(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 116627494 A
(43)申请公布日 2023.08.22

(21)申请号 202210126258.0

(22)申请日 2022.02.10

(71)申请人 格兰菲智能科技有限公司
地址 200135 上海市浦东新区中国(上海)自由贸易试验区金科路2557号201室

(72)发明人 卞仁玉 张淮声 王渊峰

(74)专利代理机构 华进联合专利商标代理有限公司 44224
专利代理师 陈小娜

(51)Int.Cl.
G06F 9/30(2006.01)
G06F 9/38(2006.01)
G06T 1/20(2006.01)

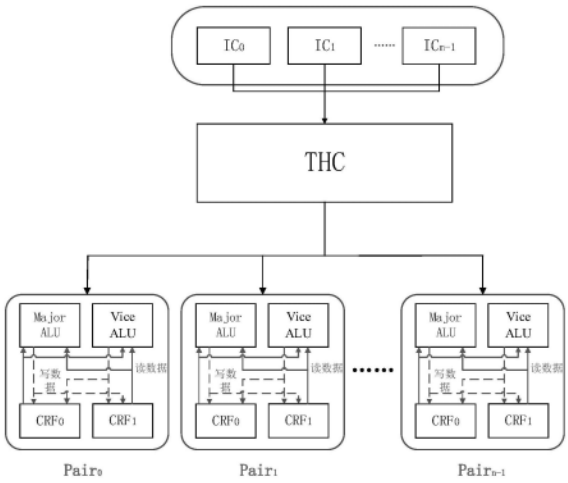
权利要求书3页 说明书10页 附图3页

(54)发明名称

处理器以及指令并行发射的处理方法

(57)摘要

本申请涉及一种处理器以及指令并行发射的处理方法。处理器包括线程控制模块以及线程控制模块对应的n个算术逻辑组，n≥2，线程控制模块，用于根据算术逻辑组的数量对线程控制模块所控制的线程集合进行分组，得到每个算术逻辑组分别对应的线程组；线程控制模块，从各个线程组中分别选取目标线程，获取目标线程对应的目标执行指令的目标指令地址，根据目标指令地址获取各个线程组对应的目标执行指令；线程控制模块，用于将线程组对应的目标执行指令发送至对应的算术逻辑组；算术逻辑组，用于执行接收到的目标执行指令，得到目标线程对应的指令执行结果。通过本申请的处理器，可以提高指令处理效率。



CN 116627494 A

1.一种处理器，其特征在于，所述处理器包括线程控制模块以及所述线程控制模块对应的n个算术逻辑组，所述n≥2，其中，

所述线程控制模块，用于根据算术逻辑组的数量对所述线程控制模块所控制的线程集合进行分组，得到每个所述算术逻辑组分别对应的线程组；

所述线程控制模块，从各个所述线程组中分别选取目标线程，获取所述目标线程对应的目标执行指令的目标指令地址，根据所述目标指令地址获取各个所述线程组对应的目标执行指令；

所述线程控制模块，用于将所述线程组对应的目标执行指令发送至对应的算术逻辑组；

所述算术逻辑组，用于执行接收到的所述目标执行指令，得到所述目标线程对应的指令执行结果。

2.根据权利要求1所述的处理器，其特征在于，所述将所述线程组对应的目标执行指令发送至对应的算术逻辑组包括：

将从上一个指令发送时钟周期后到当前指令发送时钟周期之间的多个时钟周期获取的目标执行指令进行存储，得到指令集合，上一个指令发送时钟周期与当前指令发送时钟周期之间至少间隔一个时钟周期；

当到达当前指令发送时钟周期，将所述指令集合中的所述线程组对应的目标执行指令发送至对应的算术逻辑组。

3.根据权利要求2所述的处理器，其特征在于，所述算术逻辑组中包括多个算术逻辑单元；所述将所述指令集合中的所述线程组对应的目标执行指令发送至对应的算术逻辑组包括：

当所述算术逻辑组对应的目标执行指令中存在相同类型的指令时，且所述算术逻辑组中的算术逻辑单元对应的执行指令类型互斥时，将所述算术逻辑组对应的相同类型的h-1个目标执行指令延迟到下一个指令发送周期进行发送，并将所述指令集合中剩余的目标执行指令发送至对应的算术逻辑组，h为相同类型的目标执行指令的数量。

4.根据权利要求3所述的处理器，其特征在于，所述算术逻辑组中包括第一算术逻辑单元以及第二算术逻辑单元，所述第一算术逻辑单元用于处理浮点类型的指令，所述第二算术逻辑单元用于处理整型类型的指令。

5.根据权利要求1所述的处理器，其特征在于，所述目标执行指令对应有多个操作数；所述执行接收到的所述目标执行指令，得到所述目标线程对应的指令执行结果包括：

在指令发送周期中循环执行所述目标执行指令p次，每次获取所述目标执行指令对应的不同的操作数进行处理，得到每次处理得到的所述目标线程对应的指令执行结果；所述p≥2；所述p≤k；所述k为指令发送周期对应的时钟周期数量。

6.根据权利要求5所述的处理器，其特征在于，所述算术逻辑组包括算术逻辑单元以及所述算术逻辑单元对应的寄存器；所述寄存器切分为至少p份子寄存器，所述在指令发送周期中循环执行所述目标执行指令p次，每次获取所述目标执行指令对应的不同的操作数进行处理，得到每次处理得到的所述目标线程对应的指令执行结果包括：

在每次执行所述目标执行指令时，向当前子寄存器发送操作数读取请求，以读取得到所述目标执行指令对应的当前操作数；所述当前子寄存器是从未读取过所述目标执行指令

的操作数的子寄存器中确定的；

对所述当前操作数进行处理，将处理得到的指令执行结果写入所述当前子寄存器。

7.一种指令并行发射的处理方法，其特征在于，所述指令并行发射的处理方法通过处理器执行，所述处理器包括线程控制模块以及所述线程控制模块对应的n个算术逻辑组，所述n≥2，所述方法包括：

根据算术逻辑组的数量对所述线程控制模块所控制的线程集合进行分组，得到每个所述算术逻辑组分别对应的线程组；

从各个所述线程组中分别选取目标线程，获取所述目标线程对应的目标执行指令的目标指令地址，根据所述目标指令地址获取各个所述线程组对应的目标执行指令；

将所述线程组对应的目标执行指令发送至对应的算术逻辑组；以通过所述算术逻辑组执行接收到的所述目标执行指令，得到所述目标线程对应的指令执行结果。

8.根据权利要求7所述的指令并行发射的处理方法，其特征在于，所述将所述线程组对应的目标执行指令发送至对应的算术逻辑组包括：

将从上一个指令发送时钟周期后到当前指令发送时钟周期之间的多个时钟周期获取的目标执行指令进行存储，得到指令集合，上一个指令发送时钟周期与当前指令发送时钟周期之间至少间隔一个时钟周期；

