

浮点数乘法计算的方法、装置和算术逻辑单元

本申请要求于 2019 年 12 月 31 日提交的申请号为 201911414534.8、发明名称为“浮点数乘法计算的方法、装置和算术逻辑单元”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

本申请涉及数据处理技术领域，特别涉及一种浮点数乘法计算的方法、装置和算术逻辑单元。

背景技术

浮点数是计算机中一种重要的数字格式，在计算机中浮点数由符号、阶码和尾数三部分组成。为了满足不同业务对数据精度的不同需求，在计算机的处理器中通常需要具有对不同精度浮点数的乘法运算能力。

目前，为了实现对不同精度浮点数的乘法需求，通常会对应精度需求设计多个独立的乘法器。例如，需要处理器同时支持半精度、单精度和双精度的乘法运算，则需要在处理器中独立的至少三个乘法器，使其分别满足半精度、单精度和双精度乘法。

在实现本申请的过程中，发明人发现相关技术至少存在以下问题：

在处理器中独立设计多个分别支持不同精度的乘法器，当系统只使用其中一种精度的乘法器计算时，其余精度的乘法器处于闲置状态，十分浪费计算资源。

发明内容

为了解决相关技术中计算资源浪费的问题，本申请实施例提供了一种浮点数乘法计算的方法、装置和算术逻辑单元。所述技术方案如下：

第一方面，提供了一种浮点数乘法计算的方法，该方法包括：

获取多个待计算的第一精度浮点数；

对每个待计算的第一精度浮点数进行分解，得到至少两个第二精度浮点数，其中，所述第二精度浮点数的精度低于所述第一精度浮点数的精度；

确定各种由分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合；

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的中间计算结果；

基于每种组合对应的中间计算结果，确定所述多个待计算的第一精度浮点数的计算结果。

本申请实施例所示的方案中，浮点数由符号、阶码和尾数三部分组成，而整数为 1，在浮点数的表示中可以省略。对于待计算的第一精度浮点数，可以先进行分解。此处，对于第一精度浮点数进行分解，即，将第一精度浮点数的整数和尾数由多个第二精度浮点数的和来表示。这里，第二精度浮点数的精度要低于第一精度的精度。例如，第一精度浮点数为单精度浮点数，则第二精度浮点数可以为半精度浮点数。又例如，第一精度浮点数为双精度浮点

数，则第二精度浮点数可以为单精度浮点数或者半精度浮点数。对于分解后的第二精度浮点数，可以将分解自不同的第一精度浮点数的两个第二精度浮点数进行组合，然后，对于得到的每个组合输入到用于计算第二精度浮点数乘法的第二精度乘法器中。则，第二精度乘法器可以输出每组第二精度浮点数对应的中间计算结果。最后，对多个中间计算结果进行处理，即可得到多个待计算的第一精度浮点数的计算结果。

可见，在本申请实施例所示的方案中，计算第一精度浮点数的乘法的过程中，不必使用精度较高的第一精度乘法器，而只需要使用精度相对较低的第二精度乘法器即可。也即是，在仅部署有第二精度乘法器的处理单元中，除了可以正常计算第二精度浮点数乘法外，还可以计算精度更高的第一精度浮点数的乘法，这样可以有效的利用计算资源，可以节省单独部署第一精度乘法器的成本。

在一种可能的实现方式中，所述对每个待计算的第一精度浮点数进行分解，得到至少两个第二精度浮点数，之后还包括：

确定每个第二精度浮点数对应的阶码位移值；

所述基于每种组合对应的中间计算结果，确定所述多个待计算的第一精度浮点数的计算结果，包括：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果；

将每种组合对应的调整后的中间计算结果进行求和运算，得到所述多个第一精度浮点数的计算结果。

本申请实施例所示的方案中，在对于第一精度浮点数进行分解时，要使用第二精度浮点数来分别表示出该第一精度浮点数的不同位数的尾数。同时还可以得到每个第二精度浮点数对应的阶码位移值，该阶码位移值可以包括第一精度浮点数的阶码，还可以包括有该第二精度浮点数所表示的尾数在第一精度浮点数中的固定阶码位移值。下面对于固定阶码位移值进行说明。

例如，对于一个单精度浮点数 0 10000000 00000010100000100110011，符号为“0”，阶码为“10000000”，尾数为“00000010100000100110011”，如果要使用半精度浮点数表示尾数中的第 11bit 到第 21bit，“00001001100”，实际要表示的数为“0.000000000000 0001001100”，那么，可以提取出固定阶码位移值-11。

相应的，对应每种组合对应的中间计算结果的阶码，需要使用第二精度浮点数对应的阶码位移值进行调整。最后，对于调整后的中间计算结果可以输入累加器，进行累加计算得到最后的计算结果。

在一种可能的实现方式中，所述分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果，包括：

将每种组合的第二精度浮点数对应的中间计算结果的阶码，加上每种组合的第二精度浮点数对应的阶码位移值，得到调整后的中间计算结果。

本申请实施例所示的方案中，每个第二精度浮点数对应有阶码位移值，在得到通过第二精度乘法器得到中间结果后，对于每种组合对应的中间计算结果，需要使用未参加计算的每个第二精度浮点数的阶码位移值，与该中间计算结果的阶码相加，得到调整后的中间计算结果。

在一种可能的实现方式中，所述中间计算结果为第一精度中间计算结果，所述计算结果为第一精度计算结果。

本申请实施例所示的方案中，对于第一精度浮点数的乘法计算，仍然可以得到第一精度的计算结果，即精度不会降低。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第一精度计算结果为单精度计算结果，所述第二精度乘法器为半精度乘法器；或者，

所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为单精度浮点数，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为单精度乘法器。

本申请实施例所示的方案中，可以实现待计算的单精度浮点数，通过半精度乘法器，最终得到单精度计算结果，无需使用单精度乘法器，即可实现单精度乘法计算，节省计算资源。也可以实现待计算的双精度浮点数，通过单精度乘法器，最终得到双精度计算结果，无需使用双精度乘法器，即可实现双精度乘法计算，节省计算资源。

在一种可能的实现方式中，所述将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的中间计算结果，包括：

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的第一精度中间计算结果，对每个第一精度中间计算结果进行格式转换得到每种组合对应的第三精度中间计算结果，其中，所述第三精度中间计算结果的精度高于所述第一精度中间计算结果；

所述分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果，包括：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的第三精度中间计算结果的阶码，得到调整后的第三精度中间计算结果；

所述将每种组合对应的调整后的中间计算结果进行求和运算，得到所述多个第一精度浮点数的计算结果，包括：

将每种组合对应的调整后的第三精度中间计算结果进行求和运算，得到所述多个第一精度浮点数的第三精度计算结果。

本申请实施例所示的方案中，对于第一精度浮点数的乘法计算，可以得到更高精度的第三精度计算结果。

在一种可能的实现方式中，所述对每个第一精度中间结果进行格式转换得到每种组合第二精度浮点数对应的第三精度中间计算结果，包括：

对每个第一精度中间结果的阶码和尾数分别进行补零处理，得到每种组合对应的第三精度中间计算结果。

本申请实施例所示的方案中，由于第二精度浮点数经过第二精度乘法器只能得到第一精度中间计算结果，那么，如果要最后得到更高精度的第三精度计算结果的话，需要对第一精度中间计算结果的格式进行扩展，扩展为第三精度中间计算结果，扩展方法可以为对阶码和尾数在最后一位之后补零。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第三精度中间计

算结果为双精度中间计算结果，所述第三精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

本申请实施例所示的方案中，可以实现待计算的单精度浮点数，通过半精度乘法器，最终得到双精度计算结果，无需单精度乘法器，即可实现单精度乘法计算，节省计算资源。

在一种可能的实现方式中，所述将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的中间计算结果，包括：

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的第三精度中间计算结果，对每个第三精度中间计算结果进行格式转换得到每种组合对应的第一精度中间计算结果；

