



(12) 发明专利

(10) 授权公告号 CN 101300556 B

(45) 授权公告日 2010.05.19

(21) 申请号 200680041242.9

W·E·毛勒 R·特雷梅恩

(22) 申请日 2006.11.28

(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 11038

(30) 优先权数据

代理人 李颖

11/289, 193 2005.11.28 US

(51) Int. Cl.

G06F 13/16(2006.01)

(85) PCT申请进入国家阶段日

2008.05.04

(56) 对比文件

CN 1480842 A, 2004.03.10, 全文.

(86) PCT申请的申请数据

CN 1653741 A, 2005.08.10, 全文.

PCT/EP2006/068984 2006.11.28

WO 2005038660 A2, 2005.04.28, 全文.

(87) PCT申请的公布数据

WO2007/060250 EN 2007.05.31

审查员 富瑤

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 P·W·克蒂欧斯 K·C·高沃

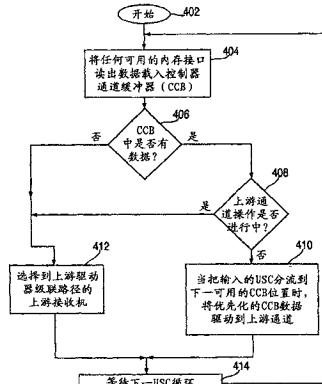
权利要求书 3 页 说明书 6 页 附图 4 页

(54) 发明名称

支持内存系统中不定的读出数据等待时间的方法和系统

(57) 摘要

提供了一种在内存系统中提供不定读出数据等待时间的方法和系统。该方法包括确定是否已经接收本地数据分组。如果已经接收本地数据分组，则将该本地数据分组存储在缓冲器设备中。该方法还包括确定缓冲器设备是否包含数据分组以及确定通过上游通道将数据分组发送到内存控制器的上游驱动器是否空闲。如果缓冲器包含数据分组且上游驱动器空闲，则将数据分组发送到上游驱动器。该方法进一步包括确定是否已经接收上游数据分组。该上游数据分组具有包括帧起始指示符和识别标志的帧格式，所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联。如果已经接收上游数据分组且上游驱动器不空闲，则将该上游数据分组存储在缓冲器设备中。如果已经接收上游数据分组且缓冲器设备不包含数据分组且上游驱动器空闲，则将该上游数据分组发送到上游驱动器。如果上游驱动器不空闲，则继续将进行中的任何数据分组发送到上游驱动器。



1. 一种支持不定的读出数据等待时间的方法,所述方法包括:

确定是否已经接收本地数据分组;

如果已经接收本地数据分组,则将所述本地数据分组存储在缓冲器设备内;

确定所述缓冲器设备是否包含数据分组;

确定通过上游通道将数据分组传送到内存控制器的上游驱动器是否空闲;

如果所述缓冲器设备包含数据分组且所述上游驱动器空闲,则将所述数据分组传送到所述上游驱动器;

确定是否已经接收上游数据分组,所述上游数据分组具有包括帧起始指示符和识别标志的帧格式,所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联;

如果已经接收上游数据分组且上游驱动器不空闲,则将所述上游数据分组存储在所述缓冲器设备中;

如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲,则将所述上游数据分组发送到所述上游驱动器;以及

如果所述上游驱动器不空闲,则继续发送进行中的任何数据分组。

2. 如权利要求1所述的方法,其中周期性地执行所述确定是否已经接收本地数据分组、所述确定所述缓冲器设备是否包含数据分组、所述确定上游驱动器是否空闲以及所述确定是否已经接收上游数据分组。

3. 如权利要求2所述的方法,其中周期性是每个上游通道循环一次。

4. 如权利要求1所述的方法,其中所述缓冲器设备包含多个数据分组,所述数据分组基于优先化算法被选择。

5. 如权利要求4所述的方法,其中所述优先化算法基于对应于所述数据分组的读取指令的年龄选择所述数据分组。

6. 如权利要求4所述的方法,其中所述优先化算法基于与所述数据分组相关的优先级选择所述数据分组。

7. 如权利要求1所述的方法,其中所述帧格式进一步包括总线循环冗余码(CRC)字段。

8. 如权利要求7所述的方法,其中所述将所述本地数据分组存储在缓冲器设备中包括格式化所述本地数据分组,所述格式化包括将所述本地数据分组串行化到所述帧格式中以及将值插入所述帧起始指示符、识别标志和总线CRC。

9. 一种内存系统中的控制中心设备,所述控制中心设备包括:

接收数据分组的设备,所述设备包括从下游控制中心设备接收上游数据分组的上游接收机和从本地存储设备接收本地数据分组的内存接口,其中每个数据分组具有包括帧起始指示符和识别标志的帧格式,所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联;

通过上游通道将数据分组发送到内存控制器的上游驱动器;以及

确定是否已经接收本地数据分组的装置;

如果已经接收本地数据分组,则将所述本地数据分组存储在缓冲器设备中的装置;

确定所述缓冲器设备是否包含数据分组的装置;

确定所述上游驱动器是否空闲的装置;

如果所述缓冲器设备包含数据分组且所述上游驱动器空闲,则将所述数据分组发送给所述上游驱动器的装置;

确定是否已经接收上游数据分组的装置,所述上游数据分组具有包括帧起始指示符和识别标志的帧格式,所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联;