当到达当前指令发送时钟周期，将所述指令集合中的所述线程组对应的目标执行指令发送至对应的算术逻辑组。

9.根据权利要求8所述的指令并行发射的处理方法，其特征在于，所述算术逻辑组中包括多个算术逻辑单元，所述将所述指令集合中的所述线程组对应的目标执行指令发送至对应的算术逻辑组包括：

当所述算术逻辑组对应的目标执行指令中存在相同类型的指令时，且所述算术逻辑组中的算术逻辑单元对应的执行指令类型互斥时，将所述算术逻辑组对应的相同类型的h-1个目标执行指令延迟到下一个指令发送周期进行发送，并将所述指令集合中剩余的目标执行指令发送至对应的算术逻辑组，h为相同类型的目标执行指令的数量。

10.根据权利要求9所述的指令并行发射的处理方法，其特征在于，所述算术逻辑组中包括第一算术逻辑单元以及第二算术逻辑单元，所述第一算术逻辑单元用于处理浮点类型的指令，所述第二算术逻辑单元用于处理整型类型的指令。

11.根据权利要求7所述的指令并行发射的处理方法，其特征在于，所述目标执行指令对应有多个操作数；所述执行接收到的所述目标执行指令，得到所述目标线程对应的指令执行结果包括：

在指令发送周期中循环执行所述目标执行指令p次，每次获取所述目标执行指令对应的不同的操作数进行处理，得到每次处理得到的所述目标线程对应的指令执行结果；所述p≥2；所述p≤k；所述k为指令发送周期对应的时钟周期数量。

12.根据权利要求11所述的指令并行发射的处理方法，其特征在于，所述算术逻辑组包括算术逻辑单元以及所述算术逻辑单元对应的寄存器；所述寄存器切分为至少m份子寄存器，所述在指令发送周期中循环执行所述目标执行指令p次，每次获取所述目标执行指令对应的不同的操作数进行处理，得到每次处理得到的所述目标线程对应的指令执行结果包括：

在每次执行所述目标执行指令时，向当前子寄存器发送操作数读取请求，以读取得到所述目标执行指令对应的当前操作数；所述当前子寄存器是从未读取过所述目标执行指令的操作数的子寄存器中确定的；

对所述当前操作数进行处理，将处理得到的指令执行结果写入所述当前子寄存器。

处理器以及指令并行发射的处理方法

技术领域

[0001] 本申请涉及计算机技术领域,特别是涉及一种处理器以及指令并行发射的处理方法。

背景技术

[0002] 随着计算机设备的发展,计算机设备中的处理器的处理能力越来越重要。例如,在通用图形处理器中,计算单元是整个处理器中最核心的模块,线程控制模块是合理调度并控制计算单元有效工作的关键。在绘制平台上,可编程的各种着色器是图形渲染中最重要且最耗时的环节,这些着色器包括顶点着色器(VS,Vertex Shader)、像素着色器(PS,Pixel Shader)、外壳着色器(HS,Hull Shader)以及区域着色器(DS,Domain Shader)。在这些着色器中,除了纹理采样指令以及内存读写指令之外,占比最多的就是计算指令,因此计算指令的执行效率在通用图形处理器中显得格外重要。

[0003] 目前,同一个线程控制模块所管理的所有线程都是串行执行,如果同一个线程控制模块同时管理的线程数目过多,会导致处理器整体的并行性比较差,导致处理效率低。

发明内容

[0004] 基于此,有必要针对上述技术问题,提供一种处理器以及指令并行发射的处理方法。

[0005] 第一方面,本申请提供了一种处理器,其特征在于,所述处理器包括线程控制模块以及所述线程控制模块对应的n个算术逻辑组,所述n≥2,其中,所述线程控制模块,用于根据算术逻辑组的数量对所述线程控制模块所控制的线程集合进行分组,得到每个所述算术逻辑组分别对应的线程组;所述线程控制模块,从各个所述线程组中分别选取目标线程,获取所述目标线程对应的目标执行指令的目标指令地址,根据所述目标指令地址获取各个所述线程组对应的目标执行指令;所述线程控制模块,用于将所述线程组对应的目标执行指令发送至对应的算术逻辑组;所述算术逻辑组,用于执行接收到的所述目标执行指令,得到所述目标线程对应的指令执行结果。

[0006] 第二方面,本申请还提供了一种指令并行发射的处理方法,所述指令并行发射的处理方法通过处理器执行,所述处理器包括线程控制模块以及所述线程控制模块对应的n个算术逻辑组,所述n≥2,所述方法包括:根据算术逻辑组的数量对所述线程控制模块所控制的线程集合进行分组,得到每个所述算术逻辑组分别对应的线程组;从各个所述线程组中分别选取目标线程,获取所述目标线程对应的目标执行指令的目标指令地址,根据所述目标指令地址获取各个所述线程组对应的目标执行指令;将所述线程组对应的目标执行指令发送至对应的算术逻辑组;以通过所述算术逻辑组执行接收到的所述目标执行指令,得到所述目标线程对应的指令执行结果。

[0007] 第三方面,本申请还提供了一种计算机设备。所述计算机设备包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现上述指令处理

方法的步骤。

[0008] 第四方面，本申请还提供了一种计算机可读存储介质。所述计算机可读存储介质，其上存储有计算机程序，所述计算机程序被处理器执行时实现指令处理方法的步骤。

[0009] 第五方面，本申请还提供了一种计算机程序产品。所述计算机程序产品，包括计算机程序，该计算机程序被处理器执行时实现指令处理方法的步骤。

[0010] 上述处理器以及指令并行发射的处理方法，处理器包括线程控制模块以及线程控制模块对应的多个算术逻辑组，由于有多个算术逻辑组，且根据算术逻辑组的数量对线程集合进行分组，得到每个算术逻辑组分别对应的线程组。对于一个线程控制模块所控制的线程集合，可以分为多个线程组分别获取执行指令，并可以分别发送至该线程组所对应的算术逻辑组进行处理，因此线程控制模块所控制的线程集合中线程对应的目标执行指令可以并行处理，即线程控制模块可以同时发射多条指令，提高了线程的并行性以及指令的执行效率。

附图说明

[0011] 图1是本发明一实施例的处理器的示意图；

[0012] 图2是本发明一实施例的处理器的示意图；

[0013] 图3是本发明一实施例的处理器的示意图；

[0014] 图4是本发明一实施例的处理器的示意图；

[0015] 图5是本发明一实施例的指令并行发射处理方法的流程示意图。

具体实施方式

[0016] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

[0017] 图1是本发明一实施例的处理器的示意图。该处理器可以是通用图形处理器。如图1所示，处理器包括线程控制模块(THC，Thread Control)，每个THC可以连接n组算术逻辑组($pair_0$，$pair_1$，…，$pair_{n-1}$)，n≥2，其中n取值可以为2、4、8、16等2的幂次方。处理器还可以包括m个指令缓冲单元(IC，instruction cache)IC_0，IC_1，…，IC_{m-1}，m≥1。