所述分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果，包括：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的第一精度中间计算结果的阶码，得到调整后的第一精度中间计算结果；

所述将各组第二精度浮点数对应的调整后的中间计算结果进行求和运算，得到所述多个第一精度浮点数的计算结果，包括：

将各组第二精度浮点数对应的调整后的第一精度中间计算结果进行求和运算，得到所述多个第一精度浮点数的第一精度计算结果。

本申请实施例所示的方案中，第二精度浮点数通过第二精度浮点数乘法器后，可能无法直接得到第一精度中间计算结果，而只能得到第三精度中间计算结果。那么，为了使最后的计算结果精度仍为第一精度，则可以对第三精度中间计算结果的格式进行扩展，得到第一精度中间结果，以使最后得到第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为半精度浮点数，所述第三精度中间计算结果为单精度中间计算结果，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

本申请实施例所示的方案中，可以实现待计算的双精度精度浮点数，通过半精度乘法器，得到单精度中间计算结果，经过对单精度中间计算结果进行格式调整后，最终得到双精度计算结果，无需使用双精度乘法器，即可得到实现双精度浮点数乘法计算，节省计算资源。

第二方面，提供了一种浮点数乘法计算的装置，该装置包括：

获取模块，用于获取多个待计算的第一精度浮点数；

分解模块，用于对每个待计算的第一精度浮点数进行分解，得到至少两个第二精度浮点数，其中，所述第二精度浮点数的精度低于所述第一精度浮点数的精度；

组合模块，用于确定各种由分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合；

输入模块，用于将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的中间计算结果；

确定模块，用于基于每种组合对应的中间计算结果，确定所述多个待计算的第一精度浮点数的计算结果。

在一种可能的实现方式中，所述分解模块，还用于：

确定每个第二精度浮点数对应的阶码位移值；

所述确定模块，用于：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果；

将每种组合对应的调整后的中间计算结果进行求和运算，得到所述多个第一精度浮点数的计算结果。

在一种可能的实现方式中，所述确定模块，用于：

将每种组合的第二精度浮点数对应的中间计算结果的阶码，加上每种组合的第二精度浮点数对应的阶码位移值，得到调整后的中间计算结果。

在一种可能的实现方式中，所述中间计算结果为第一精度中间计算结果，所述计算结果为第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第一精度计算结果为单精度计算结果，所述第二精度乘法器为半精度乘法器；或者，

所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为单精度浮点数，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为单精度乘法器。

在一种可能的实现方式中，所述输入模块，用于：

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的第一精度中间计算结果，对每个第一精度中间计算结果进行格式转换得到每种组合对应的第三精度中间计算结果，其中，所述第三精度中间计算结果的精度高于所述第一精度中间计算结果；

所述确定模块，用于：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的第三精度中间计算结果的阶码，得到调整后的第三精度中间计算结果；

将每种组合对应的调整后的第三精度中间计算结果进行求和运算，得到所述多个第一精度浮点数的第三精度计算结果。

在一种可能的实现方式中，所述输入模块，用于：

对每个第一精度中间结果的阶码和尾数分别进行补零处理，得到每种组合对应的第三精度中间计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第三精度中间计算结果为双精度中间计算结果，所述第三精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

在一种可能的实现方式中，所述输入模块，用于：

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的第三精度中间计算结果，对每个第三精度中间计算结果进行格式转换得到每种组合对应的第一精度中间计算结果；

所述确定模块，用于：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的第一精度中间计算结果的阶码，得到调整后的第一精度中间计算结果；

将各组第二精度浮点数对应的调整后的第一精度中间计算结果进行求和运算，得到所述多个第一精度浮点数的第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为半精度浮点数，所述第三精度中间计算结果为单精度中间计算结果，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

第三方面、提供了一种算术逻辑单元，所述算术逻辑单元包括浮点数分解电路、第二精度乘法器、阶码调整电路和累加器，其中：

所述浮点数分解电路，用于将输入的每个待计算的第一精度浮点数分解为至少两个第二精度浮点数，并向所述阶码调整电路输出每个第二精度浮点数对应的阶码位移值，其中，所述第二精度浮点数的精度低于所述第一精度浮点数的精度；

所述第二精度乘法器，用于接收分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合，将每种组合中的第二精度浮点数进行乘法运算，向所述阶码调整电路输出每种组合对应的中间计算结果；

所述阶码调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移值，调整输入的每种组合对应的中间计算结果的阶码，向所述累加器输出调整后的中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的中间计算结果进行求和运算，输出所述多个第一精度浮点数的计算结果。

在一种可能的实现方式中，所述阶码调整调整电路，用于将输入的每种组合中的第二精度浮点数对应的阶码位移值，与输入的每种组合对应的中间计算结果的阶码相加，向所述累加器输出调整后的中间计算结果。

在一种可能的实现方式中，所述中间计算结果为第一精度中间计算结果，所述计算结果为第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第一精度计算结果为单精度计算结果，所述第二精度乘法器为半精度乘法器；或者，

所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为单精度浮点数，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为单精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括格式转换电路；

所述第二精度乘法器，具体用于将每种组合中的第二精度浮点数进行乘法运算，向所述格式转换电路输出每种组合对应的第一精度中间计算结果；

所述格式转换电路，用于将输入的每个第一精度中间计算结果进行格式转换，向所述阶码调整电路输出每种组合对应的第三精度中间计算结果，其中，所述第三精度中间计算结果的精度高于所述第一精度中间计算结果；

所述阶码调整调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移

值，调整输入的每种组合对应的第三精度中间计算结果的阶码，向所述累加器输出调整后的第三精度中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的第三精度中间计算结果进行求和运算，输出所述多个第一精度浮点数的第三精度计算结果。

在一种可能的实现方式中，所述格式转换电路，用于：

将输入的每个第一精度中间计算结果的阶码和尾数分别进行补零处理，向所述阶码调整电路输出每种组合对应的第三精度中间计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第三精度中间计算结果为双精度中间计算结果，所述第三精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括格式转换电路；

所述第二精度乘法器，用于将每种组合中的第二精度浮点数进行乘法运算，向所述格式转换电路输出每种组合对应的第三精度中间计算结果；

所述格式转换电路，用于将输入的每个第三精度中间计算结果进行格式转换，向所述阶码调整电路输出每种组合对应的第一精度中间计算结果；

所述阶码调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移值，调整输入的每种组合对应的第一精度中间计算结果的阶码，向所述累加器输出调整后的第一精度中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的第一精度中间计算结果进行求和运算，输出所述多个第一精度浮点数的第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为半精度浮点数，所述第三精度中间计算结果为单精度中间计算结果，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括，计算模式切换电路，其中：

所述计算模式切换电路，用于在设置为第二精度浮点数计算模式时，将所述浮点数分解电路和所述阶码调整电路置为无效；

所述第二精度乘法器，用于接收所述算术逻辑单元外部的输入的多组待计算的所述第二精度浮点数，对每组第二精度浮点数进行乘法运算，输入每组待计算的所述第二精度浮点数对应的中间计算结果；

所述累加器，用于将输入的每组待计算的所述第二精度浮点数对应的中间计算结果进行求和运算，输出所述多组待计算的所述第二精度浮点数的计算结果。

第四方面、提供了一种电子设备，该电子设备包括处理器和存储器，所述存储器中存储有至少一条指令，所述指令由所述处理器加载并执行以实现如上述第一方面所述的浮点数乘法计算的方法所执行的操作。

第五方面、提供了一种处理器，所述处理器包括上述第三方面所述的算术逻辑单元。

第六方面、提供了一种计算机可读存储介质，该计算机可读存储介质中存储有至少一条指令，所述指令由处理器加载并执行以实现如上述第一方面所述的浮点数乘法计算的方法所执行的操作。

