国 家 知 识 产 权 局

250014

山东省济南市历下区经十路 17703 号华特广场 B510 室 济南圣达知识产权代理有限公司
于凤洋(0531-82961551)

发文日：

2024 年 11 月 20 日

申请号：202411660756.9　　**发文序号：**2024112000920990

专 利 申 请 受 理 通 知 书

根据专利法第 28 条及其实施细则第 43 条、第 44 条的规定，申请人提出的专利申请已由国家知识产权局受理。现将确定的申请号、申请日等信息通知如下：

申请号：2024116607569
申请日：2024 年 11 月 20 日
申请人：山东省计算中心（国家超级计算济南中心）
发明人：谭立状,董鑫,史慧玲,张玮
发明创造名称：基于接收注册表机制的数据传输优化方法及系统
经核实，国家知识产权局确认收到文件如下：
权利要求书 1 份 3 页,权利要求项数 ： 10 项
说明书 1 份 9 页
说明书附图 1 份 4 页
说明书摘要 1 份 1 页
发明专利请求书 1 份 5 页
实质审查请求书 文件份数：1 份
申请方案卷号：2024710336

提示：

1.申请人收到专利申请受理通知书之后，认为其记载的内容与申请人所提交的相应内容不一致时，可以向国家知识产权局请求更正。

2.申请人收到专利申请受理通知书之后，再向国家知识产权局办理各种手续时，均应当准确、清晰地写明申请号。

国家知识产权局 专利申查业务章 1101081359734

审 查 员：自动受理
联系电话：010-62356655

审查部门：初审及流程管理部

200101 2023.03

纸件申请，回函请寄：100088 北京市海淀区蓟门桥西土城路 6 号 国家知识产权局专利局受理处收
电子申请，应当通过专利业务办理系统以电子文件形式提交相关文件。除另有规定外，以纸件等其他形式提交的文件视为未提交。

权 利 要 求 书

1. 基于接收注册表机制的数据传输优化方法，其特征在于，应用于发送端，包括：

CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令。

2. 如权利要求 1 所述的基于接收注册表机制的数据传输优化方法，其特征在于，DMA 操作采用 Scatter−Gather 模式，使得 DMA 引擎能拉取分散在不同内存区域的数据；在大传输队列中，大型数据包将被分割成多个小的数据包，以便传输，未超过阈值的命令则进入常规队列等待处理；当 DMA 引擎拉取新的分段命令后，只需在队列头部更新基地址的偏移量和要传输的字节数，在发出最后一个命令时，特殊的状态标志保持未激活状态，形成原始完整传输的本地完成通知。

3. 如权利要求 1 所述的基于接收注册表机制的数据传输优化方法，其特征在于，DMA 引擎对每个新拉取的 DMA 操作都在网络接口的 DMA 操作表中进行记录，DMA 操作表为每个数据包分配一个唯一的操作号码，该操作号码随数据包发送至网络，并在数据包确认时返回，用于记录成功传输的数据包数量。

4. 如权利要求 3 所述的基于接收注册表机制的数据传输优化方法，其特征在于，DMA 操作表跟踪每个单独的数据包，并为每个数据包设定超时计时器，如果在超时时间内未收到确认或收到负面确认，将在 DMA 操作表中创建重传操作条目。

5. 基于接收注册表机制的数据传输优化方法，其特征在于，应用于接收端，包括：

接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

6. 如权利要求 5 所述的基于接收注册表机制的数据传输优化方法，其特征在于，当新数据包的到达后，计算其目标地址，并与接收注册表中的条目进行匹配，如果找到匹配的条目，

则继续处理；否则，将数据包发送到逃逸通道，则启动相关计时器，如果计时器超时，则丢弃该数据包及其相关数据。

7. 如权利要求 5 所述的基于接收注册表机制的数据传输优化方法，其特征在于，若匹配成功，则计算新数据包的掩码数值，如果新数据包的掩码数值未变，则说明是重复的，则立即丢弃该数据包，如果新数据包不是重复的，将更新掩码数值，并检查是否所有预期的数据包都已接收完毕，如果掩码显示所有位都已置位，表明操作已完成，随即向发送端发送本地完成通知。

8. 基于接收注册表机制的数据传输优化系统，其特征在于，包括发送端和接收端，

在所述发送端中，CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

在所述接收端中，接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

9. 如权利要求 8 所述的基于接收注册表机制的数据传输优化系统的传输优化方法，其特征在于，包括：

发送端中利用 CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成

功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

所述接收端接收到发送端的 RDMA 事务的数据包后，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

10. 一种电子设备，其特征在于，包括：处理器、存储器以及计算机程序；其中，处理器与存储器连接，计算机程序被存储在存储器中，当电子设备运行时，所述处理器执行所述存储器存储的计算机程序，以使电子设备执行实现如权利要求 9 所述的基于接收注册表机制的数据传输优化系统的传输优化方法。

