(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119166321 A
(43)申请公布日 2024.12.20

(21)申请号 202311438764.4

(22)申请日 2023.10.31

(66)本国优先权数据
202310734151.9 2023.06.19 CN

(71)申请人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 吴兴良 林越

(74)专利代理机构 北京三高永信知识产权代理有限责任公司 11138
专利代理师 李芳

(51)Int.Cl.
G06F 9/50(2006.01)
G06F 9/38(2018.01)
G06F 9/30(2018.01)

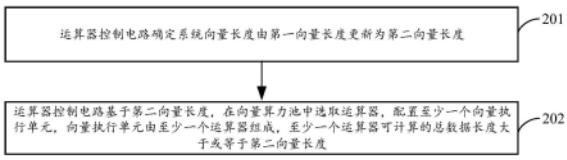
权利要求书2页 说明书15页 附图5页

(54)发明名称

向量算力分配的方法、处理器和设备

(57)摘要

一种向量算力分配的方法、处理器和设备，属于向量计算技术领域，本方法在系统向量长度更新后，根据更新后的系统向量长度，在向量算力池中选取运算器，重新配置向量执行单元，使得新配置的向量执行单元在能够执行向量计算的前提下，充分利用所包括的运算器，尽量避免运算器不参与计算，而造成的向量执行单元的算力浪费。



CN 119166321 A

1.一种向量算力分配的方法,其特征在于,所述方法包括:

确定系统向量长度由第一向量长度更新为第二向量长度;

基于所述第二向量长度,在向量算力池中选取运算器,配置第一向量执行单元,所述第一向量执行单元为执行所述第二向量长度的一个向量执行单元,所述第一向量执行单元由至少一个运算器组成,所述至少一个运算器可计算的总数据长度大于或等于所述第二向量长度;

由所述第一向量执行单元执行所述第二向量长度的向量运算。

2.根据权利要求1所述的方法,其特征在于,所述基于所述第二向量长度,在向量算力池中选取运算器,配置至少一个向量执行单元,包括:

基于所述第二向量长度和单个运算器可计算的数据长度之间的倍数关系,确定所述第一向量执行单元包括的运算器的第一数目;

在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元。

3.根据权利要求2所述的方法,其特征在于,在所述第二向量长度小于所述第一向量长度的情况下,所述在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元之前,所述方法还包括:

比较第二向量执行单元包括的运算器的第二数目和所述第一数目的大小,确定所述第二数目大于所述第一数目,其中,所述第二向量执行单元为执行所述第一向量长度的一个向量执行单元。

4.根据权利要求2所述的方法,其特征在于,在所述第二向量长度大于所述第一向量长度的情况下,所述在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元之前,所述方法还包括:

比较第二向量执行单元包括的运算器的第三数目和所述第一数目的大小,确定所述第三数目小于所述第一数目,其中,所述第二向量执行单元为执行所述第一向量长度的一个向量执行单元。

5.根据权利要求2-4中任一项所述的方法,其特征在于,所述在向量算力池中选取运算器,将选取的每第一数目个运算器配置为一个向量执行单元,包括:

根据向量算力池中每个运算器的标识,在所述向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元。

6.根据权利要求1-5中任一项所述的方法,其特征在于,所述方法还包括:

基于所述第二向量长度,选取向量物理寄存器,配置至少一个向量物理寄存器组,每个向量物理寄存器组包括至少一个向量物理寄存器,所述至少一个向量物理寄存器可存储的总数据长度大于或等于所述第二向量长度。

7.根据权利要求6所述的方法,其特征在于,所述基于所述第二向量长度,选取向量物理寄存器,配置至少一个向量物理寄存器组,包括:

基于所述第二向量长度和单个向量物理寄存器可存储的数据长度之间的倍数关系,确定每个向量物理寄存器组包括的向量物理寄存器的第四数目;

选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

8.根据权利要求7所述的方法,其特征在于,在所述第二向量长度小于所述第一向量长

度的情况下,所述选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组之前,所述方法还包括:

确定当前的每个向量物理寄存器组包括的向量物理寄存器的第五数目大于所述第四数目。

9.根据权利要求7所述的方法,其特征在于,在所述第二向量长度大于所述第一向量长度的情况下,所述选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组之前,所述方法还包括:

确定当前的每个向量物理寄存器组包括的向量物理寄存器的第六数目小于所述第四数目。

10.根据权利要求7-9中任一项所述的方法,其特征在于,所述选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组,包括:

根据每个向量物理寄存器的标识,选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

11.根据权利要求7-10中任一项所述的方法,其特征在于,所述将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组之后,所述方法还包括:

对每个向量物理寄存器组分配组标识;

建立向量物理寄存器组的组标识和向量物理寄存器组的状态指示信息之间的对应关系,所述状态指示信息用于指示向量物理寄存器组是否空闲。

12.根据权利要求11所述的方法,其特征在于,所述由所述第一向量执行单元执行所述第二向量长度的向量运算,包括:

获取向量计算指令;

根据所述对应关系,在空闲的向量物理寄存器组中,确定所述向量计算指令对应的目的向量物理寄存器组;

通过所述第一向量执行单元执行所述向量计算指令,并将所述向量计算指令对应的计算结果写入所述目的向量物理寄存器组。

13.一种向量算力分配的装置,其特征在于,所述装置包括:

确定模块,用于确定系统向量长度由第一向量长度更新为第二向量长度;

向量算力分配模块,用于基于所述第二向量长度,在向量算力池中选取运算器,配置第一向量执行单元,所述第一向量执行单元为执行所述第二向量长度的一个向量执行单元,所述第一向量执行单元由至少一个运算器组成,所述至少一个运算器可计算的总数据长度大于或等于所述第二向量长度;

向量计算模块,用于由所述第一向量执行单元执行所述第二向量长度的向量运算。

14.一种处理器,其特征在于,所述处理器包括逻辑电路和供电电路,所述逻辑电路用于执行如上述权利要求1至权利要求12所述的向量算力分配的方法。

15.一种计算设备,其特征在于,所述计算设备包括处理器和存储器,所述处理器用于执行如上述权利要求1至权利要求12所述的向量算力分配的方法。

向量算力分配的方法、处理器和设备

[0001] 本公开要求于2023年06月19日提交的申请号为202310734151.9、发明名称为“一种调度方法”的中国专利申请的优先权,其全部内容通过引用结合在本公开中。

技术领域

[0002] 本公开涉及向量计算技术领域,特别涉及一种向量算力分配的方法、处理器和设备。

背景技术

[0003] 在高性能计算(High performance computing,HPC)以及人工智能(Artificial Intelligence,AI)计算等众多领域中,向量计算被广泛使用。

[0004] 目前,在执行向量计算的指令流水线中,会配置固定数量的向量执行单元(Vector Execution Unit,VEU),每个向量执行单元由固定数量的运算器组成,每个向量执行单元可以对不大于系统向量长度的向量进行计算。

[0005] 在用户将系统向量长度更新为远小于当前的系统向量长度的情况下,每个向量执行单元在执行向量计算时,都会有很大一部分运算器不会参与计算,向量执行单元无法充分发挥计算能力,对算力造成了浪费。

发明内容

[0006] 本公开提供了一种向量算力分配的方法、处理器和设备,能够提升向量执行单元的算力利用率,相应的技术方案如下:

[0007] 第一方面,提供了一种向量算力分配的方法,方法包括:运算器控制电路确定系统向量长度由第一向量长度更新为第二向量长度,基于第二向量长度,在向量算力池中选取运算器,配置第一向量执行单元,由第一向量执行单元执行所述第二向量长度的向量运算。其中,第一向量执行单元为执行第二向量长度的一个向量执行单元,第一向量执行单元由至少一个运算器组成,至少一个运算器可计算的总数据长度大于或等于第二向量长度。

[0008] 在本公开提供的技术方案中,系统向量长度是指计算设备所执行的向量计算中的向量长度,系统向量长度可以根据实际的计算需求进行设置,通过该技术方案,可以在系统向量长度更新后,根据更新后的系统向量长度,在向量算力池中选取运算器,重新配置向量执行单元,使得新配置的向量执行单元在能够执行向量计算的前提下,充分利用所包括的运算器,尽量避免运算器不参与计算,而造成的向量执行单元的算力浪费。

[0009] 在一种可能的实现方式中,在向量算力池中选取运算器,配置向量执行单元的处理可以如下:基于第二向量长度和单个运算器可计算的数据长度之间的倍数关系,确定第一向量执行单元包括的运算器的第一数目,在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元。

[0010] 在本公开提供的技术方案中,为了使配置的向量执行单元中包括的运算器既可以满足向量计算需求,又不会在执行向量计算时闲置,在确定向量执行单元中包括的运算器