如果已经接收上游数据分组且所述上游驱动器不空闲,则将所述上游数据分组存储在所述缓冲器设备中的装置;

如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲,则将所述上游数据分组发送到所述上游驱动器的装置;以及

如果所述上游驱动器不空闲,则继续发送传输进行中的任何数据分组的装置。

10. 如权利要求 9 所述的控制中心设备,其中所述上游通道是雏菊链式通道。

11. 如权利要求 9 所述的控制中心设备,其中所述控制中心设备物理地位于内存模块上。

12. 如权利要求 9 所述的控制中心设备,进一步包括所述缓冲器设备。

13. 如权利要求 9 所述的控制中心设备,其中周期性地执行所述确定是否已经接收本地数据分组、所述确定所述缓冲器设备是否包含数据分组、所述确定上游驱动器是否空闲以及所述确定是否已经接收上游数据分组。

14. 如权利要求 13 所述的控制中心设备,其中周期性是每个上游通道循环一次。

15. 如权利要求 9 所述的控制中心设备,其中所述缓冲器设备包含多个数据分组且所述数据分组基于优先化算法被选择。

16. 如权利要求 15 所述的控制中心设备,其中所述优先化算法基于对应于所述数据分组的读取指令的年龄选择所述数据分组。

17. 如权利要求 15 所述的控制中心设备,其中所述优先化算法基于与所述数据分组相关的优先级选择所述数据分组。

18. 一种内存系统,包括:

一个或多个内存模块,具有通过雏菊链式通道连接于内存控制器的一个或多个内存设备,其中利用包括识别标志和帧起始指示符的帧格式将读出数据返回所述内存控制器;以及

内存模块上用于缓冲地址、命令和数据的一个或多个控制中心设备,所述控制中心设备包括支持不定的读出数据等待时间的装置,其中所述支持不定的读出数据等待时间的装置包括:

确定是否已经接收本地数据分组的装置;

如果已经接收本地数据分组,则将所述本地数据分组存储在位于所述控制器通道缓冲器内的缓冲器设备中的装置;

确定所述缓冲器设备是否包含数据分组的装置;

确定通过上游通道向所述内存控制器传送数据分组的上游驱动器是否空闲的装置;

如果所述缓冲器设备包含数据分组且所述上游驱动器空闲,则将所述数据分组发送给所述上游驱动器的装置;

确定是否已经接收上游数据分组的装置;

如果已经接收上游数据分组且所述上游驱动器不空闲，则将所述上游数据分组存储在所述缓冲器设备中的装置；

如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲，则将所述上游数据分组发送到所述上游驱动器的装置；以及

如果所述上游驱动器不空闲，则继续发送传输进行中的任何数据分组的装置。

19. 如权利要求 18 所述的内存系统，包括雏菊链式通道中的点对点链接。

20. 如权利要求 18 所述的内存系统，其中所述控制中心设备进一步包括控制器通道缓冲器卸载优先化算法。

21. 一种内存系统，包括：

一个或多个内存模块，具有通过雏菊链式通道连接于内存控制器的一个或多个内存设备，其中利用包括识别标志和帧起始指示符的帧格式将读出数据返回所述内存控制器；以及

连接于用于缓冲地址、命令和数据的所述内存模块的一个或多个控制中心设备，所述控制中心设备包括支持不定的读出数据等待时间的装置，其中所述支持不定的读出数据等待时间的装置包括：

确定是否已经接收本地数据分组的装置；

如果已经接收本地数据分组，则将所述本地数据分组存储在位于所述控制器通道缓冲器内的缓冲器设备中的装置；

确定所述缓冲器设备是否包含数据分组的装置；

确定通过上游通道向所述内存控制器传送数据分组的上游驱动器是否空闲的装置；

如果所述缓冲器设备包含数据分组且所述上游驱动器空闲，则将所述数据分组发送给所述上游驱动器的装置；

确定是否已经接收上游数据分组的装置；

如果已经接收上游数据分组且所述上游驱动器不空闲，则将所述上游数据分组存储在所述缓冲器设备中的装置；

如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲，则将所述上游数据分组发送到所述上游驱动器的装置；以及

如果所述上游驱动器不空闲，则继续发送传输进行中的任何数据分组的装置。

22. 如权利要求 21 所述的存储器系统，包括雏菊链式通道中的点对点链接。

23. 如权利要求 21 所述的存储器系统，其中所述控制中心设备进一步包括控制器通道缓冲器卸载优先化算法。

支持内存系统中不定的读出数据等待时间的方法和系统

技术领域

[0001] 本发明涉及内存系统以及可以在内存系统中提供的方法和组件。更为具体地，本发明涉及读出数据的流控制和通过内存系统中的控制中心设备返回到内存控制器的读出数据的识别。

背景技术

[0002] 实现本发明的内存系统可以由通过雏菊链式通道 (daisy chainedchannel) 连接于内存控制器的控制中心设备组成。控制中心设备可以被附在、或者驻留于包含内存设备的内存模块上。

[0003] 许多高性能的计算主内存系统采用通过一个或多个通道连接于内存控制器的多个完全缓冲的内存模块。内存模块包含控制中心设备和多个内存设备。控制中心设备在内存控制器和内存设备之间完全地缓冲命令、地址和数据信号。利用分级等待时间或者依赖于位置的等待时间技术控制读出数据的流。在这两种情况下，内存控制器能够预测从内存模块请求的读出数据的返回时间并且调度命令以在通过每个内存模块将读出数据合并到控制器接口时避免冲突。

