

缓存管理方法和装置、控制程序及控制器

相关申请的交叉引用

本公开基于 2021 年 09 月 02 日提交的发明名称为“缓存管理方法和装置、控制程序及控制器”的中国专利申请 CN202111028812.3，并且要求该专利申请的优先权，通过引用将其所公开的内容全部并入本公开。

技术领域

本公开涉及通信技术领域，具体而言，涉及一种缓存管理方法和装置、控制程序及控制器。

背景技术

在以太网交换芯片的应用中，往往需要根据应用场景和成本等，选择不同型号的片外缓存单元，即缓存控制器，所以芯片的缓存管理单元（Memory Management Unit, 简称 MMU）不仅要能够满足基本的功能需求，且要具有良好的兼容性和可移植性，以便在不同的应用场景下，根据存储大小、速率、功耗和成本等因素连接不同的缓存控制器，而不需要进行多次开发，从而节省人力和成本。

随着处理器和存储控制器频率和带宽的不断提升，其各自性能也在不断提高，但缓存访问效率却往往成为系统性能的瓶颈问题，而缓存访问效率与 MMU 的实现方式有关。在以太网交换芯片中，MMU 的主要功能是负责报文（packet, PK）的数据、报文描述符（packet descriptor, PD）的写请求、写数据分发，并按照写请求保序释放；再将数据根据读请求保序读回，在这个过程中，通过片外地址管理、物理地址映射和拼包等技术尽可能最大化利用缓存带宽，以提高存储控制器的效率。当下主流的缓存控制器有 DDR3/DDR4/DDR5 (Double Data Rate, DDR 双倍速率)、HBM(High Bandwidth Memory) (高带宽存储器) 等，如何在同一种框架下兼容不同的控制器且能够保证其存储效率是待解决的主要问题。

针对上述如何在同一种框架下兼容不同的控制器的问题，目前尚未提出有效的解决方案。

发明内容

本公开实施例提供了一种缓存管理方法和装置、控制程序及控制器，以至少解决相关技术中如何在同一种框架下兼容不同的控制器的问题。

根据本公开的一个实施例，提供了一种缓存管理方法，包括：缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同。

根据本公开的另一个实施例，提供了一种缓存管理装置，包括：识别单元，设置为使缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；确认单元，设置为基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；计算单元，设置为基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控

制器类型选通的对外连接通道数不同。

根据本公开的又一个实施例，还提供了一种计算机可读存储控制程序，上述计算机可读存储控制程序中存储有计算机程序，其中，上述计算机程序被设置为运行时执行上述任一项方法实施例中的步骤。

根据本公开的又一个实施例，还提供了一种控制器，包括缓存器和处理器，上述控制器中存储有计算机程序，上述处理器被设置为运行上述计算机程序以执行上述任一项方法实施例中的步骤。

通过本公开，由于采用了缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同；因此，实现了同一种框架下不同的控制器的切换，可以解决同一种框架下兼容不同的控制器的问题，进而在同一框架下可兼容多种控制器，以及提高存储效率的效果。

附图说明

图 1 是根据本公开实施例的缓存管理方法的移动终端的硬件结构框图；

图 2 是根据本公开实施例的缓存管理方法的流程图；

图 3 是根据本公开实施例的缓存管理系统的架构示意图；

图 4 是根据本公开实施例的缓存管理方法中的 HBM 的 optionE 模式的示意图；

图 5 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图一；

图 6 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图二；

图 7 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图三；

图 8 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图四；

图 9 是根据本公开实施例的缓存管理方法中的地址映射的示意图一；

图 10 是根据本公开实施例的缓存管理方法中的地址映射的示意图二；

图 11 是根据本公开实施例的缓存管理方法中的地址映射的示意图三；

图 12 是根据本公开实施例的缓存管理方法的片外地址管理示意图；

图 13 是根据本公开实施例的缓存管理装置的结构示意图。

具体实施方式

下文中将参考附图并结合实施例来详细说明本公开的实施例。

需要说明的是，本公开的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。

本公开实施例中所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。以运行在移动终端上为例，图 1 是本公开实施例的缓存管理方法的移动终端的硬件结构框图。如图 1 所示，移动终端可以包括一个或多个（图 1 中仅示出一个）处理器 102（处理器 102 可以包括但不限于微处理器 MCU 或可编程逻辑器件 FPGA 等的处理装置）和用于存储数据的存储器 104，其中，上述移动终端还可以包括用于通信功能的传输设备 106 以及输入输出设备 108。本领域普通技术人员可以理解，图 1 所示的结构仅为示意，其并不对上述移动终端的结构造成限定。例如，移动终端还可包括比图 1 中所示更多或者更少的组件，或者

