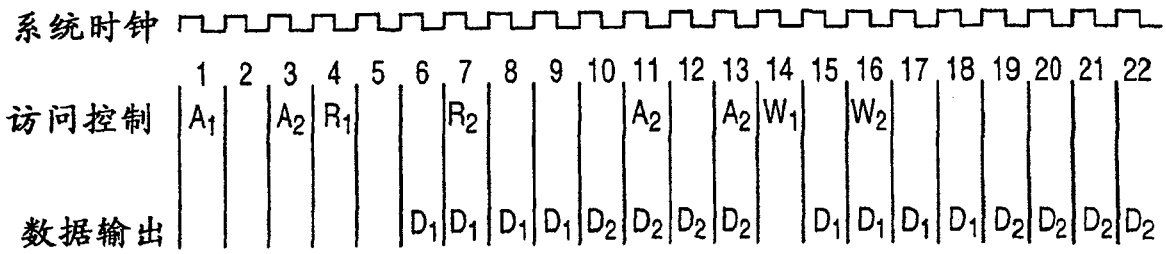

系统时钟
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
访问控制 A1 A2 R1 R2 A2 A2 W1 W2
数据输出 D1 D1 D1 D1 D2 D2 D2 D2 D1 D1 D1 D1 D2 D2 D2 D2

图15

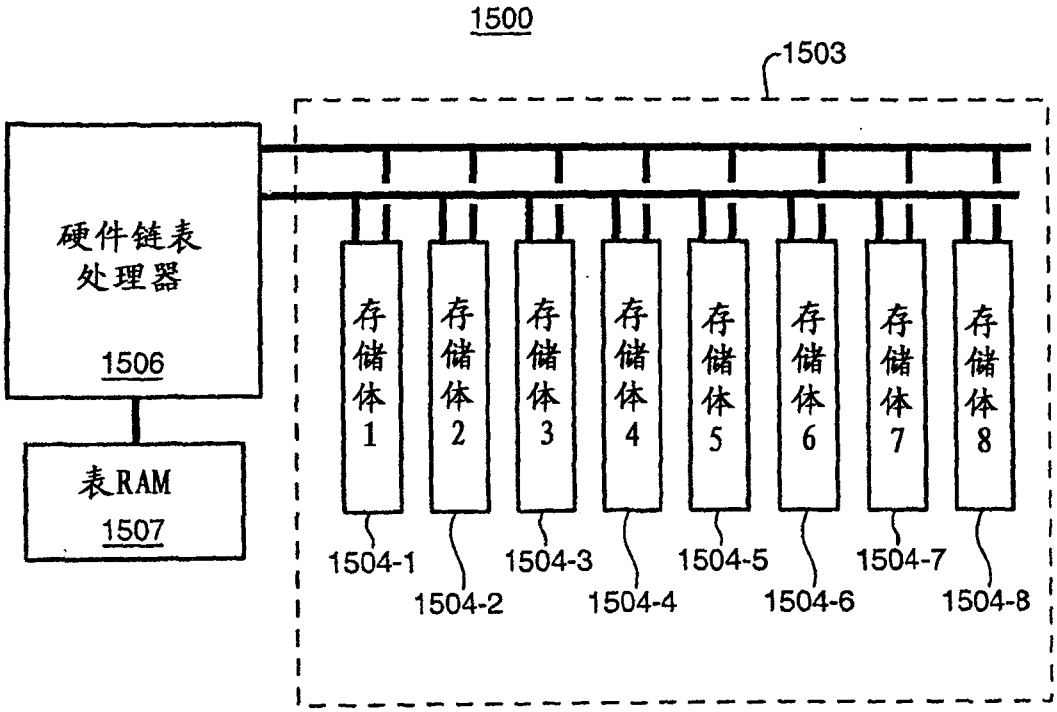

1500
1503
硬件链表处理器
1506
表RAM
1507
存储体1
存储体2
存储体3
存储体4
存储体5
存储体6
存储体7
存储体8
1504-1
1504-2
1504-3
1504-4
1504-5
1504-6
1504-7
1504-8

图16

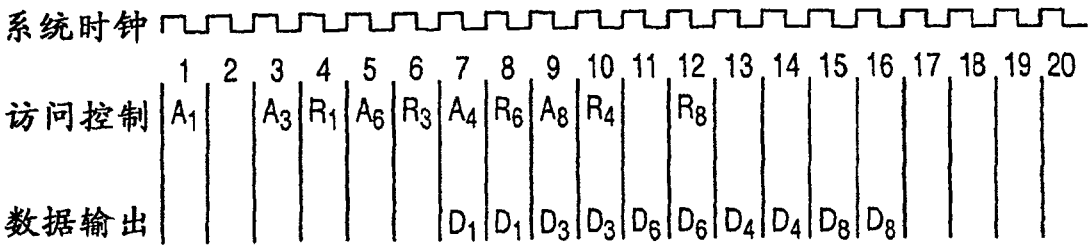

系统时钟
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
访问控制 A1 A3 R1 A6 R3 A4 R6 A8 R4 R8
数据输出 D1 D1 D3 D3 D6 D6 D4 D4 D8 D8

