EDITR, External Debug Instruction Transfer Register

The EDITR characteristics are:

Purpose

Used in Debug state for passing instructions to the PE for execution.

Configuration

EDITR is in the Core power domain.

Attributes

EDITR is a 32-bit register.

Field descriptions

When AArch32 is supported and in AArch32 state:

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
hw2	hw1

hw2, bits [31:16]

Second halfword of the T32 instruction to be executed on the PE. When EDITR contains a 16-bit T32 instruction, this field is ignored. For more information, see 'Behavior in Debug state'.

Note

The hw2 field is displayed on the left. This is not the usual convention for display of T32 instruction halfwords.

---

hw1, bits [15:0]

First halfword of the T32 instruction to be executed on the PE.

Note

The hw1 field is displayed on the right.
This is not the usual convention for display
of T32 instruction halfwords.

When AArch64 is supported and in AArch64 state:

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
A64 instruction to be executed on the PE																															

Bits [31:0]

A64 instruction to be executed on the PE.

Accessing EDITR

If EDSCR.ITE == 0 when the PE exits Debug state on receiving a Restart request trigger event, the behavior of any instruction issued through the ITR in Normal access mode that has not completed execution is constrained unpredictable, and must do one of the following:

- It must complete execution in Debug state before the PE executes the restart sequence.
- It must complete execution in Non-debug state before the PE executes the restart sequence.
- It must be abandoned. This means that the instruction does not execute. Any registers or memory accessed by the instruction are left in an unknown state.

EDITR ignores writes if the PE is in Non-debug state.

EDITR can be accessed through the external debug interface:

Component	Offset	Instance
Debug	0x084	EDITR

This interface is accessible as follows:

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