FMAXNMP (scalar)

Floating-point Maximum Number of Pair of elements (scalar). This instruction compares two vector elements in the source SIMD&FP register and writes the largest of the floating-point values as a scalar to the destination SIMD&FP register.

Regardless of the value of *FPCR*.AH, the behavior is as follows for each pairwise operation:

- 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 can generate a floating-point exception. Depending on the settings in *FPCR*, the exception results in either a flag being set in *FPSR* or a synchronous exception being generated. For more information, see *Floating-point exception traps*.

Depending on the settings in the *CPACR_EL1*, *CPTR_EL2*, and *CPTR_EL3* registers, and the current Security state and Exception level, an attempt to execute the instruction might be trapped.

It has encodings from 2 classes: [Half-precision](#) and [Single-precision and double-precision](#)

Half-precision
(FEAT_FP16)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	1	1	0	0	sz	1	1	0	0	0	0	0	1	1	0	0	1	0			Rn				Rd		

o1

```
        FMAXNMP <V><d>, <Vn>.<T>

    if !IsFeatureImplemented(FEAT_FP16) then UNDEFINED;

    integer d = UInt(Rd);
    integer n = UInt(Rn);

    constant integer esize = 16;
    if sz == '1' then UNDEFINED;
    constant integer datasize = 32;
```

Single-precision and double-precision

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	0	sz	1	1	0	0	0	0	0	1	1	0	0	1	0			Rn				Rd		

o1

```
    FMAXNMP <V><d>, <Vn>.<T>

integer d = UInt(Rd);
integer n = UInt(Rn);

constant integer esize = 32 << UInt(sz);
constant integer datasize = esize * 2;
```

Assembler Symbols

<V>

For the half-precision variant: is the destination width specifier, encoded in "sz":

sz	<V>
0	H
1	RESERVED

For the single-precision and double-precision variant: is the destination width specifier, encoded in "sz":

sz	<V>
0	S
1	D

<d>   Is the number of the SIMD&FP destination register, encoded in the "Rd" field.

<Vn>   Is the name of the SIMD&FP source register, encoded in the "Rn" field.

<T>

For the half-precision variant: is the source arrangement specifier, encoded in "sz":

sz	<T>
0	2H
1	RESERVED

For the single-precision and double-precision variant: is the source arrangement specifier, encoded in "sz":

sz	<T>
0	2S
1	2D

Operation

```
CheckFPAdvSIMDEnabled64();
bits(datasize) operand = V[n, datasize];
V[d, esize] = Reduce(ReduceOp_FMAXNUM, operand, esize, FALSE);
```

Base
Instructions

SIMD&FP
Instructions

SVE
Instructions

SME
Instructions

Index by
Encoding

Sh
Pseu