ID_AA64DFR0_EL1, AArch64 Debug Feature Register 0

The ID_AA64DFR0_EL1 characteristics are:

Purpose

Provides top level information about the debug system 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

The external register EDDFR gives information from this register.

Attributes

ID_AA64DFR0_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
HPMN0	ExtTrcBuff	BRBE	MTPMU	TraceBuffer	TraceFilt	DoubleLock	PMSVer
CTX_CMPs	SEBEP	WRPs	PMSS	BRPs	PMUVer	TraceVer	DebugVer
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

HPMN0, bits [63:60]

Zero PMU event counters for a Guest operating system. Defined values are:

HPMN0	Meaning
0b0000	Setting MDCR_EL2.HPMN to zero has constrained unpredictable behavior.
0b0001	Setting MDCR_EL2.HPMN to zero has defined behavior.

All other values are reserved.

If FEAT_PMUv3 is not implemented, FEAT_FGT is not implemented, or EL2 is not implemented, the only permitted value is 0b0000.

FEAT_HPMN0 implements the functionality identified by the value `0b0001`.

From Armv8.8, in an implementation that includes FEAT_PMUv3, FEAT_FGT, and EL2, the value `0b0000` is not permitted.

ExtTrcBuff, bits [59:56]
When FEAT_TRBE is implemented:

Trace Buffer External Mode Extension. Defined values are:

ExtTrcBuff	Meaning
`0b0000`	Trace Buffer External Mode not implemented.
`0b0001`	Trace Buffer External Mode implemented.

All other values are reserved.

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

Otherwise:

Reserved, res0.

BRBE, bits [55:52]

Branch Record Buffer Extension. Defined values are:

BRBE	Meaning
`0b0000`	Branch Record Buffer Extension not implemented.
`0b0001`	Branch Record Buffer Extension implemented.
`0b0010`	As `0b0001`, and adds support for branch recording at EL3.

All other values are reserved.

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

FEAT_BRBEv1p1 implements the functionality identified by the value `0b0010`.

From Armv9.3, if FEAT_BRBE is implemented, the value `0b0001` is not permitted.

MTPMU, bits [51:48]

Multi-threaded PMU extension. Defined values are:

MTPMU	Meaning
0b0000	FEAT_MTPMU not implemented. If FEAT_PMUv3 is implemented, it is implementation defined whether PMEVTYPER<n>_EL0.MT and PMEVTYPER<n>.MT are read/write or res0.
0b0001	FEAT_MTPMU and FEAT_PMUv3 implemented. PMEVTYPER<n>_EL0.MT and PMEVTYPER<n>.MT are read/write. When FEAT_MTPMU is disabled, the Effective values of PMEVTYPER<n>_EL0.MT and PMEVTYPER<n>.MT are 0.
0b1111	FEAT_MTPMU not implemented. If FEAT_PMUv3 is implemented, PMEVTYPER<n>_EL0.MT and PMEVTYPER<n>.MT are res0.

All other values are reserved.

FEAT_MTPMU implements the functionality identified by the value 0b0001.

From Armv8.6, in an implementation that includes FEAT_PMUv3, the value 0b0000 is not permitted.

In an implementation that does not include FEAT_PMUv3, the value 0b0001 is not permitted.

TraceBuffer, bits [47:44]

Trace Buffer Extension. Defined values are:

TraceBuffer	Meaning
0b0000	Trace Buffer Extension not implemented.
0b0001	Trace Buffer Extension implemented.

All other values are reserved.

FEAT_TRBE implements the functionality identified by the value 0b0001.

In any Armv9 implementation, if FEAT_ETE is implemented, the value `0b0000` is not permitted.

TraceFilt, bits [43:40]

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

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

All other values are reserved.

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

From Armv8.4, if an Embedded Trace Macrocell Architecture trace unit is implemented, the value `0b0000` is not permitted.

DoubleLock, bits [39:36]

OS Double Lock implemented. Defined values are:

DoubleLock	Meaning
`0b0000`	OS Double Lock implemented. OSDLR_EL1 is RW.
`0b1111`	OS Double Lock not implemented. OSDLR_EL1 is RAZ/WI.

All other values are reserved.

FEAT_DoubleLock implements the functionality identified by the value `0b0000`.

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

If FEAT_Debugv8p2 is implemented and FEAT_DoPD is not implemented, the permitted values are `0b0000` and `0b1111`.

If FEAT_DoPD is implemented, the only permitted value is `0b1111`.