数目时,先计算第二向量长度和单个运算器可计算的数据长度之间的倍数关系,再基于该倍数关系,确定第一向量执行单元包括的运算器的数目。

[0011] 在一种可能的实现方式中,在更新后的系统向量长度小于更新前的系统向量长度情况下,在确定出需要配置的第一向量执行单元包括的运算器的第一数目之后,可以先比较第二向量执行单元包括的运算器的第二数目和第一数目的大小,其中,第二向量执行单元为执行更新前的系统向量长度的向量执行单元。在确定第二数目大于第一数目的情况下,说明如果继续使用第二向量执行单元执行后续的向量计算的话,在向量计算的过程中,第二向量执行单元中会有运算器处于空闲状态,浪费算力,因此,在此情况下,可以在向量算力池中选取运算器,重新配置向量执行单元。

[0012] 在一种可能的实现方式中,在更新后的系统向量长度小于更新前的系统向量长度情况下,在确定出需要配置的第一向量执行单元包括的运算器的第一数目之后,可以先比较第二向量执行单元包括的运算器的第三数目和第一数目的大小,其中,第二向量执行单元为执行更新前的系统向量长度的向量执行单元。在确定第三数目小于第一数目的情况下,说明第二向量执行单元能够执行的向量长度无法支持后续的向量计算,因此,在此情况下,可以在向量算力池中选取运算器,重新配置向量执行单元。

[0013] 在一种可能的实现方式中,在向量算力池中选取运算器的处理可以如下:根据向量算力池中每个运算器的标识,在向量算力池中选取运算器,将选取的每第一数目个运算器配置为一个向量执行单元。

[0014] 在一种可能的实现方式中,在系统向量长度更新后,为了能够配合重新配置的向量执行单元,对计算结果进行存储,在本公开还可以对向量物理寄存器进行分组,方法如下:基于第二向量长度,选取向量物理寄存器,配置至少一个向量物理寄存器组,每个向量物理寄存器组包括至少一个向量物理寄存器,至少一个向量物理寄存器可存储的总数据长度大于或等于第二向量长度。

[0015] 在本公开提供的技术方案中,在系统向量长度更新后,可以根据更新后的系统向量长度,对向量物理寄存器进行分组,配置向量物理寄存器组,使得配置的向量物理寄存器组能够实现对后续的向量计算的计算数据和计算结果进行存储。

[0016] 在一种可能的实现方式中,选取向量物理寄存器,配置至少一个向量物理寄存器组的处理如下:基于第二向量长度和单个向量物理寄存器可存储的数据长度之间的倍数关系,确定每个向量物理寄存器组包括的向量物理寄存器的第四数目。选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

[0017] 在本公开提供的技术方案中,为了使配置的单个向量物理寄存器组中包括的向量物理寄存器即可以满足存储单次向量计算的计算结果的需求,又不会在存储单个计算结果时出现太多的剩余存储空间,在确定向量物理寄存器组包括的向量物理寄存器数目时,先计算第二向量长度和单个向量物理寄存器可存储的数据长度之间的倍数关系,再基于该倍数关系,确定单个向量物理寄存器组包括的向量物理寄存器的数目。

[0018] 在一种可能的实现中,在更新后的系统向量长度小于更新前的系统向量长度情况下,在确定出需要配置的每个向量物理寄存器组包括的向量物理寄存器的第四数目之后,可以先比较当前的每个向量物理寄存器组包括的向量物理寄存器的第五数目和第四数目的大小。在确定第五数目大于第四数目的情况下,说明如果继续使用当前已经配置好的向

量物理寄存器组存储执行后续向量计算的计算结果的话,在一个向量物理寄存器组存储一个计算结果时,在该向量物理寄存器组中可能会有较多的剩余存储空间,因此,在此情况下,可以重新配置向量物理寄存器组。

[0019] 在一种可能的实现中,在更新后的系统向量长度大于更新前的系统向量长度情况下,在确定出需要配置的每个向量物理寄存器组包括的向量物理寄存器的第四数目之后,可以先比较当前的每个向量物理寄存器组包括的向量物理寄存器的第六数目和第四数目的大小。在确定第六数目大于第四数目的情况下,说明当前已经配置好的单个向量物理寄存器组可能无法存储一次向量计算的计算结果,因此,在此情况下,可以重新配置向量物理寄存器组。

[0020] 在一种可能的实现中,选取向量物理寄存器配置向量物理寄存器的方法可以如下:根据每个向量物理寄存器的标识,选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

[0021] 在一种可能的实现中,为了对向量计算分配向量物理寄存器组,在本公开提供的技术方案中,还可以记录每个向量物理寄存器组是否空闲,处理可以如下:对每个向量物理寄存器组分配组标识,建立向量物理寄存器组的组标识和向量物理寄存器组的状态指示信息之间的对应关系,状态指示信息用于指示向量物理寄存器组是否空闲。

[0022] 在一种可能的实现中,在使用向量物理寄存器组存储计算结果的情况下,向量计算的处理流程可以如下:获取向量计算指令,根据向量物理寄存器组的组标识和向量物理寄存器组的状态指示信息之间的对应关系,在空闲的向量物理寄存器组中,确定向量计算指令对应的目的向量物理寄存器组。通过第一向量执行单元执行向量计算指令,并将向量计算指令对应的计算结果写入目的向量物理寄存器组。

[0023] 第二方面,本公开提供一种向量算力分配的装置,所述装置包括用于执行第一方面或第一方面任一种可能实现方式中的向量算力分配的方法的各个模块。

[0024] 第三方面,本公开提供了一种处理器,处理器包括逻辑电路和供电电路,逻辑电路用于执行如上述第一方面所述的向量算力分配的方法。

[0025] 第四方面,本公开提供了一种计算设备,计算设备包括处理器和存储器,处理器用于执行如上述第一方面所述的向量算力分配的方法。

[0026] 第五方面,本公开提供一种计算机可读存储介质,所述计算机可读存储介质中存储有指令,当其在计算设备上运行时,使得计算机执行上述各方面所述的方法。

[0027] 第六方面,本公开提供了一种包含指令的计算机程序产品,当其在计算设备上运行时,使得计算设备执行上述各方面所述的方法。

[0028] 本公开在上述各方面提供的实现方式的基础上,还可以进行进一步组合以提供更多实现方式。

附图说明

[0029] 图1是本公开提供的一种指令流水线的架构示意图;

[0030] 图2是本公开提供的一种向量算力分配的方法流程示意图;

[0031] 图3是本公开提供的一种向量算力分配的效果示意图;

[0032] 图4是本公开提供的一种向量算力分配的效果示意图;

[0033] 图5是本公开提供的一种向量算力分配的方法流程示意图;

[0034] 图6是本公开提供的一种向量算力分配的效果示意图;

[0035] 图7是本公开提供的一种向量计算的方法流程示意图;

[0036] 图8是本公开提供的一种处理器的结构示意图;

[0037] 图9是本公开提供的一种计算设备的结构示意图;

[0038] 图10是本公开提供的一种向量算力分配的装置结构示意图。

具体实施方式

[0039] 为了提升向量执行单元的利用率,本公开提供了一种向量算力分配的方法,在该方法中,处理器根据系统向量长度的变化,适应调整单个向量执行单元可计算的数据长度、以及向量执行单元的数量。这样,在系统向量长度较小时,可以配置更多的向量执行单元,提高向量计算的并行度,使运算器尽可能多的参与到向量计算中,从而避免了向量执行单元的算力浪费。

[0040] 本公开提供的向量算力分配的方法,可以应用于处理器中用于执行向量计算的指令流水线,其中,处理器可以部署在各类计算设备中,处理器可以为采用专用集成电路(application-specific integrated circuit,ASIC)、数字信号处理(digital signal processing,DSP)、现场可编程门阵列(field-programmable gate array,FPGA)、可编程逻辑阵列(programmable logic array,PLA)中的至少一种硬件形式来实现,当然,处理器也可以有其他的硬件实现方式,对此本公开不做限定。处理器,例如中央处理器(central processing unit,CPU)。在一些示例中,处理器还可以利用图像处理器(graphics processing unit,GPU)或数据处理单元(data processing unit,DPU)、片上系统(system on chip,SoC)、加速芯片或加速卡等方式实现。在一些示例中,处理器还可以为人工智能(artificial intelligence,AI)处理器。

