LD1RQH (scalar plus immediate)

Contiguous load and replicate eight halfwords (immediate index)

Load eight contiguous halfwords to elements of a short, 128-bit (quadword) vector from the memory address generated by a 64-bit scalar base address and immediate index that is a multiple of 16 in the range -128 to +112 added to the base address.

Inactive elements will not cause a read from Device memory or signal a fault, and are set to zero. The resulting short vector is then replicated to fill the long destination vector. Only the first eight predicate elements are used and higher numbered predicate elements are ignored.

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
1 0 1 0 0 1 0	0	1	0 0	0	imm4	0 0 1	Pg	Rn	Zt
	msz<1>	msz<0>	ssz						

```
LD1RQH { <Zt>.H }, <Pg>/Z, [<Xn|SP>{, #<imm>}]
```

```
if !HaveSVE() && !HaveSME() then UNDEFINED;
integer t = UInt(Zt);
integer n = UInt(Rn);
integer g = UInt(Pg);
constant integer esize = 16;
integer offset = SInt(imm4);
```

Assembler Symbols

<Zt>	Is the name of the scalable vector register to be transferred, encoded in the "Zt" field.
<Pg>	Is the name of the governing scalable predicate register P0-P7, encoded in the "Pg" field.
<Xn|SP>	Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field.
<imm>	Is the optional signed immediate byte offset, a multiple of 16 in the range -128 to 112, defaulting to 0, encoded in the "imm4" field.

Operation

```
CheckSVEEnabled();
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = 128 DIV esize;
bits(64) base;
bits(PL) mask = P[g, PL]; // low 16 bits only
bits(128) result;
constant integer mbytes = esize DIV 8;
```

```
    boolean contiguous = TRUE;
    boolean nontemporal = FALSE;
    boolean tagchecked = n != 31;
    AccessDescriptor accdesc = CreateAccDescSVE(MemOp_LOAD, nontemporal, co

    if !AnyActiveElement(mask, esize) then
        if n == 31 && ConstrainUnpredictableBool(Unpredictable_CHECKSPNONEA
            CheckSPAlignment();
    else
        if n == 31 then CheckSPAlignment();
        base = if n == 31 then SP[] else X[n, 64];

    for e = 0 to elements-1
        if ActivePredicateElement(mask, e, esize) then
            bits(64) addr = base + (offset * 16) + (e * mbytes);
            Elem[result, e, esize] = Mem[addr, mbytes, accdesc];
        else
            Elem[result, e, esize] = Zeros(esize);

    Z[t, VL] = Replicate(result, VL DIV 128);
```

Operational information

If FEAT_SVE2 is implemented or FEAT_SME is implemented, then if PSTATE.DIT is 1, the timing of this instruction is insensitive to the value of the data being loaded or stored when its governing predicate register contains the same value for each execution.

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