STADDH, STADDLH

Atomic add on halfword in memory, without return, atomically loads a 16-bit halfword from memory, adds the value held in a register to it, and stores the result back to memory.

- STADDH does not have release semantics.
- STADDLH stores to memory with release semantics, as described in *Load-Acquire, Store-Release*.

For information about memory accesses, see *Load/Store addressing modes*.

This is an alias of LDADDH, LDADDAH, LDADDALH, LDADDLH. This means:

- The encodings in this description are named to match the encodings of LDADDH, LDADDAH, LDADDALH, LDADDLH.
- The description of LDADDH, LDADDAH, LDADDALH, LDADDLH gives the operational pseudocode, any constrained unpredictable behavior, and any operational information for this instruction.

Integer
(FEAT_LSE)

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	0	0	0	0	R	1			Rs			0	0	0	0	0	0			Rn			1	1	1	1	1
size							A											opc									Rt				

No memory ordering (R == 0)

```
        STADDH <Ws>, [<Xn|SP>]
```

is equivalent to

```
        LDADDH <Ws>, WZR, [<Xn|SP>]
```

and is always the preferred disassembly.

Release (R == 1)

```
        STADDLH <Ws>, [<Xn|SP>]
```

is equivalent to

```
        LDADDLH <Ws>, WZR, [<Xn|SP>]
```

and is always the preferred disassembly.

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.

<Xn|SP>   Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field.

Operation

The description of [LDADDH, LDADDAH, LDADDALH, LDADDLH](#) gives the operational pseudocode for this instruction.

Operational information

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