

## SHRN, SHRN2

Shift Right Narrow (immediate). This instruction reads each unsigned integer value from the source SIMD&FP register, right shifts each result by an immediate value, puts the final result into a vector, and writes the vector to the lower or upper half of the destination SIMD&FP register. The destination vector elements are half as long as the source vector elements. The results are truncated. For rounded results, see [RSHRN](#).

The [RSHRN](#) instruction writes the vector to the lower half of the destination register and clears the upper half, while the [RSHRN2](#) instruction writes the vector to the upper half of the destination register without affecting the other bits of the register.

Depending on the settings in the [CPACR\\_EL1](#), [CPTR\\_EL2](#), and [CPTR\\_EL3](#) registers, and the current Security state and Exception level, 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
0	Q	0	0	1	1	1	1	0	!= 0000				immb		1 0 0 0		0	1	Rn						Rd						
immh												op																			

**SHRN{2} <Vd>.<Tb>, <Vn>.<Ta>, #<shift>**

```
integer d = UInt(Rd);
integer n = UInt(Rn);

if immh == '0000' then SEE(asimdimm);
if immh<3> == '1' then UNDEFINED;
constant integer esize = 8 << HighestSetBit(immh);
constant integer datasize = 64;
integer part = UInt(Q);
integer elements = datasize DIV esize;

integer shift = (2 * esize) - UInt(immh:immb);
boolean round = (op == '1');
```

### Assembler Symbols

2

Is the second and upper half specifier. If present it causes the operation to be performed on the upper 64 bits of the registers holding the narrower elements, and is encoded in “Q”:

Q	2
0	[absent]
1	[present]

<Vd>

Is the name of the SIMD&FP destination register, encoded in the “Rd” field.

<Tb>

Is an arrangement specifier, encoded in “immh:Q”:

immh	Q	<Tb>
0000	x	<a href="#">SEE Advanced SIMD modified immediate</a>
0001	0	8B
0001	1	16B
001x	0	4H
001x	1	8H
01xx	0	2S
01xx	1	4S
1xxx	x	RESERVED

<Vn>

Is the name of the SIMD&FP source register, encoded in the “Rn” field.

<Ta>

Is an arrangement specifier, encoded in “immh”:

immh	<Ta>
0000	<a href="#">SEE Advanced SIMD modified immediate</a>
0001	8H
001x	4S
01xx	2D
1xxx	RESERVED

<shift>

Is the right shift amount, in the range 1 to the destination element width in bits, encoded in “immh:immb”:

immh	<shift>
0000	<a href="#">SEE Advanced SIMD modified immediate</a>
0001	(16-UInt (immh:immb))
001x	(32-UInt (immh:immb))
01xx	(64-UInt (immh:immb))
1xxx	RESERVED

## Operation

```
CheckFPAdvSIMDEnabled64();
bits(datasize*2) operand = V[n, datasize*2];
bits(datasize) result;
integer element;

for e = 0 to elements-1
    element = RShr(UInt(Elem[operand, e, 2*esize]), shift, round);
    Elem[result, e, esize] = element<esize-1:0>;

Vpart[d, part, datasize] = result;
```

**Operational information**

If PSTATE.DIT is 1:

- The execution time of this instruction is independent of:
  - The values of the data supplied in any of its registers.
  - The values of the NZCV flags.
- The response of this instruction to asynchronous exceptions does not vary based on:
  - The values of the data supplied in any of its registers.
  - The values of the NZCV flags.

---

<a href="#">Base Instructions</a>	<a href="#">SIMD&amp;FP Instructions</a>	<a href="#">SVE Instructions</a>	<a href="#">SME Instructions</a>	<a href="#">Index by Encoding</a>
-----------------------------------	--	----------------------------------	----------------------------------	-----------------------------------

[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.