## ST1D (scalar plus immediate, consecutive registers)

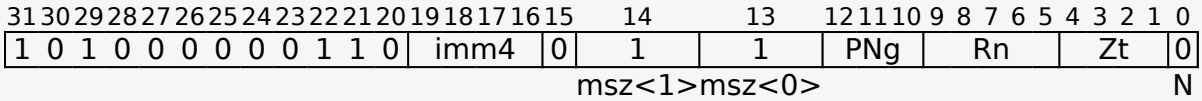
Contiguous store of doublewords from multiple consecutive vectors (immediate index)

Contiguous store of doublewords from elements of two or four consecutive vector registers to the memory address generated by a 64-bit scalar base and immediate index which is multiplied by the vector's in-memory size, irrespective of predication, and added to the base address.

Inactive elements are not written to memory.

It has encodings from 2 classes: Two registers and Four registers

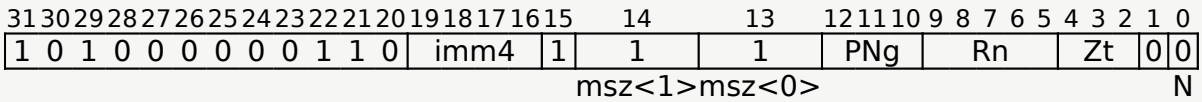
### Two registers
**(FEAT_SVE2p1)**




**ST1D { <Zt1>.D-<Zt2>.D }, <PNg>, [<Xn|SP>{, #<imm>, MUL VL}]**

```
if !HaveSME2() && !HaveSVE2p1() then UNDEFINED;
integer n = UInt(Rn);
integer g = UInt('1':PNg);
constant integer nreg = 2;
integer t = UInt(Zt:'0');
constant integer esize = 64;
integer offset = SInt(imm4);
```

### Four registers
**(FEAT_SVE2p1)**




**ST1D { <Zt1>.D-<Zt4>.D }, <PNg>, [<Xn|SP>{, #<imm>, MUL VL}]**

```
if !HaveSME2() && !HaveSVE2p1() then UNDEFINED;
integer n = UInt(Rn);
integer g = UInt('1':PNg);
constant integer nreg = 4;
integer t = UInt(Zt:'00');
constant integer esize = 64;
integer offset = SInt(imm4);
```

**Assembler Symbols**

| | |
|---|---|
| <Zt1> | For the two registers variant: is the name of the first scalable vector register to be transferred, encoded as "Zt" times 2.<br><br>For the four registers variant: is the name of the first scalable vector register to be transferred, encoded as "Zt" times 4. |
| <Zt4> | Is the name of the fourth scalable vector register to be transferred, encoded as "Zt" times 4 plus 3. |
| <Zt2> | Is the name of the second scalable vector register to be transferred, encoded as "Zt" times 2 plus 1. |
| <PNg> | Is the name of the governing scalable predicate register PN8-PN15, with predicate-as-counter encoding, encoded in the "PNg" field. |
| <Xn\|SP> | Is the 64-bit name of the general-purpose base register or stack pointer, encoded in the "Rn" field. |
| <imm> | For the two registers variant: is the optional signed immediate vector offset, a multiple of 2 in the range -16 to 14, defaulting to 0, encoded in the "imm4" field.<br><br>For the four registers variant: is the optional signed immediate vector offset, a multiple of 4 in the range -32 to 28, defaulting to 0, encoded in the "imm4" field. |

**Operation**

```
if HaveSVE2p1() then CheckSVEEnabled(); else CheckStreamingSVEEnabled()
constant integer VL = CurrentVL;
constant integer PL = VL DIV 8;
constant integer elements = VL DIV esize;
constant integer mbytes = esize DIV 8;
bits(64) base;
bits(VL) src;
bits(PL) pred = P[g, PL];
bits(PL * nreg) mask = CounterToPredicate(pred<15:0>, PL * nreg);
boolean contiguous = TRUE;
boolean nontemporal = FALSE;
boolean tagchecked = n != 31;
AccessDescriptor accdesc = CreateAccDescSVE(MemOp_STORE, nontemporal, c

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 r = 0 to nreg-1
    src = Z[t+r, VL];
    for e = 0 to elements-1
```

```
        if ActivePredicateElement(mask, r * elements + e, esize) then
            bits(64) addr = base + (offset * nreg * elements + r * eleme
            Mem[addr, mbytes, accdesc] = Elem[src, e, esize];
```

**Operational information**

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