[0041] 参见图1,指令流水线100可以包括译码单元(Decode)101、重命名单元(Rename)102、重排序缓冲区(ReOrder Buffer)103、发射单元(Issue)104、向量物理寄存器(Vector Physical registers)105、运算器106,多个运算器组成一个向量执行单元(Vector Execution Unit)1061,运算器可以执行浮点数加法、浮点数乘法、整数加法、整数乘法等运算。此外,在图1所示的架构下,还可以包括运算器控制电路107和向量物理寄存器控制电路108。其中,运算器控制电路107可以执行本公开提供的向量算力分配的方法,基于系统向量长度,对运算器进行分组,作为向量执行单元,向量寄存器控制电路108可以执行本公开提供的向量算力分配的方法,基于系统向量长度,对向量物理寄存器进行分组。

[0042] 下面结合附图对本公开提供的向量算力分配的方法进行说明。如图2所示,该方法的处理流程可以包括如下步骤:

[0043] 步骤201、运算器控制电路确定系统向量长度由第一向量长度更新为第二向量长度。

[0044] 其中,系统向量长度是指计算设备所执行的向量计算中的向量长度,系统向量长度可以由用户根据实际的计算需求进行设置。

[0045] 在实施中,用户可以根据实际的计算需求,更新系统向量长度。在更新系统向量长度时,用户可以向计算设备输入对系统向量长度的更新指令,相应的,处理器接收更新指

令,将系统向量长度由第一向量长度更新为该更新所指示的第二向量长度,运算器控制电路获取该第二向量长度,并确定系统向量长由第一向量长度更新为第二向量长度。

[0046] 在一种可能的实现中,在处理器中设置有可伸缩矢量扩展控制寄存器(Scalable Vector Extension Control Register,SVE Control Register),用户在指定系统向量长度时,可以向计算设备输入系统寄存器(Move(to)System Register,MSR)指令。处理器接收MSR指令,将SVE Control Register中的LEN字段由第一值,更新为该MSR指令所指示的第二值,LEN字段的值用于指示系统向量长度,第一值用于指示系统向量长度为第一向量长度,第二值用于指示系统向量长度为第二向量长度。运算器控制电路读取SVE Control Register中的LEN字段的第二值,并确定该第二值所指示的第二向量长度,并确定系统向量长度由第一向量长度更新为第二向量长度。

[0047] 步骤202、运算器控制电路基于第二向量长度,在向量算力池中选取运算器,配置至少一个向量执行单元,向量执行单元由至少一个运算器组成,至少一个运算器可计算的总数据长度大于或等于第二向量长度。

[0048] 其中,运算器可计算的数据类型可以为FP8类型、BF16、半精度浮点数(FP16)类型、单精度浮点数(FP32)类型、双精度浮点数(FP64)类型、INT8类型、短整数(INT16)类型、基本整数(INT32)类型、长整数(INT64)类型、BF16等。在运算器可计算的数据类型为INT8、FP8类型的情况下,单个运算器可计算的数据长度为8bit;在运算器可计算的数据类型为BF16、半精度浮点数类型、短整数类型的情况下,单个运算器可计算的数据长度为16bit;在运算器可计算的数据类型为单精度浮点数类型、基本整数类型的情况下,单个运算器可计算的数据长度为32bit;在运算器可计算的数据类型为双精度浮点数类型、长整数类型的情况下,单个运算器可计算的数据长度为64bit。此处仅以以上几种数据类型为例,对单个运算器可计算的数据长度进行示例性说明,本公开对于运算器可计算的数据类型以及相应的可计算的数据长度不做限定。

[0049] 在实施中,在向量算力池中包括有多个运算器,运算器控制电路在确定系统向量长度由第一向量长度更新为第二向量长度时,则可以基于第二向量长度和单个运算器可计算的数据长度,确定一个向量执行单元包括的运算器的数目,以及配置的向量执行单元的数目。然后,将指令流水线中发射单元的发射宽度更新为向量执行单元的数目。其中,发射宽度是指发射单元在一个时钟周期内允许向向量执行单元发射的向量计算指令的最大数目。

[0050] 对于配置多少个向量执行单元以及每个向量执行单元包括多少个运算器,可以有多种计算方法,满足每个向量执行单元可计算的总数据长度不小于第二向量长度即可,下面示例性的列举几种方法进行说明。

[0051] 方法一:

[0052] 基于向量算力池中的运算器可计算的总数据长度和第二向量长度的倍数关系,确定配置的向量执行单元的数目。然后,基于向量算力池中的运算器的数目和配置的向量执行单元的数目之间的倍数关系,确定一个向量执行单元包括的运算器的数目。具体的,可以采用如下计算公式实现。

[0053] 将向量算力池中的运算器的数目乘以单个运算器可计算的数据长度,得到第一数值。将第一数值除以系统向量长度,再对除法结果进行向上取整,得到配置的向量执行单元

的数目。将向量算力池中的运算器的数目除以配置的向量执行单元的数目,再对除法结果进行向上取整,得到一个向量执行单元包括的运算器的数目。计算公式可以如下公式(1):

[0054] $$N=\left\lceil\frac{E*L}{VL}\right\rceil$$
$$n=\left\lceil\frac{E}{N}\right\rceil \quad (1)$$

[0055] 其中,E为向量算力池中的运算器的数目,L为单个运算器可计算的数据长度,VL为第二向量长度,E*L为上述第一数值,N为配置的向量执行单元的数目,n为一个向量执行单元包括的运算器的数目,⌈ ⌉表示向上取整。

[0056] 下面通过一个示例对该方法一进行说明:

[0057] 运算器的数目为32个,单个运算器可计算的数据长度为64bit,系统向量长度为256bit。将运算器的数目乘以单个运算器可计算的数据长度,得到第一数值为2048。将第一数值除以系统向量长度256,得到第二数值为8。将运算器的数目除以第二数值,得到第三数值为4。即,将运算器分为8组,每组包括4个运算器,将每组运算器配置为一个向量执行单元,总共配置为8个向量执行单元。

[0058] 方法二:

[0059] 基于第二向量长度和单个运算器可计算的数据长度之间的倍数关系,确定一个向量执行单元包括的运算器的数目。然后,根据向量算力池中的运算器的数目和一个向量执行单元包括的运算器的数目之间的倍数关系,确定配置的向量执行单元的数目。具体的,可以采用如下计算公式实现。

[0060] 将第二向量长度除以单个运算器可计算的数据长度,再对除法结果进行向上取整,得到一个向量执行单元包括的运算器的数目。将向量算力池中的运算器的数目除以一个向量执行单元包括的运算器的数目,再对除法结果向上取整,得到配置的向量执行单元的数目。计算公式可以如下公式(2):

[0061] $$n=\left\lceil\frac{VL}{L}\right\rceil$$
$$N=\left\lceil\frac{E}{n}\right\rceil \quad (2)$$

[0062] 其中,E为向量算力池中的运算器的数目,L为单个运算器可计算的数据长度,VL为第二向量长度,E*L为上述第一数值,N为配置的向量执行单元的数目,n为一个向量执行单元包括的运算器的数目,⌈ ⌉表示向上取整。

[0063] 下面通过一个示例对该方法二进行说明:

[0064] 运算器的数目为32个,单个运算器可计算的数据长度为64bit,系统向量长度为256bit。将系统向量长度除以单个运算器可计算的数据长度,得到第三数值为4。将运算器的数目除以第三数值,得到第二数值为8。即,将运算器分为8组,每组包括4个运算器,将每组运算器配置为一个向量执行单元,总共配置为8个向量执行单元。

[0065] 在确定出配置的向量执行单元的数目以及一个向量执行单元包括的运算器的数目之后,比较当前的一个向量执行单元包括的运算器的数目和此处确定出的一个向量执行单元包括的运算器的数目,如果两个数目不相同,则可以根据运算器的标识,在向量算力池

中选取运算器,配置向量执行单元。此处,当前的向量执行单元即为执行上述第一向量长度的向量执行单元。

[0066] 运算器的数目为E个,运算器的标识分别为:C_1、C_2、C_3…C_E,将标识为C_1、C_2…C_n的n个运算器分为一组,配置为一个向量执行单元,将标识为C_{n+1}、C_{n+2}…C_{2n}的n个运算器分为一组,配置为一个向量执行单元,依次类推,将E个运算器划分为N组,配置为N个向量执行单元。

[0067] 例如,运算器的数目为32个,运算器的标识分别为:C_1、C_2、C_3…C_{32},将标识为C_1、C_2、C_3、C_4的4个运算器分为一组,配置为一个向量执行单元,将标识为C_5、C_6、C_7、C_8的4个运算器分为一组,配置为一个向量执行单元,依次类推,将32个运算器划分为8组,配置为8个向量执行单元。

[0068] 通过上述步骤201和步骤202,在系统向量长度由长变短的情况下,可以重新分配运算器,配置向量执行单元,从而可以适应调整向量执行单元的数量,增加发射宽度,提高向量计算并行度。

