TRBMAR_EL1, Trace Buffer Memory Attribute Register

The TRBMAR_EL1 characteristics are:

Purpose

Controls Trace Buffer Unit accesses to memory.

If the trace buffer pointers specify virtual addresses, the address properties are defined by the translation tables and this register is ignored.

Configuration

AArch64 System register TRBMAR_EL1 bits [63:0] are architecturally mapped to External register TRBMAR_EL1[63:0] when FEAT_TRBE_EXT is implemented.

This register is present only when FEAT_TRBE is implemented. Otherwise, direct accesses to TRBMAR_EL1 are undefined.

Attributes

TRBMAR_EL1 is a 64-bit register.

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
RES0																															
RES0																						PAS		SH		Attr					
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 [63:12]

Reserved, res0.

PAS, bits [11:10]
When FEAT_TRBE_EXT is implemented:

Physical address specifier. Defines the PAS attribute for memory addressed by the buffer in External mode.

PAS	Meaning	Applies when
0b00	Secure.	When Secure state is implemented

0b01	Non-secure.	
0b10	Root.	When FEAT_RME is implemented
0b11	Realm.	When FEAT_RME is implemented

All other values are reserved.

If the Trace Buffer Unit is using external mode and either TRBMAR_EL1.PAS is set to a reserved value, or the implementation defined authentication interface prohibits invasive debug of the Security state corresponding to the physical address space selected by TRBMAR_EL1.PAS, then when the Trace Buffer Unit receives trace data from the trace unit, it does not write the trace data to memory and generates a trace buffer management event. That is, if any of the following apply:

- `ExternalInvasiveDebugEnabled`() == FALSE.
- Secure state is implemented, `ExternalSecureInvasiveDebugEnabled`() == FALSE, and TRBMAR_EL1.PAS is `0b00`.
- FEAT_RME is implemented, `ExternalRootInvasiveDebugEnabled`() == FALSE, and TRBMAR_EL1.PAS is `0b10`.
- FEAT_RME is implemented, `ExternalRealmInvasiveDebugEnabled`() == FALSE, and TRBMAR_EL1.PAS is `0b11`.

This field is ignored by the PE when `SelfHostedTraceEnabled`() == TRUE.

The reset behavior of this field is:

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

Otherwise:

Reserved, res0.

SH, bits [9:8]

Trace buffer shareability domain. Defines the shareability domain for Normal memory used by the trace buffer.

SH	Meaning
0b00	Non-shareable.
0b10	Outer Shareable.

0b11	Inner Shareable.

All other values are reserved.

This field is ignored when TRBMAR_EL1.Attr specifies any of the following memory types:

- Any Device memory type.
- Normal memory, Inner Non-cacheable, Outer Non-cacheable.

All Device and Normal Inner Non-cacheable Outer Non-cacheable memory regions are always treated as Outer Shareable.

The reset behavior of this field is:

- On a Cold reset, when FEAT_TRBE_EXT is implemented, this field resets to an architecturally unknown value.
- On a Warm reset, when FEAT_TRBE_EXT is not implemented, this field resets to an architecturally unknown value.

Attr, bits [7:0]
When TRBMAR_EL1.Attr == 0bxxxx0000:

Trace buffer memory type and attributes. Defines the memory type and, for Normal memory, the cacheability attributes, for memory addressed by the trace buffer.

Attr	Meaning	Applies when
0x00	Device-nGnRnE memory.	
0x40	Normal memory, Inner Non-cacheable, Outer Non-cacheable with the XS attribute set to 0.	When FEAT_XS is implemented
0xA0	Normal memory, Inner Write-through Cacheable, Outer Write-through Cacheable, Non-transient, Read-Allocate with the XS attribute set to 0.	When FEAT_XS is implemented

0xF0	Tagged Normal memory, Outer Write-Back Non-transient, Read-allocate Write-allocate.	When FEAT_MTE2 is implemented

All other values are reserved.

The reset behavior of this field is:

- On a Cold reset, when FEAT_TRBE_EXT is implemented, this field resets to an architecturally unknown value.
- On a Warm reset, when FEAT_TRBE_EXT is not implemented, this field resets to an architecturally unknown value.

When TRBMAR_EL1.Attr == 0b0000xxxx and TRBMAR_EL1.Attr != 0b00000000:

Trace buffer memory attributes. Defines the Device memory attributes for memory addressed by the trace buffer.

Attr	Meaning	Applies when
0x04	Device-nGnRE memory.	
0x08	Device-nGRE memory.	
0x0C	Device-GRE memory.	
0x01	Device-nGnRnE memory with the XS attribute set to 0.	When FEAT_XS is implemented
0x05	Device-nGnRE memory with the XS attribute set to 0.	When FEAT_XS is implemented
0x09	Device-nGRE memory with the XS attribute set to 0.	When FEAT_XS is implemented
0x0D	Device-GRE memory with the XS attribute set to 0.	When FEAT_XS is implemented

All other values are reserved.

