# 面向人工智能的浮点乘加器设计

陈正博,吴铁彬,郑　方,丁亚军
(江南计算技术研究所,江苏 无锡 214000)

**摘　要**:近年来,面向人工智能领域的芯片快速发展,低精度和混合精度的乘加运算能力是人工智能芯片计算能力的核心指标,同时乘加部件也是人工智能芯片功率的主要消费者。面向人工智能领域应用需求,研究高性能、低能耗、低开销的浮点乘加器,对人工智能芯片的研发具有重要意义。文中设计了一种面向 AI 的浮点乘加器,支持单精度、半精度、单半混合精度的浮点乘加运算,也支持 32 位、16 位和 8 位的整数乘法运算。该部件采用跨精度复用的设计思想,提出乘法器复用、移位器复用、前导零预测器复用等关键技术,在保证各类操作功能和性能的基础上,有效减少了芯片面积和功耗。文中完成了该部件的正确性测试和物理综合。实验结果表明,该部件能满足正确性要求,在 28 nm 工艺条件下,对比无复用设计至少减少 50.09% 的面积和 47.91% 的功耗,综合运行频率达到 2 GHz。

**关键词**:人工智能;浮点乘加器;单精度;半精度;单半混合精度
**中图分类号**:TP302　　**文献标识码**:A　　**文章编号**:1673-629X(2019)08-0096-06
doi:10.3969/j.issn.1673-629X.2019.08.019

# Design of Floating Point Multiply-Adder for Artificial Intelligence

CHEN Zheng-bo, WU Tie-bin, ZHENG Fang, DING Ya-jun
(Jiangnan Institute of Computing Technology, Wuxi 214000, China)

**Abstract**: In recent years, chips in the field of artificial intelligence have been developing rapidly. The multiply-add operation with low precision and mixed precision is the core index of the computing power of artificial intelligence chips, and the multiply-add components are also the main consumers of the power of artificial intelligence chips. It is of great significance for the research and development of artificial intelligence chips to study floating point multiplicator with high performance, low energy consumption and low overhead for the application demand in the field of artificial intelligence. In this paper, an AI-oriented floating point multiplicator is designed, which supports the floating point multiplication and addition operations with single precision, half precision and single-half-mixed precision, as well as the integer multiplication operations with 32, 16 and 8 bits. This design adopts the idea of cross-precision reuse and reduces the area and power of chips with ensuring all kind of operation functions and performance, by using proposed key technology like multiplier-reuse, shifter-reuse, LZA-reuse. We finish the correctness test and physical syntheses. The experiment shows that this architecture is correct in all operations. Compared with the no-reuse design, at least 50.09% of the area and 47.91% of the power consumption can be reduced under the condition of 28 nm process. The comprehensive operating frequency can reach 2 GHz.

**Key words**: artificial intelligence; floating point multiplicator; single precision; half precision; single-half-mixed precision

## 0　引　言

随着人工智能应用需求的快速增长,面向人工智能推理和训练的加速芯片已成为该领域重要的发展方向,许多国内外知名的大公司都大力投入相关研发。谷歌 TPU、NVIDIA Tesla V100、寒武纪 DianNao 等是具有代表性的人工智能加速芯片。谷歌公司的 TPU[1] 以脉动阵列为基础,支持半精度和 8 位整数的推理应用;寒武纪公司的 DianNao 系列[2-5] 芯片将神经网络各层分解成神经功能单元(NFU),支持半精度的推理应用;英伟达公司于 2017 年推出了 Tesla V100[6] 芯片,引入了张量计算单元(tensor core),支持双、单、半精度以及混合精度的训练和推理应用。

浮点乘加部件能力是衡量人工智能芯片性能的主要指标。以 NVIDIA Tesla V100 为例,该芯片可达到

**收稿日期**:2018-10-18　　**修回日期**:2019-02-18　　**网络出版时间**:2019-03-28
**基金项目**:国家重点研发计划项目(2016YFB0200501)
**作者简介**:陈正博(1996-),男,硕士研究生,CCF 会员(92026G),研究方向为高性能计算和人工智能;丁亚军,硕士,高级工程师,研究方向为高性能计算和人工智能。
**网络出版地址**:http://kns.cnki.net/kcms/detail/61.1450.TP.20190327.1633.080.html