[0069] 下面通过一个示例对系统向量长度由长变短的情况下,采用本公开提供的方法所能达到的效果进行说明:

[0070] 如图3所示,运算器的数目为32个,单个运算器可计算的数据长度为64位(bit),更新前的系统向量长度为512bit,向量执行单元数量为4个,发射单元的发射宽度为4,每个向量执行单元包括8个运算器,单个向量执行单元可计算的数据长度为512bit。更新后的系统向量长度变为256bit,如果向量执行单元的配置不变,则发射宽度仍然为4,在执行向量计算时,会有一半的运算器不执行计算,浪费算力。通过上述步骤202,可以重新分配运算器,减少单个向量执行单元中运算器的数目,增加向量执行单元的数目,配置出8个向量执行单元,每个向量执行单元包括4个运算器,相应的,发射单元的发射宽度更新为8。这样,系统向量长度变短后,可以同时执行更多的向量计算指令,更加了向量计算并行度,避免了算力浪费,提高了整体计算效率。

[0071] 在系统向量长度由短变长的情况下,可以重新分配运算器,配置向量执行单元,从而可以适应调整向量执行单元可计算的数据长度,使向量执行单元可以一次执行对更长的向量的计算。

[0072] 下面通过一个示例对系统向量长度由短变长的情况下,采用本公开提供的方法所能达到的效果进行说明:

[0073] 如图4所示,运算器的数目为32个,单个运算器可计算的数据长度为64bit,更新前的系统向量长度为128bit,向量执行单元数量为16个,每个向量执行单元包括2个运算器,单个向量执行单元可计算的数据长度为128bit,发射单元的发射宽度为16。更新后的系统向量长度变为256bit,如果向量执行单元的配置不变,单个向量执行单元可计算的数据长度仍为128bit,则向量执行单元无法一次一个向量计算指令,需要分为多次执行,计算效率较低。通过上述步骤202,可以重新分配运算器,增加单个向量执行单元中运算器的数目,配置出8个向量执行单元,发射单元的发射宽度更新为8,每个向量执行单元包括4个运算器,单个向量执行单元可计算的数据长度为256bit。这样,系统向量长度变长后,每个向量执行单元可计算的数据长度也会适应变长,使向量执行单元可以一次执行完一个向量计算指令,无需分为多次执行,提高了计算效率。

[0074] 在完成运算器的分组后,运算器控制电路可以将向量执行单元的配置信息发送至发射单元。向量执行单元的配置信息可以包括向量执行单元的数目、向量执行单元的标识、每个向量执行单元包括的运算器的标识。示例性的,向量执行单元的配置信息可以如下表1所示。

[0075] 表1

[0076]

向量执行单元的数目	向量执行单元的标识	向量执行单元包括的运算器的标识
N	U_1	C_1、C_2...C_n
	U_2	C_{n+1}、C_{n+2}...C_{2n}
	U_3	C_{2n+1}、C_{2n+2}...C_{3n}
	.....	.....

[0077] 当然,上述步骤201和202也可以由处理器中的其他硬件实现,如发射单元,本公开对具体的执行主体不做限定。

[0078] 在配置完向量执行单元后,可以由配置的向量执行单元来执行上述第二向量长度的向量计算。

[0079] 在一种可能的实现中,还可以根据系统向量长度,对向量物理寄存器进行分组,分组后,可以使用一个向量物理寄存器组来存储向量,相比于使用单个向量物理存储器存储向量,使用向量物理寄存器组存储向量,可以存储长度更长的向量。相应的,参见图5,对向量物理寄存器分组的处理可以包括如下步骤:

[0080] 步骤203、向量物理寄存器控制电路确定系统向量长度由第一向量长度更新为第二向量长度。

[0081] 该步骤203中向量物理寄存器控制电路的具体处理和上述步骤201中运算器控制电路的具体处理相同,在此不再赘述。

[0082] 步骤204、向量物理寄存器控制电路基于上述第二向量长度,选取向量物理寄存器,配置至少一个向量物理寄存器组,每个向量物理寄存器组包括至少一个向量物理寄存器,至少一个向量物理寄存器可存储的总数据长度大于或等于上述第二向量长度。

[0083] 其中,单个向量物理寄存器可存储的数据长度是指一个向量物理寄存器能够存储的最大数据长度。

[0084] 在实施中,向量物理寄存器控制电路在确定系统向量长度由第一向量长度更新为第二向量长度时,可以基于第二向量长度和单个向量物理寄存器可存储的数据长度,确定一个向量物理寄存器组包括的向量物理寄存器的数目,以及配置的向量物理寄存器组的数目。

[0085] 对于配置多少个向量物理寄存器组以及每个向量物理寄存器组包括多少个向量物理寄存器,可以有多种计算方法,满足每个向量物理寄存器组可存储的总数据长度大于或等于系统向量长度即可,下面示例性的列举几种方法进行说明:

[0086] 方法一:

[0087] 基于可用的向量物理寄存器可存储的总数据长度和第二向量长度的倍数关系,确定配置的向量物理寄存器组的数目。然后,基于可用的向量物理寄存器的数目和配置的向量物理寄存器组的数目之间的倍数关系,确定一个向量物理寄存器组包括的向量物理寄存器的数目。具体的,可以采用如下计算公式实现。

[0088] 将向量物理寄存器的数目乘以单个向量物理寄存器可存储的数据长度,得到第二数值。将第二数值除以第二向量长度,再对除法结果进行向上取整,得到配置的向量物理寄存器组的数目。将可用的向量物理寄存器的数目除以配置的向量物理寄存器组的数目,再对除法结果进行向上取整,得到一个向量物理寄存器组包括的向量物理寄存器的数目。计算公式可以如下公式(3):

[0089]
$$M=\left\lceil\frac{R*l}{VL}\right\rceil \qquad m=\left\lceil\frac{R}{N}\right\rceil \tag{3}$$

[0090] 其中,R为可用的向量物理寄存器的数目,1为单个向量物理寄存器可存储的数据长度,VL为第二向量长度,R*L为上述第而数值,M为配置的向量物理寄存器组的数目,m为一个向量物理寄存器组包括的向量物理寄存器的数目。

[0091] 下面通过一个示例对该方法一进行说明:

[0092] 系统向量长度为256bit,单个向量物理寄存器可存储的数据长度为128bit,向量物理寄存器的数目为256个。将向量物理寄存器的数目乘以单个向量物理寄存器可存储的数据长度,得到第四数值为32768。将第四数值除以系统向量长度,得到第五数值128。将向量物理寄存器的数目除以第五数值,得到第六数值为2。将向量物理寄存器划分为128个向量物理寄存器组。每个向量物理寄存器组包括2个向量物理寄存器。

[0093] 方法二:

[0094] 基于第二向量长度和单个向量物理寄存器可存储的数据长度之间的倍数关系,确定一个向量物理寄存器组包括的向量物理寄存器的数目。然后,根据可用的向量物理寄存器的数目和一个向量物理寄存器组包括的向量物理寄存器的数目之间的倍数关系,确定配置的向量物理寄存器组的数目。具体的,可以采用如下计算公式实现。

[0095] 将第二向量长度除以单个向量物理寄存器可存储的数据长度,再对除法结果进行向上取整,得到一个向量物理寄存器组包括的向量物理寄存器的数目。将可用的向量物理寄存器的数目除以一个向量物理寄存器组包括的向量物理寄存器的数目,再对除法结果向上取整,得到配置的向量物理寄存器组的数目。计算公式可以如下公式(4):

[0096]
$$m=\left\lceil\frac{VL}{l}\right\rceil \qquad M=\left\lceil\frac{R}{m}\right\rceil \tag{4}$$

[0097] 其中,R为可用的向量物理寄存器的数目,1为单个向量物理寄存器可存储的数据长度,VL为第二向量长度,R*L为上述第而数值,M为配置的向量物理寄存器组的数目,m为一个向量物理寄存器组包括的向量物理寄存器的数目。

[0098] 下面通过一个示例对该方法二进行说明:

[0099] 向量物理寄存器的数目为256个,单个向量物理寄存器可存储的数据长度为128bit,系统向量长度为256bit。将系统向量长度除以单个运算器可计算的数据长度,得到第六数值为2。将向量物理寄存器的数目除以第六数值,得到第五数值为128。即,将向量物理寄存器分为128组,每个向量物理寄存器组包括2个向量物理寄存器。

[0100] 在确定出配置的向量物理寄存器组的数目以及一个向量物理寄存器组包括的向

量物理寄存器的数目之后,比较当前的一个向量物理寄存器组包括的向量物理寄存器的数目和此处确定出的一个向量物理寄存器组包括的向量物理寄存器的数目,如果两个数目不相同,则可以根据向量物理寄存器组的标识,选取向量物理寄存器,配置向量物理寄存器组。

