

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2023 年 12 月 21 日 (21.12.2023)



(10) 国际公布号
WO 2023/241418 A1

- (51) 国际专利分类号:
G06F 9/30 (2018.01) *G06F 9/355* (2018.01)
G06F 9/305 (2018.01) *G06N 3/063* (2023.01)
G06F 9/345 (2018.01) *G06N 3/08* (2023.01)
- (21) 国际申请号: PCT/CN2023/098716
- (22) 国际申请日: 2023 年 6 月 6 日 (06.06.2023)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202210674857.6 2022年6月14日 (14.06.2022) CN
- (71) 申请人: 北京有竹居网络技术有限公司 (BEIJING YOUZHUJU NETWORK TECHNOLOGY CO., LTD.)
- [CN/CN]; 中国北京市平谷区林荫北街 13 号信息大厦 802 室, Beijing 101299 (CN)。
- (72) 发明人: 曹宇辉 (CAO, Yuhui); 中国北京市朝阳区七圣中街 12 号院融中心 B1 小邮局, Beijing 100028 (CN)。施云峰 (SHI, Yunfeng); 中国北京市朝阳区七圣中街 12 号院融中心 B1 小邮局, Beijing 100028 (CN)。王剑 (WANG, Jian); 中国北京市朝阳区七圣中街 12 号院融中心 B1 小邮局, Beijing 100028 (CN)。
- (74) 代理人: 北京世辉律师事务所 (SHIHUI PARTNERS); 中国北京市朝阳区建国门外大街 2 号北京银泰中心 C 座 42 层, Beijing 100022 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU,

(54) Title: PROCESSOR, METHOD FOR DATA PROCESSING, DEVICE, AND STORAGE MEDIUM

(54) 发明名称: 处理器以及用于数据处理的方法、设备和存储介质

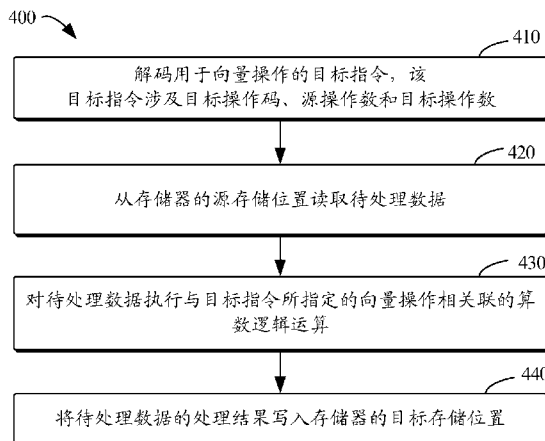


图 4

- 410 Decode a target instruction for a vector operation, the target instruction relating to a target opcode, a source operand, and a target operand
- 420 Read data to be processed from a source storage location in a memory
- 430 Perform arithmetic logic operation associated with the vector operation specified by the target instruction on said data
- 440 Write a processing result of said data into a target storage location in the memory

(57) Abstract: Embodiments of the present disclosure provide a processor, a method for data processing, a device, and a storage medium. The processor comprises an instruction decoder configured to decode a target instruction for a vector operation. The target instruction relates to a target opcode, a source operand, and a target operand. The target opcode indicates the vector operation specified by the target instruction. The source operand specifies a source storage location in a memory for reading data to be processed. The target operand specifies a target storage location in the memory for writing a processing result. The processor further comprises an

CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

arithmetic logic unit coupled to the instruction decoder and the memory. The arithmetic logic unit is configured to: read said data from the source storage location in the memory; perform arithmetic logic operation associated with the vector operation specified by the target instruction on said data; and write the processing result into the target storage location in the memory. In this way, the efficiency of vector computation can be improved.

(57) 摘要: 根据本公开的实施例, 提供了一种处理器以及用于数据处理的方法、设备和存储介质。该处理器包括指令解码器, 被配置为解码用于向量操作的目标指令。目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作。源操作数指定存储器中的用于读取待处理数据的源存储位置。目标操作数指定存储器中的用于写入处理结果的目标存储位置。该处理器还包括算术逻辑单元, 被耦合至指令解码器和存储器。算术逻辑单元被配置为: 从存储器的源存储位置读取待处理数据; 对待处理数据执行与目标指令所指定的向量操作相关联的算术逻辑运算; 以及将处理结果写入存储器的目标存储位置。以此方式, 可以提高向量计算的效率。

处理器以及用于数据处理的方法、设备和存储介质

5 本申请要求 2022 年 06 月 14 日递交的，标题为“处理器以及用于数据处理的方法、设备和存储介质”、申请号为 202210674857.6 的中国发明专利申请的优先权，该申请的全部内容通过引用结合在本申请中。

10 技术领域

本公开的示例实施例总体涉及计算机领域，特别地涉及处理器以及用于数据处理的方法、设备和计算机可读存储介质。

背景技术

15 随着信息技术的发展，各类处理器可以被应用于多种多样的场景。针对各种应用场景，目前已经提出了由处理器可采用的不同的指令集架构（ISA）。这些指令集架构往往需要兼容各种各样的使用场景。对于一些指令重复性高且数据量大的向量计算，需要更好的指令集架构来使处理器能够更好地处理这样的向量计算。

20

发明内容

在本公开的第一方面，提供了一种处理器。该处理器包括指令解码器，被配置为解码用于向量操作的目标指令。目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作。源操作数指定存储器中的用于读取待处理数据的源存储位置。目标操作数指定存储器中的用于写入处理结果的目标存储位置。该处理器还包括算数逻辑单元，被耦合至指令解码器和存储器。算数逻辑单元被配置为：从存储器的源存储位置读取待处理数据；对待处理数据执行与目标指令所指定的向量操作相关联的算数逻辑运算；以

25

在本公开的第二方面，提供了一种用于数据处理的方法。该方法包括解码用于向量操作的目标指令。目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作。源操作数指定存储器中的用于读取待处理数据的源存储位置。目标操作数指定存储器中的用于写入处理结果的目标存储位置。该方法还包括从存储器的源存储位置读取待处理数据；对待处理数据执行与目标指令所指定的向量操作相关联的算数逻辑运算；以及将处理结果写入存储器的目标存储位置。

在本公开的第三方面，提供了一种电子设备。该电子设备至少包括根据第一方面的处理器。

在本公开的第四方面，提供了一种计算机可读存储介质。计算机可读存储介质上存储有计算机程序，计算机程序可由处理器执行以实现第二方面的方法。

应当理解，该内容部分中所描述的内容并非旨在限定本公开的实施例的关键特征或重要特征，也不用于限制本公开的范围。本公开的其他特征将通过以下的描述而变得容易理解。

附图说明

结合附图并参考以下详细说明，本公开各实施例的上述和其他特征、优点及方面将变得更加明显。在附图中，相同或相似的附图标记表示相同或相似的元素，其中：

图 1 示出了本公开的实施例能够在其中实现的示例环境的示意图；

图 2 示出了根据本公开的一些实施例的示例指令的示意图；

图 3 示出了根据本公开的一些实施例的示例源操作数对应的存储位置的示意图；

图 4 示出了根据本公开的一些实施例的用于数据处理的过程的流程图；以及

器的电子设备的框图。

具体实施方式

5 下面将参照附图更详细地描述本公开的实施例。虽然附图中示出了本公开的某些实施例，然而应当理解的是，本公开可以通过各种形式来实现，而且不应该被解释为限于这里阐述的实施例，相反，提供这些实施例是为了更加透彻和完整地理解本公开。应当理解的是，本公开的附图及实施例仅用于示例性作用，并非用于限制本公开的保护范围。

10 在本公开的实施例的描述中，术语“包括”及其类似用语应当理解为开放性包含，即“包括但不限于”。术语“基于”应当理解为“至少部分地基于”。术语“一个实施例”或“该实施例”应当理解为“至少一个实施例”。术语“一些实施例”应当理解为“至少一些实施例”。下文还可能包括其他明确的和隐含的定义。

15 可以理解的是，本技术方案所涉及的数据（包括但不限于数据本身、数据的获取或使用）应当遵循相应法律法规及相关规定的要求。

20 如前所述，随着信息技术的发展，各类处理器可以被应用于多种多样的场景。针对各种应用场景，目前已经提出了由处理器可采用的不同的指令集架构。这些指令集架构往往需要兼容各种各样的使用场景。然而，这些常规的指令集架构的使用场景与诸如神经网络计算等的向量计算的使用场景并不一致。因此，对于一些指令重复性高且数据量大的向量计算，需要更好的指令集架构来使处理器能够更好地处理这样的向量计算。

25 一种常规的方案是采用标准的处理器指令集，诸如精简指令集计算机（RISC）-V 指令集等。尽管这些通用的指令集能够完成各种向量计算，诸如各类神经网络算子等，但是由于这些通用指令集需要兼容各种各样的使用场景，因此难以保证较高的执行效率。例如，神经网络算子计算时通常涉及到大量向量计算，这对于通用指令集并不友

经研究发现，针对某些大量的向量计算，常规方案的指令集架构并不适用。例如，常规的方案可以使用数字信号处理器（DSP）架构，例如单指令多数据（SIMD）架构，或者可以使用向量处理器（Vector）架构。然而，上述 DSP 架构的指令集通常不公开。而对于向量处理器架构，例如 RISC-V 标准下的向量指令集（简称为 RISC-V 向量指令集），这些指令集通常复杂度较高，对诸如神经网络算子等向量计算显得冗余。

综上，对于一些指令重复性高且数据量大的向量计算，需要设计出更适用于向量计算的指令集，以提高处理器的计算效率。

根据本公开的实施例，提出了一种用于处理器的改进方案。在该方案中，处理器包括指令解码器和算数逻辑单元。指令编码器用于接收用于处理向量操作的目标指令。该目标指令适用于存储器到存储器（MEM to MEM）的处理器架构。例如，该目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作，源操作数至少指定存储器中的用于读取待处理数据的源存储位置，目标操作数至少指定存储器中的用于写入处理结果的目标存储位置。

处理器的算数逻辑单元被耦合至指令解码器和存储器。该算数逻辑单元被配置为根据指令解码器对目标指令的解码信息来执行该目标指令的向量操作。例如，算数逻辑单元被配置为从存储器的源存储位置读取的待处理数据；对待处理数据执行与目标指令所指定的向量操作相关联的算数逻辑运算；以及将待处理数据的处理结果写入存储器的目标存储位置。

本方案通过采用适用于存储器到存储器架构的处理器，来简化处理器的操作。以此方式，处理器能够使用简单的指令集来完成大量的向量计算。例如，处理器能够使用简单的指令集来进行神经网络向量计算。通过这样，本方案能够采用简单的指令集来提高处理器执行向量计算的效率。