本申请实施例提供的技术方案带来的有益效果是：

在本申请实施例中，通过对每个待计算的第一精度浮点数进行分解，得到多个精度较低的第二精度浮点数。然后，将各种由分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合输入第二精度乘法器，得到每种组合对应的中间计算结果，最后，根据每种组合对应的中间计算结果，确定得到待计算的第一精度浮点数对应的计算结果。可见，在本申请实施例中，对于精度较高的多个第一精度浮点数的计算，可以由精度较低的第二精度乘法器实现，而不必再使用第一精度乘法器。因此，可以实现在只有较低精度的第二精度乘法器的设备中，实现较高精度的第一精度浮点数的计算，而不必额外设计第一精度乘法器，有效的节省了计算资源。

附图说明

- 图1是本申请实施例提供的一种浮点数乘法计算的方法流程图；
- 图2是本申请实施例提供的一种浮点数组成示意图；
- 图3是本申请实施例提供的一种浮点数组成示意图；
- 图4是本申请实施例提供的一种浮点数组成示意图；
- 图5是本申请实施例提供的一种第二精度浮点数输入第二精度乘法器的示意图；
- 图6是本申请实施例提供的一种第二精度浮点数输入第二精度乘法器的示意图；
- 图7是本申请实施例提供的一种浮点数乘法计算的装置的结构示意图；
- 图8是本申请实施例提供的一种电子设备的结构示意图；
- 图9是本申请实施例提供的一种浮点数乘法计算的方法流程图；
- 图10是本申请实施例提供的一种浮点数乘法计算的方法流程图；
- 图11是本申请实施例提供的一种浮点数乘法计算的方法流程图；
- 图12是本申请实施例提供的一种算术逻辑单元的结构示意图；
- 图13是本申请实施例提供的一种算术逻辑单元的结构示意图；
- 图14是本申请实施例提供的一种算术逻辑单元的结构示意图。

具体实施方式

本申请实施例提供了一种浮点数乘法计算的方法，该方法可以由电子设备实现，该电子设备可以为需要执行浮点数计算的任一设备。例如，该电子设备可以为手机、平板电脑等移动终端，也可以为台式机、笔记本电脑等计算机设备，还可以为服务器。而对于浮点数计算可以涉及到图形处理、天文、医学等诸多领域。在各领域中，使用上述类电子设备执行浮点数计算时，均可以采用本申请实施例所提供的方法，通过对高精度浮点数进行分解，得到低精度浮点数，再使用低精度乘法器对得到的低精度浮点数进行计算，最终得到高精度计算结果。可以在不损失精度的情况下，利用低精度乘法器完成相关技术中高精度乘法器才可以完

成的计算。

参见图 1，本申请实施例提供了一种浮点数乘法计算的方法，该方法的处理流程可以包括如下步骤：

步骤 101、获取多个待计算的第一精度浮点数。

其中，多个待计算的第一精度浮点数可以为一组需要进行乘法运算的第一精度浮点数。多个可以为两个，也可以为两个以上，在本申请实施例中以多个为两个的情况进行描述。

在实施中，计算机设备中的处理器可以获取到待计算的多个第一精度浮点数。该第一精度浮点数可以为单精度浮点数、双精度浮点数等。

步骤 102、对每个待计算的第一精度浮点数进行分解，得到至少两个第二精度浮点数，其中，第二精度浮点数的精度低于第一精度浮点数的精度。

在实施中，对于每个待计算的第一精度浮点数分解，可以得到多个第二精度浮点数，第二精度浮点数要小于第一精度浮点数的精度。对于第一精度浮点数和第二精度浮点数有多种可能情况，以下列举其中几种：第一精度浮点数可以为单精度浮点数（single-precision floating-point format, FP32），那么，第二精度浮点数即可以为半精度浮点数（half-precision floating-point format, FP16）。或者，第一精度浮点数可以为双精度浮点数（double-precision floating-point format, FP64），那么，第二精度浮点数即可以为 FP32，也可以为 FP16。下面对于以上几种进行分别说明。

情况一、对于第一精度浮点数为 FP32，第二精度浮点数为 FP16 的情况，对于 FP32 分解得到多个 FP16 可以有如下情况：

一、将一个 FP32 进行分解，得到三个 FP16。

目前，标准格式的 FP32 的组成如图 2 所示，包括 1bit（比特）符号，8bit 阶码（也可称为指数）和 23bit 尾数，另外，还有省略的 1bit 整数，省略的整数为 1。对于一个标准格式的 FP32，其整数加上尾数，共 24bit。标准格式的 FP16 的组成如图 3 所示，包括 1bit 符号，5bit 阶码和 10bit 尾数，另外，还有省略的 1bit 整数，省略的整数为 1。对于一个标准格式的 FP16 整数加上尾数，共 11bit。如果要将一个标准格式的 FP32 分解，得到标准格式的 FP16，需要 3 个标准格式的 FP16。

将标准格式的 FP32 的整数和尾数可以分为三部分，第一部分为整数和前 10bit 的尾数，第二部分为第 11bit 到第 21bit 的尾数，第三部分为第 22bit 和第 23bit 的尾数。将这三部分分别由一个标准格式的 FP16 表示。此处需要说明的是，对于上述第三部分第 22bit 和第 23bit 的尾数由标准格式的 FP16 表示时，可以先在第 23bit 的尾数后补 9 个 0，即将第 22bit 和第 23bit 的尾数，以及补的 0，由一个标准格式的 FP16 表示。

另外，FP16 的阶码范围为-15 到 15，即可以表示小数点左移 15 位到右移 15 位。使用标准格式的 FP16 表示上述 FP32 的第一部分时，固定阶码位移值为 0；使用标准格式的 FP16 表示上述 FP32 的第二部分时，固定阶码位移值为-11；使用标准格式的 FP16 表示上述 FP32 的第三部分时，固定阶码位移值为-22。可见，在表示第三部分时，仅对应的固定阶码位移值就已经超出了 FP16 的阶码范围。因此，可以对每个标准格式的 FP16 的阶码提取出对应的固定阶码位移值。

那么，对于一个标准格式的 FP32，可以表示为：

$A_1 = 2^{EA_1} (a_0 + 2^{-s_1} a_1 + 2^{-2s_1} a_2)$ ，其中， A_1 为标准格式的 FP32， EA_1 为 A_1 的阶码， a_0 、

a_1 和 a_2 为分解得到的三个标准格式的 FP16, S_1 为最小的固定阶码位移值, 对于该标准格式的 FP16 来说, $S_1=11$ 。

除此之外, 对于每个标准格式的 FP16 的阶码可以提取出一个公共阶码位移值。那么, 同样对于一个标准格式的 FP32, 可以表示为:

$A_1=2^{EA_1-S_1}(a_0'+a_1'+a_2')$, 其中, a_0' 、 a_1' 和 a_2' 为分解得到的三个标准格式的 FP16。在上述两种表示方法中, 分解得到的 FP16 有如下关系: $a_0=2^{-S_1}a_0'$, $a_1=a_1'$, $a_2=2^{S_1}a_2'$ 。

二、将一个 FP32 分解, 得到两个 FP16。

为了减少分解得到的 FP16 的个数, 对于目前标准格式的 FP16 可以进行调整, 将其尾数调整为 13bit, 符号和阶码的 bit 数不变, 可以称调整后的 FP16 为非标准格式的 FP16。则该非标准格式的 FP16 的整数加上尾数, 共 14bit。那么, 如果要将一个标准格式的 FP32 的尾数使用非标准格式的 FP16 表示, 则只需要 2 个非标准格式的 FP16。

将标准格式的 FP32 的整数和尾数分为两部分, 第一部分为整数和前 13bit 的尾数, 第二部分为第 14bit 到第 23bit。将这两部分分别由非标准格式的 FP16 表示。