[0018] 每个算术逻辑组中可以包括一个或者多个算术逻辑单元(ALU，algorithm logic unit)以及一个或者多个通用寄存器(CRF，common register file)。其中一个pair中ALU的数量可以与CRF的数量相同，例如：如果一个Pair中只有一个ALU，就配置一个CRF；如果一个Pair中有两个ALU，就配置两个CRF，便于两个ALU自由读写CRF。如图1所示，$pair_0$包括两个算术逻辑单元主算术逻辑单元(Major ALU)和副算术逻辑单元(Vice ALU)，两个通用寄存器CRF_0和CRF_1。一个算术逻辑组中可以包括一个或者多个算术逻辑单元。多个是指至少两个。

[0019] 举个例子，每个pair中可以只有一个ALU，所有指令都在这个ALU中执行，也可以为每个pair配置两个ALU：Major ALU(主算术逻辑单元)和Vice ALU(副算术逻辑单元)。Major ALU可以只执行浮点类型的指令，Vice ALU可以只执行整型类型的指令；或者Major ALU可以执行所有指令，Vice ALU可以只执行整型类型的指令。Major ALU和Vice ALU可以同时工

作,互不影响。ALU用于执行指令的具体操作,如乘法、加法、比较、与或非逻辑运算等等中的至少一个。可以理解,本申请实施例对每个算术逻辑组中ALU的数量不做限定,例如还可以是3个,具体可以根据实际确定。其中,图1的ALU与CRF之间的连接线中,虚线代表ALU在CRF中写数据,实线代表ALU从CRF中读数据。

[0020] 线程控制模块,用于根据算术逻辑组的数量对线程控制模块所控制的线程集合进行分组,使得每个算术逻辑组得到分别对应的线程组。

[0021] 其中,线程控制模块是管理多个线程同时运行的机制,每个THC可以同时管理a*n个线程,a的数量根据实际可以不同,例如可以为16。在a*n个线程被执行之前,线程控制模块会将a*n个线程分成n组,每组a个线程,n组线程分别对应n个pair,即一个pair可以对应一个线程组,所以每个pair中需要执行a个线程。

[0022] 当每个pair中的CRF为多个时,该算术逻辑组对应的线程组可以进一步切分为每个CRF分别对应的子线程组。例如,以b表示每个pair中CRF的数目,则可以将每个pair对应的线程组再分为b个子线程组,每个子线程组的线程个数为a/b个。例如以a为16,每个pair中包括2个CRF为例,则可以将16个线程进一步分成偶数组线程和奇数组线程,每组8个线程,偶数组线程可以始终读写CRF0,奇数组线程可以始终读写CRF1。一个线程组中可以包括多个线程。

[0023] 子线程组与pair中的CRF可以是一一对应关系,但每个ALU可以与所在pair中的每个CRF都有连接,根据指令的类型每个线程中的指令可能在对应的pair中的任意一个ALU中执行。例如一个线程有多条指令,可能一部分在Major ALU中执行,一部分在Vice ALU中执行。

[0024] 线程控制模块,用于从各个线程组中分别选取目标线程,获取目标线程对应的目标执行指令的目标指令地址,根据目标指令地址获取各个线程组对应的目标执行指令。

[0025] 其中,一个算术逻辑组可以对应一个线程组,当一个算术逻辑组包括多个CRF时,则该算术逻辑组对应的线程组可以进一步切分为每个CRF分别对应的子线程组。线程控制模块在每个指令发送周期中,从线程组中获取一个线程作为目标线程。一个指令发送周期可以包括多个时钟周期,为了减轻指令读取压力以及使得每个时钟周期处理器尽可能负载平衡,可以平均分配每个时钟周期所要读取的指令的线程组。例如,假设一个pair中有两个算术逻辑单元,且指令发送周期所对应的时钟周期为两个,即每两个时钟周期向算术逻辑组发送一次指令,则可以是设置在偶数时钟周期,从每个pair对应的偶数线程组中各选一个线程,作为目标线程,总共n个线程;线程控制模块在奇数时钟周期,从每个pair对应的奇数线程组中各选一个线程,作为目标线程,总共n个线程。当然也可以是在偶数时钟周期,从pair0～pair(n/2-1)中每个pair对应的偶数线程组以及奇数线程组各选一个线程,作为目标线程,总共n个线程;在偶数时钟周期,从pair(n/2-2)～pair(n-1)中每个pair对应的偶数线程组以及奇数线程组各选一个线程,作为目标线程,总共n个线程。

[0026] 线程控制模块在每个时钟周期,将读取得到的指令地址发送给指令缓冲单元,指令缓冲单元用于存储待执行的指令,指令缓冲单元收到线程控制模块发送过来的指令请求信号,指令缓冲单元根据指令请求信号中的指令地址从cache(缓存)或内存中取回指令,发送给线程控制模块。

[0027] 一个线程控制模块所对应的指令缓冲单元的数量可以根据其工作频率以及线程

控制模块的工作频率限定，例如，假设THC、ALU、CRF和IC工作在同频率下，THC的指令发送周期对应的时钟周期数量为2，每个pair中有两个ALU，则THC每两个时钟周期最多可以发射2n条指令，为了保证THC有充足的待发射指令，IC每个时钟周期需要提供至少n条指令给THC，也就意味着，每个THC需要连接n个IC，如果IC的工作频率是THC的两倍，就只要n/2个IC。IC的工作频率可以根据需求而定，IC的数量可以为一个或者多个。

[0028] 线程控制模块，用于将线程组对应的目标执行指令发送至对应的算术逻辑组。

[0029] 其中，指令发送周期对应的时钟周期的数量可以为多个，即一个指令发送周期可以包括多个时钟周期，指令发送时钟周期是指线程控制模块发送指令给算术逻辑组时所在的时钟周期。线程控制模块在每个指令发送周期向算术逻辑组发送一次目标执行指令。例如一个指令发送周期对应两个时钟周期，即线程控制模块每两个时钟周期发送一次指令。线程控制模块可以在每个指令发送周期中，从线程组中获取一个线程作为目标线程。因此每个算术逻辑组中的算术逻辑单元可以接收到一个目标执行指令。例如以b表示每个pair中ALU的数目，则线程控制模向每个算术逻辑组发送b个目标执行指令。例如，如果一个pair中有两个ALU，则线程控制模块每两个时钟周期，将取回来的2n条指令同时并行发射到n个pair中，每个pair接收2条指令。

[0030] 由于建立了算术逻辑组与线程组的对应关系，因此可以确定发送该目标执行指令的目标线程所在的线程组，将目标执行指令发送至该线程组对应的算术逻辑组。

[0031] 在一个实施例中，将线程组对应的目标执行指令发送至对应的算术逻辑组包括：将从上一个指令发送时钟周期后到当前指令发送时钟周期之间的多个时钟周期获取的目标执行指令进行存储，得到指令集合，上一个指令发送时钟周期与当前指令发送时钟周期之间至少间隔一个时钟周期；当到达当前指令发送时钟周期，将指令集合中的线程组对应的目标执行指令发送至算术逻辑组。

