

EDCIDSR, External Debug Context ID Sample Register

The EDCIDSR characteristics are:

Purpose

Contains the sampled value of the Context ID, captured on reading [EDPCSR](#)[31:0].

Configuration

EDCIDSR is in the Core power domain.

This register is present only when FEAT_PCSRv8 is implemented and FEAT_PCSRv8p2 is not implemented. Otherwise, direct accesses to EDCIDSR are res0.

Implemented only if the optional PC Sample-based Profiling Extension is implemented in the external debug registers space.

Note

FEAT_PCSRv8p2 implements the PC Sample-based Profiling Extension in the Performance Monitors registers space.

Attributes

EDCIDSR 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	3	2	1	0
CONTEXTIDR																															

CONTEXTIDR, bits [31:0]

Context ID. The value of CONTEXTIDR that is associated with the most recent [EDPCSR](#) sample. When the most recent [EDPCSR](#) sample is generated:

- If EL1 is using AArch64, then the Context ID is sampled from [CONTEXTIDR_EL1](#).

- If EL1 is using AArch32, then the Context ID is sampled from [CONTEXTIDR](#).
- If EL3 is implemented and is using AArch32, then [CONTEXTIDR](#) is a banked register, and EDCIDSR samples the current banked copy of [CONTEXTIDR](#) for the Security state that is associated with the most recent [EDPCSR](#) sample.

Because the value written to EDCIDSR is an indirect read of CONTEXTIDR, it is constrained unpredictable whether EDCIDSR is set to the original or new value if [EDPCSR](#) samples:

- An instruction that writes to CONTEXTIDR.
- The next Context synchronization event.
- Any instruction executed between these two instructions.

The reset behavior of this field is:

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

Accessing EDCIDSR

implementation defined extensions to external debug might make the value of this register unknown, see 'Permitted behavior that might make the PC Sample-based profiling registers UNKNOWN'.

EDCIDSR can be accessed through the external debug interface:

Component	Offset	Instance
Debug	0x0A4	EDCIDSR

This interface is accessible as follows:

- When IsCorePowered(), !DoubleLockStatus() and !OSLockStatus(), accesses to this register are **RO**.
- Otherwise, accesses to this register generate an error response.

[AArch32
Registers](#)

[AArch64
Registers](#)

[AArch32
Instructions](#)

[AArch64
Instructions](#)

[Index by
Encoding](#)

[External
Registers](#)

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

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