MPAMF_AIDR, MPAM Architecture Identification Register

The MPAMF_AIDR characteristics are:

Purpose

Identifies the version of the MPAM architecture that this MSC implements.

Note: The following values are defined for bits [7:0]:

- `0x01` == MPAM architecture v0.1
- `0x10` == MPAM architecture v1.0
- `0x11` == MPAM architecture v1.1

Configuration

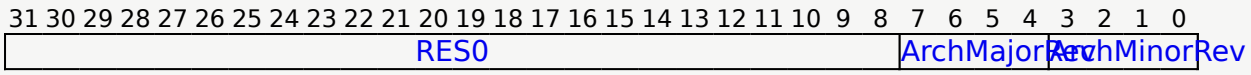
This register is present only when FEAT_MPAM is implemented. Otherwise, direct accesses to MPAMF_AIDR are res0.

The power and reset domain of each MSC component is specific to that component.

Attributes

MPAMF_AIDR is a 32-bit register.

Field descriptions



Bits [31:8]

Reserved, res0.

ArchMajorRev, bits [7:4]

Major revision of the MPAM architecture implemented by the MSC.

This table shows the only valid combinations of MPAM version numbers in an MSC. FORCE_NS functionality is only available in MPAM v0.1.

ArchMajorRev	ArchMinorRev	MPAMv	Available
0	0		None.
0	1	v0.1	MPAMv1.0 + MPAMv1.1 + FORCE_NS
1	0	v1.0	MPAMv1.0
1	1	v1.1	MPAMv1.0 + MPAMv1.1 - FORCE_NS

Use of MPAMv0.1 in MSCs is restricted to limited circumstances. The MSC must be able to initiate requests in the Secure address space which have MPAM PARTID forced to the Non-secure space with that forcing not controllable or observable by the software that configures the device for Secure requests. Please contact Arm before setting MPAMF_AIDR to report MPAMv0.1.

ArchMinorRev, bits [3:0]

Minor revision of the MPAM architecture implemented by the MSC.

See the table in the description of the ArchMajorRev field in this register.

Accessing MPAMF_AIDR

This register is within the MPAM feature page memory frames.

In a system that supports Secure, Non-secure, Root, and Realm memory maps, there must be MPAM feature pages in all four address maps.

MPAMF_AIDR is read-only.

MPAMF_AIDR must be readable from the Secure, Non-secure, Root, and Realm MPAM feature pages.

MPAMF_AIDR must have the same contents in the Secure, Non-secure, Root, and Realm MPAM feature pages.

MPAMF_AIDR can be accessed through the memory-mapped interfaces:

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_s	0x0020	MPAMF_AIDR_s

Accesses on this interface are **RO**.

Component	Frame	Offset	Instance

MPAM	MPAMF_BASE_ns	0x0020	MPAMF_AIDR_ns

Accesses on this interface are **RO**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rt	0x0020	MPAMF_AIDR_rt

When FEAT_RME is implemented, accesses on this interface are **RO**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rl	0x0020	MPAMF_AIDR_rl

When FEAT_RME is implemented, accesses on this interface are **RO**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94