7.5 TFLOPS 的双精度计算性能、15 TFLOPS 的单精度计算性能和 125 TFLOPS 的张量计算性能。Tesla V100 中包含 2 560 个 FP64 的计算单元和 5 120 个 FP32 的计算单元,同时引入了 640 个张量核心。作为人工智能卷积运算的基本单元,张量计算单元的运算方式如图 1 所示。

$$\begin{bmatrix} D_{0,0} & D_{0,1} & D_{0,2} & D_{0,3} \\ D_{1,0} & D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,0} & D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,0} & D_{3,1} & D_{3,2} & D_{3,3} \end{bmatrix} = \begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix} * \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} + \begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{bmatrix}$$

图 1　Tensor Core 基本运算方式

矩阵运算中,***A*** 和 ***B*** 都是半精度浮点数(FP16)组成的矩阵,***C*** 和 ***D*** 同为半精度浮点数(FP16)或单精度浮点数(FP32)组成的矩阵。当 ***C*** 和 ***D*** 矩阵同为半精度浮点数时,张量计算单元进行半精度的浮点乘加运算;当 ***C*** 和 ***D*** 矩阵同为单精度浮点数时,张量计算单元进行半精度乘半精度加单精度的单半混合精度浮点乘加运算。

考虑到人工智能领域对于各个精度的浮点运算和整数运算都存在需求,研究人员开始进行混合浮点乘加器的相关研究。NVIDIA 公司[7]提出了单精度和半精度混合的浮点融合乘加部件。该部件采用双模式实现不同功能,通过硬件复用技术,支持单精度浮点乘加、半精度浮点乘加和半精度浮点乘法后加法的运算模式。

为满足人工智能领域的不同应用场景,人工智能芯片需要集成多种不同精度的运算部件。但目前的人工智能芯片都采用多种独立运算部件集成的技术路线,这些独立的运算部件带来了较大的芯片面积和功耗开销。文中采用跨精度复用的思想,探索通过复用技术进行多种精度混合的浮点乘加部件实现的技术可行性。该研究成果对国产人工智能芯片研发具有重要借鉴意义。

文中设计并实现的面向人工智能的浮点乘加器,支持 IEEE-754[8]标准的单精度、并行 2 个半精度、半精度乘半精度加单精度(单半混合精度)的浮点乘加操作,也支持 32 位、并行 2 个 16 位/8 位的整数乘法操作。提出了乘法器复用、移位器复用和前导零预测复用等一系列关键技术,使用复用关键技术设计和实现该浮点乘加器,研究浮点乘加器中各模块的详细设计方法。在完成代码设计后,对其进行功能点和随机数测试,验证浮点乘加器的正确性,同时使用 DC 工具进行物理综合,结果显示综合运行频率可达 2 GHz。

## 1　面向 AI 的浮点乘加器的总体结构

文中设计的面向 AI 的浮点乘加器以经典的浮点融合乘加结构[9]为基础,采用 6 级全流水结构实现,总体实现结构如图 2 所示。浮点乘加器的输入数据为 32 位数据 A/B/C 和控制信号,在第 6 级站台输出的数据包括 32 位浮点乘加结果、浮点异常信号、32 位整数乘法结果和乘法溢出信号。该浮点乘加器支持单精度、单半混合精度和并行 2 个半精度的浮点乘加操作,也支持 32 位、并行 2 个 16 位/8 位的整数乘法操作。

文中设计的面向 AI 的浮点乘加器包括数据预处理、指数差值、对阶移位、乘法器、合并、前导零预测、规格化移位、指数调整、舍入等模块。图 2 中涂黑的乘法器、移位器、合并、前导零预测模块使用复用技术实现,既解决了跨精度复用功能,又可以优化整体实现的面积和功耗等指标。各模块的功能、跨精度复用技术和详细设计方法将在第 2 节详细介绍。