意图。在该环境 100 中，处理器 110 可以表示任何种类的指令处理装置。例如，处理器 110 可以是通用处理器或者其他任何适当的处理器。处理器 110 被配置为接收指令 140 并且执行该指令 140 所指示的操作，
5 诸如向量操作。例如，处理器 110 可以从环境 100 中的其他设备接收指令 140。在一些实施例中，指令 140 是 SIMD 指令。

处理器 110 包括指令解码器 120 和算术逻辑单元 130。备选地或附加地，处理器 110 还可以包括存储器（未示出）或者通信地耦合到存储器。例如，存储器可以是数据存储器（诸如，向量紧耦合存储器
10 （Vector Closely-coupled Memory, VCCM））。指令解码器 120、算术逻辑单元 130 和存储器可通信地耦合。也即，指令解码器 120、算术逻辑单元 130 和存储器可以根据适当的数据传输协议和/或标准实现彼此通信。在运行中，指令解码器 120 接收指令 140，并且对指令 140 进行解码。例如，指令解码器 120 可以将指令 140 解码为可以由
15 算术逻辑单元 130 进行处理的算术运算和/或逻辑运算等。指令解码器 120 可以使用各种不同的机制来实现。例如，指令解码器 120 可以使用硬件电路来实现，或者至少部分地借助于软件模块来实现。

算术逻辑单元 130 被配置为基于由指令解码器 120 对指令 140 进行解码所获得的信息来进行操作。算术逻辑单元 130 可以执行各类算
20 数操作、逻辑操作，等等。算术逻辑单元 130 可以使用各种不同的机制来实现。例如，算术逻辑单元 130 可以使用硬件电路来实现，或者至少部分地借助于软件模块来实现。

应理解，仅出于示例性的目的描述环境 100 的结构和功能，而不暗示对于本公开的范围的任何限制。例如，处理器 110 可以被应用于
25 各种现有的或将来的计算平台或计算系统中。处理器 110 可以在各种嵌入式应用（例如，移动网络基站等的数据处理系统）中实现，以提供诸如大量向量计算等服务。处理器 110 也可以被集成或嵌入到各种电子设备或计算设备中，以提供各种计算服务。处理器 110 的应用环境和应用场景在此不受限制。

在本文中，指令 140 有时也被称为“目标指令”，二者在本上下文中可以互换使用。图 2 示出了根据本公开的一些实施例的示例指令 140 的示意图。如图 2 所示，指令 140 包括目标操作码 210、源操作数 220 和目标操作数 230。在本文中，目标操作码 210 有时也被称为“操作码（opcode）”，二者在本上下文中可以互换使用。目标操作码 210 可以指示指令 140 所指定的向量操作。源操作数 220 至少用于指定存储器中的用于读取待处理数据的源存储位置。目标操作数 230 至少用于指定存储器中的用于写入处理结果的目标存储位置。

在一些实施例中，指令解码器 120 对目标操作码 210、源操作数 220 和目标操作数 230 所指示的上述信息进行解码，以供算数逻辑单元 130 进行处理。例如，算数逻辑单元 130 被配置为从由源操作数 220 指定的存储器中的源存储位置读取待处理数据。算数逻辑单元 130 对待处理数据执行与指令 140 所指定的向量操作相关联的算数逻辑运算。算数逻辑单元 130 进而将待处理数据的处理结果写入目标操作数 230 所指定的目标存储位置。

在一些实施例中，指令 140 可以使用诸如二进制来编码。在其他实施例中，指令 140 可以使用其他编码形式或者其他进制来编码。在本文中，除非特别说明，以下所描述的指令 140 的编码格式和编码表示均以二进制为例。例如，二进制形式的指令 140 可以采用如下的表 1 中的格式来定义。

表 1 指令格式定义

位 (bit)	[95:86]	[85:7 0]	[69:6 0]	[59:54]	[53:3 8]	[37:2 8]	[27:22]	[21:6]	[5:0]
--------------------	---------	-------------	-------------	---------	-------------	-------------	---------	--------	-------

	操作码	ddr	dex	A_vm	ddr	dex	B_vm	dr	C_vm
--	-----	-----	-----	------	-----	-----	------	----	------

如表 1 所示，第 86 位至第 95 位用于表示指令 140 的目标操作码 210。第 22 位至第 85 位的各个操作数用于表示指令 140 的源操作数 220。第 0 位至第 21 位的各个参数用于表示指令 140 的目标操作数 230。当然，应当理解的是，除非特别说明，否则在这里以及本文其他地方出现的任何具体数值或者位数都是示例性的。例如，以上所列举的各个操作码和/或操作数所位于的位数是示例性的，而不是限制性的。指令 140 的目标操作码 210、源操作数 220 和目标操作数 230 可以位于其他适当的位数处。

例如，第 70 位至第 85 位的源操作数 A_vaddr 用于表示存储器，例如数据存储器（诸如，VCCM）内的 A 通道（也被称为存储器的第一存储空间）的数据的地址索引，即 VCCM[A_vaddr]的地址索引。该地址索引以一个向量单词（vector word）为单位。向量单词可以表示存储器内的通道的存储单元，该存储单元以 SIMD 的宽度为宽度。即，该地址索引以一个 SIMD 宽度为单位。在一些实施例中，存储器的深度例如为 1024。在这一示例中，可以仅使用第 70 位至第 85 位中的 10 个位来表示 A 通道的地址索引。当然，应当理解，存储器可以具有其他适当的深度，地址索引也可以具有其他适当的位数。

又如，第 60 位至第 69 位的源操作数 A_index 用于表示存储器内的 A 通道的向量单词的元素索引。每个向量单词可以有例如 64 个元素。该元素索引可以用于指示 A 通道内的向量单词中的某个元素。在一些实施例中，如果 A 通道的向量单词被划分为例如 64 个元素，则可以仅使用第 60 位至第 69 位中的 6 个位来表示 A_index。再如，第 54 位至第 59 位的源操作数 A_vm 用于表示 A 通道的向量掩码（VM）寄存器的索引。在一些实施例中，A 通道具有 16 个 VM 寄存器。在这样的示例中，可以仅使用第 54 位至第 59 位的中的 4 个位来表示 A_vm。

(例如, 数据存储器 VCCM) 内的 B 通道 (也被称为存储器的第二存储空间) 的地址索引, 即 VCCM[B_vaddr] 的地址索引。该地址索引以一个向量单词为单位。即, 该地址索引以一个 SIMD 宽度为单位。

- 5 第 28 位至第 37 位的源操作数 B_index 用于表示 B 通道的向量单词的元素索引。第 22 位至第 27 位的源操作数 B_vm 用于表示 B 通道的向量掩码寄存器的索引。

表 1 中目标操作数 230 的示例包括第 6 位至第 21 位的 C_vaddr 可以表示数据存储器 VCCM 的 C 通道的地址索引, 即 VCCM[C_vaddr]
10 的地址索引。该地址索引以一个向量单词为单位。即, 该地址索引以一个 SIMD 宽度为单位。目标操作数 230 的示例还包括第 0 位至第 5 位的 C_vm, 其表示 C 通道的向量掩码寄存器的索引。

图 3 示出了根据本公开的一些实施例的示例源操作数对应的存储位置的示意图。在图 3 的示例中, 存储器的存储空间被划分为多个通道, 诸如通道 310-1、通道 310-2、... ...、通道 310-N 等, 其中 N 为
15 大于 1 的整数。为便于讨论, 下文中将通道 310-1、通道 310-2、... ...、通道 310-N 统称为或单独称为通道 310。在一些实施例中, N 的数值可以是预先设置的。例如, N 可以被设置为 1024、512 等不同的数值。各个通道 310 包括例如 1024 位或其他适当的位数。地址索引 330 (例
20 如, 源操作数 A_vaddr、B_vaddr 或者目标操作数 C_vaddr) 可以指示通道 310-1 的地址。通道 310 的地址可以以向量单词为单位。地址索引 330 可以是 16 位的。例如, 如果地址索引 330 为
“0b0000_0000_0000_0000”, 则该地址索引 330 可以指示通道 310-1。又如, 在一些实施例中, 在存储器的深度或者通道的数目为 1024 的
25 示例中, 地址索引 330 也可以是 10 位的, 例如由地址“0b00_0000_0000”来指示通道 310-1。注意, 本文中以“0b”开头的编码表示均表示二进制表示, 下文中将不再进行重复说明。二进制表示中出现的“_”仅为了便于查看, 而不具有实际含义, 并且不占据二进制位。

在一些实施例中, 各个通道 310 的向量单词可以被划分为多个元

如，源操作数 A_index 或者 B_index) 可以指示某个元素，例如元素 320 的索引。元素索引 340 可以是 10 位的。例如，元素索引 340 为 “0b00_0000_0000”，则该元素索引 340 可以指示元素 320。又如，
 5 在各个通道 310 的向量单词内的元素数目为 64 的示例中，元素索引可以是 6 位的，例如元素索引 “0b00_0000” 可以指示元素 320。

应当理解的是，除非特别说明，否则在这里以及本文其他地方出现的任何具体数值、位数以及二进制表示都是示例性的。例如，在其他实施例中，各个通道可以具有不同的位数，各个向量单词也可以采用不同的位数。相应地，地址索引和元素索引也可以具有不同的位数和不同的编码表示。本公开的范围在此方面不受限制。

以上参考表 1 列举了源操作数 220 的若干示例。下面将参考表 2 描述源操作数 220 的更多的示例。

表 2 指令格式定义

位	[95:86]	[85:54]	[53:22]	[21:6]	[5:0]
	操作 码	A_imm	B_imm	C_vaddr	C_vm

如表 2 所示，源操作数 220 可以包括位于第 54 至第 85 位的 A_imm，其表示指令 140 中的一个立即数。类似地，源操作数 220 还可以包括位于第 22 至第 53 位的 B_imm，其表示指令 140 中的另一个立即数。与表 1 类似，表 2 中的目标操作数 230 也可以包括 C_vaddr 和/或 C_vm。

应理解，以上结合表 1 和表 2 所描述的各个源操作数和/或各个目标操作数仅仅是示例性的，而不是限制性的。本公开所采用的源操作数 220 和/或目标操作数 230 可以包括以上任意一种或多种源操作数和/或目标操作数。在一些实施例中，源操作数 220 和/或目标操作数

的操作数类型。

以下的表 3 描述了指令 140 的操作码的示例编码方式。例如，如果第 0 位为 0，则表示该指令 140 为变量类型。如果第 0 位为 1，则表示该指令 140 为立即数类型。第 1 位至第 2 位表示指令 140 的子功能编码。第 3 位至第 7 位表示指令 140 的功能编码。第 8 位至第 9 位表示指令 140 的计算精度。例如，二进制“00”可以表示计算精度为单精度浮点数。其他二进制数值可以表示预留的其他计算精度。

