

## AT S1E1W, Address Translate Stage 1 EL1 Write

The AT S1E1W characteristics are:

### Purpose

Performs stage 1 address translation, with permissions as if writing to the given virtual address from EL1, or from EL2 if the Effective value of [HCR\\_EL2](#).{E2H, TGE} is {1, 1}, using the following translation regime:

- When EL2 is implemented and enabled in the Security state described by the current Effective value of [SCR\\_EL3](#).{NSE, NS}:
  - If [HCR\\_EL2](#).{E2H, TGE} is not {1, 1}, the EL1&0 translation regime, accessed from EL1.
  - If [HCR\\_EL2](#).{E2H, TGE} is {1, 1}, the EL2&0 translation regime, accessed from EL2.
- Otherwise, the EL1&0 translation regime, accessed from EL1.

When FEAT\_RME is implemented, if the Effective value of [SCR\\_EL3](#).{NSE, NS} is a reserved value, this instruction is undefined at EL3.

### Configuration

There are no configuration notes.

### Attributes

AT S1E1W is a 64-bit System instruction.

### Field descriptions

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
<a href="#">Input address for translation</a>																															
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

#### Bits [63:0]

Input address for translation. The resulting address can be read from the [PAR\\_EL1](#).

If the address translation instructions are targeting a translation regime that is using AArch32, and so has a VA of only 32 bits, then VA[63:32] is res0.

## Executing AT S1E1W

Accesses to this instruction use the following encodings in the System instruction encoding space:

AT S1E1W, <Xt>

op0	op1	CRn	CRm	op2
0b01	0b000	0b0111	0b1000	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.AT == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGITR_EL2.ATS1E1W == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.AT(X[t, 64], TranslationStage_1,
        EL1, ATAccess_Write);
    elsif PSTATE.EL == EL2 then
        AArch64.AT(X[t, 64], TranslationStage_1, EL1,
        ATAccess_Write);
    elsif PSTATE.EL == EL3 then
        AArch64.AT(X[t, 64], TranslationStage_1, EL1,
        ATAccess_Write);
```

---

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.