

一种浮点数处理方法、装置、计算设备和存储介质

相关申请的交叉引用

本申请要求在2023年12月25日提交中华人民共和国国家知识产权局、申请号为202311811917.5、申请名称为“一种浮点数处理方法、装置、计算设备和存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机技术领域，尤其涉及一种浮点数处理方法、装置、计算设备和存储介质。

背景技术

在计算机系统中，浮点（floating point，FP）是一种对于实数的近似数值表示法，也称为浮点型数据表示法。示例性地，浮点型数据表示法可以包括FP8、FP16和FP32，FP8表示8位的浮点数，FP16表示16位的浮点数，FP32表示32位浮点数。通常，浮点数包含3个域，分别为符号（sign）域，阶码（exponent）域和尾数（mantissa）域。上述每种浮点型数据表示法中，各个域的位宽都是固定不变的。例如，FP16包括1比特（bit）符号域，5bits阶码域和10bits尾数域；FP32包括1bit符号域，8bits阶码域和23bits尾数域；FP8可以包括两种，其中一种包括1bit符号域，5bits阶码域和2bits尾数域；另一种包括1bit符号域，4bits阶码域和3bits尾数域。其中，阶码域的位宽决定了浮点数能够表示的数值范围，尾数域的位宽决定了浮点数能够表示的数值精度。

随着人工智能（artificial intelligence，AI）混合精度训练和推理的快速发展，浮点数越来越多地应用到混合精度训练中。随着AI网络参数的规模急剧上涨，使用更少位宽的浮点数进行AI训练和推理，可以节约数据存储和数据转移开销。

然而，浮点数的整体位宽越小，其阶码域的位宽也越小，能够表示的数值范围也较小，往往不能满足AI训练和推理的需要，影响AI训练和推理的性能。

发明内容

本申请实施例提供一种浮点数处理方法、装置、计算设备和存储介质，可以扩大浮点数表示的数值的动态范围，满足AI训练和推理的需要。

第一方面，本申请实施例提供一种浮点数处理方法，该方法可以由计算设备执行，也可以由计算设备中的芯片、芯片系统或者电路执行。该浮点数处理方法可以包括：计算设备获取第一浮点数，其中，第一浮点数可以包括第一符号域、位宽指示域、第一阶码域和第一尾数域，第一符号域用于指示第一浮点数的符号，位宽指示域用于指示第一阶码域在第一浮点数的总位宽N中所占用的位宽D；位宽指示域还用于指示：第一尾数域表征的是第一浮点数的尾数，或，第一浮点数的阶码的偏移量。计算设备在获取第一浮点数之后，可以对第一浮点数进行解码，以得到上述的符号、阶码和尾数。

在本申请实施例中，第一浮点数除包括第一符号域、第一阶码域和第一尾数域之外，还可以包括位宽指示域，位宽指示域用于指示第一阶码域在浮点数的总位宽N中所占用的位宽D，在总位宽固定的前提下，第一阶码域的位宽和第一尾数域的位宽可以随着位宽指示域指示的数值动态变化，从而满足不同场景下对浮点数不同数值范围和精度的需求。位宽指示域还可以指示第一尾数域表征的是尾数还是阶码的偏移量，也就是说，第一尾数域可以用于表征尾数，也可以用于指示阶码，通过复用第一尾数域，既可以保证对浮点数的精度影响较小，又可以增加阶码的编码空间，从而扩大浮点数所表示的数值的动态范围，可以满足AI训练和推理的需要。

在一种可选的实现方式中，计算设备在获取第一浮点数之后，可以对第一浮点数进行解码，以得到第二浮点数。其中，第二浮点数可以包括第二符号域、第二阶码域和第二尾数域，第二符号域用于指示符号，第二阶码域用于指示阶码，第二尾数域用于指示尾数。

在一种可选的实现方式中，当第一浮点数的第一阶码域的位宽D不为0时，尾数可以是第一尾数域指示的值，阶码可以是第一阶码域指示的值；当第一浮点数的第一阶码域的位宽D为0时，第一阶码域不存在。此时，如果第一尾数域表征的是尾数，则阶码可以为第一预设阶码。如果第一尾数域表征的是阶码的偏移量，则阶码为利用该偏移量修正第二预设阶码得到的值，尾数可以是预设尾数。

上述实现方式中,当第一阶码域不存在时,如果第一尾数域表征的是尾数,阶码可以采用第一预设阶码,如果第一尾数域表征的是阶码的偏移量,则阶码为利用该偏移量修正第二预设阶码得到的值。利用偏移量修正第二预设阶码得到的值与第一阶码域指示的值属于不同的数值范围。例如,当第一浮点数的总位宽为8时,第一阶码域指示的值所属的数值范围可以是 $[-15, 15]$,利用偏移量修正第二预设阶码得到的值所属的数值范围可以是 $[-22, -16]$,因此,第一浮点数能够表示的阶码数值范围可以达到 $[-22, 15]$,与相关技术中8位浮点数表示方法可以表示的数值的动态范围相比,扩大了浮点数所表示的数值的动态范围。

在一种可选的实现方式中,位宽指示域的位宽DW与第一阶码域的位宽D负相关,在第一浮点数的总位宽固定时,可以避免由于位宽指示域和第一阶码域的位宽同时增加,造成第一尾数域位宽的跳变。尾数域的位宽决定浮点数据的精度,因此,该实现方式可以使第一浮点数所表示的数值的精度平滑变化,防止第一浮点数所表示的数值的精度发生跳变。

在一种可选的实现方式中,当位宽指示域的位宽DW为预设位宽,且位宽指示域指示的值为预设值时,第一尾数域表征的是偏移量。例如,在一种实施例,当第一浮点数的总位宽为8,位宽指示域的位宽DW为4时,如果位宽指示域指示的值为“0000”,则指示第一尾数域表征的是偏移量;或者,当第一浮点数的总位宽为8,位宽指示域的位宽DW为4时,如果位宽指示域指示的值为“0001”,则指示第一尾数域表征的是偏移量。

在一种可选的实现方式中,计算设备可以从存储器读取第一浮点数,或者,通过通信网络获取第一浮点数。在对第一浮点数进行解码得到符号、阶码和尾数之后,可以采用得到的符号、阶码和尾数进行计算。

在相关技术中,当浮点数的总位宽确定时,其阶码域和尾数域的位宽始终都是固定不变的,当计算过程需要更大的精度或者更大的数值范围时,只能选择总位宽更大的数据格式,但是,总位宽增大,意味着阶码域位宽和尾数域位宽同时增加,容易导致仅需要更大的精度时,增加的阶码域位宽浪费,仅需要更大的数值范围时,增加的尾数域位宽浪费,从而占用不必要的存储空间,增加浮点数的数据存储和数据转移的开销。在本申请实施例中,第一浮点数可以用于数据存储或数据转移,由于第一浮点数的第一阶码域和第一尾数域的位宽可以随着位宽指示域指示的数值动态变化,因此可以在不额外增加浮点数的总位宽的前提下,即不额外增加数据存储或数据转移成本的前提下,灵活满足各类场景下对浮点数的数值范围和数值精度的不同需求。

第二方面,本申请实施例提供一种浮点数处理方法,该方法可以由计算设备执行,也可以由计算设备中的芯片、芯片系统或者电路执行。该浮点数处理方法可以包括:计算设备获取符号、阶码和尾数,并基于符号、阶码和尾数,得到第一浮点数。第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域,第一符号域用于指示第一浮点数的符号,位宽指示域用于指示第一阶码域在第一浮点数的总位宽N中所占用的位宽D。位宽指示域还用于指示:第一尾数域表征的是第一浮点数的尾数,或,第一浮点数的阶码的偏移量。

在一种可选的实现方式中,当第一阶码域的位宽D为0时,第一阶码域不存在,若第一尾数域表征的是尾数,则阶码为第一预设阶码。

在一种可选的实现方式中,当第一阶码域的位宽D为0时,第一阶码域不存在,若第一尾数域表征的是阶码的偏移量,则尾数是预设尾数,阶码为利用该偏移量修正第二预设阶码得到的值。

在一种可选的实现方式中,当第一阶码域的位宽D不为0时,第一尾数域表征的是尾数,阶码为第一阶码域指示的值。

在一种可选的实现方式中,位宽指示域的位宽DW与位宽D负相关。

在一种可选的实现方式中,当位宽指示域的位宽DW为预设位宽,且位宽指示域指示的值为预设值时,第一尾数域表征的是阶码的偏移量。

第三方面,本申请实施例提供一种浮点数处理装置,该装置可以应用于计算设备。该浮点数处理装置可以包括:

浮点数获取模块,用于获取第一浮点数;第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;第一符号域用于指示第一浮点数的符号,位宽指示域用于指示第一阶码域在第一浮点数的总位宽N中所占用的位宽D;位宽指示域还用于指示:第一尾数域表征的是第一浮点数的尾数,或,第一浮点数的阶码的偏移量;

解码模块，用于对第一浮点数进行解码，以得到符号、阶码和尾数。

在一种可选的实现方式中，解码模块，具体可以用于：

对第一浮点数进行解码，以得到第二浮点数；第二浮点数包括第二符号域、第二阶码域和第二尾数域，第二符号域用于指示符号，第二阶码域用于指示阶码，第二尾数域用于指示尾数。