[0004] 在某些情况下，内存控制器能够连同读取命令一起发出读出数据延迟添加器。这将指示目标控制中心设备将额外的延迟加到读出数据的返回以便简化命令的发出并避免冲突。就所有情况而言，读出数据必须按照其被请求的顺序返回。此外，总的读出数据等待时间必须是通过内存控制器完全可预测的。在运行时间操作的过程中，这两个限制导致了额外的间隙被加在从内存模块返回的读出数据的分组上。这就给平均读出操作增加了等待时间。此外，控制中心无法使用非预定义的等待时间技术（以下被称为“不定的”等待时间技术）来返回比正常的快或者慢的读出数据。这些技术包括，但是不限于本地高速缓存读出数据、推理性地读取内存设备、独立地管理内存设备地址页、数据压缩等等。

[0005] 为了在现实工作量的条件下优化平均读出数据等待时间并且允许先进的控制中心设备能力，所需要的是允许内存模块在非预测时间将读出数据返回内存控制器的方法。这必须以不破坏读出数据并且允许内存控制器识别每个读出数据分组的方法完成。在控制中心设备将本地读出数据合并到级联的内存控制器通道时，通过避免数据冲突来防止数据损坏是特别复杂的。

发明内容

[0006] 本发明的第一方面提供一种避免受限于预定义的返回数据时间的方法，从而支持不定的（即非预定的）读出数据等待时间。在一个实施例中，所述方法包括确定是否已经接收本地数据分组。如果已经接收本地数据分组，则将所述本地数据分组存储在缓冲器设备中。所述方法还包括确定所述缓冲器设备是否包含数据分组并且确定通过上游通道将数据分组发送到内存 (memory) 控制器的上游驱动器是否空闲。如果所述缓冲器包含数据分组且所述上游驱动器空闲，则将所述数据分组发送到所述上游驱动器。所述方法进一步包

括确定是否已经接收上游数据分组。所述上游数据分组具有包括帧起始指示符和识别标志的帧格式，所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联。如果已经接收上游数据分组且所述上游驱动器不空闲，则将所述上游数据分组存储在缓冲器设备中。如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲，则将所述上游数据分组发送到所述上游驱动器。如果所述上游驱动器不空闲，则继续将进行中的任何数据分组发送到所述上游驱动器。

[0007] 本发明的另一方面提供了一种内存系统中的控制中心设备 (hubdevice)。根据一个实施例，所述控制中心设备包括接收数据分组的设备、通过上游通道将数据分组发送到内存控制器的上游驱动器以及包括便于实现不定读出数据等待时间的指令的机构。所述接收数据分组的设备包括从下游控制中心设备接收上游数据分组的上游接收机和从本地存储设备接收本地数据分组的内存接口。每个数据分组具有包括帧起始指示符和识别标志的帧格式，所述识别标志被内存控制器用来将所述上游数据分组与其对应的读取指令关联。所述机构上的指令便于确定是否已经接收本地数据分组。如果已经接收本地数据分组，则将所述本地数据分组存储在缓冲器设备中。所述指令还便于确定所述缓冲器设备是否包含数据分组以及确定所述上游驱动器是否空闲。如果所述缓冲器设备包含数据分组且所述上游驱动器空闲，则将所述数据分组发送到所述上游驱动器。所述指令进一步便于确定是否已经接收上游数据分组。如果已经接收上游数据分组且所述上游驱动器不空闲，则将所述上游数据分组存储在所述缓冲器设备中。如果已经接收上游数据分组且所述缓冲器设备不包含数据分组且所述上游驱动器空闲，则将所述上游数据分组发送到所述上游驱动器。如果所述上游驱动器不空闲，则继续将进行中的任何数据分组发送到所述上游驱动器。

[0008] 本发明的又一方面提供了一种具有一个或多个内存模块的内存子系统。在一个实施例中，所述内存模块包括通过雏菊链式通道连接于内存控制器的一个或多个内存设备。利用包括识别标志和帧起始指示器的帧格式将读出数据返回所述内存控制器。所述内存系统还包括内存模块上的用于缓冲地址、命令和数据的一个或多个控制中心设备。所述控制中心设备包括与抢先本地数据合并算法一同用于最小化读出数据等待时间并且支持到内存控制器的不定（即非预定的）读出数据返回时间的控制器通道缓冲器。

[0009] 本发明的另一方面提供了具有一个或多个内存模块的内存系统。在一个实施例中，所述内存模块包括通过雏菊链式通道连接于内存控制器的内存设备。利用包括识别标志和帧起始指示器的帧格式将读出数据返回所述内存控制器。所述内存系统还包括连接于用于缓冲地址、命令和数据的内存模块的一个或多个控制中心设备。所述控制中心设备包括与抢先本地数据合并算法一同用于最小化读出数据等待时间并且支持到内存控制器的不定（即非预定的）读出数据返回时间的控制器通道缓冲器。

