



## [12] 发明专利申请公开说明书

[21] 申请号 200380108479.0

[43] 公开日 2006 年 2 月 15 日

[11] 公开号 CN 1736066A

[22] 申请日 2003.11.11

[74] 专利代理机构 中国专利代理(香港)有限公司

[21] 申请号 200380108479.0

代理人 王 岳 陈景峻

[30] 优先权

[32] 2002.11.11 [33] GB [31] 0226249.1

[86] 国际申请 PCT/GB2003/004866 2003.11.11

[87] 国际公布 WO2004/045161 英 2004.5.27

[85] 进入国家阶段日期 2005.7.8

[71] 申请人 克利尔斯皮德科技有限公司

地址 英国布里斯托尔

[72] 发明人 A·斯潘塞

权利要求书 3 页 说明书 12 页 附图 4 页

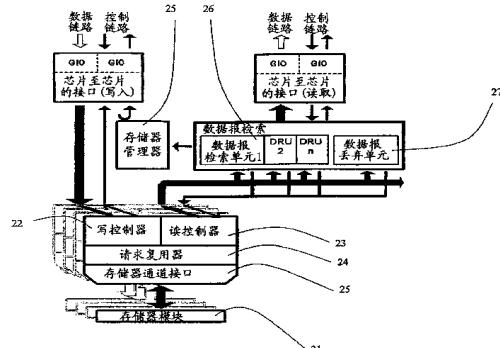
## [54] 发明名称

用于流量处理的包存储系统

## [57] 摘要

一种在通信系统中对可变大小的数据包进行排队的方法，包括：从输入数据包产生预定固定大小的记录部分并包含关于该包的信息，该包中的数据在数据部分中；在第一存储器 2 的独立存储器位置中存储数据部分，并且每一数据部分彼此之间没有联系；在第二存储器 3 的一个或多个所管理的队列中存储记录部分，其具有的固定大小存储器位置的大小等于记录部分的大小；其中：第一存储器比第二存储器大，并且具有比其小的地址带宽；并且第一存储器中的存储器位置设置成具有多个不同大小的块，并且根据该数据部分的大小将存储器位置分配给该数据部分。方便地，在设置成两个所述块的第一存储器中可以有两个存储器位置，一个用来接收相对较小的数据部分，并且另一个用来接收相对较大的数据部分，并且其中将太大而不能存储在单

个存储器块中的数据部分存储为多个块中的链接列表，其具有指向下一块的指针，但是没有任何从该包的一个数据部分指向下一数据部分的指针。块中的存储器位置优选地与该通信系统中最通常出现的数据包的大小匹配。优选地从有效地址的中央池成批地从提供给其的有效地址的池中分配在第一存储器中的存储器位置。



1. 一种在通信系统中对可变大小的数据包进行排队的方法，该方法包括：

5 从所述数据包产生预定固定大小的记录部分并包含关于该包的信息，该包中的数据在数据部分中；

在第一存储器的独立存储器位置中存储数据部分，并且每一数据部分彼此之间没有联系；

在第二存储器的一个或多个所管理的队列中存储记录部分，其具有的固定大小存储器位置的大小等于记录部分的大小；

10 其中：

第一存储器比第二存储器大，并且具有比其小的地址带宽；并且

第一存储器中的存储器位置被设置成具有多个不同大小的块，并且根据该数据部分的大小将存储器位置分配给该数据部分。

15 2. 如权利要求1中所述的方法，其中在设置成两个所述块的第一存储器中有两个存储器位置，一个用来接收相对较小的数据部分，并且另一个用来接收相对较大的数据部分，并且其中将太大而不能存储在单个存储器块中的数据部分存储为多个块中的链接列表，其具有指向下一快的指针，但是没有任何从该包的一个数据部分指向下一数据部分的指针。

20 3. 如权利要求1或权利要求2中所述的方法，其中该块中的存储器位置的大小与该通信系统中最通常出现的数据包的大小匹配。

4. 如权利要求1至3中任何一个所述的方法，进一步包括从有效地址的中央池成批地从提供给其的有效地址的池中分配在所述第一存储器中的存储器位置。

25 5. 如权利要求4中所述的方法，其中存储器块分隔成多个存储器通道，该方法进一步包括顺序地跨过通道将地址分配给数据部分，从而将数据部分的存储分散在通道上。

30 6. 如权利要求4中所述的方法，进一步包括以管线方式通过数据检索单元从第一存储器中读取数据部分，数据检索单元用于指导存储器块读出数据部分，而不必等待前一个读取完成，并且从该第一存储器中释放地址位置。

7. 如前述权利要求中任何一个所述的方法，进一步包括在所接收到

的包没有足够的存储器的情况下，将记录部分排队，如同对应的数据部分存储在第一存储器中，顺序读出对应于所述数据包的记录部分，设置标志用来表示将要丢弃所述包的数据部分，丢弃所述数据部分，并且释放名义上分配给所丢弃数据部分的存储器位置。

5 8. 如权利要求 6 中所述的方法，进一步包括从地址的位图中读取地址位置，并且当在存储器位置存储的数据已经被读出之后释放该存储器位置时，直接将所释放的存储器位置的地址发送到该池中。

9. 一种用于对所接收到的数据包进行排队的存储集线器，包括：  
到达块，其用来从所述数据包产生预定固定大小的记录部分并且包含关于该包的信息，该包中的数据在数据部分中；

第一存储器，用于在独立存储器位置中存储数据部分，并且每一数据部分彼此之间没有联系；

第二存储器，用于在一个或多个所管理的队列中存储记录部分，其具有的固定大小存储器位置的大小等于记录部分的大小；

15 其中：

第一存储器比第二存储器大，并且具有比其小的地址带宽；并且  
第一存储器中的存储器位置被设置成具有多个不同大小的块，并且根据该数据部分的大小将存储器位置分配给该数据部分。

10. 如权利要求 9 中所述的存储集线器，其中在设置成两个所述块的第一存储器中有两个存储器位置，一个用来接收相对较小的数据部分，并且另一个用来接收相对较大的数据部分，并且其中将太大而不能存储在单个存储器块中的数据部分存储为多个块中的链接列表，其具有指向下一块的指针，但是没有任何从该包的一个数据部分指向下一数据部分的指针。