[0101] 向量物理寄存器的数目为R个,向量物理寄存器的标识分别为:V_1、V_2、V_3…V_R,将标识为V_1、V_2…V_m的m个向量物理寄存器分为一组,作为一个向量物理寄存器组,将标识为V_{m+1}、V_{m+2}…V_{2m}的m个运算器分为一组,作为一个向量物理寄存器组,依次类推,将R个运算器划分为M组。

[0102] 例如,向量物理寄存器的数目为256个,向量物理寄存器的标识分别为:V_1、V_2、V_3…V_{256},将标识为V_1、V_2的2个运算器分为一组,作为一个向量物理寄存器组,将标识为V_3、V_4的2个向量物理寄存器分为一组,作为一个向量物理寄存器组,依次类推,将256个向量物理寄存器划分为128组。

[0103] 通过上述步骤203,在系统向量长度由短变长的情况下,如果单个向量物理寄存器无法存放一个向量,则可以将多个向量物理寄存器分为一组,使用一个向量物理寄存器组来存放一个向量,此处的向量可以是指向量计算的计算结果。

[0104] 下面通过示例对上述效果进行说明:

[0105] 如图6所示,向量物理寄存器的数目为256个,单个向量物理寄存器可存储的数据量为128bit,更新前的系统向量长度为128bit,一个向量物理寄存器可以存储一个向量。更新后的系统向量长度为256bit,一个向量物理寄存器无法存储一个向量。通过上述步骤203,将2个向量物理寄存器组成一个向量物理寄存器组,可以组成128个向量物理寄存器组,一个向量物理寄存器组来存储一个向量,每个向量物理寄存器组可存储的数据长度为256bit。这样,可以使指令流水线可以支持向量长度更长的向量计算。

[0106] 在完成向量物理寄存器分组后,向量物理寄存器控制电路可以将向量物理寄存器的分组信息发送给重命名单元。向量物理寄存器的分组信息可以包括向量物理寄存器组的数目、每个向量物理寄存器组包括的向量物理寄存器的标识、每个向量物理寄存器组的组标识。示例性的,如下向量物理寄存器的分组信息可以如下表2所示:

[0107] 表2

[0108]

向量物理寄存器组的数目	向量物理寄存器组的组标识	向量物理寄存器组包括的向量物理寄存器的标识
R	G_1	V_1、V_2...V_m
	G_2	V_{m+1}、V_{m+2}...V_{2m}
	G_3	V_{2m+1}、V_{2m+2}...V_{3m}
	……	……

[0109] 当然,上述步骤203和204也可以由处理器中的其他硬件实现,如重命名单元,本公开对具体的执行主体不做限定。

[0110] 在上述向量算力分配的基础上,本公开还提供了一种向量计算的方法,参见图7,该方法可以包括如下处理步骤:

[0111] 步骤301、译码单元对代码进行译码,得到至少一组向量操作信息。

[0112] 在实施中,译码单元获取代码,对代码进行译码,得到至少一组向量操作信息,其

中,每组向量操作信息包括向量计算指令、每个计算数据的架构寄存器的标识(Architected Register Number,ARN)和计算结果的架构寄存器的标识。向量计算指令可以为微操作指令(Micro operation code,μop)。计算数据是指参与向量计算的数据,例如,向量计算为A+B,则向量A和向量B为计算数据。

[0113] 步骤302、译码单元向重命名单元发送上述至少一组向量操作信息。

[0114] 步骤303、对于每个向量计算指令,重命名单元确定该向量计算指令对应的目的向量物理寄存器组。

[0115] 其中,目的向量物理寄存器组用于存储向量计算指令的计算结果。

[0116] 在实施中,在重命名单元中或者重排序缓冲区中,记录有重命名列表,该重命名列表中包括架构寄存器的标识和向量物理寄存器组的组标识之间的对应关系,该对应关系是可以动态变化的。对于接收到的每组向量操作信息,重命名单元获取该向量操作信息中的每个计算数据的架构寄存器的标识,并在上述重命名列表中,查询出每个计算数据的架构寄存器的标识对应的原向量物理寄存器组的组标识。然后,在查询出的原向量物理寄存器组中,读取计算数据。

[0117] 在重命名单元中或者重排序缓冲区中,还记录有物理向量寄存器组的状态列表,在状态列表中包括向量物理寄存器组的组标识和向量物理寄存器组的状态指示信息之间的对应关系。对于接收到的每组向量操作信息中的向量计算指令,重命名单元根据该向量计算指令和上述至少一个向量操作信息中的其他向量计算指令之间的相关性,在状态列表中,选择一个状态为空闲的物理向量寄存器组,作为该向量计算指令对应的目的向量物理寄存器组。

[0118] 此外,为了在向量计算指令乱序执行的场景下,可以按照向量计算指令在代码中的顺序返回计算结果,重命名单元可以按照向量计算指令在代码中的顺序,将上述至少一个向量操作信息中的向量计算指令,发送给重排序缓冲区。重排序缓冲区对每个向量计算指令分配ROB id,ROB id用于指示对应的向量计算指令在代码中的顺序。

[0119] 步骤304、重命名单元向发射单发送向量计算指令、向量计算指令的计算数据和向量计算指令对应的目的向量物理寄存器组的组标识。

[0120] 在一种可能的实现中,在重排缓冲区对向量计算指令分配了ROB id的情况下,重命名单元还可以获取每个向量计算指令对应的ROB id,并将向量计算指令的ROB id和向量计算指令一起发送给发射单元。

[0121] 步骤305、发射单元向向量执行单元发送向量计算指令、向量计算指令的计算数据和向量计算指令对应的目的向量物理寄存器组的组标识。

[0122] 在实施中,发射单元获取向量执行单元的状态,确定处于空闲状态的向量执行单元。具体的,发射单元可以在接收到向量计算指令时,根据向量执行单元的配置信息,确定出每个向量执行单元包括的运算器,然后,获取向量执行单元包括的运算器的状态,如果向量执行单元包括的运算器均为空闲状态,则确定该向量执行单元为空闲状态。

[0123] 对于每个向量计算指令,发射单元将该向量计算指令、该向量计算指令的计算数据和该向量计算指令对应的目的向量物理寄存器组的组标识,发送至一个空闲的向量执行单元。在有多个空闲状态的向量执行单元的情况下,发射单元可以同时发送向这多个空闲状态的向量执行单元分别发送向量计算指令。

[0124] 在一种可能的实现中,在重排缓冲区对向量计算指令分配了ROB id的情况下,重命名单元还可以将向量计算指令的ROB id和向量计算指令一起发送给向量执行单元。

[0125] 步骤306、向量执行单元根据向量计算指令,对计算数据进行计算,得到计算结果,并将计算结果写入向量计算指令对应的目的向量物理寄存器组。

[0126] 在实施中,向量执行单元中包括多个运算器,发射单元可以将向量计算指令的两个计算数据,分为多组元素,将一组元素发送给一个运算器,由该运算器执行这组元素的运算,得到元素计算结果。例如,向量计算指令所指示的向量运算为向量加法,计算数据包括A和B,其中,A为向量:(A1,A2,A3,A4),B为向量:(B1,B2,B3,B4),向量执行单元中包括四个运算器,则A1和B1作为一组元素,A2和B2作为一组元素,A3和B3作为一组元素,A4和B4作为一组元素,四个运算器分别计算A1+B1,A2+B2,A3+B3,A4+B4。

[0127] 运算器将元素计算结果按序发送至向量物理寄存器控制电路,同时,运算器还会将向量执行单元对应的目的向量物理寄存器组的组标识,发送至向量物理寄存器控制电路。向量物理寄存器控制电路中记录有向量物理寄存器的分组信息,向量物理寄存器控制电路查询目的向量物理寄存器组包括的向量物理寄存器,并将计算结果写入目的向量物理寄存器组包括的向量物理寄存器中。

[0128] 在一种可能的实现中,在重排缓冲区对向量计算指令分配了ROB id的情况下,向量执行单元还可以将向量计算指令对应的ROB id和向量计算指令的计算结果一起发送给向量物理寄存器控制电路。相应的,向量物理寄存器控制电路可以按照ROB id,将向量计算指令的计算结果依次写入对应的物理向量寄存器组中。

[0129] 图8是本公开提供的一种处理器的结构示意图,如图8所示,处理器800包括逻辑电路801和供电电路802。其中,逻辑电路801可以包括至少一个如图1所示的指令流水线。供电电路802用于向逻辑电路801供电。

