FPEXC32_EL2, Floating-Point Exception Control Register

The FPEXC32_EL2 characteristics are:

Purpose

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

Configuration

AArch64 System register FPEXC32_EL2 bits [31:0] are architecturally mapped to AArch32 System register FPEXC[31:0].

This register is present only when EL1 is capable of using AArch32. Otherwise, direct accesses to FPEXC32_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.

Implemented only if the implementation includes the Advanced SIMD and floating-point functionality.

Attributes

FPEXC32_EL2 is a 64-bit register.

Field descriptions

63	62	61	60	59	58	575655545352515049484746454443	42 41 40	39	38 37	36	35	34	33	32
RES0														
EX	EN	DEX	FP2V	VV	TFV	RES0	VECITR	IDF	RES0	IXF	UFF	OFF	DZF	IOF
31	30	29	28	27	26	252423222120191817161514131211	10 9 8	7	6 5	4	3	2	1	0

Bits [63:32]

Reserved, res0.

EX, bit [31]

Exception bit.

The reset behavior of this field is:

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

Access to this field is **RAZ/WI**.

EN, bit [30]

Enables access to the Advanced SIMD and floating-point functionality from all Exception levels, except that setting this field to 0 does not disable the following:

- VMSR accesses to the FPEXC or FPSID.
- VMRS accesses from the FPEXC, FPSID, MVFR0, MVFR1, or MVFR2.

EN	Meaning
`0b0`	Accesses to the FPSCR, and any of the SIMD and floating-point registers Q0-Q15, including their views as D0-D31 registers or S0-S31 registers, are undefined at all Exception levels.
`0b1`	This control permits access to the Advanced SIMD and floating-point functionality at all Exception levels.

Execution of Advanced SIMD and floating-point instructions in AArch32 state can be disabled or trapped by the following controls:

- CPACR.cp10, or, if executing at EL0, CPACR_EL1.FPEN.
- FPEXC.EN.
- If executing in Non-secure state:
    - HCPTR.TCP10, or if EL2 is using AArch64, CPTR_EL2.TFP.
    - NSACR.cp10, or if EL3 is using AArch64, CPTR_EL3.TFP.
- For Advanced SIMD instructions only:
    - CPACR.ASEDIS.
    - If executing in Non-secure state, HCPTR.TASE and NSACR.NSTRCDIS.

See the descriptions of the controls for more information.

Note

When executing at EL0 using AArch32:

- If EL1 is using AArch64, then the Effective value of FPEXC.EN is 1.
- If EL2 is using AArch64 and is enabled in the current Security state,

HCR_EL2.TGE is 1, and the Effective value of HCR_EL2.RW is 1, then the Effective value of FPEXC.EN is 1. However, Arm deprecates using the value of FPEXC32_EL2.EN to determine behavior.

The reset behavior of this field is:

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

DEX, bit [29]

Defined synchronous exception on floating-point execution.

This field identifies whether a synchronous exception generated by the attempted execution of an instruction was generated by an unallocated encoding. The instruction must be in the encoding space that is identified by the pseudocode function ExecutingCP10or11Instr() returning TRUE. This field also indicates whether the FPEXC32_EL2.TFV field is valid.

The meaning of this bit is:

DEX	Meaning
`0b0`	The exception was generated by the attempted execution of an unallocated instruction in the encoding space that is identified by the pseudocode function ExecutingCP10or11Instr(). If FPEXC32_EL2.TFV is RW then it is invalid and unknown. If FPEXC32_EL2.{IDF, IXF, UFF, OFF, DZF, IOF} are RW then they are invalid and unknown.
`0b1`	The exception was generated during the execution of an allocated encoding. FPEXC32_EL2.TFV is valid and indicates the cause of the exception.

On an exception that sets this bit to 1 the exception-handling routine must clear this bit to 0.

On an implementation that both does not support trapping of floating-point exceptions and implements the AArch32 FPSCR.{Stride, Len} fields as RAZ, this bit is res0.

The reset behavior of this field is:

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

FP2V, bit [28]

FPINST2 instruction valid bit. From Armv8, this bit is res0.

The reset behavior of this field is:

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

Access to this field is **RES0**.

VV, bit [27]

VECITR valid bit. From Armv8, this bit is res0.

The reset behavior of this field is:

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

Access to this field is **RES0**.

TFV, bit [26]

Trapped Fault Valid bit. Valid only when the value of FPEXC32_EL2.DEX is 1. When valid, it indicates the cause of the exception and therefore whether FPEXC32_EL2.{IDF, IXF, UFF, OFF, DZF, IOF} are valid.

TFV	Meaning
`0b0`	The exception was caused by the execution of a floating-point VABS, VADD, VDIV, VFMA, VFMS, VFNMA, VFNMS, VMLA, VMLS, VMOV, VMUL, VNEG, VNMLA, VNMLS, VNMUL, VSQRT, or VSUB instruction when one or both of FPSCR.{Stride, Len} was nonzero. If FPEXC32_EL2.{IDF, IXF, UFF, OFF, DZF, IOF} are RW then they are invalid and unknown.

