

FCVTZS (scalar, fixed-point)

Floating-point Convert to Signed fixed-point, rounding toward Zero (scalar). This instruction converts the floating-point value in the SIMD&FP source register to a 32-bit or 64-bit fixed-point signed integer using the Round towards Zero rounding mode, and writes the result to the general-purpose destination register.

A floating-point exception can be generated by this instruction. Depending on the settings in [FPCR](#), the exception results in either a flag being set in [FPSR](#), or a synchronous exception being generated. For more information, see [Floating-point exception traps](#).

Depending on the settings in the [CPACR_EL1](#), [CPTR_EL2](#), and [CPTR_EL3](#) registers, and the Security state and Exception level in which the instruction is executed, an attempt to execute the instruction might be trapped.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|-------|---|--|--|----|--|--|--|----|--|--|--|
| 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 | | | | | | | | | | |
| sf | | 0 | | 0 | | 1 | | 1 | | 1 | | 1 | | 0 | | ftype | | 0 | | 1 | | 1 | | 0 | | 0 | | 0 | | scale | | | | Rn | | | | Rd | | | |
| rmode | | | | | | | | | | | | | | | | opcode | | | | | | | | | | | | | | | | | | | | | | | | | |

Half-precision to 32-bit (sf == 0 && ftype == 11) (FEAT_FP16)

```
FCVTZS <Wd>, <Hn>, #<fbits>
```

Half-precision to 64-bit (sf == 1 && ftype == 11) (FEAT_FP16)

```
FCVTZS <Xd>, <Hn>, #<fbits>
```

Single-precision to 32-bit (sf == 0 && ftype == 00)

```
FCVTZS <Wd>, <Sn>, #<fbits>
```

Single-precision to 64-bit (sf == 1 && ftype == 00)

```
FCVTZS <Xd>, <Sn>, #<fbits>
```

Double-precision to 32-bit (sf == 0 && ftype == 01)

```
FCVTZS <Wd>, <Dn>, #<fbits>
```

Double-precision to 64-bit (sf == 1 && ftype == 01)

```
FCVTZS <Xd>, <Dn>, #<fbits>
```

```

if ftype == '10' || (ftype == '11' && !IsFeatureImplemented(FEAT_FP16))
integer d = UInt(Rd);
integer n = UInt(Rn);

constant integer intsize = 32 << UInt(sf);
constant integer decode_ftsize = 8 << UInt(ftype EOR '10');

if sf == '0' && scale<5> == '0' then UNDEFINED;
integer fracbits = 64 - UInt(scale);

```

Assembler Symbols

| | |
|---------|---|
| <Wd> | Is the 32-bit name of the general-purpose destination register, encoded in the "Rd" field. |
| <Xd> | Is the 64-bit name of the general-purpose destination register, encoded in the "Rd" field. |
| <Sn> | Is the 32-bit name of the SIMD&FP source register, encoded in the "Rn" field. |
| <Hn> | Is the 16-bit name of the SIMD&FP source register, encoded in the "Rn" field. |
| <Dn> | Is the 64-bit name of the SIMD&FP source register, encoded in the "Rn" field. |
| <fbits> | <p>For the double-precision to 32-bit, half-precision to 32-bit and single-precision to 32-bit variant: is the number of bits after the binary point in the fixed-point destination, in the range 1 to 32, encoded as 64 minus "scale".</p> <p>For the double-precision to 64-bit, half-precision to 64-bit and single-precision to 64-bit variant: is the number of bits after the binary point in the fixed-point destination, in the range 1 to 64, encoded as 64 minus "scale".</p> |

Operation

```

CheckFPEnabled64();

FPCRType fpcr = FPCR[];
bits(decode_ftsize) fltval;
bits(intsize) intval;

fltval = V[n, decode_ftsize];
intval = FPToFixed(fltval, fracbits, FALSE, fpcr, FPRounding\_ZERO, intsize);
X[d, intsize] = intval;

```

Operational information

If FEAT_SME is implemented and the PE is in Streaming SVE mode, then any subsequent instruction which is dependent on the general-purpose register written by this instruction might be significantly delayed.

[Base
Instructions](#)

[SIMD&FP
Instructions](#)

[SVE
Instructions](#)

[SME
Instructions](#)

[Index by
Encoding](#)

[Sh
Pseu](#)

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.