[AArch32 Registers](#) [AArch64 Registers](#) [AArch32 Instructions](#) [AArch64 Instructions](#) [Index by Encoding](#) [External Registers](#)

ID_MMFR3_EL1, AArch32 Memory Model Feature Register 3

The ID_MMFR3_EL1 characteristics are:

Purpose

Provides information about the implemented memory model and memory management support in AArch32 state.

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

Configuration

AArch64 System register ID_MMFR3_EL1 bits [31:0] are architecturally mapped to AArch32 System register ID_MMFR3[31:0].

Attributes

ID_MMFR3_EL1 is a 64-bit register.

Field descriptions

When AArch32 is supported:

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

Supersec	CMemSz	CohWalk	PAN	MaintBcst	BPMaint	CMaintSW	CMaintVA
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 [63:32]

Reserved, res0.

Supersec, bits [31:28]

Supersections. On a VMSA implementation, indicates whether Supersections are supported. Defined values are:

Supersec	Meaning
0b0000	Supersections supported.
0b1111	Supersections not supported.

All other values are reserved.

In Armv8-A, the permitted values are `0b0000` and `0b1111`.

CMemSz, bits [27:24]

Cached Memory Size. Indicates the physical memory size supported by the caches. Defined values are:

CMemSz	Meaning
`0b0000`	4GB, corresponding to a 32-bit physical address range.
`0b0001`	64GB, corresponding to a 36-bit physical address range.
`0b0010`	1TB or more, corresponding to a 40-bit or larger physical address range.

All other values are reserved.

In Armv8-A, the permitted values are `0b0000`, `0b0001`, and `0b0010`.

CohWalk, bits [23:20]

Coherent Walk. Indicates whether Translation table updates require a clean to the Point of Unification. Defined values are:

CohWalk	Meaning
`0b0000`	Updates to the translation tables require a clean to the Point of Unification to ensure visibility by subsequent translation table walks.
`0b0001`	Updates to the translation tables do not require a clean to the Point of Unification to ensure visibility by subsequent translation table walks.

All other values are reserved.

In Armv8-A, the only permitted value is `0b0001`.

PAN, bits [19:16]

Privileged Access Never. Indicates support for the PAN bit in CPSR, SPSR, and DSPSR in AArch32 state. Defined values are:

PAN	Meaning
`0b0000`	PAN not supported.
`0b0001`	PAN supported.

`0b0010`	PAN supported and ATS1CPRP and ATS1CPWP instructions supported.

All other values are reserved.

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

FEAT_PAN2 implements the functionality added by the value `0b0010`.

In Armv8.1, the value `0b0000` is not permitted.

From Armv8.2, the only permitted value is `0b0010`.

MaintBcst, bits [15:12]

Maintenance Broadcast. Indicates whether Cache, TLB, and branch predictor operations are broadcast. Defined values are:

MaintBcst	Meaning
`0b0000`	Cache, TLB, and branch predictor operations only affect local structures.
`0b0001`	Cache and branch predictor operations affect structures according to shareability and defined behavior of instructions. TLB operations only affect local structures.
`0b0010`	Cache, TLB, and branch predictor operations affect structures according to shareability and defined behavior of instructions.

All other values are reserved.

In Armv8-A, the only permitted value is `0b0010`.

BPMaint, bits [11:8]

Branch Predictor Maintenance. Indicates the supported branch predictor maintenance operations in an implementation with hierarchical cache maintenance operations. Defined values are:

BPMaint	Meaning
`0b0000`	None supported.

`0b0001`	Supported branch predictor maintenance operations are: • Invalidate all branch predictors.
`0b0010`	As for `0b0001`, and adds: • Invalidate branch predictors by VA.

All other values are reserved.

In Armv8-A, the only permitted value is `0b0010`.

CMaintSW, bits [7:4]

Cache Maintenance by Set/Way. Indicates the supported cache maintenance operations by set/way, in an implementation with hierarchical caches. Defined values are:

CMaintSW	Meaning
`0b0000`	None supported.
`0b0001`	Supported hierarchical cache maintenance instructions by set/way are: • Invalidate data cache by set/way. • Clean data cache by set/way. • Clean and invalidate data cache by set/way.

All other values are reserved.

In Armv8-A, the only permitted value is `0b0001`.

In a unified cache implementation, the data cache maintenance operations apply to the unified caches.

CMaintVA, bits [3:0]

Cache Maintenance by Virtual Address. Indicates the supported cache maintenance operations by VA, in an implementation with hierarchical caches. Defined values are:

CMaintVA	Meaning
`0b0000`	None supported.

0b0001	Supported hierarchical cache maintenance operations by VA are: - Invalidate data cache by VA. - Clean data cache by VA. - Clean and invalidate data cache by VA. - Invalidate instruction cache by VA. - Invalidate all instruction cache entries.

All other values are reserved.

In Armv8-A, the only permitted value is `0b0001`.

In a unified cache implementation, data cache maintenance operations apply to the unified caches, and the instruction cache maintenance instructions are not implemented.

Otherwise:

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
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 [63:0]

Reserved, unknown.

Accessing ID_MMFR3_EL1

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

```
MRS <Xt>, ID_MMFR3_EL1
```

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0001	0b111

```
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_MMFR3_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ID_MMFR3_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ID_MMFR3_EL1;
```