表 3 示例操作码编码方式

长度	2 位	5 位	2 位	1 位
范围	9:8	7:3	2:1	0:0
名称	计算精度	功能编码	子功能编码	立即数/变量
描述	2'b00: fp32 其他: 预留	功能编码	子功能编码	0: 变量 1: 立即数

当然，应当理解的是，表 3 所示出的指令 140 的操作码的编码方式仅仅是示例性的，而不是限制性的。例如，在其他实施例中，可以使用其他编码方式来对指令 140 进行编码。

在一些实施例中，可以根据指令 140 的目标操作码 210 来确定指令 140 所指定的向量操作。例如，处理器 110 可以预先存储各个指令的操作码。指令解码器 120 可以根据接收到的指令 140 的目标操作码 210 来确定该指令 140 所指定的向量操作。例如，如果指令 140 的目标操作码 210 被编码为“0b00_00110_01_0”，则指令解码器 120 可以将该指令 140 确定为 v2indexr 指令。应理解，以上所列举的操作码和指令类型的示例仅仅是示例性的，而不是限制性的。具有被编码

以下将描述指令 140 以及处理器 110 对执行 140 的示例执行方式的若干示例。在一些实施例中，源操作数 220 可以包括两个源操作数，例如 A_vaddr 与 B_vaddr，或者 A_vaddr 与 B_imm。每个源操作数的宽度可以是 SIMD 宽度。备选地或附加地，在一些实施例中，源操作数 220 可以仅包括一个源操作数 B_vaddr 等。目标操作数 230，例如 C_vaddr 可以指定将处理结果写回存储器的目标存储位置，即 VCCM[C_vaddr]处。

在一些实施例中，指令 140 的目标存储位置处包括处理结果向量。目标操作数 230 还指示目标 VM 寄存器，例如 C_vm 或者 vm3。目标 VM 寄存器的各个位置处的值指示处理结果向量的相应位置处是否要被写入相应的处理结果。例如，如果目标寄存器 vm3[i]为 1，则表示处理结果向量单词的第 i 个元素是写使能的，可以被写入相应的处理结果。反之，如果目标寄存器 vm3[i]为 0，则处理结果向量单词的第 i 个元素不能被写入相应的处理结果。

表 4 描述了处理器 110 可以支持的若干示例指令。表 4 的指令可以参考表 1 或表 2 的指令定义来描述，并且可以参考表 3 的示例编码方式进行编码。在表 4 的示例中，目标操作数 230 均包括 C_vaddr(即，&v3)和 C_vm(即，vm3)。表 4 中的预留(reserved)表示预留的一个或多个位。这些预留的位可以在后续被编码或者使用。

表 4 示例指令

[95:8 6]	[85:7 0]	[69:6 0]	[59:5 4]	[53:3 8]	[37:2 8]	[27:2 2]	[21:6]	[5:0]
操作 码	A_va ddr	A_ind ex	A_vm	B_va ddr	B_ind ex	B_vm	C_va ddr	C_vm
v2ind exl	&v1	预留	预留	&v2	index 2	预留	&v3	vm3

exli	&v1	预留	预留	imm2			&v3	vm3
v2indexr	&v1	预留	预留	&v2	index 2	预留	&v3	vm3
v2indexri	&v1	预留	预留	imm2			&v3	vm3
sindex2v	&v1	预留	预留	&v2	index 2	预留	&v3	vm3
s2v	预留	预留	预留	&v2	index 2	预留	&v3	vm3
vexp	&v1	预留	预留	预留	预留	预留	&v3	vm3
vm2index	预留	预留	vm1	预留	预留	预留	&v3	vm3
vinde x2vm	预留	预留	预留	&v2	index 2	预留	预留	vm3
vtranspose	&v1	index 1	vm1	预留	预留	预留	&v3	vm3

作为一个实施例，指令 140 包括第一索引确定指令（例如，表 4 中的 v2indexl 或 v2indexr）。在这一示例中，源操作数 220 指定存储器的第一存储空间的位置，即，通道 A 的地址索引（A_vaddr 为 &v1）。

- 5 源操作数 220 还指定存储器的第二存储空间内的待处理数据的给定索引值，即通道 B 的向量单词内的元素索引（B_vaddr 为 &v2，B_index 为 index2）。在这一示例中，算术逻辑单元 130 被配置为：确定第一索引。该第一索引指示待处理数据中的由给定索引值所指示的位置处的值在第一存储空间中的存储位置。

“0b00_00110_00_0”，指令 v2indexl v1, v2, index2, v3, vm3 表示将 v3[i] 赋值为 indext，其中 indext 是能够使得 v1[indext] 等于 v2[index2] 的从左至右的第一个元素的索引。如果没有元素满足上述条件，则将 indext 设置为由二进制补码表示的“-1”。在一些实施例中，目标操作数 230 还指示目标向量掩码寄存器。目标向量掩码寄存器的各个位置处的值指示处理结果向量的相应位置处是否要被写入相应的处理结果。例如，如果 vm3[i] 等于 1，则 v3[i] 是写使能的。

又如，指令 v2indexr 的操作码可以被编码表示为 “0b00_00110_01_0”。指令 v2indexr v1, v2, index2, v3, vm3 表示将 v3[i] 赋值为 indext，其中 indext 是能够使得 v1[indext] 等于 v2[index2] 的从右至左的第一个元素的索引。如果没有元素满足上述条件，则将 indext 设置为由二进制补码表示的“-1”。在这一示例中，如果 vm3[i] 等于 1，则 v3[i] 是写使能的。

作为另一实施例，指令 140 包括第二索引确定指令（例如，表 4 中的 v2indexli 或 v2indexri）。在这一示例中，源操作数 220 指定存储器的第一存储空间的位置，即，通道 A 的地址索引（A_vaddr 为 &v1）。源操作数 220 还指定第一立即数，即，立即数 imm2。在这一示例中，算数逻辑单元被配置 130 确定第二索引。该第二索引指示第一立即数在第一存储空间中的存储位置。

例如，指令 v2indexli 的操作码被编码表示为 “0b00_00110_00_1”。指令 v2indexli v1, imm2, v3, vm3 表示将 v3[i] 赋值为 indext，其中 indext 是能够使得 v1[indext] 等于 imm2 的从左至右的第一个元素的索引。如果没有元素满足上述条件，则将 indext 设置为由二进制补码表示的“-1”。在这一示例中，如果 vm3[i] 等于 1，则 v3[i] 是写使能的。

又如，指令 v2indexri 的操作码可以被编码表示为 “0b00_00110_01_1”。指令 v2indexri v1, imm2, v3, vm3 表示将 v3[i] 赋值为 indext，其中 indext 是能够使得 v1[indext] 等于 imm2 的从右至左的第一个元素的索引。如果没有元素满足上述条件，则将 indext 设

则 $v3[i]$ 是写使能的。

作为另一示例，指令 140 可以包括第一数值确定指令，例如表 4 中的指令 `sindex2v`。源操作数 220 指定存储器的第一存储空间的位置即，通道 A 的地址索引（`A_vaddr` 为 $\&v1$ ）。源操作数 220 还指定存储器的第二存储空间内的待处理数据的给定索引值，即通道 B 的向量单词内的元素索引（`B_vaddr` 为 $\&v2$ ，`B_index` 为 `index2`）。

在这一示例中，算数逻辑单元 130 被配置为：确定待处理数据在由给定索引值所指示的位置处的给定值，并且确定第一存储空间中的在以给定值为索引的位置处的第一值。例如，指令 `sindex2v v1, v2, index2, v3, vm3`，具有编码为“0b00_00110_10_0”的操作码。该指令表示将 $v3[i]$ 赋值为 $v1[v2[index2]]$ 。如果 $vm3[i]$ 等于 1，则 $v3[i]$ 是写使能的。

在一些实施例中，指令 140 包括第二数值确定指令。在这一示例中，源操作数 220 指定存储器的第二存储空间内的待处理数据的给定索引值，即 `B_vaddr` 为 $\&v2$ ，`B_index` 为 `index2`。算数逻辑单元 130 被配置为：确定待处理数据中的由给定索引值所指示的位置处的第二值。例如，指令 `s2v v2, index2, v3, vm3`，具有编码为“0b00_00110_10_1”的操作码。该指令表示将 $v3[i]$ 赋值为 $v2[index2]$ 。如果 $vm3[i]$ 等于 1，则 $v3[i]$ 是写使能的。

通过使用以上描述的诸如第一索引确定指令、第二索引确定指令、第一数值确定指令和第二数值确定指令中的一个或多个，处理器 110 可以更好地处理一些诸如求取坐标的算子，例如求最大值的索引（`ArgMax`）算子、求最小值的索引（`ArgMin`）算子或者求最高排序的 K 个值（`TopK`）算子等算子。以 `ArgMax` 为例，其用于求使得向量 v 中的值 $v[index]$ 为最大值的索引。64 个元素的 `ArgMax` 所需的指令如下：首先，`v2smax v1, vm1, v2, vm2`（该指令将在下文的表 5 和表 6 中进行描述），该指令求 $v1$ 中的最大的元素值，并将其写入 $v2$ ，其中， $vm1$ 和 $vm2$ 存储值全部位为 1；接下来，`v2indexl v1, v2, 0, v3`,

v3, 其中 vm3 存储值的全部位为 1。

5 在一些实施例中, 目标指令包括向量转置指令, 例如指令 vtranspose 或 vstranspose。源操作数 220 指定存储器中的第一存储空间中的第一位置, 即 A_vaddr 为 &v1, A_index 为 index1。源操作数 220 还指定源向量掩码寄存器 vm1 和可选的 vm2。在该示例中, 算数逻辑单元 130 被配置为: 将第一存储空间中的第一位置处的待处理数据进行向量转置, 以获得转置后的待处理数据。

10 例如, 向量转置指令 vtranspose v1, index1, vm1, vm2, v3, vm3, 具有编码表示为 0b00_00111_11_0 的操作码, 其用于将例如 32*32 的向量(或矩阵)进行转置。在该示例中, vm1 和 vm2 的值为读通道(lane)使能; vm3 的值为写通道使能。vm1 中连续为 1 的位的数目 R 用于表示矩阵的行数, vm2 中连续为 1 的位的数目 C 用于表示矩阵的列数, 其中 R 和 C 均为任意自然数, R 与 C 可以相同也可以不同。vm1、vm2 15 和 vm3 的有效位必须是连续的, 否则以最低位的第一个 1 为准。上述向量转置指令 vtranspose 可以用于将 R*C 的矩阵进行转置。

又如, 在一些实施例中, 可以使用向量转置指令 vstranspose v1, index1, vm1, v3, vm3, 其用于将方阵进行转置。在该示例中, vm1 的值为读通道使能; vm3 的值为写通道使能。vm1 中连续为 1 的位的数目 R 用于表示方阵的行数(或列数)。该向量转置指令 vstranspose 20 可以用于将 R*R 的方阵进行转置。