附图说明

[0010] 举例来说，下面将根据附图更加详细地描述本发明的实施例，在附图中，同样的元件以同样的方式被编号，其中：

[0011] 图 1 示出通过点对点连接具有多级雏菊链式内存模块的示范性内存系统；

[0012] 图 2 示出具有通过雏菊链式通道连接于内存模块和内存控制器的控制中心设备的示范性内存系统；

- [0013] 图 3 示出可以由示范性实施例利用的控制中心逻辑设备；
- [0014] 图 4 示出在示范性实施例中由控制中心逻辑设备实现的示范性流程；以及
- [0015] 图 5 示出可以由示范性实施例利用的读出数据格式。

具体实施方式

[0016] 示范性的实施例利用控制器通道缓冲器 (CCB)、具有识别标志的读出数据帧格式和抢先数据合并技术来支持最小化的、不定的读出数据等待时间。示范性的实施例允许内存模块在非预测时间将读出数据返回内存控制器。识别标志信息被添加到读出数据分组，以指明数据作为其读取结果的读取命令以及从中读取数据的控制中心。控制器利用该识别标志信息将读出数据分组与由控制器发出的读取命令相匹配。通过利用识别标志信息，读出数据可以按照不同于对应的读取命令的发出顺序的顺序被返回。

[0017] 示范性的实施例在实现不定读出数据等待时间时还提供了抢先数据合并过程以防止上游通道上的数据冲突。将 CCB 添加到控制中心设备以临时存储读出数据。当内存模块上的内存设备读出数据时，该数据从内存接口被传送到缓冲器。当控制中心设备检测到上游数据分组（即从在该检测控制中心设备的下游的控制中心设备发送到控制器的数据分组）不在通过上游通道被传送到检测控制中心设备的中途（典型地，需要几个传送来发送完整的数据分组），该检测控制中心设备进行检查以了解是否有读出数据分组在其 CCB 中等待被向上游发送。如果控制中心设备检测到该 CCB 中的读出数据分组，则它将该读出数据分组从 CCB 驱动到上游数据总线上。同时，如果新的上游数据分组通过上游数据总线被接收，则该数据分组被存储在控制中心设备上的 CCB 中。照这样，向上游来的数据分组不会与从控制中心设备上的 CCB 向上游发送的数据分组冲突。在 CCB 中有多于一个数据分组的情况下，可以执行多种方法来确定下次发送哪个数据分组（例如，来自最老的读取命令的数据分组可能先被发送）。

[0018] 如图 1 所示，示范性的实施例适用于由通过雏菊链式内存通道 114 连接于内存控制器 102 的一个或多个内存模块 110 构建的内存系统。内存模块 110 包含相对于控制器内存通道 114 缓冲命令、地址和数据信号的控制中心设备 112，以及连接于控制中心设备 112 的一个或多个内存设备 108。内存通道 114 的下游部分，下游通道 104 将写入数据和内存操作命令发送到控制中心设备 112。控制器通道 114 的上游部分，上游通道 106 返回被请求的读出数据（于此被称为上游数据分组）。

[0019] 图 2 示出包括内存系统的可供选择的示范性实施例，所述内存系统由连接于通过雏菊链式内存通道 114 进一步连接于内存控制器 102 的控制中心设备 112 的一个或多个内存模块 110 构建。在本实施例中，控制中心设备 112 不是位于内存模块 110 上；而是控制中心设备 112 与内存模块 110 通信。在图 2 中，内存模块 110 可通过多支路 (multi-drop) 连接和 / 或点对点连接与控制中心设备 112 通信。其他硬件配置也是可能的，例如示范性实施例可以只利用单级雏菊链式控制中心设备 112 和 / 或内存模块 110。

[0020] 图 3 示出被示范性的实施例用于执行在此所描述的处理的具有流控制逻辑 308 的控制中心设备 112。控制中心设备 112 和控制中心设备 112 内的组件可以在硬件和 / 或软件中实现。控制中心设备 112 通过接收机逻辑 304（于此也被称为上游接收机）在上游通道 104 上接收上游数据分组。上游数据分组是从在接收控制中心设备 112 下游的控制中心设

备 112 被发送到控制器 102 的数据分组。上游数据分组可以被发送到驱动器逻辑 306(于此也被称为上游驱动器)从而向着上游通道 106 上的控制器 102 被驱动,或者,如果上游通道 106 繁忙,则该上游数据分组可以被临时存储于控制中心设备 112 上的 CCB 310 中。该上游数据分组的目的地由流控制逻辑 308 确定并且通过发送信号到本地数据复用器 312 实现。

[0021] 在示范性的实施例中,CCB 310 或者缓冲器设备位于控制中心设备 112 中,并且安全地捕获在控制中心设备 112 将其本地数据分组合并到上游通道 106 上时、被分流到 CCB 310 中的上游数据传送(通过接收机逻辑 304)。本地数据分组是从附在由控制中心设备 112 控制的内存模块 110 上的内存设备 108 读出的数据分组。这些内存设备 108 于此也被称为本地存储设备。从本地存储设备读出的数据,本地数据分组为了通过上游驱动器在上游控制器接口上返回而被格式化并且被存储在 CCB 310 中。格式化包括将本地数据分组串行化在适当的帧格式(例如,参见图 5 中所示的示范性的帧格式)中,以及将值插入识别标志(来源于读取请求)、第一传送字段和总线循环冗余编码(CRC)字段。在示范性的实施例中,将本地数据分组的格式化作为将该本地数据分组存储在 CCB 310 中的一部分来执行。