[0032] 其中，由于上一个指令发送时钟周期与当前指令发送时钟周期之间至少间隔一个时钟周期，因此一个指令发送周期可以包括多个时钟周期。如上所述，为了减轻指令读取压力以及尽量保证每个时钟周期处理器的负载平衡，可以平均分配每个时钟周期所要读取指令的线程组。线程控制模块存储在每个时钟周期读取到的指令，当到达当前指令发送时钟周期时，再将指令集合中的线程组对应的目标执行指令发送至算术逻辑组。例如，假设每3个时钟周期发送一次指令，即指令发送周期为3个时钟周期，在第4个时钟周期，线程控制模块发送第1个时钟周期、第2个时钟周期以及第3个时钟周期所读取到的指令。线程控制模块将在第4个时钟周期读取的指令进行存储，将在第5个时钟周期读取的指令进行存储，读取到第6个时钟周期读取到指令后，在第7个时钟周期发送第4个时钟周期、第5个时钟周期以及第6个时钟周期所读取到的指令。由于可以将需要读取的指令分配到多个时钟周期进行读取，因此可以减轻指令读取压力。

[0033] 在一个实施例中，一个算术逻辑组中可以包括多个算术逻辑单元以及与算术逻辑单元的数量相同的CRF，从每个CRF对应的线程组中选取目标线程；将指令集合中的线程组对应的目标执行指令发送至对应的算术逻辑组包括：当算术逻辑组对应的目标执行指令中存在相同类型的指令时，且算术逻辑组中的算术逻辑单元对应的执行指令类型互斥时，将算术逻辑组对应的相同类型的h-1个目标执行指令延迟到下一个指令发送周期进行发送，并将指令集合中剩余的目标执行指令发送至对应的算术逻辑组，h为相同类型的目标执行

指令的数量。

[0034] 在每个指令发送周期中,从线程组中选取各个CRF分别对应的目标线程可以是指:每个指令发送周期中,从每个CRF对应的线程组中选取一个目标线程,从而每个算术逻辑单元在每个指令发送周期可以对应有需要执行的目标执行指令。

[0035] 当一个算术逻辑组中有多个算术逻辑单元时,则若算术逻辑组中的算术逻辑单元对应的执行指令类型互斥,则说明该算术逻辑组中并不能同时执行相同类型的两个目标执行指令,因此在该指令发送周期,对于相同类型的指令,只发送其中一个指令至算术逻辑组,剩下的h-1个目标执行指令留到下一个指令发送周期再发。可以理解,在下一个指令发送周期,则会相应减少要读取的指令的数量。例如,若有h-1个指令需要延迟到下一个指令发送周期,则下一个指令周期,对于这h-1个指令所对应的线程组,则不再取其对应的指令。

[0036] 在一个实施例中,算术逻辑组中包括第一算术逻辑单元以及第二算术逻辑单元,第一算术逻辑单元用于处理浮点类型的指令,第二算术逻辑单元用于处理整型类型的指令。

[0037] 具体地,在人工智能训练和推理的应用场景中,会存在大量计算内存地址的整型计算,同时也会存在大量涉及浮点运算的卷积算法,因此将每个pair里的ALU分隔成FP ALU和INT ALU,FP ALU和INT ALU可以独立并行执行指令,但FP ALU只能处理浮点(FP,Floating Point)类型的指令,INT ALU只能处理整型(IN,Integer)类型的指令。从而使得在人工智能训练和推理的应用场景中,指令的执行效率更高。

[0038] 算术逻辑组,用于执行接收到的目标执行指令,得到目标线程对应的指令执行结果。

[0039] 算术逻辑组包括算术逻辑单元以及寄存器,算术逻辑单元接收到目标执行指令,向对应的寄存器发送读取目标执行指令的操作数的请求,寄存器向算术逻辑单元返回操作数。以图1为例,每个pair里的Major ALU和Vice ALU分别收到目标执行指令,可以根据指令所对应的线程组确定读写的CRF,例如假设Major ALU所接收到的目标执行指令来自于CRF_0对应的子线程组,Vice ALU所接收到的目标执行指令来自于CRF_1对应的子线程组,则Major ALU给CRF0发送读请求,Vice ALU给CRF1发送读请求,CRF_0和CRF_1收到读请求之后,CRF_0会将操作数发送给Major ALU,CRF_1会将操作数发送给Vice ALU。Major ALU和Vice ALU收到寄存器返回的操作数,根据指令信息,执行具体的操作,如乘法、加法或者比较等中的至少一个。Major ALU和Vice ALU执行完指令,Major ALU将结果写回到CRF_0,Vice ALU将结果写回到CRF_1中。

[0040] 在一个实施例中,目标执行指令对应有多个操作数;执行接收到的目标执行指令,得到目标线程对应的指令执行结果包括:在指令发送周期中循环执行目标执行指令p次,每次获取目标执行指令对应的不同的操作数进行处理,得到每次处理得到的目标线程对应的指令执行结果;$p \geq 2$;$p \leq k$;k为指令发送周期对应的时钟周期数量。

[0041] 目标执行指令可以是SIMD方式(Single Instruction Multiple Data,单指令多数据),每次ALU收到一条指令,可以将指令循环执行多次,但是每次所获取的操作数不同,例如每个ALU中有16组(不局限于16,也可以是32等其他数字)乘法、加法或者比较等操作处理的单元,可以同时处理16组数据,从而实现单指令多数据的执行方式。

[0042] 例如,p可以与k相同,即可以根据指令发送周期对应的时钟周期的数量,将目标执

行指令所对应的操作数切分为k个集合，每个时钟周期获取一个集合的操作数进行操作。由于指令发送周期对应的时钟周期数量有多个，循环执行多次目标执行指令，即在每个时钟周期执行一次指令，每次对部分操作数进行操作，相对于在一个时钟周期将全部操作数操作完毕，能够减少处理器的处理压力。

[0043] 在一个实施例中，算术逻辑组包括算术逻辑单元以及算术逻辑单元对应的寄存器；寄存器切分为至少p份子寄存器，在指令发送周期中循环执行目标执行指令p次，每次获取目标执行指令对应的不同的操作数进行处理，得到每次处理得到的目标线程对应的指令执行结果包括：在每次执行目标执行指令时，向当前子寄存器发送操作数读取请求，以读取得到目标执行指令对应的当前操作数；当前子寄存器是从未读取过目标执行指令的操作数的子寄存器中确定的；对当前操作数进行处理，将处理得到的指令执行结果写入当前子寄存器。

