EDDFR, External Debug Feature Register

The EDDFR characteristics are:

Purpose

Provides top level information about the debug system.

Note

Debuggers must use EDDEVARCH to determine the Debug architecture version.

For general information about the interpretation of the ID registers, see 'Principles of the ID scheme for fields in ID registers'.

Configuration

The power domain of EDDFR is implementation defined.

Attributes

EDDFR is a 64-bit register.

Field descriptions

63 62 61 60	59 58 57 56	55 54 53 52 51 50 49 48 47 46 45 44	43 42 41 40	39 38 37 36	35 34 33 32
UNKNOWN	ExtTrcBuff	UNKNOWN	TraceFilt	UNKNOWN	

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
CTX_CMPs	SEBEP	WRPs	PMSS	BRPs	PMUVer	TraceVer	UNKNOWN

Bits [63:60]

Reserved, unknown.

ExtTrcBuff, bits [59:56]

Trace Buffer External Mode Extension. Defined values are:

ExtTrcBuff	Meaning
0b0000	Trace Buffer Extension not implemented or Trace Buffer External Mode not implemented.

0b0001	Trace Buffer Extension implemented and Trace Buffer External Mode implemented.

All other values are reserved.

FEAT_TRBE_EXT implements the functionality identified by the value 0b0001.

In an implementation that supports AArch64, this field has the same value as ID_AA64DFR0_EL1.ExtTrcBuff.

Bits [55:44]

Reserved, unknown.

TraceFilt, bits [43:40]

Armv8.4 Self-hosted Trace Extension version. Defined values are:

TraceFilt	Meaning
0b0000	Armv8.4 Self-hosted Trace Extension is not implemented.
0b0001	Armv8.4 Self-hosted Trace Extension is implemented.

All other values are reserved.

FEAT_TRF implements the functionality added by 0b0001.

From Armv8.4, the permitted values are 0b0000 and 0b0001.

Bits [39:32]

Reserved, unknown.

CTX_CMPs, bits [31:28]

Number of breakpoints that are context-aware, minus 1.

The value of this field is never greater than EDDFR.BRPs.

In an implementation that supports AArch64, this field has the same value as ID_AA64DFR0_EL1.CTX_CMPs.

If FEAT_Debugv8p9 is implemented and 16 or more breakpoints that are context-aware are implemented, this field reads as 0b1111.

SEBEP, bits [27:24]

This field either has the same value as ID_AA64DFR0_EL1.SEBEP or reads as zero.

WRPs, bits [23:20]

Number of watchpoints, minus 1.

In an implementation that supports AArch64, this field has the same value as ID_AA64DFR0_EL1.WRPs.

If FEAT_Debugv8p9 is implemented and 16 or more watchpoints are implemented, this field reads as `0b1111`.

The value of `0b0000` is reserved.

Note

Only watchpoints 0 to 15 can be accessed in AArch32 state.

PMSS, bits [19:16]

This field either has the same value as ID_AA64DFR0_EL1.PMSS or reads as zero.

BRPs, bits [15:12]

Number of breakpoints, minus 1.

In an implementation that supports AArch64, this field has the same value as ID_AA64DFR0_EL1.BRPs.

If FEAT_Debugv8p9 is implemented and 16 or more breakpoints are implemented, this field reads as `0b1111`.

The value of `0b0000` is reserved.

Note

Only breakpoints 0 to 15 can be accessed in AArch32 state.

PMUVer, bits [11:8]

Performance Monitors Extension version.

This field does not follow the standard ID scheme, but uses the alternative ID scheme described in 'Alternative ID scheme used for the Performance Monitors Extension version'

Defined values are:

PMUVer	Meaning
0b0000	Performance Monitors Extension not implemented.
0b0001	Performance Monitors Extension, PMUv3 implemented.
0b0100	PMUv3 for Armv8.1. As 0b0001, and adds support for: • Extended 16-bit PMU.PMEVTYPER<n>_EL0.evtCount field. • If EL2 is implemented, the MDCR_EL2.HPMD control.
0b0101	PMUv3 for Armv8.4. As 0b0100, and adds support for the PMMIR_EL1 register.
0b0110	PMUv3 for Armv8.5. As 0b0101, and adds support for: • 64-bit event counters. • If EL2 is implemented, the MDCR_EL2.HCCD control. • If EL3 is implemented, the MDCR_EL3.SCCD control.
0b0111	PMUv3 for Armv8.7. As 0b0110, and adds support for: • The PMU.PMCR_EL0.FZO and, if EL2 is implemented, MDCR_EL2.HPMFZO controls. • If EL3 is implemented, the MDCR_EL3.{MPMX,MCCD} controls.
0b1000	PMUv3 for Armv8.8. As 0b0111, and: • Extends the Common event number space to include 0x0040 to 0x00BF and 0x4040 to 0x40BF. • Removes the constrained unpredictable behaviors if a reserved or unimplemented PMU event number is selected.

0b1001	PMUv3 for Armv8.9. As 0b1000, and: • Updates the definitions of existing PMU events. • Adds support for the PMUSERENR_EL0.UEN control and the PMUACR_EL1 register. • Adds support for the EDECR.PME control.
0b1111	implementation defined form of performance monitors supported, PMUv3 not supported. Arm does not recommend this value for new implementations.

All other values are reserved.

FEAT_PMUv3 implements the functionality identified by the value 0b0001.

FEAT_PMUv3p1 implements the functionality identified by the value 0b0100.

FEAT_PMUv3p4 implements the functionality identified by the value 0b0101.

FEAT_PMUv3p5 implements the functionality identified by the value 0b0110.

FEAT_PMUv3p7 implements the functionality identified by the value 0b0111.

FEAT_PMUv3p8 implements the functionality identified by the value 0b1000.

FEAT_PMUv3p9 implements the functionality identified by the value 0b1001.

From Armv8.1, if FEAT_PMUv3 is implemented, the value 0b0001 is not permitted.

From Armv8.4, if FEAT_PMUv3 is implemented, the value 0b0100 is not permitted.

From Armv8.5, if FEAT_PMUv3 is implemented, the value 0b0101 is not permitted.

From Armv8.7, if FEAT_PMUv3 is implemented, the value 0b0110 is not permitted.

From Armv8.8, if FEAT_PMUv3 is implemented, the value 0b0111 is not permitted.

From Armv8.9, if FEAT_PMUv3 is implemented, the value `0b1000` is not permitted.

In an implementation that supports AArch64, this field has the same value as ID_AA64DFR0_EL1.PMUVer.

TraceVer, bits [7:4]

Trace support. Indicates whether System register interface to a trace unit is implemented. Defined values are:

TraceVer	Meaning
`0b0000`	Trace unit System registers not implemented.
`0b0001`	Trace unit System registers implemented.

All other values are reserved.

A value of `0b0000` only indicates that no System register interface to a trace unit is implemented. A trace unit might nevertheless be implemented without a System register interface.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64DFR0_EL1.TraceVer.

Bits [3:0]

Reserved, unknown.

Accessing EDDFR

EDDFR can be accessed through the external debug interface:

Component	Offset	Instance	Range
Debug	`0xD28`	EDDFR	31:0

This interface is accessible as follows:

- When IsCorePowered() and !DoubleLockStatus(), accesses to this register are **RO**.
- Otherwise, accesses to this register are **IMPDEF**.

Component	Offset	Instance	Range
Debug	`0xD2C`	EDDFR	63:32

This interface is accessible as follows:

- When IsCorePowered() and !DoubleLockStatus(), accesses to this register are **RO**.

- Otherwise, accesses to this register are **IMPDEF**.

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94