PMSVer, bits [35:32]

Statistical Profiling Extension version. Defined values are:

PMSVer	Meaning

0b0000	Statistical Profiling Extension not implemented.
0b0001	Statistical Profiling Extension implemented.
0b0010	As 0b0001, and adds: • Support for the Events packet Alignment flag. • If FEAT_SVE is implemented, support for the Scalable Vector extensions to Statistical Profiling.
0b0011	As 0b0010, and adds: • Discard mode. • Extended event filtering, including the PMSNEVFR_EL1 System register. • Support for the optional previous branch target Address packet. • If FEAT_PMUv3 is implemented, controls to freeze the PMU event counters after an SPE buffer management event occurs. • If FEAT_PMUv3 is implemented, the SAMPLE_FEED_BR, SAMPLE_FEED_EVENT, SAMPLE_FEED_LAT, SAMPLE_FEED_LD, SAMPLE_FEED_OP, and SAMPLE_FEED_ST PMU events.

`0b0100` As `0b0011`, and adds:

- If FEAT_MOPS is implemented, Operation Type packet encodings for Memory Copy and Set operations.
- If FEAT_MTE is implemented, Operation Type packet encodings for loads and stores of Allocation Tags.

`0b0101` As `0b0100`, and adds:

- Support for the Events packet Level 2 Data cache access, Level 2 Data cache miss, Cached data modified, Recently fetched cache line, and Cache snoop flags.
- Support for Data Source filtering.

All other values are reserved.

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

FEAT_SPEv1p1 implements the functionality identified by the value `0b0010`.

FEAT_SPEv1p2 implements the functionality identified by the value `0b0011`.

FEAT_SPEv1p3 implements the functionality identified by the value `0b0100`.

FEAT_SPEv1p4 implements the functionality identified by the value `0b0101`.

From Armv8.5, if FEAT_SPE is implemented, the value `0b0001` is not permitted.

From Armv8.7, if FEAT_SPE is implemented, the value `0b0010` is not permitted.

From Armv8.8, if FEAT_SPE is implemented, the value `0b0011` is not permitted.

From Armv8.9, if FEAT_SPE is implemented, the value `0b0100` is not permitted.

CTX_CMPs, bits [31:28]

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

The value of this field is never greater than ID_AA64DFR0_EL1.BRPs.

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]

Synchronous-exception-based event profiling. Defined values are:

SEBEP	Meaning
`0b0000`	Synchronous-exception-based event profiling not implemented.
`0b0001`	Synchronous-exception-based event profiling implemented.

All other values are reserved.

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

WRPs, bits [23:20]

Number of watchpoints, minus 1.

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]

PMU Snapshot extension. Defined values are:

PMSS	Meaning
`0b0000`	PMU snapshot extension not implemented.

0b0001	PMU snapshot extension implemented.

All other values are reserved.

FEAT_PMUv3_SS implements the functionality identified by the value 0b0001.

BRPs, bits [15:12]

Number of breakpoints, minus 1.

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 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 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.

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.

When trace unit System registers are implemented, see TRCIDR1 for tracing capabilities of the trace unit.

DebugVer, bits [3:0]

Debug architecture version. Indicates presence of Armv8 debug architecture. Defined values are:

DebugVer	Meaning
0b0110	Armv8 debug architecture.
0b0111	Armv8 debug architecture with Virtualization Host Extensions.
0b1000	Armv8.2 debug architecture, FEAT_Debugv8p2.
0b1001	Armv8.4 debug architecture, FEAT_Debugv8p4.
0b1010	Armv8.8 debug architecture, FEAT_Debugv8p8.
0b1011	Armv8.9 debug architecture, FEAT_Debugv8p9.

All other values are reserved.

FEAT_VHE adds the functionality identified by the value 0b0111.

FEAT_Debugv8p2 adds the functionality identified by the value 0b1000.

FEAT_Debugv8p4 adds the functionality identified by the value 0b1001.

FEAT_Debugv8p8 adds the functionality identified by the value 0b1010.

FEAT_Debugv8p9 adds the functionality identified by the value 0b1011.

From Armv8.1, when FEAT_VHE is implemented the value 0b0110 is not permitted.

From Armv8.2, the values 0b0110 and 0b0111 are not permitted.

From Armv8.4, the value 0b1000 is not permitted.

From Armv8.8, the value 0b1001 is not permitted.

From Armv8.9, the value 0b1010 is not permitted.

Accessing ID_AA64DFR0_EL1

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

MRS <Xt>, ID_AA64DFR0_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0101	0b000

```
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_AA64DFR0_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64DFR0_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64DFR0_EL1;
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94