ID_AA64PFR1_EL1, AArch64 Processor Feature Register 1

The ID_AA64PFR1_EL1 characteristics are:

Purpose

Reserved for future expansion of information about implemented PE features in AArch64 state.

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

Configuration

There are no configuration notes.

Attributes

ID_AA64PFR1_EL1 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
PFAR	DF2	MTEX	THE	GCS	MTE_frac	NMI	CSV2_frac
RNDR_trap	SME	RES0	MPAM_frac	RAS_frac	MTE	SSBS	BT
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

PFAR, bits [63:60]

Support for physical fault address registers, FEAT_PFAR. Defined values are:

PFAR	Meaning
0b0000	FEAT_PFAR is not implemented.
0b0001	FEAT_PFAR is implemented. Includes support for the PFAR_ELx and, if EL3 is implemented, MFAR_EL3 registers.

All other values are reserved.

FEAT_PFAR implements the functionality identified by the value 0b0001.

DF2, bits [59:56]

Support for error exception routing extensions, FEAT_DoubleFault2. Defined values are:

DF2	Meaning
0b0000	FEAT_DoubleFault2 is not implemented. **Note** This does not mean that FEAT_DoubleFault, as identified by ID_AA64PFR0_EL1.RAS >= 0b0010, is not implemented.
0b0001	FEAT_DoubleFault2 is implemented. As ID_AA64PFR0_EL1.RAS == 0b0010, and also includes support for routing error exceptions: • Traps for masked error exceptions, HCRX_EL2.TMEA and SCR_EL3.TMEA. • Additional controls for masking SError interrupts, SCTLR2_EL1.NMEA, and SCTLR2_EL2.NMEA. • Additional controls for taking external aborts to the SError interrupt vector, SCTLR2_EL1.EASE and SCTLR2_EL2.EASE.

All other values are reserved.

FEAT_DoubleFault2 implements the functionality identified by the value 0b0001.

MTEX, bits [55:52]

Additional tag checking modes for MTE. Defined values are:

MTEX	Meaning
0b0000	Support for Memory Tagging when Address tagging is enabled.

0b0001	As 0b0000, and the following additional modes are supported: • Canonical Tag checking, identified as FEAT_MTE_CANONICAL_TAGS. • Memory tagging with Address tagging disabled, identified as FEAT_MTE_NO_ADDRESS_TAGS.

Note

ID_AA64PFR1_EL1.MTE identifies Memory Tagging when Address tagging is enabled.

ID_AA64PFR1_EL1.MTE_frac identifies whether or not Asynchronous Faulting and asymmetric Tag Check Fault handling are supported, in conjunction with the value of ID_AA64PFR1_EL1.MTE. Support for Asynchronous Faulting and asymmetric Tag Check Fault handling applies to all tag checking modes, including those introduced by ID_AA64PFR1_EL1.MTEX.

All other values are reserved.

This field is valid only if ID_AA64PFR1_EL1.MTE >= 0b0010.

FEAT_MTE4 implements the functionality identified by the value 0b0001.

From Armv8.9, the only permitted value is 0b0001.

THE, bits [51:48]

Support for Translation Hardening Extension. Defined values are:

THE	Meaning
0b0000	Translation Hardening Extension is not implemented.

0b0001	The RCW and RCWS instructions, their associated registers and traps are supported. If EL2 is implemented, the AssuredOnly check, TopLevel check, and their associated controls are also implemented. If EL2 and FEAT_GCS are implemented, VTCR_EL2.GCSH is implemented.

All other values are reserved.

FEAT_THE implements the functionality identified by the value 0b0001.

GCS, bits [47:44]

Support for Guarded Control Stack. Defined values are:

GCS	Meaning
0b0000	Guarded Control Stack is not implemented.
0b0001	Guarded Control Stack is implemented.

All other values are reserved.

FEAT_GCS implements the functionality identified by the value 0b0001.

MTE_frac, bits [43:40]

Support for Asynchronous Faulting and asymmetric Tag Check Fault handling. Defined values are:

MTE_frac	Meaning
0b0000	Asynchronous Faulting is supported. If ID_AA64PFR1_EL1.MTE >= 0b0011, asymmetric Tag Check Fault handling is supported.
0b1111	Asynchronous Faulting is not supported. If ID_AA64PFR1_EL1.MTE >= 0b0011, asymmetric Tag Check Fault handling is not supported.

All other values are reserved.

This field is valid only if ID_AA64PFR1_EL1.MTE >= 0b0010.

NMI, bits [39:36]

Non-maskable Interrupt. Indicates support for Non-maskable interrupts. Defined values are:

NMI	Meaning
0b0000	SCTLR_ELx.{SPINTMASK, NMI} and PSTATE.ALLINT with its associated instructions are not supported.
0b0001	SCTLR_ELx.{SPINTMASK, NMI} and PSTATE.ALLINT with its associated instructions are supported.

All other values are reserved.

FEAT_NMI implements the functionality identified by the value 0b0001.

From Armv8.8, the only permitted value is 0b0001.

CSV2_frac, bits [35:32]

CSV2 fractional field. Defined values are:

CSV2_frac	Meaning
0b0000	Either ID_AA64PFR0_EL1.CSV2 is not 0b0001, or the implementation does not disclose whether FEAT_CSV2_1p1 is implemented. FEAT_CSV2_1p2 is not implemented.
0b0001	FEAT_CSV2_1p1 is implemented, but FEAT_CSV2_1p2 is not implemented.
0b0010	FEAT_CSV2_1p2 is implemented.

All other values are reserved.

FEAT_CSV2_1p1 implements the functionality identified by the value 0b0001.

