SQRSHRUN

Multi-vector signed saturating rounding shift right unsigned narrow by immediate and interleave

Shift right by an immediate value, the signed integer value in each element of the four source vectors and place the four-way interleaved rounded results in the quarter-width destination elements. Each result element is saturated to the quarter-width N-bit element's unsigned integer range 0 to (2^N)-1. The immediate shift amount is an unsigned value in the range 1 to number of bits per source element.
This instruction is unpredicated.

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 1	tsize	1	imm5	1 1 0 1 1	1	Zn	1	0	Zd
					N			U	

       SQRSHRUN <Zd>.<T>, { <Zn1>.<Tb>-<Zn4>.<Tb> }, #<const>

```
if !HaveSME2() then UNDEFINED;
if tsize == '00' then UNDEFINED;
constant integer esize = 8 << HighestSetBit(tsize);
integer n = UInt(Zn:'00');
integer d = UInt(Zd);
integer shift = (8 * esize) - UInt(tsize:imm5);
```

Assembler Symbols

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

<T>

       Is the size specifier, encoded in "tsize":

tsize	<T>
00	RESERVED
01	B
1x	H

<Zn1>   Is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 4.

<Tb>

Is the size specifier, encoded in "tsize":

tsize	<Tb>
00	RESERVED
01	S
1x	D

<Zn4>

Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zn" times 4 plus 3.

<const>

Is the immediate shift amount, in the range 1 to number of bits per source element, encoded in "tsize:imm5".

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV (4 * esize);
bits(VL) result;

for e = 0 to elements-1
    for i = 0 to 3
        bits(VL) operand = Z[n+i, VL];
        bits(4 * esize) element = Elem[operand, e, 4 * esize];
        integer res = (SInt(element) + (1 << (shift-1))) >> shift;
        Elem[result, 4*e + i, esize] = UnsignedSat(res, esize);

Z[d, VL] = result;
```