

ZIP (two registers)

Interleave elements from two vectors

Place the two-way interleaved elements from the first and second source vectors in the corresponding elements of the two destination vectors.

This instruction is unpredicated.

It has encodings from 2 classes: [8-bit to 64-bit elements](#) and [128-bit element](#)

8-bit to 64-bit elements (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	size	1		Zm					1	1	0	1	0	0			Zn				Zd		0	

ZIP { **<Zd1>.<T>-<Zd2>.<T>** }, **<Zn>.<T>**, **<Zm>.<T>**

```

if !HaveSME2() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer n = UInt(Zn);
integer m = UInt(Zm);
integer d = UInt(Zd:'0');
```

128-bit element (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	0	0	1	Zm			1	1	0	1	0	1	Zn			Zd			0					

ZIP { **<Zd1>.<Q>-<Zd2>.<Q>** }, **<Zn>.<Q>**, **<Zm>.<Q>**

```

if !HaveSME2() then UNDEFINED;
constant integer esize = 128;
integer n = UInt(Zn);
integer m = UInt(Zm);
integer d = UInt(Zd:'0');
```

Assembler Symbols

<Zd1> Is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2.

<T>

Is the size specifier, encoded in "size":

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

<Zd2>

Is the name of the second destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2 plus 1.

<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

```
CheckStreamingSVEEnabled\(\);  
constant integer VL = CurrentVL;  
if VL < esize * 2 then UNDEFINED;  
constant integer pairs = VL DIV (esize * 2);  
bits(VL) operand0 = Z[n, VL];  
bits(VL) operand1 = Z[m, VL];  
bits(VL) result;  
  
for r = 0 to 1  
    integer base = r * pairs;  
    for p = 0 to pairs-1  
        Elem[result, 2*p+0, esize] = Elem[operand0, base+p, esize];  
        Elem[result, 2*p+1, esize] = Elem[operand1, base+p, esize];  
    Z[d+r, VL] = result;
```

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.

[Base
Instructions](#)

[SIMD&FP
Instructions](#)

[SVE
Instructions](#)

[SME
Instructions](#)

[Index by
Encoding](#)

[Sh
Pseud](#)