[0044] 其中，一份子寄存器可以称为一个bank，寄存器切分的数量可以与指令发送周期对应的时钟周期的数量相同。在每次执行指令时，是从没有读取过目标执行指令的操作数的子寄存器中确定的当前子寄存器中读取当前操作数，在处理得到结果时，再将处理得到的指令执行结果写入当前子寄存器。切分为多个子寄存器一方面能够减轻寄存器读取一个目标执行指令的操作数的读取压力，有利于解决CRF的读写冲突。另一方面在前后指令存在依赖的情景中，可以减少因依赖关系导致的指令等待时间。例如，假设指令发送周期对应2个时钟周期，则一个CRF会分low和high两个bank去管理，因此同一个时钟周期，CRF的low和high两个bank可以同时被读写。ALU收到一条指令，循环执行两次，第一次读写CRF的low bank，第二次读写CRF的high bank，因此在每个时钟周期，ALU占用CRF的一个bank，另一个bank可以留给处理器的其他模块，譬如纹理采样模块、内存读写模块、像素采样模块等。而在第二个指令发送周期，执行下一个执行指令时，由于下一个执行指令也是分为两个bank分别执行，则若目标执行指令与下一个执行指令之间的数据存在依赖关系，例如目标执行指令得到的结果，会用于下一个执行指令，则在下一个执行指令执行时第一次读写CRF的low bank，与执行目标执行指令时第一次读写CRF的low bank之间，相隔执行目标执行指令时第二次读写CRF的high bank这一次指令执行的过程，从而提高了下一个执行指令执行时第一次读写CRF的low bank时，可以利用执行目标执行指令时第一次读写CRF的low bank得到的处理结果的概率。其中，low和high用于区分子寄存器的读写顺序，是一个相对概念，例如第一次读取时，可以读取low bank中的数据，第二次读取时，可以读取high bank中的数据。

[0045] 由于一个THC可以管理多个pair单元，算术逻辑单元之间独立运行。也就意味着，每个THC管理的线程数目也相应的变多，这样能实现更多线程之间的同步，例如可以在通用图形处理器中实现多指令并行发射(Multi-Instruction Transmission)。而且由于每个pair能容纳的线程数目越多，同一个线程的先后指令可能存在一些依赖关系，这样就会引入一些等待时间，但如果每个pair能容纳的线程数目比较多，可以通过插入其他线程的时间来隐藏这些等待时间。

[0046] 上述处理器包括线程控制模块以及线程控制模块对应的多个算术逻辑组，由于有多个算术逻辑组，且根据算术逻辑组的数量对线程集合进行分组，得到每个算术逻辑组分别对应的线程组。对于一个线程控制模块所控制的线程集合，可以分为多个线程组分别获

取执行指令，并可以分别发送至该线程组所对应的算术逻辑组进行处理，因此线程控制模块所控制的线程集合中线程对应的目标执行指令可以并行处理，即线程控制模块可以同时发射多条指令，提高了线程的并行性以及指令的执行效率。

[0047] 以下以每个THC都连接到4个pair上，THC、ALU和CRF的工作频率相同，但IC的工作频率是THC的两倍、每两个时钟周期发送一次指令为例，列举处理器处理指令的三个实施例，可以理解，THC并不仅限于连接到4个pair上，也可以是3个或者更多时钟周期发送一次指令。

[0048] 如图2所示，为一个实施例中处理器的示意图。每个pair只有一个ALU和一个CRF，一个算术逻辑组的所有计算指令在这一个ALU中执行，因此每个ALU在同一个时钟周期，接受一条指令。一个THC连接到4个pair，因此每个THC在同一个时钟周期可以并行发射4条指令。THC每两个时钟周期发射一次指令，因此IC每个时钟周期需返回2条指令给THC，THC连接一个IC，因此IC的工作频率为THC工作频率的2倍。ALU、CRF和THC的工作频率相同。

[0049] 每个pair中有一个CRF，这个CRF会分low和high两个bank去管理，因此同一个时钟周期，CRF的low和high两个bank可以同时被读写。ALU收到一条指令，循环执行两次，第一次读写CRF的low bank，第二次读写CRF的high bank，因此在每个时钟周期，ALU占用CRF的一个bank，另一个bank可以留给处理器的其他模块，譬如纹理采样模块、内存读写模块或者像素采样模块等。

[0050] 假设每个THC最多可以同时管理32个线程，在32个线程被执行之前，THC会将32个线程分成4组，每组8个线程，每组固定对应一个pair，每组的线程也只能被发往同一个pair。

[0051] 在偶数时钟周期上，THC会从pair0和pair1所对应的那两组线程中各选择一个线程的一条指令地址发送给IC；在奇数时钟周期上，THC会从pair2和pair3所对应的那两组线程中各选择一条指令地址发送给IC。因为IC的工作频率是THC的两倍，所以IC每个时钟可以返回两条指令信息给THC。在THC收到IC发回的4条指令之后，在需要发送指令的时钟周期，THC可以并行将这4条指令分别发射给pair0、pair1、pair2和pair3。每个pair里的ALU每两个时钟周期收到一条指令，根据指令信息，ALU在第一个时钟周期向CRF的low bank发送读请求，在第二个时钟周期向CRF的high bank发送读请求。CRF送到读请求之后，将操作数送回给ALU。每个pair的ALU在拿到CRF返回的操作数后，根据指令信息，依次执行具体的操作，如乘法、加法或者比较等等。这4条指令在各自pair中独立读写CRF，独立并行执行，从而实现了THC并行发射多指令的目的。每个pair的ALU在若干个时钟周期后会将计算结果写回给CRF的low bank，在若干个时钟周期后会将计算结果写回给CRF的high bank。

[0052] 在人工智能训练和推理的应用场景中，会存在大量计算内存地址的整型计算，同时也会存在大量涉及浮点运算的卷积算法，因此，在多指令并行发射的实例二中，将每个pair里的ALU分隔成FP ALU和INT ALU，FP ALU和INT ALU可以独立并行执行指令，但FP ALU只能处理浮点类型的指令，INT ALU只能处理整型类型的指令。每个pair里的每个CRF也被分成了CRF_0和CRF_1两份，类似于FP ALU和INT ALU，CRF_0和CRF_1也是分开管理。

[0053] 如图3所示，为一个在人工智能训练和推理的应用场景中处理器的示意图，每个THC连接4个pair，连接2个IC。THC、ALU和CRF的工作频率相同，IC的工作频率依然为THC工作频率的两倍。每个THC可以同时管理64个线程，64个线程被分成4组，分别对应4个pair，所以

每个pair中需要执行32个线程。将32个线程进一步分成偶数组和奇数组，每组有16个线程，偶数组线程始终读写CRF_0，奇数组线程始终读写CRF_1。其中，图3的ALU与CRF之间的连接线中，虚线代表ALU在CRF中写数据，实线代表ALU从CRF中读数据。

[0054] THC在偶数时钟周期，从pair0对应的偶数线程组和奇数线程组中各选择一个线程，同理从pair1对应的偶数线程组和奇数线程组中各选择一个线程，总共四个线程。

[0055] THC将来自于偶数线程组的两条指令地址发往IC0，将来自于奇数线程组的两条指令地址发往IC1，由于IC的工作频率是THC的两倍，所以在同一个时钟周期，IC可以将这4条指令送回THC。类似于偶数时钟周期，THC在奇数时钟周期会从pair2和pair3中以同样的方式取回另4条指令。THC每两个时钟周期可以从IC收到8条指令，8条指令中每两条指令属于同一个pair，并且属于同一个pair的两条指令，必定一个是偶数线程组的，一个是奇数线程组的。THC在同一个时钟周期将这8条指令同时送往4个pair中，每个pair收到两条指令，其中一条会被发往FP ALU，另一条会被发往INT ALU。假设发往同一个pair的两条指令是相同类型的指令，此时THC只能往这个pair发送其中一条指令，另一条指令等到下一次指令发射周期再发射。