25 11. 如权利要求 9 或权利要求 10 中所述的存储集线器，其中该块中的存储器位置的大小与该通信系统中最通常出现的数据包的大小匹配。

12. 如权利要求 9 至 11 中任何一个所述的存储集线器，其中从有效地址的中央池成批地从提供给其的有效地址的池中分配在所述第一存储器中的存储器位置。

30 13. 如权利要求 12 中所述的存储集线器，其中该存储器块被分隔成多个存储器通道，并且顺序地跨过通道将地址分配给数据部分，从而将数据部分的存储分散在通道上。

14. 如权利要求 12 中所述的存储集线器，进一步包括数据检索单元，其用于以管线方式从第一存储器中读取数据部分，并指导存储器块读出数据部分，而不必等待前一个读取完成，并且从该第一存储器中释放地址位置。

5 15. 如权利要求 9 至 14 中任何一个所述的存储集线器，进一步包括标志设置装置，使得在所接收到的包没有足够的存储器的情况下，将记录部分排队，如同对应的数据部分存储在第一存储器中，顺序地读出对应于所述数据包的记录部分，并且该标志设置装置设置标志，使得丢弃所述包的数据部分，并且释放名义上分配给所丢弃数据部分的存储器位  
置。

10 16. 如权利要求 14 中所述的存储集线器，进一步包括地址位置的位图和装置，其可以工作以使得当在存储器位置存储的数据已经被读出之后释放该存储器位置时，直接将所释放的存储器位置的地址发送到该池中。

## 用于流量处理的包存储系统

### 技术领域

5 在出口流量处理机中，包以超过输出线速率的速率突发到达。例如，输入速率可以是 80Gbits/s，而线速率可以“只”有 40Gbits/s。因此需要临时包缓冲。通过在逻辑队列中缓冲包，可以通过在队列中改变有效资源（线带宽和存储容量）的分配得到不同的业务级别。

### 背景技术

10 以下几点共同使得在 40Gbits/s 的线速率时特别难以缓冲包：

1、需要高数据带宽，以同时读取和写入包（在最坏的情况下结构超速）。

2、需要高地址带宽，以在最坏的情况下拷贝，其中同时写入最小量的数据包流并在随机访问模式下从存储器读取。

15 3、存储器的容量必须高，由于缓冲器在高线速率的瞬间突发期间会迅速充满。

4、在高线速率时必须减少与逻辑队列管理或存储器管理相关联的状态操作。典型地对于执行这种功能的硬件或软件装置有效的系统时钟周期的数目最小。

20 将包直接放入映射到统计分配的存储器的队列中的方案可以满足(2)和(4)，但是使用存储器的效率较低，因此不满足(3)。将包缓冲到单片存储器或 SRAM 中的方案能够满足(2)，但是不能满足(3)，由于 SRAM 是低容量的存储器。实施使用高容量的 DRAM 的方案可以满足(3)，但是难以满足(2)，因为随机访问时间短。在试图满足(1)时，  
25 方案需要具有高带宽的互连和高引线数的接口。

总之，难以设计四个条件都满足的体系结构。

### 发明内容

本发明一方面提供一种在通信系统中对可变大小的数据包进行排队的方法，该方法包括：从所述数据包产生预定固定大小的记录部分并包含关于该包的信息，该包中的数据在数据部分中；在第一存储器的独立存储器位置中存储数据部分，并且每一数据部分彼此之间没有联系；在第二存储器的一个或多个所管理的队列中存储记录部分，其具有的固定

大小存储器位置的大小等于记录部分的大小；其中：第一存储器比第二存储器大，并且具有比其小的地址带宽；并且第一存储器中的存储器位置设置成具有多个不同大小的块，并且根据该数据部分的大小将存储器位置分配给该数据部分。

5 方便地，在设置成两个所述块的第一存储器中可以有两个大小的存储器位置，一个用来接收相对较小的数据部分，并且另一个用来接收相对较大的数据部分，并且其中将太大而不能存储在单个存储器块中的数据部分存储为多个块中的链接列表，其具有指向下一块的指针，但是没有任何从该包的一个数据部分指向下一数据部分的指针。

10 块中的存储器位置优选地与该通信系统中最通常出现的数据包的大小匹配。优选地从有效地址的中央池成批地从提供给其的有效地址池分配第一存储器中的存储器位置。

存储器块可以分隔到多个存储器通道中，并且顺序地跨过通道将地址分配给数据部分，从而将数据部分的存储分散在通道上。

15 优选地以管线方式通过数据检索单元从第一存储器中读取数据部分，其用于指导存储器块读出数据部分，而不必等待前一个读取完成，并且从该第一存储器中释放地址位置。

20 在所接收到的包没有足够的存储器的情况下，可以将记录部分入队，就像对应的数据部分存储在第一存储器中，以及顺序读出对应于所述数据包的记录部分，设置标志用来表示将要丢弃所述包的数据部分，丢弃所述数据部分，并且释放名义上分配给所丢弃数据部分的存储器位置。

25 可以从地址的位图中读取地址位置，并且当在存储器在其中的数据已经被读出之后释放存储器位置时，直接将所释放的存储器位置的地址发送到该池中。在另一方面，本发明提供一种存储集线器，其包含执行如下面的权利要求中所定义的方法所必须的所有特征。

30 本发明在其最广泛的意义上，提供一种存储集线器，其用来在高线速率流量处理机中缓冲包，从而实现使用高速串行链路与存储集线器连接的大量可独立寻址的存储器通道，从而能够应用于网络线卡上的流量处理机。流量处理机的特征在于高线速率和对大存储器的高速随机访问。

更广泛地，当需要对非常大存储器进行高速随机访问（等待容限）

时，可以应用该方法。

根据优选实施例如下实施本发明：

队列管理器和存储器管理器是完全去耦合的（通过包/包记录概念表明。这针对于标准（4））。

5 包直接流入动态分配的存储器中，几乎没有第一手状态操作（通过存储器管理器和使用本地池得到支持。其针对于标准4）。

存储器地址空间分段和使用存储器管理器的位图进行的有效性跟踪对动态存储器管理提供必要的支持（通过高效存储器使用有助于满足（3））。

10 使用高速串行芯片至芯片的链路与存储集线器连接（远程输出到多个存储器通道使得地址和数据带宽能够成比率，以满足要求，而不会满足实施限制，这针对于（1）和（2））。

