AArch32
Registers

AArch64
Registers

AArch32
Instructions

AArch64
Instructions

Index by
Encoding

External
Registers

TLBI ALLE2OS, TLBI ALLE2OSNXS, TLB Invalidate All, EL2, Outer Shareable

The TLBI ALLE2OS, TLBI ALLE2OSNXS characteristics are:

Purpose

If EL2 is implemented and enabled in the current Security state, invalidates cached copies of translation table entries from TLBs that meet all the following requirements:

- The entry is a stage 1 translation table entry, from any level of the translation table walk.

- If FEAT_RME is implemented, one of the following applies:

  - SCR_EL3.{NSE, NS} is {0, 0} and the entry would be required to translate any address using the Secure EL2&0 or EL2 translation regime.
  - SCR_EL3.{NSE, NS} is {0, 1} and the entry would be required to translate any address using the Non-secure EL2&0 or EL2 translation regime.
  - SCR_EL3.{NSE, NS} is {1, 1} and the entry would be required to translate any address using the Realm EL2&0 or EL2 translation regime.

- If FEAT_RME is not implemented, one of the following applies:

  - SCR_EL3.NS is 0 and the entry would be required to translate any address using the Secure EL2&0 or EL2 translation regime.
  - SCR_EL3.NS is 1 and the entry would be required to translate any address using the Non-secure EL2&0 or EL2 translation regime.

The invalidation applies to all PEs in the same Outer Shareable shareability domain as the PE that executes this System instruction.

If FEAT_XS is implemented, the nXS variant of this System instruction is defined.

Both variants perform the same invalidation, but the TLBI System instruction without the nXS qualifier waits for all memory accesses using in-scope old translation information to complete before it is considered complete.

The TLBI System instruction with the nXS qualifier is considered complete when the subset of these memory accesses with XS attribute set to 0 are complete.

Configuration

This instruction is present only when FEAT_TLBIOS is implemented. Otherwise, direct accesses to TLBI ALLE2OS, TLBI ALLE2OSNXS are undefined.

Attributes

TLBI ALLE2OS, TLBI ALLE2OSNXS is a 64-bit System instruction.

Field descriptions

This instruction has no applicable fields.

The value in the register specified by <Xt> is ignored.

Executing TLBI ALLE2OS, TLBI ALLE2OSNXS

The Rt field should be set to `0b11111`. If the Rt field is not set to `0b11111`, it is constrained unpredictable whether:

- The instruction is undefined.

- The instruction behaves as if the Rt field is set to `0b11111`.

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

TLBI ALLE2OS{, <Xt>}

op0	op1	CRn	CRm	op2
0b01	0b100	0b1000	0b0001	0b000

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL20, Shareability_OSH, TLBI_AllAttr);
    else
```

```
              AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL2, Shareability_OSH, TLBI_AllAttr);
elsif PSTATE.EL == EL3 then
    if !EL2Enabled() then
        UNDEFINED;
    elsif HCR_EL2.E2H == '1' then
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL20, Shareability_OSH, TLBI_AllAttr);
    else
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL2, Shareability_OSH, TLBI_AllAttr);
```

TLBI ALLE2OSNXS{, <Xt>}

op0	op1	CRn	CRm	op2
0b01	0b100	0b1001	0b0001	0b000

```
if !IsFeatureImplemented(FEAT_XS) then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL20, Shareability_OSH, TLBI_ExcludeXS);
    else
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL2, Shareability_OSH, TLBI_ExcludeXS);
elsif PSTATE.EL == EL3 then
    if !EL2Enabled() then
        UNDEFINED;
    elsif HCR_EL2.E2H == '1' then
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL20, Shareability_OSH, TLBI_ExcludeXS);
    else
        AArch64.TLBI_ALL(SecurityStateAtEL(EL2),
Regime_EL2, Shareability_OSH, TLBI_ExcludeXS);
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94