向量转置指令并不是标准 RISC 类型的指令。在常规的标准 RISC 类型的指令集中, 必须通过连续多条转置指令才能完成向量的转置功能。本方案通过使用向量转置指令, 能够提高部分网络的计算能力。25 例如, 对于神经网络训练过程, 通常会涉及大量的矩阵或方阵的转置操作。利用本方案的向量转置指令, 能够提高神经网络训练过程的计算效率。

在一些实施例中, 目标指令包括指数指令, 例如 vexp。在该示例中, 源操作数 220 指定源存储位置, 即 A_vaddr 为 &v1。算数逻辑单

值（例如，自然底数 e ）为底数的指数值。例如，`vexp v1, v3, vm3`，具有编码表示为“0b00_01000_01_0”的操作码，该指令表示将 `v3[i]` 赋值为 $\exp(v1[i])$ 。如果 `vm3[i]` 等于 1，则 `v3[i]` 是写使能的。

- 5 上述指数指令适用于 sigmoid 算子以及双曲函数 $\sinh/\cosh/\tanh$ 等算子。例如，sigmoid 算子、 $\sinh$ 算子、 $\cosh$ 算子和 $\tanh$ 算子可以由以下式（1）-式（4）来表示。

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}. \quad (1)$$

$$\sinh x = \frac{e^x - e^{-x}}{2} \quad (2)$$

$$\cosh x = \frac{e^x + e^{-x}}{2} \quad (3)$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

在式（1）-（4）中， x 表示待处理数据。

例如，在神经网络激活函数中，sigmoid 与双曲函数较为常见。通过使用本方案的指数指令，能够提高这类计算的效率。

- 15 在一些实施例 140 中，指令 140 包括 VM 寄存器指令，例如 `vm2index` 指令。源操作数 220 指示存储器中的源 VM 寄存器，即 `vm1`。算数逻辑单元被配置为：将源 VM 寄存器中被启用的位置处的索引存储到目标存储位置处。例如，指令 `vm2index vm1, v3, vm3`，具有编码表示为“0b00_01100_01_0”的操作码，其表示 $v3[i] = vm1[i] ? i : -1$ 。即，
20 如果 `vm1[i]` 的值为 1，则将 `v3[i]` 赋值为 `i`、反之，如果 `vm1[i]` 的值为 0，则将 `v3[i]` 赋值为 -1（例如，二进制补码表示的“-1”）。如果 `vm3[i]` 等于 1，则 `v3[i]` 是写使能的。

在一些实施例 140 中，指令 140 包括独热（onehot）码转换指令，例如 `vindex2vm`。这是一种 VM 寄存器操作指令。在这一示例中，源操

即 B_vaddr 为 &v2, B_index 为 index2。该指令涉及的读取向量掩码寄存器的行为, 不是作为存储器写入的写使能 (其他写入存储器的指令都需要读取向量掩码寄存器作为写使能)。目标操作数 230 指定目标 VM 寄存器, 即 vm3。算数逻辑单元被配置为: 将待处理数据在给定索引值处的值转换为独热码, 并且将该独热码存储到目标 VM 寄存器中。例如, 指令 vindex2vm v2, index2, vm3, 具有编码表示为 “0b00_10000_01_0” 的操作码。该指令表示将 vm3 赋值为 onehot(v2[index2]), 其中 onehot() 表示独热码转换函数。

以上所描述的独热码转换指令适用于索引类指令, 并且可支持独热码算子。将数字转换为独热编码形式。例如, 在实际使用时可以用以下两条指令来实现: vindex2vm v1, 0, vm1 以及 vmload vm1, v2, vm2, 其中第一条指令用于将 v1[0] 的数值转换为独热编码形式, 并且写入 vm1, 第二条指令 (vmload 将在下文的表 7 和表 8 中进行描述) 用于将 vm1 中的数值存储至 v2, 其中 vm2 的存储值全部位为 1。

以上结合表 4 描述了本公开的处理器 110 所支持的各类指令 140 的示例。应理解, 本公开的处理器 110 还可以支持更多的指令。以下的表 5 示出了处理器 110 所支持的更多的常规的指令 140 的示例。表 5 的指令可以参考表 1 或表 2 的指令定义来描述, 并且使用表 3 的示例编码方式对操作码进行编码。

表 5 示例常规指令

[95:8 6]	[85:7 0]	[69:6 0]	[59:5 4]	[53:3 8]	[37:2 8]	[27:2 2]	[21:6]	[5:0]
操作 码	A_va ddr	A_ind ex	A_vm	B_va ddr	B_ind ex	B_vm	C_va ddr	C_vm
vadd	&v1	预留	预留	&v2	预留	预留	&v3	vm3

vsub	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vsubi	&v1	预留	预留	imm2			&v3	vm3
vmax	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vmaxi	&v1	预留	预留	imm2			&v3	vm3
vmin	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vmini	&v1	预留	预留	imm2			&v3	vm3
vmul	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vmuli	&v1	预留	预留	imm2			&v3	vm3
vdiv	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vdivi	&v1	预留	预留	imm2			&v3	vm3
vrecip i	&v1	预留	预留	imm2			&v3	vm3
vsignj	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vsignj i	&v1	预留	预留	imm2			&v3	vm3
vsignj n	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vsignj ni	&v1	预留	预留	imm2			&v3	vm3

x	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vsignj xi	&v1	预留	预留	imm2			&v3	vm3
vand	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vandi	&v1	预留	预留	imm2			&v3	vm3
vor	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vori	&v1	预留	预留	imm2			&v3	vm3
vxor	&v1	预留	预留	&v2	预留	预留	&v3	vm3
vxori	&v1	预留	预留	imm2			&v3	vm3
vshift l	&v1	预留	预留	&v2	index 2	预留	&v3	vm3
vshift li	&v1	预留	预留	imm2			&v3	vm3
vshift r	&v1	预留	预留	&v2	index 2	预留	&v3	vm3
vshift ri	&v1	预留	预留	imm2			&v3	vm3
vcopy	&v1	预留	预留	预留			预留	预留
si2v	预留	预留	预留	imm2			&v3	vm3
vceil	&v1	预留	预留	预留			预留	预留

vfclas s	&v1	预留	预留	预留	预留	预留
vsum	&v1	预留	vm1	预留	预留	预留
v2sm ax	&v1	预留	vm1	预留	预留	预留
v2smi n	&v1	预留	vm1	预留	预留	预留
vclear	&v1	预留	vm1	预留	预留	预留

表 5 中的各个指令的功能和定义将由表 6 示出。这些指令的功能包括了各种加法、减法、乘法、除法、求最大值、求最小值、求倒数（仅支持立即数）、根据给定值对操作数移位，等等。这些指令提供了处理器 110 的基础计算，在此不进行详细描述。

表 6 示例常规指令的功能

指令	操作码的 编码表示	功能
vadd v1, v2, v3, vm3	0b00_000 01_00_0	$v3[i] = v1[i] + v2[i]$ #如果 vm3[i] 等于 1， 则 v3[i] 是写使能的。
vaddi v1, imm2, v3, vm3	0b00_000 01_00_1	$v3[i] = v1[i] + imm2$ #如果 vm3[i] 等于 1， 则 v3[i] 是写使能的。
vsub v1, v2, v3, vm3	0b00_000 01_01_0	$v3[i] = v1[i] - v2[i]$ # 如果 vm3[i] 等于 1， 则 v3[i] 是写使能的。

imm2, v3, vm3	0000_000 01_01_1	$v3[i] = v1[i] - imm2$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vmax v1, v2, v3, vm3	0b00_000 01_10_0	$v3[i] = \text{MAX}(v1[i], v2[i])$ #如果 $vm3[i]$ 等 于 1, 则 $v3[i]$ 是写使能的。
vmaxi v1, imm2, v3, vm3	0b00_000 01_10_1	$v3[i] = \text{MAX}(v1[i], imm2)$ #如果 $vm3[i]$ 等 于 1, 则 $v3[i]$ 是写使能的。
vmin v1, v2, v3, vm3	0b00_000 01_11_0	$v3[i] = \text{MIN}(v1[i], v2[i])$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vmini v1, imm2, v3, vm3	0b00_000 01_11_1	$v3[i] = \text{MIN}(v1[i], imm2)$ #如果 $vm3[i]$ 等 于 1, 则 $v3[i]$ 是写使能的。
vmul v1, v2, v3, vm3	0b00_000 10_00_0	$v3[i] = v1[i] * v2[i]$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vmuli v1, imm2, v3, vm3	0b00_000 10_00_1	$v3[i] = v1[i] * imm2$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vdiv v1, v2, v3, vm3	0b00_000 10_10_0	$v3[i] = v1[i] / v2[i]$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vdivi v1, imm2, v3, vm3	0b00_000 10_10_1	$v3[i] = v1[i] / imm2$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。
vrecipi v1, imm2, v3, vm3	0b00_000 10_11_1	$v3[i] = imm2 / v1[i]$ #如果 $vm3[i]$ 等于 1, 则 $v3[i]$ 是写使能的。

v2, v3, vm3	00_00_0	果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vsignji v1, imm2, v3, vm3	0b00_001 00_00_1	$v3[i] = \{imm2[DW-1], v1[i][DW-2:0]\}$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vsignjn v1, v2, v3, vm3	0b00_001 00_01_0	$v3[i] = \{\sim v2[i][DW-1], v1[i][DW-2:0]\}$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vsignjni v1, imm2, v3, vm3	0b00_001 00_01_1	$v3[i] = \{\sim imm2[DW-1], v1[i][DW-2:0]\}$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vsignjx v1, v2, v3, vm3	0b00_001 00_10_0	$v3[i] = \{v1[i][DW-1] \wedge v2[i][DW-1], v1[i][DW-2:0]\}$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vsignjxi v1, imm2, v3, vm3	0b00_001 00_10_1	$v3[i] = \{v1[i][DW-1] \wedge imm2[DW-1], v1[i][DW-2:0]\}$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vand v1, v2, v3, vm3	0b00_001 01_00_0	$v3[i] = v1[i] \& v2[i]$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vandi v1, imm2, v3, vm3	0b00_001 01_00_1	$v3[i] = v1[i] \& imm2$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vor v1, v2, v3, vm3	0b00_001 01_01_0	$v3[i] = v1[i] \mid v2[i]$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
vori v1, imm2, v3, vm3	0b00_001 01_01_1	$v3[i] = v1[i] \mid imm2$ #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。