[0056] 每个pair里的FP ALU和INT ALU每两个时钟周期收到一条指令，假设发往FP ALU的指令来自偶数线程组，则根据指令信息，FP ALU在第一个时钟周期向CRF0的low bank发送读请求，在第二个时钟周期向CRF0的high bank发送读请求。CRF0收到读请求之后，会依次将操作数送回给FP ALU。假设发往INT ALU的指令来自奇数线程组，INT ALU在第一个时钟周期向CRF1的low bank发送读请求，在第二个时钟周期向CRF1的high bank发送读请求。CRF1收到读请求之后，会依次将操作数送回给INT ALU。每个pair的FP ALU和INT ALU在收到CRF0和CRF1返回的操作数，根据指令信息，依次执行具体的操作，如乘法、加法或者比较等等。这8条指令在各自pair的FP ALU和INT ALU中独立读写CRF，独立并行执行，从而实现了THC并行发射多指令的目的。每个pair的FP ALU和INT ALU在若干个时钟周期后会将计算结果写回给其对应的寄存器中的low bank，在若干个时钟周期后会将计算结果写回给其对应的寄存器中的的high bank。

[0057] 图3与图2的处理器之间相比，由于一个pair包括2个算术逻辑单元，因此同样4个pair，THC在同一个时钟周期并行发射的指令数目从4条最多可变成8条，实现了更多的并行性，同时也提高了指令执行效率。

[0058] 图4所示，为一个实施例中处理器的示意图，图4和图3相似，每个pair中有两个ALU，但这两个ALU不再是互斥关系，FP ALU只能运行浮点类型指令，但Major ALU既可以运行浮点类型指令，也可以运行整型类型指令。每个pair依然可以同时执行两条指令，但两条指令中至少有一条浮点类型的指令，图3的其他运行机制同图2相同，这里不再赘述。其中，图4的ALU与CRF之间的连接线中，虚线代表ALU在CRF中写数据，实线代表ALU从CRF中读数据。图2的处理器可以适应整型类型指令和浮点类型指令在线程中同时出现概率比较大的情况，但在D3D(Direct3D)的绘制平台上，浮点型的指令篇幅更大，图3的处理器会比图2的处理器在指令运行上更高效一些。图2与图3中每个pair中ALU的多样化，可以使处理器使用的场景更加灵活，在相同数目的pair中，实现更多的并行性，提高指令执行效率

[0059] 在一个实施例中，如图5所示，提供了一种指令并行发射的处理方法，以该方法应用于计算机设备的处理器为例进行说明，处理器包括线程控制模块以及线程控制模块对应

的n个算术逻辑组，n≥2，方法包括以下步骤：

[0060] 步骤S502，根据算术逻辑组的数量对线程控制模块所控制的线程集合进行分组，得到每个算术逻辑组分别对应的线程组。

[0061] 步骤S504，从各个线程组中分别选取目标线程，获取目标线程对应的目标执行指令的目标指令地址，根据目标指令地址获取各个线程组对应的目标执行指令。

[0062] 步骤S506，将线程组对应的目标执行指令发送至对应的算术逻辑组；以通过算术逻辑组执行接收到的目标执行指令，得到目标线程对应的指令执行结果。

[0063] 在一个实施例中，将线程组对应的目标执行指令发送至对应的算术逻辑组包括：将从上一个指令发送时钟周期后到当前指令发送时钟周期之间的多个时钟周期获取的目标执行指令进行存储，得到指令集合，上一个指令发送时钟周期与当前指令发送时钟周期之间至少间隔一个时钟周期；当到达当前指令发送时钟周期，将指令集合中的线程组对应的目标执行指令发送至对应的算术逻辑组。

[0064] 在一个实施例中，算术逻辑组中包括多个算术逻辑单元，在每个指令发送周期中，从线程组中选取各个寄存器分别对应的目标线程；将指令集合中的线程组对应的目标执行指令发送至对应的算术逻辑组包括：当算术逻辑组对应的目标执行指令中存在相同类型的指令时，且算术逻辑组中的算术逻辑单元对应的执行指令类型互斥时，将算术逻辑组对应的相同类型的h-1目标执行指令的延迟到下一个指令发送周期进行发送，并将指令集合中剩余的目标执行指令发送至对应的算术逻辑组，h为相同类型的目标执行指令的数量。

[0065] 在一个实施例中，算术逻辑组中包括第一算术逻辑单元以及第二算术逻辑单元，第一算术逻辑单元用于处理浮点类型的指令，第二算术逻辑单元用于处理整型类型的指令。

[0066] 在一个实施例中，目标执行指令对应有多个操作数；执行接收到的目标执行指令，得到目标线程对应的指令执行结果包括：在指令发送周期中循环执行目标执行指令p次，每次获取目标执行指令对应的不同的操作数进行处理，得到每次处理得到的目标线程对应的指令执行结果；p≥2；p≤k；k为指令发送周期对应的时钟周期数量。

[0067] 在一个实施例中，算术逻辑组包括算术逻辑单元以及算术逻辑单元对应的寄存器；寄存器切分为至少p份子寄存器，循环执行目标执行指令p次，每次获取目标执行指令对应的不同的操作数进行处理，得到每次处理得到的目标线程对应的指令执行结果包括：在每次执行目标执行指令时，向当前子寄存器发送操作数读取请求，以读取得到目标执行指令对应的当前操作数；当前子寄存器是从未读取过目标执行指令的操作数的子寄存器中确定的；对当前操作数进行处理，将处理得到的指令执行结果写入当前子寄存器。

[0068] 应该理解的是，虽然如上所述的各实施例所涉及的流程图中的各个步骤按照箭头的指示依次显示，但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明，这些步骤的执行并没有严格的顺序限制，这些步骤可以以其它的顺序执行。而且，如上所述的各实施例所涉及的流程图中的至少一部分步骤可以包括多个步骤或者多个阶段，这些步骤或者阶段并不必然是在同一时刻执行完成，而是可以在不同的时刻执行，这些步骤或者阶段的执行顺序也不必然是依次进行，而是可以与其它步骤或者其它步骤中的步骤或者阶段的至少一部分轮流或者交替地执行。

[0069] 故所提供的一个指令并行发射的处理方法实施例中的具体描述可以参见上文中

对于处理器的描述，在此不再赘述。

[0070] 以上实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

[0071] 以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本申请专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请的保护范围应以所附权利要求为准。

