



[12] 发明专利申请公开说明书

[21] 申请号 200480015912.0

[43] 公开日 2006年7月12日

[11] 公开号 CN 1802836A

[22] 申请日 2004.5.26
 [21] 申请号 200480015912.0
 [30] 优先权
 [32] 2003.6.11 [33] US [31] 10/460,290
 [86] 国际申请 PCT/US2004/016510 2004.5.26
 [87] 国际公布 WO2004/112350 英 2004.12.23
 [85] 进入国家阶段日期 2005.12.8
 [71] 申请人 英特尔公司
 地址 美国加利福尼亚州
 [72] 发明人 H·贝弗利 A·舒巴尔

[74] 专利代理机构 中国专利代理(香港)有限公司
 代理人 杨凯 刘杰

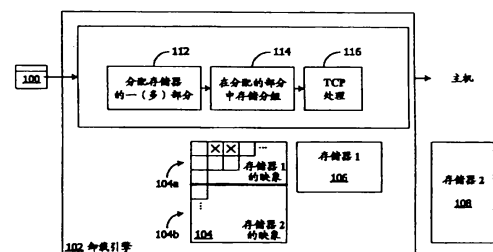
权利要求书6页 说明书9页 附图7页

[54] 发明名称

网络协议卸载引擎存储器管理

[57] 摘要

一般而言，在一个方面，公开内容描述了一种处理分组的方法。方法包括：访问网络协议卸载引擎上的分组；至少部分基于存储器映象，分配至少第一存储器和第二存储器的一部分或多部分存储空间。存储器映象通常映射和标识第一存储器和第二存储器部分的占用。方法还包括在所分配的一个或多个部分中存储分组的至少一部分。



1. 一种处理分组的方法，所述方法包括：
访问网络协议卸载引擎上的分组；
- 5 至少部分基于存储器映象，分配至少第一存储器和第二存储器的一部分或多部分存储空间，所述存储器映象通常映射所述第一存储器和所述第二存储器，所述存储器映象标识所述第一和第二存储器部分的占用；以及
在所分配的一个或多个部分中存储所述分组的至少一部分。
- 10 2. 如权利要求 1 所述的方法，其特征在于，所述存储器映象包括分成多个部分的映象，不同部分映射不同存储器提供的存储空间。
3. 如权利要求 1 所述的方法，其特征在于，所述存储器映象内的单元包括标识所述第一和第二存储器中哪个存储器与所述单元相关联的数据。
- 15 4. 如权利要求 1 所述的方法，其特征在于，所述网络通信协议卸载引擎包括传输控制协议（TCP）卸载引擎。
5. 如权利要求 1 所述的方法，其特征在于，所述存储器映象不是地址空间中连续地址的线性映射。
6. 如权利要求 1 所述的方法，其特征在于，所述第一存储器和
20 所述第二存储器包括提供不同等待时间的存储器。
7. 如权利要求 1 所述的方法，
其特征在于，所述第一存储器包括位于第一芯片上的存储器；
其特征在于，所述第二存储器包括位于第二芯片上的存储器；
以及
- 25 其特征在于，所述网络通信协议卸载引擎包括位于所述第一芯片上的逻辑。
8. 如权利要求 1 所述的方法，其特征在于，所述分配包括根据所述分组的内容分配。

9. 如权利要求 1 所述的方法，

其特征在于，所述存储包括在所述第一存储器中存储；以及还包括：

5 确定将所述分组的至少一部分从所述第一存储器移到所述第二存储器；以及

使所述分组的所述至少一部分从所述第一存储器移到所述第二存储器。

10. 如权利要求 1 所述的方法，其特征在于，所述存储器映象包括位映象，所述位映象内的各个比特标识存储器对应部分的占用。

10 11. 如权利要求 1 所述的方法，其特征在于，所述分配包括分配相连的存储单元。

12. 如权利要求 1 所述的方法，还包括将所述分组经直接存储器存取（DMA）传送到主机可访问的存储器。

15 13. 如权利要求 1 所述的方法，其特征在于，所述网络协议卸载引擎包括以下之一：网络接口卡内的组件和主机处理器芯片集内的组件。

14. 如权利要求 1 所述的方法，其特征在于，所述网络协议卸载引擎包括至少以下之一：专用集成电路（ASIC）、门阵列和网络处理器。

20 15. 一种部署在计算机可读介质上的计算机程序，所述程序包括使网络协议卸载引擎处理器执行以下操作的指令：

访问由所述网络协议卸载引擎接收的分组数据；

25 至少部分基于存储器映象，分配至少第一存储器和第二存储器的一部分或多部分存储空间，所述存储器映象通常映射所述第一存储器和所述第二存储器，所述存储器映象标识所述第一和第二存储器部分的占用；以及

