DISR_EL1, Deferred Interrupt Status Register

The DISR_EL1 characteristics are:

Purpose

Records that an SError interrupt has been consumed by an `ESB` instruction.

Configuration

AArch64 System register DISR_EL1 bits [31:0] are architecturally mapped to AArch32 System register DISR[31:0].

This register is present only when FEAT_RAS is implemented. Otherwise, direct accesses to DISR_EL1 are undefined.

Attributes

DISR_EL1 is a 64-bit register.

Field descriptions

When DISR_EL1.IDS == 0:

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																															

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
A	RES0						IDS	RES0											AET			EA	RES0			DFSC					

Bits [63:32]

Reserved, res0.

A, bit [31]

Set to 1 when an `ESB` instruction defers an asynchronous SError interrupt. If the implementation does not include any sources of SError interrupt that can be synchronized by an Error Synchronization Barrier, then this bit is res0.

The reset behavior of this field is:

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

Bits [30:25]

Reserved, res0.

IDS, bit [24]

Indicates the deferred SError interrupt type.

IDS	Meaning
0b0	Deferred error uses architecturally-defined format.

The reset behavior of this field is:

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

Bits [23:13]

Reserved, res0.

AET, bits [12:10]

Asynchronous Error Type. See the description of ESR_ELx.AET for an SError interrupt.

The reset behavior of this field is:

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

EA, bit [9]

External abort Type. See the description of ESR_ELx.EA for an SError interrupt.

The reset behavior of this field is:

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

Bits [8:6]

Reserved, res0.

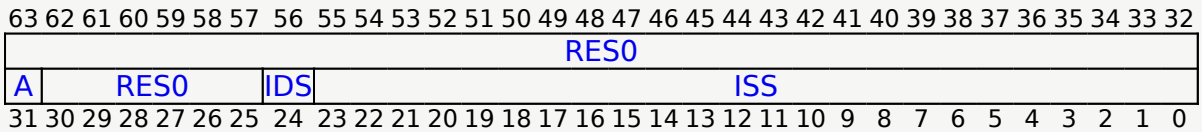
DFSC, bits [5:0]

Fault Status Code. See the description of ESR_ELx.DFSC for an SError interrupt.

The reset behavior of this field is:

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

When DISR_EL1.IDS == 1:



Bits [63:32]

Reserved, res0.

A, bit [31]

Set to 1 when an `ESB` instruction defers an asynchronous SError interrupt. If the implementation does not include any sources of SError interrupt that can be synchronized by an Error Synchronization Barrier, then this bit is res0.

The reset behavior of this field is:

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

Bits [30:25]

Reserved, res0.

IDS, bit [24]

Indicates the deferred SError interrupt type.

IDS	Meaning
`0b1`	Deferred error uses implementation defined format.

The reset behavior of this field is:

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

ISS, bits [23:0]

implementation defined syndrome. See the description of ESR_ELx[23:0] for an SError interrupt.

The reset behavior of this field is:

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

Accessing DISR_EL1

An indirect write to DISR_EL1 made by an `ESB` instruction does not require an explicit synchronization operation for the value that is written to be observed by a direct read of DISR_EL1 occurring in program order after the `ESB` instruction.

DISR_EL1 is RAZ/WI if EL3 is implemented, the PE is in Non-debug state, SCR_EL3.EA == 1, and any of the following apply:

- At EL2.
- At EL1 and ((SCR_EL3.NS == 0 && SCR_EL3.EEL2 == 0) || HCR_EL2.AMO == 0).

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

MRS <Xt>, DISR_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.AMO == '1' then
        X[t, 64] = VDISR_EL2;
    elsif HaveEL(EL3) && !Halted() && SCR_EL3.EA ==
'1' then
        X[t, 64] = Zeros(64);
    else
        X[t, 64] = DISR_EL1;
elsif PSTATE.EL == EL2 then
    if HaveEL(EL3) && !Halted() && SCR_EL3.EA == '1'
then
        X[t, 64] = Zeros(64);
    else
        X[t, 64] = DISR_EL1;
elsif PSTATE.EL == EL3 then
    X[t, 64] = DISR_EL1;
```

MSR DISR_EL1, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b000	0b1100	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.AMO == '1' then
        VDISR_EL2 = X[t, 64];
    elsif HaveEL(EL3) && !Halted() && SCR_EL3.EA ==
'1' then
        return;
    else
        DISR_EL1 = X[t, 64];
elsif PSTATE.EL == EL2 then
    if HaveEL(EL3) && !Halted() && SCR_EL3.EA == '1'
then
        return;
    else
        DISR_EL1 = X[t, 64];
elsif PSTATE.EL == EL3 then
    DISR_EL1 = X[t, 64];
```

28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94