	<u>S</u> .	h
Ps	eι	ı

Base
InstructionsSIMD&FP
InstructionsSVE
InstructionsSME
InstructionsIndex by
Encoding

CLASTB (SIMD&FP scalar)

Conditionally extract last element to SIMD&FP scalar register

From the source vector register extract the last active element, and then zero-extend that element to destructively place in the destination and first source SIMD & floating-point scalar register. If there are no active elements then destructively zero-extend the least significant element-size bits of the destination and first source SIMD & floating-point scalar register.

31 30 29	28 27	26	25	24	23 22	21	20	19	18	17	16	15	14	13	12	11 1	.0	9	8	7	6	5	4	3	2	1	0
0 0 0	0 0	1	0	1	size	1	0	1	0	1	1	1	0	0		Pg				Zm)				/dr	1	
											R																

```
CLASTB <V><dn>, <Pg>, <V><dn>, <Zm>.<T>
```

```
if !HaveSVE() && !HaveSME() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer g = UInt(Pg);
integer dn = UInt(Vdn);
integer m = UInt(Zm);
boolean isBefore = TRUE;</pre>
```

Assembler Symbols

<V>

Is a width specifier, encoded in "size":

size	<v></v>
0.0	В
01	Н
10	S
11	D

Is the number [0-31] of the source and destination SIMD&FP register, encoded in the "Vdn" field.

<Pg> Is the name of the governing scalable predicate register P0-P7, encoded in the "Pg" field.

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

Is the size specifier, encoded in "size":

size	<t></t>
0.0	В
01	Н
10	S
11	D

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
bits(PL) mask = \underline{P}[g, PL];
bits(esize) operand1 = V[dn, esize];
bits(VL) operand2 = \mathbb{Z}[m, VL];
bits(esize) result;
integer last = LastActiveElement(mask, esize);
if last < 0 then
    result = ZeroExtend(operand1, esize);
else
    if !isBefore then
        last = last + 1;
        if last >= elements then last = 0;
    result = <u>Elem</u>[operand2, last, esize];
V[dn, esize] = result;
```

<u>Base</u> <u>SIMD&FP</u> <u>SVE</u> <u>SME</u> <u>Index by</u> <u>Instructions</u> <u>Instructions</u> <u>Instructions</u> <u>Encoding</u>

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

Copyright © 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.

Sh Pseu