在所分配的一个或多个部分中存储所述分组的至少一部分。

16. 如权利要求 15 所述的程序，其特征在于，所述存储器映象

包括分成多个部分的映象，不同部分映射不同存储器提供的存储空间。

5 17. 如权利要求 15 所述的程序，其特征在于，所述存储器映象内的单元包括标识所述第一和第二存储器中哪个存储器与所述单元相关联的数据。

18. 如权利要求 15 所述的程序，其特征在于，所述网络通信协议卸载引擎包括传输控制协议（TCP）卸载引擎。

19. 如权利要求 15 所述的程序，其特征在于，所述存储器映象不是地址空间中连续地址的线性映射。

10 20. 如权利要求 15 所述的程序，其特征在于，所述第一存储器和所述第二存储器包括提供不同等待时间的存储器。

21. 如权利要求 15 所述的程序，其特征在于，使所述处理器进行分配的所述指令包括使所述处理器基于所述分组的内容进行分配的指令。

15 22. 如权利要求 15 所述的程序，还包括使所述处理器执行以下操作的指令：

确定将分组的至少一部分从所述第一存储器移到所述第二存储器；以及

20 使所述分组的所述至少一部分从所述第一存储器移到所述第二存储器。

23. 如权利要求 15 所述的程序，其特征在于，所述存储器映象包括位映象，所述位映象内的各个比特标识存储器对应部分的占用。

24. 如权利要求 15 所述的程序，其特征在于，使所述处理器进行分配的所述指令包括使所述处理器分配相连存储单元的指令。

25 25. 一种网络接口卡，所述卡包括：

至少一个物理层（PHY）装置；

耦合到所述至少一个物理层装置的至少一个介质访问控制器（MAC）；

至少一个网络协议卸载引擎，所述引擎包括执行以下操作的逻辑：

访问分组；

5 至少部分基于存储器映象，分配至少第一存储器和第二存储器的一部分或多部分存储空间，所述存储器映象通常映射所述第一存储器和所述第二存储器，所述存储器映象标识所述第一和第二存储器部分的占用；以及

在所分配的一个或多个部分中存储所述分组的至少一部分；以及

10 到总线的至少一个接口。

26. 如权利要求 25 所述的卡，其特征在于，所述至少一个接口包括外围组件互连（PCI）接口。

27. 如权利要求 25 所述的卡，其特征在于，所述网络协议卸载引擎逻辑包括至少以下之一：专用集成电路（ASIC）和网络处理器。

15 28. 如权利要求 27 所述的卡，其特征在于，所述逻辑包括网络处理器，所述网络处理器包括多个精简指令集计算（RISC）处理器。

29. 如权利要求 25 所述的卡，网络通信协议卸载引擎包括传输控制协议（TCP）卸载引擎。

20 30. 如权利要求 25 所述的卡，其特征在于，所述存储器映象不是地址空间中连续地址的线性映射。

31. 如权利要求 25 所述的卡，其特征在于，所述第一存储器和所述第二存储器包括提供不同等待时间的存储器。

32. 如权利要求 25 所述的卡，

其特征在于，所述第一存储器包括位于第一芯片上的存储器；

25 其特征在于，所述第二存储器包括位于第二芯片上的存储器；

以及

其特征在于，所述网络通信协议卸载引擎包括位于所述第一芯片上的逻辑。

33. 如权利要求 25 所述的卡，其特征在于，所述进行分配的逻辑包括根据所述分组的内容进行分配的逻辑。

34. 如权利要求 25 所述的卡，

5 其特征在于，所述网络协议卸载引擎逻辑还包括用于执行以下操作的逻辑：

确定将所述分组的至少一部分从所述第一存储器移到所述第二存储器；以及

使所述分组的所述至少一部分从所述第一存储器移到所述第二存储器。

10 35. 如权利要求 25 所述的卡，其特征在于，所述存储器映象包括位映象，所述位映象内的各个比特标识存储器对应部分的占用。

36. 如权利要求 25 所述的卡，其特征在于，所述存储器映象包括分成多个部分的映象，不同部分映射不同存储器提供的存储空间。

15 37. 如权利要求 25 所述的卡，其特征在于，所述存储器映象内的单元包括标识所述第一和第二存储器中哪个存储器与所述单元相关联的数据。

38. 一种系统，它包括：

至少一个主机处理器；

至少一个物理层（PHY）装置；

20 耦合到所述至少一个物理层装置的至少一个以太网介质访问控制器（MAC）；

至少一个传输控制协议（TCP）网络协议卸载引擎，所述引擎包括用于执行以下操作的逻辑：

25 访问经所述至少一个 PHY 和所述至少一个 MAC 接收的分组；

至少部分基于存储器映象，分配至少第一存储器和第二存储器的一部分或多部分存储空间，所述存储器映象通常映射所述第一存储器和所述第二存储器，所述存储器映象标识所述第

一和第二存储器部分的占用；以及

在所分配的一个或多个部分中存储所述分组的至少一部分。

39. 如权利要求 38 所述的系统，其特征在于，所述 PHY 包括无
5 线 PHY。

40. 如权利要求 38 所述的系统，其特征在于，所述卸载引擎包
括至少以下之一的组件：网络接口卡和主机处理器芯片集。

网络协议卸载引擎存储器管理

5 背景

网络使计算机和其它装置可进行通信。例如，网络可承载表示视频、音频、电子邮件等的的数据。通常通过网络发送的数据分成称为分组的更小消息。用类推的方法，分组很像您投入邮箱的信封。分组一般包括“有效负荷”和“报头”。分组的“有效负荷”类似于信封内的信。分组的“报头”很像在信封本身上所写的信息。报头可包括帮助网络装置适当处理分组的信息。

多个网络协议协作以处理复杂的网络通信。例如，称为传输控制协议（TCP）的协议提供允许远程应用程序进行通信的“连接”服务。换言之，很像拿起电话并且在假定电话公司将进行中间的所有工作的情况下，TCP 为应用程序提供简单的基元，以建立连接（例如，“连接”CONNECT 和“关闭”CLOSE）并传送数据（例如，“发送”SEND 和“接收”RECEIVE）。在后台，TCP 透明地处理多种通信问题，如数据重新传输、适应网络业务拥塞等等。