此处还需说明的是, 对于上述第二部分由非标准 FP16 表示时, 可以先在第 23bit 的尾数后补 4 个 0, 即将第 14bit 到第 23bit 的尾数, 以及补的 0, 由一个非标准格式的 FP16 表示。与上述情况一相同, 此处, 也可以对每个标准格式的 FP16 的阶码提取出对应的固定阶码位移值。

那么, 同样对于一个标准格式的 FP32, 可以表示为:

$A_2=2^{EA_2}(a_3+2^{-S_2}a_4)$, 其中, A_2 为标准格式的 FP32, EA_2 为 A_2 的阶码, a_3 和 a_4 为分解得到的两个非标准格式的 FP16, S_2 为固定阶码位移值, 对于该非标准格式的 FP16 来说, $S_2=14$ 。

除此之外, 对于每个标准格式的 FP16 的阶码可以提取出一个公共阶码位移值。那么, 同样对于一个标准格式的 FP32, 可以表示为:

$A_2=2^{EA_2-S_2}(a_3'+a_4')$, 其中, a_3' 和 a_4' 为分解得到的两个非标准格式的 FP16。在上述两种表示方法中, 分解得到的 FP16 有如下关系: $a_3=2^{-S_2}a_3'$, $a_4=a_4'$ 。

情况二、对于第一精度浮点数为 FP64, 第二精度浮点数为 FP32 的情况, 对于 FP64 分解得到多个 FP32 可以有如下情况:

一、将一个 FP64 进行分解, 得到三个 FP32。

目前, 标准格式的 FP64 的组成如图 4 所示, 包括 1bit (比特) 符号, 11bit 阶码 (也可称为指数) 和 52bit 尾数, 另外, 还有省略的 1bit 整数, 省略的整数为 1。对于一个标准格式的 FP64, 其整数加上尾数, 共 53bit。如上所述标准格式的 FP32 的整数加上尾数, 共 24bit。如果要将一个标准格式的 FP64 分解, 得到标准格式的 FP32, 需要 3 个标准格式的 FP32。

将标准格式的 FP64 的整数和尾数可以分为三部分, 第一部分为整数和前 23bit 的尾数, 第二部分为第 24bit 到第 47bit 的尾数, 第三部分为第 48bit 到第 52bit 的尾数。将这三部分分别由一个标准格式的 FP32 表示。

此处还需要说明的是, 对于上述第三部分第 47bit 到第 52bit 的尾数由标准格式的 FP32 表示时, 可以先在第 23bit 的尾数后补 18 个 0, 即将第 47bit 到第 52bit 的尾数, 以及补的 0, 由一个标准格式的 FP32 表示。

那么, 对于一个标准格式的 FP64, 可以表示为:

$A_3 = 2^{EA_3} (a_5 + a_6 + a_7)$ ，其中， A_3 为标准格式的 FP64， EA_3 为 A_3 的阶码， a_5 、 a_6 和 a_7 为分解得到的三个标准格式的 FP32。

二、将一个 FP64 分解，得到两个 FP32。

为了减少分解得到的 FP32 的个数，对于目前标准格式的 FP32 可以进行调整，将其尾数调整为 26bit，符号和阶码的 bit 数不变，可以称调整后的 FP32 为非标准格式的 FP32。则该非标准格式的 FP32 的整数加上尾数，共 27bit。那么，如果要将一个标准格式的 FP64 的尾数使用非标准格式的 FP32 表示，则只需要 2 个非标准格式的 FP32。

将标准格式的 FP64 的整数和尾数分为两部分，第一部分为整数和前 26bit 的尾数，第二部分为第 27bit 到第 53bit。将这两部分分别由非标准格式的 FP32 表示。

那么，同样对于一个标准格式的 FP64，可以表示为：

$A_4 = 2^{EA_4} (a_8 + a_9)$ ，其中， A_4 为标准格式的 FP32， EA_4 为 A_4 的阶码， a_8 和 a_9 为分解得到的两个非标准格式的 FP32。

情况三、对于第一精度浮点数为 FP64，第二精度浮点数为 FP16 的情况，对于 FP64 分解得到多个 FP16 可以有如下情况：

一、将一个 FP64 进行分解，得到五个 FP16。

对于一个标准格式的 FP64，其整数加上尾数，共 53bit。如上所述标准格式的 FP32 的整数加上尾数，共 24bit。如果要将一个标准格式的 FP64 分解，得到标准格式的 FP16，需要 5 个标准格式的 FP16。

将标准格式的 FP64 的整数和尾数可以分为五部分，第一部分为整数和前 10bit 的尾数，第二部分为第 11bit 到第 21bit 的尾数，第三部分为第 22bit 到第 32bit 的尾数，第四部分为第 33bit 到第 43bit 的尾数，第五部分为第 44bit 到第 52bit 的尾数。将这五部分分别由一个标准格式的 FP64 表示。此处还需要说明的是，对于上述第五部分第 44bit 到第 52bit 的尾数由标准格式的 FP16 表示时，可以先在第 52bit 的尾数后补 2 个 0，即，将第 44bit 到第 52bit 的尾数，以及补的 0，由一个标准格式的 FP16 表示。

另外，由于 FP16 的阶码范围为 -15 到 15，即可以表示小数点左移 15 位到右移 15 位。使用标准格式的 FP16 表示上述 FP64 的第一部分时，固定阶码位移值为 0；使用标准格式的 FP16 表示上述 FP64 的第二部分时，固定阶码位移值为 -11；使用标准格式的 FP16 表示上述 FP64 的第三部分时，固定阶码位移值为 -22；使用标准格式的 FP16 表示上述 FP64 的第四部分时，固定阶码位移值为 -33；使用标准格式的 FP16 表示上述 FP64 的第五部分时，固定阶码位移值为 -44。可见，在表示第三部分、第四部分以及第五部分时，仅对应的固定阶码位移值就已经超出了 FP16 的阶码范围。因此，可以对每个标准格式的 FP16 的阶码提取出其对应的固定阶码位移值。

那么，对于一个标准格式的 FP64，可以表示为：

$A_5 = 2^{EA_5} (a_{10} + 2^{-S_1} a_{11} + 2^{-2S_1} a_{12} + 2^{-3S_1} a_{13} + 2^{-4S_1} a_{14})$ ，其中， A_5 为标准格式的 FP64， EA_5 为 A_5 的阶码， a_{10} 、 a_{11} 、 a_{12} 、 a_{13} 和 a_{14} 为分解得到的五个标准格式的 FP16， S_1 为最小的固定阶码位移值，对于该标准格式的 FP16 来说， $S_1 = 11$ 。

二、将一个 FP64 分解，得到 4 个 FP16。

同样的可以将 FP64 分解得到上述非标准格式的 FP16。对于一个标准格式的 FP64 的尾数使用非标准格式的 FP16 表示，则只需要 4 个非标准格式的 FP16。

将标准格式的 FP64 的整数和尾数分为四部分，第一部分为整数和前 13bit 的尾数，第二部分为第 14bit 到第 27bit，第三部分为第 28bit 到第 41bit，第四部分为第 42bit 到第 52bit。

此处还需要说明的是，对于上述第四部分第 42bit 到第 52bit 的尾数由非标准格式的 FP16 表示时，可以先在第 52bit 的尾数后补 3 个 0，即，将第 42bit 到第 52bit 的尾数，以及补的 0，由一个标准格式的 FP16 表示。另外，由于 FP16 的阶码范围为-15 到 15，即可以表示小数点左移 15 位到右移 15 位。使用非标准格式的 FP16 表示上述 FP64 的第一部分时，固定阶码位移值为 0；使用非标准格式的 FP16 表示上述 FP64 的第二部分时，固定阶码位移值为-14；使用非标准格式的 FP16 表示上述 FP64 的第三部分时，固定阶码位移值为-28；使用非标准格式的 FP16 表示上述 FP64 的第四部分时，固定阶码位移值为-42。可见，在表示第三部分和第四部分时，仅对应的固定阶码位移值就已经超出了 FP16 的阶码范围。因此，可以对每个非标准格式的 FP16 的阶码提取出其对应的固定阶码位移值。