在设计的各模块实现后,首先进行功能点和随机数等正确性测试,然后对该设计进行物理综合,根据结果手动调节流水线的划分,使得各级站台之间的时延基本一致,最终的流水线站台划分合理后,再对比单精度和半精度浮点乘加部件比较性能。

## 2　各模块详细设计方法

本节将介绍各模块的功能、逻辑结构和详细设计方法。该设计中乘法器、移位器、合并与前导零预测等依据跨精度复用思想,使用复用技术实现。

### 2.1　数据预处理

数据预处理是对输入数据和控制信号提前进行处理,判断计算方式和数据是否符合规范。数据预处理模块的功能分别有指数尾数拆分、输入数据例外异常检测两部分。在数据进入到数据预处理模块时,根据控制信号和部件需要完成的功能,遵守一定的规则,如表 1 所示。

表 1　数据输入规则

| 部件功能 | 输入数据 *A* | 输入数据 *B* | 输入数据 *C* |
|---|---|---|---|
| 浮点乘加(减)、负乘加(减) | 操作数 *A* | 操作数 *B* | 操作数 *C* |
| 整数乘法 | 操作数 *A* | 操作数 *B* | 浮点数+0 |
| 浮点加(减) | 操作数 *A* | 浮点数+1 | 操作数 *C* |
| 浮点乘法 | 操作数 *A* | 操作数 *B* | 浮点数+0 |