为提供这些服务，TCP 对称为段的分组进行操作。通常，TCP 段在诸如因特网协议（IP）数据报等更大的分组内（由其“封装”）遍历网络。段的有效负荷承载网络上发送的部分数据流。接收器通过收集接收的段，可恢复原始数据流。

段可能不会以其正确的顺序到达其目的地，即是有也极少。例如，不同的段可通过极其不同的路径遍历网络。因此，TCP 为每个发射的数据字节指定序号。这使得接收器可以以正确的顺序重组字节。另外，由于每个字节是排序的，因此，每个字节可被确认以证实成功的传输。

许多计算机系统和其它装置具有处理多种计算任务的主机处理

器（例如，通用中央处理器（CPU））。这些任务经常包括处理网络业务。网络业务和连接速度的增加对主机处理器资源提出了更多的需求。为至少在一定程度上减轻这种负担，网络协议卸载引擎可从主机处理器卸载不同的网络协议操作。例如，传输控制协议（TCP）
5 卸载引擎（TOE）可为发送/接收的 TCP 段执行一个或多个 TCP 操作。

附图简述

图 1A-1E 示出网络协议卸载引擎的操作。

图 2 是网络协议卸载引擎的示例实施图。

10 图 3 是包括网络协议卸载引擎的网络接口卡图。

详细说明

网络协议卸载引擎可对分组执行多种协议操作。通常，卸载引擎处理分组的方式是暂时在存储器中存储分组、对分组执行协议操作并将结果转发到主机处理器。引擎使用的存储器可包括本地芯片
15 上存储器、专用于引擎的副 RAM 存储器、主机存储器等等。引擎使用的这些不同的存储器可能在等待时间（发出存储器请求与收到响应之间的时间）、容量和其它特征方面不同。因此，用于存储分组的存储器可对整体引擎性能产生重大影响，特别是在引擎尝试维持高速连接的“线速”时。
20

其它因素可使卸载引擎的存储器管理变得复杂。例如，引擎可存储比其它分组更长的一些分组。例如，引擎可缓冲失序到达的段，直至有序数据到达。另外，分组大小可有很大的不同。例如，流式视频数据可通过大量的小分组输送，而大的文件传送可通过少量的
25 很大分组输送。

图 1A-1E 示出示例卸载引擎 102 实施的操作，该实施可以以可能加快分组处理速度并有效地处理一般在网络业务中承载的不同大小的分组的方式灵活地处理存储器管理。在图 1A 所示的实施

络协议卸载引擎 102（例如，TOE）可选择在多种存储器资源中存储分组数据，包括与引擎在同一芯片上的存储器 106（芯片上存储器）和/或芯片外存储器 108。为协调存储器 106、108 中的分组存储，引擎 102 维持存储器映象 104，该映象通常映射不同存储器资源 106、108 提供的多部分存储空间。在所示实施中，映象 104 分成对应于不同存储器的不同部分。例如，104a 部分映射芯片上存储器 106 的存储空间，而 104b 部分映射芯片外存储器 108 的存储空间。

映象部分 104a、104b 具有多个单元（示为框），其中，各个单元对应于一定量的相关联存储空间。例如，映象 104 可实施为位映象，其中，映象 104 内的单独比特/单元标识 n 字节的存储空间。例如，对于 256 字节的块，单元#1 可对应于芯片上存储器 106 地址从 0x0000 到 0x00FF 的存储空间，而单元#2 可对应于地址从 0x0100 到 0x01FF 的存储空间。

单元的值指示存储器当前是否填有有效分组数据。例如，比特值“1”可标识存储有效分组数据的存储器，而“0”标识可供分配的存储器。图 1A 以示例方式显示了 104a 部分内的两个带“x”的单元，标识了芯片上 106 存储器的占用部分。

不同存储器 106、108 可能形成或不形成相连的地址空间。换言之，与一个部分 104a 中最后单元相关联的存储器地址可能与另一部分 104b 中的第一个单元相关联的存储器地址无关系。另外，不同的存储器 106、108 可以是相同或不同类型的存储器。例如，芯片外存储器 108 可以是 SRAM，而芯片上存储器 106 是将地址“关键字”与存储的数据相关联的内容可寻址存储器（CAM）。

映象 104 可以使引擎 102 对存储接收的分组数据 100 的位置进行精确的控制。例如，映象 104 可用于确保给定分组的数据完整地存储在单个存储器资源 106、108 内，或者甚至在给定存储器 106、108 的相连存储单元内。

如图 1A 中所示，引擎 102 通过使用存储器映象 104 为分组数据

100 的存储分配 (112) 存储空间而处理分组 100。在分配的部分中存储 (114) 分组数据 100 后, 引擎 102 可以对分组 100 执行协议操作 (例如, TCP 操作)。图 1B-1E 更详细地示出引擎 104 的示例操作。

5 如图 1B 中所示, 引擎 102 分配 (112) 存储空间以存储分组数据 100。此类分配可包括用于存储分组的存储器 106、108 的选择。此选择可基于多种因素。例如, 可执行选择以确保可能时给定的存储器具有足够的可用容量来存储分组 100 的完整内容。例如, 引擎可访问与每个映象 104 部分相关联的“空闲单元”计数器 (未示出),
10 以确定该部分是否具有足够的单元适应分组的大小。如果不足, 则引擎可对另一存储器重复此进程, 或者最终将分组分布在不同的存储器中。

