(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114595070 B
(45)授权公告日 2022.08.12

(21)申请号 202210501179.3

(22)申请日 2022.05.10

(65)同一申请的已公布的文献号
申请公布号 CN 114595070 A

(43)申请公布日 2022.06.07

(73)专利权人 上海登临科技有限公司
地址 201000 上海市浦东新区张江镇春晓路439号10栋

(72)发明人 牛剑锋 李晶晶

(74)专利代理机构 北京超凡宏宇专利代理事务所(特殊普通合伙) 11463
专利代理师 唐正瑜

(51)Int.Cl.
G06F 9/50(2006.01)

(56)对比文件
CN 106484519 A,2017.03.08
CN 111176806 A,2020.05.19
何炎祥等.通用图形处理器线程调度优化方法研究综述.《计算机学报》.2016,第39卷(第09期),全文.
孙琳琳等.基于多线程归并排序算法设计.《吉林大学学报(信息科学版)》.2015,第33卷(第01期),全文.

审查员 刘启军

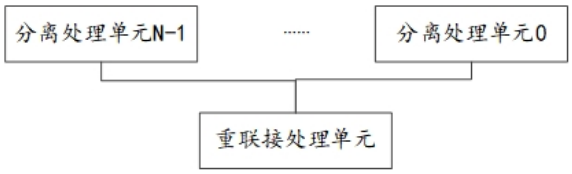
权利要求书2页 说明书12页 附图2页

(54)发明名称

一种处理器、多线程合并方法及电子设备

(57)摘要

本申请涉及一种处理器、多线程合并方法及电子设备，属于计算机技术领域。该处理器包括N个分离处理单元和重联接处理单元；N个分离处理单元与N个线程组一一对应，每个所述分离处理单元，用于对对应的线程组内的线程进行合并，得到合并后的剩余线程，每个线程组中的线程数大于等于2；所述重联接处理单元，用于将所述N个分离处理单元中任意一个分离处理单元输出的剩余线程与剩余N-1个分离处理单元输出的剩余线程进行线程合并，得到合并后的剩余线程。本申请易于流水线实现，能在较少比较逻辑电路的情况下，达到现有线程全比较合并方式所能实现的合并效果，从而减少了芯片的面积和功耗，有助于处理器频率提升，进而提高处理器性能。



CN 114595070 B

1.一种处理器,其特征在于,包括:

N个分离处理单元,与N个线程组一一对应,每个所述分离处理单元,用于对对应的线程组内的线程进行合并,得到合并后的剩余线程,N为大于等于2的正整数,每个线程组中的线程数大于等于2;

重联接处理单元,与每个所述分离处理单元均连接,所述重联接处理单元,用于将所述N个分离处理单元中任意一个分离处理单元输出的剩余线程与剩余N-1个分离处理单元输出的剩余线程进行线程合并,得到合并后的剩余线程;

若每个所述分离处理单元输出的线程数不为最大剩余线程数时,所述处理器还包括:

临时存储单元,用于临时存储线程;

聚合处理单元,与所述重联接处理单元、所述临时存储单元均连接,所述聚合处理单元,用于将所述重联接处理单元输出的线程与临时存储单元中存储的线程进行合并,并将合并后的线程存储在所述临时存储单元中。

2.根据权利要求1所述的处理器,其特征在于,所述临时存储单元所支持存储的线程数被配置为C,所述聚合处理单元包括B*C个比较逻辑电路,B为所述重联接处理单元输出的线程数。

3.根据权利要求1所述的处理器,其特征在于,每个所述分离处理单元输出的线程数被配置为A,A的取值为1至最大剩余线程数之间的整数。

4.根据权利要求1所述的处理器,其特征在于,所述重联接处理单元输出的线程数被配置为B,B 的取值为1至N*A之间的整数,A为每个所述分离处理单元输出的线程数,A的取值为1至最大剩余线程数之间的整数。

5.根据权利要求1所述的处理器,其特征在于,所述处理器还包括:内核;所述内核,用于将并行执行的多个线程分成N个线程组,并将每个线程组下发到对应的分离处理单元。

6.一种多线程合并方法,其特征在于,包括:

在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并,得到每个线程组合并后的剩余线程,N为大于等于2的正整数,每一个线程组中的线程在合并前的线程数大于等于2;

将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并,得到合并后的剩余线程;

其中,将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并,包括:

第1个时刻,将所述N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并,并保存,得到第1个时刻保存的线程合并结果,其中,所述指定数量小于初始时刻线程组中的线程合并后的剩余线程数;

第i个时刻,将所述N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并,得到第i个时刻的线程合并结果,直至所有不同线程组的剩余线程合并完,i依次取2至M的整数,M为所有不同线程组的剩余线程合并完所对应的时刻编号;

将第i个时刻的线程合并结果与第i-1个时刻保存的线程合并结果进行合并,并保存,得到第i个时刻保存的线程合并结果。

7.一种电子设备,其特征在于,包括:本体和如权利要求1-5中任一项所述的处理器。

一种处理器、多线程合并方法及电子设备

技术领域

[0001] 本申请属于计算机技术领域，具体涉及一种处理器、多线程合并方法及电子设备。

背景技术

[0002] 当前SIMT(Single Instruction Multiple Thread，单指令多线程)处理器中，多个线程并行执行访存操作时，有可能存在并行的多个线程的访问地址指向同一块空间(这里的“一块空间”指内存子系统中的基本操作单元，通常为缓存行(Cache Line))中的不同数据或同一数据。针对这一现象通常会进行合并访存操作，以减少访存的次数，节省带宽和功耗。

[0003] 当前常见的合并实现方式主要为对多线程进行全比较，将访问地址指向同一缓存行的线程进行合并。而全比较方式在线程较多情况下，需要较多比较逻辑电路，例如，16个线程的全比较需要120个比较逻辑电路，使得芯片的面积、功耗等均不理想。