0b1	FPEXC32_EL2.{IDF, IXF, UFF, OFF, DZF, IOF} indicate the presence of trapped floating-point exceptions that had occurred at the time of the exception. Bits are set for all trapped exceptions that had occurred at the time of the exception.

This bit returns a status value and ignores writes.

When the value of FPEXC32_EL2.DEX is 0 and this bit is RW, this bit is invalid and unknown.

The reset behavior of this field is:

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

Accessing this field has the following behavior:

- When !ImpDefBool("Support trapping of floating-point exceptions"), access to this field is **RAZ/WI**.
- When ImpDefBool("Implemented FPSCR LEN, STRIDE as RAZ"), access to this field is **RAO/WI**.

Bits [25:11]

Reserved, res0.

VECITR, bits [10:8]

Vector iteration count. From Armv8, this field is res1.

The reset behavior of this field is:

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

Access to this field is **RES1**.

IDF, bit [7]

Input Denormal trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether an Input Denormal exception occurred while FPSCR.IDE was 1:

IDF	Meaning
0b0	Input Denormal exception has not occurred.
0b1	Input Denormal exception has occurred.

Input Denormal exceptions can occur only when FPSCR.FZ is 1.

Note

> A half-precision floating-point value that is flushed to zero because the value of FPSCR.FZ16 is 1 does not generate an Input Denormal exception.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC32_EL2.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

Bits [6:5]

Reserved, res0.

IXF, bit [4]

Inexact trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether an Inexact exception occurred while FPSCR.IXE was 1:

IXF	Meaning
`0b0`	Inexact exception has not occurred.
`0b1`	Inexact exception has occurred.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

UFF, bit [3]

Underflow trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether an Underflow exception occurred while FPSCR.UFE was 1:

UFF	Meaning
0b0	Underflow exception has not occurred.
0b1	Underflow exception has occurred.

Underflow trapped exceptions can occur:

- On half-precision data-processing instructions only when FPSCR.FZ16 is 0.
- Otherwise only when FPSCR.FZ is 0.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC32_EL2.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

OFF, bit [2]

Overflow trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether an Overflow exception occurred while FPSCR.OFE was 1:

OFF	Meaning
0b0	Overflow exception has not occurred.
0b1	Overflow exception has occurred.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

DZF, bit [1]

Divide by Zero trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether a Divide by Zero exception occurred while FPSCR.DZE was 1:

DZF	Meaning
`0b0`	Divide by Zero exception has not occurred.
`0b1`	Divide by Zero exception has occurred.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

IOF, bit [0]

Invalid Operation trapped exception bit. Valid only when the value of FPEXC.TFV is 1. When valid, it indicates whether an Invalid Operation exception occurred while FPSCR.IOE was 1:

IOF	Meaning
`0b0`	Invalid Operation exception has not occurred.
`0b1`	Invalid Operation exception has occurred.

This bit must be cleared to 0 by the exception-handling routine.

When the value of FPEXC.TFV is 0 and this bit is RW, this bit is invalid and unknown.

On an implementation that does not support the trapping of floating-point exceptions this bit is RAZ/WI.

The reset behavior of this field is:

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

Accessing FPEXC32_EL2

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

MRS <Xt>, FPEXC32_EL2

op0	op1	CRn	CRm	op2
0b11	0b100	0b0101	0b0011	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'" && CPTR_EL3.TFP == '1' then
        UNDEFINED;
    elsif HCR_EL2.E2H == '0' && CPTR_EL2.TFP == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elsif HCR_EL2.E2H == '1' && CPTR_EL2.FPEN ==
'x0' then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elsif HaveEL(EL3) && CPTR_EL3.TFP == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x07);
    else
        X[t, 64] = FPEXC32_EL2;
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TFP == '1' then
        AArch64.SystemAccessTrap(EL3, 0x07);
    else
        X[t, 64] = FPEXC32_EL2;
```

MSR FPEXC32_EL2, <Xt>

op0	op1	CRn	CRm	op2
0b11	0b100	0b0101	0b0011	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'" && CPTR_EL3.TFP == '1' then
        UNDEFINED;
    elsif HCR_EL2.E2H == '0' && CPTR_EL2.TFP == '1'
then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elsif HCR_EL2.E2H == '1' && CPTR_EL2.FPEN ==
'x0' then
        AArch64.SystemAccessTrap(EL2, 0x07);
    elsif HaveEL(EL3) && CPTR_EL3.TFP == '1' then
        if Halted() && EDSCR.SDD == '1' then
            UNDEFINED;
        else
            AArch64.SystemAccessTrap(EL3, 0x07);
    else
        FPEXC32_EL2 = X[t, 64];
elsif PSTATE.EL == EL3 then
    if CPTR_EL3.TFP == '1' then
        AArch64.SystemAccessTrap(EL3, 0x07);
    else
        FPEXC32_EL2 = X[t, 64];
```

AArch32 Registers AArch64 Registers AArch32 Instructions AArch64 Instructions Index by Encoding External Registers

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.