那么，同样对于一个标准格式的 FP64，可以表示为：

$A_6 = 2^{EA_6} (a_{15} + 2^{-S_2} a_{16} + 2^{-2S_2} a_{17} + 2^{-3S_2} a_{18})$ ，其中， A_6 为标准格式的 FP64， EA_6 为 A_6 的阶码， a_{15} 、 a_{16} 、 a_{17} 和 a_{18} 为分解得到的四个非标准格式的 FP16， S_2 为最小的固定阶码位移值，对于该标准格式的 FP16 来说， $S_2 = -14$ 。

步骤 103、确定各种由分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合。

在实施中，对于分解自不同第一精度浮点数的第二精度浮点数两两组合。下面以两个 FP32 分别分解得到多个 FP16，两个 FP64 分别分解得到多个 FP32，两个 FP64 分别分解得到多个 FP16 为例进行说明。

情况一、两个 FP32，分别分解得到多个 FP16。

一、两个标准格式的 FP32，分别分解得到三个标准格式的 FP16。其中，两个 FP32 分别为 A_1 和 B_1 ， A_1 可以分解得到 a_0 、 a_1 和 a_2 ， B_1 可以分解为得到 b_0 、 b_1 和 b_2 。则对于 a_0 、 a_1 、 a_2 ，以及 b_0 、 b_1 、 b_2 之间，可以有如下组合： $a_0 b_0$ 、 $a_0 b_1$ 、 $a_1 b_0$ 、 $a_0 b_2$ 、 $a_1 b_1$ 、 $a_2 b_0$ 、 $a_1 b_2$ 、 $a_2 b_1$ 和 $a_2 b_2$ 。

二、两个标准格式的 FP32 分别分解得到两个上述非标准格式的 FP16。其中，两个 FP32 分别为 A_2 和 B_2 ， A_2 可以分解得到 a_3 和 a_4 ， B_2 可以分解得到 b_3 和 b_4 。则对于 a_3 、 a_4 ，以及 b_3 、 b_4 之间可以有如下组合： $a_3 b_3$ 、 $a_3 b_4$ 、 $a_4 b_3$ 和 $a_4 b_4$ 。

情况二、两个 FP64，分别分解得到多个 FP32。

一、两个标准格式的 FP64 分别分解得到三个标准格式的 FP32。其中，两个 FP64 分别为 A_3 和 B_3 ， A_3 可以分解得到 a_5 、 a_6 和 a_7 ， B_3 可以分解得到 b_5 、 b_6 和 b_7 。则对于 a_5 、 a_6 、 a_7 ，以及 b_5 、 b_6 、 b_7 之间，可以有如下组合： $a_5 b_5$ 、 $a_5 b_6$ 、 $a_6 b_5$ 、 $a_5 b_7$ 、 $a_6 b_6$ 、 $a_7 b_5$ 、 $a_6 b_7$ 、 $a_7 b_6$ 和 $a_7 b_7$ 。

二、两个标准格式的 FP64 分别分解得到两个上述非标准格式的 FP32。其中，两个 FP64 分别为 A_4 和 B_3 ， A_4 可以分解得到 a_8 和 a_9 ， B_4 可以分解得到 b_8 和 b_9 。则对于 a_8 、 a_9 ，以及 b_8 、 b_9 之间可以有如下组合： $a_8 b_8$ 、 $a_8 b_9$ 、 $a_9 b_8$ 和 $a_9 b_9$ 。

情况三、两个 FP64，分别分解得到多个 FP16。

一、两个标准格式的 FP64 分别分解得到五个标准格式的 FP16。其中，两个 FP64 分别为 A_5 和 B_5 ， A_5 可以分解得到 a_{10} 、 a_{11} 、 a_{12} 、 a_{13} 和 a_{14} ， B_5 可以分解得到 b_{10} 、 b_{11} 、 b_{12} 、 b_{13} 和 b_{14} 。则对于 a_{10} 、 a_{11} 、 a_{12} 、 a_{13} 、 a_{14} ，以及 b_{10} 、 b_{11} 、 b_{12} 、 b_{13} 、 b_{14} 之间可以有 $a_{10} b_{10}$ 、 $a_{10} b_{11}$ 、

$a_{11}b_{10}$ 、 $a_{10}b_{12}$ 、 $a_{11}b_{11}$ 、 $a_{12}b_{10}$ …… $a_{14}b_{14}$ 等 25 种组合。此处组合方式与上述相同，在此不再一一列举。

二、两个标准格式的 FP64 分别分解得到四个上述非标准格式的 FP16。其中，两个 FP64 分别为 A_6 和 B_6 ， A_6 可以分解得到 a_{15} 、 a_{16} 、 a_{17} 和 a_{18} ， B_6 可以分解得到 b_{15} 、 b_{16} 、 b_{17} 和 b_{18} 。则对于 a_{15} 、 a_{16} 、 a_{17} 、 a_{18} ，以及 b_{15} 、 b_{16} 、 b_{17} 、 b_{18} 之间可以有 $a_{15}b_{15}$ 、 $a_{15}b_{16}$ 、 $a_{16}b_{15}$ …… $a_{18}b_{18}$ 等 16 种组合。此处组合方式与上述相同，在此不再一一列举。

步骤 104、将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的中间计算结果。

在实施中，对于上述得到的每种组合，输入到第二精度乘法器中进行计算，得到每种组合对应的中间计算结果。对于第二精度浮点数的不同，输出的中间计算结果的精度也不同。例如，第一精度浮点数为 FP32，则中间计算结果为 FP64；第一精度浮点数为 FP16，则中间计算结果为 FP32。对于第二精度乘法器的个数，可以与第二精度浮点数的组合数相同，也可以不同。

第二精度乘法器与第二精度浮点数的组合数相同的情况下。如图 5 所示，两个第一精度浮点数 A 和 B，各自分解得到两个第二精度浮点数 A1、A0 和 B1、B0。A1、A0、B1、B0 可以得到四种组合，有四个第二精度乘法器，每种组合的第二精度浮点数输入一个第二精度乘法器，即每种组合对应一个第二精度浮点数。

第二精度乘法器与第二精度浮点数的组合数不同的情况下。如图 6 所示，两个第一精度浮点数 A 和 B，各自分解得到两个第二精度浮点数 A1、A0 和 B1、B0。A1、A0、B1、B0 可以得到四种组合，只有一个第二精度乘法器，则可以将四种组合的第二精度浮点数依次输入第二精度乘法器。

步骤 105、基于每种组合对应的中间计算结果，确定多个待计算的第一精度浮点数的计算结果。

在实施中，在对第一精度浮点数进行分解，得到第二精度浮点数时，还可以得到每个第二精度浮点数对应的阶码位移值。下面针对上述步骤 102 中分解待计算的第一精度浮点数的几种情况，对于第二精度浮点数对应的阶码位移值分别进行说明。

情况一、对于第一精度浮点数为 FP32，第二精度浮点数为 FP16 的情况。

一、将一个 FP32 进行分解，得到三个 FP16。

对于此种情况，FP32 可以表示为： $A_1 = 2^{EA_1} (a_0 + 2^{-S_1} a_1 + 2^{-2S_1} a_2)$ ，则 a_0 对应的阶码位移值为 EA_1 ， a_1 对应的阶码位移值为 $EA_1 - S_1$ ， a_2 对应的阶码位移值为 $EA_1 - 2S_1$ 。或者，FP32 还可以表示为： $A_1 = 2^{EA_1 - S_1} (a_0' + a_1' + a_2')$ ，则 a_0' 、 a_1' 和 a_2' 对应的阶码位移值均为 $EA_1 - S_1$ 。