[0130] 在一种可能的实现中,处理器1801可以是CPU或其他通用处理器,处理器1801还可以是一个或多个用于实现本公开方案的集成电路,例如,数字信号处理器(digital signal processing,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、可编程逻辑器件(programmable logic device,PLD)、现场可编程门阵列(Field Programmable Gate Array,FPGA)、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者是任何常规的处理器等。

[0131] 图9是本公开提供的一种计算设备的结构示意图,如图9所示,计算设备900包括:总线902、处理器904、存储器906以及通信接口908。处理器904、存储器906和通信接口908之间通过总线902通信。总线902可以是外设部件互连标准(peripheral component interconnect,PCI)总线或扩展工业标准结构(extended industry standard architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,图9中仅用一条线表示,但并不表示仅有一根总线或一种类型的总线。总线902可包括在计算设备900各个部件(例如,存储器906、处理器904、通信接口908)之间传送信息的通路。处理器904可以包括CPU、图形处理器(graphics processing unit,GPU)、微处理器(micro processor,MP)或者数字信号处理器(digital signal processor,DSP)等处理器中的任意一种或多种。存储器906可以作为计算设备900的内存或者外存。存储器906可以包括易失性存储器(volatile memory),例如随机存取存储器(random access memory,RAM)。存储器

906还可以包括非易失性存储器(non-volatile memory),例如只读存储器(read-only memory,ROM),快闪存储器,机械硬盘(hard disk drive,HDD)或固态硬盘(solid state drive,SSD)。

[0132] 图10是本公开提供的一种向量算力分配的装置,如图10所示,装置1000包括确定模块1001、运算器分配模块1002、向量计算模块1003,其中:

[0133] 确定模块1001,用于确定系统向量长度由第一向量长度更新为第二向量长度;

[0134] 运算器分配模块1002,用于基于所述第二向量长度,在向量算力池中选取运算器,配置第一向量执行单元,所述第一向量执行单元为执行所述第二向量长度的一个向量执行单元,所述第一向量执行单元由至少一个运算器组成,所述至少一个运算器可计算的总数据长度大于或等于所述第二向量长度;

[0135] 向量计算模块1003,用于执行所述第二向量长度的向量运算。

[0136] 应理解的是,本申请提供的装置可以通过中央处理单元(central processing unit,CPU)实现,也可以通过专用集成电路(application-specific integrated circuit,ASIC)实现,或可编程逻辑器件(programmable logic device,PLD)实现,上述PLD可以是复杂程序逻辑器件(complex programmable logical device,CPLD),现场可编程门阵列(field-programmable gate array,FPGA),通用阵列逻辑(generic array logic,GAL),也可以通过数据处理单元(data processing unit,DPU)、片上系统(system on chip,SoC)、卸载卡或其任意组合。也可以通过可以由上述图1中的运算器控制电路实现,或者,图8所示的处理器或图9所示的计算设备实现。还可以通过软件实现上述算力分配方法时,装置1000及其各个模块也可以为软件模块。

[0137] 在一种可能的实现方式中,所述运算器分配模块1002,用于:

[0138] 基于所述第二向量长度和单个运算器可计算的数据长度之间的倍数关系,确定所述第一向量执行单元包括的运算器的第一数目;

[0139] 在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元。

[0140] 在一种可能的实现方式中,在所述第二向量长度小于所述第一向量长度的情况下,所述在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元之前,所述运算器分配模块1002,还用于:

[0141] 比较第二向量执行单元包括的运算器的第二数目和所述第一数目的大小,确定所述第二数目大于所述第一数目,其中,所述第二向量执行单元为执行所述第一向量长度的一个向量执行单元。

[0142] 在一种可能的实现方式中,在所述第二向量长度大于所述第一向量长度的情况下,所述在向量算力池中选取运算器,将选取的第一数目个运算器配置为第一向量执行单元之前,所述运算器分配模块1002,还用于:

[0143] 比较第二向量执行单元包括的运算器的第三数目和所述第一数目的大小,确定所述第三数目小于所述第一数目,其中,所述第二向量执行单元为执行所述第一向量长度的一个向量执行单元。

[0144] 在一种可能的实现方式中,所述运算器分配模块1002,用于:

[0145] 根据向量算力池中每个运算器的标识,在所述向量算力池中选取运算器,将选取

的第一数目个运算器配置为第一向量执行单元。

[0146] 在一种可能的实现方式中,所述装置还包括向量物理寄存器分配模块,用于:

[0147] 基于所述第二向量长度,选取向量物理寄存器,配置至少一个向量物理寄存器组,每个向量物理寄存器组包括至少一个向量物理寄存器,所述至少一个向量物理寄存器可存储的总数据长度大于或等于所述第二向量长度。

[0148] 在一种可能的实现方式中,所述向量物理寄存器分配模块,用于:

[0149] 基于所述第二向量长度和单个向量物理寄存器可存储的数据长度之间的倍数关系,确定每个向量物理寄存器组包括的向量物理寄存器的第四数目;

[0150] 选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

[0151] 在一种可能的实现方式中,在所述第二向量长度小于所述第一向量长度的情况下,所述向量物理寄存器分配模块,还用于:

[0152] 确定当前的每个向量物理寄存器组包括的向量物理寄存器的第五数目大于所述第四数目。

[0153] 在一种可能的实现中,在所述第二向量长度大于所述第一向量长度的情况下,所述选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组之前,所述方法还包括:

[0154] 确定当前的每个向量物理寄存器组包括的向量物理寄存器的第六数目小于所述第四数目。

[0155] 在一种可能的实现方式中,所述向量物理寄存器分配模块,用于:

[0156] 根据每个向量物理寄存器的标识,选取向量物理寄存器,将选取的每第四数目个向量物理寄存器配置为一个向量物理寄存器组。

[0157] 在一种可能的实现方式中,所述向量物理寄存器分配模块,还用于:

[0158] 对每个向量物理寄存器组分配组标识;

[0159] 建立向量物理寄存器组的组标识和向量物理寄存器组的状态指示信息之间的对应关系,所述状态指示信息用于指示向量物理寄存器组是否空闲。

[0160] 在一种可能的实现方式中,所述第一向量执行单元还用于:

[0161] 获取向量计算指令;

[0162] 根据所述对应关系,在空闲的向量物理寄存器组中,确定所述向量计算指令对应的目的向量物理寄存器组;

[0163] 通过所述第一向量执行单元执行所述向量计算指令,并将所述向量计算指令对应的计算结果写入所述目的向量物理寄存器组。

[0164] 向量算力分配的装置1000应用于上述图8和9中的处理器,装置1000在执行向量算力分配时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将向量算力分配的装置1000的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,向量算力分配的装置1000与上述向量算力分配的方法属于同一构思,其具体实现过程详见上述向量算力分配的方法流程,这里不再赘述。

[0165] 最后应说明的是:以上实施例仅用以说明本公开的技术方案,而非对其限制;尽管

参照前述实施例对本公开进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本公开各实施例技术方案的保护范围。

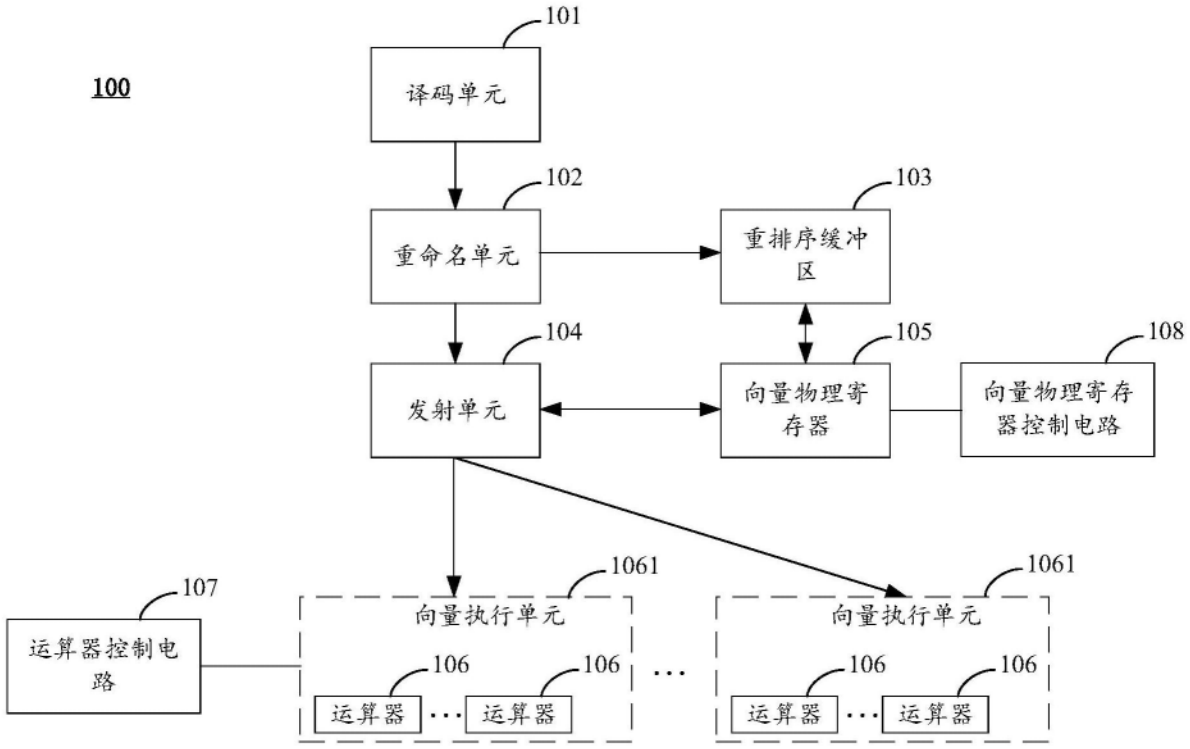

100
101
译码单元
102
重命名单元
103
重排序缓冲区
104
发射单元
105
向量物理寄存器
108
向量物理寄存器控制电路
107
运算器控制电路
1061
向量执行单元
106
运算器
1061
向量执行单元
106
运算器

