ZERO (quad-vector)

Zero ZA quad-vector groups

The instruction zeroes one, two, or four ZA quad-vector groups. The lowest of the four consecutive vector numbers forming the quad-vector group within all of, each half of, or each quarter of the ZA array are selected by the sum of the vector select register and immediate offset, modulo all, half, or quarter the number of ZA array vectors.

The vector group symbol, VGx2 or VGx4, indicates that the ZA operand consists of two or four ZA quad-vector groups respectively.

It has encodings from 3 classes: One ZA quad-vector , Two ZA quad-vectors and Four ZA quad-vectors

One ZA quad-vector
(FEAT_SME2p1)

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	1	1	1	0	1	Rv		0	0	0	0	0	0	0	0	0	0	0	off2	

ZERO ZA.D[<Wv>, <offs1>:<offs4>]

```
if !HaveSME2p1() then UNDEFINED;
integer v = UInt('010':Rv);
integer offset = UInt(off2:'00');
constant integer ngrp = 1;
constant integer nvec = 4;
```

Two ZA quad-vectors
(FEAT_SME2p1)

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	1	1	1	1	0	Rv		0	0	0	0	0	0	0	0	0	0	0	0	o1

ZERO ZA.D[<Wv>, <offs1>:<offs4>, VGx2]

```
if !HaveSME2p1() then UNDEFINED;
integer v = UInt('010':Rv);
integer offset = UInt(o1:'00');
constant integer ngrp = 2;
constant integer nvec = 4;
```

Four ZA quad-vectors
(FEAT_SME2p1)

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	1	1	1	1	1	Rv		0	0	0	0	0	0	0	0	0	0	0	0	o1

```
ZERO ZA.D[<Wv>, <offs1>:<offs4>, VGx4]
```

```
if !HaveSME2p1() then UNDEFINED;
integer v = UInt('010':Rv);
integer offset = UInt(o1:'00');
constant integer ngrp = 4;
constant integer nvec = 4;
```

Assembler Symbols

<Wv>	Is the 32-bit name of the vector select register W8-W11, encoded in the "Rv" field.
<offs1>	For the one ZA quad-vector variant: is the vector select offset, pointing to first of four consecutive vectors, encoded as "off2" field times 4. For the four ZA quad-vectors and two ZA quad-vectors variant: is the vector select offset, pointing to first of four consecutive vectors, encoded as "o1" field times 4.
<offs4>	For the one ZA quad-vector variant: is the vector select offset, pointing to last of four consecutive vectors, encoded as "off2" field times 4 plus 3. For the four ZA quad-vectors and two ZA quad-vectors variant: is the vector select offset, pointing to last of four consecutive vectors, encoded as "o1" field times 4 plus 3.

Operation

```
CheckStreamingSVEAndZAEnabled();
constant integer VL = CurrentVL;
integer vectors = VL DIV 8;
integer vstride = vectors DIV ngrp;
bits(32) vbase = X[v, 32];
integer vec = (UInt(vbase) + offset) MOD vstride;
vec = vec - (vec MOD nvec);

for r = 0 to ngrp-1
    for i = 0 to nvec-1
        ZAvector[vec + i, VL] = Zeros(VL);
    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.