

---

## TTBR1\_EL2, Translation Table Base Register 1 (EL2)

The TTBR1\_EL2 characteristics are:

### Purpose

When [HCR\\_EL2.E2H](#) is 1, holds the base address of the translation table for the initial lookup for stage 1 of the translation of an address from the higher VA range in the EL2&0 translation regime, and other information for this translation regime.

---

### Note

When [HCR\\_EL2.E2H](#) is 0, the contents of this register are ignored by the PE, except for a direct read or write of the register.

---

### Configuration

This register is present only when FEAT\_VHE is implemented. Otherwise, direct accesses to TTBR1\_EL2 are undefined.

If EL2 is not implemented, this register is res0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

TTBR1\_EL2 is a 128-bit register that can also be accessed as a 64-bit value. If it is accessed as a 64-bit register, accesses read and write bits [63:0] and do not modify bits [127:64].

### Attributes

TTBR1\_EL2 is a:

- 128-bit register when FEAT\_D128 is implemented, TCR2\_EL2.D128 == 1 and HCR\_EL2.E2H == 1
- 64-bit register when FEAT\_D128 is not implemented or TCR2\_EL2.D128 == 0

## Field descriptions

**When FEAT\_D128 is implemented, TCR2\_EL2.D128 == 1 and HCR\_EL2.E2H == 1:**

127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
RES0																															
95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
RES0								BADDR[50:43]								RES0															
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
ASID																BADDR[42:0]															
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
BADDR[42:0]																								RES0				SKL	CnF		

### Bits [127:88]

Reserved, res0.

### BADDR, bits [87:80, 47:5]

Translation table base address:

- Bits A[55:x] of the stage 1 translation table base address bits are in register bits[87:80, 47:x].
- Bits A[(x-1):0] of the stage 1 translation table base address are zero.

Address bit x is the minimum address bit required to align the translation table to the size of the table. x is calculated based on  $\text{LOG2}(\text{StartTableSize})$ , as described in VMSAv9-128. The smallest permitted value of x is 5.

The BADDR field is split as follows:

- BADDR[50:43] is TTBR1\_EL2[87:80].
- BADDR[42:0] is TTBR1\_EL2[47:5].

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

### Bits [79:64]

Reserved, res0.

### ASID, bits [63:48]

An ASID for the translation table base address. The [TCR\\_EL2.A1](#) field selects either TTBR0\_EL2.ASID or TTBR1\_EL2.ASID.

If the implementation has only 8 bits of ASID, then the upper 8 bits of this field are res0.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

#### **Bits [4:3]**

Reserved, res0.

#### **SKL, bits [2:1]**

Skip Level associated with translation table walks using [TTBR1\\_EL2](#).

This determines the number of levels to be skipped from the regular start level of the Stage 1 EL2 translation table walks using [TTBR1\\_EL2](#).

<b>SKL</b>	<b>Meaning</b>
0b00	Skip 0 level from the regular start level.
0b01	Skip 1 level from the regular start level.
0b10	Skip 2 levels from the regular start level.
0b11	Skip 3 levels from the regular start level.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

#### **CnP, bit [0]**

**When FEAT\_TTCNP is implemented:**

Common not Private. This bit indicates whether each entry that is pointed to by TBR1\_EL2 is a member of a common set that can be used by every PE in the Inner Shareable domain for which the value of TTBR1\_EL2.CnP is 1.

<b>CnP</b>	<b>Meaning</b>
------------	----------------

- 
- 0b0 The translation table entries pointed to by TTBR1\_EL2 for the current ASID are permitted to differ from corresponding entries for TTBR1\_EL2 for other PEs in the Inner Shareable domain. This is not affected by:
- The value of TTBR1\_EL2.CnP on those other PEs.
  - The value of the current ASID.
- 0b1 The translation table entries pointed to by TTBR1\_EL2 are the same as the translation table entries for every other PE in the Inner Shareable domain for which the value of TTBR1\_EL2.CnP is 1 and all of the following apply:
- The translation table entries are pointed to by TTBR1\_EL2.
  - The ASID is the same as the current ASID.
- 

This bit is permitted to be cached in a TLB.

