BFMINNM (multiple vectors)

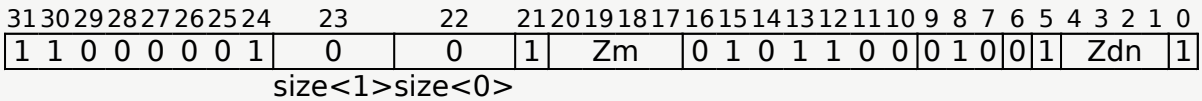
Multi-vector BFloat16 floating-point minimum number

Determine the minimum number value of BFloat16 elements of the two or four second source vectors and the corresponding BFloat16 elements of the two or four first source vectors and destructively place the results in the corresponding elements of the two or four first source vectors.
Regardless of the value of FPCR.AH, the behavior is as follows:

- Negative zero compares less than positive zero.
- If one element is numeric and the other is a quiet NaN, the result is the numeric value.
- When FPCR.DN is 0, if either element is a signaling NaN or if both elements are NaNs, the result is a quiet NaN.
- When FPCR.DN is 1, if either element is a signaling NaN or if both elements are NaNs, the result is Default NaN.

This instruction follows SME2.1 non-widening BFloat16 numerical behaviors corresponding to instructions that place their results in two or four SVE Z vectors.
This instruction is unpredicated.
ID_AA64SMFR0_EL1.B16B16 indicates whether this instruction is implemented.
It has encodings from 2 classes: Two registers and Four registers

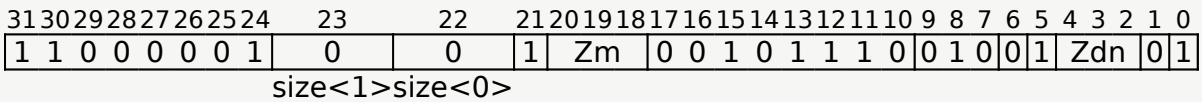
Two registers
(FEAT_SVE_B16B16)



BFMINNM { <Zdn1>.H-<Zdn2>.H }, { <Zdn1>.H-<Zdn2>.H }, { <Zm1>.H-<Zm2

```
if !HaveSME2() || !IsFeatureImplemented(FEAT_SVE_B16B16) then UNDEFINED
integer dn = UInt(Zdn:'0');
integer m = UInt(Zm:'0');
constant integer nreg = 2;
```

Four registers
(FEAT_SVE_B16B16)



```
    BFMINNM { <Zdn1>.H-<Zdn4>.H }, { <Zdn1>.H-<Zdn4>.H }, { <Zm1>.H-<Zm4
```

```
if !HaveSME2() || !IsFeatureImplemented(FEAT_SVE_B16B16) then UNDEFINED
integer dn = UInt(Zdn:'00');
integer m = UInt(Zm:'00');
constant integer nreg = 4;
```

Assembler Symbols

<Zdn1>	For the two registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zdn" times 2. For the four registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zdn" times 4.
<Zdn4>	Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zdn" times 4 plus 3.
<Zdn2>	Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zdn" times 2 plus 1.
<Zm1>	For the two registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zm" times 2. For the four registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zm" times 4.
<Zm4>	Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zm" times 4 plus 3.
<Zm2>	Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zm" times 2 plus 1.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV 16;
array [0..3] of bits(VL) results;

for r = 0 to nreg-1
    bits(VL) operand1 = Z[dn+r, VL];
    bits(VL) operand2 = Z[m+r, VL];
    for e = 0 to elements-1
        bits(16) element1 = Elem[operand1, e, 16];
        bits(16) element2 = Elem[operand2, e, 16];
        Elem[results[r], e, 16] = BFMinNum(element1, element2, FPCR[]);

for r = 0 to nreg-1
    Z[dn+r, VL] = results[r];
```

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