图17

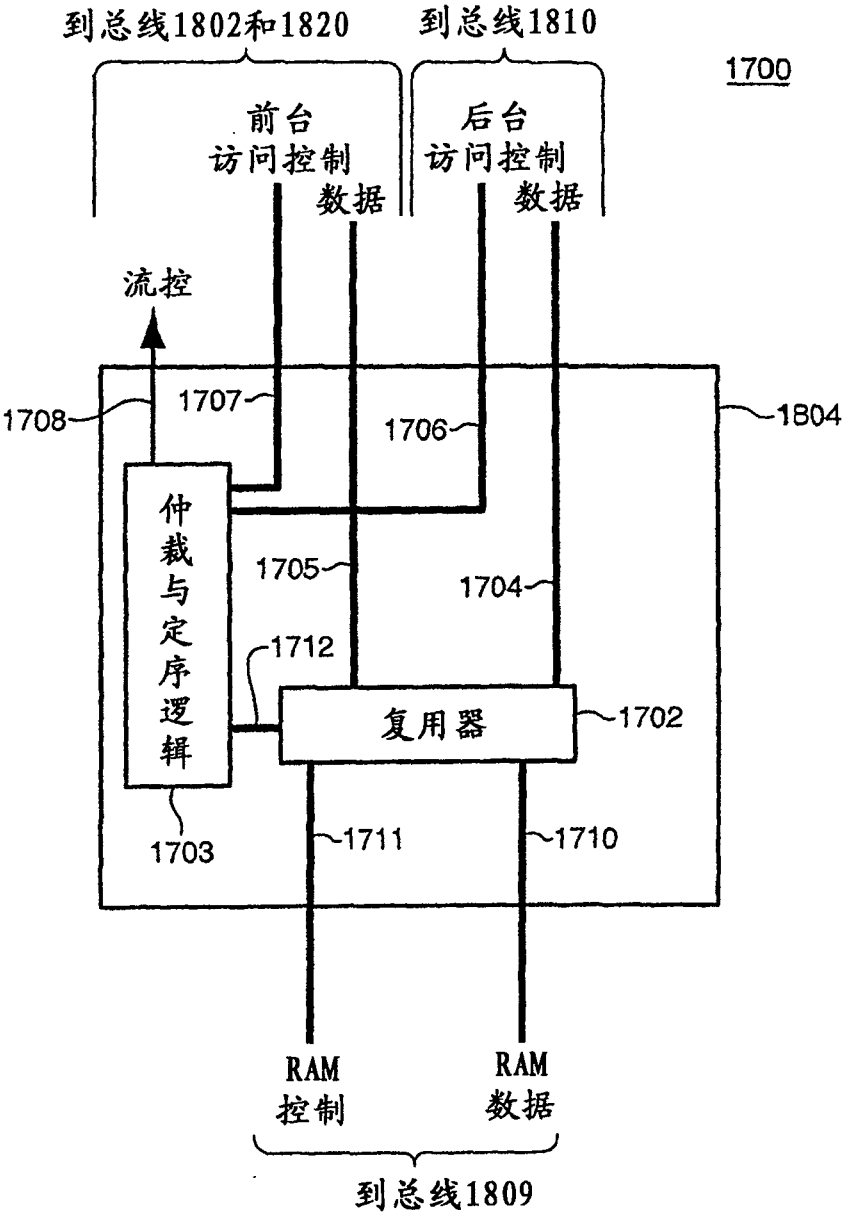

到总线1802和1820
到总线1810
1700
前台
访问控制
数据
后台
访问控制
数据
流控
1708
1707
1706
1804
仲裁与定序逻辑
1705
1704
1712
复用器
1702
1703
1711
1710
RAM
控制
RAM
数据
到总线1809