#### 附图说明

现在将参照附图描述本发明，其中：

15 图1是该包存储系统的主要组件的功能简图；

图2是概存储集线器的体系结构简图；

图3是数据报检索单元设计；

图4(a)是对于多芯片、规模可变实施方法，用于流量处理的包存储系统的实施；

20 图4(b)是对于单芯片、高度集成的技术方案，用于流量处理的包存储系统的实施。

#### 具体实施方式

本发明包括组装成为一个方案的组件特性、思想和装置，该方案满足流量处理系统中用于40Gbits/s包缓冲所需要的全部条件。存储集线器是本发明的主要实施例。将根据该主要实施例，并连同实现本发明能够传送的性能级别所需要的和/或所想要的周边设备一起描述本发明。为此，下面的说明书被分为描述本发明不同方面和/或特征的多个副标题。不可避免地，在下面的部分之间存在一定的重叠。

30 存储系统去耦合——该特征的主要目的是分离出包存储问题，使得其不与其它功能混杂在一起，并与其它功能相互不相关。相对复杂的功能可以用来控制包的入队和出队。如果包本身并不传送到该系统，这种包处理/操作的复杂度可以减少。由于在流量处理中并不需要访问包内

容，可以将该包放在存储器中，并通过少量的固定大小的包记录表示。在逻辑队列或数据结构中被处理和计算的是这些记录。按照顺序调度的包记录接着可以用来覆盖包，以在输出线上转发。于是，从包缓冲和存储器管理的任务中分离出处理和逻辑队列管理功能。接下来，对少量的固定大小的包记录元数据进行 QoS 处理。该记录典型地包括包在存储器中的位置（第一块的地址）、该包所属的流的标识（附在上游包上）、包长度和控制标志。使用该流标识符在本地查找附加数据。因此参照图 1，在到达块 1 处到达的数据包流的长度可变。从所接收到的每一包中产生包记录。每一包记录包含关于其各自数据包的信息，并因此“表示”该数据包。而且，该包记录的大小固定，使得原始包内的数据占据可变大小的数据部分。

这是本发明的关键特征，因为其将记录部分和数据部分单独地并彼此独立地进行处理，从而使得能够优化记录和数据的存储。将包的可变长度的数据部分发送到存储集线器 2 中，而记录部分发送到记录处理系统 3。该包记录部分比该数据部分短，并因此相比于处理作为单个整体的全部接收的数据包，可以在单独的处理系统 3 中有效地对其进行处理。

该存储集线器 2 根据该包在其到达存储器 4 的路上从本地池 6 中所拾取的地址，正确地将该数据传送到存储器 4 中。如此后所解释，本地池保持有一批从存储集线器 2 向其发送的有效地址。该系统工作完全不需要该池“请求”用于特定数据包的地址。中央池 7 保持有包存储器 4 中的所有存储器位置的地址。通过这种方式，数据包简单地流入存储器 4，并且具有尽可能少的处理，以通过高数据速率，例如以当前高至 80Gbits/s 的速率从输入通道拷贝。

包记录包含指向存储器中的包数据的指针。也可以有标志，即告诉系统 3 该如何处理记录的信息，诸如应该将其放入哪一队列，以及它们具有何种优先级。每一包具有队列标识符。

大多数包存储/排队系统将整个包作为单个整体处理，包括记录和数据信息，并因此与介绍部分中所述的标准相冲突。本发明通过满足所有这些标准的唯一方式对该包的记录和数据部分进行处理。

存储集线器的使用可以确保当包高速达到时，对它们进行尽可能少的处理，而可以以较低的速度“离线”地处理包记录。数据于是可以直接流入存储器。常规方法使用的技术包括标识该数据将被写入该队列中

的何处。如果已经写入了，必须通过重写指针来更新该状态。当新数据达到时，读取指针并且必须再次更新该指针。该过程显然是密集的，并且降低了整体处理的速度。在本发明的优选实施方式中，没有“握手处理”。

5 极其高速的输入数据需要物理存储器4具有许多通道。这就产生了“引线问题”，其中队列太深使得存储器必须很大（例如数百兆字节），并因此不是单片级。该集线器因此提供用于分离物理和逻辑存储器问题的装置。其然后可以是独立芯片。该存储集线器提供有第二装置，可以在其上面实施所有的存储器通道，而不必支持不同的接口。该存储器4因此优选地是一组存储器芯片，并且该存储集线器潜在的是一个芯片。  
10

该存储集线器具有关于存储器的每一块是自由的知识，并且将一组地址从中央池7传送到在到达块1中的本地池6，指示具有用于接收其包记录已经被到达块剥离的数据的有效地址。在到达块处所到达的数据的每一包都分发有从本地池6中得到的地址，并将其发送到用于存储的对应存储器位置。  
15

存储器管理——当存储器用作数据结构的存储时，其通常是静态或者动态分配的。用于将存储器分配给包的一种有效方法包括下面的特征，其在本发明的优选实施例中提供。包存储器4在该存储器地址空间中被划分成小块。这些块可以是n个不同配置的大小之一。为了减少系统的复杂性，适当地考虑n=2。没有将存储器静态地分配给队列。相反，包存储在一个或多个给定大小的块中（适当地）。给定包的每一块通过链接列表的方式指向下一个，但是在存储器4中没有逻辑队列管理。通过存储集线器中的存储器管理器记录所有存储器块的有效性。在图2所示的存储集线器的结构图中，该存储器管理器方框所示为20. 输入侧（图2中的左侧）保持简单，从而处理80Gbits/s的高数据速率，并且将该复杂度故意地保持到输出侧（图2中的右侧）。该输入侧有效地包括分布到不同存储器通道的连续数据流。通过写控制器22产生的写操作具有的优先级比通过读控制器23产生的读操作高。如前面参照图1所解释，写流入，并根据在到达块1中的本地池6所分配的地址，跨过多个存储器通道均衡地在请求复用器24和存储器通道接口25上分布。  
20  
25  
30