发明内容

[0004] 鉴于此，本申请的目的在于提供一种处理器、多线程合并方法及电子设备，以改善全比较方式需要较多比较逻辑电路，导致芯片的面积、功耗均不理想的问题。

[0005] 本申请的实施例是这样实现的：

[0006] 第一方面，本申请实施例提供了一种处理器，包括：N个分离处理单元、重联接处理单元；N个分离处理单元，与N个线程组一一对应，每个所述分离处理单元，用于对对应的线程组内的线程进行合并，得到合并后的剩余线程，N为大于等于2的正整数，每个线程组中的线程数大于等于2；重联接处理单元，与每个所述分离处理单元均连接，所述重联接处理单元，用于将所述N个分离处理单元中任意一个分离处理单元输出的剩余线程与剩余N-1个分离处理单元输出的剩余线程进行线程合并，得到合并后的剩余线程。本申请实施例中，通过利用N个分离处理单元对各自对应的线程组内的线程进行合并，之后，再利用重联接处理单元将N个分离处理单元中任意一个分离处理单元输出的剩余线程与剩余N-1个分离处理单元输出的剩余线程进行线程合并，这样可以显著减少线程合并所需的比较逻辑电路的数量，从而减少芯片的面积、功耗，其中，在线程合并时，需要使用比较逻辑电路来比较线程的访问对象是否指向同一对象，访问对象指向同一对象的线程可以合并；此外，通过合理划分处理逻辑，易于流水线实现，利于高频处理器的设计，有利于提高处理器的频率，进而提高处理器的性能。

[0007] 结合第一方面实施例的一种可能的实施方式，若每个所述分离处理单元输出的线程数不为最大剩余线程数时，所述处理器还包括：临时存储单元和聚合处理单元；临时存储单元，用于临时存储线程；聚合处理单元，与所述重联接处理单元、所述临时存储单元均连接，所述聚合处理单元，用于将所述重联接处理单元输出的线程与临时存储单元中存储的线程进行合并，并将合并后的线程存储在所述临时存储单元中。本申请实施例中，若分离处理单元输出的线程数不为最大剩余线程数时，通过进一步增加临时存储单元和聚合处理单

元,以此来消除不同分离处理单元多次输出之间的重复线程,以增强方案的灵活性,此外,由于分离处理单元输出的线程数不为最大剩余线程数,使得在分离处理单元内无需对线程进行全比较,从而还可以进一步减少线程合并所需要的比较逻辑电路的数量。以及通过划分三层处理逻辑,更易于流水线实现。

[0008] 结合第一方面实施例的一种可能的实施方式,所述临时存储单元所支持存储的线程数被配置为C,所述聚合处理单元包括B*C个比较逻辑电路,B为所述重联接处理单元输出的线程数。本申请实施例中,临时存储单元所支持存储的线程数可以根据设计需要而配置,相应地,对应调整聚合处理单元所需要的比较逻辑电路的数量即可,使得该方案的配置更灵活,使其可以适用各种合并场景。

[0009] 结合第一方面实施例的一种可能的实施方式,每个所述分离处理单元输出的线程数被配置为A,A的取值为1至最大剩余线程数之间的整数。本申请实施例中,分离处理单元输出的线程数可以在指导标准范围(A的取值为1至最大剩余线程数之间的整数)内根据设计需要而配置,使得该方案的配置更灵活,使其可以适用各种合并场景。

[0010] 结合第一方面实施例的一种可能的实施方式,所述重联接处理单元输出的线程数被配置为B,B 的取值为1至N*A之间的整数,A为每个所述分离处理单元输出的线程数,A的取值为1至最大剩余线程数之间的整数。本申请实施例中,分离处理单元输出的线程数可以在指导标准范围(B 的取值为1至N*A之间的整数)内,根据设计需要而配置,使得该方案的配置更灵活,使其可以适用各种合并场景。

[0011] 结合第一方面实施例的一种可能的实施方式,所述处理器还包括:内核;所述内核,用于将并行执行的多个线程分成N个线程组,并将每个线程组下发到对应的分离处理单元。本申请实施例中,利用已有的内核,来将并行执行的多个线程分成N个线程组,并将每个线程组下发到对应的分离处理单元,通过赋予内核更多的功能,使得不需要再额外新增其他用于线程分组的元件。

[0012] 本申请实施例还提供了一种多线程合并方法,包括:在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并,得到每个线程组合并后的剩余线程,N为大于等于2的正整数,每一个线程组中的线程在合并前的线程数大于等于2;将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并,得到合并后的剩余线程。

[0013] 结合第二方面实施例的一种可能的实施方式,将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并,包括:第1个时刻,将所述N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并,并保存,得到第1个时刻保存的线程合并结果,其中,所述指定数量小于初始时刻线程组中的线程合并后的剩余线程数;第i个时刻,将所述N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并,得到第i个时刻的线程合并结果,直至所有不同线程组的剩余线程合并完,i依次取2至M的整数,M为所有不同线程组的剩余线程合并完所对应的时刻编号;将第i个时刻的线程合并结果与第i-1个时刻保存的线程合并结果进行合并,并保存,得到第i个时刻保存的线程合并结果。本申请实施例中,通过上述方式,使得可以通过分时将N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线

程合并，将合并过程拆分为多次合并，使得每次进行线程合并的线程数减少，进而可以进一步减少线程合并所需的比较逻辑电路的数量。

[0014] 结合第二方面实施例的一种可能的实施方式，将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并，包括：将所述N个线程组中任意一个线程组中的所有剩余线程与剩余N-1个线程组中的所有剩余线程进行线程合并。

[0015] 结合第二方面实施例的一种可能的实施方式，在所述在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并之前，所述方法还包括：将并行执行的多个线程分成N个线程组，每个线程组中的线程数大于等于2。

