## UQXTNB

Unsigned saturating extract narrow (bottom)

Saturate the unsigned integer value in each source element to half the original source element width, and place the results in the even-numbered half-width destination elements, while setting the odd-numbered elements to zero.

| 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 0 | tszh | 1 | tszl | 0 0 0 0 1 0 | 0 1 0 | | Zn | Zd |

U T

        UQXTNB **<Zd>.<T>, <Zn>.<Tb>**

```
if !HaveSVE2() && !HaveSME() then UNDEFINED;
bits(3) tsize = tszh:tszl;
integer esize;
case tsize of
    when '001' esize = 16;
    when '010' esize = 32;
    when '100' esize = 64;
    otherwise UNDEFINED;
integer n = UInt(Zn);
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 "tszh:tszl":

| tszh | tszl | <T> |
|---|---|---|
| 0 | 00 | RESERVED |
| 0 | 01 | B |
| 0 | 10 | H |
| x | 11 | RESERVED |
| 1 | 00 | S |
| 1 | 01 | RESERVED |
| 1 | 10 | RESERVED |

<Zn>            Is the name of the source scalable vector register, encoded in the "Zn" field.

<Tb>

Is the size specifier, encoded in "tszh:tszl":

| tszh | tszl | <Tb> |
|------|------|----------|
| 0 | 00 | RESERVED |
| 0 | 01 | H |
| 0 | 10 | S |
| x | 11 | RESERVED |
| 1 | 00 | D |
| 1 | 01 | RESERVED |
| 1 | 10 | RESERVED |

**Operation**

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
bits(VL) operand1 = Z[n, VL];
bits(VL) result;
constant integer halfesize = esize DIV 2;

for e = 0 to elements-1
    integer element1 = UInt(Elem[operand1, e, esize]);
    bits(halfesize) res = UnsignedSat(element1, halfesize);
    Elem[result, 2*e + 0, halfesize] = res;
    Elem[result, 2*e + 1, halfesize] = Zeros(halfesize);

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