---

### Note

- TTBR1\_EL2 is accessible only when the value of [HCR\\_EL2.E2H](#) is 1, meaning the current translation regime is the EL2&0 regime.
  - If the value of the TTBR1\_EL2.CnP bit is 1 on multiple PEs in the same Inner Shareable domain and those TTBR1\_EL2s do not point to the same translation table entries when the other conditions specified for the case when the value of CnP is 1 apply, then the results of translations are constrained unpredictable, see 'CONSTRAINED UNPREDICTABLE behaviors due to caching of control or data values'.
-

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

**Otherwise:**

Reserved, res0.

**When FEAT\_D128 is not implemented or TCR2\_EL2.D128 == 0:**

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
ASID																BADDR[47:1]															
BADDR[47:1]																															CnP
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

**ASID, bits [63:48]**

An ASID for the translation table base address. The [TCR\\_EL2.A1](#) field selects either TTBR0\_EL2.ASID or TTBR1\_EL2.ASID.

If the implementation has only 8 bits of ASID, then the upper 8 bits of this field are res0.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

**BADDR[47:1], bits [47:1]**

Translation table base address:

- Bits A[47:x] of the stage 1 translation table base address bits are in register bits[47:x].
- Bits A[(x-1):0] of the stage 1 translation table base address are zero.

Address bit x is the minimum address bit required to align the translation table to the size of the table. The AArch64 Virtual Memory System Architecture chapter describes how x is calculated based on the value of [TCR\\_EL2.T1SZ](#), the translation stage, and the translation granule size.

---

**Note**

If an OA size of more than 48 bits is in use, and the translation table has fewer than eight entries, the table must be aligned to 64 bytes. Otherwise the translation table must be aligned to the size of the table.

---

If the value of [TCR\\_EL2](#).{I}PS is not 0b110, then:

- Register bits[(x-1):1] are res0.
- If the implementation supports 52-bit PAs and IPAs, then bits A[51:48] of the stage 1 translation table base address are 0b0000.

If FEAT\_LPA is implemented and the value of [TCR\\_EL2](#).{I}PS is 0b110, then:

- Bits A[51:48] of the stage 1 translation table base address bits are in register bits[5:2].
  - Register bit[1] is res0.
  - The smallest permitted value of x is 6.
  - When x>6, register bits[(x-1):6] are res0.
- 

**Note**

The OA size specified by [TCR\\_EL2](#).{I}PS is determined as follows:

- The value of [TCR\\_EL2](#).PS when the value of [HCR\\_EL2](#).E2H is 0.
- The value of [TCR\\_EL2](#).IPS when the value of [HCR\\_EL2](#).E2H is 1.

[TCR\\_EL2](#).{I}PS==0b110 is permitted when:

- FEAT\_LPA is implemented and the 64KB translation granule is used.
- FEAT\_LPA2 is implemented and the 4KB or 16KB translation granule is used.

When the value of [ID\\_AA64MMFR0\\_EL1](#).PARange indicates that the implementation does not support a 52 bit PA size, if a translation table lookup uses this register when the Effective value of [TCR\\_EL2](#).{I}PS is 0b110 and the value of register bits[5:2] is nonzero, an Address size fault is generated.

When the value of [ID\\_AA64MMFR0\\_EL1.PARange](#) indicates that the implementation supports a 56 bit PA size, bits A[55:52] of the stage 1 translation table base address are zero.