图18

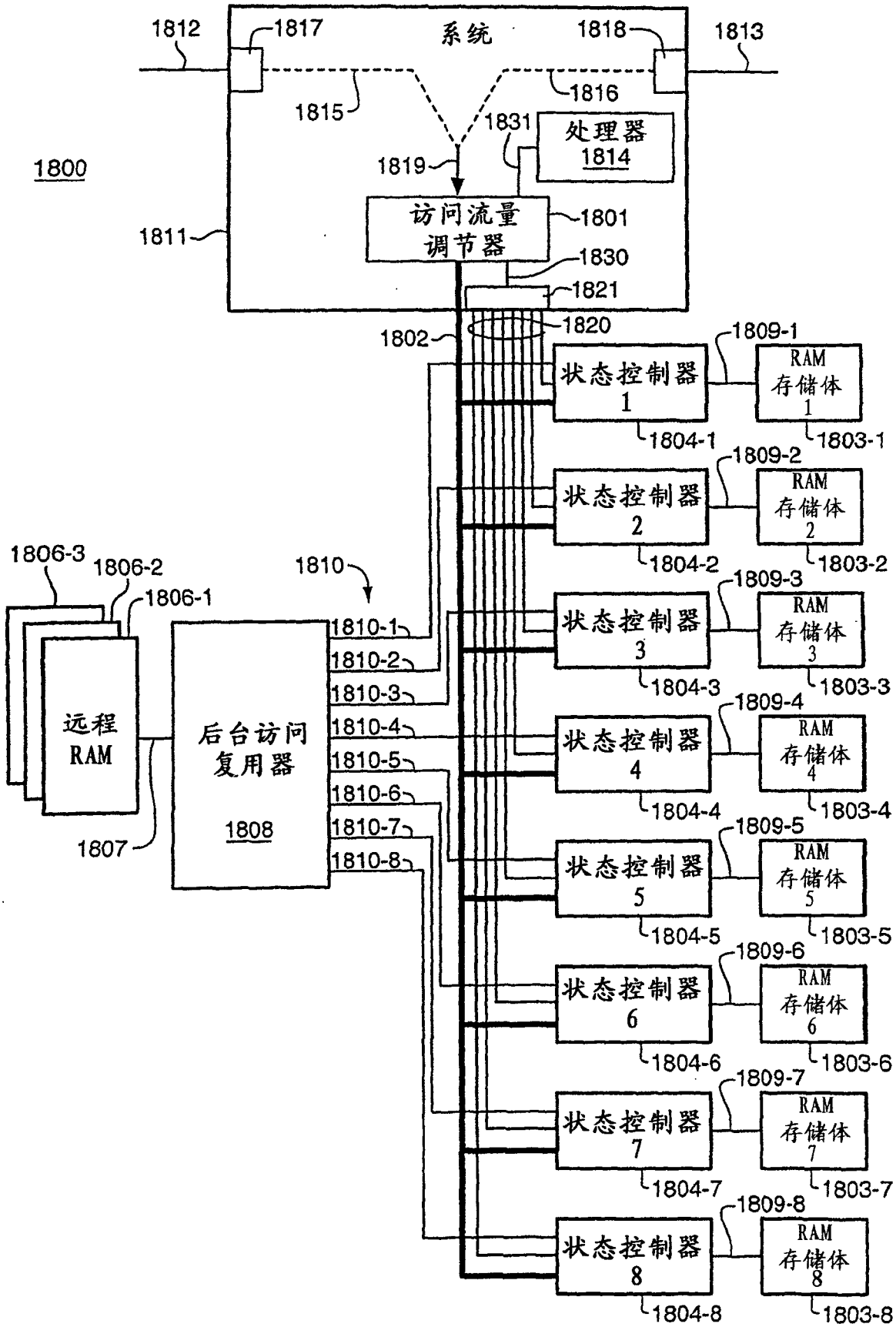

1812
1817
系统
1818
1813
1815
1816
1831
处理器
1814
1800
1819
访问流量
调节器
1801
1811
1830
1821
1820
1802
1809-1
状态控制器
1
RAM
存储体
1
1804-1
1803-1
1809-2
状态控制器
2
RAM
存储体
2
1804-2
1803-2
1806-3
1806-2
1806-1
1810
1809-3
状态控制器
3
RAM
存储体
3
1804-3
1803-3
1810-1
1810-2
1810-3
1809-4
状态控制器
4
RAM
存储体
4
1804-4
1803-4
远程
RAM
后台访问
复用器
1808
1810-4
1810-5
1810-6
1810-7
1810-8
1809-5
状态控制器
5
RAM
存储体
5
1804-5
1803-5
1807
1809-6
状态控制器
6
RAM
存储体
6
1804-6
1803-6
1809-7
状态控制器
7
RAM
存储体
7
1804-7
1803-7
1809-8
状态控制器
8
RAM
存储体
8
1804-8
1803-8

