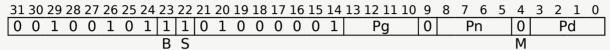
<u>x by</u>	<u>Sh</u>
ling	<u>Pseu</u>

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

BRKBS

Break before first true condition, setting the condition flags

Sets destination predicate elements up to but not including the first active and true source element to true, then sets subsequent elements to false. Inactive elements in the destination predicate register are set to zero. Sets the first (N), none (Z), !last (C) condition flags based on the predicate result, and the V flag to zero.



BRKBS <Pd>.B, <Pg>/Z, <Pn>.B

```
if !HaveSVE() && !HaveSME() then UNDEFINED;
constant integer esize = 8;
integer g = UInt(Pg);
integer n = UInt(Pn);
integer d = UInt(Pd);
boolean merging = FALSE;
boolean setflags = TRUE;
```

Assembler Symbols

<pu></pu>	Is the name of the destination scalable predicate register, encoded in the "Pd" field.
<pg></pg>	Is the name of the governing scalable predicate register,

encoded in the "Pg" field.

<Pn> Is the name of the source scalable predicate register,

encoded in the "Pn" field.

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(PL) operand = P[n, PL];
bits(PL) operand2 = P[d, PL];
boolean break = FALSE;
bits(PL) result;
constant integer psize = esize DIV 8;

for e = 0 to elements-1
   boolean element = ActivePredicateElement(operand, e, esize);
   if ActivePredicateElement(mask, e, esize) then
        break = break | element;
```

```
bit pbit = if !break then '1' else '0';
        Elem[result, e, psize] = ZeroExtend(pbit, psize);
elsif merging then
        bit pbit = PredicateElement(operand2, e, esize);
        Elem[result, e, psize] = ZeroExtend(pbit, psize);
else
        Elem[result, e, psize] = ZeroExtend('0', psize);

if setflags then
    PSTATE.<N,Z,C,V> = PredTest(mask, result, esize);
P[d, 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.

BaseSIMD&FPSVESMEIndex byInstructionsInstructionsInstructionsInstructions

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