EDPFR, External Debug Processor Feature Register

The EDPFR characteristics are:

Purpose

Provides information about implemented PE features.

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 EDPFR is implementation defined.

Attributes

EDPFR 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	UNKNOWN	RES0	UNKNOWN	AMU	UNKNOWN	SEL2	SVE
UNKNOWN	GIC	AdvSIMD	FP	EL3	EL2	EL1	EL0
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

Bits [63:60]
From Armv8.5:

Reserved, unknown.

Otherwise:

Reserved, res0.

Bits [59:56]
From Armv8.5:

Reserved, unknown.

Otherwise:

Reserved, res0.

Bits [55:52]

Reserved, res0.

Bits [51:48]
From Armv8.4:

Reserved, unknown.

Otherwise:

Reserved, res0.

AMU, bits [47:44]

Indicates support for Activity Monitors Extension. Defined values are:

AMU	Meaning
0b0000	Activity Monitors Extension is not implemented.
0b0001	FEAT_AMUv1 is implemented.
0b0010	FEAT_AMUv1p1 is implemented. As 0b0001 and adds support for virtualization of the activity monitor event counters.

All other values are reserved.

FEAT_AMUv1 implements the functionality identified by the value 0b0001.

FEAT_AMUv1p1 implements the functionality identified by the value 0b0010.

In Armv8.0, the only permitted value is 0b0000.

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

From Armv8.6, the permitted values are 0b0000, 0b0001, and 0b0010.

Bits [43:40]
From Armv8.2:

Reserved, unknown.

Otherwise:

Reserved, res0.

SEL2, bits [39:36]

Secure EL2. Defined values are:

SEL2	Meaning
`0b0000`	Secure EL2 is not implemented.
`0b0001`	Secure EL2 is implemented.

All other values are reserved.

SVE, bits [35:32]

Scalable Vector Extension. Defined values are:

SVE	Meaning
`0b0000`	SVE is not implemented.
`0b0001`	SVE is implemented.

All other values are reserved.

Bits [31:28]
From Armv8.2:

Reserved, unknown.

Otherwise:

Reserved, res0.

GIC, bits [27:24]

System register GIC interface support. Defined values are:

GIC	Meaning
`0b0000`	GIC CPU interface system registers not implemented.
`0b0001`	System register interface to versions 3.0 and 4.0 of the GIC CPU interface is supported.
`0b0011`	System register interface to version 4.1 of the GIC CPU interface is supported.

All other values are reserved.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.GIC.

AdvSIMD, bits [23:20]

Advanced SIMD. Defined values are:

AdvSIMD	Meaning
`0b0000`	Advanced SIMD is implemented, including support for the following SISD and SIMD operations: • Integer byte, halfword, word and doubleword element operations. • Single-precision and double-precision floating-point arithmetic. • Conversions between single-precision and half-precision data types, and double-precision and half-precision data types.
`0b0001`	As for `0b0000`, and also includes support for half-precision floating-point arithmetic.
`0b1111`	Advanced SIMD is not implemented.

All other values are reserved.

This field must have the same value as the FP field.

The permitted values are:

- `0b0000` in an implementation with Advanced SIMD support, that does not include the FEAT_FP16 extension.
- `0b0001` in an implementation with Advanced SIMD support, that includes the FEAT_FP16 extension.
- `0b1111` in an implementation without Advanced SIMD support.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.AdvSIMD.

FP, bits [19:16]

Floating-point. Defined values are:

FP	Meaning
0b0000	Floating-point is implemented, and includes support for: • Single-precision and double-precision floating-point types. • Conversions between single-precision and half-precision data types, and double-precision and half-precision data types.
0b0001	As for 0b0000, and also includes support for half-precision floating-point arithmetic.
0b1111	Floating-point is not implemented.

All other values are reserved.

This field must have the same value as the AdvSIMD field.

The permitted values are:

- 0b0000 in an implementation with floating-point support, that does not include the FEAT_FP16 extension.
- 0b0001 in an implementation with floating-point support, that includes the FEAT_FP16 extension.
- 0b1111 in an implementation without floating-point support.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.FP.

EL3, bits [15:12]

AArch64 EL3 Exception level handling. Defined values are:

EL3	Meaning
0b0000	EL3 is not implemented or cannot be executed in AArch64 state.
0b0001	EL3 can be executed in AArch64 state only.
0b0010	EL3 can be executed in both Execution states.

When the value of EDAA32PFR.EL3 is nonzero, this field must be 0b0000.

All other values are reserved.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.EL3.

EL2, bits [11:8]

AArch64 EL2 Exception level handling. Defined values are:

EL2	Meaning
`0b0000`	EL2 is not implemented or cannot be executed in AArch64 state.
`0b0001`	EL2 can be executed in AArch64 state only.
`0b0010`	EL2 can be executed in both Execution states.

When the value of EDAA32PFR.EL2 is nonzero, this field must be `0b0000`.

All other values are reserved.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.EL2.

EL1, bits [7:4]

AArch64 EL1 Exception level handling. Defined values are:

EL1	Meaning
`0b0000`	EL1 cannot be executed in AArch64 state. EL1 can be executed in AArch32 state only.
`0b0001`	EL1 can be executed in AArch64 state only.
`0b0010`	EL1 can be executed in both Execution states.

All other values are reserved.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.EL1.

EL0, bits [3:0]

AArch64 EL0 Exception level handling. Defined values are:

EL0	Meaning

`0b0000`	EL0 cannot be executed in AArch64 state. EL0 can be executed in AArch32 state only.
`0b0001`	EL0 can be executed in AArch64 state only.
`0b0010`	EL0 can be executed in both Execution states.

All other values are reserved.

In an Armv8-A implementation that supports AArch64, this field returns the value of ID_AA64PFR0_EL1.EL0.

Accessing EDPFR

EDPFR can be accessed through the external debug interface:

Component	Offset	Instance	Range
Debug	`0xD20`	EDPFR	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	`0xD24`	EDPFR	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**.