

## ID\_ISAR3\_EL1, AArch32 Instruction Set Attribute Register 3

The ID\_ISAR3\_EL1 characteristics are:

### Purpose

Provides information about the instruction sets implemented by the PE in AArch32 state.

Must be interpreted with [ID\\_ISAR0\\_EL1](#), [ID\\_ISAR1\\_EL1](#), [ID\\_ISAR2\\_EL1](#), [ID\\_ISAR4\\_EL1](#), and [ID\\_ISAR5\\_EL1](#).

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\_ISAR3\_EL1 bits [31:0] are architecturally mapped to AArch32 System register [ID\\_ISAR3\[31:0\]](#).

### Attributes

ID\_ISAR3\_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																															
T32EE				TrueNOP				T32Copy				TabBranch				SynchPrim				SVC				SIMD				Saturate			
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.

#### T32EE, bits [31:28]

Indicates the implemented T32EE instructions. Defined values are:

T32EE	Meaning
0b0000	None implemented.

0b0001	Adds the ENTERX and LEAVEX instructions, and modifies the load behavior to include null checking.
--------	---

All other values are reserved.

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

### TrueNOP, bits [27:24]

Indicates the implemented true NOP instructions. Defined values are:

TrueNOP	Meaning
0b0000	None implemented. This means there are no NOP instructions that do not have any register dependencies.
0b0001	Adds true NOP instructions in both the T32 and A32 instruction sets. This also permits additional NOP-compatible hints.

All other values are reserved.

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

### T32Copy, bits [23:20]

Indicates the support for T32 non flag-setting MOV instructions. Defined values are:

T32Copy	Meaning
0b0000	Not supported. This means that in the T32 instruction set, encoding T1 of the MOV (register) instruction does not support a copy from a low register to a low register.
0b0001	Adds support for T32 instruction set encoding T1 of the MOV (register) instruction, copying from a low register to a low register.

All other values are reserved.

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

### TabBranch, bits [19:16]

Indicates the implemented Table Branch instructions in the T32 instruction set. Defined values are:

TabBranch	Meaning
0b0000	None implemented.
0b0001	Adds the TBB and TBH instructions.

All other values are reserved.

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

### SynchPrim, bits [15:12]

Used in conjunction with ID\_ISAR4.SynchPrim\_frac to indicate the implemented Synchronization Primitive instructions. Defined values are:

SynchPrim	Meaning
0b0000	If SynchPrim_frac == 0b0000, no Synchronization Primitives implemented.
0b0001	If SynchPrim_frac == 0b0000, adds the LDREX and STREX instructions. If SynchPrim_frac == 0b0011, also adds the CLREX, LDREXB, STREXB, and STREXH instructions.
0b0010	If SynchPrim_frac == 0b0000, as for [0b0001, 0b0011] and also adds the LDREXD and STREXD instructions.

All other combinations of SynchPrim and SynchPrim\_frac are reserved.

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

### SVC, bits [11:8]

Indicates the implemented SVC instructions. Defined values are:

SVC	Meaning
0b0000	Not implemented.
0b0001	Adds the SVC instruction.

All other values are reserved.

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

### **SIMD, bits [7:4]**

Indicates the implemented SIMD instructions. Defined values are:

<b>SIMD</b>	<b>Meaning</b>
0b0000	None implemented.
0b0001	Adds the SSAT and USAT instructions, and the Q bit in the PSRs.
0b0011	As for 0b0001, and adds the PKHBT, PKHTB, QADD16, QADD8, QASX, QSUB16, QSUB8, QSAX, SADD16, SADD8, SASX, SEL, SHADD16, SHADD8, SHASX, SHSUB16, SHSUB8, SHSAX, SSAT16, SSUB16, SSUB8, SSAX, SXTAB16, SXTB16, UADD16, UADD8, UASX, UHADD16, UHADD8, UHASX, UHSUB16, UHSUB8, UHSAX, UQADD16, UQADD8, UQASX, UQSUB16, UQSUB8, UQSAX, USAD8, USADA8, USAT16, USUB16, USUB8, USAX, UXTAB16, and UXTB16 instructions. Also adds support for the GE[3:0] bits in the PSRs.

All other values are reserved.

In Armv8-A, the only permitted value is 0b0011.

The SIMD field relates only to implemented instructions that perform SIMD operations on the general-purpose registers. In an implementation that supports Advanced SIMD and floating-point instructions, [MVFR0](#), [MVFR1](#), and [MVFR2](#) give information about the implemented Advanced SIMD instructions.

### **Saturate, bits [3:0]**

Indicates the implemented Saturate instructions. Defined values are:

<b>Saturate</b>	<b>Meaning</b>
-----------------	----------------

0b0000	None implemented. This means no non-Advanced SIMD saturate instructions are implemented.
0b0001	Adds the QADD, QDADD, QDSUB, and QSUB instructions, and the Q bit in the PSRs.

All other values are reserved.

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

## 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\_ISAR3\_EL1

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

MRS <Xt>, ID\_ISAR3\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0010	0b011

```

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_ISAR3_EL1;
    elsif PSTATE.EL == EL2 then
        X[t, 64] = ID_ISAR3_EL1;
    elsif PSTATE.EL == EL3 then

```

```
X[t, 64] = ID_ISAR3_EL1;
```

---

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

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.