v3, vm3	01_10_0	则 v3[i]是写使能的。
vxori v1, imm2, v3, vm3	0b00_001 01_10_1	$v3[i] = v1[i] \wedge imm2$ #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vshiftr v1, v2, index2, v3, vm3	0b00_001 11_00_0	$v3[i] = v1[\text{mod}(i - v2[\text{index2}],$ LANE_NUM)], 元素循环左移。 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vshiftr v1, imm2, v3, vm3	0b00_001 11_00_1	$v3[i] = v1[\text{mod}(i - imm2, \text{LANE_NUM})]$, 元 素循环左移。 #如果 vm3[i]等于 1, 则 v3[i] 是写使能的。
vshiftr v1, v2, index2, v3, vm3	0b00_001 11_01_0	$v3[i] = v1[\text{mod}(i + v2[\text{index2}],$ LANE_NUM)], 元素循环右移。 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vshiftr v1, imm2, v3, vm3	0b00_001 11_01_1	$v3[i] = v1[\text{mod}(i + imm2, \text{LANE_NUM})]$, 元素循环右移。 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vcopy v1, v3, vm3	0b00_010 00_00_0	$v3[i] = v1[i]$ #如果 vm3[i]等于 1, 则 v3[i] 是写使能的。
si2v imm2, v3, vm3	0b00_010 00_00_1	$v3[i] = imm2$ #如果 vm3[i]等于 1, 则 v3[i] 是写使能的。
vceil v1, v3, vm3	0b00_010 00_10_0	$v3[i] = \text{ceil}(v1[i])$ #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vfloor v1, v3, vm3	0b00_010 00_11_0	$v3[i] = \text{floor}(v1[i])$ #如果 vm3[i]等于 1, 则 v3[i]是写使能的。

v3, vm3	01_00_0	v3[i]是写使能的。
vsum v1, vm1, v3, vm3	0b00_010 10_00_0	$v3[i] = \text{SUM}(vm1[j] ? v1[j] : 0)$ for j from 0 to LANE_NUM - 1 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
v2smax v1, vm1, v3, vm3	0b00_010 10_10_0	$v3[i] = \text{MAX}(vm1[j] ? v1[j] : -\text{INF})$ for j from 0 to LANE_NUM - 1 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
v2smin v1, vm1, v3, vm3	0b00_010 10_11_0	$v3[i] = \text{MIN}(vm1[j] ? v1[j] : +\text{INF})$ for j from 0 to LANE_NUM - 1 #如果 vm3[i]等于 1, 则 v3[i]是写使能的。
vclear v1, vm1, v3, vm3	0b00_011 00_00_0	$v3[i] = vm1[i] ? 0 : vm1[i]$ #如果 vm3[i]等于 1, 则 v3[i]是写使能的。

表 6 中的 MAX()和 MIN()函数分别表示求最大值和求最小值的函数。DW 表示向量单词的宽度，LANE_NUM 表示一个向量单词中元素的数目，mod()函数表示求余函数，ceil()和 floor()函数分别表示向上取整和向下取整，SUM()函数表示求和函数。

- 5 在一些实施例中，处理器 110 所支持的指令还包括各种向量掩码寄存器访问和操作指令。表 7 中示出了向量掩码寄存器访问和操作指令的若干示例。表 7 中的各个指令的功能和定义将由表 8 示出。这些指令的功能包括了向量掩码寄存器的读取、写入和操作。这些指令涉及的读取向量掩码寄存器的行为，不是作为存储器写入的写使能（其他写入存储器的指令都需要读取向量掩码寄存器作为写使能）。这些指令在此不进行详细描述。
- 10

表 7 示例向量掩码寄存器指令编码

6]	0]	0]	4]	8]	8]	2]	[41:0]	[5:0]
操作 码	A_va ddr	A_ind ex	A_vm	B_va ddr	B_ind ex	B_vm	C_va ddr	C_vm
vmset	&v1	index 1	预留	预留	预留	预留	预留	vm3
vmset i	imm1			imm2			预留	vm3
vman d	预留	预留	vm1	预留	预留	vm2	预留	vm3
vmor	预留	预留	vm1	预留	预留	vm2	预留	vm3
vmxo r	预留	预留	vm1	预留	预留	vm2	预留	vm3
vcmp eq	&v1	预留	预留	&v2	预留	预留	预留	vm3
vcmp eqi	&v1	预留	预留	imm2			预留	vm3
vcmp ne	&v1	预留	预留	&v2	预留	预留	预留	vm3
vcmp nei	&v1	预留	预留	imm2			预留	vm3
vcmpl t	&v1	预留	预留	&v2	预留	预留	预留	vm3
vcmpl	&v1	预留	预留	imm2			预留	vm3

vcmpge	&v1	预留	预留	&v2	预留	预留	预留	vm3
vcmpgei	&v1	预留	预留	imm2			预留	vm3
vmload	预留	预留	vm1	预留	预留	预留	v3	vm3

表 8 示例向量掩码寄存器指令的功能

指令	操作码的 编码表示	功能
vmset v1, index1, vm3	0b00_100 00_00_0	vm3 = {v1[index1], v1[index - 1]}
vmseti imm1, imm2, vm3	0b00_100 00_00_1	vm3 = {imm1, imm2}
vmmand vm1, vm2, vm3	0b00_100 01_00_0	vm3 = vm1 & vm2
vmor vm1, vm2, vm3	0b00_100 01_01_0	vm3 = vm1 vm2
vmxor vm1, vm2, vm3	0b00_100 01_10_0	vm3 = vm1 ^ vm2
vcmpeq v1, v2, vm3	0b00_100 10_00_0	vm3[i] = (v1[i] == v2[i]) ? 1 : 0

imm2, vm3	10_00_1	$vm3[i] = (v1[i] == imm2) ? 1 : 0$
vcmpne v1, v2, vm3	0b00_100 10_01_0	$vm3[i] = (v1[i] != v2[i]) ? 1 : 0$
vcmpnei v1, imm2, vm3	0b00_100 10_01_1	$vm3[i] = (v1[i] != imm2) ? 1 : 0$
vcmplt v1, v2, vm3	0b00_100 10_10_0	$vm3[i] = (v1[i] < v2[i]) ? 1 : 0$
vcmplti v1, imm2, vm3	0b00_100 10_10_1	$vm3[i] = (v1[i] < imm2) ? 1 : 0$
vcmpge v1, v2, vm3	0b00_100 10_11_0	$vm3[i] = (v1[i] >= v2[i]) ? 1 : 0$
vcmpgei v1, imm2, vm3	0b00_100 10_11_1	$vm3[i] = (v1[i] >= imm2) ? 1 : 0$
vmload vm1, v3, vm3	0b00_100 11_00_0	$\{v3[2j+1], v3[2j]\} = vm1$, for j from 0 to LANE_NUM / 2 - 1. # v3[i] is write-enabled where vm3[i] = 1

在一些实施例中，处理器 110 所支持的指令 140 还包括内部寄存器访问和操作指令。这类指令用于处理内部寄存器的访问以及一些特殊操作。例如，写入内部控制和状态（CSR）寄存器，写入固定值、
5 或数据存储器 VCCM 中的某个 SIMD 长度数据等。又如，读出内部 CSR 寄存器或者读出到数据存储器 VCCM；以及空指令（即，不进行任何操作，等待 1 个周期）等。

以下的表 9 示出了内部寄存器访问和操作指令的若干示例。表 10 示出了表 9 中的各个指令的功能。这些指令在本文中不进行详细描述。

10 注意，对于表 9 中的 vwcsr 指令，源操作数在 A 通道；而对于 vwcsri

表 9 示例内部寄存器指令

[95:86]	[85:70]	[69:60]	[59:54]	[53:38]	[37:28]	[27:22]	[21:6]	[5:0]
目标 操作 码	A_va ddr	A_ind ex	A_vm	B_va ddr	B_ind ex	B_vm	C_va ddr	C_vm
vnop	预留	预留	预留	预留	预留	预留	预留	预留
vwcsr	&v1	预留	预留	预留	预留	预留	预留	预留
vwcsr i	预留			imm2			预留	预留
vrcsr	预留	预留	预留	预留	预留	预留	&v3	vm3

5

表 10 示例内部寄存器指令的功能

指令	操作码的 编码表示	功能
vnop	0b00_110 00_00_0	无操作
vwcsr v1, CSRs	0b00_110 01_00_0	CSR[i] = v1[i], 对于每个 i, vm3 是写使能的。
vwcsri imm2, CSRs	0b00_110 01_00_1	CSR[i] = imm2, 对于每个 i, vm3 是写使能

vrcsr CSRs, v3, vm3	0b00_110 01_01_0	v3[i] = CSR[i] #如果 vm3[i] 等于 1, 则 v3[i] 是写使能的。
------------------------	---------------------	---

以上结合表 4-表 10 描述了处理器 110 所支持的各个指令。这些指令可以由处理器 110 的指令解码器 120 进行解码, 并且由算术执行单元 130 来执行。这些指令可以构成由处理器 110 所支持的指令集。

5 应当理解, 在一些实施例, 可以仅由上述各个指令中的部分或全部的指令来构建指令集。备选地或附加地, 还可以采用以上没有描述的其他适当的指令来构建由处理器 110 支持的指令集。

此外, 应当理解, 虽然以上参考表 1-表 3 规定的示例指令定义和示例操作码编码表示列举了表 4-表 10 中的各个指令, 但这仅仅是示例性的, 而不是限制性的。本公开的处理器所支持的指令集可以采用任何适当的方式进行定义和编码。例如, 各个指令的各个位可以具有不同于表 1 或表 2 中各个位所表示的含义。又如, 各个指令的操作码的编码表示可以具有与表 3 中不同的位数, 各个位数也可以与表 3 中的各个位具有不同的含义。上述表 4-表 10 中的各个指令的操作码的
10 编码表示可以改变或者互换。各个指令也可以使用其他的名称来表示。本公开的范围在此方面不受限制。

以上所描述的指令中不包括分支类型的指令, 也不包括加载/存储 (load/store) 类型的指令。与传统的诸如向量寄存器文件的 SIMD 处理器不同, 本公开所采用的寄存器是存储器到存储器的 SIMD 处理器
20 架构。上述指令集定义了多个 (例如, 64 个或更多或更少的) 向量掩码寄存器, 用于表示每个 SIMD 指令需要处理的具体的向量。

本方案通过采用适用于存储器到存储器架构的 SIMD 处理器, 来简化处理器 110 的操作。以此方式, 处理器 110 能够使用简单的指令集来完成大量的向量计算。例如, 处理器 110 能够使用简单的指令集
25 来进行神经网络算子的向量计算等任务。通过这样, 本方案能够采用

训练和/或推理等计算,本公开的方案能够极大地提高计算效率。例如,根据本公开的实施例的处理器可以支持各种索引确定指令,从而提高各种诸如求取坐标等向量计算的效率。又如,本公开的处理器可以处理诸如向量转置指令,从而提高神经网络训练过程中的相应计算的计
5 算效率。再如,本公开的处理器能够支持指数指令,从而使得诸如 sigmoid 算子和双曲函数算子等的计算效率得以提高和优化。

图 4 示出了根据本公开的一些实施例的用于数据处理的过程 400 的流程图。过程 400 可以在处理器 110 处实现。为便于讨论,将参考
10 图 1 的环境 100 来描述过程 400。