基于接收注册表机制的数据传输优化方法及系统

技术领域

[0001] 本公开涉及高性能计算与数据传输优化技术领域，具体涉及基于接收注册表机制的数据传输优化方法及系统。

背景技术

[0002] 本部分的陈述仅仅是提供了与本公开相关的背景技术信息，不必然构成在先技术。

[0003] 在高性能计算（HPC）系统架构优化过程中，传统通信协议如以太网和 Infiniband 虽然在众多领域中表现优异，但在基于现场可编程门阵列（FPGA）的 HPC 环境中却面临诸多挑战。这些挑战主要来源于传统协议在系统性能与资源利用上的局限性：一方面，传统协议通常依赖中央处理单元（CPU）频繁参与，导致不必要的内存复制和上下文切换，进而增加数据处理延迟，降低系统的实时响应能力；另一方面，传统协议在 FPGA 平台上的实现消耗大量逻辑资源，从而直接制约了系统性能的进一步提升。

[0004] 为解决上述局限性，现有技术中已开始研究更为高效的数据传输机制，其中远程直接内存访问（RDMA）技术因其独特优势而备受关注。RDMA 技术通过直接在内存之间传输数据，绕过 CPU 干预，实现数据的“零拷贝”传输，显著减轻了 CPU 负担并降低了延迟。此外，RDMA 的无连接特性使得 FPGA 专用加速器之间能够实现直接且高效的通信，无需建立和维护复杂的连接状态，从而进一步简化了通信过程，提升了系统的灵活性和可扩展性。

[0005] 但是目前基于 RDMA 的通信方案在 HPC 环境下的 FPGA 专用加速器间数据传输仍面临众多问题，包括：

（1）数据传输延迟和效率问题，在 FPGA 加速器应用中 RDMA 协议传输规模较大时可能导致系统资源的过度占用和计算瓶颈；

（2）接收端数据处理瓶颈问题，接收端处理乱序极易导致更过度阻塞或资源浪费；

（3）重复或延迟数据包处理问题，传统 RDMA 协议面临重复传输或延迟到达的数据包时可能导致流水线阻塞。

发明内容

[0006] 本公开为了解决上述问题，提出了基于接收注册表机制的数据传输优化方法及系统，通过微架构设计将大型 RDMA 传输分割为多个较小传输，在接收端引入接收注册表机制，通过特殊状态标志和掩码机制确保数据传输的准确性和及时性，引入逃逸通道以及超时机制避

免流水线阻塞以及处理重复数据包或者延迟到达数据包，保证数据传输的高效性、稳定性和可靠性。

[0007] 根据一些实施例，本公开采用如下技术方案：

基于接收注册表机制的数据传输优化方法，应用于发送端，包括：

CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令。

[0008] 根据一些实施例，本公开采用如下技术方案：

基于接收注册表机制的数据传输优化方法，应用于接收端，包括：

接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0009] 根据一些实施例，本公开采用如下技术方案：

基于接收注册表机制的数据传输优化系统，包括发送端和接收端，

在所述发送端中，CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

在所述接收端中，接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数

据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0010] 根据一些实施例，本公开采用如下技术方案：

基于接收注册表机制的数据传输优化系统的传输优化方法，包括：

发送端中利用 CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分割为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接分段传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

所述接收端接收到发送端的 RDMA 事务的数据包后，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0011] 根据一些实施例，本公开采用如下技术方案：

一种电子设备，包括：处理器、存储器以及计算机程序；其中，处理器与存储器连接，计算机程序被存储在存储器中，当电子设备运行时，所述处理器执行所述存储器存储的计算机程序，以使电子设备执行实现所述的基于接收注册表机制的数据传输优化系统的传输优化方法。

[0012] 与现有技术相比，本公开的有益效果为：

本公开的基于接收注册表机制的数据传输优化方法，提出了一种微架构设计，支持无连接的可靠传输。在发送端支持将大型的 RDMA 传输分割成多个较小的传输，来减轻大型传输引入的延迟，这种方法允许多个数据段并行处理和传输，更加充分地利用网络带宽，提高整体传输效率。

[0013] 本公开的基于接收注册表机制的数据传输优化方法，在接收端引入了一种接收注册表

机制，用于记录和跟踪 RDMA 操作。通过使用特殊的状态标志和掩码机制，系统能够识别数据包是否属于同一 RDMA 操作，并在操作完成时提供本地和远程通知，确保了数据传输的准确性和及时性。对于尚未在接收注册表中注册的数据包，并设计了一个逃逸通道，以避免流水线阻塞。同时，引入了超时机制，用于丢弃那些重复或延迟到达的数据包，从而保证了数据传输的稳定性和可靠性。