具有与图 1 所示不同的配置。

存储器 104 可用于存储计算机程序，例如，应用程序的软件程序以及模块，如本公开实施例中的缓存管理方法对应的计算机程序，处理器 102 通过运行存储在存储器 104 内的计算机程序，从而执行各种功能应用以及数据处理，即实现上述的方法。存储器 104 可包括高速随机存储器，还可包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器 104 可进一步包括相对于处理器 102 远程设置的存储器，这些远程存储器可以通过网络连接至移动终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

传输装置 106 用于经由一个网络接收或者发送数据。上述的网络具体实例可包括移动终端的通信供应商提供的无线网络。在一个实例中，传输装置 106 包括一个网络适配器(Network Interface Controller，简称为 NIC)，其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中，传输装置 106 可以为射频(Radio Frequency，简称为 RF)模块，其用于通过无线方式与互联网进行通讯。

图 2 是根据本公开实施例的缓存管理方法的流程图，如图 2 所示，该流程包括如下步骤：

步骤 S202，缓存管理单元 MMU 基于中央处理器 CPU 配置信息，识别外接的缓存控制器类型；

步骤 S204，基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；

步骤 S206，基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同。

通过本公开实施例，由于采用了缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同；因此，实现了同一种框架下不同的控制器的切换，可以解决同一种框架下兼容不同的控制器的的问题，进而同一框架下可兼容多种控制器，以及提高存储效率的效果。

在一个或多个实施例中，上述缓存管理方法还包括：上述 MMU 的地址映射子模块读取预设的地址映射关系；上述地址映射关系为将片内逻辑地址转换为能被缓存芯片接受的物理地址。

在一个或多个实施例中，上述缓存管理方法还包括：上述方法还包括：上述 MMU 的地址映射子模块读取预设的地址映射关系；上述地址映射关系为将片内逻辑地址转换为能被缓存芯片接受的物理地址；根据不同应用场景，通过 CPU 进行重配置，以获取重配置后与上述应用场景对应的地址映射关系。

在一个或多个实施例中，上述缓存管理方法还包括：根据片外缓存的需求和缓存控制器的结构属性，按照块单元 block 将 CPU 发送给上述 MMU 的数据包进行切分，其中，切分后的数据包与 block 地址相对应，一个数据包对应多个 block 地址，上述缓存控制器的类型不同

时地址管理的区间不同。

在一个或多个实施例中，上述缓存管理方法还包括：上述 MMU 接收到报文缓存管理单元 PMU 发送的片外数据，其中，上述片外数据包括报文描述符 PD 和报文 PK；提取上述 PD 的数据信息；将上述数据信息进行拼包得到拼包数据，并删除上述 PK 中的第一数据；这里，第一数据可以包括拼包数据中的无效数据。

对上述拼包数据按照有效数据位进行移位拼接，提取出第二数据；这里，第二数据可以包括拼包数据中的有效数据。将上述拼包数据中的上述 PD 的数据信息和提取出的上述第二数据再次进行拼包，得到目标拼包数据。上述目标拼包数据发送至片外缓存进行存储。

在一个或多个实施例中，上述缓存管理方法还包括：在上述 PD 的数据位长度加上提取出的上述第二数据的数据位长度小于或等于总线位宽时，同时输出上述 PD 和上述第二数据；