5 在一种可选的实现方式中，当位宽 D 为 0 时，第一阶码域不存在，若第一尾数域表征的是尾数，则阶码为第一预设阶码。

在一种可选的实现方式中，当位宽 D 为 0 时，第一阶码域不存在，若第一尾数域表征的是偏移量，则尾数是预设尾数，阶码为利用偏移量修正第二预设阶码得到的值。

10 在一种可选的实现方式中，当位宽 D 不为 0 时，第一尾数域表征的是尾数，阶码为第一阶码域指示的值。

在一种可选的实现方式中，位宽指示域的位宽 DW 与位宽 D 负相关。

在一种可选的实现方式中，当位宽指示域的位宽 DW 为预设位宽，且位宽指示域指示的值为预设值时，第一尾数域表征的是偏移量。

在一种可选的实现方式中，浮点数获取模块，具体可以用于：

15 从存储器读取第一浮点数，或者，通过通信网络获取第一浮点数。

在一种可选的实现方式中，上述浮点数处理装置还可以包括计算模块，计算模块可以用于：采用符号、阶码和尾数进行计算。

第四方面，本申请实施例提供一种浮点数处理装置，该装置可以应用于计算设备。该浮点数处理装置可以包括：

20 数据获取模块，用于获取符号、阶码和尾数；

编码模块，用于基于符号、阶码和尾数，得到第一浮点数；第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域；第一符号域用于指示第一浮点数的符号，位宽指示域用于指示第一阶码域在第一浮点数的总位宽 N 中所占用的位宽 D；位宽指示域还用于指示：第一尾数域表征的是第一浮点数的尾数，或，第一浮点数的阶码的偏移量。

25 在一种可选的实现方式中，当位宽 D 为 0 时，第一阶码域不存在，若第一尾数域表征的是尾数，则阶码为第一预设阶码。

在一种可选的实现方式中，当位宽 D 为 0 时，第一阶码域不存在，若第一尾数域表征的是偏移量，则尾数是预设尾数，阶码为利用偏移量修正第二预设阶码得到的值。

30 在一种可选的实现方式中，当位宽 D 不为 0 时，第一尾数域表征的是尾数，阶码为第一阶码域指示的值。

在一种可选的实现方式中，位宽指示域的位宽 DW 与位宽 D 负相关。

在一种可选的实现方式中，当位宽指示域的位宽 DW 为预设位宽，且位宽指示域指示的值为预设值时，第一尾数域表征的是偏移量。

35 第五方面，本申请实施例提供一种计算设备，该计算设备处理器和存储器；存储器上存储有计算机程序；处理器用于读取存储器中存储的计算机程序并执行上述第一方面或第二方面提供的任一种浮点数处理方法。

第六方面，本申请实施例提供了一种计算机可读存储介质，存储有计算机可执行指令，所述计算机可执行指令用于使计算机执行上述第一方面或第二方面提供的任一种浮点数处理方法。

40 第七方面，本申请实施例提供了一种计算机程序产品，包含有计算机可执行指令，所述计算机可执行指令用于使计算机执行上述第一方面或第二方面提供的任一种浮点数处理方法。

上述第二方面至第七方面中任一方面可以达到的技术效果可以参照上述第一方面中有益效果的描述，此处不再重复赘述。

附图说明

45 图 1 为本申请实施例的一种应用场景的示意图；

图 2 为本申请实施例提供的一种计算设备的结构示意图；

图 3 为本申请实施例提供的一种浮点数的有效位-阶码分布示意图；

图 4 为本申请实施例提供的一种浮点数处理方法的流程图；

图 5 为本申请实施例提供的一种解码器对第一浮点数进行解码的示意图;
图 6 为本申请实施例提供的另一种浮点数处理方法的流程图;
图 7 为本申请实施例提供的一种编码器对第一浮点数进行编码的示意图;
图 8 为本申请实施例提供的另一种浮点数处理方法的流程图;
图 9 为采用相关技术的浮点数进行 AI 训练的训练效果图;
图 10 为采用本申请实施例的浮点数进行 AI 训练的训练效果图;
图 11 为本申请实施例提供的一种浮点数处理装置的结构示意图;
图 12 为本申请实施例提供的另一种浮点数处理装置的结构示意图;
图 13 为本申请实施例提供的一种计算设备的结构示意图。

具体实施方式

为了使本申请实施例的目的、技术方案和优点更加清楚,下面将结合附图,对本申请实施例进行详细描述。本申请的实施方式部分使用的术语仅用于对本申请的具体实施例进行解释,而非旨在限定本申请。

在介绍本申请实施例提供的具体方案之前,对本申请中的部分用语进行解释说明,以便于本领域技术人员理解,并不对本申请中的用语进行限定。

(1) 整数(integer)编码:指采用固定长度的二进制对整数进行编码。例如,采用 3 个比特(bit)位的二进制编码 0~7,采用 4 个比特位的二进制编码 0~15,等等。

(2) 前缀码(prefix code)编码:也可以称为前缀编码。如果在一个编码方式中,任何一个编码都不是其他任何编码的前缀(最左子串),则该编码方式可以称为前缀编码。例如,不等长编码:1、01、001、0101;或者 00、01、10、1100、1101;以及等长编码:00、01、10、11,等,都属于前缀码编码。前缀编码可以保证对压缩文件进行解码时不产生二义性,确保正确解码。

前缀码编码可以包括常规前缀码编码和非常规前缀码编码。常规前缀码编码即采用较短的位宽编码较小的数据,采用较长的位宽编码较大的数据。非常规前缀码编码则相反,其采用较短的位宽编码较大的数据,采用较长的位宽编码较小的数据。本申请的一些实施例中,可以通过非常规前缀码编码对浮点数中的 Dot 域进行编码,具体请参考下述实施例中的描述。

本申请实施例中“多个”是指两个或两个以上,鉴于此,本申请实施例中也可以将“多个”理解为“至少两个”。“至少一个”,可理解为一个或多个,例如理解为一个、两个或更多个。例如,包括至少一个,是指包括一个、两个或更多个,而且不限制包括的是哪几个,例如,包括 A、B 和 C 中的至少一个,那么包括的可以是 A、B、C、A 和 B、A 和 C、B 和 C、或 A 和 B 和 C。“和/或”,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。另外,字符“/”,如无特殊说明,一般表示前后关联对象是一种“或”的关系。

除非有相反的说明,本申请实施例提及“第一”、“第二”等序数词用于对多个对象进行区分,不用于限定多个对象的顺序、时序、优先级或者重要程度。

浮点型数据表示法是一种科学计算法,在电气电子工程师学会(institute of electrical and electronics engineers, IEEE)754 二进制浮点数标准中,浮点数可以包含 3 个域,分别为符号(sign)域,阶码(exponent)域和尾数(mantissa)域。其中,阶码域的值表示某个基数的整数次指数,尾数域的值乘以某个基数的整数次指数,可以得到一个数据;符号域用于指示该数据的正负。例如,以二进制的数字 100.101(即十进制的 4.625)为例,该数据可以表示为 $(-1)^0 \times 2^2 \times 1.00101$,其中,底数-1 的指数 0 为浮点数中的符号域的值,用于表示数据的正负。底数 2 的指数 2 为浮点数中阶码域的值,表示小数点的移位数,比如,100.101 到 1.00101,小数点的移位数为 2。1.00101 为尾数,由于小数点前必须为 1,因此,浮点数中尾数域实际存储的值,可以只包括小数点后的数据,即 00101,从而可以节省出一个二进制位来保存更多的尾数。

示例性地,常用的浮点型数据表示法可以包括 FP16 和 FP32,FP16 表示 16 位的浮点数,FP32 表示 32 位浮点数。每种浮点型数据表示法中,各个域的位宽都是固定不变的。例如,FP16 包括 1 比特符号域,5bits 阶码域和 10bits 尾数域;FP32 包括 1 比特符号域,8bits 阶码域和 23bits 尾数域。其中,阶码域的位宽决定了浮点数能够表示的数值范围,尾数域的位宽决定了浮点数能够表示的数值精度。

随着AI网络参数的规模急剧上涨，使用更少位宽的浮点数进行AI训练和推理，可以节约数据存储和数据转移开销。随之出现了FP8浮点数，FP8浮点数包括两种，具体参见表1所示。

表 1

	Sign	Exponent	mantissa
E5M2	1bit	5bits	2bits
E4M3	1bit	4bits	3bits

如表1所示，FP8浮点数可以包括两种，一种为E5M2，另一种为E4M3。E5M2的浮点数包括1bit符号域，5bits阶码域和2bits尾数域；E4M3的浮点数包括1bit符号域，4bits阶码域和3bits尾数域。浮点数中阶码域的位宽决定了浮点数能够表示的阶码范围。E5M2的浮点数能够表示的阶码范围为[-16, 15]，而E4M3的浮点数能够表示的阶码范围更小，为[-9, 8]。

由于上述两种FP8浮点数能够表示的数值的动态范围均较小，往往不能满足AI训练和推理的需要，影响AI训练和推理的性能。而采用传统的浮点型数据表示法，只能选择总位宽更大的数据格式，例如选择FP16的浮点数或FP32的浮点数，这样容易导致总位宽较大的浮点数中阶码域位宽或尾数域位宽的浪费，使浮点数占用不必要的存储空间，大大增加浮点数的数据存储和数据转移的开销。

基于此，本申请实施例提供一种浮点数处理方法。本申请实施例提供的浮点数，除包括第一符号域之外，还可以包括位宽指示域（即Dot域）、第一阶码域和第一尾数域，其中，Dot域用于指示第一阶码域在浮点数的总位宽N中所占用的位宽D。当位宽D为非0时，第一阶码域用于指示浮点数的阶码，第一尾数域用于指示浮点数的尾数。在相同的总位宽下，第一阶码域的位宽和第一尾数域的位宽可以随着Dot域的数值动态变化，满足不同场景下对浮点数不同数值范围和精度的需求。当位宽D为0时，Dot域还可以用于指示：第一尾数域表征的是浮点数的尾数，还是浮点数的阶码的偏移量。通过复用第一尾数域，既可以保证对浮点数的精度影响较小，又可以增加阶码的编码空间，从而扩大浮点数所表示的数值的动态范围。例如，以总位宽N为8的浮点数为例，上述E4M3的浮点数表示的数值的动态范围为[-9, 8]，E5M2的浮点数能够表示的数值的动态范围为[-16, 15]，而本申请实施例提供的HiF8_DML的浮点数能够表示的数值的动态范围为[-22, 15]，大幅度扩大了浮点数所表示的数值的动态范围，可以满足AI训练和推理的需要。