图19

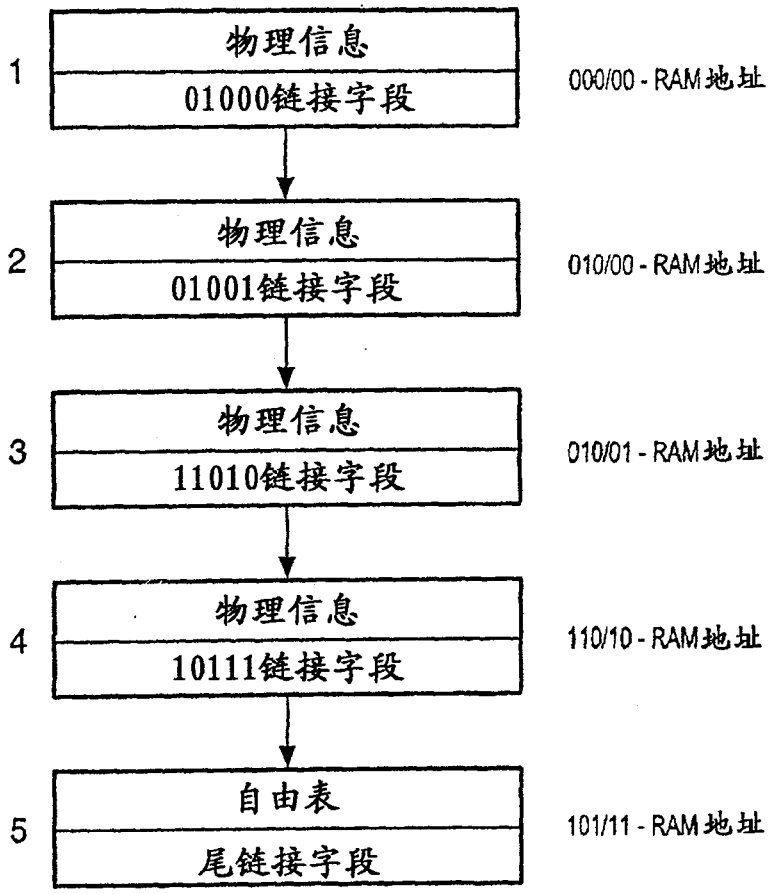

1
物理信息
01000链接字段
000/00 - RAM地址
2
物理信息
01001链接字段
010/00 - RAM地址
3
物理信息
11010链接字段
010/01 - RAM地址
4
物理信息
10111链接字段
110/10 - RAM地址
5
自由表
尾链接字段
101/11 - RAM地址

图20

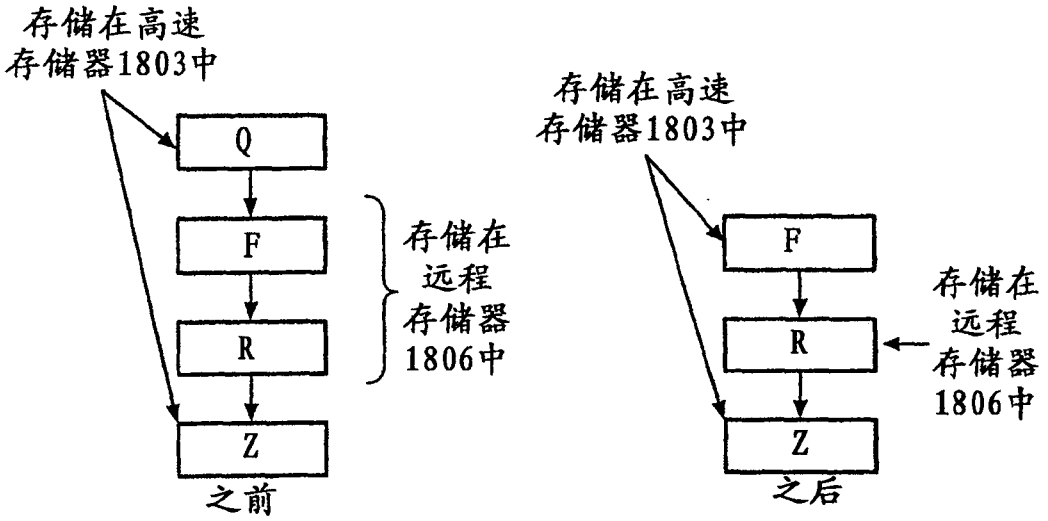

存储在高速
存储器1803中
Q
F
R
Z
之前
存储在
远程
存储器
1806中
存储在高速
存储器1803中
F
R
Z
之后
存储在
远程
存储器
1806中