该存储器管理器20使用存储器模块21中每个有效存储器块的位图，每一位表示一个块。当该存储器管理器从中央池7发送地址块到本

地池 6 时，该地址与每一个模块相关。因此，当输入数据从本地池 6 拾取地址时，每一包或包的段会被发送到不同的模块，从而实现负荷分散。如果模块中的每一存储器块具有 64 字节的容量，也就是说整个包可以全部存储在一个块中，假定包大小少于 64 字节。更大的包，例如 256 字节的包会跨过几个块作为链接列表存储。由于数据作为链接列表存储在存储器中，存储器中的每一块指向下一个，从而如果必须在读取下一项之前读取第一项数据和所提取的指针，其在检索时低效率。因此，有利地将存储器划分成为不同大小的块，以分别接受大的和小的数据包。通过该方式，大包可以存储在大块中，使得数据报检索单元可以读取第一个大块，并且如果指针返回到第一个，其可以在完成第一个读取之前，通过管线的方式发布请求读取下一个。

这样有助于在写和读控制器 21、22 之间的潜在争夺，并且后来也有助于当派送单元 5 向存储集线器发送请求需要特定的包时，该请求首先到达数据报检索单元 DRU，诸如图 2 中的 24，其然后将该请求轮流发布到每一通道。因此，在读取侧和写入侧都有分布。这样进一步增强了负荷分散。

如果考虑平均来自切换结构的输入是 80Gbits/s，并且输出侧是 40Gbits/s，可以出现数据被转发到同一输出的情况。因此，理想地在输出应该有缓冲器用来处理负荷突发。尽管存储器的大小有限，但是它们还是可以只处理超过 40Gbits/s 的第二个流量的片段。

通过从该位图读取字，该存储器可以成批地标识空闲块的地址。将位转换成为地址，并且将该地址保持在有限的但是足够大小的中央池中。这是一种数据解压缩形式，其存储效率比维持存储器“自由列表”高（在队列或链接列表中存储所有可能的地址）。通过扫描该位图、或者更直接地根据从存储器和存储器块中读取作为包达到的地址流，并且释放其占用，可以将该中央池 7 加满。如果中央池为满，必须缓冲所返回的地址，并且将它们的信息插入到该位图中。

高效包存储——将信息放在存储器中一般需要记录存在新信息的更新状态的系统开销。需要一种平滑地以 80Gbits/s 将数据流入存储器中的装置，其并不需要通过中间状态操作的帮助。到达块 1 是管线处理器，其从用来创建包记录的该包中提取信息，并且将该包切割成可以映射到存储器块中的切段。根据每一包的长度，适当地选择用于其的段大小（并

于是就是存储器块大小）。该包被转发到存储集线器，并且该包记录被转发到用于 QoS 处理和逻辑排队的系统。所实施的每一不同存储器块大小要求单独的本地（和中央）池。该本地池使得到达能够立即将（分段的）包载入存储器的自由块中。这样做唯一不重要的复杂性就是要将同一包的最后一个存储器块的地址插入到当前块中。其是一个简单快速的系统，不同于从本地池中弹出项，其不需要状态操作。该系统的一个重要特征就是，其支持在存储集线器的有效存储通道上的负荷平衡。当补充本地池时，其等量地接收映射到不同物理存储器通道中的存储器块的地址。接续块于是可以通过循环方式写到存储器块中，有效地分散了地址和数据带宽负荷。在本地池变为空的偶尔情况下，需要由到达丢弃部分或全部的包。考虑记录已经被部分地载入存储器，但是该池临时变空，例如因为存储管理器中的小故障，或者因为其不能快速返回以接收新的批，该队列可能完全装满，并因此导致存储器溢出。在该情况下，不是修正问题，而是丢弃在到达块处到达的其余包，并且还是通过处理器 3 发送该包记录，但是将该记录标记为“垃圾”。

因此还是要通过包记录创建来报告执行情况，使得可以通过 QoS 处理器集中地管理事件处理。该处理器必须从存储器中清除该包，并且必须报告所丢弃的任何包的详情。这些都是 QoS 处理器用于普通操作所已经具有的功能。

本发明优选实施例的显著特征是在派送单元 5 出现丢弃包的实际处理。记录向平常一样到达派送块 5 中的队列的报头，但是因为标志，其可以看作垃圾记录。数据还是要读取，但是然后就丢弃，从而释放分配给该数据项的存储器，而不需要将该垃圾数据发送到线上。

高效的包恢复——由于其 80Gbits/s 的性能要求有挑战性，所以故意地简化该包存储功能。结果，通过到达使用的“开环”法使得 40G 的包恢复功能变得更复杂。要点是，如果包存储在存储器中的多个块中，每一个指向下一个，一个块必须在可以发布下一个块的请求之前读取，并提取“下一个指针”。将包转发到线上的该装置是派送块。其可以看作是一种类型的 DMA 引擎，其从 QoS 处理系统中获取地址（包记录），使用该地址从存储器（存储集线器）中获取数据（包），并将该数据转发到线上。存储器标称地被组织成为 n ( $n = 2$ ) 组块——大块和小块。选择块的大小，以 (a) 通过将包大小分布中的峰值与块大小匹配来优化存