本申请实施例提供的浮点数处理方法可以应用于计算设备中，广泛适用于科学研究、工程、金融、航空航天、医疗等各行各业，这些行业每天都有大量的数据需要存储、计算和传输。图1示例性地示出了本申请实施例提供的一种应用场景的示意图，如图1所示，在该应用场景中，计算设备100、计算设备200与计算设备300可以通过网络建立通信连接。其中，该网络可以是有线网络，也可以是无线网络，例如，无线保真（wireless fidelity，WiFi）、蓝牙、移动网络等。计算设备100、计算设备200和计算设备300可以是计算机、服务器、智能可穿戴设备、智能家居、平板电脑、笔记本电脑、车载终端、智能手机等任意的电子设备。当计算设备是服务器时，可以是一台服务器，也可以是由多个服务器构成的服务器集群，或者是一个云计算服务中心，等等。计算设备100、计算设备200和计算设备300之间建立通信连接，可以进行基于浮点数的数据存储、计算和传输等。

需要说明的是，在实际的应用场景中，系统中可以包含多于3个计算设备，也可以包含少于3个计算设备，本申请对此不作限定。

计算设备100、计算设备200和计算设备300的内部结构可以相同。以计算设备100为例进行说明，如图2所示。计算设备100内部可以包括处理器110和存储器120。示例性地，处理器110可以是但不限于中央处理单元（central processing unit，CPU）、高性能计算（high performance computing，HPC）业务加速芯片、图像处理单元（graphics processing unit，GPU）或AI领域的嵌入式神经网络处理单元（neural network processing units，NPU）等芯片。处理器110中可以包括解码器（decoder）111、编码器（encoder）112和计算单元113，计算单元113的数量可以是一个，也可以是多个。

示例性地，处理器110中可以包括如下计算单元中的部分或全部：标量计算单元、向量计算单元、矩阵计算单元或张量计算单元。

下面对不同计算单元分别进行介绍。

一、标量计算单元：标量，又称纯量，只有大小，没有方向。针对标量计算的电路称为标量计算单元。标量计算多用于通用计算。本申请实施例中，可以在CPU多级流水线的执行单元（execution unit，

EXU) 部分, 或其他类似功能的处理器的标量计算部分, 嵌入基于 HiF8_DML 数据格式的算数逻辑单元 ALU (Arithmetic Logic Unit)。

二、向量计算单元: 向量, 又称矢量, 通常指长度大于 1 的一维数组。针对向量计算而特殊设计的具有一定并行度的计算单元, 称为向量计算单元, 如单指令多数据流 (single instruction multiple data, SIMD) 处理机。向量计算单元多用于 HPC 高性能计算和 AI 机器学习等领域, 包含如线性规划、傅里叶变换、滤波计算以及线性代数、偏微分方程、积分等数学问题的求解。本申请实施例中, 可以在向量计算加速单元中, 或向量处理机中, 嵌入基于 HiF8_DML 数据格式的算数执行单元 (vector unit)。

三、矩阵计算单元: 矩阵, 是一个按照长方阵列排列的 2 维数组。针对矩阵计算而特殊设计的具有相应并行度的计算单元, 称为矩阵计算单元, 如脉动阵列 (systolic array) 处理机。矩阵计算单元多用于 HPC 高性能计算和 AI 机器学习等领域的矩阵计算, 包含如矩阵乘、矩阵求逆、矩阵分解等。本申请实施例中, 可以在矩阵计算加速单元中, 嵌入基于 HiF8_DML 数据格式的矩阵单元 (matrix unit)。

四、张量计算单元: 张量, 是维数超过 2 的多维数组, 常见的是 3 维数组。针对张量计算而特殊设计的具有相应并行度的计算单元, 称为张量积算单元。张量计算单元多用于 AI 机器学习领域, 如卷积操作。本申请实施例中, 可以在张量计算加速单元中, 嵌入基于 HiF8_DML 数据格式的张量单元 (tensor unit)。

示例性地, 当计算设备 100 进行通用计算、高性能计算或者 AI 训练时, 需要使用到大量浮点数据。此时, 计算设备 100 可以通过解码器 111, 基于本申请实施例提供的一种浮点数处理方法, 对获取的浮点数进行解码, 得到解码后的数据。示例性地, 该浮点数可以从计算设备 100 本地的存储器 120 中读取, 也可以通过网络从计算设备 200 或计算设备 300 处获取, 或者, 从网络中的其他设备处获取。解码器 111 对获取的浮点数进行解码, 得到解码后的数据之后, 可以将解码后的数据传输至计算单元 113, 通过计算单元 113 完成相应的计算。计算单元 113 将计算结果传输至编码器 112, 编码器 112 将计算结果重新编码为浮点数, 该浮点数可以用于进行数据存储和数据转移。例如, 处理器 110 可以将通过编码器 112 编码得到的浮点数保存至存储器 120 中。

可选地, 图 1 中计算设备 200 和计算设备 300 的结构和功能具体可以参考图 2 所示的计算设备 100, 在一些可能的实施例中, 计算设备 100、计算设备 200 和计算设备 300 可以包括比图 2 所示更多或者更少的部件, 本申请实施例对此不做具体限定。

为更方便理解, 下面先对本申请实施例提供的浮点数的数据格式进行介绍。表 2 示出了本申请实施例提供的浮点数的数据格式。

表 2

域 (Region)	第一符号域 (Sign)	Dot域	第一阶码域	第一尾数域
位宽 (Width)/bit	1	DW	D	N-1- DW-D

如表 2 所示, 本申请实施例提供的浮点数可以包括第一符号域、Dot 域、第一阶码域和第一尾数域, 其中, Dot 域用于指示第一阶码域在浮点数的总位宽 N 中所占用的位宽 D。其中, N 为大于 1 的整数, D 为大于或者等于 0 的整数。下述实施例以二进制, 且浮点数的总位宽 N 是 8 为例进行说明, 首先结合表 2 和表 3, 对浮点数的每个域进行详细介绍。

表 3

域 (Region)	第一符号域 (Sign) 位宽	Dot域 位宽DW: 数值 (Values)	第一阶码域 位宽D: 数值 (Values)	第一尾数域 位宽
	1	2: [2、3、4]	D: $\pm[2,15]$	5-D
	1	3:1	D: ± 1	4-D
	1	4:0	D: 0	3-D
	1	4: DML		3

一、第一符号域: 第一符号域也可以称为符号位, 如表 3 所示, 符号域位于 Dot 域之前, 在浮点数的总位宽 N 中占用 1bit, 用于表示数据的正负。默认 0 表示正, 1 表示负, 也可以根据实际需求用 0 表

示负，1表示正，本申请对此不作限定。

二、Dot域：在浮点数的总位宽N中占用DW个比特位，DW的取值为2~4。Dot域表示的数值（或编码值）用于指示第一阶码域在浮点数的总位宽N中占用的位宽D，即Dot域的数值为D。当D为非0时，第一阶码域用于指示浮点数的阶码；第一尾数域用于指示浮点数的尾数。当位宽D为0时，位宽指示域还可以用于指示：第一尾数域表征的是浮点数的尾数，还是浮点数的阶码的偏移量。

可选地，Dot域的编码方式可以采用前缀码编码。前缀码编码可以包括常规前缀码编码和非常规前缀码编码。常规前缀码编码即采用较短的位宽表示较小的数值，采用较长的位宽表示较大的数值。非常规前缀码编码则相反，其采用较短的位宽表示较大的数值，采用较长的位宽表示较小的数值。

在一些实施例中，可以采用非常规前缀码编码方式对Dot域进行编码，即采用较长的位宽表示较小的数值，采用较短的位宽表示较大的数值。示例性地，当位宽DW为第一位宽值时，用Dot域占用的位宽DW1编码D1个数值，第一数值域的位宽D属于D1个数值中的任意一个数值；当位宽DW为第二位宽值时，用Dot域占用的位宽DW2编码D2个数值，第一数值域的位宽D属于D2个数值中的任意一个数值。其中，DW1小于DW2，D1个数值中的最小数值大于D2个数值中的最大数值。D1、D2为大于或者等于0的整数。

示例性地，Dot域在浮点数的总位宽N中占用2~4bits，即Dot域的位宽（width）为2~4，具体的编码方式可以如表4所示。

表 4

位宽（Width）	编码（Coding）	数值（Value）
2	11	4
	10	3
	01	2
3	001	1
4	0001	0
	0000	DML

如表4所示，当Dot域的位宽为2时，可以采用2bits的位宽来编码表示2、3、4三个数值中的任意一个数值。例如，编码（coding）“11”可以表示数值（value）“4”，编码“10”可以表示数值“3”，编码“01”可以表示数值“2”。当Dot域的位宽为3时，可以采用3bits的位宽来编码表示数值1。例如，编码“001”可以表示数值“1”。数值1、2、3、4用于指示第一阶码域的位宽D，第一阶码域用于表征浮点数的阶码。当Dot域的位宽为4时，可以采用4bits的位宽来编码表示数值0，此时，第一阶码域不存在。

位宽指示域采用非常规前缀编码，位宽指示域的位宽DW与位宽指示域指示的值负相关，也就是说，位宽指示域的位宽DW与第一阶码域的位宽D负相关，在浮点数的总位宽固定时，可以避免由于位宽指示域和第一阶码域的位宽同时增加，造成第一尾数域位宽的跳变。尾数域的位宽决定浮点数据的精度，因此，可以使第一浮点数所表示的数值的精度平滑变化，防止第一浮点数所表示的数值的精度发生跳变；尤其在阶码中心附近时，可以平滑尾数域位宽的跳变，即平滑阶码中心附近数值的精度跳变。

