MDRAR_EL1, Monitor Debug ROM Address Register

The MDRAR_EL1 characteristics are:

Purpose

Defines the base physical address of a 4KB-aligned memory-mapped debug component, usually a ROM table that locates and describes the memory-mapped debug components in the system. Armv8 deprecates any use of this register.

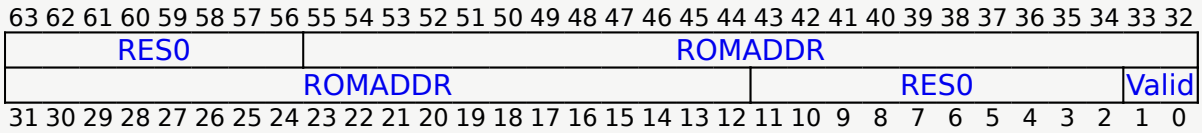
Configuration

AArch64 System register MDRAR_EL1 bits [63:0] are architecturally mapped to AArch32 System register DBGDRAR[63:0].

Attributes

MDRAR_EL1 is a 64-bit register.

Field descriptions

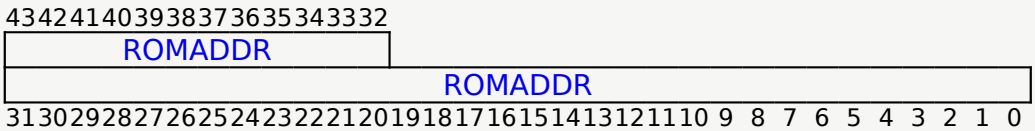


Bits [63:56]

Reserved, res0.

ROMADDR, bits [55:12]

ROMADDR encoding when FEAT_D128 is implemented and MDRAR_EL1.Valid != 0b00



ROMADDR, bits [43:0]

Bits [55:12] of the ROM table physical address.

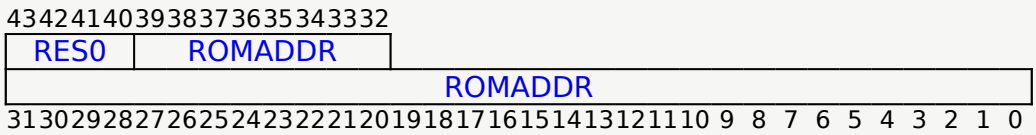
Bits [11:0] of the ROM table physical address are zero.

For implementations with fewer than 56 physical address bits, the corresponding upper bits of this field are res0

In an implementation that includes EL3, ROMADDR is an address in Non-secure PA space. It is implementation defined whether the ROM table is also accessible in Secure PA space. If FEAT_RME is implemented, it is implementation defined whether the ROM table is also accessible in the Root or Realm PA spaces.

Arm strongly recommends that bits ROMADDR[(PAsize-1):32] are zero in any system where the implementation only supports execution in AArch32 state.

ROMADDR encoding when FEAT_D128 is not implemented, FEAT_LPA is implemented and MDRAR_EL1.Valid != 0b00



Bits [43:40]

Reserved, res0.

ROMADDR, bits [39:0]

Bits [51:12] of the ROM table physical address.

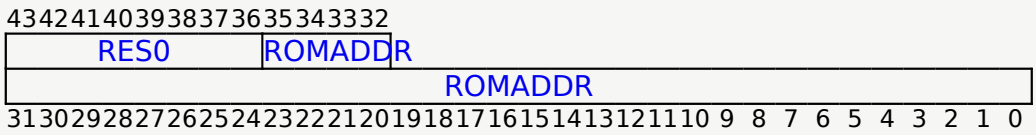
Bits [11:0] of the ROM table physical address are zero.

For implementations with fewer than 52 physical address bits, the corresponding upper bits of this field are res0

In an implementation that includes EL3, ROMADDR is an address in Non-secure PA space. It is implementation defined whether the ROM table is also accessible in Secure PA space. If FEAT_RME is implemented, it is implementation defined whether the ROM table is also accessible in the Root or Realm PA spaces.

Arm strongly recommends that bits ROMADDR[(PAsize-1):32] are zero in any system where the implementation only supports execution in AArch32 state.

ROMADDR encoding when FEAT_D128 is not implemented, FEAT_LPA is not implemented and MDRAR_EL1.Valid != 0b00



Bits [43:36]

Reserved, res0.

ROMADDR, bits [35:0]

Bits [39:12] of the ROM table physical address.

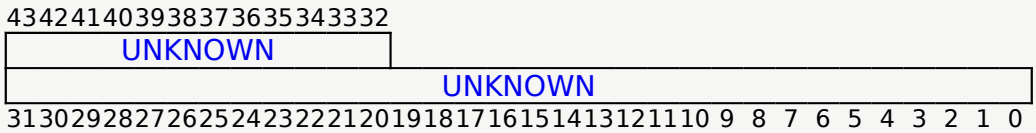
Bits [11:0] of the ROM table physical address are zero.

For implementations with fewer than 48 physical address bits, the corresponding upper bits of this field are res0

In an implementation that includes EL3, ROMADDR is an address in Non-secure PA space. It is implementation defined whether the ROM table is also accessible in Secure PA space. If FEAT_RME is implemented, it is implementation defined whether the ROM table is also accessible in Root or Realm PA spaces.

Arm strongly recommends that bits ROMADDR[(PAsize-1):32] are zero in any system where the implementation only supports execution in AArch32 state.

ROMADDR encoding when MDRAR_EL1.Valid == 0b00



Bits [43:0]

Reserved, unknown.

Bits [11:2]

Reserved, res0.

Valid, bits [1:0]

This field indicates whether the ROM Table address is valid.

Valid	Meaning
0b00	ROM Table address is not valid. Software must ignore ROMADDR.
0b11	ROM Table address is valid.

Other values are reserved.

Arm recommends implementations set this field to zero.

Accessing MDRAR_EL1

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

MRS <Xt>, MDRAR_EL1

op0	op1	CRn	CRm	op2
0b10	0b000	0b0001	0b0000	0b000

```
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.TDA == '1' then
        UNDEFINED;
    elsif EL2Enabled() && MDCR_EL2.<TDE,TDRA> !=
'00' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elsif HaveEL(EL3) && MDCR_EL3.TDA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = MDRAR_EL1;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.TDA == '1' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.TDA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
```

```
    else
        X[t, 64] = MDRAR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = MDRAR_EL1;
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94