

MPAMVPM7_EL2, MPAM Virtual PARTID Mapping Register 7

The MPAMVPM7_EL2 characteristics are:

Purpose

MPAMVPM7_EL2 provides mappings from virtual PARTIDs 28 - 31 to physical PARTIDs.

[MPAMIDR_EL1](#).VPMR_MAX field gives the index of the highest implemented MPAMVPM<n>_EL2 registers. VPMR_MAX can be as large as 7 (8 registers) or 32 virtual PARTIDs. If [MPAMIDR_EL1](#).VPMR_MAX == 0, there is only a single MPAMVPM<n>_EL2 register, [MPAMVPM0_EL2](#).

Virtual PARTID mapping is enabled by [MPAMHCR_EL2](#).EL1_VPMEN for PARTIDs in [MPAM1_EL1](#) and by [MPAMHCR_EL2](#).EL0_VPMEN for [MPAM0_EL1](#).

A virtual-to-physical PARTID mapping entry, PhyPARTID<n>, is valid only when the [MPAMVPMV_EL2](#).VPM_V bit in bit position n is set to 1.

Configuration

This register is present only when FEAT_MPAM is implemented, MPAMIDR_EL1.HAS_HCR == 1 and UInt(MPAMIDR_EL1.VPMR_MAX) == 7. Otherwise, direct accesses to MPAMVPM7_EL2 are undefined.

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

Attributes

MPAMVPM7_EL2 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 |
| PhyPARTID31 | | | | | | | | | | | | | | | | PhyPARTID30 | | | | | | | | | | | | | | | |
| PhyPARTID29 | | | | | | | | | | | | | | | | PhyPARTID28 | | | | | | | | | | | | | | | |
| 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 |

PhyPARTID31, bits [63:48]

Virtual PARTID Mapping Entry for virtual PARTID 31. PhyPARTID31 gives the mapping of virtual PARTID 31 to a physical PARTID.

The reset behavior of this field is:

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

PhyPARTID30, bits [47:32]

Virtual PARTID Mapping Entry for virtual PARTID 30. PhyPARTID30 gives the mapping of virtual PARTID 30 to a physical PARTID.

The reset behavior of this field is:

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

PhyPARTID29, bits [31:16]

Virtual PARTID Mapping Entry for virtual PARTID 29. PhyPARTID29 gives the mapping of virtual PARTID 29 to a physical PARTID.

The reset behavior of this field is:

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

PhyPARTID28, bits [15:0]

Virtual PARTID Mapping Entry for virtual PARTID 28. PhyPARTID28 gives the mapping of virtual PARTID 28 to a physical PARTID.

The reset behavior of this field is:

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

Accessing MPAMVPM7_EL2

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

MRS <Xt>, MPAMVPM7_EL2

| op0 | op1 | CRn | CRm | op2 |
|------------|------------|------------|------------|------------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b111 |

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        X[t, 64] = NVMem[0x978];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        if HaveEL(EL3) && MPAM3_EL3.TRAPLOWER == '1'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                AArch64.SystemAccessTrap(EL2, 0x18);
        else
            UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if HaveEL(EL3) && MPAM3_EL3.TRAPLOWER == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
        else
            X[t, 64] = MPAMVPM7_EL2;
    elsif PSTATE.EL == EL3 then
        X[t, 64] = MPAMVPM7_EL2;

```

MSR MPAMVPM7_EL2, <Xt>

| op0 | op1 | CRn | CRm | op2 |
|------|-------|--------|--------|-------|
| 0b11 | 0b100 | 0b1010 | 0b0110 | 0b111 |

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        NVMem[0x978] = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        if HaveEL(EL3) && MPAM3_EL3.TRAPLOWER == '1'
then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;
            else
                AArch64.SystemAccessTrap(EL3, 0x18);
            else
                AArch64.SystemAccessTrap(EL2, 0x18);
        else
            UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if HaveEL(EL3) && MPAM3_EL3.TRAPLOWER == '1' then
            if Halted() && EDSCR.SDD == '1' then
                UNDEFINED;

```

```
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
        else
            MPAMVPM7_EL2 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        MPAMVPM7_EL2 = X[t, 64];
```

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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