 另外, 可执行选择以确保可能时选择的存储器可提供足够的相连存储空间来存储分组。例如, 引擎 102 可搜索存储器映象部分 104a、
15 104b 以查找表示足够存储分组 100 的存储空间的多个连续空闲单元。虽然此类方案可能将 104a 部分映象成分散的空闲和占用单元, 但典型网络业务中发现的多种分组大小可在它们形成时自然填充此类空洞。或者, 数据分组可扩散到不相连的存储空间。此类实施可使用链接列表方案, 将不相连的存储空间链接在一起以形成完整的分
20 组。

 存储空间分配可基于其它因素。例如, 引擎 102 在可能时可在芯片上 106 存储器中存储“快速路径”数据 (例如, 在进行连接的数据段), 而将“慢速路径”数据 (例如, 连接设定段) 转移到芯片外 108 存储器。同样地, 选择可基于其它分组属性和/或内容。例
25 如, 在等待有序字节时, 具有将字节标识为失序的序号的 TCP 段可存储在芯片外 108。

 在图 1B 所示的示例中, 分组 100 具有需要两个单元的大小, 并且分配了芯片上存储器 106 内相连存储空间对应的单元。如图所示,

用于芯片上 106 存储器的映象 104 的 104a 部分内的相连单元设为占用（标有粗体“x”的单元）。如图 1C 中所示，确定与单元相关联的存储器地址（例如，第一部分单元的地址+[单元索引*单元大小]），请求使用（例如，执行 malloc）该存储器地址，并且将其用于存储
5 分组数据 100。

由于大多数分组处理操作可基于分组报头中包括的信息执行，因此，引擎 102 可分割存储的分组使得分组和/或段报头存储在与一个存储器映象 104 单元相关联的存储器中，而分组的有效负荷存储在与其它单元相关联的存储器中。引擎可能可以将分组分割到不同
10 存储器中，例如，通过将报头存储在快速芯片上 106 存储器中，而有效负荷存储在较慢速芯片外 108 存储器中。在此类解决方案中，诸如从报头部分到有效负荷部分的指针等机制将这两个部分链接在一起。或者，可存储分组数据而对报头无特殊处理。

如图 1D 中所示，在存储器中存储分组后（或与其同时），引擎
15 102 可根据引擎支持的网络协议处理分组 100。之后，引擎 102 可将分组数据传送到主机处理器可访问的存储器，例如，经直接存储器存取（DMA）传送到主机存储器（例如，主机处理器芯片集内的存储器）。

引擎 102 可能会尝试保留给定资源的存储空间。例如，虽然芯
20 片上存储器 106 可提供比芯片外存储器 108 更快的数据访问，但芯片上存储器 106 可提供的容量少得多。因此，如图 1E 中所示，引擎 102 可将芯片上存储器 106 中存储的分组数据移到芯片外存储器 108。例如，引擎 102 可标识存储在芯片上 106 存储器中的“陈旧”分组数据，如接收的失序 TCP 段字节或尚未由主机套接字进程分配
25 主机存储器的数据（例如，未收到该连接的公布“套接字接收”或“套接字接收消息”）。一些情况下，此类移动有效地表示与初始存储空间分配 112（图 1B）期间评估这些因素相比在芯片外存储数据的延迟决定。

如图所示，确定在存储器资源 106、108 之间移动至少部分分组后，引擎解除芯片上 106 存储器的分配（例如，将单元标记为空闲），并分配与芯片外 108 存储器相关联的映象 104 的 104b 部分内的空闲单元，在对应的芯片外 108 存储器中存储分组数据，并腾空先前使用的芯片上存储器的部分。

图 1A-1E 示出示例实施的操作。多种其它实施可使用上述技术。例如，引擎可不试图分配相连的存储空间，但可改为创建一个或多个存储器资源中不相连存储单元上的分组数据的链接列表。虽然重组分组的时间可能更长，但此技术可减轻可能产生的映象碎片。

另外，引擎 102 可将映象部分分成提供预分配缓冲区大小的子部分而不是统一的粒度。例如，104a 部分的一些单元可组成三单元的集合，而其它单元组成四单元的集合。引擎可将这些集合内的单元作为组分配或腾空。这些预分配的组可允许引擎 102 将为可用存储空间而对映象 104 的搜索限制为具有足够大小集合以保存分组数据的子部分。例如，对于需要四个单元的分组，引擎可先搜索具有预分配的四单元集合的存储器映象子部分。此类预分配的组可能可加快分配的速度并减少存储器碎片。

在另一替代实施中，各个单元可存储指定哪个存储器 106、108 与单元相关联的标识符，而不是将存储器映象 104 分成部分。例如，单元可具有额外的比特，它标识数据是在芯片上 106 还是芯片外 108 存储器中。在此类实施中，引擎可读取该芯片上/芯片外比特以确定检索与单元相关联的数据时要读取的存储器。例如，某个单元“N”可与地址 0xAAAA 相关联。然而，此地址可能在芯片外存储器 108 中，或者是形成芯片上存储器 106 的 CAM 中存储的地址关键字。因此，要访问正确的存储器，引擎可读取芯片上/芯片外比特。虽然，这可强加额外的操作以执行数据检索并在分配单元给分组时设置该比特，但通过翻转与分组的缓冲区相关联的单元的芯片上/芯片外比特，并移动数据，可执行将数据从一个存储器移到另一存储器的移