附图说明

[0014] 构成本公开的一部分的说明书附图用来提供对本公开的进一步理解，本公开的示意性实施例及其说明用于解释本公开，并不构成对本公开的不当限定。

[0015] 图 1 为本公开实施例的基于接收注册表机制的数据传输优化方法应用于发送端的流程图；

图 2 为本公开实施例的基于接收注册表机制的数据传输优化方法应用于接收端的流程图；

图 3 为本公开实施例的传输层收发控制流程框图；

图 4 为本公开实施例的掩码更新流程图；

图 5 为本公开实施例的分段传输模式与大传输模式的延时对比图。

具体实施方式

[0016] 下面结合附图与实施例对本公开作进一步说明。

[0017] 应该指出，以下详细说明都是例示性的，旨在对本公开提供进一步的说明。除非另有指明，本文使用的所有技术和科学术语具有与本公开所属技术领域的普通技术人员通常理解的相同含义。

[0018] 需要注意的是，这里所使用的术语仅是为了描述具体实施方式，而非意图限制根据本公开的示例性实施方式。如在这里所使用的，除非上下文另外明确指出，否则单数形式也意图包括复数形式，此外，还应当理解的是，当在本说明书中使用术语“包含”和/或“包括”时，其指明存在特征、步骤、操作、器件、组件和/或它们的组合。

[0019] 实施例 1

本公开的一种实施例中提供了一种基于接收注册表机制的数据传输优化方法，应用于发送端，包括：

发送端的 CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的

RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分段为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令。

[0020] 具体地，当需要传输数据时，CPU 或加速器会发起直接内存访问（DMA）操作请求。在此过程中，Xilinx 的 CDMA IP 作为 DMA 引擎，负责从命令队列中拉取这些 DMA 操作。这些操作采用 Scatter-Gather 模式，使得 DMA 引擎能够有效处理分散在不同内存区域的数据。

[0021] 其中，采用 Scatter-Gather 模式，使得 DMA 引擎能够有效处理分散在不同内存区域的数据，包括：

1）DMA 控制器构建一个描述符表，这个表包含了多个内存块的地址和大小，这些内存块可能分布在物理内存的不同位置。

[0022] 2）DMA 控制器根据描述符表中的信息，从多个源内存地址读取数据并将其合并写入单一目标地址，或者将数据从单一源地址读取后分散写入多个目标内存地址；

3）数据传输完成后，DMA 控制器会发出中断信号。通知 CPU，CPU 最后进行后续的处理。

[0023] 进一步地，对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，对于那些超出预设阈值大小的 RDMA 命令，它们会被送入大传输队列进行特殊处理。在这一队列中，大型数据包将被分割成多个较小的数据包，以便传输，从而减少系统延迟。相反，未超过阈值的命令则进入常规队列等待处理。当 DMA 引擎拉取新的分段命令后，只需在队列头部更新基地址的偏移量和要传输的字节数。在发出最后一个命令时，特殊的状态标志保持未激活状态，因此可以形成原始完整传输的本地完成通知。

[0024] 其中，根据预设阈值划分大传输队列和常规队列，包括：

CPU/加速器发出的 RDMA 命令包含内存基地址和要传输的字节数量。如果某个命令的大小超过了特定的阈值，则可以将其发送到特殊的“大传输”队列中。分段操作的最佳的值高度依赖于应用程序的结构，因此应在配置网络接口（NI）时由程序员定义。

[0025] 进一步地，对于那些超出预设阈值大小的 RDMA 命令，它们会被送入大传输队列进行特殊处理，包括：

当大传输队列的命令被处理时，需要在 DMA 操作表中为其分配一个状态标志。当具有此特殊状态标志的操作完成时，不会发布本地通知；只会向接收器发送通知。如果命令的大小为 M，分段大小为 N，则“大传输”队列的头部将保持 M/N 次操作不变。当 DMA 引擎拉取新

的分段命令后，只需在队列头部更新基地址的偏移量和要传输的字节数。在发出最后一个命令时，特殊状态标志保持未置位状态，因此可以形成原始完整传输的本地完成通知。

[0026] 进一步地，DMA 引擎对每个新拉取的 DMA 操作都在网络接口（NI）的 DMA 操作表（DMA OP 表）中进行记录，DMA 操作表为每个数据包分配一个唯一的操作号码（OP 号码），该操作号码随数据包发送至网络，并在数据包确认时返回，用于记录成功传输的数据包数量。