FEAT_CSV2_1p2 implements the functionality identified by the value 0b0010.

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

The values 0b0001 and 0b0010 are permitted only when ID_AA64PFR0_EL1.CSV2 is 0b0001.

RNDR_trap, bits [31:28]

Random Number trap to EL3 field. Defined values are:

RNDR_trap	Meaning
0b0000	Trapping of RNDR and RNDRRS to EL3 is not supported.
0b0001	Trapping of RNDR and RNDRRS to EL3 is supported. SCR_EL3.TRNDR is present.

All other values are reserved.

FEAT_RNG_TRAP implements the functionality identified by the value 0b0001.

SME, bits [27:24]

Scalable Matrix Extension. Defined values are:

SME	Meaning
0b0000	SME architectural state and programmers' model are not implemented.
0b0001	SME architectural state and programmers' model are implemented.
0b0010	As 0b0001, plus the SME2 ZT0 register.

All other values are reserved.

FEAT_SME implements the functionality identified by the value 0b0001.

FEAT_SME2 implements the functionality identified by the value 0b0010.

From Armv9.2, the permitted values are `0b0000`, `0b0001`, and `0b0010`.

If implemented, refer to ID_AA64SMFR0_EL1 and ID_AA64ZFR0_EL1 for information about which SME and SVE instructions are available.

Bits [23:20]

Reserved, res0.

MPAM_frac, bits [19:16]

Indicates the minor version number of support for the MPAM Extension.

Defined values are:

MPAM_frac	Meaning
`0b0000`	The minor version number of the MPAM extension is 0.
`0b0001`	The minor version number of the MPAM extension is 1.

All other values are reserved.

When combined with the major version number from ID_AA64PFR0_EL1.MPAM, The combined "major.minor" version is:

MPAM Extension version	MPAM	MPAM_frac
Not implemented.	`0b0000`	`0b0000`
v0.1 is implemented.	`0b0000`	`0b0001`
v1.0 is implemented.	`0b0001`	`0b0000`
v1.1 is implemented.	`0b0001`	`0b0001`

For more information, see 'The Memory Partitioning and Monitoring (MPAM) Extension'.

RAS_frac, bits [15:12]

RAS Extension fractional field. Defined values are:

RAS_frac	Meaning

0b0000	If ID_AA64PFR0_EL1.RAS == 0b0001, RAS Extension implemented.
0b0001	If ID_AA64PFR0_EL1.RAS == 0b0001, as 0b0000 and adds support for: • Additional ERXMISC<m>_EL1 System registers. • Additional System registers ERXPFGCDN_EL1, ERXPFGCTL_EL1, and ERXPFGF_EL1, and the SCR_EL3.FIEN and HCR_EL2.FIEN trap controls, to support the optional RAS Common Fault Injection Model Extension. Error records accessed through System registers conform to RAS System Architecture v1.1, which includes simplifications to ERR<n>STATUS, and support for the optional RAS Timestamp and RAS Common Fault Injection Model Extensions.

All other values are reserved.

FEAT_RASv1p1 implements the functionality identified by the value 0b0001.

This field is valid only if ID_AA64PFR0_EL1.RAS == 0b0001.

MTE, bits [11:8]

Support for the Memory Tagging Extension. Defined values are:

MTE	Meaning
0b0000	Memory Tagging Extension is not implemented.
0b0001	Instruction-only Memory Tagging Extension is implemented.
0b0010	Full Memory Tagging Extension is implemented. Support for Asynchronous Faulting is defined by ID_AA64PFR1_EL1.MTE_frac
0b0011	Memory Tagging Extension is implemented with support for asymmetric Tag Check Fault handling. Support for Asynchronous Faulting and asymmetric Tag Check Fault handling is defined by ID_AA64PFR1_EL1.MTE_frac.

All other values are reserved.

FEAT_MTE implements the functionality identified by the value 0b0001.

FEAT_MTE2 implements the functionality identified by the value 0b0010.

FEAT_MTE3 implements the functionality identified by the value 0b0011.

In Armv8.5, the permitted values are 0b0000, 0b0001, 0b0010, and 0b0011.

From Armv8.7, the value 0b0010 is not permitted.

SSBS, bits [7:4]

Speculative Store Bypassing controls in AArch64 state. Defined values are:

SSBS	Meaning
0b0000	AArch64 provides no mechanism to control the use of Speculative Store Bypassing.
0b0001	AArch64 provides the PSTATE.SSBS mechanism to mark regions that are Speculative Store Bypass Safe.
0b0010	As 0b0001, and adds the MSR and MRS instructions to directly read and write the PSTATE.SSBS field.

All other values are reserved.

FEAT_SSBS implements the functionality identified by the value 0b0001.

FEAT_SSBS2 implements the functionality identified by the value 0b0010.

BT, bits [3:0]

Branch Target Identification mechanism support in AArch64 state. Defined values are:

BT	Meaning
0b0000	The Branch Target Identification mechanism is not implemented.

0b0001	The Branch Target Identification mechanism is implemented.

All other values are reserved.

FEAT_BTI implements the functionality identified by the value 0b0001.

From Armv8.5, the only permitted value is 0b0001.

Accessing ID_AA64PFR1_EL1

Accesses to this register use the following encodings in the System register encoding space:

MRS <Xt>, ID_AA64PFR1_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0100	0b001

```
if PSTATE.EL == EL0 then
    if IsFeatureImplemented(FEAT_IDST) then
        if EL2Enabled() && HCR_EL2.TGE == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            AArch64.SystemAccessTrap(EL1, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64PFR1_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64PFR1_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64PFR1_EL1;
```


28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94