在上述 PD 的数据位长度加上提取出的上述第二数据的数据位长度大于总线位宽时，将上述 PD 和上述第二数据拆分为第一拆分数据和第二拆分数据；

输出上述 PD 和上述第一拆分数据；

将上述第二拆分数据进行补零操作，得到数据位长度等于上述总线位宽的补零处理数据，并输出上述补零处理数据。

在一个或多个实施例中，上述缓存管理方法还包括：报文缓存管理单元 PMU 下发写报文；

上述 PMU 存储上述写报文，发送写释放命令给流量缓存管理单元 TMMU；上述 TMMU 下发报文描述符至队列管理单元 QMU；

上述 QMU 通过上述 TMMU 下发写命令，并透传给上述 MMU，上述 QMU 完成存储上述写命令后向上述 TMMU 发写释放信号；

命令队列下发读报文命令，读取读报文数据并将上述读报文数据返回到上述 PMU；

上述 TMMU 下发读命令，并读取报文描述符数据。

基于上述实施例，在一应用实施例中，本公开提供的缓存管理装置中，MMU 位于片外存储控制器和片上控制模块之间，图 3 是根据本公开实施例的缓存管理系统的架构示意图，如图 3 所示，MMU 位于 PMU (Packet Memory Unit) 报文存储单元、TMMU (TM Memory Management Unit 内存管理单元)、CMD_FIFO (Command FIFO, 命令先进先出队列) 和 HBM (high Bandwidth Memory, 高带宽存储器)/DDR 之间，本公开实施例的缓存管理方法主要能实现功能如下：

- 1) 负责多队列报文 PK、报文描述符 PD 写请求和写数据的分发，写释放的保存；
- 2) 负责多队列 PK、PD 读请求的分发，读数据的保存；
- 3) 支持多类型的缓存控制器的模式切换；
- 4) 逻辑地址到片外物理地址的映射可重配置。

上述 PMU 存储上述写报文，发送写释放命令给流量缓存管理单元 TMMU；上述 TMMU 下发报文描述符至队列管理单元 QMU；上述 QMU 通过上述 TMMU 下发写命令，并透传给上述 MMU，

上述 QMU 完成存储上述写命令后向上述 TMMU 发写释放信号：

命令队列下发读报文命令，读取读报文数据并将上述读报文数据返回到上述 PMU；上述 TMMU 下发读命令，并读取报文描述符数据。在整个框架中，为了解决兼容性和提高片外访问带宽及效率，主要采用的技术有地址管理、PC 均衡、片外地址映射和拼包技术，兼容性贯穿整个缓存管理过程，从而兼顾所有支持控制器的功能和性能。

在一应用实施例，上述缓存管理方法包括：

第一步，多控制器切换。通过 CPU 配置，缓存管理模块识别外接的缓存控制器类型，地址管理子模块根据控制器类型对应的地址区域，通过查表的方式确认偏移地址，从而计算出该类型控制器的逻辑地址；配置不同的缓存控制器类型，选通的对外连接通道数不同。本公开根据片外缓存的需求，设计 16 个 Channel (通道) 的片外缓存接口，每个 Channel 均完整支持 AXI4 总线的写地址、写数据、写响应、读地址和读数据五个通道。在 MMU 中，分 HBM 和 DDR 模式，支持模式切换可配置，默认为 16 Channel 的 HBM。通过 CPU 配置，可切换到 DDR 模式下，用户根据自身的需求，可选择连接 DDR4 或 DDR5 等不同类型，只要连接的容量大小不小于地址管理的最大节点数对应的数据容量，连接的 HBM/DDR 空间均可被有效利用；

由于支持不同类型的缓存控制器，各类型的控制器的传输速率并不一定和 MMU 的系统时钟同步，为了适配不同的控制器和保证数据的线速传输，划分片上的部分缓存区域，以异步（先进先出队列）FIFO 的形式，先将数据缓存，同时通过自研逻辑的预读功能，将数据和命令预读出来进入 Ready 等待状态，当有效缓存控制器到来的时候，通过握手机制同周期将数据送出，这样可以保证最大程度利用片外控制器的带宽，且可靠地将数据流跨时钟处理。

第二步，配置地址映射。地址映射子模块根据配置的类型，读取系统预设的地址映射关系，此地址映射关系可根据应用场景，通过 CPU 进行重配置，以调整最佳的映射方式：

由于逻辑地址不能直接索引到 HBM/DDR 的地址引脚上，经过 MMU 的处理，将逻辑地址转换为能被缓存芯片接受的地址，称为物理地址。因为缓存 HBM/DDR 的多层次结构，地址映射的方式和片外读写带宽、存储速率及效率有很大关系。