当Dot域的位宽为4时，Dot域还用于指示第一尾数域表征的是浮点数的尾数，还是浮点数的阶码的偏移量。例如，在一种实施例中，编码“0001”可以表示数值“0”，此时，第一阶码域的位宽D为0比特，第一尾数域表征的是浮点数的尾数，编码“0000”可以表示该浮点数为DML数据，DML数据指低于正常数的（subnormal）数值或非正常数（denormal）的数值，此时，第一尾数域表征的是浮点数的阶码的偏移量。也就是说，当Dot域的位宽为4时，如果Dot域为预设值“0000”，则Dot域指示该浮点数是DML数据，或者说，指示第一尾数域表征的是浮点数的阶码的偏移量。在另一种实施例中，也可以采用编码“0000”可以表示数值“0”，此时，第一阶码域的位宽D为0比特，第一尾数域表征的是浮点数的尾数，编码“0001”可以表示该浮点数为DML数据，即Dot域为预设值“0001”，则Dot域指示该浮点数是DML数据，或者说，指示第一尾数域表征的是浮点数的阶码的偏移量。由上述描述可知，位宽指示域可以描述为Dot=[0,4]&DML，表示：除DML标志之外，位宽指示域的编码表示的数值D的范围是0~4，位宽指示域用于指示第一阶码域在浮点数的总位宽N中占用的位宽D，也就是说，第一阶码域的位宽D的范围为0~4个比特。

三、第一阶码域：用于表征浮点数的阶码，在浮点数的总位宽N中占用的位宽为D，D的取值范

围为 0~4 个比特。

示例性地，第一阶码域 Es 用于表征浮点数的阶码。假设浮点数的阶码的数值范围为 E，第一阶码域 Es 所表示的数值 Ev 属于该数值范围为 E，该数值范围 E 可以通过如下公式 1 确定：

$$E = (-1)^{Se} \times [2^{D-1}, (2^D - 1)] \quad \text{公式 1}$$

- 5 其中，Se 为数值 Ev 的符号位，也可称之为浮点数的阶码符号位，Se 在阶码域的位宽 D 中占用 1 位，用于表示第一阶码域的数值的正负。在一种实施例中，Se 为 0 表示数值 Ev 为正，Se 为 1 表示数值 Ev 为负。在其他实施例中，也可以采用 Se 为 0 表示数值 Ev 为正，Se 为 1 表示数值 Ev 为负，本申请对此不作具体限定。

- 10 结合公式 1 和表 3 所示，当第一阶码域的位宽 D 为 0 时，第一阶码域所表示的数值 Ev 为 0，即浮点数的阶码的数值范围 E 为 0。当第一阶码域的位宽 D 为 1 时，第一阶码域所表示的数值 Ev 可以是 1 或 -1，即浮点数的阶码的数值范围 E 为 ± 1 。当第一阶码域的位宽 D 为 2~4 时，第一阶码域所表示的数值 Ev 可以是 [-2, -15] 或 [2, 15] 中的任意值，即浮点数的阶码的数值范围 E 为 $\pm [2, 15]$ 。由此可见，第一阶码域能表示的浮点数的阶码的数值范围 E 为 [-15, 15]。

- 15 在一种可选的实施例中，当位宽 D 为 0 时，且非 DML 时，表示浮点数的阶码为 0。当位宽 D 为非 0 时，即位宽 D 为 1~4 中的任一数值时，第一阶码域可以采用有符号数据 (signed magnitude) 编码，也可以称为符号位 Se 尾随原码的编码方式。该表示法是在数值前面增加一位符号位，即原码的最高位为符号位，用于表示数值的正负，其中，符号位为 0 表示正数，符号位为 1 表示负数。原码中除符号位外的其余位用于表示数值的大小，即原码的幅值。例如，原码 1001，表示 -1，0011，表示 +3。

- 20 在本申请实施例中，当位宽 D 为非 0 时，浮点数中的第一阶码域可以采用阶码符号位 Se 跟随原码幅值的编码方式，可以表示为 Es: {Se+Mag[2:end]}，其中，阶码符号位 Se 为从初始原码中提取出的符号位，用于表示第一阶码域的数值的正负；Mag 用于表示第一阶码域的数值的幅值，end=D-1，Mag[2:end] 用于表示第一阶码域的第 2 位至第 D-1 位的数值。针对不同的位宽 D，第一阶码域的数值的幅值 Mag 的最高位 b1 始终为 1 (即 1'b1)，因此，最高位 1'b1 在编码时可以不占位宽，即最高位 1'b1 隐藏，不进行实际存储，在后续解码时可以直接补充上最高位 1'b1，得到规格化的浮点数中阶码域的编码值 Ei: {Se+1'b1+Mag[2:end]}。其中，规格化的浮点数指符合 IEEE754 二进制浮点数标准的浮点数，规格化的浮点数可以用于进行浮点数计算。由于阶码域的编码值 Ei 的最高位不需要在第一阶码域中占位宽，不进行实际存储，因此可以大大节省存储空间，降低数据存储和数据转移的成本。

- 30 示例性地，如表 5 所示，当 Dot 域为 DML 标志时，说明该浮点数为 DML 数据，此时，第一尾数域的位宽为 3，即第一尾数域采用 3bits 来表征浮点数的阶码的偏移量。当 Dot 域指示第一阶码域的位宽 D=0 时，浮点数中的第一阶码域的编码 Es 为 None，解码得到规格化的浮点数的阶码域的值 Ei 为 0，表示该浮点数的阶码的值 Ev 为 0，此时，Dot 域的位宽为 4，第一尾数域的位宽为 3，即第一尾数域采用 3bits 来表征浮点数的尾数。

- 35 当 Dot 域指示第一阶码域的位宽 D=1 时，浮点数中的第一阶码域的编码 Es = {Se}，而该第一阶码域的解码值，即规格化的浮点数的阶码域的值 Ei = {Se, 1}，其中，1 为 Mag 中的最高位的值。如上所述，Mag 中的最高位始终为 1。当位宽 D=1 时，第一阶码域所表示的浮点数的阶码 Ev 可以是 1 或 -1，即浮点数的阶码 Ev 的数值范围可以是 ± 1 。此时，Dot 域的位宽为 3，第一尾数域的位宽为 3，即第一尾数域采用 3bits 来表征浮点数的尾数。

- 40 当 Dot 域指示第一阶码域的位宽 D=2 时，浮点数中的第一阶码域的编码 Es = {Se, [2]}，其中，[2] 表示 Mag 中的第 2 位的值。该第一阶码域的解码值，即规格化的浮点数的阶码域的值 Ei = {Se, 1, [2]}，其中，1 为 Mag 中的最高位的值。当位宽 D=2 时，第一阶码域所表示的浮点数的阶码 Ev 可以是 [-2, -3] 或 [2, 3] 中的任意值，即浮点数的阶码 Ev 的数值范围可以是 $\pm [2, 3]$ 。此时，Dot 域的位宽为 2，第一尾数域的位宽为 3，即第一尾数域采用 3bits 来表征浮点数的尾数。

- 45 当 Dot 域指示第一阶码域的位宽 D=3 时，浮点数中的第一阶码域的编码 Es = {Se, [2:3]}，其中，[2:3] 表示 Mag 中的第 2 位和第 3 位的值。该第一阶码域的解码值，即规格化的浮点数的阶码域的值 Ei = {Se, 1, [2:3]}，其中，1 为 Mag 中的最高位的值。当位宽 D=3 时，第一阶码域所表示的浮点数的阶码 Ev 可以是 [-4, -7] 或 [4, 7] 中的任意值，即浮点数的阶码 Ev 的数值范围可以是 $\pm [4, 7]$ 。此时，Dot 域的位宽为 2，第一尾数域的位宽为 2，即第一尾数域采用 2bits 来表征浮点数的尾数。

当 Dot 域指示第一阶码域的位宽 D=4 时，浮点数中的第一阶码域的编码 Es = {Se, [2:4]}，其中，

[2:4]表示 Mag 中的第 2 位、第 3 位和第 4 位的值。该第一阶码域的解码值，即规格化的浮点数的阶码域的值 $E_i = \{Se, 1, [2:4]\}$ ，其中，1 为 Mag 中的最高位的值。当位宽 $D=4$ 时，第一阶码域所表示的浮点数的阶码 E_v 可以是 $[-8, -15]$ 或 $[8, 15]$ 中的任意值，即浮点数的阶码 E_v 的数值范围可以是 $\pm[8, 15]$ 。此时，Dot 域的位宽为 2，第一尾数域的位宽为 1，即第一尾数域采用 1bit 来表征浮点数的尾数。

- 5 由此可见，当 D 大于 1 时，浮点数中的第一阶码域的编码 $E_s = \{Se + Mag [2:end]\}$ ，Mag [2:end] 包括 Mag 中除最高位 1'b1 外的其余位，Mag [2:end] 在第一阶码域中占用的位宽为 $D-1$ 。

表 5

D	DML	0	1	2	3	4
E_s	/	None	Se	Se, [2]	Se, [2:3]	Se, [2:4]
E_i	/	0	Se, 1	Se, 1, [2]	Se, 1, [2:3]	Se, 1, [2:4]
E_v	/	0	± 1	$\pm[2, 3]$	$\pm[4, 7]$	$\pm[8, 15]$
第一尾数域的位宽	3	3	3	3	2	1