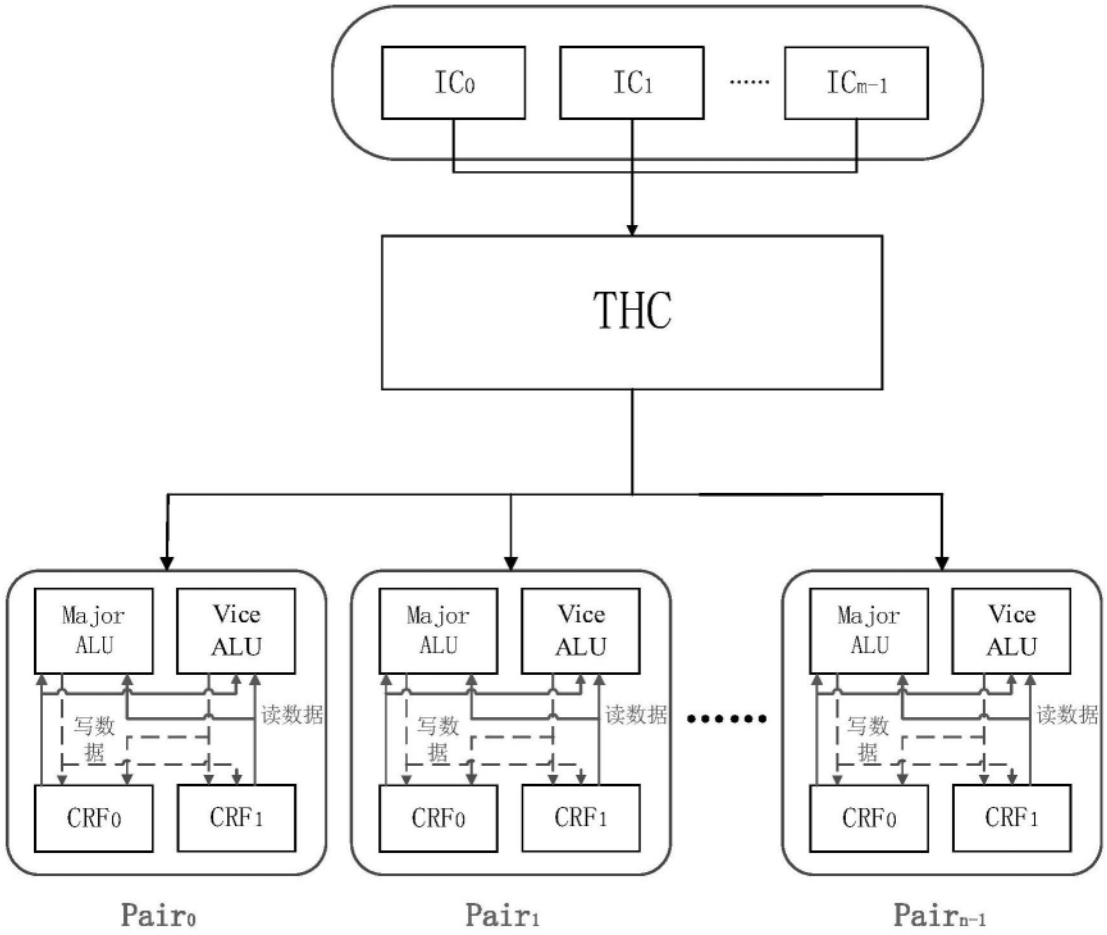

IC0
IC1
……
ICm-1
THC
Major ALU
Vice ALU
写数据
读数据
CRF0
CRF1
Pair0
Pair1
Pairn-1

图1

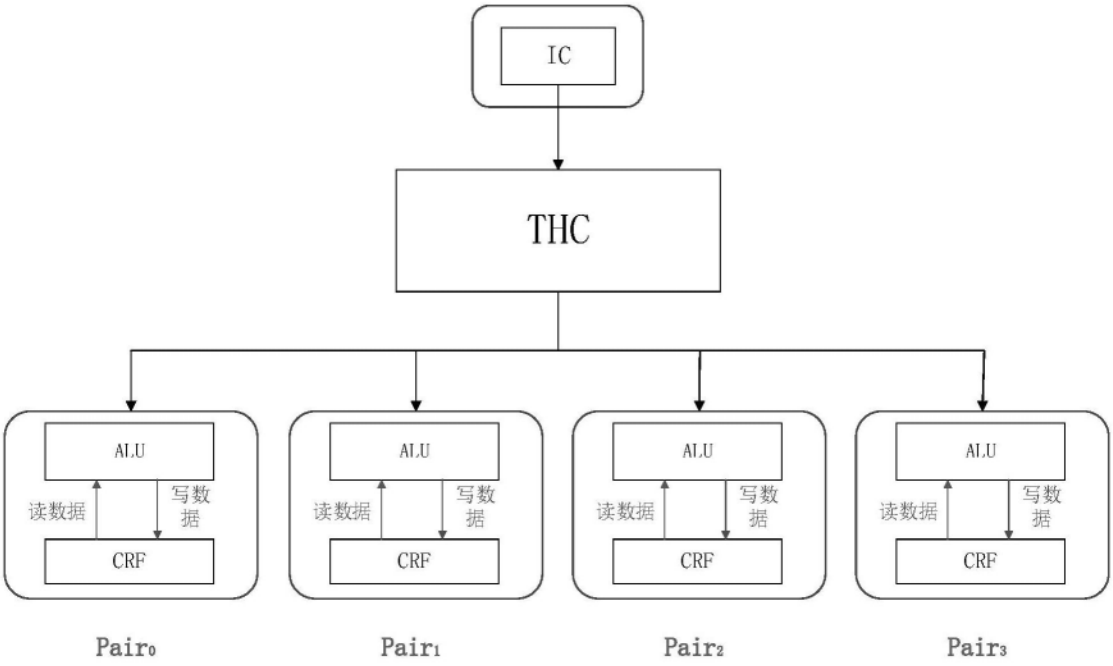

IC
THC
ALU
读数据
写数据
CRF
Pair0
Pair1
Pair2
Pair3

图2

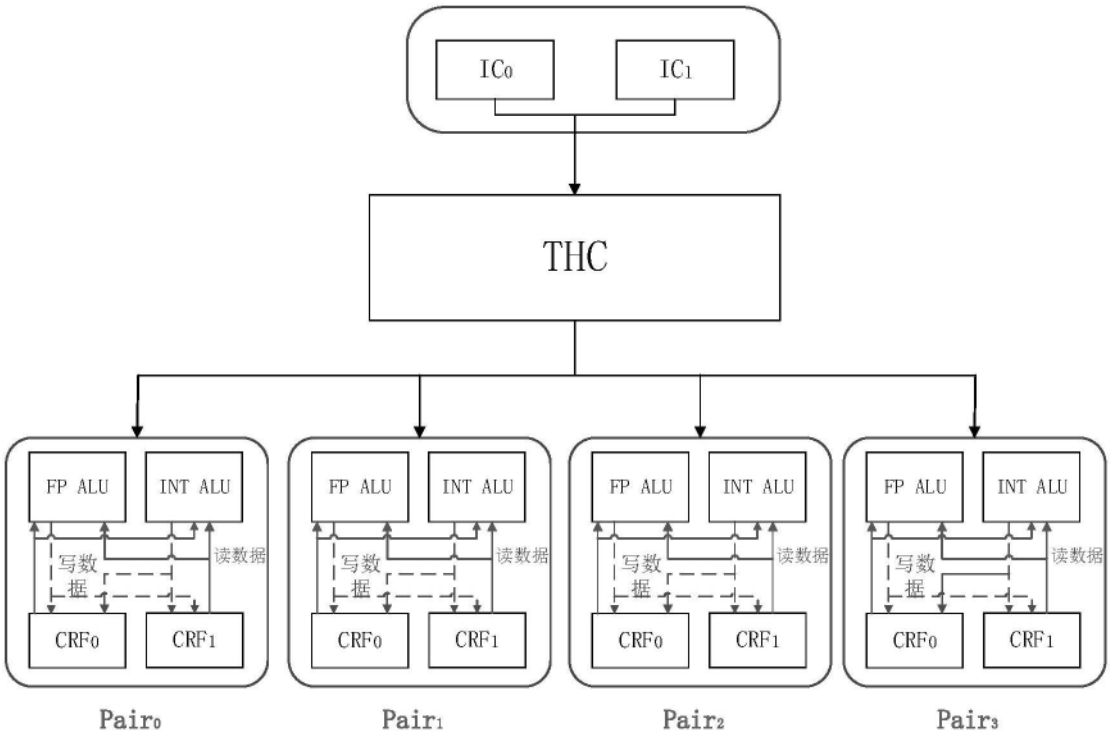

IC0
IC1
THC
FP ALU
INT ALU
写数据
读数据
CRF0
CRF1
Pair0
Pair1
Pair2
Pair3

图3

