ID_AA64ISAR2_EL1, AArch64 Instruction Set Attribute Register 2

The ID_AA64ISAR2_EL1 characteristics are:

Purpose

Provides information about the features and instructions implemented 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

Note

Prior to the introduction of the features described by this register, this register was unnamed and reserved, res0 from EL1, EL2, and EL3.

Attributes

ID_AA64ISAR2_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
RES0	CSSC	RPRFM	RES0	PRFMSLC	SYSINSTR_128	SYSREG_128

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
CLRBHB	PAC_frac	BC	MOPS	APA3	GPA3	RPRES	WFxT

Bits [63:56]

Reserved, res0.

CSSC, bits [55:52]

Indicates support for common short sequence compression instructions. Defined values are:

CSSC	Meaning

0b0000	Common short sequence compression instructions are not implemented.
0b0001	Common short sequence compression instructions are implemented.

All other values are reserved.

FEAT_CSSC implements the functionality identified by the value 0b0001.

From Armv9.4, the value 0b0000 is not permitted.

RPRFM, bits [51:48]

RPRFM hint instruction. Defined values are:

RPRFM	Meaning
0b0000	RPRFM hint instruction is not implemented and is treated as a NOP.
0b0001	RPRFM hint instruction is implemented.

All other values are reserved.

FEAT_RPRFM implements the functionality identified by the value 0b0001.

Bits [47:44]

Reserved, res0.

PRFMSLC, bits [43:40]

Indicates whether the PRFM instructions support a system level cache option. Defined values are:

PRFMSLC	Meaning
0b0000	The PRFM instructions do not support the SLC target.
0b0001	The PRFM instructions support the SLC target.

All other values are reserved.

FEAT_PRFMSLC implements the functionality identified by 0b0001.

SYSINSTR_128, bits [39:36]

SYSINSTR_128. Indicates support for System instructions that can take 128-bit inputs. Defined values are:

SYSINSTR_128	Meaning
0b0000	System instructions that can take 128-bit inputs are not supported.
0b0001	System instructions that can take 128-bit inputs are supported.

All other values are reserved.

FEAT_SYSINSTR128 implements the functionality identified by 0b0001.

SYSREG_128, bits [35:32]

SYSREG_128. Indicates support for instructions to access 128-bit System Registers. Defined values are:

SYSREG_128	Meaning
0b0000	Instructions to access 128-bit System Registers are not supported.
0b0001	Instructions to access 128-bit System Registers are supported.

All other values are reserved.

FEAT_SYSREG128 implements the functionality identified by 0b0001.

CLRBHB, bits [31:28]

Indicates support for the CLRBHB instruction in AArch64 state. Defined values are:

CLRBHB	Meaning
0b0000	CLRBHB instruction is not implemented.
0b0001	CLRBHB instruction is implemented.

All other values are reserved.

FEAT_CLRBHB implements the functionality identified by 0b0001.

From Armv8.9, the value `0b0000` is not permitted.

PAC_frac, bits [27:24]

Indicates whether the ConstPACField() function used as part of the PAC addition returns FALSE or TRUE.

PAC_frac	Meaning
`0b0000`	ConstPACField() returns FALSE.
`0b0001`	ConstPACField() returns TRUE.

All other values are reserved.

FEAT_CONSTPACFIELD implements the functionality identified by `0b0001`.

From Armv8.3, the permitted values are `0b0000` and `0b0001`.

BC, bits [23:20]

Indicates support for the BC instruction in AArch64 state. Defined values are:

BC	Meaning
`0b0000`	BC instruction is not implemented.
`0b0001`	BC instruction is implemented.

All other values are reserved.

FEAT_HBC implements the functionality identified by the value `0b0001`.

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

MOPS, bits [19:16]

Indicates support for the Memory Copy and Memory Set instructions in AArch64 state.

MOPS	Meaning
`0b0000`	The Memory Copy and Memory Set instructions are not implemented in AArch64 state.

0b0001	The Memory Copy and Memory Set instructions are implemented in AArch64 state with the following exception. If FEAT_MTE is implemented, then SETGP*, SETGM* and SETGE* instructions are also supported.

All other values are reserved.

FEAT_MOPS implements the functionality identified by the value 0b0001.