二、将一个 FP32 进行分解，得到两个 FP16。

对于此种情况，FP32 可以表示为： $A_2 = 2^{EA_2} (a_3 + 2^{-S_2} a_4)$ ，则 a_3 对应的阶码位移值为 EA_2 ， a_4 对应的阶码位移值均为 $EA_2 - S_2$ 。或者，FP32 还可以表示为： $A_2 = 2^{EA_2 - S_2} (a_3' + a_4')$ ，则 a_3' 和 a_4' 对应的阶码位移值均为 $EA_2 - S_2$ 。

情况二、对于第一精度浮点数为 FP64，第二精度浮点数为 FP32 的情况。

一、将一个 FP64 进行分解，得到三个 FP32。

对于此种情况，FP64 可以表示为： $A_3 = 2^{EA_3} (a_5 + a_6 + a_7)$ ，则 a_5 、 a_6 和 a_7 对应的阶码位

移值均为 EA_3 。

二、将一个 FP64 分解，得到两个 FP32。

对于此种情况，FP64 可以表示为 $A_4 = 2^{EA_4} (a_8 + a_9)$ ，则 a_8 和 a_9 对应的阶码位移值均为 EA_4 。

情况三、对于第一精度浮点数为 FP64，第二精度浮点数为 FP16 的情况，对于 FP64 分解得到多个 FP16 可以有如下情况：

一、将一个 FP64 进行分解，得到五个 FP16。

对于此种情况，FP64 可以表示为：

$A_5 = 2^{EA_5} (a_{10} + 2^{-S_1} a_{11} + 2^{-2S_1} a_{12} + 2^{-3S_1} a_{13} + 2^{-4S_1} a_{14})$ ，则 a_{10} 对应的阶码位移值为 EA_5 ， a_{11} 对应的阶码位移值为 $EA_5 - S_1$ ， a_{12} 对应的阶码位移值为 $EA_5 - 2S_1$ ， a_{13} 对应的阶码位移值为 $EA_5 - 3S_1$ ， a_{14} 对应的阶码位移值为 $EA_5 - 4S_1$ 。

二、将一个 FP64 分解，得到 4 个 FP16。

对于此情况，FP64 可以表示为 $A_6 = 2^{EA_6} (a_{15} + 2^{-S_2} a_{16} + 2^{-2S_2} a_{17} + 2^{-3S_2} a_{18})$ ，则 a_{15} 对应的阶码位移值为 EA_6 ， a_{16} 对应的阶码位移值为 $EA_6 - S_2$ ， a_{17} 对应的阶码位移值为 $EA_6 - 2S_2$ ， a_{18} 对应的阶码位移值为 $EA_6 - 3S_2$ 。

相应的，对于每种组合对应的中间计算结果，可以分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的中间计算结果的阶码，得到调整后的中间计算结果。然后，对调整后的中间计算结果进行累加得到计算结果。此处累加时，可以将调整后的中间计算结果输入累加器，得到计算结果。

在调整中间计算结果的阶码时，可以将每种组合将每种组合的第二精度浮点数对应的中间计算结果的阶码，加上每种组合的第二精度浮点数对应的阶码位移值，得到调整后的中间计算结果。

在一种可能的实现方式中，可以对第二精度乘法器输出的第二精度中间计算结果的格式进行调整，以最后得到更高精度的计算结果。相应的处理可以如下：将每种组合第二精度浮点数输入第二精度乘法器中，得到每种组合第二精度浮点数对应的第一精度中间计算结果，对每个第一精度中间计算结果进行格式转换得到每种组合第二精度浮点数对应的第三精度中间计算结果，其中，第三精度中间计算结果的精度高于所述第一精度中间计算结果。分别基于每种组合第二精度浮点数对应的阶码位移值，调整每种组合第二精度浮点数对应的第三精度中间计算结果的阶码，得到调整后的第三精度中间计算结果。将各组第二精度浮点数对应的调整后的第三精度中间计算结果进行求和运算，得到多个第一精度浮点数的第三精度计算结果。

在对第一精度计算结果进行格式转换时，可以对每个第一精度中间结果的阶码和尾数分别进行补零处理，得到每种组合第二精度浮点数对应的第三精度中间计算结果。

例如，第一精度浮点数为 FP32，第二精度浮点数为 FP16，那么，第二精度乘法器输出的第一精度中间计算结果也为 FP32，可以对该第一精度中间计算结果的格式进行调整，调整为第三精度中间计算结果，该第三精度中间计算结果可以为 FP64。即，将第一精度中间计算结果的阶码由 8bit 在末位补 3 个 0，将阶码位数扩展为 11bit，与 FP64 的阶码位数相同，对于第一精度中间计算结果的尾数由 23bit 在末位补 29 个 0，将尾数位数扩展为 52bit，与 FP64 的尾数位数相同。

然后，对第三精度中间计算结果的阶码进行调整后，将调整后的中间计算结果进行累加得到第三精度计算结果。同样的，此处在此累加时，可以将调整后的中间计算结果输入累加器，得到计算结果。

此处，为了更好的体现本申请实施例的方案整体流程，以第一精度浮点数 A、B 的乘法计算为例进行说明。参照图 9，为本申请实施例提供的一种浮点数乘法计算方法的流程示意图。

对 A 和 B 分别输入第一精度浮点数分解逻辑，以对 A 和 B 分别执行第一精度浮点数分解。得到 A 对应的第二精度浮点数 A1 和 A0，以及 A1 和 A0 分别对应的阶码位移值；得到 B 对应的第二精度浮点数 B1 和 B0，以及 B1 和 B0 分别对应的阶码位移值。其中，分解逻辑可以采用硬件的逻辑电路实现。具体的分解方法请参见步骤 102。

然后，将分解自不同第一精度浮点数的第二精度浮点数进行组合，输入到第二精度乘法器中，得到每个组合对应的中间计算结果。具体的组合方法，请参见步骤 103；具体的中间计算结果的计算方法，请参见步骤 104。

再然后，对于每个组合对应的中间计算结果，执行阶码调整逻辑，使用该组合中的第二精度浮点数对应的阶码位移值，对其阶码进行调整，得到调整后的中间计算结果。具体步骤请参见步骤 105 下的调整方法，上述阶码调整可以由阶码调整逻辑电路执行。

最后，将每个组合对应的调整后的中间计算结果输入到累加器中，进行累加，即可得到最终的计算结果。具体步骤请参见步骤 105 下的方法描述，所述累加器为硬件累加器电路。

同样，为了更好的体现本申请实施例的方案整体流程，以第一精度浮点数 A、B 的乘法计算为例进行说明。参照图 10，为本申请实施例提供的又一种浮点数乘法计算方法的流程示意图。

对 A 和 B 分别输入第一精度浮点数分解逻辑，以对 A 和 B 分别执行第一精度浮点数分解。得到 A 对应的多个第二精度浮点数 A3、A2、A1 和 A0，以及 A3、A2、A1 和 A0 分别对应的阶码位移值；得到 B 对应的多个第二精度浮点数 B3、B2、B1 和 B0，以及 B3、B2、B1 和 B0 分别对应的阶码位移值。其中，分解逻辑可以采用硬件的逻辑电路实现。具体的分解方法请参见步骤 102。

然后，将分解自不同第一精度浮点数的第二精度浮点数进行组合，输入到第二精度乘法器中，得到每个组合对应的第三精度中间计算结果。具体的组合方法，请参见步骤 103；具体的中间计算结果的计算方法，请参见步骤 104。

