# **H100操作数处理说明**

1. **FMA相关运算**
2. **操作数中出现非数NaN**

* **表1. 操作数中出现非数的输出结果描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A\*B | C | 结果 | 说明 |
| 1 | NaN\*有限数 | 有限数 | 固定NaN | 乘法操作数出现NaN |
| 2 | 有限数\*有限数 | NaN | 固定NaN | 加法操作数出现NaN |
| 3 | NaN\*有限数 | NaN | 固定NaN | 乘法、加法操作数出现NaN |
| 4 | NaN\*NaN | NaN | 固定NaN | 所有操作数都是NaN |
| 总结 | * 只要操作数中出现NaN，结果输出固定NaN（FP64除外） * 具体为：   FP64 : B优先级高于C高于A，即B为NaN时，优先将B的尾数最高位置1输出  FP32: 7fffffff  FP16: 7fff  BF16: 7fff | | | |
| 备注 | * NaN：Not a Number非数 | | | |

1. **操作数中出现0\*INF，或者INF-INF**

* **表2. 操作数中出现0\*INF，或者INF-INF的输出结果描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A\*B | C | 结果 | 说明 |
| 1 | 0\*INF | 有限数 | 固定NaN | 乘法操作数为0和INF |
| 2 | +INF | -INF | 固定NaN | 乘法操作数为INF且乘法结果为正无穷大，加法操作数为负无穷大 |
| 3 | -INF | +INF | 固定NaN | 乘法操作数为INF且乘法结果为负无穷大，加法操作数为正无穷大 |
| 总结 | * A\*B若为0\*INF，输出结果为固定NAN * A\*B+C若为INF-INF，-INF+INF时，输出结果为固定NAN * 具体为：   FP64 : 0xfff8000000000000  FP32: 7fffffff  FP16: 7fff  BF16: 7fff | | | |
| 备注 | * INF：无穷大 | | | |

1. **操作数中出现INF且进行正常计算**

* **表3. 操作数中出现INF且进行正常计算的输出结果描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A\*B | C | 结果 | 说明 |
| 1 | +INF | 有限数 | +INF | 乘法操作数为INF且结果为正无穷大 |
| 2 | -INF | 有限数 | -INF | 乘法操作数为INF且结果为负无穷大 |
| 3 | 有限数\*有限数 | +INF | +INF | 加法操作数为正无穷大 |
| 4 | 有限数\*有限数 | -INF | -INF | 加法操作数为负无穷大 |
| 5 | +INF | +INF | +INF | 乘法操作、加法操作数都是+INF |
| 6 | -INF | -INF | -INF | 乘法操作、加法操作数都是-INF |
| 总结 | * 当操作数中出现INF且进行正常计算，结果INF的符号位由具体结果决定； * 无穷值：   FP64 : +INF为7ff0\_0000\_0000\_0000，-INF为fff0\_0000\_0000\_0000  FP32: +INF为7f80\_0000，-INF为ff80\_0000  FP16: +INF为7c00，-INF为fc00  BF16: +INF为7f80，-INF为ff80 | | | |
| 备注 | * INF：无穷大 | | | |

1. 操作数中出现非规格化数

* **表4. 操作数中出现非规格化数的输出结果描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A\*B | C | 结果 | 说明 |
| 1 | DENOR\*有限数 | 有限数 | 正常运算 | 乘法操作数出现DENOR |
| 2 | 有限数\*有限数 | DENOR | 正常运算 | 加法操作数出现DENOR |
| 3 | DENOR\*有限数 | DENOR | 正常运算 | 乘法、加法操作数出现DENOR |
| 4 | DENOR\*DENOR | DENOR | 正常运算 | 所有操作数都是DENOR |
| 总结 | * 只要操作数中出现非规格化数DENOR，不再置零，均正常参与计算，且非规格化数异常位置有效 | | | |

1. **零的符号位**

* **表5. 零值运算的符号位描述**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A\*B | C | 结果 | 说明 |
| 1 | +0 | +0 | +0 | 乘法结果与加法操作数都为正零 |
| 2 | +0 | -0 | +0 | 乘法结果为正零，加法操作数为负零 |
| 3 | -0 | +0 | +0 | 乘法结果为负零，加法操作数为正零 |
| 4 | -0 | -0 | -0 | 乘法结果与加法操作数都为负零 |
| 总结 | * 零值的加减操作，符号位处理为：   +0+0=+0， +0-0=+0， -0+0=+0， -0-0=-0；   * 零值的乘法操作，符号位处理为：   (+0) \* (+0) = (+0) , (+0) \* (-0) = (-0) , (-0) \* (+0) = (-0) , (-0) \* (-0) = (+0)； | | | |

1. **比较操作相关**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 功能 | 操作数A或C | 结果 | 异常 |
| 1 | TCMPEQ/TCMPLT/TCMPLE/TCMPGT  （.bool/.tru模式） | 出现NaN | 0值 | - |
| 2 | TCMPLT/TCMPLE/TCMPGT  （.val模式） | 出现NaN | 输出数值  （若A、C同为NaN，则输出NaN） | - |
| 3 | 比较操作 | 出现INF | 正常比较 | - |
| 4 | 比较操作 | 出现DENOR | 正常比较 | - |
| 5 | 比较操作 | 出现0值 | 正常比较 | - |
|  | * 比较数值是指进行正常比较的相应数值结果 * NaN非数，DENOR非规格化数 * 具体为：   FP64 : A、C同为NaN，优先将C的尾数最高位置1输出  FP32: 7fffffff  FP16: 7fff  BF16: 7fff | | | |