DBGVCR32_EL2, Debug Vector Catch Register

The DBGVCR32_EL2 characteristics are:

Purpose

Allows access to the AArch32 register DBGVCR from AArch64 state only. Its value has no effect on execution in AArch64 state.

Configuration

AArch64 System register DBGVCR32_EL2 bits [31:0] are architecturally mapped to AArch32 System register DBGVCR[31:0].

This register is present only when EL1 is capable of using AArch32. Otherwise, direct accesses to DBGVCR32_EL2 are undefined.

If EL2 is not implemented but EL3 is implemented, and EL1 is capable of using AArch32, then this register is not res0.

Attributes

DBGVCR32_EL2 is a 64-bit register.

Field descriptions

When EL3 is implemented:

63	62	61	60	59	58	57	56:39	38	37	36	35	34	33	32
RES0														
NSF	NSI	RES0	NSD	NSP	NSS	NSU	RES0	SF	SI	RES0	SD	SP	SS	SU	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.

NSF, bit [31]

FIQ vector catch enable in Non-secure state.

The exception vector offset is `0x1C`.

The reset behavior of this field is:

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

NSI, bit [30]

IRQ vector catch enable in Non-secure state.

The exception vector offset is `0x18`.

The reset behavior of this field is:

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

Bit [29]

Reserved, res0.

NSD, bit [28]

Data Abort exception vector catch enable in Non-secure state.

The exception vector offset is `0x10`.

The reset behavior of this field is:

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

NSP, bit [27]

Prefetch Abort vector catch enable in Non-secure state.

The exception vector offset is `0x0C`.

The reset behavior of this field is:

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

NSS, bit [26]

Supervisor Call (SVC) vector catch enable in Non-secure state.

The exception vector offset is `0x08`.

The reset behavior of this field is:

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

NSU, bit [25]

Undefined Instruction vector catch enable in Non-secure state.

The exception vector offset is `0x04`.

The reset behavior of this field is:

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

Bits [24:8]

Reserved, res0.

SF, bit [7]

FIQ vector catch enable in Secure state.

The exception vector offset is `0x1C`.

The reset behavior of this field is:

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

SI, bit [6]

IRQ vector catch enable in Secure state.

The exception vector offset is `0x18`.

The reset behavior of this field is:

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

Bit [5]

Reserved, res0.

SD, bit [4]

Data Abort exception vector catch enable in Secure state.

The exception vector offset is `0x10`.

The reset behavior of this field is:

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

SP, bit [3]

Prefetch Abort vector catch enable in Secure state.

The exception vector offset is `0x0C`.

The reset behavior of this field is:

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

SS, bit [2]

Supervisor Call (SVC) vector catch enable in Secure state.

The exception vector offset is `0x08`.

The reset behavior of this field is:

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

SU, bit [1]

Undefined Instruction vector catch enable in Secure state.

The exception vector offset is `0x04`.

The reset behavior of this field is:

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

Bit [0]

Reserved, res0.

When EL3 is not implemented:

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																															
RES0																								F	I	RES0		D	P	S	U	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 [63:8]

Reserved, res0.

F, bit [7]

FIQ vector catch enable.

The exception vector offset is `0x1C`.

The reset behavior of this field is:

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

I, bit [6]

IRQ vector catch enable.

The exception vector offset is `0x18`.

The reset behavior of this field is:

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

Bit [5]

Reserved, res0.

D, bit [4]

Data Abort exception vector catch enable.

The exception vector offset is `0x10`.

The reset behavior of this field is:

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

P, bit [3]

Prefetch Abort vector catch enable.

The exception vector offset `0x0C`.

The reset behavior of this field is:

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

S, bit [2]

Supervisor Call (SVC) vector catch enable.

The exception vector offset is `0x08`.

The reset behavior of this field is:

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

U, bit [1]

Undefined Instruction vector catch enable.

The exception vector offset is `0x04`.

The reset behavior of this field is:

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

Bit [0]

Reserved, res0.

Accessing DBGVCR32_EL2

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

MRS <Xt>, DBGVCR32_EL2

op0	op1	CRn	CRm	op2
0b10	0b100	0b0000	0b0111	0b000

```
if !HaveAArch32EL(EL1) then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.TDA == '1' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.TDA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        X[t, 64] = DBGVCR32_EL2;
elsif PSTATE.EL == EL3 then
    X[t, 64] = DBGVCR32_EL2;
```

MSR DBGVCR32_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b10	0b100	0b0000	0b0111	0b000

```
if !HaveAArch32EL(EL1) then
    UNDEFINED;
elsif PSTATE.EL == EL0 then
    UNDEFINED;
elsif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.NV == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        UNDEFINED;
elsif PSTATE.EL == EL2 then
    if Halted() && HaveEL(EL3) && EDSCR.SDD == '1'
&& boolean IMPLEMENTATION_DEFINED "EL3 trap priority
when SDD == '1'" && MDCR_EL3.TDA == '1' then
        UNDEFINED;
    elsif HaveEL(EL3) && MDCR_EL3.TDA == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x18);
    else
        DBGVCR32_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    DBGVCR32_EL2 = X[t, 64];
```


28/03/2023 16:02; 72747e43966d6b97dcbd230a1b3f0421d1ea3d94