本公开的地址映射，是将每个物理通道（数据总线位宽 128bit）分两个虚拟通道（Pseudo Channel，伪信道，简称 PC，数据总线位宽 64bit），两个虚拟通道共用一组地址和控制总线。

图 4 是根据本公开实施例的缓存管理方法中的 HBM 的 optionE 模式的示意图，在 HBM 的 OptionE 模式下如图 4 所示，每个虚拟通道对应一个控制器，控制器以缓存频率的一半运行。Psgnt 代表的虚拟通道（PS），是控制器内部仲裁信号，控制器根据 PS 的物理接口确定 Psgnt 的值。逻辑在处理的时候，只需把一个 HBM stack 的 8 个控制器当成 16 个控制器对待，一个控制对应一个物理 PS。

SID 是数据 8Hi 的特有地址，可以当做一个 bank（存储体）地址，8Hi 的 bank 数是 4Hi 的二倍，分别是 32 个 bank，16 个 bank。连续 4 个 ID 的 bank 属于一个存储体集合 bank_group。8Hi、4Hi 分别是 8 个 bank，4 个 bank。

laddr[N:0] 是 8Byte 地址（一个控制的总线位宽为 128bit，划分为两个 PS，每个 PS 的

位宽为 64bit), 由于 HBM 控制器的预取倍为 4, 一次存储 256bit, 会占用 4 个地址, 所以 laddr[1:0] 实际逻辑不会分配地址, 实际默认为 0, 控制器也不用。

MMU 送出的地址为 384B 地址 (实际地址为 2 的整数次幂送出), AXI 的地址为 1 字节, PS 通道对应的颗粒每个地址存 128bit 数据, 所以四者地址对应关系如下:

图 5 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图一, 使用 Samsung HBM2 4Hi4G 时, 每个 PS 的存储空间为 $4GB/16 = 2Gb$, 对应关系如图 5 所示, {A[31:A28], A[4:0]} 填充为 0:

图 6 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图二, 使用 Samsung HBM2 8Hi8G 时, 每个 PS 的存储空间为 $8GB/16 = 4Gb$, 对应关系如图 6 所示, {A[31:A29], A[4:0]} 填充为 0:

图 7 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图三, 使用 Samsung HBM2E 4Hi8G 时, 每个 PS 的存储空间为 $8GB/16 = 4Gb$, 对应关系如图 7 所示, {A[31:A29], A[4:0]} 填充为 0:

图 8 是根据本公开实施例的缓存管理方法中的地址对应关系的示意图四, 使用 Samsung HBM2E 8Hi16G 时, 每个 PS 的存储空间为 $8GB/16 = 4Gb$, 对应关系如图 8 所示, {A[31:A30], A[4:0]} 填充为 0:

图 9 是根据本公开实施例的缓存管理方法中的地址映射的示意图一, 要充分发挥 HBM 的带宽, 需均衡的使用 16 个 PS, 同时通道内增加 Bank 切换, 当前采用 4Hi4G 的颗粒, 逻辑地址和物理地址间的映射关系如图 9 所示。

图 10 是根据本公开实施例的缓存管理方法中的地址映射的示意图二, 外挂 DDR5 的时候, 可配置连接 3 个通道的 DDR5, 其逻辑地址和物理地址的映射关系如图 10 所示。

图 11 是根据本公开实施例的缓存管理方法中的地址映射的示意图三, 外挂 DDR4 的时候, 可配置连接 3 个通道的 DDR5, 其逻辑地址和物理地址的映射关系如图 11 所示。

第三步, 片外地址管理。根据片外缓存的需求, 和缓存控制器的结构特点, 片外缓存地址是以块 block (可配置的固定大小数据单元) 为单位进行管理, 称为虚拟地址 (也称作逻辑地址), 处理器发送给 MMU 的一个数据包需按照 block 进行切分, 以便与 block 地址进行对应, 这样处理后, 一个数据包有可能对应多个 block 数据, 即需要产生多个 block 地址。根据片外缓存控制器的类型不同, 地址管理的区间不同。

