VDISR_EL2, Virtual Deferred Interrupt Status Register

The VDISR_EL2 characteristics are:

Purpose

Records that a virtual SError interrupt has been consumed by an `ESB` instruction executed at EL1.

An indirect write to VDISR_EL2 made by an `ESB` instruction does not require an explicit synchronization operation for the value written to be observed by a direct read of one of the following registers occurring in program order after the `ESB` instruction:

- DISR_EL1.
- DISR.

Configuration

AArch64 System register VDISR_EL2 bits [31:0] are architecturally mapped to AArch32 System register VDISR[31:0].

This register is present only when FEAT_RAS is implemented. Otherwise, direct accesses to VDISR_EL2 are undefined.

If EL2 is not implemented, this register is res0 from EL3.

This register has no effect if EL2 is not enabled in the current Security state.

Attributes

VDISR_EL2 is a 64-bit register.

Field descriptions

When EL1 is using AArch64:

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																															
A	RES0																		IDS	ISS											
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.

A, bit [31]

Set to 1 when an ESB instruction defers a virtual SError interrupt.

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]

The value copied from VSESR_EL2.IDS.

The reset behavior of this field is:

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

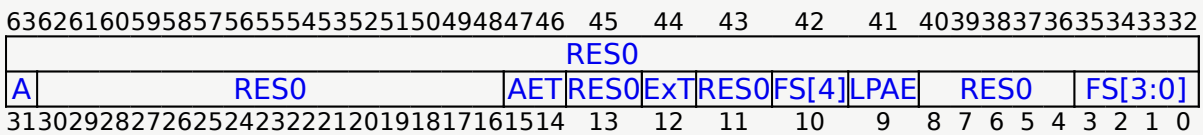
ISS, bits [23:0]

The value copied from VSESR_EL2.ISS.

The reset behavior of this field is:

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

When EL1 is using AArch32 and VDISR_EL2.LPAE == 0:



Bits [63:32]

Reserved, res0.

A, bit [31]

Set to 1 when an ESB instruction defers a virtual SError interrupt.

The reset behavior of this field is:

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

Bits [30:16]

Reserved, res0.

AET, bits [15:14]

The value copied from VSESR_EL2.AET.

The reset behavior of this field is:

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

Bit [13]

Reserved, res0.

ExT, bit [12]

The value copied from VSESR_EL2.ExT.

The reset behavior of this field is:

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

Bit [11]

Reserved, res0.

FS, bits [10, 3:0]

Fault status code. Set to `0b10110` when an `ESB` instruction defers a virtual SError interrupt.

FS	Meaning
`0b10110`	Asynchronous SError interrupt.

All other values are reserved.

The FS field is split as follows:

- FS[4] is VDISR_EL2[10].
- FS[3:0] is VDISR_EL2[3:0].

The reset behavior of this field is:

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

LPAE, bit [9]

Format.

Set to TTBCR.EAE when an `ESB` instruction defers a virtual SError interrupt.

LPAE	Meaning
0b0	Using the Short-descriptor translation table format.

The reset behavior of this field is:

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

Bits [8:4]

Reserved, res0.

When EL1 is using AArch32 and VDISR_EL2.LPAE == 1:

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																															
A	RES0																		AET		RES0		ExT		RES0		LPAE		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

Bits [63:32]

Reserved, res0.

A, bit [31]

Set to 1 when an `ESB` instruction defers a virtual SError interrupt.

The reset behavior of this field is:

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

Bits [30:16]

Reserved, res0.

AET, bits [15:14]

The value copied from VSESR_EL2.AET.

The reset behavior of this field is:

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

Bit [13]

Reserved, res0.

ExT, bit [12]

The value copied from VSESR_EL2.ExT.

The reset behavior of this field is:

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

Bits [11:10]

Reserved, res0.

LPAE, bit [9]

Format.

Set to TTBCR.EAE when an `ESB` instruction defers a virtual SError interrupt.

LPAE	Meaning
`0b1`	Using the Long-descriptor translation table format.

The reset behavior of this field is:

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

Bits [8:6]

Reserved, res0.

STATUS, bits [5:0]

Fault status code. Set to `0b010001` when an `ESB` instruction defers a virtual SError interrupt.

STATUS	Meaning
`0b010001`	Asynchronous SError interrupt.

All other values are reserved.

The reset behavior of this field is:

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

Accessing VDISR_EL2

An indirect write to VDISR_EL2 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 one of the following registers occurring in program order after the `ESB` instruction:

- DISR_EL1.
- DISR.

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

MRS <Xt>, VDISR_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b1100	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        X[t, 64] = NVMem[0x500];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    X[t, 64] = VDISR_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = VDISR_EL2;
```

MSR VDISR_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b1100	0b0001	0b001

```
if PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.<NV2,NV> == '11' then
        NVMem[0x500] = X[t, 64];
    elsif EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    VDISR_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    VDISR_EL2 = X[t, 64];
```

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