BFMUL (indexed)

BFloat16 floating-point multiply vectors by indexed elements

Multiply all BFloat16 elements within each 128-bit segment of the first source vector by the specified element in the corresponding second source vector segment and place the results in the corresponding elements of the destination vector.

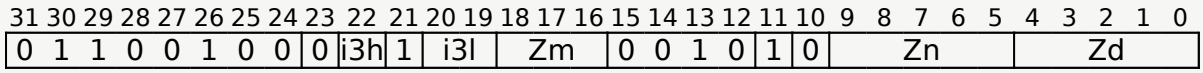
The elements within the second source vector are specified using an immediate index which selects the same element position within each 128-bit vector segment. The index range is from 0 to 7.

This instruction follows SVE2.1 non-widening BFloat16 numerical behaviors.

This instruction is unpredicated.

ID_AA64ZFR0_EL1.B16B16 indicates whether this instruction is implemented.

SVE2
(FEAT_SVE_B16B16)




```
BFMUL <Zd>.H, <Zn>.H, <Zm>.H[<imm>]
```

```
if (!HaveSVE2() && !HaveSME2()) || !IsFeatureImplemented(FEAT_SVE_B16B1
integer index = UInt(i3h:i3l);
integer n = UInt(Zn);
integer m = UInt(Zm);
integer d = UInt(Zd);
```

Assembler Symbols

<Zd>	Is the name of the destination scalable vector register, encoded in the "Zd" field.
<Zn>	Is the name of the first source scalable vector register, encoded in the "Zn" field.
<Zm>	Is the name of the second source scalable vector register Z0-Z7, encoded in the "Zm" field.
<imm>	Is the immediate index, in the range 0 to 7, encoded in the "i3h:i3l" fields.

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV 16;
```

```
constant integer eltspersegment = 128 DIV 16;
bits(VL) operand1 = Z[n, VL];
bits(VL) operand2 = Z[m, VL];
bits(VL) result;

for e = 0 to elements-1
    integer segmentbase = e - (e MOD eltspersegment);
    integer s = segmentbase + index;
    bits(16) element1 = Elem[operand1, e, 16];
    bits(16) element2 = Elem[operand2, s, 16];
    Elem[result, e, 16] = BFMul(element1, element2, FPCR[]);

Z[d, VL] = result;
```

Internal version only: isa v33.64, AdvSIMD v29.12, pseudocode no_diffs_2023_09_RC2, sve v2023-06_rel ; Build timestamp: 2023-09-18T17:56