The reset behavior of this field is:

- On a Cold reset, when FEAT_TRBE_EXT is implemented, this field resets to an architecturally unknown value.
- On a Warm reset, when FEAT_TRBE_EXT is not implemented, this field resets to an architecturally unknown value.

When TRBMAR_EL1.Attr != 0bxxxx0000 and TRBMAR_EL1.Attr != 0b0000xxxx:

Trace buffer memory type and attributes. Defines the memory type and, for Normal memory, the Outer and Inner cacheability attributes, for memory addressed by the trace buffer.

Attr	Meaning
0b0001xxxx	Normal memory, Outer Write-Through Transient, Write-allocate.
0b0010xxxx	Normal memory, Outer Write-Through Transient, Read-allocate.
0b0011xxxx	Normal memory, Outer Write-Through Transient, Read-allocate Write-allocate.
0b0100xxxx	Normal memory, Outer Non-cacheable.
0b0101xxxx	Normal memory, Outer Write-Back Transient, Write-allocate.
0b0110xxxx	Normal memory, Outer Write-Back Transient, Read-allocate.
0b0111xxxx	Normal memory, Outer Write-Back Transient, Read-allocate Write-allocate.
0b1000xxxx	Normal memory, Outer Write-Through Non-transient, No allocate.
0b1001xxxx	Normal memory, Outer Write-Through Non-transient, Write-allocate.
0b1010xxxx	Normal memory, Outer Write-Through Non-transient, Read-allocate.

`0b1011xxxx`	Normal memory, Outer Write-Through Non-transient, Read-allocate Write-allocate.
`0b1100xxxx`	Normal memory, Outer Write-Back Non-transient, No allocate.
`0b1101xxxx`	Normal memory, Outer Write-Back Non-transient, Write-allocate.
`0b1110xxxx`	Normal memory, Outer Write-Back Non-transient, Read-allocate.
`0b1111xxxx`	Normal memory, Outer Write-Back Non-transient, Read-allocate Write-allocate.
`0bxxxx0001`	Normal memory, Inner Write-Through Transient, Write-allocate.
`0bxxxx0010`	Normal memory, Inner Write-Through Transient, Read-allocate.
`0bxxxx0011`	Normal memory, Inner Write-Through Transient, Read-allocate Write-allocate.
`0bxxxx0100`	Normal memory, Inner Non-cacheable.
`0bxxxx0101`	Normal memory, Inner Write-Back Transient, Write-allocate.
`0bxxxx0110`	Normal memory, Inner Write-Back Transient, Read-allocate.
`0bxxxx0111`	Normal memory, Inner Write-Back Transient, Read-allocate Write-allocate.
`0bxxxx1000`	Normal memory, Inner Write-Through Non-transient, No allocate.
`0bxxxx1001`	Normal memory, Inner Write-Through Non-transient, Write-allocate.
`0bxxxx1010`	Normal memory, Inner Write-Through Non-transient, Read-allocate.

0bxxxx1011	Normal memory, Inner Write-Through Non-transient, Read-allocate Write-allocate.
0bxxxx1100	Normal memory, Inner Write-Back Non-transient, No allocate.
0bxxxx1101	Normal memory, Inner Write-Back Non-transient, Write-allocate.
0bxxxx1110	Normal memory, Inner Write-Back Non-transient, Read-allocate.
0bxxxx1111	Normal memory, Inner Write-Back Non-transient, Read-allocate Write-allocate.

The reset behavior of this field is:

- On a Cold reset, when FEAT_TRBE_EXT is implemented, this field resets to an architecturally unknown value.
- On a Warm reset, when FEAT_TRBE_EXT is not implemented, this field resets to an architecturally unknown value.

Otherwise:

Reserved, res0.

Accessing TRBMAR_EL1

The PE might ignore a write to TRBMAR_EL1 if any of the following apply:

- TRBLIMITR_EL1.E == 1, and either FEAT_TRBE_EXT is not implemented or the Trace Buffer Unit is using Self-hosted mode.
- TRBLIMITR_EL1.XE == 1, FEAT_TRBE_EXT is implemented, and the Trace Buffer Unit is using External mode.

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

MRS <Xt>, TRBMAR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1001	0b1011	0b100

```
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'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGRTR_EL2.TRBMAR_EL1 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.E2TB == 'x0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRBMAR_EL1;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = TRBMAR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = TRBMAR_EL1;
```

MSR TRBMAR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1001	0b1011	0b100

```
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'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif EL2Enabled() &&
IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
SCR_EL3.FGTEn == '1') && HDFGWTR_EL2.TRBMAR_EL1 ==
'1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif EL2Enabled() && MDCR_EL2.E2TB == 'x0' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRBMAR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        UNDEFINED;
    elsif HaveEL(EL3) && (MDCR_EL3.NSTB[0] == '0' ||
MDCR_EL3.NSTB[1] != SCR_EL3.NS ||
(IsFeatureImplemented(FEAT_RME) && MDCR_EL3.NSTBE !=
SCR_EL3.NSE)) then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        TRBMAR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    TRBMAR_EL1 = X[t, 64];
```