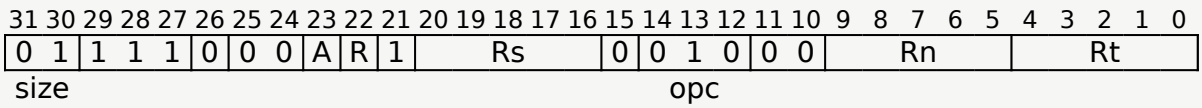
## LDEORH, LDEORAH, LDEORALH, LDEORLH

Atomic Exclusive-OR on halfword in memory atomically loads a 16-bit halfword from memory, performs an exclusive-OR with the value held in a register on it, and stores the result back to memory. The value initially loaded from memory is returned in the destination register.

- If the destination register is not WZR, `LDEORAH` and `LDEORALH` load from memory with acquire semantics.
- `LDEORLH` and `LDEORALH` store to memory with release semantics.
- `LDEORH` has neither acquire nor release semantics.

For more information about memory ordering semantics, see *Load-Acquire, Store-Release*.
For information about memory accesses, see *Load/Store addressing modes*.
This instruction is used by the alias STEORH, STEORLH.

**Integer**
**(FEAT_LSE)**




**LDEORAH (A == 1 && R == 0)**

```
LDEORAH <Ws>, <Wt>, [<Xn|SP>]
```

**LDEORALH (A == 1 && R == 1)**

```
LDEORALH <Ws>, <Wt>, [<Xn|SP>]
```

**LDEORH (A == 0 && R == 0)**

```
LDEORH <Ws>, <Wt>, [<Xn|SP>]
```

**LDEORLH (A == 0 && R == 1)**

```
LDEORLH <Ws>, <Wt>, [<Xn|SP>]
```

```
if !IsFeatureImplemented(FEAT_LSE) then UNDEFINED;

integer t = UInt(Rt);
integer n = UInt(Rn);
integer s = UInt(Rs);

boolean acquire = A == '1' && Rt != '11111';
boolean release = R == '1';
boolean tagchecked = n != 31;
```

### Assembler Symbols

| | |
|---|---|
| <Ws> | Is the 32-bit name of the general-purpose register holding the data value to be operated on with the contents of the memory location, encoded in the "Rs" field. |
| <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. |

### Alias Conditions

| Alias | Is preferred when |
|---|---|
| STEORH, STEORLH | `A == '0' && Rt == '11111'` |

### Operation

```
bits(64) address;
bits(16) value;
bits(16) data;

AccessDescriptor accdesc = CreateAccDescAtomicOp(MemAtomicOp_EOR, acqui

value = X[s, 16];
if n == 31 then
    CheckSPAlignment();
    address = SP[];
else
    address = X[n, 64];

bits(16) comparevalue = bits(16) UNKNOWN;    // Irrelevant when not exe
data = MemAtomic(address, comparevalue, value, accdesc);

if t != 31 then
    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.

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