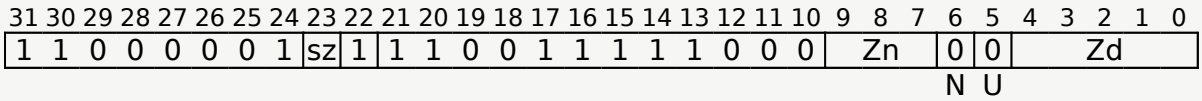
SQCVTU (four registers)

Multi-vector signed saturating unsigned extract narrow

Saturate the signed integer value in each element of the four source vectors to unsigned integer value that is quarter the original source element width, and place the results in the quarter-width destination elements.
This instruction is unpredicated.

SME2
(FEAT_SME2)




```
SQCVTU <Zd>.<T>, { <Zn1>.<Tb>-<Zn4>.<Tb> }
```

```
if !HaveSME2() then UNDEFINED;
constant integer esize = 8 << UInt(sz);
integer n = UInt(Zn:'00');
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 “sz”:

sz	<T>
0	B
1	H

<Zn1> Is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 4.

<Tb> Is the size specifier, encoded in “sz”:

sz	<Tb>
0	S
1	D

<Zn4> Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zn" times 4 plus 3.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV (4 * esize);
bits(VL) result;

for r = 0 to 3
    bits(VL) operand = Z[n+r, VL];
    for e = 0 to elements-1
        integer element = SInt(Elem[operand, e, 4 * esize]);
        Elem[result, r*elements + e, esize] = UnsignedSat(element, esiz

Z[d, VL] = result;
```

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