[0027] DMA 操作表负责跟踪每个单独的数据包，并为每个数据包设定超时计时器。如果在超时时间内未收到确认或收到负面确认，将在 DMA 操作表中创建重传操作条目。

[0028] 最后，系统采用无连接分段传输方式，数据包将以无序方式在网络中传输。当所有分段的数据包成功发送后，发送端通过 DMA 引擎更新本地状态，以通知处理器 DMA 操作已经完成。同时，发送端还会通知远端接收端，告知其在特定位置有新数据到达，使接收端能够开始处理接收到的数据。这一流程确保了数据传输的效率和可靠性。

[0029] 实施例 2

本公开的一种实施例中提供了一种基于接收注册表机制的数据传输优化方法，应用于接收端，包括：

接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0030] 作为一种实施例，当 RDMA 事务到达接收端时，系统首先检查数据包头部的“类型”字段。若该字段表明这是操作的起始部分，接收端便进行注册操作。在接收注册表中，记录下操作的预期地址范围和数据包数量。

[0031] 其中，若该字段表明这是操作的起始部分，接收端便进行注册操作，包括：

注册操作实际上是解析数据包头中“类型”字段的特殊条目，该特殊条目表示该数据包是否为此次操作的第一个传输数据包。当接收端检测到该特殊条目后，会在接收注册表记录下此次传输操作的地址范围和数据包数量。

[0032] 进一步地，随着新数据包的到达，系统会计算其目标地址，并与接收注册表中的条目进行匹配。如果找到匹配的条目，则继续处理；否则，将数据包发送到逃逸通道，则启动相

关计时器，如果计时器超时，则丢弃该数据包及其相关数据。

[0033] 其中，所述逃逸通道是一种用于处理乱序数据包的缓冲机制，由 FPGA 内部资源 ram 块实现，它允许接收器在尚未准备好处理特定 RDMA 操作时，临时存储这些数据包，从而避免了流水线的阻塞，并保证了数据的顺序性和完整性。

[0034] 如果匹配成功，硬件会计算新数据包的掩码数值，如果新数据包是重复的（即掩码未变），则立即丢弃该数据包。如果新数据包不是重复的，系统将更新掩码，并检查是否所有预期的数据包都已接收完毕。如果掩码显示所有位都已置位，表明操作已完成，系统随即提供本地完成通知。与等待发送方在收到最后一条确认消息后提供完成通知相比， 这节省了完整的往返数据包延迟时间。

[0035] 进一步地，计算新数据包的掩码数值，包括：

掩码是通过一个桶形移位器创建的，其操作基于第一个数据包头中的一个字段，该字段表示预期的数据包数量。为了形成适当的初始操作状态，在这个字段中移入零。在创建过程的最后一步添加一个 1，以表明已经接收到第一个数据包，这是开始此注册过程的必要条件。以具体参数为例，如果设想一个 4KB 的操作规模，一个 16KB 的基于 FPGA 网络的增强 RDMA 接收端通知掩码大小，以及一个 512B 的数据包大小，那么注册后的初始掩码将是'b1111 1111 1111 1111 1111 1111 1111 1111 0000 0001。这个掩码表示需要接收 8 个数据包，并且已经成功接收到了第一个数据包。

[0036] 进一步地，如果新数据包不是重复的，系统将更新掩码，包括：

如果检查到的表条目与传入数据匹配，则继续创建新掩码。使用条目的基地址和操作的字节数来计算表中正在检查的操作是否与到达的新数据包相关。然后为桶形移位器创建一个偏移量，该移位器生成一个掩码以引起位翻转。如果发现掩码全为 1，则操作必须已完成。

[0037] 如果 Newmask 等于 Originalmask，则数据包必须是重复的，可以丢弃。

[0038] 实施例 3

本公开的一种实施例中提供了一种基于接收注册表机制的数据传输优化系统，包括发送端和接收端，

在所述发送端中，CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分段为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

在所述接收端中，接收来自发送端的 RDMA 事务的数据包，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0039] 实施例 4

本公开的一种实施例中提供了一种基于接收注册表机制的数据传输优化系统的传输优化方法，包括：

发送端中利用 CPU 发起直接内存访问 DMA 操作请求，组成 RDMA 命令队列；CDMA IP 作为 DMA 引擎，从 RDMA 命令队列中拉取 DMA 操作；

对 RDMA 命令设置传输阈值，根据预设阈值划分大传输队列和常规队列，将超出预设阈值的 RDMA 命令送入大传输队列，并将大传输队列中的大型数据包分段为小数据包；

当 DMA 引擎拉取新的分段命令后，在队列头部更新基地址的偏移量和传输的字节数，采用无连接传输方式，将数据包以无序方式在网络中向接收端传输，当所有分段的数据包成功发送后，通过 DMA 引擎更新本地状态，同时，向接收端发送数据到达指令；