图 12 是根据本公开实施例的缓存管理方法的片外地址管理示意图, 如图 12 所示, chunk (中继端口) 地址为 128K, T[16:13] 为 16 个大链表 ID, T[12:10] 为每个大链表下面的子链表 ID, T[9:0] 为每个子链表内链表编号, B[3:0] 为每 chunk 下面 blk 的个数, C[2:0] 为每个 blk 下面的切片个数 (数据+1 代表切片个数)。链表 ID 申请采用 RR(Round-Robin (轮询)) 调度, 当一条流进来时会先 RR 申请选中大链表, 大链表内再 RR 选中一个子链表, 然后从子链表内申请一个链表。同一条流会优先耗尽一个 chunk 中的 blk, 不同流会重新申请 chunk。地址管理的 {T[9:0], B[3:0], T[12:10], T[16:13]} 可作为一个 counter, 此地址是连续变

化。当外挂 HBM 的时候，T[16:13] 个 16 个 channel 一一对应；当外挂 DDR 的时候，对应选通 3 个通道，使用 T[14:13] 作为 channel_ID，即选通 0 (4、8、12)、1 (5、9、13)、2 (6、10、14) 通道，3、7、11、15 通道可配置不使用。由于 T[16:13] 的低两 bit 的粒度更小，所以其变化更快，通过地址映射后的缓存访问效率会更好。根据设定的位宽大小，MMU 可管理 $2(4+3+10+4)=2M$ 个节点，总共可管理 $2M \times 3k$ (blk 大小可配，以 3k 为例) = 48G 的数据。

为了保证整体的性能和带宽，在地址管理模块需保证各伪信道 PC 间的均衡访问，避免出现短时间内个别 PC 频繁访问，而其余 PC 空闲的情况，如果出现某个 PC 返回较慢，反馈给 MMU 忙状态时候，可根据实时命令统计和单 PC 历史命令统计，减少一次调度，从数据流的整个过程分析，PC 之间依然是均衡访问。

第四步，小包拼接技术。将片上 PD 存储至片外，理论上因其包长的原因，会降低片外存储的带宽和效率，但因其存储数量较大，为了节省片上资源，将 PD 搬移至片外进行存储，此处采用拼包技术对其进行存储，拼包的目的是通过挤泡将片外需要拼包的 PK 和 PD 中的无效字节挤掉，然后将有效字节进行拼接，以此来提高片外缓存的利用率。

拼包的步骤是 MMU 收到 PMU 送过来的片外包 PD 和 PK，首先进行 PD 信息的提取和拼包，同时将 PK 中的无效字节挤掉，对小包进行移位拼接，提取出有效数据，然后将拼包后的 PD 信息和提取出有效数据的 PK 再次进行拼包，将结果发送至片外进行存储。

由于片外的总线位宽是 384B，除了单包（不需要拼包），不同种类的拼包情况对于每一拍输出的结果也不一样，小包拼接分两种情况：小包拼小包 ($PD_len + PK_len \leq 384B$)；小包拼大包 ($PD_len + PK_len > 384B$)。当第一种小包拼小包时，PD 的长度加上提取出的数据长度小于总线位宽，可以一拍输出；当第二种小包拼小包的时候，由于长度大于总线位宽，需要分两拍输出，第一拍输出高 384B，剩下的部分在第二拍尾部补零输出，这样会对线速产生影响，需尽量规避。

本公开的方案不仅能够兼容多种类型的缓存控制器，且能够提升片外读写带宽和访问效率。以下通过实测数据，分别举例说明外挂 HBM/DDR5/DDR4 三种类型控制器的情况下，带宽和效率提升前后的实测数据。

使用标准地址映射的测试外挂 HBM 的结果如表 1 所示。本公开的效率提升方法测试结果如表 2 所示，表 3 为外挂 DDR5 的测试数据，表 4 为外挂 DDR4 的测试数据，经过实测结果对比分析，各模式下的片外总带宽和存储效率均有提升。

表 1

用例编号	片外存储包长 (Byte)	总带宽 (Gbps)	效率
1	128	862	0.35
2	192	1352	0.55
3	256	1643	0.67
4	288	1607	0.65
5	352	1739	0.71
6	384	1769	0.72
7	768	1648	0.67