图21

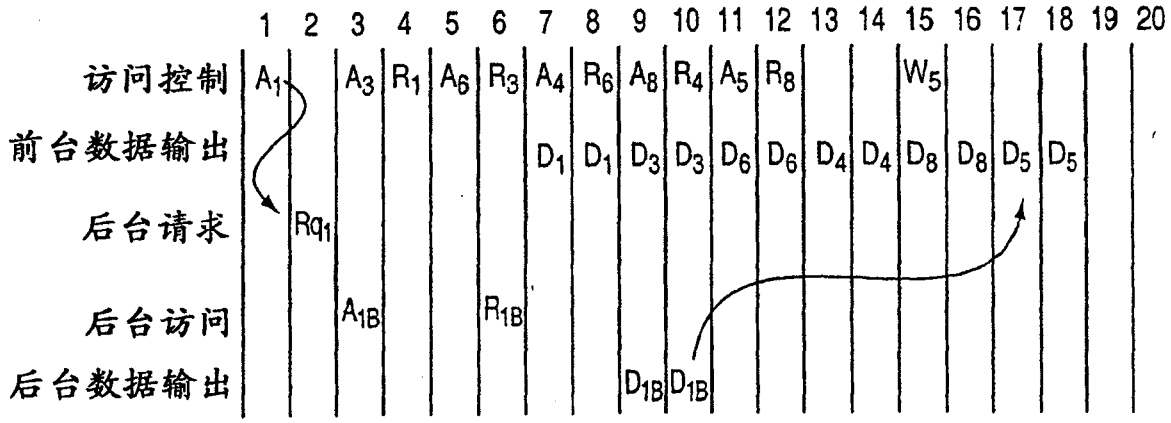

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
访问控制
前台数据输出
后台请求
后台访问
后台数据输出
A1 A3 R1 A6 R3 A4 R6 A8 R4 A5 R8 W5
D1 D1 D3 D3 D6 D6 D4 D4 D8 D8 D5 D5
Rq1
A1B
R1B
D1B D1B

