

# AFSR1\_EL1, Auxiliary Fault Status Register 1 (EL1)

The AFSR1\_EL1 characteristics are:

## Purpose

Provides additional implementation defined fault status information for exceptions taken to EL1.

## Configuration

AArch64 System register AFSR1\_EL1 bits [31:0] are architecturally mapped to AArch32 System register [AIFSR\[31:0\]](#).

## Attributes

AFSR1\_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
IMPLEMENTATION DEFINED																															
IMPLEMENTATION DEFINED																															
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

### IMPLEMENTATION DEFINED, bits [63:0]

implementation defined.

The reset behavior of this field is:

- On a Warm reset, this field resets to an architecturally unknown value.

## Accessing AFSR1\_EL1

When [HCR\\_EL2.E2H](#) is 1, without explicit synchronization, access from EL3 using the mnemonic AFSR1\_EL1 or AFSR1\_EL12 are not guaranteed to be ordered with respect to accesses using the other mnemonic.

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

## MRS <Xt>, AFSR1\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0101	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TRVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGTR_EL2.AFSR1_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
        X[t, 64] = NVMem[0x130];
    else
        X[t, 64] = AFSR1_EL1;
elseif PSTATE.EL == EL2 then
    if HCR_EL2.E2H == '1' then
        X[t, 64] = AFSR1_EL2;
    else
        X[t, 64] = AFSR1_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = AFSR1_EL1;
```

## MSR AFSR1\_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b0101	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TVM == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() &&
        IsFeatureImplemented(FEAT_FGT) && (!HaveEL(EL3) ||
        SCR_EL3.FGTEn == '1') && HFGWTR_EL2.AFSR1_EL1 == '1'
    then
        AArch64.SystemAccessTrap(EL2, 0x18);
    elseif EL2Enabled() && HCR_EL2.<NV2,NV1,NV> ==
        '111' then
        NVMem[0x130] = X[t, 64];
    else
        AFSR1_EL1 = X[t, 64];
elseif PSTATE.EL == EL2 then
```

```

        if HCR_EL2.E2H == '1' then
            AFSR1_EL2 = X[t, 64];
        else
            AFSR1_EL1 = X[t, 64];
    elsif PSTATE.EL == EL3 then
        AFSR1_EL1 = X[t, 64];

```

## MRS <Xt>, AFSR1\_EL12

op0	op1	CRn	CRm	op2
0b11	0b101	0b0101	0b0001	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
    then
        X[t, 64] = NVMem[0x130];
        elsif EL2Enabled() && HCR_EL2.NV == '1' then
            AArch64.SystemAccessTrap(EL2, 0x18);
        else
            UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' then
            X[t, 64] = AFSR1_EL1;
        else
            UNDEFINED;
    elsif PSTATE.EL == EL3 then
        if EL2Enabled() && !ELUsingAArch32(EL2) &&
        HCR_EL2.E2H == '1' then
            X[t, 64] = AFSR1_EL1;
        else
            UNDEFINED;

```

## MSR AFSR1\_EL12, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b101	0b0101	0b0001	0b001

```

if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV1,NV> == '101'
    then
        NVMem[0x130] = X[t, 64];
        elsif EL2Enabled() && HCR_EL2.NV == '1' then

```

```

        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
    elsif PSTATE.EL == EL2 then
        if HCR_EL2.E2H == '1' then
            AFSR1_EL1 = X[t, 64];
        else
            UNDEFINED;
        elsif PSTATE.EL == EL3 then
            if EL2Enabled() && !ELUsingAArch32(EL2) &&
HCR_EL2.E2H == '1' then
                AFSR1_EL1 = X[t, 64];
            else
                UNDEFINED;

```

---

[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.