[0016] 第三方面，本申请实施例还提供了一种电子设备，包括：本体和如上述第一方面实施例和/或结合第一方面实施例的任一种可能的实施方式提供的处理器。

[0017] 本申请的其他特征和优点将在随后的说明书阐述，并且，部分地从说明书中变得显而易见，或者通过实施本申请实施例而了解。本申请的目的和其他优点可通过在所写的说明书以及附图中所特别指出的结构来实现和获得。

附图说明

[0018] 为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。通过附图所示，本申请的上述及其它目的、特征和优势将更加清晰。在全部附图中相同的附图标记指示相同的部分。并未刻意按实际尺寸等比例缩放绘制附图，重点在于示出本申请的主旨。

[0019] 图1示出了本申请实施例提供的一种处理器的结构示意图。

[0020] 图2示出了本申请实施例提供的又一种处理器的结构示意图。

[0021] 图3示出了本申请实施例提供的一种多线程合并方法的流程示意图。

[0022] 图4示出了本申请实施例提供的一种电子设备的结构示意图。

具体实施方式

[0023] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。

[0024] 应注意到：相似的标号和字母在下面的附图中表示类似项，因此，一旦某一项在一个附图中被定义，则在随后的附图中不需要对其进行进一步定义和解释。同时，在本申请的描述中诸如“第一”、“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0025] 再者，本申请中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存

在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。

[0026] 鉴于现有线程合并方式(如全比较方式)存在的需要较多比较逻辑电路，导致芯片的面积、功耗均不理想的问题，本申请实施例提供了一种处理器，以实现在减少比较逻辑电路的情况下，达到现有线程合并方式所实现的合并效果。

[0027] 本申请实施例提供的处理器，如图1所示，该处理器包括N个分离处理单元和重联接处理单元，N为大于等于2的正整数。本申请通过在现有处理器的基本架构的基础上，新增了N个分离处理单元以及重联接处理单元，以实现在保证线程合并效果的情况下，达到减少使用的比较逻辑电路的目的。

[0028] 该N个分离处理单元与N个线程组一一对应，每个分离处理单元，用于对对应的线程组内的线程进行合并，得到合并后的剩余线程，每个线程组中的线程数大于等于2。需要说明的是，分离处理单元的数量也可以多于线程组的数量，即可以存在冗余的分离处理单元，因此不能将分离处理单元的数量与线程组的数量相等的情况理解成是对本申请的限制。

[0029] 一种实施方式下，可以是处理器的内核，将并行执行的多个线程分成N个线程组，并将每个线程组下发到对应的分离处理单元。此时，处理器还包括内核。例如，假设内核将并行执行的16个线程分成2个线程组，每个线程组包含8个线程，相应地，分离处理单元的数量也为2。并行执行的多个线程通常为8的整数倍，例如为16、24、32、64……个线程。通过赋予内核更多的功能，使得不需要再额外新增其他用于线程分组的元件，从而有利于减少芯片的面积、功耗。

[0030] 在分组时，可以是将并行执行的多个线程均匀地分为N个线程组，使得每个线程组中的线程数相等。例如，将并行执行的16个线程分成2个线程组，每个线程组中的线程数为8；又例如，将并行执行的24个线程分成4个线程组，每个线程组中的线程数为6。需要说明的是，在对并行执行的多个线程进行分组时，也可以不均分，因此，不能将此处示例的分组方式，理解成是对本申请的限制。

[0031] 其中，所谓剩余线程是指线程合并后剩余的线程，每一个剩余线程均对应有一编码信息，以用于记录该剩余线程的合并情况，根据该编码信息，即可获悉该剩余线程是否由线程合并得到，以及由哪些线程合并得到。该编码信息包含的比特(bit)位可以与分组前的总线程数的数量一致，当然也可以多于总线程数，即可以允许存在冗余比特位。

[0032] 例如，以16个线程(如为线程0~线程15)的线程合并为例进行说明，则合并后剩余的每一个剩余线程的编码信息都对应有16比特位(如为比特位0~比特位15)，编码信息中的每一个比特位对应一个线程。例如，比特位0对应线程0，比特位1对应线程1，比特位2对应线程2，以此类推，比特位15对应线程15，每一个比特位的值为0或1，其中可以用1来记录合并的线程数，当然也可以用0来记录合并的线程数。

[0033] 为了更好的理解，下面举例进行说明：假设这16个线程中，线程0和线程15可以合并，合并后为线程0，其余的线程(即线程1~线程14)均不能合并，则合并后的剩余线程为15条。若用编码信息中比特位的值为1来记录合并的线程数，则合并后的线程0对应的编码信息中比特位0和比特位15对应的数值为1，其余比特位的数值为0。而线程1~线程14对应的编码信息中各比特位的值均为0。也即，此时，线程0对应的编码信息为1000000000000001，线程1~线程14对应的编码信息为0000000000000000。

[0034] 又例如，这16个线程中，线程0~线程7这8条线程可以合并成一条线程，合并后为线程0；线程8~线程15这8条线程可以合并成一条线程，合并后为线程1，则合并后的剩余线程为2条。若用编码信息中比特位的值为1来记录合并的线程数，则合并后的线程0对应的编码信息中比特位0至比特位7对应的数值为1，其余比特位的数值为0；合并后的线程1对应的编码信息中比特位8至比特位15对应的数值为1，其余比特位的数值为0。此时，合并后的线程0对应的编码信息为0000000011111111，合并后的线程1对应的编码信息为1111111100000000。

[0035] 每个分离处理单元在对对应线程组内的线程进行合并时，可以是对线程组中访问对象指向同一对象的线程进行合并，例如，以内存访问请求线程为例，则在合并时，在线程组内对访问地址指向同一缓存行的线程进行合并，得到线程组合并后的剩余线程。在进行线程合并时，需要对线程的访问对象进行比较，访问对象指向同一对象的线程可以合并，也即，可以是先识别出哪些线程的访问对象指向同一对象，然后再对访问对象指向同一对象的线程进行合并。在比较线程的访问对象时可以基于比较逻辑电路实现。需要说明的是，本申请实施例所示的多线程合并方法适用于所有线程合并需求的场景，并不限于内存访问场景，例如，可以是适用于多组数据的比较操作之类的特定场景。