在框 410,由处理器 110 解码用于向量操作的目标指令,例如指令 140。例如,可以由处理器 110 的指令解码器 120 解码指令 140。指令 140 涉及目标操作码 210、源操作数 220 和目标操作数 230。目标操作码 210 指示指令 140 所指定的向量操作。源操作数 220 至少指
15 定存储器中的用于读取待处理数据的源存储位置。目标操作数 230 至少指定存储器中的用于写入处理结果的目标存储位置。

在框 420,由处理器 110 从存储器的源存储位置读取待处理数据。例如,可以由处理器 110 的算数逻辑单元 130 从存储器的源存储位置读取待处理数据。在框 430,由处理器 110 对待处理数据执行与目标
20 指令所指定的向量操作相关联的算数逻辑运算。例如,可以由处理器 110 的算数逻辑单元 130 执行上述算数逻辑运算。在框 440,由处理器 110 将待处理数据的处理结果写入存储器的目标存储位置。例如,可以由处理器 110 的算数逻辑单元 130 将处理结果写入目标存储位置。

25 在一些实施例中,指令 140 包括索引确定指令。索引确定指令可以是第一索引确定指令(v2indexl 或 v2indexr)或者第二索引确定指令(v2indexli 或 v2indexri)。源操作数 220 指定存储器的第一存储空间的位置,源操作数 220 还指定存储器的第二存储空间内的待处理数据的给定索引值或者第一立即数。在框 430 处,处理器 110 执行的算

处理数据中的由给定索引值所指示的位置处的值在第一存储空间中的存储位置。第二索引指示第一立即数在第一存储空间中的存储位置。

- 5 在一些实施例中，指令 140 包括第一数值确定指令（例如，指令 `sindex2v`），源操作数 220 指定存储器的第一存储空间的位置，源操作数 220 还指定存储器的第二存储空间内的待处理数据的给定索引值。在框 430 处，处理器 110 执行的算数逻辑运算包括：确定待处理数据在由给定索引值所指示的位置处的给定值；以及确定第一存储空间中的在以给定值为索引的位置处的第一值。

 在一些实施例中，指令 140 包括第二数值确定指令，例如，指令 `s2v`。源操作数 220 指定存储器的第二存储空间内的待处理数据的给定索引值。在框 430 处，处理器 110 执行的算数逻辑运算包括：确定待处理数据中的由给定索引值所指示的位置处的第二值。

- 15 在一些实施例中，指令 140 包括向量转置指令，例如指令 `vtranspose` 或 `vstranspose`。源操作数 220 指定存储器中的第一存储空间中的第一位置。在框 430 处，处理器 110 执行的算数逻辑运算包括：将第一存储空间中的第一位置处的待处理数据进行向量转置，以获得转置后的待处理数据。

- 20 在一些实施例中，指令 140 包括指数指令，例如指令 `vexp`。源操作数 220 指定源存储位置。在框 430 处，处理器 110 执行的算数逻辑运算包括：以源存储位置处的待处理数据为幂来确定以预定数值为底数的指数值。

- 在一些实施例中，指令 140 包括 VM 寄存器指令，例如 `vm2index`。
- 25 指令 140 的源操作数 220 指示存储器中的源 VM 寄存器。在框 430 处，处理器 110 执行的算数逻辑运算包括：将源 VM 寄存器中被启用的位置处的索引存储到目标存储位置处。

 在一些实施例中，以上所描述的各个指令 140 的目标存储位置处包括处理结果向量。目标操作数 230 还指示目标 VM 寄存器。目标 VM 寄

应的处理结果。例如，如果目标寄存器 vm3[i] 为 1，则表示处理结果向量单词的第 i 个元素是写使能的，可以被写入相应的处理结果。反之，如果目标寄存器 vm3[i] 为 0，则处理结果向量单词的第 i 个元素不能被写入相应的处理结果。

在一些实施例中，指令 140 包括独热码转换指令，例如指令 vindex2vm。源操作数 220 指定存储器的第二存储空间内的待处理数据的给定索引值。目标操作数 230 指定目标 VM 寄存器。在框 430 处，处理器 110 将待处理数据在给定索引值处的值转换为独热码。处理器 110 还被配置为将独热码存储到目标 VM 寄存器中。

图 5 示出了其中可以包括根据本公开的一个或多个实施例的处理器 110 的电子设备 500 的框图。应当理解，图 5 所示出的电子设备 500 仅仅是示例性的，而不应当构成对本文所描述的实施例的功能和范围的任何限制。

如图 5 所示，电子设备 500 是通用电子设备或计算设备的形式。电子设备 500 的组件可以包括但不限于一个或多个处理器 110、存储器 520、存储设备 530、一个或多个通信单元 540、一个或多个输入设备 550 以及一个或多个输出设备 560。在一些实施例中，处理器 110 可以根据存储器 520 中存储的程序来执行各种处理。处理器 110 可以是多核处理器，该多核处理器可以并行执行计算机可执行指令，以提高电子设备 500 的并行处理能力。

电子设备 500 通常包括多个计算机存储介质。这样的介质可以是电子设备 500 可访问的任何可以获得的介质，包括但不限于易失性和非易失性介质、可拆卸和不可拆卸介质。存储器 520 可以是易失性存储器（例如寄存器、高速缓存、随机访问存储器（RAM））、非易失性存储器（例如，只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、闪存）或它们的某种组合。存储设备 530 可以是可拆卸或不可拆卸的介质，并且可以包括机器可读介质，诸如闪存驱动、磁盘或者任何其他介质，其可以能够用于存储信息和/或数据（例如用

电子设备 500 可以进一步包括另外的可拆卸/不可拆卸、易失性/非易失性存储介质。尽管未在图 5 中示出，可以提供用于从可拆卸、非易失性磁盘（例如“软盘”）进行读取或写入的磁盘驱动和用于从可拆卸、非易失性光盘进行读取或写入的光盘驱动。在这些情况中，每个驱动可以由一个或多个数据介质接口被连接至总线（未示出）。存储器 520 可以包括计算机程序产品 525，其具有一个或多个程序模块，这些程序模块被配置为执行本公开的各种实施例的各种方法或动作。例如，这些程序模块可以被配置为实现处理器 110 的各种功能或动作，诸如实现指令解码器 120 和算术逻辑单元 130 的功能。

通信单元 540 实现通过通信介质与其他电子设备或计算设备进行通信。附加地，电子设备 500 的组件的功能可以以单个计算集群或多个计算机器来实现，这些计算机器能够通过通信连接进行通信。因此，电子设备 500 可以使用与一个或多个其他服务器、网络个人计算机（PC）或者另一个网络节点的逻辑连接来在联网环境中进行操作。

输入设备 550 可以是一个或多个输入设备，例如鼠标、键盘、追踪球等。输出设备 560 可以是一个或多个输出设备，例如显示器、扬声器、打印机等。电子设备 500 还可以根据需要通过通信单元 540 与一个或多个外部设备（未示出）进行通信，外部设备诸如存储设备、显示设备等，与一个或多个使得用户与电子设备 500 交互的设备进行通信，或者与使得电子设备 500 与一个或多个其他电子设备或计算设备通信的任何设备（例如，网卡、调制解调器等）进行通信。这样的通信可以经由输入/输出（I/O）接口（未示出）来执行。

根据本公开的示例性实现方式，提供了一种计算机可读存储介质，其上存储有计算机可执行指令，其中计算机可执行指令被处理器执行以实现上文描述的方法。根据本公开的示例性实现方式，还提供了一种计算机程序产品，计算机程序产品被有形地存储在非瞬态计算机可读介质上并且包括计算机可执行指令，而计算机可执行指令被处理器执行以实现上文描述的方法。

图和/或框图描述了本公开的各个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。

- 5 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理单元，从而生产出一种机器，使得这些指令在通过计算机或其他可编程数据处理装置的处理单元执行时，产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介
- 10 质中，这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作，从而，存储有指令的计算机可读介质则包括一个制造品，其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

- 可以把计算机可读程序指令加载到计算机、其他可编程数据处理
- 15 装置、或其他设备上，使得在计算机、其他可编程数据处理装置或其他设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机、其他可编程数据处理装置、或其他设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

- 根据本公开的一个或多个实施例，示例 1 描述了一种处理器，该
- 20 处理器包括指令解码器，该指令解码器被配置为解码用于向量操作的目标指令。目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作。源操作数至少指定存储器中的用于读取待处理数据的源存储位置。目标操作数至少指定存储器中的用于写入处理结果的目标存储位置。该处理器还包括算数逻辑单
- 25 元，该算数逻辑单元被耦合至指令解码器和存储器。该算数逻辑单元被配置为：从存储器的源存储位置读取待处理数据；对待处理数据执行与目标指令所指定的向量操作相关联的算数逻辑运算；以及将待处理数据的处理结果写入存储器的目标存储位置。

 根据本公开的一个或多个实施例，示例 2 包括根据示例 1 所描述

器的第一存储空间的位置，源操作数还指定存储器的第二存储空间内的待处理数据的给定索引值。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：确定第一索引，
5 该第一索引指示待处理数据中的由给定索引值所指示的位置处的值在第一存储空间中的存储位置。

根据本公开的一个或多个实施例，示例 3 包括根据示例 1 所描述的处理器，其中目标指令包括第二索引确定指令，源操作数指定存储器的第一存储空间的位置，源操作数还指定第一立即数。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：确定第二索引，该第二索引指示第一立即数在第一存储空间中的存储位置。

根据本公开的一个或多个实施例，示例 4 包括根据示例 1 所描述的处理器，其中目标指令包括第一数值确定指令，源操作数指定存储器的第一存储空间的位置，源操作数还指定存储器的第二存储空间内的待处理数据的给定索引值。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：确定待处理数据在由给定索引值所指示的位置处的给定值；以及确定第一存储空间中的在以给定值为索引的位置处的第一值。

根据本公开的一个或多个实施例，示例 5 包括根据示例 1 所描述的处理器，其中目标指令包括第二数值确定指令，源操作数指定存储器的第二存储空间内的待处理数据的给定索引值。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：确定待处理数据中的由给定索引值所指示的位置处的第二值。

根据本公开的一个或多个实施例，示例 6 包括根据示例 1 所描述的处理器，其中目标指令包括向量转置指令，源操作数至少指定存储器中的第一存储空间中的第一位置。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：将第一存储空间中的第一位置处的待处理数据进行向量转置，以获得转置后的

根据本公开的一个或多个实施例，示例 7 包括根据示例 1 所描述的处理器，其中目标指令包括指数指令，源操作数指定源存储位置。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：以源存储位置处的待处理数据为幂来确定以预定数值为底数的指数值。