动。这可以避免对与目的地存储器相关联的空闲单元进行搜索。

图 2 示出 TCP 卸载引擎 170 逻辑的示例实施。在所示实施中，IP 处理 172 逻辑对接收的分组 100 执行多种操作，如验证存储在分组内的 IP 校验和、执行分组过滤（例如，丢弃特定来源的分组）、标识封装的分组的传输层协议（例如，TCP 或用户数据报协议（UDP））等等。逻辑 172 可使用如上所述的存储器映象执行对芯片上和/或芯片外存储器的初始存储空间分配。

在所示示例中，对于包括 TCP 段的分组 100，协议控制块（PCB）查找 174 逻辑尝试检索有关在进行连接的信息，如下一预计的序号、连接窗口信息、连接错误和标记及连接状态。可基于从分组 IP 来源和目的地地址获得的关键字、传输协议及来源和目的地端口检索连接数据。

基于为段检索的 PCB 数据，TCP 接收 176 逻辑处理接收的分组。此类处理可包括段重组、更新 TCP 状态机器的状态（例如，“已关闭” CLOSED、“侦听” LISTEN、SYN RCVD、SYN SENT、“已建立” ESTABLISHED 等等）、选项和标记处理、窗口管理、ACK-确认消息生成及“请求注解”（RFC）793、1122 和/或 1323 中所述的其它操作。

基于收到的段，TCP 接收 176 逻辑可选择发送先前存储在芯片上存储器中的分组数据到芯片外存储器。例如，TCP 接收 176 逻辑可基于段的报头数据，将段归类为“快速路径”或“慢速路径”。例如，没有有效负荷的段或设有 SYN 或 RST 标记的段可以以较低的紧急程度来处理，因为此类段可能是“管理性的”（例如，打开或关闭连接）而不是承载数据，否则数据可能失序。同样地，如果先前分配了芯片上存储，则引擎可将“慢速路径”数据移到芯片外（参见图 1E）。

在 TCP 处理后，结果（例如，重组的字节流）被传送到主机。所示实施具有 DMA 逻辑以将数据从芯片上 184 和芯片外 182 存储器

传送到主机存储器。对于存储在芯片上的数据和存储在芯片外的数据，逻辑可使用不同的 DMA 方法。例如，芯片外存储器可以是主机存储器的一部分。在此类情况下，芯片外到芯片外 DMA 可使用在主机存储器内移动数据的复制操作，而无需在主机存储器与其它存储器（例如，NIC 存储器）之间来回移动数据。

实施还具有逻辑 180，以处理与和卸载引擎 170 接口连接的进程（例如，主机套接字进程）的通信。TCP 接收 176 进程不断检查以看看是否有数据可转发到主机，即使此类数据只是特定段内包括的数据子集。这不但更快地腾空存储空间，而且防止引擎 170 在数据输送中引入过多的延迟。

引擎逻辑可包括其它组件。例如，逻辑可包括用于根据远程直接存储器存取（RDMA）和/或 UDP 处理分组的组件。另外，图 2 示出引擎 170 的接收路径。例如，引擎 170 还可包括发射路径逻辑，它执行 TCP 发射操作（例如，生成段以承载数据流、处理数据重新传输和超时等等）。

图 3 示出具有卸载引擎 156 的装置 150 的示例。所示装置 150 是网络接口卡（NIC）的示例。如图所示，NIC 150 具有端接物理网络连接（例如，有线、无线或光学连接）的物理层（PHY）装置 152。第 2 层装置 154（例如，以太网介质访问控制器（MAC）或同步光纤网络（SONET）成帧器）处理由 PHY 152 接收的比特，例如，通过标识称为帧的逻辑比特组内的分组。卸载引擎 156 对经 PHY 152 和第 2 层装置 154 接收的分组执行协议操作。这些操作的结果经主机接口（例如，到主机总线的外围组件互连（PCI）接口）传递到主机。此类通信可包括 DMA 数据传送和/或警告主机处理器有结果数据的中断信令。

卸载引擎虽然示为 NIC，但可并入多种装置内。例如，通用处理器芯片集可具有卸载引擎组件。另外，部分或所有 NIC 可包括在母板上，或者包括在已经在母板上的另一芯片（例如，通用输入/输

出 (I/O) 芯片) 内。

引擎组件可使用多种硬件和/或软件配置实施。例如, 逻辑可实施为专用集成电路 (ASIC)、门阵列和/或其它电路。卸载引擎可在其自己的芯片上提供 (例如, 如图 1A-1E 中所示, 芯片上存储器位于引擎的芯片内)、可由多个芯片形成或者可与其它电路集成。

技术可在计算机程序中实施。此类程序可存储在计算机可读介质上, 并包括为处理器 (例如, 控制器或引擎处理器) 编程的指令。例如, 逻辑可通过编程的网络处理器实施, 如具有多个多线程处理器的网络处理器 (例如, Intel® 的 IXP 1200 和 IXP 2400 系列网络处理器)。此类处理器可具有适用于分组处理操作的精简指令集计算 (RISC) 指令集。例如, 这些指令集可能缺少用于浮点运算或整数除法和/或乘法的指令。

