$\mathbf{y}$	<u>Sn</u>
<u>ıg</u>	<u>Pseu</u>

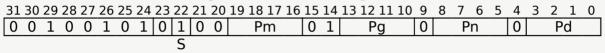
<u>Base</u>	SIMD&FP	<u>SVE</u>	<u>SME</u>	Index by
<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	<b>Encoding</b>

### **ANDS**

Bitwise AND predicates, setting the condition flags

Bitwise AND active elements of the second source predicate with corresponding elements of the first source predicate and place the results in the corresponding elements of the destination predicate. 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.

This instruction is used by the alias MOVS (predicated).



```
ANDS \langle Pd \rangle.B, \langle Pg \rangle / Z, \langle Pn \rangle.B, \langle Pm \rangle.B
```

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

# **Assembler Symbols**

<pd></pd>	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></pn>	Is the name of the first source scalable predicate register, encoded in the "Pn" field.
<pm></pm>	Is the name of the second source scalable predicate register, encoded in the "Pm" field.

#### **Alias Conditions**

Alias	Is preferred when
MOVS (predicated)	S == '1' && Pn == Pm

#### 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) operand1 = P[n, PL];
bits(PL) operand2 = P[m, PL];
bits(PL) result;
constant integer psize = esize DIV 8;

for e = 0 to elements-1
    bit element1 = PredicateElement(operand1, e, esize);
    bit element2 = PredicateElement(operand2, e, esize);
    if ActivePredicateElement(mask, e, esize) then
        Elem[result, e, psize] = ZeroExtend(element1 AND element2, 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\_SVE2 is implemented or FEAT\_SME is implemented, then if PSTATE.DIT is 1:

- The execution time of this instruction is independent of:
  - The values of the data supplied in any of its operand registers when its governing predicate register contains the same value for each execution.
  - The values of the NZCV flags.
- The response of this instruction to asynchronous exceptions does not vary based on:
  - The values of the data supplied in any of its operand registers when its governing predicate register contains the same value for each execution.
  - The values of the NZCV flags.

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.

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

 $Internal\ version\ only: is a\ v33.64,\ AdvSIMD\ v29.12,\ pseudocode\ no\_diffs\_2023\_09\_RC2,\ sve\ v2023-06\_rel\ ;\ Build\ timestamp:\ 2023-09-18T17:56$ 

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

<u>Sh</u> Pseu