
TLBIP RVAE3, TLBIP RVAE3NXS, TLB Range Invalidate by VA, EL3

The TLBIP RVAE3, TLBIP RVAE3NXS characteristics are:

Purpose

If EL3 is implemented, invalidates cached copies of translation table entries from TLBs that meet all the following requirements:

- The entry is a 128-bit stage 1 translation table entry, from any level of the translation table walk up to the level indicated in the TTL hint.

Or the entry is 64-bit a stage 1 translation table entry, from any level of the translation table walk, if TTL is 0b00.
- The entry would be used to translate any of the VAs in the specified address range using the EL3 translation regime.
- The entry is within the address range determined by the formula $[\text{BaseADDR} \leq \text{VA} < \text{BaseADDR} + ((\text{NUM} + 1) * 2^{(5 * \text{SCALE} + 1) * \text{Translation_Granule_Size}})]$.

The invalidation applies to the PE that executes this System instruction.

For 128-bit translation table entry, the range of addresses invalidated is unpredictable when Block or Page size corresponding to TTL and TG, for the translation system is not aligned.

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_D128 is implemented. Otherwise, direct accesses to TLBIP RVAE3, TLBIP RVAE3NXS are undefined.

Attributes

TLBIP RVAE3, TLBIP RVAE3NXS is a 128-bit System instruction.

Field descriptions

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																		BaseADDR[55:12]																	
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				
BaseADDR[55:12]																																			
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				
RES0																TG	SCALE	NUM				TTL	RES0												
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				
RES0																																			

Bits [127:108]

Reserved, res0.

BaseADDR[55:12], bits [107:64]

The starting address for the range of the maintenance instructions. This field is BaseADDR[55:12] for all translation granules.

Bits [63:48]

Reserved, res0.

TG, bits [47:46]

Translation granule size.

TG	Meaning
0b00	Reserved.
0b01	4K translation granule.
0b10	16K translation granule.
0b11	64K translation granule.

The instruction takes a translation granule size for the translations that are being invalidated. If the translations used a different translation granule size than the one being specified, then the architecture does not require that the instruction invalidates any entries.

SCALE, bits [45:44]

The exponent element of the calculation that is used to produce the upper range.

NUM, bits [43:39]

The base element of the calculation that is used to produce the upper range.

TTL, bits [38:37]

TTL Level hint. The TTL hint is only guaranteed to invalidate:

- Non-leaf-level entries in the range up to but not including the level described by the TTL hint.
- Leaf-level entries in the range that match the level described by the TTL hint.

TTL	Meaning
0b00	The entries in the range can be using any level for the translation table entries.
0b01	The TTL hint indicates level 1. If FEAT_LPA2 is not implemented, when using a 16KB translation granule, this value is reserved and hardware should treat this field as 0b00.
0b10	The TTL hint indicates level 2.
0b11	The TTL hint indicates level 3.

Bits [36:0]

Reserved, res0.

Executing TLBIP RVAE3, TLBIP RVAE3NXS

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

TLBIP RVAE3{, <Xt>, <Xt2>}

op0	op1	CRn	CRm	op2
0b01	0b110	0b1000	0b0110	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    AArch64.TLBIP_RVA(SecurityStateAtEL(EL3),
Regime_EL3, VMID[], Shareability_NSH, TLBILevel_Any,
TLBI_AllAttr, X[t2, 64]:X[t, 64]);

```

TLBIP RVAE3NXS{, <Xt>, <Xt2>}

op0	op1	CRn	CRm	op2
0b01	0b110	0b1001	0b0110	0b001

```

if !IsFeatureImplemented(FEAT_XS) then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    UNDEFINED;
elsif PSTATE.EL == EL2 then
    UNDEFINED;
elsif PSTATE.EL == EL3 then
    AArch64.TLBIP_RVA(SecurityStateAtEL(EL3),
Regime_EL3, VMID[], Shareability_NSH, TLBILevel_Any,
TLBI_ExcludeXS, X[t2, 64]:X[t, 64]);

```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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