PNEXT

Find next active predicate

An instruction used to construct a loop which iterates over all true elements in the vector select predicate register. If all elements in the first source predicate register are false it determines the first true element in the vector select predicate register, otherwise it determines the next true element in the vector select predicate register that follows the last true element in the first source predicate register. All elements of the destination predicate register are set to false, except the element corresponding to the determined vector select element, if any, which is set to true. Sets the first (N), none (Z), !last (C) condition flags based on the predicate result, and the V flag to zero.

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	1	0	0	1	0	1	size		0	1	1	0	0	1	1	1	0	0	0	1	0	Pv				0	Pdn			

PNEXT <Pdn>.<T>, <Pv>, <Pdn>.<T>

```
if !HaveSVE() && !HaveSME() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer v = UInt(Pv);
integer dn = UInt(Pdn);
```

Assembler Symbols

<Pdn> Is the name of the first source and destination scalable predicate register, encoded in the "Pdn" field.

<T> Is the size specifier, encoded in “size”:

size	<T>
00	B
01	H
10	S
11	D

<Pv> Is the name of the vector select predicate register, encoded in the "Pv" field.

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
bits(PL) mask = P[v, PL];
bits(PL) operand = P[dn, PL];
```

```
bits(PL) result;
constant integer psize = esize DIV 8;

integer next = LastActiveElement(operand, esize) + 1;

while next < elements && (!ActivePredicateElement(mask, next, esize)) d
    next = next + 1;

result = Zeros(PL);
if next < elements then
    Elem[result, next, psize] = ZeroExtend('1', psize);

PSTATE.<N,Z,C,V> = PredTest(mask, result, esize);
P[dn, PL] = 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 NZCV condition flags written by this instruction might be significantly delayed.

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