

MOVA (vector to array, two registers)

Move two vector registers to two ZA single-vector groups

The instruction operates on two ZA single-vector groups. The vector numbers forming the single-vector group within each half of the ZA array are selected by the sum of the vector select register and immediate offset, modulo half the number of ZA array vectors.

The vector group symbol VGx2 indicates that the instruction operates on two ZA single-vector groups.

The preferred disassembly syntax uses a 64-bit element size, but an assembler should accept any element size if it is used consistently for all operands. The vector group symbol is preferred for disassembly, but optional in assembler source code.

This instruction is unpredicated.

This instruction is used by the alias [MOV \(vector to array, two registers\)](#).

SME2

(FEAT_SME2)

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
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Rv	0	1	0			Zn		0	0	0		off3		

MOVA ZA.D[<Wv>, <offs>{, VGx2}], { <Zn1>.D-<Zn2>.D }

```
if !HaveSME2() then UNDEFINED;
integer v = UInt('010':Rv);
integer offset = UInt(off3);
integer n = UInt(Zn:'0');
constant integer nreg = 2;
```

Assembler Symbols

- <Wv> Is the 32-bit name of the vector select register W8-W11, encoded in the "Rv" field.
- <offs> Is the vector select offset, in the range 0 to 7, encoded in the "off3" field.
- <Zn1> Is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 2.
- <Zn2> Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zn" times 2 plus 1.

Operation

```
CheckStreamingSVEAndZAEnabled();
constant integer VL = CurrentVL;
```

```

integer vectors = VL DIV 8;
integer vstride = vectors DIV nreg;
bits(32) vbase = X[v, 32];
integer vec = (UInt(vbase) + offset) MOD vstride;

for r = 0 to nreg-1
    bits(VL) result = Z[n + r, VL];
    ZAvector[vec, VL] = result;
    vec = vec + vstride;

```

Operational information

If PSTATE.DIT is 1:

- The execution time of this instruction is independent of:
 - The values of the data supplied in any of its registers.
 - 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 registers.
 - The values of the NZCV flags.

[Base
Instructions](#)

[SIMD&FP
Instructions](#)

[SVE
Instructions](#)

[SME
Instructions](#)

[Index by
Encoding](#)

[Sh
Pseudocode](#)

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.