所述接收端接收到发送端的 RDMA 事务的数据包后，检查数据包头部的类型字段，若该字段是操作的起始部分，则在接收注册表中进行注册操作，记录操作的预期地址范围和数据包数量；

当新数据包到达，计算其目标地址，并与接收注册表中的条目进行匹配，并建立逃逸通道，根据接收注册表条目匹配结果，对到达的数据包进行逃逸判断并处理；

当完成所有接收操作后，向发送端发送本地完成通知。

[0040] 实施例 5

本公开的一种实施例中提供了一种电子设备，包括：处理器、存储器以及计算机程序；其中，处理器与存储器连接，计算机程序被存储在存储器中，当电子设备运行时，所述处理器执行所述存储器存储的计算机程序，以使电子设备执行实现基于接收注册表机制的数据传输优化

系统的传输优化方法。

[0041] 本公开是参照根据本公开实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0042] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0043] 上述虽然结合附图对本公开的具体实施方式进行了描述，但并非对本公开保护范围的限制，所属领域技术人员应该明白，在本公开的技术方案的基础上，本领域技术人员不需要付出创造性劳动即可做出的各种修改或变形仍在本公开的保护范围以内。

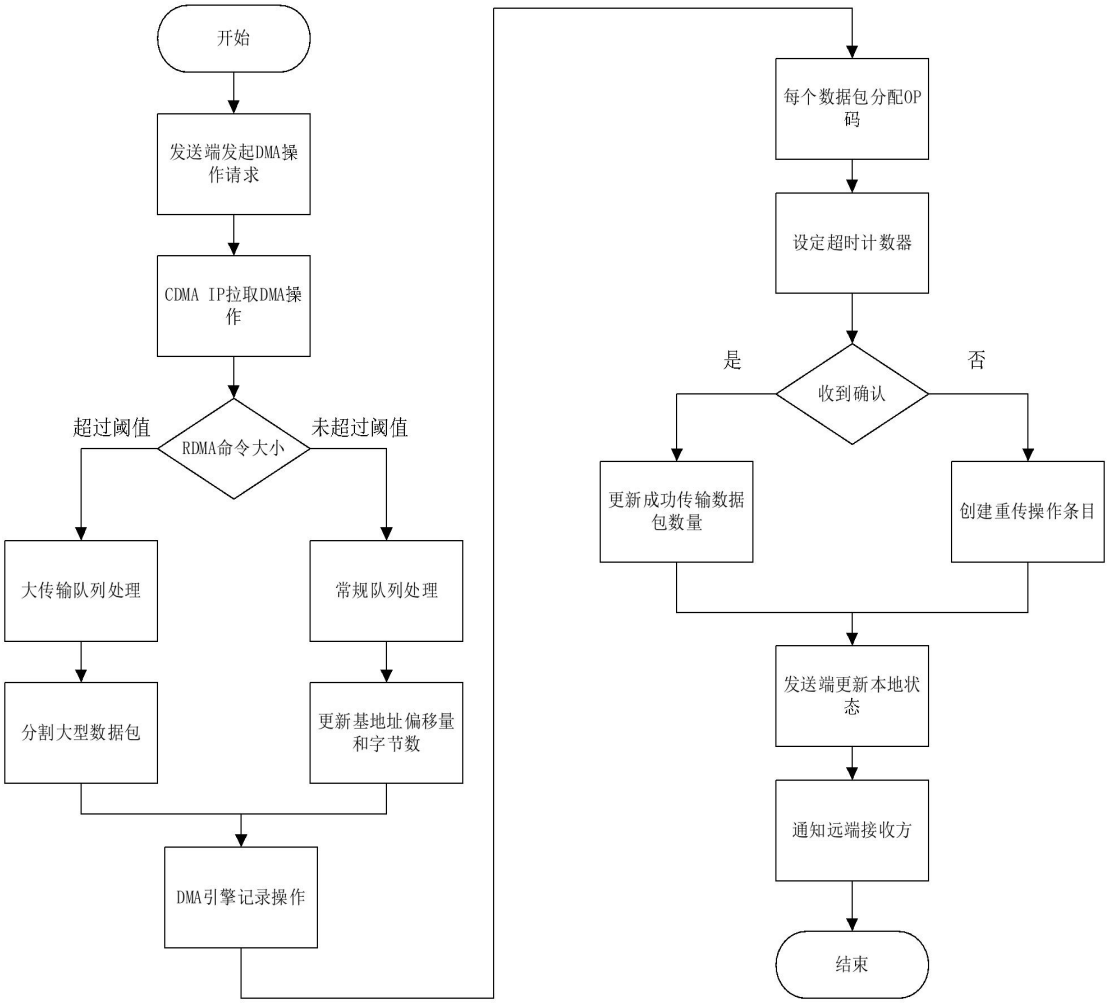
开始
发送端发起DMA操作请求
CDMA IP拉取DMA操作
超过阈值
RDMA命令大小
未超过阈值
大传输队列处理
常规队列处理
分割大型数据包
更新基地址偏移量和字节数
DMA引擎记录操作
每个数据包分配OP码
设定超时计数器
是
收到确认
否
更新成功传输数据包数量
创建重传操作条目
发送端更新本地状态
通知远端接收方
结束