- 四、第一尾数域：用于表征浮点数的尾数，或者浮点数的阶码的偏移量。第一尾数域在浮点数中占用的位宽为 $(N-1-DW-D)$ bits。示例性地，Dot 域用于指示第一尾数域表征的是浮点数的尾数，还是浮点数的阶码的偏移量。在一些实施例中，当 Dot 域的位宽 DW 为 4，且 Dot 域的值为“0001”时，用于指示第一尾数域表征的是浮点数的尾数；当 Dot 域的位宽 DW 为 4，且 Dot 域的值为“0000”时，用于指示第一尾数域表征的是浮点数的阶码的偏移量。在另一些实施例中，也可以采用相反的设置，当 Dot 域的位宽 DW 为 4，且 Dot 域的值为“0000”时，用于指示第一尾数域表征的是浮点数的尾数；当 Dot 域的位宽 DW 为 4，且 Dot 域的值为“0001”时，用于指示第一尾数域表征的是浮点数的阶码的偏移量。

当第一尾数域表征的是浮点数的尾数时，第一尾数域用于存储小数点后的数值，比如，如果存储 1.xxx 的小数位，假设整数位 1'b1 隐藏。例如，如果第一尾数域中保存的编码是 10011，则第一尾数域的解码值 $M = 0.10011$ ，第一尾数域表征的数值是 1.M，即 1.10011。

- 如表 3 所示，当第一尾数域表征的是浮点数的阶码的偏移量时，第一尾数域的位宽为 3，即第一尾数域采用 3bits 编码 0~7 的整数值 M ，用于表征 $M-23$ 的阶码值，其中， M 可以称为阶码的偏移量。由此可见，第一尾数域可以补充表征阶码的数值范围 $[-23, -16]$ 。其中，第一尾数域的 3bits 均为 0 时，即 $M=0$ ， $M-23=-23$ 时，浮点数用于表示特殊值（将在下文中详细描述）。因此，第一尾数域可以表征的阶码的数值范围 $[-22, -16]$ 。与上述第一阶码域（如表 5 所示）结合后，本申请实施例的提供的 HiF8_DML 浮点数，可以表征的浮点数的阶码范围为 $[-22, 15]$ 。当 HiF8_DML 表征的浮点数的阶码为 15 时，所表示的浮点数的值 X 最大，可以近似为 $X_{\max\text{-pos}} = 2^{15} = 32768$ 。当 HiF8_DML 表征的浮点数的阶码为 -22 时，所表示的浮点数的值 X 最小，可以近似为 $X_{\min\text{-pos}} = 2^{-22} \approx 2.38 \times 10^{-7}$ 。

当第一尾数域表征的是浮点数的阶码的偏移量时，对应的浮点数为 DML 数据，该浮点数的值 X 可以表示为：

$$X = (-1)^s \times 2^{M-23}$$

- 30 其中， S 为浮点数的符号域的值， M 为浮点数的第一尾数域表征的数值。此时，浮点数的尾数采用默认值 1。

当第一尾数域表征的是浮点数的尾数时，对应的浮点数为正常数据，该浮点数的值 X 可以表示为：

$$X = (-1)^s \times 2^{E_v} \times 1.M$$

- 35 其中， S 为浮点数的符号域的值， E_v 为浮点数的第一阶码域表征的数值，即浮点数的阶码。 M 为浮点数的第一尾数域表征的数值，1.M 为浮点数的尾数。1.M 中的 1 也视为一个有效位，可见，浮点数的尾数的有效位的位数为第一尾数域的位宽+1。

综上所述，浮点数的值 X 可以表示为：

$$X = \begin{cases} (-1)^s \times 2^{E_v} \times 1.M & \text{正常数据} \\ (-1)^s \times 2^{M-23} & \text{DML 数据} \end{cases}$$

本申请实施例提供的 HiF8_DML 浮点数，还可以支持四个特殊值，分别是不区分正负的 0 (Zero)，非数值 (not a number, NaN)，正无穷 (+inf)，负无穷 (-inf)。四个特殊值可以由 HiF8_DML 能表示的 4 个边界值来表达。

- 40 (1) Zero: 当浮点数的符号域 S 的值为 0，Dot 域为 0000，也就是说，Dot 域指示该浮点数为 DML

数据,即第一尾数域表征的是浮点数的阶码的偏移量,且第一尾数域的3bits均为0时,对应的浮点数可以表示 ± 0 ,即Zero。上述描述可以概括为,HiF8=8'b 0 0000 000时,表示的数值X=Zero。

(2) NaN: 当浮点数的符号域S的值为1, Dot域为0000,也就是说, Dot域指示该浮点数为DML数据,即第一尾数域表征的是浮点数的阶码的偏移量,且第一尾数域的3bits均为0时,对应的浮点数可以表示非数值,即NaN。上述描述可以概括为,HiF8=8'b 1 0000 000时,表示X=NaN。

(3) 正无穷: 当浮点数的第一阶码域的位宽为4,第一尾数域的位宽为1,且第一阶码域的值为0111,第一尾数域的值1时,如果浮点数的符号域S的值为0,则对应的浮点数可以表示+inf。上述描述可以概括为,第一阶码域Es=4'b0111=15,第一尾数域M=1'b1,即HiF8=8'b 0 11 0111 1时,表示X=+inf。

(4) 负无穷: 当浮点数的第一阶码域的位宽为4,第一尾数域的位宽为1,且第一阶码域的值0111,第一尾数域的值1时,如果浮点数的符号域S的值为1,则对应的浮点数可以表示-inf。上述描述可以概括为,第一阶码域Es=4'b0111=15,第一尾数域M=1'b1,即HiF8=8'b 1 11 0111 1时,表示X=-inf。

图3是本申请实施例提供的一种HiF8_DML的浮点数的有效位精度(significant precision)-阶码分布示意图,其中,有效位的大小可以表示浮点数的精度大小,当第一尾数域用于表征浮点数的尾数时,尾数的有效位可以是第一尾数域的位宽加1。如图3所示,HiF8_DML具有锥型精度特征,最高可以提供4位有效位的精度,同时HiF8_DML具有[-22, 15]的阶码范围,远大于FP8的[-9, 8]或[-16, 15]的阶码范围,几乎等同于FP16的[-24, 15]的阶码范围,可见,HiF8_DML可以扩大浮点数所表示的数值的动态范围。并且,阶码中心附近的数值精度明显高于远离阶码中心的数值。例如,当阶码的值为[-3, 3]时,尾数的有效位最高,为4位有效位,当阶码的值在 $\pm[4, 7]$ 之间时,尾数的有效位为3位;当阶码的值在 $\pm[8, 15]$ 之间时,尾数的有效位为2位;当阶码的值小于-15时,尾数的有效位为1位,此时,尾数为默认值1。

本申请实施例提供的HiF8_DML可用于数据存储或数据转移。相对应地,本申请实施例还提供一种浮点数处理方法,该方法可以由如2中所示的处理器110执行。如图4所示,该方法可以包括如下步骤:

S401, 获取第一浮点数。

处理器可以从处理器所属计算设备本地的存储器中读取第一浮点数,或者,可以通过网络从其他计算设备获取第一浮点数。第一浮点数可以包括第一符号域、位宽指示域、第一阶码域和第一尾数域。其中,第一符号域用于指示第一浮点数的符号,即指示第一浮点数的正负,位宽指示域用于指示第一阶码域在第一浮点数的总位宽N中所占用的位宽D;位宽指示域还可以用于指示:第一尾数域表征的是第一浮点数的尾数,还是第一浮点数的阶码的偏移量。

S402, 对第一浮点数进行解码,以得到符号、阶码和尾数。

在一些实施例中,可以对第一浮点数进行解码,以得到第二浮点数;第二浮点数可以包括第二符号域、第二阶码域和第二尾数域,第二符号域用于指示上述符号,第二阶码域用于指示上述阶码,第二尾数域用于指示上述尾数。

示例性地,可以将第一浮点数的第一符号域的值作为第二浮点数的第二符号域的值,基于第一浮点数的第一阶码域和第一尾数域的值,确定第二浮点数的第二阶码域和第二尾数域。

当第一阶码域的位宽D为0时,如果位宽指示域指示,第一尾数域表征的是第一浮点数的尾数,则可以将第一尾数域指示的值,作为第二浮点数的第二尾数域的指示的值,并将第一预设阶码作为第二浮点数的第二阶码域指示的值,其中,第一预设阶码可以是-15。如果位宽指示域指示,第一尾数域表征的是第一浮点数的阶码的偏移量,则可以将预设尾数作为第二浮点数的第二尾数域指示的值,并采用第一尾数域的值对第二预设阶码进行修正,得到第二浮点数的阶码域的值。其中,预设尾数可以是1,第二预设阶码可以是-23。

当第一阶码域的位宽D为非0时,第一阶码域用于表征第一浮点数的阶码,第一尾数域用于表征第一浮点数的尾数。此时,可以基于位宽指示域,在第一浮点数中确定出第一阶码域和第一尾数域,然后将第一尾数域指示的值,作为第二浮点数的第二尾数域指示的值,并将第一阶码域指示的值,作为第二浮点数的第二阶码域指示的值。

在一些实施例中,处理器中的解码器可以用于将第一浮点数转换为第二浮点数。示例性地,解码器

的工作原理示意图可以如图 5 所示,该解码器也可以称为 HiF8_DML 解码器。如图 5 所示,第一浮点数依次包括符号域、Dot 域、第一阶码域和第一尾数域,第一浮点数的总位宽 N 为 8。其中,符号域占用 1bit,是第一浮点数中的最高位。Dot 域占用 2~4bits,第一阶码域占用 0~4 bits,第一尾数域占用 1~3bits。首先,对于浮点数中的符号域,解码器可以直接读取并输出其数值, $S=0$ 或者 1。其次,解码器可以

5 基于第一浮点数中的第二位、第三位或第四位的数值,确定 Dot 域的位宽。例如,如果第一浮点数中的第二位的数值为“1”,则可以确定 Dot 域的位宽为 2;如果第一浮点数中的第二位的数值为“0”,且第三位的数值为“1”,则可以确定 Dot 域的位宽也为 2;如果第一浮点数中的第二位和第三位的数值均为“0”,且第四位的数值为“1”,则可以确定 Dot 域的位宽为 3;如果第一浮点数中的第二位、第三位和第四位的数值均为“0”,则可以确定 Dot 域的位宽为 4。

