ID_ISAR4_EL1, AArch32 Instruction Set Attribute Register 4

The ID_ISAR4_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_ISAR3_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_ISAR4_EL1 bits [31:0] are architecturally mapped to AArch32 System register ID_ISAR4[31:0].

Attributes

ID_ISAR4_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																															
SWP_frac				PSR_M				SynchPrim_frac				Barrier				SMC				Writeback				WithShifts				Unpriv			
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.

SWP_frac, bits [31:28]

Indicates support for the memory system locking the bus for SWP or SWPB instructions. Defined values are:

SWP_frac	Meaning

0b0000	SWP or SWPB instructions not implemented.
0b0001	SWP or SWPB implemented but only in a uniprocessor context. SWP and SWPB do not guarantee whether memory accesses from other Requesters can come between the load memory access and the store memory access of the SWP or SWPB.

All other values are reserved. This field is valid only if ID_ISAR0.Swap is 0b0000.

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

PSR_M, bits [27:24]

Indicates the implemented M-profile instructions to modify the PSRs. Defined values are:

PSR_M	Meaning
0b0000	None implemented.
0b0001	Adds the M-profile forms of the CPS, MRS, and MSR instructions.

All other values are reserved.

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

SynchPrim_frac, bits [23:20]

Used in conjunction with ID_ISAR3.SynchPrim to indicate the implemented Synchronization Primitive instructions. Possible values are:

SynchPrim_frac	Meaning

0b0000	If SynchPrim == 0b0000, no Synchronization Primitives implemented. If SynchPrim == 0b0001, adds the LDREX and STREX instructions. If SynchPrim == 0b0010, also adds the CLREX, LDREXB, LDREXH, STREXB, STREXH, LDREXD, and STREXD instructions.
0b0011	If SynchPrim == 0b0001, adds the LDREX, STREX, CLREX, LDREXB, LDREXH, STREXB, and STREXH instructions.

All other combinations of SynchPrim and SynchPrim_frac are reserved.

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

Barrier, bits [19:16]

Indicates the implemented Barrier instructions in the A32 and T32 instruction sets. Defined values are:

Barrier	Meaning
0b0000	None implemented. Barrier operations are provided only as System instructions in the (coproc==0b1111) encoding space.
0b0001	Adds the DMB, DSB, and ISB barrier instructions.

All other values are reserved.

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

SMC, bits [15:12]

Indicates the implemented SMC instructions. Defined values are:

SMC	Meaning
0b0000	None implemented.
0b0001	Adds the SMC instruction.

All other values are reserved.

In Armv8-A, the permitted values are:

- If EL3 is implemented and EL1 can use AArch32, the only permitted value is 0b0001.
- If neither EL3 nor EL2 is implemented, the only permitted value is 0b0000.

If EL1 cannot use AArch32, this field has the value 0b0000.

Writeback, bits [11:8]

Indicates the support for Writeback addressing modes. Defined values are:

Writeback	Meaning
0b0000	Basic support. Only the LDM, STM, PUSH, POP, SRS, and RFE instructions support writeback addressing modes. These instructions support all of their writeback addressing modes.
0b0001	Adds support for all of the writeback addressing modes.

All other values are reserved.

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

WithShifts, bits [7:4]

Indicates the support for instructions with shifts. Defined values are:

WithShifts	Meaning
0b0000	Nonzero shifts supported only in MOV and shift instructions.
0b0001	Adds support for shifts of loads and stores over the range LSL 0-3.

0b0011	As for 0b0001, and adds support for other constant shift options, both on load/store and other instructions.
0b0100	As for 0b0011, and adds support for register-controlled shift options.

All other values are reserved.

In Armv8-A, the only permitted value is 0b0100.

Unpriv, bits [3:0]

Indicates the implemented unprivileged instructions. Defined values are:

Unpriv	Meaning
0b0000	None implemented. No T variant instructions are implemented.
0b0001	Adds the LDRBT, LDRT, STRBT, and STRT instructions.
0b0010	As for 0b0001, and adds the LDRHT, LDRSBT, LDRSHT, and STRHT instructions.

All other values are reserved.

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

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_ISAR4_EL1

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

MRS <Xt>, ID_ISAR4_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0010	0b100

```
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_ISAR4_EL1;
elsif PSTATE.EL == EL2 then
    X[t, 64] = ID_ISAR4_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = ID_ISAR4_EL1;
```

28/03/2023 16:01; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94