储器的使用，和 (b) 检索作为链接列表存储的包更高效。于是，如果说所有的包具有 64b/s 和 256b/s 的峰值，可以选择该块大小来匹配。然后将非常大量的包装入单个块中，并且不需要分布在超过一个的块之间。  
5 可以设置到达块，以进行关于两个大小块中的哪一个的值调整，其地址保持在两个本地池中，而不是图 1 中所示的单个池 6 中，以分配所讨论的包。当包被存储在多个存储器块中时，在所发布的请求与返回的第一数据（其具有下一个块的地址）之间存在延迟。如果块足够大，那么可以从第一个少数字节的块中提取下一个指针，并且发布下一个请求，同时10 还从存储器中读取其余块。通过选择存储器块大小，使得大多数包可以存储在单个存储器块中，就可以有效地聚合包恢复。在图 2 的完全管线模式中（即：在前一个请求的响应返回之前可以发送包请求）的存储集线器中，可以通过数据报检索单元 (DRU) 26 提取包。如果每一包在单个块中，可以管线传输该请求。然后所需要的就是通过图 3 中的重排序缓冲器进行任何所需要的重新排序，如下将详细解释。然而，如果包存15 储在链接列表中，在可以请求下一个包之前必须检索每个包的数据，从而降低了检索过程的速度。图 3 表示数据报检索单元 (DRU)。记录 30 到达控制器 31，其从记录 32 中提取指针并发布请求 33，但是其知道该记录是否是链接列表中的一个串。如果其不需要等待回来的响应，其可以发布下一个请求。其可以按照这种方式继续。另一方面，如果其需要20 等待，其停止并建立环路。包的起始回来，并且提取指针。如果块非常大，如几百字节长，实际上保持该指针，从而在读取控制器中读取后几个字节之前，块的前几个字节已经回来了。因此可以提取该指针，并循环回到该控制器的数据报指针 FIFO34 中，并且发布下一个包的请求。

一旦链接列表中的包已经返回，它们可以在重排序缓冲器 36 的存储25 单元阵列 35 中重新排序。该存储单元阵列等同于一系列鸽笼 (pigeon hole)，将数据按照其所读出的顺序存储在其中。在等待数据填充到下一个空鸽笼的序列的末端有一个指针。只有接收到该数据，其可以向前移动，并且在其碰到缺口的时候，缺口表示该数据还没有返回。

尽管缓冲器以固定的速度发送数据，在速率控制反馈输入 38 的命令30 下对其进行速率控制 37。在指针返回的缓冲器 39 中，提取已经读取的块的地址，使得可以释放该块，并且将指针输出馈送到存储器管理器。

于是，通过实施在多个不同的级别具有数据恢复的分层包检索系

统，可以有效地进行多块包的恢复。派送请求完成来自数据报检索单元的包。该 DRU 从各个存储器通道的存储器读取控制器中提取存储器块，并重新组装包。该存储器读取控制器通过突发读取访问从存储器中提取字，通过标识该块内的有效数据的起始和末端对该块内容进行重新组 5 装。对于该 DRU 读取的每一存储器块，将该块地址传送到存储器管理器，使得可以更新相关的存储器位图（或者重新使用的块地址）。现在返回到图 2，数据报丢弃单元 27 以类似于 DRU 的模式工作，但是不返回数据。其目的是更新存储器管理器中的存储器块的状态位。可以丢弃包，因为作为它们已经被标记为垃圾的结果就已经被截去了，因为数据输入速率 10 超过了存储集线器和存储器模块的存储容量，或者因为该流量处理机已经实施算法来随机地丢弃包，以管理队列。虽然与存储集线器本身没有关系，但是负责实施该实际丢弃功能的还是该数据报丢弃单元。然而在丢弃之前要从存储器中读取该包，只是要提取指针。这可以为存储在单个块中的包直接完成。存储在多个块中的包必须从存储器中读取，使得 15 可以恢复“下一个块”的指针。

现在考虑存储器管理器，当存储器用于存储时，其通常静态或动态地分配。然而在本发明中，在静态和动态存储器之间采取“中间结构”。如果一部分存储器静态地分配给一个队列，而另一部分分配给另一个队列，这样的效率非常低，由于存在所有的流量都进入一个队列的风险。如果动态地分配存储器，系统在每一个存储器为满的时候就不能够接收数据。相反，该存储器被分段成为较小的、位图块。 20

因此为这些块保持位图。其在该存储器管理器中，优选地存储在 SRAM 中，作为该存储器中每一个块的记录。其包括关于该块的状态的信息，诸如其是否被使用。可以通过从中央池 7 中读取来刷新该本地池 6。如果地址字是 32 位宽，其将从零至 32 个该有效存储器的块中返回标识。其以高效的方式提供任何有效的空闲存储器。这可以通过使用自由列表的方式来代替，其带有指向每个位置的指针的链接列表。该方法虽然十分彻底，但是在读取指向第一个的指针时效率低，其表示该特定块是自由的，并然后指向下一个自由块。通过读取该位图，可以以高的速率读取指针。在单个时钟周期中，可以一次读取 32 个自由块的标识符，并将它们转换成完整的地址。 25

在最差的情况下，即从存储器的一端工作到另一端，一次读取用于

32 个块的标志，其可能是这样的，在两个或三个周期上遇到密集占据的存储器部分，并且没有明显的标志。为了对付可能由于这种不测事件所引发的延迟，该存储器管理器包括所谓的“捷径”模式。当地址被 DRU 释放并返回到存储器管理器时，它们并非必须返回到位图中。相反，可以将它们缓冲并循环使用。当接收到刷新本地池的请求时，该中央池并非将它们从位图中除去。如果其 FIFO 已经缓冲有一批返回用于重发布的地址，其可以替代地释放它们。

该存储器管理器因此包括用于位图的 SRAM。当例如 12 位宽的地址从 DRU 到达时，其表示存储器的某个地方刚刚被释放。其劳动强度较大，必须将该地址转换成为位，从 SRAM 中读出该地址，插入该位并然后将其写回。该读取和写操作包括 32 位字。专用 FIFO 保持该 SRAM。已经解释了，本地池需要周期性的刷新。因此替代从位图中读出 32 位字，将它们转换成为指针，并然后将这些指针发送回到该池中，相反优先地使从 DRU 中发送回来的这些地址循环。

实施不同的系统——系统的划分必须考虑现实情况，诸如装置引脚数、功率消耗、硅工艺技术、PCB 总数和布线等。通过存储集线器访问多个可独立寻址的存储器通道。该集线器将存储器类型和该存储器所需要的制作技术与系统的其它部分隔开。该集线器也可以使得实施大量可变数目的存储器通道，由于 (a) 集线器封装将用于存储器通道实施的引线数目最大化；和 (b) 多个集线器可以与单个 QoS 处理芯片连接。参照图 4，该存储集线器 40 通过窄的高速串行芯片至芯片的链路 41、42 与处理芯片 (QoS 和队列芯片) 连接。于是减少了与处理芯片的引线数目接口连接的存储器的负担，减少了整体的引线数目并减少了潜在的复杂 ASIC 的封装成本。