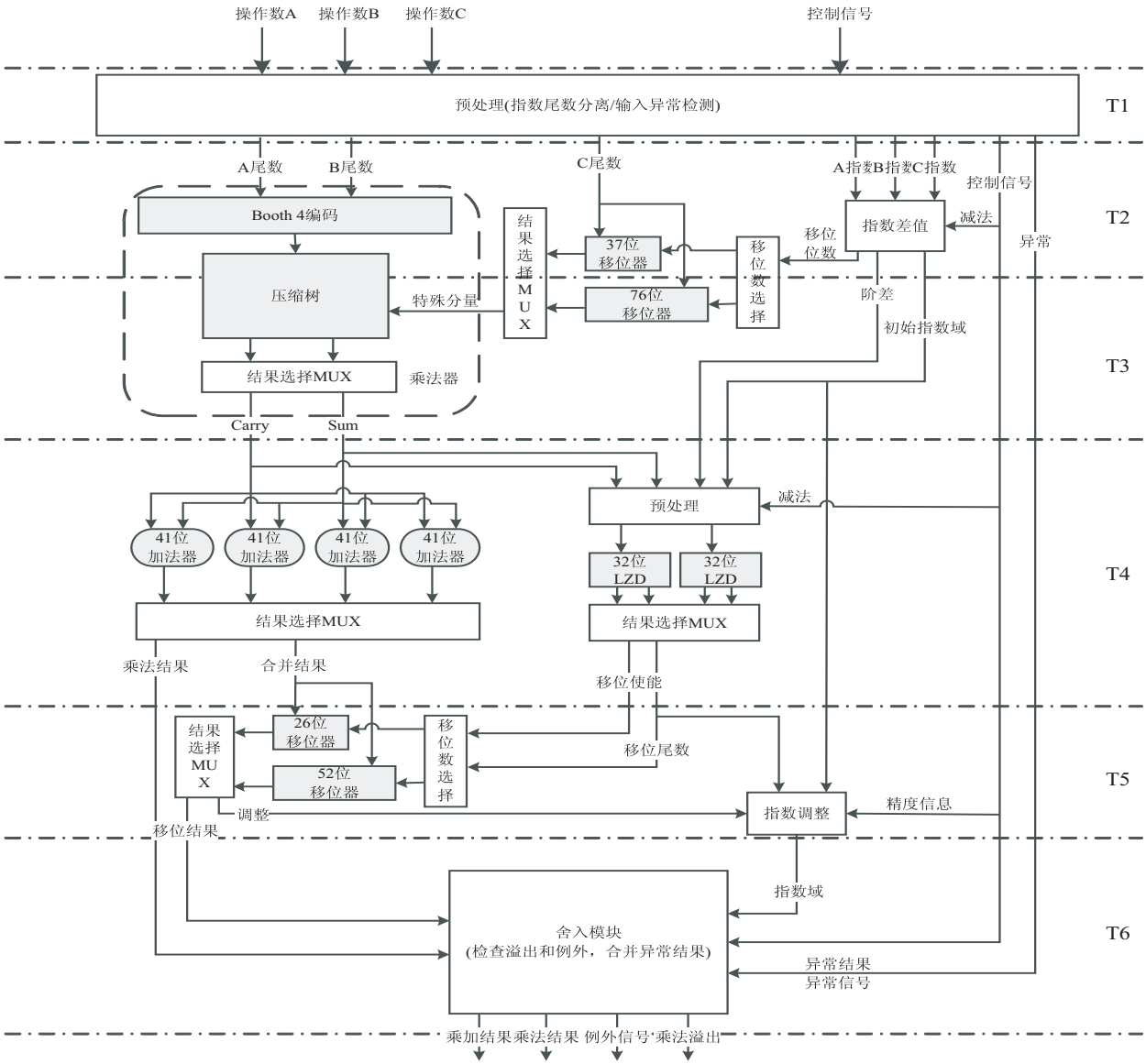


图 2 面向 AI 的浮点乘加器总体结构

输入数据满足规则后,还需要对输入异常和例外进行检测,异常情况直接旁路处理,以减少后续计算复杂度。例外和异常结果主要由非数和无穷数产生,除此以外,非规格化数在实际运算过程中会作为 0 处理。浮点乘加操作的异常产生条件和处理结果如表 2 所示。

表 2 浮点乘加操作异常条件与结果

| 操作数 $A * B + C$ | 结果 |
|---|---|
| 含非数 NaN | 按照 $C$、$B$ 和 $A$ 的顺序,以第一个非数 NaN 为结果 |
| 含∞ | ∞ 与非 0 的乘法,且 $C$ 非∞:±∞<br>∞ 与非 0 的乘法, $C$ 是∞,乘积结果符号与 $C$ 相同:±∞<br>∞ 与非 0 的乘法, $C$ 是∞,乘积结果符号与 $C$ 相反:非数 NaN<br>∞ 与 0 的乘法:非数 NaN<br>有限数乘法, $C$ 是∞:±∞ |

## 2.2 指数差值

指数差值模块是根据输入数据 $A$、$B$ 和 $C$ 的指数值,计算对阶移位量和初始结果的阶码。由于不同精度数据的指数值不同,尾数长度也不同,所以分别计算各个精度的对阶移位量和初始结果阶码。在此分别介绍单精度、半精度和单半混合精度的浮点乘加操作中指数差值模块的计算方法。

各种的精度浮点乘加操作,阶差的表达式均为: $d = C_e - A_e - B_e + \text{bias}$ 。

在得到指数域差值 $d$ 后,可以分别计算对阶移位量和指数阶码。单精度浮点乘加操作的尾数对阶移位示意如图 3 所示,其中数据位 0 后表示补 0,数据位 0 前表示尾数在小数点前的部分,小数点前表示尾数在小数点后的部分,可知对阶移位量应为 $\text{ASC} = 27 - d$ ,结果阶码的初值也应根据阶差大小决定。若 $d > 27$,此时 ASC 为负数,对阶移位量为 0,取加数阶码的大小作

为乘加结果基础解码;若 $1< d \leqslant 27$,取加数阶码+ASC 为乘加结果阶码;若 $d \leqslant 1$,取乘加阶码+2 为乘加结果阶码;若 ASC 过大,应限制其不超过 76 位,否则移位过多,移位结果一定为 0。