10 如果 Dot 域的位宽为 4,则可以根据 Dot 域的值,确定第一尾数域表征的是第一浮点数的尾数,还是第一浮点数的阶码的偏移量。

如果第一尾数域表征的是第一浮点数的尾数,说明第一浮点数为正常数据。解码器可以读取第一尾数域的值,将第一尾数域的值作为第二浮点数的第二尾数域的值,并将第一预设阶码作为第二浮点数的第二阶码域的值。其中,第一预设阶码可以是-15。

15 如果第一尾数域表征的是第一浮点数的阶码的偏移量,说明第一浮点数为 DML 数据。解码器可以读取第一尾数域的值,利用第一尾数域的值修正第二预设阶码,得到第二浮点数的阶码域的值。例如,第二预设阶码可以是-23,当第一尾数域的值 M 时,第二浮点数的第二阶码域的值可以是 $M-23$ 。解码器可以将预设尾数 1 作为第二浮点数的第二尾数域表示的值。

如果 Dot 域的位宽为 2 或 3,则解码器可以提取出 Dot 域,并通过对 Dot 域进行多路复用(multiplexer, MUX)操作,基于预设的编码规则,对 Dot 域进行解码,得到 Dot 域的值 D。其中,预设的编码规则可以如上文中的表 4 所示,编码“11”可以表示数值(value)“4”,编码“10”可以表示数值“3”,编码“01”可以表示数值“2”,编码“001”可以表示数值“1”。

解码器可以基于 Dot 域的值 D,在 Dot 域后提取出第一阶码域和位于第一阶码域之后的第一尾数域,并将第一阶码域指示的值,作为第二浮点数的第二阶码域指示的值;将第一尾数域指示的值,作为第二浮点数的第二尾数域指示的值。至此,解码器可以根据上述的第一浮点数中第一阶码域和第一尾数域的值以及第一符号域的值,确定第二浮点数的第二阶码域和第二尾数域的值以及第二符号域的值。

例如,输入解码器的第一浮点数为“11010110”,总位宽 N 为 8bits,第一位“1”为符号域,第 2 位至第 3 位“10”为 Dot 域,根据预设的编码规则,解码得到该 Dot 域的值 D 为 3,即确定第一阶码域的位宽为 3bits,因此,解码器可以准确地在第一浮点数中提取出第 4 位至第 6 位“101”为第一阶码域,第一阶码域用于表征第一浮点数的阶码。剩余的 7 位至第 8 位“10”为第一尾数域,第一尾数域用于表征第一浮点数的尾数。基于此,可以解码得到第二浮点数的阶码域表示的值为 5,第二浮点数的尾数域表示的值为 0.10。

在另一些实施例,也可以对第一浮点数进行解码,直接得到符号、阶码和尾数。示例性地,可以根据第一浮点数的第一符号域的值确定符号。例如,如果第一符号域的值 D 为 0,表示符号为正,如果第一符号域的值 D 为 1,表示符号为负。

35 当第一阶码域位宽 D 为 0 时,说明第一阶码域不存在,此时,如果 Dot 域指示第一尾数域表征的是第一浮点数的尾数,则阶码可以是第一预设阶码-15。如果 Dot 域指示第一尾数域表征的是第一浮点数的阶码的偏移量,则阶码可以是利用该偏移量修正第二预设阶码-23 得到的值,尾数可以是预设尾数 1。

40 当第一阶码域位宽 D 不为 0 时,尾数为第一浮点数的第一尾数域指示的值,阶码为第一浮点数的第一阶码域指示的值。

在得到符号、阶码和尾数之后,可以基于符号、阶码和尾数进行计算。示例性地,在另一些实施例中,如图 6 所示,处理器执行的浮点数处理方法可以包括如下步骤:

S601,获取第一浮点数。

45 S602,对第一浮点数进行解码,以得到符号、阶码和尾数。

S603,采用符号、阶码和尾数进行计算,得到计算结果。

在一些实施例中,处理器中包括解码器,处理器可以通过解码器对第一浮点数进行解码,得到符号、阶码和尾数。解码器可以将解码得到的符号、阶码和尾数传输至与处理器中的计算单元,该计算单元接

收符号、阶码和尾数,并进行相应的计算,得到计算结果。其中,计算结果可以是与上述第二浮点数采用相同编码方式的浮点数,也就是说,计算结果可以包括符号域、阶码域和尾数域,符号域用于指示计算结果的符号,阶码域用于指示计算结果的阶码,尾数域用于指示计算结果的尾数。

S604,基于计算结果的阶码和尾数,将计算结果转换为第三浮点数。

- 5 第三浮点数是与第一浮点数采用相同编码方式的浮点数,第三浮点数可以包括第一符号域、Dot域、第一阶码域和第一尾数域。

处理器可以将计算结果的符号域的值作为第三浮点数的第一符号域的值,并根据计算结果的阶码和尾数,确定计算结果所属的数值范围。

- 10 如果计算结果所属的数值范围为第一设定范围,则可以根据计算结果的阶码域和尾数域确定第三浮点数的第一尾数域的值,并将Dot域设为第一值;第一值用于指示第一尾数域表征的是第一浮点数的阶码的偏移量。其中,第一设定范围指DML数据的范围。

- 如果计算结果所属的数值范围为第二设定范围,则可以根据计算结果的阶码域和尾数域,分别确定第三浮点数的第一阶码域、第一尾数域和Dot域的值。其中,第二设定范围指正常数据的范围。若根据计算结果的阶码域和尾数域,确定第一阶码域的值是第一预设阶码,即第一阶码域的位宽D为0,则可以
- 15 可以将Dot域设为第二值,第二值用于指示第一尾数域表征的是第三浮点数的尾数。

- 在一些实施例,处理器中还可以包括编码器。处理器中的计算单元得到计算结果之后,可以将计算结果传输至编码器,通过编码器将计算结果转换为第三浮点数。编码器也可以称为HiF8_DML编码器,示例性地,编码器的工作原理示意图可以如图7所示。计算结果可以包括符号域、阶码域和尾数域。编码器可以将计算结果的符号域的值,作为第三浮点数的第一符号域的值。编码器可以根据计算结果的
- 20 阶码域的值,确定计算结果所属的数值范围。

- 如果计算结果所属的数值范围为第一设定范围,即计算结果为DML数据,则编码器可以根据计算结果的阶码和尾数确定第三浮点数的第一尾数域的值。示例性地,编码器可以对计算结果的尾数进行处理,并根据处理后的尾数对计算结果的阶码进行调整,基于调整后的阶码得到第三浮点数的第一尾数域。例如,假设计算结果的尾数为1.XXXXXX,编码器可以对计算结果的尾数进行尾数移动和舍入操作,
- 25 舍入为一个整数位,得到处理后的尾数1.0,此时会导致阶码的值加1,因此可以调整计算结果的阶码,将调整后的阶码的值加上23,得到第三浮点数的第一尾数域的值,按照第一尾数域的值对第三浮点数的第一尾数域进行编码。编码器可以将Dot域设置为第一值,第一值用于指示第一尾数域表征的是第三浮点数的阶码的偏移量。

- 如果计算结果所属的数值范围为第二设定范围,即计算结果为正常数据,则编码器可以根据计算结果的阶码和尾数,分别确定第三浮点数的第一阶码域和第一尾数域,并基于第一阶码域确定第三浮点数的位宽指示域。示例性地,编码器可以对计算结果的尾数进行处理,并根据处理后的尾数对计算结果的阶码进行调整,基于处理后的尾数得到第三浮点数的第一尾数域的值,并基于调整后的阶码得到第三浮点数的第一阶码域的值,基于第一阶码域的值,确定第一阶码域所占用的位宽D,基于第一阶码域所占用的位宽D,确定位宽指示域的值。当第一阶码域所占用的位宽D为0时,将位宽指示域设为第二值;
- 35 第二值用于指示第一尾数域表征的是第一浮点数的尾数。例如,编码器可以通过对计算结果中的阶码的绝对值进行前找1操作,以及对计算结果中的尾数移位和舍入等操作,分别实现对第三浮点数的Dot域、第一阶码域和第一尾数域进行编码。

- 例如,假设计算结果的符号位为1,阶码为“00011”(即3),尾数为“1111”。编码器可以确定第三浮点数的符号位为“1”。编码器通过对阶码的绝对值“00011”进行前找1操作,直至找到最前面的1,可以确定第三浮点数的第一阶码域的位宽为2bits,第一阶码域的编码值为“01”,其中,“0”为阶码符号位,表示阶码为正,而阶码幅值“11”中的最高位“1”可隐藏,不占位宽,上文中已经描述,在此不再赘述。编码器根据第一阶码域的位宽2bits,可以确定第三浮点数的Dot域表示的数值为2。仍以表4中所示的编码规则为例,Dot域表示的数值为2,可以确定Dot域对应的编码值为“01”,占用2bits的位宽,则剩余可编码的第一尾数域的位宽为3bits。而输入的计算结果的尾数“1111”的位宽大于第三浮点数的第一尾数域的位宽,因此,可以对尾数“1111”进行舍入操作,得到包括小数点左侧隐藏位的尾数“10.000”,此时尾数“10.000”中的小数点要再向左移一位,得到包括小数点左侧隐藏位的尾数“1.0000”。由于小数点向左移一位,相应地,阶码需再加1(即3再加1,为4),即第一阶码域的幅值为4,对应的编码值为“100”,其中,“100”中的最高位“1”隐藏,第一阶码域最终的编码值为“00”,第一尾数域最终的编码值为“000”。
- 40
- 45

