SQRDMLSH (vectors)

Signed saturating rounding doubling multiply-subtract high from accumulator (unpredicated)

Multiply then double the corresponding signed elements of the first and second source vectors, and destructively subtract the rounded high half of each result from the corresponding elements of the addend and destination vector. Each destination element is saturated to the N-bit element's signed integer range $-2^{(N-1)}$ to $(2^{(N-1)})-1$. 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 1 0 0 0 1 0 0	size	0	Zm	0 1 1 1 0	1	Zn	Zda
					S		

        SQRDMLSH **<Zda>.<T>, <Zn>.<T>, <Zm>.<T>**

```
if !HaveSVE2() && !HaveSME() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer n = UInt(Zn);
integer m = UInt(Zm);
integer da = UInt(Zda);
```

Assembler Symbols

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

<T>

              Is the size specifier, encoded in "size":

size	<T>
00	B
01	H
10	S
11	D

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

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

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV esize;
bits(VL) operand1 = Z[n, VL];
bits(VL) operand2 = Z[m, VL];
```

```
    bits(VL) operand3 = Z[da, VL];
    bits(VL) result;

    for e = 0 to elements-1
        integer element1 = SInt(Elem[operand1, e, esize]);
        integer element2 = SInt(Elem[operand2, e, esize]);
        integer element3 = SInt(Elem[operand3, e, esize]);
        integer res = (element3 << esize) - (2 * element1 * element2);
        Elem[result, e, esize] = SignedSat((res + (1 << (esize - 1))) >> es

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

Operational information

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.

Base Instructions	SIMD&FP Instructions	SVE Instructions	SME Instructions	Index by Encoding	Sh Pseu