

MPAMCFG_CPBM<n>, MPAM Cache Portion Bitmap Partition Configuration Register, n = 0 - 1023

The MPAMCFG_CPBM<n> characteristics are:

Purpose

The MPAMCFG_CPBM<n> register array gives access to the cache portion bitmap. Each register in the array is a read/write register that configures the cache portions numbered from <n * 32> to <31 + (n * 32)> that a PARTID is allowed to allocate.

After setting [MPAMCFG_PART_SEL](#) with a PARTID, software writes to the MPAMCFG_CPBM<n> register to configure which cache portions the PARTID is allowed to allocate.

The MPAMCFG_CPBM<n> register that contains the bitmap bit corresponding to cache portion p has n equal to p[15:5]. The field, P<x>, of that MPAMCFG_CPBM<n> register that contains the bitmap bit corresponding to cache portion p has x equal to p[4:0].

MPAMCFG_CPBM<n>_s controls cache portions for the Secure PARTID selected by the Secure instance of [MPAMCFG_PART_SEL](#). MPAMCFG_CPBM<n>_ns controls the cache portions for the Non-secure PARTID selected by the Non-secure instance of [MPAMCFG_PART_SEL](#). MPAMCFG_CPBM<n>_rt controls cache portions for the Root PARTID selected by the Root instance of [MPAMCFG_PART_SEL](#). MPAMCFG_CPBM<n>_rl controls the cache portions for the Realm PARTID selected by the Non-secure instance of [MPAMCFG_PART_SEL](#).

If [MPAMF_IDR](#).HAS_RIS is 1, the control settings accessed are those of the resource instance currently selected by [MPAMCFG_PART_SEL](#).RIS and the PARTID selected by [MPAMCFG_PART_SEL](#).PARTID_SEL.

Configuration

This register is present only when FEAT_MPAM is implemented and MPAMF_IDR.HAS_CPOR_PART == 1. Otherwise, direct accesses to MPAMCFG_CPBM<n> are res0.

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

Attributes

MPAMCFG_CPBM<n> is a 32-bit register.

Field descriptions

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
P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4

P<x>, bit [x], for x = 31 to 0

Portion allocation control bit. Each cache portion allocation control bit, MPAMCFG_CPBM<n>.P<x>, grants permission to the PARTID selected by [MPAMCFG_PART_SEL](#) to allocate cache lines within cache portion $\langle n * 32 \rangle + x$.

P<x>	Meaning
0b0	The PARTID is not permitted to allocate into cache portion $\langle n * 32 \rangle + x$.
0b1	The PARTID is permitted to allocate within cache portion $\langle n * 32 \rangle + x$.

The number of bits in the cache portion partitioning bit map of this component is given in [MPAMF_CPOR_IDR](#).CPBM_WD.

[MPAMF_CPOR_IDR](#).CPBM_WD contains a value from 1 to 2^{15} , inclusive. Values of [MPAMF_CPOR_IDR](#).CPBM_WD greater than 32 require an array of 32-bit [MPAMCFG_CPBM<n>](#) registers to access the cache portion bitmap, up to 1024 registers.

When $(n * 32) + x > \text{UInt}(\text{MPAMF_CPOR_IDR.CPBM_WD})$, access to this field is **RES0**.

Accessing MPAMCFG_CPBM<n>

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:

- MPAMCFG_CPBM<n>_s must only be accessible from the Secure MPAM feature page.
- MPAMCFG_CPBM<n>_ns must only be accessible from the Non-secure MPAM feature page.
- MPAMCFG_CPBM<n>_rt must only be accessible from the Root MPAM feature page.
- MPAMCFG_CPBM<n>_rl must only be accessible from the Realm MPAM feature page.

MPAMCFG_CPBM<n>_s, MPAMCFG_CPBM<n>_ns, MPAMCFG_CPBM<n>_rt, and MPAMCFG_CPBM<n>_rl must be separate registers:

- The Secure instance (MPAMCFG_CPBM<n>_s) accesses the cache portion bitmap used for Secure PARTIDs.
- The Non-secure instance (MPAMCFG_CPBM<n>_ns) accesses the cache portion bitmap used for Non-secure PARTIDs.
- The Root instance (MPAMCFG_CPBM<n>_rt) accesses the cache portion bitmap used for Root PARTIDs.
- The Realm instance (MPAMCFG_CPBM<n>_rl) accesses the cache portion bitmap used for Realm PARTIDs.

When RIS is implemented, loads and stores to MPAMCFG_CPBM<n> access the cache portion bitmap configuration settings for the cache resource instance selected by [MPAMCFG_PART_SEL](#).RIS and the PARTID selected by [MPAMCFG_PART_SEL](#).PARTID_SEL.

When RIS is not implemented, loads and stores to MPAMCFG_CPBM<n> access the cache portion bitmap configuration settings for the PARTID selected by [MPAMCFG_PART_SEL](#).PARTID_SEL.

When PARTID narrowing is implemented, loads and stores to MPAMCFG_CPBM<n> access the cache portion bitmap configuration settings for the internal PARTID selected by [MPAMCFG_PART_SEL](#).PARTID_SEL, and [MPAMCFG_PART_SEL](#).INTERNAL must be 1.

When PARTID narrowing is not implemented, loads and stores to MPAMCFG_CPBM<n> access the cache portion bitmap configuration settings for the request PARTID selected by [MPAMCFG_PART_SEL](#).PARTID_SEL, and [MPAMCFG_PART_SEL](#).INTERNAL must be 0.

MPAMCFG_CPBM<n> can be accessed through the memory-mapped interfaces:

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_s	0x1000 + (4 * n)	MPAMCFG_CPBM<n>_s

Accesses on this interface are **RW**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_ns	0x1000 + (4 * n)	MPAMCFG_CPBM<n>_ns

Accesses on this interface are **RW**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rt	0x1000 + (4 * n)	MPAMCFG_CPBM<n>_rt

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

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rl	0x1000 + (4 * n)	MPAMCFG_CPBM<n>_rl

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

[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.