[0036] 其中，分离处理单元包含用于线程比较的比较逻辑电路以及进行线程合并的执行逻辑部分。比较逻辑电路用于比较线程的访问对象是否指向同一对象，也即用于比较线程的访问对象是否相同。执行逻辑部分用于将线程的访问对象指向同一对象的线程进行合并，可以基于现有的执行逻辑部分实现。在进行线程合并时，需要对线程的访问对象进行比较，访问对象指向同一对象的线程可以合并。

[0037] 每个分离处理单元输出的线程数被配置为A，A的取值为1至最大剩余线程数(可以等于对应线程组中的线程数)之间的整数。例如，该分离处理单元对8个线程进行合并，最坏情况下，8个线程均不能合并，则最大剩余线程数为8，则A的取值可以是1、2、3、4、5、6、7、8中的任一数值。

[0038] 需要说明的是，分离处理单元输出的线程数被配置为A，则分离处理单元输出的最大线程数不能超过A，只能小于等于A。A值越小所需比较逻辑电路越少，时序也会越好，但过小会导致分离处理单元的吞吐量(throughput)降低。例如，以16个线程并行执行为例，被均匀分为2个线程组，每个线程组包含8个线程；当分离处理单元的输出被配置为1时，分离处理单元仅需要7个比较逻辑电路；当分离处理单元的输出被配置为2时，分离处理单元仅需要7+6个比较逻辑电路；当分离处理单元的输出被配置为3时，分离处理单元仅需要7+6+5个比较逻辑电路；以此类推，当分离处理单元的输出被配置为6时，分离处理单元仅需要7+6+5+4+3+2个比较逻辑电路，当分离处理单元的输出被配置为7或8时，分离处理单元仅需要7+6+5+4+3+2+1个比较逻辑电路。

[0039] 为了更好的理解，假设输出被配置为1，则该输出可以是线程0~线程7中任一个线程与其余7个线程合并后的结果，以线程0为例，该输出为线程0与线程1~线程7合并后的结果，假设不能合并，则该输出即为线程0；若线程0和线程1、线程3可以合并，则该输出为由线程0、线程1、线程3合成后的全新线程0。

[0040] 重联接处理单元与每个分离处理单元均连接。重联接处理单元，用于将N个分离处理单元中任意一个分离处理单元输出的剩余线程与剩余N-1个分离处理单元输出的剩余线

程进行线程合并，得到合并后的剩余线程。例如，N为3时，重联接处理单元用于将分离处理单元1输出的剩余线程分别与分离处理单元2输出的剩余线程、分离处理单元3输出的剩余线程进行线程合并，以及还将分离处理单元2输出的剩余线程与分离处理单元3输出的剩余线程进行线程合并。也即，将3个分离处理单元中任意一个分离处理单元输出的剩余线程与其余两个分离处理单元输出的剩余线程进行合并。

[0041] 重联接处理单元包含用于线程比较的比较逻辑电路以及进行线程合并的执行逻辑部分。其中，重联接处理单元输出的线程数被配置为B，B 的取值为1至N*A之间的整数，即为1、2、……N*A-1、N*A之间的任一数值。其中，A为每个分离处理单元输出的线程数。例如，N为2，A为2，则 B的取值为1、2、3、4中的任一数值。

[0042] 需要说明的是，重联接处理单元输出的线程数被配置为B，则重联接处理单元输出的最大线程数不能超过B，只能小于等于B。B值越小所需比较逻辑电路越少，时序也会越好，但过小会导致重联接处理单元的吞吐量(throughput)降低，B值越小所需比较逻辑电路越少。例如，假设N为2，A为2，若B为1，则只需要1*2个比较逻辑电路；若B为2、3、4，则只需要2*2个比较逻辑电路。又例如，N为3，A为3，若B为1，则只需要1*6个比较逻辑电路；若B为2，则只需要2*6个比较逻辑电路；若B为3，只需要3*6个比较逻辑电路；若B为4，只需要3*6+1*3个比较逻辑电路；若B为5；只需要3*6+2*3个比较逻辑电路；若B为6、7、8、9；只需要3*6+3*3个比较逻辑电路。需要说明的是，此处的示例，仅是为了说明不同的B被配置为不同的值，所需的比较逻辑电路可能会不同。

[0043] 若每个分离处理单元输出的线程数不为最大剩余线程数，也即不为全输出(所谓全输出也即一次将所有合并后的剩余线程输出)时，即需要多次输出才能将合并后的剩余线程输出完，此时，处理器还包括：聚合处理单元和临时存储单元，以消除不同分离处理单元多次输出之间的重复线程，此时，处理器的原理示意图如图2所示。通过合理划分处理器的合并逻辑(分离处理单元合并-重联接处理单元合并-聚合处理单元合并)，易于流水线实现，有利于提高处理器的频率，进而提高处理器的性能。

[0044] 临时存储单元用于临时存储线程。聚合处理单元与重联接处理单元、临时存储单元均连接。聚合处理单元，用于将重联接处理单元输出的线程与临时存储单元中存储的线程进行合并，并将合并后的线程存储在临时存储单元中。

[0045] 为了更好的理解，举例进行说明，假设N为3，对应的3个分离处理单元分别为分离处理单元0、分离处理单元1分离处理单元2；每个线程组中的线程数为8，A为1(即最大输出为1)，B为3(即最大输出为3)，此时，重联接处理单元包含1*2+1*1个比较逻辑电路，则有：

[0046] 第1个时刻，重联接处理单元分别将分离处理单元0输出的线程与分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，以及将分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，也即将3个线程中任意一个线程与其余2个线程进行合并，得到第1个时刻的合并结果并输出给聚合处理单元，聚合处理单元将其保存在临时存储单元中；