---

If any register bit[47:1] that is defined as res0 has the value 1 when a translation table walk is done using TTBR1\_EL2, then the translation table base address might be misaligned, with effects that are constrained unpredictable, and must be one of the following:

- Bits A[(x-1):0] of the stage 1 translation table base address are treated as if all the bits are zero. The value read back from the corresponding register bits is either the value written to the register or zero.
- The result of the calculation of an address for a translation table walk using this register can be corrupted in those bits that are nonzero.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

#### **CnP, bit [0]**

**When FEAT\_TTCNP is implemented:**

Common not Private. This bit indicates whether each entry that is pointed to by TBR1\_EL2 is a member of a common set that can be used by every PE in the Inner Shareable domain for which the value of TTBR1\_EL2.CnP is 1.

<b>CnP</b>	<b>Meaning</b>
0b0	The translation table entries pointed to by TTBR1_EL2 for the current ASID are permitted to differ from corresponding entries for TTBR1_EL2 for other PEs in the Inner Shareable domain. This is not affected by: <ul style="list-style-type: none"><li>• The value of TTBR1_EL2.CnP on those other PEs.</li><li>• The value of the current ASID.</li></ul>

0b1      The translation table entries pointed to by TTBR1\_EL2 are the same as the translation table entries for every other PE in the Inner Shareable domain for which the value of TTBR1\_EL2.CnP is 1 and all of the following apply:

- The translation table entries are pointed to by TTBR1\_EL2.
- The ASID is the same as the current ASID.

---

This bit is permitted to be cached in a TLB.

---

### Note

- TTBR1\_EL2 is accessible only when the value of [HCR\\_EL2.E2H](#) is 1, meaning the current translation regime is the EL2&0 regime.
- If the value of the TTBR1\_EL2.CnP bit is 1 on multiple PEs in the same Inner Shareable domain and those TTBR1\_EL2s do not point to the same translation table entries when the other conditions specified for the case when the value of CnP is 1 apply, then the results of translations are constrained unpredictable, see 'CONSTRAINED UNPREDICTABLE behaviors due to caching of control or data values'.

---

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

### Otherwise:

Reserved, res0.

## Accessing TTBR1\_EL2

When [HCR\\_EL2.E2H](#) is 1, without explicit synchronization, access from EL2 using the mnemonic TTBR1\_EL2 or TTBR1\_EL1 are not guaranteed to be ordered with respect to accesses using the other mnemonic.



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

## MRS <Xt>, TTBR1\_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b0010	0b0000	0b001

```
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
    X[t, 64] = TTBR1_EL2<63:0>;
elsif PSTATE.EL == EL3 then
    X[t, 64] = TTBR1_EL2<63:0>;
```

## MSR TTBR1\_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b0010	0b0000	0b001

```
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
    TTBR1_EL2<63:0> = X[t, 64];
elsif PSTATE.EL == EL3 then
    TTBR1_EL2<63:0> = X[t, 64];
```

## MRS <Xt>, TTBR1\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0000	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGTR_EL2.TTBR1_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
        X[t, 64] = NVMem[0x210];
    else
        X[t, 64] = TTBR1_EL1<63:0>;
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = TTBR1_EL2<63:0>;
    else
        X[t, 64] = TTBR1_EL1<63:0>;
elsif PSTATE.EL == EL3 then
    X[t, 64] = TTBR1_EL1<63:0>;

```

## MSR TTBR1\_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0000	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGWTR_EL2.TTBR1_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
        NVMem[0x210] = X[t, 64];
    else
        TTBR1_EL1<63:0> = X[t, 64];
elsif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        TTBR1_EL2<63:0> = X[t, 64];
    else
        TTBR1_EL1<63:0> = X[t, 64];
elsif PSTATE.EL == EL3 then
    TTBR1_EL1<63:0> = X[t, 64];