图1

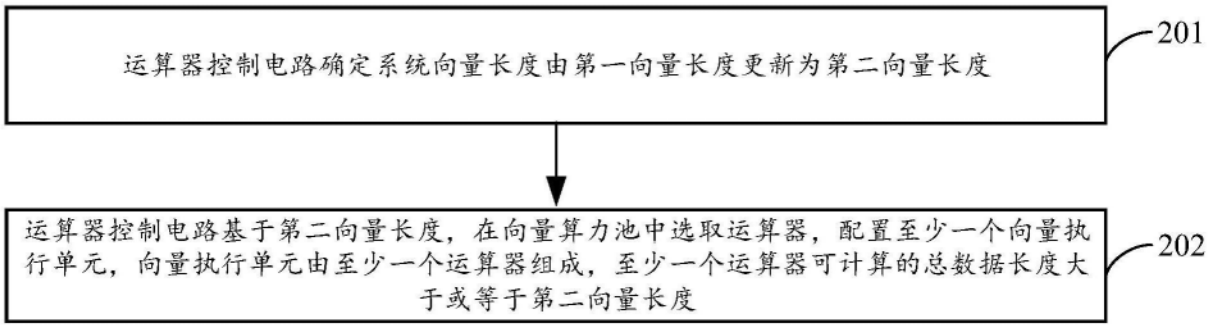

运算器控制电路确定系统向量长度由第一向量长度更新为第二向量长度
201
运算器控制电路基于第二向量长度，在向量算力池中选取运算器，配置至少一个向量执行单元，向量执行单元由至少一个运算器组成，至少一个运算器可计算的总数据长度大于或等于第二向量长度
202

图2

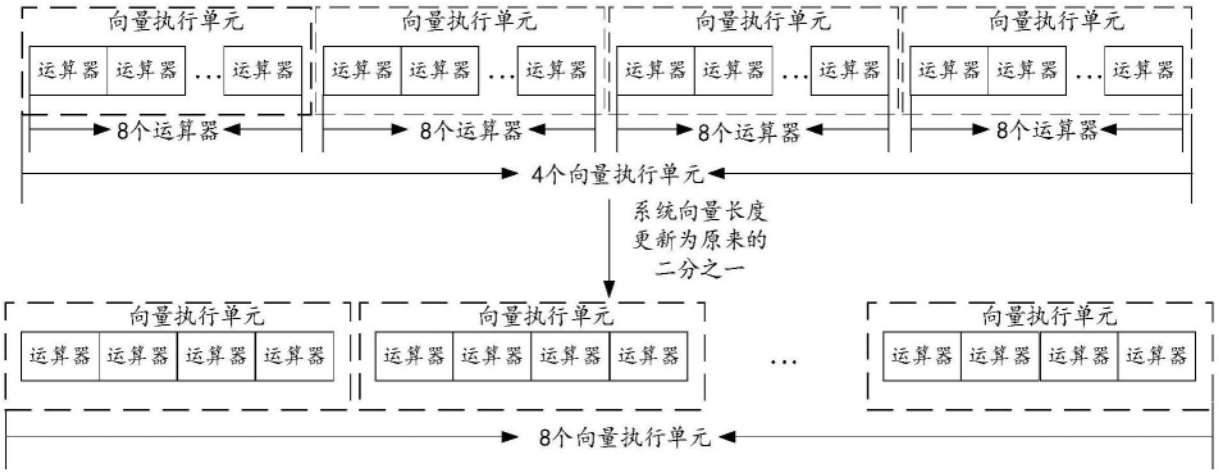

向量执行单元
运算器
8个运算器
4个向量执行单元
系统向量长度
更新为原来的
二分之一
8个向量执行单元

图3

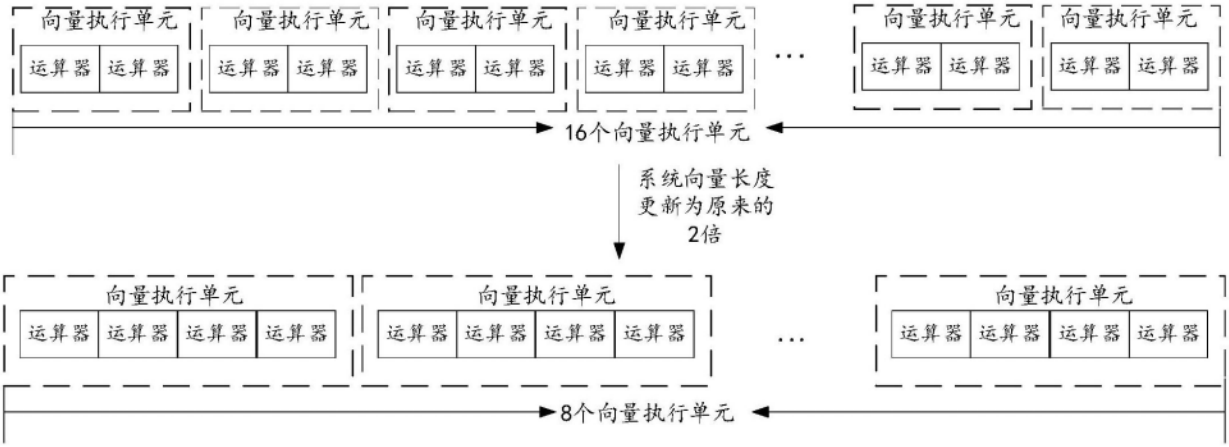

向量执行单元
运算器
16个向量执行单元
系统向量长度
更新为原来的
2倍
8个向量执行单元

图4

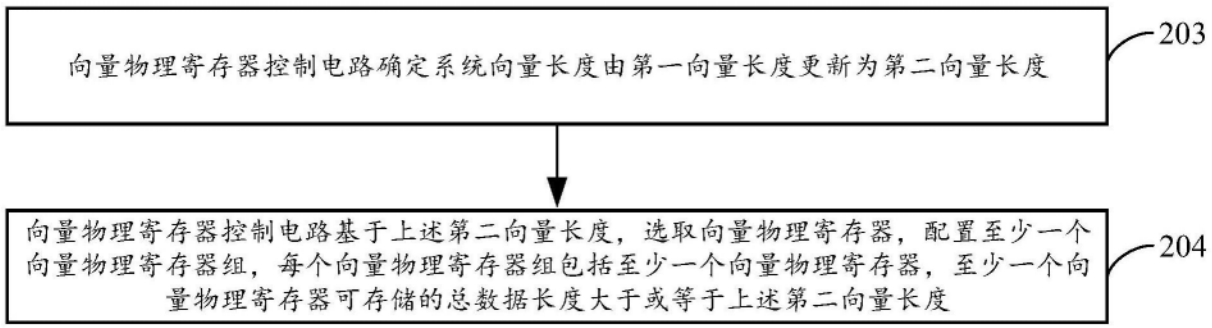

向量物理寄存器控制电路确定系统向量长度由第一向量长度更新为第二向量长度
203
向量物理寄存器控制电路基于上述第二向量长度，选取向量物理寄存器，配置至少一个向量物理寄存器组，每个向量物理寄存器组包括至少一个向量物理寄存器，至少一个向量物理寄存器可存储的总数据长度大于或等于上述第二向量长度
204