[0047] 第2个时刻，重联接处理单元分别将分离处理单元0输出的线程与分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，以及将分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，得到第2个时刻的合并结果并输出给聚合处理单元，聚合处理单元将其与第1个时刻保存在临时存储单元中的线程合并结果进行合并，并更新临时

存储单元中保存的结果，得到第2个时刻保存的线程合并结果；

[0048] 第3个时刻，重联接处理单元分别将分离处理单元0输出的线程与分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，以及将分离处理单元1输出的线程、分离处理单元2输出的线程进行合并，得到第3个时刻的合并结果并输出给聚合处理单元，聚合处理单元将其与第2个时刻保存在临时存储单元中的线程合并结果进行合并，并更新临时存储单元中保存的结果，得到第3个时刻保存的线程合并结果；

[0049] 以此类推，直至所有不同线程组的剩余线程合并完。

[0050] 其中，临时存储单元所支持存储的线程数被配置为C，则聚合处理单元包括B*C个比较逻辑电路以及进行线程合并的执行逻辑部分，B为重联接处理单元输出的线程数，例如，B为2，C为8，则聚合处理单元包括16个比较逻辑电路。C的最大取值为分组前的总线程数-1。C值越大，可处理的复杂请求组合形式越多，效果越好，但相应的面积和功耗也会越大，因此，可以根据性能需求选择一个合理值。

[0051] 需要说明的是，当处理器不包含聚合处理单元时，此时，分离处理单元的输出为全输出(所谓全输出也即一次将所有合并后的剩余线程输出)相应地，分离处理单元、重联接处理单元在进行线程合并时，均为全比较。而当处理器包含聚合处理单元时，分离处理单元和/或重联接处理单元的输出可以不为全输出，相应地，分离处理单元和/或重联接处理单元在进行线程合并时，可以不为全比较，能进一步减少线程合并所需的比较逻辑电路。如上述示例的那样，假设N为2，每个线程组的数量为8，则A为2时，分离处理单元仅需要7+6个比较逻辑电路，若B为2，则重联接处理单元只需要2*2个比较逻辑电路；若C为8，则聚合处理单元需要2*8个比较逻辑电路，此时，共计需要2*(7+6)+2*2+2*8=46个比较逻辑电路。

[0052] 上述的处理器可以是具备多线程并行访问功能的处理器，例如，可以是单指令多线程(Single Instruction Multiple Thread，SIMT)处理器、多指令多线程(Multiple Instruction Multiple Thread，MIMT)处理器、单指令多数据流(Single Instruction Multiple Data，SIMD)处理器、多指令多数据流(Multiple Instruction Multiple Data，SIMD)处理器等。而这些处理器可以是通用处理器，例如，央处理器(Central Processing Unit，CPU)、网络处理器(Network Processor，NP)、数字信号处理器(Digital Signal Processor，DSP)、图形处理器(Graphics Processing Unit，GPU)等。

[0053] 采用本申请所示的多线程合并方法，能减少比较逻辑电路数量。下面对采用本申请所示的多线程合并方法，能减少比较逻辑电路的原理进行说明。若是采用现有线程合并方法，16个线程的全比较需要120个比较逻辑电路，具体为：第一个线程需要与其余的15个线程分别进行比较，需要15个比较逻辑电路，对于第二个线程需要与除第一个线程外的其余的14个线程分别进行比较，需要14个比较逻辑电路，对于第三个线程需要与除第一个线程、第二个线程外的其余的13个线程分别进行比较，需要13个比较逻辑电路，以此类推，第15个线程只需要与第16个线程进行比较，需要1个比较逻辑电路，共计需要15+14+13+12+……+2+1=120个比较逻辑电路。

[0054] 而采用本申请所示的多线程合并方法，通过将16个线程分成N个线程组，然后在每个线程组内对线程进行合并，之后再对不同线程组的线程合并结果再次进行合并，这样可以减少所使用的比较逻辑电路。例如，假设将这16个线程分组2个线程组，每个线程组包含8个线程，则在每个线程组进行线程合并时，即便是在每个线程组内进行线程全比较，只需要

7+6+5+4+3+2+1=28个比较逻辑电路，之后再对这2个线程组的线程合并结果再次进行全比较合并，经过在线程组内进行线程合并后，每个线程组合并后的剩余线程必然小于等于8，假设其中一个线程组合并后的剩余线程为7，另一个线程组合并后的剩余线程为5，则这2个线程组的线程合并结果再次进行全比较合并时，只需要7*5=35个比较逻辑电路，此时共计使用的比较逻辑电路为28+28+35=91个，明显小于现有的120个比较逻辑电路。

[0055] 此外，若是采用分时也即多次输出的方式进行线程合并时，则在每个线程组内进行线程合并时，也可以不进行线程全比较，同理在对不同线程组输出的剩余线程进行合并时，也可以不进行全比较，这样还可以进一步减少比较逻辑电路的数量。例如，假设每个线程组对应的输出线程为1，还是以每个线程组包含8个线程为例，则仅需要7个比较逻辑电路；若每个线程组对应的输出线程为2，则只需要7+6个比较逻辑电路，这样相对于线程全比较来说，可以减少所使用的比较逻辑电路的数量。

[0056] 基于同样的发明构思，本申请实施例还提供了一种电子设备，该电子设备包含上述处理器以及本体(电子设备基本组件)。该电子设备可以是智能手机、平板电脑、个人电脑、显示屏、服务器等设备。

[0057] 电子设备实施例所提供的处理器，其实现原理及产生的技术效果和前述处理器实施例相同，为简要描述，电子设备实施例部分未提及之处，可参考前述处理器实施例中相应内容。