同样地, 多种实施可使用上述的一种或多种技术。例如, 虽然示例实施描述为 TCP 卸载引擎, 但卸载引擎可实施在网络协议栈内不同层 (例如, 作为异步传送模式 (ATM)、ATM 适配层、RDMA、实时协议 (RTP)、高级数据链路控制 (HDLC) 等等) 的一个或多个协议的操作。另外, 虽然在上面通常描述为 IP 数据报和/或 TCP 段, 但引擎处理的分组可以是第 2 层分组 (称为帧)、ATM 分组 (称为信元) 或 SONET 上的分组 (POS) 分组。

其它实施例在下列权利要求的范围内。

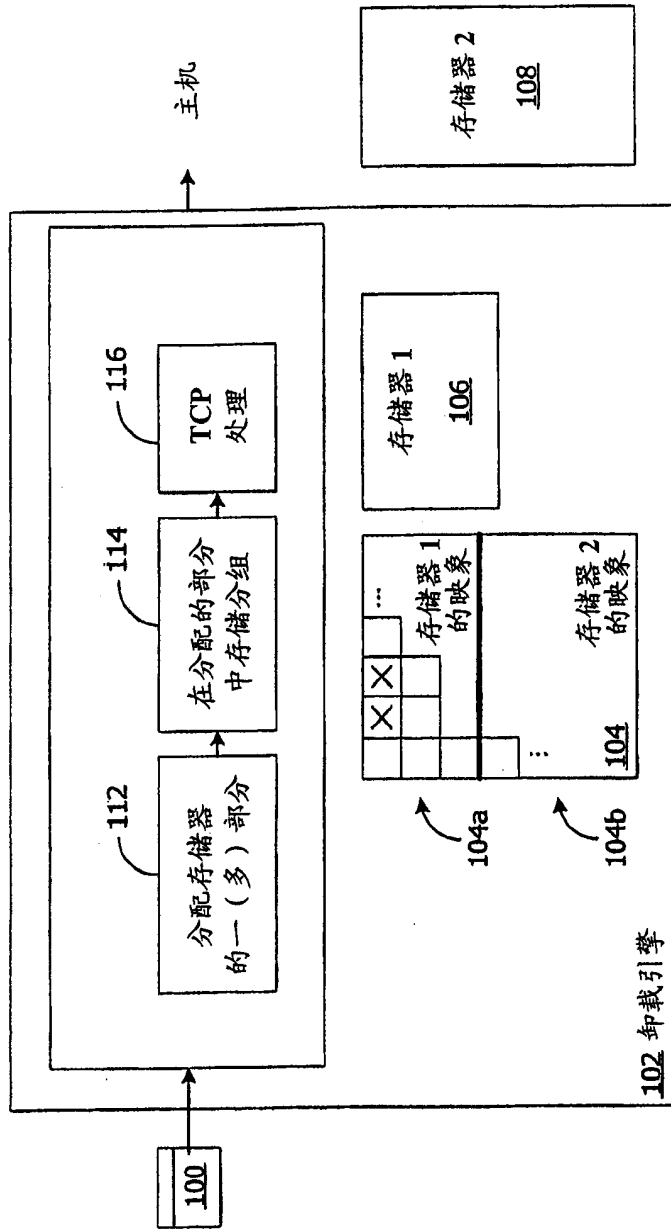


图 1A

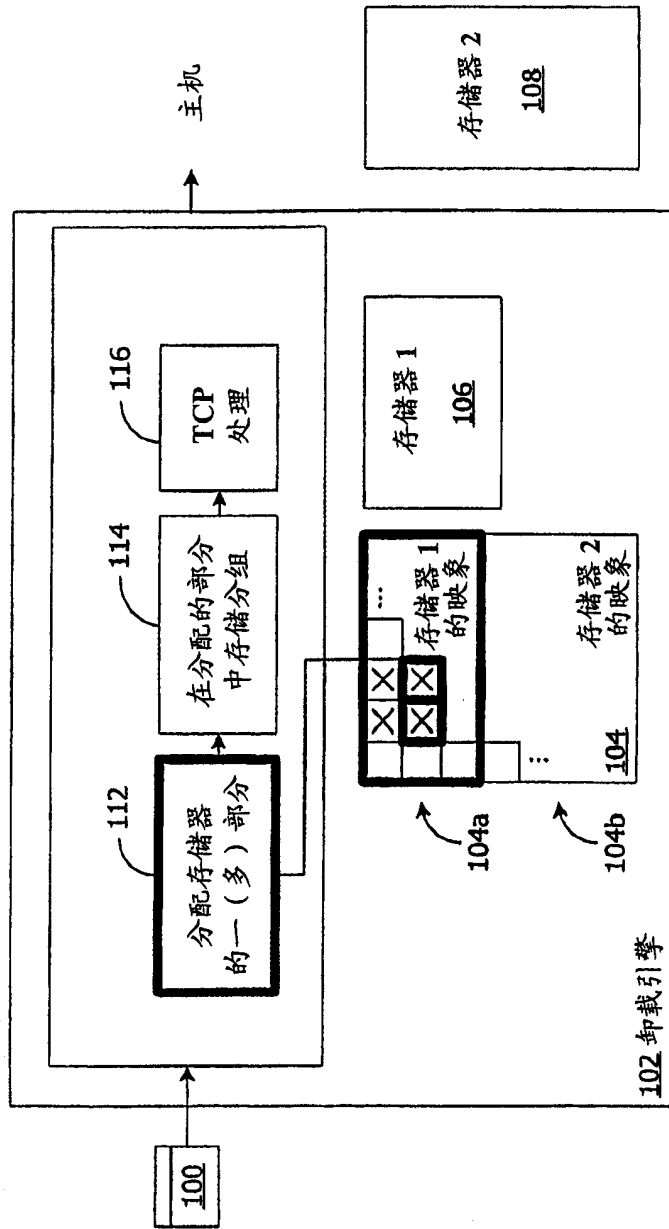


图 1B

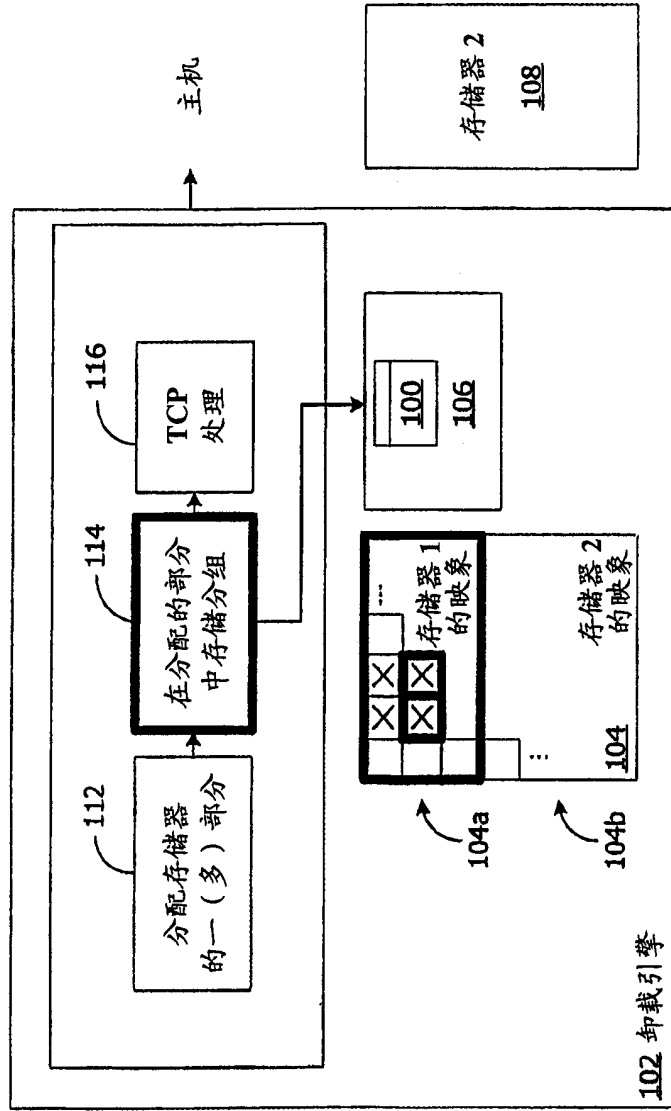