[0022] 当数据分组在内存接口 302 被接收时,它在本地数据分组等待被合并到上游通道 106 上(通过驱动器逻辑 306)的时候被存储在 CCB 310 中。数据分组内的识别标志允许内存控制器 102 使返回的读出数据分组与其对应的读出数据请求命令相关联。数据分组还包含在上游读出数据帧(数据分组被格式化为读出数据帧)的开头附近的小的、易于解码的“start”或者第一传送(“ft”)字段(于此也被称为帧起始指示符),其表示读出数据帧存在于数据分组中。这被控制中心设备 112 中的流控制逻辑 308 用来监控通道读出数据活动。

[0023] 当 CCB 310 中有来自本地读出操作或者来自以前从下游控制中心设备分流的读出数据分组的数据时(CCB 中的数据分组于此被称为存储的数据分组),一旦该数据被允许,控制中心设备 112 就会通过驱动器逻辑 306 将其合并到上游通道 106 上。通道控制中心设备 112 在上游通道 106 空闲的任何时候,或者紧跟在目前进行中的数据分组的最后传送之后立即将本地数据合并到上游通道 106。利用此方法,读出数据帧绝不会被截开,但是在上游通道 106 上传输的还未到达控制中心设备 112 的本地数据复用器 312 的读出数据帧可能被抢先并被分流到 CCB 310 中。这允许上游通道 106 上读出数据中的间隙被最小化,这将在现实工作量的条件下提高总线效率并导致降低的平均读出数据等待时间。

[0024] 当 CCB 310 中存在多个读出数据分组时,控制中心设备 112 可以被配置为发送对应于最早的读取命令的读出数据分组。这将使得发到远离内存控制器 102 许多雏菊链式位置的控制中心设备 112 的读出请求上的不适当的等待时间最小化。也可以实现其他 CCB 310 卸载优先化算法。例如,读出数据帧的识别标志字段可以包含优先级字段。优先级字段可以用于指导 CCB 310 的卸载。可选择地,当读出数据被请求时,可以传送优先级信息。然后控制中心设备 112 可以将该识别标志与以前记录的优先级信息相比较以确定在 CCB 310 中的位置来发送下一个。还可以采用偶尔在高优先级数据之前发送低优先级数据的方法来确保低优先级数据不会被标志为较高优先级的请求完全延迟。

[0025] 图 4 为由示范性实施例中位于控制中心设备 112 中的流控制逻辑 308 促进的流程。图 4 所述的过程执行抢先本地数据合并并且可以由包括诸如流控制逻辑 308 中的有限

状态机等硬件和 / 或软件指令的机构实现。在示范性的实施例中，该过程开始于框 402 并且周期性（例如，在每个控制器通道传送或者上游通道循环之后）重复。在框 404，将内存接口 302 中的任何本地读出数据分组（即，来自附于控制中心设备 112 的内存模块 110 上的内存设备 108）装入 CCB 310。这确保流控制逻辑 308 知道并管理本地读出数据的上游驱动。在框 406，确定 CCB 310 中是否有数据。如果 CCB 310 中没有数据，则在框 412 将数据从接收机逻辑 304 路由到驱动器逻辑 306。通过将本地数据复用器 312 设置为发送上游数据分组到驱动器逻辑 306 以将上游数据分组驱动到上游通道 106 上朝向控制器 102，该路由由流控制逻辑 308 控制。然后，处理继续到 414，其中在下一上游通道循环中使处理返回框 404。

[0026] 如果在框 406 确定 CCB 中有数据，则执行框 408 以确定是否有上游通道操作在进行中（即，有正在通过驱动器逻辑 306 被驱动到上游通道 106 过程中的上游数据分组或者本地读出数据分组）。如果有上游通道操作在进行中（即驱动器繁忙），处理继续到框 412。在框 412，通过设置本地数据复用器 312 为发送上游数据分组到驱动器逻辑 306，将上游读出数据分组从接收机逻辑 304 路由到驱动器逻辑 306。可选择地，如果没有上游通道操作在进行中（即驱动器空闲）且 CCB 310 中有数据，则处理继续到框 410。在框 410，将来自 CCB 310 的数据驱动到上游通道 106，同时来自上游通道 106、在接收机逻辑 304 中接收的任何数据分组被分流（存储）到下一可用 CCB 310 位置。通过流控制逻辑 308 控制上游数据分组被装入 CCB 310 来执行分流。然后处理继续到 414，其中在下一上游通道循环中使处理返回框 404。

[0027] 图 5 是上游通道 106 上的上游数据分组和本地读出数据分组的示范性读出数据帧格式。图 5 中所述的帧格式使用 21 个信号通道且每个分组包括 16 个传送。它包括 1 比特第一起始指示符 502 和识别标志 504，以及具有用于传输错误检测的总线 CRC 508 的 256 比特（32 字节）读出数据 506。可以使用信号通道和传送深度的其他组合来创建与本发明兼容的包括帧起始指示符、读出数据识别标志和读出数据的帧格式。

[0028] 示范性的实施例涉及由连接于或者包含于内存模块上的雏菊链式控制中心逻辑设备构建的计算机内存系统。控制中心以雏菊形式链接在内存控制器通道上并且进一步被附在内存模块的内存设备上。内存控制器向将读出数据从内存模块合并到内存通道的控制中心发出对该读出数据的请求。利用通道缓冲器和分组识别标志，控制中心能够在不是内存控制器预测的时间、以及在可能抢先于较早已经发出的读出请求的时间返回读出数据，而不会释放或者破坏在通道上返回至内存控制器的任何读出数据。

