

FCVTZU

Multi-vector floating-point convert to unsigned integer, rounding toward zero

Convert to the unsigned 32-bit integer nearer to zero from single-precision, each element of the two or four source vectors, and place the results in the corresponding elements of the two or four destination vectors.

This instruction follows SME2 floating-point numerical behaviors corresponding to instructions that place their results in one or more SVE Z vectors.

This instruction is unpredicated.

It has encodings from 2 classes: [Two registers](#) and [Four registers](#)

Two registers

(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	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	Zn		1		Zd		0			
																										U					

FCVTZU { [<Zd1>.S](#)–[<Zd2>.S](#) }, { [<Zn1>.S](#)–[<Zn2>.S](#) }

```
if !HaveSME2() then UNDEFINED;
integer n = UInt(Zn:'0');
integer d = UInt(Zd:'0');
constant integer nreg = 2;
boolean unsigned = TRUE;
FPRounding rounding = FPRounding\_ZERO;
```

Four registers

(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	1	0	0	1	1	0	0	0	1	1	1	1	0	0	0	Zn		0	1	Zd		0	0		
																										U					

FCVTZU { [<Zd1>.S](#)–[<Zd4>.S](#) }, { [<Zn1>.S](#)–[<Zn4>.S](#) }

```
if !HaveSME2() then UNDEFINED;
integer n = UInt(Zn:'00');
integer d = UInt(Zd:'00');
constant integer nreg = 4;
boolean unsigned = TRUE;
FPRounding rounding = FPRounding\_ZERO;
```

Assembler Symbols

<Zd1>	For the two registers variant: is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2. For the four registers variant: is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 4.
<Zd4>	Is the name of the fourth destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 4 plus 3.
<Zd2>	Is the name of the second destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2 plus 1.
<Zn1>	For the two registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 2. For the four registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 4.
<Zn4>	Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zn" times 4 plus 3.
<Zn2>	Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zn" times 2 plus 1.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV 32;
array [0..3] of bits(VL) results;

for r = 0 to nreg-1
    bits(VL) operand = Z[n+r, VL];
    for e = 0 to elements-1
        bits(32) element = Elem[operand, e, 32];
        Elem[results[r], e, 32] = FPToFixed(element, 0, unsigned, FPCR[

for r = 0 to nreg-1
    Z[d+r, VL] = results[r];
```

[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.