MPAMCFG_EN_FLAGS, MPAM Partition Configuration Enable Flags Register

The MPAMCFG_EN_FLAGS characteristics are:

Purpose

Enable flags for 32 PARTIDs.

MPAMCFG_EN_FLAGS_s gives read/write access to 32 Secure PARTIDs. MPAMCFG_EN_FLAGS_ns gives read/write access to 32 Non-secure PARTIDs. MPAMCFG_EN_FLAGS_rl gives read/write access to 32 Realm PARTIDs. MPAMCFG_EN_FLAGS_rt gives read/write access to 32 Root PARTIDs.

Configuration

This register is present only when (FEAT_MPAMv0p1 is implemented or FEAT_MPAMv1p1 is implemented) and MPAMF_IDR.HAS_ENDIS == 1. Otherwise, direct accesses to MPAMCFG_EN_FLAGS are res0.

Attributes

MPAMCFG_EN_FLAGS 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
EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16	EN15	EN14

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

PARTID Enable flags. The group of flags accessed is selected by MPAMCFG_PART_SEL.PARTID_SEL & `0xFFE0` in bit [0] to (MPAMCFG_PART_SEL.PARTID_SEL & `0xFFE0`) + 31 in bit [31].

EN<x>	Meaning
`0b0`	The PARTID is disabled.
`0b1`	The PARTID is enabled.

Each bit in MPAMCFG_EN_FLAGS gives access to the same state as controlled by MPAMCFG_EN and MPAMCFG_DIS.

Bits MPAMCFG_EN_FLAGS.EN<x>, where (MPAMCFG_PART_SEL.PARTID_SEL & `0xFFE0`) + x is greater than MPAMF_IDR.PARTID_MAX, are not required to be implemented.

As with other partitioning controls, the enable flag for PARTID 0 must be reset to `0b1` (enabled).

Accessing MPAMCFG_EN_FLAGS

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_EN_FLAGS_s must only be accessible from the Secure MPAM feature page.
- MPAMCFG_EN_FLAGS_ns must only be accessible from the Non-secure MPAM feature page.
- MPAMCFG_EN_FLAGS_rt must only be accessible from the Root MPAM feature page.
- MPAMCFG_EN_FLAGS_rl must only be accessible from the Realm MPAM feature page.

MPAMCFG_EN_FLAGS_s, MPAMCFG_EN_FLAGS_ns, MPAMCFG_EN_FLAGS_rt, and MPAMCFG_EN_FLAGS_rl must be separate registers:

- The Secure instance (MPAMCFG_EN_FLAGS_s) accesses the PARTID enable used for Secure PARTIDs.
- The Non-secure instance (MPAMCFG_EN_FLAGS_ns) accesses the PARTID enable used for Non-secure PARTIDs.
- The Root instance (MPAMCFG_EN_FLAGS_rt) accesses the PARTID enable used for Root PARTIDs.
- The Realm instance (MPAMCFG_EN_FLAGS_rl) accesses the PARTID enable used for Realm PARTIDs.

When RIS is implemented, loads and stores to MPAMCFG_EN_FLAGS access the PARTID enable configuration settings for the PARTID enable 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_EN_FLAGS access the PARTID enable configuration settings for the PARTID selected by MPAMCFG_PART_SEL.PARTID_SEL.

When PARTID narrowing is implemented, loads and stores to MPAMCFG_EN_FLAGS access the PARTID enable 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_EN_FLAGS access the PARTID enable configuration settings for the request PARTID selected by MPAMCFG_PART_SEL.PARTID_SEL, and MPAMCFG_PART_SEL.INTERNAL must be 0.

MPAMCFG_EN_FLAGS can be accessed through the memory-mapped interfaces:

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_s	`0x0320`	MPAMCFG_EN_FLAGS_s

Accesses on this interface are **RW**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_ns	`0x0320`	MPAMCFG_EN_FLAGS_ns

Accesses on this interface are **RW**.

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rt	`0x0320`	MPAMCFG_EN_FLAGS_rt

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

Component	Frame	Offset	Instance
MPAM	MPAMF_BASE_rl	`0x0320`	MPAMCFG_EN_FLAGS_rl

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