[0058] 基于同样的发明构思，本申请实施例还提供了一种线程合并方法，本申请所示的线程合并方法，可以应用于所有具有线程合并需求的场景，包括但不限于数据访问场景、多组数据的比较场景等。例如，可以应用于处理器访问存储器的访问场景，在多个线程并行执行访存操作时，有可能存在并行的多个线程的访问地址指向同一块空间(缓存行)，为了减少访存的次数，节省带宽和功耗，需要对访问对象指向同一对象的线程进行合并。

[0059] 下面将结合图3，对本申请实施例提供的多线程合并方法进行说明。

[0060] S1：在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并，得到每个线程组合并后的剩余线程。

[0061] 在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并，得到每个线程组合并后的剩余线程，获取N个线程组中每一个线程组的线程合并后的剩余线程，N为大于等于2的正整数，每一个线程组中的线程在合并前的线程数大于等于2。同一个线程组的线程合并后的剩余线程是完全独立的线程，同一个线程组内的各个剩余线程不能再进行合并。

[0062] 在对线程组内对访问对象指向同一对象的线程进行合并时，可以是基于处理器中的分离处理单元来对其线程进行合并。

[0063] 为了减少在进行线程合并时使用的比较逻辑电路的数量，本申请通过将并行执行的多个线程分成N个线程组，并在每个线程组内进行线程合并，得到N个线程组中每一个线程组的线程合并后的剩余线程。相应地，在S1之前，该多线程合并方法还包括：将并行执行的多个线程分成N个线程组，每个线程组中的线程数大于等于2。可以是利用处理器中的内核来将并行执行的多个线程分成N个线程组。

[0064] S2：将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并，得到合并后的剩余线程。

[0065] 在获取到N个线程组中每一个线程组的线程合并后的剩余线程后，将N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并，得到合并后的剩余线程。通过将不同线程组的线程合并后的剩余线程再次合并，以消除不同线程组之间的重复线程。

[0066] 第一种实施方式下，在将N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并时，可以是将N个线程组中任意一个线程组中的所有剩余线程与剩余N-1个线程组中的所有剩余线程进行线程合并。此时，可以是利用上述的重联接处理单元来将N个线程组中任意一个线程组中的所有剩余线程与剩余N-1个线程组中的所有剩余线程进行线程合并。例如，分离处理单元将对应线程组中的所有剩余线程全输出至重联接处理单元，重联接处理单元将N个线程组中任意一个线程组中的所有剩余线程与剩余N-1个线程组中的所有剩余线程进行线程合并。

[0067] 为了便于理解，举例进行说明，假设N为2，其中，一个线程组(如为第一线程组)合并后的剩余线程数为3，另一个线程组(如为第二线程组)合并后的剩余线程数为4，则将第一线程组中的每一个剩余线程分别与第二线程组中的4个线程依次进行线程合并。

[0068] 第二种实施方式下，在将N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并时，可以是：第1个时刻，将N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并，并保存，得到第1个时刻保存的线程合并结果，其中，指定数量小于初始时刻线程组中的线程合并后的剩余线程数；第i个时刻，将N个线程组中任意一个线程组中的指定数量的剩余线程与剩余N-1个线程组中的指定数量的剩余线程进行线程合并，得到第i个时刻的线程合并结果，直至所有不同线程组的剩余线程合并完，i依次取2至M的整数，M为所有不同线程组的剩余线程合并完所对应的时刻编号；将第i个时刻的线程合并结果与第i-1个时刻保存的线程合并结果进行合并，并保存，得到第i个时刻保存的线程合并结果。此时，可以是利用上述的重联接处理单元与聚合处理单元的配合来实现分时将N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并。

[0069] 需要说明的是，在第二种实施方式下，在将不同线程组的剩余线程进行合并时，只能选取不超过指定数量的剩余线程进行合并。若指定数量为2，则每次至多只能选取2个剩余线程与其余线程组中的至多2个剩余线程进行线程合并。

[0070] 为了便于理解，结合上述举例进行说明，假设N为2，其中，一个线程组(如为第一线程组)合并后的剩余线程数为3，另一个线程组(如为第二线程组)合并后的剩余线程数为4。若指定数量为1，则有：

[0071] 第1个时刻(初始时刻)，从第一线程组的3个剩余线程中选择一个剩余线程与从第二线程组的4个剩余线程中选择一个剩余线程进行线程合并，并保存，得到第1个时刻保存的线程合并结果；

[0072] 第2个时刻，从第一线程组的2个剩余线程中选择一个剩余线程与从第二线程组的3个剩余线程中选择一个剩余线程进行线程合并，得到第2个时刻的线程合并结果，再将第2个时刻的线程合并结果与第1个时刻保存的线程合并结果进行合并，并保存，得到第2个时刻保存的线程合并结果；

[0073] 第3个时刻，从第一线程组的1个剩余线程中选择一个剩余线程与从第二线程组的

2个剩余线程中选择一个剩余线程进行线程合并，得到第3个时刻的线程合并结果，再将第3个时刻的线程合并结果与第2个时刻保存的线程合并结果进行合并，并保存，得到3个时刻保存的合并结果；

[0074] 第4个时刻，从第一线程组的0个剩余线程中选择一个剩余线程与从第二线程组的1个剩余线程中选择一个剩余线程进行线程合并，得到第4个时刻的线程合并结果，再将第4个时刻的线程合并结果与第3个时刻保存的线程合并结果进行合并，并保存，得到第4个时刻保存的线程合并结果。

[0075] 在该示例中，所有的不同线程组的剩余线程合并完所对应的时刻编号为4，因此，M的取值为4。其中，上述的实现过程可以是：

[0076] 第1个时刻(初始时刻)，分离处理单元1从第一线程组的3个剩余线程中选择一个剩余线程输出至重联接处理单元，分离处理单元2从第二线程组的4个剩余线程中选择一个剩余线程输出至重联接处理单元，由重联接处理单元进行线程合并，并将线程合并结果输出给聚合处理单元，聚合处理单元将其保存在临时存储单元中，得到第1个时刻保存的线程合并结果；