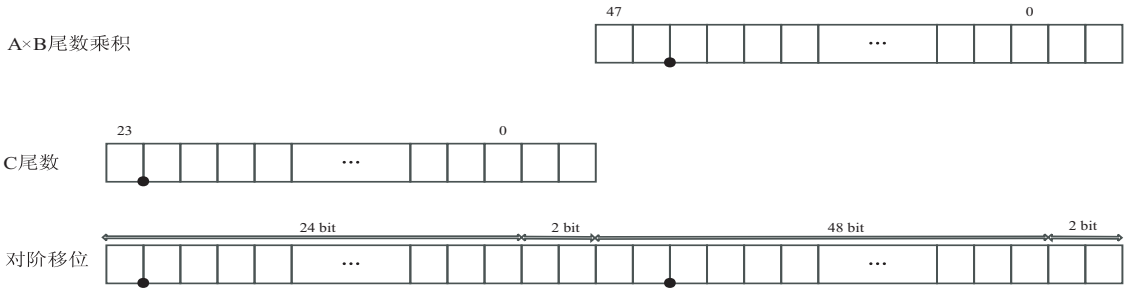


图 3　单精度浮点乘加尾数对阶移位示意

文中设计的浮点乘加器分别支持单精度、半精度和单半混合精度的浮点乘加运算,而不同精度的浮点乘加运算中的对阶移位量和乘加结果阶码的计算方式也不相同,需要分别进行分析。单精度、半精度和单半混合精度的浮点乘加运算的 ASC、对阶移位量和乘加结果阶码计算方式如表 3 所示。

表 3　对阶移位量和结果阶码的计算方式

| 计算精度 | ASC | 对阶移位量 | 乘加结果阶码 |
|---|---|---|---|
| 单精度 | $27-d$ | ASC,取[0,76]之间 | $d>27$,加数阶码<br>$1<d\leqslant 27$,加数阶码+ASC<br>$d\leqslant 1$,乘加阶码+2 |
| 半精度 | $14-d$ | ASC,取[0,37]之间 | $d>14$,加数阶码<br>$1<d\leqslant 14$,加数阶码+ASC<br>$d\leqslant 1$,乘加阶码+2 |
| 单半混合精度 | $27-d$ | ASC,取[0,50]之间 | $d>27$,加数阶码<br>$1<d\leqslant 27$,加数阶码+ASC<br>$d\leqslant 1$,乘加阶码+2 |

## 2.3　移位器

在该设计中,移位器包括对阶移位器和规格化移位器两种。对阶移位器的功能是进行尾数 $C$ 和乘法结果的对阶,规格化移位器的功能是进行中间尾数结果与规格化数的对阶。虽然各模块的功能不同,但移位器的跨精度复用思想一致。

移位器跨精度复用设计方法介绍如下:以对阶移位为例,单精度浮点乘加需求 76 位移位器,半精度浮点乘加需求 37 位移位器,混合精度浮点乘加需求 50 位移位器,考虑到该设计支持一个单精度乘加、并行两个半精度乘加、一个混合精度乘加的操作,使用一个 37 位和一个 76 位的桶形移位器[10]可满足需求,在移位器前后分别加上移位位数选择器和结果选择器,完成对阶移位器模块的复用设计。

以规格化移位为例,单精度浮点乘加需求 52 位移位器,半精度浮点乘加需求 26 位移位器,混合精度浮点乘加可直接复用单精度操作完成。同样地,使用一个 26 位和一个 52 位的移位器,并在前后分别加上移位位数选择器和结果选择器,即可完成规格化移位器模块的复用设计。