#### 25 优先实施例的概括和进一步详情

如图 1 中所示，到达和派送提供了该包流中的叉点和结合点。它们将存储集线器分开，其只简单地分发从与存储器连接的高速链路接收到的包组块，或者按照要求从存储器检索包。

如图 2 中所示，该总线用作 DRU 与存储器通道控制器之间的纵横连接器。这表示包可以存储在多个存储器通道上的存储器块中，并可以通过单个 DRU 被提取/重构。存在有多个 DRU，以增加读取带宽，并为单线输出 (OC-768) 系统提供负荷均衡。可替换地，在多线系统（例如 4 ×

10G 以太网) 中, 每线分配单个 DRU.

该存储器管理器包含中央池和存储器位图。该存储器位图在 SRAM 中实施, 并且典型地为 256k 字节, 以支持组织为 512 字节存储器块的 1G 包存储器。

5 如图 3 中所示, 该控制器监管读取管线处理或从链接列表中的包恢复处理。

10 图 4 所示为两个可能的系统实施方式。存储器的 8 个通道提供大约 20GBytes/s 的数据带宽和 300M 每秒的随机访问。这满足了 40G 流量处理 ( $2 \times$  切换结构超高速) 的要求。虽然可以看到, 该整个系统可以在单个装置中实施, 如图 4 (b) 中所示, 但是更实际地是在多个装置上分布该功能, 如图 4 (a) 中所示。存储集线器的使用使得避免了在一个集线器上具有所有的存储器。例如, 两个单芯片的范例允许每一芯片“拥有”半个通道。该到达块然后可以集线器之间分布, 并且该派送块可以恢复来自该集线器的数据。可能需要平衡集线器之间的流量。

15 使用多个装置有许多很好的理由, 包括:

如果所有装置并不围绕单个高集成的处理器集群, 就可以减轻物理分布和大量存储器装置的二级互连;

20 该存储器技术与该主要的专有处理逻辑分开。这样能更新芯片, 同时还可以维持该同一存储器系统, 并且该存储器系统可以缩放, 而不会过渡地增加引线的数目。现在, 数据存储与该处理系统存在物理分离和逻辑分离;

如果存储器通道紧密地围绕单个芯片的周边集群, 难以观察到电子特性、信号线分离和线终端上的紧规范;

25 功率耗散更均匀地分散。单个装置中高功率和引线数目的组合可能需要昂贵的封装技术; 在所使用的多个集线器可以被缩放以满足该应用的存储器要求 (例如从 10G 放大到 40G, 以及更多) 时, 也可以适用该多芯片方法。

30 该存储集线器可以设计为芯片内的逻辑块, 如图 4B 中所示。然而, 该存储集线器的一个优点在于, 存储在存储器中的队列与对这些队列所进行的处理逻辑之间没有密切的关系。图 4A 中所示的体系结构具有将该数据存储功能的存储器与处理侧分离的优点, 使得其非常容易地产生接口。在图 4A 中, 数据流入和流出该流集线器 43。在该集线器与存储集线