最终，编码器编码得到的第三浮点数为“10100000”。

最终得到的第三浮点数为本申请实施例提供的 Hif8_DML 的数据，第三浮点数可以用于数据存储和数据转移，以减少数据存储或数据转移所需的资源。

与上述实施例基于相同的发明构思，本申请实施例还提供一种浮点数处理方法，用于得到上述过程中使用的的第一浮点数。如图 8 所示，该方法可以包括如下步骤：

S801，获取符号、阶码和尾数。

S802，基于符号、阶码和尾数，得到第一浮点数。

例如，处理器可以获取待编码数据，待编码数据可以是与第一浮点数的格式不同的其他格式的数据，也可以是计算单元输出的操作结果，待编码数据可以包括符号、阶码和尾数。处理器从待编码数据中提取符号、阶码和尾数，基于该符号、阶码和尾数得到第一浮点数。

处理器基于该符号、阶码和尾数得到第一浮点数的过程，可以参照上文中基于计算结果得到第三浮点数的过程执行，在此不再赘述。

本申请实施例提供的浮点数，在标准的符号域、阶码域和尾数域的基础上，额外增加一个 Dot 域，通过 Dot 域指示浮点数的第一阶码域的有效位宽，该位宽为浮点数的阶码在实际存储时所占用的位宽 D，从而使得浮点数中第一阶码域的位宽可以随着 Dot 域的数值动态变化，相应的，浮点数中第一尾数域的位宽也随之动态变化，从而使浮点数具有锥型精度特征，使得阶码中心的数据，拥有较高的尾数位宽，即拥有较高的精度；离阶码中心越远的数据，尾数位宽逐渐减少，精度降低。浮点数的阶码可以决定浮点数的数值范围，因此，本申请实施例可以有效平衡浮点数的总位宽，数值范围和数值精度，实现在不额外增加总位宽，不额外增加数据存储或数据转移成本的前提下，满足各类场景下对浮点数的数值范围和数值精度的不同需求，提升浮点数的使用效果。

示例性地，本申请实施例提供的浮点数表示方法，可以适用于高精度数据的有损压缩、混合精度矩阵求解器的前期近似解计算中。针对通用计算和高性能计算，本申请实施例可以在相同的总位宽，即相同的数据存储或数据转移的开销下，获得更高的计算任务收敛速度和精度。针对 AI 神经网络训练和推理，本申请实施例可以在相同的总位宽下，满足神经网络的训练和推理的功能和准确度需求。本申请实施例在第一阶码域所占用的位宽为 0 时，为尾数提供 3 比特的编码空间，3 比特的尾数精度已经可以满足 AI 神经网络训练和推理的需要，因此，本申请实施例将 Dot 域的位宽设置为 4 比特，其中的 1 个比特用于指示第一尾数域表征的是尾数还是阶码的偏移量，如果第一尾数域表征的是阶码的偏移量，说明该浮点数为 DML 数据。当浮点数为 DML 数据时，尾数可以采用默认值 1，尾数精度可以满足 AI 神经网络训练和推理的需要，此时，借用第一尾数域来表征阶码的偏移量，可以扩大阶码的动态范围，扩大所能表示的浮点数的数值范围。

例如，上文中已经介绍，本申请实施例提供的浮点数表示方式具有 $[-22, 15]$ 的数值范围，远大于 FP8 的 $[-9, 8]$ 或 $[-16, 15]$ 的数值范围。本申请实施例提供的数值范围也可以满足 AI 神经网络训练和推理的需要。比如，针对大语言模型（large language model, LLM）的端到端的训练，如图 9 所示，采用传统的浮点数表示方式 FP8，在对 LLM 进行端到端训练时，由于浮点数能表示的数据的数值范围较小，当出现大量数值很小的数据时，这些数据不能区分，被认为是同一值，从而导致无法得到正确的训练结果，表现为在训练至一百多次时，损失值突然增大，LLM 训练，无法收敛。而采用本申请实施例提供的浮点数表示方式 Hif8_DML，对 LLM 进行端到端训练时，如图 10 所示，由于浮点数能表示的数据的数值范围较大，可以满足 LLM 训练的需要，因此，训练准确度提高，LLM 可以正常收敛，从而可以提升 LLM 训练的性能和稳定性。

综上，本申请实施例既可以保证对 AI 神经网络训练和推理的数值精度影响较小，同时又可以腾出编码空间来表达 DML 数据，从而很好平衡数值精度和数值范围。

此外，本申请实施例还限定了第一阶码域的阶码在不同位宽下可编码的数值范围，以及第一尾数域在表征阶码的偏移量时可编码的数值范围，有效避免了在不同位宽下，阶码的数值产生重叠的问题，使本申请实施例的数据编码方式无信息重复，是无冗余编码。基于此，第一阶码域中原码幅值的最高位可以隐藏不存储，能够进一步降低浮点数的数据存储或数据转移成本。

与上述方法实施例基于相同的设计构思，本申请实施例还提供一种浮点数处理装置。该浮点数处理装置可以应用于图 1 或图 2 所示的计算设备中，或者，应用于图 2 所示的解码器 111 中，该浮点数处理装置可以用于实现上述方法实施例的功能，因此可以实现上述方法实施例所具备的有益效果。如图 11

所示,该浮点数处理装置 1100 可以包括浮点数获取模块 1101 和解码模块 1102。

其中,浮点数获取模块 1101,可以用于获取第一浮点数;第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;第一符号域用于指示第一浮点数的符号,位宽指示域用于指示第一阶码域在第一浮点数的总位宽 N 中所占用的位宽 D;位宽指示域还用于指示:第一尾数域表征的是第一浮点数的尾数,或,第一浮点数的阶码的偏移量;

解码模块 1102,可以用于对第一浮点数进行解码,以得到符号、阶码和尾数。

在一些实施例中,解码模块 1102,具体可以用于:对第一浮点数进行解码,以得到第二浮点数;第二浮点数包括第二符号域、第二阶码域和第二尾数域,第二符号域用于指示符号,第二阶码域用于指示阶码,第二尾数域用于指示尾数。

在一些实施例中,浮点数处理装置 1100 还可以包括计算模块,计算模块可以用于:采用符号、阶码和尾数进行计算。在另一些实施例中,计算模块也可以位于解码器之外的计算单元中,浮点数处理装置 1100 可以将符号、阶码和尾数传输至计算单元,由计算单元采用符号、阶码和尾数进行计算。

需要说明的是,在一些实施例中,浮点数获取模块 1101 可以用于执行浮点数据处理方法中的任意步骤,解码模块 1102 可以用于执行浮点数据处理方法中的任意步骤。浮点数获取模块 1101 和解码模块 1102 负责实现的步骤可根据需要指定,浮点数获取模块 1101 和解码模块 1102 分别实现浮点数据处理方法中不同的步骤来实现浮点数处理装置的全部功能。

在本申请的实施例中的各功能模块可以集成在一个处理器中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

与上述方法实施例基于相同的设计构思,本申请实施例还提供一种浮点数处理装置。该浮点数处理装置可以应用于图 1 或图 2 所示的计算设备中,或者,应用于图 2 所示的编码器 112 中,该浮点数处理装置可以用于实现上述方法实施例的功能,因此可以实现上述方法实施例所具备的有益效果。如图 12 所示,该浮点数处理装置 1200 可以包括数据获取模块 1201 和编码模块 1202。

其中,数据获取模块 1201,可以用于获取符号、阶码和尾数;编码模块 1202,可以用于基于符号、阶码和尾数,得到第一浮点数。第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;第一符号域用于指示第一浮点数的符号,位宽指示域用于指示第一阶码域在第一浮点数的总位宽 N 中所占用的位宽 D;位宽指示域还用于指示:第一尾数域表征的是第一浮点数的尾数,或,第一浮点数的阶码的偏移量。

需要说明的是,在一些实施例中,数据获取模块 1201 可以用于执行浮点数据处理方法中的任意步骤,编码模块 1202 可以用于执行浮点数据处理方法中的任意步骤。数据获取模块 1201 和编码模块 1202 负责实现的步骤可根据需要指定,数据获取模块 1201 和编码模块 1202 分别实现浮点数据处理方法中不同的步骤来实现浮点数处理装置的全部功能。

在本申请的实施例中的各功能模块可以集成在一个处理器中,也可以是各个模块单独物理存在,也可以两个或两个以上模块集成在一个模块中。上述集成的模块既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

与上述方法实施例基于相同的技术构思,本申请实施例中还提供一种计算设备。该计算设备可以是图 1 或图 2 中所示的任意一个计算设备。该计算设备可以用于实现上述图 4、图 6 或图 8 所示的方法实施例的功能,因此可以实现上述方法实施例所具备的有益效果。

在一些实施例中,该计算设备 1300 的结构可以如图 13 所示,包括处理器 1301 以及与处理器 1301 连接的存储器 1302。处理器 1301 和存储器 1302 之间可以通过总线相互连接,处理器 1301 可以是通用处理器,如微处理器,或其他常规的处理器。总线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。

其中,存储器 1302 可用于存储软件程序以及模块,处理器 1301 通过运行存储在存储器 1302 中的软件程序以及模块,从而执行该终端设备 1300 的各种功能应用以及数据处理,如本申请实施例提供的任何一种浮点数处理方法。

存储器 1302 可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个应用的应用程序等;存储数据区可用于存储用户数据等。此外,存储器 1302 可以包括高速随机存取存

存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

计算设备 1300 中的处理器 1301 用于运行存储器 1302 中保存的计算机指令或者程序,执行上述任意一种方法实施例中的功能。在一些实施例中,处理器 1301 可以包括一个或多个处理单元,不同的处理单元可以是独立的器件,也可以集成在一个或多个处理器中。处理器 1301 中还可以包括控制器,控制器可以根据指令操作码和时序信号,产生操作控制信号,完成取指令和执行指令的控制。

