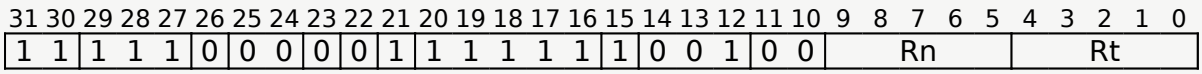
## ST64B

Single-copy Atomic 64-byte Store without Return stores eight 64-bit doublewords from consecutive registers, Xt to X(t+7), to a memory location. The data that is stored is atomic and is required to be 64-byte-aligned.

### Integer
**(FEAT_LS64)**




```
ST64B <Xt>, [<Xn|SP> {,#0}]
```

```
if !IsFeatureImplemented(FEAT_LS64) then UNDEFINED;
if Rt<4:3> == '11' || Rt<0> == '1' then UNDEFINED;

integer n = UInt(Rn);
integer t = UInt(Rt);
MemOp memop = MemOp_STORE;
boolean tagchecked = n != 31;
```

### Assembler Symbols

| | |
|---|---|
| <Xt> | Is the 64-bit name of the first general-purpose register to be transferred, 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

```
CheckLDST64BEnabled();

bits(512) data;
bits(64) address;
bits(64) value;

AccessDescriptor accdesc = CreateAccDescLS64(memop, tagchecked);
for i = 0 to 7
    value = X[t+i, 64];
    if BigEndian(accdesc.acctype) then value = BigEndianReverse(value);
    data<63+64*i:64*i> = value;

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

MemStore64B(address, data, accdesc);
```

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.