8	1152	1634	0.67
9	1536	1628	0.66
10	1568	1601	0.65

表 2

用例编号	片外存储包长 (Byte)	总带宽 (Gbps)	效率
1	128	1602	0.65
2	192	1899	0.77
3	256	2139	0.87
4	288	2063	0.84
5	352	2186	0.89
6	384	2217	0.90

表 2 (续)

用例编号	片外存储包长 (Byte)	总带宽 (Gbps)	效率
7	768	2169	0.88
8	1152	2166	0.88
9	1536	2133	0.86
10	1568	2028	0.83

表 3

用例编号	片外存储包长 (Byte)	总带宽 (Gbps)	效率
1	128	136.89	0.59
2	192	154.22	0.67
3	256	163.54	0.71
4	384	168.49	0.73
5	768	170.63	0.74
6	1152	172.45	0.75
7	1184	169.61	0.74
8	1536	173.55	0.75
9	1568	169.95	0.74
10	12288	166.54	0.72

表 4

用例编号	片外存储包长 (Byte)	总带宽 (Gbps)	效率
1	128	200.64	65.35
2	192	247.85	80.73
3	256	267.03	86.98
4	384	272.29	88.69
5	768	270.61	88.15
6	1152	243.78	79.41
7	1536	252.68	82.31
8	1568	232.25	75.65

9	6144	233.99	76.22
10	12288	220.05	71.68

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本公开的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质（如ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端设备（可以是手机，计算机，服务器，或者网络设备等等）执行本公开各个实施例上述的方法。

在本实施例中还提供了一种图形渲染的处理装置，该装置用于实现上述实施例及优选实施方式，已经进行过说明的不再赘述。如以下所使用的，术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现，但是硬件，或者软件和硬件的组合的实现也是可能并被构想的。

图 13 本公开实施例的缓存管理装置的结构框图，如图 13 所示，该装置包括：

识别单元 1302，设置为使缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；

确认单元 1304，设置为基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；

计算单元 1306，设置为基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同。

通过本公开，由于采用了缓存管理单元 MMU 基于 CPU 配置信息，识别外接的缓存控制器类型；基于地址管理子模块和上述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；基于上述偏移地址计算出上述缓存控制器类型的逻辑地址；其中，不同的上述缓存控制器类型选通的对外连接通道数不同；因此，实现了同一种框架下不同的控制器的切换，可以解决同一种框架下兼容不同的控制器的的问题，进而同一框架下可兼容多种控制器，以及提高存储效率的效果。

需要说明的是，上述各个模块是可以通过软件或硬件来实现的，对于后者，可以通过以下方式实现，但不限于此：上述模块均位于同一处理器中；或者，上述各个模块以任意组合的形式分别位于不同的处理器中。

本公开的实施例还提供了一种计算机可读存储控制程序，该计算机可读存储控制程序中存储有计算机程序，其中，该计算机程序被设置为运行时执行上述任一项方法实施例中的步骤。

在一个示例性实施例中，上述计算机可读存储控制程序可以包括但不限于：CPU 和存储控制器的驱动程序、FPGA 与 HBM/DDR3/DDR4/DDR5 连接的控制程序等。

本公开的实施例还提供了一种控制器，包括缓存器（缓存部分数据）和处理器，该控制器中存储有计算机程序，该控制器被设置为运行计算机程序以执行上述任一项方法实施例中的步骤。

在一个示例性实施例中，上述控制器还可以包括协议转换的传输设备，其中，该传输设备和上述控制器连接，实现与缓存控制器的连接。

本实施例中的具体示例可以参考上述实施例及示例性实施方式中所描述的示例，本实施例在此不再赘述。

显然，本领域的技术人员应该明白，上述的本公开的各模块或各步骤可以用通用的计算装置来实现，它们可以集中在单个的计算装置上，或者分布在多个计算装置所组成的网络上，它们可以用计算装置可执行的程序代码来实现，从而，可以将它们存储在存储装置中由计算装置来执行，并且在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤，或者将它们分别制作成各个集成电路模块，或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样，本公开不限制于任何特定的硬件和软件结合。

以上所述仅为本公开的优选实施例而已，并不用于限制本公开，对于本领域的技术人员来说，本公开可以有各种更改和变化。凡在本公开的原则之内，所作的任何修改、等同替换、改进等，均应包含在本公开的保护范围之内。