器之间是用于指针的请求、返回的指针流 41 和送入的指针流以及返回的数据 42。不需要复杂的握手处理。

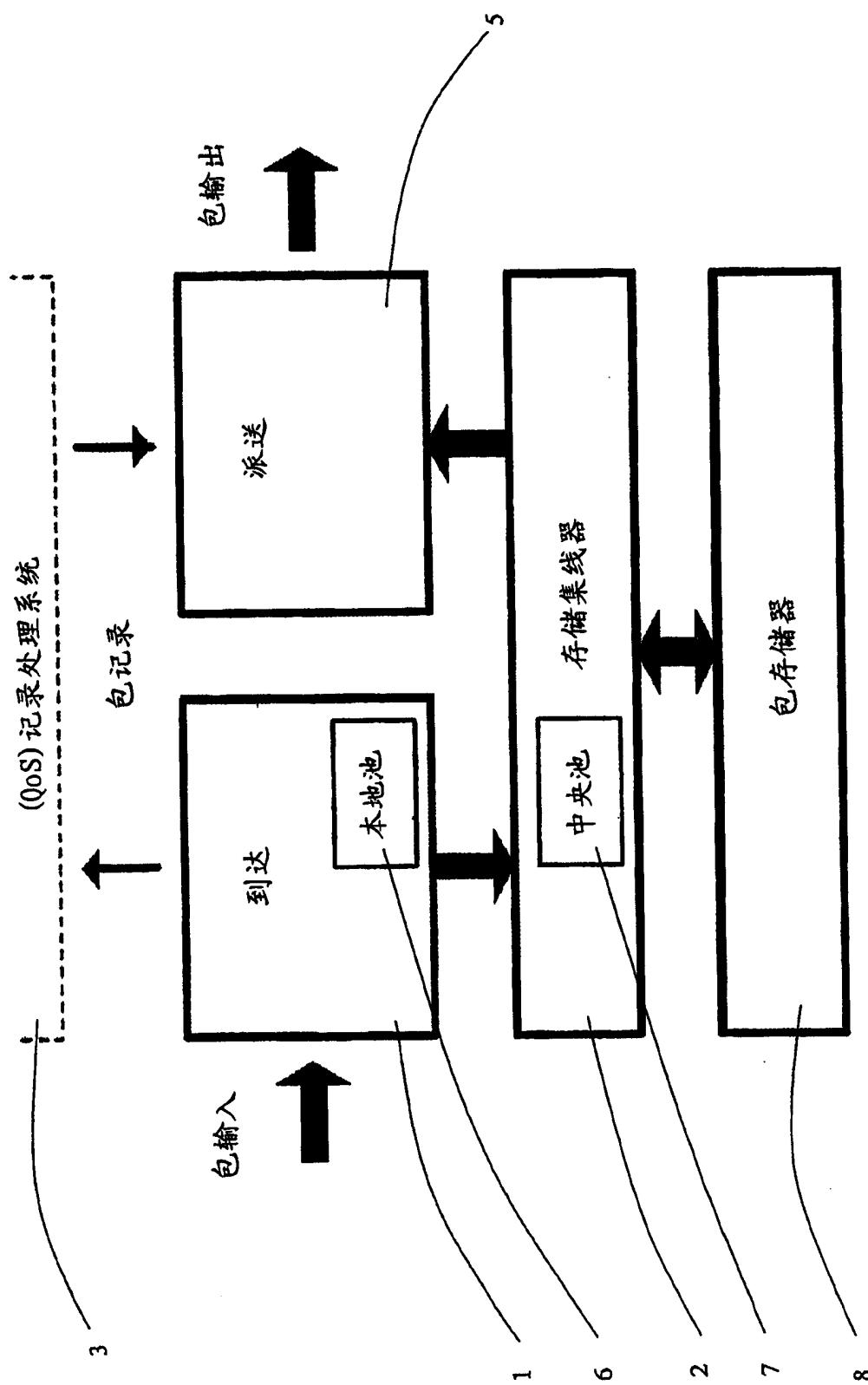


图 1

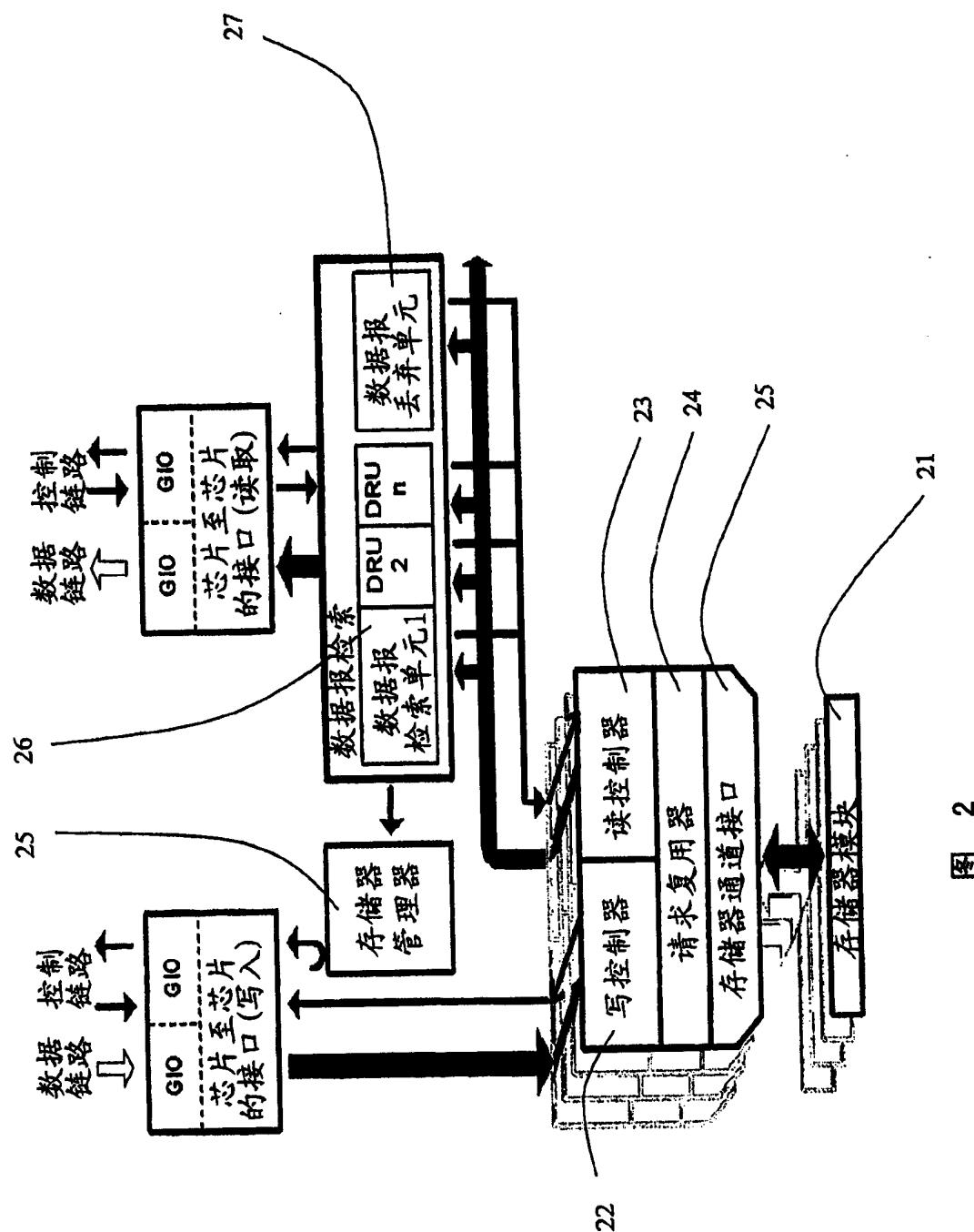