图 1

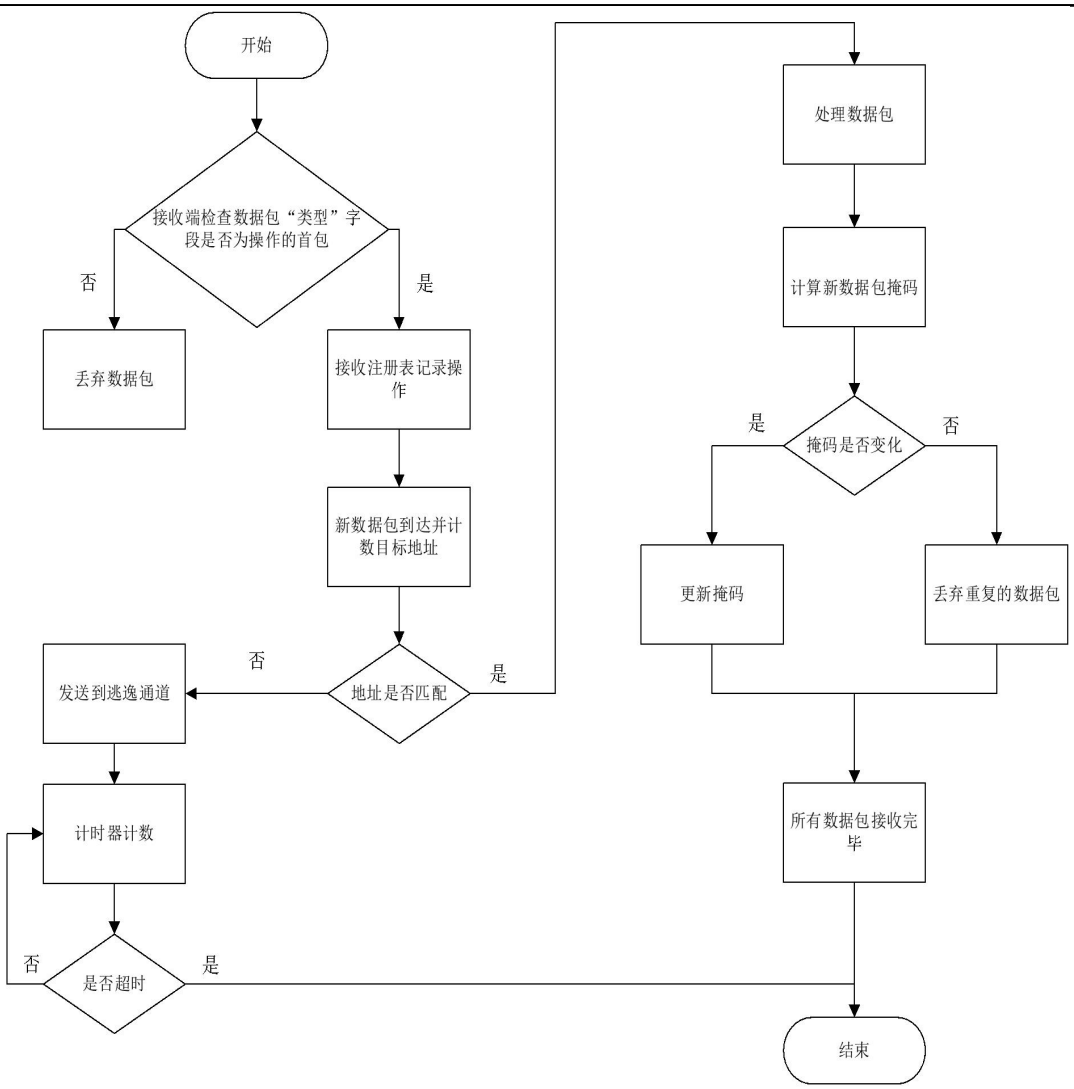
开始
接收端检查数据包“类型”字段是否为操作的首包
否
是
丢弃数据包
接收注册表记录操作
新数据包到达并计数目标地址
地址是否匹配
否
是
发送到逃逸通道
计时器计数
是否超时
否
是
处理数据包
计算新数据包掩码
掩码是否变化
是
否
更新掩码
丢弃重复的数据包
所有数据包接收完毕
结束