[0029] 示范性的实施例可以用于通过更加完全地利用上游通道来优化平均读出数据等待时间。通过使用 CCB、具有识别标志的读出数据帧格式和抢先数据合并技术，可以执行不定读出数据等待时间以更加完全地利用控制器通道。

[0030] 如上所述，本发明的实施例可以以计算机可实现的进程和实施那些进程的设备的形式被具体化。本发明的实施例也可以以包含在有形介质，例如软盘、CD-ROM、硬盘或者任何其他计算机可读的存储介质中具体化的指令的计算机程序代码的形式被具体化，其中当计算机程序代码被载入计算机且被计算机执行时，该计算机就变为实施本发明的设备。本发明还可以以计算机程序代码的形式被具体化，例如，是否被存储在存储介质中、是否被载入计算机和 / 或由计算机执行、或者是否在某些传输介质上，例如在电线或者线缆上、通过

光纤、或者通过电磁辐射被传输，其中当该计算机程序代码被载入计算机且被计算机执行时，该计算机就变为实施本发明的设备。当该计算机程序代码在通用的微处理器上被执行时，计算机程序代码段配置微处理器以创建具体的逻辑电路。

[0031] 尽管已经根据实施例描述了本发明，本领域的技术人员应该理解可以对本发明做出各种变化并且可以以等价物替代其元件而不背离本发明的范围。此外，可以做出许多修改，从而使特定的情况或者材料适应于本发明的示教而不背离其主要范围。因此，意图是不将本发明限制于如实现本发明预期的最佳模式所公开的特定实施例，而是本发明将包括所有在附加的权利要求书的范围之内的实施例。而且，第一、第二等术语的使用不表示任何顺序或者重要性，而是将第一、第二等术语用于区分一个元件与另一个元件。