图 2

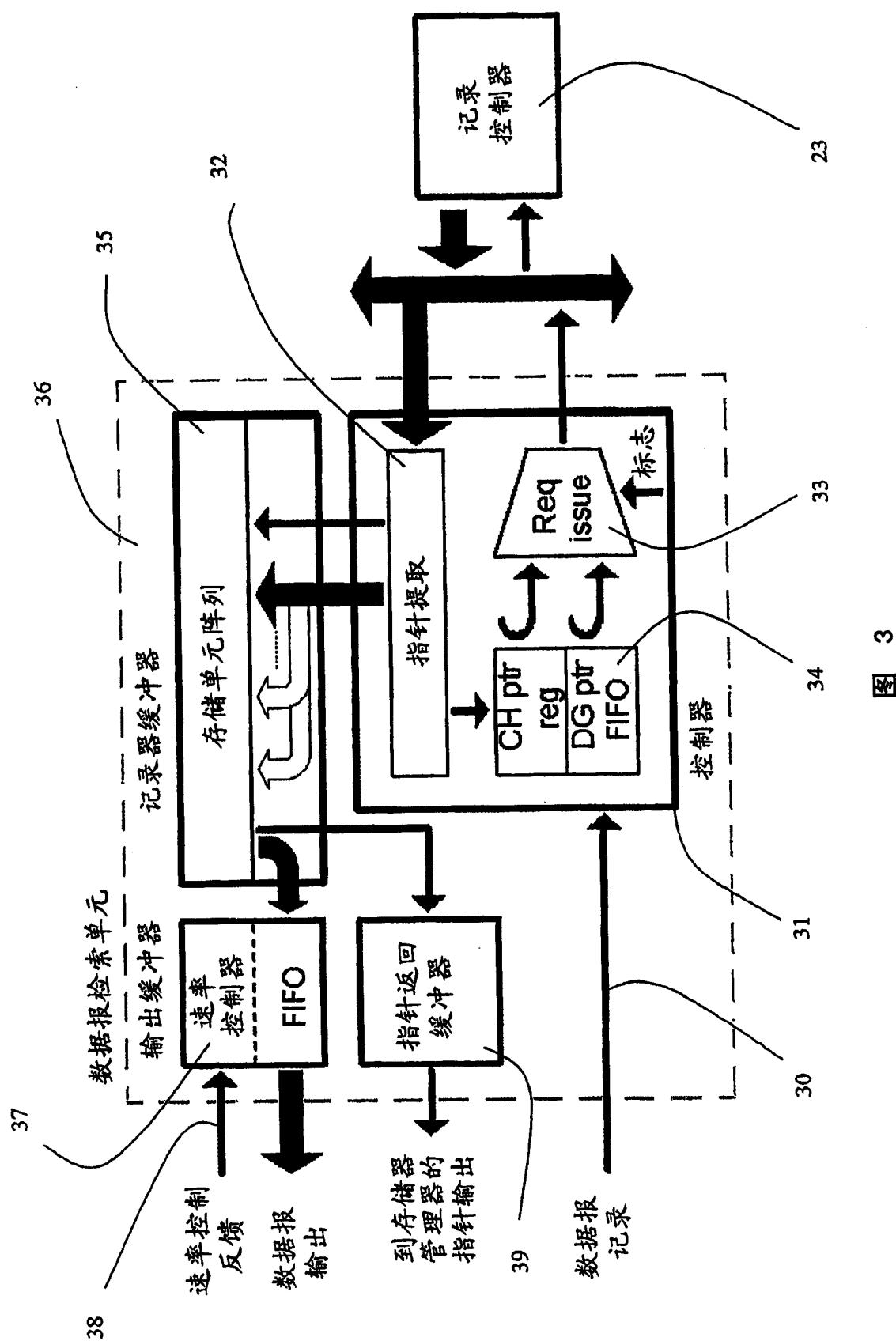


图 3

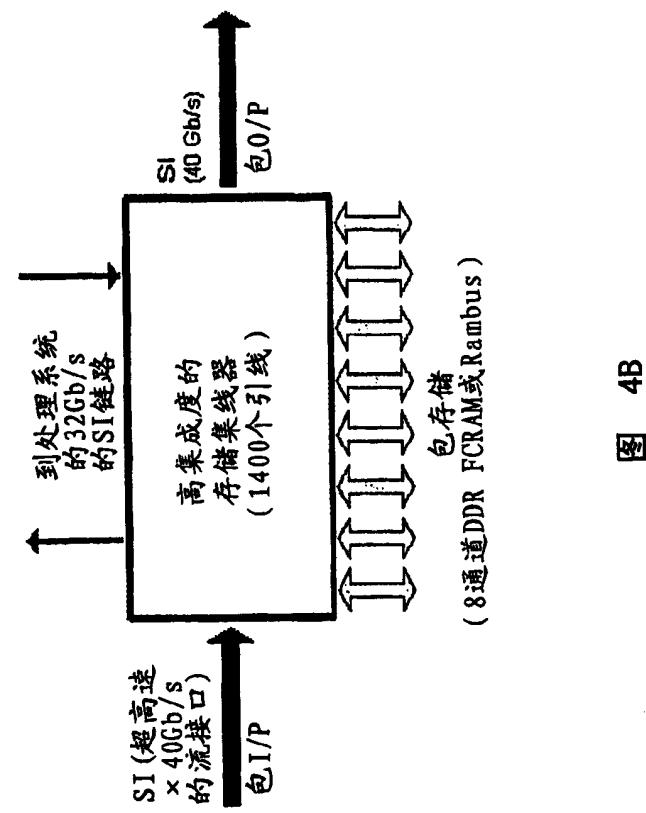


图 4B

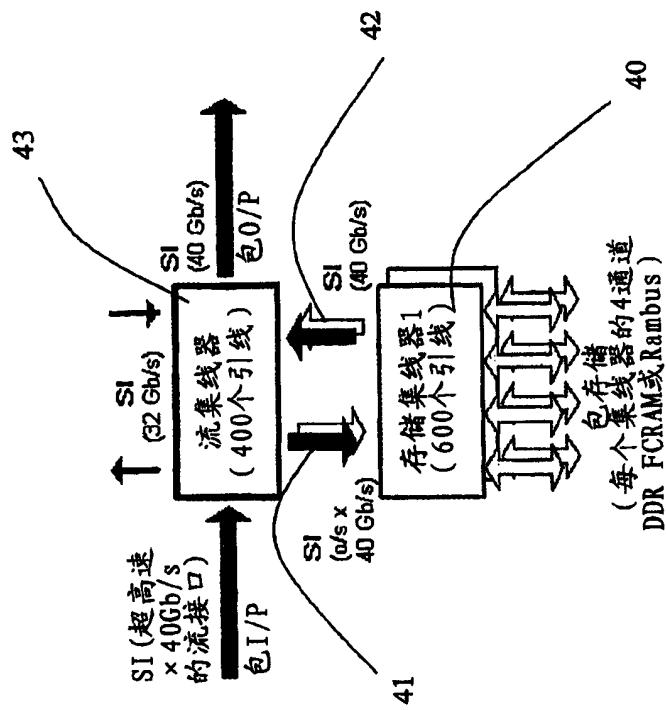


图 4A