[0077] 第2个时刻，分离处理单元1从第一线程组的2个剩余线程中选择一个剩余线程输出至重联接处理单元，分离处理单元2从第二线程组的3个剩余线程中选择一个剩余线程至重联接处理单元，由重联接处理单元进行线程合并，得到第2个时刻的线程合并结果，并将第2个时刻的线程合并结果输出至聚合处理单元，聚合处理单元再将第2个时刻的线程合并结果与第1个时刻保存在临时存储单元的线程合并结果进行合并，并保存，得到第2个时刻保存的线程合并结果；

[0078] 第3个时刻，分离处理单元1从第一线程组的1个剩余线程中选择一个剩余线程输出至重联接处理单元，分离处理单元2从第二线程组的2个剩余线程中选择一个剩余线程出至重联接处理单元，由重联接处理单元进行线程合并，得到第3个时刻的线程合并结果并输出至聚合处理单元，聚合处理单元再将第3个时刻的线程合并结果与第2个时刻保存在临时存储单元的线程合并结果进行合并，并保存，得到3个时刻保存的合并结果；

[0079] 第4个时刻，分离处理单元1从第一线程组的0个剩余线程中选择一个剩余线程输出至重联接处理单元，分离处理单元2从第二线程组的1个剩余线程中选择一个剩余线程输出至重联接处理单元，由重联接处理单元进行线程合并，得到第4个时刻的线程合并结果并输出给聚合处理单元，聚合处理单元再将第4个时刻的线程合并结果与第3个时刻保存在临时存储单元的线程合并结果进行合并，并保存，得到第4个时刻保存的线程合并结果。

[0080] 为了避免不同时刻在进行线程合并时，将之前已合并后的线程再次进行合并，造成资源浪费，可选地，每一个线程组均对应有一个唯一的线程组编号，以此来区分不同的线程组，同一个线程组中的每一条线程均携带有该线程组对应的线程组编号。由于在S1中已经在线程组内对具有相同线程组编号的不同线程进行了合并，因此在S2中进行剩余线程合并时，不再对具有相同线程组编号的剩余线程进行合并，仅对具有不同线程组编号的剩余线程进行合并，合并后的剩余线程会继承合并成该条剩余线程的所有线程的线程组编号，例如，合并后的剩余线程由线程组编号为id1的线程和线程组编号为id2的线程合并得到，则该合并的剩余线程的线程组编号既包含线程组编号id1，又包含线程组编号id2，之后，既包含线程组编号id1，又包含线程组编号id2的剩余线程不会再与线程组编号为id1的线程，

和/或,线程组编号为id2的线程进行合并。

[0081] 为了更好的理解,以上述的示例进行说明,假设第一线程组的3个剩余线程分别为线程0、线程5、线程7;第二线程组的4个剩余线程,分别为线程1、线程2、线程3、线程6,则有:

[0082] 第1个时刻,将第一线程组中的线程0和第二线程组中的线程1进行合并,假设可以合并,得到新的线程0(或线程1),该线程0既包含线程组编号id1,又包含线程组编号id2;

[0083] 第2个时刻,将第一线程组中的线程5和第二线程组中的线程2进行合并,假设不可以合并,由于第一个时刻合并保存的线程0,既包含线程组编号id1,又包含线程组编号id2,因此不需要再与线程5(包含线程组编号id1)以及线程2(包含线程组编号id2)再进行合并,直接将线程5、线程2进行保存,得到第2个时刻保存的线程合并结果;

[0084] 第3个时刻,将第一线程组中的线程7和第二线程组中的线程3进行合并,假设不可以合并,则在与第2个时刻保存的线程合并结果(包括线程5、线程2、全新线程0)进行合并时,只需要将线程5、线程3进行合并,假设可以合并,得到全新线程3(既包含线程组编号id1,又包含线程组编号id2),以及将线程7与线程2进行合并,假设可以合并,得到全新线程2(既包含线程组编号id1,又包含线程组编号id2)并保存,得到第3个时刻保存的线程合并结果(全新线程0、全新线程2、全新线程3);

[0085] 第4个时刻,由于3个时刻保存的线程合并结果中的全新线程0、全新线程2、全新线程3均包含线程组编号id2,因此不在与线程6进行合并,直接保存线程6,得到第4个时刻保存的线程合并结果(全新线程0、全新线程2、全新线程3、线程6)。

[0086] 如图4所示,图4示出了本申请实施例提供的一种用于执行上述所示的多线程合并方法的电子设备200的结构框图。所述电子设备200包括:收发器210、存储器220、通讯总线230以及处理器240。

[0087] 所述收发器210、所述存储器220、处理器240各元件相互之间直接或间接地电性连接,以实现数据的传输或交互。例如,这些元件相互之间可通过一条或多条通讯总线230或信号线实现电性连接。其中,收发器210用于收发数据。存储器220用于存储计算机程序,该计算机程序包括至少一个可以软件或固件(Firmware)的形式存储于所述存储器220中或固化在所述电子设备200的操作系统(Operating System,OS)中的软件功能模块。所述处理器240,用于执行存储器220中存储的软件功能模块或计算机程序。例如,处理器240,用于在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并,得到每个线程组合并后的剩余线程,N为大于等于2的正整数,每一个线程组中的线程在合并前的线程数大于等于2;将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并,得到合并后的剩余线程。

[0088] 其中,存储器220可以是,但不限于,随机存取存储器(Random Access Memory,RAM),只读存储器(Read Only Memory,ROM),可编程只读存储器(Programmable Read-Only Memory,PROM),可擦除只读存储器(Erasable Programmable Read-Only Memory,EPROM),电可擦除只读存储器(Electric Erasable Programmable Read-Only Memory,EEPROM)等。

[0089] 处理器240可能是一种集成电路芯片,具有信号的处理能力。上述的处理器可以是通用处理器,包括中央处理器(Central Processing Unit, CPU)、网络处理器(Network Processor,NP)、数字信号处理器(Digital Signal Processor,DSP)、图形处理器(Graphics Processing Unit,GPU)。可以实现或者执行本申请实施例中的公开的各方法、

