BDEP

Scatter lower bits into positions selected by bitmask

This instruction scatters the lowest-numbered contiguous bits within each element of the first source vector to the bit positions indicated by non-zero bits in the corresponding mask element of the second source vector, preserving their order, and set the bits corresponding to a zero mask bit to zero. This instruction is unpredicated.

ID_AA64ZFR0_EL1.BitPerm indicates whether this instruction is implemented.

This instruction is illegal when executed in Streaming SVE mode, unless FEAT_SME_FA64 is implemented and enabled.

SVE2
(FEAT_SVE_BitPerm)

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 1	size	0	Zm	1 0 1 1	0	1	Zn	Zd

    BDEP <Zd>.<T>, <Zn>.<T>, <Zm>.<T>

```
if !HaveSVE() || !HaveSVE2BitPerm() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer n = UInt(Zn);
integer m = UInt(Zm);
integer d = UInt(Zd);
```

Assembler Symbols

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

```
CheckNonStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV esize;
bits(VL) data = Z[n, VL];
bits(VL) mask = Z[m, VL];
bits(VL) result;

for e = 0 to elements − 1
    Elem[result, e, esize] = BitDeposit(Elem[data, e, esize], Elem[mask
Z[d, 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.