XAR

Bitwise exclusive OR and rotate right by immediate

Bitwise exclusive OR the corresponding elements of the first and second source vectors, then rotate each result element right by an immediate amount. The final results are destructively placed in the corresponding elements of the destination and first source vector. This instruction is unpredicated.

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
0	0	0	0	0	1	0	0	tszh	1	tszl	imm3	0	0	1	1	0	1	Zm	Zdn

      XAR <Zdn>.<T>, <Zdn>.<T>, <Zm>.<T>, #<const>

```
if !HaveSVE2() && !HaveSME() then UNDEFINED;
bits(4) tsize = tszh:tszl;
if tsize == '0000' then UNDEFINED;
constant integer esize = 8 << HighestSetBit(tsize);
integer m = UInt(Zm);
integer dn = UInt(Zdn);
integer rot = (2 * esize) - UInt(tsize:imm3);
```

Assembler Symbols

<Zdn>   Is the name of the first source and destination scalable vector register, encoded in the "Zdn" field.

<T>   Is the size specifier, encoded in "tszh:tszl":

tszh	tszl	<T>
00	00	RESERVED
00	01	B
00	1x	H
01	xx	S
1x	xx	D

<Zm>   Is the name of the second source scalable vector register, encoded in the "Zm" field.

<const>   Is the immediate shift amount, in the range 1 to number of bits per element, encoded in "tszh:tszl:imm3".

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV esize;
bits(VL) operand1 = Z[dn, VL];
```

```
    bits(VL) operand2 = Z[m, VL];
    bits(VL) result;

    for e = 0 to elements-1
        bits(esize) element1 = Elem[operand1, e, esize];
        bits(esize) element2 = Elem[operand2, e, esize];
        Elem[result, e, esize] = ROR(element1 EOR element2, rot);
    Z[dn, VL] = 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 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.

This instruction might be immediately preceded in program order by a MOVPRFX instruction. The MOVPRFX instruction must conform to all of the following requirements, otherwise the behavior of the MOVPRFX and this instruction is unpredictable:

- The MOVPRFX instruction must be unpredicated.
- The MOVPRFX instruction must specify the same destination register as this instruction.
- The destination register must not refer to architectural register state referenced by any other source operand register of this instruction.