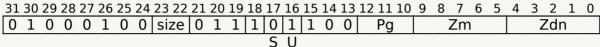
y	<u>Sh</u>
<u>ıg</u>	<u>Pseu</u>

<u>Base</u>	SIMD&FP	<u>SVE</u>	<u>SME</u>	Index by
<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	<u>Instructions</u>	Encoding

USQADD

Unsigned saturating addition of signed value

Add active signed elements of the source vector to the corresponding unsigned elements of the addend vector, and destructively place the results in the corresponding elements of the addend vector. Each result element is saturated to the N-bit element's unsigned integer range 0 to (2^N) -1. Inactive elements in the destination vector register remain unmodified.



```
USQADD <Zdn>.<T>, <Pg>/M, <Zdn>.<T>, <Zm>.<T>

if !HaveSVE2() && !HaveSME() then UNDEFINED;
constant integer esize = 8 << UInt(size);
integer g = UInt(Pg);
integer dn = UInt(Zdn);
integer m = UInt(Zm);</pre>
```

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></t>
0.0	В
01	Н
10	S
11	D

<Pg>

Is the name of the governing scalable predicate register P0-P7, encoded in the "Pg" field.

<Zm>

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

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
bits(PL) mask = P[g, PL];
bits(VL) operand1 = Z[dn, VL];
```

```
bits(VL) operand2 = if AnyActiveElement(mask, esize) then Z[m, VL] else
bits(VL) result;

for e = 0 to elements-1
   bits(esize) element1 = Elem[operand1, e, esize];
   bits(esize) element2 = Elem[operand2, e, esize];
   if ActivePredicateElement(mask, e, esize) then
        Elem[result, e, esize] = UnsignedSat(UInt(element1) + SInt(element2));

Elem[result, e, esize] = Elem[operand1, e, esize];
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, or be predicated using the same governing predicate register and source element size as this instruction.
- 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.

BaseSIMD&FPSVESMEIndex byInstructionsInstructionsInstructionsInstructions

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 © 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.

Sh Pseu