图 2

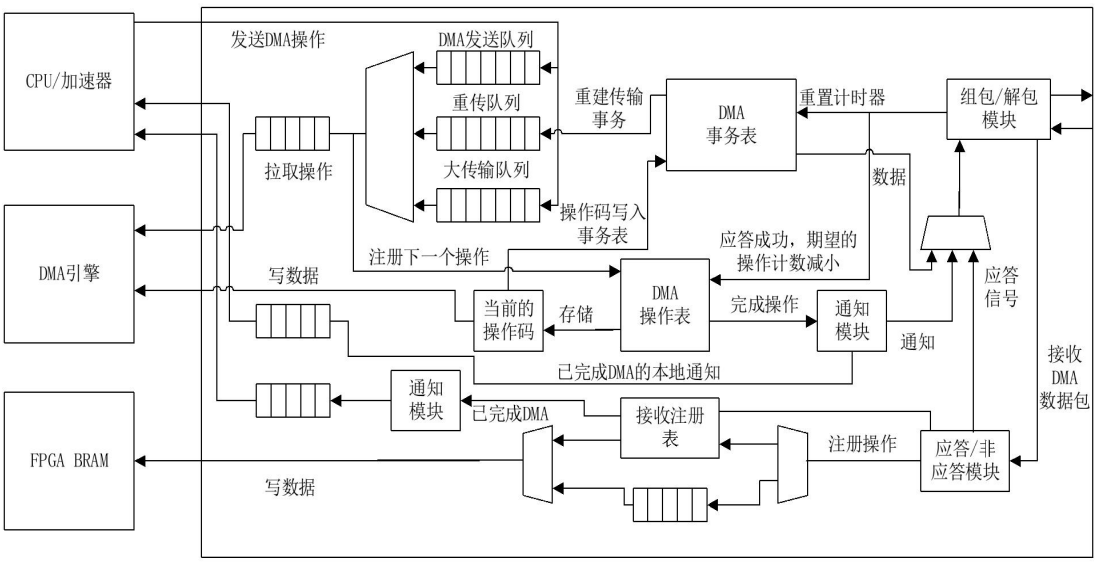
CPU/加速器
DMA引擎
FPGA BRAM
发送DMA操作
DMA发送队列
重传队列
大传输队列
拉取操作
重建传输事务
DMA事务表
重置计时器
组包/解包模块
数据
操作码写入事务表
注册下一个操作
应答成功，期望的操作计数减小
写数据
当前的操作码
存储
DMA操作表
完成操作
通知模块
通知
应答信号
已完成DMA的本地通知
接收DMA数据包
通知模块
已完成DMA
接收注册表
注册操作
应答/非应答模块
写数据

