LUTI2 (two registers)

Lookup table read with 2-bit indexes

Copy 8-bit, 16-bit or 32-bit elements from ZT0 to two destination vectors using packed 2-bit indices from a segment of the source vector register. A segment corresponds to a portion of the source vector that is consumed in order to fill the destination vector. The segment is selected by the vector segment index modulo the total number of segments.

This instruction is unpredicated.

It has encodings from 2 classes: Consecutive and Strided

Consecutive

(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	0	1	0	0	0	1	1	i3			1	size		0	0	Zn					Zd				0

LUTI2 { <Zd1>.<T>-<Zd2>.<T> }, ZT0, <Zn>[<index>]

```
if !HaveSME2() then UNDEFINED;
if size == '11' then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer isize = 2;
integer n = UInt(Zn);
integer dstride = 1;
integer d = UInt(Zd:'0');
integer imm = UInt(i3);
constant integer nreg = 2;
```

Strided

(FEAT_SME2p1)

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	0	1	0	0	1	1	1	i3			1	size		0	0	Zn					D	0	Zd		

LUTI2 { <Zd1>.<T>, <Zd2>.<T> }, ZT0, <Zn>[<index>]

```
if !HaveSME2p1() then UNDEFINED;
if size == '10' || size == '11' then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer isize = 2;
integer n = UInt(Zn);
integer dstride = 8;
integer d = UInt(D:'0':Zd);
integer imm = UInt(i3);
constant integer nreg = 2;
```

Assembler Symbols

<Zd1>	For the consecutive variant: is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2. For the strided variant: is the name of the first destination scalable vector register Z0-Z7 or Z16-Z23 of a multi-vector sequence, encoded as "D:'0':Zd".

<T>

For the consecutive variant: is the size specifier, encoded in “size”:

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

For the strided variant: is the size specifier, encoded in “size<0>”:

size<0>	<T>
0	B
1	H

<Zd2>	For the consecutive variant: is the name of the second destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2 plus 1. For the strided variant: is the name of the second destination scalable vector register Z8-Z15 or Z24-Z31 of a multi-vector sequence, encoded as "D:'1':Zd".
<Zn>	Is the name of the source scalable vector register, encoded in the "Zn" field.
<index>	Is the vector segment index, in the range 0 to 7, encoded in the "i3" field.

Operation

```
CheckStreamingSVEEnabled();
CheckSMEZT0Enabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV esize;
integer segments = esize DIV (isize * nreg);
integer segment = imm MOD segments;
bits(VL) indexes = Z[n, VL];
bits(512) table = ZT0[512];

for r = 0 to nreg-1
    integer base = (segment * nreg + r) * elements;
    bits(VL) result;
    for e = 0 to elements-1
        integer index = UInt(Elem[indexes, base+e, isize]);
        Elem[result, e, esize] = Elem[table, index, 32]<esize-1:0>;
    Z[d, VL] = result;
    d = d + dstride;
```

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.

Internal version only: isa v33.64, AdvSIMD v29.12, pseudocode no_diffs_2023_09_RC2, sve v2023-06_rel ; Build timestamp: 2023-09-18T17:56