

## USMMLA (vector)

Unsigned and signed 8-bit integer matrix multiply-accumulate. This instruction multiplies the 2x8 matrix of unsigned 8-bit integer values in the first source vector by the 8x2 matrix of signed 8-bit integer values in the second source vector. The resulting 2x2 32-bit integer matrix product is destructively added to the 32-bit integer matrix accumulator in the destination vector. This is equivalent to performing an 8-way dot product per destination element.

From Armv8.2 to Armv8.5, this is an optional instruction. From Armv8.6 it is mandatory for implementations that include Advanced SIMD to support it. [ID\\_AA64ISAR1\\_EL1](#).I8MM indicates whether this instruction is supported.

### Vector (FEAT\_I8MM)

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	1	1	1	0	1	0	0	Rm				1	0	1	0	1	1	Rn				Rd						
U										B																					

**USMMLA** <Vd>.4S, <Vn>.16B, <Vm>.16B

```
if !IsFeatureImplemented(FEAT_I8MM) then UNDEFINED;
integer n = UInt(Rn);
integer m = UInt(Rm);
integer d = UInt(Rd);
```

### Assembler Symbols

- <Vd> Is the name of the SIMD&FP third source and destination register, encoded in the "Rd" field.
- <Vn> Is the name of the first SIMD&FP source register, encoded in the "Rn" field.
- <Vm> Is the name of the second SIMD&FP source register, encoded in the "Rm" field.

### Operation

```
CheckFPAdvSIMDEnabled64();
bits(128) operand1 = V[n, 128];
bits(128) operand2 = V[m, 128];
bits(128) addend = V[d, 128];

V[d, 128] = MatMulAdd(addend, operand1, operand2, TRUE, FALSE);
```

**Operational information**

Arm expects that the USMMLA (vector) instruction will deliver a peak integer multiply throughput that is at least as high as can be achieved using two USDOT (vector) instructions, with a goal that it should have significantly higher throughput.

[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.