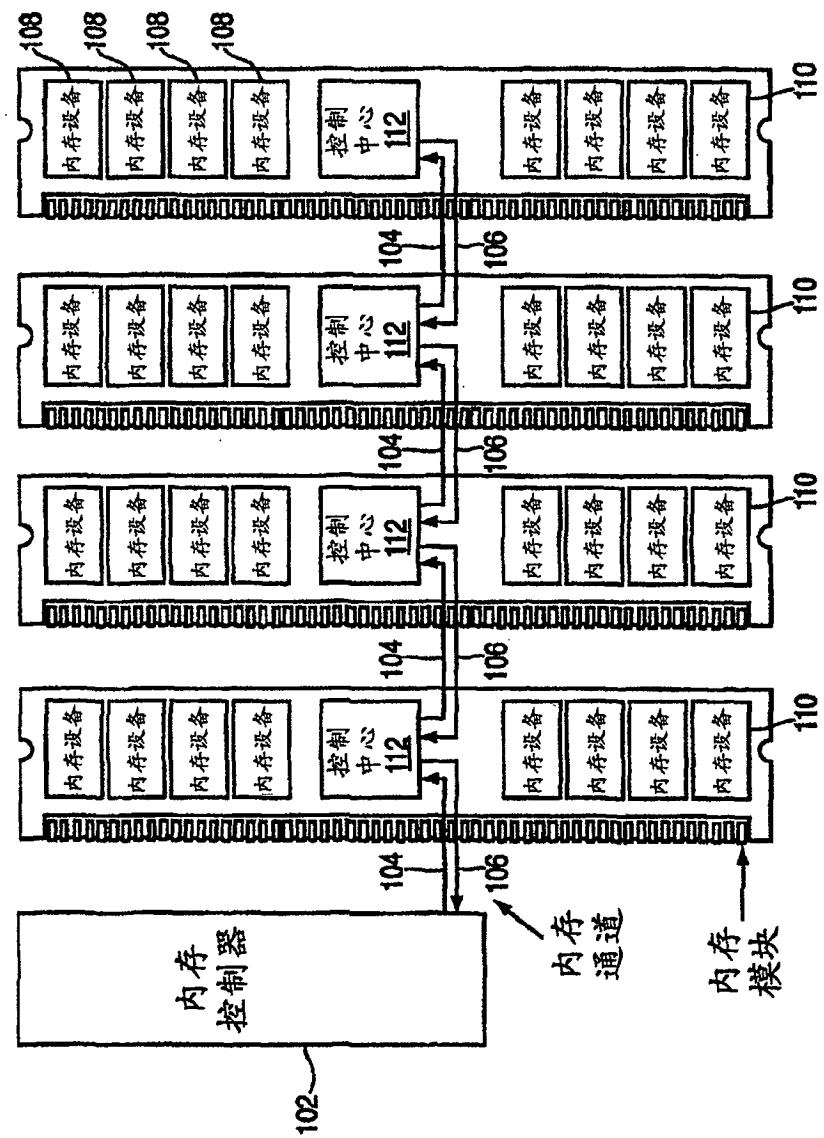


图 1

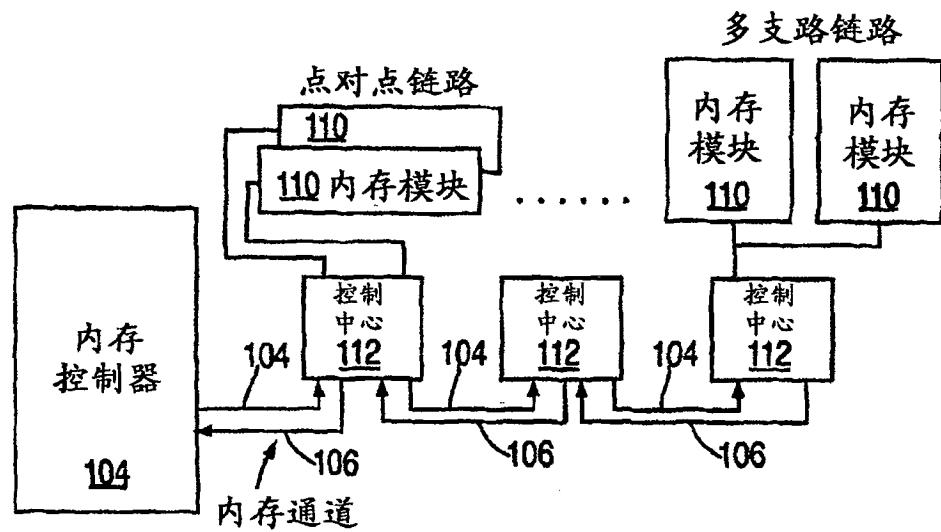


图 2

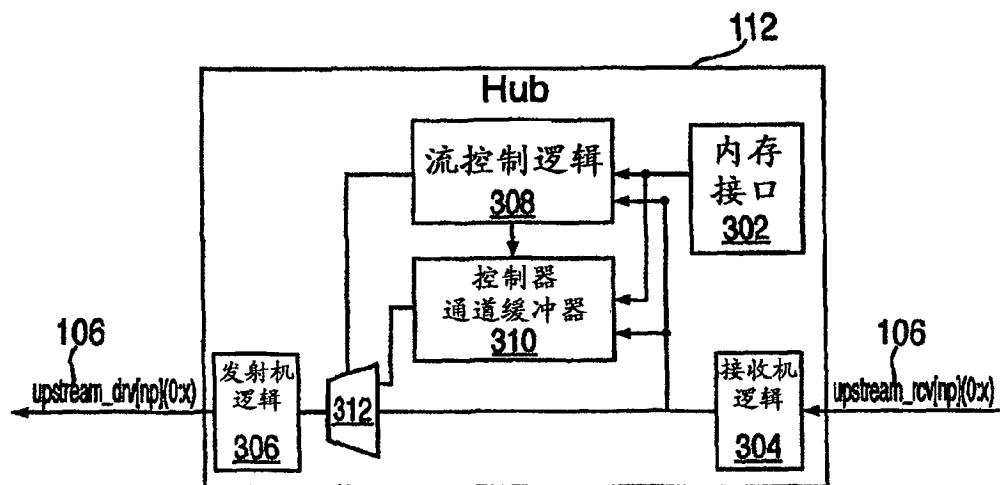


