<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

Pseu

LDAPRH

Load-Acquire RCpc Register Halfword derives an address from a base register value, loads a halfword from the derived address in memory, zero-extends it and writes it to a register.

The instruction has memory ordering semantics as described in *Load-Acquire*, *Load-AcquirePC*, and *Store-Release*, except that:

- There is no ordering requirement, separate from the requirements of a Load-AcquirePC or a Store-Release, created by having a Store-Release followed by a Load-AcquirePC instruction.
- The reading of a value written by a Store-Release by a Load-AcquirePC instruction by the same observer does not make the write of the Store-Release globally observed.

This difference in memory ordering is not described in the pseudocode. For information about memory accesses, see *Load/Store addressing modes*.

Integer (FEAT LRCPC)

```
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 1 1 1 0 0 0 1 0 1 (1)(1)(1)(1)(1) 1 1 0 0 0 0 Rn Rt

size

Rs
```

```
LDAPRH <Wt>, [<Xn | SP> {,#0}]
integer n = UInt(Rn);
integer t = UInt(Rt);
boolean tagchecked = n != 31;
```

Assembler Symbols

<Wt> Is the 32-bit name of the general-purpose register to be

loaded, encoded in the "Rt" field.

<Xn|SP> Is the 64-bit name of the general-purpose base register or

stack pointer, encoded in the "Rn" field.

Operation

```
else
    address = X[n, 64];

data = Mem[address, 2, accdesc];
X[t, 32] = ZeroExtend(data, 32);
```

Operational information

If PSTATE.DIT is 1, the timing of this instruction is insensitive to the value of the data being loaded or stored.

<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

Internal version only: is a 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