在一些实施例中,计算设备 1300 的处理器 1301 中可以包括解码器,处理器 1301 可以用于实现图 4 所示的浮点数处理方法;在另一些实施例中,计算设备 1300 的处理器 1301 中可以包括编码器,处理器 1301 可以用于实现图 8 所示的浮点数处理方法;或者,在另一些实施例中,计算设备 1300 的处理器 1301 中可以包括解码器和编码器,处理器 1301 可以用于实现图 6 所示的浮点数处理方法。

在一种实施例中,计算设备 1300 还可以包括通信模块,通信模块可以用于与网络设备进行通信。

可以理解的是,本申请实施例示意的结构并不构成对终端设备的具体限定。在本申请另一些实施例中,终端设备可以包括比图示更多或更少的部件,或者组合某些部件,或者拆分某些部件,或者不同的部件布置。图示的部件可以以硬件,软件或软件和硬件的组合实现。

本申请实施例还提供一种计算机程序产品,包含有计算机可执行指令。在一种实施例中,该计算机可执行指令用于使计算机执行上述方法实施例中的功能。

计算机可执行指令可以被存放于计算机可读存储介质中,本申请实施例还提供一种计算机可读存储介质,所述计算机可读存储介质内存储有可执行指令。在一种实施例中,该计算机可执行指令用于使计算机执行上述方法实施例中的功能。

本申请实施例提供的计算机可读存储介质可以是随机存取存储器(random access memory, RAM)、闪存、只读存储器(read-only memory, ROM)、可编程只读存储器(programmable ROM, PROM)、可擦除可编程只读存储器(erasable PROM, EPROM)、电可擦除可编程只读存储器(electrically EPROM, EEPROM)、寄存器、硬盘、移动硬盘、CD-ROM 或者本领域熟知的任何其它形式的计算机可读存储介质。

计算机可执行指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,所述计算机程序或指令可以从一个网站站点、计算机、服务器或数据中心通过有线或无线方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是集成一个或多个可用介质的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,例如,软盘、硬盘、磁带;也可以是光介质,例如,数字视频光盘(digital video disc, DVD);还可以是半导体介质,例如,固态硬盘。

在本申请的各个实施例中,如果没有特殊说明以及逻辑冲突,不同的实施例之间的术语和/或描述具有一致性、且可以相互引用,不同的实施例中的技术特征根据其内在的逻辑关系可以组合形成新的实施例。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元。方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

尽管结合具体特征及其实施例对本申请进行了描述,显而易见的,在不脱离本申请的精神和范围的情况下,可对其进行各种修改和组合。相应地,本说明书和附图仅仅是所附权利要求所界定的方案进行示例性说明,且视为已覆盖本申请范围内的任意和所有修改、变化、组合或等同物。

显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的范围。这样,倘若本申请实施例的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

权利要求

- 1.一种浮点数处理方法,其特征在于,所述方法包括:
获取第一浮点数,所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D;所述位宽指示域还用于指示:所述第一尾数域表征的是所述第一浮点数的尾数,或,所述第一浮点数的阶码的偏移量;
5 对所述第一浮点数进行解码,以得到所述符号、所述阶码和所述尾数。
- 2.根据权利要求 1 所述的方法,其特征在于,所述对所述第一浮点数进行解码,以得到所述符号、
10 所述阶码和所述尾数,包括:
对所述第一浮点数进行解码,以得到第二浮点数;所述第二浮点数包括第二符号域、第二阶码域和第二尾数域,所述第二符号域用于指示所述符号,所述第二阶码域用于指示所述阶码,所述第二尾数域用于指示所述尾数。
- 15 3.根据权利要求 1 或 2 所述的方法,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存在,若所述第一尾数域表征的是所述尾数,则所述阶码为第一预设阶码。
- 4.根据权利要求 1~3 中任一项所述的方法,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存在,若所述第一尾数域表征的是所述偏移量,则所述尾数是预设尾数,所述阶码为利用所述偏移量修正第二预设阶码得到的值。
20
- 5.根据权利要求 1~4 中任一项所述的方法,其特征在于,当所述位宽 D 不为 0 时,所述第一尾数域表征的是所述尾数,所述阶码为所述第一阶码域指示的值。
- 25 6.根据权利要求 1~5 中任一项所述的方法,其特征在于,所述位宽指示域的位宽 DW 与所述位宽 D 负相关。
- 7.根据权利要求 1~6 中任一项所述的方法,其特征在于,当所述位宽指示域的位宽 DW 为预设位宽,且所述位宽指示域指示的值为预设值时,所述第一尾数域表征的是所述偏移量。
30
- 8.根据权利要求 1~7 中任一项所述的方法,其特征在于,所述获取第一浮点数,包括:
从存储器读取所述第一浮点数,或者,通过通信网络获取所述第一浮点数。
- 9.根据权利要求 1~8 中任一项所述的方法,其特征在于,所述方法还包括:
35 采用所述符号、所述阶码和所述尾数进行计算。
- 10.一种浮点数处理方法,其特征在于,所述方法包括:
获取符号、阶码和尾数;
基于所述符号、阶码和尾数,得到第一浮点数;所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述
40 所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D;所述位宽指示域还用于指示:所述第一尾数域表征的是所述第一浮点数的尾数,或,所述第一浮点数的阶码的偏移量。
- 11.根据权利要求 10 所述的方法,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存在,若
45 所述第一尾数域表征的是所述尾数,则所述阶码为第一预设阶码。
- 12.根据权利要求 10 或 11 所述的方法,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存

在,若所述第一尾数域表征的是所述偏移量,则所述尾数是预设尾数,所述阶码为利用所述偏移量修正第二预设阶码得到的值。

13.根据权利要求 10~12 中任一项所述的方法,其特征在于,当所述位宽 D 不为 0 时,所述第一尾数域表征的是所述尾数,所述阶码为所述第一阶码域指示的值。

14.根据权利要求 10~13 中任一项所述的方法,其特征在于,所述位宽指示域的位宽 DW 与所述位宽 D 负相关。

15.根据权利要求 10~14 中任一项所述的方法,其特征在于,当所述位宽指示域的位宽 DW 为预设位宽,且所述位宽指示域指示的值为预设值时,所述第一尾数域表征的是所述偏移量。

16.一种浮点数处理装置,其特征在于,所述装置包括:
浮点数获取模块,用于获取第一浮点数,所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D;所述位宽指示域还用于指示:所述第一尾数域表征的是所述第一浮点数的尾数,或,所述第一浮点数的阶码的偏移量;
解码模块,用于对所述第一浮点数进行解码,以得到所述符号、所述阶码和所述尾数。

17.根据权利要求 16 所述的装置,其特征在于,所述解码模块,具体用于:
对所述第一浮点数进行解码,以得到第二浮点数;所述第二浮点数包括第二符号域、第二阶码域和第二尾数域,所述第二符号域用于指示所述符号,所述第二阶码域用于指示所述阶码,所述第二尾数域用于指示所述尾数。

18.一种浮点数处理装置,其特征在于,所述装置包括:
数据获取模块,用于获取符号、阶码和尾数;
编码模块,用于基于所述符号、阶码和尾数,得到第一浮点数;所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D;所述位宽指示域还用于指示:所述第一尾数域表征的是所述第一浮点数的尾数,或,所述第一浮点数的阶码的偏移量。

19.根据权利要求 18 所述的装置,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存在,若所述第一尾数域表征的是所述偏移量,则所述尾数是预设尾数,所述阶码为利用所述偏移量修正第二预设阶码得到的值。

20.一种计算设备,其特征在于,包括处理器和存储器;
所述存储器上存储有计算机程序;
所述处理器用于读取所述存储器中存储的所述计算机程序并执行如下步骤:
获取第一浮点数,所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D;所述位宽指示域还用于指示:所述第一尾数域表征的是所述第一浮点数的尾数,或,所述第一浮点数的阶码的偏移量;
对所述第一浮点数进行解码,以得到所述符号、所述阶码和所述尾数。

21.根据权利要求 20 所述的计算设备,其特征在于,所述处理器,具体用于:
对所述第一浮点数进行解码,以得到第二浮点数;所述第二浮点数包括第二符号域、第二阶码域和第二尾数域,所述第二符号域用于指示所述符号,所述第二阶码域用于指示所述阶码,所述第二尾数域用于指示所述尾数。

22.一种计算设备,其特征在于,包括处理器和存储器;
所述存储器上存储有计算机程序;
所述处理器用于读取所述存储器中存储的所述计算机程序并执行如下步骤:
获取符号、阶码和尾数;

5 基于所述符号、阶码和尾数,得到第一浮点数;所述第一浮点数包括第一符号域、位宽指示域、第一阶码域和第一尾数域;所述第一符号域用于指示所述第一浮点数的符号,所述位宽指示域用于指示所述
10 所述第一阶码域在所述第一浮点数的总位宽 N 中所占用的位宽 D; 所述位宽指示域还用于指示: 所述第
15 一尾数域表征的是所述第一浮点数的尾数, 或, 所述第一浮点数的阶码的偏移量。

23.根据权利要求 22 所述的计算设备,其特征在于,当所述位宽 D 为 0 时,所述第一阶码域不存在,
若所述第一尾数域表征的是所述偏移量,则所述尾数是预设尾数,所述阶码为利用所述偏移量修正第二
预设阶码得到的值。

24.一种计算机可读存储介质,其特征在于,存储有计算机可执行指令,所述计算机可执行指令用
15 于使计算机执行如权利要求 1~9 任一项所述的方法; 或者, 如权利要求 10~15 任一项所述的方法。

25.一种计算机程序产品,其特征在于,包含有计算机可执行指令,所述计算机可执行指令用于使
计算机执行如权利要求 1~9 任一项所述的方法; 或者, 如权利要求 10~15 任一项所述的方法。