图22

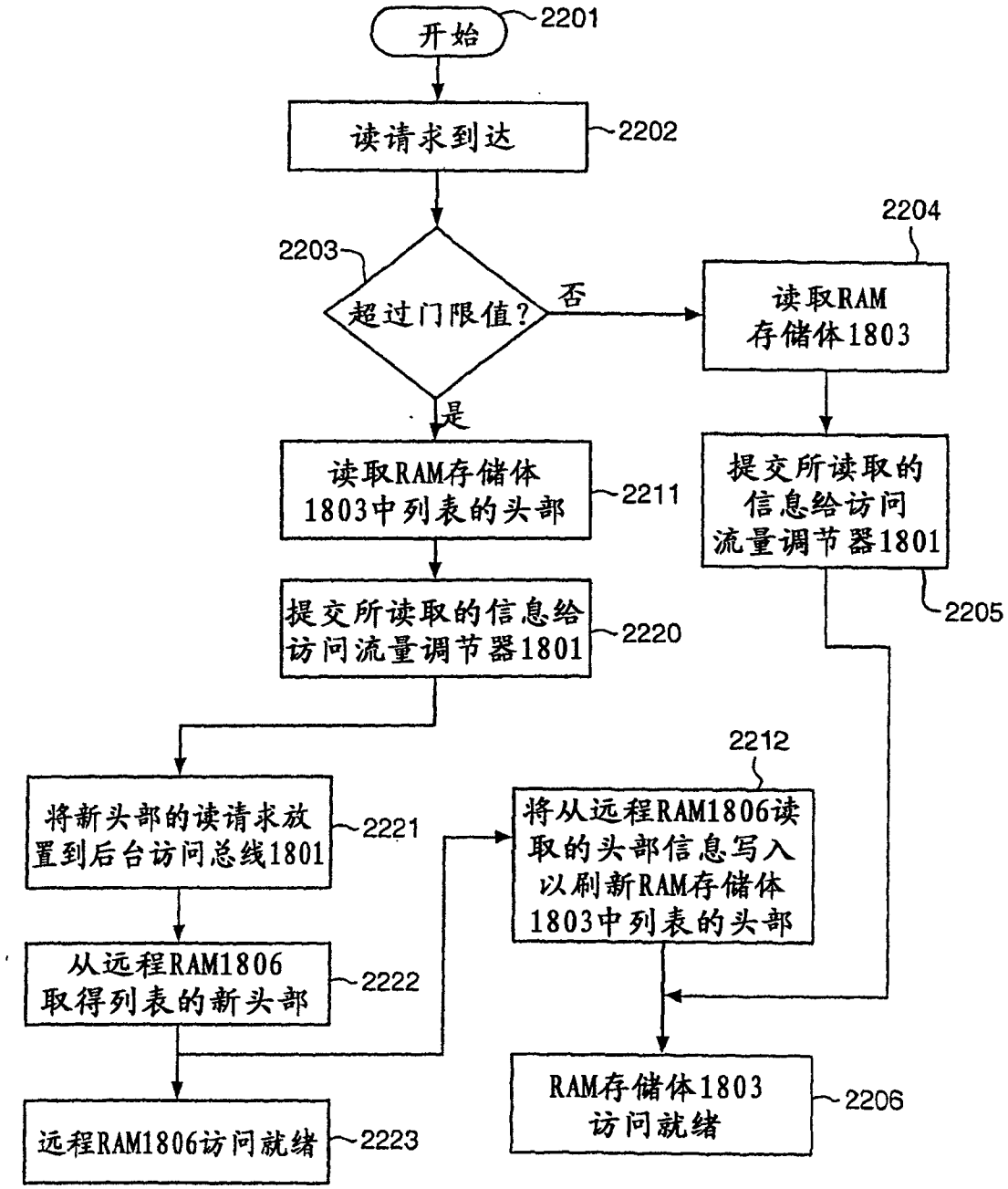

开始
2201
读请求到达
2202
2203
超过门限值?
否
是
2204
读取RAM
存储体1803
读取RAM存储体
1803中列表的头部
2211
提交所读取的
信息给访问
流量调节器1801
2205
提交所读取的信息给
访问流量调节器1801
2220
2212
将新头部的读请求放
置到后台访问总线1801
2221
将从远程RAM1806读
取的头部信息写入
以刷新RAM存储体
1803中列表的头部
从远程RAM1806
取得列表的新头部
2222
RAM存储体1803
访问就绪
2206
远程RAM1806访问就绪
2223

