CLASTA (scalar)

Conditionally extract element after last to general-purpose register

From the source vector register extract the element after the last active element, or if the last active element is the final element extract element zero, and then zero-extend that element to destructively place in the destination and first source general-purpose register. If there are no active elements then destructively zero-extend the least significant element-size bits of the destination and first source general-purpose register.

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	0	0	0	0	1	0	1	size	1	1	0	0	0	0	1	0	1	Pg	Zm	Rdn
														B						

```
CLASTA <R><dn>, <Pg>, <R><dn>, <Zm>.<T>
```

```
if !HaveSVE() && !HaveSME() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer g = UInt(Pg);
integer dn = UInt(Rdn);
integer m = UInt(Zm);
constant integer csize = if esize < 64 then 32 else 64;
boolean isBefore = FALSE;
```

Assembler Symbols

<R>

Is a width specifier, encoded in “size”:

size	<R>
01	W
x0	W
11	X

<dn>	Is the number [0-30] of the source and destination general-purpose register or the name ZR (31), encoded in the "Rdn" 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.

<T>

Is the size specifier, encoded in “size”:

size	<T>
00	B
01	H
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 = P[g, PL];
bits(esize) operand1 = X[dn, esize];
bits(VL) operand2 = Z[m, VL];
bits(csize) result;
integer last = LastActiveElement(mask, esize);

if last < 0 then
    result = ZeroExtend(operand1, csize);
else
    if !isBefore then
        last = last + 1;
        if last >= elements then last = 0;
    result = ZeroExtend(Elem[operand2, last, esize], csize);

X[dn, csize] = result;
```

Operational information

If FEAT_SME is implemented and the PE is in Streaming SVE mode, then any subsequent instruction which is dependent on the general-purpose register written by this instruction might be significantly delayed.