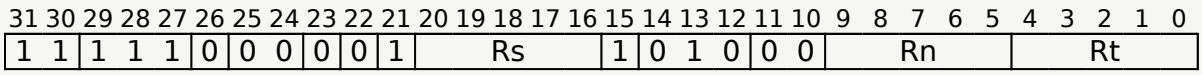
ST64BV0

Single-copy Atomic 64-byte EL0 Store with Return stores eight 64-bit doublewords from consecutive registers, Xt to X(t+7), to a memory location, with the bottom 32 bits taken from *ACCDATA_EL1*, and writes the status result of the store to a register. The data that is stored is atomic and is required to be 64-byte aligned.

Integer
(FEAT_LS64_ACCDATA)




```
ST64BV0 <Xs>, <Xt>, [<Xn|SP>]

if !IsFeatureImplemented(FEAT_LS64_ACCDATA) then UNDEFINED;
if Rt<4:3> == '11' || Rt<0> == '1' then UNDEFINED;

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

Assembler Symbols

<Xs>
: Is the 64-bit name of the general-purpose register into which the status result of this instruction is written, encoded in the "Rs" field.

  The value returned is:

  0xFFFFFFFF_FFFFFFFF
  : If the memory location accessed does not support this instruction. In this case, the value at the memory location is UNKNOWN.

  != 0xFFFFFFFF_FFFFFFFF
  : If the memory location accessed does support this instruction. In this case, the peripheral that provides the response defines the returned value and provides information on the state of the memory update at the memory location.

  If XZR is used, then the return value is ignored.

<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

```
CheckST64BV0Enabled();

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

AccessDescriptor accdesc = CreateAccDescLS64(memop, tagchecked);
bits(64) Xt = X[t, 64];
value<31:0> = ACCDATA_EL1<31:0>;
value<63:32> = Xt<63:32>;
if BigEndian(accdesc.acctype) then value = BigEndianReverse(value);
data<63:0> = value;
for i = 1 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];

status = MemStore64BWithRet(address, data, accdesc);

if s != 31 then X[s, 64] = status;
```


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