图 1C

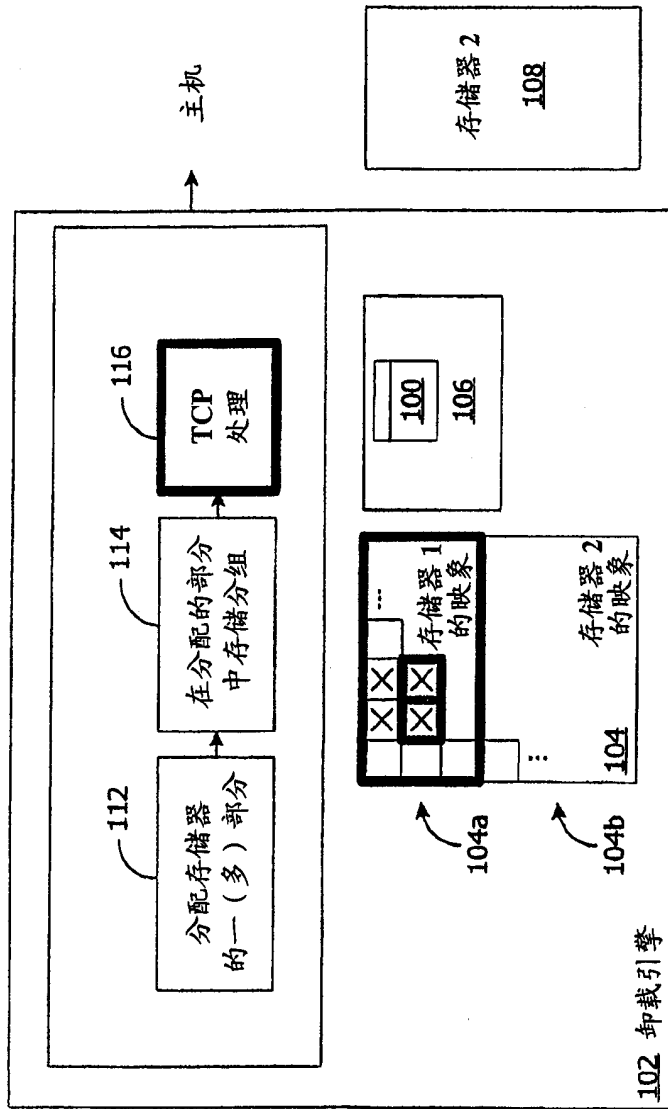


图 1D

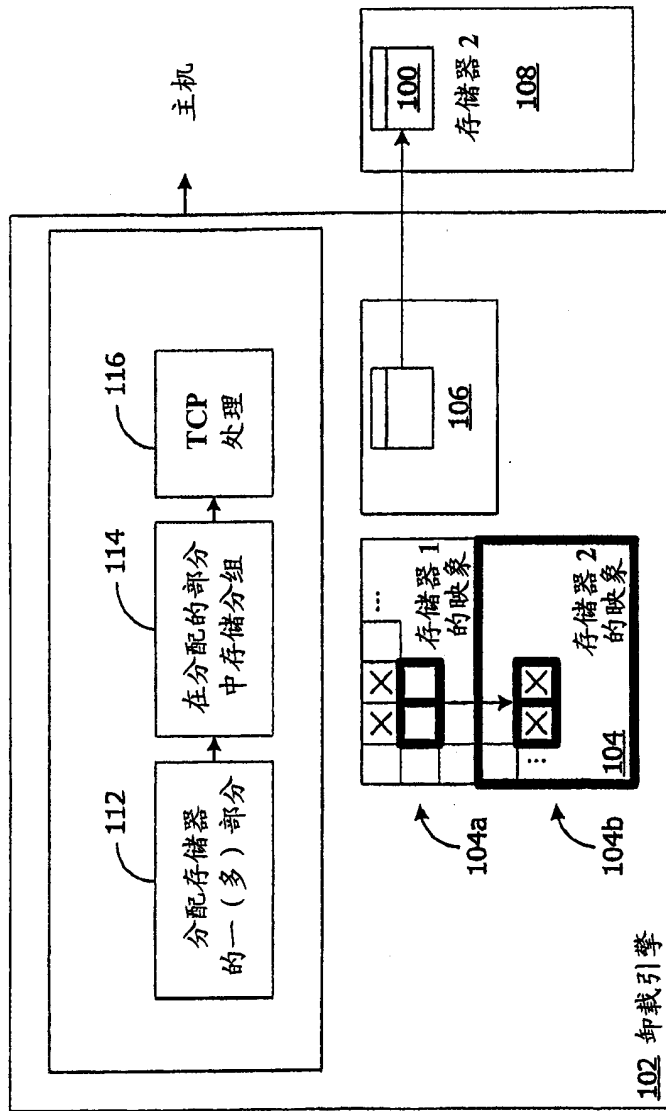


图 1E

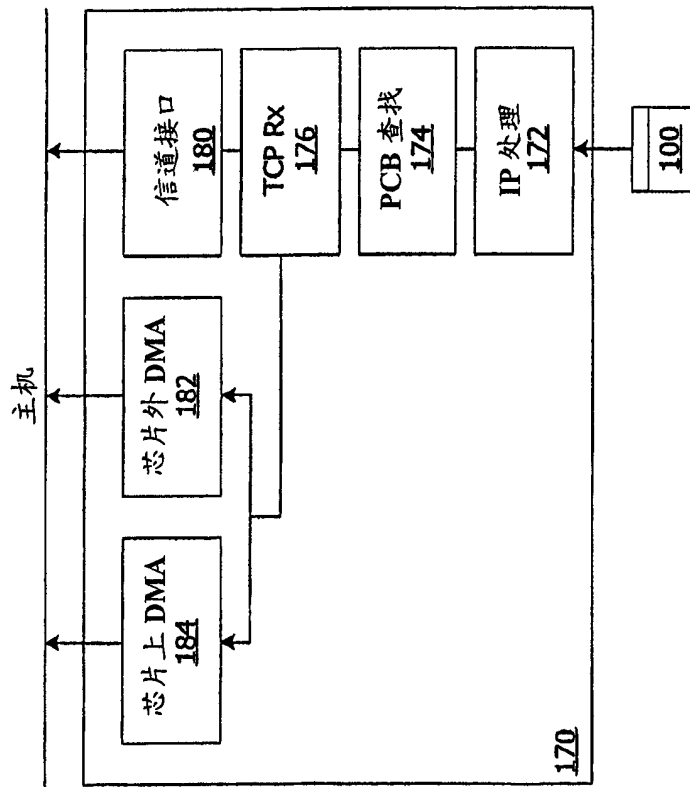


图 2

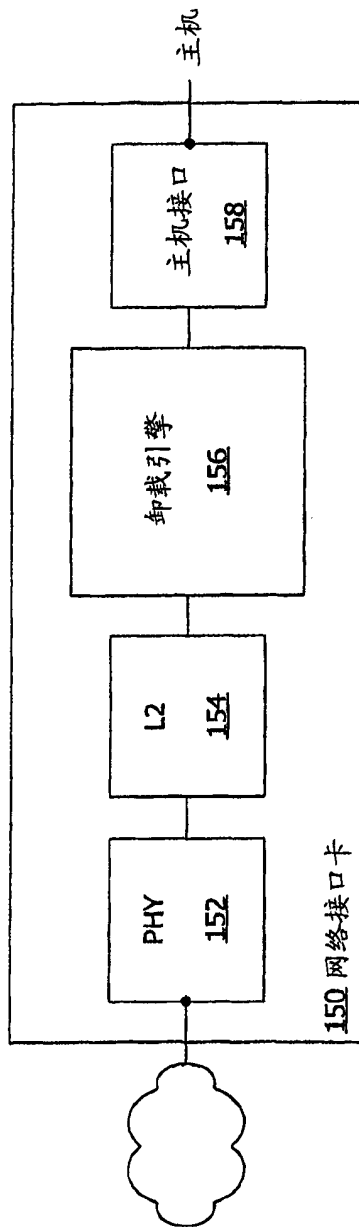


图 3