## 2.4　乘法器

乘法器是该设计中最重要的单元,包括部分积生成和部分积压缩两个部分,计算方法是按位拆分,分别计算再进行合并。乘法计算公式为:

$$A*B=\sum_{i=0}^{\frac{n}{2}-1}(b_{2i}+b_{2i-1}-2*b_{2i+1})*A*2^{2i}$$

将公式中括号内的内容进行 Booth-4 编码[11-12],生成以 $A$ 为基础的 16 个部分积,再通过部分积压缩树将部分积压缩成 Carry 和 Sum 结果。跨精度复用技术在部分积生成和部分积压缩树两部分中都有体现。进行并行两个的半精度数计算的操作中,在生成后 8 个部分积时,$A$ 和 $B$ 低位填充第一个半精度数,高位填零;生成前 8 个部分积时,$A$ 和 $B$ 低位填零,高位填充第二个半精度数。这样的填充方法可直接进行压缩,不影响最终结果,得到的后 8 个部分积压缩后得到第一个半精度数乘法结果,前 8 个部分积压缩后得到第二个半精度数乘法结果。部分积生成完成后进行部分积压缩,复用的部分积压缩树的结构如图 4 所示。

产生的 16 个部分积首先经过两层 4∶2 的 CSA 压缩树,此时产生的 4 个分量结果与 2 个半精度特殊分量分别进行 3∶2CSA 压缩后即可输出 2 个半精度结果;若其中两个分量结果直接与单精度特殊分量进行压缩可输出单半混合精度结果;而 4 个分量经过 4∶2CSA 压缩树后再与单精度特殊分量进行压缩可得到单精度结果,所以此时需要增加一个 MUX 选择器,根据控制信号中的精度信息进行分量选择,这样最终可得到半精度、单精度或是单半精度的浮点数乘法中间结果 Carry 和 Sum。值得注意的是,整数乘法也可借用此逻辑复用实现,8 位整数乘法套用半精度整数乘法,16 位整数乘法套用半精度整数乘法,32 位整数乘法套用单精度整数乘法即可。

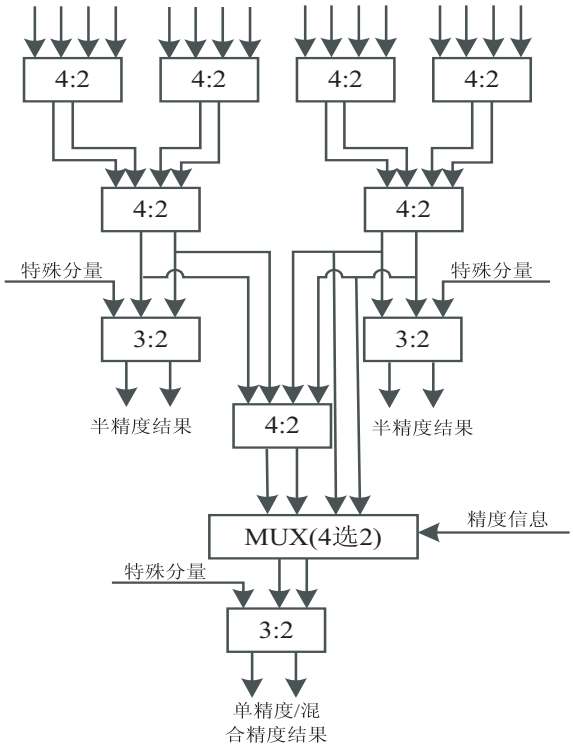


图 4 部分积压缩树

### 2.5 合并与前导零预测

得到 Carry 和 Sum 后,需要将这两个分量合并成一个分量,同时并行使用这两个分量进行前导零预测,预测合并后的分量前方有多少个零,这样在分量合并得到结果后,同时通过前导零预测模块得到的规格化移位位数,直接进行规格化移位,从而减少关键路径时延。

合并模块的功能是将 Sum 和 Carry 分量合并到一起,得到尾数计算的中间结果,通常使用加法器即可实现。合并模块的跨精度复用实现技术比较直观,由于单精度、半精度和单半混合精度的输入数据大小不同,按照最低位数拆分并处理,并行使用 4 个 41 位的超前进位加法器[13],输入进位分别设置为 0 和 1,4 个加法器根据控制信号分别输入各种精度的浮点或整数乘加的 Sum 和 Carry(位数不足时分别补 0 和 1),然后将加法器结果通过选择器可得到不同精度的合并数值。