根据本公开的一个或多个实施例，示例 8 包括根据示例 1 所描述的处理器，其中目标指令包括向量掩码 VM 寄存器指令，源操作数指示存储器中的源 VM 寄存器。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：将源 VM 寄存器中被启用的位置处的索引存储到目标存储位置处。

根据本公开的一个或多个实施例，示例 9 包括根据示例 2 至示例 8 任一项所描述的处理器，其中目标存储位置处包括处理结果向量，目标操作数还指示目标向量掩码 VM 寄存器，目标 VM 寄存器的各个位置处的值指示处理结果向量的相应位置处是否要被写入相应的处理结果。

根据本公开的一个或多个实施例，示例 10 包括根据示例 1 所描述的处理器，其中目标指令包括独热码转换指令，源操作数指定存储器的第二存储空间内的待处理数据的给定索引值，目标操作数指定目标向量掩码 VM 寄存器。算数逻辑单元被配置为如下，以执行与目标指令所指定的向量操作相关联的算数逻辑运算：将待处理数据在给定索引值处的值转换为独热码；以及将独热码存储到目标 VM 寄存器中。

根据本公开的一个或多个实施例，示例 11 描述了一种数据处理的方法。该方法包括：解码用于向量操作的目标指令，目标指令涉及目标操作码、源操作数和目标操作数。目标操作码指示目标指令所指定的向量操作。源操作数至少指定存储器中的用于读取待处理数据的源存储位置。目标操作数至少指定存储器中的用于写入处理结果的目标存储位置。该方法还包括：从存储器的源存储位置读取待处理数据；

运算；以及将待处理数据的处理结果写入存储器的目标存储位置。

根据本公开的一个或多个实施例，示例 12 包括根据示例 11 所描述的方法，其中目标指令包括索引确定指令，源操作数指定存储器的第一存储空间的位置，源操作数还指定以下至少一项：存储器的第二存储空间内的待处理数据的给定索引值、第一立即数。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括以下至少一项：确定第一索引，该第一索引指示待处理数据中的由给定索引值所指示的位置处的值在第一存储空间中的存储位置；确定第二索引，第二索引指示第一立即数在第一存储空间中的存储位置。

根据本公开的一个或多个实施例，示例 13 包括根据示例 11 所描述的方法，其中目标指令包括第一数值确定指令，源操作数指定存储器的第一存储空间的位置，源操作数还指定存储器的第二存储空间内的待处理数据的给定索引值。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括：确定待处理数据在由给定索引值所指示的位置处的给定值；以及确定第一存储空间中的在以给定值为索引的位置处的第一值。

根据本公开的一个或多个实施例，示例 14 包括根据示例 11 所描述的方法，其中目标指令包括第二数值确定指令，源操作数指定存储器的第二存储空间内的待处理数据的给定索引值。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括：确定待处理数据中的由给定索引值所指示的位置处的第二值。

根据本公开的一个或多个实施例，示例 15 包括根据示例 11 所描述的方法，其中目标指令包括向量转置指令，其中源操作数至少指定存储器中的第一存储空间中的第一位置。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括：将第一存储空间中的第一位置处的待处理数据进行向量转置，以获得转置后的待处理数据。

根据本公开的一个或多个实施例，示例 16 包括根据示例 11 所描述的方法，其中目标指令包括指数指令，源操作数指定源存储位置。

存储位置处的待处理数据为幂来确定以预定数值为底数的指数值。

根据本公开的一个或多个实施例，示例 17 包括根据示例 11 所描述的方法，其中目标指令包括向量掩码 VM 寄存器指令，源操作数指示存储器中的源 VM 寄存器。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括：将源 VM 寄存器中被启用的位置处的索引存储到目标存储位置处。

根据本公开的一个或多个实施例，示例 18 包括根据示例 11 所描述的方法，其中目标指令包括独热码转换指令，源操作数指定存储器的第二存储空间内的待处理数据的给定索引值，目标操作数指定目标向量掩码 VM 寄存器。执行与目标指令所指定的向量操作相关联的算数逻辑运算包括：将待处理数据在给定索引值处的值转换为独热码；以及将独热码存储到目标 VM 寄存器中。

根据本公开的一个或多个实施例，示例 19 描述了一种电子设备，其至少包括根据示例 1 至示例 10 任一项所述的处理器。

根据本公开的一个或多个实施例，示例 20 描述了一种计算机可读存储介质，其上存储有计算机程序。该计算机程序被处理器执行以实现根据示例 11 至 18 任一项所述的方法。

附图中的流程图和框图显示了根据本公开的多个实现的系统、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、程序段或指令的一部分，模块、程序段或指令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。在有些作为替换的实现中，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，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。

5 尽性的，并且也不限于所公开的各实现。在不偏离所说明的各实现的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择，旨在最好地解释各实现的原理、实际应用或对市场中的技术的改进，或者使本技术领域的其他普通技术人员能理解本文公开的各个实现方式。

权利要求书

1. 一种处理器，包括：

指令解码器，被配置为解码用于向量操作的目标指令，所述目标指令涉及目标操作码、源操作数和目标操作数，所述目标操作码指示所述目标指令所指定的向量操作，所述源操作数至少指定存储器中的用于读取待处理数据的源存储位置，所述目标操作数至少指定所述存储器中的用于写入处理结果的目标存储位置；以及

算数逻辑单元，耦合至所述指令解码器和所述存储器，并且被配置为：

从所述存储器的所述源存储位置读取所述待处理数据；

对所述待处理数据执行与所述目标指令所指定的向量操作相关联的算数逻辑运算；以及

将所述待处理数据的处理结果写入所述存储器的所述目标存储位置。

2. 根据权利要求 1 所述的处理器，其中所述目标指令包括第一索引确定指令，所述源操作数指定所述存储器的第一存储空间的位置，所述源操作数还指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

确定第一索引，所述第一索引指示所述待处理数据中的由给定索引值所指示的位置处的值在所述第一存储空间中的存储位置。

3. 根据权利要求 1 所述的处理器，其中所述目标指令包括第二索引确定指令，所述源操作数指定所述存储器的第一存储空间的位置，所述源操作数还指定第一立即数，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的

确定第二索引，所述第二索引指示所述第一立即数在所述第一存储空间中的存储位置。

4. 根据权利要求1所述的处理器，其中所述目标指令包括第一数值
5 确定指令，所述源操作数指定所述存储器的第一存储空间的位置，所述源操作数还指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

- 10 确定所述待处理数据在由所述给定索引值所指示的位置处的给定值；以及

确定所述第一存储空间中的在以所述给定值为索引的位置处的第一值。

5. 根据权利要求1所述的处理器，其中所述目标指令包括第二数值
15 确定指令，所述源操作数指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

- 20 确定所述待处理数据中的由给定索引值所指示的位置处的第二值。

6. 根据权利要求1所述的处理器，其中所述目标指令包括向量转置指令，所述源操作数至少指定所述存储器中的第一存储空间中的第一位置，并且

- 25 所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

将所述第一存储空间中的第一位置处的所述待处理数据进行向量转置，以获得转置后的待处理数据。

所述源操作数指定所述源存储位置，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

5 以所述源存储位置处的所述待处理数据为幂来确定以预定数值为底数的指数值。

8. 根据权利要求 1 所述的处理器，其中所述目标指令包括向量掩码 VM 寄存器指令，所述源操作数指示所述存储器中的源 VM 寄存器，并且

10 所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

将所述源 VM 寄存器中被启用的位置处的索引存储到所述目标存储位置处。

9. 根据权利要求 2 至 8 中任一项所述的处理器，其中所述目标存储位置处包括处理结果向量，所述目标操作数还指示目标向量掩码 VM 寄存器，所述目标 VM 寄存器的各个位置处的值指示所述处理结果向量的相应位置处是否要被写入相应的处理结果。

10. 根据权利要求 1 所述的处理器，其中所述目标指令包括独热码转换指令，所述源操作数指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，所述目标操作数指定目标向量掩码 VM 寄存器，并且

所述算数逻辑单元被配置为如下，以执行与所述目标指令所指定的向量操作相关联的算数逻辑运算：

25 将所述待处理数据在给定索引值处的值转换为独热码；以及
将所述独热码存储到所述目标 VM 寄存器中。

11. 一种数据处理的方法，包括：

解码用于向量操作的目标指令，所述目标指令涉及目标操作码、源

操作，所述源操作数至少指定存储器中的用于读取待处理数据的源存储位置，所述目标操作数至少指定所述存储器中的用于写入处理结果的目标存储位置；

5 从所述存储器的所述源存储位置读取所述待处理数据；

对所述待处理数据执行与所述目标指令所指定的向量操作相关联的算数逻辑运算；以及

将所述待处理数据的处理结果写入所述存储器的所述目标存储位置。

10 12. 根据权利要求 11 所述的方法，其中所述目标指令包括索引确定指令，所述源操作数指定所述存储器的第一存储空间的位置，所述源操作数还指定以下至少一项：所述存储器的第二存储空间内的所述待处理数据的给定索引值、第一立即数，并且

15 其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括以下至少一项：

确定第一索引，所述第一索引指示所述待处理数据中的由给定索引值所指示的位置处的值在所述第一存储空间中的存储位置，

确定第二索引，所述第二索引指示所述第一立即数在所述第一存储空间中的存储位置。

20 13. 根据权利要求 11 所述的方法，其中所述目标指令包括第一数值确定指令，所述源操作数指定所述存储器的第一存储空间的位置，所述源操作数还指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，并且

25 其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括：

确定所述待处理数据在由所述给定索引值所指示的位置处的给定值；以及

第一值。

14. 根据权利要求 11 所述的方法，其中所述目标指令包括第二数值确定指令，所述源操作数指定所述存储器的第二存储空间内的所述待处理数据的给定索引值，并且

其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括：确定所述待处理数据中的由给定索引值所指示的位置处的第二值。

15. 根据权利要求 11 所述的方法，其中所述目标指令包括向量转置指令，所述源操作数至少指定所述存储器中的第一存储空间中的第一位置，并且

其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括：将所述第一存储空间中的第一位置处的所述待处理数据进行向量转置，以获得转置后的待处理数据。

16. 根据权利要求 11 所述的方法，其中所述目标指令包括指数指令，所述源操作数指定所述源存储位置，并且

其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括：以所述源存储位置处的所述待处理数据为幂来确定以预定数值为底数的指数值。

17. 根据权利要求 11 所述的方法，其中所述目标指令包括向量掩码 VM 寄存器指令，所述源操作数指示所述存储器中的源 VM 寄存器，并且

其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括：将所述源 VM 寄存器中被启用的位置处的索引存储到所述目标存储位置处。

18. 根据权利要求 11 所述的方法，其中所述目标指令包括独热码转换指令，所述源操作数指定所述存储器的第二存储空间内的所述待处理

且

其中执行与所述目标指令所指定的向量操作相关联的算数逻辑运算包括:

