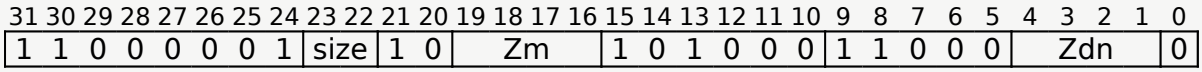
ADD (to vector)

Add replicated single vector to multi-vector with multi-vector result

Add elements of the second source vector to the corresponding elements of the two or four first source vectors and destructively place the results in the corresponding elements of the two or four first source vectors.
This instruction is unpredicated.
It has encodings from 2 classes: Two registers and Four registers

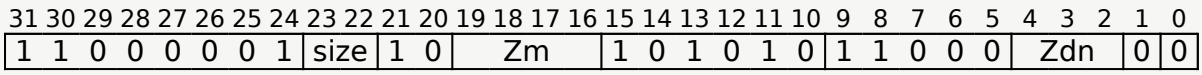
Two registers
(FEAT_SME2)



ADD { <Zdn1>.<T>-<Zdn2>.<T> }, { <Zdn1>.<T>-<Zdn2>.<T> }, <Zm>.<T>

```
if !HaveSME2() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer dn = UInt(Zdn:'0');
integer m = UInt('0':Zm);
constant integer nreg = 2;
```

Four registers
(FEAT_SME2)



ADD { <Zdn1>.<T>-<Zdn4>.<T> }, { <Zdn1>.<T>-<Zdn4>.<T> }, <Zm>.<T>

```
if !HaveSME2() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer dn = UInt(Zdn:'00');
integer m = UInt('0':Zm);
constant integer nreg = 4;
```

Assembler Symbols

<Zdn1> For the two registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zdn" times 2.

For the four registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zdn" times 4.

<T>

Is the size specifier, encoded in “size”:

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

<Zdn4> Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zdn" times 4 plus 3.

<Zdn2> Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zdn" times 2 plus 1.

<Zm> Is the name of the second source scalable vector register Z0-Z15, encoded in the "Zm" field.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV esize;
array [0..3] of bits(VL) results;

for r = 0 to nreg-1
    bits(VL) operand1 = Z[dn+r, VL];
    bits(VL) operand2 = Z[m, VL];
    for e = 0 to elements-1
        bits(esize) element1 = Elem[operand1, e, esize];
        bits(esize) element2 = Elem[operand2, e, esize];
        Elem[results[r], e, esize] = element1 + element2;

for r = 0 to nreg-1
    Z[dn+r, VL] = results[r];
```

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.