再然后，对于每个组合对应的第三精度中间计算结果，执行格式转换逻辑，将每个组合对应的第三精度中间计算结果的格式转换为第一精度中间计算结果。具体步骤请参见步骤 105 下的格式转换方法，上述格式转换可以由格式转换逻辑电路执行。

再然后，对于每个组合对应的第一精度中间计算结果，执行阶码调整逻辑，使用该组合中的第二精度浮点数对应的阶码位移值，对其阶码进行调整，得到调整后的第一精度中间计算结果。具体步骤请参见步骤 105 下的调整方法，上述阶码调整可以由阶码调整逻辑电路执行。

最后，将每个组合对应的调整后的第一精度中间计算结果输入到累加器中，进行累加，即可得到最终的第一精度计算结果。具体步骤请参见步骤 105 下的方法描述，所述累加器为硬件累加器电路。

同样，为了更好的体现本申请实施例的方案整体流程，以第一精度浮点数 A、B 的乘法计算为例进行说明。参照图 11，为对本申请实施例提供的又一种浮点数乘法计算方法的流程示意图。

对 A 和 B 分别输入第一精度浮点数分解逻辑，以对 A 和 B 分别执行第一精度浮点数分解。得到 A 对应的多个第二精度浮点数 A1 和 A0，以及 A1 和 A0 分别对应的阶码位移值；得到 B 对应的多个第二精度浮点数 B1 和 B0，以及 B1 和 B0 分别对应的阶码位移值。其中，分解逻辑可以采用硬件的逻辑电路实现。具体的分解方法请参见步骤 102。

然后，将分解自不同第一精度浮点数的第二精度浮点数进行组合，输入到第二精度乘法器中，得到每个组合对应的第一精度中间计算结果。具体的组合方法，请参见步骤 103；具体的中间计算结果的计算方法，请参见步骤 104。

再然后，对于每个组合对应的第一精度中间计算结果，执行格式转换逻辑，将每个组合对应的第一精度中间计算结果的格式转换为第三精度中间计算结果。具体步骤请参见步骤 105 下的格式转换方法，上述格式转换可以由格式转换逻辑电路执行。

再然后，对于每个组合对应的第三精度中间计算结果，执行阶码调整逻辑，使用该组合中的第二精度浮点数对应的阶码位移值，对其阶码进行调整，得到调整后的第三精度中间计算结果。具体步骤请参见步骤 105 下的调整方法，上述阶码调整可以由阶码调整逻辑电路执行。

最后，将每个组合对应的调整后的第三精度中间计算结果输入到累加器中，进行累加，即可得到最终的第三精度计算结果。具体步骤请参见步骤 105 下的方法描述，所述累加器为硬件累加器电路。

另外，还需说明的是本申请实施例提供的浮点数计算方法可以实现精度高于或者等于第二精度的浮点数的计算。此处，第二精度指第二精度乘法器所支持计算的浮点数的精度。

例如，第二精度乘法器为半精度乘法器，即支持计算的浮点数精度为半精度，那么，本申请实施例可以实现对于半精度浮点数、单精度浮点数、双精度浮点数以及更高精度浮点数的计算。可以理解的是，对于半精度浮点数的计算，无需进行分解，只要将待计算的半精度浮点数输入半精度乘法器即可，而对于单精度浮点数以及更高精度浮点数的计算，则可以采用上述浮点数乘法计算的方法实现。

在本申请实施例中，对于精度较高的多个第一精度浮点数的计算，可以由精度较低的第二精度乘法器实现，而不必再使用第一精度乘法器。因此，可以实现在只有较低精度的第二精度乘法器的设备中，实现较高精度的第一精度浮点数的计算，而不必额外设计第一精度乘法器，有效的节省了计算资源。

基于相同的技术构思，本申请实施例还提供了一种浮点数乘法计算的装置，如图 7 所示，该装置包括：

获取模块 710，用于获取多个待计算的第一精度浮点数，具体可以实现上述步骤 201 中的获取功能，以及其他隐含步骤；

分解模块 720，用于对每个待计算的第一精度浮点数进行分解，得到至少两个第二精度浮点数，其中，所述第二精度浮点数的精度低于所述第一精度浮点数的精度，具体可以实现上述步骤 202 中的分解功能，以及其他隐含步骤；

组合模块 730, 用于确定各种由分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合, 具体可以实现上述步骤 203 中的组合功能, 以及其他隐含步骤;

输入模块 740, 用于将每种组合中的第二精度浮点数输入第二精度乘法器中, 得到每种组合对应的中间计算结果, 具体可以实现上述步骤 204 中的输入功能, 以及其他隐含步骤;

确定模块 750, 用于基于每种组合对应的中间计算结果, 确定所述多个待计算的第一精度浮点数的计算结果, 具体可以实现上述步骤 205 中的确定功能, 以及其他隐含步骤。

在一种可能的实现方式中, 所述分解模块 720, 还用于:

确定每个第二精度浮点数对应的阶码位移值;

所述确定模块 750, 用于:

分别基于每种组合中的第二精度浮点数对应的阶码位移值, 调整每种组合对应的中间计算结果的阶码, 得到调整后的中间计算结果;

将每种组合对应的调整后的中间计算结果进行求和运算, 得到所述多个第一精度浮点数的计算结果。

在一种可能的实现方式中, 所述确定模块 750, 用于:

将每种组合的第二精度浮点数对应的中间计算结果的阶码, 加上每种组合的第二精度浮点数对应的阶码位移值, 得到调整后的中间计算结果。

在一种可能的实现方式中, 所述中间计算结果为第一精度中间计算结果, 所述计算结果为第一精度计算结果。

在一种可能的实现方式中, 所述第一精度浮点数为单精度浮点数, 所述第二精度浮点数为半精度浮点数, 所述第一精度中间计算结果为单精度中间计算结果, 所述第一精度计算结果为单精度计算结果, 所述第二精度乘法器为半精度乘法器; 或者,

所述第一精度浮点数为双精度浮点数, 所述第二精度浮点数为单精度浮点数, 所述第一精度中间计算结果为双精度中间计算结果, 所述第一精度计算结果为双精度计算结果, 所述第二精度乘法器为单精度乘法器。

在一种可能的实现方式中, 所述输入模块 740, 用于:

将每种组合中的第二精度浮点数输入第二精度乘法器中, 得到每种组合对应的第一精度中间计算结果, 对每个第一精度中间计算结果进行格式转换得到每种组合对应的第三精度中间计算结果, 其中, 所述第三精度中间计算结果的精度高于所述第一精度中间计算结果;

所述确定模块 750, 用于:

分别基于每种组合中的第二精度浮点数对应的阶码位移值, 调整每种组合对应的第三精度中间计算结果的阶码, 得到调整后的第三精度中间计算结果;

将每种组合对应的调整后的第三精度中间计算结果进行求和运算, 得到所述多个第一精度浮点数的第三精度计算结果。

在一种可能的实现方式中, 所述输入模块 740, 用于:

对每个第一精度中间结果的阶码和尾数分别进行补零处理, 得到每种组合第二精度浮点数对应的第三精度中间计算结果。

在一种可能的实现方式中, 所述第一精度浮点数为单精度浮点数, 所述第二精度浮点数为半精度浮点数, 所述第一精度中间计算结果为单精度中间计算结果, 所述第三精度中间计算结果为双精度中间计算结果, 所述第三精度计算结果为双精度计算结果, 所述第二精度乘

法器为半精度乘法器。

在一种可能的实现方式中，所述输入模块 740，用于：

将每种组合中的第二精度浮点数输入第二精度乘法器中，得到每种组合对应的第三精度中间计算结果，对每个第三精度中间计算结果进行格式转换得到每种组合对应的第一精度中间计算结果；