图 3

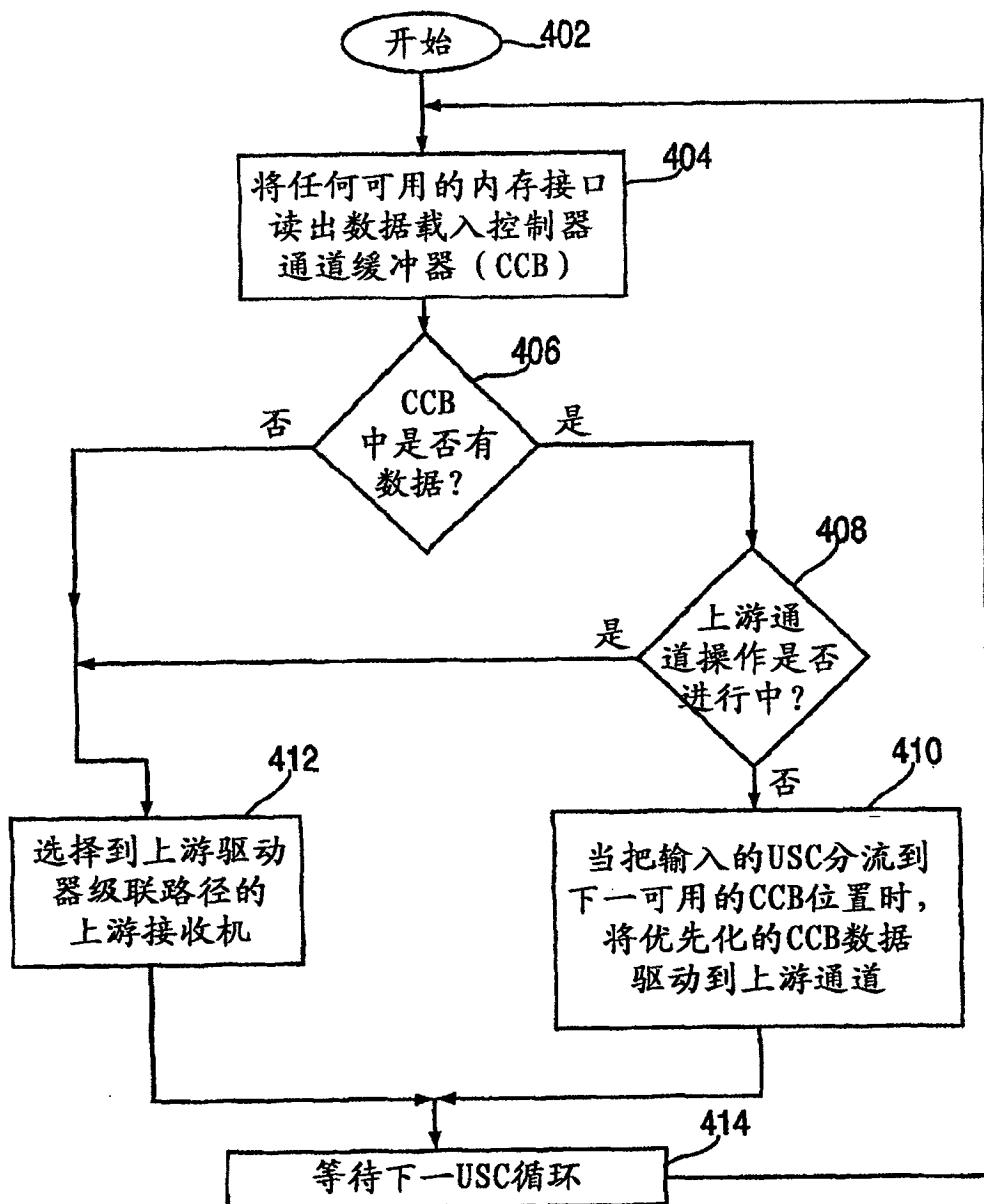


图 4

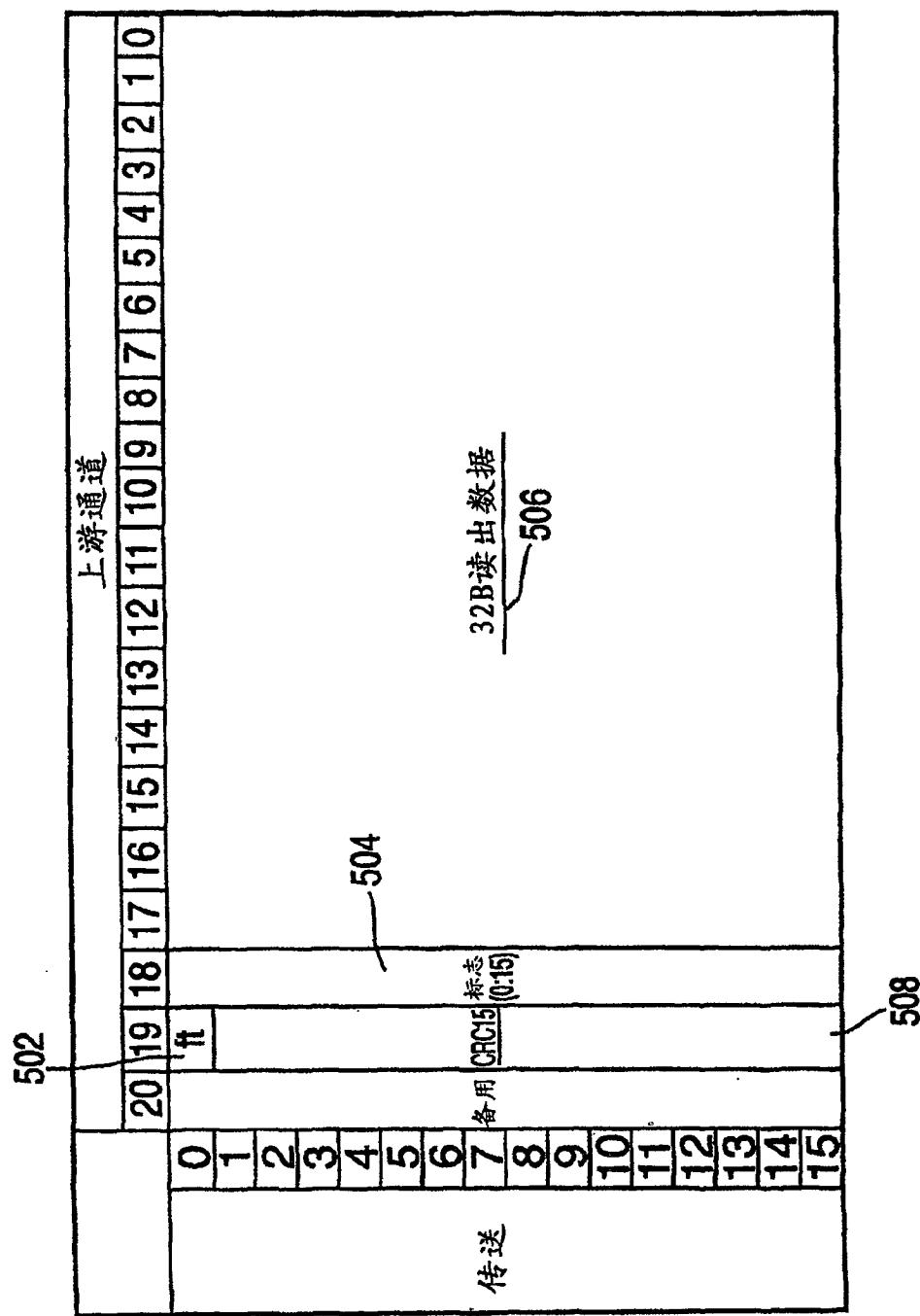


图 5