

## FTMAD

Floating-point trigonometric multiply-add coefficient

The FTMAD instruction calculates the series terms for either  $\sin(x)$  or  $\cos(x)$ , where the argument  $x$  has been adjusted to be in the range  $-\pi/4 < x \leq \pi/4$ .

To calculate the series terms of  $\sin(x)$  and  $\cos(x)$  the initial source operands of FTMAD should be zero in the first source vector and  $x^2$  in the second source vector. The FTMAD instruction is then executed eight times to calculate the sum of eight series terms, which gives a result of sufficient precision.

The FTMAD instruction multiplies each element of the first source vector by the absolute value of the corresponding element of the second source vector and performs a fused addition of each product with a value obtained from a table of hard-wired coefficients, and places the results destructively in the first source vector.

The coefficients are different for  $\sin(x)$  and  $\cos(x)$ , and are selected by a combination of the sign bit in the second source element and an immediate index in the range 0 to 7.

Double-precision coefficient table for  $\sin(x)$  ( $s2<63> == '0'$ )

Index	Hexadecimal	Decimal	Exact Value
0	3ff0 0000 0000 0000	1.0	$= 1/1!$
1	bfc5 5555 5555 5543	-0.16666666666666661	$> -1/3!$
2	3f81 1111 1110 f30c	0.83333333333320002e-02	$< 1/5!$
3	bf2a 01a0 19b9 2fc6	-0.1984126982840213e-03	$> -1/7!$
4	3ec7 1de3 51f3 d22b	0.2755731329901505e-05	$< 1/9!$
5	be5a e5e2 b60f 7b91	-0.2505070584637887e-07	$> -1/11!$
6	3de5 d840 8868 552f	0.1589413637195215e-09	$< 1/13!$
7	0000 0000 0000 0000	0.0	$> -1/15!$

Double-precision coefficient table for  $\cos(x)$  ( $s2<63> == '1'$ )

Index	Hexadecimal	Decimal	Exact Value
0	3ff0 0000 0000 0000	1.0	$= 1/0!$

Index	Hexadecimal	Decimal	Exact Value
1	bfe0 0000 0000 0000	-0.5000000000000000	= -1/2!
2	3fa5 5555 5555 5536	0.41666666666666645e-01	< 1/4!
3	bf56 c16c 16c1 3a0b	-0.13888888888886111e-02	> -1/6!
4	3efa 01a0 19b1 e8d8	0.2480158728388683e-04	< 1/8!
5	be92 7e4f 7282 f468	-0.2755731309913950e-06	> -1/10!
6	3e21 ee96 d264 1b13	0.2087558253975872e-08	< 1/12!
7	bda8 f763 80fb b401	-0.1135338700720054e-10	> -1/14!

Single-precision coefficient table for sin(x) (s2<31> == '0')

Index	Hexadecimal	Decimal	Exact Value
0	3f80 0000	1.0	= 1/1!
1	be2a aaab	-1.666666716337e-01	> -1/3!
2	3c08 8886	8.333330973983e-03	< 1/5!
3	b950 08b9	-1.983967522392e-04	> -1/7!
4	3636 9d6d	2.721174723774e-06	< 1/9!
5	0000 0000	0.0	> -1/11!
6	0000 0000	0.0	< 1/13!
7	0000 0000	0.0	> -1/15!

Single-precision coefficient table for cos(x) (s2<31> == '1')

Index	Hexadecimal	Decimal	Exact Value
0	3f80 0000	1.0	= 1/0!
1	bf00 0000	-5.000000000000e-01	= -1/2!
2	3d2a aaa6	4.166664928198e-02	< 1/4!
3	bab6 0705	-1.388759003021e-03	> -1/6!
4	37cd 37cc	2.446388680255e-05	< 1/8!
5	0000 0000	0.0	> -1/10!
6	0000 0000	0.0	< 1/12!
7	0000 0000	0.0	> -1/14!

Half-precision coefficient table for sin(x) (s2<15> == '0')

Index	Hexadecimal	Decimal	Exact Value
0	3c00	1.0	= 1/1!
1	b155	-1.666666716337e-01	> -1/3!
2	2030	8.333330973983e-03	< 1/5!
3	0000	0.0	> -1/7!
4	0000	0.0	< 1/9!
5	0000	0.0	> -1/11!
6	0000	0.0	< 1/13!

Index	Hexadecimal	Decimal	Exact Value
7	0000	0.0	> -1/15!

Half-precision coefficient table for cos(x) (s2<15> == '1')

Index	Hexadecimal	Decimal	Exact Value
0	3c00	1.0	= 1/0!
1	b800	-5.00000000000000e-01	= -1/2!
2	293a	4.166664928198e-02	< 1/4!
3	0000	0.0	> -1/6!
4	0000	0.0	< 1/8!
5	0000	0.0	> -1/10!
6	0000	0.0	< 1/12!
7	0000	0.0	> -1/14!

This instruction is illegal when executed in Streaming SVE mode, unless FEAT\_SME\_FA64 is implemented and enabled.

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	1	0	0	1	0	1	size	0	1	0	imm3	1	0	0	0	0	0	0						Zm					Zdn	

**FTMAD** <Zdn>.<T>, <Zdn>.<T>, <Zm>.<T>, #<imm>

```
if !HaveSVE() then UNDEFINED;
if size == '00' then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer dn = UInt(Zdn);
integer m = UInt(Zm);
integer imm = UInt(imm3);
```

## Assembler Symbols

<Zdn> Is the name of the first source and destination scalable vector register, encoded in the "Zdn" field.

<T> Is the size specifier, encoded in "size":

size	<T>
00	RESERVED
01	H
10	S
11	D

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

<imm> Is the unsigned immediate operand, in the range 0 to 7, encoded in the "imm3" field.

## Operation

```
CheckNonStreamingSVEEnabled();
constant integer VL = CurrentVL;
```

```

constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
bits(VL) operand1 = Z[dn, VL];
bits(VL) operand2 = Z[m, VL];
bits(VL) result;

for e = 0 to elements-1
    bits(esize) element1 = Elem[operand1, e, esize];
    bits(esize) element2 = Elem[operand2, e, esize];
    Elem[result, e, esize] = FPTrigMAdd(imm, element1, element2, FPCR[])

Z[dn, VL] = result;

```

## Operational information

This instruction might be immediately preceded in program order by a MOVPRFX instruction. The MOVPRFX instruction must conform to all of the following requirements, otherwise the behavior of the MOVPRFX and this instruction is unpredictable:

- The MOVPRFX instruction must be unpredicated.
- The MOVPRFX instruction must specify the same destination register as this instruction.
- The destination register must not refer to architectural register state referenced by any other source operand register of this instruction.

[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.