前导零预测模块的功能是根据 Sum 和 Carry 分量预测计算两个分量合并后的首 1 位置。前导零预测模块的核心算法是 LZA 算法[14],即对两个操作数做减法计算,得到数零位串,对得到的数零位串通过树形结构 LZD 预测第一个 1 的位置,即可得到前导零预测的结果。

前导零预测的跨精度复用思想和设计技术与合并模块类似,单精度和半精度的数零位串的长度分别为 52 位和 26 位,考虑到 LZD 预测树的高度可扩展性,使用长度为 2 的次方数的 LZD 树形结构。按照最低位数拆分并处理,并行使用两个 32 位的 LZD 预测树,根据两个预测树分别输出的有效位和预测数值进行逻辑判断,通过选择器后,最终得到不同精度需求的规格化移位位数。由于前导零预测算法本身可能会存在 1 位的误差,且加法和减法可能存在 1 位误差,所以至多存在 2 位的误差,需要在规格化移位结束后进行检测,若存在误差则进行指数和尾数的修正。值得注意的是,在乘法器输出结果之后,单半混合精度与单精度结果的处理方式完全相同,不会额外增加逻辑。

### 2.6 指数调整与舍入

指数调整模块的功能是消除误差,得到准确的指数值。由于前导零预测本身可能存在至多 2 位的误差,故在规格化移位后进行简单的逻辑判断,计算出指数调整和尾数移位的数值,使用加法器即可得到最终结果的指数值,并对尾数再进行一定量的移位得到最终结果的尾数值。

舍入模块的功能是对最终结果进行舍入,并合并指数尾数结果得到最终的输出结果和异常或溢出信号。得到最终结果的指数值和尾数值后,结合舍入位和粘贴位的数据,舍入模块根据控制信号提供的 4 种不同的舍入方式进行舍入,得到正常的计算结果,同时与异常处理结果合并,即可最终输出浮点乘加结果、浮点异常信号、整数乘法结果和整数溢出信号,完成面向 AI 的浮点乘法器功能。

## 3 实验分析

### 3.1 正确性测试

完成代码设计后,撰写激励使用 PSL 语言对浮点乘加器进行正确性测试,验证面向人工智能的浮点乘加器是否满足正确性要求。正确性测试主要分为功能点测试和随机数测试两个方面。第一是功能点测试,对于各种异常的输入分别制造激励进行测试,直到所有的异常情况都被覆盖,从而检测浮点乘加器的异常处理机制是否完备;第二是随机数测试,对于表 1 中列举的各种部件功能,包括单精度、半精度和混合精度三种精度浮点乘加,和 32/16/8 位定点乘法,分别生成十万组不同的随机数数据,测试各个功能的结果是否保持正确,从而检测浮点乘加器的各功能都能正确使用。功能点和随机数测试的结果显示,文中设计的面向 AI 的浮点乘加器结果满足正确性要求。

### 3.2 物理综合

使用 Synopsys 公司的 Design Compiler 工具,基于 28 nm 工艺库条件对浮点乘加器进行物理综合。物理综合结果显示,该部件的综合运行频率可达 2 GHz。

为了评估该部件的功耗和面积等指标,体现该设计的优越性,在相同条件下,分别对单精度浮点乘加部

件[9]、半精度浮点乘加部件[9]和设计的面向 AI 的浮点乘加器进行物理综合,并对面积和功耗等性能指标进行对比,结果如表 4 所示。

表 4　性能对比

| 性能指标 | 文中设计 | 单精度[9] | 半精度[9] |
|---|---|---|---|
| 面积/$\mu m^2$ | 10 631.08 | 8 356.67 | 3 505.25 |
| 功耗/mW | 8.391 | 6.100 | 2.783 |

对比的结果可以发现,文中设计的面积比经典的单精度浮点乘加部件增加了 27.2%,功耗增加了 37.55%。考虑完全无复用的设计,完成面向 AI 的浮点乘加器的计算功能需求至少需要 1 个单精度部件和 2 个半精度部件,同时需要 1 个单半精度混合部件,且单半精度混合部件的面积和功耗比半精度部件要大,在计算无复用结果时可进行累加。最终结果显示,该设计比完全无复用设计的面积至少减少 50.09%,功耗至少减少 47.91%。