图 3

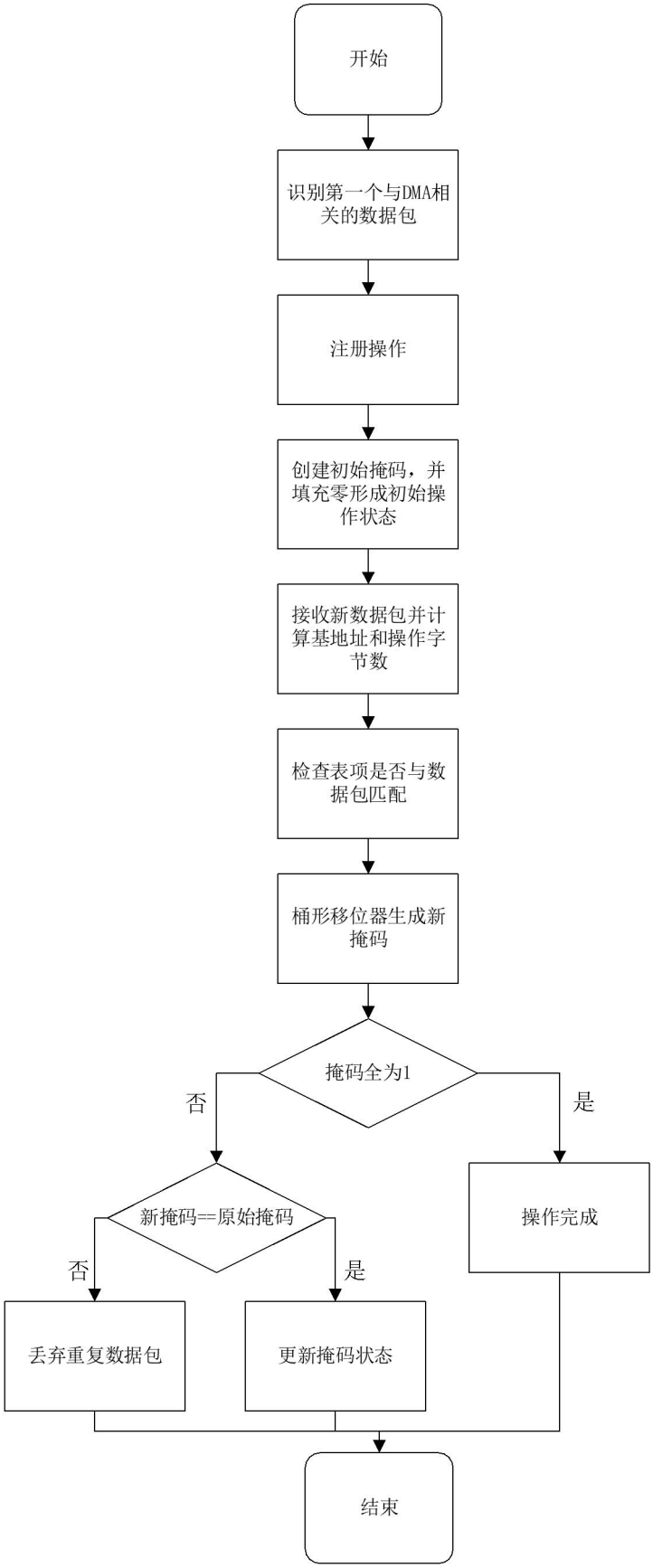
开始
识别第一个与DMA相关的数据包
注册操作
创建初始掩码，并填充零形成初始操作状态
接收新数据包并计算基地址和操作字节数
检查表项是否与数据包匹配
桶形移位器生成新掩码
掩码全为1
否
是
新掩码==原始掩码
操作完成
否
是
丢弃重复数据包
更新掩码状态
结束

图 4

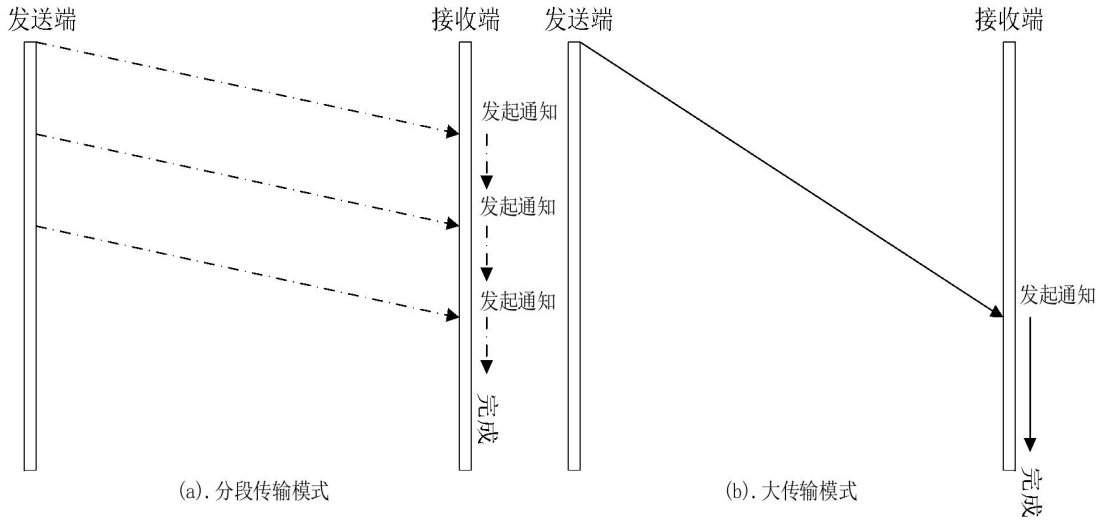

图 5

说 明 书 摘 要

本公开提供了基于接收注册表机制的数据传输优化方法及系统，涉及高性能计算与数据传输优化技术领域，包括在发送端将大型的 RDMA 传输分割成多个较小的传输，允许多个数据段并行处理和传输；在接收端引入一种接收注册表机制，记录和跟踪 RDMA 操作；通过使用特殊的状态标志和掩码机制，识别数据包是否属于同一 RDMA 操作，并在操作完成时提供本地和远程通知，确保了数据传输的准确性和及时性。对于尚未在接收注册表中注册的数据包，设计逃逸通道，以避免流水线阻塞。同时，引入了超时机制，用于丢弃那些重复或延迟到达的数据包，从而保证了数据传输的稳定性和可靠性。