图23

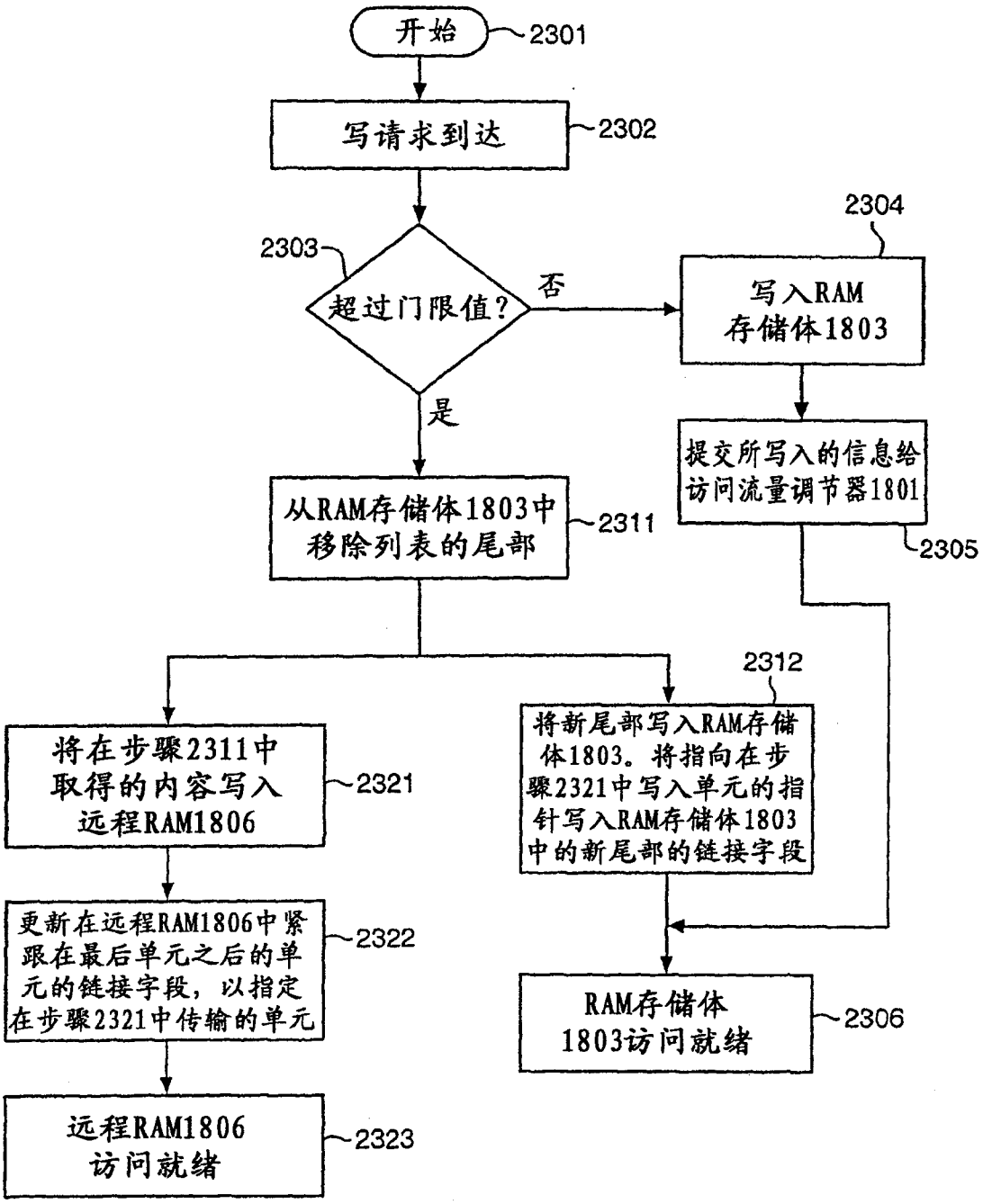

开始
2301
写请求到达
2302
2303
超过门限值?
否
2304
写入RAM
存储体1803
是
提交所写入的信息给
访问流量调节器1801
2305
从RAM存储体1803中
移除列表的尾部
2311
2312
将在步骤2311中
取得的内容写入
远程RAM1806
2321
将新尾部写入RAM存储
体1803。将指向在步
骤2321中写入单元的指
针写入RAM存储体1803
中的新尾部的链接字段
更新在远程RAM1806中紧
跟在最后单元之后的单
元的链接字段，以指定
在步骤2321中传输的单元
2322
RAM存储体
1803访问就绪
2306
远程RAM1806
访问就绪
2323

图24

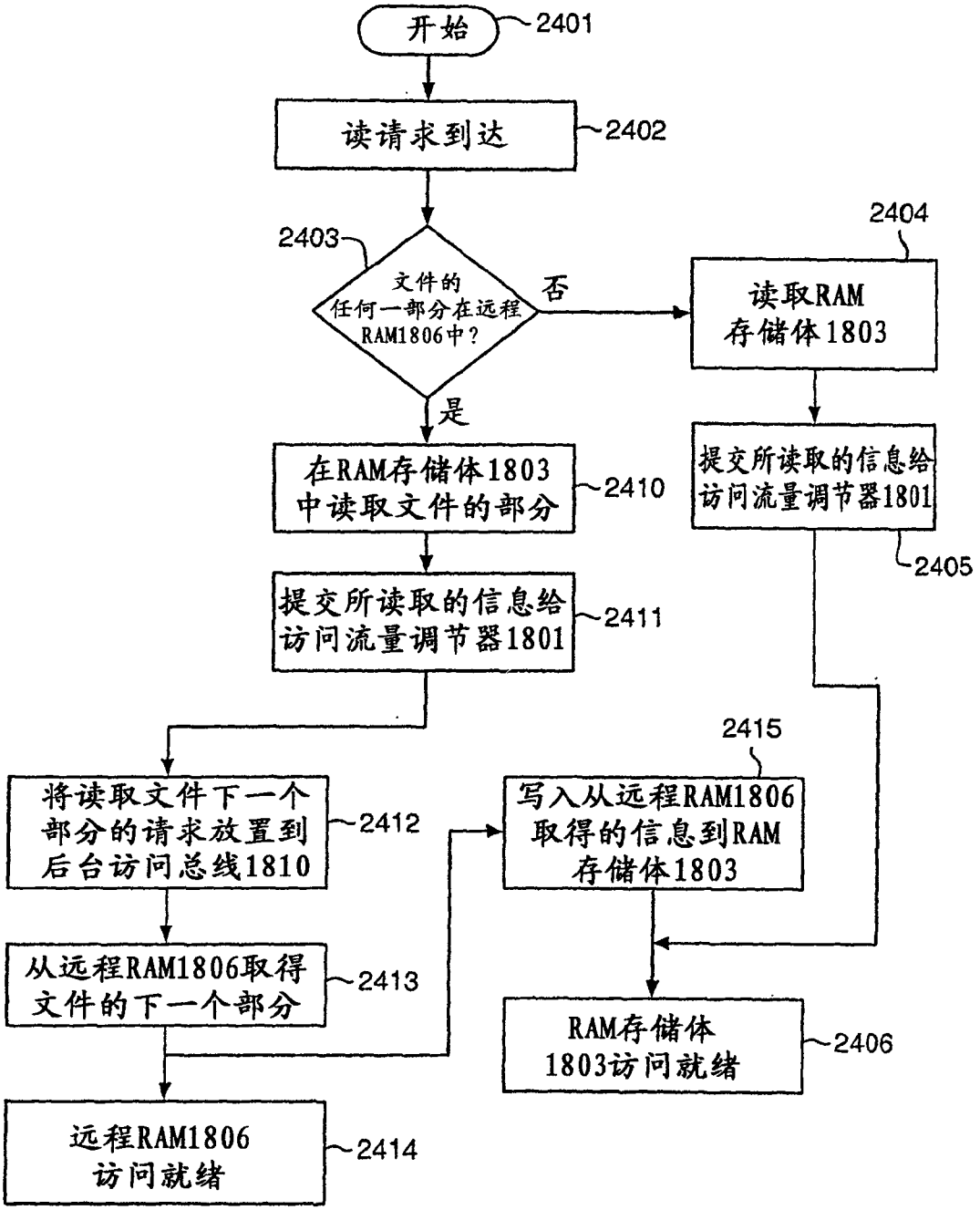
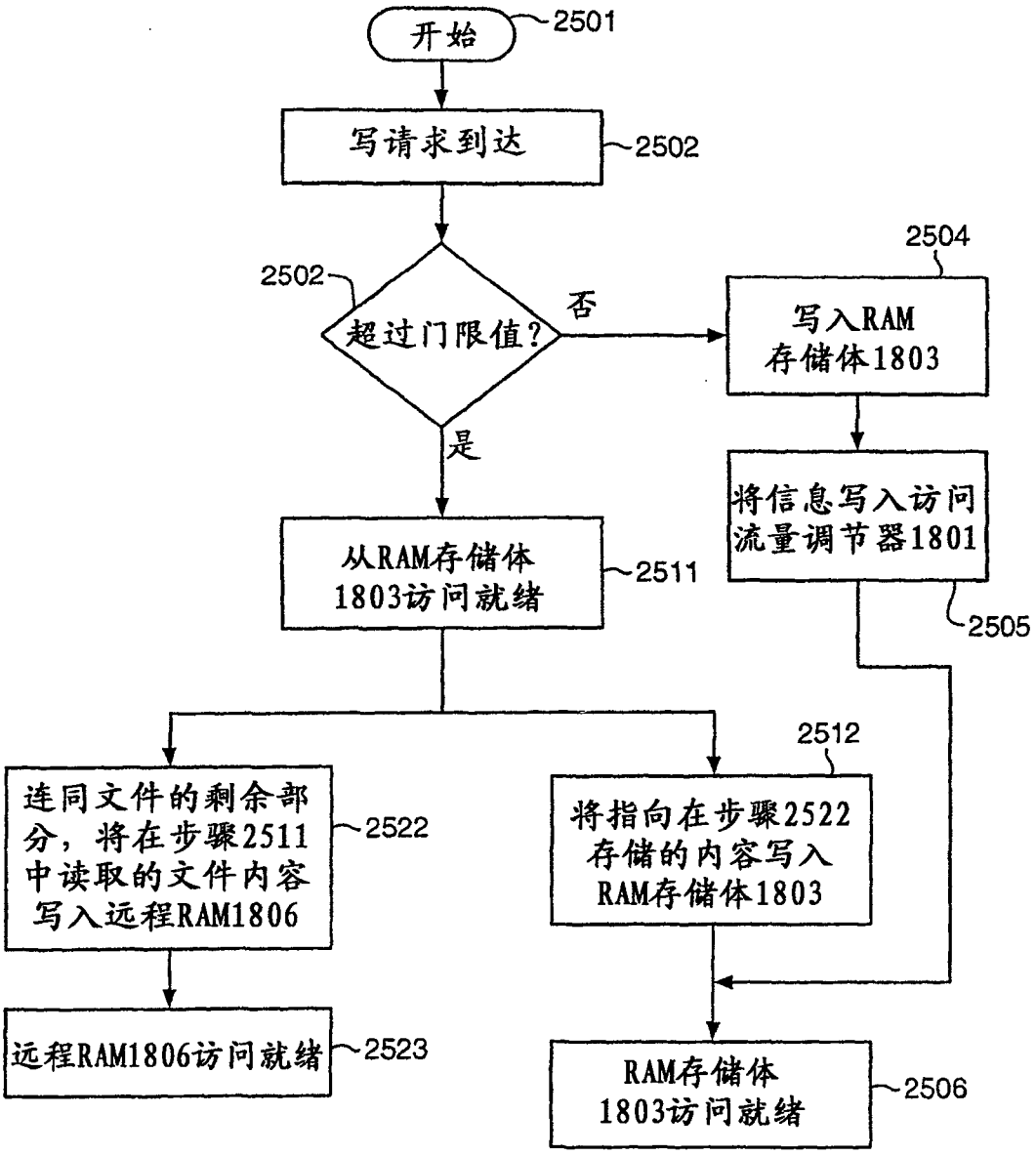

开始
2401
读请求到达
2402
2403
文件的任何一部分在远程RAM1806中？
否
是
2404
读取RAM存储体1803
提交所读取的信息给访问流量调节器1801
2405
在RAM存储体1803中读取文件的部分
2410
提交所读取的信息给访问流量调节器1801
2411
2415
将读取文件下一个部分的请求放置到后台访问总线1810
2412
写入从远程RAM1806取得的信息到RAM存储体1803
从远程RAM1806取得文件的下一个部分
2413
RAM存储体1803访问就绪
2406
远程RAM1806访问就绪
2414

图25


开始
2501
写请求到达
2502
2502
超过门限值?
否
是
2504
写入RAM
存储体1803
将信息写入访问
流量调节器1801
2505
从RAM存储体
1803访问就绪
2511
连同文件的剩余部
分，将在步骤2511
中读取的文件内容
写入远程RAM1806
2522
远程RAM1806访问就绪
2523
2512
将指向在步骤2522
存储的内容写入
RAM存储体1803
RAM存储体
1803访问就绪
2506