BFMAX

BFloat16 floating-point maximum (predicated)

Determine the maximum of active BFloat16 elements of the second source vector and corresponding BFloat16 elements of the first source vector and destructively place the results in the corresponding elements of the first source vector.
When FPCR.AH is 0, the behavior is as follows:

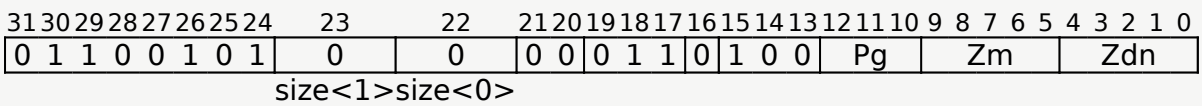
- Negative zero compares less than positive zero.
- When FPCR.DN is 0, if either element is a NaN, the result is a quiet NaN.
- When FPCR.DN is 1, if either element is a NaN, the result is Default NaN.

When FPCR.AH is 1, the behavior is as follows:

- If both elements are zeros, regardless of the sign of either zero, the result is the second element.
- If either element is a NaN, regardless of the value of FPCR.DN, the result is the second element.

Inactive elements in the destination vector register remain unmodified.
This instruction follows SVE2.1 non-widening BFloat16 numerical behaviors.
ID_AA64ZFR0_EL1.B16B16 indicates whether this instruction is implemented.

SVE2
(FEAT_SVE_B16B16)



BFMAX <Zdn>.H, <Pg>/M, <Zdn>.H, <Zm>.H

```
if (!HaveSVE2() && !HaveSME2()) || !IsFeatureImplemented(FEAT_SVE_B16B1
integer g = UInt(Pg);
integer dn = UInt(Zdn);
integer m = UInt(Zm);
```

Assembler Symbols

<Zdn>	Is the name of the first source and destination scalable vector register, encoded in the "Zdn" field.
<Pg>	Is the name of the governing scalable predicate register P0-P7, encoded in the "Pg" field.

<Zm>	Is the name of the second source scalable vector register, encoded in the "Zm" field.

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV 16;
bits(PL) mask = P[g, PL];
bits(VL) operand1 = Z[dn, VL];
bits(VL) operand2 = if AnyActiveElement(mask, 16) then Z[m, VL] else Ze
bits(VL) result;

for e = 0 to elements-1
    bits(16) element1 = Elem[operand1, e, 16];
    if ActivePredicateElement(mask, e, 16) then
        bits(16) element2 = Elem[operand2, e, 16];
        Elem[result, e, 16] = BFMax(element1, element2, FPCR[]);
    else
        Elem[result, e, 16] = element1;

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

Operational information

This instruction might be immediately preceded in program order by a MOVPRFX instruction. The MOVPRFX instruction must conform to all of the following requirements, otherwise the behavior of the MOVPRFX and this instruction is unpredictable:

- The MOVPRFX instruction must be unpredicated, or be predicated using the same governing predicate register and source element size as this instruction.
- The MOVPRFX instruction must specify the same destination register as this instruction.
- The destination register must not refer to architectural register state referenced by any other source operand register of this instruction.

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