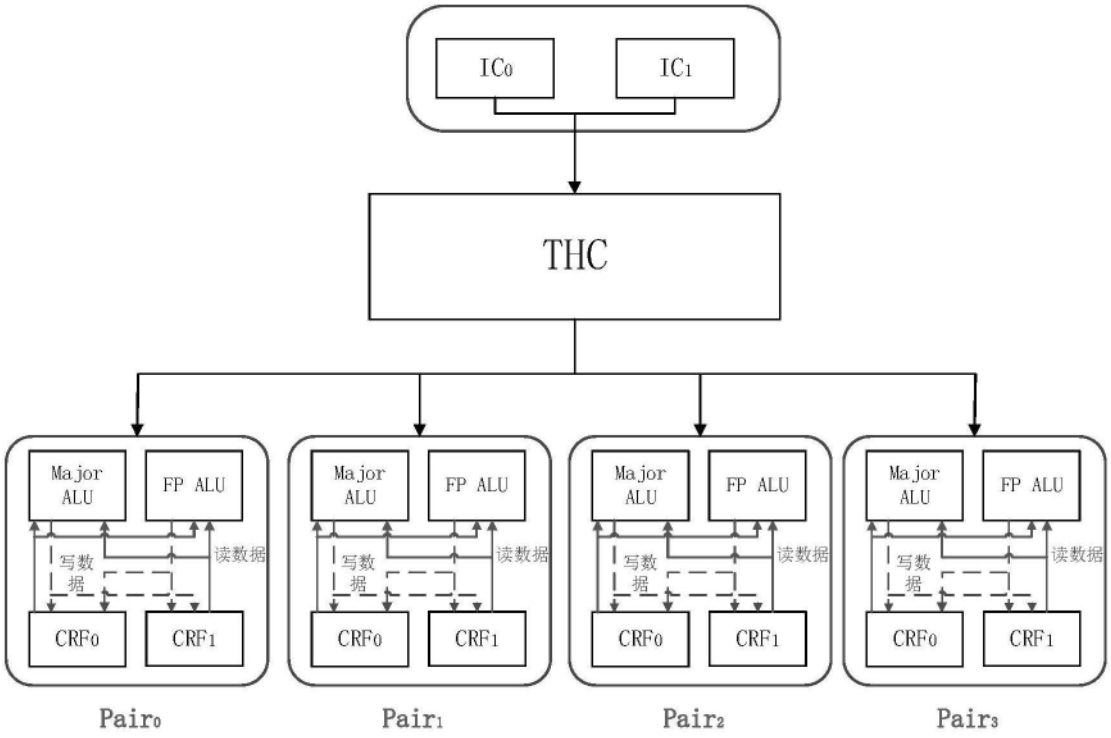

IC0
IC1
THC
Major ALU
FP ALU
写数据
读数据
CRF0
CRF1
Pair0
Pair1
Pair2
Pair3

图4

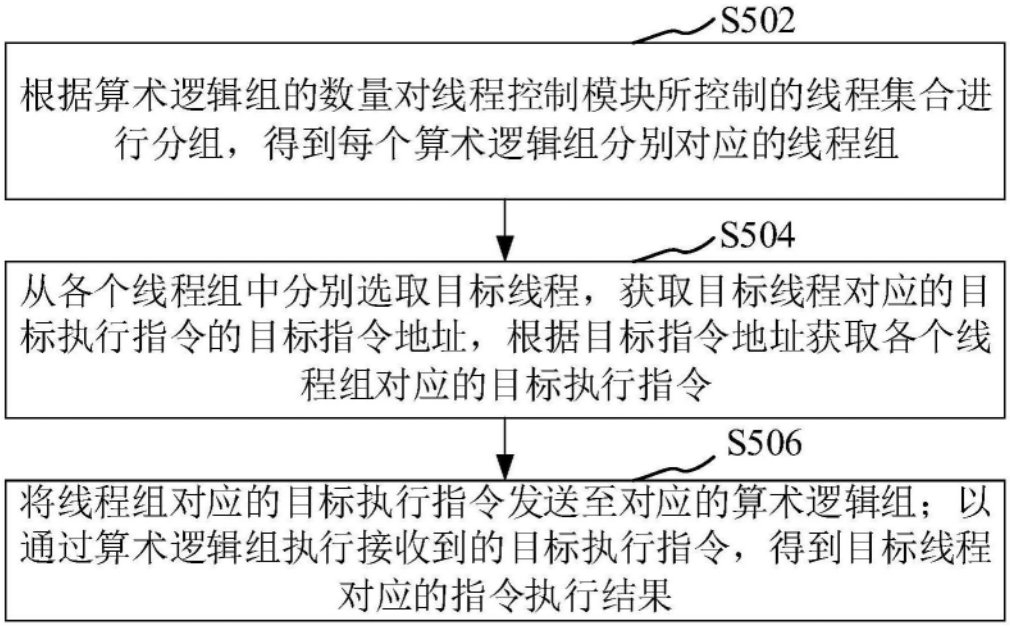

S502
根据算术逻辑组的数量对线程控制模块所控制的线程集合进行分组，得到每个算术逻辑组分别对应的线程组
S504
从各个线程组中分别选取目标线程，获取目标线程对应的目标执行指令的目标指令地址，根据目标指令地址获取各个线程组对应的目标执行指令
S506
将线程组对应的目标执行指令发送至对应的算术逻辑组；以通过算术逻辑组执行接收到的目标执行指令，得到目标线程对应的指令执行结果

图5