所述确定模块 750，用于：

分别基于每种组合中的第二精度浮点数对应的阶码位移值，调整每种组合对应的第一精度中间计算结果的阶码，得到调整后的第一精度中间计算结果；

将各组第二精度浮点数对应的调整后的第一精度中间计算结果进行求和运算，得到所述多个第一精度浮点数的第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为半精度浮点数，所述第三精度中间计算结果为单精度中间计算结果，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

需要说明的是，上述各模块可以由处理器实现，或者处理器配合存储器来实现，或者，处理器执行存储器中的程序指令来实现。

还需要说明的是，上述实施例提供的浮点数乘法计算的装置在计算浮点数时，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，即将电子设备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，上述实施例提供的浮点数乘法计算的装置与浮点数乘法计算的方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

基于相同的技术构思，本申请实施例还提供了一种算术逻辑单元，所述算术逻辑单元为处理器中的硬件计算单元。如图 12 所示，该算术逻辑单元包括浮点数分解电路、第二精度乘法器、阶码调整电路和累加器，其中：

所述浮点数分解电路，用于将输入的每个待计算的第一精度浮点数分解为至少两个第二精度浮点数，并向所述阶码调整电路输出每个第二精度浮点数对应的阶码位移值，其中，所述第二精度浮点数的精度低于所述第一精度浮点数的精度；其中，多个第一精度浮点数可以依次输入浮点数分解电路进行分解计算，也可以由多个浮点数分解电路分别为一个第一精度浮点数提供分解计算。

所述第二精度乘法器，用于接收分解自不同的第一精度浮点数的两个第二精度浮点数组成的组合，将每种组合中的第二精度浮点数进行乘法运算，向所述阶码调整电路输出每种组合对应的中间计算结果；

所述阶码调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移值，调整输入的每种组合对应的中间计算结果的阶码，向所述累加器输出调整后的中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的中间计算结果进行求和运算，输出所述多个第一精度浮点数的计算结果。

在一种可能的实现方式中，所述阶码调整电路，用于将输入的每种组合中的第二精度浮点数对应的阶码位移值，与输入的每种组合对应的中间计算结果的阶码相加，向所述累

加器输出调整后的中间计算结果。

在一种可能的实现方式中，所述中间计算结果为第一精度中间计算结果，所述计算结果为第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第一精度计算结果为单精度计算结果，所述第二精度乘法器为半精度乘法器；或者，

所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为单精度浮点数，所述第一精度中间计算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为单精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括格式转换电路；

所述第二精度乘法器，具体用于将每种组合中的第二精度浮点数进行乘法运算，向所述格式转换电路输出每种组合对应的第一精度中间计算结果；

所述格式转换电路，用于将输入的每个第一精度中间计算结果进行格式转换，向所述阶码调整电路输出每种组合对应的第三精度中间计算结果，其中，所述第三精度中间计算结果的精度高于所述第一精度中间计算结果；

所述阶码调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移值，调整输入的每种组合对应的第三精度中间计算结果的阶码，向所述累加器输出调整后的第三精度中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的第三精度中间计算结果进行求和运算，输出所述多个第一精度浮点数的第三精度计算结果。

在一种可能的实现方式中，所述格式转换电路，用于：

将输入的每个第一精度中间计算结果的阶码和尾数分别进行补零处理，向所述阶码调整电路输出每种组合对应的第三精度中间计算结果。

在一种可能的实现方式中，所述第一精度浮点数为单精度浮点数，所述第二精度浮点数为半精度浮点数，所述第一精度中间计算结果为单精度中间计算结果，所述第三精度中间计算结果为双精度中间计算结果，所述第三精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括格式转换电路；

所述第二精度乘法器，用于将每种组合中的第二精度浮点数进行乘法运算，向所述格式转换电路输出每种组合对应的第三精度中间计算结果；

所述格式转换电路，用于将输入的每个第三精度中间计算结果进行格式转换，向所述阶码调整电路输出每种组合对应的第一精度中间计算结果；

所述阶码调整电路，用于基于输入的每种组合中的第二精度浮点数对应的阶码位移值，调整输入的每种组合对应的第一精度中间计算结果的阶码，向所述累加器输出调整后的第一精度中间计算结果；

所述累加器，用于将输入的每种组合对应的调整后的第一精度中间计算结果进行求和运算，输出所述多个第一精度浮点数的第一精度计算结果。

在一种可能的实现方式中，所述第一精度浮点数为双精度浮点数，所述第二精度浮点数为半精度浮点数，所述第三精度中间计算结果为单精度中间计算结果，所述第一精度中间计

算结果为双精度中间计算结果，所述第一精度计算结果为双精度计算结果，所述第二精度乘法器为半精度乘法器。

在一种可能的实现方式中，所述算术逻辑单元还包括，计算模式切换电路，其中：

所述计算模式切换电路，用于在设置为第二精度浮点数计算模式时，将所述浮点数分解电路和所述阶码调整电路置为无效；

所述第二精度乘法器，用于接收所述算术逻辑单元外部的输入的多组待计算的所述第二精度浮点数，对每组第二精度浮点数进行乘法运算，输入每组待计算的所述第二精度浮点数对应的中间计算结果；

所累加器，用于将输入的每组待计算的所述第二精度浮点数对应的中间计算结果进行求和运算，输出所述多组待计算的所述第二精度浮点数的计算结果。

如图 14 所示，该算术逻辑单元还可以支持模式切换，即第一精度浮点数运算模式，以及第二精度浮点数运算模式。在第一精度浮点数运算模式下，即可以通过浮点数分解电路、第二精度乘法器、格式转换电路、阶码调整电路和累加器，实现第一精度浮点数的乘法运算。在第二精度浮点数运算模式下，即可以使浮点数分解电路、格式转换电路和阶码调整电路无效，而只使用第二精度乘法器和累加器。将多组待计算的所述第二精度浮点数直接输入到第二精度乘法器中，输出多组待计算的所述第二精度浮点数分别对应的中间计算结果，再将输入累加器，进行累加运算，得到多组待计算的所述第二精度浮点数对应的计算结果。

此处需要说明的是，上述实施例提供的逻辑运算单元与浮点数乘法计算的方法实施例属于同一构思，其具体实现过程详见方法实施例，这里不再赘述。

参见图 8，本申请实施例提供了一种电子设备。该电子设备 800 包括至少一个处理器 801，总线系统 802，存储器 803。

上述处理器 801 可以是一个通用中央处理器（central processing unit，CPU），网络处理器（network processor，NP），图形处理器（graphics processing unit）微处理器，特定应用集成电路（application-specific integrated circuit，ASIC），或一个或多个用于控制本申请方案程序执行的集成电路。

上述总线系统 802 可包括一通路，在上述组件之间传送信息。

上述存储器 803 可以是只读存储器（read-only memory，ROM）或可存储静态信息和指令的其他类型的静态存储设备，随机存取存储器（random access memory，RAM）或者可存储信息和指令的其他类型的动态存储设备，也可以是电可擦可编程只读存储器（electrically erasable programmable read-only memory，EEPROM）、只读光盘（compact disc read-only memory，CD-ROM）或其他光盘存储、光碟存储（包括压缩光碟、激光碟、光碟、数字通用光碟、蓝光光碟等）、磁盘存储介质或者其他磁存储设备、或者能够用于携带或存储具有指令或数据结构形式的期望的程序代码并能够由计算机存取的任何其他介质，但不限于此。存储器可以是独立存在，通过总线与处理器相连接。存储器也可以和处理器集成在一起。

其中，存储器 803 用于存储执行本申请方案的应用程序代码，并由处理器 801 来控制执行。处理器 801 用于执行存储器 803 中存储的应用程序代码，从而实现本申请提出的浮点数计算方法。

在具体实现中，作为一种实施例，处理器 801 可以包括一个或多个 CPU。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，上述提到的计算机可读存储介质可以是只读存储器，磁盘或光盘等。

以上所述仅为本申请一个实施例，并不用以限制本申请，凡在本申请的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。