

## LUTi4 (single)

Lookup table read with 4-bit indexes

Copy 8-bit, 16-bit or 32-bit elements from ZT0 to one destination vector using packed 4-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.

### SME2

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

**LUTi4** <Zd>.<T>, ZT0, <Zn> [<index>]

```
if !HaveSME2() then UNDEFINED;
if size == '11' then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer isize = 4;
integer n = UInt(Zn);
integer d = UInt(Zd);
integer imm = UInt(i3);
constant integer nreg = 1;
```

### 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	RESERVED

<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+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](#)

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

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This  
document is Non-Confidential.