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

APA3, bits [15:12]

Indicates whether the QARMA3 algorithm is implemented in the PE for address authentication in AArch64 state. This applies to all Pointer Authentication instructions other than the PACGA instruction. Defined values are:

APA3	Meaning
0b0000	Address Authentication using the QARMA3 algorithm is not implemented.
0b0001	Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC() and HaveEnhancedPAC2() functions returning FALSE.
0b0010	Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC() function returning TRUE and the HaveEnhancedPAC2() function returning FALSE.
0b0011	Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC2() function returning TRUE, the HaveFPAC() function returning FALSE, the HaveFPACCombined() function returning FALSE, and the HaveEnhancedPAC() function returning FALSE.

0b0100	Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC2() function returning TRUE, the HaveFPAC() function returning TRUE, the HaveFPACCombined() function returning FALSE, and the HaveEnhancedPAC() function returning FALSE.
0b0101	Address Authentication using the QARMA3 algorithm is implemented, with the HaveEnhancedPAC2() function returning TRUE, the HaveFPAC() function returning TRUE, the HaveFPACCombined() function returning TRUE, and the HaveEnhancedPAC() function returning FALSE.

All other values are reserved.

FEAT_PAuth implements the functionality identified by 0b0001.

FEAT_EPAC implements the functionality identified by 0b0010.

FEAT_PAuth2 implements the functionality identified by 0b0011.

FEAT_FPAC implements the functionality identified by 0b0100.

FEAT_FPACCOMBINE implements the functionality identified by 0b0101.

When this field is nonzero, FEAT_PACQARMA3 is implemented.

In Armv8.3, the permitted values are 0b0000, 0b0001, 0b0010, 0b0011, 0b0100, and 0b0101.

From Armv8.6, the permitted values are 0b0011, 0b0100, and 0b0101.

If the value of ID_AA64ISAR1_EL1.API is nonzero, or the value of ID_AA64ISAR1_EL1.APA is nonzero, this field must have the value 0b0000.

GPA3, bits [11:8]

Indicates whether the QARMA3 algorithm is implemented in the PE for generic code authentication in AArch64 state. Defined values are:

GPA3	Meaning
0b0000	Generic Authentication using the QARMA3 algorithm is not implemented.
0b0001	Generic Authentication using the QARMA3 algorithm is implemented. This includes the PACGA instruction.

All other values are reserved.

FEAT_PACQARMA3 implements the functionality identified by 0b0001.

From Armv8.3, the permitted values are 0b0000 and 0b0001.

If the value of ID_AA64ISAR1_EL1.GPI is nonzero, or the value of ID_AA64ISAR1_EL1.GPA is nonzero, this field must have the value 0b0000.

RPRES, bits [7:4]

Indicates support for 12 bits of mantissa in reciprocal and reciprocal square root instructions in AArch64 state, when FPCR.AH is 1. Defined values are:

RPRES	Meaning	Applies when
0b0000	Reciprocal and reciprocal square root estimates give 8 bits of mantissa, when FPCR.AH is 1.	When FPCR.AH == 1
0b0001	Reciprocal and reciprocal square root estimates give 12 bits of mantissa, when FPCR.AH is 1.	When FPCR.AH == 1

All other values are reserved.

FEAT_RPRES implements the functionality identified by the value 0b0001.

WFxT, bits [3:0]

Indicates support for the `WFET` and `WFIT` instructions in AArch64 state. Defined values are:

WFxT	Meaning
`0b0000`	`WFET` and `WFIT` are not supported.
`0b0010`	`WFET` and `WFIT` are supported, and the register number is reported in the ESR_ELx on exceptions.

All other values are reserved.

FEAT_WFxT implements the functionality identified by the value `0b0010`.

From Armv8.7, the only permitted value is `0b0010`.

Accessing ID_AA64ISAR2_EL1

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

MRS <Xt>, ID_AA64ISAR2_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0110	0b010

```
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() &&
(IsFeatureImplemented(FEAT_FGT) || !
IsZero(ID_AA64ISAR2_EL1) || boolean
IMPLEMENTATION_DEFINED "ID_AA64ISAR2_EL1 trapped by
HCR_EL2.TID3") && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ISAR2_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ISAR2_EL1;
elsif PSTATE.EL == EL3 then
```

```
        X[t, 64] = ID_AA64ISAR2_EL1;
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94