图5

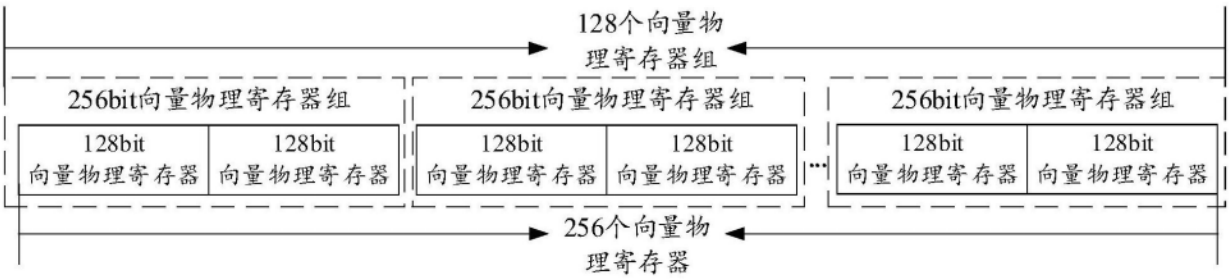

128个向量物理寄存器组
256bit向量物理寄存器组
128bit向量物理寄存器
128bit向量物理寄存器
256bit向量物理寄存器组
128bit向量物理寄存器
128bit向量物理寄存器
256bit向量物理寄存器组
128bit向量物理寄存器
128bit向量物理寄存器
256个向量物理寄存器

图6

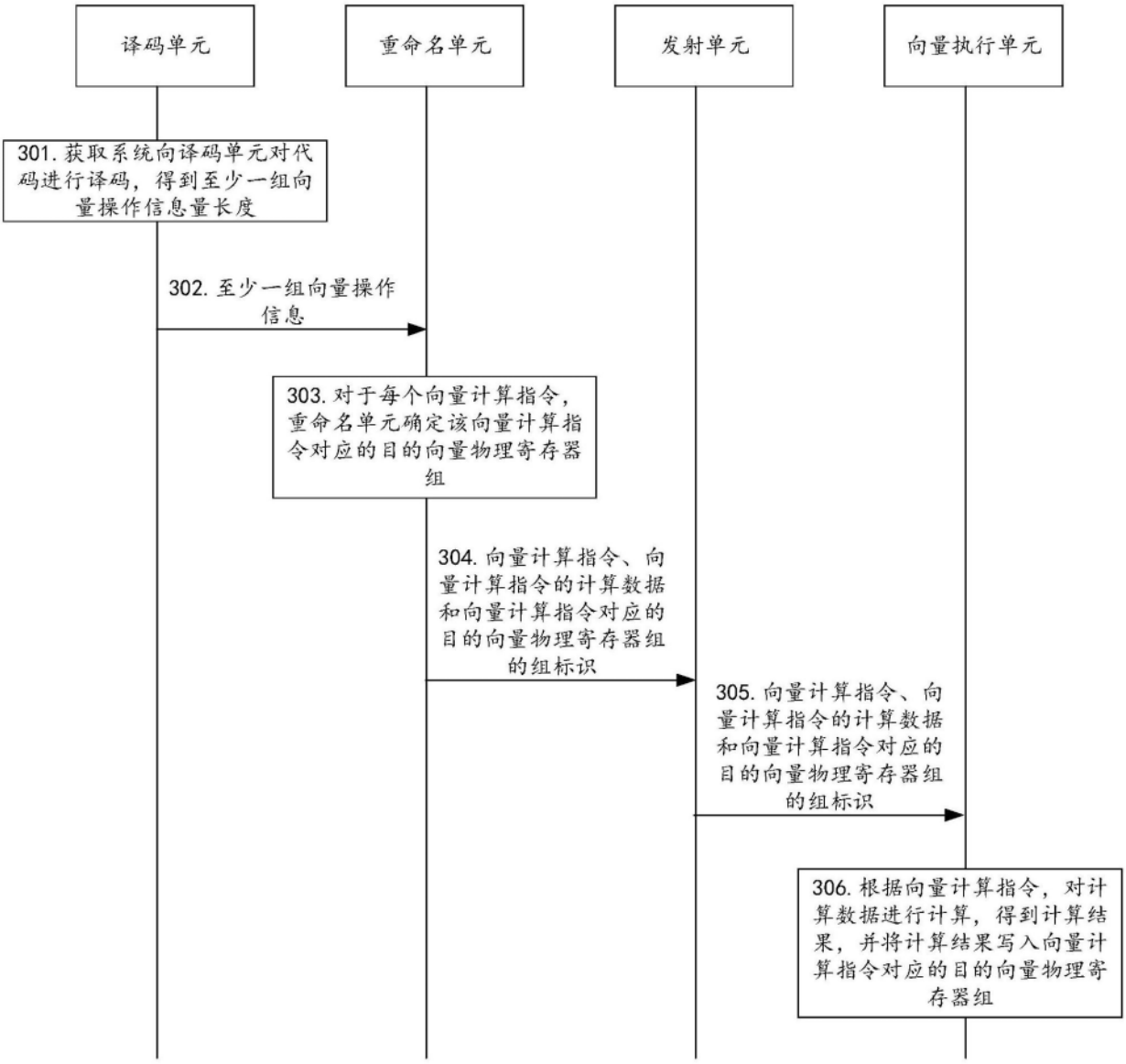

译码单元
重命名单元
发射单元
向量执行单元
301. 获取系统向译码单元对代码进行译码，得到至少一组向量操作信息量长度
302. 至少一组向量操作信息
303. 对于每个向量计算指令，重命名单元确定该向量计算指令对应的目的向量物理寄存器组
304. 向量计算指令、向量计算指令的计算数据和向量计算指令对应的目的向量物理寄存器组的组标识
305. 向量计算指令、向量计算指令的计算数据和向量计算指令对应的目的向量物理寄存器组的组标识
306. 根据向量计算指令，对计算数据进行计算，得到计算结果，并将计算结果写入向量计算指令对应的目的向量物理寄存器组

图7

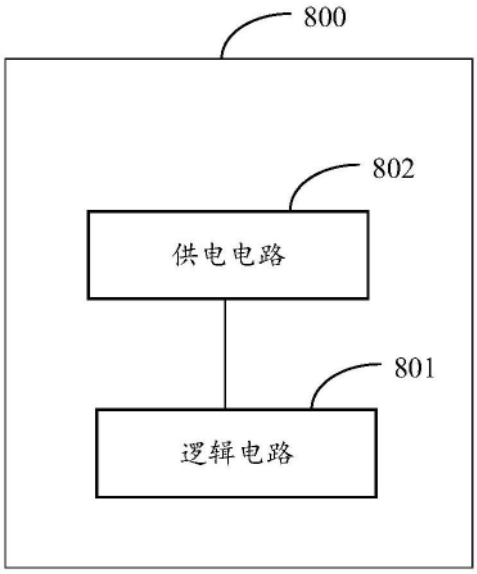


图8

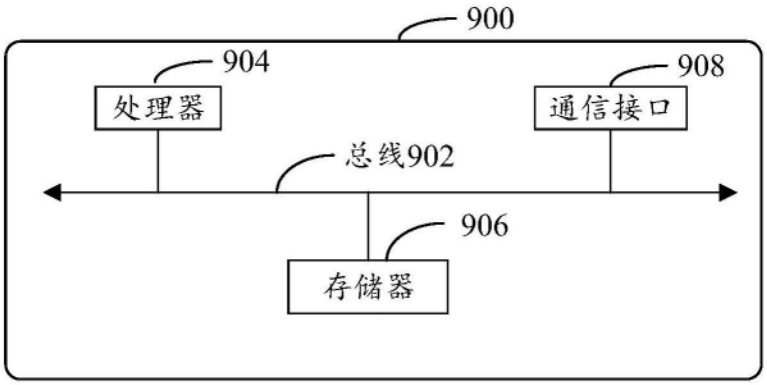


图9

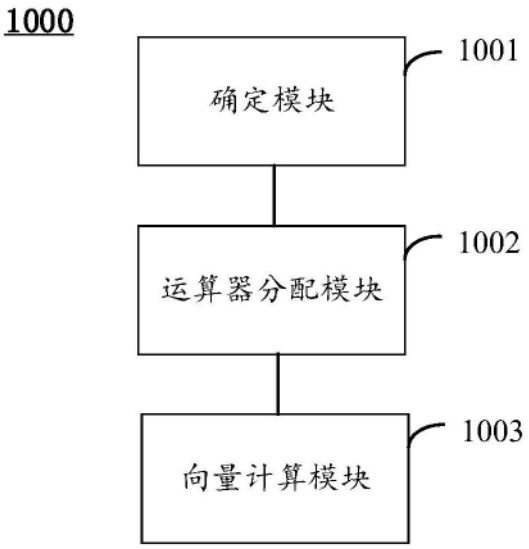


图10