步骤及逻辑框图。此外，通用处理器可以是微处理器。

[0090] 需要说明的是，本说明书中的各个实施例均采用递进的方式描述，每个实施例重点说明的都是与其他实施例的不同之处，各个实施例之间相同相似的部分互相参见即可。

[0091] 在本申请所提供的几个实施例中，应该理解到，所揭露的装置和方法，也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的，例如，附图中的流程图和框图显示了根据本申请的多个实施例的装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现方式中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0092] 另外，在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分，也可以是各个模块单独存在，也可以两个或两个以上模块集成形成一个独立的部分。

[0093] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案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计算机可读存储介质中，包括若干指令用以使得一台计算机设备(可以是个人计算机，笔记本电脑，服务器，或者电子设备等)执行本申请各个实施例所述方法的全部或部分步骤。

[0094] 以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应所述以权利要求的保护范围为准。

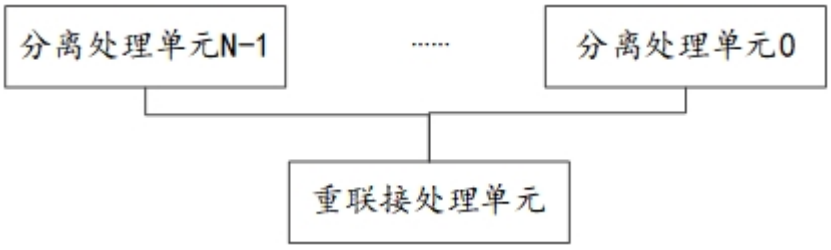

分离处理单元N-1
……
分离处理单元0
重联接处理单元

图1

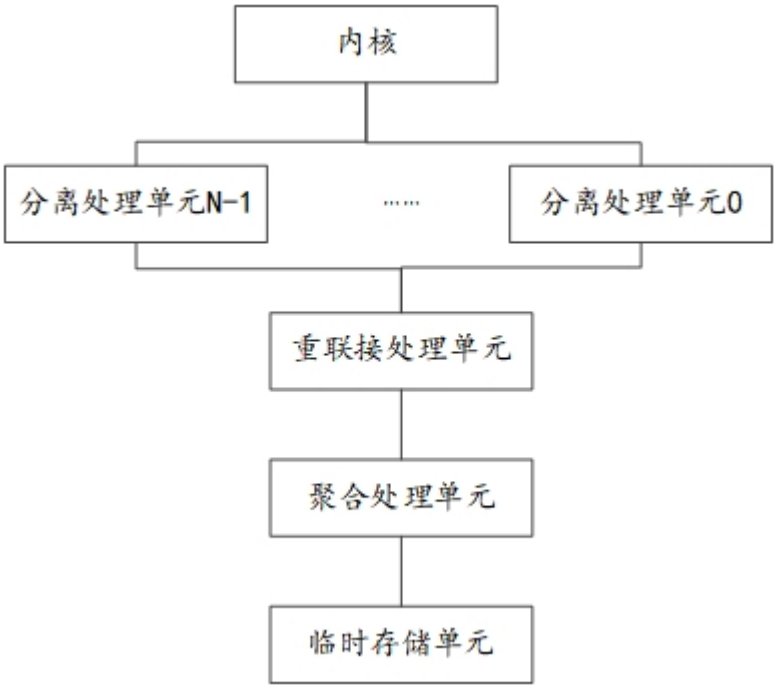

内核
分离处理单元N-1
……
分离处理单元0
重联接处理单元
聚合处理单元
临时存储单元

图2

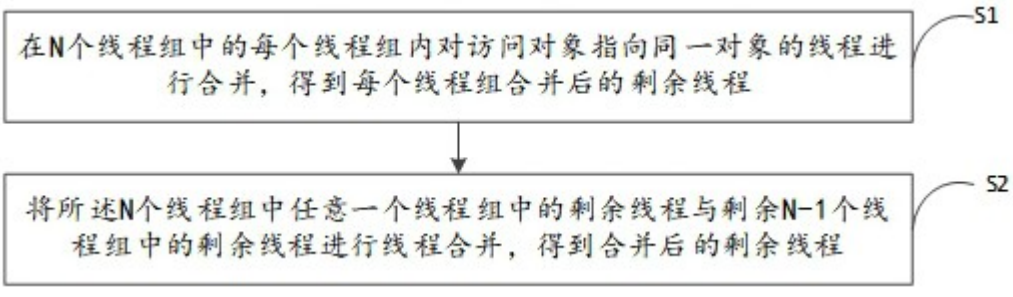

在N个线程组中的每个线程组内对访问对象指向同一对象的线程进行合并，得到每个线程组合并后的剩余线程
S1
将所述N个线程组中任意一个线程组中的剩余线程与剩余N-1个线程组中的剩余线程进行线程合并，得到合并后的剩余线程
S2

图3

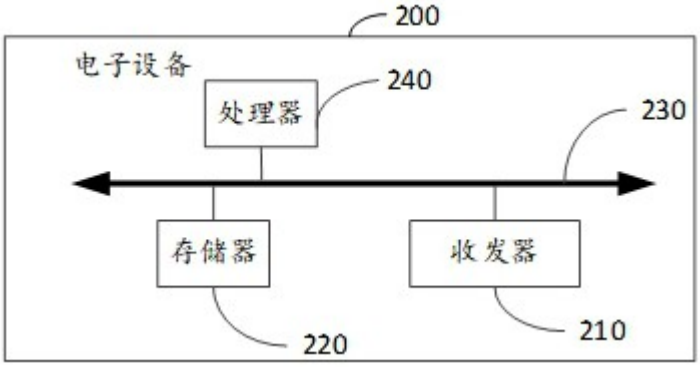


图4