总而言之,该设计使用跨精度复用技术[15],在保证频率可接受范围内,大幅度减少了设计的硬件实现消耗,优化了浮点乘加器的性能,提出的复用技术具有显著意义。

## 4　结束语

文中研究和设计了一种面向 AI 的浮点乘加器,支持 32 位、并行 2 个 16 位/8 位的整数乘法运算,同时支持单精度、单半混合精度、并行 2 个半精度的浮点乘加运算。采用跨精度复用思路,详细讲解了各模块的功能、详细设计方法和跨精度复用实现方案。跨精度复用技术是文中的主要创新点,提出了乘法器复用、移位器复用和前导零预测复用等一系列关键技术。完成代码设计后,对浮点乘加器进行了正确性测试,结果表明浮点乘加器能够满足正确性要求。使用 DC 工具对浮点乘加器进行硬件综合,结果显示该设计类比于无复用设计,减少了至少 50.09% 的面积、47.91% 的功耗,综合运行频率可达 2 GHz。

**参考文献:**


[1] JOUPPI N P, YOUNG C, PATIL N, et al. In-data center performance analysis of a tensor processing unit[C]// ACM/IEEE 44th annual international symposium on computer architecture. Toronto: ACM, 2017: 1-12.

[2] CHEN Tianshi, DU Zidong, SUN Ninghui, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine learning[J]. ACM SIGPLAN Notices, 2014, 49(4): 269-284.

[3] CHEN Yunji, LUO Tao, LIU Shaoli, et al. DaDianNao: a machine learning supercomputer[C]//IEEE/ACM international symposium on microarchitecture. Cambridge, UK: IEEE, 2014: 609-622.

[4] DU Zidong, FASTHUBER R, CHEN Tianshi, et al. ShiDianNao: shifting vision processing closer to the sensor[J]. ACM SIGARCH Computer Architecture News, 2015, 43(3): 92-104.

[5] LIU Daofu, CHEN Tianshi, LIU Shaoli, et al. PuDianNao: a polyvalent machine learning accelerator[J]. ACM SIGPLAN Notices, 2015, 50(4): 369-381.

[6] NVIDIA. NVIDIA Tesla V100: the world's most advanced data center GPU[R]. [s.l.]: NVIDIA, 2017.

[7] NVIDIA Corporation. Logic circuitry configurable to perform 32-bit or dual 16-bit floating-point operations: United States, US20150169289A1[P]. 2015-06-18.

[8] ANSI/IEEE Std 754-2008. Binary floating-point arithmetic[S]. [s.l.]: IEEE, 2008.

[9] HONKENEK E, MONTOYE R K, COOK P W. Second-generation on RISC floating point with multiply-add fused[J]. IEEE Journal of Solid-State Circuits, 1990, 25(5): 1207-1213.

[10] 向　奔,蒋剑飞,毛志刚. 一种 DSP 桶形移位器的设计与优化方法[J]. 微电子学与计算机,2009,26(6):25-28.

[11] 周婉婷,李　磊. 基 4 BOOTH 编码的高速 32×32 乘法器的设计与实现[J]. 电子科技大学学报,2008,37:106-108.

[12] 何　军,朱　英. 一种 64 位 Booth 乘法器的设计与优化[J]. 计算机工程,2012,38(16):253-254.

[13] 王云贵,杨　靓. 一种快速超前进位加法器的优化设计[J]. 科学技术与工程,2010,10(33):8262-8266.

[14] HOKENEK E, MONTOYE R K. Leading-zero anticipator (LZA) in the IBM RISC System/6000 floating-point execution unit[J]. IBM Journal of Research & Development, 2010, 34(1): 71-77.

[15] 吴铁彬. 面向 LTE 的高性能向量浮点 MAC 单元的研究与实现[D]. 长沙:国防科技大学,2011.