权 利 要 求 书

1、一种缓存管理方法，包括：

缓存管理单元 MMU 基于中央处理器 CPU 配置信息，识别外接的缓存控制器类型；

基于地址管理子模块和所述缓存控制器类型对应的地址区域，以查表的方式确认偏移地址；

基于所述偏移地址计算出所述缓存控制器类型的逻辑地址；其中，不同的所述缓存控制器类型选通的对外连接通道数不同。

2、根据权利要求 1 所述的方法，其中，所述方法还包括：

所述 MMU 的地址映射子模块读取预设的地址映射关系；所述地址映射关系为将片内逻辑地址转换为能被缓存芯片接受的物理地址。

3、根据权利要求 1 所述的方法，其中，所述方法还包括：

所述 MMU 的地址映射子模块读取预设的地址映射关系；所述地址映射关系为将片内逻辑地址转换为能被缓存芯片接受的物理地址；

根据不同应用场景，通过 CPU 进行重配置，以获取重配置后与所述应用场景对应的地址映射关系。

4、根据权利要求 1 所述的方法，其中，所述方法还包括：

根据片外缓存的需求和缓存控制器的结构属性，按照块单元 block 将 CPU 发送给所述 MMU 的数据包进行切分，其中，切分后的数据包与 block 地址相对应，一个数据包对应多个 block 地址，所述缓存控制器的类型不同时地址管理的区间不同。

5、根据权利要求 1 所述的方法，其中，所述方法还包括：

所述 MMU 接收到报文缓存管理单元 PMU 发送的片外数据，其中，所述片外数据包括报文描述符 PD 和报文 PK；

提取所述 PD 的数据信息；

将所述数据信息进行拼包得到拼包数据，并删除所述 PK 中的第一数据；

对所述拼包数据按照有效数据位进行移位拼接，提取出第二数据；

将所述拼包数据中的所述 PD 的数据信息和提取出的所述第二数据再次进行拼包，得到目标拼包数据

将所述目标拼包数据发送至片外缓存进行存储。

6、根据权利要求 5 所述的方法，其中，所述方法还包括：

在所述 PD 的数据位长度加上提取出的所述第二数据的数据位长度小于或等于总线位宽时，同时输出所述 PD 和所述第二数据；

在所述 PD 的数据位长度加上提取出的所述第二数据的数据位长度大于总线位宽时,将所述 PD 和所述第二数据拆分为第一拆分数据和第二拆分数据;

输出所述 PD 和所述第一拆分数据;

将所述第二拆分数据进行补零操作,得到数据位长度等于所述总线位宽的补零处理数据,并输出所述补零处理数据。

7、根据权利要求 1 至 6 中任一项所述的方法,其中,所述方法还包括:

报文缓存管理单元 PMU 下发写报文;

所述 PMU 存储所述写报文,发送写释放命令给流量缓存管理单元 TMMU;所述 TMMU 下发报文描述符至队列管理单元 QMU;

所述 QMU 通过所述 TMMU 下发写命令,并透传给所述 MMU,所述 QMU 完成存储所述写命令后向所述 TMMU 发写释放信号;

命令队列下发读报文命令,读取读报文数据并将所述读报文数据返回到所述 PMU;

所述 TMMU 下发读命令,并读取报文描述符数据。

8、一种缓存管理装置,包括:

识别单元,设置为使缓存管理单元 MMU 基于中央处理器 CPU 配置信息,识别外接的缓存控制器类型;

确认单元,设置为基于地址管理子模块和所述缓存控制器类型对应的地址区域,以查表的方式确认偏移地址;

计算单元,设置为基于所述偏移地址计算出所述缓存控制器类型的逻辑地址;其中,不同的所述缓存控制器类型选通的对外连接通道数不同。

9、一种计算机可读存储控制程序,所述计算机可读存储程序中存储有计算机程序,其中,所述计算机程序被设置为运行时执行所述权利要求 1 至 7 任一项中所述的方法。

10、一种控制器,包括缓存器和处理器,所述控制器中存储有计算机程序,所述处理器被设置为运行所述计算机程序以执行所述权利要求 1 至 7 任一项中所述的方法。