- 5 将所述待处理数据在给定索引值处的值转换为独热码; 以及
 将所述独热码存储到所述目标 VM 寄存器中。

19. 一种电子设备, 至少包括根据权利要求 1 至 10 任一项所述的处理器。

20. 一种计算机可读存储介质, 其上存储有计算机程序, 所述计
10 算机程序被处理器执行以实现根据权利要求 11 至 18 任一项所述的方法。

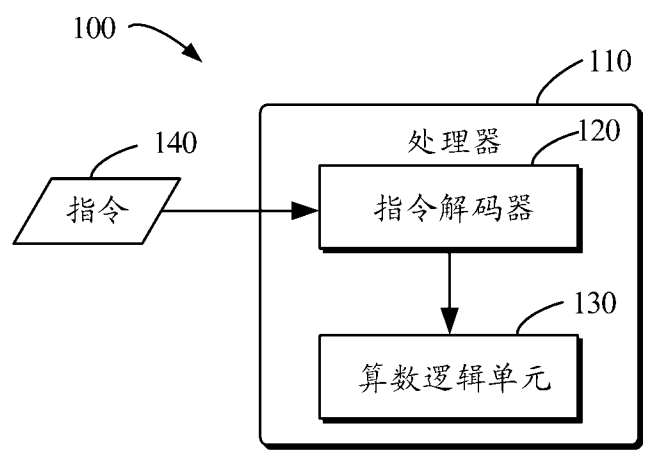


图 1

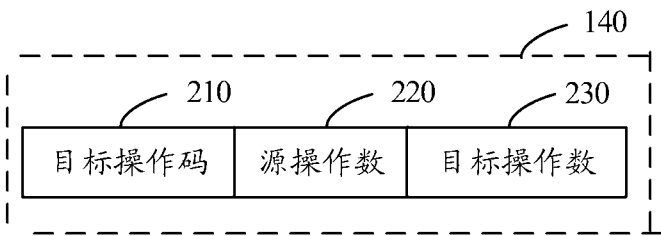


图 2

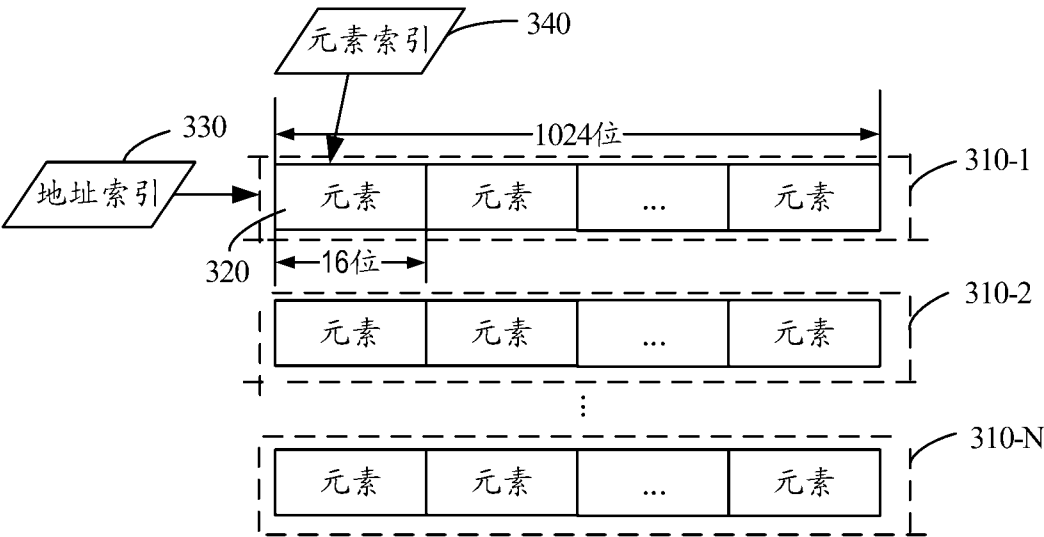


图 3

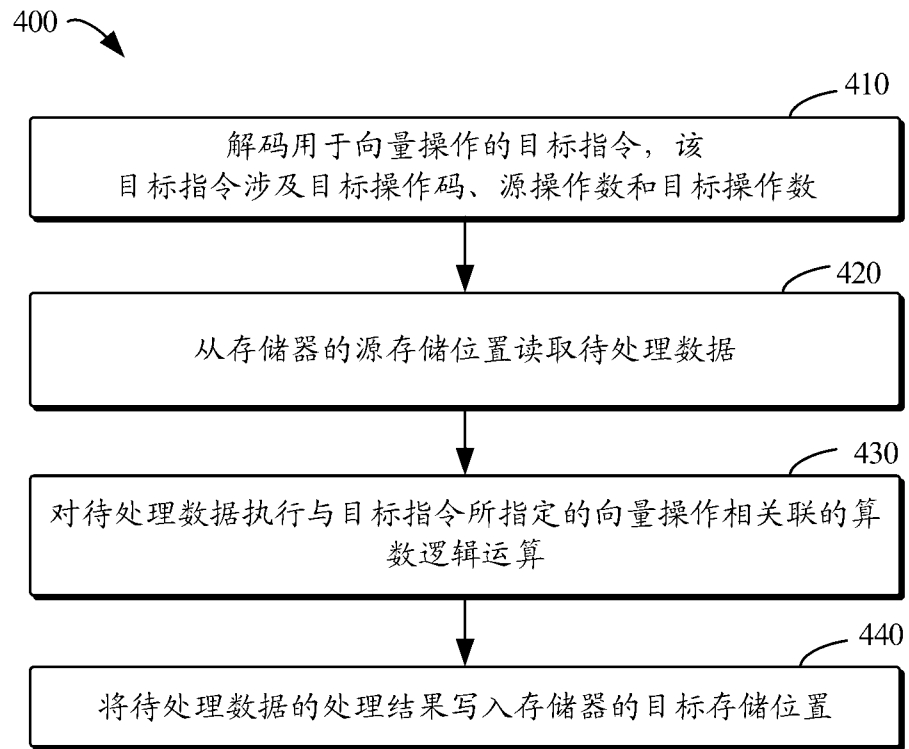


图 4

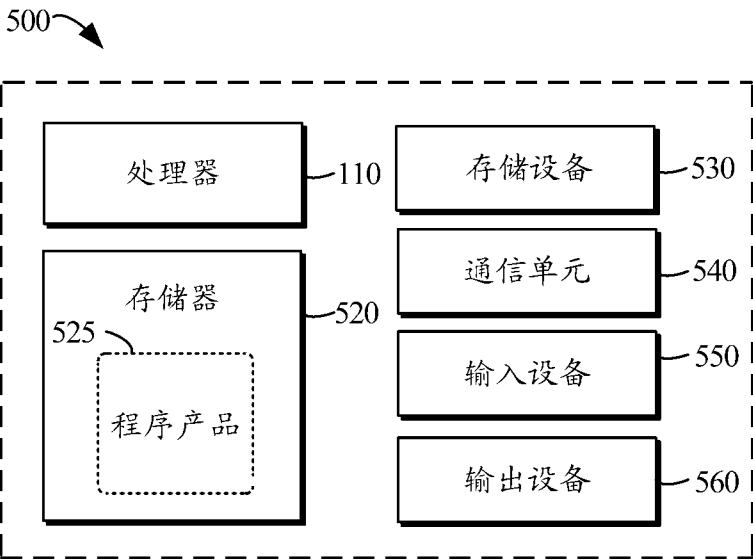


图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/098716

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/30(2018.01)i; G06F 9/305(2018.01)i; G06F 9/345(2018.01)i; G06F 9/355(2018.01)i; G06N 3/063(2023.01)i; G06N 3/08(2023.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; CNKI; VEN; USTXT; WOTXT; EPTXT: 处理器, 解码, 操作码, 源操作数, 目的地操作数, 向量, 存储器, 地址, 位置, CPU, decoding, opcode, source operand, destination operand, vector, storage, address, position

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 109791488 A (INTEL CORP.) 21 May 2019 (2019-05-21) description, paragraphs [0003]-[0223]	1-20
A	CN 113849769 A (INTEL CORP.) 28 December 2021 (2021-12-28) entire document	1-20
A	CN 109992240 A (INTEL CORP.) 09 July 2019 (2019-07-09) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

03 July 2023

Date of mailing of the international search report

14 July 2023

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2023/098716

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109791488	A	21 May 2019	TW	201820125	A	01 June 2018
				EP	3519943	A1	07 August 2019
				US	2018095758	A1	05 April 2018
				US	11023231	B2	01 June 2021
				WO	2018063513	A1	05 April 2018
				TW	202223631	A	16 June 2022
				TW	756251	B1	01 March 2022
CN	113849769	A	28 December 2021	US	2021405974	A1	30 December 2021
				EP	3929733	A1	29 December 2021
CN	109992240	A	09 July 2019	US	2019205137	A1	04 July 2019
				DE	102018129341	A1	04 July 2019

国际检索报告

国际申请号

PCT/CN2023/098716

A. 主题的分类

G06F 9/30(2018.01)i; G06F 9/305(2018.01)i; G06F 9/345(2018.01)i; G06F 9/355(2018.01)i; G06N 3/063(2023.01)i; G06N 3/08(2023.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC: G06F G06N

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNABS;CNTXT;CNKI;VEN;USTXT;WOTXT;EPTXT: 处理器, 解码, 操作码, 源操作数, 目的地操作数, 向量, 存储器, 地址, 位置, CPU, decoding, opcode, source operand, destination operand, vector, storage, address, position

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	CN 109791488 A (英特尔公司) 2019年5月21日 (2019 - 05 - 21) 说明书第[0003]-[0223]段	1-20
A	CN 113849769 A (英特尔公司) 2021年12月28日 (2021 - 12 - 28) 全文	1-20
A	CN 109992240 A (英特尔公司) 2019年7月9日 (2019 - 07 - 09) 全文	1-20

☐ 其余文件在C栏的续页中列出。

☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2023年7月3日

国际检索报告邮寄日期

2023年7月14日

ISA/CN的名称和邮寄地址

中国国家知识产权局

中国北京市海淀区蓟门桥西土城路6号 100088

授权官员

郭俊

电话号码 (+86) 0512-88995915

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2023/098716

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	109791488	A	2019年5月21日	TW	201820125	A	2018年6月1日
				EP	3519943	A1	2019年8月7日
				US	2018095758	A1	2018年4月5日
				US	11023231	B2	2021年6月1日
				WO	2018063513	A1	2018年4月5日
				TW	202223631	A	2022年6月16日
				TW	756251	B1	2022年3月1日
CN	113849769	A	2021年12月28日	US	2021405974	A1	2021年12月30日
				EP	3929733	A1	2021年12月29日
CN	109992240	A	2019年7月9日	US	2019205137	A1	2019年7月4日
				DE	102018129341	A1	2019年7月4日