```

**When FEAT\_D128 is implemented**

**MRRS <Xt+1>, <Xt>, TTBR1\_EL2**

op0	op1	CRn	CRm	op2
0b11	0b100	0b0010	0b0000	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.D128En == '0' then
        UNDEFINED;
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
    else
        (X[t + 1, 64], X[t, 64]) =
        (TTBR1_EL2<127:64>, TTBR1_EL2<63:0>);
elsif PSTATE.EL == EL3 then
    (X[t + 1, 64], X[t, 64]) = (TTBR1_EL2<127:64>,
    TTBR1_EL2<63:0>);
```

**When FEAT\_D128 is implemented**

**MSRR TTBR1\_EL2, <Xt+1>, <Xt>**

op0	op1	CRn	CRm	op2
0b11	0b100	0b0010	0b0000	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.D128En == '0' then
        UNDEFINED;
```

```

        elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x14);
            end
        else
            (TTBR1_EL2<127:64>, TTBR1_EL2<63:0>) = (X[t
+ 1, 64], X[t, 64]);
        elsif PSTATE.EL == EL3 then
            (TTBR1_EL2<127:64>, TTBR1_EL2<63:0>) = (X[t + 1,
64], X[t, 64]);

```

#### When FEAT\_D128 is implemented

MRRS <Xt+1>, <Xt>, TTBR1\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0000	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.D128En == '0' then
        UNDEFINED;
    elsif EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HFGTR_EL2.TTBR1_EL1 == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
        end
    elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
'111' then
        (X[t + 1, 64], X[t, 64]) = (NVMem[0x218],
NVMem[0x210]);
    else
        (X[t + 1, 64], X[t, 64]) =
(TTBR1_EL1<127:64>, TTBR1_EL1<63:0>);
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && SCR_EL3.D128En == '0' then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;

```

```

        else
            AArch64.SystemAccessTrap(EL3, 0x14);
        elsif HCR_EL2.E2H == '1' then
            (X[t + 1, 64], X[t, 64]) =
                (TTBR1_EL2<127:64>, TTBR1_EL2<63:0>);
        else
            (X[t + 1, 64], X[t, 64]) =
                (TTBR1_EL1<127:64>, TTBR1_EL1<63:0>);
        elsif PSTATE.EL == EL3 then
            (X[t + 1, 64], X[t, 64]) = (TTBR1_EL1<127:64>,
                TTBR1_EL1<63:0>);

```

**When FEAT\_D128 is implemented**

**MSRR TTBR1\_EL1, <Xt+1>, <Xt>**

op0	op1	CRn	CRm	op2
0b11	0b000	0b0010	0b0000	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
    && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
    when SDD == '1'" && SCR_EL3.D128En == '0' then
        UNDEFINED;
    elsif EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif EL2Enabled() &&
    IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
    SCR_EL3.FGTEn == '1') && HFGWTR_EL2.TTBR1_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x14);
    elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
        elsif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
            (NVMem[0x218], NVMem[0x210]) = (X[t + 1,
            64], X[t, 64]);
        else
            (TTBR1_EL1<127:64>, TTBR1_EL1<63:0>) = (X[t
            + 1, 64], X[t, 64]);
    elsif PSTATE.EL == EL2 then
        if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
        && boolean IMPLEMENTATION_DEFINED "EL3 trap priority
        when SDD == '1'" && SCR_EL3.D128En == '0' then
            UNDEFINED;
        elsif HaveEL(EL3) && SCR_EL3.D128En == '0' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;

```

```
        else
            AArch64.SystemAccessTrap(EL3, 0x14);
        elsif HCR_EL2.E2H == '1' then
            (TTBR1_EL2<127:64>, TTBR1_EL2<63:0>) = (X[t
+ 1, 64], X[t, 64]);
        else
            (TTBR1_EL1<127:64>, TTBR1_EL1<63:0>) = (X[t
+ 1, 64], X[t, 64]);
        elsif PSTATE.EL == EL3 then
            (TTBR1_EL1<127:64>, TTBR1_EL1<63:0>) = (X[t + 1,
64], X[t, 64]);
```

---

[AArch32  
Registers](#)

[AArch64  
Registers](#)

[AArch32  
Instructions](#)

[AArch64  
Instructions](#)

[Index by  
Encoding](#)

[External  
Registers](#)

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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