MOVAZ (tile to vector, single)

Move and zero ZA tile slice to vector register

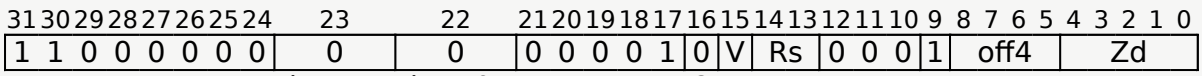
The instruction operates on a horizontal or vertical slice within a named ZA tile of the specified element size. The tile slice is zeroed after moving its contents to the destination vector.

The slice number within the tile is selected by the sum of the slice index register and immediate offset, modulo the number of such elements in a vector. The immediate offset is in the range 0 to the number of elements in a 128-bit vector segment minus 1.

This instruction is unpredicated.

It has encodings from 5 classes: 8-bit , 16-bit , 32-bit , 64-bit and 128-bit

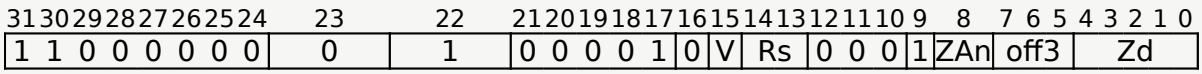
8-bit
(FEAT_SME2p1)



MOVAZ <Zd>.B, ZA0<HV>.B[<Ws>, <offs>]

```
if !HaveSME2p1() then UNDEFINED;
integer s = UInt('011':Rs);
integer n = 0;
integer offset = UInt(off4);
constant integer esize = 8;
integer d = UInt(Zd);
boolean vertical = V == '1';
```

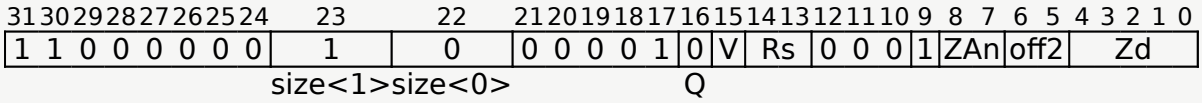
16-bit
(FEAT_SME2p1)



MOVAZ <Zd>.H, <ZAn><HV>.H[<Ws>, <offs>]

```
if !HaveSME2p1() then UNDEFINED;
integer s = UInt('011':Rs);
integer n = UInt(ZAn);
integer offset = UInt(off3);
constant integer esize = 16;
integer d = UInt(Zd);
boolean vertical = V == '1';
```

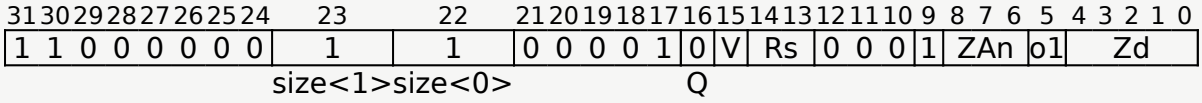
32-bit
(FEAT_SME2p1)



MOVAZ <Zd>.S, <ZAn><HV>.S[<Ws>, <offs>]

```
if !HaveSME2p1() then UNDEFINED;
integer s = UInt('011':Rs);
integer n = UInt(ZAn);
integer offset = UInt(off2);
constant integer esize = 32;
integer d = UInt(Zd);
boolean vertical = V == '1';
```

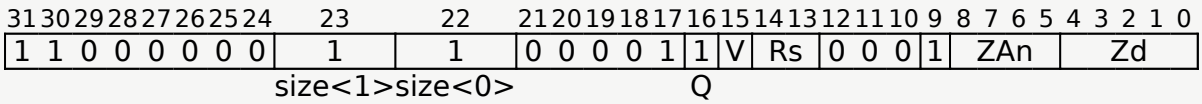
64-bit
(FEAT_SME2p1)



MOVAZ <Zd>.D, <ZAn><HV>.D[<Ws>, <offs>]

```
if !HaveSME2p1() then UNDEFINED;
integer s = UInt('011':Rs);
integer n = UInt(ZAn);
integer offset = UInt(o1);
constant integer esize = 64;
integer d = UInt(Zd);
boolean vertical = V == '1';
```

128-bit
(FEAT_SME2p1)



MOVAZ <Zd>.Q, <ZAn><HV>.Q[<Ws>, <offs>]

```
if !HaveSME2p1() then UNDEFINED;
integer s = UInt('011':Rs);
integer n = UInt(ZAn);
integer offset = 0;
constant integer esize = 128;
integer d = UInt(Zd);
boolean vertical = V == '1';
```

Assembler Symbols

<Zd>	Is the name of the destination scalable vector register, encoded in the "Zd" field.

<ZAn>

For the 16-bit variant: is the name of the ZA tile ZA0-ZA1 to be accessed, encoded in the "ZAn" field.

For the 32-bit variant: is the name of the ZA tile ZA0-ZA3 to be accessed, encoded in the "ZAn" field.

For the 64-bit variant: is the name of the ZA tile ZA0-ZA7 to be accessed, encoded in the "ZAn" field.

For the 128-bit variant: is the name of the ZA tile ZA0-ZA15 to be accessed, encoded in the "ZAn" field.

<HV>

Is the horizontal or vertical slice indicator, encoded in “V”:

V	<HV>
0	H
1	V

<Ws>

Is the 32-bit name of the slice index register W12-W15, encoded in the "Rs" field.

<offs>

For the 8-bit variant: is the slice index offset, in the range 0 to 15, encoded in the "off4" field.

For the 16-bit variant: is the slice index offset, in the range 0 to 7, encoded in the "off3" field.

For the 32-bit variant: is the slice index offset, in the range 0 to 3, encoded in the "off2" field.

For the 64-bit variant: is the slice index offset, in the range 0 to 1, encoded in the "o1" field.

For the 128-bit variant: is the slice index offset 0.

Operation

```
CheckStreamingSVEAndZAEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer dim = VL DIV esize;
bits(32) index = X[s, 32];
integer  slice = (UInt(index) + offset) MOD dim;
bits(VL) operand = ZAslice[n, esize, vertical, slice, VL];
ZAslice[n, esize, vertical, slice, VL] = Zeros(VL);
Z[d, VL] = operand;
```

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.

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